

ONLINE APPOINTMENT SYSTEM  
FOR STUDENT AND LECTURER

MUHAMMAD ZUL FAHMI BIN ZAHER

Bachelor of Computer Science

UNIVERSITI MALAYSIA PAHANG

## UNIVERSITI MALAYSIA PAHANG

### DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Muhammad Zul Fahmi Bin Zaher

Date of Birth : 14 April 1995

Title : E-Appointment Mobile Application

Academic Session : Session 2018/2019

I declare that this thesis is classified as:

- ☐ CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)\*
- ☐ RESTRICTED (Contains restricted information as specified by the organization where research was done)\*
- ☒ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

\_\_\_\_\_  
(Student's Signature)

\_\_\_\_\_  
(Supervisor's Signature)

\_\_\_\_\_  
New IC/Passport Number  
Date:

\_\_\_\_\_  
Name of Supervisor  
Date:

NOTE : \* If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

## THESIS DECLARATION LETTER (OPTIONAL)

Librarian,  
*Perpustakaan Universiti Malaysia Pahang*,  
Universiti Malaysia Pahang,  
Lebuhraya Tun Razak,  
26300, Gambang, Kuantan.

Dear Sir,

### CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.

Author's Name  
Thesis Title

Reasons	(i)
	(ii)
	(iii)

Thank you.

Yours faithfully,

---

(Supervisor's Signature)

Date:

Stamp:

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.



## **SUPERVISOR'S DECLARATION**

I/We\* hereby declare that I/We\* have checked this thesis/project\* and in my/our\* opinion, this thesis/project\* is adequate in terms of scope and quality for the award of the degree of \*Doctor of Philosophy/ Master of Engineering/ Master of Science in .....

---

(Supervisor's Signature)

Full Name :

Position :

Date :

---

(Co-supervisor's Signature)

Full Name :

Position :

Date :



## **STUDENT'S DECLARATION**

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

---

(Student's Signature)

Full Name : MUHAMMAD ZUL FAHMI BIN ZAHER

ID Number : CB15083

Date :

ONLINE APPOINTMENT SYSTEM  
FOR STUDENT AND LECTURER

MUHAMMAD ZUL FAHMI BIN ZAHER

Thesis submitted in fulfillment of the requirements  
for the award of the degree of  
Bachelor of Computer Science (Software Engineering) with Hons

Faculty of Computer System & Software Engineering  
UNIVERSITI MALAYSIA PAHANG

DECEMBER, 2018

## **ACKNOWLEDGEMENTS**

I am grateful and would like to express my special thanks of gratitude to my supervisor for this project, Dr Adzhar Bin Kamaluddin who had always patiently guide me whenever I ran into trouble spot or had tones of questions regarding my project in which helped me in finishing this project completely and successfully.

I must express my very profound gratitude to my parents and family for providing me unfailing supports and encouragement for me throughout my study years and throughout the process of developing this project as well as writing this thesis.

Next, this appreciation also goes to my friends that always give support, opinion and advices for me to complete this report.

Finally, I would like to express my gratitude to Universiti Malaysia Pahang for giving me such great opportunity to handle own project.

## **ABSTRAK**

Tajuk Projek Sarjana Muda (PSM) ialah “Online Appointment System for Student and Lecturer” yang dibangunkan untuk Fakulti Sistem Komputer dan Kejuruteraan Perisian (FSKKP). Objektif utama sistem ini adalah untuk mengatasi masalah yang dihadapi oleh sistem sedia ada. Medium perantaraan untuk sistem ini adalah talian Internet bagi membolehkan pelajar-pelajar melayari sistem ini di mana sahaja mereka berada selagi terdapatnya talian Internet. Sistem ini dibangunkan berlandaskan web dan dihasilkan menggunakan Bahasa pengaturcaraan web seperti PHP, HTML dan MYSQL sebagai pangkalan data untuk sistem ini beroperasi. Sasaran pengguna sistem ini terdiri daripada pelajar dan pensyarah. Secara keseluruhannya, sistem ini menerangkan latarbelakang projek, metodologi yang digunakan, bagaimana analisis dilaksanakan, rekabentuk pangkalan data dan antaramuka, pelaksanaan dan seterusnya melaksanakan pengujian kepada sistem tersebut. Sistem yang berasaskan komputer lebih sistematik dan teratur serta mengelakkan kehilangan data. Selain dari itu, fungsi-fungsi tambahan yang ada pada sistem seperti penggunaan katalaluan sebagai langkah keselamatan.



## **ABSTRACT**

The title of Projek Sarjana Muda (PSM) is “Online Appointment System for Student and Lecturer” that will be developed to Faculty of Computer Science and Software Engineering (FSKKP). The main objective of the development of this system is to overcome the problems exist in the current manual system. The medium for this system is Internet so students can access the system wherever they are as long as there is Internet connection. This system will be developed as 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 MYSQL as a database for the system. The target users of this system are students and lecturers. Generally, this system explanation about background project, methodology that will be used, how the analysis executed, design of database and interface, implementation and testing of system. The online system become systematic and also regular missing data. Besides, extra features in the system such as security protection by using password is included.

## **TABLE OF CONTENT**

**DECLARATION**

**TITLE PAGE**

**ACKNOWLEDGEMENTS** **ii**

**ABSTRAK** **iii**

**ABSTRACT** **iv**

**TABLE OF CONTENT** **v**

**LIST OF TABLES** **viii**

**LIST OF FIGURES** **ix**

**LIST OF SYMBOLS** Error! Bookmark not defined.

**LIST OF ABBREVIATIONS** **x**

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

1.1 INTRODUCTION 1

1.2 PROBLEM STATEMENT 2

1.3 OBJECTIVE 2

1.4 SCOPE 2

1.5 SIGNIFICANCE 4

1.6 REPORT ORGANIZATION 4

**CHAPTER 2 LITERATURE REVIEW** **6**

2.1 INTRODUCTION TO OAS 6

2.2 TYPE OF MANAGEMENT SYSTEM 6

2.2.1 Standalone System 6

2.2.2	Web Based System	7
2.2.3	Mobile Application System	7
2.3	EXAMPLE OF ONLINE APPOINTMENT SYSTEM	8
2.2.1	Clinikage Clinic Management System	8
2.3.2	Salonlite	9
2.3.3	OpenMRS Medical Record System	10
2.4	COMPARING THE EXISTING SYSTEMS	11
2.5	CONCLUSION	12
	<b>CHAPTER 3 METHODOLOGY</b>	<b>13</b>
3.1	INTRODUCTION	13
3.2	METHODOLOGY	14
3.2.1	Discussion on Rapid Application Development	14
3.2.2	Rapid Application Development Phases	14
3.3	HARDWARE AND SOFTWARE	18
3.3.1	Hardware Requirement	18
3.3.2	Software Requirement	19
3.4	GANTT CHART	20
3.5	TESTING PLAN	20
	<b>CHAPTER 4 RESULTS AND DISCUSSION</b>	<b>22</b>
4.1	INTRODUCTION	22
4.2	IMPLEMENTATION	22

4.3	TESTING AND RESULT DISCUSSION	28
	<b>CHAPTER 5 CONCLUSION</b>	<b>31</b>
5.1	INTRODUCTION	31
5.2	RESEARCH CONSTRAINTS	31
5.3	FUTURE WORK	32
	<b>REFERENCES</b>	<b>33</b>
	<b>APPENDIX A: GANTT CHART</b>	<b>34</b>
	<b>APPENDIX B: SRS</b>	<b>35</b>
	<b>APPENDIX C: SDD</b>	<b>55</b>
	<b>APPENDIX D: UAT</b>	<b>73</b>
	<b>APPENDIX E: USER MANUAL</b>	<b>86</b>

## **LIST OF TABLES**

<b>Table No</b>	<b>Title</b>	<b>Page</b>
2.1	Comparison between Existing Systems	11
2.2	Advantage of Existing Systems	11
2.3	Disadvantages of Existing Systems	12
3.1	Hardware Requirement for the Development	19
3.2	Software Requirement for the Development	20
3.3	Use Case Login	21

## LIST OF FIGURES

Figure No	Title	Page
2.1	Clinikage Management System	9
2.2	Salonlite Multiple Link Appointment System	9
2.3	OpenMRS Medical Record System	10
3.1	Phases in RAD Model	15
3.2	Use Case Diagram for Online Appointment System	16
3.3	Context Diagram for OAS	17
4.1	Xampp Control Panel	23
4.2	Address Name When Running on Server	24
4.3	PhpMyAdmin Database Environment	24
4.4	Example of Database Query for OAS	25
4.5	HTML Code for OAS	26
4.6	CSS Code for OAS	27
4.7	Part of PHP Code in OAS	28
4.8	Student Login Page for OAS	29
4.9	Book Appointment Page	29

## **LIST OF ABBREVIATIONS**

CASE	Computer Aided Software Engineering
CSS	Cascading Style Sheets
OAS	Online Appointment System
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor
UAT	User Acceptance Test
XAMPP	Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P)

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 INTRODUCTION**

Online Appointment System for Student and Lecturer is a proposed system for the Projek Sarjana Muda (PSM) that allow students to book appointment. This system required internet connection as the medium to access. Students can access to the system anywhere as long they have internet connection.

This system is developed as web based system that use PHP as programming language and side scripting such as HTML and CSS to build interface and also use MySQL as the database.

Lecturer need to update their schedule to this system. It will helps students to check the availability of lecturer before making an appointment booking. It also will help to reduce time for both users either students or lecturers.

Therefore, the system made processes become a lot easier, communication between students and lecturers become smooth and managed. The time consuming will be reduced for students and lecturer

#### **1.2 PROBLEM STATEMENT**

Usually the process of booking appointment is done manually. The development of this system is to computerize the manual system. This is some problem occur for the manual system.

- i) Student hard to get in touch with lecturers.



Usually student hard to meet the lecturer at their office. Most of the time, some lecturer does not leave a notice if they have emergency leave. Students also do not know lecturer's timetable.

ii) Difficult to manage report of appointment

Lecturers will write the appointment on a paper or just keep it in notebook. Sometimes this approach will make lecturer forget about the appointment or lost the notebook. With this new proposed system, record of appointment time will be more systematic and regular that means the lecturers did not worry about losing their notebook. This system will remind the lecturers and students about their appointment.

iii) Apply appointment with lecturers manually

Students need to meet lecturer and request to get permission in order to have an appointment with him/her. If she/he not available in their room, student must leave note to the lecturer or come again to meet the lecturer. Therefore students need to spend lots of time for waiting time to meet lecturer.

### **1.3 OBJECTIVES**

The objectives of this system are:

- i. To build computerized lecturer's appointment system that can replace manual system.
- ii. To develop an online application system with provides user level and security level.
- iii. To test the prototype using formal testing method.

### **1.4 SCOPE**

The scope of this project had been agreed and discussed. The scope for this proposed project are:

Users:

There are three users in this system which is admin, students and lecturers.

Function Available:

### Registration Management

Students need to register in order to proceed for the process of applying or booking appointment.

### Login

Students and lecturers need to enter the correct username and password to login on the system.

### Appointment Management

Students can make appointment by referring the lecturer's availability. Students or lecturers also can cancel the appointment if they want to cancel.

### Profile Management

Students and lecturers can update their profile such as email, contact no and also room.

### Add Lecturer

Admin will add lecturer to the system if there any new lecturer in the faculty.

### Log Activity Report

Admin can view all the logs of students and lecturers to the system.

### Development:

#### Software

The development of graphical user interface (GUI) is using HTML and CSS while the system was developed using PHP programming language.

#### Hardware

Computer and printer are the hardware that been used in development of this system.

Data Storage:

Data storage function for this system is using MySQL phpMyAdmin database.

## **1.5 SIGNIFICANCE**

Online Appointment System for Student and Lecturer is a web-based system that will help students and lecturers to manage appointment. The user's information is stored in the database. By using this system, time consuming for making appointment is reduced.

The most crucial part in this system is database because it requires complex data handling. In order to prevent duplicate data, the design of the database must be more details and it will make the database easy to be maintained.

## **1.6 REPORT ORGANIZATION**

There are five chapters include in this project. Chapter 1 is about the introduction of the project. Chapter 1 consists the introduction, problem statement, objective of the project, significance, scope of project and also report organization. The development for the solution system is identified according to the problem statement stated.

In chapter 2, literature review is discussed for the project. There are some sub topic in chapter 2 which is the techniques or method that suit to the project, references for the project and also the introduction about project in general.

The methodology used in this system is discussed in the chapter 3. This chapter explain and state the methodology that has been choose for this project. There are others items discussed in this chapter including hardware and software, introduction about methodology used and Gantt chart.

The implementation, testing and result discussion is discussed in chapter 4. The implementation to the system is referring the requirements. Test cases also been discussed in this chapter to validate and verify the system free from any bug and error. The test results then is stated here.

The conclusion of the project is stated in chapter 5. The introduction in this chapter concludes the project that have been done, the description on the constraints in research constraints and future planned work of the system prototype.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 INTRODUCTION TO ONLINE APPOINTMENT SYSTEM**

OAS is a web-based system that provide student-lecturer appointment management features for user to manage their appointment booking, profile and schedule. Students and lecturers can logging to the system with a valid username and password.

OAS allow lecturer to view student lists, add schedule and update profile. Lecturer also can view, add, delete and update appointment details of student. Lecturer can add come comment to notify student either they have emergency leave or other important things.

#### **2.2 TYPE OF MANAGEMENT SYSTEM**

A management system is a set of policies, processes and procedures used by an organization to ensure that it fulfil the tasks required to achieve its objectives (Institution, 2015). Standalone, mobile application and web based system are the example for the type of management system.

##### **2.2.1 STANDALONE SYSTEM**

Standalone device is any mechanism or system that can perform its function without the need of another device, computer or connection. (Hope, 2017) For an instance, for an office with 20 systems, the software needs to be installed on each of these 20 systems. Standalone system is a computer software that can work offline, i.e. not necessarily require network connection to function (Wikipedia, 2016) and it refer to a device that is self-contained, one that does not require any devices to function. (Beal, 2017)

The advantage of a standalone system is that it is damage control where whenever something goes wrong with the system, only the computer with the installed system will be affected. It will not affect other computer users. Another advantage of a standalone system is simplicity. Managing one computer uses less expertise rather than managing several.

The disadvantage of a standalone system is users are restricted to a single computer. Besides, it would also take a long time to install a standalone system since it need to be done one by one at a time on each computer.

### **2.2.2 WEB BASED SYSTEM**

Web based system is known as web applications. Web based application refers to any program that is accessed over a network connection using HTTP. Web based system is often run inside a web browser ("What is a Web Based Application", 2016).S.Housley, 2010 states that web based system is an application system that is deliver to users from a web servers like the internet. Developer need to understand about tool and technologies because it is really important in order to develop web based system. The examples of tools and technologies are XML, UML, Java and database.

The advantage of a web based system is easy to install. Setting up on a web based applications would only takes a couple of minutes. They then will be provided with the URL, a username and password. Moreover, web based applications is worldwide access. People can access the applications from anywhere at any time unlikely standalone system which can only be accessed from the computer with the installed system.

The disadvantage of web based system is that it is internet reliance. In order to get access to a web system, internet connection is very crucial. There will be problems whenever the internet goes down. The internet is also not everywhere. Users can only access the system anywhere at any time whenever they have internet connection

### **2.2.3 MOBILE APPLICATION SYSTEM**

A mobile application is a computer program designed to run on mobile devices such as smartphones and tablet computers. Most such devices are sold with several apps bundled as pre-installed software, such as a web browser, email client, calendar, mapping

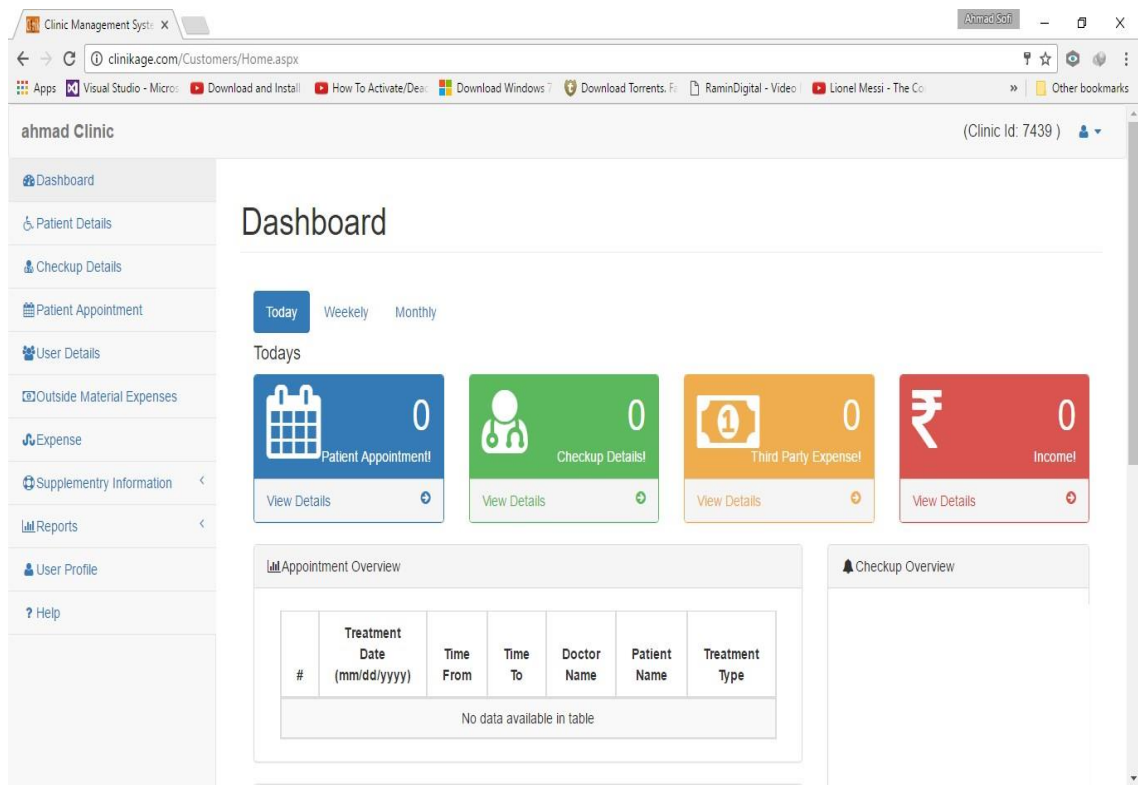
program, and an app for buying music or other media or more apps ("Mobile Application", 2015). The development of mobile application will be decided by how and where it will be distributed. For example like android or iOS. Mobile application are categorized into two types which are native and hybrid. Native application is written for a specific operating system meanwhile hybrid application is coded like a website using industry standard web technologies such as HTML, CSS and JavaScript. The code is then "wrapped" into a native container that allows the application to access specific features to be distributed and installed like a native application. Basically, mobile application development will support all development approaches either it is web, hybrid or native.

The advantage of mobile application system is that it runs on any device such as Blackberry, iPhone, Android and etc. Mobile application system is not restricted to one platform only. Another advantage of mobile application system is they are single codebase which means that it's easy to update and maintain unlike standalone and web based system which takes time to update and maintain. Limitation is one of the disadvantage of mobile application system. This is because mobile devices has limited amount of storage and size of the product screen compared to computer. Another disadvantage of mobile application system include the privacy and security because when users download an application, users possibly have to allow the publishers to track and analyse their actions. This would cause to breaking of personal information or on the other hand, user's mobile might get virus downloaded without even they noticing.

## **2.3 EXAMPLE OF APPOINTMENT MANAGEMENT SYSTEM**

### **2.3.1 Clinikage Clinic Management System**

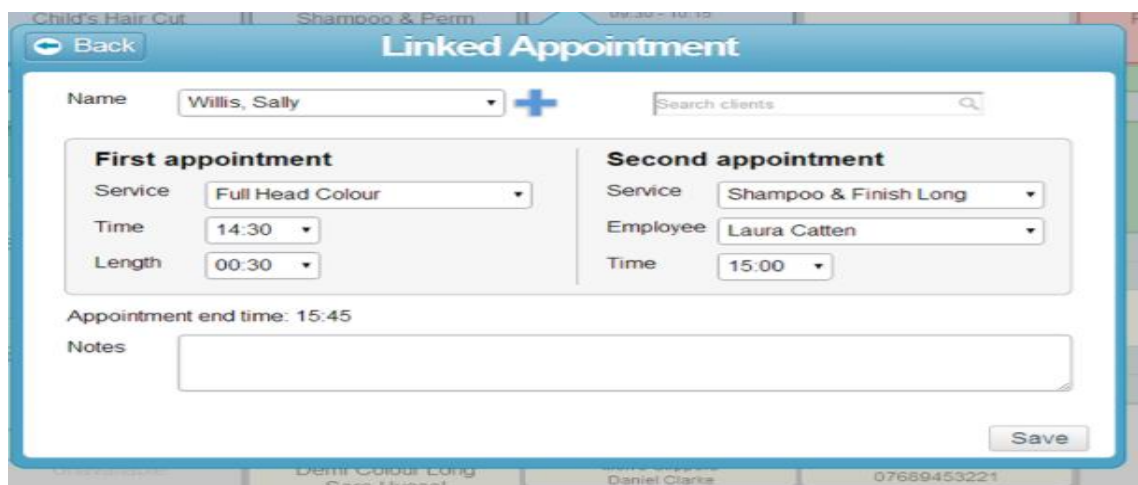
Clinikage.com offer basic office solution such as patient appointment, check-up details, treatment planning. The system does not have a free user friendly patient portal and the patient are not able to see their appointment online. Figure 2.1 shows the patient page of Clinikage.com web based system.



**Figure 2.1:** Clinikage Clinic Management System

### 2.3.2 Salonlite

Salonlite is an appointment scheduling system that are specifically designed for salon and spa services. The system also allow user to link multiple appointment together including any processing time. Figure 2.2 show the multiple link appointment page.



**Figure 2.2:** Salonlite Multiple Link Appointment Page



### 2.3.3 OpenMRS Medical Record System

OpenMRS is a clinic management system that allow user which is the doctors to manage their patient and appointment. There are many features in the system such as appointment scheduling, registration of patients and capture vitals. It is a web based system that can be access by any platform. Figure 2.3 shows the patients page of the system.

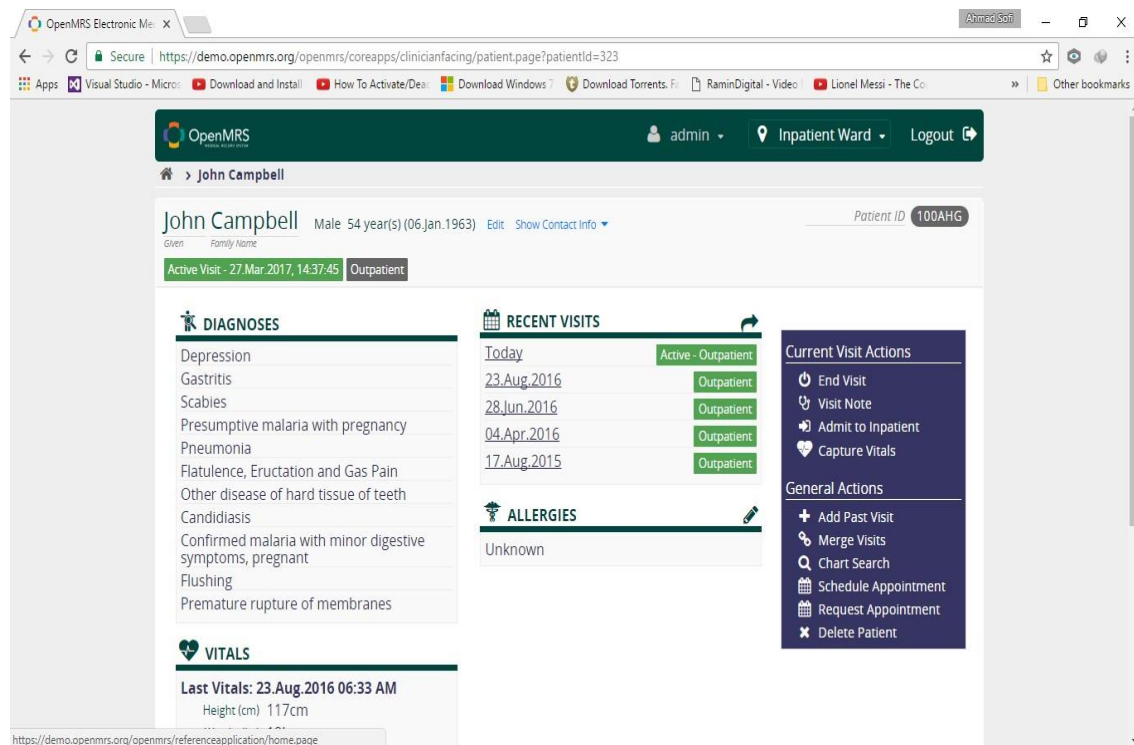


Figure 2.3: OpenMRS Medical Record System

## 2.4 Comparison of the existing system

Table 2.1 shows the comparison between the Clinikage, Salonlite and OpenMRS.

Aspect	Clinikage	Salonlite	OpenMRS
<b>Deployment</b>	Web Based	Web Based, Mobile Application	Web Based, Desktop Application
<b>Specific Function Available</b>	a. Patient Registration b. Appointment Management c. Supplementary Information d. Treatment e. Check-up Detail	a. Customer Registration b. Customer Management c. Imaging Reports d. Reports e. Salon and spa booking appointment	a. Patient Registration b. Patient Management c. Active visits d. Capture Vitals e. Data management
<b>Supported Operating Systems</b>	Web browser for Mac OS, Windows	Android, IOS	Web browser for Mac OS, Windows.

**Table 2.1:** Comparisons between Existing Systems

Table 2.2 shows the advantages of each existing systems

Aspect	Clinikage	Salonlite	OpenMRS
<b>Advantages</b>	a. Can be accessed from anywhere b. Data is secured c. No upgrade software d. Easy to install	a. Can be accessed from anywhere b. Data is secured c. Multi-role g. Easy to install	a. Can be accesses anywhere b. Data is secured c. Advanced scheduling d. Manageable e. Advanced administration f. Easy to use

**Table 2.2:** Advantages of Existing Systems

Table 2.3 shows the disadvantages of the existing systems

Aspect	Clinikage	Salonlite	OpenMRS
<b>Disadvantages</b>	a. Required internet b. Vulnerability to attack c. User have limited control and flexibility d. Downtime : system drop offline	a. Required Internet b. Vulnerability to attack c. Downtime : system drop offline	a. Required Internet b. Vulnerability to attack c. Downtime : system drop offline

**Table 2.3:** Disadvantage of Existing Systems

## 2.5 Conclusion

Appointment management system is a system that allow user to manage their appointment via online system. Usually the appointment systems have some functionality such as registration, manage appointment schedule, report and manage user information. There are many appointment system exist today either it was developed as web based system or mobile application. The proposed system for this project is developed as web-based system because the database and all information is stored in the server and it will not be a problem for limitation of storage size. Therefore, web based system is more reliable in order to build this system.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

Chapter 3 will describe about what type of methodology used to in the development process of this project. Usually, methodology is the organized, theoretically analysis of the method applied to a field of study which contain the theoretical analysis of the body methods and principles combine with a section of knowledge ("Methodology", n.d.). Framework is used to structure, plan and control the process of developing an information system. Organization or project teams commonly use the software development methodology as an approach to apply the software development methodology framework.

There are many variety of methodologies such as waterfall, prototyping, iterative and incremental development that are commonly used. Each of the methodologies have advantages and disadvantages. The use of methodology is not specified for all project because it depend on the type of project.

For this project, the methodology that will be used is Rapid Application Development. This methodology used to structure, plan and control the process of developing system. It will be used from the start of the development process until the system fully developed.

## **3.2 METHODOLOGY**

### **3.2.1 Discussion on Rapid Application Development (RAD)**

Rapid Application Development (RAD) is a software development methodology with iterative framework type. The requirements is identify by reviewing some parts of software at the beginning of the development. Within the one iteration, a rough product is created through iterative development. For the next iteration, the product is reviewed and improved until it is fully completed.

Rapid Application Development (RAD) emphasizes user involvement, prototyping, reuse, the use of automated tools, and small development team (Vliet, 2008). It often involves the use of database programming and development support tools such as screen and report generators (Sommerville, 2010). Vliet H.V., 2008, states that time frame in Rapid Application Development (RAD) is decided upon first and then the project tries to realize the requested functionality or requirements within that time frame.

Rapid Application Development (RAD) is the iterative process and it was choose because it is fast development and delivery if high system at a relatively low cost It is included in the basic principles of Rapid Application Development (RAD). The basic principles of Rapid Application Development (RAD) are ("Introduction to Software Engineering", 2015)

- i. The development of the system is fast with a high quality system at low investment cost.
- ii. The project will be break into small segment to reduce inherent project risk.
- iii. Aims to produce high quality system quickly.
- iv. Key emphasis is on fulfilling the business need, while technological or engineering excellence is of lesser importance.
- v. Framework can fit standard system analysis and design methods.
- vi. Active user involvement is imperative.

#### **3.2.1.1 Advantages on Rapid Application Development (RAD)**

There are some benefits of using RAD model for this project. One of the benefit is it provides greater flexibility for developer redesign and also this method is used to

shorten development time. The iterative development of RAD allow to do quick initial review. In rapid application development, the changes of requirements can be accommodated because the requirements do changes throughout the project.

### 3.2.2 Rapid Application Development (RAD) Life Cycle Phases

The Rapid Application Development (RAD) approach was introduced by James Martin who are the author and consultant of British Information Technology. The approach consists four phases which is:

- i. Requirements planning
- ii. Application design
- iii. Construction
- iv. Cutover

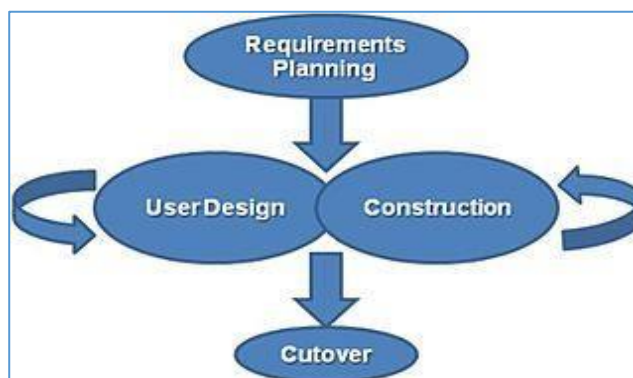


Figure 3.1: Phases in RAD Method

#### 3.2.2.1 Requirement Planning

System planning and system analysis phase of SDLC is the involvement of the requirement gathering process. The objective of the phases is to get the requirement right at the first time.

Use case diagram and context diagram is designed and documented during the requirement finalization.

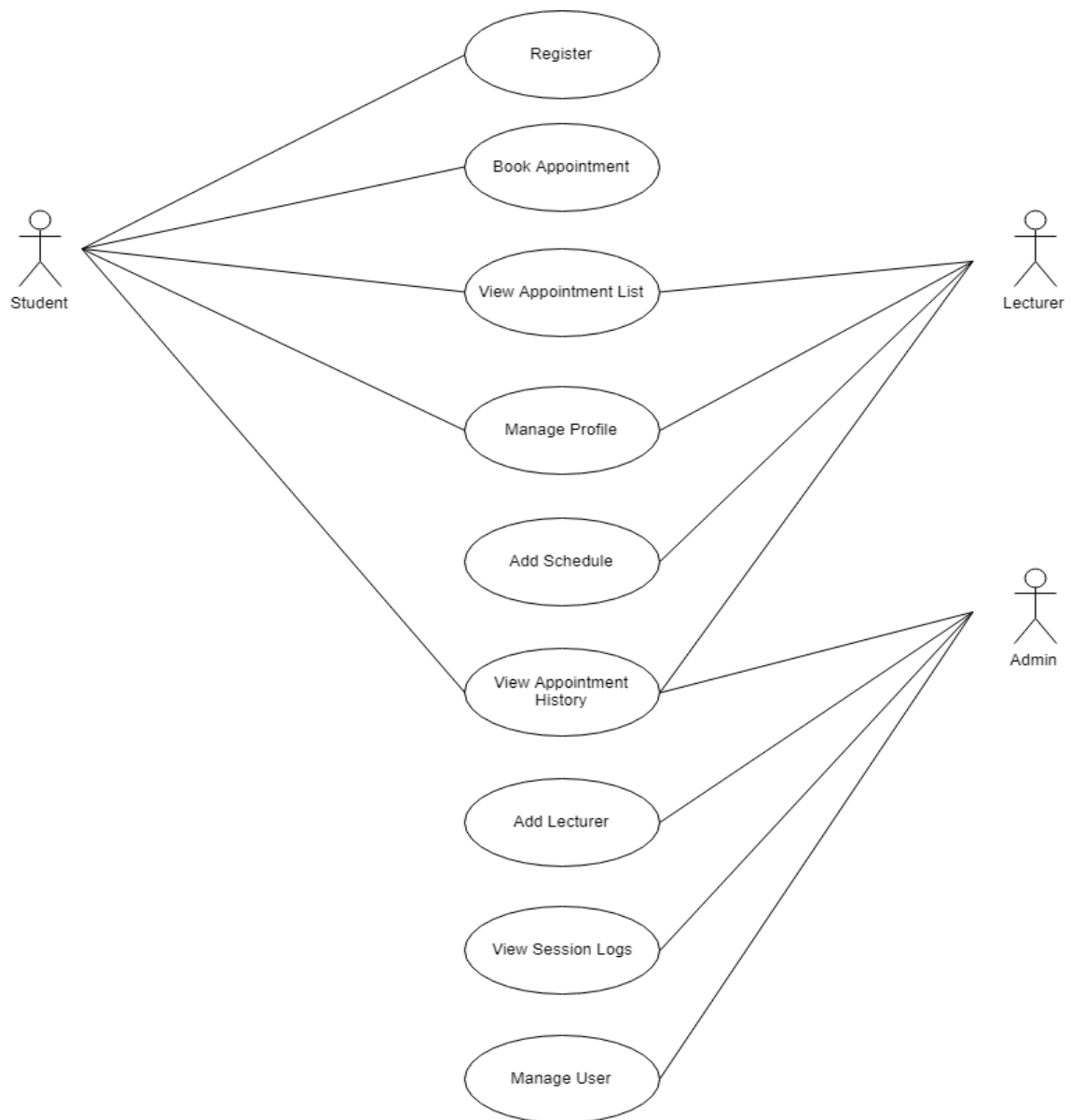


Figure 3.2: Use Case Diagram for Online Appointment System (OAS)

Figure 3.2 shows the use case diagram that had been designed with functions that will be implemented in the system. Software Requirement Specification (SRS) document will discuss about the details of the use case.

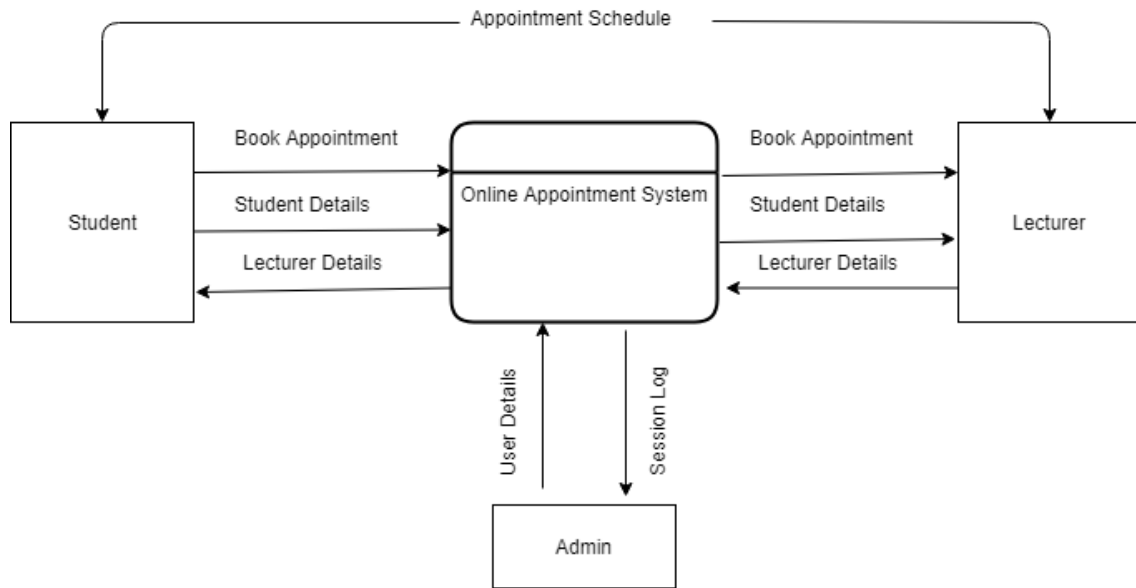


Figure 3.3: Context Diagram for Online Appointment System (OAS)

Figure 3.3 shows the system in context diagram designed. The interaction between the user and the system can be shown through the context diagram. Software Requirement Specification (SRS) also will include with Data Flow Diagram with describe the details of the development.

### 3.2.2.2 User Design

The combination of CASE tool and Joint Application Design (JAD) is the technique used during the application design phase. User design is a continuous interactive process. The developer will develop models and prototypes to show related system processes, input and outputs.

### 3.2.2.3 Construction

The developer develop a prototype to be tested by the user because in this phase, user will participate. Any changes or improvements for the system will come from the user after tested the prototype. During this phase, the system is completely designed with detailed design.



Tasks performed during the construction phase for this system are:

- Construct system: System function development is using PHP programming language. The development of user interface and database is developed using HTML and CSS language and MySQL database is use for database design.
- Test system: In order to determine either the system meets the objectives and requirements, testing should be done after the system is implemented.
- Generate test data and system documents: Testing report is generated to list out which area is affected.

The graphical user interface (GUI) is also designed during the construction phase. The details of the designs is describe in the Software Design Document (SDD) in Appendix C. The testing of the system are described in the User Acceptance Test (UAT) document which can be referred in Appendix D.

#### **3.2.2.4 Cutover**

Final testing of OAS will take place during this phase where users are trained and the system is then installed.

### **3.3 HARDWARE AND SOFTWARE REQUIREMENT**

Hardware and software will be discussed in this section because it is important during the system development for OAS.

#### **3.3.1 HARDWARE REQUIREMENT**

Table 3.1 below shows the hardware requirement specification needed in developing the system. The table describes the minimum specification of hardware needed in developing the system.

Hardware	Specification	Purpose	Quantity
----------	---------------	---------	----------

Laptop	Intel ® Core™ i5-5200U CPU @ 2.20GHz  4.00GB RAM  WINDOWS 10  64-bit Operating System, x64-based processor	Used to prepared the documentations required  Used to design and develop the system  Used as a hardware to run the operating system	1
USB Storage	Kingston Data Traveler 3.0 8GB	Used to stored document  Used as a backup storage	1
Printer	HP Desk Jet	Used to print documents	1

**Table 3.1** Hardware Requirement for the development

### 3.3.2 SOFTWARE REQUIREMENT

Table 3.2 below shows the software requirement needed in developing the system. The table describe the minimum specification of software needed.

Software	Purpose
Microsoft Windows Operating System	Platform to run the development software

Microsoft Office 2013 <ul style="list-style-type: none"> <li>▪ Microsoft Words</li> <li>▪ Microsoft PowerPoint</li> <li>▪ Microsoft Project</li> </ul>	<ul style="list-style-type: none"> <li>▪ To prepare related documents for the system</li> <li>▪ To prepare a presentation slide</li> <li>▪ To create Gantt chart for the project</li> </ul>
Notepad ++	Used to develop the system
IBM Rational Software Architect	Used to design UML diagram

**Table 3.2** Software Requirement for the development

### 3.4 GANTT CHART

The project take two session which are during PSM1 and PSM2. During PSM1 session, requirement planning and application design phase is done based on development method which is Rapid Application Development (RAD) while during PSM2 session is the development phase for the application. The project started from 12/02/2018 and ended on 12/12/2018. The Gantt chart in Appendix A showed the details of the project timeline.

### 3.5 TESTING PLAN

The purpose of testing plan is to test the system based on the use case designed for the system. The test case is designed to find any errors or problem that occur for the system correction.

	<b>Event</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Use correct username and password to login	Enter username and password	System will go to main page of the system			
2	Use incorrect username and password	Enter invalid username	System will notify user to enter match username and password			

**Table 3.3**      Use Case Login

## **CHAPTER 4**

### **RESULTS AND DISCUSSION**

#### **4.1 INTRODUCTION**

This chapter explained about the implementation of OAS. During the implementation process, all work from starting of the system construction until test generation is planned and finalized. The development of OAS starts with designing new GUI since the previous proposed design is not user friendly and hard to understand.

#### **4.2 IMPLEMENTATION**

The procedure and implementation of OAS will be described in this section. The construction of the system is recorded and described for all the processes involved. The system is set up to development environment, system functionality is constructed and the system is set up as a client-server based system.

##### **4.2.1 Development Environment**

The computer must be set up as the localhost server before starting the development of OAS. In this project, the web server chosen is XAMPP localhost.

XAMPP stands for the Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P). It is an open source package, easy to install the Apache distribution and also supports PHP because Apache distribution contains PHP. The database for this server is named as phpMyAdmin.

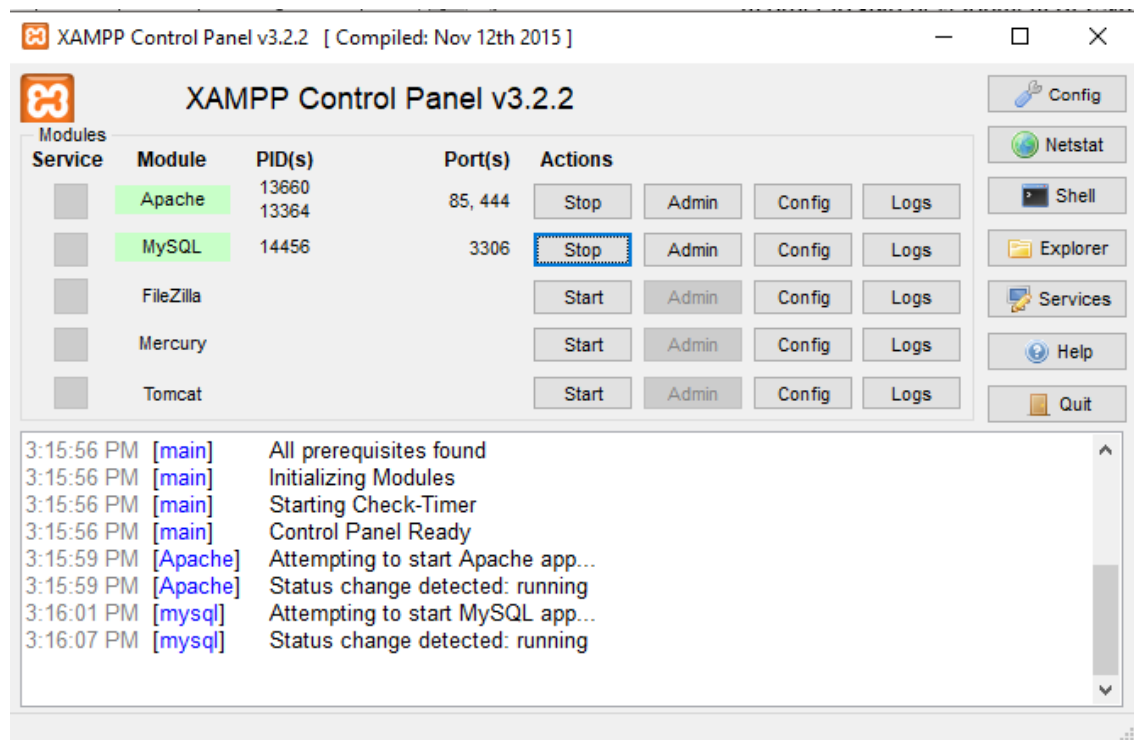


Figure 4.1: XAMPP Control Panel

In order to run localhost, the XAMPP localhost must be configured while the system is running. Figure 4.1 shows the XAMPP control panel window. To allow the system run in the server, the Apache and MySQL module must be started. MySQL is the database for the system in which it requires the Apache service to run.

All the project files is stored in XAMPP htdocs folder. Htdocs file act as a server and the project can run through the web browser because all file is stored in the htdocs folder. The project can be run in the server by typing address in the browser. The standard of address to run the project is in form of localhost/name of the file to be run.

In order to store the data from system, phpMyAdmin is choosen as the database server for this system.

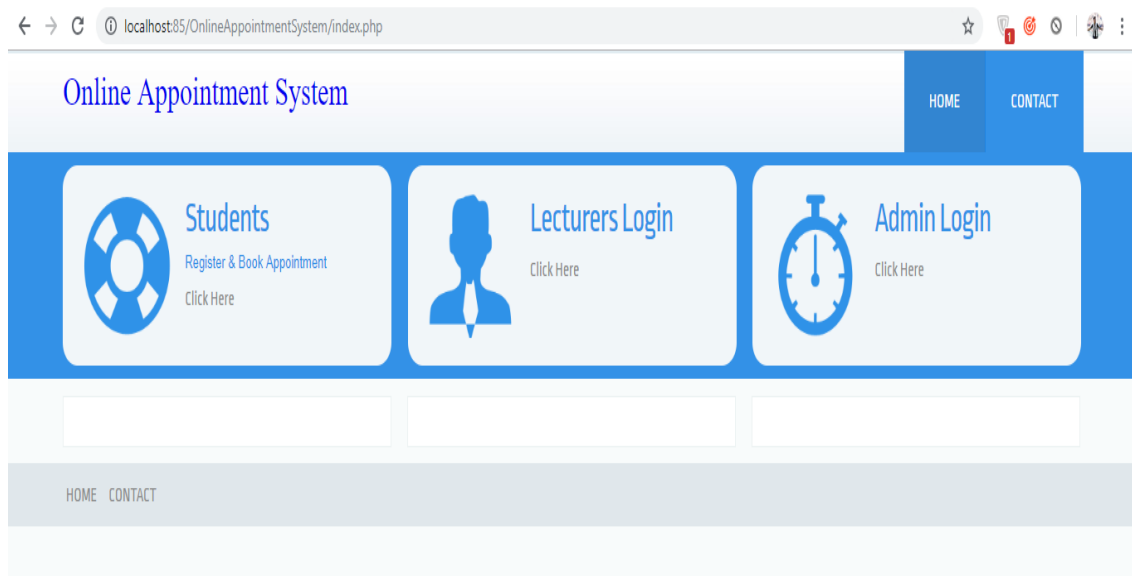


Figure 4.2: Address Name When Running On Server

Figure 4.2 shows the example of address name of the system when running the system on the browser. The address name to run this system should be in the form of localhost:85/OnlineAppointmentSystem/filename.php.

As for the database, the database server that is used are phpMyAdmin for this system in order to store the data from the system.

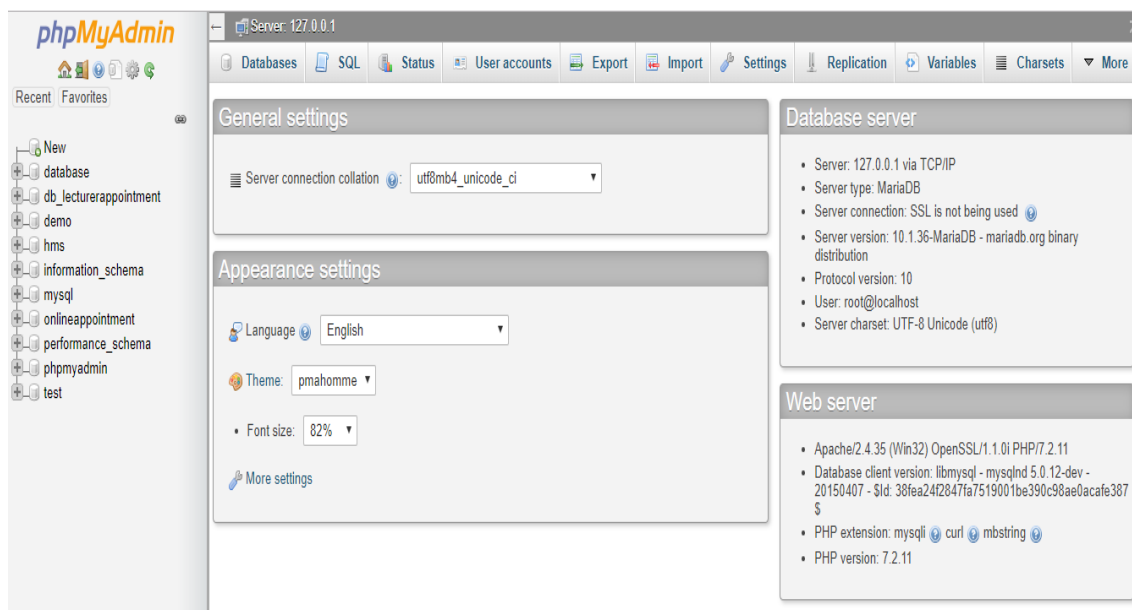


Figure 4.3: PhpMyAdmin Database Development Environment

Figure 4.3 shows the place of the database stored in the phpMyAdmin working environment. The database name and tables are designed here. Database query is used for insert, delete, update and search data in order to make connection between the system and database.

```

-----
--
-- Table structure for table `lecturers`
--
CREATE TABLE IF NOT EXISTS `lecturers` (
  `id` int(11) NOT NULL,
  `specilization` varchar(255) NOT NULL,
  `lecturerName` varchar(255) NOT NULL,
  `lecturerRoom` varchar(255) NOT NULL,
  `contactNo` bigint(11) NOT NULL,
  `lecturerEmail` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `creationDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updatetime` varchar(255) NOT NULL
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=latin1;

--
-- Dumping data for table `lecturers`
--

INSERT INTO `lecturers` (`id`, `specilization`, `lecturerName`, `lecturerRoom`, `contactNo`, `lecturerEmail`, `password`, `creationDate`, `updatetime`) VALUES
(1, 'Software', 'Roslina Mohd Sidek', 'Cabin Level 2, LE-2010', 09-5492347, 'roslinams@ump.edu.my', 'Roslinal23', '2018-11-06 06:25:37', '04-12-2018 01:27:51 PM'),
(2, 'Networking', 'Dr. Mohamed Fadli Zolkipli', 'A2, Ground Level', 09-5492467, 'fadli@ump.edu.my', 'Fadli123456', '2018-11-06 06:25:37', '04-12-2018 02:15:00 PM');
-----

```

Figure 4.4: Example of Database Query for OAS

The figure 4.4 shows one of query example used to make database connection of the system. The query used in the Figure 4.4 is to create lecturers table and also insert lecturer's data into database.



```

1  <!DOCTYPE HTML>
2  <html>
3  <head>
4      <title>Online Appointment System</title>
5      <link href="css/style.css" rel="stylesheet" type="text/css" media="all" />
6      <link href="http://fonts.googleapis.com/css?family=Ropa+Sang" rel="stylesheet" type="text/css">
7      <link rel="stylesheet" href="css/responsiveslides.css">
8      <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
9      <script src="js/responsiveslides.min.js"></script>
10
11      <script>
12          // You can also use "$(window).load(function() {"
13          $(function () {
14
15              // Slideshow 1
16              $("#slider1").responsiveSlides({
17                  maxwidth: 1600,
18                  speed: 600
19              });
20          });
21      </script>
22  </head>
23  <body>
24      <!--start-wrap-->
25
26      <!--start-header-->
27      <div class="header">
28          <div class="wrap">
29              <!--start-logo-->
30              <div class="logo">
31                  <a href="index.html" style="font-size: 30px;">Online Appointment System</a>
32              </div>
33              <!--end-logo-->
34              <!--start-top-nav-->
35              <div class="top-nav">
36                  <ul>
37                      <li class="active"><a href="index.html">Home</a></li>
38                      <li><a href="contact.php">contact</a></li>
39                  </ul>
40          ...

```

Figure 4.5: HTML Code for OAS

Figure 4.5 show the html code used for designing the main web page of the system in the web browser. The HTML and CSS is used to design the interface of the system. HTML stands for hypertext mark-up language and the CSS stands for cascading style sheets.

```

/* reset */
html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,img,
article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,section{display: block;}
ol,ul{list-style:none;margin:0px;padding:0px;}
blockquote,q{quotes:none;}
blockquote:before,blockquote:after,q:before,q:after{content:'';content:none;}
table{border-collapse:collapse;border-spacing:0;}
/* start editing from here */
a{text-decoration:none;}
.txt-rt{text-align:right;}/* text align right */
.txt-lt{text-align:left;}/* text align left */
.txt-center{text-align:center;}/* text align center */
.float-rt{float:right;}/* float right */
.float-lt{float:left;}/* float left */
.clear{clear:both;}/* clear float */
.pos-relative{position:relative;}/* Position Relative */
.pos-absolute{position:absolute;}/* Position Absolute */
.vertical-base{ vertical-align:baseline;}/* vertical align baseline */
.vertical-top{ vertical-align:top;}/* vertical align top */
.underline{ padding-bottom:5px; border-bottom: 1px solid #eee; margin:0 0 20px 0;}/* Add 5px bottom padding and a underline */
nav.vertical ul li{ display:block;}/* vertical menu */
nav.horizontal ul li{ display: inline-block;}/* horizontal menu */
img{max-width:100%;}
/*end reset*/
body{
    background:#F8FBFB;
}
.wrap{
    width:70%;
    margin:0 auto;
}
.header{
    background:url(../images/header-bg.png);
}
.logo{
    float:left;
    margin-top: 14px;
}
.top-nav{
    float:right;
}
.top-nav ul li{

```

Figure 4.6: CSS Code

Figure 4.6 shows the css code that can arrange the structure of the system that had been created using HTML code.

OAS is developed using a scripting language, PHP. PHP is a server scripting language which is widely used for making dynamic and interactive web pages (w3school.com, 2016). Since the system is embedded with HTML, PHP is the most suitable language to be used in this system.

```

<?php
include_once('include/config.php');
if(isset($_POST['submit']))
{
    $fname=$_POST['full_name'];
    $email=$_POST['email'];
    $matricid=$_POST['matricId'];
    $password=md5($_POST['password']);
    $query=mysqli_query($con,"insert into students(fullname,email,matricId,password) values('$fname','$email','$matricid','$password')");
    if($query)
    {
        echo "<script>alert('Successfully Registered. You can login now');</script>";
        //header('location:user-login.php');
    }
}
?>

```

Figure 4.7: Part of PHP Code in OAS

Figure 4.7 shows the example of PHP code used in the Online Appointment System. The code will start with “<?php” tag and end with “?>” tag.

### 4.3 System Functionality

This section will show the system function of OAS and how the system interact with the user. The GUI is the important thing to be highlight in this system and also the interaction between user and system. The GUI is user friendly and ease to use.

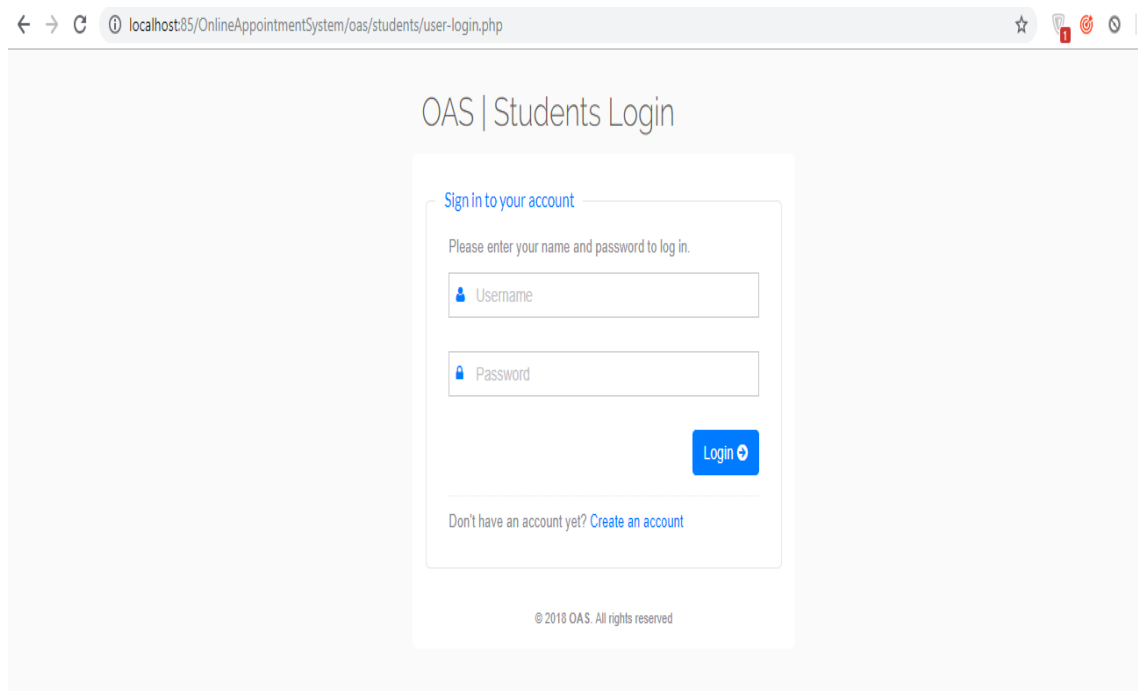


Figure 4.8: Students Login Page for OAS

Figure 4.8 above shows the login interface for OAS. Students need to fill the username and password in order to proceed to next page. Only valid username and password can get access to login to the system.

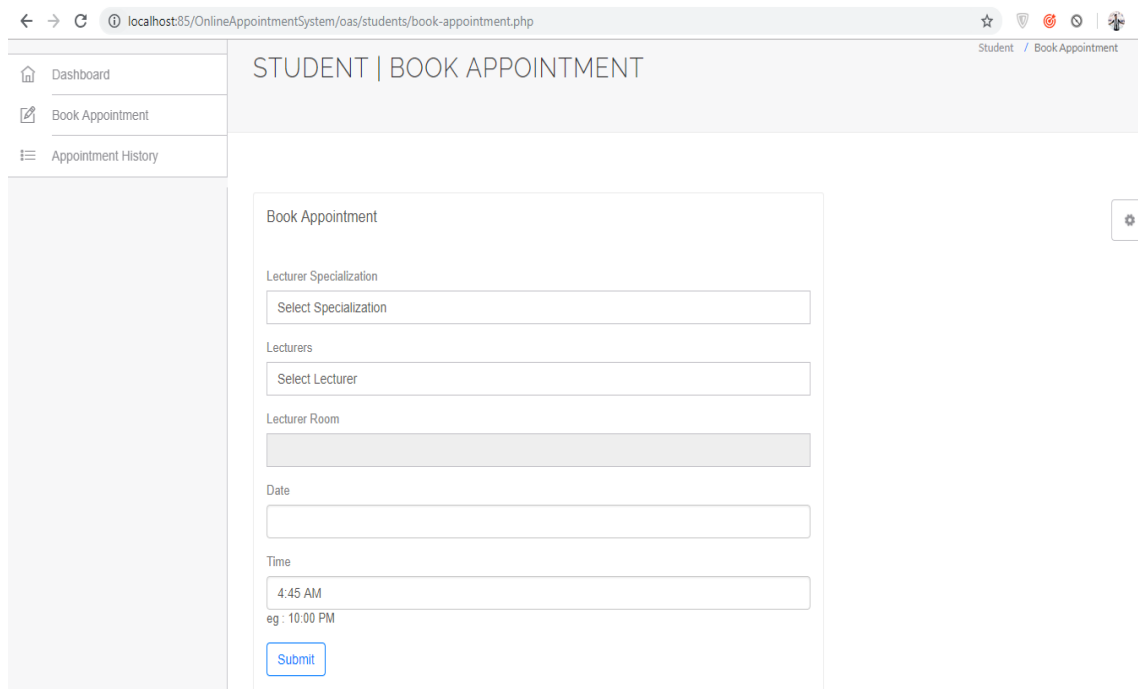


Figure 4.9: Book Appointment Page

The Figure 4.9 above show the booking appointment page. Students can select lecturer based on lecturer specialization. After completing the needed information, student can book the appointment by clicking submit button.

#### **4.4 Testing and Result Discussion**

OAS is developed using php and connected with the MySQL phpMyAdmin. One of the important phase in the development of this system is testing phase. Testing will take place after the implementation phase is completed. Based on the test cases on the specification of the software component under test, black box testing will choose as the main testing to the system.

#### **4.5 User Manual**

User manual is a guide for user how to use the system. In Appendix C, the details of user manual can be referred.

#### **4.6 User Acceptance Test**

User acceptance test is a test process to make the developed system can handle required tasks in real-world. The details about user acceptance test can be referred in Appendix D.

## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Introduction**

The Online Appointment System for Student and Lecturer (OAS) is established to improve the manual appointment system by computerizing the manual management. The users of OAS are students and lecturers. The system is developed to solve the growing issues in manual system which one of the issues is low work efficiency and performance due to mass number of files to handle manually.

#### **5.2 Research Constraint**

##### **5.2.1 Development Constraint**

Constraint that is experienced during the development are:

- 1) Time Constraint

To develop a complete and robust system, more time is needed.

- 2) Internet Connection

Internet connection is really important during the development phase. There are some internet connection problem while developing the system, therefore it become some of the constraint to complete the system.

##### **5.2.2 System Constraint**

Constraint of the system are:

- 1) Platform

This system is developed based on web browser. It not available to run as a mobile application. The system can be access through tablet, laptop, phone or computer as a web-based system through the web browser.

## 2) Language

The system use English language to make it comfort with all users either students or lecturers because English can be understand by all users.

### **5.3 Future Work**

There are some improvements that can be apply for the future update of OAS since the system is still yet perfect. The system has to be improve which allow user receive a notification as an appointment reminder a day before appointment day. Besides, this system can be enhance by converting this system to mobile application system. Therefore, the new version of OAS will be develop to ensure the system more efficient and complete to use.

## REFERENCES

Vliet, H. V. (2008). Software Engineering: Principles and Practice (3rd ed., Vol. 1).

Southern Gate, West Sussex: John Wiley & Sons

Sommerville. I. (2010). Software Engineering (9th ed.). Boston, Massachusetts:

Pearson Education

What is Iterative model- advantages, disadvantages and when to use it? (n.d.). Retrieved April 22, 2016, from <http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/>

SDLC - RAD Model. (n.d.). Retrieved April 22, 2016, from

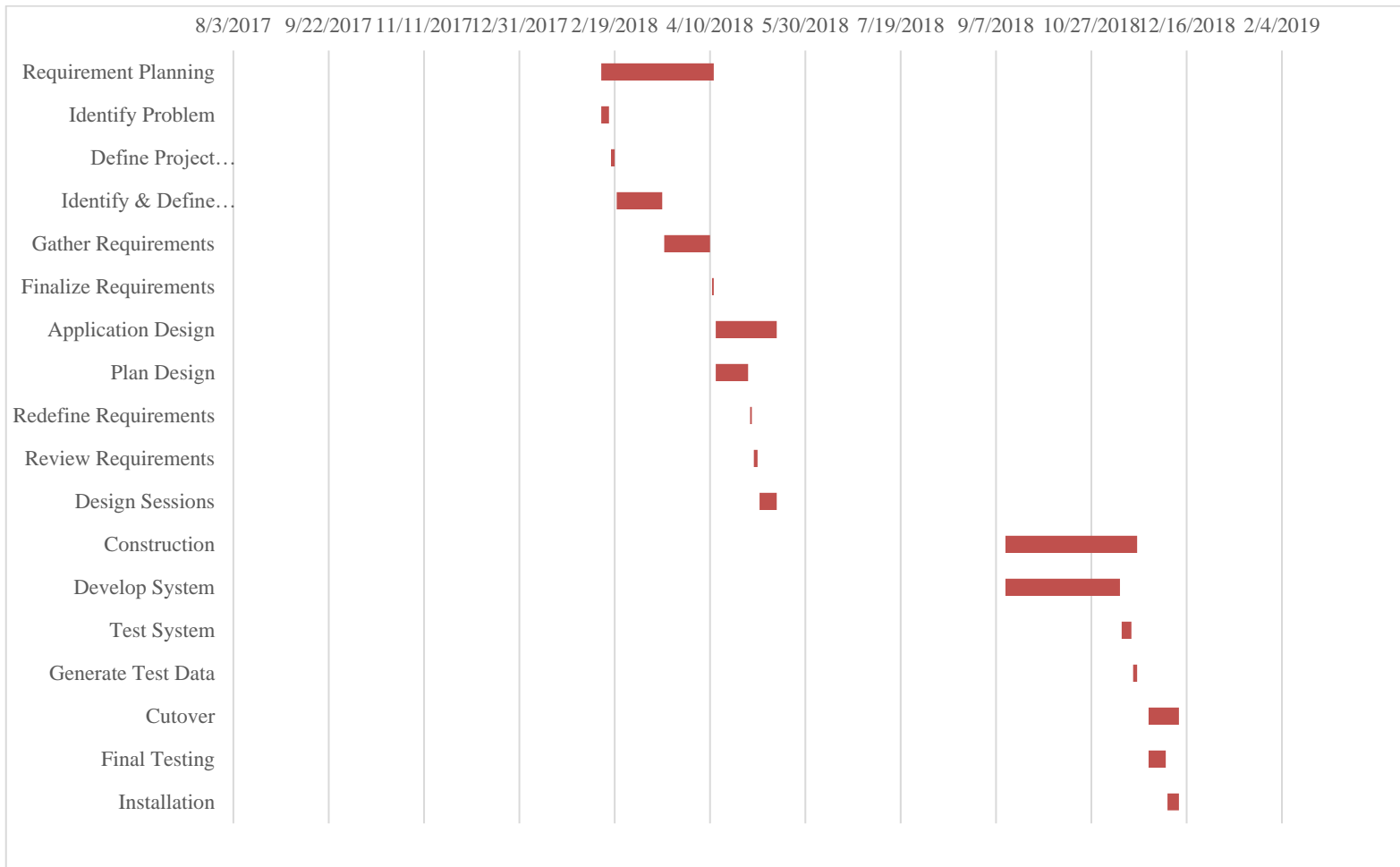
[http://www.tutorialspoint.com/sdlc/sdlc\\_rad\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm)

Joint Application Design. (n.d.). Retrieved April 24, 2016, from

[https://en.wikipedia.org/wiki/Joint\\_Application\\_Design](https://en.wikipedia.org/wiki/Joint_Application_Design)



## APPENDIX A GANTT CHART



**APPENDIX B**  
**SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

**Version**

**1**

## ONLINE APPOINTMENT SYSTEM

---

Faculty of Computer System & Software Engineering

# Software Requirement Specification (SRS)

### 5.3.1 DOCUMENT APPROVAL

	Name	Date
<b>Authenticated by:</b>  <hr/> Project Manager	MUHAMMAD ZUL FAHMI BIN ZAHER	12 December 2018
<b>Approved by:</b>  <hr/> Client		

Software : Microsoft Word 2016, IBM Rational Software

Archiving Place : D:\Zul\PSM 1

Copies Available : 2 copies

## 1.0 FUNCTIONS

This section describes all functions that is developed in Online Appointment System for Students and Lecturers. OAS consists three basic modules which are user management, appointment management and report. For user management module, it consists of several submodules which are student registration, lecturer registration and profile management

Table 1.1: Online Appointment System Functions

Function	Description	
User Management	Manage Student Registration	This is the function where student make registration to the system. Student will need to fill in the required student details and click “Submit” button which then will save student’s details into database.
	Manage Lecturer Registration	This is the function where admin will add new lecturer to the system. Admin need to fill in the required details of lecturer and click “Submit” button. The lecturer’s details will be saved in database
	Manage Profile	Student or lecturer can update their profile

Appointment Management	Student and lecturer can view, edit or delete their appointment.
Reports	This function will allow admin to view the session logs of student and lecturer to the system

## 2.0 INTERFACE REQUIREMENTS

### 2.1 USER INTERFACE

Language	Purpose
HTML	It used to defines the layout and structure of the document using variety of tags and attributes.
CSS	It used to present the style of the pages such as colour, layout and fonts.

### 2.2 HARDWARE INTERFACE

Hardware	Purpose
Laptop	Used to develop and create the system.

Thumb Drive	Used to save document and as backup.
Printer	Used to print documents related.

### 2.3 SOFTWARE INTERFACE

Software	Description
Front End Client	Web Browser (Chrome, Mozilla), Operating System (Windows)
Back End	Notepad ++, Rational Software Architect

## 3.0 SOFTWARE PRODUCT FEATURES

### 3.1 Use Case Login

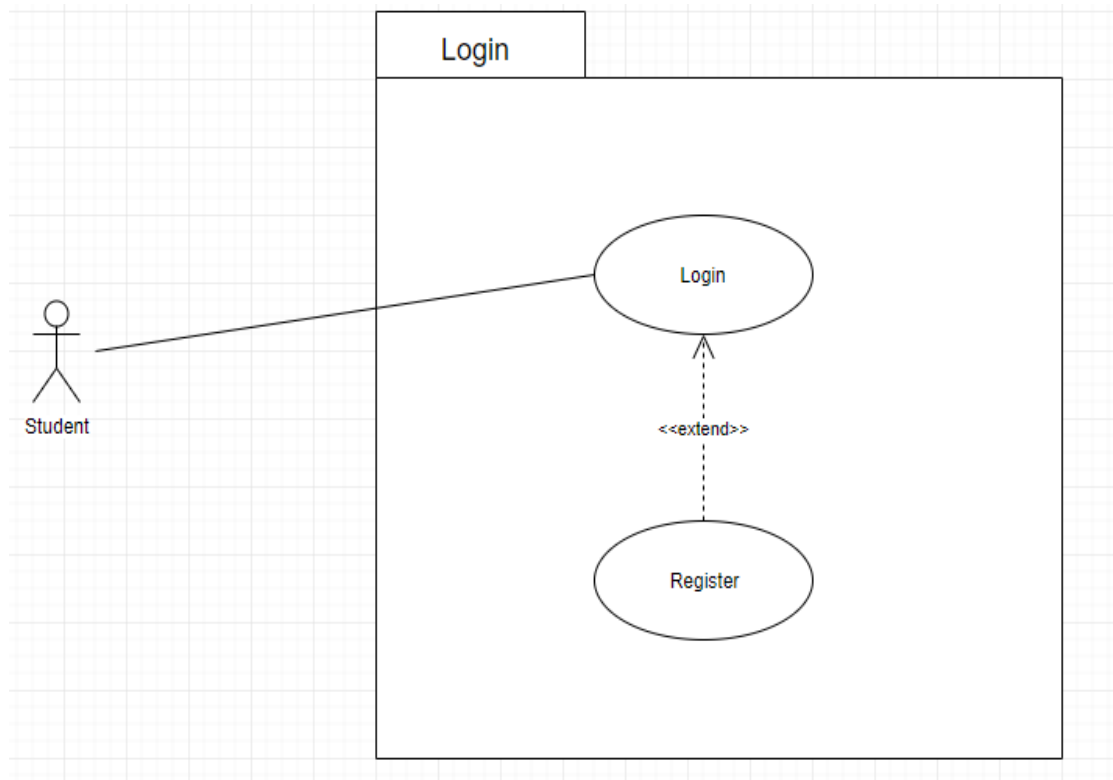


Figure 3.1: Use Case Login

**Table 3.1 Use Case Login**

<b>Use Case ID</b>	UC-M01-1
<b>Use Case Name</b>	Login into system
<b>Brief Description</b>	This use case describe about the login function for student or lecturer to use the system. Student or lecturer must login to the system before proceed for making appointment
<b>Actor</b>	Student, Lecturer
<b>Pre-Conditions</b>	Student/Lecturer already register
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Enter Username and password</li> <li>2. Hit Login button</li> </ol>



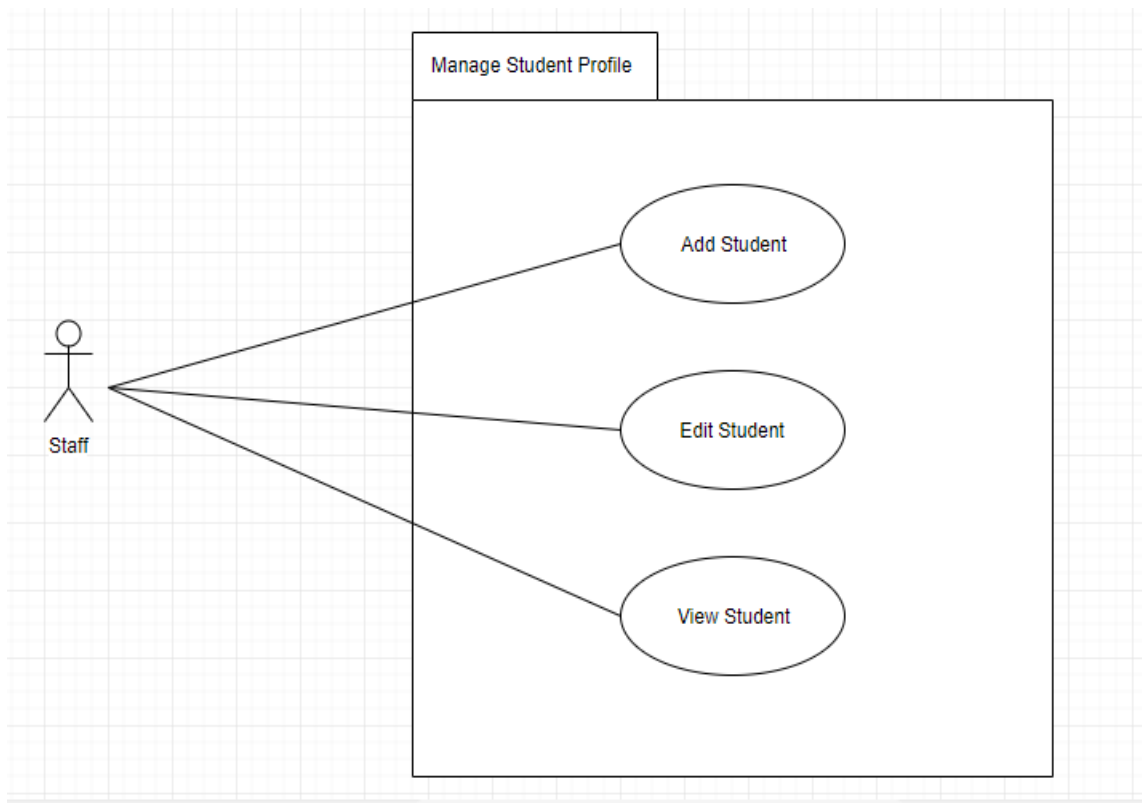
<b>Alternative Flow</b>	<p>If student/lecturer is not registered</p> <ol style="list-style-type: none"> <li>1. Provide personal information as followed: <ol style="list-style-type: none"> <li>a. Full name</li> <li>b. IC</li> <li>c. Phone number</li> <li>d. Email address</li> <li>e. Permanent address</li> <li>f. Password</li> <li>g. age</li> </ol> </li> <li>2. Commit/save</li> <li>3. Account successfully created</li> <li>4. Login</li> </ol>
<b>Exception Flow</b>	none
<b>Post-Conditions</b>	Student can proceed for managing appointment
<b>Rules</b>	Must registered in the system
<b>Constraints</b>	none
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow
<b>Interface</b>	

**Table 3.2 Use Case Register**

<b>Use Case ID</b>	UC-M01-2
<b>Use Case Name</b>	Register

<b>Brief Description</b>	This use case functions describe about how to register new student to the system.
<b>Actor</b>	Student
<b>Pre-Conditions</b>	New student for the system
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Provide personal information as followed: <ol style="list-style-type: none"> <li>a. Full name</li> <li>b. IC</li> </ol> </li> </ol>
	<ol style="list-style-type: none"> <li> <ol style="list-style-type: none"> <li>c. Phone number</li> <li>d. Email address</li> <li>e. Permanent address</li> <li>f. Password</li> <li>g. age</li> </ol> </li> <li>2. Commit/save</li> <li>3. Account successfully created</li> </ol>
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	New student can login and start manage appointment
<b>Rules</b>	Must registered in the system
<b>Constraints</b>	None
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

### **3.2 Use Case Manage Student Profile**



**Table 3.3 Use Case Add Student**

<b>Use Case ID</b>	UC-M02-1
<b>Use Case Name</b>	Add Student Profile
<b>Brief Description</b>	This use case functions to register new student to the system.
<b>Actor</b>	Staff
<b>Pre-Conditions</b>	New Student

<b>Basic Flow</b>	1. New student provide their details: <ul style="list-style-type: none"> <li>a. Full name</li> <li>b. IC</li> <li>c. Phone number</li> </ul>
	<ul style="list-style-type: none"> <li>d. Email address</li> <li>e. Permanent address</li> <li>f. Date of birthday</li> </ul> 2. Save
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	1. Student Record is saved 2. Wait for their turn
<b>Rules</b>	All information must be legit
<b>Constraints</b>	none
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

**Table 3.4 Use Case Edit Student**

<b>Use Case ID</b>	UC-M02-2
<b>Use Case Name</b>	Edit Student
<b>Brief Description</b>	This use case functions as to edit or update the profile of student if some change might be needed

<b>Actor</b>	Staff
<b>Pre-Conditions</b>	Need some changes
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Staff can search student by IC</li> <li>2. Hit button edit <ol style="list-style-type: none"> <li>a. Staff Full name</li> <li>b. IC</li> <li>c. Phone number</li> <li>d. Email address</li> <li>e. Permanent address</li> <li>f. Date of birthday</li> </ol> </li> <li>4. save</li> </ol>
	5. Account successfully edited
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	Account is updated
<b>Rules</b>	
<b>Constraints</b>	None
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

**Table 3.5 Use Case View Student**

<b>Use Case ID</b>	UC-M02-3
--------------------	----------

<b>Use Case Name</b>	View Student
<b>Brief Description</b>	This use case functions as to view patient profile
<b>Actor</b>	Staff
<b>Pre-Conditions</b>	Need correct IC
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Enter Student's IC</li> <li>2. The system will view the following details: <ol style="list-style-type: none"> <li>a. Full name</li> <li>b. IC</li> <li>c. Phone number</li> <li>d. Email address</li> <li>e. Permanent address</li> <li>f. Date of birthday</li> </ol> </li> </ol>
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	Full details of patient will be viewed
<b>Rules</b>	<ol style="list-style-type: none"> <li>1. Correct and registered IC will be viewed</li> <li>2. If Incorrect and unregistered IC is entered,</li> </ol>
	Messagebox will appear with message "the inserted IC is not in database"
<b>Constraints</b>	None
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

### 3.3 Use Case Manage Student Queue

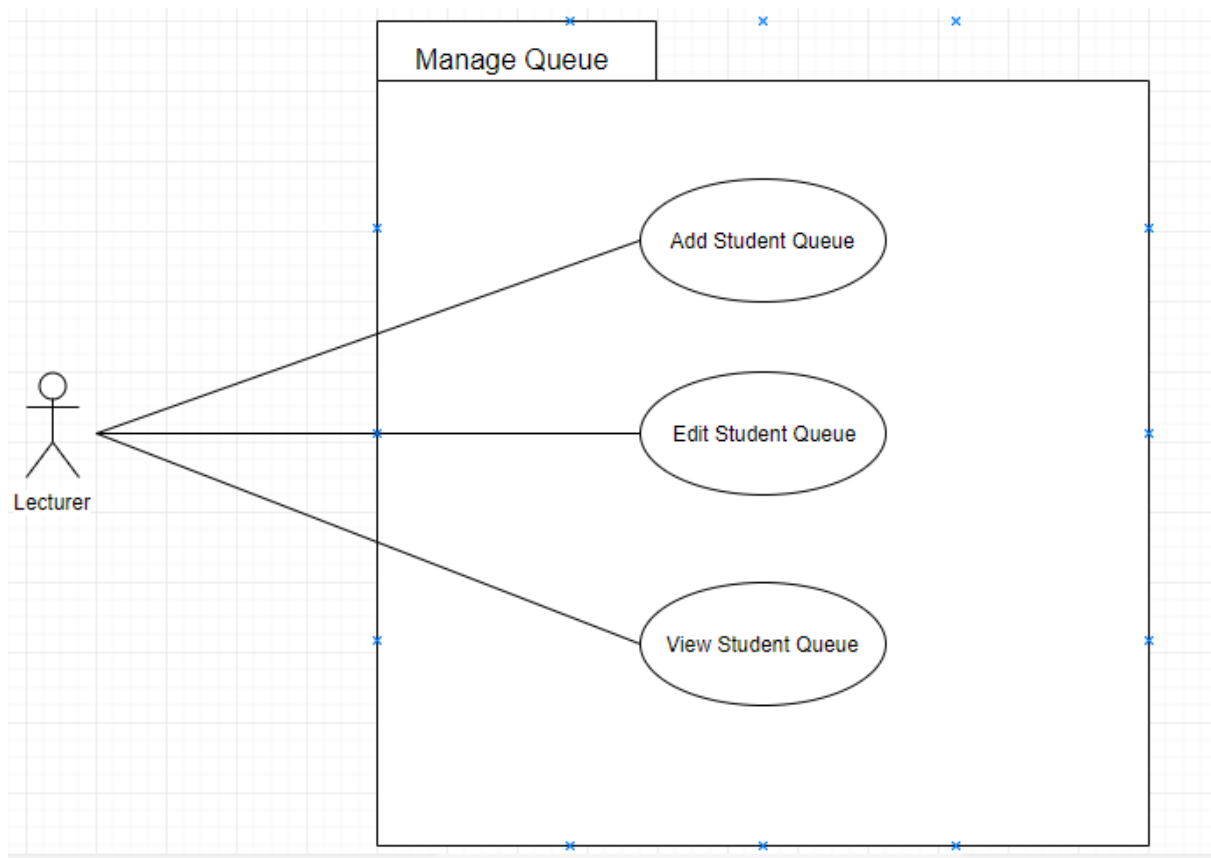




Figure 3.3 Use Case Manage Student Queue

**Table 3.6 Use Case Add Student Queue**

<b>Use Case ID</b>	UC-M03-1
<b>Use Case Name</b>	Add Student Queue
<b>Brief Description</b>	This use case functions as to register student in a queue. Student have to wait until their turn is called.
<b>Actor</b>	Lecturer
<b>Pre-Conditions</b>	Student is a regular student
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Student make booking for appointment</li> <li>2. Student stated their reason to meet the lecturer</li> <li>3. Lecturer will view the student's application</li> <li>4. Click add to queue</li> </ol>
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	Student have to wait until their turn is called.
<b>Rules</b>	All information must be legit
<b>Constraints</b>	none
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

**Table 3.7 Use Case Edit Student Queue**

<b>Use Case ID</b>	UC-M03-2
<b>Use Case Name</b>	Edit Patient Queue
<b>Brief Description</b>	This use case functions as to edit or change the sequence of queue of patient if some change might be needed
<b>Actor</b>	Staff
<b>Pre-Conditions</b>	Patient might having some problem before their turn
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Staff click edit patient queue</li> <li>2. Staff click on patient name</li> <li>3. Staff can drag their name to any part of the list</li> <li>4. Save</li> </ol>
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	Patient have to wait until their turn is called.
<b>Rules</b>	None
<b>Constraints</b>	None
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

**Table 3.8 Use Case View Student Queue**

<b>Use Case ID</b>	UC-M02-4
<b>Use Case Name</b>	View Patient Queue
<b>Brief Description</b>	This use case functions as to view patient queue

<b>Actor</b>	Staff
<b>Pre-Conditions</b>	Previous patient done their treatment
<b>Basic Flow</b>	<ol style="list-style-type: none"> <li>1. Staff click view patient queue button</li> <li>2. Patient queue list is viewed</li> </ol>
<b>Alternative Flow</b>	None
<b>Exception Flow</b>	None
<b>Post-Conditions</b>	Full list of patient queue will be viewed
<b>Rules</b>	The current patient on the treatment is done on their turn
<b>Constraints</b>	None
<b>Sequence Diagram</b>	Refer Appendix A-1.1 : Basic Flow A-1.2 : Alternative Flow A-1.3 : Exception Flow
<b>Interface</b>	

### 3.4 Use Case Manage Appointment

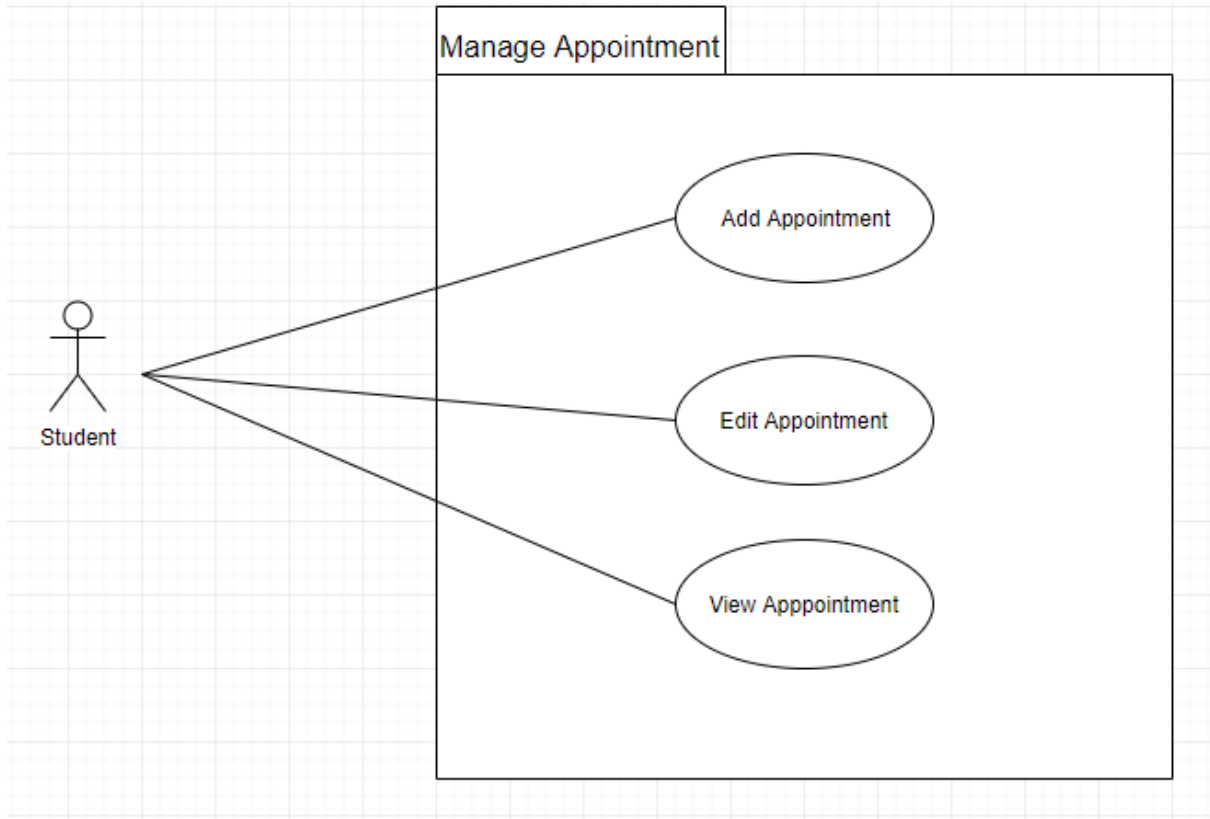


Figure 3.4 Use Case Manage Appointment

### 3.5 Use Case Booking Appointment

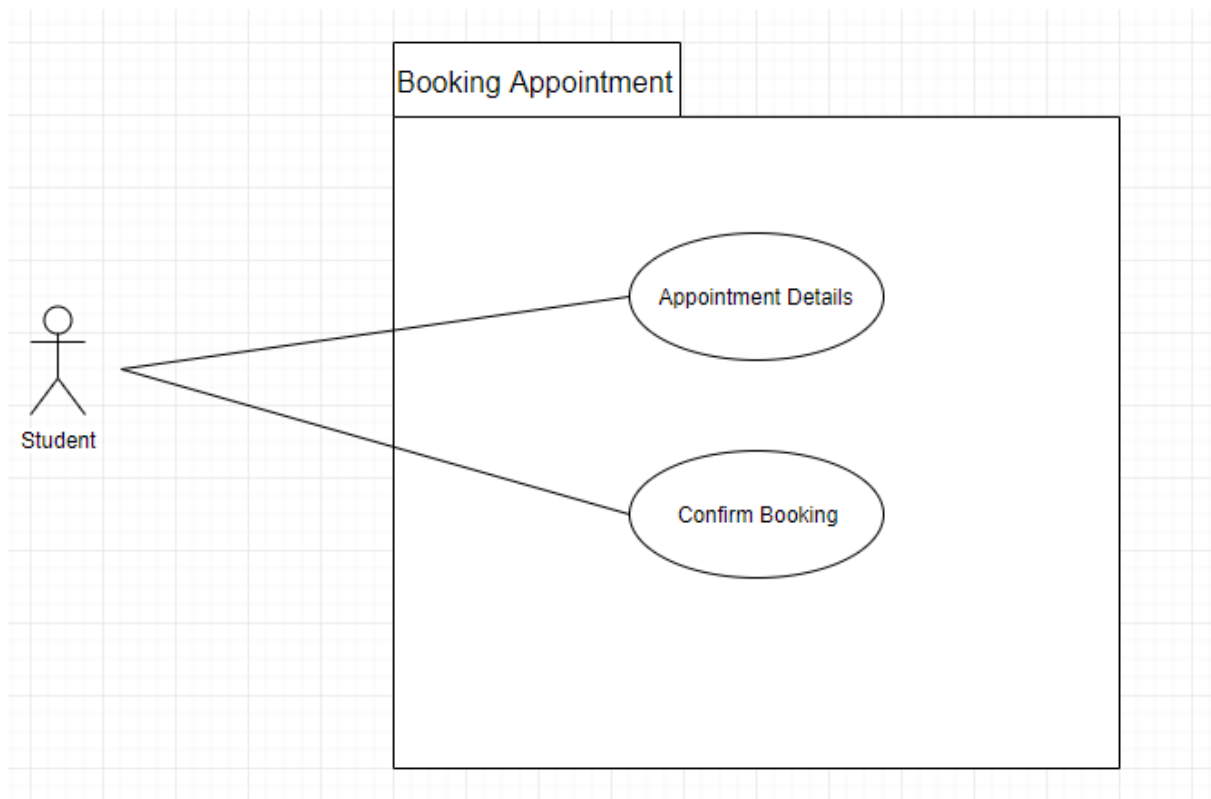


Figure 3.5 Use Case Booking Appointment

## **APPENDIX C: SOFTWARE DESIGN DOCUMENT (SDD)**

**Version**

**1**

## ONLINE APPOINTMENT SYSTEM

---

Faculty Computer Systems & Software Engineering  
(FSKKP)

# Software Design Document (SDD)



**CHAPTER 6Table of Contents**

**1.0 DATA DICTIONARY ..... 4**

**2.0 PRELIMINARY DESIGN ..... 6**

**3.0 DETAILED DESIGN ..... 7**

**3.0 SYSTEM DESIGN APPROVAL..... 7**

## 1.0 DATA DICTIONARY

Table 1.1: Data Dictionary for Student

Field Name	Description	Data Type	Constraint
Id	Student ID	Integer(10)	Primary key
fullName	Student Name	Varchar(50)	-
matricId	Student Matric ID	Integer(10)	-
Email	Email address	Varchar(50)	-
Password	Password of the username to login	Varchar(50)	-

Table 1.2: Data Dictionary for Lecturer

Field Name	Description	Data Type	Constraint
Id	Student ID	Integer	Primary key
Specialization	Lecturer specialization	Varchar(50)	-
lecturerName	Lecturer Name	Varchar(50)	-
lecturerRoom	Lecturer Room	Varchar(50)	-
contactNo	Lecturer Contact Number	Integer	-
lecturerEmail	Lecturer Email	Varchar(50)	-

Password	Password for lecturer to login	Varchar(50)	-
----------	--------------------------------	-------------	---

Table 1.3: Data Dictionary for Appointment

Field Name	Description	Data Type	Constraint
Id	Appointment ID	Integer	Primary key
lecturerSpecialization	Lecturer Specialization	varchar(50)	-
lecturerId	Lecturer ID	Integer	-
lecturerRoom	Lecturer Room	Varchar(50)	-
appointmentDate	Date of appointment	Date	-
appointmentTime	Time of appointment	Time	-

Table 1.4: Data Dictionary for Lecturerspecialization

Field Name	Description	Data Type	Constraint
Id	Lecturer specialization id	int	Primary key
Specialization	Lecturer specialization	Varchar(50)	-
CreationDate	Date of creation	Date	-



## 2.0 PRELIMINARY DESIGN

### 2.1 System Archicture

The E-Appointment has four subsystems which are Student Registration, Manage Student Record, Manage Queue and Manage Appointment. Database is used to keep all the information and can be retrieve anytime if user want to update.

#### 2.1.1 Static Organization

Figure 2.1 shows the static organization of the E-Appointment Application System.

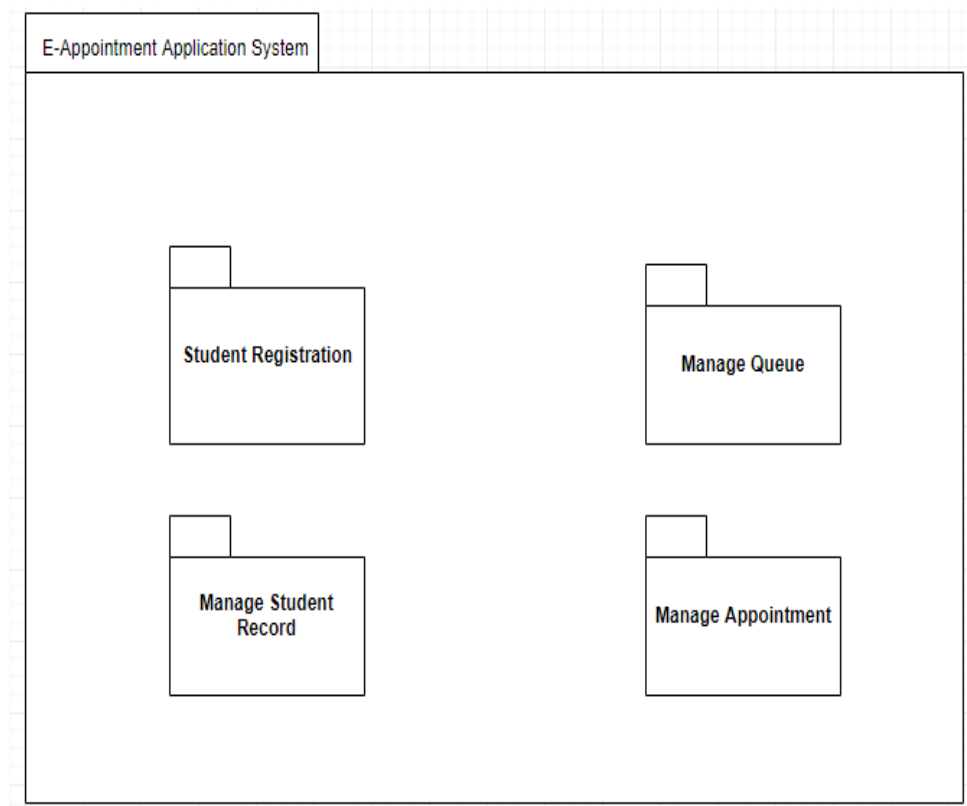


Figure 2.1: Static Organization of the E-Appointment for Student/Lecturer

This section will describe the details for each subsystem/package

#### 1. Student Registration Package

This package describe about the details of student that are newly register by using this system.

This package has:

- i) Class StudentRegistrationUI
- ii) Class StudentRegistrationController
- iii) Class StudentRegistrationModel

#### 2. Manage Student Record

This package describe about how to manage the record of the student that are already exist in the system. It consists:

- i) Class StudentRecordUI
- ii) Class StudentRecordController
- iii) Class StudentRecordModel

#### 3. Manage Queue

This package describe about how to manage queue for the waiting list for student to make appointment with the lecturer. It consists:

- i) Class QueueUI
- ii) Class QueueController
- iii) Class QueueModel

#### 4. Manage Appointment

This package is about the responsible to manage appointment. It has some class:

- i) Class AppointmentUI
- ii) Class AppointmentController
- iii) Class AppointmentModel

### **3.0 DETAILED DESIGN**

#### **3.1 Student Registration**

##### **3.1.1 Class StudentRegistrationUI**

Class Type : View Class

Responsibility : This class is responsible to display all information related to  
the Student Registration.

Attributes : None

Methods : void display data

To display information of Student Registration.

##### **3.1.2 Class StudentRegistrationController**

Class Type : Controller Class

Responsibility : This class is responsible to act as controller between view and  
model class.

Attributes : None

Methods : void getStudentDetails

To retrieve information of Student.

### 3.1.3 Class StudentRegistrationModel

Class Type : Model Class

Responsibility : This class is responsible to manage all the operation that are related to the Student Registration database

Attributes : Id : int  
: name : varchar(50)  
: address :varchar(MAX)  
: day : nchar(10)  
: month : nchar(10)  
: year : nchar(10)  
: telephone : nchar(10)  
: problem : varchar(MAX)

Methods : void addStudentDetails



To add information of Student

## **3.2 Manage Student Record**

### **3.2.1 Class StudentRecordUI**

Class Type : View Class

Responsibility : This class responsible to display information that related to the Student Record.

Attributes : None

Methods : void display data

To display information of Student Record

### **3.2.2 Class StudentRecordController**

Class Type : Controller Class

Responsibility : This class is responsible to act as controller between view and model class

Attributes : None

Methods : void getStudentRecord

To obtain information of Student Record

### 3.2.3 Class StudentRecordModel

Class Type : Model Class

Responsibility : This class is responsible to manage all the operation that are related to the Student Record database.

Attributes : Id : int  
: name : varchar(50)  
: address :varchar(MAX)  
: day : nchar(10)  
: month : nchar(10)  
: year : nchar(10)  
: telephone : nchar(10)  
: problem : varchar(MAX)

Methods : Void updateStudentRecord

To update information of Student

### **3.3 Manage Queue**

#### **3.3.1 Class QueueUI**

Class Type : View Class

Responsibility : This class is responsible to display information that are related to the Queue Details

Attributes : None

Methods : void display data

To display queue list of student

#### **3.3.2 Class QueueController**

Class Type : Controller Class

Responsibility : This class is responsible to act as controller between view and model class.

Attributes : None

Methods : Void getQueue

To obtain information of queue details.

#### **3.3.3 Class QueueModel**

Class Type : Model Class

Responsibility : This class is responsible to manage all the operation that are related to the Queue database.

Attributes : Id : int

: name : varchar(50)

: address :varchar(MAX)

: day : nchar(10)

: month : nchar(10)

: year : nchar(10)

: telephone : nchar(10)

: problem : varchar(MAX)

: date : datetime

Methods : Void addQueue

To add Queue list

: Void updateQueue

To update the Queue list

### **3.4 Manage Appointment**

#### **3.4.1 Class AppointmentUI**

**Class Type** : View Class

**Responsibility** : This class is responsible to display information that related to the Appointment Details

**Attributes** : None

**Method** : void display data

To display appointment details.

#### **3.4.2 Class AppointmentController**

**Class Type** : Controller Class

**Responsibility** : This class is responsible to act as controller between view and model class.

**Attributes** : None

**Methods** : Void getAppointment

To obtain information of appointment details.

#### **3.4.3 Class AppointmentModel**

**Class Type** : Model Class

Responsibility : This class is responsible to manage all the operation that are related to the Appointment database.

Attributes	: AppointmentId	: int
	: AppointmentStart	: datetime
	: AppointmentEnd	: datetime
	: LecturerId	: int
	: AppointmentStudentName	: nvarchar(100)
	: AppointmentStatus	: nvarchar(100)
	: AppointmentStudentSession	: nvarchar(100)
Methods	:Void addAppointment	
	To add the appointment	

i.

#### 4.0 SYSTEM DESIGN APPROVAL

	Name	Date
<b>Verified by:</b>  <hr/> Developer	MUHAMMAD ZUL FAHMI BIN ZAHER	12 December 2018
<b>Approved by:</b>  <hr/> Client		

## **APPENDIX D: USER ACCEPTANCE TEST (UAT)**



**Version**

**1**

ONLINE APPOINTMENT SYSTEM (OAS)

---

Faculty Computer Systems & Software Engineering (FSKKP)

# User Acceptance Test (UAT)

*6.1.1.2*

6.1.1.1 Table of Contents

1.0	TESTING REPORT	3
2.0	SYSTEM TESTING APROVAL	13

List of Table

Table	1.1:	Use	Case
Testing.....			4
Table	1.2:	Use	Case
Login.....			5
Table	1.3:	Use	Case
Registration).....		Manage	6
Table	1.4:	Use	Case
Profile.....			7
Table	1.5:	Use	Case
Appointment.....			7
Table	1.6:	Use	Case
Report.....		View	8

Table	1.7:	Use	Case	Manage	User
(Lecturer).....					9

## 1.0 TESTING REPORT

This section discusses the User Acceptance Test (UAT) process for the system. Functional testing is the testing process that is used for the testing in order to ensure that it conforms to all functionality being specified within functional requirements. Functional testing are composed of few testing type which are unit testing, system testing, integration testing and acceptance testing.

As for the acceptance testing, the system are being tested by the user according to the test cases designed. The test cases designed are regarding the system test process where all the components interaction of the system are included. Any errors or problems occur during testing are noted in the test cases for system correction. The test cases of the system are as follows.

Table 1.1: Use Case Testing

	<b>Event</b>	<b>Test Data</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/ Fail</b>	<b>Comment</b>

Table 1.2: Use Case Login

	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	Login with correct username and password	Enter username and password	System will access into the main page of the system based on the user (Student/ Lecturer/ Admin)	System can login and go to main page	Pass	
2	Login with invalid username	Enter invalid username	System will notify user had invalid match of username and password	User will be notify for inserting valid username	Pass	
3	Login with invalid password	Enter invalid password	System will notify user had invalid match of username and password	User will be notify to insert valid password	Pass	

4	Login with invalid username and password	Enter invalid username and password	System will notify user had invalid match of username and password	User will be notify to enter valid password and username	Pass	
---	--	-------------------------------------	--	--	------	--

Table 1.3: Use Case Manage Student (Student Registration)

	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	Register new student	Click 'Create an account' on sidebar	System will direct user to student registration page		Pass	
2	Insert all required information	Fulfil the required information for register new account	System will stored all the information into database		Pass	
3	Register with same email	Enter the same email to register new account	System will prevent the registration of new account		Pass	

4	Cancel registration	Click “Log-in” to cancel registration	System will direct to student-login page		Pass	
---	---------------------	---------------------------------------	--	--	------	--

Table 1.4: Use Case Manage Profile

	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	Edit profile information	Click on My profile	System will direct to edit-profile page		Pass	
2	Change password	Click on change profile and click "Submit" to update new password	System will store the new password in database		Pass	

Table 1.5 Use Case Manage Appointment



	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	Add new appointment	Click on “Book Appointment”	System will direct user to book-appointment page		Pass	
2	Edit appointment details	Edit related details	System will direct user to book-appointment page to edit appointment		Pass	
3	Delete appointment	Delete the appointment	System will direct admin to manage-student page to delete student appointment		Pass	
4	View appointment	Click on “Appointment History”	System will display all related history information		Pass	

Table 1.6: Use Case View Session Report

	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	View session report	Admin will click “Lecturer session logs” or “Student session logs”	The system will view all session logs for student and lecturer		Pass	

Table 1.7: Use Case Manage User (Lecturer)

	Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
1	Add new lecturer	Add new username, full name and password of lecturer	System will save new lecturer into database and new lecturer can log into the system		Pass	
2	Edit lecturer	Edit lecturer details	System will save updated details into database and lecturer can log into the system using the updated details		Pass	

3	Delete lecturer	Delete lecturer from the system	System will remove lecturer details from the database and lecturer cannot log into the system anymore		Pass	
---	-----------------	---------------------------------	---	--	------	--

## 2.0 SYSTEM TESTING APPROVAL

	Name	Date
<b>Verified by:</b>  <hr/> Developer	MUHAMMAD ZUL FAHMI BIN ZAHER	12 December 2018
<b>Approved by:</b>  <hr/> Client		

## **APPENDIX E: USER MANUAL**

## ONLINE APPOINTMENT SYSTEM (OAS)

---

Faculty Computer Systems & Software Engineering

(FSKKP)

# User Manual

## CHAPTER 7Table of Contents

<b>1.0 GENERAL INFORMATION .....</b>	<b>3</b>
1.1 SYSTEM OVERVIEW .....	3
<b>2.0 SYSTEM SUMMARY .....</b>	<b>4</b>
2.1 SYSTEM CONFIGURATION .....	4
2.2 USER ACCESS LEVEL .....	4
<b>3.0 GETTING STARTED .....</b>	<b>5</b>
3.1 INSTALLATION AND LOGGING IN .....	5
3.2 SYSTEM MENU .....	5
3.2.1 MAIN PAGE .....	6
3.2.2 LOGIN PAGE .....	7
3.2.3 CREATE ACCOUNT PAGE .....	8
3.2.4 LECTURER LOGIN PAGE .....	9
3.2.5 ADMIN LOGIN PAGE .....	10
3.2.6 STUDENT DASHBOARD PAGE .....	11

3.2.7 EDIT PROFILE PAGE .....	12
3.2.8 BOOK APPOINTMENT PAGE .....	13
3.2.9 APPOINTMENT HISTORY PAGE .....	14
3.2.10 CHANGE PASSWORD PAGE .....	15
3.2.11 LECTURER DASHBOARD PAGE .....	16
3.2.12 ADMIN DASHBOARD PAGE .....	17
3.2.13 SESSION LOG PAGE .....	18
3.2.14 MANAGE STUDENT PAGE.....	19
3.2.15 ADD LECTURER SPECIALIZATION PAGE.....	20
3.2.16 ADD LECTURER PAGE .....	21
 <b>3.0 STARTING THE SYSTEM.....</b>	<b>21</b>
 <b>3.0 EXITING THE SYSTEM .....</b>	<b>21</b>



## **1.0 GENERAL INFORMATION**

### **7.1            1.1        SYSTEM OVERVIEW**

Online Appointment System (OAS) is a web based system that is developed for Faculty of Computer System and Software Engineering in order to improve current appointment management of the lecturer has been done manually. The lecturer use the system to manage student's appointment. The system allows to manage student registration, manage profile, manage appointment and view session logs of the system. The objective of developing this system are to improve the manual appointment management system by computerizing the manual appointment management into a system using structured approach in order to develop a prototype of the OAS. Rapid Application Development (RAD) is the methodology used to develop this system where in RAD method, it consists of four stages which are the requirement planning, application design, construction and also cutover phase. The test is done to ensure that all functions of the system is working according to the requirement stated without any error. Therefore, the successful user acceptance test then shows that the system has achieved the objectives of developing this system.

## **2.0 SYSTEM SUMMARY**

### **7.2            2.1        SYSTEM CONFIGURATION**

Online Appointment System is an appointment management system that runs on a web browser. Thus, internet connection is required in order to use the system. User logging can be validated, data can be saved in the database, data retrieval from the database can be done and report generation can be generated as well. After successful logging, the system then can be used without any further configuration.

### **7.3            2.2        USER ACCESS LEVELS**

The access levels for the system are categorized into three levels. First level is on the admin of the system and the second level is for student and the third level is for the lecturer access. Admin of the system can add, delete, view and update users of the system. Admin also can view the session logs of the user in the system.

Second access level of the system is the student where the management of student is initiated by themselves since registration is done at the beginning of the work flow. Student can register, book appointment and manage profile.

Third access level of the system is the lecturer where they can view appointment application, add schedule to the system and manage profile.

### **3.0 GETTING STARTED**

This sections describes and shows the instructions to user on how to use and manage the system.

#### **7.4            3.1        INSTALLATION AND LOGGING IN**

This system requires internet connection to allow running on the web browser. It is compatible to be run in any of the web browser available like the Internet Explorer, Mozilla Firefox, Torch browser and Google Chrome.

#### **7.5            3.2        SYSTEM MENU**

This section describes the step by step guidance to user on how to use the system. The details on how each interface design works is described and instructed in this section.

##### **7.5.1            3.2.1    Main Page**

Figures below show the main page of the system for the student, lecturer and admin. The main page for student and lecturer are same design but for admin there is a slight different in the design. However, the ways it works is just the same.




Figure 3.1: Main Page


The figure 3.1 above shows the main page of the system. There are three login type which refer to student login, lecturer login and admin login. When the user click on the click here link, the user will be redirected to the page according to their types.


# OAS | Students Login

[Sign in to your account](#)

Please enter your name and password to log in.

 Username

 Password

Login 

---

Don't have an account yet? [Create an account](#)

© 2018 OAS. All rights reserved

Figure 3.2: Student Login Page

Figure 3.2 shows the login page for student. The student need to insert username, password and then click the login button. After successful login, student is directed to the page where student can manage the profile or make appointment booking.

# Student Registration

## Sign Up

Enter your personal details below:

Enter your account details below:

☐ I agree

Already have an account? [Log-in](#)

© 2018 OAS. All rights reserved

Figure 3.3: Create Account Page

Figure 3.3 shows the creating new account for student. Student can create new account by inserting full name, their matric id, email, password and click the submit button to finish the

creating account process. If the student already registered in the system, they can directly login to the system by clicking Log-in link.

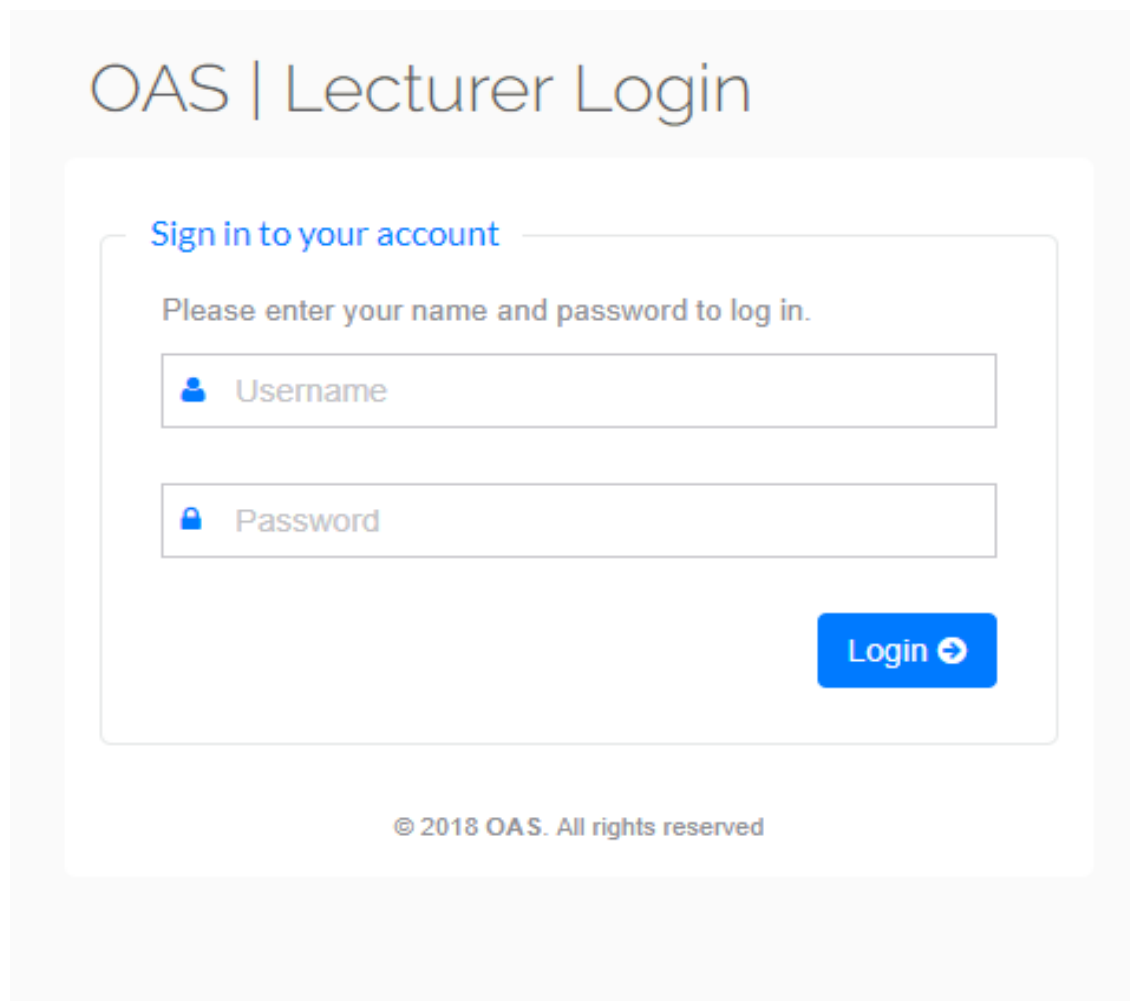
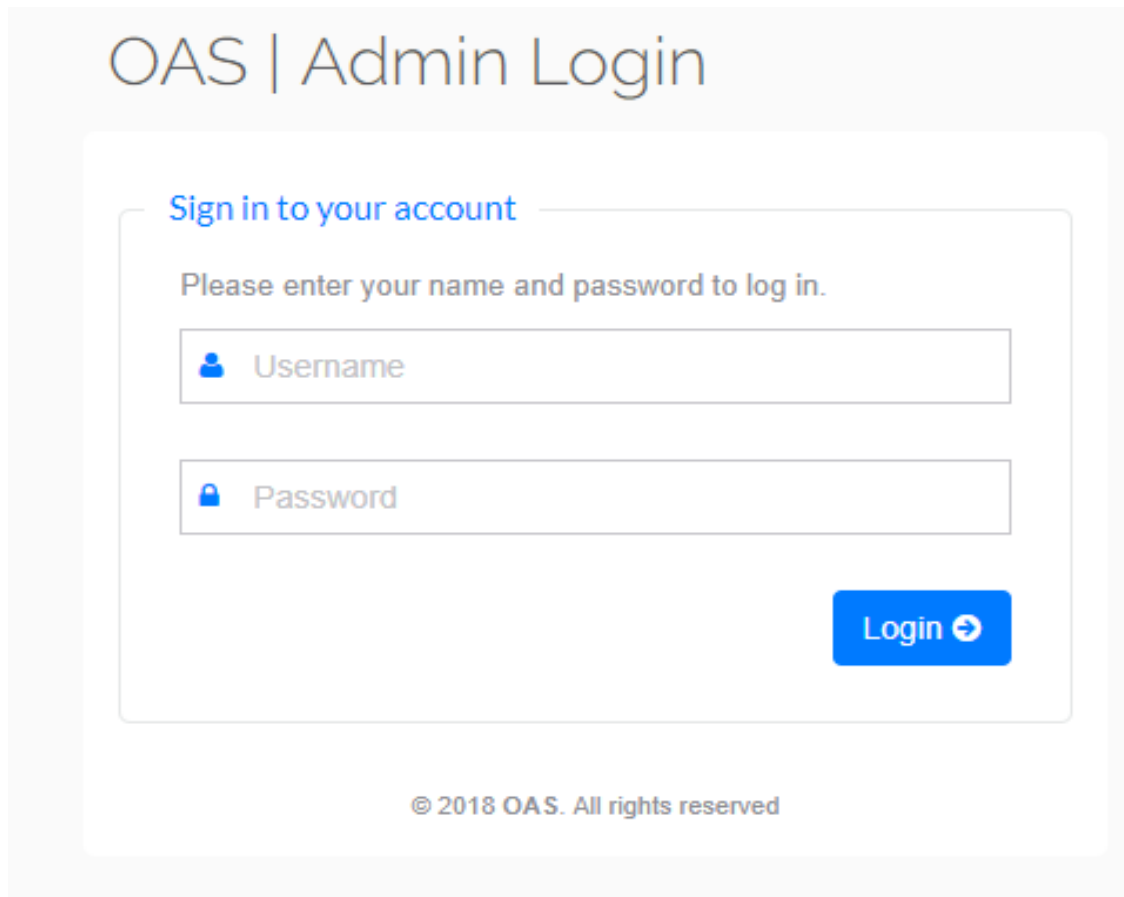
The image shows a web page titled "OAS | Lecturer Login". Below the title is a section titled "Sign in to your account" in blue text. Underneath this is a prompt: "Please enter your name and password to log in." There are two input fields: the first is labeled "Username" with a person icon, and the second is labeled "Password" with a lock icon. To the right of the password field is a blue "Login" button with a right-pointing arrow. At the bottom of the form area, there is a copyright notice: "© 2018 OAS. All rights reserved".

Figure 3.4: Lecturer Login Page


Figure 3.4 shows the lecturer login page. Lecturer need to insert username and password in order to use the system. By clicking the login button, lecturer is redirect to lecturer management page.





OAS | Admin Login

[Sign in to your account](#)

Please enter your name and password to log in.

 Username

 Password

Login 

© 2018 OAS. All rights reserved

Figure 3.5: Admin Login Page

Figure 3.5 shows the admin login page. Admin need to enter the username and password. After successful login, admin is directed to the page where admin can manage user of the system.



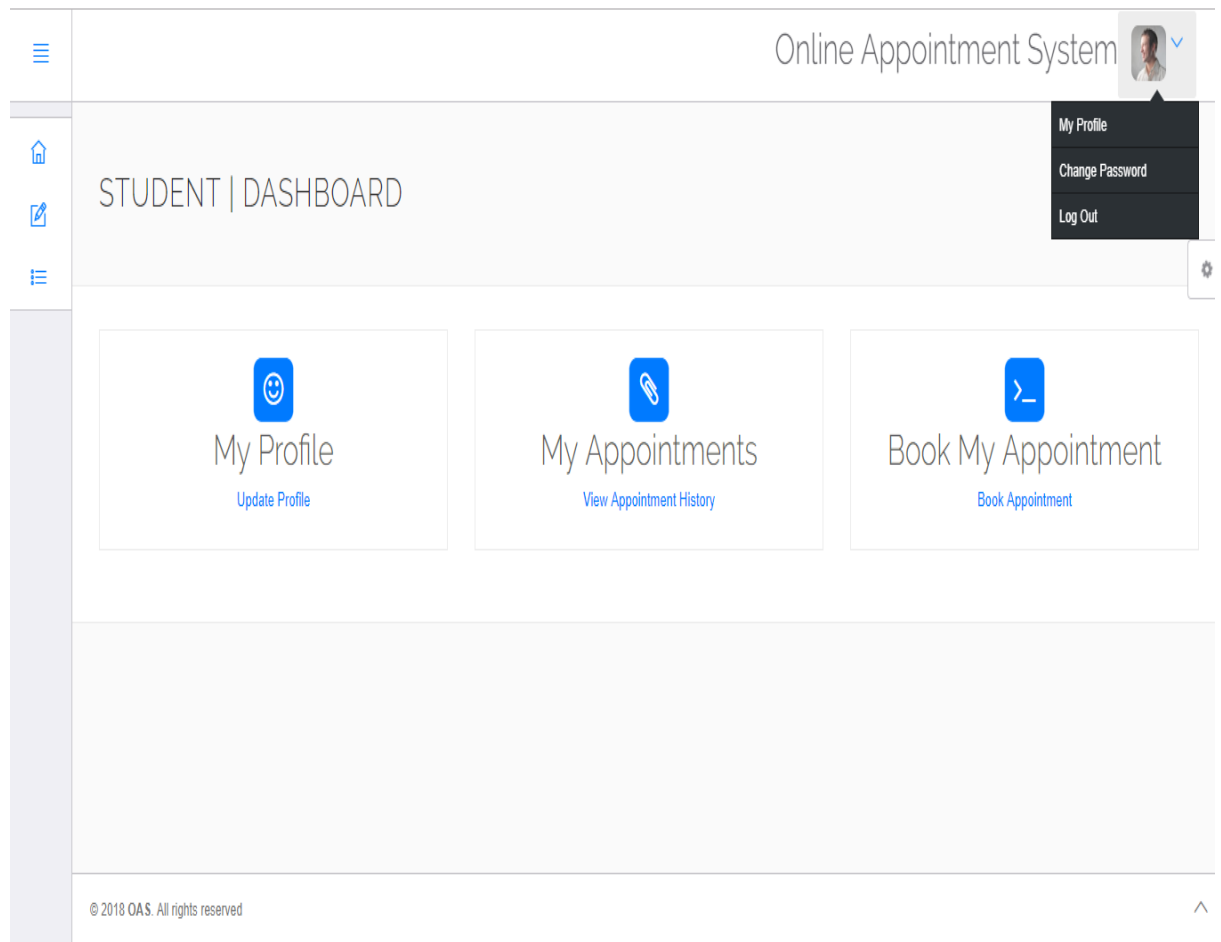


Figure 3.6: Student Dashboard Page

Figure 3.6 shows the student dashboard page. This page will allow student to view their profile, book an appointment and also view their appointment history. There also exist navigation panel on the left side of the page. On the top right corner of the page, student can view their profile, change password and also logout from the system.

Online Appointment System

STUDENT | EDIT PROFILE

STUDENT / EDIT PROFILE

Edit Profile

User Name

muhammad

Address

City

User Email



muhdzulfahmi@gmail.com

[Update your email id](#)

Update

Figure 3.7: Edit Profile Page

Figure 3.7 shows the edit profile page for student. Student can edit their username, address, city and email. By clicking the update button, all the new information will be updated to the database.

STUDENT | BOOK APPOINTMENT

STUDENT / BOOK APPOINTMENT

Book Appointment

Lecturer Specialization

Software

Lecturers

Select Lecturer

Lecturer Room

Date

Time

11:15 AM

eg : 10:00 PM

Submit




Figure 3.8: Book Appointment Page

Figure 3.8 shows the booking appointment page. Student can book their appointment by entering required information and click the submit button to confirm the booking.

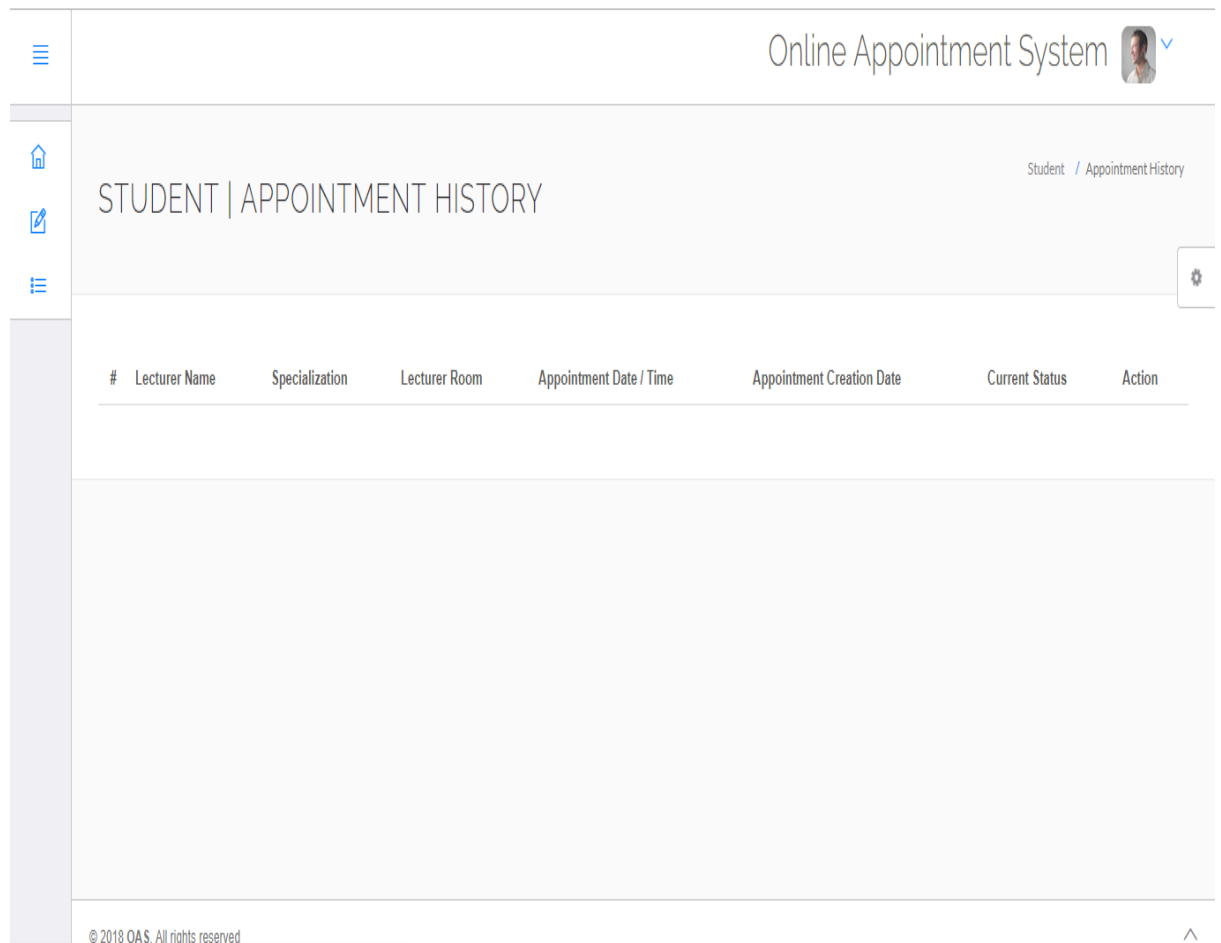


Figure 3.9: Appointment History Page

Figure 3.9 above shows the appointment history page for student. The page will view all the history of appointment made by student before.

The screenshot displays a web interface for an 'Online Appointment System'. At the top right, the system name is shown next to a user profile icon with a dropdown arrow. Below this, a navigation bar contains the text 'STUDENT | CHANGE PASSWORD' and a breadcrumb trail 'Student / Change Password' with a settings gear icon. The main content area is titled 'Change Password' and contains three input fields: 'Current Password' (placeholder: 'Enter Current Password'), 'New Password' (placeholder: 'New Password'), and 'Confirm Password' (placeholder: 'Confirm Password'). A blue 'Submit' button is located at the bottom of the form.

Figure 3.10: Change Password Page

Figure 3.10 shows the change password page. This page allow student to update their new password by entering required information. By clicking submit button, the new password will be updated to the database.

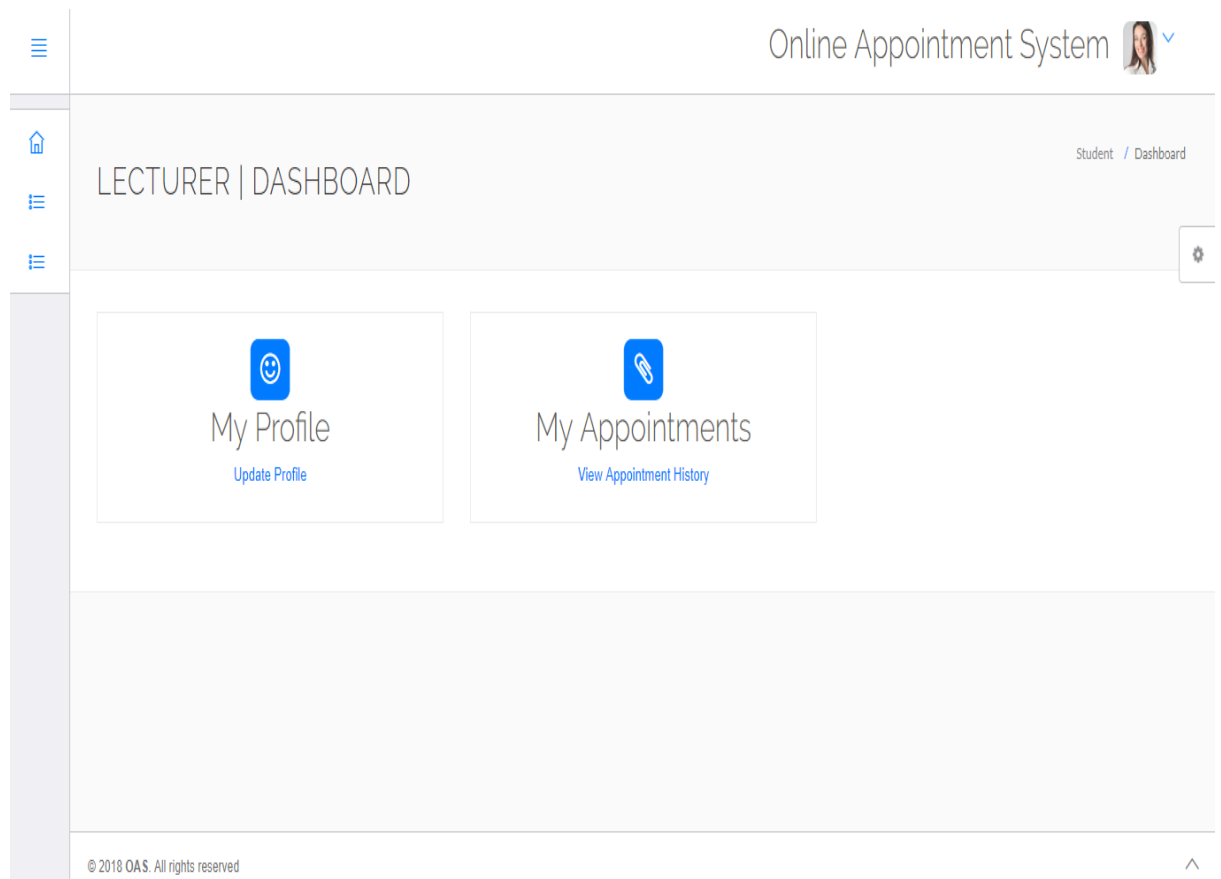


Figure 3.11: Lecturer Dashboard Page

Figure 3.11 shows the lecturer dashboard page. Lecturer can view their profile, appointment application list and add schedule.

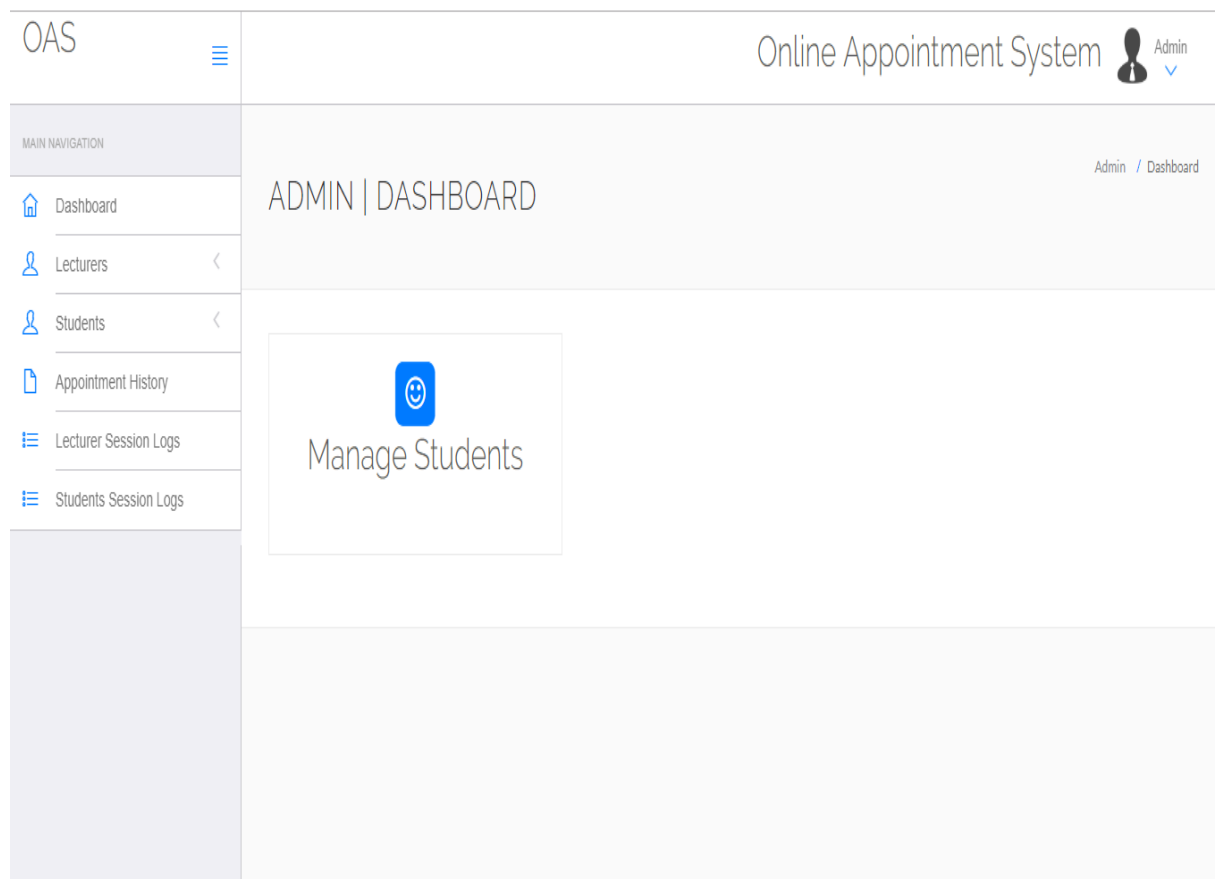


Figure 3.12: Admin Dashboard Page

Figure 3.12 shows the admin dashboard page. Admin can manage lecturer, student, view history of appointment and view the session logs of students and lecturers.

OAS

MAIN NAVIGATION

Dashboard

Lecturers

Students

Appointment History

Lecturer Session Logs

Students Session Logs

Online Appointment System

Admin

ADMIN | LECTURER SESSION LOGS

#	User id	Username	User IP	Login time	Logout Time	Status
1.	0	roslina@gmail.com	::1	2018-12-10 12:36:36		Failed
2.	0	roslinams@ump.edu.my	::1	2018-12-10 12:38:49		Failed
3.	0	fadli@ump.edu.my	::1	2018-12-10 13:51:10		Failed
4.	4	fadli@gmail.com	::1	2018-12-12 11:15:46	12-12-2018 04:23:05 AM	Success

© 2018 OAS. All rights reserved

Figure 3.13: Session Logs Page for Lecturer

Figure 3.13 shows the lecturer session logs. The lists show that the details of session log either the lecturer success or fail to access the system.



OAS Online Appointment System Admin

ADMIN | MANAGE STUDENTS Admin / Manage Students

MAIN NAVIGATION

- Dashboard
- Lecturers
- Students
- Appointment History
- Lecturer Session Logs
- Students Session Logs

Manage Patients

#	Full Name	Matric ID	Email	Creation Date	Updation Date	Action
1.	Muhammad Zul Fahmi	15083	zulfahmi@gmail.com	2018-12-08 07:03:09	10-12-2018 11:27:47 AM	X
2.	Muhammad	15080	muhammad@gmail.com	2018-12-08 07:03:09	10-12-2018 11:50:00 AM	X
3.	muhammad	15022	muhdzulfahmi@gmail.com	2018-12-11 04:32:45		X

Figure 3.14: Manage Student Page

Figure 3.14 shows the manage student page. Admin can delete the student from the list if need to do so.

OAS

Online Appointment System

Admin

MAIN NAVIGATION

Dashboard

Lecturers

Students

Appointment History

Lecturer Session Logs

Students Session Logs

ADMIN | ADD LECTURER SPECIALIZATION

ADMIN / Add lecturer specialization

Lecturer Specialization

Lecturer Specialization

Enter Lecturer Specialization

Submit

Manage Lecturer Specialization

#	Specialization	Creation Date	Update Date	Action
1.	Software	2018-12-01 06:37:25		<a href="#">/</a> <a href="#">x</a>
2.	Networking	2018-12-01 06:38:12		<a href="#">/</a> <a href="#">x</a>
3.	Graphic	2018-12-01 06:38:48		<a href="#">/</a> <a href="#">x</a>

Figure 3.15: Add Lecturer Specialization Page

Figure 3.15 shows the add lecturer specialization page. Admin can add, delete or update the type of lecturer specialization. When the submit button is clicked, the information will be update in the database.

OAS

Online Appointment System

Admin

MAIN NAVIGATION

Dashboard

Lecturers

Students

Appointment History

Lecturer Session Logs

Students Session Logs

Add Lecturer

Lecturer Specialization

Select Specialization

Lecturer Name

Enter Lecturer Name

Lecturer Room

Enter Lecturer Room

Lecturer Contact no

Enter Lecturer Contact no

Lecturer Email

Enter Lecturer Email id

Password

New Password

Confirm Password

Confirm Password

Submit

Figure 3.16: Add Lecturer Page

Figure 3.16 shows the adding lecturer page. Admin will add new lecturer by inserting the required information and click the submit button. The new information about new lecturer will be store in the database.

### 3.3 STARTING THE SYSTEM

To start using the system, user need to ensure that the device is connected with the internet. Then type in the address of the system in the browser. The login page will appear when the address is fulfil and user can start the operation of appointment management.

### **3.4 EXITING THE SYSTEM**

To exit the system, user just has to click on the logout button. User need to point their pointer to the profile on the top right corner of page and click logout button. User successfully logout from the system as the page is redirected to the main page of the Online Appointment System.