FACULTY OF COMPUTER SCIENCE &

INFORMATION TECHNOLOGY,

UNIVERSITY OF MALAYA

E-FACULTY:

POSTGRADUATE INFORMATION SYSTEM

A Thesis in

Computer Science

By

Ng Theen Weng

Submitted in Partial Fulfilment of the

Requirement for the Degree of

Bachelor (Hons) of Computer Science

February 2001

ABSTRACT

E-Faculty: Postgraduate Information System

By Ng Theen Weng

The computer industry has gone through a dramatic change since its first introduction, about 30 years ago. Information technology has pushed us to a degree of revolution that we could not have possibly foreseen in the last 30 years or so. Changes include many aspects such as business, games, education, administration and politics and most of all, our daily life. This thesis is about the introduction of the usage of information technology in full force to the local higher-learning institution: university.

In the process, a new system for managing the postgraduate information will be introduced and how it will tackle the latest problems in the administration of this information.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my Project Supervisor, Pn. Rodina Ahmad. Her constructive advice and dedicated assistance throughout the year undeniably played an important role in the successful completion of my final year thesis project. Also to my goodwill moderator, En. Kalid Othman, I would like to show my greatest appreciation for his advice and assistance to review my VIVA and thesis report.

Special thanks to Dr. Mahfuzah Kamsah and Pn. Salimah Moktar who have given me valuable advises directly and indirectly.

My sincere gratefulness to Cik Siti Rohaizah, the system administrator in the FSCIT office. Without her, this project will not be successful. Her cooperation and guidance that has helped me to understand the flow of the postgraduate information in the old system is truly worth of mentioning.

Last but not least, I would like to acknowledge the suggestions and assistance of my fellow course-mates that have contributed to my project in all ways especially those who helped setting up the servers and computers in the laboratory for our usage.

DEDICATION

To my family, for their tolerance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF DATA FLOW DIAGRAMS

Chapter 1: Introduction

1.1 Project Definition

University of Malaya is one of the pioneers in the usage of computers in this country, both management and academic level. Although the fact that its rather new Faculty of Computer Science and Information Technology (FCSIT) is well-equipped with the latest computer technology, many administrative tasks is still carried out manually. There is a lack of office automation systems that can help to ease workloads. Data of students, inventory and others are all kept in physical files; some tasks such as timetable scheduling is still done manually.

E-faculty

Generally, the project *e-Faculty* will prepare an electronic platform in FCSIT where:

i. most of the activities in the faculty such as management, lectures and learning, and student services, will be fully automated or computerized.
ii. the *e-Faculty* is Internet enabled in the sense that information and partial of the system can be reached online.

Postgraduate Information System

Currently, there are over 100 undergraduates who are undertaking Master Degree and more than 50 postgraduates taking Philosophy of Doctorate (Ph.D.) in FCSIT in the University of Malaya. But unfortunately, the current system, which manages all their information, is still in its infancy. The faculty's office kept a copy of the students' information in physical files and processes are manually operated. Therefore, a new Postgraduate Information System (**PIS**) is being introduced here and is planned to meet all the shortcomings of the current system, plus some other extra useful features. The

Postgraduate Information System will be one of the integrated systems in the *e-Faculty* where:

i. it will be served online and can be reached by all authorized stakeholders.

ii. student's data will be kept, processed and maintained in a database through an admin-level interface.

1.2 Objectives

Mission

"E-University is a university that exploits the potentials of technology to meet educational objectives, foster active engagement of students in community building, and develop civic competency and further career aspirations."

Comments:

E-University seeks to integrate current technology into the university's delivery of educational and administrative services to students, to lead and anchor the deployment of technologies, to connect and provide services to other educational, civic, and business entities beyond the campus, and to contribute to formulating and implementing nation-wide plans for acquiring and using technologies.

E-Faculty

i. To reorganize and enhance the main functions in the faculty such as administration tasks, academic activities, student affair administration and service support and some other research activities.

ii. To create a paperless environment in the faculty's management where computers are to replace papers and pens in the administrative operations.

iii. To give the opportunity to final-year undergraduates to develop a system which reaches commercial values for thesis purposes.

Postgraduate Information System

i. To have a better management on the master students' information: - data will be kept in a database and the system will be able to retrieve, edit and even sort the data according to preference.

ii. To be able to sort out students according to their program or other criteria: - this will help co-coordinators to handle and keep track of the student's progress.

iii. To create an interactive and dynamic managing information system: - if there is any violation situation about a student's status or progress such as he/she did not re-register for a following semester, the system will alert the administrator.

iv. To ease the work administration staffs and reduce work redundancy: - administrators need not to work with the students' information manually and sometimes can avoid searching for a particular record over again and again, for example. Tasks that took days traditionally can now be completed in only 1 to 2 minutes, such as listing students according to preference.

v. To enhance the current system from all perspectives, bringing in new utilities such as online registration and online information retrieving.

vi. To build a Postgraduate Information System which hopefully, not only fit for FCSIT usage but for other faculty as well.

1.3 Scope

This *e-Faculty* project is actually a stepping-stone for FCSIT to reach the *e-Faculty* status and also the very first step for University of Malaya to meet the standard of *e-University*. The scope of the project will firstly cover on this faculty (FCSIT). The project will bring attention to some stakeholders such as the faculty's staffs and administrators, lecturers, students and parents and the university's main administrative sections. Initially, the Postgraduate Information System will be used in this faculty only. Stakeholders using it will be the faculty's staffs and administrators, lecturers, students and parents.

1.4 Expected Outcome

The final result of this project will be a complete Postgraduate Information System that can be installed and used by the Faculty of Computer Science and Information Technology. Also there will be a web page of the system, which can be browsed by lecturer, as well as students who wants to look for some information.

Chapter 2: Literature Review

2.1 General Research on E-Faculty

In the age of information technology (IT), computer and cyber, electronically enhanced system is blooming in every aspect of our life. Systems that we worked with everyday are moving fast into half or full automated systems such as office automation (OS), e-mailing system, managing information system (MIS) and a number of other business-related systems.

In western universities, IT is already not a new toy. The concept of *e-university* was first introduced when some if the colleges and universities in the US tried to apply networked computers and telecommunication technology to reach learners at a distance. One of the examples shown (Bee, 1995) is the University of Akron, USA.

Trowler,P. (1998) once generalized the universities according to their ideology into four main categories, which are *traditionalist, progressivist, enterprising* and *social-contructivist*. These universities have been evolving to meet demands of the society, the universities own internal factors or external factors such as the ever-growing Information Technology sector for some time. Of course, not all of these evolutions are at the same par with each other. This is because each university has their own or different ideology and some would not deliberately change to suit some unimportant demands, entirely.

In Malaysia itself, the higher-learning institutions can also be categorized in these four ideologies, generally. Malaysia's public universities fit accordingly to the description of *traditionalist* while private colleges are best to be described as either *progressive* or *enterprising*. Religious-based institutions may be subsumed under the umbrella of *social-reconstruction*.

The use of IT in Malaysia is somewhat in its infancy. What we can observe of the IT usage is the e-government (administration), smart-school (education), CyberJaya and

PutraJaya, both of which are part of the implementation of the Multimedia Super Corridor (MSC) project. The main purpose is to introduce the 'paperless environment' in all area with the idea: a better administrative platform is a better control of information. In the higher-learning institutions, there are not much signs of IT usage especially in the administrative section or wide-campus system; the latter means system such as a complete and working Student Information System that can connect to all stakeholders.

Reengineering (Snover, 1999) is the fundamental rethinking and radical redesign of support processes to bring about dramatic improvements in performance. By applying a methodology know as UPR (University Process Reengineering), the local university is reviewed and the e-university idea is introduced. The BPR (Business Process Reengineering) methodology is also opted for in redesigning the faculty. The idea is to bring IT into the management and administration sectors in order to automate all the work, either is it data processing or managing information.

University Process Reengineering (Tait, 1999), UPR stresses the need for an across-the-board shift in thinking and commitment to innovation. A university must change its focus from a hierarchical structure to one that consolidates related cross-university tasks into cohesive processes, in order to achieve that result. A simple example is when admitting a student that requires more than one person (and more than one department) to find the prospect, review the application, compile the financial aid package, and send the offer letter. An efficient workflow management system speeds this process (and hundreds of others like it) and elevates its quality by automating, simplifying, measuring, directing, and managing the flow of information from department to department across the university.

In addition, the UPR methodology requires the processes and objectives be reengineered. As it is campus wide, an UPR effort must address all parts of the university: jobs, skills, structure, information technology, management systems, business processes and

even values and beliefs (mindset). An undertaking of this magnitude probably seems overwhelming. However, the UPR methodology is scalable; that is, it can be applied within a single department or process (in this case, the faculty), or throughout the organization.

Business process reengineering (Hammer and Champy, 1993) is essentially value engineering applied to the system to bring forth, sustain, and retire the product, with an emphasis on information flow. By mapping the functions of the business process to the faculty's administrative processes, low value functions can be identified and eliminated, thus reducing cost. Alternatively, a new and less costly process, which implements the function of the current process, can be developed to replace the current one. According to Davenport and Short (1990), business process redesign is

"*the analysis and design of workflow and processes within and between organizations*".

Meanwhile, Teng et al. (1994) defines BPR as

" *the critical analysis and radical redesign of existing business processes to achieve breakthrough improvements in performance measures.*"

Davenport and Short (1990) prescribe a five-step approach to BPR:

1. Develop the Business Vision and Process Objectives: BPR is driven by a business vision, which implies specific business objectives such as Cost Reduction, Time Reduction, Output Quality improvement, QWL/Learning/Empowerment. (cf: SharedVision of Senge 1990, Ikujiro & Nonaka 1995).
2. Identify the processes to be redesigned: Most firms use the *High-Impact* approach which focuses on the most important processes or those that conflict most with the business vision. Lesser number of firms uses the *Exhaustive* approach that attempts to identify all the processes within an organization and then prioritize them in order of redesign urgency.

3. Understand and Measure the Existing Processes: For avoiding the repeating of old mistakes and for providing a baseline for future improvements.
4. Identify IT Levers: Awareness of IT capabilities can and should influence process design.
5. Design and Build a Prototype of the New Process: The actual design should not be viewed as the end of the BPR process. Rather, it should be viewed as a prototype, with successive iterations. The metaphor of prototype aligns the BPR approach with quick delivery of results, and the involvement and satisfaction of customers.

Using the scale-downed UPR approach, the e-university is implemented in three levels: macro, meso, micro (Mahfuzah, Salimah, Rodina, 2000). By applying this 'Divide & Conquer' method, the basic level that has to be dealt with is the faculty. Kindly refer to the diagram below:

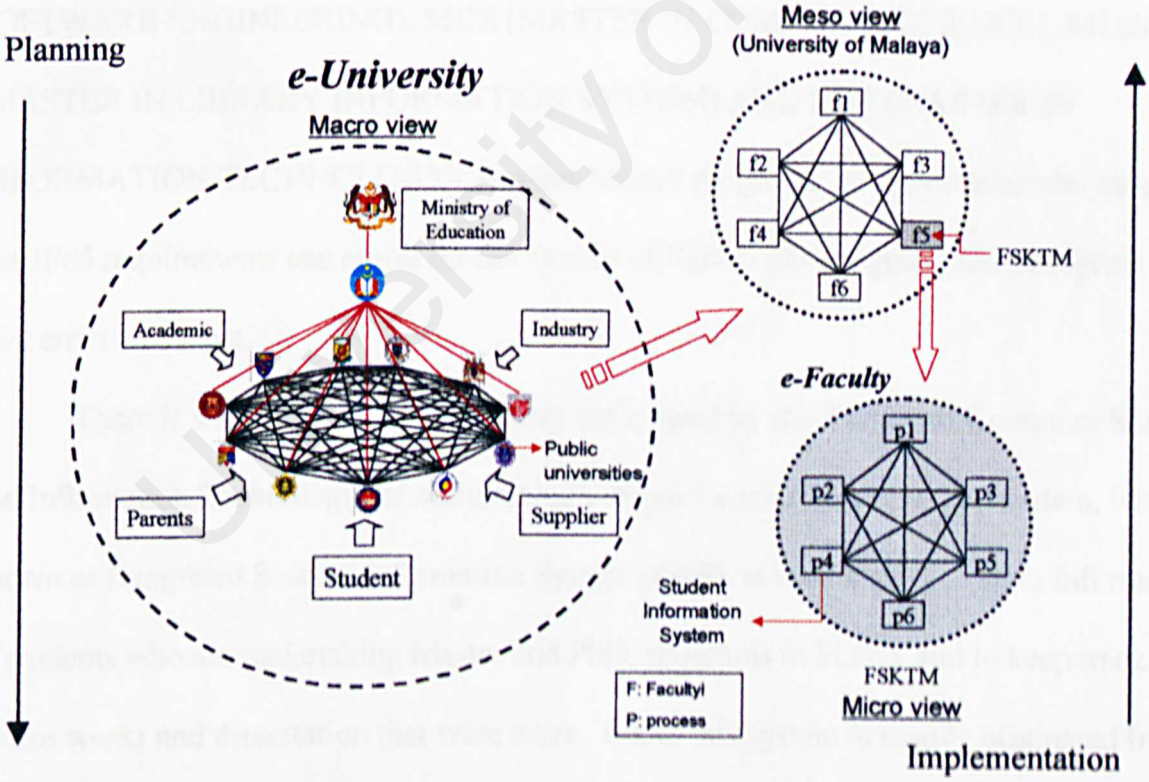


Fig 2.1a: An Overview of E-University

The first phase of the e-faculty will be implemented in **FCSIT**. Adopting the BPR approach: most of the older system of administration will be replace by better systems and some new systems will be introduced so that all can be managed electronically and online.

If success can be monitored, all faculties in University Malaya and hopefully the whole nation's public universities will implement *e-faculty* based on this idea. Only then, Malaysia's public universities can bear the name: *e-University*.

2.2 Description of Postgraduate Information System

There are basically 4 Master programs offered by the Faculty of Computer Science and Information Technology in University of Malaya. They are **MSE** (MASTER IN SOFTWARE ENGINEERING), **MCS** (MASTER IN COMPUTER SCIENCE), **MLIS** (MASTER IN LIBRARY INFORMATION SYSTEM) AND **MIT** (MASTER IN INFORMATION TECHNOLOGY). Besides Master programs, postgraduates who have the qualified requirements can enroll for the Doctor of Philosophy program. Each program has its own criteria to meet.

There is a system, which is currently being used by the Faculty of Computer Science and Information Technology for maintaining postgraduate information. The system, which is known as Integrated Student Information System (**ISIS**), is used mainly to store information of students who are undertaking Master and PhD. programs in FCSIT and to keep track of the course works and dissertation that were taken. While this system is mainly originated from the Postgraduate Research Institution (IPSP) department, the students' data are all kept in FCSIT manually.

Anyway, the current system is still said to be half-manual and half-automated, not interactive and dynamic and also very troublesome. Generally, it is rather like Bachelor Students Information System that performs ordinary administrative tasks but very poor in efficiency. It lacks a lot of features too unlike its counterpart. For example, it could not distinguish which student is taking which master program. Its functions only limited to displaying information of a student, entering course codes and printing a course-registering slip. This slip is the only printing it can offers, which shows some information of a student and the course codes and dissertation title they had registered.

As stated by the Chancellor of University of Malaya in His Majesty's opening speech during the Convocation Ceremony 2000,UM:

"*The overall ratio of the postgraduates to undergraduates is approximately 1:3.7 in UM, as aimed by the Ministry of Education.*"

In FCSIT alone, the amount of postgraduate had increased a few folds from about 40 people in the year 1999 to approximately 120 people in the year 2000. As this trend surely continues, there is a need of better maneuver of information by the system. The co-coordinators of the four master programs are having trouble sorting out students under their supervision with the data all bundled up only in physical files. Under these circumstances, it is very difficult for those co-coordinators to keep track of the students' progress.

2.3 Methodology

Most applications developer and system analyst use a common methodology known as *hard systems approach* to design their systems. Users usually come up with models (Data-Flow model, semantic model, object-oriented model and/or data dictionary) that mirror the desired system. The *hard systems approach* is essentially about **defining the problem solving sequence**.

1. Problem definition. This involves defining the problem and what has to be done.
2. Choice of Objectives. This involves deciding what would be required to reach each objective and formulating the measures of effectiveness, which then form the basis for making comparisons between strategies.
3. Systems Synthesis. This involves identifying the various possible alternative strategies.
4. System Analysis. This involves analyzing and evaluating the various hypothetical systems in the light of the objectives.
5. System Selection. This means choosing the most promising alternative.
6. System Development. This entails developing the chosen alternative up to the prototype stage.
7. Current Engineering. This consists if the realizing of the system, but also includes the essential processes of monitoring the system, feeding this information back to the design stage, and then modifying the system as necessary.

The *hard systems approach* also focuses on **the costs and benefits of alternative programs**.

1. Defining Objectives. This involves clarifying the desired aims and goals.
2. Describing the various alternative techniques or systems available for achieving those objectives.
3. Identifying the cost and resources required achieving each alternative.

4. Developing systems models, in the form of a mathematical or logical framework that can show the interdependence of the objectives, the systems, the environment and available resources.

While this method is quite widely practiced because it really structured the design and is very objective in the process, it will sometimes too rigid for any unexpected conditions to be put in. As a result, some system requirements or software requirements may have to be pulled out or ignored.

There is another method or approach, which is called *Soft System Methodology* (SSM). The *Soft System Methodology* is developed by Peter Checklands and his colleague at the University of Lancaster, under to great influences from Ludwig von Bertalanffy and Sir Geoffrey Vickers. It is more subjective comparing *hard approach*. With soft system approach, system thinking should be regarded as a contribution to problem solving, rather than as a goal-directed methodology and this applies to all situations where the task itself cannot be entirely and objectively defined. As opposed to *hard system method*, SSM gives a more human touch to the system development process, together with the use of SSM tools such as rich picture to describe a system. Peter Checkland says:

"*SSM is a systemic process of inquiry which happens to make use of system models. Thus, it subsumes the hard approach, which is a special case of it, one arising when there is a local agreement on some system to be engineered.*" (Checkland, 1990; 25)

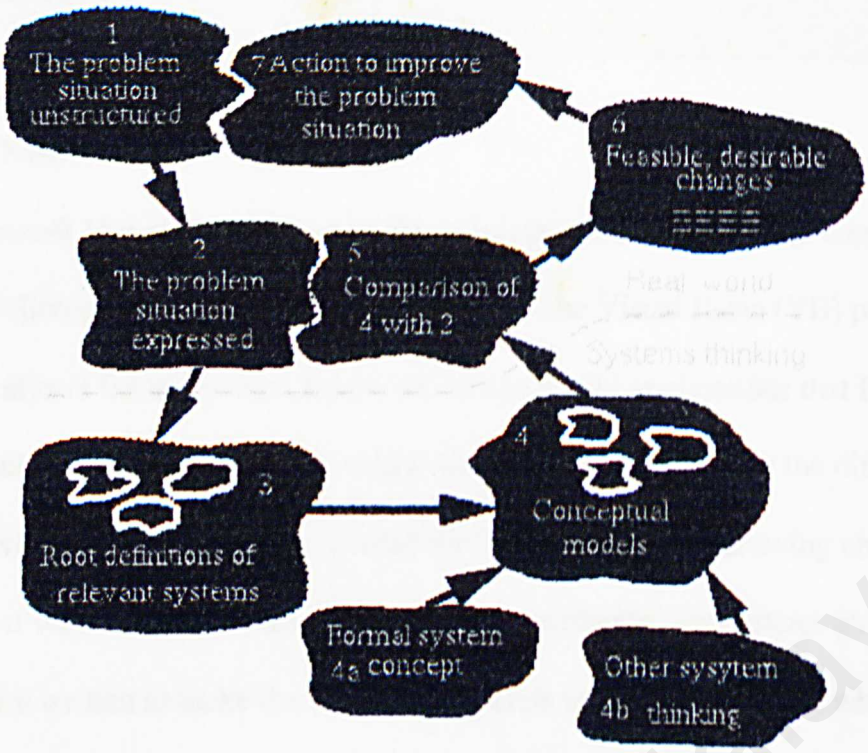


Fig 2.3a: Soft System Methodology

SSM in particular recognizes the "Weltanschauungen" or "world-views" of researcher and subject as both being integral to the investigation. As suggested by its adoption of the German term, "Weltanschauung", this approach is certainly phenomenological, and perhaps philosophically idealistic rather than empiricist.

2.4 Proposed Tools

A) Visual Basic

Microsoft Visual Basic is one of the quickest and easiest ways to create applications for the Microsoft Windows Operating System. The Visual Basic (VB) programming system allows the users to create attractive and useful applications that fully exploit the graphical user interface (GUI). VB provides appropriate tools for the different aspects of GUI development. GUI can be created for the application by drawing objects in a graphical way. Properties can be set to refine the objects' appearance and behavior. Codes are written to make the interface responds to event that occurs when the users react with it.

Using VB, powerful, full-featured application can be created that exploit the key features of Microsoft Windows, including multiple-documented interface (MDI), object linking and embedding (OLE), dynamic data exchange (DDE), graphics and more. VB programming can also be extended by adding custom controls and by calling procedures in dynamic-link libraries (DLLs). The finished application is a true .EXE file that uses run-time DLL that are freely distributed.

There are more powerful database features on VB such as strong connections with Jet database engine and ODBC database engine that are compliant with Access 97 and SQL Server.

VB 6.0 is enhanced with web programming such as VBScript and DHTML that enable programmers to develop not only client-side but server-side applications.

B) Borland Delphi

Delphi combines the power of a true object-oriented language with the convenience of a visual development environment. On its release in 1996, Delphi began to exploit the 32-bit power of Windows 95/98/NT, supporting such Win32 features as multithreading, ActiveX/OLE and the Windows common controls. Delphi began life as a revision to the vulnerable Borland Pascal product line. The differences between Pascal and Delphi are more than cosmetic; in addition to improving the underlying Object Pascal language.

Delphi programs use an object-oriented framework called the Visual Component Library (VCL). It is the VCL that really sets Delphi apart from the competition and brings RAD tools to the next level. Using object-oriented features like polymorphism, encapsulation and inheritance, Delphi VCL provides unparalleled control and power to programmers.

C) Active Server Pages (ASP)

Active Server Pages (ASP) is a great tool for creating dynamic web pages. It is actually not a programming language but rather a 'technology' that allows for the programmatic construction of HTML pages just before they are delivered to the browser. The term 'technology' is rather ambiguous because it is not a programming language (such as C++ or Visual Basic) – although it makes use of existing scripting languages such as JavaScript or VBScript. It is also not really an application (such as FrontPage or Word 97) either.

The power of ASP lies in two facts: firstly, the HTML is not created until the user wants to see the web page (dynamic), and second, it does not care what web browser is being used. It can be executed in any server-computers that support it. Moreover, any

client-computers can view an ASP web page, whether it a computer running on Linux, Unix, Windows or even an Apple Macintosh provided that it has a modern browser.

ASP enabled developers to enhance their web pages with interactive features. It is sensitive to factors such as time and place, and the user's identity and previous choices and actions. It can customize web pages to the specific needs of each individual user.

D) Microsoft SQL Server

All of data processing is involved with the operations of storing and retrieving data. A database, such as Microsoft SQL Server, is designed as the central repository for all the data of an organization. The MS SQL Server database is accessed by users called clients from other computers. Within the framework of a client/server database, a server database also requires integration with communication components of the server in order to enable connections with client systems. Microsoft SQL Server's client/server connectivity uses the built-in network components of Windows NT. It also adds service-specific middleware components--such as Open Database Connectivity (ODBC) on top of the network components. ODBC enables the interconnection of different client applications without requiring changes to the server database or other existing client applications. The Microsoft Open Database Connectivity model (ODBC) uses SQL to connect to databases even when the underlying database does not natively support SQL. In those cases, SQL is translated into a set of commands that accomplish the requested call for the given database.

The core components of SQL Server are the relational database and its structure. SQL Server allows up to 32767 databases definition, a limit constrain that will hardly be encountered. On top of that, some 2 billion tables can be defined in each of the 32767 databases mentioned. However, most typical systems have no more than several

hundreds tables in a database. For each table can be fitted with 250 columns. The number of rows in a table is effectively limitless for SQL Server. In practice, the only limitation is the capacity of storage medium. SQL Server allows databases to expand to include up to 32 physical disks.

Structured Query Language (SQL), the query language developed by IBM in the 1970s, has become the de facto standard database query language for relational databases. The dialect of SQL that are used with SQL Server is Transact-SQL, which Microsoft implements as a core component of SQL Server. Transact-SQL adds to the original SQL keywords for the formation of data retrieval, storage, and manipulation statements.

E) Oracle

The Oracle database system can run on more than 100 different hardware and operating system platform (including all different versions of UNIX). Oracle can support databases ranging in size from a few megabytes to hundreds of gigabytes. The database files can reside either on hard-disk drives or on CD-ROMs, which can be particularly useful for archival or historical data.

Without any extra application development effort, an Oracle database can support from one to hundreds of users with the necessary data locking and protection. There also a large range of tools that provides front-end access to Oracle database.

The backup and recovery feature provided in Oracle enables a system to be available 24 hours a day. Oracle also supports a wide range of client/server machines, offering a tremendous choice in platform for database engine (the server) and front-end programs (either Oracle front-end programs or non-Oracle tools). It also supports a wide number of network protocols and topologies that allow client and server machines to communicate.

F) Seagate Crystal Report

Besides acting as a powerful stand-alone report creating application, Seagate Crystal Report provides a report module that can be added to applications. As a developer using C, C++, Visual Basic, ObjectVision, Turbo Pascal, Delphi or any other programming language that can access a DLL, sophisticated report generating and printing capabilities can be added to applications without the time-consuming task of writing codes.

2.5 Summary

In typical Software Engineering, system development lifecycle (SLDC) can define the process of a system design and development in a quite accurate manner. Throughout the lifetime of computer era, it has never fail miserably in the development of any software or system that we see nowadays. This robust approach or methodology is the contributions of renowned researchers like B. W. Boehm, P. Coad and Michael A. Jackson. There are the object-oriented approach, status-oriented approach and function-oriented approach.

A totally different technique is the Soft System Methodology or SSM introduced by Peter Checklands. Besides giving system development a more human touch, it also looks at the environment of the system. It is important to view the system development in every aspect, be it hard approach or SSM. This is because system developers should apply all kind of methodologies to ensure that the final product is satisfactory.

With Waterfall Modeling stands as the background for system development, I will also use both hard and soft approach to design the **PIS**. Because of the property of the Postgraduate Information System, the use of function-oriented approach is more appropriate as it is mostly data processing. **SSM**, which is now getting more popular among system

developers in the UK, is also used to help define and analyze the related system. SSM tools such as rich picture is some of the helpful tools that enable me to recognize the current system flow.

As for development software, I will be using Microsoft's products such as Visual Basic 6.0, SQL Server 7.0, Active Server Pages (ASP) and maybe Seagate Crystal Report 6.0. ASP serves as the HTML platform for building dynamic web pages. It is also easy to use as it only utilizes VB scripting property. The selection on Visual Basic 6.0 together with SQL Server 7.0 is because of the tight integration and compatibility among Microsoft's product.

Chapter 3: System Analysis and Specification

3.1 Scheduling

As included in the appendices is the Postgraduate Information System development planning and procedure on preparing the system and the documentation. The schedule is constructed entirely using Microsoft Project 2000. Kindly refer Appendix A.

3.2 Problems Analysis

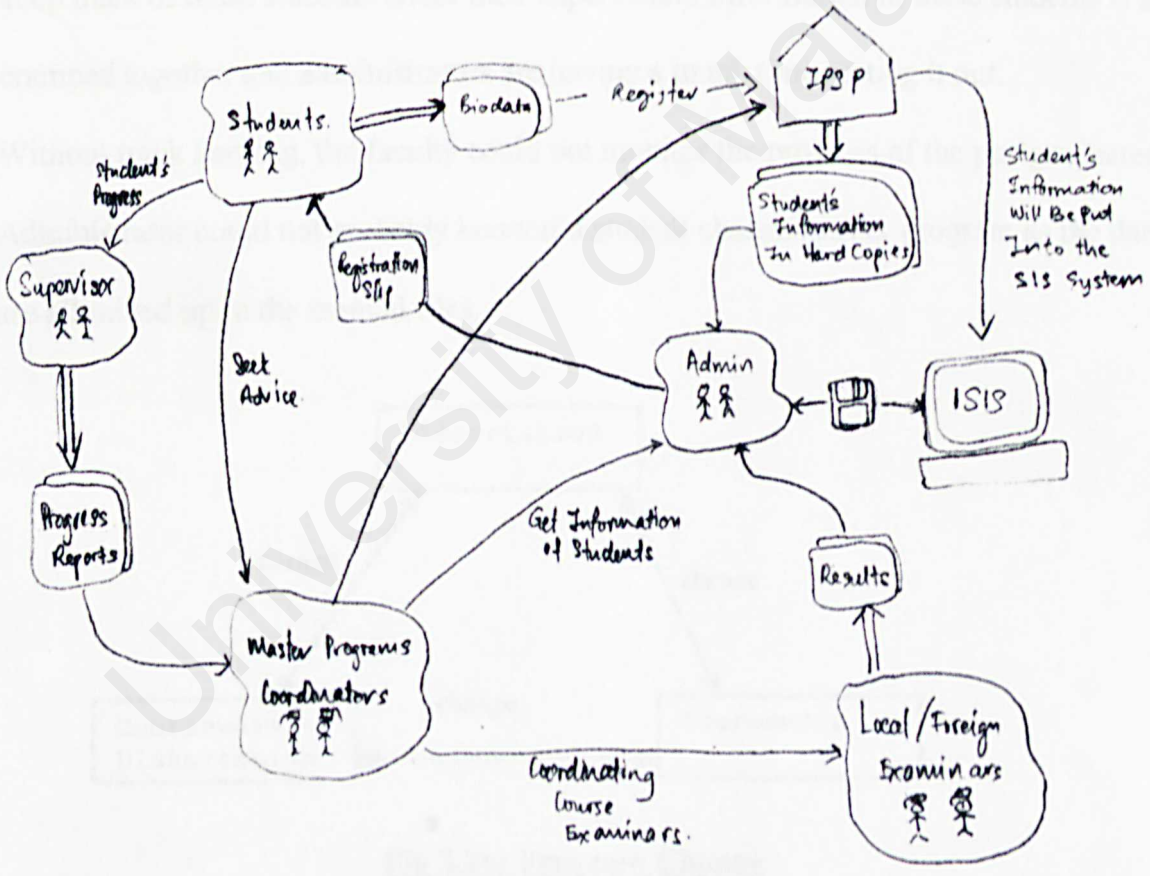


Fig 3.2a: Rich Picture of the current Integrated Student Information System (ISIS) for Postgraduate

From the rich picture, it can be seen that the system is still very primitive and half-manual half-computerized. Information of postgraduates are still kept in physical files and managed manually by administrators in the faculty side. This aroused many problems from different aspects and angles.

- There is a computerized system (ISIS) which data are directly transferred from the IPSP. But administrator can only depict one particular student's information according to IC number or matrix-card number. They can only display, print and add in course codes taken by that student.
- The co-coordinators of the MCS, MSE, MLIS and MIT or even Ph.D. programs could not keep track of those students under their supervision. Information of these students is all cramped together and administrators are having a tough time sorting it out.
- Without track keeping, the faculty could not monitor the progress of the postgraduates.
- Administrator could not probably know if a student change his/her program as the data are all mixed up in the manual files.

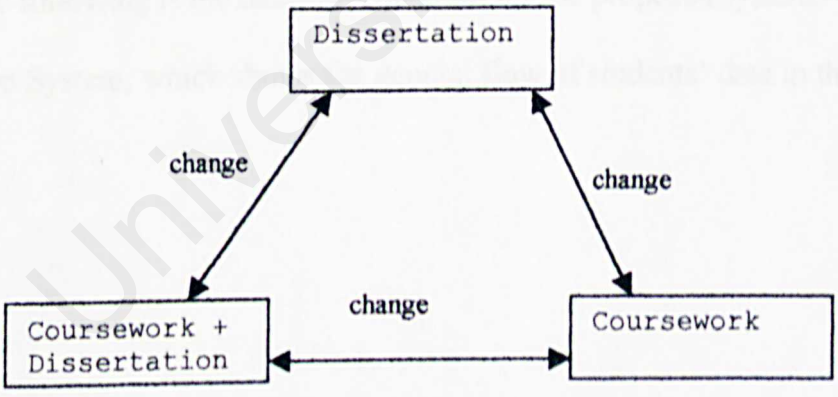


Fig 3.2b: Program Change

- Any major/minor problems with a student and his information could not be discovered and alerted until his particular are come across or when he comes to see the administrators.

3.3 Proposed System

The new Postgraduate Information System is proposed to have a database and two interfaces that interact with it. The database will be put in a server running with MS SQL Server on the Window NT operating system.

The first interface will be the admin-level application placed in the faculty's office. It is for administrators to key in and manage the students' data. The students' personal will have to be keyed in here to ensure a better security of information. The system also allows allocation of work such as tutoring to postgraduates and notices will be mail to respective postgraduates.

The second is the web page for the Postgraduate Information System. Its prime offering will be the online checking of course works and dissertation taken. It is also an information-driven page as all sorts of related information can be obtained here. Students only can obtain personal information but also the latest news and messages from the faculty.

The following is the data-flow diagram for the proposed system – Postgraduate Information System, which shows the general flow of students' data in the system:

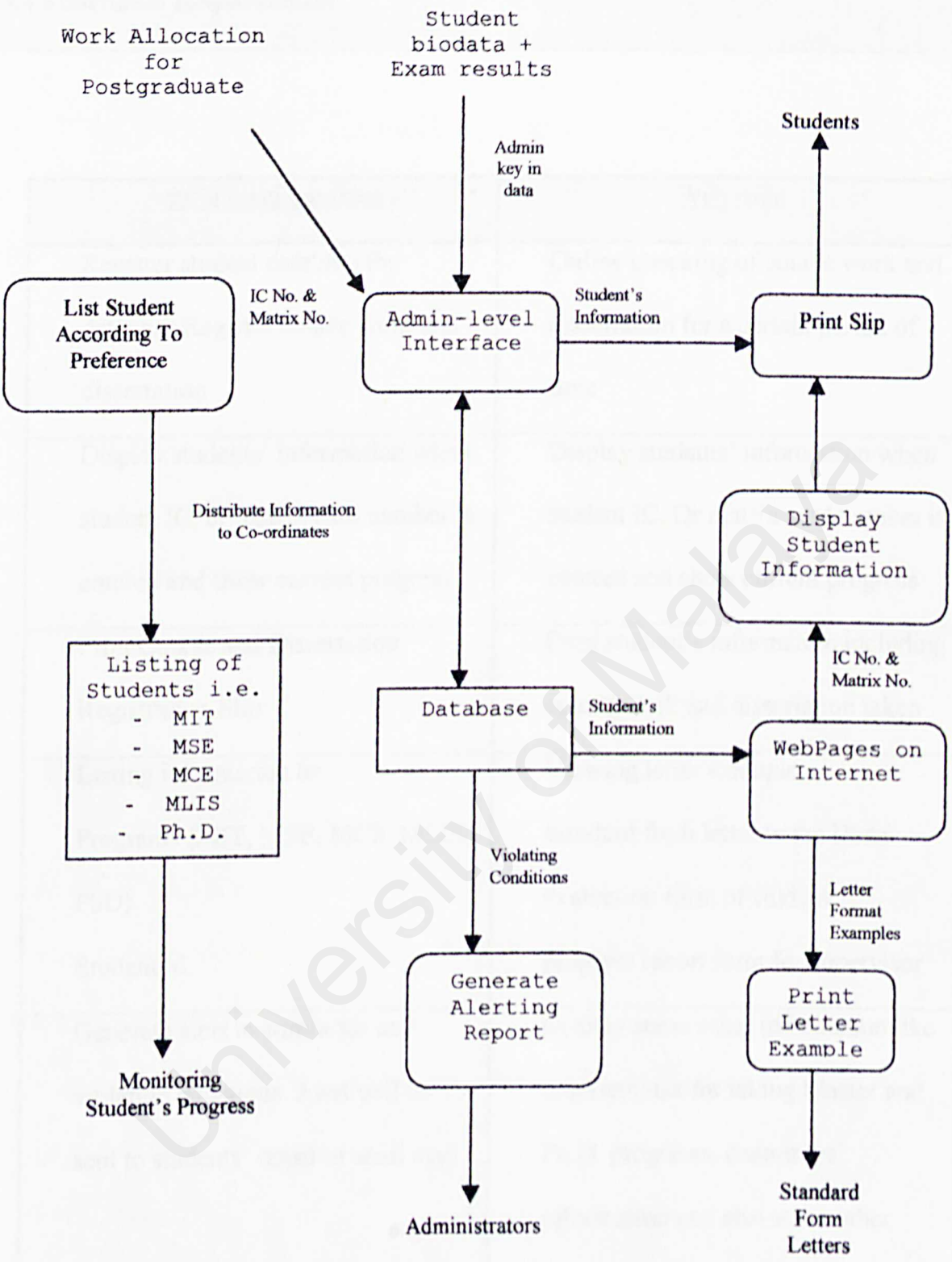


Fig 3.3a: Postgraduate Information System Data-Flow

3.4 Functional Requirements

Admin-level interface	Web page
Register student data into the database Register course work and dissertation	Online checking of course work and dissertation for a certain period of time
Display students' information when student IC. or matrix card number is entered and show current progress	Display students' information when student IC. Or matrix card number is entered and show current progress
Print Course and Dissertation Registration Slip	Print student's information including course work and dissertation taken
Listing information by Programs (MIT, MSE, MCS, MLIS, PhD) Student id.	Viewing letter examples: standard form letter to the Dean evaluation form of student progress report form for supervisor
Generate alert to admin for any violation situations. Alert will be sent to students' email or snail mail	Display some other information like requirements for taking Master and Ph.D. programs, examiners' information and also some other links (FCSIT homepage)

Table 3.4a: Functional Requirements

3.5 Non-functional Requirements

The system is speculated to have a respond time of less than 1 minute for displaying data and less than 2 minutes for listing data according to preference. The dependence is on the amount of data being processed. The system will probably need approximately 2 gigabytes of hard-disk space and 32 megabytes of memory. (subject to the current changes of database size)

Administrators should be able to use the system after training and knowing for an hour or so. Help files and instruction manual can be referred occasionally.

It is also speculated that the system will fail once each 5 years of operation time if the database is not maintained properly. This due to the database corruption although the well-known stability of a Microsoft SQL Server database. The system will be available everyday provided that the e-faculty server is switched on all the time.

The web page will be available 24 hours a day. Accessing time will depends on the communicating network. Although the dynamic web page is already processed by the server upon user request, the user still cannot get it as the line is busy especially peak hours. Thus, the system accessing is slowed down accordingly.

All requirements stated are only a minimum guideline. Of course, the higher the better for the system to full flex.

Hardware requirement:

- minimum CPU speed: 166 MHz
- minimum RAM: 32 MB/ PC66
- minimum HDD: 1 GB
- Internet enabled

Software requirement:

- Operating system: Window 95 or higher
- Seagate Crystal Report 6.0

Server:

- WindowsNT 4.0, Windows 2000 Server
- MS SQL Server 7.0

3.6 VIVA Feedback and Comments

During VIVA, the feedback and comments received for the proposed system is that the respond time is not stated in detail and clearly. One extra function is to be added which is the work allocation (tutors) for postgraduates in the faculty.

4.1 Introduction

When building a house, engineers need a blueprint. To build something even complex such as a software system, software engineers also need a blueprint – a design. Design sits at the technical kernel of software engineering and is applied regardless of the system process model that is used. Beginning once system requirements have been analyzed and specified, software design is the first of the three technical activities – design, code generation and test – which are required to build and verify the software system. Each activity transforms information in a manner that ultimately results in validated computer software.

In the process of designing the PIS system, three kinds of modeling have been used. They are Entity-Relationship Diagram, Data Flow Diagram and Data Dictionary. In addition, flow-charts are used to show the overall flow of the system.

Entity – Relationship Diagrams

E-R diagram is a detailed, logical representation of the data for an organization or for a business application. The representation is expressed in terms of entities in an environment whether it is business or organizational, the relationship between entities and the attributes of both entities and relationship.

Data Flow Diagram

Data Flow Diagram (DFD) is a technique used to show graphically the flow of data through a system and the processes that is performed by the system. The DFD provides an overview of the system inputs and outputs, processes and the flow of data through each process.

Data Dictionary

A Data Dictionary, also known as Catalog or Repository, acts as storage for metadata of an information system. Metadata is data about data. The data dictionary stores information about entities and their relationships with one another, attributes of entities, primary and foreign keys, validation rules and triggers.

4.2 System Architectural View

The Postgraduate Information System's architecture is divided into two structures, a window-based application and a web-based application. The window-based application is the kernel of the system where most information management functions can be found. Basically, it allows add and change of the postgraduate information. It also manages the Master course codes and registrations. Each year, postgraduates are ought to re-register and PIS can keep track all the students who did or did not re-register eventually. Administrator can also perform a search to list out all the students based on some criteria such as: -

- All the postgraduates
- All the postgraduates who are in MCS, MSE, MIT and MLIS
- All the postgraduates who register under a certain session and semester and etc.

The second application is the system web page. It consists of a main page that offers a variety of information related to the postgraduate studies in the Faculty of Computer Science and Information Technology in University of Malaya and also acts as a login page for two categories of users who are students and coordinators (lecturers). Each login categories has their own functions and information related to them.

4.3 Web Page Design

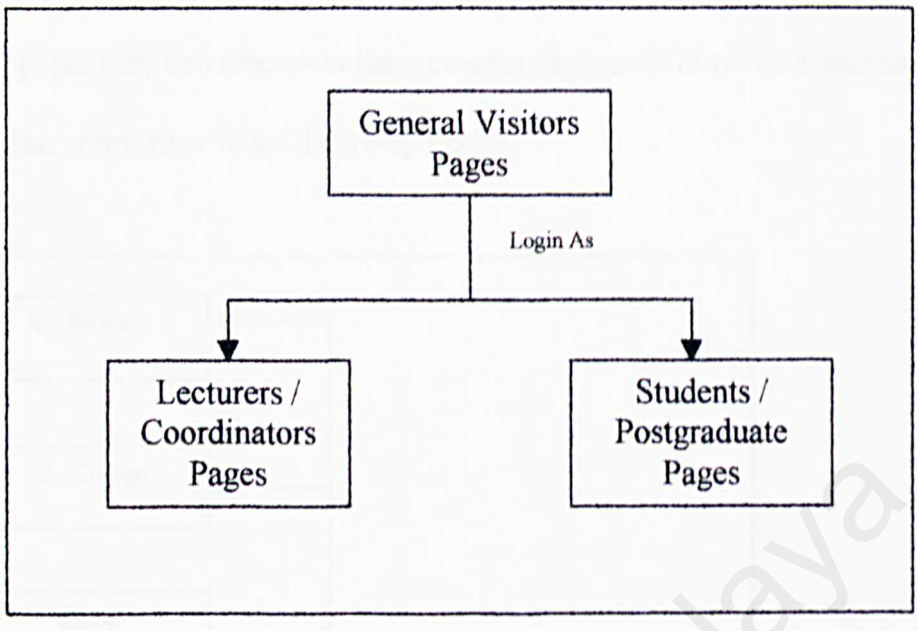


Fig 4.3a: General Flow of the Web Page

The PIS web page is divided into three main structures as illustrated in Fig4.3a. General visitors can roam about in the General Visitors Pages that offers a lot of information regarding postgraduate studies. Fig 4.3b shows the pages that are available for selection.

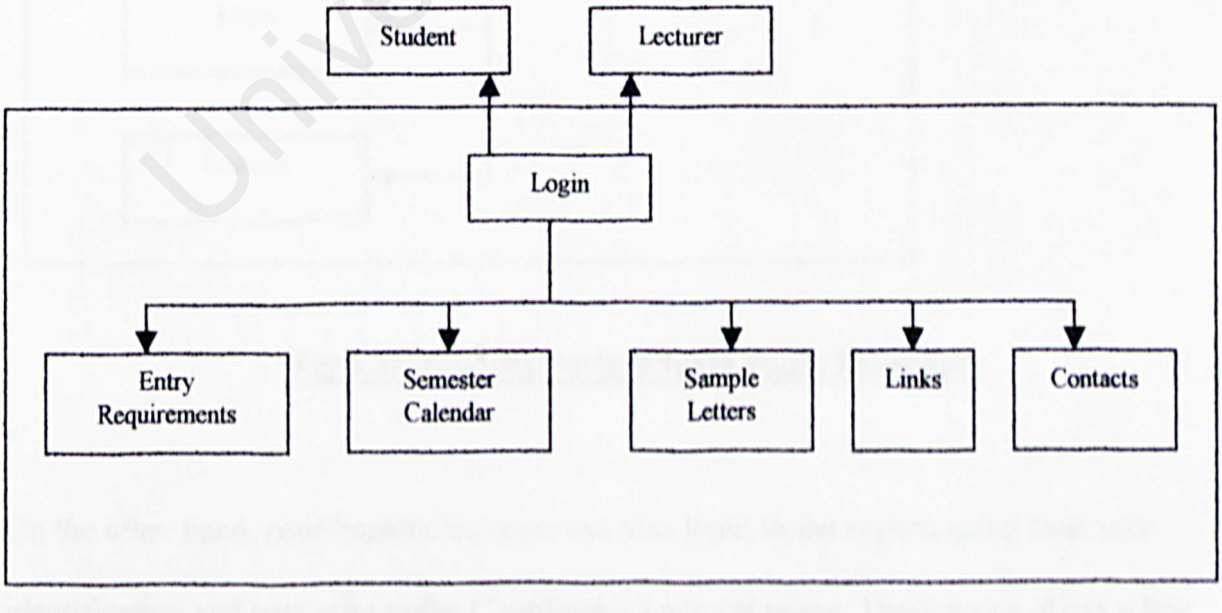


Fig 4.3b: General Visitors Pages Flowchart

Through the login page, when a student (postgraduate) login using his/her matrix number and pre-selected password it will bring him/her to their personal information details page. In this page, they can also view their courses registered over the semesters and possibly results too. The entire flow is as illustrated below.

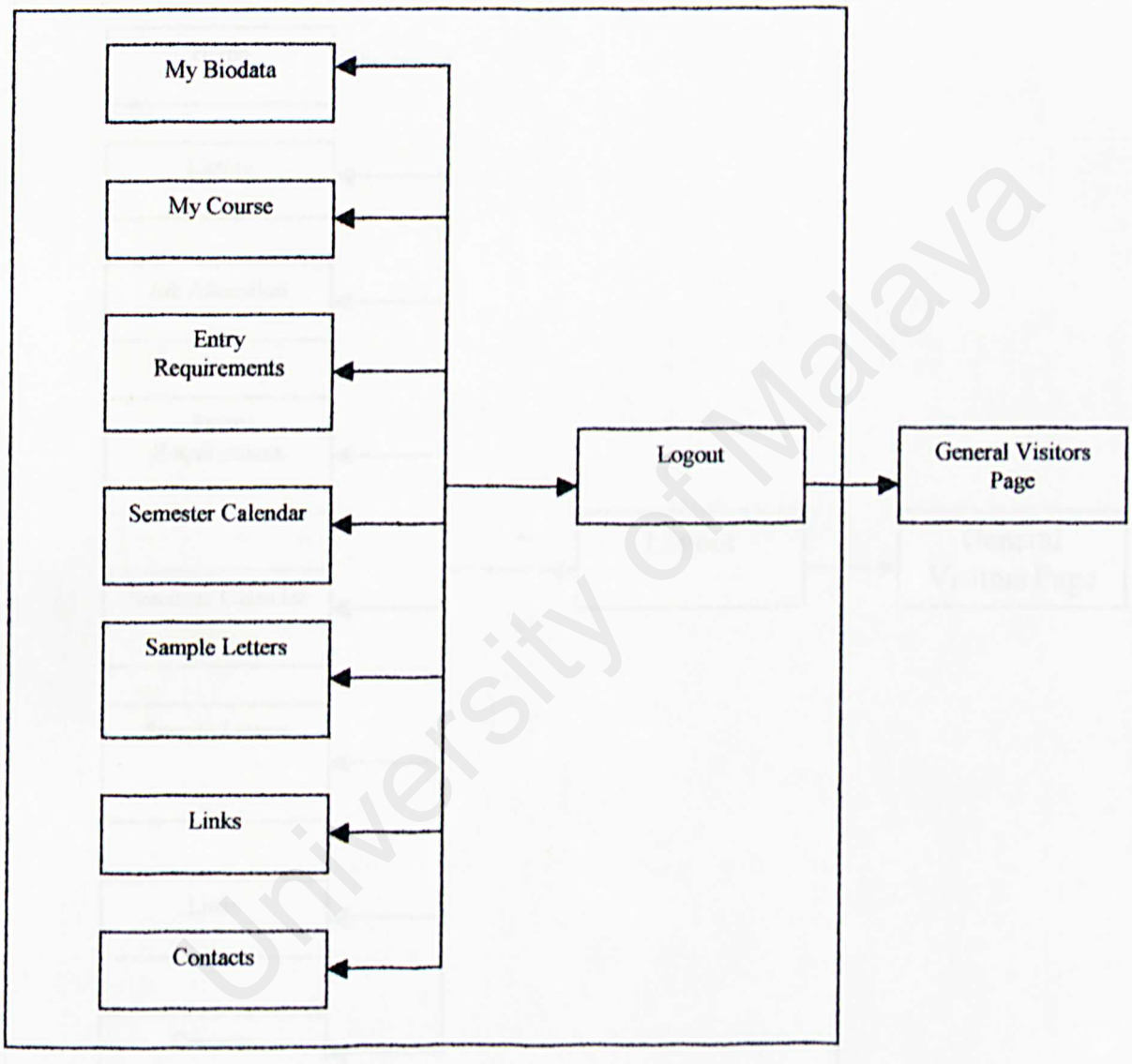


Fig 4.3c: Student/Postgraduate Pages Flowchart

On the other hand, coordinators/lecturers can also login to the system using their user identification and password to the Coordinator/Lecturer pages. These pages offers a few functions for coordinators to search for information regarding the postgraduates such as the

courses they had registered, dissertation titles, listing of postgraduates by session and semester and so on. There is one more special function for allocating tutoring assignment to postgraduates who are working for the faculty as part-time. The overall setup is as shown in Fig 4.3d below.

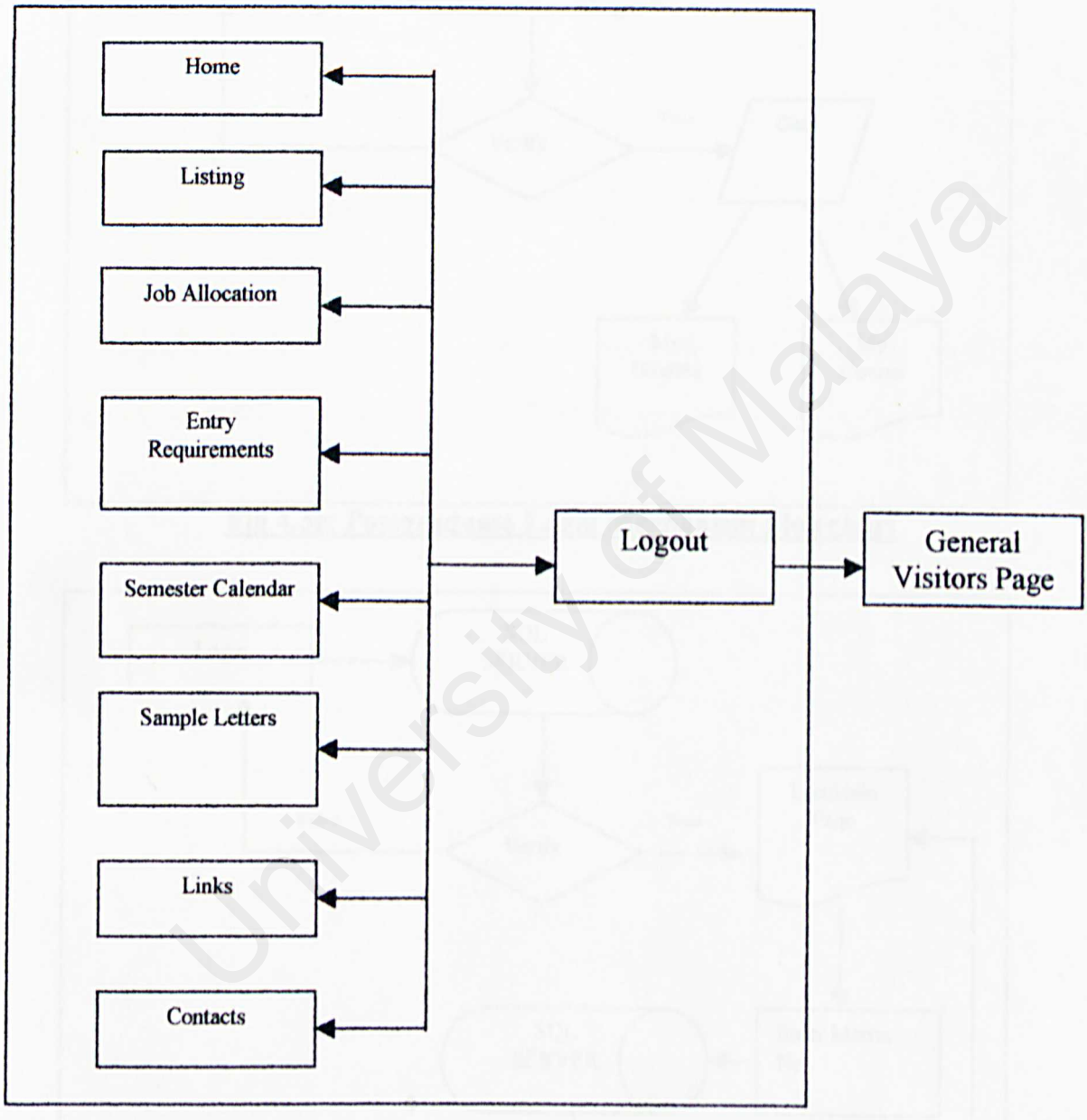


Fig 4.3d: Coordinator/Lecturer Pages Flowchart

Below are the two login flows of the web page system. The first is the login for student and the second is for the coordinators.

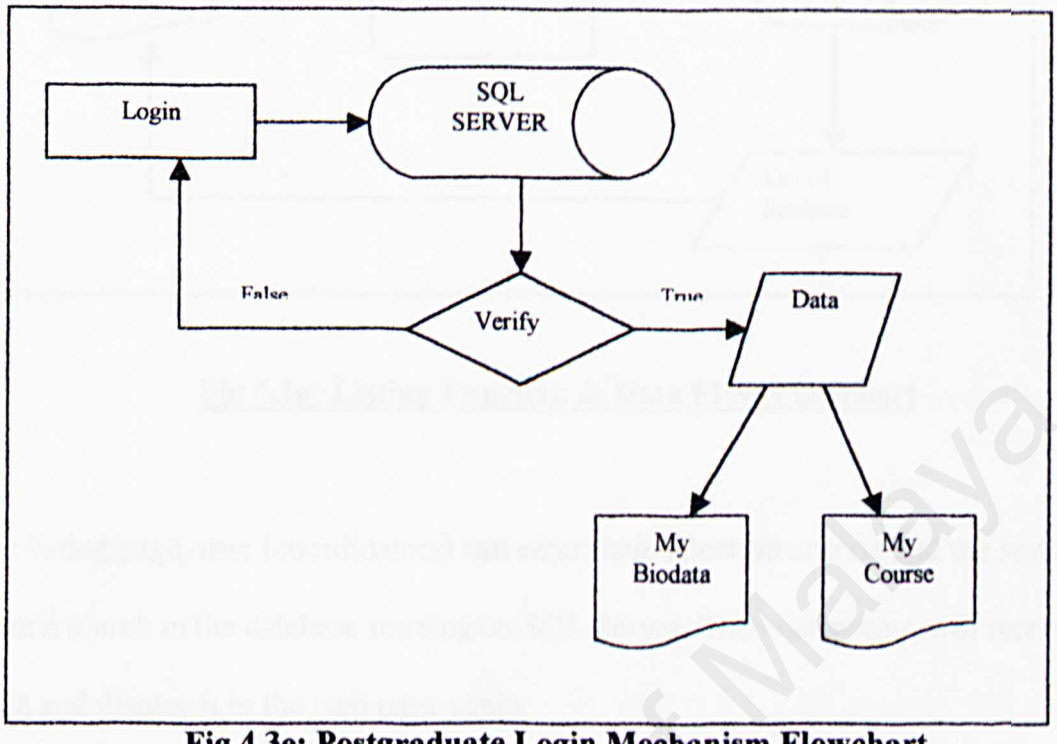


Fig 4.3e: Postgraduate Login Mechanism Flowchart

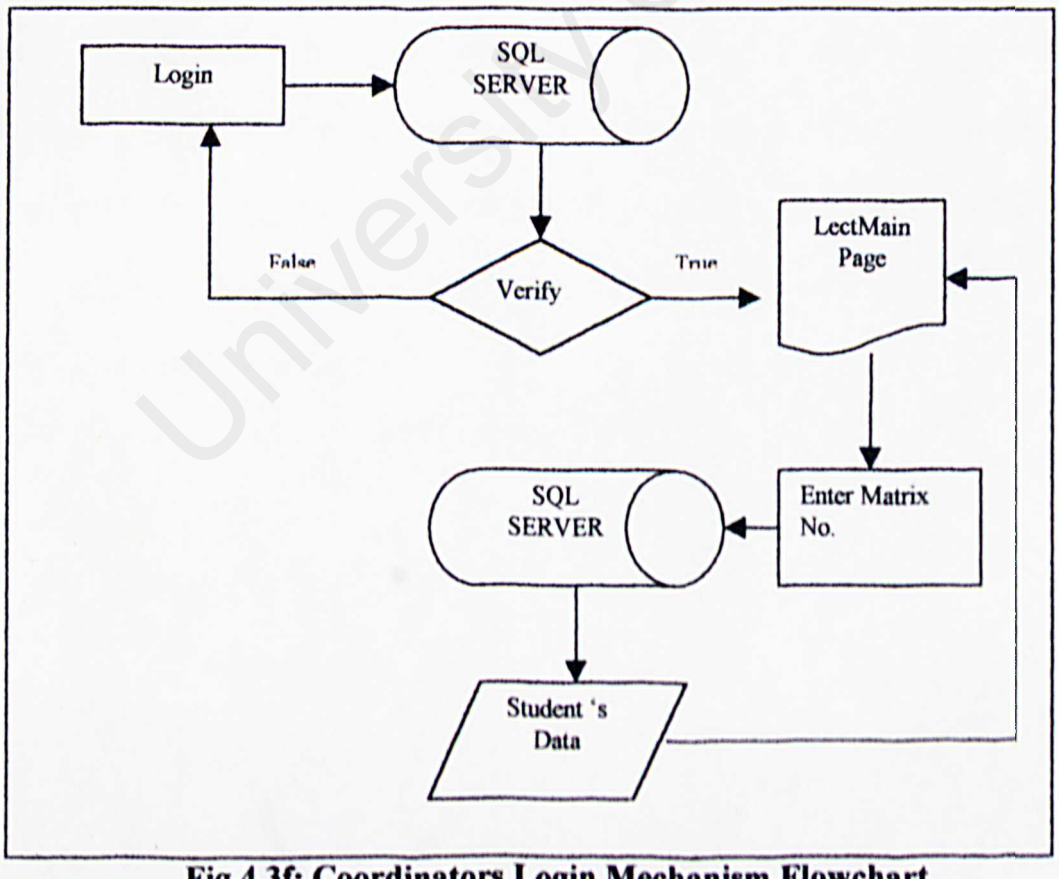


Fig 4.3f: Coordinators Login Mechanism Flowchart

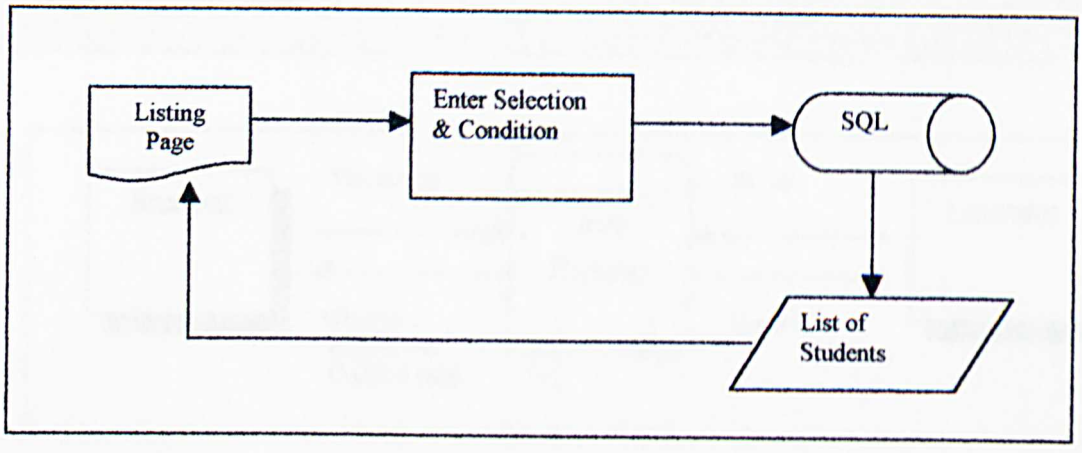
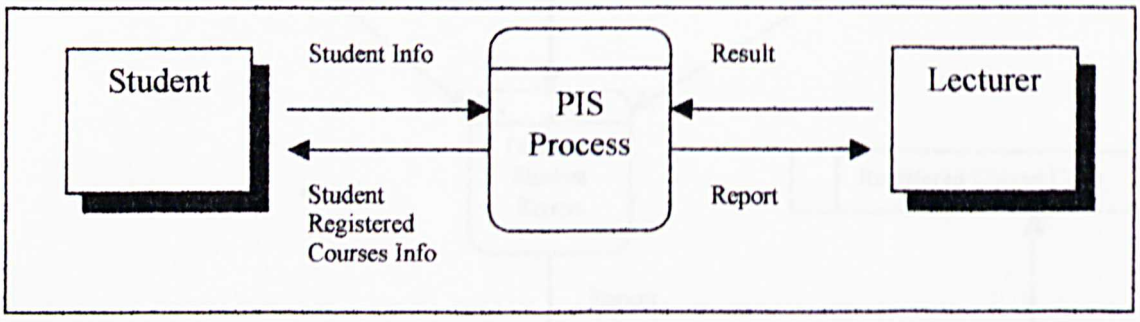


Fig 4.3g: Listing Function & Data Flow Flowchart

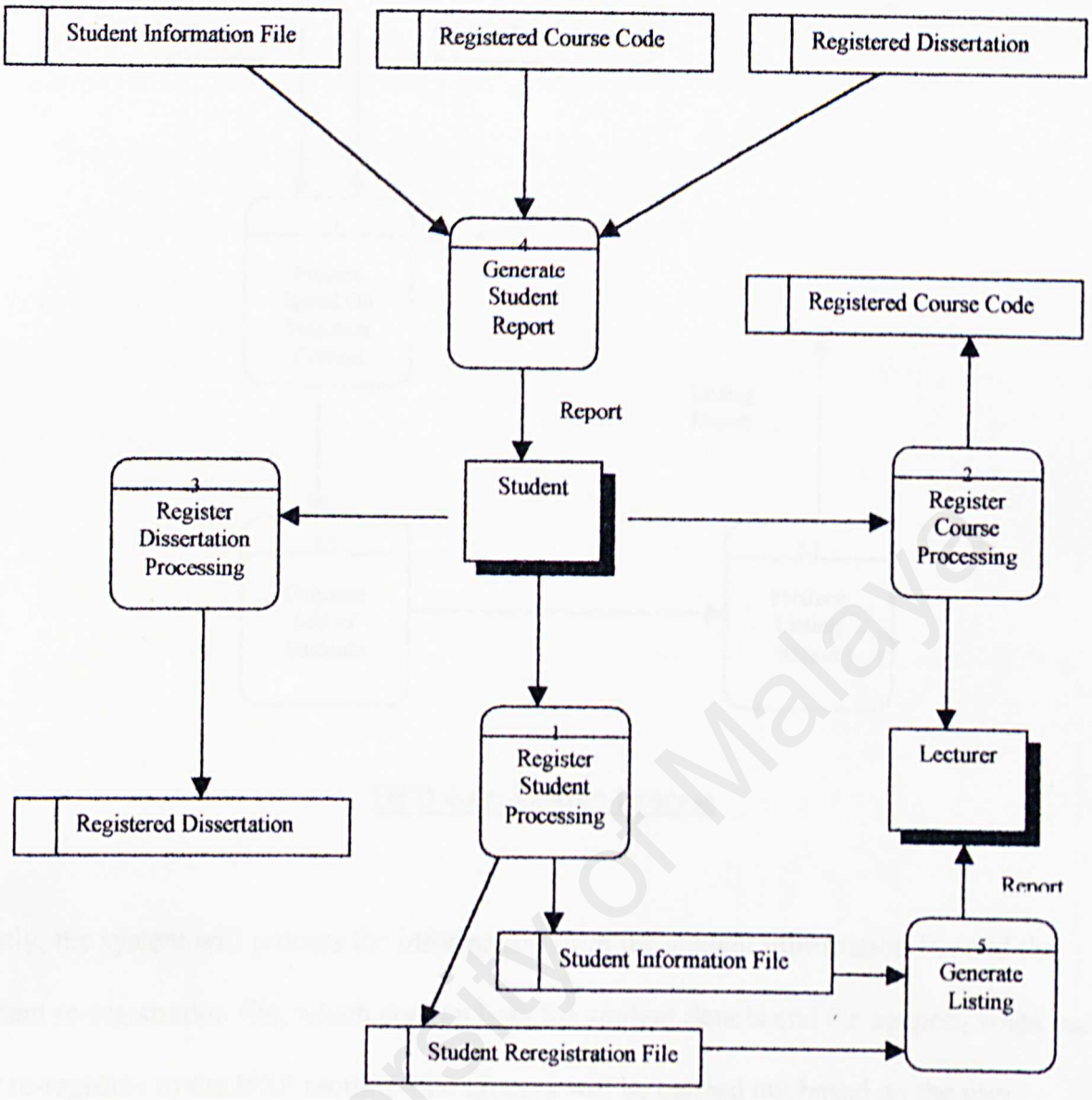
On the listing page, user (coordinators) can enter their selection criteria and the system will perform a search in the database running on SQL Server. Then the system will receive back the data and display it in the web page again.

4.4 Application Design



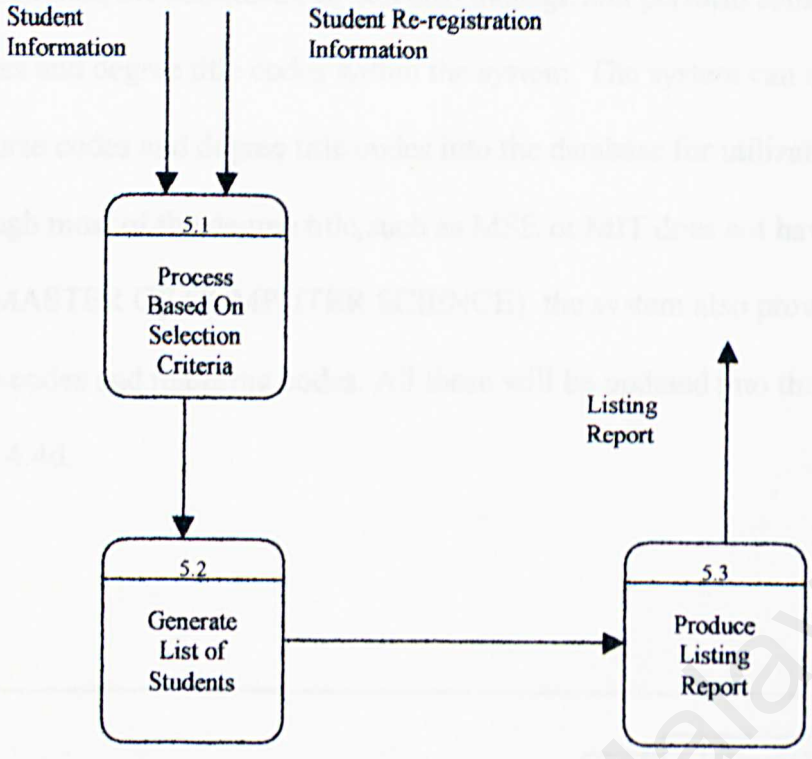
DFD 4.4a: Level 0 DFD Postgraduate Information System Application

Postgraduate actually are students taking Master Degree and Doctor of Philosophy. So, there are actually three modules of application in PIS: PGIS, PGIS2 and Maintenance File. PGIS is for Master Students while PGIS2 is for Ph.D. students, and also Maintenance File for master-files maintenance purpose. The PIS process involves two main entities: the student and lecturer/coordinator. The student (postgraduate) will provide the student information into the system. At the same time, the system can print out registration slip and course registration slip to students. Lecturers will pass in the examination result to the system. The system can also provide listing functions for coordinators to list the postgraduate and their information when required.



DFD 4.4b: Level 1 DFD PIS Application

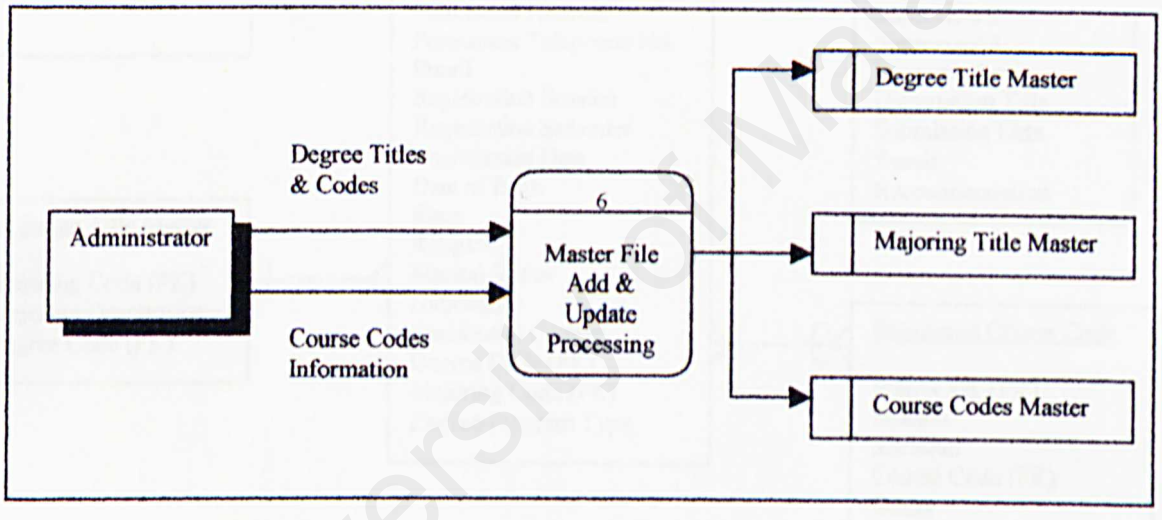
The data-flow diagram above show a more detail flow of data around the system. It consists of three main processes: the register student process, the register course process and the register dissertation process. There is also a general reporting process and a listing process which mainly for coordinators. Actually, the system administrators will maneuver all the processing. The general reporting process basically reports the student's details and information while the listing process is as shown below:



DFD 4.4c: Listing Process

Firstly, the system will process the information from the student information file and the student re-registration file, which contain both the student details and the session, when each one re-registers to the IPSP section. The process will be carried out based on the user selection and criteria. Then, a list of students with relevant information will be generated and then will be produced in a form of soft copy (which can then also be printed out) eventually.

Afar from that, the administrator can also manage and perform some maintenance on the course codes and degree title codes within the system. The system can actually store and update any course codes and degree title codes into the database for utilization within the system. Although most of the degree title such as MSE or MIT does not have any majoring except MCS (MASTER OF COMPUTER SCIENCE), the system also provide links between the degree title codes and majoring codes. All these will be updated into three master-files as shown in DFD 4.4d.



DFD 4.4d: Maintenance File

4.5 Database Design

A complete software system usually is linked to a database to store all the data. PIS will be using MS SQL Server as its database system. The E-R diagram below shows the relationship between each of the entities existed.

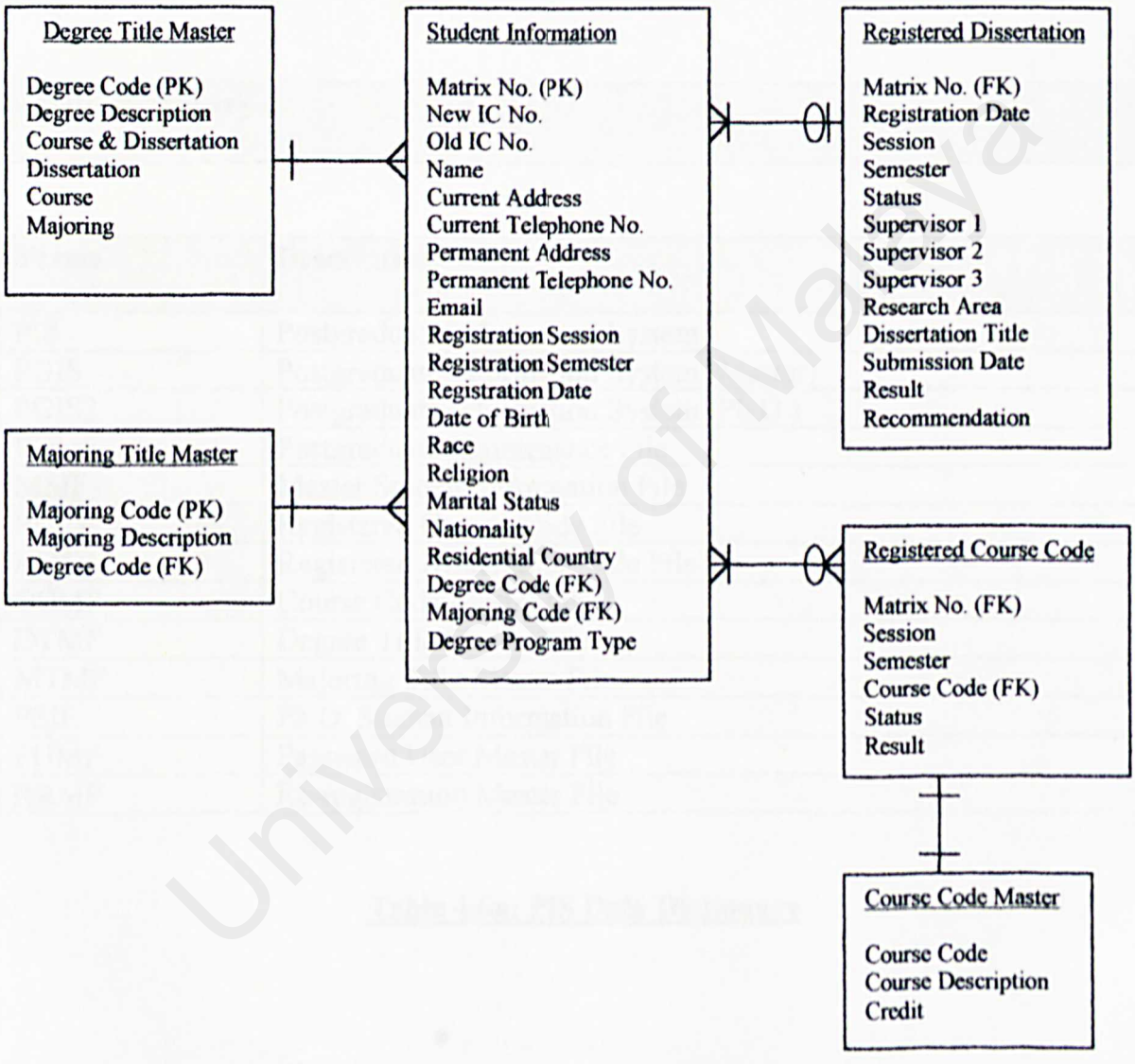


Fig 4.5a: Entity-Relationship Diagram of the PIS Database

As it appears above, there are approximately six tables in this database. The student information table will store each student's details. Degree Title Master, Majoring Title

Master and Course Code Master are to store all the degree title code, major title codes and course codes respectively. The registered course code table is to keep track what course each student has taken on a certain session and semester. The same goes to registered dissertation table but it also keep some other important information such as datelines, supervisors and some comments.

4.6 Data Dictionary

Terms	Description
PIS	Postgraduate Information System
PGIS	Postgraduate Information System (Master)
PGIS2	Postgraduate Information System (Ph.D.)
PGMF	Postgraduate Maintenance File
MSIF	Master Student Information File
RCCF	Registered Course Code File
RDTF	Registered Dissertation Title File
CCMF	Course Code Master File
DTMF	Degree Title Master File
MTMF	Majoring Title Master File
PSIF	Ph.D. Student Information File
PUMF	Password User Master File
RRMF	Re-registration Master File

Table 4.6a: PIS Data Dictionary

4.7 Screen & Interface Design

The web page will be having a login section on the main page as shown below. On the right side, it will be the bulletin section where important messages concerning postgraduate are published from time to time.

Fig 4.7a: Web Page of PIS

The window-based application on the other hand, is divided into three parts. There is a main application-selecting screen for user/administrator to choose from.

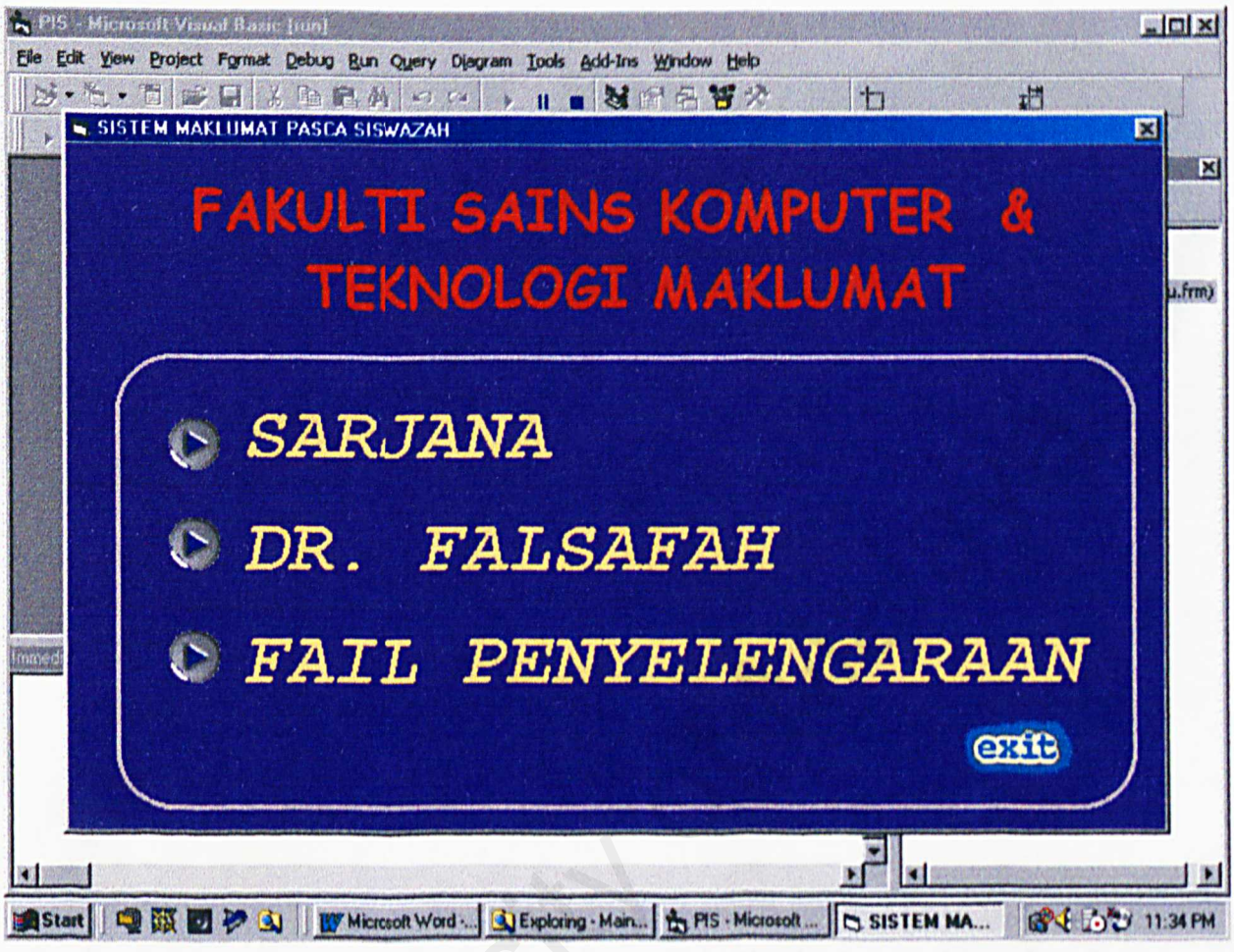


Fig4.7b: PIS Application Main Menu

Once selected, the system will prompt for a login form and will go to the application module after passing the userid-password authentication. The following are the background of the application module and the student's details form.

Fig 4.7c: PIS Application Background Interface

After logging into the system, the first interface will be this background interface. It comprises of the system title (Postgraduate Information System) and the faculty logo.

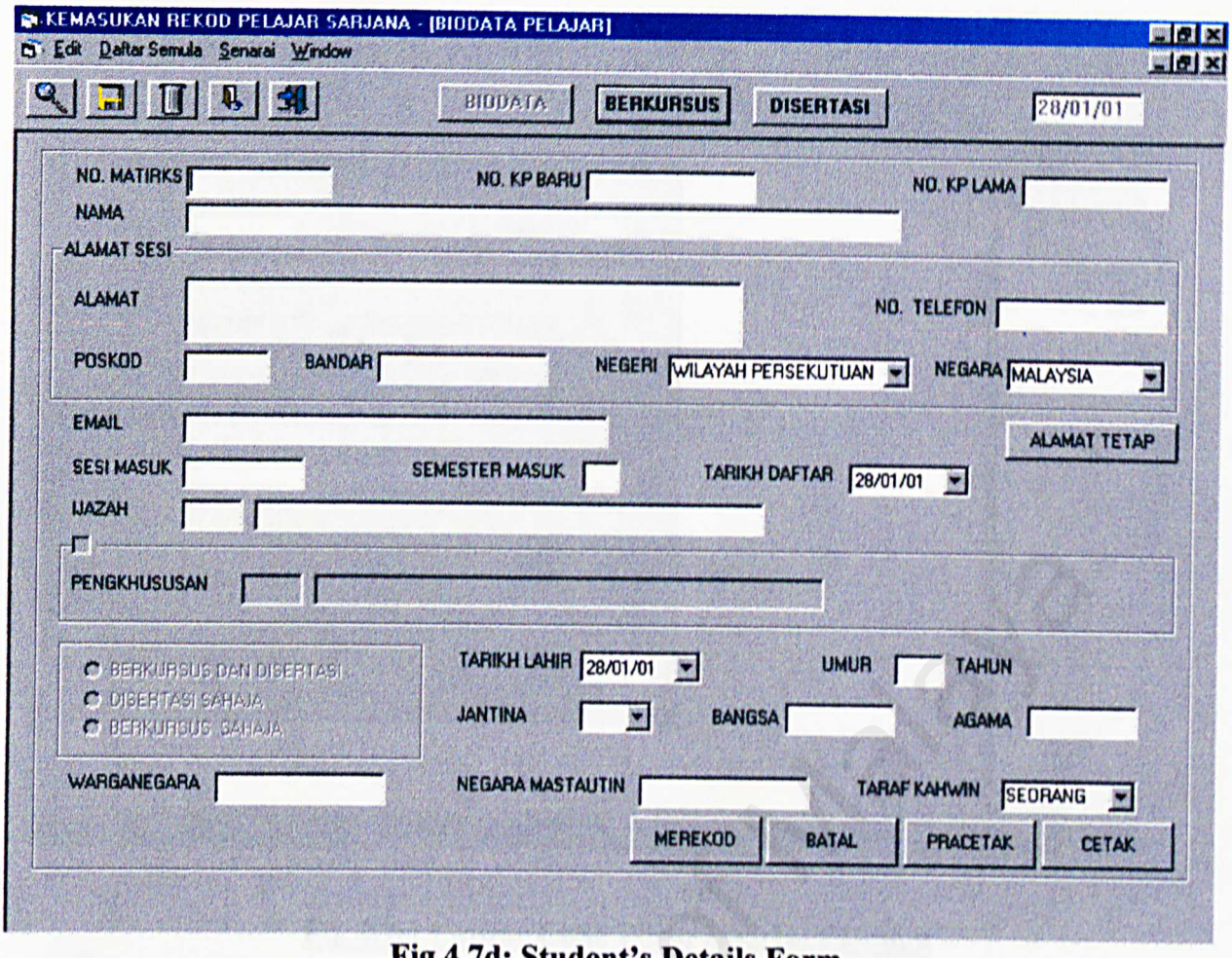


Fig 4.7d: Student's Details Form

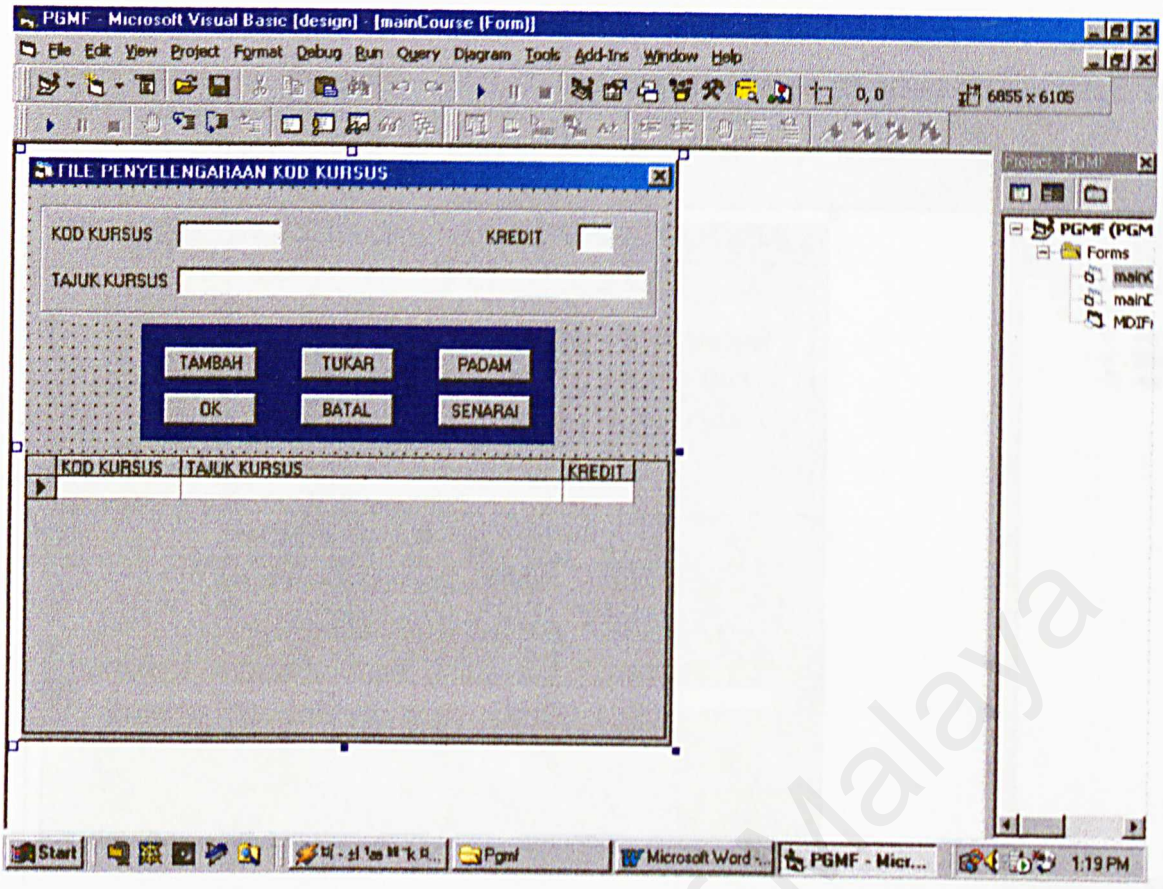


Fig 4.7e: Course Code Form Interface Design

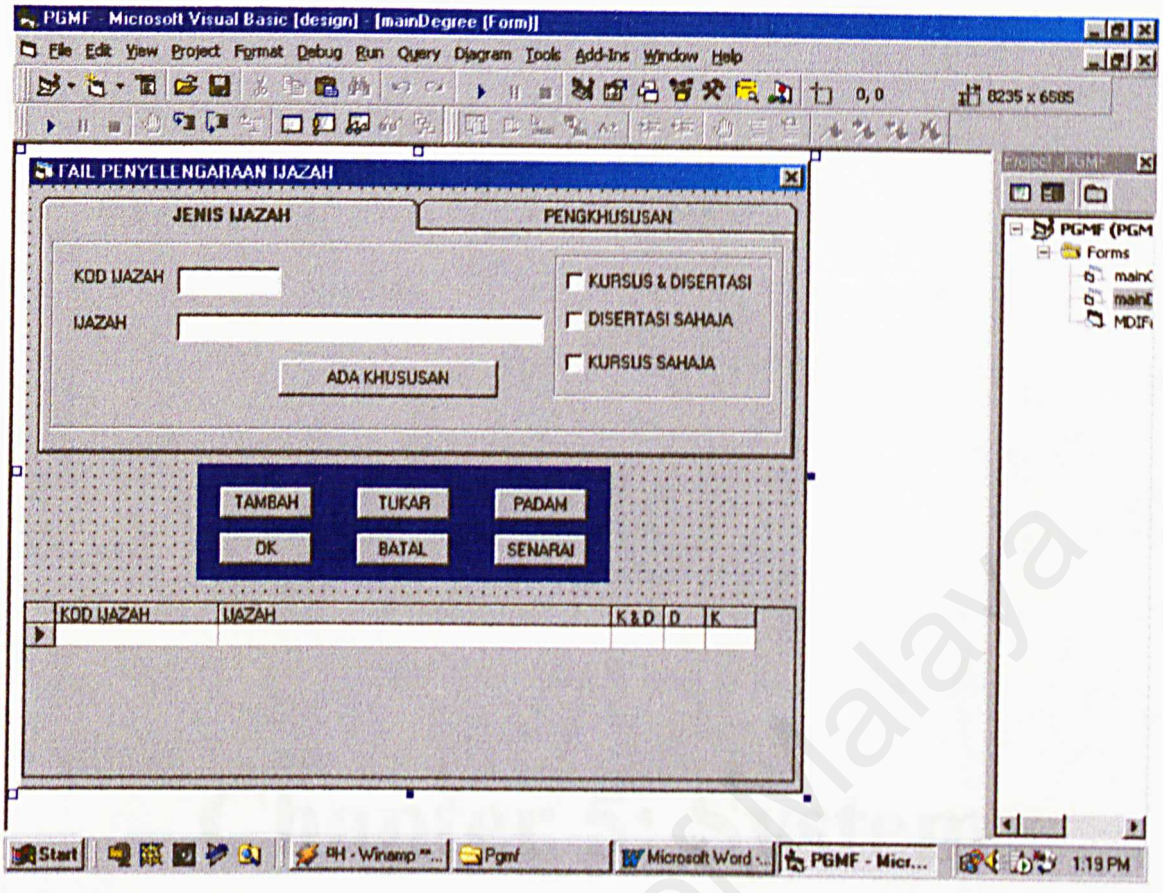


Fig 4.7f: Degree Title Code Form Interface Design

All of these designs took place in Microsoft Visual Basic 6.0 as it can really prototype the system in the interface phase first before starting to code its function. As the MS Visual Basic is a 4th Generation Programming Language, it makes prototyping or screen design very easy and flexible.

Chapter 5: System Implementation

5.1 Introduction

According to the classical SDLC (System Development Life Cycle), system implementation is performed right after designing the system. It consists of coding, testing and document the system as well as training for end user or administrator.

After constructing the structure of the system, it is time to implement it. In another words, it will be put into action by coding it in MS Visual Basic, for the window-based application, and Active Server Page for the web page. After that, it is tested by connecting to the database system for adding and updating dummy data into the database. Then, the three application modules (PGIS, PGIS2 and Maintenance File) are combined into a complete application, the PIS.

5.2 Tools Utilized

There are a few software tools involved during this implementing phase. All of them are basically Microsoft products. This is to avoid any incompatible among the tools utilized. Here are the list of software tools:

Tools Utilized	Usage
Microsoft Office 2000	Analysis & Documentation Purposes
Microsoft InterDev 6.0	Web page Design & Implementation
Microsoft Visual Basic 6.0	Application Design & Implementation
Microsoft SQL Server 7.0	Database Server

Microsoft Window 98	Operating System Platform
Microsoft NT 4.0 Server and Workstation	Testing Platform
Microsoft Window 2000 Server	Testing Platform

Table 5.2a: Software Tools Justification

Apart from this, the system also needed a list of required hardware shown below.

Here is the list:

Hardware Utilized	Usage
Processor Speed > 200MHz	CPU
32 MB RAM	Minimum Memory
1 GB Hardisk Space	Minimum Space
Modem or a LAN connection to Internet	Connection to Internet

Table 5.2b: Hardware Tools Justification

Besides the software tools, the system also requires some resource and reference components.

These are some files that are referred by the application at run-time.

Component Utilized	Description
COMDLG32.OCX	MS Common Dialog Control 6.0
MSMASK32.OCX	MS MaskedEdit Control 6.0
COMCTL32.COX	MS Window Common Control 5.0 (SP2)
MSCOMCT2.OCX	MS Window Common Control-2 6.0
COMCT332.OCX	MS Window Common Control-3 6.0

MSADODC.OCX	MS ADO Data Control 6.0 (OLEDB)
MSDATGRD.OCX	MS DataGrid Control 6.0 (OLEDB)
TABCTL32.OCX	MS Tabbed Dialog Control 6.0 (OLEDB)
PIS32.RES	Resource File

Table 5.2c: System Components

5.3 Special Algorithm

The MainMenu screen has utilized a resource file named 'PIS32.res'. This resource file provided a source of mousepointer icons. The algorithm managing this file is as below.

```
Private Sub Form_Load()

  Set curSelect = LoadResPicture(1, vbResCursor)
  For lintloop = 0 To ImageCmd.Count - 1
    With ImageCmd(lintloop)
      .MousePointer = 99
      .MouseIcon = curSelect
    End With
  Next

  For lintloop = 0 To Label1.Count - 1
    With Label1(lintloop)
      .MousePointer = 99
      .MouseIcon = curSelect
    End With
  Next

End Sub
```

When the mouse hovers over the ImageCmd object and Label object, the mousepointer will become a pointing hand.

5.4 Implementation Phase Problem

There a few minor problems arose during this phase, such as date conversion problem, numerical conversion problem and window sizing problem. The date and numerical problems were caused by their storing format in the SQL Server. As there are long and short format in storing and retrieving dates, it just simply would not fit into an ordinary textbox. So I have proposed the use of the 'Date Time Picker ActiveX Control Version 6.0' that can be obtained in any windows offering the MSCOMCT32.OCX system file. Meanwhile, all the numeric fields are stored in string format, as no calculation is necessary.

The only major problem encountered is the reporting function in the system. This is because Visual Basic 6.0 has its own, low-end reporting system, which is the data-report function and it has to incorporate with a data-environment. While it makes the system developed very independent of other software (Seagate Crystal Report), it also makes it troublesome when connecting to a database dynamically or a recordset generated by the system during run-time. The solution to this problem is to refresh the report and re-query the recordset of the data-environment on every print or preview. Also, temporary tables have to be setup for some of the reports. Then, data from the recordset could be inserted into and referred from the tables.

When implementing the system in other families of Windows, it was found that there is a problem with component files version. This is because some of the platforms have newer version of the components used while the **PIS** is only developed in the Window 98 platform.

Chapter 6: Evaluation

6.1 Introduction

System evaluation is actually the testing phase, the most popular approach to verification and validation of software. Verification and validation (V & V) is the generic term used for checking process which ensures that the software meets its requirements and that the requirement meet the needs of the customer (P.Sellapan, 2000). While Boehm puts V & V in this way:

Verification means "are we building the product right? "

Validation means "are we building the right product? "

6.2 Method of Data Generation

During the whole SDLC, every completing module was tested with some dummy data to ensure the modules are bug-free before it was integrated with one another. Because the postgraduate information is private and confidential, it was not advisable to use real data obtained from the faculty. So, to test the system thoroughly, some data that is nearly similar to the real data (which are actually personal details of students) was made up. Although it is not the real data, it has gone through consultation of the administrator.

6.3 Testing Strategies

Although the Postgraduate Information System is not very large in size, the incremental strategies were used. But it is just to test the application module, while the web pages written in ASP were tested using the 'big-bang' strategies, where once all the web pages were produced it was integrated and tested as a whole. This is because the linkage between pages is very important and can only be successful when every page is already developed. The application on the other hand, was tested from bottom-up where each small but tested module was integrated with one another for a wider system test.

Generally, the testing steps are as such:

1. Code & functional testing – testing of algorithms and codes in a small section.
2. Module testing – testing of a whole application module.
3. Integration testing – testing the integration of all the application modules together.
4. System testing – testing of the overall system as a whole

Code & functional testing

Every source code and function written was tested to make sure the logical flow of the system is correct, its procedure must be operating as required. It is done to keep the smoothness of the system development. Code and functional testing is carried out when every piece of code and function is written. For example, the printing codes has to be tested in this way because data report in Visual Basic is directly linked to the database and it will not preview a new print before the data is refreshed in the temp table. But doing this requires precise codes that manipulates the refresh and requery that binds the recordset again.

Module Testing

When each of the application modules was produced, it was also tested. Modules such as the Listing, Data Entry, Reporting, Master File Maintenance and so on were tested separately to make sure they behave according to requirements. Testing on this stage is to make sure all the modules have a correct logical flow and the SAVE, CLEAR, PREVIEW, PRINT functions are working properly.

Integration Testing

All the tested modules will be integrated. This stage is to confirm that the user interface is correct and system is integrating well. In this stage, the PIS.exe, PGIS.exe, PGIS2.exe and PGMF are integrated together for this purpose.

System Testing

System testing is carried out for testing the overall system to find any major or minor bugs that are produced unexpectedly from the reaction of the components and modules. For this purpose, real data was utilized.

6.4 List of Major Bugs & Remedies

Until this stage, the Postgraduate Information System was already tested fully, both web pages and window-application. As the system is not a very large-scale project, no major bugs were actually found during the testing phase. Although the fact that no discovery of bugs, this does not mean that the system is robust. In software engineering and software quality assurance aspect, it is creating a possibility that the major bugs may still in the loose

and if it does exist, it might cause the system to crash when in real implementation. But of course, the possibility of happening is as low as one or two percent only.

Chapter 7: Conclusion

7.1 Introduction

Lastly, in the final chapter presents how well the objectives of this project were met and also some recommendation on future enhancement is discussed and the lesson learned from the project.

7.2 Meeting Objectives – Its Successful Story

As the project's objectives are divided into two, it is best to discuss it one by one. On the faculty side, it really had given an opportunity for the final year undergraduates to create a system that has commercial values for their final year project. As a result, computerized systems are introduced in the faculty and this eventually has created a paperless environment in the administration of the faculty. But to know how well the systems have really reorganized and enhanced the main administrative functionality in the faculty has yet to be found out.

On the objective of the PIS development however, it has been fully met as the system really does ease the work of the system administrator and provide a better management of the postgraduate information. Now, the administrator can sort the students according preference such as students under MSE, MCS, MLIS or MIT. The system is also able to recognize which student did not re-register currently, something that cannot be done on the previous system.

7.3 System Strength

- **Effective information managing**

  Information that is saved in the database can be manipulated very easily. Especially, there are some master files where important codes can be referred. No manual operation is needed.

- **Effective System Security**

  On every start-up of the application is required to login for students and lecturers/coordinators. Without it, users cannot go into the system directly. User ID and password is kept in a table in the system database and authorization to the database is only for system administrator.

- **Automatic & Independent Reporting Function**

  PIS can produce detail reports where user can refer in the future. Also, the system is independent of other reporting software or component because it uses the data report that is already integrated in Visual Basic 6.0 so the original utilization of Seagate Crystal Report was discarded.

- **A Complete System**

  The system is complete because it consists of two sections, the data entry application and information searching web page. As there will be time where the network or Internet connection is extremely busy especially during peak hours, putting data entry to the web page may actually slows down the work of the administrator. So the data entry application can be installed on the administrator's computer for directly data managing purposes. But when users want to search for information, they can go to the web page, which will only display the required information, depending on the user's privilege.

7.4 System Weakness

- **Lack of attractive interface**

  As the system is developed under a great constrain of time, the user interface lacks of those very attractive interfaces even though it has graphical user interface.

7.5 Special Problem & Solution

When testing phase is begun, it was found that there were no real data available for real testing. This is because the real data is actually postgraduate information and it is private and confidential to others except the administrator. Finally it was replaced with dummy data but quite similar to the real ones, with collaboration from the system administrator.

As the scope of the system is quite broad, it is very difficult to finish every part of the system is full version or having very interactive and attractive, completely in a short period of time. Due to this, there may be bugs that are not discovered yet and also some relevant functions are not included in the system. The solution is to really focus on the more important functions first.

7.6 Future Enhancement Suggestion

Currently, the Postgraduate Information System is dealing with management of the postgraduate information. It allows add and update of the postgraduate information. Data will be stored into a database. But when, a postgraduate has finished his/her degree, all his/her information will be still kept in the database. Although the MS SQL Server can manage a very big database, the PIS database will keep on growing as the years go by as new postgraduates will be coming in. So, for future enhancement, it is recommended that another function is to be added-on in order to transfer the 'graduated' postgraduate's information into a special database or table. With this table, the administrator can choose whether to delete the information permanently or have it stored for future reference.

7.7 Overall Conclusion

The project has taught us final year undergraduates a lot about how a real SDLC is carried out. From the stage of gathering information on system requirements to the final stage of documentation and maintenance, it has directly pointed out many aspects that a computer science student especially a software engineer, has be attentive to. Although the entire project can be considered of small scale, it has given us an insight of what we are going face in the future career. Also, it had given an opportunity for us to develop a single system individually, to test us of our analytical skill and also our management skill. Finally, it gives me great honor to present this final year report and a complete Postgraduate Information System.

Reference & Bibliography

Bibliography

- Alan Davis, "Software Requirements: Objects, Functions & States", Prentice Hall, USA

- B. Dean, Edwin. "Business Process Reengineering: From the Perspective of Competitive Advantage" NASA, USA December 6, 1996. Available at http://mijuno.larc.nasa.gov/dfc/bpre.html

- Bee, J. (1995) The University of Akron Computing and Communication Technologies Committee Five Year Plan. Available at http://www.uakron.edu/cctc/cctc.bee.html

- Chris Ullman, David Buser, Jon Duckett, Brain Francis, John Kauffman, Juan T. Llibre, David Sussman, "Beginning ASP 3.0", Wrox Press Ltd., UK, 1999

- Clayton AMH and Radcliffe, NJ (1996) "Sustainability - a systems approach" Earthscan: ISBN 1 85383 3193. Available at http://www.srds.co.uk

- Hammer, M. and J. Champy (1993). Reengineering the Corporation: A Manifesto for Business Revolution, HarperBusiness, New York NY.

- Hammer, M. (1990, July-August). "Reengineering Work: Don't Automate, Obliterate," Harvard Business Review, pp. 104-112.

- Ian Sommerville, "Software Engineering", Addison Wesley, 5th Edition, 1998

- Kenneth E. Kendall, Julie E. Kendall, "System Analysis and Design", Prentice Hall, 4th Edition, 1999

- Lave Singh, Kelly Leigh, Joe Zafian, et. al., "Oracle 7.3: Developer's Guide", SAM Publishing, 5th Edition, 2000

- Malhotra, Yogesh. "Business Process Redesign: An Overview," IEEE Engineering Management Review, vol. 26, no. 3, Fall 1998. Available at http://www.brint.com/papers/bpr.htm

- M. Grotevant, Susan. "Business Engineering and Process Redesign in Higher Education: Art or Science?" CAUSE 98 Seattle, Washington December 8, 1998. Available at http://www.educause.edu/ir/library/html/cnc9857/cnc9857.html

- Paul Thurrott, Gary Brent, Richard Bagdazian, Steve Tendon, "Delphi 3 Superbible", Waite Group Press, 1999

- Piper, James. "Business Process Reengineering: To rethink, redesign processes, you'll need to know what BPR is – and isn't" Trade Press Publication Corporation 1995 Available at http://www.facilitiesnet.com/NS/NS3bm5a.html

Reference

- Davenport, T.H. & Short, J.E. (1990 Summer). "The New Industrial Engineering: Information Technology and Business Process Redesign," Sloan Management Review, pp. 11-27.

- From CLANN to UNILINC: An Automated Library Consortium from a Soft Systems Perspective (Australia, Peter Checkland). Thesis [Ph.D.]; 1996. University of Toronto (Canada); 0779. ISBN: 0-612-11862-2. DAI, Vol. 57-08A, Page 3309, 00308 Pages. Available at http://valinor.purdy.wayne.edu/ssm3.html

- Kamsah, Mahfuzah. , Mokhtar, Salimah. , Ahmad, Rodina., "Toward a model of *e-University*: A Case of Malaysian Higher Learning Institutions" Faculty of Computer Science and Information Technology, University of Malaya, Malaysia, 2000

- P.Sellappan "Software Engineering, Management & Methods" Sejana Publishing,ISBN 983-2017-18-1, Petaling Jaya, Malaysia, 2000

- Snover, Jane. "Overview OF The Reengineering Project in MIT" Massachusetts Institute of Technology (MIT), US, June 1999. Available at http://web.mit.edu/afs/athena.mit.edu/org/r/reeng/www/

- Tait, F. (1999), Enterprise Process Engineering: A Template Tailored for Higher Education. *CAUSE/EFFECT*, Vol 22, No.1.

- Trowler, P. (1998), "Academics responding to change: New higher education frameworks and academic cultures." Buckingham, SRHE and Open University Press.

Postgraduate Information System
User Manual

User Manual

System Requirement

Before installing and using the PIS software, please make sure that the computer system has the following.

Hardware:

- Processor Speed > 200 MHz
- 32 MB RAM
- 3.2 GB Hardisk Space
- A secure network connection to the internet

Software:

- Window 95, 98, 2000, NT 4.0, ME
- ODBC
- Internet Explorer or Netscape Communicator browser

Installing the software

- Before using the software, it has to be installed first, of course.
- Put the installation CD into the CD-ROM
- Look in the directory ' X:\PIS\ ' ; X will be the CD-ROM drive
- Double-click on the Setup.exe file and just follow the setup instructions.
- It is recommended to restart the system after finished setting up.
- All the installed files including *.exe files will be installed into ' C:\PIS ' directory.

Running the software

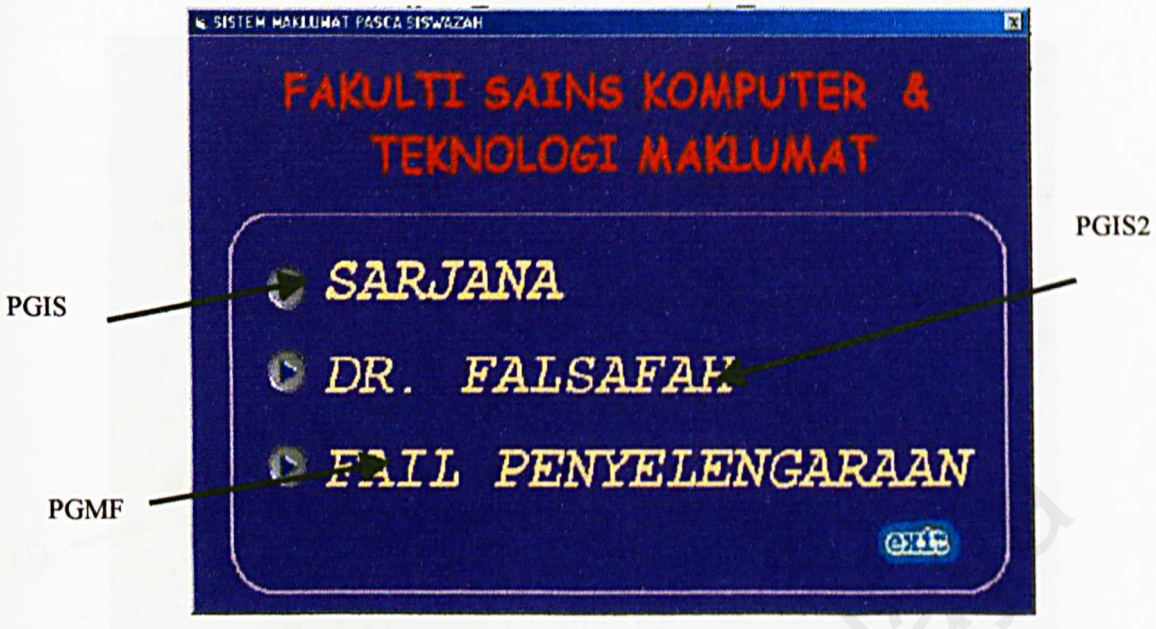
- Look for PIS icon in the program group under **Start\Programs\PIS**
- Click on the PIS icon and the program will start.
- Make sure the local area network is working and linked to the PIS database.
- Make sure the screen mode is in 800*600 pixels

Using the software

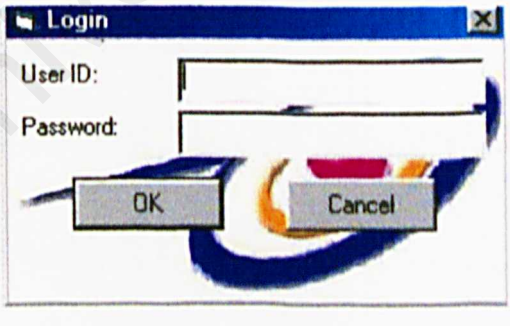
List of important icons:

-  Search or look
-  Save the data into database
-  Clear all the data in the textbox
-  Close the active form
-  Exit the application

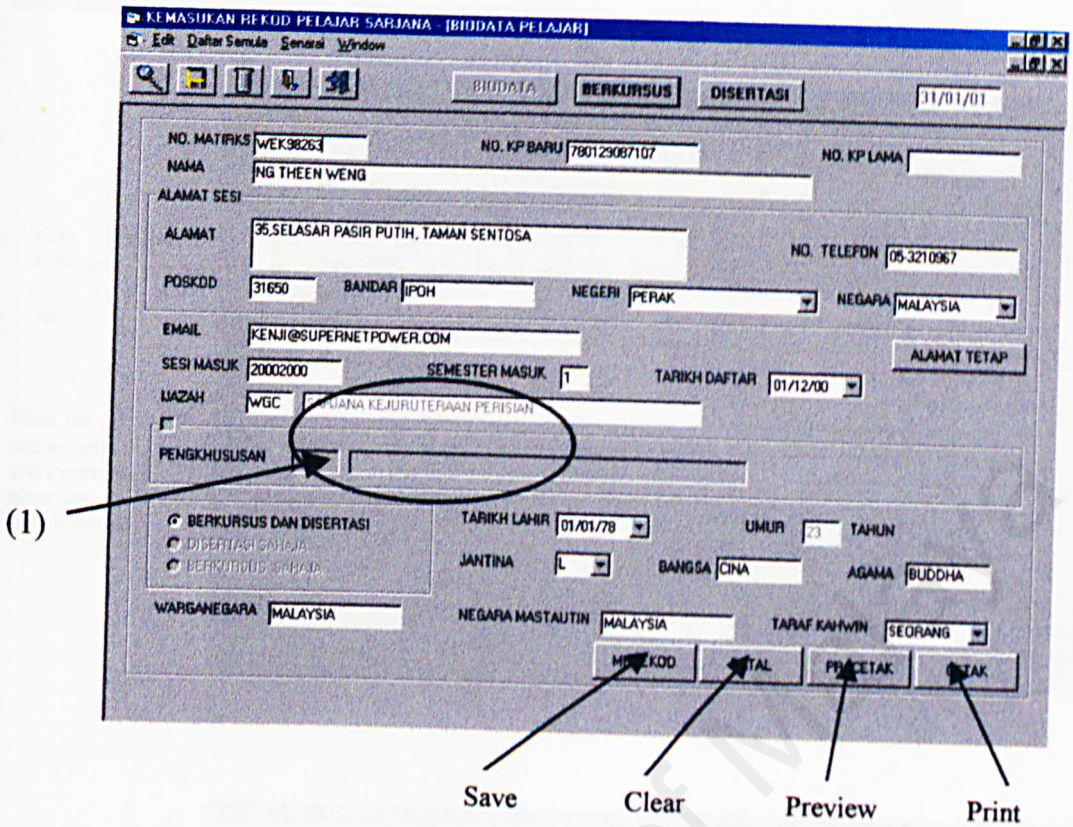
Main Menu



- In this Main Menu screen, user can have the options of going into the Master Student application module, Ph.D. Students application module or Maintenance File application module.
- Click on any of the options, user will be passed on to the application module and a password textbox appears for login into any of the system module.

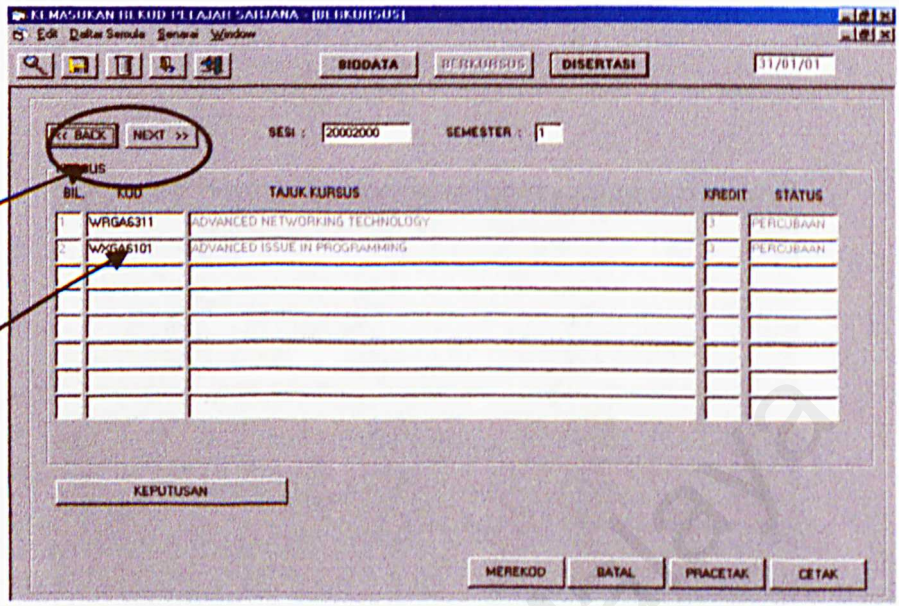


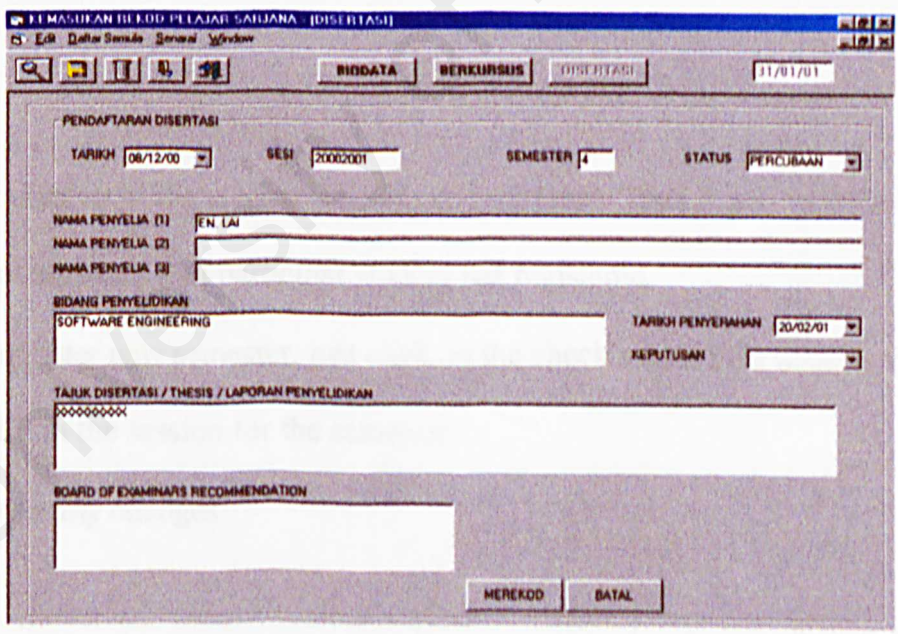
PGIS



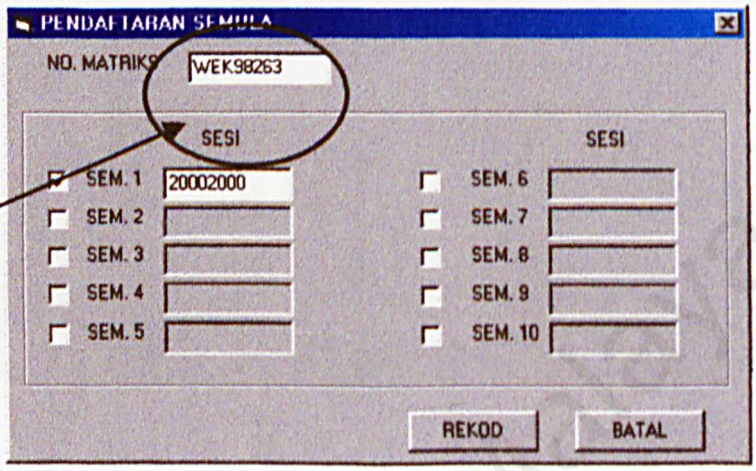
- This form is for entering master student's details.
- Some of the fields can be displayed automatically such as the Degree Title Description and the Majoring Description, when its code is entered. (1)

- The next form is for entering the courses registered for a particular session and semester and also the dissertation title registered.





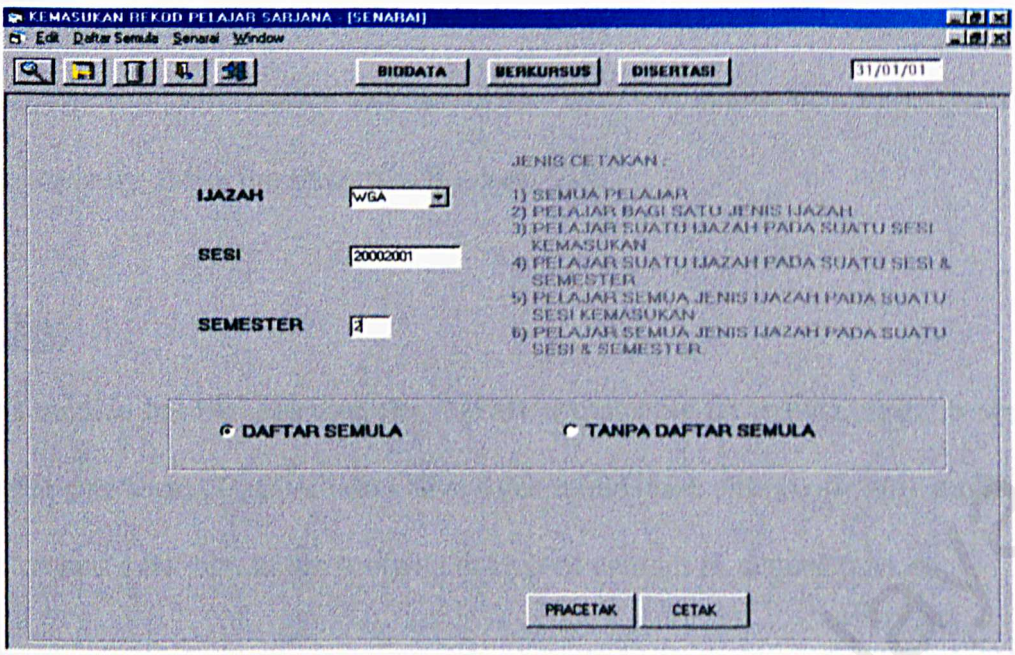
- Then when the postgraduate re-register for a particular session and semester, click on the "Daftar Semula" menu on the toolbar.



Enter Matrix Number

- Enters a matrix number of the student who has re-registered. Then it will show how many session and what semester that particular student has registered.
- If user want to register new semester, just click on the checkbox and the textbox will be enabled. Then fill in the session for the semester.
- Remember to save any changes.

- The listing function form



This form actually can perform a series of listing such as

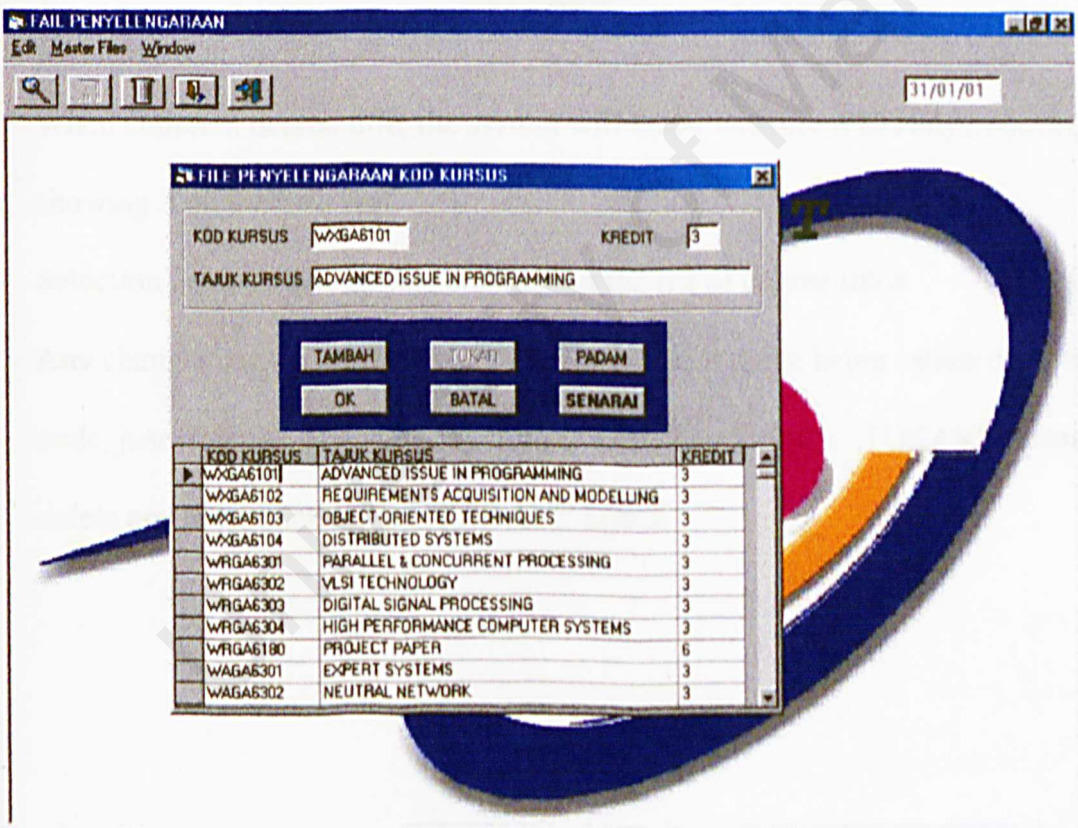
- list all postgraduate (choose ALL in the IJAZAH box, others blank)
- list postgraduate according to MCS, MSE, MLIS or MIT (choose any of the code in the IJAZAH box, others blank)
- list all postgraduate who registered for a particular session (choose ALL for IJAZAH, the session wanted and "1" for the semester box)
- list postgraduate according to MCS, MSE, MLIS or MIT of a particular session (choose any of the code in the IJAZAH box, the session wanted and "1" for the semester box)
- list all postgraduate who did or did not registered for a particular session and semester (choose ALL for IJAZAH, the session and semester wanted and whether did or did not register)
- list postgraduate according to MCS, MSE, MLIS or MIT of a particular session and semester, whether did or did not register (choose any of the code in the IJAZAH box, the session and semester wanted and whether did or did not register)

PGIS2

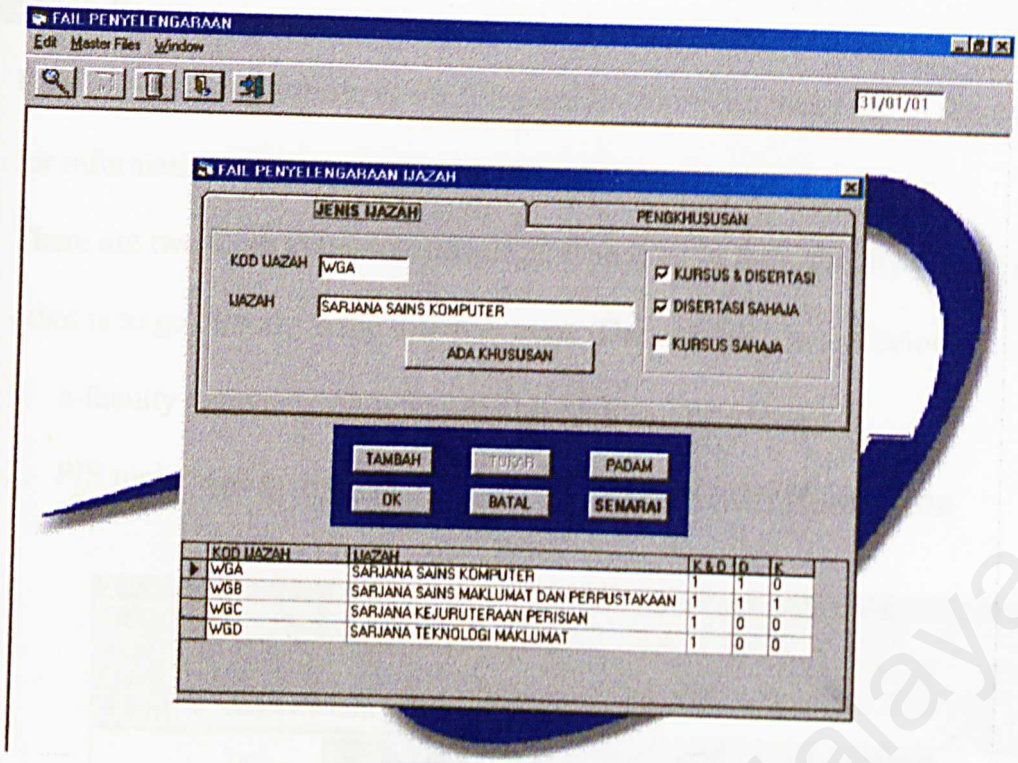
The PGIS2 application module is only a smaller version of PGIS because it is for managing Ph.D. students' information and they do not have any degree title, majoring or courses taken. But basically, it has the same functions as PGIS.

PGMF

This module has two sub modules that are responsible for maintaining the course codes and degree title codes. Both of them have three mode (add, change, delete), an OK button, cancel button and a list button for showing the list of courses or degree titles.



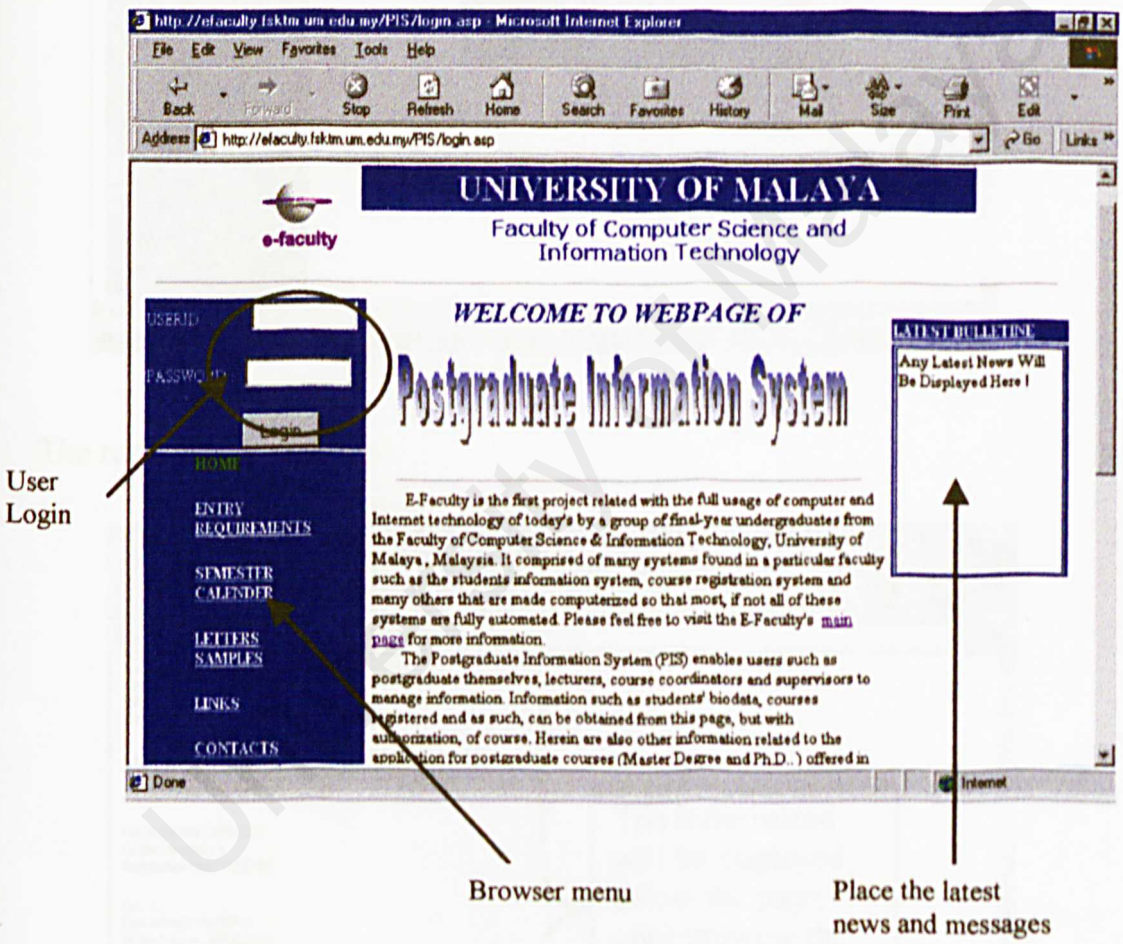
- Selection can be made by just clicking on the list of degree titles



- When chosen a degree title, the system will know whether it has any majoring and showing it on the next leaf.
- Selection can be made by just clicking on the list of degree titles
- Any changes can be made directly with the correct mode being selected. If want to add a code, just click the "TAMBAH" button, if update, click the "TUKAR" button and if delete any data; click on the "PADAM" button.

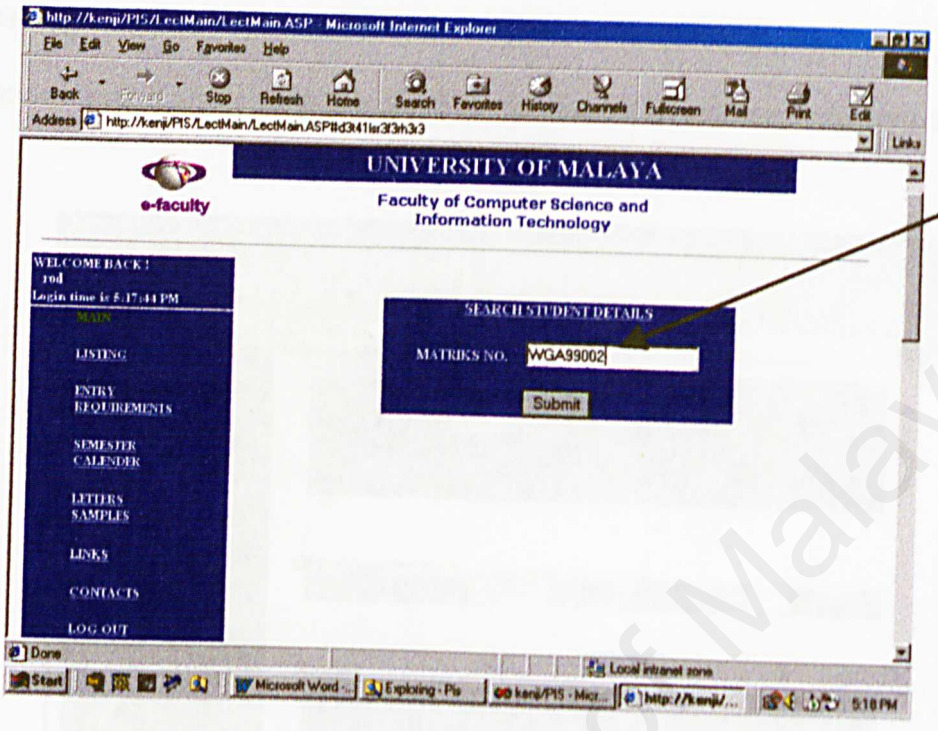
The Web Page

- User such as postgraduate or coordinators/lecturers can make use of this page to search for information.
- There are two ways to browse this page. One is go to the e-faculty project main page, the other is to go directly to the PIS web page. The two URL are as below:
  1. e-faculty main page @ www.efaculty.fsktm.um.edu.my
  2. PIS main page @ www.efaculty.fsktm.um.edu.my/PIS/login.asp

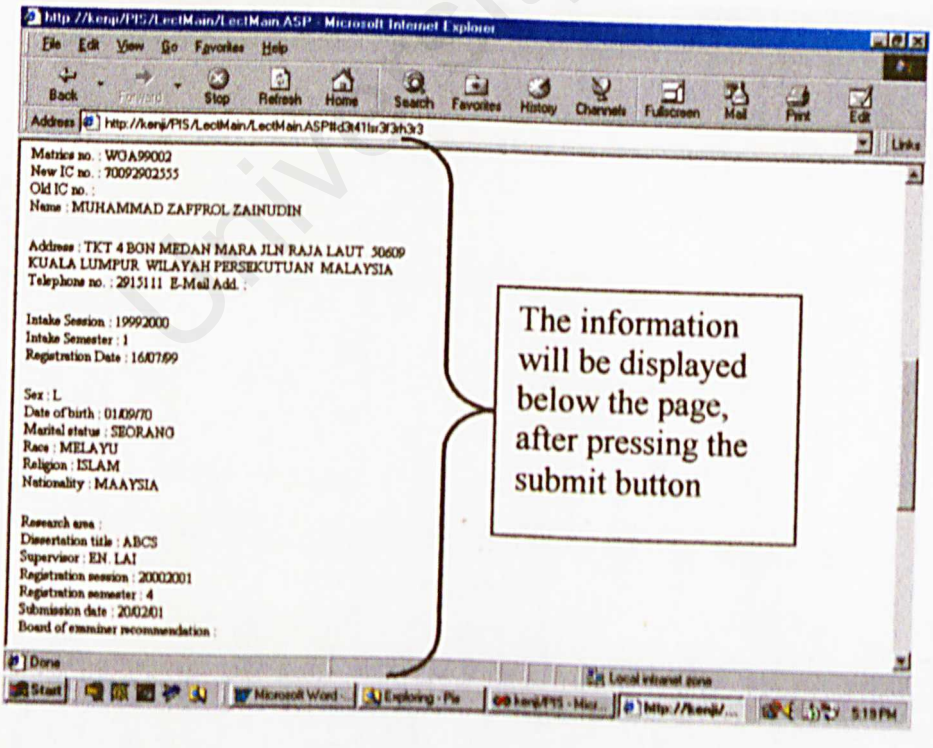


- This is the main page of the Postgraduate Information System web page. It is also the page for logging into the system.
- The Latest Bulletin column is for the administrator to put up any news or messages that are relevant to the postgraduates.

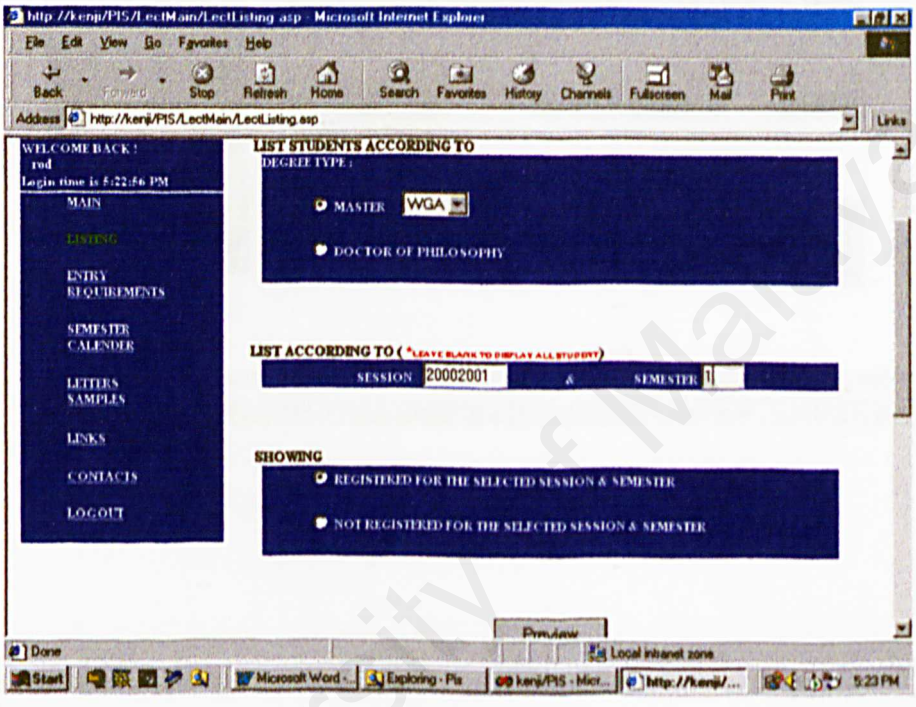
- If logging in as a lecturer, the system will be brought to the lecturer/coordinator page. This section brings two functions. The first function is for searching a particular student's details such as student's biodata, courses and dissertation registered.



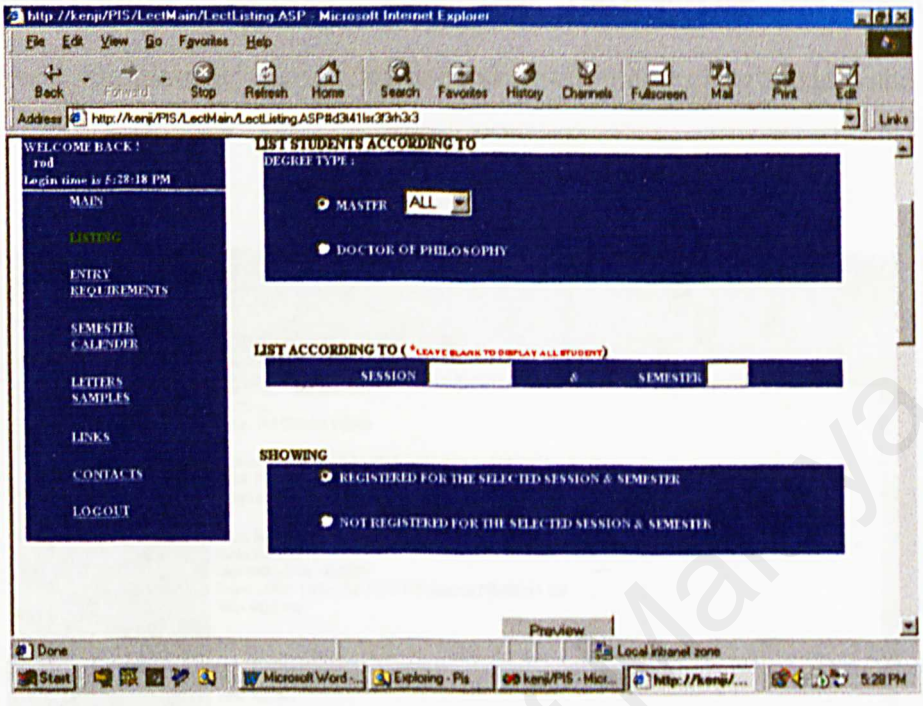
- The result of the function:



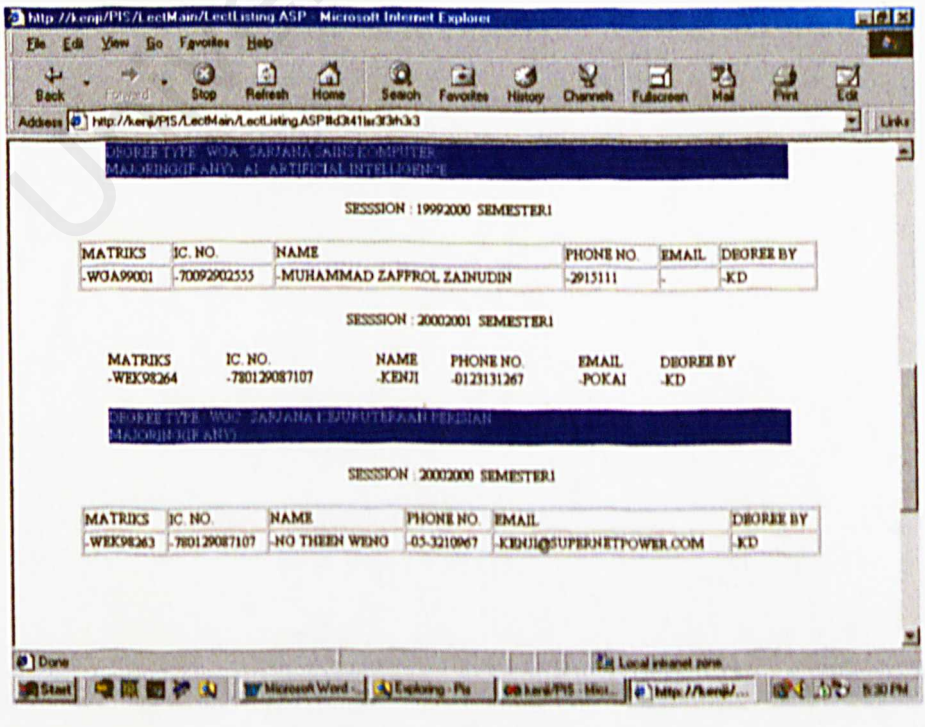
- The second function is the listing function that can be found in the Listing Page from the browser menu. It actually has the same function as the listing function in the administrator application software.
- Just enter the required criteria such as session and semester, Master or Ph.D., press preview and it will search the database for the list of students who matches the criteria.



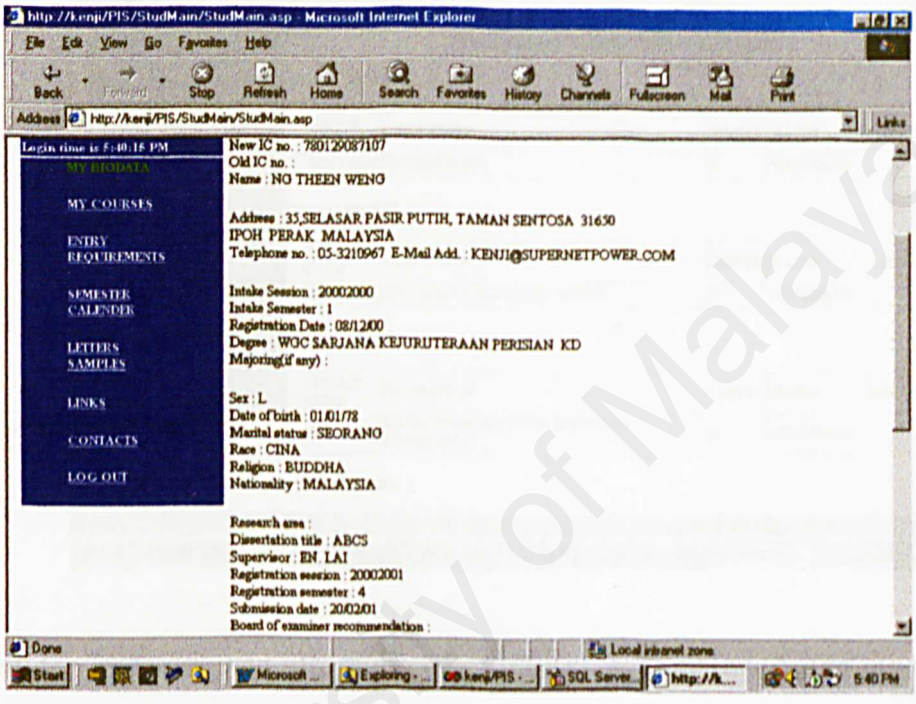
- Besides having to enter the criteria, user can also leave all the textbox blank to list all the Master or Ph.D. students, as shown below:



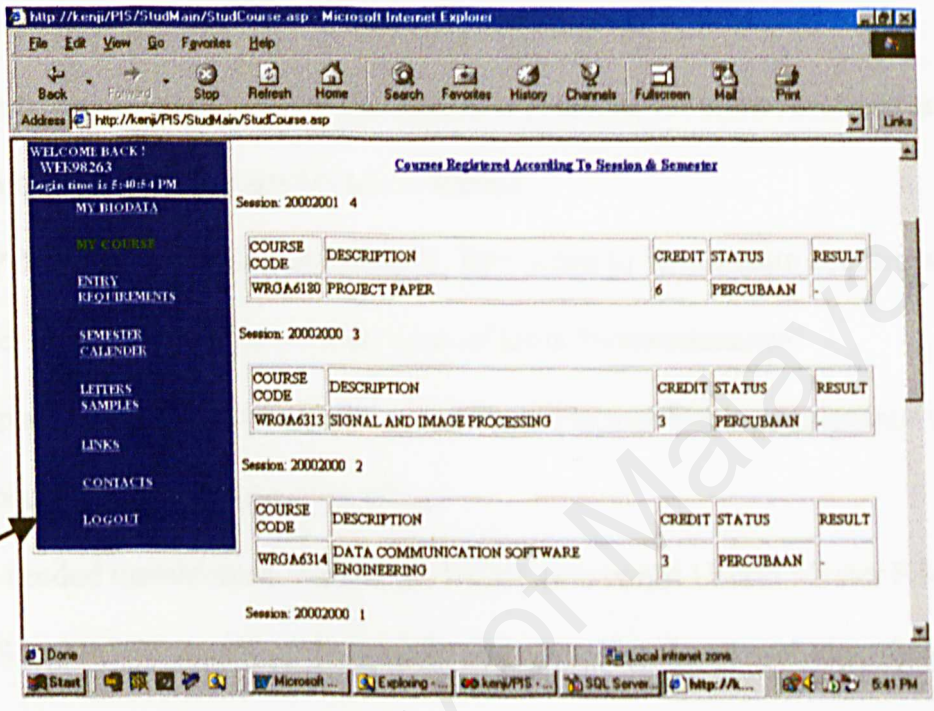
- The result is as shown below



- On the other hand, if user login as a student the system will be brought to the student page.
- On this page consists of two main function page. The first is the "My Biodata" page where it will automatically list all his/her information including biodata and dissertation title.



- The second page which is known as "My Course".
- This page is also automated in displaying the courses that have been taken, in reverse order of session and semester (from current to previous)



Logout (1)

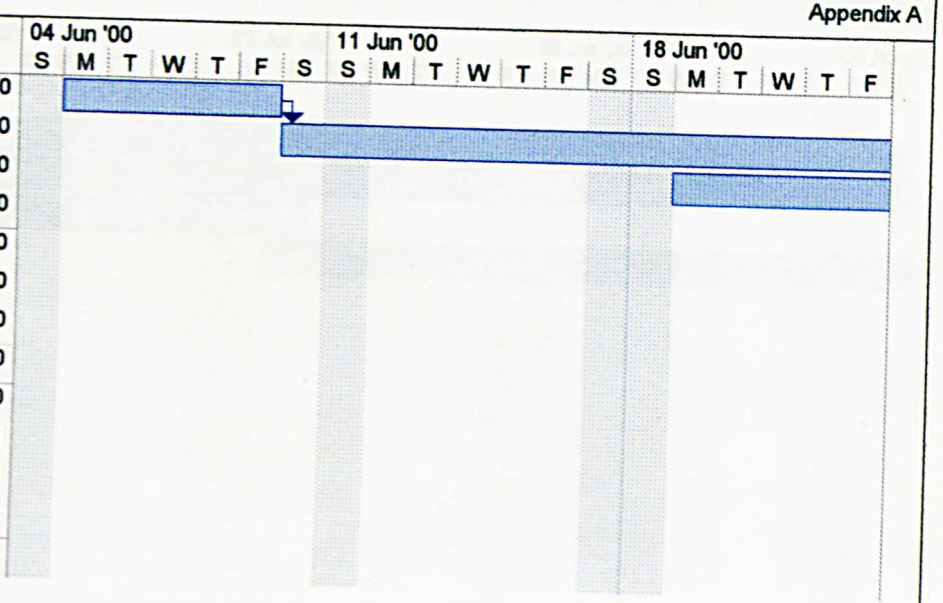
- After all the browsing and information searching, users are needed to logout of the system clicking on the logout menu. (1)

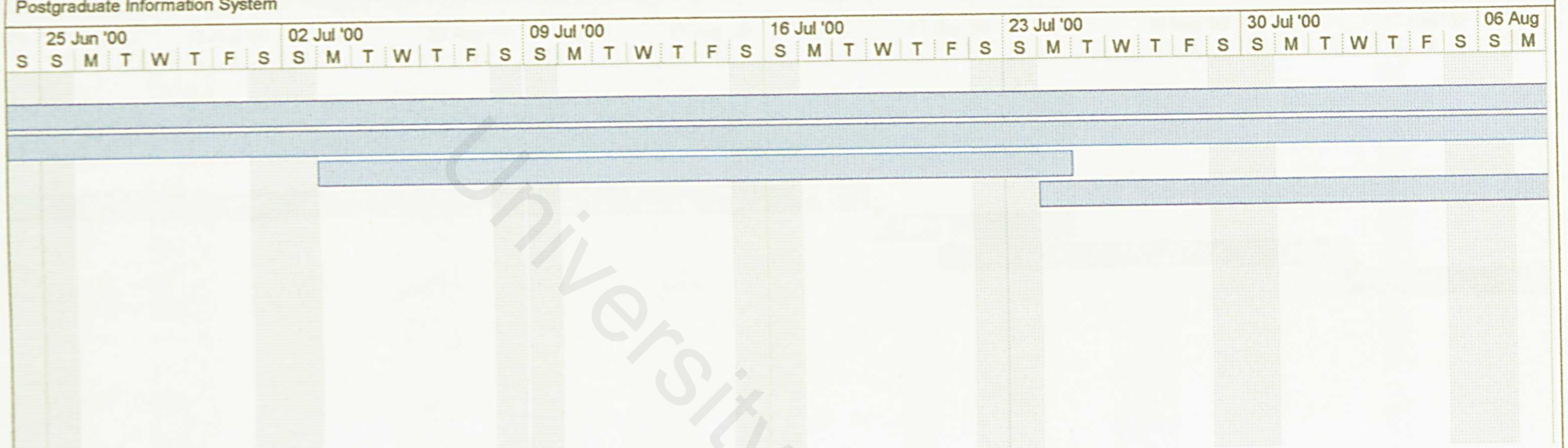
Maintenance

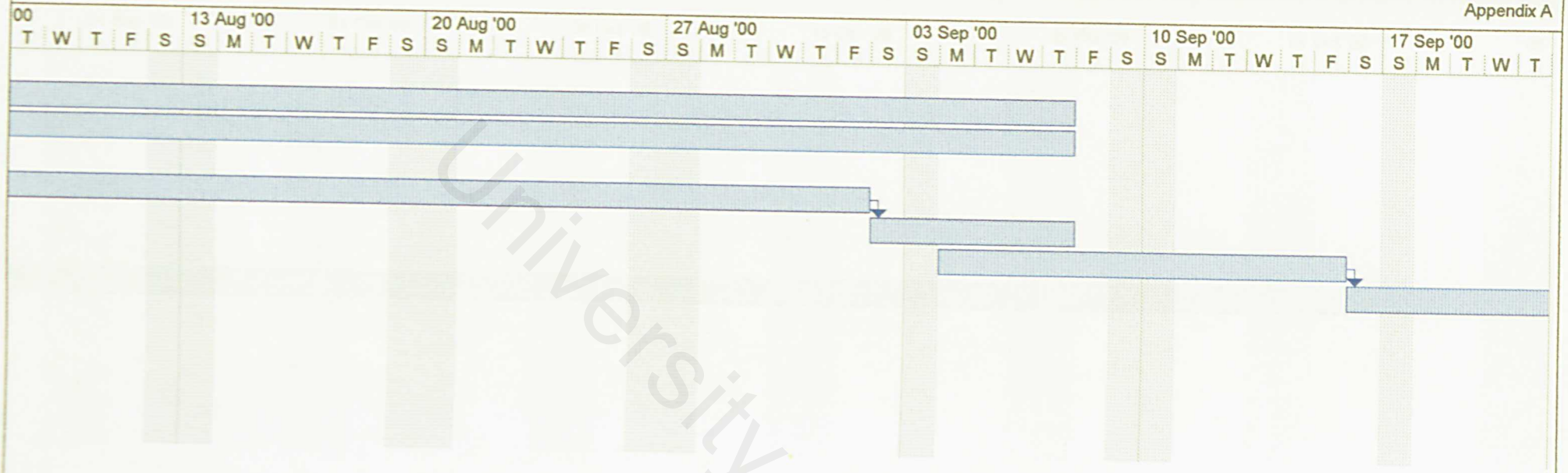
- Maintenance is quite easy for the Postgraduate Information System. The software itself can be divided into two - the web page and the application.
- The web pages' latest bulletin board (on the right side of the main page) just needs to be updated every now and then.
- The application only needs an upgrade version if possible, for extra functions and enhancements. No further maintenance is needed.
- Only the **PIS** database running on MS SQL Server has to be maintain occasionally. The SQL Server software itself has already a set of tools for maintenance.
- Do backup all databases and tables periodically and before every maintenance to ensure any unexpected inconvenience or problem.
- The most needed maintenance table is the PUMF (Password Userid Master File) table in both PGIS and PGIS2 database. It contains all the password and user identification of the administrator to login to the application.
- Another table is the WPUMF (Web Password Userid Master File) table in the WEBBING database. It contains all the passwords and User Identifications of all the postgraduates who could login to the **PIS** web page.
- WPUMF is even serious as the number of Postgraduates ascends every year because of the intake and increment of postgraduate quota in University of Malaya.
- Postgraduate who has left or graduated, must be deleted its Userid and Password from these tables to avoid them getting bigger and bigger.

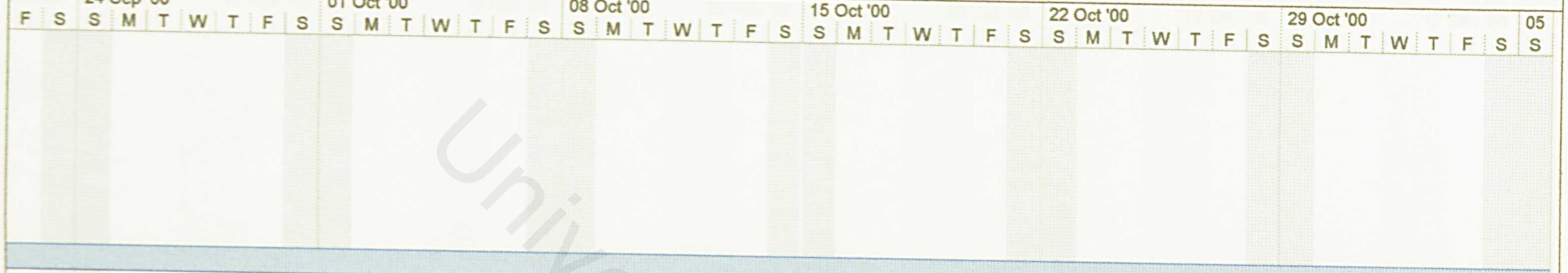
Appendix A

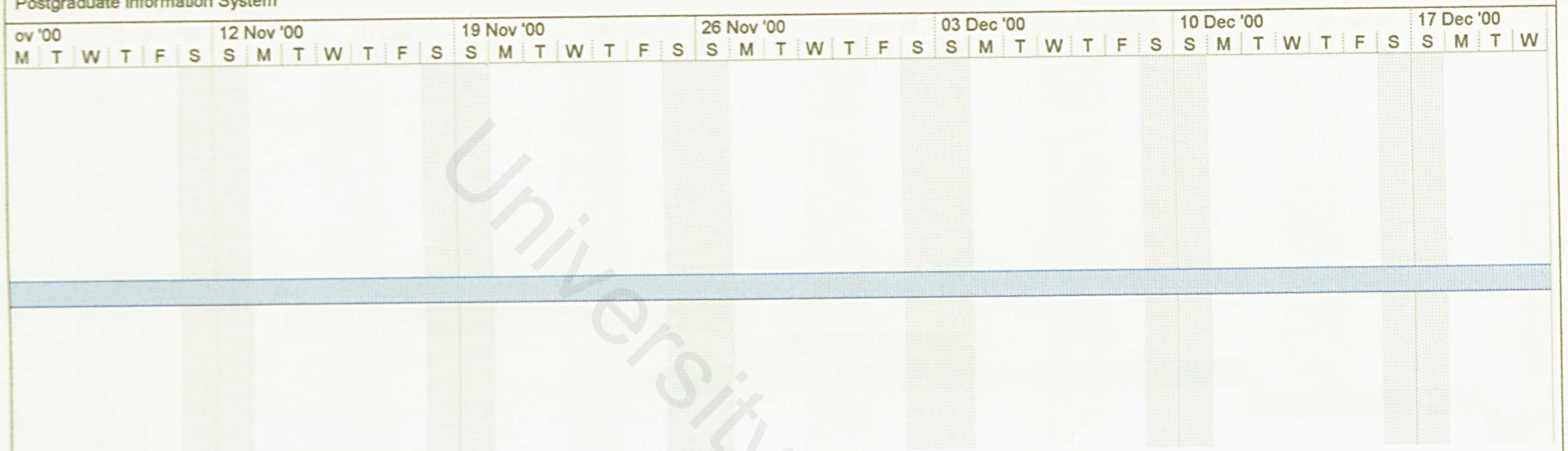
ID	Task Name	Duration	Start
1	Thesis title hunting	5 days	Mon 05/06/00
2	Literature review and survey	66 days	Sat 10/06/00
3	Analyse current system	60 days	Mon 19/06/00
4	Interview stakeholders	16 days	Mon 03/07/00
5	Understand proposed system and requirements	30 days	Mon 24/07/00
6	State all requirements plus new useful function	5 days	Sat 02/09/00
7	Prepare final proposal draft	10 days	Mon 04/09/00
8	System design	75 days	Sat 16/09/00
9	Implementation test on test data	9 days	Fri 29/12/00
10	Prototyping	5 days	Thu 11/01/01
11	Prepare final report and completed system	14 days	Mon 08/01/01
12	Prepare instruction manuals/ user manuals	5 days	Fri 26/01/01
13	Submission	1 day	Fri 02/02/01

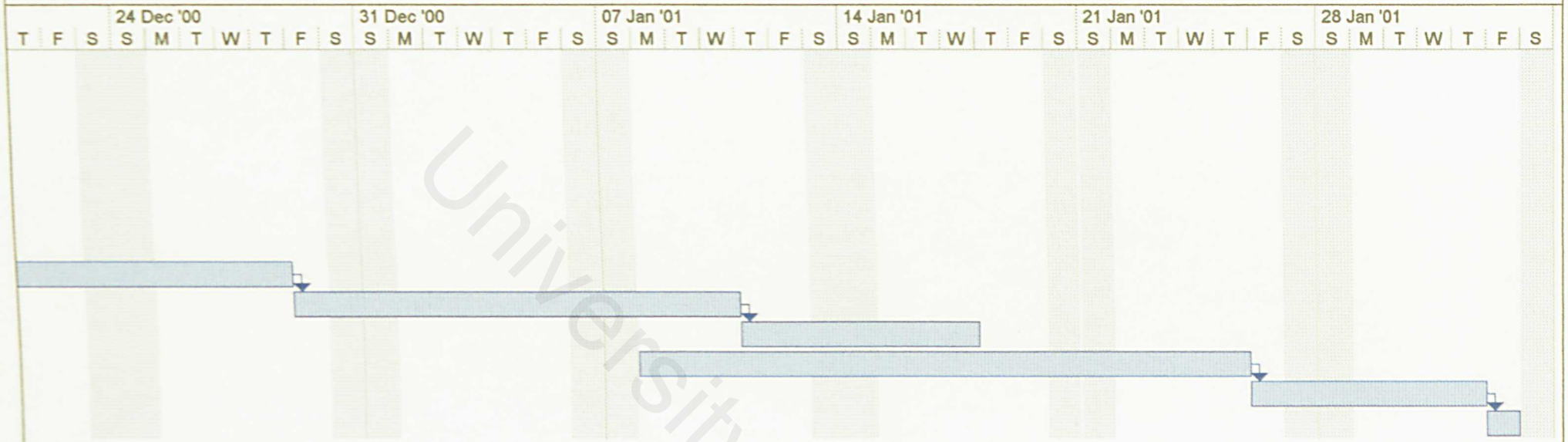












Appendix B

Example of Program Codes

Codes for running the application module:

```
Private Sub RunProgram(Index As Integer)

  Dim Path As String

  If Index = 0 Then
    Path = Dir("C:\PIS\PGIS.EXE")
    If Path <> "" Then
      Shell "C:\PIS\PGIS.EXE", vbNormalFocus
    Else
      MsgBox "Application not found ! Please contact the supplier."
    End If
  ElseIf Index = 1 Then
    Path = Dir("C:\PIS\PGIS2.EXE")
    If Path <> "" Then
      Shell "C:\PIS\PGIS2.EXE", vbNormalFocus
    Else
      MsgBox "Application not found ! Please contact the supplier."
    End If
  ElseIf Index = 2 Then
    Path = Dir("C:\PIS\PGMF.EXE")
    If Path <> "" Then
      Shell "C:\PIS\PGMF.EXE", vbNormalFocus
    Else
      MsgBox "Application not found ! Please contact the supplier."
    End If
  Else
    Unload Me
  End If

End Sub
```

Codes for Checking the user name and password in the Login.frm

```
Dim varCon As ADODB.Connection
Dim varCom As ADODB.Command
Dim varRst As ADODB.Recordset

Public LoginSucceeded As Boolean
-----------------------------------------
Private Sub cmdCancel_Click()
    'set the global var to false
    'to denote a failed login
    LoginSucceeded = False
    'Me.Hide
    Unload Me
End Sub
-----------------------------------------
Private Sub cmdOK_Click()
    'check for correct password
   If txtUserName <> "" Then
     If Verification Then
       MDIFrm.Show vbModeless
       Unload Me
     End If
   Else
     MsgBox "Invalid User, try again!", , "Login"
     txtUserName.SetFocus
     SendKeys "{Home}+{End}"
   End If
End Sub
-----------------------------------------
Private Function Verification() As Boolean

  Dim lstrcnn As String

  Screen.MousePointer = vbHourglass

  lstrcnn = "Provider=sqloledb;Data Source=kenji;Initial Catalog=PGIS2;User
Id=sa;Password=; "
  Set varCon = New ADODB.Connection
  Set varRst = New ADODB.Recordset

  On Error GoTo ErrorDB

  varCon.Open lstrcnn
  varRst.Open "PUMF", varCon, adOpenStatic, adLockReadOnly, adCmdTable
  On Error GoTo 0

  With varRst
    If .RecordCount <> 0 Then
```

```
        .MoveFirst
        While Not .EOF
          If txtUserName.Text = varRst!userid Then
            If txtPassword.Text = varRst!Password Then
              Verification = True
              GoTo Exiting
            Else
              MsgBox "Incorrect password !"
              txtPassword.SetFocus
              Verification = False
              GoTo Exiting
            End If
          End If
          .MoveNext
        Wend
        MsgBox "Cannot find User's ID."
      End If
    End With

    varRst.Close
    varCon.Close
    Set varRst = Nothing
    Set varCon = Nothing
    Screen.MousePointer = vbNormal
    Exit Function

Exiting:
    varRst.Close
    varCon.Close
    Set varRst = Nothing
    Set varCon = Nothing
    Screen.MousePointer = vbNormal
    Exit Function


ErrorDB:
    MsgBox "Error in in retrieveing data.Check the database system"
    Screen.MousePointer = vbNormal
    Verification = False

End Function
```

Codes for Resizing The Child Form in the MDIForm

```
Public Sub gfResizingChild(ByRef MDIObject As Form, ByRef ChildObject As Form)

  With ChildObject
    .fraMain.Height = .Height * 93 / 100
    .fraMain.Width = .Width * 48 / 50
    .fraMain.Top = .Height * 1 / 100
    .fraMain.Left = (.Width - .fraMain.Width) / 2
  End With

End Sub
-----------------------------------------
Public Sub gfMoveCmdBtn(FormObject As Form, Index1 As Integer, Index2 As Integer,
Index3 As Integer, Index4 As Integer)

  With FormObject
    .cmdCtr(Index1).Move 5920, (.fraMain.Top + .fraMain.Height - 650), 1300, 460
    .cmdCtr(Index2).Move 7280, (.fraMain.Top + .fraMain.Height - 650), 1300, 460
    .cmdCtr(Index3).Move 8640, (.fraMain.Top + .fraMain.Height - 650), 1300, 460
    .cmdCtr(Index4).Move 10000, (.fraMain.Top + .fraMain.Height - 650), 1300, 460
  End With

End Sub
-----------------------------------------
Public Sub gfMoveCmdBtnPrn(FormObject As Form, Index1 As Integer, Index2 As Integer)

  With FormObject
    .cmdBtn(Index1).Move 5920, (.fraMain.Top + .fraMain.Height - 750), 1300, 460
    .cmdBtn(Index2).Move 7280, (.fraMain.Top + .fraMain.Height - 750), 1300, 460
  End With

End Sub
```

Codes in Main1.frm That Controls The Command Buttons

```
Public Sub cmdCtr_Click(Index As Integer)
  If Index = coBtn_AlamatTetap Then
    main4.Show

  ElseIf Index = coBtn_Merekod Then
    gfProgOption
    If gfSearchNoMatriks Then     'make sure a valid no. matriks in database
      If MsgBox("Sistem telah mendapati adanya data bagi no. matrik ini." & Chr(13) &
"Teruskan pengemaskinian data?", vbExclamation + vbOKCancel) = vbOK Then
        If ValidAllFields Then
          If UpdateRecords Then
            lstrGeneral = txtKey(coFld_NoMatriks).Text
            MsgBox "Rekod telah dikemaskini dalam pangkalan data", vbInformation
          Else
            MsgBox "Pengemaskinian rekod dibatalkan", vbExclamation
          End If
        End If
      Else
      End If
    Else
      ' if there's changes in the no matriks field then just update the data
      If lstrGeneral <> "" Then
        If ValidAllFields Then
          If MsgBox("Sila pastikan tiada kursus yang didaftarkan di bawah no. matriks yang
hendak diubahkan kerana semua data berkenaan akan dihapuskan." & Chr(13) & "Teruskan
pengemaskinian ?", vbCritical + vbOKCancel) = vbOK Then
            If UpdateRecords Then
              MsgBox "Rekod telah dikemaskini dalam pangkalan data", vbInformation
            Else
              MsgBox "Pengemaskinian rekod dibatalkan", vbExclamation
            End If
          End If
        End If
      Else
        ' if new matriks no, data will be inserted as new batch of information
        If ValidAllFields Then
          If MsgBox("Simpan rekod ?", vbInformation + vbOKCancel) = vbOK Then
            If AddRecords Then
              lstrGeneral = txtKey(coFld_NoMatriks).Text
              MsgBox "Rekod telah disimpan dalam pangkalan data", vbInformation
            Else
              MsgBox "Penyimpanan rekod dibatalkan", vbExclamation
            End If
          End If
        End If
      End If
```

```
    End If

  ElseIf Index = coBtn_Batal Then
    If MsgBox("Semua data akan dikosongkan.Teruskan ?", vbOKCancel + vbExclamation)
= vbOK Then
      ClearAll
    Else
    End If
    txtKey(coFld_NoMatriks).SetFocus

  ElseIf Index = coBtn_Pracetak Then
    If ValidAllFields Then PrintReport coPrn_ToScreen

  ElseIf Index = coBtn_Cetak Then
    If ValidAllFields Then PrintReport coPrn_ToPrinter

  End If
End Sub
```

Codes for Turning All Character Keyed into Uppercase

```
Private Sub txtKey_KeyPress(Index As Integer, KeyAscii As Integer)
  KeyAscii = Asc(UCase(Chr(KeyAscii)))
End Sub
```

The Basic Codes for Connecting and Manipulating The Database

```
Dim lstrcnn As String

Screen.MousePointer = vbHourglass
lstrcnn = "Provider=sqloledb;Data Source=kenji;Initial Catalog=PGIS;User
Id=sa;Password=; "
Set varCon = New ADODB.Connection
Set varRst = New ADODB.Recordset

On Error GoTo ErrorDB

varCon.Open lstrcnn
varRst.Open "MSIF", varCon, adOpenKeyset, adLockOptimistic, adCmdTable
On Error GoTo 0

varCon.BeginTrans
varRst.AddNew
' Here is the codes for manipulating the PIS databases in SQL Server
varRst.Update
varCon.CommitTrans

varRst.Close
varCon.Close
Screen.MousePointer = vbNormal
Exit Function

ErrorDB:
varCon.RollbackTrans
MsgBox "Error in saving data.Check the database system"
Screen.MousePointer = vbNormal

End Function
```

Codes for Searching A Matrix Number in the Database

```
Private Function gfSearchNoMatriks() As Boolean

  Dim lstrcnn As String

  Let gfSearchNoMatriks = False
  Screen.MousePointer = vbHourglass

  lstrcnn = "Provider=sqloledb;Data Source=kenji;Initial Catalog=PGIS;User
Id=sa;Password=; "
  Set varCon = New ADODB.Connection
  Set varRst = New ADODB.Recordset

  On Error GoTo ErrorDB

  varCon.Open lstrcnn

  lstrSQL = "SELECT MATRIKS FROM MSIF WHERE MATRIKS = '" &
txtKey(coFld_NoMatriks).Text & "'"
  varRst.Open lstrSQL, varCon, adOpenStatic, adLockReadOnly, adCmdText
  On Error GoTo 0

  With varRst
    If Not .EOF Then
      GoTo SearchSuccess
    End If
  End With

'  On Error GoTo 0
  varRst.Close
  varCon.Close
  Screen.MousePointer = vbNormal
  Exit Function

SearchSuccess:
  varRst.Close
  varCon.Close
  gfSearchNoMatriks = True
  Screen.MousePointer = vbNormal
  Exit Function


ErrorDB:
  MsgBox "Error in in retrieveing data.Check the database system"
  Screen.MousePointer = vbNormal

End Function
```

Function for Calculate and Return the age

```
Private Function ReturnAge(lstrDate As String) As String

  Dim lDateVar As Date

  If lstrDate <> "" Then
    lDateVar = CDate(lstrDate)
    ReturnAge = CStr(DateDiff("yyyy", lDateVar, Date))
  Else
    ReturnAge = ""
  End If

End Function
```

Codes for Printing The Report

```
Private Sub PrintReport(Destination As Integer)

  lintFirstPrint = lintFirstPrint + 1
  If GetBiodataToTmp Then
    ' to update the recordset for further reporting
    If lintFirstPrint <> 1 Then DEMain.rscmdBiodata.Requery
    If Destination = 0 Then
      RptBiodata.Refresh
      RptBiodata.Show
    ElseIf Destination = 1 Then
      RptBiodata.Refresh
      RptBiodata.PrintReport True
    End If
  End If

End Sub
```

Codes for putting the required data into a temporary table (Main1.frm)

```
Private Function GetBiodataToTmp() As Boolean

  Dim lstrcnn As String

  Screen.MousePointer = vbHourglass
  Let lstrcnn = "Provider=sqloledb;Data Source=kenji;Initial Catalog=PGIS;User
Id=sa;Password=; "

  Set varCon = New ADODB.Connection

  On Error GoTo ErrorDB

  varCon.Open lstrcnn

  'clear temp database first
  lstrSQL = "DELETE FROM MSIF_TMP"
  varCon.Execute lstrSQL, , adCmdText

  'put all the biodata into temp db
  lstrSQL = "INSERT INTO MSIF_TMP"
  lstrSQL = lstrSQL & " VALUES ('"
  lstrSQL = lstrSQL & txtKey(coFld_NoMatriks).Text & "','" & txtKey(coFld_KPBaru).Text
& "','"
  lstrSQL = lstrSQL & txtKey(coFld_KPLama).Text & "','" & txtKey(coFld_Nama).Text &
"','"
  lstrSQL = lstrSQL & txtKey(coFld_AlamatS).Text & "','" & txtKey(coFld_PoskodS).Text
& "','"
  lstrSQL = lstrSQL & txtKey(coFld_BandarS).Text & "','" & ComboNegeri.Text & "','"
  lstrSQL = lstrSQL & ComboNegara.Text & "','" & txtKey(coFld_NoTelS).Text & "','"
  lstrSQL = lstrSQL & lstrAlamatT & "','" & lstrPoskodT & "','" & lstrBandarT & "','"
  lstrSQL = lstrSQL & lstrNegeriT & "','" & lstrNegaraT & "','" & lstrNoTelT & "','"
  lstrSQL = lstrSQL & txtKey(coFld_EMail).Text & "','" & MEBSesi.Text & "','"
  lstrSQL = lstrSQL & txtKey(coFld_SemMasuk).Text & "'," & DTPDaf.Value & ","
  lstrSQL = lstrSQL & DTPLah.Value & ",'" & ComboJantina.Text & "','"
  lstrSQL = lstrSQL & txtKey(coFld_Bangsa).Text & "','" & txtKey(coFld_Agama).Text &
"','"
  lstrSQL = lstrSQL & ComboKahwin.Text & "','" & txtKey(coFld_Warga).Text & "','"
  lstrSQL = lstrSQL & txtKey(coFld_Mastautin).Text & "','" &
txtKey(coFld_KodIjazah).Text & "','"
  lstrSQL = lstrSQL & txtKey(coFld_Ijazah).Text & "','" & txtKey(coFld_KodMajor).Text &
"','"
  lstrSQL = lstrSQL & txtKey(coFld_Major).Text & "','" & ProgOption & "')"
  varCon.Execute lstrSQL, , adCmdText

  On Error GoTo 0

  varCon.Close
```

```
  Screen.MousePointer = vbNormal
  GetBiodataToTmp = True
  Exit Function

ErrorDB:
  varCon.Close
  Screen.MousePointer = vbNormal
  MsgBox "Error in database.Please check the database system"
  GetBiodataToTmp = False

End Function
```

Codes for Enabling and Disabling the Semester Box in Re-register Form

```
Private Sub ChkSem_Click(Index As Integer)

  If ChkSem(Index).Value = 0 Then
    If Index = 9 Then
      MEBSesi(Index).Enabled = False
      MEBSesi(Index).BackColor = &H8000000F
    Else
    If ChkSem(Index + 1).Value <> 0 Then
      If MEBSesi(Index).Text = "_____" Then
        For lintloop = Index To ChkSem.Count - 1
          ChkSem(lintloop).Value = 0
          MEBSesi(lintloop).Enabled = False
          MEBSesi(lintloop).BackColor = &H8000000F
        Next
      Else
        MsgBox "Tidak boleh padam kerana sudah daftar bagi semester ini"
        ChkSem(Index).Value = 1
      End If
    Else
      MEBSesi(Index).Enabled = False
      MEBSesi(Index).BackColor = &H8000000F
    End If
    End If
  Else
    If Index = 0 Then
      MEBSesi(Index).Enabled = True
      MEBSesi(Index).BackColor = &H80000005
    Else
      If ChkSem(Index - 1).Value <> 1 Then
        MsgBox "Tidak boleh daftar semester ini kerana pandaftaran semester sebelumnya
tiada lagi"
        ChkSem(Index).Value = 0
      Else
        MEBSesi(Index).Enabled = True
        MEBSesi(Index).BackColor = &H80000005
      End If
    End If
  End If
End Sub
```

A Tipical MDIForm Codes

```
Private Const coTaskbarHeight As Integer = 400

'command control
Private Const cmdLook As Integer = 0
Private Const cmdSave As Integer = 1
Private Const cmdClear As Integer = 2
Private Const cmdClose As Integer = 3
Private Const cmdExit As Integer = 4

'window menu control
Private Const Cascade As Integer = 0
Private Const Tile As Integer = 1

'edit menu control
Private Const Cut As Integer = 0
Private Const Copy As Integer = 1
Private Const Paste As Integer = 2
-------------------------------------------
Private Sub cmdCtrl_Click(Index As Integer)
  Dim TmpIndex As Integer

  If Index = cmdLook Then
    If ActiveForm Is Nothing Then
    Else
      If ActiveForm.ActiveControl <> 0 Then
        TmpIndex = ActiveForm.ActiveControl.Index
        ActiveForm.txtkey_KeyDown TmpIndex, vbKeyReturn, 0
      End If
    End If

  ElseIf Index = cmdSave Then

  ElseIf Index = cmdClear Then
    If ActiveForm Is Nothing Then
    Else
      ActiveForm.ClearAll
    End If

  ElseIf Index = cmdClose Then
    If ActiveForm Is Nothing Then
    Else
      ActiveForm.Form_Unload 0
    End If

  ElseIf Index = cmdExit Then
    MDIForm_Unload 0
```

```
  Else
    MsgBox "Error !Please close all pages and exit program.", vbCritical
  End If
End Sub
-----------------------------------------
Private Sub Initialize()
  'setting the mdiform to whole screen desktop size
  Top = 0
  Left = 0
  Height = Screen.Height - coTaskbarHeight
  Width = Screen.Width

End Sub
-----------------------------------------
Private Sub MDIForm_load()
  txtDate.Text = Date
  Initialize
  cmdCtrl(cmdSave).Enabled = False
End Sub
-----------------------------------------
Private Sub MDIForm_Unload(Cancel As Integer)
  Unload Me
End Sub
-----------------------------------------
Private Sub mnu_edit_Click(Index As Integer)

  If Index = Cut Then
    Clipboard.Clear
    Clipboard.SetText Screen.ActiveControl.SelText
    Screen.ActiveControl.SelText = ""
  ElseIf Index = Copy Then
    Clipboard.Clear
    Clipboard.SetText Screen.ActiveControl.SelText
  Else
    Screen.ActiveControl.SelText = Clipboard.GetText()
  End If

End Sub
-----------------------------------------
Private Sub mnu_editExit_Click()
  MDIForm_Unload 0
End Sub
-----------------------------------------
Private Sub mnu_editPrinter_Click()
  With CmnDlg
    .CancelError = False
    .ShowPrinter
  End With
```

```
End Sub
-----------------------------------------------
Private Sub mnu_MF_Click(Index As Integer)
  If Index = 0 Then
    mainCourse.formLoading
  Else
    mainDegree.formLoading
  End If
End Sub
-----------------------------------------------
Private Sub mnu_Window_Click(Index As Integer)
  If Index = Cascade Then
    'cascade child form
    MDIFrmMtn.Arrange vbCascade
  ElseIf Index = Tile Then
    'tile horizontal child form
    MDIFrmMtn.Arrange vbTileHorizontal
  Else
    MsgBox "Error !Please close all pages and exit program.", vbCritical
  End If
End Sub
```

Codes for Redirecting the Web Pages After Logging into the System

```
<%@ Language=VBScript %>
<% dim objCon, objRst
        dim lstrsql
        Response.Buffer = True
%>

<!-- METADATA TYPE="typelib" FILE="C:\Program Files\Common
Files\System\ado\msado15.dll" -->  ' for reference purposes

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>
<BODY>

<P> </P>

<%
dim lstruserid
dim lstrpassword
dim lstrconnection

set objCon = Server.CreateObject("ADODB.Connection")
set objRst = Server.CreateObject("ADODB.Recordset")

lstruserid = Request("userid")                    'set variable to request form's userid
lstrpassword = Request("password")

lstrconnection = "Provider=SQLOLEDB;Data Source=Kenji;Database=WEBBING;User
ID=sa;Password =;"

objCon.Open lstrconnection

lstrsql = "SELECT * FROM WPUMF WHERE USERID = '" & lstruserid & "';"
objRst.Open lstrsql,objCon,adOpenForwardOnly,adLockReadOnly,adCmdText

if objRst.EOF then
        Response.Write "No User Found"
else
        while not objRst.EOF
                If objrst("PASSWORD") = lstrpassword Then
                        dim lstrname,lstrvalue,lstrfield
                        for each lstrfield in objRst.Fields
                                lstrname = lstrfield.name
                                lstrvalue = lstrfield.value
                                Session(lstrname) = lstrvalue
```

```
                    next

                    Select case objRst("TYPE")
                    case "stud"
                        Response.Redirect "StudMain/StudMain.asp"
                    case "lect"
                        Response.Redirect "LectMain/LectMain.asp"
                    end select
                End If
                objRst.MoveNext
        wend
        Response.Write "<P><H2>Incorrect Password.</H2></P>"
        Response.Write "<P><H2>Hit The 'Back' Button To Try Again</H2></P>"
end if


objRst.Close
objCon.Close

Set objRst = nothing
Set objCon = nothing

%>

</BODY>
</HTML>
```

Appendix C

UNIVERSITI MALAYA
FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT

e-faculty

SISTEM MAKLUMAT PASCA SISWAZAH TARIKH : 07/02/01

SLIP BIODATA PELAJAR SARJANA MUKASURAT : 1

NO. MATRIKS: WGA98021 **SESI :** 19981999 **SEMESTER :** 1

NO. K/P BARU: 740609045270 **NO. K/P LAMA:**

NAMA: NOR AINI BINTI HARON

ALAMAT SESI : 34,JALAN 12/11,SEKSYEN 17/14

POSKOD : 50605 **BANDAR :** KUALA LUMPUR **NO. TELEFON :** 032354658

NEGERI : WILAYAH **NEGARA :** MALAYSIA

ALAMAT TETAP:

POSKOD : **BANDAR :** **NO. TELEFON :**

NEGERI : **NEGARA :**

EMAIL : AININOR@YAHOO.COM

IJAZAH : WGA SARJANA SAINS KOMPUTER **SECARA :** KD

PENGKHUSUSAN AI KEPINTARAN BUATAN

TARIKH LAHIR : 09/06/74 **WARGANEGARA :** MALAYSIA

JANTINA : P **MASTAUTIN :** MALAYSIA

BANGSA : MELAYU **TARAF KAHWIN :** SEORANG

AGAMA : ISLAM

UNIVERSITI MALAYA
FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT

e-faculty

SISTEM MAKLUMAT PASCA SISWAZAH

TARIKH : 07/02/01

SLIP PENDAFTARAN BERKURSUS & DISERTASI PELAJAR SARJANA

MUKASURAT : 1

NO. MATRIKS :	WGA98021		**SESI :**	19981999	
NO. K/P BARU :	740609045270		**SEMESTER :**	1	
NAMA:	NOR AINI BINTI				
ALAMAT :	34,JALAN 12/11,SEKSYEN 17/14				
POSKOD :	50605	**BANDAR :**	KUALA LUMPUR	**NO. TELEFON**	032354658
NEGERI :	WILAYAH	**NEGARA :**	MALAYSIA		
EMAIL :	AININOR@YAHOO				
IJAZAH :	WGA	SARJANA SAINS KOMPUTER			
PENGKHUSUSAN	AI	KEPINTARAN BUATAN			

KOD KURSUS	TAJUK KURSUS	KREDIT	STATUS
WXGA6101	ADVANCED ISSUE IN PROGRAMMING	3	PERCUBAAN
WXGA6102	REQUIREMENTS ACQUISITION AND MODELLING	3	PERCUBAAN
WXGA6103	OBJECT-ORIENTED TECHNIQUES	3	PERCUBAAN
WXGA6104	DISTRIBUTED SYSTEMS	3	PERCUBAAN

TARIKH DAFTAR : **SESI DAFTAR :** _____ **SEMESTER DAFTAR**

NAMA PENYELIA :

BIDANG PENYELIDIKAN

TAJUK DISERTASI

TARIKH SERAH

JUMLAH MUKASURAT : 1

UNIVERSITI MALAYA
FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT

e-faculty

SISTEM MAKLUMAT PASCA SISWAZAH

TARIKH : 07/02/01

SLIP BIODATA PELAJAR SARJANA

NO. MATRIK	NO. K/P	NAMA	NO. TELEFON	EMAIL	IJAZAH	KHUSUSAN	SESI	SEMESTE
WGA98008	72041208550	SANTHIMATHY A/P T VENUGOPAL			WGA	AI KD	19981999	1
WGA98011	70030108511	HADZARIAH BINTI ISMAIL			WGA	AI KD	19981999	1
WGA98014	74091908631	RADIN IZZATUL MUNA BINTI AHMAD ZABIDI			WGA	AI KD	19981999	1
WGA98016	72071902528	KAMARUL NAEMI KAMARUDIN			WGA	AI KD	19981999	1
WGA98018	74112214585	AEDAH BT ABD RAHMAN			WGA	AI KD	19981999	1
WGA98019	73031303523	HASLINA BINTI MAHMOOD			WGA	AI KD	19981999	1
WGA98020	73110510562	ROZIANA BINTI IBRAHIM			WGA	AI KD	19981999	1
WGA98021	74060904527	NOR AINI BINTI HARON	032354658	AININOR@YAHOO.COM	WGA	AI KD	19981999	1
WGA98023	74102907515	HAFIZA BINTI AHMAD			WGA	AI KD	19981999	1
WGA98028	1694361	SOLTANE MOHAMED			WGA	AI KD	19981999	1
WGA98031	73081012504	AUGUSTINA AEGIDIUS SITIOL			WGA	AI KD	19981999	1

JUMLAH MUKASURAT : 1

Appendix D

UNIVERSITI MALAYA
RANCANGAN IJAZAH TINGGI
Borang Notis Penyerahan Laporan Penyelidikan / Disertasi / Tesis

Bahagian A
(Untuk diisi oleh calon)

Kepada : Institut Pengajian Siswazah dan Penyelidikan
Universiti Malaya, 50603 Kuala Lumpur

Nama : .....

Alamat (terbaru) : .....
.....
.....

Tarikh : .....

Saya ..... seorang calon Ijazah Sarjana ...../ Doktor Falsafah ..... dengan ini menyatakan bahawa saya akan menyerahkan Laporan Penyelidikan / Disertasi / Tesis saya untuk diperiksa di dalam masa ..... bulan bermula dari .....

2. Tajuk akhir Laporan Penyelidikan / Disertasi / Tesis saya ialah :

Sekian, terima kasih.

.....
Tandatangan Calon)

Bahagian B
(Untuk diisi oleh Penyelia Utama)

Bahawa saya .....
Penyelia Utama bagi Encik/Puan/Cik calon Ijazah
.....telah diberitahu tentang hasratnya untuk menyerahkan Laporan Penyelidikan/ Disertasi/ Tesisnya untuk diperiksa dalam masa ..... bulan mulai dari .....

Dalam hal ini saya mengesahkan bahawa saya berpuashati dengan kemajuan yang dicapai oleh calon ini dan tiada apa-apa halangan terhadap calon tersebut menyerahkan Laporan Penyelidikan/ Disertasi/ Tesisnya untuk diperiksa dalam masa ..... bulan mulai dari .....

....
(Tandatangan Penyelia) (Tarikh)

Bahagian C
(Untuk diisi oleh Dekan/Pengarah, Akademi/ Fakulti / Institut / Pusat berkenaan)

Saya ..... Dekan/ Pengarah,Akademi/ Fakulti/ Institut/ Pusat
..... ingin mengesahkan perakuan yang dibuat oleh Penyelia Utama calon ini seperti yang tercatat di Bahagian B di atas.

....
(Tandatangan Dekan/ Pengarah dan cop) (Tarikh)

UNIVERSITI MALAYA
RANCANGAN IJAZAH TINGGI

Borang Notis Penyerahan Laporan Penyelidikan / Disertasi / Tesis

Bahagian A
(Untuk diisi oleh calon)

Kepada : Institut Pengajian Siswazah dan Penyelidikan
Universiti Malaya 50603 Kuala Lumpur

Nama : ----------

Alamat (terbaru) : ----------

No. Tel. Rumah : ----------

No. Tel. Pejabat : ----------

Alamat E-mail (jika ada) : ----------

Tarikh : ----------

Bersama-sama ini saya kemukakan dalam bentuk Jilid berkulit nipis empat (4) naskah Laporan Penyelidikan / Disertasi/ Tesis bertajuk:

---

untuk diperiksa sepertimana yang diperlukan.

2. Saya juga ingin mengesahkan bahawa Laporan Penyelidikan/ Disertasi/ Tesis tersebut telah pun disediakan oleh Penyelia Utama saya dan komen beliau adalah seperti yang terdapat di Bahagian B borang ini.

Sekian, terima kasih.

(Tandangan calon)

Bahagian B
(Untuk diisi oleh Penyelia Utama)

Nama Penyelia : .....

Akademi/ Fakulti/ Institut/ Pusat : .....

Saya telah menyemak Laporan Penyelidikan/ Disertasi/ Tesis , Encik/ Puan/ Cik seorang calon ljazah Sarjana/
Doktor Falsafah yang bertajuk:

.....

.....

Saya ingin mengesahkan bahawa:

1. Saya berpuashati dengan kemajuan yang dicapai dan saya tiada halangan terhadap Laporan Penyelidikan
   Disertasi/ Tesis tersebut diserahkan untuk diperiks
2. Saya bersetuju bahawa Laporan Penyelidikan/ Disertasi/ Tesis tersebut boleh diserahkan untuk diperiksa.
3. Lain-lain komen saya terhadap penyerahan tesis tersebut adalah seperti berikut (jika ada):

.....
.....

.....

Sekian, terima kasih.

.....
(Tandatangan Penyelia)

.....
(Tarikh)

Bahagian C
(Untuk diisi oleh Dekan/ Pengarah, Akademi/ Fakulti/ Institut/ Pusat)

Saya ..... Dekan/ Pengarah, Akademi/ Fakulti/ Institut/ Pusat ..... ingin:

1. Mengesahkan perakuan yang dibuat oleh Penyelia calon

.....
.....

.....
(Tandatangan Dekan/ Pengarah)

.....
(Tarikh)

UNIVERSITI MALAYA
RANCANGAN IJAZAH TINGGI
Borang Notis Penyerahan Laporan Penyelidikan / Disertasi / Tesis

Bahagian A
(Untuk diisi oleh calon)

Kepada : Institut Pengajian Siswazah dan Penyelidikan
Universiti Malaya, 50603 Kuala Lumpur

Nama : ----------

Alamat (terbaru) : ----------

No. Tel. Rumah : ----------

No. Tel. Pejabat : ----------

Alamat E-mail (Jika ada) : ----------

Tarikh : ----------

Bersama-sama ini saya kemukakan:

Empat (4) naskah berjilid tebal Laporan Penyelidikan/ Disertasi/ Tesis Ijazah Sarjana/ Doktor Falsafah bertajuk:

---

2. Saya juga ingin mengesahkan bahawa Laporan Penyelidikan/ Disertasi/ Tesis tersebut telah disemak oleh Penyelia Utama saya dan Dekan/ Pengarah, Akademi/ Fakulti/ Institut/ Pusat dan komen mereka adalah seperti yang terdapat pada Bahagian B dan C borang ini.

Sekian, terima kasih.

.....
(Tandatangan Calon)

Bahagian B
(Untuk diisi oleh Penyelia Utama)

Nama : .....

Akademi/ Fakulti/ Institut/ Pusat :

.....

Saya telah menyemak semua pembetulan/pindaan yang dilaksanakan oleh Encik/ Puan/ Cik .....
mengenai Laporan Penyelidikan/ Disertasi/ Tesisnya sebagaimana yang dipersetujui oleh Ahli Jawatankuasa Pemeriksa.

2. Saya ingin mengesahkan bahawa saya berpuashati dengan pembetulan yang dilaksanakan oleh calon.

Sekian, terima kasih.

.....
(Tandatangan Penyelia)

Bahagian C
(Untuk diisi oleh Dekan/ Pengarah, Akademi/ Fakulti/ Institutl Pusat)

Saya .....
Dekan/ Pengarah, Akademi/ Fakulti/ Institut/ Pusat

.....

ingin:

(i) Mengesahkan perakuan yang dibuat oleh Penyelia calon
(ii) Mengesahkan bahawa saya berpuashati dengan pembetulan/pindaan yang dilaksanakan oleh,calon sebagaimana yang dipersetujui oleh Ahli Jawatankuasa Pemeriksa.

.....
(Tandatangan Dekan/ Pengarah)

.....
(Tarikh)