Nama	:	Leow Weng Yuen
Nombor Matrik	:	WEK98104
Kod Subjek	:	WXES3182
Nama Pensyarah	:	Prof Madya Dr P. Sellapan
Nama Moderator	:	Pn. Rodina Ahmad

Smart Healthcare Information System
Pharmacy Management System

By
Leow Weng Yuen
WEK98104
Session 2000/2001

A graduation exercise submitted to the
Faculty of Computer Science
& Information Technology
University of Malaya

In partial fulfillment of the requirements
for a Bachelor's Degree in Computer Science

Department of Software Engineering
Faculty of Computer Science and Information Technology
University of Malaya
January 2001

By Leow Weng Yuen

Table of Contents

By Leow Weng Yuen

Table of Contents

Acknowledgements

I'd like to thank my supervisor Dr. P. Sellapan for his support and the freedom he gave me in researching and developing the Pharmacy Management System. His insight and views have been of immense helpful in getting the final end product out.

Thanks also goes out to Pn. Rodina Ahmad who made use of her time to look at this project and evaluate it for this effort; my final year project. Thank you so very much for the time spent in moderating my thesis.

I would also like to thank the members of the Smart Healthcare Information System team who gave me the opportunity to act as a leader of sorts in terms of making development decisions for the overall system. Their trust in me has also enabled me to find out what it is like to work in a programming team and has given me some experience as to what it would be like outside in the working world. So I my thanks go out to Gan Lee Yean, Choong Fui Ken, Soo Wei Bing and Jonathan Romulo Geronimo for all their support and help in this group exercise.

I would also like to mention my appreciation to my housemate, Mr. Kwa Kheng Hoe for his insights regarding pharmacy issues as well as his explanations on how some processes happen in a pharmacy. Thanks to everyone who helped out.

By Leow Weng Yuen

Abstract

This document covers 3 main areas in outlining the intended system in question; that is the Pharmacy Management System. The first section of this document is basically an introduction to the system. It covers the objective of the said system as well as the scope and area that it would cover. It also explains briefly its interaction and integration with other components being developed by 4 other course mates in the team. The Pharmacy Management System is a part of a larger system that we have dubbed as the "Smart Healthcare Information System". The second section covers an in-depth study of the tools that will be used to develop the system. This part of the document will contain an analysis as well as summaries of the various tools that have been looked at. It contains technical details as well as some conclusions that have been reached after studying the specifications of the various tools. Most of the information gleaned for this section was taken off various Internet articles as well as some books that are related to the project being proposed. The third section suggests the methodology that will be used to develop and test the system. The system design as well as the process taken to develop the system is covered in the later sections. This whole report caps everything off with screenshots and short captions to explain the final system that was developed.

Introduction

Project Definition

The system being proposed is a Pharmacy Management System and the most important feature of the proposed system is that it will be a web-based information system. It is to be used in a hospital environment and is intended to be used by medical staff especially those who deal with medication and its prescription.

Traditionally pharmacies are managed in the proceeding manner in hospitals around the country. When it comes to prescribing medication the process is as follows:

1. The patient goes to the doctor to be treated for his/her ailment. After an examination the doctor gives a diagnosis and then prescribes some medication on a sheet of paper, which is then given to the nurse or patient. The patient then leaves the examination room.
2. The nurse/patient then takes the prescription to the pharmacy, which is located in the hospital and gives it to the pharmacist on duty. If the medication cannot be found in the hospital the patient is then required to get the prescription from outside the hospital.
3. The pharmacist looks over the prescription; checking it for drug interactions. Should any negative interactions be found the pharmacist would then suggest alternative drugs to replace the ones that were prescribed by the physician.

4. The pharmacist then goes about preparing the medication in bottles or packets to be given to the patient. The proper dosage based on the physician's prescription is filled and the medication is labeled with instructions on how and when to consume them. The patient is then called and instructed verbally again as to how and when to consume the medicine.
5. The patient pays for the medication (either at the pharmacy counter or at the billing counter) then leaves the hospital with it.

The proposed system would get rid of a lot of these steps and the process by which medication is prescribed would occur in this manner:

1. The patient goes to the doctor for a treatment of his/her ailment. After an examination the doctor gives a diagnosis and then prescribes some medication by punching the data directly into a computer terminal. The patient is given an identification number; leaves the examination room and goes directly to the pharmacy to collect the medicine.
2. The pharmacist can then check for the prescription based on the identification number given to the patient and he/she then proceeds to fill the prescription. Drug interactions can be checked on the system itself using a drug knowledge base, which is available on the system.

3. The patient is called and given instructions for the medication being prescribed. Billing information is straightaway submitted to the payment counter. The patient is then directed to the counter where he/she pays for the treatment rendered.

In terms of keeping an inventory of the medication that is currently in stock in the hospital, the pharmacist would traditionally have to go through these steps:

1. Every time a prescription is filled the pharmacist would have to enter the data into a form and occasionally keep track of the amounts that have been dispensed.
2. At periodic intervals this information is used to keep track of the total amount of medication that is in stock in the hospital. Should any item dip below a certain value then the pharmacist would have to make an order with distributors to re-supply the hospital's pharmacy.
3. Verification is done between the pharmacist and the distributor and finally the supply of medicine is delivered to the hospital.
4. Payment is done via the accounts department of the hospital.

By implementing an online Pharmacy Management System, keeping track of the pharmacy's inventory will become a routine and simple chore; with a number of functions being automated. The proposed system would work in the following manner:

1. Since prescription data is entered directly into the system; stock information is updated instantly thus precluding the need to manually update the data.
2. Specific stock levels can be entered into the system; which would warn the pharmacist whenever certain medication dips below the minimum level.
3. The pharmacist will then be able to make the appropriate orders immediately.

Another feature that can be included into the system is a medication management system that will be used internally in the hospital to manage the dispensation of medication to patients in wards.

Project Objectives

The objectives of the proposed system are pretty straightforward and they shall be outlined in the proceeding section. They will be covered in as brief a manner as possible in the following paragraphs.

Technical Objectives

The main objective of the proposed system is to have a computerized system that is **easily accessible** by medical staff and the general public. Each class of user is given different kinds of access to the system depending on their needs and access privileges. This is the reason why the application being proposed is going to be a web-based one as it is the easiest and most effective way to make the system available to large groups of users without having to install software on client machines. A web-based application also enables to system to reach a wider audience through the Internet.

Ease of use is also a primary objective that needs to be addressed when it come to building the Pharmacy Management System. This objective is promoted through the use of web pages, which are familiar to most users. Web pages are flexible enough to be changed and redesigned should the need arise after getting feedback from users regarding the layout and design of the interface. With enough feedback from users the most effective user interface can be designed for the system.

Keeping costs down is also a major objective that needs to be fulfilled by the proposed system. Due to its web based implementation the cost would only be limited to development costs (including design and testing) and licensing fees for the software that will be used to host the site and database.

Ease of maintenance is also an important objective that must be achieved by the proposed system. As the entire system will be centralized; maintenance is concentrated on only one or two machines and with proper procedures taken; downtime can be minimized or even taken out of the equation all together.

Security is also another major consideration that needs to be taken into account for the proposed Pharmacy Management System. Having a centralized system enables system administrators to track users as well as system access through logging facilities that are inherent features of most web servers. It also allows administrators to lock out users should they try to exceed their user privileges or if they try to hack into the system. Security measures can also be taken when writing the scripts that access the database, which adds to the overall security of the application.

System Objectives

Besides all the objectives that need to be fulfilled to satisfy technical excellence the following objectives need to be fulfilled as well to improve the services provided by a hospital.

Improved productivity is one of the main reasons why a Pharmacy Management System is needed in the day-to-day operations of a hospital. Fewer steps need to be taken in hospital operations especially when it comes to prescribing and collecting medication from hospital pharmacies. Productivity is also increased because better information is available via the proposed system, which includes a drug knowledge base as part of the system. The sharing of information that comes naturally from the use of such a system also helps each user to perform better in their respective jobs. This is because confusion can be avoided by procuring information directly from a shared source.

The **quality of health care can also be improved** by using a computerized system to manage medication distribution. This is because drug interactions can be checked instantly via the drug knowledge base and ensures that allergic reactions do not occur when patients are given a particular drug.

Project Scope

This section covers the scope of the application; it explains the areas that will be covered by the application so that no deviation occurs after development has started.

Tasks that need to be performed

First and foremost, **a machine needs to be setup** with the proper operating system. The machine needs to be web ready and it also needs to have a number of applications installed on it so that it'll be ready to serve a website over the Internet. The machine would need to have a web server installed to service the requests being made by clients for the web site. In addition a database server will also be hosted on the machine so as to facilitate storage and retrieval of data on the site since pharmaceutical data needs to be kept on the machine.

One of the first few tasks that must be done is to **design a solid database** that can be applied in a web-based environment. A proper database server needs to be setup for this purpose on the machine that will host the web site so as to facilitate data access. It must be robust enough to handle request from multiple users and should also be a relational database. Database rules need to be applied in the design of the database such as:

- Entity relationship rules.
- Logical and physical views of data.

- Normalization of databases and so on.

The third task that needs to be done to make the proposed system a reality is **programming** work mainly via scripting languages to interface the front end with the database. Scripts written need to be able to connect to the database securely to make changes to the database. The scripting language used will also need to be served via a web server and if possible the scripts need to be kept hidden from client-side browsers. The purpose of keeping these scripts hidden is to help maintain the security of the site especially when it comes to database access.

In addition to all the tasks mentioned previously, an **interface** needs to be designed using Hyper Text Markup Language (HTML) as a tool. Web pages need to be designed and built to function as the graphical front end of the application. A proper tool such as Microsoft FrontPage or Macromedia Dreamweaver can be used to design the pages rapidly so that more time can be devoted to database design and scripting.

Functionalities that must be included in the system

Among the functionalities that the proposed application would need to have is the ability to connect to a database server remotely using a scripting language. The scripts would need to have security features as well so that user tracking can be done throughout the site especially when it comes to database access.

The scripts would also need to have the following capabilities to enable users to manage the contents of the database:

1. Adding new entries into the database.
2. Deleting entries within the database.
3. Updating data in the database.
4. Enforcing data integrity rules to ensure that data is not corrupted.

Some form of security needs to be a functional part of the proposed system as well so as to ensure that only authorized users are allowed to access this portion of the web application. This can be implemented via user IDs and passwords that need to be entered in at a login screen. This would help to maintain another layer of security in addition to the inherent security features of the operating system and database server that is being used.

Most importantly is that the application functions totally on the web and need not require any other software except for a web browser to run it. All users need to do is point their browsers to the proper URL and from there be able to access all features via the Internet or over an intranet after logging in to the secure site.

Literature Review

Overview

Literature reviews were conducted from the beginning of July till early September to gather material on various items and issues related to the proposed system. Articles and text were obtained from various printed publications but the most useful and up to date information was obtained from the Internet. The issues reviewed in this section is not a wholesale reproduction of the various material that was looked at but rather is a summarized analysis that paints a broader view on the best steps to take to implement the proposed system in a real world situation.

A number of areas will be covered in this section but it mostly will concentrate on what a Pharmacy Information System is all about as well as a quick look at the tools that will be used to build the proposed Pharmacy Management System. The areas being reviewed are as follows:

1. Pharmacy Information Systems: An Overview.
2. Pharmacy Information Systems: An Analysis.
3. Synthesis of the Pharmacy Management System.
4. Operating Systems: A Comparison of the Various Environments being used in Server-Based Computers.

5. Web Servers: An Overview of the Various Web Servers in use by the Web Industry Today.
6. Scripting Languages: An Analysis of the Scripting Languages available for the Web.
7. Database Servers: A Comparative Analysis.
8. Documentation Processes.

The information presented here does not promise to be the most authoritative and complete analysis of the all material covered but it should at the very least give an unbiased summary of the strengths and weaknesses of the tools that will be used. In addition it should also give you; the reader a better awareness of the software and languages being used on the Internet currently as well as a better understanding of the role of computers in hospitals today.

Pharmacy Information Systems: An Overview

The following text is a summary on what a Pharmacy Information System is and how it would improve workflow processes in a hospital's day-to-day ward operations. There are 2 major aspects that need to be taken into account when designing a Pharmacy Information System for practical use:

- An **administrative oriented inventory system.** (Stock monitoring, stock valuation & pricing, reorder level assessment etc.)
- An extensive **drug delivery and administering system.** This would refer to medication delivery on a large scale like hospital level or to a smaller scale, for example in outpatient treatment.

Administrative Oriented Inventory System

An administrative oriented inventory system is based on standard stock controls and it really is very similar to current inventory management systems. Such a system needs to have the following features in order to be effective in managing the inventory in pharmacies:

- Stock monitoring.
- Stock valuation and pricing.
- Reorder level assessment.
- Overdue reminders to supplies.
- Delivery checks against orders.

- Demand forecasting.
- Usage and cost analysis.
- Decentralized inventory control.
- Invoice checking.
- Stock movements.
- Drug expiry dates.

Stock monitoring features are required to keep an eye on supplies that are currently available in storage. Because data is updated in real-time when the pharmacist uses the system stock data is always current and this allows the system to accurately monitor supplies for fluctuations and if necessary alert pharmacists should any need arise.

A centralized database that contains pricing information on current stocks also allows the user to evaluate the total value of supplies that reside in the pharmacy at any given moment. Pricing information can also be accessed easily, which would allow users to get pricing information whenever they need it. It also allows the user to update pricing information and run cost analyses easily at the click of a button.

Live updates of stock information would also allow such a system to assess reorder levels. This allows the system to warn the user should there be any item that is severely under stocked. This allows the user to place orders for the said items immediately without delay and thus ensure that the hospital's supply of medication never runs low.

The system should also be able to prompt the user with reminders for orders that have been overdue. Such a capability would make sure that the user is sufficiently reminded about all outstanding orders that haven't arrived yet especially those orders that are late. Such a feature would help especially in an understaffed hospital where employees are overworked and usually busy at all times.

Another standard feature that should be part of a pharmacy information system's repertoire of inventory tools is the ability to run delivery checks against orders made. Delivery checking capability ensures that supply deliveries are accurate so as to correct possible errors made by suppliers.

The management tool should also be able forecast demands on different medication based on data fluctuations during the everyday operations of a pharmacy. This data can be procured after usage analysis is done in the system. Processing of such data enables the pharmacist to project demand requirements for various medication being prescribed to

patients. Forecasting would also allow the pharmacist to place orders in advance for drugs that are in heavy demand so as to avoid running out of supplies in the future.

Stock movements can also be checked via the system by examining the data in the database and tracking the number of supplies currently in stock. Checkout dates can also be searched to determine dates when supplies were dispensed and large shipments can be flagged to alert the pharmacist of such movements.

In addition to all the above; the system also enables expiry dates to be checked automatically. If any of the medication is close to expiry then the system should alert the user to order a new shipment of the particular drug as well as inform the user of the impending expiry date. When steps like this are taken patients can rest assured that the medication they receive are not expired and safe to consume.

All the features mentioned here help to manage the hospitals inventory of medication, which in the end also benefits the patient by making sure that supply is constant and fresh to dispense to the general public.

Drug Delivery & Administering System

The drug delivery and administration portion of the system would have to be able to do the following tasks during the prescription process:

- Selection & printing of drug labels.
- Dosage checking by the pharmacist.
- Drug profile registration.
- Drug interaction lookup.
- Possible alternative drug lookups.
- Printing of prescriptions.

The process of delivering and administering drugs is a time consuming and labor intensive process that requires hospital pharmacy staff to do the above tasks manually by hand in most cases. This part of the system can be implemented successfully only after the inventory portion of the system is up and running. Once the inventory system is stable then it'd be very easy to setup modules to computerize some aspects of administering drugs to patients.

It is not necessary to explain the functionalities of the drug delivery system, as the activities that have been defined are self-explanatory and shouldn't require any more elaboration to elaborate the purpose of each activity.

Functionalities of a Pharmacy Information System

The functionalities of a Pharmacy Information System can be defined and represented in the data flow diagram (DFD) below. Please take note that the system being reviewed here is a proposed system that is to be used in administering drugs in wards based on medication records issued by physicians.

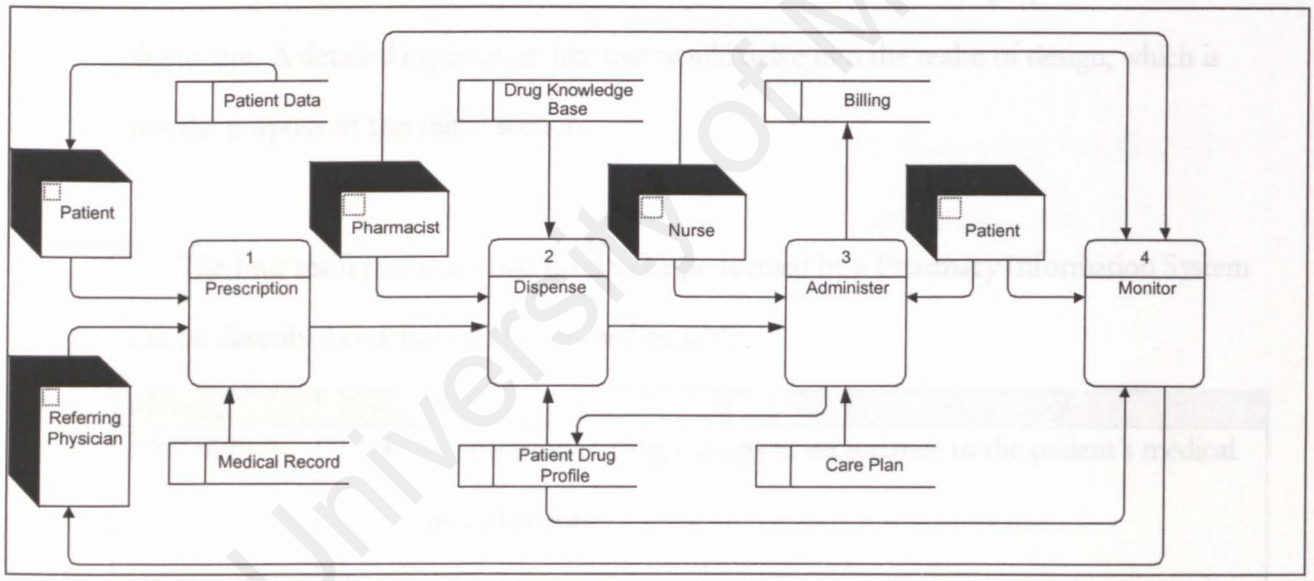


Diagram 1: Functions of the Pharmacy Information System

The four main processes that are present in the context diagram above picture the overall flow of data in the system from a top down perspective. It doesn't describe in detail

what the overall system will do but rather represents a generalized view of the main functionalities that need to be present in a Pharmacy Information System.

The discussion here will be limited to these four main processes and will only represent a general view of what each process does. A thorough explanation of each process and further explosions into smaller more detailed DFDs would defeat the purpose of this document. A detailed explanation like that would delve into the realm of design, which is not the purpose of this initial section.

The four main processes that need to be performed by a Pharmacy Information System can be described concisely in the proceeding table.

Process	Tasks
Prescription	• Preparation of drug therapy in accordance to the patient's medical record is done. • The order for the drug is filled by selecting the appropriate drug. • Checking for duplicate medication, allergies, medication control and maximum dosage is done in this phase. • User authorization and validation is also done here.

Process	Tasks
Dispense	• The prescription is read from the prescription file. • The validity of the drugs prescribed is checked against the patient's medical profile. This includes duplicate orders, drug allergies, drug-drug interactions, drug laboratory interactions as well as maximum dosages. • Prepare the drugs for dispensation. • Give advice on drug usage upon dispensation of the medication. • Update inventory information as drugs are dispensed.
Administer	• Maintain a patient drug profile. • Maintain nurse care plan/medication chart. • Notify the billing system of the applicable charges that have been incurred due to the prescription of drugs to the patient. • Return un-administered doses as well as discontinued medication back to the pharmacy.

Process	Tasks
Monitor	• Observe patients and respond accordingly to medication alerts. • Drug therapies and problems are evaluated by creating drug-use review reports. • Inventory is maintained which also includes reordering drugs as the need arises.

Table 1: Main Processes of the Pharmacy Information System

Drug Databases

A drug database is the central repository of data that contains information on all available pharmaceutical products in the hospital. It basically functions as a knowledge base for the pharmacist to refer to during the course of his/her work. This database will grow as time passes due to the development and introduction of new drugs and products in the market by pharmaceutical companies worldwide.

Drug databases contain multiple information categories that describe different aspects of the drugs already input in the system. Among the categories that need to be managed are:

- Drug identification data.
- Stock and purchase administrative data.
- Medically related data.

It is hoped that by building such a database, the contents can be later turned into a drug information base that serves as a complete knowledge repository that describes the following:

- Applicability of drugs.
- Drug side effects.
- Adverse drug reactions.

- Drug-drug interaction.
- Normal dosages.
- Kinetic parameters.
- Chemistry information.
- Administration instructions.

Pharmacy Information System: An Analysis

The Pharmacy Information System that was reviewed basically was thought up as a management tool to aid in drug dispensation in hospital wards as well as an inventory tool to monitor drug supplies in the hospital.

The drug dispensation part of the system ties in closely to the inventory management modules because naturally as drugs get dispensed items in the inventory will be reduced. As a result the drug dispensary and administration module ought to have access to update information regarding drug supplies in the hospital. The updating of information occurs in real-time and occurs in the background as the pharmacist prepares labels for drugs that are being administered to patients.

As a whole the Pharmacy Information System requires access to patient medical records, which means that this module needs to communicate with the Patient Information Management System. Such a relationship would require a complete medical profile of the patient that will be provided by the fore mentioned system. The building of such a profile is not easy and requires input from physicians, which in some cases will be difficult. This is because doctors would need to interact directly with the system, as they are the most qualified people to maintain a detailed patient profile of their own patients. Other medical

staff are not qualified to create the needed information for the profile because it requires medical details that most often than not will only be known by physicians in the hospital.

After analyzing the details about previously thought up Pharmacy Information System it can be concluded that such a system can be set up in hospitals with modifications to some of the proposed aspects by researchers. Firstly, the dispensary module needn't be implemented as a ward administration system but rather it can first be written to handle day-to-day functions for servicing outpatient requests and as users can used to it be slowly expanded to encompass ward drug dispensation. This means that the monitoring process (the final process in the DFD of the main functions of the Pharmacy Information System) can either be scaled down or even taken out of the system all together. The other three major processes can be maintained as is or with some minor modifications to suit each process for use in an outpatient environment.

The drug knowledge base suggested by researchers is a useful but practically it may be difficult to implement because it would require a whole lot more details and information, which must be gleaned from professionals in the field. To build the database for such a system would be easy in terms of designing and normalizing the tables but the time needed to compile the information would exceed allotted time frame given for the project. This is because a knowledge base of that size and detail would require explicit information from

medical professionals so that the knowledge base will be of use to pharmacy staff. It can be included in the system but realistically it would be difficult to create a large enough drug database that will be of any use in the real world.

In short, many of the processes suggested can be used in the Pharmacy Management System being proposed in this document. The scope suggested would have to be scaled down to suit the time frame given for the project but modules can be built in such a way so that the capabilities of the system can be extended in future. This means that the omitted features can be added on easily by extending the capabilities of the scripts or even writing new scripts to provide the added features without having to modify the older scripts.

Synthesis of the Pharmacy Management System

By analyzing literature related to Pharmacy Information Systems; synthesizing an outline for the proposed Pharmacy Management System wasn't that hard. This is because most of the processes suggested were logical everyday models of how a pharmacy in a hospital operates and there is no need to change the way they work to suit the system being proposed. Also inventory systems are widely in used and there is no need to reinvent the wheel but rather modifying existing inventory systems for use in a hospital environment.

The Pharmacy Management System would have to contain the following 3 modified processes to represent the functions that will be offered by the proposed system. The modifications reflect the functions that are needed when the system is to be used in the pharmacy to service outpatients.

Process	Tasks
Prescription	• Preparation of drug therapy in accordance to the doctor's prescription. • The appropriate drug is selected. • Checking for duplicate medication, allergies, medication control and maximum dosage is done by the pharmacist. • User authorization and validation is also done here.

Dispense	• The prescription is read from the system. • The validity of the drugs prescribed is checked against the patient's medical profile. This includes duplicate orders, drug allergies, drug-drug interactions, drug laboratory interactions as well as maximum dosages. • Prepare the drugs for dispensation. • Give advice on drug usage upon dispensation of the medication. • Update inventory information as drugs are dispensed.
Administer	• Maintain a patient drug profile. • Notify the billing system of the applicable charges that have been incurred due to the prescription of drugs to the patient.

Table 2: Main Processes of the Pharmacy Management System

The monitor process has been removed because it is a phase that is appropriate for use in drug dispensation in hospital wards; which is not the focus of the system being proposed for development.

The Pharmacy Management System will be built with a lot of focus on the inventory module because it is the backbone of the entire system. Without a solid inventory module, the other parts of the system would most certainly fail, as every one of them has to interact with the inventory system in one way or another. In addition to that the inventory system manages the database, which is the central repository of information. Improper management of the database would result in a weak system with faults that may even cause the entire system to fail.

The database created for this application will have to include pricing information as well as a description of the drugs. The purpose of the drug description is to provide for detailed information that can be used later to build a drug knowledge base that can be used as a reference tool by pharmacists.

Operating Systems: A Comparison of the Various Environments being used in Server-Based Computers.

Windows 2000 Server

Windows 2000 is available in a number of different flavors each serving a different niche in the server operating systems market. Windows 2000 Server provides the following important features that make it suitable to run a web server:

- Comprehensive clustering for scalability and availability.
- Support for large SMP servers.
- High performance applications with Microsoft Transaction Server (MTS).

Comprehensive clustering for scalability and availability

Windows 2000 Server has clustering features that enable it to distribute incoming traffic across up to 32 servers, transparently to the client using a single "virtual" IP address. This makes it ideal for high-volume Web services such as e-commerce. Windows NT's support for clustering enables administrators to scale their performance as needed to the mirror demands made on the site. This feature is called the Windows NT Load Balancing Service (WLBS) and is capable of reconfiguring the cluster to direct client requests to other servers, thereby maintaining continuous availability of network should any computer fail or go offline for maintenance.

Windows 2000 also has a service called Microsoft Clustering Service (MCS) that monitors the health of standard applications and services on the server. It can automatically recover mission-critical data and applications from many common types of failure. MCS uses a graphical management console that allows the administrator to monitor resources visually as well as to move workloads around with a point and click interface.

Support for large SMP servers.

Windows 200 Server is also capable of Symmetric Multi Processing (SMP) thus it is able to run systems with up to eight processors and in special versions this support is extended up to 32 processors. This gives the administrator a powerful upgrade path for enterprise applications that need to handle more users and data as the system grows larger.

High performance applications with Microsoft Transaction Server (MTS)

Microsoft Transaction Server makes it easier to develop and deploy high performance, scalable, and robust enterprise, Internet, and intranet applications. It defines an application-programming model for developing distributed component-based applications as well as a run-time infrastructure for deploying and managing these applications. MTS 1.1 has a new client configuration utility, support for "Cedar" type libraries, and bug fixes via a Service Pack Update.

Windows 98 2nd Edition

Windows 98 was considered as a cheap alternative to serve as the development platform for the proposed Pharmacy Management System. It is perfectly capable of administering a small site but unfortunately it is unable to handle high loads due to the unavailability of certain features like load balancing, which is available in Windows NT.

It is able to serve web pages due to the inclusion of Personal Web Server on the Windows 98 CD. Installation of this utility would enable Windows 98 to function as a web server for small networks thus precluding the need to pay for the higher fee that is required to run a copy of Windows NT Server on a machine.

Windows 98 also has a better File Allocation Table format called FAT32. More disk space is available as a result of Windows' ability to store information more efficiently via FAT32. Disk wastage is lessened and more life can be squeezed out of a hard drive so that upgrades can be put off for a longer period.

Windows 98 also supports a wide range of hardware and peripherals. In this case this feature is not a useful one as Windows 98 sacrifices its stability by supporting all kinds of peripherals in the market. Since the system proposed is web-based, a better alternative would be an operating system that is more suitable for a server environment.

Linux

Linux is a completely free re-implementation of the POSIX specification. It comes complete with SYSV and BSD extensions meaning that it looks like Unix, but does not come from the same source code base that Unix comes from. Linux is available in both source code and binary form. Linus Torvalds and other contributors own its copyright; due to its open source nature it is freely re-distributable under the terms of the GNU General Public License (GPL).

Because Linux is Unix like, it is well suited to function as a development environment for web applications. Its superior stability is a feature that cannot be beaten even by Windows. Linux is capable of running 24 hours 7 days a week without system failures or crashes. Memory management is dynamic and used memory is released after a particular application ends unlike Windows, which hogs memory even though, by rights memory should be released after an application ends. In addition Linux has the following features:

- It is capable of multitasking.
- It is a multi-user platform.
- Has support for Netware clients and servers.
- Includes a LAN Manager/Windows Native (SMB) client and server.

- It multi-platform, that is it can run on any processor.
- It supports multiprocessors via SMP support.
- Linux supports multithreading.
- Support for multiple virtual consoles.
- Many networking protocols supported.
- Has memory protection between processes ensuring that a program cannot crash the entire system.
- Supports several common file systems.

Linux's stability makes it an ideal solution to develop web applications on. Linux is an open source project, which means that bug fixes and patches are always in development thus ensuring Linux's continued stability. Being open source, Linux is virtually free. The only fee that the user has to pay is a distribution fee for the various flavors of Linux like Red Hat, Debian and Mandrake.

Linux's only weakness is a lack of support for hardware making it a little difficult to setup a machine with Linux. Fortunately support for Linux is growing every single day and more peripherals are being added to Linux's list of supported hardware.

Analysis

A table comparing the different operating systems can be viewed in the following pages. Table 1 compares each operating system's features and capabilities to serve as a web development platform for the Pharmacy Management System.

Based on the table, it can be seen that Microsoft already has a number of tools that can integrate really well with Windows 2000 Server. Windows NT's graphical user interface (GUI) also makes it easy to setup applications. Monitoring the system is easier as well because the administrator is able to view the system's status visually in graphical form.

	Windows 2000 Server	Windows 98	Linux
Web server ready	Internet Information Server & Netscape Enterprise Server.	Personal Web Server (PWS) & Apache.	Apache.
Database	SQL Server, Oracle & MySQL.	MS Access & MySQL.	PostgresSQL & MySQL.
Scripting languages	ASP, JSP, Perl, JavaScript, VBScript. Perlscript & AWK.	JSP, Perl, JavaScript, VBScript, Perlscript & AWK.	JSP, Perl, JavaScript, Perlscript & AWK.
Load Balancing	Available in Enterprise Edition.	Not capable.	-
Load handling	Not robust enough for high traffic.	Unable to handle high traffic.	Able to handle high traffic on the server.
Stability	Stable.	Not so stable.	Very stable.
User Interface	GUI.	GUI.	Strong command line plus GUI via Windows Managers.
Pricing	Pretty steep.	Expensive.	Free via download.
Security	Secure login procedures and support for web security (SSL for e.g.)	Not secure.	Secure due to login facilities. Also able to log users who have entered the system.

	Windows 2000 Server	Windows 98	Linux
Remote Administration	Difficult to administer remotely.	Unable to be administered remotely unless 3rd party software is installed.	Remote administration is good and also secure.

Table 2: A Comparison of the Different Operating Systems Reviewed

Web Servers: An Overview of the Various Web Servers in use by the Web Industry Today

Overview

This section is going to be brief, as the data gathered has already been summarized into the form of a table. Since the web server is just a tool, there is no need to go into the technical aspects of each web server that was reviewed. A short comparison of the 3 most popular web servers today can be viewed in the table below:

	Internet Information Server	Netscape Enterprise Server 3.01	Apache Server 1.2
Platforms available on	Windows NT.	Digital UNIX, AIX, HPUX, Windows NT & IRIX.	NetBSD, Digital UNIX, BSDI, AIX, OS/2, SCO, HPUX, Novell NetWare, Macintosh, Be OS, Windows NT, Linux, Windows 95, FreeBSD, Windows 98, IRIX, & Solaris.
Pricing	Free with NT 4.0 option pack.	US $1,295.	Free.

	Internet Information Server	Netscape Enterprise Server 3.01	Apache Server 1.2
Setup and Administration	Simple setup procedures and integrated management tools.	Setup more difficult has strengths in terms on administration of the server.	Difficult setup and poor administrative tools.
Logging	Extended logging facilities. Able to log into any ODBC database.	Has a number of logging facilities. Unable to log into every ODBC database.	Has about the same number of logging facilities as Netscape Enterprise Server. Able to log on any ODBC database via an add-on.
Authentication & security	Very strong authentication & security features.	No certification support.	No certification support.
Content Management	Content management can be done graphically because of integration with WinNT file management systems.	Netscape Enterprise Server has some strong management features.	Poor content management features.

	Internet Information Server	Netscape Enterprise Server 3.01	Apache Server 1.2
Internet Services Supported	HTTP 1.1 compliant. Able to access HTTP Host Headers sites using any browser. Integrated SMTP support. Integrated NNTP support.	HTTP 1.1 compliant. Requires add on for SMTP and NNTP support.	HTTP 1.1 compliant.

Table 3: A Comparison of the Different Web Servers

Analysis

Based on the data gathered above, Internet Information Server (IIS) is the natural choice to function as the web server to run the web application that is being proposed. This is because of its tight integration with Windows NT and its graphical management facilities. Also IIS comes free with Windows NT without the need to purchase additional licenses. Other web servers would require additional costs with the exception of Apache Web Server. Due to the nature of the software industry, licensing is perhaps the most expensive cost that has to be incurred by users. So an integrated solution like IIS is a welcome change because it is an added value application that comes standard with Windows 2000 Server.

Scripting Languages: An Analysis of the Scripting Languages available for the Web.

Overview

There are so many different scripting languages and tools available in the market today for web development work. A study of the different tools has revealed the following 3 scripting tools to be the most talked about and most widely used on the web today:

- Active Server Pages (currently version 3)
- Java Server Pages
- Perl (currently version 5)

Active Server Pages

Active Server Pages (ASP) was first introduced in Microsoft's Internet Information Server 2.0 under the codename Denali. ASP is an alternative to Common Gateway Information (CGI) services and is a form of Internet Server Application Programming Interface (ISAPI). ISAPI technology works in this manner:

- Client requests load each ISAPI application (in the form of a DLL) once.
- The application then stays in memory to answer requests from other clients for that same service.

- The ISAPI application is relieved from memory explicitly by a system call to the application thus alleviating memory problems associated normally with CGI applications.

ASP technology is encapsulated in a single DLL file called ASP.DLL. ASP.DLL is a very small file with a file size of about 300K. It functions as an ISAPI filter that handles the interpretation of ASP pages being requested by the client. When a client requests a ASP page, the process by which the request is handled occurs in this way:

- ASP.DLL loads all required scripting language interpreter DLLs into memory to execute any server side code found in the page.
- The resulting HTML is passed to the web server.
- The web server then sends the page to the browser that requested it.

The end result is that dynamic pages are served to clients that request ASP pages. These pages have dynamic content that change every single time a client browser requests for the page. This enables web content to be more lively and interactive, giving users a sense of action occurring on the page instead of normal static HTML pages. ASP also has database connection capabilities via ActiveX Data Objects (ADO) that is based on Object Linking and Embedding Database (OLE DB) technology to connect and access information from database servers like SQL Server 7.0 for example.

ASP uses an object model called the ASP Object Model that encapsulates a number of properties and methods of six built in objects. The six objects that the ASP Object Model encapsulates are as follows:

- The Application object.
- The ObjectContext object.
- The Request object.
- The Response object.
- The Server object.
- The Session object.

Each object handles different request made by the client. The Application object handles the ASP application itself and is universal to all users. It is able to start ASP applications when the first user requests the application and ends applications when the administrator explicitly unloads in from memory.

The ObjectContext object is part of Microsoft Transaction Server (MTS) and it functions to support transactions like file access. Transactions either succeed or fail so the functions contained in transactional ASP pages either fail or succeed completely as a single unit thus ensuring that transactions are stable and secure.

The Request object handles the way interaction with client's HTTP requests occur. The Request object enables ASP pages to access HTML based form data and address line parameters. It is also able to access cookie as well as client certificate information.

The Response object handles access and control over HTTP responses being sent to the client. It is also able to send cookies and set information within cookies. It allows the web application designer total control over how data is sent to clients.

The Server object gives access to the web server where it can set many of the utilities made available by the web server. It enables ASP pages to set virtual path settings, script timeout variables and even encode information to be sent over address lines.

The Session object contains information that identifies users' sessions uniquely. This enables tracking of user's sessions via cookies that are specially set for each user.

Each object contained within ASP technology makes it a robust server side-scripting tool that makes it easy for developers to write interactive scripts for use on the World Wide Web. ASP pages also help to hide scripting information from the prying eyes of clients' browsers without compromising on the dynamic nature of ASP pages.

Java Server Pages

Java Server Pages (JSP) is a web technology that is based on a multi-tiered architecture that has more layers when compared to the traditional (two-tiered) client/server architecture. It is an evolved form of Java's servlet technology that enables server-side applications to be run.

JSP uses server-side (middle-tier) processing to create dynamic web pages and this process is simplified because JSP separates the application logic from the page design. The logic is then encapsulated in portable, reusable Java components that can be implemented on a number of the most popular platforms in the market today.

JSP technology is implemented as a Java API, which is part of the Java 2 Platform, Enterprise Edition (J2EE). Pages created with JSP actually compile into servlets meaning the JSP builds on and extends the servlet API as well. This means that JSP has access to the same set of APIs in the core Java language that allows Java programmers to build powerful web applications without too much difficulty.

JSP like ASP is simple to implement because the logic for the scripts is embedded directly into HTML or XML. This means that JSP has its own tags that can be placed into HTML pages. Like ASP, JSP pages are processed on the server first before being served to

browsers that request JSP pages. The way JSP works can be simplified in the following diagram.

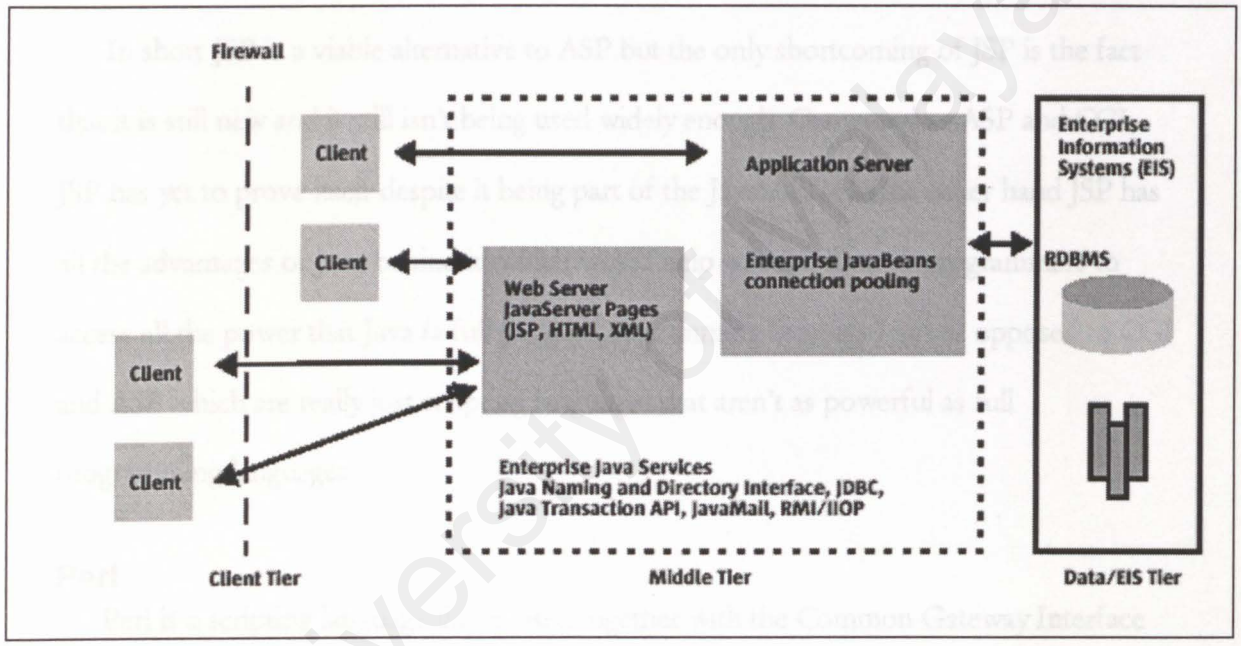


Diagram 2: Multi-tiered View of JSP Technology

JSP technology surpasses CGI and ASP in 2 fundamental areas:

- Portability – Pages built with JSP technology are portable across platforms and servers, and work with portable, reusable components.
- Easier Maintenance and Development – Because the page design is truly separate from the application logic, JSP enables tiered development and maintenance tasks.

Page authors and developers can focus on specific areas of interest without requiring the others' help.

In short JSP is a viable alternative to ASP but the only shortcoming of JSP is the fact that it is still new and it still isn't being used widely enough. Compared to ASP and CGI, JSP has yet to prove itself despite it being part of the Java API. On the other hand JSP has all the advantages of Java behind it, which would help web application programmers to access all the power that Java (a full fledged programming language) has as opposed to CGI and ASP which are really just scripting languages that aren't as powerful as full programming languages.

Perl

Perl is a scripting language that is used together with the Common Gateway Interface (CGI) standard to deliver dynamic content on the web. Processing for Perl scripts occurs separate from web pages. This means that when a web page makes a call to a Perl script, the processing for the script is done independently of the web page by the Perl processor. The client has to wait for the Perl script to finish processing before the dynamic content can be loaded up on the browser.

Perl's greatest strength is its scripting capabilities that enable web application programmers to access databases and sort data using simple commands and code. Perl even

has encryption capabilities on Unix systems for example because it can make system calls to the operating system itself.

In addition to that, Perl is supported directly out of the box by most Unix flavors. Perl runs on any web server as long as a Perl processor is installed on the machine. In Windows' case the most popular one would be ActiveState's ActivePerl that installs on Windows machines and enables the web server to process Perl scripts. Perl exists currently in its 5th incarnation and is used by most sites on the Internet due to its robustness and powerful scripting capabilities.

Analysis

ASP was chosen over the other web technologies available because it is supported by IIS directly without having to install 3rd party software. The primary reason being that it is a fully integrated solution to Microsoft's Windows NT. ASP is also growing in terms of user bases as can be seen evidently on the Internet by the growing number of web sites using ASP to create dynamic content. ASP's ability to connect to databases is also another plus. It's implementation of ADO that is based on OLE DB and ODBC technology makes it a reliable way to connect and serve information from database servers. A full breakdown comparing the 3 different web technologies reviewed can be viewed in the table on the next page:

	Active Server Pages	Java Server Pages	CGI via Perl
Duration since first introduced	3 to 4 years.	Fairly new; 1 to 2 years old.	Has been around far longer than ASP or JSP.
Web Server	Microsoft IIS or Personal Web Server	Any Web server, including Apache, Netscape, and IIS	Any Web server, including Apache, Netscape, and IIS
Platforms	Microsoft Windows	Most popular platforms, including the Solaris Operating Environment, Microsoft Windows, Mac OS, Linux, and other UNIX platform implementations.	Most platforms Windows NT included.
Database connectivity	Via ODBC.	Via ODBC & JDBC.	Via DBI libraries Perl is able to make connections to most Database Servers.

Table 4: The Different Scripting Tools Available

Database Servers: A Comparative Analysis.

Overview & Analysis

A summarized comparison of the 3 most popular database servers can be viewed in the table below:

	SQL Server 7.0	Oracle DB	MySQL
Platforms Supported	Windows NT/95/98 only.	Most platforms.	Most platforms.
Large database handling	Yes	Yes	Yes
Connection Capabilities	TCP/IP Sockets, Named Pipes (Windows NT).	TCP/IP Sockets, Unix Sockets (Unixes), Named Pipes (Windows NT).	TCP/IP Sockets, Unix Sockets (Unixes), Named Pipes (Windows NT).
Security	Integrated security with WinNT.	Fail-safe security features to limit and monitor data access offered by Oracle DB itself.	Security offered via SSH, Access Control Lists and some support for SSL encrypted connections.

Table 5: The 3 most popular Database Servers in the market

SQL Server 7.0 was chosen as the database server to be implemented in the proposed application because of its performance, capabilities and scalability. Also it integrates fully with Windows NT and it supports ASP directly out of the box, which means that ASP scripts are guaranteed to work together with SQL Server 7.0.

In terms of performance issues, SQL Server 7.0 is touted to be the fastest and most reliable database server on the market. Studies done by independent bodies have to some extent confirmed this. An independent survey done confirms that Microsoft SQL Server is the most popular database server in the market; the results of the survey can be viewed in the chart below.

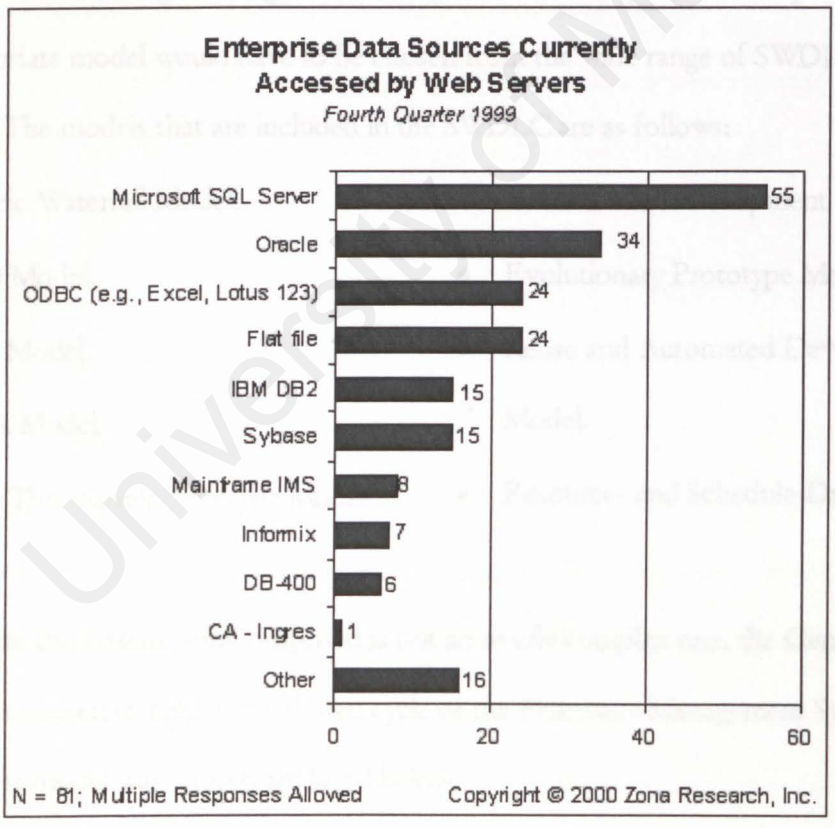


Diagram 3: Statistics on Database Server use

Methodologies

Chosen Methodology

There are a number of models that could have been used to design and implement the Pharmacy Management System in question, but only the most suitable one was chosen so that the whole development process would be smooth and wrinkle free. Since the whole system would have to go through the entire Software Development Life Cycle (SWDLC), an appropriate model would have to be chosen from the wide range of SWDLC models available. The models that are included in the SWDLC are as follows:

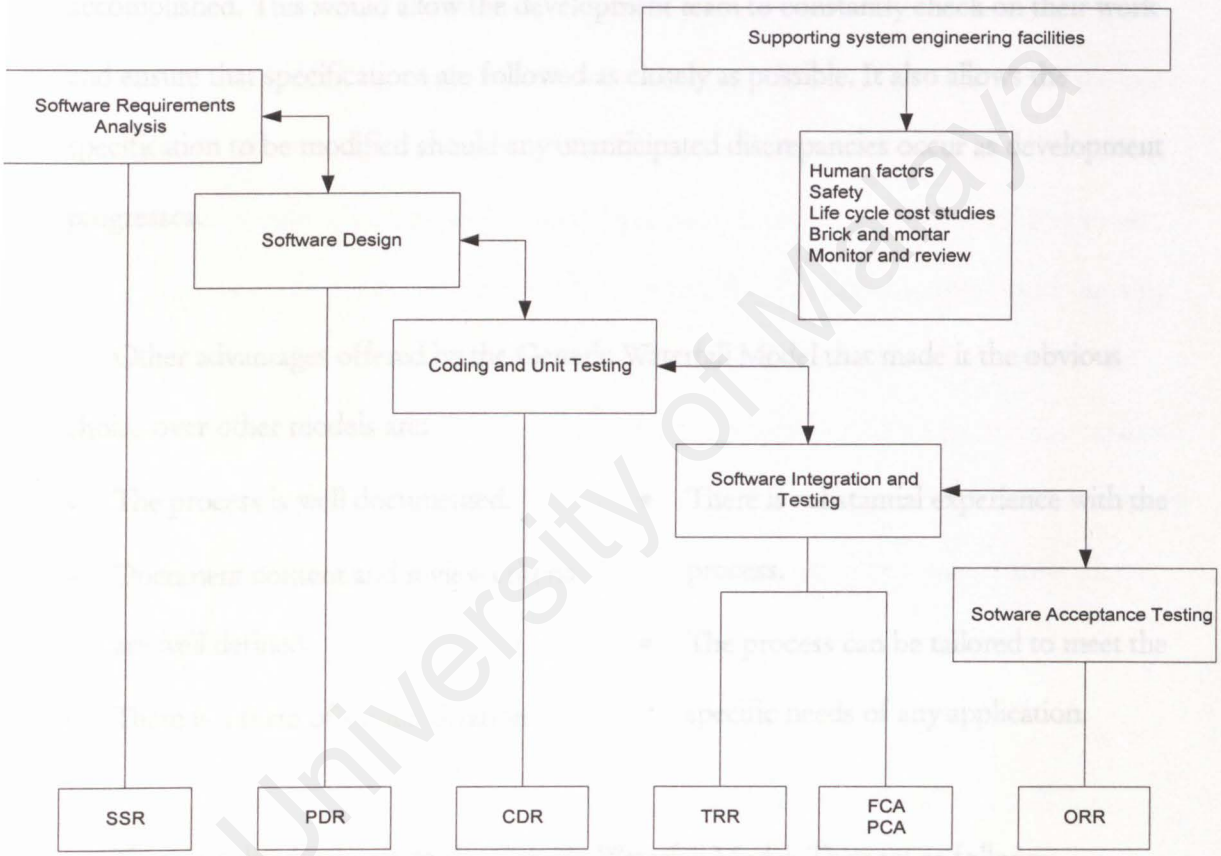
- Generic Waterfall Model.
- DOD Model.
- Spiral Model.
- NASA Model.
- Rapid Throwaway Prototype Model
- Incremental Development Model
- Evolutionary Prototype Model.
- Reuse and Automated Development Model.
- Resource- and Schedule-Driven Model.

Because the system being proposed is not an overly complex one, the Generic Waterfall Model was chosen to help in the design cycle of the Pharmacy Management System. The reasons this model was chosen are listed below:

- The proposed system is a web application and the Generic Waterfall Model makes it easy to go back and forth from one stage to another should any errors occur.

- Web application development fits into the Generic Waterfall Model really well because of its dynamic nature.
- Web application coding requires that scripts be run in order to find bugs and errors because of the unavailability of tools like compilers which are available for standard programming languages. The Generic Waterfall Model suits this form of development as coding and unit testing can be done easily.
- Integration testing can also be done easily and should errors be found in any module then it'd be easy to return to the coding and testing phase.

The Generic Waterfall Model enables the development of modules concurrently in small increments. This means that modules can be developed independently and allows the Pharmacy Management System as on overall module to me integrated into the Smart Healthcare Information System. The requirement specification phase can be revisited indefinitely as coding occurs, thus allowing more flexibility to the entire development process. This model can be pictured in the DFD on the following page:



SSR = System software review
PDR = Preliminary design review
CDR = Critical design review
TRR = Test readiness review
FCA = Functional configuration audit
PCA = Physical cofiguration audit
ORR = Operatinal readiness review

Diagram 4: The Generic Waterfall Model

Using the Generic Waterfall Model reviews of each phase of development can be accomplished. This would allow the development team to constantly check on their work and ensure that specifications are followed as closely as possible. It also allows the specification to be modified should any unanticipated discrepancies occur as development progresses.

Other advantages offered by the Generic Waterfall Model that made it the obvious choice over other models are:

- The process is well documented.
- Document content and review criteria are well defined.
- There is a form of standardization.
- There is substantial experience with the process.
- The process can be tailored to meet the specific needs of any application.

There are disadvantages to the Generic Waterfall Model. They are as follows:

- The paradigm fails to recognize the concurrent and iterative nature of the process.
- The process may not be effectively tailored to the application.
- It is incompatible with the use of the Ada language.
- It is thought to be incompatible with Expert System development.
- The process does not formally involve users and operators.

- The process is thought to be expensive.

The disadvantages above apply to most applications but for a system like the Pharmacy Management System, they don't apply. This is because the system is small and it interfaces minimally with other modules in the Smart Healthcare Information System (SHIS) so not much tailoring of the model needs to be done to suit the system Being web based the only thing that will be shared by each module of the Smart Health Information System is the database so the loose format of the Generic Waterfall Model is suitable for the system as a whole. Because the database is the core component of SHIS, extra care needs to be taken to design a well thought out database that is efficient enough to handle requests from all the different modules.

Project Plan & Procedure

The entire project plan will be based on the Generic Waterfall Model that was mentioned in the previous section.

Software Requirements Analysis

The first phase that needs to be carried out is the Software Requirements Analysis phase. It involves conducting interviews with people who are going to use the system. The parties that need to be involved are:

- Pharmacists
- Doctors
- Nurses
- Sales staff from pharmaceutical companies
- Hospital information systems personnel

These people are interviewed to find out the processes that are involved in each of their jobs. By understanding each person's role; a tailor made system for medical personnel can be created.

Vital information that is needed to design the database is also procured in this phase. This is done by questioning staff (the pharmacist in particular) to capture details like stock information and important pharmaceutical details.

Software Design

The software design phase includes designing the database that would contain all the data that is needed in the system. Besides that a list of modules would need to be thought up so that the system can be built in a modular manner thus ensuring that each component is capable of running independent of each other. Modularity ensures that problems can be isolated and fixed should any bugs surface during the coding phase.

There are 2 kinds of modules that need to be written for the system. The first module that needs to be written is an administration (admin) module that can only be accessed by authorized staff like information systems personnel. Admin modules give access to the inner functions of the system. Among the capabilities that would need to be included in the admin modules are:

Admin Module	Function
User administration	To allow the administrator to add new users into the system.
Password administration	Allows the administrator the administer changes to passwords.
Logs	Allows the administrator to check the logs of the system via the web without needing physical access to the server.
Path settings	Enables admin staff to change path settings of the scripts so that relevant paths are always correct.

Admin Module	Function
Full access to SQL functions.	Administrators should be given full access to the database; that means they can check and make changes to the database according to their discretion.

Table 6: Admin Modules in the Pharmacy Management System

User modules enable to user to use the system while allowing the system administrator to limit their access. User modules for the Pharmacy Management System can be divided into a number of categories as each user is given different privileges according to their roles and functions.

The most important user of this system is the pharmacist. So the modules that are required by pharmacists are listed in the table below:

User module	Function
Database editing	Enables the pharmacist to edit records in the database of pharmaceutical information.
Dispensary module	Enables the pharmacist to check for physicians' prescriptions and check for drug interactions.

User module	Function
Drug Knowledge Base	The pharmacist can check this module for drug information as well as enter in new information. Searches can be run in this module as well.

Table 7: User Modules for Pharmacists

The second most important user of the system would be the physician who is prescribing the medication to the patient. Doctors are restricted in terms of database access. This is to ensure that pharmaceutical information is not compromised. A small table would be sufficient to contain the prescription information. This table would have to be cleared frequently to prevent it from growing to large proportions. The modules that the physician will have access to are listed in the table on page 66:

User Module	Function
Prescription submission	Prescriptions are sent directly to the pharmacist.
Pharmacist confirmation	The doctor can discuss details with the pharmacist via phone so that prescriptions can be double-checked. This allows the doctor to get advice from pharmacists regarding drug interactions.
Drug Knowledge Base	Physicians' can only access the drug database without editing facilities.

Table 8: User Modules for Physicians

Patients too will be able to access the Pharmacy Management System but their access will just be limited to the Drug Knowledge Base where they can get medication information like pricing, recommended dosages as well as side effects. The modules that the patient will need access to, are in Table 9.

User Module	Function
Search module	The patient can search the database for the drug or drugs that they are seeking.
Viewing module	Enables the user to view the drug information that was requested.

Table 9: User Modules for Patients

The user interface is also designed in this phase and this takes into account user preferences as well as good design principles to ensure that the system is easy to use.

Coding and Unit Testing

This phase is the most time consuming part of the whole process. In this phase, scripts are written to give all the functionalities that were outlined in the design phase. The scripts are written in the form of modules that then have to be tested to ensure that they work. Bugs are ironed out in this phase. The interface is also created in this phase via HTML coding using HTML editors. After testing of each module is done then integration testing occurs.

Software Integration and Testing

All the modules are tested together in this phase and if any problems are detected then the coding phase is repeated for the problematic modules. The integration phase also

occurs on a bigger scale when the Pharmacy Management System is to be tested as part of the Smart Healthcare Information System.

Software Acceptance Testing

This phase involves testing the application with users and receiving feedback from them. If necessary, the coding phase is repeated should the user come across any errors. Once the application receives the thumbs up from users then it is ready to go live. Documentation like manuals for example need to be created after the software is ready to be released.

Documentation

Documentation needs to be created to aid users who are going to use the system. This section will look at some procedures that need to be taken to create good documentation for systems.

One of the first few things that need to be done is to name and define the document being produced. Below are the steps that should be taken before documentation work actually occurs:

- Name the item that is being documented & start thinking of the document in that term.
- Define the area of interest of the document & review the purpose of the system. Terms must be concrete, explicit & definitive.
- When there are competing claims for the manual then separate manuals should be written for each claim.

The next step to be taken is to define the user of the system. It means that users' needs and points of view have to be investigated to establish the users' mindset. The steps in this phase are as follows:

- Build a profile of the group that you will be writing for.

- The definition must be explicit & definitive and it'll help structure the approach taken when writing the manual.
- Investigate the users' needs even if specific details were given by management who ordered the manual to be written.

Research & organization of material also has to be undertaken by the document writer to make sure that the document being written for users are relevant. This involves gathering material and conducting interviews. In this phase differentiate the test system, the training system or the live system that'll be worked with when gathering material, which helps to ensure that the material picked up would be useful for the live system.

Designing the manual itself is a process that involves designing the page format, type and graphics with the user firmly in mind. The way the manual is presented is integral to getting a good first impression from the user. A graphics artist will be invaluable help to come up with a good design and layout for the entire manual. Some formats are transportable and can be lifted off-the-shelf from other documents already available. If off-the-shelf methods don't work then a custom design would have to be done. Standard page layouts help to maintain a common look and feel for different sections that would help users understand the manual better. Some techniques used to create standard page layouts are as follows:

- Use the standard only where appropriate.
- Avoid complicated layouts.
- Keep the user in mind.

The 3-part technique breaks up the manual into three distinct sections:

- Introduction part.
- Information part.
- Technical part.

Graphics, diagrams and pictures can be used to spruce up the documentation and help to explain material in the manual. They should only be used for the most important topics. Good typography also plays a prominent part in maintaining readability and legibility. Where typography is concerned, these areas need to be covered:

- Typeface style.
- Sans or sans serif fonts (sans serif fonts shouldn't be used as body text).
- Typeface consistency.
- Special typefaces for special needs.
- Italics used to highlight important elements.
- Line leading.
- Line length.
- Justified or ragged right.
- Color.

- A screen typeface for representing on-screen elements.
- ALL caps should be avoided
- Column headings.

When the look of the page is being designed the following considerations have to be made:

- Page size.
- Paper choice.
- Page element styles.

The look of the manual should also be taken into account when it comes to designing the manual. The following items need to be taken into account when the look of the manual has to be decided upon:

- Section summaries.
- Page numbers.
- Section numbering.
- Paragraph numbers (use only when absolutely necessary).
- Numbered items in a list.
- Figure & table numbering.
- Revisions, maintenance & numbering.
- Section dividers.
- Conventions & style.

Once all the above is done a blueprint is created. The table of contents is designed as a blueprint for the manual and it involves the merging of the author's understanding of the user and knowledge of the system into it. Part of this process is also to do an estimated page count for each section of the manual. Users should be involved at this stage to make them feel like they are a part of the decision making process for the new system. Once the blueprint is completed into a finished page format then it is time to gain approval for the manual's plan.

In the blueprint the activities index is used as a place where users can look up topics based upon the task that they want to accomplish. Numbering in this section is not sequential but rather is topic based. Users refer to the section that they want based on the topic that they are interested in. Contents by sections help to guides users through each chapter at the start of each section. By focusing on contents and their emphasis, a manual can be suited to cater for both advanced users & beginners. Sections and topics can be shifted around to help users to move around in the manual. Topics can also be shifted around to help out users by using a logical pattern that the user can follow.

In conclusion the contents page should serve as a guide to what the system does instead of being just a pointer to different sections in the manual. By taking all the steps outlined

above, good manuals can be written. By following the process above, it is highly probable that the manual produced will be used instead of being tucked away onto a shelf.

Software Design

Overall Design of the Smart Healthcare Information System

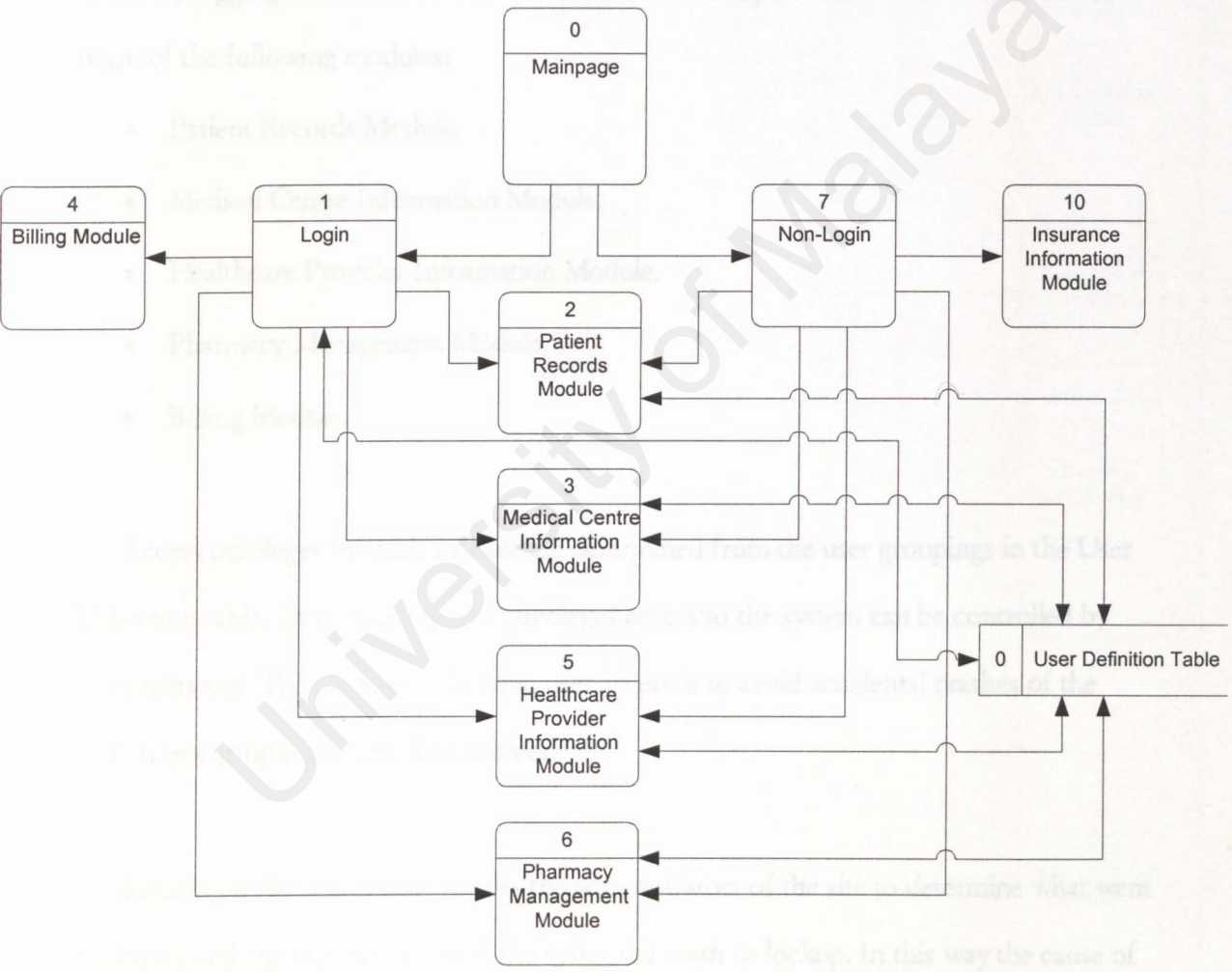
Overview

After much discussion, an overall view of how the different modules will interact with each other was thought up and a Level 0 data flow diagram (DFD) was drawn out to picture the overall system in a graphical manner.

The User Definition table is the only shared data source at this level of the system design. It contains the user names and definitions for every user in the system. Each module in the system accesses this table to verify users before they are allowed entry into any particular module.

The main entry point into the system is via the main page. At the main page the user can access a number of general functionalities available to everyone without having to become a registered user. Registered users are able to access more functions by logging into the system using a specified user name and password which are determined by the system administrator.

agram 5: Level 0 Data Flow Diagram Depicting the Smart Healthcare
formation System



Login Users

Login users have access to other value added services that are not available to regular users. By logging in the user will be able to view or modify information in at least one or more of the following modules:

- Patient Records Module.
- Medical Centre Information Module.
- Healthcare Provider Information Module.
- Pharmacy Management Module.
- Billing Module.

Access privileges for each module are determined from the user groupings in the User Definition table. By controlling user privileges access to the system can be controlled by administrators. The purpose of limiting user access is to avoid accidental crashes of the system or corruption of the data sources.

Restricting site access also enables the administrators of the site to determine what went wrong should any user access cause the system to crash or lockup. In this way the cause of errors can be determined without having to check the entire system. A more detailed description of the how the Login portion of the system works can be viewed in the Level 1 DFD in the diagram on the following page.

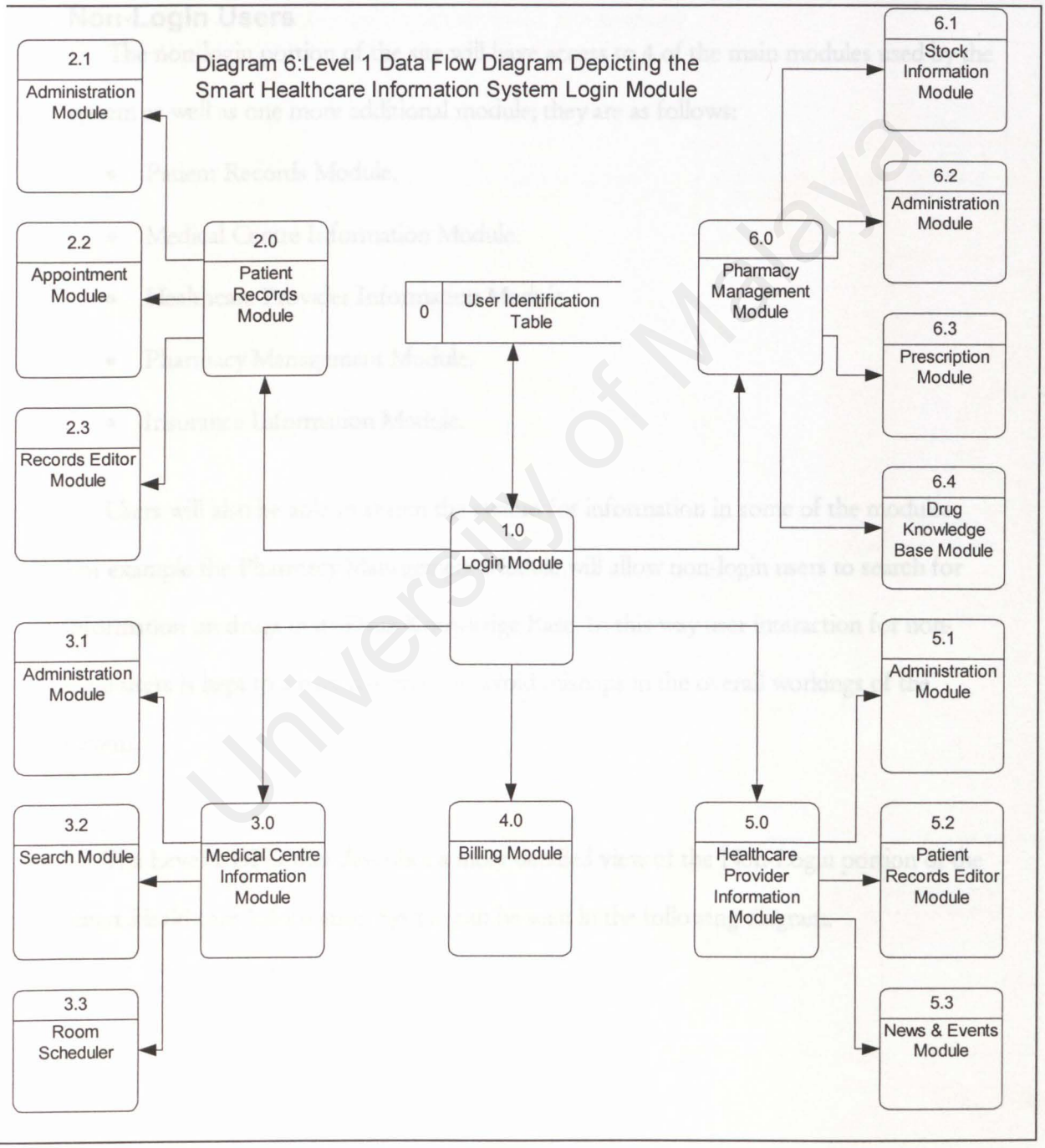


Diagram 6:Level 1 Data Flow Diagram Depicting the Smart Healthcare Information System Login Module

Non-Login Users

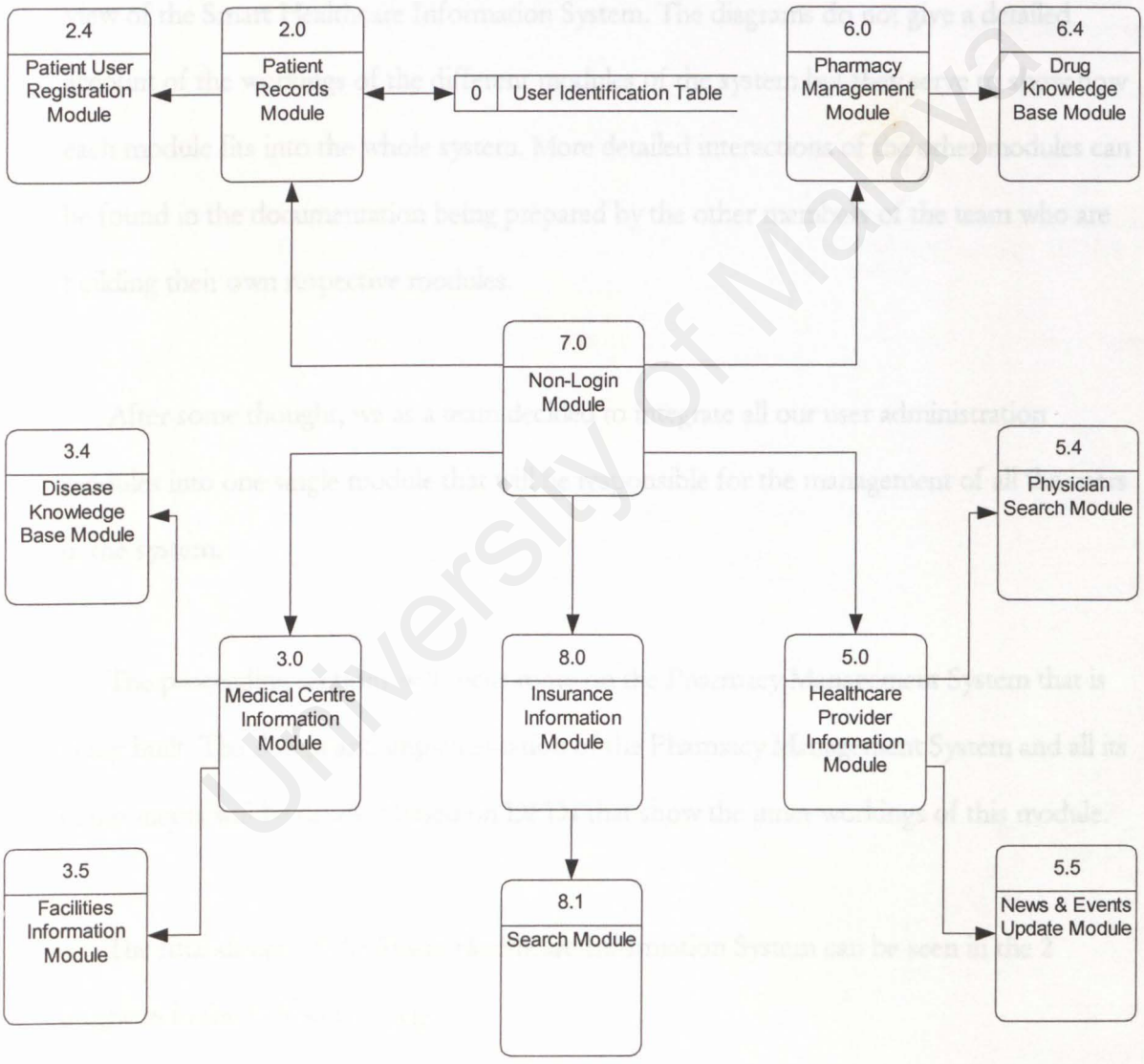
The non-login portion of the site will have access to 4 of the main modules used by the system as well as one more additional module; they are as follows:

- Patient Records Module.
- Medical Centre Information Module.
- Healthcare Provider Information Module.
- Pharmacy Management Module.
- Insurance Information Module.

Users will also be able to search the system for information in some of the modules. For example the Pharmacy Management Module will allow non-login users to search for information on drugs in its Drug Knowledge Base. In this way user interaction for non-login users is kept to a bare minimum to avoid mishaps in the overall workings of the system.

The Level 1 DFD that describes a more detailed view of the Non-Login portion of the Smart Healthcare Information System can be seen in the following diagram.

Diagram 7: Level 1 Data Flow Diagram Depicting the Smart Healthcare Information System Non-Login Modules



Summary

The design of the system in the preceding diagrams only show an overall and general view of the Smart Healthcare Information System. The diagrams do not give a detailed account of the workings of the different modules of the system but they serve to show how each module fits into the whole system. More detailed interactions of the other modules can be found in the documentation being prepared by the other members of the team who are building their own respective modules.

After some thought, we as a team decided to integrate all our user administration modules into one single module that will be responsible for the management of all the users in the system.

The proceeding sections will focus more on the Pharmacy Management System that is being built. The design and implementation of the Pharmacy Management System and all its components will be detailed based on DFDs that show the inner workings of this module.

The final design of the Smart Healthcare Information System can be seen in the 2 diagrams in the following 2 pages.

Diagram 8:Final Design of the modules in Smart Healthcare Information System for Login Users

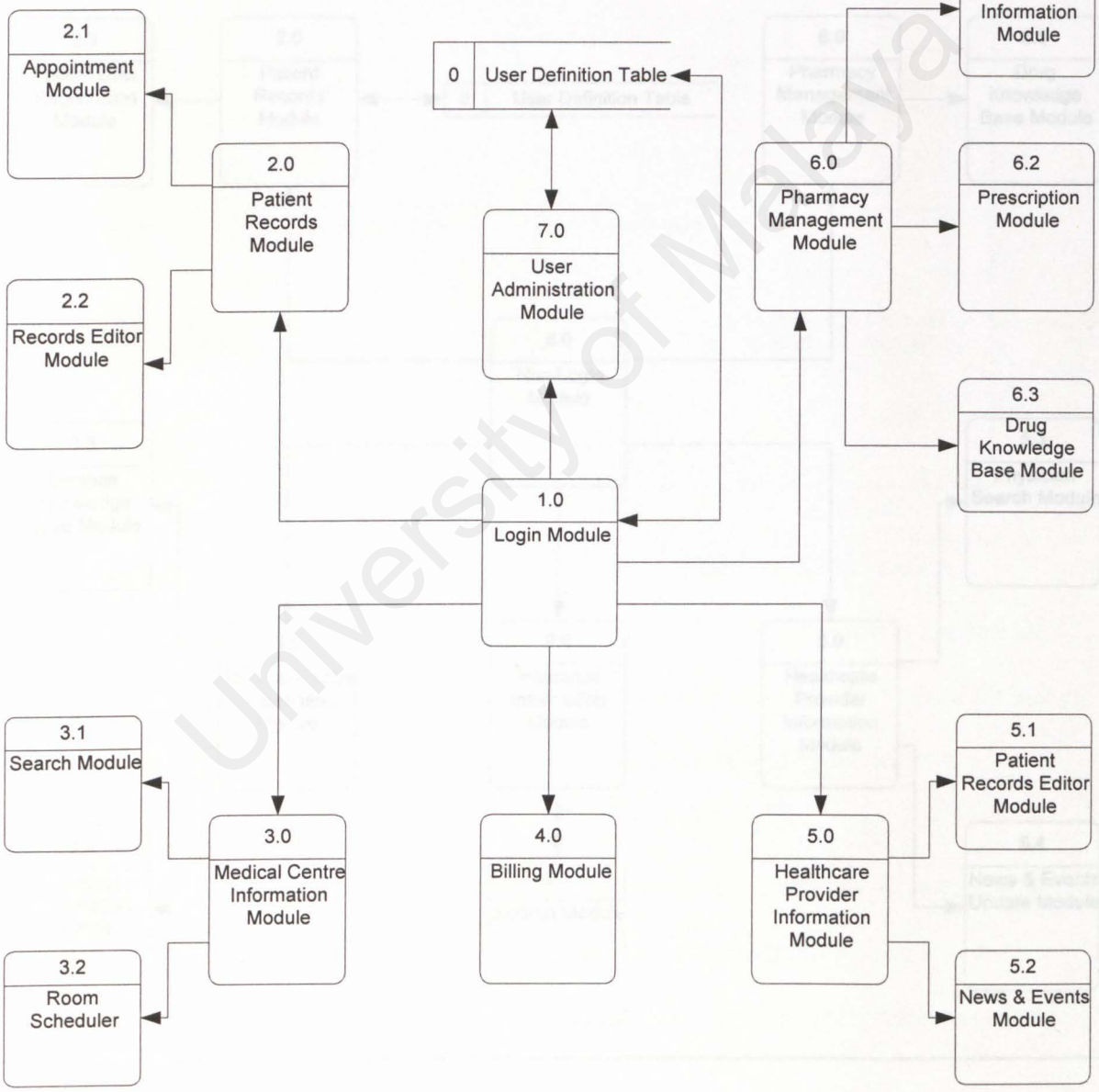
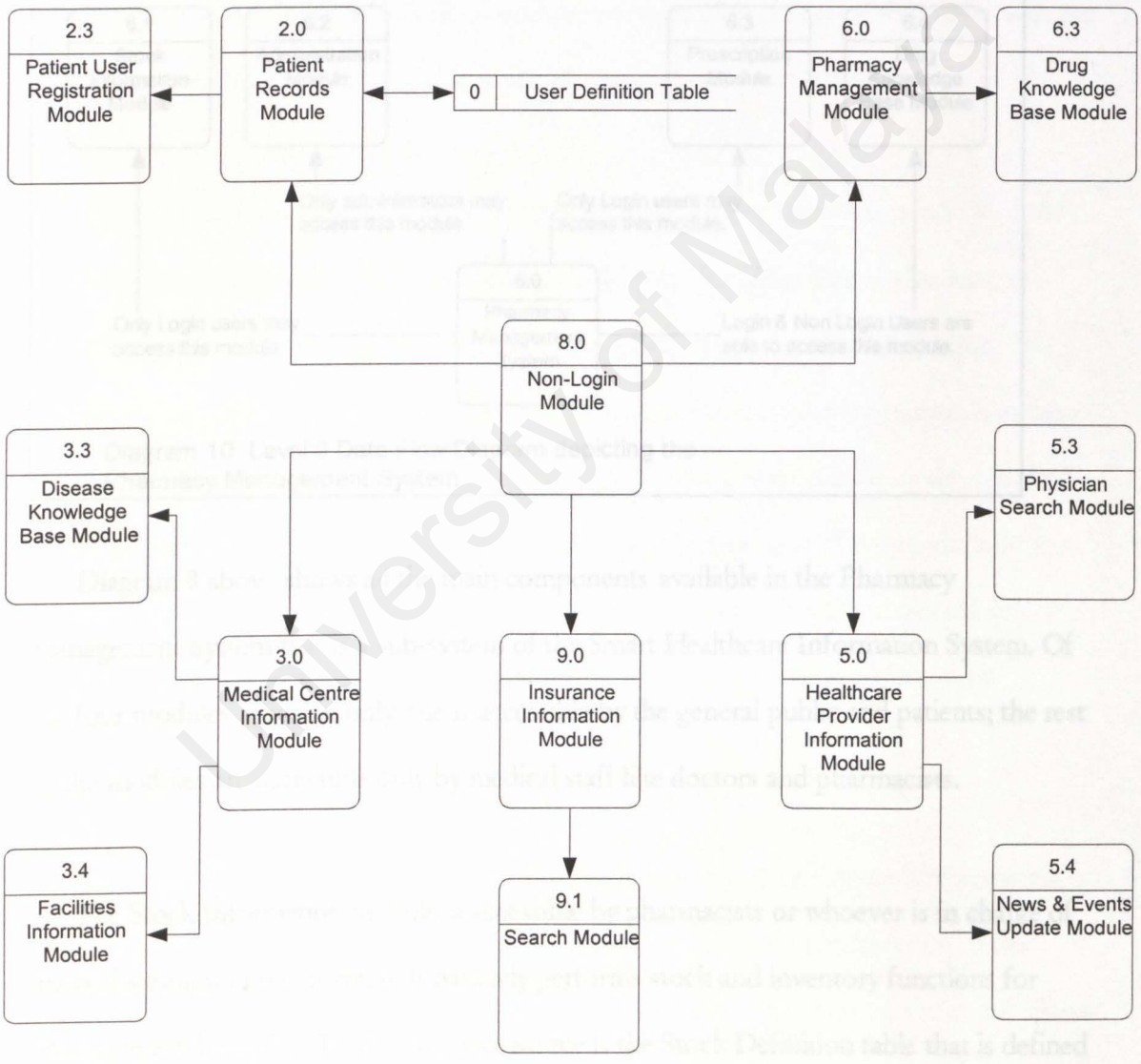


Diagram 9: Level 1 Data Flow Diagram Depicting the Smart Healthcare Information System Non-Login Modules



Detailed Design of the Pharmacy Management System

Overview

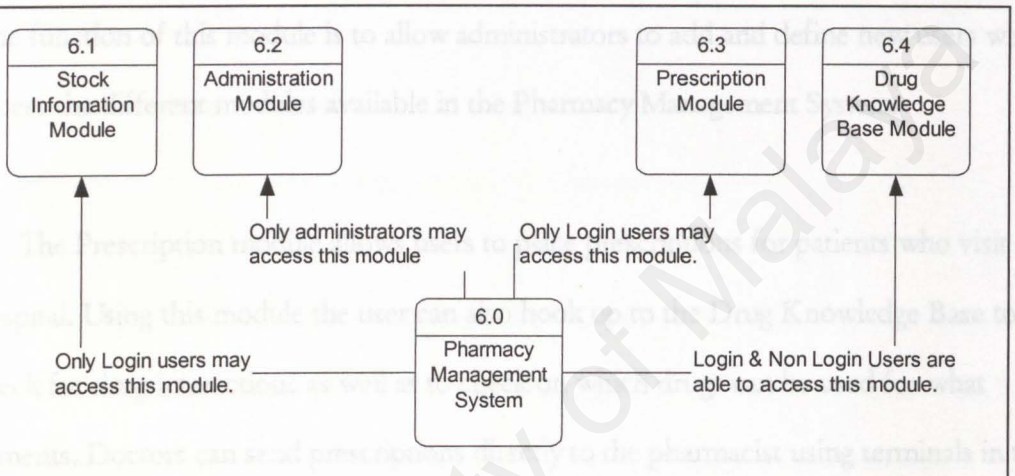


Diagram 10: Level 0 Data Flow Diagram depicting the Pharmacy Management System

Diagram 8 above shows all the main components available in the Pharmacy Management System that is a sub-system of the Smart Healthcare Information System. Of the four modules available only one is accessible by the general public and patients; the rest of the modules are accessible only by medical staff like doctors and pharmacists.

The Stock Information module is accessible by pharmacists or whoever is in charge of medical supplies in the hospital. It basically performs stock and inventory functions for pharmaceutical supplies. Its primary data source is the Stock Definition table that is defined in the main database of the Smart Healthcare Information System.

The Administration module is open to system administrators and human resource staff. The function of this module is to allow administrators to add and define new users who can access the different modules available in the Pharmacy Management System.

The Prescription module allows users to place prescriptions for patients who visit the hospital. Using this module the user can also hook up to the Drug Knowledge Base to check for drug interactions as well as to check on which drugs can be used for what ailments. Doctors can send prescriptions directly to the pharmacist using terminals in their offices. The pharmacist can check for the prescription online after that he or she can then fulfill the prescription manually based upon the doctors' suggestion. If the medication has negative interactions with one another then the pharmacist can alert the physician and suggest an alternative. The billing is done automatically and the cost is added to the patient's bill.

The Drug Knowledge Base is a central repository that contains information about each drug that is used in the hospital. Other drug information can also be placed here, based upon the administrator's discretion. The information can be accessed by all users and allows people to check on different drugs and find out what each drug can do as well as their side effects. The Knowledge Base is especially useful to pharmacists who can check for drug

interactions here. The Drug Knowledge Base is stored in a table that is defined in the main database of the Smart Healthcare Information System. The Drug Knowledge Base basically consists of two main parts:

- Values contained in the database table.
- Text documents containing descriptive information.

All these modules make up the Pharmacy Management System and each module performs specific functions unique to itself. Every module with the exception of the Drug Knowledge Base has its own defined set of users to limit access in the Pharmacy Management System.

The Stock Information Module

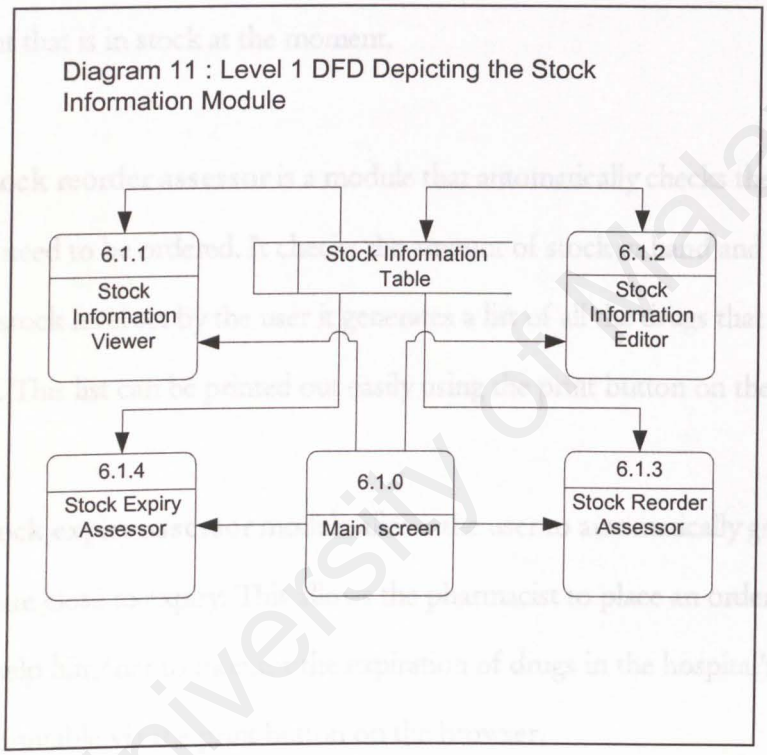


Diagram 11 : Level 1 DFD Depicting the Stock Information Module

The Stock Information Module is the main access point for hospital staff to access information regarding pharmaceutical stock in the hospital. It also allows users to edit details that are already present in the database.

The **stock information viewer** allows the pharmacist to view what drugs are available in storage. It also allows the user to check on other details; for example pricing as well as the amount that is in stock at the moment.

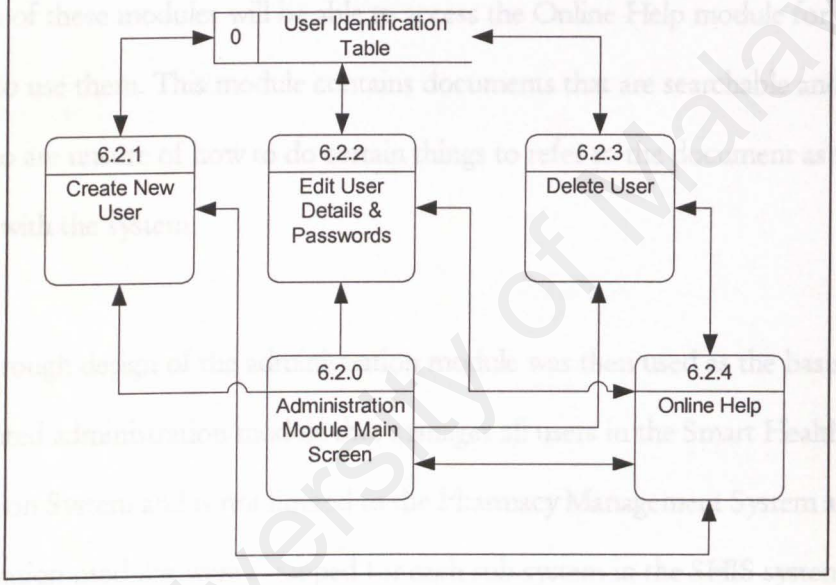
The **stock reorder assessor** is a module that automatically checks the database for drugs that need to be ordered. It checks the amount of stock in hand and based on the minimum stock level set by the user it generates a list of all the drugs that need to be ordered in. This list can be printed out easily using the print button on the user's browser.

The **stock expiry assessor** module allows the user to automatically generate a list of drugs that are close to expiry. This allows the pharmacist to place an order for fresh stock as well as help him/her to monitor the expiration of drugs in the hospital's inventory. This list is also printable via the print button on the browser.

The **stock information editor** allows the user to edit the stock information in the relevant tables. It allows the user to update all the columns that are in the Stock Definition Table. The most important aspect of this module is its ability to change pricing information if price shifts occur.

The Administration Module

Diagram 12 : Level One Data Flow Diagram Depicting the Pharmacy Management System Administration Module



The Administration Module comprises of a number of modules that help the system administrator to organize the users who are able to access the system. From the main screen the administrator can access 3 main functions that directly access the User Definition Table; they are as follows:

- User Creation Module – ASP module that allows the creation of new users.
- User Details Editor – ASP module that allows the administrator to edit user details.

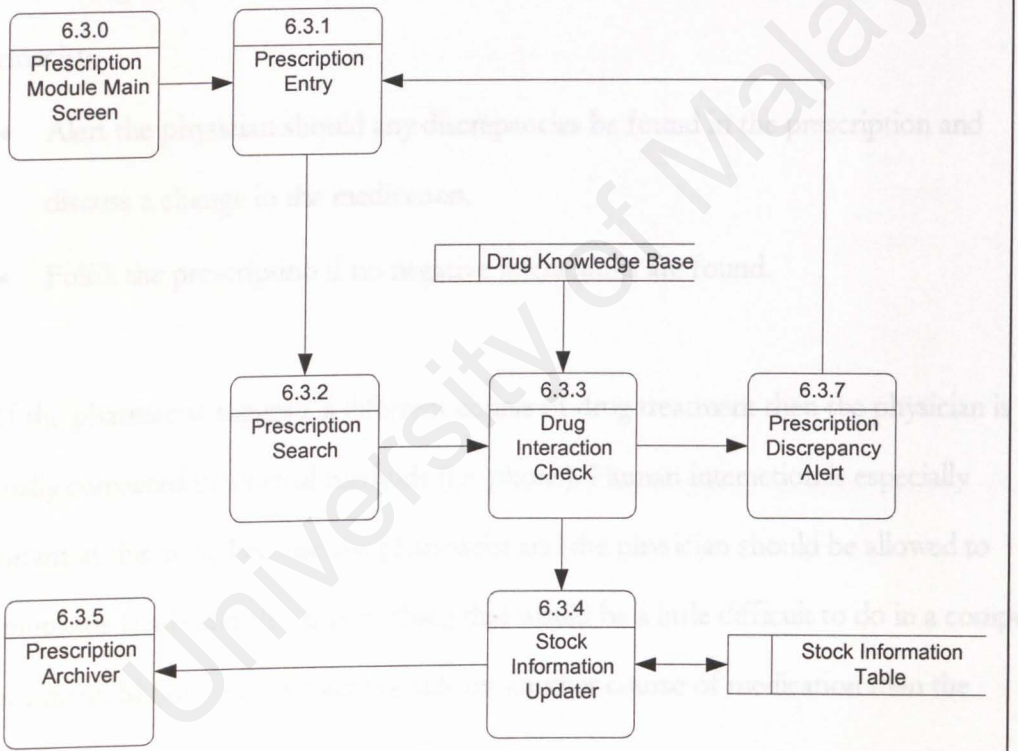
- User Deletion Module – ASP module that allows the removal of users from the User Definition Table.

Each of these modules will be able to access the Online Help module for instructions on how to use them. This module contains documents that are searchable and would allow users who are unsure of how to do certain things to refer to the document as they are working with the system.

This rough design of the administration module was then used as the basis of a consolidated administration module that manages all users in the Smart Healthcare Information System and is not limited to the Pharmacy Management System alone. Separate administration modules were scrapped for each sub system in the SHIS system as an integrated one would serve the needs of everyone better.

The Prescription Module

Diagram 13 : Data Flow Diagram depecting the Prescription Module of the Pharmacy Management System



The Prescription Module was designed to be used by both physicians and the pharmacists in a hospital. The system has two main parts. The first part of the system allows prescriptions to be made by the physician in the Prescription Entry module. Once that is done the pharmacist can check for the prescription anytime.

The pharmacist can then check for drug interactions after looking at the prescription. He/she does this by referring to the information contained in the Drug Knowledge Base. Once the drug interactions have been checked then there are two options available to the pharmacist:

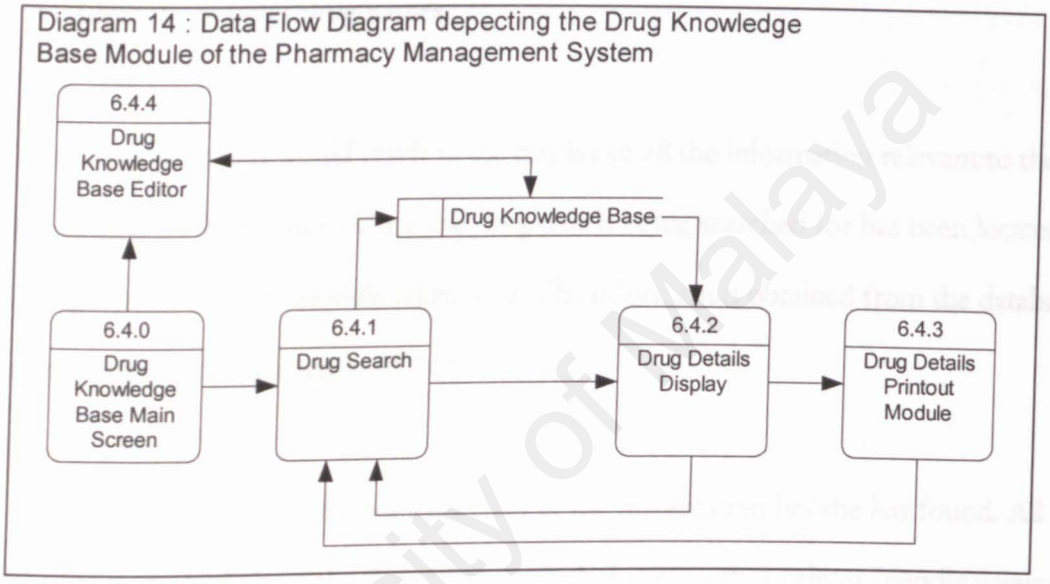
- Alert the physician should any discrepancies be found in the prescription and discuss a change in the medication.
- Fulfill the prescription if no negative interactions are found.

If the pharmacist suggests a different course of drug treatment then the physician is manually contacted by normal methods (i.e. phone). Human interaction is especially important at this stage because the pharmacist and the physician should be allowed to communicate freely and this is something that would be a little difficult to do in a computer environment. Should the physician decide on another course of medication then the pharmacist just updates the prescription.

After the prescription has been fulfilled, the system automatically deletes the prescription request. This means that the user would not have to worry about keeping the prescription list up to date. A temporary store of past prescriptions are then archived so that the user will be able to refer to it should the need arise later.

The Drug Knowledge Base Module

Diagram 14 : Data Flow Diagram depecting the Drug Knowledge Base Module of the Pharmacy Management System



The Drug Knowledge Base module is the only module in the Pharmacy Management System that is accessible to both regular users (i.e. patients and other web users) and hospital staff (i.e. pharmacists and doctors).

The Drug Knowledge Base basically consists of 4 main modules that handle the functions that need to be accessed by the users of the system. The Drug Search module basically allows the user to run a search of the database that contains all the information regarding drugs. In this initial implementation of the system; the search module would be limited to searches with the following keywords:

- Scientific names of drugs.
- Over the counter drug names.

A successful search would result in the display of all the information relevant to the search conducted by the user. Once the drug that is being searched for has been located; the Drug Details Display module takes over. The information obtained from the database is then formatted and displayed in HTML.

Should the user want to have a hard copy of the information he/she has found. All he/she has to do is click on the dynamic hyperlink that formats a printer friendly version of the page so that the user can print it out.

The last module in this sub-system is the Drug Knowledge Base Editor. This module is accessible only by users who are authorized to edit the table that contains information regarding the drug knowledge base. The editor is capable of adding, removing and modifying information regarding drugs in the knowledge base. The Drug Knowledge Base Editor also allows the user to edit the text files where the description of each drug is stored.

Summary

Because of the changes made after the design of the modules. The final version of Pharmacy Management System only contains three modules. This is because the original idea of having separate Administration modules for each sub system was scrapped and a single Administration module was developed to replace all the other proposed modules. The three modules that have been kept in the Pharmacy Management System are as follows:

- The Stock Information Module.
- The Prescription Module.
- The Drug Knowledge Base Module.

Therefore the Data Flow Diagram that depicts an overall view of the Pharmacy Management System can be seen in the DFD below.

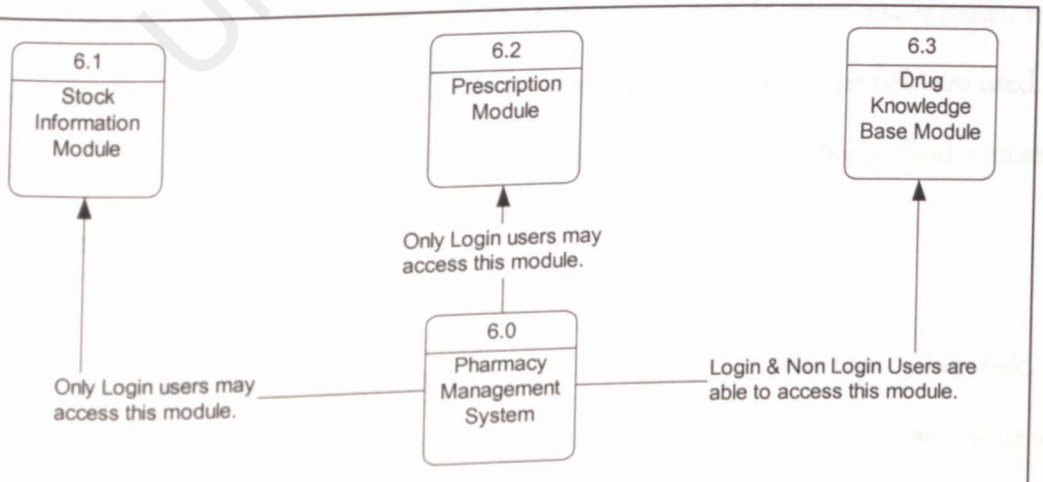


Diagram 15: Level 0 Data Flow Diagram depicting the Pharmacy Management System

Database Design

Design of the User Definition Table

The User Definition Table basically contains information that would allow the system to identify users as they come into the system. Using the information in the table users' access to the system as a whole can be restricted easily. The four fields that will be used in the User Definition Table are as follows:

Field Name	Data Type
User_ID	int
User_Name	char(15)
Full_Name	varchar(50)
Password	char(8)
User_Code	int

The User_ID field is of type integer and would basically contain a uniquely assigned identification number that is assigned by the system to the user. The User_Name field is of type character and contains a maximum of 15 characters in it. It stores a user name, which the user selects for himself/herself. Both the User_ID and User_Name field are used as primary keys to uniquely identify each user in the system. The Full_Name field contains the full name of the users in the system.

The password field is self-explanatory and is of type integer. The User_Code field determines access privileges to the entire system and is of type character as well. Due to deletion anomalies the definition for each user group is stored in a separate table called the

Group Definition table. The Group Definition table contains two fields that describe the user groups in the entire system. The other fields denote the access privileges accorded to a particular user group with "Y" signifying access privilege to each section.

Field Name	Data Type
User_Code	int
Group_Definition	varchar(100)
Patient_Records	char(1)
Medical_Center_Info	char(1)
Healthcare_Provider	char(1)
Pharmacy_Info	char(1)
Insurance	char(1)
Billing	char(1)

The User_Code field is of type integer and contains codes that signify which group a particular user belongs to. The Group_Definition field contains the full text description of the user groups in the system. The remaining fields denote access privileges in the system. For example a user group with access to the Patient Records portion of the system will have a "Y" filled in the Patient_Records field and so on and so forth.

All of the user groups contained in the Smart Healthcare Information System can be seen in the following table:

Group Code	User Group
0	System Administrators
1	Patient
2	Nurse
3	Lab technician
4	Medical Center Information System Administrators
5	Physicians (Doctors)
6	Healthcare Provider Information System Administrators
7	Pharmacists
8	Pharmacy Stock Information Administrators
9	Insurance Information System Administrators
10	Billing Information System Administrators
11	Cashiers

The User Definition table is the main shared data source among every single module in the Smart Healthcare Information System. It determines user access for every part of the system; which makes it integral that this table be made as secure as possible.

Design of the Stock Definition Table

The Stock Definition table contains information that directly relates to drug stock in the hospital. The fields that would accurately capture stock information are stated in the table below.

Field Name	Data Type
Stock_ID	varchar(20)
Name_Scientific	varchar(50)
Name_Generic	varchar(50)
Reorder_Level	float
Quantity_Stock	float
Quantity_Per_Unit	int
Quantity_Total	int
Dosage_Per_Unit	float
Unit	char(10)
Price	money
Expiry_Date	datetime

The Stock_ID field refers to unique identification codes for each drug that is available in the hospital. This identification ID is a code assigned by the National Drug Control Agency and each drug has a unique value associated with it. The Stock_ID field is of type varchar, which means that the length of characters that can fit into this column is of variable length with a maximum of 20 characters.

The Name_Generic column refers to the name of the medication and is of type varchar. A maximum size of 50 characters is set so as to accommodate drugs with long names. This

field together with the Stock_ID field serves as an identifier for a drug in the hospital's database.

The Name_Trade field contains the trade name for a drug. For example Panadol is one of the brand names for the drug paracetamol. Therefore the name Panadol will be entered into this field. The maximum number of characters that can be entered into this field is 50 characters.

The Reorder_Level field allows the pharmacist to specify a stock quantity value that determines when orders for a particular drug need to be made. Should the drug quantity fall below the value of the Reorder_Level field then the system can automatically take note of the drug and place it on a list of drugs that need to be ordered by the pharmacist.

The Quantity_Stock field reflects the amount of stock that is left in the hospital's inventory in terms of number of boxes. The unit of measurement for the drug is not specified here but rather is specified in another field, the Unit field. This is because different types of drugs have different units of measure. This is dependent on the physical form of the drug (i.e. solid, liquid, etc.).

The Quantity_Per_Unit field states the quantity of medication contained within a box or carton of medication. For example a pack of Panadol will have 20 tablets in a foil strip. And each box of Panadol contains 20 strips. A carton of Panadol will contain this number of boxes therefore the unit of measure will be 20×20 tablets of Panadol, which equals 400 tablets of Panadol. The value 400 will therefore be entered into this field..

The Quantity_Total field shows the total number of stock left in storage. The value contained in this field is derived by multiplying the Quantity_Stock field together with the Quantity_Per_Unit field. This field is used by the system to automatically keep track of the stock of a particular drug.

The Dosage_Per_Unit field contains information that is used to specify the dosage contained in every serving of a drug. Let's take Panadol as an example. If every tablet of Panadol contains 250 milligrams of Paracetamol then the value 250 will be entered into this field for the drug Panadol.

The values in the Unit field contain the units of measure used to quantify drug dosages in the hospital. An example of a unit of measure would be milligrams.

Pricing information of medication in storage is stored in the Price field. This field is of type money and denotes the price of each box of medication in the hospital. For example a value of 120.00 would mean that the price of a box of that particular drug would cost RM 120.00 each.

The Expiry_Date field contains the date when the particular shipment will expire. It is of type datetime and the value here can be updated using the Stock Information Editor everytime a new shipment of a drug is delivered.

After normalization the Stock Definition table was divided into 2 tables. The reason why this is done is to avoid loss of unit of measures should the details for any particular drug be deleted from the system. The two tables are as follows:

- The Stock Definition table.
- The Stock Unit table.

The fields that are contained in the Stock Definition Table are as follows.

Field Name	Data Type
Stock_ID	varchar(20)
Name_Generic	varchar(50)
Name_Trade	varchar(50)
Reorder_Level	float
Quantity_Stock	float
Quantity_Per_Unit	int
Quantity_Total	int
Dosage_Per_Unit	float
Unit_Code	varchar(10)
Price	money
Expiry_Date	datetime

The Unit field has been replaced by the Unit_Code field. The value in this field serves as a pointer to the Stock Unit Table which has two fields. The table in the following page shows the fields that are contained in the Stock Unit Table.

Field Name	Data Type
Unit_Code	varchar(10)
Unit_Description	varchar(20)

The Unit_Code field contains a specific code; which identifies the unit of measure that is used for a particular drug. The Unit_Description field contains the full text description of a unit of measure used in relation to the corresponding unit code. The data in this table is used in correlation to the Dosage_Per_Unit field.

Design of the Drug Knowledge Base Table

The Drug Knowledge Base is contained within a single table and just has three fields defined. The Drug Knowledge Base Table is treated as a unique data source separate from the Stock Definition Table. The fields in this table are described below:

Field Name	Data Type
Name_Trade	varchar(50)
Name_Generic	varchar(50)
Drug_Description_File	varchar(50)

The Name_Trade and Name_Generic database fields have the same values as the corresponding fields in the Stock Definition table. The reason why the fields are replicated in the Drug Knowledge Base Table is because the administrator might wish to include information on medication that may not be available in the hospital.

The Drug_Desription_File is filled with the name of the text file that contains the description of a drug. This file also includes all other details about the drug that the knowledgebase administrator wishes to include in the Drug Knowledge Base.

Design of the Medication Table

The Medication Table is a section of the database that is used to store the prescription history of the patient's who come into the hospital. Information regarding the drugs prescribed for every single patient is entered into this table whenever the pharmacist updates the stock information in the Prescription module. The fields in the Medication table are as follows:

Field Name	Data Type
Patient_ID	int
Date_Prescribed	datetime
Quantity	int
Stock_ID	varchar(20)
Attend_PhyID	int

The Patient_ID field contains the patient's ID Number and the Date_Prescribed field stores the date that the prescription was made. Both these fields are used to identify each person who has been prescribed drugs in the hospital. The Attend_PhyID field contains the attending physician's ID number for easy identification of the doctor who made a particular prescription.

Drug information is stored in the Quantity field and the Stock_ID field. The Quantity field shows the quantity of a particular drug that was prescribed to the patient whereas the Stock_ID field contains the unique codes that identifies each drug in the pharmacy's inventory.

Coding and Unit Testing

Coding

Coding was done over a one-month period and 3 major web-scripting languages were used together with HTML to code the relevant web pages that would eventually make up the Pharmacy Management System.

A more traditional programming paradigm was used to code the web application. Because of unfamiliarity with the languages used, structured programming was used to create the modules for the Pharmacy Management System. As development progressed, code that could be reused was separated out from the different modules and placed into shared libraries. For example database connection code was placed into a file and shared out over all the modules that need to make database connections using Server Side Include directives. In addition to that, HTML templates were created and shared out among some modules so that dynamic content can be produced via Active Server Pages (ASP), VBScript and JavaScript.

3 main modules were developed over that one-month period. The first module to be developed was the Prescription module followed by the Drug Knowledge Base module and last but not least the Stock Information module was coded. Each module makes use of a number of different functions that are made available in ASP and VBScript.

The most notable function that was used prominently in most of these modules was the database connection capability of ASP that simplifies the creation of connections to SQL Server. Using ADODB objects, database access to the Smarthealth database could be done easily by making calls to ODBC drivers that are a standard part of Microsoft Windows. Once a connection was made to the database data could be retrieved and manipulated easily using recordset objects and SQL statements.

Another major function called in all three modules was the VBScript TextStream object that allowed the manipulation of text data. The TextStream object and the Replace function was used also to implement HTML templates that could be called by a number of pages to generate HTML content that is dynamic in nature. This allowed HTML templates to be shared out over a few pages thus reducing the work needed to make cosmetic changes to a particular section of the Pharmacy Management System.

File system access for creating and moving files around the Windows environment through the web was made possible by VBScript's File object that communicates directly with Windows' file system. It allowed the creation of code that would enable the system to create text files to store data for the Prescription module and also the movement of files for archival purposes in the system that was being developed.

Subroutines and functions were also used extensively so that the code created would be modular and more readable. The advantage of using shared functions for a particular web page was the ability to use these pieces of code as many times as was needed in a page being developed. This method of programming also simplified debugging procedures since each significant piece of code resided in its own subroutine or function.

Unit Testing

The test cycle in the development phase was conducted during and after the completion of each module of the Pharmacy Management System. The four major reasons why testing was done are as follows:

- Bug detection.
- Bug eradication.
- Bug tracking.
- Regression testing.

Each unit/module in the Pharmacy Management System was tested separately to ensure that no bugs were present in each of them. Test values were entered into each form that accepts data from the user to test user input into the system. Logical dependencies were

also tested; test values were included in the code itself to test every important logical statement that would access a particular segment of code.

Logical dependencies were gotten rid of at the module level by taking a few precautionary steps. Dependencies and their effects were determined by looking at data and logical structures that exist in the code being developed. Code segments were rearranged and loop structures were checked as thoroughly as possible to ensure that all code segments could be reached. Both steps were taken also to ensure that there was no possibility of infinite loops in the system. This especially must be done for shared libraries that are used in a number of pieces of code.

Software Integration and Testing

Software Integration

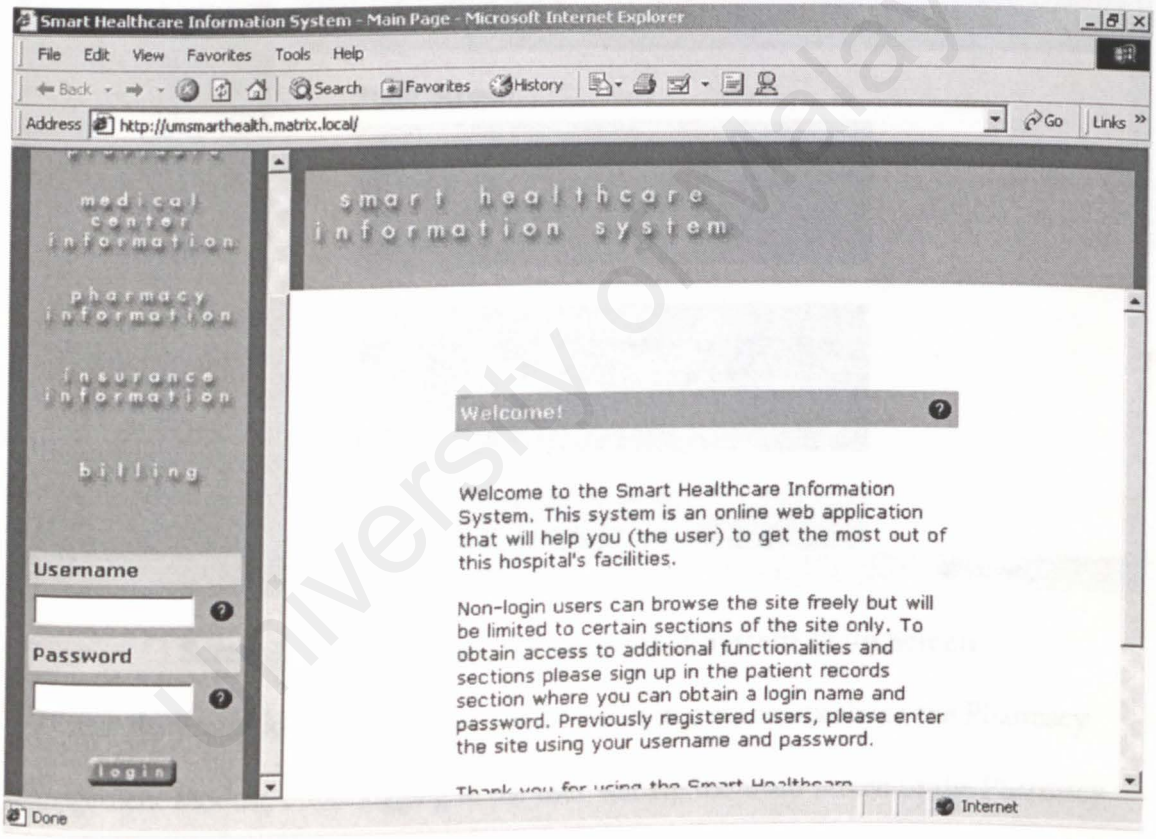
Not much effort was needed to integrate each module since each module is essentially independent of each other in terms of data and logical dependencies. Every module is accessed via HTML anchor tags placed into the navigation bars on the left of every web page in the system so this means that access to each module is easily implemented.

Testing

System testing also wasn't that hard to do as most of the testing was already done at the module level. Since each module is independent of each other; not much testing was required at the system level. The only kind of test that was done was to try to use the system as much as possible and to see if any bugs surfaced. Bugs that were found were placed into a list and testing would go on. Once enough tests were run, the list of bugs was used to fix all errors in the code. Testing was then conducted one final time to make sure that no other bugs emerged because of the bug fixes applied earlier. If any errors were found, earlier versions of code that were back-upped earlier could be put back into place to ensure system integrity.

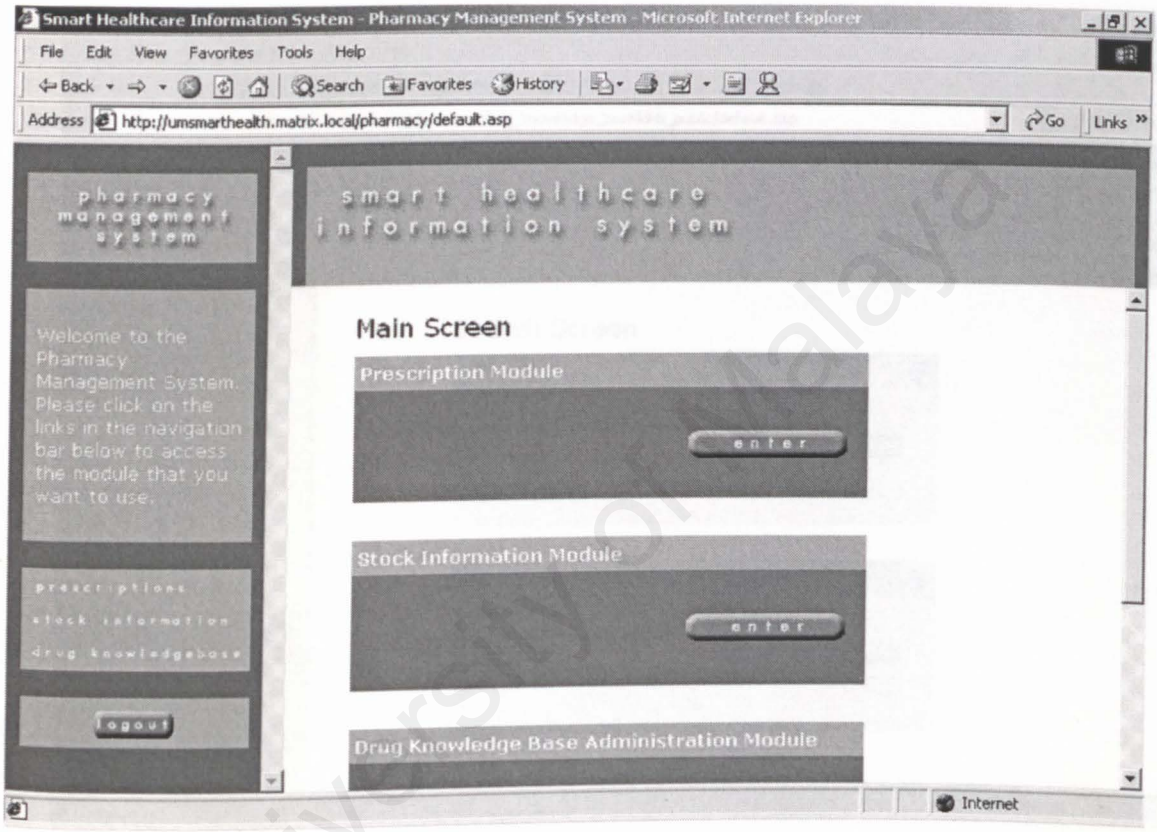
System Description

Login and Non-Login Users



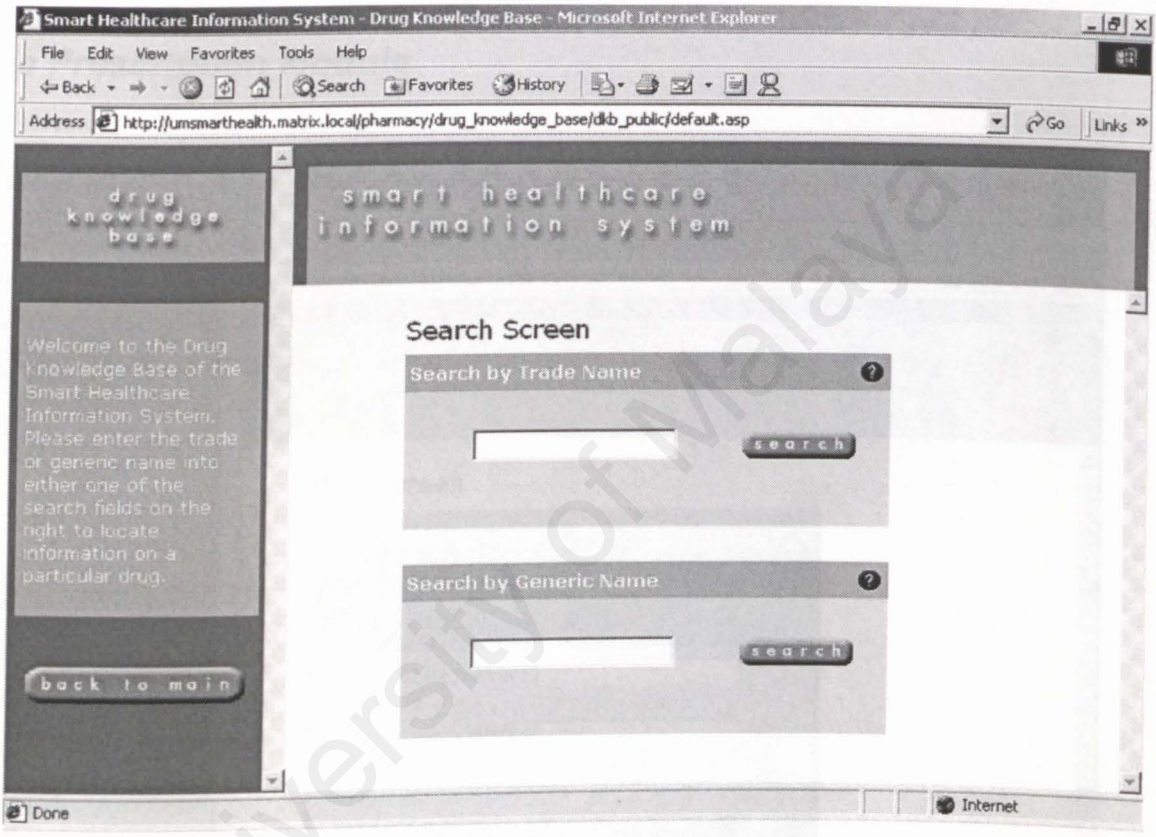
Screenshot 1: Login Screen

Logging into the system is done easily by entering your username and password into the two fields that can be found on the left navigation bar on the main screen.



Screenshot 2: Pharmacy Management System Main Screen

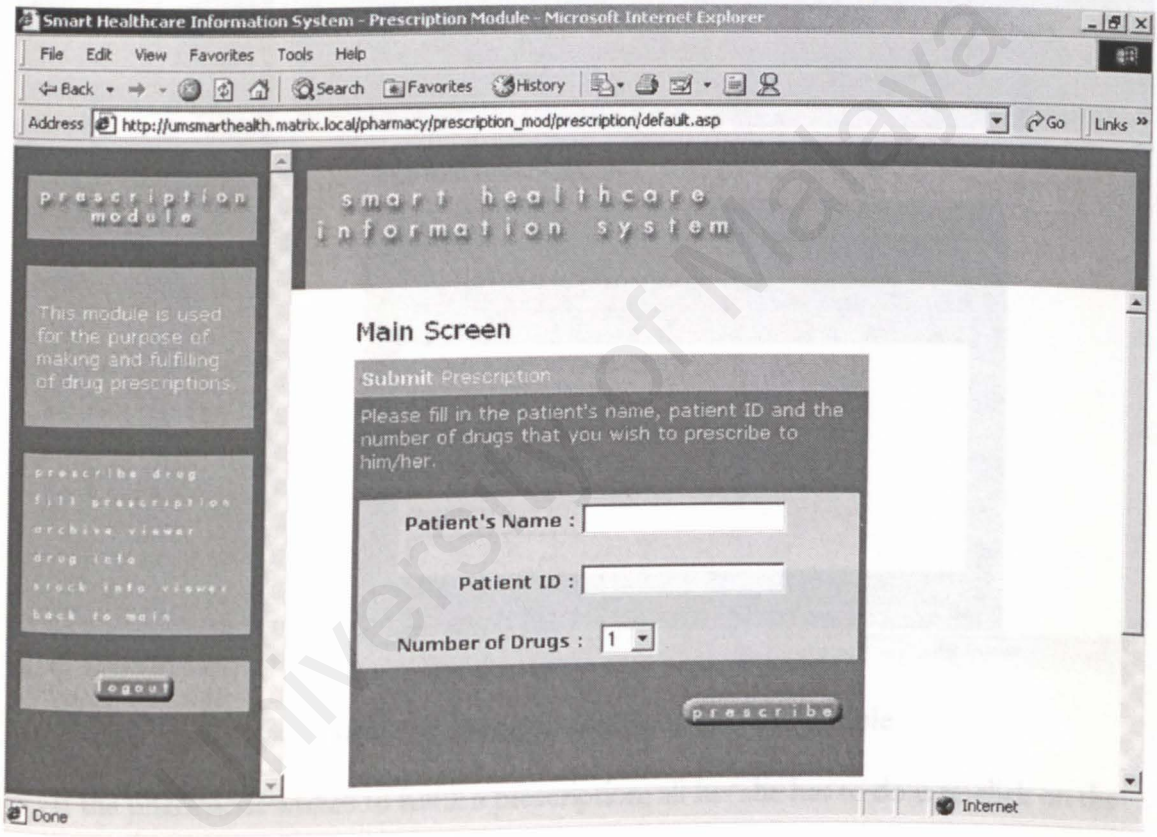
Once the user is logged in; he/she will be redirected automatically to the Pharmacy Management Module. After logging in the user will see the main screen of the Pharmacy Management System which allows a registered user (the pharmacist) to access all the different functionalities available in the system.



Screenshot 3: Drug Knowledge Base Search Screen

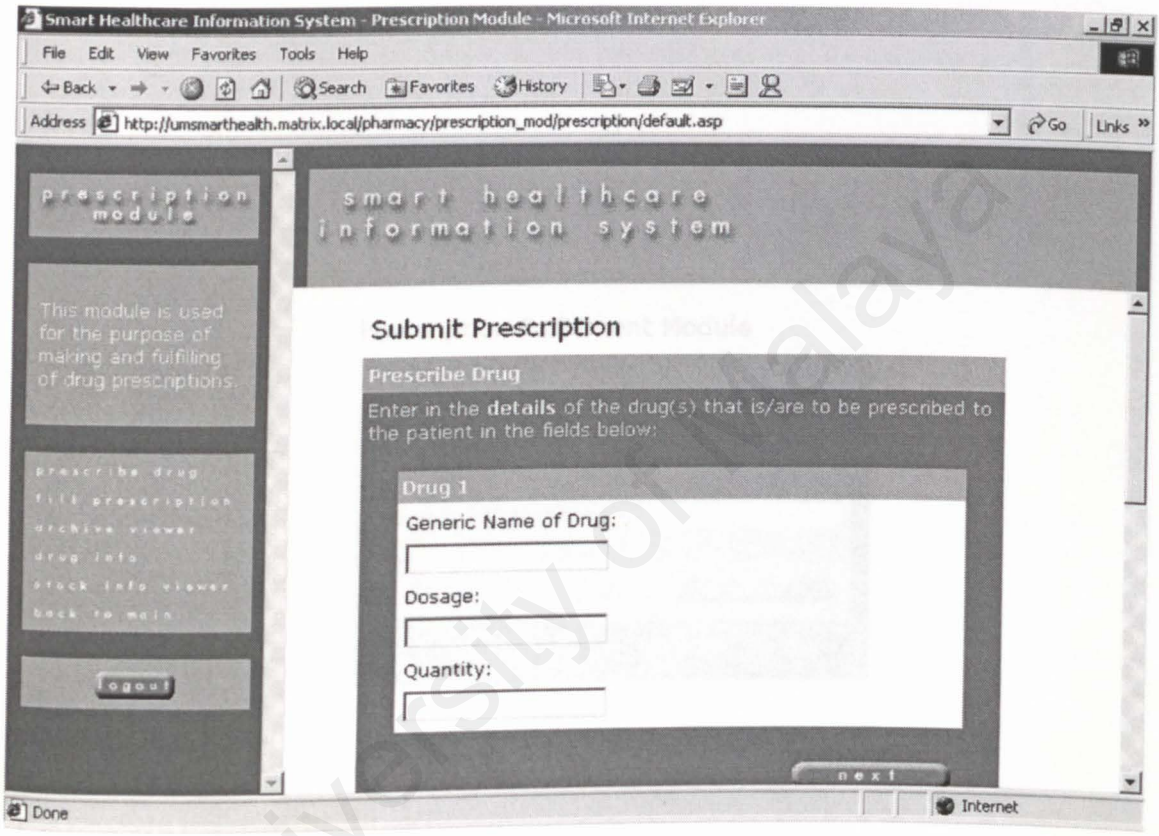
Non-login users can access the Drug Knowledge Base by clicking on the Pharmacy Information link on the navigation bar. Drug information can be found by entering either the trade or generic name for a drug into the form presented on screen (Screenshot 3). After entering the search parameter, the system will conduct a search of the database and if the information is found then it is displayed to the user; otherwise a message telling the user that the drug information requested was not found is displayed.

The Prescription Module



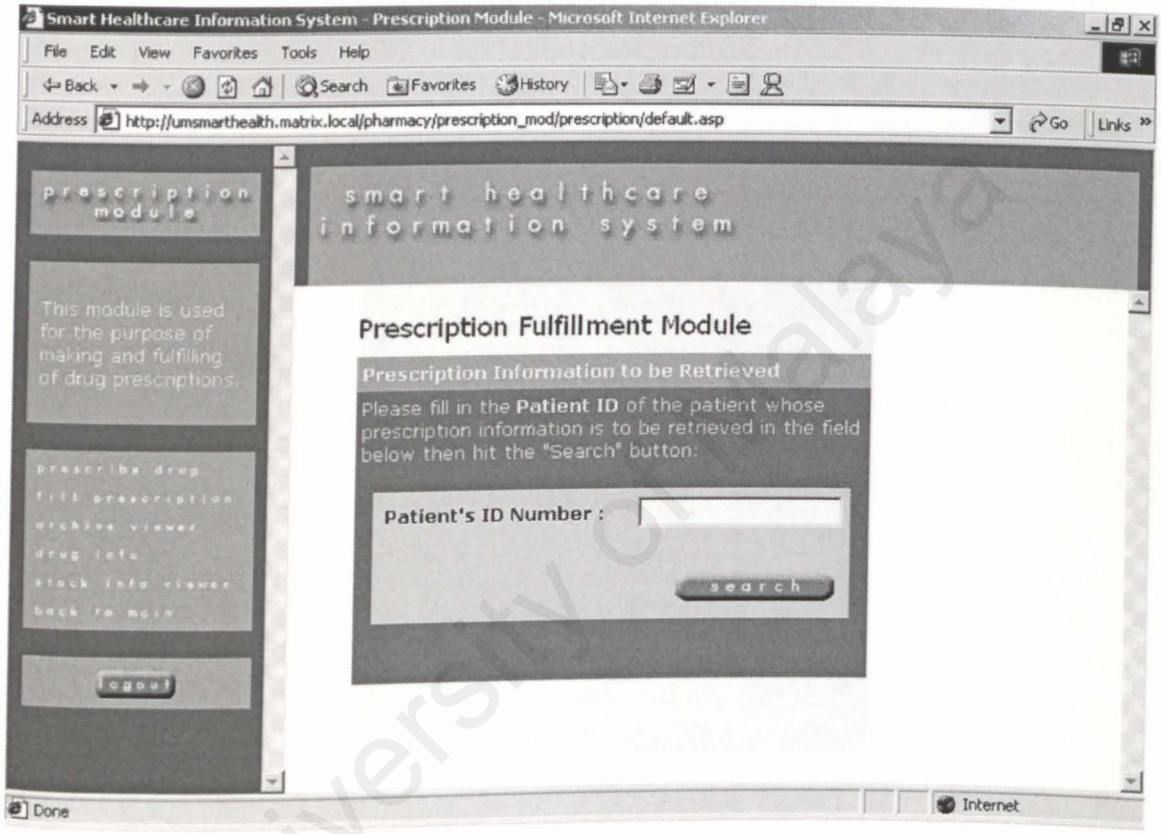
Screenshot 4: Prescription Module Main Screen

Upon entry into the Prescription module, the first screen that greets the user is the prescription submission screen where the physician can enter in his/her prescription for a particular user. The physician has to enter in the patient's name, the patient ID number and the amount of drugs being prescribed to him/her. Once that is done then the physician must enter the details of the drugs being prescribed to the patient.



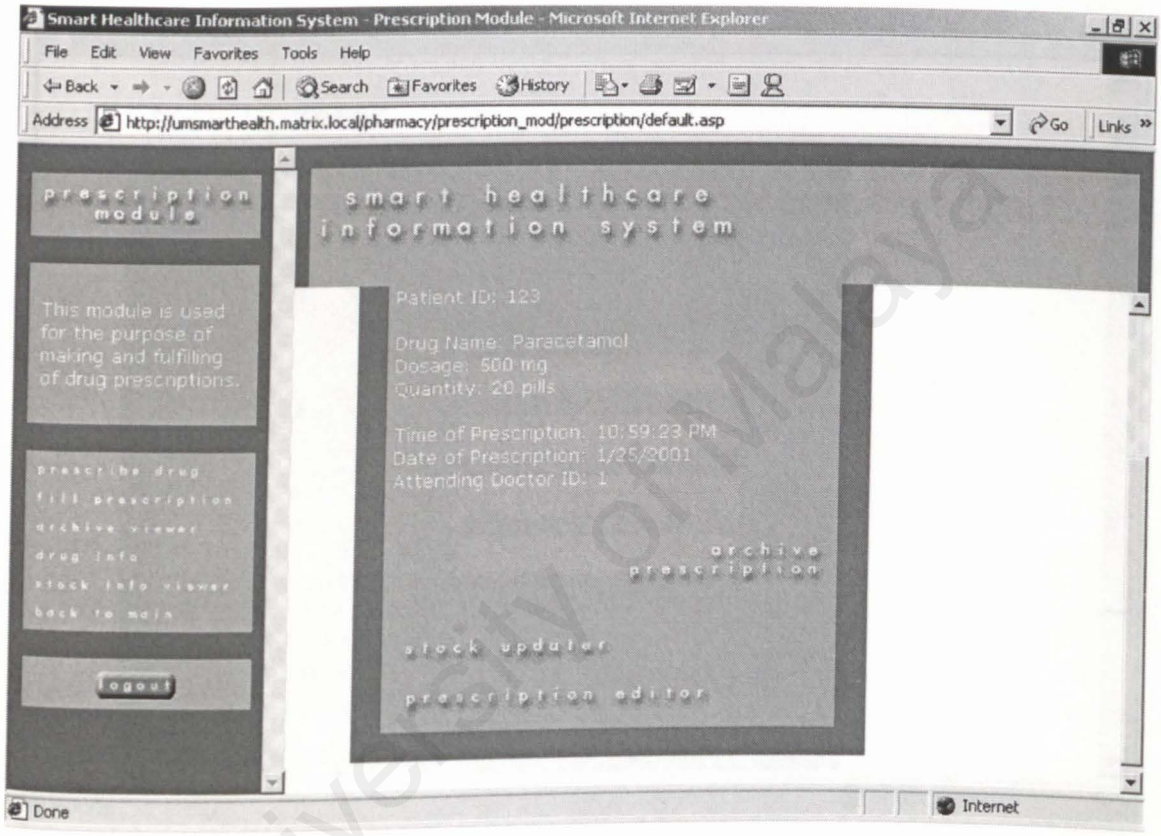
Screenshot 5: Prescription Submission Module

If the pharmacist wishes to fulfill a prescription; all he/she has to do is to click on the prescription fulfillment module link and he/she will be taken to the Prescription Fulfillment screen.



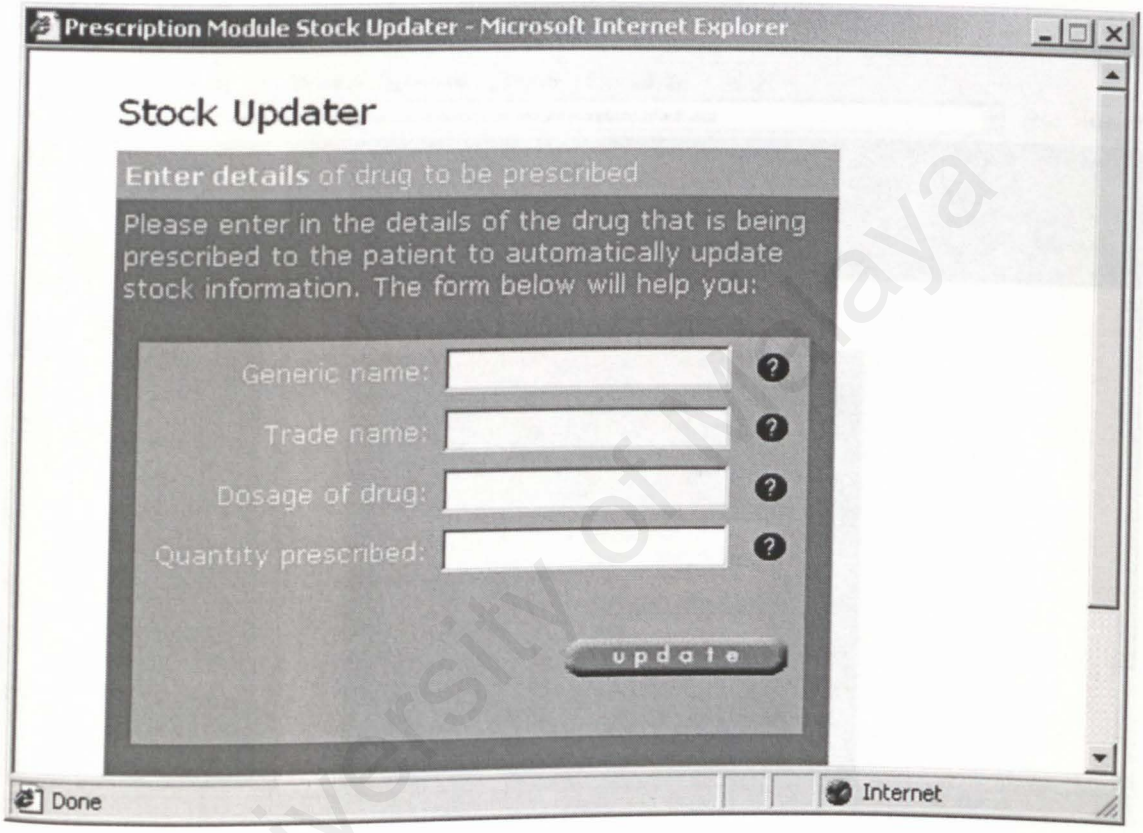
Screenshot 6: Prescription Fulfillment Module

The first screen that appears before the user allows the pharmacist to fill in the patient's ID number in the form presented to search for the prescription that was made earlier by the doctor. You can view the screen in the image on the following page (Screenshot 6).



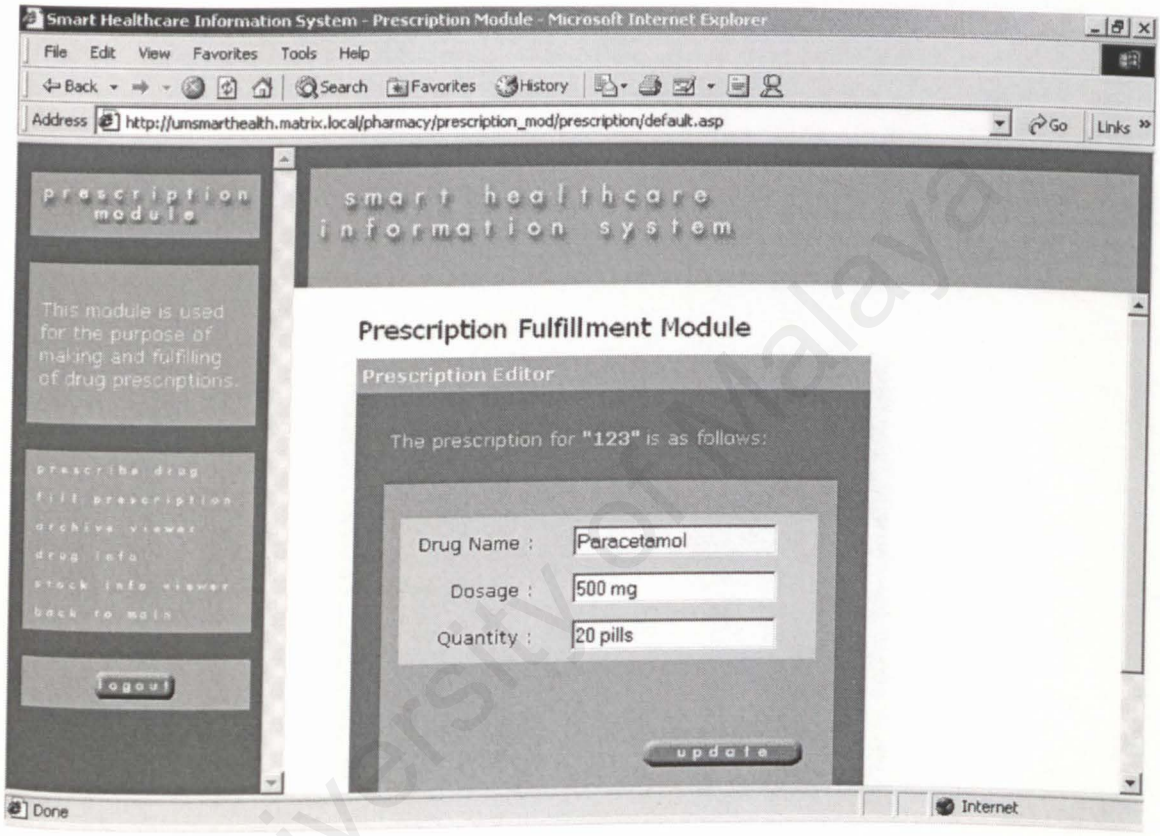
Screenshot 7: Prescription Fulfillment Module

Once the prescription has been found, it will be displayed to the pharmacist so that he/she can fulfill the prescription to the patient. The pharmacist then has the option of archiving the prescription, launching the stock updater or editing the prescription.



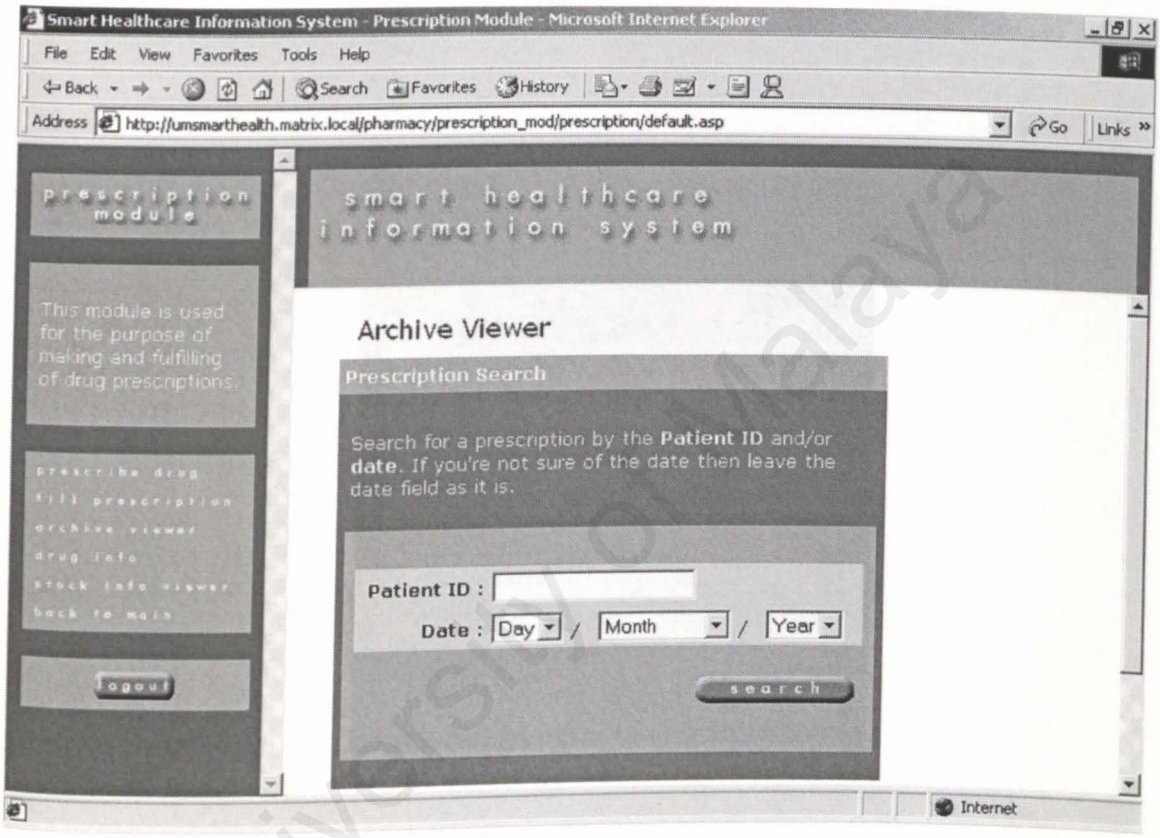
Screenshot 8: Stock Updater

The Stock Updater is used to update the stock information in the database and the pharmacist is required to use this module so that stock information can be kept up to date during the day-to-day operation of the pharmacy. This module also places an entry into the database so that the prescription history of the patient can be maintained easily.



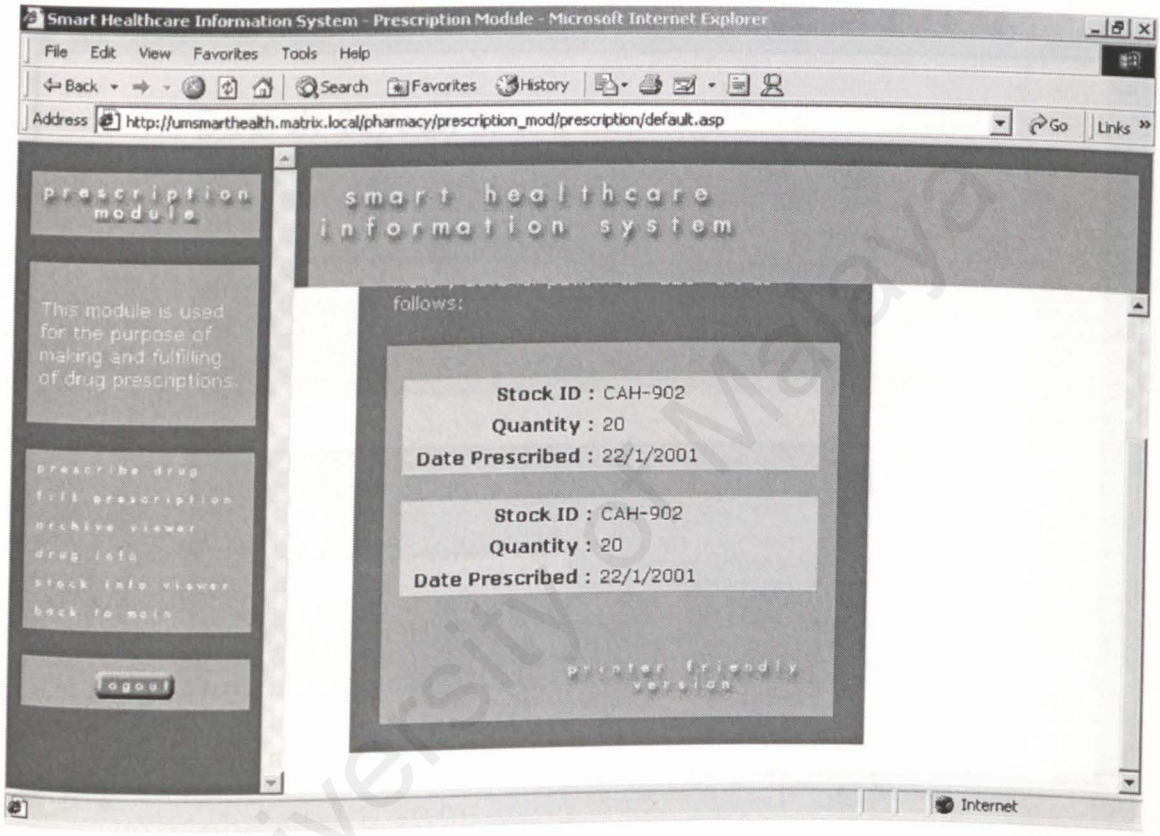
Screenshot 9: Prescription Editor

The Prescription Editor allows the pharmacist to edit the prescription should there be changes in the prescription. This is only done if the pharmacist sees it fit to change the prescription after consultation with the doctor. Once the changes have been done, the pharmacist can update the prescription easily using this simple web based utility. Once the stock information has been updated and the prescription clarified, the pharmacist then needs to archive the prescription for backup purposes.



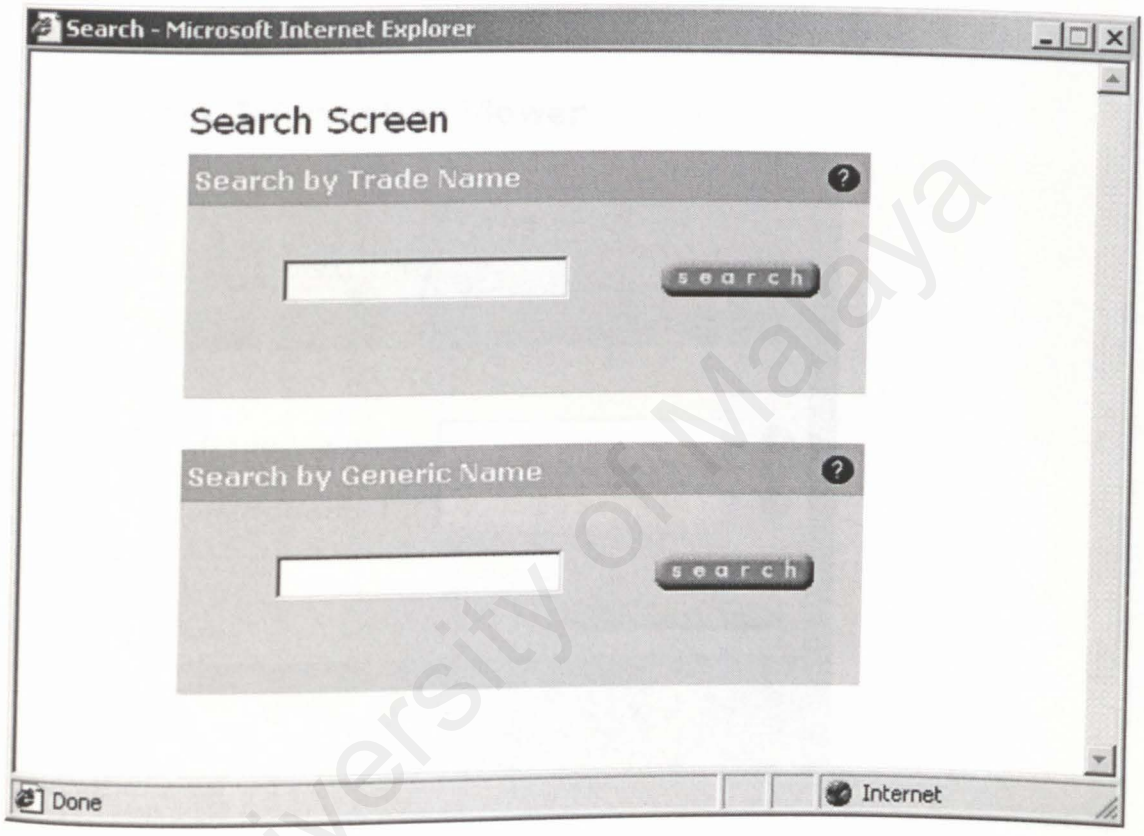
Screenshot 10: Archive Viewer

The Prescription Archive Viewer allows the user to search for past prescriptions that were made for a particular patient based on the patient's ID and the date.



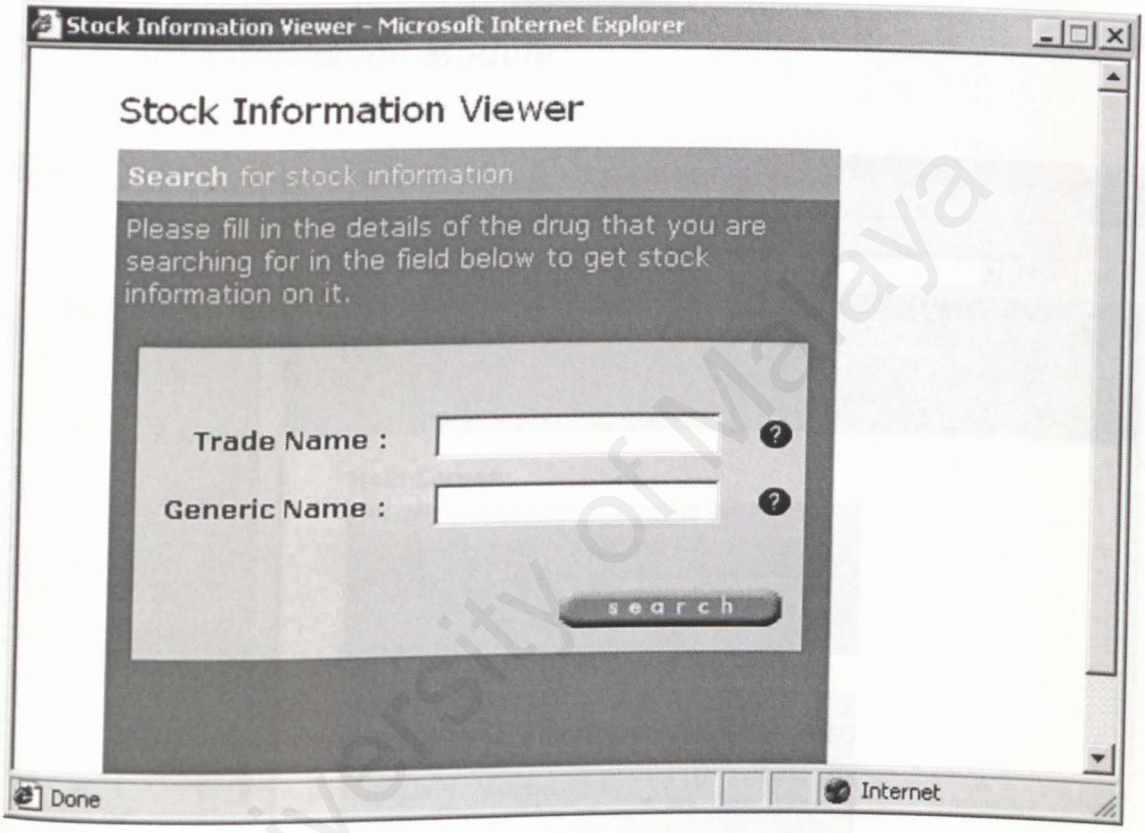
Screenshot 11: Archive Viewer Search Results

If a search proves successful then the details will be displayed to the user on screen for examination or printing. Should the user want to print out the information; a printer friendly version can be accessed on the same screen.



Screenshot 12: Drug Knowledge Base Search

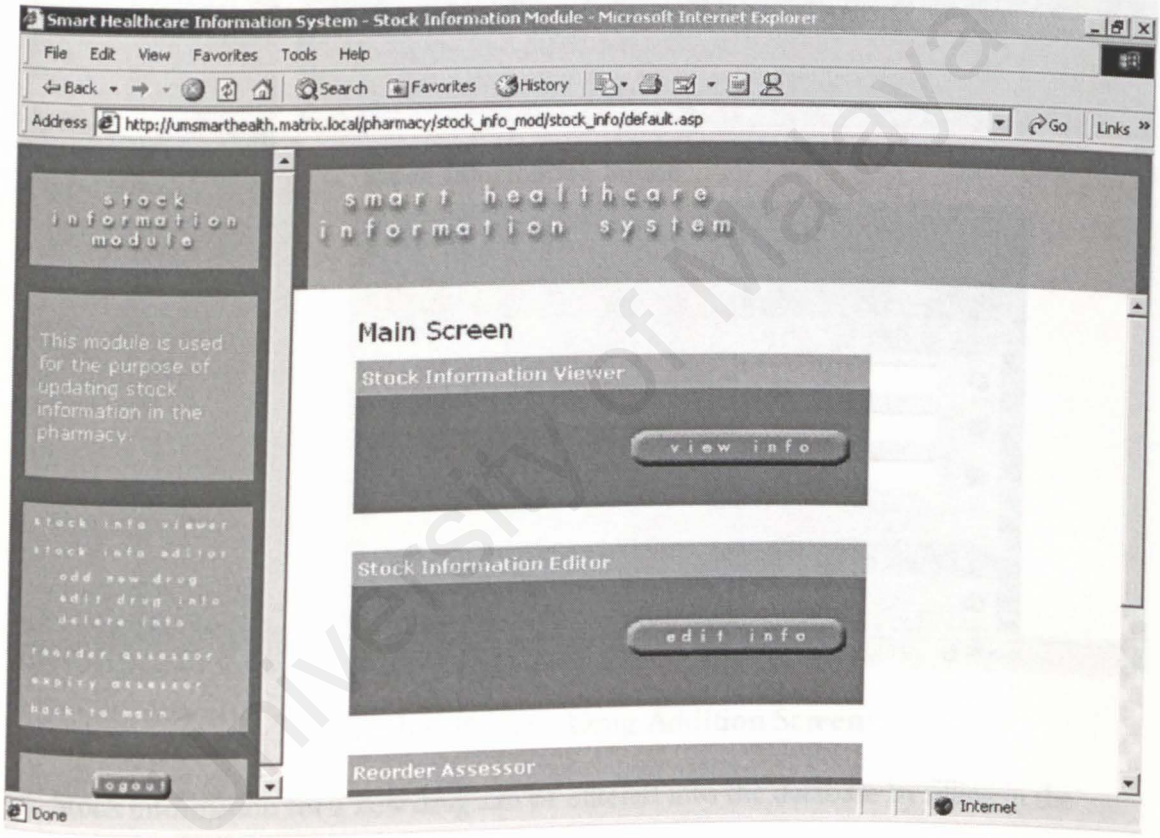
The pharmacist can also access information from the Drug Knowledge Base as well as check on stock information using the Stock Information Viewer.



Screenshot 13: Stock Information Viewer

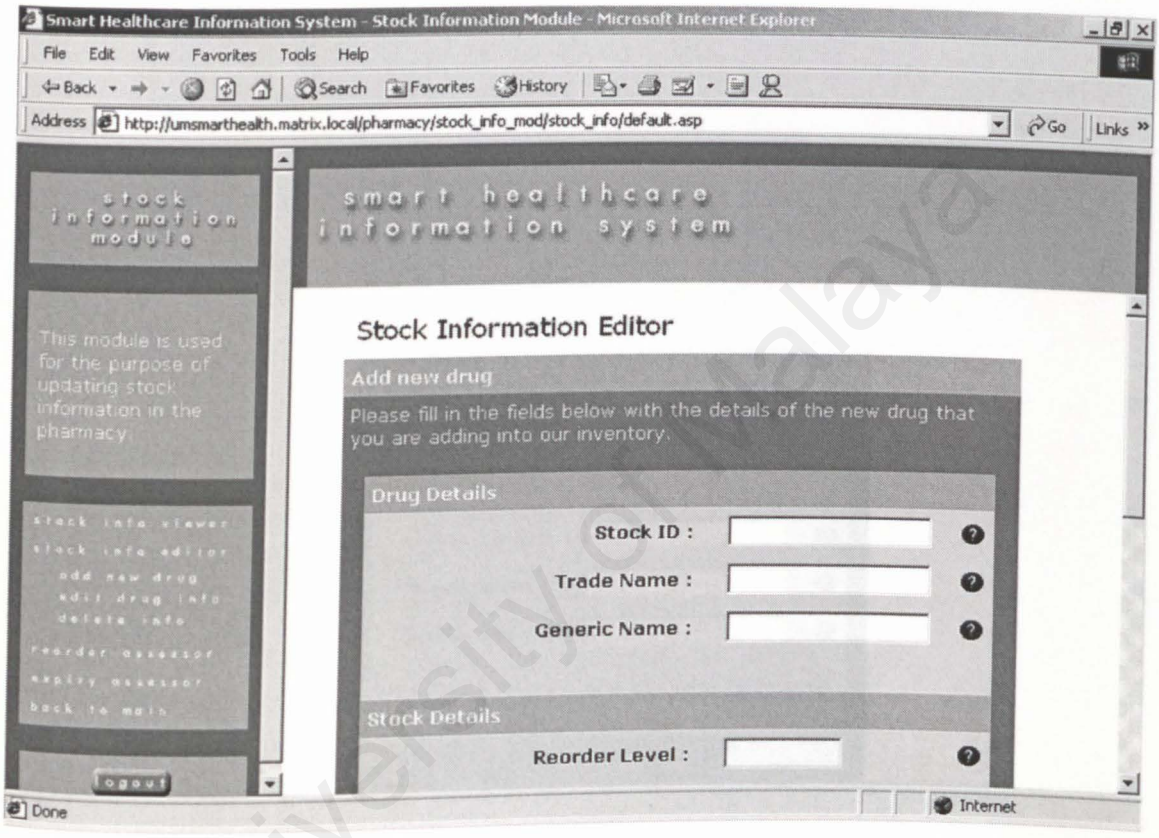
Both these applications are accessible by clicking the hyperlinks on the left navigation bar. Each application launches in its own window so that it doesn't interrupt the flow of the system when it is being used.

The Stock Information Module



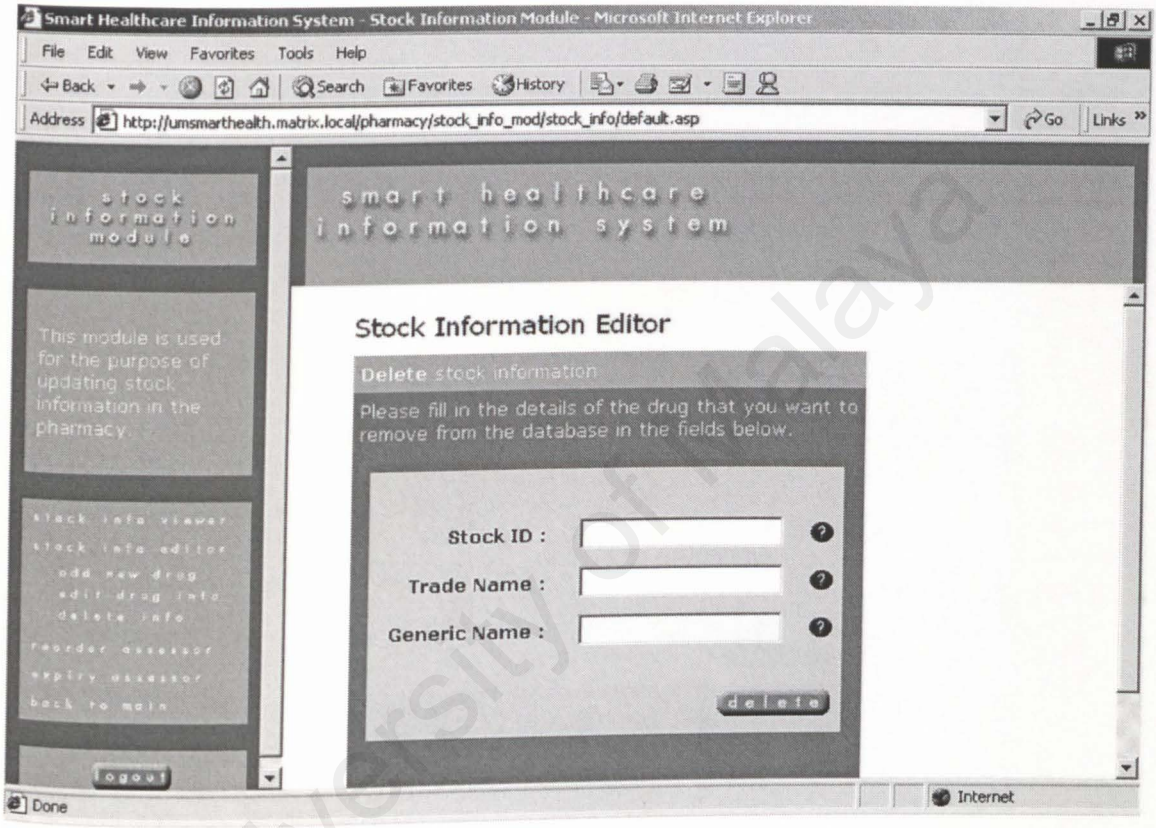
Screenshot 14: Stock Information Module Main Screen

The Stock Information Module is used to maintain stock information in the pharmacy by allowing the user to insert, edit and delete entries that relate to stock information in the database. The user will see the Main Screen upon entry into the system that shows him/her the main modules available in this part of the system.



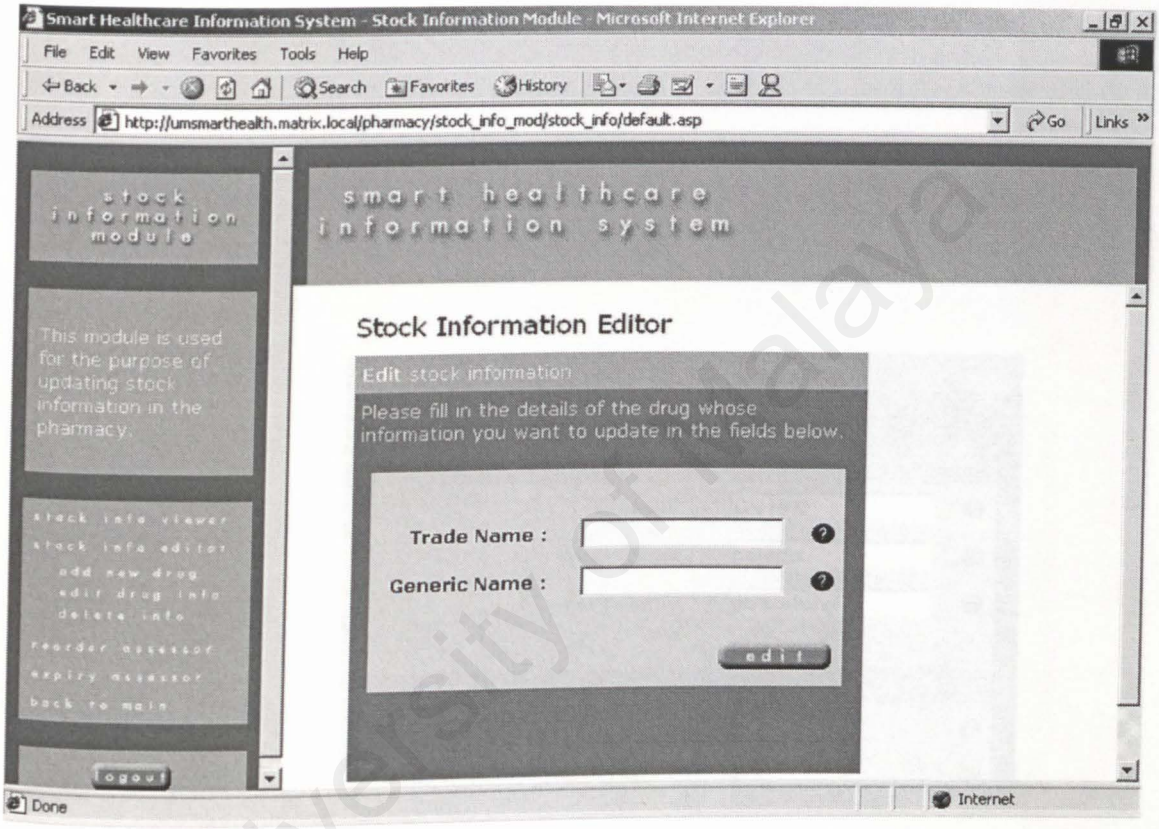
Screenshot 15: New Drug Addition Screen

Stock information for a new drug can be entered into the database by filling in the required details in the Add Drug section (screenshot 15). Once it is confirmed that no drug of that type exists in the database; the new entry will be inserted into the system.



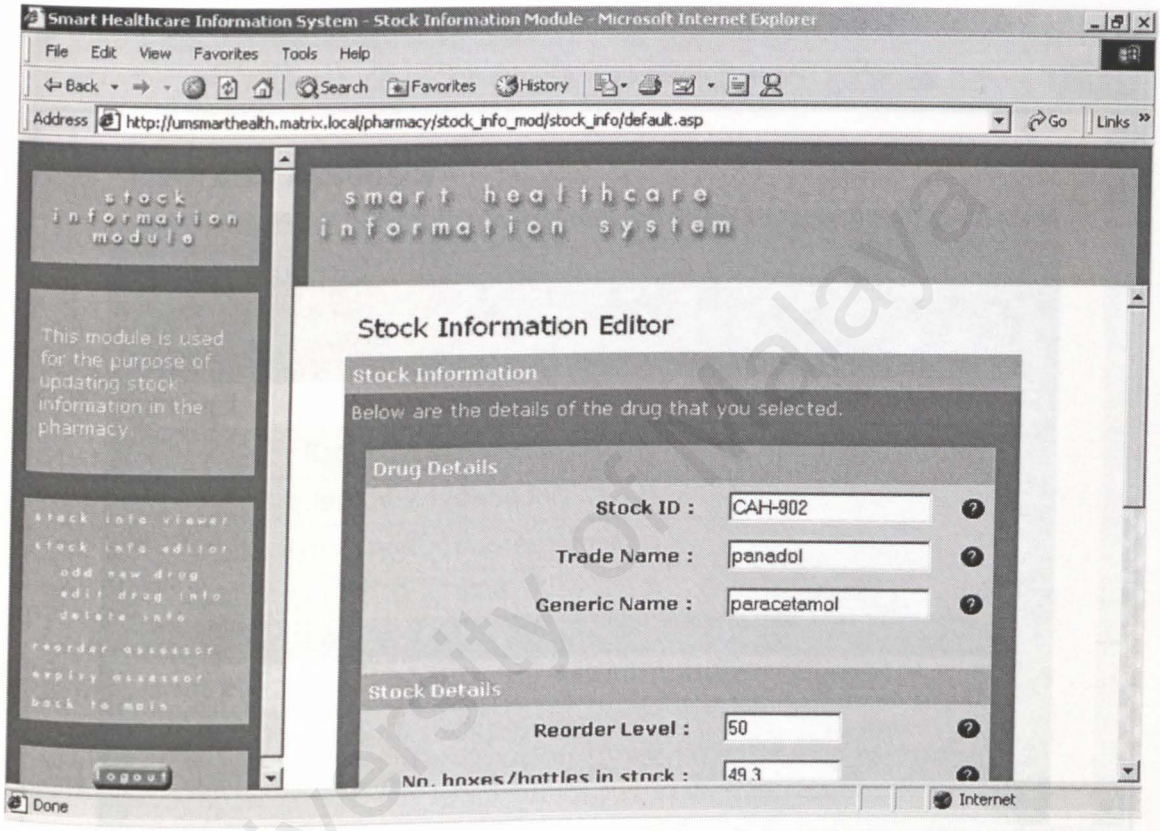
Screenshot 16: Stock Information Deletion Screen

Stock information can also be removed from the database easily using another section in the Stock Information Editor module that allows the user to search for the information that he/she wants to delete (screenshot 16).



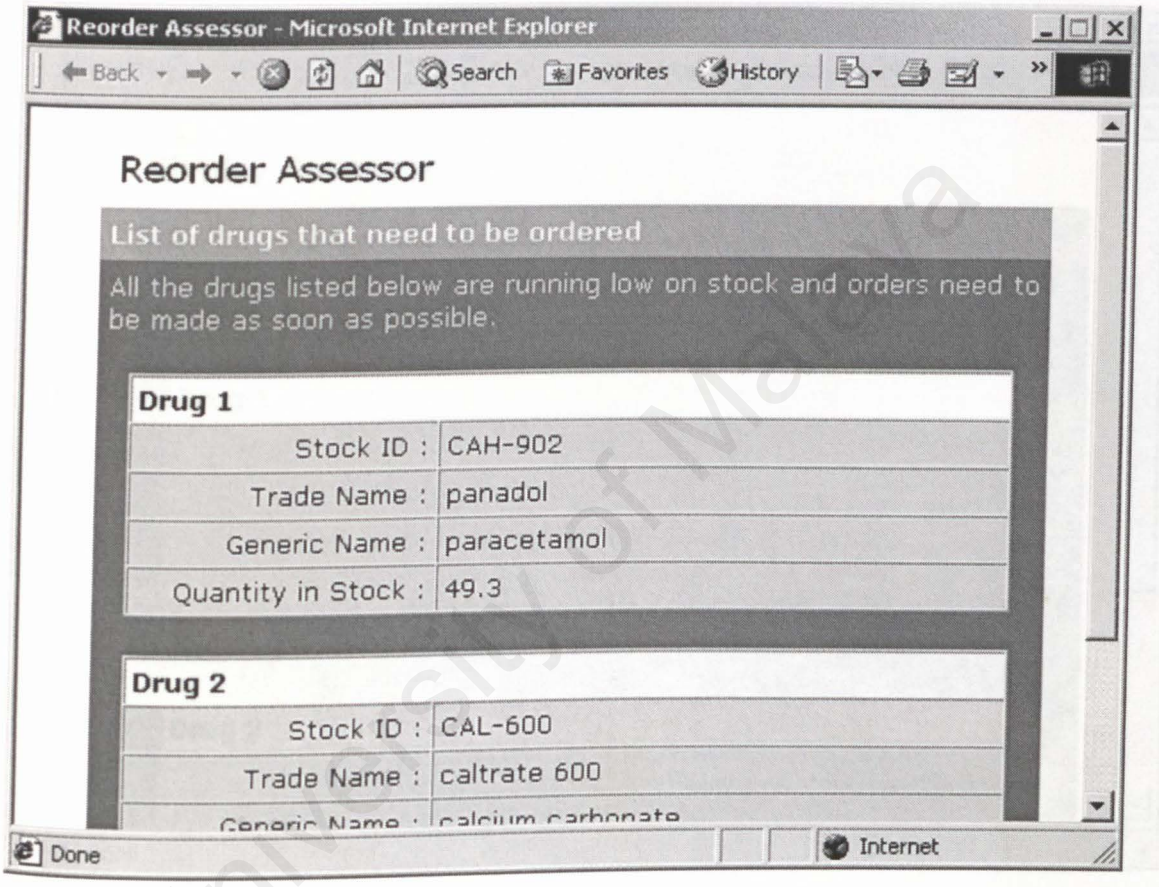
Screenshot 17: Stock Information Editor Search Screen

In addition to that, the Stock Editor module also allows the user to edit information that already exists in the database. He/she has to search for the information first using the search screen where the generic and trade name of the drug has to be filled into the form.



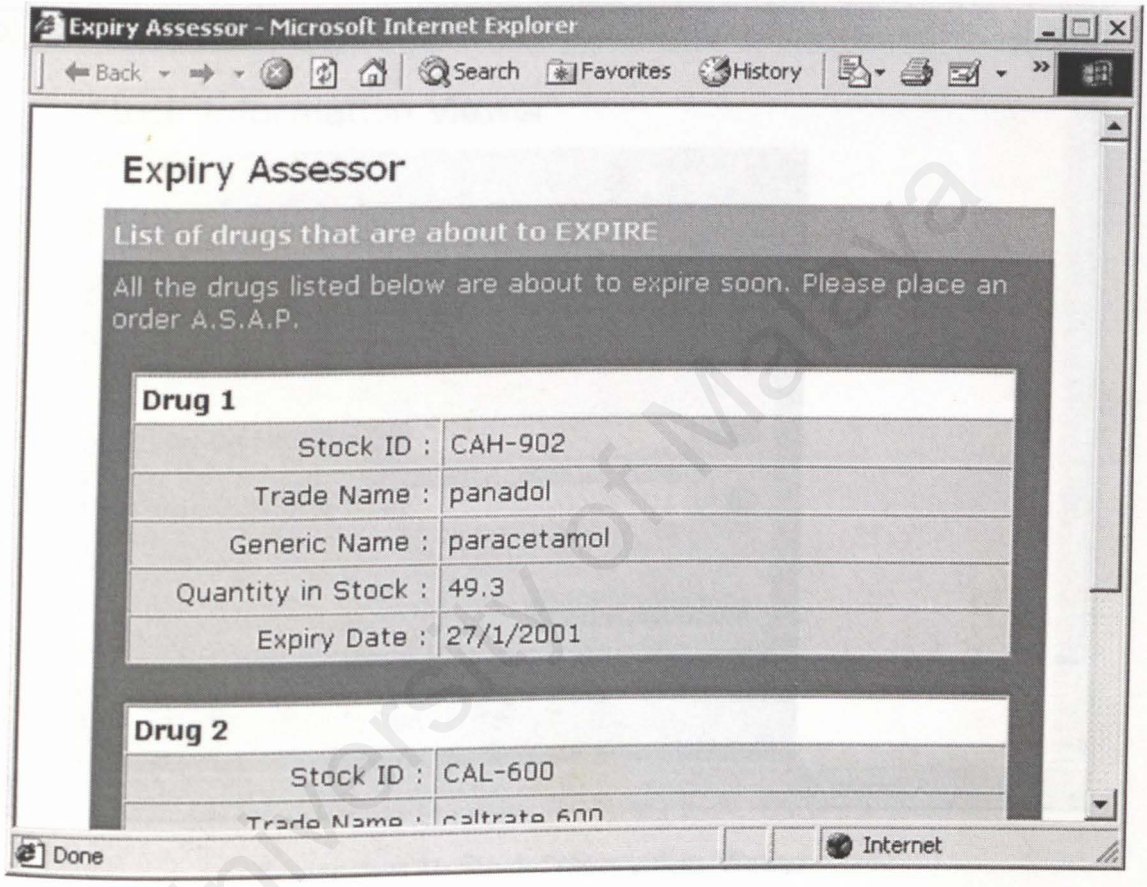
Screenshot 18: Stock Information Editing Screen

If the search is successful then the information for that drug will appear on screen so that the user can edit the information. Once he/she is satisfied with the changes then the database will be updated upon submission of the form that contains the data.



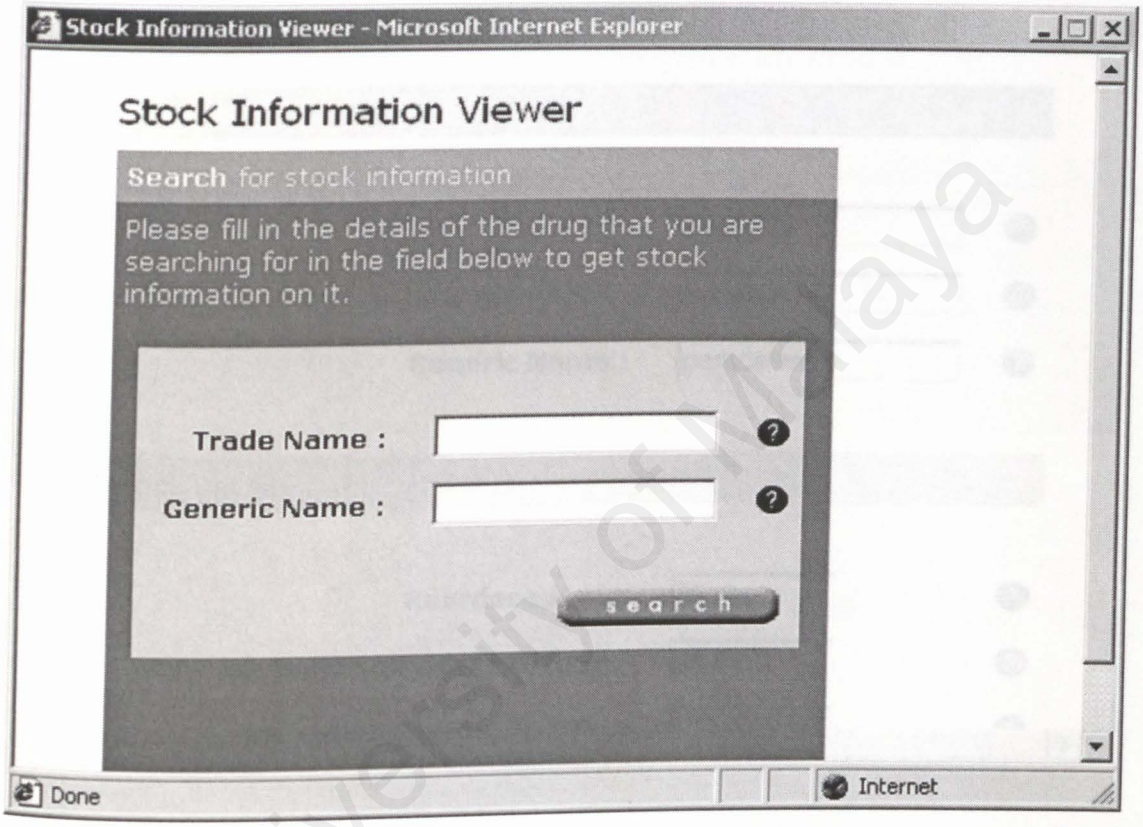
Screenshot 19: Reorder Assessor

Another useful facility that the Stock Information Module has is the Reorder Assessor that automatically checks the database and displays a list of all the drugs that are running low on stock. The list can then be printed out and an order placed with the supplier to procure new stock. This screen appears as a pop up window so that the user can check on orders not need to be made any time without having to disrupt the flow of using the system.



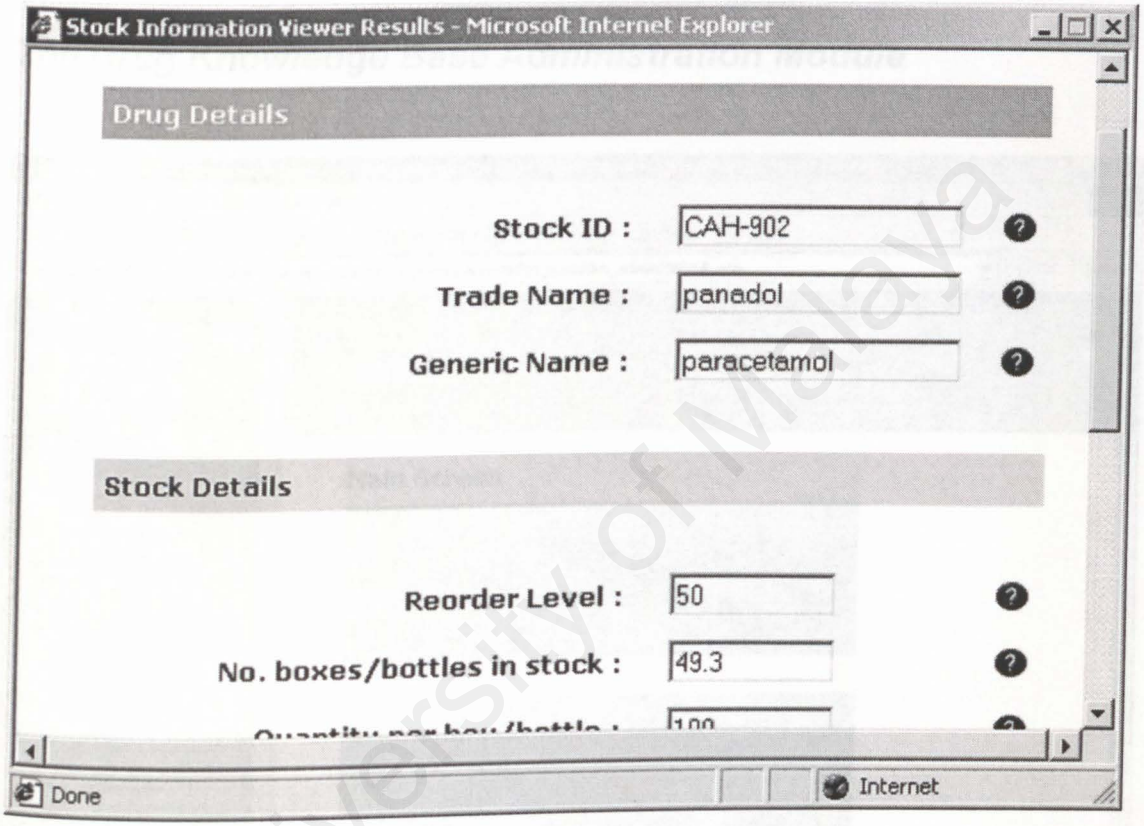
Screenshot 20: Stock Expiry Assessor

The Stock Expiry Assessor checks the database for drugs that are about to expire. Once the search is complete a list of drugs that are about to expire will be generated in HTML and displayed.



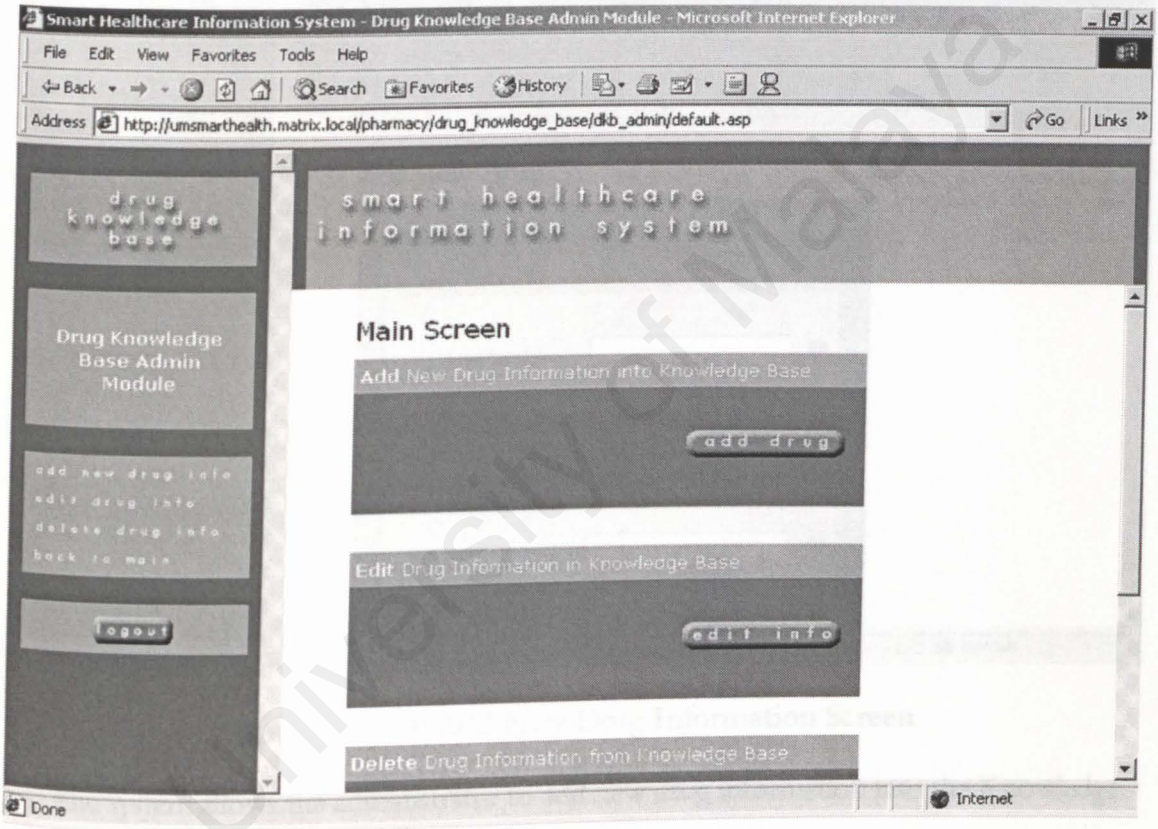
Screenshot 21: Stock Information Viewer

Stock information can just be viewed without editing facilities to avoid any accidental mistakes being made while viewing the information. First a search is made then the information required is displayed.



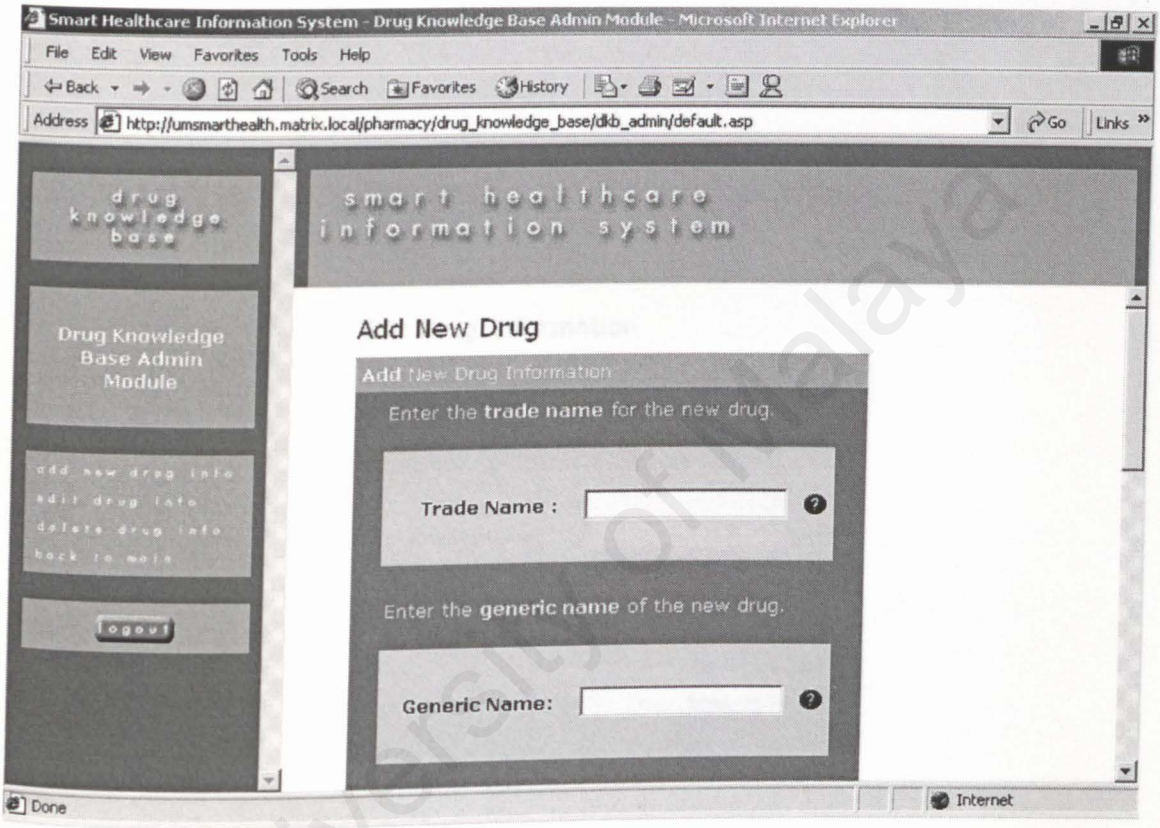
Screenshot 22: Stock Information Viewer Results

The Drug Knowledge Base Administration Module



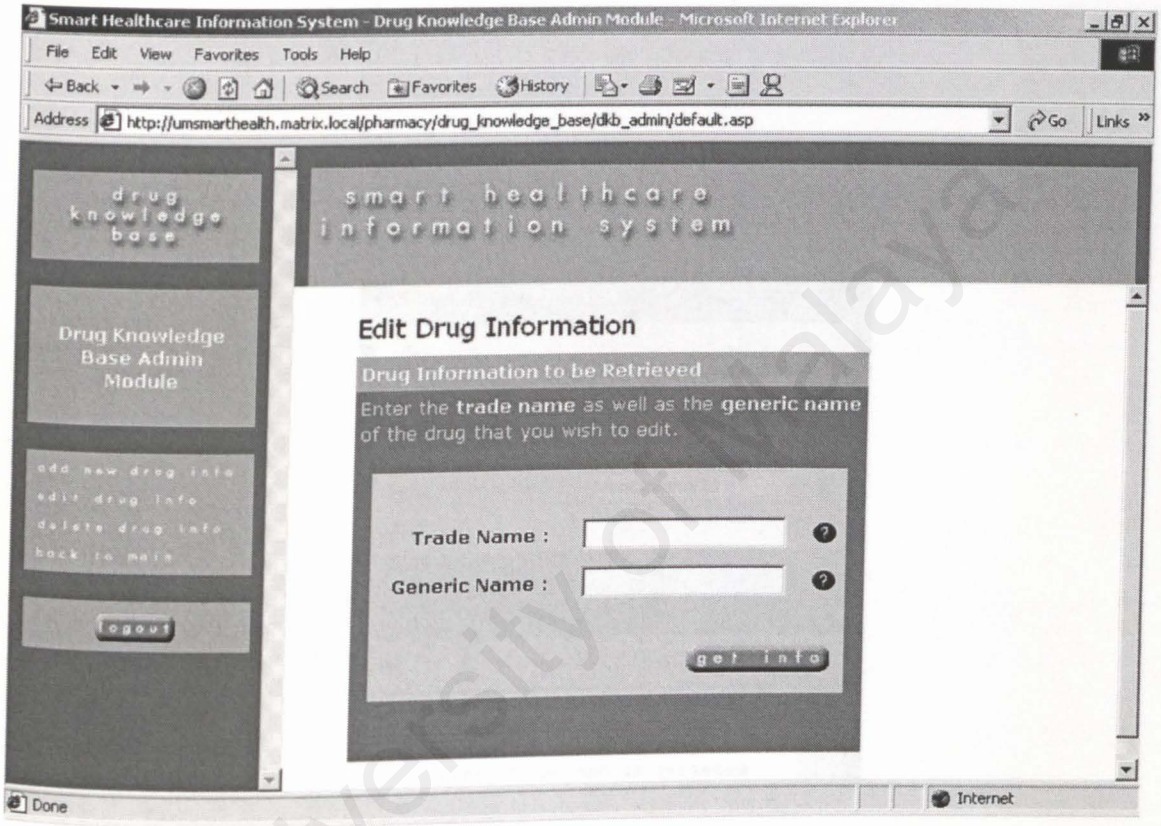
Screenshot 23: Drug Knowledge Base Admin Main Screen

The Drug Knowledge Base Administration module allows the Drug Knowledge Base administrator to manage information that will be displayed during Drug Knowledge Base searches done by users of the system.



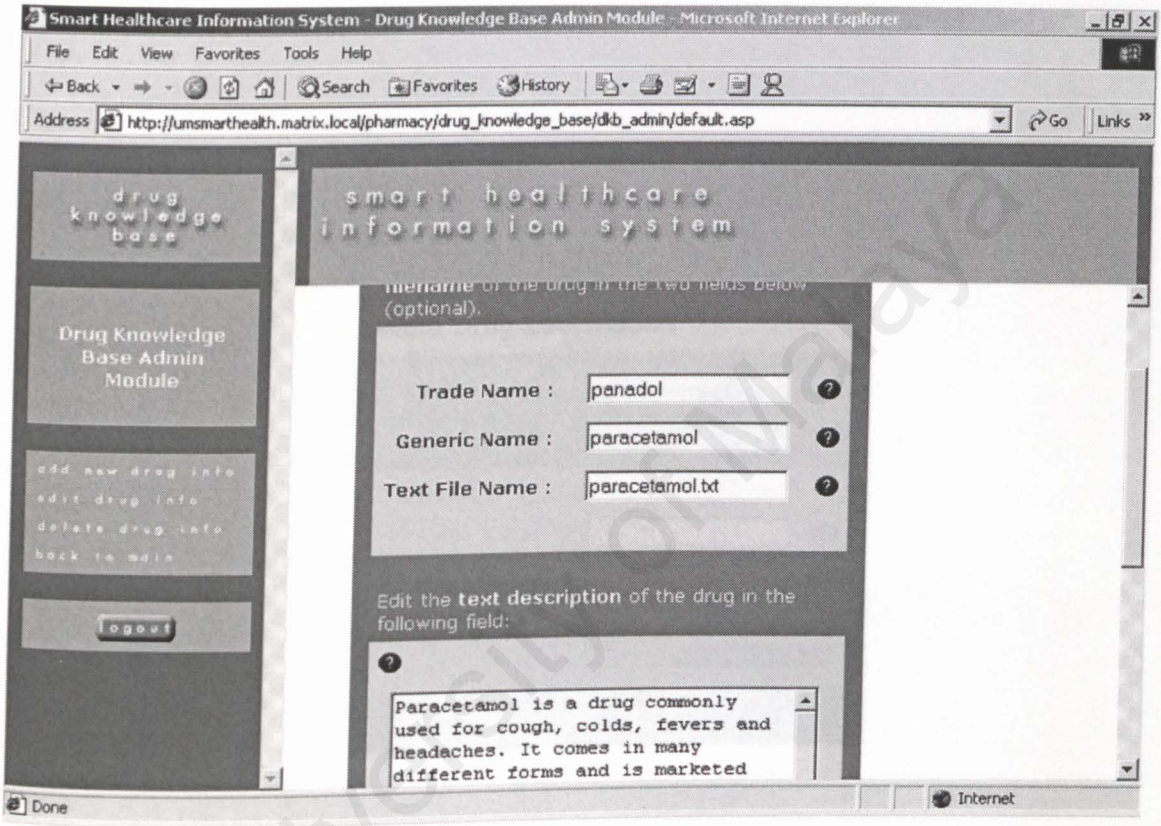
Screenshot 24: Add New Drug Information Screen

The system allows the administrator to add new drug information into the Knowledge Base using a standard form format that is used throughout the Pharmacy Management System.



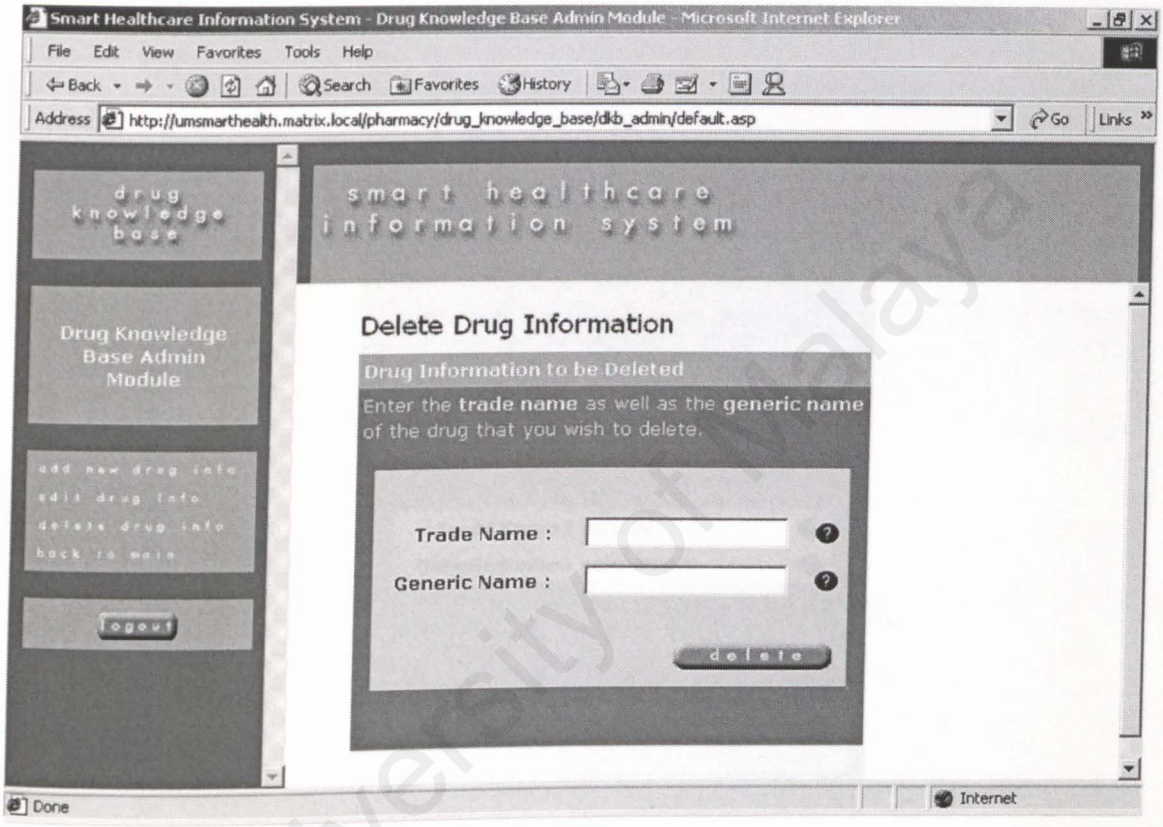
Screenshot 25: Edit Drug Information Screen

Drug information that currently exists in the system can also be edited using the same form format that is used throughout the entire Pharmacy Management System. First the user has to specify the generic and trade name of the drug that is to be edited in the search screen that appears.



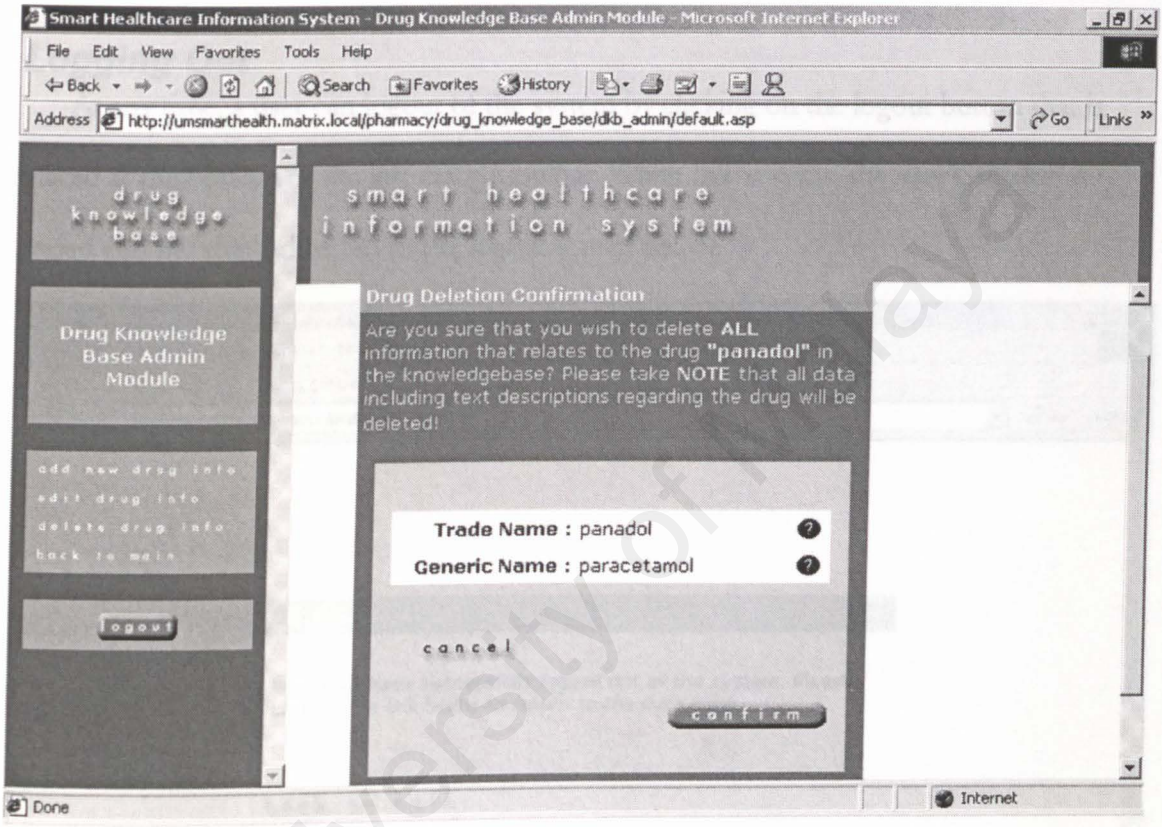
Screenshot 26: Edit Drug Information Screen

Once the information has been found, it will be displayed for editing in a form that will appear on screen. The information here can be edited and then updated. Changes will automatically be reflected as the change is made.



Screenshot 27: Drug Deletion Screen

The Drug Knowledge Base module also allows the user to remove drug information from the Knowledge Base as he/she sees fit. The first screen that the user has to go through to delete a drug can be seen in the screenshot above.

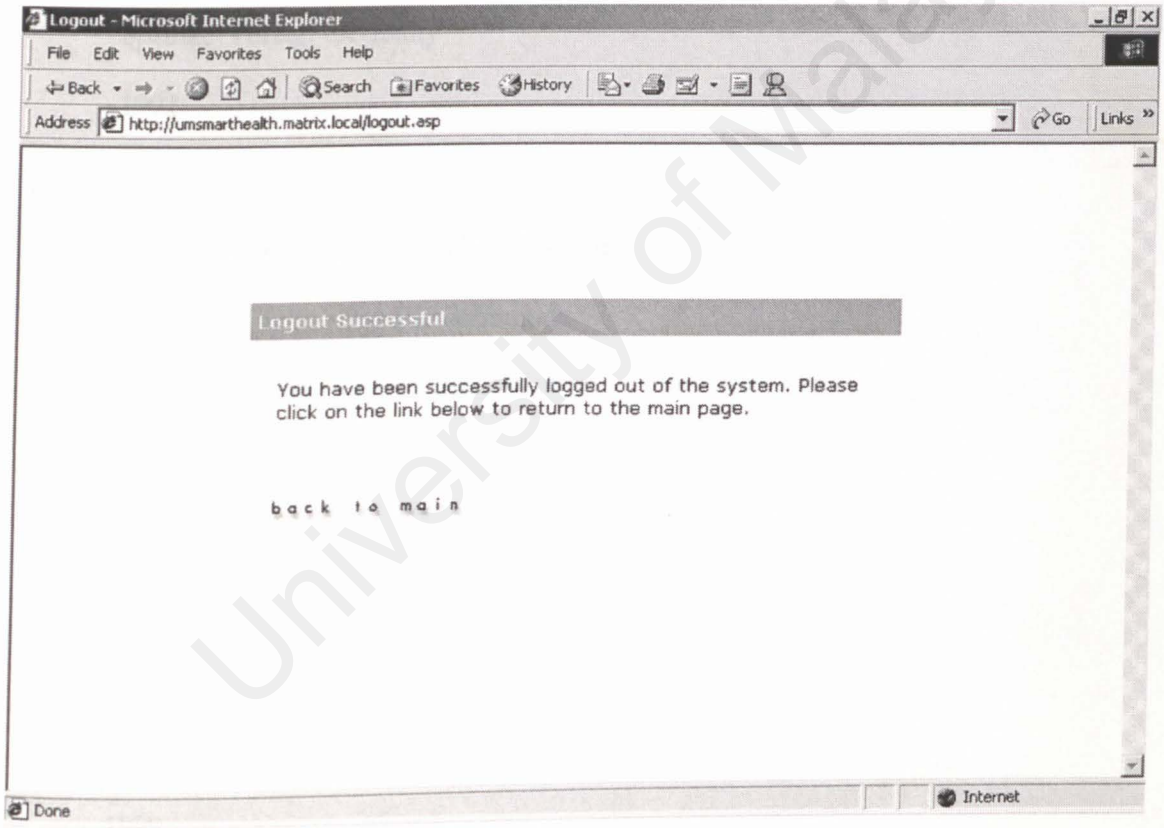


Screenshot 28: Drug Deletion Screen

A confirmation screen comes up to ensure that the user truly wants to delete the drug information that was requested. The user has the option of canceling the deletion or going through with it.

Logging Out

At any time, a user can logout of the system by clicking on the logout button that is placed at the bottom of the left navigation bar. When this is done, the user's session is ended and he/she is directed to the logout screen below.



Screenshot 29: Logout Screen

If the system is inactive for a period of 20 minutes then the user is logged out automatically and if he/she wants to use the system again, he/she has to log in again.

References

Books:

1. Rudi Van de Velde

   Health Information System, First Edition

   Springe-Verlag, Germany

   1992.

2. Ali Behforooz and Frederick J. Hudson,

   Software Engineering Fundamentals,

   Oxford University Press, New York

   1996.

3. A.Keyton Weissinger,

   ASP in a Nutshell, First Edition

   O' Reilly, United States of America

   1999.

4. Eric A.Smith,
Active Server Pages Bible
IDG Books, United States of America
1999.

5. Kenneth E. Kendall and Julie E. Kendall,
Systems Analysis and Design, Fourth Edition
Prentice Hall, United States of America
1999.

6. Raymond McLeod Jr.,
Management Information Systems
Prentice Hall, United States of America
1998.

7. Matt Childs, Paul Lomax & Ron Petrusha
VBScript in a Nutshell, First Edition
O'Reilly & Associates, United States of America
May 2000

8. M. Morris Mano

Digital Design

Prentice Hall, United States of America

1991.

9. Tim Evans

10 Minute Guide to HTML 4

QUE, United States of America

1998.

10. David B Bromilow

MIMS Annual Malaysia DIMS 1997/1998, 8th Edition

MediMedia International Group, Phillipines

1997.

Websites:

1. The Microsoft Website
   www.microsoft.com
2. The Linux Documentation Project
   www.aklug.org/LDP/
3. The Sun Java Website
   java.sun.com
4. The Netscape Website
   home.netscape.com/servers/index.html
5. Intranet Design Magazine
   idm.internet.com
6. IRT.org
   www.irt.org
7. internet.com
   www.internet.com
8. ASPAlliance.com
   www.aspalliance.com
9. The National Pharmaceutical Control Bureau
   www.bpfk.gov.my
10. Programmer's Resource.com
    www.programmersresource.com
11. The Apache Software Foundation
    www.apache.org
12. The Malaysian Pharmaceutical Society
    www.mps.org.my