University of Malaya

a tradition of excellence

e-Faculty Project

Financial Information System (FIS)

submitted by

Kuan Chien Wee
WET 98060

Under Supervision of

Pn. Salimah Mokhtar

Submission Date: 8 February 2001

Faculty of
Computer Science &
Information Technology

University of Malaya

Abstract

Financial Information System for the Faculty of Computer Science and Information Technology (FCSIT-FIS) is a client/server system introduced as a sub-system in the e-Faculty project. Its main purpose is to create a paperless, web-based system by providing on-line functions that allow users to access to FCSIT financial information.

The main functions of FCSIT-FIS are: on-line spending request, financial information retrieval and account maintenance.

FCSIT-FIS will be developing using ASP technology, VBScript, JavaScript and HTML language. Microsoft SQL Server will be use as the database server of the system.

University of Malaya

Acknowledgement

First, I must thank my supervisors, Pn. Mahfuzah Kamsah, Pn. Rodina Ahmad and Pn. Salimah Mokhtar, for their support, encouragement and thoughtful contribution throughout this project. I'd also like to thank Cik Nor Aniza Abdullah, my thesis moderator, for giving me helpful opinion and suggestion during the project viva.

I would also like to thank my senior buddy, Howe Chee Lin, who has been guiding me throughout this project and other university courses as well.

I am also grateful to Assoc. Professor Dr. Diljit Singh, vice dean of FCSIT, En. Yusoff Musa, assistant registrar, and Cik Zarina for explaining the system's need and the process flow of the project.

Special thanks must also go to all e-Faculty's project members, whose friendship over these past months helped make the whole effort possible.

But most of all, I thank my family for their love, patient and support throughout my life.

University of Malaya

Contents

University of Malaya

University of Malaya

University of Malaya

University of Malaya

List of Figures

University of Malaya

List of Tables

University of Malaya

1. Introduction

1.1 Introduction of e-University

The concept of the e-University was introduced during the mid-1980s when some colleges and universities, mainly in the US, attempted to make use of networked computer and telecommunication technologies to reach adult learners at a distance.

e-University could best be defined as a conceptual model that incorporates *stability* (*traditionalist*), *opportunities* (*progressivist*), *production efficiency* (*enterprising*), and *values* that meets the expectations of the society (*social constructivism*) [5].

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 nationwide plans for acquiring and using technologies.

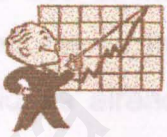
The new e-University will have the following features:

- High availability of ICT
- Integrated and accessible information systems
- A critical mass of technology users (faculty, students, and staff)
- Improved products (teaching and learning materials) and services to internal and external clients
- Improved collaboration with other institutions

University of Malaya

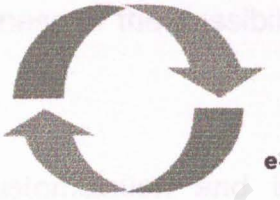
The following operational model is adapted from Bernbrock (2000) to suit the environment and the needs of Malaysian public universities. The model consists of 5 core processes: academic, administrative, student services, research, and institutional advancement.

e-student Services
- application and admission
- advising and counseling
- registration and scheduling
- access to records and library
- billing and payment

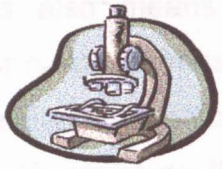


e-academics
- distributed learning
- distance learning
- collaboration
- online faculty services
- management of intellectual capital
- electronic classrooms

e-research
- high performance computing and communication
- collaboration and virtual meetings
- Internet 2



e-administration
- workflow
- integration of back office systems
- decision support tools
- e-procurement
- e-commerce with suppliers



e-institutional advancement
- public information and promotion
- course and program publication
- lifelong learning markets
- web marketing and business intelligence for research
- online bookstore

Figure 1.1 Operational Model of e-University

University of Malaya

1.2 Introduction of e-Faculty

The first level of the implementation of e-University will be at the faculty. Upon the successful implementation at the faculty level, the next implementation will be at each public university.

As a whole, e-Faculty project is to create an electronic environment that permits the transformation of key relationships and processes through the use of multimedia and web-based technologies. It aims to develop workable procedures for the use of technology in administration, building on and extending techniques already being used in teaching and research. By piloting these approaches the Faculty's intention is to trigger a greater awareness of the possibilities of electronic communication at all levels in the University.

This also means mass customization and intelligent integration of data and information across the campus in ways designed to provide higher levels of effectiveness, efficiency, and satisfaction. The goal of e-Faculty project is to create an infrastructure that supports constituent-centered communications and enables community-building and new connections throughout the campus.

1.3 Project Definition

Financial Information System of Faculty of Computer Science and Information Technology (FCSIT-FIS) is a sub-system of the e-Faculty project. It is an online program that allows users to access to FCSIT finance information.

The main function of FCSIT-FIS is to promote paperless and system's efficiency office in FCSIT by providing on-line spending request, financial information retrieval and account maintenance.

University of Malaya

The system also provides a means to reduce the process time of current FIS in FCSIT.

1.4 Objectives of FCSIT-FIS

The objectives of FCSIT-FIS are:

- Improve financial administration at the faculty.
- Automating manual procedures to improve them by reducing errors to less than1%, increasing speed and accuracy.
- Shortening data-processing time by 50%.
- Provide different level of access to the system.
- Create a real paperless office using electronic documents by reducing paper using.

1.5 Scope of FCSIT-FIS

FCSIT-FIS is a client/server application system where it can be divided into client site, application server and database server. The database will be setup in the database server, i.e. MS SQL server. The data retrieval will be done through online interaction with the database server or through application server.

In the FCSIT-FIS, there are three different categories of users: normal user, database administrators and super user.

Normal User

Normal users are the largest group. Basically there are two types of normal user: FCSIT workers and students. FCSIT workers are provided with common access to the system where they can only navigate through the application to apply for money

University of Malaya

withdrawal. They are not authorized to access to the system's account. On the other hand, students are only allowed to view general students' financial information.

System Administrators

Basically there are two types of administrator: database administrator and the department-head. Database administrator is a normal user with additional function. They are able to access to the system's account and make changes or modification on the data including all the data maintenance. They have to ensure the correctness, accuracy and completeness of data in the system as well as to update the data with latest information. On the other hand department-head has extra functions to approve or reject normal user's application forms.

Super User

Users with highest authority are called super users. Their roles are to give a user authority to access the system and control the accessibility of other user. Normally, Super user is not involved in database maintenance.

1.6 System Schedule

No.	Task	Start Date	Complete Date
1	Literature Review	July 1, 2000	September 5, 2000
2	System Analysis	August 1, 2000	October 15, 2000
3	System Design and Coding	September 15, 2000	December 30, 2000
4	System Implementation	October 10, 2000	January 31, 2001
5	System Testing	January 10, 2001	February 5, 2001
6	Documentation	January 15, 2001	February 2, 2001

Table 1.1 System Schedule Table

University of Malaya

1.6.1 FIS Development Schedule

ID	Task Name	Start	Finish	July	August	September	October	November	December	January	February
				Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
1	Literature Review	Sat 01/07/00	Tue 05/09/00								
2	System Analysis	Wed 02/08/00	Sun 15/10/00								
3	System Design	Fri 15/09/00	Sat 30/12/00								
4	System Implementation & Coding	Tue 10/10/00	Wed 31/01/01								
5	System Testing	Wed 10/01/01	Mon 05/02/01								
6	Documentation	Mon 15/01/01	Fri 02/02/01								

Figure 1.2 FIS Development Gantt Chart

Universiti Utara Malaysia

2. Literature Review

Methods of data collection in this project includes:

- Internet surfing – survey on the existing system and related sites
- Printed materials – magazines, manuals and reference books
- Interviews with current system users

2.1 Methodology - Soft System Methodology

Soft system Methodology (SSM), developed by Checkland (1981), is a framework for system analysis that provides very powerful techniques for the analysis of systems with human and social components, and has been widely applied to difficult problem areas

The Soft Systems Methodology is described by Wilson (1984) as a seven-stage process of analysis which uses the concept of a human activity as a means of getting from finding out about the situation to taking action to improve the situation. The initial stages are concerned with system analysis and the later stages with system design. The rich picture (Figure 4.1) can be applied to the initial stage of the knowledge elicitation process to help develop a representation of relevant domains, and an understanding of the views of people within each domain. Stowell and West (1990) suggest that the rich picture is very useful as a summary of the knowledge elicited from the expert. The analysis can use it as a prompt for discussion with expert, as an aid for assimilating knowledge elicited, and as a mean of identifying the areas in which knowledge is limited.

One of the key features of SSM process diagram (*Figure 3.1*) is the distinction between the "real world" and "systems thinking about the real world". This distinction is a key element in the SSM process, and one it shares with other techniques in this section – learning about the system of interest is driven by the mismatch between

University of Malaya

the "manager's" understanding of the system and the observed behavior in the real world.

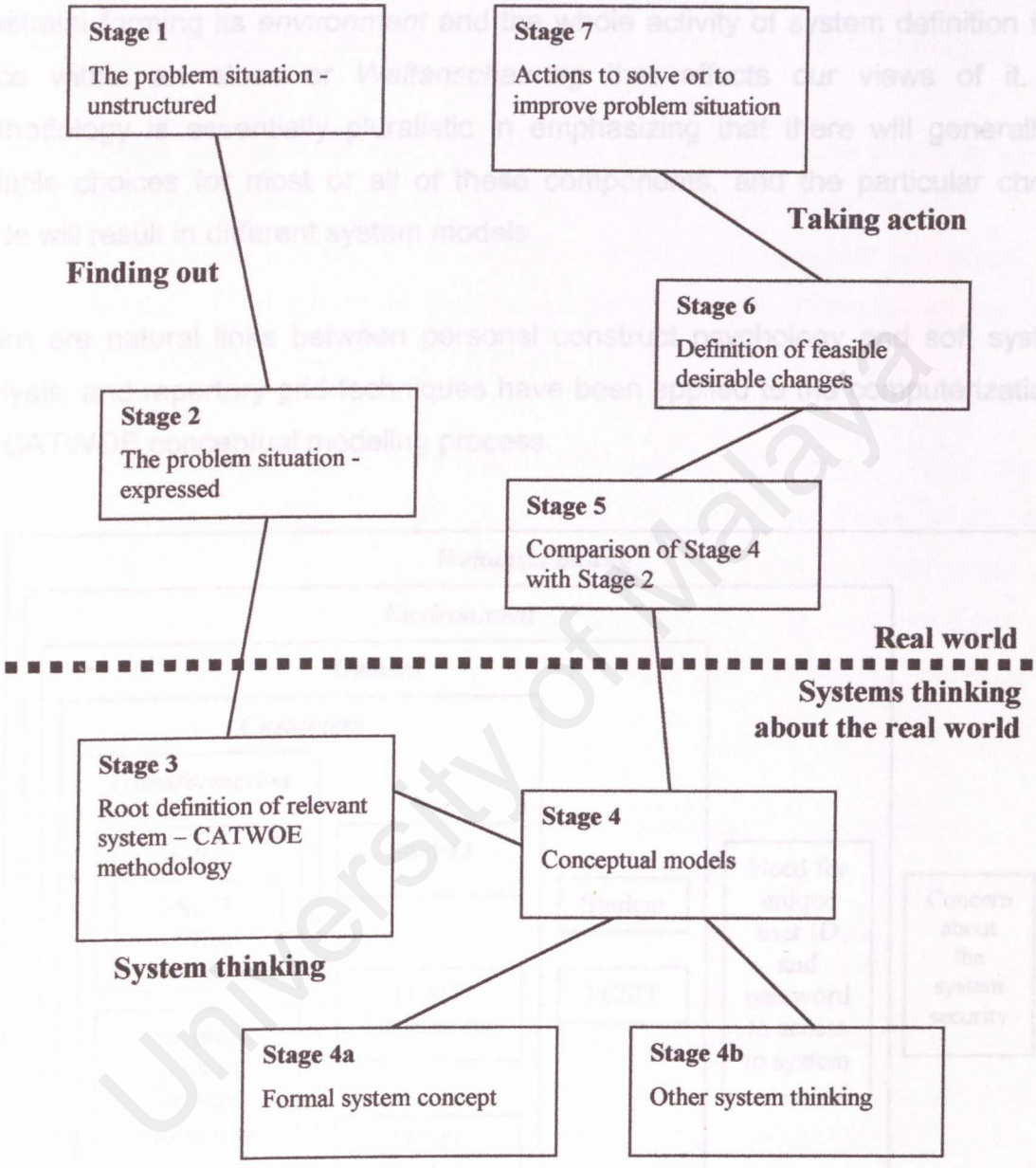


Figure 2.1 Seven Stages of Soft System Methodology

Checkland's methodology prescribes six essential components of a system that must be identified at the conceptual modeling stage. The CATWOE mnemonic is a reminder to search for each of these components in the system situation and make them overt in modeling. A system is defined through a *transformation* carried out by

people who are the *actors* within it. The system affects beneficially or adversely other people who are its *customers* and there is some agency with power of existence over it that is its *owner*. The system has to exist within an outside constraint forming its *environment* and the whole activity of system definition takes place within an ethos or *Weltanschauung* that affects our views of it. The methodology is essentially pluralistic in emphasizing that there will generally be multiple choices for most or all of these components, and the particular choices made will result in different system models.

There are natural links between personal construct psychology and soft systems analysis, and repertory grid techniques have been applied to the computerization of the CATWOE conceptual modeling process.

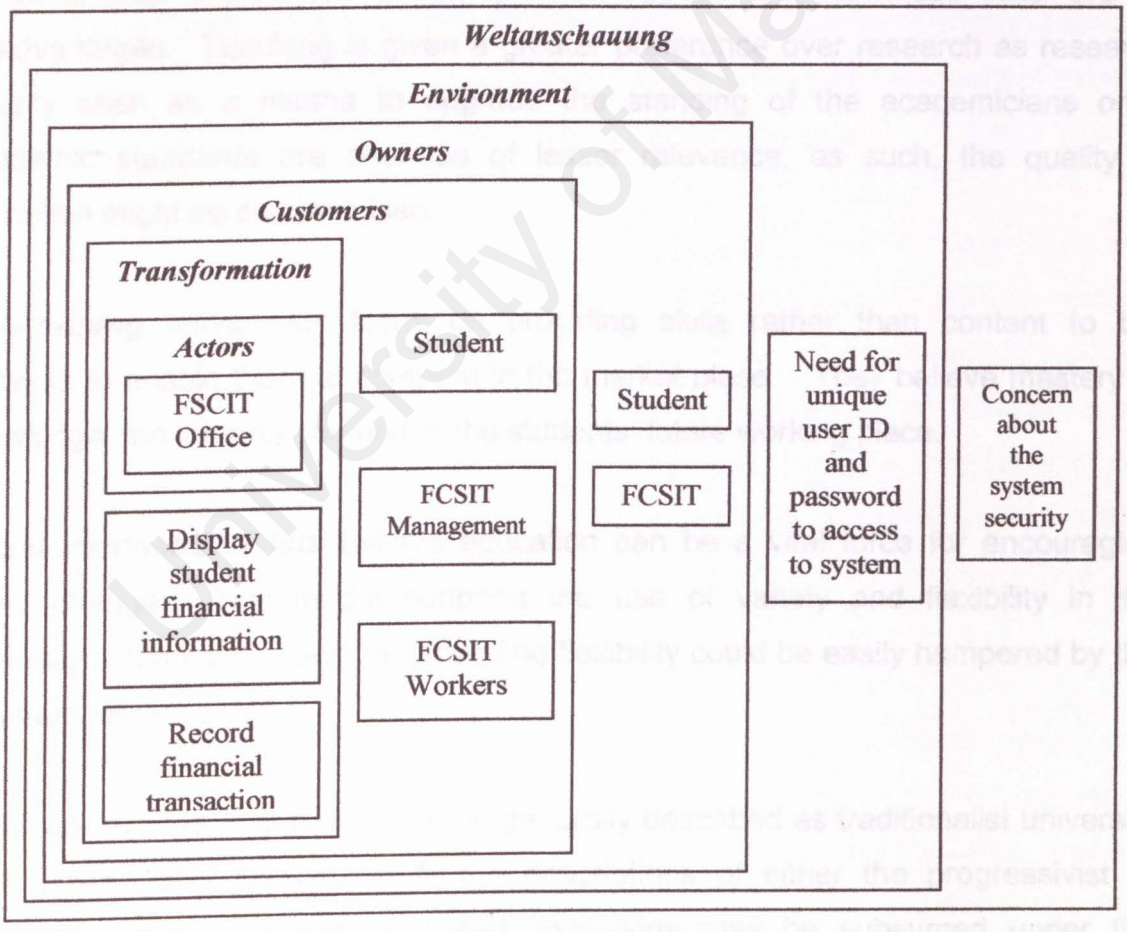


Figure 2.2 CATWOE analysis operationalized in a concept mapping tool

University of Malaya

2.2 Model of Malaysian Universities

Basically, universities in Malaysia fall into four ideological models (as classified by Trowler): *traditionalist*, *progressivist*, *enterprising*, and *social constructivist* [5].

Traditionalists view universities predominantly as elitist in nature where academics are custodian of cultural, research, and disciplinary heritage of society. In such universities, teaching and research are given comparatively equal emphasis. The major weakness of this model is the bureaucratic structure of the university, which may hamper real exploitation of technologies.

Progressivists believe on the equal opportunity in learning. They strongly advocate that universities should provide opportunities to those who are at disadvantaged. Teaching is given a greater preference over research as research is only seen as a means to upgrade the standing of the academicians only. Academic standards are seen as of lesser relevance, as such, the quality of education might be compromised.

Enterprising universities focus on providing skills rather than content to the students to enable them to compete in the market place. They believe mastery of knowledge can later be pursued in the students' future working place.

Social reconstructivists believe education can be a vital force for encouraging social change. This model supports the use of variety and flexibility in the university's curricula. However, providing flexibility could be easily hampered by the constraint of resources.

Public universities in Malaysia can be generally described as traditionalist university while private colleges closely fit the descriptions of either the progressivist or enterprising model. Religious-based institutions may be subsumed under the umbrella of social reconstructivism.

University of Malaya

2.2.1 Current Scenario of Malaysian Public Universities

Higher education in Malaysia, up to 1990, has been preoccupied with two main concerns: expanding progressively to meet the educational needs and demands of a growing population, and distributing tertiary education opportunities to redress ethnic imbalances (Gan and Ismail, 1998). Traditionally, Malaysian society puts a high premium on education and looks upon education as an important means of social immobility and economic advancement across lines of class or races.

In terms of overall policy for higher education in the 1990s, at least three trends have been discernible so far (Gan and Liang, 1998). First, there is the move towards the democratization of higher education to further expand access. Second, with an unprecedented increase in the participation of the private sector in the arena of higher education, there is a trend towards more and closer cooperation between the private sector and the public sector. Third, there is the empowerment of existing public universities to motivate and steer them to greater heights.

Expansion in terms of access has been rather dramatic: up until 1969, there was only one university in the whole country. Today, there are eleven public universities, five private universities – including a virtual university - and about 600 private colleges that run twinning programs with foreign universities. The Private Higher Educational Institutions Act 1996 officially recognized the role of private higher educational institutions in providing sufficient educational institutions in providing sufficient educational infrastructure as well as increasing the capacity to meet the demand for higher education.

Lack of flexibility in meeting the demands for dynamic education programs and lifelong learning, inadequate systems of governance, and rigorous admission requirements in the public universities have been cited as among the reasons why a growing number of prospective students in Malaysia view private learning institutions as a better option to pursue their tertiary education. Public universities now are under increasing pressure to be more flexible to the extent that there is a pressing need for them to review their missions and to reengineer their functions.

University of Malaya

2.3 Active Server Pages

Active Server Pages (ASP) is a tool for creating *dynamic web pages*. ASP is a technology developed by Microsoft. Pages using ASP are developed in JavaScript, VBscript, or Perlscript and are integrated into the HTML web pages. The ASP code is compiled on-the-fly by the server and the resulting output is standard HTML, which is then passed to the browser and used to create the page itself, on the user screen. By using ASP, web pages can be dynamic, full of ever-changing content, and browser independent.

The power of ASP lies in two facts: first, the HTML is not created until the user wants to see the web page, and second, it doesn't care what web browser is being used.

2.3.1 Static Pages vs. Dynamic Pages

A *static web page* is a page whose content consists of some HTML that was typed directly into a text editor and saved as an .htm or .html file. Thus, the author of the page has already completely determined the exact content of the page, in HTML, at some time before any user visits the page.

Static, pure-HTML files make perfectly serviceable web pages. We can even spruce up the presentation and usability of such pages by adding more HTML to create frames and tables. However, there's only so much we can achieve by writing pure HTML, precisely because their content is completely determined before the page is ever requested.

In order to overcome the weaknesses of static web page we need to replace the hard-coded HTML source with a set of instructions, which will be used to generate HTML for the page at the time the user requests the page. In others word, the page

University of Malaya

is generated *dynamically* on request. With this technology, we can capture all sorts of information that isn't known at the time the instruction are written – for example:

- The exact time when the page is requested
- The user's identity and preferences
- The type of browser they are using
- Other information provided by the user's request
- Information contained in databases, text files, XML files, etc

The HTML-generation instruction can be written in such a way that they use this newly captured information to create up-to-the-minute, personalized, interactive web pages that serve fresh information every time they are requested.

2.3.2 Dynamic Web Pages and ASP

ASP is a technology that allows for the programmatic construction of HTML pages just before they delivered to the browser. In other words, with ASP we can write a set of instructions that can be used to generate HTML just after the web page has been requested by the client, and just before it is delivered.

ASP is not a programming language although it does make use of existing languages such as JavaScript or VBScript. Moreover, it's not really as application either. Instead, ASP is more suitable described as a technology for building dynamic and interactive web pages.

2.3.3 ASP Code

When a web author writes an ASP page, it is likely to be composed of a combination of three types of syntax – some parts ASP, some parts HTML tags, and some parts pure text. All these constitution parts of the ASP are saved in a file with an .asp extension.

University of Malaya

ASP code is executed on the web server, and generates pure HTML; the client machine doesn't need to provide any kind of ASP support at all. The web browser handles .htm page and .asp pages in exactly the same way because from the browser point of view, the process sending a page request to a web server and receiving a stream of pure HTML.

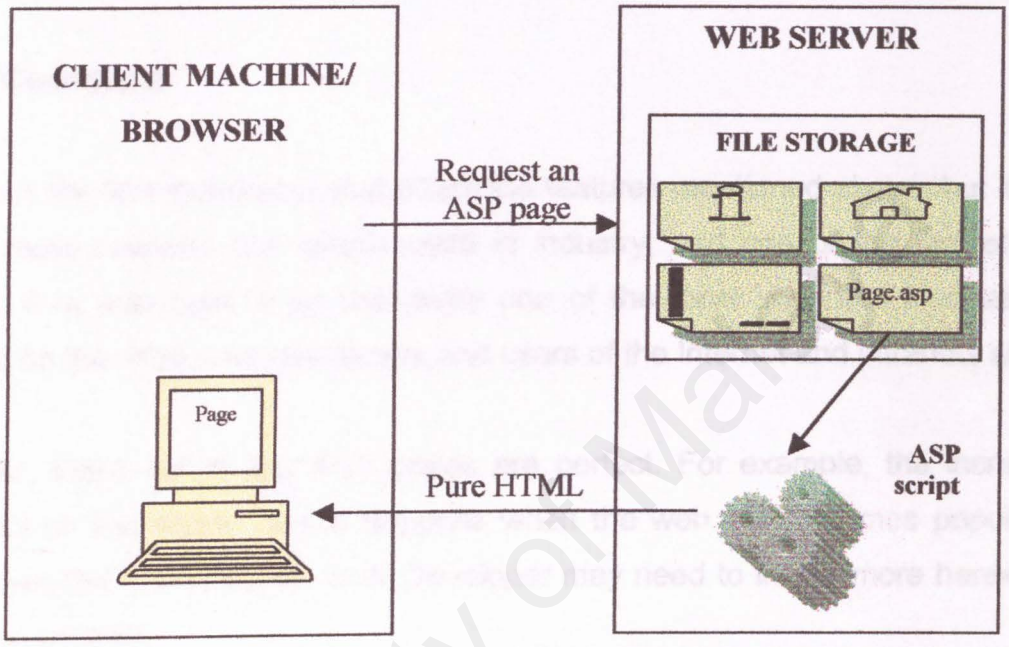


Figure 2.3 Requesting an ASP page

2.3.4 Advantages of Using a Server-side Technology

The advantages of performing actions on the web server are:

- Allows web developer to run programs in programming language that are not supported by user's browser
- Enables web developer to program dynamic web applications browser-independently, without recourse to client-side programming features such Java applets, Dynamic HTML, ActiveX controls, all which are browser specific
- Can provide the client (browser) with data that does not reside at the client

University of Ma

- Often makes for quicker loading times than with client-side dynamic web technologies such as Java applets or ActiveX controls, because in the end, user are actually only downloading a page of HTML
- Provides improved security measures, where web developer can writes code which can never be viewed from the browser

2.3.5 Comments

ASP isn't the first technology that offers the features mentioned above, but it is one of the most powerful and widely used in industry; and one of the fastest in the market. It is also said to be potentially one of the most important innovations to emerge on the Web – for developers and users of the Internet and intranets alike.

However, that's not to say ASP pages are perfect. For example, the increase of workload on the server, which happens when the web site becomes popular, will slow down the ASP process time. Developer may need to invest more hardware to solve the problem.

University of Malaya

2.4 Microsoft SQL Server

Microsoft SQL Server is a high-performance, *client/server* relational database management system (RDBMS). It was designed to support high-volume transaction processing (such as that for online order entry, inventory, accounting, or manufacturing) as well as data warehousing and decision-support applications.

2.4.1 The SQL Server Engine

The SQL Server engine is designed to support a variety of demanding applications, such as *online transaction processing* (OLTP) and decision-support applications. At the core of its decision-support capabilities is Transact-SQL, Microsoft's version of Structured Query Language. Beneath this query language are the components that support transaction processing and recoverability.

Transact-SQL

Industry wide, SQL is a well-known and widely used data access language. Every mainstream database management system (DBMS) product implements SQL in some way. Transact-SQL (T-SQL) is a powerful and unique superset of the SQL standard.

The SQL SELECT statement provides tremendous power and flexibility for retrieving information. Data from multiple tables can be easily projected and the results returned in tabular format with information chosen and correctly combined from the multiple tables.

Transact-SQL Extensions

Transact-SQL provides a number of capabilities that extend beyond typical implementations of SQL. Queries that are difficult to write in standard SQL can be easily and efficiently written using these capabilities. Some favorites include the ability to embed additional SELECT statements in the SELECT list and the ability to

University of Malaya

drill into a result set by further selecting data directly from a SELECT statement, a feature known as a *derived table*. Transact-SQL provides many system functions for dealing with strings (for finding sub strings and so on), for converting data types, and for manipulating and formatting date information.

Transact-SQL also provides mathematical operations such as square root. In addition, special operators, such as CUBE and ROLLUP, allow multidimensional analysis to be efficiently projected at the database server, where the analysis can be optimized as part of the execution plan of a query. The CASE expression allows for complex conditional substitutions to be made easily in the SELECT statement. Multidimensional (sometimes referred to as OLAP, or online analytical processing) operators, such as CUBE, and conditional expressions, such as CASE, are especially useful in implementing data warehousing solutions with SQL Server.

The Query Optimizer

In Transact-SQL, a cost-based query optimizer determines the likely best way to access data. This automatic optimization allows you to concentrate on defining your query criteria rather than defining how the query should be executed.

The SQL Server optimizer maintains statistics about the volume and dispersion of data, which it then uses to estimate the plan most likely to work best for the operation requested. Because a cost-based optimizer is by definition probability-based, an application might want to override the optimizer in some specialized cases. In your application, you can specify *optimizer hints* that will direct the execution plan chosen. In addition, you can use one of SQL Server's SHOWPLAN options, which explains the execution plan chosen, provides insight into why it was chosen and even allows for tuning of the application and database design.

University of Malawi

The Programmable Server

Transact-SQL provides programming constructs—such as variables, conditional operations (IF-THEN-ELSE), and looping—that can dramatically simplify application development by allowing you to use a simple SQL script rather than a third-generation programming language (3GL). These branching and looping constructs can dramatically improve performance in a client/server environment by eliminating the need for network conversations. Minimizing network latency is an important aspect of maximizing client/server application performance.

2.4.2 DBMS-Enforced Data Integrity

SQL Server enforces data integrity within the database itself, guaranteeing that complex business policies are followed and that mandatory relationships between data elements are complied with.

SQL Server uses advanced data integrity features, such as stored procedures, declarative referential integrity (DRI), data types, constraints, rules, defaults, and triggers, to enforce data integrity.

2.4.3 Symmetric Server Architecture

SQL Server uses a single-process, multithreaded architecture known as Symmetric Server Architecture that provides scalable high performance with efficient use of system resources. With Symmetric Server Architecture, only one memory address space is provided for the DBMS, eliminating the overhead of having to manage shared memory.

University of Malaya

2.4.4 Security

SQL Server provides numerous levels of security. At the outermost layer, SQL Server logon security is integrated directly with Windows NT security, allowing a Windows NT server to authenticate users. With this *Windows NT authentication* in place, SQL Server can take advantage of the security features of Windows NT, such as password encryption, password aging, and maximum length restrictions on passwords.

Windows NT authentication relies on *trusted connections,* which make use of the impersonation feature of Windows NT. Through impersonation, SQL Server can take on the security context of the Windows NT user account initiating the connection and test whether the security identifier (SID) has a valid privilege level. Windows NT impersonation and trusted connections are available with any of the available network interfaces (Net-Libraries) when connecting to SQL Server running under Windows NT.

For SQL Server running under Windows 95 or Windows 98, Windows NT authentication isn't available. An administrator must create SQL Server login accounts within SQL Server. Any user connecting to SQL Server must supply a SQL Server login name and password, regardless of whether she has already logged on to the network. This type of validation is what's known as *SQL Server authentication.*

SQL Server can be installed in a *mixed security* model, which means that Windows NT based clients can connect using Windows NT authentication, and connections that don't come from Windows NT clients, or that come across the Internet, can connect using SQL Server authentication. In addition, when connecting to an instance of SQL Server that has been installed with mixed security, a connection can always supply a SQL Server login name explicitly. This would allow a connection to be made using a login name distinct from the username in Windows NT. However, if you connect to an instance of SQL Server configured for only

Windows NT authentication, you won't be able to supply a SQL Server logon name, and your Windows NT username determines your level of access to SQL Server.

2.4.5 Distributed Data Processing

SQL Server provides features such as linked servers, remote stored procedure calls, and two-phase commit protocol that enables you to easily manage and use data in distributed environments. Microsoft Distributed Transaction Coordinator (MS DTC) was designed to be the vote collector and coordinator of transactions, and it allows many different types of systems to participate, laying the foundation for ACID transactions among heterogeneous systems.

A system participating in a transaction coordinated by MS DTC manages its own work and is called a *resource manager.* This resource manager system communicates with MS DTC, which coordinates all the resource managers participating in the transaction to implement the two-phase commit protocol. Distributed transactions honoring the ACID properties are supported as a whole: the entire distributed transaction at all sites either commits or aborts.

In the first phase of the two-phase commit protocol, all participating resource managers (that is, those that have enlisted in the transaction) *prepare to commit.* This means that they have acquired all the locksand resources they need to complete the transaction. MS DTC then acts as a vote collector. If it gets confirmation that all participants are prepared to commit, it signals to go ahead and commit.

The actual COMMIT is the second phase of the protocol. If one or more participants notify the system that it can't successfully prepare the transaction, MS DTC automatically sends a message to all participants indicating that they must abort the transaction. (In this case, an *abort,* rather than a commit, is the second phase of the protocol.) If one or more participants don't report back to MS DTC in phase one, the resource managers that have indicated they're prepared to commit (but haven't

committed because they haven't received the instruction to do so yet) are said to be *in doubt*. Resource managers that have transactions in doubt indefinitely hold the locks and resources necessary to ultimately commit or roll back the transaction, preserving the ACID properties. (SQL Server provides a way to force in-doubt transactions to abort.)

Another important distributed capability allows SQL Server to issue a *remote procedure call (RPC)* to another server running SQL Server. Remote procedure calls are stored procedures that can be invoked from a remote server, allowing server-to-server communication. This communication can be accomplished transparently to the client application, because the client can execute a procedure on one server, and that procedure can then invoke a procedure located on a different server.

Using RPCs can easily extend the capacity of an application without the added cost of reengineering the client application. Remote procedure calls optionally can be coordinated by the MS DTC service to ensure that the transactions maintain their ACID properties. The default behavior is to *not* execute the RPC in the context of a transaction so that if the local transaction is rolled back, work done by the RPC will still be committed. This behavior can be overridden by using the command BEGIN DISTRIBUTED TRANSACTION, or by setting the configuration option *remote proc trans* to 1.

Distributed data can also be managed by defining and accessing *linked servers*. A linked server can be any data source for which an OLE DB provider exists. The most commonly used OLE DB providers are the SQL Server OLE DB provider; the Jet provider, which allows you to connect to Microsoft Access databases; and the Open Database Connectivity (ODBC) provider, which allows you to connect to any ODBC data source.

A linked server is a logical name, defined using the stored procedure *sp_addlinkedserver*. The information you provide when you define the linked server includes the name and location of the datasource to connect to. Once you've

defined the name, you can use that name as the first part of a four-part name for any object, such as tables, views, or stored procedures. Objects on the linked server can be queried as well as modified, and they can be used in joins with local objects. The types of queries possible on the linked server depend on the capabilities of the particular OLE DB providers. Some providers, such as the one for SQL Server, allow all data modification operations and full transaction control across linked servers. Others, such as the OLE DB provider for text files, allow only querying of data.

2.5 Departmental Server

In the simplest case, all the documents available via a single web server come from a single source and none of the materials needs to be hidden from the users of that machine. In this case, using a single account to create web documents and making all of them world-readable allows them to be served over the web.

One significant drawback of this approach is that any user can modify all the documents with access to the single account being used and only by such a user. An obvious solution to this problem is to use separate accounts for different users or groups of users. This allows different individuals and groups to use a common server to distribute files provided that those files are readable by the server. In many cases, "readable by the server" can be adequately translated into "readable by any user on the system." After all, the web server is going to publish the documents to the world. But web access to various documents and services can be limited by a variety of mechanisms, and one is still left with the question of limiting access to documents by other users of the system. Running web servers only on systems with severely restricted login bases is one way to approach this problem. Judicious structuring of the login accounts and groups is the approach this proposal outlines.

There are situations where it is not appropriate to have all web documents left world-readable. The documents may come from multiple, unrelated sources. The

documents may contain sensitive information, and will need to be protected from access by other users on the machine. The web server may get the documents from a file system that is also exported to another networked machine, whose users should not have access to the information.

The solution is simplest when the documents only need to be protected from users not involved in their creation. A special group can be created that all information providers and the web server itself will share. The documents can be set to this group ownership, and will no longer need to be left world readable.

If information providers need to protect documents from each other, several groups can be created one for each information provider. The server will need to belong to all of these groups. This method is most secure, but is limited by the number of distinct groups that the operating system allows any single user to belong to.

All of these situations can be addressed by creating special web information providing accounts and groups, and setting memberships and privileges appropriate to individual situations.

2.5.1 Users Involve in Web Operation

Information Providers

Information Providers are users who write, edit and maintain documents that are published through the web. There are two categories of information providers. There are two types of information providers: individual information providers and institutional information providers.

Individual information providers are users who make their own documents available through the web. Individual information providers can be added to the web using two different methods. The web server often allows for a special directory in every user's account that is automatically available. For web servers lacking this feature, a special directory can be created in the web server's document space, and

individuals added with symbolic links pointing outside of the document space. All individual information providers are treated as a whole by the web server, so they will not be able to put special restrictions on their files.

Institutional information providers are users who provide institutional information through the web. Institutional information providers use special accounts (distinct from their personal use account) for placing documents into the web. Each such account may be used either by a single person or by several to promote collaborative work. In any case, all persons using this single account are treated as a unit. Files created by each institutional information provider will be protected from modification by others. Additional groups can be created to to protect files from access by other users of the system, while still allowing them to be served over the web. Because the web server pseudo-user must have access to all files to be served, the OS limit on the number of groups to which a single user can belong also limits the number of special information provider groups which can have materials served to the web but not readable by all all users on the system.

Web Server Pseudo-User

The web server itself must have access to all of the documents that can be made available via the web. In all other respects, it must be completely unprivileged. This is best accomplished by creating a special "user". This "user" isn't any individual or group of individuals, just a convenient/necessary mechanism. To call attention to this fact, we call it a "pseudo-user." By the very nature of its role, this pseudo-user must be able to access all information that any information provider wants to make available via the web.

Web Administrators

A *Web Administrator* is any one of the groups of individuals responsible for maintaining the web server. Via a common login, web administrators handle web server configuration and installation, as well as web-related communication with those using the web server to publish/provide information/services. Thus, that common login must have the same access to the files being served as the login and full write and update access to web configuration files.

System Administrators

System administrators are the team responsible for administration of the machine, such as creating accounts or managing disks. While it is possible to run a web server without any involvement from system administrators, this prevents almost all of the structure features mentioned in this document. At a minimum, system administrators must be involved in creating the special accounts and groups, arranging for the web server to start automatically, and restarting it when configuration changes are made. While some overlap between the system administrators and the web administrators is not unusual, the roles are distinct.

2.6 Client/Server

Client/server computing has created a deep paradigmatic shift in IT industry. It is replacing monolithic mainframe applications split across client and server lines. The client – typically a PC – provides the graphical interface, while the server provides access to shared resources – typically a database. Distributed objects and the Internet are new client/server revolution within the client/server revolution. Objects break-up the client and server sides of an application into smart components that can play together and roam across networks.

2.6.1 Client/Server Definition

Even though client/server is the leading industry buzzword, there is no consensus on what that term actually means. Basically, clients and servers are separate logical entities that work together over a network to accomplish a task. All client/server system will have the following distinguishing characteristics that differentiate it from other forms of distributed software:

- ***Service:*** Client/server is primarily a relationship between processes running on separate machines. The server is a provider of services. The client is a

consumer of services. In essence, client/server provides a clean separation of function based on the idea of service.

- ***Shared resources:*** A server can service many clients at the same time and regulate their access to shared resources.
- ***Asymmetrical protocols:*** There is a many-to-one relationship between clients and server
- ***Transparency of location:*** The server is a process that can reside on the same machine as the client or on a different machine across a network. Client/server software usually masks the location of the server from the clients by redirecting the service calls when needed. A program can be a client, a server, or both.
- ***Mix-and-match:*** The ideal client/server software is independent of hardware or operating system software platforms. User should be able to mix-and-match client and server platforms.
- ***Message-based exchanges:*** Client and server are loosely coupled systems that interact through a message-passing mechanism. The message is the delivery mechanism for the service request and replies.
- ***Encapsulation of services:*** The server is a "specialist". A message tells a server what service is requested; it is then up to the server to determine how to get the job done. Server can be upgraded without affecting the clients as long as the published message interface is not changed.
- ***Scalability:*** Client/server systems can be scaled horizontally or vertically. Horizontal scaling means adding or removing client workstations with only a slight performance impact. Vertical scaling means migrating to a larger and faster server machine or multi-servers.
- ***Integrity:*** The server code and server data is centrally maintained, which results in cheaper maintenance and the guarding of shared data integrity. At the same time, the clients remain personal and independent.

The client/server characteristics described above allow intelligence to be easily distributed across a network. These features also provide a framework for the design of loosely coupled network-based application.

2.6.2 Fat Servers vs. Fat Client

Client/server models can be distinguished by the server they provide and how the distributed application is split between the client and the server.

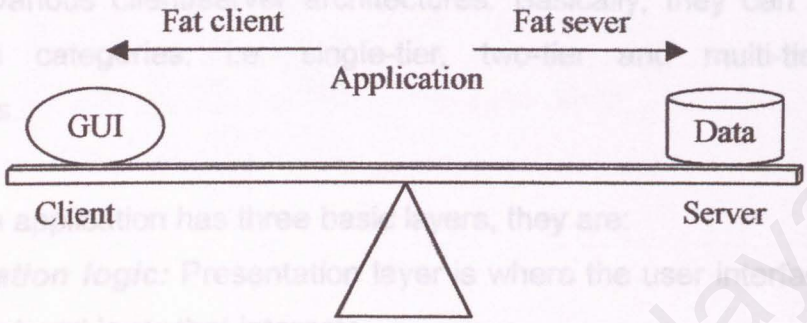


Figure 2.4 Fat Servers vs. Fat Client

Fat clients are the more traditional forms of client/server. The bulk of the application runs on the client side of the equation. In both the file server and database server models, clients know how the data is organized and stored on the server side. Fat clients are used for decision support and personal software. They provide flexibility and opportunities for creating front-end tools that let end-users create their own application.

Fat server applications are easier to manage and deploy on the network because most of the code runs on the servers. Fat servers try to minimize network interchanges by creating more abstract levels of service. Transaction and object servers, for example, encapsulate the database. Instead of exporting raw data, they export the procedures (or methods in object-oriented terminology) that operate on that data. The client in fat server model provides the GUI and interacts with the server through remote procedure calls (or method invocations).

Each client/server model has its uses. In many cases, the models complement each other, and it is not unusual to have them coexist in one application. For example, a GroupWare imaging application could require an "all-in-one" server that combines

University of Malaya

file, database, transaction, and object services. Fat servers, used for mission-critical application, represent the new growth area for PC-based client/server computing.

2.6.3 Types of Client/Server Architecture

There are various client/server architectures. Basically, they can be divided into three main categories, i.e. single-tier, two-tier and multi-tier client/server architectures.

Typically, an application has three basic layers, they are:

- ***Presentation logic:*** Presentation layer is where the user interfaces are build. It is the front-end layer that interacts.
- ***Business logic:*** Business layer will handle the business rules of the application.
- ***Data access logic:*** Data storage and retrieval are handled by the data access logic. It plays important roles in data integrity and maintenance.

Single-tier Client/Server

The single-tier client/server architecture is the first generation of client/server system and it has a single application layer to support the user interface, business rules and the data access logic. In this architecture, data can be stored not in the same location as the application, but the data access logic is in the application layer.

Two-tier Client/Server

The two-tier client/server architecture is more regarded to splitting of process to workstation layer and database. It is mainly for network traffic reduction as the processing of data is being carried out in the database server instead of transfer all the data to client site (a workstation) to be processed. There are two types of two-tier client/server i.e. fat client/thin server and thin client/fat server. Fat client/thin server is more commonly used because it is most simple to implement. The presentation and business logic are implemented in the client site, while the data access layer is with the database server. In the thin client/fat server architecture, business logic is pushed to the server and it is invoked by sending a request to the

University of Malaya

server that includes the stored procedure name and any other parameter values. This request can be easily transmitted using remote database protocols and implemented simply using database middleware like Microsoft's Open Database Connectivity (ODBC).

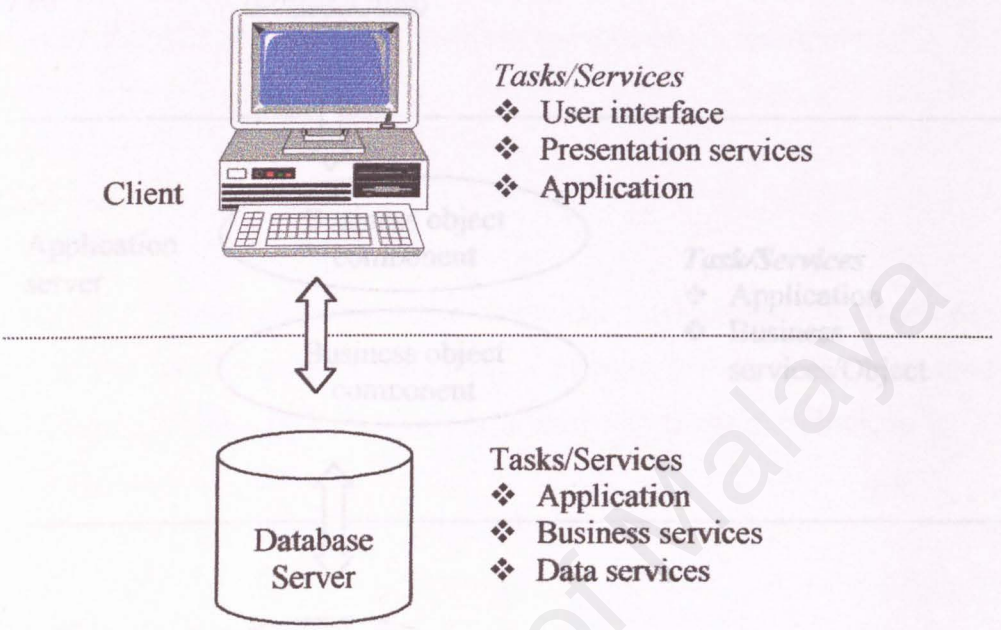


Figure 2.5 Two-tier Client/Server

Multi-tier Client/Server

Multi-tier client/server architecture is design to solve the problems encountered in two-tier system. There are two categories of multi-tier system, which is three-tier client/server and N-tier client/server.

Three-tier client/server is an architecture where an additional middle tier is inserted between a thin client and thin server. The middle tier is responsible to act upon client request, apply business logic and invoke the database server request. It also will have to handle the database responses, apply further business logic and generate a client response. Thus, when the business logic need to be enhanced, the system only need to develop a new user interface and used the existing layer of business logic and data access logic.

University of Malaya

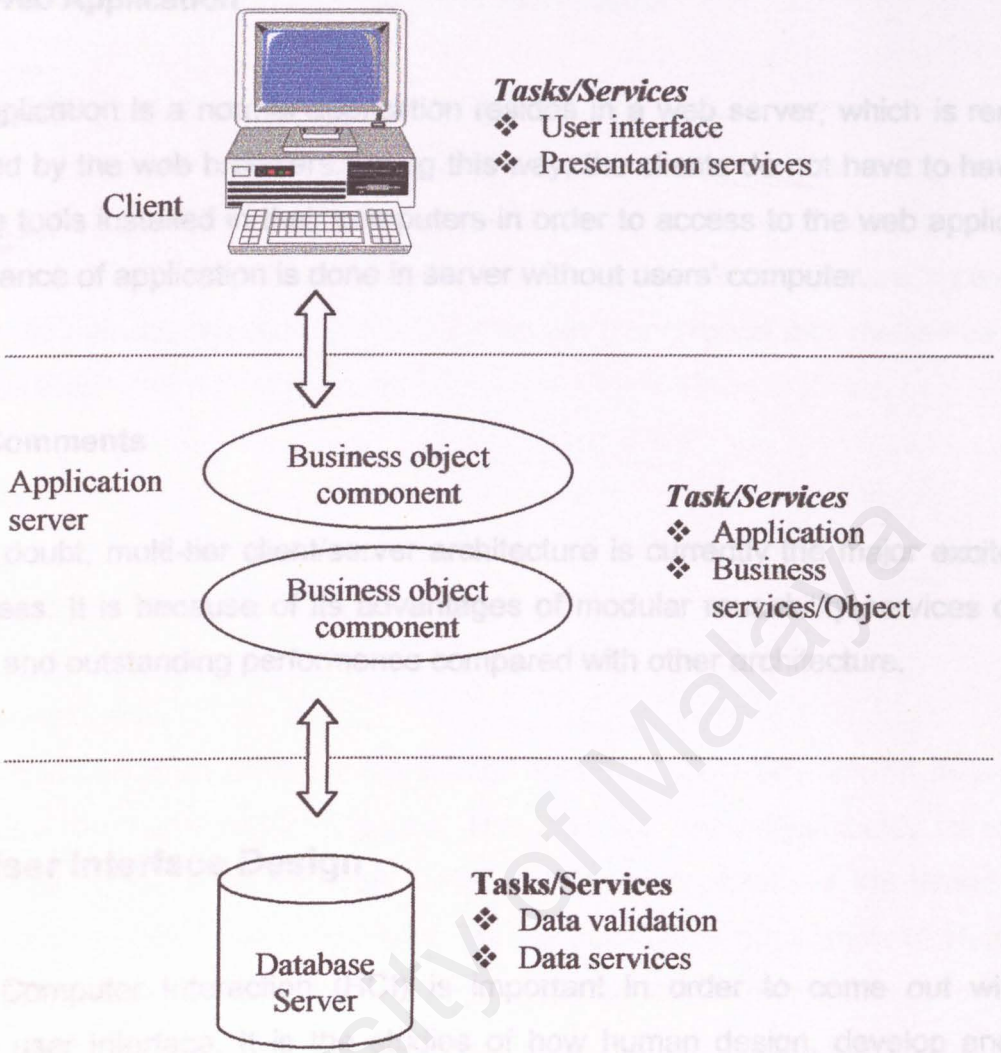


Figure 2.6 Three-tier Client/Server

A further extension of three-tier architecture is the N-tier client/server. In this system, the business logic is partitioned and distributed over several machines. Whereas the middle tiers will have connection to various types of services, integrate them with the client, and to each other. Besides that, the N-tier system is scalable as it enables distribution of the workload over many CPUs (using either symmetric multiprocessing or massively parallel clustered technology).

University of Malaya

2.6.4 Web Application

Web application is a normal application resides in a web server, which is remotely accessed by the web browsers. Using this way, the clients do not have to have any software tools installed in their computers in order to access to the web application. Maintenance of application is done in server without users' computer.

2.6.5 Comments

Without doubt, multi-tier client/server architecture is currently the major excitement in business. It is because of its advantages of modular reusability services on the network and outstanding performance compared with other architecture.

2.7 User Interface Design

Human Computer Interaction (HCI) is important in order to come out with an effective user interface. It is the studies of how human design, develop and use interactive computer system as well as how the computers affect individuals, organizations, and society. The studies include new interaction techniques for supporting user tasks, a better access to information and more powerful forms of communication.

HCI involves:

- Input and output as well as the interaction techniques of using them
- The way of control and monitor computer's actions
- The way of request and present information
- Types of help, documentation and training
- Processes involve during interface development

Tools used to design, develop, test and evaluate user interface

University of Malaya

2.7.1 Techniques of Human Computer Interaction

- ***Direct Manipulation of Graphical Objects***

  Direct manipulation of graphical objects is regarded to the manipulation of objects using a light-pen, grabbing objects, moving of objects, changing of size as well as using constrains, widgets, iconic representation, gesture recognition, dynamic menus, selection of icons by pointing, and mode and mode-free styles of interaction.

- ***Windows***

  Multiple tiled windows are used for views of multiple programs or data or for multiple views of a single program or data block.

- ***Hypertext Link***

  Hypertext link is regarding a highlighted text that could be click on in order to navigate to other screens or pages. This idea has been significantly introduced by HyperCard from Apple (1988) widely. The rapid growth of the World Wide Web is one of the main results of Tim Berners-Lee's application of Hypertext Link as the interface to mostly existing capabilities of the Internet.

- ***Others***

  There are many other examples of HIC research including work that led to drawing programs, paint programs, animation systems, text editing, spreadsheets, multimedia, 3D, virtual reality, interface builders, event-driven architectures, usability engineering and a very long list of other significant developments.

2.7.3 Graphical User Interface and Standards

Graphical User Interface (GUI) provides a standard look and feel of applications. It has contribute to the less time spent to learn and develop an application. A GUI is an application environment that can work with graphical object.

University of Malaya

2.7.3 Benefits of GUI

- Consistency and standard user interfaces enable users to learn easily and faster especially when come to second application because they have familiar with the environment.
- What You See Is What You Get (WYSISYG) concept is very useful. For example, output can be preview before printing.
- Enable manipulation of objects, lines and curves and filled closed areas with color.
- GUI able to specify the data fields when setting up reports and select sort keys. It also enables transfer data from or to other applications.
- Extend the direct manipulation approach. The user can, for example, drag file icons to the printer without loading the application that created them.

2.7.4 Drawbacks of GUI

Space and memory consumed. The costs include the expense of graphic cards, pointing devices and extra memory. As GUI run in graphic mode, screen refresh is usually slower as well. If speed is important, a GUI's consistency may not be sufficient compensation.

2.7.5 Comments

In the system design user interface will be design based on consistency, simplicity and user friendly. Besides that, the system will be designed, as it is suitable to deploy as a web-page application.

University of Malaya

3. System Analysis

Requirements engineering is the process of establishing the services the system should provide and the constrains under which it must operate. Requirements analysis is part of the requirements engineering process.

Requirements analysis is the process of deriving the system requirements through observation of existing system, discussions with potential users and procures, task analysis and so on. It may involve the development of one or more different system models. System prototypes may also be developed to help understand the requirements.

3.1 Current Environment

Currently, there's no computerize financial information system in the Faculty of Computer Science and Information Technology. All financial transactions in the faculty are recorded manually in an account book. Besides that, the faculty also doesn't have a web page for student financial information.

Basically, the current financial system in FSCIT can be divided into two transactions: income and expenses transaction. There are two main source of income for the FCSIT, the first and main income is from the Treasurer of University of Malaya and the secondary income is from renting out properties of FSCIT.

Every year UM's Treasurer will allocate an amount of budget for all faculties and departments in UM; the amount of money will then be debited into each faculties' or departments' account separately. The total figure of the budget will be send to the faculty's dean by mail. Then FCSIT's dean will validate the amount of money allocated and send a mail stating the amount of money to the faculty's account clerk

University of Malaya

for recording purpose. If the faculty's dean would like to allocate more budget for the faculty, he will apply from the UM's Treasurer through mail.

Another income of the FSCIT is through renting out the faculty's properties such as: lecturer hall, computer laboratory and so on. Vice-dean is authorize to approve the renting and all the income statement will be send to the account clerk for recording.

Expenses or credit from the account book happens when someone in the faculty spends on FCSIT's account. If a FCSIT worker, e.g. lecturer, system analyst, chief officer or technician, wanted to buy something for faculty use or spend on the faculty's account, they will apply the budget or claims from dean through order forms. Currently there are three types of order form (Appendix B) according to the amount of money applied. The order form will be send to the dean via the office clerk. If the dean approve the application, the order form will be send back to the account clerk; else, a reject letter will be send to the applicant.

When the office clerk receives the approved order form, she will issue purchase order and send it back to the dean for approval. After that, the purchase order will be sending to the supplier and at the same time the account clerk will record the transaction into the account book.

University of Malaya

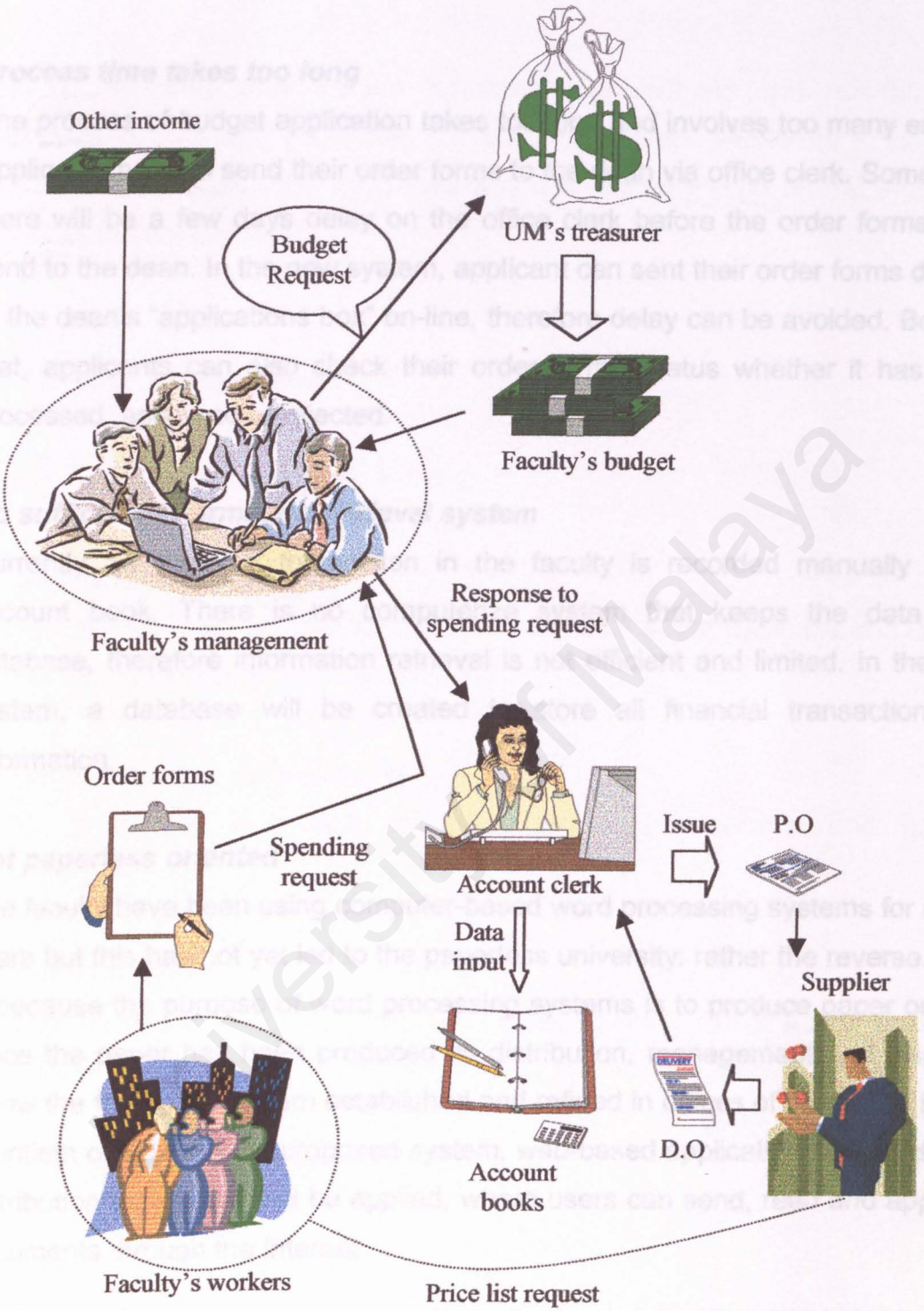


Figure 3.1 A rich picture of current FCSIT's financial system

University of Malaya

There are a few problems or weakness in the current system:

- ***Process time takes too long***

  The process of budget application takes too long and involves too many entities. Applicants have to send their order forms to the dean via office clerk. Sometimes there will be a few days delay on the office clerk before the order forms were send to the dean. In the new system, applicant can sent their order forms directly to the dean's "applications box" on-line, therefore delay can be avoided. Besides that, applicants can also check their order form's status whether it has been processed, approved or rejected.

- ***No sufficient information retrieval system***

  Currently, all financial transaction in the faculty is recorded manually in an account book. There is no computerize system that keeps the data in a database, therefore information retrieval is not efficient and limited. In the new system, a database will be created to store all financial transaction and information.

- ***Not paperless oriented***

  The faculty have been using computer-based word processing systems for many years but this has not yet led to the paperless university: rather the reverse. This is because the purpose of word processing systems is to produce paper output. Once the paper has been produced its distribution, management and disposal follow the traditional pattern established and refined in offices of every sort in the twentieth century. In the proposed system, web-based application and electronic distribution technology will be applied, where users can send, read and approve documents through the Internet.

University of Malaya

3.2 Requirements Analysis

System requirements may include functional requirements and non-functional requirements. A functional requirement describes a system service or function; on the other hand, non-functional requirement is a constrain placed on the system or on the development process.

3.2.1 Functional Requirements

Functional requirements needed in FIS are:

- ***Data searching***
  The system must be able to search archive data in the database and display the information in appropriate form.

- ***Display current financial status***
  The system must be able to display the account balance and monthly expenses.

- ***Authentication***
  Security is one of the most important features in FIS. Thus authentication is needed to control the access level of each user. Basically, there are three levels of access control and they are mainly for super user, database administrator and normal user.

- ***Multiple user access***
  The system must be able to be accessed by more than one user at the same time.

University of Malaya

- ***Data maintenance***

  Data in the database can be maintained by the system. Updating of the latest account must be handled.

3.2.2 Non-functional Requirements

Non-functional requirements that must be included in FCSIT-FIS are:

- ***Maintainability***

  The system can be corrected should an error is encountered, can be adapted should there be changes in requirements, or enhanced in the future.

- ***User friendliness***

  GUI design will focus on user friendliness and the ease of learning and understanding by all levels of users.

- ***Help***

  The system should be able to provide user manual to guide user in using the system.

- ***Reliability, robustness and accuracy***

  The system needs a robust interface to prevent interface errors from corrupting the system. Besides that, the system also needs to have the ability to checks on the system inputs to ensure correct data is provided in order to protect system integrity.

University of Malaya

3.2.3 Hardware and Software Requirements

Hardware and software required in the Financial Information System can be divided to client site and server site.

The clients' computer is any personal computer with network interface card (NIC), network connection and a browser (IE or Netscape).

Hardware requirements for the server are:

- A multimedia PC with a Pentium II processor
- 64MB RAM or more
- 2 GB of hard disk space
- Network Interface Card and network connection with recommended n\bandwidth at 10Mbps or more

Software that is needed in the server for FIS are:

- Microsoft Windows 2000
- Internet Information Server (ISS)
- Microsoft Visual Basic

University of Malaya

4. System Design

The information collected earlier in the System Analysis stage is used to accomplish the logical design of FIS. However, designing is a more creative and challenging process than analyzing. It includes:

1. system architecture design;
2. database design;
3. user interface design; and
4. system functionality design

4.1 FIS Architecture Design

FIS will deploy the client/server architecture and will be distributed into several layers.

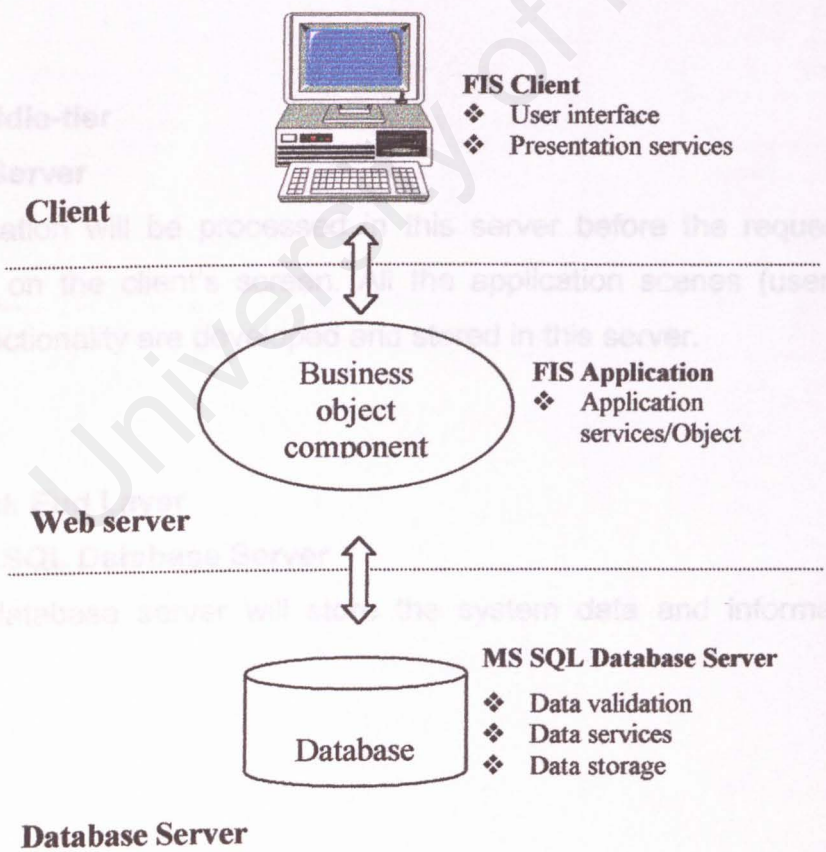


Figure 4.1 The FCSIT-FIS Architecture

University of Malaya

The FCSIT-FIS will be using the distributed architecture where the application components of the system are distributed to several layers by taking into consideration the nature of the application and the development tool used. The purpose of using this type of architecture is to maximize the performance of FCSIT-FIS.

The FCSIT-FIS application is executed on the client machine. The client as the front-tier communicates with the backend-tier, Microsoft SQL server through the middle-tier whereby the middle-tier consists of the Microsoft IIS.

4.1.1 Front End Layer

FIS Client

The FIS web client is a thin client, which needs only a web browser such as: Internet Explorer and Netscape to access to the system.

4.1.2 Middle-tier

FIS Web Server

FIS application will be processed in this server before the requested pages are presented on the client's screen. All the application scenes (user interface) and system functionality are developed and stored in this server.

4.1.3 Back End Layer

Microsoft SQL Database Server

The FIS database server will store the system data and information in objects manner.

University of Malaya

4.1.4 Communication Among Tiers

When a client wanted to access to the system, a connection between the client site and web server will be established, and from there, the web server will interact with the database server to retrieve the necessary data.

4.2 Database Design

In database design, structures and framework of FIS database are defined. The database is designed base on the object database model.

4.2.1 Database Object Model

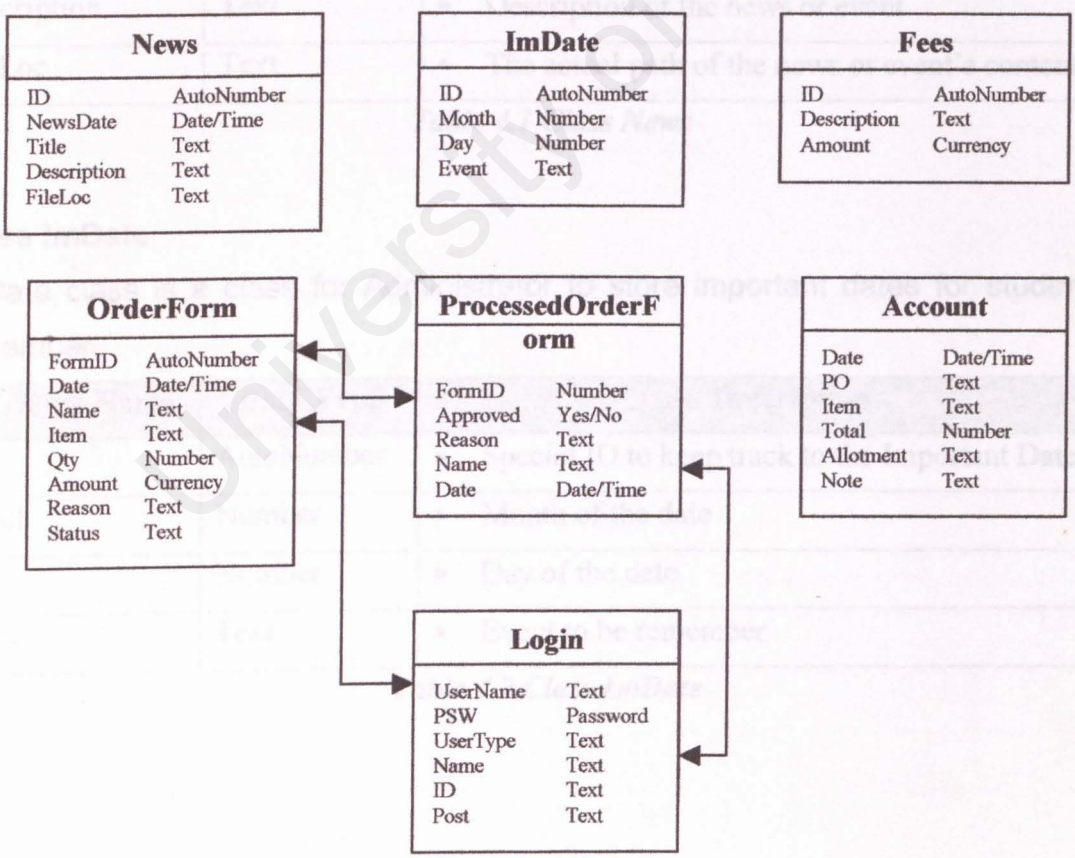


Figure 4.2 Database Object Model

From the object database model, the objects (as classes) and their relationships are drawn. Arrow lines connecting the classes represent relationships between each class.

4.2.2 Data Dictionary

Class News

News class is a class for Administrator to store latest financial news and events in the FCSIT.

Attribute Name	Data Type	Description
ID	AutoNumber	• Special ID
NewsDate	Date/Time	• The news or event posted date
Title	Text	• The news or event title
Description	Text	• Description of the news or event
FileLoc	Text	• The actual path of the news or event's content

Table 4.1 Class News

Class ImDate

ImDate class is a class for Administrator to store important dates for students to remember.

Attribute Name	Data Type	Description
ID	AutoNumber	• Special ID to keep track to the Important Date
Month	Number	• Month of the date
Day	Number	• Day of the date
Event	Text	• Event to be remember

Table 4.2 Class ImDate

University of Malaya

Class Fees

Fees class is a class for Administrator to store general fees information.

Attribute Name	Data Type	Description
ID	AutoNumber	• Special ID to trace fees
Description	Text	• Description of the fees
Amount	Currency	• To store the cost of fees

Table 4.3 Class Fees

Class Account

Account class is to store the account information of FCSIT.

Attribute Name	Data Type	Description
Date	Date/Time	• Date of the PO issued
PO	Text	• To Store the Purchase Order ID
Item	Text	• Description of the item
Total	Currency	• Total amount
Allotment	Text	• To store the allotment code
Note	Text	• Special note added by administrator

Table 4.4 Class Account

Class Login

Login class is needed to store the information of users' details. It is also very important to limit certain users' authorities.

Attribute Name	Data Type	Description
UserName	Text	• User's login ID
PSW	Text	• User's password
UserType	Text	• User type limits user's functionality
Name	Text	• User's full name
ID	Text	• User's employee ID
Post	Text	• User's post in FCSIT

Table 4.5 Class Login

University of Malaya

Class OrderForm

OrderForm class is to store information of user's submitted order form.

Attribute Name	Data Type	Description
FormID	AutoNumber	• Special ID to keep track of the order form
Date	Date/Time	• Date of the form submitted
Name	Text	• Name of the applicant
Item	Text	• Item requested
Qty	Number	• Quantity of the item
Amount	Currency	• Amount of the item
Reason	Text	• Reasons to buy the ordered item
Status	Text	• Indicate whether the form has been processed

Table 4.6 Class OrderForm

Class ProcessedOrderForm

ProcessedOrderForm is needed to store the processed order form information.

Attribute Name	Data Type	Description
FormID	Text	• Special ID to keep track of the order form
Approved	Yes/No	• To indicate the form has been approved or rejected
Reason	Text	• Reason to approve or reject the order form
Name	Text	• Name of user
Date	Date/Time	• Date of the form processed

Table 4.7 Class ProcessedOrderForm

University of Malaya

4.3 User Interface Design

FIS is designed in such a way to be published through web. The following are guidelines used in FIS user interface design.

- **Consistent labels, standard abbreviations and predictable colors**
  Scenes of the FIS use a consistent format of label, command input, color selection, etc. this will improve the user friendliness of the system and provide a more tidy design.

- **Offer meaningful feedback**
  FIS provides the user with visual feedback. This ensures that two-way communication between the user and the interface is established.

- **Page layout and presentation**
  The page layout and presentation is regarded how the page look like it supposed to when rendered by different browser.

University of Malaya

4.3.1 Prototype of User Interface

The user interface design of FIS are shown as the following:

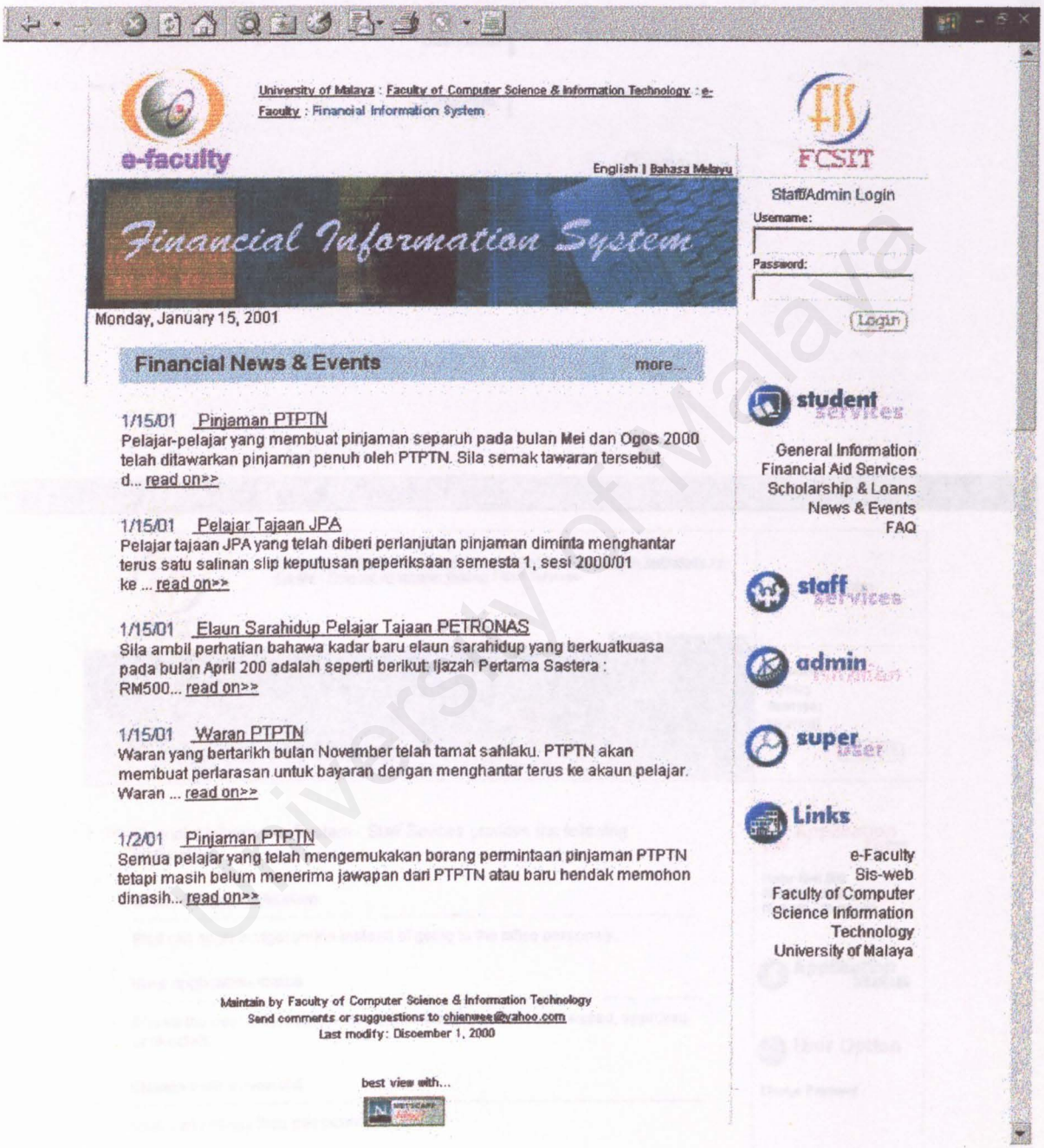


Figure 4.3 FIS Main Page

University of Malaya

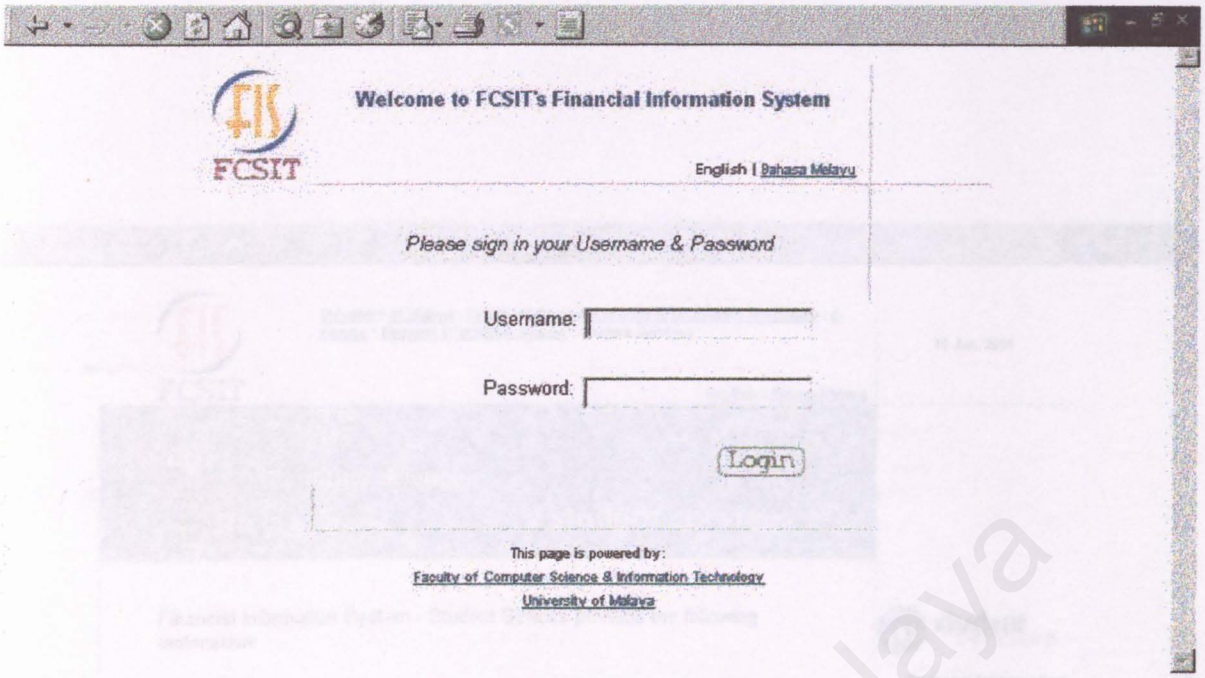


Figure 4.4 FIS Login Page

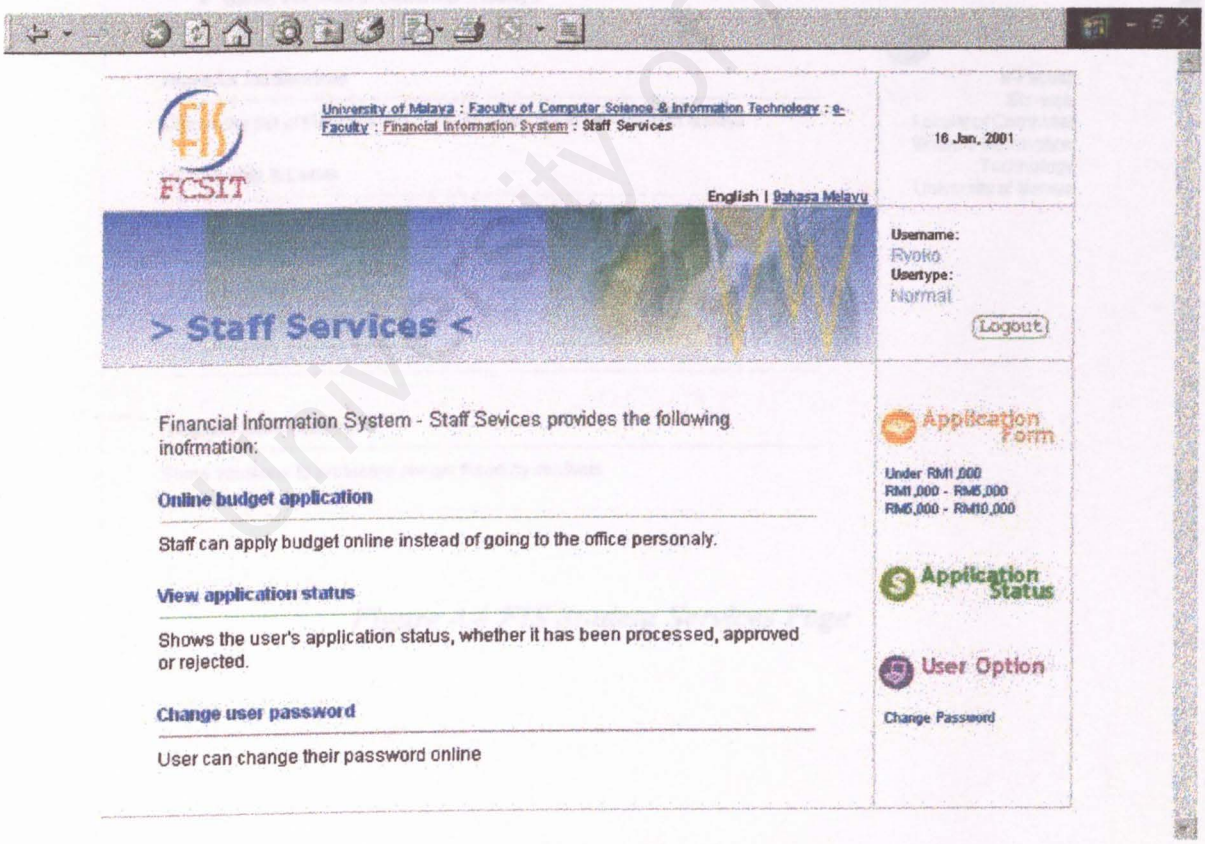


Figure 4.5 FIS Staff Services Page

University of Malaya

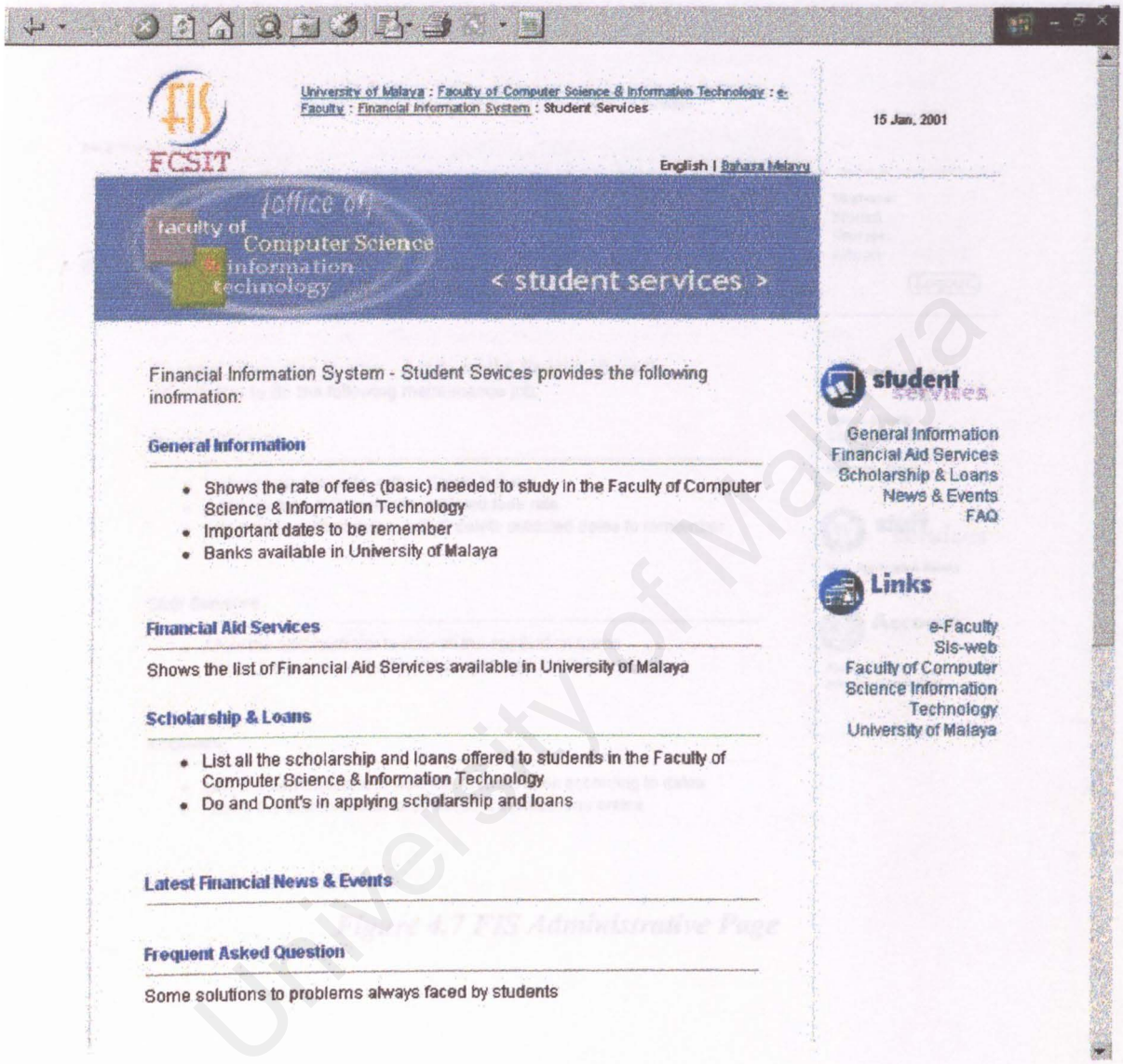


Figure 4.6 FIS Student Services Page

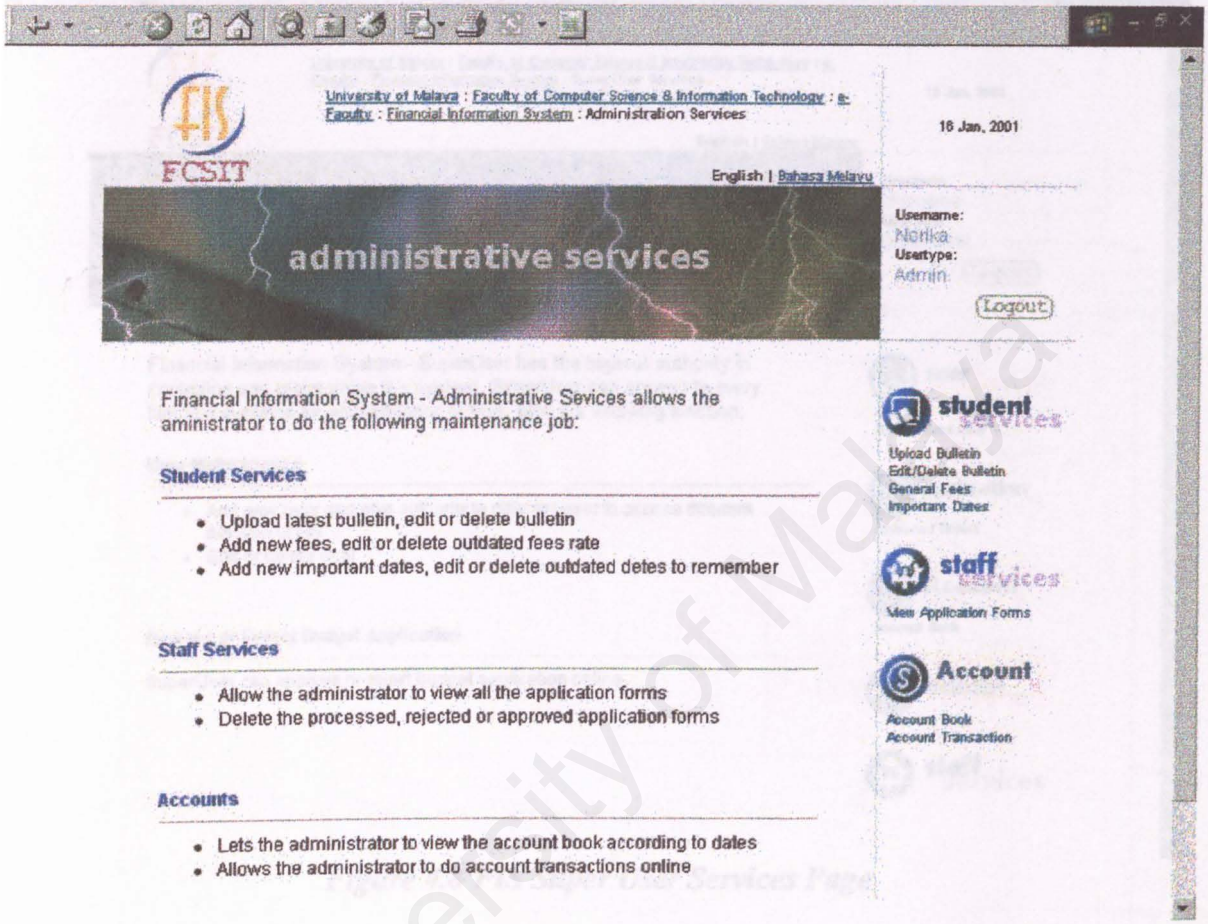


Figure 4.7 FIS Administrative Page

University of Malaya

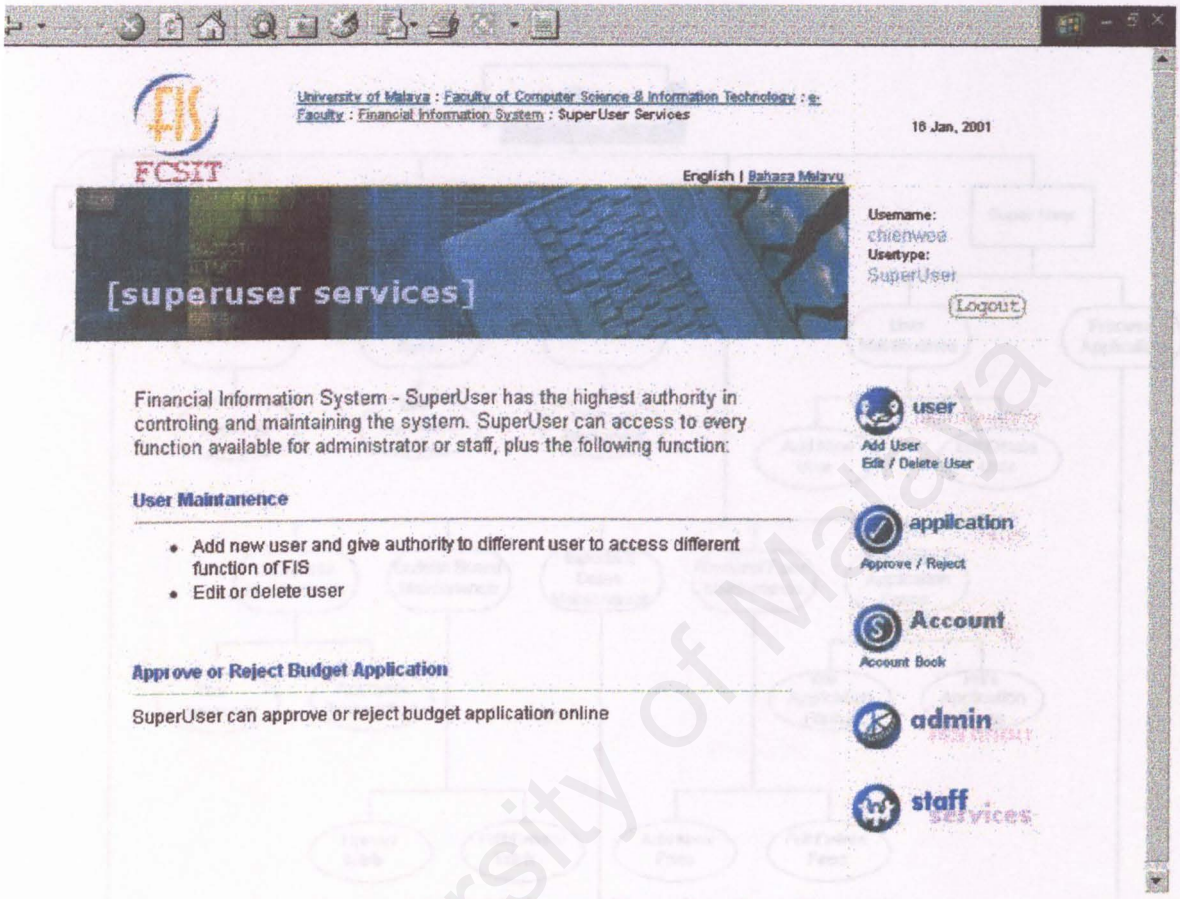


Figure 4.8 FIS Super User Services Page

University of Malaya

4.4 Process Design

4.4.1 System Model of FIS

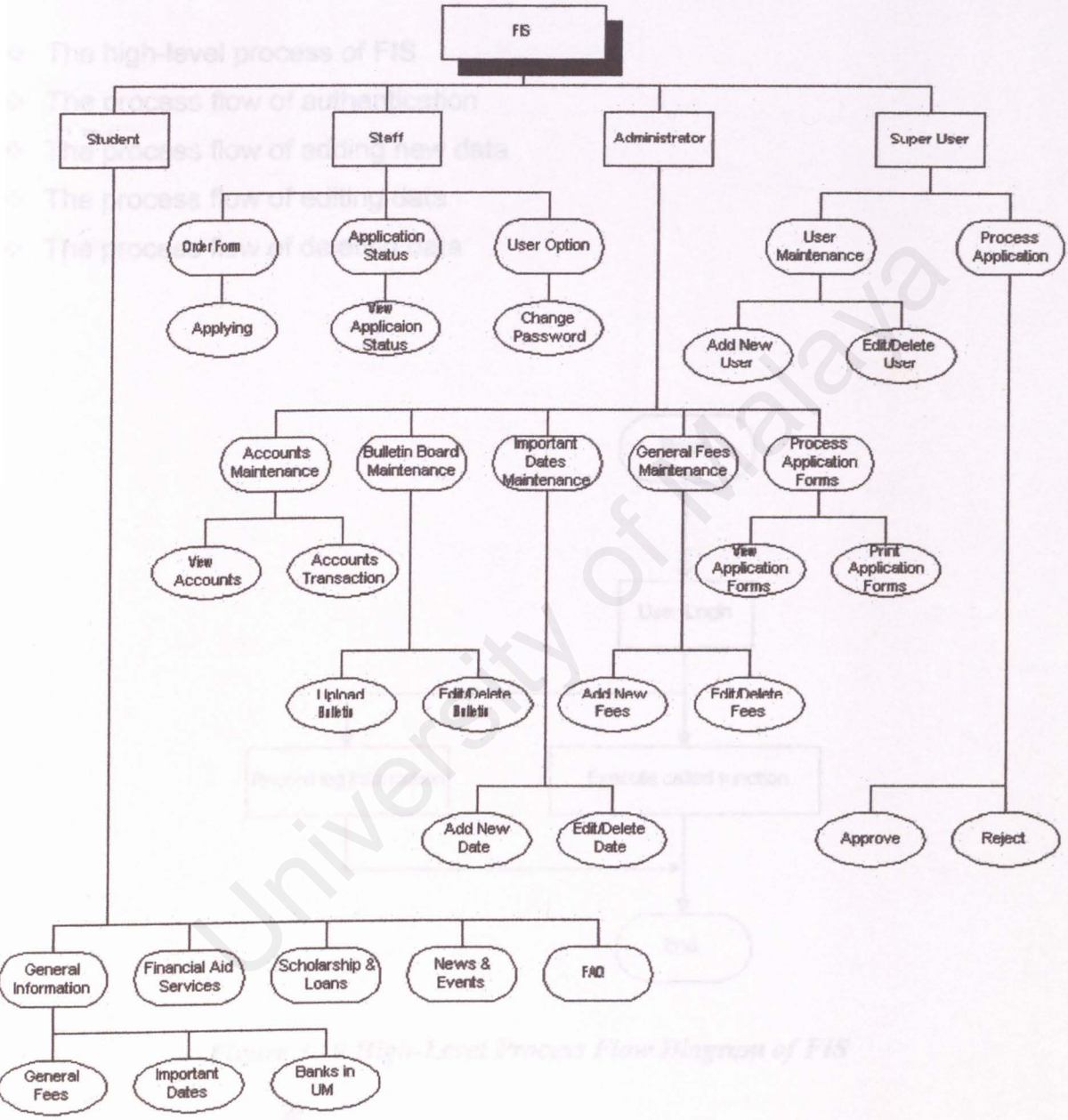


Figure 4.9 The System Model of FIS

The system model is clearly describing the FIS system together with its subsystem.

4.4.2 Process Flow Diagram of FIS

The more detail description of FIS processes is shown in the following process flow diagrams. The following process flow diagrams includes:

- The high-level process of FIS
- The process flow of authentication
- The process flow of adding new data
- The process flow of editing data
- The process flow of deleting data

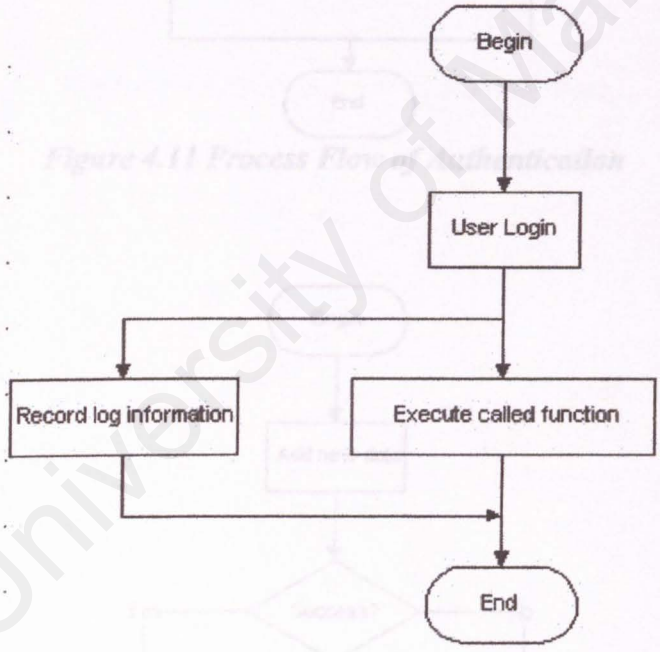


Figure 4.10 High-Level Process Flow Diagram of FIS

University of Malaya

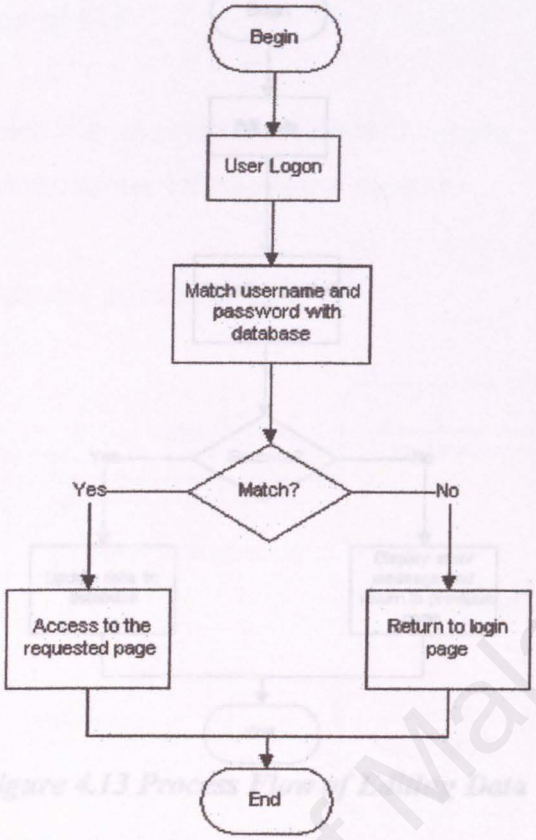


Figure 4.11 Process Flow of Authentication

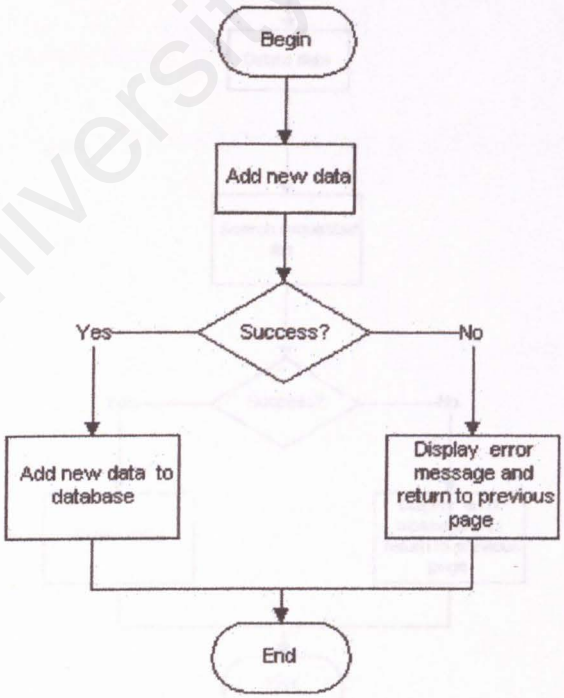


Figure 4.12 Process Flow of Adding New Data

University of Malaya

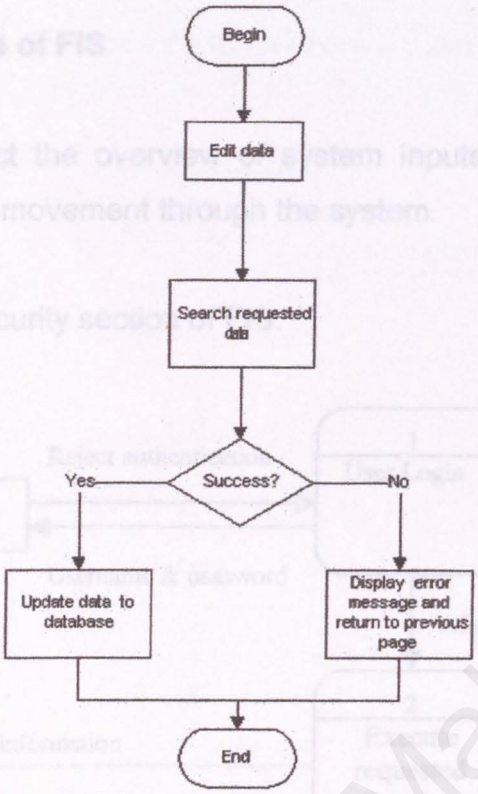


Figure 4.13 Process Flow of Editing Data

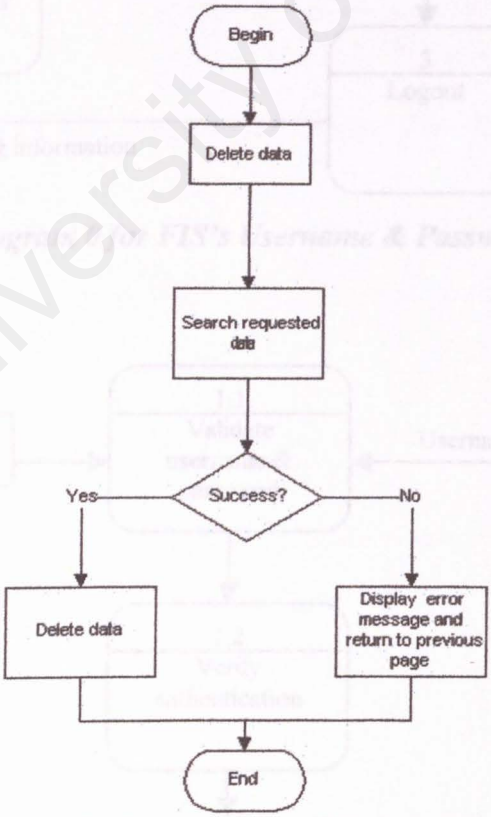


Figure 4.14 Process Flow of Deleting Data

University of Malaya

4.4.3 Data Flow Diagram of FIS

The following DFD depict the overview of system inputs, process, and outputs, which correspond to data movement through the system.

Below are the DFD for security section of FIS.

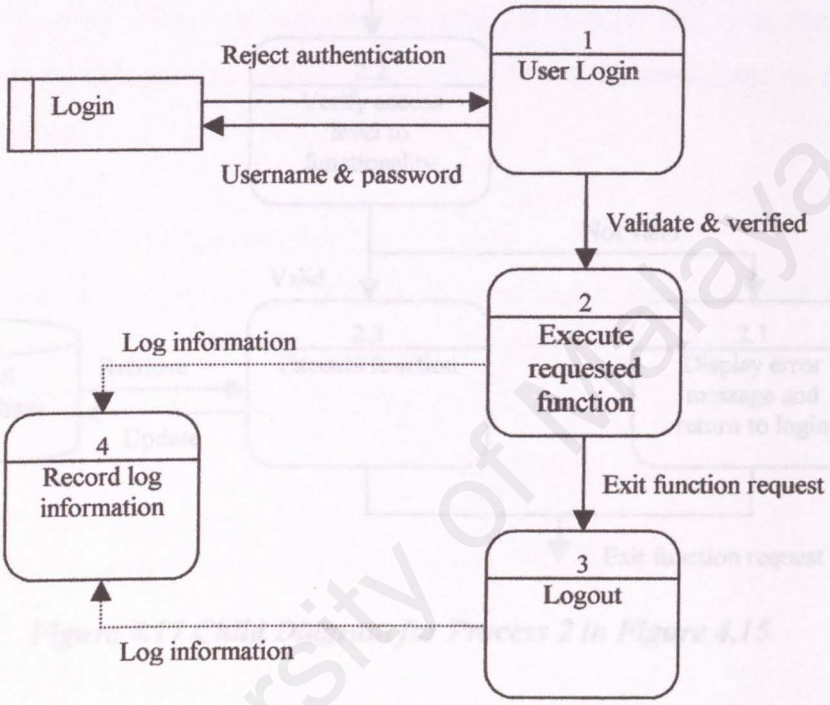


Figure 4.15 Diagram 0 for FIS's Username & Password Validation

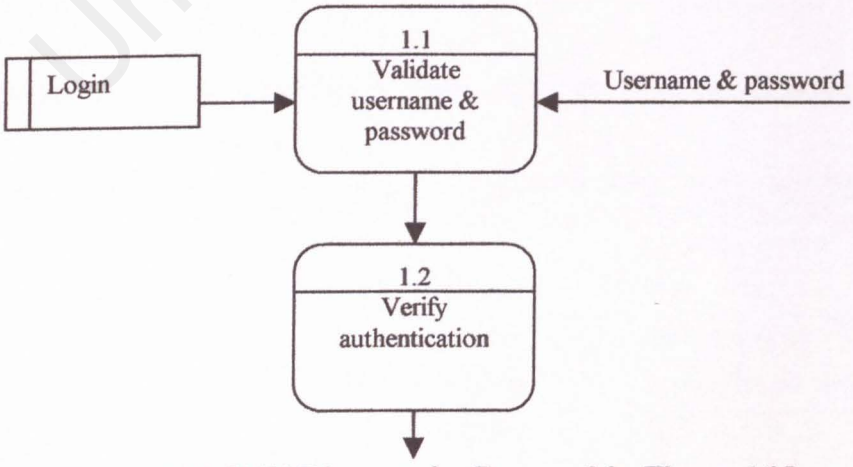


Figure 4.16 Child Diagram for Process 1 in Figure 4.15

University of Malaya

The following are the child diagram for FIS function request.

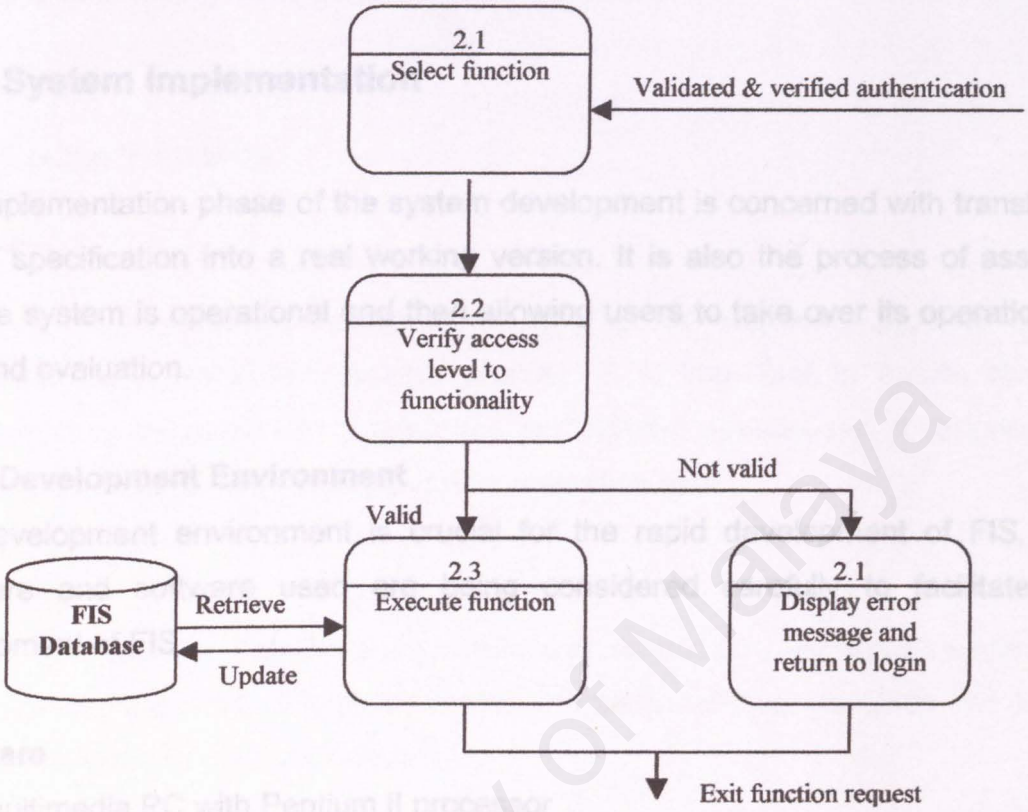


Figure 4.17 Child Diagram for Process 2 in Figure 4.15

University of Malaya

5. System Implementation and Testing

5.1 System Implementation

The implementation phase of the system development is concerned with translating design specification into a real working version. It is also the process of assuring that the system is operational and then allowing users to take over its operation for user and evaluation.

5.1.1 Development Environment

The development environment is crucial for the rapid development of FIS. The hardware and software used are being considered carefully to facilitate the development of FIS.

Hardware

- A multimedia PC with Pentium II processor
- 64MB RAM (or more)
- 2GB of hard disk space (minimum)
- Server with Pentium II processor and 128MB RAM
- Network connection

Software

The table below shows the software tools that are needed directly for system development and implementation.

Software	Module	Description
MS Windows NT Server	System requirement	Operating system
Internet Information Server (IIS)	System requirement	Web server
MS SQL Server	System requirement	Database server
MS Visual InterDev	System development	Application developer
MS Internet Explorer	System development	Web browser

Table 5.1 Tools for System Development

University of Malaya

Other software tools such as: Paint Shop Pro, MS Access, Word, PowerPoint, Visio, FrontPage, Project and Personal Web Server are also been used indirectly in the process of developing the system.

5.1.2 Coding Principles

Reuse

Reuse is an important principle as a method for improving product quality throughout the system development process. It is important to create classes' components to be reused in subsequent and related applications. It reduces the coding time as well as the testing and documentation time.

Readability

Readability is important for the ease of future enhancement. Several strategies are used in preserving readability in the codes, including meaning variables and labels name, comment and proper identification.

Robustness

The system has a robust interface to prevent interface errors from corrupting the system. Beside that, the system also has the ability to checks on the system inputs to ensure correct data is provided in order to protect system integrity.

University of Malaya

5.1.3 Database Development

The database of FIS is first created using MS Access 2000 for convenience and testing purposes. It is migrated to MS SQL Server later for higher performance purpose. The following figure shows the steps taken to create FIS's database in SQL Server.

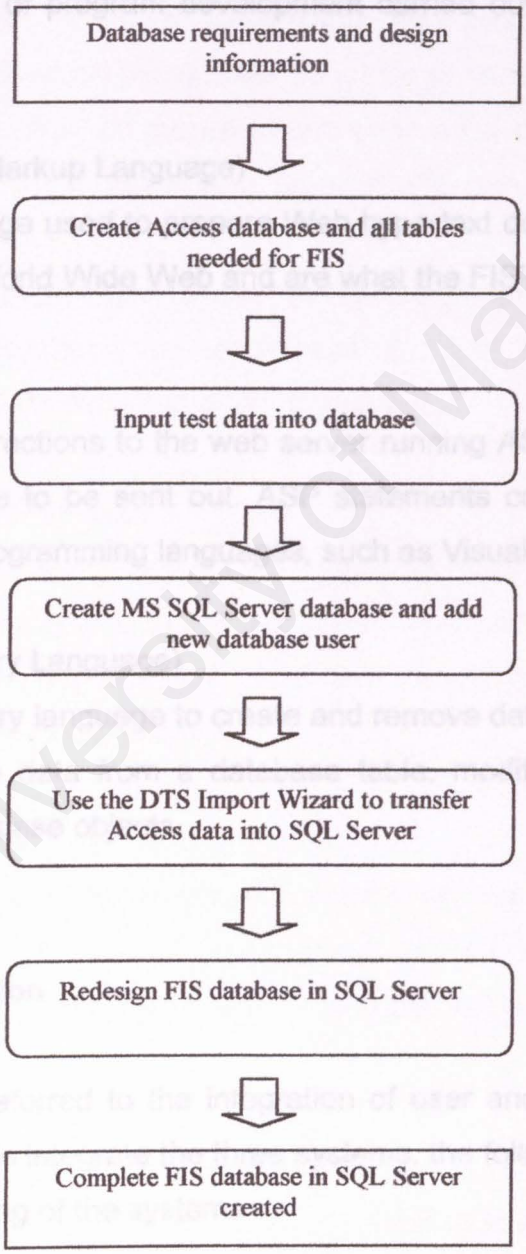


Figure 5.1 Database Setup

University of Malaya

Though the database can be created in SQL Server without creating an Access database first, it is easier to use Access database for testing purposes. Besides, Access database can be easily copy out to other PCs or diskettes.

5.1.4 Application Development

There are three types of program development carried out during developing the FIS application:

1. HTML (HyperText Markup Language)
   HTML is the language used to prepare Web hypertext documents, which will be distributed on the World Wide Web and are what the FIS's client will see.

2. ASP Code
   ASP codes are instructions to the web server running ASP about how to create portions of the page to be sent out. ASP statements contain the structures of more "traditional" programming languages, such as Visual Basic and Java.

3. SQL (Structure Query Language)
   SQL is simply a query language to create and remove database tables and store procedures, retrieve data from a database table, modify the table, and grant permissions on database objects.

5.1.5 System Integration

System integration is referred to the integration of user and FIS, Web application and database. In order to integrate the three systems, the following steps involved:

1. Loading and uploading of the systems
2. Integration of database
3. Integration of application

University of Malaya

5.2 System Testing

System testing is essential to ensure the system performs according to its specifications and in line with users' requirements and expectations. Testing is done throughout system development, not just at the end. It is meant to turn up heretofore-unknown problems, not to demonstrate the perfection of the programs.

Testing is accomplished on FIS's subsystems and program modules as work progresses. Testing is done on many different levels at various intervals. Before the system is put into production, all programs has been desk-check, checked with test data and check to see if the modules work together with one another as planned.

The system as a working whole is also being tested. This includes testing the interfaces between subsystems, the correctness of output, and the usefulness and understandability of system documentation and output.

Program Testing with Test Data

Programs are checked to verify the way the system work. In this stage, each step in the program on paper is followed to check whether the routine works as it is written. Then, the program is tested with both valid and invalid test data to see if base routines work and also to catch errors.

Link Test with Test Data

When programs pass desk checking and checking with test data, they are tested with link testing method, which is also to as string testing. Link testing checks to see if programs that are interdependent actually work together as planned

Full System Testing with Test Data

When link test are satisfactorily concluded, the system as a complete entity must be tested. At this stage, users becomes actively involved in testing. Test data, created for the purpose of testing system objectives, are used.

University of Malaya

Full System Testing with Live Data

After system test using test data is proved satisfactory, the new system is tried with "live data". This step allows accurate comparisons of the new system's output with the correct processed output.

5.3 User Evaluation

The following table shows the users' feedback on the system.

Test	User										Average
	1	2	3	4	5	6	7	8	9	10	
Interface Design	4	4	4	5	3	5	4	4	5	4	4.2
Process Flow	4	4	4	3	3	4	3	4	4	4	3.7
Stability	4	3	4	4	3	4	4	4	3	3	3.6
Accuracy	4	4	3	3	4	3	4	4	3	4	3.6
Speed	3	3	4	3	4	4	3	3	4	3	3.4

Reference: 1 = very poor; 2 = poor; 3 = good; 4 = very good; 5 = excellent

Table 5.2 Users' Feedback

University of Malaya

6. Problems Encountered

This chapter outlines all the obstacles and constraints faced while developing FIS. Solutions were also given to the shortcomings encountered.

6.1 Migrating MS Access Database to MS SQL Server 7.0

The first FIS's database was developed in MS Access 2000 (at home) and has to be migrated to MS SQL Server (at the e-Faculty) for the following reasons:

- More benefits form SQL Server's client/server architecture, compare to Access's file-server environment.
- MS SQL Server can run many processes in parallel.
- MS SQL runs 24 hours a day, 7 days a week.
- MS SQL can store larger database compare to MS Access

However, problems occurred after the migration from Access Database to SQL Server because certain data types in Access 2000 are different from SQL Server and therefore some query syntax that can be used in MS Access are invalid in SQL Server. For example, "Password Text" data type in MS Access is not provided in SQL Server. The following table shows the differences between MS Access and MS SQL Server query syntax.

MS Access	MS SQL Server
ORDER BY in queries	ORDER BY in views not supported
DISTINCTROW	DISTINCT
String concatenation with "&"	String concatenation with "+"
Not Supported: COMPUTE, FOR BROWSE, OPTION	Not Supported: WITH OWNERACCESS, FIRST, LAST
TRANSFORM (SELECT statement) PIVOT	WITH ROLLUP, WITH CUBE on SELECT statements
MAKE TABLE	CREATE TABLE

Table 6.1 Differences Between MS SQL Server and MS Access Query Syntax

Therefore changes in both database and program's code have to be made in order to run the system smoothly in the e-Faculty.

6.2 Setup of Client/Server

As I am not familiar with setup of client/server system and at the same time, the reference on setting up client/server application are not adequate, try and error approach has been carried out to try out the correct way to setup client site and connect to the database.

6.3 Information Gathering

Gathering information about scholarships and loans are difficult, as the Office of Student Affairs doesn't keep an organized list of all the organizations and companies that offer scholarships or loans to students in the Faculty of Computer Science and Information Technology. Therefore I had to do a little survey on students in the faculty to gather more information.

6.4 Unanticipated appearance of web pages during testing

Different browser has different features and capabilities. In the overall development process, Internet Explorer 5.0 (IE5) was used as a viewer. Some JavaScript are not able to execute from Netscape 4.5 browser. IE5 offers the best viewer since all of the ASP code was written in VBScript, which has the Microsoft product compatibility. The compatibility is actually based on the web server that currently used (Personal Web Server). When the client side request a page from ASP, the PWS will process the page and then only produce the output

University of Malaya

into the Web browser. Client will not be able to see the script in server side include due to the VB tag in server side.

6.5 Conclusion

The problems encountered were various. Most of them can be narrowed down to a single cause: lack of knowledge, experience and guidance on developing the system.

University of Malaya

7. System Evaluation

System evaluation is the ending of the report with a look at final results of the project. It outlines all the strengths and limitations of FIS application. It also discusses the future enhancement of the application.

7.1 Strengths of FIS

7.1.1 Real Time Database Information

The information display is real-time and any changes made can be updated to the database immediately.

7.1.2 Organized and User Friendly Interface

FIS's interface is designed to be simple, organize and user friendly. Consistent screen design and color selection also improve the user friendliness of user interface. Besides, user can also easily access to any information needed as sufficient links are provided.

7.1.3 Security

FIS has good security as the authority is designed that each user type can only access to certain functions. Authorized users are given a username and password to prevent unauthorized user from accessing the protected information. In addition, users can always change their password if they feel the password is not secure anymore.

University of Malaya

7.1.4 Error Detection

User input is always validated and verify so that invalid data input will not cause serious program error. Besides, appropriate feedback will also be generated automatically to inform the user if there's an error.

7.1.5 Multi-user

MS IIS and SQL Server provides concurrency control over the FIS application, thus, it fully support multi-user environment without affecting the system integrity.

7.1.6 Cross Browser Support

A lot of the personal web pages that researching would mention which browser the web page author preferred. This will limited the visitor revisit rate. The web page web page can be browsed using both Netscape Communicator and Internet Explorer since both are the most popular browsers in the market currently. The web page is designed not to be dependent on any one of them. Instead, the web page should work well in both browsers.

7.1.7 System Transparency

System transparency refers to the condition where the users do not need to know where the database resides, how is the system structure, its database management system and anything related to the system built.

University of Malaya

7.2 Limitations of FIS

Despite it is a fully functioning system, FIS still has some limitation because of the limitation of the development tools, integration of real environment and time constrain.

7.2.1 Performance Depends on Transmission Lines

The overall performance of FIS is very much depending on the user's connection and transmission line. If the system becomes "slow", it might discourage the user from using it.

7.2.2 Constrain of The Current System Integration

Currently, each department in University of Malaya is acting as an independent entity. Therefore, information sharing and data pulling are not possible because every department would keep their own database from outsiders for security purposes. This constrain will only be solved if the e-Faculty project has been fully developed and implemented.

7.2.3 User Practice

University administrations still depends very much on papers; every accounts transaction or finance transfer needs "black and white" to prove the process validation. FIS users might not feel secure to use the system to do accounts transaction especially if the transaction involves a large amount. Therefore, real paperless office is not likely to be achieved at this stage. Current FIS version doesn't provide budget application over RM1, 000.00. However user still can download and print the order form the web; this method, somehow, helps to reduce paper consumption as the forms are printed out "just-in-time" when they are needed.

University of Malaya

7.3 Future Enhancements

7.3.1 Interactive and Context Sensitive Help

An interactive context sensitive help function will be integrated into FIS in order to provide online helpdesk for user to find the relevant information.

7.3.2 Graphical Reports

Current FIS does not provide graphical graph or chart reports, as the user does not really need it now. However, graphical reports can be added into the system to improve visibility of the system.

7.4 Overall Conclusion

In conclusion, FIS has achieved and fulfilled the objectives and requirements as an on-line financial information provider. The online account transaction provides a secure transaction where only authorized user can access to the function.

This project has been thoroughly meaningful because there was a lot of knowledge gained through development of this system. These include knowledge in client-server systems in the Internet environment, Internet technologies, programming, and database development. Knowledge in ASP, HTML, VBScript and JavaScript proves to be a valuable experience and can be used as a precious skill when searching for a job in future. ASP technology turned out to be the most suitable technology to develop such systems. But, while programming skills and techniques are important in developing the system, it must not be discounted that in any system development, good software engineering techniques must also be applied.

Finally, there's still a lot to be improved in FIS in other to make it a powerful and stable online system.

University of Malaya

References

[1] Walter, Stephen, et al. (1999), *Active Server Pages 2.0 Unleashed*, Sams Publishing.

[2] Buser, D.; Kauffman, J.; T.Llibre, J.; Francis, B.; Sussman, D.; Ullman, C.; and Duckett, Jon, *Beginning Active Server Pages 3.0*, Wrox (1999).

[3] Sommervile, Ian, *Software Engineering 5th Edition*, Addison Wesley (1997, 1998).

[4] *SQL Server 7 Administration Study Guide*, Microsoft (1999).

[5] Kamsah, M.; Mokhtar, S.; and Ahmad, R., (2000), *Towards a model of e-University: A Case of Malaysian Higher Learning Institutions*, University of Malaya.

[6] Hawryszkiewycz, Igor, *Introduction to System Analysis and Design 4th Edition*, Prentice Hall.

[7] Orfali, Robert.; Harkey, Dan.; and Edwards, Jeri., *The Essential Client/Server Survival Guide Second Edition*, Wiley Computing Publishing, John Wiley & sons, INC.

[8] Graham, Ian S., *HTML 4.0 Sourcebook*, Wiley Computing Publishing, John Wiley & sons, INC.

[9] Corey, Michael; Abbey, Michael; Abramson, Ian; Barnes, Larry; Taub, Benjamin; and Venkitavhalam, Rajan, *SQL Server 7 Data Warehousing*, Osborne, McGrae-Hill

University of Malaya

[10] Delaney, Kalen, *Inside Microsoft SQL Server*, Microsoft Press, Microsoft Corporation (1997, 1999).

[11] Hettihewa, Sanjaya, *Windows NT Internet and Intranet Development*, SAM Publishing, Macmillan Computer Publishing.

[12] Craig, Deborah, *JavaScript Manual of Style,* Macmillan Computer Publishing USA (1996).

[13] Reynolds, Mark C.; Daly, Ray; Darnell. Rick; Dr. Dohery, Donald; and Dortch, Bill, *Special Edition Using Java Script*, Que, Macmillan Computer Publishing.

[14] Oliver, Dick, *Teach Yourself HTML 3.2 in 24 Hours*, Sams.net Publishing

[15] Cadenhead, Rogers, *Teach Yourself Java 1.1 Programming in 24 Hours*, Sams.net Publishing.

[16] L. Michael Van Hoozer, Jr., *Teach Yourself Microsoft Visual InterDev in 21 Days*, Sams.net Publishing.

[17] Brophy, Keith; Koets, Timothy, Teach Yourself VBScript in 21 Days, Sams.net Publishing.

[18] Petroutsos, Evangelos; Schongar, Bill; Lagasse, Paul; and Eddy, Craig, *VBScript Unleashed*, Sams.net Publishing.

[19] Accounting Office, University of California at Santa Cruz, *http://www.ucsc.edu/AccountingHomePage.htm* 2000

[20] Campus Integrated Systems, University of Califonia, San Diego, *http://www.act.ucsd.edu/ACT-CampusIntegratedSystems.htm* 2000

University of Malaya

[21] Applying for Financial Aid at OSU, OSU Office of Student Financial Aid, *http://www.act.ucsd.edu/ApplyingforFinancialAidatOSU.htm*

[22] Schofield, Steve, ASP Today,
http://www.asptoday.com

[23] Bahagian Biasiswa, Dana Pendidikan
http://moe.gov.my/ bb.htm

[24] College Costs, University of Wisconsin System
http://www.uwhelp.wisconsin.edu/CollegeCosts.htm

[25] Stanford University Cost Policy, Stanford University,
http://www.co.stanford.edu/CostPolicy-StanfordUniversity.htm

[26] Educational Costs 97-98 Academic Year, UNIVERSITY OF COLORADO,
http://www.uchsc.edu/

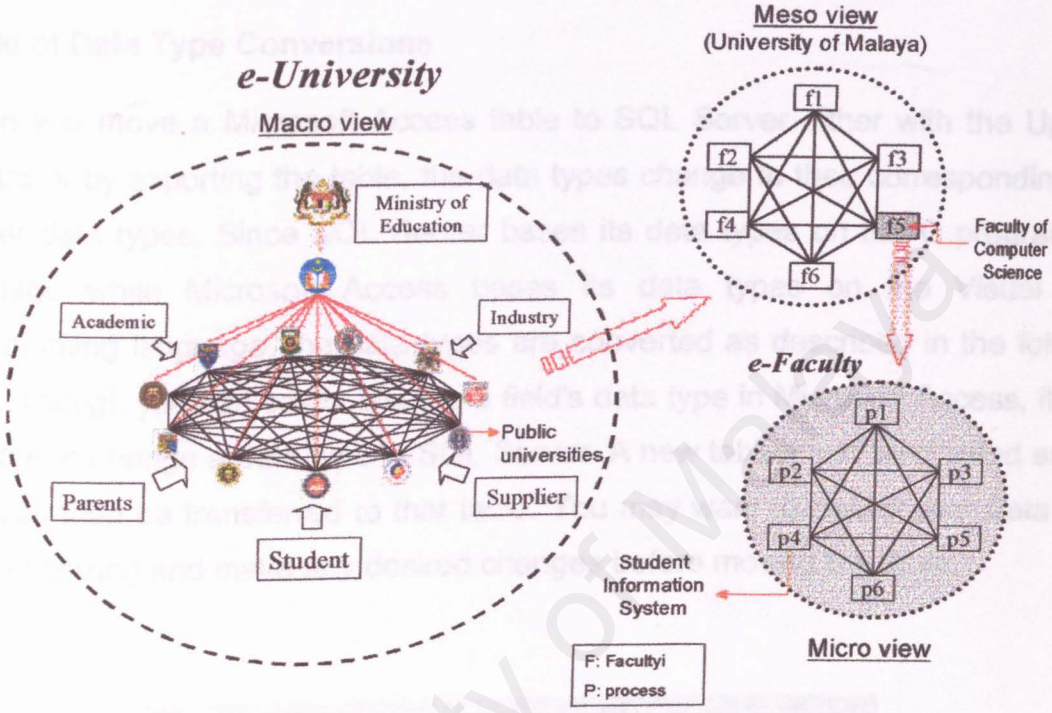
[27] Compare Student Loans and Apply Online, eStudent Loan,
http://www.studentadvantage.com/

[28] Financial Aid, Student Central, University of Texas Austin,
http://www.utexes.edu/

University of Malaya

Appendix A

An overview of e-University



University of Malaya

Appendix B

Migrating MS Access Database to MS SQL Server 7.0

Table of Data Type Conversions

When you move a Microsoft Access table to SQL Server either with the Upsizing Wizard or by exporting the table, the data types change to their corresponding SQL Server data types. Since SQL Server bases its data types on the C programming language while Microsoft Access bases its data types on the Visual Basic programming language, the data types are converted as described in the following table. Though you can easily change a field's data type in Microsoft Access, it is not possible to change a data type in SQL Server. A new table must be created and the data can then be transferred to that table. You may want to review your data types before upsizing and make any desired changes before moving the table.

Microsoft Access	SQL Server
Text	**Varchar ***
Memo	**Text**
Byte	**Smallint**
Integer	**Smallint**
Long Integer	**Int**
Single	**Real**
Double	**Float**
Replication ID	**Varbinary**
Date/Time	**Datetime**
Currency	**Money**
Autonumber (Long Integer)	**Int** (Identity)
Yes/No	**Bit**
OLE Object	**Image**

University of Malaya

The following table shows a SQL Server data type and how Microsoft Access displays it when the table is linked.

SQL Server	Microsoft Access
Binary	Binary
Varbinary	Binary
Char	Text
Varchar	Text
Datetime	Date/Time
Smalldatetime	Date/Time
Decimal	Text
Numeric	Text
Float	Double
Real	Single
Int	Integer
Smallint	Integer
Tinyint	Integer
Identity	Autonumber
Money	Currency
Smallmoney	Currency
Bit	Yes/No
Timestamp	Binary
Text	Memo
Image	OLE Object
User-defined data types	Varies

University of Malaya

Expression Translation Table

When you add defaults to SQL Server with the SQL Server Browser, the SQL Server Browser converts most common Microsoft Access expressions to SQL Server expressions. The following expressions are converted:

Microsoft Access functions	SQL Server functions
String functions	
chr$(x)	char(x)
asc(x)	ascii(x)
str$(x)	str(x)
space$(x)	space(x)
lcase$(x)	lower(x)
ucase$(x)	upper(x)
len(x)	datalength(x)
ltrim$(x)	ltrim(x)
rtrim$(x)	rtrim(x)
right$(x,y)	right(x,y)
mid$(x,y,z)	substring(x,y,z)
Conversion functions	
cint(x)	convert(smallint,x)
clng(x)	convert(int,x)
csng(x)	convert(real,x)
cdbl(x)	convert(float,x)
cstr(x)	convert(varchar,x)
ccur(x)	convert(money,x)
cvdate(x)	convert(datetime,x)
Date functions	
now(x)	getdate(x)
date(x)	convert(datetime,convert(varchar,getdate(x)))

University of Malaya

year(x)	datepart(yy,x)
month(x)	datepart(mm,x)
day(x)	datepart(dd,x)
weekday(x)	datepart(dw,x)
hour(x)	datepart(hh,x)
minute(x)	datepart(mi,x)
second(x)	datepart(ss,x)
datepart("<Access datepart>", x)	datepart(<SQL Server datepart>, x)
dateadd("<Access datepart>", x, y)	dateadd(<SQL Server datepart>, x, y)
datediff("<Access datepart>", x, y)	datediff(<SQL Server datepart>, x, y)
Math functions	
int(x)	floor(x)
sgn(x)	sign(x)

The Upsizing Wizard and the SQL Server Browser replace a number of delimiters, operators, constants, and wildcard characters, as listed in the following table.

Description	Microsoft Access	SQL Server
Date delimiter	#	'
String delimiter	"	'
Mod operator	mod	%
Concatenation operator	&	+
Wildcard character	?	_
Wildcard character	*	%
Constant	Yes	1
Constant	On	1
Constant	True	1
Constant	No	0
Constant	Off	0
Constant	False	0

Appendix C

Order Forms Examples

FAKULTI SAINS KOMPUTER & TEKNOLOGY MAKLUMAT
UNIVERSITY MALAYA

BORANG PESANAN

Saya/Kami memerlukan perkara beriktu:

Jenis Peralatan: _____ Bilangan: _____

Sebab-sebab diperlukan: _____

_____

Nama Pemohon: _____

Tandatangan: _____ Tarikh: _____

Untuk Kegunaan Penolong Pendaftar:

Permohonan diperakukan kerana: _____

_____

Permohonan tidak diperakukan kerana: _____

_____

Tarikh: _____ _____

Tandatangan

Untuk Kegunaan Dekan:

Permohonan diluluskan Permohonan tidak diluluskan

Tarikh: _____ _____

Tandatangan

University of Malaya

FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT
Syor Setuju Terima Sebutharga (Bagi Pembelian Di antara RM1,000 – RM5,000)

Kegunaan Pembelian :

Peruntukan :

Bil	Penender	Perkara	Kuantiti	Harga 1 Unit (RM)	Jumlah (RM)	Catitan
1.						
2.						
3.						

Kajian Dan Cadangan Pemilihan :

Nama & Tandatangan Pemohon : Tarikh:

Ulasan Dan Perakuan :

Peruntukan :

Tandatangan Penolong Pendaftar: Tarikh:

Kelulusan Setuju Terima :

Tandatangan Dekan : Tarikh:

s.k. Pejabat Bendahari (U.P: Pegawai yang bertanggungjawab atas bayaran)

Nota: 1) Bagi bekalan lebih dari satu perkara, sila gunakan Lampiran 1
2) Borang ini hendaklah dilampirkan bersama dengan borang pesanan salinan Bendahari (sebutharga dari pembekal tidak perlu disertakan) kepada Bendahari

YW/Nurul (sebutharga RM1000-RM5000)

University of Malaya

FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT
Syor Setuju Terima Sebutharga (Bagi Pembelian Di antara RM1,000 – RM5,000)

Kegunaan Pembelian :

Peruntukan :

Bil	Penender	Perkara	Kuantiti	Harga 1 Unit (RM)	Jumlah (RM)	Catitan
1.						
2.						
3.						
4.						
5.						

Kajian Dan Cadangan Pemilihan :

Nama & Tandatangan Pemohon : Tarikh:

Ulasan Dan Perakuan :

Peruntukan :

Tandatangan Penolong Pendaftar: Tarikh:

s.k. Pejabat Bendahari (U.P: Pegawai yang bertanggungjawab atas bayaran)

Nota: 1) Bagi bekalan lebih dari satu perkara, sila gunakan Lampiran 1
2) Borang ini hendaklah dilampirkan bersama dengan borang pesanan salinan Bendahari (sebutharga dari pembekal tidak perlu disertakan) kepada Bendahari

YM/Nurul (sebutharga <RM5000)

University of Malaya

FIS User Manual

University of Malaya

Contents

University of Malaya

University of Malaya

FCSIT

Chapter 1 Before You Begin

Welcome

Welcome to Financial Information System (FIS). This system is specially designed to computerize financial information system of the Faculty of Computer Science and Information. FIS is a web base application with user-friendly interface, meaningful icons and links that guide user in using this system. All the system functions can be easily executed by a simple click on the available buttons or icons.

About This Manual

This manual will introduce you to FIS and provide all information you need to use the system.

Who Should Use This Manual

This manual will intended for the following categories of users:

- Staff
  Normal user (faculty's staff) that would want to apply for budget or order needed item online.
- Administrator
  User that will maintain student's financial information data, faculty's accounts, process normal user's order forms, and approve or reject normal user's order form.
- Super User
  User who has the authority to create new user, maintain existing user's data, and extra authority to access to any function other user can.

University of Malaya

What's in This Manual

Chapter 1

"Before You Begin" provides an overview of FIS. This chapter also defines the categories of user for this manual and explains the conventions used throughout this manual.

Chapter 2

"System Requirements" tells you what you need to runs the system.

Chapter 3

"Getting Started" describes how to start FIS and login to the system.

Chapter 4

"Staff Section" shows normal user the way to apply budget or order item needed with FIS.

Chapter 5

"Administrator Section" provides guidelines for database administrator on how to maintain FIS database and accounts, as well as processing user's application forms, and approve or reject user's application forms.

Chapter 6

"Super User Section" provides guidance for super user on how to maintain user data.

University of Malaya

Chapter 2 System Requirement

System Requirements

Hardware Requirement

FIS has the following hardware requirement and recommendations:

- A multimedia PC with Pentium processor or above
- Minimum 16MB RAM
- Internet or e-Faculty Intranet connection
- Other common computer peripherals

Software Requirement

The following software is needed to run FIS:

- Windows NT or Windows 95 platforms
- Web browser – IE5.0 or Netscape 4.5 and above are recommended

University of Malaya

Chapter 3 Getting Started

FIS is a web application, you need to open your browser by clicking the browser's icon, before you can access to the system.

Starting FIS

Type the URL address of FIS in the browser's address bar and press <Enter>. Make sure your PC is connected to the Internet or you are using a network PC within e-Faculty's intranet. A successful connection will display the main page of FIS. Within this page, you can read some general information, such as the faculty's financial news and events, student services information or login directly to the system.

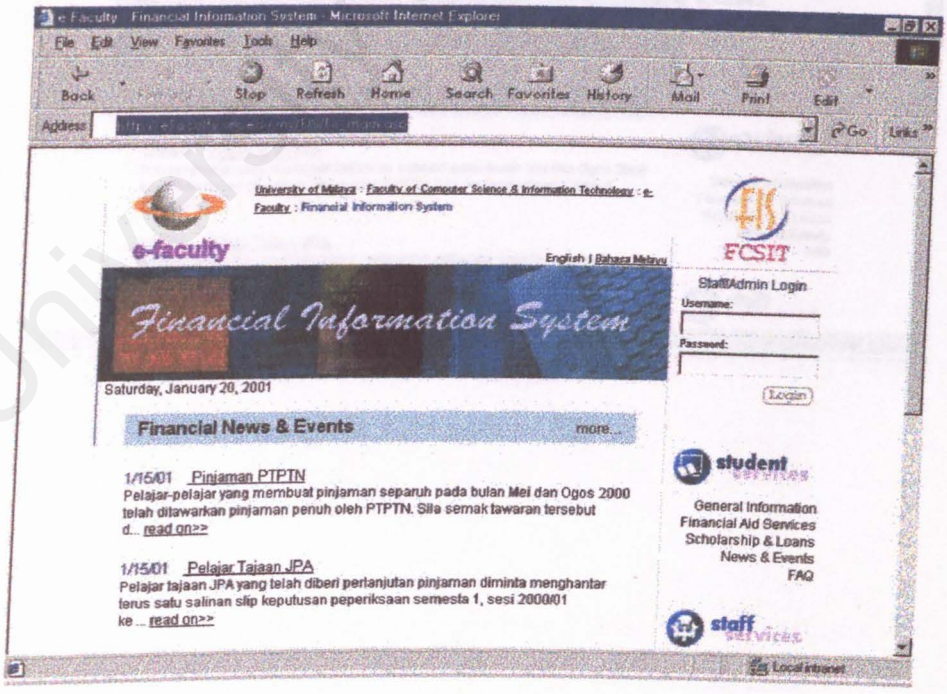


Figure 3.1 FIS Start Page

Login to FIS

To login to FIS, you must have a valid username and password. If you don't have one now, please refer to the system administrator to create one for you. Simply type in your username and password in the text box at the upper right of the screen and click on the "Login" button. Take note that the password is case sensitive.

If you are successfully login to the system, a menu page will be display where you can choose to navigate to where you are authorized.

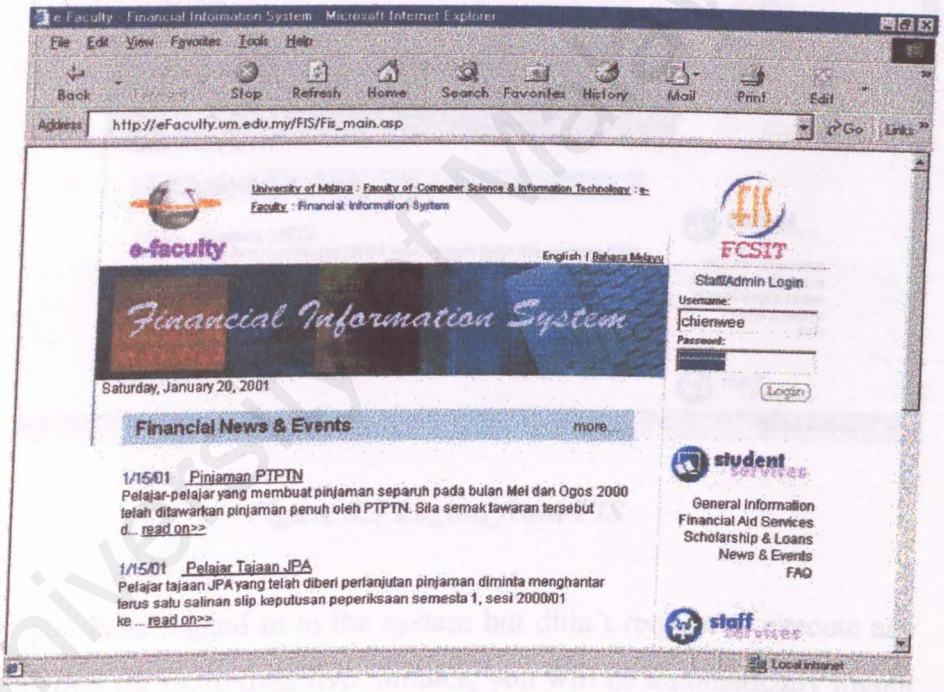


Figure 3.2 Login to FIS

Logout FIS

User can always logout by clicking the <Logout> button at the upper right corner of the page. To prevent other users from using your account, you are adviced to logout from the system every when you want to quit using FIS.

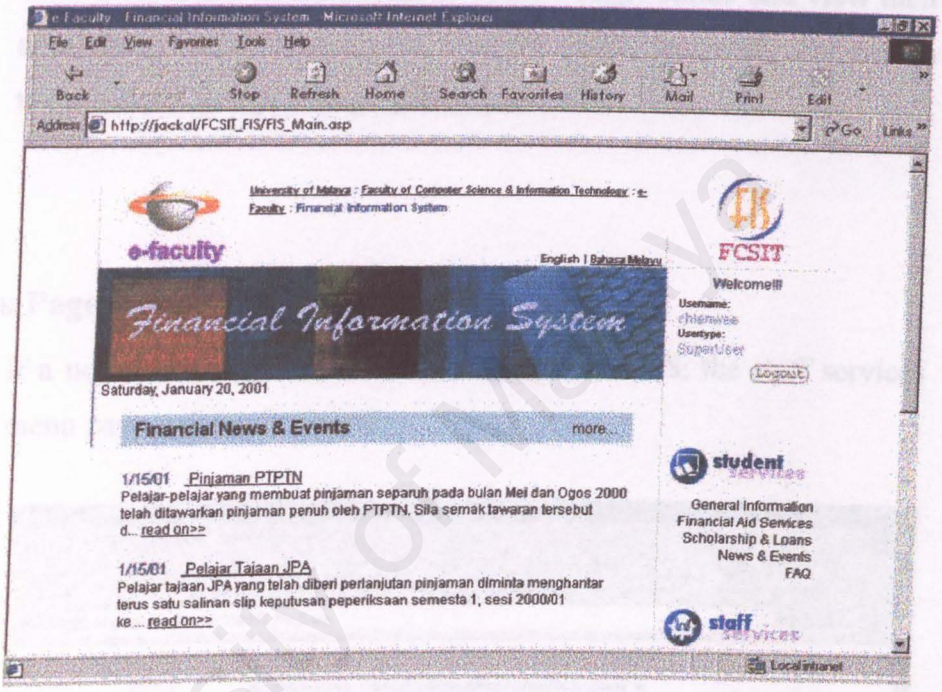


Figure 3.3 Logout from FIS

If you have logged in to the system but didn't request to execute any function for more than five minutes, you will be automatically logout from the system.

Quit

To quit completely from FIS just close the browser by clicking the "X" sign at the upper right of the window, or select the **Exit** function from the **File** menu.

University of Malaya

Chapter 4 Staff Section

Every registered staff in the Faculty of Computer Science and Information Technology are normal user of FIS. The normal users are users who are only authorized to apply budget online and view their application status. They do not have the ability to modify any data in the database, except changing their password.

Staff Services Menu Page

If a normal user has successfully logged in to FIS, the staff services menu page will be displayed.

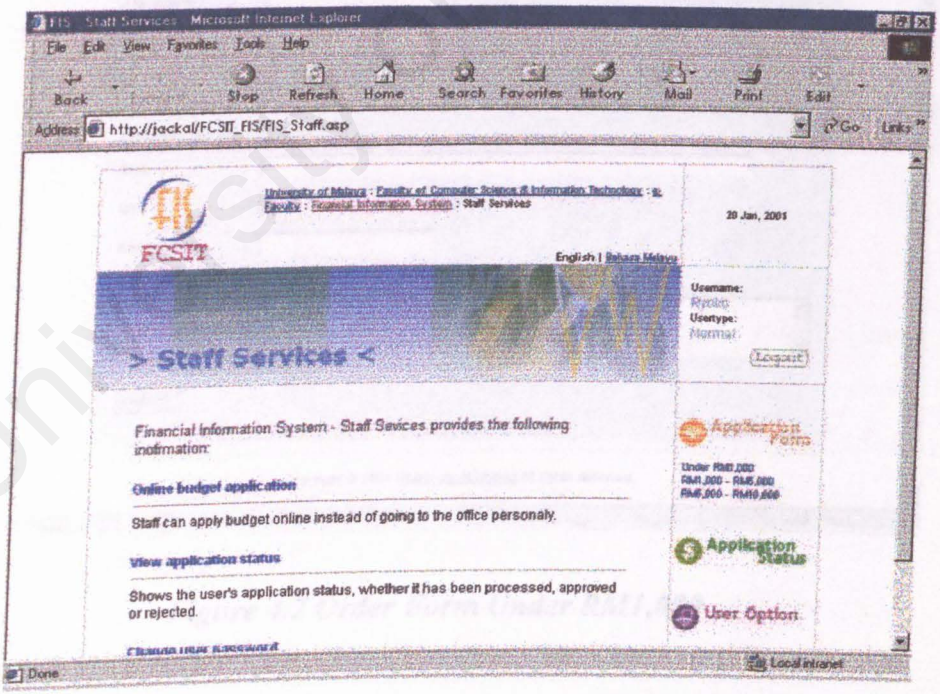


Figure 4.1 Staff Services Page

Online Budget Application

To apply budget under RM1,000, click the Under RM1,000 link and you will see the following page. Fill in all the needed details in the text boxes and click the <Submit> button. If the submission is success, a summary of the order form will be shown; else an error message will appear directing the user to fill in the form correctly.

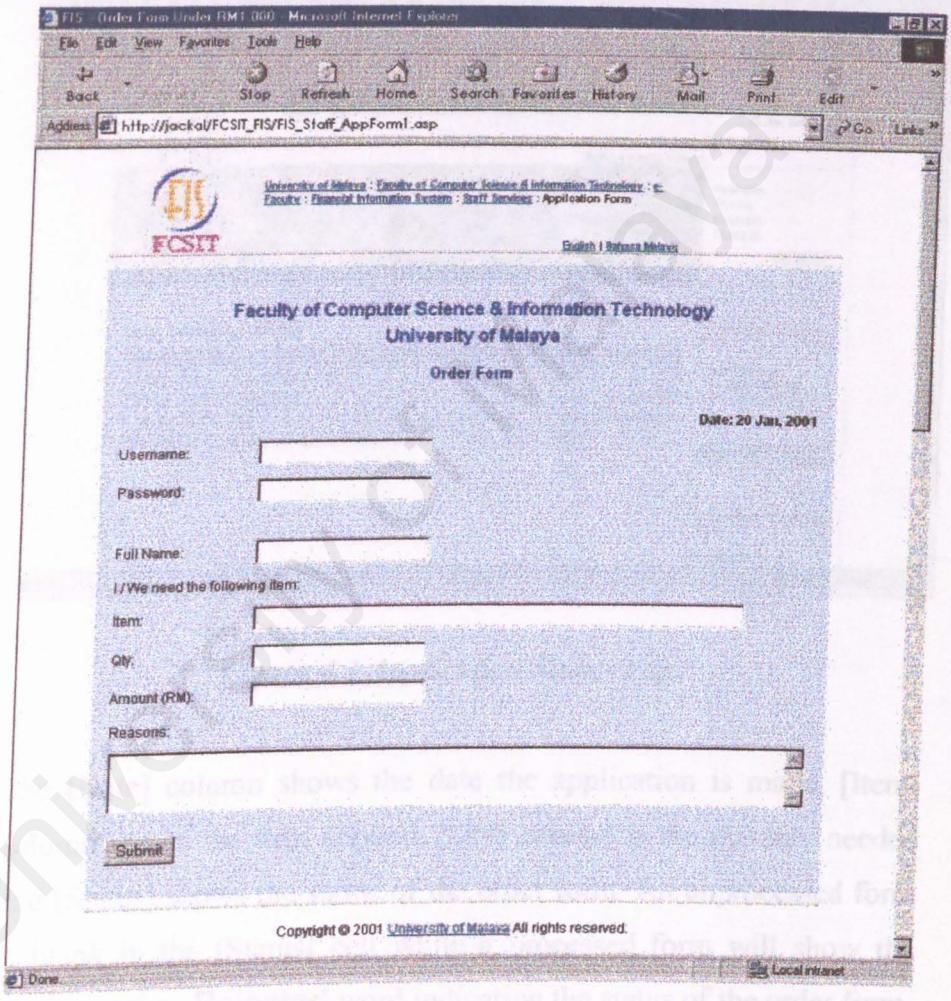


Figure 4.2 Order Form Under RM1,000

Currently, online applications for budget over RM1,000 are not allow, however users can download and print the order forms the from Staff Services page by clicking the RM1,000-5,000 or RM5,000-10,000 link.

View Application Status

To check whether your applications have been processed, click on the <Application Status> icon in the main menu page and the following page will be shown.

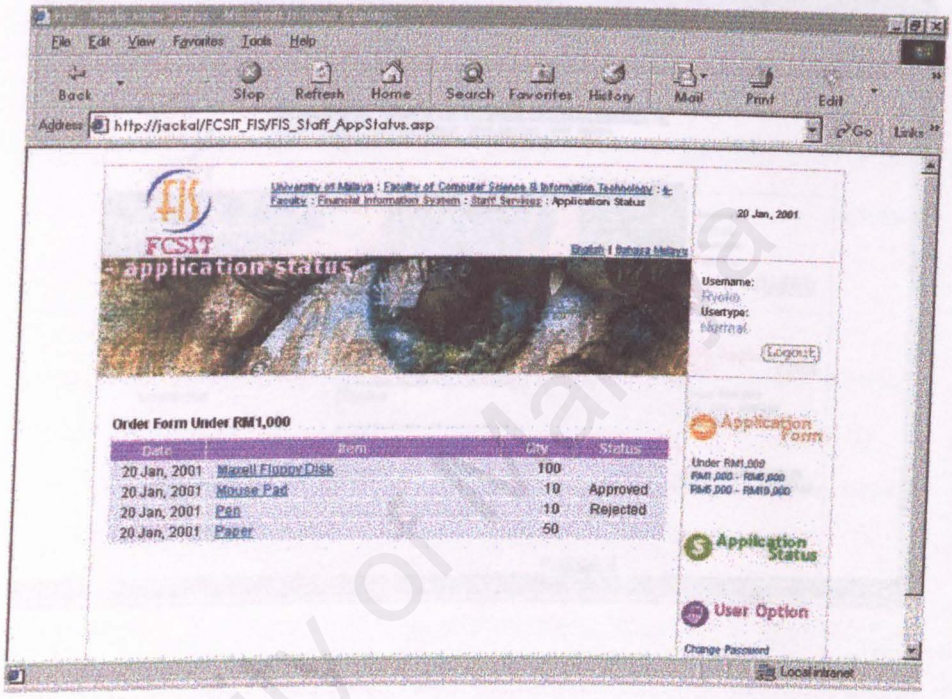


Figure 4.3 Application Status Page

The [Date] column shows the date the application is made, [Item] column shows the item applied, [Qty] column is the quantity needed and [Status] shows the status of the order form. An unprocessed form is blank in the [Status] cell while a processed form will show the **Approved** or **Rejected** word indicating the status of the order form. To view the order form in detail, click on the item link and a new page showing the details of the order form will be display.

University of Malaya

Change Password

User can always change his password by choosing Change Password link under the User Option icon from the Staff Services menu page.

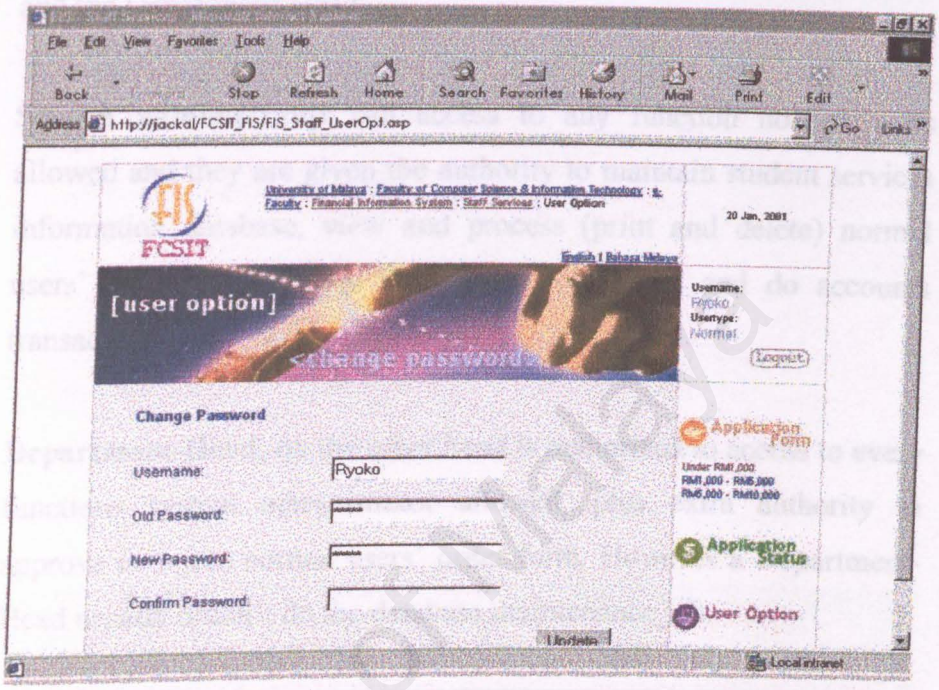


Figure 4.4 Change Password

As for security purpose, users are required to key in their old password before they can change their password. All users are advised to set a password with at least 6 characters with combination of characters, digit and symbols. After inputting the new password, reenter the password in the **Confirm Password** text box then press <Enter> or click on the <Update> button and a summary page will be display.

Chapter 5 Administrator Section

There are two levels of administrator in FIS, the system administrator and the Department-Head.

System administrator can access to any function normal users allowed and they are given the authority to maintain student services information database, view and process (print and delete) normal users' order forms, view the faculty's accounts and do accounts transaction.

Department-Head, on the other hand is authorized to access to every functions system administrator allowed, plus extra authority to approve or reject normal users' order form. However a Department-Head usually doesn't do the database maintenance job.

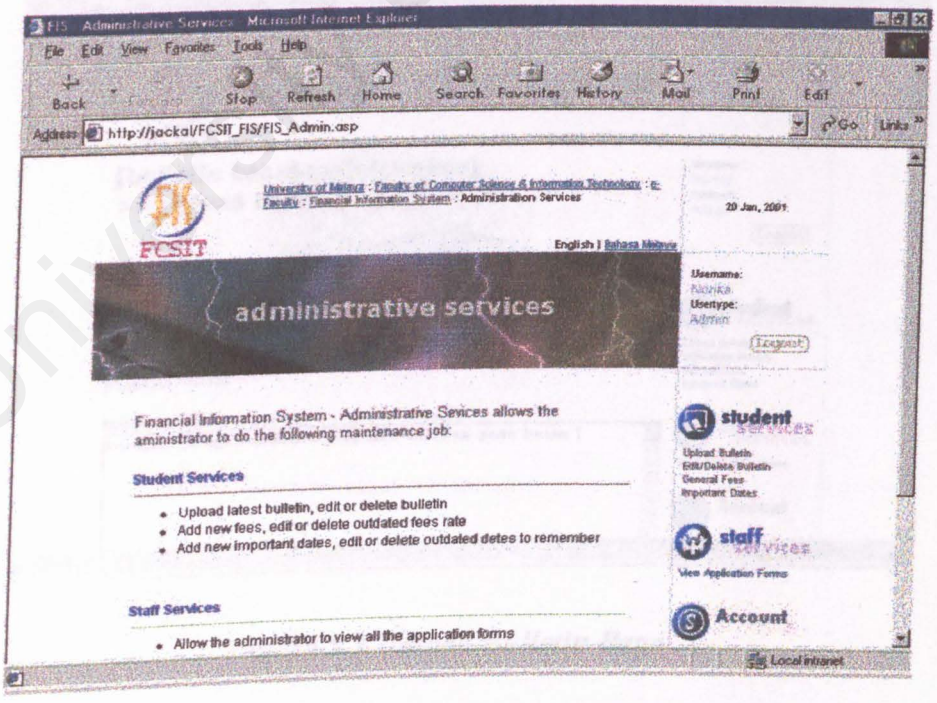


Figure 5.1 Administrative Services Menu Page

Student Services Information Maintenance

From the Administrative Services menu page, system administrator can choose to:

- Upload financial bulletin
- Edit/Delete financial bulletin
- Maintain general fees
- Maintain important dates

Uploading Financial Bulletin

By clicking the Upload Bulletin link the following page will be display.

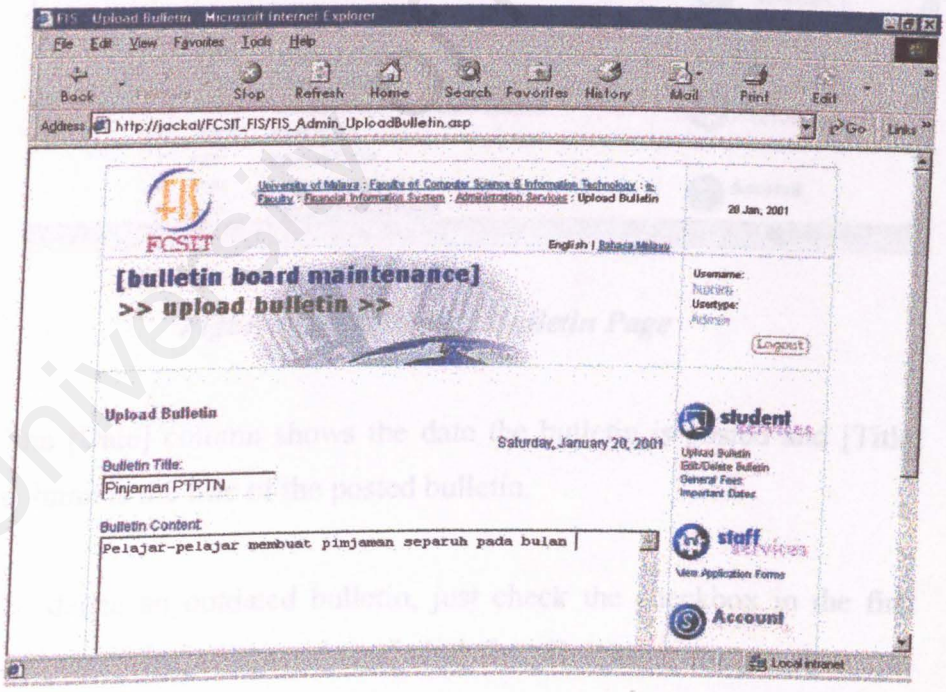


Figure 5.2 Upload Bulletin Page

Key in the bulletin title in the **Bulletin Title** text box and bulletin content in the **Bulletin Content** text area then click on the <Upload

Bulletin> button and a new bulletin is uploaded. A summary of the uploaded bulletin will then be display.

Edit/Delete Bulletin

To edit or delete a posted bulletin, click on the Edit/Delete Bulletin link.

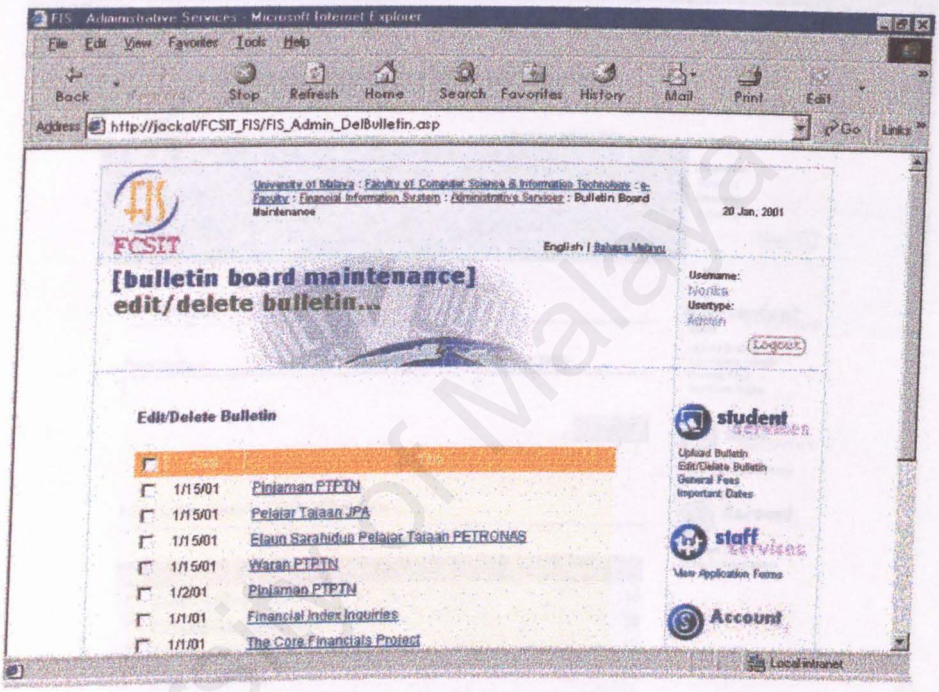


Figure 5.3 Edit/Delete Bulletin Page

The [Date] column shows the date the bulletin is posted and [Title] column is the title of the posted bulletin.

To delete an outdated bulletin, just check the checkbox in the first column of the bulletin title and click the <Delete> button.

To edit a bulletin click on the bulletin title link and a new page will appeared allowing system administrator to edit and update bulletin. Basically the "Edit Bulletin" page has the same functions as the "Upload Bulletin" page.

Maintaining General Fees Information

To maintain the general fees information, click on the General Fees link. The following page will be display.

Figure 5.4 General Fees Maintenance Page

To add new general fees just key in the fee description in the **Description** text box and Amount (RM) in the **Amount** text box, then press the <Upload> button.

To edit a general fee information, click on the fee description link and change the corresponding data and press the <Update> button. To delete outdated general fees, check the checkbox at the left of the "Edit/Delete General Fees Information" table and click on the <Delete> button at the bottom of the table.

Maintaining Important Dates Information

Click on the Important Dates link in the Administrator Services page to show the following page.

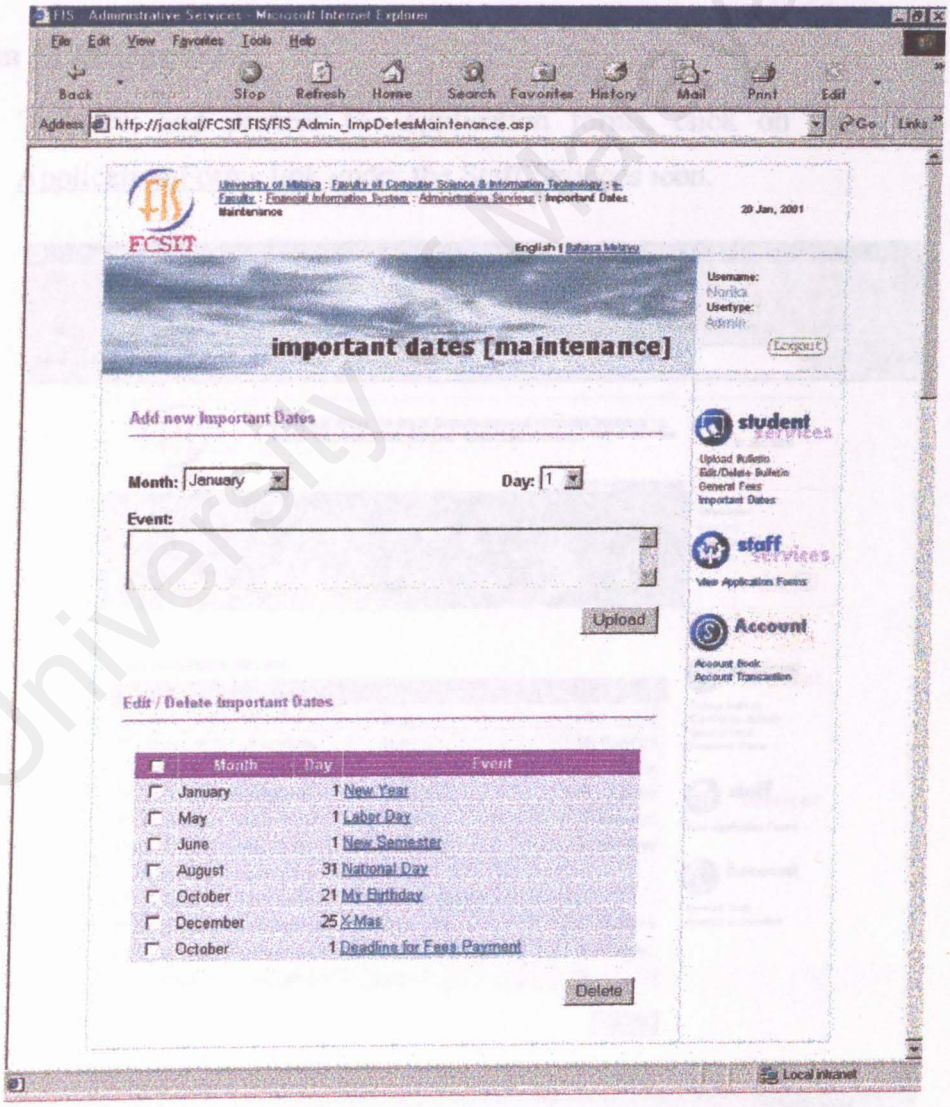


Figure 5.5 Important Dates Maintenance Page

To add new important date select the month and day of the date from the **Month** and **Day** menu then type in the event in the **Event** text area and click on the <Upload> button.

To edit a posted important data, just click on the date event link in the "Edit/Delete Important Dates" table. Change what data needed and click the <Save> button. To delete posted important dates check the checkbox in the table and press the <Delete> button at the bottom of the table.

Staff's Order Form Maintenance

To view and process the application forms, click on the View Application Forms link under the Staff Services icon.

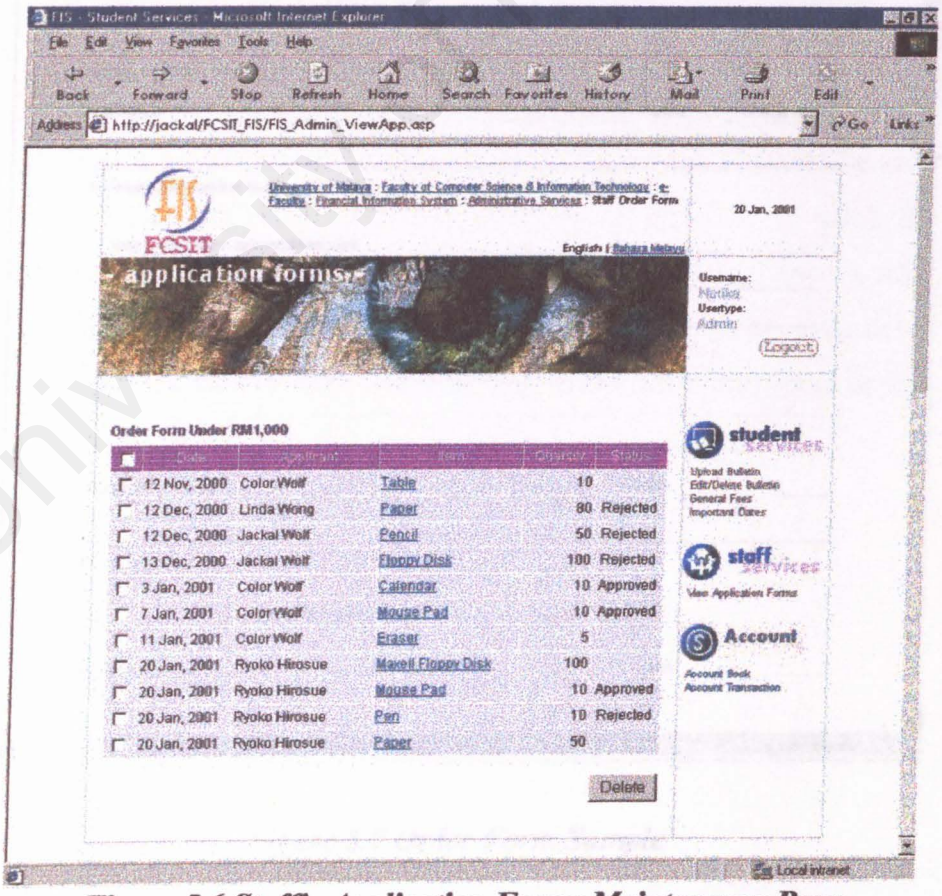


Figure 5.6 Staff's Application Forms Maintenance Page

Print Order Form

To view and print an order form just click on the item link in the "Order Form Under RM1,000" table and the following page will be display.

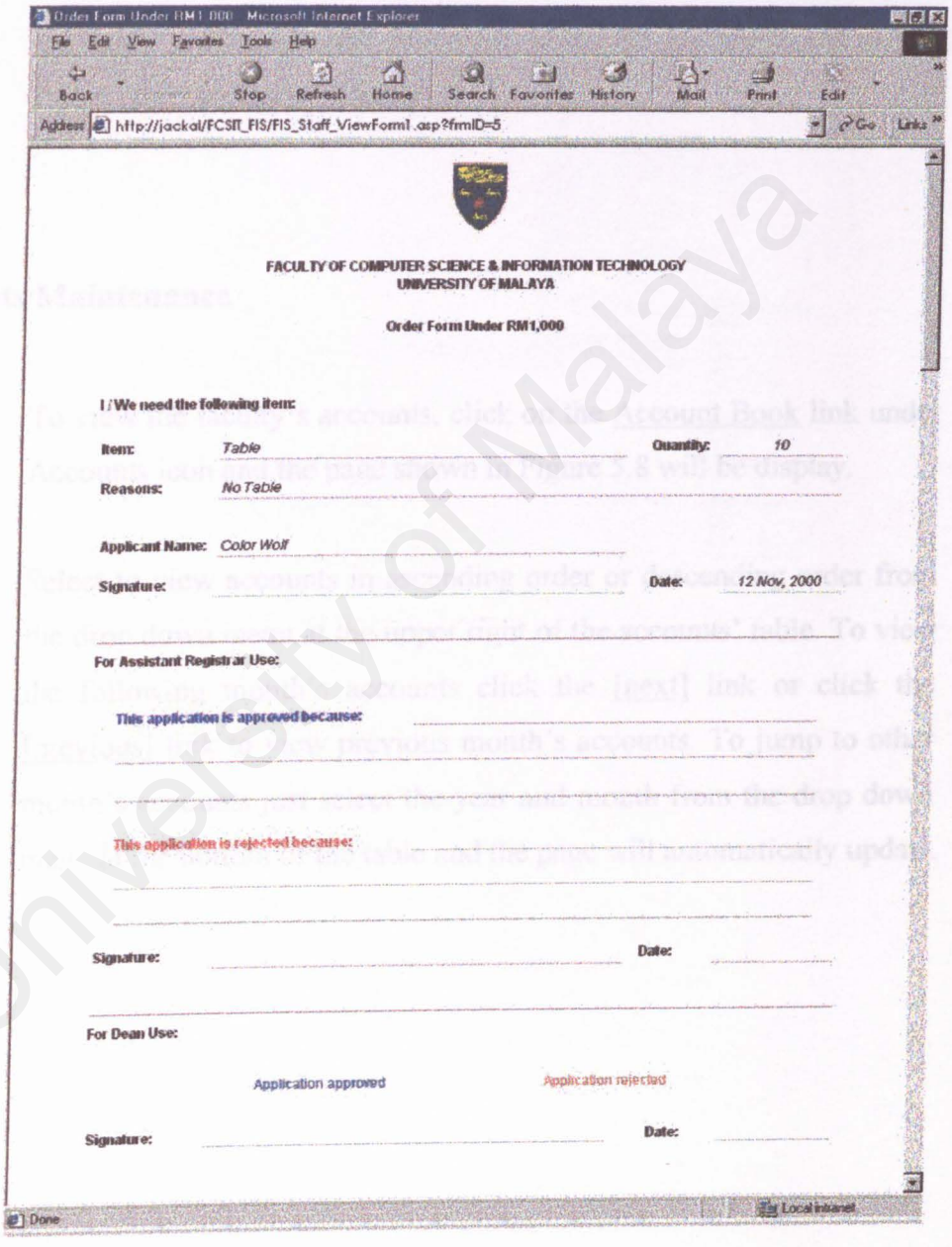


Figure 5.7 Order Form Sample

University of Malaya

To print out the order form select the **Page Setup** function from the **File** menu and delete the "&w&bPage &p of &P" and "&u&b&d" words in the **Header** and **Footer** text box and press <OK>. Then click on the Printer icon on the toolbar or select **Print** function from the **File** menu. Make sure your printer connection is ready.

To delete processed order forms just check the checkboxes at the left of the table and press <Delete>.

Faculty's Accounts Maintenance

To view the faculty's accounts, click on the Account Book link under Accounts icon and the page shown in Figure 5.8 will be display.

Select to view accounts in ascending order or descending order from the drop down menu at the upper right of the accounts' table. To view the following month's accounts click the [next] link or click the [previous] link to view previous month's accounts. To jump to other month's accounts just select the year and month from the drop down menu at the bottom of the table and the page will automatically update.

University of Malaya

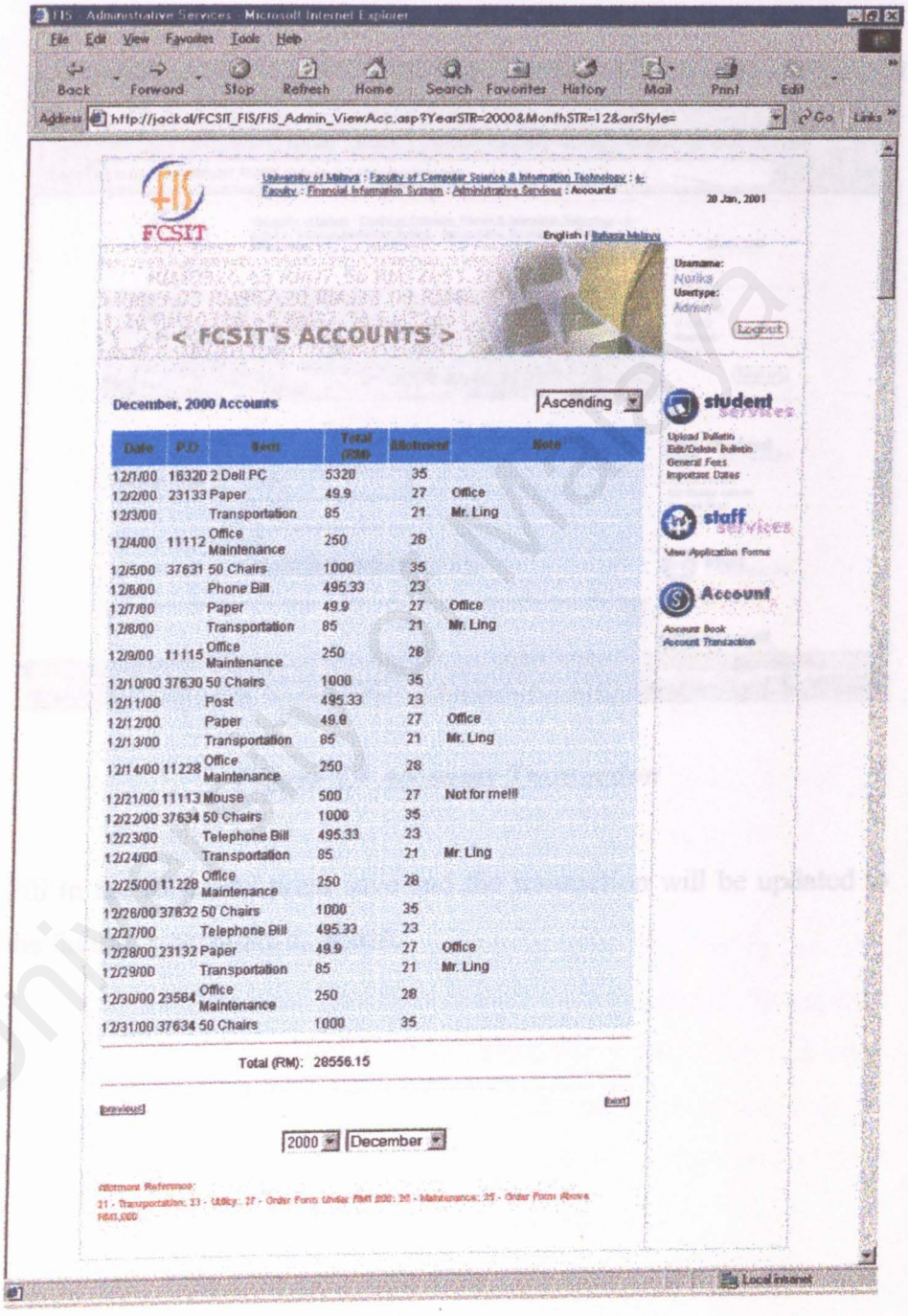


Figure 5.8 Faculty's Accounts Page

Accounts Transaction

To do an account transaction click the Account Transaction link under Accounts icon and the following page will be display.

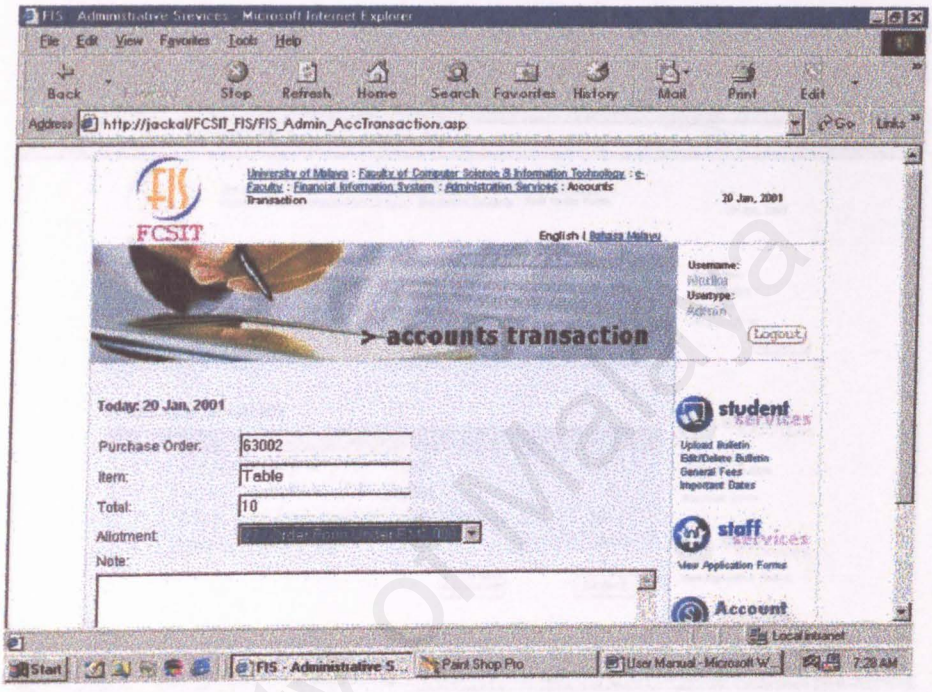


Figure 5.9 Accounts Transaction

Fill in the form and press save and the transaction will be updated to the faculty's accounts database.

University of Malaya

Approve / Reject Order Form

This function is specially authorized for the Department-Head Administrator. To approve or reject an order form, click on the Approve/Reject link, the following page will be display.

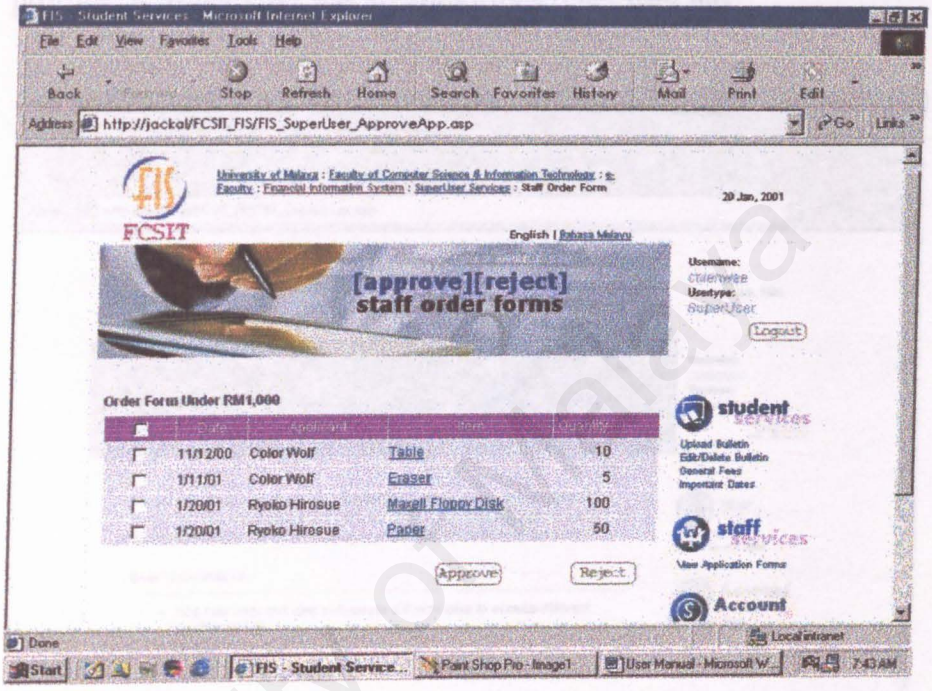


Figure 5.10 Approve/Reject Order Form Page

To view the application form just click on the item link in the table and a new page will be display showing the form's detail. To approve or reject application forms check the checkboxes on the left and press the <Approve> or <Reject> button and the information will be updated to the database where the applicant can view their application's status.

Chapter 6 Super User Section

Super User is a category of users who are able to maintain users profiles. A Super User has all authorities other user has.

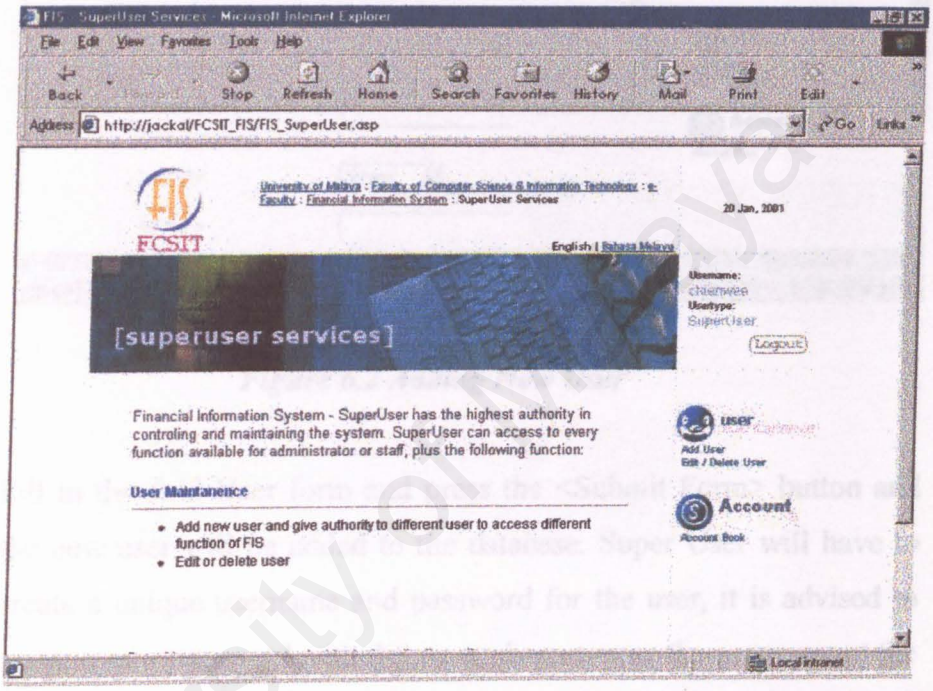


Figure 6.1 Super User Services Page

User Maintenance

User maintenance is regarding the creation of new user as well as updating an existing user profile.

Creating New User

To add a new user or authorize someone to use FIS, click the Add User link under User Maintenance icon.

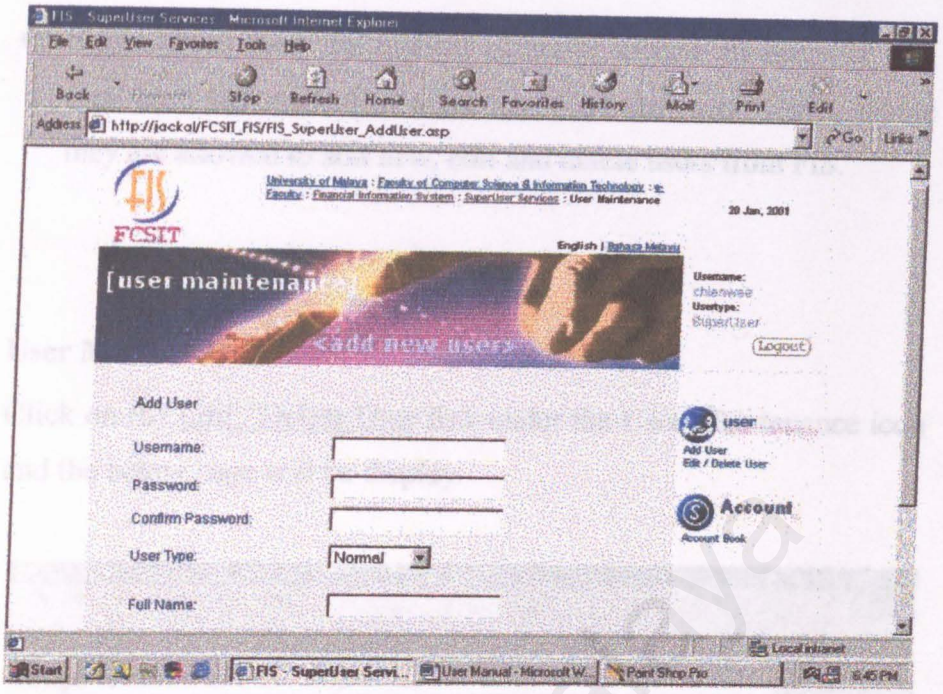


Figure 6.2 Adding New User

Fill in the Add User form and press the <Submit Form> button and the new user will be added to the database. Super User will have to create a unique username and password for the user, it is advised to create a new password with the same characters as the username at the first time, so that user can easily change their password when they first login.

User Type is referrer to the authorization level of a user.

- A Normal User is only allowed to apply budget, view their application forms status and change password.
- A System Administrator has the authority to maintain the student's financial information, process order form and maintain the faculty's account.
- A Department-Head Administrator can access to all functions System Administrator can plus extra function to approve or reject order forms.

University of Malaya

- Super Users have the highest authority among all users of FIS, they have full control of other users' access level to FIS, where they are allowed to add new, edit and delete users from FIS.

User Maintenance

Click on the Edit / Delete User link under the User Maintenance icon and the below page will be display.

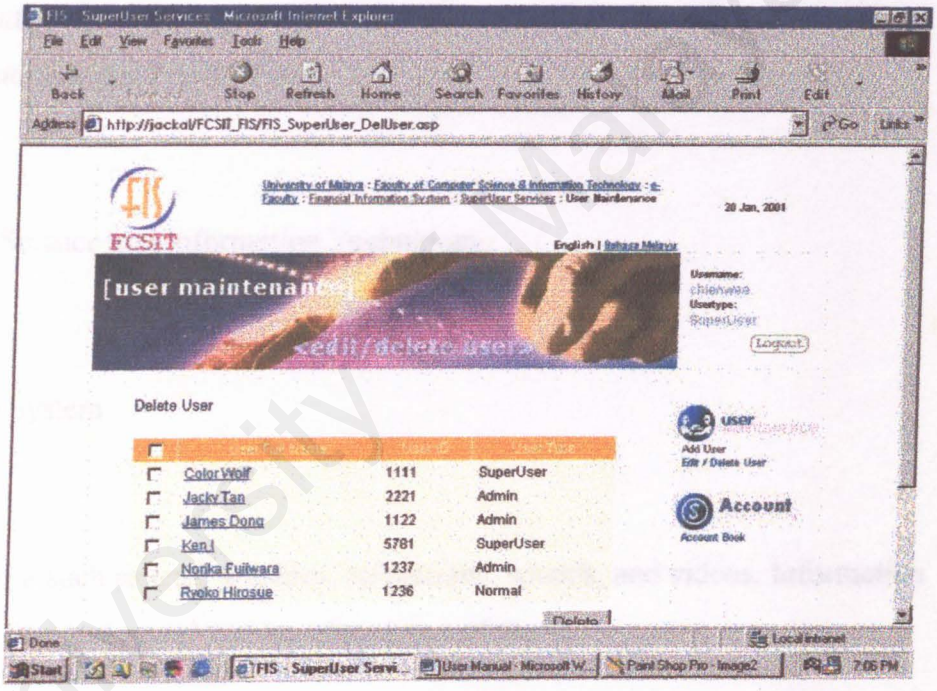


Figure 6.3 User Maintenance Page

To edit a current user's profile, click on the User Full Name link in the table and a new page showing all details of the user will be display. Change whatever data needed and press the <Submit Form> button and the new data will be updated to the database.

To delete users from FIS, check the corresponding checkboxes at the left of the table and press the <Delete> button at the bottom.

Appendix - Glossary

Data

Facts that can be recorded into some descriptive forms

Database

A shared collection of logically related data designed to meet the information needs of multiple users in an organization

Database administrator

User who wants to maintain the FIS database

FCSIT

Faculty of Computer Science and Information Technology

FIS

Financial Information System

Multimedia

The integration of media such as text, graphics, animations, sounds, and videos. Information in the form of images, text and sound various other types of media

Network

A group of interconnected computers

Normal User

Normal user here referrer to all Staff in FCSIT (not the students)

Super User

User, who wants to create authority to a new user and maintains existing user's data

University of Malaya