SOFTWARE ENGINEERING COURSEWARE

LOW SIEW WEI

COLLEGE OF INFORMATION TECHNOLOGY
UNIVERSITI TENAGA NASIONAL

2004

SOFTWARE ENGINEERING COURSEWARE

by

LOW SIEW WEI

PROJECT SUPERVISOR: HAJAR MAT JANI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF INFORMATION TECHNOLOGY (HONS.)
COLLEGE OF INFORMATION TECHNOLOGY
UNIVERSITI TENAGA NASIONAL

2004

DECLARATION

I hereby declare that this thesis, submitted to Universiti Tenaga Nasional as a partial fulfillment of the requirements for the degree of Bachelor of Information Technology has not been submitted as an exercise for a degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the references.

This thesis may be made available within the university library and may be photocopied or loaned to other libraries for the purposes of consultation.

May 2004

Low Siew Wei
IT06810

DEDICATION

I dedicate this thesis to my project supervisor, Miss Hajar Mat Jani.

ACKNOWLEDGEMENT

I am very indebted to my project supervisor, Miss Hajar Mat Jani for all the advice and help extended to me during the accomplishment of this thesis. A million thanks for your time and enthusiasm.

Next, I would like to express my gratitude to my parents for being so supportive of me. Thanks for believing in me and giving me all those little things in life that I have always taken for granted.

To all my friends, especially Ho May Yee, Tee Shwo Ni and Liew Jing Ting, thanks for giving assistance and suggestions for this project.

Last but not least, I would like to thank people whose name I did not mention, who has helped me in one way or another. Without all of you, this project would not be a reality.

ABSTRACT

I have chosen "Software Engineering Courseware" as my final year project title. This is a courseware that teaches second year students of Universiti Tenaga Nasional Software Engineering (CMPB225)'s course. Microsoft Visual Basic is chosen as the main programming language and software to author the courseware. The proposed hardware is a 32MB personal computer with sound card. The main objective of this courseware is to make learning Software Engineering fun and educational because this subject is always portrayed by students as being dull and boring. Other than that, the courseware is intended to make memory retention better by enabling students to interact with it. Last but not least, the courseware is hoped to gauge students' knowledge & performance and improving them. The finished product is a complete courseware in Software Engineering based on Ian Sommerville's Software Engineering 6th Edition published by Addison-Wesley. It will include notes, examples, diagrams and also narration. In addition, quizzes and tests will be provided. As you can see, multimedia elements like text, audio, graphics and interactivity are included sufficiently for the benefit of users.

A structured and object – oriented approach is used, as this is an interactive courseware. The methodology adopted is prototyping model. A prototype is first built to accept feedback from users. After that, changes can be made to accommodate their requests. To implement of this product, researches will be done by means of referring to Software Engineering and Visual Basic books, the Internet, conducting surveys and interviews with students and instructors alike, and product testing by students.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

SE	Software Engineering
UNITEN	Universiti Tenaga Nasional

CHAPTER 1

INTRODUCTION

As mentioned in the abstract, the courseware will help students learn **independently** and **effectively**. We cannot deny that instructors are indeed needed because they have the 'human touch'. For example, when a student did not understand a certain concept in Software Engineering (or any other subject), he or she will ask the instructor about it. After the instructor has given his or her explanation, the student may still do not understand (perhaps the explanation is too abstract, perhaps the student does not get the instructor's point. Well, this is a two-way communication so there might be miscommunication from either or both parties). The student will then rephrase his or her question and voila! The instructor now gets the point and thus explains the concept in a much simpler way.

The main point that the author wishes to make is, the courseware is intended to help students in learning Software Engineering **more effectively and enjoyably instead of replacing the role of instructor altogether.** The other important point is, as we all know, students have different capabilities, study techniques, memory retention and attention spans. Hence, the courseware is created in mind with the above variables.

As an example, a 'normal' lesson will be delivered by the courseware but **the students are 'in charge' of the learning process. They will be able to read the material at their own speed, re-read the material with a click of a 'back' button, skip some sections (if the students found them to be easy) and read further about a concept by clicking a link.**

Once again, the author wishes to stress that this courseware is **tailored for students with different learning abilities. Slow learners can spend more time on reading the materials. In addition, there will be a tool tip to definition of a particular term. On the other hand, fast learners can skip this and also some sections but can still continue with the lessons.**

It is hoped that this system will assist students throughout their studies of Software Engineering. Contrary to popular belief, Software Engineering does not have to be a boring subject!

1.1 Problem Statement

The main objective of this project is to build a courseware that enables students to learn Software Engineering independently.

1.2 Background

The current way of teaching Software Engineering is by means of traditional face-to-face lecture. In addition, a website is provided by the current Software Engineering lecturer as a supplement to class lectures. Information about the website and its limitations can be found in page 35.

1.3 Objectives

The proposed courseware aims to:

a. Make learning of SE **fun**

   This is the main aim of the courseware as most students find that SE is a boring subject. With the inclusion of graphics, animation, audio, video and interaction, it is hoped that students will take a liking to the courseware and hence make learning fun.

b. **Assist** students in learning SE

   It is hoped that with the presence of the tool tip feature, students are able to get assistance on concepts they do not understand.

c. Enable students to **learn at their own pace**

As stated in the Introduction chapter (refer to page 1), each student can take their time to read each chapter because the students are 'in charge', not the courseware or the instructors!

d. **Motivate** students in learning SE

To motivate students, quizzes and test will be provided so that the students get instant feedback on their performance.

1.4 Scope

1.4.1 User Scope

Software Engineering Students of UNITEN

1.4.2 System Scope

Lessons, Tests, Quizzes, and Help features

1.5 Requirements and Constraints

1.5.1 Software Requirement

Windows 98 / ME / 2000 / XP

1.5.2 Hardware Requirements

Pentium 11 233 MHZ

32 Megabyte RAM

4X Speed CDROM

Sound Card

CHAPTER 2

RESEARCH

2.1 Research Methodology / Software Design Methodology

A research is conducted to choose the best **research methodology** and **software design methodology**. In doing research, the author surfed the Internet on Courseware Engineering and Instructional Design, read books on Software Engineering, Design for Computer-Based and Multimedia Learning, Visual Basic, Project Management, Systems Analysis and Design, and Database Systems.

2.1.1 Data Gathering Techniques – Interview and Survey

For this thesis, the author has decided to interview Software Engineering lecturers in UNITEN and do a survey by giving out questionnaires to 18 IT (Degree) students who have taken the course. The questionnaire is on page 23.

2.1.2 Software Design Methodology

Software Design Methodologies are also called Software Process Models in many Software Engineering books. As we know, there are different types of software process models, namely the waterfall model, evolutionary development, formal systems development, and reuse-oriented development [2].

The author will discuss each of these briefly in this chapter. Further discussions on the chosen Software Design Methodology is presented in the Analysis Chapter on page 20.

The **Waterfall** model, also known as the **Classic Life Cycle** or the **Linear Sequential Model** (Pressman, 2001), is a systematic, sequential approach to Software Development. Each process of analysis, design, coding and support has to be completed (or *signed off* [approved] by authorities before progressing on the next process [2].

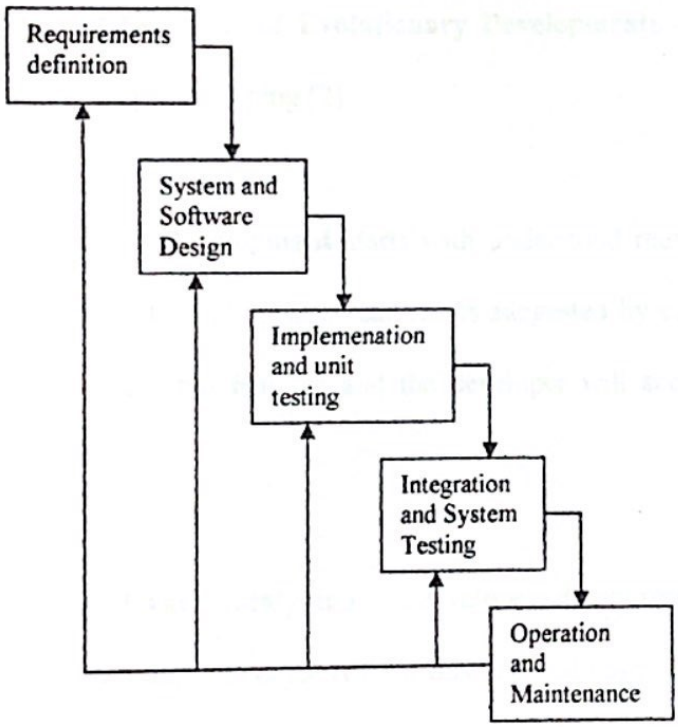


Figure 1 The Software Life Cycle for Waterfall Model [2]

"Although the linear model is often derided as "old-fashioned," it remains a reasonable approach when requirements are well understood." [3].

However, there are some problems in this model. Firstly, iteration is done indirectly and this can cause confusion to the project team. Secondly, it is often difficult for the customer to state all requirements explicitly. Last but not least, the customer can only see the outcome of the project at the end of its completion. In other words, the customer cannot involve directly in the production process of the project [3].

There are two types of **Evolutionary Developments** – Exploratory Development and Throw Away Prototyping [2].

Exploratory Development starts with understood requirements. The requirements are then refined by adding new features as suggested by customer. The customer will then suggest additional features and the developer will accommodate these into the initial system.

In **Throw Away Prototyping**, the developers do not really understood what the customer wants. In fact, the objective of this methodology is to understand the customer's requirements and incrementally improve on the system, hence the name Throw Away Prototyping.

"Prototype development can be a very effective strategy, as it can reduce risk, increase user involvement, and save development time and cost in some cases." [4].

Evolutionary Approach is only suitable for small systems (less than 100,000 lines of code) and medium-sized systems (up to 500, 000 lines of code) with a fairly short lifetime [2].

Formal Systems Development involves on mathematical transformation of a system specification to an executable program [2]. It allows "...a software engineer to specify, develop and verify a computer-based system by applying a rigorous, mathematical notation[3]."

The advantage of this approach is, problems can be discovered and corrected more easily through the application of mathematical analysis. Furthermore, the distance between each transformation is less than the distance between a specification and a program, which makes the program easy to deal with [2].

This model is quite time consuming and expensive [3]. Expertise is needed in order to use this model and it can be difficult for them to communicate with technically unsophisticated customers. "...this process does not offer significant cost or quality advantages over other approaches [2]."

When creating a project, there might be some functions that are similar to that of a previous project. In this case, **Reuse-Oriented Development** is needed. The similar codes can be taken directly or modified to suit the current program. One of the advantages

are, it saves time and effort because we do not have to *reinvent the wheel*. It also reduces costs and risks, and "...leads to faster delivery of the software [2].

There are statistics based on studies of reusability by QSM Associates, Inc. **Component-Based Development** reduces 70 percent in development cycle time; reduces 84 percent in project cost, and a productivity index of 26.2, compared to an industry norm of 16.9 [3].

The system that is developed using this method might not meet the real needs of users, and lose of some control over the system evolution [2].

CHAPTER 3

LITERATURE REVIEW

3.1 Courseware

HyperDictionary [http://www.hyperdictionary.com] defines courseware as "programs and data used in Computer-Based Training". To understand this, we shall look at the definition of Computer-Based Training. The same dictionary defines it as "(CBT) Training (of humans) done by interaction with a computer. The programs and data used in CBT are known as "courseware"".

Here is another definition of courseware: "Courseware, a term that combines the words *course* with *software*, is educational material intended as kits for teachers or trainers or as tutorials for students, usually packaged for use with a computer" [21] According to the same website, courseware includes the following:

- Material for instructor-led classes
- Material for self-directed computer-based training (CBT)
- Web sites that offer interactive tutorials
- Material that is coordinated with distance learning, such as live classes conducted over the Internet
- Videos for use individually or as part of classes

This project revolves around the second point, that *is material for self-directed computer-based learning.*

3.2 Computer-Based Training (CBT)

There are many other similar terms for CBT: **Computer-based instruction (CBI), computer-assisted instruction or computer-aided instruction (CAI)** [9]. "There is no clear difference between these materials, the names being largely the personal preference of the designer[9]."

Today, CBT provides **hypertext** where users can click on one point of information and be directed to another, and there is even **hypermedia** where users can move from one information to another in many types of media [1].

3.3 Courseware Engineering

To develop a business software, we use Software Engineering methodology. What do we use if we want to develop a courseware for learning purposes? Do we use Software Engineering as the only methodology? We need both **Software Engineering** and **Instructional Design** [11]. Mix these two and we have **Courseware Engineering**.

For a start, we need to understand the meaning of instructional design. As its name implies, instructional design means "...the process of developing instructions (for delivery by computers or other means) [11]."

Software Engineering, on the other hand, "...is the systematic approach to the development, operation, maintenance and retirement of software (IEEE Glossary, in van Vliet 1993) [11].

"Courseware engineering is software engineering applied to courseware but the requirement for learning means most activities are different. The **analysis & design** phases are from ID while production & testing come from software engineering[11]."

3.4 Instructional Design and Learning Theory

There are many types of learning theories and associated instructional design strategies. Brenda Mergel, a graduate student of the University of Saskatchewan explains in her paper entitled *Instructional Design & Learning Theory* the meanings of a theory [12]:

- *A theory provides a general explanation for observations made over time.*
- *A theory explains and predicts behavior.*
- *A theory can never be established beyond all doubt.*
- *A theory may be modified.*
- *Theories seldom have to be thrown out completely if thoroughly tested but*
- *sometimes a theory may be widely accepted for a long time and later*
- *disproved.*

She also wrote that "A model is a mental picture that helps us understand something we cannot see or experience directly[12]."

According to her, there are three basic theories of learning - Behaviorism, Cognitivism and Constructivism [12].

Behaviorism is based on changes in behavior. It focuses on a new behavioral pattern being repeated until it becomes automatic. **Cognitivism** looks at changes in a behavior to find out what is happening inside the learner's mind. **Constructivism** believes that each of us have our own perspective of the world based on individual experiences and schema. It prepares the learner to solve problems in ambiguous situations.

The following explains further on each of the learning theories.

Behaviorism studies the overt behaviour (response to stimulus) that can be observed and measured in terms of quantity. As described by Brenda Mergel, this theory is not new [12]. In fact, Aristotle (384-322 BCE.), a famous Greek Philosopher had associated lightning and thunder in his essay "Memory". Then there is Pavlov (1849 - 1936) who worked on classical conditioning and stimulus substitution. In his experiment, he placed a dog together with food (Unconditioned Stimulus) and a bell (conditioned stimulus). When he rang the bell, the dog showed no response. He then placed food in front of the dog and causes the latter to salivate (Unconditioned Response [natural, not learned]). He then began to "condition", or, in other word, teach the dog. The bell was rung a few seconds before food was given. Thereafter, the ringing of the bell alone produced salivation (Conditioned Response [to bell]).

There are other researchers in Behaviourism theory like Watson, Thorndike and Skinner but their experiments are not included in this thesis.

Cognitivism theory emerges because of limitations in the Behaviourism theory. by Brenda Mergel gave an example [12]:

...children do not imitate all behavior that has been reinforced. Furthermore, they may model new behavior days or weeks after their first initial observation without having been reinforced for the behavior. Because of these observations, Bandura and Walters

departed from the traditional operant conditioning explanation that the child must perform and receive reinforcement before being able to learn.

Most of the learning happens through associations of contiguity and repetition. Other than that, feedback is important as a motivator. Some of the key concepts of Cognitivism are Schema, Three-Stage Information Processing Model, Meaningful Effects, Serial Position Effects and Practice Effects [12].

Schema is a structure where knowledge is stored. New knowledge is compared to existing ones and the schema may be combined, extended or altered. The **Three-Stage Information Processing Model** explains how information is processed and stored in Sensory Register, Short-Term and Long-Term memory. **Meaningful Effects** stressed that meaningful information is easier to learn and remember. **Serial Position Effects** state that we find it easy to remember from the beginning or end of a list rather than those in the middle, unless the item is distinctly different. As its name suggests, Practice Effects conforms to the adage *practice makes perfect.*

Constructivism states that every individual's knowledge is based on the person's experiences, mental structures and beliefs. As we learn, we construct knowledge for ourselves. It is interesting to note that there are two dramatic consequences of this view, as stressed by Arts in Education Institute of Western New York [14]:

1. We have to focus on the learner in thinking about learning (not on the subject/lesson to be taught):

2. There is no knowledge independent of the meaning attributed to experience (constructed) by the learner, or community of learners.

Arts in Education Institute of Western New York explains the second point as:
"We have to recognize that there is no such thing as knowledge "out there" independent of the knower, but only knowledge we construct for ourselves as we learn."

3.4.1 Instructional System Design (ISD): Using the ADDIE Model

Instructional Design is the systematic approach to the **Analysis, Design, Development, Implementation, and Evaluation of learning materials and activities [13].**

Unlike traditional learning in classrooms, ADDIE's approach to teaching is learner-centered in order for students to learn effectively. This is done by analyzing the learner's needs.

In the **Analysis** phase, we determine *what* we want students to learn. After that, we list out the instructional goals and a list of tasks to be instructed. These will be inputs to the Design phase.

The **Design** phase will plan *how* to develop the instruction using the outputs from the Analysis phase, that is, how to reach the instructional goals. "Some of the elements of the Design Phase may include writing a target population description, conducting a learning

analysis, writing objectives and test items, selecting a delivery system, and sequencing the instruction [13]."

Using the outputs from the Design phase, we generate the lesson plans and lesson materials. The **Develop** phase is where the instruction, media and supporting documentation, which comprises of hardware and software, are created.

After going through the Develop phase, it is time for **Implementation**. This phase focuses on the actual delivery of the instruction effectively and efficiently. The instruction can be delivered in a classroom, lab or computer. It is not just simply delivering what has prepared but to also make sure that the students understood what is being taught.

In the **Evaluation** phase, the effectiveness and efficiency of the instruction is measured. "**Evaluation** should actually occur throughout the entire instructional design process - within phases, between phases, and after implementation. Evaluation may be Formative or Summative [13]."

Formative and Summative Evaluation [13]:

Formative Evaluation is ongoing during and between phases. The purpose of this type of evaluation is to improve the instruction before the final version is implemented. **Summative Evaluation** usually occurs after the final version of instruction is implemented. This type of evaluation assesses the overall effectiveness of the instruction. Data from the Summative Evaluation is often used to make a decision about the instruction (such as whether to purchase an instructional package or continue/discontinue instruction).

It is important to note that "These phases sometimes overlap and can be interrelated; however, they provide a dynamic, *flexible* guideline for developing effective and efficient instruction [13]."

CHAPTER 4

REQUIREMENTS SPECIFICATION AND ANALYSIS

4.1 Chosen Software Design Methodology

Based on the research done on the pros and cons of each of the software design methodologies (refer to page 7), the author has chosen the **prototyping** methodology to develop the courseware. **Prototype** is *first of a type* [4]. Prototyping is actually a kind of evolutionary development. An initial system is built and exposed to user comments. The prototype is then refined many times until a satisfied system is developed. All these stages run at the same time.

Prototyping (evolutionary development) should be used when developing educational software [11]. The reason is: learning depends on people and situation so it is hard to get our analysis and design right the first time. Therefore prototypes are needed, to allow us to get user comments in order to improve on the courseware. However, prototypes alone do not give a clear picture of whether users are able to learn from it. Hence, pilot and field testing is required.

4.2 Chosen Instructional System Design Model

The author has chosen the Analysis, Design, Development, Implementation, and Evaluation (ADDIE) as the Instructional Design Model for this courseware (refer to page 17 for more information on this model). Readers might wonder why this model is chosen among others. "There are more than 100 different ISD models, but almost all are based on the generic "ADDIE" model, which stands for Analysis, Design, Development, Implementation, and Evaluation [15]." Therefore, this model is chosen based on its commonality.

But then again, readers may have another question in mind: Since the writer has selected the prototyping approach to develop the courseware, why use the ADDIE model as the latter was very structured in approach? To answer this, the author wishes to quote from the e-learningGuru website once more:

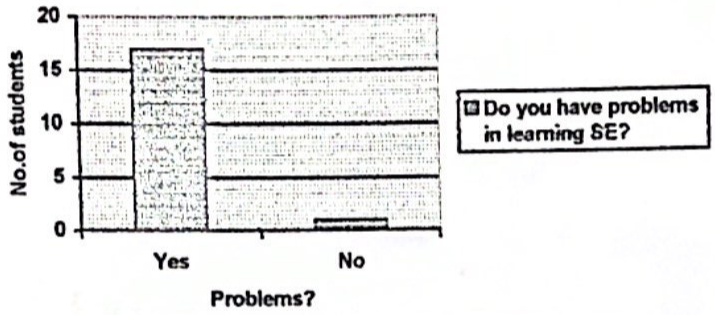
*For best results, the development process for CD-ROM or Web-based training programs should use a **modified ADDIE model**, which borrows from the most valuable aspects of the systemic approach. Specifically, a rapid prototype phase is inserted after, or as an extension of, the design phase. A rapid prototype is simply a quickly assembled module that can be tested with the student audience early in the ISD process.*

As described in the previous paragraph, a modified ADDIE model is used instead of the traditional one. By incorporating the rapid prototyping process into the traditional ADDIE model, the courseware can have the best of both worlds!

4.3 Survey (Questionnaire) Feedback

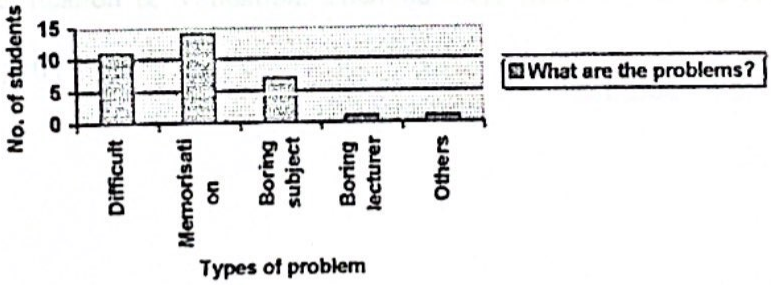
A survey has been conducted to 18 students to find out their opinions of learning Software Engineering (SE). The following is the outcome of the survey. The percentages are rounded up to two decimal places.

1. Do you have problems in learning Software Engineering (SE)?



According to the survey, 17 (94.44 %) of the respondents answered Yes while only one (5.55 %) answered No. This shows that almost everyone of the 18 students surveyed have problems in learning SE.

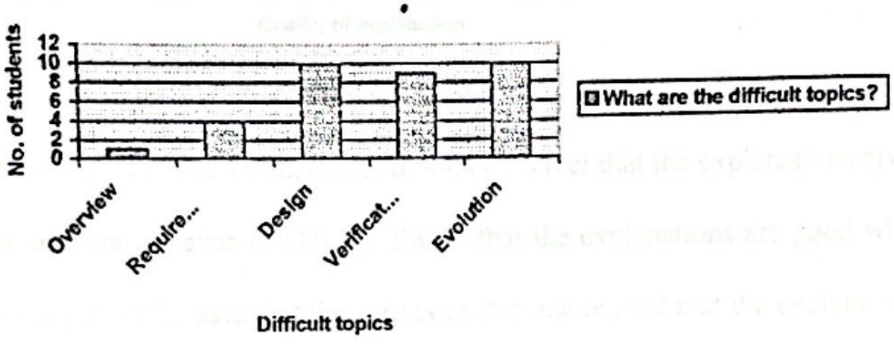
If yes, what are the problems?



The graph above shows that the most common problem is the large amount of memorisation involved in learning SE in which 14 students (77.78 %) have agreed, followed by the difficulty of the subject (11 students or 61.11 %) and the nature of the

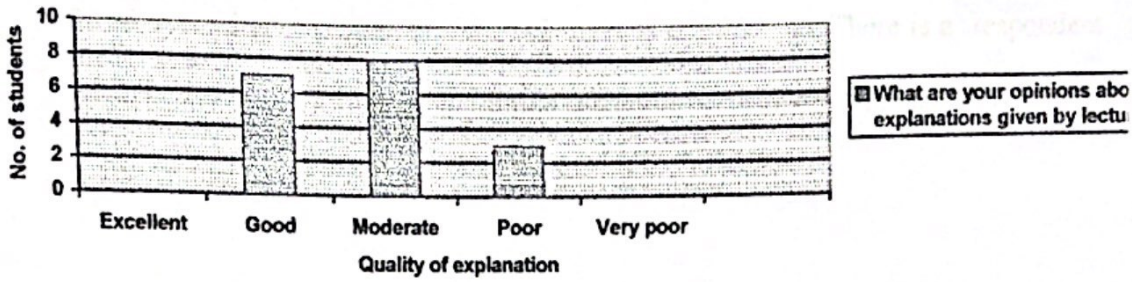
subject itself – boring subject (7 students or 38.89 %). There is another problem not included in the options (of the survey), which is lack of practical work, as suggested by one student.

2. What are the difficult topics?



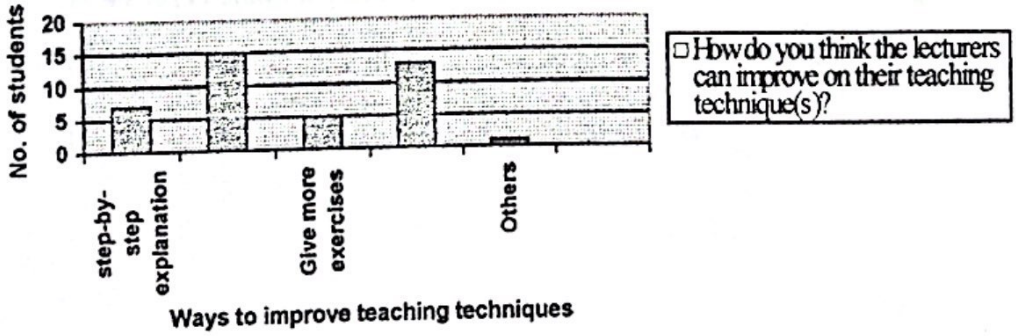
Most students have difficulty in the later topics in SE – Design, Evolution and Verification & Validation. Their numbers (from highest to lowest) are 10 (55.55 %),10 (55.55 %) and 9 (50 %) respectively.

3. What are your opinions about explanations given by lecturers?



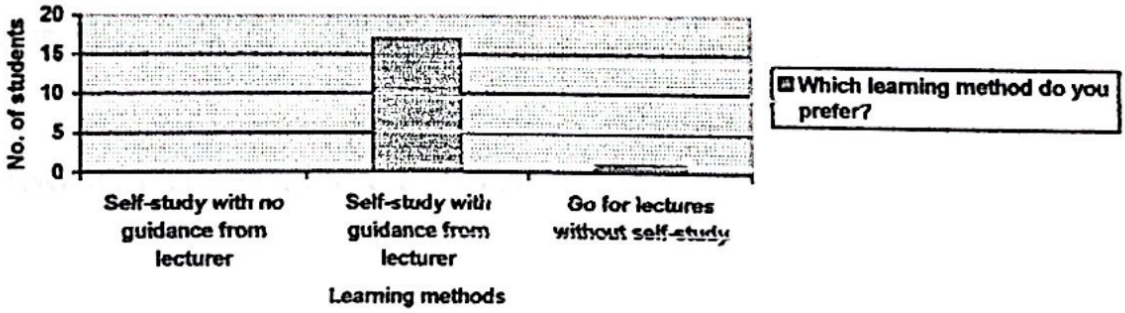
The majority (8 students out of 18 surveyed, or 44.44 %) feel that the explanations given by lecturers are moderate. Seven (38.89 %) thinks that the explanations are good while only three persons (16.67 %) said that they are poor. No one replied that the explanations given are excellent or very poor. This means that the quality of the explanations given by lecturers ranges from poor to good.

4. How do you think the lecturers can improve on their teaching technique?



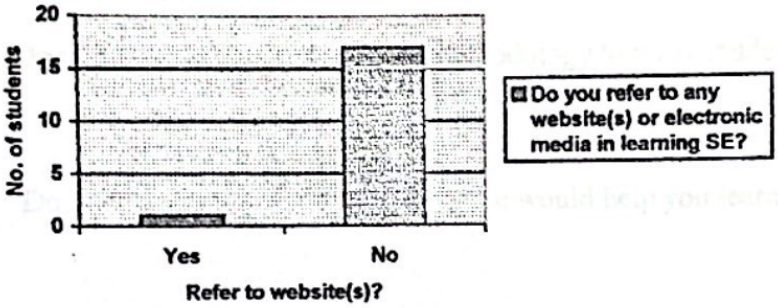
Fifteen of the eighteen respondents (83.33 %) belief that the lecturers should give more examples while teaching SE. Other suggestions include making the lesson interesting (13 students or 72.22 %), giving step-by-step instrucitons (7 students or 38.89 %) and giving more exercises (5 students or 27.78 %). There is a respondent who suggested that a project be given to students using proper methodology (this idea is categorised under the "others" technique).

5. Which learning method do you prefer?



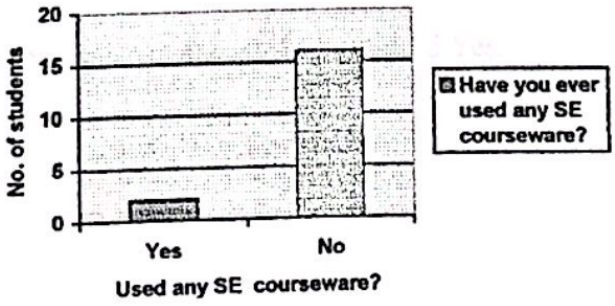
The percentage of students who prefer to self-study with guidance from lecturer is 94.44 % (17 students). Only 5.55 % (1 student) who would like to only go for lectures without self-study. No students have selected "self-study with no guidance from lecturer" in the survey.

6. Do you refer to any website(s) or electronic media in learning SE?



Only one student refers to the website or electronic media in learning SE. The rest answered No.

7. Have you ever used any SE courseware?

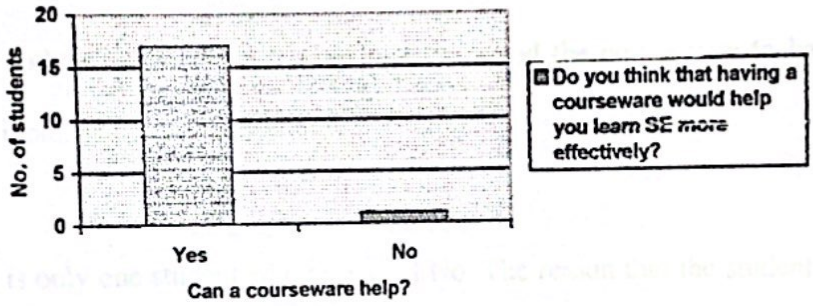


Two students (11.11 %) answered Yes for this question while 16 students (88.89 %) answered No.

If yes, what are the advantages / disadvantages of the courseware?

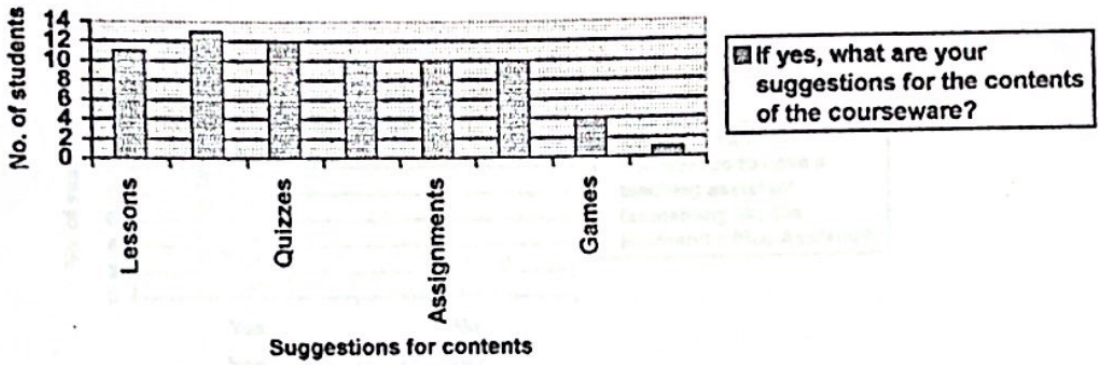
The advantages that the students gave are ease of use and understanding while the disadvantages are the unsuitability of use in today's world and that the courseware do not provide proper format or methodology to make students adapt to it.

8. Do you think that having a courseware would help you learn SE more effectively?



Seventeen students (94.44) answered Yes.

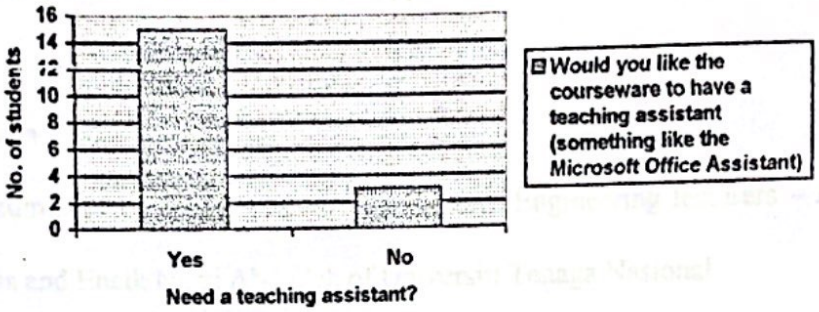
If yes, what are your suggestions for the contents of the courseware?



In the "others" category, a student has suggested the courseware to have a format which includes different chapters.

There is only one student who answered No. The reason that the student gave was: a lecturer's guidance is needed for learning SE and face-to-face explanation will be better than a courseware.

9. Would you like the software to have a teaching assistant (something like the Microsoft Office Assistant) ?



Fifteen students (83.33 %) surveyed said Yes while three students (16.67 %) said No.

10. Do you have any other comments / concerns regarding the learning of SE / SE courseware?

Please elaborate if you have.

Below are the students' responses:

SE is a thereotical subject which contains many concepts and thus demands much understanding in it. Therefore, the courseware should really show in a clearer way so that students can understand. I would prefer to have a simple and accurate explanation in SE theory. The SE courseware should be attractive and interactive.

4.4 Interview Result

An interview was also conducted to three lecturers. They are the head of Software Engineering, an ex-Software Engineering lecturer and the current Software Engineering lecturer. The interview can be viewed in Appendix.

4.5 Interview Summary

This is a summary of the interview with Software Engineering lecturers – Assoc. Prof. Jamilin Jais and Encik Nazri Abdullah of Universiti Tenaga Nasional.

Both lecturers find teaching SE an easy task but they do not get all the students' attention. However, the majority did all the assignments given. Prof. Jamilin and Encik Nazri's textbooks do not contain SE courseware. They agreed on using a courseware to teach SE, and their suggestions include inserting multimedia elements like pictures, diagram, examples on project development, sample on documentation, actual works on software development, animations, video and PowerPoint slides.

This is a summary of the interview with En. Yunus Yusoff, the head of the Software Engineering unit.

The author gets some pointers on how to prepare interview questions, the possibility of evaluating the Management Information Systems (MIS) courseware and the importance of a proper definition of Software Engineering (SE).

4.6 Decisions that I have made based on my research (survey and interview)

All but one students surveyed have encountered one or more problems in learning Software Engineering. The three main problems are the **difficulty, memorisation and dullness** of the subject. With these in mind, the author aspires to solve them by making the subject less difficult, more interesting and helping students memorise easier. The explanations will be simple and clear following comments by students in the survey.

To make **learning SE effortless**, clear explanations complete with examples will be given. Multimedia elements like sound, animation and interaction will be provided to **make the subject more interesting**. Wade (1995) wrote that "...learning theorists say that people master knowledge, concepts, and skills through certain senses more than others." According to her, the percentage of learning that occurs to sight is 80%, hearing is 10%, touch is 5% and smell/taste is 5%. She also wrote that "Information presented through learning tools that simultaneously use both words and pictures is seven times more likely to be retained than words alone." After a student finishes studying, he/she can do some quizzes, tests and assignments to test their knowledge and understanding.

Given that the majority (8 students out of 18 surveyed, or 44.44 %) feel the explanations given by lecturers are moderate, seven (38.89 %) think that the explanations are good while only three persons (16.67 %) said that they are poor, the courseware will have more examples, interesting lessons and step-by-step instructions.

Concerning the methods of learning, almost all students prefer to study with guidance from lecturer. So the courseware will serve as a tool for independent learning but if the students do not understand the subject, he/she can still ask the lecturer for help. According to Assoc. Prof. Jamilin Jais in the interview, computerized supplementary learning tools are good because it helps lecturers a lot. Students can study on their own just like their counterparts in Universiti Tun Abdul Razak, where they only meet their lecturers once a month.

Only a student out of the 18 surveyed has referred to website or electronic media and two students have used (an) SE courseware(s). The author does not think this practice as good because students should explore as much as they can and not just relying on the course notes or text book. However, 17 of students agreed that a courseware would help them learn SE more effectively. This further explains the urgent need of an SE courseware.

Since most of the contents (lessons, multimedia elements, quizzes, tests, assignments, etc.) proposed in the survey are selected by students, the author decides to include them. However, games will not be included as they are not in demand. The author will include ideas by Assoc. Prof. Jamilin Jais (previous Software Engineering lecturer) to add in animations and also those suggested by Encik. Nazri Abdullah (current Software Engineering lecturer) to include pictures and diagrams.

CHAPTER 5

DESIGN

5.1 Review of current system

The current way of teaching Software Engineering is by traditional class lectures. In addition, En. Nazri Abdullah, the present lecturer has provided a website for students. The explanation of the website follows next.

5.1.1 Explanation of Current System

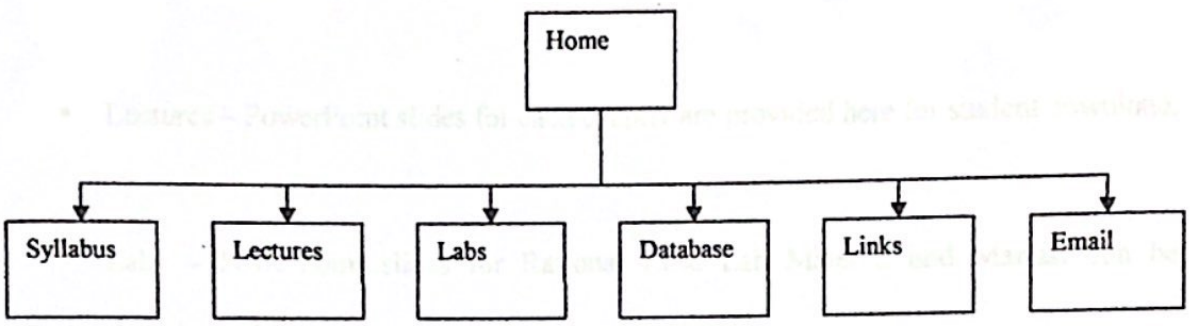


Figure 2 Structure of the Website

Explanation of the Website's Contents

Currently there is no courseware to teach Software Engineering (CMPB 225) students. However, En. Nazri has a website for SE. Figure 2 is a structure of En. Nazri's website. The URL for his website is http://metalab.uniten.edu.my/~nazri/CMPB225.

In the main page (home), there are sections entitled **Course Overview, Course Meeting** (Lecture and Lab hours), **Course Objectives, Course Material, Course Grading** and **Course .my Approach** (En. Nazri's Approach of Teaching SE).

There are also links to **Syllabus, Lectures, Labs, Database, Links** and **Email** (En. Nazri's email).

The contents of the links are as follows:

- Syllabus – This page contains the syllabus of SE in terms of chapters to be taught with respect to time. The dates of quizzes and exams are also given.

- Lectures – PowerPoint slides for each chapter are provided here for student download.

- Labs – PowerPoint slides for Rational Rose Lab Material and Manual can be downloaded here.

- Database – There are students' quiz marks, quizzes, project titles, and previous mid-term and final exams on this page.

- Links – This page contains links to other SE websites.

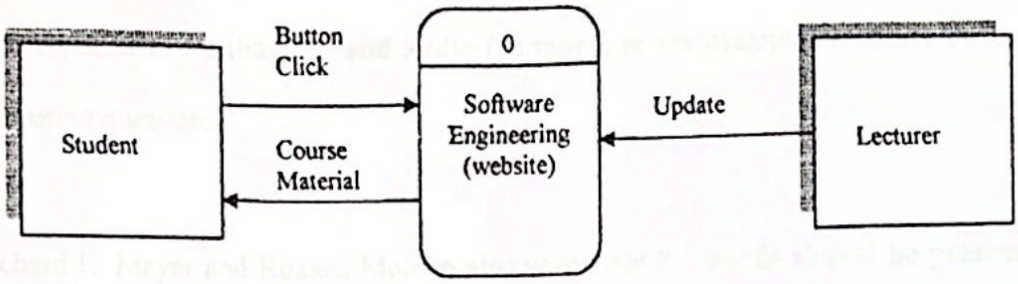


Figure 3 Context Diagram of En. Nazri's Software Engineering Website.

With regards to the context diagram of En. Nazri's Software Engineering Website in Figure 2, when a student clicks on the links (explained in page 36), the latter will display course materials to the student.

Explanation of Current System (PowerPoint Slides from Sommerville textbook)

As we know, multimedia consists of text,graphics, animation, video, audio and interaction. The PowerPoint slides used currently have only **text, graphics and linear interaction.** This is not enough for a full multimedia learning and hence it could not bring out the best in students. It lacks animation, video and audio. Richard E. Mayer and Roxana Moreno wrote that

...students who listened to a narration explaining how a bicycle tire pump works while also viewing a corresponding animation generated twice as many useful solutions to subsequent problem-solving transfer questions than did students who listened to the same narration without viewing any animation [10].

This explains that **animation and audio** (narration, in the example case) fare better in answering questions.

Richard E. Mayer and Roxana Moreno also wrote that " ...**words should be presented auditorily rather than visually**." According to them,

Sweller and his colleagues call this a split attention effect (Chandler & Sweller, 1991; Mousavi, Low & Sweller, 1995; Sweller, Chandler, Tierney and Cooper, 1990). This result is consistent with the cognitive theory of multimedia learning because the on-screen text and animation can overload the visual information processing system whereas narration is processed in the verbal information processing system and animation is processed in the visual information processing system[10].

Although students can interact with the PowerPoint slides by means of clicking the mouse or pressing the right arrow on the keyboard, these interactions are linear. **Students are unable to skip a section they have already understood. They are "forced" to read slide by slide until the end, unless of course, if they decide to "jump" to other slides themselves.**

Explanation of Current System (Quizzes and Tests)

Currently, the lecturer tests students' understanding of SE by giving quizzes and tests. These are given on hard copy (paper) and students are required to do them manually. As there is no courseware for Software Engineering, students are unable to practise on these

quizzes and tests before they sit for the actual ones. A courseware will benefit students by having quizzes and tests where students can test their own understanding and practise on their test or quiz answering skills.

5.2 Proposed System

5.2.1 Data Flow Diagram (DFD)

5.2.1.1 Context Diagram

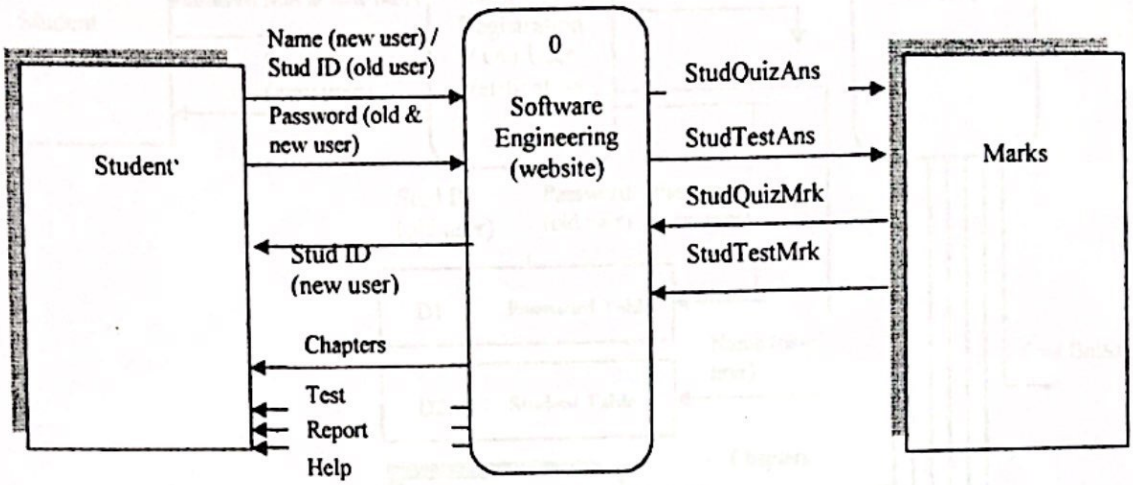
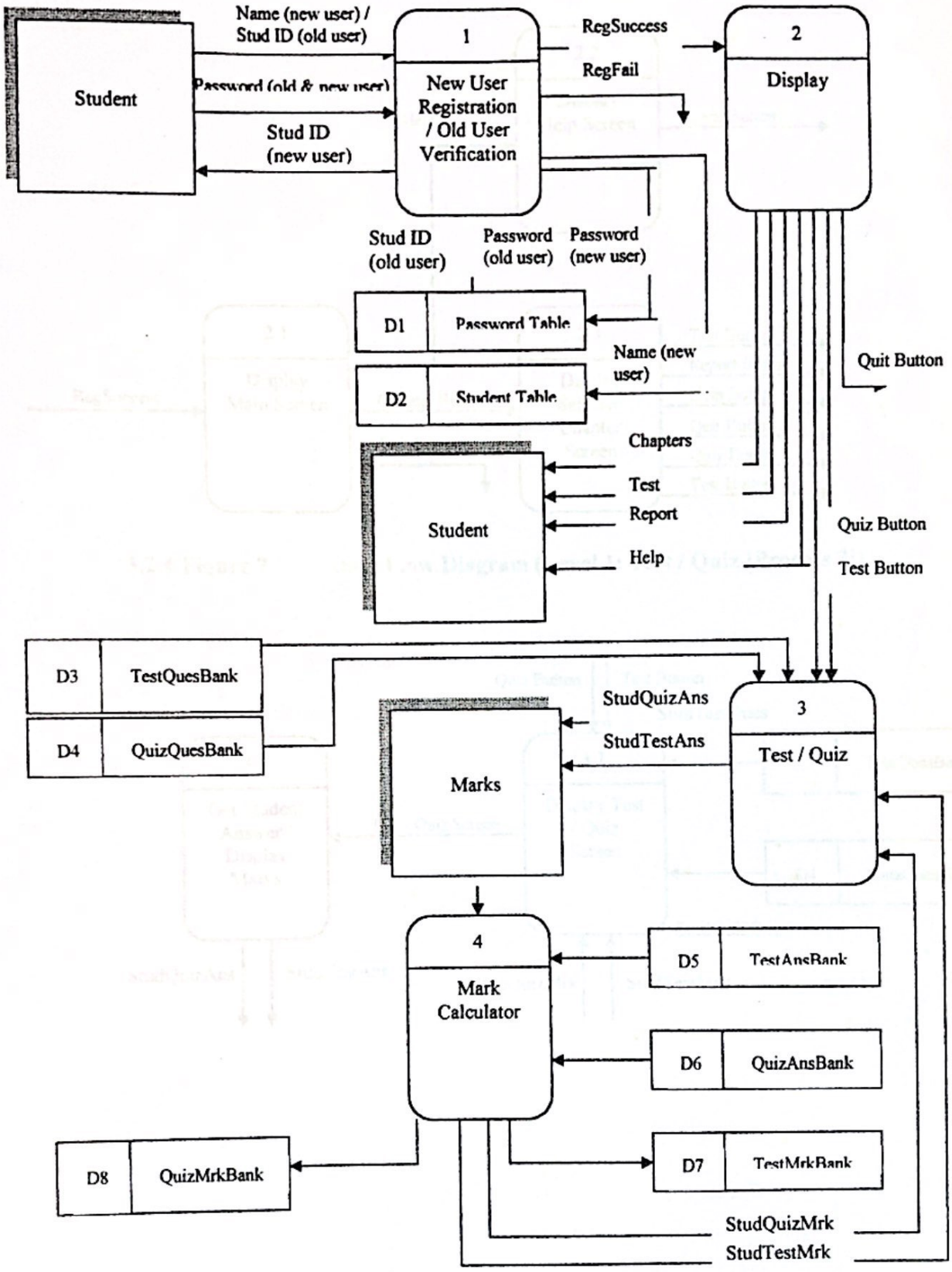
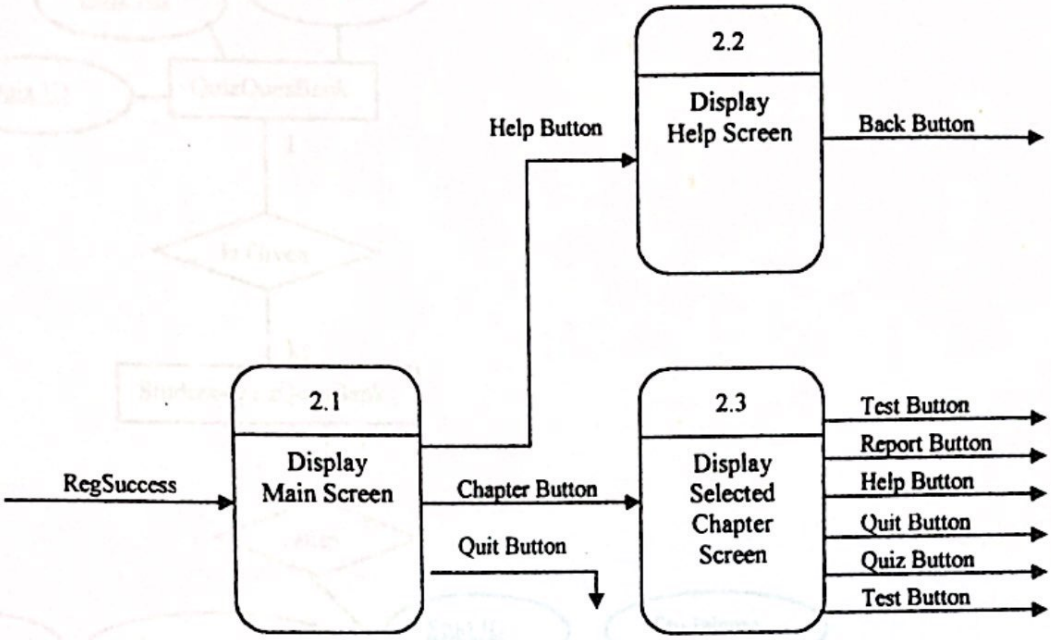


Figure 4 Context Diagram of the proposed system

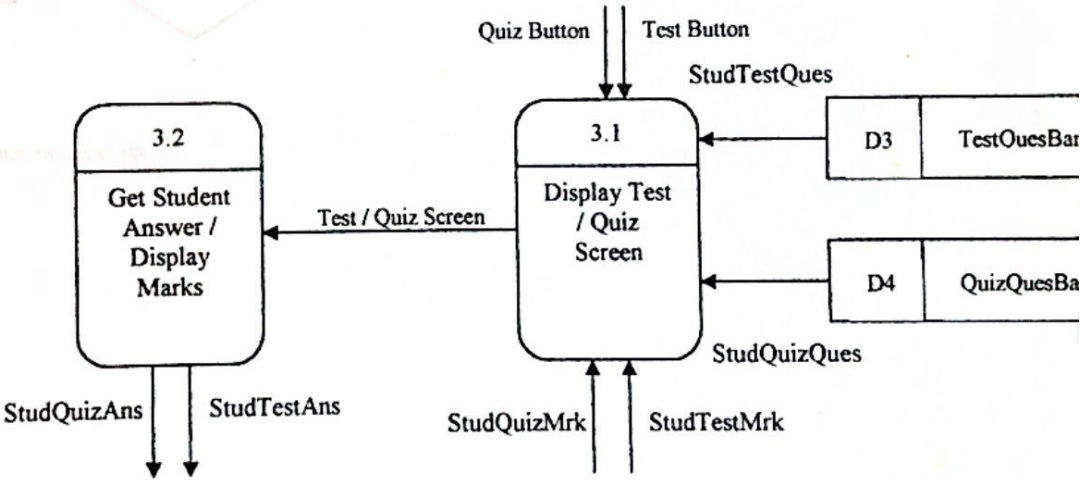
5.2.1.2 Figure 5 Data Flow Diagram (Level 0)



5.2.3 Figure 6 Data Flow Diagram (Level 1: Display [Process 2])

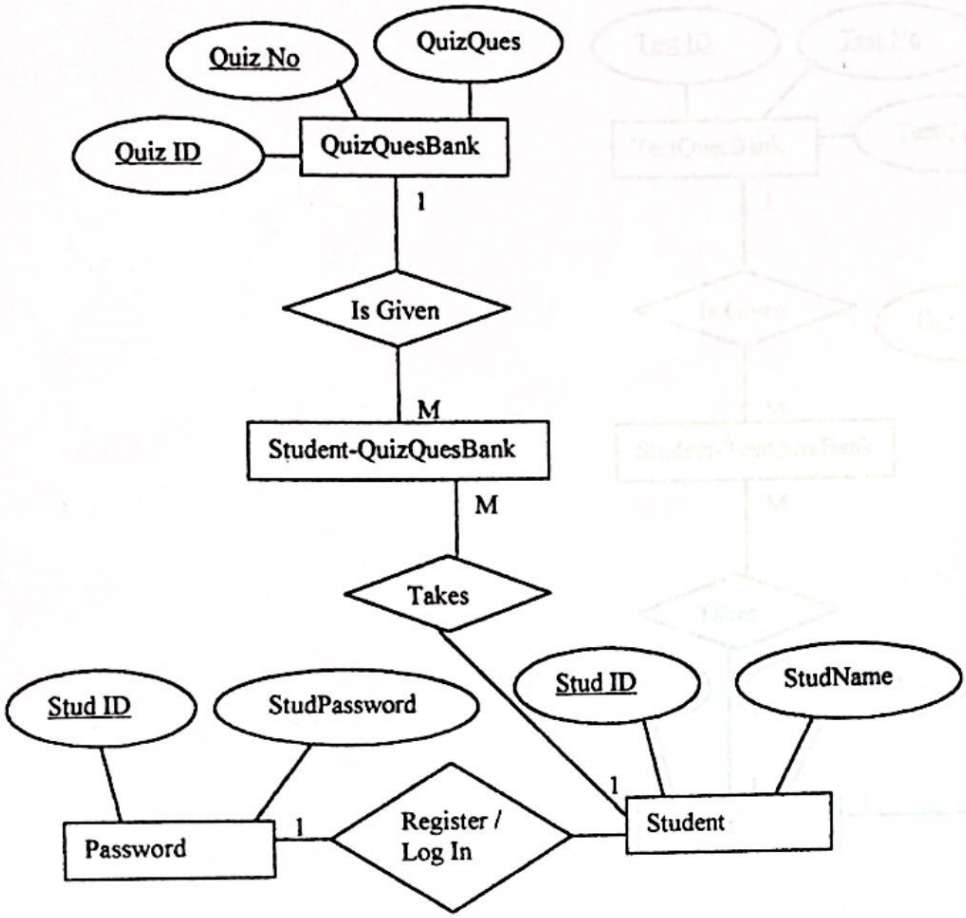


5.2.4 Figure 7 Data Flow Diagram (Level 1: Test / Quiz [Process 3])



5.2.5 Database Design

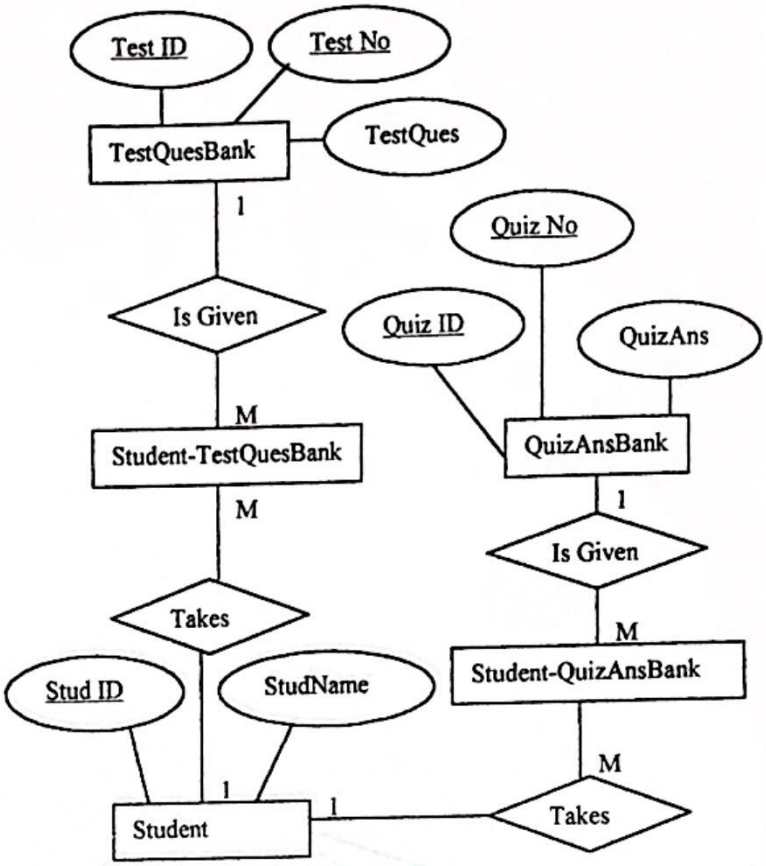
5.2.5.1 Entity Relationship Diagram (ERD)



Continue on next page...

5.2.5 Database Design

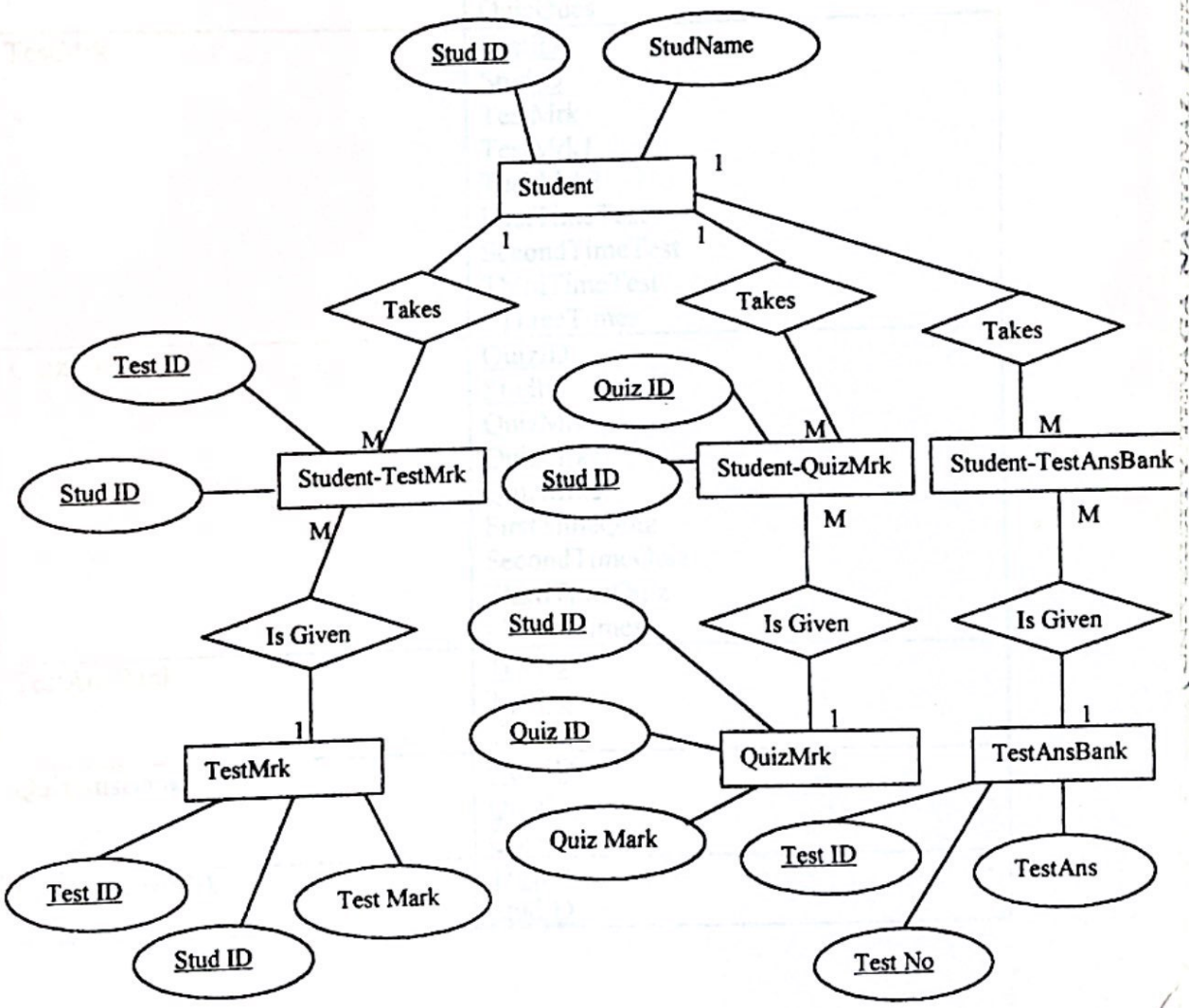
5.2.5.1 Entity Relationship Diagram (ERD)



Continue on next page...

5.2.5 Database Design

5.2.5.1 Entity Relationship Diagram (ERD)



5.2.5 Database Design
5.2.5.2 Normalization

The purpose of Normalization is to remove redundant data and to eliminate update anomalies like insertion, deletion and modification.

Note: The underlined fields are *primary keys*.

Table Name	Normalized Table Fields
Student Table	StudID StudName
Password Table	StudID Password
TestQuesBank	TestID TestNo TestQues
QuizQuesBank	QuizID QuizNo QuizQues
TestMrk	TestID StudID TestMrk TestMrk1 TeskMrk2 FirstTimeTest SecondTimeTest ThirdTimeTest >ThreeTimes
QuizMrk	QuizID StudID QuizMrk QuizMrk1 QuizMrk2 FirstTimeQuiz SecondTimeQuiz ThirdTimeQuiz >ThreeTimes
TestAnsBank	TestID TestNo TestAns
QuizAnsBank	QuizID QuizNo QuizAns
Student-TestMrk	TestID StudID

Student-QuizMrk	QuizID StudID
Student-TestQuesBank	StudID TestID TestNo
Student-QuizQuesBank	StudID QuizID QuizNo
Student-TestAnsBank	StudID TestID TestNo
Student-QuizAnsBank	StudID QuizID QuizNo
QuizOptions	QuizID QuizNo QuizOption QuizOptionAns

5.2.5 Database Design

5.2.5.3 Table Description

Student

Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
StudName	Student name	Not Null	Char

Password

Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
StudPassword	Student Password	Not Null	Character

TestQuesBank

Attribute	Description	Constraint	Data Type
TestID	Identifies a particular test	Primary Key	Alphanumeric
TestNo	Test question number	Primary Key	Alphanumeric
TestQues	Test question	Not Null	Character

QuizQuesBank

Attribute	Description	Constraint	Data Type
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
QuizNo	Quiz question number	Primary Key	Alphanumeric
QuizQues	Quiz question	Not Null	Character

TestMrk

Attribute	Description	Constraint	Data Type
TestID	Identifies a particular test	Primary Key	Alphanumeric
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
TestMrk	Student test mark no.1	Null	Numeric
TestMrk1	Student test mark no.2	Null	Numeric
TestMrk2	Student test mark No.3	Null	Numeric
FirstTimeTest	Indicates that this is the first attempt	False	Boolean

SecondTimeTest	Indicates that this is the second attempt	False	Boolean
ThirdTimeTest	Indicates that this is the third attempt	False	Boolean
>ThreeTimes	Indicates that this is more than three attempts	False	Boolean

QuizMrk

Attribute	Description	Constraint	Data Type
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
QuizMrk	Student quiz mark	Null	Numeric
QuizMrk1	Student quiz mark no.2	Null	Numeric
QuizMrk2	Student quiz mark No.3	Null	Numeric
FirstTimeQuiz	Indicates that this is the first attempt	False	Boolean

SecondTimeQuiz	Indicates that this is the second attempt	False	Boolean
ThirdTimeQuiz	Indicates that this is the third attempt	False	Boolean
>ThreeTimes	Indicates that this is more than three attempts	False	Boolean

TestAnsBank

Attribute	Description	Constraint	Data Type
TestID	Identifies a particular test	Primary Key	Alphanumeric
TestNo	Test question number	Primary Key	Alphanumeric
TestAns	Answer to that particular test question	Not Null	Character

QuizAnsBank

Attribute	Description	Constraint	Data Type
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
QuizNo	Quiz question number	Primary Key	Alphanumeric
QuizAns	Answer to that particular quiz question	Not Null	Character

Student-TestMrk

Attribute	Description	Constraint	Data Type
TestID	Identifies a particular test	Primary Key	Alphanumeric
StudID	Uniquely identifies a student	Primary Key	Alphanumeric

Student-QuizMrk

Attribute	Description	Constraint	Data Type
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
StudID	Uniquely identifies a student	Primary Key	Alphanumeric

Student-TestQuesBank

Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
TestID	Identifies a particular test	Primary Key	Alphanumeric
TestNo	Test question number	Primary Key	Alphanumeric

Student-QuizQuesBank

Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
QuizNo	Quiz question number	Primary Key	Alphanumeric

Student-TestAnsBank

Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
TestID	Identifies a particular test	Primary Key	Alphanumeric
TestNo	Test question number	Primary Key	Alphanumeric

Student-QuizAnsBank

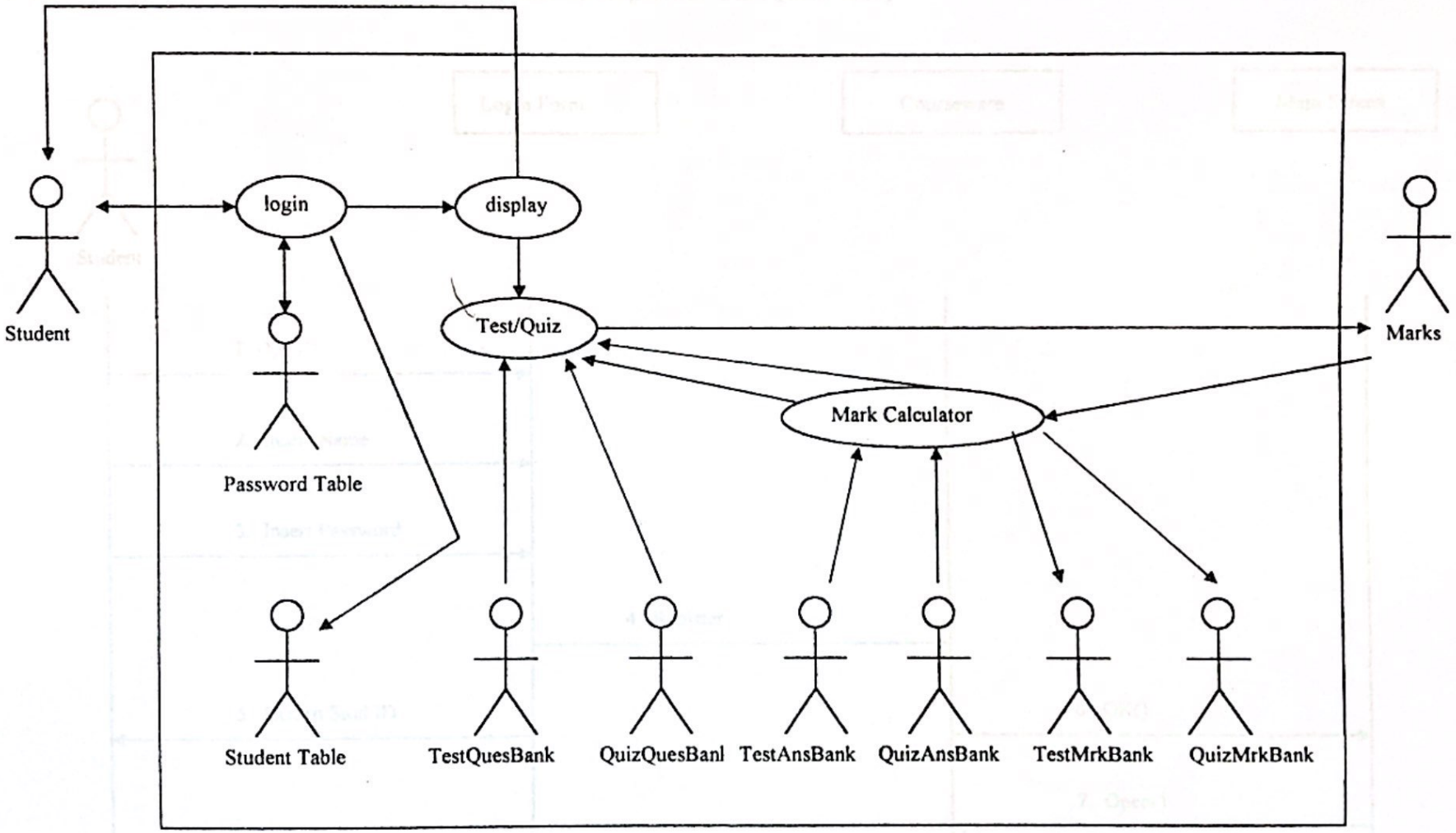
Attribute	Description	Constraint	Data Type
StudID	Uniquely identifies a student	Primary Key	Alphanumeric
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
QuizNo	Quiz question number	Primary Key	Alphanumeric

QuizOptions

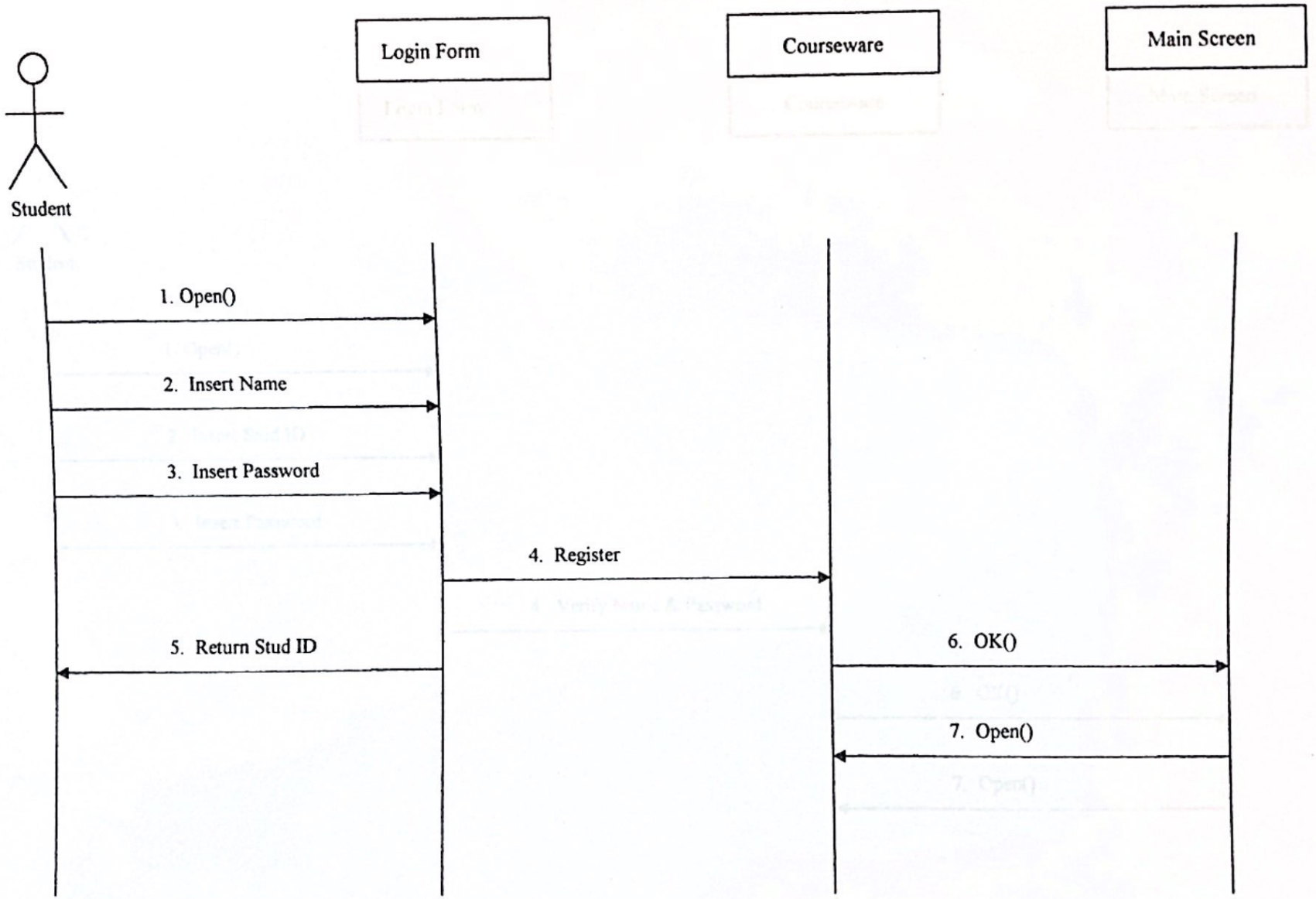
Attribute	Description	Constraint	Data Type
QuizID	Identifies a particular quiz	Primary Key	Alphanumeric
QuizNo	Quiz question number	Primary Key	Alphanumeric
QuizOption	Quiz options for objective questions	Not Null	Alphanumeric
QuizOptionAns	The correct answer for quiz/test	Not Null	Alphanumeric

5.2.6 Use Cases

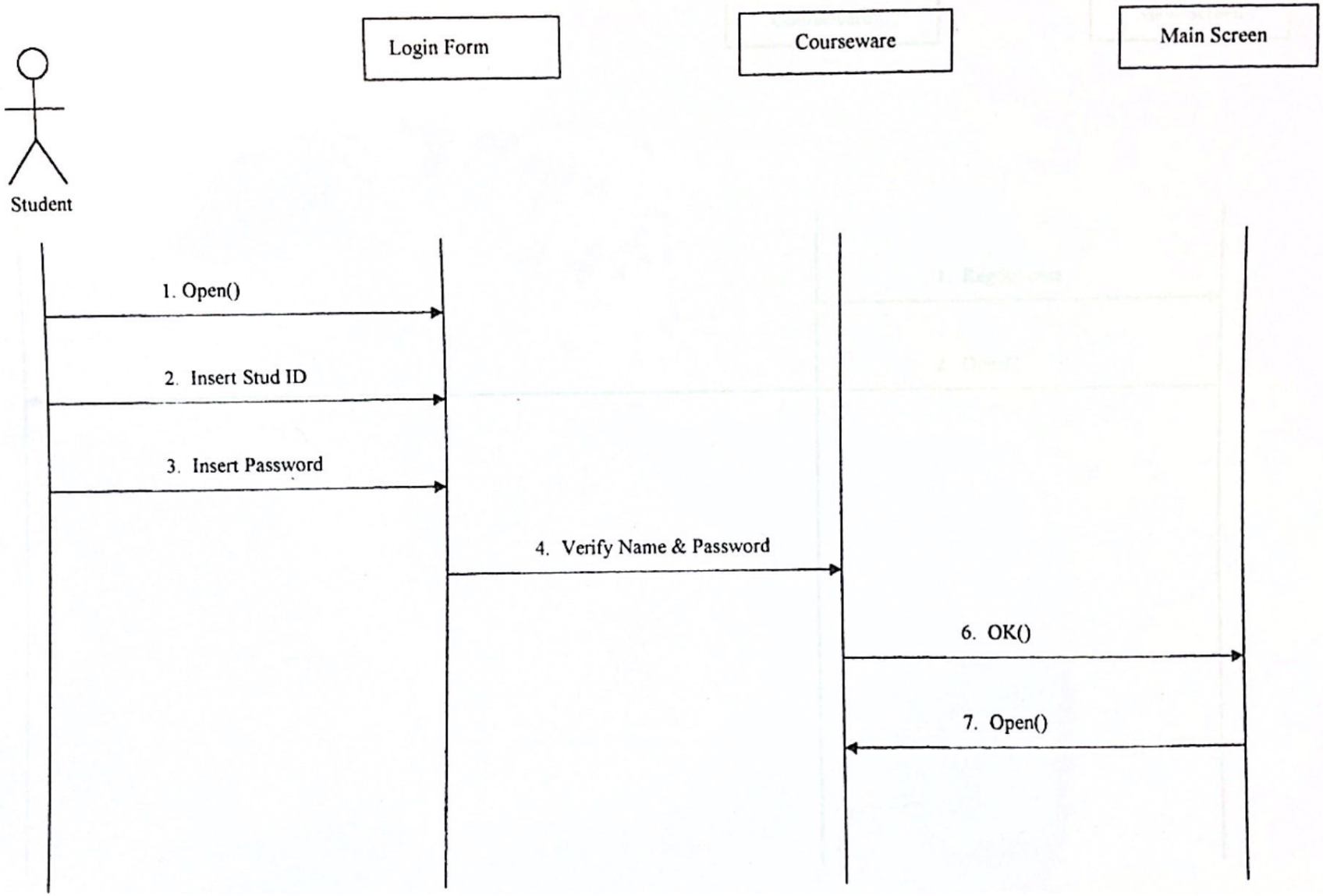
5.2.6.1 The Main Use Case Diagram



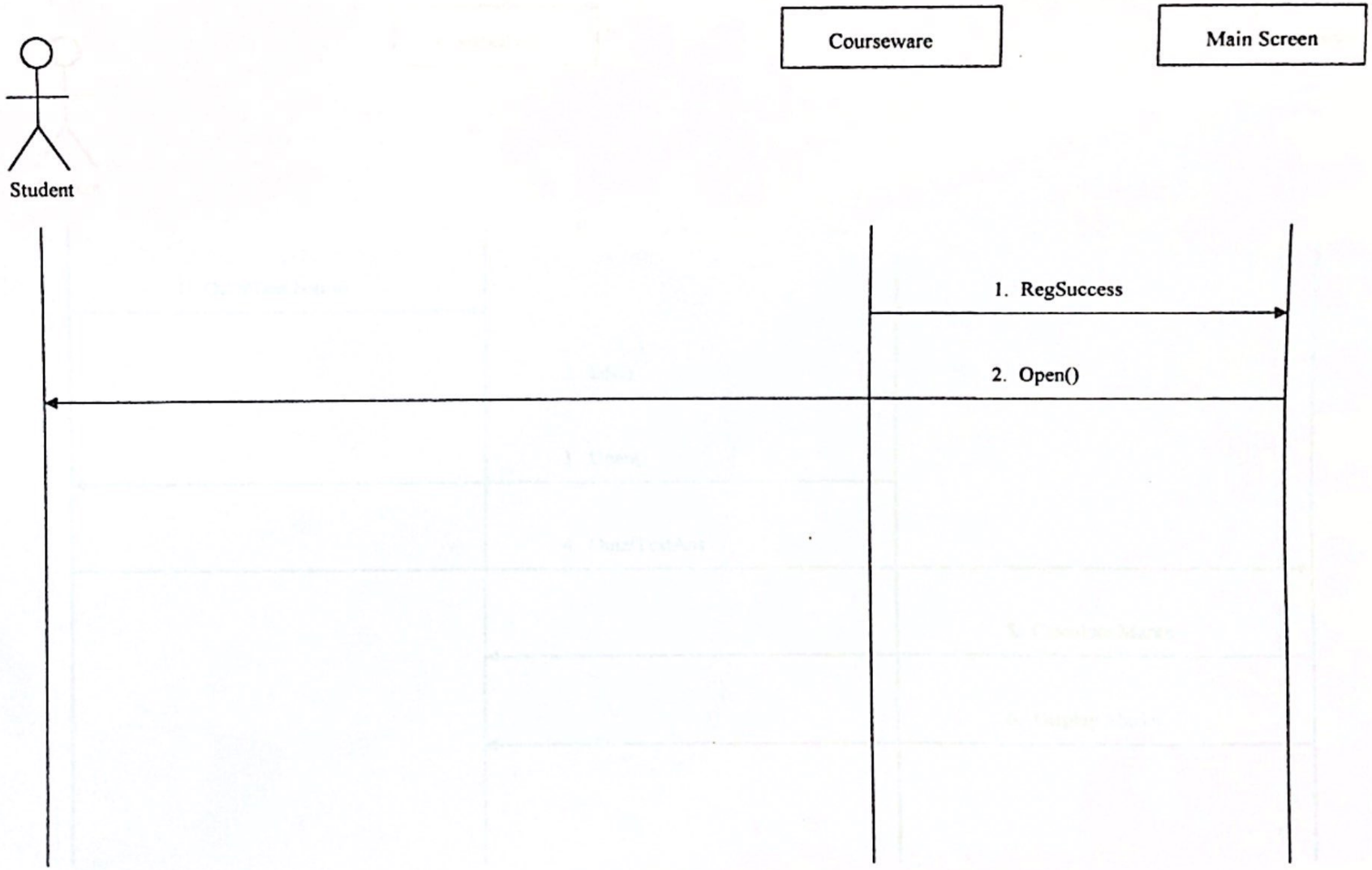
5.2.6.2 Happy Day Scenario Sequence Diagram for Login Use Case (new user)



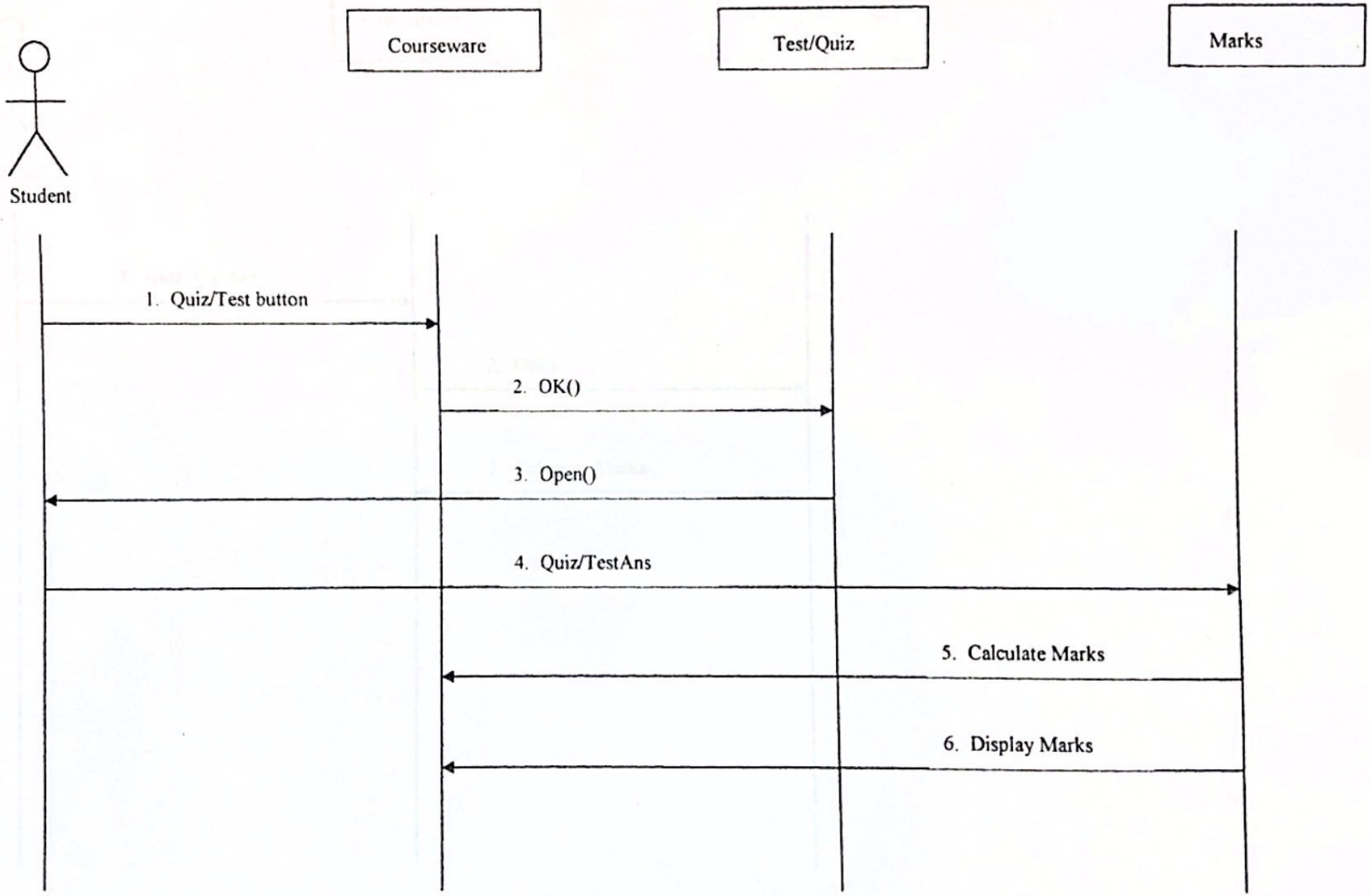
5.2.6.3 Happy Day Scenario Sequence Diagram for Login Use Case (registered/old user)



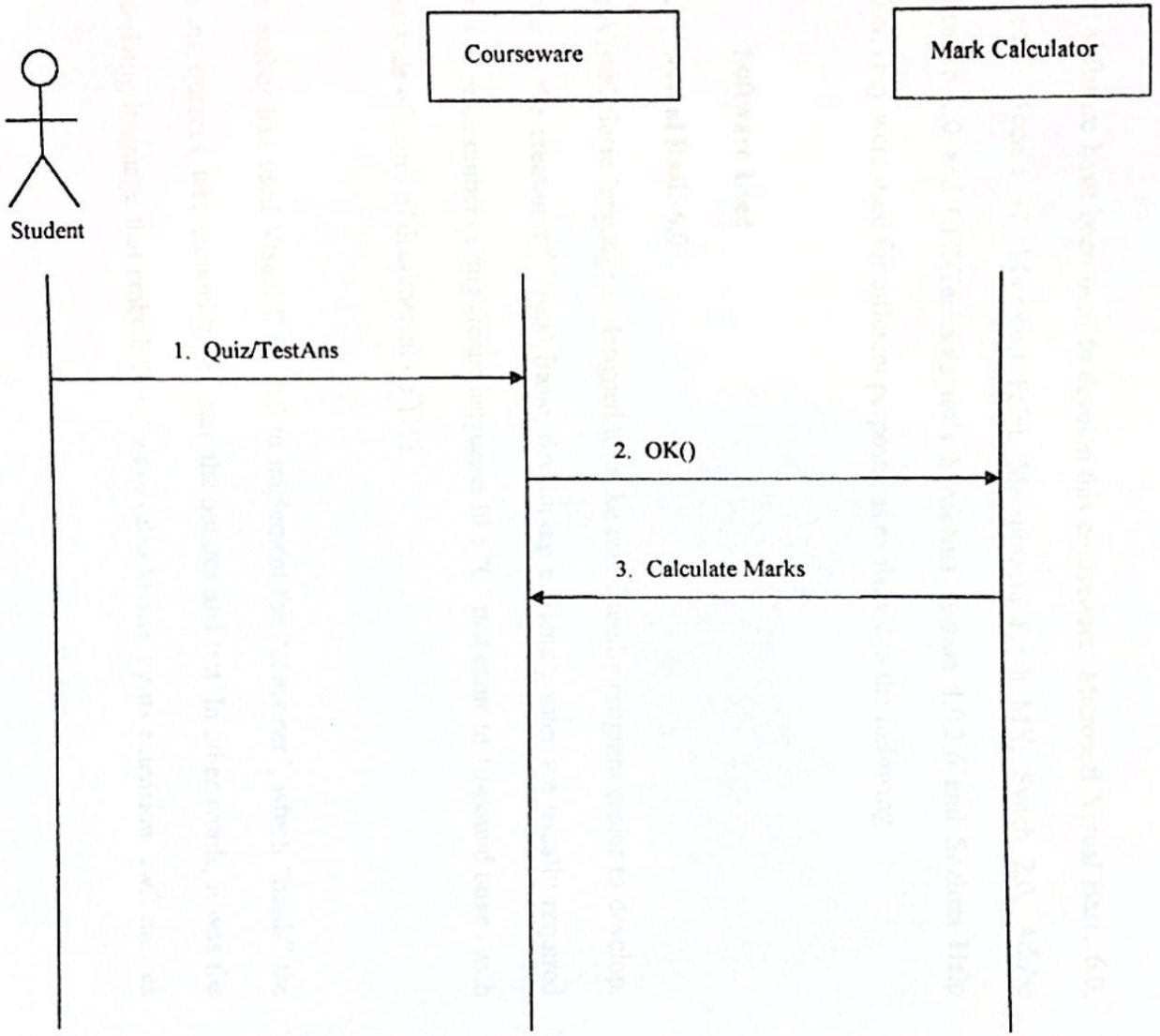
5.2.6.4 Happy Day Scenario Sequence Diagram for Display Use Case



5.2.6.5 Happy Day Scenario Sequence Diagram for Test/Quiz Use Case



5.2.6.6 Happy Day Scenario Sequence Diagram for Mark Calculator Use Case



CHAPTER 6

IMPLEMENTATION

Eight software have been used to develop this courseware: Microsoft Visual Basic 6.0, Microsoft Access 97, Microsoft Paint, Macromedia Flash MX, Swish 2.0, Adobe Photoshop 6.0 and CFS-Technologies's Speakonia version 1.0.3.6 and Shalom Help Maker.They were used for different purposes, as explained in the following:

6.1 Software Used

6.1.1 Visual Basic 6.0

The Visual Basic language is designed to make user-friendly program easier to develop. "Prior to the creation of Visual Basic, developing a friendly interface usually required teams of programmers using arcane languages like "C" that came in 10-pound boxes with thousands of pages of documentation"[19].

The author had used Visual Basic 6.0 to implement the "container", which "holds" the lessons, quizzes, test, etc and to program the quizzes and test. In other words, it was the underlying language that embeds Shockwave (also known by its extension .swf) movies

created using Flash and Swish. The quizzes and test are written using Visual Basic 6.0 which connects to the database that stores the quizzes/test questions and answers.

6.1.2 Macromedia Flash MX

Macromedia Flash MX was used to create the lessons. The author had chosen this software because of its capability to create non-linear lessons and of its ability to provide nice user interfaces. This solves the problem of dull user-interface provided by Visual Basic.

Upon research, the author has found out that it is possible to create more appealing user-interfaces in Visual Basic (they are also known as "skins") but because of the steep learning curve and time-constraints, she had to abandon the idea. This was because the author has to write her own program to create the skins. There are free skins available for downloads but according to the creator of the skins, some bugs were still unresolved [22]. Bearing this in mind, the author had decided to just improve the courseware's user-interface by using the available tools in Visual Basic, like adding colours to the forms. They could still be appealing, nevertheless, if done properly.

6.1.3 Microsoft Paint

The author used Microsoft Paint to draw pictures for the lessons. The apple tree mind map that appears in every chapter's "Objectives" was drawn using this software.

6.1.4 Adobe Photoshop 6.0

Adobe Photoshop was used for editing pictures created using Microsoft Paint. The tree and apples created in Microsoft Paint was not background-transparent. Therefore, the author has used a feature in Adobe Photoshop to achieve the result so that they can match with any background color of the courseware.

6.1.5 Microsoft Access 97

For the database, Microsoft Access 97 is linked to Visual Basic's controls (responsive objects that a programmer places on the screen).

6.1.6 Speakonia

Speakonia, a freeware, was used to create narrations for the diagrams. These narrations are saved as .wav files. They were then imported from Flash and Swish.

6.1.7 Swish

Swish was used to create lessons for Chapter 1 and 3. This software is similar to Flash.

6.1.8 Shalom Help Maker

Shalom Help Maker is a freeware used to create the help files for this courseware.

6.2 Why is implementation done using Visual Basic, Flash and Swish at the same time?

The fact that the courseware could be accomplished using the three software mentioned was because of the versatility of Visual Basic 6.0, which could import Shockwave Flash files. After creating the lessons in Flash MX and Swish, the author converted the files to Shockwave Flash files. The files were then imported from Visual Basic 6.0.

Why not creating everything in Flash MX or Swish? The reader might ask. Well, this is not done because the author wanted to have an advantage from the best of Flash MX, Swish and Visual Basic 6.0. Visual Basic 6.0 is easy to use and learn. It can also connect to Microsoft Access database directly by setting the properties of the controls in Visual Basic. Furthermore, it is similar to Windows-based Graphical User-Interface (GUI) that most of us are so familiar to. There are also ready-made controls where the programmer can just drag and drop onto the form. True, Visual Basic is simple to learn but is it not that easy when we delve deeper into it. There are classes, API and Registry programming where the author was not familiar in.

The advantage of Flash is the flexibility of creating pleasant user-interfaces to the programmer, or users' liking. Controls can be drawn by own-self. In fact, the author has drawn the buttons using Flash.

What about Swish? Swish is good in creating "instant" animations, especially text effects. We can also create text animations in Flash but it is time-consuming. The disadvantage of Swish is its limited capabilities compared to Flash.

The reader may deduce that the courseware is fast and easy to develop, since that author had taken the advantages of each software to produce it, it was not all the time a bed of roses. The author had to first learn how to use them before any programming can be done. There were integration problems like the inability of importing some features such as the hover effects in .swf files into Flash and Swish because of the limitations of both software.

To cut a long story short, Visual Basic, Flash and Swish has their own plus points and shortcomings. Put the good features together, and a courseware is born.

6.3 Implementation of the System using Visual Basic

Twenty one forms were created in this courseware. The first form that the user will see when he/she runs the program is the **Welcoming** form. There is a Flash introduction to the courseware on the page. The user can see the introduction or log in straightaway to the courseware. The codes for this form are as follows:

```
'For student Id
Public displayId As String 'this var contains student ID (got from database)

'For student Password
Public criteria1 As String
Public myPassword1 As String
Public myPassword As String 'student password

'For student name
Public criteria As String
Public myName1 As String
Public myName As String 'student name, it's a global variable: to insert this global variable,select
General and Declarations at the pull-down menus above
Public registered As Boolean 'sets to True if student has already registered
Private Sub cmdAdmin_Click()
Unload Me
If frmAdminLogin.LoggedIn = False Then
frmAdminLogin.Show
Else
frmAdmin.Show

End If

End Sub

Private Sub cmdClear_Click()
txtStudName.Text = ""
txtPassword.Text = ""
End Sub


Private Sub cmdHelp_Click()
CommonDialog1.HelpCommand = cdlHelpForceFile
CommonDialog1.helpfile = App.Path & "\SEC HELP.Hlp"
CommonDialog1.ShowHelp
End Sub


'Makes the fields visible after the user clicks on the New User button
Private Sub cmdNewUser_Click()
'MsgBox registered
'MsgBox LoginSucceeded
If Me.registered = True Or frmLogin.LoginSucceeded = True Then
```

```
    MsgBox "You have already registered/logged in. Please log in again to register a new student"
    txtStudName.Visible = False
    txtPassword.Visible = False
    lblStudName.Visible = False
    lblPassword.Visible = False
    cmdRegister.Visible = False
    lblStudNameMax.Visible = False
    lblPasswordMax.Visible = False
    cmdClear.Visible = False
Else
    txtStudName.Visible = True
    txtPassword.Visible = True
    lblStudName.Visible = True
    lblPassword.Visible = True
    cmdRegister.Visible = True
    lblStudNameMax.Visible = True
    lblPasswordMax.Visible = True
    cmdClear.Visible = True

    txtStudName.SetFocus

End If
End Sub
'Show the login form after the user clicks the Old User button
Private Sub cmdOldUser_Click()
'MsgBox LoginSucceeded
If frmLogin.LoginSucceeded = True Then
    frmMain.Show
Else
    Me.Hide
    frmLogin.Show
End If
End Sub

Private Sub cmdQuit_Click()
Dim iResponse As Integer
iResponse = MsgBox("Are you sure you want to quit?", vbYesNo, "Quit")
If iResponse = vbYes Then
    End
Else
    Me.Show
End If
End Sub

'Registration Process
Private Sub cmdRegister_Click()
''''''''''''''''''''''''''''''''''''''''''''''''''''''
'The following codes check whether the name the user
'has just entered is in the database or not


'txtStudName.Text = ""
'txtPassword.Text = ""
```

```
myName = txtStudName 'To assign txtStudName to myName
myPassword = txtPassword 'To assign txtPassword to myPassword

If myName = "" Or myPassword = "" Then
MsgBox "Name and Password cannot be empty!"

Else:
If Len(myPassword) < 8 Then
MsgBox "Password is less than eight characters"

Else
myName1 = "StudName="
criteria = myName1 & "'" & myName & "'"
Data2.Recordset.FindFirst criteria
If Data2.Recordset.NoMatch = False Then
'MsgBox "Name and/or password taken, please register with new ones!"
MsgBox "The name you register is already taken, please register with a new one!"
'MsgBox "Name and Password cannot be empty!"

txtStudName.SetFocus
Me.Show
Else

'txtPassword.SetFocus
'SendKeys "{Home}+{End}"
''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'The following codes check whether the password the user
'has just entered is in the database or not
'myPassword1 = "Password="
'criteria1 = myPassword1 & "'" & myPassword & "'"
'Data1.Recordset.FindFirst criteria1
'If Data1.Recordset.NoMatch = False Then
'MsgBox "Name and/or Password taken, please register with new ones!"
'txtStudName.SetFocus
'Me.Show
'Else

''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'The following codes Register the new student
'(put in Student database)

If Data2.Recordset.EOF Then 'if no record
Data2.Recordset.AddNew
Data2.Recordset.Fields("StudID").Value = 1 'create student ID for the first student
displayId = Data2.Recordset.Fields("StudId").Value 'to get student ID
Data2.Recordset.Fields("StudName").Value = myName
Data2.Recordset.Update

Else
Data2.Recordset.MoveLast
previousID = Data2.Recordset.Fields("StudID").Value
Data2.Recordset.AddNew
```

```
Data2.Recordset.Fields("StudID").Value = previousID + 1
displayId = Data2.Recordset.Fields("StudId").Value 'to get student ID
Data2.Recordset.Fields("StudName").Value = myName
Data2.Recordset.Update

End If

'the following codes put the student's password into Password database
Data1.Recordset.AddNew
Data1.Recordset.Fields("StudID").Value = displayId
Data1.Recordset.Fields("Password").Value = myPassword

Data1.Recordset.Update


registered = True
Me.Hide
'Unload Me
frmMain.Show

End If
End If
End If
End Sub
'When this form loads, all the fields are invisible
Private Sub Form_Load()
Data2.DatabaseName = App.Path & "\Student.mdb"
Data1.DatabaseName = App.Path & "\Password.mdb"

txtStudName.Visible = False
txtPassword.Visible = False
lblStudName.Visible = False
lblPassword.Visible = False
lblStudNameMax.Visible = False
lblPasswordMax.Visible = False

registered = False

ShockwaveFlash1.Movie = App.Path & "\tour1.swf"
ShockwaveFlash2.Movie = App.Path & "\logo.swf"
End Sub

Private Sub txtPassword_GotFocus() 'if txtPassword got focus, highlight the password
txtPassword.SelStart = 0
txtPassword.SelLength = Len(txtPassword)
End Sub
Private Sub txtStudName_GotFocus() 'if txtStudName got focus, highlight the student name
txtStudName.SelStart = 0
txtStudName.SelLength = Len(txtStudName)
End Sub
```

If the user is new, he/she can click the **REGISTER** button to register. Upon clicking the button, the user has to fill in his or her user name and password. The maximum length for the name is twenty while the password is ten. The minimum length of the password is eight. As a guide to the user, these maximum and minimum lengths are indicated on the form. Anyway, the courseware is written in such a way that it will not let the user to fill in more than the required characters.

On the other hand, if the user has already registered for the courseware, he/she needs to click on the **LOGIN** button. The user then enters his/her user name and password. The maximum lengths for both are twenty and eight respectively.

After clicking the **CONTINUE** button (for New User) or the **OK** button (for Old User), a second **Welcoming** form appears. The user is greeted and his/her user ID is provided. The user does not have to remember his/her ID as the purpose of showing the ID is to let the user know his/her position as an *nth* user in the courseware. Let's say that the user's ID is 23. This means that he/she is the 23rd user who has registered for the system.
For the courseware, however, the ID is very important because it functions as the primary key for a particular user.

In addition to the greeting and ID, there are also five buttons for the user to click on: **Chapter 1, Chapter 2, Chapter 3, Test (Chapter 1 to 3) and Student Report**. Chapters 1,2 and 3 are lessons while Test (Chapter 1 to 3) is a test, as its name implies. Student Report displays the test and quiz marks.

In each chapter, there is a **lesson** and a **quiz.** Students can look at the lessons and take a quiz after that. The author will only explain about the quiz here as the lesson is described under the **Implementation of the System using Flash and Swish** in Section 6.4.

The are six objective questions for the quizzes and they appear randomly. There is a random number generator function behind the screen. Therefore, there will be different sets of questions each time the user take the quiz. Because of time-limitations, there are only 15 questions in the question bank so the questions may repeat. The codes for the random number generator are as follows:

```
'Generate random numbers and put them in array where each of them is unique
Function GetRandomNumbers()

'Dim i As Integer
Dim j As Integer 'no. of comparisons
Dim temp As Integer 'temp storage of random numbers

'i = 0

Do
  Randomize
  temp = Int(15 * Rnd) + 1 'randomize number, put in temp
  'MsgBox temp
  If i > LBound(numArr) Then 'is this not the first round?
    j = LBound(numArr)
    Do
      If numArr(j) <> temp Then 'if the current random number is not a repeated number
        j = j + 1
      Else
        Call GetRandomNumbers
        Exit Function 'exit if the current random number is already in array (so don't put it in array
anymore)
      End If
    Loop Until j = i
  End If
  numArr(i) = temp
  'MsgBox numArr(i)
  'Call GetRandomQuestions(numArr, i) 'pass the whole array and i
```

```
    i = i + 1
  Loop Until i > UBound(numArr) - LBound(numArr)
End Function
```

There are three pages(forms) altogether for a certain Quiz. To proceed to the next question, the user clicks the **Next** button. Respectively, the user clicks **Back** to go to the previous screen. At the last page of the quiz, there is a Submit button where the user can click, after he/she is satisfied with the answers. Upon clicking the button, three marks are shown on the screen. They are the **current, last attempt** and **second-last attempt** marks. This is because the courseware is designed in a way that stores the marks of *the last three attempts* that a user has made. If the submit button is not clicked, the marks will not be counted and therefor will not be stored in the database.

If the current attempt is the one and only attempt that a user has made, the word "not applicable" will be displayed at the last attempt and second-last attempt fields. Likewise, if the current attempt is the second attempt of a particular user, the word "not applicable" will be displayed at the second-last attempt's field. The last-attempt field will be filled with the previous marks that the user got.

The codes to get random questions are as follows:

```
'Get questions based on the random numbers just generated
Function GetRandomQuestions(DataControl As data, textBoxName As TextBox, k As Integer, chpt As
Integer, textBoxIndex As Integer)
Dim data As String 'temp storage of question
DataControl.RecordSource = "select * from QuizQuesBank where QuizId= " & chpt & "AND
QuizNo= " & numArr(textBoxIndex + k) & ""
DataControl.Refresh
If DataControl.Recordset.NoMatch = False Then 'if found
textBoxName = DataControl.Recordset.Fields("QuizQues")
```

```
    Else
    MsgBox "No such question in database"
    End If

End Function
```

The features of the test are about the same as the quizzes', in terms of random questions, navigation buttons and display of quiz marks. The only difference is the number and type of questions in the test. There are 6 objective questions, 5 fill-in-the-blanks and 5 true-false questions for the test. In the question bank, there are 15 objective questions (taken from the quiz question bank), 15 fill-in-the-blanks and 15 true-false questions. All questions for quizzes and test were stored in Microsoft Access 97 database.

6.4 Implementation of the System using Flash and Swish

The three lessons (chapters) were done mainly using Flash and Swish, although some pictures were drawn using Microsoft Paint and edited using Macromedia Photoshop. The narration was done using Speakonia. Figure 8 is the screenshot of the **Objectives** screen in Chapter 2. Chapter 2 is done using Flash.

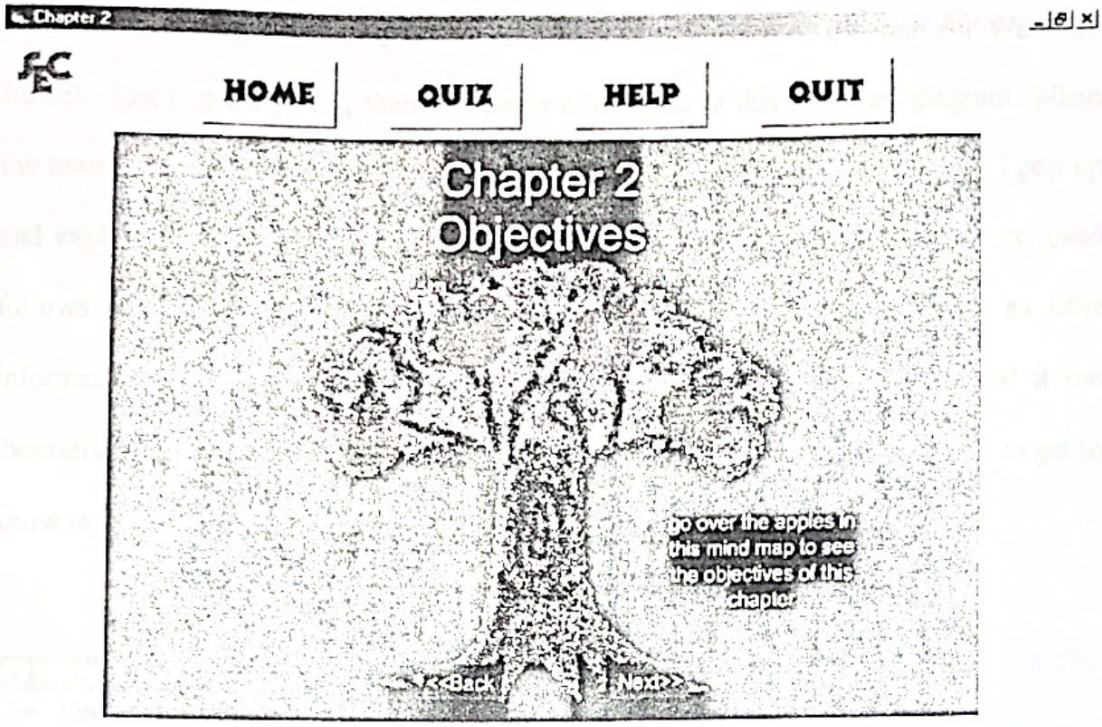


Figure 8 Screenshot of Chapter 2 Objectives

When the user rolls their mouse over one of the apples, a tooltip will pop up and states one of the objectives of Chapter 2. Each of the apples contains objectives for that chapter. Below is the Actionscript on how the hover effect is achieved.

```
on (rollOver) {
  attachMovie("objective1","windowMC",1);
  windowMC._x = 85.9;
  windowMC._y = 113.9;
}

on (rollOut) {
  windowMC.removeMovieClip();
}
```

Chapter 1 and 3 is done using Swish. Figure 9 is the animated diagram for Waterfall Model. Just like Chapter 2, there are hover effects too in this animated diagram. When the mouse is rolled over a particular phase in the Waterfall model, a tool-tip will pop up and explains about the model, as can be seen in the diagram. The hover effects used follows the minimalist approach to learning, where the user is only given as little information as possible. If the users need more information, they should find it out themselves. In this example, the user can roll their mouse over a particular phase to get to know more about it.

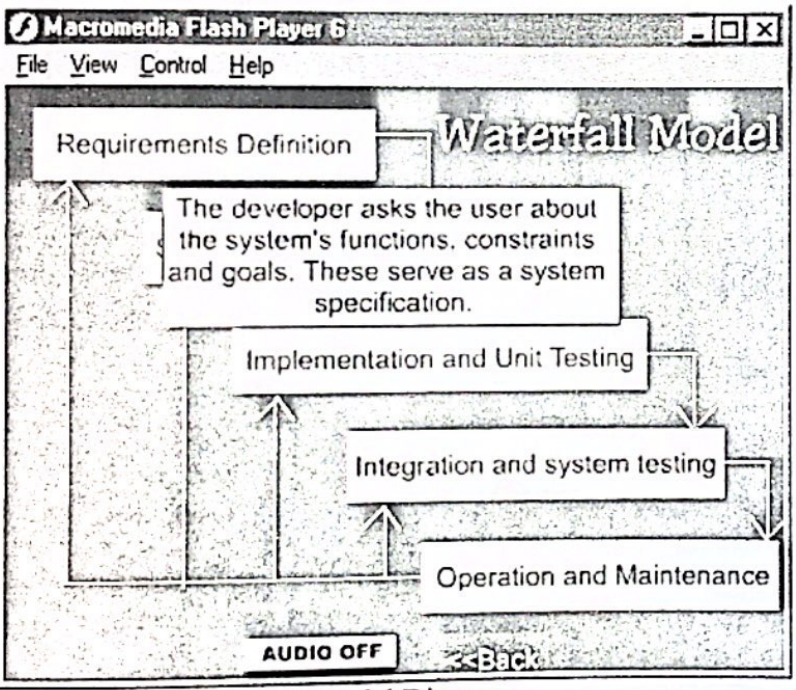


Figure 9 Waterfall Model Diagram

Figure 10 are the actions for the roll over effects:

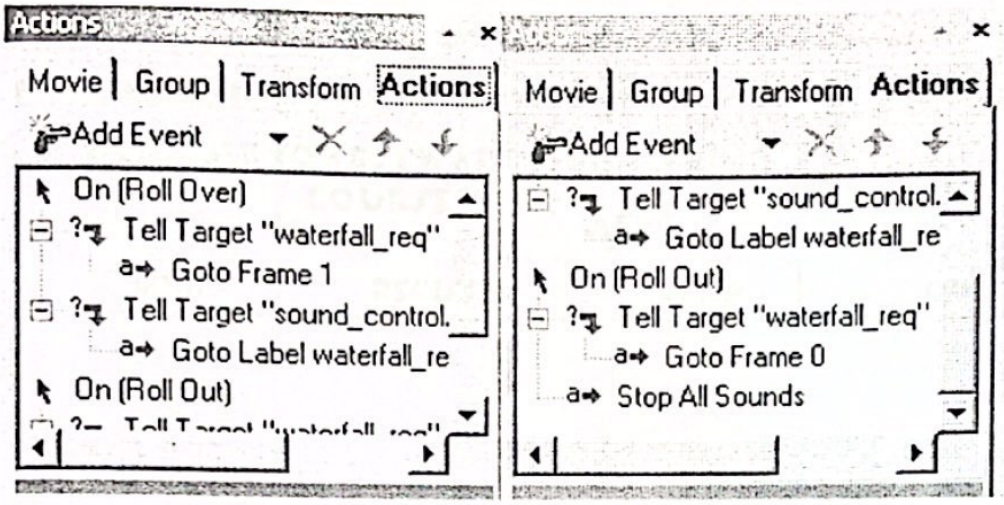


Figure 10 Actions for Waterfall Diagram

6.5 User Interface Design

The following are the main interfaces of the courseware.

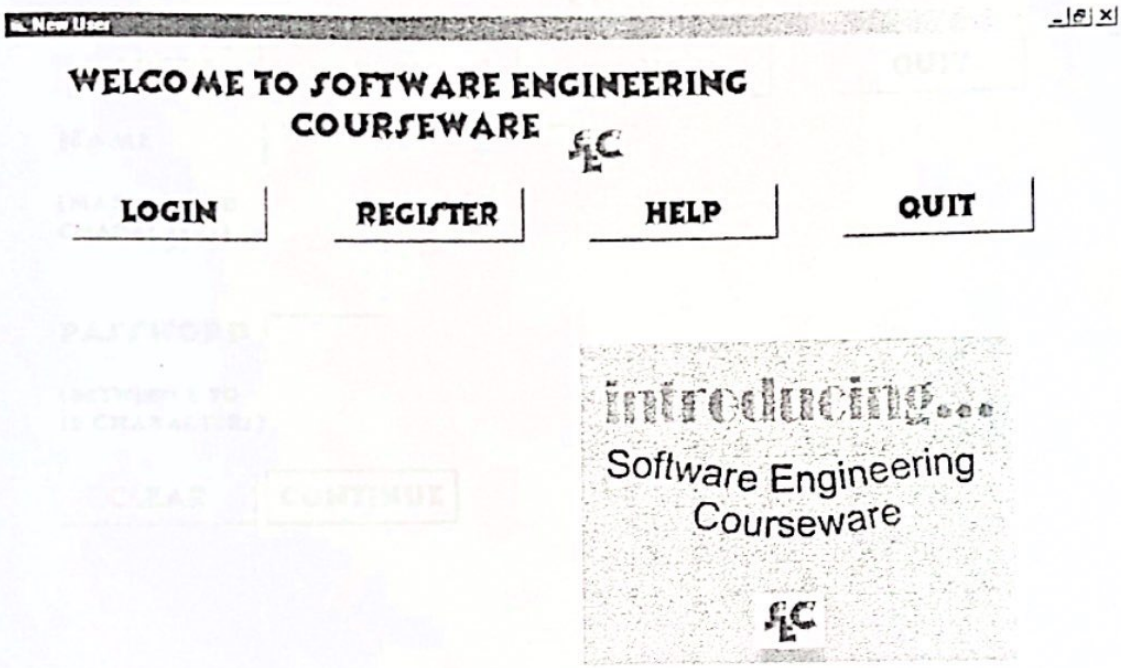


Figure 11 Login/Registration Screen

This is the very first screen that the user will see upon starting the courseware. The movie on the right serves as an introduction and the user can choose to view or ignore it. For a new user, he/she should click the REGISTER button to, as the name implies, register hi/herself. The next time around, the user can click LOGIN to log into the courseware. If the user needs help in navigating the courseware, the HELP button can be clicked. To exit from the courseware, click QUIT.

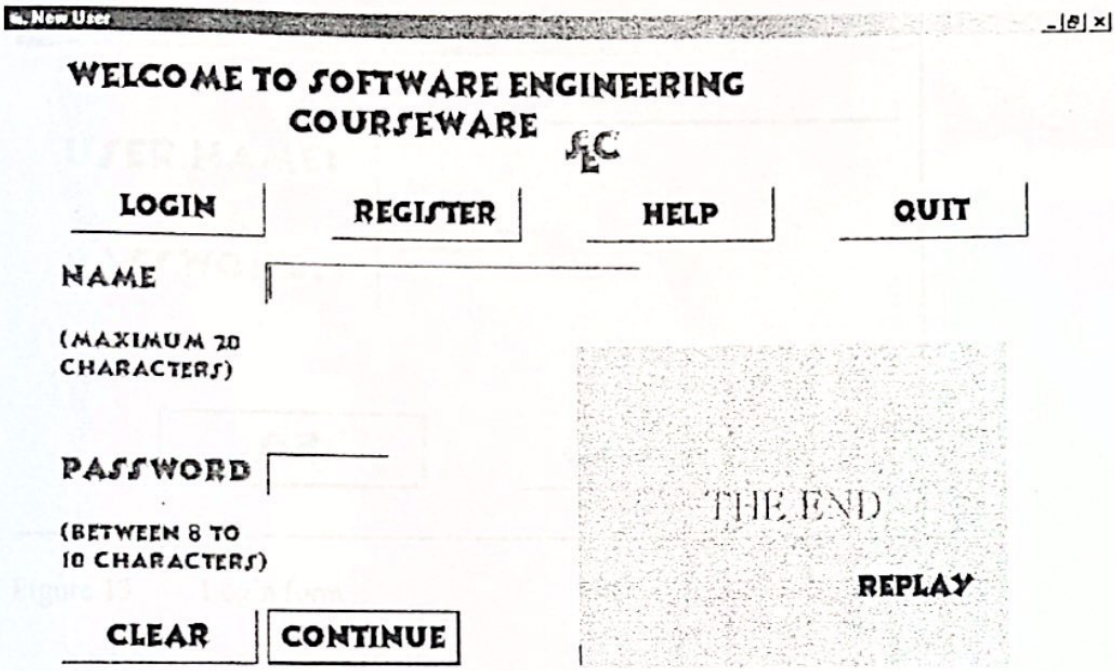


Figure 12 Registration Form

When REGISTER is clicked, a form appears. The user has to key in his/her name and password according to the minimum and maximum lengths shown on the screen. To proceed, CONTINUE is clicked. To clear the name and password, CLEAR is clicked.

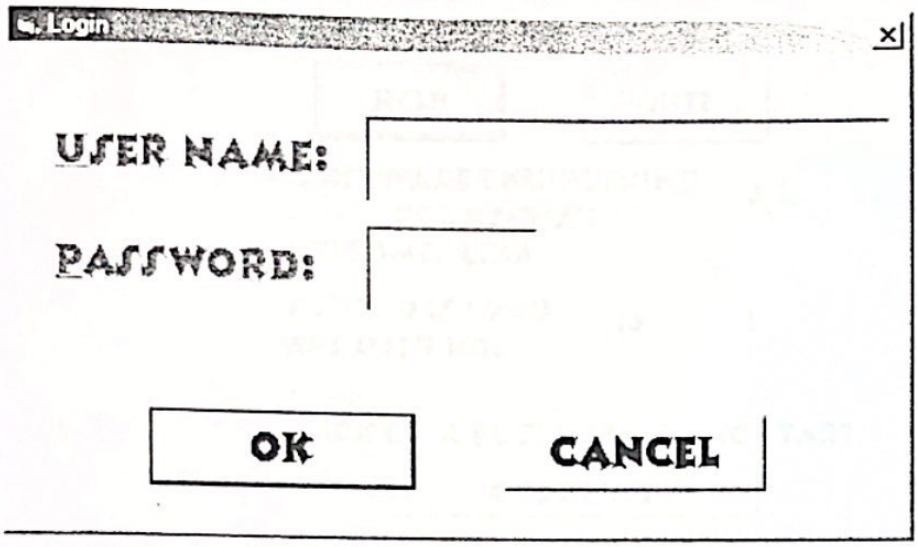


Figure 13 Login form

When the user clicks LOGIN, this form will appear. The user then keys in his/her correct name and password. OK is clicked to proceed while CANCEL is clicked to halt the login.

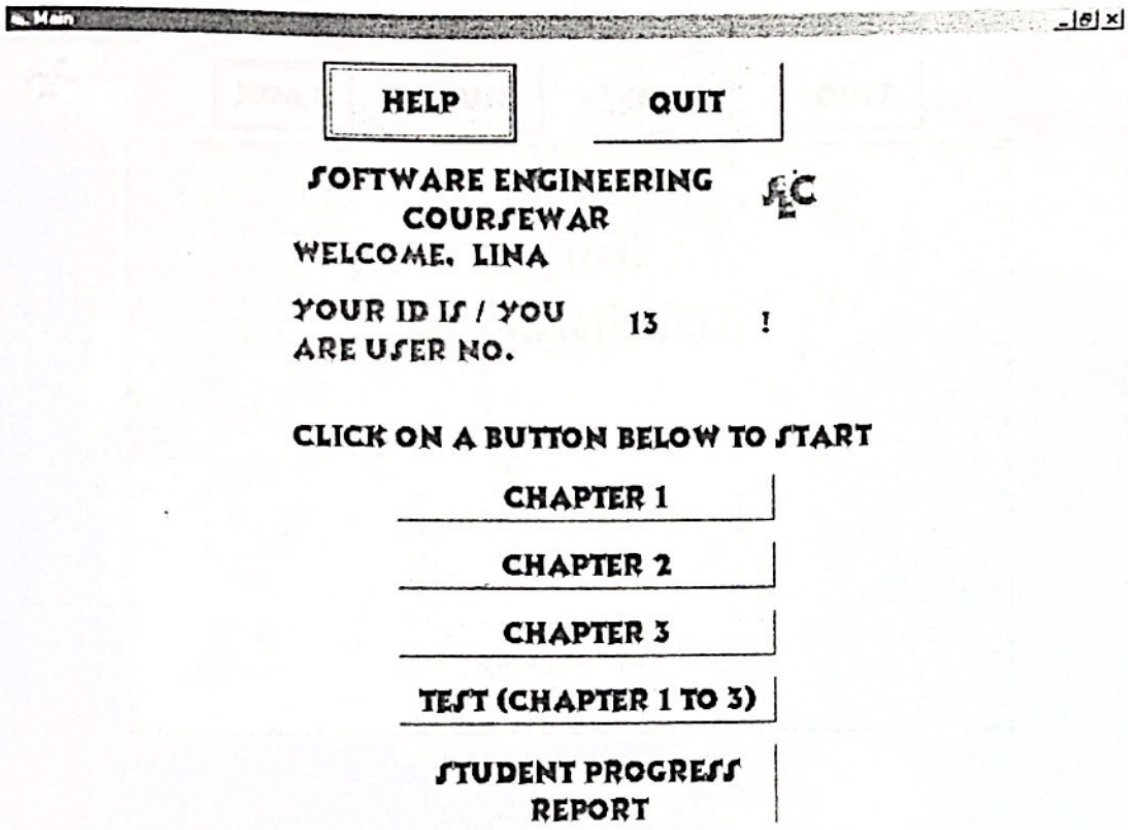


Figure 14 Main Screen

This is the main menu of the courseware. There are buttons, when clicked, will bring the user to specific chapters. The TEST button is clicked if the user wants to do a test that is based on Chapters 1 to 3. The STUDENT PROGRESS REPORT displays the user's test and quiz marks for the last three attempts.

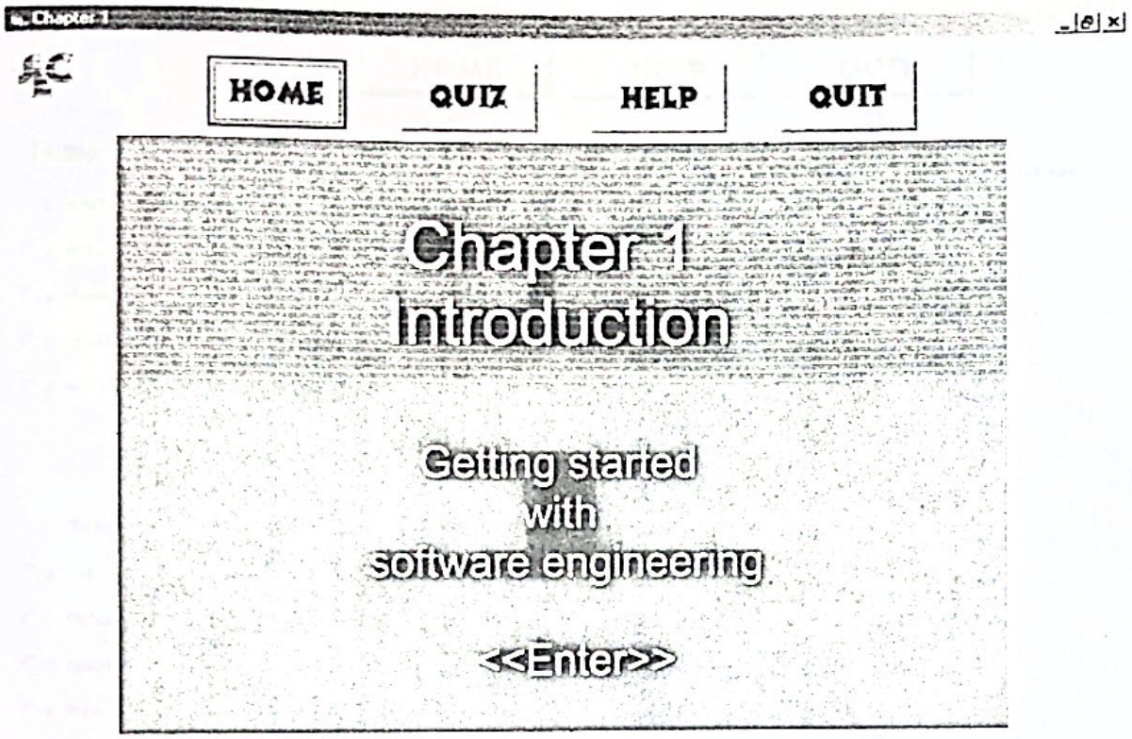


Figure 15 Chapter Screen

This is the Chapter Screen where the lessons for that particular chapter are displayed. On the top of the screen, as we can see, is the HOME, QUIZ, HELP and QUIT button. The HOME button will bring the user to the Main Screen. If the user wants to attempt a quiz for this chapter, the QUIZ button is clicked. The Help button will bring users to the HELP Screen. As its name implies, the QUIT button enable the user to exit the courseware.

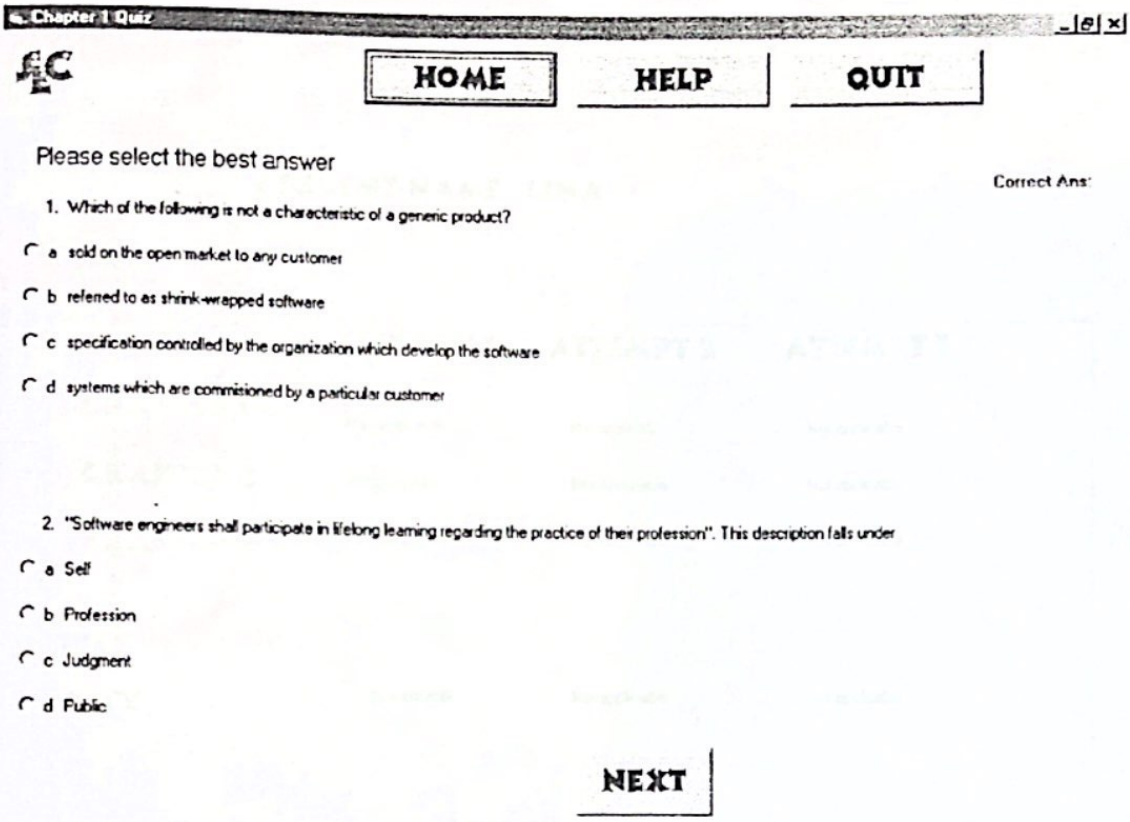


Figure 16 Quiz/Test Screen

This is the screen where students take quizzes. Students have a few options of answers they can choose from. They can proceed to the next question by clicking NEXT. After the student has satisfied with his/her answers, he/she can click SUBMIT at the last page so that the quiz/test can be corrected and marks will be given.

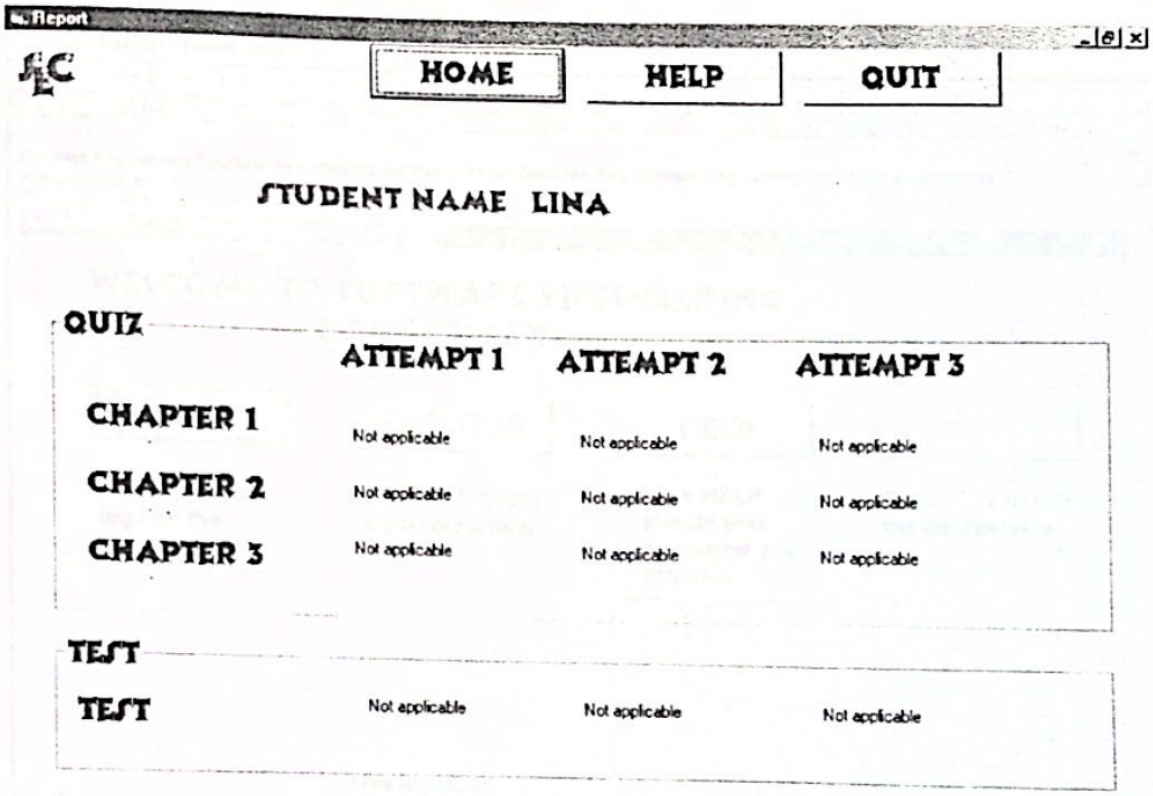


Figure 17 Student Report

This screen shows the last three quiz and test marks that the user got. It is hoped that with the display of the previous quiz and test marks, the user would be able to know where does he/she stand so that further improvements can be made.

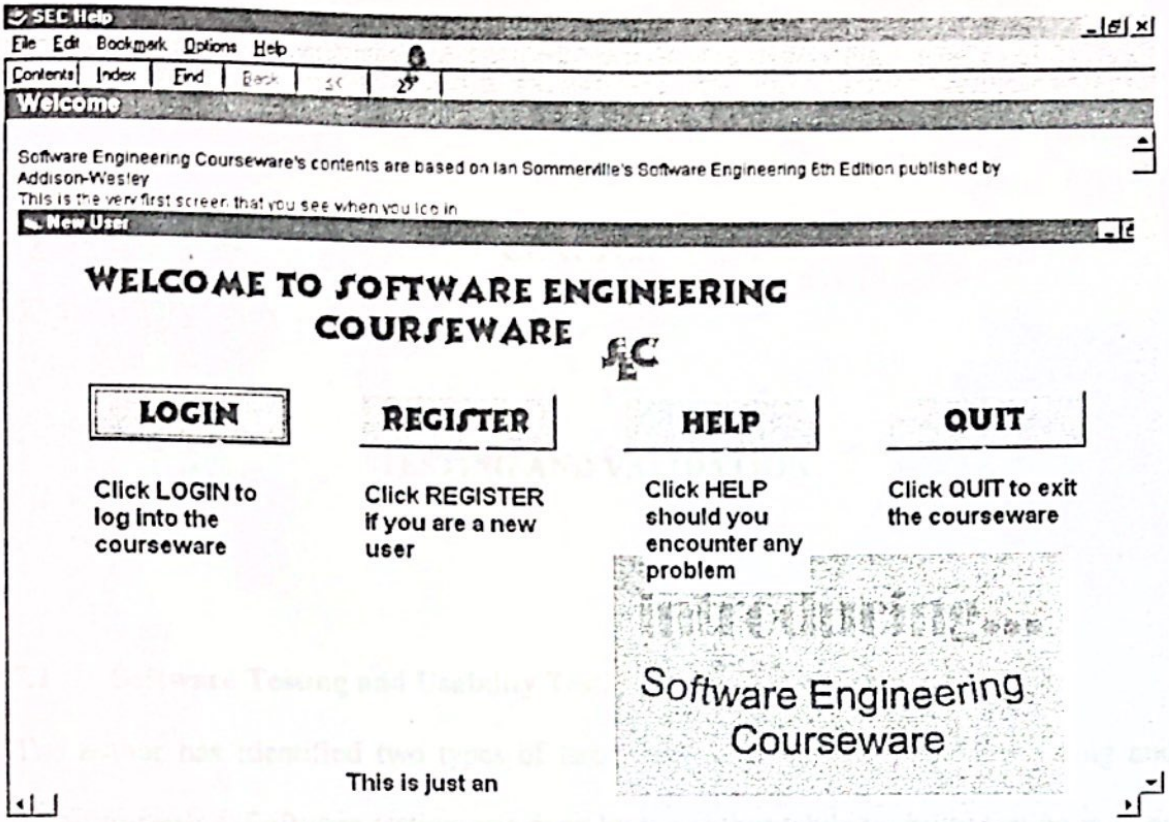


Figure 18 Help Screen

This HELP screen is intended to guide users through the courseware. Every navigation buttons are explained so that the courseware is user-friendly. The HELP button is available throughout the courseware.

CHAPTER 7

TESTING AND VALIDATION

7.1 Software Testing and Usability Testing

The author has identified two types of tests for the courseware: software testing and usability testing. Software testing was done by the author while usability testing is done by a test subject. There are two types of software testing that the author utilised: component and integration testing.

7.1.1 Component Testing

This involves testing of individual program components. In this case, the author has tested the individual functions in the system, to make sure that they do what they are supposed to do. The functions are also tested for defects (in this case, errors). In order to perform Component Testing, the author has provided test data and test cases. The test cases are classified into Equivalence Partitioning, so that only one case is chosen from each partition. The test cases for the courseware are shown in Section 7.1.1.1

7.1.1.1 Test Cases

Test Data No.1 : **Registration Form**

Input Data No.1 : None.

Description : Name and Password cannot be empty (null).

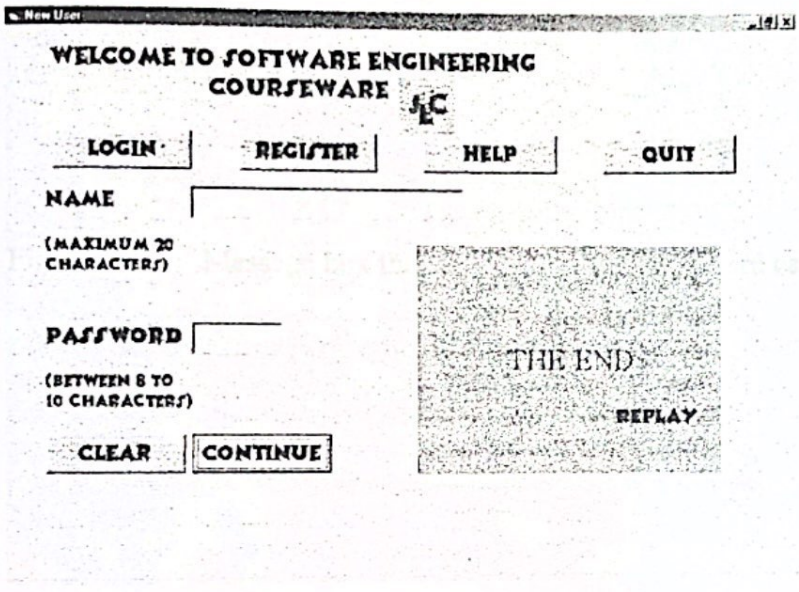


Figure 19 Registration form with empty name and password.

Expected Response : A message box that pops up and says "Name and Password cannot be empty"

Actual Response : A message box that pops up and says "Name and Password cannot be empty"

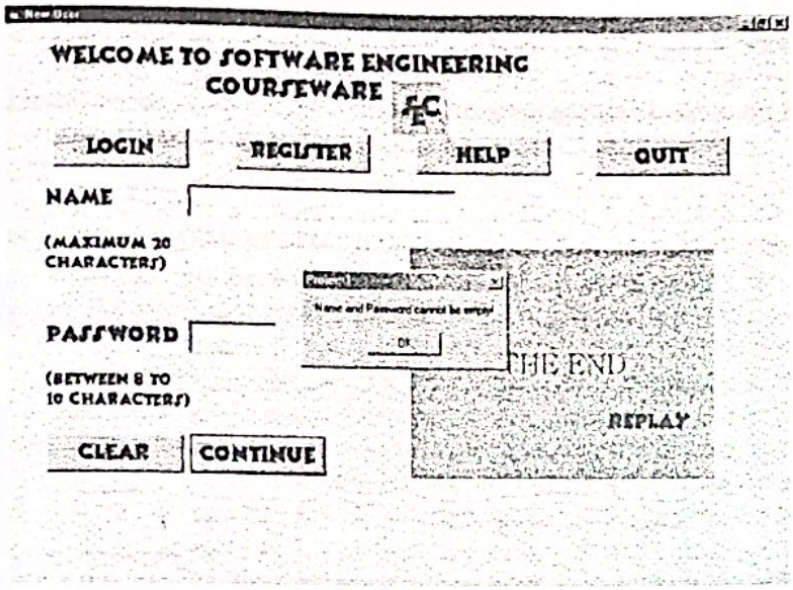


Figure 20 Message box that says "Name and Password cannot be empty"

Input Data No.2 : Password that is less than eight characters.

Description : Password must be at least eight characters.

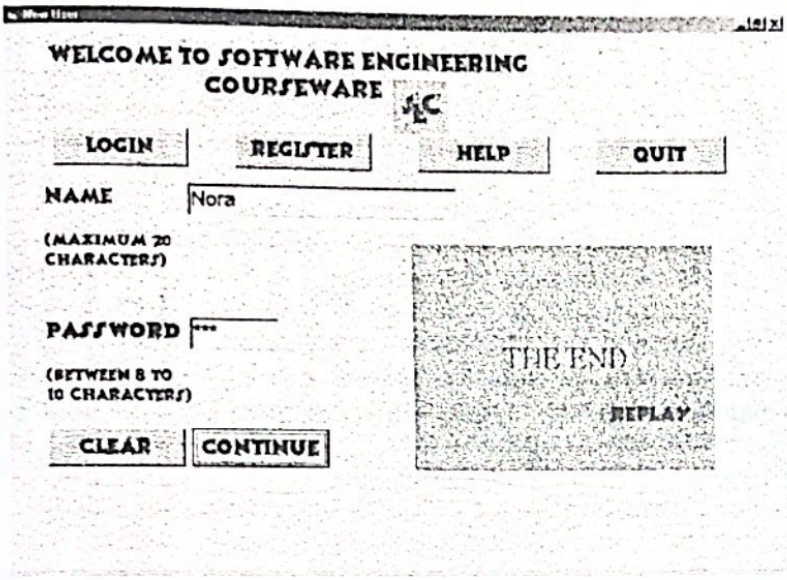


Figure 21 Registration form with the password less than eight characters.

Expected Response : A message box that pops up and says "Password is less than eight characters"

Actual Response : A message box that pops up and says "Password is less than eight characters"

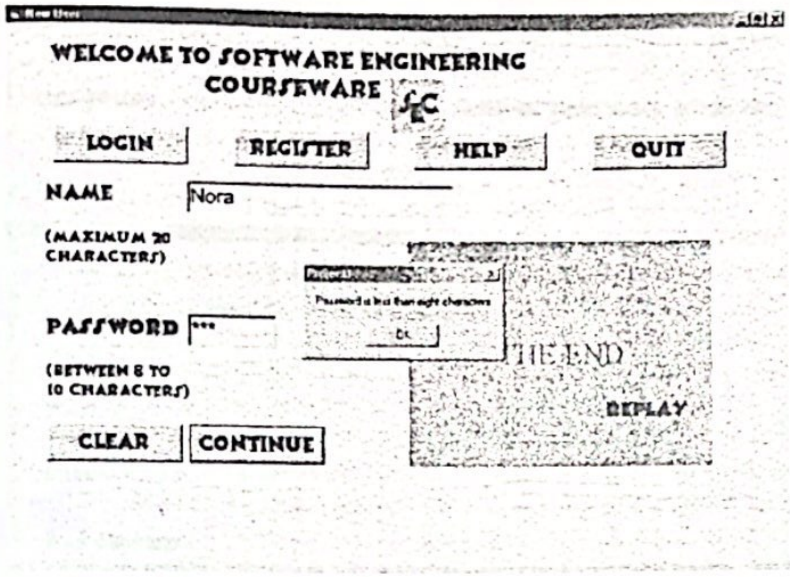


Figure 22 Message box that says "Password is less than eight characters"

Input Data No.3 : Empty (null) name and/or password

Description : Name and/or password must not be empty (null)

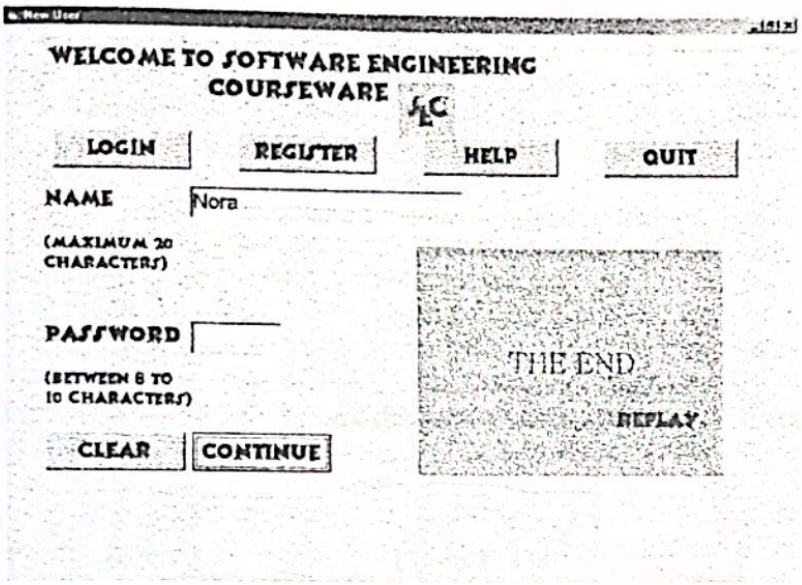


Figure 23 Registration form with empty password.

Expected Response : A message box that pops up and says "Name and password cannot be empty"

Actual Response : A message box that pops up and says "Name and password cannot be empty"

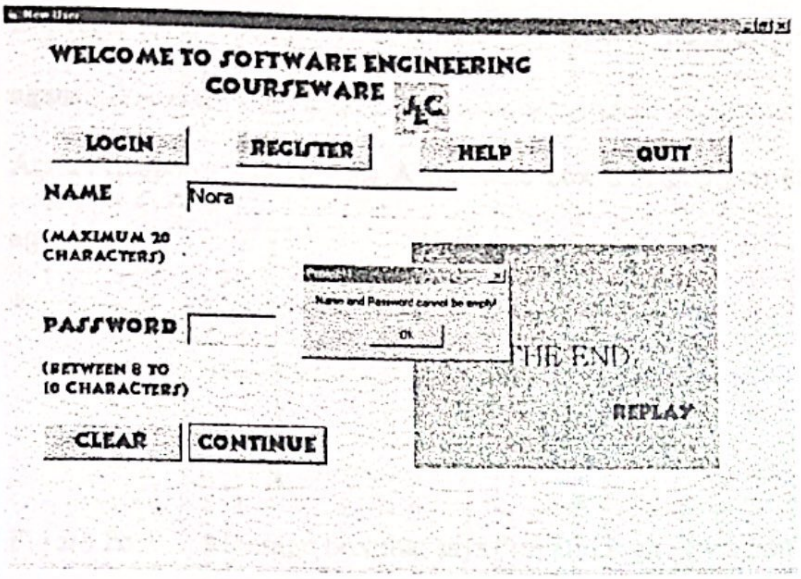


Figure 24 Message box that says "Name and password cannot be empty"

Test Data No.2 : **Login Form**

Input Data No.1 : None, or Name and Password that does not match.

Description : Name and Password must match with those stored in the database.

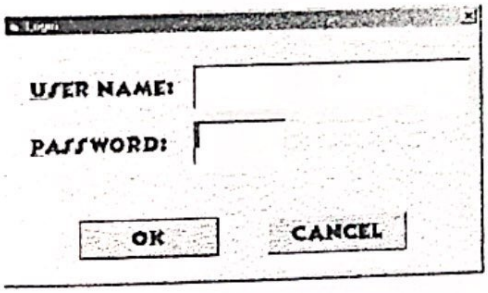


Figure 25 Login form with empty or invalid name and password.

Expected Response : A message box that pops up and says "Invalid login, try again"

Actual Response : A message box that pops up and says "Invalid login, try again"

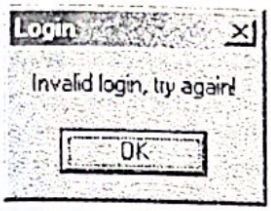


Figure 26 Message box that says "Invalid login, try again"

Input Data No.2 : None, or Name and Password that match.

Description : Name and Password should match with those in the database.

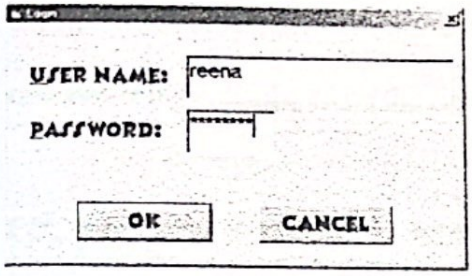


Figure 27 Login form with correct name and password.

Expected Response : The main(menu) screen appears.

Actual Response : The main(menu) screen appears.

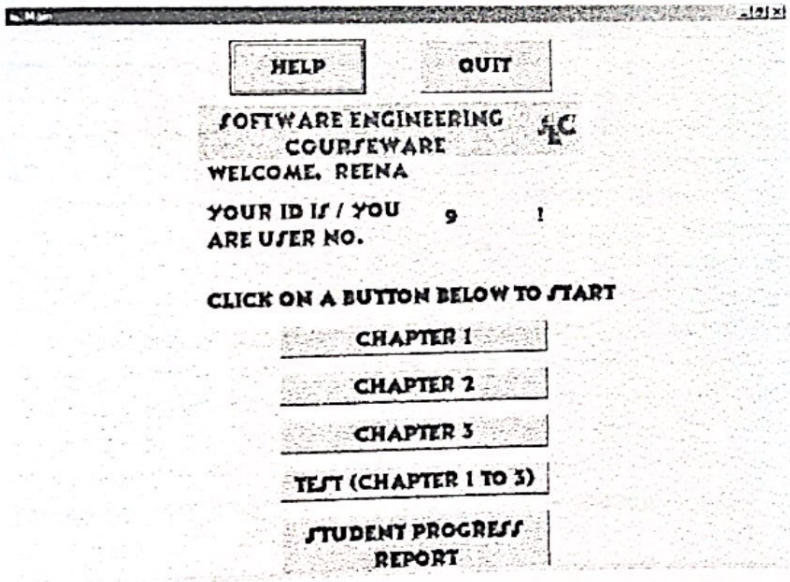


Figure 28 Main (menu) screen.

Test Data No.3 : **Quiz**

Input Data No.1 : Incorrect or blank Answer

Description : If an answer is incorrect or blank, there is no "tick" beside the question number after the student submitted the answers. The correct answer for each question will be displayed on the right hand side.

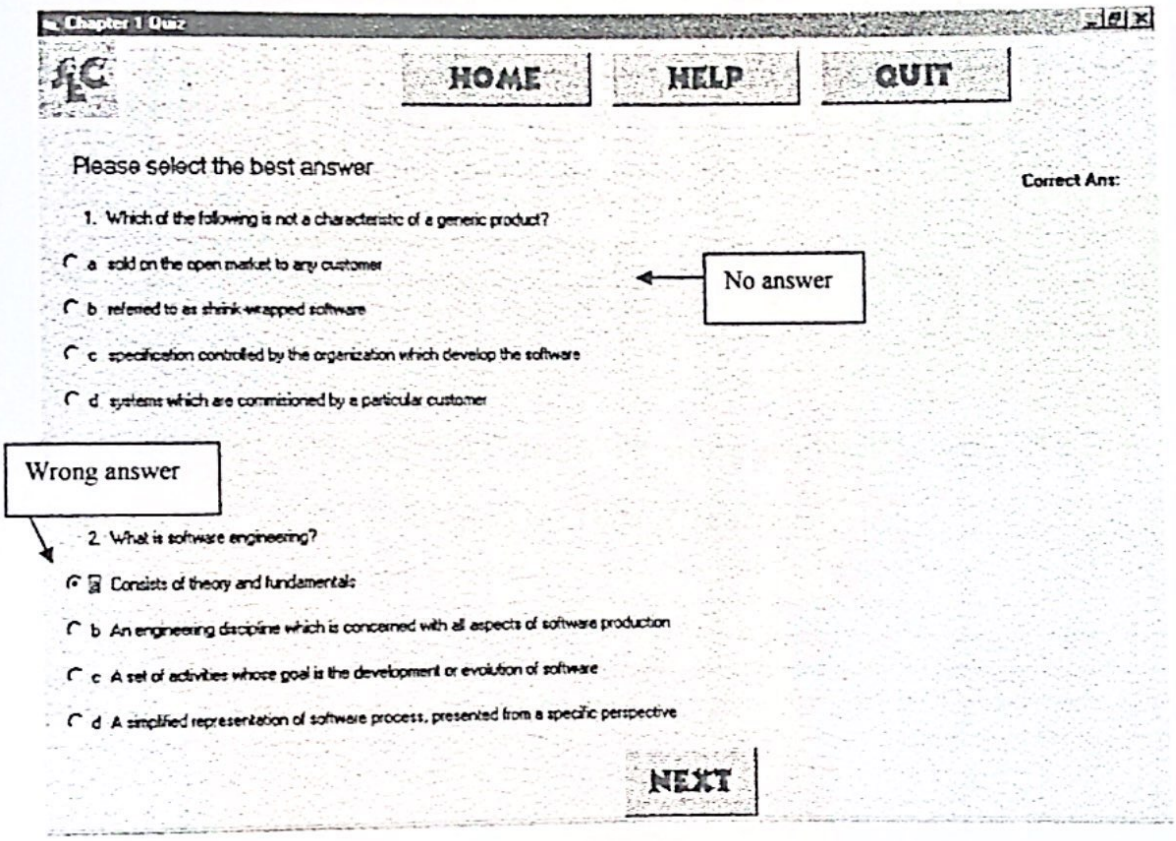


Figure 29 A wrong or empty answer is chosen.

Expected Response : No "tick" was displayed beside the question number.

Actual Response : No "tick" was displayed beside the question number.

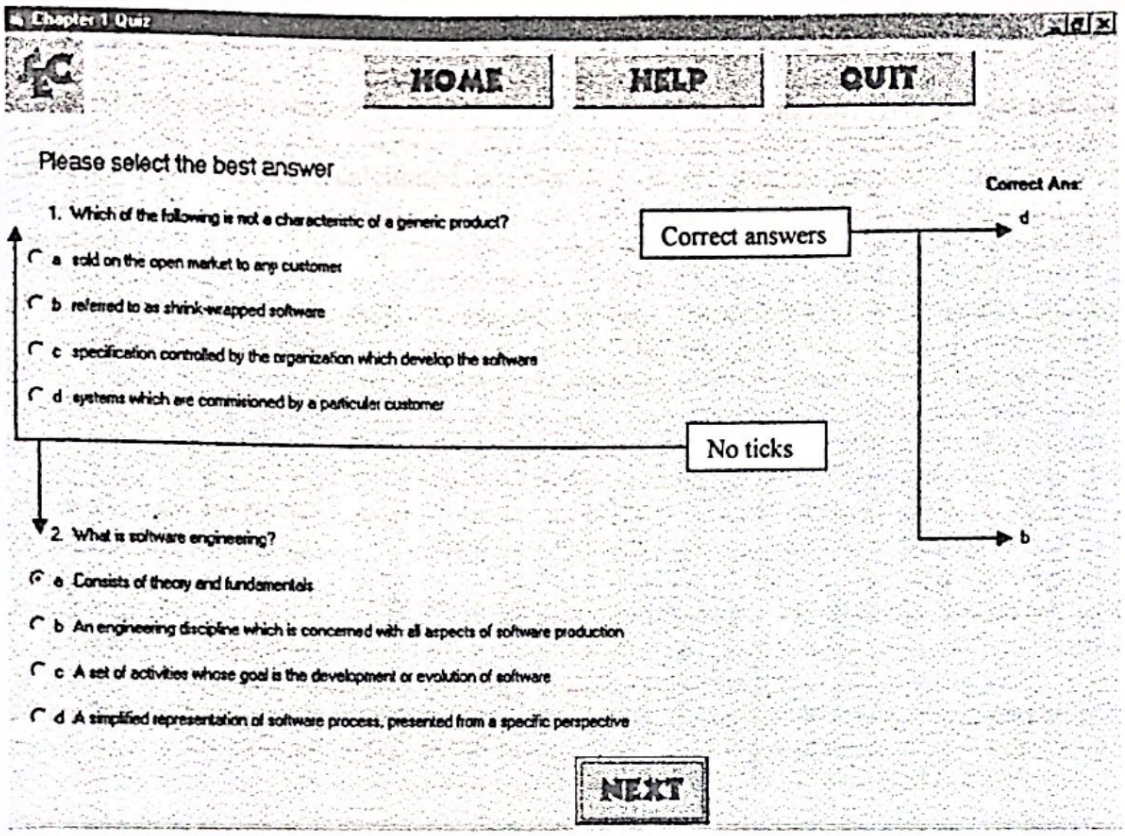


Figure 30 There are no "ticks" beside the wrong and blank answers. The correct answers (in red) are given at the right hand side of the screen.

Input Data No.2 : Only one correct answer.

Description : If one answer is correct out of six, the marks will be 16.67%. The marks were calculated as 1/6 x 100. It will then be displayed on the top of the screen.

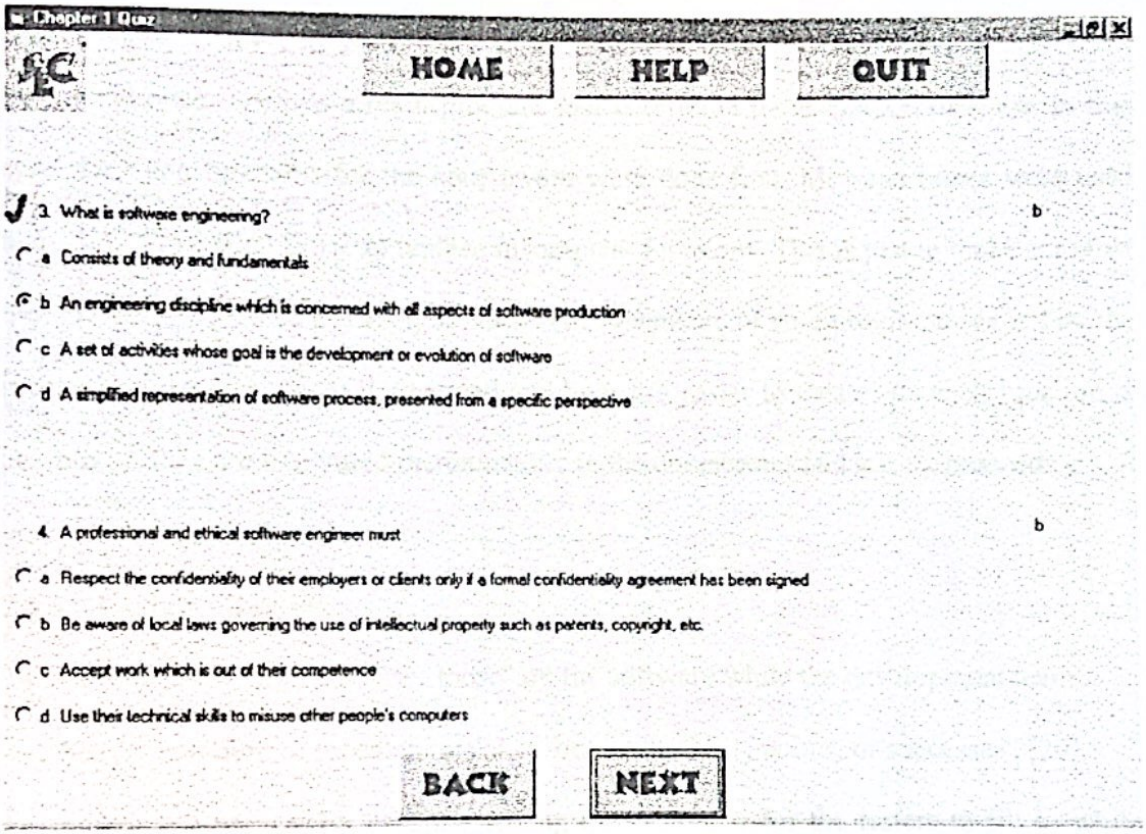


Figure 31 A correct answer with a tick.

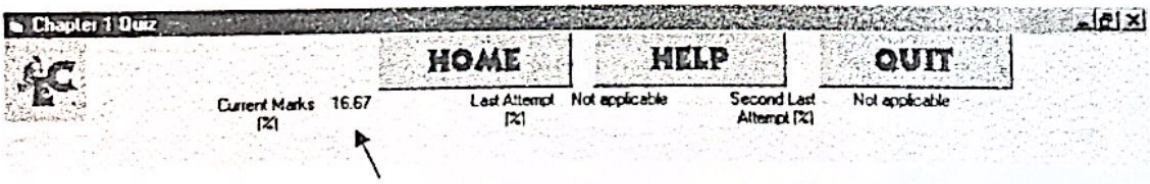


Figure 32 The marks 16.67 is displayed at the top of the screen.

7.1.2 Integration Testing

The entire system that comprised of integrated components is tested. This testing is done because although the courseware passes the component testing, the components might not interact well with one another. The author has chosen top-down testing for the courseware. Incomplete functions are made as stubs so as not to interfere with the completed ones. There is a main program that controlled the execution of stubs. In this case, the main functions for the courseware were done first. Message boxes were used many times in Visual Basic to replace incomplete functions. This is just to make sure that the functions are called at the right place. The advantage of stub programming is that the incomplete program can still be tested and bugs are easier to locate. As each function is completed, they are integrated incrementally to the courseware and testing proceeds.

7.1.3 Usability Testing

"A usability study "turns the user loose" on the software while the development team watches for common or repeated mistakes, bugs, stumbling points, or successes" [20].

The author has prepared a usability testing questionnaire for the testers to fill in while they test the system. To date, the author has asked a user to perform the test on the system. The answer to the questionnaire is on the following page.

Software Engineering Courseware

Usability Testing Questionnaire

Note: This questionnaire is adapted from http://best.me.berkeley.edu/~pps/pps/survey.doc

---

Dear Testers:

Thank you for your participation in the usability testing of **Software Engineering Courseware.** This project is aimed to provide students a multimedia way of studying the subject to maintain interest and understanding. The purpose of this questionnaire is to get your feedback on the user interface design, navigation schemes and also the quality of the lessons. In addition, you can report any malfunctions, inconsistencies, or confusions to the author via this questionnaire. The result of this questionnaire is important as changes may be made to the system to further improve it.

Thanks again for your time and efforts!

Consistency

1. Does this courseware have a consistent, clearly recognizable "look-&-feel"?

   Yes. Moderate.

2. Were you able to quickly locate the information you wanted according to the links the author provided?

   No. In Chapter 2's lesson, the link **click for example** is not meaningful enough for me to deduce its meaning. It should be changed to **click here for example** instead.

3. Please rate the consistency within this courseware

   ___Very consistent ___average ___poorly consistent

   It is average.

4. Please comment on the overall consistency in this courseware.

   Average.

Navigation

1. Were you lost or disoriented at any point while navigating this courseware? If so, please specify the page(s).

   Yes. I was lost in Chapter 2's lessons when I encounter the **click for example's** link. I am unsure of where to click.

2. Was the courseware easy or difficult to navigate? Where did you find most difficult?

   Same as above.

3. Are the links obvious in their intent and destination?

   No. the wordings of the links have to be changed, like mentioned in Question 1 of this part of questionnaire.

4. Is there a convenient, obvious way to maneuver among related pages and between different sections?

   Yes, except for the click for examples link.

5. Did you click on something where nothing happened? If so, what was it?

   Yes, there were some broken links to quizzes.

6. How soon could you get familiar with the navigation schemes in this courseware? Did you feel confident and comfortable after you got used to it?

   Quite soon. Yes, I feel comfortable after I got used to it.

7. Do you like the colors of the courseware?

   Yes.

8. How would you improve the presentation of the material?

   In the lessons part where there were roll over effects that I did not know of. I suggest that some instructions be written in the courseware to tell the user about this. The diagram of the process models in Chapter 3 should be made larger. The presentation can be made interesting by adding more graphics and animation.

9. Please comment on the overall navigation schemes of this courseware.

   Overall it is alright, except for some non-obvious and broken links.

Quality of Lessons

1. Please rate the quality of the lessons provided by this courseware

   ___Very good ___average ___very poor

   Average

Other suggestions/concerns

1. Do you have any other suggestions/concerns?

   In the Welcoming page, I am unsure of whether I could start when the Flash movie is played. The software should indicate to the user that they can skip the movie if they want to. In the quiz part, I would like to see whether my answer is correct before I proceed to the next one. In the test part, there should be some instructions to ask user to fill in the blanks.

7.2 Discussion on the Outcome

Using the result of the test data and top-down testing, the author is able to improve the system to perform validation on wrong user name or password, and empty user name or password. The top-down testing with stubs has shown some bugs to the courseware where the author could correct them. Some of the functions can be used by both the quiz and the test form, while some cannot. The testing has uncovered some errors because the author has made a mistake by assuming that the quiz and test can call the same function. There need to be some differences in the functions. So the author then rectified this problem by creating different functions for both the quiz and test.

The usability testing also has opened the author's eyes. Testing was never sufficient if it was to be done by only one party. Sometimes we just could not notice the mistake we made, especially since we were the programmers of a program. By asking a test subject to perform testing on the courseware, she has found some loopholes like non-obvious

buttons and broken links. The author has solved these problems, as well as other suggestions made by the test subject.

CHAPTER 8

CONCLUSION

8.1 Progress / Outcome of Project 1

After going through the project selection, planning, implementation, testing and finally the authoring (thesis-writing) phases, the writer is indeed very glad to have completed this thesis. Gone are the many sleepless nights and anxiety. Although the author does not claim this thesis to be perfect, every effort has been put into the accomplishment of it and the author felt very much indebted to individuals who have contributed in one way or the other.

8.2 Problems Encountered

The process of doing the courseware and writing this thesis is not at all smooth. The writer has encountered some problems along the way and they are as follows:

8.2.1 Time Limitation

Good planning is the key of a good project implementation. One major problem of writing this thesis is time limitation. The author is hard-pressed for time most of the time.

There is something where the author wishes to express her regret: The author has a lot of ideas to further improve the courseware but she is unable to do so because of this problem. There are good points, though. Looking from a different angle, the time constraints pushed the author to work harder and be able to multitask!

8.2.2 Knowledge Insufficiency

The author felt that she still lack knowledge Visual Basic programming language, Flash and Swish. The author has to learn how to use the software and also code the courseware at the same time.

8.3 Results and Achievement

The courseware is accomplished on time and it is able to overcome the weaknesses of the current system. The target for Project 1 has been achieved. However, due to time limitation and insufficient knowledge, the system could not be enhanced to include additional functionalities.

CHAPTER 9

EVOLUTION

9.1 Suggestions for Further Improvement

Talking about quizzes and test, more questions can be put into the database so that the probability of the user getting similar questions is small.

The layout of the courseware can be improved by adding "skins" to it, rather than relying on the basic controls that Visual Basic provide.

There should also be updates where the students can simply download from the web and this would automatically update the courseware. Or perhaps the courseware could be made web-based.

Ideally, the courseware should have an Artificial Intelligence (AI) way of teaching. The courseware would be able to "learn" the learning styles of students so that it can teach the students according to their weaknesses.

Last but not least, it has been a fruitful experience to go through the final year project. Although the process is not easy, the author has learnt project management (and effective time management!) and she considered these as a learning experience.

REFERENCES

[1] Wade, Pamela A. 1995, *Producing High-Impact Learning Tools*, London: Kogan Page Ltd.

[2] Sommerville, Ian. 2001, *Software Engineering.* England: Pearson Education Limited

[3] Pressman, Roger S. 2001, Software Engineering A Practitioner's Approach. ,Mcgraw-Hill

[4] Telecommunications Management Project Management. 1992, The National Computer Centre Limited, Hobbs the Printers of Southampton.

[5] Boyle, Tom. 1997, Design for Multimedia Learning. Prentice Hall

[6] Schneider, David I. 1999, An Introduction to Programming Using Visual Basic 6.0 Fourth Edition, Prentice Hall

[7] Kendall, Kenneth E., Kendall, Julie E. 1999, Systems Analysis and Design Fourth Edition, Prentice Hall

[8] Connolly, T.M. 1999, Database Systems A Practical Approach to Design, Implementation and Management 2nd Edition, Addison-Wesley

[9] Clarke, Alan 2001, Designing Computer-Based Learning Materials, Gower

[10] A Cognitive Theory of Multimedia Learning: Implications for Design Principles
Richard E. Mayer and Roxana Moreno
University of California, Santa Barbara
http://www.eng.auburn.edu/csse/research/research_groups/vi3rg/ws/mayer.rtf
accessed : 1.9.03

[11] Bostock, S. February 1996 (revised March 1998)

CoursewareEngineering
- an overview of the courseware development process

Internet: http://www.keele.ac.uk/depts/cs/Stephen_Bostock/docs/atceng.htm

[12] Merger, B. 1998,
Instructional Design & Learning Theory. May 1998
Internet: http://www.usask.ca/education/coursework/802papers/mergel/mergel.PDF

[13] McGriff, S.J.

Instructional System Design (ISD): Using the ADDIE Model. 2003 Internet: http://www.seas.gwu.edu/~sbraxton/ISD/general_phases.html as of 23.9.2000

[14] Arts in Education Institute of Western New York. 2002
Internet: http://www.artsined.com/teachingarts/Pedag/Dewey.html (date: 5.8.2003)

[15] Introduction to Instructional Design and the ADDIE Model
by Kevin Kruse
Internet http://www.e-learningguru.com/articles/art2_1.htm (date: 10.9.2003)

[16] Central Queensland University
Faculty of Informatics and Communication
Internet:http://infocom.cqu.edu.au/Courses/spr2000/95169/Extra_Examples/ERD.htm

[17] HyperDictionary
Internet: http://www.hyperdictionary.com/computing/courseware

[18] Journal of Courseware Engineering
Internet: http://www.ifi.uib.no/cce/jce.html

[19] Schneider, David I. 1999, An Introduction to Programming using Visual Basic 6.0
Fourth Edition, New Jersey: Prentice-Hall , Inc.

[20] TeachIT Spring 2000
Internet: http://www.citl.ohiou.edu/TeachIT

[21] searchVB.com
Internet: http://searchvb.techtarget.com

[22] Planet Source Code
Internet: http://www.planet-source-code.com/URLSEO/vb/scripts/ShowCode!asp/txtCodeId!40685/lngWId!1/anyname.htm

APPENDICES

Appendix 1

INTERVIEW QUESTIONS FOR SOFTWARE ENGINEERING LECTURERS
Interviewee: Assoc. Prof. Jamilin Jais

Note: Below are not the direct answers quoted from the interviewees. The are instead rewritten and/or rephrased by the author.

1. What are your techniques / approaches in teaching SE?

2. How do you find teaching Software Engineering (SE)?
   Is it easy? Do you have any problems? Do you need some teaching aid?

   It depends. If I am prepared, it is easy. However, it might be hard for a beginner to teach Software Engineering.

3. Do you get full cooperation from students in class?
   Are they responsive?
   Do the majority of them do the assignments given?

   Mostly not. Yes, the majority of students do the assignments given. If they don't, no marks will be given.

4. Do you think students grasp SE concepts easily? Why and why not?

   Only sometimes. If they are not in a good mood, feel sleepy or hungry in class, I have to keep on calling their names (to make them respond to me).

5. What about the students' progress?
   How are their quiz, test, project and exam scores?

   It depends. If they have missed a quiz they will complain.

6. In your opinion, what are the problems students usually face?

   It depends. If they are hardworking, they will get good results. Some students are enjoying themselves in the university - they are rich and have cars.

7. Does the Computer Science & IT Department in Uniten currently utilizes any courseware to teach SE? If yes, can you please elaborate? Do you think the current system should be enhanced to be a better one? Why and why not?

   No. I am using books to teach Software Engineering. I have about ten of them.

8. Have you ever used / came across any SE courseware? What are the advantages / disadvantages of the courseware?

I downloaded from the Internet.

9. Are there any CD-ROM that accompanies the SE textbook? If yes, do you recommend your students to try it out? How do you find the content of the CD-ROM?

   No. It has a website instead.

10. Do you recommend students to try out additional SE courseware in the Internet? If yes, please state the websites' URL.

11. Do you refer to any websites or electronic media in teaching SE?

12. From a lecturer's perspective, what is your opinion on computerized supplementary educational tool?

    It is good because it helps us a lot. We don't have to teach. We just ask students to study – just like Universiti Tun Abdul Razak (UNITAR), students only meet their lecturers once a month.

13. If a courseware in SE is being developed as a teaching aid, would you use it? If yes, what would you like its content to have and what are your expectations of the software?

    The courseware must follow the syllabus. Animations, video and PowerPoint slides can be provided.

14. Do you enjoy teaching SE? Why and why not?

    Yes. I enjoy teaching. I will try my best in teaching whatever subjects.

15. Any other suggestions for the courseware?

Appendix 2

INTERVIEW QUESTIONS FOR SOFTWARE ENGINEERING LECTURERS
Interviewee: En. Nazri Abdullah

1. What are your techniques / approaches in teaching SE?

   I give normal lectures.

2. How do you find teaching Software Engineering (SE)?
   Is it easy? Do you have any problems? Do you need some teaching aid?

   It is easy. Yes, I need some teaching aid.

3. Do you get full cooperation from students in class?
   Are they responsive?
   Do the majority of them do the assignments given?

   Yes. Not all students, around 50-50. Yes, the majority do the assignments given.

4. Do you think students grasp SE concepts easily? Why and why not?

   Yes because SE is more on technical and theoritical knowledge method of software development. Therefore it is not hard for students. SE is just a knowledge.

5. What about the students' progress?
   How are their quiz, test, project and exam scores?

   Good.

6. In your opinion, what are the problems students usually face?

   The have problems in their assignments – they don't have ideas on documentation.

7. Does the Computer Science & IT Department in Uniten currently utilizes any courseware to teach SE? If yes, can you please elaborate? Do you think the current system should be enhanced to be a better one? Why and why not?

   No, only case tools.

8. Have you ever used / came across any SE courseware? What are the advantages / disadvantages of the courseware?

   No.

9. Are there any CD-ROM that accompanies the SE textbook? If yes, do you recommend your students to try it out? How do you find the content of the CD-ROM?

No.

10. Do you recommend students to try out additional SE courseware in the Internet? If yes, please state the websites' URL.

Internet links for references (this is provided in En. Nazri's website). The links are interactive and students can get more knowledge.

11. Do you refer to any websites or electronic media in teaching SE?

Mostly books.

12. From a lecturer's perspective, what is your opinion on computerized supplementary educational tool?

I agree (on computerized supplementary educational tool).

13. If a courseware in SE is being developed as a teaching aid, would you use it? If yes, what would you like its content to have and what are your expectations of the software?

Yes. I expect the courseware to be more on technical method. They should have diagrams, use cases (diagram from Rational Rose). The diagrams should be linked. The courseware should also contain certain examples on project development, sample on documentation and actual works on software development (medium size).

14. Do you enjoy teaching SE? Why and why not?

Yes because SE is my field of study in the degree level. From experience, I was a software developer and I am able to apply practical knowledge (in teaching).

15. Any other suggestions for the courseware?

Please concentrate on contents and attractiveness of elements – visual (pictures, diagrams).

Appendix 3

INTERVIEW QUESTIONS FOR SOFTWARE ENGINEERING LECTURERS

Interviewee: En. Yunus Yusoff

The author went to interview En. Yunus Yusoff regarding this project to find out that the latter is not teaching Software Engineering currently. The interview could not proceed as all the questions prepared are meant for a Software Engineering lecturer. Nevertheless, a discussion ensued where En.Yunus gave the author very useful advices and suggestions, which are summarised shortly.

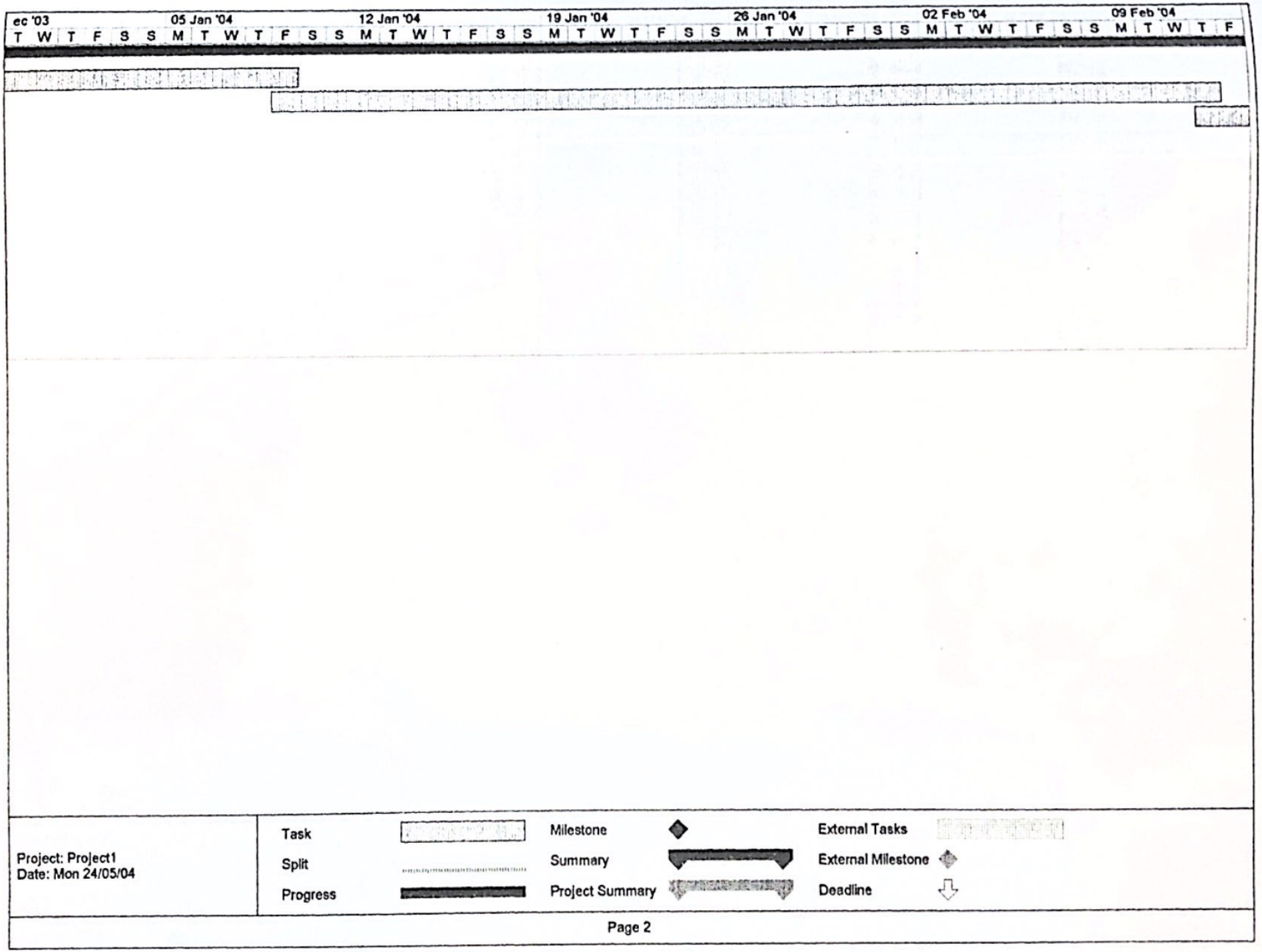
En. Yunus gave some pointers on preparing interview questions. He said that the purpose of interview is to solicit information from interviewees. Hence, right questions should be ask to the right persons, which is why he had suggested to the author to interview students besides lecturers because they are the intended users. The "right questions" means questions that are specific and straight to the point. For example, the author should not ask "Do you think a software is needed in learning SE?" because the author is going to develop the software anyway, no matter what the students' responses are. En Yunus stressed that interview questions should encompass the subject (in this case, courseware development) itself.

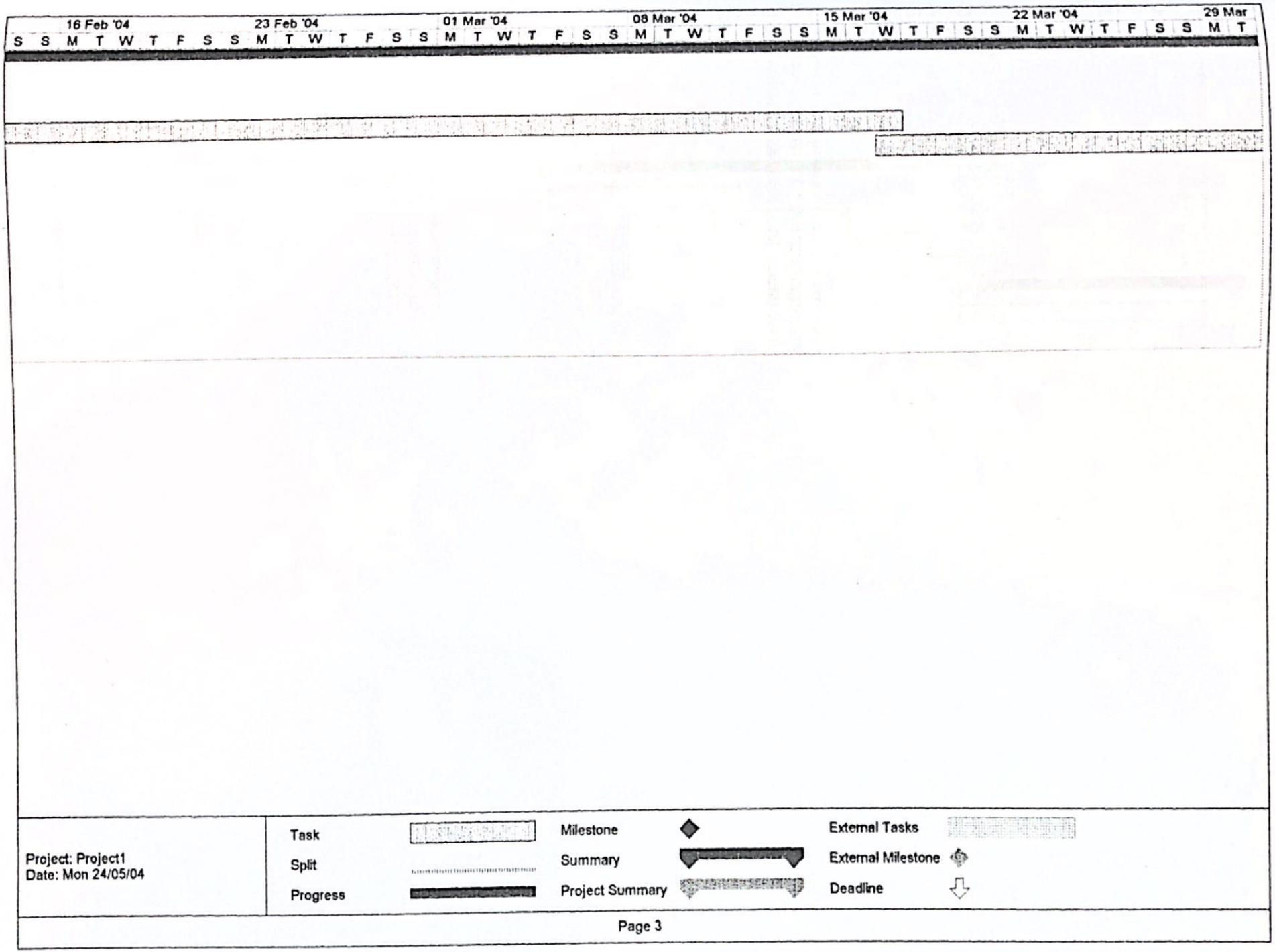
When En. Yunus and the author discussed about the difficulty of obtaining a Software Engineering courseware, the latter mentioned about the availability of a courseware of a different subject – Management Information Systems (MIS). En. Yunos said that the author can actually evaluate the MIS couseware since the nature of MIS is similar to that of SE.

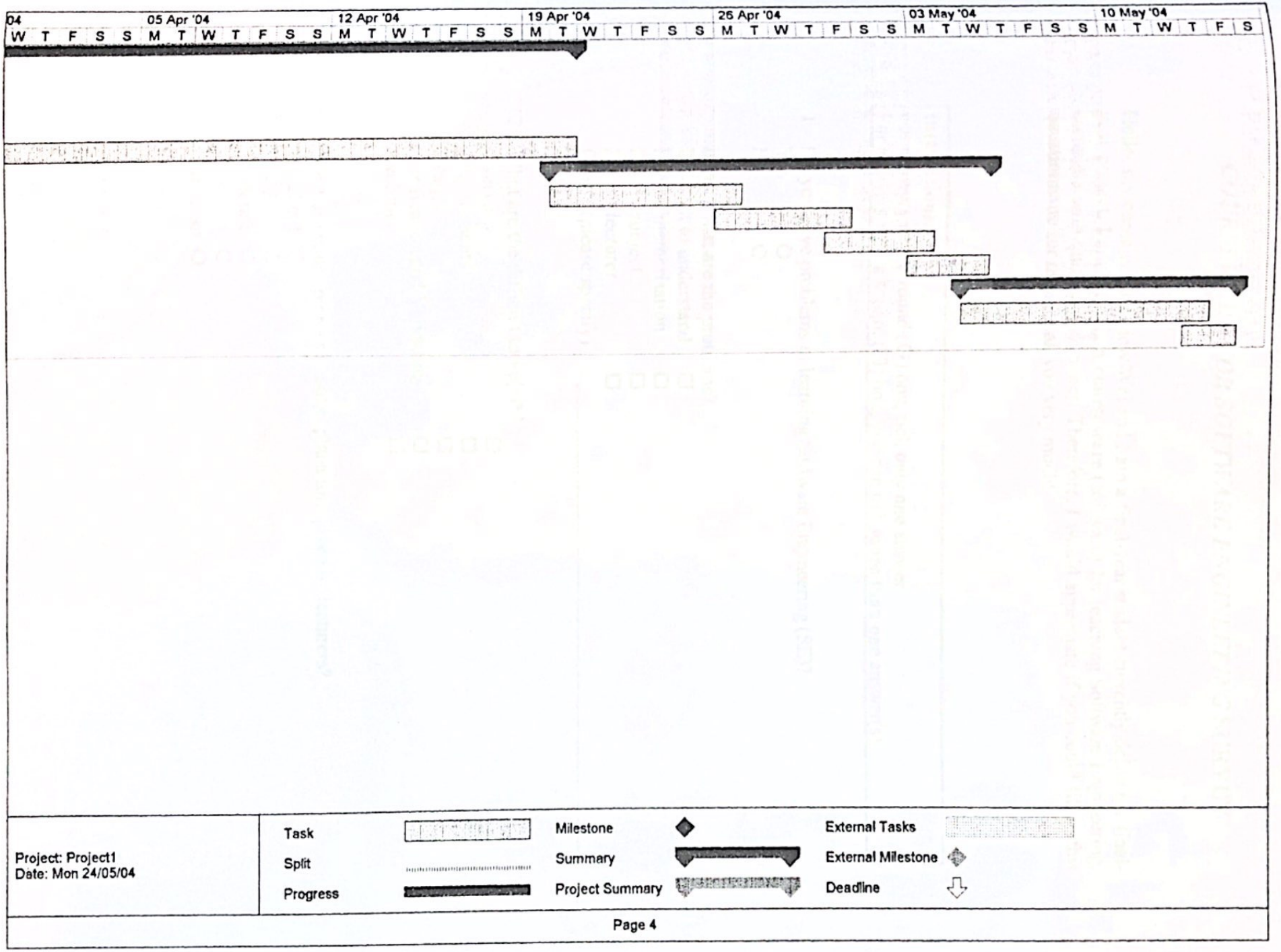
When the author asked En. Yunus whether he has used any courseware, the latter replied whether a PowerPoint presentation is considered a courseware. The author answered "no" as a courseware should provide interactivity. En. Yunus said further that a PowerPoint presentation could also provide that with the use of scripting. The interviewee later explained that the purpose of asking the author the question is to let the author think what actually constitutes a courseware - the author needs to define the meaning of "courseware".

Appendix 4

ID	Task Name	Duration	Start	Finish
1	**Implementation**	**97 days**	**Mon 08/12/03**	**Tue 20/04/04**
2	Lesson Animation	25 days	Mon 08/12/03	Fri 09/01/04
3	Flash Files	25 days	Fri 09/01/04	Thu 12/02/04
4	Codification	25 days	Thu 12/02/04	Wed 17/03/04
5	Visual Basic Files	25 days	Wed 17/03/04	Tue 20/04/04
6	**Integration and System Testir**	**12 days**	**Tue 20/04/04**	**Wed 05/05/04**
7	Unit Testing	5 days	Tue 20/04/04	Mon 26/04/04
8	System Testing	5 days	Mon 26/04/04	Fri 30/04/04
9	User Feedback	2 days	Fri 30/04/04	Mon 03/05/04
10	System Enhancement	3 days	Mon 03/05/04	Wed 05/05/04
11	**Thesis**	**8 days**	**Wed 05/05/04**	**Fri 14/05/04**
12	Thesis Writing	7 days	Wed 05/05/04	Thu 13/05/04
13	Compilation	2 days	Thu 13/05/04	Fri 14/05/04







Appendix 5

COURSEWARE FOR SOFTWARE ENGINEERING SURVEY

Hello, my name is Low Siew Wei and I am a final-year student currently doing my final-year project. I am creating a **courseware** (software) for teaching Software Engineering, complete with quizzes and games. Therefore, I would appreciate if you could fill in this questionnaire for me. Thank you very much!

Instructions:
For answers with a round (O) box, tick only **one** answer.
For answers with a square (☐) box, you can tick **more than one** answer(s).

1. Do you have problems in learning Software Engineering (SE)?
   Yes O
   No O

   If **yes**, what are the problems?
   Difficult to understand ☐
   Lots of memorisation ☐
   Boring subject ☐
   Boring lecturer ☐
   Others (please specify) : _____

2. What are the difficult topic(s)*?
   Overview ☐
   Requirements ☐
   Design ☐
   Verification and Validation ☐
   Evolution ☐

3. What are your opinions about explanations given by lecturers?
   Excellent O
   Good O
   Moderate O
   Poor O
   Very poor O

*With reference to the SE textbook.

4. How do think the lecturers can improve on their teaching technique(s)?
   Explaining concepts step-by-step ☐
   Give more examples ☐
   Give more exercises ☐
   Make the lesson interesting ☐
   Others (please specify): _____

5. Which learning method do you prefer?
   Self-study with **no** guidance from lecturers ○
   Self-study with guidance from lecturers ○
   Go for lectures **without** self-study. ○

6. Do you refer to any website(s) or electronic media in learning SE?
   Yes ○
   No ○

   If **yes**, please state their URL(s):

   _____
   _____
   _____

7. Have you ever used any SE courseware?
   Yes ○
   No ○

   If **yes**, what are the advantages / disadvantages of the courseware?

   _____
   _____
   _____

8. Do you think that having a **courseware** would help you learn SE more effectively?
   Yes ○
   No ○

   If **yes**, what are your suggestions for the contents of the courseware?
   Lessons ☐
   Multimedia elements – sound, video, animation and interaction ☐
   Quizzes ☐
   Tests ☐
   Assignments ☐
   Links to Internet sources ☐
   Games ☐
   Others (please specify) : _____

   If **no**, please give suggestion(s) on a better learning technique for SE.
   _____
   _____
   _____

9. Would you like the courseware to have a teaching assistant (something like the Microsoft Office Assistant)?
   Yes ○
   No ○

10. Do you have any other comments/concerns regarding the learning of SE / SE courseware?
    Please elaborate if you have.
    _____
    _____
    _____
    _____
    _____