Chapter 1

Introduction

1.1 Introduction

With the amazing growth of web technology in the past few decades, great changes have taken place in every area of our life, for example university environment. It has brought about the evolution on the ways of teaching, administration activities and processes, such as e-learning, e-library, online portal for administrative system and so forth. However, it is not in the case of appointment management handling between lecturers and college students.

There are two common approaches practicing in the universities/colleges. Lecturers either put sign-up sheets on their doors (manually) or release a fixed time over a period for students to reserve a meeting. These methods do not reflex the dynamic changes or cancellation of the appointments; take an example, managing consultation time, if the lecturer only release the consultation time, many students may want to meet at the same time. There is an urge for the development of efficient web based appointment management system to solve the above problems.

A web based appointment management system is to utilize the web technology to manage the appointments online. Though there are some web based supports in the form of static web pages showing lecturers’ available time for consultation, it does not provide appointment management or any interactions with the students. Therefore, it is necessary to refer to similar systems for other services in order to define the features.

Based on the literature review, most of the existing appointment management systems were developed using ad-hoc approach with consideration on functionality only. There is little emphasis on the design and development process. Only few appointment management systems adopted UML, which is a widely recognized modeling standard to design system. However, because UML adapted to web environment from other sector
like software engineering and due to its complexity; UML can not perfectly achieve the goals of web application development. Web Modeling Language (WebML) which is specified for Web application can reduce development efforts and allows a more structured development process.

Three-tier architecture is the prevailing architecture for web based system during past few years. Most of the existing appointment management systems adopted it. But, as modern Web applications become more sophisticated, that require advanced features like multi-device access (PCs, PDAs, digital televisions, and WAP phones), the one-to-one personalization (myYahoo, myCDNOW), and evolution management[52], three-tier approach falls short in several key areas, such as lack of flexibility and scalability, inefficiency and complex in the development cycle and so on[26, 41, 42].

By taking an evolutionary step, four-tier architecture which separates presentation logic from business logic offers maximum functionality and flexibility in a heterogeneous, Web based environment [11]. The presentation server solves the problems brought on by the proliferation of devices and content types [10].

This project is to turn the traditional approach of appointment management into a web based system called WBAMS by applying WebML and .Net four-tier architecture so that students and lecturers can arrange meetings in an effective and efficient way.

In this system lecturers can manage the consultation schedule over the web, for example, set up recurrent consultation time with booking restriction and appointment volume by specifying dates, times, and durations. Students are able to make appointments, login with password authenticated and cancel appointments and so on. The system also provides the appointment reminder and important notice for both lecturers and students. Appointment cancellation will cause notification in the important notice section.

This project is not only to fulfill these functional requirements, but also designed to satisfy the non-functional requirements like usability, performance, and maintainability.
1.2 Problems Statement
1. The rapid growth and diffusion of the Web are nurturing a novel generation of applications which grow in size and complexity. The development of such applications is a hybrid between traditional information system development and hypermedia authoring [39]. It challenges the existing tools and approaches for software production. The development of the Web application is somewhat chaotic with little emphasis on the design and development process. Most of the web-based systems did not use any formal modeling language for designing the system. Many companies today are experiencing severe problems in the management and maintenance of the web system. Ginige et al. use the term Web crisis to describe this situation [14].

2. It is also found that most of the similar systems were built on the three-tier architecture which cannot measure up performance and maintainability requirements. Because three-tier architecture does not truly separate the application into specialized, functional layers [26]. The middle tier is still too broad with so many functions grouped into one tier. It falls short in several key areas, such as lack of flexibility and scalability, inefficiency and complex in the development cycle [26, 41, 42]. Besides it, there are some limitations on the usability aspect; some of the similar systems do not give appropriate feedback and assistance to the users. There are no “back to Menu” and help functions. Some of the systems are not easy to learn due to the system complexity.

1.3 Project Significance
The significant of the project is to develop a web-based appointment management system for lecturers and students to eliminate the above mentioned problems through new disciplined approach Web Modeling Language (WebML) and .Net four-tier architecture.

The model-driven approach has proved very effective in extending the classical methods and best practices of software engineering to the Web [47]. WebML guarantees the
model-driven approach which can reduce development efforts and allows a more structured development process.

.Net is Microsoft's newly redesigned, revolutionary platform aimed to tie together applications, development languages, operating systems and data stores into a unified, distributed enterprise platform. By adopting .Net four-tier architecture, it provides better performance and maintainability.

This project is an artful blend of these state of the art approaches to fulfil the long-standing needs.

1.4 Project Objectives

The objectives of this project are shown as below:

- To identify the current approaches and the problems in the web-based appointment management systems.
- To design and develop the WBAMS by applying WebML models, .Net four-tier architecture and HCI guidelines based on the books [65, 66] to achieve the non-functional requirements that are usability, performance and maintainability.
- To evaluate and test the WBAMS using testing strategy, software metrics and Microsoft web application stress tool.

1.5 Project Scope

The project is restricted to appointment management for the meeting between lecturers and college students in terms of functionality. There are no scheduling and other educational functions, such as course management, timetable scheduling, and resource booking etcetera. The project will be developed using .Net framework with IIS web server and SQL server support.

The research forces on web design process by considering software architecture (four-tier architecture) and web engineering approach (WebML).
1.6 End Users
In our system there are three groups of users.

- Administrator
- Lecturers
- Students

Each category user has different perspective on the system. The administrator is to maintain the user groups and produce reports. The lecturers concern with their own appointment management, for instance setting up consultation time, canceling appointments; and the students are only interested in the appointments with relevant lecturers.

1.7 Research Methodology
1. The WebML approach includes different phases (i.e. requirement specification, data design, hypertext design, and architecture design) is applied in an iterative and incremental manner to the development of WBAMS.

In the WebML Hypertext Design, there are six types of content units to compose pages: data, multi-data, index, filter, scroller and direct units. Hypertext design is the phase of the entire lifecycle that mostly benefits from a conceptual and model-driven approach [48]. Reasoning about the functions to be delivered by site views and pages is much easier at the conceptual level and with a visual model than at the source code level, results into a more consistent and qualitative design [47].

2. .NET four-tier architecture which separates the functionality into four layers: client tier, web tier, business tier (business logic layer and data access layer) and data tier.

3. There are five common fact finding techniques: research, observation, interview, emailing and questionnaire used in this project. The first four techniques are applied
to help define user requirements; and questionnaire is adopted to collect the user’s feedback on WBAMS for evaluation.

4. Goal-Question-Metric Paradigm is adopted to select the appropriate metrics for questionnaire which evaluates the efficiency of WBAMS.

5. System testing is a critical point to the successful development of WBAMS. It is broken down into the classical activities of unit testing, bottom-up integration testing, and system testing. Black-box testing and white box testing will be used. For the non-functional evaluation, such as usability, performance, and maintainability, the following methods will be adopted, i.e. questionnaire, Microsoft Web Application Stress tool and software metrics.

1.8 Project Schedule

The dissertation was registered on 26th July 2006 and got approval on 30th August 2006. It was completed on 22nd August 2007. The whole project took thirteen months to accomplish. Figure 1.1 Gantt chart shows the detailed schedule.
1.9 Dissertation Organization

There are eight chapters included in this dissertation. The summary of the contents of each chapter is as followed.

Chapter 1: Introduction

This chapter includes the introduction, problem statements, objectives of this project, project scope, research methodology, project schedule and dissertation organization.

Chapter 2: Literature Review

This chapter describes and analyses the similar appointment management systems for hospital, school and other services. Five systems are selected because of functionality, usage of different scripting language, software architecture and usability issues. The dominant platforms (J2EE and .Net platform) that are based on the concept of four-tier architecture are compared in this chapter.

Chapter 3: Research Methodology

The chapter discusses the research methodology like WebML, fact-finding techniques that are research, observation, interview, emailing and questionnaire used in this project. The WebML process, such as requirement analysis, system design, test and evaluation is included in this chapter.

Chapter 4: System Analysis

The chapter covers the data analysis by applying fact-finding techniques, and requirements specifications, which are user identification, use case diagram, functional requirements, non-functional requirements, data dictionary specification, site view specification and hardware requirements.

Chapter 5: System Design
This chapter focuses on designing the WebML models for instance, Data Model, Hypertext Model as well as architecture. This chapter also includes the description of WebML case tool WebRatio which is adopted to visualize the WebML models.

Chapter 6: Implementation
The development tools and programming languages applied to the project are discussed. This chapter also covers the implementation details of WBAMS.

Chapter 7: Evaluation and Testing
A series of test plan will be carried out to make sure the system is run under control. White box and Black box testing are applied. For the non-functional evaluation, such as usability, performance, and maintainability, the following methods will be adopted, i.e. questionnaire, Microsoft Web Application Stress Tool and software metrics. In addition to that, this chapter describes about the degree of success and summarizes the results.

Chapter 8: Conclusion
This Chapter concludes the completed project and gives some suggestions for future enhancements, which can be used as a reference.
Chapter 2

Literature Review

2.1 Appointment Management System
This section defines and discusses the appointment management, current methods and the trends.

Appointment is an arrangement to meet someone at a particular time and place. In this paper, appointment definition is limited to the meeting between lecturers and students in the university/college environment. Appointment management mainly includes consultation setup, appointments and/or consultation management for lecturers and student appointment reservation and so forth. Appointment management system is a method or system designed to manage the appointments.

2.1.1 Current Methods
Currently there are two existing methods to arrange appointments for lecturer and student in the university.

2.1.1.1 Using Sign-up Sheets (Manual)
Every time the lecturers need to put the sign-up sheets on their doors and collect those before meeting for record purpose. Students have to reach the department in advance to fill the sign-up sheet for appointment reservation.

2.1.1.2 Allotting a fixed time over a period
In this method, lecturers release the consultation time through website for a period of one month or one semester. Any students can meet up during the fixed time period. Take an example (see figure 2.1),

![Yuping Wang's Consultation Time](http://www.cit.gu.edu.au/~linzhong/wangpage/consult.htm)

Figure 2.1 Screenshot of one of the lecturer’s web page

### 2.1.2 Problems in Current Methods

For the manual method, students have to reach the department in advance to book the appointment. It often occurs that student could not sign up the meeting due to appointment occupied or time clash with their class or lab sessions. In that case, they need check again in order to fix an appointment. Besides it, the sign-up sheets may at times be lost. It’s troublesome and inconvenient for students as well as lecturers.

Similarly in the releasing consultation time method, lecturers have no idea about how many students, which students and their purposes for consultation before meet them.

The two current approaches can not manage the appointment time and control the number of appointments properly, for instance, many students may come at the overlapped time, the lecturer may exceed the consultation time because of overfull volume of appointments. If the appointments have to be cancelled due to unpredicted
events like medical leave, emergence leave, there is no way to inform students. These methods do not reflect the dynamic changes or cancellation of the appointments.

2.1.3 Trends in Appointment Management
The world has become smaller since the World Wide Web spun around it many years ago. Everyday life goes through a constant revolution. Web based systems enable people to directly access the information and interact in an efficient and organized manner. Furthermore, recent few years, with Mobile phones, handheld game consoles and cellular routers, the Internet can be accessed virtually anywhere.

It is found that there are many web based commercial products in scheduling and managing appointments for various industries. However, it does not apply for lecturers and students in the university (survey conducted at KBU international college). It is necessary to fill up this gap to follow the trends. The similar systems for other services are investigated for features consideration.

2.2 Analysis of Similar Web-based Appointment Management System
This section particularly analyses the similar web-based appointment management systems for other services to define the system features.

2.2.1 Why Choose These Systems?
There are four reasons to justify why these systems had been selected.

2.2.1.1 Functionality Factor
As there is no existing system (exclusive of static Web pages) in the university/college for lecturers and students to arrange the meeting, the similar systems for school, hospital and other services are searched. In the current market, there are lots of calendar systems or appointment scheduling systems, however, most of them do not support appointment booking which is one of the most important features. Thereby, these five systems which have appointment booking function are screened out to analysis and compare.

2.2.1.2 Server-side Programming Language
The choice of a server-side programming language is a constant source of heated debate today [28]. In the last decade, there are four prominent and popular server-side scripting languages existing, namely CGI/PERL, Active Server Pages, Java Server Pages, and PHP.

CGI is a platform-independent solution with a simple, well-known interface. The CGI programs can be written in any programming language, for instance Perl.

Active Server Pages (ASP) is Microsoft's server-side script engine for dynamically-generated web pages [2]. ASP is a framework that allows combination one of a number of scripting languages like VBScript and JavaScript with an expandable set of software components [3].

Java Server Pages (JSP) is a Java technology that enables dynamically generation HTML, XML or other types of documents in response to a Web client request [61]. JSP combines HTML with custom tags and Java code,

PHP has been dubbed by some to be 'Perl killer' of the server-side scripting world [28]. It is a reflective programming language originally designed for producing dynamic web pages.

These five systems are chosen using the above four scripting languages in order to compare the sever side programming approaches in developing applications. It is one of the important factors affecting the performance.

2.2.1.3 System Architecture

Most of the existing systems are using either two-tier or three-tier architecture. Only few large and formal systems applied four-tier architecture. These systems were chosen because of more structure and complete which used three-tier or four-tier architecture. By introducing a separation of business logic from the client and database tier, three-tier/four-tier approach improves flexibility and modifiability compared to two tier architecture. Applications are easier to manage and change without affecting the other tiers.
2.2.1.4 Usability Issue
In addition to functionality factor, server-side scripting language and software architecture, these systems were selected by considering usability issue, such as visual calendar, use of color, navigation.

2.2.2 Existing Similar Systems
Five existing similar systems are screened out according to functionality factor, server-side scripting language, software architecture and usability issues as discussed in 2.2.1 Why Choose These Systems

2.2.2.1 Web-based Appointment Booking System for career service (McMaster University) [ABS]
The system is one of the application systems used open source meeting room booking system. It is applied in McMaster University, Canada for students to book career service session. Students can choose a specified date and check the availability. To schedule an appointment, student just need select the white box with “+” in the center and click it. Confirmation email will be sent when the appointment is successfully reserved. Please refer to [http://www.science.mcmaster.ca/scs/appointment/help.php?day=3& month=02 & year=2007](http://www.science.mcmaster.ca/scs/appointment/help.php?day=3& month=02 & year=2007)

![Figure 2.2 Screen shot of Appointment booking system for Career Service](image)

There is no explicit information about modeling language used for designing this system. Through email contact, it is acquired that no modeling language was adopted for
designing on account of its complexity, but some kind of notations were applied to sketch the system.

Three-tier architecture was used in this system. It consisted of the Apache web server, database server (MySQL) and clients (web browser). PHP was used to implement this system.

![Figure 2.3 Architecture of ABSMU](image)

2.2.2.2 Subsystem of Web-based Home Schooling Application [HSS]

This is a subsystem of a web-based home schooling application which developed by LinChen and Yuwen Ouyang [30]. This system allows teachers to maintain their schedules on-line. Parents can view a teacher’s daily schedule as shown in the figure 2.4. By selecting an open time slot and submitting appropriate information, parents can secure an appointment. This appointment subsystem frees both the teachers and parents from exchanging phone messages in order to settle appointments.

![Figure 2.4 Screen Shot of HSS](image)
UML was applied for this web-based home schooling application. This application used three-tier architecture with Java support. The architecture is shown as following:

![Architecture of HSS](image)

**Figure 2.5 Architecture of HSS**

### 2.2.2.3 Appointmentdiary Demo system ---Brainiac Tutor Company Online Scheduling System [ADS]

AppointmentDiary is an appointment booking software solution which designed for small businesses and professional practices. Here Brainiac Tutor Company Online Scheduling System is one of the demo systems. There are two ways to book the individual tuition for the students in this system. The student can either select the desired date and let Appointmentdiary tell which tutors are available on that day or choose the particular tutor to confirm booking by selecting available booking periods. Please refer to [http://www.appointmentdiary.net/demos.htm](http://www.appointmentdiary.net/demos.htm)
As it is commercial project, there is no formal modeling language used for design process.

The system was implemented using ASP based on three-tier architecture as shown in figure 2.7:

2.2.2.4 CyberMatrix Meeting Manager (web version) [CMM]
CyberMatrix Meeting Manager is an easy to use multi-user application for scheduling meetings and meeting room resource scheduling. It can be used by offices, schools, libraries, churches, community associations and driving schools.

The user can view and book the interested session/meeting by selecting booking periods and confirming as the following screen. The system will automatically e-mail meeting attendees or resource contacts when a booking has been changed. Please refer to http://www.cyber-matrix.com/cmm.html

![Booking Schedule](image)

Figure 2.8 Screen Shot of CyberMatrix Meeting Manager

Meeting Manager (Web version) requires a web server that can serve Windows executable CGI scripts. When a client specifies the CGI program with parameters, the CGI program transforms relevant query into SQL statement that the DBMS can process. Any results produced by the database operation will be packed into HTML documents, which are transmitted to the client. There is no formal modeling language used for
design process.

The architecture is shown in figure 2.9:

![Architecture of CyberMatrix Meeting Manager](image)

**Figure 2.9 Architecture of CyberMatrix Meeting Manager**

### 2.2.2.5 Timetrade (Higher Education Solutions) [TT]

Timetrade is a Web-based appointment scheduling system with many features and functionalities. It helps colleges and universities managing and scheduling the appointment.

The system can be set to be accessible only by authorized staff members, or open things up for self-service scheduling by students and applicants via the Internet. Please refer to [http://www.timetrade.com/](http://www.timetrade.com/)

![Screen shot of TimeTrade system](image)

**Figure 2.10 Screen shot of TimeTrade system**
Partially UML modeling was adopted in the system design. Use case, database model and flowchart were produced after analyzed the functions of the system.

**Architecture**

J2EE Four-tier architecture was used for this system. It was built on open J2EE platform running on the Windows 2003 Server operating system, IBM WebSphere Express application server, and Microsoft SQL 2005 database.

![Figure 2.11 Architecture of TimeTrade system](image)

**2.3 System Comparison**

This comparison consists of two aspects: features and web system quality. Six main areas: appointment management, customization, account setup and management, software architecture, server-side programming language and modeling language are included in the features comparison. For the web system quality comparison, usability, performance and maintainability are considered as the criteria.

**2.3.1 Features Comparison**
The features of appointment management system as shown in the table 2.1 is divided into six sessions: appointment management, customization, account setup and management, software architecture, server-side programming language and modeling language

**Appointment Management**
Does the system offer the basic features for appointment management, like available data and time navigation, appointment confirmation, appointment subject description and appointment report? Does the system provide advanced features to help the user manage the appointments effectively and efficiently, such as appointment search, automatic cancellation notification, attendance tracking, avoid conflict booking, and booking restriction etcetera? Can the system suit for the special requirements for lecturers and college students, for instance setting book deadline, controlling the volume of appointment (set maximum no. of appointments with one session) and automatically calculating appointment duration?

**Customization**
Is the system developed to allow user customization, for example, calendar customization (break, holidays)? Can the report be export to Excel for further analysis or other format?

**Account Setup and Management**
Does the system provide account setup (user registration or setup by administrator) and management? Do the users need login before use to ensure users’ privacy?

**Software Architecture**
Does the system use any software architecture? If so, what type of architecture was applied?

**Server-side programming language**
What scripting language was adopted for developing the system?
Modeling language

Was there any formal modeling language applied in the system during design phase? If so, what is the modeling language?

<table>
<thead>
<tr>
<th>Features</th>
<th>ABS</th>
<th>HSS</th>
<th>ADS</th>
<th>CMM</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available Date and Time Navigation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Book Deadline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic availability Check/avoid conflict booking</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Appointment Confirmation</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Appointment Subject Description</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Booking Restriction (e.g. Open to BSE only)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Control the volume of appointments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatically Calculate each Appointment’s Duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recurrent Appointment</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Automatic Cancellation Notification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Appointment Reminder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Features</td>
<td>ABBUM</td>
<td>HSS</td>
<td>Appointm</td>
<td>CyberM</td>
<td>TimeT</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>--------</td>
<td>-------</td>
</tr>
<tr>
<td>Attendance Tracking</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Appointment Search</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Appointment Lists and History</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Appointments Reports</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Set Available Time</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Customized view</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Customize Calendar (holidays,</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>break etc)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Export (excel / pdf format)</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Registration</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Permission (login)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Access Management(User Privilege)</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Management (Add, Edit, Delete)</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
</tbody>
</table>

**Software Architecture**

<table>
<thead>
<tr>
<th></th>
<th>3-tier</th>
<th>3-tier</th>
<th>3-tier</th>
<th>3-tier</th>
<th>4-tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server-side Programming Language</td>
<td>PHP</td>
<td>JSP</td>
<td>ASP</td>
<td>CGI</td>
<td>JSP</td>
</tr>
<tr>
<td>Modeling Language</td>
<td>own Notation</td>
<td>UML</td>
<td>own Notation</td>
<td>UML</td>
<td>Partial UML</td>
</tr>
</tbody>
</table>

Table 2.1 Features Comparison

**Appointment Management**

All the five systems, Appointment Booking System for McMaster University (ABSUM), Home Schooling System (HSS), Appointment Diary, CyberMatrix and TimeTrade offer the basic features. In Appointment Diary, CyberMatrix and TimeTrade, there are some additional features such as appointment search, automatic cancel notification, attendance tracking. Only Timetrade provides the advanced features like avoid conflict booking, individual/group appointment, and booking restriction. None of the system has the following functions: booking deadline, control the volume of
appointment (set maximum no. of appointments with one session) and automatically calculate appointment duration.

**Customization**

All the five systems allow the users to set available time. CyberMatrix and TimeTrade provide customized calendar. In the AppointmentDiary and TimeTrade, the report can be exported to Outlook or Excel.

**Account Setup and Management**

HSS and CyberMatrix have the registration function. The rest of the systems such as ABSUM, AppointmentDiary, and TimeTrade required the administrator to setup the user account. In AppointmentDiary, CyberMatrix and TimeTrade, the users need to login before use to ensure users’ privacy.

As mentioned earlier, there is no existing web-based appointment management system specified for lecturers and college students. The similar systems in the current market can not fulfill all the user’s requirements, like setting booking deadline, restricting maximum number of student during the consultation time and automatically calculating appointment duration. It is obviously seen from table 2.1 (the first three sections). In order to solve these problems, our system is designed to cover all the listed features to maximum the functionality.

Regarding the modeling language, most of the systems were not using any formal language due to its complexity and time consuming. Only HSS fully utilized UML for design process. UML is a widespread language used by the industries to model system in different areas, like software engineering modeling and business process modeling. However, due to its general purpose which is not specifically designed for web development and gratuitous complex, it is not best suited for web application development. By comparing UML and WebML, WebML are chosen for our project which is web based system. Refer to Chapter 3.1.1 Justification of Using WebML for detailed information.
Most of the systems adopted three-tier architecture, and only TimesTrade applied four-tier architecture. Different server-side scripting languages were used to develop the systems. These two factors, software architecture and server-side programming language are the most important factors affecting software quality such as performance, scalability and maintainability. The following section 2.3.2 Web System Quality Comparison is shown the detailed information.

2.3.2 Web System Quality Comparison

There are three typical quality attributes requirements: usability, performance, and maintainability for web application. The following are the comparisons of existing similar systems based on the three criteria.

**Usability**: Does the system provide an intuitive interface and easily support the users’ tasks? Can the system provide one-to-one personalization? Is the system designed by considering human-computer interactions factors like navigation, feedback, help and use of color and so forth?

**Performance**: Does the system perform adequately and able to sustain the expected workload when running, expressed by parameters response time, number of events processed per second, number of concurrent users, and so on?

**Maintainability**: Can the system or component be modified to correct faults, improve quality or adapt to a changed environment at ease and speed? Is the system design structure amenable to effective modification?

<table>
<thead>
<tr>
<th><strong>Usability</strong></th>
<th>ABS</th>
<th>HSS</th>
<th>ADS</th>
<th>CMM</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Calendar</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Easy to Learn</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Navigation Ease</td>
<td></td>
<td>√</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Provide Feedback</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Help function</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Performance</strong></th>
<th>ABS</th>
<th>HSS</th>
<th>ADS</th>
<th>CMM</th>
<th>TT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>limited</td>
<td>limited</td>
<td>limited</td>
<td>limited</td>
<td>Good</td>
</tr>
<tr>
<td>Maintainability</td>
<td>limited</td>
<td>limited</td>
<td>limited</td>
<td>limited</td>
<td>Good</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>Table 2.2 Web Quality Factor Comparison</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Usability**
All these five systems (ABSUM, HSS, AppointmentDiary, CyberMatrix and TimeTrade) provide visual calendar for easy access. ABSUM and CyberMaxis are easy to learn, but bad navigation. No “back to menu” button and feedback are provided. The users may lose in the navigation of website. Appointmentdiary and HSS have no help function. Timetrade J2EE system provides java technologies that enable the rendering of content to suit different user. But on the other hands, the user need some time to familiarize and learn due to the system complexity.

**Performance**
The first four systems (ABSUM, HSS, AppointmentDiary, and CyberMatrix) used three-tier architecture with different server-side scripting. As in the three-tier architecture, too many functions need to be process within the middle-tier; it results in limitations on the performance aspect.

For examples, in CyberMatrix CGI architecture, the web server need create a new process and terminate at the end of execution for each HTTP request. Process creation and termination slow down the performance.

ABS used PHP, MySQL and Apache web server rather than CGI. Apache 2.0 offers operation in multi-threaded mode. It has better performance, but the use of scripting languages strains a Web server’s capacity due to runtime interpretation. The server need use extra resources to parse each page it sends out.

This performance drawback is also existed in AppointmentDiary which the server-side scripting language is ASP. In addition, Sessions in IIS are persistent, and the memory allocated to a session will not be freed until the session has been terminated or has timed out [35]. So it is often the case that it simply "times out" in the user's browser when running ASP pages. All these factors limit the performance.
The fifth system TimeTrade adopted J2EE four-tier architecture which provides good performance. JSP/Servlets are much higher performance than CGI, due to the in-process nature of the execution and memory sharing between scripts [64]. A JSP/Servlet is an in-process function call, which does not have the overhead of starting a new process.

Maintainability

Three-tier approach which is used by the first four systems does not truly separate the application into specialized, functional layers. It leads to maintenance problems for large applications. Timetrade J2EE four-tier architecture supports maintainability. The important techniques in the J2EE, like separation of concerns, separation of business functions from infrastructure technology help achieve the maintainability.

From the summary listed in Table 2.2, it is clearly seen that the similar systems not only fall shorts in the functional requirements, but also can not fully satisfy the non-functional requirements due to the architecture and techniques used.

The three-tier architecture has significant drawbacks. It does not truly separate the application into specialized, functional layers. It has too many functions grouped into business logic/middle tier. Because of that, it falls short in several key areas, such as inefficiencies and complex in the development cycle, lack of flexibility and scalability.

Four-tier approach further evolves today’s three-tier system architecture. By separating the presentation logic from the business logic, it allows for the separation of concerns between the content to be rendered and the rendering logic. The presentation server solves the problems brought on by the proliferation of devices and content types. The business logic and data can be centrally exposed through HTTP and XML, thereby it allows any device, any where on the Internet to access data and business logic.

In addition to these, by dividing the work between software and user interface developers, four-tier architecture cleanly separates development efforts and drastically
streamlines the development process. It also offers a variety of application services to speed up business transactions and offer additional services.

So far four-tier architecture offers maximum functionality and flexibility in a heterogeneous, Web based environment. Therefore, four-tier architecture is chosen for our web-based appointment management system to achieve the non-functional requirements (usability, performance, and maintainability).

### 2.4 J2EE and .NET platform Comparison

A Web platform is an execution environment for Web applications. The two dominant platforms that are based on the concept of four-tier architecture are J2EE platform and .NET platform.

#### 2.4.1 J2EE

J2EE is Sun Microsystems’s reference standard for enterprise development, first introduced in December 1999. It is an industry standard, not a product.

The J2EE platform, built on Java programming language and Java technologies, is a proven and established platform for distributed computing. It standardizes a number of different technologies to provide a portable environment to support the requirements of large enterprise Web applications.

Some important parts in the J2EE are described below:

**Java: The foundation for J2EE**

Java is an object-oriented language derived from C++, with features that simplify coding such as memory management through garbage collection, no pointers etcetera [24]. The J2EE architecture is based on the Java programming language. Thru bytecode which is a cross-platform intermediary, it enables organizations to write their code once, and
deploy that code onto any platform.

**Java Runtime Environment (JRE)**

The Java Runtime Environment (JRE) consists of the Java Virtual Machine, a just-in-time compiler and some foundation classes. It interprets the bytecode and executes it at run-time.

**Applet components**

Java Applets that run inside an applet container are used to package Java code to run within the Web browser.

**Servlets and JSPs**

Servlets and JSPs are web-tier components that run on the web server. They offer mechanisms for dynamic content preparation, processing, and formatting related to presentation.

- **JavaServer Pages (JSPs):** Generate dynamic content for Web browsers and mobile devices. JSPs are analogous to ASP technology. It provides the capability to build dynamic web pages composed of HTML with embedded dynamic components, take an example; references to Beans.

- **Servlets:** Build control and navigation logic into J2EE applications. It typically follows a Model-View-Controller design pattern in conjunction with JSPs.

**Enterprise Java Beans (EJB)**

The Enterprise Java Beans™ (EJB™) technology is one of the prominent, promising technologies in the J2EE platform. The EJB architecture provides a standard for developing reusable Java server components that run in an application server. Therefore, various application vendors, such as IBM (Websphere), BEA (WebLogic) are able to adopt and implement the EJB specification into their application server products.
There are two types of EJB components or enterprise beans: session beans and entity beans. Session beans are enterprise beans that are suitable for processing or workflow. It is used to model business logic. Entity beans represent passive data objects that are stored persistently in the databases. It is applied when a business component need to be persisted and shared among multiple users.

**Java Database Connection (JDBC)**
It is a standard API to manipulate database resources, and process the input/output via SQL in a vendor-independent manner. Nowadays numerous database vendors have developed drivers based on the JDBC API.

**Java Message Service (JMS)**
Java Message Service (JMS) is another standard API to provide asynchronous messaging capabilities to the J2EE platform. It brings the same kind of standardization to messaging as JDBC brought for databases. By using message-oriented middleware (MOM), it offers vendor independence for point-to-point and publish/subscribe messaging between systems. There are many established message-oriented middleware (MOM) vendors like IBM (MQ Series) and Tibco (Rendezvous) available in the market to provide JMS faces to their products.

**Java Naming and Directory Interface (JNDI)**
It’s an API used to register and look up business components and other service-oriented objects in a J2EE environment. JNDI includes support for Lightweight Directory Access Protocol (LDAP), the CORBA Object Services (COS) Naming Service, and the Java RMI Registry.

**JavaMail**
JavaMail is an API that provides a platform-independent and protocol-independent framework to build mail and messaging applications in Java.

**Remote Method Invocation/Internet Inter-ORB Protocol (RMI/IIOP)**
RMI is a relatively simple protocol that enables Java object to communicate remotely with other Java objects. IIOP is a protocol to communicate with CORBA-compliant clients that have been developed using any language compliant with CORBA.

**Java Connectivity Architecture (JCA)**
It allows Java enterprise applications interface with existing non-Java enterprise applications like SAP so as to fully utilize legacy applications in J2EE architecture.

**J2EE Four Tier Architecture**
By putting the components together, it forms J2EE four-tier architecture as figure 2.12. Each tier provides a specific type of functionality to an application.

**Client tier** (sometimes referred to as Presentation Tier or Application Tier),
The Client tier contains presentation programs that interact with the user and display information from the system. The J2EE platform supports different types of clients, including HTML clients, Java applets, and Java applications.

**Web tier**
The Web tier generates presentation logic and accepts user responses from the presentation clients. Based on the received client request, the presentation tier produces the appropriate response to a client request. In the J2EE platform, servlets and JSPs in a Web container implement this tier.
Enterprise JavaBeans Tier/ Business Tier

The Enterprise JavaBeans Tier is the keystone to J2EE application. It provides the necessary interfaces to the underlying business service components which are typically implemented as EJB components. Because of that, it enables multiple instances of an application to concurrently access business logic and data.

Enterprise information systems (EIS)

This tier is responsible for the enterprise information systems, including database systems, transaction processing systems, legacy systems, and enterprise resource planning systems. It links a J2EE application to resources and non-J2EE or legacy systems using CORBA or Java connectors, referred to as J2EE Connector Extensions.

2.4.2 .NET

The Microsoft.Net, initially release in 2001, is an attempt to tie together applications, development languages, operating systems and data stores into a unified, distributed enterprise platform. It provides as a set of products to build enterprise web application and web services for Microsoft operating systems.

The features of .NET are comparable to those of J2EE. The important difference is that .NET is a product strategy, whereas J2EE is a standard. Please refer to section 2.4.3 J2EE and .Net Comparison for detailed information.

The overall .NET platform architecture can be divided into the following parts:

Visual Studio.NET

Closely associated with the .NET Framework is the main programmer development tool, Visual Studio.NET. It is a single integrated development environment which a variety of languages can be plugged in to build .Net applications. The "standard" Microsoft languages come with Visual Studio.NET are Visual Basic, Visual C++, Visual C# which is semantically equivalent to Java and the scripting languages (VBScript and Jscript). Other languages including COBOL from Fujitsu and Eiffel from interactive software engineering and so on are also available through third parties.
**Common Language Runtime (CLR)**

Thru Common Language Runtime (CLR) and Microsoft Intermediate Language (IL), Microsoft.NET offers language-independence and language-interoperability. All Visual Studio.NET languages can be compiled into IL code which is analogous to Java bytecode. Virtual machine CLR will interpret and translate the IL code. The CLR provides many exciting features such as Garbage collection, polymorphic method resolution, type definitions, and error handling.
ASP.NET
ASP.NET is not just the next version of ASP (Active Server Pages), but it is also a unified web development platform that provides necessary services for developers to build dynamic web based applications and XML web services. ASP.Net pages are compiled and executed in CLR. Therefore, ASP.NET can seamlessly integrate with different programming languages for instance, C# or VB.NET to build web applications easily.

.Net Managed Components
.NET managed components is designed to use the features (such as object pooling, queued components, transaction processing, security, and so on) of the COM+ component services as well as other enterprise technologies like Microsoft® Internet Information Services (IIS) and Microsoft® Message Queue server (MSMQ) by providing managed classes to implement those features.

DCOM
Distributed Component Object Model (DCOM) is based on the Component Object Model (COM), which provides a set of program interfaces allowing COM components (i.e., clients and server) to communicate across networks.

ADO.NET
ADO.Net is the new generation to Microsoft’s ActiveX Data Objects (ADO). It describes data as objects and makes extensive use of inheritance and name spaces to provide a more general solution for data access and data services.

Active Directory
Active Directory first released with Windows 2000 Server is the directory service that lets any object on a network be tracked and located. Active Directory stores information and settings relating to an organization in a central and organized database.
Microsoft Message Queuing (MSMQ)

MSMQ is a messaging infrastructure and a development tool for creating distributed messaging applications. It allows applications running on different servers to communicate in a failsafe manner. Messages is temporarily stored in a queue and sent when conditions permit. This enables communication at different times across heterogeneous networks and systems that may be temporarily offline.

Messaging Application Programming Interface (MAPI)

It is a messaging architecture and a COM based API that provides messaging functions including addressing, sending, receiving and storing messages. MAPI consists of a set of common application programming interfaces and a dynamic-link library (DLL) component.

.Net Four-Tier Architecture

.Net four-tier architecture comprises of Client Tier, Web Tier, Business Tier and the EIS Tier. The four tiers together form the database independent, GUI-based for efficient and rapid application development.

![Figure 2.13 .NET Four-Tier Architecture](image)
Client Tier
The Client Tier provides the visual interface for presenting and gathering data. It can be a standard Web Browser or non-browser interface.

Web Tier
The Web Tier uses ASP.Net to compile the server-side code to one or a few DLL files on the web server. It allows for easy and flexible development of web based interfaces. A number of server components are available for inclusion in the interfaces.

Business Tier
The Business Tier uses .Net Enterprise Services like COM+ Services, Web Services to achieve a scalable and flexible architecture. It can interface with the Web Tier or with other third party applications. The Business objects built into the .Net framework provide ease of development and deployment.

Enterprise Information System (EIS) Tier
The Enterprise Information System (EIS) Tier is the enterprise information infrastructure. It includes the database management systems, Active Directory Services Interfaces, Transaction Processing Monitors and Legacy Enterprise applications and so on.

A .Net application can integrate existing databases and applications with new functionality. It provides seamless business-to-business integration.

2.4.3 J2EE and .NET Comparison based on Technology and Architecture

J2EE is a winner in industry maturity and portability (operation system) support. .NET surpass in its performance, tools and language support. Both platforms are conceived for multi-tier, component-based distributed applications, running in a managed and feature-rich execution environment.
A parallel overview and brief description of the two platforms is given as below:

<table>
<thead>
<tr>
<th>Platform feature</th>
<th>J2EE</th>
<th>.NET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Technology</td>
<td>Standard</td>
<td>Product</td>
</tr>
<tr>
<td>Operating systems</td>
<td>Any</td>
<td>Windows 2000, Windows XP</td>
</tr>
<tr>
<td>Browser</td>
<td>Any</td>
<td>Any (Internet Explorer, if ActiveX components are used)</td>
</tr>
<tr>
<td>Client-side components</td>
<td>Java applets</td>
<td>ActiveX components</td>
</tr>
<tr>
<td>Web server</td>
<td>Any</td>
<td>IIS</td>
</tr>
<tr>
<td>Server-side scripts</td>
<td>Servlet and JSP</td>
<td>ASP.NET and Web Forms</td>
</tr>
<tr>
<td>Middle Tier infrastructure</td>
<td>Enterprise Java Beans (EJB)</td>
<td>.NET Managed Components</td>
</tr>
<tr>
<td>Programming languages</td>
<td>Java</td>
<td>VB.NET, C++, C#</td>
</tr>
<tr>
<td>Interpreter</td>
<td>JRE</td>
<td>CLR</td>
</tr>
<tr>
<td>Distribution Protocol</td>
<td>RMI/IIOP</td>
<td>DCOM</td>
</tr>
<tr>
<td>Distributed Transaction</td>
<td>Java Transaction Server</td>
<td>Microsoft Distributed Transaction Coordinator (MS-DTC) / COM+ Services</td>
</tr>
<tr>
<td>Security</td>
<td>Java Security Services</td>
<td>COM+ Security Call Context</td>
</tr>
<tr>
<td>Asynchronous Components</td>
<td>MDB</td>
<td>Queued(COM+)</td>
</tr>
<tr>
<td>Msg Queue API</td>
<td>JMS</td>
<td>MSMQ</td>
</tr>
<tr>
<td>Email/Msg API</td>
<td>JavaEmail</td>
<td>MAPI</td>
</tr>
<tr>
<td>Relational DB API</td>
<td>JDBC</td>
<td>ADO.NET</td>
</tr>
<tr>
<td>Mainframe DB connectivity</td>
<td>Java Connector</td>
<td>Host Integration Server</td>
</tr>
<tr>
<td>Directory services</td>
<td>Java Naming and Directory Interface (JNDI )</td>
<td>Active Directory Services Interface (ADSI)</td>
</tr>
</tbody>
</table>

Table 2.3 J2EE and .Net Comparison

**Presentation layer: HTML generation**

JSPs and servlets generate HTML for J2EE applications. In .NET, ASP.Net builds and hosts web applications under Microsoft’s Internet Information Server (IIS). JSPs use Java code only. In contrast, Asp.net can use all the languages, for instance, Visual Basic and C#.
**Business logic EJB vs. .NET managed components**

Both the Microsoft and J2EE support the concept of interactions with stateless components that are managed by the middleware layer. In .NET, a component can access COM+ services by deriving from the .NET Enterprise Services. In J2EE, this model is implemented by building stateless session beans.

The EJB specification also includes the concept of entity beans which provide an object-oriented interface to allow developers to access data as objects. There are two main types: container-managed persistence (CMP) and bean-managed (BMP).

In the current technical preview, ObjectSpaces in .NET includes concepts that map to both CMP and BMP. It provides an object-oriented mapping layer to access data.

**Execution engine CLR vs. JVM**

.NET CLR supports multiple languages which are Visual Basic, C#, C++, Jscript, COBOL, FORTRAN, and Perl to use a shared set of components on windows. JVM allows Java bytecodes to run on any platform with a compliant JVM. Thru JVM, J2EE offers complete cross-platform portability.

**Language Support**

J2EE promotes Java-centric computing. All components such as EJB components and servlets must be written in the Java language. Other languages can be bridged into a J2EE solution through web services, CORBA, JNI, or the JCA. However, there will be no support for non-Java EJB development. By way of comparison, .NET supports development all major languages except Java. It is obviously a feature advantage that .NET over J2EE.

**Tools support**

In J2EE, it supports numerous third party tools like WebGain’s Visual Café, IBM’s VisualAge for Java, Borland’s JBuilder and open source-code products. However, these tools are not 100% interoperable, because they do not originate from a single vendor.
Microsoft offers an integrated development tools--Visual Studio .Net. It supports all languages except Java. Visual Studio.NET enables developers to take advantage of .NET’s support for cross-language inheritance.

**Advantages that .NET over J2EE**

Microsoft .NET offers a better integrated, lower cost, and more manageable environment for software development than the Java J2EE platform. According to the test and evaluation, .Net has a much lower price and delivers three times the performance of Java [13].

The learning curve for J2EE developers can be relatively lengthy, especially if EJB is used, due to the complexity of the component architecture. While in .Net platform, Microsoft .NET offers a variety of time-to-market features for instance supports cross-language (Perl, C#, VB) inheritance.

By considering all the above factors and non-functional requirements in our project, .Net is selected.
Chapter 3

Research Methodology

3.1 Methodology

3.1.1 Justification of Using Web Modeling Language [WebML]

The rapid growth and diffusion of the Web are nurturing a novel generation of applications which grow in size and complexity. The development of such applications is a hybrid between traditional Information System development and Hypermedia authoring [39]. It challenges the existing tools and approaches for software production. Currently the development of a Web application is somewhat chaotic and often ad-hoc process lacking systematic techniques and methodologies [22]. As a result, many companies today are experiencing severe problems in the management and maintenance of the web application. Ginige et al. use the term Web crisis to describe this situation [14].

The best solution for it is to put into practice appropriate design methodology that could complement current web technology in an effective way. So far most methodologies used for website design is taken from different sectors for example DB, software engineering, it cannot perfectly meet website’s specified requirements. For instance, data centric methods do not cover the hypertext front-end; OO methods like UML profiles do not capture the essence of web-based system and so on [62]. Please refer to 3.1.2 UML and WebML comparison for detailed explanation.

In response to this need, the W3I3 Project funded by the European Community under the Fourth Framework Program is focusing on "Intelligent Information Infrastructure" and has produced a novel Web modeling language, called WebML [49]. WebML is a design methodology that fully exploits the conceptual modeling approach of software engineering, from ideas to application [50]. It addresses the high-level,
platform-independent specification and targets Web sites that require such advanced features like multi-device access (PCs, PDAs, and WAP phones), the one-to-one personalization (myYahoo, myCDNOW), and evolution management.

The model-driven approach has proved very effective in extending the classical methods and best practices of Software Engineering to the Web [47]. WebML guarantees the model-driven approach which can reduce development efforts (cost and time) and allows a more structured development process.

Besides these, according to the WebML official website and research, so far only few industrial projects such as acer, DEI Web application etcetera are developed with WebML. WebML which is customized for web application design need further promote in the industries.

3.1.2 UML and WebML Comparison
Since WebML is the only modeling language specially designed for web application, the comparison will be discussed between WebML and UML. According to literature review, UML is the only formal modeling language applied for designing process among the existing similar systems. UML is a widespread language used by the industries to model system. However, it is not best suited for web application development. The below are the comparison between UML and WebML:

UML is a widely recognized modeling standard used to specify, visualize, construct and document the artifacts of a system for different areas, like software engineering modeling, business process modeling [43]. It is a general purpose modeling language, which tries to achieve compatibility with every possible implementation language [58] while WebML is specifically developed for modeling web applications.

WebML has support for both modeling web application design and html forms. In the hypertext model, which pages the application consists of, what they contain and how
they are linked are defined [51]. It also shows when data has to be sent between pages to display the content.

These features give both the customer and the developer a good overview of the web application structure and design. Hypertext model which uses data entry units to model the html form is better than any model that UML provides [6].

In addition, UML is considered relatively large and complex with twelve models according to OMG (Object Management Group) while WebML consists of only four models. WebML is a small modeling language in comparison to UML. Because of that, it is easier to understand and less time consuming than UML in the modeling process.

From the above comparison and previous section justification, it is clearly seen that WebML is more suitable for web application development; thereby WebML is chosen for our project.

### 3.1.3 Models of WebML

The specification of a site in WebML is under distinct orthogonal dimensions: data model, Hypertext model (composition model, navigation model), presentation model and personalization model [52]. The figure 3.1 shows brief preview of WebML application.
Data/Structural Model: It expresses the data content of the site, in terms of the relevant entities and relationships [52]. There is no new language proposed for data modeling in WebML. It is compatible with the Entity-Relationship data model, used in conceptual database design, and UML class diagrams which used in object-oriented modeling.

Hypertext Model: It describes the web application structure composed by units, pages, and links. Each different hypertext defines a so-called site view. There are three site views: students’ site view, lecturers’ site view and administrator’s site view in WBAMS. Site view description consists of two sub-models: composition model and navigation model.

Composition model: it specifies which pages compose the hypertext, and which content units make up a page. There are six types of content units to compose pages: data, multi-data, index, filter, scroller and direct units. Data units are used to publish the information of a single object for example an appointment, while the rest types of units are used to represent alternative ways to browse a set of objects like the lecturer’s daily schedule.

Navigation Model: it defines the links between pages and units. Links that carry context information are called contextual links whereas links that have no associated context information are called non-contextual links [59]. For instance, the link between homepage and registration page for students or lecturers is non-contextual link.

Presentation Model: It describes the visual aspects by defining the layout and graphic appearance of pages in a site view. Since WebML specifications can be represented using XML, presentation actually is a document transformation mapping the WebML specification into a page written by a concrete implementation language like ASP.NET [60].
**Personalization Model**: it is considered as the customization features for one-to-one content delivery, like shopping suggestions, list of favorites, and resources for graphic customization. Users and user groups are explicitly modeled in the structure schema in the form of predefined entities called user and group [52].

### 3.1.3 Development Phases of WebML

The WebML approach to the development of web applications consists of different phases. These phases are shown in figure 3.2: Inspired by boehm’s spiral model, and in line with modern methods for web and software applications development [36], the WebML process is applied in an iterative and incremental manner. Various phases are repeated and refined until results meet the application requirements. Such and iterative and incremental lifecycle appears particularly appropriate for the web context, where applications must be deployed quickly (in ‘internet time’) and requirements are likely to change during the development time [36].

![Figure 3.2 WebML Process](image-url)
Requirement Specification /analysis

In the WebML process, requirement specification stage divides into two phases which are requirement collection and requirement analysis. The application requirements which are objectives of the system, its target user group, functional requirement and non-functional requirements are collected by applying fact finding techniques, such as interview, observation, emailing and literature review. The specification of the collected requirements is produced after requirement analysis in this phase.

System Design

In this stage, the best choice is determined from a number of design options based on technical, operational, economic, and time constraints. WebML Models (i.e. Data Model, Hypertext Model, Presentation Model and Personalization Model) and other diagramming techniques known as ER Diagram, structure chart are produced to describe the system.

The system design is divided into three design phases (Data Design, Hypertext design and architecture design) which are applied in an iterative and incremental manner. Hypertext design is the phase of the entire lifecycle that mostly benefits from a conceptual and model-driven approach [48]. Because the functions to be delivered by site views and pages are much easier at the conceptual level with a visual model than at the source code level, it results into a more consistent and qualitative design.

In addition to that, a .Net four-tier architecture suited for our project is designed to best meet the application requirements.

Human Computer Interaction factors and WebRatio which is CASE tool to compose both the Entity-Relationship diagram and the site views of the application are applied to cover the phases of hypertext design.
Implementation:

Implementation is to transform the design into an application running on the .Net four-tier architecture. Visual Studio. Net, IIS and SQL data server are used to implement the system.

Testing and Evaluation

Testing which is iterated throughout the implementation phase involves functional testing and non-functional testing. Testing is broke down into the classical activities of unit testing, bottom-up integration testing, and system testing. Black-box testing and white box testing is adopted in unit test. For the non-functional evaluation, such as usability, performance, and maintainability, the following methods are used, i.e. questionnaire, Microsoft Web Application Stress Tool and software metrics.

Deployment and Maintenance

After the testing and evaluation, deployment is carried out on top of .Net four-tier architecture. Maintenance encompasses the modifications effected after the application has been deployed in the production environment. In a model-driven process like that based on WebML, maintenance and evolution benefits from the existence of a conceptual model of the application[36]. Requests for changes can in fact be turned into changes at the conceptual level, either to the data model or to the hypertext model. in the dissertation, maintenance phase is not discussed, but maintainability of WBAMS is tested in the section 7.1.3.3

3.2 Fact-finding Technique

Fact-finding is the formal process of using techniques to collect information about systems, requirements and preferences. It is also called information gathering or data collection [25]. There are five common fact finding techniques: research, observation, interview, emailing and questionnaire. The first four techniques are applied to help
define user requirements. Questionnaire is adopted to collect the user’s feedback on WBAMS for lecturers and college students.

3.2.1 Literature Review

Literature review is an important technique based on the study of other similar systems or applications. There are five existing similar systems which are web-based appointment booking system for career service, subsystem of web-based home schooling application, AppointmentDiary demo system, CyberMatrix meeting manager and Timetrade system. These systems were selected because of functionality, usage of different scripting language, software architecture and usability issue. The main approaches to do the literature review are Internet browsing and library research.

1. Internet browsing

Through the powerful search engines, for instance, Yahoo and Google (Google Scholar), much information like similar web-based appointment systems, review of technologies, design and development process and so on can be found to help determine the system requirements and the development process. Besides that, many tutorials, articles, even e-book and trial kits are available to download from the internet. The most frequent visited digital library/web site for relevant articles, conference proceedings, and sample chapters are ACM Digital Library, IEEE Xplore, ScienceDirect and SpringerLink. The e-books are available in Books 24x7 and Wrox Press.

2. Library research

There are many books, journals, magazines and relevant thesis in the library of University of Malaya. These materials could either be browsed or borrowed for reference. Mostly much time were spent in browsing the senior’s thesis as they are not allowed to borrow. Besides UM libraries, some bookstores for example MPH bookstore, are good place to find newest version of books relating to technology and methodology.

3.2.2 Observation
Observation is one of the most effective data collection techniques for obtaining an understanding of a system. This technique was adopted because through direct observation, firsthand objective information about how activities are carried out for arranging meeting, tools used, complexity of the work can be obtained. The sources of error (omissions or exaggerations) can be eliminated.

A few lecturers in Department of Computing and Language Center of KBU International College were selected to be observed. KBU international College is one of the most popular private colleges in Malaysia. The new principal encourages computerized systems for college in order to improve efficiency and accuracy.

3.2.3 Interview
Informal interview was used to get the potential and existing users’ (administrator, lecturers and students) opinions of the current process and the requirements of new system. There is not much information on user’s opinions through observation. Besides that, this flexible method was adopted because of time-consuming for formal interview and difficulties to administer questionnaires for acquiring the current way of arranging appointment in a college environment. Some informal interviews with administrator, lecturers and students were taken placed at the canteen or classroom in KBU international College.

3.2.4 Emailing
This approach was used to get the information about the existing similar systems for investigation. Since most of the systems are commercial product, there is no explicit information on their design process in those websites. They only provide contact email addresses which are more related to customers. There are some appropriate replies when emailed about the process or on the design questions.

3.2.5 Questionnaire
Questionnaire is used as a reference for WBAMS evaluation. The wide distribution ensures greater anonymity for respondents, which can lead to more honest responses. Goal-Question-Metric approach has proven to be a particularly effective approach to
select and implement metrics [37]. Through GQM, mixed type (i.e. open questions and closed questions) questionnaire is designed. The below table 3.1 shows the questionnaire based on GQM:

<table>
<thead>
<tr>
<th>Questionnaire</th>
<th></th>
<th></th>
</tr>
</thead>
</table>

### General Information

<table>
<thead>
<tr>
<th>Goal</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>To know the familiarity of computer and internet</td>
<td>1. Which User Group are you in</td>
<td>• Total Number of respondent</td>
</tr>
<tr>
<td></td>
<td>2. What is your level of Computer knowledge</td>
<td>• Type of respondents</td>
</tr>
<tr>
<td></td>
<td>3. How often do you use Internet</td>
<td>• Knowledge level of respondent on computer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Frequency of respondent on internet usage</td>
</tr>
</tbody>
</table>

### System Usability

<table>
<thead>
<tr>
<th>Goal</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>To evaluate the system usability</td>
<td>4. Are the characters on the screen easy to read?</td>
<td>• Mean of respondent’s answer</td>
</tr>
<tr>
<td></td>
<td>5. Is the organization of information on the screen clear?</td>
<td>• Percentage of response</td>
</tr>
<tr>
<td></td>
<td>6. Is the sequence of the screen clear?</td>
<td>• Mode of response</td>
</tr>
<tr>
<td></td>
<td>7. Is the style of the screen consistent?</td>
<td>• Learnability</td>
</tr>
<tr>
<td></td>
<td>8. Is it easy to find the information needed?</td>
<td>• Handling ability</td>
</tr>
<tr>
<td></td>
<td>9. Is the help function and reference material sufficient?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10. Is it easy to learn to operate the system?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11. Can you perform the tasks easily?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12. Are you satisfied with the system interface?</td>
<td></td>
</tr>
</tbody>
</table>

### System Capability

<table>
<thead>
<tr>
<th>Goal</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>To evaluate the completeness of system</td>
<td>13. Does the system response fast?</td>
<td>• Mean of respondent’s answer</td>
</tr>
<tr>
<td></td>
<td>14. Are the system functions sufficient?</td>
<td>• Percentage of response</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Mode of response</td>
</tr>
</tbody>
</table>

### Overall

<table>
<thead>
<tr>
<th>Goal</th>
<th>Questions</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>To summarize</td>
<td>15. Overall, do you satisfy with the system?</td>
<td>• Mean of response</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Percentage of response</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Mode of respondent’s answer</td>
</tr>
</tbody>
</table>

Table 3.1 Questionnaire using GQM

The analysis of the questionnaire consists of the following steps:
1. Prepare the questionnaire and print 40 copies
There are four sections: general information, system usability, system capability and overall with total of 16 questions designed in the questionnaire. There is one or few metrics used to evaluate the response from each question.

2. Demo the system scope and let the user test the system
The lecturers and students in the School of Computing and Language Center of KBU International College were chosen to test the system. The reason to select these two departments is to get balanced range for distributing the questionnaire. Students in language center are pre-university students with relatively little IT knowledge, while people in Department of Computer have strong IT background. Different levels of users involved can ensure the dependability and authenticity of this survey. Before testing, the system scope and requirements were explained to help the users understand the system so as to improve the quality of response.

3. Issue the questionnaire to the users
Each user tested the system individually. The questionnaires were issued during or after their testing process.

4. Collect the response from the users
The testing was carried out during the lab sessions for the students and in the lecturer’s office with the lecturers.

5. Summarize and analyze the results (both report and charts)
After examining each feedback, the questionnaires were sorted out and analyzed. Please refer to 7.1.3.1 Usability Testing for the results.
Chapter 4

System Analysis

The chapter covers the data analysis by applying fact-finding techniques, and requirements specification which are user identification, use case diagram, functional requirements, non-functional requirements, data dictionary specification, site view specification and hardware requirements.

4.1 Analysis of Data Collection
Through fact-finding techniques, much information about systems, requirements and preferences are gathered as the following:

4.1.1 Data Analysis from Literature Review
Based on the literature review, the existing similar appointment scheduling or booking systems cannot meet the user’s requirements because they are not specially designed for lecturers and college students to arrange meeting. Some functional requirements for instance controlling the volume of appointments in the consultation time are not achieved. What is more, as most existing similar systems developed on three-tier architecture without any formal modeling language, there are many limitations on performance, scalability, and maintainability. The existing systems are also more or less lack of usability. From the research, it is understood that there is much space to develop this project in order to fulfill the needs.

4.1.2 Data Analysis from Observation
Currently there are two ways to arrange appointments for lecturers and college students. From observation, most lecturers adopt the method of releasing the consultation time to students for a fixed period. Few lecturers put the sign up sheet on their doors. Before the meeting, the lecturers collect the sign-up sheets and roughly read through how many students and who will attend. After checking, the sign-up sheets are kept for filing.
However, none of the observed lecturers uses the web based or computerized appointment management system to arrange the meeting.

Several problems are found out from observation of the process:
For the first method (using sign up sheet), it is observed that there is no standard sign-up sheet format for booking the meeting. Different lecturers have different styles. It’s often the case that the sign-up sheet is messy due to not enough space for student to fill in, students’ different handwriting, and corrections. Even sometimes the sign-up sheet is lost.

No matter which methods applied, if the appointments have to be cancelled due to unpredicted events like Medical Leave, Emergence Leave, there is no way to inform students. It is noticed that students only came to know the cancellation upon their arrival at lectures’ office.

In addition to that, it is observed that sometimes there is not even one appointment during one consultation session, whereas sometimes so many students come for meeting. Few students may come at the overlapped time. The lecturer handles the meeting one by one unless for group assignment. The rest students need wait either insides the meeting room or lecturer’s office. At times the lecturer exceed the consultation time due to overfull volume of appointments.

Occasionally it is occurred that lecturers can not give answers and need time to refer because there is no preparation for consultation subject. In current approach, no such information can be obtained before the meeting. There is also no attendance tracking/records for consultation session.
Besides the problems, since this web-based system is proposed, the working environment of the lecturers was observed. Windows Operating System is the only system platform for the office computers. Each lecturer has computer or laptop inside his or her office. They use it to surf Internet, send and receive email. Even some lecturers who are good at computer can write the software program.

4.1.3 Data Analysis from Interview
From informal interview, it is found that currently administrator has no information on each lecturer’s consultation section, not to mention about report and management. Most of the lecturers also have no records on their own consultation section. All the interviewees are willing to adapt to the changes if there is web-based system to replace the current way. By studying these users’ feedback, user groups and a ‘wish-list’ of requirements are defined.

Regarding to computer skill level, most of the interviewee defined themselves as moderate computer skill users.

4.1.4 Data Analysis from Emailing
From the email replies, most of the existing systems are not using any formal modeling language to design the system due to time-consuming and complexity. But some kinds of notations were adopted to achieve the design goals. Only few appointment management systems use UML.

4.2 User identification
There are three groups of users in the system.

1. Administrator: Administrator is the person who need control user group and access management. He/She is given the privilege to setup calendar, view appointments and maintain user groups.

2. Lecturers: Lecturers are one of the main user groups in the system. They can update their user profile, set up the consultation time and manage the appointments.
3. Students: College students can register themselves for this system to make appointment with relevant lecturers.

Each group was given a group specification. Please find Appendix A.1 Group Speciation for detailed information.

4.3 Use case Diagram

Use case diagram as shown in figure 4.1 is used to identify the users and functionality of the system. There are three actors: student, lecturer, and administrator. For each actor, there are few use cases to capture the functional requirements. At first, before the users use the system, login is required. In this system, lecturer can setup available slot, cancel appointments, view appointments (report), and update their own information. Students are able to book appointment, view their own appointments including appointment history, cancel appointments if emergency and update their own profile. The cancellation of appointments will automatically inform another party through the important notice. Administrator has the privilege to manage lecturers and students, setup calendar, change his/her password and view the appointments reports. Please refer to Appendix A.2 Use Case Description.
Figure 4.1 Use Case Diagram
4.4 Functional Requirements
There are six modules involved in the functional requirements.

4.4.1 User Access Module
1. Login function:
Username and password are required for users (lecturers, students and administrator) to access the system in order to protect user’s privacy.
2. Registration
Lecturers and students can register themselves online at any time, anywhere before use the appointment management system. The system will create a new user account with different privilege.

4.4.2 User Maintenance Module
1. Maintain User
Administrator can define user groups and access privileges. Lecturer and students accounts can be added/updated/deleted.
2. Update user profile
Administrator is allowed to change his/her password for login. Lecturers and students can also update their contact information and password through WBAMS.

4.4.3 Consultation Module
This module is mainly for lecturers to manage their consultation.
1. Set up consultation time
Lecturers can set up consultation and control the volume of appointments for the consultation time. A maximum number of appointments during the available period can be specified to limit no. of students. Lecturer can also select the booking restriction to students group, for example open to BSC Software Engineering students only. Based on the above setting, the system will automatically calculate each appointment’s duration and allocate on the lecturer’s schedule. Recurrent consultation is allowed to re-occur automatically on a weekly basis, fortnight basis or monthly basis.
2. Manage/Track appointment attendance
The system will keep the appointment status to “OK” once the appointment time is over. If student is absent, the lecturers need to change the appointment status to “absent”. Through this way, the administrator can track the appointment attendance for further action.

3. Cancel consultation
For the lecturer, he/she can cancel the whole consultation section or a particular appointment at any time before the appointment.

4. View personal appointment
Lecturers can check their own appointment schedule in the calendar view and detailed grid view. The system provides important notice section which contains the cancellation of appointment and reminder to notify their upcoming appointment prior to the appointment.

4.4.4 Appointment Module
1. Book appointment
Student can select the available timeslot of particular lecturer in the same faculty to make appointment reservation. The system will automatically check the student’s status according to the booking restriction set by the lecturer. If qualified, the appointment will be confirmed immediately. Besides that, the system does a complete check to avoid appointment conflict. Once appointments are full, the system will change the consultation status to “full”. During booking, student need to specify the appointment purpose and make the reservation at least one day before so that the lecturer can do the preparation.

2. Cancel appointment
Students are asked to cancel the appointment at least one day prior to the appointment. When lecturer login the system, the cancellation message will appear in the important notice.
3. View personal appointment
Students can view their current appointments and appointments history. Similar to consultation module, it provides important notice section which contains the cancellation of appointment and reminder to notify their upcoming appointment prior to the appointment.

4.4.5 Setup Calendar Module
The administrator can set customized calendar (public holidays, break) in this system.

4.4.6 Report Module
Administrator has the right to view all the lecturers’ appointments and absent students list in the report format. The lecturer also can view their appointments list.

Since all the appointments are stored in the database, the users are able to trace back whenever they need to. There is also appointment search function to allow the user to locate appointment quickly.

In addition, the user can easily print the report through PDF format or export the appointment information into Excel.

4.5 Data Dictionary Specification
In WebML, Data dictionary is a list of the most relevant information objects in the application domain. Each entry can be specified by; Name, Synonyms, Description, Sample Instances, Properties, Relationships, Components, Super Concept and Sub Concept. Please refer to Appendix A.3 Data dictionary specification for details.

4.6 Site View Specification
Different group of users have different perspective. Each user group will be provided with at least one site view supporting the functions identified for the group. The administrator is to manage the user group and view appointments reports. The lecturers concern with their own appointment management for instance, set up consultation time, cancel appointments. The students are only interested in their own appointments with relevant lecturers.
Site view specification inputs the list of user groups, use cases and the data dictionary. It facilitates the production of the hypertext model. Please refer to Appendix A.4 Site View Specification for detailed specifications.

### 4.7 Hardware and Software Requirements

Below are the minimum hardware and software specification for the development of this project.

- 600 MHz Pentium III-class processor
- 512 MB RAM
- 1 GHz hard disk space
- Super VGA (1024x768) or higher resolution display with 256 colors
- 12X CD-ROM drive or DVD-ROM drive
- Windows XP/2000
- Microsoft Internet Explorer 6.0 SP1 or later (prerequisite for .NET Framework)

### 4.8 Non functional Requirements

**Usability:** the system need provide an intuitive interface and easily support the users’ tasks, by considering human-computer interactions factors, such as navigation, feedback, help and use of color. It is designed to have features like one-to-one personalization.

**Performance:** As the user expect low latency and will not tolerate the site simply refusing the request, The response time should not exceed the user's thinking time (3 second) and be relatively stable in peak conditions.

**Maintainability:** The system or component can be easily modified through one or few modules without affecting the rest.
Chapter 5

System Design

5.1 WebML Models Design
WebML models, for example, Data model, Hypertext model, and User model (refer to Figure 3.1) are designed using WebRatio. WebRatio is a CASE tool to support the WebML design process. It offers a visual environment for drawing the data and hypertext conceptual schemas [50].

![WebML Models Diagram](image)

Figure 5.1 WebML Models

5.1.1 Data Model / Structure Model
Data modeling is one of the most traditional and consolidated disciplines of Information Technology [48]. In WebML process, it does not propose another data modeling language, but exploits the most successful and popular notation, which is Entity-Relationship (E-R) model as Figure 5.2.
Figure 5.1 shows the system data schema describing information about Faculty, Department, Course, Student, Student Account, Lecturer, Lecturer Account, Appointment, Appointment Status, Consultation Time, its Status, Admin and Calendar. The last two tables Admin which stores the admin user name and password and Calendar that used to record the non-working days have no relationship. The relationship between the first eleven tables is either one to many or one to one, for instance, there are one or many departments within the Faculty, and one lecturer is only allowed to have one lecturer account. Please refer to Appendix B System Design for more information on Table Structure.
5.1.2 Hypertext Model

The overall hypertext structure is defined in terms of units, pages and links, organized into modularization constructs called areas and site views. Please refer to Appendix B.2 for WebML Hypertext Model Notation Description.

Units are the atomic pieces of the actual interface elements delivered to the users. They offer alternative ways of arranging the content which dynamically extracted from the entities and relationships of the data schema [50].

Typically pages comprise several various kinds of units such as index Unit, Data Unit, Entry Unit to gain the desired communication effect. For example, in the figure 5.2, Registration page contain Registration Entry Unit.

Page and units are linked to form a hypertext structure. Through links, one unit/point can be connected to another unit/point in the hypertext, for instance, OK and KO-links capture the concept of operation success and failure.

A site view is a coherent hypertext to serve a well-defined set of requirements. In large applications, site views can be hierarchically decomposed into areas, which are clusters of pages with a homogeneous purpose [50]. There are three site views designed for WBAMS.

5.1.2.1 Student Site View

Figure 5.3 represents student site view which consists of appointment area and student area.
Figure 5.3 Student Site View
Student Site View Description
The home page is presented by default when the user accesses the URL. From it, the user can select and login into a different protected site view, for example student site view. The homepage is preliminary for accessing the other pages.

A landmark page is globally reachable from all the other pages in the same site view or area. In the student site view, the student menu page is the landmark page. It links almost all the other pages.

In the student site view, there are two areas which are appointment and student. Registration page and update student information page are included in the student area. The appointment area contains make appointment page and manage appointment page. Take an example, BookAppointment page, the selection of a lecturer from the lecturer index leads to the grid list showing this particular lecturer’s available slot. The ConsultationTimedetail is built using a very simple core hypertext sub-schema centered on the ConsultationTime entity. Through Appointment Entry Unit, the student can book the appointment in the available slot. If succeed showed by OK link green color, the information will be stored into the database and display the successful information. ManageAppointment page contains two alternative sub pages: CurrentAppointment and AppointmentHistory. CurrentAppointment page is set to be the default page.

5.1.2.2 Lecturer Site View
Similarly like student site view, there are two areas: lecturer and appointment in the lecturer site view. Lecturer area consists of two pages: Registration and UpdateUser Profile. Appointment area includes SetupAvailableSlot, ViewReport and ManageAppointment pages. For instance, in ViewReport page, LecturerID is a Global Parameter. From it, the Particular Lecturer’s Appointment Lists can be retrieved from database and displayed as index unit.
Figure 5.4 Lecturer Site View
5.1.2.3 Administrator Site View
**Administrator Site View Description**

The home page is presented by default when the users access the URL. From it, the user can login into AdminMenu page which is landmark page reachable from all the other pages in the site view.

In the administrator site view, there are four areas: admin, appointment, lecturer and student. Each area comprises two pages. For example, AddLecturer page, through lecturer entry unit, administrator can add lecturer account. If succeed (OK link green color), the information will be stored into the database and display the successful information. Else (KO link red color), the error message will appear in the AddLecturer page.

**5.1.3 User Model**

According to system analysis, there are three user groups: student, lecturer and administrator. Different site views shown in the section 5.1.2 are defined for these three groups of users.

**5.1.4 Presentation Model**

The proposed system is designed to be simple and straightforward in order to eliminate the complexity and provide simplicity to the processes and operation. The human computer interaction (HCI) guidelines based on the books [65, 66] are considered to design the presentation model. The following sections focus on the HCI issues.

**5.1.4.1 Interaction Style**

The interaction style between the system and the user is direct manipulation. WYSIWYG (what you see is what you get) is the style of the screen. Users do not need to learn any extra computational skill to use the system.

The screen layout for lecturer is shown in the below (please refer to appendix B.3 for all screen layouts):
5.1.4.2 Navigation

The navigation of the system is based on the options provided. There are four options in the student menu, for instance book appointments and five options in the lecturers menu. The structure chart for the user website part is displayed as following: Please refer to appendix B.4 and B.5 for more information.

![User Part Diagram]

Figure 5.5 Lecturer Menu Layout

Figure 5.6 Structure Chart for User Website
5.1.4.2 Feedback

Appropriate feedback is necessary for every operation or action selected by the users. This is important, because the users need to know how far they have progressed and to confirm that the system has performed their operation. Take an example, in WBAMS, when the mouse is over a certain button, the help content for the particular function will display.

5.1.4.4 Minimize User Action

Keying is often the slowest input compared with other input methods. Therefore a good design will minimize the number of keystroke required, for instance, once the student select the start time for appointment, the end time field will be added automatically according to the appointment duration which is automatically calculated by the system.

5.1.4.5 Use of Color

Color is the most crucial factor to consider in designing the interface of the system. The use of color in the system is taken careful consideration, including consideration on the partial user. Throughout the system, color is used sparingly to make it more presentable.

The use of color for the screen design is following the contrast rule. The background of the system is white, and the objects in the interface for instance buttons are dark blue. Dark blue leads to a serious inward outlook. The background color of the text is also set to white, and text set in black color. This high contrast combination is good evidence that for some users who are older or partially sighted. Besides it, the use of color will attract the user’s attention and keep the interface in a uniform style.
5.1.4.6 Font
The choice of the font is more likely to affect the presentation of a screen.

The same font type is applied throughout the system to avoid confusion. By doing some research, standard serif or sans-serif fonts, with familiar, easily recognizable characters are best [19]. Therefore serif font, for example Times New Roman is adopted in the interface of the system.

The size of the font depends on the place where it is allocated. On the whole, the font size is 12pt. For the important texts that need to attract the users’ attention, such as the title of the menu, the size of the font is 16 point.

5.1.4.7 Help System and Documentation
As the system is new to the users, it is necessary to provide help function to guide them to use the system. There are separate help function and user manual (see Appendix C.2 User Manual) in the system.

5.2 Software Architecture Design
This section includes goals and constraints of architecture design, comparison between J2EE and .Net on cost, complexity and corporate standard, and WBAMS architecture design.

5.2.1 Goals of Architecture Design
The definition of the application architecture is to ensure the achievement of some of the non-functional requirements i.e. performance and maintainability.

5.2.2 Constraints of Architecture Design
Architecture design is not only a matter of goals, but also of constraints, such as physical, financial and organizational constraints [50]. The following criteria affect decision-making:

- Cost: The application budget is one of the most important factors on the choice of the hardware resources and of the software products.
• Complexity: in principle, it is advisable to select the simpler configurations to set up and maintain. The unavailability or the cost of specialized technical skills may restrict the architecture design [50].

• Corporate standards and infrastructures: the application must be deployed within the organization, which may limit the selection of processors and software products.

5.2.3 Comparison between J2EE and .Net based on cost, complexity and corporate standard

According to literature review, the choice of the architecture is between .Net four-tier architecture and J2EE four-tier architecture. J2EE is a winner in industry maturity and portability (operation system) support. .NET surpass in its performance, tools and language support. The comparison based on the above criteria is shown as following:

Cost
Infrastructure costs are about the same regardless of the platform chosen [15]. If a different application server is chosen in the J2EE world such as BEA’s WebLogic, then .Net is a cheaper application.

In addition to that, according to PayScale.com, .NET resources are 6.7% cheaper than correspondingly skilled J2EE developers. .Net resources offset Java’s industry maturity superiority with a slight cost advantage [15].

Complexity
Come to the complexity, both architectures require high technical skill. The learning curve for J2EE developers can be relatively lengthy, especially if EJB is used, due to the complexity of the component architecture. While in .net platform, Microsoft.NET offers a variety of time-to-market features for instance supports cross-language (e.g. Perl, C#, VB) inheritance.
Corporate standards and infrastructures

According to system analysis, the platform for all the computers in the KBU college environment is Windows. It is not necessary to have portability (operating system) support. Therefore .Net is chosen due to its better performance, relatively simpler configuration and lower cost.

5.2.4 System Architecture Design

By considered the constraints, .Net four-tier architecture is best suit for WBAMS in order to achieve the functional and non-functional requirements at lower cost.

The mapping of our web application into .Net Four-tier architecture is pictorially illustrated in Figure 5.8.
Client Tier
The client tier provides the visual interface for presenting and gathering data. It is typically embedded in a standard Web browser. Any changes taken place for the web application do not result in any alteration at the client side. It makes the system easier to maintain.

Web Tier
Web tier contains the UI elements of the site, and includes all the logic that manages the interaction between the client tier and business tier. It is designed that the server controls and html elements are placed in .aspx file while the presentation logic is separated as .cs file. Three categories of .asp files (lecturer, student and administrator) shown as figure 5.8 are designed according to user groups. After the client clicks on a hyperlink in the .aspx page, an HTTP requests is addressed to the HTTP server, which then route it to the Web Server.

Business Tier
There are two layers: business logic layer and data access layer in the business tier. Both of the layers use the .Net Managed components which provide ease of development and deployment.

The business logic layer that contains appointment component and user component represents the business rules of WBAMS. Appointment component includes various functions relating to appointment management such as count appointments, cancel appointments, cancel consultation, and check appointment time. User object contains the user specific functionality, like lecturer login, student login, update user’s information etc. As shown in figure 5.7, 23 methods are included in user component and 10 methods in appointment component.

Data access layer is to encapsulate the functionality required to talk to the data tier. There is only one component called database component in the data access layer.
ADO.NET that is the most modern Microsoft data-access technology is adopted in
database component to access the database.

**Data tier**
Data tier includes SQL database and stored procedure which contains logic about how to
retrieve and update data. Appointment management database with total 13 tables is
designed according to WebML data model. Instead of SQL text queries, stored
procedure is chosen to use in WBAMS to acquire better performance and
maintainability. There are total 22 stored procedures such as get studentID, check
Time, get holidays etcetera to provide a further layering of code for data access and
manipulation.
Chapter 6

System Implementation

6.1 Development Platform and Tools
The section discusses and justifies the development tools used for WBAMS.

6.1.1 Development Platform
As discussed in literature review and system design, .Net platform was selected for this project to fulfill the functional and non-functional requirements.

6.1.2 Development Tools
Powerful system requires powerful tools. On the whole, this application was developed using Visual Studio.Net and SQL Sever platform, which are widely recognized as the most scalable, secure and robust in the industry.

6.1.2.1 Visual Studio.Net
Closely associated with the .NET Framework is the main programmer development tool, Visual Studio.NET. It is a single integrated development environment (IDE) which a variety of languages can be plugged in to build .Net applications. The "standard" Microsoft languages come with Visual Studio.NET are Visual Basic.Net, J#, Visual C++, Visual C# and the scripting languages (VBScript and Jscript).

In the system development, Visual Studio.NET was used to define the logic that delivers HTML pages to thin client systems in the web tier. It is also to implement business logic and data access layer using C# and then to package them in business tier.
6.1.2.2 Web Server Comparison

In the current market, the web servers that confirmed to compliant with .Net are Abyss Web Server, UltiDev Cassini, Servertec internet server and IIS.

**Abyss Web Server**
Abyss Web Server is a compact web server for Microsoft Windows, Linux, Mac OS X and FreeBSD operating systems developed by Aprelium Technologies. It is claimed that it has low usage of system resources and stability. Abyss web server works on most UNIX based systems and on Win32 systems.

**Servertec Internet Server**
Servertec Internet Server .NET Edition is a small and easy Application/Web Server for serving static and dynamic web pages using Servlets, Java Server Pages (JSP), Common Gateway Interface (CGI) and Server Side Includes (SSI). It is also a feature rich servlet engine that runs with Microsoft's .NET framework.

**Internet Information Services (IIS)**
Microsoft Internet Information Services (IIS) is a set of Internet-based services (including a Web or Hypertext Transfer Protocol server and a File Transfer Protocol server) with additional capabilities for Microsoft Windows operating systems. It is one of the world's most popular web servers in terms of overall websites.

**UltiDev Cassini Web Server** (Asp.Net Development Server)
UltiDev Cassini Web Server is a simple, light-weight, redistributable web server software that can host ASP.NET applications and static HTML sites on Microsoft Windows computers that either don't have IIS installed, or can't run IIS at all, like Windows XP Home Edition.

It supports all ASP.NET features, but not JSP, ASP, CGI, PHP and other non-ASP.NET server-side application server technologies.
<table>
<thead>
<tr>
<th></th>
<th>Abyss</th>
<th>Cassini</th>
<th>Servertec IS</th>
<th>IIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>Free-$59.95</td>
<td>Free</td>
<td>$100-$250</td>
<td>Free</td>
</tr>
<tr>
<td>Vendor</td>
<td>Aprelium</td>
<td>UltiDev LLC</td>
<td>Servertec</td>
<td>Microsoft</td>
</tr>
<tr>
<td><strong>Administration</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUI configuration</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>GUI setup</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Remote Administration</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>SNMP configurable</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>.Net Compliant</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>64-bit port</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Cluster Support</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>IPv6 support</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>J2EE compliant</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.Net Compliant</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>64-bit port</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Cluster Support</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>IPv6 support</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>J2EE compliant</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td><strong>Other features</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiple logs</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Support Microsoft ISAPI</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Visual Servers</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Web-based User interface</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Own API</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Own scripting</td>
<td></td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Support external scripting</td>
<td>√</td>
<td></td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Active Directory Authentication</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Anti-Virus</td>
<td></td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Built-in Firewall</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Built-in Proxy</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Internal user access scheme</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>LDAP authentication</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>SSL (software)</td>
<td>√</td>
<td></td>
<td></td>
<td>√</td>
</tr>
</tbody>
</table>

Table 6.1 Web Server Comparison
By considering the price and features, IIS is chosen for this project. As shown in the above table, IIS supports almost all the listed features at zero cost.

### 6.1.2.3 Data Server

Visual Studio 2005 comes with SQL Server 2005. Microsoft SQL Server 2005 promises a comprehensive, next-generation data management platform with increased scalability, availability, and security of enterprise data and analytical applications [8].

It is commonly used by businesses for medium-sized or larger enterprise databases. With the SQL Server and appropriate infrastructure, many hundreds, or even thousands of concurrent users can be supported without significant performance degradation [5].

### 6.2 Implementation of WebML Models

#### 6.2.1 Data Model Implementation

Data implementation is concerned with the mapping of the conceptual data schema to concrete data sources, which enables the publishing and management of content on the Web [50]. In the proposed system, SQL Server is used to handle the data infrastructure, ensuring optimum performance and reliability. Figure 6.1 is the database diagram built according to Data Model Design.
As shown in Figure 6.1, there are 13 tables namely Faculty, Department, Course, Student, Lecturer, Student Account, Lecturer Account, Appointment, Appointment Status, Consultation Time, Status (consultation), Admin and Calendar in the database. The entire database was developed from inside visual studio 2005 without using any of the SQL server administration tools.

6.2.2 Hypertext Model Implementation

The key ingredients of WebML are pages, units, and links, organized into modularization constructs called areas and site views [50]. The hypertext model is to
specify hyper textual front-end interfaces of a Web application. Each WebML elements is transformed accordingly to Hypertext model.

6.2.2.1 Student Site View Implementation

In the student site view, there are two areas: appointment and student. Registration page and UpdateStudentInformation page are included in the student area. The appointment area contains MakeAppointment page and ManageAppointment page. For example, implementation for part of book appointment page in student site view shown as below:

Figure 6.2: Student Site View Implementation
In the figure 6.2, index unit “LecturerList” is implemented as dropdown list displaying all the lecturers within the faculty. Index unit “ConsultationTime” transformed into GridView control to show the particular lecturer’s available consultations. Rendition of selector “Select Available one” is the button “Select” within GridView control. The data entry unit “Appointment” translated as a set of controls to gather the input for appointment details.

6.2.2.2 Lecturer Site View Implementation

Similarly like student site view, there are two areas: lecturer and appointment in the lecturer site view. Lecturer area consists of two pages: Registration and UpdateUserProfile. Appointment area includes SetupAvailableSlot, ViewReport and ManageAppointment pages.

Figure 6.3 shows the implementation for ManageAppointment page which is one of the most important pages in lecturer site view. There are two alternative pages: Calendar view and Grid View in ManageAppointment page. Grid View is the default page. Get unit “GetLecturerID” retrieves the global parameter LecturerID. In the lecturer site view, the index units, for example ConsultationList and AppointmentList are rendered into GridView control. The calendar unit is transformed to calendar control to display the consultation information of a particular lecturer.
Figure 6.3: Lecturer Site View Implementation

```
SELECT ConsultationTime.ID, ConsultationTime.LecturerID, ConsultationTime.Date, ConsultationTime.StartTime, ConsultationTime.AppointmentDuration, ConsultationTime.StatusID, Status.StatusType FROM ConsultationTime JOIN Status ON ConsultationTime.StatusID = Status.ID WHERE (ConsultationTime.LecturerID = LecturerID) AND (ConsultationTime.Date = GETDATE())
```

```
UPDATE [ConsultationTime] SET [StatusID] = @StatusID WHERE [ID] = @ID
```

```
UPDATE [Appointment] SET [StatusID] = 2 WHERE [ID] = @ID
```

```
Label txtDur = new Label();
double duration = Convert.ToDouble(txtTime.Text);
DateTime startDateTime = Convert.ToDateTime(txtTime.Text);
DateTime appointmentDateTime = startDateTime.AddHours(duration);
lbl.Text = "<br /> + startDateTime.ToString("yyyy-MM-dd") + "

cell.Controls.Add(txtId);
```
6.2.2.3 Admin Site View Implementation

In the administrator site view, there are four areas: admin, appointment, lecturer and student. Each area comprises two pages. The Figure 6.4 shows the implementation of ManageLecturerInfo page included the lecturer area.

The selection of a department leads to the grid list showing the lecturers’ information within the department. Through “LecturerDetail” entry unit which represent as text box within GridView control, the administrator can update the lecturer’s info. If succeed (OK link green color), the information will be stored into the database and display the successful information in the ManageLecturerInfo page.

Figure 6.4: Admin Site View Implementation
6.2.3 User Model Implementation

Different group of user has different perspective of the system. Because of that, the application is designed for three groups of user: student, lecturer and administrator. Figure 6.5 shows the brief graphic representation for site views of these three user groups.

As shown in Figure 6.5, three diverse set of menu and functions are implemented for administrator, student and lecturer based on site views.

6.2.4 Presentation Model Implementation

Building a professional web application involves much more than designing individual web pages. During implementation of presentation model, Master pages and themes are applied to integrate web pages in a complete, unified website. Master pages standardize the layout of the website while themes effortlessly apply groups of formatting settings [32].
6.2.4.1 Master Page Implementation

Essentially, a master page is a blueprint to define the overall layout, such as headers, footers, menus. All the web pages can share the layout based on the master page. Figure 6.6 provides a graphical representation of the master page feature.

Figure 6.6 graphical representation of the master page feature [33]

Three master pages are created for these three groups of user, i.e. lecturer, student and administrator. Take an example, lecturer master page shown in Figure 6.7

Figure 6.7 Lecturer Master Page
The ContentPlaceHolder control is added after the header, footer, menu, and banner are set in the lecturer master page. The ContentPlaceHolder control is to define the content that varies from page to page. Each content page automatically acquires the layout and the content of the linked master page.

6.2.4.2 Theme Implementation

A theme is a collection of related files stored in a subfolder under the site’s /APP_Themes folder like Figure 6.8, which contains the following items:

1. StyleSheet.css file that defines the appearance of html objects.
2. Skin file that defines the appearance of server-side Asp.Net controls.

![Figure 6.8 Application Theme Folder](image)

A premade theme can be applied to web page to effortlessly update the appearance of all the controls it contains, instead of painstakingly formatting the controls on every web page.

6.2.4.2.1 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) files are a standard repository for font and formatting information which can be easily applied to various parts of the site.

The primary benefit of using CSS is to enforce a common look and feel among many pages for easy manipulation and maintenance. Beyond that, the CSS file which is applied at the client side can speed up the loading of web page in the user’s browser. The browser will download it once and then cache it. The pages will just link to that cached instance of the .css file and not contain all the styles again, so they will be much smaller and faster to download [33]. The figure 6.9 shows the partial content of CSS file in WBAMS.
6.2.4.2.2 Skin File

Skin is used to define the look and feel of a web site, similarly to CSS. However, unlike CSS, a skin can override various visual properties that were explicitly set on server controls within a page, for instance, a calendar control.

Skins are applied at server side. Skin definitions are saved in files with the .skin extension as shown in figure 6.10.

6.2.4.3 Usability implementation

Besides master pages and themes, the presentation model implementation also considers HCI guidelines based on the books [65, 66] mentioned in Chapter 5 to achieve usability.
<table>
<thead>
<tr>
<th>Design Specification</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation Ease</td>
<td>The navigation of the system is based on the options provided. For example administrator menu</td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="admin_menu" /> <img src="image2" alt="appointment_menu" /> <img src="image3" alt="lecturer_menu" /> <img src="image4" alt="student_menu" /></td>
</tr>
<tr>
<td></td>
<td>All the options are displayed in the menu. WYSIWYG (what you see is what you get) is the style of the screen. Users do not need to learn any extra computational skill to use the system.</td>
</tr>
<tr>
<td>Provide Feedback</td>
<td>Appropriate feedback is implemented for every operation or action selected by the users. For example, registration</td>
</tr>
<tr>
<td></td>
<td><img src="image5" alt="lecturer_account_registration" /></td>
</tr>
<tr>
<td></td>
<td>The username is occupied. Pls Try Other name.</td>
</tr>
<tr>
<td></td>
<td>As shown above, when the user types in the data, the system checks and provides feedback accordingly.</td>
</tr>
</tbody>
</table>
### Help System & Documentation

There are help function (see the below screenshot) and user manual in the system.

![Help System Screenshot](image)

Frequently Asked Question List

- **1. How to register for this Appointment Management system?**
  - Only KBU Lecturers and Students are allowed to register and use this Appointment Management system for their easy meeting arrangement. For example student, during sign up, StudentID and IC No. are required to validate the student’s status.

- **2. Can I change my personal information, such as Email, Mobilephone No., and password?**
  - Yes. After login, you can update these information.

- **3. How to make appointment?**

### Minimize User Action

Keying is often the slowest input compared with other input methods. Therefore, selection and automatic calculation by the system are implemented.

**Make an Appointment**

- **Subject:**

- **Start Time:** 11/22/2007 15:00:00 PM

- **End Time:** 11/22/2007 2:10:00 PM

After the user selects the start time from drop down list, the end time is displayed automatically.

### Text readability

By considering color and font which discussed in the chapter 5, the text readability can be achieved.

### Table 6.2 Usability Implementation

<table>
<thead>
<tr>
<th>Data tier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data tier includes SQL database and stored procedure which contains logic about how to retrieve and update data. Separating data from business logic and data access into its own tier is one of the most important steps to improve performance and scalability. As data increase, this tier can be move to its own (or a more powerful) machine and</td>
</tr>
</tbody>
</table>

---

**6.3 Implementation of .Net Four-Tier architecture**

This section includes the implementation of .Net four-tier architecture in detail.

**6.3.1 Data tier**

Data tier includes SQL database and stored procedure which contains logic about how to retrieve and update data. Separating data from business logic and data access into its own tier is one of the most important steps to improve performance and scalability. As data increase, this tier can be move to its own (or a more powerful) machine and
eventually to its own cluster of machines. Since SQL database has already discussed in 6.2.1, this session will focus on stored procedure.

### 6.3.1.1 Stored Procedure

SQL statements are stored as stored procedures to acquire better performance and maintainability.

Instead of SQL text queries which may have hundreds of characters, a single procedure with a short name can be called for data operations. This minimizes the network traffic. SQL server also caches the stored procedure execution plan when it is executed at the first time. Because of these, it results in better performance.

In addition, it provides a further layering of code for data access and manipulation. By using them, the data access layer just creates a command that references a stored procedure without SQL code. It enables the modification of stored procedure’s SQL code without any impact to the data access layer. Refer to Figure 6.11.

![Figure 6.11 Stored Procedures](image)

### 6.3.2 Business Tier

The business tier or middle tier is the brain of the application because it manages the application’s business logic. It receives requests from the presentation tier, calls the data tier for information and returns a result to the presentation tier depending on the business logic it contains.

There are two layers: data access layer and business logic layer included in the business tier.
6.3.2.1 Data Access Layer

The data access layer is the code that executes queries to retrieve, update, add, and delete data. ADO.NET that is the most modern Microsoft data-access technology is adopted in database component to access the database. The database component is written in C# to help achieve performance and maintainability.

By using this component, the program can be broken down into smaller, tighter blocks of code for easier maintenance and modification. Furthermore, during performing a long, time-consuming operation, the component can work asynchronously so that other tasks with the web page code can perform first and return to pick up the component result later.

In addition, it can be reused for other web or ordinary Windows applications that built on the database.

![Database Component in the Bin Directory](image)

The database component `component.dll` is placed in the bin directory of web application (see Figure 6.12). This file is automatically overwritten with the most recent compiled version of the assembly every time run the project.
Stateless programming is adopted for this component implementation, though it does not satisfy from an object-oriented perspective. The reason is classes that retain little state information supports high performance. There will be no information retained in memory and fewer server resources are used. If a software or hardware failure occurs, no danger of losing valuable data.

6.3.2.2 Business logic layer

Business objects are developed to enforce business rules in this layer. Generally, business objects are the components including application specific logic.

The business objects are created in App_Code folder within the web application project as shown in Figure 6.14. There are two business objects

1. Appointment object includes various functions relating to appointment management such as count appointments, cancel appointments, cancel consultation, and check appointment time.
2. User object contains the user specific functionality, like lecturer login, student login, updating user’s information etcetera.

![Figure 6.14 Business objects in App_Code folder]

The business logic layer consumes the data from data access layer by using “component.DBUtil” class and exposes it to the UI layer as displayed in figure 6.15.

```csharp
namespace appointment
{
    public class appointment
    {
        public appointment()
        {
        }

        public string countconsultation(string lecturerid)
        {
            return Db.Countconsultation(lecturerid);
        }

        public string countappointment(string studentid)
        {
        }
    }
}
```

![Figure 6.15 Coding of Appointment Business Object]

### 6.3.3 Web Tier

The Web Tier uses ASP.Net to compile the server-side code to one or a few DLL files on the web server. It contains the UI elements of the site, and includes all the logic that manages the interaction between the client tier and business tier. The server controls and html elements are placed in .aspx file while the presentation logic is separated as .cs file which use C# programming as shown in Figure 6.16.
6.4 Implementation of Functions

Based on the design, all the functions like calendar setup, appointment reservation, and recurrent consultation setup etcetera are implemented. In this session, four of the most important and typical functions are discussed.

6.4.1 Make Appointment

Before the student can make an appointment reservation, the system is programmed to check the criteria, such as booking restriction set by lecturer, appointment time to avoid conflict booking, and student’s booking status. Besides that, the system is set to automatically calculate the appointment duration and check whether the number of appointment reaches the maximum appointment volume. If so, the consultation status will be set to “Full”. No students can reserve further appointments for this consultation session.

6.4.1.1 Select lecturer

All the lecturers’ name within the same faculty will be displayed in a drop down list control when the student login. The below is the SQL coding to retrieve the lecturers name within the same faculty:

```sql
SELECT Lecturer.Name, Lecturer.LecturerID
FROM Course INNER JOIN Student ON Course.CourseID = Student.CourseID
INNER JOIN Faculty ON Course.FacultyID = Faculty.FacultyID
INNER JOIN Department ON Faculty.DepartmentID = Department.DepartmentID
INNER JOIN Lecturer ON Department.DepartmentID = Lecturer.DepartmentID
WHERE (Student.StudentID = @StudentID);
```

Figure 6.17 SQL coding for retrieving the lecturers within the same faculty
6.4.1.2 Check Booking Restriction

Before student makes an appointment reservation, the booking restriction which is set by lecturer is checked as in Figure 6.18. For instance, some lecturer set consultation with booking restriction to BSC Software Engineering students only. In that case, students in other courses are not allowed to book the appointment.

```csharp
if (bookingrestriction == "Due to Faculty Student") {
    for (int i = 0; i < volume; i++)
    { 
        DropDownList2.Items.Add(time.AddHours(appointmentduration * i).ToString());
        Label15.Text = appointmentduration.ToString();
        DateTime endtime = Convert.ToDateTime(DropDownList2.SelectedValue.ToString()).AddHours(0);
        Label12.Text = endtime.ToString();
    }
}
else if (bookingrestriction == db.getstudentcourse(studentid)) {
    for (int i = 0; i < volume; i++)
    { 
        DropDownList2.Items.Add(time.AddHours(appointmentduration * i).ToString());
        Label15.Text = appointmentduration.ToString();
        DateTime endtime = Convert.ToDateTime(DropDownList2.SelectedValue.ToString()).AddHours(0);
        Label12.Text = endtime.ToString();
    }
}
else {
    e.Cancel = true;
    Label14.Text = "Error! You are restricted to book this appointment";
}
```

Figure 6.18 Partial coding for checking booking restriction

6.4.1.3 Check Appointment Time

The system is programmed to check the appointment time shown in Figure 6.19 to avoid conflict booking by students. If the particular time period within a consultation is reserved by some student, the rest of students are restricted to book that time period.

```csharp
if (dv.checktime(consultationtimeid, starttime)==1)
    Label3.Text = "This Time already booked by Other Student. Pls select another time.");
else if (dv.checktime(consultationtimeid, starttime) == 0) {
    int studentid = Convert.ToInt32(Session["StudentID"].ToString());
```

Figure 6.19 Partial coding for checking appointment Time

6.4.1.4 Check Student booking Status

Each student is only allowed to book one appointment for each consultation. If the student booked or canceled the appointment in a particular consultation before, then this student is restricted to make the reservation for this consultation again.
6.4.1.5 Calculate the Appointment Duration

Based on the consultation duration and maximum appointment volume set by lecturer, the system can automatically calculate the appointment duration as shown in Figure 6.21:

```csharp
if (db.maxappointment(consultationtimeid) == max)
{
    db.updateconsultationstatus(consultationtimeid);
    return 0;
}
else if (db.maxappointment(consultationtimeid) == -1)
```

Figure 6.22 Partial coding for checking Number of appointment

6.4.1.6 Check Number of appointment for one consultation

The system is implemented to check whether the number of appointments reach the maximum appointment volume specified by lecturer. If reach, this consultation status will be updated to “Full”. Figure 6.22 is the part of the coding for performing this task

```csharp
if (db.maxappointment(consultationtimeid) == max)
{
    db.updateconsultationstatus(consultationtimeid);
    return 0;
}
else if (db.maxappointment(consultationtimeid) == -1)
```

Figure 6.22 Partial coding for checking Number of appointment

6.4.2 Setup Consultation

The lecturer can setup recurrent consultation time with booking restriction and maximum number of appointments.

6.4.2.1 Calendar

The calendar is designed for lecturer to select the consultation date easily. The holidays which set by administrator are displayed in red color in the calendar without selection option as shown in Figure 6.23:
Besides disabling selection of holidays, the calendar also disable the selection of past days and weekdays to avoid lecturer accidentally set invalid consultation date.

```csharp
if (e.Day.IsWeekend || e.Day.Date < DateTime.Now) 
{
    e.Day.IsSelectable = false;
}
```

### 6.4.2.2 Recurrent Consultation

The lecturer can setup recurrent consultation in a weekly basis, fortnight basis or monthly basis. After the lecturer select the recurring times and recurrent basis, the consultation will be added into the database accordingly. For example, if the lecturer wants to set recurrent consultation in weekly basis for three times, there will be four consultation records with weekly increment stored in the database.

```csharp
protected void DayKender(DataTableReader myread) 
{
    while (myread.Read()) 
    { 
        if (e.Day.Date.ToString() == myread["NotWorkingday"].ToString()) 
        { 
            e.Day.IsSelectable = false;
        }
    }
}
```
6.4.3 Manage Appointment (Lecturer)

There are two types of view: grid view and calendar view created for performing this function.

6.4.3.1 Cancel Consultation in Grid view

The system has the ability to check whether there are any reserved appointments for a particular consultation time. If got, the system will automatically cancel all the appointments within when the lecturer confirms to cancel the consultation period.

Figure 6.26 shows the coding in the business logic layer

```csharp
public int cancelconsultation(int consultationtimeid)
{
    component.DBUtility db = new component.DBUtility();
    DataTableReader myread = db.getappointmentid(consultationtime
myread.Read();
    if (myread.HasRows.ToString() == "True")
    {
        if (db.updateappointmentstatus(consultationtimeid) == 0)
    
    Figure 6.26 Partial coding of canceling consultation
```

6.4.3.2 Display Consultation information in Calendar View

New labels are created in the calendar to display the consultation information. The coding is shown as following:

```csharp
if (e.Day.Date == Convert.ToDateTime(myread["Date"],ToString())
{
    Label lbl = new Label();
    double duration = Convert.ToDouble(myread["Duration"]);
    DateTime starttime = Convert.ToDateTime(myread["StartTime"]
DateTime endtime = starttime.AddHours(duration);
    lbl.Text = "<br />" + starttime.ToString() + "-
    e.Cell.Controls.Add(lbl);
```

Figure 6.27 Coding of Displaying Consultation Information

6.4.4 View Report (Lecturer)

Lecturers can view their appointment list in the report format and export into Excel or PDF file for further analysis (see Figure 6.28). The report is implemented as a separate file (report1.rdlc) connecting database through object data source.
6.5 Implementation of Other Features

This section describes the implementation of other features like exception handling, view state management and caching.

6.5.1 Exception Handling

Exceptions are the modern way of intercepting and handling runtime errors in object oriented languages [16]. When a runtime error occurs, the execution is interrupted, and an exception is generated. The structured exception handling is used throughout the web tier and the business tier (business logic layer and data access layer). The code is wrapped in the block structure “try” “catch” “finally” as shown in figure 6.29:

```plaintext
try
{
    conn. Open();
    cmd1. ExecuteNonQuery();
    return 0;
}
catch
{
    return 1;
}
finally
{
    conn. Close();
}
```

Figure 6.29 Sample code for Exception Handling

The “try” statement enables error handling. When an error is detected, the execution is immediately passed to the “catch” block. At the end, the “finally” block is executed no matter what happens. This guarantees to execute and release resources even if an unrecoverable exception halts the application. Besides that, “finally” block is used to close a database connection. This is useful because open connections consume resources on the database server, and it can also keep database resources locked, which may cause problems for other concurrently running database activities.
6.5.2 View State Management

View State is used to maintain the state of the web page across postbacks. In other words, when the user performs any action that triggers a postback event, the page persists changes the state after the event handler method executes at the server.

6.5.2.1 Problems on View State

By default, the view state is enabled for all server controls. The problems are that the view state is transferred on every request between the client and the server. It also can reach a considerable size, especially for feature-rich pages that contain a large number of controls. These may cause a lot of network traffic.

6.5.2.2 Solutions to The problems

By considering these problems caused by view state, some of the controls are set to false so as to save processing time and speed up the download process (see Figure 6.30).

```
<table style="width: 100%" 
<tr><td style="width: 100%">
<asp:Image ID="Image1" runat="server" ImageUrl="/images/word2.JPG"
 Width="100%" BorderStyle="None" EnableViewState="False" /></td></tr></table>
```

Figure 6.30 Code of Disabling View State

6.5.3 Caching

Caching consists of temporarily storing resources in a fast access location, for later retrieval; it is a consolidated practice in computer and software architectures [50]. In the system, output caching is applied for better performance.

6.5.3.1 Benefits of Caching

With output caching, the HTML content generated from dynamic pages or controls is cached. When the same page or control is requested again, the HTML output is sent directly from the cache, instead of being executed again. The time that required to run the page and its code is saved.
6.5.3.2 Caching Implementation

The OutputCache directive at the top of .aspx file, as shown in figure 6.31:

```xml
<%@ Page Language="C#" MasterPageFile="~/Lecturer.master" AutoEventWireup="false"%>
<%@ OutputCache Duration="10" VaryByParam="None" %>
<asp:Content ContentID="Content1" ContentPlaceHolderID="Content1" Runat="Server"/>
```

Figure 6.31 Code of Disabling View State
Chapter 7

Evaluation and Testing

This chapter discusses the strategies used to test and evaluate the system and the results. Testing which are iterated throughout the implementation phase involves functional testing and non-functional testing. It is broken down into the classical activities of unit testing, bottom-up integration testing, and system testing. Black-box testing and white-box testing were adopted in unit test. For the non-functional evaluation, such as usability, performance, and maintainability, the following methods were used, i.e. questionnaire, Microsoft Web Application Stress Tool and software metrics.

7.1 Testing

Testing which can be considered as the process of verification and validation is the critical element to the successful development of WBAMS. It is divided into three activities of unit testing, integration testing, and system testing.

7.1.1 Unit Testing

Unit testing is the first stage of WBAMS testing. It focuses on one unit derived from a program module that performs a specific function. Two types of testing strategies which are black box testing and white box testing were used in this stage. Please refer to appendix C.1 Unit testing for WBAMS. It includes a series of test cases for each individual test section with a specific objective.
7.1.1.1 Black Box Testing

Black box testing was used to test software interface. It can demonstrate whether software functions are operational, the input is properly accepted, and output is correctly produced.

7.1.1.2 White box testing

White box testing provides test cases that exercise specific sets of condition and test logical paths through the software. The status of the program may be examined at various points to determine if the expected or asserted status corresponds to actual status. A directed graph can represent the control flow of the program or module. The detailed information is described in 7.2.1 Cyclomatic Complexity.

7.1.2 Integration testing

After the unit testing, integration testing took place to assess whether a set of modules or programs works together without error. Bottom up integration testing was adopted in WBAMS to ensure that the interface and linkages between different parts of the system work properly. The components were integrated by moving upward through the hierarchy as shown in Figure 7.1.
7.1.3 System testing
The last phase of testing is system testing. System testing is the testing of the entire integrated system to evaluate the system's compliance with its specified requirements (functional and non-functional requirements). It is used to uncover WBAMS’ limitations and to measure its capabilities.

7.1.3.1 Usability testing
Usability testing is to determine the effectiveness of the interface. The questionnaire which applied the Goal-Question-Metrics (GQM) approach was used to collect the user’s options on the system, especially interface.

The lecturers and students in Department of Computing and Language Center of KBU International College were selected to test the system. The reason to choose these two departments is to get balanced range for distributing the questionnaire. Students in language center are pre-university students with relatively little IT knowledge, while people in Department of Computer have strong IT background. Different levels of users in terms of PC knowledge involved can ensure the dependability and authenticity of this survey. The testing was carried out during the lab session for the students and in the lecturer’s office with the lecturers. Each chosen user tested the system individually and filled the questionnaire.

There were 40 users involved in this testing which conducted from 16th July 2007 to 20th July 2007. 28 of them are students and 12 are lecturers. 4 lecturers have advanced level of IT knowledge and 2 lecturers have little knowledge on computer. While 25 students rate themselves as moderate user and the remaining 3 students have little knowledge on computer. In addition, 8 lecturers access the internet 3 to 6 times per week and the rest 4 of them utilize it everyday. For the students group, 12 out of 28 students use internet everyday and the rest 16 students surf internet 3-6 times per week. Figure 7.2 shows the respondents’ familiarity on computer knowledge and internet.
Figure 7.2 Bar chart for respondents’ Familiarity of computer and Internet

The results collected for system usability section are analyzed below:

<table>
<thead>
<tr>
<th>System Usability</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Mode</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>A The characters on the screen are easy to read</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>16</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>B The organization of information on the screen is clear</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>17</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>C The sequence of the screen is clear</td>
<td>L</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>12</td>
<td>14</td>
<td>5</td>
</tr>
<tr>
<td>D The style of the screen is consistent</td>
<td>L</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>E It is easy to find the information I needed</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>F The help function and reference material is sufficient</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>15</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>G It is easy to learn to operate the system</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>12</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>H I can perform the tasks easily</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>I I am satisfied with the system interface</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7.1 Analysis of data on system usability

Note: L- Lecturer S- Student
The above figure 7.3 displays the mean response on the set of usability questions. It clearly shows that the response was above average, which the value ranges is from 3.5 to 4.5. The below figures 7.4 and 7.5 are lecturers and students’ opinions on system interface. Most of them chose either agree or absolutely agree as their views. Therefore from the survey it is proven that overall the system fulfills the usability requirements.

**Lecturer's Opinion on System Usability**

![Figure 7.4 Bar chart for Lecturers’ Opinion on System Usability]
Students' Opinion on System Usability

![Bar chart for Students' Opinion on System Usability](image)

Questions

Figure 7.5 Bar chart for Students’ Opinion on System Usability

Table 7.2 shows the results for system capability from the questionnaire.

<table>
<thead>
<tr>
<th>System Capability</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Mode</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>A The system responds fast</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>13</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>B The system functions are sufficient</td>
<td>L</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7.2 Analysis of data on system capacity

Opinions on System Capacity

![Bar chart for users’ Opinion on System Capacity](image)

Figure 7.6 Bar chart for users’ Opinion on System Capacity
As shown in Table 7.2 and Figure 7.6, 67% lecturers and 72% students agree or absolutely agree on fast response and sufficient functions. None of them selected absolutely disagree.

So it is concluded that still there is room for improvement, but overall the system has good performance and its functions are sufficient.

The final section about the overall options on the system is displayed as below

<table>
<thead>
<tr>
<th>Overall</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Mode</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am satisfied with the system</td>
<td>L</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>12</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 7.3 Analysis of data on overall system

67% of lecturers and 75% of students satisfied the overall system. None of them choose absolutely disagree. Thereby, overall the system achieves the requirements from the users’ perspective.

7.1.3.2 Performance testing

The performance testing for WBAMS is to investigate the system’s performance information (i.e. throughput and response time) about this web application to the overloaded condition on a slow machine. Microsoft's Web Application Stress (WAS) Tool was adopted in the stage to simulate multiple users requesting pages from a web site.
The test tried to run as many hits as possible until the system’s CPU processing became 100%.

The below are the PC configuration running WAS against WBAMS:

- Intel Pentium 4 CPU 1300 MHz
- 640 MB RAM
- Windows XP Professional Version 2002 SP2

As the number of threads increase, the throughput number grows. When the number of threads reaches 100 and number of hits nearly 30,000 in 3 minutes period (see figure 7.8), the CPU usage come to the peak 100% as shown in Figure 7.9
The reports generated by WAS are shown as below

![Image](image1)

<table>
<thead>
<tr>
<th>Overview</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Run on:</td>
<td>8/3/2007 2:07:07 AM</td>
</tr>
<tr>
<td>Run length:</td>
<td>00:03:00</td>
</tr>
</tbody>
</table>

Web Application Stress Tool Version:1.1.293.1

<table>
<thead>
<tr>
<th>Number of test clients:</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of hits:</td>
<td>29550</td>
</tr>
<tr>
<td>Requests per Second:</td>
<td>163.91</td>
</tr>
</tbody>
</table>

Socket Statistics

<table>
<thead>
<tr>
<th>Socket Connects:</th>
<th>29630</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Bytes Sent (in KB):</td>
<td>7409.50</td>
</tr>
<tr>
<td>Bytes Sent Rate (in KB/s):</td>
<td>41.09</td>
</tr>
<tr>
<td>Total Bytes Recv (in KB):</td>
<td>99953.01</td>
</tr>
<tr>
<td>Bytes Recv Rate (in KB/s):</td>
<td>554.42</td>
</tr>
</tbody>
</table>

Figure 7.10 Throughput for Performance testing

The throughput numbers shows the total number of hits and how many requests the Web server processed per second. In this example, 30 thousands links were served with an average of almost 164 a second. That is impressive for such low configuration PC that WBAMS was run on.

![Image](image2)

<table>
<thead>
<tr>
<th>Page Summary</th>
<th>Page</th>
<th>Hits</th>
<th>TTFB Avg</th>
<th>TTLB Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET /default.aspx</td>
<td>2995</td>
<td>537.43</td>
<td>538.23</td>
<td></td>
</tr>
<tr>
<td>GET /lecturermenu.aspx</td>
<td>2992</td>
<td>542.64</td>
<td>545.34</td>
<td></td>
</tr>
<tr>
<td>GET /registration.aspx</td>
<td>2972</td>
<td>545.13</td>
<td>547.90</td>
<td></td>
</tr>
<tr>
<td>GET /help.aspx</td>
<td>2970</td>
<td>556.22</td>
<td>559.22</td>
<td></td>
</tr>
<tr>
<td>GET /manageconsultation.aspx</td>
<td>2964</td>
<td>557.57</td>
<td>558.38</td>
<td></td>
</tr>
<tr>
<td>GET /manageappointment.aspx</td>
<td>2951</td>
<td>541.15</td>
<td>544.70</td>
<td></td>
</tr>
<tr>
<td>GET /setupconsultation.aspx</td>
<td>2944</td>
<td>552.13</td>
<td>554.11</td>
<td></td>
</tr>
<tr>
<td>GET /makeappointment.aspx</td>
<td>2930</td>
<td>541.60</td>
<td>544.39</td>
<td></td>
</tr>
<tr>
<td>GET /shelp.aspx</td>
<td>2921</td>
<td>545.04</td>
<td>547.38</td>
<td></td>
</tr>
<tr>
<td>GET /studentmenu.aspx</td>
<td>2911</td>
<td>541.71</td>
<td>545.20</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.11 Page Performance Summary

As seen in Figure 7.11, the response time in milliseconds for each page can be calculated through TTFB (Total Time the first byte is received) and TTLB (last byte is received). It is calculated that all the pages are returned within 10 milliseconds which is much faster than accepted rate (3 seconds) with this load.

From the above test, it is clearly shows that the system achieves the performance requirement.
7.1.3.3 Maintainability testing

One of the major concerns in Software Engineering today is the maintenance process, as maintenance has a staggering financial impact on the software industry. A typical study shows that on average sixty percent of human resources in software organization is spent on software maintenance, and forty percent on the development of new applications. Therefore maintainability is considered as one significant non-functional requirement here.

The maintainability testing was carried out to use appropriate metrics to check whether WBAMS satisfy that requirement. Software metrics can provide an insight into the software and the processes used to maintain it. The detailed metrics and calculation are shown in section 7.2 software metrics.

7.2 Software Metrics

Over the last few years, the field of software metrics has gained much popularity. It has been widely advocated as a fundamental element of an engineering approach in planning and controlling software development, especially in Object oriented software development [20]. As Object Oriented requires a different approach from traditional functional decomposition and data flow development methods, the OO metrics is important and necessary to provide dependable guidelines [31]. In this session, Cyclomatic complexity, Weighted Methods per Class using Cyclomatic complexity, coupling between object class and response for a class are discussed because these factors are directly or indirectly affect the maintainability which is one of the non-functional requirements.

7.2.1 Cyclomatic Complexity

Cyclomatic complexity, which developed by Thomas McCabe is one of the most popular metrics used in software development. It measures the number of linearly independent paths through a program. The formula is shown as below:
Cyclomatic Complexity (cc) = E - N + 2

Where E is the number of edges in the flow graph and N is the number of nodes in the graph.

As Cyclomatic complexity has enormous impact on the testability and maintainability of code, it was selected to calculate WMC in order to predicate how many number of tests for a class and whether the class is risky. The number of tests for a specific method is equal to the CC measure of the method. Therefore, if the CC value increases, it means more risky and more effort to test. For example, cancel consultation method in database component shown in Figure 7.12.

```java
public int cancelconsultation(int consultationtimeid) {
    component.DBUtility db = new component.DBUtility();
    DataTableReader myread = db.getAppointmentId(consultationtimeid);
    if (myread.HasRows.ToString() == "True") {
        if (db.updateAppointmentStatus(consultationtimeid) == 0) {
            return 0;
        } else {
            return 1;
        }
    } else {
        return 2;
    }
    myread.Close();
}
```

Figure 7.12 Flow Graph for Cancel consultation coding

Based on the above flow graph, the cyclomatic complexity (cc) for cancel consultation method in business tier calculated as below:

\[ CC = 8 - 7 + 2 = 3 \]

There are total 136 methods with coding in WBAMS. It is calculated that the maximum CC in WBAMS is add consultation method which the value is 9 (see table 7.4). The CC value for most of the methods is 1. Please refer to table 7.5 and figure 7.13.
Various authors and studies have suggested that a cyclomatic complexity value of 10 or higher for a particular method is considered complex [12]. The following table 7.6 summarizes the impact of CC values.

<table>
<thead>
<tr>
<th>Cyclomatic Complexity</th>
<th>Risk Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>A simple and low risk program</td>
</tr>
<tr>
<td>11-20</td>
<td>more complex, moderate risk</td>
</tr>
<tr>
<td>21-50</td>
<td>complex, high risk</td>
</tr>
<tr>
<td>51+</td>
<td>Most complex and highly unstable method</td>
</tr>
</tbody>
</table>

Table 7.6 Impact of Cyclomatic Complexity Values [44]
From the above calculations, it is concluded that WBAMS which the maximum CC value is less than 10 is low risk.

### 7.2.2 Weighted Methods per Class (WMC) using Cyclomatic Complexity

Consider a Class C1, with methods M1,... Mn that are defined in the class. Let c1,... cn be the complexity of the methods. Then:

\[ WMC(C) = \sum_{i=1}^{n} c_i \]

The WMC metric is the sum of the complexities of the class methods. It is a predictor of how much time and effort is required to develop and maintain the class [4]. Because WMC measures both the number of methods and their complexities, it has been proven to be useful in predicting maintenance and testing effort [29]. The larger the number of methods in a class, more time and effort are required to develop and maintain the class.

There are total 30 classes in WBAMS. Take an example, manage consultation class shown as Figure 7.14. There are six methods which is calendarview, gridview, selectchange, dayrender, updating, show in class manage consultation.

```java
public partial class manageconsultation : System.Web.UI.Page
{
    protected void Calendarview(object sender, EventArgs e)
    {
        Panel1.Visible = false;
        Panel2.Visible = false;
        Panel3.Visible = true;
    }

    protected void GridView(object sender, EventArgs e)
    {
        Panel2.Visible = false;
        Panel1.Visible = true;
    }

    selectchange(object sender, EventArgs e)
    Calendarview(object sender, EventArgs e)
    DayRender(Object source, DayRenderEventArgs e)
    GridView(object sender, EventArgs e)
    selectchange(object sender, EventArgs e)
    Show(object sender, EventArgs e)
    Updating(object sender, GridViewEventArgs e)
}
```

**Figure 7.14 Coding for Manage consultation method**

The WMC using Cyclomatic Complexity is calculated as

\[ WMC = Cc(gridview) + CC(calendarview) + cc(dayrender) + cc(selectchange) + cc(show) + cc(updating) = 1 + 1 + 3 + 1 + 1 + 3 = 10 \]
Most classes in WBAMS have a WMC of less than 20. There is only one class which is the Database component with highest WMC is 89. According to the study done by Linda H. Rosenberg [31], the class with WMC that is greater than 100 is required for inspection and revision. So WBAMS is considered as passed.

<table>
<thead>
<tr>
<th>WBAMS</th>
<th>Median</th>
<th>Mode</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMC</td>
<td>8</td>
<td>5</td>
<td>89</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.7 WMC Analysis

7.2.3 Coupling between objects

This measure counts the number of other classes to which the class is coupled [37]. It is useful to determine how complex the testing of various parts. The larger the number of couples, the higher the sensitivity to changes in the design, and therefore maintenance is more difficult.

In four tier architecture, layer n may only use the services of layer n-1. Because of that, the classes in the presentation layer are only coupled with business layer which just have 3 classes. Therefore, the maximum number of CBO is 3.

<table>
<thead>
<tr>
<th>WBAMS</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBO</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 7.8 Coupling between objects Analysis

In WBAMS, most of the classes are self-contained. The Maximum CBO is 3. Hence, it is concluded that classes in WBAMS is relatively easy to maintain.

7.2.4 Response for class (RFC)

RFC captures the size of the response set of a class. It can be calculated as the number of local methods plus the number of methods called by local methods [37]. It is indirectly indicator of maintainability. The larger the number of methods that can be invoked from class, the greater the complexity of the class, and hence more difficulty to maintain. For instance, registration class in WBAMS as Figure 7.15:
There are four local methods (lcheckavailable, scheckavailable, sregister, lregister) in the registration class. Each method calls one other method from the user component. The calculation is

\[ RFC = 4 + 4 = 8 \]

In WBAMS, RFC for most classes is less than 10. The only one outstanding class is the class in data access layer with highest RFC value 65. Similar to WMC, if the RFC value is bigger than 100, the revision is needed. Here WBAMS with maximum RFC value 62 is considered as pass.

<table>
<thead>
<tr>
<th>WBAMS</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC value</td>
<td>65</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.9 Response for class Analysis

From the above testing using this suite of metrics, it is concluded that WBAMS achieve the maintainability requirement.
Chapter 8

Conclusion

This chapter summarizes the results of WBAMS and how this work met the requirements that have been described in chapter 7. Besides that, this chapter also provides some references for future enhancements.

8.1 System Strengths

The developed WBAMS turns the traditional appointment management approach into web based system by applying Web Modeling Language (WebML) and .Net four-tier architecture so that students and lecturers can arrange meetings in an effective and efficient way. It has various strengths. Some of the major strengths are described as following;

1. Functionality
   Compared with the existing similar systems for other services, WBAMS fulfills the specific users’ functional requirements. Thru applying fact-finding techniques, especially interview and research, a ‘wish-list’ of requirements are defined. WBAMS supports all the wanted features which are mentioned in Table 2.1 (Functionality Comparison) such as setting recurrent consultation, automatically calculating appointment duration, restricting the maximum appointment volume for one consultation and so on.

2. Usability
   WBAMS provides a simple and user-friendly interface. The user can perform their tasks easily. It is designed by considering HCI issues like font, color, navigation, feedback. During implementation, master page and themes were used to keep the interface consistent and to integrate web pages in a complete, unified website. Through questionnaire in Chapter 7, the result has proven to have satisfied usability.
3. **Performance**

It is proven through performance testing that WBAMS has achieved the performance requirements even under overloaded conditions. The use of four-tier architecture, Asp.net, and C# helps to improve performance. Further more, the implementation of View state management and Caching also enable WBAMS fast response.

4. **Maintainability**

By applying .Net four-tier architecture and WebML, WBAMS achieves the maintainability. .Net Four-tier architecture offers maximum functionality and flexibility in a heterogeneous web based environment. While WebML guarantees the model-driven approach which can reduce development efforts (cost and time) and allows a more structured development process.

8.2 **System Limitations**

Though it is proven that WBAMS has met the user requirements, there are still some limitations on the current version of WBAMS. So far WBAMS has not yet applied in the real environment. The limitations were collected from users’ feedback during the testing.

1. **Limitation on the methods of informing cancellation**

Currently, there is only one way to inform the cancellation, which is through notice section provided in WBAMS. When the users login to the system, they will be prompted if any cancellation. This method has limitations. The user may not know the cancellation if he/she does not login this system again before the appointment.

2. **No Timetable integration**

Each system has its own boundary and limitations. In the current version of WBAMS, timetable is not integrated in the system. The system is purely for consultation management. The users need refer their timetables to setup consultation and make appointments in order to avoid time conflict.
8.3 Problems faced
From the initial stage till the closing project, some of the tasks are very challenging and takes time to resolve. The following are some of the main problems encountered.

1. Difficulties in data collection
During data collection, many problems were faced, such as some people were unwilling to be interviewed or fill the questionnaire due to the time factor, few replies from email, insufficient information can be found and so on. To solve these problems and get sufficient data, many fact-finding techniques are adopted to gather the information.

2. New to WebML
Since Web Modeling Language (WebML) is new disciplined design approach for web application, everything is needed to learn from the beginning. Much time was spent to find materials and do comparison.

3. Unfamiliar to .Net four tier architecture
There was not much knowledge on .Net four-tier architecture that is relatively new compared with J2EE architecture. Because of these, there was some confusion on the middle tier of .Net architecture. It is not like J2EE, where the middle tier is clearly stated (Enterprise Java Bean). Much effort was put to get the understanding.

4. Unfamiliar to programming language (Asp.net and c#)
At the initial stage of development, lot of difficulties were encountered, like uncertain about how Asp.net works, how to use some advance options and how to implement in four tier architecture. It is found that some books are really useful to help study and master from the beginner to advanced level.
8.4 Future Enhancement
The following are some of the features which can be considered in the near future.

1. SMS Cancellation Notifications
The system can be enhanced to have SMS cancellation notification. It could boost the system to the next level of appointment management. Wireless technology will be used so that users can use mobile devices to make the tasks easier and on the move.

2. Calendar and Timetable integration
Timetable can be integrated into the system. Through timetable, WBAMS provides convenience to the user for manage appointments and avoid time clash.

3. Improve System interface [3D]
More meaningful 3D images, animation can be added to make WBAMS more attractive and interactive. 3D interface may also be considered as it is not going to affect the business layer.

8.5 Summary
The whole process of this dissertation is a journey of exploring. WBAMS has been successfully designed and developed by applying the latest methodology WebML and .Net four-tier architecture. WebML has support for both modeling web application design and HTML forms. The Hypertext model in WebML which use data entry units to model the html form is better than any model that UML provides. However, WebML does not support use case diagram which shows the interaction between the application and the users. During WBAMS design, use case diagram that is part of UML was adopted to overcome this design limitation in WebML. The testing and evaluation results have proved that WBAMS has met all the requirements. In the future, this version of WBAMS is expected to run in the real environment and be enhanced as mentioned in section 8.6 to perfection.