



Faculty of Computer Science and
Information Technology
University Malaya
2001/2002

**Online Room Reservation And Scheduling System
For A Hospital
(ORRASH)**

Submitted by
Lee Heng Siong
WEK 990080

Under The Supervision Of
Assoc. Prof. Raja Noor Ainon

Moderator
Mr. Chiew Thiam Kian

A project proposal and report in partial fulfillment of the requirements of the degree
program of Bachelor of Computer Science University of Malaya

Contents

List of Figure	IV
List of Table	VI
ACKNOWLEDGEMENT	VII
ABSTRACT	VIII
Chapter 1: Introduction	1
1.1 Project Definition	1
1.2 Project Motivation	2
1.3 Project Objective	2
1.4 Project Scope	4
1.5 Project Important	4
1.6 Expected Outcome	5
1.7 Project Schedule	6
1.8 A Tour of The Report	8
Chapter 2: Literature Review	9
2.1 Introduction	9
2.2 Client/Server Architecture	10
2.2.1 What Is Client/Server Architecture?	10
2.2.2 Benefit of Client/Server Architecture	11
2.3 Development Tool	12
2.3.1 Microsoft Visual InterDev	12
2.3.2 Microsoft Visual Basic 6.0	14
2.4 Web Technology	16
2.4.1 CGI (Common Gateway Interface)	16
2.4.2 Cold Fusion	17
2.4.3 Active Server Pages (ASP)	19
2.4.4 JavaServer Pages (JSP)	22
2.5 Programming Language	23
2.5.1 Hypertext Markup Language (HTML)	23
2.5.2 VBScript	24
2.5.3 JavaScript	25
2.6 Database	27
2.6.1 Microsoft Access 2000	27
2.6.2 Microsoft SQL Server 7.0	28
2.6.3 Oracle8i	29
2.7 Web Server	31
2.7.1 Personal Web Server (PWS) 4.0	31

2.7.2 Microsoft Internet Information Server 4.0	31
2.8 Platforms and Operating System	32
2.8.1 Microsoft Windows NT Server 4.0	32
2.8.2 Microsoft Windows 2000 Server	33
2.9 Overview of The Existing System	34
2.9.1 Advantages of The Existing System	34
2.9.2 Weaknesses of The Existing System	35
Chapter 3: Methodology & System Analysis	36
3.1 Project Prototyping	36
3.1.1 Uses of Prototyping	39
3.1.2 Advantages of Prototyping	39
3.2 Project Life Cycle	41
3.2.1 Waterfall Model	41
3.2.2 Advantages of Waterfall Model	43
3.2.3 Disadvantages of Waterfall Model	43
3.3 System Requirement	44
3.3.1 Information Gathering	44
3.3.2 Functional Requirement	46
3.3.3 Non-Functional Requirement	53
3.3.4 Hardware Requirement	55
3.3.5 Software Requirement	56
Chapter 4: System Design	58
4.1 Data Flow Diagram	58
4.2 Entity-Relationship Diagram	66
4.3 Data Dictionary	68
4.4 User Interface Design	78
Chapter 5: System Implementation	
5.1 Introduction	83
5.2 Coding Styles	83
5.3 Coding Approach	85
5.4 Database Development	86
5.5 Software Tools for Design and Documentation	87
5.6 Software Tools for Testing	88
Chapter 6: System Testing	89
6.1 Introduction	89
6.2 Testing Techniques	89
6.3 Unit Testing	91
6.4 Integration Testing	91
6.5 System Testing	92
6.6 System Debugging	94

Chapter 7: System Evaluation	95
7.1 Problem Encountered and Solutions	95
7.2 System Strengths	96
7.3 System Limitations	99
7.4 Future Enhancements	100
7.5 Conclusion	101
REFERENCES	102
APPENDIX	103

List of Figure

Figure 1.1: Project Schedule	7
Figure 2.1: Client/Server Architecture	11
Figure 2.2: Sample diagram of CGI	17
Figure 2.3: Component of JavaScript	26
Figure 3.1: Prototyping Model	37
Figure 3.2: Waterfall Model With Prototyping	41
Figure 3.3: Structure of Online Room Reservation And Scheduling System For A Hospital (ORRASH)	46
Figure 3.4: Patient Module	47
Figure 3.5: Physician Module	47
Figure 3.6: Nurse Module	48
Figure 4.1: Context Level Diagram for ORRASH	60
Figure 4.2: DFD for Patient Module	61
Figure 4.3: DFD for Physician Module	62
Figure 4.4: DFD for Nurse Module	63
Figure 4.5: DFD for Decide Ward Type Sub Module	64
Figure 4.6: DFD for Security Sub Module	64
Figure 4.7: DFD for Patient Information Sub Module	65
Figure 4.8: DFD for Ward Booking Sub Module	65
Figure 4.9: Entity-Relationship Diagram for ORRASH	67
Figure 4.10: Main Page for ORRASH	78
Figure 4.11: Interface for Nurse Login	79

Figure 4.12: Interface for Register Form – Part 1	80
Figure 4.13: Interface for Register Form – Part 2	80
Figure 4.14: Interface for Decide Ward Type Form – Part 1	81
Figure 4.15: Interface for Decide Ward Type Form – Part 2	81
Figure 4.16: Interface for Ward Booking Form – Part 1	82
Figure 4.17: Interface for Ward Booking Form – Part 2	82
Figure 5.1: Part of Source Code	84
Figure 5.2: Part of Source Code	85

List of Table

Table 1.1: Project Divisions	3
Table 3.1: The Server Side Hardware Requirements for Developing ORRASH	55
Table 4.1: Components of DFD	59
Table 4.2: Physician_Personal_Info	68
Table 4.3: Physician_Professional	69
Table 4.4: Physician_Professional_Info	69
Table 4.5: Physician_Timetable	70
Table 4.6: Patient_Personal_Info	71
Table 4.7: Physician_Login	72
Table 4.8: Nurse_Login	72
Table 4.9: Patient_Login	72
Table 4.10: Patient_Treatment_Record	73
Table 4.11: Operation	73
Table 4.12: Operation_Room	74
Table 4.13: Operation_Room_Booking	74
Table 4.14: Operation_Room_Timetable	75
Table 4.15: Operation_Type	76
Table 4.16: Room	76
Table 4.17: Emergency_Type	76
Table 4.18: Ward_Type	76
Table 4.19: Ward	77
Table 5.1: Table of <i>Hospital</i> Database in Design View	87

ACKNOWLEDGEMENT

Many thanks to my project supervisor, Prof. Madya Raja Noor Ainon who is willing to allocate her precious time to had meeting together with me regularly to discuss the problem I met. Thanks for her precious advices and suggestions.

Besides that, I want to thank to my moderator, Mr. Chiew Thiam Kian. When in the viva, he had find out my system problems and give me some suggestions to improve my system.

I want to thank to the authority of University of Malaya Medical Center, Mr. Megat who had help me a lot in understanding the process of booking a room in a Hospital when interviewing him.

I also want to thank to my project partner, Lau Chong Ee who has the best cooperation with me to develop the system and willing to share his idea to improve the system.

Lastly, I want to thank to my parents and my family members because of their supports and encourages during the development of the system.

ABSTRACT

Online Room Reservation and Scheduling System For A Hospital (ORRASH) is a system that help the medical staff (nurse, physician) in a hospital to book a room (a bed of a ward, an operation room) and administrate room scheduling in an easy way by eliminate the conventional manual way. This system also provides the online patient information for the patient and medical staff to view the patient records using network.

Before develop this system, an interview was made with the authority of a hospital to know the process of booking a room in the hospital and a research was made on a similar system in the Internet.

The Waterfall model with prototyping was chooses as the project life cycle for this system. Microsoft Visual InterDev is selected as the development tool whereas the Internet Information Server 4.0 is the server and the back-end database is Microsoft SQL Server 7.0. Both servers are deployed on Windows 2000 Server platform.

This system is divided into three modules. There are patient module, nurse module, and physician module. Each module will describe the function that can be done under the module.

This system is important as it creates a new administration way to automate the process of booking room in a hospital. By using this system, the medical staff workload will be reduced and the mistake of booking a room, which is not available can be minimized.

Chapter 1: Introduction

1.1 Project Definition

The Digital Kuala Lumpur and Cyberjaya Smart City (DKLSC) is a smart city project in which a smart city is built for a smart and sustainable living environment. The system also provides the public information about the city and the smart city project. The system also provides the public information about the city and the smart city project.

Chapter 1:

Introduction

Chapter 1: Introduction

1.1 Project Definition

The Online Room Reservation And Scheduling System For a Hospital (**ORRASH**) is a system that allows medical staff in a hospital to book ward for a patient and operation room for a physician through network. The system also provide the patient information online to let the physician to check their patient information and let the patient themselves to check their medical information.

This system is divided into three modules; there are patient module, physician module and nurse module. Each module describe the function will be do and can be do in the system under that module. The description of the modules will be described in Chapter 3.

All the data and information is stored in the centralized database. The data and information store in the database is the patient information, room that has been booked and room which is still available. This will help the nurse in the hospital to get the patient information more easily and quickly and book an available room for a patient.

The maintainability of the data and information become easier as the changes to the certain data can be updated directly to the centralized database. This enabled the users from other corner of the system to get the most updated data and information. Furthermore this system is also realizing the concept of paperless office. The paperless office virtually eliminates spaces and folder for keeping the data with manual efforts.

1.2 Project Motivation

In the past, medical staff in a hospital has to manually manage the patient's information like storing, searching, and updating the patient's records. They also have to read some document before booking a room to make sure the room they want to book is available. This is wasting of time and often making mistake.

The problems mentioned above motivate this project to be carried out. Now, with the help of intelligent software, **ORRASH**, it will provide the best deal. In this system, the way to manage the patient's information and the process of booking a room become easy and saving a lot of time. The system of this project will eliminate the above problem.

1.3 Project Objective

The objective of this project is to provide sophisticated software in order to ease the process of booking room in a hospital. The hospital rooms including the patient wards and operation rooms. The ward is booked for patient by the nurse and the physician to do operation books the operation room.

The other objective of this project is to provide the quick searching of patient information by the medical staff of the hospital and let the patients to check their medical information online.

In short, the anticipated benefits to the hospital's medical staff and patient can be summarized as follows:

- Reduction of time required getting the patient information.

- Reduce cost, workload and staff level for the booking and scheduling process.
- Allows patient to make an appointment to see a physician through network.
- To provide a list out of ward available for a patient to choose.
- To provide the time an operation room is available for physician to book.
- Allow patients to change their information themselves, which is stored in the database (certain field only).
- To provide an easy to use and user friendly graphical user interface.
- To make a hospital system more computerize.

As the project involved two people in the process of development, I will cover some part of this system and my friend Lau Chong Ee will cover the other part, which is showed at **Table 1.1**.

Lee Heng Siong	Lau Chong Ee
1. Registration	1. Patient Module
2. Nurse module	2. Appointment
3. Booking ward	3. Operation room booking

Table 1.1 Project Divisions

1.4 Project Scope

The scope of this project is to:

- Develop an online registration system for a patient as an inpatient or outpatient under a certain medical center group.
- Develop a function that allows patients to make medical appointment.
- Develop an online reservation and scheduling for hospital's room.
- Develop a database system to house all data pertaining to the system.
- Develop a collection of interactive web pages as interface of this project.

1.5 Project Important

Using the conventional manual way to process or administrate the patient records, and book a room in a hospital is not cost and time-effective since they require spaces, energy and time to handle the data.

This project is important as it ensures the accurate, relevant, structured and timely information is provided to the appropriate personal at all level (patient, physician, nurse). It also provide the most up-to-date patients information in a second and eliminating the transfer of patient's records or patient's medical report from one location to another location manually.

The project also creates a new administration way to automate the process of booking room in a hospital. The system will do the room scheduling and provide accurate and the most up-to-date room records in time to the medical staff when needed. This will reduce the medical staff workload and minimize the mistake on booking a room that is not available.

This project will help to boost and maybe change the way Malaysian hospital strategy works into a better and competitive one.

1.6 Expected Outcome

The expected outcome of this project are listed as below:

- The new designing system is simple and user friendly, it is easy to operate, as it is menu driven. You just have to need some training and a little general knowledge. You can operate the system without any problem.
- Session object provided by IIS will keep track the user's state and navigation flow.
- The input data will be checked and if there are any errors, it will report to the user in a user-friendly manner.
- Provides the potential for vastly improving the quality, quantity, accuracy, and effectiveness of shared information.
- All web pages with standard graphic user interface and also same interface across multiple browser display.

- Acceptable response time when browsing the requested web page from the web server.
- The database can be easily expanded if the capabilities and functionalities of the system increase in the future.

1.7 Project Schedule

For a project developer, time is very important because it will determine the project can be finished in time. As a result, a project schedule is highly needed to ensure the effort is distributed within the prescribed time frame to make the best use of resources. **Figure 1.1** is my project schedule.

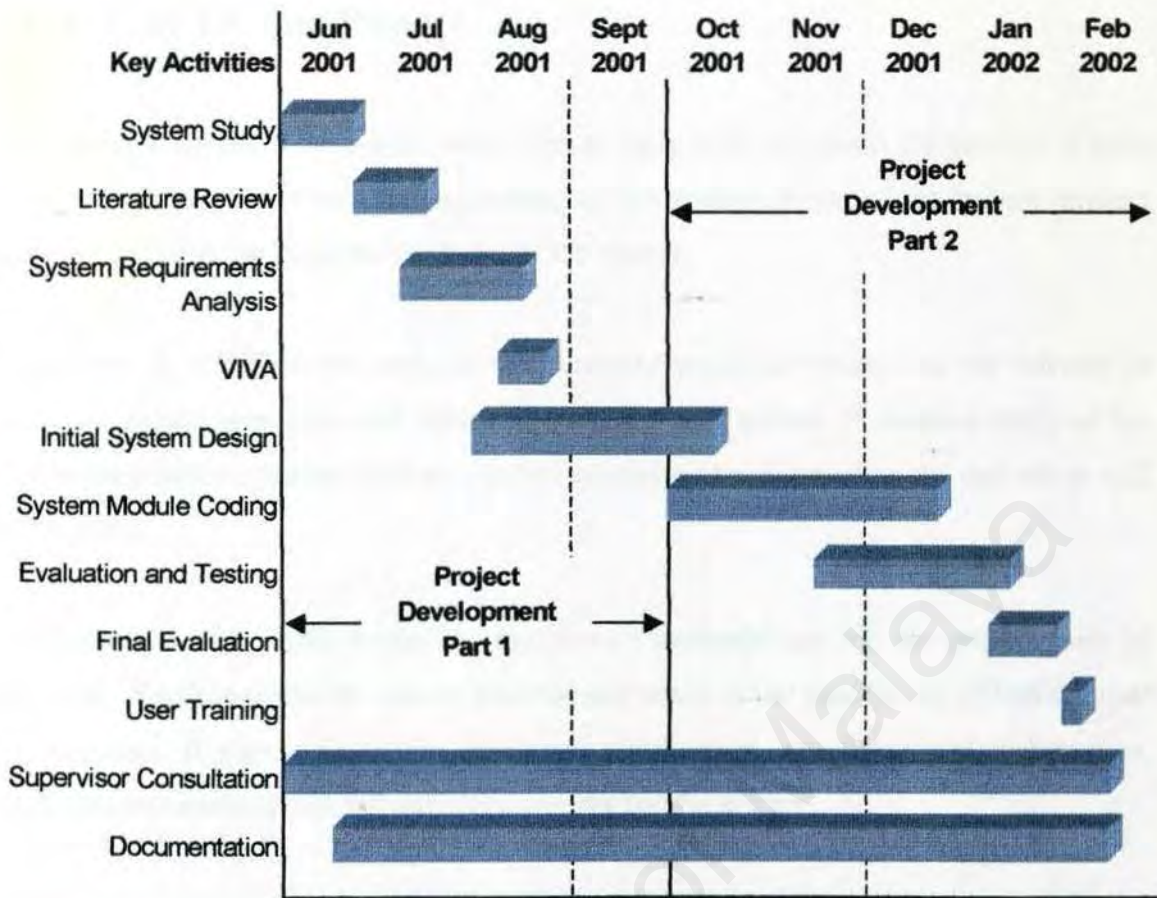


Figure 1.1: Project Schedule

1.8 A Tour Of The Report

In **Chapter 1**, it covers the project definition, as view from an overall perspective. It talks about the system objectives, the importance of the system, scope of the system, project schedule and also the expected outcome of the system.

In **Chapter 2**, it covers the study of the currently available systems on the Internet to know the system pros and cons before develop our new system. A detailed study on the system architecture, system platform, database system, development tools, and others will be included.

In **Chapter 3**, the justifications for the chosen methodology for the project will be specified. It will in-depth the system analysis and some of the method for collect the user requirements. It also includes the functional requirement, non-functional requirement, hardware requirement and software requirement for the system.

In **Chapter 4**, it describes the system design phase; it shows the data flow diagrams of the system. Database design and interface design of the system will be included. Data Dictionary, which comprises E-R diagram and table structure of the system, also will be explained here.

In **Chapter 5**, it describes the system implementation phase; it shows the coding styles and approaches. Besides that, it also shows the system database development, software tools for design and documentation, and software tools for testing.

In **Chapter 6**, it describes the system testing phase; it shows the testing techniques, unit testing, integration testing, system testing and system debugging.

In **Chapter 7**, it describes the system evaluation; it shows the encountered problems and it solutions, the system strengths, system limitations, future enhancements and the conclusion.

Chapter 2: Literature Review

Introduction

Chapter 2:

Literature Review

Chapter 2: Literature Review

2.1 Introduction

The world of computer technology is growing rapidly. In the beginning, computer technology is used in data processing. However, today computer technology is not limited applied in data processing, but also being used as a communication technology. Due to this existing technology, we make use of this technology to build a hospital system for reservation and scheduling.

In order to ensure the success in a software system development, a careful planning and research are essential. Therefore, in developing this project, research has been conducted in several related areas to gather the prerequisite information. We have make researches of some hospital system to know how a hospital operates. We also learn the client/server architecture that is using in our project. Information and data of web technology, programming language we consider to use is gathered through net surfing, reading on books, references, journals, newspapers and magazines. This is to compare which technology is most suitable for our system. Besides from supervisor, opinions from friends are also playing an important role to make this project successful.

2.2 Client/Server Architecture

Client/Server technology is the dominant computing architecture and evolved in response to critical issues organizations face in trying to use a variety of information technology effectively. [10]

2.2.1 What Is Client/Server Architecture?

The client/server model is an approach to software in which one application (the client) asks for and receives services from another application (the server). In a client/server environment, data are manipulated at the user level. Client/server computing can be considered totally user-driven, and the client/server environment allows for multivendor, multiproduct, multiapplication implementation. Essentially, client/server computing is a software-based architecture that enables distributed computing resources on a network to share common resources among groups of users at workstations.

Client/Server technology is a model, for the interaction between concurrently executing software processes. It is important to understand that the relationship between client and server is a command/control relationship. In any given exchange, the client initiates the request and the server responds accordingly. A server cannot initiate dialog with clients. The interaction between the client and server processes is a cooperative, transactional exchange in which the client is proactive and the server is reactive. This is the main differentiation between client/server and other, less constrained.

The simplicity of client/server model makes it very powerful. This definition highlights the four fundamental building blocks in client/server architecture as shown in **Figure 2.1**.

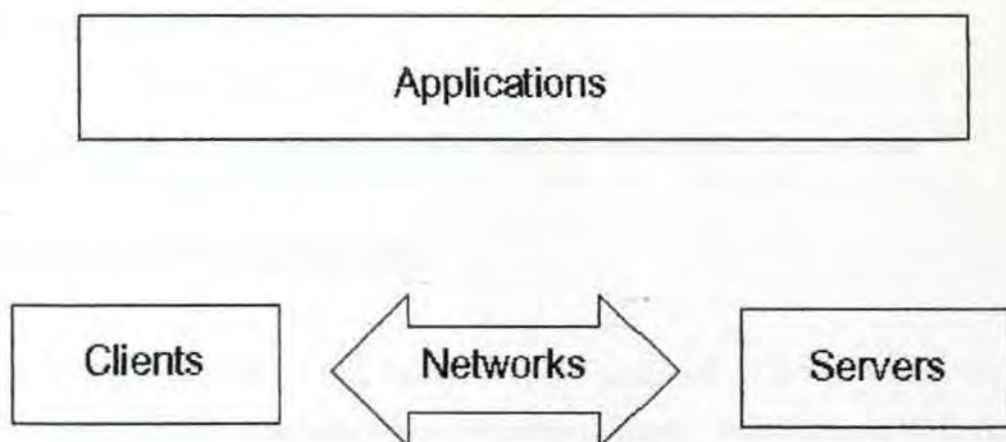


Figure 2.1: Client/Server Architecture

2.2.2 Benefit of Client/Server Architecture

Client/server is an open system. It offers organizations the ability to distribute processing and data across networks using powerful graphical workstations, servers, and mainframes. One of the prime benefits of a client/server system is lower costs. Another is increased productivity from the individual to the corporation that results from better access to information and the distribution of resources through the corporation.

Additional benefits of client/server include:

- *Interoperability - key components (client/server/network) work together.*
- *Scalability - any of the key elements may be replaced when the need to either grow or reduce processing for that element dictates, without major impact on the other elements.*
- *Adaptability - new technology may be incorporated into the system.*
- *Data Integrity - entity, domain and referential integrity are maintained on the database server.*
- *Accessibility - data may be accessed from WANs and multiple client applications.*
- *Performance - performance may be optimized by hardware and process.*
- *Security - data security is centralized on the server.*

2.3 Development Tool

In below are some of the development tools been considered for our system.

2.3.1 Microsoft Visual InterDev

Microsoft Visual InterDev 1.0 is built on top of Microsoft 's Visual Studio 97, which give it a very robust editing and online reference system. Visual Studio 97 is the same programming environment used in Microsoft Visual C++ and Microsoft Visual J++. It supports macros, VBA Automation, floating/dockable windows and tool bars along with many other features to make programming large applications fairly easy. Also integrated are a number of essential tools for developing dynamic Web sites referred to as content editors. Visual InterDev serves as the focal point for HTML, ActiveX components, graphics, video and audio. [13]

The Microsoft Visual InterDev development system provides the comprehensive resources necessary for successful Windows development. From building e-commerce Web solutions that take advantage of the new Windows 2000 clustering technologies, such as network and component load balancing, to scalable data-driven business applications.

For content editors, Visual InterDev ships with the FrontPage Editor for WYSIWYG HTML editing. A source editor enables you to edit HTML and ASP pages directly as well as SQL (Structured Query Language) stored procedures. The editor provides support for the full range of current browser technologies including generic HTML, Dynamic HTML, and additional add-ins such as Java applets, Microsoft ActiveX controls, and Netscape plug-ins. It provides maximum control of the appearance of the HTML while still allowing complete access to the source code, without sacrificing the formatting of the source code. This editor offers color-coding for quick visual feedback to syntax errors. In addition, editors for creating or editing images, animation, and even audio are included.

The Project Workspace gives you quick access to entire Web site in the FileView, while the DataView displays database connection, and the InfoView offers very thorough online help.

Integrated Visual Tool

The Project Workspace uses the standard hierarchical view of data and documents that Windows Explorer uses. When creating a new Web Project in the workspace, the Visual InterDev extensions on the Web server set up all the needed directories and permissions automatically.

By specifying a connection to an ODBC –compatible database, a database project can be added to the workspace. In the new DataView, you can view, edit, and create tables, data diagrams, stored procedures, and views.

Robust Development Support

For advanced Web site development, Visual InterDev supports both client- and server-side scripting. Currently, Visual InterDev support Visual Basic Scripting Edition (VBScript), Microsoft JavaScript (Jscript), and other plug-in scripting languages. A Script Wizard enables developer to quickly generate client- and server-side scripts with a point-and-click interface. ActiveX controls and Java Applets can be embedded into pages as well as server-side components for Active Server Pages.

Database Connectivity

Database connectivity is provided by means of the new Microsoft Visual Database Tools. These are a set of components that run inside of the Visual Studio 97 environment. They offer full support of ODBC, enabling Visual InterDev to connect to almost any database from high-end databases, such as Oracle, IBM DB/2, and SQL server, down to desktop

databases, such as Microsoft Visual FoxPro, Microsoft Access, and Borland dBASE. Visual InterDev ships with ODBC drivers for Microsoft SQL server, Oracle, and Microsoft Access (the Microsoft Jet database engine). Visual InterDev requires ODBC Version 3.0 or greater.

Web Site Management

It is very easy to keep the largest web site under control by organizing the web sites into workspaces and projects. A link view offers a visual representation of the web site's links from page to page. Creating folders and files in the FileView automatically creates directories and files on the web server.

Visual InterDev enables multiple webs authors to work on the same site at the same time by a number of version-control methods. You can simply "check out" files to work on them locally, and whenever you save the file in your editor, Visual InterDev detects the save and sends the changes back to the web server. This enables both designers and programmers to work on the same site at the same time without fear of overwriting each other's work.

Visual InterDev provides the ease of use and power of comparable tools available. Strong team development support, extensibility, and sever-side scalability make InterDev a great choice for many developers.

2.3.2 Microsoft Visual Basic 6.0

Microsoft Visual Basic 6.0 programming language is excellent to be implemented as Windows based application, where it provides an environment for fast and easy development of applications through the use of Graphical User Interface (GUI). It is an event driven language and supports some object-oriented programming. An application developer with an event driven model is more interactive compared to procedural

language and responds to events that happen in the computer environments, such as clicking a mouse button. With Microsoft Visual Basic, class modules (all object-oriented concepts) can be written and reused or distributed. [16]

Besides that, there are many other OCX and Active X controls that can be implemented according to the needs of the program being designed. A set of toolbar is prepared in Visual Basic to assist in the application development, where each component in it provides certain functionality. Microsoft Visual Basic also provides a huge library of references. The most often used reference is the "Microsoft DAO Library", This library provides methods to manipulate the database. With the DAO (Data Access Library), tables and databases could be created dynamically for any point of execution. The database will be in the Microsoft Access format. The available libraries and Active X controls in the Visual Basic enable systems, developers to manipulate the program written in minimum coding and frustrations.

Microsoft Visual Basic is one of the most popular programming tools in Windows environment, due to its RAD (Rapid Application Development) capability associated with it. It also includes a package and development wizard that will determine all the required dependency files that must be distributed with the project. This wizard will be used to create setup disk(s) for installing the application in any platform recommended thus making the installation process of the viewing created program easier.

2.4 Web Technology

In development of our project, some of the web technologies have been considered. In below are some of the technologies been considered for our system.

2.4.1 CGI (Common Gateway Interface)

As you traverse the vast frontier of the World Wide Web, you will come across documents that make you wonder, "How did they do this?" These documents could consist of, among other things, forms that ask for feedback or registration information, imagemaps that allow you to click on various parts of the image, counters that display the number of users that accessed the document, and utilities that allow you to search databases for particular information. In most cases, you'll find that these effects were achieved using the Common Gateway Interface, commonly known as CGI. [2]

One of the Internet's worst kept secrets is that CGI is astoundingly simple. That is, it's trivial in design, and anyone with an iota of programming experience can write rudimentary scripts that work. It's only when your needs are more demanding that you have to master the more complex workings of the Web. In a way, CGI is easy the same way cooking is easy: anyone can toast a muffin or poach an egg. It's only when you want a Hollandaise sauce that things start to get complicated.

CGI is the part of the Web server that can communicate with other programs running on the server. With CGI, the Web server can call up a program, while passing user-specific data to the program (such as what host the user is connecting from, or input the user has supplied using HTML form syntax). The program then processes that data and the server passes the program's response back to the Web browser.

CGI isn't magic; it's just programming with some special types of input and a few strict rules on program output. Everything in between is just programming. Of course, there are

special techniques that are particular to CGI, and that's what this book is mostly about. But underlying it all is the simple model shown in **Figure 2.2**.

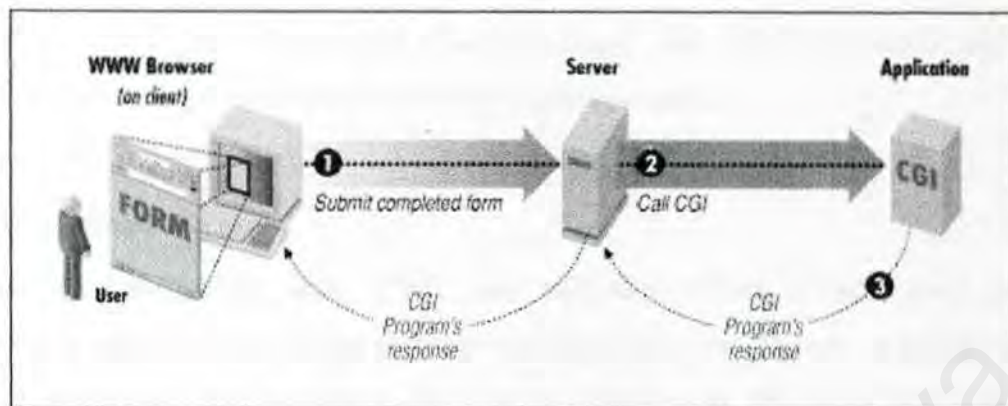


Figure 2.2: Simple diagram of CGI

2.4.2 Cold Fusion

Cold Fusion was developed by the Allaire Corporation (Nasdaq: ALLR) to be a simple to use yet powerful alternative to Perl and other CGI technologies. [1]

It is a Web Application Server:

Cold Fusion is an application that runs on a web server. Currently, it runs on Linux, Solaris, and Windows Servers. (yes, even personal web server on win98) The Cold Fusion Web Application Server works with the HTTP server to process requests for web pages. Whenever a Cold Fusion page is requested, the Cold Fusion Application Server executes the script or program the page contains.

It is a programming Language:

Cold Fusion is a language. It can create and modify variables just like other languages you may be familiar with. It has program flow controls like "IF, Switch Case, Loop, etc". It has many built in functions for performing more complicated tasks (like finding what day the 3rd of august will be in 2007 "`DayOfWeekAsString(DayOfWeek('2007/08/03'))`")

It is a Database Language

Actually, Cold Fusion is not a Database Language, but Cold Fusion makes interacting with your database (Sybase, Oracle, MySQL, SQL, or Access) simple. Using standard SQL (Structured Query Language) your web pages and web applications can easily retrieve, store, format, and present information dynamically.

It is Tag Based:

If you are comfortable with HTML you will love CFML (Cold Fusion Markup Language). Many of Cold Fusion powerful features (Like reading from and writing to the servers hard drive) are tag based. Just like a `<Table>` tag can take "arguments" like 'width' or 'align', the `<CFFILE>` tag takes "arguments" specifying 'action=read/write/copy/delete', 'path=' etc.

It Integrates Technologies:

Wouldn't it be nice if you didn't have to write all the JavaScript for your pages? The CF tag `<CFFORM>` will automatically build all the JavaScript code to verify required fields before the form submits. Cold Fusion also has tags to embed COM, Corba, and Java Applets/Servlets.

It is Scalable:

Cold Fusion was designed to build complex, high traffic web sites. Sometimes a web designers biggest problem is that a site becomes popular. Cold Fusion is designed to run on multi-processor machines, and allows you to build a site that can be run on a "cluster" of servers.

It is a "Server Side" Language

Unlike JavaScript, and Java Applets, which run on the "client", or "web browser", Cold Fusion runs on the Web Server. This means that scripts you write in Cold Fusion will run the same way on Every Browser.

Why would you use Cold Fusion?

Cold Fusions' "Cold Fusion Markup Language" (CFML) makes web programming easy for new developers. With 70+ CFML tags and over 200 custom functions, practically any web application can be built quickly. Tools like the server side includes "<CFINCLUDE>" help speed the development and improve the readability of any web site.

Cold Fusion can be used on your web site whenever you need customer interaction. Use it to process forms, make parts of your web site "secure", and gather or publish data. You can use it to build web applications like bulletin boards, pop mail clients, online-calendars, and chat rooms. Cold Fusion scripts can be written to track hits, clicks, return visits and other valuable traffic statistics.

2.4.3 Active Server Pages (ASP)

Active Server Pages is an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions. Active Server Pages enables server side scripting for IIS with native support for both VBScript and JScript. [3]

Files created with Active Server Pages have the extension .ASP. With ASP files, you can activate your Web site using any combination of HTML, scripting, such as JavaScript or VBScript and components written in any language. This means your ASP file is simply a file that can contain any combination of HTML, scripting, and calls to components.

When you make a change on the ASP file on the server, you need only save the changes to the file—the next time the Web page is loaded, the script will automatically be compiled. How does this happen? It works because ASP technology is built directly into Microsoft Web servers, and is thus supported on all Microsoft Web servers: Windows NT Internet Information Server (IIS) 3.0, Windows NT Workstation, and Windows 95 Personal Web Server.

When a browser requests an ASP file from your Web server, your Web server calls Active Server Pages to read through the ASP file, executing any of the commands contained within and sending the resulting HTML page to the browser. An ASP file can contain any combination of HTML, script, or commands. The script can assign values to variables, request information from the server, or combine any set of commands into procedures.

ASP uses the delimiters "<%" and "%>" to enclose script commands. For example, the code below sets the value of the variable "MyFavTVShow" in the user cookie to "I Dream of Jeannie."

```
<%Response.Cookies("MyFavTVShow")="I Dream of Jeannie"%>
```

The scripting languages supported by ASP in turn support use of the If-Then-Else construct. Finally, you can embed some real logic into your HTML. For example, the following code from the IIS documentation shows how you can set the greeting shown based upon the time of day.

```
<FONT COLOR="GREEN">
```

```
<%If Time >= #12:00:00 AM# And Time < #12:00:00 PM# Then%>
```

```
Good Morning!
```

```
<%Else%>
```


Hello!

<%End If%>

Built-in Objects

ASP includes five standard objects for global use:

Request—to get information from the user

Response—to send information to the user

Server—to control the Internet Information Server

Session—to store information about and change settings for the user's current Web-server session

Application—to share application-level information and control settings for the lifetime of the application

The Request and Response objects contain *collections* (bits of information that are accessed in the same way). Objects use *methods* to do some type of procedure and *properties* to store any of the object's attributes (such as color, font, or size).

2.4.4 JavaServer Pages (JSP)

JavaServer Pages (JSP) technology allows web developers and designers to rapidly develop an easily maintain, information-rich, dynamic web pages. As part of the Java family, JSP technology enables rapid development of web-based applications that are platform independent. JavaServer Pages technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content. [4]

JavaServer Pages technology uses XML-like tags and scriptlets written in the Java programming language to encapsulate the logic that generates the content for the page. Additionally, the application logic can reside in server-based resources (such as JavaBeans component architecture) that the page accesses with these tags and scriptlets. Any and all formatting (HTML or XML) tags are passed directly back to the response page. By separating the page logic from its design and display and supporting a reusable component-based design, JSP technology makes it faster and easier than ever to build web-based applications.

JavaServer Pages technology is an extension of the Java Servlet technology. Servlets are platform-independent, 100% pure Java server-side modules that fit seamlessly into a web server framework and can be used to extend the capabilities of a web server with minimal overhead, maintenance, and support. Unlike other scripting languages, servlets involve no platform-specific consideration or modifications; they are Java application components that are downloaded, on demand, to the part of the system that needs them. Together, JSP technology and servlets provide an attractive alternative to other types of dynamic web scripting/programming that offers platform independence, enhanced performance, separation of logic from display, ease of administration, extensibility into the enterprise and most importantly, ease of use.

2.5 Programming Language

In below are some of the programming languages been considered for our system.

2.5.1 Hypertext Markup Language (HTML)

Hypertext Markup Language is the "language" used to create nearly all WWW documents. HTML allows you to create formatted text, include images on your pages, and add links from one document to another. [9]

HTML is a simple subset of SGML, the Standard Generalized Markup Language. Specifically, HTML is what is called a DTD, a Document Type Definition which defines precisely those descriptive elements-the-syntax-needed for a specific type of document: in this case, a hypertext document.

As a subset of SGML, HTML is not concerned with the format (i.e. appearance) of a document; instead, it describes a document's logical structure leaving it up to the client side – the browser – to render the document as desired by the user.

The HTML is the primary language of documents served by Web servers. It provides a rich and growing set of tags that are embedded in documents to specify how the content should be formatted on a page. These tags also enable you to establish hypertext links from content in one document to content in other documents (which can be local or on a server anywhere in the world). HTML also provides mechanisms for invoking programs and services on Web servers.

An HTML document is an ASCII file (a pure file with no word processing coding). It contains text (what people see on the screen) and tags that tell browsers (like Netscape or Internet Explorer) how to format the text. This simple example shows both text and the tags (in angle brackets).


```
<html>  
<head>  
<title>Title of The Project</title>  
</head>  
<body>  
<p>This is the contents of the project </p>  
</body>  
</html>
```

2.5.2 VBScript

VBScript, the newest member of the Visual Basic family of programming languages, brings active scripting to a wide variety of environments, including Web client scripting in Microsoft Internet Explorer version 3.0 and Web server scripting in Microsoft Internet Information Server version 3.0. [6]

VBScript is case insensitive. For example, changing the variable I to a lowercase in does not change the behavior. VBScript syntax is based on that of Visual Basic. If you already know Visual Basic or Visual Basic for Applications, VBScript will be very familiar. Even if you don't know Visual Basic, once you learn VBScript, you're on your way to programming with the whole family of Visual Basic languages.

VBScript talks to host applications using ActiveX Scripting. With ActiveX Scripting, browsers and other host applications don't require special integration code for each scripting component. ActiveX Scripting enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer. With ActiveX Scripting, language vendors can create standard language run times for scripting. Microsoft will provide run-time support for VBScript.

2.5.3 JavaScript

JavaScript is a platform-independent, event-driven, interpreted programming language. JavaScript is useful for adding interactivity to the World Wide Web because scripts can be embedded in HTML files (i.e., web pages) simply by enclosing code in a `<SCRIPT>` `</SCRIPT>` tag pair. All modern browsers can interpret JavaScript but with some caveats. [8]

In practice, JavaScript is a fairly universal extension to HTML that can enhance the user experience through event handling and client-side execution, while extending a web developer's control over the client's browser.

JavaScript is object-based scripting language for client and server applications. JavaScript lets you create applications that run over the Internet. Client applications run in a browser, such as Netscape Navigator, and server applications run on a server, such as Netscape Enterprise Server. Using JavaScript, you can create dynamic HTML pages that process user input and maintain persistent data using special objects, files, and relational databases. Through JavaScript's LiveConnect functionality, your applications can access Java and CORBA distributed-object applications.

Server-side and client-side JavaScript share the same core language. This core language corresponds to ECMA-262, the scripting language standardized by the European standards body, with some additions. The core language contains a set of core objects, such as the Array and Date objects. It also defines other language features such as its expressions, statements, and operators. Although server-side and client-side JavaScript use the same core functionality, in some cases they use them differently.

The components of JavaScript are illustrated in **Figure 2.3**.

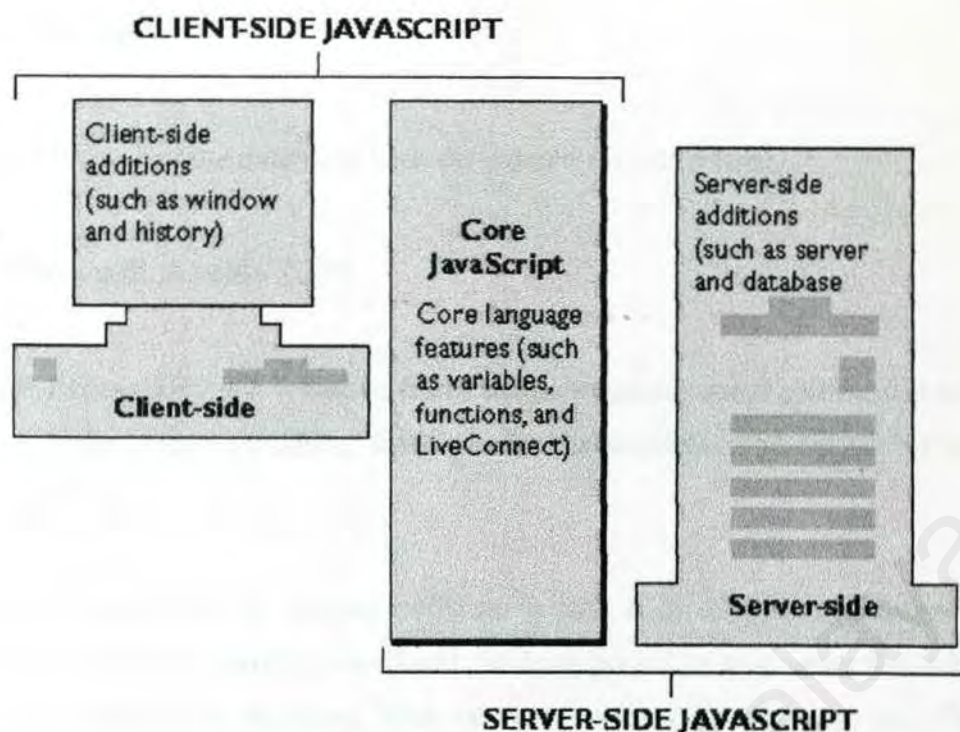


Figure 2.3: Component of JavaScript

Client-side JavaScript (or Navigator JavaScript) encompasses the core language plus extras such as the predefined objects only relevant to running JavaScript in a browser. *Server-side JavaScript* encompasses the same core language plus extras such as the predefined objects and functions only relevant to running JavaScript on a server. This guide provides information and instructions for using the core and client-side JavaScript.

2.6 Database

In below are some of the databases been considered for our system.

2.6.1 Microsoft Access 2000

Microsoft Access 2000 is a Windows-based database management system. It is one of the programs in the Microsoft Office. It runs under the Windows 95/98/2000/NT operating systems. [19]

Microsoft Access 2000 is flexible software where it is used as a database tool in developing systems. It provides relational database power to give users the information they need to make better decisions. With Microsoft Access 2000 you can enter, store and manipulate your personal or company data in a variety of ways. It integrates data from spreadsheets and other databases, and to share information over Intranet and the Internet, and build faster business solutions. It has several features that integrate network, Intranet and Internet features, allowing user to produce professional reports on paper, online or in HTML.

In addition, Microsoft Access 2000 is cheap, easy to be obtained and user friendly. In Microsoft Access 2000, data is divided into separate storage containers called table. The data can be viewed, added and updated by using form. Retrieval of data is done by using query. Report is used for analyzing and printing. These characteristics enable the data to be maintained, operated, updated and managed easily.

2.6.2 Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 makes giant strides in performance, reliability, and scalability, giving your organization many opportunities to create intelligent, real-world business solutions. [11]

From the beginning, Microsoft designed SQL Server to make it easy for database administrators to build, manage, and deploy business applications. This latest version automates standard database administration operations. Plus, sophisticated new tools simplify managing complex operations.

Microsoft SQL Server is a significant tool in many regards. From data warehousing to applications that require not only a large amount of information, but also many different simultaneous users, SQL server is a key component in answering data management requirement. It is a comprehensive database.

Microsoft SQL Server is a perfect example of an n-tier system. The user can manipulate the data directly from the client side. Most of the time, the data is validated first before it is updated into the database in server side. It is the best database for Windows NT server.

Microsoft SQL Server maintains referential integrity and security and ensures that operation can be recovered in the event of numerous types of failure. SQL server can control the access for the type of information that can be retrieved by the user.

SQL Server scales from laptop databases and small business servers all the way up to terabyte-size databases. It integrates well with existing applications, and it provides a cost-effective environment for customizing and building new applications that meet unique requirements.

SQL server supports Internet database integration. It allows the user to automate the publishing of database information in HTML documents. It allows us to build active web sites and let us conduct processes on the Internet. When combining with Internet Information Server and the SQL Server Internet Connector, it gives user the complete Internet database publishing capabilities.

It provides the function for transparent distributed transactions. This means that it provides automatic distributed update capability across two or more SQL server transparent to the desktop application, making it a simple to use. It guarantees the integrity of transaction of updating spanning multiple servers.

Microsoft SQL Server 7.0 allows two billion tables within each of 32,767 databases to be defined. The number of rows in a table is effectively unlimited. It allows user to define up to 250 columns for each table. SQL server allows user to combine column from as many as 16 tables in a single query.

2.6.3 Oracle8i

Oracle8i is one of the most stable databases available in the market. It can run on almost on every platform. Oracle8i supports Java natively inside the database (the developer can write in PL/SQL or Java). It also supplies a lot of built-ins. No other databases have this level of integration with Java. [12]

Besides that, it is also designed as an Internet development and deployment platform. Oracle InterMedia enable Oracle8i to manage text, documents, image, audio, video, and locator data in an integrated fashion with other enterprise information. It includes Internet services supporting popular web client interfaces, web development tools, web servers, and streaming media servers.

Oracle WebDB allows non-programmers to easily develop web database applications and still have time to concentrate on other jobs. Oracle8i's Java offering Oracle Jserver Option, that is the Java Virtual Machine (Java VM), it can runs within the Oracle8i database server address space.

Oracle8i also supports data partitioning with hash and composite partitioning. It has features such as enhanced ANALYZE, statistics management capabilities, indexes based on functions and transportable table spaces, which are useful in Very Large Database Support (VLDB) and can handle much larger databases – multi-terabytes of data. Oracle8i also includes some features that maintain high availability such as automate standby database (improved standby feature from Oracle 7 and 7.3), fast-start fault recovery feature (a new data recovery feature) and online index builds or rebuilds (allowing users to update and query the base table while creating the index). Oracle8i provides Oracle Parallel server and resource manager, which is for better scalability. For the cost, Oracle is quite expensive, it cost around US40,000 for the software and US100,000 for professional setup.

2.7 Web Server

In below are some of the web servers been considered for our system.

2.7.1 Personal Web Server (PWS) 4.0

Personal Web Server is Microsoft's slimmed-down web server, which provides a basis on which to develop corporate networked applications. Personal Web Server 4.0 can be run on any home machine that runs Windows 95, Windows 98 or Windows NT Workstation. Just like IIS 4.0, PWS 4.0 comes with ASP 2.0 as standard. With PWS 4.0, ASP programs can be created and test just as though as an external user, running on the same machine.

2.7.2 Microsoft Internet Information Server 4.0

ISS 4.0 is designed to deliver a wide range of Internet and intranet server capabilities. Below are several important features of ISS 4.0:

- *Integration with Windows NT Server*

Due to tight integration with Windows NT Server, IIS is easy to setup and manage, fast, and secure.

- *Enable easy-to-develop, powerful Web-based applications*

IIS introduces Active Server Pages, which enables posting of dynamic content and development of web based applications to be carried out easily.

- *Easy installation for secure environment*

IIS includes security features that are easy to install. It works closely with Microsoft Transaction Server to access databases and to provide control at the transaction level. [15]

2.8 Platforms and Operating System

Below are some of the platform and operating system been considered for our system.

2.8.1 Microsoft Windows NT Server 4.0

Microsoft Windows NT Server 4.0 is a popular network operating system. It is designed to help developers to build and deploy business applications ever faster than before.

Below are several important features of Windows NT Server 4.0:

- Window NT is a powerful operating system that provides an integrated environment for developing client/server application. It provides ease-to-use interface and a set of management tools such as assistance to set up web site, simplified access to resources and so on.
- Window NT Server is a complete platform for building and hosting web-based application. It supports multiple Web sites on a single machine. It also provides innovative web publishing features, customizable tools and new wizard technologies. Thus, Windows NT is the best platform to publish and share information securely over corporate intranets and the Internet.
- Extensive security support is provided in Windows NT. This enables the implementation of user access control in certain file or application.
- Windows NT supports a wide range of networks protocol and Remote Access Protocol. This simplifies the development process of distributed application.
- Windows NT allows Object Linking and Embedding (OLE). Information from several applications can be combined into one compound document using the special OLE capabilities of window-based application. [17]

2.8.2 Microsoft Windows 2000 Server

Windows 2000 Server is the entry-level version of the server family. The multipurpose network operating system for business of all sizes, it is the perfect solution for file, print, intranet, and infrastructure servers. It scales from 1 to 4 processors and up to 4 gigabytes. Building on the strengths of Windows NT Server 4.0, the Windows 2000 Server Family delivers three increasingly powerful products that set a new standard for reliability and scalability. The Windows 2000 Server Family also demonstrates how well an operating system can be integrated with a standards-based directory, Web, application, network, file and print services, and end-to-end management. This combination of reliability and functionality provides the best foundation for integrating your business with the Internet. [18]

Active Server Pages (ASP), first introduced as a component of Windows NT Server 4.0, revolutionized the way Web content was served. This technology allowed organizations to create dynamic and highly personalized Web sites. The implementation of Active Server Pages in Windows 2000 Server is faster, more reliable, more scalable, and ready to run on high-end multi-processor hardware.

Windows 2000 Server also introduces new technologies that let you build richer Web applications and solutions, such as the next generation of the Microsoft Component Object Model, COM+. Developers using COM+ find it much easier to create and use software components, and benefit from a runtime environment and services that are easily used from any programming language or tool.

In addition, Windows 2000 includes integrated support for streaming media, which allows organizations to develop and distribute real-time presentations and rich multimedia content to both internal and external audiences. Imagine being able to send full screen video to your users' desktops on demand, while providing CD-quality audio, digital rights management, and great integration with other application software.

2.9 Overview Of The Existing System

The overview of the existing system is to determine the advantages and the weaknesses of the system. Therefore, it can become the guidelines to develop the more advance new system by taking the advantages and avoiding the weaknesses. The online hotel room reservation system in the Internet (<http://www.hotels-moscow.ru/>) is taking for this purpose because it is quite similar with the **ORRASH**. [14] In observation, the advantages in the existing system can be simplified as follows:

2.9.1 Advantages of the Existing System

- ✓ **24 Hours Accessible**

User can access the Web site anytime (24 hours per day) they like to book a room in the hotel as long as the user is connected to the Internet.

- ✓ **Accessible At Difference Places**

As long as the user has a computer and connected to the Internet, he/she can book a room in the hotel at anywhere, in the office, at home, colleges, cyber cafes or any place without go to the hotel counter.

- ✓ **Security**

The security in Internet is not a problem. This is because there exist a wide range of security such as firewalls, digital signatures.

- ✓ **Always Updated Information**

User will never book a room that is already booked or resided by other user because when a room is booked, it will be recorded in the database and cannot be booked by anymore until the room's customer moves out.

2.9.2 Weaknesses Of The Existing System

The weaknesses of the existing system can be simplified as follows:

✖ Too Much Information

The Web pages is full of text information, it is too complicated and not easy to understand. Most customers do not have the time to read or even browse through the information. This will discourage people to visit the site.

✖ Poor User Interface Design

The Design of the user interface is full of text and the background color is too bright. This will let the user feel inconvenient and uncomfortable.

✖ Too Many Field To Fill In

The booking form contains a lot of field to be filled in and some of the field is not so understandable, user does not know what to be filled in that field. This makes the user feel inconvenient.

✖ Lack Of Description Of Rooms

The Web site provides poor description of the hotel room. The users only know the room is for how many people and the price for it. They do not know what are the facilities provided for the room and the design of the room. Without enough information, user would not feel interesting to book the room.

Chapter 3: Methodology & System Analysis

3.1 Project Formulating

The software development process is a series of steps that lead to the final product. It is a process that involves the analysis, design, development, testing, and deployment of a system. The process is iterative, meaning that it can be repeated as many times as needed to refine the system and improve its quality.

Chapter 3: Methodology & System Analysis

The process starts with the identification of the problem and the requirements. This is followed by the analysis of the problem and the design of the system. The design is then implemented as a prototype, which is tested and refined. The final step is the deployment of the system and the evaluation of its performance. The process is iterative, meaning that it can be repeated as many times as needed to refine the system and improve its quality.

Chapter 3: Methodology & System Analysis

3.1 Project Prototyping

The software prototyping methodology is used due to the fact that it allows the entire system to be constructed quickly. This will assist us in understanding and clarifying issue, which is uncertainly and therefore will reduce the risk found in the development. [23]

Prototyping is a technique for building a quick and rough version of a desired system or parts of that system. The prototype illustrates the system to users and designers. It allows them to see flaws and invent ways to improve the system. It serves as a communications vehicle for allowing person who requires the system to review the proposed user interaction with the system.

- A prototype is used where the functions and detailed design of a system are not yet fully understood.
- The prototype is used to explore and solidify the functions and design.

As with all other software development prototyping begins with requirements gathering. After identify the requirements, a design is formulated. It focuses on the top-level architecture and data description issue. This causes the construction of a prototype. This prototype is to be tested and evaluated to refine the requirements. This process iterates until the prototypes has evolved into a production system. This prototyping model consists of the following 6 steps illustrate in **Figure 3.1**.

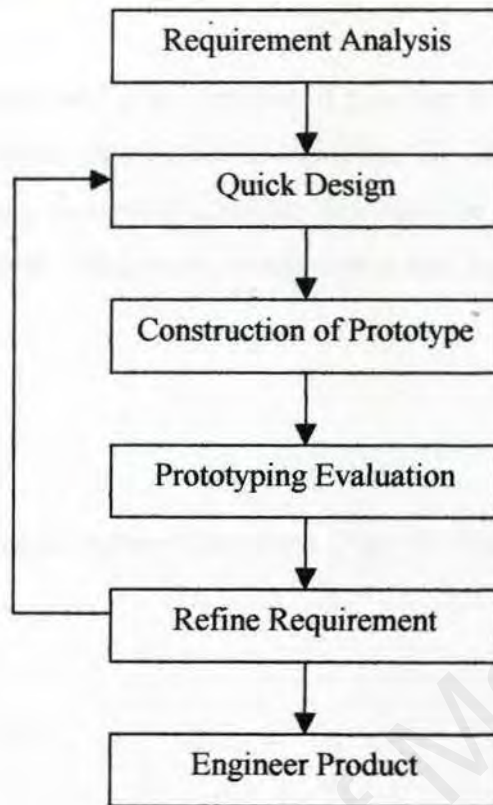


Figure3.1: Prototyping Model

Requirement Analysis

Prototype begins with requirement gathering and analysis. In this step, developer and user meet and define the overall objective for the software, identify whatever requirement are known, and outline areas where further definition is mandatory.

Quick Design

After the requirement analysis phase is done, a “quick design” takes place. The quick design focuses on a representation of those aspects of the software that will be visible to the user. This phase is the construction of a prototype.

Construction of Prototype

Prototype software is created and it will consist of program to move data back and forth between screens, the database, report, and the input and the output used in the interface. At first, these programs may do little processing; they may use “hard coded” data instead of extracting it from database. The prototype software is also tested and redefined.

Prototype Evaluation

The user evaluates the tested version prototype. User is allowed to test and suggest modification.

Refine Requirement

The prototype evaluation step is used to redefine requirement for software to be developed. A process of iteration occurs as the prototype is “turned” to satisfy the needs of the customer while understand what needs to be done. At this step, not only requirements are justified, but also new requirements are added when possible.

Engineer Product

Step from quick design to redefine requirement is interactively until all requirements are formatted or until the prototype has amend into a production system.

3.1.1 Uses of Prototyping

Prototyping is particularly valuable in the following situations:

- There is scope for user creativity to improve the system.
- Users are unsure of exactly what they want.
- The users do not understand all the impacts of the new system.
- Screens and reports should be checked with management to see if they can be made more useful or easy to use.
- The users have difficulty expressing all the system requirements.
- The prototype may act as a catalyst to elicit alternative ideas.

3.1.2 Advantages of Prototyping

Some of the benefits of prototyping model are:

- In the earlier of development stage, changes still can be made.
- User's needs and expectations can be followed-up more closely.
- Prototyping introduces early reality testing into a project. The users can see what is being built for them and critique it. Without prototyping, there is a substantial risk of building an inadequate system, wrong features or, at worst, a system that users will reject.

- It encourages users to contribute creative input into the design process.
- Prototyping enables errors and weaknesses to be caught before expensive design and programming are done.
- Prototypes provide users with early experience with the system and may be used as training tools.
- Prototyping can give fast development by having the prototype evolve into the final system.

3.2 Project Life Cycle

The software engineering process consists of a set of steps that encompass methods, tools, and procedures. These steps are often referred to as software life-cycle models. A model for software engineering is chosen based on the nature of the project and applications, the methods and tools to be used, and the controls and deliverables that are required.

3.2.1 Waterfall Model

The Waterfall Model with prototyping has been selected as the project life-cycle model. Waterfall model with prototyping process's flow is showed in **Figure3.2**.

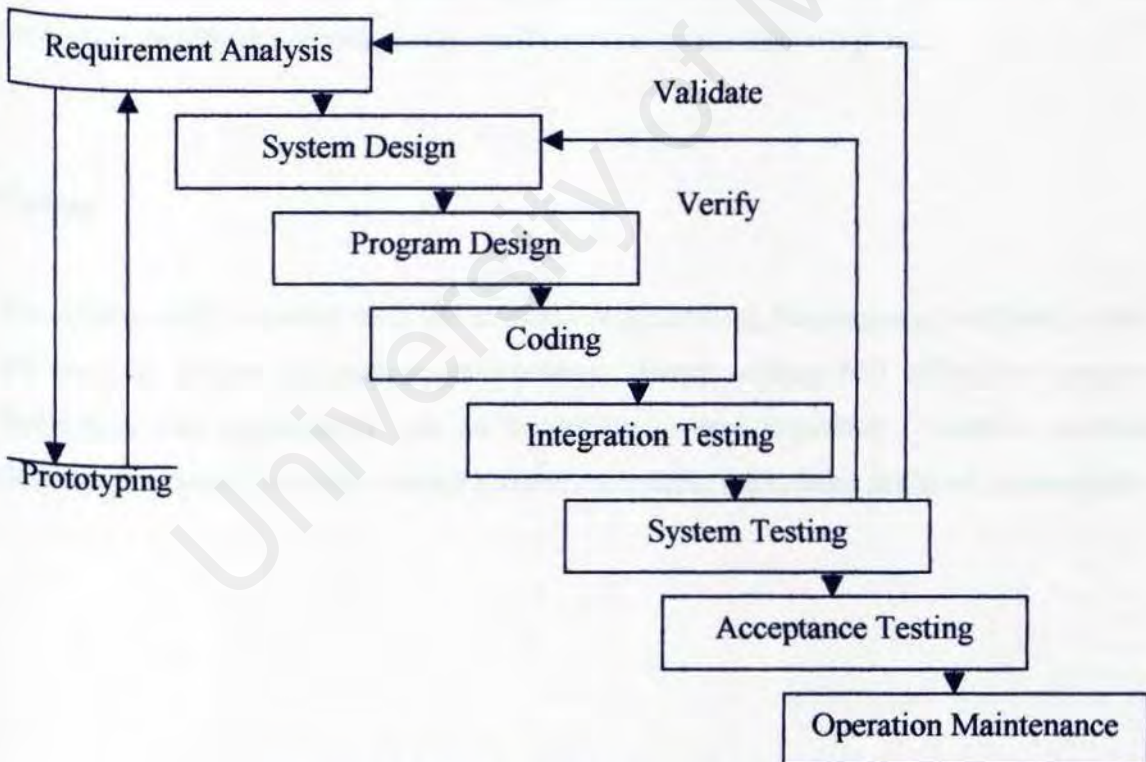


Figure 3.2 Waterfall Model With Prototyping

Requirement Analysis

The first process in the entire methodology process is Requirement Analysis. It mentions about verified specification of the required functions, interfaces, and performance for the software product. To determine the requirement of the system, gathering information is important. There are variety of techniques can be used, this includes sampling and investigating hard data, interviewing, using questionnaires, prototyping and observing decision-maker behavior and the office environment. However, not all of the techniques can be used at the same time, it depends on the situation. [23]

System Design

System Design involves a complete verified specification of the overall hardware-software architecture, control structure, and data structure for the product, along with such other necessary components as draft user's manuals and test plans.

Coding

The system will be coded with the suitable programming languages or software, which are used to design the system environment. Every coding will affect the success, throughput and maintenance job, so this phase is very important. Therefore, suitable development tools should be chosen properly to enable the system designed successfully.

Integration Testing

Testing is the process to find the errors of the existing program. Thus, a successful test is considered to be one, which establishes the presence of one or more errors in the software being tested. Therefore, in this level, all of the units are combined and now the whole is tested. When the combined programs are successfully tested the software product is finished. No prototype will be developed in the last few phases due to the time constraints.

3.2.2 Advantages of Waterfall Model

The waterfall model has the following main advantages:

- It is easy to identify milestone.
- It is easy to separate one stage from another.

3.2.3 Disadvantages of Waterfall Model

The waterfall model has the following main disadvantages:

- It implies that any stage should be frozen before continuing with the later stages (resulting in premature requirements, design, coding, etc.)
- It assumes that user requirements can be precisely specified. Unfortunately customers rarely know precisely what they want, and software engineers rarely understand the business context of their customers.

3.3 System Requirement

The system analysis phase is needed to identify the system requirement. The system's functional and non-functional requirement as well as the software and hardware requirement is system requirement needed to support the identified functions. Information gathering is a very important method to determine the system requirement. Approaches will be taken are surfing the Internet, interview, research and discussion.

3.3.1 Information Gathering

3.3.1.1 Surfing the Internet

Internet surfing is a very efficient way of information gathering in today's world. There are a lot of Web site available that provides the useful and expertise information needed for this system. The information is collected and then the information is analyzed to determine the system requirement.

3.3.1.2 Interview

An interview is a directed conversation with a specific purpose that uses a question and answer format. The respondent, who is also the user of the systems, was interviewed to get the required information on the system to be developed. Adequate information provided by the user or the system requirement and existing problems will ease the development of the new system, which will be built to fulfill the requirements mentioned while eliminating those existing problems. [21] Interviews have been conducted with the authority of the University of Malaya Medical Center. He gave a clearer view of the way they store the patient information and the process of booking rooms in the hospital.

3.3.1.3 Research

Research involves reviewing books and journals that contain relevant information. Books and journals were widely used to obtain information. They provide related issues for **ORRASH**. The guideline is based on the information that was gathered from all of the books and journals.

3.3.1.4 Discussion

Discussion is made regularly with supervisor, especially when problem arises and certain clarifications are needed. Advice and guidelines from my helpful supervisor, Prof. Madya Raja Noor Ainon, is very important in developing our project. Discussion is also conducted with other lecturers and friends on certain matter related for better understanding and improvements in **ORRASH**.

3.3.2 Functional Requirement

Functional requirement are function or subsystem that are mandatory to the system. It explain what the system will do, independent from the implementation of the solution. It describes an interaction between the system and its environment to determines functional requirement, a design has to be made on what states are acceptable for the system to be in. The absence of the functional requirement will make the whole system incomplete. [20] The structure of Online Room Reservation And Scheduling System For A Hospital (ORRASH) illustrate in **Figure 3.3**, **Figure 3.4**, **Figure 3.4**, **Figure 3.6** explain the functional requirement. As the project involved two people in the process of development, I will cover the part with gray color.

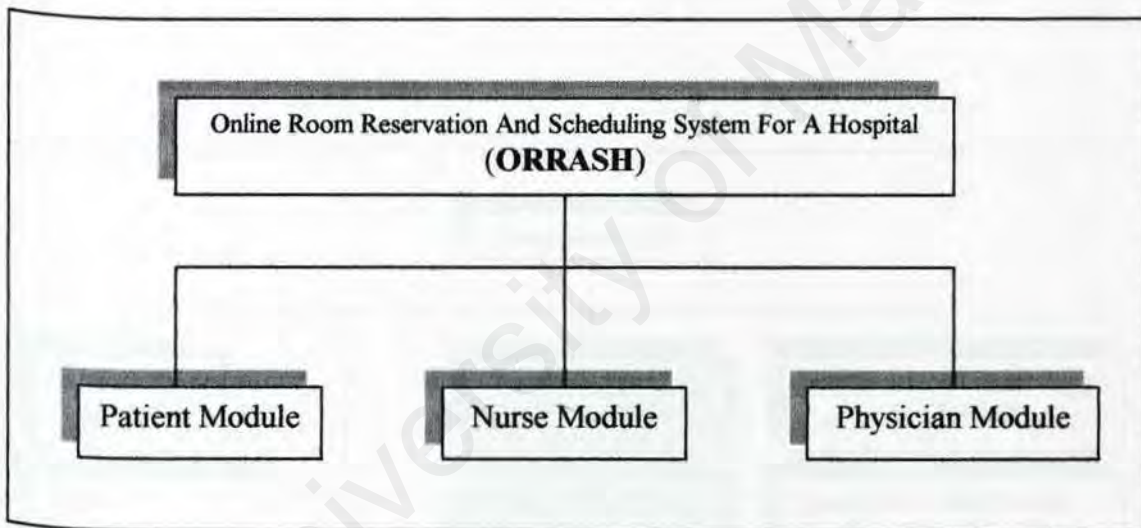
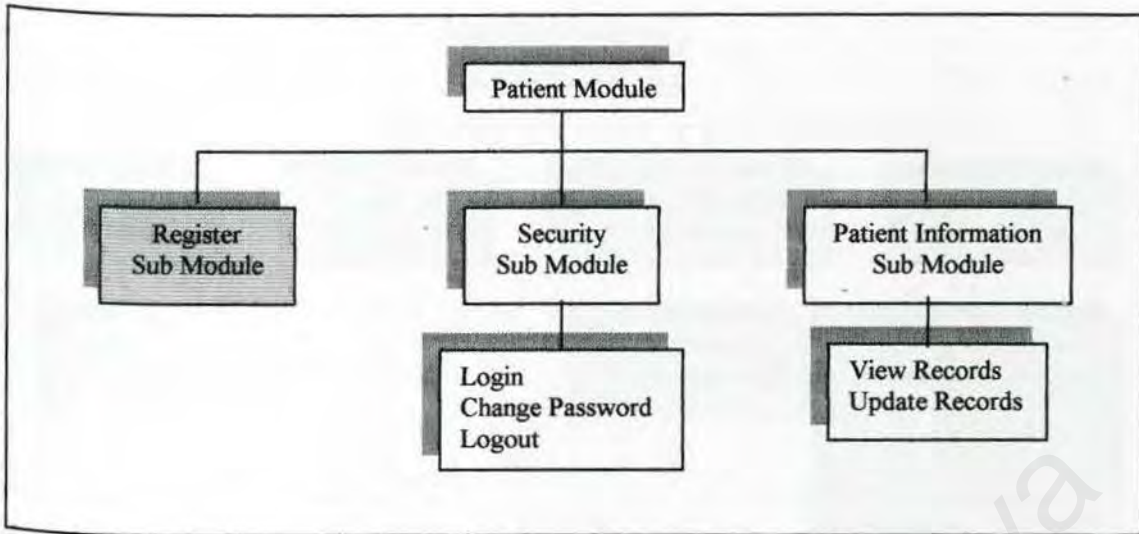
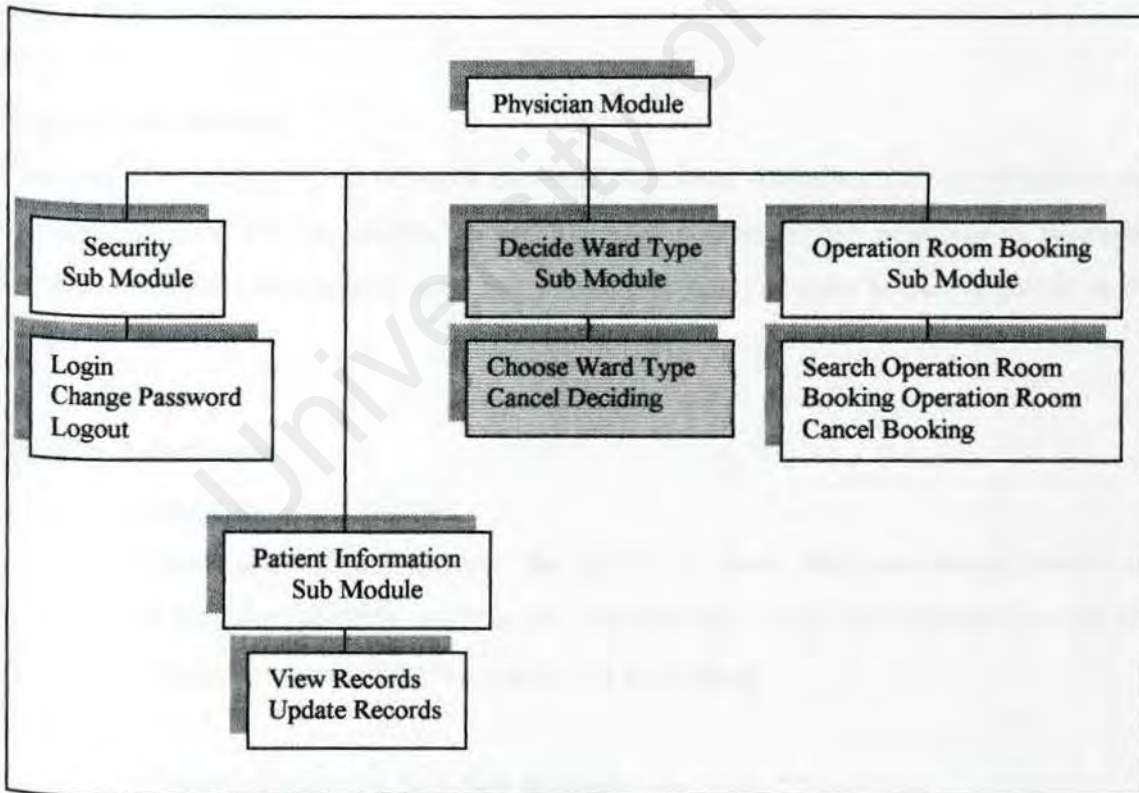


Figure 3.3: Structure of Online Room Reservation And Scheduling System For A Hospital (ORRASH)

**Figure 3.4: Patient Module****Figure 3.5: Physician Module**

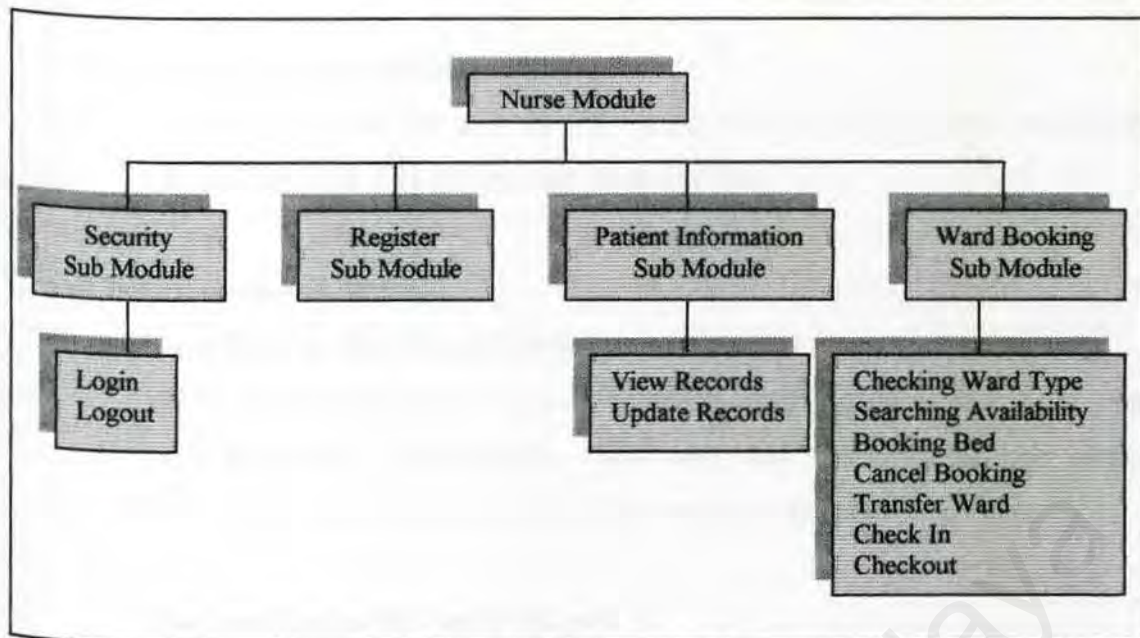


Figure 3.6: Nurse Module

3.3.2.1 Patient Module

Register Sub Module

User (patient) is required to fill up a personal data form including the identification and demographic data. This form also requires the user to input his/her new user id, password and password hints that will be used every time they want to login to the ORRASH in the future.

Security Sub Module

- Login Sub-Sub Module

Allow patients to login into the system by given their user id and password. When the patients were login successfully, they are allowed to use the functions provided by the system for the patient.

- Change Password Sub-Sub Module

Allow patients to change their password and user id to provide the security.

- **Logout Sub-Sub Module**

Allow patients to logout from the system to prevent the others to use the functions provided by the system for the patient.

Patient Information Sub Module

- **View Records Sub-Sub Module**

When the patients were login successfully, they are allowed to view their medical records, demographic data and ADT records in this Patient Information sub module by selecting to view information.

- **Update Records Sub-Sub Module**

Only changes to patient demographic data are allowed in this Patient Information sub module. On the other hand, patients are not allowed to do any changes to their medical records or ADT records.

3.3.2.2 Physician Module

Security Sub Module

- **Login Sub-Sub Module**

Allow physicians to login into the system by given their user id and password. When the physicians were login successfully, they are allowed to use the functions provided by the system for the physician.

- **Change Password Sub-Sub Module**

Allow physicians to change their password and user id to provide the security.

- **Logout Sub-Sub Module**

Allow physicians to logout from the system to prevent the others to use the functions provided by the system for the physician.

Patient Information Sub Module

- **View Records Sub-Sub Module**

When the physicians were login successfully, they are allowed to view patient's medical records, demographic data and ADT records in this Patient Information sub module by selecting to view information.

- **Update Records Sub-Sub Module**

Only changes to patient's treatment records are allowed in this Patient Information sub module. On the other hand, physicians are not allowed to do any changes to patient's personal data.

Decide Ward Type Sub Module

- **Choose Ward Type Sub-Sub Module**

Allow physicians to choose the type of ward for the patient and decide the check in and checkout dates for the ward depend on the patient's situation. Physicians have to fill up a ward type deciding form and press "Submit" button to complete the ward type choosing process.

- **Cancel Deciding Sub-Sub Module**

Allow physicians to cancel the ward type decided records that have send to the database. The records, which are stored in the database, will be deleted.

Operation Room Booking Sub Module

- **Search Operation Room Sub-Sub Module**

Physician can search the available operation room based on the purpose that the operation room will be used. After the related operation room found, physician can view its information.

- **Booking Operation Room Sub-Sub Module**

Physician can book an operation room as a preparation for surgery treatment in this sub module. If the time selected by physician was available, the system will requires he/she to fill up an room booking form and press "Submit" button to complete the booking process.

- **Cancel Booking Sub-Sub Module**

Allow physicians to cancel the booking of an operation room. The status of the operation room will change to available and can be booked by other physicians next time.

3.3.2.3 Nurse Module

Security Sub Module

- **Login Sub-Sub Module**

Allow nurses to login into the system by given their user id and password. When the nurses were login successfully, they are allowed to use the functions provided by the system for the nurse.

- **Logout Sub-Sub module**

Allow nurses to logout from the system to prevent the others to use the functions provided by the system for the nurse.

Register Sub Module

Allows nurse to help the patient to fill up a personal data form including the identification and demographic data. This form also requires the patient to input his/her new user id, password and password hints that will be used every time he/she want to login to the ORRASH in the future.

Patient Information Sub Module

- **View Records Sub-Sub Module**

When the nurses were login successfully, they are allowed to view the patients' medical records, demographic data and ADT records in this Patient Information sub module by selecting to view information.

- **Update Records Sub-Sub Module**

Allow nurses to add, delete, or modify certain field in the patient demographic data, medical records or ADT records in this Patient Information sub module.

Ward Booking Sub Module

- **Checking Ward Type Sub-Sub Module**

Allow nurses to check the patient ward type from the database, which is given by the physician.

- **Searching Availability Sub-Sub Module**

Allow nurses to search the available beds in wards for the ward type given by the physician.

- **Booking Bed Sub-Sub Module**

Allow nurses to book a bed in a ward, which is available for a patient. The bed, which is booked, will be updated into the database and cannot be booked by other patient until the patient checkout from the bed.

- **Cancel Booking Sub-Sub Module**

Allow nurses to cancel the booking of a bed in a ward. The status of the bed in the ward will change to available and can be booked for the next time.

- **Transfer Ward Sub-Sub Module**

Allow nurse to transfer a patient from a ward to another ward because the patient request for it or due to some conditions.

- **Check In Sub-Sub Module**

Let a patient to check in for a ward, and the ward cannot be checked in or reserved by other patient until the patient resides in the ward checkout.

- **Checkout Sub-Sub Module**

Let a patient to checkout from a ward, the ward become available and can be checked in or reserved by other patient.

3.3.3 Non-Functional Requirement

Non-functional requirements are essential definition of the system properties and constraints under which a system must operate. [22] Non-functional requirement are as important as functional requirement. A non-functional requirement describes the features will be included in the system to provide a feasibility and ease of use to the users. The following states the non-functional requirement for the **ORRASH**.

- User Friendly and Usability
- Reliability
- Maintainability and Expandability
- Modularity
- Response Time

3.3.3.1 User Friendly and Usability

The system can be considered as an attractive or as an easy-to-use application because the users only have to click on the task or image by using the mouse. The user of the system will be more confidence using the system due to the usage of suitable and meaningful icons or buttons. The use of menu should give the user sufficient information to use the

system. Any non-trivial process such as updating or deleting any records should have confirmation message and error message to be displayed to make sure that the user could do final decision before certain action is taken.

3.3.3.2 Reliability

If a system does not produce dangerous or costly failure when it is used in a reasonable manner is said to have reliability. The system to be developed must be reliable because reliability is one of the essential software qualities. It is crucial in maintenance and operational, as frequent breakdown of the system will increase the cost of maintenance and development for enhancing and debugging the system. Therefore, it should process the input data and produce the expected output without any errors.

3.3.3.3 Maintainability and Expandability

To possess a high degree of maintainability and expandability, the system must also be designed to be understood, corrected, adapted, and able to be enhanced without much difficulty. Architecture components, algorithm, data structure and procedures design should be able to extend and modify with ease. This is important so that any future enhancements and expansion can be done easily.

3.3.3.4 Modularity

Software architecture of the system should embody modularity, that is, software is divided into separately named and addressable components, called module, which is integrated to satisfy problem requirements. This is done to isolate function codes from one another. Therefore, testing, debugging and maintenance can be done easily.

3.3.3.5 Response Time

The response time to retrieve the data and information should be in a reasonable interval time. This means that all desirable data and information should be available to users at any point in time.

3.3.4 Hardware Requirement

3.3.4.1 Server Side Hardware Requirement

The server computer requirements are:

Processor	At least with a Pentium 266MHz processor
Memory	64 MB RAM
Disk space	4.0 GB
Other	Other standard computer peripherals

Table 3.1: The server side hardware requirements for developing ORRASH

3.3.4.2 Client Side Hardware Requirement

The client side hardware requirements are quite minimal as long as it has enough RAM and a reasonable fast modem to connect to the Internet.

The recommended configurations are:

- At least 32 MB of RAM.
- Network connection through existing network configuration or modem.

3.3.5 Software Requirement

This section will state out the reasons why the software are used to run or develop the system.

3.3.5.1 Programming Language

On both server-side and client-side scripting, both VBScript and JavaScript will be used to write ASP based on the suitability on performing certain functions in ASP architecture.

3.3.5.2 Database

Microsoft SQL Server 7.0 is chosen as the database server, as it is used for manipulation and storage of large amount of data. It consists of relational tables, database structure, data, stored procedures and task scheduling.

3.3.5.3 Web Server

Given that vast different web server, it is decided that this project will use Internet Information Server (IIS) 4.0 as the web server. The reason on choosing this web server is due to its support for ASP and tight integration with Windows 2000 server Family.

3.3.5.4 Platform and Operating System

Microsoft Windows 2000 Server is chosen, as it is a complete and powerful platform that provides server operating system. Moreover, Microsoft Windows 2000 Server is tightly

integrated with the Windows Exchange 2000 Server. Windows 2000 Server serves as a platform to publish and share information in a secure way over Internet and intranets.

3.3.5.5 Development Tool

Microsoft Visual InterDev is chosen, as it support both client-side and server-side scripting. Moreover, it supports both VBScript and JavaScript to write ASP. Microsoft Visual InterDev can connect to Microsoft SQL Server 7.0.

Chapter 4:

System Design

Chapter 4: System Design

System design is an important stage of system development where the requirements for the system are translated into the system characteristics to meet the user requirement and satisfaction. It is a creative process of transforming problems into a solution. The system design includes a complete description of the functions and interactions involved. Although the system design phase of a software project describes only appearance and functionality but it is a key factor in determining the success and accomplish of a software project.

To design a system is to determine a set of components and inter-component interfaces that satisfy a specified set of requirements. There are many ways to create good designs. The choice is based on designer preferences. However, every system design involves some kind of decomposition. It is a process that start with a high-level depiction of the system's key elements and creating lower level looks at how the system's features and functions will fit together.

4.1 Data Flow Diagram

Data flow diagram is a graphically characterization of data process and flows in a system. Data flow diagram depicts the broadest possible overview of system inputs, processes and outputs, which corresponds to data movement through the system.

DFD for the **ORRASH** will be separated into smaller modules. Therefore it will be drawn based on the modules and functions as it was divided out in the system. **Table 4.1** explains the components of DFD.

These are the objectives of DFD

- 1) To show the movement of data between the system and its environment.
- 2) To document the infra system information flows.
- 3) To graphically document the boundaries of the system.
- 4) To provide a hierarchical functional breakdown of the system.
- 5) To aid communications.



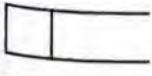

Component	Description
	Entity, that can send or receive data from the system and is considered as outside of the boundaries of the system.
	Process, which transforms within the system. A process is represented by a rectangle which has three parts; identifier, location or person. Performs the process and the process name.
	Data store; where the data is held for a time within the system. It consists of two parts; identifier and the simple description of data stored.
	Data flow is a directed line, which represents the information from between two objects. The arrow denotes the direction of the data flow. Each data flow is labeled with the name or details of the information represented by the data flow.

Table 4.1: Components of DFD

The following figures show the DFD of **ORRASH**.

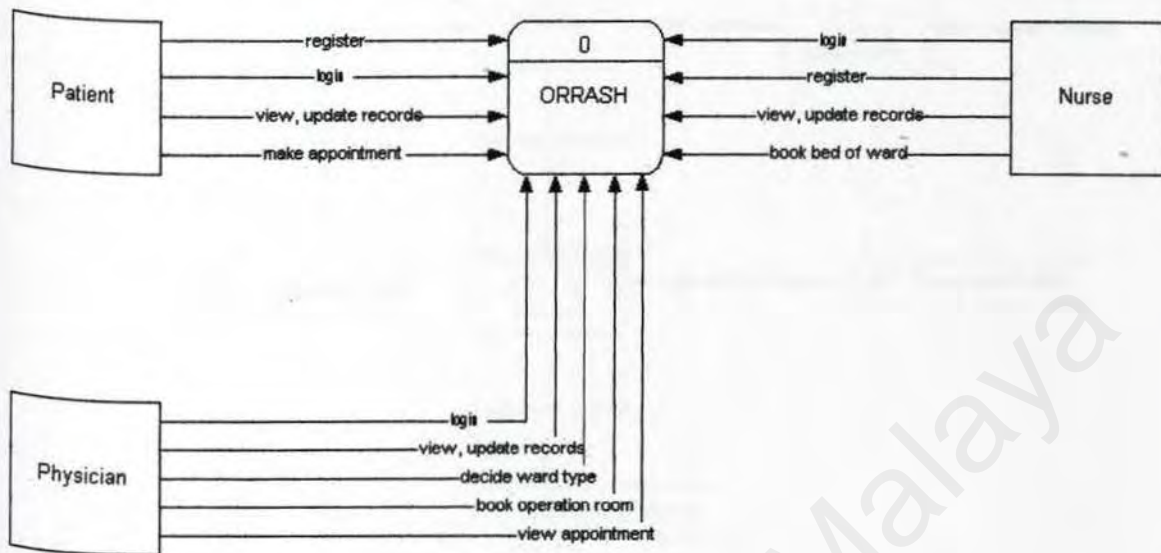


Figure 4.1: Context Level Diagram for ORRASH

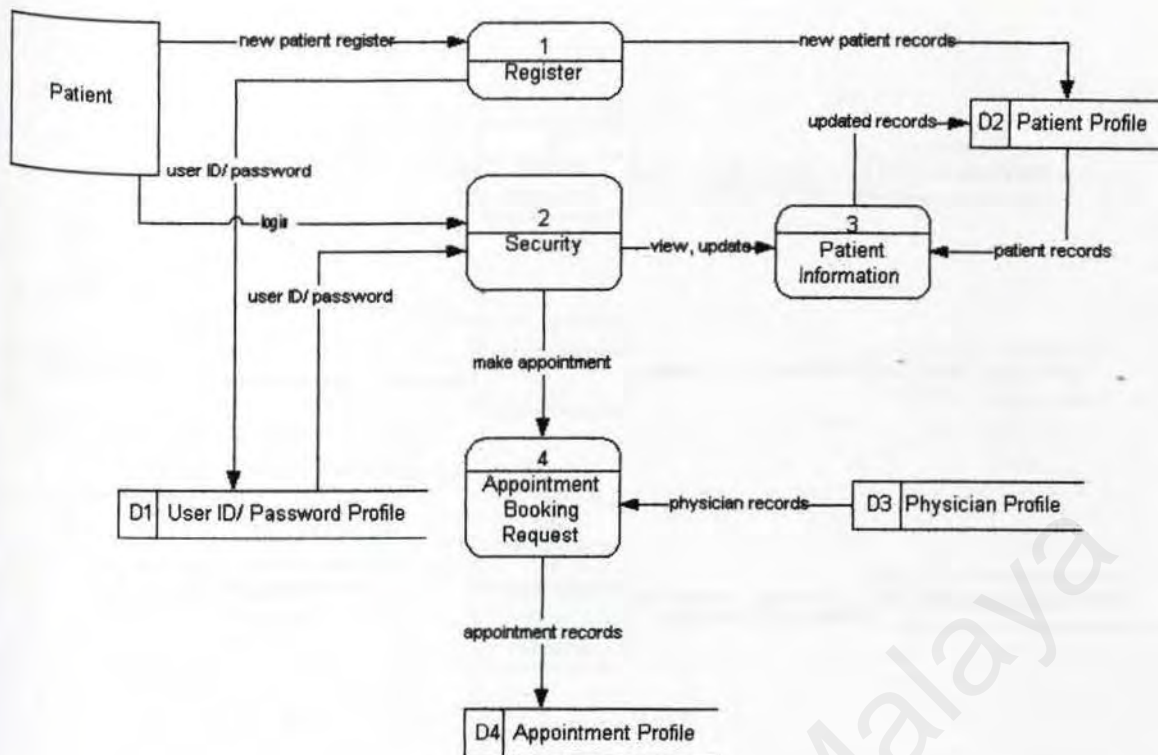


Figure 4.2: DFD for Patient Module

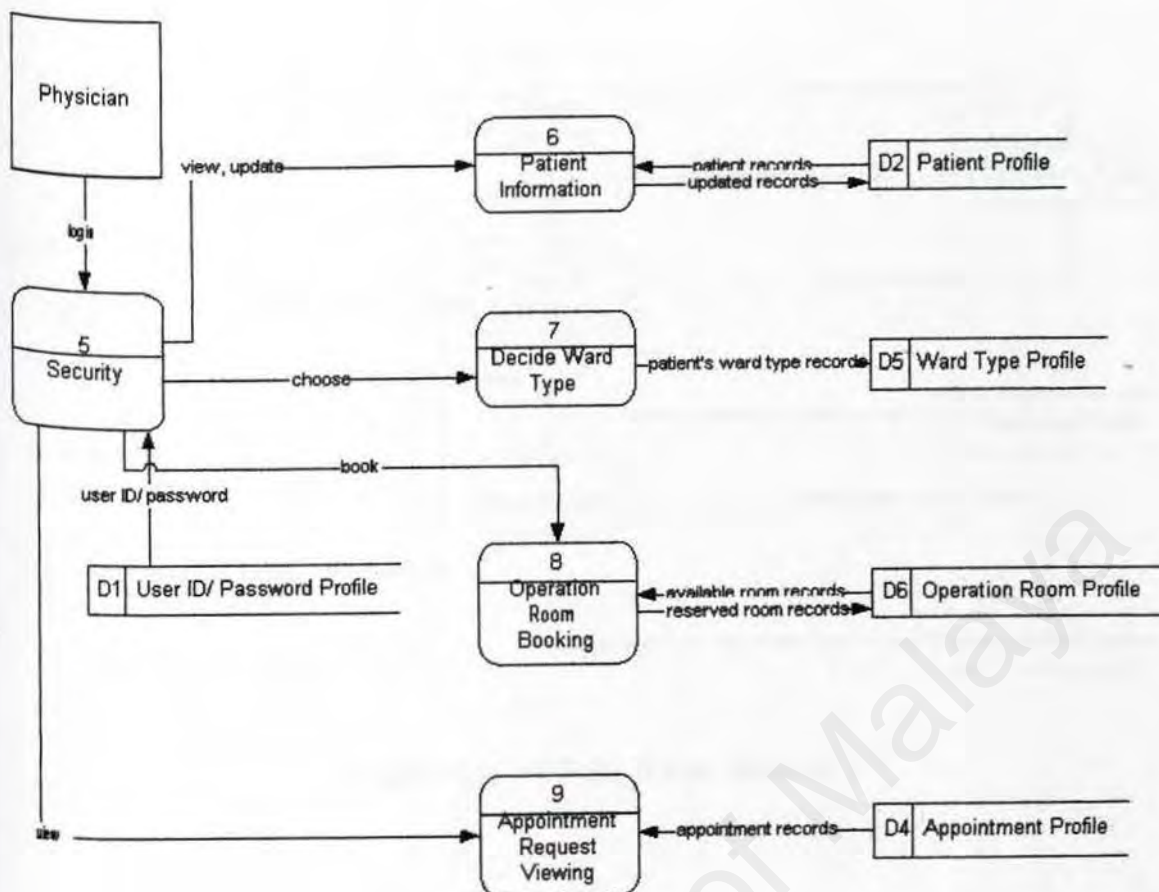


Figure 4.3: DFD for Physician Module

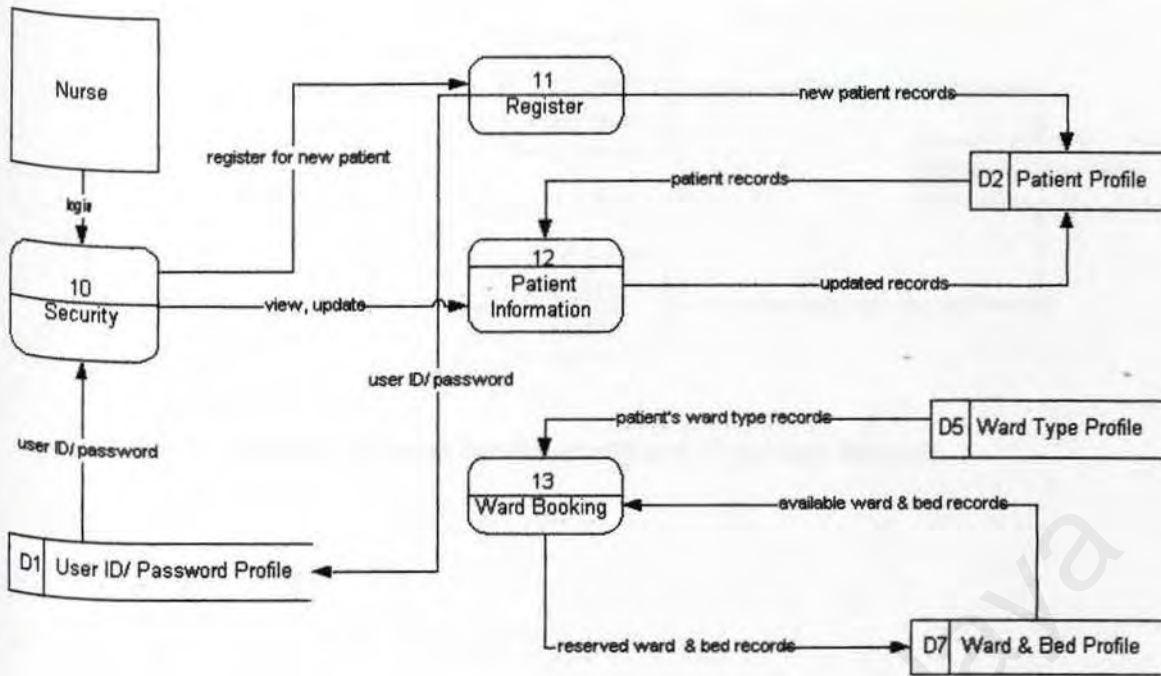


Figure 4.4: DFD for Nurse Module

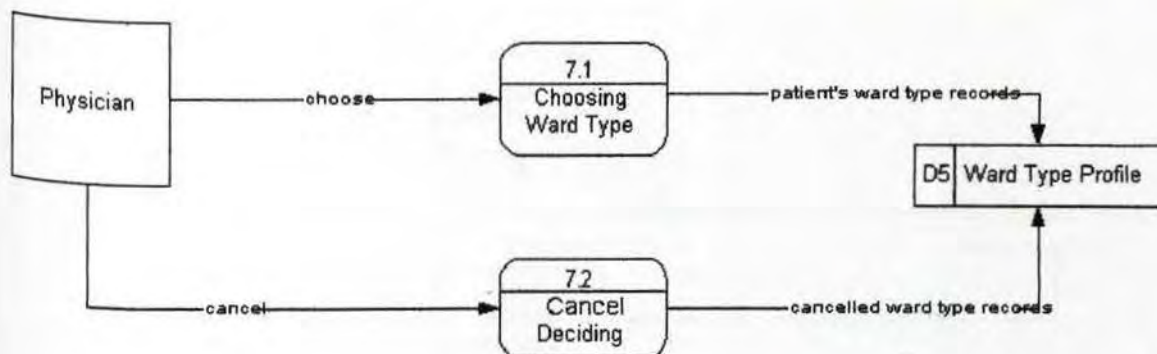


Figure 4.5: DFD for Decide Ward Type Sub Module

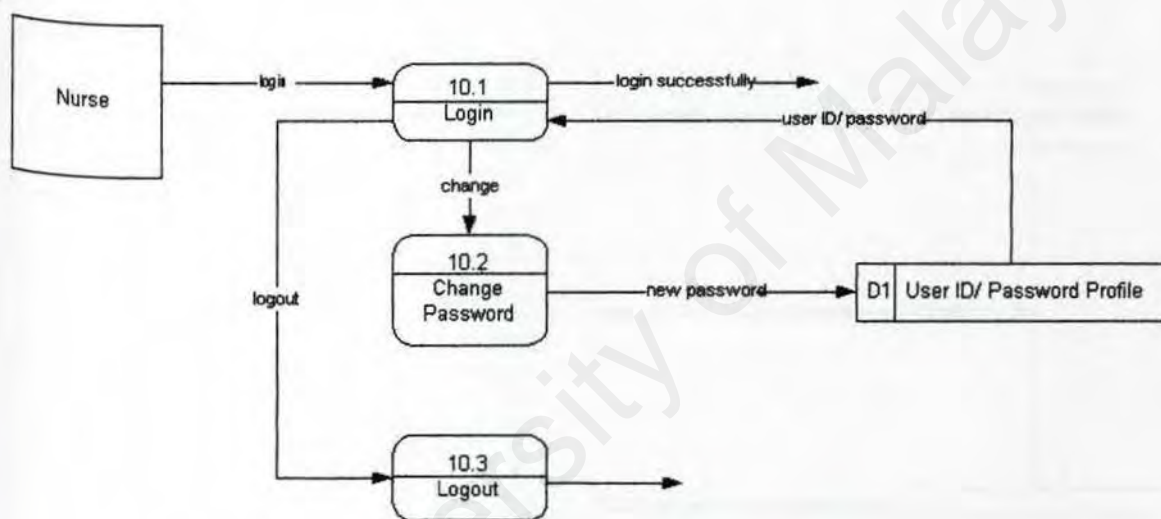


Figure 4.6: DFD for Security Sub Module

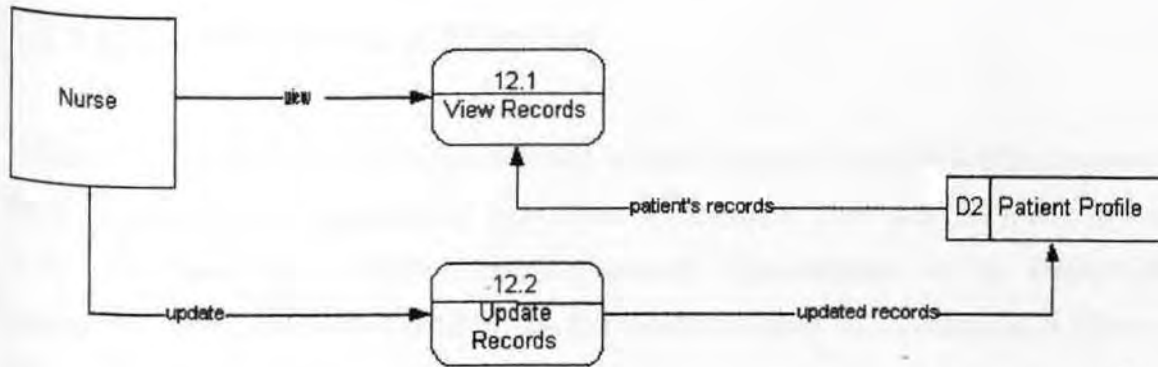


Figure 4.7: DFD for Patient Information Sub Module

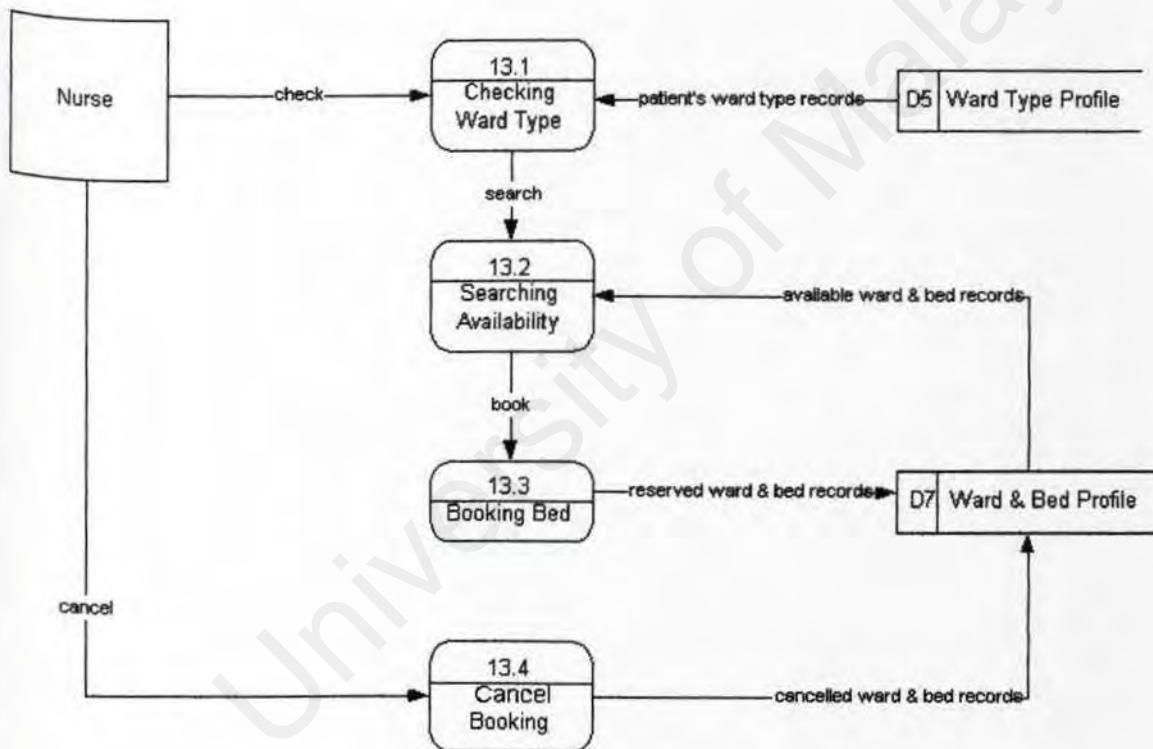


Figure 4.8: DFD for Ward Booking Sub Module

4.2 Entity-Relationship Diagram

Relationships are associations between tables in the database. In the **ORRASH** database, there are three types of established inter-table relationships; one: one (1:1), one: many (1:N), and many: many (M:N). The diagrammatic representation of the **ORRASH** database relationships is illustrated in the Entity-Relationship (E-R) diagram in **Figure 4.9**.

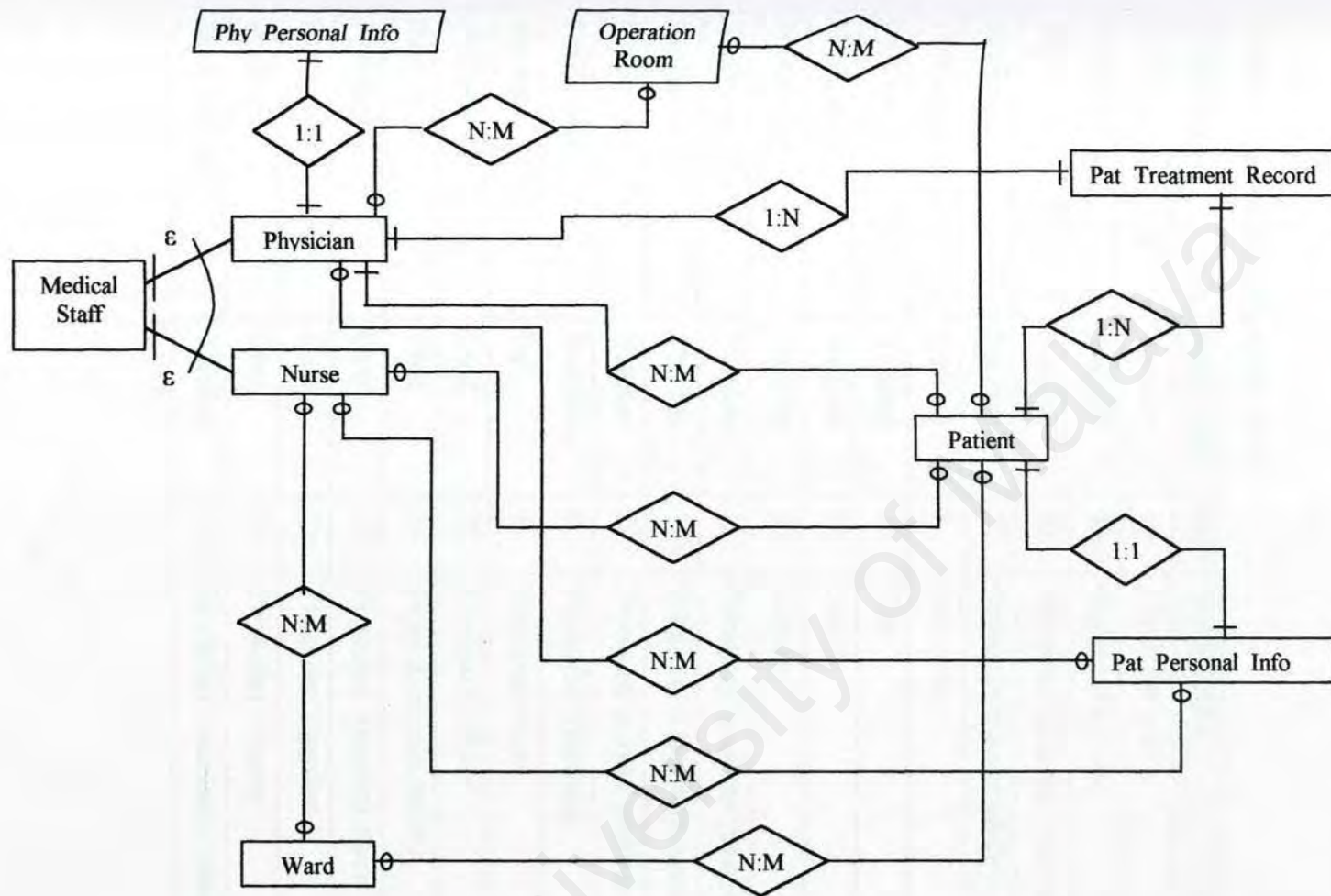


Figure 4.9: Entity-Relationship Diagram for ORRASH

4.3 Data Dictionary

Data dictionary is a specialized application of the kinds of dictionaries used as references in everyday life. The data dictionary is a reference work of data about data compiled by system analysts to guide them through analysis and design.

These are the following tables used in the ORRASH.

Table 4.2: Physician_Personal_Info

Field Name	Data Type	Size	Description
phyID (primary key)	Varchar	10	Physician's ID
phyName	Varchar	30	Physician's name
phyIC	Varchar	20	Physician's identity card number
phyAdd	Varchar	50	Physician's home address
phyPostcode	Varchar	5	Post code
phyCity	Varchar	20	City
phyState	Varchar	20	State
phyCountry	Varchar	20	Country
phyPhone	Varchar	15	Physician's contact number
phyFacsimile	Varchar	15	Physician's facsimile number
phyEmail	Varchar	20	Physician's email address
phyBirthday	Datetime	8	Physician's birthday
phyAge	Int	4	Physician's age
phyRace	Char	20	Physician's Race
phyNation	Char	20	Physician's nationality
phyMarital	Char	10	Physician's marital status
phyGender	Char	10	Physician's gender
phyRegion	Char	20	Physician's region
phyAcademic	Varchar	200	Physician's academic status

Table 4.3: Physician_Professional

Field Name	Data Type	Size	Description
PhyProfID	Varchar	10	Physician's professional ID
phyID (primary key)	Varchar	10	Physician's ID
PhyName	Char	30	Physician's name
phyThumbnail	Varchar	20	Thumbnail of the physician
phyPicture	Varchar	20	Picture of the physician

Table 4.4: Physician_Professional_Info

Field Name	Data Type	Size	Description
PhyProfID (primary key)	Varchar	10	Physician's professional ID
PhyProf	Varchar	20	Physician's professional

Table 4.5: Physician_Timetable

Field Name	Data Type	Size	Description
phyID (primary key)	Varchar	10	Physician's ID
TDate	Datetime	8	Date
tTime1	Varchar	210	Description of physician's job at the time
tTime2	Varchar	210	Description of physician's job at the time
tTime3	Varchar	210	Description of physician's job at the time
tTime4	Varchar	210	Description of physician's job at the time
tTime5	Varchar	210	Description of physician's job at the time
tTime6	Varchar	210	Description of physician's job at the time
tTime7	Varchar	210	Description of physician's job at the time
tTime8	Varchar	210	Description of physician's job at the time
tTime9	Varchar	210	Description of physician's job at the time
tTime10	Varchar	210	Description of physician's job at the time
tTime11	Varchar	210	Description of physician's job at the time
tTime12	Varchar	210	Description of physician's job at the time
tTime13	Varchar	210	Description of physician's job at the time
tTime14	Varchar	210	Description of physician's job at the time
tTime15	Varchar	210	Description of physician's job at the time
tTime16	Varchar	210	Description of physician's job at the time
tTime17	Varchar	210	Description of physician's job at the time
tTime18	Varchar	210	Description of physician's job at the time
tTime19	Varchar	210	Description of physician's job at the time
tTime20	Varchar	210	Description of physician's job at the time
tTime21	Varchar	210	Description of physician's job at the time
tTime22	Varchar	210	Description of physician's job at the time
tTime23	Varchar	210	Description of physician's job at the time
tTime24	Varchar	210	Description of physician's job at the time

Table 4.6: Patient_Personal_Info

Field Name	Data Type	Size	Description
patID (primary key)	Varchar	10	Patient's ID
PatName	Varchar	50	Patient's name
PatIC	Varchar	20	Patient's identity card number
PatAdd	Varchar	50	Patient's home address
patPostcode	Varchar	5	Post code
PatCity	Varchar	30	City
PatState	Varchar	30	State
patCountry	Varchar	30	Country
patOfficePhone	Varchar	15	Patient's contact number (office)
PatMobile	Varchar	15	Patient's mobile phone number
patHomePhone	Varchar	15	Patient's contact number (home)
patFacsimile	Varchar	15	Patient's facsimile number
PatEmail	Varchar	50	Patient's email address
patBirthday	Datetime	8	Patient's birthday
PatAge	Varchar	3	Patient's age
PatRace	Varchar	20	Patient's Race
PatNation	Varchar	20	Patient's nationality
patMarital	Varchar	10	Patient's marital status
patGender	Varchar	10	Patient's gender
PatRegion	Varchar	20	Patient's region
PatOccup	Varchar	50	Patient's occupation
patRegDate	Datetime	8	Patient's registration date

Table 4.7: Physician_Login

Field Name	Data Type	Size	Description
phyID (primary key)	Varchar	10	Physician's login ID
phyPassword	Varchar	10	Physician's password

Table 4.8: Nurse_Login

Field Name	Data Type	Size	Description
nurID (primary key)	Varchar	10	Nurse's login ID
nurPassword	Varchar	10	Nurse's password
NurName	Varchar	50	Nurse's name

Table 4.9: Patient_Login

Field Name	Data Type	Size	Description
patID (primary key)	Varchar	10	Patient's login ID
patPassword	Varchar	10	Patient's password

Table 4.10: Patient_Treatment_Record

Field Name	Data Type	Size	Description
treCode (primary key)	Int	4	Treatment code
PatID	Varchar	10	Patient's login ID
PatName	Char	30	Patient's name
PatIC	Varchar	20	Patient's IC number
PhyID	Varchar	10	Physician's ID who provide the treatment
PhyName	Char	30	Physician's name who provide the treatment
TreDate	Datetime	8	Patient's treatment date
treDiagnosis	Varchar	50	Diagnosis of patient
treTreatment	Varchar	50	Type of treatment taken by patient
TreDiseases	Varchar	50	Disease infected by patient
treMedication	Varchar	50	Description of medication given to patient including name and quantity of drug
OprRoom	Char	5	Operation room used by patient
Ward	Char	5	Ward resided by patient
oprRoomID	Varchar	100	Operation room's ID
WardID	Varchar	50	Ward's ID

Table 4.11: Operation

Field Name	Data Type	Size	Description
oprID (primary key)	Varchar	10	Operation's ID
Roomed	Varchar	20	Room's ID
RoomName	Varchar	20	Name of the operation room
roomThumbnail	Varchar	20	Thumbnail of the operation room

Table 4.12: Operation_Room

Field Name	Data Type	Size	Description
roomID (primary key)	Varchar	20	Room ID
RoomName	Varchar	20	Name of the operation room
roomLocation	Varchar	50	Location of the operation room
roomFunction	Varchar	50	Function of the operation room
roomEquipment	Varchar	50	Equipment of the operation room
roomThumbnail	Varchar	20	Thumbnail of the operation room
roomPicture	Varchar	20	Picture of the operation room

Table 4.13: Operation_Room_Booking

Field Name	Data Type	Size	Description
ID (primary key)	Int	4	Operation room booking ID
TreCode	Int	4	Treatment code
Roomed	Varchar	20	Room's ID
PhyID	Varchar	10	Physician's ID
PatID	Varchar	10	Patient's ID
PhyName	Char	30	Physician's name
PatName	Char	30	Patient's name
RoomName	Varchar	20	Name of the operation room
bookPurpose	Varchar	100	Purpose of booking
BookDate	Datetime	8	Date of booking
BookTime	Varchar	100	Time of booking
Ecase	Varchar	20	Case of emergency

Table 4.14: Operation_Room_Timetable

Field Name	Data Type	Size	Description
roomID (primary key)	Varchar	20	Operation room ID
BookDate	Datetime	8	Date of booking the operation room
RoomName	Varchar	20	Name of operation room
bookTime1	Varchar	100	Description of operation at the time
bookTime2	Varchar	100	Description of operation at the time
bookTime3	Varchar	100	Description of operation at the time
bookTime4	Varchar	100	Description of operation at the time
bookTime5	Varchar	100	Description of operation at the time
bookTime6	Varchar	100	Description of operation at the time
bookTime7	Varchar	100	Description of operation at the time
bookTime8	Varchar	100	Description of operation at the time
bookTime9	Varchar	100	Description of operation at the time
bookTime10	Varchar	100	Description of operation at the time
bookTime11	Varchar	100	Description of operation at the time
bookTime12	Varchar	100	Description of operation at the time
bookTime13	Varchar	100	Description of operation at the time
bookTime14	Varchar	100	Description of operation at the time
bookTime15	Varchar	100	Description of operation at the time
bookTime16	Varchar	100	Description of operation at the time
bookTime17	Varchar	100	Description of operation at the time
bookTime18	Varchar	100	Description of operation at the time
bookTime19	Varchar	100	Description of operation at the time
bookTime20	Varchar	100	Description of operation at the time
bookTime21	Varchar	100	Description of operation at the time
bookTime22	Varchar	100	Description of operation at the time
bookTime23	Varchar	100	Description of operation at the time
bookTime24	Varchar	100	Description of operation at the time

Table 4.15: Operation_Type

Field Name	Data Type	Size	Description
oprID (primary key)	Varchar	10	Operation ID
OprType	Varchar	20	Type of operation

Table 4.16: Room

Field Name	Data Type	Size	Description
roomID (primary key)	Varchar	20	Room ID
RoomType	Tinyint	1	Type of the operation room
RoomName	Varchar	20	Name of the operation room

Table 4.17: Emergency_Type

Field Name	Data Type	Size	Description
eID (primary key)	Tinyint	1	Emergency ID
Ecase	Varchar	20	Case of emergency

Table 4.18: Ward_Type

Field Name	Data Type	Size	Description
WtPatName	Varchar	50	Patient's name
wtPatIC (primary key)	Varchar	20	Patient's IC number
wtPatCheckinDate	Datetime	8	Patient's check in date
wtWardType	Varchar	10	Type of the ward
wtPhyName	Varchar	50	Physician's name

Table 4.19: Ward

Field Name	Data Type	Size	Description
wWardID (primary key)	Varchar	6	Ward ID
wWardType	Varchar	10	Type of the ward
WWardBed	Int	4	Number of beds in the ward
WPatName	Varchar	50	Patient's name
WPatIC	Varchar	20	Patient's IC number
wPatCheckinDate	Datetime	8	Patient's check in date
wPhyName	Varchar	50	Physician's name
WStatus	Varchar	1	Ward status
wReservePatIC	Varchar	20	Reserved patient's IC number
wReservePatCheckinDate	Datetime	8	Reserved patient's check in date

4.4 User Interface Design (Prototype)

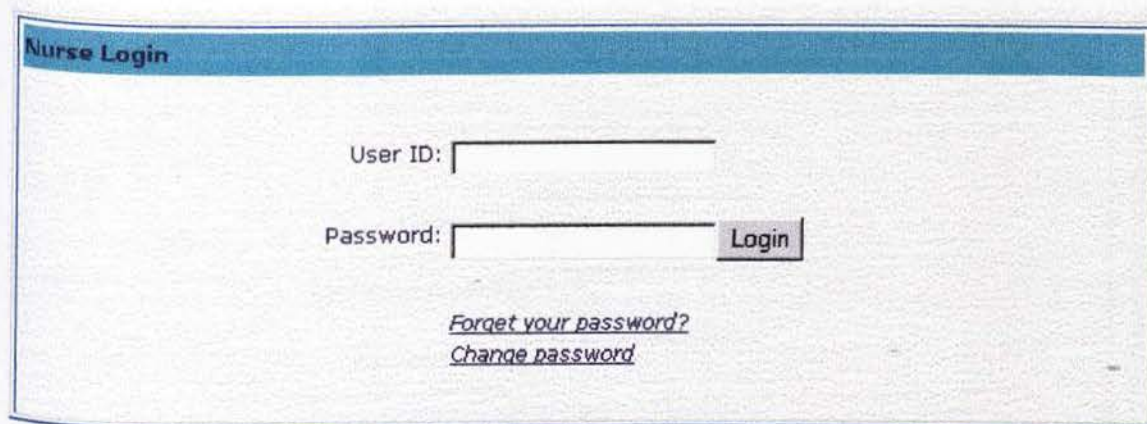
According to Academic's Computer Dictionary, user interface design means designing and interaction between end users and computer systems, including input/output methods and the conversion of data between human readable and machine-readable forms.

The goal of interface design is to provide the best way for people to interact with computers. Well-designed screen display and input forms should meet the objectives for effectiveness, efficiency, productivity, accuracy, consistency, easy to use as well as attractiveness. With a user-friendly system, it will improve the efficiency and the effectiveness when using the system.

The following figures show the user interface of **ORRASH**.

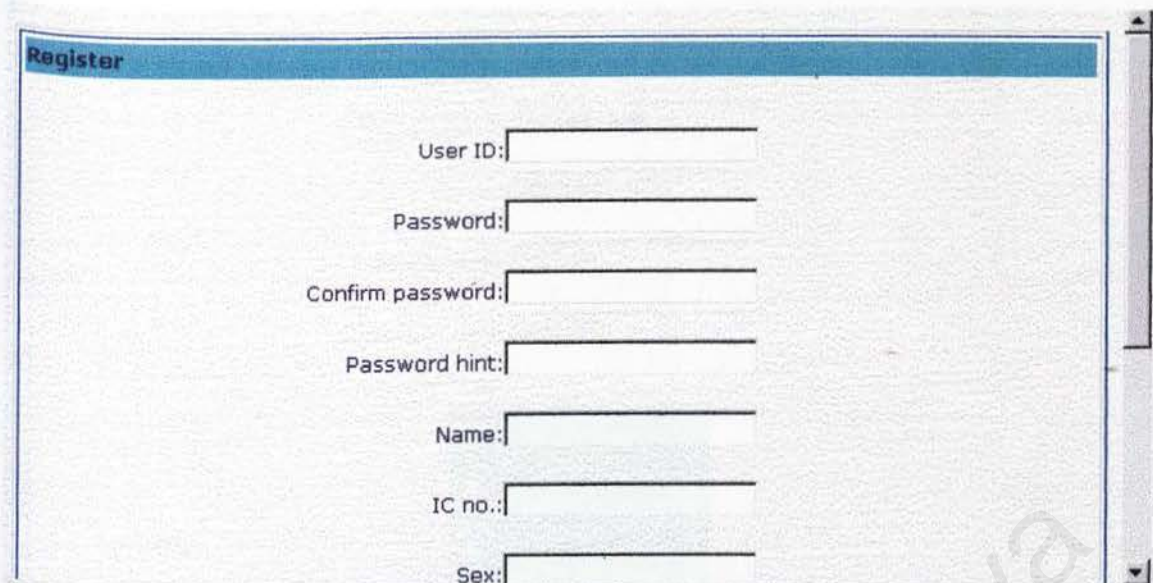


Figure 4.10: Main page for ORRASH



The image shows a web interface for Nurse Login. It has a blue header bar with the text "Nurse Login". Below the header, there are two input fields: "User ID:" and "Password:". To the right of the "Password:" field is a "Login" button. Below the input fields, there are two links: "[Forget your password?](#)" and "[Change password](#)".

Figure 4.11: Interface for Nurse Login



A screenshot of a web browser displaying a registration form titled "Register" in a blue header bar. The form contains seven input fields arranged vertically, each preceded by a label: "User ID:", "Password:", "Confirm password:", "Password hint:", "Name:", "IC no.:", and "Sex:". The form is set against a light gray background with a vertical scrollbar on the right side.

Register

User ID:

Password:

Confirm password:

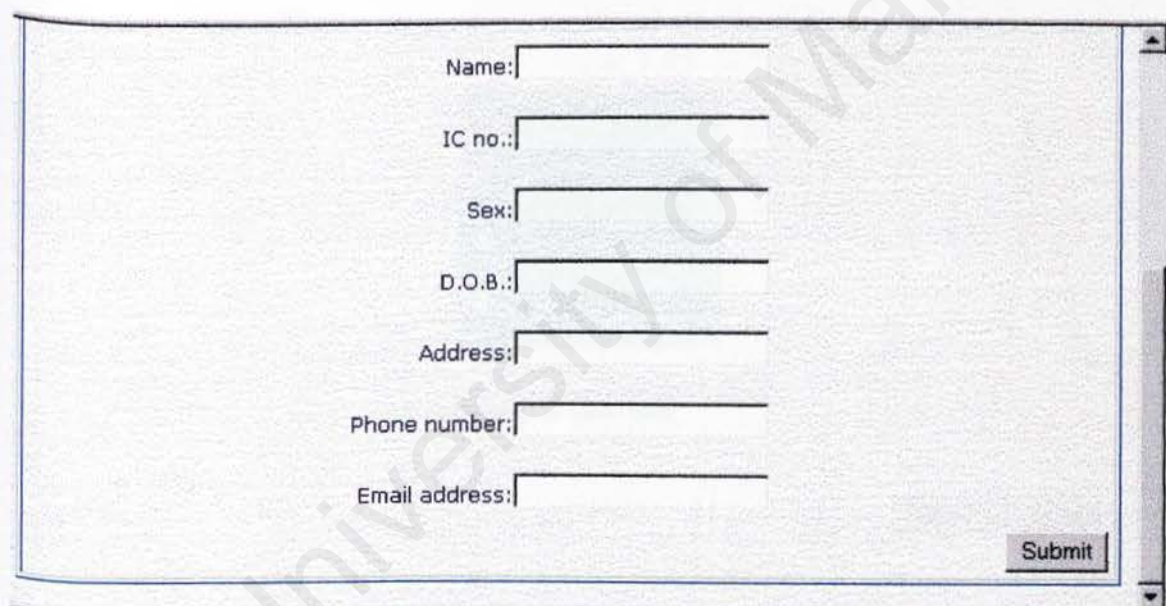
Password hint:

Name:

IC no.:

Sex:

Figure 4.12: Interface for Register Form – Part 1



A screenshot of a web browser displaying the second part of a registration form. It contains six input fields arranged vertically, each preceded by a label: "Name:", "IC no.:", "Sex:", "D.O.B.:", "Address:", and "Phone number:". Below these fields is an "Email address:" label followed by an input field. A "Submit" button is located in the bottom right corner of the form area. The form has a light gray background and a vertical scrollbar on the right.

Name:

IC no.:

Sex:

D.O.B.:

Address:

Phone number:

Email address:

Submit

Figure 4.13: Interface for Register Form – Part 2

Decide Ward Type Form

Patient name: Ng Kee Huat

IC no.: 780909-01-1234

Check in date: 1/9/2001

Checkout date: 3/9/2001

☒ common ward

☐ special ward

☐ labor ward

Figure 4.14: Interface for Decide Ward Type Form – Part 1

☒ common ward

☐ special ward

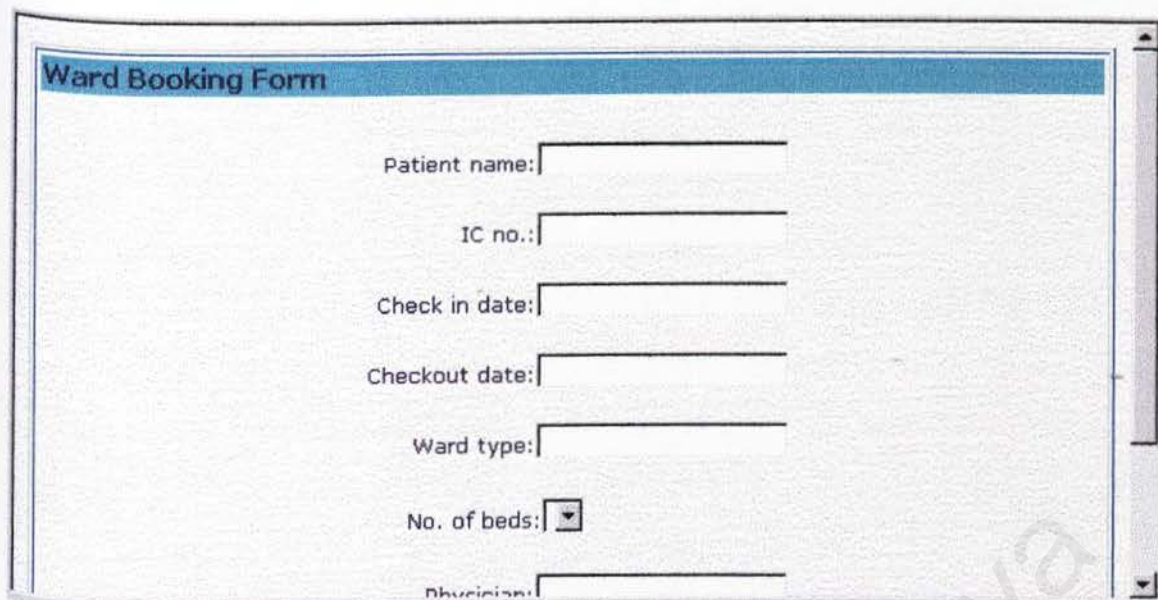
☐ labor ward

☐ children ward

Physician: Tan Guan Hooi

Submit Cancel

Figure 4.15: Interface for Decide Ward Type Form – Part 2



Ward Booking Form

Patient name:

IC no.:

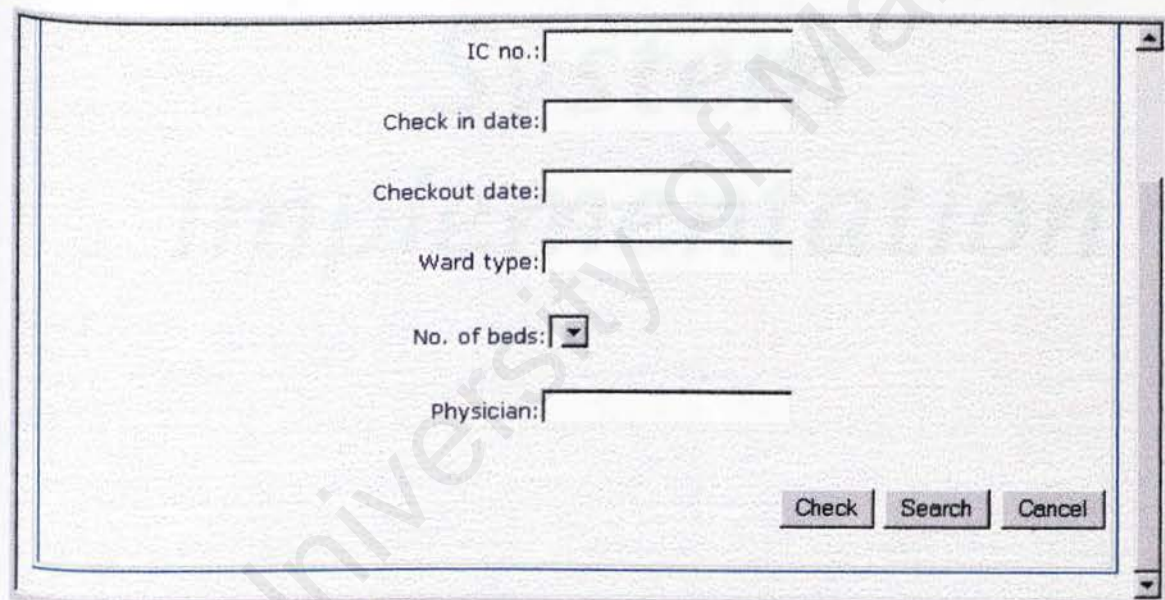
Check in date:

Checkout date:

Ward type:

No. of beds:

Physician:

Figure 4.16: Interface for Ward Booking Form – Part 1

IC no.:

Check in date:

Checkout date:

Ward type:

No. of beds:

Physician:

Figure 4.17: Interface for Ward Booking Form – Part 2

Chapter 5: System Implementation

Chapter 5: System Implementation

5.1 Introduction

System implementation in a nutshell is the construction of the application and the delivery of the application in to the 'production phase'. System implementation includes building and testing its contained modules and sub-modules, involving system requirements and design conversion into program codes. Coding the program is actually the phase where the software representation produced in the design phase is translated into computer readable form.

5.2 Coding Styles

To ensure the quality and proper structure in the code generated for ORRASH, there are several principles to be followed.

Readability

This is an important factor especially when it comes to future enhancing the system by another developer. Codes should be easily understood.

To cater for this, meaningful variables and labels names have been used; comments are written in each module explaining their functionality and proper indentation are followed to enhance readability.

Figure 5.1 is a part of source code takes from the ORRASH system showing how the readability is achieved.

```

If varTempPatID <> "" Then
    rsLogin.Filter = "patID = '" & varTempPatID & "'"
    If Not rsLogin.EOF Then
        rsLogin("patID") = Request.Form("PersonID")
        rsLogin("patPassword") = Request.Form("password")
        rsLogin.Update
    End If
Else
    rsLogin("patID") = Request.Form("PersonID")
    rsLogin("patPassword") = Request.Form("password")
    rsLogin.Update
End If

```

Figure 5.1: Part of Source Code

Maintainability

Codes should be easily read, corrected and revised. To achieve this, codes should be readable (as explained above), highly cohesive and loosely coupled. Codes that perform functions for one module should be grouped together and try as much as possible to achieve high cohesion and loose coupling.

Robustness

Codes should be robust in terms of handling errors and responding by displaying appropriate error messages and try to avoid system failure.

For example, if an ORRASH user try to insert invalid values into the system database through the provided form. The system will block the insertion and displays appropriate error messages. The insertion is only allowed when the values is valid.

Figure 5.2 is a part of source code takes from the ORRASH system showing how the robustness is achieved.

```

var PatientName=document.frmUser.PatientName.value
if (PatientName==""){
    alert ("The Patient Name field is empty. Please fill in");
    document.frmUser.PatientName.focus();
    return false;
}

var ICNumber=document.frmUser.ICNumber.value
if (ICNumber==""){
    alert ("The IC Number field is empty. Please fill in");
    document.frmUser.ICNumber.focus();
    return false;
}

```

Figure 5.2: Part of Source Code

5.3 Coding Approach

The coding approach of ORRASH uses 2 approaches and they are the top-down approach and the bottom-up approach. These approaches are not only used on individual basis but also as a combination to obtain the benefits from the both techniques.

Top-Down Approach

This approach starts by looking at the large picture of the system and then exploding to smaller parts or subsystem. Top-down approach allows the higher-level modules to be coded first before the lower level modules.

This method ensures that the important or core modules of the system be developed and tested first. Deploying the method gives a preliminary version of the system sooner. The following outlines some of the advantages using this approach:

- Avoiding the chaos of attempting to code a system all at once.
- Prevents the developer from getting so mired in the detail that they loose track of what the system is suppose to do.
- This method is compatible with the general system thinking of humans.

Bottom-Up Approach

In contrast with the top-down approach, the bottom-up approach starts coding at the lower level modules before the higher-level modules. The higher-level module act as an empty shell that calls these lower level modules. The completed lower level module will then be integrated with the newly completed higher-level module.

5.4 Database Development

In the system development, the system database is developed base on the logical data model for ORRASH created during the system design phase.

The database development is started by creating an empty database called *Hospital* using Microsoft SQL Server 7.0. All the tables needed are then created by specify all the fields for each table and the field properties. A primary key is allocated for each table in the database. After all tables being created, relationships between the tables is established using SQL 7 provided feature to enforce referential integrity. The referential integrity is an important constraint on a relationship that ensures consistency between related tables.

Table 5.1 is one of the tables of the *Hospital* database in design view.

SQL Server Enterprise Manager - [2:Design Table 'Patient_Personal_Info']

Console Window Help

Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value
patID	varchar	10	0	0	<input type="checkbox"/>	
patName	varchar	50	0	0	<input checked="" type="checkbox"/>	
patIC	varchar	20	0	0	<input checked="" type="checkbox"/>	
patAdd	varchar	50	0	0	<input checked="" type="checkbox"/>	
patPostCode	varchar	5	0	0	<input checked="" type="checkbox"/>	
patCity	varchar	30	0	0	<input checked="" type="checkbox"/>	
patState	varchar	30	0	0	<input checked="" type="checkbox"/>	
patCountry	varchar	30	0	0	<input checked="" type="checkbox"/>	
patOfficePhone	varchar	15	0	0	<input checked="" type="checkbox"/>	
patMobile	varchar	15	0	0	<input checked="" type="checkbox"/>	
patHomePhone	varchar	15	0	0	<input checked="" type="checkbox"/>	
patFacsimile	varchar	15	0	0	<input checked="" type="checkbox"/>	
patEmail	varchar	50	0	0	<input checked="" type="checkbox"/>	
patBirthday	datetime	8	0	0	<input checked="" type="checkbox"/>	
patAge	varchar	3	0	0	<input checked="" type="checkbox"/>	
patRace	varchar	20	0	0	<input checked="" type="checkbox"/>	
patNation	varchar	20	0	0	<input checked="" type="checkbox"/>	
patMarital	varchar	10	0	0	<input checked="" type="checkbox"/>	
patGender	varchar	10	0	0	<input checked="" type="checkbox"/>	
patRegion	varchar	20	0	0	<input checked="" type="checkbox"/>	
patOccup	varchar	50	0	0	<input checked="" type="checkbox"/>	
patRegDate	datetime	8	0	0	<input checked="" type="checkbox"/>	

Table 5.1: Table of *Hospital* Database in Design View

5.5 Software Tools for Design and Documentation

'Design' in this section explicitly refers to the processes of structure chart, data flow diagrams and entity-relationship diagram drawing in the initial development phase of ORRASH. The tools used in the development of ORRASH are Microsoft Word 2000 and Microsoft PowerPoint 2000.

5.6 Software Tools for Testing

Testing in the development of ORRASH uses the debugger provided in the Microsoft Visual InterDev 6.0. This debugger provides program stepping, breakpoints and watch windows functions. This debugger is very efficient in determining the cause of errors in the program and it also allows for debugging within the web browser environment.

Chapter 6:

System Testing

Chapter 6: System Testing

6.1 Introduction

System testing is a verification and validation process. A successful testing will uncover errors in the software and demonstrates that system functions appear to be working according to specification. System testing is a critical phase of its quality control and assurance. Several rules serve well as program testing objectives.

- Testing is a process of program execution with explicit intents to find errors and run-time program bugs.
- An effective test case is one which contain unexpected testing record sets with high probability of detecting undiscovered errors during the program design and development phases.
- A successful test is also not one which uncovers only few expected errors, it is one which constantly provides new challenges to its programmers over time.

6.2 Testing Techniques

Two main testing techniques are used namely the White Box Testing and the Black Box Testing.

White-Box Testing

Also known as glass-box testing, this testing technique uses the control structure of the procedural design to derive test cases. This method ensure that

- All independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false side.
- Execute all loops at their boundaries and within their operational bounds.
- Exercising internal data structures to ensure their validity.

The white-box testing generally ensures that all detail and often-left unnoticed errors are taken care off. These errors though may not even occur during processing, may still pop up on a regular basis. Furthermore another frequent problem in coding is the typographical errors. These happen randomly and at times are difficult to detect. Using this approach (white box testing) will far more likely to uncover these bugs.

Black-Box Testing

This method focuses on the functional requirements of the software. It ensures that a given set of input will fully exercise all functional requirements of the software. This technique is used to demonstrate that software functions are operational, that input is properly accepted, and the output is correctly produced. It complements white-box testing and is likely to uncover a different class of errors.

Black box testing is also called functional testing. Test cases used in black box testing are derived from the specifications of the software. When doing black box testing, functionality is taken into consideration rather than the implementation, whereby the program internals are not given serious attention. The program is viewed as mapping of points from an input space to an output space.

The main objective of black box testing is to uncover error in

- Incorrect or missing functions
- Interface errors
- Errors in data structures of external database access
- Performance errors
- Initialization and terminating errors

Unlike white-box testing, which is performed early in the testing process, black-box testing tends to be applied during later stages of testing.

6.3 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design that is system module. During this step, all important control paths are tested to uncover errors within the boundary of the module by using the component-level design description as a guide.

The goal of unit testing is to find faults in components. First, I examine my code by reading through it, trying to spot algorithm, data, and syntax faults. I even compare the code with the specifications and with my design to make sure that I have considered all relevant cases. Next, I compile the code and eliminate remaining syntax faults. Finally, I develop test cases to show that the input is properly converted to the desired output. Unit testing follows exactly these steps. [24]

6.4 Integration Testing

When we are satisfied that individual components are working correctly and meet our objectives, we combine them into a working system. This integration is planned and coordinated so that when a failure occurs, we have some idea of what caused it. In addition, the order in which components are tested affects our choice of test cases and tools. [24]

The integration testing was conducted incrementally where the system is constructed and tested in small increments to isolate and correct the errors easier and the interfaces are more likely to be tested completely.

There are a number of different incremental integration strategies available including top-down integration, bottom-up integration, regression testing and smoke

testing. Based on the system characteristics and project schedule, a combined approach that uses top-down tests for upper levels of the program structure, coupled with bottom-up tests for subordinate levels was selected as ORRASH integration testing.

Top-down integration beginning with the main control module as a test driver and stubs are substituted for all components where modules are integrated moving downward through the control hierarchy. Tests are conducted as each component is integrated. Top-down integration enables the detection of design error early in the testing phase and avoiding extensive redesign or re-implementation.

Bottom-up integration begins construction and testing with atomic modules where low-level components are combined into cluster to perform a specific system sub function and tested. Bottom-down integration is an easier test case design because processing required for component subordinate to given level is always available and thus the need for stubs is eliminated.

6.5 System Testing

System testing is designed to reveal bugs not possibly attributed to individual components or to interactions between components and modules. System test is carried out on the entire integrated system as one unit. Its activities include testing of system performance, stress, security, configuration sensitivity, usability, data integrity, error handling and recovery. System testing verifies that ORRASH is functioning properly and all design and development objectives are met. Several steps to test ORRASH include function testing, performance testing and acceptance testing.

Function Testing

System testing begins with function testing. This focuses on system functionalities. Each function can be associated with system components that accomplish it. Some functions

6.6 System Debugging

Debugging is the process of finding and correcting bugs. Debugging and testing are often treated as being synonymous, however, debugging may involve inspection and other manual processes as well as testing, and debugging is generally considered to precede the formal portion of the testing process.

The debugging tool I use to debug ORRASH source codes is the debugger provided by the Microsoft Visual InterDev 6.0. This debugger let us to step through the source codes, add breakpoint at places suspected faulty areas. When debugging the system, the local window and immediate window are used to check the value of variables in the source codes.

Chapter 7: System Evaluation

7.1 Problems Encountered and Solutions

In any system development, problems will be encountered throughout. During the research and development ORRASH, several setbacks have been encountered to challenge the ongoing activities. These problems, along with solution approaches are highlighted in the following sections.

Lack of Knowledge on Web-Based Programming

This is the major problem as the concept of web-based programming is very much different from the normal stand-alone programming. I only learn simple HTML in designing web pages from one of my course. But the HTML is only can support static web pages, not for dynamic web pages that I need to develop the ORRASH. Therefore the lack of exposure to web-based programming has increased my learning curve.

However, all the problems had been solved finally by reading a lot of references, exploring the Internet and discuss with my friends, which also use the same programming language to develop their system.

Lack of Latest References

This is the problem faced when finding references for the literature review and finding programming reference books. This problem had solved by visiting to University of Malaya main library, exploring the Internet and by buying new reference books after comparing the price on some bookstores.

Lack of Knowledge on Ward Reservation

Because of lack of the knowledge on the actual process of ward reservation and scheduling in a hospital, I find difficult to design the ORRASH system. At the beginning, I try to use my common sense to design the system but finally find it cannot work.

This problem is solved by taking an interview with the authorities of University of Malaya Medical Center and asking my relations and friends who have stay in a ward.

Research Challenges

When time came to decide on the system development tools for ORRASH, several ambiguities are met due to the wide range of available Software Development Kits (SDKs) in the market. Although Microsoft Visual InterDev and Microsoft SQL Server seems to be the de facto software choice of developing web-based system, many other tools such as Java SDK, Microsoft Visual Basic, Oracle8i offer powerful features as well. Doubts on choosing the right tools are gradually overcome after taking into account the functionalities of ORRASH, research on relevant software related materials, and most importantly advice from previous system developers and the project supervisor.

7.2 System Strengths

Suitable and User Friendly Interface

The advantage of ORRASH is it provides the suitable interface for the hospital user. Hospital is a serious place, not a place for entertainment; colorful interface is not suitable for the circumstance. So, I use the blue color as the system interface main color because blue color giving the expression of soft, peace and comfortable. Besides that, this system is easy to use. The user only has to key in a few data in the text box and most of the

operation can be performed by point and click, hence the user can navigate from page to page without much effort.

Easy For Searching

The system support easily searching, for example, a nurse just has to key in the patient name and then she can retrieve the patient personal information. I know the patient name is not unique because there has possibility be more than one person has the same name, but if the system requires the nurse to key in the patient IC number, it is more difficult to key in and more easily to get wrong and bring the inconvenient to the nurse. Furthermore, if the nurse key in the name which belong to two person, the system will require her to key in the IC number. This condition will not happen often, because there would not have so many people have the same name.

Searching Availability

The system can search available ward for the patient easily depend on the ward type (Common, Children, Labor, Surgery, Medical) given by the physician and the number of beds (1, 2, 4, 6) in a ward he/she prefer. First, the system will search for the ward that is empty which means no patient stay in and not reserved by any patient. If there are no more empty wards, the system will search for the ward which is reserved by other patient but still no patient stay in. This is to avoid too many transferring doing by the nurse, because if a patient stay in a ward which is reserved by other patient, when the reserved patient come, that patient has to be transferred to another ward.

Validation On Input Data

ORRASH performs strenuous validation on user input. This further enhances the system reliability and handling. Certain characters that are not accepted in certain cases are blocked to avoid errors from happening. Therefore users will not be bogged down by sudden errors happening at entry. For example, the check in date for a ward given by a

physician to a patient cannot be smaller than the current date. If the check in date is smaller, the system will not accept it into the database until the date is changed to the valid date.

Highly Integrated Modules

All modules and sub-modules in ORRASH are highly integrated, where data change and updates in any one module can be detected and copied to other linked modules. This reduces data entry and management time.

Secure Data Integrity

Data integrity is an important issue in all database related systems, and guarantees highly secure data integrity. Any records in any one module with existing related records in other modules cannot be deleted. Duplicate records and invalid data types are also handled effectively in ORRASH.

System Security

Login ID and password are required to access the certain modules of the system and the authorized person is allowed to change the password and given permission to access to certain modules. For example, a patient can access he/she own modules but cannot access the physician modules and the nurse modules. But, for the physician and nurse, they are giving permission to access the patient modules like insert patient records, view the patient records and update the patient records.

7.3 System Limitations

As in other systems, there are also several setbacks and limitations in ORRASH. These limitations can be addressed in future development and system enhancements.

Output Printing Function

ORRASH has not provided users the ability to print query results or any other data management output directly from its applications. If a patient reserve a ward, he/she would not received a print out as a reference for his/her check in date and as the verification of the reservation when he/she come to check in next time.

Support for multi-user environment

If more than one person access for the same point at the same time, what will happen? I still cannot find what will happen to the system if face with this situation because I do not know the way to test for this situation. Fortunately, this situation can be said would not happen because this is not a hot web site and the possible users involve in accessing the same point at the same time is just the hospital physicians and the counter nurse (nurse whose responsible is to manage the ward check in and reservation). The number is very small, and the possibility can be said equal to zero.

Resources Using

Because of do not know the checkout date for a patient before the patient checkout from a ward, once a ward is checked in or reserved by a patient, other patients cannot reserve that ward anymore. This is waste of resources.

7.4 Future Enhancements

System limitations should be addressed to enhance the functionality and features of ORRASH. The current version of ORRASH can be updated with some enhancement features discussed in the following section.

Printing Module

Provide the printing function for the system; the system can print out records and documents, which is need by the users. For example, verification for reservation, patient medical records and so on.

Allow Concurrently Access

Although the possibility of access the same point at the same time by more than one user is very low. But the possibility is still there, so, to prevent the system meet unexpected errors. An approach should be found to solve this problem.

Optimize Resources Using

A way must be found to allow a patient to reserve a ward that is checked in or reserved by other patient. This is to optimize the using of hospital resources and let more patients to gain advantages from these resources provided by the hospital.

7.5 Conclusion

ORRASH has been successfully achieved and fulfilled its objectives as an easy to use hospital room reservation and scheduling system. It helps the hospital employees to reduce their jobs and make their jobs easy. The process of reserve and schedule the hospital rooms become smooth and effective.

A lot of valuable knowledge has been gained throughout the development of ORRASH including knowledge of the reservation and scheduling of hospital rooms, programming knowledge, database designing and so on.

The project has been a good practical testing on undergraduates' capabilities in handling and developing a project. It provides the opportunity for them to apply all gained knowledge to a real world environment.

References

REFERENCES

- 1) <http://cfhub.com/aboutcf/>
- 2) http://www.oreilly.com/openbook/cgi/ch01_01.html
- 3) <http://msdn.microsoft.com/library/default.asp?URL=/library/en-us/dnasp/html/ASPOver.asp>
- 4) <http://java.sun.com/products/jsp/>
- 5) <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/>
- 6) <http://msdn.microsoft.com/scripting/default.htm?scripting/vbscript>
- 7) <http://developer.netscape.com/docs/manuals/communicator/jsguide4/index.htm>
- 8) <http://www.intranetjournal.com/faqs/jsfaq/gen1.html>
- 9) <http://ag.arizona.edu/ecat/html-basics/html-is.html>
- 10) <http://www.personal.kent.edu/~jnattey/natteypres.htm>
- 11) <http://www.wsbtch.com/mssql.htm#sql7>
- 12) <http://www.oracle.com/ip/deploy/database/8i/index.html?capabilities.html>
- 13) <http://msdn.microsoft.com/vinterdev/prodinfo/evaluation/default.asp>
- 14) <http://www.hotels-moscow.ru/>
- 15) <http://www.microsoft.com/iishelp/iis/htm/asp/iivarvelc.htm/>
- 16) http://www.advantageware.com/VBAD_Overview.htm#What%20is%20VB%20Advantage
- 17) http://www.carib-link.net/ms_software/products/winnt4server.html
- 18) <http://www.microsoft.com/catalog/display.asp?subid=22&site=656>
- 19) P. Sellapan, "Access 2000 Through Examples", Times; 1999
- 20) Alan M. Davis, "Software Requirements: Objects, Functions, and States", Prentice Hall; 1993
- 21) Kenneth E. Kendall, "Systems Analysis And Design", Prentice Hall; 1999
- 22) Shari Lawrence Pfleeger, "Software Engineering Theory And Practice", Prentice Hall; 1998
- 23) P. Sellapan, "System Analysis And Design", Sejana Publishing; 2000
- 24) David M. Kroenke, "Database Processing Fundamentals, Design and Implementation Seventh Edition", Prentice Hall; 2000

APPENDIX

Appendix

APPENDIX

In the appendix, I will attach the interview questions prepared by my project partner, Mr. Lau Chong Ee and I for the University Malaya Medical Centre authorities to get the information about the process of reserve a room (ward and operation room) in a Hospital.

Besides that, the reply from the University Malaya Medical Centre authorities depends on our interview questions will also be attached.

At the end of the appendix, I will attach the sample of my program codes on reserve a ward in a hospital.



PUSAT PERUBATAN UNIVERSITI MALAYA

University Malaya Medical Centre
(sebelum ini dikenali sebagai Hospital Universiti)

ALAMAT: LEMBAH PANTAI, 59100 KUALA LUMPUR, MALAYSIA
TELEFON: 03-79564422, 03-79574422
FAX NO: 603-79562253 E-mail: ummc@uhkl.edu.my

BIL. KAMI: HEK-27/1 Jld.8
BIL. TUAN:

24 Disember 2001

Tel.: 7953-5605

Encik Lau Chong Ee
385, Jalan 17/19
46400 Petaling Jaya
Selangor Darul Ehsan

Saudara,

Permohonan Menjalankan Kajian

Dengan hormatnya saya merujuk kepada surat saudara bertarikh 7 Ogos 2001 mengenai perkara di atas.

2. Bersama-sama ini dikembalikan borang-borang yang telah dilengkapkan. Saya berharap pihak saya telah memberikan respon yang bermanfaat untuk kertas penyelidikan saudara.

Semoga saudara mendapat kejayaan yang cemerlang kelak.

Sekian, terima kasih.

"BERKHIDMAT UNTUK NEGARA"

Saya yang menurut perintah

b/p 

Nurri bte Kadarman

Pengurus Kanan

PTJ Hal Ehwal Korporat

b.p. Pengarah

Pusat Perubatan Universiti Malaya

MS/maj

Questionnaire:

Prepare by: Mr. Lau Chong Ee
Software Engineering Department
Faculty of Computer Science & Information Technology
University of Malaya

Operation Room Booking System

1. How is the current *operation room booking system* run in your hospital?
- a. Manually
 - b. Computerized
 - c. Web-based
 - d. Other. Please specify _____
- Answer: (A)

2. Please describe the procedures of the *operation room booking* in your hospital in detail.

E.g. Fill in the form → submit form to office → ...compile

The compiled elective operation booking is then sent to all Surgical
Departmental heads, wards, Pathology, Medical Transfusion Unit.

3. What are the types of operation room that can found in your hospital?

Types of operations : - Orthopaedic, Gynaecology, General Surgery,
Urology, Eye, ENT, Cardiac, Neurology, Plastic, Endoscopy, Minor O.T.

4. Did doctors set the duration for the booked room in order to conduct an operation?

(E.g. booking an operation room for 3 hours).

- a. Yes
- b. No

Answer: (b)

5. If the duration of an operation is over the expected time, what are the solutions needed to arrange the coming up operation where this operation will be held at the same operation room?

The coming up operation will have to follow the case as scheduled.

6. Is an operation room can be used sequentially without needing an interval time for cleaning the operation room?
 - a. Yes
 - b. No
 Answer: (a)
 If "No", how long the time needed to clean the operation room?

7. What kind of data that the doctor needs to fill up during the operation room booking process?
 Name, Registration Number, Sex, Age, Ward, Diagnosis, Operation to be done and any other request.

8. According to your opinion, what is the advantages and limitation of the current operation room booking system running on your hospital?
 Advantage : Complete information can be obtained regarding operation.
 Limitation : Overbooking of cases.

Appointment Booking System

1. How is the current *appointment booking system* run in your hospital?
 - a. Manually
 - b. Computerized
 - c. Web-based
 - d. Other. Please specify _____
 Answer: ()
2. Please describe the procedures of the *appointment booking* in your hospital in detail.
 E.g. Patient arrival → register → waiting

3. What are the data that the patients need to fill up during they booking an appointment?

6. Is an operation room can be used sequentially without needing an interval time for cleaning the operation room?

a. Yes

b. No

Answer: ()

If "No", how long the time needed to clean the operation room?

7. What kind of data that the doctor needs to fill up during the operation room booking process?

8. According to your opinion, what is the advantages and limitation of the current operation room booking system running on your hospital?

Appointment Booking System

1. How is the current *appointment booking system* run in your hospital?

a. Manually

b. Computerized

c. Web-based

d. Other. Please specify manually if scheduled allocation is fully booked and

Answer: (b.)

if patients need to be seen early.

2. Please describe the procedures of the *appointment booking* in your hospital in detail.

E.g. Patient arrival → register → waiting

New to PPUM → triage at RUKA (Rawatan Utama) → appointment for specific ^{discipline}

Seen by Doctor → given appointment at specific clinic

3. What are the data that the patients need to fill up during they booking an appointment?

For Antenatal and Specialist clinic (no need urgent treatment), a form

(please see as attached) (lampiran a & b)

will be given to patient with instructions for guarantee letter, if appropriate
 On day of appointment, the said form to be produced for registration.
 Data included are patients particulars, address, demographic data and
 next of kin address and occupation -

4. Is there any solution that can solve the problem of dummy booking?
 (E.g. patient who booked an appointment but not coming at that time.)
 a. Yes
 b. No
 Answer: (a)
 If "Yes", what are the solutions?

Patient can call up the relevant clinic to postpone and make new appointment.

5. According to your opinion, what are the advantages and limitations of the current appointment booking system running on your hospital?

Computer system is 'slow'
Two screens before the actual required screen and have to get out of system to check the data.

Patient Information System

1. How is the current patients' records management system run in your hospital?
 a. Manually
 b. Computerized
 c. Web-based
 d. Other. Please specify _____
 Answer: (a)

2. Are the following information valid or enough for a registration process? Please tick ✓ at the related columns that specifies the needs for a registration process.
 (✓) Patient identification data (e.g.: name, date of birth, sex, race and etc.)
 (✓) Patient demographic data (e.g.: address, job, marital status, phone No, and etc.)
 () Patient vital data (e.g.: height, weight, blood pressure, temperature and etc.)
 () Patient allergies.
 () Patient special disease
 () Patient handicap.

Please specify the data that are relevant if the above data are incorrect and not enough.
The rest of the data above are done by nurses or doctor in the clinic not at registration -

3. What are the necessary data need to be recorded after a doctor diagnoses the patient? Please list down.

Patient's conditions and findings from examination:

Answer can be more than one for the following question 4 and 5.

4. Who is allowed to update the patient's identification data or demographic data?

() Patient

(☒) Nurse

(☒) Doctor

() Others. Please specify clerk

5. Who is allowed to add, update and deleting the following medical record for a patient? Please fill in (a), (b) or (c) for the relevant question on the line after the question.

(a). Doctor (b). Nurse (c). Others, please specify

• Medications (e.g. medical data that given to patient)

a

• Allergies

a

• Vital signs (e.g. blood pressure, pulse, temperature, Height, weigh)

b

• Lab record (e.g. test urine analysis)

b

• Patient's problem history

a

"Thank you for kindly answer the above questions."

** Can I have a set of form that used for registration, record patient's record, booking operation room and appointment?

Ward Booking System

1. What type of ward do you have in the hospital? *classroom*
Single, double, 4 bedded and 6 bedded rooms
2. Can you give me the list of ward type of the hospital?
as above
3. What is the process of booking a ward in a hospital?
admission by rotation through admission counter
4. Who determine the type of ward a patient can book?
admitting doctors
5. Who determine the check in and checkout date for a ward?
ward doctors
6. If patient books a ward, do you know the date the patient checkout?
No
7. After a patient checkout from a ward, when is the new patient can reside in the ward?
immediately
8. If you know the patient checkout date for a ward, is that mean the ward can be booked after the checkout date?
No booking
9. Do the hospital limit the range of time that a ward can be booked?
For example, some hospital does not allow a patient to book a ward for 1 year later.)
not relevant.
10. If a ward is booked for 5 days later, can the ward still be booked before that day?
If the patient needs to be admitted earlier, doctor concerned will arrange.
11. If a patient has to stay in the ward exceed of the checkout date, and other patient has booked the ward after the checkout date, how do you solve this problem?
we transfer or discharge other patients to make way for new patient.
12. Can you give me a form for a person to register as a patient for the hospital?
as attached. (a + b)
13. Can you give me a form for booking a ward if you have this kind of form?
'elective admission form' as attached (c)
14. Do you keep the information of patient who has resided in a ward?
Yes, all patients data will be kept by Medical Record Department

RP 600.1

nda dikehendaki hadir ke Kaunter 1, Klinik Ante-Natal pada pagi.

DILENGKAPKAN OLEH JURURAWAT

Klinik: _____ Kod Klinik: _____ Tarikh: _____
 Nama Jururawat: _____ Tandatangan: _____

DILENGKAPKAN OLEH PEMBANTU TADBIR

No. Rujukan: _____ Kod Klinik: _____
 Tarikh Penetapan Temujanji: _____ Masa: _____
 Nama Pembantu Tadbir: _____ Tandatangan: _____

DILENGKAPKAN OLEH PESAKIT

No. Pendaftaran _____
 (Jika ada)

Nama Penuh (mengikut K.P.): _____

Nama Lain (jika ada): _____

No. K.P.: _____ Jantina: _____

Tarikh Lahir: _____ Umur: _____

Negeri Lahir: _____

Kumpulan Etnik: ☐ Melayu ☐ India
☐ Cina ☐ Orang Asli
☐ Lain-lain

Taraf Perkahwinan:
☐ Bujang ☐ Janda
☐ Kahwin ☐ Lain-lain

Warganegara: _____

Ugama: _____

Pekerjaan: _____

Sekiranya pelajar, pekerjaan bapa: _____

Sekiranya berkahwin atau surirumah, pekerjaan suami: _____

ALAMAT

Rumah: _____

No. Tel: _____

Pejabat: _____

No. Tel. _____ Samb. _____ Fax: _____

WARIS - PERTALIAN:

Nama: _____

ALAMAT

Rumah: _____

No. Tel.: _____

Pejabat: _____

No. Tel: _____ Samb. _____



PUSAT PERUBATAN UNIVERSITI MALAYA

ALAMAT: LEMBAH PANTAI, 59100 KUALA LUMPUR, MALAYSIA
TELEFON: 03-79564422, 79574422, KEBEL: UNIHOS KUALA LUMPUR
FAX NO: 6-03-79562253

BIL. KAMI:

RP 674(Pind.4)

BIL. TUAN:

Tarikh:

Tuan/Puan,

KEMASUKAN KE PUSAT PERUBATAN UNIVERSITI MALAYA

Adalah dimaklumkan bahawa persediaan telah dibuat untuk anda/anak anda masuk ke wad pada hari jam pagi/petang.

2. Sila bawa bersama anda:

- ☐ Wang sebanyak RM bagi bayaran pendahuluan pada/atau sebelum tarikh kemasukan wad.
- ☐ Surat jaminan dari Ketua Jabatan.
- ☐ Surat jaminan dari majikan dimana pihak majikan anda ada perjanjian dengan hospital melalui surat jaminan bank.
- ☐ Kad pencen dan wang pendahuluan sebanyak RM30.00.
- ☐ Kad Kedatangan Hospital dan Kad Pengenalan.
- ☐ Kad-kad kredit/caj yang diterima oleh Hospital sekiranya ada.

- 3. Kegagalan membayar wang pendahuluan akan menyebabkan tarikh kemasukan wad anda dibatal atau ditunda.
- 4. Tuan/puan akan diminta membayar wang pendahuluan tambahan sekiranya tempoh tinggal di wad agak lama dan bayaran pendahuluan pertama tidak mencukupi.
- 5. Anda dikehendaki melaporkan diri pada tarikh tersebut di atas di Kaunter Kemasukan yang terletak di Tingkat Bawah, atau di Kaunter Pendaftaran Klinik Ante-Natal (pesakit bersalin sahaja).

Yang benar,

Sila lihat lampiran sebelah bagi lain-lain maklumat

#Searchward.asp#

```
<%Response.Buffer = True%>
<BASEFONT COLOR="DarkBlue" face=serif>
<HTML>
<HEAD>
```

```
<SCRIPT language="JavaScript">
```

```
!--
function VerifyData()
{
    var PatientName=document.frmUser.PatientName.value
    if (PatientName==""){
        alert ("The Patient Name field is empty. Please fill in");
        document.frmUser.PatientName.focus();
        return false;
    }
}
-->
```

```
</SCRIPT>
```

```
<TITLE>Search Ward</TITLE>
```

```
</HEAD>
```

```
<BODY bgcolor=LightCyan>
```

```
<%If Session("nurID") = "" Then
    Response.Redirect "NurseLogin.asp?NotFound=True"
End If%>
```

```
<!--#include file="NurseResponsibility.asp"-->
```

```
<TABLE BORDER=0><TR><TD bgcolor=LightBlue WIDTH=10%>
```

```
<CENTER><H1>Search Ward</H1></CENTER>
```

```
</TD></TR></TABLE>
```

```
<%
```

```
If Request("NotFound") = "True" Then
    Response.Write "<BR><STRONG>This person is not given permission by any " &
        "physician to stay in a ward.</STRONG>"
Else
```

```
If Request("SameName") = "True" Then
    Response.Write "<BR><STRONG>Because there are more than one person " &
        "with the same name, please key in IC number.</STRONG>"
Else
```

```
If Request("Reserved") = "True" Then
    Response.Write "<BR><STRONG>This person already reserve a ward, " &
        "cannot reserve again.</STRONG>"
Else
```

```
If Request("JustGone") = "True" Then
    Response.Write "<BR><STRONG>The ward you choose just checked in "
    "by other patient.</STRONG>"
End If
```

```
End If
```

```
End If
```

```
End If
```

```
<%
```

```
>
```

```
<FORM ACTION="AddWard.asp" NAME="frmUser" METHOD="POST"
    onSubmit="return VerifyData()">
```

```
<TABLE BORDER=0>
```

```
<TR>
```



```

ddward.asp#

Response.Buffer = True
If Session("nurID") = "" Then
    Response.Redirect "NurseLogin.asp?NotFound=True"
End If%>
--#include file="wardTypeClassfd.asp"-->
ASEFONT COLOR="DarkBlue" face=serif>
TML>
EAD>
ITLE>Book A Ward For A Patient</TITLE>
HEAD>
ODY bgcolor=LightCyan>

If Request.Form("CheckinDate") = "" Then
    Dim rsUsers, varCounter, strName, strValue
    Set rsUsers = Server.CreateObject("ADODB.Recordset")
    rsUsers.Open "Ward_Type", objConn, adOpenForwardOnly, adLockOptimistic, ad
dTable

    If Request.Form("ICNumber") = "" Then
        rsUsers.Filter = "wtPatName = '" & Request.Form("PatientName") & "'"
        If Not rsUsers.EOF Then
            While Not rsUsers.EOF
                varCounter = varCounter + 1
                rsUsers.MoveNext
            Wend
            If varCounter > 1 Then
                Response.Redirect "Searchward.asp?SameName=True"
            End If
        End If
        rsUsers.Filter = "wtPatName = '" & Request.Form("PatientName") & "'"
        If rsUsers.EOF Then
            For each strField in rsUsers.Fields
                strName = strField.Name
                Session(strName) = ""
            Next
            rsUsers.Close
            Set rsUsers = Nothing
            Session("wtPatName") = Request.Form("PatientName")
            Response.Redirect "Searchward.asp?NotFound=True"
        Else
            For each strField in rsUsers.Fields
                strName = strField.Name
                strValue = strField.value
                Session(strName) = strValue
            Next
            rsUsers.Close
            Set rsUsers = Nothing
            Response.Redirect "Searchward.asp"
        End If
    Else
        rsUsers.Filter = "wtPatIC = '" & Request.Form("ICNumber") & "'"
        If rsUsers.EOF Then
            For each strField in rsUsers.Fields
                strName = strField.Name
                Session(strName) = ""
            Next
            rsUsers.Close
            Set rsUsers = Nothing
            Session("wtPatName") = Request.Form("PatientName")
            Response.Redirect "Searchward.asp?NotFound=True"
        End If
    End If
End If

```



```
Else
    For each strField in rsUsers.Fields
        strName = strField.Name
        strValue = strField.value
        Session(strName) = strValue
    Next
    rsUsers.Close
    Set rsUsers = Nothing
    Response.Redirect "Searchward.asp"
```

[illegible]

```

        <%Else%>
            ACTION="Choosedward.asp"
        <%End If%>
        NAME="frmUser" METHOD="POST">
        <TABLE BORDER=0><TR><TD WIDTH=80%></TD></TR>
        <%while Not rsSearchward.EOF
            varRoomCmp2 = Left(rsSearchward("wardID"),4)
            If varRoomCmp2 <> varRoomCmp1 Then
                Response.Write "<TR><TD><BR></TD></TR>"
                varRoomCmp1 = varRoomCmp2
            End If%>
            <TR><TD bgcolor=LightBlue><INPUT TYPE="RADIO" NAME="Availableward"
                VALUE="<%=rsSearchward("wardID")%>"
                <%=rsSearchward("wardID")%>
            </TD></TR>
            <%Session("CiwardID") = rsSearchward("wardID")
            rsSearchward.MoveNext
        Wend%>
        </TABLE>
        </BR>
        <INPUT TYPE=HIDDEN Name="ICNumber" VALUE="<%=Request.Form("ICNumber")%>
        <INPUT TYPE=HIDDEN Name="CheckinDate" VALUE="<%=Request.Form("CheckinDa
te")%>">
        <INPUT TYPE="Submit" VALUE="Submit">
        </FORM>
        <%Session("CiPatName") = Request.Form("PatientName")
        Session("CiPatIC") = Request.Form("ICNumber")
        Session("CiPatCheckinDate") = Request.Form("CheckinDate")
    Else
        If CDate(Request.Form("CheckinDate")) = date() Then
            objCommand.CommandText = "SELECT wardID, wReservePatCheckinDate FRO
M ward " & _
                "WHERE wardType = '" & varwardType & "' AN
D " & _
                "wardBed = '" & varNoOfBed & "' AND " & _
                "wStatus = 'R' " & _
                "ORDER BY wardID"

            set rsSearchward = objCommand.Execute
            set objCommand = Nothing

            If Not rsSearchward.EOF Then
                Dim varNoOfDays
                varNoOfDays = rsSearchward("wReservePatCheckinDate") - CDate(Requ
est.Form("CheckinDate"))
                If varNoOfDays > 3 Then
                    varRoomCmp1 = Left(rsSearchward("wardID"),4)%>
                    <TABLE BORDER=0><TR><TD WIDTH=10%>You are going to choose the bed
                        which is already reserved by other person. when the person comes
                        you have to transfer to another bed.</TD></TR></TABLE>
                    <FORM ACTION="CheckinForm.asp" NAME="frmUser" METHOD="POST">
                    <TABLE BORDER=0><TR><TD WIDTH=80%></TD></TR>
                    <%while Not rsSearchward.EOF
                        varRoomCmp2 = Left(rsSearchward("wardID"),4)
                        If varRoomCmp2 <> varRoomCmp1 Then
                            Response.Write "<TR><TD><BR></TD></TR>"
                            varRoomCmp1 = varRoomCmp2
                        End If%>
                        <TR><TD bgcolor=LightBlue><INPUT TYPE="RADIO" NAME="Availablew

```


rd"

```
                VALUE="<%=rsSearchward("wardID")%>">
                    <%=rsSearchward("wardID")%>
                </TD></TR>
                <%Session("CiWardID") = rsSearchward("wardID")
                    rsSearchward.MoveNext
                Wend%>
            </TABLE>
            <BR>
            <INPUT TYPE=HIDDEN Name="ICNumber" VALUE="<%=Request.Form("ICNumb
r")%>">
            <INPUT TYPE=HIDDEN Name="CheckinDate" VALUE="<%=Request.Form("Che
kinDate")%>">
            <INPUT TYPE="Submit" VALUE="Submit">
            </FORM>
            <%Session("CiPatName") = Request.Form("PatientName")
                Session("CiPatIC") = Request.Form("ICNumber")
                Session("CiPatCheckinDate") = Request.Form("CheckinDate")
            End If
        End If
    Else
        Response.write "<BR><STRONG>There is no more available ward for this
type, " & _
            "please change to another type.</STRONG>"
    End If
End If
rsSearchward.Close
set rsSearchward = Nothing

End If
>
/BODY>
/HTML>
```



```

<TR>
  <TD>IC no.:</TD>
  <TD><INPUT TYPE="Text" NAME="ICNumber" VALUE="<%= Session("CiPatIC")%>"
    SIZE="40"></TD>
</TR>
<TR>
  <TD>Check In Date:</TD>
  <TD><INPUT TYPE="Text" NAME="CheckinDate" VALUE="<%= Session("CiPatChecki
nDate")%>"
    SIZE="40"></TD>
</TR>
<TR>
  <TD>ward ID:</TD>
  <TD><INPUT TYPE="Text" NAME="Availableward" VALUE="<%= Session("CiwardID"
)%>"
    SIZE="40"></TD>
</TR>
<TR>
  <TD></TD>
  <TD ALIGN=CENTER COLSPAN=2><BR>
    <INPUT TYPE="Submit"
      <% If Session("CiwardID") = "" Then %>
        VALUE="Search Patient"
      <% Else %>
        VALUE="Submit"
      <% End If %>
    id=Submit1 name=Submit1>
    &nbsp;&nbsp;&nbsp;&nbsp;<INPUT TYPE="RESET" id=RESET1 name=RESET1></TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```



```

#Choosedward.asp#

%Response.Buffer = True
If Session("nurID") = "" Then
    Response.Redirect "NurseLogin.asp?NotFound=True"
End If%>
<!--#include file="wardTypeClsfd.asp"-->
%
If Request.Form("Availableward") = "" Then
    Dim rsUsers, varCounter
    Set rsUsers = Server.CreateObject("ADODB.Recordset")
    rsUsers.Open "Ward_Type", objConn, adOpenForwardOnly, adLockOptimistic, ad
CmdTable

If Request.Form("ICNumber") = "" Then
    rsUsers.Filter = "wtPatName = '" & Request.Form("PatientName") & "'"
    If Not rsUsers.EOF Then
        While Not rsUsers.EOF
            varCounter = varCounter + 1
            rsUsers.MoveNext
        Wend
        If varCounter > 1 Then
            Response.Redirect "CheckinForm.asp?SameName=True"
        End If
    End If
    rsUsers.Filter = "wtPatName = '" & Request.Form("PatientName") & "'"
    If rsUsers.EOF Then
        rsUsers.Close
        Set rsUsers = Nothing
        Session("CiPatName") = Request.Form("PatientName")
        Response.Redirect "CheckinForm.asp?NotFound=True"
    Else
        Session("CiPatName") = rsUsers("wtPatName")
        Session("CiPatIC") = rsUsers("wtPatIC")
        Session("CiPatCheckinDate") = rsUsers("wtPatCheckinDate")
        rsUsers.Close
        Set rsUsers = Nothing
        Set rsUsers = Server.CreateObject("ADODB.Recordset")
        rsUsers.Open "Ward", objConn, adOpenForwardOnly, adLockOptimistic, a
CmdTable
        rsUsers.Filter = "WReservePatIC = '" & Session("CiPatIC") & "'"
        If Not rsUsers.EOF Then
            Session("CiwardID") = rsUsers("wardID")
            End If
            rsUsers.Close
            Set rsUsers = Nothing
            Response.Redirect "CheckinForm.asp"
        End If
    Else
        rsUsers.Filter = "wtPatIC = '" & Request.Form("ICNumber") & "'"
        If rsUsers.EOF Then
            rsUsers.Close
            Set rsUsers = Nothing
            Session("CiPatName") = Request.Form("PatientName")
            Response.Redirect "CheckinForm.asp?NotFound=True"
        Else
            Session("CiPatName") = rsUsers("wtPatName")
            Session("CiPatIC") = rsUsers("wtPatIC")
            Session("CiPatCheckinDate") = rsUsers("wtPatCheckinDate")
            rsUsers.Close
            Set rsUsers = Nothing
            Set rsUsers = Server.CreateObject("ADODB.Recordset")

```



```

rsUsers.Open "ward", objConn, adOpenForwardOnly, adLockOptimistic, a
adCmdTable
rsUsers.Filter = "wReservePatIC = '" & Session("CiPatIC") & "'"
If Not rsUsers.EOF Then
    Session("CiwardID") = rsUsers("wwardID")
    End If
    rsUsers.Close
    Set rsUsers = Nothing
    Response.Redirect "CheckinForm.asp"
End If
End If

Else

If CDate(Request.Form("CheckinDate")) < Date() Then
    Session("CiPatName") = ""
    Session("CiPatIC") = ""
    Session("CiPatCheckinDate") = ""
    Session("CiwardID") = ""
    Response.Redirect "CheckinForm.asp?DueDate=True"
End If

Dim rswardType, rsward
Set rswardType = Server.CreateObject("ADODB.Recordset")
rswardType.Open "ward_Type", objConn, adOpenForwardOnly, adLockOptimistic,
adCmdTable

Set rsward = Server.CreateObject("ADODB.Recordset")
rsward.Open "ward", objConn, adOpenForwardOnly, adLockOptimistic, adCmdTab
le

rswardType.Filter = "wtPatIC = '" & Request.Form("ICNumber") & "'"
rsward.Filter = "wwardID = '" & Request.Form("Availableward") & "'" AND " &
    "wPatIC = '"

If Not rsward.EOF Then ' write personal details to record
    If rswardType("wtPatCheckinDate") > date() Then
        rsward("wStatus") = "R"
        rsward("wReservePatIC") = rswardType("wtPatIC")
        rsward("wReservePatCheckinDate") = rswardType("wtPatCheckinDate")
    Else
        rsward("wPatName") = rswardType("wtPatName")
        rsward("wPatIC") = rswardType("wtPatIC")
        rsward("wPatCheckinDate") = Request.Form("CheckinDate")
        rsward("wPhyName") = rswardType("wtPhyName")
        rsward("wStatus") = "I"
        If Request.Form("ICNumber") = rsward("wReservePatIC") Then
            rsward("wReservePatIC") = ""
            rsward("wReservePatCheckinDate") = NULL
        End If
    End If
    rsward.Update ' update the database
Else
    Response.Redirect "Searchward.asp?JustGone=True"
End If

Dim strName
For each strField in rswardType.Fields
    strName = strField.Name
    Session(strName) = ""
Next

```

```

For each strField in rsward.Fields
    strName = strField.Name
    Session(strName) = ""
Next

rswardType.Close
Set rswardType = Nothing

rsward.Close
Set rsward = Nothing

Session("CiPatName") = ""
Session("CiPatIC") = ""
Session("CiPatCheckinDate") = ""
Session("CiwardID") = ""

Response.Write Request.Form("ICNumber")
Response.Write "<BR>"
Response.Write "Book successfully<BR>"

End If

Dim strConnect%>
<!--#include file="wardClassfd.asp"-->

%If CDate(Request.Form("CheckinDate")) = date() Then
    Dim objCommand, rsDeletewardType
    Set objCommand = Server.CreateObject("ADODB.Command")

    objCommand.ActiveConnection = strConnect
    objCommand.CommandText = "DELETE FROM ward_Type " & _
        "WHERE wtPatIC = '" & Request.Form("ICNumber") & _
        "'"

    objCommand.CommandType = adCmdText

    objCommand.Execute rsDeletewardType
    Response.Write "This delete Command has affected " & _
        rsDeletewardType & " records<BR>"

    Set objCommand = Nothing
    Response.Redirect "CheckinForm.asp"
Else
    Response.Redirect "Searchward.asp"
End If
%>

```