## Faculty of Computer Science and Information Technology

## Universiti Malaya


| | |
|---|---|
| Name | : **NG MUN TING** |
| Matrix No. | : **WEK 98090** |
| Project Title | : **Meeting Scheduling System (MSS)** |
| Supervisor Name | : **Pn. Rodina Ahmad** |
| Moderator Name | : **Dr. Ow Siew Hock** |
| Course Code | : **WXES 3182** |
| Course Name | : **Projek Ilmiah Tahap II** |


**Session 2000/2001**

**Semester II**

# TABLE OF CONTENTS

**LIST OF FIGURES** ..... **VII**

**LIST OF TABLES** ..... **VIII**

**ABSTRACT** ..... **IX**

**ACKNOWLEDGEMENT** ..... **X**

**CHAPTER 1 INTRODUCTION** ..... **1**

1.1 PROJECT OVERVIEW ..... 1

1.2 PROJECT OBJECTIVES ..... 3

1.3 PROJECT SCOPE ..... 4

*1.3.1 System Administrator Section* ..... *4*

*1.3.2 Meeting Room Administrator Section* ..... *4*

*1.3.3 Equipment Administrator Section* ..... *5*

*1.3.4 User Section* ..... *5*

1.4 PROJECT MANAGEMENT ..... 6

**CHAPTER 2 LITERATURE SURVEY** ..... **8**

2.1 WHAT IS MEETING SCHEDULING? ..... 8

*2.1.1 The General Meeting Scheduling Problem* ..... *8*

*2.1.2 Automated vs. Interactive Meeting Scheduling* ..... *9*

2.2 INTRODUCTION TO COLLABORATIVE SYSTEM ..... 10

*2.2.1 Messaging Systems* ..... *11*

*2.2.2 Tracking Systems* ..... *12*

2.3 WEB BASED APPLICATION ..... 13

*2.3.1 What is Web Based Application?* ..... *13*

*2.3.2 Advantages of Web Application* ..... *13*

2.4 INTERNET AND INTRANET ..... 15

*2.4.1 What is Internet?* ..... *15*
*2.4.2 What is Intranet?* ..... *16*
*2.4.3 Implications on University Administrator* ..... *17*
2.5 CLIENT-SERVER CONCEPTS ..... 18
*2.5.1 Client-Server Computing* ..... *18*
*2.5.2 Client Process* ..... *19*
*2.5.3 Server Process* ..... *19*
2.6 REVIEW ON SYSTEM ARCHITECTURE ..... 20
*2.6.1 Classical Client-Server Architecture (Two-tier Architecture)* ..... *20*
*2.6.2 Multi-tier Architecture* ..... *20*
2.7 REVIEW ON PROGRAMMING LANGUAGE ..... 22
*2.7.1 Livewire (Server-Side JavaScript)* ..... *22*
*2.7.2 Active Server Pages (ASP)* ..... *22*
*2.7.3 Comparison Between Livewire and Active Server Pages* ..... *23*
*2.7.4 JavaScript* ..... *24*
*2.7.5 VBScript* ..... *24*
*2.7.6 Comparisons Between JavaScript and VBScript* ..... *26*
2.8 REVIEW ON DATABASE ..... 27
*2.8.1 Microsoft SQL Server 7.0* ..... *27*
*2.8.2 Microsoft Access 2000* ..... *28*
2.9 REVIEW ON COMMUNICATION AND COLLABORATION SERVER ..... 29
*2.9.1 Microsoft Exchange Server 5.5* ..... *29*
*2.9.2 Lotus Notes & Domino 4.6* ..... *30*
*2.9.3 Comparison of MS-Exchange Server 5.5 and Lotus Notes/Domino 4.6* . *31*
*2.9.4 Microsoft Exchange 2000 Server* ..... *32*
2.10 REVIEW ON WEB SERVER ..... 35
*2.10.1 Personal Web Server (PWS) 4.0* ..... *35*
*2.10.2 Microsoft Internet Information Server (IIS) 4.0* ..... *35*
*2.10.3 Microsoft Internet Information Server (IIS) 5.0* ..... *36*
2.11 REVIEW ON PLATFORM AND OPERATING SYSTEM ..... 38

*2.11.1 Microsoft Windows NT Server 4.0* ..... 38
*2.11.2 Microsoft Windows 2000 Server* ..... 38
*2.11.3 Microsoft Windows 2000 Advanced Server* ..... 39
2.12 REVIEW ON SOFTWARE DEVELOPMENT METHODOLOGIES ..... 41
*2.12.1 Waterfall model* ..... 41
*2.12.2 Spiral model* ..... 44
*2.12.3 Prototyping* ..... 45
**CHAPTER 3 SYSTEM ANALYSIS ..... 47**
3.1 TECHNIQUES USED TO DEFINED REQUIREMENTS ..... 47
*3.1.1 Interview* ..... 47
*3.1.2 Internet Research* ..... 47
3.2 MEETING SCHEDULING IN FSKTM ..... 48
*3.2.1 Current Scheduling Process* ..... 48
*3.2.2 Major Problems Encountered* ..... 49
3.3 PROPOSED MEETING SCHEDULING PROCESS USING MSS ..... 51
3.4 MEETING SCHEDULING SYSTEMS OBJECTIVES ..... 53
3.5 MEETING SCHEDULING SYSTEMS TERMS DEFINITION ..... 55
3.6 SOFTWARE DEVELOPMENT METHODOLOGY BEING EMPLOYED ..... 57
*3.6.1 The "Waterfall" Model* ..... 57
*3.6.2 Appropriateness of "Waterfall" Model To The Project* ..... 57
3.7 TOOLS BEING CHOSEN IN MEETING SCHEDULING SYSTEMS ..... 58
*3.7.1 Programming Language* ..... 58
*3.7.2 Database server* ..... 58
*3.7.3 Web server* ..... 58
*3.7.4 Communication and Collaboration Server* ..... 58
*3.7.5 Platform and Operating System* ..... 59
3.8 SYSTEMS REQUIREMENTS ..... 59
*3.8.1 Functional Requirements* ..... 59
*3.8.2 Non-Functional Requirements* ..... 66
3.9 EXPECTED OUTCOME ..... 68

**CHAPTER 4 SYSTEM DESIGN ..... 69**

4.1 SYSTEM FUNCTIONALITY DESIGN ..... 69

*4.1.1 System Architecture ..... 69*

*4.1.2 System Context Diagram ..... 70*

*4.1.3 System Structure Chart ..... 72*

4.2 SYSTEM DATABASE DESIGN ..... 73

*4.2.1 Entity-Relationship (E-R) Model ..... 73*

*4.2.2 Data Dictionary ..... 74*

4.3 DESIGN OF GRAPHICAL USER INTERFACE ..... 80

**CHAPTER 5 SYSTEM IMPLEMENTATION ..... 81**

5.1 INTRODUCTION ..... 81

5.2 ACTIVE DIRECTORY SERVICES ..... 81

*5.2.1 Lightweight Directory Access Protocol (LDAP) ..... 82*

*5.2.2 ADSI & CDO ..... 82*

*5.2.3 Building the LDAP Binding String ..... 82*

*5.2.4 Organizational Unit and Group in MSS ..... 83*

*5.2.5 Opening A Container or Organizational Unit ..... 83*

*5.2.6 Setting Properties Values in Active Directory ..... 84*

*5.2.7 Querying Active Directory Using ADO ..... 84*

5.3 CDO MESSAGING CLASSES AND INTERFACES ..... 85

5.4 CODING ..... 87

*5.3.1 Coding Methodology ..... 87*

*5.3.2 Coding Standards Used ..... 88*

**CHAPTER 6 SYSTEM TESTING ..... 90**

6.1 TESTING TECHNIQUES USED ..... 90

*6.1.1 White Box Testing ..... 90*

*6.1.2 Black box testing ..... 91*

6.2 TESTING STRATEGIES ..... 92

*6.2.1 Unit testing ..... 92*

*6.2.2 Integration Testing* ..... *95*
*6.2.3 System Testing* ..... *95*

**CHAPTER 7 SYSTEM EVALUATION & CONCLUSION** ..... **96**

7.1 PROBLEM ENCOUNTERED AND SOLUTIONS ..... 96
*7.1.1 Scope Not Properly Defined* ..... *96*
*7.1.2 Inexperienced in the chosen programming language* ..... *96*
*7.1.3 Lack of knowledge on Database server* ..... *96*
*7.1.4 Lack of knowledge on Microsoft Exchange 2000 Server* ..... *97*
7.2 SYSTEM STRENGTHS ..... 97
*7.2.1 Ease Meeting Scheduling Process* ..... *97*
*7.2.2 Manageability and Security* ..... *97*
*7.2.3 General Solutions* ..... *98*
*7.2.4 Graphical User Interface* ..... *98*
*7.2.5 Auto Sending E-mail Notification* ..... *98*
*7.2.6 Status View and Reports* ..... *99*
7.3 SYSTEM CONSTRAINTS ..... 99
*7.3.1 Browser Limitation* ..... *99*
*7.3.2 Report Is Not in "Printer-Friendly Format"* ..... *99*
*7.3.3 Checking of Availability is Based-On Whether Being Scheduled to Another Meeting Only* ..... *99*
7.4 FUTURE ENHANCEMENT ..... 100
*7.4.1 To Fully Utilize Microsoft Exchange 2000 Server* ..... *100*
*7.4.2 To Integrate With User Calendar or Task Schedule* ..... *100*
*7.4.3 To Provide "Printer-Friendly Format"* ..... *100*
*7.4.4 To Offer A Date Range in Organization of Meeting and Booking of Resources* ..... *100*
7.5 KNOWLEDGE AND EXPERIENCE GAINED ..... 101
7.6 CONCLUSION ..... 102

**REFERENCE** ..... **103**

**BIBLIOGRAPHY** ..... **105**

**APPENDICES** ..... **106**

MANUAL MEETING ROOM BOOKING FORM ..... 106

MEETING SCHEDULING SYSTEM - USER MANUAL ..... 108

# LIST OF FIGURES

Figure 2.1: The "Waterfall" model ..... 42
Figure 2.2: Spiral Model ..... 44
Figure 3.1 Flowchart of the current meeting scheduling process in FSKTM, UM .. 48
Figure 3.2: Flowchart of the proposed meeting scheduling process using MSS ..... 51
Figure 4.1: System architecture of MSS ..... 69
Figure 4.2: MSS Context Diagram ..... 71
Figure 4.3: MSS Structure Chart ..... 72
Figure 4.3: MSS E-R Diagram ..... 74

# LIST OF TABLES


Table 2.1 : Advantages and disadvantages of Internet ..... 15

Table 2.2 : Advantages and disadvantages of Intranet ..... 16

Table 2.3: Comparisons between Livewire and Active Server Pages ..... 23

Table 2.4: Comparison of Information Store Between MS-Exchange Server 5.5 and Lotus Notes/Domino 4.6 ..... 31

Table 4.1: MSS Database General Profile ..... 74

Table 4.2: Table Structure for MSSUser ..... 75

Table 4.3: Table Structure for Department ..... 75

Table 4.4: Table Structure for MeetRoom ..... 75

Table 4.5: Table Structure for Equipment ..... 76

Table 4.6: Table Structure for MeetingMaster ..... 76

Table 4.7: Table Structure for MeetingDetails ..... 77

Table 4.8: Table Structure for MRoomDetails ..... 77

Table 4.9: Table Structure for EquipDetails ..... 78

Table 4.10: Table Structure for ID ..... 78

Table 4.11: Table Structure for RefNo ..... 79

Table 4.12: HCI General Principles ..... 80

# ABSTRACT

Meeting Scheduling System (MSS) is a web-based system developed to support the organization of meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and the number of invitees that can attend the meeting. Besides that, Meeting Scheduling System (MSS) allow users to check if the invitees are available to attend a meeting, and will inform participants on details about meetings that involve them. It is to avoid the possibility of scheduling a person to attend two or more meetings at one time.

Other than that, Meeting Scheduling System (MSS) is to help user to keep track of availability of staff, maintain meeting room and equipment schedules and others right on user computer desktop. With Meeting Scheduling System (MSS), scheduling and tracking meetings rooms, equipment and other facilities is easier than ever. User can create schedule views with any number of rooms side by side. This makes meeting room scheduling fast, efficient and more professional.

Meeting Scheduling System (MSS) is communication and collaboration system. This allows user to communicate effectively via e-mail and to collaborate effectively through a central calendar, meeting system and workflow system.

This report will stress on MSS feature and design. The decomposition of MSS into modules and sub-modules, and function and features of each module are described in this document. Every phase throughout the entire development of the Meeting Scheduling System are discussed in this report.

# ACKNOWLEDGEMENT

I would like to take this opportunity to express my gratitude to all those who have helped and guided me in the completion of this report.

Firstly, I would like to thank my project supervisor, Pn. Rodina Ahmad, for her invaluable guidance, advice, encouragement, patient and constructive suggestions throughout the development of this project.

A special thanks to Dr. Ow Siew Hock, the project moderator for her comments and constructive suggestion.

I would like to extend my thanks to my friends and course mates for those who have given me a lot of assistance, comments and encouragement to make this project a successful.

Last but not least, my grateful thanks to all lecturers and staff of the Faculty of Computer Science and Information Technology, University of Malaya for being helpful during the requirement gathering stage.

# Chapter 1 INTRODUCTION

## 1.1 Project Overview

An organization usually consists of many members that are involved in the activities of the organization. When the need to make a decision arises, all parties in the associated with the problem will have to come together to discuss and decide the appropriate course of action. The process of discussing and deciding in a group is called a meeting. These meetings enable decision-making to be simplified, as more ideas will be conceived, probably less time-consuming and most important of all is that decisions are made in unison. However, the process of calling for a meeting is tedious as much overhead and planning is needed to make sure that the appropriate time, venue and attendees are determined and that all the attendees are notified.

University administrators are especially overwhelmed with the scheduling demands placed on them to find available appointment and meeting times, confirm appointments, and locate existing ones when they need to be changed or cancelled. For most of them, most of the time is spent doing nothing but scheduling as planning events while taking into consideration all the lecturers' busy schedules is a very difficult task. Sometimes, they have to swap back and forth 25 or more email messages just to schedule a single meeting between few members in the organization. The chaos often means missed calls, conflicting schedules, and long waiting times.

Meeting Scheduling System (MSS) is to support the organization of meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and number of attendees that can attend the meeting. It is also to avoid the possibility of scheduling a person to attend two or more meetings at one time. This is feasible to be achieved as Meeting Scheduling System allow users to check if all the lecturers are available to attend a meeting, and will inform participants of details about meetings that involve them.

Other than that, Meeting Scheduling System (MSS) is to help user to keep track of availability of staff, maintain meeting room and equipment schedules and more right on user computer desktop. With Meeting Scheduling System (MSS), scheduling and tracking meetings rooms, equipment and other facilities is easier than ever. User can create schedule views with any number of rooms side by side. This makes meeting room scheduling fast, efficient and more professional.

Meeting Scheduling System not only save time and effort on the part of human, but also lead to more efficient results.

## 1.2 Project Objectives

MSS is designed to support administrators, clerks and lecturers in the Faculty of Computer Science and Information Technology, University of Malaya in organizing meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and the number of attendees that can attend the meeting. It is also to avoid the possibility of scheduling a person to attend two or more meetings at one time. Lastly, MSS is to provide a feasible environment for them to organize meetings from the perspective of cost and time.

The objectives of MSS can be divided into five main points:

(i) To create an integrated system to schedule meeting and booking of meeting resources (rooms and equipment) effectively and systematically.

(i) To solve conflicts in meeting scheduling process, that is to avoid the possibility of scheduling a person to attend two or more meetings at one time and solve conflicts in meeting resources (room, equipment) booking.

(ii) Reducing faculty administrators' workload and redundant jobs in respect of scheduling meetings and clerks' workload in distributing memos.

(iii) To provide a communication and collaboration feature for the entire environment in meeting scheduling, that will enable administrators, clerks and lecturers to communicate easily and work together effectively.

(iv) To develop an Internet web based Meeting Scheduling System (MSS) to allow user to schedule, edit or cancel meeting and administrator to maintain the MSS from any location at any time at their convenience.

## 1.3 Project Scope

MSS project will include the every staff and academic user in the faculty of Computer Science & Information Technology, University of Malaya.

### 1.3.1 System Administrator Section

System Administrator Section contain the following modules and sub-modules:-

- **Authentication and Authorization Module**
- **User Setting Module**
- **Maintenance Module**
  1. Meeting Room Maintenance
  2. Equipment Maintenance
- **Cancellation Module**
  1. Meeting Request
  2. Meeting Room Booking
  3. Equipment Booking
- **Report Module**
  1. Meeting Request
  2. Meeting Room Booking
  3. Equipment Booking

### 1.3.2 Meeting Room Administrator Section

Meeting Room Administrator Section contain the following modules and sub-modules:-

- **Authentication and Authorization Module**
- **Maintenance Module**
- **Approval Module**
- **Cancellation Module**
- **Report Module**

### 1.3.3 Equipment Administrator Section

Equipment Administrator Section contain the following modules and sub-modules:-

- **Authentication and Authorization Module**
- **Maintenance Module**
- **Approval Module**
- **Cancellation Module**
- **Report Module**

### 1.3.4 User Section

User Section contain the following modules and sub-modules:-

- **Authentication and Authorization Module**
- **Personal Scheduler Module**
- **New Request Module**

1. Initiate Meeting
2. Book Meeting Room Only
3. Book Equipment Only

- **Incoming Request Module**

1. Reply Meeting Invitation
2. Emergency Withdrawal

- **Cancellation Module**

1. Meeting Request
2. Meeting Room Booking
3. Equipment Booking

- **Amendment Module**

1. Meeting Request
2. Meeting Room Booking
3. Equipment Booking

- **Report Module**

1. Meeting Request
2. Meeting Room Booking

3. Equipment Booking

- **Reminder Module**

## 1.4 Project Management

As written, "Effective management of a software project depends on thoroughly planning the progress of the project, anticipating problems which might arise and preparing tentative solutions". This shows how important project management should be considered and is an on-going process that is from start to end.

A project schedule describe the software development cycle for a particular project by enumerating the phases or stages of a project and breaking each into discrete tasks or activities to be done. The schedule also portrays the interactions among these activities and estimates the time that each task or activity will take [1]. A project schedule is needed to serve as the time guidance for a developer to determine what type of tasks to be carried and what goals should be achieved when a certain milestone is met. Sometimes due to unforeseen factors, certain tasks have to be prolonged by getting more time allocation. Attempted have been made to reduce inherent uncertainty in determining time estimates by projecting most likely, pessimistic, and optimistic estimates to determine the expected time an activity will take.

Project schedule are divided into ten items as listed below :-

1. System analysis
2. Planning
3. System design
4. Design review
5. Documentation 1
6. System development
7. Development review
8. System Implementation and Testing

9. Implementation and Testing Review
10. Documentation 2

After every stage, reviews are carried out in order to point out the improvements and discover the defects in each stage. This is essential as each defect can be corrected before the subsequent stage of the development process commences.

# Chapter 2 LITERATURE SURVEY

## 2.1 What is Meeting Scheduling?

Meeting Scheduling System (MSS) is to support the organization of meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and number of attendees that can attend the meeting. It is also to avoid the possibility of scheduling a person to attend two or more meetings at one time and to allocate the resources so that its usage wills most benefits the organizations.

### 2.1.1 The General Meeting Scheduling Problem

The process of calling for a meeting is tedious as much overhead and planning is needed to make sure that the appropriate time, venue and attendees are determined and that all the attendees are notified.

Planning meetings have to take consideration of people's busy schedules, and are a job no one will to want to do it alone. Most of our time is spent doing nothing but scheduling. Another problem was the lack of interaction between other people. This cause the initiator of the meeting have to swap back and forth 25 or more e-mail messages just to schedule a single meeting with few of the member in the organization.

The front desk of many office staffs can be a hectic place as it overwhelmed with the many scheduling demands placed on them to answer calls, find available appointment and meeting times, confirm appointments, and locate existing ones when they need to be changed or cancelled. The chaos often means missed calls, conflicting schedules, long waiting room times, frustrated customers and staff, and lost revenue.

### 2.1.2 Automated vs. Interactive Meeting Scheduling

The size and complexity of modern university scheduling problem has provoked a trend, towards more general problem-solving algorithms with the automated system [8]. A full search of all possible schedules would be unacceptable for anyone needing to produce schedules in a restricted amount of time. The scheduler must settle for an optional solution in return for a gain in speed and this could be done with the automated system [8]. But many people believed that the scheduling problem couldn't be completely automated.

There are two reasons :-

1. There are some reasons that make one schedule better than other one that cannot easily determined in an automated system.
2. Searching for a schedule that satisfies all the constraints usually complicated in the system, a human intervention may be needed to produce an ideal schedule, which the system by itself may be not able to find a solution. As a result, most of the system allows the (semi-automated) scheduling systems [9].

## 2.2 Introduction to Collaborative System

Collaboration is the integration of many different technologies into a single application or environment to facilitate information sharing and information management [2]. Integrated technology, however, is only one aspect of collaboration as we're defining it. Timing is another. In real-time collaboration in which user work with others at the same moment, taking turns communicating ideas. But new technology offers user an entirely different way to collaborate, which is asynchronous collaboration; in which user don't have to be present to participate. Asynchronous collaboration allows user, at user convenience. E-mail, public databases, the Internet, and intranets are all forms of asynchronous communication.

Collaborative technology provides these key benefits :-

- Extensive, secure communication.

  Collaboration technologies enable extensive communication through many different mediums and secure communications through encryption and digital signature technology, which is critical as organization increases their use of the Internet.

- Storage of information in a central location.

  Information is placed in a central repository, or database, so that individuals inside and outside a corporation can access it. If shown in a threaded view, the history of the information is accessible and new information can be added to it.

- Ability to extend technologies with new functionality and bridge islands of information.

  Collaborative systems connect disparate systems and facilitate finding and sharing information stored in existing technologies.

Characteristics of a collaborative system that enable it to provide the above-mentioned benefits to an organization :-

1. It have a robust, replicated object database that can store many different types of information such as web pages, office documents, and e-mail messages, and it must support replication both from server to server and from server to client. To work with data, the database needs to allow many different clients, ranging from web browsers to e-mail clients.
2. It must support the Internet and industry standards. New technologies are connecting disparate networks to form one global, cohesive network. A collaborative system must be able to interoperate with these networks over the Internet, and it must follow industry standards to allow openness to a large number of external systems as well as guarantee the integrity of the data.
3. A collaborative system must offer powerful, easy-to-use development tools and technologies. The environment must be open so that developers can use any tool to develop solutions and users can access and customize the user interface.

Below are the few example of collaborative solutions:-

### 2.2.1 Messaging Systems

An organization needs information in order to get work done. Thus, messaging has become a mission-critical function in most organization. While e-mail is still the core ingredients, other applications are now included in this category.
The category of messaging can be divided into the following subcategories :-

#### 2.2.1.1 E-mail

Electronic Mail, also known as e-mail programs allows user to create, send, read, store and manipulate electronic messages and attachments. E-mail is an example of "push-style" communication, meaning that the sender initiates the communication of organization, these programs have evolved from merely creating and sending text messages, into multi-featured programs [3].

#### 2.2.1.2 Workflow

Workflow applications are primarily constructed around three concepts, which are:-

1. Roles.

A role is the logical representation of a person or a person or an application in a workflow process. Roles can change dynamically depending on who is involved in the particular process.

2. Routes.

A route defines what information will route and who will receive it. Routes can be sequential, parallel, conditional, or any combination of these.

3. Rules.

A rule is conditional logic that assesses the status of the workflow process and determines the next steps. A rule can be based on the properties of a message or on some other data source.

#### 2.2.1.3 Electronic forms

Electronic forms are electronic messages with built-in fields. They can be used instead of paper-based forms to automate and streamline organizational processes. Items such as expense reports, meeting room booking, equipment requests can all be implemented with electronic forms [3].

### 2.2.2 Tracking Systems

Tracking applications manage and track information, such as a list of contacts, form its creation to its deletion or "completion". Tracking applications usually require the integration of many different data sources because the information needing to be tracked resides in more than one location [2].

## 2.3 Web Based Application

### 2.3.1 What is Web Based Application?

Web based functionality is achieved when built on the foundations of the World Wide Web, such applications can be run anywhere in the world at any time and are completely cross platform. All data and input by the user using the application is used to update a database or document that is then made available by the system to global users in real time.

The only client side software needed to access and execute Web-based applications is a web browser environment. An example of such an application would be an online store accessed via Netscape Navigator or Microsoft Internet Explorer.

Web applications provide a rich interactive environment through which the user can further define their unique online experience. Without web applications to breathe life and provide user-interaction, a web page is limited to static electronic text.

Key features of a Web based application are centrally located application, database and documents, information is updated in real time, and system as well as applications is accessible remotely.

### 2.3.2 Advantages of Web Application

Web based applications have many advantages over traditional network architecture. Some of the advantages of Web based architecture are:

- It allows users to log onto the system from anywhere in the world as long as they have a computer, an Internet connection and a Web browser.
- Applications are resident on the server.
- All data input into the system is done so in real-time.
- Data is available in real-time.

- User accessibility to data is definable.
- System administration can be performed remotely.
- Distributed architecture provides for a stable system with feature like database mirroring and redundant servers eliminating any type of system downtime.
- Web based solutions are based on the web principle of open architecture which eliminates the integration problem.

## 2.4 Internet and Intranet

### 2.4.1 What is Internet?

The Internet is a global network of computers that communicate using a common language. It is similar to the international telephone system-no one owns or controls the whole thing, but it is connected in a way that makes it work like one big network.

The World Wide Web (WWW) gives you a graphical, easy-to-navigate interface for looking at documents on the Internet. These documents, as well as the links between them, comprise a "web" of information. Files or "pages" on the Web are interconnected. User can connect to other pages by clicking special text or graphics, which are called hyperlinks.

Pages can contain news, images, movies, and sounds, just about anything. These pages can be located on computers anywhere in the world. When you are connected to the Web, you have equal access to information worldwide; there are no additional long-distance charges or restrictions.

#### 2.4.1.1 Advantages and Disadvantages of Internet

Table 2.1 : Advantages and disadvantages of Internet

| Advantages | Disadvantages |
|---|---|
| 1. Can access any computer anywhere with Internet access. | 1. Limited access to intranet resources. |
| 2. In terms of design, anything is possible such as great multimedia and video presentations. | 2. In terms of security, there is a concern that unauthorized users could access proprietary information. |
| 3. Take advantage of external resources by linking to vendor websites, health databases, etc. | |

### 2.4.2 What is Intranet?

The Intranet is essentially a private Internet operating on an organization's internal network. Intranets exploit the incredibly popular and low-cost Internet tools to gain strategic advantage over competitors, cut costs, and improve operational effectiveness. It may consist of many interlinked local area networks and also use leased line in the wide-area network.

It may or may not include connections through one or more gateways to the outside Internet. An intranet uses TCP/IP, HTTP, and other Internet protocols and in general looks like a private version of the Internet. With tunneling, companies can send private messages through the public network, using the public network with special encryption/decryption and other security safeguards to connect one part of their intranet to another.

#### 2.4.2.1 Advantages and Disadvantages of Intranet

Table 2.2 : Advantages and disadvantages of Intranet

| Advantages | Disadvantages |
|---|---|
| 1. All computers on the intranet have easy, instant access. | 1. Only computers linked to the company intranet have access. Limited or no access from home, satellite offices, or any computer not on the intranet. |
| 2. All computers are connected by high-speed connections. | 2. Limits on design: some elements, especially multimedia, may not be possible due to hardware/software restrictions. |
| 3. In terms of security, limited worries about unauthorized access. | 3. Limited access to other resources, such as resources on the Internet may not be accessible due to security reasons. |

### 2.4.3 Implications on University Administrator

Administrators and lecturers in university see intranets and the Internet as a way to:

- Increase student access and the equity of that access to instructional information and educational resources for better education.
- Increase staff and teacher access to real-time administrative data for decreased administrative tasks and costs and better decision-making and efficiency.
- Increase the dynamic nature of community collaboration and better prepare for the future.

Based on a quantitative survey of academic staff use of Internet in International Islamic University Malaysia (UIA) done by a group of lecturer in Australia, 63.3% of response rate is achieved to a mail questionnaire. It showed that the uses of Internet among university administrators are quite high. In that survey, almost all respondent found e-mail useful for communication with colleagues. Besides that, the respondents think that Internet is very useful for research, publications, and for making a contribution to a profession [4].

## 2.5 Client-Server Concepts

Client/server is a distributed software architecture within which systems are divided into autonomous processes, allowing clients to request functions, which are then carried out by the server. In such an environment, the end users are more flexible and they can manipulate the data they need on a local system. Other than that, client/server is a concept in which tasks are distributed over clients and servers, allowing tasks to be carried out by those machines, which are most suitable for task execution.

The client/server concept is based on a dialog between the client and server. A client passes a given request on to the server and the server executes a task. Each task is processed on a client or server is executed by software module. The software modules in a client/server concept are able to work together. The server takes action only after having received a software request from the client. The server tasks do not in fact necessarily have to run on different machines [5].

### 2.5.1 Client-Server Computing

Client/server computing is the logical extension of modular programming. Modular programming has as its fundamental assumption that separation of a large piece of software into its constituent parts ("modules") creates the possibility for easier development and better maintainability. Client/server computing takes this a step further by recognizing that those modules need not all be executed within the same memory space. With this architecture, the calling module becomes the "client" (that requests a service), and the called module becomes the "server" (that provides the service).

The logical extension of this is to have clients and servers running on the appropriate hardware and software platforms for their functions. For example, database management system servers running on platforms specially designed and configured to perform queries, or file servers running on platforms with special

elements for managing files. It is this latter perspective that has created the widely believed myth that client/server has something to do with PCs or Unix machines [6].

### 2.5.2 Client Process

The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service). Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic. The client-based process is the front-end of the application that the user sees and interacts with. The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. The client process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation CPU and peripherals. One of the key elements of a client workstation is the graphical user interface. Normally, a part of operating system i.e. the window manager detects user actions, manages the windows on the display and displays the data in the windows [7].

### 2.5.3 Server Process

A server process (program) fulfils the client request by performing the task requested. Server programs generally receive requests from client programs, execute database retrieval and updates, and manage data integrity and dispatch responses to client requests. Sometimes server programs execute common or complex business logic. The server-based process may run on another machine on the network. This server could be the host operating system or network file server; the server is then provided both file system services and application services. Or in some cases, another desktop machine provides the application services. The server process acts as a software engine that manages shared resources such as databases, printers, communication links, or high powered- processors. The server process performs the back-end tasks that are common to similar application [7].

## 2.6 Review on System Architecture

### 2.6.1 Classical Client-Server Architecture (Two-tier Architecture)

Classical client/server application architecture (two-tier architecture) is based on client computer where services and queries are constructed and a data call connection is made to the server computer, with no intervening server. The results of the query are returned as data stored within the local cache of client computer. A visual display will be connected to this cache for displaying the query's results and data. High-performance data retrieval technologies became an important consideration when constructing the entire architecture.

It is typically used in small environments (less than 50 users). A common error in client/server development is to prototype an application in a small, 2-tier environment and then scale up by simply adding more users to the servers. This approach will usually result in an ineffective system, as the server becomes overwhelmed. To properly scale to hundreds or thousands of users, it is usually necessary to move to a three-tier architecture.

### 2.6.2 Multi-tier Architecture

The core design of a multi-tier, or three-tier, computing is to enable application codes to be run separately from the client computer and the database server. Three- tier architecture introduces a server (or an "agent") between the client and the server. The role of the agent is many folds. It can provide translation services (as in adapting a legacy application on a mainframe to a client/server environment), metering services (as in acting as a transaction monitor to limit the number of simultaneous requests to a given server), or intelligent agent services (as in mapping a request to a number of different servers, collating the results, and returning a single response to the client [6].

The introduction of a server (or an "agent") between the client and the server supports the performance of large application processing in the middle tier. This application architecture is beneficial to developers as moving the data access

and business logic to the middle tier denotes less application code for client's computers. It also eases the management of business process components and data access logic. Therefore, multi-tier architectures present a more scalable and flexible way for building application.

## 2.7 Review on Programming Language

### 2.7.1 Livewire (Server-Side JavaScript)

Livewire is the server side JavaScript that is used in creating programs which runs on the server side. It has been introduce in the Netscape Enterprise server. It functions capabilities are similar with the JavaScripting.

### 2.7.2 Active Server Pages (ASP)

Active Server Pages is an open application environment programming language in which HTML pages, scripts, and ActiveX components can be combined to create Web-based applications. It is the latest server based technology from Microsoft Corporation. The basis of ASP is the Internet Information Server (IIS). An .asp file is a special HTML file containing hypertext, client-side script and Active Server Page Script for a web page. In other words, it is a standard HTML documents interfaced with ActiveX script code that call specific Active Server Components. It is used to create dynamic and interactive web pages that are able to include Active X controls and Java Applets. The pre-built Active Server Components provide plug-in objects that will perform specific tasks. The script is able to instantiate objects, calling their methods, manipulating their properties and producing results that are related to the web page. Client side script is decoded and run by the browser itself. Meanwhile, Asp script is decoded and run by Microsoft Internet Server (IIS) in order to create web pages to the browser.

The client side and ASP scripts can be written in any scripting language like VBScript, Jscript or Java Script. Script is inserted between anywhere in a html file. Active Database objects, one of the Active server Components allows easy but powerful connections to be made to almost any database system suitable for publishing and collecting data on the web. ASP provides a way for building secure transactions, server-based applications and web sites that work together with Windows NT and IIS. The entire combination provides a comprehensive set of key software technologies, which enables secure exchange of information over public

networks, access control to server resources and confidential identification of server and client. It is also considered as a glue technology, which binds together other various server-based systems to help build interactive web pages. ASP is able to interact with almost any existing dynamic web page technology such as CGI (Common Gateway Interface), ISAP (Internet server Application Programming Interface) and scripts written in PERL, Python and Awk. Besides that, ASP is suitable for building multi tier Internet and Intranet applications and supports client-server programming. In addition, it is able to spread processing load between client and server by implementing and integrating client-side processing as well as server side processing.

### 2.7.3 Comparison Between Livewire and Active Server Pages

Table 2.3: Comparisons between Livewire and Active Server Pages

| | Netscape LiveWire (sever-side JavaScript) | Microsoft ASP (Active Server Pages) |
|---|---|---|
| Platforms | Runs on Unix platforms, OS/2, Novell, and Windows NT | Runs on Windows 95 and Windows NT |
| Languages | Supports JavaScript | Supports VBScript, JavaScript |
| Session Maintenance | Saves and retrieves variables per user session. Advantage over CGI. | Same as LiveWire. |
| Database Connectivity | Connects via ODBC and native drivers for Oracle, Sybase, Informix, and DB 2 Pools db connections for better performance. Provides access to db cursors and also provides execution of SQL statements and stored procedures. | Connects via ODBC, same as LiveWire on other connectivity points. |
| Calling External Libraries | Provides interfaces to C, C++, and Java LiveConnect can create JavaScripts objects,which call Java | Provides interfaces to C++, visual basic and Java. |

| | | |
|---|---|---|
| | classes, which call JavaScript. | |
| Development Libraries | Sports debugging feature in Application Manager. Useful for debugging SQL statements and other errors. | Debugging features available in future Microsoft Visual Development environment for db access and site management tools (Front Page). |

### 2.7.4 JavaScript

JavaScript is a cross-platform scripting language, which is simple, interpreted, and object-oriented. It can be used to add simple interactive behaviors to an HTML page by means of a script of keywords inserted into a web page. It's originated from Live Script that developed by Netscape to provide a way to interface with Java. Sun, the developer of Java, helped Netscape rework LiveScript and called it JavaScript. Anyway, JavaScript is not derivative of java. It lacks of power of full-featured programming language. Netscape support the language since its version 2.0 and Microsoft Internet Explore giving support since its Internet Explorer 3.0 through Jscript.

The main roles of JavaScript play in the web pages are form validation, responding to input, dialog boxes, detecting browser characteristics, updating the browser properties, math capabilities, using cookies to keep visitor information, date and time information, integrating with Java, basic graphics and dynamic HTML.

### 2.7.5 VBScript

VBScript, developed by Microsoft, is a powerful, lightweight, easy-to-use, freely available, cross-platform, and cross-language scripting language for the Internet. VBScript is designed to leverage the skills of millions of Visual Basic programmers to the Internet.

Below are the characteristics of VBScript :-

- VBScript is powerful. Various capabilities of VBScript can be used to develop richly interactive Web pages that respond to user input in an intelligent manner. For example, when a user submits a form, a VBScript subroutine can be triggered to verify that the form is properly filled in with valid values.

- VBScript is lightweight. VBScript code is lightweight, fast, and has been optimized to be transmitted via the Internet. Because VBScript code is lightweight, it can be quickly transmitted to users browsing a Web site--even via relatively slow Plain Old Telephone Service (POTS) links to the Internet.

- VBScript is easy to use. VBScript is easy to use compared to scripting languages such as JavaScript. This is because VBScript is based on the easy-to-learn BASIC (Beginner's All-Purpose Symbolic Instruction Code) language.

- VBScript is cross-platform. VBScript can be functioning on UNIX as well as Macintosh computers, in addition to Windows 95 and Windows NT computers.

- VBScript is cross-language. VBScript supports many languages, such as C++ and Java, which allows objects to be compiled as ActiveX controls. VBScript is the glue that binds various ActiveX controls to create sophisticated Web applications. For example, a Timer control developed using the C++ language is capable of keeping the time and a Stock Sticker control developed using Java is capable of displaying a stock sticker. By themselves, these two components perform very limited tasks. However, with the aid of VBScript, the Timer control can "talk" to the Stock Sticker control to create a more sophisticated Web application. The Timer control, for exarnple, can trigger a VBScript subroutine, which in turn triggers the Stock Sticker control to update itself.

- VBScript is a subset of Microsoft Visual Basic, and is upwardly compatible with Visual Basic for Applications (VBA). Indeed, VBScript is a subset of VBA. VBA is shipped with Microsoft Office applications to make it easier for developers to build custom solutions using Microsoft Office applications.

### 2.7.6 Comparisons Between JavaScript and VBScript

VBScript is more strategic than Jscript as VBScript has been designed to leverage the skills of millions of Visual Basic programmers to the Internet. Although JavaScript is a powerful scripting language, it is not as easy to learn and use as VBScript. Virtually any task that can be accomplished using JavaScript can be accomplished using VBScript. VBScript can be used to easily use to automate various ActiveX controls in a Web page, it can be used to develop sophisticated and intelligent Web applications. As VBScript is supported by Microsoft, there is a great deal of integration between VBScript and Internet Explorer, Windows NT/95, Microsoft Office, and Microsoft BackOffice in the near future. Spend some time with VBScript to learn how it can enhance a Web site by making it easier and more exciting to navigate. Other than that, the primary and default scripting language for ASP is VBScript, most of the example are presented only in VBScript, largely because it provides the greatest transparency and therefore, functionality with the web server.

## 2.8 Review on Database

### 2.8.1 Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 is a relational database management system for Windows NT-based systems. It is designed to meet the requirements of distributed client-server computing. Microsoft SQL Server 7.0 is tightly integrated and coupled with other Microsoft BackOffice family of servers to enable organizations to improve decision making processes and enhancing their business processes. It allows information to be replicated to non-SQL Server databases including Microsoft Access, ORACLE, Sybase, and DB2.

Microsoft SQL Server 7.0 is a scalable, high-performance database management system designed specifically for distributed client/server computing. Microsoft SQL Server 7.0 provides tight integration with windows and windows-based applications helping reduce the cost and complexity of deploying sophisticated applications. Microsoft SQL Server 7.0 is an ideal database engine for powering web sites. Through tight integration with Internet Information Server, Microsoft SQL Server 7.0can be queried and updated via popular web browsers. SQL Server's native Open Database Connectivity (ODBC) lets it inter-operate smoothly with the Internet Database Connector interface included with Internet Information server.

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 columns from as many as 16 tables in a single query.

### 2.8.2 Microsoft Access 2000

Microsoft Access is a Windows-based relational database management system. Microsoft Access 2000 provides powerful tools that help in organizing and sharing of the database so the team member in an organization make better decisions. Microsoft Access is one of the most popular database management software as it is powerful and it is also easy to learn and use. Microsoft Access can be used to enter, store and manipulate data in a variety of ways. Other than that, it can be used to query for specific information or to produce (detailed or summary) reports based on certain criteria.

Microsoft Access uses the new Microsoft Data Engine (MSDE), a SQL Server–compatible client/server engine designed for a single-user computer or small workgroup server. However, there are some weaknesses in Microsoft Access, which is very slow transaction speed. Other than that, queries are stored locally (all data is pulled over the network instead of just the few records you want).

## 2.9 Review on Communication and Collaboration Server

### 2.9.1 Microsoft Exchange Server 5.5

Microsoft Exchange Server 5.5 is the ideal platform for critical messaging. With unlimited storage capacity, built-in SMP support, improved backup performance, and enhanced security management; Microsoft Exchange 5.5 is the best choice for your messaging foundation. You won't find a better performing, more scalable, more stable messaging platform on the market.

Microsoft Exchange 5.5 is built on architecture of Internet standards such as SMTP, POP3, IMAP4, LDAP v.3, and NNTP, with high-fidelity message delivery across a variety of clients. Microsoft Exchange 5.5 gives you the tools and technology to develop the widest range of collaborative solutions. Start with instant collaboration using Microsoft Outlook 2000 desktop information manager, Microsoft Office 2000, and the drag-and-drop Microsoft Exchange Server design environment or build fully featured Web applications using standard Web tools such as Microsoft Visual InterDev integrated web application development system or Microsoft Visual Studio developmental studio. Microsoft Exchange 5.5 offers the freedom to match development solution to user needs while taking advantage of the extended integration with other Microsoft BackOffice family of servers such as Microsoft SQL Server, Microsoft Systems Management Server, and Microsoft Internet Information Server (IIS).

The integrated diagnostics, remote management, and flexible directory support in Microsoft Exchange 5.5 results in the easiest setup, configuration, and administration on the market. In addition, Exchange continues to be the lowest cost client/server messaging solution available

Integration with Microsoft SQL Server allows user to send messages triggered by SQL Server events. Exchange Scripting Agent enables automated data transfer between the Exchange and SQL stores. Integration with Microsoft Systems Management Server allows user to deploy Exchange clients and Outlook clients

painlessly to user desktops and monitor the client software for upgrades. Integration with IIS using Collaboration Data Objects allows you to access all of the Exchange Resources through a Web browser.

### 2.9.2 Lotus Notes & Domino 4.6

Lotus Note & Domino 4.6 is a groupware applications built to support communication and collaboration of process which flows within a company. Lotus Domino is a server that maintains the entire management of the groupware. However, the integration of Lotus Notes within the environment of Microsoft Operating System and Offices has draw a clear description regarding the loopholes faced by Lotus Notes. Firstly, Notes on the client does not work and cooperate poorly with other Microsoft Office products. For many development of enterprise applications, it only allows its own scripting (Lotus Script) and objects models. Lotus Notes only works well with data that are stored in Notes database and other database remains second priority in its deployment. Due to its operating system-independent feature, Lotus Notes will need to duplicate basic functions such as directory, security and administration within its boundary.

### 2.9.3 Comparison of MS-Exchange Server 5.5 and Lotus Notes/Domino 4.6

Table 2.4: Comparison of Information Store Between MS-Exchange Server 5.5 and Lotus Notes/Domino 4.6

| | MS-Exchange Server 5.5 | Lotus Notes/Domino 4.6 |
|---|---|---|
| Information Store Size | Size of single Information Store can be up to 16 gigabytes | Size of single Information Store is only up to 4 gigabytes |
| Transaction Logs | It logs all steps during messaging and collaboration transactions that take place on the system. Transactions are not committed until they are complete. In the event of power loss or unexpected server shut down, valuable data remains intact and recoverable, right up to the last transaction. This also prevents data from being recorded in an inconsistent or corrupted state. The result is improved performance and data integrity. | Lotus Notes and Domino do not use transaction logs, and IBM and Lotus have argued that transaction logging requires overhead that slows system performance. |
| Fast recovery using transaction rollback. | When Microsoft Exchange is restarted after abnormal server shutdown, the transaction log file is scanned for incomplete transactions. If one is found, the transaction rolls back to the state it was before the incomplete transaction took place. | Notes and Domino, which must rebuild database views from the ground up any time data integrity is violated. |
| Fast recovery of | The information store in Microsoft Exchange Server 5.5 allows | Lotus Notes and Domino provide no such functionality. |

| | | |
|---|---|---|
| accidentally deleted items | administrator to grant end users the right to recover items that were deleted accidentally quickly and easily from the client desktop. | |
| Single-instance storage with referential integrity | Microsoft Exchange Server 5.5 provides administrators with end-to-end message tracking. For example, if a user complains that a message was sent but never received, the administrator can easily recover the tracking history of the message in graphical display that maps every server, and connector that ever touched the message. | Lotus Notes and Domino have only limited facilities to set up manual tracking on messages before they are sent and are unable to locate arbitrary messages after they have been sent out. |

### 2.9.4 Microsoft Exchange 2000 Server

Microsoft Exchange 2000 Server inherits all the feature of Microsoft Exchange Server 5.5 and above that, with added features as it combines messaging, collaboration, and application development. It helps businesses improve the way employees work together by integrating workflow design with Web communications, and by using a common way to work with messages, documents, and applications. Microsoft Exchange 2000 also provides a communications infrastructure that gives employees access to information at any time, from anywhere, using technologies including wireless communication, handheld devices, and teleconferencing. The benefits of using Microsoft Exchange 2000 include improved individual and team productivity, streamlined business processes with reduced management expense, and reduced hardware and software cost.

Microsoft Exchange 2000 Server helps you get the most out of your investment in Windows 2000 Server by taking advantage of the following Windows 2000 services and capabilities:

- **Directory**: Integration with Windows 2000 Server Active Directory service results in a single directory of user information, networking, and messaging resources. This simplifies overall systems management and reduces cost of ownership. In addition, a unified directory makes it easier for users to find all kinds of information such as e-mail addresses and shared printers. Implementing Microsoft Exchange 2000 efficiently depends on first setting up Windows 2000 Server and Active Directory suitably for both your network configuration and your messaging and collaboration requirements. The payoff comes in overall planning and design costs, which can be reduced by using a single directory of all networking and messaging resources.

- **Management**: Systems administrators can manage both Microsoft Exchange 2000 and Windows 2000 Server using a common set of management interfaces, making it simpler to learn and perform management tasks. Further, policy-based administration simplifies tasks such as changing a set of user mailboxes all at once.

- **Security**: Microsoft Exchange 2000 uses the Windows 2000 Server security services. This provides a comprehensive set of security and permission services, such as standardized methods for encrypting e-mail, and simplifies system management by letting administrators use common security controls across both Exchange and Windows 2000 Server.

- **Internet**: Microsoft Exchange 2000 Server is tightly integrated with the Windows 2000 Internet Information Service. In addition to helping remove the boundaries between traditional e-mail and Internet-based communications, this gives developers a simpler way to build collaboration solutions.

- **Scalability/Availability**: The improved clustering support in Microsoft Exchange 2000 takes advantage of the Windows 2000 Advanced Server

Cluster service. This ensures the highest level of system reliability, reduces user interruptions, and decreases hardware costs.

## 2.10 Review on Web Server

On the Web, everything depends on the server. The ability of the customer to find the application web site depends on the server. The security of the transactions and the integrity of the data depend on the web server.

In this new transaction-based world, server choice is critical. Consider that in the E-Commerce world, a server that goes down is a customer service problem. A server that isn't secure can permanently destroy consumer confidence.

A server that can't scale up to increased traffic on a web site can end up turning profitable customers away and then cause major disruption when it is scaled up.

### 2.10.1 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.

### 2.10.2 Microsoft Internet Information Server (IIS) 4.0

Microsoft Internet Information Server (IIS) 4.0 is the latest addition to Microsoft's BackOffice family of products that enables the publishing and distributing of information on the Internet. In other words, it is a web server and integrated into Windows NT server. IIS is optimized for Windows NT Server, and takes advantage of the security, which NT Server provides.

IIS provides the ability to provide web services, not only for web pages, but also ftp sites, nntp services and video and audio services. It integrates with the database facilities of SQL Server, the management facilities of Site Server, and e-mailing and messaging abilities of Exchange.

The configuration and management of Internet services that can be done through the Internet service manager. Any services in any of the servers can be started and stopped as well as configuration and management properties such as access permissions and logon requirements for clients can be employed and managed. IIS provides some basic and enhanced security. IIS requires a valid username and password for authentication in order to gain access to the information. The usernames must be valid on the computer running IIS, or from a Windows NT domain accessible from the IIS machine. It also offers controlling access via an IP address.

### 2.10.3 Microsoft Internet Information Server (IIS) 5.0

Internet Information Services (IIS), called IIS 5.0. Internet Information Services runs as an enterprise service within Windows 2000. It uses other services provided by Windows 2000, such as security and the Active Directory service. This version improves the Web server's reliability, performance, management, security, and application services. Many of these improvements result from the way IIS 5.0 incorporates new operating system features provided in Windows 2000.

IIS 4.0 focus on security, administration, programmability, and support for Internet standards while IIS 5.0 builds on the features and capabilities needed to deliver Web sites required in an increasingly Internet-centric business environment. And it makes it even easier to use the technologies delivered in prior versions. In particular, IIS 5.0 features improvements in the following major areas:

- **Reliability and performance.** A number of features make IIS more reliable and better performing. To make it faster and easier to restart IIS, the reliable

restart feature of IIS 5.0 allows an administrator to restart Web services without rebooting the computer. To improve reliability, Application Protection provides the ability to run applications in a pool, separate from the Web services. The new CPU Throttling and Socket Pooling features in IIS 5.0 can also improve reliability. For application developers, Web site performance can be improved through new features such as scriptless Microsoft Active Server Pages (ASP) processing, ASP self-tuning, and performance-enhanced ASP objects.

- **Management.** IIS 5.0 is easier to install and maintain. A number of features support this increased ease-of-maintenance, including a simplified installation process, new security task wizards, the ability to account for time used by processes, more flexible remote administration, and the ability to create custom error messages.
- **Security.** IIS 5.0 adds support for important industry-standard security protocols, including Digest Authentication, Server Gated Cryptography, Kerberos V5 authentication protocol, Transport Layer Security, and Fortezza. In addition, three new task wizards make it easier for administrators to manage a site's security settings.

Application environment. Developers will find that IIS 5.0 expands the Web server's application development environment by building on new technologies included in Windows 2000 Server. These include Active Directory and the expanded Component Object Model (COM+). In addition, enhancements to IIS Active Server Pages, such as scriptless ASP processing, as well as improved flow control and error handling, let developers write more efficient Web-centric applications.

## 2.11 Review on Platform and Operating System

### 2.11.1 Microsoft Windows NT Server 4.0

Microsoft Windows NT server 4.0 is a complete and powerful platform that provides server operating system. Besides that it is a server that offers the building, hosting and integration of web-based applications. Windows NT Server 4.0 was designed to assist developers in constructing, building and deploying different types of business applications in a fast way. Moreover, the Option Pack allowed the integration of web functionality, transaction processing, language scripting, component deployment, and message queuing services directly into Widows NT Server 4.0. The introduction of new management tools in Windows NT Server 4.0 and Option Pack has provided great assistance in setting up web sites, managing the content, and analyzing usage patterns for improvement. Besides that, with its innovative web publishing features, customizable tools, and new wizard technologies, Windows NT Server 4.0 serves as a platform to publish and share information in a secure way over Internet and intranets.

### 2.11.2 Microsoft Windows 2000 Server

Windows 2000 Server is the entry-level version of the server family. The multipurpose network operating system for businesses 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.

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 easer 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.



### 2.11.3 Microsoft Windows 2000 Advanced Server

The Windows 2000 Advanced Server operating system contains all the functionality and reliability of the standard version of Windows 2000 Server, plus additional features for applications that require higher levels of scalability and availability. Windows 2000 Advanced Server delivers enhanced reliability, availability, and scalability. It scales from 1 to 8 processors and up to 8 gigabytes, and provides enhanced reliability and availability, with two-node clustering and 32-node network load balancing.

Advanced Server helps ensure your systems are available by addressing the causes of both planned and unplanned network and server downtime. It also has features that let your applications grow to support large numbers of users and data.

The major functionality that Advanced Server adds to Windows 2000 Server is support for extremely high-performance servers and the ability to link servers

together to handle larger loads. These capabilities provide a reliable foundation that lets you make sure your systems are available regardless of a system failure or how large an application becomes.

Advanced Server lets you deploy highly available and scalable applications on industry-standard PC hardware. Common examples of applications suitable for Advanced Server include database, messaging, and file and print servers. No matter the size of your organization, Advanced Server is a good choice for server applications that must be available at all times. It also gives growing organizations the ability to ensure the availability of critical applications while being able to scale those applications both up and out to meet increased demand.

## 2.12 Review on Software Development Methodologies

In the 1960s, the problems of developing software were not clearly understood, but it was realized that a software crisis existed. The term software engineering was coined to refer to the management and technical methods, procedures, and tools required to develop a large-scale software system effectively. With the application of software engineering concepts, many large-scale software systems have been developed using a software life cycle [10]. Below are the reviews of some of the most widely used software development methodologies.

### 2.12.1 Waterfall model

According to Boehm, the Waterfall model of software development is the first lifecycle developed in 1970 by W. W. Royce to establish a disciplined approach for software development, and it was derived from other engineering processes. It was considered a superior method to the "code-and-fix" practices previously used. It is called the Waterfall Model because of its cascading nature, from one step to the next [11].

Waterfall's simplistic nature makes it one of the most used lifecycle models in business today. In the waterfall method, there is a distinct boundary between systems engineering, analysis, design, code, testing, and maintenance. The output of one stage becomes the input for the next stage; for example, the output of analysis becomes the input for design. Each of the five steps must be completed before you can go on fully to the next - although there is, many times, overlap between each of the steps. At each step, testing is done. If problems are found, the developers either put in necessary corrections, or go back to the previous step. In this classic approach, the requirements are all defined in detail and cast in stone.

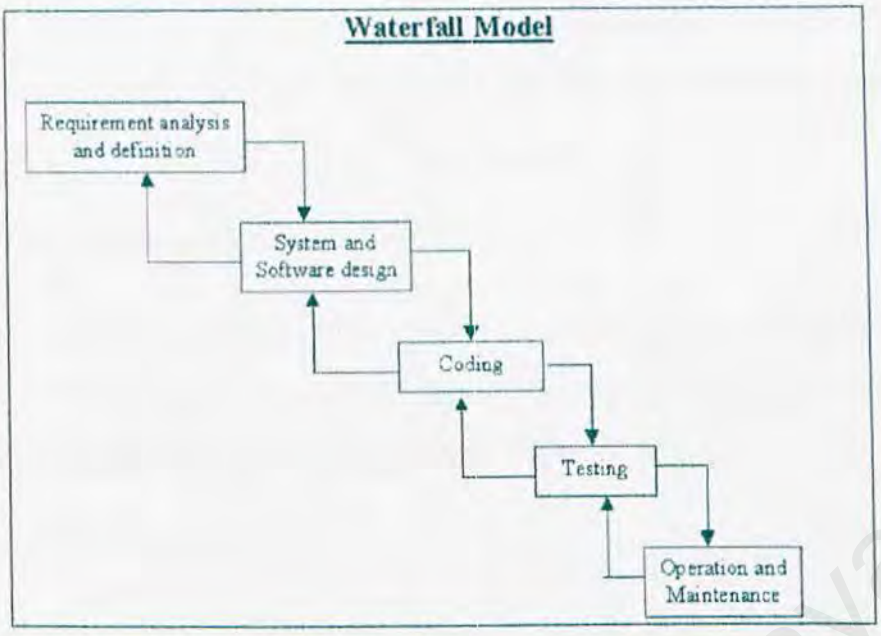


Figure 2.1: The "Waterfall" model

**Step 1: Requirements Definition**

The overall system is designed in this phase:- services, constraints, goals, time frames, and others are all put into existence.

**Step 2: System and Software Design**

In this step, the system architecture is established. Constraints are refined with the actual system architecture constraints. Services are refined into pseudeocode for a particular language.

**Step 3: Implementation and Unit Testing**

The system is still a set of programs, not integrated into a whole fleshed out product. Each unit is tested to make sure it performs to specifications. Documentation is done, often at this stage for the individual units.

**Step 4: Integration and System Testing**

All of the separate programs, or units, are put together and flesh out as a system, or finished product. The overall system is tested to make sure the finished product works as well as the individual units had before. After testing, the finished product is

made available to the clients or customers. More documentation is written here, such as system documentation rather than individual unit documentation. Installation manuals, help files for overall system usage, and such.

**Step 5: Integration and System Testing**

It includes possible installation of the system, training end users, and supporting the product over its lifetime. Because of the support this stage offers, it can lead back to any of the previous stages in the cycle when are encountered.

Advantages of "Waterfall" model :-

- It flows well, and is easy to understand from a broad viewpoint.
- It is easy to implement.
- Flaws are easy to find, as they are flushed out as every stage.
- Product is well documented.
- High Visibility.
- Well known and well used.

### 2.12.2 Spiral model

The spiral model was developed by Dr. Barry Boehm in 1987 as a risk-reducing approach to software development. The spiral model encompasses features of the phased lifecycle as well as the prototype lifecycle. However, unlike those lifecycles, the spiral model uses risk analysis as one of its elements. It also uses the waterfall model for each step so as to avoid any risks. The spiral model overcomes major sources of project risk with the Risk Management Plan. This also helps in being more compatible with other models [12].

However, the spiral model is not as sophisticated and elaborate as some of the other lifecycle models. It needs further planning and development in such areas as contracting, specifications, milestones, reviews, scheduling, status monitoring, and risk area identification.

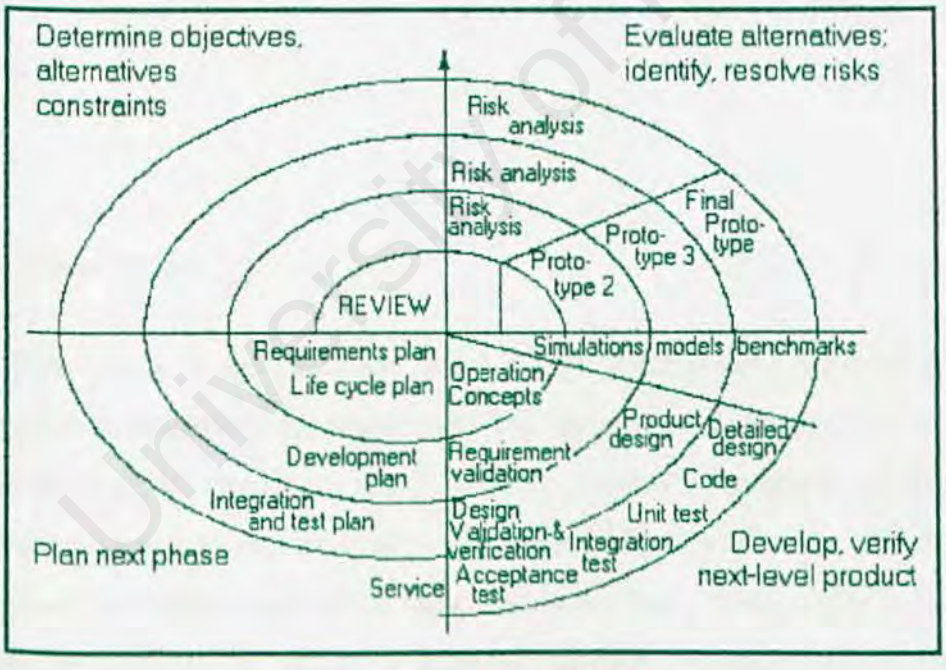


Figure 2.2: Spiral Model

The radial dimension in the figure above shows the cumulative cost of the accomplishment of the steps at that time. The angular dimension shows the progress made in the completion of each cycle of the spiral. Every cycle of the model represents the same sequence of steps in the completion of the software. The spiral model includes four major elements, which include:

- Planning- Determination of objectives, alternatives, and constraints.
- Risk Analysis- Analyzes alternatives and attempts to identify and resolve the risks involved.
- Engineering- Development of the product as well as the incorporation of testing.
- Customer Evaluation- Assessment of the products of the engineering element.

An important feature of the spiral model is that each cycle is completed with a review by the people concerned with the project (designers and programmers). This review consists of a review of all the products developed up to that point and includes the plans for the next cycle. These plans may include a partition of the product into smaller portions for development or components that are implemented by individual organizations or persons. If the plan for the development fails, then the spiral is terminated. Otherwise it terminates with the initiation of new or modified software.

### 2.12.3 Prototyping

Prototyping is an approach based on an evolutionary view of software development and having an impact on the development process as a whole. Prototyping involves producing early working versions (prototype) of the future application system and experimenting with them [13]. Specification, development, and validation activities occur at the same time, since rapid development is important for prototype systems. To deliver a prototype quickly, you may have to leave out some system functionality or relax non-functional factors such as response speed and reliability.

There are two types of prototyping, which represent two different objectives and two different types of implementation :-

#### 2.12.3.1 Evolutionary Prototyping

The objective of evolutionary prototyping is to deliver a working system to end-users. Evolutionary prototyping starts with the parts of the system, which are clearly understood by the customer/end-user. The system is continually developed by adding new features/requirements as they are discovered or proposed by the customer, until a final system is delivered. The evolutionary approach helps anticipate how end-users will use new software systems. The user is given a system, which is unfinished, then the system is modified and fine tuned as the user requirements become clear.

#### 2.12.3.2 Throw-Away Prototyping

The objective of throw-away prototyping is to validate or determine the system requirements. By understanding the customer's requirements, a better requirements definition for the system can be developed. Throw-away prototyping focuses on undefined or unclear portions of the requirements. During implementation, the parts of the system, which are not understood, are developed first. Throw-away prototyping is intended to determine the system specification so that the end-product of the prototype development phase is that specification.

A fixed decision is made to build a throw-away prototype to help requirements analysis and validation. After evaluation, the prototype is 'thrown-away' and a production-quality system is built. Throw-away prototyping extends the requirements analysis process with the intention of reducing overall life cycle costs. The principal function of the prototype is to clarify requirements and provide additional information for managers to assess process risks. Components from the prototype may be reused in the production-quality system.

# Chapter 3 SYSTEM ANALYSIS

## 3.1 Techniques Used To Defined Requirements

System analysis is carried out when the user's requirement and needs are to be identified. Besides that, the feasibility of the system needs to be identified. Technical analysis of the system will assist in allocating the resources for each functional and non-functional requirement.

Techniques used to define requirements:

### 3.1.1 Interview

Interview activities have been carried out in order to understand the loopholes and weakness of the current meeting scheduling process. Besides that, it has also enriched the knowledge of how meeting is being schedule in the real environment and idea on how to improve the current process.

### 3.1.2 Internet Research

Internet is used as the main resource for referring any ambiguities that arise during the entire development period. Web sites of the developers have assisted in broadening the methods and approach used in tackling every problem.

## 3.2 Meeting Scheduling in FSKTM

### 3.2.1 Current Scheduling Process

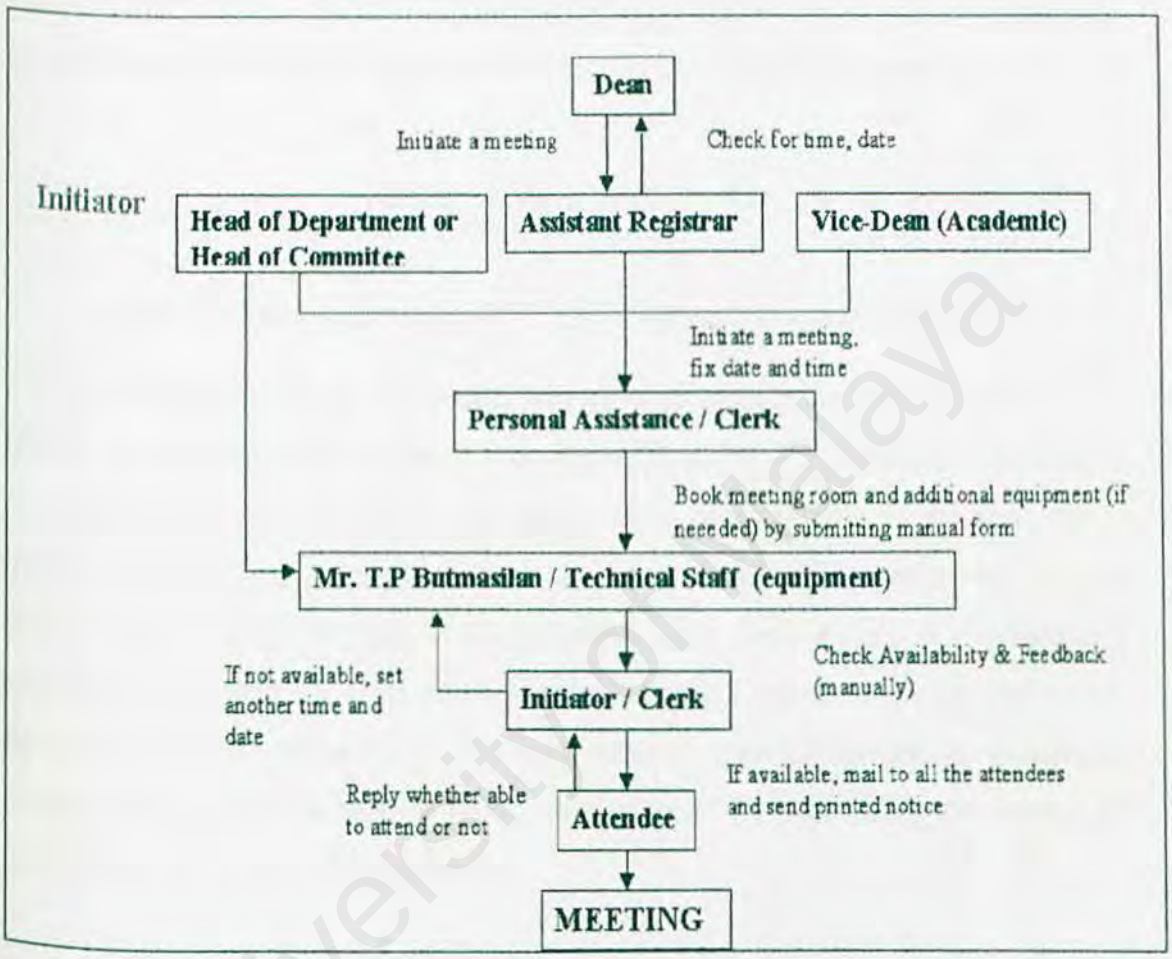


Figure 3.1 Flowchart of the current meeting scheduling process in FSKTM, UM

Most of the meeting is initiate by the Dean as the Dean is the chairman to the Faculty Meeting and to most the committee in the Faculty of Computer Science and Information Technology in University of Malaya such as Committee of Exam Board, Committee of Higher Education and others. Usually, the Assistant Registrar will schedule and initiate the meeting on behalf of the Dean with the help of his or her personal assistant. The Vice-Dean (Academic) and Head of Department or head of certain committee also initiate a meeting, either by themselves or with the help of their personal assistant. The initiator or the clerk will then submit booking form or check with Mr. T.P. Butmasilan on the availability of the meeting room on the

proposed date and time. If the time slot is unavailable, the clerk or initiator needs to resubmit form. This is a very tedious work. Then, if additional equipment is needed, clerk need to inform the technical staff separately. The clerk needs to print notice to hand out to the lecturer and at the same time send mail to them. The clerk needs to scan through the reply mail one by one when the lecturer reply to check whether he or she is attending the meeting or not. This is very slow and unproductive.

### 3.2.2 Major Problems Encountered

**1. VERY SLOW AND TEDIOUS TO SCHEDULE A MEETING**

This is because to schedule a meeting, several steps need to be done separately. The initiator or the clerk need to submit form manually to Mr. T.P Butmasilan and this is extremely troublesome for those who reside in the old building. If the time slot is unavailable, the clerk or initiator needs to resubmit form. This is a very tedious work. Then, if additional equipment is needed, clerk need to inform the technical staff separately. Next, the clerk needs to print notice to hand out to the lecturer and at the same time send mail to them. The clerk needs to scan through the reply mail one by one when the lecturer reply to check whether he or she is attending the meeting or not. This is very slow and unproductive.

**2. Problems in booking for meeting room**

As there is no timetable for the room booking, there is problem of booking of meeting room, which fit the number of attendees and equipment needed. This often results in meeting of less priority will have to give way to meeting which has higher priority.

**3. Possibility of a person being invited to two or more meetings is quite high**

As there is no central repository of information holdings where information could be better shared and reused from different locations. The initiator did not know whether all the attendees have meeting on the same time or not. These leads to possibility of a person being invited to two or more meetings are quite high

**4. Possibility of invitees is not notify**

There is a possibilities that the invitees did not know that there is a meeting. This is maybe due to the invitees didn't check mail and did not receive the notice.

**5. Paper does not function as an efficient media.**

As it does not provide easy updates methods and the information contained on a piece of paper does not relate to information contained on a piece of paper does not relate to information on other pieces of paper in a direct way. The current meeting room booking system is based on submitting from manually, as well as distribution of minutes. Moreover, paper is very difficult to keep as it takes away a lot of spaces. So, it can be said that printed documents are easily misplaced.

**6. Not systematic**

There is no guideline on how to schedule a meeting. From the interview, I found out that all the clerk initiate meeting base on their preference way. There is a major difference of way to initiate a meeting between those staff resides in old and new building.

**7. Invitees forget to attend meetings**

As there is no systematic way to schedule a meeting, so there are some initiator who will send reminder and some who did not do so.

## 3.3 Proposed Meeting Scheduling Process Using MSS

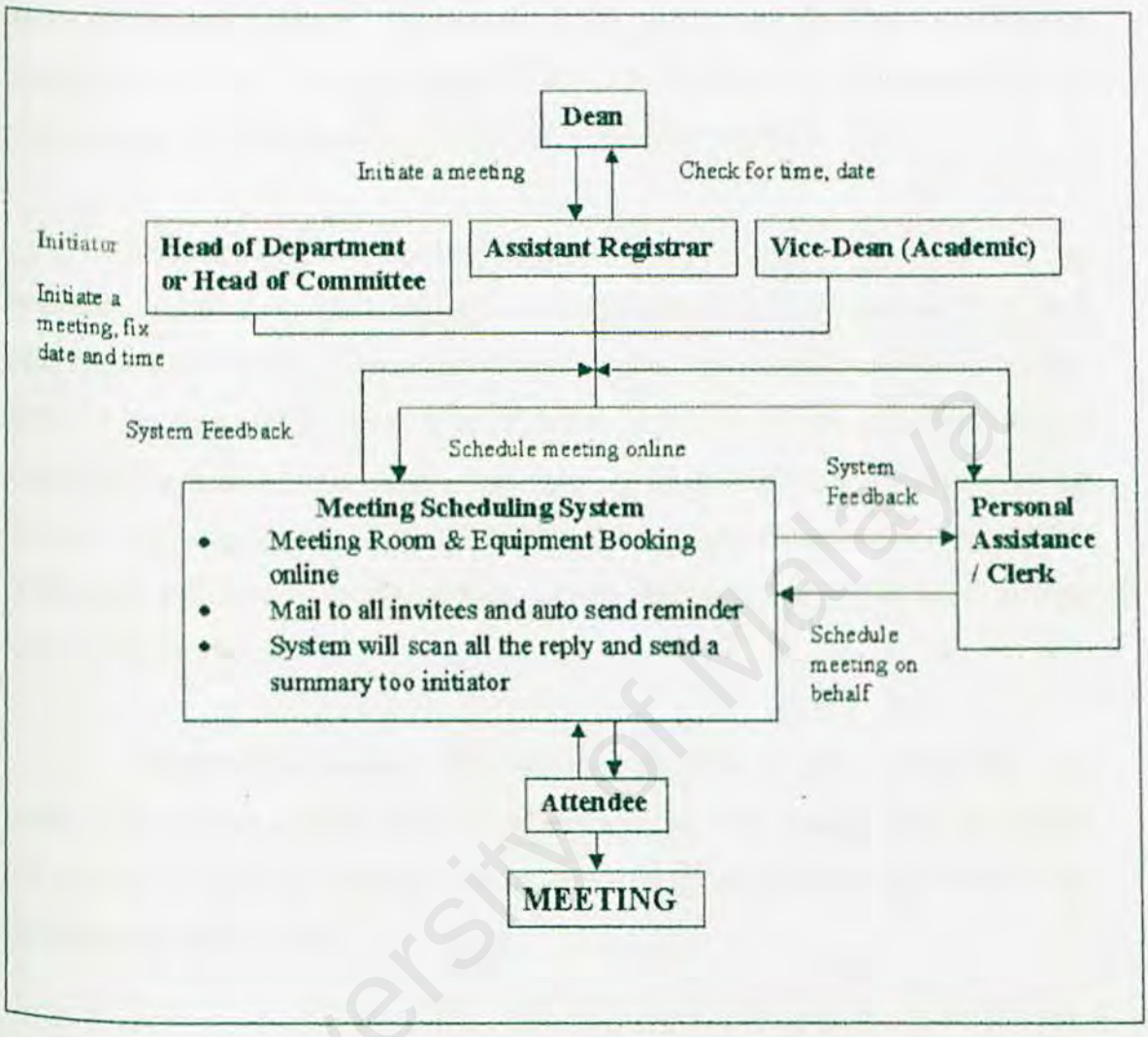


Figure 3.2: Flowchart of the proposed meeting scheduling process using MSS

The proposed meeting scheduling process is to integrate meeting scheduling and booking of meeting resources (rooms and equipment). It is to provide an easy to use environment in meeting scheduling so that meeting can be schedule effectively and systematically. Its easy to use feature enables initiator to schedule meeting by their own without the help of personal assistants.

To schedule a meeting, initiator or clerk need to check the availability of the invitees and resources based on a desire date and time. The status of the invitees is then check by the system whether there is clash in time, which is whether the invitees are scheduled to attend other meeting. Initiator can see the list of invitees

who are tied up with other meetings with the respective initiator and details. This will help them to decide and plan a better schedule or to have negotiation when the desirable time-slot is taken. If the resources are booked, the meeting room booking details can be view. This is to help the initiator to plan a better schedule and to enable initiator to have negotiation when the desirable time-slot is taken.

When the desirable meeting time is determined, initiator will send meeting invitation to the invitee and resource booking request to both meeting room and equipment administrator. When the invitee receives the e-mail notification, invitee needs to log on to MSS system to reply to the invitation. The same goes to meeting room and equipment administrator, that is to log on to MSS system to approve the resource booking request. When the administrator approved the request, e-mail notification will be sent to the initiator. Beside that, user can opt to book meeting room or equipment only.

Initiator of the meeting can always log on to the system to check the reply status of the invitees and the status of the resource booking request. One day before the meeting, a reminder message will be sent through e-mail to all the invitees and the initiator of the meeting.

## 3.4 Meeting Scheduling Systems Objectives

1. **To schedule meeting and booking of meeting resources (rooms and equipment) effectively and systematically.**

Meeting Scheduling System (MSS) is to support the organization of meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and number of attendees that can attend the meeting, providing a systematic way to initiate a meeting. MSS includes modules like Personal Scheduler, Meeting Room Booking, Equipment Booking and Meeting Scheduling.

2. **To solve conflicts in meeting scheduling process.**

It is also to avoid the possibility of scheduling a person to attend two or more meetings at one time. Besides that, on booking of meeting resources (meeting room and equipment), user can check the availability of the resources.

3. **Reducing faculty administrators' workload and redundant jobs in respect of scheduling meetings and clerk's workload in distributing memos.**

Integrate all the previously separated manual task into an integrated system. Other than that, initiator needs not to read through all the reply. MSS will provide an agent to scan through all the reply concerning a particular meeting and provide a summary to the initiator. This will speed up meeting scheduling process. The productivity of clerk can be fully optimized with types of task other than concentrating on distributing and retrieving documents, as the document will be transferred electronically through the e-mail system and through collaboration functions.

4. **To provide a communication and collaboration feature for the entire environment in meeting scheduling**

The communication of MSS includes a set of fully integrated communication services, which allows users to utilize the communication technology such as the email that will enable administrators, clerks and lecturers to communicate easily and work together effectively. The overall delivery time can be reduced, as the document will be transferred electronically through the email system and through collaboration

functions. The collaboration of MSS involves a system of electronic workgroup computing which meets the needs for highly structure information flow. This will update the information routing in a more structured and flexible way and information sharing.

**5. To be able to access, use and maintain at user convenience**

Meeting Scheduling System (MSS) is an Internet web based application. These allows the initiator to initiate or cancel meeting, invitees to check their schedule and send notice of unavailability and Administrator to process and maintain the MSS from any location at any time and their convenience.

## 3.5 Meeting Scheduling Systems Terms Definition

Meeting Scheduling System has its own associated terminology. The following is a list of the terms used in this project in relation to meeting scheduling :-

1. **System Administrator**
   The administrator of the entire Meeting Scheduling System (MSS).

2. **Meeting Room Administrator**

   The Meeting Room administrator that is responsible for Meeting Room maintenance and booking.

3. **Equipment Administrator**

   The Equipment administrator that is responsible for Equipment maintenance and booking.

4. **Initiator**

   The person who initiate the meeting and book the necessary resources for the meeting.

5. **Invitees**

   Invitees are those people being invited to the meeting.

6. **Meeting Status**

   - Confirmed – invitee have confirmed to attend the meeting
   - Pending – invitee have not reply to the meeting invitation
   - Reject – invitee have reject the meeting invitation
   - Cancelled - indicate that the meeting has been cancelled by the initiator or administrator
   - Amend - indicate that the meeting has been amended by the initiator

7. **Resource (Meeting Room and Equipment) Booking Status**

- Approved – the booking have been approved by the administrator
- Pending – the booking waiting for approval
- Amended - indicate that the booking has been amended by the requester
- Cancelled – indicate that the booking has been cancelled by the requester or administrator

## 3.6 Software Development Methodology Being Employed

The software process is the set of activities and associated results, which produce a software product. Different software processes decompose these activities in different ways. If the wrong way process is used, this will probably reduce the quality or the software product to developed.

### 3.6.1 The "Waterfall" Model

After much reviews on the software development methodologies in section 2.12, the software development process that is being chosen to be based on is the "Waterfall Model". The waterfall model is a systematic sequential approach to software development model after a conventional engineering cycle. One phase is completed before the next phase is entered.

### 3.6.2 Appropriateness of "Waterfall" Model To The Project

"Waterfall " model is chosen in this project because it is appropriate to this project based on the reasons listed below :-

- The project is relatively short (less than one year)
- There is a good specification to begin with
- The scope of the project is well understood
- There is a defined change control procedure
- The project risks have been accessed and are considered to be low
- The users do involved in the project.
- End user requires the entire system at one time.

## 3.7 Tools being chosen in Meeting Scheduling Systems

### 3.7.1 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.

Microsoft Visual Basic version 6.0 is used in writing dynamic link library (dll) that is used to generate e-mail messages.

### 3.7.2 Database server

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.

### 3.7.3 Web server

Given that vast different web server, it is decided that this project will use IIS 5.0 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.

### 3.7.4 Communication and Collaboration Server

Microsoft Exchange 2000 Server is chosen as the mail server, as it is used for system emailing, scheduling, calendar and meeting management. It consists of Collaboration Data Objects, which are used for providing the above functions and support.

### 3.7.5 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.

## 3.8 Systems Requirements

A requirement is a feature of the system or description of something the system is capable of doing in order to fulfill the system's purposes [1].

### 3.8.1 Functional Requirements

A functional requirement describes an interaction between the system and its environment [1].

#### 3.8.1.1 System Administrator Section

##### 3.8.1.1.1 Authentication and Authorization Module

It is vital to protect MSS database and mail messages kept in the mail server from the non-authorized user. To access MSS, a NT login is required. Based on user log in, access level will be set.

##### 3.8.1.1.2 User Setting Module

To allow user to maintain user profile and setting on Windows 2000 Active Directory.

#### 3.8.1.1.3 Maintenance Module

1. Meeting Room Maintenance

To allow administrator to maintain meeting room details such as number of seats and status from time to time.

2. Equipment Maintenance

To allow administrator to maintain equipment details and status from time to time.

#### 3.8.1.1.4 Cancellation Module

1. Meeting Request

To allow administrator to cancel meeting request initiated. E-mail notification will be sent to initiator of the meeting.

2. Meeting Room Booking

To allow administrator to cancel meeting room booking request. E-mail notification will be sent to requester.

3. Equipment Booking

To allow administrator to cancel equipment booking request. E-mail notification will be sent to requester.

#### 3.8.1.1.5 Report Module

1. Meeting Request

To provide status report on meeting being initiated. User can filter report by date range and status of the meeting.

2. Meeting Room Booking

To provide status report on meeting room being booked. User can filter report by date range and status of the meeting room booking.

3. Equipment Booking

To provide status report on equipment being booked. User can filter report by date range and status of the equipment booking.

#### 3.8.1.2 Meeting Room Administrator Section

##### 3.8.1.2.1 Authentication and Authorization Module

It is vital to protect MSS database and mail messages kept in the mail server from the non-authorized user. To access MSS, a NT login is required. Based on user log in, access level will be set.

##### 3.8.1.2.2 Maintenance Module

To allow meeting room administrator to maintain meeting room details such as number of seats and status from time to time.

##### 3.8.1.2.3 Approval Module

To allow meeting room administrator to approve meeting room booking request. E-mail notification will be sent to requester when the request is approved.

##### 3.8.1.2.4 Cancellation Module

To allow administrator to cancel meeting room booking request. E-mail notification will be sent to requester.

##### 3.8.1.2.5 Report Module

To provide status report on meeting room being booked. User can filter report by date range and status of the meeting room booking.

#### 3.8.1.3 Equipment Administrator Section

##### 3.8.1.3.1 Authentication and Authorization Module

It is vital to protect MSS database and mail messages kept in the mail server from the non-authorized user. To access MSS, a NT login is required. Based on user log in, access level will be set.

##### 3.8.1.3.2 Maintenance Module

To allow administrator to maintain equipment details and status from time to time.

##### 3.8.1.3.3 Approval Module

To allow equipment administrator to approve equipment booking request. E-mail notification will be sent to requester when the request is approved.

##### 3.8.1.3.4 Cancellation Module

To allow administrator to cancel equipment booking request. E-mail notification will be sent to requester.

##### 3.8.1.3.5 Report Module

To provide status report on equipment being booked. User can filter report by date range and status of the equipment booking.

#### 3.8.1.4 User Section

##### 3.8.1.4.1 AUTHENTICATION AND AUTHORIZATION MODULE

Is vital to protect MSS database and mail messages kept in the mail server from the non-authorized user. To access MSS, a NT login is required. Access level will be set based on user log in.

#### 3.8.1.4.2 PERSONAL SCHEDULER MODULE

The Personal Scheduler displays user's schedule for a selected period and allows user to filter the status of the meeting. User can view the meeting details that the user are scheduled to attend.

#### 3.8.1.4.3 NEW REQUEST MODULE

1. Initiate Meeting

To allow initiator to check invitee, meeting room, equipment availability. If the invitees already are being scheduled to attend other meeting, initiator can view the list of invitees who are tied up with other meetings with the respective initiator and details. This will help them to decide and plan a better schedule or to have negotiation if necessary with the respective initiator when the desirable time-slot is taken. This is the same situation with meeting room and equipment.

After a suitable time is fixed, initiator can book meeting room and equipment also. E-mail notification will then be sent to each invitee, meeting room administrator and equipment administrator (if equipment is booked).

2. Book Meeting Room Only

To allow requester to check meeting room availability. If the meeting room is already being occupied, requester can view the list of booking with the respective requester and details. This will help them to decide and plan a better schedule or to have negotiation if necessary with the respective requester when the desirable time-slot is taken.

After a suitable time is fix, requester can book then book the meeting room. E-mail notification will then be sent to meeting room administrator for approval.

3. Book Equipment Only

To allow requester to check equipment availability. If the equipment is already being occupied, requester can view the list of booking with the respective requester and details. This will help them to decide and plan a better schedule or to have negotiation if necessary with the respective requester when the desirable time-slot is taken.

After a suitable time is fix, requester can book then book the equipment. E-mail notification will then be sent to equipment administrator for approval.

#### 3.8.1.4.4 Incoming Request Module

1. Reply Meeting Invitation

Invitee will receive e-mail notification when are invited to attend meeting. Invitee can view meeting request and the details of the meeting. This module is to enable invitee to reply to the meeting invitation.

2. Emergency Withdrawal

To allow invitee to withdraw from the meeting if there is emergency matter, after the invitee has confirmed attendance to the meeting,

#### 3.8.1.4.5 Cancellation Module

1. Meeting Request

To allow meeting initiator to cancel meeting request initiated. E-mail notification will be sent to all the invitee of the meeting.

2. Meeting Room Booking

To allow requester to cancel meeting room booking request. E-mail notification will be sent to meeting room administrator.

3. Equipment Booking

To allow requester to cancel equipment booking request. E-mail notification will be sent to equipment administrator.

#### 3.8.1.4.6 Amendment Module

1. Meeting Request

To allow meeting initiator to amend meeting request initiated. E-mail notification will be sent to all the invitee of the meeting.

2. Meeting Room Booking

To allow requester to amend meeting room booking request. E-mail notification will be sent to meeting room administrator.

3. Equipment Booking

To allow requester to amend equipment booking request. E-mail notification will be sent to equipment administrator.

#### 3.8.1.4.7 Report Module

1. Meeting Request

To provide status report on meeting being initiated. Initiator can filter report by date range and status of the meeting. Initiator can view the reply status of the invitee on the meeting request.

2. Meeting Room Booking

To provide status report on meeting room being booked. User can filter report by date range and status of the meeting room booking.

3. Equipment Booking

To provide status report on equipment being booked. User can filter report by date range and status of the equipment booking.

#### 3.8.1.4.8 Reminder Module

One day before meeting, a reminder message will be sent through e-mail to all the invitees and also the initiator of the meeting.

### 3.8.2 Non-Functional Requirements

A non-functional requirements or constraint describes a restriction on the system that limits our choice for constructing a solution to the problem [1]. Non-functional requirements are essential definition of system properties and constraints under which a system must operate.

#### 3.8.2.1 Reliability

The system shall be reliable in performing its functions and operation and shall not cause unnecessary unplanned downtime. Besides that, does not produce dangerous or costly failures when it is used in a reasonable manner, that is, in a manner that a typical user expects is normal. This definition recognizes that a system may not always be used in the ways that the designer expects.

#### 3.8.2.2 Usability

The system shall be easy to use. Interface shall be intuitive and consistent with other modules in the system. The usage of suitable and meaningful captions and icons will help the user to use the system with more confidence. The system should display a confirmation message for any non-trivial process such as deletion, and error message if an error occurs, such as invalid data input.

#### 3.8.2.3 Robustness

Robustness refers to quality that causes a system to be able to handle, or at least avoid minor disaster in the face of unexpected circumstances such as improper input data or power failure. When the system is restarted after abnormal server shutdown, the transaction log file is scanned for incomplete transactions. If one is found, the transaction rolls back to the state it was before the incomplete transaction took place.

#### 3.8.2.4 Security

The system should be equipped with sufficient security. Each access by user should be authenticated and validated by the system. The system should not show any potential of leakage of information especially through the mail server.

#### 3.8.2.5 Maintainability

Maintainability is the degree to which architectural, data or procedural design can be extended. MSS is designed to be expandable in the future.

## 3.9 Expected Outcome

At the end of the project, it is expected that the Meeting Scheduling System (MSS) will be :-

- Able to support the organization of meetings, that is for each meeting request, a suitable meeting date can be determine based on constraints like meeting location, equipment and number of attendees that can attend the meeting.

- It is also to avoid the possibility of scheduling a person to attend two or more meetings at one time and to allocate resources efficiently so that it most benefits the organization by helping to keep track of availability of staff, maintain meeting room and equipment schedules and more on user computer desktop.

- Meeting Scheduling System allow users to check if all the lecturers are available to attend a meeting, and will automatically inform participants of details about meetings that involve them.

# Chapter 4 SYSTEM DESIGN

System design is the nucleus of the software development process and is applied regardless of the development model or standard that is used.

## 4.1 System Functionality Design

### 4.1.1 System Architecture

The architecture is the kind and number of servers involved.

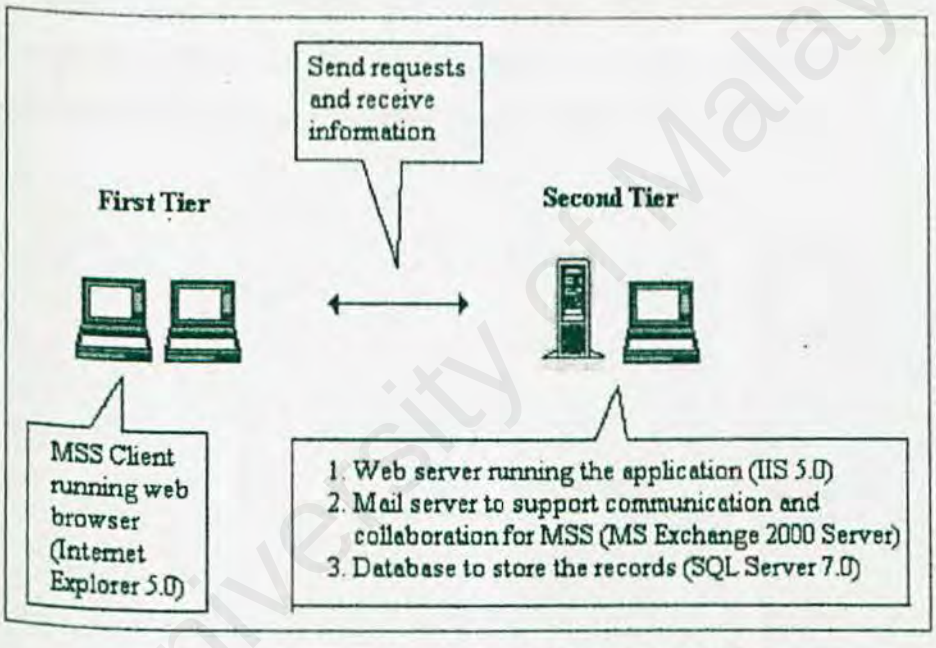


Figure 4.1: System architecture of MSS

The system will base on 2-tier client/server architecture. It consists of a client tier and server tier (with rules that are applied and kept within a web server and the Microsoft Exchange and Microsoft SQL Server).

The client tier shall be the Web Browser (IE 5.0) which will provides the user interface. Client shall act as the first tier to access and use the system from any location within any time frame.

The application shall reside in the second tier that is within a Web Server (IIS 5.0) that will process the request from the client. The web server will first authenticate the user NT login. The processing is done on the second tier as well as update of data in to the database. Microsoft Exchange 2000 Server will serve as the mail server to send mail. Data records are kept in the SQL server and mails on the Exchange Server.

### 4.1.2 System Context Diagram

Figure 4.2 is the context diagram for Meeting Scheduling System. There are five external entities for this system, namely System Administrator, Meeting Room Administrator, Equipment Administrator, Initiator and Invitee.

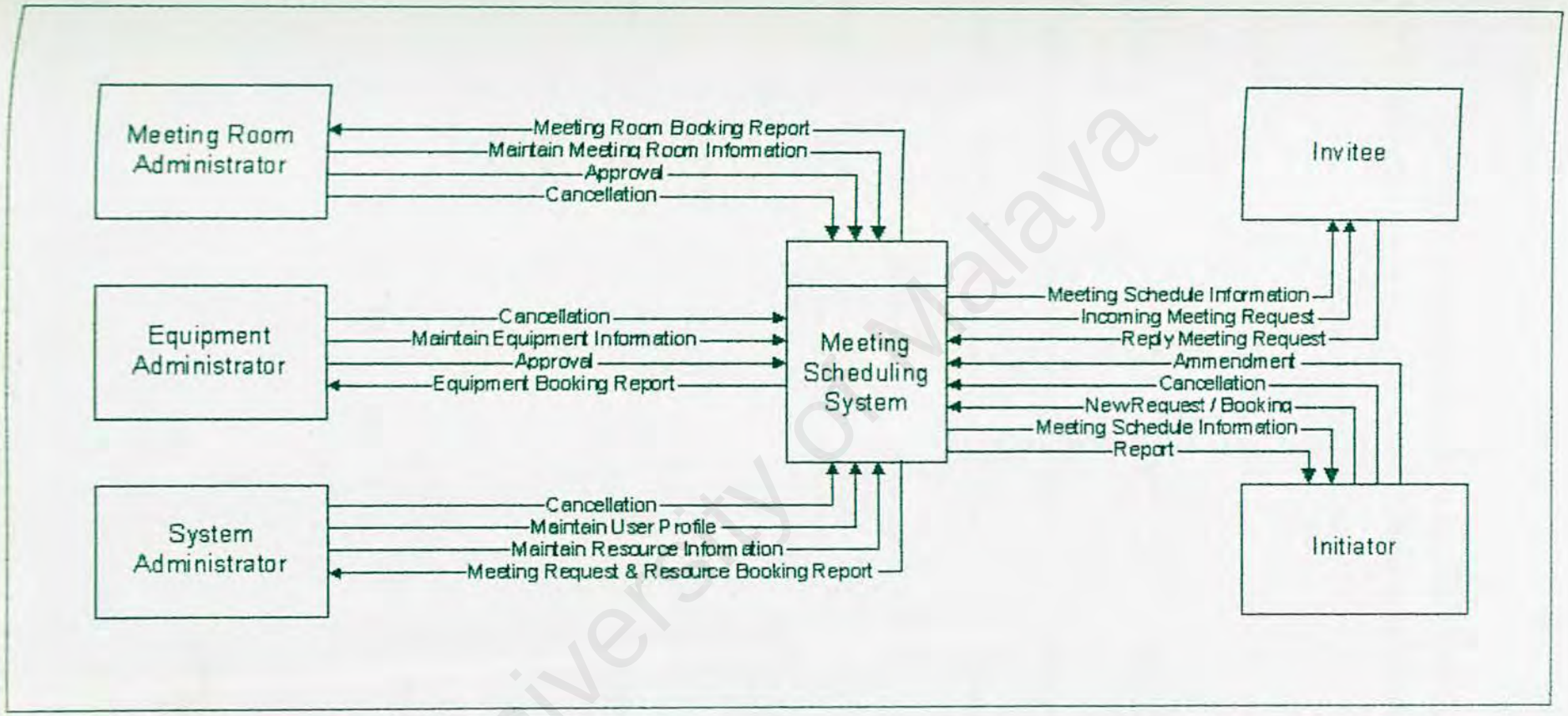


Figure 4.2: MSS Context Diagram

### 4.1.3 System Structure Chart

The following structure chart shows the hierarchical representation between the sections, modules and sub-modules in the Meeting Scheduling System.

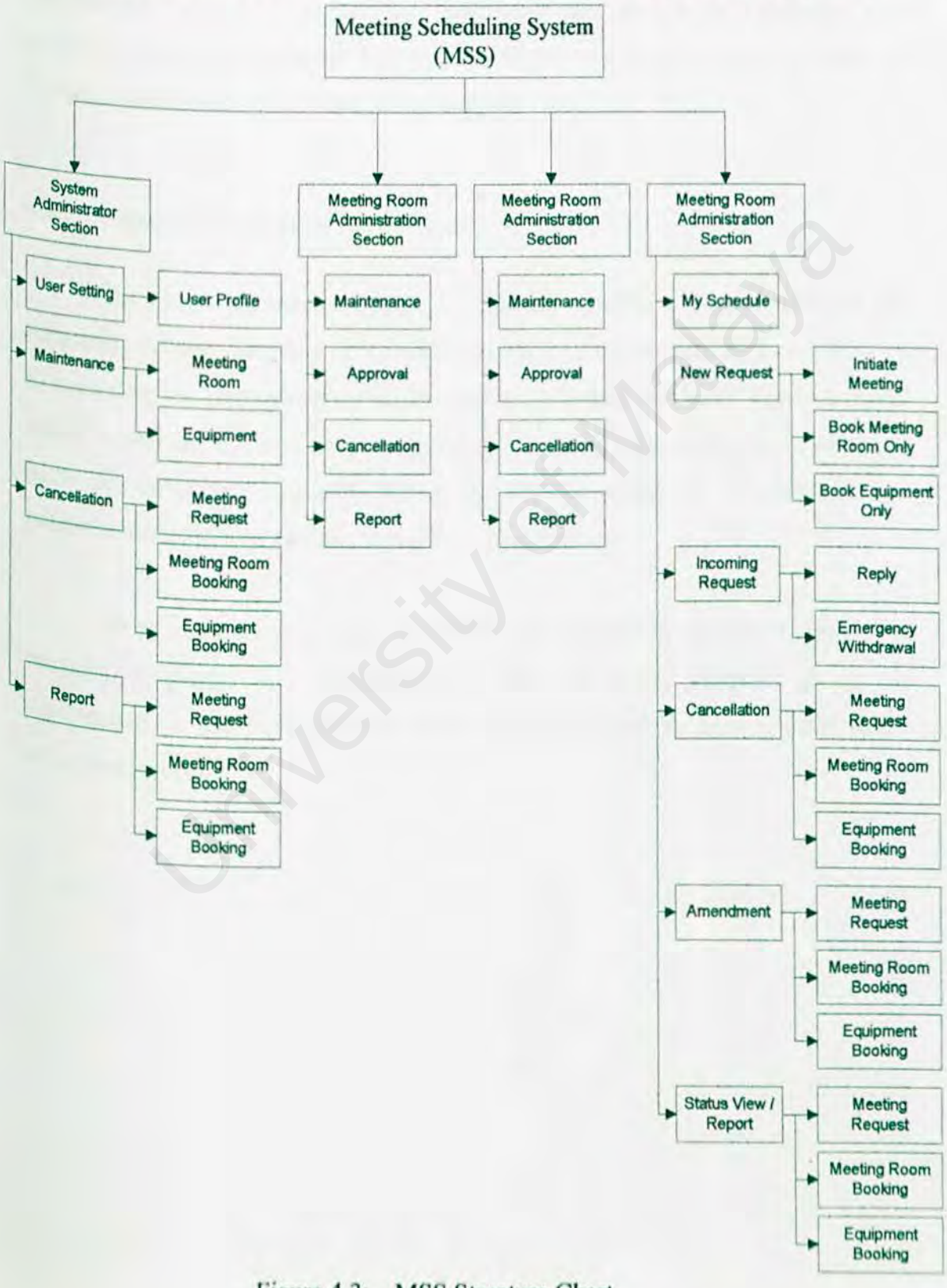


Figure 4.3: MSS Structure Chart

## 4.2 System Database Design

Database design involves the activity on modeling the structure of a database that will store and maintain the data records. It will also include the transformation of user's processing requirement and unordered information into proper functional requirements. The Meeting Scheduling System (MSS) is based on relational database model to keep and manipulate data.

### 4.2.1 Entity-Relationship (E-R) Model

The entity-relationship model is frequently used to logically represent the connection between data and entity within a system. It can be seen as a methodology in expressing the relationship among the entities with their attributes. The E-R model assists in defining the data processing and requirements constraints that lie ahead. It helps the developer especially during the implementation of the database and interprets different view of data from different perspective.

There are several components, which are included in the E-R diagram. For example, rectangles are representing the sets of entity, diamond shape are representing the sets of relationship while the line connects the sets of entity to the set of relationship.

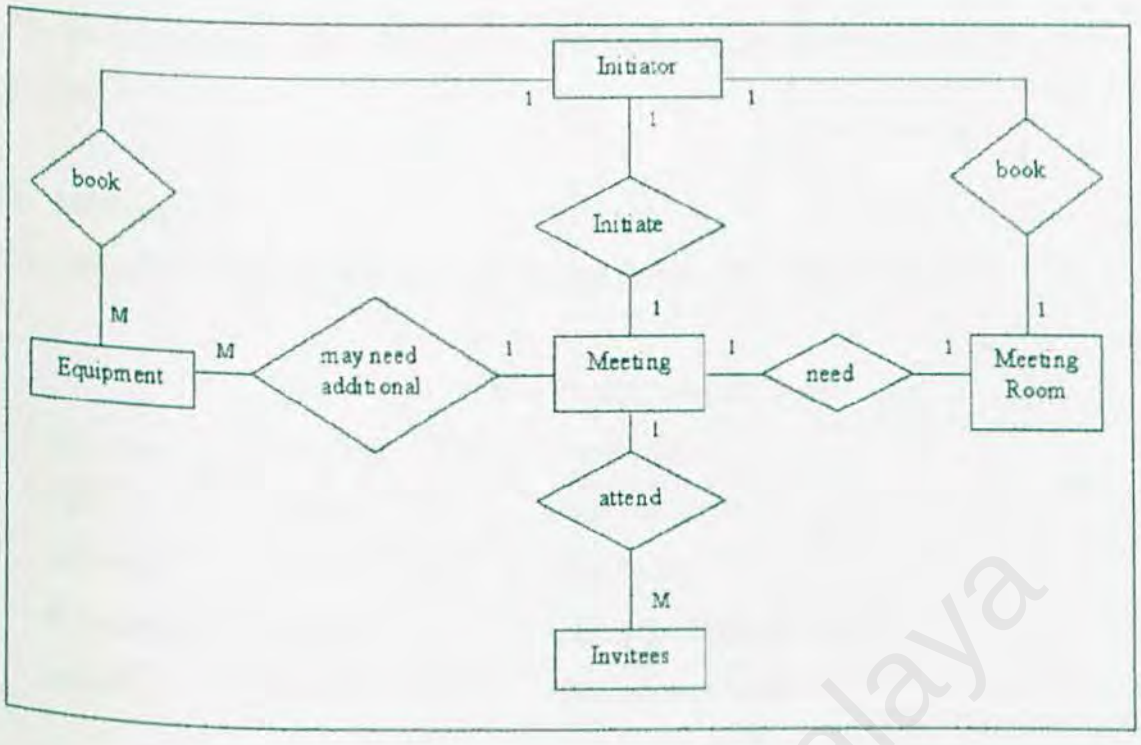


Figure 4.3: MSS E-R Diagram

### 4.2.2 Data Dictionary

MSS uses the relational database model in its database implementation. The database is constructed using the SQL Server 7.0. Listed below are the attributes related to the database.

Table 4.1: MSS Database General Profile

| | |
|---|---|
| Data source name | MSS |
| Type | SQL Server 7.0 |
| Usage | Maintains and keeps the record related to system. |
| Number of tables | 10 |

The table structure of the 10 relations in the MSS database is listed in the following sections.

**1. MSSUser**

This table store MSS user profile. The primary key for this table is EmpId.

Table 4.2: Table Structure for MSSUser

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| EmpId | Varchar | 8 | User unique ID |
| EmpName | Varchar | 50 | User name |
| DepartmentId | Varchar | 5 | User department unique ID |
| ContactNo | Varchar | 12 | User contact number |

**2. Department**

This table store information of departments. The primary key for this table is DepartmentId.

Table 4.3: Table Structure for Department

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| DepartmentId | Varchar | 5 | Department unique ID |
| Description | Varchar | 50 | Department name description |

**3. MeetRoom**

This table store information of meeting rooms. The primary key for this table is MRId.

Table 4.4: Table Structure for MeetRoom

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| MRId | Varchar | 5 | Meeting Room unique ID |
| MRName | Varchar | 50 | Meeting Room name |
| Capacity | Int | 4 | Meeting Room capacity |
| Remark | Varchar | 100 | Meeting Room description or remarks |
| ApproverId | Varchar | 8 | Meeting Room booking approver unique ID |

**4. Equipment**

This table store information of equipments. The primary key for this table is EQId.

Table 4.5: Table Structure for Equipment

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| EQId | Varchar | 5 | Equipment unique ID |
| EQName | Varchar | 50 | Equipment name |
| Remark | Varchar | 100 | Equipment description or remarks |
| ApproverId | Varchar | 8 | Equipment booking approver ID |

**5. MeetingMaster**

This table store information of meetings being initiated. The primary key for this table is MTRefNo.

Table 4.6: Table Structure for MeetingMaster

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| MTRefNo | Varchar | 12 | Meeting reference number |
| MeetName | Varchar | 50 | Meeting name |
| InitiatorId | Varchar | 8 | Meeting initiator unique ID |
| MRId | Varchar | 5 | Meeting Room unique ID |
| MeetDate | Datetime | 8 | Meeting date |
| StartTime | Datetime | 8 | Meeting start time |
| EndTime | Datetime | 8 | Meeting end time |
| Purpose | Varchar | 100 | Meeting purpose |
| Remark | Varchar | 100 | Meeting description or remarks |
| Status | Char | 1 | Status of meeting initiated |
| Reason | Varchar | 100 | Reject meeting invitation reason |
| ReminderChk | Char | 1 | Status flag for reminder |

**6. MeetingDetails**

This table store information of invitees being invited. The primary key for this table is MTRefNo.

Table 4.7: Table Structure for MeetingDetails

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| MTRefNo | Varchar | 12 | Meeting reference number |
| InviteeId | Varchar | 8 | Meeting invitee unique ID |
| Remark | Varchar | 100 | Remarks |
| Status | Char | 1 | Invitee reply status |
| ReminderChk | Char | 1 | Status flag for reminder |

**7. MRoomDetails**

This table store information of meeting rooms booking. The primary key for this table is MRRefNo.

Table 4.8: Table Structure for MRoomDetails

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| MRRefNo | Varchar | 12 | Meeting room reference number |
| EmpId | Varchar | 8 | User/Requester unique ID |
| MRId | Varchar | 5 | Meeting Room unique ID |
| MTRefNo | Varchar | 12 | Meeting reference number |
| BookDate | Datetime | 8 | Meeting Room booking date |
| StartTime | Datetime | 8 | Meeting Room booking start time |
| EndTime | Datetime | 8 | Meeting Room booking end time |
| Purpose | Varchar | 100 | Meeting Room booking purpose |
| NoPerson | Int | 4 | Number of people using Meeting Room |
| Status | Char | 1 | Status of meeting room |
| Reason | Varchar | 100 | Reject Meeting Room booking reason |

**8. EquipDetails**

This table store information of equipment bookings. The primary key for this table is EQRefNo.

Table 4.9: Table Structure for EquipDetails

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| EQRefNo | Varchar | 12 | Equipment reference number |
| EmpId | Varchar | 8 | Requester/User unique ID |
| EQId | Varchar | 5 | Equipment unique ID |
| MRId | Varchar | 5 | Meeting Room unique ID |
| MTRefNo | Varchar | 12 | Meeting reference number |
| BookDate | Datetime | 8 | Equipment booking date |
| StartTime | Datetime | 8 | Equipment booking start time |
| EndTime | Datetime | 8 | Equipment booking end time |
| Purpose | Varchar | 100 | Equipment booking purpose |
| Status | Char | 1 | Status of equipment |
| Reason | Varchar | 100 | Reject Equipment booking reason |

**9. ID**

This table store the last running number for meeting rooms and equipments unique ID. The primary key for this table is ReqType.

Table 4.10: Table Structure for ID

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| ReqType | Char | 2 | Request type – e.g. MT (meeting), MR (meeting room) and EQ (equipment) |
| LastID | Varchar | 8 | Last unique ID |

**10. RefNo**

This table stores the last reference number for meeting initiated, meeting rooms and equipments booking. The primary key for this table is ReqType.

Table 4.11: Table Structure for RefNo

| Field Name | Data Type | Size | Description |
|---|---|---|---|
| ReqType | Char | 2 | Request type – e.g. MT (meeting), MR (meeting room) and EQ (equipment) |
| LastRefNo | Varchar | 3 | Last reference number |

## 4.3 Design of Graphical User Interface

The overall process for design of a user interface begins with the creation of different models of system function. The user interface design is based on the GUI approach. Some of the Human-Computer Interface (HCI) general principles of designing an interactive system have been considered and applied. These HCI general principles are listed in Table 4.11

Table 4.12: HCI General Principles

| Principles | Description |
|---|---|
| Consistency | Consistent format for command input, data display, button selection and placing of the control objects. |
| Confirmation and verification message | Ask for verification of any non-trivial destructive action such as deleting document. |
| Recoverability | Ability of the user to take corrective action once an error has been recognized. |
| Forgive mistake | The system should protect itself from user error that might cause it to fail. |
| Reverse action | Allow the user to return to the previous state. |
| Responsiveness | How the user perceives the rate of communication with the system. For example, the mouse pointer changes to hourglass or display a wait message when the system is processing data. |

# Chapter 5 SYSTEM IMPLEMENTATION

## 5.1 Introduction

During this phase, the design model of MSS is transform into workable product. This phase involves the coding of the program using the appropriate programming language and coding approach, testing of the system to ensure every functions of the system work properly and debugging the code, which will identify and correct the bug within the program.

Before the coding starts, the communication has to be setup and several installation configuration and settings need to be done, especially for providing the features in MSS. Where applicable, each installation that are carried out will be discussed relating to the features being used in MSS and the manual on how to set up is documented in the User Manual in Appendices 2.

## 5.2 Active Directory Services

MSS utilize Active Directory services provided by Windows 2000 Server. Active Directory manages all the user and group information in a Windows domain. Besides that, Active Directory also tightly integrates with Exchange 2000 Server that enable Active Directory and Windows 2000 Server to take over user and mailbox management responsibilities. Because Active Directory already manages the domain accounts, it makes sense that Active Directory should manage the mailboxes that belong to those same users. This means the administrator of the system get better manageability and stronger security.

In addition, Active Directory manages the information specific to an individual Exchange server. When developing solutions with Exchange 2000, Active Directory can be utilize to create users with mailboxes, or create mailboxes for existing user accounts and create security groups for applying access rights to group of individuals.

### 5.2.1 Lightweight Directory Access Protocol (LDAP)

Lightweight Directory Access Protocol (LDAP) is the standard Internet protocol used for directory access. It is the wire protocol used to conduct conversations between a client application and an Active Directory across the network. Although code can be written directly to LDAP to access Active Directory, but it is better to implement interfaces, such as Active Directory Service Interfaces (ADSI) or Collaboration Data Objects (CDO). These interfaces use the LDAP wire protocol as the communication layer through an LDAP provider.

### 5.2.2 ADSI & CDO

Active Directory Service Interfaces (ADSI) is Component Object Model (COM) interfaces that provide an abstraction layer for manipulating resources stored in a directory service. Collaboration Data Objects (CDO) also provides a number of objects and interfaces for managing users and mailboxes.

In MSS, ADSI is chosen because ADSI is implemented as a set of COM interfaces, any application that supports COM, including Microsoft Visual Basic development system and Active Server Pages (ASP) technology can use it.

Whether ADSI or CDO is being used to access Active Directory, a binding string of LDAP need to be built. Everything in Active Directory has an associated path string that represents its exact location in the directory store. LDAP binding string is used to connect to Active Directory, open user objects, access groups, and navigate containers.

### 5.2.3 Building the LDAP Binding String

The LDAP binding string consists of the protocol identifier followed by the server name and the distinguished name (DN) of the object:

LDAP://servername/DN

The provider portion (LDAP) of the string is case sensitive. Attributes that make up the distinguished name are:-

- CN, which is the common name of the object or container
- OU (Organizational unit), which behave similarly to containers and are used to manage objects in Active Directory.
- DC, which is the Domain controller identifier.

In MSS, the LDAP binding string are:-

```
strLDAP = "OU=MSSUsers, DC=servername, DC=net"
```

where MSSUsers is the organizational unit created in Active Directory Users and Computers. (Please refer to User Manual on steps to create organizational unit).

### 5.2.4 Organizational Unit and Group in MSS

MSSUsers is the organizational unit created in Active Directory Users and Computers. Besides that, 4 groups are being created as follow:-

- **AdminJob** group – contains user that are given System Administrator rights
- **MSSMRAdmin** group - contains user that are given Meeting Room Administrator rights
- **MSSEQAdmin** group - contains user that are given Equipment Administrator rights
- **MSSUsers** group - contains all the user that are eligible to use MSS

At user log on, the system will check on which group a user belongs to, access level will be determined and different task can be perform by each level.

### 5.2.5 Opening A Container or Organizational Unit

To open a container or organizational unit in Active Directory, the following code is being used.

```
strLDAP = "OU=MSSUsers, DC=servername, DC=net"
Set objOU = GetObject("LDAP://" & strLDAP)
```

### 5.2.6 Setting Properties Values in Active Directory

The Put and PutEx method is used to set the property values in Active Directory. To set the property value for a single-valued property, call the Put method. On the other hand, the PutEx method gives you added control when writing to the cache. PutEx method can be used to append a value to an existing one in a multi-valued property, to make changes to an individual value, to delete all the values or even to remove a specific value from the set.

Following is the code use to update user profile entered into Active Directory:

```
If Attrib_Value = "" or isnull(Attrib_Value) then
' If Not, Send a Clear message for this attribute to the directory.
' Just in Case the Value is NOW set to Nothing
objUserTemp.PutEx 1, Attrib_Name, vbNullString
Else
objUserTemp.Put Attrib_Name, Attrib_Value
End If
```

### 5.2.7 Querying Active Directory Using ADO

ActiveX Data Object (ADO) 2.5 can be used to navigate in the Active Directory store and query for particular objects.
Following is the code use to retrieve user e-mail address from Active Directory:

```
Function GetADSIConnect(ByVal P_EmpNo As String, ByVal P_RefNumber As
                        String, ByVal P_Ldap As String) As String
Dim strSQL As String          'Select statement
Dim rst As ADODB.Recordset    'Recordset for ADSI
Dim cnn As Connection         'Connection to ADSI
Dim EmpEmail As String        'E-Mail Address

Set cnn = New Connection
With cnn
   .Provider = "adsDSOObject"
```

```
    .Open P_Ldap
End With

strSQL = "SELECT employeeid,mail FROM '" & P_Ldap & "' " & _
        "WHERE employeeid = '" & strEmpID & "' and objectcategory='user'"
Set rst = New Recordset
rst.Open strSQL, cnn

rst.Close
cnn.Close
Set rst = Nothing
Set cnn = Nothing

GetADSIConnect = EmpEmail
End Function
```

## 5.3 CDO Messaging Classes and Interfaces

Microsoft Exchange 2000 Server and CDO support the Internet standards that specify how messages should be formatted for transmission across a network. The Message class is the cornerstone of CDO messaging solutions.

Following is the code use to send messages to all user e-mail address from Active Directory:-

```
Function SentMail(ByVal varFrom As String, ByVal P_EMail As String, ByVal
                  strmsg As String, ByVal StrSubject As String) As Boolean
Dim objMSG
Dim objInfo
On Error GoTo errorhandler

Set objInfo = CreateObject("ADSystemInfo")
```

```
Set objMSG = CreateObject("CDO.message")

objMSG.To = P_EMail
objMSG.From = varFrom
objMSG.TextBody = strmsg
objMSG.Subject = StrSubject
objMSG.Send
Set objInfo = Nothing
Set objMSG = Nothing
SentMail = True
Exit Function

errorhandler:
  Set objInfo = Nothing
  Set objMSG = Nothing
  Err.Raise Err.Number, Err.Source, Err.Description
  SentMail = False
End Function
```

Both SentMail and GetADSIConnect function and is written in a dll(dynamic link library) and installed as a COM+ in the server. This is to fasten the process of sending e-mail notification. (Please refer User Manual on how to install the dll that have been written).

## 5.4 Coding

Coding involves the activity of writing the modeled program into computer code by using an appropriate programming language. In addition, every algorithms that are designed during the design phase are transform into lines of codes, which forms functions in the program.

### 5.3.1 Coding Methodology

The two approaches used for designing systems are the function-based approach and the object-based approach. The function-based or decomposition approach is a top-down approach. It decomposes a system into hierarchy of modules such that the higher-level modules describe the system in general terms while the lower-level modules describe the system in specific terms. Thus in this approach the system development begins at a high-level (or general) description and then down on to a low-level (or detailed) description.

The object-based or composition is a bottom-up approach. It builds a system using modules called objects. The objects are self-contained modules that encapsulate both data and methods (or functions or procedures). The objects are self-contained in the sense that data within an object are manipulated by the methods within the object. Higher-level objects are then built using the lower-level objects. Finally, the entire system is built.

MSS is developed using top-down approach, where the entire system is divided into different modules. As the breakdown goes deeper, each module consists of sub-modules and functions. Each individual function will form the basis where the coding initiates. This approach ensures that every module are developed and tested completely and individually.

MSS is broken up into simpler modules and well-defined interfaces, as it will be more modular. The reason why modularity is desirable is because a modular system is easy to understand, code, debug, and maintain.

A module is said to be cohesive if all the instructions in the module perform a single task or small set of related tasks. A module that performs several different tasks is not cohesive. In MSS, every module is well-defined to perform certain task, as cohesive module is structured and easy to code, debug and maintain.

Coupling refers to the degree of dependence between modules. Two modules are coupled if each depends on the other for its proper functioning. That means is that a bug in one module can infect the other the other module. That also means that if one module is modified, the other may also need to be modified. High coupling between modules is therefore not a desirable design characteristic. IN MSS, low coupling between modules is being practiced.

For ease of maintenance, the program modules must be properly structured. A program module is said to be structured if it has exactly one entry point and one exit point. This can be accomplished by using the structured constructs: sequence, selection and repetition (or iteration). The use of GOTO is usually not recommended as it usually leads to unstructured code. Unstructured modules are difficult to understand, code, debug and maintain.

### 5.3.2 Coding Standards Used

#### 5.3.2.1 File Name Conventions

Each program file in MSS is named using the file name conventions below:-

1. FileName_**FRM**.asp – forms that accept user input data.
2. FileName_**VIEW**.asp – forms that display record retrieve from database. For example is report.
3. FileName_**PRC**.asp - forms that will do the back-end processing.

#### 5.3.2.2 Program File Documentation

Each program of MSS should include documentation, which consists of the file name, module, date, programmer, function and last modified developer and date. This documentation serves as based line for the programmer to ease the job of doing maintenance. Besides that, documentation should also be placed in some complex program or some uncommon used code.

The opening documentation of each program should look like below:

```
'************************************************************
'File Name:
'Module:
'Date:
'By :
'Functions:
'************************************************************
'Last modified:
'************************************************************
```

# Chapter 6 SYSTEM TESTING

Testing is the process of establishing the existence of errors. The purpose of testing is to detect the presence of errors in software-errors that have not been discovered yet.

## 6.1 Testing Techniques Used

### 6.1.1 White Box Testing

White box testing is the type of testing that deals directly with the structure of the code within a module or a code segment. There is six types code coverage in white box testing. Most of the testing is discussed in the Unit Testing (refer section 6.2.1). Following is the types of white box testing:-

#### 6.1.1.1 Segment Coverage

Every segment of the code between control structures is supposed to execute at least once.

#### 6.1.1.2 Branch Node Coverage

Every branch at every possible direction is taken at least once.

#### 6.1.1.3 Compound Condition Coverage

When multiple conditions appear in the code, every possible combination is tested based on a truth table.

#### 6.1.1.4 Basis Path and Path Testing

Each independent path throughout the code is usually taken at a predetermined order. When dependencies appears in the code, each path where dependency appears exists must be tested.

#### 6.1.1.5 Data Flow Testing

Data flow testing is meant for reflecting dependencies, which are mainly caused by sequence of data manipulations. This approach is to uncover anomalies such as variables that are used but not initialized, declared but are not used. It is not used to package the data to minimize the data dependencies.

#### 6.1.1.6 Loop Testing

This type of testing is related to testing single loop (WHILE, REPEAT-UNTIL, FOR loop), concatenated loops (sequence of loop) and nested loops (one or more loops within loops). Loop testing is difficult to test when dependencies exist among the loops or between a loop and the code that it contains. This will cause the number of permutations of tests to approach infinity.

### 6.1.2 Black box testing

This type of testing involves testing the functions of a module without knowing the logic structure of the code. It focuses on the most important aspect of a module that is how well the module meets its specification. There are several tests that belong to black box testing

#### 6.1.2.1 Error guessing

This approach uses test cases that are written to test functions and part of functions. This approach are similar to 'ad hoc testing' where the tester will try any type of test cases which come across his mind or any preplanned test cases.

#### 6.1.2.2 Boundary Value Analysis

This type of testing involves the boundaries of equivalent classes where the coverage of test cases will involve inside the boundary, on the boundary and outside the boundary. The following examples show the boundary value analysis on the meeting scheduling system.

Each meeting initiated, equipment and meeting room booking is supposed to be supplied with a date, start date and end date. There are several of test cases:

a. The meeting or booking date is not within a historic date and time. A historic date and time means a date and time, which has occurred in the past.
b. The meeting or booking date does not overlapped with any existing meeting or booking period. The booking period should not lie within another existing record booking period.
c. The meeting or booking date should not overlap with the starting period of an existing record of meeting or booking.
d. The meeting or booking date should not overlap with the ending period of an existing record of meeting or booking.
e. The meeting or booking date should not overlap with 2 or more existing record of meeting or booking.

#### 6.1.2.3 Module Interface Testing

In this type of testing, each value within the interface is assured as correct as they relate to modules that call them. This means that specific calls in the calling modules must be tested to see that they are in the right sequence and of the right type.

## 6.2 Testing strategies

The testing strategies involve the unit testing, integration testing and system testing.

### 6.2.1 Unit testing

The unit testing aimed at the verification of the smallest unit within a program, namely the module. The primary goal of unit testing is to confirm that the unit is correctly coded and that it carries out the functions it is supposed to carry out.

Each unit is tested independently in order to assure their accuracy. In MSS, each module may contain sub-modules, which in turn consists of functions. These functions are individually tested before the entire module tested.

#### 6.2.1.1 MSS Unit Testing

Following is the unit testing being done:-

1. Test whether the user can successfully logged into the Domain Web Server and Exchange Server.
2. Test whether the send e-mail function can sent e-mail to the appropriate receiver stating the correct information as intended by the sender. If the function is unable to deliver any message, the sender will be prompted with the error message stating that the sending process was unsuccessful or else stating that the message has been properly sent.
3. After a single program file has been developed, it is tested for the display of HTML page and to ensure that all the display is correct and expected. These interfaces contains lots of button and testing this button is needed in order for this program to perform correct action or to link to right location
4. The program files may include various type of JavaScript checking. For example, functions like checkNumber(), checkEmpty() and checkDate(). These functions are very useful for online error checking. So ensuring this function is right and can perform well is a must in system testing.
5. After transaction occurred, the database table related should be checked either the insert, modify or delete action has been perform correctly by performing query on this table.
6. Test whether the data that being passed to the next program for processing contain the right value.
7. Test whether the records being displayed in the report is correct and matches the filtering criteria.

#### 6.2.1.2 MSS Debugging Strategies

1. Using the command prompt

If a program not performing well, which means this program occurred some error. The command prompt has been used to view the error message occurred in coding. This is useful in order to debug the program.

2. Reviewing the algorithms used

If a program is running well, but the information is not what as intended, then maybe a logic error or database error have occurred. Reviewing algorithms and computations for their correctness and efficiency is needed for this problem. Sometimes, by using different algorithms, the efficiency of the program will be increased.

3. Display on screen the passing value

One of the possibilities of wrong information being retrieved is that the wrong value is being passed from one page to another page that will do the processing. To ensure that right value has been passed to the next program for processing, the passing value is display on screen for viewing.

4. Check Success Status

Some processes are dependent where failure in the previous process will affect many other processes. In order to avoid chain reaction from this kind of process, a success status is purposely set to return a true or false value. The success status is checked to determined whether to continue the process or to exit from the program and display error message.

5. Using Query Analyzer provided by SQL Server 7.0

When the database transaction error occurred, then the error message will be displayed and most of the time, it is cause by incorrect syntax in the SQL statement. Therefore, SQL statement being used to execute the transaction will be test using Query Analyzer provided by SQL Server 7.0 that will provide more information on the error. Other than that, Query Analyzer is used to correct the SQL statement when

wrong information is being retrieved although the SQL statement being used return no errors.

### 6.2.2 Integration Testing

The purpose of the integration testing is to know whether the entire software is able to work as one program. It will also verify that each module will be able to function together. Integration testing is usually black box testing although it is possible to link processing or logic paths between two modules. The approach applied in testing the system is referred as Top-Down Integration where integration will starts with the highest-level main program and modules & sub-modules are gradually added until the bottom is reached.

### 6.2.3 System Testing

A system testing is a series of different tests designed to fully exercise the system to uncover its limitation and to measure its capabilities. The objective is to test an integrated system and verify that it meets the specified requirements. At this point, the behaviour of the tests will define the quality of the systems.

Due to the heavy workload of the staff of Faculty of Computer Science and Information Technology. The system testing for MSS is tested by my peer friends, where we took turn testing each other system.

# Chapter 7 SYSTEM EVALUATION & CONCLUSION

## 7.1 Problem Encountered and Solutions

### 7.1.1 Scope Not Properly Defined

In the initial stage of development, several problems were encountered in specifying the scope of the system. The scope is need before coding be done because it is also needed in the design of the system. The solution was to capture the system requirements through interviews with all the potential user in Faculty of Computer Science and Information Technology, check the available similar sites in Internet and joining discussion group.

### 7.1.2 Inexperienced in the chosen programming language

Since there was no prior knowledge of programming in ASP, there was an uncertainty in on how to develop a web-based system using ASP. These new programming languages and concepts were never taught, and to implement such a application requires a fair grasp of the language.

Though those were one of the obstacles encountered in the early stages, choosing to program in ASP proves to be a wise move, as it is a very powerful technology to built web-based application. Problems were solved through research on related material and referring to some of the reference book available in the market. Discussion with friends using the same technology also helps in solving the problems faced.

### 7.1.3 Lack of knowledge on Database server

Using an extensive and relational database like Microsoft SQL Server 7.0 indeed requires much understanding of its concepts and its features. The way of

administering and manipulating them are far different form ordinary stand-alone database systems. Therefore, much reading and research is done to be able to have a fair understanding on how to use the feature provided by Microsoft SQL Server 7.0.

### 7.1.4 Lack of knowledge on Microsoft Exchange 2000 Server

As Microsoft Exchange Server 2000 Server is new technology just being released, not much reference and help can be found in how to use and fully utilize the powerful feature that Exchange 2000 Server provides. Much reading and research is done in this area in order for me to have a basic understanding on how to configure and use Exchange 2000 Server especially in the area of Active Directory services, which integrate tightly with Windows 2000 Server.

## 7.2 System Strengths

### 7.2.1 Ease Meeting Scheduling Process

MSS is being developed to simplify the process of meeting initiation and meeting room and equipment booking. MSS provide the feature to check the availability of the invitees whether being invited to other meeting in the same time. This is to ensure that no one is being scheduled to attend two or more meeting in the same time. On resources booking (meeting room and equipment), MSS also provide the feature to check availability of the resources. User can then initiate meeting and place their booking on-line. Invitee will reply on-line to the meeting invitation. As for resource booking, approver of the resource will approve or reject the request on-line too.

### 7.2.2 Manageability and Security

MSS utilize NT login, which take advantages of Active Directory services provided by Windows 2000 Server. Active Directory manages all the user and group information in a Windows domain. Besides that, Active Directory also tightly

integrates with Exchange 2000 Server that enable Active Directory and Windows 2000 Server to take over user and mailbox management responsibilities. Because Active Directory already manages the domain accounts, it makes sense that Active Directory should manage the mailboxes that belong to those same users. This means the administrator of the system get better manageability and stronger security.

### 7.2.3 General Solutions

MSS is developed based on requirement of Faculty of Computer Science and Information Technology, but the algorithms used can be easily extended to other faculties in campus. These provide a general solution that can be customized for other organization with similar needs.

### 7.2.4 Graphical User Interface

As a we-based application, MSS shows some of the advantages in terms of its usability. Windows feature such as consistent user interfaces are available to aid users in using the system. MSS provides a standard interface appearance through the whole system. Other than that, MSS provides a drop-down menu bar that is easy to use and similar to window environment. Besides that, it has a friendly graphical user interface where all type of buttons is well defined. This is to ensure that the user can easily use the system without any briefing or with minimal training.

### 7.2.5 Auto Sending E-mail Notification

MSS will auto send email to related person when transaction including initiate meeting, book meeting room and equipment, cancellation, amendment and reminder. This is to ensure that the related person will be notified and there is no need for them always login to the system for checking the status of any meeting organization or resource booking transaction.

### 7.2.6 Status View and Reports

MSS reports list all status of the transaction being made such as checking all the invitees reply status by the meeting initiator or checking the status of the meeting room or equipment being booked. Report can be filtered by using some criteria like date range and status. This reports is to provide searching and listing functions, which will ease the job of the initiator of the meeting, invitee, requester of meeting room and equipment and the approver in this system.

## 7.3 System Constraints

### 7.3.1 Browser Limitation

Most of the scripts are written in VBScript, which is not supported by other browser such as Netscape Communicator. This means that MSS are unable to be use with certain type of browser that does not support VBScript.

### 7.3.2 Report Is Not in "Printer-Friendly Format"

As MSS is view using web browser and the interface is designed to be very attractive and user-friendly, therefore it is not very suitable to print MSS report, as it is not in formal report format and moreover, it will consume much ink.

### 7.3.3 Checking of Availability is Based-On Whether Being Scheduled to Another Meeting Only

Checking of availability of the invitee of the meeting is solely based on the concept that if the invitee is not being scheduled to attend other meeting, then they are free to attend to any meeting being initiated. But, the invitee may be busy although they are not scheduled to attend meeting. This is however acceptable as the main objective of MSS is to avoid two or more meeting being scheduled to a person in a same time.

## 7.4 Future Enhancement

### 7.4.1 To Fully Utilize Microsoft Exchange 2000 Server

Microsoft Exchange 2000 Server is a very powerful tool as it combines messaging, collaboration, and application development. Microsoft Exchange 2000 Server can serve as a database, a web server, and a collaboration server using new Microsoft Web Storage System technology. It will be a future enhancement to be able to use Microsoft Exchange 2000 Server to fully support MSS entirely.

### 7.4.2 To Integrate With User Calendar or Task Schedule

As MSS only provide availability check that solely based on the concept that if the invitee is not being scheduled to attend other meeting, then they are free to attend to any meeting being initiated, therefore as a future enhancement, MSS should integrate with user calendar or task schedule to provide a better availability check.

### 7.4.3 To Provide "Printer-Friendly Format"

This is to provide a feature that will format the report into a "printer-friendly format" and formal report format. This is to enable user of MSS to print the report in offered MSS.

### 7.4.4 To Offer A Date Range in Organization of Meeting and Booking of Resources

Currently, MSS only offer user to enter one date to organize meeting or to book resources. MSS should offer user to enter a date range for organization of meeting or booking of resources in future.

## 7.5 Knowledge and Experience Gained

Beside knowledge on technical aspect such as Windows 2000 Server Family Operating System, SQL Server 7.0 database, ASP, VBScript, JavaScript, VB and Microsoft Exchange 2000 Server, there are also other valuable experience gained from working on this project such as: -

1. Being exposed to the real system development environment, especially dealing with users.
2. Learn how to manage a project as in time and resource.
3. Concept on how to integrate and fully utilize various technologies into a developing a system.
4. Get experience on how to set up and configure various technologies to be able to serve as a live system.

## 7.6 Conclusion

As a conclusion, the project has met its objectives of developing web-based messaging-enabled systems. MSS can support the organization of meetings, that is, for each meeting request, a suitable meeting date can be determined based on constraints like meeting location, equipment and number of attendees that can attend the meeting. In addition, it also projected the main idea of Generic Office Environment as to promote a paperless environment with the routing of information through the workflow application.

Throughout the development of the project, a lot of precious knowledge on web-based programming was gained. This includes configuration & management of Windows 2000 Server Family and IIS 5.0, programming knowledge in Active Server Pages (ASP), Visual Basic 6.0 (VB 6.0), VBScript and JavaScript as well as the techniques and concept in implementing the database, Microsoft SQL Server 7.0 and mail server (Microsoft Exchange 2000 Server). This project has been a very wonderful experience, which exposes the idea of research work to me.

# REFERENCE

[1] Pleeger, Shari Lawrence. 1998. *Software engineering theory and practice.* United States of America. Prentice_Hall Inc.

[2] Razzo, Thomas. 1999. *Programming Microsoft Outlook and Microsoft Exchange.* United States of America. Microsoft Press. pp. 3-15.

[3] Easlick, Richard and Chellis, James. 1997. *MCSE: Exchange Server 5.5 study guide.* United States of America. SYBEX Inc.

[4] Applebee, Ann., Clayton, Peter., Pascoe, Celina. And Bruce, Harry. 2000. *Australian academic use of the Internet: implications for university administrators.* Internet Research: Electronic Networking Applications and Policy. Vol. 10, No. 2, pp. 141-149.

[5] Zantinge, Dolf and Adriaans, Pieter. 1997. *Managing client server.* Addison-Wesley Publishing Co. pp. 1-67.

[6] Benson, A. 1996. *Client/server architecture.* New York. McGraw-Hill.

[7] Renaud, P. 1996. Introduction to client/server systems. New York. John Wiley & Son.

[8] Burke, E.K., Ellimen D.G. and Weare, R. F. The automation of timetabling process in techmology system. Journal Of Education Techonology System. Vol 23(4). 1995. pp 257-266.

[9] Schaerf, A. 1995. A survey of automated timetabling. Centrim voor Wiskund en Informatica (CWI) report. CS-R9567.

[10] Gomaa, Hassaa. 1993. *Software design methods for concurrent and real-time system.* Reading, Massachusettes. Addison-Wesley Pub. Co. Inc.

[11] Boehm, Dr. Barry W. 1981. *Software Engineering Economics*. Englewood Cliffs, NJ. Prentice-Hall.

[12] Boehm, Barry W. 1988. *A Spiral Model of Software Development and Enhancement*. IEEE Computer.

[13] Budde, Reinhard and Kautz, Karlheinz. 1992. *Prototyping an approach to evolutionary system development*. Berlin Heidelberg. Springler – Verlag.

# BIBLIOGRAPHY

Joshi, Kent. et al. 1997. *Speacial edition using Microsoft Exchange Server 5*. United States of America. Que Corporation.

Kendall, Kenneth E., Kendall, Julie E. 1999. *System Analysis And Design*. United States of America. Prentice-Hall, International, Inc.

Martin, Mindy C. 2000. *Programming Collaborative Web Applications with Microsoft Exchange 2000 Server*. Redmond, Washington. Microsoft Press.

Sellappan, P. 2000. *Software Engineering: Management & Methods*. Malaysia. Sejana Publishing.

Redmil, felix. 1997. *Software projects evolutionary vs. big-bang delivery*. England. John Wiley & Son.

Sen, S. et al. An Automated Meeting Scheduling System that Utilizes User Preferences, in, *Proceeding of the First International Conference on Autonomous Agents (Agents'97), Marina del Ray, CA, 1997*. pp 308-315.

Sen, S., Haynes, Thomas, and Arora, Neeraj. 1997. Satisfying User Preferences While Negotiating Meetings. *International Journal of Human-Computer Studies*. Vol. 47. Special issue on Group Support Systems. pp. 407-427

Sen, Sandip. An automated distributed meeting scheduler. *Publication in IEEE Expert*.