SIRI INDEKS AL-QURAN : SURAH AN-NAS

(SIASA)

By

Rose Zaimi bt Mohamed Zaid
WEK 990276

Under the supervision of

Puan Raja Jamilah Raja Yusof

Moderated by

En Khalit Othman

A Project Submitted to the
Faculty of Computer Science and Information Technology
University of Malaya
Session 2001/2002

Abstract

Siri Index Al-Quran : Surah An-Nas (SIASA) is an interactive multimedia software which function is to provide an indexing of the Surah An-Nas by conducting search using users selection. Not only that, SIASA will also cater other functions such as display an introduction to the Al-Quran and the surah An-Nas, display the surah An-Nas in its Arabic transcript, provide a few translation and tafseer of the surah An-Nas and provide recitation of the surah An-Nas by a few qari.

This report will give an overview on steps that have been taken into measure to ensure the success of this project. This report is divided into five parts. The first part is an introduction to the project. This section will include topics such as project definition, project objectives, scope of project, target users and project schedule.

The second part of this report is the literature review. In this section, a few relevant topics to this project were picked out to be analyze to enhance knowledge on what have been done by other people. Methodologies that have been adopted to develop this project will be covered in the third part of this report.

The fourth part of this report is the system analysis. This section will consist an introduction, task analysis, functional requirements and external interface requirements. The fifth part is the system design. Here, the system functionalities, database design and user interface design will be discussed.

The last three parts will discuss on techniques and approach that will be adopted to implement, test and evaluate the system.

Acknowledgement

Praise and thankful to Allah SWT for the blessing bestowed upon me that enabled me to complete my project. I would like to take this opportunity to express my sincere appreciation and gratitude to the entire individuals that had given me their support and inspiration in completing this project.

I would like to convey my highest appreciation to my supervisor, Pn Raja Jamilah Raja Yusof for her guidance, advice and comments throughout my project. Thank you for all the help and time spent discussing the contents and progress of this project.

A special thanks goes to En Mohd Khalit Othman, the moderator of this project for all your thoughtful comments and suggestion during the moderating period of this project.

Many thanks also go to my parents for all the advice and guidance, friends and all those who have helped me directly or indirectly in completing my project and preparing this report.

Table of Contents

List of Figures

List of Tables

Part 1 : Introduction

1.1 *Project Definitions*

Naturally, indexing of the Al-Quran is done through the traditional Al-Quran book. This is a tedious task because the Al-Quran has 114 surah and thousands of translation. If the user wants to locate a certain phrases in a certain surah, he/she must first locate the surah's translation then read it to locate the correct phrases. To help user perform this task in a much easier way, this project will be able to search and display any user input phrases. The search will not only be based on the Arabic transcript of the surah but also the translation and the tafseer of the surah. Furthermore, this software will be able to cater some other functions that will be discuss in details later in this report.

This interactive multimedia software that will be developed would be named the Siri Index Al-Quran : Surah An-Nas or SIASA, as referred throughout the rest of this report.

1.2 *Project Objectives*

The objectives of this project are as follows :

1. Develop an interactive multimedia software with high usability
2. Designing the user interface using the best available design interface guidelines
3. Create an interactive multimedia environment for educational purposes
4. To provide functions such as :
    i. Access to a short introduction of the Al-Quran and surah An-Nas.
    ii. Search the translation of the surah An-Nas
    iii. Listen to different qari reciting the surah An-Nas.
    iv. Access the tafseer of the surah An-Nas.
    v. Gain knowledge on the "hikmah" in the surah An-Nas.

1.3 *Scope of Project*

This project will be emphasizing on the interaction between the user and the system on performing the user's tasks. Development of this project will be focused on developing an interactive user interface and efficiently and precisely conducting the search based on user's input.

This software will be available in two different languages which are either in English or Bahasa Malaysia. The information that will be displayed in this software will be only about surah An-Nas. However, in the introduction a small part of what is the Al-Quran and why was it sent to Muslim will be displayed for user's knowledge.

Other functions that this software will be able to perform are displaying a few translation and tafseer either in English or Bahasa Malaysia from some of the most popular translation available, search the database of translation, tafseer or surah itself base on user's input and play several recitations from different qari. However, the limitation of this software is that it cannot display, search or play any translation, tafseer or recitation other than the surah An-Nas only. Maybe, when the prototype of this software is fully operational other surahs of the Al-Quran could be inserted into this software.

1.4 *Target Users*

This interactive multimedia software is targeted to be used by higher primary and secondary school students, university students, religious teachers and adults. To be able to cater all the psychology characteristic of each category, the user interface of this software must be design so that it will be easy to understand and used, fun, simple and follow all the user interface guidelines available.

1.5 *Project Schedule*

Activities	Jun 2001	Jul 2001	Aug 2001	Sept 2001	Oct 2001	Nov 2001	Dec 2001	Jan 2002	Feb 2002
Preliminary Investigation Phase ¤ Design objectives									
Literature Review									
System Analysis Phase ¤ Study current software ¤ Improvements ¤ Prepare the new model									
System Design Phase ¤ Study available guidelines ¤ User defined interface									
System Development Phase ¤ Coding									
Testing Phase ¤ Operational software interface ¤ Benchmarking									

Figure 1-1 SIASA Project Schedule

Part 2 : Literature Review

2.1 *Introduction*

According to Cooper (1988) '...a literature review uses as its database reports of primary or original scholarship, and does not report new primary scholarship itself. The primary reports used in the literature may be verbal, but in the vast majority of cases reports are written documents. The types of scholarship may be empirical, theoretical, critical/analytic, or methodological in nature. Second a literature review seeks to describe, summarize, evaluate, clarify and/or integrate the content of primary reports.'

The review of relevant literature is nearly always a standard chapter of a thesis or dissertation. The review forms an important chapter in a thesis where its purpose is to provide the background to and justification for the research undertaken (Ref.1). Bruce, who has published widely on the topic of the literature review, has identified six elements of a literature review. These elements comprise a list; a search; a survey; a vehicle for learning; a research facilitator; and a report (Ref.2).

A crucial element of all research degrees is the review of relevant literature. So important is this chapter that its omission represents a void or absence of a major element in research (Ref.3). According to Bourner (1996) there are good reasons for spending time and effort on a review of the literature before embarking on a research project.

These reasons include:

- To identify gaps in the literature
- To avoid reinventing the wheel that can prevent from making the same mistakes as others
- To identify other people working in the same fields
- To increase breadth of knowledge of the project subject area
- To identify opposing views
- To identify information and ideas that may be relevant to the project
- To identify methods that could be relevant to the project

As far as the literature review process goes, ultimately the goal for students is to complete their review in the allocated time and to ensure they can maintain currency in their field of study for the duration of their research (Ref. 6).

To achieve the above objective, several steps are adapted (Ref.7). The steps are as follows :

1. Be organize around and related directly to the thesis that is going to be developed
2. Synthesize results into summary of what is and is not know
3. Identify areas of controversy in the literature
4. Formulate questions areas that need further research
5. Produce a report base on all the findings

In general, this project literature review will be focused on topics that are relevant to the project objectives. The review will cover topics below :

1. Software development model for multimedia software
2. Analysis of existing relevant software
3. Interactive multimedia software as a medium for education
4. Development tools for multimedia software

The items reviewed were books, web documents, magazines, online articles and existing relevant software.

2.2 *A Summary of Software Development Models*

Waterfall Model was first put forward by Royce in 1970(Ref. 8). It is a well-defined development process in which one phase has to be finished before the next phase. The model is very simple to use. The model can be used if the requirement is well understood and defined.

V-Shaped Model is similar to waterfall model. The difference is that each test phase matches each development phase: requirements with system testing, high-level design with integration testing, and detailed design with unit testing.

Rapid Prototyping Model is the technique that helps designers and users to clarify the requirement of the system. A throwaway prototype is developed by designers and is evaluated by users. From feedback of users, designers will understand the system better and improve the prototype.

Incremental Model. The designers develop the software in a number of stages and are able to deliver the product early. At each phase the designers have a goal to deliver certain features to customers. Incremental model is good for fast delivering product to the marker place.

Spiral Model is an iterative approach. The model carefully take risks into account. The designers develop a small part of the project and evaluate the risks. If the risk is low, designers keeps developing more features. For each iteration, there are six steps:

- Determine objectives, alternatives, and constraints.
- Identify and resolve risks.

- Evaluate alternatives.
- Develop deliverables and verify that they are correct.
- Plan the next iteration.
- Commit to an approach for the next iteration.

2.2.1 *Analysis and Synthesis*

Waterfall model is used if the problem is very well understood. Usually, designers use the waterfall model to develop a simple system because it is hard to change if the model is used. If customers come by and ask for changing requirements, designers will have to start from the scratch because there is no fast way to design the system with the new requirement. Also there is a big problem for testing later if there is a change in requirement. The testers do not have well-define set of test cases, so the test is easy to fail. In brief, waterfall model is used if designers have a well-defined list of user requirements.

V-shape model is an improved version of waterfall model. V-shape model does not run into the problem that the software is impossible to be tested because system test, integration test, and unit test are planed ahead. For example, when we plan the requirement, we also plan for system testing. Therefore, when the system is built, we have a whole set of test cases for system testing.

The rapid prototype model is a good model for the project which has unambiguous user requirement. The model will help users to understand what they actually want. A throwaway prototype is developed so that users can realize what

the system like. Human computer interface (HCI) is a big problem in requirement engineering because HCI depends on different user groups. Different user groups have different need or desire for the interface; therefore, designers have to build the prototype so that users can see and feel it. The prototype model is also good for deploying the new technology. Before the technology is used, users are interested in know whether the technology works or not. Therefore, the prototype is a neat way to demonstrate the idea to users or customers.

Incremental model has many advantages over the other techniques. One of advantages is that the system can be developed at several stages. Each stage has its own requirement; usually it has certain features or core of the system. Each stage can use V-shape, prototype or waterfall model to develop the requirement for this stage. Regardless what kind of model is used in each stage, the product with certain features must be done at the end of the stage.

Spiral model is heavily involved in risks management. If you have a project with very high risks, you should use spiral model. Every iteration, you have a chance to evaluate the risks and to forecast whether the project keeps going or stops. For each iteration, similar to incremental model stage, designer can use V-shape, prototype, or waterfall. Spiral model is usually used in the large project such as financial system, wireless cellular communication management system, network management system.

Waterfall model, V-shape model, and rapid prototyping model are three basic models that can be used in small project. Incremental model and spiral model are usually used in the large project. When the large project is divided into well-

defined small project (or phase or stage or iteration), a small project can use waterfall, V-shape or rapid prototyping model. Which model is chosen for development depends on the organization.

As for SIASA, the rapid prototyping and the HCI are chosen to help guide the development stage.

2.3 *Analysis of Existing Relevant Software*

In this chapter, three software and two webs application will be analyze. The purpose of this analysis is to have an overview of what other developer had developed, to have an idea of what the SIASA will look like and to improve the functionalities of the SIASA compare to the other products available. The application that will be analyze are as stated below :

1. Al-Quran CD version 2000
2. The Koran version 5.2
3. The Holy Quran version 7.0
4. Al-Quran Database
5. The Meaning of the Glorious Quran

2.3.1 *Al-Quran CD version 2000*

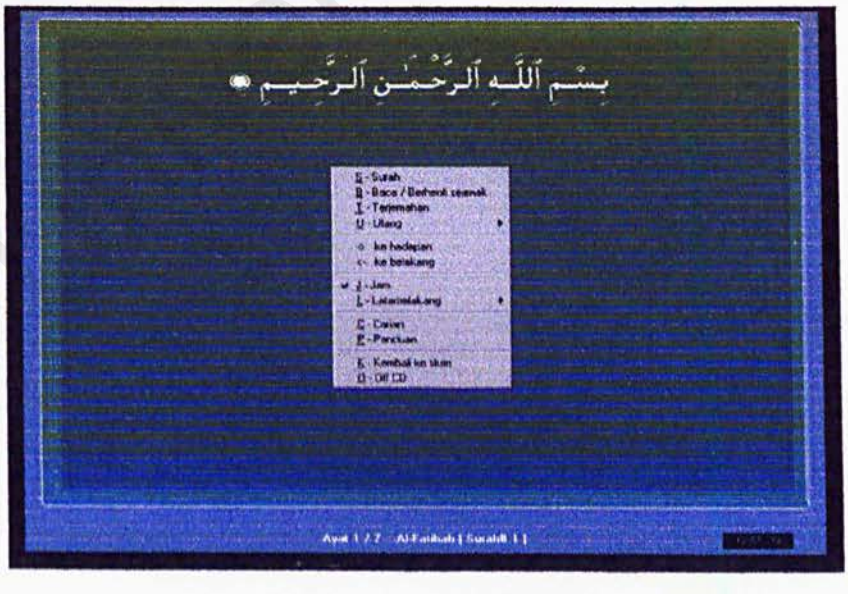

Figure 2-1 Al-Quran CD ver 2000

The figure 2-1 shows the main page of the Al-Quran ver 2000 software. En Muhammad from Negeri Sembilan, Malaysia had developed this software. The scope of this software is the Al-Quran and the language used is Bahasa Malaysia. Functions that are available for the user are display any surah in the Al-Quran, display one translation of each surah selected, listen to a recitation of each surah, display clock, change background, conduct a search on the surah's translation and a help section for user guidance. The advantages of this software are as follows :

1. User can select any surah to be displayed by scrolling down a combo box
2. User will be able to hide or show clock on the screen
3. User can change the background of the screen
4. The color combination used in this software follows the color contrast rules
5. User can perform a search on the surah's translation in the Al-Quran by typing Bahasa Malaysia phrases

Even though this software has many advantages, however some drawbacks were detected which are as follows :

1. A first time user will not know what to do on the first main page until he/she press or click any keyboard button or mouse. Then a drop down menu will appear. There are no instruction for the user on how to start using the software
2. The drop down menu is hide from the user

3. The software only provides one translation and it is only in Bahasa Malaysia
4. Display space on the screen is not fully utilize because only one sentence at a time will be displayed

2.3.2 *The Koran version 5.2*

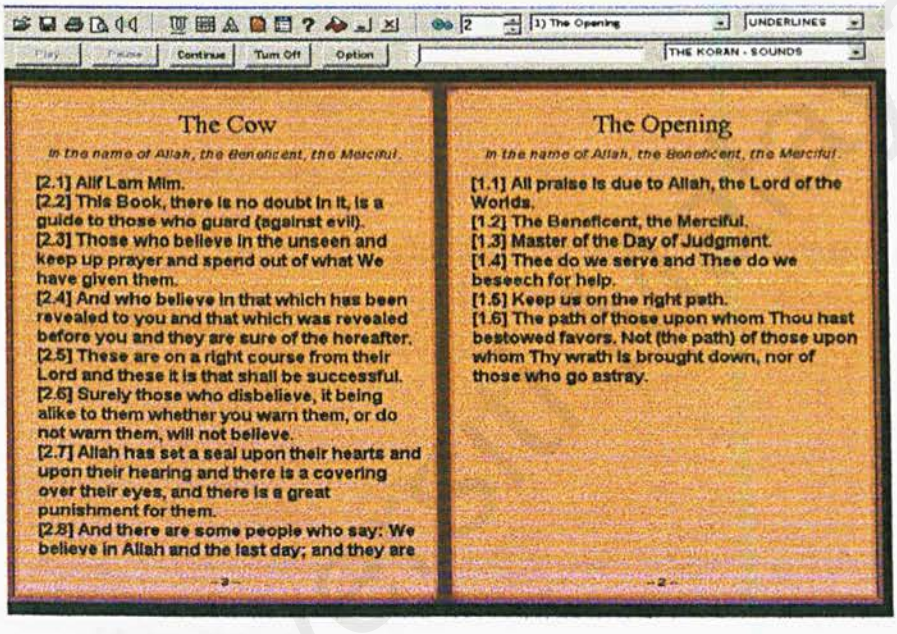

Figure 2-2 The Koran ver 5.2

The figure above shows the interface of the Koran software. This software was created by Samir Alicehajic from Zagred, Croatia. The software covers the whole Al-Quran and uses English as its language. Some of the functions that are available for users are open and save underline data, print any particular surahs

translation, perform a translation search throughout the Al-Quran, listen to recitation of surahs and change text, color and page image.

The advantages from this software are as follows :

1. Selection of surah is by scrolling down the available combo box
2. User can search the Al-Quran translation by typing English phrases
3. User can load their own wav file for the recitation or listen to the existing one
4. User can change the screen background with available images or just by loading a new image
5. User can print any surahs translation
6. User can use the underline function to underline certain phrases in the translation

The disadvantages of this software are :

1. It only display the translation of the Al-Quran and not the original Arabic transcript
2. The selections of surah to be displayed are in the surahs translation name. It is quite difficult for user who does not know the translation name of the surah
3. There are no choices of translation to be selected

2.3.3 *The Holy Quran version 7.0*

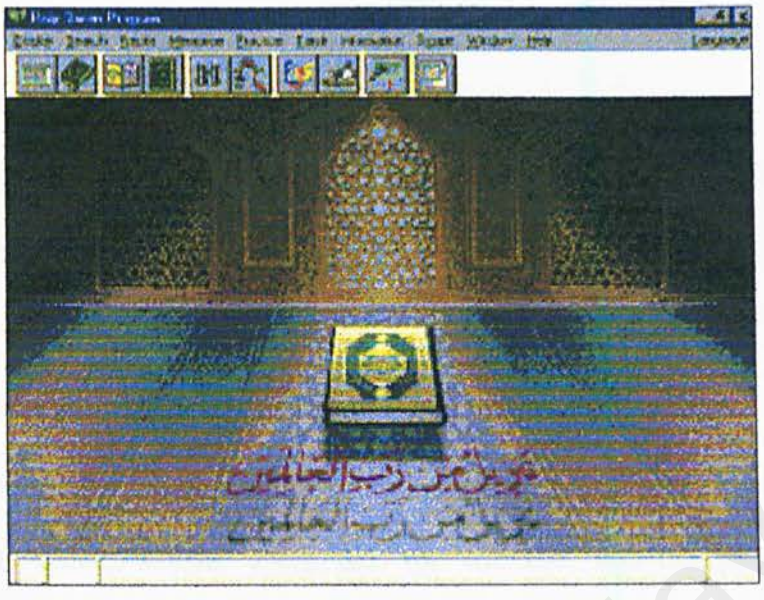

Figure 2-3 The Holy Quran ver 7.0

This software was developed by Sakh. The above figure 2-3 shows the main page of the software. The difference of this software compare to the other two software mentioned in section 2.3.1 and 2.3.2 is that the user can choose the language. It can either be English, Bahasa Malaysia, Turkish, French, Bahasa Indonesia or German. The scope of this software is the Al-Quran. The advantages that can be obtained from this software are as follows :

1. It provides explanation of difficult term in the pronunciations of the Al-Quran
2. User can select any translation from Al-Qurtubi, Al-Tabari, Ibn Khatir and Al-Jalalain
3. This software presents an audio visual display and animation of points and manner of articulation

4. It provide some examples and demonstration of each sound
5. User can search the Al-Quran by phrases, word derivation or even at root level
6. It presents a remarkable of practical explanations of recitation rules
7. Enables the user to record and playback their own versus

The drawbacks of this software are that some of its function is complicated that it may not be suitable for school children, the user must have the proper equipments to record his/her voice into the software and the software cannot compare the users pronunciations with the correct articulation.

2.3.4 *Al-Quran Database*

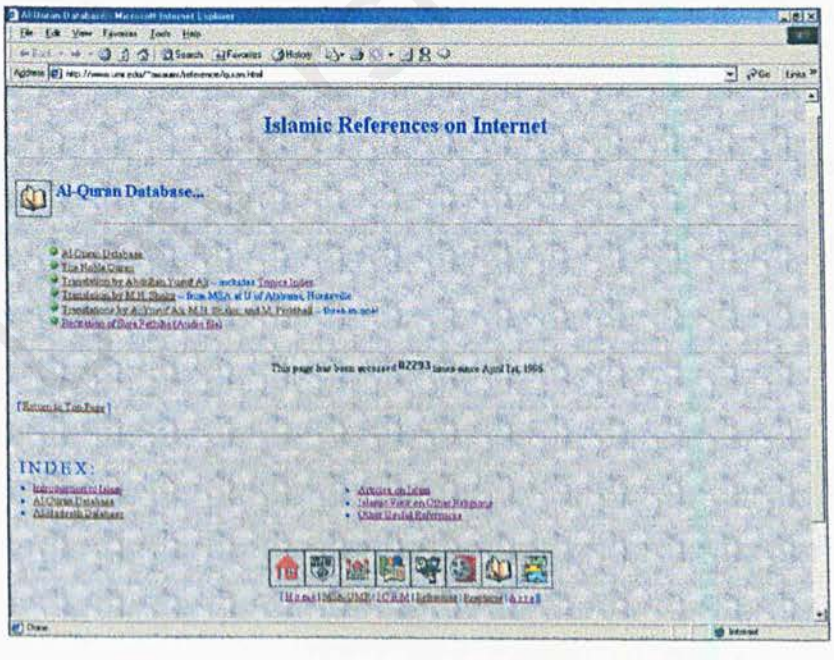

Figure 2-4 Al-Quran Database

The figure 2-4 show the website of the Al-Quran Database. This web application is own by the Muslim Student Organization of Missouri-Rolla (Ref. 23). The scope of this application is Al-Quran and Islam. The language used is English. The advantages of this website are :

1. It provides scripts, recitations and translations from a few different people
2. It provides audio recitation of the translation in English, Urdu and Bangladesh
3. Index and search of the Al-Quran are by topic index in English, French, German, Turkish and Spanish.
4. It provides multilanguage translation of the surahs
5. It provides references of the translation and the information on the website

The disadvantages of this website are it has not been updated for a long time since 1997 and a few of the links does not function properly. It also took quite some time to be uploaded.

2.3.5 *The Meaning of the Glorious Quran.*

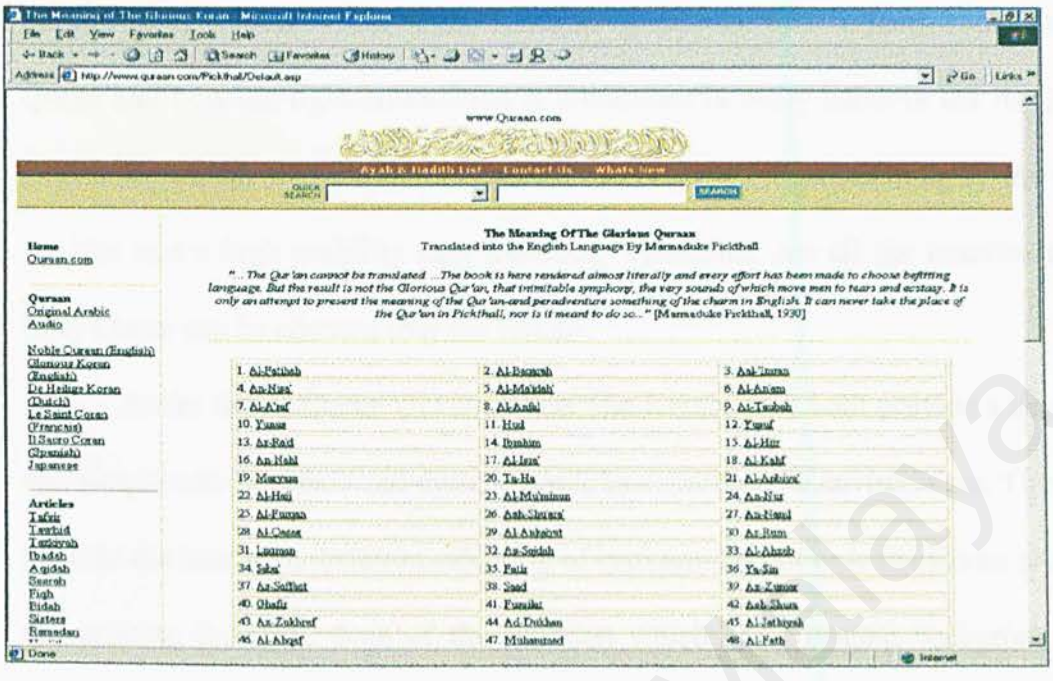

Figure 2-5 The Meaning of the Glorious Quran

The figure 2-5 shows the website of the Meaning of the Glorious Quran. This web application was created by Jawad (Ref.24). This website is in English and covers the Al-Quran and some links on Islam. This website provides Al-Quran transcripts in original Arabic, English, Dutch, Japanese and France. It also has some links on articles such as tafseer, "tauhid", "ibadah", "aqidah" and many more. There are links to hadith from Salih Bukhari, Sahih Muslim and many more. Links of books on various Islamic topics are also available on this website. It also provides an index of the Al-Quran and its translation for easy user navigation.

2.3.6 *Analysis and Synthesis*

In comparison, the Holy Quran software is certainly the best software among the others. It provides most of the function needed for user to learn about the Al-Quran and how the right articulation is pronounce in every surah in the Al-Quran. But as for the SIASA, the main focus of this software is an indexing of the surah An-Nas and a high usability user interface. Therefore, not all the function in the Holy Quran can be adopted into the SIASA.

As for the Al-Quran CD 2000 and The Koran, they both provide a beautiful and simple user interface and more towards to a multimedia environment. They also provide the user with different selection of background but only the Koran provides the printing function. Both of the function which are changing background and printing documents will be adapted in the SIASA software. For displaying the surah, most of the software except the Koran provides an easy access to the selection of surah. This display method will also be adapted in the SIASA. The display method in the Koran is not as good because the choices are in the surah's translation name.

It is determine that most of the software does not provide a selection of translation. This is an exceptional for the Holy Quran and both of the web site. Therefore, as for the SIASA, a few translations will be provided for the user's selection. This is important so that user can compare each translation from one another and by that way they can broaden their knowledge and enhance their thinking capabilities. Furthermore, the user is more aware of the different types of

versus of translation. SIASA will also provide a few tafseer for the same reason stated above.

Most of the software also provides a few selection of recitation of the surah. For a better user interaction it is foresee that the best way for user to learn on how to pronounce the surah An-Nas in a correct way is to listen to different type of qari. This is a much faster way to learn rather than listening to only one recitation over and over again.

For the search function, all the software have the same common methods which is the user type any phrases than the software will perform the search and display the result on the surah itself or its translation. Since SIASA only involves surah An-Nas, therefore user can search the translation via selecting the verses provided.

2.4 *Development Tools for Multimedia Software*

There are so many development tools that can be used to develop good multimedia software. This chapter will cover on what and why are the tools are chosen for the development of this project. The tools that are going to be used are as follows :

1. Visual Basic 6.0

Visual Basic is a product made by Microsoft to allow the common person to easily make full featured and powerful windows programs, database programs, ActiveX controls, fully Network Capable Client/Server programs and almost anything (Ref. 9). It is also know as the fastest technology for creating Windows-style user interface prototype (Ref. 10). There are also many resources available via the Internet and books. Visual Basic gets the application to the desktop faster and able to develop a quick and easy pump out application. It also promises a short development cycles and easy learning curve. The developer will always know what they are doing and can see an overview visual of their project. Visual Basic 6.0 able to create native-complied standalone executables and it is an important element for prototyping purposes. It also uses ActiveX controls that are a magnitude of times faster then previous OLE controls.

2. *Microsoft Access 2000*

Microsoft Access is also a product by Microsoft. It is a relational database management system designed to provide a rich and highly flexible environment for managing data (Ref. 13). Access is actually made up of two components: a database engine, called Jet and a user interface to that engine. Jet is the component of Access that does all the actual work of storing data to disk and retrieving it. Access can interpret code written in Visual Basic syntax and stored in modules in Jet databases. Part of Access' data management environment is a high degree of programmability. So, when it comes to databases, there is a lot of common ground between Visual Basic and Access. It is also more compatible because both of the products are produce by Microsoft.

3. *Macromedia Swish 2.0*

Macromedia Swish 2.0 was developed by Macromedia. It is a most useful for multimedia and low bandwidth animation (Ref. 11). It can design beautiful text animation. It can also build intuitive interfaces that maintain branding and produce standalone run-time animation application. Macromedia Swish 2.0 is also known to be easy to use and learn in a short time (Ref. 12). There are also many resources, tips and guidance available via the Internet and books.

4. *ParsNegar II*

ParsNegar II is a simple word processor with very powerful features for Arabic script type languages. It allows user to easily type text under English and western versions of 32-bit Windows and able to write Arabic fonts without using an Arabic keyboard. It uses an on screen keyboard where user can click on the Arabic character. It is designed to work well with most western Windows applications. User can use Windows copy and paste or drag and drop to transfer user text into their favorite Windows applications (Ref. 16).

5. *Adobe Photoshop 6.0*

Adobe Photoshop is one of the industry standards for desktop image editing and graphic manipulation. It provides functions such scanning images, make original art, or composite images as well as performing color correction, retouching and other image manipulations (Ref. 14). Adobe Photoshop 6.0 software introduces the next generation of image editing with powerful new features that offer something for every user. Delivering the broadest and most productive toolset available, Photoshop helps you explore your creativity, work at peak efficiency, and achieve the highest quality results across all media (Ref. 15). It uses an output sharp type edges with user image to produce high-quality results. It is faster with user-interface enhancements that help you to get to work quickly and make it easy to take full advantage of the comprehensive tool set.

2.5 *Interactive Multimedia Software as a Medium for Education*

According to Bass (1997) 'Interactive Multimedia has been called a "hybrid technology". Its combines the storage and retrieval capabilities of computer database technology with advance tools for viewing and manipulating these materials. Multimedia has a lot of different connotations, and definitions vary depending on the context. In the context of upper secondary and postsecondary education, interactive multimedia is defined by three criteria (Ref. 17):

- Interactive Multimedia is any package of materials that includes some combination of texts, graphics, still images, animation, video and audio
- These materials are packaged, integrated and linked together in some way that offers users the ability to browse, navigate and analyze these materials through various searching and indexing features, as well as the capacity to annotate or personalize these materials
- Interactive multimedia is always "reader-centered." In interactive multimedia, the reader controls the experience of reading the material by being able to select among multiple choices, choosing unique paths and sequences through the materials. One of the key features of interactive multimedia is the ability to navigate through material in whatever ways are most meaningful for individual users

Interactive multimedia is synonymous with another frequently used term: hypermedia. Hypermedia is the multimedia version of the term hypertext. A hypertext is defined as any non-sequential, electronic text, assembled not as a seamless sequence of material with a beginning, middle and end, but as a web of interrelated chunks of text. In a hypertext, the reader controls the sequence of reading by choosing how to navigate among these chunks of text by various electronic links.

The term hypermedia was coined to mean a hypertext that uses multiple media. In other words, hypermedia is a collection of multimedia materials with multiple possible arrangements and sequences. Hypertext and hypermedia are electronic concepts that can only exist in a computer-based environment. Only in a computer-based environment can materials can be linked and organized in multiple ways simultaneously, searched, sorted and navigated in hundreds of possible combinations by different users.

Business Week Magazine (February 28, 1994) ran a report called "The Learning Revolution" in which educational experts agreed that interactive resources, which make learning fun, would revamp American education. This resource tool, although not new, will change the way we view educational development and learning (Ref. 19). More specifically, it is about the multimedia format and its CD-ROM extension. It is a combination of graphics and sound, of video and music, a kind of smorgasbord of things that would be appealing to children and for that matter, adults too. And, if presented in the right way, the student will learn more

than during an equivalent amount of time spent with traditional teaching methods (e.g., blackboard, written text, and lecture). If the cd-rom software is really good and renders a true interactive interface, then the child will learn at a more accelerated pace. To quote Donavan A. Merck, manager of the Educational-Technology Office for the California Education Department, "the multimedia format is primary to the enhancement of all educational curriculum. Its importance should not be taken lightly."

Many multimedia projects are best distributed on a CD-ROM because they hold up a lot of data - high data capacity 650Mb. They are convenience and are relatively inexpensive.

We've seen that interactive multimedia, by definition, has the capacity to deliver large amounts of materials in multiple forms and to deliver them in an integrated environment that allows users to control the reading and viewing experience. How then do these defining characteristics and virtues translate into benefits in an educational environment?

First of all, multimedia programs bring to education the extraordinary storage and delivery capabilities of computerized material. This is especially important for schools, libraries and learning institutions where books are difficult to obtain and update. Multimedia is a powerful and efficient source for acquiring learning resources. Multimedia can also provide educational institutions access to other kinds of inaccessible materials, such as hard to find historical films, rare

sound recordings of famous speeches, illustrations from difficult to obtain periodicals, and so on. Multimedia can put primary and secondary source materials at the fingertips of users in even the remotest locations from major research facilities.

Secondly, it is not just sheer access to these materials that makes multimedia a powerful tool, but the control over those materials that it gives to its users. Interactive multimedia programs enable the user to manipulate these materials through a wide variety of powerful linking, sorting, searching and annotating activities. Each of these activities can be made to reinforce and inculcate various intellectual skills, in addition to satisfying certain cognitive needs for quality learning, such as the ability to follow through links at the immediate moment when curiosity is aroused, and the ability to view different forms of the same information side-by-side.

Furthermore, interactive multimedia programs usually integrate some combination of orientation tools, such as timelines, graphs, glossaries and other pedagogical guides. These kinds of tools further point to the third major benefit of multimedia: the personalization or individualization of the learning experience.

By allowing users to control the sequence and the pacing of the materials, multimedia packages facilitate greater individualization in learning, allowing students to proceed at their own pace in a tailored learning environment. Furthermore, interactive multimedia can be a powerful learning and teaching tool

because it engages multiple senses. Students using multimedia are reading, seeing, hearing, and actively manipulating materials.

The basic steps in which the process of universal multimedia project can be broken down into are the following:

1. *Concept*
    - Conceiving an idea
    - Planning the development process
    - Outlining and prototyping

2. *Interface Content*
    - Designing the architecture
    - Acquiring and interface; producing content
    - Creating interface elements
    - Preparing content.

3. *Product*
    - Assembling the elements into a cohesive production
    - Testing and making adjustments
    - Duplicating and distributing the final project

2.5.1 *Analysis and Synthesis*

At it most basic, Multimedia refers to the personal computer's growing ability to process not just text, but all sorts of visual and sonic information. One can say that multimedia is a broad-based phenomenon that already has a fare aching impact in business, education, entertainment, and personal creativity. Its emerging influence on the desktop is being driven by increasingly sophisticated personal computers and by the metamorphosis of information into digital form.

Multimedia has no set development formula because each project has its own specific individual set of requirements. Each project will have its own particular purpose and needs that will create variations in things such as display (viewing), sound (volume), color or black and white and many more.

Multimedia materials help students and teachers by way of reinforcement and extension, not substitution. What hypermedia provides is access to materials and unique personalized control over them. In other words, interactive multimedia isn't about replacing books, but about replacing the absence of books; hypermedia doesn't do what books do, but what books can't do.

With the inclusion of computers at work nearly twenty years ago, the importance of this tool to business was clear. It did not take long for educators to see the many advantages of computers. The potential is truly staggering. If we could have three or four computers in each classroom with the right curriculum, where

students could spend one hour or two a day with this resource, the benefits to the child and to society in the long run would be immeasurable. If the curriculum included the 'core' subjects (i.e., reading, writing, math and science) and was truly interactive, where the computer became the tutor, the teacher would have more time to devote to those who need additional help and guidance. With computers, the slow learners could work at their own pace and the accelerated student could jump ahead and develop more of their talents and potential. Such learning tools require the student to interact and pay attention. He cannot simply daydream or talk to friends, since the computer requires input at every turn. But this is what makes it fun.

The future of education will be about equipping the students with the social, vocational, and intellectual tools needed to work, to contribute, and to prosper in the world. This means teaching them basic skills and the ability of using those skills successfully. Consequently, they will become efficient and contributing members of society. In this respect, knowledge of computers will be of great importance to future employment. Students must understand and become comfortable with computers, its peripherals and a variety of software. Besides this, this resource will help students become independent seekers of knowledge and free them to become better thinkers. These predictions are not impossible to attain, although we do need to find a way to get these tools into every classroom, onto every desk, and into every home. Until then, the predictions of what computers and multimedia can do for education and our children; remain at best, only a "possibility" and a "promise".

Part 3 : Methodologies Adopted

3.1 *Human Computer Interaction (HCI)*

Methodology is a description of the process required for the development of a software system. It also helps software developer identify the major steps involved form the conception a software system to the final use of that system (Ref. 20). SIASA will be developing not only based on a traditional software engineering model but with a cognitive approach of HCI.

HCI is a communications channel between the human and the functional elements of the machines. The goal is to have the learner, not the technology, drive the interaction process. The current role of HCI designers is to (Ref. 21) :

1. Analyze each problem situation
2. Apply some reason and theory
3. Combine it with experience and intuition

HCI disciplines are psychology, computer science, information System, engineering, education and communication (Ref. 21). HCI advantages are it can save money, lowers operational errors, increase the like hood of success and increase the chance of a good user interface. Since this SIASA project depends a lots on the user interface so HCI is a very good model to be used.

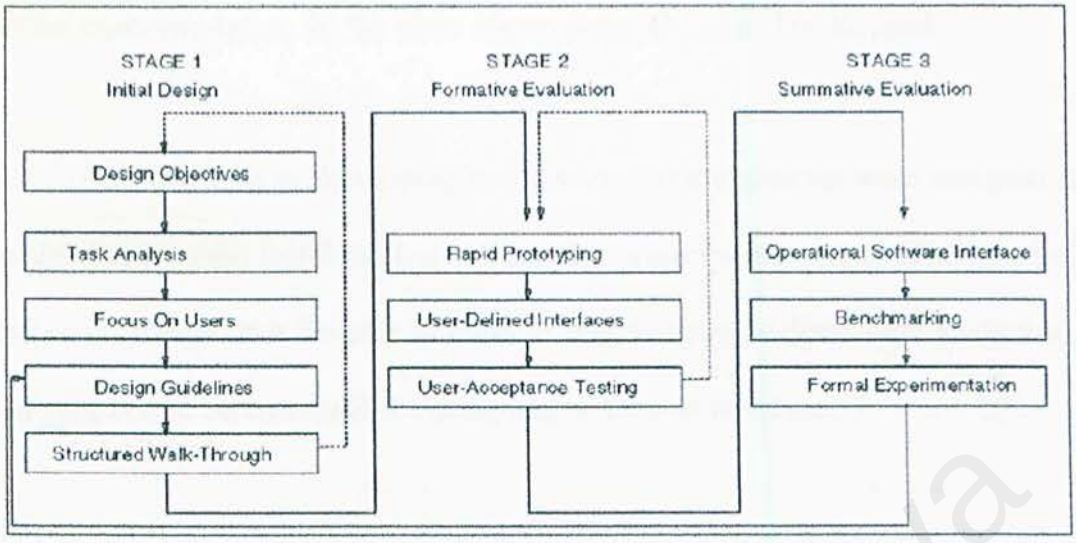

Figure 3-1 HCI Lifecycle

The above figure shows the lifecycle used in the HCI development. This project will emphasis on the design objectives, task analysis, design guidelines, rapid prototyping and benchmarking. Since this project is only a prototype, user acceptance testing and formal experimentation is not going to be conducted.

As show in Figure 3-1, there are three stages of development which are initial design, formative design and summative design. The initial design includes steps such as design objectives, task analysis, focus on user, design guidelines and structured walk-through. In the formative design there are two important steps which include rapid prototyping and user defined interface. The use acceptance testing is not going to be conducted because this project does not involve real user testing. Meanwhile, in the summative evaluation stage, steps such as operational

software interface and benchmarking will be conducted. The last step which is formal experimentation for the same reason stated above will be skipped.

The first step in developing the SIASA, some objectives were designed to ensure that it would fulfill the lecturer and this project requirements. Then the task analysis is design with the user in mind. A few design guidelines were studied and analyzed before continuing to the designing of the user interface.

The rapid prototyping will help to visualize the task analysis that had been design earlier. Here, user interface must be design according to the guidelines that are studied earlier and based on the task analysis which focus on the user. When the entire prototype is operational, benchmarking will be conducted to evaluate the operation of the software based on the task analysis and the user interface. If there is any non-satisfied functionality, the process will backtrack to the formative evaluation stage. This process may be conducted for a few times until the operational software interface fulfilled its objectives, task analysis and user interface guidelines.

Four major approaches to interface designs are :

1. Craft – design is through skill and experience
2. Enhanced software engineering – incorporates HCI techniques into traditional software design
3. Technologist – focus is automating and quantifying the design process

4. Cognitive - applies psychological knowledge to design

The cognitive approach is the most theoretical approach because it incorporates the knowledge base of cognitive technology in order to achieve the optimal design for the human user. This "know the user" approach to design becomes an important feature by explaining how the mind works. In order for designers to reach the goal of producing interfaces that enrich those areas of cognition, they must first understand user interfaces in terms of these areas.

Cognitive psychology uses scientific methods to examine the concepts of representation and process of individuals. Representation deals with what the person actually knows and how that knowledge is structured. An analytical model is build in order to predict user's actions. It is derived from the cognitive psychology paradigm of information processing. This paradigm explains how information is processed in the human
mind.

The following explains why the traditional waterfall does not work in an HCI development environment (Ref. 22):

1. None of its steps relate to the development of the interface specifically
2. There is an agreement within the HCI community that an interactive design technique is required for the development of the user interface
3. A user interface cannot be specified without repeated testing by the user

4. A single late evaluation of usability is too risky in terms of resources and quality objectives
5. The user plays a relatively a minor role
6. It implies that any stage should be frozen before continuing with the later stages
7. It is still unrealistic because many software project sequencing of phase is not followed
8. It assumes that user requirements cab be precisely specified

The table 3-1 describes the differences between traditional software engineering and HCI.

Difference between traditional software engineering and HCI

Traditional software engineering	Human computer interaction
▪ Main emphasis seems to be on the technical aspects of the task (i.e. getting the data structure & programs right). ▪ System centered. ▪ Good at modeling certain aspects of the problem domain using formal methods. ▪ Deals with managerial and financial issues very well. ▪ Specifying and building functional aspects of the software system. ▪ Ignore the contribution of the HCI community to the design process. ▪ Preference for specifying a work strategy or methodology. ▪ Prefer formal methods and devalue informal methods. ▪ End user often seen as playing a relatively minor role. ▪ User might have to 'lump it or leave it'. ▪ Concerned with logical flow of data, computational efficiency, ease of development and maintenance.	▪ Focus more on the goals, need abilities and knowledge. ▪ Preferences to the user and involves them much more closely at all stages of the design. ▪ User centered ▪ Emphasize developing a deep understanding of user characteristic and a clear awareness of the task a user must perform. ▪ Use formal evaluation technique to replace intuition in guiding design. This reality check improves final product. ▪ Resist defying a process maybe due to the need for flexibility or lack of maturity in their disciplines. ▪ Formal methods are part of HCI process but many informal methods are used and valued. ▪ Multidisciplinary area from psychology, computer science, information science, engineering, education and communications. ▪ Concerned more with quality of the interface.

Table 3-1 Difference between traditional software engineering and HCI

3.2 *Rapid Prototyping*

Prototyping is a process that enables developer to create a model of the software to be built. It is a working model that is functionally equivalent to a subset of the target software. The subset usually consist of data input screens (i.e. forms), user interfaces (i.e. menus, dialog box, etc) and reports (Ref. 8). For this project, the prototyping will be done by using Visual Basics 6.0 that was explained earlier.

As follows are advantages of rapid prototyping and why it is used in this SIASA project :

1. Advocated when the form that human-machine interaction should take is not clear
2. Recognizes that interactions of the design steps are a normal part of the process
3. Save time and resources
4. Easy and inexpensive to create
5. Can find usability and design problems and adjust it before investing heavily in the final design and technologies
6. Can help illustrate that an idea works, express its qualities in a visual and interactive way
7. Can be built at any time during the project but usually created early in the project which is during the planning and specification phase
8. Concentrate on smart design based on available interface design guidelines

Part 4 : System Analysis

4.1 *Introduction*

This chapter provides details of the activities undertaken for system analysis purposes. Activities that have been done are broken down to three major steps which are task analysis, functional requirements and external interface requirements.

4.2 *Task Analysis*

The second step in the initial design stage is the task analysis. Task analysis involves hierarchical task analysis (HTA), sampling and investigation of data, observation and interviews. Before this step is conducted, few questions should be design as a guide to fulfill all the requirements of this SIASA project. The questions design must be relevant to the objectives of the SIASA. The end of this chapter must answer all the questions below :

1. What is Al-Quran ?
2. What is surah An-Nas ?
3. What are surah An-Nas's translation, tafseer and "hikmah"?
4. How many reliable translation and tafseer are available ?

5. Should there be more than one person's recitation ?
6. What are the user interface design guidelines ?

To answer the first and second questions, many books and articles were revised (Ref. 23-27). There are many interpretation of what exactly is the Al-Quran. A summary had been done from the information gathered from all of the resources. This information will be an introduction page for the SIASA. The user can user whether to view the introduction. If the user is not interested, they can skip the introduction and start using the SIASA software to perform any of their tasks.

There are many different versus of translation and tafseer on the surah An-Nas itself. To answer the third question, the fourth question must be performed first to gain knowledge on which sources are reliable and most suitable to answer question three. After some analysis was done (Ref 23-27), it is concluded that :

1. There are three different types of tafseer which are tafseer bil-riwaya (by transmission), tafseer bil-diraya (by sound opinion) and tafseer bil-ishara (by indication, from signs). After further investigation, it is learned that the most reliable tafseer is the tafseer bil-riwaya. This tafseer explains the Al-Quran by the Al-Quran itself which can be traced backed through a chain of transmission to a sound source example the Al-Quran itself, the explanation of the Prophet Muhammad and the explanation by companions of the Prophet.

2. The most popular tafseer of the Al-Quran are :
    - Tafseer Al-Tabari by Ibn Jarir Al-Tabari
    - Tafseer Kabir by Fakhruddin Muhammad Razi
    - Tafseer Al-Samarqandi by Abu al-Laith al-Samarqandi
    - Anwarut Tanzil by Nasiruddin Abu Said Baidawi
    - Tafseer Ibn Kathir by Ismail bin Amr bin Kathir ad-Damashqi
    - Tafseer Al-Jalalain by Jalal al-Din al-Mahalli and Jalal al-Din al-Suyuti
    - Tafseer Al-Maraghi by Ahmad Mustafa al-Maraghi

3. The most popular tafseer that is translated from Arab to English are from:
    - Abdullah Yusof Ali
    - Maulana Abdul Ala Maududi
    - Muhammad Asad
    - Muhammad Marmaduke Pickthall

After both third and fourth questions are answered it is determine that SIASA should have more than one translation and tafseer. This is to enable the user to expand their knowledge in different types of versus and compare them to one another. This will also be something different from other software that has been discussed earlier.

Question five is based on the analysis done in the section 2.3. There should be more than one recitation to enable user to compare each recitation and enhance user's knowledge on the articulation, pronunciation and the "tajwid" of the surah An-Nas. At the end of the task, user may feel confident to recite the surah An-Nas on their own.

An interview with a few ustaz or people with Al-Quran knowledge will be conducted in the second phase of this project. This is to certify whether all the information gathered on the Al-Quran and surah An-Nas are correct and appropriate to be used in the SIASA.

There is a lot of user interface guideline available to be used. Giant computer products company such as the IBM, Macintosh and Windows have their own user interface guidelines. As for this SIASA project, the combination from the three guidelines will be used. This factor will be discussed later in the system design phase.

From all the knowledge gained from this system analysis, the figure 4-1 will show an overall overview what are task that can be conducted by the user.

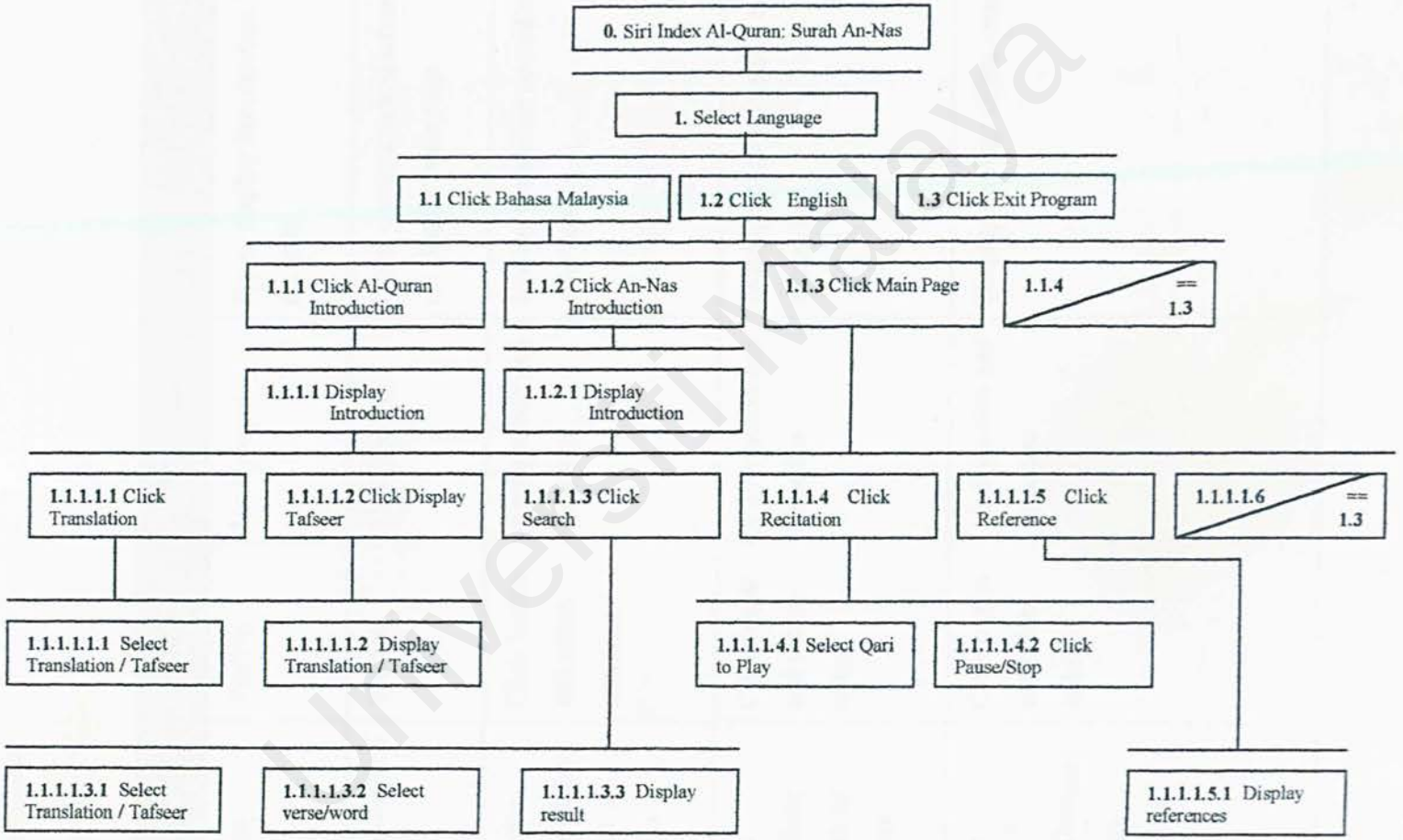

Figure 4-1 Hierarchical Task Analysis of SIASA

4.3 *Functional Requirements*

This chapter will cover the entire functional requirement for the user in the SIASA software.

Function Name	Inputs	Processing	Outputs
Select Language	Yes/No	Verify input	If yes, display introduction. If no, exit
View Al-Quran/An-Nas Introduction	Yes/No	Verify input	If yes, continue introduction. If no, display main page
Display translation - 8 different translation either in English or Bahasa Malaysia	Click button and select translator	Verify action and selection	Display translation according to selection
Display tafseer - 4 different tafseer either in English or Bahasa Malaysia	Click button and select tafseer	Verify action and selection	Display tafseer according to selection
Display search – Selection of translation and tafseer or surah in either English or Bahasa Melayu	Click button and select search	Verify action and selection	Display findings of the search

Function Name	Inputs	Processing	Outputs
Listen to Recitation – 4 choices of Qari	Click button and select Qari	Verify action and selection	Play recitation according to selection
Display references	Click button	Verify action	Display references

Table 4-1 SIASA User Functional Requirements

4.4 *External Interface Requirements*

4.4.1 *Hardware Interface*

SIASA would interface with the following hardware requirements :

- Personal computer with at least Pentium 166 Mhz Processor
- 1 GB of hard disk space
- At least 32 MB of RAM
- Multimedia drivers which must include a speaker
- Other standard computer peripherals

4.4.2 *Software Interfaces*

SIASA would interface with the following software requirements :

- Windows 95,98,2000 or ME
- Visual Basics 6.0
- Macromedia Flash player

Part 5 : System Design

5.1 *Introduction*

After all the requirements needed are gathered in the system analysis phase, the system design can be conducted. This chapter will list down all the steps taken in designing the SIASA functionalities and user interface.

5.2 *System Functionalities*

System functionalities provided by SIASA would be illustrated via the following structure charts.

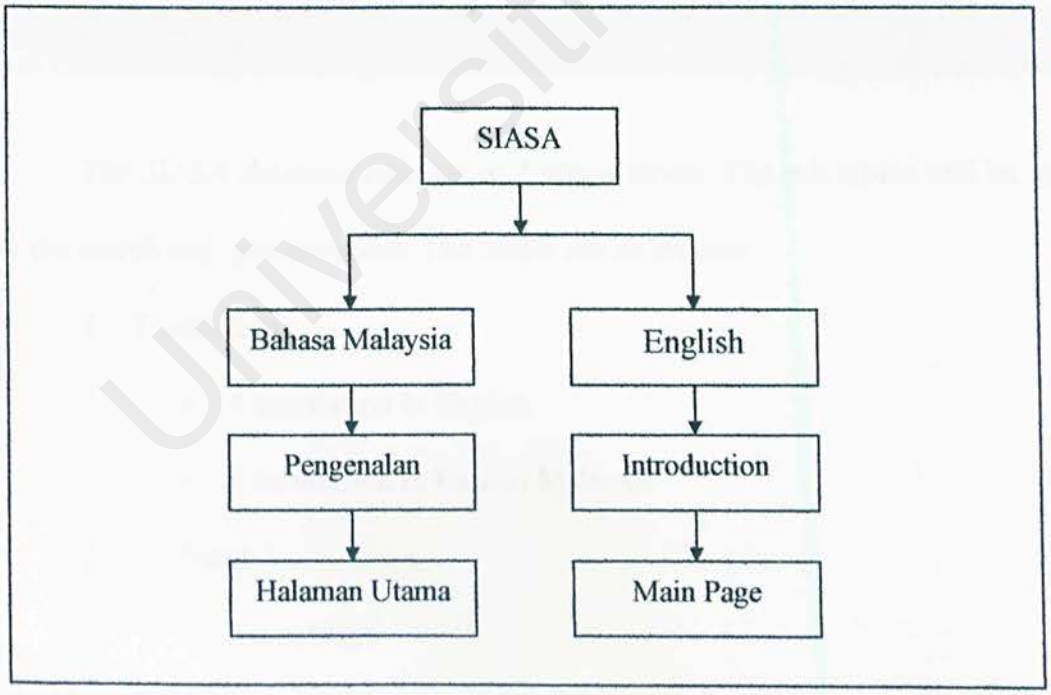

Figure 5-1 SIASA Introduction Stage

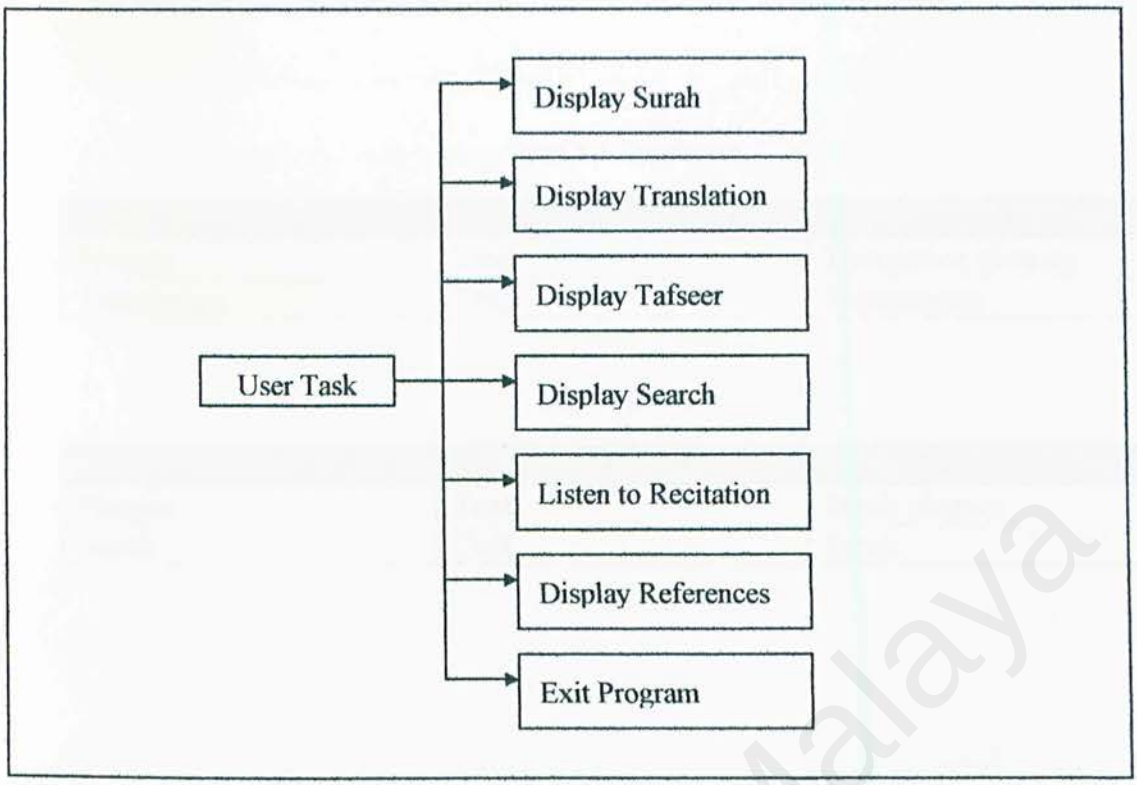

Figure 5-2 SIASA User's Task

5.3 *Database Design*

The SIASA database consists of 2 major tables. The sub tables will be used for the search and quiz purposes. The tables are as follows :

1. Translation
    - 4 translation in English
    - 4 translation in Bahasa Malaysia
2. Surah
    - Translation
    - Tafseer

The table structures are as follows :

Table 5-1 Translation

Column Name	Data Type	Description
Phrases	Text	Translation phrases
Translation	Text	Translations

Table 5-2 Surah

Column Name	Data Type	Description
Phrases	Text	Surah phrases
Surah	Text	Surah

5.4 *User Interface Design*

User interface design considerations play a crucial role, as SIASA is multimedia interactive software. Below is the summary of user interface guidelines.

5.4.1 *User Interface Design Tips and Techniques*

1. Consistency

The most important thing is to make sure that the software user interface works consistently. If user can double-click on items in one list and have something happen then he should be able to double-click on items in any other list and have the same sort of thing happen. Buttons should be placed

consistently on all the windows. Labels and messages should use the same wording and use a consistent color scheme throughout the software. Consistency allows users to build an accurate mental model of the way that it works, and accurate mental models lead to lower training and support costs.

2. *Set standards and stick to them*

The only way to ensure consistency within the software application is to set design standards and then stick to them. The best approach is to adopt an industry standard and then fill any missing guidelines that are specific to your needs. Industry standards, such as the ones set by IBM and Microsoft, will often define 95%-99% of what you need. By adopting industry standards you not only take advantage of the work of others you also increase the chance that your application will look and feel like other applications that your users purchase or have built. User interface design standards should be set during the Define Infrastructure Stage.

3. *Explain the rules*

Users need to know how to work with the application that is built for them. When an application works consistently it means the rules will need to be explain only once. This is a lot easier than explaining in detail exactly how to use each and every feature in an application step by step.

4. *Support both novices and experts*

This is to make sure that casual users can understand how the application works and does not boredom the expert users.

5. *Navigation between screens is important*

When the flow between screens matches the flow of the work that the user is trying to accomplish, then your application will make sense to your users. Interface-flow diagrams can be used during the Model Stage to model the flow between screens.

6. *Navigation within a screen is important*

In Western societies people read left to right and top to bottom. Because people are used to this should you design screens that are also organized left to right and top to bottom. You want to organize navigation between widgets on your screen in a manner that users will find familiar to them and related to the software application.

7. *Word messages and labels appropriately*

The text that will be displayed on the screens is a primary source of information for users. Using full words and sentences, as opposed to abbreviations and codes makes text easier to understand. Messages should be worded positively, imply that the user is in control, and provide insight into how

to use the application properly. Furthermore, messages should be worded consistently and displayed in a consistent place on the screen.

8. *Understand the software widgets*

Software developer should use the right widget for the right task, helping to increase the consistency in the software application and probably making it easier to build the application in the first place. The only way to learn how to use widgets properly is to read and understand the user interface standards and guidelines that are available.

9. *Look at other applications with a grain of salt*

It is always a good idea to look at the work of others to get ideas, until you know how to distinguish between good user-interface design and bad user-interface design. Too many developers make the mistake of imitating the user interface of another application that was poorly designed.

10. *Use color appropriately*

Color should be used sparingly in the software applications, and to use it a secondary indicator must also be used. The problem is that some of the users may be colorblind – if color is used to highlight something on a screen then it will need something else to make it stand out so that people will notice it, such as display a symbol beside it. Colors combination must also be consistent in the application so that it will have a common look and feel throughout the

application. Also, color generally does not port well between platforms – what looks good on one system may look poor on another system.

11. Follow the contrast rule

The best way to make sure that screens are still readable contrast rule must be follow: Use dark text on light backgrounds and light text on dark backgrounds. It is very easy to read blue text on a white background but very difficult to read blue text on a red background. The problem is that there is not enough contrast between blue and red to make it easy to read, whereas there is a lot of contrast between blue and white.

12. Use fonts appropriately

Old English fonts might look good on the covers of William Shakespeare's plays, but they are really hard to read on a screen. Use fonts that are easy to read, such as serif fonts like Times Roman. Furthermore, use your fonts consistently and sparingly. A screen using two or three fonts effectively looks a lot better than a screen that uses five or six.

13. Gray things out, do not remove them

Should the button be removed or grayed out? Gray it out, never remove it. By graying things out when they shouldn't be used people can start building an accurate mental model as to how your application works. If a widget or menu are simply remove instead of graying it out then it is much more difficult for the

users to build an accurate mental model because they only know what is currently available to them, and not what is not available. The old adage that out of sight is out of mind is directly applicable here.

14. Use non destructive default buttons

It is quite common to define a default button on every screen, the button that gets invoked if the user presses the Return/Enter key. The problem is that sometimes people will accidentally hit the Enter/Return key when they do not mean to, consequently invoking the default button. The default button shouldn't be something that is potentially destructive, such as delete or save.

15. Alignment of fields

When a screen has more than one editing field it should be organize in a way that is both visually appealing and efficient. The best way to do so is to left-justify edit fields, or in other words make the left-hand side of each edit field line up in a straight line, one over the other. The corresponding labels should be right justified and placed immediately beside the field. This is a clean and efficient way to organize the fields on a screen.

16. Justify data appropriately

For columns of data it is common practice to right justify integers, decimal align floating-point numbers, and left justify strings.

17. Do not create busy screens

Crowded screens are difficult to understand and hence are difficult to use. Experimental results (Mayhew, 1992) show that the overall density of the screen should not exceed 40%, whereas local density within groupings shouldn't exceed 62%.

18. *Group things on the screen effectively*

Items that are logically connected should be grouped together on the screen to communicate that they are connected, whereas items that have nothing to do with each other should be separated.

19. Open windows in the center of the action

When the user double-clicks on an object to display its edit/detail screen then his or her attention is on that spot. Therefore it makes sense to open the window in that spot, not somewhere else.

20. Pop-up menus should not be the only source of functionality

Users cannot learn how to use the application if it hides major functionality. One of the most frustrating practices of developers is to misuse pop-up, also called context-sensitive, menus. Typically there is a way to use the mouse on the computer to display a hidden pop-up menu that provides access to functionality that is specific to the area of the screen that you are currently working in.

5.4.2 *Overview Design*

An overview design of the SIASA user interface will be illustrated via the following figures. These design are only the first draft. Hence, it may change as system implementation is progress.

SIASA Main Page

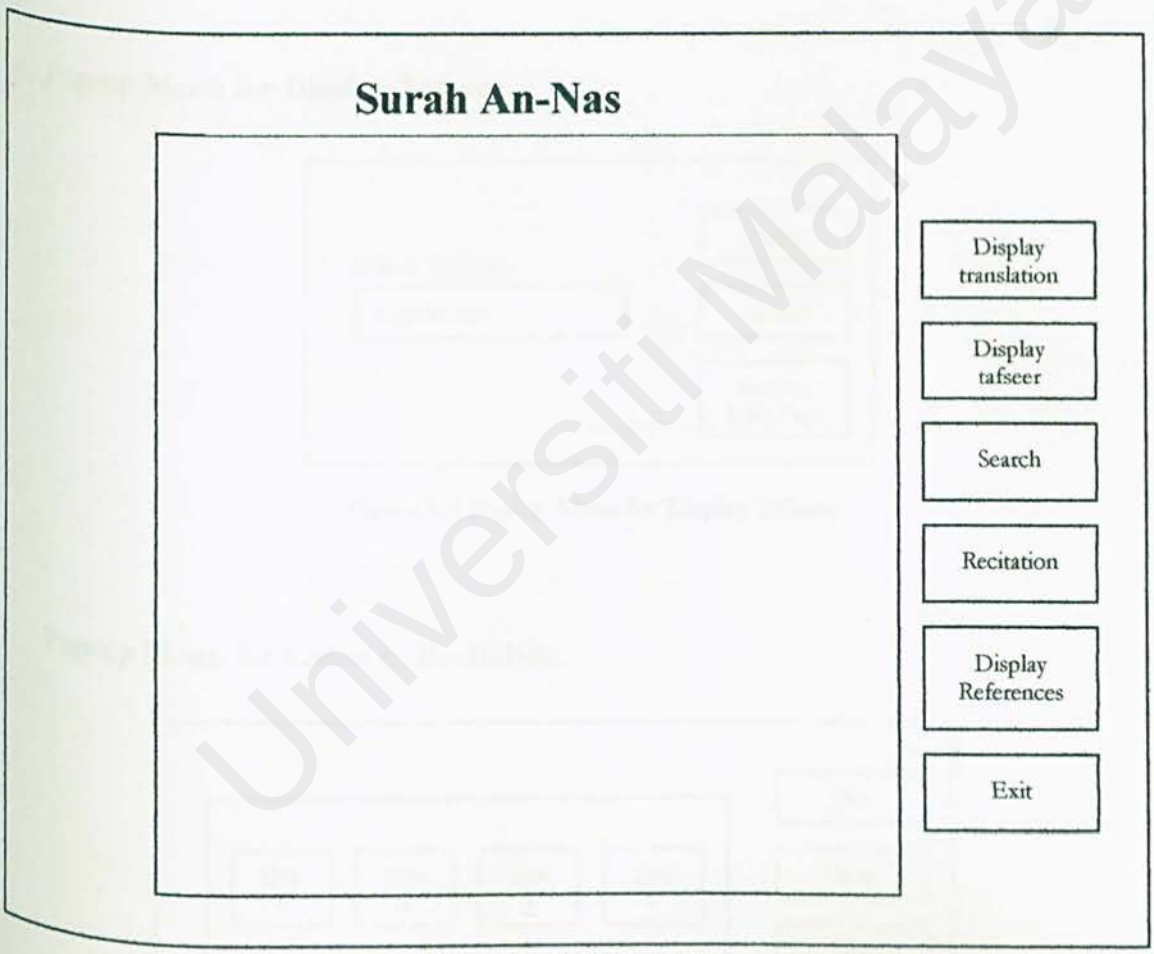

Figure 5-3 SIASA Main Page

Pop-up Menu for Display Translation

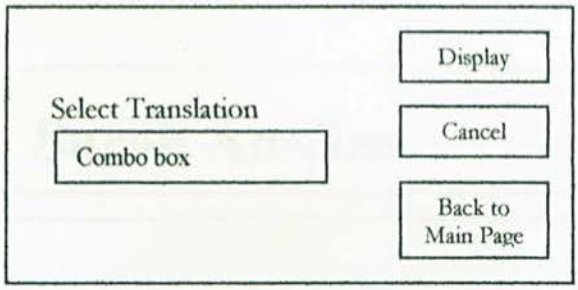

Figure 5-4 Pop-up Menu for Display Translation

Pop-up Menu for Display Tafseer

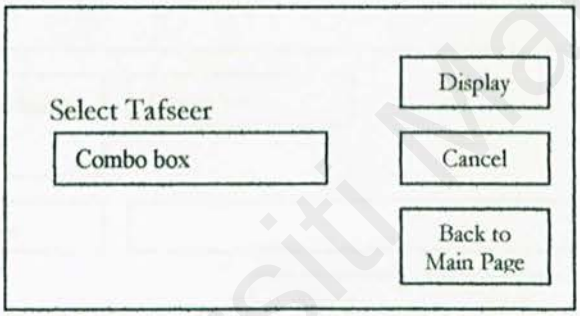

Figure 5-5 Pop-up Menu for Display Tafseer

Pop-up Menu for Listen to Recitation

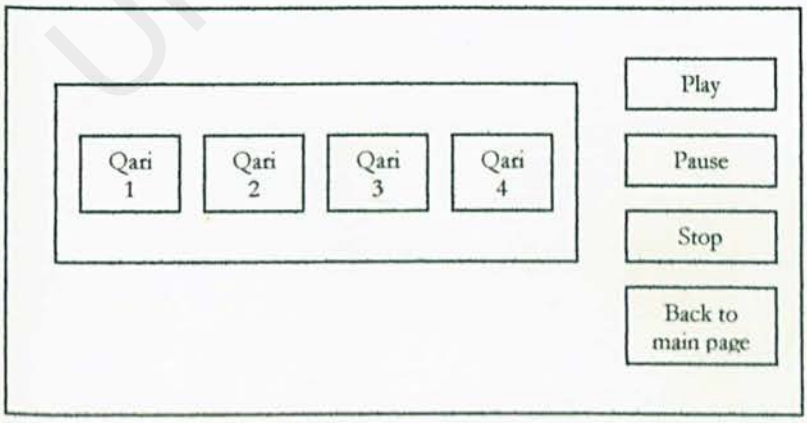

Figure 5-6 Pop-up Menu for Listen to Recitation

Search Page

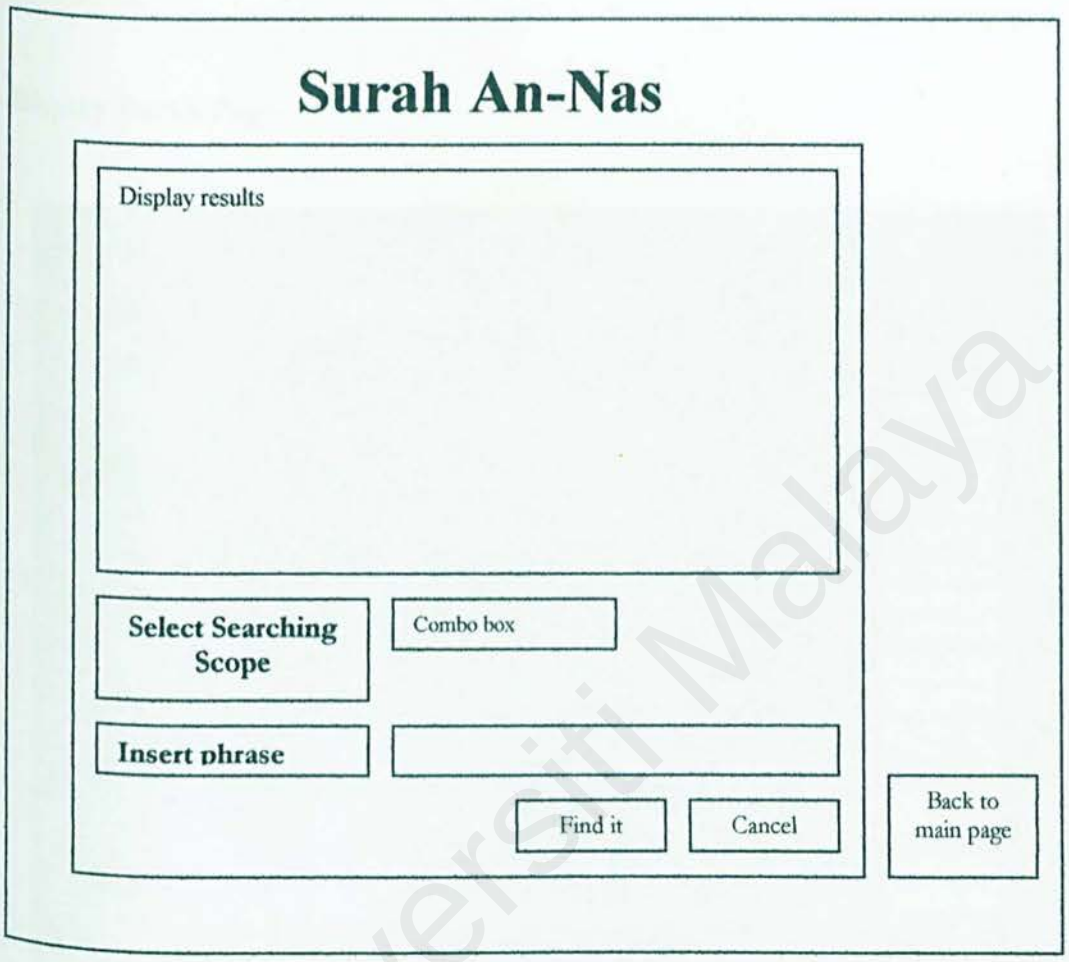

Figure 5-7 Search Page

5.4.3 *Prototype*

Figures below will show a few screenshots from the SIASA prototype.

Display Surah Page

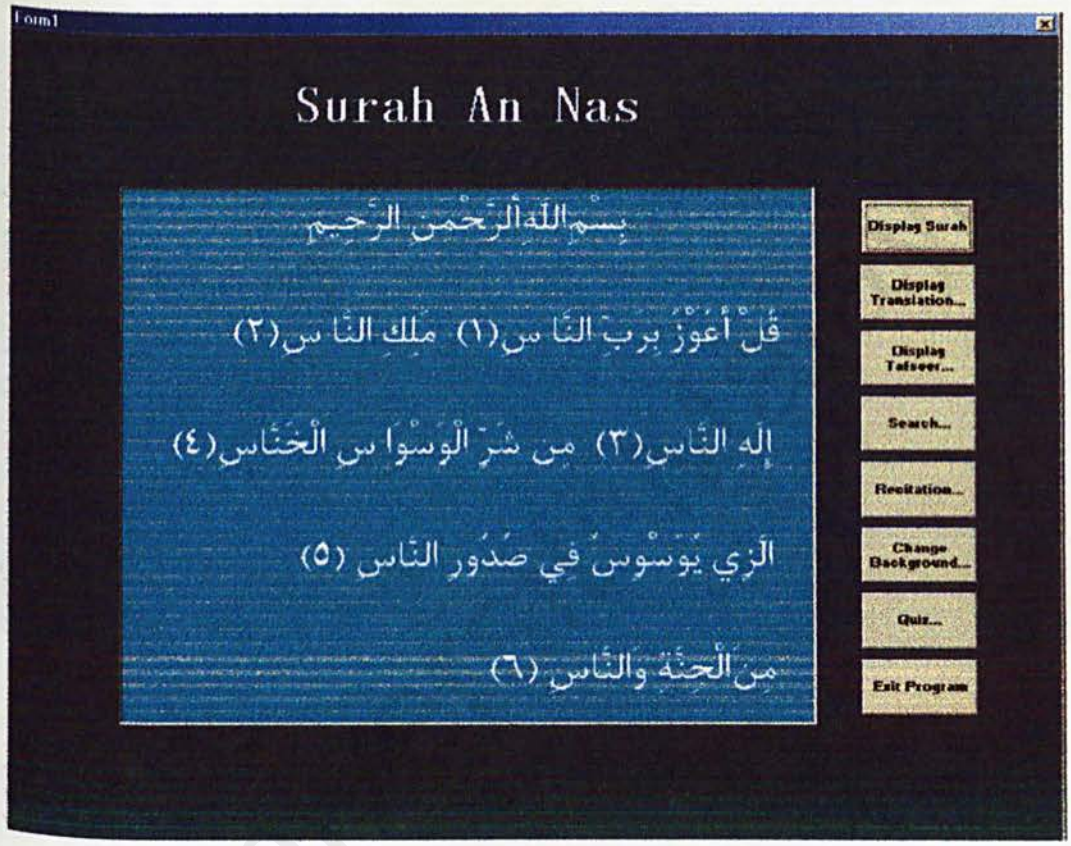

Figure 5-8 Display Surah Page

Display Translation Page

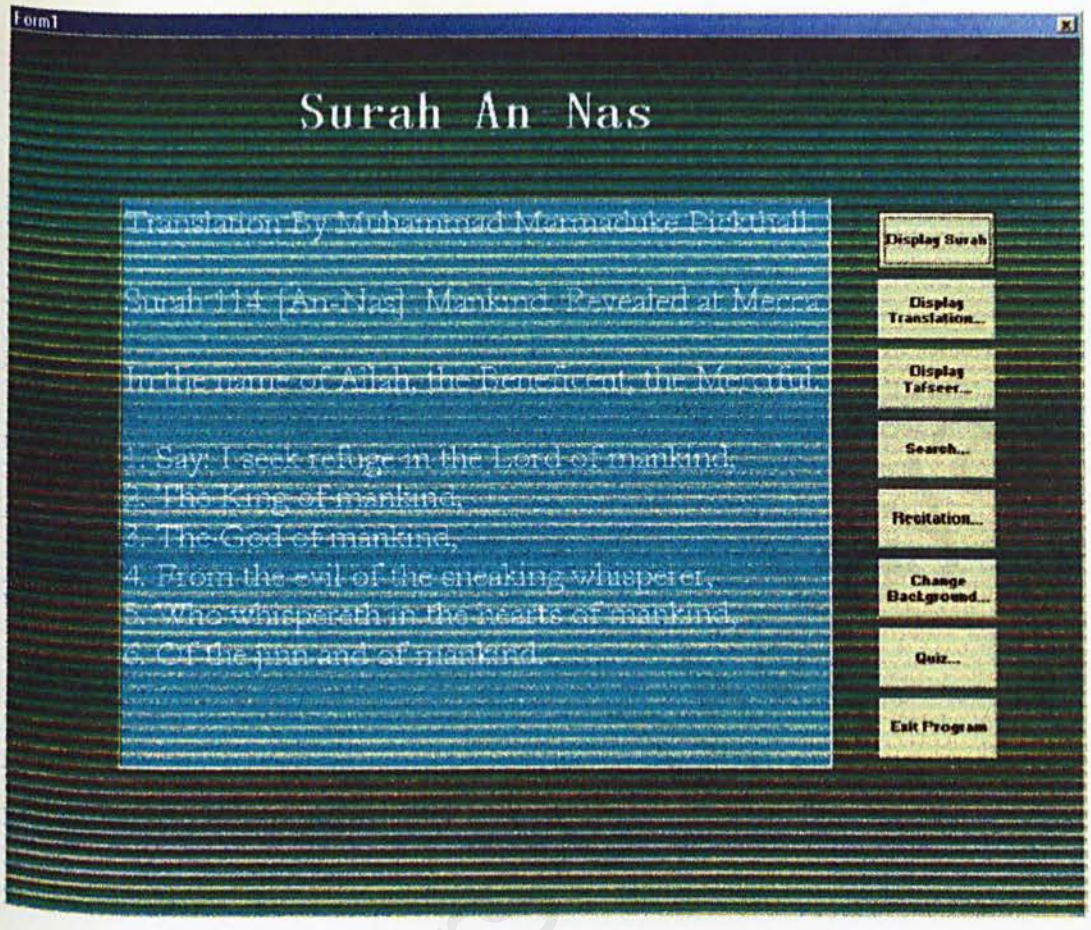

Figure 5-9 Display Translation Page

Pop-up Menu for Listen to Recitation

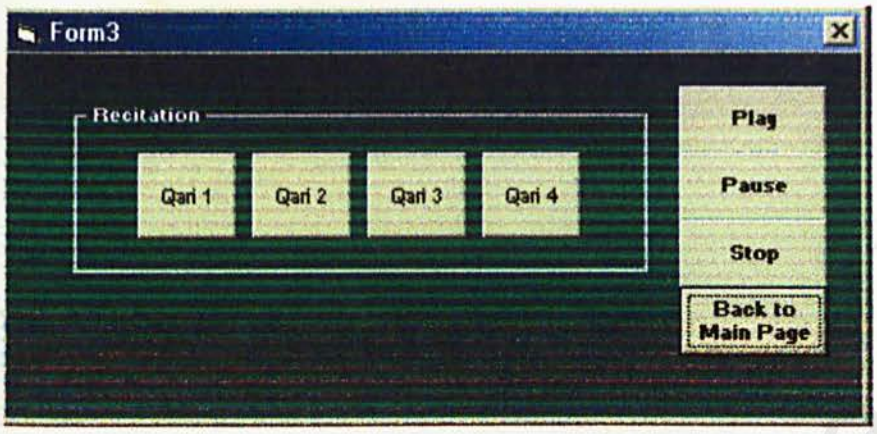

Figure 5-10 Pop-up Menu for Listen to Recitation

Part 6 : System Implementation

6.1 *Introduction*

System implementation is a process of writing the programming codes that implement the design. This task can be daunting for several reasons. First, the designer may not have addressed all of the idiosyncrasies of the platform and programming environment. Second, designer must write the codes in a way that is understandable to others. Third, designer must also take advantages of the characteristic of the design's organization, the data structure and the programming language's construct while still creating code that is easily reusable. Lastly, designer need to very familiar with the organization's standards and procedures so that not only others can understand what they have written but also why they are written.

6.2 *Coding*

Coding is the process of translating the design specification into source codes that can be process by the computer. Translation of the design into codes will become easier if the standards and procedures are in place. Standards and procedures can help the developer to organize their thoughts, avoid mistakes and maintain correspondence between design and code components.

6.2.1 *Coding Approach*

Various procedures involve methods for documenting code with the purpose of to construct an easier coding style and easy to follow. In addition to that, standardize documentation can clarify the functions performed by each section of the program. Thus, good documentation can help to locate faults without difficulty and make changes effortlessly. Likewise, modifications to code that result from changes in hardware or interface specifications are straightforward and the likelihood of error to occur is minimized.

It is essential to have a direct correspondence between program design and code components. Design characteristic such as low coupling, high cohesion and well defined interfaces should be integrated so that algorithms, functions, interfaces and data structure can be traced easily from design code and vice versa.

Programming involves an immense of creativity and the design is a guide to the function or purpose of each component. On the other hand, the programmers have extensive flexibility in implementing the design as codes.

Regardless the programming language used, each program component involves at least three major concepts which are control structure, algorithms and data structure. Consequently, the implementation of the SIASA will adhere to the three concepts mentioned above. This is to ensure software quality persist within the SIASA.

6.2.2 *Coding Design*

There are several general strategies that be applied in upholding the design quality of the programming codes. Hence, the design codes of the SIASA will follow the strategies that will be discussed below.

1. *Isolate input and output*

   It is beneficial to isolate or localized program sections that perform input and output functions from the rest of the code. This will simplifies maintenance when input/output hardware is changed as only input/output code need to be changed.

2. *Include pseudocode*

   The design usually lays out a framework for each program component. Pseudocode can be utilized to adapt the design to the chosen language. By adopting constructs and data representation without becoming involved in the specifics of each commands, it can be experiment and decide which implementation is most enviable.

3. *Revising and rewriting instead of patching*

   Commonly, the designer will revise and rewrite design until they are contented with the result. Easily, they can return to the design if the control flow is distorted or difficult to understand. Reexamining the design able to help to foresee whether the problems we encounter are inherent in the design or in the translation of the code.

4. *Reuse*

There are two method of reuse: producer reuse, refers to the creation of new components and consumer reuse, refers to the use of existing components.

6.2.3 *Pseudocode*

1. Search translation

1.1 Select translator

```
If (translator textbox) is click then
   Open Database
   Open Table = value of textbox
   Display translation
End if
```

1.2 Select verse

```
If (translator textbox) NULL then
     Msgbox "Select Translator"
Else
     Change font of display = value of verse
End if
```

1.3 Search tafseer

```
If (tafseer translator textbox) NULL then
     Msgbox "Select translator"
```

```
Else
    Open Database
    Open Table = value of tafseer textbox
    Display result
End if
```

2. *Recitation*

2.1 Play Wav

```
If MMControlMode = Not Open then
    "Open"
End if
If MMControlMode = Pause then
    "Pause"
Else Play
End if
Do DoEvent
Loop Until MMControlMode = "Play"
```

2.2 Pause Wav

```
MMControlCommand = "Pause"
Do DoEvent
Loop Until MMControlMode = "Pause"
```

2.3 Stop Wav

```
MMControlCommand = "Stop"
MMControlCommand = "Close"
Do DoEvents
Loop Until MMControlMode = Not Open
```

Part 7 : System Testing

7.1 *Introduction*

Software errors and failures occur mainly because of inadequate or improper testing. Quality software however demands that software be tested carefully. Testing is not the first place where faults finding occurs because the requirements and design reviews helped to ferret out the problems early in the development. According to Alka Jarvis (Ref. 32), there is a different between the four basic concepts related to software testing. There are :

1. Error detection - It involves identifying errors, inspection and walk through in the unit level.
2. Error removal - It involves debugging and other strategies for identifying where the error occurs in the code
3. Error tracking - It is important to find and correct the cause of the error, as it is to fix the error itself.
4. Regression testing - It is testing to see if the fix or rework to the code actually fixes the error, fixes it in one place and breaks it in another, or breaks the code in other places without actually fixing it at the point of the software where the fix was attempted.

The purpose of testing is to detect the presence of errors in SIASA to maintain its quality. There are several testing principles that have been followed to ensure that SIASA are well tested. The principles adopted are as follows :

1. Test should be planned before testing begins.
2. All tests should be traceable to the requirements, which means that SIASA must meet all the requirements of the customer.
3. Testing should begin in the small and progress towards testing in the large.

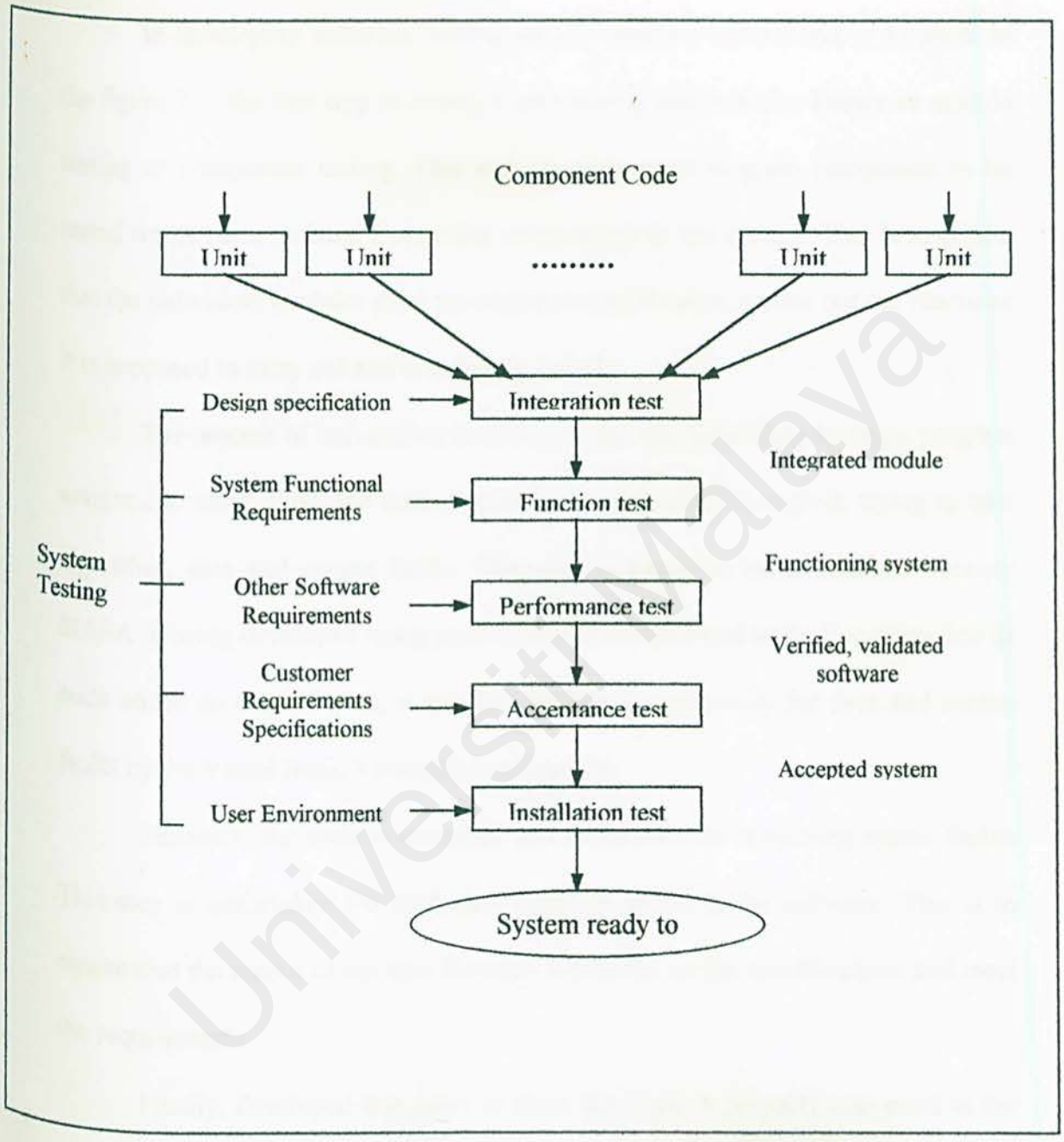

Figure 7-1 Levels of System Testing

7.2 *Unit Testing*

In developing software, testing usually involves several steps. As show in the figure 7-1, the first step in testing is unit testing which is also known as module testing or component testing. This step involves each program component to be tested on its own, isolated from other components in the system. This is to ensure that the individual modules meet the required specification, carries out the functions it is supposed to carry out and is correctly coded.

The process of unit testing is similar to the one used when testing a program assigned in class. First, the code is examined by reading through it, trying to spot algorithm, data and syntax faults. This step is easier to be undertaken because SIASA is being developed using prototyping techniques and tools. For every line of code added to the software, it can be checked automatically for data and syntax faults by the Visual Basic's compilation function.

Secondly, the code is complied and eliminates the remaining syntax faults. This step is undertaken for each new function added to the software. This is to ensure that the output of the new function is relevant to the specifications and meet the requirements.

Finally, developed test cases to show the input is properly converted to the desired output. During this step, valid and invalid actions are created. This is because the SIASA only involves user's action and does not need any input from the user.

The functions that have been tested in this unit level are translation, tafseer, search, recitation and references. These functions are checked to verify whether it meets the user requirements, user's actions and display the output requested by the user in the correct manner.

7.3 *Integration Testing*

When all the individual components or functions have satisfactory results which show that they are working correctly and meet the software objectives. Then all the components are integrated or combined into a working system. This integration is planned and coordinated so that when a failure occurs, it can be solved immediately. The goal is to determine if the system or subsystem meets the system requirements and functions properly and to test interfaces among the modules.

The system is viewed as a hierarchy of components, where each component belongs to a layer of the design. It can begin from the top and work the way down as testing is done, work from the bottom up or use a combination of these two approaches.

There are four major approaches for merging components to test the larger system (Ref. 33) :

1. Bottom up integration
2. Top down integration
3. Bing bang integration
4. Sandwich integration

Among these approaches, the Top down Integration is used in the SIASA integration testing where testing begins from the top and works the way down.

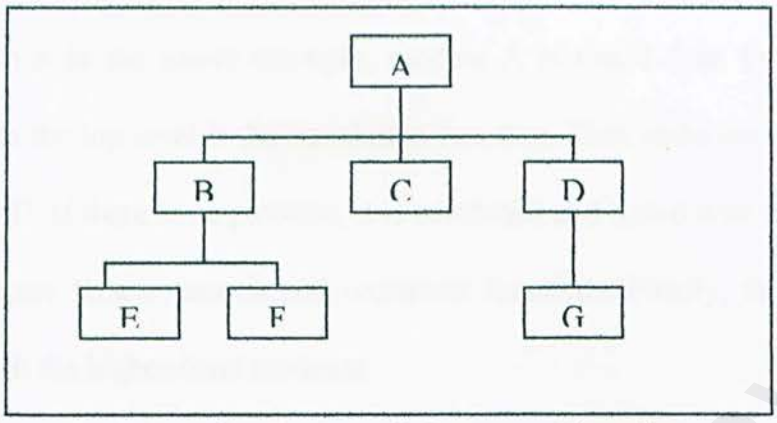

Figure 7-2 Example Component Hierarchies

7.3.1 *Top down Integration*

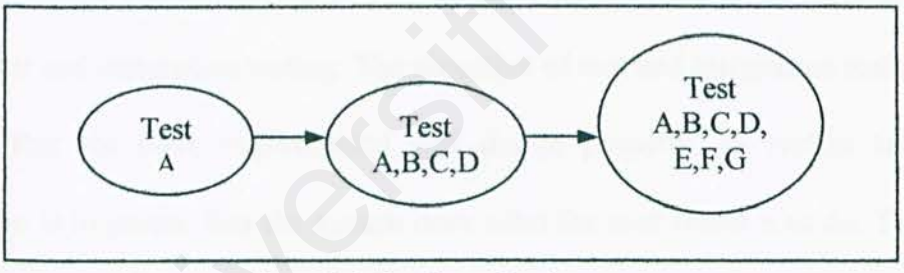

Figure 7-3 Top Down Testing

In the top down testing, the top-level modules are test first. The top level, usually one controlling component is tested by itself. Next, all components called by the tested component(s) are integrated and tested as a larger unit. This approach is reapplied until all components are incorporated.

A component being tested may call another that is not yet tested, so a special module called stud, a special program to stimulate the activity of the missing

component (the called module) is written. The purpose of the stub is to accept the input from the calling module (the module being tested) and pass back the output data so that the process can continue.

Thus in the above example, module A is tested first. In SIASA the first module in the top level is the translation function. Then stubs are used for modules B, C and D. If there is no problem, it is combined and tested with modules B, C and D which are tafseer, search and recitation functions. Finally, the lowest level is tested with the higher-level modules.

7.4 *System Testing*

Finally, system testing is performed. Testing the system is very different from unit and integration testing. The objective of unit and integration testing was to ensure that the code implemented was design properly. In system testing, the objective is to ensure that the system does what the user wants it to do. To perform this task, developer must work together with user. Because system testing takes place at a higher level, the testing focuses on behavior rather than function or functional structure and this cannot be tested through code audits based on pattern matching. The system testing result will show whether the entire system requirements, specification and objectives are achieved.

There are several steps in testing a system which are as follow :

1. Function testing
2. Performance testing

3. Acceptance testing
4. Installation testing

Since the SIASA is only prototype software, all the above steps will be undertaken except the installation testing.

7.4.1 *Function Testing*

The primary goal of this step is to tests if all the functions required by the application and specified in requirements specification documents are working properly. All the functions in SIASA will be tested to ensure that it can operate according to user's actions.

7.4.2 *Performance Testing*

When all the system function work perfectly according to specifications the performance test is conducted. This testing step will compare the integrated components with the nonfunctional system requirements. These requirements, including security, accuracy, speed and reliability, constrain the way in which the system functions are performed. The SIASA is tested to evaluate the followings :

1. Security precautions
2. Precisions accuracy in data, references, functions and process
3. Speed of data retrieval
4. System reliability and robustness
5. Response time to user's actions and error detection

When the system operates the way it was design, it is called a verified system. This verified system is the designers' interpretation of the requirements specification. Next, it is compared with the user's expectations by reviewing the requirements definition. Upon satisfaction, the system is now called a validated system, which verified that the requirements have been met.

7.4.3 *Acceptance Testing*

So far all the test have been done by the developer based on understanding of the system, its objectives and requirements. To ensure that the system meets the user's understanding of the requirements, which maybe different from the developer's point of view, the system must be tested by the user them. This purpose of acceptance test is to confirm that the system is developed according to the user's requirements and it is ready for operational use. Normally, it is part of a formal handoff or release process.

During an acceptance test, end users of the system compare the system to its requirements. A few measures that can be adopted during this test are :

1. Execute the test procedures
2. Evaluate test result
3. Check for discrepancies
4. Accept or reject the system based on the acceptance test criteria established in the beginning of the project

7.5 *Testing Result*

Upon completing the testing via questionnaire (Appendix 1) that have been distributed to 6 users's as a conclusion below is a graph that illustrate the end result.

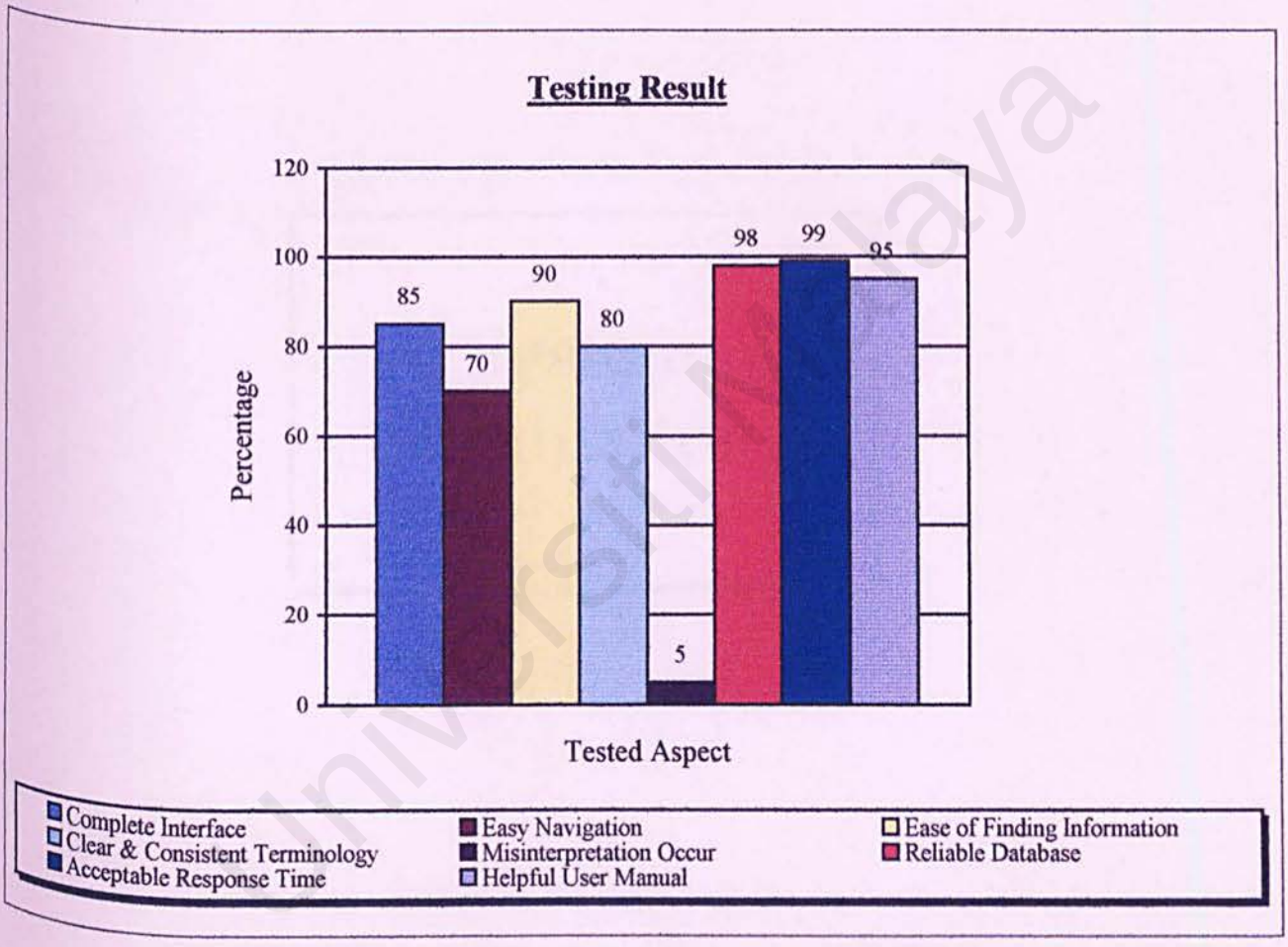

Figure 7-4 Testing Result Graph

Part 8 : System Evaluation

8.1 *Introduction*

When all the previous task and steps have been undertaken and the results were satisfactory, the system is then evaluated to determine the system's strength, limitation, constraints and problems encountered during the system development process. It also highlights on the knowledge gained and identifies the steps taken in solving problems.

8.2 *Problems Encountered and Solutions*

Problems always occur during any development project and the development process of SIASA has been no exception. Throughout the development of SIASA, many problems arise due to may reasons that will be stated below.

8.2.1 *Lack of Experience in the Chosen Programming Language*

Due to inexperience in VB Script, the functions provided in VB environment cannot be manipulated to the maximum extent. This is because all the previous semester subjects do not require knowledge in VB Script. However, with the edge of Internet today, VB Script and its environment can be learned through thousands of relevant example which are free and can be downloaded easily. Therefore, these

advantages had miraculously helped to enhance knowledge in VB Script and its environment.

8.2.2 *Determining the Functions of the System*

In view of the fact that there was no prior experience in developing a new system, it was quite difficult to determine to which extent to define the scope of the system consequently it can be developed within the given time frame. In spite of this, this problem was overcome with the help from the lecturers and analyzing existing relevant system.

8.2.3 *Time Constraint*

Throughout the analysis and design phase in Semester 1 there was not sufficient time to study and create the best solution of the system interface design. Primarily, this was due to inexperience and inadequate knowledge in designing a multimedia system. Therefore, when system implementation is undertaken, there were many changes made to the original interface design of the system. Nevertheless, with the tips and techniques in user interface design that was analyze in the system design this problem was overcome.

8.2.4 *Lack of Knowledge in Al-Quran Translation and Tafseer*

Even though there was insufficient knowledge in Al-Quran translation and tafseer but with numerous references on the Internet and in books, compiling the translation and tafseer was easier. On the other hand, since there was countless translator to choose from choosing which one to be used in the system was quite tricky. However, it was decided only four from the best and most accurate was chosen.

8.2.5 *Lack of Skills in Creating Graphical Effects*

Creating graphical effects was a big challenge in this project. Generally, this is due to lack of experience and interest. These effects consume a lot of time and try and error action. Nevertheless, the SIASA have several graphical effects which are simple and does not consume a lot of time to load.

8.3 System Strength

All of the SIASA software assets are discussed below :

1. *High Usability*

The SIASA have a simple yet elegance user interface. This is because the SIASA generally will be use most by students and adults. Since the SIASA was

developed based on HCI principle, the interface is intuitive where users can easily get accustomed to the system functions and style. On the whole, users won't face any problem using the SIASA even though they are computer illiterate.

2. *Transparent*

The system is transparent as users do not need to know how the database performs its search, how the system is structured, etc. As an example, when user needs to search any translation on the surah An-Nas by any translator, they only have to click on the specific function and follow the instruction. They do not need to know how to retrieve information from the database.

3. *Search function*

Since the SIASA only restricted to surah An-Nas, the search function is simple. The user can first choose the translator they preferred, and then select the verse they want to seek for. Subsequently, the result will be highlighted in the surah An-Nas according to the user's request.

4. *Several translator of the surah An-Nas*

The SIASA provide four selections of translator for the translation of the surah An-Nas and two translators for the tafseer. This will advantage the user so that they can compare translation of the surah An-Nas from different points of view.

5. *Several recites of the surah An-Nas*

Besides the selection of translator, the SIASA also provide four selection of qari who will recite the surah An-Nas. Therefore, user can always select the qari they preferred and compare their articulation, 'tajweed' and intonation.

6. *References*

For user references, SIASA endow with a few links and books that could be used by user to enhance their knowledge and to get more information on the translation and tafseer feature in the SIASA.

7. *Presents a few information on Al-Quran and surah An-Nas*

For users with no knowledge of what is Al-Quran, SIASA present a simple introduction on Al-Quran. To look furthermore on Al-Quran, user can utilize the reference provided in the SIASA

8. *Dual language*

SIASA provide user with selection of interfaces in English or Bahasa Malaysia. This is to cater for Malay and non-Malay users.

9. *Users feedback*

The user can submit their comments and opinions to the developer of SIASA via email in the reference section. From the user's feedback, the developer will

acknowledge whether the SIASA is working perfectly in fulfilling user's need and request and changes can also be made to improve the SIASA.

8.4 *System Limitation*

In spite of the strengths discussed in the later section, there are a number of limitations which the SIASA cannot performed due to lack of research and time constrain throughout the development stage. These limitations are as follow :

1. The search function is only restricted to the words provided in the search selection. Users cannot search beyond the verse of surah An-Nas.
2. The scope of the SIASA only covers a small amount of introduction of Al-Quran and surah An-Nas.

8.5 *Future enhancement*

At present, the SIASA only provide a few simple functions for the user to utilize. This is due to time constraint and lecturer's specification. In spite of this, the SIASA could be evolving into a more useful, reliable and complex software in the near future.

Below are several suggestions for the enhancement of the SIASA.

1. To include all the 114 surah in the Al-Quran into the SIASA software. This transition can revolutionize the SIASA into an electronic Al-Quran. Furthermore, it can broaden the scope of translation and tafseer in the SIASA.
2. In view of the fact that there is numerous Al-Quran translator, the number of translator should be increased with the intention that user can broaden their knowledge in Al-Quran translation and tafseer.
3. The number of qari should also be increased in order to enhance users knowledge in Al-Quran articulation, pronunciation and "tajwid".
4. Seeing that articulation, pronunciation and "tajwid" are an important aspect in reciting Al-Quran, therefore functions that will instruct user in the correct manner to articulate Al-Quran can be added into the SIASA.
5. A few other functions that are relevant to Al-Quran can also be added into the SIASA such as quizzes, tutorials, the Prophets history, 'aqidah' and much more Islamic information.

8.6 *Project Conclusion*

On the whole, this project has achieved the objectives and user requirements that have been determined in the system analysis process. It has also managed to be completed within the time given by the lecturer. Moreover, the user can communicate with the SIASA developer via email to convey their suggestions and comments. Furthermore, the response time to display output based on user's request is toleratable. Since the design of the user interface are based on HCI and approved guidelines, the software should be able to use by anyone from the age 7 and above even though they are computer illiterate.

There was scores of knowledge and experience gained throughout this project. It does not cover only the technical and theoretical part of the development process but also enhance the knowledge on Al-Quran generally and surah An-Nas specifically.

There are several new concepts have been learned during the development of this project such as the HCI concept. Learning the VB Script and its environment was priceless knowledge and unforgettable experience. Furthermore, skills in using new software such as Swish 2.0 and Pars Negar II have been acquired and it is also a very significant experience.

Although programming skills are vital in any system development, high-quality practice on software engineering techniques must also be applied proficiently. This is to ensure that not only the system developed will meet all the

user requirements but also a system with high quality standards. This project has provide the requisite opportunity to apply all the techniques, paradigms, concepts and approaches learned from System Analysis and Design, Software Quality and Software Engineering courses.

Conversely, there is still space of improvement that could be done in the SIASA as mentioned earlier. Since the SIASA was development through prototyping model so it will be easier to carry out enhancement process.

In conclusion, this project was a very good opportunity for students to illustrate their skill in developing a system. This is an outstanding practice to guide and prepare students towards the working environment.

References

[1] Bruce, C. S. (1994) 'Research student's early experiences of the dissertation literature review' Studies in Higher Education, vol. 19, no. 2, pp. 217-229.

[2] Bruce, C. (1994) 'Supervising literature reviews', in Zuber-Skerritt, O. and Ryan, Y. (eds), Quality in postgraduate education, Kogan Page, London.

[3] Afolabi, M. (1992) 'The review of related literature in research' International journal of information and library research, vol. 4, no. 1, pp. 59-66.

[4] Bourner, T. (1996) 'The research process: four steps to success', in Greenfield, T. (ed), Research methods: guidance for postgraduates, Arnold, London.

[5] Cooper, H. M. (1988) 'The structure of knowledge synthesis' Knowledge in Society, vol. 1, pp. 104-126

[6] Bruce, C. S. (1990) 'Information skills coursework for postgraduate students: investigation and response at the Queensland University of Technology' Australian Academic & Research Libraries, vol. 21, no. 4, pp. 224-232.

[7] Dena Taylor (2000) 'The Literature Review: A Few Tips On Conducting It' URL: http://www.infocommons.utoronto.ca/writing/litrev.html

[8] Dr. P. Sellapan (2000) 'Software Engineering Management & Methods', Sejana Publishing, Malaysia

[9] F.O.S Inc. & Charles M. Carroll (1997) 'Visual Basic Information' URL: http://www.thebestweb.com/VBFAQS/faq.vbinfo.asp

[10] Scott Berkum (2000) 'The Art of UI Prototyping'
URL: http://msdn.microfoft.com/ui/

[11] Ayean Gulez (2000) 'Macromedia-Tool Information'
URL: http://www.wowwebdesign.com/tools/id_8

[12] October 2001 Techie Toy of the Month 'Swish 2.0'
URL: http://www.techietoyofthemonth.html

[13] Bret Holtz (1998) 'Microsoft Access, Visual Basic; Which to use?'
URL: http://w3.arizona.edu/' ccitinfo/newsletter,mayjune98/extendu.html

[14] Arch640 Development Team (1995) 'Photoshop home Page'
URL: http://www.rice.edu/computer/Tutorials/ravl/pshop//

[15] Adobe System Inc. (2001) 'Adobe Photoshop 6.0'
URL: http://www.pacific.adobe.com/products/photoshop/main.html

[16] International System Consultancy (2000) 'Home of ParsNegar'
URL : http://www.isc.com.au/

[17] Kandall Bass (1997) 'A Brief Guide to Introduction Multimedia and the Study of United States'
URL: http://www.georgetown.edu/crossroads/mltmedia.html#2

[18] 'Business Week Magazine', Feb 28 1994

[19] Cox Family (1996) 'Computer as Teacher and Tutors'
URL: http://www.concentric.net/~skiplac/computer.html#INTRODUCTION

[20] Judy Brown (1997), 'HCI and Requirements Engineering - Exploring Human-Computer Interaction and Software Engineering Methodologies for the Creation of Interactive Software'
URL: http://www.cwi.nl/~steven/sigchi/bulletin/1997.1/brown.html#HDR0

[21] Jim W. Lai (1993) 'Human-Computer Interaction: The User-Interface'
URL: http://www.io.com/~jwtlai/usermodel.html

[22] Jakob Nielsen (2000) 'The usability lifecycle'
URL: http://www.developer.ibm.com/library/articles/nielsen3.html

[23] Muslim Student Organization of UMR (1995) 'Al-Quran Database'
URL : http://www.umr.edu/~msaumr/reference/quran.html

[24] Jawad 'The Meaning of the Glorious Quran'
URL : http://www.quraan.com/Pickthall/Default.asp

[25] Ustaz Ahmad Sohadji Mohammad (1997) 'Tafsir Al-Quran Di Radio', Prospecta Printers Sdn Bhd, Kuala Lumpur Malaysia.

[26] Ahmad von Denffor (1997) 'Interpreting The Text'
URL: http://www.ummah.net/islamaware/Quran/Tafseer/Ulum/Denffor6.html #Principles

[27] Dato' Ahmad Ibrahim 'Maklumat Perundangan Islam: Kenyataan dan Permasalahan', Seminar Sistem Maklumat, Kuala Lumpur, Malaysia

[28] Ambron, S. and K. Hooper, eds. 'Interactive Multimedia: Visions of Multimedia for Developers, Educators and Information Providers Redmond, WA: Microsoft Press', 1988.

[29] Barrett, Edward, ed. 'The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information. Cambridge', MA: MIT Press, 1989.

[30] Bevilacqua, Ann F. "Hypertext: Behind the Hype." American Libraries 20 (2), (February, 1989): 158-162 Delaney, Paul and George Landow. eds. Hypermedia and Literary Studies. Cambridge, MA: MIT Press, 1991.

[31] Galbreath, Jeremy. "The Educational Buzzword of the 1990's: Multimedia, or Is It Hypermedia, or Interactive Multimedia or...?" Educational Technology 32 (April, 1992): 15-19.

[32] Jarvis, Alka and Crandall,Vern. 'Inroads to Software Quality "How To" Guide and Toolkit', Prentice Hall, 1997.

[33] Shari Lawrence Pfleeger, 'Software Engineering Theory and Practice, 2nd Edition', Prentice Hall, 2001.