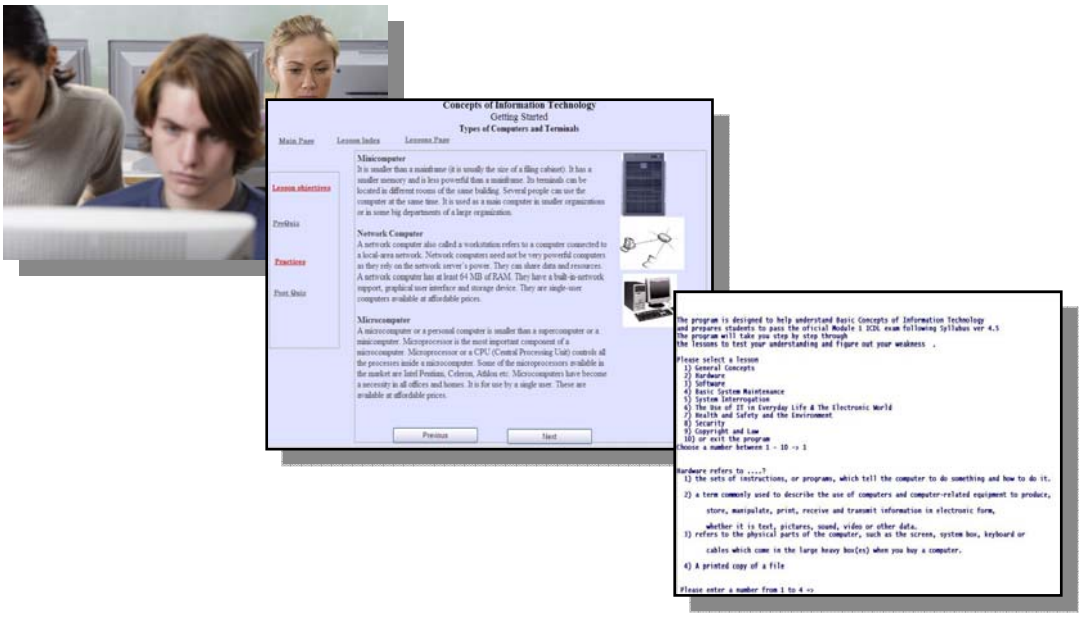


Dissertation Title:

An Online Cognitive, Adaptive, Practice based Approach for ITS: An Approach to IT professional certifications

Supervised by: Dr. Khaled Shaalan

Done by: Hikmat Jadalla Duweik

Student No: 20050032

Department of Informatics

Dissertation for the Msc. in Information Technology

October 2008

DISSERTATION RELEASE FORM

Student Name	Student ID	Programme	Date
Hikmat Jadalla Duweik	20050032	MSc. in Information Technology	30/10/2008

Title
An Online Cognitive, Adaptive, Practice based Approach for ITS: An Approach to IT professional certifications

I warrant that the content of this dissertation is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that one copy of my dissertation will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make that copy available in digital format if appropriate.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my dissertation for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

Signature
Hikmat Duweik

Acknowledgement

I would like to express my deep and sincere gratitude to all those who gave me the possibility to complete this thesis. I deeply indebted to my supervisor Dr. Khaled Shaalan, Senior Lecturer, Faculty of Informatics, The British University in Dubai (BUiD) whose wide knowledge, logical way of thinking, encouragement and support were very important and helped me during all the time of research for and the write-up of this thesis. I have furthermore to thank Lynn Randall, Study Skills Support, The British University in Dubai (BUiD), for her support, time and valuable advice.

I would like to thank my instructors, professors and all the people in The British University in Dubai who taught me and helped me to complete my study program.

I'm grateful to the teachers and the school students who helped me and gave me their valuable time and feedback during the survey and testing phases.

My warm sincere thanks are to my parents, whose constant encouragement, support and love have helped me during the difficult times.

Especially, and most importantly, I would like to give my special thanks to my husband Ghassan and my kids Yasmeen, Noor, Anass and Eman for their encouragement, understanding, and patience. They provided the comfortable environment which enabled me to complete this work. To them I dedicate this thesis.

Abstract

With the development of WWW technology, web-based Intelligent Tutoring Systems (ITSs) are becoming an important area of research and development. The key benefit of web-based ITS is that it can be accessed by thousands of learners all over the world. Even though many web-based educational systems have been developed recently, most of them are restricted in functionality, do not provide direction and guidance for individualized learning, and thus fail to take advantage of the distributed nature of the Internet.

In this thesis, we present an approach to a practice-based adaptable web-based cognitive intelligent tutoring system that has been applied to build a system that prepares learners for the exam of the International Certificate of ICDL official exam for module 1 (Basic Concepts of Information Technology). ICDL has recently become a de facto standard professional certificate in a wide spectrum of jobs. The system can be used for distance learning/ e-learning or for blended learning approach. It consists of two main modes: the traditional learning mode and the intelligent mode. The traditional learning mode facilitates the learning of concepts by providing full lessons with a fair amount of questions divided between pre lesson quizzes, post lesson quizzes and end of module assessment. The intelligent interactive mode allows the learner to practice concept understanding of the subject matter through means of adaptive self-assessment practices/tests. This is in addition to many other system features including tracking, reporting, assessment/ evaluation tools and some e-communication services and methods. Teachers and students have different interfaces, means of communication, and authorities. The current system consists of nine full lessons with more than 400 questions. An experiment has been conduced to test the system on a random sample of learners to test the effect of the intelligent part over the traditional part on learners learning. The t-test results were impressive and showed that students learning gain on average was 8.1% which is about 3 standard deviations higher when intelligent practiced were involved. The approach is promising and could be extended to cover other subjects and examinations such as SAT, GRE, LSAT, and FE, among others.

Table of contents

Chapter 1: Introduction 1
1.1 Motivation and General Background 1
1.2 Proposal and objectives 3
1.3 Thesis structure 5

Chapter 2 Intelligent Tutoring systems: A Review 6
2.1 Overview of ITSs 6
2.2 Historical Review 8
2.2.1 Educational software development from CAI to ITS 9
2.3 Types and classifications of ITSs 14
2.3.1 Learner Control versus Program Control 14
2.3.2 Domain Content versus Practice-based classification method 19
2.3.3 Abstraction of Learning Environment versus Knowledge Type of Instruction classification method 20
2.3.4 Scope and Functionality classification method: 22
2.4 Types of Web-based Tutoring 27
2.4.1 Static, Personalized and Adaptive Types of Web based Tutoring Systems 27
2.5 A survey of ITSs in the computer field 28
2.6 Conclusion 33

Chapter 3: System analysis and features 36
3.1 Methodology 36
3.1.1 Community needs survey 36
3.1.2 Learning and educational theories survey 37
3.1.3 Technical survey 44
3.2 Analysis of system components: 45
3.2.1 System Interface 46
3.2.2 System Components: 46
3.3 Conclusion 54

Chapter 4: Technologies and tools used to build the proposed system 56
4.1 Website Development tools 57
4.1.1 ASP .NET 2005 (Active Server Pages) 57
4.1.2 Visual Basic .NET 2005 59
4.1.3 Microsoft SQL Server 2005 59
4.1.4 XML (Extensible Markup Language) 60
4.2 Expert System Development tools 60
4.2.1 CLIPS - C Language Integrated Production System 60
4.3 Integration tools 62
4.3.1 Visual C++ .NET 2005 62
4.3.2 ZIP 2 Secure EXE 62
4.4 Conclusion 64

Chapter 5: The Proposed System Architecture 65
5.1 Overall System architecture 65
5.2 Creating the Web pages using ASP.NET 67
5.2.1: The Main Page (Default Page) 70
5.2.2 The Lessons 77
5.3 Designing and constructing the database 81
5.3.1 Identifying the relations 81
5.3.2 Building the tables 83
5.3.3 Relating tables ..... :84
5.3.4 Building Queries 88
5.4 The Reporting system 90
5.4.1. The Lessons visited report 91
5.4.2 Visited Assessments 92
5.4.3 The visited Post-Quizzes and the visited Pre-Quizzes reports 94
5.4.4 The Students Progress Report 96
5.4.5 The Quizzes Analysis Report 101
5.5 The evaluation tool 104
5.5.1 Assessment 107
5.5.2 Pre-Quizzes 107
5.5.3 Post-Quizzes 107
5.5.4 Constructing a quiz 108
5.5.5 Results Calculations 111
5.5.6 Displaying results 114
5.6 The communication tools 116
5.6.1 The Comments / Feedback Tool 117
5.6.2 The Questions Tool 118
5.7 Intelligent Practice based Instruction Tutoring System Development 119
5.7.1 The Intelligent Practices Architecture 120
5.7.2 The algorithm 126
5.7.3 Intelligent Practices Implementation 129
5.8 Other system components and features 136
5.8.1 Change Password 136
5.8.2 The Site Map 138
5.8.3 Adding Students 140
4.9 Conclusion 141

Chapter 6: System Evaluation 142
6.1 Technical testing 142
6.2 Website theory and content testing 144
6.2.1 Running the test: 144
6.2.2 Analysis 147
6.3 Results 152
6.3.1 Significance of data: 152
6.3.2 Calculating Averages and standard deviation: 155
6.4 Testing phase challenges 156
6.5 Conclusion 156

Chapter 7 Conclusion **158**

References: **165**

Appendix A: Educational Theories **179**
A1.1 Blooms Taxonomy 179
A1.2 Krathwohl Taxonomy 185
A1.3 Gagné's "*Conditions of Learning*" Theory 187

Appendix B: System tables and relations – Diagram 1 ..... **189**

Appendix B2: System tables and relations – Diagram 2 ..... **190**

Appendix C: Example of an intelligent practice/lesson in CLIPS **191**

Chapter 1: Introduction

This thesis examines the development of an online intelligent tutoring system (ITS) which is adaptive to students' levels. The system combines the benefits of the distance learning/Web-based tutoring systems and the benefits of ITSs/individualized instruction. Distance learning is directed to people who for some reason cannot attend classes in person. On the other hand, individualized instruction is designed for working with the students on a one-to-one basis, reproducing the instructor behavior in helping, directing and individualizing the instruction.

1.1 Motivation and General Background

Education of today is faced by many challenges and innovations. One of the most important aspects of innovation is instructional methodology. It has become very important that educators incorporate new technologies and methodologies into instructional methodology (Yousoof et al. 2006). On the other hand, computers have been employed to achieve a variety of educational goals since the early 1960. These have included automated testing, routine drills and practice tasks that used earlier technologies. Computer-based training (CBT) and computer-aided instruction (CAI) were developed as an attempt to teach using computers. In such kinds of systems, instruction was not individualized to the learner's needs. Instead, the decisions about how to move a student through the material were branch-like, such as "if question 15 is answered correctly, proceed to question 31 otherwise go to question 16."(Beck et al. 2004). These systems have successfully broken into all education and training fields: home, schools, universities, business and government but they remain far from the model educational experience (Corbett at al 1997).

Studies showed that no effective education can be made as efficient as a professional human tutor (Bloom 1984; vanLehn et. al. 2005). Therefore, in 1970 scientists adopted

the human tutor as their educational model and added artificial intelligent components and techniques to release the new intelligent model of computer-based instruction which has been referred to as Intelligent Tutoring systems (Cohen et al. 1982; Corbett et al. 1997). The essence of ITSs is to provide the one-to-one tutoring benefits. It provides individualized instruction, interaction with learners and customized constructive feedback. Such systems use artificial intelligence to design and implement multimedia learning besides other means of educational technologies. (Lane 2006; Ong and Ramachandran 2000). ITSs offer considerable flexibility in presentation of material and a greater ability to respond to individual student needs. These systems achieve their "intelligence" by representing pedagogical decisions about how to teach as well as information about the learner. This allows for greater adaptability by altering the system's interactions with the student. In addition, intelligent tutoring systems have been shown to be highly effective in increasing students' performance and motivation. For example, according to Shute, Glaser, and Raghaven (1989), students using Smithtown, an ITS for economics, performed just as well as students taking a traditional economics course, but needed only half as much time to cover the material.

To produce an effective ITS two important questions have been raised about how people learn and what makes tutoring the most effective way of learning. The two questions are related in the sense that if we know how people learn this will help us understand why tutoring is the most effective way of learning which in turn helps us to develop effective ITSs. To answer these questions many studies have been conducted and many findings were summarized in the National Research Council report (Bransford et al. 2000). For the first question the findings stressed many important points such as the importance of helping learners develop deeper conceptual learning and the importance of reflection or meta-cognition in learning. These points, besides some other studies, have emphasized the importance of proper feedback and interaction between learner and tutor suggesting that new developed ITS must implement ways of better interaction and constructive feedback customized to students needs and relative to their progress(Merrill et al. 1992; Chi et al. 2001).

Moreover, with the development of WWW technology, web-based ITSs are becoming an important area of research and development. The web-based ITS, installed and supported in one place, can be used by thousands of learners all over the world (Brusilovsky et al. 1996a). E-learning consists of replacing the physical location of the real educational institute for that of the computer. E-learning, web-based, online instruction or virtual education is widely used by educational institutes, schools, academies and universities to offer different types of online education. These concepts respond to distance learning where the physical gap between learners and educators is narrowed down by means of computing aids and technologies such as the Internet. This is by offering a greater two way of communication between learners and educators. At present, the majority of these E-learning systems do not offer individualized instruction to the students learning process (Contreras et al. 2006).

Combining distance learning with ITS brings new learning quality by merging learner guided and directed learning. It brings solutions to traditional distance learning programs where learners feel unguided.

1.2 Proposal and objectives

In this thesis, we propose an online, cognitive, and practice based instruction ITS system. To the best of our knowledge, it is a new approach to IT professional certifications tutoring systems especially for Module 1 in the International Computer Driving License[1] (ICDL) certificate. The goal of the proposed online system is to fill the gap between distance learning where guidance is missing, and traditional classrooms. It provides one-to-one individualized instruction automatically and cost-effectively while learners can still enjoy the benefit and convenience of distance learning. At the same time while all existing tutoring systems provides feedback to the student only, this system also provides

[1] For more information about ICDL, please visit their website at http://www.icdlgcc.com/

feedback to teachers and educators to enhance the system and teaching methodologies as required for the best results and outcomes.

This system serves two different users: the teacher and the student. The system also provides two main modes: the traditional learning mode, which facilitates the learning of concepts in a simple easy to navigate interface and the intelligent interactive mode that emulate teacher's decision making. The intelligent part allows the students to practice concepts (practice-based approach), assess their understanding, and get individualized instruction, feedback and recommendations using a rule-based expert system technology which is carried out using a cognitive pedagogical model. Besides that the system provides feedback to teachers through a detailed reporting system. Another important feature of the system is that it provides means of E-communication functionalities such as the "comments" facility where students and teachers can share and exchange ideas and opinions with other students and teachers besides the "ask questions" facility which enables students to view and ask questions while it enables teachers to view and answer them.

The system involves different assessment tools to assess students' progress including a pre- and post-quiz for each unit besides the assessment tool which is used to assess the overall student's performance at the end of the module. These tools can be used in various ways and serve different needs and purposes as will be demonstrated through out thesis. Another feature of the system is the reporting facility which can be used to analyze students' performance, progress and to update and improve the system. A student can track his/her progress with lessons, quizzes and assessments while the teacher can track all students progress in lessons, quizzes and assessments besides many other analysis reporting facilities. Teachers have the authority to update any of the quizzes or assessments stored in the question (item) bank. To achieve this, we build a comprehensive database where all required information is saved and used when needed.

The proposed system can be used either in a class for blended learning or alone online for distance learning. This system can be used for Module 1 - ICDL/ECDL test preparations

or as a computer basics training course material suitable for K12, college students, corporate training, etc.

Students may use the proposed system in different ways. It can be used far longer for students to learn missing knowledge and skills without coaching from a human instructor. Besides going through and reading the lessons, the system enables participants to assess their skills by doing practices and answering questions within an interactive learning environment. ITS goes beyond giving the right answer by giving each participant different sequence of questions depending on their level and by providing individualized guidance.

The proposed system is a prototype that is limited to multiple-choice type of testing. The reason is that this is the type of questions typically used in the real ICDL test for Module 1. This system can be utilized to be used with other international exam preparation that uses the same testing technique. For example it can be utilized to be used with SAT (Standardized test for college admissions in United States), MCAT (The Medical College Admission Test), GRE (The Graduate Record Examination), IB Diploma Program (The International Baccalaureate (IB) Diploma Programme), among others.

1.3 Thesis structure

In the following chapters the system is discussed in more details documenting the stages of the master project. In Chapter 2 more information about ITSs and different existing systems is provided, such as an overview of ITSs, different categories of ITS systems, and previous related work. In Chapter 3 more details about the proposed methodology are given and system analysis is explained. Following that in Chapter 4, the system architecture and design steps, including the algorithm used, is explained in details. In Chapter 5, details of the system development, what and how software and other tools were used will be listed and explained. Next, in Chapter 6 we will demonstrate the testing phase and evaluate the results. Finally, in Chapter 7 future research and development is suggested and the key points form this study are discussed.

Chapter 2

Intelligent Tutoring systems: A Review

Tutoring and learning are two important aspects in education. Tutoring helps learners to acquire knowledge while at the same time it enhances their learning skills (Li and Soh 2003). Many studies have been made in this area which resulted in the nontraditional learning. Nontraditional learning uses computers and is designed to support individual learning or organizational learning. This nontraditional learning can be summarized as e-learning such as Computer-Based Training (CBT), Computer-Aided Instruction (CAI), Intelligent Tutoring Systems (ITSs), and recently in Web-based education (Virvou, and Moundridou 2000). One important objective of these tutoring systems is to improve the efficiency of students learning.

This Chapter introduces the concept of ITSs, review their history and discuss their development in the context of Artificial Intelligence (AI) and educational theory.

2.1 Overview of ITSs

ITS is a broad term which could be defined as a set of computer-based instructional tools that relies on AI techniques to represent the pedagogical decisions and generate individualized interactions tailored to a student's learning needs (Conati 2004).

In general, ITSs consist of four basic components (Mandl, and Lesgold, 1988; Salgueiro et al 1998): expert knowledge model, learner (student) model, tutorial planning model, and tutorial planning model.

The *Expert knowledge Model* (aka *Domain knowledge Model*) consists of facts and ideas of a particular subject to be learned, usually from and with the help of a specialist or "expert" in the subject. The second model is the *Learner (Student)* Model which is about the individual characteristics of the student, and consists of the system's ability to continually, diagnose, and adapt the student developing knowledge and skills. Furthermore, the third model is the *Tutorial Planning* Model (aka *Tutor/Teacher Model*). This model works close to the learner model to guide the student by presenting appropriate learning activities, providing customized feedback, determining progress, and tailoring individual review topics. The fourth model is the *Communication Model* (aka *Interface Model*). It can be described as means of students' interaction through words or graphical interfaces, providing the customized instruction and evidence when using an ITS. This may also include the ability of the student and the instructor themselves to communicate with each other, through built-in E-mail-like, forums, blogs and other means of electronic capabilities which make online ITSs useful for web-based courses. In addition to the four basic models, the *Evaluation Model* has been introduced later in some studies (Salgueiro et al 1998). The Evaluation Model performs a general evaluation of the system and creates the necessary feedback to the teacher to help him modify teaching methods used, assessment questions, and clarity of instruction with the students during classes. Figure 2.1 illustrates the ITS components and interaction between them.

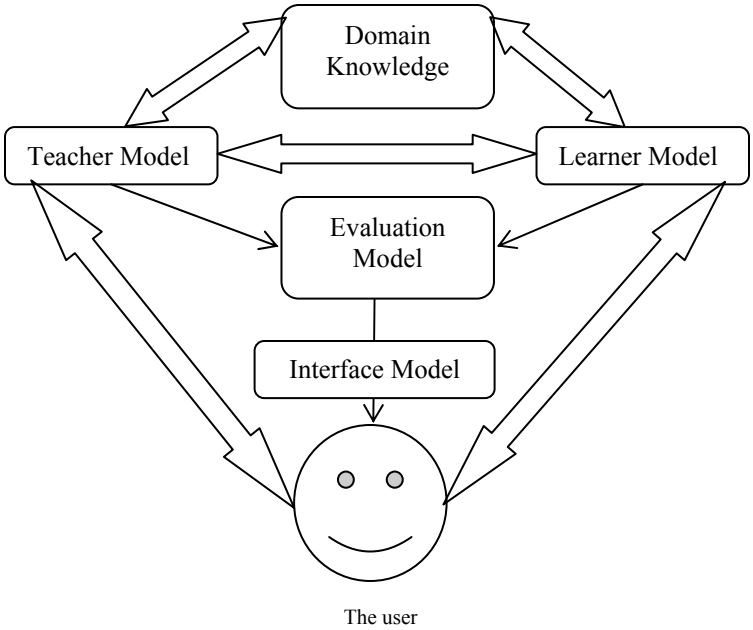


Figure 2.1 ITS's components

2.2 Historical Review

In 1960s, educators were under huge pressure to improve quality of education as well as to educate a wider sector of society (Steinberg 1977). This is how CAI systems first came to scene. In general CAI refers to a frame-based system with hard-coded links, such as hypertext with an instructional purpose (Freedman 2000). These applications included scheduling resources, managing teaching aids, and grading tests. Then, major applications quickly changed to using the computer to interact directly with the student rather than only using it to assist the human teacher. The first CAI programs were either computerized versions of textbooks or tools and practice monitors to present questions and compare answers to the pre-scored answers (Butz 1999). Both educators and computer scientists saw the importance to add some “intelligence” to these systems. This is in order for these systems to have richer presentations of their domain. Moreover, by modeling student knowledge, these systems could individualize their instruction and tailor the presentation to the level of student’s level of knowledge and rate of learning (Rickel 1989).

In the 1970s and 1980s, the concept of ITS has been pursued by researchers in education, psychology, and AI for more than three decades. ITS is a development of the earlier CAI model. Until very recently, scientists and developers in the AI community carried out the most of their work on ITS with minimum interaction with educational researchers. In general, recent ITS have the ability to teach a given subject, detect student errors, try to analyze where and how the student made an error, provide proper feedback to students, correct faults in the student′s logic and to clear up any doubts, confusion or misunderstandings the student may have about the material (Ong and Ramachandran 2000). Some ITSs provide practice-based instruction to support different levels of corporate training, i.e. K-12 and college education, and military training (Ong and Ramachandran 2000).

In the following sections, more details about the development of tutoring systems from CAI to ITS are reviewed and discussed. This is in addition to some examples and previously related work in the ITS field. Following that, different ITS types and classification methods of existing ITS systems are explained and examples for some current online ITS systems for computer subjects are listed.

2.2.1 Educational software development from CAI to ITS

CAI attempts to automate the programmed instruction (PI) pedagogical method. PI is the structured, goal oriented instruction technique. This technique was popular during the 1960s. With PIs, educators determine inputs, skills, outputs and behaviors learnt upon finishing the PI. Learners are led through lessons and errors are corrected immediately. CAI was a result of implementing PI on computers. (Thompson 1996 , Shute et al. 1994)

By the late 1960s and early 1970s, many researchers had moved beyond simply presenting problems to learners while collecting and tabulating their responses, to considering the student a factor in the overall instructional system (Suppes 1967). Many researchers developed systems that altered the presentation of new materials based on the history of a student's responses. While it was no longer sufficient to present information in a predetermined pattern to all students, system designers still had to foresee all

possible responses. The programmers had to identify in advance what types of student responses were possible and decide what information the system would then present. These systems were the first to "model" students, although they only modeled the students' behavior and did not make any attempt to model their knowledge states. Even though these systems were relatively simple, they were very effective. Students who used these systems improved on measures of the relevant skills and accurate recall. The implied learning theory assumed that students needed to first learn these basic skills and facts in order to prepare them for higher skills (Urban-Lurain 2002).

In the 1970s and 1980s CAI systems were developed into intelligent computer-aided instruction (ICAI) systems. ICAI systems attempt to evaluate why a student has missed a particular problem, then varies the curriculum depending on its evaluation. The step from CAI to ICAI was a gradual evolution from a basic computer-instruction method to a refined computer evaluation of student progress. At some point of refining ICAI and by applying AI principles, ITSs came to exist. Both ICAI and ITSs contain explicit knowledge of the subject taught. Besides that, ITSs also attempt to simulate the behavior of an intelligent human tutor in addition to acting as a domain expert. While there are numerous examples of ITSs employing different intelligent approaches, there are two term most often associated with them: cognitive diagnosis and adaptive modeling. Cognitive diagnosis entails forming an idea of the student cognition and knowledge usually in form of student modeling while adaptive remediation is the dynamic application of tutoring based on specific difficulties encountered by the student. Adaptive remediation usually is included in the instructor (tutor or pedagogical) model (Thompson 1996, Butz 1999).

The first ITS, SCHOLAR (Carbonell, 1970) attempted to engage the student in a mixed initiative dialogue on South American geography. The tutor program and student communicated through a sequence of natural language questions and answers. The tutor could both ask questions and answer them besides keeping track of the ongoing dialogue structure. This tutor was built around a semantic network model of domain knowledge.

Such models of conceptual knowledge were transforming common understanding of question answering and deduced reasoning in cognitive science (Anderson and Bower 1973; Collins and Loftus 1975).

Recent ITS models are based on early models such as BUGGY, and others including EXCHECK, SOPHIE and GUIDON. These models are discussed below.

BUGGY was developed in 1975, (Brown and Burton 1978; Brown et al 1977; and Brown et al 1975). It began with the study of procedures produced by students involved in algebraic operation such as the subtraction of multi-digit numbers.

Developers of BUGGY focused on developing a complete diagnostic model of the student rather than providing BUGGY with the tutorial skills needed to recognize and run counter to student mistakes. They attempted to enumerate the different possible procedural bugs (mistakes) students might get while trying to solve math problems. BUGGY used its student model to simulate a student with "buggy" thinking which is a reversal of the usual roles of ITS, teacher and student. The teacher then diagnoses the student bug based on BUGGY's examples. BUGGY could generate general diagnostic tests to identify students' mistakes by using its catalog of possible bugs such as "the student subtracts the smaller digit in each column from the larger digit regardless of which is on top." Building the enumeration of bugs required considerable effort thus somehow restricting the use of BUGGY to well defined procedural domains. The use of a fixed library of bugs also restricted the power of the system to already known errors (Antonella 1998). Burton elaborated on this model with the DEBUGGY system (Burton, 1982). DEBUGGY identified 130 "bugs" designed to account for mistakes in subtraction. The challenge was to analyze the problem space represented by the student's answers and determine which bug or set of bugs best accounted for incorrect subtraction (Urban-Lurain 2002.

One of the very important early results in intelligent tutoring research was the recognition of the importance of the cognitive reliability of domain knowledge module (Corbett et al 1997). It is important for the tutor to reason about the problem in the same way that

humans do. Such cognitively valid models can be contrasted with “black box” and “glass box” models (Anderson 1988). A black box model is one that fundamentally describes problem states differently than to the student. The classic example of such a system is SOPHIE I (Brown et al 1973). It was a tutor for electronic troubleshooting that used its expert system to evaluate the measurements students were making in troubleshooting a circuit. The expert system did not apply causal human reasoning, but based its decisions by solving sets of equations. As a consequence the tutor could recommend optimal actions in each problem solving context, but it was entirely up to the student to construct a description of that problem solving context and the rationale for the appropriate action.

A glass box model (Corbett et al 1997) is an intermediate model that reasons in terms of the same domain constructs as a human experts, but with a different control structure. The classic example of such a system is Clancey’s (1982) GUIDON system that tutored medical diagnosis. This system was constructed around the expert diagnostic system MYCIN. MYCIN consists of several hundred if-then rules that probabilistically relate disease states to diagnoses. These rules reference the same symptoms and states doctors employ in reasoning, but with a radically different control structure. MYCIN reaches a diagnosis by an exhaustive backward reasoning mechanism. Although the expert system and doctor would agree on the description of any problem solving state, Clancey pointed out that it is difficult to tutor a doctor on what to do next when the expert system and student are following a different trajectory into and out of the state.

In 1982, Sleeman and Brown reviewed the state of the art in CAI and first invented the term *Intelligent Tutoring Systems* (ITS) to describe these evolving systems and distinguish them from the previous CAI systems. The implicit assumption about the learner now focused on *learning-by-doing*. They classified the existing ITS as being computer-based problem-solving monitors, coaches, laboratory instructors and consultants (Sleeman & Brown, 1982)

The prominence of these systems was still at research stages and proposals for refining AI theories, but now researchers were thinking about representing student knowledge within these systems. At this point, the first use of the term student model to describe an abstract representation of the learner within the computer program has arisen. They

classified the student model into three different models (Urban-Lurain 2002): Overlay, differential, and perturbation models. The *Overlay model* is a model of student knowledge as a subset of an expert's knowledge. The D*ifferential model* is similar to the overlay model, but focuses on the differences between the student's knowledge and that of the expert. Another model is the Perturbation model which represents student misconceptions and misunderstandings as variants of the procedural structure of the experts' correct skill. These are sometimes known as *mal-rules* (bad rules) that describe student misconceptions in relationship to the experts' knowledge.

The student model addressed in this thesis is closely related to the Perturbation model. Our proposed ITS asks questions and finds out students' misconceptions in a procedure structure of the expert system to correct mistakes and misunderstandings. Specific mal rules were created to recognize the misconception and guide students, and give them instructions and recommendations.

Another early ITS model is EXCHECK. According to Etienne Wenger (1987) EXCHECK is a proof checker that uses natural inference methods. It has been proven to be useful in the classroom as the core of an undergraduate course at Stanford University for years. It creates an intelligent interface with a powerful model of domain expertise to provide a learning environment where the student can get feedback during problem solving. It imitates human proof techniques with macro-operators that make use of theorem verification. It communicates with the student via a formal language of abbreviations. It provides a friendly environment, gives intelligent feedback and verifies students' work in terms they understand. It lacks most of the features of AI in that it does not form a global model of the student and it does not use pedagogical strategies to make its interventions contextually relevant and effective. (McDonald, 1981; Blaine, 1981)

ITS implementation has not advanced significantly since the 1980s. During the last decade, software has developed, improved and expanded including interactive multimedia development tools making a good environment for ITSs (Butz 1999).

2.3 Types and classifications of ITSs

There are several ways of categorizing ITSs; in this section we will show four different categorization methods.

2.3.1 Learner Control versus Program Control

Learner control is a feature of tutoring systems where learners can direct the tutoring sequence and flow and thus guiding the system to respond to their needs and interests (Steinberg 1977, Duchastel 1986, Clark and Mayer 2002, Li and Soh 2003). Program control, also is called tutor or system control, let the software make the learning decision for the learner such as controlling the flow and the sequence of instruction. In the program control strategy, learners are led through the instruction in a lock-step manner and this strategy is associated with the traditional way of teaching and learning where the tutor controls the overall learning environment. In the following we discuss the learner control strategy in more depth.

2.3.1.1 Learner control

Previous ITSs were focusing on building a smart tutor that can communicate knowledge to individuals. Recent and emerging work is focusing on the learning control over the learning process such as using adaptive systems (Kay 2001). In 1961, the term "leaner control" was officially created and used in education by Mager (Mager 1961, Niemiec et al. 1996). The concept of learner control is comparative to the tutor control and the system control. Our system in this paper falls in the learner Control category. The system is said to be a learner control system when it gives the option to the learner to select the topics they want, control the sequence and the pace at which they progress, decide whether and how to use the tools provided by the system(Li and Soh 2003). This type of systems symbolizes both the subject and objective of tutoring and learning. Such systems enable learners to skip over material they studied before or pass up material they are not

ready for. Accordingly, learner control systems are appealing and they avoid frustration and anxiety (Steinberg 1977, Steinberg 1989).

In the following, learner control systems and options are discussed. Though, in this paper, a new approach of learner control is introduced by providing the lessons and the practices which gives learners the opportunity to assess their knowledge level before going to lessons and/or gives them feedback and recommendations that specifically ask them to read specific parts to cover some identified knowledge gaps.

Learner control strategies are built on and specific to the human learning mechanism as in the following.

2.3.1.2 The Human Learning Mechanism and the role of cognitive science

During the 1980s, computer scientists specializing in AI continued to focus on the problems of natural language, student models, and deduction. However, the field also attracted researchers from outside the computer science discipline, most particularly John Anderson. He was working in cognitive science, developing the Adaptive Control of Thought (ACT*) theory of cognition (Anderson 1983), see Figure 2.2. The theory focuses on memory processes. It is an elaboration of the original ACT theory (Anderson, 1976) and builds upon HAM, a model of semantic memory proposed by Anderson & Bower (1973). Anderson provides a complete description of ACT*, besides his own critique of ACT* and the outline for a broader development of the theory (Anderson 1983, 1990, 1993). Anderson's name has become well known with ITS work as "Anderson-style tutors" (Chipman, 1993). Perhaps this is because his systems have actually been used in classroom settings and were not only for research purposes rather than practices. Two of the best examples of ACT* are the Geometry Tutor (Koedinger and Anderson, 1993) and LISPITS “LISP Intelligent Tutoring System” (Ong and Ramachandran, 2000).

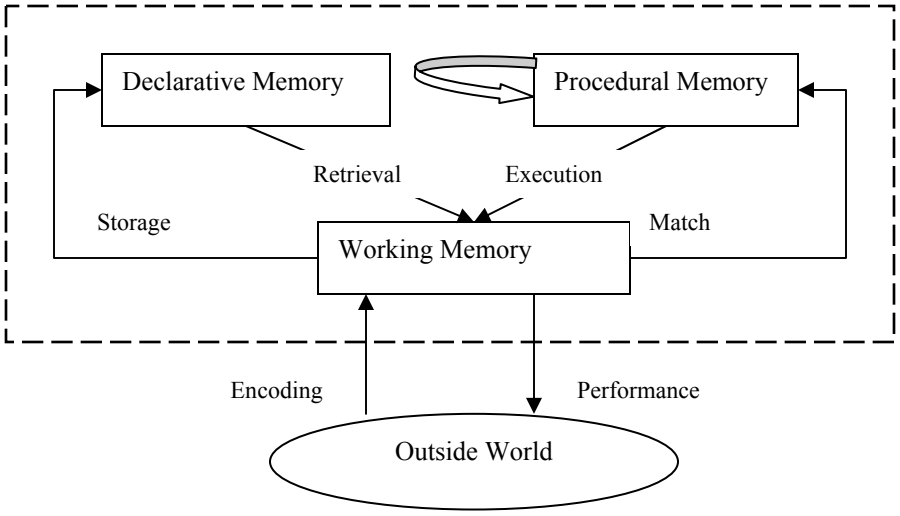


Figure 2.2: ACT* theory of cognition (adapted from Anderson 1993)

As shown in Figure 2.2, ACT* distinguishes between three types of memory structures: declarative, procedural and working memory (Anderson, 1993). *Declarative memory* takes the form of a semantic net linking propositions, images, and sequences by associations. While *Procedural (production) memory* and also called long-term memory, represents information in the form of production rules; each production has a set of conditions and actions based in declarative memory. The nodes of long-term memory all have some degree of activation. The *Working memory* is that part of long-term memory that is most highly activated.

According to ACT*, all knowledge begins as declarative information. Procedural knowledge is learned by making inferences from already existing factual knowledge. Moreover, ACT* supports three fundamental types of learning: generalization, discrimination, and strengthening. The first type is *Generalization*, in which productions become broader in their range of application. The second type is *Discrimination*, in which productions become narrow in their range of application. Moreover, the third type is *Strengthening*, in which some productions are applied more often. New productions are formed by the conjunction or disjunction of existing productions.

In 1993 Clark has explained cognitive science as the following: "Cognitive science seeks to understand how the mind emerges from the brain. Artificial intelligence seeks to

implant a mind within the machine. To their detriment, both have largely ignored work on the natural history of the mind." (Clark, 1993).

Cognitive Tutoring Systems are a subtype of ITS which incorporate domain-specific production rules which are based on a cognitive theory of skill acquisition, such as ACT-R. (Anderson. 1993). Intermediate cognitive steps are first identified using empirical methods such as cognitive task analysis (Lovett 1998; Koedinger and Anderson 1993; Gertner and Andes 2000). Most cognitive tutors can be used for procedural lessons, for example Science and Math lessons, flight simulator training, (Sarter and Amalberti, 2000) and training in the workplace (Lesgold et al 1993).

2.3.1.3 Learner control options

In general there are four options regarding learner control: the first one is content and content sequencing. This option gives learners the flexibility to select instructional goals and contents and control the display order of the courses, topics and screens within a lesson. Secondly, the pacing option where learners can control the time spent on each lesson. Third is the access to learning support. This option enables learners to control when to access which instruction components of lessons such as lessons, quizzes and practices. Finally, is the Instructional interaction method. This option gives the learner the opportunity to decide whether to learn individually or collaboratively (Clark and Mayer 2002, Li and Soh 2003). The following systems show the use of learner control strategies in ITS.

TICCIT (Time-shared Interactive Computer Controlled Information Technology Television) was designed and developed Merril and his colleagues to teach “higher order “concepts. It uses learner controlled content selection, self pacing, sequencing and other instructional features. The system allows learners to access and change difficulty levels, number of examples and practice items and the nature of advice. Another example for learner control systems is MAIS (Minnesota Adaptive Instructional System). It was developed by Tennyson who used Bayesian statistical method to adjust the systems instructional support on selecting the appropriate amount and sequence of instruction for learners (Tennyson 1980). Moreover, the SCHOLAR system, as previously explained, is

the first ITS and is an example of the learner control systems where students are involved in a mixed-initiative dialogue in which eother students or the sytem can ask questions and expect response. The learner can at any time interrupt the dialog and reorient the interaction (Duchastel 1986, Carbonell 1970). In addition is another system that is named WEST. WEST is a game environment that enables learners to progress through the game according to the learner strategy. The system is thus responsive to the learner initiated moves.

2.3.1.4 Types of Learner Control Strategies

In general, learner control strategies can be classified in to 6 types while these types are not absolute and there are some overlaps (Li and Soh 2003). The first strategy to discuss is the selection of instructional goal and content. This strategy gives the learner the opportunity to select the learning topics, content, goals and difficulty level. The second strategy is the sequencing and pacing of the instructional material and units. This type gives the learner the flexibility with content sequencing, speed and self pacing. The third type is the selection and control of instructional components as example the learner can control the presentation of the instructional components step by step as in the rule example practice. The learner can set the number of examples and practices besides other instructional tools. Meta-cognitive control is the fourth type to discuss. The learner is given the chance for self- regulated learning and control strategic planning such as teaching strategies and reviewing the past errors. The fifth type is the learner construction type. The learner in this type negotiates the instruction goals and objectives with the tutor and can do self-evaluation and assessment. The last type is the selection and control of instructional interaction method. In this category the learner decides to work alone or collaboratively and decides the group, select partners (via Emails, chats or forums) and select the coordination and communication tools, mixed initiatives dialogue between learners and tutors or between learning partners (Li and Soh 2003).

2.3.2 Domain Content versus Practice-based classification method

One way to organize tutoring systems is around what role they are anticipated to play. At one end of the scale, some systems are proposed to replace a textbook or classroom instruction to deliver domain content to a student for the first time e.g. intelligent hypermedia systems (Lane 2006).

At the other end are systems designed to directly support practice. These usually complement an existing instructional component such as lectures, books, etc. ITS systems tend to lean on the practice end. Practice is when "the rubber hits the road" in learning. It is when knowledge gaps are revealed and skills are automated. Modern theories of learning stress the critical role of practice and highlight the importance of feedback because of the risks of unguided learning. (Kirschner et al., 2006; Lane 2006; Clark, 2004)

Tutoring systems usually supports practice in two ways. The first is called Product tutors which evaluate final outcomes. Feedback is given after the student completes the work and submits it. The ITS analyzes the system and gives the feedback using some engineering solutions such as plan recognition to identify the reasoning underlies the solution. A drawback of this method is that students may get stuck before they are able to generate a solution.

The second way is by means of Process tutors**.** These tutors are more interactive and pro-active. This type of tutors provides support while student is working. This is the most familiar category because human tutors work in this mode. The student is observed step by step, feedback, hints and recommendations are given, questions asked and answered, etc. (Kirschner et al. 2006; Lane 2006; Clark, 2004)

One example of practice based process tutors is SHERLOCK. SHERLOCK is a practice environment for electronics troubleshooting commissioned by the American Air Force (Lesgold, et al. 1992; Katz and Lesgold, 1993). SHERLOCK represents a diagnosis stream, (Brownet al. 1973; Clancey 1982; Gitomer et al. 1995; and Tenney and Kurland,

1988). SHERLOCK is designed to give technicians practice in troubleshooting the F-IS Avionics Test Station.

2.3.3 Abstraction of Learning Environment versus Knowledge Type of Instruction classification method

In the following method of categorization we will concentrate on two dimensions: abstraction of the learning environment and the knowledge type of the instruction

Abstraction of the learning environment

Many systems attempt to provide instruction by simulating a realistic working environment in which the student can learn the task. There are many reasons for developing such systems, including the possible danger of training using the actual equipment and the lack of domain experts who can devote their expensive time to training novices. Therefore, a realistic simulated learning environment can reduce both the cost and the risks of training. An example of a simulation-based ITS is the Advanced Cardiac Life Support (ACLS) Tutor (Eliot and Woolf 1995). Another example of a system which may take less rigorous approach is Smithtown (Shute et al 1989). It takes this approach by providing a simulated setting for students to test hypotheses about economics. Another example of such a system is the Design for Manufacturing Tutor (Haugsjaa, and Woolf. 1996).

Emphasis of Instruction

There is a long history of classifying instructional goals according to the type of knowledge being taught. An important early attempt at this classification is Bloom's taxonomy (Bloom[2] 1956) and much recent work in categorizing knowledge has been derived from this. In addition to classifying learning goals by knowledge type, one can also examine what the student will be able to do upon completion of the ITS lesson. This

[2] Please refer to Appendix A for more details about Bloom taxonomy/ theory and other learning theories.

can vary from the student being able to perform a set of skills in a manner similar to an expert to understanding abstract concepts such as Newton's third law (Beck et al. 2004). According to this classification, there are two main types of tutor: cognitive tutors and knowledge based tutors.

Cognitive tutors

Modeling students' cognitive skills is one of the most important research issues for Cognitive Tutors (Greer and McCalla, 1994). As discussed earlier in this chapter, Cognitive Tutors are intelligent tutoring systems based on a rule-based (expert systems) cognitive model of the subject domain being taught to students (Anderson et al. 1990). It develops a cognitive model of a student as they interact with the program, providing problems and individualized instruction based on this model. Based on the ACT theory (Anderson et al. 1995), this cognitive model is written as a system of if-then production rules that are capable of generating the multitude of solution steps and misconception-steps typical of students (Koedinger and Anderson 1997). Cognitive Tutors are effective in increasing student learning rate over other alternative teaching methods (Anderson et al. 1995).

According to Matsuda, Cohen and Koedinger (2005), the cognitive model is the basis for two student modeling techniques: model tracing and knowledge tracing. *Model tracing* is used to monitor student's progress through a problem solution (Anderson et al 1990). This tracing is done in the background by matching student actions to those the model might generate. The tutor is mostly silent. However, when help is needed, the tutor knows where the student is and can provide hints that are individualized to the student's particular approach to the problem. *Knowledge tracing* is used to monitor students' learning from problem to problem (Corbett & Anderson, 1992).

An example of a Model tracing cognitive tutor is SHERLOCK (Lajoie and Lesgold. 1992). Another example of an ITS that uses an analysis of expert behavior is the LISP tutor (Anderson and Reiser1985).

Knowledge based tutors

These kind of tutors concentrate on teaching concepts and "mental models" to students. Such systems encounter two main difficulties. The first difficulty is that more substantial domain knowledge is needed for instruction. The second difficultly is that since learning concepts and frameworks is less well understood than learning procedures, there is less cognitive theory to guide knowledge representation and the pedagogical module. For these reasons, ITSs of this type require a larger domain knowledge base and are referred to as knowledge based tutors. As a result of not having a strong model of skill acquisition or expert performance, these systems are forced to use general teaching strategies. They also place more emphasis on the communication and presentation system in order to achieve learning gains. An example of such a system is the Pedagogical Explanation Generation (PEG) system (Suthers 1992) which uses a substantial domain knowledge base to construct answers to student queries in the domain of electrical circuits.

Emphasis on instruction is not a definitive method of classifying intelligent tutors. An example of a system that does not fall into either of these categories is Coach (Winkels et al 1990). It teaches how to use UNIX mail. This is a procedural skill, and hence cognitive in nature. However, the emphasis of this system is also knowledge based and involves generating explanations and using general pedagogical tactics for generating feedback.

2.3.4 Scope and Functionality classification method:

According to Murray in 2003 there is another method of categorizing ITSs into seven known categories depending on scope and functionality. Categories are Curriculum Sequencing and Planning, Tutoring Strategies, Device Simulation and Equipment Training, Domain Expert System, Multiple Knowledge Types, Special Purpose and Intelligent/Adaptive Hypermedia. As explained below.

The Curriculum Sequencing and Planning category is the first category in this classification method. In this category systems organize instructional units into a hierarchy of prerequisites related courses, modules, lessons, presentations, and so on. These instructional units usually have instructional objectives. Some systems in this

category include instructional units that address misunderstandings, misconceptions or corrective and remedial material. The second category is the Tutoring Strategies category. Systems in this category surpass at representing various teaching strategies. The third category is the Device Simulation And Equipment Training category. With such systems in this category the student is usually shown a piece of equipment and is asked to identify its components, do some operating steps, perform some maintenance steps, and/or identify defective device behavior and repair or restore the concerned parts. The Domain Expert System category (aka Expert Systems and Cognitive Tutors) represents an important class of intelligent tutors. These systems include rule-based cognitive models of domain expertise. Such kind of tutors (aka model tracing tutors), observe student behavior and build a cognitive model of the student's knowledge that can be compared with the expert model. In the Multiple Knowledge Types category, the instructional design theories classify knowledge and tasks into diverse categories, and recommend instructional methods for each category. Systems in this category have a tendency to be restricted to types of knowledge that can be easily defined, for example: facts, concepts, and procedures. The sixth category in this classification method is the Special Purpose Category. |Systems in this category focus on more specific, less general tasks. These systems specialize in particular tasks or domains. The last category in this system is the Intelligent/Adaptive Hypermedia. Systems in this category are designed to manage the hyperlinks between parts and units of content as well as managing the form and sequencing of the content itself. The links available to the student can be intelligently filtered, sorted, and explained based on a student model or profile for example, it could be based on prerequisites, cognitive load, subject suitability, complexity, and so on (Anderson and Pelletier 1991, Brusilovsky 1998, Murray 1999)

These categories are not strict and many ITS may fall in more than one of them.

Table 2.1 discusses the categories of each ITS clarifying each category's strengths, weaknesses along with a summary of some features/points that might make systems differ within the category (Murray 2003).

No.	Category	Strengths	Limitations	Variations	Projects/systems
1	Curriculum Sequencing and Planning	- Rules, constraints, or strategies for sequencing courses, modules, and presentations - Separates content from presentation and sequencing	- Low fidelity Interfaces from student's perspective - Low skill representation - Canned content - Simple students model	- Whether sequencing rules are fixed or could be authored - Scaffolding of the authoring process	Swift/DOCENT, IDE, ISD Expert, Expert CML
2	Tutoring Strategies	- Micro-level and explicit tutoring strategies - Sophisticated set of instructional primitives - Multiple tutoring strategies	- Low fidelity interfaces from student's perspective - Low skill representation - Canned content - Simple student model	- Strategy - Representation method - Source of instructional expertise	REDEEM (& COCA) ,Eon, GTE, Smart Trainer AT
3*	Device Simulation and Equipment Training	- Micro-world/simulation-based learning environments - Authoring and tutoring matched to device component identification, operation, and	- Limited instructional strategies - Limited student modeling - Mostly for procedural skills	- Reliability of the simulation - Simplicity of authoring	RIDES, DIAG, MITT-Writer, ICAT, SIMQUEST, XAIDA, Instructional Simulator

		troubleshooting			
4*	Domain Expert System	- Executable/deep models of domain expertise - Detailed (fine chunked) student diagnosis and modeling - Buggy and novice rules included	- Building the expert system is difficult - Limited to procedural and problem solving expertise - Limited instructional strategies	- Cognitive vs. performance models of expertise	Cognitive Tutor Authoring Tools (CTAT), ASPIRE, Demonstr8, D3 Trainer, SimStudent DIAG, Training Express
5	Multiple Knowledge Types	- Clear representation - Template like framework for decomposing content into facts, concepts and procedures. - Many are based on instructional design theory principle facts, concepts and procedures	- Limited to relatively simple fact, concepts, and procedures - Pre-defined tutoring strategies	- Inclusion of intelligent curriculum sequencing - Types of knowledge/tasks supported	CREAM-Tools, DNA, ID-Expert, IRIS, XAIDA, AuthorIT, Instructional Simulator & IDVisualizer
	Special Purpose	- Template-based systems provide strong authoring	- Each tool limited to a specific type of tutor	- Degree of inflexibility	IDLE-Tool/IMap/Indie,

6*		guidance - particular design and pedagogical principles can be enforced	- Inflexibility of representation and pedagogy		Leap-AT, BioWorld Case Builder, LAT, WEAR
7	Intelligent/Adaptive Hypermedia	- Similar to category one but also deals with Navigation and (dis)orientation issues. -WWW has accessibility & UI uniformity - Adaptive selection and annotation of hyperlink- - Potential for making inferences from large numbers of students	- Limited interactivity and learning environment - Limited student model bandwidth	- Macro vs. micro level focus - Degree of interactivity	ActiveMath, InterBook, MetaLinks, CALAT, GETMAS, TANGOW, ECSAIWeb

Table 2.1: ITSs categories (adapted from Murray 1999 & 2003)

2.4 Types of Web-based Tutoring

In this chapter, we explain the different types of available Web-based ITSs. This is by discussing the different web-based tutoring systems, and showing the differences and similarities between them combined with some given examples.

2.4.1 Static, Personalized and Adaptive Types of Web based Tutoring Systems

From the very early days of the World Wide Web (WWW) there were several efforts to use WWW facilities for distance learning. At the same time most existing educational Web-based applications use the simplest solutions and are generally weaker and more restricted than existing 'on-site' educational systems and tools (Frasson et al. 1992). ITSs use the Internet as a constructivist learning environment to provide intelligent assistance and improve both quality of training and distribution of knowledge in a distance learning situation. Web-based ITS systems are available to learners without the restriction of time and location. These kinds of systems provides learning opportunities to all people who cannot attend class for some reason or those who would like to learn at their own pace and convenience. It does not have end-users installation cost and provides learners the maximum time/place flexibility besides guidance and direction.

The most obvious way to bring ITS to WWW is to port some existing ITS technologies to WWW. The most appropriate technique for that is intelligent knowledge sequencing and problem sequencing (Brusilovsky, 1992). Intelligent knowledge sequencing implies adaptive selection of the next topic to be learned using the student model and the knowledge about the learning material. Intelligent problem sequencing in the narrow sense implies choosing or generating the next problem for the student to solve, which should be most relevant to his level of knowledge and skills. In a broader sense it implies adaptive sequencing of all kinds of teaching operations - examples, tests, problems, questions, (Brusilovsky 1995).

The following are the main kinds of available Web-Based Tutoring (WBT) methodologies (Capuano et al 2000): static, personalized, and adaptive. The *Static WBT* is simply web based training provided by preparing learning material that covers one or more topics and converts them into interactive linked HTML pages or any type of Web-deliverable object. Material is then sited on-line to make it available to users. Learners can make use of it only by following the path established by teachers. Personalized WBT is prepared by using some specialized authoring tools to perform manually a set of additional tasks. They can monitor student knowledge by assessing them, and define different paths through learning objects for different kind of learning goals. Adaptive WBT includes all features of a Personalized WBT but uses AI techniques to reproduce the teacher decision making and support. Example of adaptive ITS is ELM-ART (ELM Adaptive Remote Tutor) which is a WWW based ITS to support learning programming in Lisp. ELM-ART is based on ELM-PE (Weber & Möllenberg 1994), an Intelligent Learning Environment that supports example-based programming, intelligent analysis of problem solutions, and advanced testing and debugging facilities. For several years, ELM-PE was used in introductory Lisp courses at the University of Trier. The course materials were presented to students in lecture form and ELM-PE was used to practice the obtained knowledge by problem-solving. ELM-ART, which is projected to be used for distance learning, provides on-line both the course materials and the problem solving support (Brusilovsky et al 1996).

2.5 A survey of ITSs in the computer field

ITSs have been used to teach various types of scientific topics and professional skills. Examples may includes, different languages, mathematics, physics, chemistry, algebra, computer subjects, maps, radar skills and air force jet aircraft and navigation tools, flight simulators, various military training and medical education such as how to treat heart attack patients (Ong J. and Ramachandran S. (2000), Rane A. (2005), Henke S. (2006)). According to Brusilovsky, P. et al. (1996), most of the intelligent techniques applied in

ITS can be classified into the following three technologies: curriculum sequencing, interactive problem solving support, and intelligent analysis of student solutions. These technologies are meant to support the "intelligent" tasks that the human teacher offers. Curriculum sequencing and intelligent analysis of student solutions are the oldest and best-studied technologies in the domain of ITS. Most of the ITSs developed during the first 10 years of ITS history belong to these technologies. The technology of interactive problem solving support is a newer one, but it is more "intelligent" and supportive. It helps students in the most difficult part of the learning process. Nowadays, it became a dominating technology. Furthermore, making ITS available on the World Wide Web (WWW) is a way to incorporate the flexibility and intelligence of ITS with world-wide availability and easy accessibility of Web applications (Brusilovsky, P. et al. 1996). In this section, we discuss some related ITSs work on computer science subjects.

ELM-ART (Brusilovsky, Pet al. 1996) is a WWW-based ITS that supports a learning of LISP programming. It is considered as an on-line intelligent textbook with an integrated problem solving environment. The system provides all the course materials such as presentations of new concepts, test, examples, and problems in hypermedia form. ELM-ART differs from simple WWW "hyperbooks" in that it "knows" the material it presents to the students and supports them in learning and navigating the course material. Besides, all examples and problems, which are important components of any textbook, are "live experience" in ELM-ART and not just a text as in other textbooks. The student can investigate all examples and solve all problems on-line. To support the student navigating through the course, the system uses two adaptive hypermedia techniques: *adaptive annotation* and *adaptive sorting of links*. The former technique means that the system uses visual cues such as icons, fonts and colors to show the type and the educational state of each link. The later technique is used to present similarity links between cases. Also, as ELM-ART is a system that supports example-based programming, it encourages the students to re-use the code of previously analyzed examples when solving a new problem.

SQL-Tutor (Mitrovic, 1997) is a problem-solving ITS based on learning-by-doing for the Structured Query Language (SQL) of databases. The system uses constraint-based modeling to form models of its students. The student can learn about concepts and elements of SQL. It supports the acquisition of domain knowledge in a declarative form such as constraints and strengthening of this knowledge in the course of practice. The system provides assistance in problem solving and advice against mistaken actions. It helps learning by supporting details and clarifications on the basis of error messages and by providing the correct solutions to the student. Weaknesses of this system are that it is neither web-enabled nor connected to real Relational Database Management System (RDBMS) in such as way that it helps in providing a real learning environment. Consequently, students cannot view the result of their query, after they successfully finishing it (Bhagat S. & L et al. 2002, Rane A 2005).

ACHARYA (Bhaghat L. et.al, 2002) is an ITS that focuses also on teaching SQL. It is an adaptive instruction sequencing system based on student model. Content is broken up into units, with each unit related to a specific concept that may or may not have pre-requisite concepts. A concept should be recommended for study only if the student has completed all the pre-requisite concepts. A set of problems are presented to the student based on concepts that the student is expected to know from that concept and any pre-requisite concepts. The primary components of ACHARYA are (Bhagat1 S.& L et al. 2002): Traditional course material and intelligent problem solving environment. The former provides material on SQL organized to develop ITS technology. The later provides a problem solving environment where students try to solve problems posed by the system and get qualitative feedback on their solution to the problem posed by the system. Furthermore, in ACHARYA, the system focuses on hands-on practice exercises as the basic learning methodology. It follows the constructive model for delivering the instruction (Rane A 2005).

ADIS (The Animated Structure Intelligent Tutoring system) (Warendorf K.1997) is a constraint-based modeling ITS developed as a teaching aid for a course on Data Structures to enhance students understanding of this subject such as linked lists, stacks,

queues, trees and graphs. ADIS is entirely implemented in Java to allow platform-independent standalone usage or internet delivery. ADIS has the potential to show data structures graphically on the computer monitor in addition to allowing graphical management of the data structure created. It consists of a tutorial mode that integrates exercises, where students can learn fundamental algorithms such as insertion and deletion of data structures visually (Warendorf K., Tan C., 1997, Rane A 2005, Ohlsson 1992).

BITS (Butz C. et al.2006) is a Bayesian Web-based ITS that supports and guides students learning process in C++ programming. The decision making process performed in BITS intelligent system is guided by a Bayesian network (BN). BNs are a formal framework for uncertainty management in Artificial Intelligence based on the probability theory. The system can help a student navigate through the online course materials, recommend learning goals, and generate an appropriate reading sequence. The four main modules of BITS are (Butz C.J et al. 2004): BNs, the knowledge base, the user interface, and the study module. The study or tutorial module guides the student through the class concepts using three sub-modules called: Regular Study, Problem Study, and Quick Study. In general, the regular study module integrates exercises, where students can learn fundamental algorithms visually. The system keeps track of the student's knowledge of a concept and accordingly, BNs is updated and the navigation menu is shown while additional help is provided, if necessary, and a sample quiz on the concept may be retrieved and shown to the student with immediate feedback. While in the problem study module, if the student indicates that the concept is not understood, then BITS displays the prerequisite concepts. Here BITS confirms that each of the prerequisite concepts is understood. Only then, BITS goes back to the problem concept. If a student wants to go directly to a specific concept, the quick study module facilitates this procedure and then displays the ancestor topic of the chosen concept. (Butz C.J. et al.. 2004).

JV^2M is a game-driven ITS to teach the compilation process of Java programs. JV^2M was developed by Gomez-Martin et al. (2004). The system aims is to relate and put into practice the game idea combined with ITS. This system teaches programmers with Java knowledge, the internal mechanism of the Java Virtual Machine, and how the compiler

translates source code into object code (JVM code) through game-like simulations. It uses the Case Based Teaching (CBT) and pedagogical agents approach. JV^2M provides a background story and a virtual, three dimensions environment, world wherein the student can manoeuvre objects and interact with agents and other characters. The user also owns an inventory where collected objects are stored for later use. The student has to solve exercises which include concepts at different levels. A pedagogical agent called JAVY resides in the learning environment and provides the user with help. According to Gomez-Martin et al. (2004) JV^2M was under development and it needed more work besides that system evaluation was planned to be conducted to assess the system effectiveness (Fasli M. and Michalakopoulos M. 2005; Gomez-Martin et al. 2004).

DB-TUTOR by Bouhadada T. and Laskri M.T. (2001) is based on the Double Test Learning (DTL) strategy. It uses two companions as co-students where one of both is a troublemaker. DB-TUTOR is designed for teaching the databases for students at the third year of the graduate cycle. The idea of the DTL strategy is based on that the student will benefit from the co-students' mistakes and behavior. It starts with a training phase, then Post-Test1 phase where the teacher tests the student. The student will follow the interaction between the tutor and the co-students and he can write observations about the behavior of the co-students in a notebook that is created for that purpose. The interaction consists of a sequence of question/answer that is evaluated by the tutor. In the third phase called post-test2, the tutor will evaluate the human student, which will react to the interaction without his notebook. The system proposes a student model taking into account the DTL strategy for the evaluation phase. The prototype of the DB-TUTOR system was developed under DELPHI 5 environments.

Our system in this paper is a web-based ITS which teaches computer concepts to students and prepares them for the ICDL/ECDL module 1 professional exam. The system provides the course material, pre and post quizzes for each chapter, an assessment tool at the end of the module, and intelligent practice exercises for each chapter and another practice for the overall module. Like in SQL Tutor and ACHRYA, the students can learn the concepts from the course material then assess their gained knowledge by carrying out the practice exercises included in each chapter. The system uses the learner control

strategy where students can select the topic they want directly while lessons are presented in a logical perspective. This is unlike ACHARYA and BITS which are program control software systems that recommend a concept only after pre-requisites are completed. The intelligent part uses the adaptive practice-based approach which helps students to identify their weakness points through a sequence of questions generated according to the students level and supports them by providing feedback, hints, explanations and suggestions when needed. The system is connected to a database that helps teachers and students track progress, generate different reports, and according to the given permissions update the quizzing and assessment systems. More details about the system are explained in the following chapters.

2.6 Conclusion

In this chapter we presented a review of ITSs. We started with an overview of ITSs defining ITSs and explaining the main different components making an ITS. The ITS components discussed were the Expert Knowledge Model, the Model, the Tutorial Planning Model, the Communication Model and the Evaluation Model with some description for each model role and functionality. Following the overview, we discussed ITS history and some previously related work of ITS. We learnt about the development of previous CAI and the importance of adding some "intelligence" to these systems to enhance education and individualize instruction which lead to the different ITSs. In our research we focused on adaptive and practice based systems giving some examples of developed systems. One very important area of our study was stressing on the importance of the role of cognitive science and reasoning about the problem the same way human do. We explored the ACT* theory which distinguishes between three types of memory structures: declarative, procedure and working memory with some explanation for each memory function and role in the cognitive process.

Furthermore, we explained the importance of Web-based tutoring systems and explored the different available types of Web-based Tutoring with comparison made between them and examples given. The discussed types were: the Static, the Personalized and the

Adaptive types of web-based tutoring systems. Moreover, we studied and discussed the different types of ITSs. To categorize existing ITSs we found that scientists had used different classification methods. In this chapter we introduced the different categorization methods with some examples of ITSs for each method. The three classification methods were: The Domain Method versus Practice based classification method, Abstraction of learning versus knowledge type of instruction (aka Emphasis of Instruction which includes two tutors: Cognitive tutors and knowledge based tutors) classification method and the third method was classifying ITSs according to their scope and functionality into seven categories: Curriculum Sequencing and Planning, Tutoring Strategies, Device Simulation and Equipment Training, Domain Expert Systems, Multiple Knowledge Types, Special Purpose Systems and Intelligent/Adaptive Hypermedia Systems. At last, we have explored some current ITSs which teach computer subjects in an attempt to plan and build a system that combines the benefits of other systems while avoid the drawbacks. Our system is the first ITS system that covers ICDL/ECDL module 1 material taking into consideration the different levels, backgrounds and needs of learners and by combing all the 6 types of learner control strategies to produce a flexible comprehensive system.

Our system in this paper is a cognitive tutor which teaches computer basics in a procedural way. However, the system embeds AI in the practice section where it asks a question, evaluates the answer, and depending on the student response it gives feedback, explanations and recommendations if needed. Then, it generates the next question and individualized instruction is given depending on the student model. This is rule-based that uses a set of "if-then" rules and follows a specific procedure to make sure students understand a topic before going to the next one. According to this and from our research in this chapter, the intelligent part of our system is classified as an adaptive WBT (online) system which uses AI to imitate the teacher decision making and support. In addition to this our system is considered as a model tracing cognitive, domain expert intelligent tutoring system at the same time it is using the problem sequencing and planning approach and techniques. The system uses the process practice based instruction cognitive system approach that monitors student progress through a problem solution by matching the student response to a student module an reproduce teacher help, direction

and guidance to teach computer basics fundamentals as module one for ICDL International Certification. This approach can be used and fulfill the needs of other reputed International exams and certificates too.

Our system in this paper combines the benefit of the student control system where students have the flexibility of selecting the topic, the sequence, practices and performing the quizzes and the program control system where during the intelligent practices the system make the learning decision of sequence of questions generated according to the learner level to help and guide the student through the learning process. In addition, students can control the time spent on any chapter, lesson or practice. Moreover, students can access the learning material anywhere internet exists and at anytime day or night. The system provides traditional lessons where learners can acquire information and help especially that intelligent practices recommend the chapter and lesson to refer to. Students can seek help from their teachers through the Questions and answers section where they can ask questions and learn from their peers' questions too. Also, the system provides a comments section too to encourage collaborative learning and exchanging experiences. From the comments and the questions sections, students can enter their email and work collaboratively with each other if they like.

Our system benefits from the convenience of the online systems, the cognitive ITSs to full fill knowledge gaps, besides many other features which will be explained later in the following chapters. It is a comprehensive system that serves different types of learners who can access it at their own convenience.

Chapter 3: System analysis and features

In this Chapter, we propose a new initiative for an online cognitive practice-based ITS to teach the ICDL/ECDL module 1 in an attempt to enhance the instructional method and to cover a broader sector of the community. We will discuss the methodology, analysis, and techniques used to identify the system's main components and requirements.

3.1 Methodology

Besides the literature review presented in Chapter 2, community needs were studied, in one important step of this project, to survey and identify learners' and teachers' different needs for the ICDL/ECDL module 1. Moreover, additional survey was done to study and understand the different learning and educational theories. This is in addition to the technical survey done to select the suitable software and to follow the right educational web design techniques.

3.1.1 Community needs survey

Two different versions of the survey were prepared to understand both online learners' and educators' needs. The survey included 12 Information Technology teachers, 21 high school students and 13 adult learners from different organizations. The learners' survey inquired about their needs in terms of functionality and design. The educators' version of the survey concentrated on two aspects. The first was on their needs and requirements as educators and the second was on their feedback, from their experience with learners, inquiring about what they believe a successful education system should include.

To conclude the survey results, learners emphasized the importance of the system interface in terms of simple straight forward design, navigation smoothness, instruction

clarity, language proficiency, flexibility of content, and instructional units accessibility. Moreover, some learners mentioned the importance of the human teacher in providing help, feedback and assurance and asked for established communication methods to keep in touch and feel connected with the teacher and other learners.

On the other side, teachers have emphasized the importance of tracking students, providing help and assurance, and keeping in touch with them. Moreover, teachers asked for a reporting system where they could generate reports to evaluate both students progress and system effectiveness besides an available evaluation tool. This was in addition to the importance of clear instruction and lesson explanation, clear feedback to students and the importance of methods of electronic communication which can offer real human care and touch to motivate and support learners and enable them to share information, collaborate and learn from each other.

The following is a further survey carried out to study, relate, and benefit from the learning and educational theories.

3.1.2 Learning and educational theories survey

In this section, we discuss the different types of learning and educational theories, learning process and requirements, a comparison between learner control and program control approaches, then we discuss in more details some educational theories and how our system meets them.

3.1.2.1 Types of learning and educational theories

According to Clark and Mayer (2002), three types of learning were identified during the past hundreds of years. These types can be seen in different software tutoring systems as follows. The first type is *Receptive learning*: this type of learning involves information acquisition used in lessons. It includes presenting information with little practice in order to satisfy the “inform” goals. The second type is *directive learning*: this is a response strengthening type of software which requires frequent responses from the learner with immediate feedback in order to satisfy the “perform-procedure” goals. The third type is

Guided discovery: this type is concerned with knowledge construction by providing lessons, practice exercises and resources in order to satisfy "perform-principle" goals. These types reflect the learning procedures for knowledge acquisition to improve learners' problems solving skills (Li and Soh 2003).

3.1.2.2 Learning process and requirements

It is very important to understand how people learn and what affects the learning process in order to design successful learning software. Due to the human learning mechanism which was previously discussed in Chapter 2, ITSs rely on the following four key processes (Clark and Mayer 2002). First, the learner should focus on key text or graphics to select what will be learned. Then, the learner must rehearse the information in the working memory to organize it and integrate it with existing knowledge in the long term memory. Next, the knowledge in the long-term memory is available to be retrieved when needed. Finally, the meta-cognitive skills manage and adjust these processes.

Meta-cognitive refers to the mental management processes that supervise information processing. A good ITS should provide some management processes to help learners with low meta-cognitive skills and enable learners with high meta-cognitive skills to take full advantage of them and enhance learning (Li and Soh 2003).

3.1.2.3 Learner Control and Program Control

In Chapter 2, we introduced the concepts of learner control and program control. In this chapter, we want to differentiate between these two strategies in order to build our system in a way where we can benefit from both systems.

Previous studies have been conducted to assess the learner control effectiveness and efficiency in comparison with program control systems. Lee and Lee (1991) examined these studies and focused on two learning phases: knowledge acquisition and review in conjunction with learners' generic meta-cognitive skills and previous domain knowledge. Their findings stated that learner control is superior to program control for tasks of simple

content structure in which a low level of domain knowledge is required (Lee and Lee 1991).

In 1996, Young also compared the outcomes of learner control in contrast to program control. He mainly compared learners with high and low meta-cognitive levels in either learner control or in program control. He found that learners with low meta-cognitive skills learnt less in learner control than in program control while learners with high meta-cognitive skills learnt much better in learner control than in program control (Young 1996).

Other studies compared the two strategies through complex evaluation work (examples are Park and Tennyson (1980), Ross et al. (1980), Gray (1985), and Goetzfried and Hannafin (1985)). Their findings were that learner control strategy skills largely depend on learners' meta-cognitive skills, content structure, and quality of the provided lessons. These studies concluded that the learner must know when and how to control the learning process. A learner with high meta-cognitive skills may do better with learner control while a learner with low meta-cognitive skills and with poor lesson design and structure will do worse than with a good lesson in program control (Lee and Lee 1991).

The above studies show the necessity for well designed and structured lessons to give the highest opportunity for all learner levels to benefit from the proposed system and improve their learning skills. This was helpful in identifying lesson objectives and work needed on designing straight forward and well structured lessons. Another important point is to take into consideration the student meta-cognitive level which leads to combining the benefits of the two strategies in one system. To facilitate this we offer the flexibility of selection and pace and at the same time provide guidance and direction.

3.1.2.4 Educational theories

To design effective educational software, educational theories and different types of learners should be taken into consideration (Kearsley 2008). We discussed earlier in this chapter the different types of learning and now we discuss the different educational theories and how we meet them. Our system mainly depends on the cognitive load theory which we have discussed earlier in Chapter 2 but at the same time the system meets many other theories. Examples are as follows.

Experiential learning

'Experiential learning' is often referring to a structured learning sequence which is guided by a repetitive model of experiential learning. Experiential learning includes accidental or unintentional learning in its basic realistic forms. It is frequently described as 'learning through/from experience'. This type of learning can apply to *any* kind of learning through experience. It addresses the needs of the learner. The most lasting learning is the self-initiated learning (Kolb 1984).

Our system supports this type of learning by offering the learners the flexibility of selecting the topic to study and the pace besides providing the intelligent practices where students apply the knowledge they have acquired, identify the gaps in their knowledge and learn from the system hints, feedback and recommendations. The learner has the flexibility to redo these practices as many times as they want and re-read the lessons as many times as necessary.

Levels of processing theory

The 'levels of processing' theory suggests that the deeper the information is processed the more likely learners can remember it. It can be summarized by saying that superficial processing occurs at the early stages such as coding the stimulus in means of their physical characteristics such as the visual characteristics of the letters and fonts where a word is printed, or the auditory features of a sound. Information is processed more deeply as it passes on through the system. Deep processing involves coding the stimulus more conceptually in terms of its meaning. Hence, visual and auditory coding are shallow, but semantic coding is deep (Crane J. 2008).

Our system supports this theory by providing a logical content sequencing and giving the learner control in selecting the topic to read, the intelligent practice to perform. This is in addition to offering the flexibility in time and number of repetitions where learners' can get back to any of the tutoring system component.

Component Display Theory

Component Display Theory (CDT) classified learning along two aspects. The first is content which includes facts, concepts, procedures, and principles. The second is

performance which consists of remembering, using, and generalizing. The theory identifies four main presentation forms: 1) rules which are the expository presentation of a generality. 2) Examples which are the expository presentation of instance. 3) Recall and 4) practice. Other secondary presentation forms may include prerequisites, objectives, help, mnemonics, and feedback (Merrill, 1994).

Our system provides content by means of a sequence of lessons, examples and images, practices where the learners recall the information and practice the knowledge they acquired besides the quizzes which help learners assess their knowledge level. Moreover, the system provides lesson objectives for every lesson, feedback and help when students carry on practices besides the feedback given after each quiz.

Structural learning theory

According to structural learning theory, students learn rules which consist of a domain, range, and procedure. Problem solving may be facilitated when higher order rules are used, such as rules that generate new rules. Unlike information processing theories which often consider complex control techniques and production rules, structural learning theory teaches the simplest solution path for a problem first then teaches a more complex path until the entire rule has been mastered (Scandura 2004).

The intelligent part in our system uses the procedure approach in teaching any topic. It starts with a basic question and advances gradually to introduce more complex material. It is a rule-based system where learners should go through a specific sequence.

Information Processing Theory

Miller (1956) provided two theoretical ideas that are fundamental to cognitive psychology and the information processing framework. The first concept is "chunking" and the limited capacity of short term memory. He suggested that short-term memory could only grasp 5-9 chunks of information. A chunk is any meaningful unit and could refer to digits, words or people's faces. The second concept is the Test-Operate-Test-Exit (TOTE). The concept suggests that TOTE should replace the stimulus-response as the

basic unit of behavior. In a TOTE unit, a goal is tested to see if it has been achieved or not. If not a process is performed to achieve the goal. This cycle of test-operate is repeated until the goal is finally achieved or discarded (Miller et al. 1960).

In our system, the module is chunked into chapters and consequently in lessons. Chapters have their own quizzes and practices. Intelligent practices can be done as many times as necessary for the learner to comprehend and make sure they meet all chapter outcomes. Learners can go back and read the recommended lesson by the intelligent tutor and redo the practices.

Situated Learning

Situated learning is a general theory of knowledge acquisition. It has been applied in the context of technology-based learning activities for schools that focus on problem-solving skills (Cognition & Technology Group at Vanderbilt, 1993). Social interaction is an important component of situated learning. It encourages learners to become involved in a "community of practice" which represents certain beliefs and behaviors to be acquired. As the beginner or newcomer moves from the periphery of this community to its center, they become more active and engaged within the culture and hence assume the role of expert or old-timer (Lave & Wenger (1991).

"Cognitive apprenticeship" supports learning in a domain by enabling students to acquire, develop and use cognitive tools in authentic domain activity. Learning, both outside and inside school, advances through collaborative social interaction and the social construction of knowledge." Brown et al. (1989) also highlights the need for a new epistemology for learning. A theory puts emphases on active perception over concepts and representation. Suchman (1988) investigated the situated learning framework in the context of artificial intelligence.

In our system in this paper, we use Artificial Intelligence as a guide to help learners individually and identify the knowledge gaps learners may have. It gives learners recommendations and direction to enhance their learning process and skills. In addition, the system provides two communication tools, the comments and the questions areas. These tools are designed to introduce learners to each other where they can exchange

information, knowledge and emails if they like. Learners have the flexibility to submit comments, ideas and/or share experiences. Moreover, learners can ask the teacher questions and find answers besides learning from other learners' questions and teachers' feedback.

Conditions of Learning

This theory suggests that there are several different types or levels of learning. Each different type of learning entails different types of instruction. According to Gagne et al. (1992), there are five major categories of learning: verbal information, intellectual skills, cognitive strategies, motor skills, and attitudes. Different internal and external conditions are required for each type of learning.

The theory outlines nine instructional measures related to cognitive processes (Gagne, Briggs & Wager, 1992):

1. gaining attention (reception)
2. informing learners of the objective (expectancy)
3. stimulating recall of prior learning (retrieval)
4. presenting the stimulus (selective perception)
5. providing learning guidance (semantic encoding)
6. eliciting performance (responding)
7. providing feedback (reinforcement)
8. assessing performance (retrieval)
9. enhancing retention and transfer (generalization)

In our system, examples of how we meet the nine instructional events are as follows:

1. Gain attention -through the software availability, interface, and flexibility, the pre-quiz that learners do before they start the lesson
2. Identify objective – through the provided lessons objectives link and the pre-quizzes they carry on at the beginning of each chapter.
3. Recall prior learning – ask students to perform the pre-quiz.

4. Present stimulus – traditional lessons with images and easy with easy navigation and simple web page layout
5. Guide learning- through the sequence of lessons, the feedback of the pre-quiz and most important is the intelligent practices guidance and direction.
6. Elicit performance – ask students to perform the practice provided in each chapter
7. Provide feedback – feedback is provided intelligently with guidance in the practice part besides the feedback given after each type of quiz.
8. Assess performance- scores with the right answers are provided after each type of evaluation the pre- quiz, post-quiz and the assessment. In addition, the practice part gives immediate feedback and direction with explanations and recommendations.
9. Enhance retention/transfer – ask students to perform the post-test after carrying out the practice part as many times as required the required knowledge. After finishing all chapters, students are asked to do an assessment which covers all module objectives.

More details about the system design are explained in Chapter 4. Details about technical analysis and system implementation are discussed in Chapter 5.

3.1.3 Technical survey

To design our system we conducted a research to select the right software which could satisfy the planned system requirements. We needed to do some studies in the areas presented in the following subjections.

3.1.3.1 Web design

In order for learners to access the system online at their own pace and wherever they are, the system should be designed as a website. This website should be dynamic, stable and able to be accessed by different web browsers. Moreover, the design study should include the design of content, layout, navigation, flexibility, and interface.

3.1.3.2 Designing techniques

The study included the designing techniques for the look, navigation methods, and different functionalities of the website besides the technical side which includes different available platforms and software. The designing techniques, interface and functionalities of each part of the system are discussed in detail in Chapter 4, explaining why and how each component was designed and how it works. In Chapter 5, we discuss the system implementation in addition to the software used, why and how it was used in much detail.

3.2 Analysis of system components:

From the literature review and the research conducted, we identified the need for two different interfaces for different users: teachers and students. Each interface includes the required components and has the necessary permission as in Figure 3.1. In this section, we talk about the system different interfaces and also discuss in more details the system components.

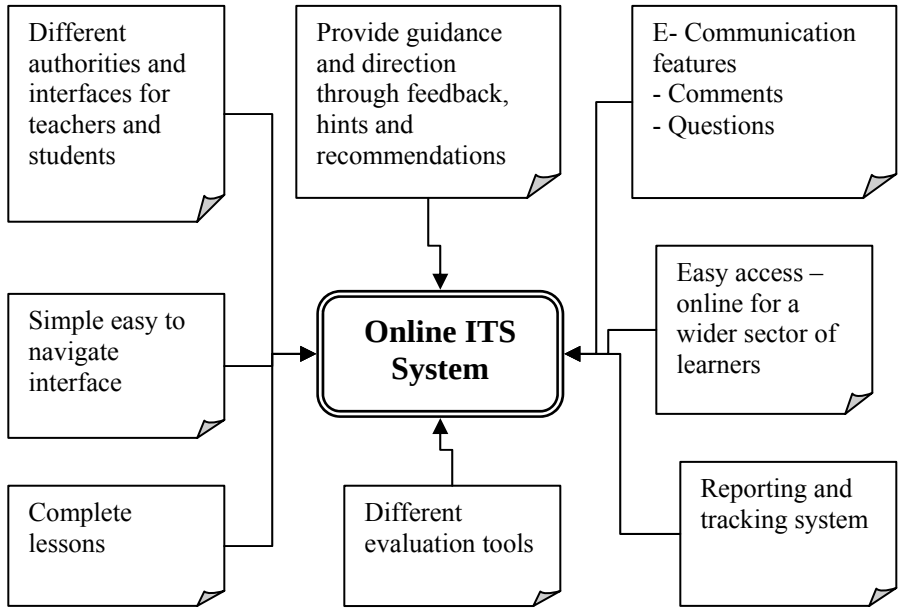


Figure 3.1: Online ITS features and requirements

3.2.1 System Interface

In this section we discuss in details the different user interfaces: the students and the teacher as follows.

3.2.1.1 The Student Interface

This interface provides complete lessons, assessment tools, basic reports, electronic communication tools with teachers and other students, practice exercises, individualized feedback and customized instruction.

3.2.1.2 The Teacher Interface

Besides all the student interface features that were mentioned above, the system provides an advanced reporting system which enable teachers to generate various types of reports, the flexibility to modify some features of the database contents such as questions, and answer students questions. Additionally, the system enables teachers to provide help with a human touch by providing established different communication methods where they can for example review and answer students' questions, add and review students' comments, and share their Emails.

3.2.2 System Components:

In the following, we talk about the main system components such as the lessons, the quizzes, the questions, the practices, the reporting system, and the communication tools besides explaining some other features.

3.2.2.1 Lessons:

Lessons are planned to meet the outcomes and objectives of the ICDL/ECDL syllabus with some images and clear instruction. Language used is English and lessons should be simple and forward to cater for non native English speakers as well.

3.2.2.2 Quizzes:

All assessments and quizzes are direct, with similar difficulty level of that in the real exam. The language is forward, clear and address all learners' levels. Pre-quizzes and post quizzes particularly are of the same level for accurate comparisons and evaluation of students' progress. Questions cover all the lesson objectives and the number of questions is flexible depending on the lesson and the intended outcomes. Assessment covers the complete module. After each quiz or assessment, a complete report with the wrong and right answers is displayed with the total score explaining the right answer for those wrong ones. Furthermore, teachers have the required permissions and flexibility to add, delete and modify any of the questions through a customized form.

3.2.2.3 Questions:

The ICDL/ECDL real exam environment uses multiple choice questions (MCQ) to evaluate the learners' level of knowledge. To prepare students for the real exam, it was decided to follow the same style of question in all quizzes, practice exercises and assessment. In the following, we explore MCQ in more details.

Multiple Choice Questions

Multiple choice questions are a form of assessment questions where examinees are asked to select one or more of the choices from a list. Test authors are usually trained using Bloom's taxonomy (Rizvanov F. and Lizotte R. (2000); Carneson, J et al. (n.d.), Kehoe J.(1995)).

Structure and considerations

Multiple-choice questions consist of a stem and a set of options. The *stem* is the beginning part of the item that presents the problem to be solved, a question asked of the respondent, an incomplete statement to be completed, or in some cases graphs, a case study or any other relevant information. The options are a list of possible answers that the examinee can choose from. The correct answer is called the *key* and the incorrect answers called *distracters* (Kehoe, Jerard 1995). Figure 3.2 gives an example of an MCQ.

Multiple choice questions have many advantages if the author of the questions is well trained (Kehoe, Jerard 1995). Test reliability is improved with larger numbers of questions on a test (Downing 2004). This is in addition to the fact that multiple choice tests often require less time to administer which results in a better and more comprehensive evaluation tool. With well trained examiners, multiple choice questions help in the development of objective assessments. Examinees are graded only on their selections and hence on their knowledge of the subject. Factors irrelevant to the assessed material, such as clarity of presentation, do not affect the result in a multiple choice assessment.

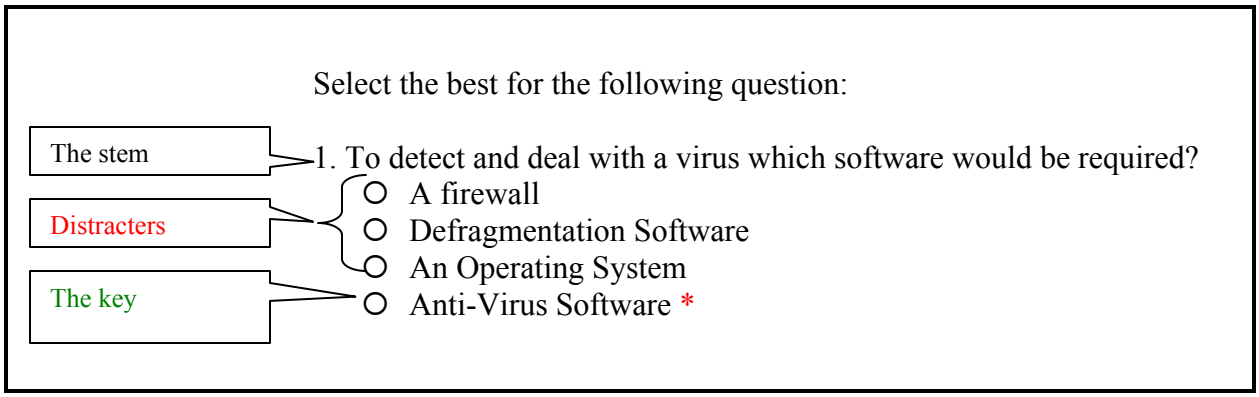


Figure 3.2: Example of a multiple choice question

MCQ are not easy to construct, however with experienced authors, a good set of MCQs can be designed for the cognitive levels they can test. With careful writing, it is quite possible to construct MCQs which test all cognitive levels including the more complex

ones. In addition, MCQ tests can be reused, item banked, autocorrected, and gives quick feedback on students performance (Isaacs, G 1994).

On the other hand, with untrained authors, ambiguity and subjectivity may result. If the writer could not interpret the information right, this can result in an "incorrect" response, even if the examinee's response is potentially valid. This is in addition to the fact that some examinees, who do not know the answer, can guess the answer and get the mark while some other examinees who may know parts of the answer, select the wrong response and loose the mark. Furthermore, MCQ tests can never test creativity and literacy (Isaacs, G 1994).

Many notable international examinations use MCQ style to assess students' knowledge. Examples are in the following table (table 3.1):

Test Name	Description
SAT	**The SAT Reasoning Test is a standardized test for college admissions in the United States.**
MCAT	**The Medical College Admission Test, commonly known as the MCAT, is a computer-based standardized examination for prospective medical students in the United States and Canada**
LSAT	**The Law School Admission Test (LSAT) is an examination administered by the Law School Admission Council (LSAC), intended to provide law schools in the United States and Canada with (to quote the LSAC) "a standard measure of acquired reading and verbal reasoning skills that law schools can use as one of several factors in assessing applicants."**
Examination of the IB Diploma Program	**The International Baccalaureate (IB) Diploma Programme (DP) is an educational programme examined in one of three languages (English, French or Spanish) and is a leading university entrance course.**
ASVAB	**The Armed Services Vocational Aptitude Battery (ASVAB) is a multiple choice test, administered by the United States Military Entrance Processing Command, used to determine qualification for enlistment in the United States armed forces**
TOEIC	**Test of English for International Communication (TOEIC) measures the ability of non-native English-speaking examinees to use English in everyday work activities.**
Multistate Bar Examination	**The Multistate Bar Examination (MBE) is a six-hour, two-hundred multiple-choice question examination administered as a part of the bar examination in almost all jurisdictions of the United States.**
AP	**The Advanced Placement Program, commonly called AP is a program that offers college level courses at high schools across the United States and Canada**
PSAT	**The PSAT/NMSQT, or Preliminary SAT/National Merit**

	Scholarship Qualifying Test, is a multiple-choice standardized test administered by the College Board and National Merit Scholarship Corporation (NMSC) in the United States.
GRE	**The Graduate Record Examination or GRE is a standardized test that is an admissions requirement for many graduate schools principally in the United States, but also in other English speaking countries**
FE	**Fundamentals of Engineering exam**

Table 3.1 – Examples of international examinations that use MCQ[1]

3.2.2.4 Practices

Practices are the intelligent part of the system. They intend to identify learners knowledge gaps while the learners answer a sequence of MCQs. Questions are designed to individualize the instruction in a procedural manner. Therefore, learners are examined in specific areas and according to the selected answer, the next question is generated. Feedback, explanations and recommendations are immediately provided to the learner after each question besides recommendations when there is a need for them. Each wrong answer chosen is authored to reveal learner misconceptions and weaknesses and therefore provide reinforcement in this particular area. Each student path differs based on his/her responses. More details and the algorithm are explained in Chapter 4.

3.2.2.5 Reporting systems

Students and teachers need to track progress in different ways. Students need to track the lessons they visit, the quizzes and assessments they finish including the scores. On the other hand, teachers need to track individual student progress and the progress of the students' overall. To fulfill the different needs for both teachers and students, each party is given the necessary permissions and different reporting systems were created.

Students' reports. Students can access the following reports: the first report is the lessons visited report which gives a good summary of the lessons and when they were visited by the student. The other report is the test history report. This report shows details about each chapter assessment performed by the student. It includes information about the assessment number, content, score and time carried out.

Teachers' reports. Teachers can generate 6 different reports to evaluate both the students' progress and the system functionality. These reports are as follows: 1) report lists all the lessons visited by students including details about the lessons, the student who visited the lesson and time the lesson was visited, 2) The assessment report gives the information and details about students who performed the module assessment and their scores. 3) The pre quizzes report, lists information about students who have completed the pre quizzes, their scores and the time the quiz was submitted, 4) the post quizzes report is similar to the previous one but the details given are about the post-quizzes. 5) The students' progress report compares the post-quiz results with the pre-quiz results and calculates the progress rate. This report can be used to monitor a student's progress and at the same the time system effectiveness. 6) The last report analyses the quiz results and this can be done for any quiz or assessment to find the difficulties students face with the content and identify system weaknesses and how to fix them. This report can be exported to either PDF or excel for further analysis.

Students' information and results are kept in the database for future reference and more reports can be generated when there is a need.

3.2.2.6 Communication tools

As mentioned previously in Chapter 1, studies have shown that the most effective and efficient education is done by using the professional human tutors (Bloom 1984; vanLehn et. al. 2005). Although scientists have adopted the human tutor as a model in their ITSs to guide students through the learning process, still learners miss the support and human contact tough when they are for any reason stuck in the process or need extended help or

even when they need at least assurance. To tackle this, two different communication areas have been provided in this system: the questions area and the comments area.

The questions area is where students can refer to a real tutor who can answer their questions and deal with their concerns, crossing the distance barriers and providing support help when extra help is needed. This tool can be used to extend learners thoughts and knowledge by helping them learn from other learners' questions and feel more connected and belongs more to a virtual community instead of feeling lonely in the learning process.

Another tool is the comments tool where learners can share their comments with each other and with teachers or educators. This is a good tool to consider while evaluating the overall system performance. Learners can also feel the importance of their feedback in enhancing the system. This tool can be used as a collaborative tool where learners can help each other and they may share their emails as an option if they like to create study groups or collaborative groups. Older students may find it a very good opportunity to give hints and share their experience. These tools for security and validity reasons are monitored by the teacher. The teacher has the necessary permission to delete any unwelcome or irresponsible comments.

3.2.2.7 Other features:

In addition to the previously mentioned system features, both teachers and students can change their password. Furthermore teachers have the flexibility to add students to the system. Moreover, the system is designed in a flexible way using and storing information in a database for future enhancements and to accommodate more functionalities as required by the different users.

3.3 Conclusion

In this chapter, we discussed the methodology of our thesis and talked in details about the main components of our proposed system. We carried out different surveys in addition to the literature review done in Chapter 2. In these surveys, we studied the community needs, explored different learning and educational theories, learnt about the learning process and requirements, differentiated between learner control and program control tutoring systems and compared our system to some educational theories. Furthermore, a technical survey was performed and more information about the selected software and the designing techniques used to develop the online system that will be discussed later in chapters 4 and 5.

The conclusion of the above community survey, both online learners and educators emphasized the importance of an attractive, straight forward interface with different user permissions, instruction clarity, language proficiency, and system accessibility, availability of a reporting system, assessment tools and established communication. These findings and more have been taken into consideration during the phases of analyzing and designing the system. To fulfill users' needs, two interfaces were designed: the student and the teacher. Each interface provides different accessibility permissions.

To design an effective educational website, further studies were performed in the learning and education field. From these studies we learnt the three types of learning used in different software tutoring systems and how they work. The three introduced types were: the receptive learning, directive and guided discovery learning. Our system provides lessons, practice exercises and other resources which mainly describes the *Guided discovery* type of learning. Furthermore, it was important to understand how people learn and understand the learning process which helped us better in identifying the system components during the system analysis phase. Also, we introduced the concepts of learner control and program control strategies to help us building a system those benefits from both strategies and improve the learning process. Hence, our system is designed as a learner control which gives the user the flexibility to select the lesson as an example

while during the intelligent practice the system guides the student in a specific sequence of questions according to the student level and responses.

From the literature review and the different surveys, we identified the system components to include web-based: Lessons, different types of quizzes and assessment, intelligent practice exercises, reporting system, two types of communication tools, besides some other features like the flexibility to change the password. Questions used in the different types of quizzes, assessments and the practice exercises were MCQ. This is to imitate the real exam environment and for the MCQ objectivity. We explained in details the structure and how this type of assessment questions work giving some examples of other international examinations that uses MCQ.

In the following chapters we discuss in more details the system design and implementation in addition to the testing phase results and conclusions.

Chapter 4:

Technologies and tools used to build the proposed system

To develop the system described in Chapter 3, different technologies have been used to satisfy different needs and uses. The proposed system consists of two main parts: the website and the embedded expert system. In general, the website provides the online domain information for students to acquire knowledge and learn new concepts. This is in addition to the evaluation and communication tools offered to serve different users for various purposes as discussed in Chapter 3. The expert system is the intelligent practice based part of the system where misconceptions are revealed and according to students' levels, the instruction is adapted and individualized, and guidance is offered in a cognitive manner.

Each part of the proposed system requires different tools and technologies to be constructed and integrated. A technical survey was carried out to identify the required technologies and tools. According to the survey results, for the website part, we used ASP.NET as a development tool and VB.NET as the programming tool while the XML was used to build the site map, and SQL 2005 Server to create and manipulate the database system including tables and relations. While to develop the expert system part, we used CLIPS, a well-known expert systems building tool, as our intelligent development environment and VC++ to create the standalone expert system executable files, and the "ZIP 2 Secure EXE" tool, a self extracting software, which enables users to run the compressed lesson automatically after extracting it to the desktop.

In this chapter, more information is given about each used technology and its role.

4.1 Website Development tools

The technologies used to develop the website are discussed in the following subsections. They are: ASP.NET 2005, VB.NET 2005, XML and Microsoft SQL Server 2005.

4.1.1 ASP .NET 2005 (Active Server Pages)

Since 1995, Microsoft has been constantly working to shift its focus from Windows-based platforms to the Internet. As a result, Microsoft introduced ASP (Active Server Pages) in November 1996 (Microsoft 2008).

ASP.NET is a platform for rapid development of powerful web-based sophisticated, interacting applications. It is not the only platform available for Web-based applications. There are many alternatives such as PHP, ColdFusion, and several Java based tools but these tools cannot compete with ASP.NET which is the integrated development environment that combined a web-page editor, a code editor, a debugger and several other development tools into one programs (Lowe et al 2006).In addition, ASP .NET is a powerful database-driven development tool. Moreover, ASP.Net is object-oriented and has many programming tools that allow for faster development and more functionality ((Lowe D. et al. 2006, Microsoft 2008). ASP.NET easily works with ADO .NET using data-binding and page formatting features. According to Mischel J. in startvbdotnet website (2008), some ASP.NET features and comparisons with other tools are as follows (Startvbdotnet.com[3], Mischel J. 2008).

Performance and scalability

[3] Startvbdotnet.com is a resource for new comers and VB veterans migrating to VB .NET. Teaches VB .NET and ASP.NET

ASP.Net was tested and found to be over 10 times faster for the average user than Java's J2EE technology. While there have been some debates about the methods of the testing it is interesting to note that this has been validated by 3rd parties. The two aspects of ASP.Net that make it fast are compiled code and caching. The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested.

Multiple language support

Programmers can actually write their code in more than 25 .Net languages (including VB.Net, C#, and JScript.Net). This allows programmers to develop sites in the language they know best.

Simpler, less coding, easier code and development and validation controls

ASP.NET makes development simpler and easier to maintain with an event-driven, server-side programming model. Additionally, ASP.NET drastically reduces the amount of code required to build large applications. Moreover, there is no need to register components because the configuration information is built-in. This is in addition to the fact that ASP.NET pages are easy to write and maintain because the source code and HTML are together in the same page. Also, ASP.NET validates information entered by the user without writing a single line of code.

Security, memory leak, deadlock and crash protection

The HTML code produced by the ASP.NET page is sent back to the browser. The application source code written by the developer is not sent and accordingly is not easily stolen. In addition, the Web server continuously monitors the pages, components and

applications running on it. If it notices memory leaks, infinite loops, other illegal software or activities, it seamlessly kills those activities and restarts itself.

Flexibility and Simplified deployment

An important feature of ASP.NET is that the source code is executed on the server. This approach gives pages more power and flexibility. With ASP.NET, the developer can deploy an entire application as easily as an HTML page. He only has to copy the page to the server. No need to run regsvr32 to register any components, and configuration settings are stored in an XML file within the application.

In our proposed system we used ASP.NET 2005 platform to construct the web-based system taking into consideration future improvements and expansion of the system. The selected programming language for the code was Visual Basic and the SQL server 2005 as the data platform to communicate with the database.

4.1.2 Visual Basic .NET 2005

Visual Basic .NET or VB.NET is an object-oriented computer language that can be viewed as an evolution of Microsoft's Visual Basic (VB) implemented on the Microsoft .NET framework. It has been reengineered to be a first-class language on the Microsoft .NET Framework. It provides the easiest and most productive route for new developers and existing Visual Basic programmers to quickly build Windows–based applications, next generation XML Web services, thin-client Web applications, and software for Mobile devices (Lowe D. et al. 2006).

In this system, Visual basic .NET is used in the code-behind files and to calculate students' marks after each Quiz or assessment.

4.1.3 Microsoft SQL Server 2005

The Structured Query Language (SQL) is used to communicate with a database. It is the standard language for relational database management systems. Some common relational database management systems that use SQL are Oracle, Sybase, Microsoft SQL Server, Access, Ingres (Microsoft 2008).

Microsoft SQL Server 2005 is a data platform leader. It is comprehensive, integrated data management and analysis software. It enables organizations to reliably manage critical information and confidently run complex business applications. In this system we have used Microsoft SQL Server 2005 to create and alter our tables. We have used it to run queries, create reports and with the communication tool to store data (Microsoft 2007, Lowe D. et al. 2006).

4.1.4 XML (Extensible Markup Language)

XML has become a common format for storing and transferring data. In this website we have created an xml file to store website information to use as a data source for the site map.

4.2 Expert System Development tools

In this is section we present the intelligent part of the system. It is built in the form of practice exercises to identify knowledge gaps and treat them. The system generates a sequence of questions adapted to students' levels and cognitive instruction is provided accordingly to clear misconceptions. In short, technically speaking, the system branches to the next step according to the student response in an "if then" manner. As a result, to develop such expert system we required a rule based intelligent language. According to the technical survey we conducted, we used CLIPS as our intelligent language. CLIPS is discussed in the following subsection.

4.2.1 CLIPS - C Language Integrated Production System

CLIPS is a public domain, productive -chaining rule-based language expert system tool. It provides a complete environment for the construction of rule and/or object based expert

systems. It was developed by the Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center. CLIPS is designed to facilitate the development of software to model human knowledge or expertise. (NASA 1991; Martin et al 1992; Gonzalez et al 1993; Giarratano et al 1994).

Rule-based programming is one of the most commonly used techniques for developing expert systems. In this programming paradigm, rules are used to represent heuristics, or "rules of thumb," which specify a set of actions to be performed for a given situation. A rule is composed of *if* and *then* portions. The inference engine is a tool which automatically matches facts against patterns and determines which rules are applicable. The actions of applicable rules are executed when the inference engine is instructed to begin execution. The inference engine selects a rule and then the actions of the selected rule are executed. The inference engine then selects another rule and executes its actions. This process continues until no applicable rules remain. In summary, Figure 4.1 shows the basic components of the intelligent system (Madey et al 1992).

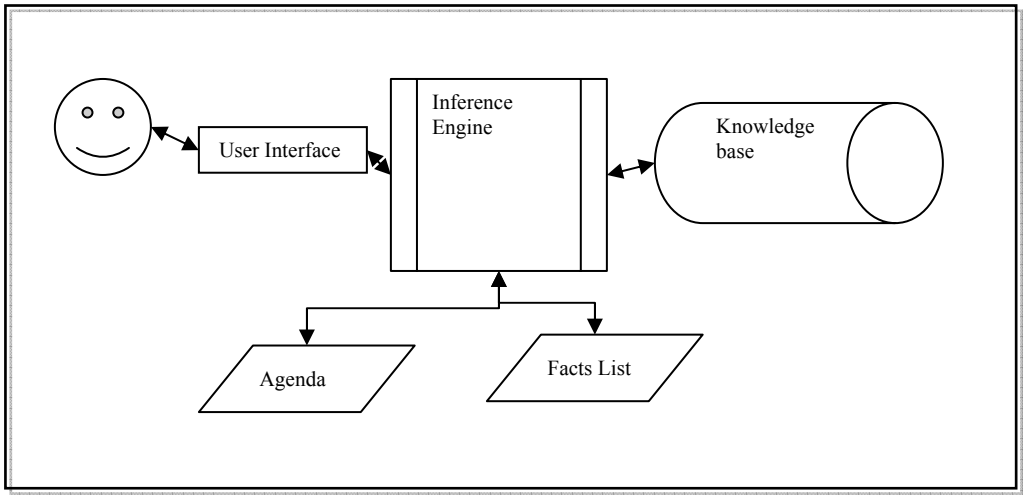


Figure 4.1: Basic components of an intelligent IT system (adapted from Madey et al. 1992)

4.3 Integration tools

In the previous section, we discussed CLIPS as the intelligent language we utilized to build the intelligent part of our proposed system. The next step was to find the tools needed to integrate the two parts making up our system in this thesis.

CLIPS is been designed for full integration with other languages such as C. In fact, CLIPS is an acronym for C Language Integrated Production System. Rules and objects form an integrated system too since rules can pattern-match on facts and objects. In addition to being used as a stand-alone tool, CLIPS can be called from a procedural language, perform its function, and then return control back to the calling program. Procedural code too can be defined as external functions and called from CLIPS. When the external code completes execution, control returns to CLIPS (Giarratano, 2002). In this system our CLIPS programs are called from Visual C++ .NET 2005. The next integration step was to create a self extractable zipped file which requires the least user interference taking into consideration the students capabilities. To do so, and from the technical survey we decided to use “ZIP 2 Secure EXE” software. In the following sub-sections, we discuss in brief, both Visual C++ and the ZIP 2 Secure EXE tools.

4.3.1 Visual C++ .NET 2005

Visual C++ provides a powerful and flexible development environment for creating Microsoft Windows–based and Microsoft .NET–based applications besides many other forms of applications (Riley G. 2008).

In our proposed system, we employed Visual C++ to create the expert system exe files from the CLIPS files. The algorithm and implementation are discussed in details in chapter 5.

4.3.2 ZIP 2 Secure EXE

"ZIP 2 Secure EXE" is a utility program that creates self-extracting EXE files for Microsoft Windows© platform. Self-extracting EXE files are executable programs that

contain a ZIP file and the software necessary to unzip the contents. This program runs the required (target) program automatically after unzipping. Users can unzip the contents of a self-extracting EXE simply by running it like any other program. To be able to add the file to the website so the student can access them online, files were zipped using WinZip then and exe file was created using ZIP to Secure EXE program. ZIP to Secure EXE program creates a self extracting exe file besides that it automatically runs the program while WinZip exe self extracting file which only unzips the file. This is very convenient to beginner users who may find it difficult to find out and run the file without assistance (Chilkat Software 2008). Figure 4.1 is a screen shot of the ZIP to Secure EXE program dialog box.

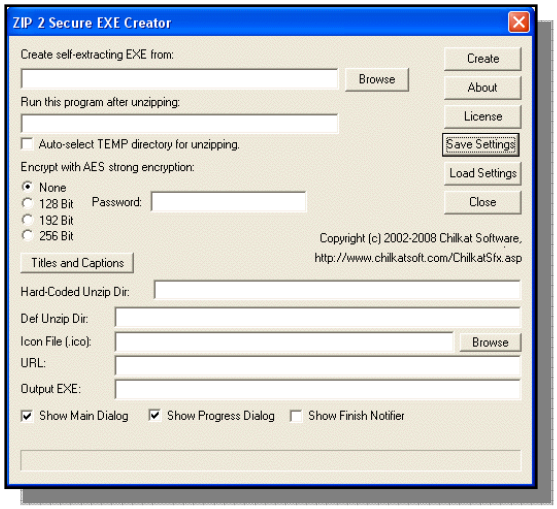


Figure 4.2: Screen shot for the ZIP to Secure EXE program dialog box.

After creating the Zip to secure exe file, we add it to the website and the user can simply download any of the intelligent practice exercise from a link can be accessed from every lesson.

4.4 Conclusion

In this Chapter we discussed the technical survey findings. This Survey covered a study about the different technologies and tools we required to build our system as described in Chapter 3. The survey included technologies used to build the different parts making up our proposed web based cognitive, adaptive practice based system. At the beginning we discussed the technologies used to build the overall website which included ASP.NET, VB .NET, Ms SQL 2005 Server and XML. Then we illustrated the expert system part and the need for the intelligent language to build the practice based intelligent part of the tutoring system in an adaptive, cognitive approach. We introduced CLIPS with some details as our intelligent language. At last, we described the integration tools we entailed to integrate the both parts and make the intelligent practice exercises accessible online. To accomplish this, we employed Ms VC++ to create the exe files then "ZIP to Secure EXE" tool to create the self extracting zipped file which can be added to the website and simply accessed online from a hyperlink.

Chapter 5: The Proposed System Architecture

In Chapter 3, we analyzed and identified the system different interfaces and components according to the various surveys carried out besides the literature review and study conducted in Chapter 2. As introduced in the previous chapters, our system in this thesis is an online intelligent tutoring system which helps students to prepare for the ICDL/ECDL module 1 certification exam. The system can be used also to teach computer fundamentals and concepts for beginners. This system combines the benefits of the traditional learning systems and the intelligent cognitive, practice-based tutoring systems. Also, it combines the learner control environment used for the traditional part of the system with the program control environment used in the intelligent practice exercises. To fulfill the proposed system needs, we developed a dynamic, database driven website with an embedded intelligent tutoring system.

In this chapter, we explain the system architecture, design and implementation. Therefore, we explain in details the design, algorithm, code and implementation of every part making up the online/ web-based system.

5.1 Overall System architecture

Figure 5.1 shows the overall architecture of the proposed web-based ITS. This system is designed to meeting different educational and learning theories as mentioned in Chapter 3. It consists of the teacher and the student interfaces. Depending on the username the system directs users to the appropriate page giving them the required/necessary permission.

Moreover, the system is made up of the following components: lessons where students acquire knowledge and learn concepts, different types of quizzes to assess students previous and acquired knowledge, intelligent practice exercises which is the intelligent

part of the system to assess students readiness and reveal knowledge gaps, different reporting systems to track and analyze students progress, and the communication tools which allow students to ask questions, give feedback, and exchange experiences . These are in addition to some other tools such as the change password tool, and add new students' tool and the questions

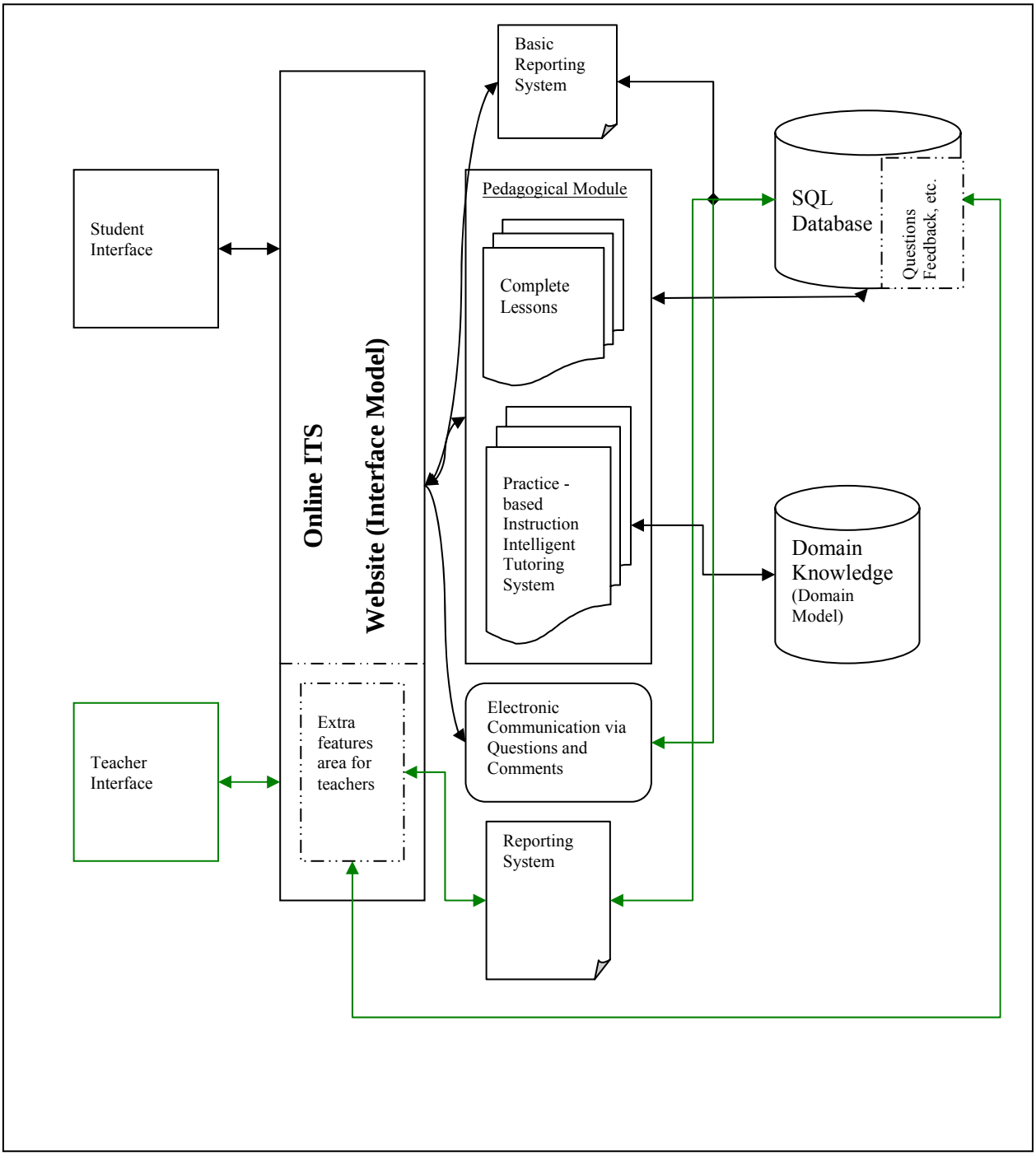


Figure 5.1: System Architecture

Figure 5.1 shows that we have two different interfaces. The teacher interface includes all the features that the student interface has besides other features that are necessary to teachers only. Both interfaces have access to a complete list of lessons, practices, different types of quizzes and assessment, electronic communication methods and a shared basic reporting system. Teachers have access to more advanced reporting system that enable to them to monitor students progress and analyze the system efficiency. Data and information are stored in a database while the intelligent part, which is the practices, is stored in the domain model. In the following sections, we describe how web pages are created using ASP.NET then, talk about each component design and implementation in details.

5.2 Creating the Web pages using ASP.NET

To create any webpage two source files are created, the markup files (the .aspx files) which define the appearance of the webpage and the code-behind file which provides the code to be executed when the user submits the page (Lowe D. et al. 2006).

In the following, Figure 5.2 shows an example of the .aspx (Markup code) and Figure 5.3 shows an example of the code-behind file.

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ClipsOutPut.aspx.vb" Inherits="ClipsOutPut"
%>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms, Version=8.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Progress Report</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
              
        <rsweb:ReportViewer ID="ReportViewer1" runat="server" Font-Names="Verdana" Font-Size="8pt"
            Height="476px" Style="z-index: 102; left: 6px; position: absolute; top: 0px"
            Width="1205px">
            <LocalReport ReportPath="Report2.rdlc">
                <DataSources>
                    <rsweb:ReportDataSource DataSourceId="ObjectDataSource2"
Name="DataSet1_DataTable1" />
                    <rsweb:ReportDataSource DataSourceId="ObjectDataSource1"
Name="DataSet3_DataTable1" />
                </DataSources>
            </LocalReport>
        </rsweb:ReportViewer>
        <asp:ObjectDataSource ID="ObjectDataSource2" runat="server"
OldValuesParameterFormatString="original_{0}"
            SelectMethod="GetData"
TypeName="DataSet3TableAdapters.DataTable1TableAdapter"></asp:ObjectDataSource>
        <asp:ObjectDataSource ID="ObjectDataSource1" runat="server" SelectMethod="GetData"
            TypeName="DataSet3TableAdapters.DataTable1TableAdapter"
OldValuesParameterFormatString="original_{0}"></asp:ObjectDataSource>
            

    </div>
    </form>
</body>
</html>
```

Figure 5.2: An example of the markup code (.aspx)

```
Partial Class Moduleselection
    Inherits System.Web.UI.Page
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

        If IsPostBack = False Then

            If Not Request.QueryString("moduleid") Is Nothing Then
                Dim moduleid As Integer =
Integer.Parse(Request.QueryString("moduleid"))
                Session.Add("moduleID", moduleid)

                Response.Redirect("topic.aspx")
            End If

        End If

    End Sub
    Sub Page_PreInit(ByVal sender As Object, ByVal e As EventArgs) Handles
Me.PreInit

        If Profile.IsAnonymous = False Then
            If Profile.Theme <> "" Then
                Page.Theme = Profile.Theme
            End If

        Else
            Response.Redirect("default.aspx")
        End If
    End Sub

End Class
```

Figure 5.3: An example of the code-behind file (in Visual Basic .NET)

5.2.1: The Main Page (Default Page)

The front page, which is the system main page, shows the anonymous login view at the beginning as in Figure 5.4. When the user enters the log in data to the system, the system checks the given username and password with its database. If the system acknowledges the given data, it directs the user to the appropriate logged in login view as explained in Figure 5.5. Otherwise, the system asks the user to re-enter the login data. As mentioned earlier in this chapter, the system consists of two interfaces (logged in views): the teachers interface and the students interface.

Figure 5.4: Log in web page

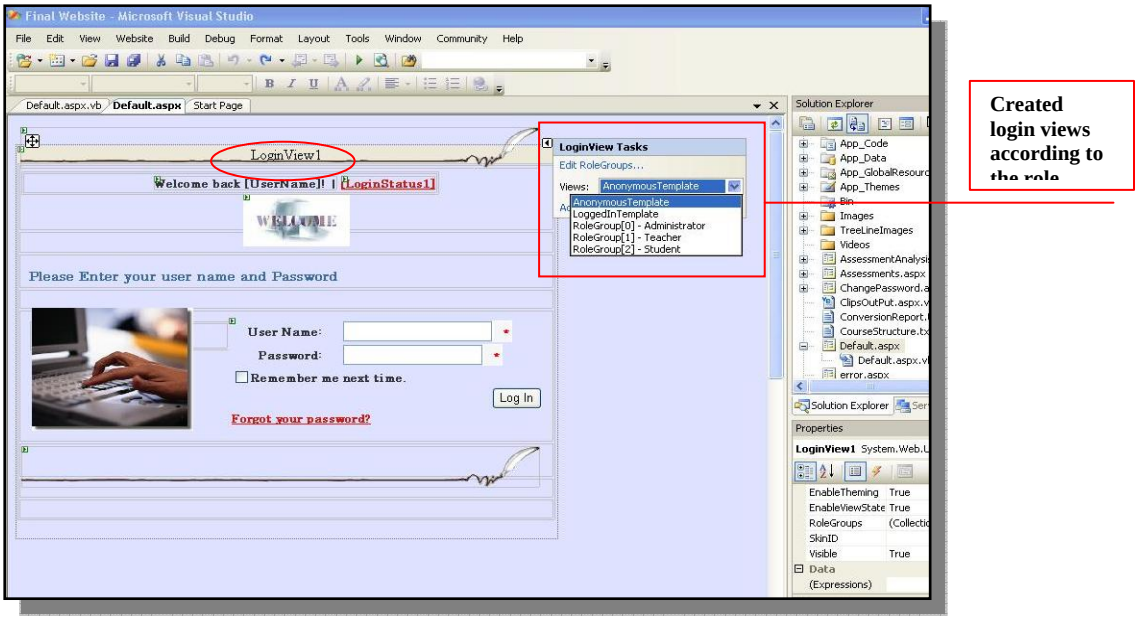


Figure 5.5: The available login views

To create the main page, two files were created as explained earlier: the markup file and the code-behind file. We employed Visual Basic .NET as our programming language in the code behind file. Figures 5.6, 5.7 and 5.8 show examples of the markup code, the design view and code-behind file for the Main Page Anonymous view. Then we will discuss the students and teachers interfaces in more details.

```
<AnonymousTemplate>
                <table width="560" align="center">
                    <tr>
                        <td style="width: 100%; height: 70px;">
                            <asp:Image ID="imgWelcome" runat="server" SkinID="welcome" style="z-
index: 100; left: 245px; position: absolute; top: 87px" Width="111px" />
                             
                        </td>
                    </tr>
                    <tr>
                        <td style="width: 100%; height: 21px;"> </td>
                    </tr>
                    <tr>
                        <td class="h1" style="width: 100%">
                            Please Enter your user name and Password</td>
                    </tr>
                    <tr>
                        <td style="width: 100%; height: 18px;"></td>
                    </tr>
                    <tr>
                        <td style="width: 100%; height: 143px;">
                             
                            <img id="IMG1" src="Images/computers-basics.jpg" style="z-index: 100;
left: 25px;
                                width: 168px; position: absolute; top: 213px; height: 136px"
language="javascript" onclick="return IMG1_onclick()" />
                            <asp:Image ID="imgLine1" runat="server" SkinID="line" style="z-index:
100; left: 13px; position: absolute; top: 13px" />
                              
                            <table style="z-index: 99; left: 23px; position: absolute; top: 228px">
                                <tr>
                                    <td style="width: 202px; height: 26px">
                                    </td>
                                    <td style="width: 100px; height: 26px">
                            <asp:Login ID="Login1" runat="server" TitleText=""
PasswordRecoveryText="Forgot your password?" PasswordRecoveryUrl="~/recoverpassword.aspx"
Width="327px" style="z-index: 100; left: 207px; position: absolute; top: -4px" >
                                <FailureTextStyle CssClass="incorrect" />
                                <CheckBoxStyle CssClass="login" />
                                <ValidatorTextStyle CssClass="incorrect" />
```

Figure 5.6: a sample of the markup code (.aspx) used for the anonymous interface (view)

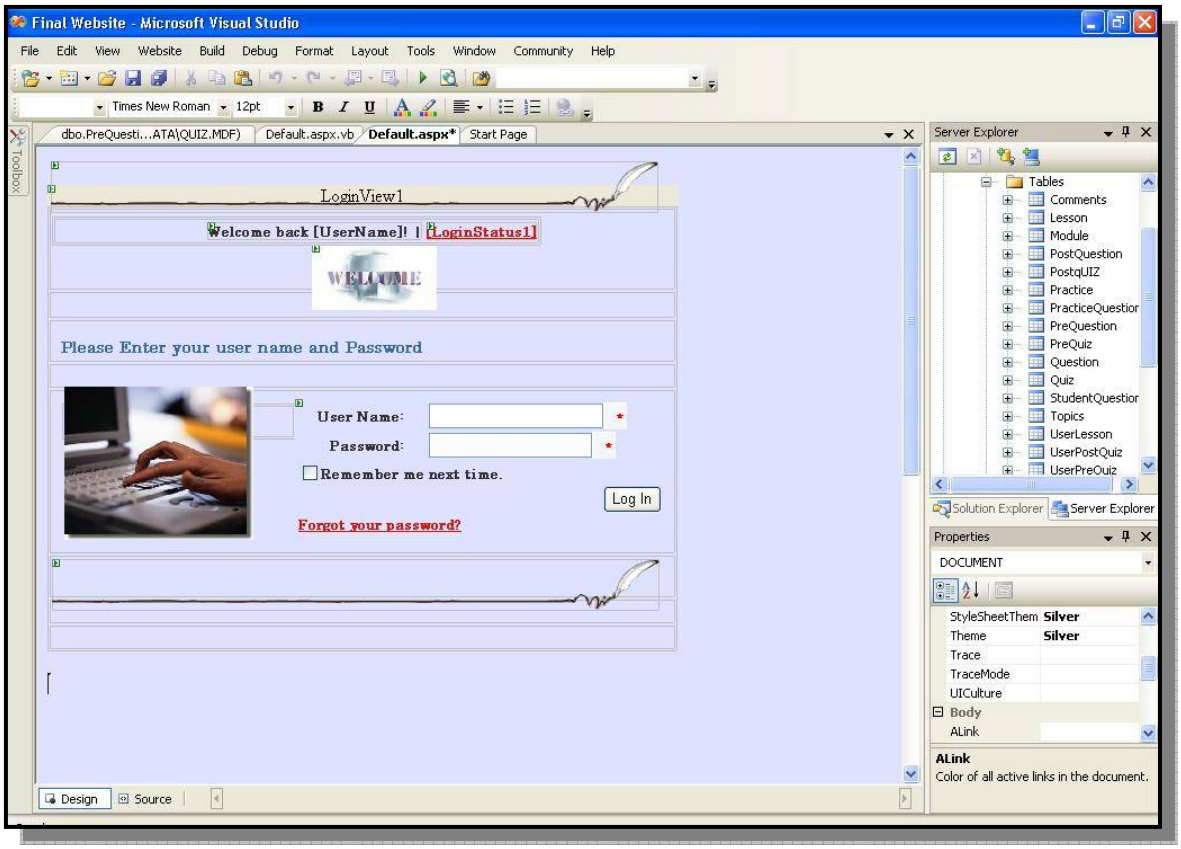


Figure 5.7: a sample of the design interface (.aspx) used to create the anonymous view

```
Partial Class _Default
   Inherits System.Web.UI.Page

   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

      If Profile.IsAnonymous = False Then
         Session.Add("UserName", User.Identity.Name)
      End If

   End Sub

   Sub Page_PreInit(ByVal sender As Object, ByVal e As EventArgs) Handles Me.PreInit

      If IsNothing(Request.QueryString("theme")) = False Then
         Profile.Theme = Request.QueryString("theme")
      End If

      If Profile.IsAnonymous = False Then
         If Profile.Theme <> "" Then
            Page.Theme = Profile.Theme
         End If
      End If
```

Figure 5.8: a sample of code behind file (in VB .NET) used to create the anonymous view

5.2.1.1 The Student Interface:

As illustrated in Figure 5.9, in the student interface, students are able to view and select from the available list of lessons. In the lessons, objectives link explains to the student what to expect in this lesson (chapter). Each lesson contains some information, and the necessary links to the associated practices. Practices are some intelligent exercises where students answer multiple choice questions and get the necessary feedback and accordingly a sequence of questions. Students can also perform all lesson related quizzes and do the end of chapter assessment. Furthermore, students can track which lessons they have visited, and view quizzes results and feedback. Moreover, students can add and view comments and feedback sent to teachers by him/her and colleagues. Also, they can ask and view questions sent to and answered by teachers. Another system feature is that students can change their password at anytime.

Figure 5.9 Student Main page

This interface is created as a view from the default page (main page) as in Figure 5.10. This view is associated with a markup code as shown in Figure 5.11

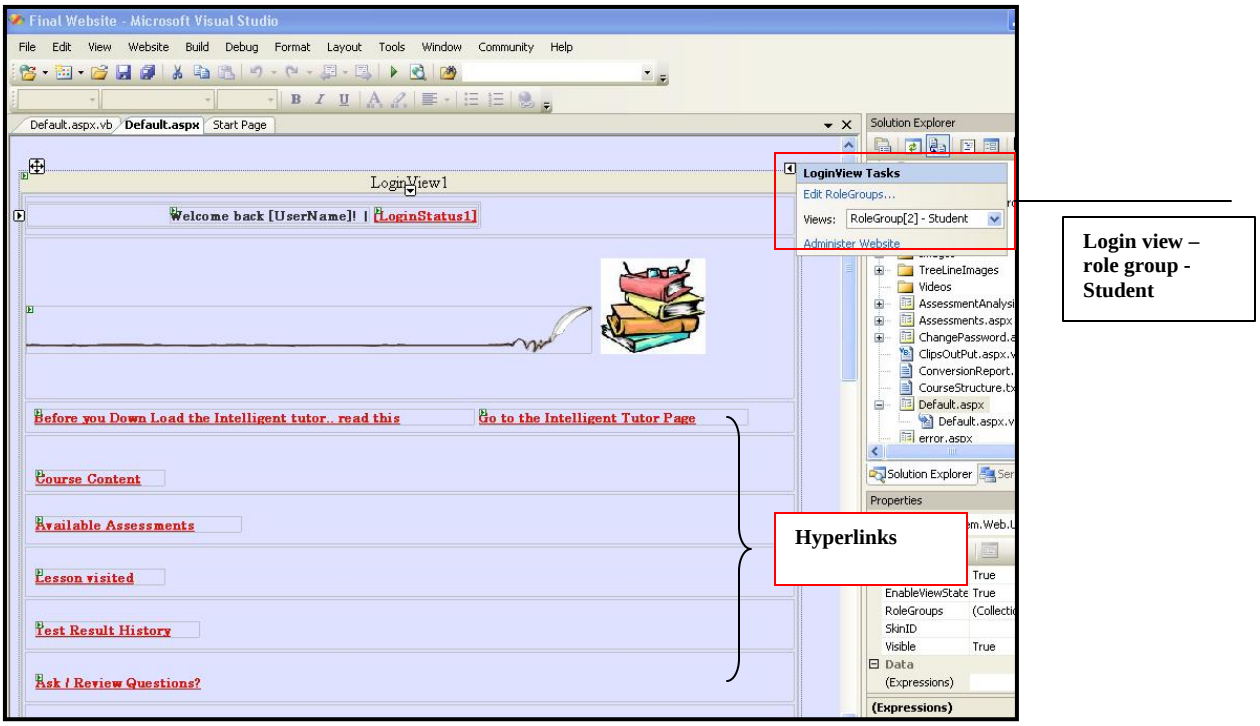


Figure 5.10: student role group design view

```
<asp:RoleGroup Roles="Student">
                    <ContentTemplate>
                        <table style="width: 742px; text-align: left; border-right: silver thin solid;
border-top: silver thin solid; border-left: silver thin solid; border-bottom: silver thin solid;"
id="TABLE1" language="javascript" onclick="return TABLE1_onclick()">
                            <tr>
                                <td class="h1" style="width: 825px; height: 1px">
                                     
                                </td>
                            </tr>
                            <tr>
                                <td style="width: 825px; height: 71px;">
                                    <br />
                                    <asp:Image ID="imgMainMenu" SkinID="line" runat="server" />
                                     <img id="IMG2" src="Images/books2.jpg" style="width: 100px;
height: 91px;" language="javascript" onclick="return IMG2_onclick()" />
                                     <br />
                                    <br />
                                </td>
                            </tr>
                            <tr>
                                <td style="width: 825px; height: 28px">
                                     
                                    <asp:HyperLink ID="HyperLink10" runat="server"
NavigateUrl="~/Readme.txt" Width="420px">Before you Down Load the Intelligent tutor.. read
this</asp:HyperLink>
                                    <asp:HyperLink ID="HyperLink9" runat="server" Height="16px"
NavigateUrl="~/IntelligentPractices.aspx"
                                        Width="257px">Go to  the Intelligent Tutor
Page</asp:HyperLink></td>
                            </tr>
```

Figure 5.11: Sample of the markup code (.aspx) for the student interface.

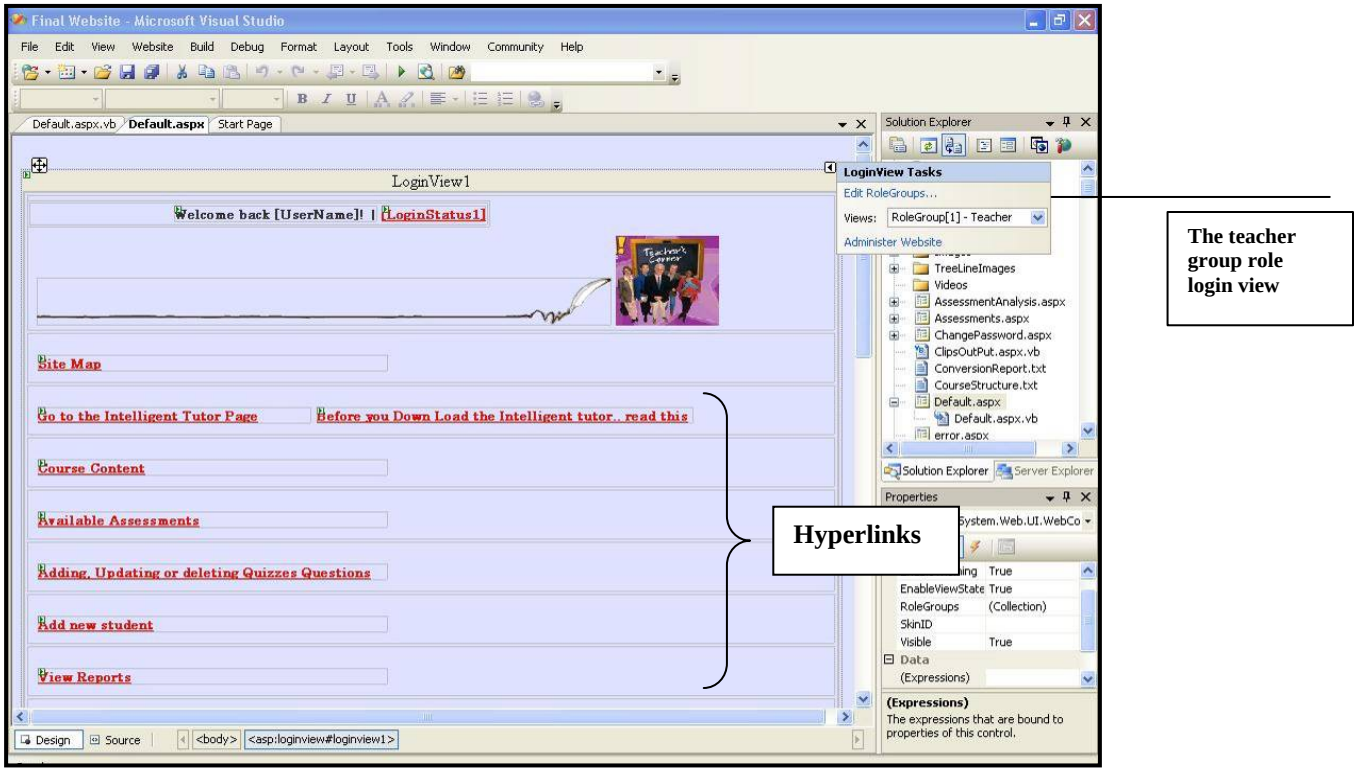
5.2.1.2 The Teacher Interface:

Besides everything that is available for students, teachers can also answer students' inquiries. Additionally, teachers can track all students' progress in lessons and the different quizzes and assessment. This is besides that teachers have the ability to generate more detailed reports from data stored in the database. Furthermore, teachers are given the necessary permission to modify the question database to design a variety of up-to-date quizzes and assessment. Another privilege teachers have is to add/enroll new students to the system.

The teacher main page is shown in Figure 5.12

Figure 5.12: teachers' main page

The teacher interface is also created as a login view in the main page as shown in figure 5.13. Figure 5.14 shows a sample of the markup code for the teacher group role template (.aspx).




```
<asp:RoleGroup Roles="Teacher">
                    <ContentTemplate>
                        <table align="center" style="width: 767px; height: 353px; border-right:
silver thin solid; border-top: silver thin solid; border-left: silver thin solid; border-bottom:
silver thin solid;">
                            <tr>
                                <td style="width: 798px; height: 28px">
                                    <br />
                                    <br />
                                     
                                    <img src="App_Themes/Silver/images/line3.gif" />
                                    <img src="Images/teacher1.jpg" style="width: 97px; height: 86px"
id="IMG3" language="javascript" onclick="return IMG3_onclick()" /></td>
                            </tr>
                            <tr>
                                <td class="h1" style="width: 798px; height: 28px">
                                    <asp:HyperLink ID="HyperLink11" runat="server" Height="16px"
NavigateUrl="~/SiteMap.aspx"
                                        Width="329px">Site Map</asp:HyperLink></td>
                            </tr>
                            <tr>
                                <td class="h1" style="width: 798px; height: 28px">
                                    <asp:HyperLink ID="HyperLink9" runat="server"
NavigateUrl="~/IntelligentPractices.aspx" Width="257px" Height="16px">Go to  the Intelligent Tutor
Page</asp:HyperLink>
```

Figure 5.14: sample of the markup code (.aspx) for the teachers login view

5.2.2 The Lessons

It is important to consider the educational and pedagogical point of view when we design the interface for an educational website. A well-designed educational website must appear attractive, simple, descriptive, informative and organized; setting the environment for learning new ideas (Jeffrey 2001; Krug and Steve 2001). Important considerations were taken while designing and building the website interface. One important consideration was to design a simple lesson layout web pages template that does not contain too much information in one page (information is stored in separate web pages). Another point that was taken into consideration was avoiding eye strain and distraction by selecting eye-relaxing colors to help student concentrate while reading the lessons or doing the quizzes. Furthermore, making navigation easy between pages and lessons was another important point to consider. Lessons were divided into different pages with next and previous buttons to avoid too much content in one page and to minimize scrolling up and down. In addition, clear instruction and consistency between pages were important points that were sought of and considered during the design phase. Figure 5.15 shows some examples of different lessons.

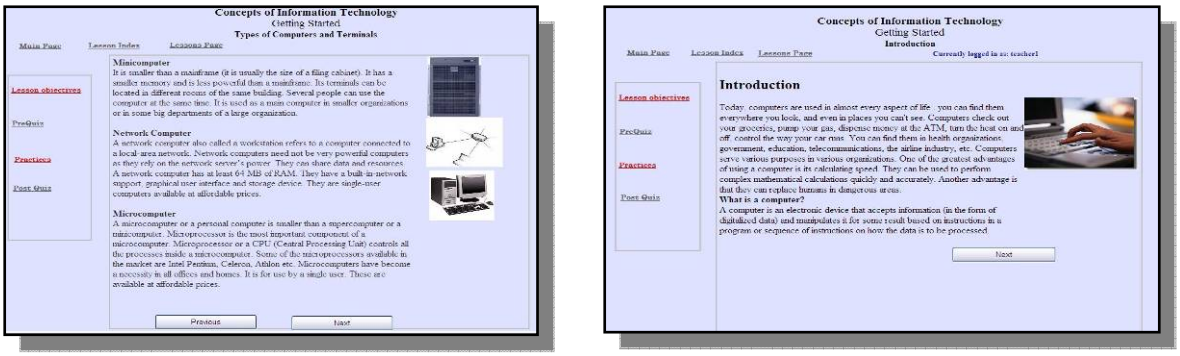


Figure 5.15 examples of different lessons

All lessons were designed to meet the ICDL/ECDL Module1 certification syllabus. Lessons are presented in a logical hierarchy in a learner control environment where students can directly go to any desired lesson or follow the structure. This is to serve different learners needs. Furthermore, lessons were designed to meet different

educational theories as discussed in Chapter 3. Lessons can be used by independent learners where students can use the system at their own pace like in distance learning or can be used in classroom as an e-learning tool in a blended learning approach as a reference material for students to get back to. Lessons include images where applicable and follow a simple layout where only important links are available. These links are either lesson related or navigation related. Lesson related links include: the lesson objectives link which is for students to understand what to expect in a specific chapter, “Pre-Quizzes” (per chapter) to evaluate previous knowledge and for students to get an idea about the lesson content, “Post-Quizzes” to assess students’ progress and to evaluate their readiness for the exam, and “Practices” which is the intelligent part of the system that help students assess their understanding of the lesson, reveal the knowledge gaps, correct misconceptions and assist students acquire the needed knowledge for the exam. Additionally, the navigation links include: the “Main Page” link which gives students the flexibility to go back to the main page anytime, “Lessons Index” where students can go and select between different lessons and chapters. To move between lessons in the same chapter, students are provided with two command buttons: “Previous“ and “Next“ buttons as shown in Figure 5.16

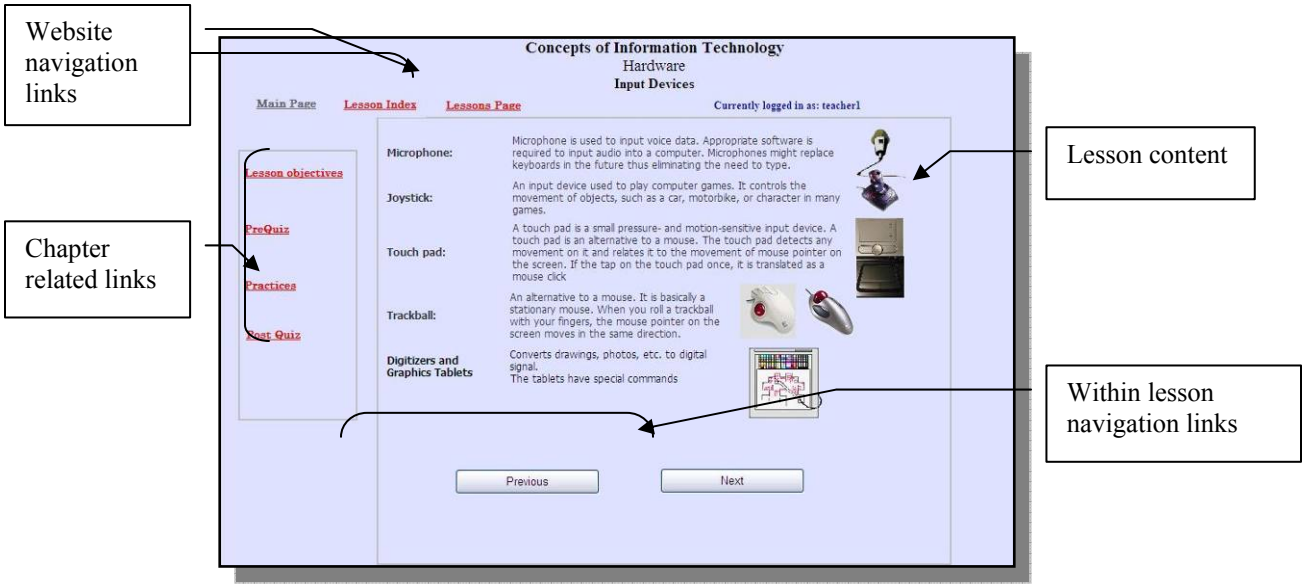


Figure 5.16: Sample lesson

In our proposed system we created 40 main lessons while many of these lessons contain a different number of sub-lessons. Every lesson and sub-lesson is created as a separate webpage consists of the markup .aspx code file and the code-behind VB.NET file like the examples in Figures 5.17 and 5.18. The mark up file contains the design information and the VB connects the current lesson with other lessons.

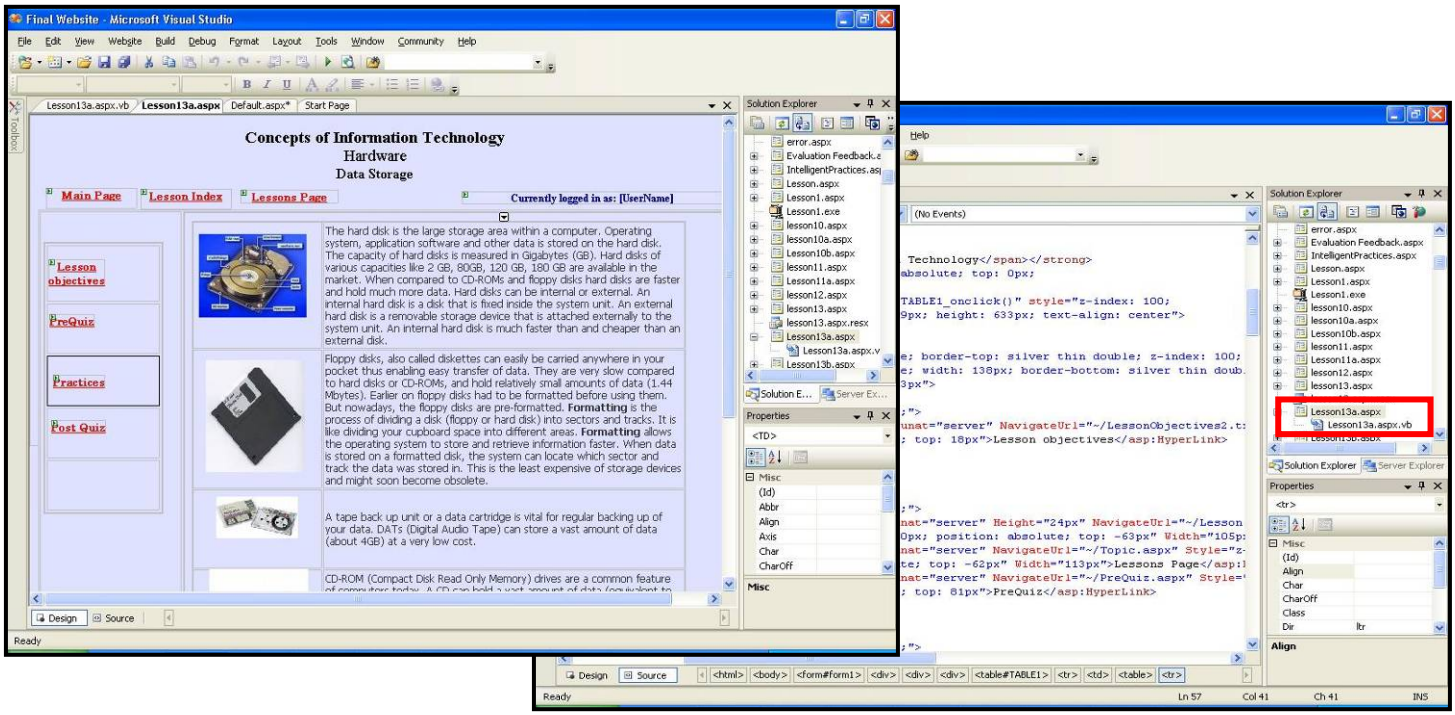


Figure 5.17: A sample of a lesson webpage in the design and markup code. Aspx modes

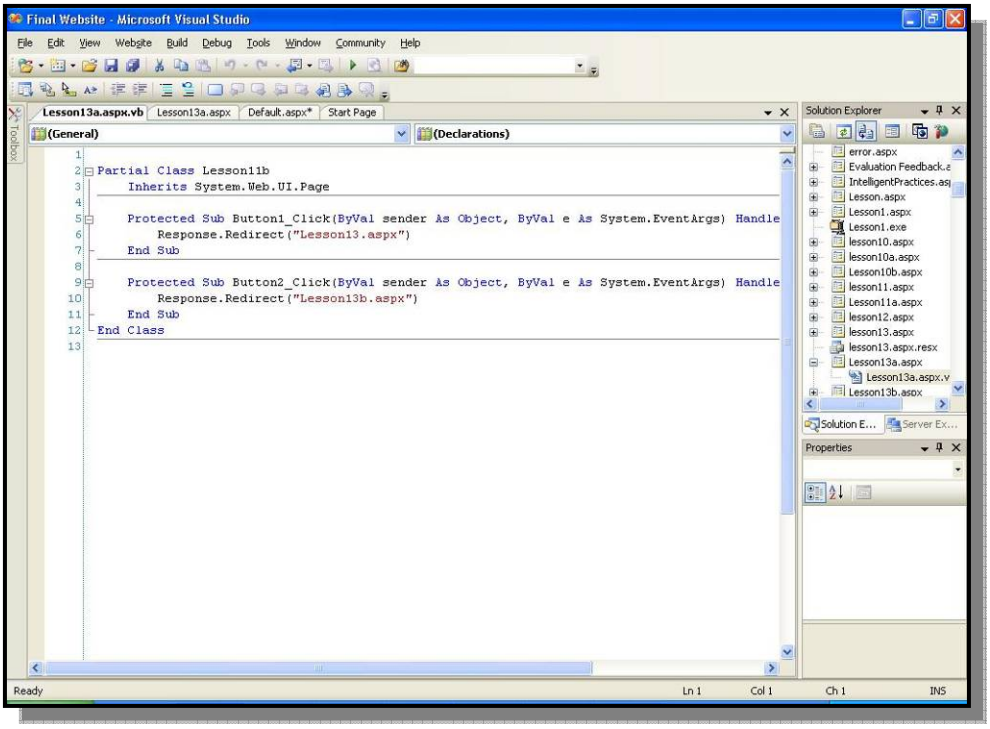


Figure 5.18: An example of the visual Basic code of a lesson

5.3 Designing and constructing the database

To make the system flexible, easily modified and can be expanded to meet future needs, many features of the system were designed in a database driven approach. The database contains the users' information relations and the tutoring system relations. This database is used in the system by many components such as the reporting system, communication tools and evaluation tools as will be discussed later in this chapter.

5.3.1 Identifying the relations

Building a relational database was essential for the system in order to store, access, and manipulate data. List of relations to fulfill the system needs has been identified as shown in Figure 5.19

Relation	Description
Module(**ModuleID, ModuleTitle, ModuleDescription)**	**This table contains all available modules. For now only Module 1 is available but the system can be easily be expanded to include all seven modules.**
Topic**(ModuleID, TopicID, Topictitle, TopicDescription)**	**Topics here mean chapters. Module 1 contains 9 full chapters.**
Lesson**(TopicID, LessonID, LessonTitle, LessonDescription)**	**Lessons are the content lessons. Each lesson has a number to be identified with.**
UserLesson**(UserLessonID, LessonID, DateTimeVisited, Username)**	**UserLesson is used to track visited lessons**
Quiz**(QuizID, ModuleID, Title, Description)**	**Quiz refers to the available assessment. The system accepts as many assessments as needed.**
UserQuiz**(UserQuizID, QuizID,DatetimeComplete,Score, Username)**	**UserQuiz is used to track Assessment (who has done them and with some details about the score and time)**
Question**(QuestionID, ModuleID, Title, Answer1, Answer2, Answer3, Answer4, CorrectAnswer, AnswerExplanation, QuestionOrder, QuizID)**	**In this relation Assessment questions and answers are listed with the correct answer. Assessment are done per module. Number of questions if flexile and the system accept as many questions as needed per assessment.**
PreQuiz**(PreQuizID, TopicID, Title, Description)**	**This is a table containing information about pre-quizes for all lessons. Number of PreQuizes can be extended as needed. Pre-quizes are designed for each chapter (number of lessons). We have 9 ready**

	Pre-quizzes ready with the system
UserPreQuiz**(UserPreQuizID, PreQuizID,DatetimeComplete,Score, Username)**	**UserPreQuiz is used to track Pre-Quizzes (who has done them and with some details about the score and time)**
PreQuestion**(QuestionID, ModuleID, Title, Answer1, Answer2, Answer3, Answer4, CorrectAnswer, AnswerExplanation, QuestionOrder, QuizID)**	**In this relation Pre-Quizzes questions and answers are listed with the correct answer. Assessment are done per module. Number of questions if flexile and the system accept as many questions as needed per assessment.**
PostQuiz**(PostQuizID, TopicID, Title, Description)**	**This is a table containing information about Post-quizes for all lessons. Number of PostQuizes can be extended as needed. Post-quizes are designed for each chapter (number of lessons). We have 9 ready Post-quizzes ready with the system**
UserPost**Quiz(UserPostQuizID, PostQuizID,DatetimeComplete,Score, Username)**	**UserPostQuiz is used to track Post-Quizzes (who has done them and with some details about the score and time)**
PostQuestion**(QuestionID, ModuleID, Title, Answer1, Answer2, Answer3, Answer4, CorrectAnswer, AnswerExplanation, QuestionOrder, QuizID)**	**In this relation Post-Quizzes questions and answers are listed with the correct answer. Assessment are done per module. Number of questions if flexile and the system accept as many questions as needed per assessment.**
Comments**(CommentID, Name, Email, Comment)**	**This relation is one mean of Electronic communication where students can send their feedback and comment about the system overall and how to improve it.**
StudentQuestion**(StudentQuestionID,StudentName, StudentEmail, StudentQuestion, TeacherAnswer)**	**Another means of Electronic communication is the students questions feature which gives students the opportunity to post questions and seek teachers help when needed. Teachers can read these questions and answer them in return.**

Figure 5.19 Samples of the database relations/tables

There are additional relations such as those concerned with users' information, passwords, roles and other personal information including emails.

5.3.2 Building the tables

As explained before all necessary relations were identified during the design time and Microsoft SQL 2005 server was selected to create these relations and build all required queries as in the following examples.

Figure 5.20 is an example that presents the available pre-chapters quiz table definition and table content. Another example is in Figure 5.21 which presents the Pre-Quizzes questions table definition and table content.

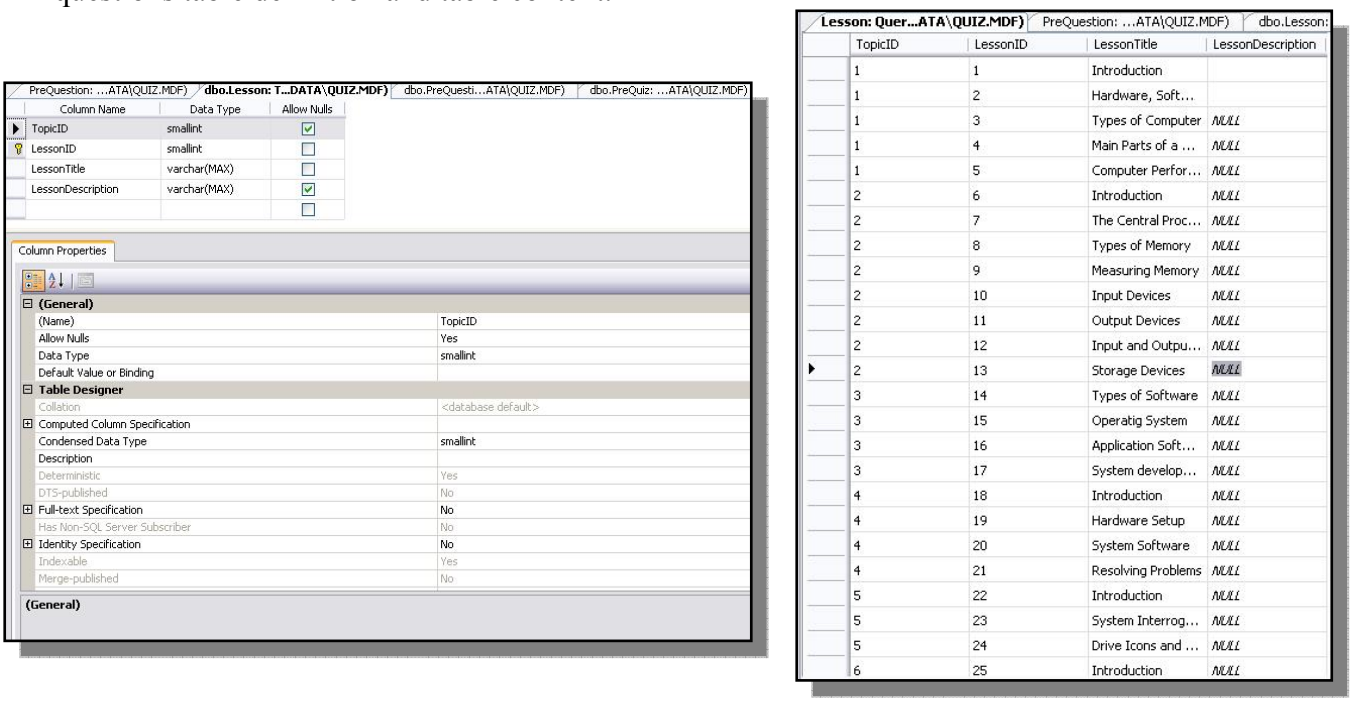
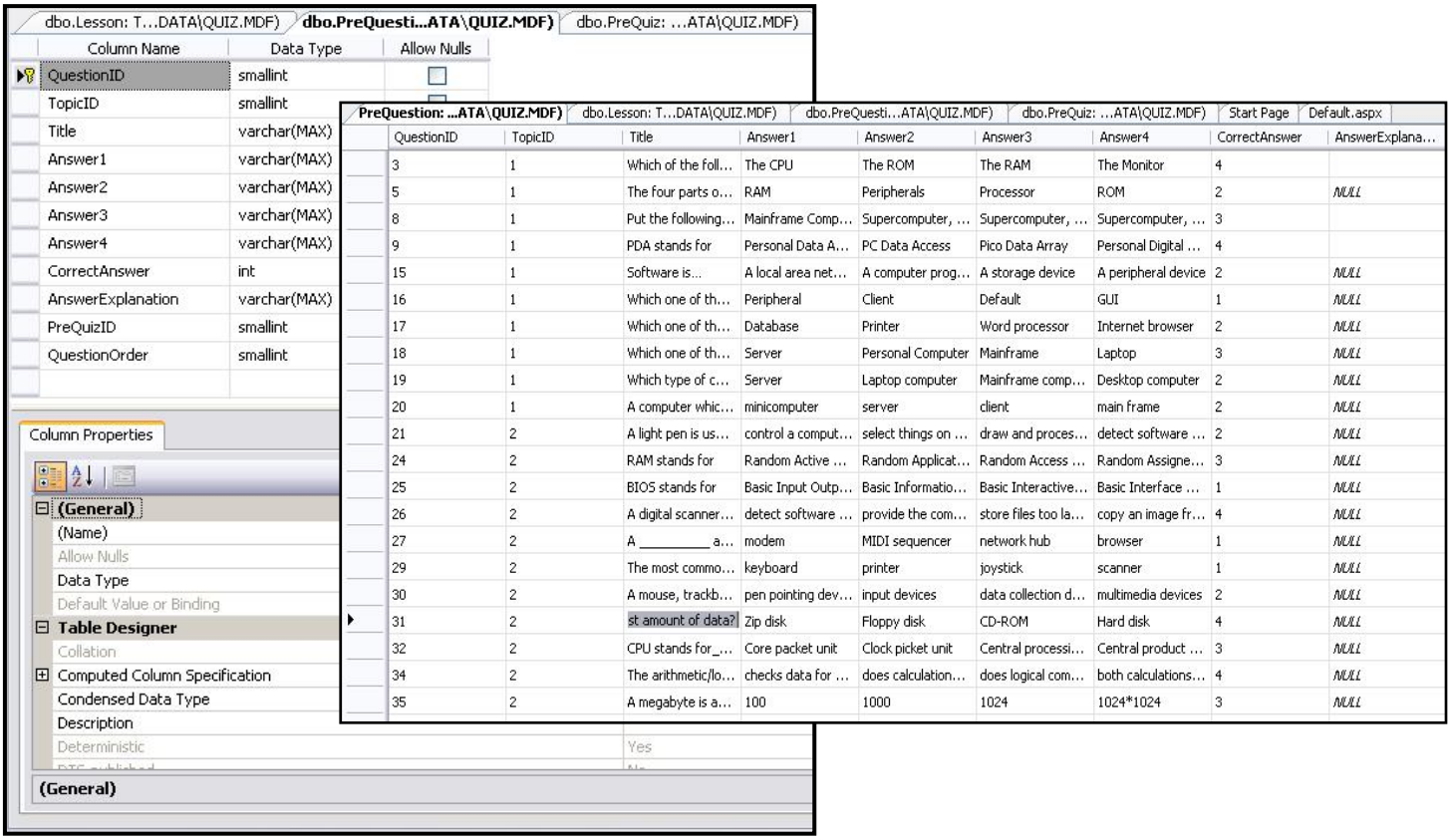


Figure 5.20: An example of the available pre-Quizzes table and its content



dbo.Lesson: T...DATA\QUIZ.MDF) | dbo.PreQuesti...ATA\QUIZ.MDF) | dbo.PreQuiz: ...ATA\QUIZ.MDF)

Column Name	Data Type	Allow Nulls
QuestionID	smallint	
TopicID	smallint	
Title	varchar(MAX)	
Answer1	varchar(MAX)	
Answer2	varchar(MAX)	
Answer3	varchar(MAX)	
Answer4	varchar(MAX)	
CorrectAnswer	int	
AnswerExplanation	varchar(MAX)	
PreQuizID	smallint	
QuestionOrder	smallint	

Column Properties

(General)
(Name)
Allow Nulls
Data Type
Default Value or Binding
Table Designer
Collation
Computed Column Specification
Condensed Data Type
Description
Deterministic — Yes

(General)

PreQuestion: ...ATA\QUIZ.MDF) | dbo.Lesson: T...DATA\QUIZ.MDF) | dbo.PreQuesti...ATA\QUIZ.MDF) | dbo.PreQuiz: ...ATA\QUIZ.MDF) | Start Page | Default.aspx

QuestionID	TopicID	Title	Answer1	Answer2	Answer3	Answer4	CorrectAnswer	AnswerExplana...
3	1	Which of the foll...	The CPU	The ROM	The RAM	The Monitor	4	
5	1	The four parts o...	RAM	Peripherals	Processor	ROM	2	NULL
8	1	Put the following...	Mainframe Comp...	Supercomputer, ...	Supercomputer, ...	Supercomputer, ...	3	
9	1	PDA stands for	Personal Data A...	PC Data Access	Pico Data Array	Personal Digital ...	4	
15	1	Software is...	A local area net...	A computer prog...	A storage device	A peripheral device	2	NULL
16	1	Which one of th...	Peripheral	Client	Default	GUI	1	NULL
17	1	Which one of th...	Database	Printer	Word processor	Internet browser	2	NULL
18	1	Which one of th...	Server	Personal Computer	Mainframe	Laptop	3	NULL
19	1	Which type of c...	Server	Laptop computer	Mainframe comp...	Desktop computer	2	NULL
20	1	A computer whic...	minicomputer	server	client	main frame	2	NULL
21	2	A light pen is us...	control a comput...	select things on ...	draw and proces...	detect software ...	2	NULL
24	2	RAM stands for	Random Active ...	Random Applicat...	Random Access ...	Random Assigne...	3	NULL
25	2	BIOS stands for	Basic Input Outp...	Basic Informatio...	Basic Interactive...	Basic Interface ...	1	NULL
26	2	A digital scanner...	detect software ...	provide the com...	store files too la...	copy an image fr...	4	NULL
27	2	A _____ a...	modem	MIDI sequencer	network hub	browser	1	NULL
29	2	The most commo...	keyboard	printer	joystick	scanner	1	NULL
30	2	A mouse, trackb...	pen pointing dev...	input devices	data collection d...	multimedia devices	2	NULL
31	2	st amount of data?	Zip disk	Floppy disk	CD-ROM	Hard disk	4	NULL
32	2	CPU stands for_...	Core packet unit	Clock picket unit	Central processi...	Central product ...	3	NULL
34	2	The arithmetic/lo...	checks data for ...	does calculation...	does logical com...	both calculations...	4	NULL
35	2	A megabyte is a...	100	1000	1024	1024*1024	3	NULL

Figure 5.21: An example of the available pre-Quizzes questions table and its content

5.3.3 Relating tables:

Every created table has a primary key and in some cases a secondary key. For effective use of the database, tables were related using mainly the primary key.

Figure 5.22 shows the database diagram. A better view of the system tables/relations diagram is in Appendix B.

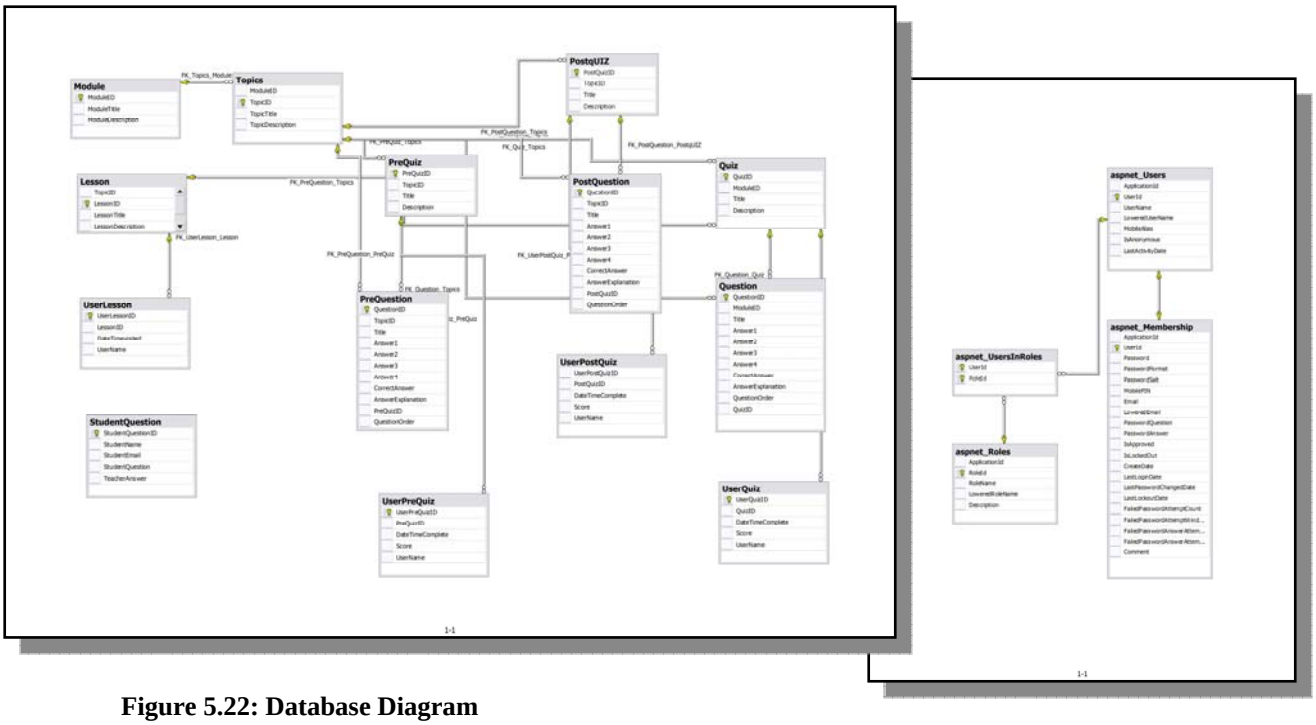


Figure 5.22: Database Diagram

An example of different tables relationships is the lesson selection mechanism. At first, the system makes a list of modules available for users to select from. This list is taken from the modules table. According to the module selection (Module ID), a list of chapters (in the database we call them topics) is shown to the user to select from. Each chapter or topics as named in the database, has a unique ID number and a list of lessons related to this number. With reference to the selected topic ID, a list of lessons will be available for users to select from. Each lesson in turn has its own ID too. Pre- and post-chapter quizzes are called in each chapter according to the correspondence topic ID. This example explains that, depending on the module number, a list of topics are displayed to the user and in turn, according to the selected topic (topic number) a list of lessons is displayed for users to select from. Figure 5.23 illistrates how modules, topics and lessons are related in the database. Next, Figure 5.24 explains the hirarachy of modules, chapters

(topics) and lessons and how are topics related to the module number and the lessons are displayed according to the topic number. Then, Figure 5.25 shows screen shots of how the displayed lists of lessons may differ depending on the selection.

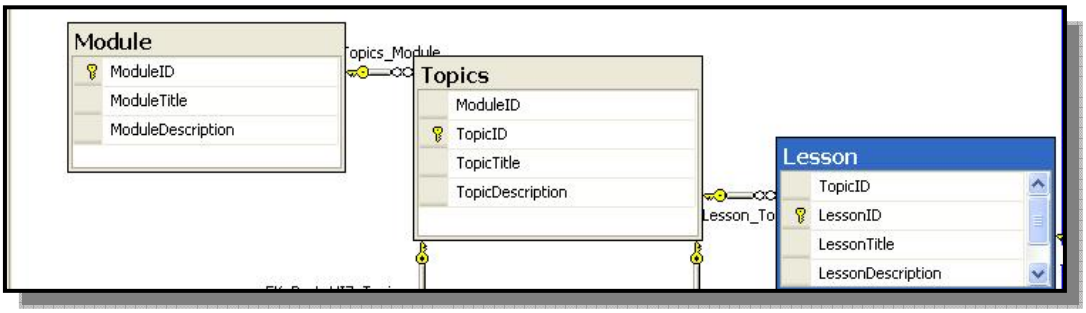


Figure 5.23: An example of some table relationships

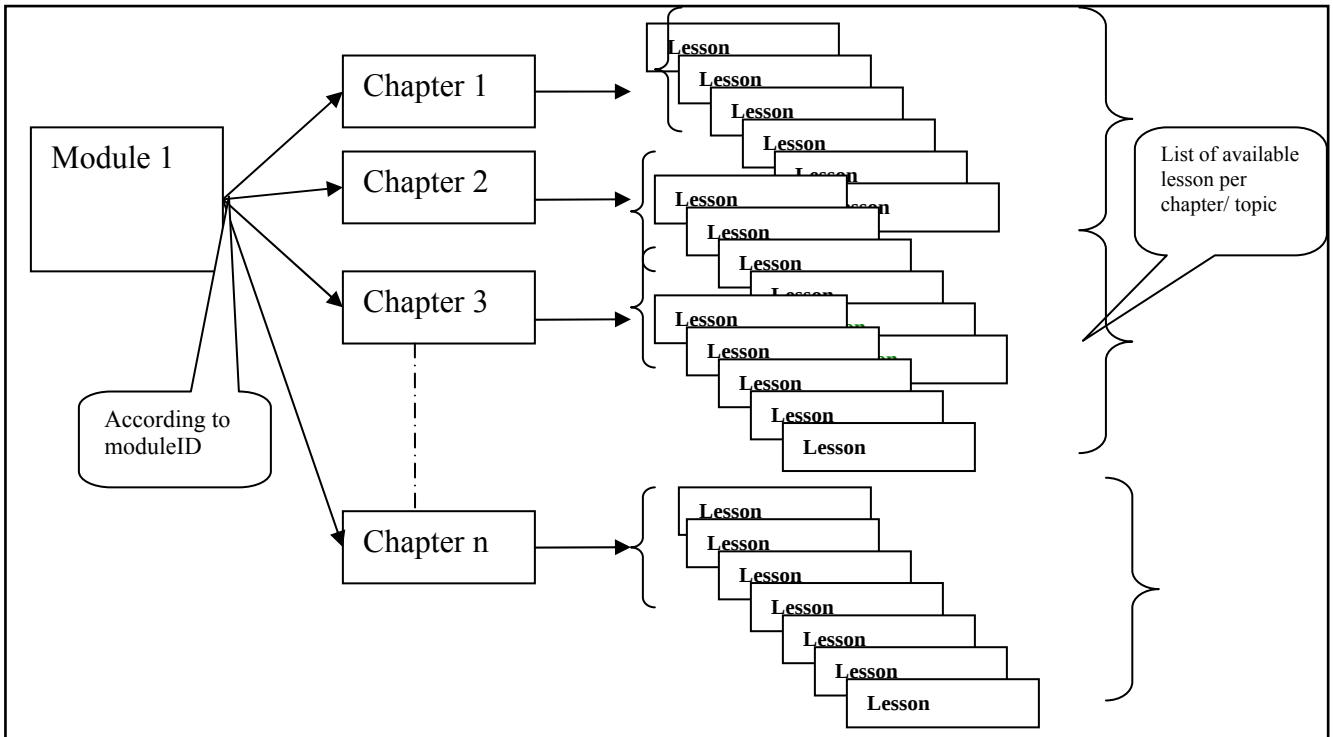


Figure 5.24: Modules, chapters (Topics) , lessons hierarchy and dependency

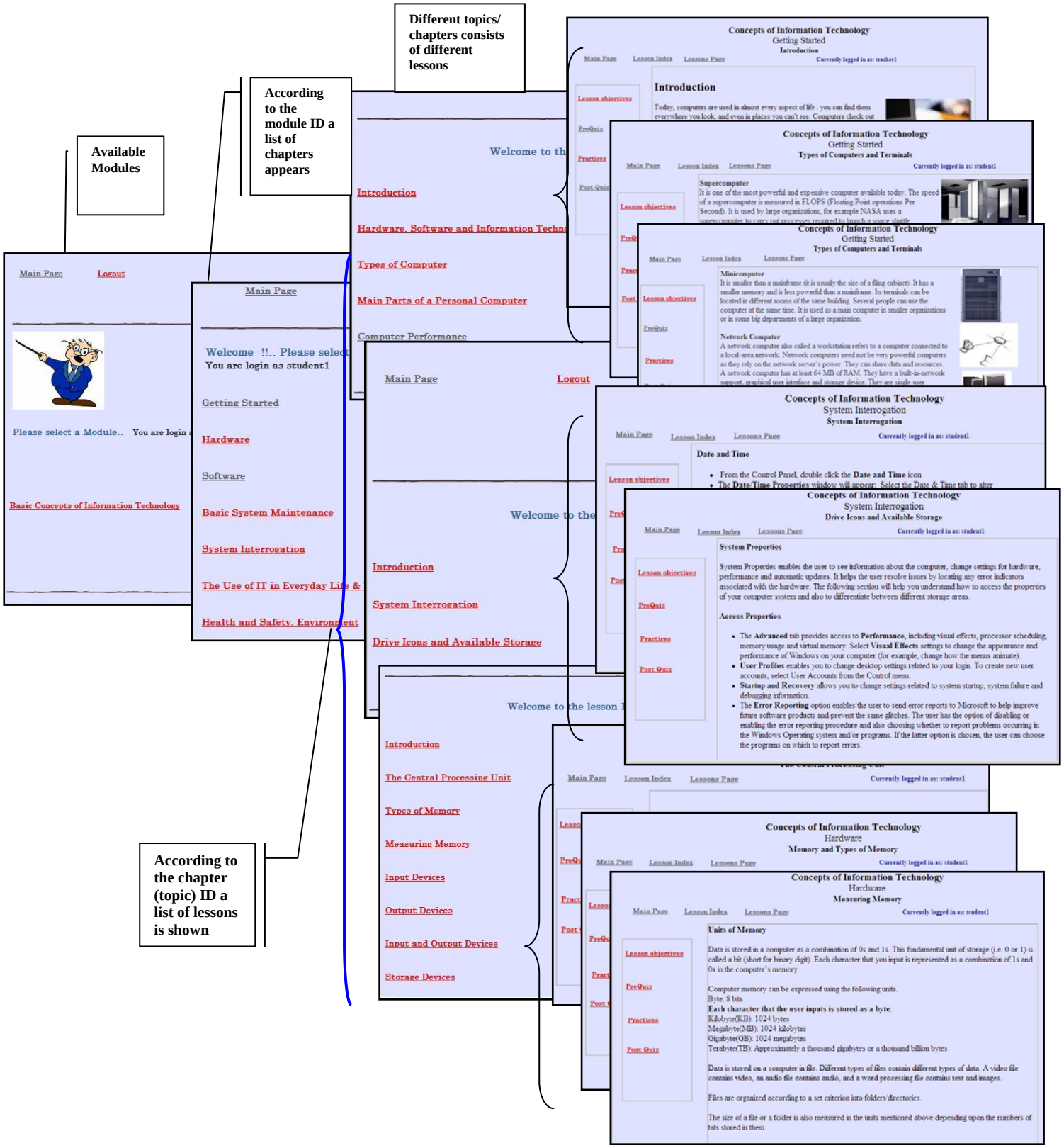


Figure 5.25: Example screen shots of different selections of topics/ chapters and lessons

Other relations will be explained later in this chapter when we talk about the reporting system, different quizzes and the communication tools.

5.3.4 Building Queries

A number of queries were built to retrieve information from one table or more according to specific criteria or to undertake some specific action like calculation or sorting. We used SQL as our query language as explained in Chapter 4. Then embedded the query with the .aspx code. To do this we used some tools from Microsoft visual studio 2005 such as the GridView, DataLists, DetailsView, FormView and the Repeater tools besides the sqlDataSource tool.

Figure 5.26 is an example of a query used to retrieve information and built the students progress report as will be explained in more details later in this chapter. Following is Figure 5.27 which illustrates how a query is embedded within the .aspx markup code.

```
SELECT UserPreQuiz.PreQuizID, UserPreQuiz.DateTimeComplete,
UserPreQuiz.Score, UserPreQuiz.UserName, UserPostQuiz.UserPostQuizID,
UserPostQuiz.PostQuizID, UserPostQuiz.DateTimeComplete AS Expr1,
UserPostQuiz.Score AS Expr2, UserPostQuiz.Score - UserPreQuiz.Score AS
Progress, PreQuiz.TopicID FROM UserPostQuiz INNER JOIN UserPreQuiz ON
UserPostQuiz.UserName = UserPreQuiz.UserName INNER JOIN PreQuiz ON
UserPreQuiz.PreQuizID = PreQuiz.PreQuizID INNER JOIN PostqUIZ ON
UserPostQuiz.PostQuizID = PostqUIZ.PostQuizID AND PreQuiz.TopicID =
PostqUIZ.TopicID
```

Figure 5.26: This query is used to calculate students' progress

```
</div>
        <asp:ObjectDataSource ID="ObjectDataSource1" runat="server" SelectMethod="GetData"
            TypeName="DataSet1TableAdapters.DataTable1TableAdapter"></asp:ObjectDataSource>
        <img src="Images/report2.jpg" />
          
        <br />
        <asp:HyperLink ID="HyperLink5" runat="server" Height="10px"
NavigateUrl="~/Default.aspx" Width="105px">Main Page</asp:HyperLink> 
        <asp:HyperLink ID="HyperLink2" runat="server" NavigateUrl="~/Reports.aspx" Style="z-
index: 100;
            left: -12px; top: 2px">Reports Page</asp:HyperLink><br />
        <br />
        <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/studentsProgressReport.aspx">Report view ( export to excel or pdf
?)</asp:HyperLink><br />
        <br />
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
            DataSourceID="SqlDataSource1" Style="color: black; background-color: gray;"
AllowPaging="True" AllowSorting="True" Height="527px">
            <Columns>
                <asp:BoundField DataField="UserName" HeaderText="UserName"
SortExpression="UserName" />
                <asp:BoundField DataField="PreQuizID" HeaderText="PreQuizID"
SortExpression="PreQuizID" />
                <asp:BoundField DataField="TopicID" HeaderText="TopicID"
SortExpression="TopicID" />
                <asp:BoundField DataField="Score" HeaderText="pre quiz Score"
SortExpression="Score" >
                    <ItemStyle BackColor="#E0E0E0" Font-Bold="True" ForeColor="Navy" />
                </asp:BoundField>
                <asp:BoundField DataField="PostQuizID" HeaderText="PostQuizID"
SortExpression="PostQuizID" />
                <asp:BoundField DataField="Expr2" HeaderText="Post Quiz Score"
SortExpression="Expr2" >
                    <ItemStyle Font-Bold="True" ForeColor="Navy" BackColor="#E0E0E0" />
                </asp:BoundField>
                <asp:BoundField DataField="Progress" HeaderText="Progress"
```

The SQL query

Figure 5.27 illustrate an embedded query within the markup code.

5.4 The Reporting system

An important system feature is the ability to generate different reports for the students and the teachers to track students' progress. Some reports provide an analysis as a feedback and evaluation method for teachers to continuously enhance the system functionality and teaching methodology. In this section and subsections we explain the available different reports in details.

Figure 5.28 is a snapshot of the reports page in the teacher interface

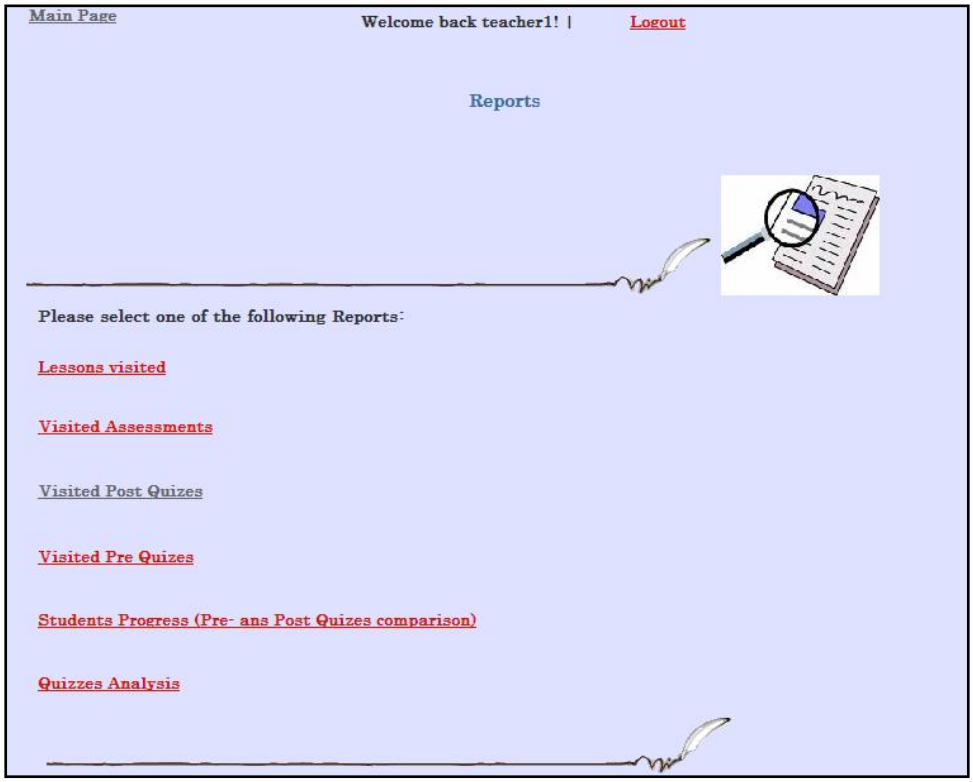


Figure 5.28: different reports available for teachers use

5.4.1. The Lessons visited report

The lessons visited report is available for both teachers and students. Every time a lesson is visited by a student or a teacher, the action is recorded in the database. In the student interface, students can track the lessons they have visited with some details such as the time of visit, the visited topic title and lesson ID. Any lesson may be visited as many times as required. While in the teacher interface, teachers can track all visited lessons by any student or teacher with all the mentioned details as in the student interface. This report gives the student an idea on their study progress. On the other hand, teachers can monitor students' progress besides some other analysis that can be conducted and realized from the report results. The report results can be sorted according to the topic title, lesson ID, student username or the date and time of when the lesson was visited.

Figure 5.29 is a snapshot of the lessons visited report

Back to main Page

Currently logged in as: student1

Visited Lessons

TopicTitle	LessonID	DateTimevisited	UserName
Basic System Maintenance	20	12/26/2007 12:19:00 PM	student1
Hardware	7	12/26/2007 12:21:00 PM	student1
Hardware	7	12/26/2007 12:36:00 PM	student1
Software	15	12/26/2007 12:40:00 PM	student1
Getting Started	1	12/28/2007 10:56:00 PM	student1
Basic System Maintenance	19	1/5/2008 1:42:00 PM	student1
Basic System Maintenance	21	1/5/2008 1:42:00 PM	student1
The Use of IT in Everyday Life & The Electronic World	27	1/5/2008 1:43:00 PM	student1
Hardware	7	1/6/2008 12:24:00 PM	student1
Basic System Maintenance	20	1/6/2008 12:25:00 PM	student1

1 2

Figure 5.29: visited lessons report as seen by the student

To create this report, we built a query as in Figure 5.30

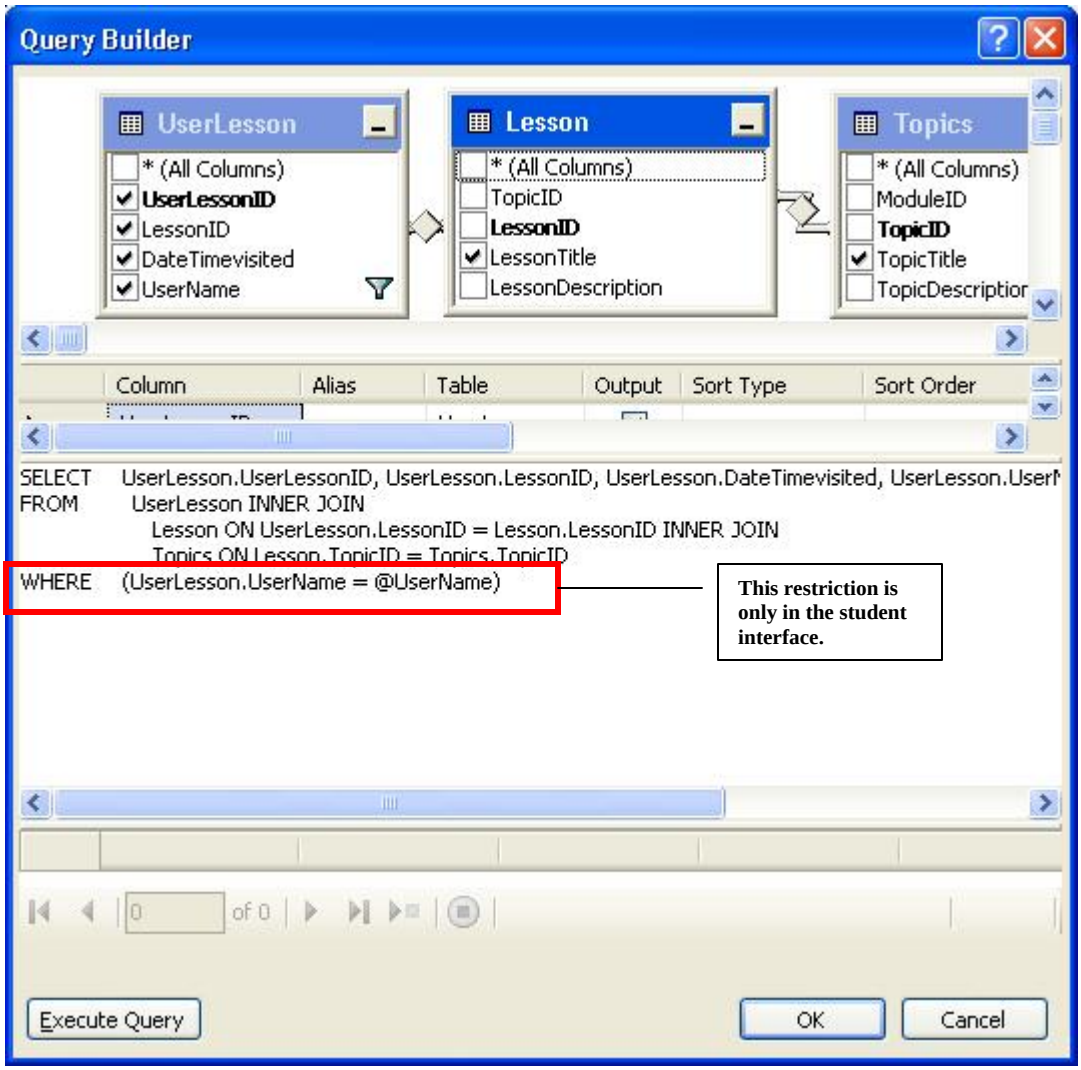


Figure 5.30: lessons visited query – Student Interface

5.4.2 Visited Assessments

Both students and teachers can also see the visited assessment or the assessment history report. The word assessment here means the final exam which includes the full module chapters and lessons. Usually this assessment is done at the end of the module after students feel ready to perform the real module international examination. This assessment

will be discussed in more details later in this chapter when we talk about the evaluation tools. As with the visited lessons report, students can only see their visited assessments while teachers can see visited assessments and history done by all students. Figure 5.31 shows an example of the visited assessments report.

Figure 5.31: An example of the visited assessments report

To build this report, we used the GridView tool (as in Figure 5.32) then ran a query as explained in Figure 5.33.

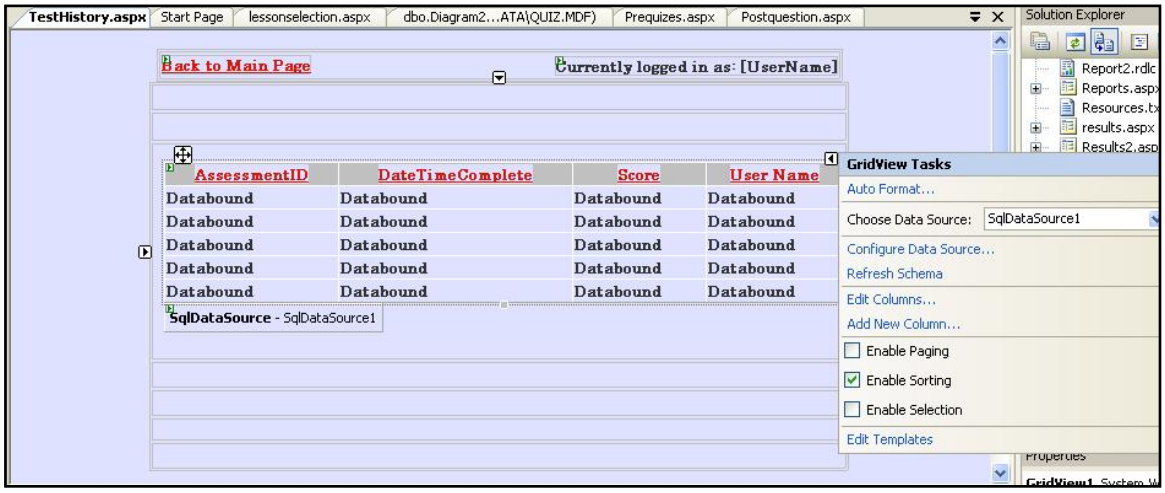


Figure 5.32 illustrates the GridView used for the assessments history report

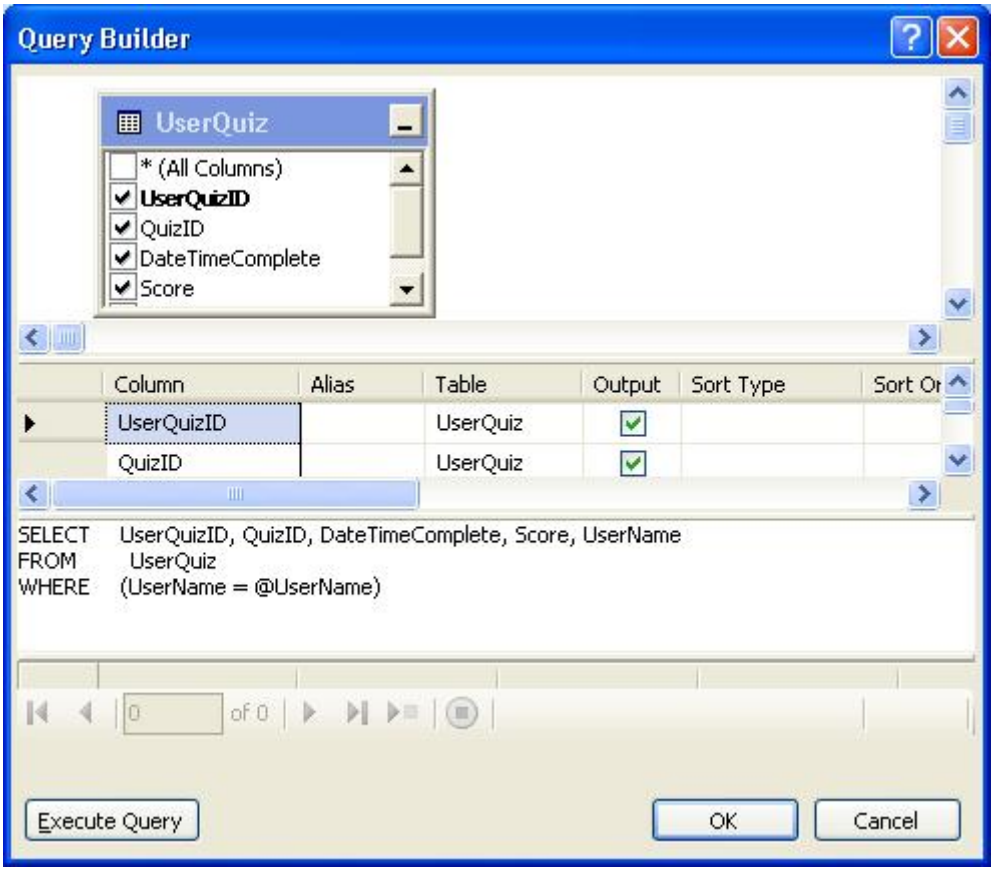


Figure 5.33 The query used for the assessments history report

5.4.3 The visited Post-Quizzes and the visited Pre-Quizzes reports

The visited Post-Quizzes and the visited Pre-Quizzes reports are both available for teachers to track students study progress and at the same time the Post-Quizzes can be used as an evaluation tool if the system is employed to teach computer fundamentals. More details about these two reports will be explained later in this chapter when we talk about the different evaluation tools.

To create these two reports we followed the same concept as in the visited assessment report. In other words, we created the interface using the GridView tool and ran a query for each report. Figure 5.34 shows the .aspx code of the query formed for the visited Post-Quizzes report while Figure 5.35 shows the query formed for visited Pre-quizzes report .

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
                        SelectCommand="SELECT * FROM [UserPostQuiz]">
                    </asp:SqlDataSource>
```

Figure 5.34: Query for the visited Post-Quizzes report – from the markup .aspx code

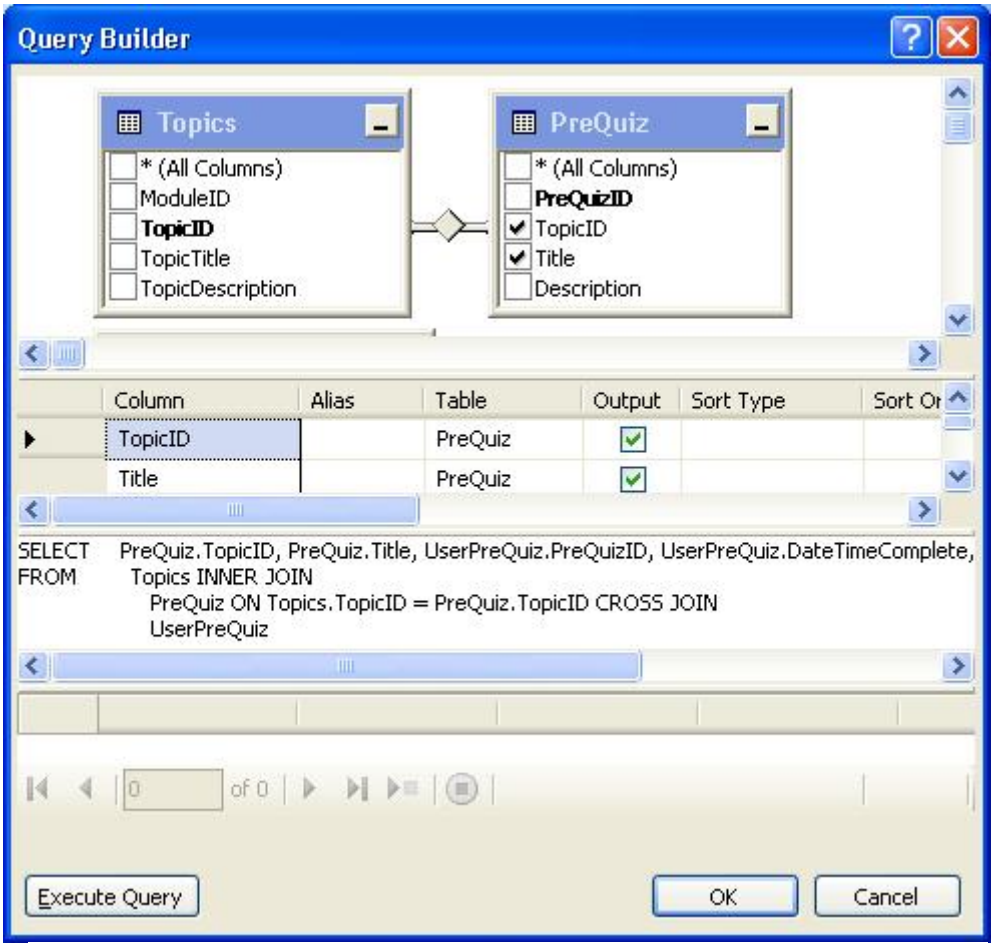
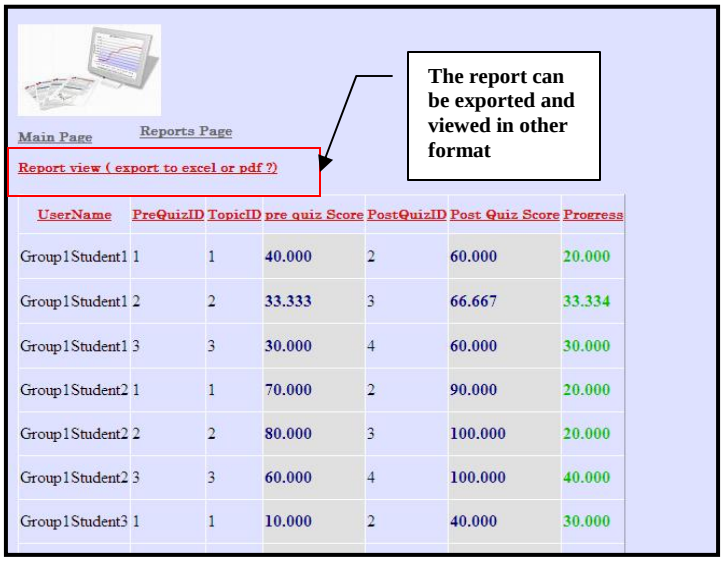


Figure 5.35: the query built for the visited Pre-Quizzes report

5.4.4 The Students Progress Report

The students' progress report is designed to give feedback and help teachers and educators assess students' progress and identify system weaknesses. It compares the students' pre-quiz score with the post-quiz score to calculate the progress. Report results can be sorted according to the student username, Topic ID, quiz ID, pre-quiz results, post quiz results or the progress. Figure 5.36 is an example of the report results.



UserName	PreQuizID	TopicID	pre quiz Score	PostQuizID	Post Quiz Score	Progress
Group1Student1	1	1	40.000	2	60.000	20.000
Group1Student1	2	2	33.333	3	66.667	33.334
Group1Student1	3	3	30.000	4	60.000	30.000
Group1Student2	1	1	70.000	2	90.000	20.000
Group1Student2	2	2	80.000	3	100.000	20.000
Group1Student2	3	3	60.000	4	100.000	40.000
Group1Student3	1	1	10.000	2	40.000	30.000

Figure 5.36 the Students Progress Report

To design this report, we created a query which involved many tables besides performing some calculations. The following figure (Figure 5.37) shows the query in details.

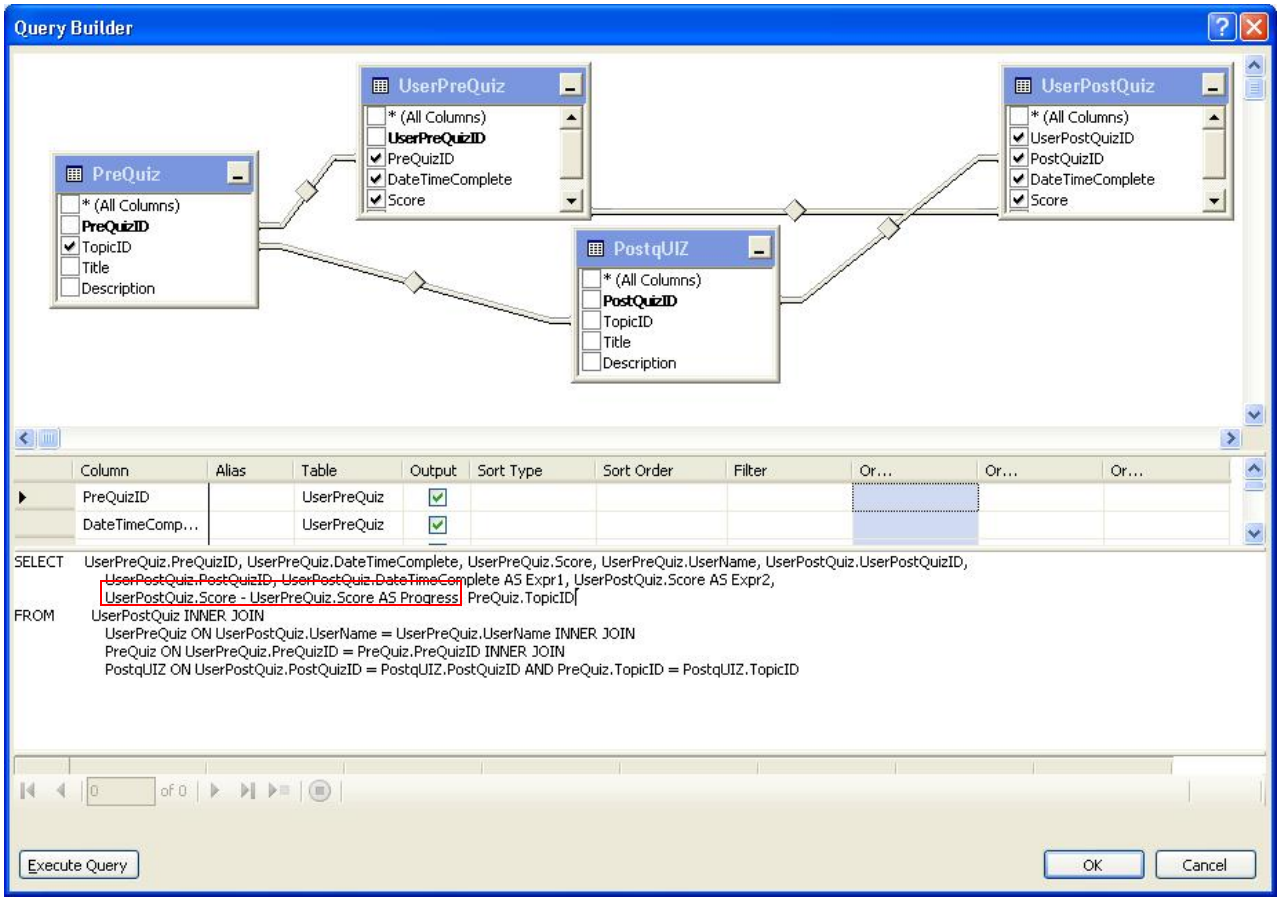
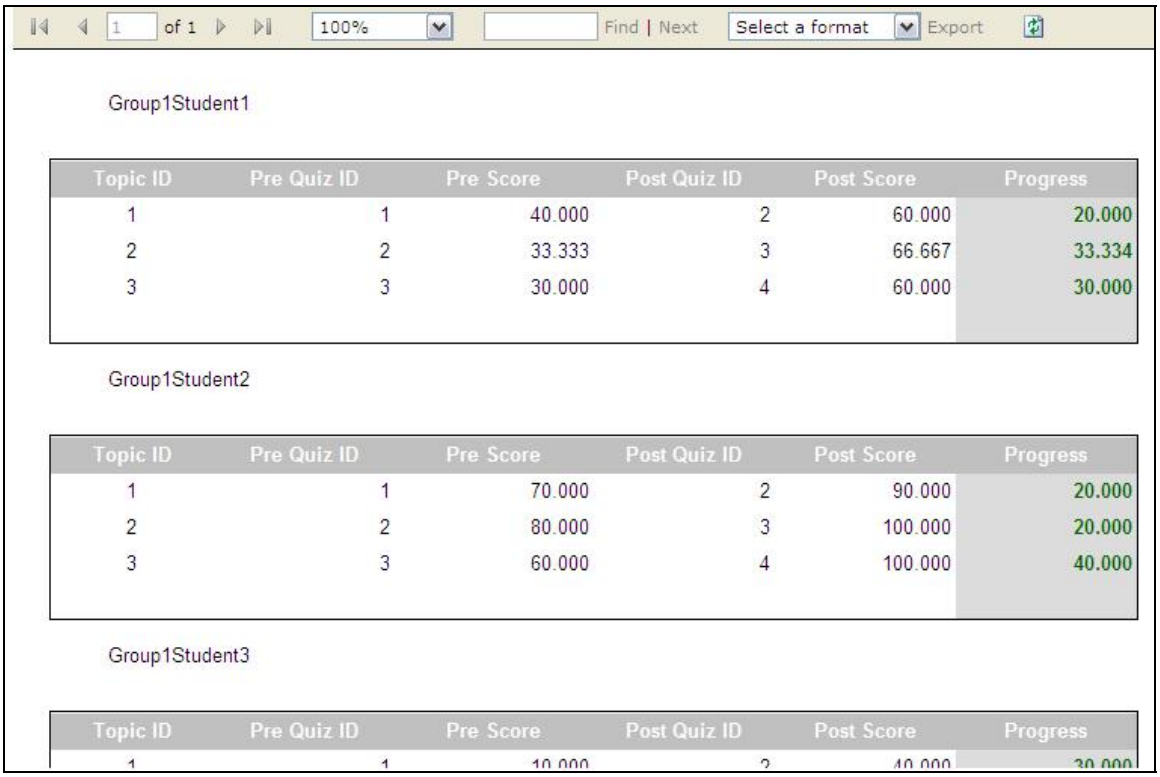


Figure 5.38: The Students progress report viewed in excel form

The results can be viewed in and exported to different formats like excel and pdf. This is to help users further analyze and learn from the results. Figure 5.38 is an example of the export where the user can view and export results to other formats while Figure 5.39 shows exported results to a workbook (excel format).



1 of 1 100% Find | Next Select a format Export

Group1Student1

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	40.000	2	60.000	20.000
2	2	33.333	3	66.667	33.334
3	3	30.000	4	60.000	30.000

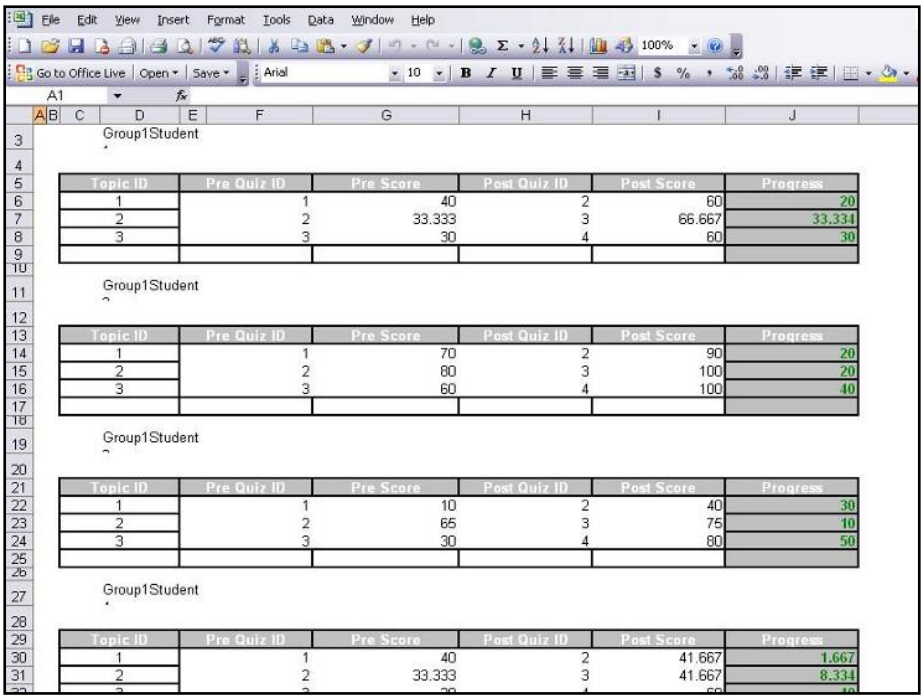
Group1Student2

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	70.000	2	90.000	20.000
2	2	80.000	3	100.000	20.000
3	3	60.000	4	100.000	40.000

Group1Student3

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	10.000	2	40.000	30.000

Figure 5.38: The Students progress report where results can be exported to other formats



File Edit View Insert Format Tools Data Window Help

Go to Office Live | Open | Save | Arial 10 100%

A1

Group1Student

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	40	2	60	20
2	2	33.333	3	66.667	33.334
3	3	30	4	60	30

Group1Student

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	70	2	90	20
2	2	80	3	100	20
3	3	60	4	100	40

Group1Student

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	10	2	40	30
2	2	65	3	75	10
3	3	30	4	80	50

Group1Student

Topic ID	Pre Quiz ID	Pre Score	Post Quiz ID	Post Score	Progress
1	1	40	2	41.667	1.667
2	2	33.333	3	41.667	8.334

Figure 5.39: The Students progress report exported to and excel sheet

To make results available to be viewed in other formats, we utilized the report viewer tool as in Figure 5.40. The report viewer tool provides a simple report preview with some export options and navigation tools as in Figure 5.41

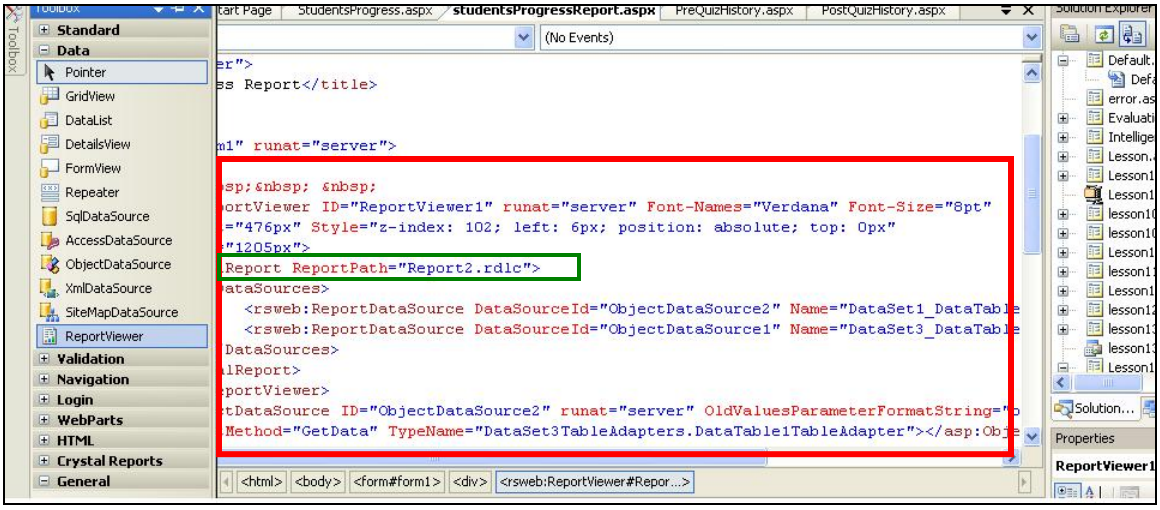


Figure 5.40 shows the report viewer tool and its code

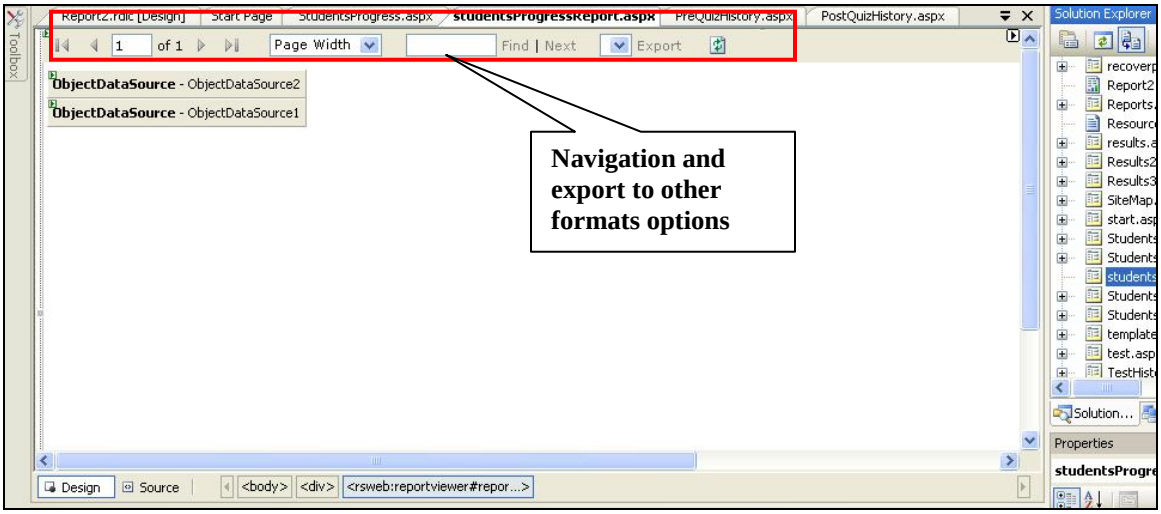


Figure 5.41 – Report preview interface

As shown in Figure 5.40, the report viewer displays Report2.rdlc which is the report we have designed as in Figure 5.42

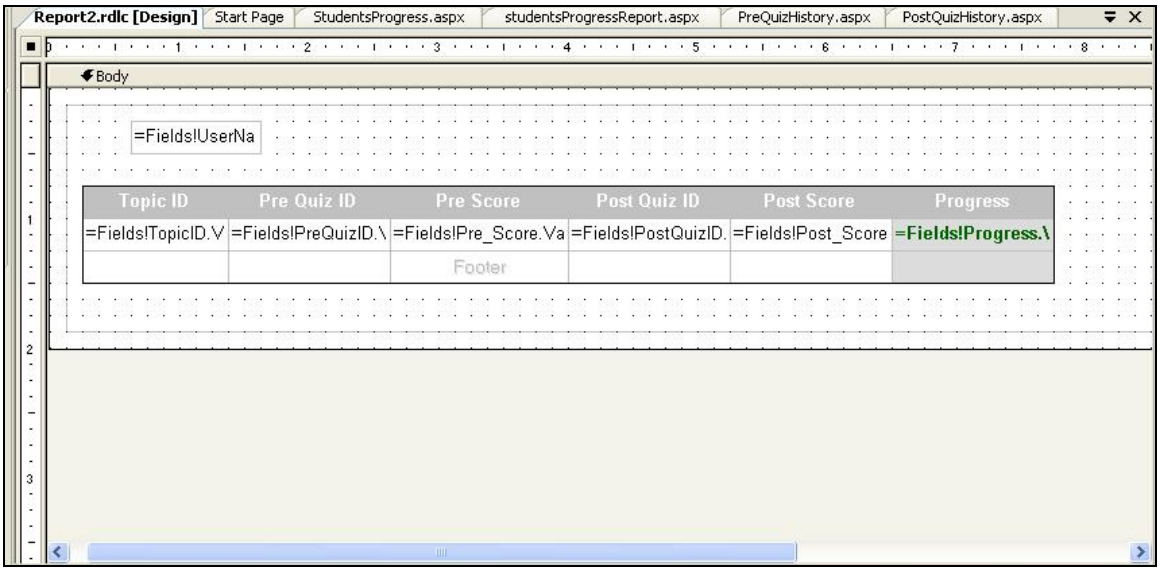


Figure 5.21 report2.rdlc design (the report design)

While the dataset table is an xsd file defined as in Figure 5.22

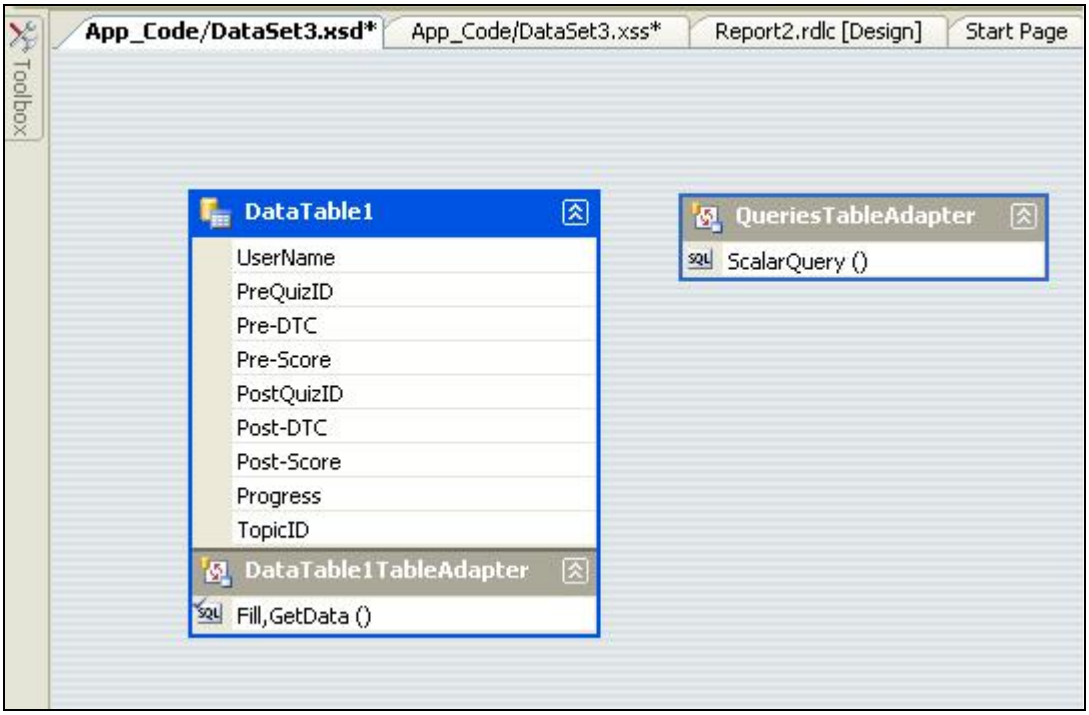


Figure 5.22 the report data table

5.4.5 The Quizzes Analysis Report

Another important report that teachers can generate is the quizzes analysis report. This report analyzes any type of quizzes as in Figure 5.23 to give a clear picture about student's previous knowledge, acquired knowledge and to find out common difficulties amongst students. This is besides that this type of report can be used to enhance the lessons and the questions bank.

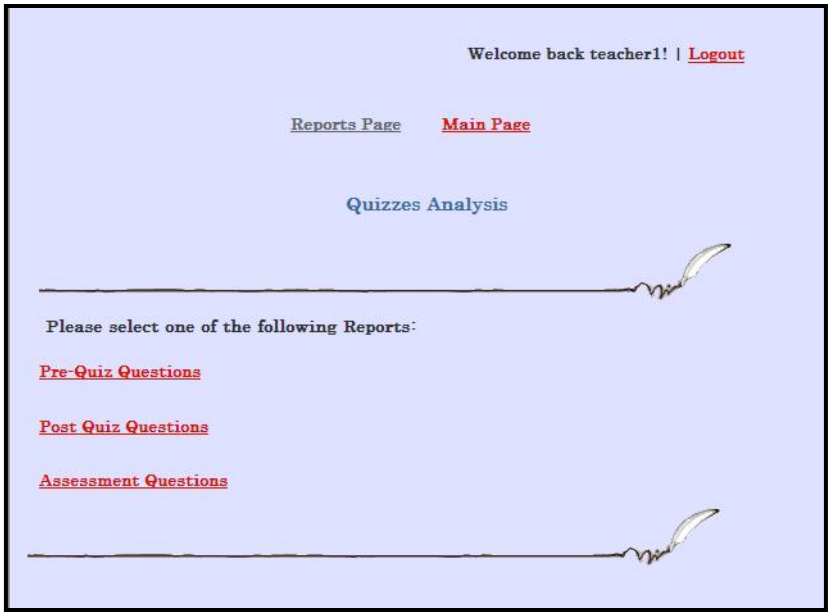


Figure 5.23 Quiz analysis reports

Figure 5.24 shows an example of the pre-quiz analysis, while Figure 5.25 is an example of the post quiz analysis.

Reports Page Main Page

Welcome back teacher1! | Logout

Pre Quizzes
Analysis

ModuleTitle	TopicTitle	PreQuizID	Score	count
Basic Concepts of Information Technology	Getting Started	1	10.000	2
Basic Concepts of Information Technology	Getting Started	1	20.000	1
Basic Concepts of Information Technology	Getting Started	1	40.000	7
Basic Concepts of Information Technology	Getting Started	1	50.000	4
Basic Concepts of Information Technology	Getting Started	1	58.333	1
Basic Concepts of Information Technology	Getting Started	1	60.000	2
Basic Concepts of Information Technology	Getting Started	1	70.000	3
Basic Concepts of Information Technology	Getting Started	1	80.000	1
Basic Concepts of Information Technology	Hardware	2	33.333	6
Basic Concepts of Information Technology	Hardware	2	50.000	1

1 2 3

For example, this analysis shows that for the getting started lesson, most students have some back ground about the subject and in average they score between 40% and 50% while no students got higher than 80%

Figure 5.24: an example of the pre-quiz analysis report

ModuleTitle	TopicTitle	PostQuizID	Score	count
Basic Concepts of Information Technology	Getting Started	2	40.000	1
Basic Concepts of Information Technology	Getting Started	2	50.000	2
Basic Concepts of Information Technology	Getting Started	2	60.000	3
Basic Concepts of Information Technology	Getting Started	2	70.000	1
Basic Concepts of Information Technology	Getting Started	2	80.000	6
Basic Concepts of Information Technology	Getting Started	2	83.333	2
Basic Concepts of Information Technology	Getting Started	2	90.000	3
Basic Concepts of Information Technology	Hardware	3	41.667	1
Basic Concepts of Information Technology	Hardware	3	58.333	1
Basic Concepts of Information Technology	Hardware	3	66.667	1

1 2

For the same group of students as in figure 3.24, the post quiz analysis shows that most students got 80% and above while no students got less than 40% but still no students got 100% which means that the lessons and practices could be enhanced.

Figure 5.25: an example of the post-quiz analysis report

This report is a prototype which could be enhanced in the future to include percentage or even go to the question level analysis.

To construct this report, we created a query for each type of quizzes .Figure 5.26 is an example of the query used for the post quizzes analysis report.

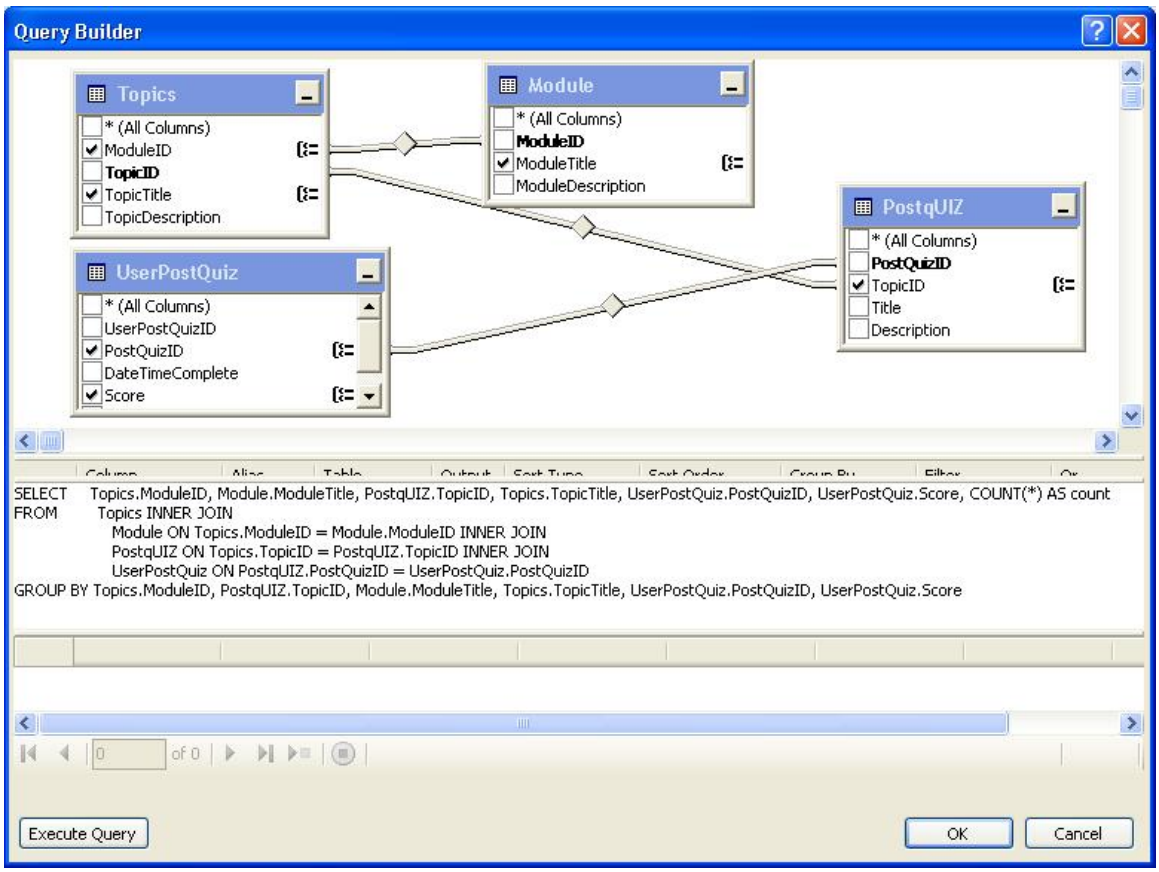


Figure 5.26: the post Quiz analysis Query

To view the report, we used the GridView Data tool. Figure 5.27 demonstrates the GridView for the post Quizzes analysis report.

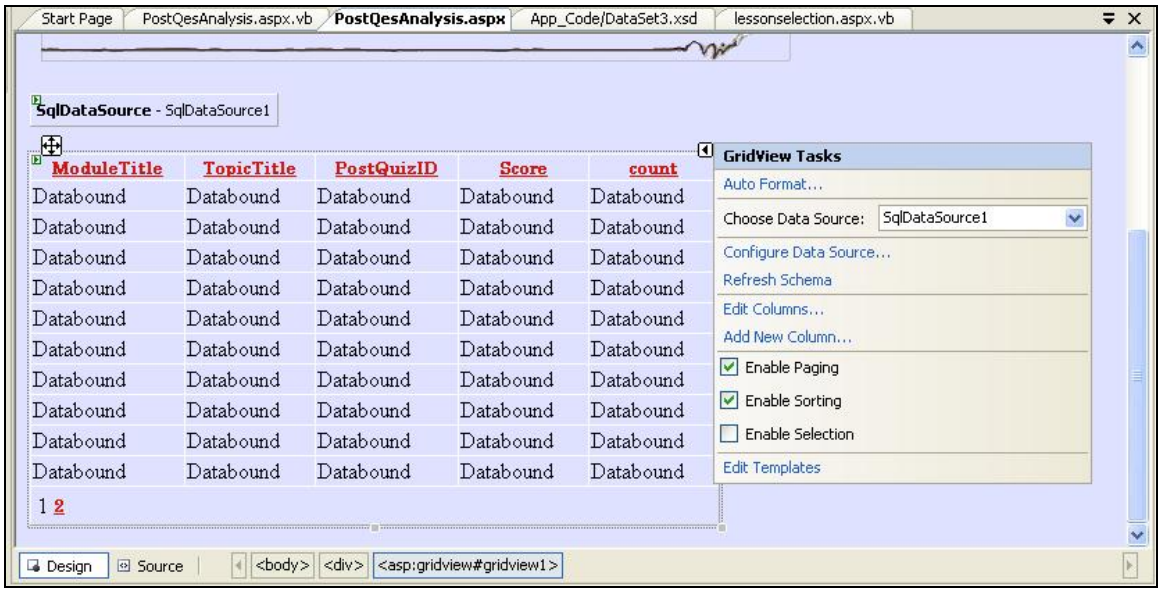


Figure 5.27: Post Quizzes analysis Data Grid View

5.5 The evaluation tool

The system can be used as an evaluation tool in different ways**.** The system contains more than 400 different quiz questions distributed between pre-quizzes, post-quizzes, assessment and the intelligent practice questions. While all are considered as evaluation tools, as mentioned before, in this thesis we employ the different types of evaluation tools to perform different tasks. The pre-quizzes are meant to estimate students' previous knowledge while Post –quizzes are designed to evaluate students post lessons knowledge. We can compare the two results of the pre-quizzes and the post-quizzes to assess students gained knowledge. Moreover, in this thesis the word "assessment(s)" is used to identify the end of module exam(s) that assess students understanding of the overall module and readiness for the international examination. Figure 5.28 explains the different types of evaluation tools and the way they are distributed.

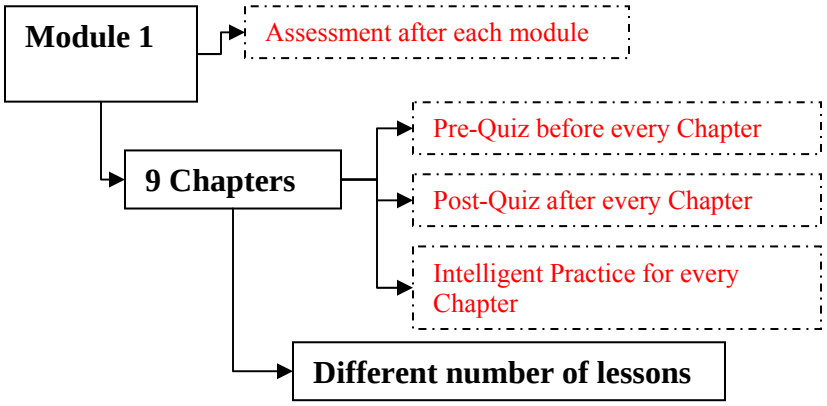


Figure 5.28: Evaluation tool

From the Figure 5.28, the system consists of one module and an assessment at the end of the module to assess students understanding of the module overall. The module is made up of 9 chapters. Each chapter contains different lessons and two different evaluation tools: a pre-quiz and a post quiz for each chapter besides the intelligent practice which we will explain in details later in this chapter.

As discussed in Chapter 3, all questions are MCQs. To make it easy for students to concentrate on the question, the system introduces one question at a time in a simple and straightforward layout. After every assessment or quiz, a results report including the score with brief feedback is shown. The report shows the correct, incorrect and the correct answers for all questions. Figure 5.29 shows an example of a quiz question and a result report.

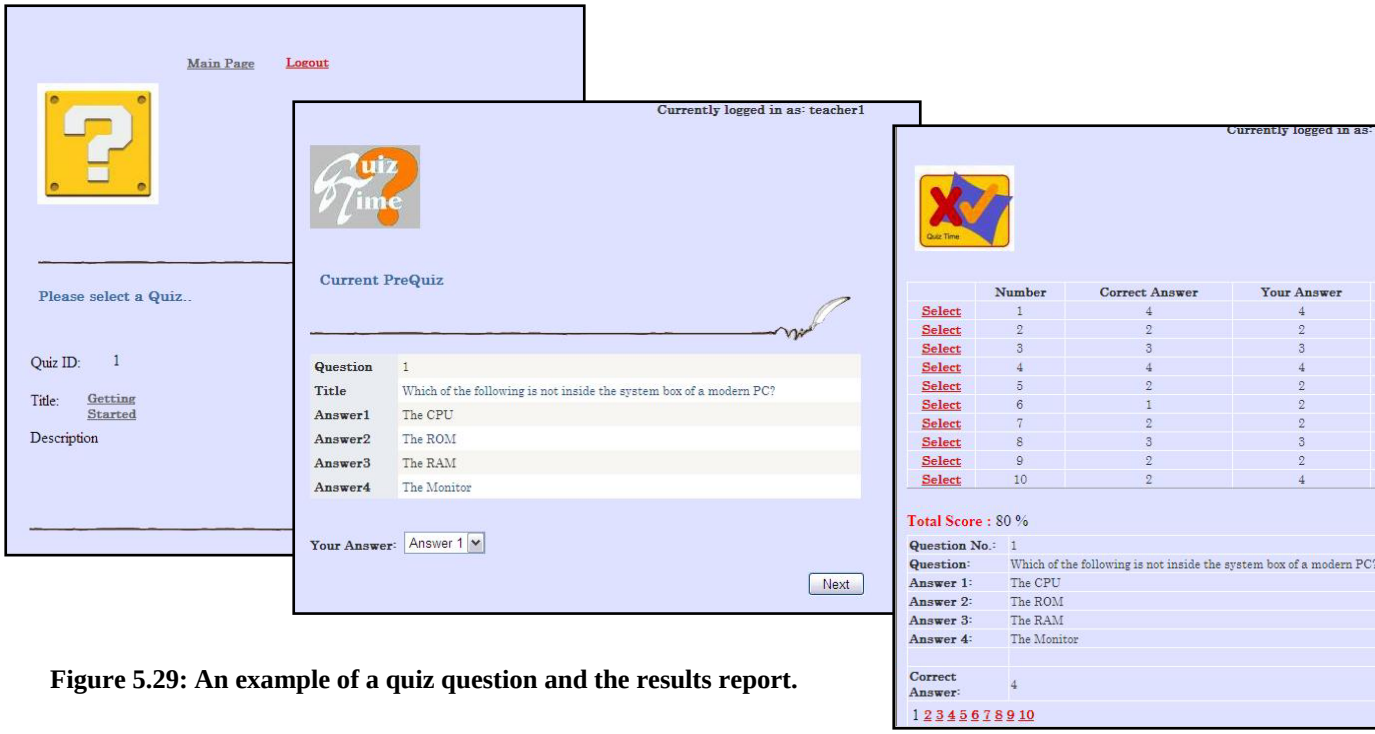


Figure 5.29: An example of a quiz question and the results report.

Teachers can view, modify, add, and delete questions easily from a tool provided in the teachers. Figure 5.30 shows an example of how teachers can add, modify and delete questions from any of the quizzes or the assessment. In the following subsections, the different evaluation tools are discussed in more details.

Figure 5.30: teachers can add, modify and delete questions in any quiz or assessment

5.5.1 Assessment

Assessment test is done after every module. It can be used to assess the students' readiness for the ICDL/ECDL real test or as an assessment tool if used for school education as material for a computer basics course.

5.5.2 Pre-Quizzes

Pre-quizzes are done before each chapter. There are many benefits for both teachers and students from doing these quizzes such as the following:

Students can use the quizzes system to get some idea about their level of knowledge with the material in the chapter. They can find out what to expect and what lessons will be covered in a specific chapter. They can also anticipate their weakness points. Teachers can use the system to get some information about students IT background knowledge related to a specific lesson and their weakness points (this helps for future system enhancements too), and to evaluate system performance under a controlled environment.

For now the system includes one Pre-quiz per chapter but at the same time is has the flexibility to accommodate as many quizzes as needed.

5.5.3 Post-Quizzes

Post-Quizzes are given after each chapter. There are many benefits for both teachers and students from doing these quizzes. While students can evaluate their progress and assess their readiness to go further with other chapters and get an idea about the points they did not understand, teachers can use the system to assess students learning gain, evaluate a student's progress, evaluate system performance, and evaluate teaching strategy.

5.5.4 Constructing a quiz

All the evaluation tools that we discussed in this section were constructed using the same methodology. To explain how we built them, we will take the post-Quiz as an example.

Post quizzes are available for each chapter from the lessons page as in Figure 5.31

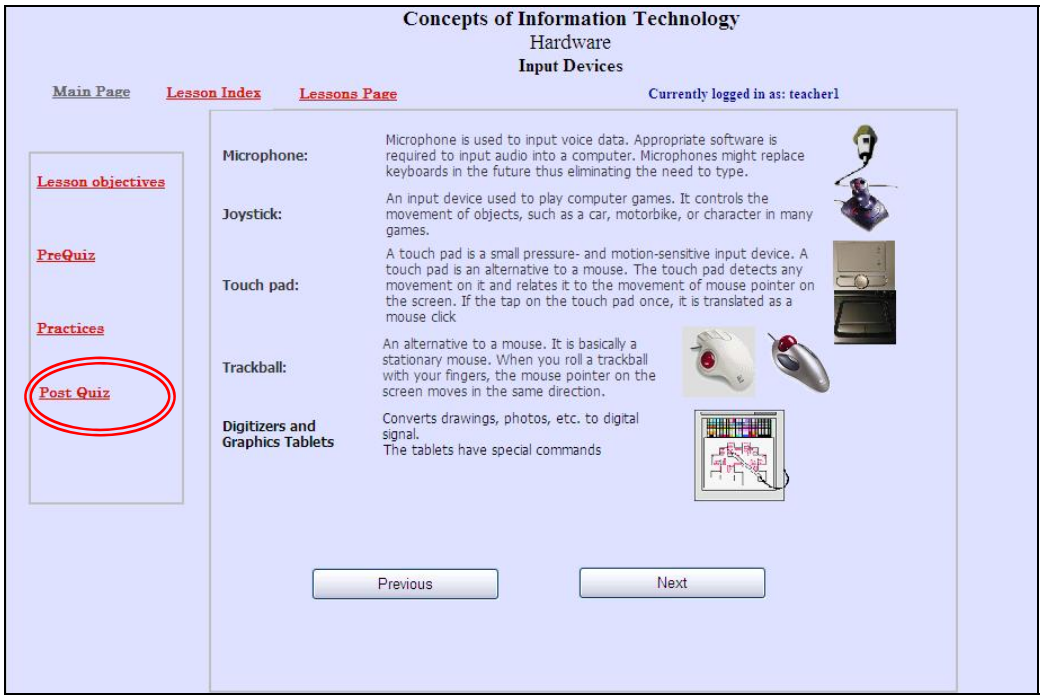


Figure 5.31 shows how to access a post quiz from a lesson page

From the hyperlink, the system leads the users to the available post quizzes list for the specific chapter according to the topic ID number. In our system we included one Pre-Quiz and one Post Quiz for every chapter but the system is built in a flexible mechanism to expand and include more Pre- and Post-quizzes as needed as shown in Figure 5.32.

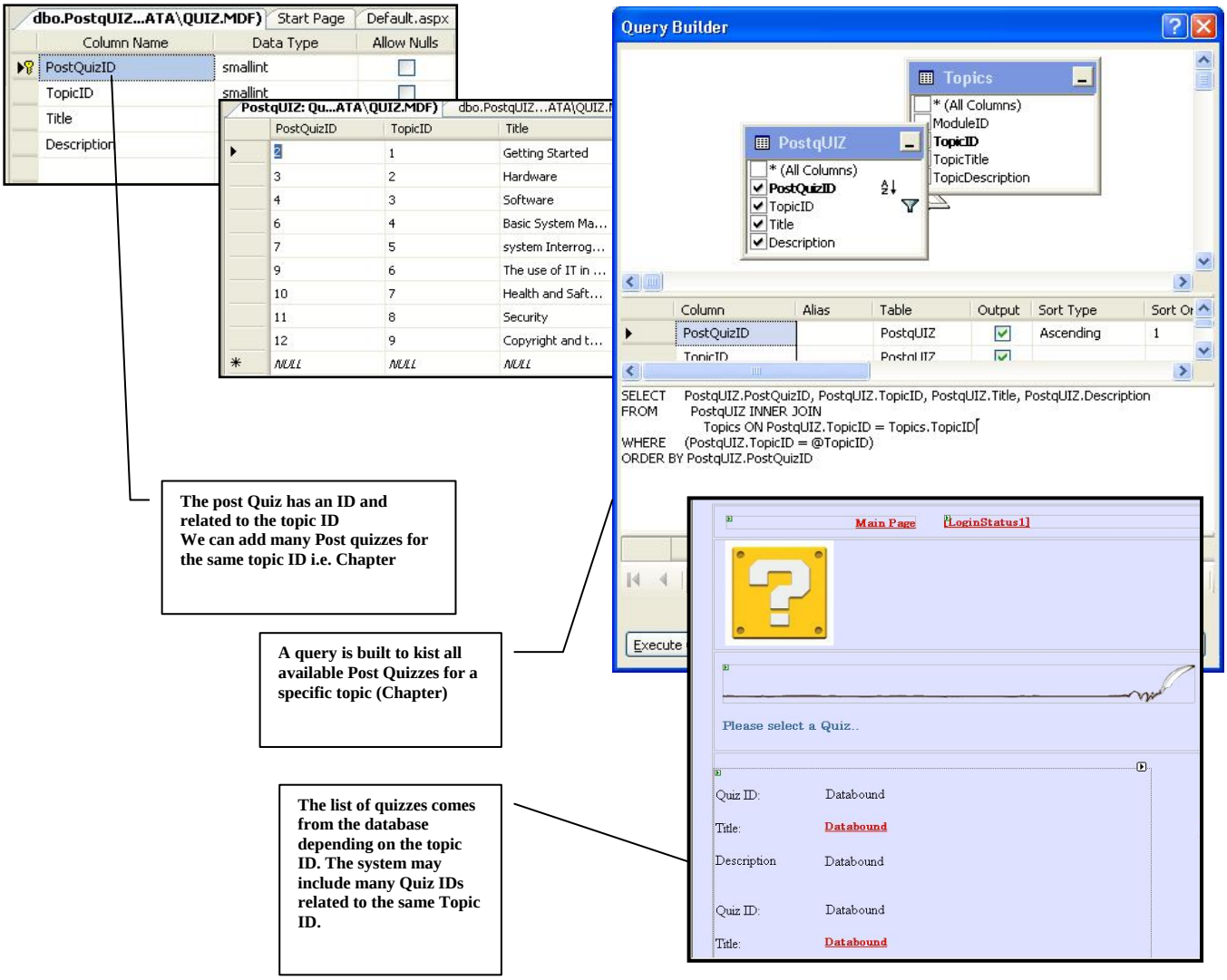


Figure 5.32: The list of available Post Quizzes comes from the database

After a specific post quiz is selected, the system starts displaying the related MCQs one question at a time. Figure 5.33 shows the questions table, the table content, and then the SQL query employed to display only related questions to the user in the user interface.

.

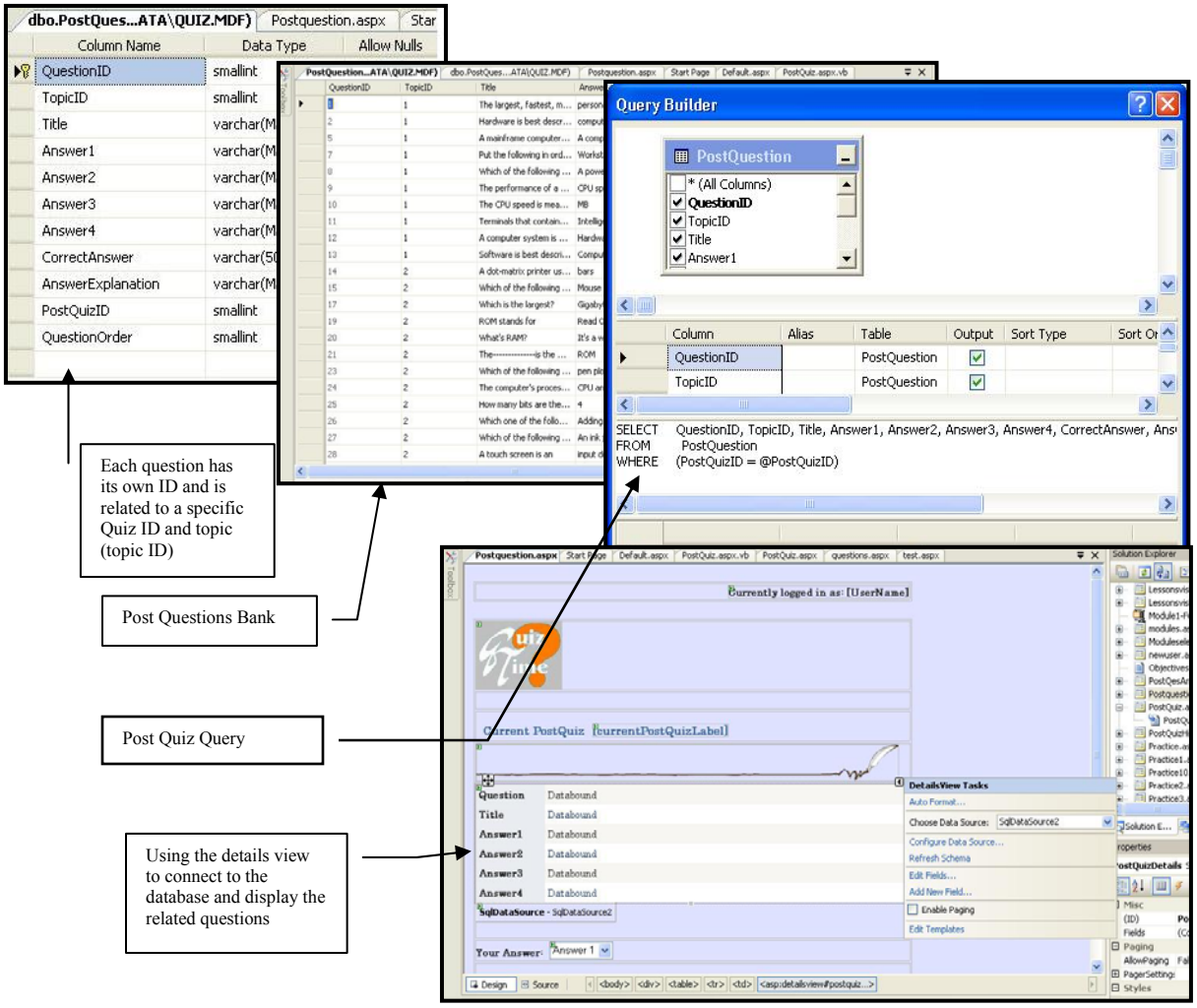


Figure 5.33 illustrates the Post Quiz building stages

5.5.5 Results Calculations

When the user performs any type of exam, all selected answers are stored in an array with the question ID, order and correct answer. Then the user answer is compared to the correct answer and the score is calculated accordingly. Figure 5.34 shows the Post Quizzes questions code where answers are recorded with their Question ID, order number and correct answer in an array to be evaluated. The system checks if the question is the last question or not as in Figure 5.35. If the question is the last one, the system directs the user to the results page where score calculation takes place with some feedback about correct and incorrect answers. If the answer is incorrect the correct answer is displayed. Figure 5.36 shows the VB application code (evaluation) then Figure 5.37 explains the score calculation criteria and how the results are stored in the database for later reference. The stored results are used, as previously explained, in the post, pre quizzes and assessment history (visited post-quizzes, visited pre-quizzes and visited assessments) report.

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles buttonNext.Click

  Try
    ' Save off previous answers
    Dim dr As System.Data.DataRowView
    dr = CType(PostQuizDetails.DataItem, System.Data.DataRowView)

    ' Create Answer object to save values
    Dim a As Evaluation = New Evaluation()
    a.QuestionID = dr("QuestionID").ToString()
    a.QuestionOrder = dr("QuestionOrder").ToString()
    a.CorrectAnswer = dr("CorrectAnswer").ToString()
    a.UserAnswer = answerDropDownList.SelectedValue.ToString()

    Dim al As ArrayList
    al = CType(Session("AnswerList"), ArrayList)
    al.Add(a)

    Session.Add("AnswerList", al)

  Catch ex As Exception
  End Try
```

Figure 5.34 - Answers stored in an array (VB Code)

```
If PostQuizDetails.PageIndex = PostQuizDetails.PageCount - 1 Then
        ' Go to evaluate answers
        Response.Redirect("results3.aspx")
    Else
        answerDropDownList.SelectedValue = 1
        PostQuizDetails.PageIndex += 1
    End If

    If PostQuizDetails.PageIndex = PostQuizDetails.PageCount - 1 Then
        buttonNext.Text = "Finished"
    End If
```

Figure 5.35 – Last question check VB code

```
Public Class Evaluation

    Private the_QuestionID As String
    Private the_QuestionOrder As String
    Private the_CorrectAnswer As String
    Private the_UserAnswer As String
    Private the_Result As String

    Public Property QuestionID() As String
        Get
            Return the_QuestionID
        End Get
        Set(ByVal value As String)
            the_QuestionID = value
        End Set
    End Property
    Public Property QuestionOrder() As String
        Get
            Return the_QuestionOrder
        End Get
        Set(ByVal value As String)
            the_QuestionOrder = value
        End Set
    End Property

    Public Property CorrectAnswer() As String
        Get
            Return the_CorrectAnswer
        End Get
        Set(ByVal value As String)
            the_CorrectAnswer = value
        End Set
    End Property

    Public Property UserAnswer() As String
        Get
            Return the_UserAnswer
        End Get
        Set(ByVal value As String)
            the_UserAnswer = value
        End Set
    End Property

    Public ReadOnly Property Result() As TheResult
        Get
            If the_UserAnswer = the_CorrectAnswer Then
                Return TheResult.Correct
            Else
                Return TheResult.Incorrect
            End If
        End Get
    End Property

    Public Enum TheResult
        Correct
        Incorrect
    End Enum
End Class
```

```
If IsPostBack = False Then

        ' calculate score
        Dim questions As Double = al.Count
        Dim correct As Double = 0

        Dim i As Integer = 0
        For i = 0 To al.Count - 1
           Dim a As Evaluation = CType(al(i), Evaluation)
           If a.Result = Evaluation.TheResult.Correct Then
              correct += 1
           End If
        Next

        Dim score As Double = (correct / questions) * 100
        Label1.Text = score.ToString() & " %"

        Dim UserPostQuizDataSource As SqlDataSource = New SqlDataSource()
        UserPostQuizDataSource.ConnectionString =
ConfigurationManager.ConnectionStrings("ConnectionString").ToString()
        UserPostQuizDataSource.InsertCommand = "INSERT INTO [UserPostQuiz] ([PostQuizID],
[DateTimeComplete], [Score], [UserName]) VALUES (@PostQuizID, @DateTimeComplete, @Score, @UserName)"

        UserPostQuizDataSource.InsertParameters.Add("PostQuizID", Session("PostQuizID").ToString())
        UserPostQuizDataSource.InsertParameters.Add("DateTimeComplete", DateTime.Now.ToString())
        UserPostQuizDataSource.InsertParameters.Add("Score", score.ToString())
        UserPostQuizDataSource.InsertParameters.Add("UserName", User.Identity.Name)

        Dim rowsAffected As Integer = UserPostQuizDataSource.Insert()
        If rowsAffected = 0 Then
           ' Let's just notify that the insertion didn't
           ' work, but let's continue on ...
           errorLabel.Text = "There was a problem saving your quiz results " & _
                 "into our database.  Therefore the results from this quiz " & _
                 "will not be displayed on the list on the main menu."
        End If

     End If
```

Figure 5.37 Results calculation and results storage in the database code

5.5.6 Displaying results

In the previous subsection, we explained how the results are calculated using Visual Basic code, and then the results are stored in the database for future reference. Results can be used for example to run and generate different reports in order to track students' progress and enhance the system. In this section we will explain how the results are displayed after each type of Quiz or test.

To display results we utilized both the GridView and DetailsView data tools as demonstrated in Figure 5.38. The GridView is utilized to display the user correct and incorrect answers with the score calculated just under the Grid. Figure 5.39 shows the aspx code of how results are displayed in the Grid view. The user can select any question to view more details about it such as the question stem, the available list of choices and the correct answer. These details are displayed using the DetailsView tool as explained in Figure 5.40.

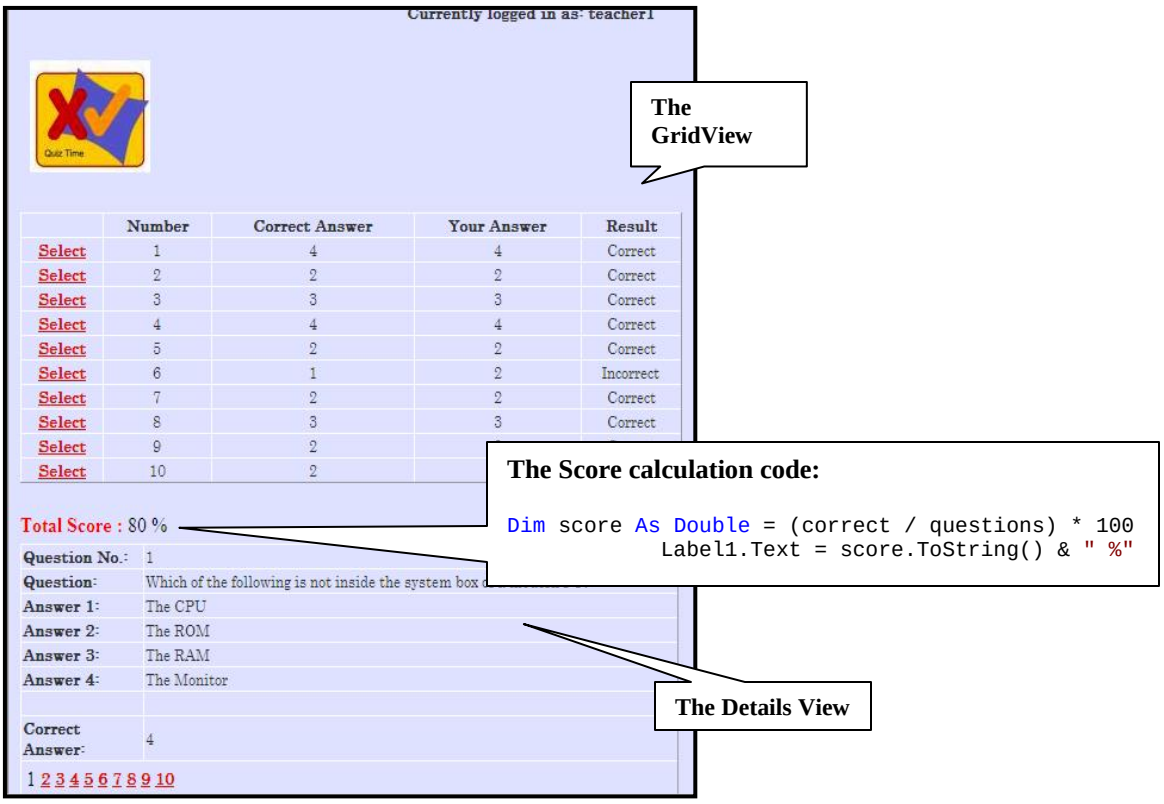


Figure 5.38 – the results Page

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:ConnectionString %>"
            SelectCommand="SELECT Title, Answer1, Answer2, Answer3, Answer4, QuestionID, QuestionOrder,
CorrectAnswer, AnswerExplanation, PostQuizID FROM PostQuestion WHERE (PostQuizID = @PostQuizID) ORDER BY
QuestionOrder">
            <SelectParameters>
              <asp:SessionParameter Name="PostQuizID" SessionField="PostQuizID" />
            </SelectParameters>

          </asp:SqlDataSource>
        <td style="height: 208px">
          <asp:GridView ID="resultGrid" runat="server" DataKeyNames="QuestionOrder" SelectedIndex="0"
          AutoGenerateColumns="False" AutoGenerateSelectButton="True"
OnSelectedIndexChanged="resultGrid_SelectedIndexChanged" Width="555px" >
            <RowStyle CssClass="generaltext" HorizontalAlign="Center" />
            <Columns>
              <asp:BoundField DataField="QuestionOrder" HeaderText="Number" SortExpression="QuestionOrder" />
              <asp:BoundField DataField="CorrectAnswer" HeaderText="Correct Answer" />
              <asp:BoundField DataField="UserAnswer" HeaderText="Your Answer" />
              <asp:BoundField DataField="Result" HeaderText="Result" />

            </Columns>
            <HeaderStyle CssClass="boldtext" />
```

The SQL query

Figure 5.39 – the Grid View aspx (markup) code including the sql query

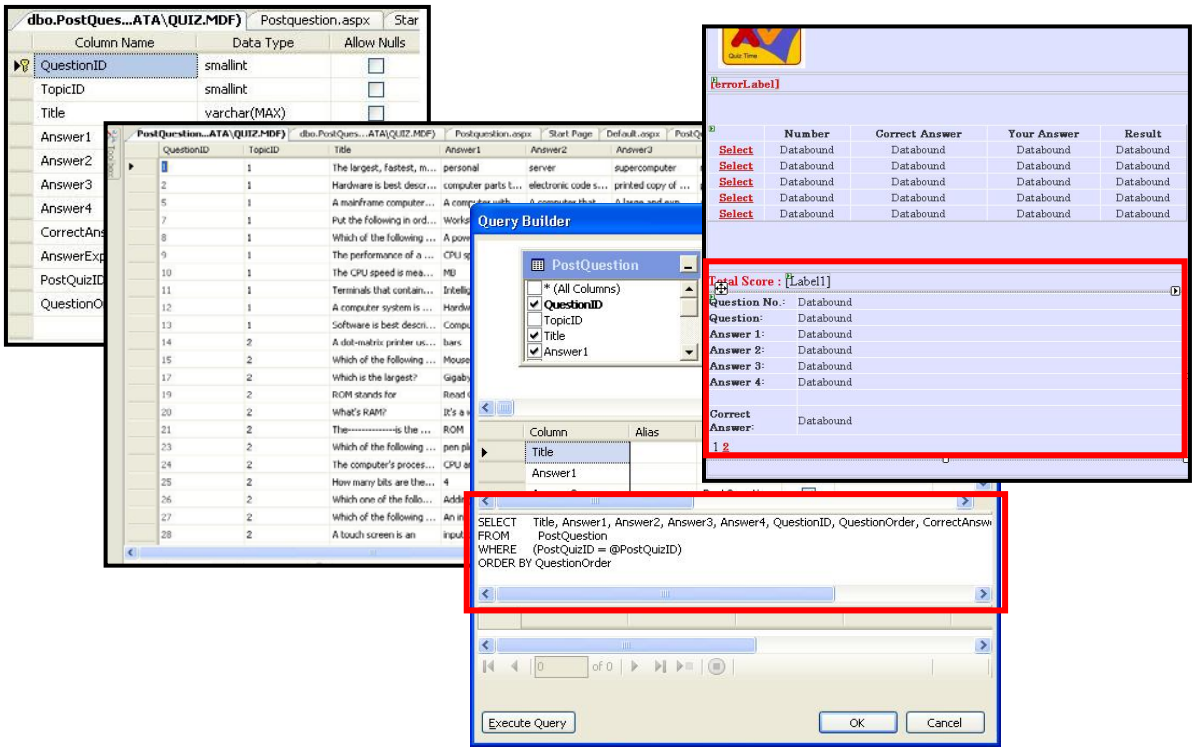


Figure 5.40 – the DetailsView sql query

5.6 The communication tools

Human touch and real communication help students feel more attached and confident with the system. With web-based education people can communicate via electronic means of communications such as Emails, forums. In this research, we decided that we want students to share information, support each other and give each other feedback besides the teachers support and help. Hence, we created two communication tools: the questions section and the comments section as illustrated in Figure 5.41. Students are given the required permission to ask questions, and view other students' questions to learn from. Teachers have permission to view questions and answer them. In addition, students can provide feedback to each other and to their teacher through the comments section and view other students' feedback under teachers' supervisions in the comments section. Teachers can delete any misleading or wrong information and/or correct them as required. This opens the door for discussions and collaboration where students learn from each other. Such environment gives the feel of real classroom and support from teachers and peers.

All questions and feedback information are stored in a database table. The page layout is designed in a simple way taking in consideration that students level. Only one record is displayed at a time for easier reading and navigation. Figure 5.42 provides a feel of how the questions and the comments section look from both the teacher and student interfaces.

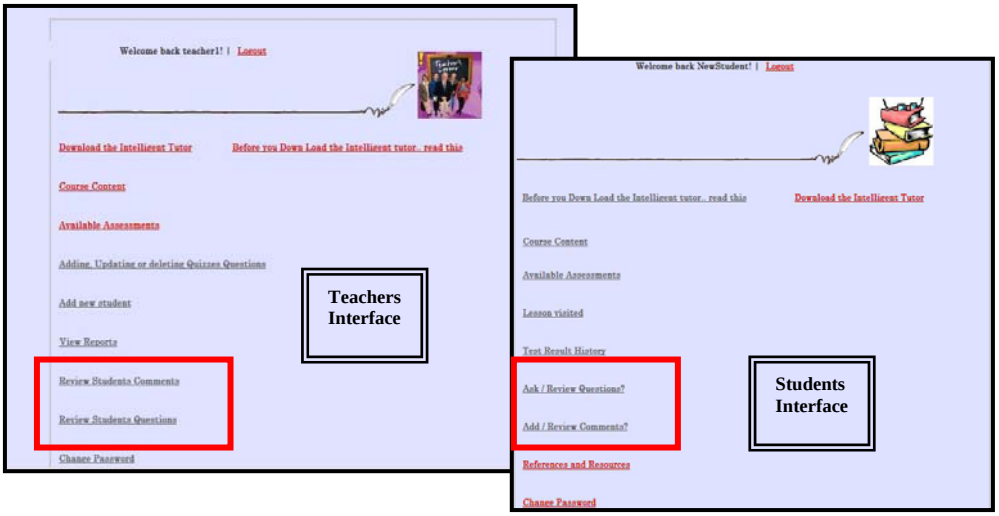


Figure 5.41- Communication tools

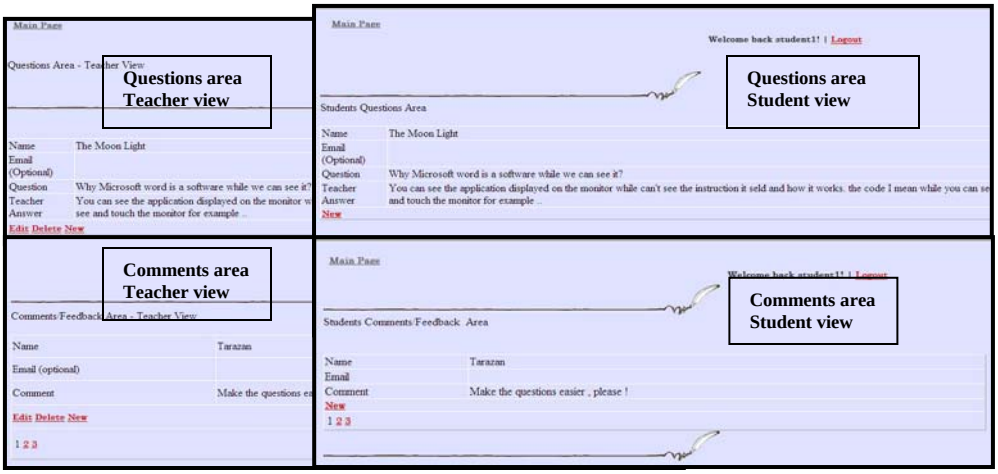


Figure 5.42: communication tools, Students and teachers views of the questions area tool

5.6.1 The Comments / Feedback Tool

As mentioned above, the comments area is created for students to give us feedback about the lessons and the system. At the same time this tool can be used as a collaboration tool where students can submit suggestions, share experiences and support each other. The comments tool is accessible by students and teachers with different permissions. Students can submit new entries and view other students' entries while teachers can view, modify, add and delete entries. These permissions are necessary for teachers to supervise and monitor participations and control inappropriate submissions.

To create this tool, we utilized the DetailsView Data tool then built a query to save and retrieve data from the comments table as shown in Figure 5.43.

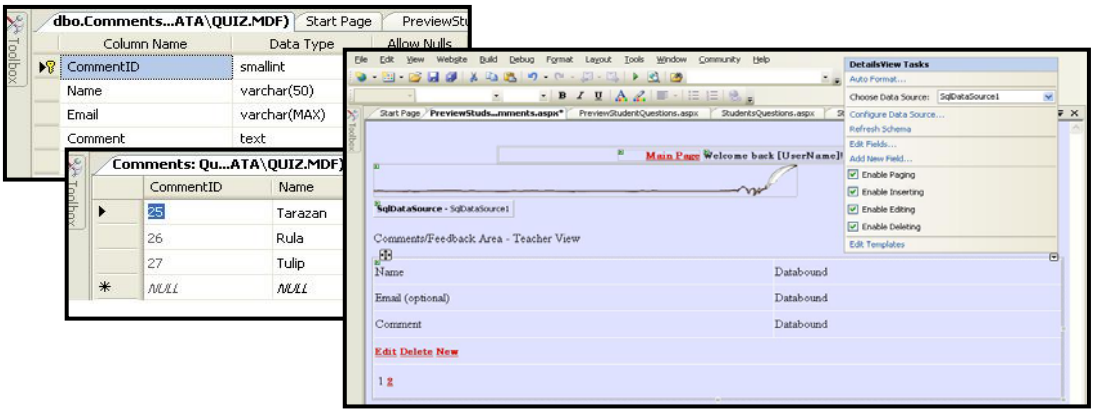


Figure 5.43 Comments/ Feedback area

5.6.2 The Questions Tool

The questions area is an important communication tool that keeps students in touch with a real teacher when they feel stuck, need help and support. Students can submit questions and view other students' questions to learn from. Teachers can view, delete and modify to answer the questions. Just like the comments communication tool above, we utilized the DetailsView Data tool to construct the Questions area tool. Figure 5.44 shows the Questions database table, the table content and the Details View interface.

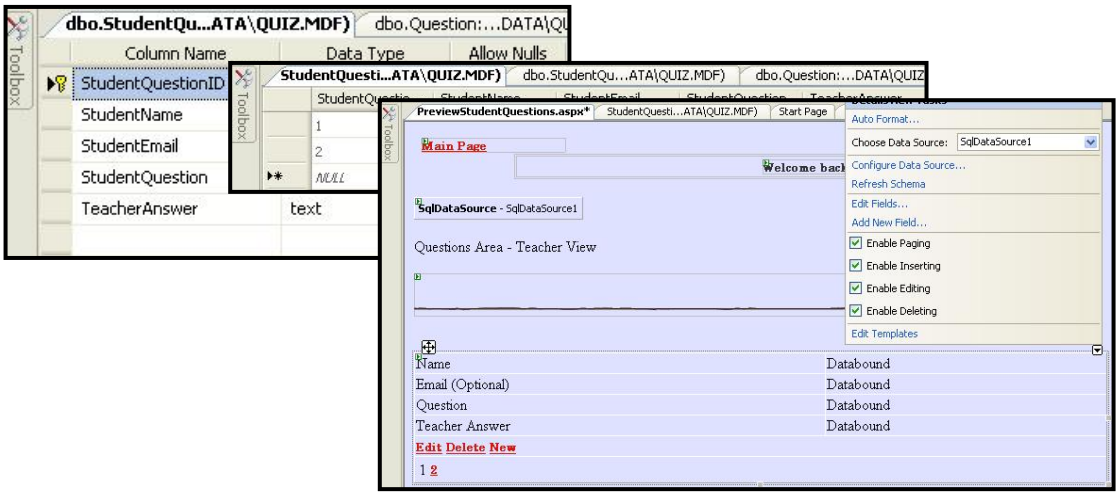


Figure 5.44: Constructing the student Questions Area

5.7 *Intelligent Practice based Instruction Tutoring System Development*

As mentioned earlier in this chapter and in Chapter 3, our online system contains two parts: the traditional learner controlled part and the intelligent practices part. The traditional part mainly consists of the lessons and their objectives. The online practices are the intelligent tutoring system part. Some tools like the reporting systems, evaluation and communication tools as explained earlier in this chapter can be used by and to support both parts.

The online intelligent practices system is an embedded rule-based expert system designed in a cognitive mechanism to imitate the human teacher in guiding students through their learning process. It adapts students levels through providing a sequence of questions based on students responses. This intelligent part is intended to help students identify and fill any gaps they have in their knowledge. The intelligent component can be accessed either from the main page (default page) or from the lessons page. It currently contains 10 different tutors. Nine tutors for the 9 chapters and each one of them is accessed from the specific chapter it relates to. The 10th is accessed from the main page including all module 1 chapters as shown in Figure 5.45. In the following subsections, we will talk in details about the system's architecture, algorithm, and implementation.

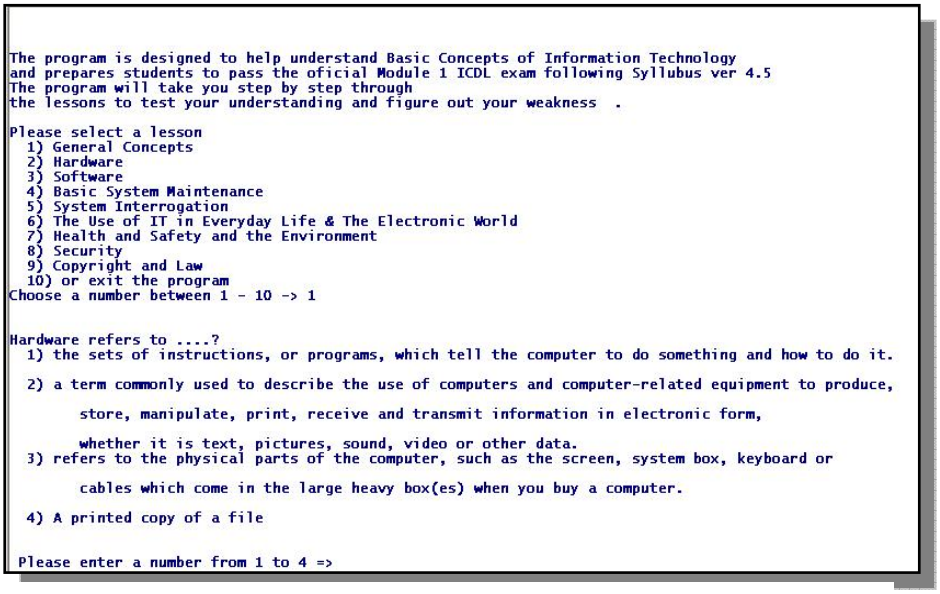


Figure 5.45: A snap shot for the full module intelligent practice

5.7.1 The Intelligent Practices Architecture

As illustrated in Figure 5.46, the intelligent lessons or practices consist of 5 main models: the Domain Model, the Student Model, the Tutor Model, the Interface Model, and the Evaluation model (Mandl and Lesgold 1988; Salgueiro et al 1998). In the following, we will explain each model in details.

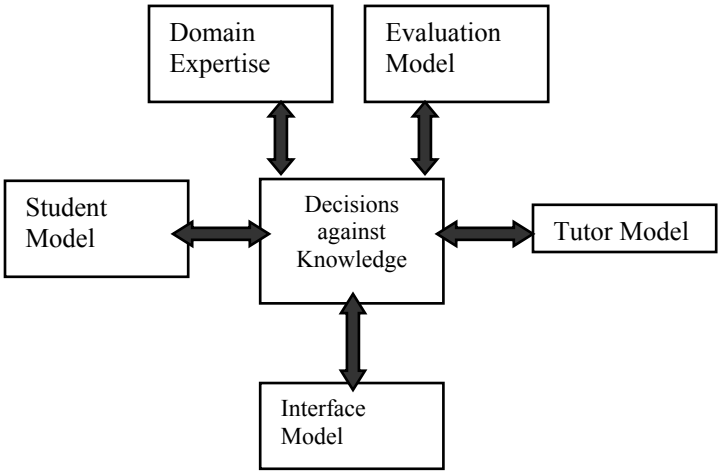


Figure 5.46: ITS system main components (Adapted from Mandl and Lesgold 1988; Salgueiro et al 1998)

5.7.1.1 Expert knowledge

Expert knowledge or the Domain model consists of the facts and questions of the practice based instruction intelligent tutoring system. A rule based expert system has been built for this purpose containing all the necessary rules and facts in terms of questions and answers, feedback, guidance and direction. The system facts and questions were built and reviewed with the help of a group of domain experts (IT/ICDL teachers) fulfilling the requirements of the standard ECDL/ICDL syllabus

for Module one. Figure 5.47 shows an example of a wrong answer with the feedback, guidance and the necessary information.

```
Which one of the following is not a function of an operating system...?
 1) The editing of text in a word processor
 2) Managing output to the printer.
 3) Managing memory allocation in RAM.
 4) Managing files and data on the hard drive

Please enter a number from 1 to 4 => 3


This is a wrong answer... the answer is: The editing of text in a word processor

Operating System is the heart of a computer system. It is essential to keep everything
flowing smoothly inside your computer. All the basic operations of a computer are controlled
by an operating system. It is the operating system that establishes the rule of interaction
between the hardware and software of your system. All operations such as storing, processing,
running applications and programs etc. are handled by an operating system.


What is operating system software?
 1) Software that controls the allocation and usage of hardware resources
 2) Software for carrying out spreadsheet charting and calculations
 3) Software for operating virus-scanning procedures
 4) Software for operating databases

Please enter a number from 1 to 4 => _
```

Figure 5.47: An example of a student answer and the system feedback

5.7.1.2 The Student Model

The Student Model is about the individual characteristics of the student, and consists of the system's ability to diagnose, and adapt the student's developing knowledge and skills. The system asks questions and depending on the student response, feedback is given, individual direction and a sequence of questions is provided and recommendations are specified. Figure 5.48 shows different branching schemes and different student models depending on student response and knowledge level.

```
Which one of the following is not a function of an operating system...?
 1) The editing of text in a word processor
 2) Managing output to the printer.
 3) Managing memory allocation in RAM.
 4) Managing files and data on the hard drive

Please enter a number from 1 to 4 => 3


This is a wrong answer... the answer is: The editing of text in a word processor

Operating System is the heart of a computer system. It is essential to keep everything
flowing smoothly inside your computer. All the basic operations of a computer are controlled
by an operating system. It is the operating system that establishes the rule of interaction
between the hardware and software of your system. All operations such as storing, processing,
running applications and programs etc. are handled by an operating system.
```

When a student weakness appears on one topic, more questions on the same topic are given after some nation

Figure 5.48

```
What is operating system software?
 1) Software that controls the allocation and usage o
 2) Software for carrying out spreadsheet charting an
 3) Software for operating virus-scanning procedures
 4) Software for operating databases

Please enter a number from 1 to 4 => _
```

With correct answer the system moves to the second topic giving a good feedback

```
Which one of the following is not Operating System Software?
 1) Windows XP
 2) Linux
 3) Microsoft Word
 4) Mac OS

Please enter a number from 1 to 4 => 3


This is Correct ..Well done..


What does GUI stand for?
 1) General User Interface
 2) Graphical User Interchange
 3) Graphical User Interface
 4) Global User Interface

Please enter a number from 1 to 4 => _
```

5.7.1.3 The teacher Model

The Teacher Model or the Tutorial Planning model works closely with the learner model to guide the student by presenting appropriate questions, providing customized feedback, determining progress, and tailoring individual recommendations. Figures 5.49 A and 5.49 B show and example of how teachers guide students differently depending on student level and how the instruction is customized.

```
Which one of the following is not a function of an operating system...?
 1) The editing of text in a word processor
 2) Managing output to the printer.
 3) Managing memory allocation in RAM.
 4) Managing files and data on the hard drive

Please enter a number from 1 to 4 => 3

This is a wrong answer... the answer is: The editing of text in a word processor

Operating System is the heart of a computer system. It is essential to keep everything
flowing smoothly inside your computer. All the basic operations of a computer are controlled
by an operating system. It is the operating system that establishes the rule of interaction
between the hardware and software of your system. All operations such as storing, processing,
running applications and programs etc. are handled by an operating system.

What is operating system software?
 1) Software that controls the allocation and usage of hardware resources
 2) Software for carrying out spreadsheet charting and calculations
 3) Software for operating virus-scanning procedures
 4) Software for operating databases

Please enter a number from 1 to 4 => _
```

Feedback and Lesson explanation for incorrect answers

More questions about the same topic to make sure the student understood the point

Figure 5.49 A – teacher feedback on a student response 1

```
CPU stands for....?
 1) Central progressing Unit
 2) Central Processing Unit
 3) Central Programming Unit
 4) Central Paging Unit

Please enter a number from 1 to 4 => 1

This is a wrong answer... CPU stands for Central Processing Unit

It is very important to understand and remeber abbreviations stands for what ...

It sounds that you need to work on this area better ...
I strongly suggest that you read and learn the lesson abbreviations  ..

This is lesson 2 :Hardware

Do you want to continue to the next topic and later read the lesson? ..
==> y or n  n
```

Further recommendations and test hints are given depending on students' model and

Figure 5.49 B – Teacher feedback on a student response 2

5.7.1.4 The Communication Model

The Communication model or the Interface Model can be described as a means of students' interaction with the system. This is through the student answering the multiple choice questions and the system providing the customized instruction and feedback. The ITS interface is simple use and interact with. It is straightforward to download, self-extracts and runs automatically. Figure 5.50 demonstrates an example of a question and answer

```
This is Lesson 2 - Hardware ...

 The three main parts of a CPU are....?
  1) ALU, immediate access memory, control unit
  2) ALU, RAM, control unit
  3) system box, control unit, VIDEO CARD
  4) ALU, ROM, control unit

 Please enter a number from 1 to 4 => 4

This is a wrong answer... CPU uses The immediate access memory ( where the data used by the ALU is stored)
So the right answer is: ALU, immediate access memory, control unit
```

The system asks the student to select the right answer by entering its number; feedback is given immediately after the student answer as a discussion between the student and the tutor

Figure 5.50- Example of a question and student answer in a simple straight forward MCQ

This model may also include the facility for the student and the instructor themselves to communicate with each other, through means of electronic capabilities like sending questions, comments and Emails which makes the online intelligent tutoring systems useful for web-based courses.

5.7.1.5 The Evaluation Model

The Evaluation module performs a general evaluation of the system and creates the necessary feedback to the teacher to help him/her modify teaching his/her methods. It is used also to enhance the ITS system questions and branching methodology, the assessment questions and the clarity of instruction for the students during classes. Furthermore and as mentioned before, this system has a good reporting system which plays the role of a basic tracking system. It helps teachers improve their teaching methodology and course content through monitoring the visited quizzes and assessment results. Students' progress can be analyzed by comparing pre-quizzes with post quizzes. Other reports are included to monitor lessons visited, and review quizzes analysis reports, and viewing students comments and feedback to enhance the regular lesson contents and the ITS system questions and methodology (questions sequencing and branching, and/or add more questions, better feedback).

5.7.2 The algorithm

As seen in the examples above, the intelligent practices were designed in an interactive fashion. The student is offered different routes including further practice in the area where weakness occurs or divert to a new question area. There are different practices covering different chapters and one covers all module chapters. First, the student selects the chapter if more than one is available. Then the system starts asking questions and a sequence of questions are built according to the student response as explained in the following figure. To accomplish this task a number of rules of the type IF/THEN are implemented. The system stops when either the student decides to quit and do some reading on a specific point as recommended or when the system covers all the lesson objectives through the questions asked. Figure 5.51 is a flowchart that explains the intelligent practices algorithm in details.

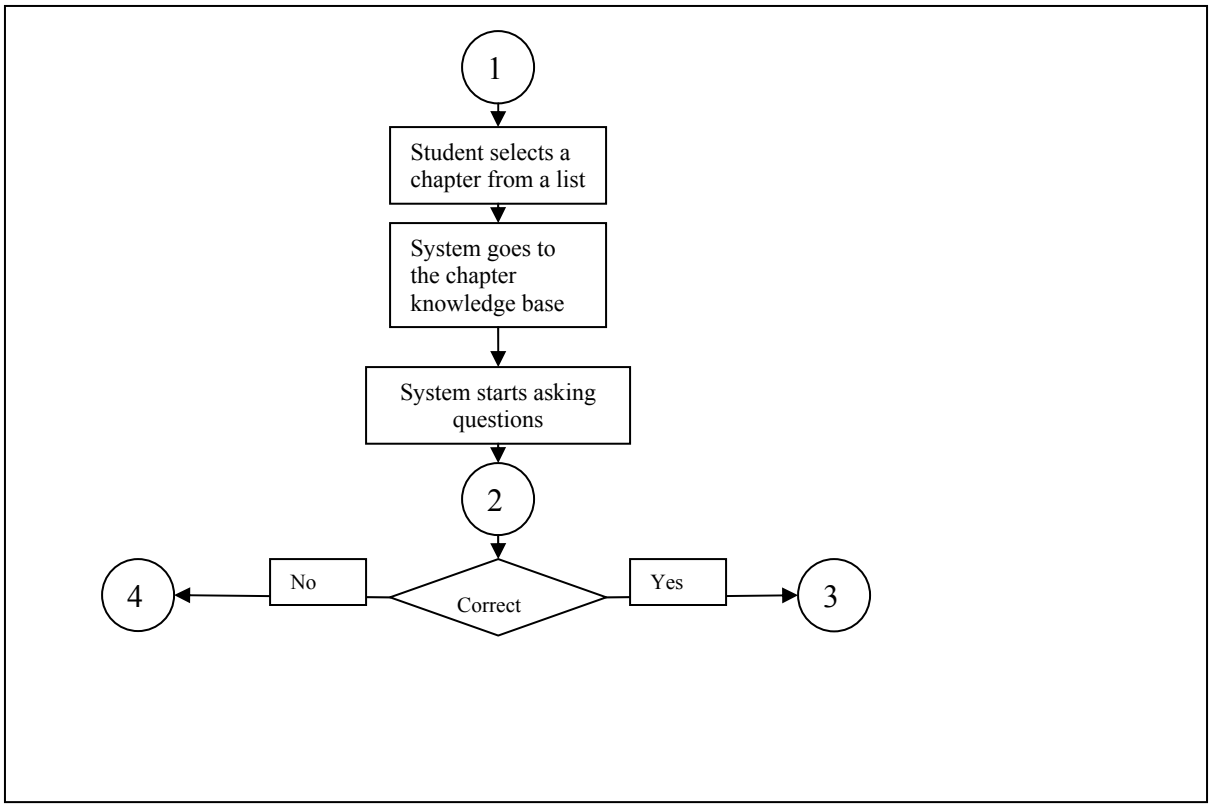


Figure 5.51-A: Practice based instruction Intelligent Tutoring System Algorithm

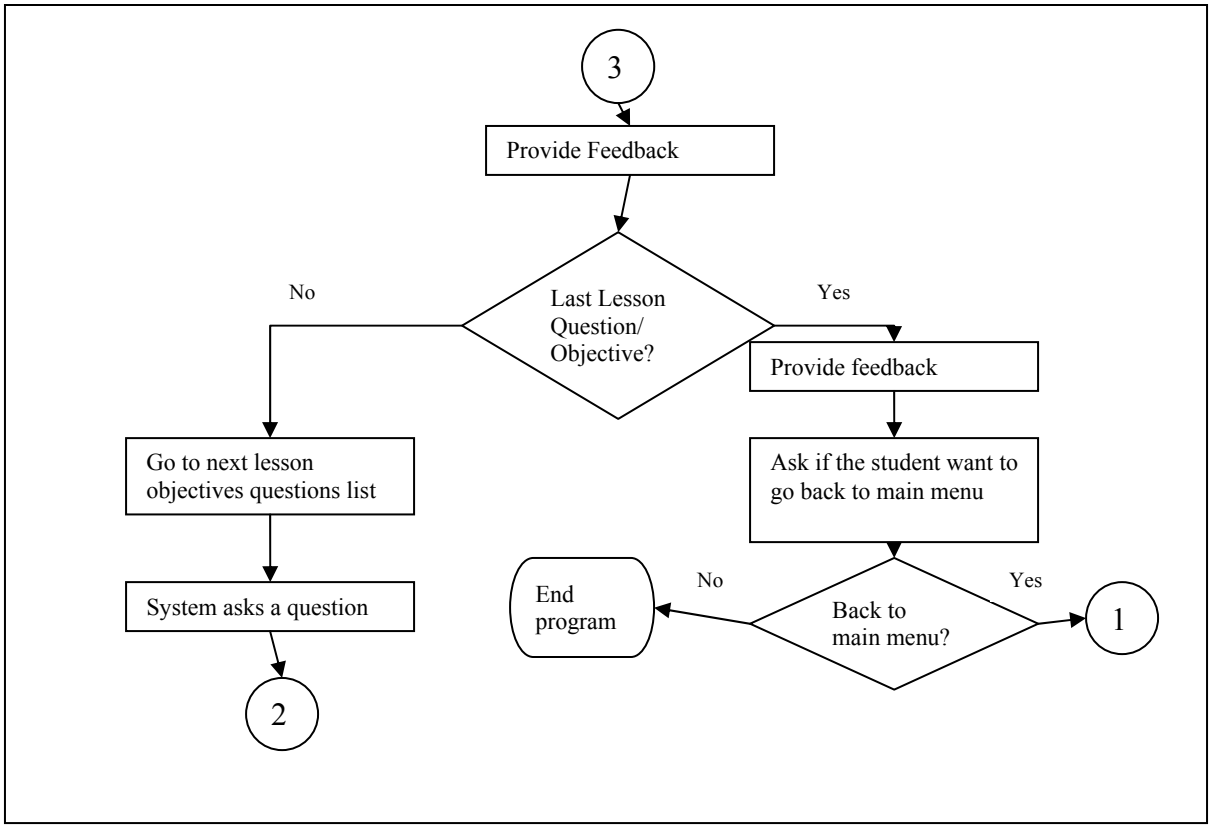


Figure 5.51-B: Practice based instruction Intelligent Tutoring System Algorithm

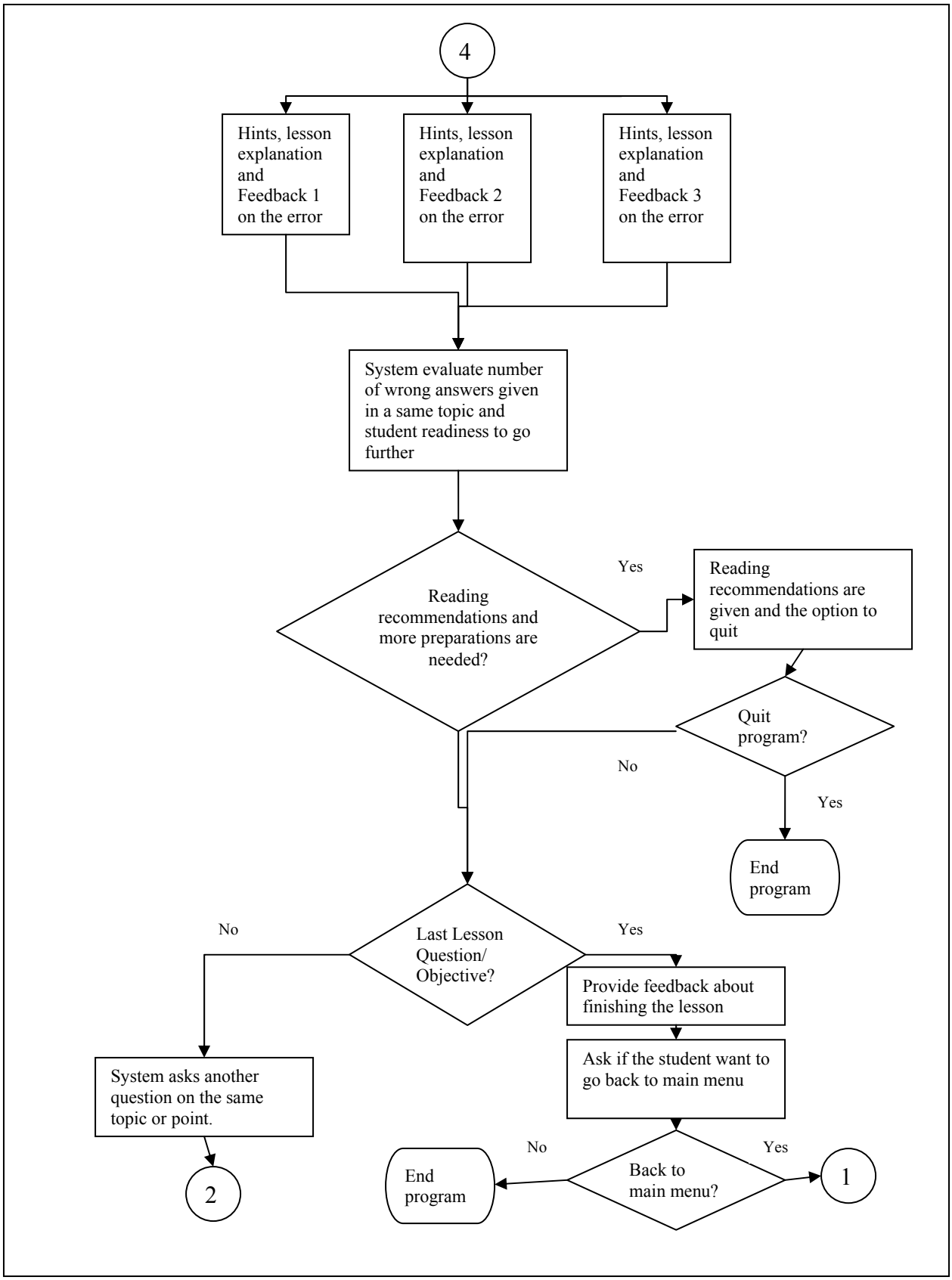


Figure 5.51-C: Practice based instruction Intelligent Tutoring System

5.7.3 Intelligent Practices Implementation

Each tutor has been developed as in the following steps. First, the CLIPS code was written. Then VC++ code was written to be used in creating a folder with all necessary files to run the tutor including an exe file for each tutor/lesson in a separate folder. After creating the exe file for each tutor, the folder was compressed using ZIP application. Next, the compressed file was converted to a self extracting file using "ZIP 2 Secure EXE" special software to make it easy for new users to access any of the tutors with least efforts. After that, we linked / added the self-extracting file to the website as a link in the relative web pages. Figure 5.52 show the coding sequence from CLIPS code till the final self extract exe file accessed from the website.

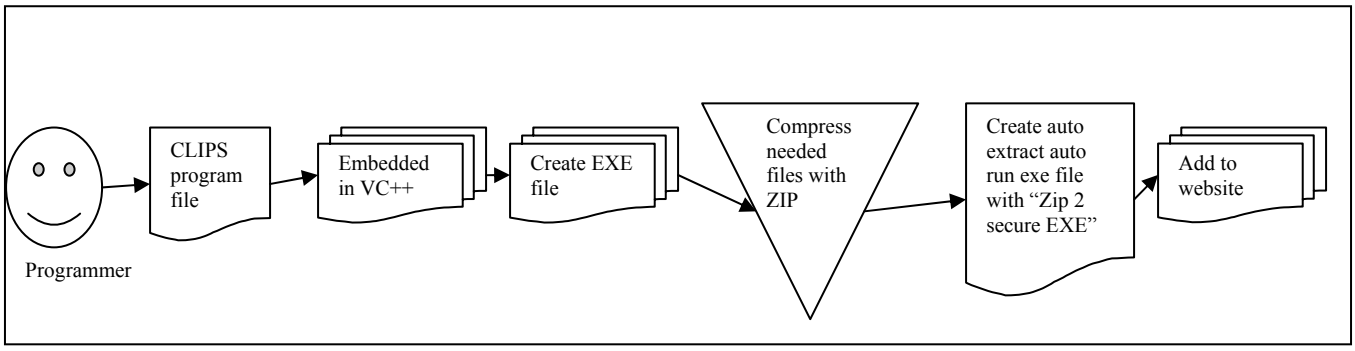


Figure 5.52: The steps sequence of creating the intelligent component and adding it to the website

5.7.3.1 The CLIPS - C Language Integrated Production System Code

To develop the intelligent part of the system, we considered necessary a rule-based programming language. As briefly introduced in Chapter 4, we considered CLIPS as our rule-based building tool. CLIPS is a public domain, productive -chaining rule-based language expert system tool. It provides a complete environment for the construction of rule and/or object based expert systems (NASA 1991; Martin et al 1992; Gonzalez et al 1993; Giarratano et al 1994). Figure 5.53 is an example of CLIPS syntax for asking a question. Then Figure 5.54 shows some examples of rule codes.

```
(defrule Lesson2-Question1-Question
 (declare (salience 1500))
 (Lesson number 2) =>
 (printout t crlf crlf
          " The three main parts of a CPU are....?" crlf

          "  1) ALU, immediate access memory, control unit " crlf
          "  2) ALU, RAM, control unit"crlf

          "  3) system box, control unit, VIDEO CARD" crlf

          "  4) ALU, ROM, control unit " crlf crlf
          " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer1 =(read))))
```

Figure 5.53 – CLIPS example for asking a question

```
(defrule Lesson2-Question1-answer1
 ?x <- (Lesson2 answer1 1) =>
 (retract ?x)


            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf

crlf )
(assert (Lesson2 question2 y))

)
```

```
(defrule Lesson2-Question1-answer2
 ?x <- (Lesson2 answer1 2) =>
 (retract ?x)
 (printout t crlf crlf
            "This is a wrong answer... CPU uses The immediate access memory (
where the data used by the ALU is stored) not RAM." crlf crlf
            "So the right answer is: ALU, immediate access memory, control unit" crlf
crlf)



(assert (Lesson2 question2 y))

)
```

```
(defrule Lesson2-Question1-answer3
 ?x <- (Lesson2 answer1 3) =>
 (retract ?x)
 (printout t crlf crlf
            "This is a wrong answer... The three main parts of a CPU are the arithmetic logic unit
(ALU), "crlf
            "the immediate access memory and the control unit.)." crlf crlf
            "So the right answer is: ALU, immediate access memory, control unit" crlf crlf)



(assert (Lesson2 question2 y)))
```

Figure 5.54- Different answers responses – rules codes

Figure 5.55 shows a sample discussion where the system generates a question and gives feedback according to the student response.

```
Which one of the following is not a function of an operating system...?
 1) The editing of text in a word processor
 2) Managing output to the printer.
 3) Managing memory allocation in RAM.
 4) Managing files and data on the hard drive

Please enter a number from 1 to 4 => 3


This is a wrong answer... the answer is: The editing of text in a word processor

Operating System is the heart of a computer system. It is essential to keep everything
flowing smoothly inside your computer. All the basic operations of a computer are controlled
by an operating system. It is the operating system that establishes the rule of interaction
between the hardware and software of your system. All operations such as storing, processing,
running applications and programs etc. are handled by an operating system.



What is operating system software?
 1) Software that controls the allocation and usage of hardware resources
 2) Software for carrying out spreadsheet charting and calculations
 3) Software for operating virus-scanning procedures
 4) Software for operating databases

Please enter a number from 1 to 4 => _
```

```
CPU stands for....?
 1) Central progressing Unit
 2) Central Processing Unit
 3) Central Programming Unit
 4) Central Paging Unit

Please enter a number from 1 to 4 => 1


This is a wrong answer... CPU stands for Central Processing Unit

It is very important to understand and remeber abbreviations stands for what ...

It sounds that you need to work on this area better ...
I strongly suggest that you read and learn the lesson abbreviations  ..

This is lesson 2 :Hardware

Do you want to continue to the next topic and later read the lesson? ..
==> y or n  n
```

Figure 5.55 Sample discussion

5.7.3.2 The Visual C ++ Code

As discussed in Chapter 4, CLIPS has been designed for full integration with other languages such as C. In fact, CLIPS can be called from a procedural language, perform its function, and then return control back to the calling program (Giarratano, 2002). In this system our CLIPS programs are called from Visual C++ .NET 2005 to be compiled and to create an EXE file.

Visual C++ provides a powerful and flexible development environment for creating Microsoft Windows–based and Microsoft .NET–based applications (Riley G. 2008). Figure 5.56 shows a screenshot sample VC++ code and Figure 5.57 shows a sample VC++ code that is utilized to run the CLIPS file.

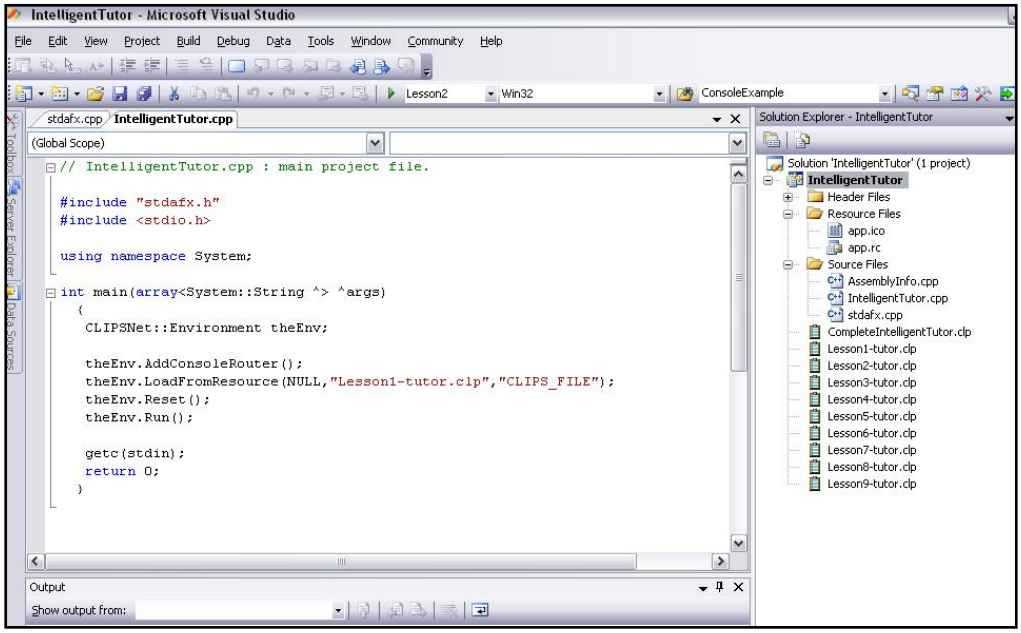


Figure 5.56- Screen shot sample for VC++ code

```
// IntelligentTutor.cpp : main project file.

#include "stdafx.h"
#include <stdio.h>

using namespace System;

int main(array<System::String ^> ^args)
  {
   CLIPSNet::Environment theEnv;

   theEnv.AddConsoleRouter();
   theEnv.LoadFromResource(NULL,"Lesson1-tutor.clp","CLIPS_FILE");
   theEnv.Reset();
   theEnv.Run();

   getc(stdin);
   return 0;
  }
```

Figure 5.57- Sample VC++ code to compile CLIPS files and create EXE files

This step is done to prepare all needed files and the exe file to be zipped and called and added to the website. Figure 5.58 shows the compilation and file building result. These are the files used by the website after being compressed.

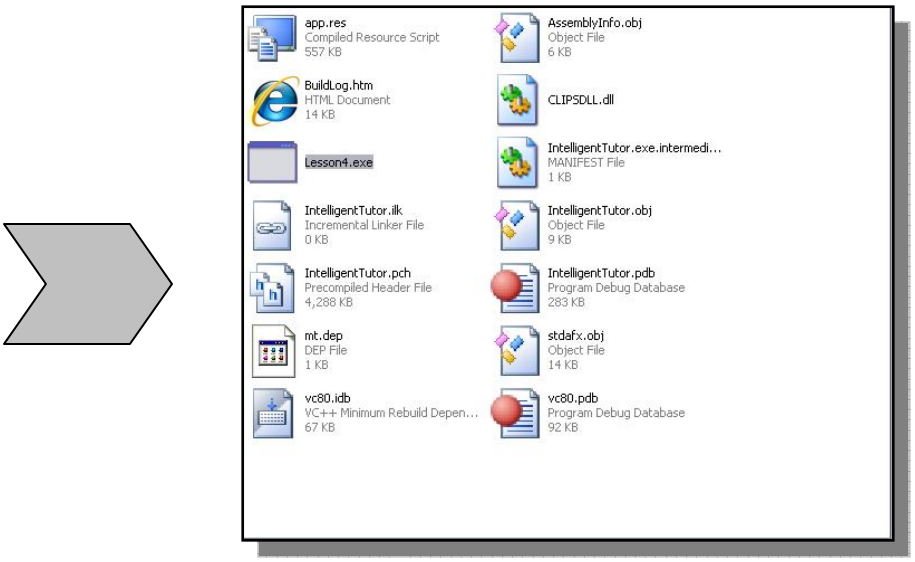


Figure 5.58 –example of the VC++ files of the Intelligent Practices

Figure 5.59 An example of an output screen as a result of running the exe file

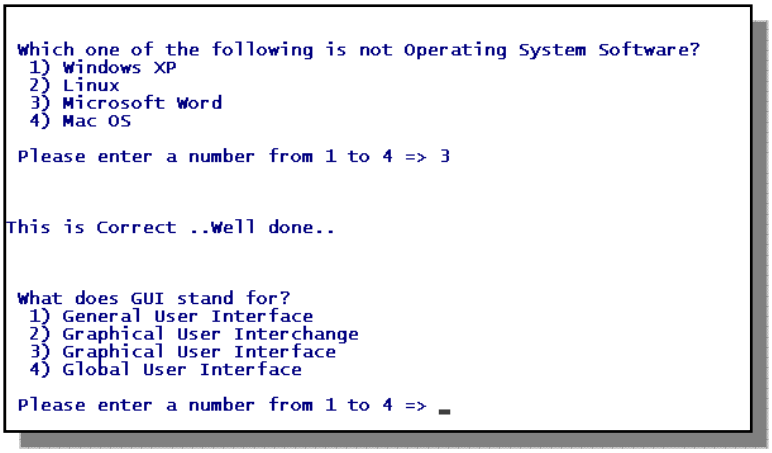


Figure 5.59 – Running the exe file

5.7.3.3 ZIP 2 Secure EXE

In our system, and as discussed in Chapter 4, we employed "ZIP 2 Secure EXE" as the utility program that creates self-extracting EXE files. This is to be able to add the file to the website in order to make it accessible for students online. Files were zipped using WinZip then and exe file was created using ZIP to Secure EXE program. (Chilkat Software 2008).Figure 5.60 shows what the student will see when they select to download the practice (intelligent tutor).

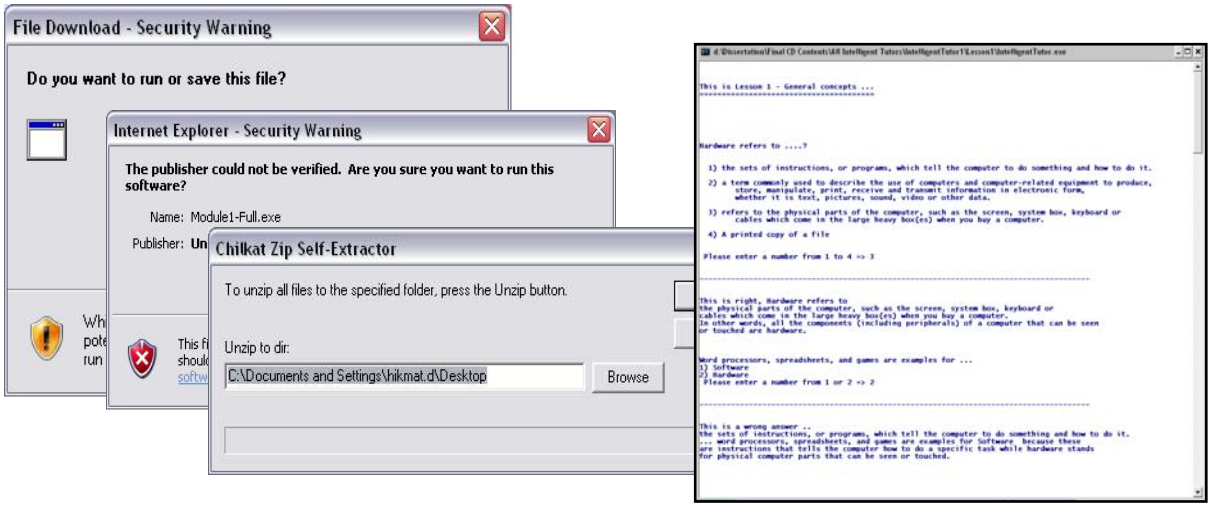


Figure 5.60– Running the exe self extract file

5.8 Other system components and features

In addition to the features discussed in this chapter, the system includes the change password feature which is designed for security and privacy reasons and the Site Map feature which is an added feature to ease navigation and to save time and effort by allowing the teachers to access system components directly. In the following subsections we will discuss these two features in more details.

5.8.1 Change Password

To maintain privacy and for security reasons, teachers and students are encouraged to change the passwords they are given. To make this possible, the change password facility is provided and can be accessed by all users through a link available in the main page. Figure 5.61 shows a snapshot of the change password facility.

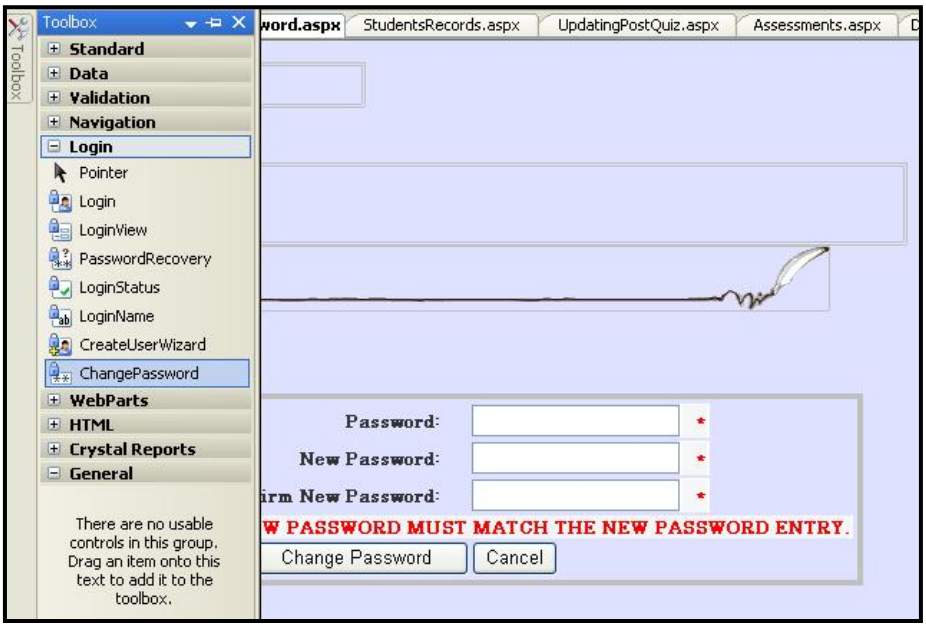


Figure 5.61: Change password facility

To provide this service, we have customized a form with the help of the change password login tool as in Figure 5.62. The system validated the current password with the one entered in the first box. It also validates that the entered new password matches the confirmed new password. Figure 5.63 demonstrate the validation massages they system may display. When the user changes his/her password, the change immediately is effective and reflects on the database.

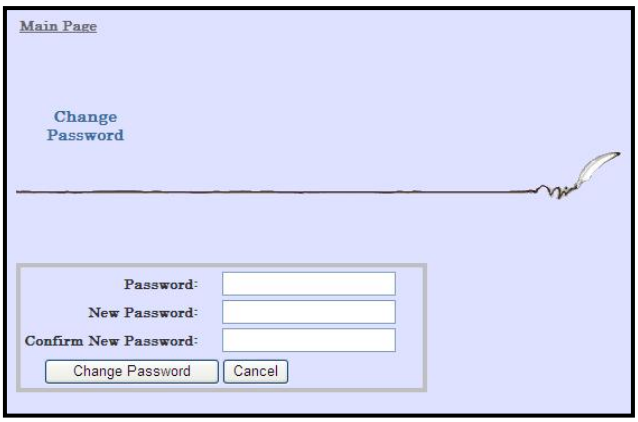


Figure 5.62: creating the change password form

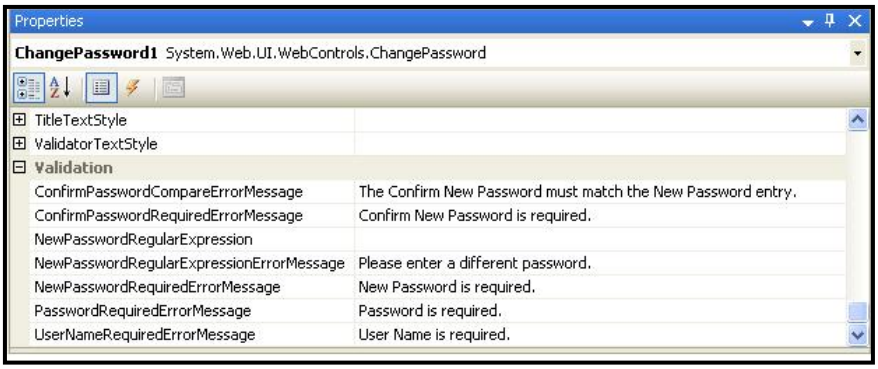


Figure 5.63: change password validation massages

5.8.2 The Site Map

The site map is accessible for teachers fro the main page to ease navigation and to save time and efforts. Figure 5.64 shows a snapshot of the sitemap.

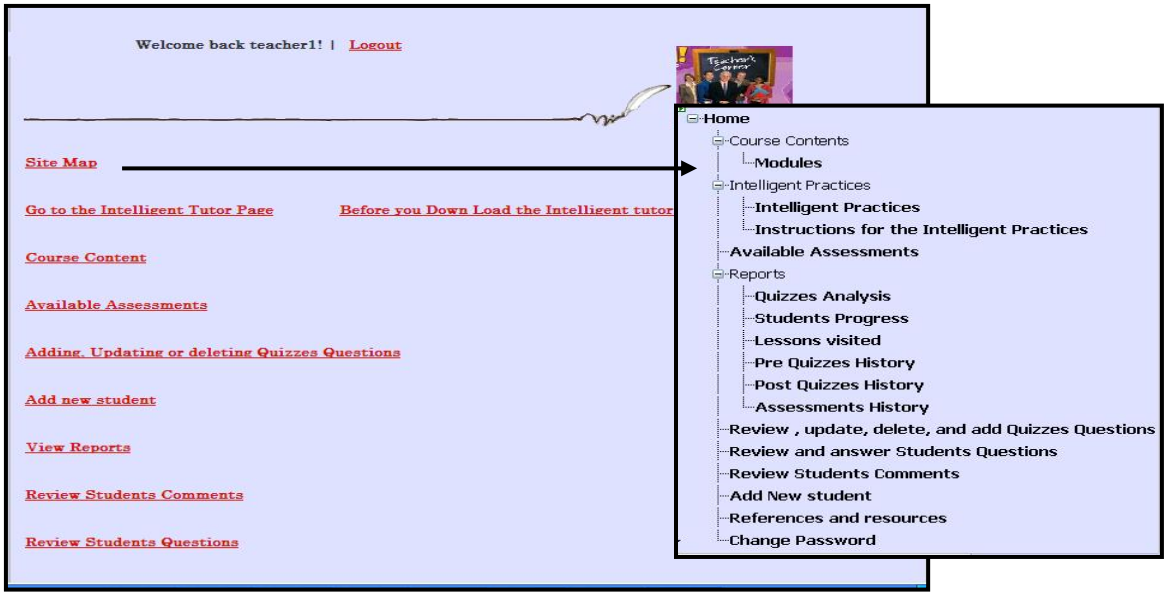


Figure 5.64: creating the change password form

To create the site map we utilized the tree view navigation tool as illustrated in Figure 5.65. Then we created the site map data source file "web.sitemap" which is an xml file that includes the site information as in Figure 5.66. As in the figure, each node has a title and a description besides a URL link if applicable.

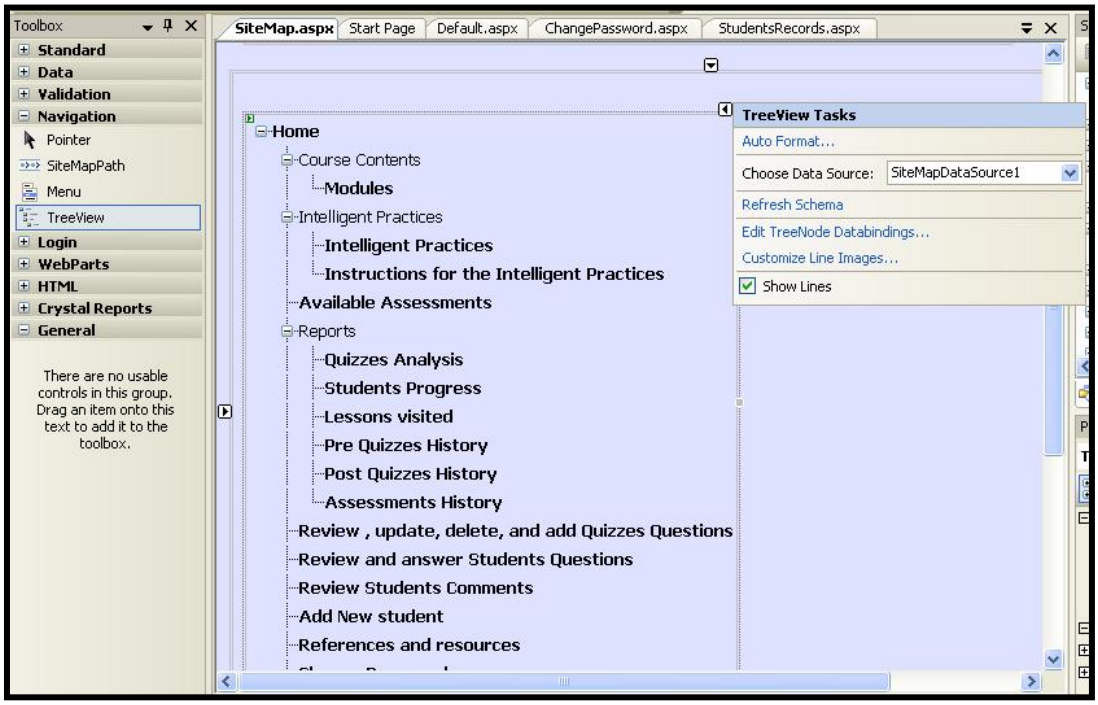


Figure 5.65: The tree view sitemap

```xml
xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="~/default.aspx" title="Home"  description=" First Page">
   <siteMapNode url="" title="Course Contents"  description="Course Contents" >
    <siteMapNode url="~/modules.aspx" title="Modules"  description="Available Modules" >
     <!--<siteMapNode url="~/Topic.aspx" title="Chapters"  description="Available Chapters" >
      <siteMapNode url="~/Lesson.aspx" title="Lessons"  description="Available Lessons" />
     </siteMapNode>-->
    </siteMapNode>
   </siteMapNode>
   <siteMapNode url="" title="Intelligent Practices"  description="Available Practice based ITS" >
   <siteMapNode url="~/IntelligentPractices.aspx" title="Intelligent Practices"  description="Available Practice based ITS" />
    <siteMapNode url="~/Readme.txt" title="Instructions for the Intelligent Practices"  description="Available Practice based ITS" />
   </siteMapNode>
    <siteMapNode url="~/test.aspx" title="Available Assessments"  description="Available Assessments" />
    <siteMapNode url="~/Reports.aspx" title="Reports"  description="Available Reports" >
    <siteMapNode url="~/QuizAnalysis.aspx" title="Quizzes Analysis"  description="Different Quizzes analysis" />
    <siteMapNode url="~/StudentsProgress.aspx" title="Students Progress"  description="Students Progress" />
    <siteMapNode url="~/Lessonsvisited2.aspx" title="Lessons visited"  description="Different Quizzes analysis" />
    <siteMapNode url="~/PreQuizHistory.aspx" title="Pre Quizzes History"  description="Pre Quizzes History" />
    <siteMapNode url="~/PostQuizHistory.aspx" title="Post Quizzes History"  description="Post Quizzes History" />
    <siteMapNode url="~/TestHistory2.aspx" title="Assessments History"  description="Test History" />
   </siteMapNode>
   <siteMapNode url="~/Assessments.aspx" title=" Review , update, delete, and add Quizzes Questions"  description="Modify Quizzes Questions" />
   <siteMapNode url="~/PreviewStudentQuestions.aspx" title=" Review and answer Students Questions"  description="Review and answer Students Questions" />
   <siteMapNode url="~/PreviewStudsentsComments.aspx" title=" Review  Students Comments"  description="Review  Students Comments" />
   <siteMapNode url="~/StudentsRecords.aspx" title="Add New student"  description="Add new students" />
   <siteMapNode url="~/Resources.txt" title="References and resources"  description="References and resources" />
   <siteMapNode url="~/ChangePassword.aspx" title="Change Password"  description="Change Password" />
  </siteMapNode>
 </siteMap>
```

Figure 5.66: The sitemap xml data source file

5.8.3 Adding Students

System Administrators can create and manage users, create, disable and manage roles in addition to create and manage access rules from the website administration tool. To make the system flexible, a tool is provided for the teachers. This tool enables teachers to add students with accessibility to the logged in default view.

To create this facility, we customized the create user wizard from the Login options tool as shown in figure 5.67. The system validates the entry data and requires a password that includes combination of alphabets and non-alphabets for security reasons. As shown in the figure all fields are mandatory. The system validates the accuracy of the confirmed password and that all fields are filled. If a field is not filled, the password entry doesn't meet the security requirements or the confirmed password is different than the password, an error massage is displayed. If all entry requirements are fulfilled then the new user data is added to the aspnet_users table in the database.

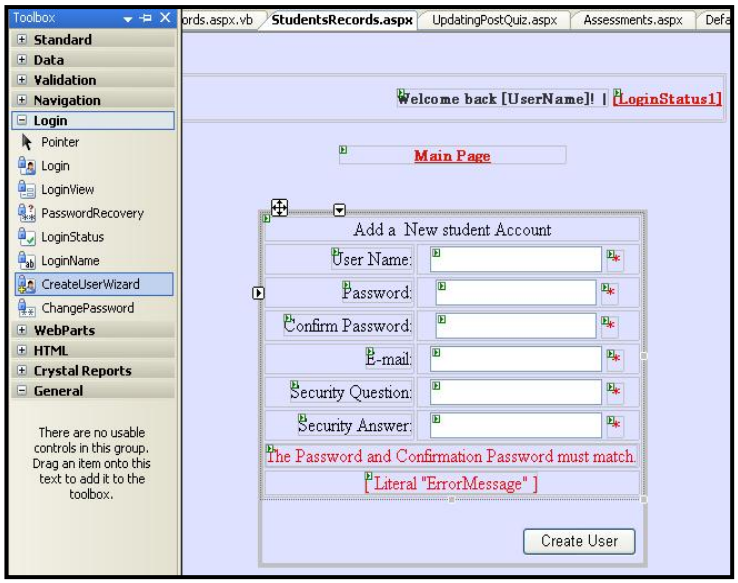


Figure 5.67: the create user wizard tool and the customized add student form

4.9 Conclusion

In this chapter, we discussed the system and its components architecture, algorithm and design in some details. The system is designed to meet different learning theories. This prototype system helps students to prepare for their ICDL/ECDL module 1 certification examination. Also, the system can be used to teach beginners the computer basic concepts and fundamentals. As illustrated above the system is an online/web-based system that can be used for distance learning or in classroom as an e-learning tool in the blended learning approach.

We have started this chapter by explaining the overall system architecture then started taking in details about the algorithm, design and implementation of each component making up the system. As explained in this chapter, this web-based system consists of two different parts: the student control traditional part which includes the different lessons and their objectives and the intelligent part which includes the adaptive, cognitive intelligent practice exercises. Also, the system includes many other components which can be used by both the traditional and the intelligent parts like the database, the reporting system, the evaluation tools and the communication tools. Furthermore, the system has two interfaces, the students and the teachers. Each interface is given different accessibility and permissions.

Chapter 6: System Evaluation

In this chapter we explain the testing methodology and discuss results. We ran many tests to examine the application effectiveness. Tests had covered different areas like technicality and effectiveness of the intelligent part of it. To achieve our goals in this phase we have used several tools and programs such as Microsoft Excel for calculations and charting, Microsoft SQL Server 2005, Microsoft ASP .NET 2005, besides online T-test calculators.

6.1 Technical testing

The technical testing phase has been done to ensure that all website links and functionalities are working fine. The website project was given to 3 different ICDL certified IT teachers who have been asked to go through it, explore it and give feedback. In case of any problems found an excel file was given to fill out. The excel file contained 3 different sheets are for the errors related to links and the functionality, the errors related to quizzes and assessment and the third one are for the errors related to lessons. Figures 6.1 through 6.3 show different views of the excel file sheets.

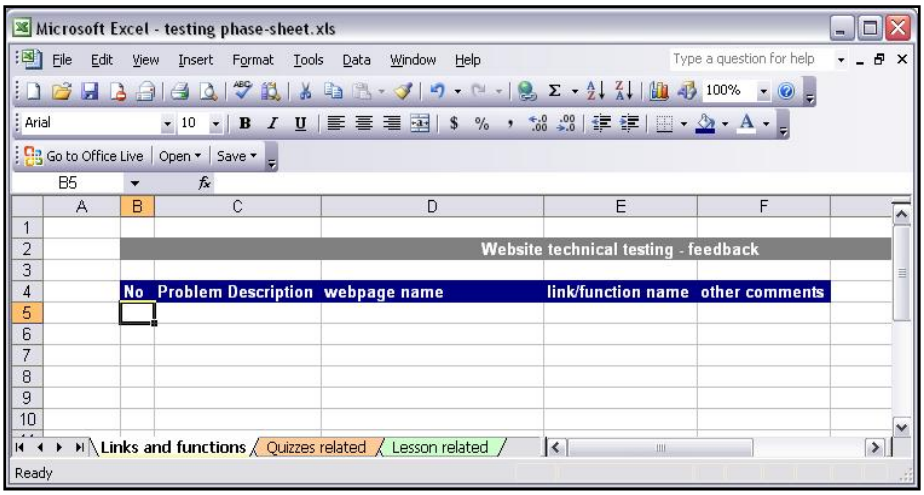


Figure 6.1: Errors related to links and functions sheet

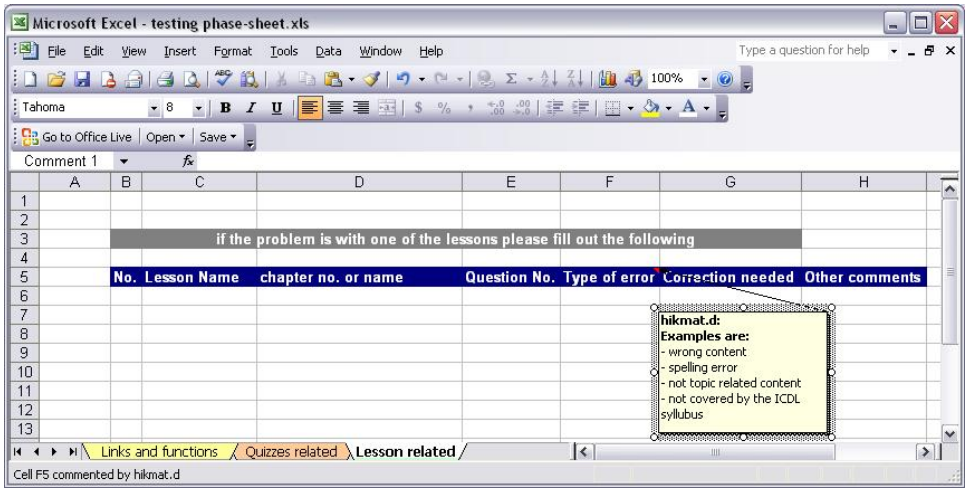


Figure 6.2: Errors related to lessons sheet

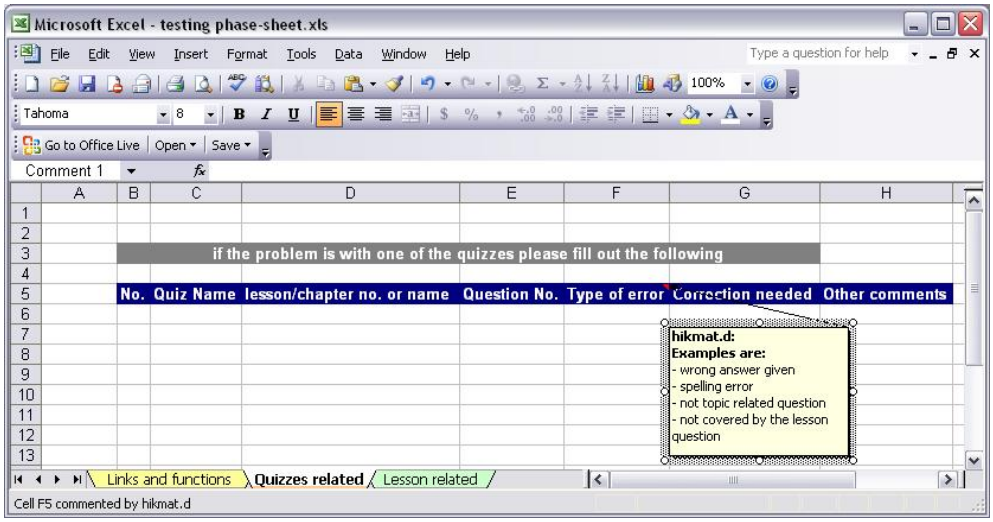


Figure 6.3: Errors related to Quizzes sheet

After getting the teachers feedback, corrections were made and another testing was carried out to ensure technical quality.

6.2 Website theory and content testing

6.2.1 Running the test:

To test the practice based instruction intelligent tutoring system, two different groups of students and teachers were formed as can be seen in the following table (Figure 6.4).

No.	Group	Student	Educational level	Lesson1	Lesson2	Lesson3
1	Group1	Student1	Grade 6	1/7/08	1/8/08	1/9/08
2	Group1	Student2	English Teacher	1/7/08	1/8/08	1/9/08
3	Group1	Student3	Grade 6	1/13/08	1/14/08	1/20/08
4	Group1	Student4	Grade 9	1/13/08	1/14/08	1/20/08
5	Group1	Student5	Science Teacher	1/13/08	1/14/08	1/20/08
6	Group1	Student6	Grade 11	1/21/08	1/22/08	1/23/08
7	Group1	Student7	Grade 11	1/21/08	1/22/08	1/23/08
8	Group1	Student8	Grade 9	1/27/08	1/28/08	1/29/08
9	Group1	Student9	Grade 9	1/27/08	1/28/08	1/29/08
10	Group2	Student1	Grade 6	1/7/08	1/8/08	1/9/08
11	Group2	Student2	Math Teacher	1/7/08	1/8/08	1/9/08
12	Group2	Student3	Grade 6	1/13/08	1/14/08	1/20/08
13	Group2	Student4	Grade 9	1/13/08	1/14/08	1/20/08
14	Group2	Student5	Biology Teacher	1/13/08	1/14/08	1/20/08
15	Group2	Student6	Grade 11	1/21/08	1/22/08	1/23/08
16	Group2	Student7	Grade 11	1/21/08	1/22/08	1/23/08
17	Group2	Student8	Grade 11	1/21/08	1/22/08	1/23/08
18	Group2	Student9	Grade 9	1/27/08	1/28/08	1/29/08

Figure 6.4: information about the 2 different test groups

Students were selected with the help of their teacher and were average students who could read, write and understand English. They were attending American curriculum schools. As in the sheet the study included adults by asking some volunteer teachers to participate.

The website has been deployed on 4 computers and each group went through the website according to the plan arranged for them. Group 1 went through the website as follows. Every student in the group had to do the first three chapters/ lessons. They started with the pre-test, then read the lessons again (time was extended according to their needs and pace of learning) and after they finished the lessons they completed the post-test. The Group 2 students did the pre-test, read the lessons (time was flexible according to their needs and pace of learning), then went through the cognitive, adaptive practice based ITS part. After that they completed the post-test. Results were as in Figure 6.5 and Figure 6.6:

For Group 1

UserName	TopicID	pre quiz Score	Post Quiz Score	Progress
Group1Student1	1	40	60	20
Group1Student1	2	33.333	66.667	33.334
Group1Student1	3	30	60	30
Group1Student2	1	70	90	20
Group1Student2	2	80	100	20
Group1Student2	3	60	100	40
Group1Student3	1	10	40	30
Group1Student3	2	65	75	10
Group1Student3	3	30	80	50
Group1Student4	1	40	50	10
Group1Student4	2	33.333	41.667	8.334
Group1Student4	3	20	80	60
Group1Student5	1	58.333	83.333	25
Group1Student5	2	56.25	91.667	35.417
Group1Student5	3	90	100	10
Group1Student6	1	50	70	20
Group1Student6	2	33.333	58.333	25
Group1Student6	3	40	80	40
Group1Student7	1	40	80	40
Group1Student7	2	58.333	83.333	25
Group1Student7	3	80	100	20
Group1Student8	1	60	80	20
Group1Student8	2	58.333	83.333	25
Group1Student8	3	70	80	10
Group1Student9	1	40	80	40
Group1Student9	2	58.333	91.667	33.334
Group1Student9	3	60	90	30

Figure 6.5: Group1 results

For group 2

UserName	TopicID	pre quiz Score	Post Quiz Score	Progress
Group2Student1	1	40	80	40
Group2Student1	2	33.333	83.333	50
Group2Student1	3	30	80	50
Group2Student2	1	70	90	20
Group2Student2	2	50	83.333	33.333
Group2Student2	3	60	90	30
Group2Student3	1	10	60	50
Group2Student3	2	66.667	81.818	15.151
Group2Student3	3	30	70	40
Group2Student4	1	40	50	10
Group2Student4	2	33.333	81.818	48.485
Group2Student4	3	20	80	60
Group2Student5	1	50	83.333	33.333
Group2Student5	2	58.333	91.667	33.334
Group2Student5	3	80	90	10
Group2Student6	1	40	80	40
Group2Student6	2	33.333	75	41.667
Group2Student6	3	30	70	40
Group2Student7	1	50	80	30
Group2Student7	2	66.667	83.333	16.666
Group2Student7	3	60	80	20
Group2Student8	1	20	60	40
Group2Student8	2	58.333	75	16.667
Group2Student8	3	20	70	50
Group2Student9	1	60	90	30
Group2Student9	2	58.333	100	41.667
Group2Student9	3	80	100	20

Figure 6.6: Group 2 results

6.2.2 Analysis

In Figure 6.7, Chart 1 shows group 1 results while in Figure 6.8, Chart 2 shows group 2 results. The two charts show that both groups have benefited from the system while Group2 showed better results. This comparison between the two groups shows the importance of the intelligent practice exercises in enhancing students learning.

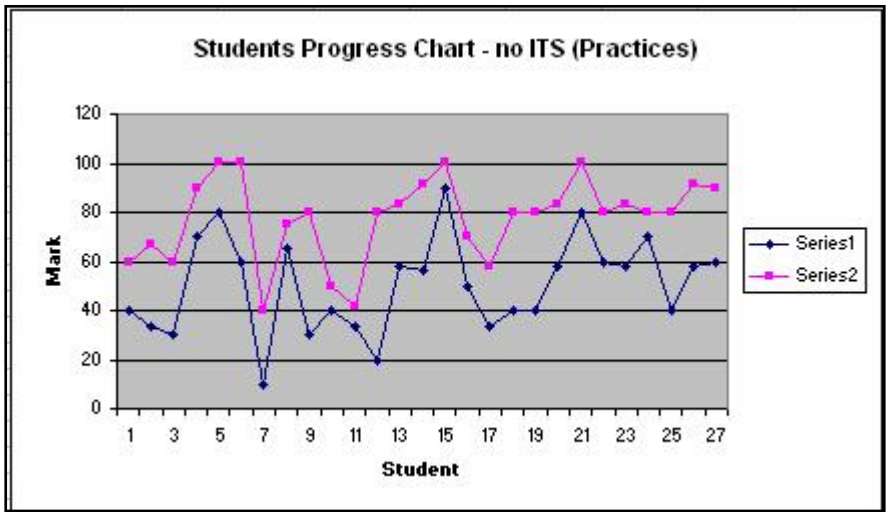


Figure 6.7- Chart1- Practice based instruction ITS was NOT included

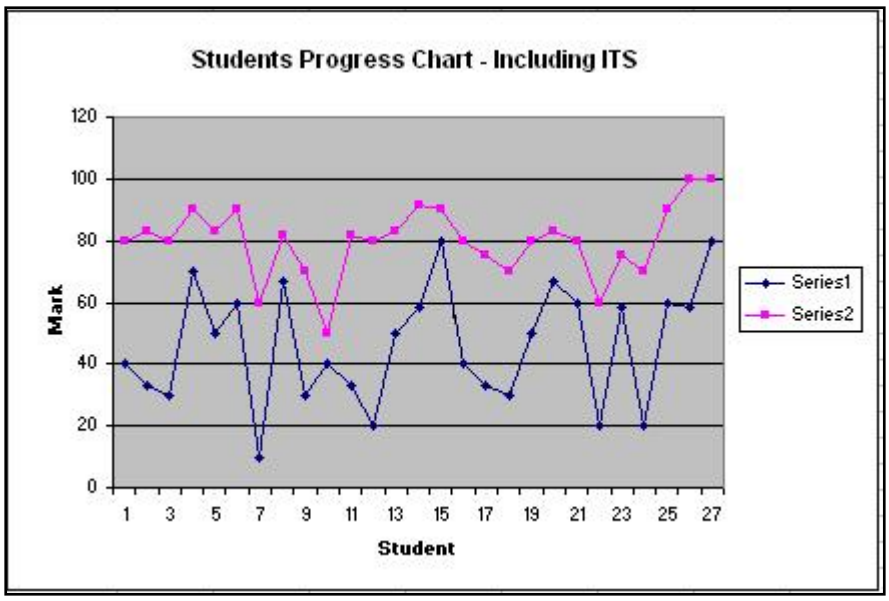


Figure 6.8- Chart 2- Practice based instruction ITS was included

To make the study more accurate we have done separate calculations for each lesson as follows. Figure 6.9 shows the results per chapter for group 1 and Figure 6.10 shows the results per chapter for group 2. Following that, we performed a comparison between the two groups' results as in Figure 6.11. The calculation results and the comparison suppose the same result as above which is that both groups have benefited from the system but Group 2 showed better results and progress. This again emphasize on the important of the intelligent part of the system in enhancing education and learning.

UserName	TopicID	pre quiz Score	Post Quiz Score	Progress
Group1Student1	1	40	60	20
Group1Student2	1	70	90	20
Group1Student3	1	10	40	30
Group1Student4	1	40	50	10
Group1Student5	1	58.333	83.333	25
Group1Student6	1	50	70	20
Group1Student7	1	40	80	40
Group1Student8	1	60	80	20
Group1Student9	1	40	80	40
	Average	45.37033333	70.37033333	25
Group1Student1	2	33.333	66.667	33.334
Group1Student2	2	80	100	20
Group1Student3	2	65	75	10
Group1Student4	2	33.333	41.667	8.334
Group1Student5	2	56.25	91.667	35.417
Group1Student6	2	33.333	58.333	25
Group1Student7	2	58.333	83.333	25
Group1Student8	2	58.333	83.333	25
Group1Student9	2	58.333	91.667	33.334
	Average	52.91644444	76.85188889	23.93544444
Group1Student1	3	30	60	30
Group1Student2	3	60	100	40
Group1Student3	3	30	80	50
Group1Student4	3	20	41.667	21.667
Group1Student5	3	90	100	10
Group1Student6	3	40	80	40
Group1Student7	3	80	100	20
Group1Student8	3	70	80	10
Group1Student9	3	60	90	30

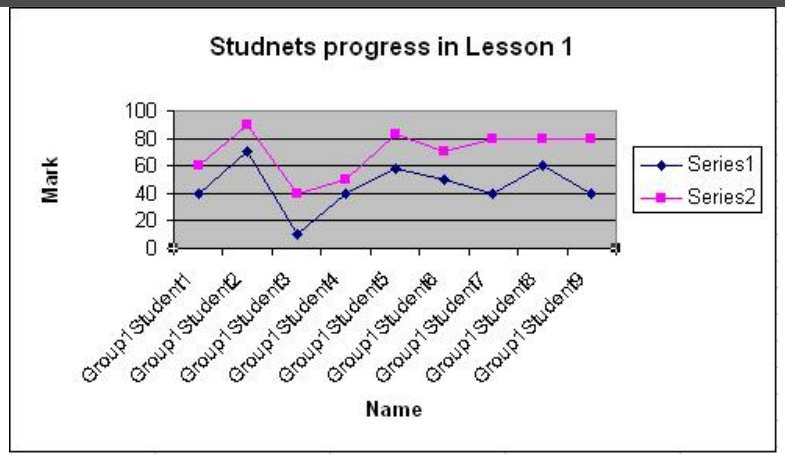
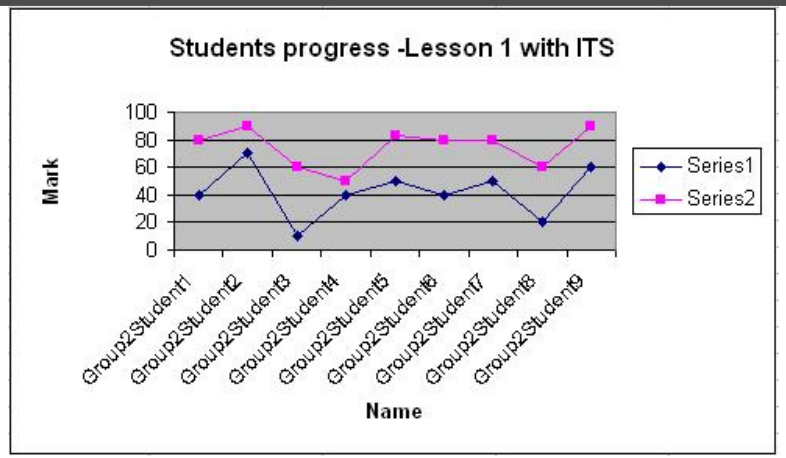
	Average	53.33333333	81.29633333	27.963

Figure 6.9- Results per chapter for group 1

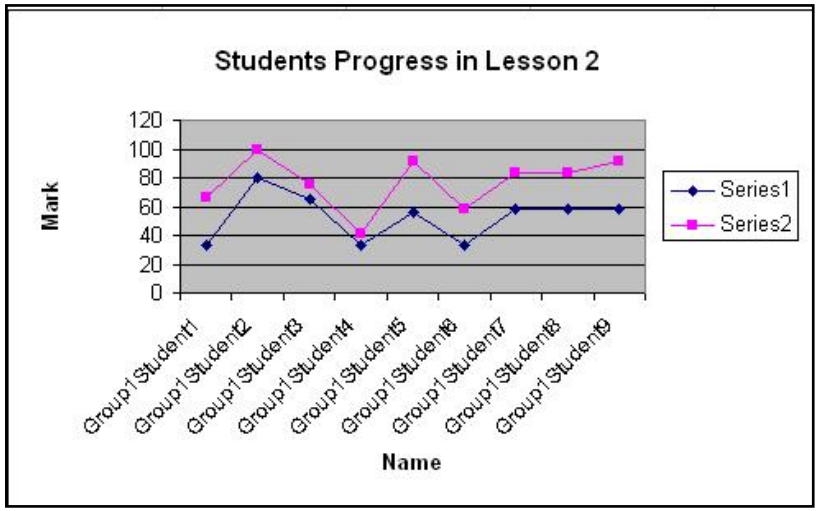
UserName	TopicID	pre quiz Score	Post Quiz Score	Progress
Group2Student1	1	40	80	40
Group2Student2	1	70	90	20
Group2Student3	1	10	60	50
Group2Student4	1	40	50	10
Group2Student5	1	50	83.333	33.333
Group2Student6	1	40	80	40
Group2Student7	1	50	80	30
Group2Student8	1	20	60	40
Group2Student9	1	60	90	30
Average		42.22222222	74.81477778	32.59255556
Group2Student1	2	33.333	83.333	50
Group2Student2	2	50	83.333	33.333
Group2Student3	2	66.667	81.818	15.151
Group2Student4	2	33.333	81.818	48.485
Group2Student5	2	58.333	91.667	33.334
Group2Student6	2	33.333	75	41.667
Group2Student7	2	66.667	83.333	16.666
Group2Student8	2	58.333	75	16.667
Group2Student9	2	58.333	100	41.667
Average		50.92577778	83.92244444	32.99666667
Group2Student1	3	30	80	50
Group2Student2	3	60	90	30
Group2Student3	3	30	70	40
Group2Student4	3	20	80	60
Group2Student5	3	80	90	10
Group2Student6	3	30	70	40
Group2Student7	3	60	80	20
Group2Student8	3	20	70	50
Group2Student9	3	80	100	20
Average		45.55555556	81.11111111	35.55555556

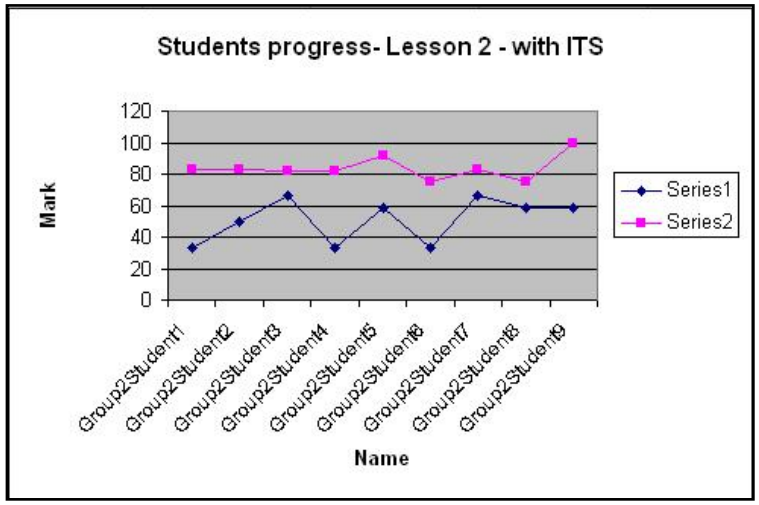
Figure 6.10- Results per chapter for group 2

Comparison between the two groups' results

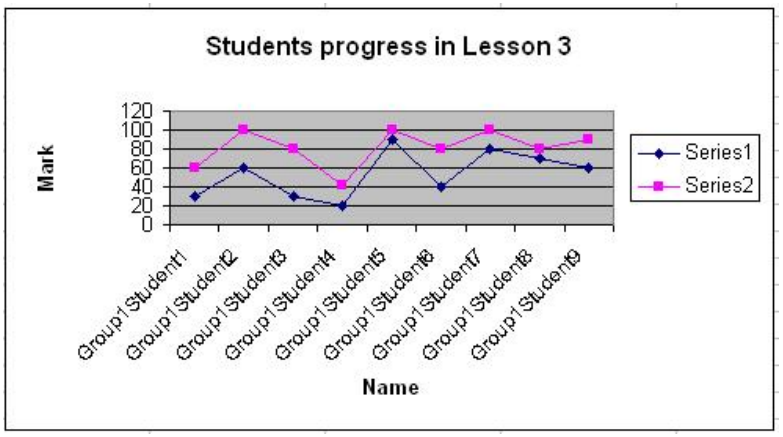
	With out using the practice based instruction ITS	With using the practice based instruction ITS
Lesson 1		

esson 2





Lesson 3



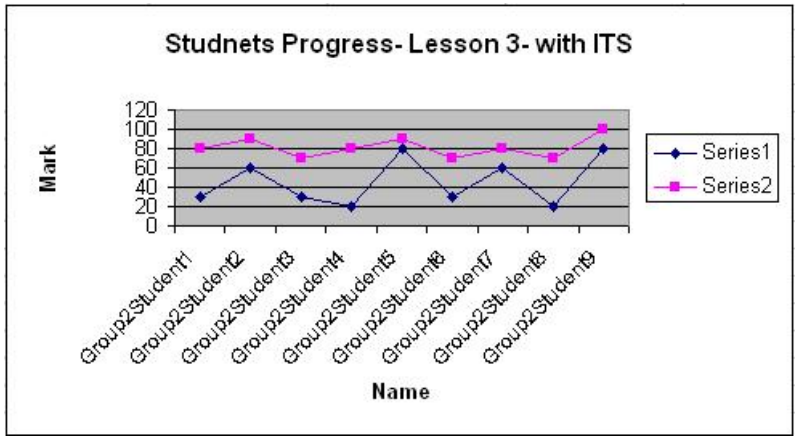


Figure 6.11- Comparison between the 2 groups' results

6.3 Results

To make sure that the results are acceptable, we ran a significance of data test. Then, we carried out some comparisons and other calculations as in the following subsections.

6.3.1 Significance of data:

Since the data is considered small, a student's t-test for each group data was done to determine the accuracy and statistical significance of the data.

What is a student's t-test?

A student's *t*-test deals with the problems associated with inference based on "small" samples student's-'t' test is probably still the most popular of all statistical tests. The test compares two mean (average) values to judge if they are different or not. The Student's – 't' test is the most sensitive test for interval data, but it also requires the most stringent assumptions. The variables/data are assumed to be normally distributed. If there is any reason to doubt this assumption, another distribution-free, test should be used such as Wilcoxon Test (Sivagnanam).

Student's t-test calculations

Using an online "t-test" calculator from GraphPad software the following calculations/results were obtained: Figure 6.12 illustrates the first group results and Figure 6.13 describes the result analysis for group one. Following, Figure 6.14 illustrates the second group results and Figure 6.15 shows the second group results analysis. From these t-test results, we realize that data is extremely statistically significant.

First group (No ITS) results:

	Two tailed P	statistical significance	Confidence interval	t	df	Standard error of difference
Chapter 1	**0.0001**	**extremely statistically significant**	**-32.68668 to -17.31332**	**7.5000**	**8**	**3.333**
Chapter 2	**0.0001**	**extremely statistically significant**	**-40.04800 to -20.04511**	**6.9277**	**8**	**6.9277**
Chapter 3	**0.0003**	**extremely statistically significant**	**-38.60043 to -17.32557**	**6.0619**	**8**	**4.613**

Figure 6.12- First group results calculations

Data analysis details for the first group

For Group 1 Chapter 1

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	47.73122	77.77778
SD	13.86186	13.81945
SEM	4.62062	4.60648
N	9	9

For Group 1 Chapter 2

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	47.73122	77.77778
SD	13.86186	13.81945
SEM	4.62062	4.60648
N	9	9

For Group 1 Chapter 3

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	53.33333	81.29633
SD	24.49490	19.85666
SEM	8.16497	6.61889
N	9	9

Figure 6.13- First group data analysis

Second group (with ITS) results:

	Two tailed P	statistical significance	Confidence interval	t	df	Standard error of difference
Chapter 1	**0.0001**	**extremely statistically significant**	**-41.81104 to -23.37407**	**t = 8.1530**	**8**	**3.998**
Chapter 2	**0.0001**	**extremely statistically significant**	**-43.63167 to -22.36166**	**7.1547**	**8**	**4.612**
Chapter 3	**0.0010**	**very statistically significant**	**-54.44 to -16.68**	**3.9922**	**16**	**8.906**

Figure 6.14- Second group results calculations

Data analysis details for the second group

For Group 2 Chapter 1

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	42.22222	74.81478
SD	18.55921	14.44442
SEM	6.18640	4.81481
N	9	9

For Group 2 Chapter 2

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	50.92578	83.92244
SD	14.09587	7.80693
SEM	4.69862	2.60231
N	9	9

For Group 2 Chapter 3

Quiz	**Pre-Quiz**	**Post Quiz**
Mean	45.56	81.11
SD	24.55	10.54
SEM	8.18	3.51
N	9	9

Figure 6.15- Second group data analysis

6.3.2 Calculating Averages and standard deviation:

In Figure 6.16 we illustrate the average progress for each chapter. Progress for each student is calculated in the tables above as shown in Figure 6.9 and Figure 6.10 by means of the equation: Progress = Post-quiz result - Pre-quiz result. At the end the average progress score is calculated as:

Average progress = Progress made by all students / number of students.

Group/Ave Progress	Chapter 1	Chapter 2	Chapter 3	Average
Group 1	**25**	**23.93544444**	**27.963**	**25.63281481**
Group 2	**32.59255556**	**32.99666667**	**35.55555556**	**33.71492593**
Difference	**7.59255556**	**9.06122223**	**7.59255556**	**8.08211112**

Figure 6.16- Average progress calculation for both groups

From Figure 6.16, we deduce that the average progress with group 2 (the group that have used the practice based instruction ITS) was about 8.1% higher than group 1 progress.

Group/standard deviation	Chapter 1	Chapter 2	Chapter 3	Average
Group 1	**10**	**9.778682**	**13.83878**	**11.20582**
Group 2	**11.99279**	**13.83562**	**16.66667**	**14.16503**
Difference	**1.992794**	**4.056935**	**2.827887**	**2.959205**

Figure 6.17- standard deviation calculation

While from Figure 6.17, we found that the average standard deviation learning gained using the ITS system is about 3 points higher than students went through traditional online learning. This result is a very promising step towards enhancing education and learning.

6.4 Testing phase challenges

To do the testing we faced many challenges. Some of these were the availability of English speaking volunteer students, lack of continued parents support during the experiment time, restricted available time which made students rush and did not get the full benefit of the system. In many cases smart guesses were made during the pre-test which made it difficult to measure the real learning curve and students' commitment, sincerity and seriousness.

To overcome these challenges we talked to school teachers to help us in the selection and to motivate students and parents.

6.5 Conclusion

In this chapter, we explained the testing phase. We performed different testing procedures to test the system effectiveness and efficiency. At first we carried out a technical test to reveal and fix any technical problems. Then, we carried out the content and system effectiveness test. This test measured the system effectiveness on general and the effect of the intelligent part on the system in particular. To run the test, we included two groups of students. These groups were distributed evenly from different school grades besides some non-IT teachers who presented the adults group of age.

The first group was given the traditional part of the system only (excluding the intelligent part) in a student centric approach while IT teachers were around as facilitators only. This

group was asked to carry out the pre-quiz before they start with the lessons to measure their previous knowledge about the given topic. The Students then were given enough time to navigate through and read the lessons. After they finish reading the chapter lessons, students were asked to complete the post-quiz. Each student had to go through the same procedure for 3 different lessons.

On the other hand the second group was given the traditional system besides the intelligent practices in a student centric approach. Therefore, students had to carry out the pre-quiz, then navigate through and read the lessons besides performing the intelligent practice exercises. After they are ready, students were asked to complete the post-quiz. Students were asked to do the same procedure for 3 different lessons (the same lessons the first group completed).

Results from both groups were calculated and analyzed to show that both groups have benefited from the system but the second group showed much better progress than the first group. In other words, the online system was a good learning tool for students to use in their learning process but the cognitive, adaptive intelligent practice exercises part is an important asset that guides, helps and supports students learning to acquire the missing knowledge and correct any misconceptions.

Chapter 7

Conclusion

ITSs have proven to be a very powerful and effective technology to enhance education and the learning process. Actually with rapid technological advances and the complexities of the modern world, it became necessary to find new educational technologies and a system that could cope with social and community needs by helping learners to acquire important skills needed to succeed. This has motivated research in the field of intelligent tutoring systems (ITSs). Furthermore, Web-based education has numerous advantages such as the convenience of taking a course without leaving the workplace or home and the reduced cost (Berz et al., 1999). Web-based tutoring systems are powerful technologies which have proved their effectiveness for distance learning that uses the knowledge about the domain, the learner and teaching strategies to support flexible individualized learning and tutoring instead of only providing the simplest solutions for distance learning (Brusilovsky et al. 1996; Eliot et al. 1995; Ritter 1997). At present, most existing educational applications (for example: tutorials, course notes besides many other applications) that have been delivered through the World Wide Web are just electronic books with very limited interactivity and diagnostic capability.

Our system in this thesis proposed is an online adaptive practice based ITS embedded within traditional student control tutoring software and including other facilities provided for both learners and teachers designed to improve the learners learning process. To explain our system, we started this thesis in chapter 1 by discussing the motivation and

the importance of an online adaptive, cognitive and practice based ITS accompanied with some brief background to support the idea. Then, proposed a system that we believe can help in bridging the gap between traditional education and distance learning were learners miss the teacher help, guidance and support while enjoying the convenience and flexibility of learning on their own pace. This system combines the benefits of traditional online, learner control tutoring systems where learners acquire the required knowledge and find information through a well designed sequence of chapters and lessons and the ITS through the adaptive, cognitive intelligent practices where learners apply their knowledge, the system identifies the knowledge gaps, help and guide students through adapting and individualizing the instruction according to the learners levels and by providing immediate feedback and recommendations when needed. This system is built inline with the cognitive theory of learning and is adaptive to learners' levels. Questions in the practices and the intelligent system were designed as MCQ to imitate the real exam environment and for objectivity where students are assessed purely according to their responses. Besides the cognitive theory of learning, the system meets many other well-known educational theories as discussed in chapter 3. Moreover, the system is in general student control software which provides the students with the flexibility of selecting the content and the sequence as example but at the same time intelligent practice are considered as program control. Thus, during the practices the program guide the student through a sequence of questions generated according the student level and progress.

Next, in chapter 2, we conducted an intelligent survey to overview ITSs, study their historical background in some details, and previously related work to learn from their experiences. In addition, we explored the different categories and classification methods

of existing ITSs and studied some online ITSs that teach computer subjects and how they work. Through the study we explored in depth some interesting areas and theories like the cognitive theory and the learner control approach. At the end of the chapter, we concluded by explaining where and how our system is categorized using the different methods and classifications mentioned in the chapter.

Following that, in chapter 3, we conducted other field and online research besides the survey done in chapter 2 to analyze our system and identify its main components. We explored the types of learning and some educational theories and how our system meets them. Moreover, we discussed MCQ assessments and how they work. This is in addition to taking about each component in details and its purpose and functionality.

In chapter 4, we surveyed and discussed the technical survey. and discussed the different software utilized to build the system. To build the system we employed different software and tools, such as Microsoft ASP.NET, Visual Basic .NET, Visual C++, Microsoft SQL server 2005, XML, CLIPS and some tools to produce the auto extract file like WinZip and ZIP to Secure EXE programs.

In Chapter 5, we discussed the system architecture, design and implementation in depth. This is including the database tables and relations, the intelligent practices algorithm and implementation besides overall website design and layout. In details, we discussed each system component design and implementation. Finally, in chapter 6, we carried on different tests to evaluate and examine the efficiency and effectiveness of our system. Some tests were done to examine the website technicality such as links and other tests were done to acquire feedback regarding the interface and navigation convenience. The main test was on the concept and effectiveness of the intelligent part of the system. This

test proved the system effectiveness and showed that students who used the intelligent part gained 8.1% better results in the average learning gain.

In conclusion, our research has shown the importance of integrating traditional online/web-based tutoring systems with ITSs. At the same time, from the research we understood how the cognitive theory works and how to plan practice exercises in a cognitive way modeling students' behaviors and guiding them through their learning process. Besides the cognitive theory of learning, our system meets other learning and educational theories as discussed in details in Chapter 3. This is in addition to exploring and discussing the adaptive practice based systems concept which individualize the instruction for students according to their levels and help students to find out, identify and fill knowledge gaps and correct misconceptions through applying and practicing what they learnt and receive appropriate feedback based on their responses and levels.

In this Thesis, we applied our research findings and built a system that meets students and teachers needs. This system is planned to emulate teacher one to one tutoring. It uses the classroom teacher model as the teaching methodology by giving continuous support, comments, feedback, guidance and direction.

Using appropriate knowledge domain modules, with a combination of intelligent adaptive cognitive practices raises the average learning gain for students dramatically. Students can read the lessons using the traditional mode then self assess their knowledge and fill knowledge gaps by doing the adaptive cognitive practices. Students can assess their knowledge before starting any lesson by completing the pre-quizzes. This approach is used for many reasons; one of them is for students to get an idea about their pre-

knowledge and what to expect from the lesson and for teachers to monitor and evaluate knowledge gain, students' progress and evaluate teaching methodology and lesson plan.

Furthermore, Intelligent Practices in this system start with an average level question. After studying the student response and evaluating student understanding of the lesson, the system generates questions appropriate for student understanding. Studied comments and feedback are given too to guide the student and to make sure that the student understands the topic well. Sometimes recommendations of readings are given when further help is needed.

Students benefited from the post-test at the end of each chapter. These quizzes were used to evaluate their readiness and for teachers to measure student learning gains and students' progress. This technique can be used as an evaluation tool if used in the classroom or any other system which requires evaluating students' performance regularly. This is besides the assessment tool provided at the end of module containing questions covering all chapters and lessons in the module. Again this tool can be used an assessment tool or for students to assess their exam readiness.

The system took the real advantage of interaction and distributed nature of the Internet. One good use of the internet is providing different communication tools where students and teachers interact. Moreover, the system has been built as a complete system that serves many other purposes besides teaching and providing feedback to the student. For example, teachers can add students, generate reports and update question banks to continuously improve and update the system. Students can track their progress through the reports from the lessons visited and assessments taken while teachers can track all student progress through the mentioned reports besides the progress reports and Quizzes

analysis reports. One important feature is to keep the human touch and existence via e-communications where students can submit their feedback and comments for teachers to read and consider. At the same time students can ask questions and view other students' questions while teachers have the authority to read, and answer these questions. This is in addition to the fact that these systems help peer students to collaborate, share experiences, know each other, create groups, help and support each other.

Both students and teachers gave excellent feedback about the website facilities. They found it easy to navigate, informative and helped them to progress and track progress. From one end, students liked the practices because they felt it helped directing them and showed them where they needed to concentrate on and what knowledge was missing. In most cases they were provided with all needed information and didn't have to go back to read the lessons. On the other end, test results with students had shown a clear evidence of the importance of the intelligent part on students' knowledge gain improvement. While this system proved to be successful in increasing the learning gain acquired by students and helped teachers in following up with their students' progress and continuously on improving the system.

The system has been tested and proved its effectiveness and importance in enhancing students' results by studying and comparing results and education gain between the pre lessons test and the post lessons test. While both group of students showed good improvement, the group who have used the intelligent part showed about 8.1% better results in average learning gain though.

In future, research on different cognitive questioning and assessment techniques should be carried on to enhance the way misconceptions and knowledge gaps are detected and to overcome MCQs weakness areas. Examples of such research maybe the integration of assay assessments with the current system and/or providing in-application intelligent practices depending on the topic of study nature. This is in addition to work on improving the interface and adding some hyper-linking techniques to provide suggested readings links related to the current lesson when a weakness is identified. For better future research results, educators and IT specialists should work together and an ongoing research should be maintained to follow up with new technologies and new educational and learning theories and approaches. Integrating new technologies with the new educational approaches shall continuously improve the way ITSs are designed and implemented.

Talking about our system in this thesis, the system can be expanded to cover all ICDL modules. The website interface can be easily modified and more analysis and reports can be added. While using the same questioning or practice logic as mentioned in this thesis, practices interface to other ICDL modules may be modified to include in application practices using different technologies with CLIPS. This prototype system can be used too to prepare for other international examinations, some examples are: SAT, GRE, LSAT, and FE besides many others. Finally, this system is only a prototype or only one example of how can we use ITSs to enhance online education and improve students learning gains in a cognitive, interactive and adaptive approach which proved its success.

References:

1. Active Math website at http://lecture.activemath.org/ActiveMath2/main/menu.cmd [last visited 20/4/2008]
2. Ainsworth S. and Grimshaw S. (2002); Evaluating The Effectiveness And Efficiency Of The Redeem Intelligent Tutoring System Authoring Tool; ESRC Centre for Research in Development, Instruction and Training School of Psychology, University of NottinghamUniversity Park, Nottingham NG7 2RD,United Kingdom [online] http://www.psychology.nottingham.ac.uk/staff/sea/techreport_69.pdf [last visited 22/4/2008]
3. Albert T. Corbett and Kenneth R. Koedinger, John R. Anderson (1997), *Handbook of Human-Computer Interaction*, in M. Helander, T. K. Landauer, P. Prabhu (Eds), *Elsevier Science B. V.* [online] http://act-r.psy.cmu.edu/papers/173/Chapter_37_Intelligent_Tutoring_Systems.pdf [last visited 25/3/2008]
4. Aleven V., McLaren B. M., Sewall, J., and Koedinger, K. (2006), *The Cognitive Tutor Authoring Tools (CTAT): Preliminary evaluation of efficiency gains*. In M. Ikeda, K. D. Ashley, and T. W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems (ITS 2006)*,[online] http://www.pitt.edu/~bmclaren/CTAT-ITS2006.pdf [last visited 15/4/2008]
5. Aleven V., Sewall J., McLaren B. M., and Koedinger, K. R. (2006), *Rapid authoring of intelligent tutors for real-world and experimental use*. In Kinshuk, R. Koper, P. Kommers, P. Kirschner, D. G. Sampson, and W. Didderen (Eds.), *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, (pp. 847-851). Los Alamitos, CA: IEEE Computer Society.[online] http://ctat.pact.cs.cmu.edu/pubs/CTAT-ICALT2006.pdf [last visited 30/3/2008]
6. Anderson J., Corbett A., Koedinger K and Pelletier R. (1995) *Cognitive Tutors: Lessons Learned. Journal of the Learning Sciences* 1995; [online] available at http://act-r.psy.cmu.edu/papers/Lessons_Learned.html [last visited 31/3/2008].
7. Anderson JR. (1993), *Rules of the Mind*. Hillsdale, NJ: Erlbaum
8. Anderson, J. and Reiser, B. (1985), *The LISP Tutor*. [online] http://www.cs.mdx.ac.uk/staffpages/serengul/tutorial/LISP.TUTOR.htm [last visited 20/4/2008]

9. Anderson J. R., Bower G. H. (1973), *Human associative memory*. Washington: Winston and Sons.[online] http://www.questia.com/PM.qst?a=oand d=94197099 [last visited 18/4/2008]
10. Anderson J. R., Boyle C. F., Corbett A. T., and Lewis M.W. (1990), *Cognitive modeling and intelligent tutoring. Artificial Intelligence*, 42,7-49.
11. Anderson J. R. and Pelletier R. (1991), *A development system for model-tracing tutors*. In Proc.of *the International Conference on the Learning Sciences*, Evanston, IL, pp. 1-8.
12. Anderson L. W., Krathwohl D. R., Airasian P. W., Cruikshank K. A., Mayer R. E., Pintrich P. R., Raths J. and Wittrock M. C. (2001), *A Taxonomy for Learning, Teaching, and Assessing — A Revision of Bloom's Taxonomy of Educational Objectives*; Addison Wesley Longman, Inc. 2001. Retrieved from [online] " http://www.pilambda.org/horizons/v83-3/bkreview.pdf "[last visited 22/4/2008]
13. Bereiter, C., and Scardamalia, M. (1998), *Beyond Bloom's taxonomy: Rethinking knowledge for the knowledge age*. In A. Hargreaves, A. Lieberman, M. Fullen, and D. Hopkins, (Eds.), *International handbook of educational change*. Boston: Kluwer Academic.
14. Berz Martin, Erdélyi Béla, Hoefkens Jens (1999), *Experiences with interactive remote graduate instruction in beam physics,* Journal of *Interactive Learning Research, v.10 n.1*

15. Bhagat1 S.& L. , Kavalan J., Sasikumar M.(2002), *Acharya: An Intelligent Tutoring Environment for Learning SQL*, [online] http://www.cdacmumbai.in/design/corporate_site/override/pdf-doc/acharya.pdf [last visited 12/5/2008]
16. Blaine, L.H. (1981), *Programs for structured proofs*. In Suppes, P. (Ed.) *University-level Computer-assisted Instruction at Stanford: 1968-1980*.Institute for Mathematical Studies in the Social Sciences, Stanford University, Stanford, California.
17. Bloom, B. S. (1980), *All Our Children Learning*. New York: McGraw-Hill.
18. Bloom, B. S. (1956), *Major categories in the cognitive domain of the taxonomy of educational objectives*. [online] http://krummefamily.org/guides/bloom.html, [last visited 12/4/2008]
19. Bloom B. S. (1956), *Taxonomy of Educational Objectives: The Classification of Educational Goals*; pp. 201-207; Susan Fauer Company, Inc.
20. Bloom, B.S., Englehart, M.D., Furst, E.J., Hill, W.H., and Krathwohl, D.R. (1956), *A Taxonomy of Educational Objectives: Handbook 1: Cognitive Domain.* New York: McKay. [online] http://www.humboldt.edu/~tha1/bloomtax.html [last visited 12/4/2008]
21. Boose J. H. (1988), *A Survey of Knowledge Acquisition Techniques and Tools*. 3rd *AAAI-Sponsored Knowledge Acquisition for Knowledge-Based Systems Workshop*, November 1988, pg. 3.1-3.23. Banff, Canada

22. Bouhadada T., Laskri M.T. (2001), *DB-TUTOR: An Intelligent Tutoring System Using a Troublemaker Companion*, ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'01) [online] http://csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/proceedings/&to

c=comp/proceedings/aiccsa/2001/1165/00/1165toc.xml&DOI=10.1109/AICCSA.2001.933949 [last visited 25/5/2008]

23. Brown J. S., Burton R. R., Hausman C. L., Goldstein I. , Huggins B. , and Miller M. L. (1977), *Aspects of a Theory for Automated Student Modeling*. *BBN Report 3549 (ICAI Report 4)*. Bolt, Beranek and Newman, Inc., Cambridge, MA,. [online] http://eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/31/86/be.pdf , [last visited [12/4/2008]
24. Brown J. S. and Burton R. R. (1978), *Diagnostic models for procedural bugs in basic mathematical skills*. Cognitive Science. [online] http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/38/57/4e.pdf, [last visited 10/4/2008]
25. Brown J. S., Burton R. R., Miller M. L., de Kleer J., Purcell S., Hausman C. L., and. Bobrow R (1975), *Steps toward a Theoretical Foundation for Complex Knowledge-Based CAI.* BBN Report 3135 (ICAI Report 2). Bolt, Beranek and Newman, Inc., Cambridge, MA
26. Brown J.S., Collins A. & Duguid S. (1989), *Situated cognition and the culture of learning.* Educational Researcher, 18(1), 32-42
27. Brusilovsky P.L. (1992), *A framework for intelligent knowledge sequencing and task sequencing*. In Frasson C., Gauthier G. and McCalla G.I. (eds.) Intelligent Tutoring Systems. *Proceedings of the Second International Conference, ITS'92*. Springer-Verlag, Berlin.
28. Brusilovsky, P.(1995); Intelligent tutoring systems for the world wide web; university of Trier, Germany [online] http://www.igd.fhg.de/archive/1995_www95/proceedings/posters/48/index.html [last visited 15/4/2008]
29. Brusilovsky P. (1998), *Methods and Techniques of Adaptive Hypermedia*. In P. Brusilovsky,Kobsa, and J. Vassileva, editors, *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publishers, The Netherlands,. [online] http://www.sis.pitt.edu/~peterb/papers/UMUAI96.pdf [last visited 2/5/2008]
30. Brusilovsky P., Schwarz E., and Weber G. (1996). *ELM-ART: An intelligent tutoring system on World Wide Web*. In Frasson, C., Gauthier, G., and Lesgold, A. (Ed.), *Intelligent Tutoring Systems (Lecture Notes in Computer Science, Vol. 1086).* Berlin: Springer Verlag.[online] http://www.contrib.andrew.cmu.edu/~plb/ITS96.html [last visited 22/4/2008]
31. Burton R. (1982), *Diagnosing bugs in a simple procedural skill. In D.H. Sleeman, and J.S. Brown (Eds.)* Intelligent Tutoring Systems. Academic Press, London
32. Butz B.P. (1999), *Freedom of Choice in an Intelligent Tutoring System*; Electrical and Computer Engineering Department; Temple University; Philadelphia
33. Butz C.J., Hua S., and Maguire R.B. (2004), *Bits: a Bayesian Intelligent Tutoring System For Computer Programming*, Department of Computer Science, University of Regina, Canada, [online] http://www2.cs.uregina.ca/~butz/publications/wccce04.doc [last visited 22/5/2008]
34. Butz C. J., Hua S. and Maguire R. B. (2006), *A web-based bayesian intelligent tutoring system for computer programming,* IOS Press, Amsterdam, The Netherlands, The Netherlands. [online]

http://portal.acm.org/citation.cfm?id=1183074.1183079&coll=GUIDE&dl=GUIDE&CFID=7799445&CFTOKEN=54937698 [last visited 12/5/2008]

35. Canfield W.(2001) ; *ALEKS: A Web-based intelligent tutoring system, Mathematics and Computer Education* [online] http://findarticles.com/p/articles/mi_qa3950/is_200104/ai_n8932950 [last visited 24/4/2008]
36. Capuano N., Marsella M., Salerno S. (2000), *ABITS: An Agent Based Itelligent Tutoring System for Distance Learning,* http://www.capuano.biz/Papers/ITS_2000.pdf
37. Carneson, J., Delpierre, G., & Masters, K. (n.d.). *Designing and managing multiple-choice questions: Appendix C, multiple-choice questions and Bloom's taxonomy.* [online]: http://www.uct.ac.za/projects/cbe/mcqman/mcqappc.html [last visited 15/5/2008]
38. Cashin W.E. (1987), *Improving multiple-choice tests*; Idea Paper No. 16. Manhattan, KS: Center for Faculty Development and Evaluation, Kansas State University. [online] http://ctl.unc.edu/fyc8.html [last visited 3/3/2008]
39. Chilkat Software, *Zip 2 Secure EXE* (2008): http://www.chilkatsoft.com/ChilkatSfx.asp
40. Chipman S.F. (1993), *Gazing once more into the silicon chip: Who's revolutionary now?* In S.P. Lajoie and S.J. Derry, *Computers as Cognitive Tools*. Hillsdale, NJ: Erlbaum, 341-367
41. Clark, A.J. (1993). *Associative Engines: Connectionism, Concepts, and Representational Change.* Cambridge, MA: MIT Press ISBN 0262032104, 9780262032100.
42. Clark R. E. (2004), *Design document for a guided experiential learning course*. Final report on contract DAAD 19-99-D-0046-0004 from TRADOC to the Institute for Creative Technologies and the Rossier School of Education [online]http://www.usc.edu/dept/education/cct/publications/clark_gel.pdf [last visited 15/2/2008]
43. Cognition & Technology Group at Vanderbilt (1993),. *Anchored instruction and situated cognition revisited. Educational Technology*, 33(3), 52-70
44. Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal, [online http://www.cemcentre.org/Documents/CEM/publications/downloads/CEMWeb024%20Educational%20Outcomes%20Of%20Tutorings%20Meta%20Analysis.pdf] last visited 12/1/2008]
45. Collins and Loftus (1975). *A spreading activation theory of semantic memory*.
46. Contreras W. F., Galindo E. G., Caballero E. M. and Caballero G. M. (2006), *An Intelligent tutoring system for a Virtual E-Learning Center*.
47. Corbett A.T. and Anderson J.R. (1992), *Student modeling and mastery learning in a computer-based programming tutor. In Proceedings of the Second International Conference on Intelligent Tutoring Systems*. Montreal, Canada.
48. Corbet, Koedinger and Anderson (1997) , *Intelligent Tutoring Systems* [online] http://128.2.67.57/papers/173/Chapter_37_Intelligent_Tutoring_Systems.pdf, [last visited 22/4/2008]

49. Crane John P. (2008), *Levels of Processing Theory*, IB Psychology website, [online] http://www.cranepsych.com/Psych/Levels_of_Processing.pdf , [last visited 22/5/2008]
50. Crowley R. S., Legowski E., Medvedeva O., Tseytlin E., Roh E., and Jukic D., (2007), *Evaluation of an Intelligent Tutoring System in Pathology: Effects of External Representation on Performance Gains, Metacognition, and Acceptance* [online] http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2213473 [last visited 15/4/2008]
51. Dirnberger Mary T., Wenzel and Brenda M (1996), *Experimental advanced instructional design advisor (xaida) training: a multidimensional approach to training evaluation;* Mei Technology Corporation; San Antonio, TX [online] http://www.ijoa.org/imta96/paper80.html, [last visited 4/4/29008]
52. Eliot C. and Woolf B. (1995), *An Adaptive Student Centered Curriculum for an Intelligent Training System. User Modeling and User-Adapted Interaction*
53. Fasli M. and Michalakopoulos M. (2005), *Learning Through Game-like Simulations* ,University of Essex, Department of Computer Science, UK , [online] http://www.ics.heacademy.ac.uk/italics/vol5iss2/MFasli.htm [last visited 20/6/2008]
54. Freedman R. (2000), *What is an Intelligent Tutoring System?* [online] http://www.cs.niu.edu/~freedman/papers/link2000.pdf [last visited 25/5/2008]
55. Gagne R., Briggs L. & Wager W. (1992). *Principles of Instructional Design* (4th Ed.). Fort Worth, TX: HBJ College Publishers.
56. Gertner A, VanLehn K. (2000), ANDES: a coached problem solving environment for physics. Paper presented at: Intelligent Tutoring Systems: 5th International Conference (ITS-2000); Berlin.
57. Gomez-Martin M. A., Gomez-Martin P. P., and Gonzalez-Calero P. A., (2004), *Game-driven Intelligent Tutoring Systems*. In *Proceedings of the Third International Conference on Entertainment Computing (ICEC)*, [online] http://gaia.fdi.ucm.es/grupo/publications/2004_icec_marcoa.pdf [last visited 20/6/2008]
58. GraphPad software , *online quick calculators* at http://www.graphpad.com/quickcalcs/ttest1.cfm [Last visited 20/5/2008]
59. Greer J. E., and McCalla G.I., eds. (1994), *Student Modelling: The Key to Individualized Knowledge-Based Instruction.* Nato ASI Series F: *Computer and Systems Sciences.* Vol: 125. Springer Verlag.
60. Gronlund N.E. (1988), *How to construct achievement tests (4th ed.)*. NJ: Prentice-Hall.
61. Guzmán E., Conejo R. and García-Hervás, E. (2005), *An Authoring Environment for Adaptive Testing*. *Educational Technology and Society Journal*. [online] http://www.ifets.info/journals/8_3/7.pdf [last visited 12/4/2008]
62. HALLoGRAM Publishing, Aurora Co, *Visual Basic .NET Standard* at http://www.hallogram.com/vb/
63. Harris J. (2002), An Introduction to Authoring Tools, Learning circuits;[online] http://www.learningcircuits.org/2002/mar2002/harris.html [last visited 12/4/2008]
64. Haugsjaa E. and Woolf B. (1996), *3D Visualization Tools in a Design for Manufacturing Tutor*. In *Proceedings of Educational Multimedia and Hypermedia*, Boston, Mass.

65. Heffernan N. T., Koedinger K. R., and Aleven V. (2003), *Tools Towards Reducing the Costs of Designing, Building, and Testing Cognitive Models*. *The 2003 Conference on Behavior Representation in Modeling and Simulation,* BRIMS [online] http://ctat.pact.cs.cmu.edu/pubs/03-BRIMS-063.doc [last visited 10/4/2008].
66. Hoffman B. (General Editor), *The Encyclopedia of Educational Technology (EET)*, The San Diego State University Department of Educational Technology [online] http://coe.sdsu.edu/eet/Admin/Biblio/index.htm [last visited 22/4/2008]
67. Henke S. (2006), *Smarter Software solutions*: *Education and training*, [online] http://www.awaresearch.us/solutions/training/index.htm and http://www.awaresearch.us/solutions/training/training_projects.htm [last visited 20/6/2008]
68. Isaacs, G (1994), *Multiple Choice Testing*, HERDSA Green Guide No 16
69. Janicki T., Steinberg G.(2000); *Evaluation of Posttest Scores for a Web-Based Tutorial; Authoring Tool that Encompasses Pedagogy in the Development Process*; University of North Carolina NC and , Kent State University, Kent, OH In *The Proceedings of ISECON 2000*, v 17 Philadelphia. [online] http://isedj.org/isecon/2000/500/ISECON.2000.Janicki.pdf [last visited 22/4/2008]
70. John R. Anderson , C. Franklin Boyle , Albert T. Corbett , Matthew W. Lewis (1990), *Cognitive modeling and intelligent tutoring*, Artificial Intelligence, v.42 n.1, p.7-49
71. Jonassen, D.H., Tessmer, M., Hannum, W.H. (1999). *Task Analysis Methods for Instructional Design.* Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers. [online] http://www.questia.com/PM.qst?a=oand d=91844871 [last visited 12/4/2008]
72. Kay, J. (2001), *Learner Control*. *User Modeling and User-Adapted Interaction*, 11:111-127.
73. Kearsley, G. (1985), *Microcomputer Software: Design and Development Principles*. *Journal of Educational Computing Research*, 1: 209-220.
74. Kearsley, G.(2008), *Explorations in Learning & Instruction: The Theory Into Practice Database*,[online] http://tip.psychology.org/ [last visited 20/5/2008]
75. Kehoe and Jerard (1995), *Writing multiple-choice test items.* Practical Assessment, Research & Evaluation, [online] http://PAREonline.net/getvn.asp?v=4&n=9 [last visited 12/5/2008]
76. Kirschner, P., Sweller, J., and Clark, R. E. (2006), *Why minimally guided learning does not work: An analysis of the failure of discovery learning, problem-based learning, experiential learning and inquiry-based learning*. *Educational Psychologist, 41(2),* 75-86. [online] http://projects.ict.usc.edu/dlxxi/materials/clark/Constructivism_EP_05_Kirschner_Sweller_Clark_Ss.pdf [last visited 10/4/2008]
77. Koedinger KR, Anderson JR.(1993), *Effective use of intelligent software in high school math classrooms* Proc *World Conf Art Intell Edu 1993*:241-248. [online] http://act-r.psy.cmu.edu/papers/227/Effectiv.pdf [last visited 12/4/2008]
78. Koedinger K. R., Anderson J. R. (1997*), Intelligent Tutoring Goes To School in the Big City* ; *International Journal of Artificial Intelligence in Education (1997),* 8,30-43 [online] http://www.carnegielearning.com/web_docs%5CIntelligent_tutoring.pdf [last visited 10/4/2008]

79. Koedinger, K. R., Aleven, V. and Heffernan, N. T. (2003), *Toward a Rapid Development Environment for Cognitive Tutors.* In U. Hoppe, F. Verdejo, and J. Kay (Eds.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education,* AI-ED 2003 (pp. 455-457). Amsterdam: IOS Press.[online] http://www.cs.usyd.edu.au/~aied/vol2/vol2_Koedinger.pdf [last visited 12/4/2008]
80. Koedinger, K. R., Aleven, V., Heffernan. T., McLaren, B. and Hockenberry, M. (2004), *Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration*. In *the Proceedings of 7th Annual Intelligent Tutoring Systems Conference.* Maceio, Brazil.[online] http://ctat.pact.cs.cmu.edu/pubs/KoedingerEtAlTutorToolsITS2004.pdf [last visited 10/4/2008]
81. Kolb D. A. (1984), *Experiential Learning: experience as the source of learning and development* New Jersey: Prentice-Hall (0 13 295261 0)
82. Krathwohl, D., Bloom, B., and Masia, B. (1956), *Taxonomy of educational objectives. Handbook II: Affective domain.* New York: David McKay. [online] http://chiron.valdosta.edu/whuitt/col/affsys/affdom.html [last visited 12/4/2008]
83. Krathwohl, D.R., Bloom, B.S., and Masia, B.B. (1964), *Taxonomy of educational objectives: Handbook II: Affective domain*. [online] http://deepblue.lib.umich.edu/bitstream/2027.42/43808/1/11217_2004_Article_BF00373956.pdf [last visited 12/4/2008]
84. Krige A. (1999), *Intelligent Tutoring Systems: BUGGY*, [online] http://tecfa.unige.ch/staf/staf-d/krige/staf11/buggy.html [last visited 12/4/2008]
85. Krug, Steve (2001), *Don't make me think! A common sense approach to Web usability.* Indianapolis, Ind.: Que. [sample chapter online] http://www.sensible.com/chapter.html [last visited 12/4/2006]
86. Kulik CC, Kulik JA.(1991), *Effectiveness of computer-based instruction: an updated analysis.* Comp Hum Beh 1991; 7:75-95. [online] http://deepblue.lib.umich.edu/bitstream/2027.42/29534/1/0000622.pdf [last visited 12/4/2008]
87. Kulik JA, Kulik CC, Cohen PA (1980), *Effectiveness of computer-based college teaching: a meta-analysis of findings* Rev Edu Res 1980;50:525-544. [online] http://www.eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/30/41/e7.pdf [last visited 10/4/2008]
88. Lajoie, S. and A. Lesgold. (1992), *Apprenticeship Training in the Workplace: Computer-Coached Practice Environment as a New Form of Apprenticeship*. In *Intelligent Instruction by Computer, Farr and Psotka, eds*., Taylor and Francis, Washington, DC, pp. 15-36.
89. Lane, H. Chad (2006), *Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning*, Institute for Creative Technologies ; University of Southern California; [online] http://people.ict.usc.edu/~lane/papers/ITSProspectsLane-Aug06.pdf ; [last visited 15/4/2008]
90. Lave, J. (1988). Cognition *in Practice: Mind, mathematics, and culture in everyday life*. Cambridge, UK: Cambridge University Press.[online] http://homepage.mac.com/scottlab/situated.html [last visited 20/5/2008]

91. Lave, J., & Wenger, E. (1990). *Situated Learning: Legitimate Periperal Participation*. Cambridge, UK: Cambridge University Press. [online] http://homepage.mac.com/scottlab/situated.html [last visited 20/5/2008]
92. Lee, S. and Lee, Y. H. K. (1991), *Effects of Learner-Control Versus Program-Control Strategies on Computer-Aided Learning of Chemistry Problems: For Acquisition of Review?* ; *Journal of Educational Psychology*, 83(4): 491-498.
93. Lesgold A., Lajoie S., Bunzo M, Eggan G. (1993), *SHERLOCK: a coached practice environment for an electronics troubleshooting job* In: *Larkin J, Chabay R, editors. Computer Assisted Instruction and Intelligent Tutoring Systems: Established Communication and Collaboration*. Hillsdale, NJ: Lawrence Erlbaum Associates; 1993. pp. 201-238.[online] http://eric.ed.gov/ERICDocs/data/ericdocs2sql/content_storage_01/0000019b/80/1c/fb/12.pdf [last visited 22/4/2008]
94. Lovett, M. C. (1998), *Cognitive task analysis in service of intelligent tutoring systems design: A case study in statistics*. In B. P. Goettl, H. M. Halff, C. L. Redfield, and V. J. Shute (Eds.) *Intelligent Tutoring Systems, Lecture Notes in Computer Science Volume 1452* pp. 234-243. New York: Springer.
95. Lowe D., Cogswell J. and Cox K.(2006), ASP.NET 2.0 All In One Desk Reference For Dummies, Willey ISBN: 0-471-78598-9
96. Major, N., Ainsworth, S. and Wood, D. (1997), *REDEEM: Exploiting symbiosis between psychology and authoring environments*. *International Journal of Artificial Intelligence in Education,* 8 (3-4) 317-340. [online] http://aied.inf.ed.ac.uk/members/archive/vol_8/major/paper.pdf [last visited 12/4/2008]
97. Mandl H. and Lesgold A. (1988), *Preface, Learning Issues for Intelligent Tutoring Systems*, Springer-Verlag, Inc., New York, NY.
98. Mason G., *Blooms taxonomy, Instructional Technology Program* [online] http://classweb.gmu.edu/ndabbagh/Resources/Resources2/bloomstax.htm , [last visited 12/4/2008]
99. Mason G., *Instructional Design Knowledge Base, Perform A Task Analysis*; [online] http://classweb.gmu.edu/ndabbagh/Resources/Resources2/Task%20Analysis.htm [last visited 12/3/2008]
100. Mason G., *Gagné's Five Learned Capabilities* [online] http://classweb.gmu.edu/ndabbagh/Resources/Resources2/gagnetax.htm [last visited 13/3/2008]
101. Mason G., *Krathwohl's taxonomy*; [online] http://classweb.gmu.edu/ndabbagh/Resources/Resources2/krathstax.htm [last visited 12/3/2008]
102. Matsuda, N., Cohen, W. W., and Koedinger, K. R. (2005). *Applying Programming by Demonstration in an Intelligent Authoring Tool for Cognitive Tutors*. In *AAAI Workshop on Human Comprehensible Machine Learning* (Technical Report WS-05-04) (pp. 1-8). Menlo Park, CA: AAAI association. [online] http://ctat.pact.cs.cmu.edu/pubs/McLarenICWE2004.pdf [last visited 12/4/2008]
103. McDonald, J. (1981), *The EXCHECK CAI system*. In Suppes, *P. (Ed.) University-level Computer-assisted Instruction at Stanford: 1968-1980*. Institute for

Mathematical Studies in the Social Sciences, Stanford University, Stanford, California
104. McLellan, H. (1995), *Situated Learning Perspectives*. Englewood Cliffs, NJ: Educational Technology Publications.
105. Merrill, M.D. (1994), *Instructional design theory*. Englewood Cliffs, NJ: Educational Technology Publications.
106. Microsoft (2008), MSDN, *ASP.NET Developer Center* at http://msdn2.microsoft.com/en-us/asp.net/bb498194.aspx [last visited 25/4/2008]
107. Microsoft, Microsoft SQL Server , *SQL Server 2005* (2007) at http://www.microsoft.com/sql/prodinfo/overview/default.mspx .[last visited 12/4/2008]
108. Miller, G.A. (1956), *The magical number seven, plus or minus two: Some limits on our capacity for processing information*. Psychological Review, 63, 81-97. [online] http://www.well.com/user/smalin/miller.html , [last visited 20/5/2008]
109. Miller, G.A., Galanter, E., & Pribram, K.H. (1960), *Plans and the Structure of Behavior*. New York: Holt, Rinehart & Winston.
110. Mitchell CM (1997), *Horizons in pilot training: desktop tutoring systems* In: Sarter N, Amalberti R, editors. Cognitive Engineering in the Aviation Domain. Erlbaum: Mahwah, NJ; [online] http://citeseer.ist.psu.edu/cache/papers/cs/15479/http:zSzzSzwww.isye.gatech.eduzSzchmsrzSzpublicationszSzcmmzSzsarter.chapter.1997zSzpaper.pdf/mitchell97horizons.pdf [last visited 10/4/2008]
111. Mischel J. (2008*),.NET reference Guide*, InformIT website [online] , http://www.informit.com/guides/content.aspx?g=dotnetand seqNum=54 [last visited 2/5/2008]
112. Mitrovic, A. (1997), SQL-*Tutor: a preliminary report*. [online] http://www.cosc.canterbury.ac.nz/research/reports/TechReps/1997/tr_9708.pdf [last visited 10/4/2008]
113. Mitrovic, A. (2003), *An intelligent SQL tutor on the Web. International Journal of Artificial Intelligence in Education*, vol 13, 2003, pp. 171-195. [online] http://aied.inf.ed.ac.uk/members02/archive/Vol_13/mitrovic/paper.pdf [last visited 10/4/2008]
114. Mitrovic, A.(2005), *The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor*. In: C-K Looi, G. McCalla, B. Bredeweg, J. Breuker (eds) *Proc. Artificial Intelligence in Education, 2005*, IOS Press, 499-506. [online] http://www.cs.usyd.edu.au/~aied/vol9/vol9_roger_Mitrovic.pdf [last visited 10/4/2008]
115. Mitrovic, A., Koedinger, K. and Martin, B. (2003), *A comparative analysis of cognitive tutoring and constraint-based modeling.* In: Brusilovsky, *P., Corbett, A. and Rosis, F.d. (eds.) Proc. 9th International conference on User Modelling* UM2003, Pittsburgh, USA, Springer-Verlag, pp. 313-322. [online] http://nth.wpi.edu/classes/cs525t_tutoring_systems/PaperForClass/UM03-Tanya_Ken.doc [last visited 12/4/2008]
116. Mitrovic, A., Suraweera, P., Martin, B., Weerasinghe (2004), *A. DB-suite: Experiences with Three Intelligent, Web-based Database Tutors. Journal of Interactive Learning Research*, 15, 409-432. [online]

http://ir.canterbury.ac.nz/bitstream/10092/309/1/12592814_db-suite.pdf [last visited 15/4/2008]

117. Mitrovic A., Suraweera P., Martin B., Zakharov K., Milik N. and Holland J. (2006), *Authoring Constraint-based Tutors in ASPIRE*; Intelligent Computer Tutoring Group, University of Canterbury, Christchurch, New Zealand [online] http://www.cosc.canterbury.ac.nz/tanja.mitrovic/ASPIRE-ITS06.pdf [last visited 22/4/2008]

118. Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J.(2005), *ASPIRE: Student Modelling and Domain Specification*. Tech. Report TR-COS 08/05, University of Canterbury, [online] http://www.cosc.canterbury.ac.nz/research/reports/TechReps/2005/tr_0508.pdf [last visited 15/4/2008]

119. Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J. (2005), *ASPIRE: Functional Specification and Architectural Design*. Tech. Report TRCOSC 05/05, University of Canterbury. [online] http://www.cosc.canterbury.ac.nz/research/reports/TechReps/2005/tr_0506.pdf [last visited 15/4/2008]

120. Mogulla S. (2004) *Start VB Dot Net , VB dot Net from developers view*, Microsoft .Net Explained. [online] http://www.brillianceweb.com/betterwebdesign/tips_52.aspx [last visited 25/4/2008]

121. Moore, D.S. (1982). *Reconsidering Bloom's Taxonomy of educational objectives, cognitive domain*. Educational Theory, 32(1) 29-34.

122. Munro, A., Johnson, M.C., Pizzini, Q.A., Surmon, D.S., Towne, D.M. and Wogulis, J.L. (1997). *Authoring simulation-centered tutors with RIDES. International Journal of Artificial Intelligence in Education*. 8(3-4), 284-316.[online] http://aied.inf.ed.ac.uk/members/archive/vol_8/munro/paper.pdf [last visited 15/4/2008]

123. Murray T. (1998). *Authoring knowledge-based tutors: Tools for content, instructional strategy, student model, and interface design*. *Journal of the Learning Sciences, 7(1).*

124. Murray T (1998) ; *Expanding the Knowledge Acquisition Bottleneck for Intelligent Tutoring Systems*; (*Preface to the IJAIED Special Issue on Authoring Systems for Intelligent Tutoring Systems*); Computer Science Department, University of Massachusetts, Amherst, MA [online] http://aied.inf.ed.ac.uk/abstracts/Vol_8/murray.html [last visited 22/4/2008]

125. Murray, T (1999). *Authoring Intelligent Tutoring Systems: An analysis of the state of the art*. *International Journal of Artificial Intelligence in Education*, 10, 98 [online] http://aied.inf.ed.ac.uk/members99/archive/vol_10/murray/paper.pdf [last visited 15/4/2008]

126. Murray, T. (2003). *An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art*. Chapter 17 in *Murray, T., Blessing, S. and Ainsworth, S. (Eds.). Authoring Tools for Advanced Technology Learning Environments.* Kluwer Academic Publishers, Dordrecht.; [online] http://home.comcast.net/~perspegrity5/atoolsbook/ITS_AT_Overview_Tables.pdf [last visited 15/4/2008]

127. Niederst, Jennifer. (1999). *Web design in a nutshell: A desktop quick reference*. Sebastopol, Calif.: O'Reilly.
128. Ohlsson, S. (1992): *Constraint-Based Student Modeling*. Journal of *Artificial Intelligence in Education* 3(4), pp. 429-447.
129. Ohlsson, S. (1994), *Constraint-based Student Modeling*. *Proc. Student Modeling: the Key to Individualized Knowledge-based Instruction*, Berlin, Springer-Verlag, pp. 167-189,
130. Ohlsson, S. (1996) *Learning from Performance Errors. Psychological Review*, vol 103 (2), , pp. 241-262.
131. Ong J. and Ramachandran S. , *Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI;* [online] http://www.stottlerhenke.com/papers/ITS_using_AI_to_improve_training_performance_and_ROI.pdf [last visited 10/4/2008]
132. Ong J. and Ramachandran S., (2000), *Intelligent Tutoring Systems: The What and the How*, [online] http://www.learningcircuits.org/2000/feb2000/ong.htm [last visited 15/3/2008]
133. Ozdemir, B. Alpaslan, F.N. *An intelligent tutoring system for student guidance in Web-based courses,* Dept. of Computer. Eng., Middle East Tech. Univ., Ankara;
134. Ramachandran, S. (2004). *An Intelligent Tutoring System Approach to Adaptive Instructional Systems*, Phase II SBIR Final Report, Army Research Institute, Fort Rucker, AL.
135. Rane A. (2005), *Intelligent Tutoring System For Marathi* , [online] http://www.cdacmumbai.in/design/corporate_site/override/pdf-doc/marathi-tutor.pdf [last visited 22/5/2008]
136. Ramachandran S., Remolina E., and Fu D. (2004) ; *FlexiTrainer: A Visual Authoring Framework for Case-Based Intelligent Tutoring Systems*. Stottler Henke Associates, Inc., San Mateo, CA, 94404 .[online] http://www.stottlerhenke.com/papers/ITS-04-flexitrainer.pdf [last visited 22/4/2008]
137. Rickel, J.W. (1989), *Intelligent Computer-Aided Instruction: A survey Organized Around System Components.* [online] http://citeseer.ist.psu.edu/cache/papers/cs/2021/http:zSzzSzwww.isi.eduzSzisdzSzrickelzSzicai_survey.pdf/rickel89intelligent.pdf [last visited 2/5/2008]
138. Riley G. (2008), *CLIPS: A Tool for Building Expert Systems* at :http://clipsrules.sourceforge.net/ [last visited 25/4/2008]
139. Ritter, S., Brusilovsky, P., & Schwarz, E. (1997). *Distributed intelligent tutoring on the Web*. In *B. du Boulay and R. Mizoguchi* (Eds.), Artificial Intelligence in Education (pp. 482-489)..
140. Rizvanov F. and Lizotte R.(2000), *Critical Thinking, Bloom"s Taxonomy and Test Writing*, Ashland Community College, [online] http://cit.necc.mass.edu/atlt/ACC_Inst.htm [last visited 20/5/2008]
141. Salgueiro Fernando., Costa Guido, Zulma Cataldi, Fernando Lage, Ramón García-Martínez (1998) ;*Redefinition of basic modules of an intelligent tutoring system: the tutor module*;[online] http://dc.exa.unrc.edu.ar/wicc/papers/InteligenciaArtificial/98.pdf [last visited 10/4/2008]

142. Sax, G. (1974) *Principles of educational measurement and evaluation*. Belmont, CA: Wadsworth.
143. Scandura, J.M. (2004). *Structural Learning Theory: Current Status and New Perspectives.*
144. Seels and Glasgow (1990). *Exercises in instructional design*. Columbus OH: Merrill Publishing Company. ISBN-10: 0675208270,ISBN-13: 9780675208277
145. Shaaron Ainsworth (1996 – modified 2005); REDEEM: Creating Reusable Intelligent Tutoring Systems; [online] http://www.psychology.nottingham.ac.uk/research/credit/projects/redeem/ [last visited 22/4/2008]
146. Shute, V., R. Glaser, and K. Raghaven. (1989). *Inference and Discovery in an Exploratory Laboratory.* Learning and Individual Differences, Ackerman, P., R. Sterberg, and R. Glaser, eds., pp. 279-326.
147. Sivagnanam, G. *Students t-tes*t [online] http://ampiweb.org/indphar/Statistics/tandWilcox.doc [last visited 12/4/2008]
148. Sklar, Joel. (2000). *Principles of Web design. Cambridge*, Mass.: Course Technology.
149. Smith, P.L. and Ragan, T. J. (1999). *Instructional Design. 2nd edition*. Upper Saddle River, New Jersey: Merrill.
150. Startvbdotnet.com 2004 website at http://www.startvbdotnet.com/aspsite/asp/ [last visited 20/5/2008]
151. Steinberg, E. R. (1977), *Review of Student Control in Computer-Assisted Instruction. Journal of Computer-Based Instruction*, 3(3): 84-90, 1977.
152. Steinberg, E. R. (1989), *Cognition and Learner Control: A Literature Review*, 1977-1988. Journ*al of Computer-Based Instruction*, 16(4): 117-121, 1989.
153. Steven Linton Supervisor: Dr Brent Martin (2003); *Knowledge Representation and Acquisition for Intelligent Tutoring Systems*; [online] http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/2003/hons_0310.pdf [last visited 22/4/2008]
154. Stigler J. (2002), *Redesigning Professional Development, Creating a Knowledge Base for Teaching: Educational Leadership*, Volume 59 ,Pages 6-11
155. Suchman, L. (1988). *Plans and Situated Actions: The Problem of Human/Machine Communication*. Cambridge, UK: Cambridge University Press
156. Sugrue B. (2002), *Problems of Bloom's Taxonomy* , [online] http://www.performancexpress.org/0212/mainframe0212.html [last visited 12/3/2008]
157. Suraweera, P. and Mitrovic, A. (2002), *KERMIT: a Constraint-based Tutor for Database Modeling*. In: *Cerri, S., Gouarderes, G. and Paraguacu, F. (eds.) Proc. 6th Int. Conf on Intelligent Tutoring Systems ITS*, Biarritz, France, LCNS 2363, pp. 377-387, [online] http://citeseer.ist.psu.edu/cache/papers/cs2/10/http:zSzzSzwww.pramu.orcon.net.nzzSzpaperszSzsuraweera-its2002.pdf/suraweera02kermit.pdf [last visited 22/4/2008]
158. Suraweera, P., Mitrovic, A. (2004), *An Intelligent Tutoring System for Entity Relationship Modeling. Artificial Intelligence in Education*, 14, 375-417.
159. Suraweera P., Mitrovic A. and Martin B. (2005); *A Knowledge Acquisition System for Constraint-based Intelligent Tutoring Systems*; Intelligent Computer Tutoring

Group Department of Computer Science, University of Canterbury Private Bag 4800, Christchurch, New Zealand [online] http://citeseer.ist.psu.edu/cache/papers/cs2/10/http:zSzzSzwww.pramu.orcon.net.nzzSzpaperszSzsuraweera-aied2005.pdf/a-knowledge-acquisition-system.pdf [last visited 15/4/2008]

160. Suraweera P., Mitrovic A. and Martin B. (2007); *Constraint Authoring System: An Empirical Evaluation*; Intelligent Computer Tutoring Group, Department of Computer Science, University of Canterbury, Private Bag 4800, Christchurch, New Zealand [online] http://www.cosc.canterbury.ac.nz/tanja.mitrovic/Suraweera-AIED07.pdf [last visited 15/4/2008]

161. Suraweera P., Mitrovic A. and Martin B. (2004), *The role of domain ontology in knowledge acquisition for ITSs* ;Intelligent Computer Tutoring Group, Department of Computer Science, University of Canterbury Private Bag 4800, Christchurch, New Zealand [online] http://www.cosc.canterbury.ac.nz/tanja.mitrovic/ITS04.pdf [last visited 15/4/2008]

162. Suthers, D. (1992). *Answering Student Queries: Functionality and Mechanisms*. In *Proceedings of the 2nd International Conference on ITS (ITS-92),* Montreal, June.

163. Tabor B. (2002), *LearnVisualStudio.NET* at http://www.learnvisualstudio.net/

164. Tennyson, R. D. and Buttrey, T. (1980). *Advisement and Management Strategies as Design Variables in Computer-Assisted Instruction*. *Educational Communication and Technology Journal*, 28(3): 169- 176, 1980.

165. Tennyson, R. D., Park, O. C., and Christensen, D. L. (1985). *Adaptive Control of Learning Time and Content Sequence in Concept Learning Using Computer-Based Instruction*. *Journal of Educational Psychology*, 77: 481-491, 1985.

166. Twidale, M. B. (1990), *Knowledge Acquisition for Intelligent Tutoring Systems*; Computing Department, University of Lancaster, Lancaster, LA1 4YR, UK [online] http://citeseer.ist.psu.edu/cache/papers/cs/421/ftp:zSzzSzftp.comp.lancs.ac.ukzSzpubzSzreportszSz1993zSzCSCW.13.93.pdf/twidale90knowledge.pdf [last visited 12/4/2008]

167. Urban-Lurain M. (2002), *Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology*. [online] http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm [last visited 12/4/2008]

168. Veen, Jeffrey (2001). *The art and science of Web design. Indianapolis, Ind.: New Riders.* ISBN: 0789723700 [online] http://mirror.randomfoo.net/books/artsci/veen-artsci.pdf [last visited 25/4/2008]

169. Virvou M., Moundridou M.(2000), *A Web Based Authoring Tool for Algebra Related Intelligent Tutoring Systems*; Department of Informatics, University of Piraeus, Greece; Educational Technology and Society 3(2) 2000, ISSN 1436-4522.[online] http://citeseer.ist.psu.edu/cache/papers/cs/15706/http:zSzzSzifets.massey.ac.nzzSzperiodicalzSzvol_2_2000zSzvirvou.pdf/virvou00webbased.pdf [last visited 10/4/2008]

170. Warendorf, K. (1997) , *ADIS - An Animated Data Structure Intelligent Tutoring System orPutting an Interactive Tutor* [ONLINE] http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Warendorf/Warendorf.html [last visited 15/4/2008]

171. Warendorf K., Tan C., ADIS (1997)- *An Animated Data Structure Intelligent Tutoring System or Putting an Interactive Tutor on the WWW*, Proceedings of the workshop: *Intelligent Educational Systems on the World Wide Web*, 8th World Conference of the AIED Society, Japan [online] http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Warendorf/Warendorf.html [last visited 15/5/2008]
172. Weber, G., and Brusilovsky, P. (2001). *ELM-ART: An Adaptive Versatile System for Web-based Instruction. International Journal of Artificial Intelligence in Education*, 12, 351-383. [online] http://www.sis.pitt.edu/~peterb/papers/JAIEDFinal.pdf [last visited 2/5/2008]
173. Welsh, A.L. (1978) *Multiple choice objective tests*. In P. Saunders, A.L. Welsh, and W.L. Hansen (Eds.), *Resource manual for teacher training programs in Economics (pp 191-228).* New York: Joint Council on Economic Education.
174. Wenger E. (1987), *Artificial Intelligence And Tutoring Systems; Computational and Cognitive Approaches to the Communication of Knowledge*
175. Wikipedia, the free encyclopedia website, http://en.wikipedia.org/wiki/
176. Wikipedia, the free Encyclopedia, *Student's t-test* http://en.wikipedia.org/wiki/Student's_t-test
177. Williams B. (2001), *The Role of External Representations in Intelligent Tutoring System authoring: Supporting localized decision making in a complex and evolving global context*. ESRC Centre for Research in Development, Instruction, and TrainingSchool of Psychology, University of Nottingham, University Park, Nottingham
178. Winkels, R., J. Sandberg, and J. Breuker (1990). *The Coach In EUROHELP: Developing Intelligent Help Systems,* Copenhagen.
179. Yang A., Kinshuk and Patel A. (2002). *A Plug-able Web-based Intelligent Tutoring system.* In S. Wrycza (Ed.) *Proceedings of the Xth European Conference on Information Systems* (June 6-8, 2002, Gdańsk, Poland), Gdansk, Poland ISBN 83-7326-077-3. [online] http://is2.lse.ac.uk/asp/aspecis/20020077.pdf [last visited 2/5/2008]
180. Young, J. D. (1996). *The Effect of Self-Regulated Learning Strategies on Performance in Learner Controlled Computer-Based Instruction*. *Educational Technology Research and Development*, 44: 17-27
181. Yousoof M., Sapiyan M., Kamaludin K. (2006); *Role of Intelligent Tutoring Systems in Education: A case study* [online] http://ickm.upm.edu.my/Parallel%20Session%206/MuhammedYousoof%20and%20MohdSapiyan_Role%20of%20Intelligent%20Tutoring%20Systems%20in%20Education.doc [last visited 15/4/2008]
182. Zimmerman, B. (1990).*Self-Regulated Learning and Academic Achievement: An Overview*. *Educational Psychologist*, 25(1): 3-17.

Appendix A: Educational Theories

In this appendix we explain some educational theories that we considered while creating our tutoring system. It starts with Blooms taxonomy, then Krathwohl Taxonomy after that we talk about Gagné's "*Conditions of Learning*" Theory which we have described briefly in chapter 3.

A1.1 Blooms Taxonomy

Who is Bloom[4]?

Benjamin Bloom was an American educational psychologist .He was born in 21 February 1913 and died in September 13, 1999. Benjamin Bloom headed a group of Cognitive psychologists at the University of Chicago and made significant contributions to the classification of educational objectives and the theory of mastery learning. His group developed a taxonomic hierarchy of cognitive-driven behavior which very important to learning and measurable capability

His research focus was on the study of educational objectives. He proposed that any given task favors one of three psychological domains:

- Cognitive: deals with learners abilities to process and utilize information in a meaningful way.

---

[4] Summarized from Wikipedia, the free encyclopedia - http://en.wikipedia.org/wiki/Main_Page

- Affective: is concerned with the attitudes and feelings that result from the learning process.
- or psychomotor. Is about manipulative or physical skills.

Bloom's taxonomy provides structure in which to categorize instructional objectives and instructional assessment. His taxonomy was designed to help educators and Instructional Designers to classify instructional objectives and goals. The foundation of his taxonomy was based on the idea that not all learning objectives and outcomes are equal. For example, memorization of facts, while important, is not the same as the learned ability to analyze or evaluate. In the absence of a classification system (i.e., a taxonomy), educators and Instructional Designers may choose, for example, to emphasize memorization of facts (which make for easier testing) than emphasizing other (and likely more important) learned capabilities.

Bloom's Taxonomy of Cognitive Domain(Bloom 1956)[5]:

Bloom identified six levels within the cognitive domain (Mason G.) as in Figure A1

- Knowledge
- Comprehension
- Application
- Analysis
- Synthesis
- Evaluation

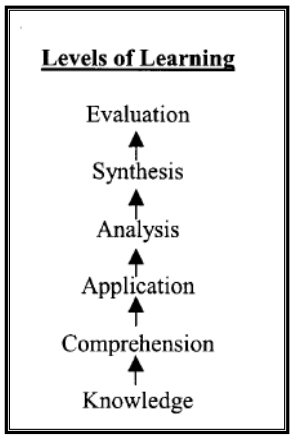


Figure A1

Knowledge: refers to remembering of previously learned material. It represents the lowest level of learning outcomes in the cognitive domain.

Comprehension is the ability to grasp the meaning of material by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas. Translating problems

[5] Summarized from: Reference: *Major categories in the cognitive domain of the taxonomy of educational objectives* (Bloom, 1956). : http://classweb.gmu.edu/ndabbagh/Resources/Resources2/bloomstax.htm

from words to numbers is an example. These learning outcomes go one step beyond the simple remembering of material, and represent the lowest level of understanding.

Application is the ability to use learned material in new and concrete situations. Using new knowledge to solve problems to new situations.The application of such things as rules, methods, concepts, principles, laws, and theories is a good example.

Analysis is the ability to break down material into its component parts by identifying motives or causes so that its organizational structure may be understood. This may include the identification of the parts, analysis of the relationships between parts, and recognition of the organizational principles involved besides make inferences and find evidence to support generalizations. Learning outcomes require an understanding of both the content and the structural form of the material.

Synthesis is the ability to compile and put parts together in a different way to form a new concept or propose new solutions. This may involve the production of a unique communication (theme or speech), a plan of operations (research proposal), or a set of abstract relations (scheme for classifying information). Learning outcomes stress creative behaviors, with major emphasis on the formulation of new patterns or structures.

Evaluation is the ability to present and defend by making judgments about the value of material (statement, novel, poem, research report) for a given purpose based on clearly defined internal or external set of criteria. Learning outcomes in this area are highest in the cognitive hierarchy (Mason G.)

Problems of (critique on) Bloom's Taxonomy (Sugrue B.; Bereiter C. and Scardamalia M. 1998; Merrill 1994; Moore D.S. 1982; Bloom B.S. et al. 1956)

While there are still true believers and keen users of Bloom's taxonomy others have some critique on the taxonomy. Some critique on Bloom's Taxonomy's (cognitive domain) agrees on the existence of these six categories, but questioning the existence of a sequential, hierarchical link (Paul, R. (1993)). Paul (1993) argues that while the distinctions in cognitive levels may be important, there is not necessarily a sequential, hierarchical link between the levels since 'the categories themselves are not independent but interdependent' (p. 375). The revised edition of Bloom's taxonomy has moved Synthesis in higher order than Evaluation. Others agree on that the three lowest levels are hierarchically ordered, but consider the three higher levels as parallel. Some say that it is better sometimes to move to Application before introducing Concepts (most likely related to the method of Problem Based Learning).

The following is another more detailed critique of the theory:

Invalidity

When Bloom's taxonomy was developed (almost 50 years ago) educators and researches didn't have enough information and knowledge about cognitive processes involved in learning and performance. The Bloom's taxonomy categories are not supported by any research on learning. The only distinction that is supported by research is the distinction between declarative/conceptual knowledge and procedural knowledge.

Unreliability

Learning objective might be classified into either of the two lowest levels (knowledge or comprehension) or into any of the four highest levels (application, analysis, synthesis, or evaluation) by different designers which makes application consistency of Bloom's taxonomy across several designers/developers is impossible. This means that there is no reliability in what comprises instruction or assessment that targets different/separate levels. A better reliable approach might be to separate objectives and practice/assessment items into:

- declarative/conceptual knowledge
- Task performance/procedural knowledge.

Impracticality

Everything above the "knowledge" level in Bloom's taxonomy is usually treated as "higher-order thinking" which reduces the taxonomy in fact into to two levels.

In the following we will discuss two alternatives to the taxonomy:

1. The Content-by-Performance Alternative

Recent taxonomies of objectives and learning object strategies distinguish among:

- Types of content (usually facts, concepts, principles, procedures, and processes) as well as
- Levels of performance (usually remember and use).

This content-by-performance approach leads to general prescriptions for informational content and practice/assessment (Sugrue B.) such as those presented in Table A1 (from Sugrue B.).

Content Type	Information to Present (Regardless of Level of Performance	Practice/Assessment (Depending on Level of Performance)	
		Remember	**Use**
Fact	the fact	recognize or recall the fact	recognize or recall during task performance
Concept	the definition, critical attributes, examples, non-examples	recognize or recall the definition or attributes	Identify, classify, or create examples
Principle/ Rule	the principle/rule, examples, analogies, stories	recognize, recall, or explain the principle	decide if the principle applies, predict an event, apply the principle to solve a problem
Procedure	list of steps, demonstration	recognize, recall, or reorder the steps	perform the steps
Process	description of stages, inputs, outputs, diagram, examples, stories	recognize, recall, or reorder the stages	identify origins of problems in the process; predict events in the process; solve problems in the process

Table A1: Information and Practice Based on Content-Performance (adapted from Sugrue B.)

2. The Pure Performance Alternative

A more radical approach would be to have no taxonomy at all. It is simply assumes that all "performance" objectives are at the use level and that learners will practice or be assessed on the particular performance in representative task situations. All objectives and sub-objectives can be treated as performance objectives without further classification (Sugrue B.).

A1.2 Krathwohl Taxonomy

Who is David Krathwohl[6]?

David R. Krathwohl is currently the Hannah Hammond Professor of Education Emeritus at Syracuse University. He has made important contributions to the field of educational psychology. He Co-authored the *Taxonomy of Educational Objectives*, with Benjamin Bloom which is known as Bloom's taxonomy.

Krathwohl's Taxonomy of Affective Domain

Krathwohl's affective domain taxonomy is perhaps the best known of any of the affective taxonomies. "The taxonomy is ordered according to the principle of internalization as in Figure A2 (Mason G.; Krathwohl et al. 1956, 1964; Seels and Glasgow 1990).

Figure A2

Skills in the affective domain describe the way people react emotionally and their ability to feel and understand other people emotions. Affective objectives typically target the awareness and growth in attitudes, emotion, and feelings.

The taxonomy identified 5 levels within the affective domain:

- Receiving
- Responding
- Valuing
- Organizational
- Characterization

[6] Summarized from Wikipedia, the free encyclopedia - http://en.wikipedia.org/wiki/Main_Page

Receiving is being aware of or attending to something in the environment which means that learners passively pay attention. It is the lowest level .Without this level no learning can occur. Examples include: to differentiate, to accept, to listen (for), to respond to.

Responding is showing some new behaviors as a result of experience such as The learner actively participates in the learning process. Examples are: to comply with, to follow, to commend, to volunteer, to spend leisure time in, to acclaim.

Valuing is showing some definite involvement or commitment. The learner attaches a value to an object, phenomenon, or piece of information Examples include: to increase measured proficiency in, to relinquish, to subsidize, to support, to debate.

Organization is integrating a new value into one's general set of values, giving it some ranking among one's general priorities, which means that the learner can put together different values, information, and ideas and accommodate them within his/her own schema; comparing, relating and elaborating on what has been learned. Examples are: to discuss, to theorize, to formulate, to balance, to examine.

Characterization by value or value set is acting consistently in accordance with the new values he or she has internalized. The learner has held a particular value or belief that now exerts influence on his/her behavior so that it becomes a characteristic. This is the highest level. Examples include: to revise, to require, to be rated high in the value, to avoid, to resist, to manage, to resolve.

A1.3 Gagné's "*Conditions of Learning*" Theory

Who is Gagné [7]?

Robert Gagne was born in 1916 in North Andover, MA. He is currently a professor in the Department of Education Research at Florida State University in Tallahassee. He is considered as an experimental psychologist who is concerned with learning and instruction. And he has developed a learning theory called the "*Conditions of Learning*".

The "*Conditions of Learning*" Theory[8]

Gagne suggests that learning tasks for intellectual skills can be organized in a hierarchy according to complexity:

- stimulus recognition,
- response generation,
- procedure following,
- use of terminology,
- discriminations,
- concept formation,
- rule application,
- and problem solving.

The primary significance of the hierarchy is to identify prerequisites that should be completed to facilitate learning at each level. Prerequisites are identified by doing a task analysis of a learning/training task. Learning hierarchies provide a basis for the sequencing of instruction. As show in table A2, the theory outlines nine instructional events and corresponding cognitive processes:

Instructional event	***Example***
(1) gaining attention (reception)	variety of computer generated shapes
(2) informing learners of the objective (expectancy)	Question: "What is a square?"
(3) stimulating recall of prior learning	review definition of square

[7] Summarized from http://www.psy.pdx.edu/PsiCafe/KeyTheorists/Gagne.htm
[8] Summarized from http://tip.psychology.org/gagne.html

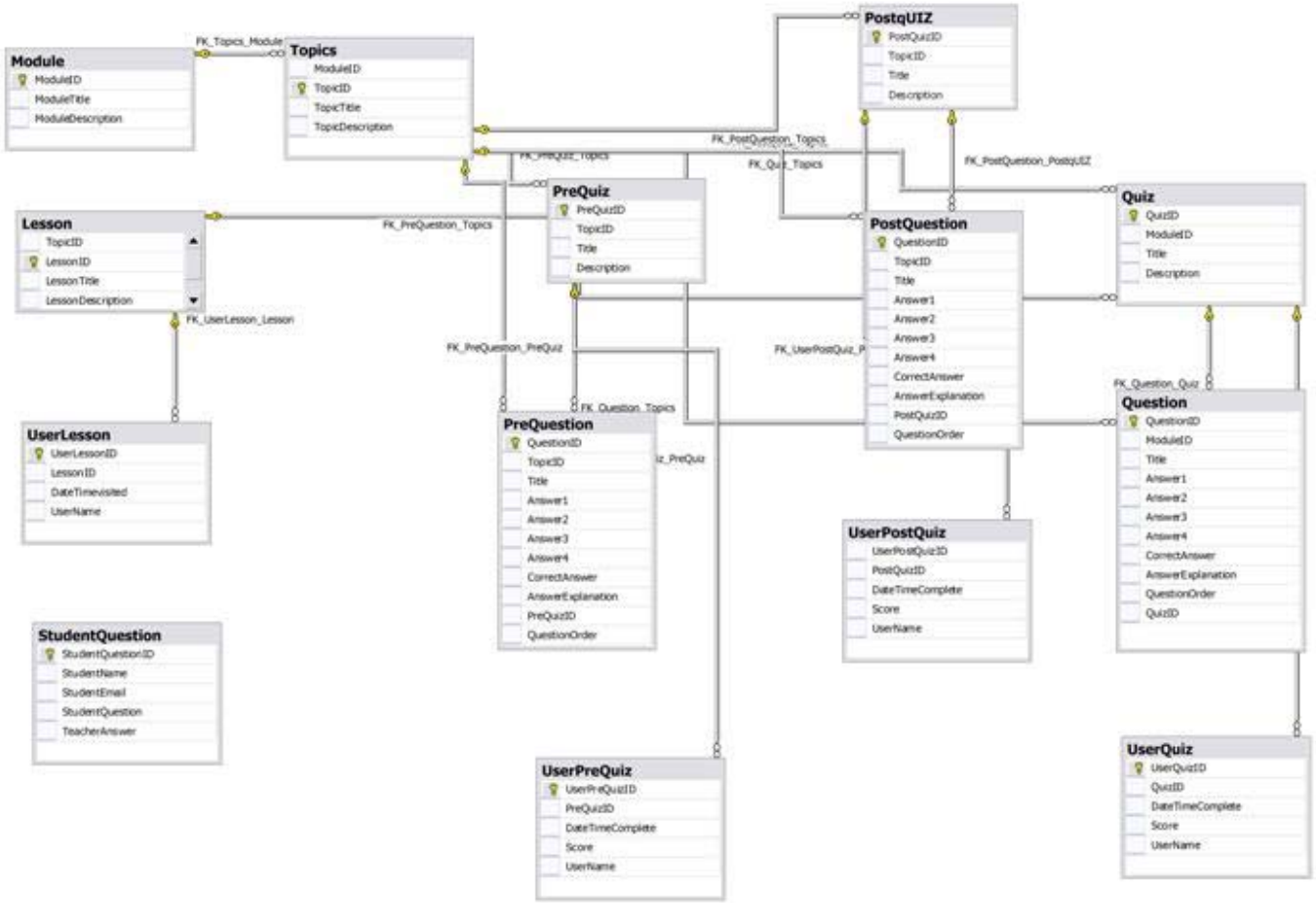
(retrieval)	
(4) presenting the stimulus (selective perception)	give definition of a square
(5) providing learning guidance (semantic encoding)	show example of how to draw a square
(6) eliciting performance (responding)	Ask students to draw 4 different size squares
(7) providing feedback (reinforcement)	Check students work for all drawn squares with feedback of correct/ incorrect and why?
(8) assessing performance (retrieval)	provide assessment grade and suggest readings to correct mistakes
(9) Enhancing retention and transfer (generalization).	Show different shapes again and ask students to identify squares

Table A2: Examples of events for each category of learning outcomes From Gagne (1985, chapter 12).

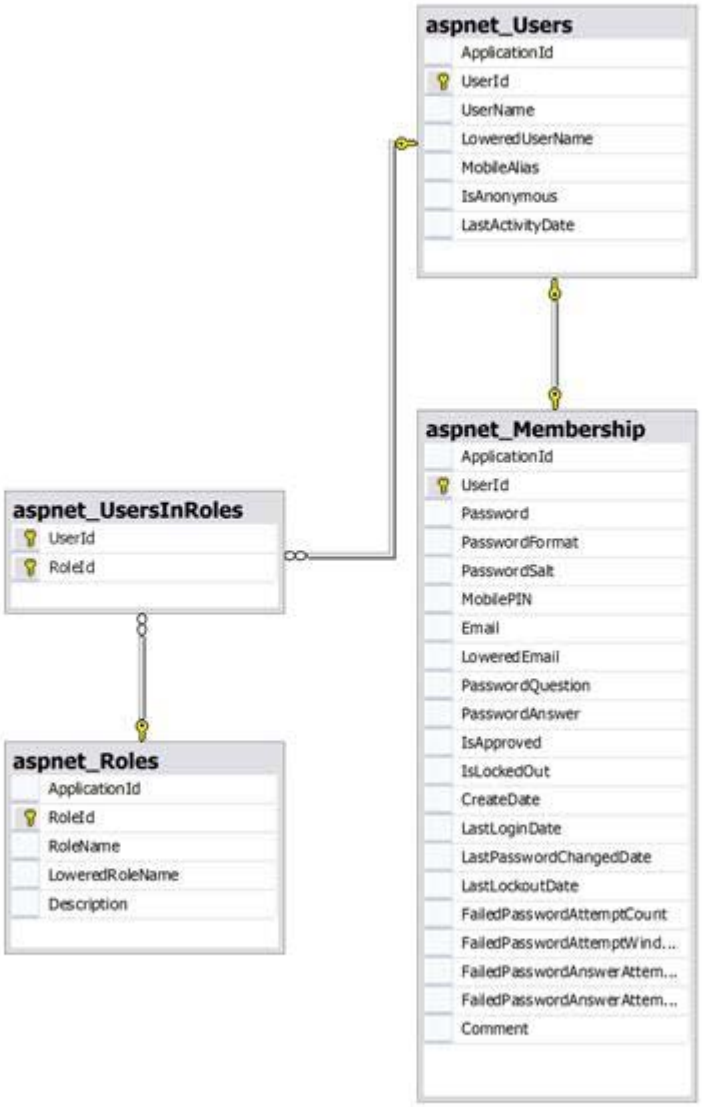
Principles:

1. Different outcomes needs different types of instructions
2. The specific processes that comprise instructional measures are different for each different type of learning outcome.
3. Sequence of instruction and what intellectual skills to be learned are defined by learning hierarchies.
4. Measures of learning control the learner in ways that make up the conditions of learning.

Appendix B: System tables and relations – Diagram 1



Appendix B2: System tables and relations – Diagram 2



Appendix C: Example of an intelligent practice/lesson in CLIPS

```
;  About THE PROGRAM
;  <<<<<<<<>>>>>>>>>
;
; The program is designed to help students understand Basic Concepts of Information Technology and be
able to pass ICDL
; Module 1 exam.
;
; First the program gives a list of all available topics for students to pick up from as their progress.
;
; then the program ask questions and take the student step by step as per student response.
; The program suggests what lessons the student need to go and read and customizes the path according to
the student level
; and prompt the user for:
;
; 1. Suggested readings
; 2. emphasizes on student weaknesses
;
;
;
;
; I have used ICDL official syllabus when planned my lessons and my intelligent tutor.
;
;


;
;''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
;Initializes the program
;
;''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

(defrule init
 (declare (salience 9980))
 ?x <- (initial-fact) =>
 (retract ?x)
 (assert (need to-read-the-lesson n))
 (assert (start)))


;''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
;To end the program and call the init rule to
;            reset the program
;
;''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
(defrule end1
 (declare (salience 9200))
```

```
 ?s <- (stop)
 =>
 (retract ?s)
 (reset)
 (halt))

;"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
;
;Gives the opening message and asks the student to select a lesson..
;
;"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
;

(defrule begin
 (declare (salience -9000))
 ?x <- (start) =>
 (printout t crlf crlf crlf
            "This is Lesson 2 - Hardware ...")
 (retract ?x)
 (assert (Lesson number 2)))


;=================================================================================
=========================

;"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
;
;Lesson 2- Hardware
;
;"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
;


(defrule Lesson2-Question1-Question
 (declare (salience 1500))
 (Lesson number 2) =>
 (printout t crlf crlf
            " The three main parts of a CPU are....?" crlf

            "  1) ALU, immediate access memory, control unit " crlf
            "  2) ALU, RAM, control unit"crlf

            "  3) system box, control unit, VIDEO CARD" crlf

            "  4) ALU, ROM, control unit " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer1 =(read))))

;=================================================================================
(defrule Lesson2-Question1-answer1
 ?x <- (Lesson2 answer1 1) =>
 (retract ?x)


           (printout t crlf crlf crlf
           "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question2 y))

)
```

```
;==========================================================================
(defrule Lesson2-Question1-answer2
 ?x <- (Lesson2 answer1 2) =>
 (retract ?x)
 (printout t crlf crlf
          "This is a wrong answer... CPU uses The immediate access memory ( where the data used by the
ALU is stored) not RAM." crlf crlf
          "So the right answer is: ALU, immediate access memory, control unit" crlf crlf)

(assert (Lesson2 question2 y))

)

;==========================================================================

 (defrule Lesson2-Question1-answer3
 ?x <- (Lesson2 answer1 3) =>
 (retract ?x)
 (printout t crlf crlf
          "This is a wrong answer... The three main parts of a CPU are the arithmetic logic unit (ALU), "crlf
          "the immediate access memory and the control unit.)." crlf crlf
          "So the right answer is: ALU, immediate access memory, control unit" crlf crlf)


(assert (Lesson2 question2 y)))

;==========================================================================
(defrule Lesson2-Question1-answer4
 ?x <- (Lesson2 answer1 4) =>
 (retract ?x)
 (printout t crlf crlf
          "This is a wrong answer... CPU uses The immediate access memory ( where the data used by the
ALU is stored) not ROM." crlf crlf
          "So the right answer is: ALU, immediate access memory, control unit" crlf crlf)



(assert (Lesson2 question2 y)))


;==========================================================================

(defrule Lesson2-Question2-Question

?x <- (Lesson2 question2 y) =>
(retract ?x)
 (printout t crlf crlf
          " ALU stands for....?" crlf

          "  1) Access Logical Utility " crlf
          "  2) ACcess Local Utilities "crlf

    "  3) Arithmetic Logic Unit" crlf

          "  4) Arithmetic Logic Utility " crlf crlf
```

```
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer2 =(read))))

;==========================================================================

(defrule Lesson2-Question2-answer4
(or(Lesson2 answer2 4)(Lesson2 answer2 1)(Lesson2 answer2 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            "This is a wrong answer... ALU stands for Arithmetic Logic Unit " crlf crlf crlf)
(assert (Lesson2 question3 y)))

;==========================================================================
(defrule Lesson2-Question2-answer3
 ?x <- (Lesson2 answer2 3) =>
 (retract ?x)


            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question3 y))

)

;==========================================================================

(defrule Lesson2-Question3-Question

?x <- (Lesson2 question3 y) =>
(retract ?x)
 (printout t crlf crlf
            " RAM stands for....?" crlf

            "  1) Read Access memory " crlf
            "  2) Read application memory "crlf

            "  3) Random access memory" crlf

            "  4) Ready always memory " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer3 =(read))))

;==========================================================================
(defrule Lesson2-Question3-answer1
(or(Lesson2 answer3 4)(Lesson2 answer3 1)(Lesson2 answer3 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            "This is a wrong answer... RAM stands for Random access memory " crlf crlf
            " It is very important to understand and remeber abbreviations stands for what" crlf crlf crlf)



(assert (Lesson2 question4 y)))

;==========================================================================
```

```
(defrule Lesson2-Question3-answer3
 ?x <- (Lesson2 answer3 3) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf

crlf )
(assert (Lesson2 question6 y))

)
;==================================================================================

(defrule Lesson2-Question4-Question

?x <- (Lesson2 question4 y) =>
(retract ?x)
 (printout t crlf crlf
            " ROM stands for....?" crlf

            "  1) Read Only memory " crlf
            "  2) Read Object memory "crlf

            "  3) Random Objects memory" crlf

            "  4) Ready Objectives memory " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer4 =(read))))

;==================================================================================
 (defrule Lesson2-Question4-answer2
(or(Lesson2 answer4 4)(Lesson2 answer4 3)(Lesson2 answer4 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            "This is a wrong answer... ROM stands for Read Only memory " crlf crlf
            " It is very important to understand and remeber abbreviations stands for what ..." crlf crlf
            " It sounds that you need to work on this area better ... " crlf)

(assert (Lesson2 question5 y)))

;==================================================================================
(defrule Lesson2-Question4-answer1
 ?x <- (Lesson2 answer4 1) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question6 y)))
;==================================================================================
(defrule Lesson2-Question5-Question

?x <- (Lesson2 question5 y) =>
(retract ?x)
 (printout t crlf crlf
            " CPU stands for....?" crlf
```

```
            "  1) Central progressing Unit " crlf
            "  2) Central Processing Unit "crlf
            "  3) Central Programming Unit" crlf
            "  4) Central Paging Unit" crlf crlf
            " Please enter a number from 1 to 4 => ")

 (assert (Lesson2 answer5 =(read))))

;==========================================================================
(defrule Lesson2-Question5-answer1
(or(Lesson2 answer5 4)(Lesson2 answer5 3)(Lesson2 answer5 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... CPU stands for Central Processing Unit " crlf crlf
            " It is very important to understand and remeber abbreviations stands for what ..." crlf crlf
            " It sounds that you need to work on this area better ... " crlf
            " I strongly suggest that you read and learn the lesson abbreviations  .." crlf crlf
            " This is lesson 2 :Hardware" crlf crlf
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  " )
(assert (Lesson2 question6 =(read))))

;==========================================================================
(defrule Lesson2-Question5-answer2
 ?x <- (Lesson2 answer5 2) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question6 y)))

;==========================================================================

(defrule Lesson2-Question6-Question

?x <- (Lesson2 question6 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which of the following holds the instructions that the computer needs to know immediately
upon starting up?" crlf

            "  1) Virtual memory " crlf
            "  2) Immediate access memory "crlf
            "  3) Random Access Memory" crlf
            "  4) ROM " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer6 =(read))))

;==========================================================================
(defrule Lesson2-Question6-answer1
(or(Lesson2 answer6 2)(Lesson2 answer6 3)(Lesson2 answer6 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " Read-only memory (ROM) is stored on ‘memory chips’. With these chips," crlf
            " data is permanently burned into them" crlf
```

```
            " when they are manufactured and usually cannot be altered afterwards." crlf
            " The information and instructions are there " crlf
            " before you turn the computer on and still there when the power is off. " crlf
            " Unlike RAM, the computer cannot place data in ROM. It can only read from it, thus giving it its
name." crlf crlf
            " This kind of memory is typically used to store the instructions that the computer needs to start
up" crlf
            " so that it can run a self-check and detect attached equipment." crlf crlf crlf )

(assert (Lesson2 question7 y)))

;==================================================================================
==========================
(defrule Lesson2-Question6-answer4
 ?x <- (Lesson2 answer6 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question9 y)))

;==================================================================================
(defrule Lesson2-Question7-Question

?x <- (Lesson2 question7 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following statements about ROM is true?" crlf

            "  1) ROM is a computer peripheral  " crlf
            "  2) One buys extra ROM for one's computer in order to increase its performance "crlf
            "  3) ROM holds key information needed for the computer to operate immediately on power-up"
crlf
            "  4) ROM is memory that can be both read and written to " crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer7 =(read))))

;==================================================================================
 (defrule Lesson2-Question7-answer1
(or(Lesson2 answer7 2)(Lesson2 answer7 4)(Lesson2 answer7 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer. The right answer is: "crlf
            " ROM holds key information needed for the computer to operate immediately on power-up" crlf
            " Read-only memory (ROM) is stored on ‘memory chips’. With these chips," crlf
            " data is permanently burned into them" crlf
            " when they are manufactured and usually cannot be altered afterwards." crlf
            " The information and instructions are there " crlf
            " before you turn the computer on and still there when the power is off. " crlf
            " Unlike RAM, the computer cannot place data in ROM. It can only read from it, thus giving it its
name." crlf crlf
            " This kind of memory is typically used to store the instructions that the computer needs to start
up" crlf
            " so that it can run a self-check and detect attached equipment."   crlf crlf crlf )
```

```
(assert (Lesson2 question8 y)))

;==========================================================================
(defrule Lesson2-Question7-answer3
 ?x <- (Lesson2 answer7 3) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question9 y)))
;==========================================================================

(defrule Lesson2-Question8-Question

?x <- (Lesson2 question8 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following statements about ROM is true?" crlf

            "  1) ROM is a form of backing store  " crlf
            "  2) ROM will retain data when the computer is switched off "crlf
            "  3) ROM will lose data when the computer is switched off" crlf
            "  4) ROM is memory that can be both read and written to " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer8 =(read))))

;==========================================================================
 (defrule Lesson2-Question8-answer1
(or(Lesson2 answer8 4)(Lesson2 answer8 3)(Lesson2 answer8 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is ROM will retain data when the computer is
switched off" crlf
             crlf crlf
            " Read-only memory (ROM) is stored on 'memory chips'. With these chips," crlf
            " data is permanently burned into them" crlf
            " when they are manufactured and usually cannot be altered afterwards." crlf
            " The information and instructions are there " crlf                    " before you turn the computer on
and still there when the power is off. " crlf
            " Unlike RAM, the computer cannot place data in ROM. It can only read from it, thus giving it its
name." crlf crlf
            " This kind of memory is typically used to store the instructions that the computer needs to start
up" crlf
            " so that it can run a self-check and detect attached equipment." crlf crlf crlf
            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Memory..." crlf
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  " )

(assert (Lesson2 question9 =(read))))

;==========================================================================
(defrule Lesson2-Question8-answer2
 ?x <- (Lesson2 answer8 2) =>
```

```
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question9 y)))


;==========================================================================

(defrule Lesson2-Question9-Question

?x <- (Lesson2 question9 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following best describes the Random Access Memory of a computer?" crlf

            "  1) RAM is the user's program area  " crlf
            "  2) RAM is the fastest memory in the computer system "crlf
            "  3) RAM is permanent" crlf
            "  4) RAM is backing store " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer9 =(read))))

;==========================================================================
(defrule Lesson2-Question9-answer2
(or(Lesson2 answer9 4)(Lesson2 answer9 3)(Lesson2 answer9 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is :RAM is the user's program area" crlf
             crlf crlf
            " Random access memory (RAM) is the main working memory of the computer." crlf
            " RAM is measured in megabytes, so a computer could be said to have 512 MB of RAM" crlf
            " RAM is empty when the computer is first switched on." crlf
            " The system software, such as Windows, is loaded into it so that the computer can carry out all "
crlf
            " the functions that make it work. " crlf
            " If you are using a word processing program to produce a document," crlf
            "  the word processing program is first loaded into RAM and the document you are preparing " crlf
            " is also stored in RAM as you work on it." crlf crlf
            " Should the computer be switched off before your document is saved or should there be a power
cut," crlf
            " everything in RAM is lost and you will have to start all over again when the power is restored."
crlf
            )

(assert (Lesson2 question10 y)))

;==========================================================================
(defrule Lesson2-Question9-answer1
 ?x <- (Lesson2 answer9 1) =>
 (retract ?x)


            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
```

```
crlf )
(assert (Lesson2 question12 y)))

;==========================================================================
(defrule Lesson2-Question10-Question

?x <- (Lesson2 question10 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following describes the RAM of a computer?" crlf
            "  1) RAM is read-access memory  " crlf
            "  2) RAM is read and write memory  "crlf
            "  3) RAM is write-only memory" crlf
            "  4) RAM is a computer peripheral " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer10 =(read))))

;==========================================================================
 (defrule Lesson2-Question10-answer1
(or(Lesson2 answer10 4)(Lesson2 answer10 3)(Lesson2 answer10 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
           " This is a wrong answer... The right answer is: RAM is read and write memory" crlf
            crlf crlf
           " Random access memory (RAM) is the main working memory of the computer." crlf
           " RAM is measured in megabytes, so a computer could be said to have 512 MB of RAM" crlf
           " RAM is empty when the computer is first switched on." crlf
           " The system software, such as Windows, is loaded into it so that the computer can carry out all "
crlf
           " the functions that make it work. " crlf
           " If you are using a word processing program to produce a document," crlf
           "  the word processing program is first loaded into RAM and the document you are preparing " crlf
           " is also stored in RAM as you work on it." crlf crlf
           " Should the computer be switched off before your document is saved or should there be a power
cut," crlf
           " everything in RAM is lost and you will have to start all over again when the power is restored."
crlf
           )

(assert (Lesson2 question11 y)))

;==========================================================================
(defrule Lesson2-Question10-answer2
 ?x <- (Lesson2 answer10 2) =>
 (retract ?x)
           (printout t crlf crlf crlf "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question12 y)))

;==========================================================================

(defrule Lesson2-Question11-Question

?x <- (Lesson2 question11 y) =>
(retract ?x)
 (printout t crlf crlf
```

```
            " Which one of the following statements about RAM is true?" crlf

            "  1) RAM retains its data when the computer power is off  " crlf
            "  2) RAM is a computer peripheral  "crlf
            "  3) RAM contains computer start-up instructions " crlf
            "  4) RAM is a form of read and write memory " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer11 =(read))))

;==========================================================================
 (defrule Lesson2-Question11-answer1
(or(Lesson2 answer11 2)(Lesson2 answer11 3)(Lesson2 answer11 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: RAM is a form of read and write memory" crlf
             crlf crlf
            " Random access memory (RAM) is the main working memory of the computer." crlf
            " RAM is measured in megabytes, so a computer could be said to have 512 MB of RAM" crlf
            " RAM is empty when the computer is first switched on." crlf
            " The system software, such as Windows, is loaded into it so that the computer can carry out all "
crlf
            " the functions that make it work. " crlf
            " If you are using a word processing program to produce a document," crlf
            "  the word processing program is first loaded into RAM and the document you are preparing " crlf
            " is also stored in RAM as you work on it." crlf crlf
            " Should the computer be switched off before your document is saved or should there be a power
cut," crlf
            " everything in RAM is lost and you will have to start all over again when the power is restored."
crlf crlf crlf

            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Memory..." crlf
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  " )


(assert (Lesson2 question12 =(read))))

;==========================================================================
(defrule Lesson2-Question11-answer4
 ?x <- (Lesson2 answer11 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question12 y)))

;==========================================================================

(defrule Lesson2-Question12-Question
?x <- (Lesson2 question12 y) =>
(retract ?x)
 (printout t crlf crlf
             " Which one of the following measurements of computer memory is the second largest in size?"
crlf
```

```
            "  1) Terabyte  " crlf
            "  2) Kilobyte  "crlf
            "  3) Gigabyte " crlf
            "  4) Megabyte " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer12 =(read))))

;==========================================================================
 (defrule Lesson2-Question12-answer1
(or(Lesson2 answer12 2)(Lesson2 answer12 4)(Lesson2 answer12 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Gigabyte" crlf
             crlf crlf
            " Computer memory can be expressed using the following units." crlf
            " Byte: 8 bits " crlf
            " Each character that the user inputs is stored as a byte." crlf  crlf
            " If you want to store the character “A”, you need 1 byte of memory " crlf
            " If you want to store the characters “AB2”, you need 3 bytes of memory " crlf
            " Kilobyte(KB): 1024 bytes " crlf
            " Megabyte(MB): 1024 kilobytes" crlf
            " Gigabyte(GB): 1024 megabytes " crlf
            " Terabyte(TB): Approximately a thousand gigabytes or a thousand billion bytes  " crlf
            )

(assert (Lesson2 question13 y)))

;==========================================================================
(defrule Lesson2-Question12-answer3
 ?x <- (Lesson2 answer12 3) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question15 y)))

;==========================================================================

(defrule Lesson2-Question13-Question
?x <- (Lesson2 question13 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following measurements of computer memory is the second largest in size?"
crlf
            "  1) Terabyte  " crlf
            "  2) Kilobyte  "crlf
            "  3) Gigabyte " crlf
            "  4) Megabyte " crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer13 =(read))))

;==========================================================================
(defrule Lesson2-Question13-answer1
(or(Lesson2 answer13 2)(Lesson2 answer13 4)(Lesson2 answer13 1)) =>
 ;(retract ?x)
```

```
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Gigabyte" crlf
             crlf crlf
            " Computer memory can be expressed using the following units." crlf
            " Byte: 8 bits " crlf
            " Each character that the user inputs is stored as a byte." crlf  crlf
            " If you want to store the character “A”, you need 1 byte of memory " crlf
            " If you want to store the characters “AB2”, you need 3 bytes of memory " crlf
            " Kilobyte(KB): 1024 bytes " crlf
            " Megabyte(MB): 1024 kilobytes" crlf
            " Gigabyte(GB): 1024 megabytes " crlf
            " Terabyte(TB): Approximately a thousand gigabytes or a thousand billion bytes  " crlf
            )

(assert (Lesson2 question14 y)))

;==========================================================================
(defrule Lesson2-Question13-answer3
 ?x <- (Lesson2 answer13 3) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question15 y)))

;==========================================================================

(defrule Lesson2-Question14-Question

?x <- (Lesson2 question14 y) =>
(retract ?x)
 (printout t crlf crlf
             " In relation to the units of computer memory which of the following statements is false?" crlf

             "  1) 1000 KB = 1 MB approximately " crlf

             "  2) 100,000 KB = 1 MB approximately  "crlf

             "  3) 1000 Gigabytes = 1 Terabyte approximately " crlf

             "  4) 1,000,000 MB = 1 Terabyte approximately " crlf crlf

             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer14 =(read))))
;==========================================================================
 (defrule Lesson2-Question14-answer1
(or(Lesson2 answer14 3)(Lesson2 answer14 4)(Lesson2 answer14 1)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The false statement is: 100,000 KB = 1 MB approximately " crlf
             crlf crlf
            " Computer memory can be expressed using the following units." crlf
            " Byte: 8 bits " crlf
            " Each character that the user inputs is stored as a byte." crlf  crlf
            " If you want to store the character “A”, you need 1 byte of memory " crlf
```

```
            " If you want to store the characters “AB2”, you need 3 bytes of memory " crlf
            " Kilobyte(KB): 1024 bytes " crlf
            " Megabyte(MB): 1024 kilobytes" crlf
            " Gigabyte(GB): 1024 megabytes " crlf
            " Terabyte(TB): Approximately a thousand gigabytes or a thousand billion bytes  " crlf
            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Memory..." crlf
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  " )
(assert (Lesson2 question15 =(read))))

;==========================================================================
(defrule Lesson2-Question14-answer2
 ?x <- (Lesson2 answer14 2) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question15 y)))

;==========================================================================

(defrule Lesson2-Question15-Question

?x <- (Lesson2 question15 y) =>
(retract ?x)
 (printout t crlf crlf
            " The arithmetic/logic unit performs the following actions" crlf

            "  1) checks data for accuracy " crlf
            "  2) does calculations using addition, subtraction, multiplication, and division  "crlf
            "  3) does logical comparisons, such as equal to, greater than, less than " crlf
            "  4) both calculations and logical comparisons " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer15 =(read))))
;==========================================================================
(defrule Lesson2-Question15-answer1
?x<- (Lesson2 answer15 1) =>
 (retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: ALU does both calculations and logical
comparisons" crlf
             crlf crlf
            " All the arithmetical operations and logical comparisons are carried out in ALU unit." crlf
            " Arithmetical operations refers to operations like addition, subtraction etc and logical operations "
crlf
            " refers to comparisons like finding a larger number, etc." crlf crlf crlf )

(assert (Lesson2 question16 y)))
;==========================================================================

(defrule Lesson2-Question15-answer2
(or(Lesson2 answer15 2)(Lesson2 answer15 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
```

```
            " This is a wrong answer... You have partially answered the Question.." crlf
            " The right answer is: ALU does both calculations and logical comparisons" crlf
             crlf crlf

            " All the arithmetical operations and logical comparisons are carried out in ALU unit." crlf
            " Arithmetical operations refers to operations like addition, subtraction etc and logical operations "
crlf
            " refers to comparisons like finding a larger number, etc." crlf crlf crlf)

(assert (Lesson2 question17 y)))
;==========================================================================================
(defrule Lesson2-Question15-answer4
 ?x <- (Lesson2 answer15 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question18 y)))

;==========================================================================================
(defrule Lesson2-Question16-Question
?x <- (Lesson2 question16 y) =>
(retract ?x)
 (printout t crlf crlf
             " The control unit performs the following actions" crlf
             "  1) The control unit manages and directs instructions flowing into the computer " crlf
             "  2) Controls how many hard disks you have in your computer  "crlf
             "  3) carry out all the arithmetical operations and logical comparisons " crlf
             "  4) is temporary storage area where all the data is fed into the computer" crlf crlf
             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer16 =(read))))


;==========================================================================================
(defrule Lesson2-Question16-answer2
?x<- (Lesson2 answer16 2) =>
 (retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is:  The control unit manages and directs instructions
" crlf
            " flowing into the computer " crlf
             crlf crlf
            " The control unit manages and directs instructions flowing into the computer." crlf
            " It is the control unit that decides what data should be sent to what component of the computer "
crlf
             crlf crlf )

(assert (Lesson2 question17 y)))
;==========================================================================================
(defrule Lesson2-Question16-answer3
?x <- (Lesson2 answer16 3) =>
(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... this is the ALU unit function .." crlf
            " The right answer is: The control unit manages and directs instructions" crlf
             crlf crlf
```

```
            " The control unit manages and directs instructions flowing into the computer." crlf
            " It is the control unit that decides what data should be sent to what component of the computer "
crlf
             crlf crlf )


(assert (Lesson2 question17 y)))
;=================================================================================
(defrule Lesson2-Question16-answer4
?x <- (Lesson2 answer16 4) => =>
 (retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... this is the CPU Internal memory function .." crlf
            " The right answer is: The control unit manages and directs instructions" crlf
             crlf crlf
            " The control unit manages and directs instructions flowing into the computer." crlf
            " It is the control unit that decides what data should be sent to what component of the computer "
crlf
             crlf crlf )


(assert (Lesson2 question17 y)))
;=================================================================================
(defrule Lesson2-Question16-answer1
 ?x <- (Lesson2 answer16 1) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question18 y)))

;=================================================================================

(defrule Lesson2-Question17-Question

?x <- (Lesson2 question17 y) =>
(retract ?x)
 (printout t crlf crlf
             " The CPU Internal memory ..." crlf
             "  1) The control unit manages and directs instructions flowing into the computer " crlf
             "  2) Controls how many hard disks you have in your computer  "crlf
             "  3) carry out all the arithmetical operations and logical comparisons " crlf
             "  4) is temporary storage area where all the data is fed into the computer" crlf crlf

             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer17 =(read))))


;=================================================================================
(defrule Lesson2-Question17-answer2
?x<- (Lesson2 answer17 2) =>
 (retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: It a is temporary storage area where all the data "
crlf
            " is fed into the computer " crlf
```

```
                    crlf crlf

          " CPU Internal Memory is a temporary storage area where all the data is fed into the computer,"
crlf
          " the data being processed, the results and the processing instructions are temporarily stored.." crlf
          " I strongly suggest that you read the lesson thoroughly .." crlf
          " This is lesson 2 : Hardware / Central Processing Unit ..." crlf
          " Do you want to continue to the next topic and later read the lesson? .."crlf
          "==> y or n  "
           crlf crlf )


(assert (Lesson2 question18 =(read))))
;=================================================================================
(defrule Lesson2-Question17-answer3
?x <- (Lesson2 answer17 3) =>
(retract ?x)
 (printout t crlf crlf
          " This is a wrong answer... this is the ALU unit function .." crlf
          " The right answer is: It a is temporary storage area where all the data" crlf
           crlf crlf

          " CPU Internal Memory is a temporary storage area where all the data is fed into the computer,"
crlf
          " the data being processed, the results and the processing instructions are temporarily stored.." crlf

           " I strongly suggest that you read the lesson thoroughly .." crlf
          " This is lesson 2 : Hardware / Central Processing Unit..." crlf
          " Do you want to continue to the next topic and later read the lesson? .."crlf
          "==> y or n  " crlf crlf )

(assert (Lesson2 question18 =(read))))
;=================================================================================
(defrule Lesson2-Question17-answer1
?x <- (Lesson2 answer17 1) => =>
 (retract ?x)
 (printout t crlf crlf
          " This is a wrong answer... this is the Control Unit function .." crlf
          " The right answer is: The control unit manages and directs instructions" crlf
           crlf crlf

          " CPU Internal Memory is a temporary storage area where all the data is fed into the computer,"
crlf
          " the data being processed, the results and the processing instructions are temporarily stored.." crlf
           " I strongly suggest that you read the lesson thoroughly .." crlf
          " This is lesson 2 : Hardware / Central Processing Unit..." crlf
          " Do you want to continue to the next topic and later read the lesson? .."crlf
          "==> y or n  " crlf crlf )

(assert (Lesson2 question18 =(read))))
;=================================================================================
(defrule Lesson2-Question17-answer4
 ?x <- (Lesson2 answer17 4) =>
 (retract ?x)
```

```
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question18 y)))
;==========================================================================

(defrule lesson2-Question18-Question
?x <- (Lesson2 question18 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following is not an input device? ..." crlf
            "  1) Microphone " crlf
            "  2) Digital scanner  "crlf
            "  3) Touchpad " crlf
            "  4) Speakers" crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer18 =(read))))

;==========================================================================
(defrule Lesson2-Question18-answer1
(or(Lesson2 answer18 1)(Lesson2 answer18 2)(Lesson2 answer18 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Speakers" crlf
             crlf crlf
            " A device that is used to input data into a computer for processing is called an input device." crlf
            " Examples of Input devices include:" crlf
            " KEYBOARD, MOUSE , SCANNER, MICROPHONE, JOYSTICK, TOUCHPAD,
TRACKBALL, DIGITAL CAMERA, WEB CAM , LIGHT PEN " crlf
            crlf crlf
            )

(assert (Lesson2 question19 y)))

;==========================================================================
(defrule Lesson2-Question18-answer4
 ?x <- (Lesson2 answer18 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question21 y)))

;==========================================================================

(defrule lesson2-Question19-Question

?x <- (Lesson2 question19 y) =>
(retract ?x)
 (printout t crlf crlf
            " A digital scanner is used to ..." crl
            "  1) detect software problems with the computer  " crlf

            "  2) provide the computer with information on software  "crlf
            "  3) store files too large for a floppy disk " crlf
```

```
            "  4) copy an image from paper to your computer " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer19 =(read))))
;==========================================================================================
(defrule Lesson2-Question19-answer1
(or(Lesson2 answer19 1)(Lesson2 answer19 2)(Lesson2 answer19 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: copy an image from paper to your computer " crlf
             crlf crlf
            " A scanner allows you to scan and transfer documents, pictures, etc. into your computer." crlf
            " It is an examples of Input devices " crlf

            crlf crlf
            )
(assert (Lesson2 question20 y)))

;==========================================================================================
(defrule Lesson2-Question19-answer4
 ?x <- (Lesson2 answer19 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question21 y)))

;==========================================================================================
(defrule lesson2-Question20-Question

?x <- (Lesson2 question20 y) =>
(retract ?x)
 (printout t crlf crlf
            " A Light pen is used to ..." crlf
            "  1) control a computer game " crlf
            "  2) select things on a computer screen  "crlf
            "  3) draw and process computer graphics" crlf
            "  4) copy an image from paper to your computer " crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer20 =(read))))


;==========================================================================================
(defrule Lesson2-Question20-answer1
(or(Lesson2 answer20 1)(Lesson2 answer20 4)(Lesson2 answer20 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: select things on a computer screen  " crlf
             crlf crlf
            " Light Pen A pointing device that inputs data into a computer. It uses a light-sensitive detector"
crlf
            " that allows you to select objects and data on the screen." crlf
            " It is an examples of Input devices " crlf
            crlf crlf
            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Input Devices ..." crlf
```

```
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  "
            )
(assert (Lesson2 question21 =(read))))
;==========================================================================================
(defrule Lesson2-Question20-answer2
 ?x <- (Lesson2 answer20 2) =>
 (retract ?x)

            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question21 y)))

;==========================================================================================
(defrule lesson2-Question21-Question

?x <- (Lesson2 question21 y) =>
(retract ?x)
 (printout t crlf crlf
             " Which of the following is not an output device?" crlf
             "  1) pen plotter " crlf
             "  2) data projector  "crlf
             "  3) dot-matrix printer" crlf
             "  4) keyboard " crlf crlf
             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer21 =(read))))

;==========================================================================================
(defrule Lesson2-Question21-answer1
(or(Lesson2 answer21 1)(Lesson2 answer21 2)(Lesson2 answer21 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: keyboard  " crlf
             crlf crlf
            " An output device is a device used to display information from a computer. Output devices
include" crlf
            " Conputer screen or monitor, printer, plotter, speakers, data projector, Speech Synthesizers, etc."
crlf        crlf crlf
            )
(assert (Lesson2 question22 y)))

;==========================================================================================
(defrule Lesson2-Question21-answer4
 ?x <- (Lesson2 answer21 4) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question24 y)))

;==========================================================================================
 (defrule lesson2-Question22-Question

?x <- (Lesson2 question22 y) =>
(retract ?x)
```

```
 (printout t crlf crlf
            " Which of the following is an output device?" crlf
            "  1) A mouse " crlf
            "  2) A digital plotter  "crlf
            "  3) Scanner" crlf
            "  4) Light pen " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer22 =(read))))

;=================================================================================
(defrule Lesson2-Question22-answer1
(or(Lesson2 answer22 1)(Lesson2 answer22 4)(Lesson2 answer22 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: A digital plotter " crlf
             crlf crlf
            " An output device is a device used to display information from a computer. Output devices
include" crlf
            " Conputer screen or monitor, printer, plotter, speakers, data projector, Speech Synthesizers, etc."
crlf
            crlf crlf
            )

(assert (Lesson2 question23 y)))

;=================================================================================
(defrule Lesson2-Question22-answer2
 ?x <- (Lesson2 answer22 2) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question24 y)))

;=================================================================================
 (defrule lesson2-Question23-Question

?x <- (Lesson2 question23 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following displays the results of processing carried out by a computer?" crlf
            "  1) Digital Scanner " crlf
            "  2) Light pen  "crlf
            "  3) Visual Display Unit " crlf
            "  4) Keyboard " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer23 =(read))))

;=================================================================================
 (defrule Lesson2-Question23-answer1
(or(Lesson2 answer23 1)(Lesson2 answer23 4)(Lesson2 answer23 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Visual Display Unit " crlf
             crlf crlf
```

```
            " An output device is a device used to display information from a computer. Output devices
include" crlf
            " Conputer screen or monitor (visual display), printer, plotter, speakers, data projector, Speech
Synthesizers, etc." crlf
            crlf crlf
            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Input Devices ..." crlf
            " Do you want to continue to the next topic and later read the lesson? .."crlf
            "==> y or n  " )
(assert (Lesson2 question24 =(read))))

;=========================================================================================
(defrule Lesson2-Question23-answer3
 ?x <- (Lesson2 answer23 3) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question24 y)))

;=========================================================================================
 (defrule lesson2-Question24-Question

?x <- (Lesson2 question24 y) =>
(retract ?x)
 (printout t crlf crlf
             " Which one of the following acts as both an output and an input device?" crlf
             "  1) Digital Scanner " crlf
             "  2) Light pen  "crlf
             "  3) Visual Display Unit " crlf
             "  4) Touch screen " crlf crlf
             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer24 =(read))))

;=========================================================================================
(defrule Lesson2-Question24-answer1
(or(Lesson2 answer24 1)(Lesson2 answer24 3)(Lesson2 answer24 2)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Touch screen " crlf
             crlf crlf
            " Some devices can be classified as both input and output devices. These include touch screens, "
crlf
            " modems, Removable Media and specialised equipment, such as electronic musical devices." crlf
            crlf crlf)

(assert (Lesson2 question25 y)))

;=========================================================================================
(defrule Lesson2-Question24-answer4
 ?x <- (Lesson2 answer24 4) =>
 (retract ?x)


            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
```

```
crlf )
(assert (Lesson2 question26 y)))

;==========================================================================
(defrule lesson2-Question25-Question

?x <- (Lesson2 question25 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which one of the following acts as both an output and an input device?" crlf
            "  1) Digital Scanner " crlf
            "  2) Modem  "crlf
            "  3) Speakers " crlf
            "  4) Lazer Printer " crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer25 =(read))))


;==========================================================================
 (defrule Lesson2-Question25-answer1
(or(Lesson2 answer25 1)(Lesson2 answer25 3)(Lesson2 answer25 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
           " This is a wrong answer... The right answer is: Modem " crlf
            crlf crlf
           "  crlf crlf
           " Some devices can be classified as both input and output devices. These include touch screens, "
crlf
           " modems,Removable Media and specialised equipment, such as electronic musical devices." crlf
" crlf
           " A modem is a piece of hardware enables a computer to connect to telephone lines" crlf
           " so that the user can send and receive data." crlf
            crlf crlf
           " I strongly suggest that you read the lesson thoroughly .." crlf
           " This is lesson 2 : Hardware / Input Devices ..." crlf
           " Do you want to continue to the next topic and later read the lesson? .."crlf
           "==> y or n  " )

(assert (Lesson2 question26 =(read))))

;==========================================================================
(defrule Lesson2-Question25-answer2
 ?x <- (Lesson2 answer25 2) =>
 (retract ?x)
           (printout t crlf crlf crlf
           "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question26 y)))

;==========================================================================
 (defrule lesson2-Question26-Question

?x <- (Lesson2 question26 y) =>
(retract ?x)
 (printout t crlf crlf
```

```
            "DVD stands for ?" crlf
            "  1) Data Versatile Disc " crlf
            "  2) Digital Video Disc  "crlf
            "  3) Digital Versatile Disc " crlf
            "  4) Desktop Video Disc " crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer26 =(read))))

;==========================================================================
(defrule Lesson2-Question26-answer1
(or(Lesson2 answer26 1)(Lesson2 answer26 2)(Lesson2 answer26 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Digital Versatile Disc " crlf
             crlf crlf
            " DVD (digital versatile disc) is a high-capacity development of the CD-ROM that can store
gigabytes"crlf
            " of information. While the CD was originally designed for recorded music, the DVD has had
many uses" crlf
            " from the beginning.It has replaced standard videocassette, for example, as it can store " crlf
            " an entire feature film with very high picture quality. Like CD-ROM, DVD discs cannot be " crlf
            " recorded on by the user, but DVD-R, a recordable version, is now becoming popular with home
users." crlf
            crlf crlf
            )

(assert (Lesson2 question27 y)))

;==========================================================================
(defrule Lesson2-Question26-answer3
 ?x <- (Lesson2 answer26 3) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question28 y)))

;==========================================================================

(defrule lesson2-Question27-Question
?x <- (Lesson2 question27 y) =>
(retract ?x)
 (printout t crlf crlf
            "Which of the following storage devices use optical (laser) storage technology?

" crlf
            "  1) Floppy Disk " crlf
            "  2) Hard Disk  "crlf
            "  3) Compact Disc " crlf
            "  4) Data Cartridge" crlf crlf

            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer27 =(read))))

;==========================================================================
 (defrule Lesson2-Question27-answer1
```

```
(or(Lesson2 answer27 1)(Lesson2 answer27 2)(Lesson2 answer27 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: Compact Disc  " crlf
             crlf crlf
            " Compact discs: While floppy and hard disks are magnetic, a compact disc (CD) is an optical
disc" crlf
            " that uses a laser to read the information. It can store very large amounts of data," crlf
            " typically 650 MB.s"crlf
            crlf crlf
            )

(assert (Lesson2 question28 y)))
;=========================================================================================
(defrule Lesson2-Question27-answer3
 ?x <- (Lesson2 answer27 3) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question28 y)))

;=========================================================================================
(defrule lesson2-Question28-Question

?x <- (Lesson2 question28 y) =>
(retract ?x)
 (printout t crlf crlf
             "Which of the following statements is not true" crlf
             "  1) Floppy Disks are slower than hard disks " crlf
             "  2) Hard disks are faster than Magnetic Tape Data Cartridges  "crlf
             "  3) DVD is slower than a Hard Disk" crlf
             "  4) RAM is slower than the Hard Disk" crlf crlf
             " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer28 =(read))))

;=========================================================================================
 (defrule Lesson2-Question28-answer1
(or(Lesson2 answer28 1)(Lesson2 answer28 2)(Lesson2 answer28 3)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is: RAM is slower than the Hard Disk  " crlf
             crlf crlf
            " Hard disk speed is defined by the disk access time. Access time refers to the speed " crlf
            " with which data stored on it is accessed. The lesser the time taken to access hard disk data," crlf
            " the faster the speed of the hard disk. This is measured in milliseconds. " crlf
            " Data is retrieved from a hard disk and temporarily stored in RAM for faster access." crlf
            " While accessing the data some free space is required on the hard disk to speed up the data
access." crlf
            " If there is not enough free disk space, the operating system will not be able to run the programs."
crlf
            " The capacity of the hard disk is very important to the performance of the CPU." crlf
            " It is measured in Gigabytes (GB). " crlf
            crlf crlf
            )
```

```
(assert (Lesson2 question29 y)))
;==========================================================================================
(defrule Lesson2-Question28-answer4
 ?x <- (Lesson2 answer28 4) =>
 (retract ?x)
          (printout t crlf crlf crlf
          "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question30 y)))

;==========================================================================================

 (defrule lesson2-Question29-Question
?x <- (Lesson2 question29 y) =>
(retract ?x)
 (printout t crlf crlf
          "Which of the following statements is true?" crlf

          "  1) Cache memory is slower than RAM " crlf

          "  2) CDs hold approximately the same as DVD's (give or take 10%)  "crlf

          "  3) Hard disks are both faster AND higher capacity than all DVD and CD type disks" crlf

          "  4) Floppy disk technology has a bright future with the modern computer" crlf crlf

          " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer29 =(read))))


;==========================================================================================
 (defrule Lesson2-Question29-answer1
(or(Lesson2 answer29 1)(Lesson2 answer29 2)(Lesson2 answer29 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
          " This is a wrong answer... The right answer is:" crlf
          " Hard disks are both faster AND higher capacity than all DVD and CD type disks " crlf
           crlf crlf
          "Speed, Cost and Capacity " crlf crlf crlf
          " Drive   Access Speed                          Sample Cost                  Capacity" crlf
          "
==========================================================================================
====" crlf
          " Hard disk        150 Mb per second                  350 units                    80–120 GB " crlf
          " Floppy disk      0.06 Mb per second                 1 unit                       1.44 MB " crlf
          " CD-R             3.3 Mb per second (at 32x)         2 units              650 MB " crlf
          " CD-RW                                               8 units              650 MB " crlf
          " DVD              4.8 Mb per second (at 32x)         15 units             4.5 GB " crlf
          " Zip disks        1.4 Mb per second                  17 units;30 units  100 MB; 250 MB" crlf
          " Data cartridges  7 Mb per second  68 units          20 GB " crlf crlf

          crlf crlf
 )


(assert (Lesson2 question30  y)))
```

```
;==========================================================================
(defrule Lesson2-Question29-answer3
 ?x <- (Lesson2 answer29 3) =>
 (retract ?x)
          (printout t crlf crlf crlf
          "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question32 y)))

;==========================================================================

 (defrule lesson2-Question30-Question
?x <- (Lesson2 question30 y) =>
(retract ?x)
 (printout t crlf crlf
           " Which of the following statements about the hard disk is false?" crlf

           "  1) Hard Disk is non-volatile memory " crlf

           "  2) Hard Disks can hold over a third of a terabyte of data  "crlf

           "  3) Hard Disk is the fastest memory in the computer system" crlf

           "  4) Hard Disk is the highest capacity storage medium in the computers in our classroom" crlf
crlf
           " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer30 =(read))))


;==========================================================================
 (defrule Lesson2-Question30-answer1
(or(Lesson2 answer30 1)(Lesson2 answer30 2)(Lesson2 answer30 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
          " This is a wrong answer... The right answer is:" crlf
          " Hard Disk is the fastest memory in the computer system " crlf
           crlf crlf
          " RAM is faster than the Hard Disk .. this is why " crlf crlf
          " Data is retrieved from a hard disk and temporarily stored in RAM for faster access." crlf
          " While accessing the data some free space is required on the hard disk to speed up the data
access." crlf
          crlf crlf

          " I strongly suggest that you read the lesson thoroughly .." crlf
          " This is lesson 2 : Hardware / Storage Devices ..." crlf
          " Please note that it is very important to learn about capacity and speed for each besides the
technology used.."crlf
          " Do you want to continue to the next topic and later read the lesson? .."crlf
          "==> y or n  " )
(assert (Lesson2 question31 =(read))))
;==========================================================================
(defrule Lesson2-Question30-answer3
 ?x <- (Lesson2 answer30 3) =>
 (retract ?x)
```

```
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf
crlf )
(assert (Lesson2 question31 y)))

;==========================================================================
 (defrule lesson2-Question31-Question

?x <- (Lesson2 question31 y) =>
(retract ?x)
 (printout t crlf crlf
            " Which of the following statements about formatting a floppy disk for the first time is true?" crlf
            "  1) Formatting prepares a floppy disk to store data " crlf
            "  2) Formatting backs up data onto a floppy disk  "crlf
            "  3) Formatting copies a floppy disk" crlf
            "  4) Formatting unlocks a floppy disk" crlf crlf
            " Please enter a number from 1 to 4 => ")
 (assert (Lesson2 answer31 =(read))))

;==========================================================================
(defrule Lesson2-Question31-answer2
(or(Lesson2 answer31 3)(Lesson2 answer31 2)(Lesson2 answer31 4)) =>
 ;(retract ?x)
 (printout t crlf crlf
            " This is a wrong answer... The right answer is:" crlf
            " Formatting prepares a floppy disk to store data " crlf
             crlf crlf
            " The surface of some disks is magnetic, and before formatting this surface is a " crlf
            " confusion of magnetic signals. When formatting is complete, this confusion has been ordered
into " crlf
            " tracks and sectors that accept data.Therefore, in order to prepare floppy disks for use they must "
crlf
            " be formatted to suit the type of computer they will be used in " crlf crlf

            crlf crlf

            " I strongly suggest that you read the lesson thoroughly .." crlf
            " This is lesson 2 : Hardware / Storage Devices ..." crlf crlf crlf

            "This is the end of Lesson 2 ...Good bye and with our best wishes ..  "crlf crlf

))
;==========================================================================
(defrule Lesson2-Question31-answer1
 ?x <- (Lesson2 answer31 1) =>
 (retract ?x)
            (printout t crlf crlf crlf
            "This is Correct ..Well done.. " crlf crlf crlf
            "This is the end of Lesson 2 ...  Goodbye and with our best wishes ..."crlf
            ))
;==========================================================================
```