ONLINE APPOINTMENT SYSTEM FOR FSKKP

PERPUSTAKAAN UMP
0000103227

LILIANA BINTI AZMI

REPORT SUBMITTED IN FULFILMENT OF THE DEGREE OF

COMPUTER SCIENCE (COMPUTER SYSTEMS AND NETWORKING

FACULTY OF COMPUTER SYSTEMS AND SOFTWARE ENGINEERING

UNIVERSITI MALAYSIA PAHANG

2014

BORANG PENGESAHAN STATUS TESIS

JUDUL : **ONLINE APPOINTMENT SYSTEM FOR FSKKP (OASF)**

SESI PENGAJIAN: 2014/2015

Saya: LILIANA BINTI AZMI

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (X)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☒	TIDAK TERHAD	

Disahkan oleh

(TANDATANGAN PENULIS) (TANDATANGAN PENYELIA)

Alamat Tetap:
42 , SOLOK KAMPUNG JAWA 5,
11950 BAYAN BARU ,
PULAU PINANG

Nama penyelia:
WAN NURULSAFAWATI BINTI WAN MANAN

Tarikh: 6 JANUARI 2015 Tarikh: 6 JANUARI 2015

CATATAN: * Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD.
*** Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara

STUDENT'S DECLARATION

I declare that this thesis entitled "Online Appointment System for FSKKP (OASF)" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature: .....

Name: Liliana Binti Azmi

Student Number: CA11069

Date: 24th DECEMBER 2014

SUPERVISOR'S DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of **Bachelor's Degree of Computer Science (Computer Systems and Networking) with Honors.**

Signature: .....

Supervisor: Mrs. Wan Nurulsafawati Binti Wan Manan

Date: 6 JANUARY 2015

DEDICATION

This thesis is specially dedicated for my beloved parents father, ***Azmi Bin Othman*** *and Mother* ***Faizah Binti Mohammed Bakri****, who guide me to this path.*

Also special thanks to all my lovely friends for their help and understanding throughout the project development... ***Khalisah Binti Abdul Basid, Ahmed Ali, Nur Zuraina, Osama Rasheed , Rima Sinno , Tan Cheng Chan , Noorfazilah*** *and etc... I won't forget you all!*

ACKNOWLEDGMENTS

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing for me to complete this project.

First of all, I would like to express my gratitude to my supervisor, Puan Wan Nurulsafawati Binti Wan Manan for guiding me throughout the process of finishing this system. Also thanks to all lecturers at Faculty of Computer Science and Software Engineering (FSSKP) for giving me ideas and suggestions on my system to make it possible to be completed.

Special appreciation to Khalisah Binti Abdul Basid for giving me continuous support and told me to never give up on making this project possible. Also thanks to all of my friends that helped and gave me ideas on solving the puzzle of my system.

I also want to thank all the staff in Faculty of Computer Science and Software Engineering (FSSKP) UMP for being cooperative in helping me to get information needed for my system.

Last but not least, my greatest thanks and appreciation to my parents, family and relatives for giving me full support and motivation from beginning till end of this project and also having faith in me.

ABSTRACT

Appointments are made manually by students to see lecturers. There are various ways students make appointment with lecturers in UMP such as going to see the lecturer, make phone call or message on social network. However, there is no proper information linkage of lecturer's availability that allows lecturer to update their availability regularly. Therefore, Online Appointment System for FSKKP is developed to reduce the difficulties in meeting lecturers among FSKKP students. Lecturers can update their schedule not just based on class but they can update with other activities such as university activities and events. This system is a web-based platform and is created using server side scripting PHP and Apache Web Server, user side scripting such HTML and CSS also MYSQL as a database for the system. This system is a web based application that allows user to access the system anywhere with an internet connection. This system will be developed using Rapid Application Development (RAD) model. This system gives benefits to both lecturers and students.

ABSTRAK

Perlantikan dibuat secara manual oleh pelajar untuk bertemu pensyarah. Terdapat pelbagai cara bagi pelajar-pelajar membuat temujanji dengan pensyarah di UMP seperti bertemu pensyarah secara peribadi, membuat panggilan telefon atau mesej melalui rangkaian sosial. Walau bagaimanapun, tiada rangkaian maklumat betul ketersediaan pensyarah yang membolehkan pensyarah untuk mengemaskini ketersediaan mereka dengan kerap. Oleh itu, sistem perlantikan dalam talian bagi FSKKP dibangunkan untuk mengurangkan kesukaran pelajar FSKKP bertemu pensyarah. Pensyarah boleh mengemaskini jadual mereka yang tidak hanya berdasarkan kelas tetapi mereka boleh mengemaskini dengan aktiviti lain seperti aktiviti-aktiviti universiti dan program. Sistem ini merupakan satu platform berasaskan web dan dicipta dengan menggunakan skrip PHP dan pelayan Web Apache, pengguna sampingan skrip HTML dan CSS tersebut juga MYSQL sebagai pangkalan data untuk sistem pelayan. Sistem ini merupakan aplikasi berasaskan web yang membolehkan pengguna untuk mengakses sistem ini di mana-mana dengan sambungan internet. Sistem ini akan dibangunkan menggunakan pembangunan aplikasi pantas (RAD).

TABLE OF CONTENT

CHAPTER 1__INTRODUCTION 1

1.0 Overview 1

1.1. Problem Statement 2
1.1.1. Unable to reach lecturer 2
1.1.2. No records of availability 3

1.2 Motivation 3

1.3 Objective 4

1.4 Scope 4

CHAPTER 2__LITERATURE REVIEW 6

1.0 Overview 6

2.1. Existing Systems on Online Appointment System 7
2.1.1. E-Appointment Scheduling (EAS) 7
2.1.2. Student's Module 8
2.1.3. Lecturer Module 8
2.1.4. Constraint of E-Appointment Scheduling (EAS) 10
2.1.5. Web Based Intelligent Appointment System 11
2.1.6. Patient Appointment Reservation System (PARS) 15
2.1.7. WAS-GN: Web-based Appointment System with GSM Network 18

2.2 Comparison of the Existing Systems 29
2.2.1 Advantages and Disadvantages of Existing Systems 29
2.2.2 The comparison between existing systems and Online Appointment System for FSKKP Lecturers and Students 30

2.3 Web-based System Techniques 30
2.3.1 Form Processing 30
2.3.2 Navigation 31
2.3.3 Database Operations 31
2.3.4 Authentication 31
2.3.5 Error Handling 32

2.4 Development Tools 33
2.4.1 Software Tools 33

2.4.2 Web-based Programming 34
2.4.3. Database 36
2.4.4. Web Server: Apache Webserver 37

2.5. Conclusion.....37

CHAPTER 3__METHODOLOGY.....38

3.0 Overview.....38

3.1. Introduction 38

3.2. Project Methodology.....39

3.3. Project Development Phase.....39
3.3.1. Requirement/ Planning phase.....40
3.3.2. System Design Phase 41
3.3.3. Development phase.....52
3.3.4. Cutover Phase.....53

3.4 System Requirements 54
3.4.1. Hardware Requirements.....54
3.4.2. Software Requirements.....54

3.5. Conclusion.....55

CHAPTER 4__IMPLEMENTATION 56

4.0 Overview.....56

4.1. Design 57

4.2. Database Design 57
4.2.1. Data Dictionary 58

4.2.2. SQL Statement 60

4.3. Coding Explanations.....63
4.3.1. Login Page 63
4.3.2. Appointment Page 65
4.3.3. Appointment Request 67
4.3.4. Approved Appointment 69
4.3.5. Update Schedule Page 71
4.3.6. Student Registration Page 73
4.3.7. Lecturer Registration Page.....74

CHAPTER 5__RESULT AND DISCUSSION 75

5.0 Overview 75

5.1. Result 76
5.1.1. Login Page 77
5.1.2. Registration Page 77
5.1.3. User Profile View and Update 78
5.1.4. Create Lecturer's Schedule 80
5.1.5. Update lecturer's schedule 82
5.1.6. Appointment Page 83
5.1.7. Student Appointment Request 84

5.2. Discussions 85

5.3. Advantages and Disadvantages 86
5.3.1. Advantages of OASF 86
5.3.2. Disadvantages of OASF 86

5.4. Assumption 87

5.5. Constraint 87
5.5.1. Technical Constraint 87

5.6. Future Development OASF 87

CHAPTER 6__CONCLUSION 88

6.0 Conclusion 88

REFERENCES 89

APPENDIX A 90

APPENDIX B 92

CHAPTER 1

INTRODUCTION

1.0 Overview

At the present time, everything in this world is depending on Information Communication Technology (ICT). With the rapid of usage of computers and gadgets, everything is computerized and it gives an enormous impact on our lives.

Nowadays, most organizations such as schools, hospitals, universities, and the government have started to do everything in a computerized way as it is painless and faster. In order to meet important people, an appointment should be made. Nevertheless, manual appointment system is not every efficient as it does not save time and money.

In Faculty of Computer Systems and Software Engineering (FSKKP), students still make appointment with lecturers manually. Therefore, the Online Appointment System for FSKKP Lecturers and Students (OASF) is developed to reduce the difficulties in meeting lecturers among FSKKP students.

This system is a web-based platform and will be created using server side scripting such as PHP with Apache Web Server, user side scripting such as

HTML and CSS also MYSQL as a database for the system. This shows that it is mobile as users can access to the system anywhere as long as there is an internet connection.

Appointments are made based on the time slots of the lecturer which can be updated by lecturer and also administrator. Lecturer will have to update their availability because in case they are on leave or have meetings .The students will check the lecturer's availably and pick the time slot they would like to meet the lecturer. Then, a request will be submitted and lecturers will be notified via email. If the lecturer is not available, the system will suggest other time slot that the student can pick.

In FSKKP, the online appointment system between lecturers and lecturers has already been developed. However, there is no similar system built to make an appointment between student and lecturer. The existing system is built in IMS (Information Management System) but this system is a standalone system.

1.1. Problem Statement

1.1.1. Unable to reach lecturer

Some lecturers can be very busy or have other duties and responsibilities other than teaching. This makes it hard for the students to meet them in their office room. In UMP, students will directly go to the lecturer's room or call to make confirmation. The lecturers often take an extended period of time to reply to students' calls or SMS (Short Messaging System). Waiting for a long reply from the lecturers who are busy or unavailable is considered time consuming to some students. This system will help to reduce students' waiting time for the lecturers as they may know the status of the lecturer before going to meet them.

Apart from that, students do not have the lecturer's phone number to contact. When they try to reach them at the faculty, the lecturer is not in the room either because of having class, meetings or on leave. On UMP's E-comm website, lecturers contact information can be found in user directory. The drawback is some lecturers only put office room's extension phone number. This is making it hard for students to contact lecturers after office hour to make an appointment on the next day especially for urgent matters such as discussing for Undergraduate Research Project.

1.1.2. No records of availability

Moreover, instructors tend not to abide by appointments that they assign to the students. Although they put notes at their doors or updates their availability on Facebook accounts, this is not practical or professional. With this system, lecturers are able to manage their appointments with students and can also check the approved appointments whenever they are logged in into the system.

1.2 Motivation

Conventionally, before students would meet up with the lecturers for discussions, they would have to go all the way to the faculty. Otherwise, they have to contact the lecturers via phone or through sending messages. However, this is costly for some students. Consequently, developing a system or program that is free and easier to use by students and lecturers was a considered thought to facilitate communication among them

Constructing this system looks challenging and intriguing since it needs independent thinking and intuition. Also, it helps in building up social and life skills and in incorporating previous knowledge with recent ones.

1.3 Objective

1. To study the lecturer-student appointment system that enables lecturer manages their time slot so that students can view to choose the suitable time slot to make an appointment with the lecturer.

2. To develop a system that allows students to make request to have an appointment with the lecturer after viewing the lecturer's availability and lecturer will choose either to deny or accept the request

3. To test the system by eliminating the possibility of reiteration of the same time slot with other students or in case lecturer is not available because of other meetings or university's activities

1.4 Scope

- **UMP's Faculty Of Computer Systems And Software Engineering (FSKKP)**
  - The target users of this system are students, lecturers and administrator. Students can make appointment only if they have logged in to this system's account. Therefore, only registered users can make appointment with the lecturers.

1. Administrator:

- Manage faculty record and lecturers that exist in the system
- Manage public holiday records and important updates
- Manage database

2. Lecturer

-Manage Profile and account

-Manage table and time slot

- View requests by students either to reject or accept

3. Student

- Check date and time slot before proceeding with the appointment process.
-Make booking and check the appointment request status.
-View the record of the sent appointment request.

CHAPTER 2

LITERATURE REVIEW

1.0 Overview

This chapter elaborates about the existing systems that are related to the Online Appointment System for FSKKP Lecturers and Students (OASF) and explains the existing technique/methods /languages used for each system.

Appointment is a time reserved for something such as a doctor visit, business deal, and much like a reservation. Recipient notification agents accept message notifications on behalf of recipients. Getting systems with many independent participants to behave is a great challenge (Mohd Helmy Abd Wahab, N. H. , 2008).

Nowadays, people demand to use computerized systems in their organizations. The reason is to make the human's workload to be minimized and at the same time this will need less workers or employees to handle various systems in an organization. An organization might need just one worker for one each system. Apart from that, technology helps people to reduce their time by using electronic systems instead of recording data manually. For that reason, online appointment systems are built in some organizations to make meetings and appointments can be made in a more appropriate way.

Online appointment system is a paperless electronic application that is designed with high flexibility and ease of usage, implemented for organizations such as faculties, administrations, hospitals, clinics and other business organizations to handle meetings with customers or clients in more efficient way. There are many kinds of online appointment system that exist nowadays. This system is generally built to avoid reiteration o of the same time slot for different user.

2.1. Existing Systems on Online Appointment System

2.1.1. E-Appointment Scheduling (EAS)

E-Appointment Scheduling (EAS) has been developed to handle appointment for UMP students, lecturers in Faculty of Computer Systems & Software Engineering (FCSSE) and Student Medical Center. It is an online application for FCSSE's student whenever applying to make appointment with lecturers or doctor. All applications have to be sent to the lecturers or doctor for approval. This system will give more interactive for student to make an appointment through an online system. By deploying this system, we can avoid wasting time and cost because this application will set an appointment by auto-generated. Therefore, this system is hopefully to solve problem for scheduling. (Noraziah Ahmad, Roslina Mohd Sidek, and Mohd Affendy Omardin, 2010).

In order to solve the scheduling drawbacks of this system, Constraints Logic Programming (CLP) has implemented in this system. by giving suggestions to the users in part of determining any available slots from the lecturers and doctors' timetable

2.1.2. Student's Module

From students' page view, all students are allowed to use their student id as username and password for the first time login into the system. Before making an appointment, student can view availability of lecturers and doctors. To make an appointment with the lecturer, student must search the lecturer by lecturer's name, date and time. System view available slots that student request, if not system suggests other slot to make an appointment. It also similar with the doctor module but the appointment only generated to doctor.

Next, the system will display for search lecturer's schedule or doctor's schedule by constraint inserted by student to do the appointment. Available slots that student needs will be searched by the system otherwise, gives other available slot suggestions if the constraints do not match. Students just click the result to do the appointment.

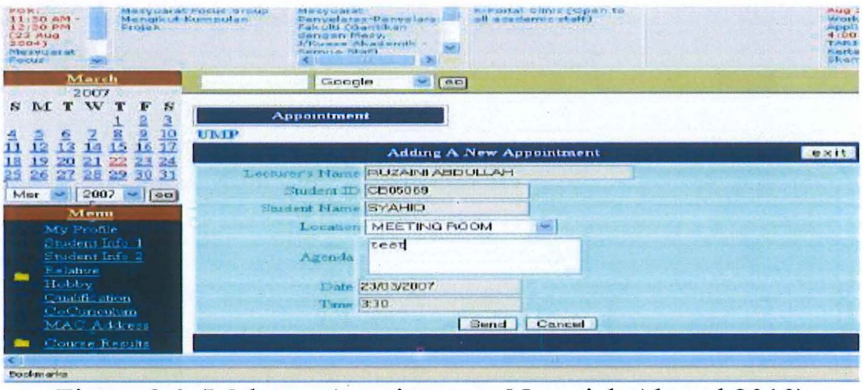


Figure 2.0 (Make an Appointment, Noraziah Ahmad,2010)

Next, as shown Figure 2.0, student is required to insert appointment's location and agenda and click send button or cancel or exit to abort the appointment.

2.1.3. Lecturer Module

After appointment has been made, the database is then updated and enables lecturer to view the request. Even though every appointment request

made is based on available slots, lecturer still can reject or change the time and date in case of emergency. Lecturers also can edit schedule to update the available schedule. Report button as in figure 2.1 below is for lecturer to view the appointment records.

Figure 2.1 (List of Approved Appointment Appointment, Noraziah Ahmad,2010)

Figure 2.1 shows the approved appointment that is automatically made by the system. Information about the application and also can be viewed .By clicking the image view at detail column lecturer can check the detail of the applicant.

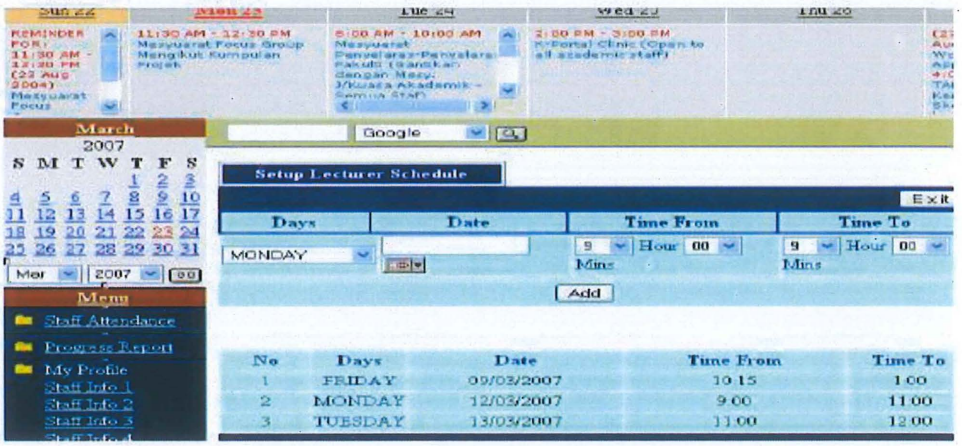


Figure 2.2 (Lecturer Setup Schedule, Noraziah Ahmad,2010)

Figure2.2 shows that lecturer is able to setup the schedule for the appointment.

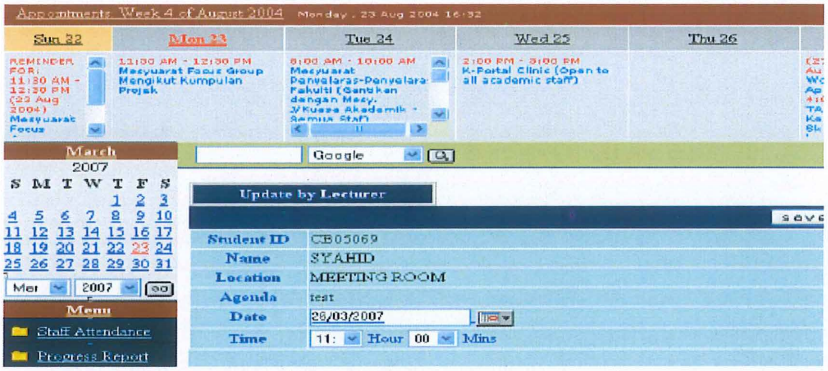


Figure 2.3 (Lecturer Change Date and Time, Noraziah Ahmad,2010)

Figure 2.3 shows lecturer can update the appointment requested by student in case there are changes to be made and it will be sent to student's memo.

2.1.4. Constraint of E-Appointment Scheduling (EAS)

Based on the research, EAS is in an IMS (Integrated Management System) which is a single integrated system used by an organisation to manage the fullness of its processes, in order to meet the organisation's objectives and fairly satisfy the stakeholders. Combines all related components of a business into one system for easier management and operations (Sciqual.com.au, 2015). Therefore, when there are so many things going on a website, the appointments might be missed or forgotten. Therefore Standalone system is better for an appointment because it operates independently which means there is only one system.

Lecturers need to setup their timetable themselves. This will make it hard if lecturer does not update or forgot to setup the schedule. Administrator should be responsible to make sure the schedule is always up to date.

Besides that, the EAS does not provide a timetable that enables student to check if the lecturer is available or not but allows lecturer to change the time of the requested appointment instead. This will cause difficulties in case student has class or other university activities when the lecturer updates the time. Besides that this system is also time consuming because students have to check

one at a time for the available time by using the search of lecturer's name, date and time.

2.1.5. Web Based Intelligent Appointment System

Web Based Intelligent Appointment System is an online appointment system developed by integrating with Intelligent System techniques. The purpose of an appointment is for students to reserve time for any academic-related activities such as discussion and weekly meeting with lecturers. The main orientation of the prototype is to manage appointment and calendar updating.

2.1.5.1. Database design

Database is used as the platform for most information systems that stores data. It is the ultimate instrument for most systems.

There are several steps in database design as described by inflow schema that consists of

i) process event
ii) function links and
iii) directed communications
(King, 1985).

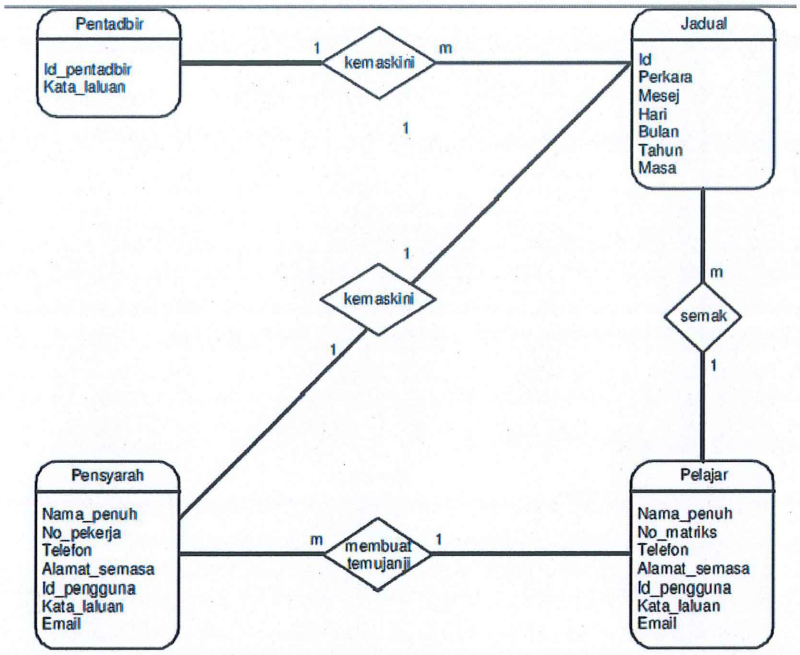


Figure 2.4(Mohd Helmy, Entity Relationship Diagram for Web-based Intelligent appointment system,2009)

2.1.5.2. Interface Design

Figure 2.5 (Interface of Appointment Timetable, Mohd Helmy, 2009)

In figure 2.5, it shows that students can make an appointment by choosing the blue coloured time slots.

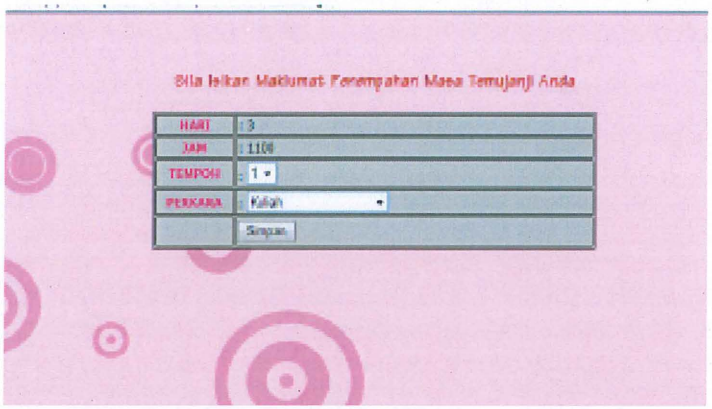


Figure 2.6 (Interface of Online Appointment Form, , Mohd Helmy, 2009)

In figure 2.6, students can make an appointment by selecting the appointment duration, and purpose of meeting.

Figure 2.7 (Interface of Lecturer's Timetable in the Lecturer's Mode, Mohd Helmy, 2009)

Figure 2.7 show that lecturer can change the time slot by clicking on the time slot that needs to be changed.

Figure 2.8 (Interface of New User Registration, Mohd Helmy, 2009)

Figure 2.8 is the interface for new user registration.

2.1.5.3. Intelligent Agents

Agent-based computing has taken place as "the next significant break-through software development (Jenning and Woodridge, 1998). Different types of agents have with different role. For this system, agent's role is to manage information in databases and offer a status by comparing it with inputs provided by the users and capable of autonomous action to meet its design objectives.

Agent is a computer program that assist user with a routine computer task and represents on behalf of human agents (Noraziah Ahmad, R. M. 2010). At the user interface, the user interacts with the agent while the agent senses and acts independently in a work environment such as an operating system. Using the information taken from its environment, the agent performs a given task. The role of agent is to respond the user's request in ad hoc and an Intelligent Agent is placed in the prototype. It allows both students and lecturers to easily access the system in any terminal connected to the Internet while in a time restrain.

2.1.5.3.1. Advantages of Intelligent Agents

i. Higher efficiency in work such as less time used, work autonomously, and can search huge amounts of information and filter out important things that would be impossible for humans

i. Opens new opportunities like an arrangement of appointments inclusive of searching for the available slot for an appointment and respond to

2.1.5.3.2. Constraint of Web Based Intelligent Appointment System

After understanding the research, the system does not have so many constraints but however it is still lacking in notification feature. This system does not notify lecturer on whether they have an appointment to be checked or not. The appointment will be approved automatically and lecturer does not need to approve or reject the appointment request. This will cause complications when students do not check whether the lecturer has changed the time slot for the meeting.

2.1.6. Patient Appointment Reservation System (PARS)

Based on the research on Patient Appointment Reservation System (PARS), it is a system that has been developed to use the opportunity of possibilities to reduce administration costs, provide availability and high quality service in health care and more efficient human and material resources for health care organizations that is provided by the advanced Internet and Technologies in medical.

For now PARS is one of the most modern projects in Lithuania's medicine sphere, linking registries of 40 different health care institutions:

University Hospitals, Regional Hospitals, City Hospitals and Policlinics (Primary and Secondary Outpatient Care Centers) (Vilnius,2008)

2.1.6.1. System operation principles

1. Specialists -
   - Make a consultation time schedule.
   - Enter scheduled consultation times of physician into PARS by the reception personnel.
   - Can enter planned consultation times themselves if signed in.
   - A specialist can be chosen by health care institution, family name, specialty and consulting-room or by the set of all these criteria. When finding a proper specialist patient is able to view all available times of visits and select the most convenient one.
   - Can view a list of registered patients for a particular date and their complaints.
   - Can send SMS for a patient to bring all the necessary documents or analysis that might be useful.

2. Patients
   - Able to register for a visit by the phone or at the reception desk.
   - All patients' details are entered into PARS and can register online.
   - Can reserve consultation time by entering name, family name, mobile phone number and other contact information of patient.
   - Will receive SMS for confirmation, reminder about upcoming visit and information of the reservation cancelation if there are there are things that cannot be circumvented

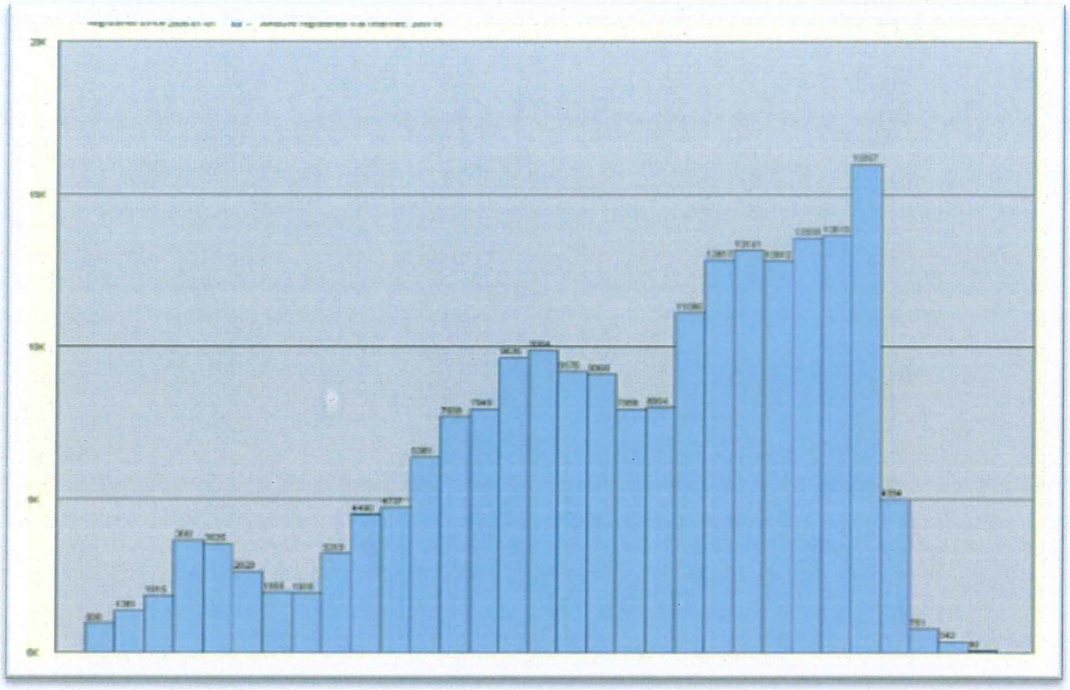


Figure 2.9 Patients appointment reservations via Internet since 01.01.2008, Vilnius,2008)

Figure 2.9 shows that since year 2008 where this project has begun, the patient appointment reservation via internet have been increasing steadily

2.1.6.2. Constraint of Patient Appointment Reservation System (PARS)

PARS is a huge system and it needs to work perfectly to get the users' satisfaction. However, there are problems in the system. Firstly is the system is created in Lithuanian Language. For a system like this, it is better to be implemented in English or make an option either to use English or Lithuanian language. This is because; in case a non-Lithuanian wants to use the system, it will cause difficulties and still a time consuming. Besides that it will cause false

information entered because of the misunderstanding the Lithuanian language. Therefore, if this system is going to be implemented in other countries, they should consider making it user friendly and understandable.

2.1.7. WAS-GN: Web-based Appointment System with GSM Network

Based on the research paper of WAS-GN: Web- Based Appointment System with GSM Network, this system works with the two way communication using Web-to-SMS technology. At the present time, WAS-GN has already been conveyed onto the mobile phone. Deploying the GSM enables the communications between the different technologies.

2.1.7.1. System Architecture

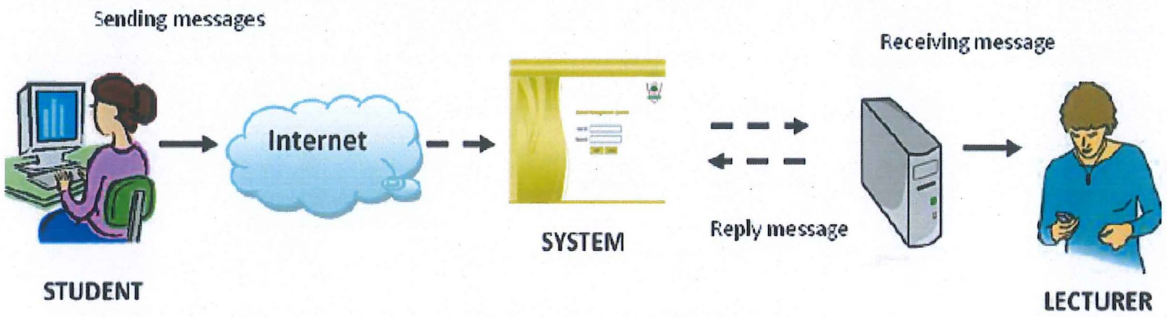


Figure 2.10 (WAS-GN Architecture, Mohd Helmy, 2011)

Figure 2.10 shows other system architecture and how this system basically works. Student will apply for appointment with the lecturer through the internet on the website. Then the lecturer will receive message or SMS to reply to the message.

2.1.7.2. Hardware Requirement

GSM modem is a wireless embedded GSM communicating module similar to the SIM card in cell phones (Vilnius,2008). GSM constructs wireless connection with the SMS Center. Besides, it also can be connected to the computer through RS232 serial port to make calls, receive/send messages and deal with other business transactions. There are many advantages of GSM modem such as high reliability, no batteries, wireless, sending /receiving SMS at any time and low price.

For this project WaveCom GSM modem is used. Mobile Station consists of physical terminal and contains the radio transceiver, the display and digital signal processor and Subscriber Identity Module (SIM). SIM allows users to access a network and provides subscribed services for them (Helmy, Gopalakrishna and Johari, 2008).

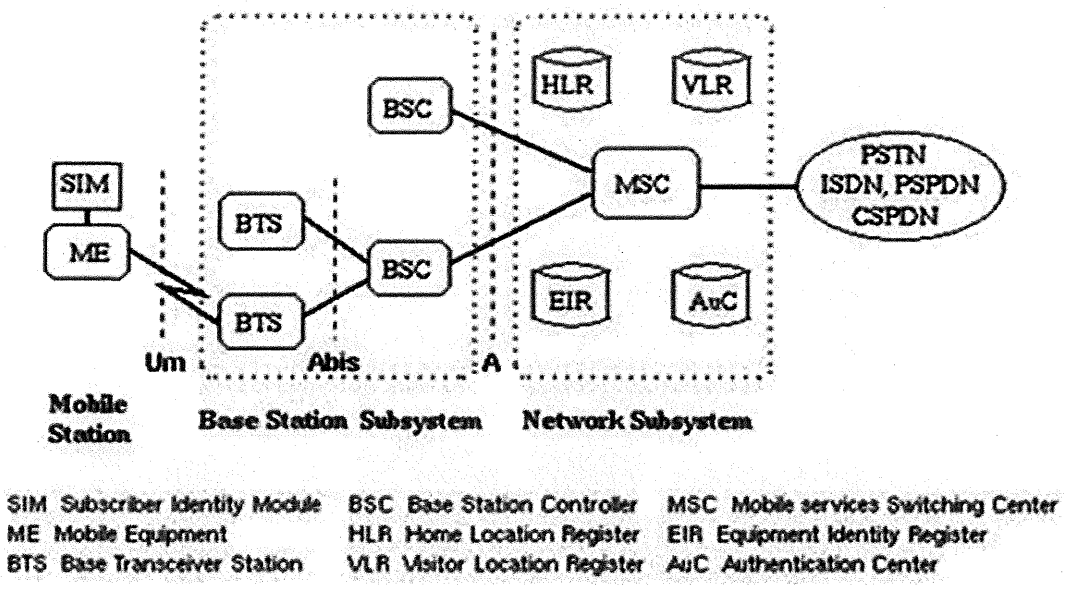


Figure 2.11 (GSM components, Mohd Helmy,2011)

Figure 2.11 shows the GSM network architecture that consists of Mobile Station (MS), Base Station Subsystem (BSS) and the Network Subsystem (NS) and common for both SMS and MMS

2.1.7.2.1. Software Development

WAS-GN was initially developed on a Web-based environment. Figure 2.12, shows four components involved in the development of WAS-GN which are the database (Ms Access), client application (Dreamweaver), server application (HypertextPreprocessor – PHP), and the server (XAMPP Server).

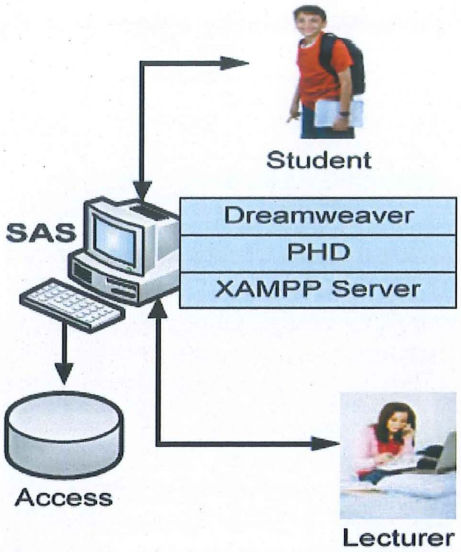


Figure 2.12 WAS-GN system context

Figure 2.12 illustrates that users (i.e. students and lecturers) can access WAS-GN through its client application with interfaces in between. Users can input new data, and update or delete the existing data from the client side. The information retrieval is smooth, and all transactions are processed by the server side scripting using PHP.

2.1.7.3. Interface of WAS-GN

2.1.7.3.1. Student

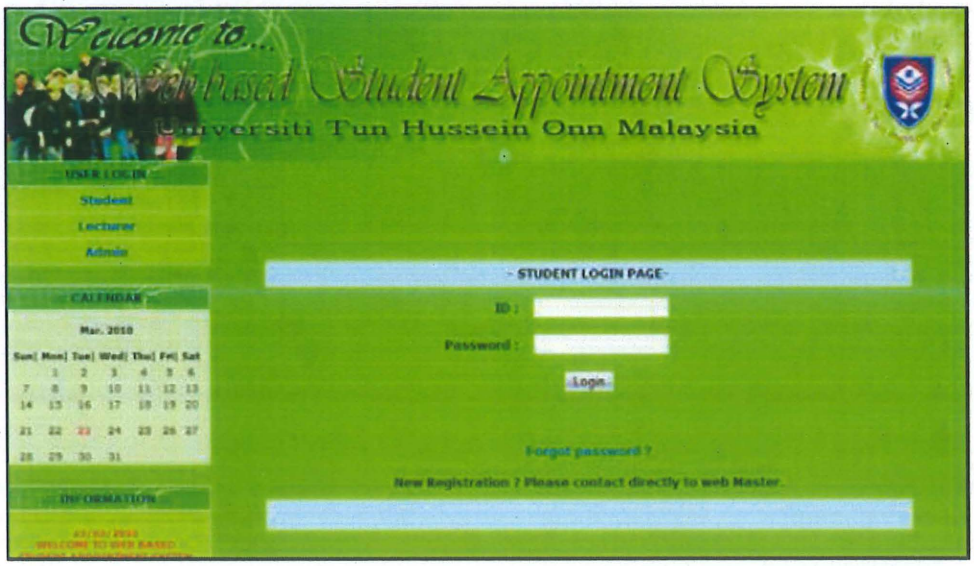


Figure 2.13 (WAS-GN architecture, Mohd Helmy, 2011)

As shown Figure 2.13, it is the authentication page that gives verification for registered students only to access into the system. Students need to enter the matrix number and password to log into the system.

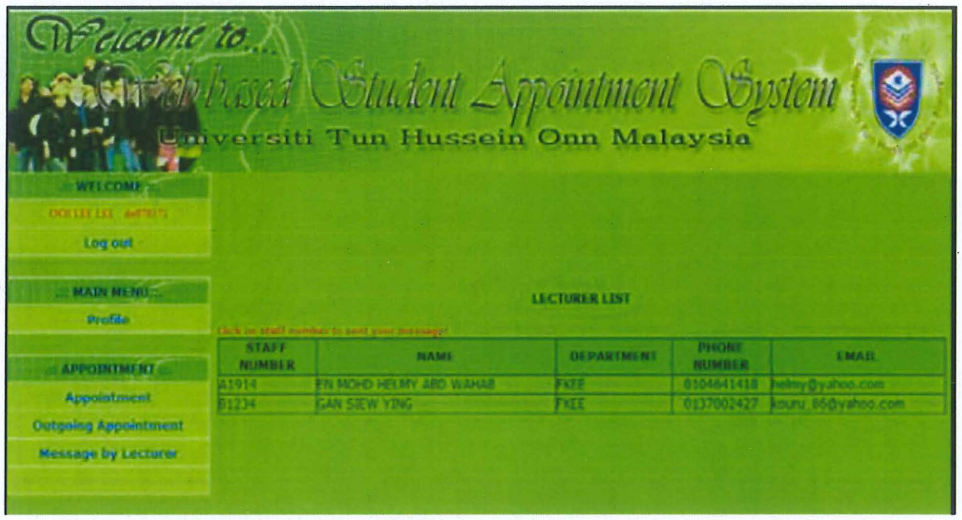


Figure 2.14 (Home Page Student, Mohd Helmy Abdul Wahab, 2011)

In WAS-GN, besides making an appointment, students can view the appointments status, and update the personal information. Students do not have to remember the lecturers' name because the system will provide the list of lecturers' name as shown in figure 2.14.

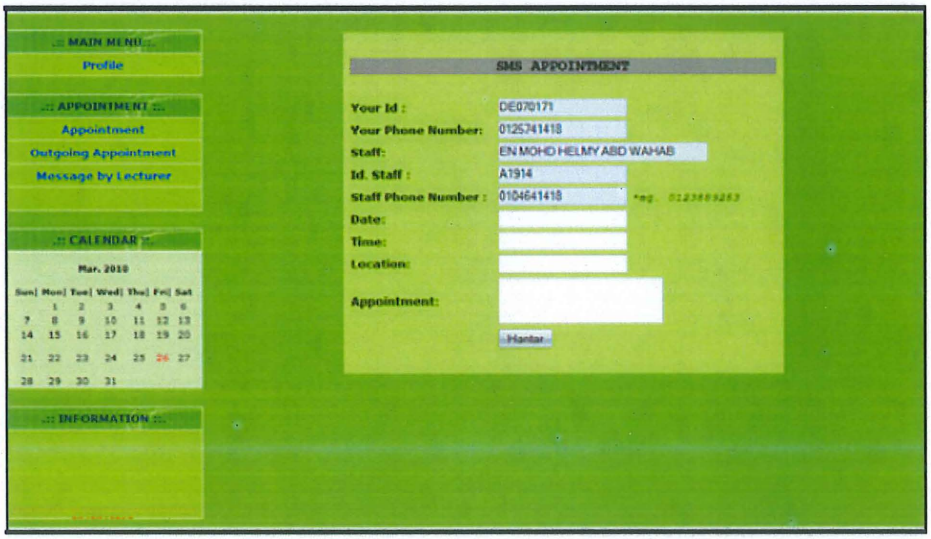


Figure 2.15 (Form to make appointment, Mohd Helmy Abdul Wahab, 2011)

After choosing the lecturer, the interface as in Figure 2.15 will appear. Students will fill the details of the appointment which are the date, time,

message and location. Lecturer will receive the details of the appointment via SMS.

Figure 2.16 (Information and Appointment Status, Mohd Helmy Abdul Wahab, 2011)

This interface shows that students can view the appointment status and the information. Lecturer name, date, time, location, and message set by the student will be displayed as shown in Figure 2.16. Lecturer will set the appointment status.

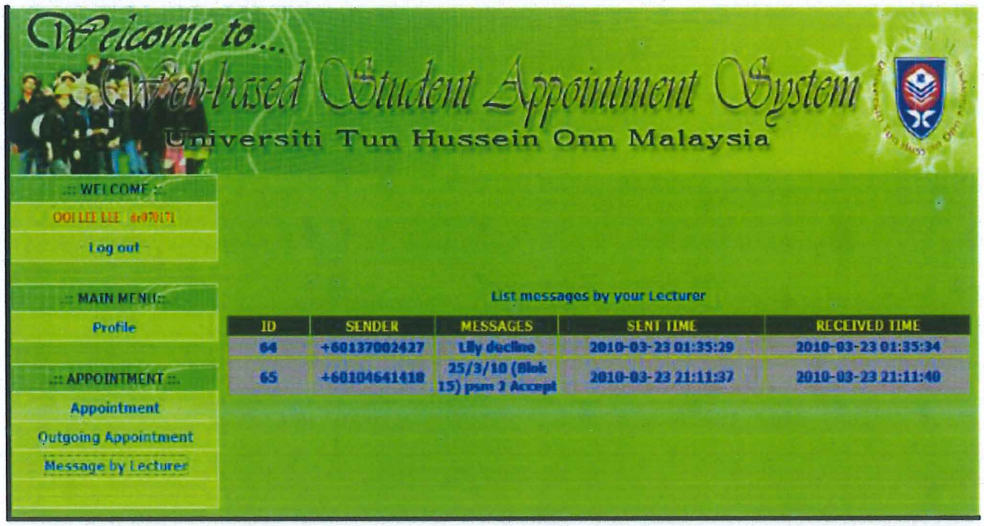


Figure 2.17 (Information displayed from SMS. Mohd Helmy Abdul Wahab, 2011)

In figure 2.17, shows and interface of the appointment status which displays lecturer's mobile phone number, messages, sent and received time of the message.

Figure 2.18 (Student Information Update,Mohd Helmy Abdul wahab , 2011)

When the update the personal information selected, student will see a form as shown in Figure 2.18. Then students can change their password, user name and other detail.

2.1.7.3.2. Lecturer

Lecturer also will see the authentication page just like in figure 2.12 to make sure that only registered user can access to the system.

After lecturer has logged in into the system, interface in figure2.19 is displayed by default. Lecturers can also check for the requested appointments, update personal information or log out from the system.

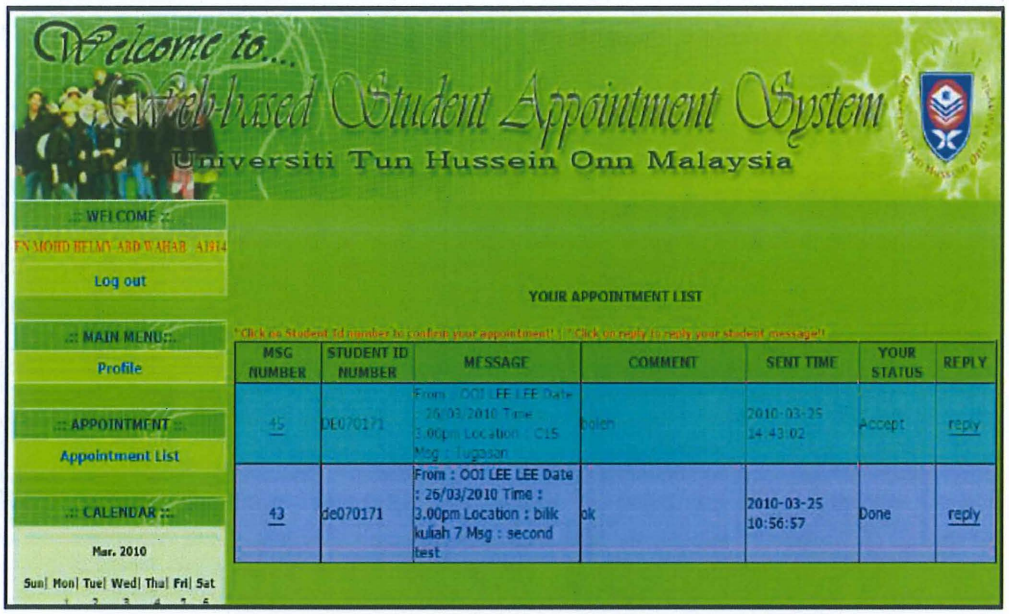


Figure 2.19 (Home Page Lecturer, Mohd Helmy Abd Wahab, 2011)

Figure 2.20 (Appointment List, Mohd Helmy Abdul Wahab, 2014)

Lecturers can check the requested appointment which also displays name, matrix number, details of appointment and appointment status. The, lecturers are able to accept or reject the appointment then it is sent and updated to the system like in figure 2.20

Figure 2.21 (Lecturer Information Update, Mohd Helmy Abdul Wahab, 2011)

On interface like in figure 2.20, lecturer can update personal information such as change password and mobile phone number.

2.1.7.3.3. Administrator

The interface of the administrator's login page is also the similar like student and lecturer.

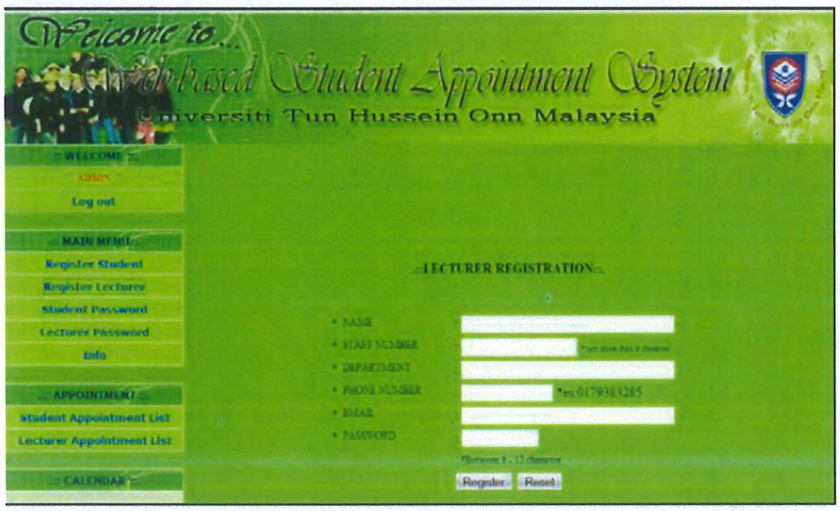


Figure 2.22 (New Lecturer Registration, Mohd Helmy Abdul Wahab, 2011)

Figure 2.22 shows that admin can make new registration for lecturer.

Figure 2.23 (New Student Registration, Mohd Helmy Abdul Wahab, 2011)

Interface in figure 2.23 shows that admin can make new registration for student.

Figure 2.24 (Forgot Password Page, Mohd Helmy Abdul Wahab, 2011)

Figure 2.24 shows that if users forgot their password, administrator can generate a new default password they will be informed via email

Figure 2.25 (Addition Information Page, Mohd Helmy Abdul Wahab, 2011)

Figure 2.25 shows that administrator can update information by adding info title, date, and complete information.

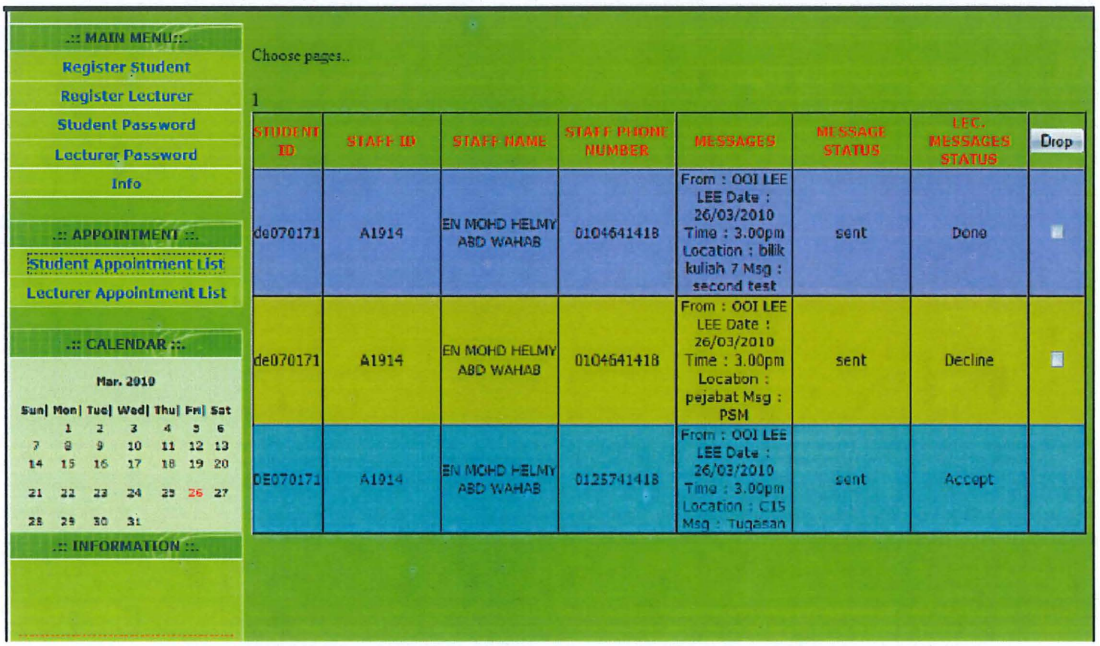


Figure 2.26 (Student Appointment List, Mohd Helmy Abdul Wahab, 2011)

In figure 2.26, administrator can update student's appointment list by deleting the appointments are done or declined.

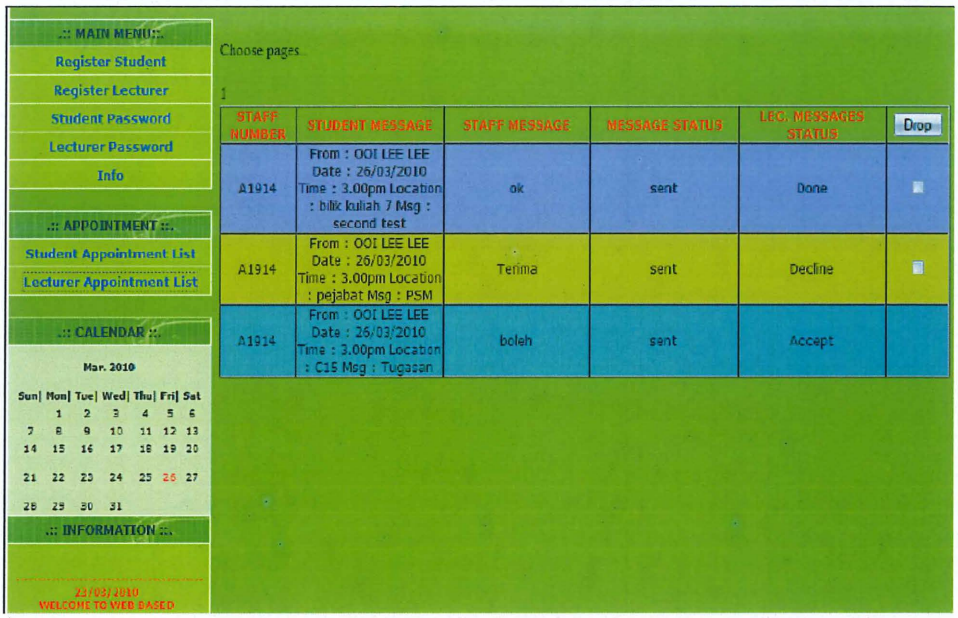


Figure 2.27 (Lecturer Appointment List, Mohd Helmy Abdul Wahab, 2011)

In figure 2.27, administrator can update lecturer's appointment list by deleting the appointments are done or declined.

2.1.7.4. Constraint of WAS-GN: Web-based Appointment System with GSM Network

WAS-GN: Web-based Appointment System with GSM Network is basically system for students to make an appointments with lecturer through web via the internet and the lecturer will reply through SMS and students can view the status on the web. There are a few problems found in this system. The first thing is there is no lecturer's time table provided. This means students cannot know the lecturer's availability and will just blindly apply for an appointment and probably will keep requesting for the wrong time. Time slot should be provided to make it easier for students to arrange time. This is because students also might be busy with classes and other activities.

The second problem is the SMS notification which will cost money .If a lecturer keeps receiving too many SMS, it will cost him/her more. Even though SMS is fast, e-mail will not cost much as long as there is internet connection, email can be the fastest. The delay time taken is just a few seconds and nowadays emails can be accessed on smartphones too.

2.2 Comparison of the Existing Systems

2.2.1 Advantages and Disadvantages of Existing Systems

Existing System	**EAS**	**PARS**	**Web Based Intelligent Appointment System**	**WAS-GN**
Features	Allows students to make appointment with lecturers and doctors.	Patient can make appointment reservation in 40 different health care institutions	Students can make appointment with lecturers by picking the time slot they want	Students can make appointment on the web and lecturer can reply using SMS
Advantages	Accessible anywhere with internet connection	Registered patient will receive SMS for confirmation or cancellation	Students can check the lecturer's availability	Receive appointment approval faster
Disadvantages	Delay in response from lecturer	Stressful to decide on the time of the visit when calling or talking to the personnel of the registry feeling indirect pressure to choose quickly	No notification	Costs money

2.2.2 The comparison between existing systems and Online Appointment System for FSKKP Lecturers and Students

Existing System	**EAS**	**PARS**	**Web Based Intelligent Appointment System**	**WAS-GN**	**OASF**
Type of System	IMS	Standalone	Standalone	Standalon e	Standalone
Notification	Memo	SMS	No	SMS	Email
Technique	Constraint Logic Programmin g (CLP)	Data exchange and synchronizati on	Intelligent Agent	Not mentioned	Web-based System and Web-based Programmin g
System Access	Only students and lecturer can access to the system	The public can access to the system	Only registered user can access to the system	Only registered user can access to the system	Only registered user can access to the system.

2.3 Web-based System Techniques

This section will explain about web- based system techniques that will be during the implementation of the OASF.

2.3.1 Form Processing

Form processing is a process that captures a problem of getting and authenticating input from a user entered into data fields and transforms it into an automated format. This process can be done manually or automatically, but the general process is that hard copy data is filled out by humans and then "captured" from their respective fields and entered into a database or other electronic format. Form processing consists of two sides which are the client-side and the server-side. The client-side is for the user's use and the server-side is when the browser sends the data of the form to the server. After data has been

sent, the server should know what to do next with the data received. The web developer should to provide those instructions to the server.

2.3.2 Navigation

Navigation should be neat, structured and user friendly. The navigation of a website should also have a consistent look and feel throughout the site. A good website should provide a search engine to make it easier for user to find what they are looking for on a website. Poor navigation will make user feels lazy to use the system especially when the users are computer literate, they will expect a good navigation to be implemented.

2.3.3 Database Operations

Many web applications and systems revolve by letting the user to insert new

information into the database and edit existing entries. For this system, it will allow lecturer to update the time table in case there are changes for meetings or on leave

2.3.4 Authentication

Authentication is used to declare or verify if someone is allowed to access to a system. Normally, login password is required to make sure that the access is only for the right person only. For example of authentication in daily life is private Wi-Fi password. Only user who knows the password can access to Wi-Fi network. There are a few levels of authentication used to authenticate a request on a website. The levels of authentication are related and have similar implementations.

- None – does not do anything to record or track the request.
- Session – tracking anonymous user by their current visit only
- Visitor - tracking anonymous user across numerous sessions at the site.
- User – Required to authenticate based on the information recorded in the database.

Authentication technique will be applied is to this system is to make sure that only registered user can access to the system with the account they registered.

2.3.5 Error Handling

There are circumstances that cannot be prevented while developing a website using programming. One of it is the occurrence of error because creating free error degradation into a web site is a major challenge. Indeed it is not easy to catch errors before they confuse users and also to avoid the error from recurring .However, when problem occur, it can be fixed by handling the problem with patients and effort.

For example, if the database is being backed up and thus read-only can we continue to serve database requests without trying to write. Another example is if the database is offline completely website is still able to continue to show pages that do not really need to use the database.

2.4 Development Tools

These are the tools that will be used to develop the system

Software	Adobe Dreamweaver CS5, Adobe Photoshop CS5
Programming Language	HTML, CSS,PHP 5.0
Database Management System (DBMS)	MySQL, phpMyAdmin
Web Server	XAMPP (Apache HTTP Server)

2.4.1 Software Tools

2.4.1.1 Adobe Dreamweaver CS5

Adobe Dreamweaver is a tool used to design, develop and deploy websites. It is also used to create web interface using less coding by just drag and drop in order to create a clean professional looking website. Adobe Dreamweaver offers unparalleled CSS features. It has wonderful features that can make the complicated coding processes become easy and simple. The advantage of using Dreamweaver is it includes the visual workflow, advance site management capabilities and support for technologies for instance JavaScript, PHP, and XML.

2.4.1.2. Adobe Photoshop CS6

Adobe Photoshop CS6 is one of Adobe product which is an application that is used to edit graphics and images to look more interesting. The images like banners or logo that is related and going to be used on the website will be edited or enhanced using Adobe Photoshop CS6. Besides that , there are various techniques offered in Adobe Photoshop CS6 to make graphics look the way you want and it is also easy to understand.

2.4.2 Web-based Programming

2.4.2.1. Programming: PHP (Hypertext Preprocessor)

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language.It is originally designed for producing dynamic web pages. It is also used to embed scripting languages into HTML source document instead of calling an external file for data processing.

PHP is mainly used in server-side scripting, this is the most traditional and main target field for PHP. There are there things to make this work ;

1. The PHP parser (CGI or server module)
2. A web server -- needs a connected PHP installation
3. A web browser -- access PHP page through URL

PHP is also used for Command Line scripting which can make a PHP script to run without any server or browser. PHP parser is needed to use it this way. These scripts can also be used for simple text processing tasks similar to PERL. PHP can also be used when writing client-side GUI applications .However, PHP is probably not the very best language to write windowing applications.

2.4.2.2. HTML (Hyper-text Mark-up Language)

HTML or Hyper-text Mark-up Language is a programming language that is recognised worldwide and is used on the World Wide Web. Nowadays, it is generally used along with JavaScript and Cascading Style Sheets (CSS) to give the web pages the look as however we wish for. By using HTML, every element s can be formatted. For examples are headings, links, images, texts and many others.

2.4.2.3. Advantages and disadvantages of HTML

Advantages	Disadvantages
1. Easy to use and learn especially for novice programmers	1. Not easy to read
2. Supported on all browsers	2. not rendered correctly in all browsers
3. Free to use with no plugins or software	3. cannot produce dynamic output alone
4. Case Insensitive	4. Must edit all pages individually
5. Very similar to XML	5. Limited security features

2.4.2.4. CSS (Cascading Style Sheets)

CSS (**C**ascading **S**tyle **S**heets) is a markup language that is used for describing the look and feel of user interface and also for formatting a document. It is also a way to display HTML elements in style. The purpose of designing CSS is to allow the separation of document content from document presentation that includes elements such as the interface elements such as colours, fonts and layout.

Just like HTML, CSS is supported on all browsers. Style has solved a big problem which is the process of developing a big website that takes quite a long time. This problem happened when adding a colour and fonts on a website needed to be added on every page. Hence CSS helps in saving time and also can save a lot of work.

2.4.3. Database

2.4.3.1. MySQL

MySQL is a database system used on the web and runs on a server. It is suitable for both small and big applications. Besides, it is very fast, consistent, and easy to use also supports standard SQL. It compiles on a number of platforms. The advantage of using MySQL is that can download and use it for free. It is developed, distributed, and supported by Oracle Corporation. This system will use MySQL to keeps every details of information of lecturers, students, administrator in the database storage.

2.4.3.2. Database Administrator: PHPMyAdmin

PHPMyAdmin is designed to handle the administration and management of MySQL databases through a graphic user interface. It is an open source software which means it is free to use. It comes with detailed documentation and is being supported by a large multi-language community. It is useful to create and drop databases, create, copy, drop and alter tables, delete, edit and add fields, execute any SQL-statement, even batch-queries.

MySQL users and user privileges can be managed by using PHPMyAdmin. Besides, the common feature is its import function. With PHPMyAdmin, MySQL database import from backup is made easy and you can import an SQL or CSV dump with a few mouse clicks. Database can also easily be exported in CSV, SQL, XML, Excel and other popular formats.

2.4.4. Web Server: Apache Webserver

Apache in general is known as the world's most popular Web server. A HTTP Web Server that is flexible as it is adaptable for everyone to use its server for specific needs. The source code is always free and available. Apache's add-ons have a huge number in the public library. The Apache Web Server is providing a full range of Web Server features that includes CGI, SSL, and virtual domains.

2.5. Conclusion

Based on the information above, the Online Appointment System for FSKKP Lecturers and Students (OASF) can suggest time slot that student can choose if the lecturer is not available for the chosen time. Besides that lecturers will be notified via their email instead of memo because it is a standalone system. It is an online system that allows user to use the system everywhere. Lecturer also can always update their availability if there are any changes on the time slot set by admin and can reject or accept the request. However, only registered students can use this system

CHAPTER 3

METHODOLOGY

3.0 Overview

This chapter is about the brief explanation of the methodology and procedures of Online Appointment System for FSKKP Lecturers and Students or OASF including the methods, techniques that are applied, hardware and software specification.

Methodology is generally a guideline for solving a problem, with specific components such as phases, tasks, methods, techniques and tools (Irny Suzila Ishak, Designing a Strategic Information Systems Planning Methodology for Malaysian Institutes of Higher Learning ISP-IPTA, 2005)

3.1. Introduction

This chapter also explains about the project development phases of Online Appointment System for FSKKP Lecturers and Students (OASF) that is used as a guide and direction throughout the project.For this project, Rapid

Application Development (RAD) Model is the Software Development Life Cycle (SDLC) applied for this project. In the next topic, will be discussed the explanation of the chosen methodology and its development stages.

3.2. Project Methodology

The methodology chosen for this project is Rapid Application Development (RAD). RAD includes with writing the software in the planning of a system or project. This methodology permits the system to be written faster and easier to change requirements because it does not need of wide pre-planning. The whole process of software development for this project is divided into four phases which are Requirements Planning, System Design, Development and Cutover.

There are benefits that this model offers. Firstly, is its flexibility and adaptability to changes of the system Secondly, it handles huge projects without a doubt. Besides that, it increases the speed and quality of development.

3.3. Project Development Phase

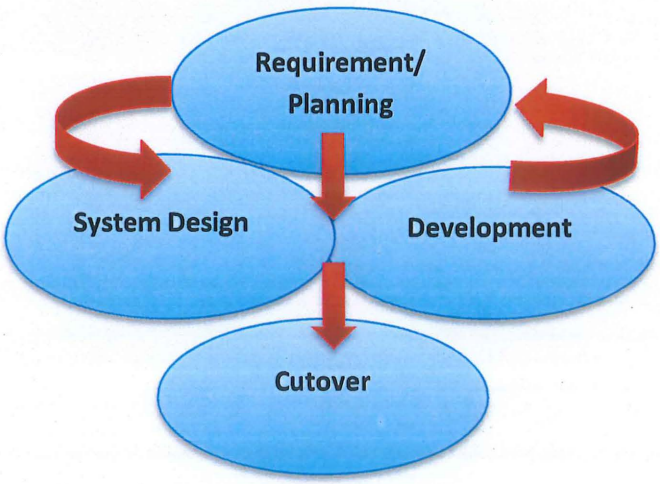


Figure 3.0 The Rapid Application Development (RAD) Model.

3.3.1. Requirement/ Planning phase

This phase includes system planning and system analysis. During this phase developer and users discuss on what is needed in the system. The discussion included the requirements, project scope and constraints of the system. For the system requirement, it includes the user requirement and data requirement.

3.3.1.1. User requirements

The users of this system are lecturers, students and administrator. User requirements are gathered based on observation on student and meeting with a few lecturers.

3.3.1.2. Observation

The first problem faced by students is the student does not know the lecturer's availability. Student needs to go to the faculty to find the lecturer and can only see the timetable on lecturer's door. Thus, this system will provide lecturer's timetable for student's view and they can pick the desired time to make appointment with lecturer.

Besides that, the lecturers are unreachable. This is because lecturers might have other meetings or involved in university activities. Calling the lecturers can cost money and waiting consumes time. Therefore, this system can be accessed everywhere with internet connection and it will save students time.

3.3.1.3. Meeting

Based on the meeting with Dr. Nizam, he suggested that the system will view the other time slot if the time picked by the student is clashing or lecturer is not available at that time. Madam Wan Nurulsafawati also suggested that this system is built as a Standalone System because Integrated Management System (IMS) has many applications in a system. This might cause lecturer miss to check on appointment notification.

3.3.1.4. Data Requirements

Before proceeding to the development phase, data requirement also collected from the faculty officer. A few of lecturer's timetable are collected from FSKKP's administration department that includes lecturer's staff ID and full name. The data collected is going to be used by administrator to register lecturers in the system and managing lecturer's timetable.

3.3.2. System Design Phase

The System Design Phase shows the interaction of users with OASF. The requirements from the first phase are studied and designs for Flow Chart, Context Diagram, Data Flow Diagram (DFD), Entity-Relationship Diagrams (ERD) design is prepared. System Design helps in identifying hardware and system requirements and also helps in building overall system architecture. The programming language that is going to be used to design the user interface is HTML, PHP and CSS. These languages are chosen because they are flexible and easy to be implemented.

There are modules that are related to each other in this system. The first module is the lecturer module for the lecturers. The second module is the student module for students and the administrator module for the administrator. Each relation can be shown in a few designs such as the Flow Chart, Context Diagram, Data Flow Diagram (DFD) and Entity Relationship Diagram (ERD).

3.3.1.5. The Proposed Design

3.3.1.5.1. Flow Chart

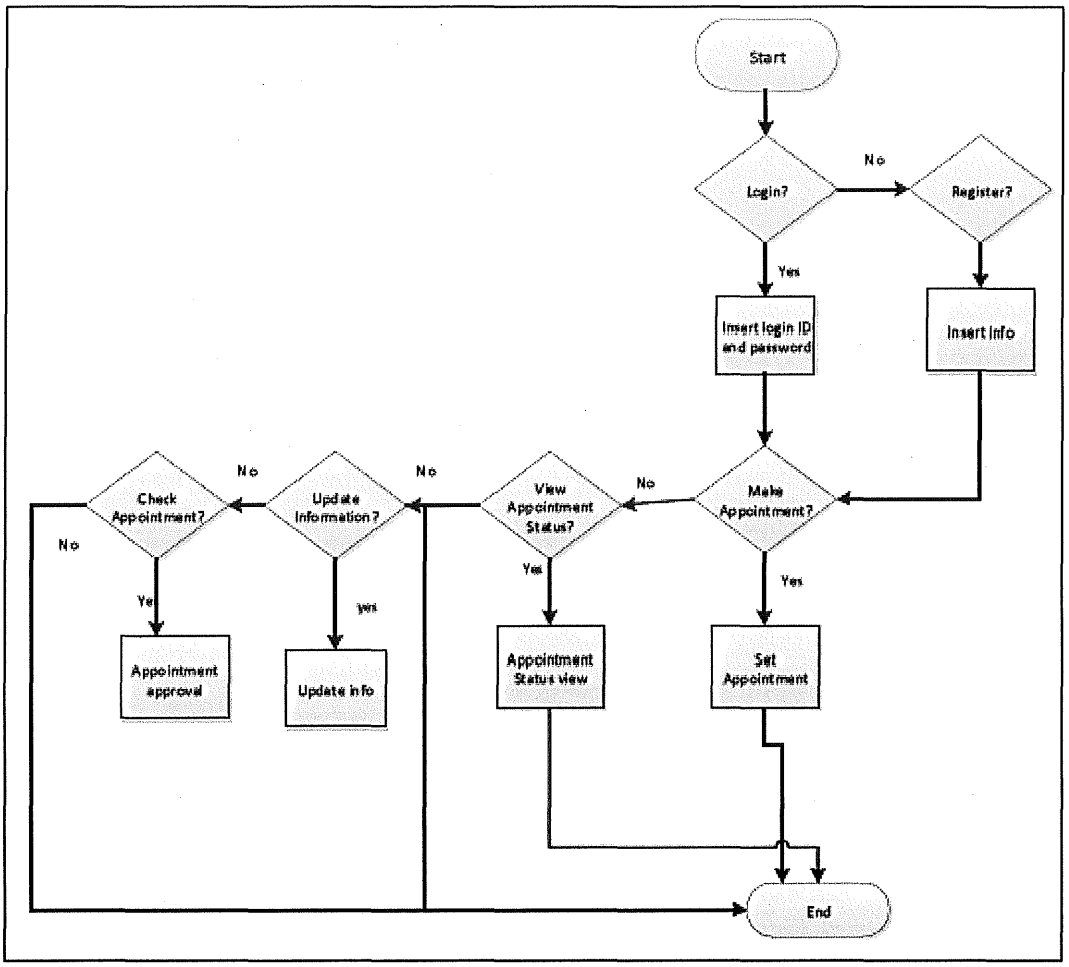


Figure 3.1 Flow Chart of OASF

Figure 3.1 is the flowchart of OASF. User will either choose to login or register. For registered user, they can just login to use the system. For new user, they have to register first. Then, student can start making an appointment or choose to view appointment status or update information. For lecturers, can view appointment status, update information and check appointment that needs approval.

3.3.2.1.2. Context Diagram

A context diagram is a system's highest level of view which is also known as Level 0 of a data flow diagram. It generally defines the limit between the system's environment and it shows the entities that are related with each other. The diagram below shows the Context Diagram of OASF to show the interaction of the system with users.

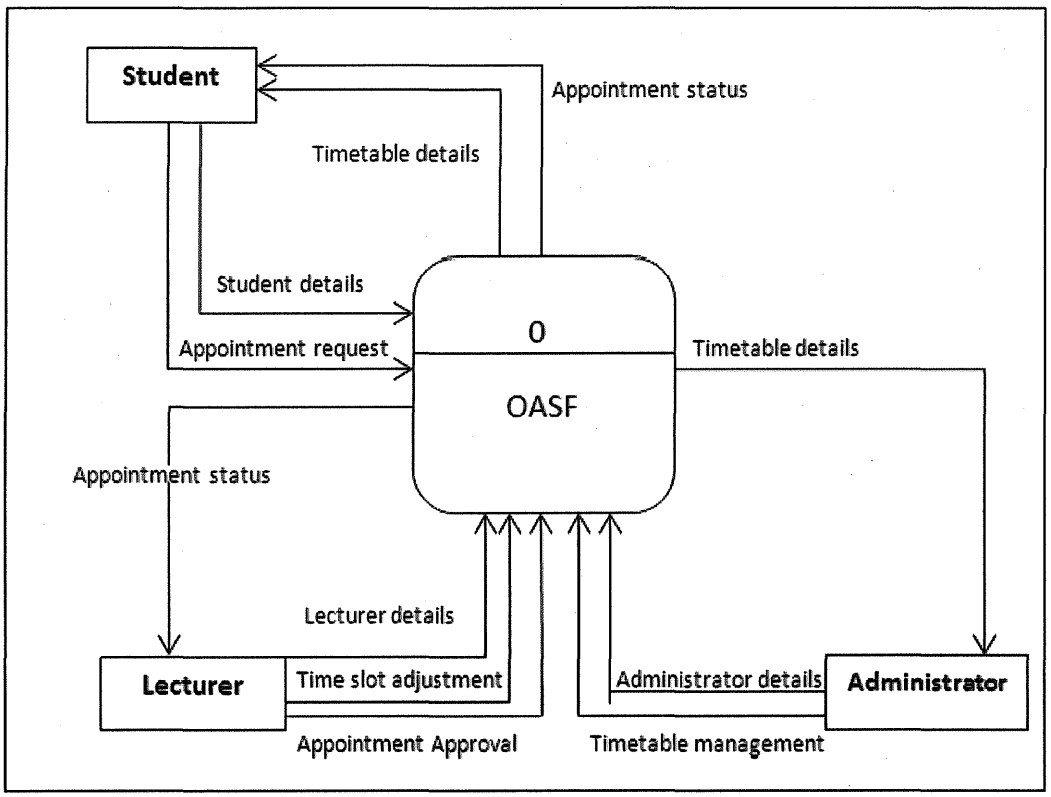


Figure 3.2: The Context Diagram of OASF

The arrow pointing towards the system means user will have to give input or fill in the form to be sent to the system whereas the arrow pointing towards the user is what the system will show for user's view.

In this system, students will register and update their details. Student can make appointment as well. Besides that, students can check lecturer's availability and check requested appointment status. For lecturers, they can update personal details, update timetable based on availability and approve appointment requested by students. Then, they can view the appointment statuses that have been approved. Administrator can update his own information and manage timetable also view the timetable details.

3.3.2.1.3. Data Flow Diagram Level 0

Data flow diagram (DFD) shows flows that occur in the system in the simplest form. It I represented in a graphical way of data flow and enables developer to show the processes that occurs in the system. The purpose of DFD is to help system developer or designers initiating the analysis stages of current or new system in meeting the requirements. Principally, it shows where data comes from then where it goes and also where it will be stored. Below is the DFD of OASF of the entities and data that will be stored.

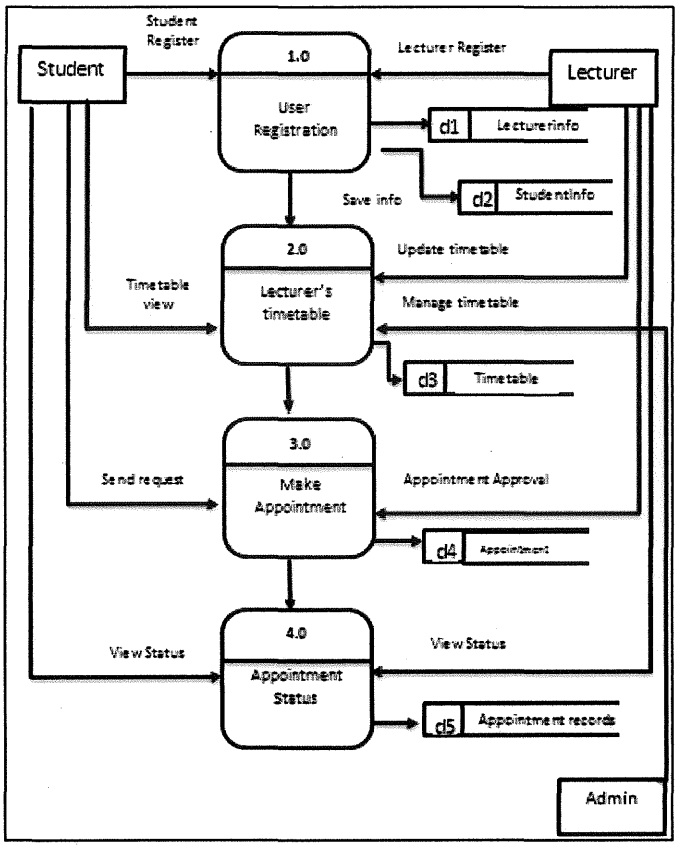


Figure 3.3 Data Flow Diagram Level 0

During process 1.0, this is where registration of students and lecturers takes place. Students will need to register themselves using student ID as username and admin will register for lecturer. When registration process is successful, data of student will be stored in d1 and data of lecturer will be stored in d2.

Then in process 2.0, administrator is able to manage timetable for lecturers while lecturers can update the timetable if there are changes to be made and students can view the timetable before making an appointment.

In process 3.0, students can make an appointment and data will be stored in d4. Then lecturer can check the appointment requested for approval and students.

In process 4.0, the appointment records will be stored in d5 data storage. And at this point, lecturer can view the appointment records and students can check the status of the appointment.

3.3.2.1.4. Entity Relationship Diagram

Entity Relationship Diagram represents the relationships between entities and how data is shared between each other in database. Normally, ERDs use symbols to represent three different types of information which are boxes, diamonds and oval. Below is the ERD for OASF.

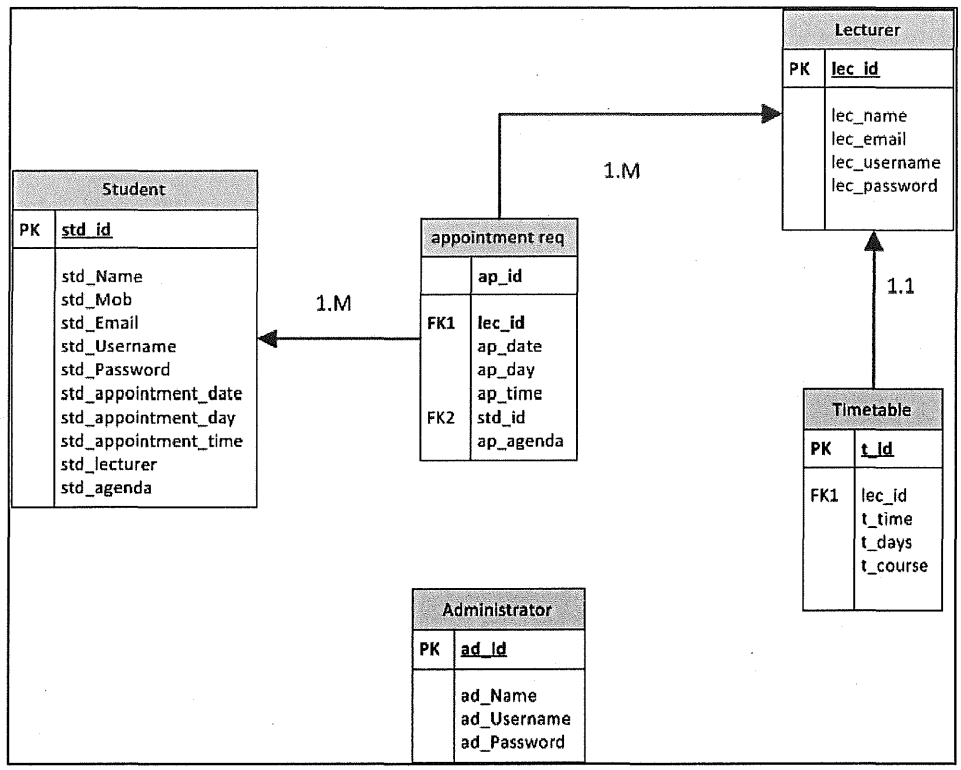


Figure 3.4 Entity Relationship Diagram

There five database tables proposed for The Online Appointment System for FSKKP Lecturers and Students (OASF) which are Lecturer, Student, Administrator and Timetable databases. Below are the data dictionaries of database tables for this system.

Data Field	Data Type	Length	Description	Constraint
ad_Id	varchar	10	Admin id	PK
ad_Name	varchar	10	Admin name	
ad_Username	varchar	50	Admin username	
Ad_password	varchar	20	Admin password	

Table 3.1 Data Dictionary for Admin table

Data Field	Data Type	Length	Description	Constraint
lec_id	int()	11	User id	PK
lec_name	text		Username of user	
lec_email	text		User full name	
lec_username	text		User phone number	
lec_password	text		User's password	

Table 3.2 Data Dictionary for Lecturers table

Data Field	Data Type	Length	Description	Constraint
std_ID	int()	11	User id	PK
std_Name	varchar	32	User full name	
std_mob	varchar	30	User phone number	
std_Email	varchar	50	User email	
std_Username	varchar	50	Username of user	
std_Password	varchar	50	User's password	
std_appointment_date	text		Appointment date	
std_appointment_day	text		Appointment day	
std_appointment_time	text		Appointment time	
std_lecturer	text		Lecturer that student make appointment with	
std_agenda	text		Agenda of the appointment	

Table 3.3 Data Dictionary for Student table

Data Field	Data Type	Length	Description	Constraint
ap_Id	int	11	Id of the appointment	PK
lec_id	int	11	Id of the lecturer	FK1
std_ID	text	50	The student id	FK2
ap_date	text	20	The date of appointment	
ap_day	text	10	The day of appointment	
ap_time	text		The time of appointment	
ap_agenda	text		The agenda of appointment	

Table 3.4 Data Dictionary for appointment_req table

Data Field	Data Type	Length	Description	Constraint
t_id	int	11	The id for timetable	PK
lec_id	int	11	The lecturers ID	FK1
t_time	text	50	Time slot	
t_days	text	20	Day of time slot	
t_course	text	10	Subject in time slot	

Table 3.5 Data Dictionary for Timetable table

3.3.2.2. The Proposed Interface Design

3.3.2.2.1. Login page

Figure 3.5 Login page

The interface in Figure 3.5 shows the login page of OASF. On this page, user will sign in with username and password. The username for student is student ID and for lecturer is the staff ID. Administrator will have their own username and password. By selecting category, user will be directed to the user mode. Administrator is responsible for lecturers' registration and students will have to register themselves.

3.3.2.2.2. Registration Page

Administrator is responsible for lecturers' registration and students will have to register themselves. This is to lessen the work load for administrator as there are more students than lecturers to register.

3.3.2.2.3. Student Registration Page

Figure 3.6 Student's Registration Page

Figure 3.6 shows the student's registration page where students need to fill in their information. Administrator will register for lecturers and they still can update their information later. Therefore only registered user can access to make appointment.

3.3.2.2.4. Lecturer's Timetable page

ONLINE APPOINTMENT SYSTEM FOR FSKKP

Home|Appointment|Status|Profile|

select lecture WAN NURULSAFAWATI BINTI WAN MANAN

Day\Time	8.00-9.00	9.00-10.00	10.00-11.00	11.00-12.00	12.00-13.00	13.00-14.00	14.00-15.00	15.00-16.00	17.00-18.00
Monday	Lecture	Lecture							
Tuesday			Lecture	Lecture					
Wednesday			Lecture	Lecture					
Thursday	Lecture	Lecture						Lecture	Lecture
Friday	on leave	on leave	on leave	on leave	on leave	on leave	on leave	on leave	on leave

Proceed With Appointment | Cancel

Figure 3.7 Lecturer's Timetable Page

Figure 3.7 shows the lecturer's timetable that student can view before making an appointment. By selecting the lecturer's name, system will show the selected lecturer's timetable. This feature is useful to make student decide whether they want to proceed with the appointment or cancel the operation.

3.3.2.2.5. Appointment page

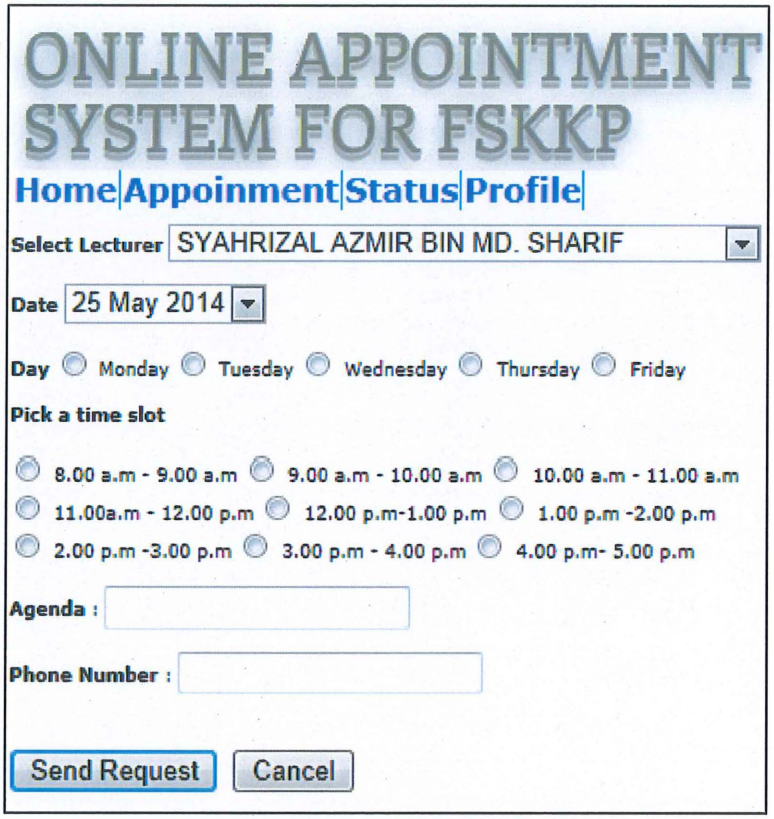


Figure 3.8 Appointment Page

Figure 3.8 shows the appointment page. Student will have to pick their desired time slot to make appointment with the lecturer. If student choose time slot which lecturer is unavailable or already booked, the system will suggest other time slot available for the student to pick. This is to avoid reiteration of appointment

3.3.3. Development phase

Development phase is where the building of system begins .The development is based on the design planned in the second phase. During this phase, developers will focus on coding because it is longest phase that needs more time to meet the user's satisfaction. Besides that, user will be participating in the development of the system. This will make it easier if there are changes need to be made in the system. This phase includes coding, integration and testing.

For this system, the source code used to implement the design is HTML and CSS which is for the interface design and PHP: Hypertext Preprocessor as the server scripting language. Database design will be involved in this phase. Complete system architecture and functions will be formed with the combination of PHP source code and MySQL.

During integration or installation phase, the system will be ready to be used by user of OASF. Each individual of users will be combined together as a unit. For this system there will be a lecturer, a student and an administrator to test the system together.

During testing, There are testing requirements that need to be tested the unit will inform developer the errors and problems that exist in the system. Then the developer will find solution to overcome the errors and failure of the system by modifying the codes. For this system, developer should ensure that the system works smoothly. The first thing is to make sure the database interactions are correct. Secondly, to make sure the system can suggest other time slot if student picked the wrong time. Thirdly is to make sure appointments can be made successfully.

3.3.4. Cutover Phase

During Cutover phase, the final tasks in SDLC should be executed. The tasks are testing, data conversion, maintenance and user training.

The system should be ready to be used and tested by users and developer will clarify the users about how this system works. This is to certify user's knowledge on how the system operates. Besides that, when users of OASF understands the system fully, they will give feedback on how they feel about the system. For example when a lecturer dislikes the user interface, developer can still improve the systems interface and do support on maintenance if there are problems that occur during while using the system.

3.4 System Requirements

According to Wikipedia, the purpose of system requirements is "To be used efficiently, all computer software needs certain hardware components or other software resources to be present on a computer" (Wikipedia, 2015). System requirements are divided into two parts which are hardware and software requirement.

3.4.1. Hardware Requirements

Below is the list of hardware requirements for the system development

I. Laptop

- Windows 7 or above
- Minimum 2GB Random Access Memory (RAM) or above
- Minimum 250 GB Hard Disk Storage

II. External Hard Disc

→ For backup in case something happens to the internal hard disc of the laptop that is used for the project development.

III. Cable/wireless network

→ For internet access

3.4.2. Software Requirements

Software requirements or software specification requirements (SRS) are a detail description of the software environment that is required in the development of the system. This requirement specification is important to make sure the system will work in a good performance. Bad requirements will result in a lot of problems such as slowing down the system's development, execution failure and many more.

The software requirements specifications of the system are as listed below.

Software	Purpose
1. Microsoft Office (Microsoft Word, Microsoft Visio, Microsoft Project)	• Used to make word processing, Gantt chart and draw design diagrams
2. Adobe Dreamweaver CS	• For user interface design and development of the system
3. Adobe Photoshop CS6	• For creating and editing of image
4. PhpMyAdmin	• A public-domain open source Web server
5. Apache	• Database over the Web of the administration handling

3.5. Conclusion

Methodology can be used as a direction or guide for this system. The steps in methodology help to make sure the process of making this project is always on track until it is completed timely. By using SDLC Rapid Application Development (RAD) Model, the requirements for this project can be understood a better during the early phase of the project cycle. Design for this system is also proposed to show the developer and user the interface and structure of this system.

CHAPTER 4

IMPLEMENTATION

4.0 Overview

During the implementation phase, it is a process to perform a system result and analysis from previous phase into an executable system. This includes the development of graphical user interface design (GUI), database design and the engine program which implemented with programming codes. This phase will combine with those three criteria to make the system can running and using. The related function and module consist on this project will be explained on this phase.

Generally, this chapter will explain about project development that is designed and also phase also to show the process of user interface and system has been developed. The explanation on source code and interface created can make the more understanding on how it works. This system there are a number of modules which will make it will function more smoothly.

4.1. Design

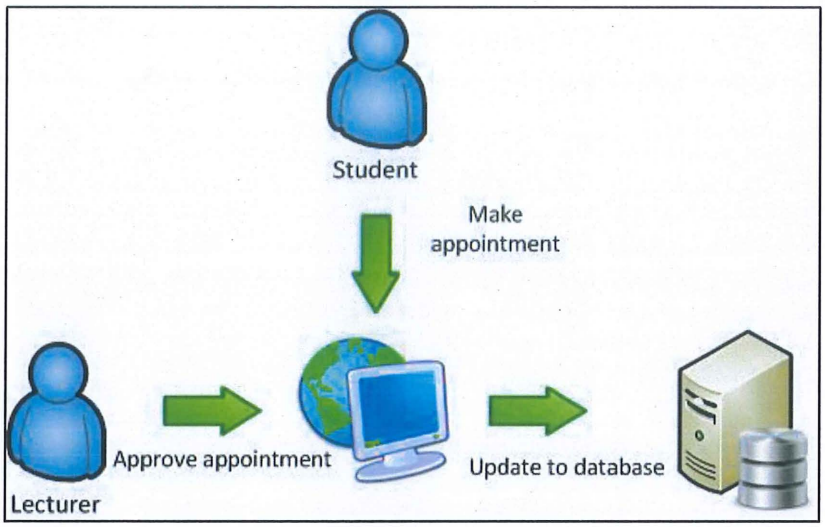


Figure 4.10 OASF Architecture

OASF contains there different modules which are admin, lecturer and student. Each module is for different type of users. Admin will set the default timetable for registered lecturers. Lecturers can edit the timetable according to their availability while students can make appointment based on registered lecturers. Therefore every lecturers and student of FSKKP need to register first.

Lecturers will need to approve the students' request to acknowledge the students whether their request is accepted or rejected. Students can check the appointment's status and meet the lecturer.

4.2. Database Design

Generally database is known as a collection of organized information that is easy be accessed, managed, and updated. OASF uses database to store all the information needed for the system to work such as student and lecturer information, records of appointments and also lecturers' timetable.

4.2.1. Data Dictionary

Data dictionary defines the structure of a database. In a data dictionary, there is a list of all files in a database, the number of records in each file, and the names and types of each field. It is used to control and act as the maintenance of large databases. These are the data dictionaries in OASF.

Below are table 4.1, 4.2, 4.3, 4.4 and 4.5 show the data dictionary for the admin, lecturer , student , appointment and timetable . Data dictionary does not display the actual data but only show the data field, data type, length and constraint

Data Field	Data Type	Length	Description	Constraint
ad_Id	varchar	10	Admin id	PK
ad_Name	varchar	10	Admin name	
ad_Username	varchar	50	Admin username	
Ad_password	varchar	20	Admin password	

Table 4.1 Data Dictionary for Admin table

Data Field	Data Type	Length	Description	Constraint
lec_id	int()	11	User id	PK
lec_name	text		Username of user	
lec_email	text		User full name	
lec_username	text		User phone number	
lec_password	text		User's password	

Table 4.2 Data Dictionary for Lecturers table

Data Field	Data Type	Length	Description	Constraint
std_ID	int()	11	User id	PK
std_Name	varchar	32	User full name	
std_mob	varchar	30	User phone number	
std_Email	varchar	50	User email	
std_Username	varchar	50	Username of user	
std_Password	varchar	50	User's password	
std_appointment_date	text		Appointment date	
std_appointment_day	text		Appointment day	
std_appointment_time	text		Appointment time	
std_lecturer	text		Lecturer that student make appointment with	
std_agenda	text		Agenda of the appointment	

Table 4.3 Data Dictionary for Student table

Data Field	Data Type	Length	Description	Constraint
ap_Id	int	11	Id of the appointment	PK
lec_id	int	11	Id of the lecturer	FK1
std_ID	text	50	The student id	FK2
ap_date	text	20	The date of appointment	
ap_day	text	10	The day of appointment	
ap_time	text		The time of appointment	
ap_agenda	text		The agenda of appointment	

Table 4.4 Data Dictionary for appointment_req table

Data Field	Data Type	Length	Description	Constraint
t_id	int	11	The id for timetable	PK
lec_id	int	11	The lecturers ID	FK1
t_time	text	50	Time slot	
t_days	text	20	Day of time slot	
t_course	text	10	Subject in time slot	

Table 4.5 Data Dictionary for Timetable table

4.2.2. SQL Statement

This system required many queries table such as admin, timetable, appointment, student and lecturer. The entire database table must be connected to display the information through interface that created using Adobe Dreamweaver CS6 and PHP. This part will explain about the table detail that applied on OASF.

Table	Action
admin	Browse Structure Search Insert Empty Drop
appointment_req	Browse Structure Search Insert Empty Drop
lecturers	Browse Structure Search Insert Empty Drop
student	Browse Structure Search Insert Empty Drop
timetable	Browse Structure Search Insert Empty Drop
5 tables	Sum

Figure 4.11 Tables created on OASF.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	t_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	lec_id	int(11)			No	None		Change Drop Primary
3	t_time	text	latin1_swedish_ci		No	None		Change Drop Primary
4	t_days	text	latin1_swedish_ci		No	None		Change Drop Primary
5	t_course	text	latin1_swedish_ci		No	None		Change Drop Primary

Figure 4.12 Table for timetable

```
SELECT * FROM `timetable`
```

Figure 4.13: SQL command to display all information on the table

```
INSERT INTO `timetable`(`t_id`, `lec_id`, `t_time`, `t_days`, `t_course`) VALUES ([value-1],[value-2],[value-3],[value-4],[value-5])
```

Figure 4.14: SQL command to insert information on the table

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ap_Id	int(11)			No	*None*	AUTO_INCREMENT	Change Drop Primary
2	lec_id	int(11)			No	*None*		Change Drop Primary
3	std_ID	int(11)			No	*None*		Change Drop Primary
4	ap_date	text	latin1_swedish_ci		No	*None*		Change Drop Primary
5	ap_day	text	latin1_swedish_ci		No	*None*		Change Drop Primary
6	ap_time	text	latin1_swedish_ci		No	*None*		Change Drop Primary
7	ap_agenda	text	latin1_swedish_ci		No	*None*		Change Drop Primary

Figure 4.15 Table for appointment

```
SELECT * FROM `appointment_req` WHERE ap_id=1
```

Figure 4.16: SQL command to display all information on the table of a row

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ad_Id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	ad_Name	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary
3	ad_Username	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary
4	ad_Password	varchar(255)	latin1_swedish_ci		No	None		Change Drop Primary

Figure 4.17 Table for admin

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	lec_id	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	lec_name	text	latin1_swedish_ci		No	None		Change Drop Primary
3	lec_email	text	latin1_swedish_ci		No	None		Change Drop Primary
4	lec_username	text	latin1_swedish_ci		No	None		Change Drop Primary
5	lec_password	text	latin1_swedish_ci		No	None		Change Drop Primary

Figure 4.18 Table for lecturers

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	std_ID	int(11)			No	None	AUTO_INCREMENT	Change Drop Primary
2	std_Name	varchar(32)	latin1_swedish_ci		No	None		Change Drop Primary
3	std_mob	varchar(30)	latin1_swedish_ci		No	None		Change Drop Primary
4	std_Email	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary
5	std_Username	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary
6	std_Password	varchar(50)	latin1_swedish_ci		No	None		Change Drop Primary
7	std_appointment_date	text	latin1_swedish_ci		No	None		Change Drop Primary
8	std_appointment_day	text	latin1_swedish_ci		No	None		Change Drop Primary
9	std_appointment_time	text	latin1_swedish_ci		No	None		Change Drop Primary
10	std_lecturer	text	latin1_swedish_ci		No	None		Change Drop Primary
11	std_agenda	text	latin1_swedish_ci		No	None		Change Drop Primary

Figure 4.19 Table for student

```
SELECT * FROM `student` WHERE std_USername='ca11069'
```

Number of rows: 25

+ Options

←T→						std_ID	std_Name	std_mob	std_Email	std_Username
☐	Edit	Copy	Delete			14	LILIANA BINTI AZMI	016-86730484	liliana@gmail.com	ca11069

Figure4.20: SQL select information of a student

4.3. Coding Explanations

This system is developed by using html for the interface and PHP to connect to the database. The platform used was Adobe Dreamweaver CS6 Then, the source code file is stored in web servers called apache for simulator to test this system.

There are many pages that was developed using source codes to make this system functioning and can carry out the process to reach the objective

4.3.1. Login Page

In order to access into the system, users need to login first to make sure that only registered user can access into this system. This means all users need to register first before login in into the system.For this system, it allows admin, lecturer and students to login into the system. Both student and lecturer will use their staff id or student id as username to login. The password for lecturers will be set by admin based on the combination of staff id and the first letter of their name but for students, they will need to register themselves and create their own password.

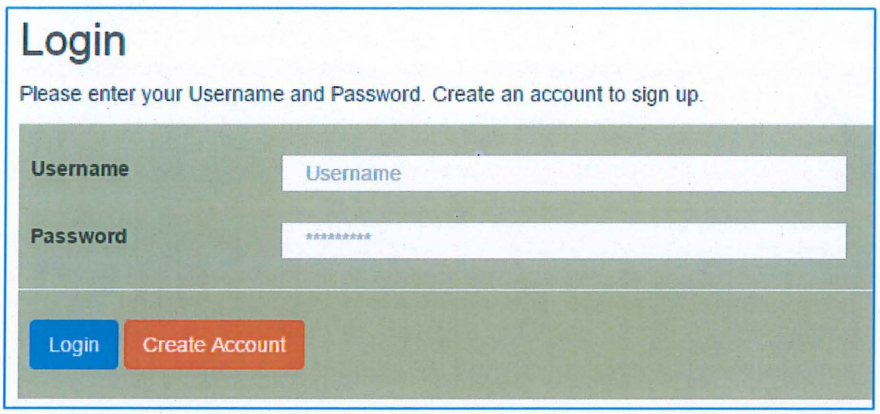


Figure 4.21 The login interface

The login page is developed by using the login form in 'login.php' file. Then , the action of the process will be passed from form the 'login_action.php' file. On login_action.php' file , the source code that has connection with database is written to ensure that the data entered by user matches the data stored in the database.

```
<?php


 //require "session_admin.php";

if(isset($_POST['submit']))
   {

        session_start();
        require "connect.inc.php";

        $username = $_POST['username'];
        $password=  $_POST ['password'];

        $sql = "Select std_Username , std_Password from student WHERE std_Username='".$username."' AND std_Password= '".$password."'" ;
        $retval = mysql_query($sql);

        $sql1 = "SELECT  lec_Username , lec_Password FROM lecturers WHERE lec_Username= '".$_POST['username']."' AND lec_Password = '".$_POST['password']."' ";
        $retval1 = mysql_query($sql1);

        $sql2 = "SELECT  ad_Username , ad_Password FROM admin WHERE ad_Username= '".$_POST['username']."' AND ad_Password = '".$_POST['password']."' ";
        $retval2 = mysql_query($sql2);

                        if ( $row = mysql_fetch_array($retval,MYSQL_ASSOC)){

                                        $_SESSION['current_user'] = $username;
                                        header('Location: http://localhost/OASF/faculty_appointment.php');
                        }

                       else if ( $row1 = mysql_fetch_array($retval1,MYSQL_ASSOC)){
```

Figure 4.22 PHP source codes of login interface

```
$_SESSION['f_user'] = $_POST['username'];
$_SESSION['temp']="";
header('Location: http://localhost/OASF/appointment_notification.php');

}


 else if ( $row2 = mysql_fetch_array($retval2,MYSQL_ASSOC)){

$_SESSION['admin'] = $_POST['username'];
$_SESSION['temp']="";
header('Location: http://localhost/OASF/admin.php');
}

else
{
            session_start();
            $_SESSION['error'] = "Wrong Username or Password, Please Try Again";
            header('Location: http://localhost/OASF/login.php');

}



}

?>
```

Figure 4.23 PHP source codes of login interface

4.3.2. Appointment Page

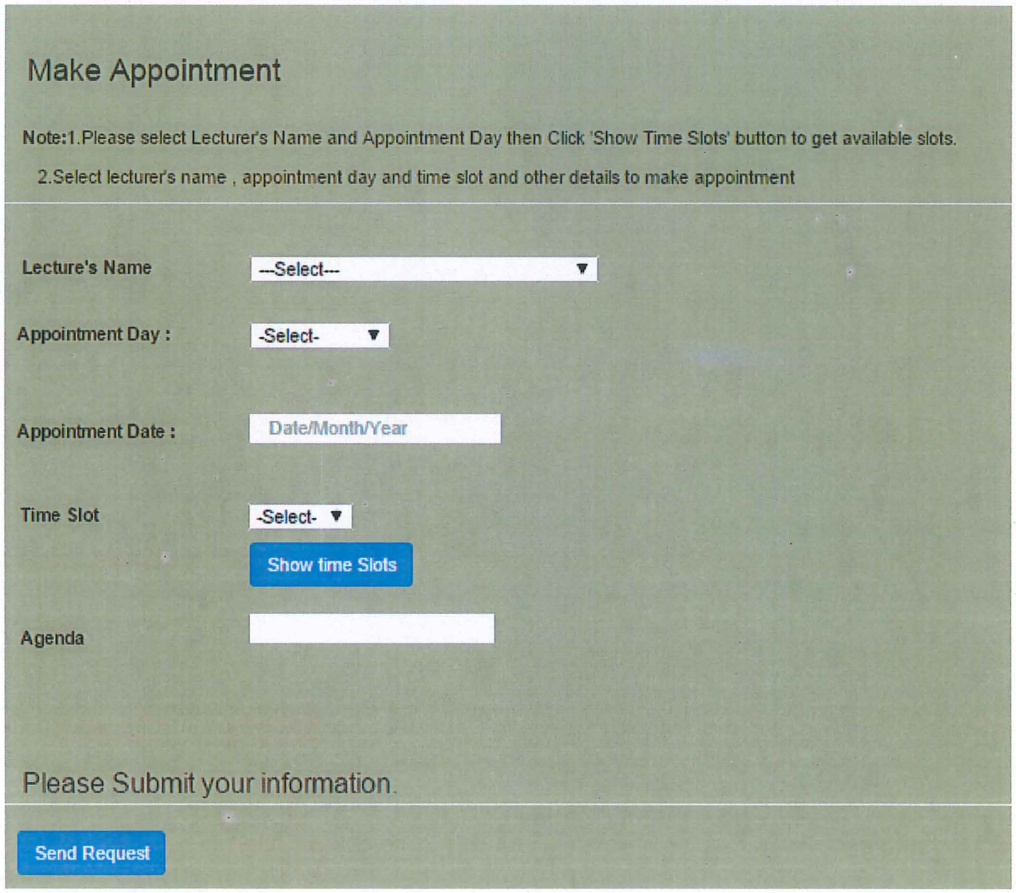


Figure 4.24 the appointment interface

Figure 4.24 shows the interface of the form used for students to make appointment with lecturer. Students will have to select lecturer and the day they want to meet the lecturer and click on 'Show Time Slots' button. Then the system will display time slot that is empty or written 'empty' on the lecturer's schedule. The action for this form in taken in 'faculty_appointment_action.php' as in figure 4.25.

```
<?php
if(isset($_POST['submit']))
{
    require "session_user.php";
    require 'connect.inc.php';

       $sql = mysql_query("select * from lecturers,student WHERE lec_name='".$_POST['lec_name']."'
       AND std_username='".$_SESSION['current_user']."'");

                                                    while($row=mysql_fetch_array($sql))
                                                     {
                                                        $lec_id= $row['lec_id'];
                                                        $std_ID= $row['std_ID'];


    $query = "INSERT INTO appointment_req(lec_id,std_ID,ap_date,ap_day,ap_time,ap_agenda)
    VALUES('$lec_id','$std_ID','$_POST[ap_date]','$_POST[ap_day]','$_POST[ap_time]','$_POST[ap_agenda]')" ;


    if ($query_run = mysql_query($query))
     {

      $_SESSION['data_success']= "Appointment Request Send to Lecturer!";

        header('Location: http://localhost/OASF/faculty_appointment.php');
     }
    else {


      $_SESSION['data_error']= "Data Failed to Insert!";
         header('Location: http://localhost/OASF/faculty_appointment.php');
    }



}

}


?>
```

Figure 4.25 Source code for appointment page process.

4.3.3. Appointment Request

Figure 4.24 Interface of Student Appointment request

After all appointments request has been submitted, when lecturer login into the system, they will see appointment request that needs approval as in Figure 4.24. Lecturer will decide whether to approve or reject the request. This will be displayed on 'appointment_notification.php' file as shown in figure 4.25 and figure 4.26. The process of the approving request is in the 'appointment_notification_action.php' file is shown in figure 4.27.

```
<?php
require 'connect.inc.php';
if(isset($_POST['approve']))
{

$query = "UPDATE student
        SET std_appointment_date='".$_POST['ap_date']."', std_appointment_day='".$_POST['ap_day']."', std_appointment_time='".$_POST['ap_time']."',std_lecturer='".$_POST['lec_name']."',std_agenda='".$_POST[
'ap_agenda']."'

        WHERE std_ID='".$_POST['std_ID']";

$query_add="UPDATE timetable
        SET t_course='".$_POST['std_Name']."\n Agenda:".$_POST['ap_agenda']."
        WHERE t_time='".$_POST['ap_time']."' AND t_day='".$_POST['ap_day']."';

$query_delete = "Delete From appointment_req
                WHERE ap_id='".$_POST['ap_id']."';

    if ($query_run = mysql_query($query) && $query_run1 = mysql_query($query_add) && $query_run2 = mysql_query($query_delete) )
    {

        echo "good";
       session_start();
      $_SESSION['data_success']= "Appointment Approved !";
      header('Location: http://localhost/OASP/appointment_notification.php');

    }

    else {
        session_start();
      $_SESSION['data_error']= "Error in Approving!";
      header('Location: http://localhost/OASP/appointment_notification.php');

    }

}


else


{
$query1 = "Delete From appointment_req

        WHERE ap_id='".$_POST['ap_id']."';

    if ($query_run = mysql_query($query1))
    {

      session_start();
      $_SESSION['data_success']= "Appointment Rejected Successfully !";
      header('Location: http://localhost/OASP/appointment_notification.php');

    }

    else {

        session_start();
      $_SESSION['data_error']= "Error in reject!";
      header('Location: http://localhost/OASP/appointment_notification.php');

    }


}
?>
```

Figure 4.27 source code for the appointment request action.

4.3.4. Approved Appointment

Figure 4.28 Interface of approved appointment

After appointment has been approved, student will see the appointment details in the 'Approved Appointment' tab. So if students do not see the details of the appointments made, it means that the appointment is not approved yet or

rejected. The page will be displayed on 'appointment_aproval.php' page and also the deletion process also takes place here and the source code is shown in figure 4.29.

```
<?php if (isset($_SESSION['data_success'])){ ?>
            <div class="panel panel-success">
                        <div class="panel-heading">
                        <h3 class="panel-title">Success!!</h3>
                        </div>
                       <div class="panel-body">

                            <?php echo $_SESSION['data_success'];?>
                       </div>
                  </div>
            <?php }
              else if(isset($_SESSION['data_error'])) { ?>
              <div class="panel panel-danger">
                        <div class="panel-heading">
                        <h3 class="panel-title"> Error!! </h3>
                        </div>
                       <div class="panel-body">

                            <?php echo $_SESSION['data_error'];?>
                       </div>
                  </div>


              <?php } else {} ?>
    <?php
          require "connect.inc.php";

                        $sql = mysql_query("SELECT * from student");

                                  while($row=mysql_fetch_array($sql))
                                  {
                                     $std_appointment_date= $row['std_appointment_date'];
                                     $std_appointment_day= $row['std_appointment_day'];
                                     $std_appointment_time= $row['std_appointment_time'];
                                     $std_lecturer= $row['std_lecturer'];
                                     $std_agenda= $row['std_agenda'];
                                      $std_ID= $row['std_ID'];
 ?>
<?php if (isset($_SESSION['data_success'])){ ?>
            <div class="panel panel-success">
                        <div class="panel-heading">
                        <h3 class="panel-title">Success!!</h3>
                        </div>
                       <div class="panel-body">

                            <?php echo $_SESSION['data_success'];?>
                       </div>
                  </div>
            <?php }
              else if(isset($_SESSION['data_error'])) { ?>
              <div class="panel panel-danger">
                        <div class="panel-heading">
                        <h3 class="panel-title"> Error!! </h3>
                        </div>
                       <div class="panel-body">

                            <?php echo $_SESSION['data_error'];?>
                       </div>
                  </div>


              <?php } else {} ?>
    <?php
          require "connect.inc.php";

                        $sql = mysql_query("SELECT * from student");

                                  while($row=mysql_fetch_array($sql))
                                  {
                                     $std_appointment_date= $row['std_appointment_date'];
                                     $std_appointment_day= $row['std_appointment_day'];
                                     $std_appointment_time= $row['std_appointment_time'];
                                     $std_lecturer= $row['std_lecturer'];
                                     $std_agenda= $row['std_agenda'];
                                      $std_ID= $row['std_ID'];
 ?>
<?php if($std_lecturer != "" && $std_agenda!="" && $std_appointment_date!="" && $std_appointment_day!="" && $std_appointment_time!="")
 {


 ?>
<div class="ahmed" style="background-color:#E8E2A1; position:relative; top:40px;">

<article class="uploadform">
<br>

    <h3>   Approved Appointment </h3>


        <fieldset id="personalinfo">
        <legend>  </legend>


        <div style="position:relative; left:20px;">


        <label>Lecturer's Name : </label>    <b><?php echo $std_lecturer;?> </b><br>
        <label>Appointment Agenda: </label>    <b><?php echo $std_agenda;?></b><br>
        <label>Appointment Date: </label>    <b><?php echo $std_appointment_date;?></b><br>
        <label>Appointment Day : </label>    <b><?php echo $std_appointment_day;?></b><br>
        <label>Appointment Time : </label>    <b><?php echo $std_appointment_time;?></b><br>

        </div>
        <form action="$" method="post">
         <input type="hidden" value="<?php echo $std_ID;?>" name="std_ID">


         <br><br>

          <input type="submit"  value="Delete" name="Delete" class="btn btn-danger">    
        <br><br>
       </form>

<?php

require 'connect.inc.php';
if(isset($_POST['Delete']))
{

$query = 'UPDATE student
        SET std_appointment_date="", std_appointment_day="", std_appointment_time="",std_lecturer="",std_agenda=""

        WHERE std_ID='.$_POST['std_ID'];

        if ($query_run = mysql_query($query))
      {


        $_SESSION['data_success']= "Delete Successfull !";
       // header('Location: http://localhost/OASF/appointment_aproval.php');

      }

      else {


        $_SESSION['data_error']= "Error in Deleting!";
       // header('Location: http://localhost/OASF/appointment_aproval.php');

      }

}


?>
```

Figure 4.29 Source code of displaying approved appointment and deletion of data

4.3.5. Update Schedule Page

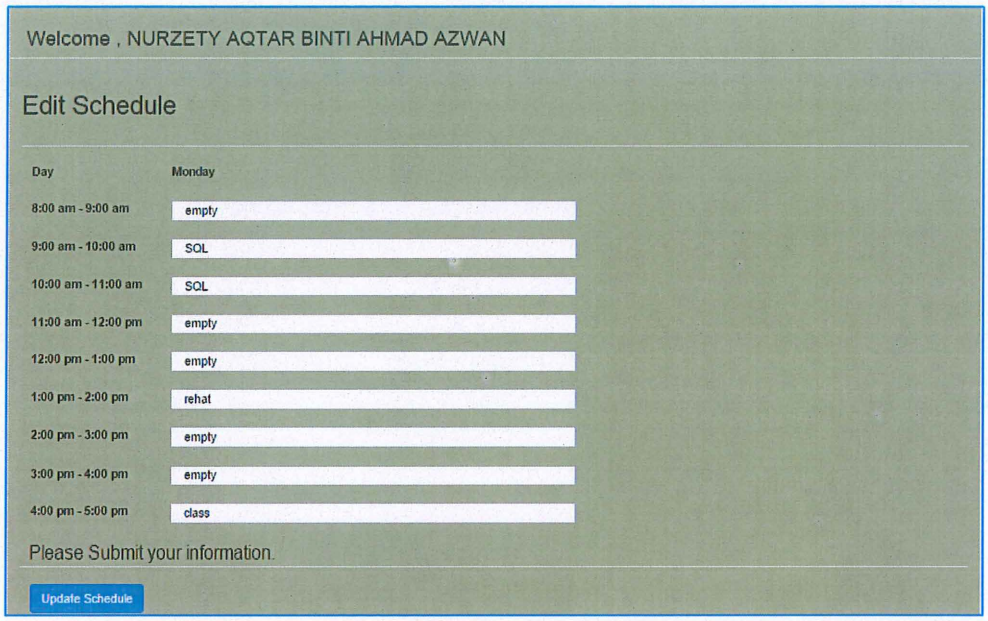


Figure 4.30 Edit schedule interface

This is the interface for lecturers to update their schedule where they can change their timetable on time slot and set their availability by changing on the text field and update their schedule. The file name for this form is 'timetable_edit.php' and the process for this form is carried by codes on 'timetable_edit_action.php' file.

```
<?php
    require "connect.inc.php";
?>
<?php
    $day = $_REQUEST['day'];
    $lec_id= $_REQUEST['lec_id'];
    $sql = mysql_query("SELECT
a.t_course,a.t_time , b.lec_username FROM timetable a, lecturers b  WHERE
a.lec_id=b.lec_id AND a.t_days ='".$day."' AND b.lec_username='".$_SESSION['f_user']."'"
);
    $result_empty=array("","","","","",
"","","","");

    while($row=mysql_fetch_array($sql))
    {
        $result[] = $row['t_course'];
        $time = $row['t_time'];
        $combine=  $result+
$result_empty;
    }
?>
```

Figure 4.31 source code for update schedule process in 'timetable_edit.php'

```php
<?php
require "connect.inc.php";
require "session_faculty.php";

 if(isset($_POST['update']))
 {

 $day = $_POST['day'];
 $lec_username= $_SESSION['f_user'];
$courses=array("8-9","9-10","10-11","11-12","12-1","1-2","2-3","3-4","4-5");
$time=array("8:00 am - 9:00 am","9:00 am - 10:00 am","10:00 am - 11:00 am","11:00 am - 12:00 pm","12:00 pm - 1:00 pm","1:00 pm - 2:00 pm",
"2:00 pm - 3:00 pm","3:00 pm - 4:00 pm","4:00 pm - 5:00 pm");

/*print_r($_REQUEST);
die;
*/


$sql1 = 'UPDATE timetable
        SET t_course="'.$_POST['8-9'].'" WHERE t_time="8:00 am - 9:00 am" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';

$sql2 = 'UPDATE timetable
        SET t_course="'.$_POST['9-10'].'" WHERE t_time="9:00 am - 10:00 am" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';

$sql3 = 'UPDATE timetable
        SET t_course="'.$_POST['10-11'].'" WHERE t_time="10:00 am - 11:00 am" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';


$sql4 = 'UPDATE timetable
        SET t_course="'.$_POST['11-12'].'" WHERE t_time="11:00 am - 12:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';


$sql5 = 'UPDATE timetable
        SET t_course="'.$_POST['12-1'].'" WHERE t_time="12:00 pm - 1:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';


$sql6 = 'UPDATE timetable
        SET t_course="'.$_POST['1-2'].'" WHERE t_time="1:00 pm - 2:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';


$sql7 = 'UPDATE timetable
        SET t_course="'.$_POST['2-3'].'" WHERE t_time="2:00 pm - 3:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';


$sql8 = 'UPDATE timetable
        SET t_course="'.$_POST['3-4'].'" WHERE t_time="3:00 pm - 4:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';

 $sql9 = 'UPDATE timetable
        SET t_course="'.$_POST['4-5'].'" WHERE t_time="4:00 pm - 5:00 pm" AND t_days="'.$day.'" AND lec_id="'.$_REQUEST['lec_id'].'"';

$retval1 = mysql_query( $sql1);
$retval2 = mysql_query( $sql2);
$retval3 = mysql_query( $sql3);
$retval4 = mysql_query( $sql4);
$retval5 = mysql_query( $sql5);
$retval6 = mysql_query( $sql6);
$retval7 = mysql_query( $sql7);
$retval8 = mysql_query( $sql8);
$retval9 = mysql_query( $sql9);

if(!$retval1 && !$retval2 && !$retval3 && !$retval4 && !$retval5 && !$retval6 && !$retval7 && !$retval8 && !$retval9 )
{
  die('Could not update data: ' . mysql_error());
}
else
    {
        $_SESSION['update_success']= "Schedule Successfully Updated!";
         header('Location: http://localhost/OASF/timetable_edit.php?day='.$day.'&lec_id='.$_REQUEST['lec_id'].'&lec_name='.$_POST['lec_name']);

    }

 }
 ?>
```

Figure 4.32 source code for update schedule process in 'timetable_edit_action.php'

4.3.6. Student Registration Page

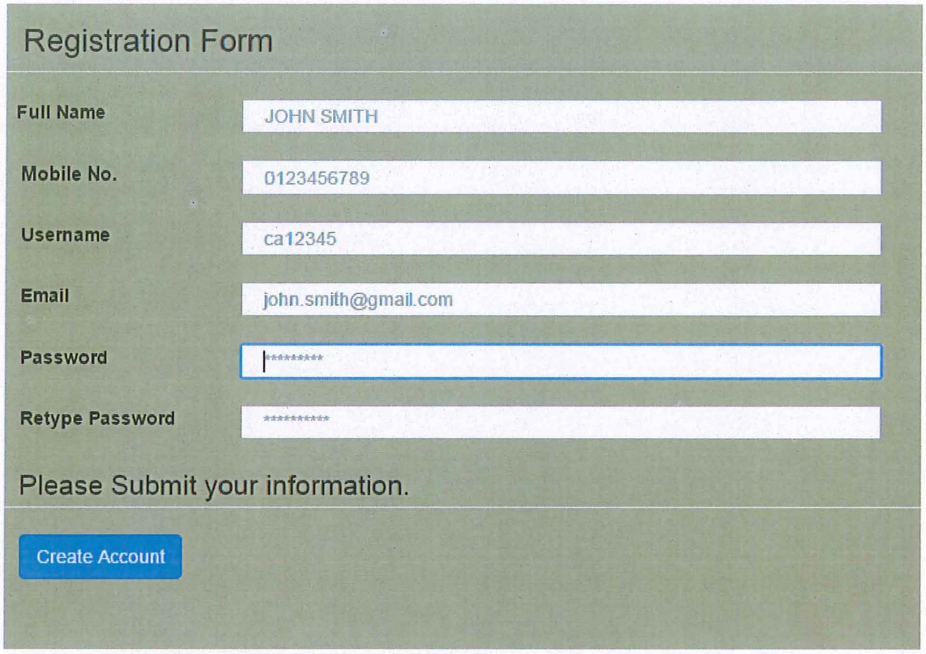


Figure 4.32 Registration form for students

Figure 4.26 shows the student registration form student can register to create and account to access into the system. Students will use student id as username which is will be used during login process. The form is in 'signup.php' file and the registration process will be carried in 'signup_action.php' as in figure 4.27.

```
<?php
if(isset($_POST['submit']))
{
    require 'connect.inc.php';


    $query = "INSERT INTO student(std_Name,std_mob,std_Email,std_Username,std_Password)
    VALUES('$_POST[fullname]','$_POST[mob_no]','$_POST[email]','$_POST[username]','$_POST[password]')" ;

    if ($query_run = mysql_query($query))
     {
      session_start();
      $_SESSION['data_success']= "Data Successfully Inserted!";

       header('Location: http://localhost/OASF/signup.php');
     }
    else {

        session_start();
        $_SESSION['data_error']= "Data Failed to Insert!";
        header('Location: http://localhost/OASF/signup.php');
    }



}


?>
```

Figure 4.33 source code for student's registration process

4.3.7. Lecturer Registration Page

Figure 4.34 Lecturer's registration form

Figure 4.28 shows the lecturer's registration form where each lecturers will be registered by admin. Admin needs to login to create an account for lecturer .Lecturers will use staff id as username which will be used during login process. The form is in 'admin.php' file and the registration process will be carried in 'lecturer_sign_action.php' as in figure 4.27.

```
<?php
if(isset($_POST['submit']))
{
    require 'connect.inc.php';


    $query = "INSERT INTO lecturers(lec_name,lec_email,lec_username,lec_password)
    VALUES('$_POST[fullname]','$_POST[email]','$_POST[username]','$_POST[password]')" ;

    if ($query_run = mysql_query($query))
     {

        $id= mysql_insert_id();

       header('Location: http://localhost/OASF/create_schedule.php?id='.$id);
     }
    else {


         header('Location: http://localhost/OASF/admin.php');
     }



}


?>
```

Figure 4.35 source code for student's registration process

CHAPTER 5

RESULT AND DISCUSSION

5.0 Overview

In this chapter, will be discussing the description of the result of the testing phase. For the purpose of future development of the system, advantages and constraints of the project will also be recognized. This chapter is also to shows the results and analysis of the data obtained. With the results obtained, the benefits and ideas also can be achieved to make sure the system runs in optimum level and a very and user friendly.Further changes of the system can be done by keep enhancing when suggested to be applied to the system for making the ease of use for users of the system.

OASF is developed by using the PHP language on a web based platform. The system works on a browser with internet connection. Students can view lecturer's availability on timetable and make appointment based on lecturer's free time. Lecturer can view the appointment request and give approval or reject.

5.1. Result

OASF can be used on web-based applications and second is by mobile web. This system is build using responsive HTML that it will make it suitable for all types of platform. This system is built using Rapid Application Development (RAD) methodology.

OASF is only a prototype and hopefully the result should meet the user requirement. The result is obtained by testing on three modules which are lecturer, student and admin. Each module has different function which however will affect the result on other modules. There are some advantages and disadvantages that can be improved for future development.

Figure 5.1 Home Page of OASF

5.1.1. Login Page

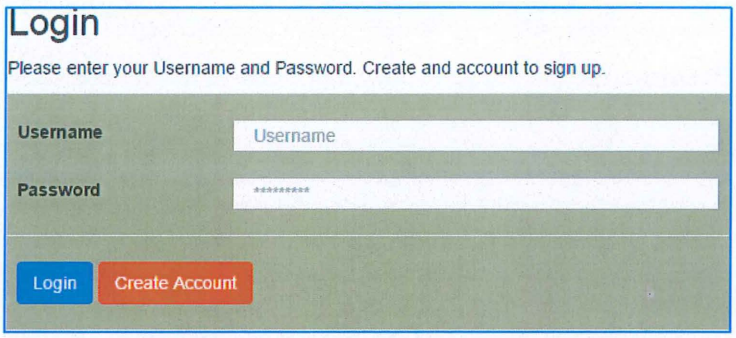


Figure 5.1.2 The login interface

For the login page, users will need to login using username and password. The username for lecturer would be staff id and username for student would be student id. Both users need to register first to login into the system. Therefore only authorized users are allowed to access into the system.

5.1.2. Registration Page

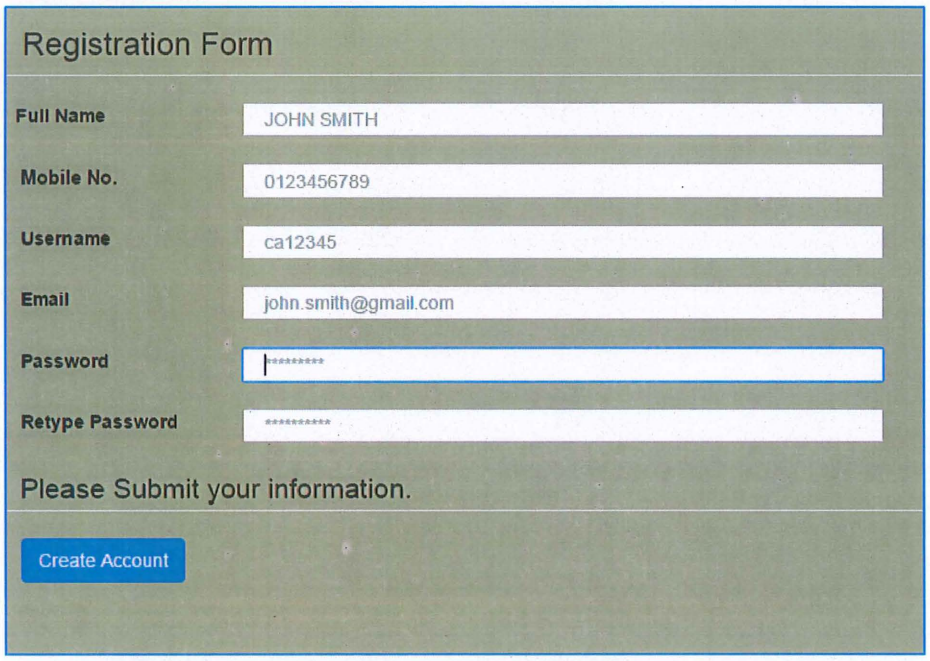


Figure 5.13 Registration form for students

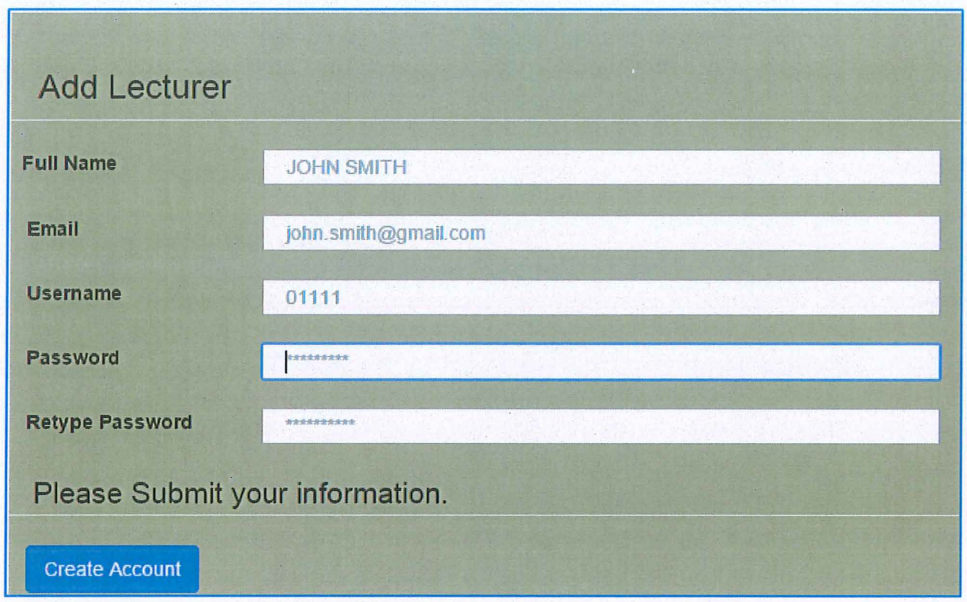


Figure 5.13 Registration form for lecturer

Figure 5.13 shows that students can register by filling in the form of the details. Students will need to register using student id for username. Figure 5.14 shows the registration form of lecturers that admin will help lecturer register by filling in the form of the details. Lecturer will use staff id for username.

5.1.3. User Profile View and Update

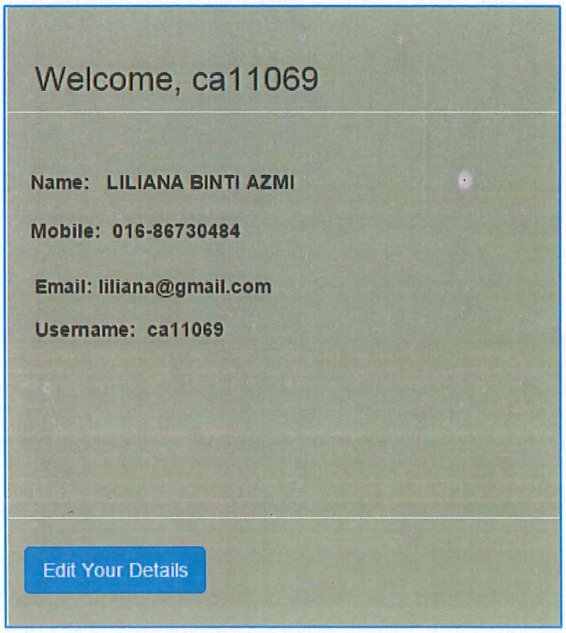


Figure 5.14 registered student's details in profile

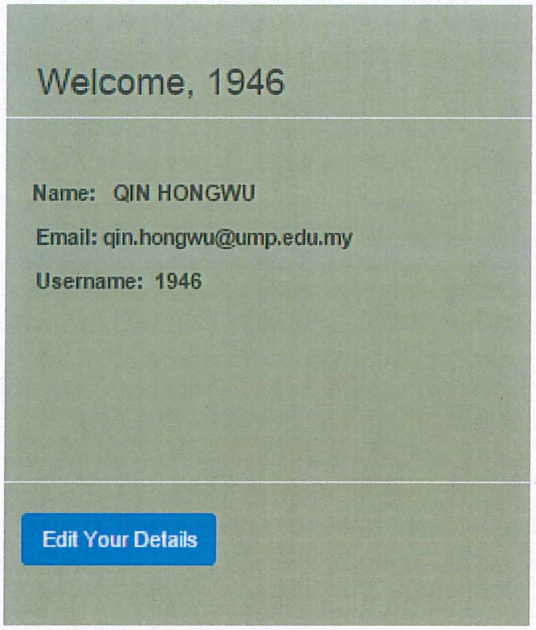


Figure 5.15 registered lecturer's details in profile

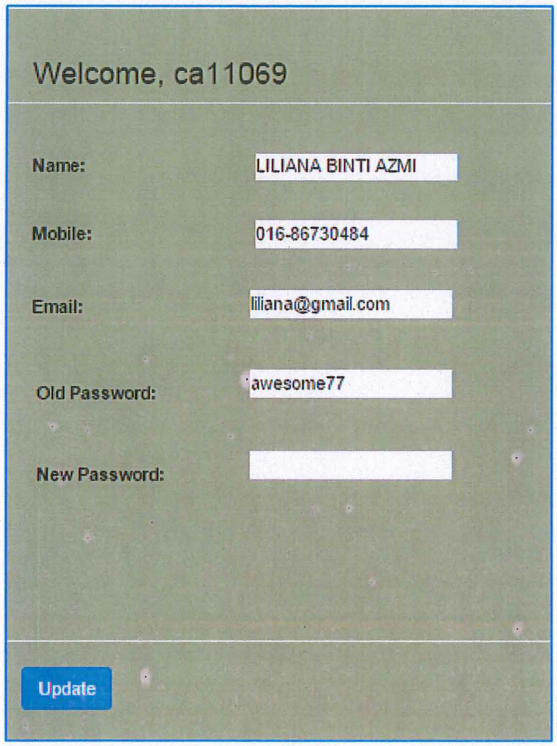


Figure 5.16 shows student's update profile form

Figure 5.17 lecturer's update profile form

5.1.4. Create Lecturer's Schedule

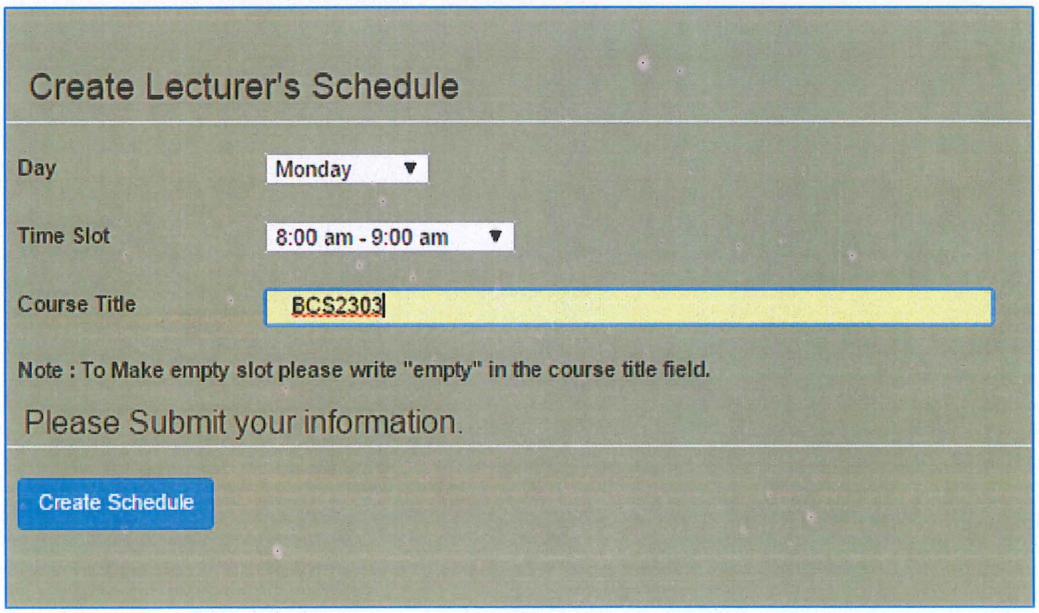


Figure 5.18 lecturer's create schedule form

Figure 5.18 above shows the interface where admin will insert the timetable for lecturer. All time slots should be filled in beginning from 8.00 a.m -9.00 a.m to 4.00 p.m to 5.00 p.m. Else , the timetable would not work correctly. Then, the schedule will be shown for lecturers like in Figure 5.19 below.

Welcome , QIN HONGWU

Schedule

Days	8:00am - 9:00am	9:00am- 10:00am	10:00am- 11:00am	11:00am- 12:00pm	12:00pm- 1:00pm	1:00pm- 2:00pm	2:00pm- 3:00pm	3:00pm- 4:00pm	4:00pm- 5:00pm	
Monday	OOP	OOP	empty	empty	empty	Break	empty	empty	empty	Edit
Tuesday	OOP	OOP	empty	empty	empty	Break	PHP	PHP	empty	Edit
Wednesday	WEB APP	WEB APP	empty	empty	empty	Break	meeting	meeting	meeting	Edit
Thursday	Exhibit	Exhibit	Exhibit	Exhibit	Break	PHP	PHP	empty	empty	Edit
Friday	empty	empty	empty	empty	empty	empty	empty	empty	PA	Edit

Figure 5.19 Lecturer's schedule on lecturer's view

Schedule

Days	8:00am - 9:00am	9:00am- 10:00am	10:00am- 11:00am	11:00am- 12:00pm	12:00pm- 1:00pm	1:00pm-2:00pm	2:00pm- 3:00pm	3:00pm- 4:00pm	4:00pm- 5:00pm
Monday	on leave	on leave	on leave	on leave	on leave	on leave	on leave	on leave	on leave
Tuesday	empty	empty	empty	empty	empty	ahmed ali Agenda simulation	empty	empty	empty
Wednesday	empty	empty	empty	empty	empty	Break	meeting	meeting	meeting
Thursday	empty	empty	empty	empty	empty	empty	empty	empty	empty
Friday	KHALISAH BINTI ABDUL BASID Agenda Quiz	empty	empty	empty	empty	empty	empty	empty	empty

Figure 5.20 Lecturer's schedule on student's view

5.1.5. Update lecturer's schedule

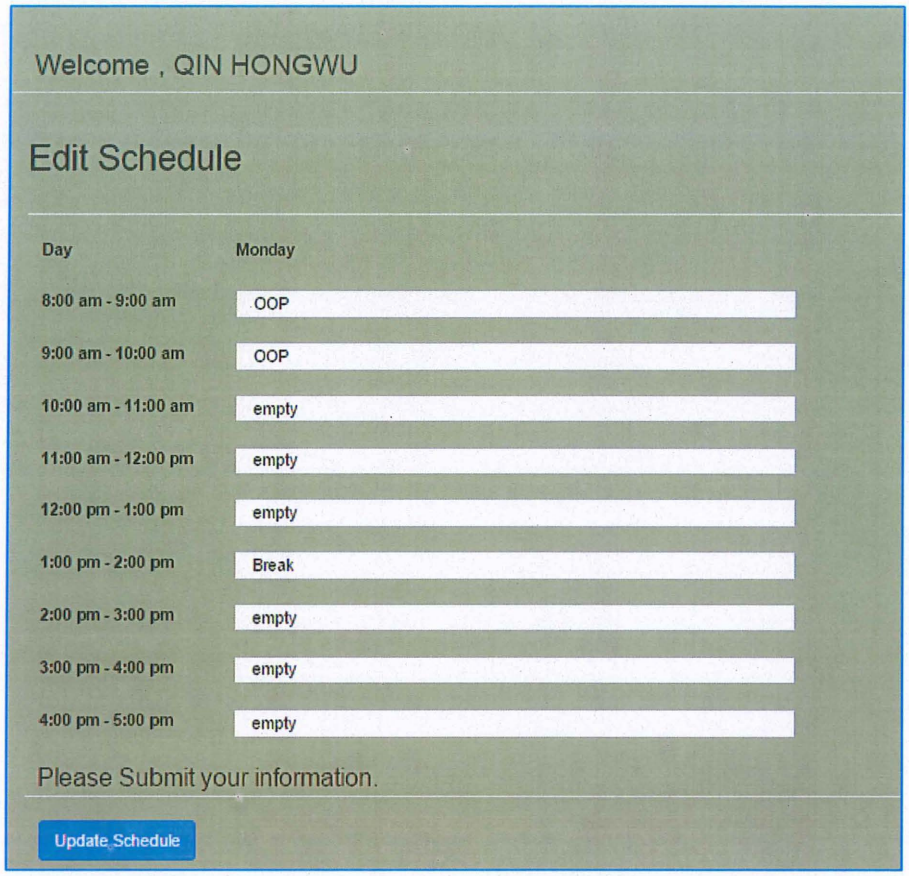


Figure 5.21 Update lecturer's schedule

Figure 5.21 shows the interface where lecturers can edit schedule according to time slots. Lecturers need to fill time slot with the word 'empty' for free or available time.

5.1.6. Appointment Page

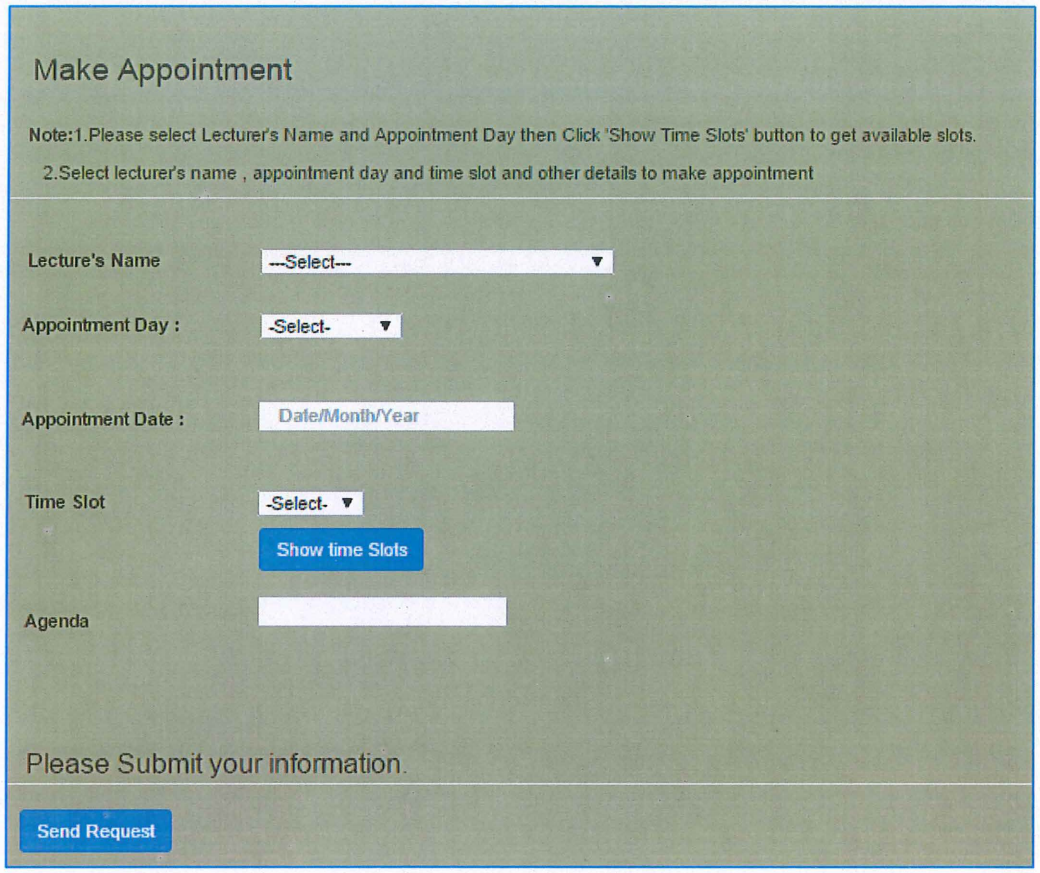


Figure 5.22 Make appointment page

Figure 5.22 shows that the interface that students will see to make an appointment. When student click on 'show time slots' button, it will only display the time slots that are free. Students will send request to make appointment with lecturer and wait for the lecturer's approval.

5.1.7. Student Appointment Request

Figure 5.22 Appointment request page

After appointment has been successfully submitted, lecturers will see the request when logged in into OASF. Lecturers will need to approve or reject the request. When the appointment request has been approved, the chosen time slot will be changed from empty to filled with appointment details like in figure 5.23 below. Student also will see the time slot has changed.

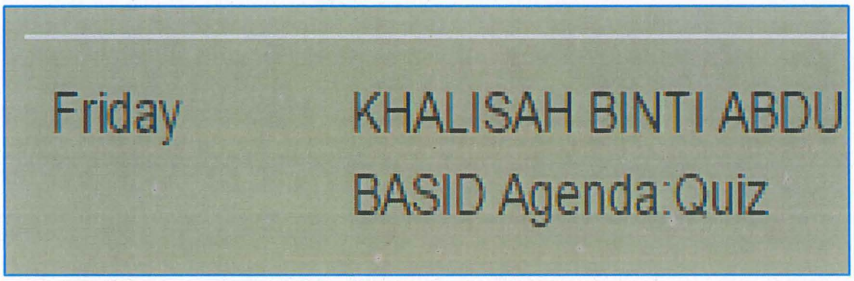


Figure 5.23 Appointment booked in time slot

Figure 5.24 Approved appointment viewed by student.

5.2. Discussions

On this section, we will discuss on the problems based in the problem statement. It shows that there are objectives that need to be accomplished. Begin from the earliest starting point of arranging stage while creating this system. Information are gained by interviewing a few students, observation and also based on user experience. The phase has successfully achieved

During the development of the system, there are constraints that makes the development of the system hard which was creating the schedule of the lecturers because different lecturer has different schedule.

5.3. Advantages and Disadvantages

This section will discuss and focus the advantages and disadvantages of OASF

5.3.1. Advantages of OASF

This system will give advantages to both lecturers and students such as:

1. This system allows lecturers have a proper information linkage to update their availability on timetable. Lecturers usually will put their timetable in front of their door which contains only classes they are teaching but does not show if lecturer is having a meeting or other activities.
2. Students can always view lecturer's timetable before making an appointment. This will save student's time instead of reaching lecturers at the faculty but lecturers are still unreachable.
3. This system is not going to cost money because students can make appointment online and does not need to make a phone call or send a text message to ask lecturer if they are available
4. No redundancy or clashing occurs because students are not allowed to make an appointment with the same lecturer on the same day and time.

5.3.2. Disadvantages of OASF

There are also disadvantages on the system that will affect all users:

1. The admin of the system will have to register for all lecturers of FSKKP.
2. Admin needs to fill in each time slot for each lecturer. This will give loads of work for admin

3. Lecturers will need to make sure their timetable is always updated every week.
4. Students will not be updated about lecturer's availability if lecturer does not update the timetable regularly. This will cause problems to make an appointment.

5.4. Assumption

There are some assumptions by Faculty and the user of the system. Such as:

1. The users of the system are from FSKKP where they are used to using computerized or online applications.
2. It assumed the information is timetable is added by admin of the system

5.5. Constraint

5.5.1. Technical Constraint

During the development of the system, there are some issues that occurred. First is the overlapping of timetable occurs such as when students made appointment, other lecturer's timetable also affected and cause confusion.

5.6. Future Development OASF

Based on the constraint and weakness identified, there are suggestion to enhance this system functionality.

1. Lecturer will receive email notification whenever there are new appointments made.
2. Lecturer will need to register themselves and create the timetable themselves to reduce lecturer's workload.
3. Students can see appointment status whether it is pending or rejected.

CHAPTER 6

CONCLUSION

6.0 Conclusion

Based on the system developed, we can conclude that OASF was developed to help students make appointment online .This will not cost money for students to see lecturers of FSKKP. Students also can view lecturer's availability to make appointment and will be notified once appointment is approved.

The Online Appointment System for FSKKP Lecturers and Students (OASF) is an online system that allows user to access to it everywhere with internet connection. In hostels and faculty, there are internet connections. Therefore, students will not have problems to access it anywhere around the campus.

REFERENCES

1) Mohd Helmy Abd Wahab, N. H. (2008). A Web-Based Appointment System through GSM Network. 133-137.

2) Noraziah Ahmad, R. M. (2010). E-Appointment Scheduling (EAS). *World Academy of Science*, 695-699.

3) *Sci Qual International Pty Ltd (SQI)*. (n.d.). Retrieved May 7, 2014, from Sci Qual International: http://www.sciqual.com.au/what-integrated-management-system-ims

4) Vilnius (2008). PatientAppointmentReservation System Retrieved from http://www.biti.lt/get.php?f.93

5) Helmy Abd Wahab, O. L. (2011, June). WAS-GN: Web-based Appointment System with GSM Network. *International Journal of Advancements in Computing Technology,* III(5).

6) King, R. (1985). A Database Design Methodology and Tools for Information Systems. *ACM transaction on Office Information Systems.* III (1), 2

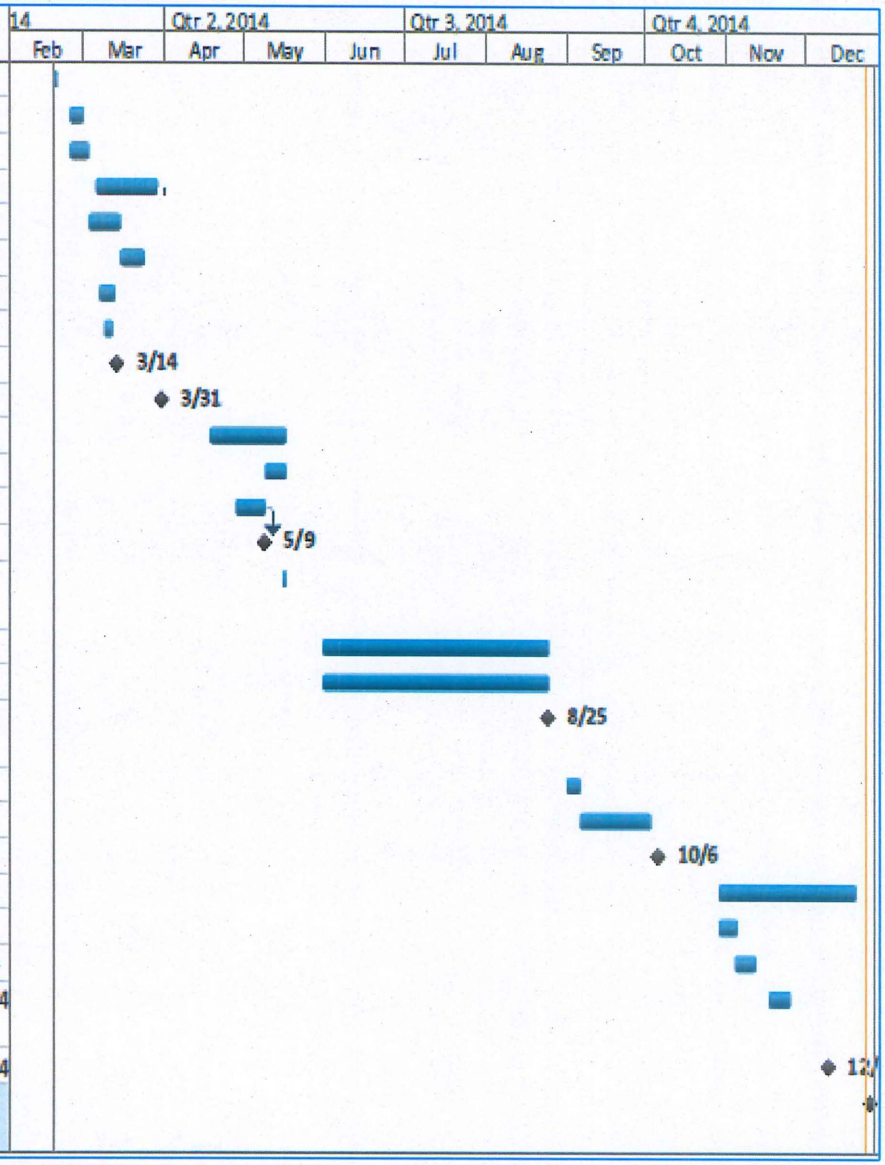
7) Mohd Helmy Abd Wahab, Gopalakrishna, S. and Johari, A. (2008) "Speed Trap Image Transfer through GSM Network". *In Proceedings of the World Congress on Engineering and Computer Science 2008 WCECS 2008* , 231-235

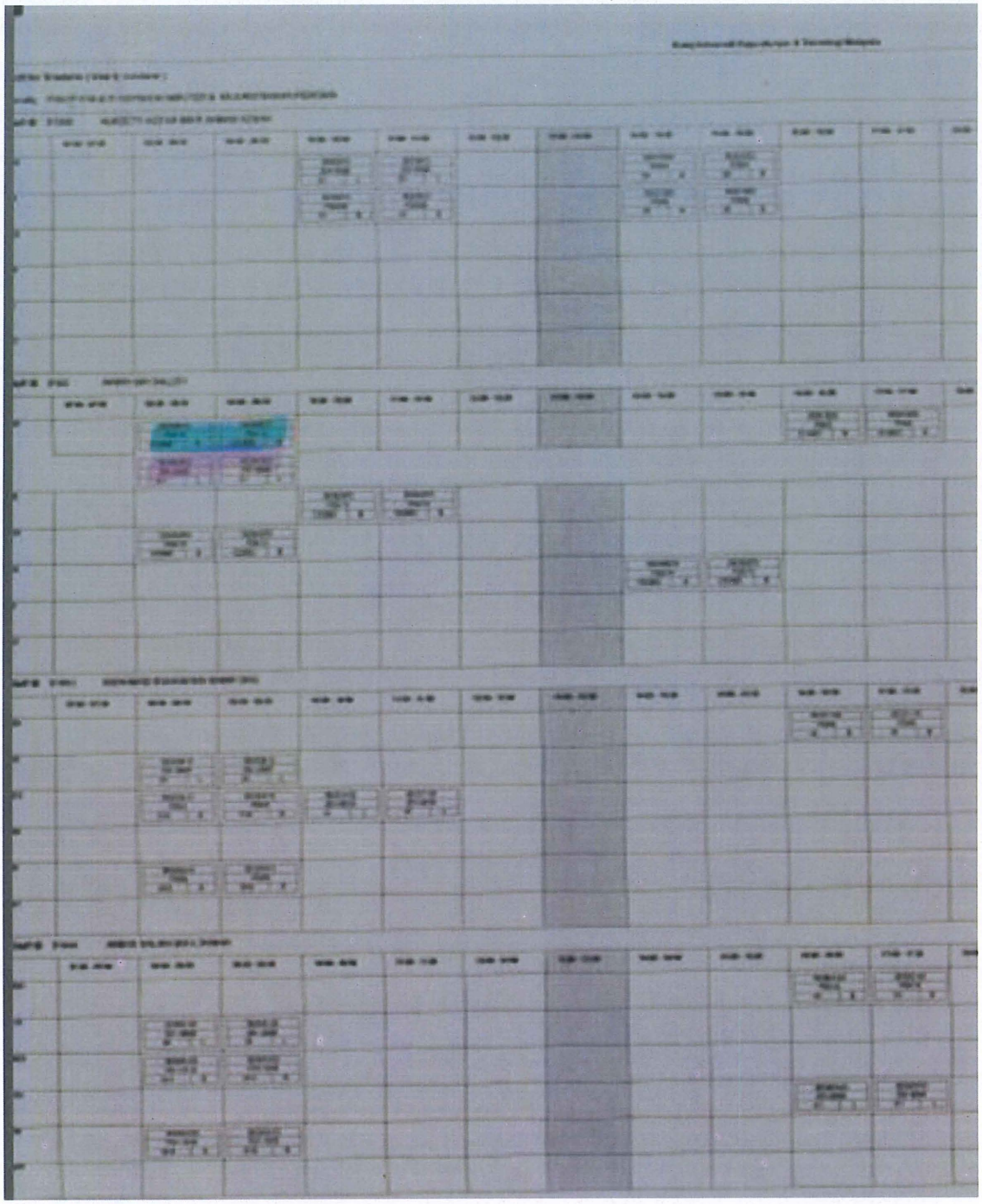
8) Irny Suzila Ishak, R. A.(2005).Designing A Strategic Information Systems Planning. *Issues in Information Systems*, VI(I) , 325-331

9) *System Requirements*.Wikipedia. 7 March 2014 Retrieved from http://en.wikipedia.org/wiki/System_requirements

APPENDIX A
PROJECT GANTT CHART

ID	Task Name	Duration	Start	Finish
1	Find title	1 day	Tue 2/18/14	Tue 2/18/14
2	Make research and draft	5 days	Mon 2/24/14	Fri 2/28/14
3	Confirm Title	6 days	Mon 2/24/14	Sun 3/2/14
4	**REQUIREMENT/ANALYSIS**	18 days	Thu 3/6/14	Mon 3/31/14
5	Gets information	11 days	Mon 3/3/14	Fri 3/14/14
6	Find existing systems	7 days	Sat 3/15/14	Sun 3/23/14
7	Analyze the system	5 days	Fri 3/7/14	Wed 3/12/14
8	Requirement research	3 days	Sun 3/9/14	Tue 3/11/14
9	Chapter 1 Submission	0 days	Fri 3/14/14	Fri 3/14/14
10	Chapter 2 Submission	0 days	Mon 3/31/14	Mon 3/31/14
11	**DESIGN**	21 days	Fri 4/18/14	Fri 5/16/14
12	Draw diagrams	6 days	Fri 5/9/14	Fri 5/16/14
13	Interface design	9 days	Mon 4/28/14	Thu 5/8/14
14	Chapter 3 submision	0 days	Fri 5/9/14	Fri 5/9/14
15	Complete Report Submission	1 day	Fri 5/16/14	Fri 5/16/14
16	**DEVELOPMENT**	62 days	Sat 5/31/14	Sun 8/24/14
17	Coding	62 days	Sat 5/31/14	Sun 8/24/14
18	System ready to use and examine	0 days	Mon 8/25/14	Mon 8/25/14
19	Test with a unit of users	5 days	Mon 9/1/14	Fri 9/5/14
20	Overcome errors	20 days	Sat 9/6/14	Thu 10/2/14
21	chapter 4 submission	0 days	Mon 10/6/14	Mon 10/6/14
22	**CUTOVER**	38 days	Wed 10/29/14	Fri 12/19/14
23	Test with users	5 days	Wed 10/29/14	Tue 11/4/14
24	Maintenance	6 days	Tue 11/4/14	Tue 11/11/14
25	chapter 5 and 6 submission	6 days	Mon 11/17/14	Mon 11/24/14
26	Draft report submission	0 days	Wed 12/10/14	Wed 12/10/14
27	Hard Cover report Summission	0 days	Fri 12/26/14	Fri 12/26/14



APPENDIX B

DATA REQUIREMENT