User Interface Design For Higher Education Usage

By

Iryani binti Lob

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Technology)

JUNE 2004

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

User Interface Design For Higher Education Usage

By

Iryani binti Lob

A project final draft submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

Approved by,

_____

(Miss Eliza Mazmee Mazlan)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and the original work contained herein have not been undertake of done by unspecified sources or persons.

IRYANI BINTI LOB

ABSTRACT

User interface is a part of computer and its software that people (user) see, hear, touch or talk to. It is a set of all things that allow you and your computer to communicate with each other. A good interface design is crucial to make a software/application/system a success.

In this project the author concentrate on development of user interface design for higher educational usage, specifically UTP e-learning system interface. This project being develop to overcome several problems face by user such as making lots of frequent errors due to misunderstanding or confusion, not using or realize some of feature or functions, and inappropriate user interface or interaction .

The objectives of the project are to develop a good interface design for e-learning system for intuitive usage, create a didactic environment and also to design interface that require minimal page-through to complete task(s).

The methodology of the project requires several steps which are planning where preliminary investigation on the project area and project scheduling are performed, analysing where analysis being perform to understand the existing system interface function, data collection; to gather information on user preference and interface development, designing and testing that involve user acceptance testing.

The result and discussion will be analysing of the previous questionnaire that had been distributed to the user in order to get the user requirements and expectation towards the new interface developed. Beside there the author also perform analysis in the design area where the author stated a few assumptions made that will contribute top the development of the new interfaces.

ACKNOWLEDGEMENT

In the name of God, the Compassionate, the Merciful, the author would like to thank Him for the strength, skill, knowledge, patience and health that has been given through this time. It is an honour for the author to acknowledge the effort and assistance that has been shown by many people whose name is not mention on the cover. Without them, producing this report would have been impossible.

First of all, a lot of thanks to the most supportive and encouraging supervisor, Miss Eliza Mazmee Mazlan, who had given all the details and guidance in completing this final year project. Despite her busy schedule, there will always be time for her to meet and discuss arising matters.

I also would to thank all my colleagues and other final year students of Information Technology and Information System for the moral supports and sharing of information in striving through the semester completing the final year project.

Not to forget, thanks to all the respondents in the survey for the time and effort to answer the questionnaire and comments. Thank you for all the cooperation.

Last but not least, to those whose name is not mentioned, you have directly or indirectly contribute to this project. The author would like to thank you very much.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background

Poor user interface design can hide even the most powerful and useful website/system from all but the most advance and patient users. A poor user interface will result a low usage of the site, not been able to serve the users and it is an ultimate failure.

The failure of website/system is often due to their designers not considering their users and designing with the assumption of too much technical knowledge. In education area user interface problems happen in term of interface design for institution education website or system such as e-learning system-tutor or other student learning support mechanism

1.2 Problem Statement

1.2.1 Problem Identification

This project will concentrate on the e-learning system in higher education institution specifically "UTP E-Learning System". From observation a few problems have been frequently encountered by students such as:

- making lots of frequent errors due to misunderstanding or confusion
- not using or realizing some of the features or functions
- inappropriate user interface or interaction

All these problems will be taken into consideration in this project.

1.2.2 Significant of the project

The main purpose of developing this project will be to meet the requirements for final year subject which is Project. Beside that it is also consider as a little contribution from the author toward the institution (Universiti Teknologi PETRONAS) where the author develop a project to enhance the existing UTP e-learning system interface for the benefits of UTPs' student and UTP indirectly.

1.3 The Objectives

The objectives of this project are to develop a good user interface design for UTP e-learning system that will be considering aspect such as:

- To design the interface for intuitive usage
  The interface that will be developed gives users the immediate understanding of the interface without conscious reasoning or study.

- To create a didactic environment for learner
  The interface will provide a learning environment for the user. User will know and understand that such interfaces developed are intended for learning purposes.

- To design a user interface that require minimal page through to complete task.
  The interface will tried to eliminate any unnecessary pages as used in the existing UTP e-learning interface in order to reduce page through in completing certain task.

1.4 Scope of Studies

1.4.1 Relevancy of the project

Studies on Human Computer Interaction will be perform by focusing on how to develop a good interface design base on design principle, specifically interface design for e-learning system. An analysis on the existing e–learning system will be perform in order to develop a good interface design that will address any weaknesses or problem that exist in present system in order to meet user requirements. This project will not be covering the functionalities aspect of Learning Management System (LMS) and independent of any LMS in the market

1.4.2 Feasibility of the project.

1.4.2.1 Study

This project required the author to perform a study on Human Computer Interaction area. The author has always been interested in the subject. Therefore it is a bonus to the author.

1.4.2.2 Time

The project that will be developed required a suitable amount of time as to the time that already provided for this subject. Theoretically this project can be completed successfully within the 5 months provided by the university.

1.4.2.3 Economic

The project that will be developed required a very low amount of cost that is just appropriate to the author economic capability.

1.4.2.4 Future Implementation

The author will develop this project in a systematic documentation so that future reviews on the documentation for further enhancement. The author only concentrates on a small portion of the e-learning system for the project with a hope that this project and research can be used to improve the UTP e-learning system as a whole.

CHAPTER 2
BACKGROUND AND LITERITURE REVIEW

2.1 Research Area Overview

Human computer interaction has been on widespread use for over a decade. According to Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale human computer interaction studies become very demanding because of the need to create a user friendly interface computer or system that also reduce tiredness and stress in users. User interface are the things that related closely to human computer interaction studies. In other word both are interrelated. Base on IBM website user interface are simply parts of a computer and its software that you (the computer *user*) see, hear, touch, or talk to. It's allowed and assists two way communications between human and the system. It also stated that the best interface design would be the one that we don't have to pay to much attention to which means that user will simply understand what need to be done and how to react with the interface.

User interface design plays an important role in the development of e-learning system. According to Anne Nicol form Human Interface Group Apple Computer Inc. interfaces act as coaches. Interface usually have more sign of intelligent than what we realize. As a learning tool, interfaces seem superior because it calls user's attentions systematically to each consideration that must be made in producing a finished document. In other words an effective and good interface actually able to grab user attentions to what is important and the task that need to be completed.

In order to design a good user interface that will fully utilize its functions we first need to understand the user; human. According to Kevin Krause in his articles entitle Designing e-Learning User Interface there is four important things that need to be consider in developing interface design for e-learning system that are way to

assist user memory, way to put user in control, way to produce logical and consistence screen design and also way to provide user guidance.

Although human brains still are vastly more powerful and intelligent than computers, computers are much better than people when it comes to remembering things. Much of the work done in human-computer interaction is focused purely on ways to reduce the load on the human user's memory. This is where the knowledge of human computer interaction area is needed in order to assist user memory. As for given user control of the navigation in the system an effective interface need to be produce because by putting the user in control of the program, or at the very least, lets them feel like they are in control. By doing so their anxieties are eased, confusion is minimized, and an environment conducive to learning is created

Consistent and logical screen design also help user greatly in using e-learning system. User will create expectations that the interface they see will be consistent. There is example given by Kevin Krause in his article entitle Designing e-Learning User Interface: Logical and Consistent Screen design where he said that imagine, for example, that you are walking down a long hallway in a building and you must pass through several doors. When you get to the first door you won't know whether to pull it or push it open, unless it is clearly marked. You grab the handle and pull but the door doesn't budge, so you push and it opens. When you get to the second door you notice that it seems identical to the door you just passed through. Perhaps unconsciously you push the door first this time, and again it opens. Here we could make a conclusion that user tend to act similarly to interface or object in the interface that have similar shape, color or specification as the previous one that they encountered.

2.2 Project Overview

This project will be the development of new enhance user interface design for UTP e-learning system. The reason behind this development would be to find a good solution to cater problems encountered by UTP students in using the e-learning system. The author strongly belief despite other problems of the system user interface plays an important role in the effectiveness of the e-learning usage. According to Kevin Krause from e-learning guru website the most neglect topic in the e-learning area will be the interaction between student and the system. When student complaining about the problems that they facing with the e-learning system the problem that they address to is usually regarding the confuse menus, the unclear buttons, or illogical links. Feeling lost, confused, or frustrated will become a barrier to effective learning and information retention.

The author than used this statement and refer to the current problems that face by the students in UTP. The problems face by students in UTP more or less similar to the statement produce by Kevin Krause.

The project develop will mostly tried to solve problems face by student and in the process the author also hope to be able to create a didactive environment for the student when using the system and also try to create the environment where users are given appropriate power and freedom to navigate through the system. This project also only going to revolve around the interface design of the UTP e-learning system without taking into consideration the functionalities aspect of Learning Management System (LMS) and independent of any LMS in the market.

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

3.1.1 Planning Phase

Through out the planning phase the author had perform a preliminary investigation on the project area to get information in depth of what will be developed. To do so the author had studied a few books on related issues to get a clear idea on the topic.

3.1.2 Analysis Phase

The author had performed analysis on the existing UTP e-learning system base on the design principle (*Refer to Appendix 3-1*) and principle of user interface design (*Refer to Appendix 3-2*). The analysis is done by preparing a checklist that covers all the design principle area. (*Refer to Appendix 3-3*).As for this purpose the author adapt the checklist prepared by Kevin Krause; the e-learning columnist for CLO, Chief Learning Officer and President of AXIOM Professional Health Learning. From the checklist result the author then detected several weaknesses of the existing system. The author used the finding as an input in developing the new interface design with the consideration that the changes are appropriate to the needs of the user. The author also develops a theoretical framework that is significant with the research part of the project. (*Refer to Appendix 3-5*)

Beside that the author also tried to understand the main function and the task flow interface of the system. In order to do so the author had observe and also develop a task flow interface of the system in order to see the UTP e-learning system in a bird view. (*Refer to Appendix 3-7*)

3.1.3 Data Collection 1 Phase

Since the project outcome really dependent on user requirement the author had perform a data collection targeting on the potential user of the system interface. To collect the data the author had develop questionnaires (*Refer to Appendix 3-4*) and distribute it to the user which is the UTP students.

Beside the questionnaire the author also conducted interview with users of the existing system in order to identify their problems and dissatisfactions. The data collection perform by the author had been able to achieve the objective of data collections that the author had stated earlier which are to get user opinions and preferences on the interface aspect of the system and to identify features required by the user.(*Refer to chapter 4*)

3.1.4 Design Phase (iterative)

The author had developed the storyboard of the interface design. It's started with a few sketches of the interface emphasis on the positioning and specifications of the interface design before proceed to the flow of the interface and navigation part. (*Refer to Appendix 3-6*). The first storyboard design by the author only concentrated on the positioning part of the interface elements such as buttons, icons, text boxes, and drop-down menu. After finished designing for the positioning part the author then proceed with another part of the interface design which is the colours for the elements in the interfaces. Then after that the author only done a few minor changes in the interface to enhance the look of the interface to make it better and also proceed with the navigation part of the interface prototype. As mention before this prototype is an evolutionary prototype that will experience changes through out the design and testing processes. The author personally didn't expect that this first design will be accepted fully by the user. The author assumed that the good interface will be produce after 2 to 3 time changes. The design was tested part by part. During the design process of the prototype the author had already perform the testing

because the author couldn't wait until the whole development completed and then proceed with testing because of the limited time in completing the whole project. The author had prepared test script and had chosen several users to test the interface part by part. The feedback from the user then was used to complete the development process.

3.1.5 Testing Phase (iterative)

During the testing phases a User Acceptance testing will be performed to get user feedback on the new interface design. The author will develop a test script for user to perform the testing. The test script is a guide and also a controlled mechanism for the testing process in order to get a standard result referring on targeted area from the user. The testing had been performed in a parallel way as the design phase in order to make sure that the design will be completed on time. This is due to the fact that too many changes will effect the product completion date and also the limited time that provided for this project.

3.1.6 Data Collection 2 Phase (further data collection)

This second data collection phase is needed in the further development of the interface design. Since the interface prototype is an evolutionary prototype, further data collection is needed to help the development of the prototype. The test script result form the testing phase was analysis by the author to identify the weaknesses of the prototype and used the new finding in order to improve it.

3.1.7 Testing (final)

A final test was conducted in order to get final feedback on the interface design after the prototype had been completed. The author had chosen a few users to test the complete interface design. The result of the testing will prove whether this interface is acceptable by the user and if there is any improvement or changes needed by the user then it will be the inputs for further enhancement.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Data Gathering and Analysis

The author had obtained data from primary and secondary resources. The primary data refer to information obtained firsthand by the author from target user while the secondary resources are obtained from existing record. For this project the primary source of data comes from data collected from questionnaires distributed to target users and the secondary data collected from observation and journal/internet browsing.

4.2 Questionnaires

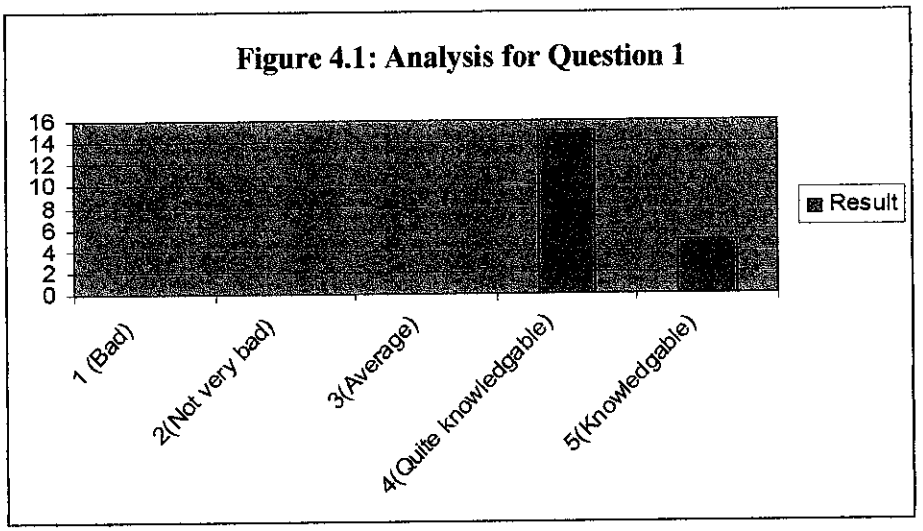
The author had distributed a set of questionnaires in order to get user feedback on the existing UTP e-learning interface as well as the new one. The author chooses to use a group of user that consists of 20 users from various courses. This is because the author strongly belief that user from same course tend to have the same perception on e-learning. In order to cater requirements and need from UTP students that comes from different course and different need the author had make that decision. The questionnaires consist of 10 close ended questions and 1 open ended question.

4.2.1 Analysis for Question 1

The objective of this question is to know the level of computer competent among UTP student. The significant of this questions with the project will be if the user have a good level of computer competent then user will be able to handle any system specifically UTP e-learning interface at the maximum level of use. The results of the question are.

Answer Rating	Result
1 (Bad)	0
2(Not very bad)	0
3(Average)	0
4(Quite knowledgeable)	15
5(Knowledgeable)	5

Table 4.1 Result for Question 1



The result indicate that UTP student have a good level of computer competent. This is due to the fact that most of UTP student own a personal workstation in hostel and also the ratio computer terminals in UTP to number of UTP student is very low. This will make every individual have a good change to used computer in an unlimited time and the resources.

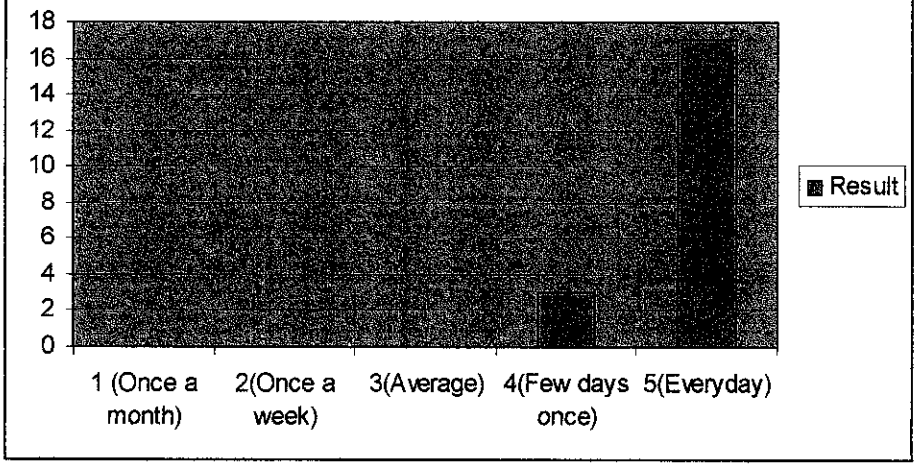
4.2.2 Analysis for Question 2

Question two referring to the how often user usually use the UTP e-learning system. The results are as below.

Answer Rating	Result
1 (Bad)	0
2(Not very bad)	0
3(Average)	0
4(Quite knowledgeable)	3
5(Knowledgeable)	17

Table 4.2 Result for Question 2

Figure 4.2: Analysis for Question 2



The result indicates that most of the user uses the e-learning system everyday. Among their reason is because that they have to get lecture notes from the system. This means users spend a lot of time using and navigation the UTP e-learning interface.

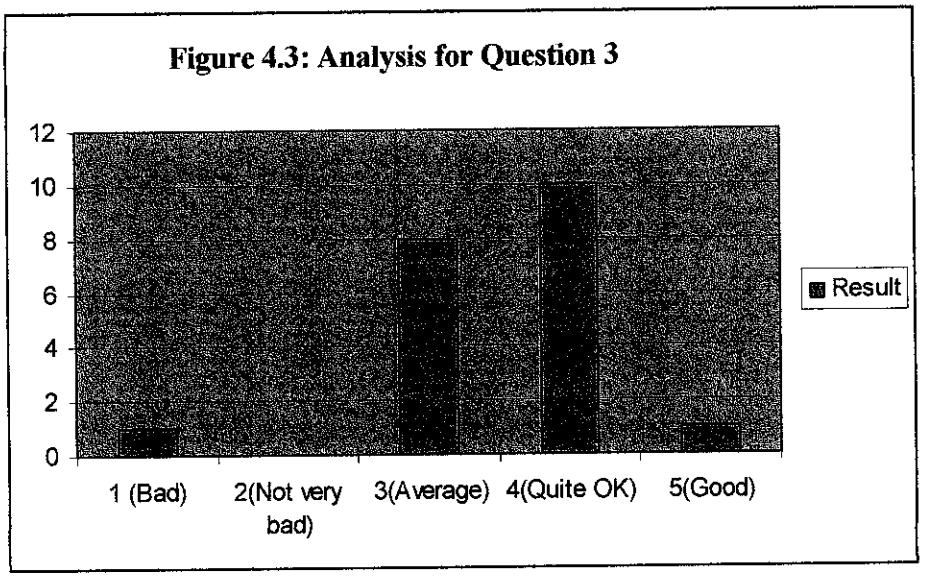
4.2.3 Analysis for Question 3

Question 3 means to get user feedback on what they think the level of the existing UTP e-learning interface. Whether they think it is good enough or bad.

Answer Rating	Result
1 (Bad)	1
2(Not very bad)	0
3(Average)	8
4(Quite OK)	10
5(Good)	1

Table 4.3 Result for Question 3

Figure 4.3: Analysis for Question 3



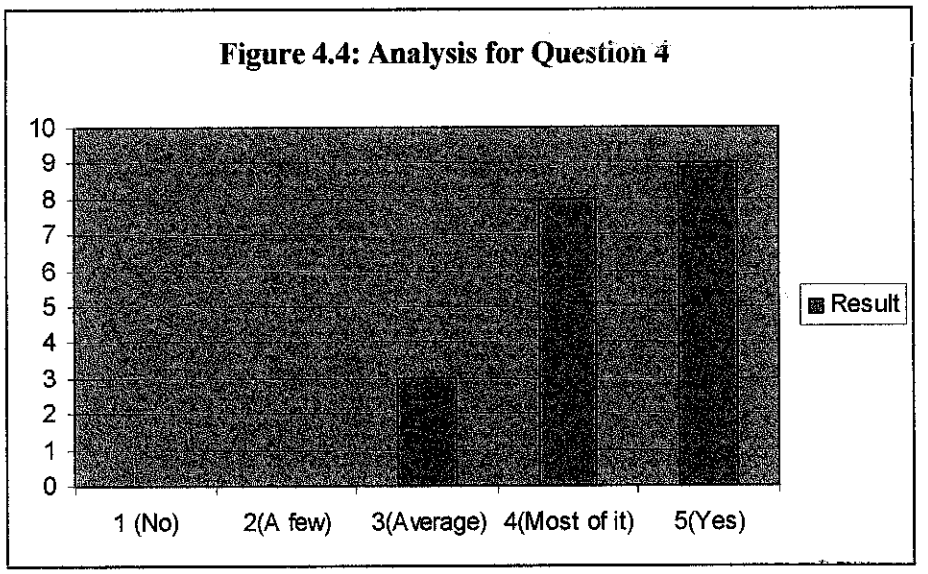
The result for this question shows that user think or percept that the existing user interface is not good enough. There even some user that thinks that the interface is bad, but mostly thinks that it is just OK and quite OK. There still room for improvement.

4.2.4 Analysis for Question 4

In this question the author tried to get user feedback whether the existing user interface design require changes or not.

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	3
4(Most of it)	8
5(Yes)	9

Table 4.4 Result for Question 4



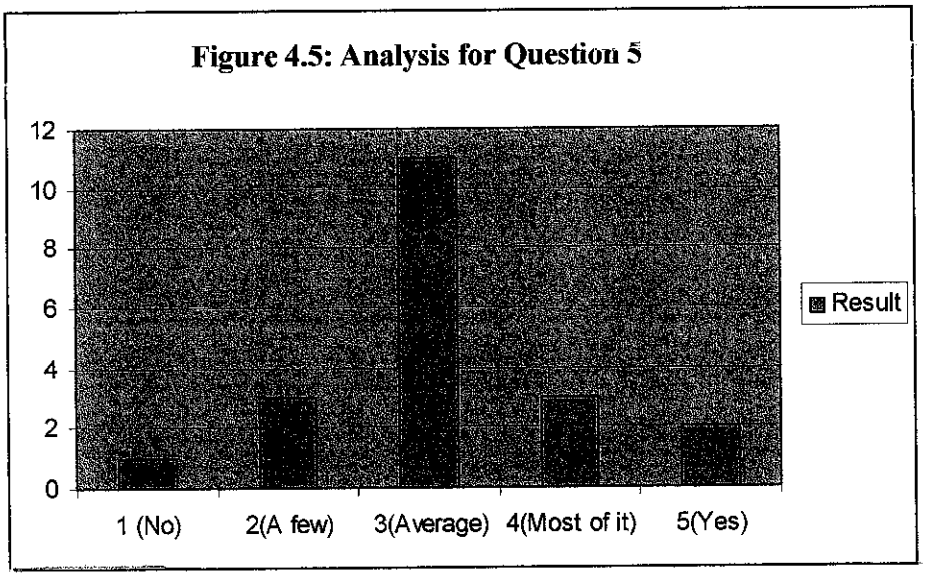
The result for this question indicates that nearly all the user suggested that changes and improvement need to be made to the existing interface. There are no users that say otherwise.

4.2.5 Analysis for Question 5

This question produced is to get to know whether user used all the functions offered by existing e-learning interface or not.

Answer Rating	Result
1 (No)	1
2(A few)	3
3(Average)	11
4(Most of it)	3
5(Yes)	2

Table 4.5 Result for Question 5



The result shows that there is average feedback for each rating but the most users indicate that they used some of the function.

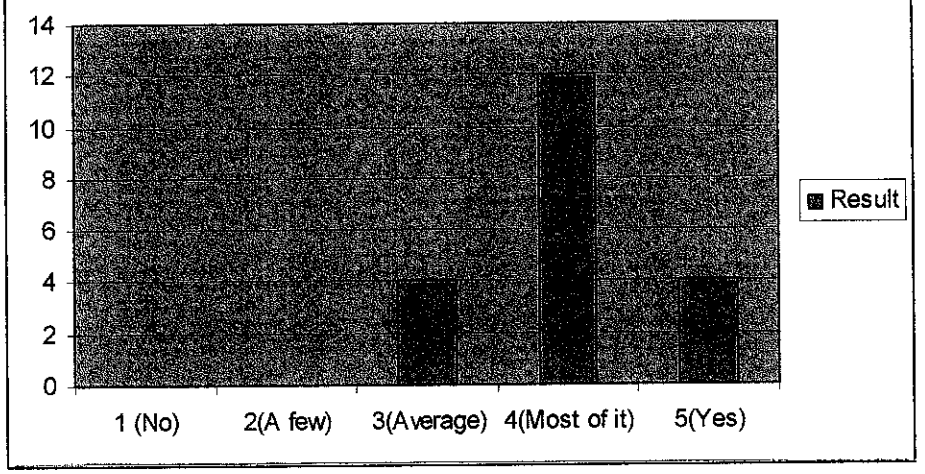
4.2.6 Analysis for Question 6

This question and the next 5 questions objective is base on the assumption that user agreed to have changes for the e-learning system interface. This question try to get user feedback whether the existing color used need to be change or not.

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	4
4(Most of it)	12
5(Yes)	4

Table 4.6 Result for Question 6

Figure 4.6: Analysis for Question 6



The result show that most user say that most of the color usage in the existing interface system need to be change. Some say a few needs to be change and there is a few that indicate that a major changes for color used need to be done.

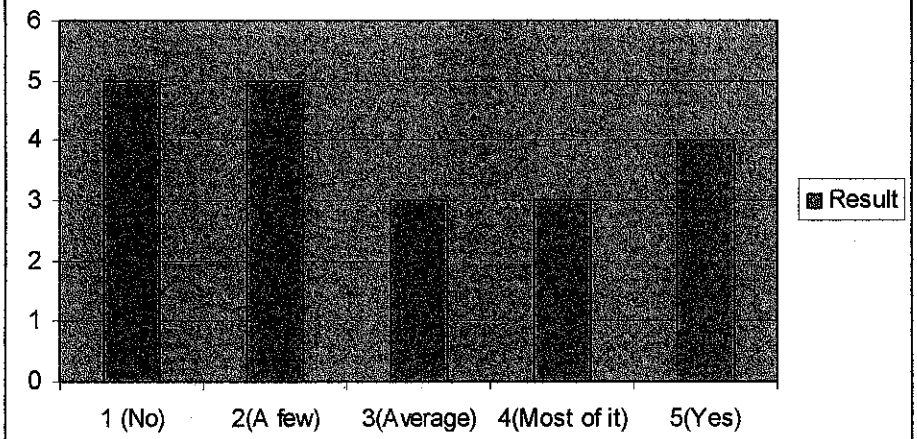
4.2.7 Analysis for Question 7

This question tried to get user feedback on whether or not to change the type of font used in the existing interface.

Answer Rating	Result
1 (No)	5
2(A few)	5
3(Average)	3
4(Most of it)	3
5(Yes)	4

Table 4.7 Result for Question 7

Figure 4.7: Analysis for Question 7



Result show that there is an average demand from user for every rating. Some say that the existing font use is OK and there also some other that say otherwise.

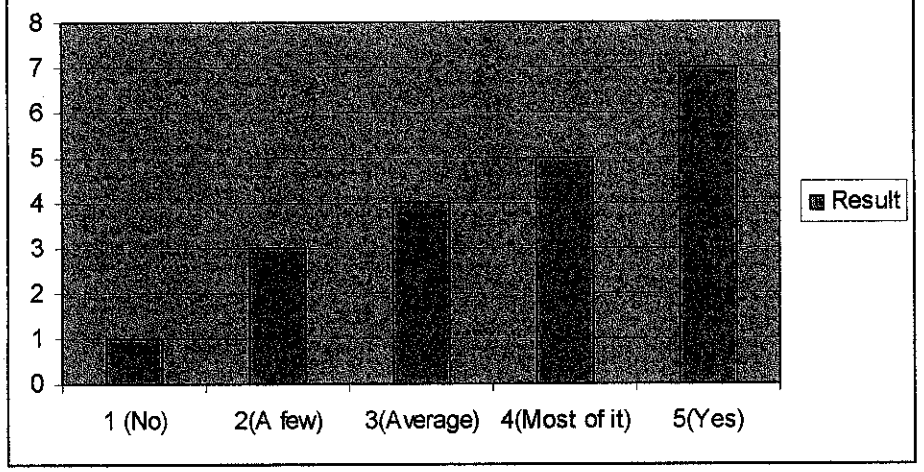
4.2.8 Analysis for Question 8

This question tried to get user respond on whether or not to change the icons used in the existing interface.

Answer Rating	Result
1 (No)	1
2(A few)	3
3(Average)	4
4(Most of it)	5
5(Yes)	7

Table 4.8 Result for Question 8

Figure 4.8: Analysis for Question 8



The result shows that most user thinks that icon used need to change to a more appropriate one.

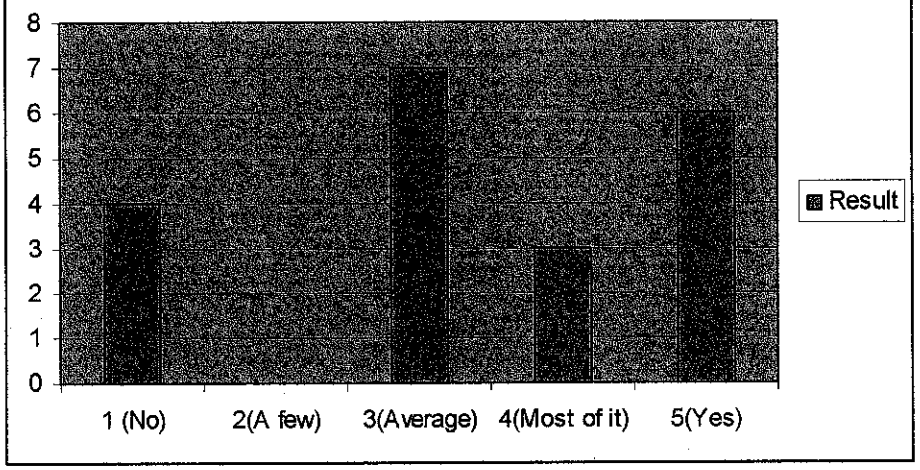
4.2.9 Analysis for Question 9

This question is the extension of question number 8 regarding the use of icons. This question tried to get user feedback on whether or not the existing icons create confusion to the user.

Answer Rating	Result
1 (No)	4
2(A few)	0
3(Average)	7
4(Most of it)	3
5(Yes)	6

Table 4.9 Result for Question 9

Figure 4.9: Analysis for Question 9



The result produce state that most users think that the icons used is quite confusing although some say that it's not. But comparing the total number the number of user that says the icons used are confusing is greater that the number of user saying that it's OK.

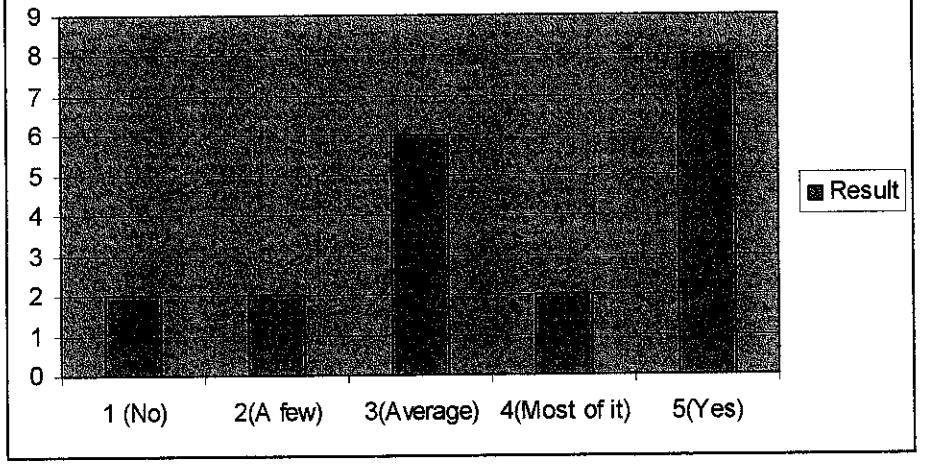
4.2.10 Analysis for Question 10

This question objective is to get user feedback on control on navigation area. This question tried to get user respond on whether they need more control on the new interface or just stick to the same control given to them in the existing UTP e-learning interface navigation.

Answer Rating	Result
1 (No)	2
2(A few)	2
3(Average)	6
4(Most of it)	2
5(Yes)	8

Table 4.10 Result for Question 10

Figure 4.10: Analysis for Question1 10



Result indicates that most of the user thinks that they need more control on navigation part of the interface but there is a few who think what ever given by the existing system enough for them is.

4.2.11 Analysis for question 11

This last question is an open ended question. The author just put in this question in order to get user overview on the concept and image of the new developed interface. The question for this question is: Please state in 3 words what you want the new UTP e-learning system to look like. Here are some of the results

User	Answers
1	• Cool • Fast • Every lecturer uses IT
2	• Smart • Efficient • Informative
3	• More user friendly • Less confusion • Predicted interface
4	• Put in video conferencing function • Online discussion • Bulletin board

Table 4.11 Result for Question 11

The author realizes that some answer seem irrelevant to the project but there is also several answer that contribute to the project development. Some of the answer given by the user shows that there is several improvement that had to be done in order to meet user expectation of the new enhance interface.

4.3 Journal, Internet Browsing

From this activity the author had gathered several important points for designing purposes. The important information is as below:

I. Design interface must be base on user prior knowledge; especially knowledge they have gained from experience in the real world.
II. Make objects and their controls visible and intuitive which mean including familiar looks.
III. The menu should ideally have no more than seven items on it.
IV. Place screen objects together in logical order.
V. Place buttons where the user's eye can easily find them
VI. Gives buttons clear symbols or labels.
VII. Group button together based on their function and frequency of use.

All these information were then used in the development of the design as guidelines.

4.4 Result and Discussion for Data Gathering and Analysis

Base on all the data collected from the questionnaires the author able to make a few assumptions that will help in the development of the new interface design. This assumption made have several similarities with the hypotheses that had been develop before. Besides helping the design development process it also show that the hypotheses made before is almost accurate to the real situation.

The main objective of the development of this new interface for UTP e-learning system would be to develop a design that support intuitive usage, creating didactic environment and also to minimize page through in completing one task. The author had developed several assumptions in order to assist the design development process.

4.4.1 To create design for intuitive usage

The author understands that intuitive means immediate understanding of something without conscious reasoning or study. To create such usage the author believes that familiarity and consistency plays an importance factor to tackle this matter. The author strongly believes that using standard buttons for menu and standard icons through out the system interfaces will be a good way to create such usage.

4.4.2 To develop a didactic environment

Didactic environment means an environment where the developer intended to convey instruction and information as well as pleasure and entertainment to the user. To achieve this objectives the author had made an assumption that in order to create such environment all the basic element in interface design such as color, font, button, icons, animation and others need to satisfied user needs. If the basic things offered according to user requirement the impact to user will be good. Then after being able to complete such task then the author believes then we can apply other element for navigation purpose such as using metaphors or exploration style to the new interface.

4.4.3 To create minimal page through

Creating minimal page through meaning to cut down several redundant interface in order to minimize the number of pages in completing one task. Too much pages will create tiredness and indirectly will started to reduce user motivation in using the system. The author had tried to analyze the existing UTP e-learning interface by developing a task flow diagram. From this diagram the author then tried to detect any redundant interface or any inappropriate interface in the system.

4.5 Design Phase (iterative)

Base on the assumption that the author had made after getting the result and analysis from the questionnaires distributed to select users the author had started the development of the interface design. The author had developed the interface design by phases. The design phases are as below:

- Storyboarding
  - Storyboard 1 – Positioning
  - Storyboard 2 – Colour Usage
  - Storyboard 3 – Navigation Plan
- Development of the interface

According to Kevin Krause interface is the door between the student and the instruction to facilitate access and reduce confusion, consistency in interface appearance and behavior. In order to design an intuitive interface the author had begins the design phase with the overall layout and design of the screen. There are four things that the author taken into considerations which are:

- Give buttons clear symbols or labels.
- Place buttons where the user's eye can easily find them.
- Group buttons together based on their function and frequency of use.
- Place screen objects together in a logical order.

4.5.1 Storyboard 1: Positioning

The author had arrange the position of all the elements in the interfaces according to the objectives of the interface with the support from the design principles form IBM(*Refer to Appendix 3-1)* and also the user interface design principles by Talin(*Refer to Appendix 3-2).* The storyboards are as below:

a) Log in page

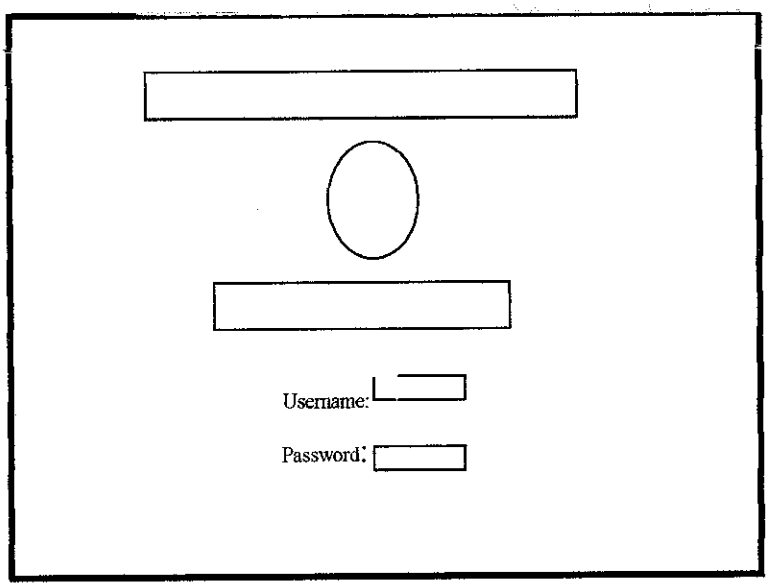


Figure 4.11 – Log in page positioning

This Log in page was design according to most interface design log in. The reason behind it is to create such familiarity where the author builds the interface depending on user existing knowledge for such design. The author had borrowed behaviors from system that is familiar to user. The objectives of such design is so that whenever user open this page the user will know what to do and what is the function of this page. The title of this system is intentionally put at the top of the screen and in a big font size in order to attract and capture user attention when they see this page

b) Main page

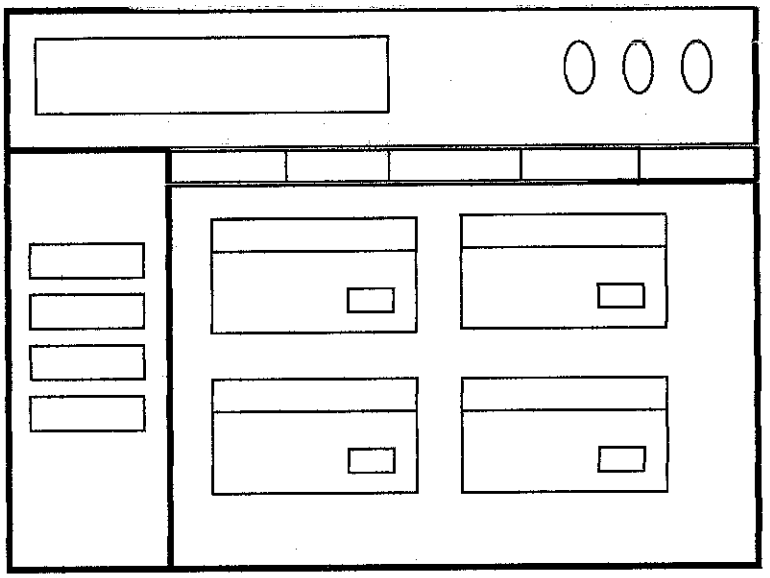


Figure 4.12 – Main page positioning

This main page has most of the look and feel of the existing system interface. This is to preserve the feeling of familiarity and simplicity as well as encouragement to the user. Beside that the button used in this new interface also been reduce to a much smaller number to instill simplicity feeling where the author tried to avoid using to many button and advance button that may distract user from accomplishing their tasks. The objective is to keep the interface simple and straightforward

c) The Course List

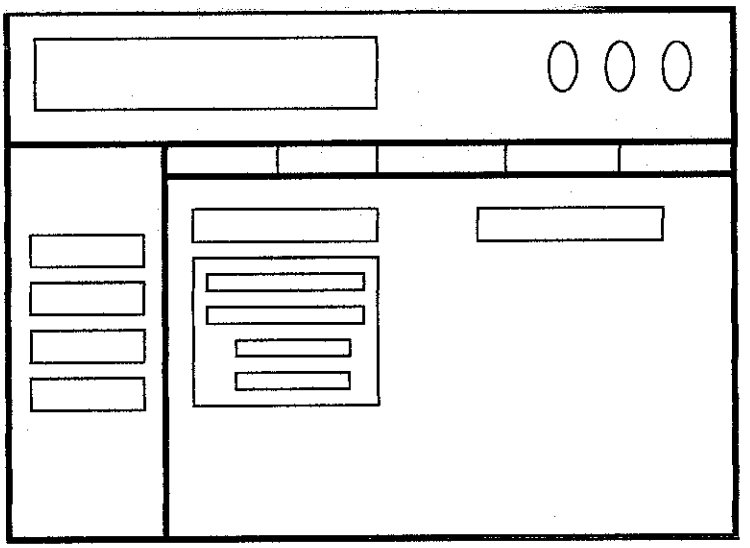


Figure 4.13 – Course List page positioning

The course list page is somehow similar to the main page. The author used the same layout, and positioning for each pages because the author had apply the principle of coherence in it where all pages have similar or consistence design to create the sense of intuitive interface in this system interface.

The other pages created by the author also apply this same layout and positioning but with a slight changes. In this page the graphical exploration is used in the course list. This is adapted from Microsoft Explorer interface. The significant behind it because all the computers provided in UTP Laboratory used Windows as the platform and all of student are very much familiar with the Window-base usage. The design are implemented in this design to create sense of conformability and familiarity to the user, plus the graphical exploration is much more user friendly, easy to understand and contain the element of simplicity in its design.

The author also apply graphical exploration interface in other pages such as:

- Organization List
- Services Page
- Academic Web Resources page

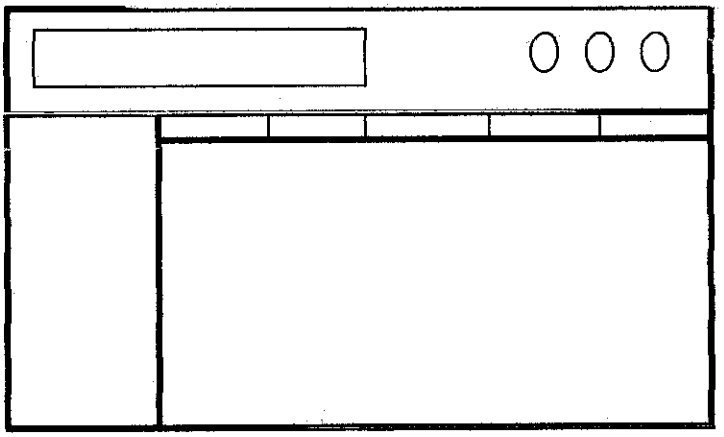


Figure 4.14 – Organizational page positioning

4.5.2 Storyboard 2 – Colour Usage

Colour of the interface can convey message and affect user mood while using the interface. Selecting the right colour will help create the environment that we wanted. In this project the author had chosen colour with high white intensity. Colour that is also known as cool colour group gave a pleasing effect to human eyes. The specific colour that been choose in this interface design will be light blue colour with several shade and white intensity levels. The author believe the same effect produce by using the colour shade chosen also can be achieve by using other colour from the same group. The other colour from cool colour group is as below. For this project colour from the cool shades are chosen because it gave a soothing and pleasing effect to human eyes and will not give negative effect to human eyes such as eye strain or headache after staring at the page for quite a long time. The font colour used in the interface will be standard black colour. A dark font colour on a light colour background will not hurt the user eyes after staring at the page for a certain time. The author had chosen such combination because the author realize that user

usually spent a longer time reading notes, announcement and other information from e-learning interface

4.5.3 Storyboard 3 – Navigation Plan

The navigation parts of the interface are more or less the same as the existing navigation but with several minor changes. The changes are:

- Log in task – the existing interface used 2 pages for this task which is then reduce to only one page in the new interface.
- The existing e-learning directory consist of 7 elements/functions are reduce to 4 element because some of the elements in the directory are seldom used by the users.
- The search page that being applied to all pages with different search category and uses lots of space before are emerged as one search page that can be search according to category.

4.5.4 Development of the actual interfaces

a) Login Page

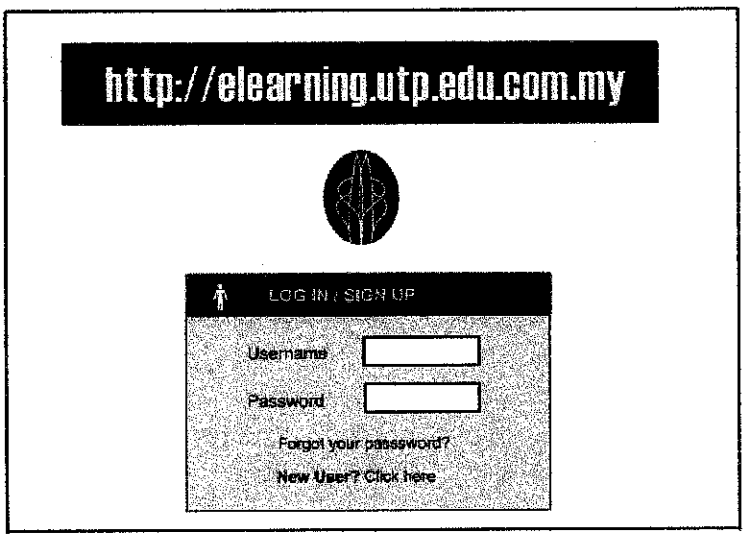


Figure 4.15 – Log in Page

b) Main Page

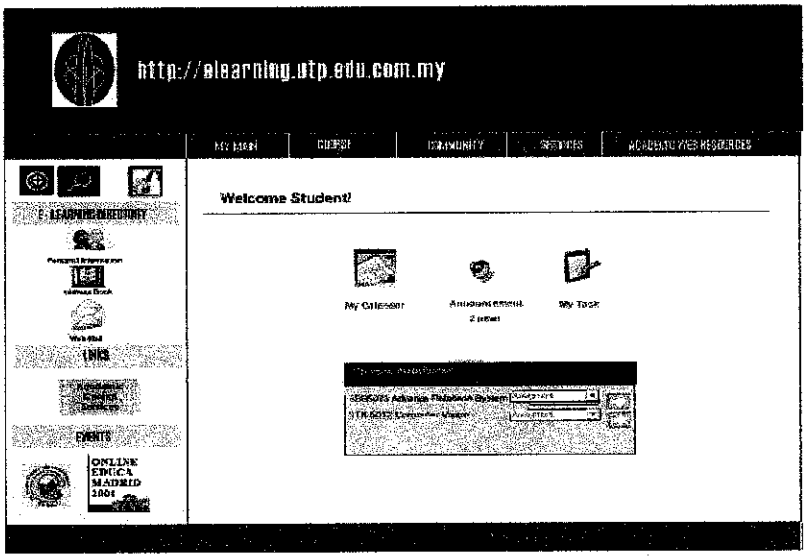


Figure 4.16 – Main Page

c) Search page

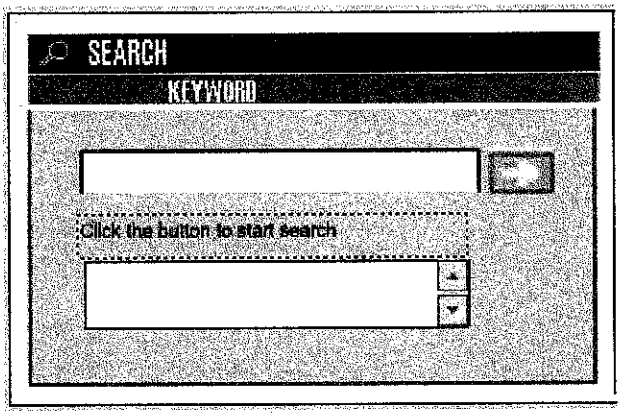


Figure 4.17 – Search Page

d) My Calendar Page

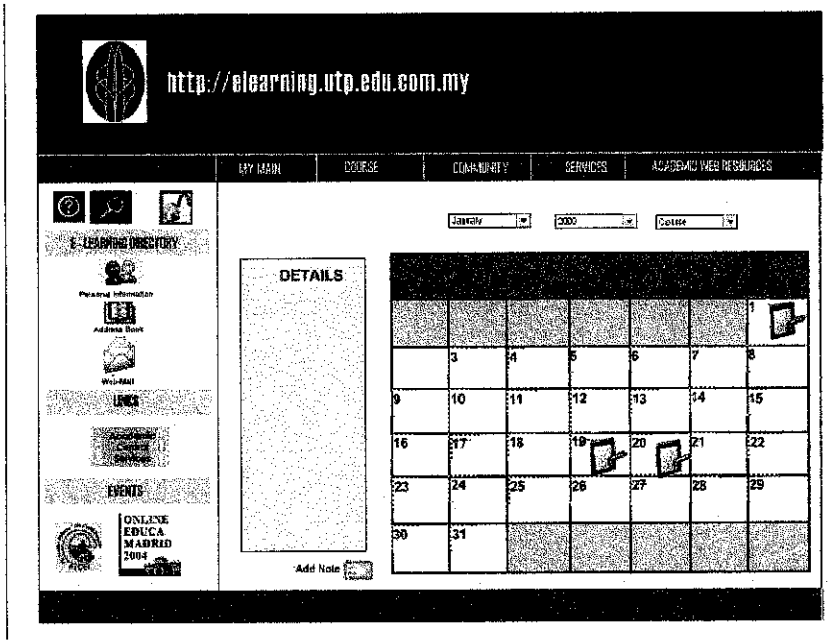


Figure 4.18 – Search Page

4.6 Testing Phase - User Acceptance Testing

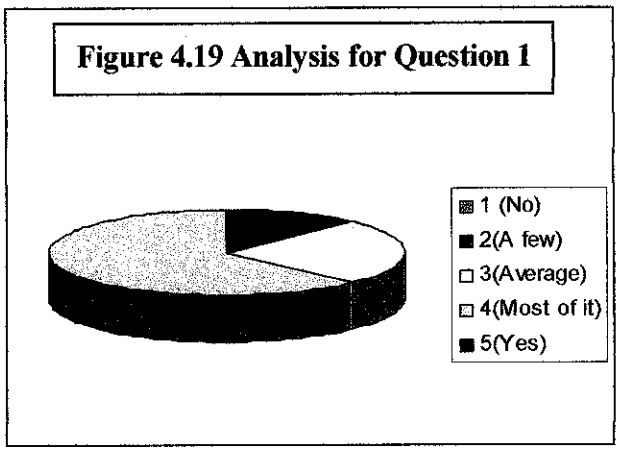
User Acceptance testing was conducted in order to gain user feedback on the new interface developed. The main purpose of the review is to evaluate the content or instructional design, the screen design and layout, appropriateness of metaphor, and the technical performance. For this purpose the author had select 8 users randomly to perform the testing. The users are consisting of UTP student form various programme and year of studies. A test script had been provided *(See Appendix 3-8)*.The author had performed two (2) User Acceptance Testing with specific treatment in order to identify whether the hypotheses develop before can be proven or not. The second User Acceptance Testing treatment was adding more icon base element to the design as well as more metaphor application compared to the first one. Here is the result for the first User Acceptance Testing.

4.6.1 Analysis for Question 1

This question objective is to get user opinion on the overall design of the interfaces whether the interface is good and attractive enough or not.

Answer Rating	Result
1 (No)	0
2(A few)	1
3(Average)	2
4(Most of it)	5
5(Yes)	0

Table 4.12 Result for Question



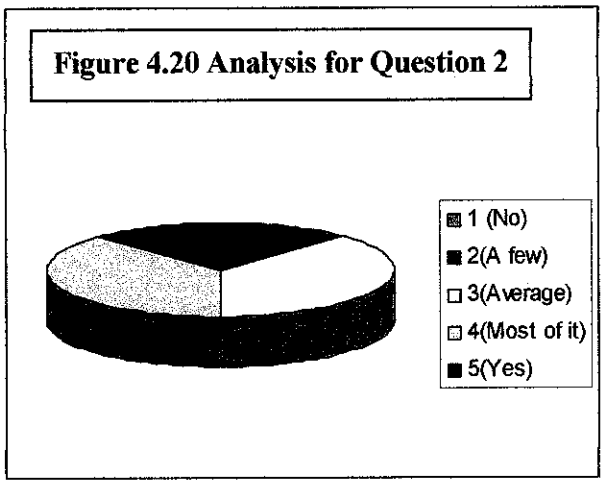
The result indicate that user find that the interface is quite impressive and attract their attention.5 out of 8 user totally agreed with the statement that all the interface is attractive,4 out of 8 user find that most of the interface are attractive, 3 said that it's average while 2 said that only a few are attractive.

4.6.2 Analysis for Question 2

This question tried to asked user feedback on the colour used in this interface design whether the usage of colour in the interface design is good enough and appealing or not.

Answer Rating	Result
1 (No)	0
2(A few)	1
3(Average)	3
4(Most of it)	3
5(Yes)	1

Table 4.13 Result for Question 2



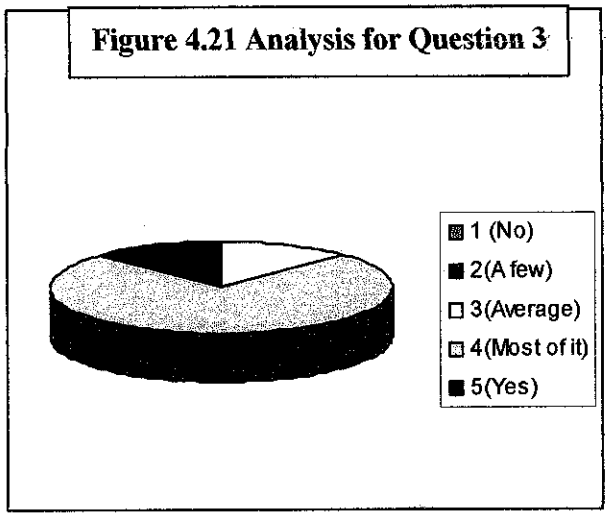
The result indicate that most of the user agreed that the colour used is appealing and good where 3 say it's average ,4 say most of it are good and 5 say all of it are good. Only a small portion of user which is 2 people says that a few of the colour usage in this interface design is good.

4.6.3 Analysis for Question3

The question objectives is to get user feedback on the navigation elements of the interface and the button used itself

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	1
4(Most of it)	6
5(Yes)	1

Table 4.14 Result for Question3



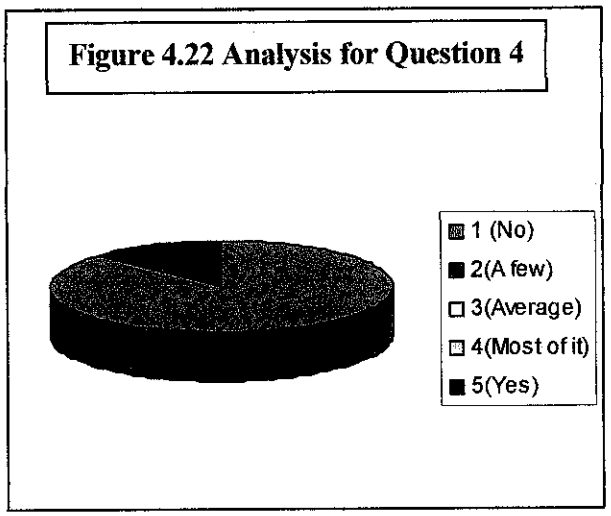
The result shows that most of the users are satisfied with the navigation and the buttons usage in the interface.

4.6.4 Analysis for Question 4

This question objective is to know whether the function used in the interface is clearly define or make users confuse.

Answer Rating	Result
1 (No)	7
2(A few)	1
3(Average)	0
4(Most of it)	0
5(Yes)	0

Table 4.15 Result for Question 4



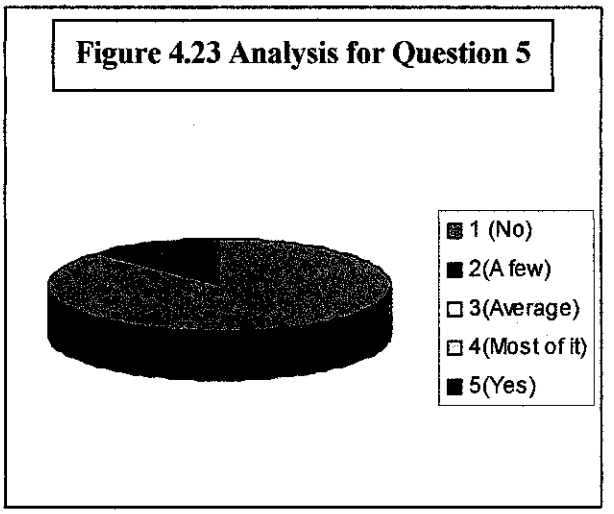
The result shows that most of the users think that the functions in this interface are clearly defined. One of the reason that user didn't experience confusion is because the author had use most of the functions in the existing UTP e-learning in this new interface to create familiarity and simplicity in the environment.

4.6.5 Analysis for Question 5

The question objective is to get user feedback on the search interface provided.

Answer Rating	Result
1 (No)	7
2(A few)	1
3(Average)	0
4(Most of it)	0
5(Yes)	0

Table 4.16 Result for Question 5



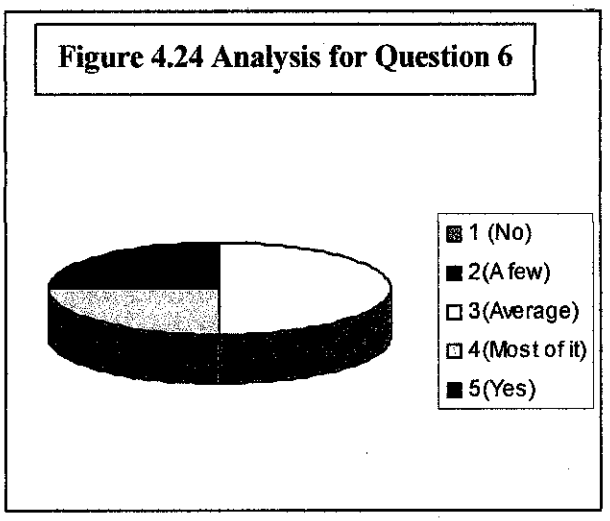
The result shows that users are satisfied with the search page provided.

4.6.6 Analysis for Question 6

This question objective is to get user feedback whether the buttons functionality in the interface are easily identified or not.

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	4
4(Most of it)	2
5(Yes)	2

Table 4.17 Result for Question 6



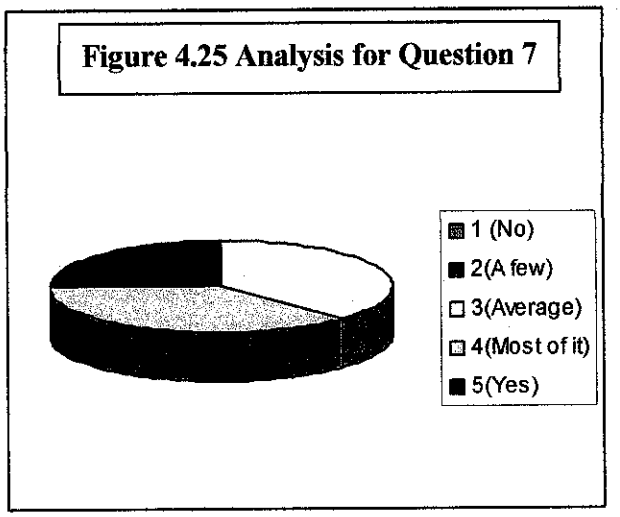
The result show that the buttons functionality in the interface are clearly define and the user didn't have so many problems to figure out the button functions.

4.6.7 Analysis for Question 7

These questions aims are to get user feedback on whether the environment created are acceptable by the user or not as an e-learning environment.

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	3
4(Most of it)	3
5(Yes)	2

Table 4.18 Result for Question 7



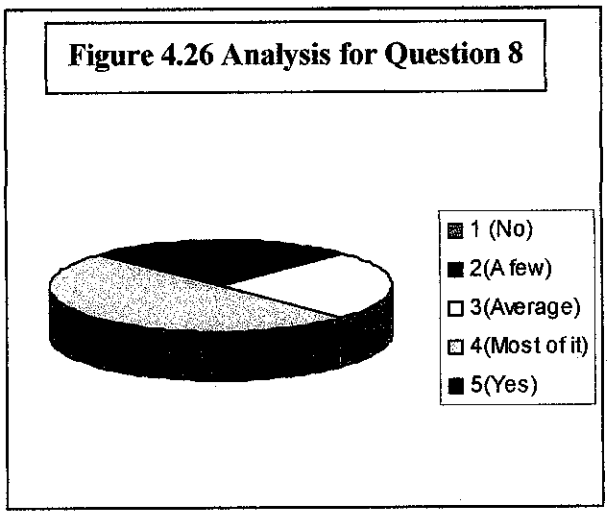
The result indicates that most of the users say that this environment reflects the e-learning environment but there is still a few of the user that think that only a few of the interface reflect that.

4.6.8 Analysis for Question 8

This question aim is to get user feedback on whether user need help in navigation the interface or not. This is to ensure whether the intuitive design that the author had apply in this interface are functioning.

Answer Rating	Result
1 (No)	0
2(A few)	1
3(Average)	2
4(Most of it)	4
5(Yes)	1

Table 4.19 Result for Question 8



The results indicate that the users think that they didn't need help in navigating the interface.

The Descriptive Statistic of the data:

	N	Min	Max	Mean
Overall interface	8	2	4	3.5
Colour	8	2	5	3.5
Navigation and Buttons	8	3	5	4.0
Function offered	8	4	5	4.875
Level of attractiveness	8	4	5	4.875
Button	8	3	5	3.75
Environment	8	3	5	3.875
Help	8	2	5	3.625

Table 4.20 Descriptive Statistic of User Acceptance Testing 1

The overall conclusion of the whole testing phase shows that user level of acceptance of the new system is good. All variables mean show result that is more than half (Total = 5.00) and this is a good respond. It shows that the interfaces are accepted by users.

Below is the second User Acceptance Testing result. The data had been analyse and conclude in one table base on the question by question analysis (*Refer to Appendix 4-1*)

	N	Min	Max	Mean
Overall interface	8	3	5	3.875
Colour	8	2	5	3.5
Navigation and Buttons	8	3	5	4.125
Function offered	8	4	5	4.875
Level of attractiveness	8	4	5	4.875
Button	8	3	5	4.25
Environment	8	3	5	4.125
Help	8	2	4	3.375

Table 4.21 descriptive Statistic of User Acceptance Testing 2

The table indicate that mean values for each variable are more than half. (Total = 5.00).This show that user acceptance are more than 50% and consider as good. The second User Acceptance Testing also shows a positive increase in user acceptance by comparing the value of mean for every variable with the mean values in User Acceptance Testing 1. It shows that there is a relationship between level of icon and metaphor used in the design with user acceptance level. The higher level of icon and metaphor used the higher the level of user acceptance. Directly this result support the hypotheses develop before (*Refer to Appendix 3-5*) and indicated that the alternate hypothesis is proven and reject null hypotheses.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Through out the process of completing this project the author has encountered a problem when trying to develop the design of the project. A few studies had been done in order to choose the good one. This turns out to be a good activity because the author had been able to understand more on the pros and cons of every approach and design chosen and get to know each specialty.

From the process the author also being able to discover that good interface will enhance user performance and also user likeliness to use the system. In educational usage for example in e-learning system it will improve user performance in studies. (In the condition where user fully utilizes the system).

By developing this report the author also being able to understand and see the research area more clearly and being able to define clearly what need to be done in developing this project. This report is a good guide for the author in continuing the development of the project in the near future.

The User Acceptance Testing results indicate that the design developed is acceptable by the user and intuitive enough where used doesn't need explanation or help to test the simulated design interface. The testing also shows that the author had been able to achieve the objectives of the project. The author also hopes that the design ideas can help enhance the effectiveness of UTP E-Learning system as well as enhance the quality of learning in campus.

5.2 Recommendations

The produce developed by the author is merely a prototype. There are always a few rooms for enhancement. This product only concentrates on student as the user of the interface. For further enhancement the author suggest that the user is expand to a bigger and broader group for example involving academic and non-academic staff in UTP that are also used the e-learning system interface.

The product also can be improved in term of cosmetic and creative features to make it move lively and appealing. In term of the principle design applied in the development of the interfaces, the author suggest that for future development that a much thorough used and application of design principle are used in the navigation part of the interface. A much through study on user needs and wants can be done to better understand their requirements.

In term of testing the author also think that applying the design to the real system for field testing purposes is better then laboratory testing as done in this project to get more accurate result as well as been able to identify other factors that influent the result of the testing in real environment.

REFERENCES

Alan Dix, Janet Finlay, Grerory Abowd, Russell Beale1997.*Human Computer Interaction*

(2) Essex, Prentice Hall

IBM United State.8 March 2004< http://www-306.ibm.com/ibm/easy/eou_ext.nsf /Publish/6b >

Anne Nicol.2001, "Interfaces for Learning: in Brenda Laurel .*The Art of Human Computer Interface Design,* USA, Addison Wesley

Kevin Kruse.2000-2004 < http://www.e-learningguru.com/articles/art4_2.htm>

Kevin Kruse.2000-2004 < http://www.e-learningguru.com/articles/art4_3.htm>

Kevin Kruse.2000-2004 < http://www.e-learningguru.com/articles/art4_4.htm>

Kevin Kruse.2000-2004 < http://www.e-learningguru.com/articles/art4_5.htm>

APPENDICES

APPENDICES

APPENDIX 3-1

DESIGN PRINCIPLE

Design Principles

Except from IBM Homepage http://www.ibm.com/us/

Simplicity: Don't compromise usability for function

Keep the interface simple and straightforward. Users benefit from function that is easily accessible and usable. A poorly organized interface cluttered with many advanced functions distracts users from accomplishing their everyday tasks. A well-organized interface that supports the user's tasks fades into the background and allows the user to work efficiently.

Basic functions should be immediately apparent, while advanced functions may be less obvious to new users. Function should be included only if a task analysis shows it is needed. Therefore, keep the number of objects and actions to a minimum while still allowing users to accomplish their tasks.

Support: Place the user in control and provide proactive assistance

To give users control over the system, enable them to accomplish tasks using any sequence of steps that they would naturally use. Don't limit them by artificially restricting their choices to your notion of the "correct" sequence.

The system should also allow users to establish and maintain a working context, or frame of reference. The current state of the system and the actions that users can perform should be obvious. Users should be able to leave their systems for a moment or a day and find the systems in the same familiar state when they return. This contextual framework contributes to their feeling of stability.

Most users perform a variety of tasks, being expert at some and novice at others. In addition to providing assistance when requested, the system should recognize and anticipate the user's goals, and offer assistance to make the task easier. Ideally, assistance should provide users with knowledge that will allow them to accomplish their tasks quickly. Intelligent assistance is like the training wheels on a bicycle - at some point, most users will want to take them off and go forward on their own. The assistance should allow them to become independent at some point when they choose to be so.

Familiarity: Build on users' prior knowledge

Allow users to build on prior knowledge, especially knowledge they have gained from experience in the real world. A small amount of knowledge, used consistently throughout an interface, can empower the user to accomplish a large number of tasks. Concepts and techniques can be learned once and then applied in a variety of situations. Users should not have to learn new things to perform familiar tasks. The use of concepts and techniques that users already understand from their real world experiences allows them to get started quickly and make progress immediately.

The metaphors used in today's user interfaces tend to be inadequate when compared to the real world. Through the use of visuals and interaction techniques that more closely resemble users' real world experiences, there should be little need to continue reliance on such metaphors.

In the past, designers tended to invoke a principle of consistency when no single design alternative appeared to be the best answer. By choosing to be consistent with something the user already understands, an interface can be made easier to learn, more productive, and even fun to use.

Avoid the tendency to employ consistency without understanding your users, their tasks, and their shared experiences. When choosing a dimension within which to be consistent, seek to understand what the user expects and be consistent with those expectations. Providing a familiar experience is the ultimate use of consistency in which a truly intuitive interface will result.

Obviousness: Make objects and their controls visible and intuitive

Where you can, use real-world representations in the interface. Real-world representations and natural interactions (direct action) give the interface a familiar look and feel and can make it more intuitive to learn and use. Icons and windows were early attempts to draw on user experiences outside the computing domain. As we move toward real-world representations, reliance on such computer artifacts should decline. In an object-oriented interface the objects and concepts presented to users parallel familiar things from the real world; for example:

- Trash can - when we throw things away we usually use some type of trash receptacle or "trash can". An object on the desktop displayed as a trash can communicates to users that it is a place for discarding things. It should look like the real object rather than like an abstract container, and the user should be able to show its contents in a meaningful way.
- Telephone - the actions we take with telephones are so familiar to most of us that they require little thought. A telephone object on the desktop indicates to users that it will allow them to perform phone-related tasks, and users will expect it to behave like the real thing.

The controls of the system should be clearly visible and their functions identifiable. Visual representations provide cues and reminders that help users understand roles, remember relationships, and recognize what the computer is doing. For example, the numbered buttons on the telephone object indicate that they can be used to key in a telephone number.

Allow users to interact directly with objects and minimize the use of indirect techniques. Identifying an object and doing something with it (like picking up the handset of a phone to answer it) usually are not separate actions in the real world. Likewise, with direct action techniques, explicit selection is not necessary because selection is implicit in the actions users take with objects. Real-world 3D interfaces are especially conducive to direct interaction.

Encouragement: Make actions predictable and reversible

A user's actions should cause the results the user expects. In order to meet those expectations, the designer must understand the user's tasks, goals, and mental model. Use terms and images that match users' task experience, and that help users understand the objects and their roles and relationships in accomplishing tasks.

Users should feel confident in exploring, knowing they can try an action, view the result, and undo the action if the result is unacceptable. Users feel more comfortable with interfaces in which their actions do not cause irreversible consequences.

Even seemingly trivial user actions, such as deselection or moving objects, should be reversible. For example, a user who spends several minutes deliberating and selecting individual files to be archived from a group will be very upset if all the files are accidentally deselected and the deselection cannot be undone.

Avoid bundling actions together, because the user may not anticipate the side effect. For example, if a user chooses to cancel a request to send a note, only the send request should be cancelled. Do not bundle another action, such as deletion of the note, with the cancel request. Rather than implementing composite actions, make actions independent and provide ways to allow users to combine them when they wish.

Satisfaction: Create a feeling of progress and achievement

Allow the user to make uninterrupted progress and enjoy a sense of accomplishment. Reflect the results of actions immediately; any delay intrudes on users' tasks and erodes confidence in the system. Immediate feedback allows users to assess whether the results were what they expected and to take alternative action immediately. For example, when a user chooses a new font, the font of all applicable text, or of sample text, should change immediately. The user can then decide if the effect is what was desired and, if not, can change it before switching attention to something else.

Offer a preview of the results of an action when it would be inconvenient for a user to apply the action and then reverse it. For example, if a user wants to bold, underscore, and use Helvetica font in certain places throughout a document, provide a sample part of that document with those changes applied, allowing the user to decide if that is the right action to take. This saves the user a lot of time by not having to reverse the action that's been applied to an entire document and enhances the user's confidence in the system.

Avoid situations where users may be working with information that is not up-to-date. Information should be updated immediately or refreshed as soon as possible so that users are not making incorrect decisions or assumptions. If, for some reason, the results of a refresh cannot be displayed immediately, the situation should be communicated to users. This becomes especially important in networked environments where it is more difficult to maintain state between networked systems dynamically. For example, most Web browsers display a completion percentage in the information area so that users know the progress of the graphics loading process.

Availability: Make all objects available at all times

Users should be able to use all of their objects in any sequence and at any time. Avoid the use of modes, those states of the interface in which normally available actions are no longer available, or in which an action causes different results than it normally does.

Modes restrict the user's ability to interact with the system. For example, one of the most common uses of modes in menu-driven systems is the modal dialog box (such as "Print" and "Save as") used to request command parameters. Modal dialogs tend to lock users out of their system; to continue, users must complete - or cancel - the modal dialog. If users need to refer to something in an underlying window to complete the dialog, they must cancel the dialog, access the information they need and re-invoke the dialog.

Safety: Keep the user out of trouble

Users should be protected from making errors. The burden of keeping the user out of trouble rests on the designer. The interface should provide visual cues, reminders, lists of choices, and other aids, either automatically or on request. Humans are much better at recognition than recall. Contextual and hover help, as well as agents, can provide supplemental assistance. Simply stated, eliminate the opportunity for user error and confusion.

Users should never have to rely on their own memory for something the system already knows, such as previous settings, file names, and other interface details. If the information is in the system in any form, the system should provide it.

Two-way communication may be necessary at times to allow users to clarify or confirm requests, or to remedy a problem. In the past, many interfaces have treated communication with users as primarily one-way, computer-to-user. The communication should be interactive - as rich in presentation and interaction capabilities as the rest of the interface. It should present relevant information, provide access to related information and help, and allow users to make task-specific decisions to continue. For instance, spell check, as designed in some systems, highlights potentially misspelled words as users work, allowing them to either select a new word or continue to work until they reach a point where they can go back and validate the potentially misspelled words.

Adopt the following design perspective: users know what they want to accomplish, but sometimes they find it difficult to express their desires using the objects and actions provided, and the system is unable to recognize their request. Two-way communication may be used to help users reach their goals.

Versatility: Support alternate interaction techniques

Allow users to choose the method of interaction that is most appropriate to their situation. Interfaces that are flexible in this way are able to accommodate a wide range of user skills, physical abilities, interactions, and usage environments.

Each interaction device is optimized for certain uses or users and may be more convenient in one situation than another. For example, a microphone used with voice-recognition software can be helpful for fast entry of text or in a hands-free environment. Pen input is helpful for people who sketch, and mouse input works well for precisely indicating a selection. Alternative output formats, such as computer-generated voice output for foreign language instruction, are useful for some purposes. No single method is best for every situation.

Users should be allowed to switch between methods to accomplish a single interaction. For example, allow the user to swipe-select using the mouse, then to adjust the selection using the keyboard. At the same time, users should not be *required* to alternate between input devices to accomplish what they perceive as a single step or a series of related steps in a

task. For example, it would be tedious to require the use of a mouse for scrolling while editing text from the keyboard. Users should be able to complete an entire useful sequence through the same input device.

Providing a range of interaction techniques recognizes that users are individuals with different abilities and situations. The differences include disabilities, preferences, and work environments.

Personalization: Allow users to customize

The interface should be tailorable to individual users' needs and desires. No two users are exactly alike. Users have varying backgrounds, interests, motivations, levels of experience, and physical abilities. Customization can help make an interface feel comfortable and familiar.

Personalizing a computer interface can also lead to higher productivity and user satisfaction. For example, allowing users to change default values can save them time and hassle when accessing frequently used functions.

In an environment where multiple users are using a shared machine, allow the users to create their own system personality and make it easy to reset the system. In an environment where one user may be using many computers, make personalization information portable so the user can carry that "personality" from one system to another.

Affinity: Bring objects to life through good visual design

The goal of visual design in the user interface is to surface to the user in a cohesive manner all aspects of the design principles. Visual design should support the user model and communicate the function of that model without ambiguities. Visual design should not be the "icing on the cake" but an integral part of the design process. The final result should be an intuitive and familiar representation that is second nature to users.

APPENDIX 3-2

PRINCIPLE OF USER INTERFACE DESIGN

A Summary of Principles for User-Interface Design.

By Talin

This document represents a compilation of fundamental principles for designing user interfaces, which have been drawn from various books on interface design, as well as my own experience. Most of these principles can be applied to either command-line or graphical environments. I welcome suggestions for changes and additions -- I would like this to be viewed as an "open-source" evolving document.

1. The principle of user profiling

-- Know who your user is.

Before we can answer the question "How do we make our user-interfaces better", we must first answer the question: Better for *whom*? A design that is better for a technically skilled user might not be better for a non-technical businessman or an artist.

One way around this problem is to create user models. [TOG91] has an excellent chapter on brainstorming towards creating "profiles" of possible users. The result of this process is a detailed description of one or more "average" users, with specific details such as:

- What are the user's goals?
- What are the user's skills and experience?
- What are the user's needs?

Armed with this information, we can then proceed to answer the question: How do we leverage the user's strengths and create an interface that helps them achieve their goals?

In the case of a large general-purpose piece of software such as an operating system, there may be many different kinds of potential users. In this case it may be more useful to come up with a list of *user dichotomies*, such as "skilled vs. unskilled", "young vs. old", etc., or some other means of specifying a continuum or collection of user types.

Another way of answering this question is to talk to some real users. Direct contact between end-users and developers has often radically transformed the development process.

2. The principle of metaphor

-- Borrow behaviors from systems familiar to your users.

Frequently a complex software system can be understood more easily if the user interface is depicted in a way that resembles some commonplace system. The ubiquitous "Desktop metaphor" is an overused and trite example. Another is the *tape deck* metaphor seen on many audio and video player programs. In addition to the standard transport controls (play, rewind, etc.), the tape deck metaphor can be extended in ways that are quite natural, with functions such as *time-counters* and *cueing buttons*. This concept of "extendibility" is what distinguishes a powerful metaphor from a weak one.

There are several factors to consider when using a metaphor:

- Once a metaphor is chosen, it should be spread widely throughout the interface, rather than used once at a specific point. Even better would be to use the same metaphor spread over several applications (the tape transport controls described above is a good example.) Don't bother thinking up a metaphor which is only going to apply to a single button.
- There's no reason why an application cannot incorporate several different metaphors, as long as they don't clash. Music sequencers, for example, often incorporate both "tape transport" and "sheet music" metaphors.
- Metaphor isn't always necessary. In many cases the natural function of the software itself is easier to comprehend than any real-world analog of it. Don't strain a metaphor in adapting it to the program's real function. Nor should you strain the meaning of a particular program feature in order to adapt it to a metaphor.
- Incorporating a metaphor is not without certain risks. In particular, whenever physical objects are represented in a computer system, we inherit not only the beneficial functions of those objects but also the detrimental aspects.
- Be aware that some metaphors don't cross cultural boundaries well. For example, Americans would instantly recognize the common U.S. Mailbox (with a rounded top, a flat bottom, and a little red flag on the side), but there are no mailboxes of this style in Europe.

3. The principle of feature exposure

-- Let the user see clearly what functions are available

Software developers tend to have little difficulty keeping large, complex mental models in their heads. But not everyone prefers to "live in their heads" -- instead, they prefer to concentrate on analyzing the sensory details of the environment, rather than spending large amounts of time refining and perfecting abstract models. Both type of personality (labeled "Intuitive" and "Sensable" in the Myers-Briggs personality classification) can be equally intelligent, but focus on different aspects of life. It is to be noted that according to some psychological studies "Sensables" outnumber "Intuitives" in the general population by about three to one.

Intuitives prefer user interfaces that utilize the power of abstract models -- command lines, scripts, plug-ins, macros, etc. Sensables prefer user interfaces that utilize their perceptual abilities -- in other words, they like interfaces where the features are "up front" and "in their face". Toolbars and dialog boxes are an example of interfaces that are pleasing to this personality type.

This doesn't mean that you have to make everything a GUI. What it does mean, for both GUI and command line programs, is that the features of the program need to be *easily exposed* so that a quick visual scan can determine what the program actually does. In some cases, such as a toolbar, the program features are exposed by default. In other cases, such as a printer configuration dialog, the exposure of the underlying printer state (i.e. the buttons and controls which depict the *conceptual printing model*) are contained in a dialog box which is brought up by a user action (a feature which is itself exposed in a menu).

Of course, there may be cases where you don't wish to expose a feature right away, because you don't want to overwhelm the beginning user with too much detail. In this case, it is best to structure the application like the layers of an onion, where peeling away each layer of skin reveals a layer beneath. There are various levels of "hiding": Here's a partial list of them in order from most exposed to least exposed:

- Toolbar (completely exposed)
- Menu item (exposed by trivial user gesture)
- Submenu item (exposed by somewhat more involved user gesture)
- Dialog box (exposed by explicit user command)
- Secondary dialog box (invoked by button in first dialog box)
- "Advanced user mode" controls -- exposed when user selects "advanced" option
- Scripted functions

The above notwithstanding, in no case should the primary interface of the application be a reflection of the *true complexity* of the underlying implementation. Instead, both the interface and the implementation should strive to match a simplified conceptual model (in other words, the *design*) of what the application does. For example, when an error occurs, the explanation of the error should be phrased in a way that relates to the current user-centered activity, and not in terms of the low-level fault that caused there error.

4. The principle of coherence

-- The behavior of the program should be internally and externally consistent

There's been some argument over whether interfaces should strive to be "intuitive", or whether an intuitive interface is even possible. However, it is certainly arguable that an interface should be *coherent* -- in other words logical, consistent, and easily followed. ("Coherent" literally means "stick together", and that's exactly what the parts of an interface design should do.)

Internal consistency means that the program's behaviors make "sense" with respect to other parts of the program. For example, if one attribute of an object (e.g. color) is modifiable using a pop-up menu, then it is to be expected that other attributes of the object would also be editable in a similar fashion. One should strive towards the principle of "least surprise".

External consistency means that the program is consistent with the environment in which it runs. This includes consistency with both the operating system and the typical suite of applications that run within that operating system. One of the most widely recognized forms of external coherence is compliance with *user-interface standards*. There are many others, however, such as the use of standardized scripting languages, plug-in architectures or configuration methods.

5. The principle of state visualization

-- Changes in behavior should be reflected in the appearance of the program

Each change in the behavior of the program should be accompanied by a corresponding change in the appearance of the interface. One of the big criticisms of "modes" in interfaces is that many of the classic "bad example" programs have modes that are visually indistinguishable from one another.

Similarly, when a program changes its appearance, it should be in response to a behavior change; A program that changes its appearance for no apparent reason will quickly teach the user not to depend on appearances for clues as to the program's state.

One of the most important kinds of state is the *current selection*, in other words the object or set of objects that will be affected by the next command. It is important that this internal state be visualized in a way that is consistent, clear, and unambiguous. For example, one common mistake seen in a number of multi-document applications is to forget to "dim" the selection when the window goes out of focus. The result of this is that a user, looking at several windows at once, each with a similar-looking selection, may be confused as to exactly which selection will be affected when they hit the "delete" key. This is especially true if the user has been focusing on the selection highlight, and not on the window frame, and consequently has failed to notice which window is the active one. (Selection rules are one of those areas that are covered poorly by most UI style guidelines, which tend to concentrate on "widgets", although the Mac and Amiga guidelines each have a chapter on this topic.)

6. The principle of shortcuts

-- Provide both concrete and abstract ways of getting a task done

Once a user has become experienced with an application, she will start to build a mental model of that application. She will be able to predict with high accuracy what the results of any particular user gesture will be in any given context. At this point, the program's attempts to make things "easy" by breaking up complex actions into simple steps may seem cumbersome. Additionally, as this mental model grows, there will be less and less need to look at the "in your face" exposure of the application's feature set. Instead, pre-memorized "shortcuts" should be available to allow rapid access to more powerful functions.

There are various levels of shortcuts, each one more abstract than its predecessor. For example, in the emacs editor commands can be invoked directly by name, by menu bar, by a modified keystroke combination, or by a single keystroke. Each of these is more "accelerated" than its predecessor.

There can also be alternate methods of invoking commands that are designed to increase power rather than to accelerate speed. A "recordable macro" facility is one of these, as is a regular-expression search and replace. The important thing about these more powerful (and more abstract) methods is that they should not be the most exposed methods of accomplishing the task. This is why emacs has the non-regexp version of search assigned to the easy-to-remember "C-s" key.

7. The principle of focus

-- Some aspects of the UI attract attention more than others do

The human eye is a highly non-linear device. For example, it possesses edge-detection hardware, which is why we see Mach bands whenever two closely matched areas of color come into contact. It also has motion-detection hardware. As a consequence, our eyes are drawn to animated areas of the display more readily than static areas. Changes to these areas will be noticed readily.

The mouse cursor is probably the most intensely observed object on the screen -- it's not only a moving object, but mouse users quickly acquire the habit of tracking it with their eyes in order to navigate. This is why global state changes are often signaled by changes to the appearance of the cursor, such as the well-known "hourglass cursor". It's nearly impossible to miss.

The text cursor is another example of a highly eye-attractive object. Changing its appearance can signal a number of different and useful state changes.

8. The principle of grammar

-- A user interface is a kind of language -- know what the rules are

Many of the operations within a user interface require both a *subject* (an object to be operated upon), and a *verb* (an operation to perform on the object). This naturally suggests that actions in the user interface form a kind of grammar. The grammatical metaphor can be extended quite a bit, and there are elements of some programs that can be clearly identified as adverbs, adjectives and such.

The two most common grammars are known as "Action->Object" and "Object->Action". In Action->Object, the operation (or tool) is selected first. When a subsequent object is chosen, the tool immediately operates upon the object. The selection of the tool persists from one operation to the next, so that many objects can be operated on one by one without having to re-select the tool. Action->Object is also known as "modality", because the tool selection is a "mode" which changes the operation of the program. An example of this style is a paint program -- a tool such as a paintbrush or eraser is selected, which can then make many brush strokes before a new tool is selected.

In the Object->Action case, the object is selected first and persists from one operation to the next. Individual actions are then chosen which operate on the currently selected object or objects. This is the method seen in most word processors -- first a range of text is selected, and then a text style such as bold, italic, or a font change can be selected. Object->Action has been called "non-modal" because all behaviors that can be applied to the object are always available. One powerful type of Object->Action is called "direct manipulation", where the object itself is a kind of tool -- an example is dragging the object to a new position or resizing it.

Modality has been much criticized in user-interface literature because early programs were highly modal and had hideous interfaces. However, while non-modality is the clear winner in many situations, there are a large number of situations in life that are clearly modal. For example, in carpentry, its generally more efficient to hammer in a whole bunch of nails at once than to hammer in one nail, put down the hammer, pick up the measuring tape, mark the position of the next nail, pick up the drill, etc.

9. The principle of help

-- Understand the different kinds of help a user needs

In an essay in [LAUR91] it states that are five basic types of help, corresponding to the five basic questions that users ask:

- **1. Goal-oriented:** "What kinds of things can I do with this program?"
- **2. Descriptive:** "What is this? What does this do?"
- **3. Procedural:** "How do I do this?"
- **4. Interpretive:** "Why did this happen?"
- **5. Navigational:** "Where am I?"

The essay goes on to describe in detail the different strategies for answering these questions, and shows how each of these questions requires a different sort of help interface in order for the user to be able to adequately phrase the question to the application.

For example, "about boxes" are one way of addressing the needs of question of type 1. Questions of type 2 can be answered with a standard "help browser", "tool tips" or other kinds of context-sensitive help. A help browser can also be useful in responding to questions of the third type, but these can sometimes be more efficiently addressed using "cue cards", interactive "guides", or "wizards" which guide the user through the process step-by-step. The fourth type has not been well addressed in current applications, although well-written error messages can help. The fifth type can be answered by proper overall interface design, or by creating an application "roadmap". None of the solutions listed in this paragraph are final or ideal; they are simply the ones in common use by many applications today.

10. The principle of safety

-- Let the user develop confidence by providing a safety net

Ted Nelson once said "Using DOS is like juggling with straight razors. Using a Mac is like shaving with a bowling pin."

Each human mind has an "envelope of risk", that is to say a minimum and maximum range of risk-levels which they find comfortable. A person who finds herself in a situation that is too risky for her comfort will generally take steps to reduce that risk. Conversely, when a person's life becomes too safe -- in other words, when the risk level drops below the *minimum* threshold of the risk envelope -- she will often engage in actions that *increase* their level of risk.

This comfort envelope varies for different people and in different situations. In the case of computer interfaces, a level of risk that is comfortable for a novice user might make a "power-user" feel uncomfortably swaddled in safety.

It's important for new users that they feel safe. They don't trust themselves or their skills to do the right thing. Many novice users think poorly not only of their technical skills, but of their intellectual capabilities in general (witness the popularity of the "...for Dummies" series of tutorial books.) In many cases these fears are groundless, but they need to be addressed. Novice users need to be assured that they will be protected from their own lack of skill. A program with no safety net will make this type of user feel uncomfortable or frustrated to the point that they may cease using the program. The "Are you sure?" dialog box and multi-level undo features are vital for this type of user.

At the same time, an expert user must be able to use the program as a *virtuoso*. She must not be hampered by guard rails or helmet laws. However, expert users are also smart enough to turn off the safety checks -- if the application allows it. This is why "safety level" is one of the more important application configuration options.

Finally, it should be noted that many things in life are not meant to be easy. Physical exercise is one -- "no pain, no gain". A concert performance in Carnegie Hall, a marathon, or the Guinness World Record would be far less impressive if anybody could do it. This is especially pertinent in the design of computer game interfaces, which operate under somewhat different principles than those listed here (although many of the principles in fact do apply).

11. The principle of context

-- Limit user activity to one well-defined context unless there's a good reason not to

Each user action takes place within a given context -- the current document, the current selection, the current dialog box. A set of operations that is valid in one context may not be valid in another. Even within a single document, there may be multiple levels -- for example, in a structured drawing application, selecting a text object (which can be moved or resized) is generally considered a different state from selecting an individual character within that text object.

It's usually a good idea to avoid mixing these levels. For example, imagine an application that allows users to select a range of text characters within a document, and also allows them to select one or more whole documents (the latter being a distinct concept from selecting all of the characters in a document). In such a case, it's probably best if the program disallows selecting both characters and documents in the same selection. One unobtrusive way to do this is to "dim" the selection that is not applicable in the current context. In the example above, if the user had a range of text selected, and then selected a document, the range of selected characters could become dim, indicating that the selection was not currently pertinent. The exact solution chosen will of course depend on the nature of the application and the relationship between the contexts.

Another thing to keep in mind is the relationship between contexts. For example, it is often the case that the user is working in a particular task-space, when suddenly a dialog box will pop up asking the user for confirmation of an action. This sudden shift of context may leave the user wondering how the new context relates to the old. This confusion is exacerbated by the terseness of writing style that is common amongst application writers. Rather than the "Are you sure?" confirmation mentioned earlier, something like "There are two documents unsaved. Do you want to quit anyway?" would help to keep the user anchored in their current context.

12. The principle of aesthetics

-- Create a program of beauty

It's not necessary that each program be a visual work of art. But it's important that it not be ugly. There are a number of simple principles of graphical design that can easily be learned, the most basic of which was coined by artist and science fiction writer William Rotsler: "Never do anything that looks to someone else like a mistake." The specific example Rotsler used was a painting of a Conan-esque barbarian warrior swinging a mighty broadsword. In this picture, the tip of the broadsword was just off the edge of the picture. "What that looks like", said Rotsler, "is a picture that's been badly cropped. They should have had the tip of the sword either clearly within the frame or clearly out of it."

An interface example can be seen in the placement of buttons -- imagine five buttons, each with five different labels that are almost the same size. Because the buttons are packed using an automated-layout algorithm, each button is almost but not exactly the same size. As a result, though the author has placed much care into his layout, it looks carelessly done. A solution would be to have the packing algorithm know that buttons that are almost the same size look better if they are exactly the same size -- in other words, to encode some of the rules of graphical design into the layout algorithm. Similar arguments hold for manual widget layout.

Another area of aesthetics to consider is the temporal dimension. Users don't like using programs that feel sluggish or slow. There are many tricks that can be used to make a slow program "feel" snappy, such as the use of off-screen bitmaps for rendering, which can then be blitted forward in a single operation. (A pet peeve of this particular author is buttons that *flicker* when the button is being activated or the window is being resized. Multiply redundant refreshing of buttons when changing state is one common cause of this.)

13. The principle of user testing

-- Recruit help in spotting the inevitable defects in your design

In many cases a good software designer can spot fundamental defects in a user interface. However, there are many kinds of defects which are not so easy to spot, and in fact an experienced software designer is often less capable of spotting them than the average person. In other cases, a bug can only be detected while watching someone else use the program.

User-interface testing, that is, the testing of user-interfaces using actual end-users, has been shown to be an extraordinarily effective technique for discovering design defects. However, there are specific techniques that can be used to maximize the effectiveness of end-user testing. These are outlined in both [TOG91] and [LAUR91] and can be summarized in the following steps:

- Set up the observation. Design realistic tasks for the users, and then recruit end-users that have the same experience level as users of your product (Avoid recruiting users who are familiar with your product however).
- Describe to the user the purpose of the observation. Let them know that you're testing the product, not them, and that they can quit at any time. Make sure that they

understand if anything bad happens, it's not their fault, and that it's helping you to find problems.
- Talk about and demonstrate the equipment in the room.
- Explain how to "think aloud". Ask them to verbalize what they are thinking about as they use the product, and let them know you'll remind them to do so if they forget.
- Explain that you will not provide help.
- Describe the tasks and introduce the product.
- Ask if there are any questions before you start; then begin the observation.
- Conclude the observation. Tell them what you found out and answer any of their questions.
- Use the results.

User testing can occur at any time during the project, however, it's often more efficient to build a mock-up or prototype of the application and test that before building the real program. It's much easier to deal with a design defect before it's implemented than after. Tognazzini suggests that you need no more than three people per design iteration -- any more than that and you are just confirming problems already found.

14. The principle of humility

-- Listen to what ordinary people have to say

Some of the most valuable insights can be gained by simply watching other people attempt to use your program. Others can come from listening to their opinions about the product. Of course, you don't have to do exactly everything they say. It's important to realize that each of you, user and developer, has only part of the picture. The ideal is to take a lot of user opinions, plus your insights as a developer and reduce them into an elegant and seamless whole -- a design which, though it may not satisfy everyone, will satisfy the greatest needs of the greatest number of people.

One must be true to one's vision. A product built *entirely* from customer feedback is doomed to mediocrity, because what users want most are the features that they cannot anticipate.

But a single designer's intuition about what is good and bad in an application is insufficient. Program creators are a small, and not terribly representative, subset of the general computing population.

Some things designers should keep in mind about their users:

- Most people have a biased idea as to the what the "average" person is like. This is because most of our interpersonal relationships are in some way self-selected. It's a rare person whose daily life brings them into contact with other people from a full range of personality types and backgrounds. As a result, we tend to think that others think "mostly like we do." Designers are no exception.

- Most people have some sort of core competency, and can be expected to perform well within that domain.
- The skill of using a computer (also known as "computer literacy") is actually much harder than it appears.
- The lack of "computer literacy" is not an indication of a lack of basic intelligence. While native intelligence does contribute to one's ability to use a computer effectively, there are other factors which seem to be just as significant, such as a love of exploring complex systems, and an attitude of playful experimentation. Much of the fluency with computer interfaces derives from play -- and those who have dedicated themselves to "serious" tasks such as running a business, curing disease, or helping victims of tragedy may lack the time or patience to be able to devote effort to it.
- A high proportion of programmers are introverts, compared to the general population. This doesn't mean that they don't like people, but rather that there are specific social situations that make them uncomfortable. Many of them lack social skills, and retreat into the world of logic and programming as an escape; As a result, they are not experienced people-watchers.

The best way to avoid misconceptions about users is to spend some time with them, especially while they are actually using a computer. Do this long enough, and eventually you will get a "feel" for how the average non-technical person thinks. This will increase your ability to spot defects, although it will never make it 100%, and will never be a substitute for user-testing.

APPENDIX 3-3

CHECKLIST

CHECKLIST: Evaluating e-Learning User Interfaces

by Kevin Kruse

When reviewing and evaluating the computer interface of your e-learning program, you should be able to answer yes to the questions below.

- Do all buttons and icons have a consistent and unique appearance?
- Are visual cues like mouse cursor changes and roll-over highlights used consistently on all buttons?
- Are buttons labeled with text descriptions (or with roll-over text)?
- Do buttons gray-out or disappear when they are inactive?
- Do *non-button* graphics have their own design properties distinct from that of buttons?
- Are navigation buttons displayed in exactly the same screen position every time they appear?
- Are buttons grouped logically and located where the user is likely to be looking?
- Do users have one-click access to help, exit, and the Main Menu?
- Are users returned to where they left off after closing the help window and canceling out of the exit screen?
- Does every menu have a title?
- Does every menu screen include an option to return to the previous or Main Menu?
- Are there fewer than three levels of menus?
- Do menus have nine or fewer items on them?
- Are items on menus descriptive rather than general?
- Are menu items listed in a sequential or logical order?
- Do menus indicate which items the student has completed?
- Are confirmation messages used in areas such as student registration, exit, and final exams?
- Are there clear instructions associated with menus, questions, and other tasks?
- Are error messages written in plain language?
- Are status messages displayed during delays greater than four seconds?
- Are exclamation points and sound effects used sparingly?

- Is there a bookmarking feature that enables students to exit and resume later where they left off?
- Can students move backward, as well as forward, in linear tutorials?
- Are page or screen counters used to show progress within linear lessons?
- Is the visual metaphor consistent and intuitive in non-linear simulations?
- Are all pop-up windows positioned on the screen so they do not cover up relevant information?
- Does text appear clearly and with normal margins and spacing?
- Do information input screens force all capital letters, and is the evaluation of inputs case insensitive?
- Can users interact with the program from either the keyboard or the mouse?
- Are text fonts used consistently?
- Are audio volume levels consistent?
- Do users have the option to replay video or audio narration?

APPENDIX 3-4

QUESTIONAIRES

Questionnaires for Final Year Project

This questionnaire is part of the survey on the use of UTP E-Learning system in order to detect interface design problems. Please take a moment to fill in this questionnaire.
Thank You.
**interface is the from end of the e-learning system; the one that visible to user vision.*

1. Knowledge in IT field

	1	2	3	4	5	
Bad						Good

2. How often you use UTP e-learning system

	1	2	3	4	5	
Once a month						Everyday

3. How do you find the interface of the e-learning system

	1	2	3	4	5	
Bad						Good

4. Do you think it require improvement/changes

	1	2	3	4	5	
No						Yes

5. You use all the function offered in e-learning system

	1	2	3	4	5	
No						Yes

Assuming that you agreed to have changes for the e-learning system interface

1. Do you think the existing colour need to be change

	1	2	3	4	5	
No						Yes

2. Do you think the existing fonts need to be change

	1	2	3	4	5	
No						Yes

3. Do you think the icons need use need to be change

	1	2	3	4	5	
No						Yes

4. Do you thinks the existing icon confusing

	1	2	3	4	5	
No						Yes

5. Do you need more control on the navigation part

	1	2	3	4	5	
No						Yes

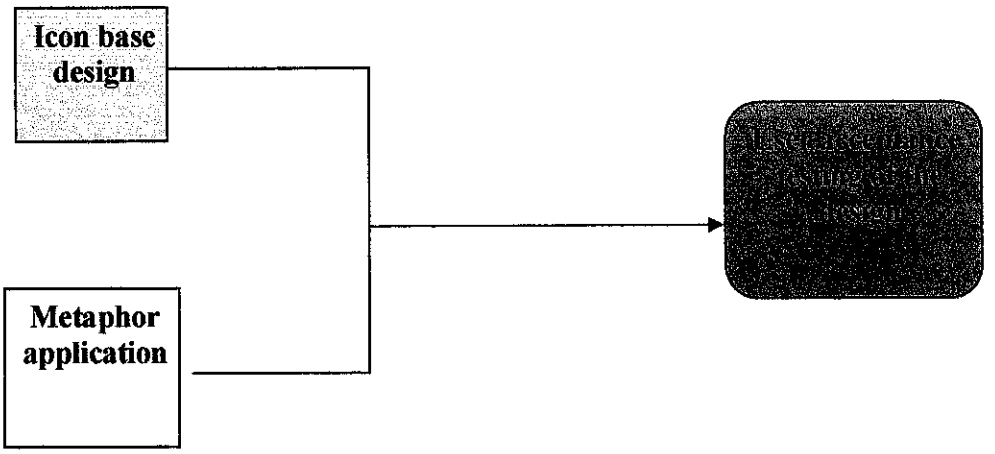
6. Please state in 3 words what you want the new UTP e-learning system to look like

.....

APPENDIX 3-5

THEORITICAL FRAMEWORK

THEORITICAL FRAMEWORK



Ho = There's no relationship between icon base design, metaphor application with level of user acceptance

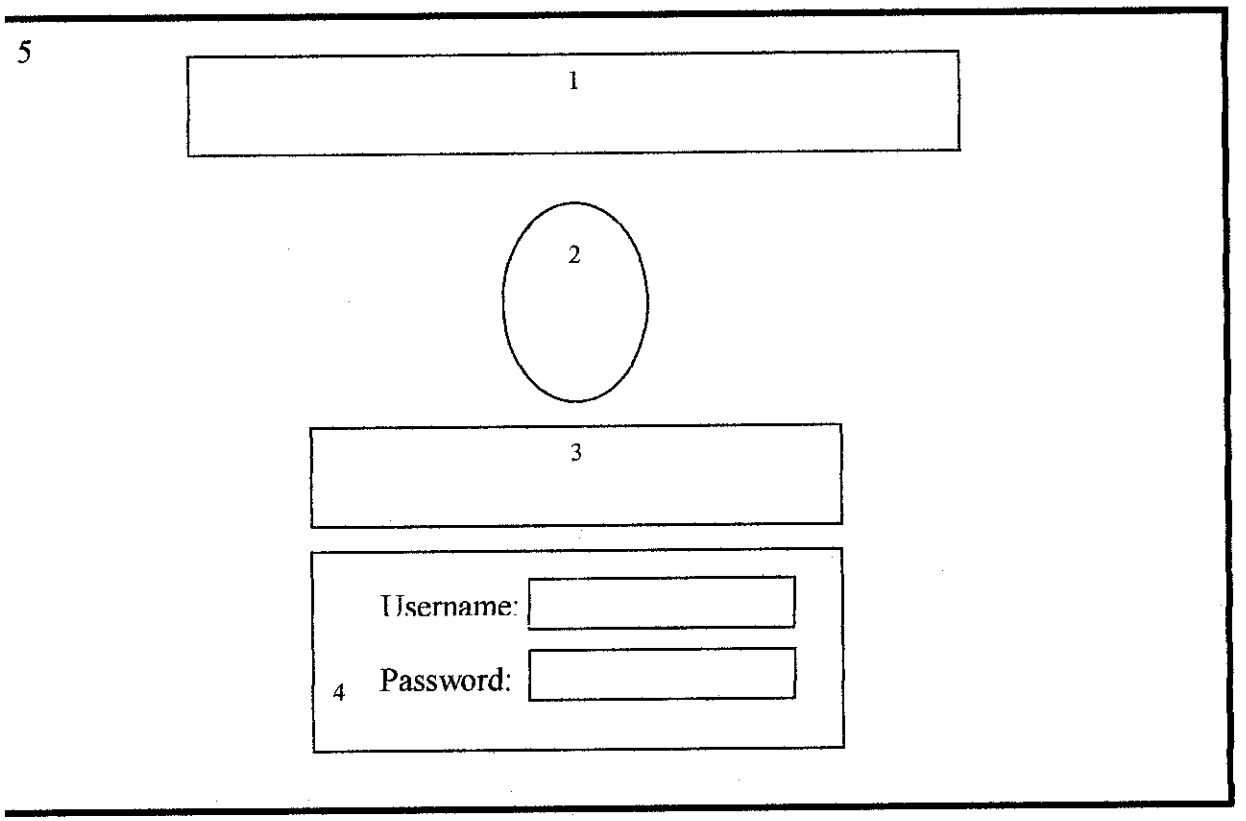
HA = There is relationship between icon base design, metaphor application with level of user acceptance

APPENDIX 3-6

STORYBOARD 1 : Positioning and Specification

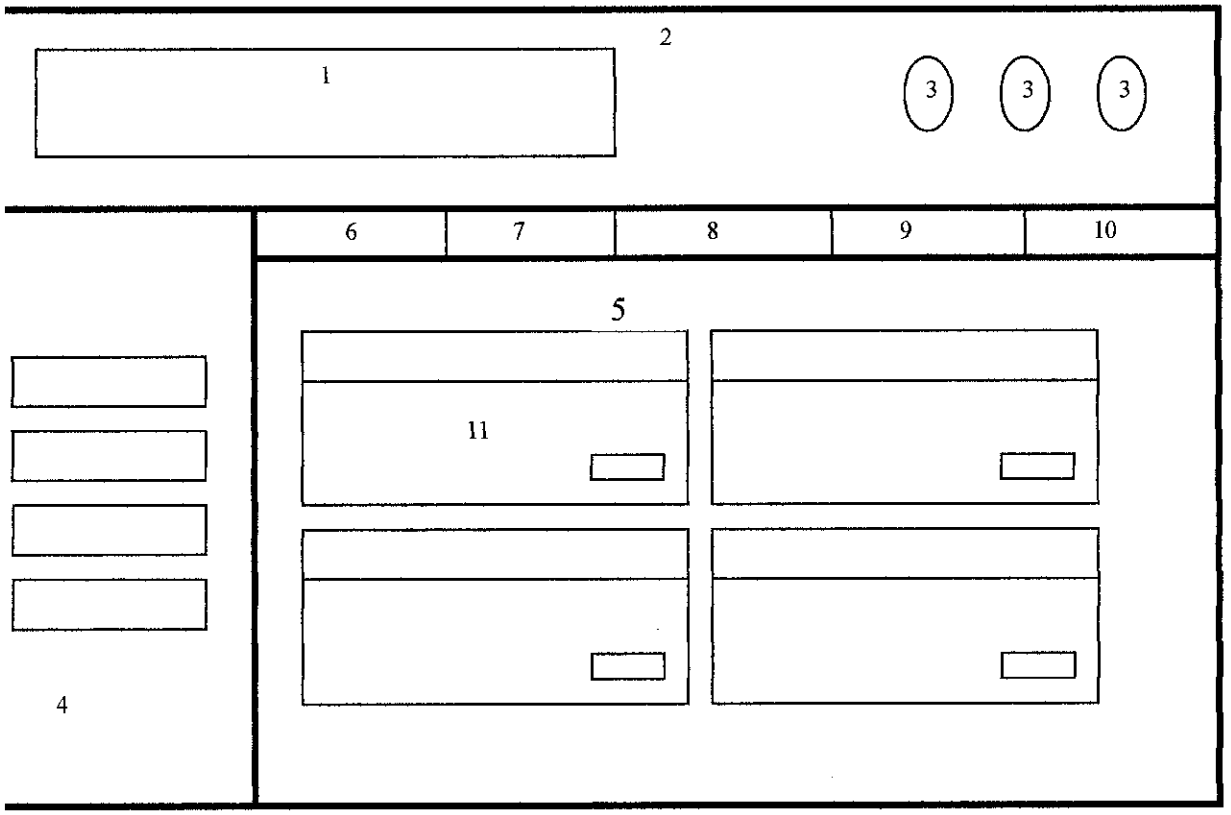
STORYBOARD 1: Positioning and specification

a) The Log-in Page



1. Main Title – The simple title that introduce the whole system.
2. UTP logo – As a way to show the image of UTP
3. Explanation – Simple explanation to describe the usage of the system. Must be simple because users don't want to spend too much time in the login page.
4. Login box – previously the existing system use two page for this purposes but to minimize the page through to complete task the author reduce it to just one page.
5. Background – The background colour for this interface will be from a light colour family and the font use will be from dark colour category in order to avoid user experiencing eye tiredness.

b) The Main Page



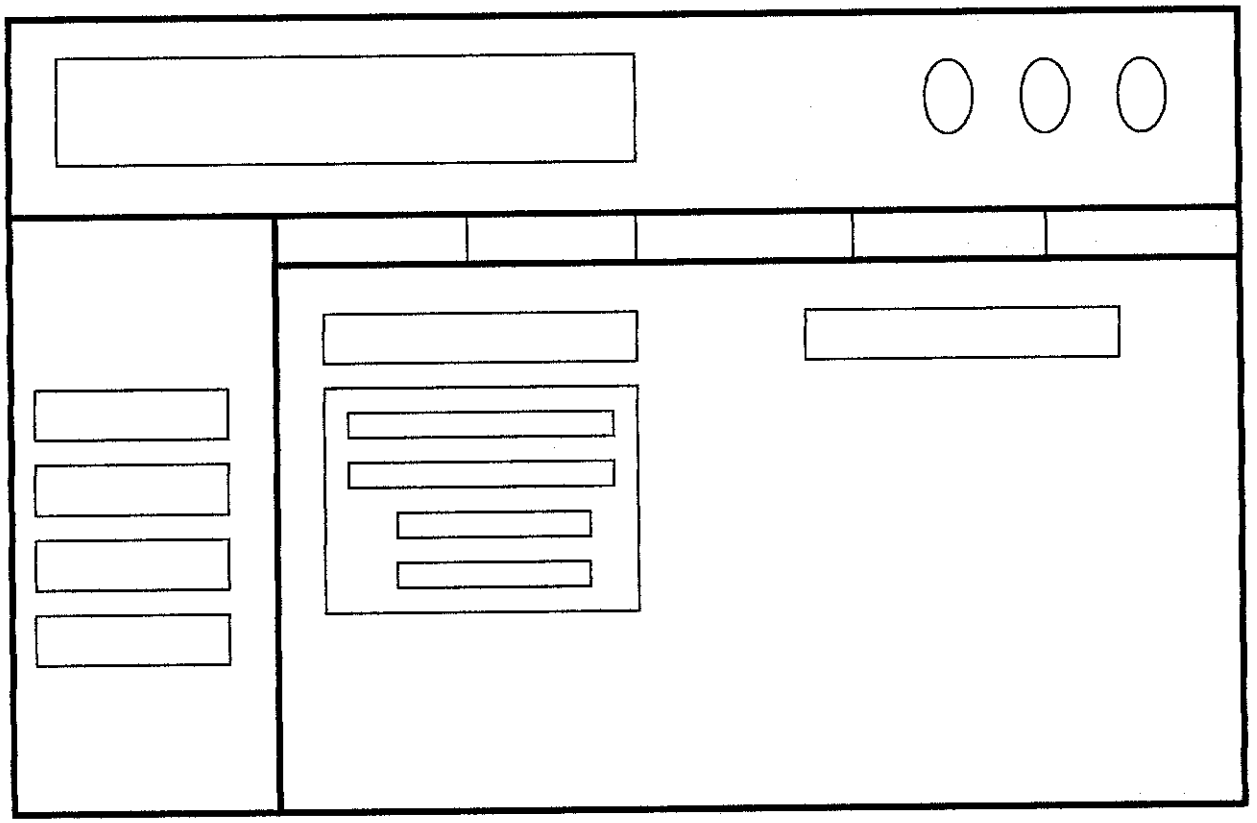
1. Main title – The simple title that introduce the whole system. Font in light colour category.
2. The title bar background – Solid dark colour. This is so in order to make the title stand out from the entire interface.
3. The Basic Icons – Consist of 'Home', 'Help', and 'Log out" icons. The icon will be provided with small text information at the bottom of the icons to avoid confusion.

4. Side bar – The side bar background will be in a lighter colour then the title bar. The author decided to use the same colour with different tone in order to avoid too many colour type used. The appropriate colour used in a good interface will be 3 plus minus 2 (3+-2). There is menu buttons in this side bar. The menu button are :
    - User Directory
    - Address Book
    - Personal Information
    - Web Email
5. Information section (Dynamic section) – Will be in white or any white shade (e.g. off white, apple white etc.)Since this is the main focus of user a colour from alight colour category is use. This is because the author wants to avoid user experiencing eye tiredness.
6. Number 6, 7, 8, 9 and 10 will be having the same specification. This will be a menu bar that consist of :
    - My Main
    - Courses
    - Community
    - Services
    - Academic Web Resources

11. The boxes in the information section (dynamic section) will be consist of boxes with title such as:
    - My Announcement
    - My Task
    - My Course
    - My Calendar

    All this boxes will be containing new or updated information for each title. The box title will be in a darker colour the information area of the boxes. The small boxes at the right corner of the boxes will be a 'More' button in case the user wanted to see the entire information in the box.

c) The Course List



All the other interface design will be as mention above. The only differences will be in the dynamic section.

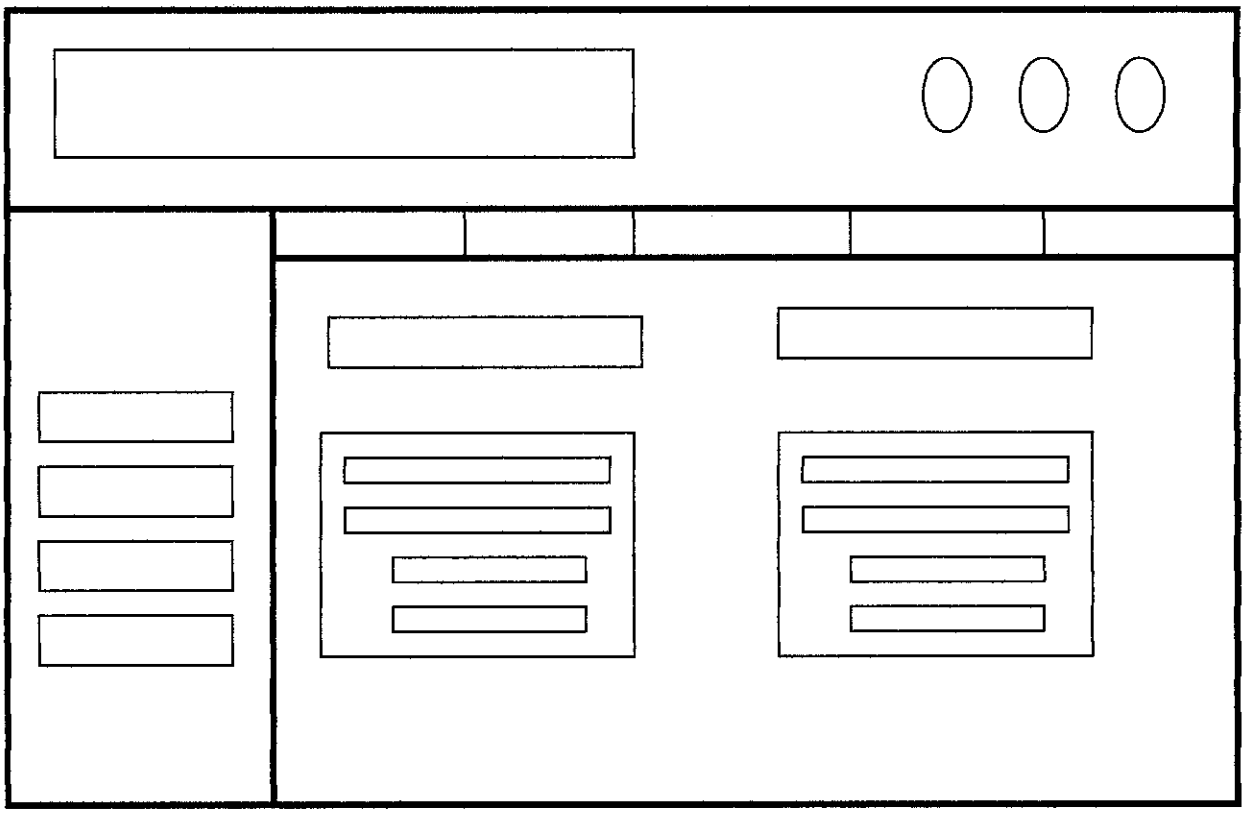
Dynamic section – For this page the dynamic section will be containing two important items which are:

- Course List
  - Course you teaching (lecturer)
  - Course you participate in (student)
- Course catalog – Containing all the list of courses offered in UTP for all programmes.

All this information will be presenting in a graphical drop down menu to help student understanding and avoid confusion.

By clicking on the organization list item or the discussion board item user will be redirect to requested page.

d) Organization List



All the other interface design will be as mention above. The only differences will be in the dynamic section.

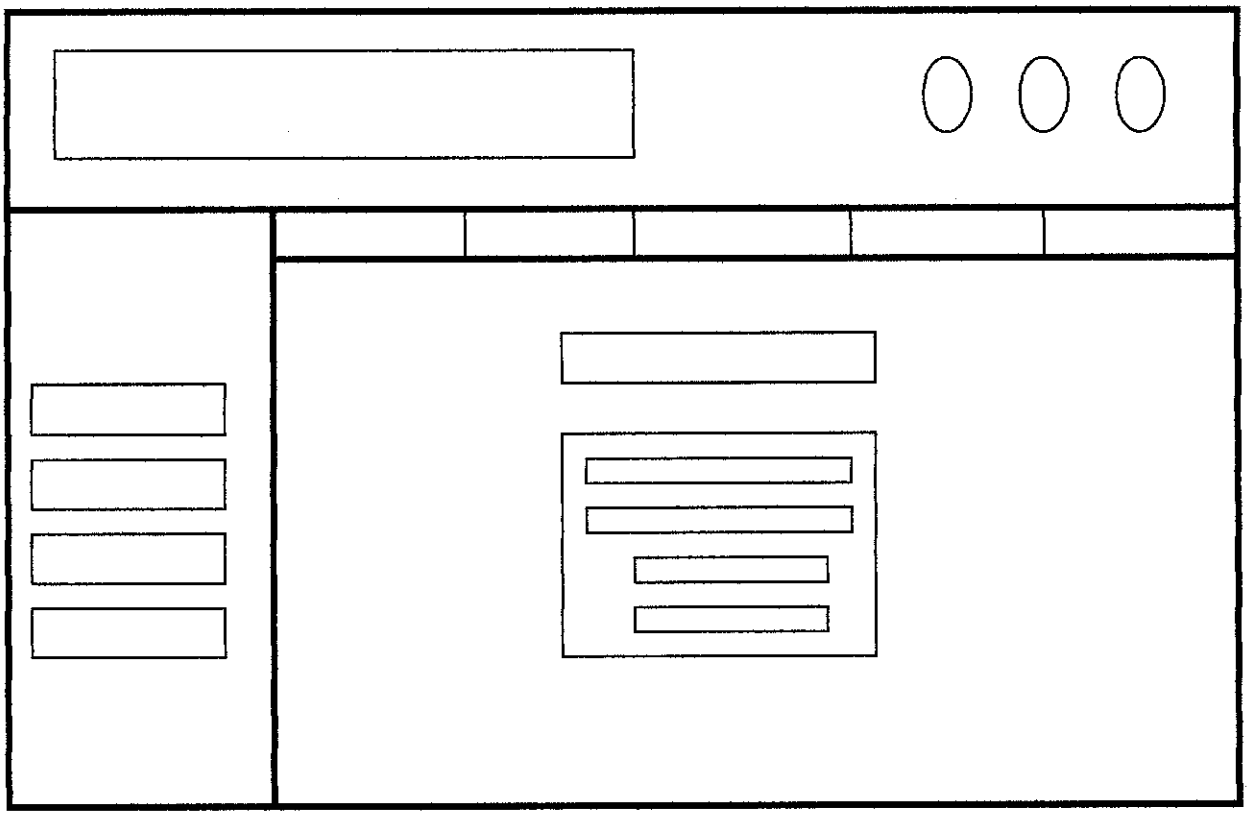
Dynamic section – For this page the dynamic section will be containing two important items which are:

- Organization list
- Discussion Board

All this information will be presenting in a graphical drop down menu to help student understanding and avoid confusion.

By clicking on the organization list item or the discussion board item user will be redirect to requested

e) Services Page



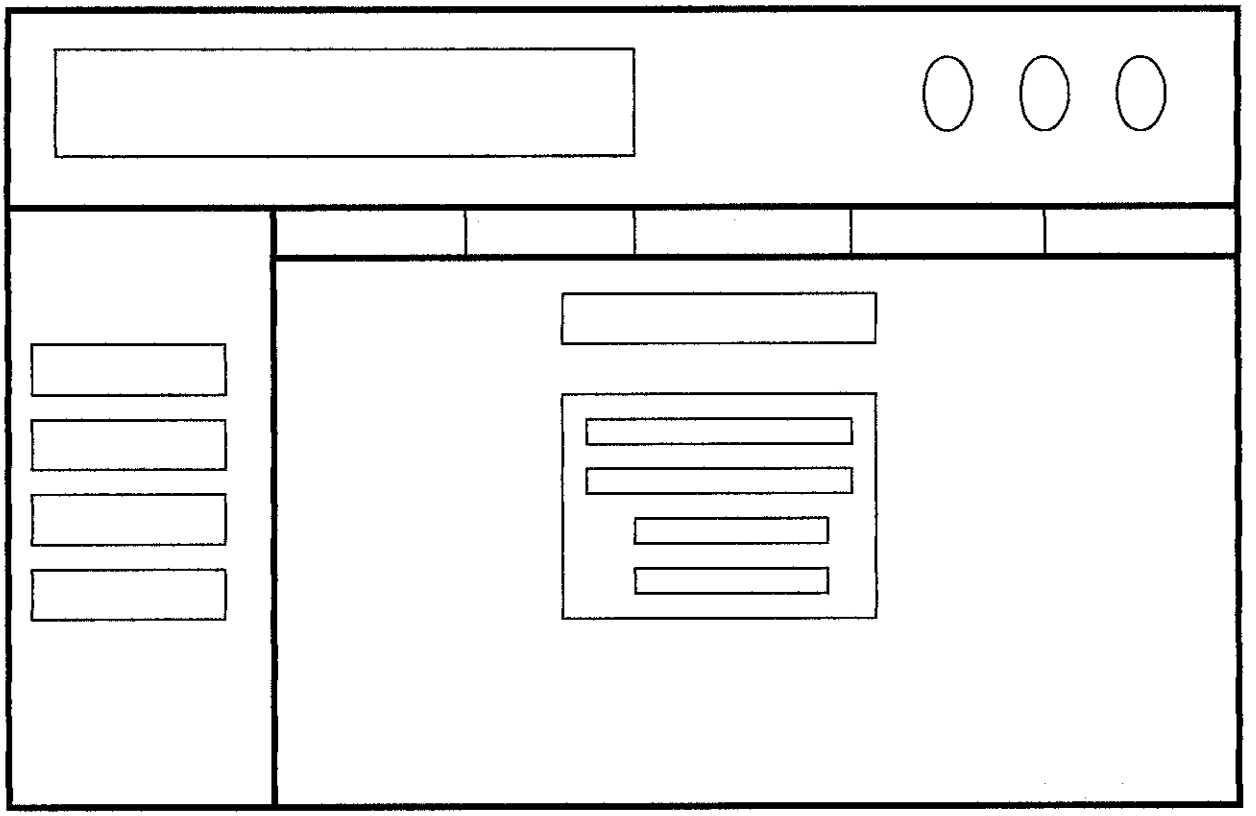
All the other interface design will be as mention above. The only differences will be in the dynamic section.

Dynamic section – For this page the dynamic section will be containing one important item which is:

- List of services provided

All this information will be presenting in a graphical drop down menu to help student understanding and avoid confusion.

f) Academic Web Resources page



All the other interface design will be as mention above. The only differences will be in the dynamic section.

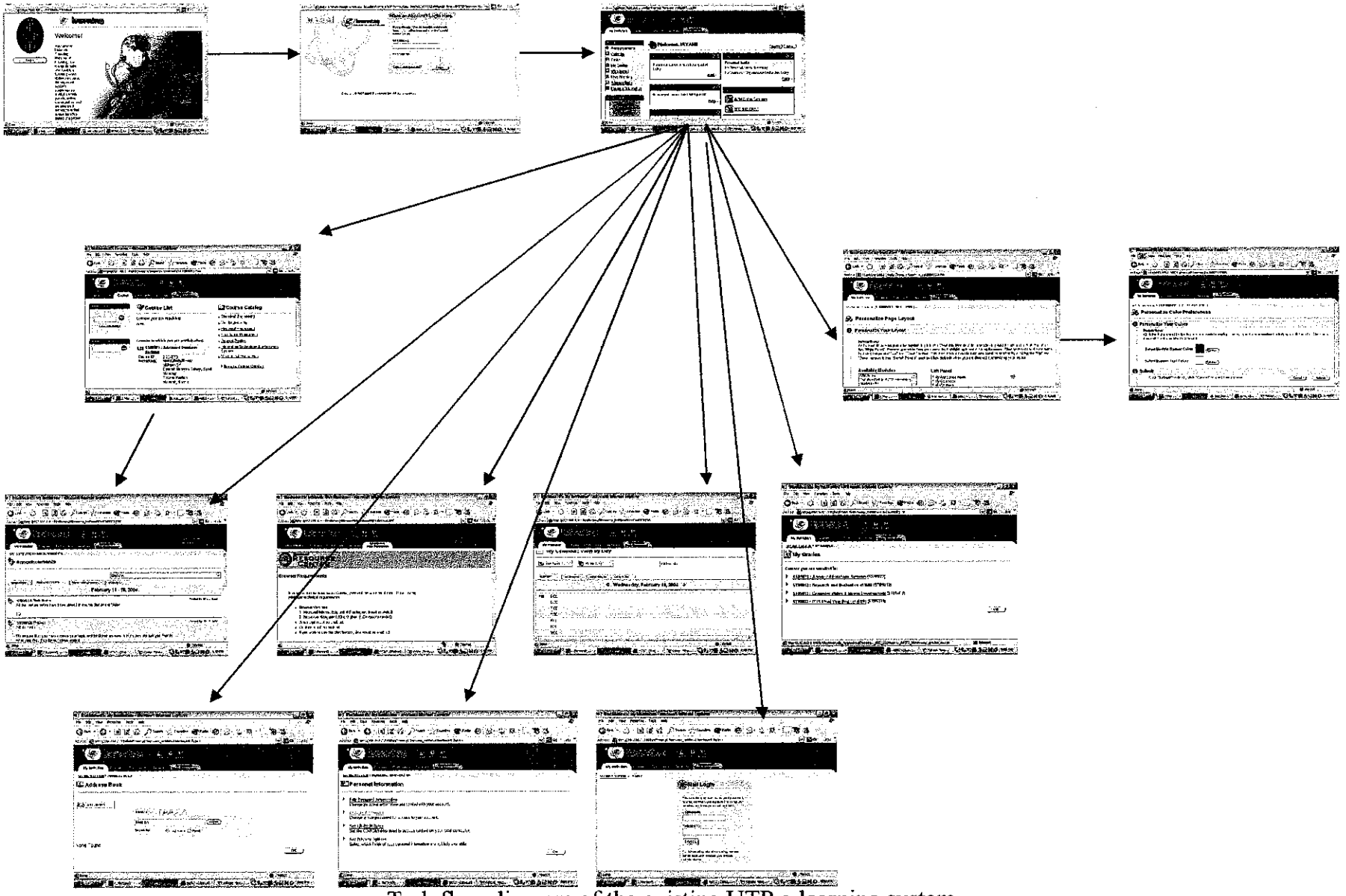
Dynamic section – For this page the dynamic section will be containing one important item which is:

- List of services provided

All this information will be presenting in a graphical drop down menu to help student understanding and avoid confusion.

APPENDIX 3-7

TASK FLOW OF EXISTING INTERFACE



Task flow diagram of the existing UTP e-learning system

APPENDIX 3-8

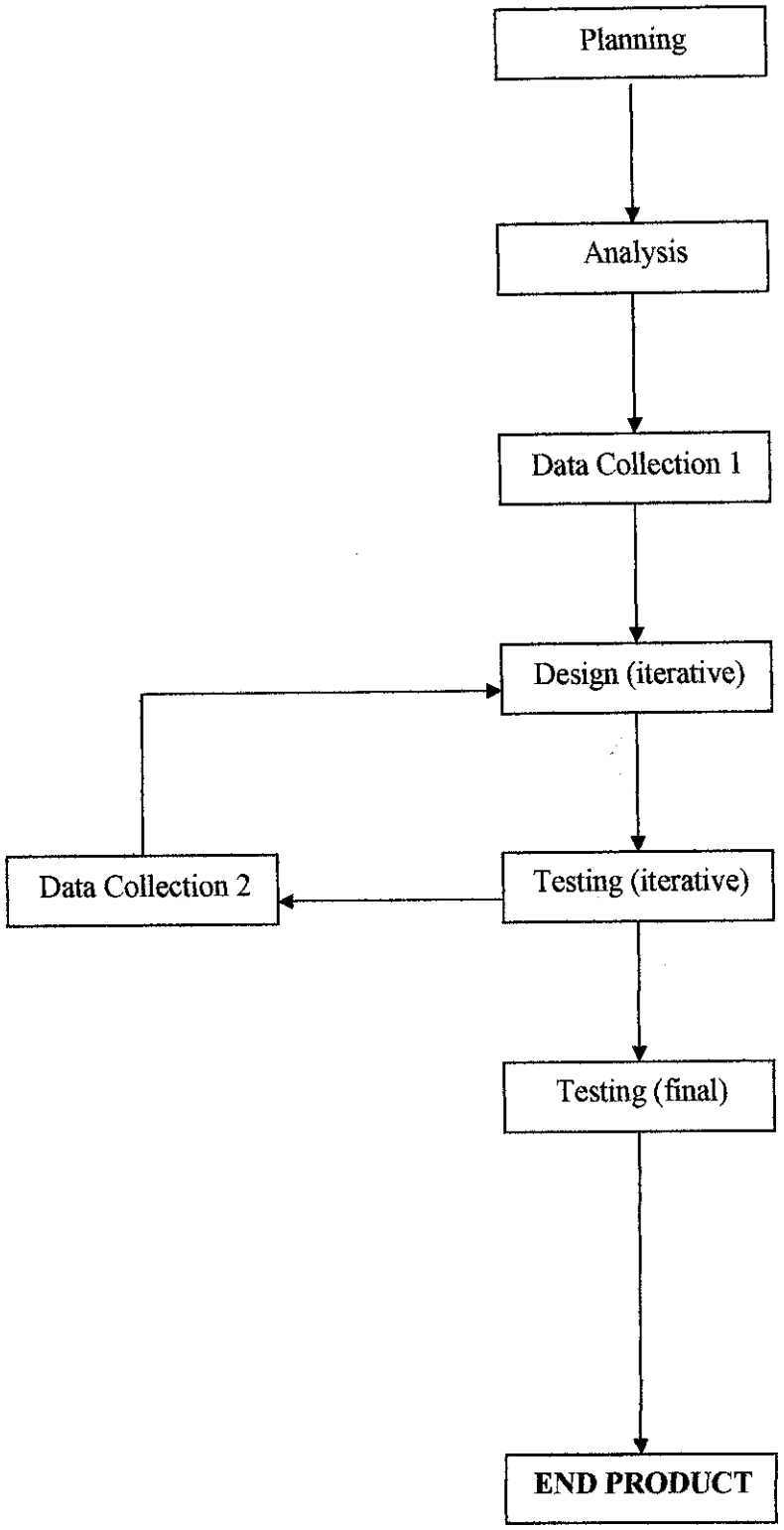
TEST SCRIPT

User Acceptance Testing
Test Script
Testing (Iterative / Final) Date:

No	Test Script	Scale					Remarks
		1	2	3	4	5	
1	Did the interface immediately capture your attention?						
2	Was the creative colour use engaging and appealing?						
3	Was it easy to navigate the interfaces? Are the buttons' functionality easily identified?						
4	Did you ever feel confused of the function provided?						
5	Was the search page interactive and engaging?						
6	Are the buttons' functionality easily identified?						
7	Was it acceptable for learning environment						
8	Did you need help to navigate such interface?						

APPENDIX 3-9

PROJECT METHODOLOGY



User Interface for Higher Education Usage project methodology

APPENDIX 3-10

PROJECT TIMELINE

Project Timeline
User Interface Design for Higher Education Usage

NO	TASK NAME	TIMELINE(WEEK)													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Proposal Submission														
2	Preliminary Research														
3	Preliminary Report Submission														
	PROJECT WORK														
4	Requirement Planning														
5	Analysis Phase														
6	Data Collection 1														
7	Design Prototype(iterative)														
8	Testing the Prototype(iterative)														
9	Data Collection 2														
10	Testing Final														
11	Progress Report Submission														
12	Supervisor Final Draft Submission														
13	Final Draft Submission														
14	Oral Presentation														
15	Project Dissertation Submission														

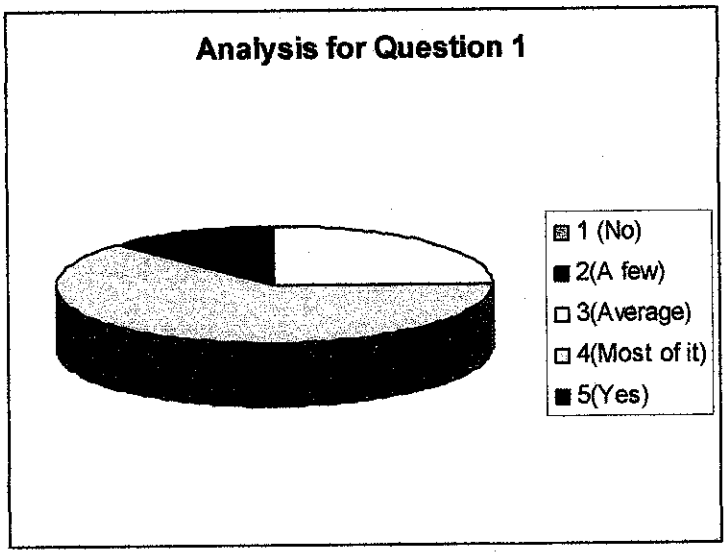
APPENDIX 4-1

DETAILED ANALYSIS OF USER ACCEPTANCE TESTING 2

Detailed Analysis of User Acceptance Testing 2

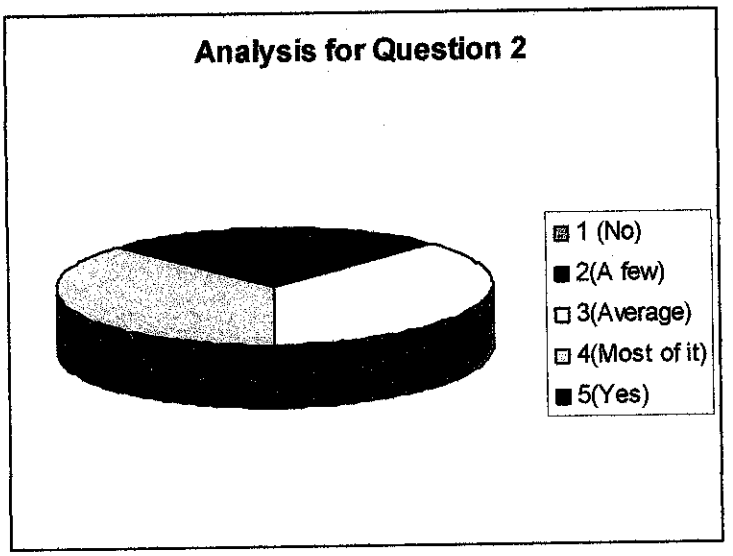
Analysis for Question 1

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	2
4(Most of it)	5
5(Yes)	1



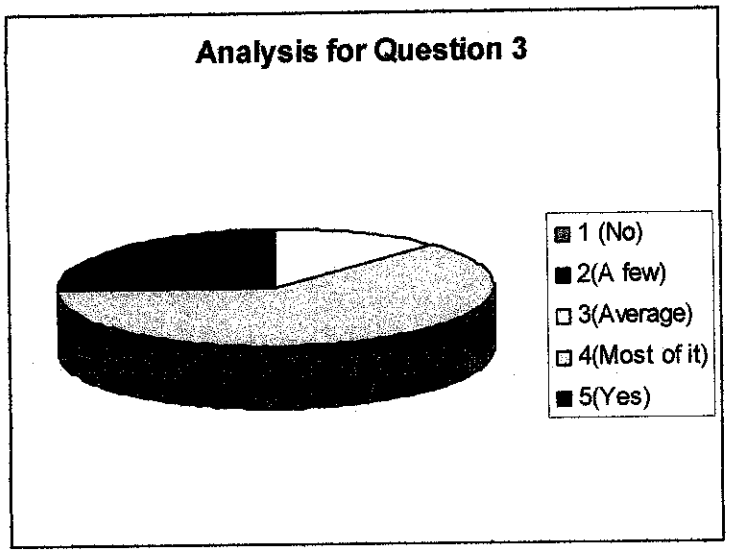
Analysis for Question 2

Answer Rating	Result
1 (No)	0
2(A few)	1
3(Average)	3
4(Most of it)	3
5(Yes)	1



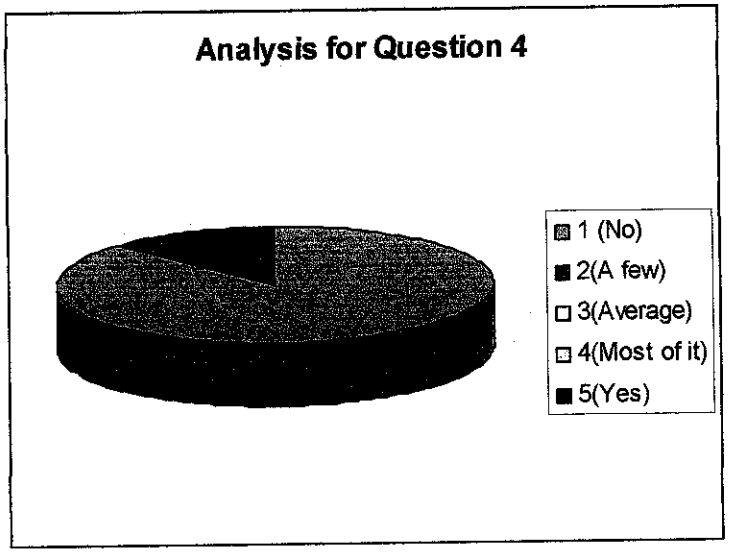
Analysis for Question 3

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	1
4(Most of it)	5
5(Yes)	2



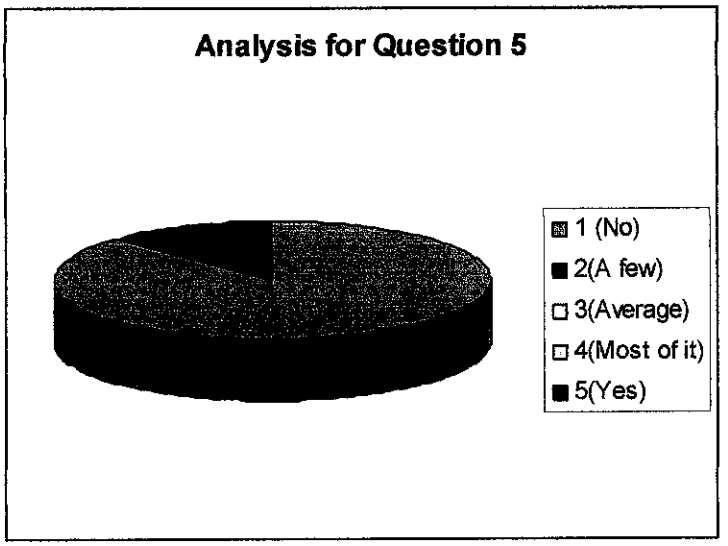
Analysis for Question 4

Answer Rating	Result
1 (No)	7
2(A few)	1
3(Average)	0
4(Most of it)	0
5(Yes)	0



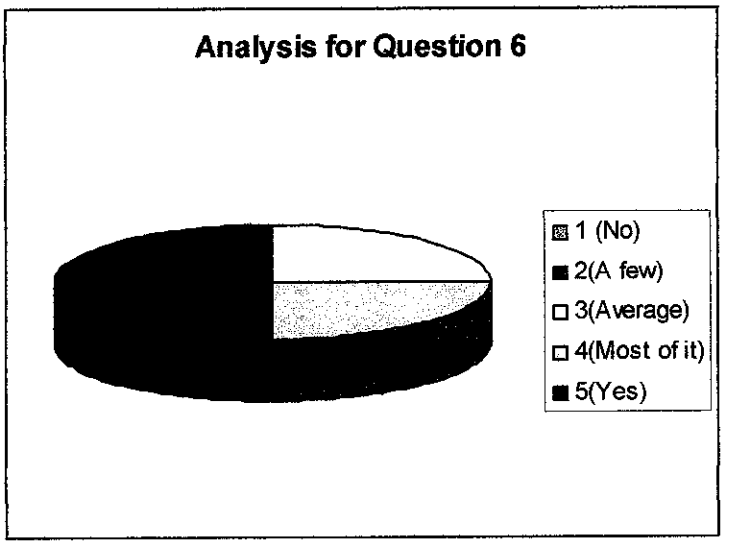
Analysis for Question 5

Answer Rating	Result
1 (No)	7
2(A few)	1
3(Average)	0
4(Most of it)	0
5(Yes)	0



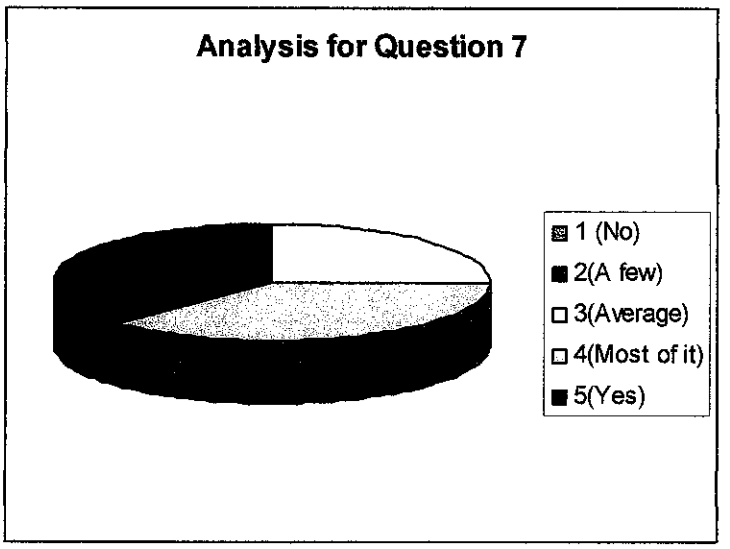
Analysis for Question 6

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	2
4(Most of it)	2
5(Yes)	4



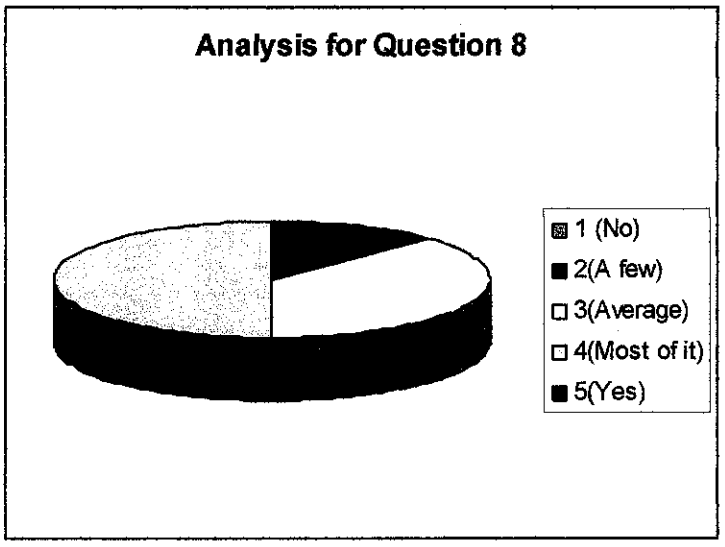
Analysis for Question 7

Answer Rating	Result
1 (No)	0
2(A few)	0
3(Average)	2
4(Most of it)	3
5(Yes)	3



Analysis for Question 8

Answer Rating	Result
1 (No)	0
2(A few)	1
3(Average)	3
4(Most of it)	4
5(Yes)	0


Analysis for Question 8
1 (No)
2(A few)
3(Average)
4(Most of it)
5(Yes)

APPENDIX 4-2

SAMPLE CODING

```
on (release) {
	gotoAndStop(1);

}
on (release) {
	getURL("C:\Documents and Settings\Iryani\My Documents\fyp interface project\Coursepage.html");
}
on (release) {
	getURL("");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\orgpage.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\servicespage.html");
}
on (release) {
	getURL("");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\acadwebpage.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\mycalenderpage.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\announcementpage.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\mytaskpage.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\helppage.html");
}
on (release) {
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\searchpagereal.html");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\personal info page.html");
}
on (release) {
	getURL("");
}
on (release) {
	getURL("C:\\Documents and Settings\\Iryani\\My Documents\\fyp interface project\\addressbookpage.html");
}
function onClick(btn) {
	if (btn == submit_btn) {
		getResults();
		gotoAndStop("pg2");
	} else if (btn == return_btn) {
		gotoAndStop("pg1");
	}
}

function getResults() {
	sweepstakes_result = sweepstakes_box.getValue();
	color_result1 = color_box.getSelectedItem().label;
	selectedItem = color_box.getSelectedIndex();
}
function getResults() {
	sweepstakes_result = sweepstakes_box.getValue();
	color_result2 = color_box2.getSelectedItem().label;
	selectedItem = color_box2.getSelectedIndex();
}
function initValues() {
	sweepstakes_box.setValue(sweepstakes_result);
	if (!started) {
		color_box.setSelectedIndex(0);
		started = true;
	} else {
```

```
                color_box.setSelectedIndex(selectedItem);
        }
}


function initValues2() {
        sweepstakes_box.setValue(sweepstakes_result);
        if (!started) {
                color_box2.setSelectedIndex(0);
                started = true;
        } else {
                color_box2.setSelectedIndex(selectedItem);
        }
}
```