



Online Appointment Management System for New Philip Hospital

H.D Damith Madushan

BIT Registration number: R141139

Index number: 1411391

Name of the supervisor: Mr. H.M.D.S Herath

2017



**This dissertation is submitted in partial fulfilment of the
requirement of the Degree of Bachelor of Information
Technology (external) of the University of Colombo School of
Computing**

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a Degree or Diploma in any University and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations.

Signature of Candidate:*Damith*.....

Date: *02/11/2017*

Name of Candidate: H. D Damith Madushan

Countersigned by:

Signature of supervisor:*Santh*.....

Date: *02/11/2017*

Name of Supervisor: Mr. Daminda Santha Herath

Abstract

New Philip Hospital is a country's leading hospital, situated in the heart of the Kalutara Town, which was established in 1955. They are well-known for their high quality health care services. New Philip Hospital don't have their own Appointment management system currently. They use third party services to perform day to day appointment management activities. This process has few disadvantages. Since they are using 3rd party services, hospital has to pay for those service providers for their services and patients have to pay some extra money when they are booking appointments through these services. Another disadvantage is these systems do not provide all functionalities required by hospital since they are common systems.

The new system going to help users in numerous ways. New system will offer patient records management, fast online and local payments, up-to-date and accurate records that will help and benefit to patients, doctors and system admins.

Rational Unified Process model was chosen for the development methodology which supports Object Orientation and iterative development process. The system was developed using the PHP open source server side scripting language. Apache web server was chosen to run the system. MySQL Database Management System was used for manipulating and storing records.

Acknowledgement

I would like to take this opportunity to express my gratitude to everyone who helped me during this project. First of all I wish to express my gratitude to BIT Coordinator of University of Colombo School of Computing (UCSC) and project examination board of Bachelors of Information Technology (BIT) for giving me this valuable opportunity to follow a world recognized degree program.

I wish to thank my project supervisor Mr.Daminda Santha Herath who guided me and giving technical support during this project. I take this opportunity to express my gratitude to staff of Esoft Metro Campus for providing me academic and additional guidance during degree program.

I would like to acknowledge with much appreciation the assistance and guidance provide by Mr.Damson Fernando (Director of New Philip Hospital) which was really helpful for the successful completion of the project. Also special thanks goes to Dr.Christo Fernando (Chairman of New Philip Hospital) for his great support. I wish to express my gratitude to Mrs.Dilani, Mrs.Achini and New Philip Hospital staff for their support which was really helpful for the successful completion of the project.

I am also grateful for my parents for their unconditional love and support given me all the time. I take this opportunity express my gratitude to my friends and relatives who gave me a great support throughout the process of this three years degree program.

Table of Contents

Declaration	ii
Abstract	iii
Acknowledgement.....	iv
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xi
LIST OF ACRONYM.....	xii
CHAPTER 1 - INTRODUCTION.....	1
1.1 About New Philip Hospital	1
1.2 Problem Domain	1
1.3 Motivation for the project	1
1.4 Objectives of the project.....	2
1.5 Scope.....	2
1.6 Structure of Dissertation.....	3
CHAPTER 2 - ANALYSIS.....	5
2.1 Current System.....	5
2.2 Drawbacks of Existing System.....	6
2.3 Requirements Gathering.....	6
2.4 Fact finding techniques	6
2.5 Functional Requirements.....	7
2.6 Non-functional Requirements.....	8
2.6.1 Security.....	9
2.6.2 Usability.....	9

2.6.3	Efficiency	9
2.6.4	Accuracy	9
2.6.5	Reliability.....	9
2.6.6	Privacy	9
2.7	Outline of Existing Similar Solutions	10
2.7.1	Bayanno Hospital Management System	10
2.7.2	Hemas Hospitals.....	11
CHAPTER 3 - DESIGN.....		12
3.1	Introduction	12
3.2	Software Process Models.....	12
3.2.1	Waterfall Model	13
3.2.2	Spiral Model	13
3.2.3	Prototyping Model	13
3.2.4	Rapid Application Development (RAD) Model.....	13
3.3	Methodology for the proposed system	13
3.4	Alternative Solutions.....	14
3.5	Selected Solution	15
3.5.1	High level use case diagram for proposed system.....	16
3.5.2	Activity Diagram for Patient Registration	17
3.5.3	Sequence Diagram for put appointments.....	18
3.5.4	Class diagram for proposed system	19
3.6	Database designing	19
3.6.1	Entity Relationship Diagram for proposed system	19
3.6.2	Database Normalization.....	20
3.7	User Interface Design.....	21
3.7.1	Website Interface.....	22

3.7.2	Login interface.....	23
3.7.3	Main Interface.....	24
3.7.4	User Management Interface.....	26
3.7.5	Add user interface.....	27
3.7.6	Notifications.....	27
3.7.7	Confirm Messages.....	27
CHAPTER 4 - IMPLEMENTATION.....		28
4.1	Introduction	28
4.2	Hardware and software Requirements.....	28
4.3	Development tools.....	29
4.4	Languages and Technologies	30
4.5	Reused existing codes	30
4.6	Module Structure.....	32
4.7	Major Code Segments.....	34
4.7.1	Database Connection	35
4.7.2	Session Start.....	36
4.7.3	Logout and unset sessions	36
4.7.4	Encryption	36
4.7.5	Client Side Validation	37
4.7.6	Server Side Validation	38
CHAPTER 5 - EVALUATION.....		39
5.1	Introduction	39
5.2	Testing Strategies.....	39
5.2.1	Unit Testing	39
5.2.2	Black box Testing.....	39
5.2.3	White Box Testing	40

5.2.4	Integration Testing	40
5.2.5	System Testing	40
5.3	TEST PLAN AND TEST CASES.....	40
5.4	Test Results	45
5.5	User acceptance testing.....	49
CHAPTER 6 - CONCLUSION.....		50
6.1	Problems encountered.....	50
6.2	Critical assessments of the project	51
6.3	Lessons Learnt.....	51
6.4	Future Improvements	51
REFERENCES		52
APPENDIX A - SYSTEM DOCUMENTATION.....		56
APPENDIX B - DESIGN DOCUMENTATION.....		59
APPENDIX C – USER DOCUMENTATION.....		67
APPENDIX D - MANAGEMENT REPORTS.....		75
APPENDIX E – TEST RESULTS		78
APPENDIX F – CODE LISTINING		81
APPENDIX G - CLIENT CERTIFICATE.....		88
GLOSSARY		89
INDEX		90

LIST OF FIGURES

Figure 2.1 - Top level Use case diagram for current system.....	5
Figure 2.2 - Bayanno Hospital Management System.....	10
Figure 2.3 - Hemas Hospitals [4]	11
Figure 2.4 - Screenshot of Hemas Channel Online [4]	11
Figure 3.1 - Rational unified process	14
Figure 3.2 - High level use case diagram for proposed system.....	16
Figure 3.3 - Activity Diagram for Patient Registration.....	17
Figure 3.4 - Sequence diagram for put appointments	18
Figure 3.5 - Class diagram for proposed system.....	19
Figure 3.6 - Entity Relationship Diagram	20
Figure 3.7 - Website Home Page	23
Figure 3.8 - System login Screen	24
Figure 3.9 - Admin dashboard	25
Figure 3.10 - User Management.....	26
Figure 3.11 - Add user interface	27
Figure 3.12 - User Notification	27
Figure 3.13 - Schedule cancellation confirm message	27
Figure 4.1 - Twilio sms system.....	32
Figure B.1 - Use case diagram for payments	59
Figure B.2 - Use case diagram for schedule management	60
Figure B.3 - Use case diagram patient registration	61
Figure B.4 - Use case diagram for add a user	62
Figure B.5 - Sequence diagram for schedule management.....	63
Figure B.6 - Sequence diagram for report generation.....	64
Figure B.7 - Activity diagram for backup module	65
Figure B.8 - Database table structure for few tables	66
Figure C.1 - Login screen of the system	67
Figure C.2 - System dashboard.....	68
Figure C.3 - Logout screen	68
Figure C.4 - Notifications for users	69

Figure C.5 - Add user interface.....	69
Figure C.6 - Verification example	69
Figure C.7 - User functionalities example	70
Figure C.8 - Update, view, activate or deactivate user profiles	70
Figure C.9 - Add an appointment.....	70
Figure C.10 - Search for an appointment	71
Figure C.11 - Disable old dates.....	71
Figure C.12 - Search error message	72
Figure C.13 - Channel your doctor	72
Figure C.14 - Appointment booking	72
Figure C.15 - Payment completed message	73
Figure C.16 - View appointments	73
Figure C.17 - View appointments page.....	74
Figure C.18 - Invoice for appointment.....	74
Figure D.1 - Daily income report.....	75
Figure D.2 - Report for system users by role	76
Figure D.3 - Detailed overall user report	76

LIST OF TABLES

Table 4.1 - Hardware and Software Requirements	28
Table 4.2 - Languages and Technologies.....	30
Table 5.1 - Test case for login.....	42
Table 5.2 - Test case for patient registration.....	42
Table 5.3 - Test case for user management.....	43
Table 5.4 - Test case for schedule management	44
Table 5.5 - Test case for payment management.....	44
Table 5.6 - Test case for notification management	45
Table 5.7 - Test results for Login.....	47
Table 5.8 - Test Results for patient registration.....	48
Table 5.9 - Overall test questions.....	49
Table 5.10 - User ratings for questions	49
Table A.1 - Minimum hardware requirements.....	56
Table A.2 - Minimum software requirements.....	57
Table B.1 - Payments	60
Table B.2 - Schedule management.....	61
Table B.3 patient registration.....	62
Table B.4 - Add a user	63
Table E.1 - Test results for user management.....	79
Table E.2 - Test results for notification management	80

LIST OF ACRONYM

1NF	– First Normal Form
2NF	– Second Normal Form
3NF	– Third Normal Form
AJAX	– Asynchronous JavaScript and XML
CSS	– Cascading Style Sheet
DBMS	– Database Management System
ER	– Entity Relationship
GUI	– Graphical User Interface
HTML	– Hypertext Mark Up Language
MVC	– Model View Controller
PK	– Primary Key
PHP	– Hypertext Preprocessor
RAD	– Rapid Application Development
SQL	– Structured Query Language
URL	– Uniform Resource Locator
UML	– Unified Modelling Language

CHAPTER 1 - INTRODUCTION

1.1 About New Philip Hospital

New Philip Hospital is one of the oldest private hospital in Sri Lanka. It's situated in heart of Kalutara city. The hospital was established in 1955.They are providing advanced medical technology and expert medical care to patients.

New Philip Hospital has 75 Beds,3 Operating Theatres, an 8 bed ICU , a Dialysis Unit with 4 Beds and an Out Patient Department (OPD), 24Hour Emergency Treatment Unit (ETU), Pharmacy and an Ambulance service and well trained doctors ,nurses and hospital staff.

Their vision is *“To be dynamic & HealthCare Provider, thereby achieving superior patient care”* which is proven by New Philip Hospital staff. They treat their patients with love and care to until patients leave New Philip Hospital with a smile.

1.2 Problem Domain

Information technology involves businesses, day to day life more than ever. When Information Technology comes to hospitals it affects to almost every activity in a hospital. E-channeling is the new and growing trend in hospital industry. Patients would like to check about doctors and available sessions when they stay in home. In past few years Sri Lankan hospitals also began to develop their own e channeling systems. Currently New Philip Hospital using 3rd party systems to e channeling and appoint management which is costly and causing some inconsistency issues.

1.3 Motivation for the project

New Philip Hospital using several 3rd party services for eChannelling & Appointment management purpose. These Systems causing several problems to systems admins & patients.

- Hospital and patients have to pay for service providers for using their systems.
- Does not provide complete functionalities to users under one system.
- Record Patient information manually.
- Lack of system processes to generate reports.

- Concerns relating to security and privacy issues.
- Inability to get patient's past treatment information.

1.4 Objectives of the project

The main objective was to implement an Online Appoint Management System that cover most of eChannelling and appoint management requirements with high level information availability and security.

Objectives for the proposed systems are as follows,

- Ability to make online payments securely
- Efficient patient management
- Generate reports and analytical information accurately.
- Support backing up patient, payment and other necessary information.
- Send relevant notifications to patients timely.
- Ability to track all employee activities.
- Support making errorless schedules for e channeling.

1.5 Scope

Scope of the proposed system is determined by client's requirements and allocated time for the project.

- **Payments**

Patients should be able to pay for their appointments online

- **Patient Management**

Admin should be able to manage patient details. Patients should be able to view their treatments records, manage their profiles.

- **Doctor Management**

Admin should be able to create, manage doctor profiles. Doctors should be able to view appointments, manage their profiles, view patients records.

- **Schedule Management**

Admins should be able to create schedules, manage schedules.

- **Session Management**

System should be able to handle login sessions, transactions securely and accurately.

- **User Management**

Admin should be able to manage all users. Grant permissions to users, access control, activate/deactivate users.

- **Notification**

System should be able handle notifications for users. System can send emails, mobile notifications and send messages between system users.

- **Employee Tracking**

System should be able to track user activities, generate reports using log records.

- **Backup**

System admin should be able to back up system.

- **Reports Management**

System should be able to generate various kind of reports according to user requirements.

- **Additional Treatment Management**

System has ability to add, view records about additional treatments.

- **Billing**

System has ability to generate bills for various payments.

1.6 Structure of Dissertation

The dissertation contains man six chapters including Introduction to provide overall understanding about the system.

CHAPTER 01 - INTRODUCTION

Briefly explain about system scope, motivation and objectives.

CHAPTER 02 – ANALYSIS

This chapter describe about the current system, requirement gathering and functional and non-functional requirements.

CHAPTER 03 – DESIGN

This chapter defines different design methodologies for the system. This chapter also includes relevant Use Case Diagrams, Class Diagrams and Sequence Diagrams. This chapter gives some features of Graphical User Interfaces.

CHAPTER 04 – IMPLEMENTATION

This Chapter describes the development procedure of the system. It will describe the implementation environment (Hardware and software), existing code, development tools that was used and source code samples from implementation code base.

CHAPTER 05 – EVALUATION

This chapter describes the about test cases planning and test results. User acceptance testing is also carried out at this stage.

CHAPTER 06 – CONCLUSION

This chapter will discuss lessons leant from this project and gives details about recommended further improvements.

CHAPTER 2 - ANALYSIS

Requirements gathering and analyzing is one of the main phase in software development process. This chapter gives overview of existing system, drawbacks of present system as well as fact gathering techniques, functional and non-functional requirements of the proposed system. Also discuss similar systems to proposed system.

2.1 Current System

Currently New Philip Hospital uses a 3rd party systems for eChannelling. It is a common system any hospital can use. Therefore it does not provide all requirements required by New Philip Hospital. It only provide common features to users.

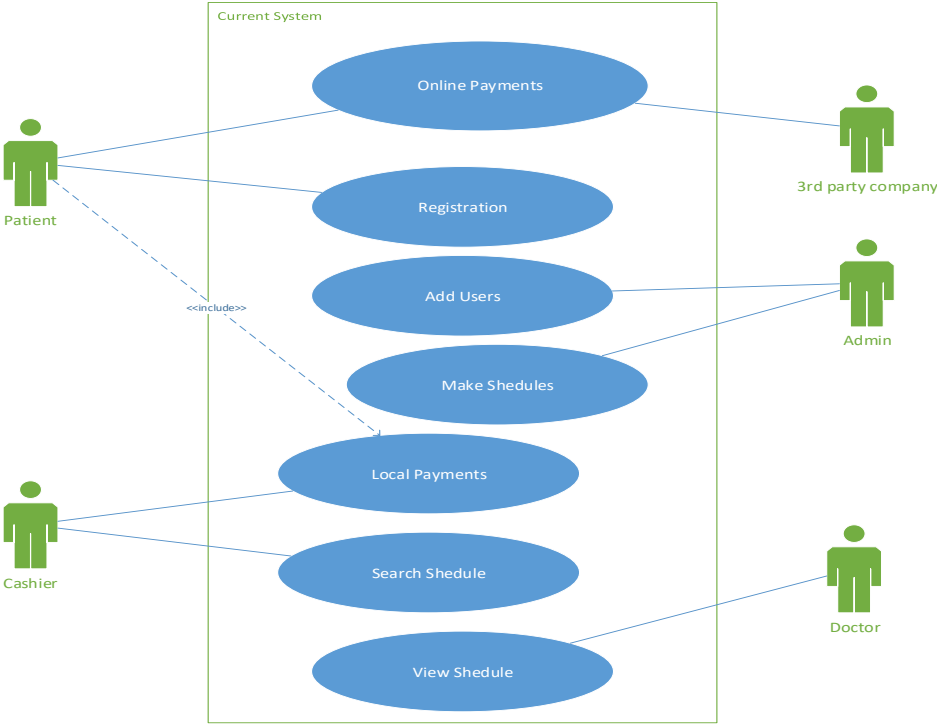


Figure 2.1 - Top level Use case diagram for current system

2.2 Drawbacks of Existing System

The following major drawbacks identified from existing system

- There is no proper synchronization process between hospital appointment management system and e channeling systems. Since few e-channeling services provide their services.
- Hospital has no control over online appointment booking and payments.
- Patients have to pay considerable amount of extra money to third parties for eChannelling.
- There is no way to check patients' past treatment records.
- Hospital can't add doctor profiles.
- Current system provide only basic reports to admins.
- Poor communication between customers and hospital.

2.3 Requirements Gathering

Requirement gathering is very important stage in software development process. Gathering requirements earlier stages in project is very important for success of the project.

2.4 Fact finding techniques

Fact finding techniques describe the different ways to identify the different type of requirements. In order to find functional and non-functional requirements, various fact finding techniques has been used on stakeholders. Some of fact finding techniques are interviews, observation and document review.

Interviews are the most common fact gathering technique used when gathering requirements. Several interviews were conducted with the director and the hospital staff.

Another fact gathering technique is observation. Through observation it could be understood how the overall system works. Using document review technique it could be understood how the current documenting process and billing systems works.

2.5 Functional Requirements

A functional requirement describes what the system should do. Based on functional requirements, a software engineer determines the requirements to be implemented by the software solution. [1]

Payments

- System should allow make online payments for registered and non-registered patients.

Patient Management

- Patients can register as a new user.
- Patients can search schedule using date, doctor specialty and doctor name.
- Registered patients can make appointments after login into the system. Non registered patients can make appointment after providing necessary valid information to the system.
- System should allow patient to edit their profile.
- System should allow patients to provide a feedback about their experience, complains or suggestions.

Doctor Management

- Doctor should be able to search schedule.
- Doctors can view their daily schedule.
- Doctors can search patient's records.
- System should allow print daily patients list.

Schedule Management

- Admin should be able to create daily schedule for doctors.
- System should allow admins to view, update or search schedules.

Session Management

- System should be able to create, manage and control user and transaction sessions.

User Management

- System should allow add new users with different privileges after providing required valid information.
- Users can log in to system by providing valid username and password.
- System should allow users to change password.
- Users can logout from system.

Notification

- System can send email and mobile notifications.
- System should display notifications user receive a message.
- Authorized users should be able to create, edit, and send notifications.

Employee Tracking

- System can track user activities, logins and transactions.

Backup

- System should provide necessary facilities to backup systems.

Reports Management

- System should be able to generate various kind of reports.
- Admin should be able to view reports.
- Admin should be able to search for reports by date, person or other relevant sorting types.

Additional Treatment Management

- System should be able to accept payments, store data recommended by doctor for a patient.
- Patients should be able to view their treatment reports.

Billing

- System should be able to generate various kind of billings and print them.

2.6 Non-functional Requirements

A Non-functional requirement describes the how the system works. Sometime these requirements are called quality attributes of the system. [6]

2.6.1 Security

The system should use security mechanisms to ensure user data, transaction data are securely stored and only accessible by authorized users. Password encrypting, access control, warning messages, verification emails are used to ensure system functioning securely.

2.6.2 Usability

Usability means how easy to interact with the software system interfaces. [7]

System should be useful to users in various ways. Ability to put online appointments, access past medical records, view appointments and ability to easily communication between system admin make system very usable to users.

2.6.3 Efficiency

Efficiency is the ability to achieving maximum productivity using minimum inputs. In a more general sense, it is the ability to do things well without waste. [8]

System allows users to check schedule, put appointments online. This function is saving user time and resources preventing need of physically travelling to hospital, ask for a schedule and put appointments.

2.6.4 Accuracy

System should be able to process online payments, local payments accurately. Also making schedules, putting appointments without duplicating issues is necessary.

2.6.5 Reliability

Reliability refers to perform intended functions and operations in a system's environment, without experiencing failures. [9]

System should be operated without a failure. To ensure system is functioning without failures and user data is well protected, a backup module is integrated to the system. Backup module ensure all user data is backed up without a failure.

2.6.6 Privacy

System is designed to ensure user privacy is well protected. Access control is used to control which user has access to which data, modules and functions. Only relevant patient and doctors can view patient's medical records.

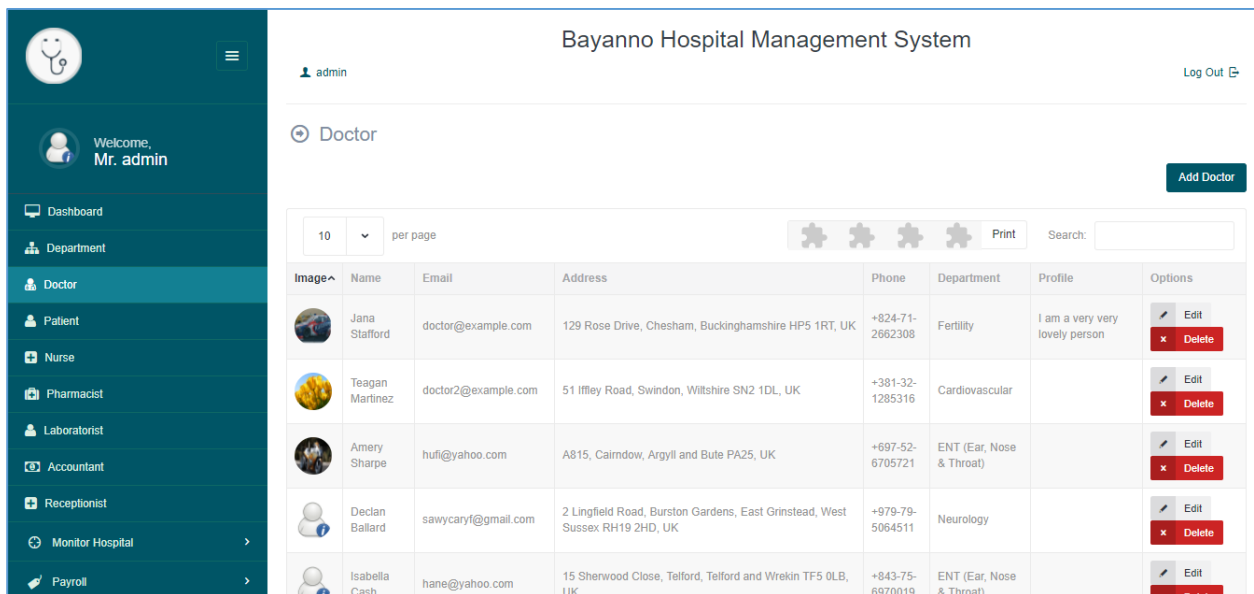
2.7 Outline of Existing Similar Solutions

2.7.1 Bayanno Hospital Management System

Bayanno Hospital Management System is a commercial hospital management system developed by Creativeitem. It is an online system. Some main functions of the system are listed below. [2]

- Appointment scheduling
- SMS notification
- Patient case history
- User management
- Receive payments

Following Figure 2.2 illustrates a screenshot of Bayanno Hospital Management System.



The screenshot displays the Bayanno Hospital Management System interface. On the left is a dark teal sidebar with a navigation menu including: Dashboard, Department, Doctor (selected), Patient, Nurse, Pharmacist, Laboratorist, Accountant, Receptionist, Monitor Hospital, and Payroll. The main content area is titled "Bayanno Hospital Management System" and shows the user "admin" with a "Log Out" link. Below the title, there is a "Doctor" section with an "Add Doctor" button. A table lists doctors with columns for Image, Name, Email, Address, Phone, Department, Profile, and Options. The table contains six rows of doctor data.

Image	Name	Email	Address	Phone	Department	Profile	Options
	Jana Stafford	doctor@example.com	129 Rose Drive, Chesham, Buckinghamshire HP5 1RT, UK	+824-71-2662308	Fertility	I am a very very lovely person	Edit Delete
	Teagan Martinez	doctor2@example.com	51 Iffley Road, Swindon, Wiltshire SN2 1DL, UK	+381-32-1285316	Cardiovascular		Edit Delete
	Amery Sharpe	hufi@yahoo.com	A815, Cairndow, Argyll and Bute PA25, UK	+697-52-6705721	ENT (Ear, Nose & Throat)		Edit Delete
	Declan Ballard	savycaryf@gmail.com	2 Lingfield Road, Burston Gardens, East Grinstead, West Sussex RH19 2HD, UK	+979-79-5064511	Neurology		Edit Delete
	Isabella Cash	hane@yahoo.com	15 Sherwood Close, Telford, Telford and Wrekin TF5 0LB, UK	+843-75-6970019	ENT (Ear, Nose & Throat)		Edit Delete

Figure 2.2 - Bayanno Hospital Management System

2.7.2 Hemas Hospitals



Figure 2.3 - Hemas Hospitals [4]

Hemas Hospitals, is a Sri Lankan chain of hospitals based in Wattala. It holds its prominence as one of the largest private hospitals in the Colombo suburban areas. They use their own eChannelling system for appointment management. [4]

Online payments, searching schedules by date or doctor name or hospital branch and doctor's specialty are main identical functions of this system.

A screenshot of a web form titled "SEARCH FOR YOUR DOCTOR". The form has a light blue background and contains several input fields: "Date" with the value "2017/08/13" and a calendar icon; "Doctor's Name" with the placeholder "Enter First or Last Name"; "Speciality" with a dropdown menu showing "Choose Speciality"; and "Hospital" with three radio button options: "All" (selected), "Wattala", and "Thalawathugoda". A blue "Search" button is located at the bottom of the form.

Figure 2.4 - Screenshot of Hemas Channel Online [4]

These systems was used to understand general functionalities of online appointment management system and get an overall idea about how proposed system should function.

CHAPTER 3 - DESIGN

3.1 Introduction

Systems design is the process of defining elements of the system like architecture, components, interfaces, modules, and data for a system to satisfy specified requirements. [5]

Object oriented design is the process of planning a system of interacting objects for solving a software problem. [10]

Unified Modeling Language is one of the most popular OOD modeling language in the software industry. There are some object models defined by the UML such as class diagram, Use case diagrams and Sequence diagrams that aid software development process.

Following UML diagram model used for system designing,

- Use case diagram
- Sequence diagram
- Entity relationship diagram
- Activity diagram
- Class diagram

3.2 Software Process Models

A software process model is an abstract representation of a software process. These models can be used to explain different approaches to software development. They can be considered as process frameworks that may be extended and adapted to create more specific software engineering processes. [11]

Rational Unified Process was chosen to develop the system over other models such as waterfall model, Rapid Application Development model and Prototype model.

3.2.1 Waterfall Model

The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily like a waterfall through the phases of conception, initiation, analysis, design, construction, testing, implementation and maintenance. [12]

3.2.2 Spiral Model

The spiral model is similar to the incremental model. It is a combination of sequential and prototype model. The spiral model has four phases: Planning, Risk Analysis, Engineering and Evaluation. The baseline spiral, starting in the planning phase, requirements are gathered and risk is assessed. Each subsequent spirals builds on the baseline spiral. [13]

3.2.3 Prototyping Model

The Prototyping Model is a system development method in which a prototype is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved. This model works well when all project requirements are known in detail. [14]

3.2.4 Rapid Application Development (RAD) Model

RAD is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. This can quickly give something to client use and get client's feedback about developed software. [15]

3.3 Methodology for the proposed system

Rational unified process was selected to develop the system. Rational unified process is an adaptable process framework which enable developers to change software development process if user need to make changes. RUP divides the development process into four phases. The four phases are,

- Inception – The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.

- Elaboration – This is where the project architecture and required resources are further evaluated.
- Construction – The project is developed and completed. The software is designed, written, and tested.
- Transition – System is delivered to end users and beta testing carried out. Final adjustments or updates are made based on feedback from end users.

Since Rational unified process provides a specific plan for each step of the development process, it helps prevent resources being wasted and reduces unexpected development costs. [16]

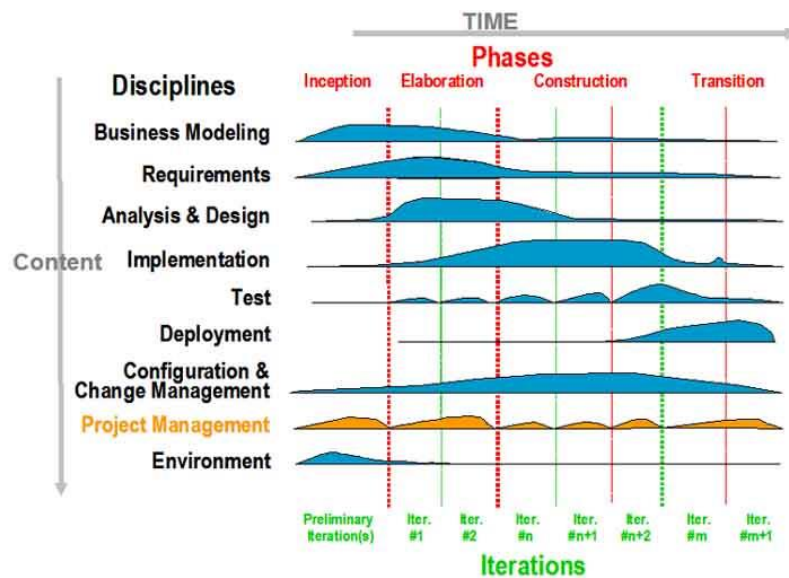


Figure 3.1 - Rational unified process

3.4 Alternative Solutions

There are some alternative solutions available for online appointment management system. They are,

➤ **Standalone system**

A standalone system can be developed for appointment management purpose. But a standalone system is platform dependent. Therefore implement a standalone system

requires separate operating systems, a physical location which will be very expensive. Most users won't be access the system.

➤ Use Off-the-shelf software

Using off-the-shelf software won't satisfy user requirements. User may have to pay additional fee for features user doesn't require.

3.5 Selected Solution

- Mainly client requested for a web based system.
- Web based systems are Platform Independent.
- Patients does not require install additional softwares to use the system.
- Can be accessed by various internet enabled devices.
- Easy to deploy the system.
- MVC architecture makes future enhancements easy to implement.
- Doesn't require a lot of resources to operate.

3.5.1 High level use case diagram for proposed system

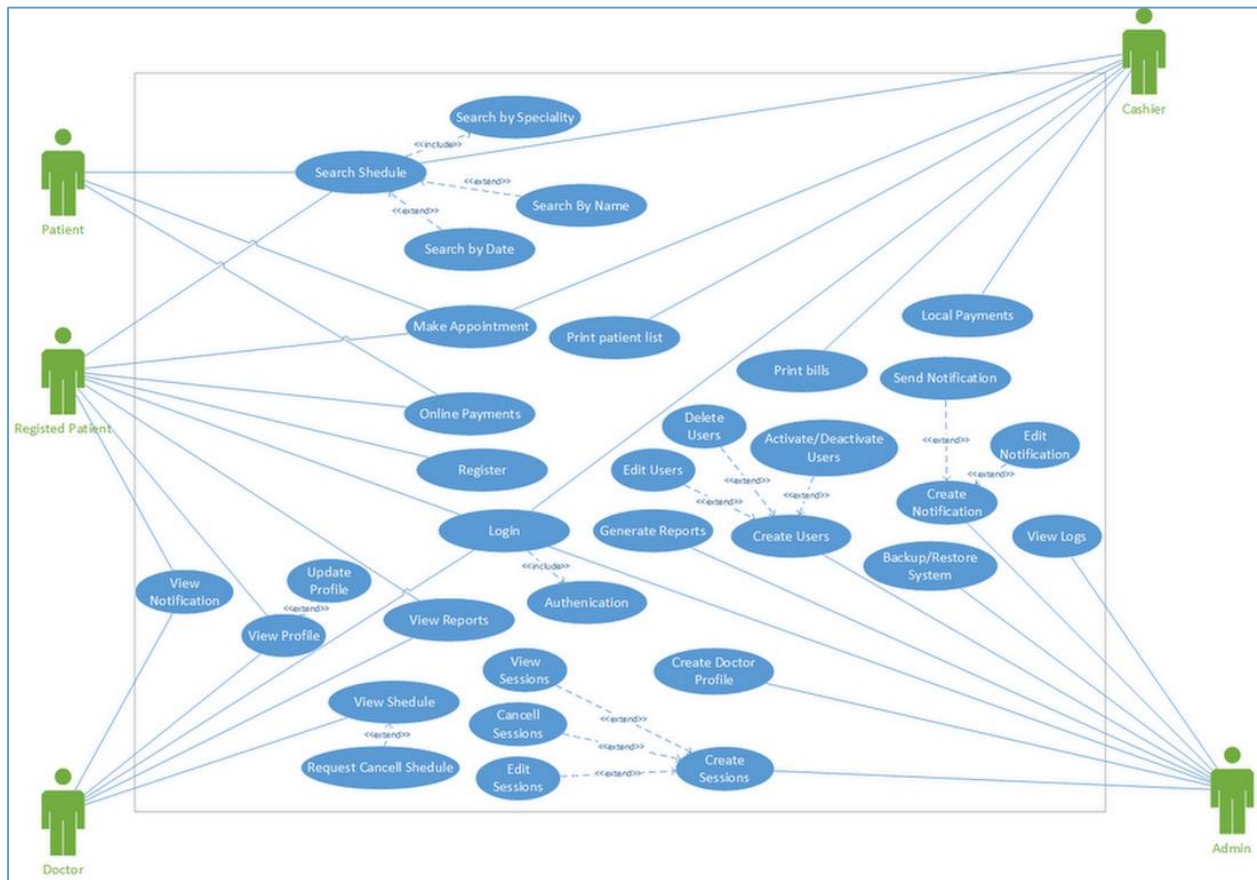


Figure 3.2 - High level use case diagram for proposed system

3.5.2 Activity Diagram for Patient Registration

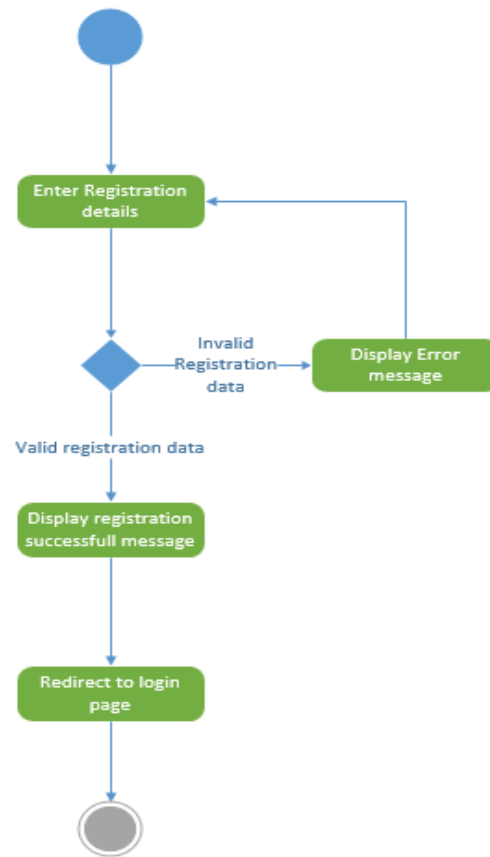


Figure 3.3 - Activity Diagram for Patient Registration

3.5.3 Sequence Diagram for put appointments

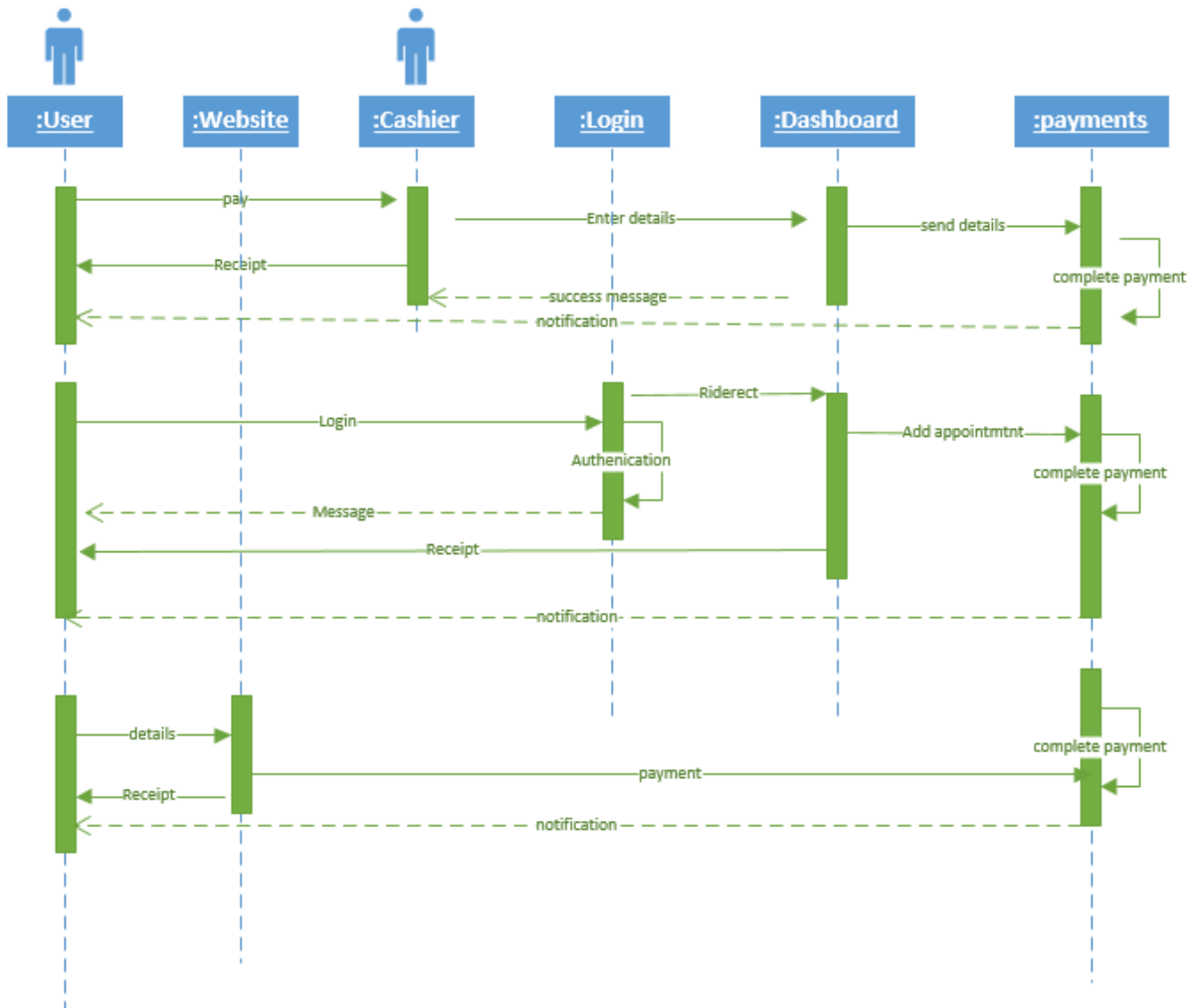


Figure 3.4 - Sequence diagram for put appointments

3.5.4 Class diagram for proposed system

Class diagram describe the structure of the system. They show relationship between classes, attributes, Objects and operations.

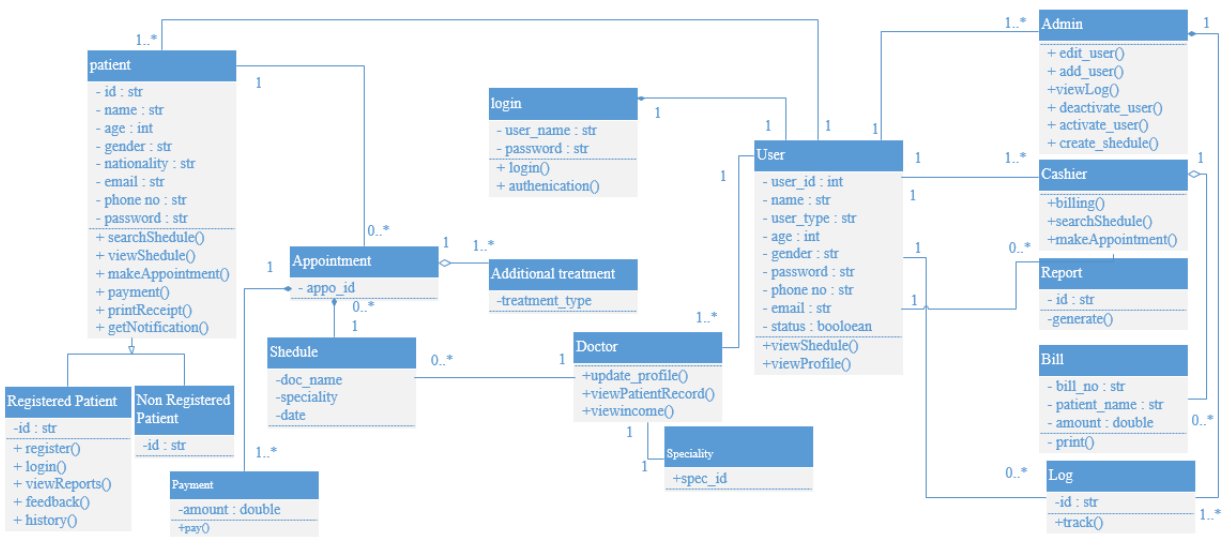


Figure 3.5 - Class diagram for proposed system

3.6 Database designing

Database design is the process of designing a detailed data model of database. This will help us to build a fast optimized, reliable database. [17]

3.6.1 Entity Relationship Diagram for proposed system

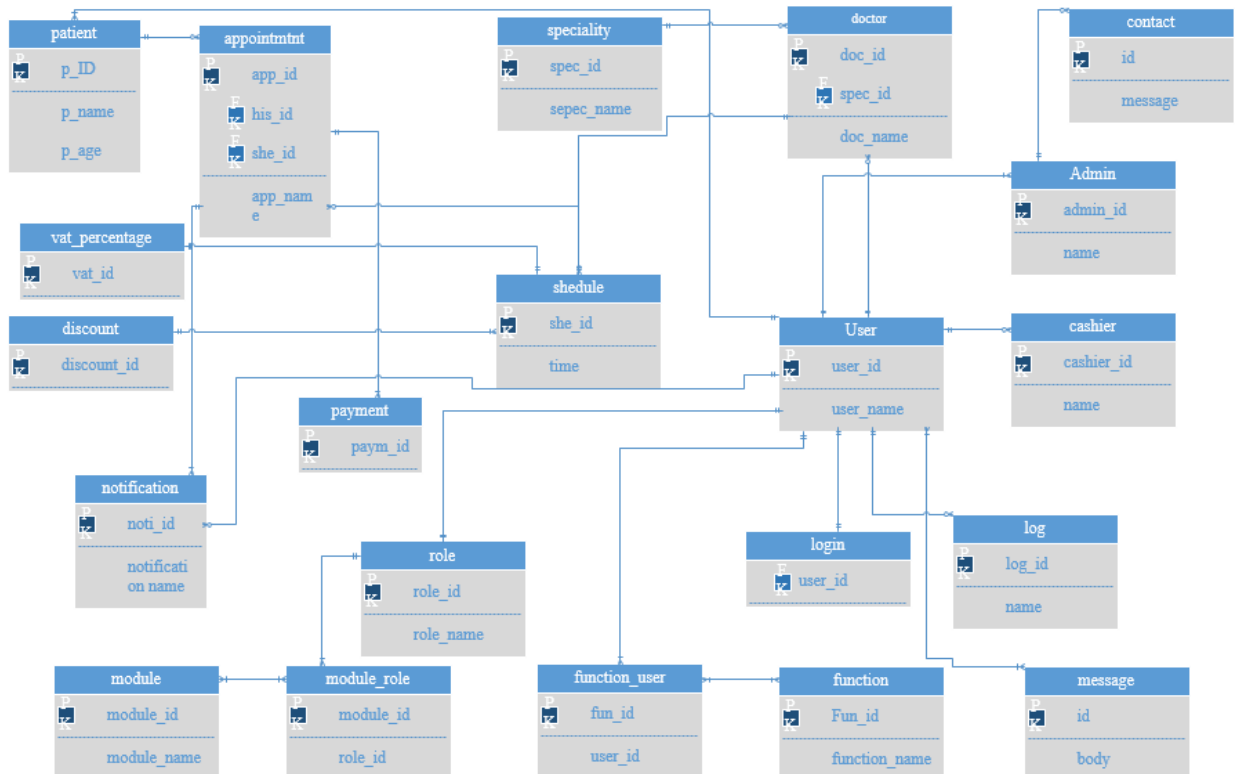


Figure 3.6 - Entity Relationship Diagram

3.6.2 Database Normalization

Database Normalization is a technique of organizing the data in the database. Normalization is a process of organizing the data in database to eliminate data redundancy and undesirable characteristics like Insertion, Update and Deletion Anomalies. [18]

- First Normal Form (1NF)

No two rows of data must contain repeating group of information.

E.g. each set of column must have a unique value. Each table in database should be organized into rows and each row should have a unique primary key.

- Second Normal Form (2NF)

There must not be any partial dependency of any column on the primary key.

- Third Normal Form (3NF)

Every non-prime attribute of the table must be dependent on primary key.

System database is optimized up to Third Normal Form.

3.7 User Interface Design

User Interface is very important to successful implementation of the system. User interface allows users to work with the system without worrying about backend processes or technologies. A nice and clean user interface allows users to work comfortably.

User interface design is the design of user interfaces for machines and software, such as computers, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. [19]

When designing user interface it is important to keep design simple as possible. Simple user interface is easy to understand and helps to get client approval for the system.

When designing user interface, it is necessary to meet client expectations and requirements. Following main interface design rules were followed while designing the system.

- Using icons, buttons and colors consistently. This helps the user to use the system without worrying what each button will do.

- Users can easily navigate the system module wise.
- Using familiar names to user that can be easily understand.
- Design interfaces according to system domain.
- Design system to error prevention and easily handle errors.
Ex : Display error messages on interface or alert boxes
- Enable users to navigate system using shortcuts.
- Design relevant input fields to auto fill. This feature will help to reduce work need to done by user.

Proposed Online Appointment management system consists of two parts. One is external website and other one is internal system. Some main user interfaces are shown below.

3.7.1 Website Interface

Website can be used to provide information about hospital to patients, put appointments online and provide contact facilities with system admin.

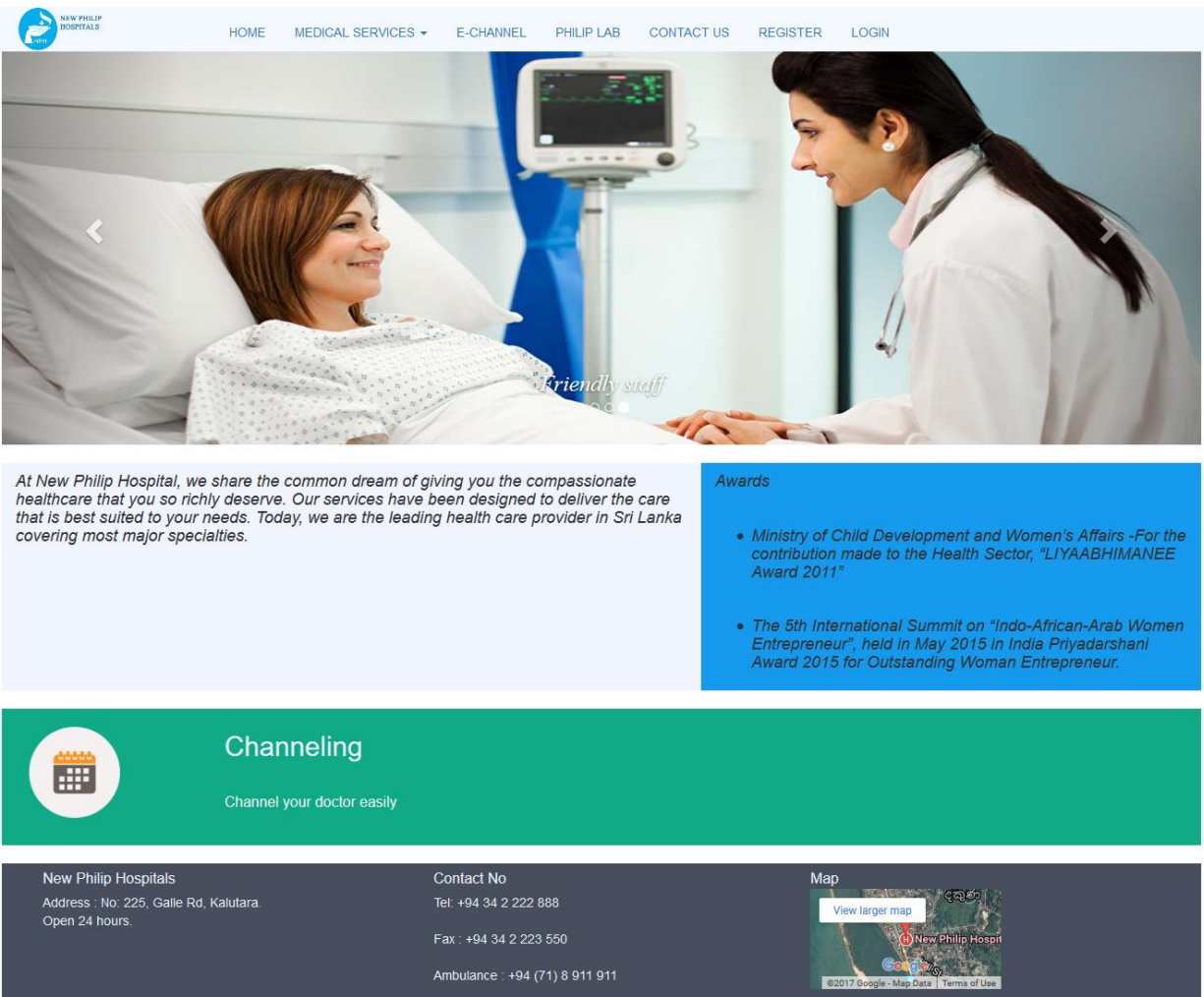


Figure 3.7 - Website Home Page

3.7.2 Login interface

This is the first screen user see when trying to log into the system. User can use their email and password to log into the system. Users can go to registration page or password recovery page by clicking relevant links on this login interface.

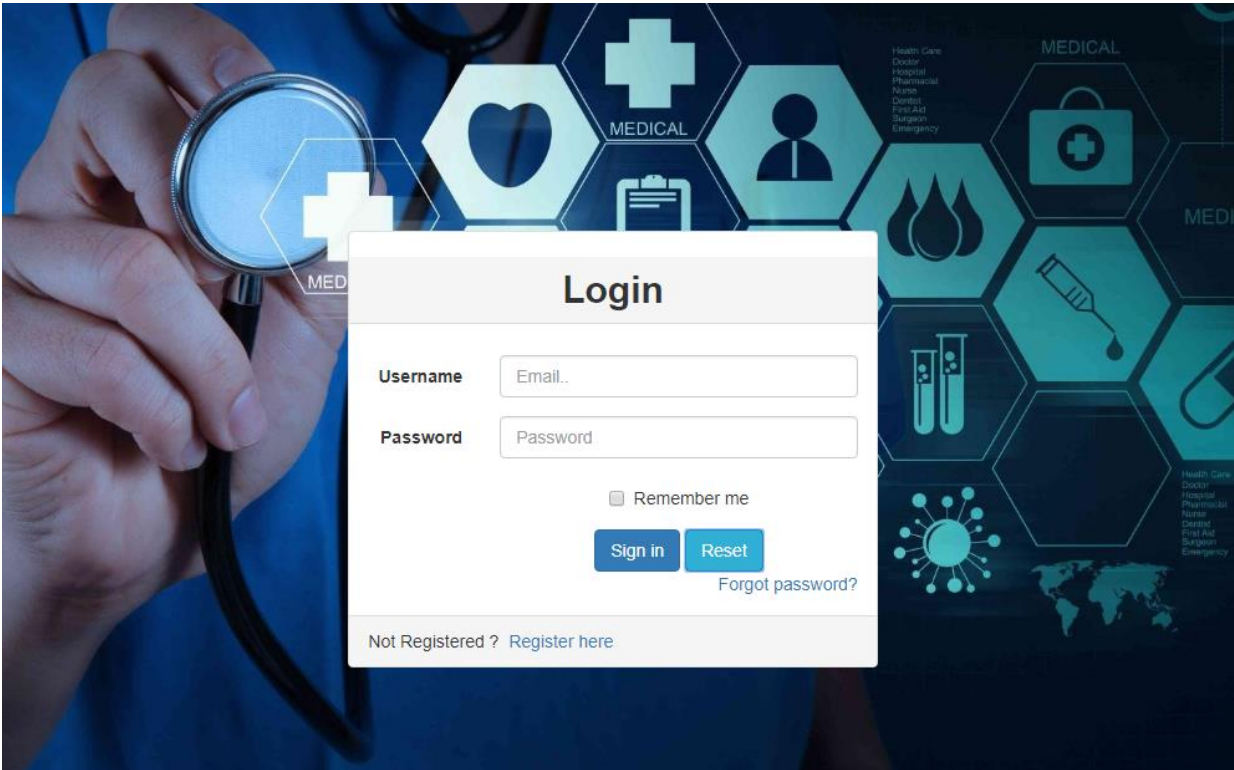


Figure 3.8 - System login Screen

3.7.3 Main Interface

After successfully logging into the system users will redirect to the dashboard following Figure 3.9 illustrates the admin dashboard.

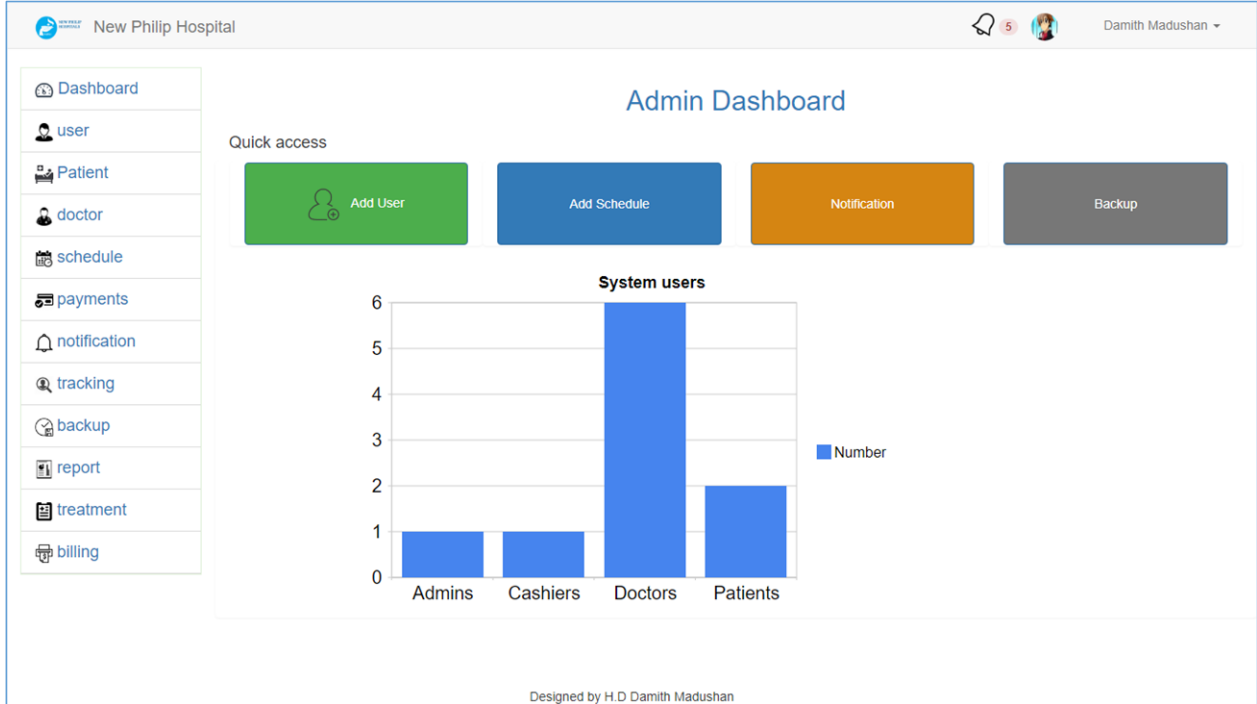


Figure 3.9 - Admin dashboard

3.7.4 User Management Interface

After login to the system admin can add users to the system. Figure 3.10 illustrate the user management interface.

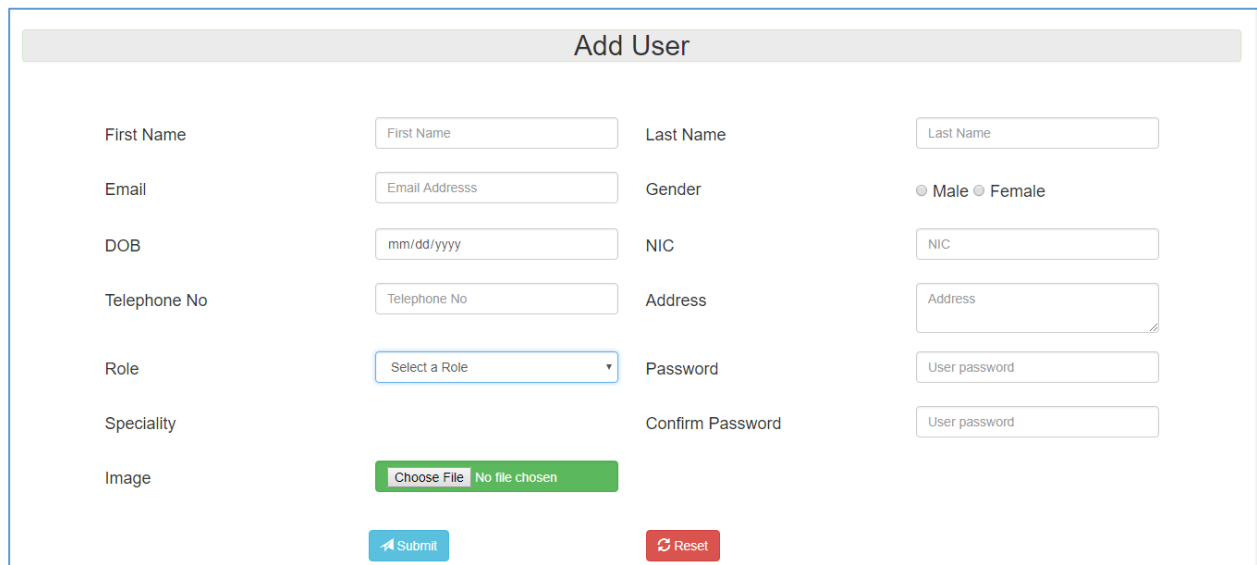
The screenshot displays the 'User Management' interface for 'New Philip Hospital'. At the top right, the user 'Damith Madushan' is logged in. The interface includes a '+ Add User' button, a 'Show 10 entries' dropdown, and several utility buttons (Copy, CSV, Excel, PDF, Print, Print selected, Column visibility). A search bar is located on the right. Below these elements is a table listing five users with their details and an 'Action' button for each.

Image	First Name	Last Name	DOB	Gender	Address	Email	Tel	NIC	Role	Status	Action
	sahan	Fernando	1982-09-28	Male	Panadura	sh@example.com	+94776543458	826545658V	Patient	Active	Action
	Nalin	santha	1985-09-19	Male	Mathugama	nalinoams.com	+94776567324	856787455V	Doctor	Active	Action
	Sahan	Fernando	1988-08-05	Male	Galle 08	sahanoams.com	+94776543458	886545658V	Doctor	Active	Action
	Nisal	Fernando	1985-09-14	Male	Kurunagala	nisaloams.com	+94776567876	855676545V	Doctor	Active	Action
	Priyan	Fonseka	1989-09-11	Male	Panadura	priyanoams.com	+9477656435	894356786V	Doctor	Active	Action

Designed by H.D Damith Madushan

Figure 3.10 - User Management

3.7.5 Add user interface



The 'Add User' form contains the following fields and controls:

- First Name: Text input field
- Last Name: Text input field
- Email: Text input field labeled 'Email Address'
- Gender: Radio buttons for 'Male' and 'Female'
- DOB: Text input field with placeholder 'mm/dd/yyyy'
- NIC: Text input field
- Telephone No: Text input field
- Address: Text input field
- Role: Dropdown menu with 'Select a Role' as the selected option
- Password: Text input field labeled 'User password'
- Speciality: Text input field
- Confirm Password: Text input field labeled 'User password'
- Image: File upload button labeled 'Choose File' with 'No file chosen' text
- Submit: Blue button
- Reset: Red button

Figure 3.11 - Add user interface

3.7.6 Notifications

Following Figure 3.12 illustrates the notification bar.



Figure 3.12 - User Notification

3.7.7 Confirm Messages

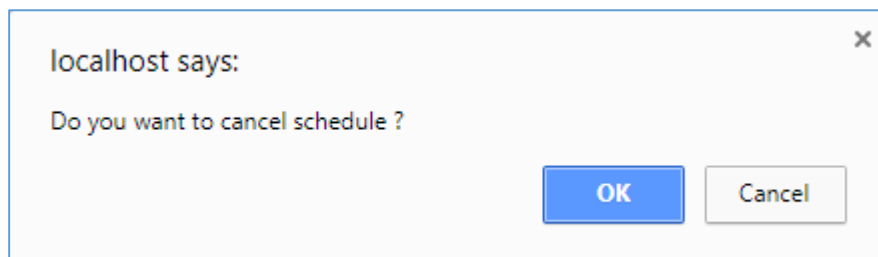


Figure 3.13 - Schedule cancellation confirm message

CHAPTER 4 - IMPLEMENTATION

4.1 Introduction

This chapter is focused on actual implementation process carried out in order to implement the proposed system to real world application. Implementation environment, used tools, plugins, technologies will be discussed through this chapter. Codes were written in an understandable format and commented important sections in order to improve readability, maintainability and make it easy to future development of the system.

4.2 Hardware and software Requirements

Hardware and software requirements for proposed system are listed in following table.

Hardware	Software
<ul style="list-style-type: none">➤ Intel Core i3 processor➤ 4GB RAM➤ 500GB Hard Disc with 200GB free space	<ul style="list-style-type: none">➤ Microsoft Windows 10➤ Software bundle - XAMPP Server-Version 3.2.1<ul style="list-style-type: none">• Web Server - Apache – 2.4.1• Web Scripting - PHP – 5.6.3• phpMyAdmin 4.0.1• Database Management System - MySQL – 5.0.11

Table 4.1 - Hardware and Software Requirements

4.3 Development tools

Net Beans IDE 8.0.1

Net Beans IDE was used as development tool. Net Beans IDE is an open source software. It offers fast development with auto complete feature. It also helps to organize project file structure.

Adobe Photoshop

Adobe Photoshop was used to image editing.

Composer

Composer dependency manager was used for manage and install relevant dependencies for the system. [20]

XAMPP

XAMPP is an open source platform independent web server solution package mainly consisting with Apache HTTP Server, MySQL Database and interpreters for scripts written in PHP and Perl. Following packages were used from XAMPP software bundle,

- Web Server - Apache – 2.4.1
- Web Scripting - PHP – 5.6.3
- phpMyAdmin 4.0.1
- Database Management System - MySQL – 5.0.11

Google Chrome

Google Chrome (Version 61.0.3163.100) and Firefox (Version 47.0.2) were used as browsers while developing the system.

4.4 Languages and Technologies

Following table describes the Languages and Technologies used while developing the system.

Languages and Technologies	Description
PHP	PHP stands for Hypertext Preprocessor. It was the main server side scripting language used to develop the system.
HTML	Hypertext Markup Language (HTML) was used to design the system frontend.
Ajax	AJAX stands for Asynchronous JavaScript And XML. AJAX was used for client side validation and receive information from server without refreshing the web page.
CSS	Cascading style sheets (CSS) was used for style system user interface
JavaScript	JavaScript was used for client side validations and create conformation boxes, warning alerts.
JQuery	JQuery is a widely used JavaScript library for simplify client side scripting.

Table 4.2 - Languages and Technologies

4.5 Reused existing codes

Following modules were used while developing the system.

Bootstrap

Bootstrap is an open source frontend web framework for designing websites and web applications.

Unlike other frameworks bootstrap only focus on frontend development. [21, 22]

JQuery

JQuery is a popular cross platform JavaScript library. [23]

DataTables

DataTables is a plug-in for the jQuery JavaScript library. DataTables generates high flexible tables. [24]

Google charts

Google Charts is an interactive Web service from google that creates graphical charts from user-supplied information.

Pickdate.js

Pickdate.js is a lightweight jQuery date & time input picker. [25]

TimePicki

TimePicki is a light weight jQuery Time Picker plugin. [26]

PayPal PHP SDK

PayPal provide PHP SDK for integrate online payments to PHP based websites. [27]

PHPMailer

PHPMailer is a code library to send emails via PHP code from a web server. It allows send HTML emails and email attachments. [28]

MySQL Database backup script

Database backup php script from GitHub user tazotodua. [29]

TWILIO PHP HELPER LIBRARY

Twilio is a cloud communications platform as a service. Twilio allows software engineers to send text and sms over internet via its web service APIs. Twilio's services are accessed over HTTP protocol. [30, 31]

Figure 4.1 illustrate the how Twilio sms system works.

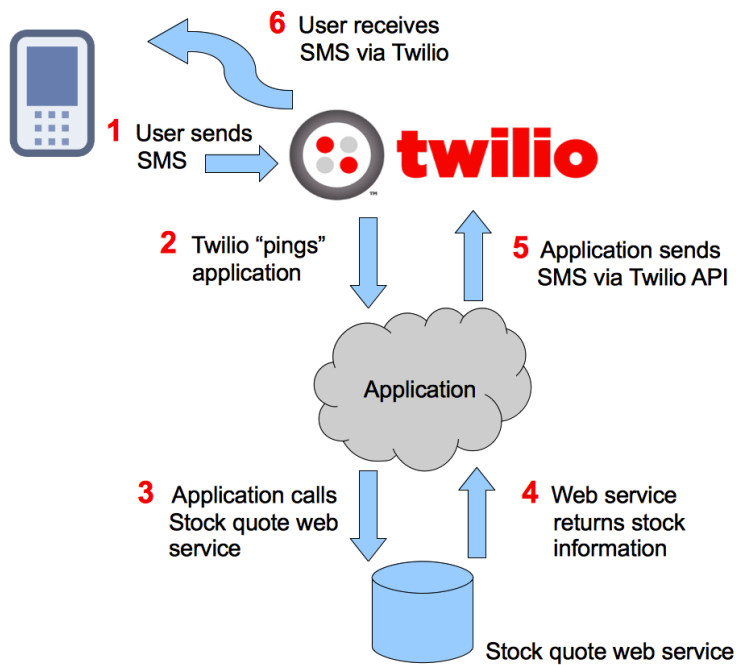


Figure 4.1 - Twilio sms system

4.6 Module Structure

The Online Appointment Management System was built using MVC architecture, which allows programmer to modify and maintain code easily. MVC architecture, mainly has three main logical components. The Model, View and Controller.

Figure 4.2 illustrate the MVC architecture of the system.

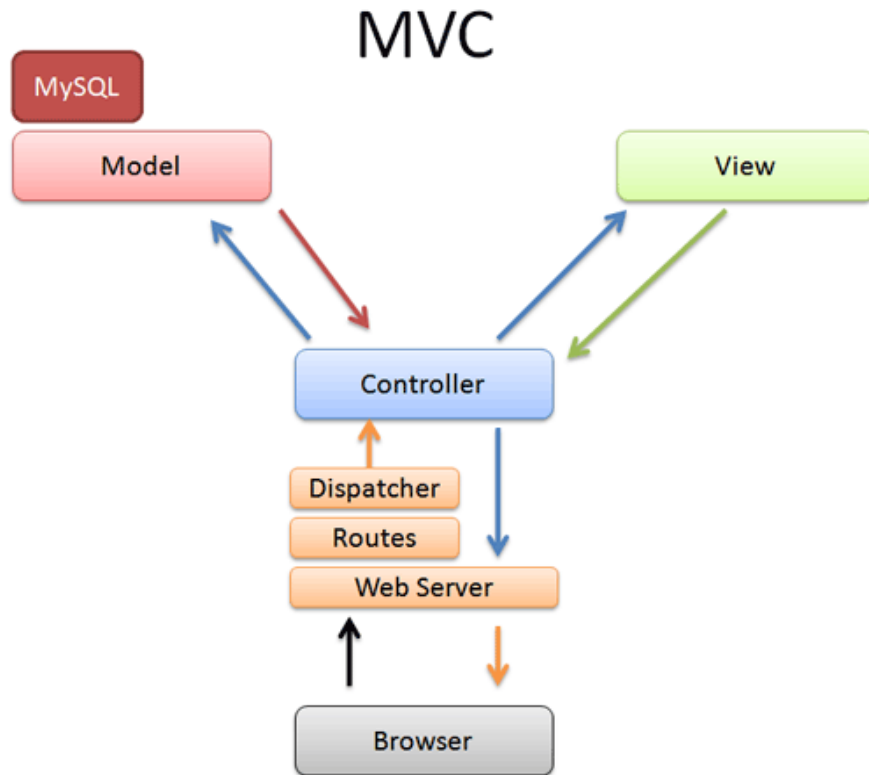


Figure 4.2 - MVC architecture

Model – Model interacts with the database. It retrieves information, manipulates and updates the database in response to controller calls.

View – View is the components user sees on system interface. User interact with the view and perform actions through view.

Controller - Controllers act as an interface between Model and View components to process all incoming requests, manipulate data using the Model and interact with the Views to render the final output.

Figure 4.3 shows directory structure of the system. Folders are organized into MVC architecture and relevant packages. Proper folder structure help to understand the system.

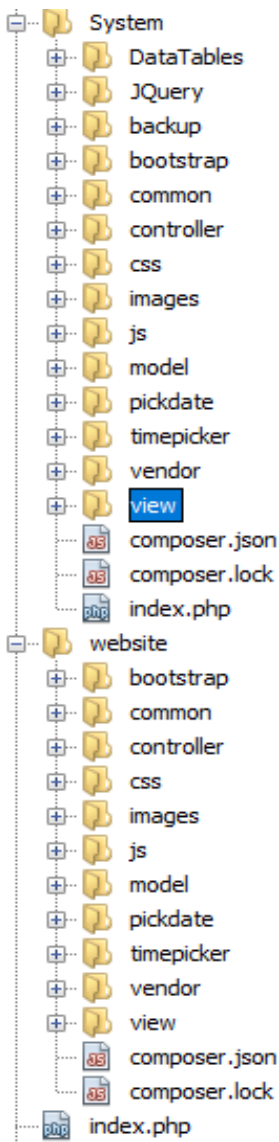


Figure 4.3 - Directory Structure

4.7 Major Code Segments

The main code segments developed has been described below. Comment line have been added to identify the task of relevant code segment.

4.7.1 Database Connection

The system is connected to MySQL database through following code segment.

```
<?php
class dbconnection {
    function connection() {
        $host = "localhost";
        $username = "root";
        $password = "";
        $database = "oams";

        //database connection

        $con = new mysqli($host, $username, $password, $database);

        // Check connection

        if (!$con) {
            die("Connection failed: " . mysqli_connect_error());
        }
        return $con;
    }
}

//database connection object
$con_obj = new dbconnection();
$con = $con_obj->connection();
?>
```

4.7.2 Session Start

User sessions start when the user login into the system. It is very important to handle user sessions correctly.

```
//start session  
  
if(!isset($_SESSION)){ session_start() ; }  
  
date_default_timezone_set("Asia/Colombo");//change user time zone  
  
//create rowuser session to store user information  
  
$rowuser=$_SESSION['rowuser'];
```

4.7.3 Logout and unset sessions

When user logout from the system this code segment will unset all global sessions and automatically redirect user to login page within 5 seconds.

```
unset($_SESSION['rowuser']); //unset user session  
  
unset($_SESSION['rowmodule']); //unset module sessions  
  
header("refresh:5,url=../index.php");//Redirecting
```

4.7.4 Encryption

Following code segment was used for encrypt user password.

```
$password = sha1($user_pass); //using secure hash algorithm 1 to hash user password
```


4.7.5 Client Side Validation

Following code shows Client side validation with JQuery to prevent input errors.

```
$(document).ready(function () {  
  
    $('form').submit(function () {  
  
        var email = $('#email').val();  
  
        var tel = $('#tel').val();  
  
        var pattel = /^\\+94[0-9]{9}$/;  
  
        var patemail = /^[a-zA-Z0-9_\\.\\-]+@((([a-zA-Z0-9\\-]+\\.)+)([a-zA-Z]{2,6})+)$/;  
  
        if (email == "") {  
  
            $('#error_msg').text("Email Address is empty"); //To display error  
  
            $('#error_msg').addClass('alert-danger');  
  
            $('#user_email').focus();  
  
            return false; }  
  
        if (!(email.match(patemail))) { //To check email validity  
  
            $('#error_msg').text("Email Address is invalid"); //To display error  
  
            $('#error_msg').addClass('alert-danger');  
  
            $('#user_email').focus();  
  
            return false; }  
  
        if (tel != "" && !(tel.match(pattel))) { //check telephone number validity  
  
            $('#error_msg').text("Telephone No is invalid");  
  
            $('#error_msg').addClass('alert-danger');  
  
            $('#user_tel').focus();  
  
            return false; }  
  
    });  
  
});
```

4.7.6 Server Side Validation

Following code shows server side email validation for password reset form.

```
$email = $_POST['user_email'];

$code = rand(45345, 87456); //generate code

//set sessions for code and email

$_SESSION['verification_email'] = $email;

$_SESSION['code'] = $code;

//email pattern check

$patern_email = '/^([a-zA-Z0-9_\. -])+\@((([a-zA-Z0-9\ -])+\.)+([a-zA-Z]{2,6})+)$/';

if (preg_match($patern_email, $email)) {

    $password_obj = new password();

    $result = $password_obj->reset($email);

    $no = $result->num_rows; //check existing email address or not

} else {

    //redirect if invalid email

    $msg = "Invalid email !";

    $e_msg = base64_encode($msg);

    header("Location:../view/forgotpassword.php?msg=$e_msg");

}
```

CHAPTER 5 - EVALUATION

5.1 Introduction

System evaluation is a very important phase in software development life circle. At this phase software was tested for identify successes and failures of the system. The testing process is consists of verification and validation. Verification process checks whether system implements specified functions properly. Validation refers to whether system satisfy the client requirements. Evaluation and testing procedures necessary to build a quality system.

5.2 Testing Strategies

A proper software testing should be carried out to verify and validate the system software. The primary purpose of software testing is to detect software failures. These defects may be discovered and corrected while carrying out software testing.

Following test strategies were used for the software testing.

5.2.1 Unit Testing

Individual units are tested to determine if there are any issues. The objective of the unit testing is to isolate each unit of the system to identify, analyze and fix the defects. [32]

5.2.2 Black box Testing

Black box testing, also known as Behavioral Testing. Black box testing is mainly focus on input and output of the software system without bothering about internal knowledge. These tests can be functional or non-functional. [33]

5.2.3 White Box Testing

White box testing is a testing technique that examines the program internal structure. The tester should have excellent knowledge of how software system components work. [34]

5.2.4 Integration Testing

Integration testing is a software development process. Individual units are combined and tested as groups.

5.2.5 System Testing

Complete and integrated software is tested. The purpose is to evaluate the system's compliance with the specified requirements. [35]

5.3 TEST PLAN AND TEST CASES

A test plan is a document describing the scope, approach, objectives, resources, and schedule of a software testing. The test plan document was used to ensure the system meets design specifications and client requirements. A test plan identifies the items to be tested, items not be tested, who will do the testing and the test approach followed.

Test cases were created after designed test plan. A test case include test inputs, execution conditions, and expected results developed for a particular module. In order to reduce test complexity system was divided into modules and each module was tested separately.

Following Tables show the test cases for the system.

Test case for Login

Id	Description	Steps	Expected Results
1	Login into system	Insert username and password	Successfully redirect to system dashboard
2	Try to log into system wrong username and password	Insert invalid username and password	Display error message
3	Login to system with blank username	Only type password	Display error message
4	Login into system with blank password	Only type username	Display error message
5	Remember me	Click on remember me checkbox	Display a alert message to confirm user action
6	Forgot password	Click on forgot password	Redirect to password recovery page
7	Recover password using email address	Enter email address to recover password	Display registered user or not
8	Enter Invalid email address	Type invalid email address on text box	Display error message
9	Click send button with non-existing email	Enter non registered email address and click send button	Display error message
10	Enter existing email	Enter existing email and click send button	Send verification code to email and redirect to password reset page
11	Enter Wrong verification code	Enter wrong verification code	Display error message on submit
12	Enter a short password	Enter a password length is less than 8	Display error message
13	Check whether password is matching	Enter different password on confirm password input box	Display error message

14	Enter correct details on all fields	Enter correct verification code copied from user email and fill all other fields correctly	Reset password and display success message
----	-------------------------------------	--	--

Table 5.1 - Test case for login

Test case for patient registration

Id	Description	Steps	Expected Results
1	Enter all fields correctly	Fill all forum fields	Successfully register and display message
2	Blank user name	Submit without filling username	Display appropriate error message
3	Gender field blank	Leave gender field blank	Display appropriate error message
4	Under age	Submit under age date of birth	Display appropriate error message and popup calendar
5	Existing email	Enter existing email address	Display appropriate error message
6	Invalid email	Enter invalid email address	Display appropriate error message
7	Empty address	Empty address field	Display appropriate error message
8	Invalid NIC	Enter invalid NIC	Display appropriate error message
9	NIC and DOB not matching	Enter not matching values to NIC and DOB	Display appropriate error message
10	Short password	Password length is less than 8 characters	Display appropriate error message
11	Confirm password not matching	Enter not matching confirm password	Display appropriate error message

Table 5.2 - Test case for patient registration

Test case for user management

Id	Description	Steps	Expected Results
1	Enter all fields correctly	Fill all form fields	Successfully register and display message
2	Blank user name	Submit without filling username	Display appropriate error message
3	Gender field blank	Leave gender field blank	Display appropriate error message
4	Under age	Submit under age date of birth	Display appropriate error message and popup calendar
5	Existing email	Enter existing email address	Display appropriate error message
6	Invalid email	Enter invalid email address	Display appropriate error message
7	Empty address	Empty address field	Display appropriate error message
8	Empty role	Submit form without selecting role	Display appropriate error message
9	Empty specialty	Leave specialty field empty if role is doctor	Display appropriate error message
10	Invalid NIC	Enter invalid NIC	Display appropriate error message
11	NIC and DOB not matching	Enter not matching values to NIC and DOB	Display appropriate error message
12	Short password	Password length is less than 8 characters	Display appropriate error message
13	Confirm password not matching	Enter not matching confirm password	Display appropriate error message
14	Activate user	Activate user if user was deactivated	Display success message
15	Deactivate user	Deactivate user	Display success message
16	View user	Click on view user	Redirect to view user page
17	Edit user	Click on edit user	Redirect to edit user page

Table 5.3 - Test case for user management

Test case for schedule management

Id	Description	Steps	Expected Results
1	Add a schedule	Fill all fields correctly	Add schedule and display success message
2	Doctor name is empty	Submit form without select doctor	Display appropriate error message
3	Empty schedule time	Submit form without enter schedule time	Display appropriate error message and popup time select box
4	Hospital name is empty	Submit without selecting hospital name	Display appropriate error message
5	Doctor fee value is empty	Doctor fee value is empty	Display appropriate error message
6	Schedule date is empty	Submit form without selecting date	Display appropriate error message and pop up date picker
7	Vat value is empty	Submit form without selecting vat value	Display appropriate error message
8	Discount value is empty	Submit form without selecting discount value	Display appropriate error message

Table 5.4 - Test case for schedule management

Test case for payment management

Id	Description	Steps	Expected Results
1	Pay for appointment	Search a schedule and then select a schedule and click channel	Redirect to PayPal payment gateway
2	Cancel payment before complete payment	Click on cancel payment link on PayPal page	Redirect to system with appropriate message
3	Complete payment	Complete payment	Redirect to system with appropriate message, print receipt and send notifications

Table 5.5 - Test case for payment management

Test case for notification management

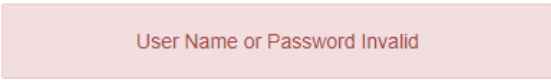
Id	Description	Steps	Expected Results
1	Admin get message through website	External user send a message through contact from on web site	Display new notifications
2	User put an appointment using system	User put an appointment	Send appointment receipt through email and send appointment details to user mobile as a SMS
3	Users and admin send messages within system	A user or admin send a message	Display new notifications



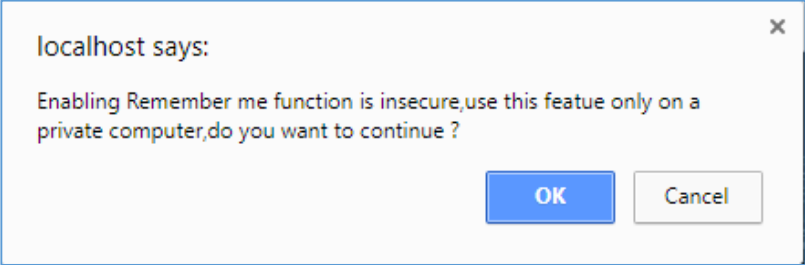
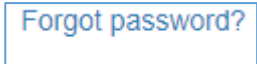
Table 5.6 - Test case for notification management

5.4 Test Results

Test results ensure that system functions was tested properly and system has no defects. Almost every field were tested with sample data and validated functions related to those fields. Some of those results are shown in Table 9 and rest of test results are listed in Appendix chapter.

Test results for Login

No	Description	Actual Output	Status
1	Login to system with valid username and password	Successfully redirect to system	Pass
2	Login to system with wrong username and password		Pass

3	Login into system with a blank username	 Please fill out this field.	Pass
4	Login into system with a blank password	 Please fill out this field.	Pass
5	Check Remember me	 <input checked="" type="checkbox"/> Remember me	Pass
6	Click on Forgot password	Redirect to password recovery page 	Pass
7	Recover password using email address	<p style="text-align: center; color: green;">Registered User !</p> <p>Email <input type="text" value="sh@example.com"/></p>	Pass
8	Enter invalid email address to get verification code	<p style="text-align: center; color: red;">Invalid email</p> <p>Email <input type="text" value="shexample.com"/></p>	Pass
9	Click send button with non-existing email	Email does not exist !	Pass

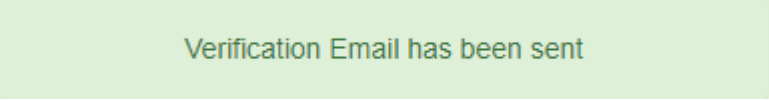
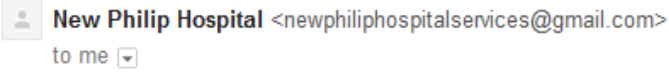
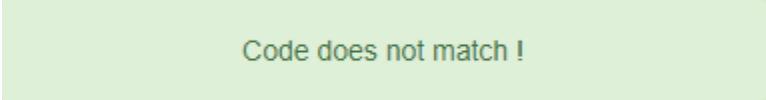
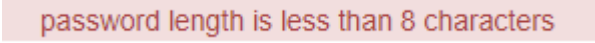


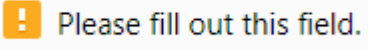
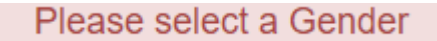
10	Enter existing email and click send button	  Verification Code : 56956	Pass
11	Enter wrong verification code		Pass
12	Enter a short password		Pass
13	Check Whether confirm password is matching		Pass
14	Enter correct details on all fields		Pass

Table 5.7 - Test results for Login

Test Results for patient registration

No	Description	Actual Output	Status
1	Enter all fields correctly	Successfully register	Pass
2	Blank username		Pass
3	Gender field is blank		Pass

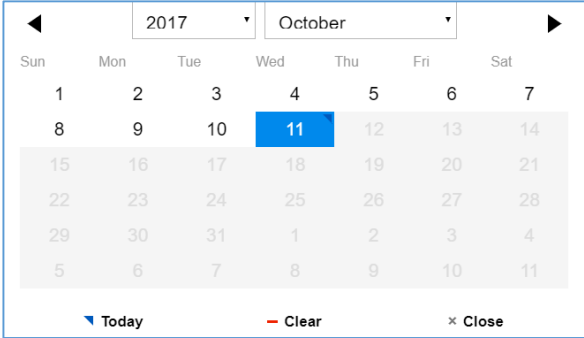
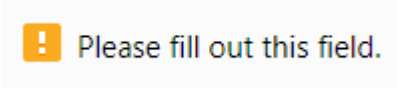
4	Under age		Pass
5	Existing email	Existing Email Address	Pass
6	Invalid email	Invalid email	Pass
7	Empty address		Pass
8	Invalid NIC	NIC is invalid	Pass
9	NIC and DOB are not matching	DOB and NIC are not matching	Pass
10	Short password	password length is less than 8 characters	Pass
11	Confirm password is not matching	The password confirmation field does not match the password field	Pass

Table 5.8 - Test Results for patient registration

5.5 User acceptance testing

User acceptance testing done at the client environment with the help of hospital IT staff. Time to time demonstration was done to the client and completed system was tested with appropriate users. According to results of the acceptance testing and client feedback some minor modifications done to the system.

Following graph shows the summarized results of user acceptance testing.

Question no	Question
Q1	System is useful to users
Q2	User friendliness
Q3	Satisfy with the functionalities if the system
Q4	Satisfy with ability to enter data to forms
Q5	Satisfy with system interface and navigations
Q6	Satisfy with understanding with error messages
Q7	Overall system rating

Table 5.9 - Overall test questions

User rating for questions

Question	User(Admin 1)	User(Admin 2)	User(Cashier)
Q1	8	7	7
Q2	7	8	7
Q3	8	8	6
Q4	7	8	7
Q5	9	8	8
Q6	8	8	8
Q7	8	7	7

Table 5.10 - User ratings for questions

CHAPTER 6 - CONCLUSION

New Philip Hospital is a leading hospital in Sri Lanka. Their vision is “*To be dynamic & HealthCare Provider, thereby achieving superior patient care*” which was proven by their advanced medical care. They are well-known for their high quality health care services. They are using a third party system for appointment management purpose which are costly and lack of required functions. Current system does not provide full control over the system to New Philip Hospital IT staff and since this third party software providing services to outsiders, it has common interface and common features.

New Proposed system was developed to overcome difficulties and inefficiencies of current system. A user friendly and simple interface has been provided to users that helps to easily understand the system and easy navigation through system.

Report management helps to monitor various types of situations and make decisions according to them. New Philip Hospital system admins found that easy to carrying out day to day operations since the system provide detailed reports. System was successfully developed after achieving proposed system requirements and functionalities.

Positive feedback from the client proves that system full filled required functional and nonfunctional requirements and project was a success.

6.1 Problems encountered

One of major problem encounter during the project is entering records to the system. Since a hospital is safeguarding patients’ records it was impossible to collect real patient’s data for the system. After discussing with New Philip Hospital system admin test data were entered into the system without entering real patients’ data. Another problem encountered was initial lack of knowledge regarding development language and tools. Required level of knowledge gained by following online books, tutorials, videos and relevant documentations.

6.2 Critical assessments of the project

There are many paid online appointments management systems available in the market. Most of them provide common features and allow hospitals to use those products after signing an agreement. Therefore developing a well customized and specific system for a hospital allows to get more control over the system and save money spent for paid services. Implemented system has unique features simplify day to day activities.

Registered users can view their medical history and view prescriptions for their appointments. Admins can add users, block users, generate reports, add schedules, cancel schedules, response to user inquires and make decision based on reports generated by system. Cashiers can accepts user payments and print receipts for patients. Unregistered users can put an appointments after providing required data. These functionalities make system very useful to users.

6.3 Lessons Learnt

The knowledge gained by developing the system was very valuable. Is was a great opportunity to apply knowledge gain throughout past three years to develop a practical real world system within given time frame.

Lot of experience gain by working with various development tools. Analytical skills also sharpen by applying knowledge gained throughout past three years into solve real world problems. It was a great experience to work with a real client.

Writing the dissertation was another great experience. It helped to understand how to write a formal report correctly.

6.4 Future Improvements

The system can be improved further in many ways by adding new features and functionalities.

- Obtain a SSL certificate to make transactions through a secure connection.
- Create an android/ios application to put appointments without login or visiting hospital website.
- Improve security by using more advanced cryptography methods protect user records

REFERENCES

- [1] Techopedia.com. (2017). what is a Functional Requirement? - Definition from Techopedia. [Online] Available at: <https://www.techopedia.com/definition/19508/functional-requirement> [Accessed: 02-Aug-2017]
- [2] CodeCanyon. (2017). Bayanno Hospital Management System. [online] Available at: <https://codecanyon.net/item/bayanno-hospital-management-system/5814621>[Accessed 2 Aug. 2017].
- [3] Creativeitem.com. (2017). Login | Bayanno Hospital Management System. [online] Available at: <http://creativeitem.com/demo/bayanno/> [Accessed 2 Aug. 2017]. [Accessed 4 Aug. 2017].
- [4] Doctors.hemashospitals.com. (2017). Hemas Hospitals | Hemas Channel Online. [online] Available at: <https://doctors.hemashospitals.com/channel> [Accessed 5 Aug. 2017].
- [5] Anon, (2017). [online] Available at: <https://www.linkedin.com/pulse/what-system-design-why-important-development-process-fareed> [Accessed 15 Aug. 2017].
- [6] En.wikipedia.org. (2017). Non-functional requirement. [online] Available at: https://en.wikipedia.org/wiki/Non-functional_requirement [Accessed 2 Oct. 2017].
- [7] Usability.gov. (2017). Usability Evaluation Basics | Usability.gov. [Online] Available at: <https://www.usability.gov/what-and-why/usability-evaluation.html>[Accessed 2 Oct. 2017].
- [8] En.wikipedia.org. (2017). Efficiency. [Online] Available at: <https://en.wikipedia.org/wiki/Efficiency> [Accessed 2 Oct. 2017].
- [9] En.wikipedia.org. (2017). Non-functional requirement. [Online] Available at: https://en.wikipedia.org/wiki/Non-functional_requirement [Accessed 2 Oct. 2017].
- [10] En.wikipedia.org. (2017). Object-oriented design. [Online] Available at: https://en.wikipedia.org/wiki/Object-oriented_design [Accessed 2 Oct. 2017].
- [11] Moodle.autolab.uni-pannon.hu. (2017). Chapter 3. Software processes. [Online] Available at:http://moodle.autolab.unipannon.hu/Mecha_tananyag/szoftverfejlesztési_folyamatok_angol/ch03.html#d0e501 [Accessed 2 Oct. 2017].

- [12] En.wikipedia.org. (2017). Waterfall model. [Online]
Available at: https://en.wikipedia.org/wiki/Waterfall_model [Accessed 2 Oct. 2017].
- [13] What is Spiral model- advantages, disadvantages and when to use it? [Online]
Istqbexamcertification.com.
Available at: <http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 2 Oct. 2017].
- [14] SearchCIO. (2017). what is Prototyping Model? - Definition from WhatIs.com. [Online]
Available at: <http://searchcio.techtarget.com/definition/Prototyping-Model> [Accessed 2 Oct. 2017].
- [15] What is RAD model- advantages, disadvantages and when to use it? [Online]
Istqbexamcertification.com. Available at: <http://istqbexamcertification.com/what-is-rad-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 2 Oct. 2017].
- [16] Techterms.com. (2017). RUP (Rational Unified Process) Definition. [Online] Available at:
<https://techterms.com/definition/rup> [Accessed 2 Oct. 2017].
- [17] En.wikipedia.org. (2017). Database design. [Online] Available at:
https://en.wikipedia.org/wiki/Database_design [Accessed 2 Oct. 2017].
- [18] Studytonight.com. (2017). 1NF, 2NF, 3NF and BCNF in Database Normalization | DBMS Tutorial | Study tonight. [Online] Available at: <http://www.studytonight.com/dbms/database-normalization.php> [Accessed 2 Oct. 2017].
- [19] En.wikipedia.org. (2017). User interface design. [online] Available at:
https://en.wikipedia.org/wiki/User_interface_design [Accessed 9 Oct. 2017].
- [20] Getcomposer.org. (2017). Composer. [Online] Available at: <https://getcomposer.org/>
[Accessed 10 Oct. 2017].
- [21] En.wikipedia.org. (2017). Bootstrap (front-end framework). [Online] Available at:
[https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))[Accessed 10 Oct. 2017].
- [22] Mark Otto, a. (2017). Bootstrap. [Online] Getbootstrap.com. Available at:
<http://getbootstrap.com/> [Accessed 10 Oct. 2017].

- [23] JQuery.org, j. (2017). JQuery. [Online] JQuery.com. Available at: <https://jquery.com/>[Accessed 10 Oct. 2017].
- [24] Datatables.net. (2017). DataTables | Table plug-in for jQuery. [Online] Available at: <https://datatables.net/>[Accessed 10 Oct. 2017].
- [25] <http://amsul.ca>, A. (2017). Date picker – pickadate.js. [Online] Amsul.ca. Available at: <http://amsul.ca/pickadate.js/date/> [Accessed 10 Oct. 2017].
- [26] Senthilraj.github.io. (2017). TimePicki - how to use jQuery Time picker. [Online] Available at: <http://senthilraj.github.io/TimePicki/howtouse.html>[Accessed 10 Oct. 2017].
- [27] GitHub. (2017). PayPal/PayPal-PHP-SDK. [Online] Available at: <https://github.com/paypal/PayPal-PHP-SDK> [Accessed 10 Oct. 2017].
- [28] GitHub. (2017). PHPMailer/PHPMailer. [Online] Available at: <https://github.com/PHPMailer/PHPMailer>[Accessed 10 Oct. 2017].
- [29] GitHub. (2017). tazotodua/useful-php-scripts. [Online] Available at: <https://github.com/tazotodua/useful-php-scripts>[Accessed 11 Oct. 2017].
- [30] En.wikipedia.org. (2017). Twilio. [Online] Available at: <https://en.wikipedia.org/wiki/Twilio>[Accessed 10 Oct. 2017].
- [30] Twilio.com. (2017). The Twilio PHP Helper Library. [Online] Available at: <https://www.twilio.com/docs/libraries/php>[Accessed 10 Oct. 2017].
- [32] www.tutorialspoint.com. (2017). Unit Testing. [Online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/unit_testing.htm[Accessed 11 Oct. 2017].
- [33] Software Testing Fundamentals. (2017). Black Box Testing - Software Testing Fundamentals. [Online] Available at: <http://softwaretestingfundamentals.com/black-box-testing/>[Accessed 11 Oct. 2017].
- [34] www.tutorialspoint.com. (2017). White box Testing. [Online] Available at: https://www.tutorialspoint.com/software_testing_dictionary/white_box_testing.htm[Accessed 11 Oct. 2017].

[35] Software Testing Fundamentals. (2017). System Testing - Software Testing Fundamentals. [Online] Available at: <http://softwaretestingfundamentals.com/system-testing/> [Accessed 11 Oct. 2017].

APPENDIX A - SYSTEM DOCUMENTATION

This documentation provides step by step guidance how to install the system properly. This document consists of minimum hardware and software requirements to operate the system.

Table A.1 shows the minimum hardware requirements and table A.2 shows the minimum software requirements for a particular computer in order to run the system.

Minimum Hardware requirements

Hardware	Minimum requirements
Processor	Intel Core i3 processor
RAM	4GB RAM
Hard disk	500GB Hard Disc with 1 GB free space or higher
Printer	Dot-matrix printer or Ink jet printer or Laser printer.
Internet	Minimum 512kbps ADSL connection

Table A.1 - Minimum hardware requirements

Minimum software requirements

Software	Minimum requirements
Operating system	Microsoft windows 10 or higher
Bundle package	XAMPP Server-Version 3.2.1 Web Server - Apache – 2.4.1 Web Scripting - PHP – 5.6.3 Database Management System - MySQL – 5.0.11 phpMyAdmin 4.0.1

Code editor	Net Beans IDE 8.0.1
Image editor	Adobe Photoshop
Web browser	Google Chrome (Version 61.0.3163.100) / Firefox (Version 47.0.2) / Internet Explorer 11 or higher

Table A.2 - Minimum software requirements

Installing System

The installation process has three stages,

1. Required software installations
2. Database installation
3. System installation

Step 1 - Required software installations

Installing XAMPP

- Download XAMPP software bundle from <https://www.apachefriends.org>.
- Click on the downloaded file to install.
- Check “I have read and accept the license terms” check box.
- Set installation path to C:\ drive and install.

Installing browser software

- Download and install the latest version of Mozilla Firefox from <https://www.mozilla.org> or Google chrome from <https://www.google.com/chrome/browser/desktop/index.html>

Step 2 - Database installation

- Open XAMPP control panel in the desktop by double clicking it.
- Press start button in front of apache module and MySQL module.
- Press Admin in MySQL module or type `http://localhost/phpmyadmin` in the browser window and open phpMyAdmin.
- In the phpMyAdmin window click in the field named “Create new database” and type "oams" and press "Create" button. A new database named "oams" will be created.
- Select the 'oams' database from the database list and click on the import button under the database.
- Browse through the supplementary CD's database folder (The path would be `.../Database/oams.sql`) and select oams.sql file.
- Click the "Go" button to import the database.

Step 3 - System installation

- Open “`c:\xampp\htdocs`” path from windows explorer and copy paste the OAMS folder in the given CD-ROM.

Launching the system

- Double click on XAMPP icon on desktop and run Apache web server and MySQL server.
- Open web browser and type “`http://localhost/oams`” to visit website.
Type “`http://localhost/oams/system`” to directly log into the system. User can log into system by providing correct username and password.

APPENDIX B - DESIGN

DOCUMENTATION

Use case diagrams

Use case diagram for payments

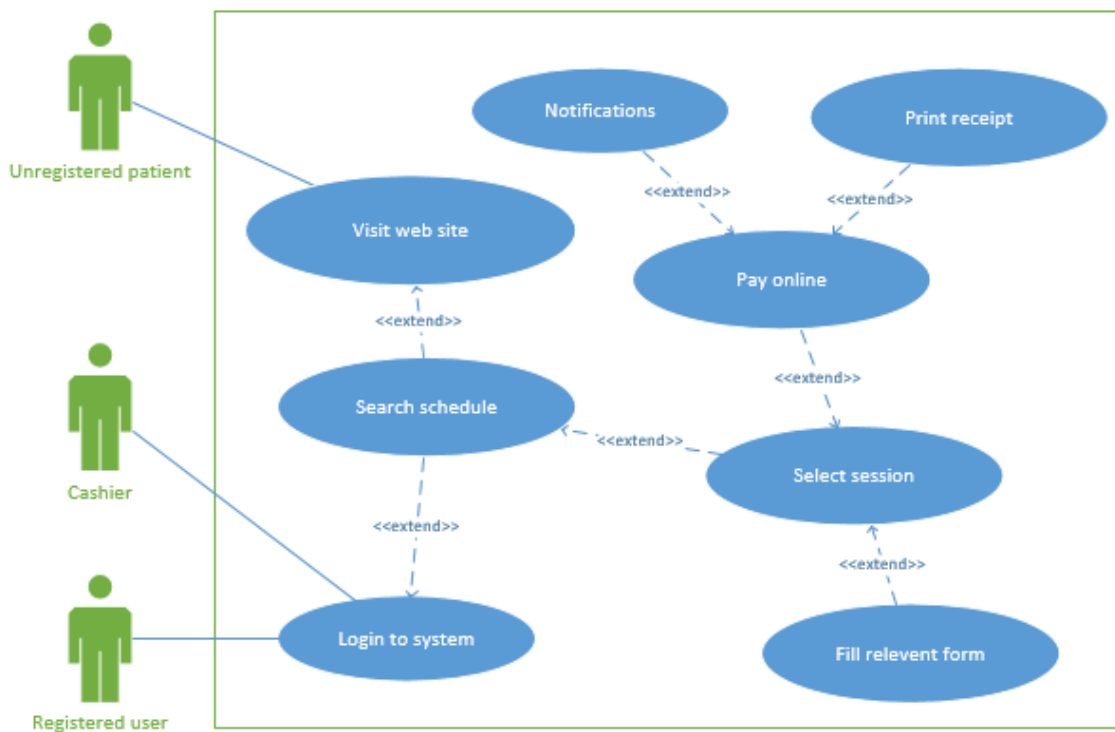


Figure B.1 - Use case diagram for payments

Use case name	Payments
Actors	<ul style="list-style-type: none"> • Registered Patient • Unregistered Patient • Cashier
Preconditions	Visit website or login to system
Overview	Pay for doctor appointment
Flow of events	<ul style="list-style-type: none"> • Patient search for a doctor session and pay online • Cashier search for a doctor session and accept local payments • Cashier or patient may print invoice
Post conditions	Payment data will be stored in database Relevant notifications will be sent

Table B.1 - Payments

Use case diagram for schedule management

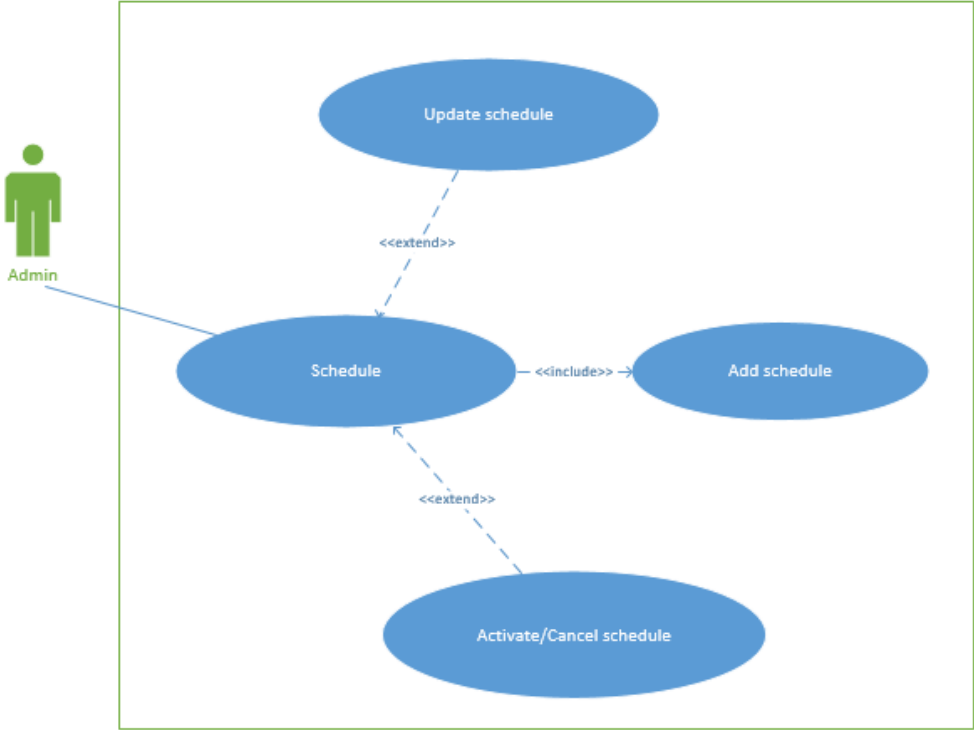


Figure B.2 - Use case diagram for schedule management

Use case name	Schedule management
Actors	<ul style="list-style-type: none"> Admin
Preconditions	Doctor should be available
Overview	Admin create a new schedule
Flow of events	<ul style="list-style-type: none"> Admin click on add schedule Select doctor, session date, session time, add fee and fill other form fields Click submit button and add schedule Cancel or activate schedule upon doctor request and after considering situation
Post conditions	Schedule data will be stored in database

Table B.2 - Schedule management

Use case diagram patient registration

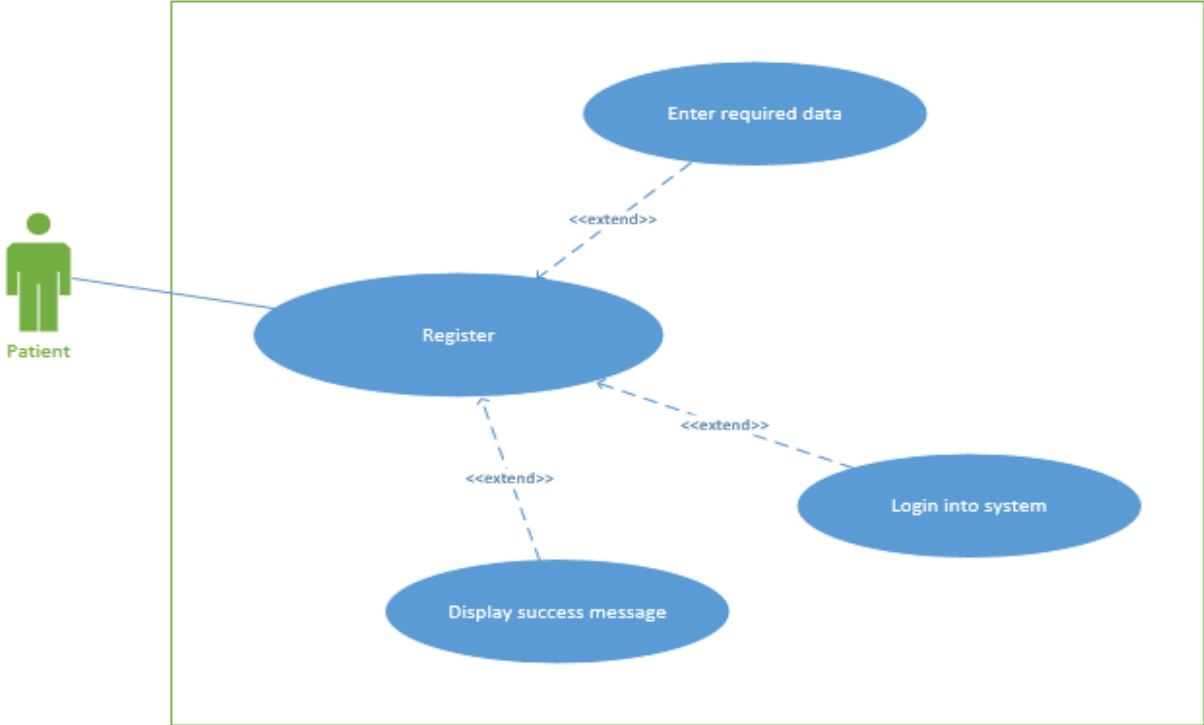


Figure B.3 - Use case diagram patient registration

Use case name	Patient registration
Actors	<ul style="list-style-type: none"> • Patient
Preconditions	
Overview	Patient register through web site
Flow of events	<ul style="list-style-type: none"> • Patient visit hospital website • Fill form on website and submit • Click submit button and add schedule • Patient may login to the system
Post conditions	Patient data will be stored in database

Table B.3 patient registration

Use case diagram for add a user

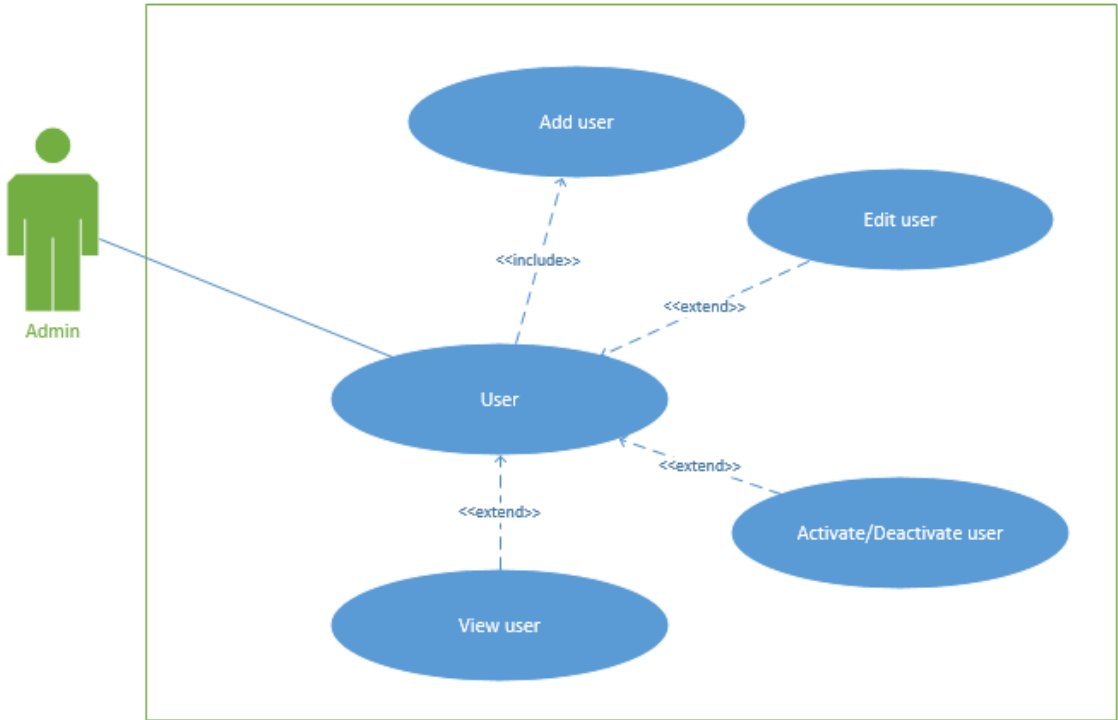


Figure B.4 - Use case diagram for add a user

Use case name	Add a user
Actors	<ul style="list-style-type: none"> • Admin
Preconditions	
Overview	Admin can add users
Flow of events	<ul style="list-style-type: none"> • Admin fill relevant form with relevant data and add user • Admin may edit user • Admin may deactivate/activate user • Admin may view user profile
Post conditions	User data will be stored in database

Table B.4 - Add a user

Sequence diagram for schedule management

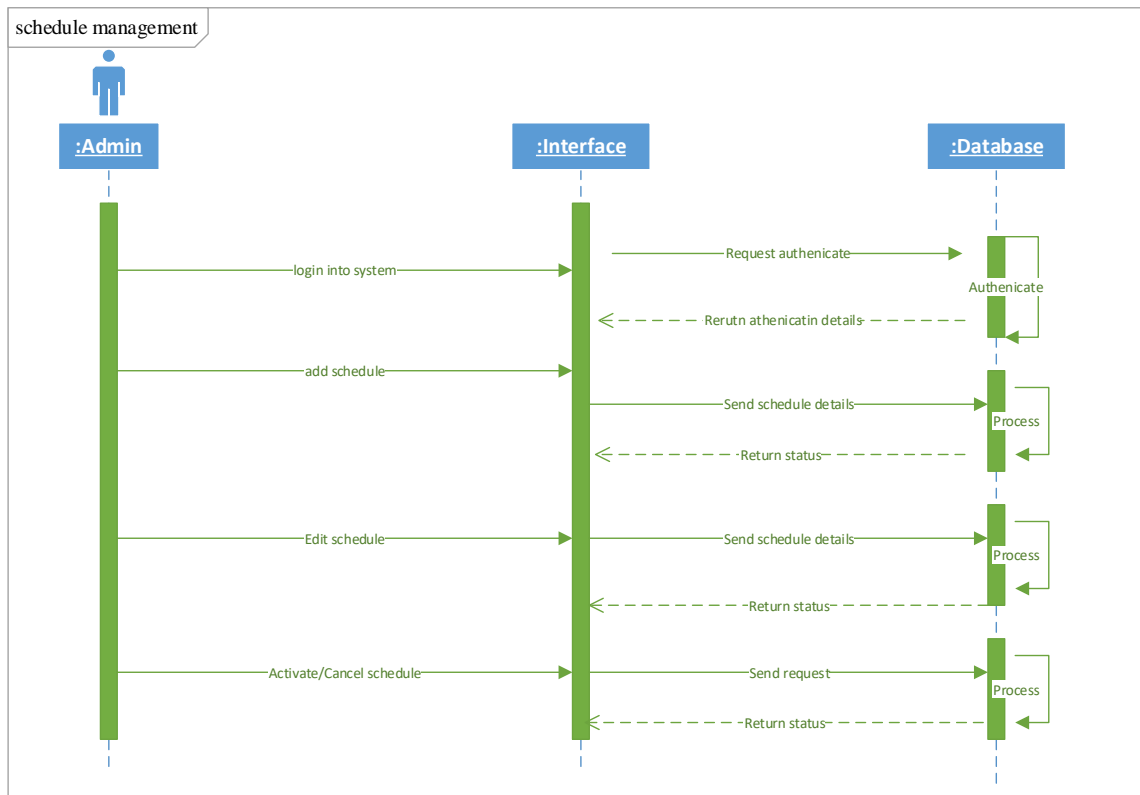


Figure B.5 - Sequence diagram for schedule management

Sequence diagram for report generation

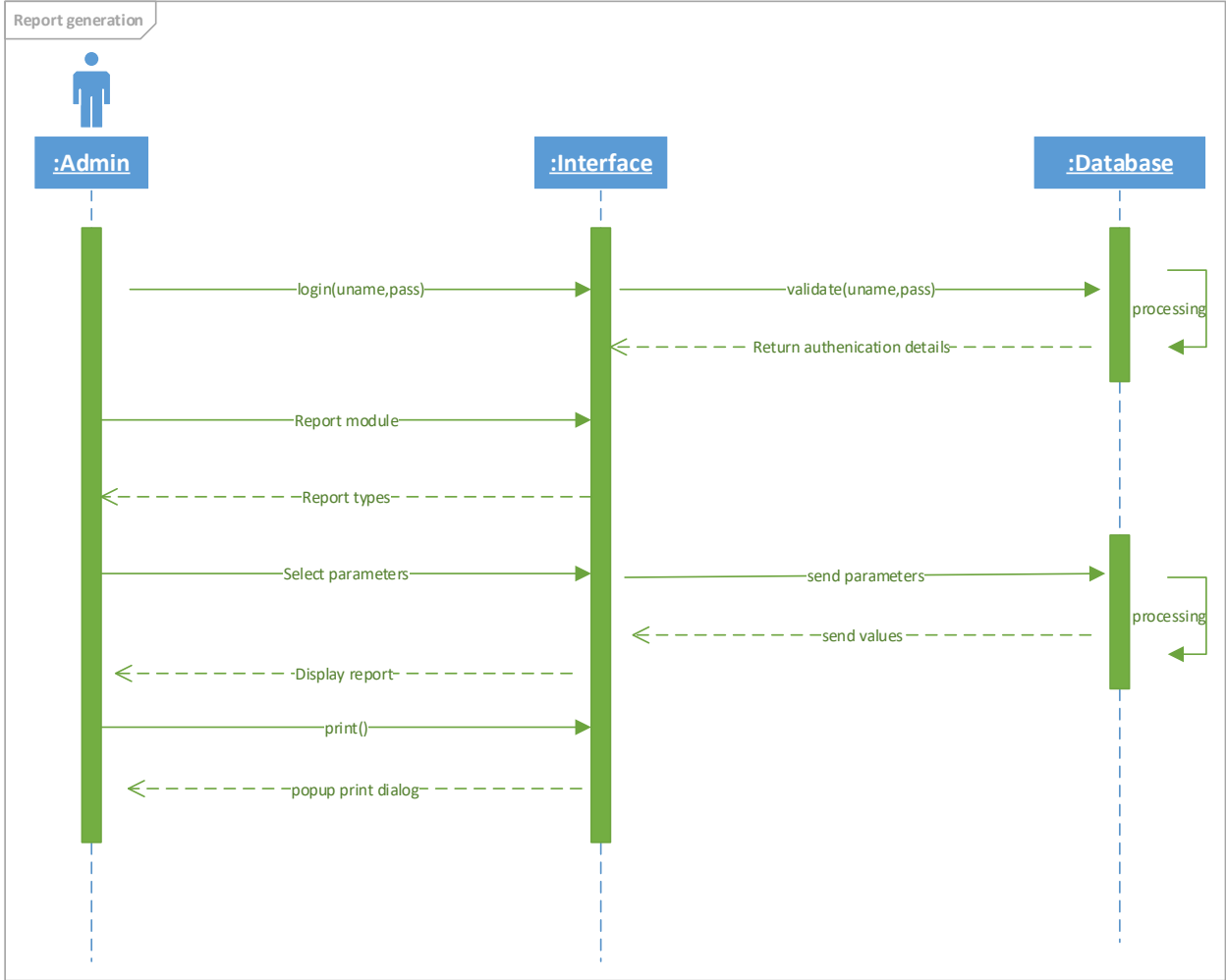


Figure B.6 - Sequence diagram for report generation

Activity diagram for backup module

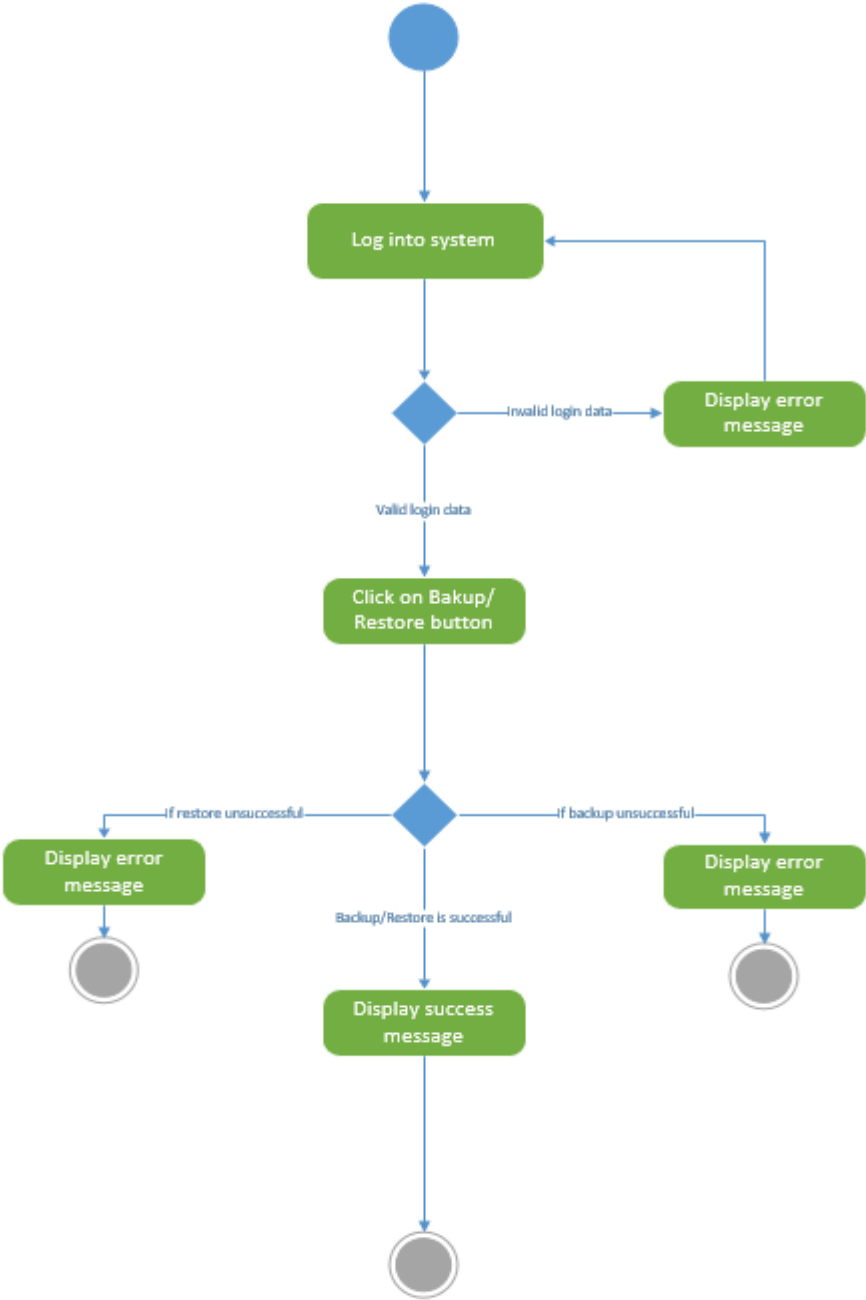


Figure B.7 - Activity diagram for backup module

Following Figure B.8 shows how schedule table interact with user table, doctor table and speciality table.

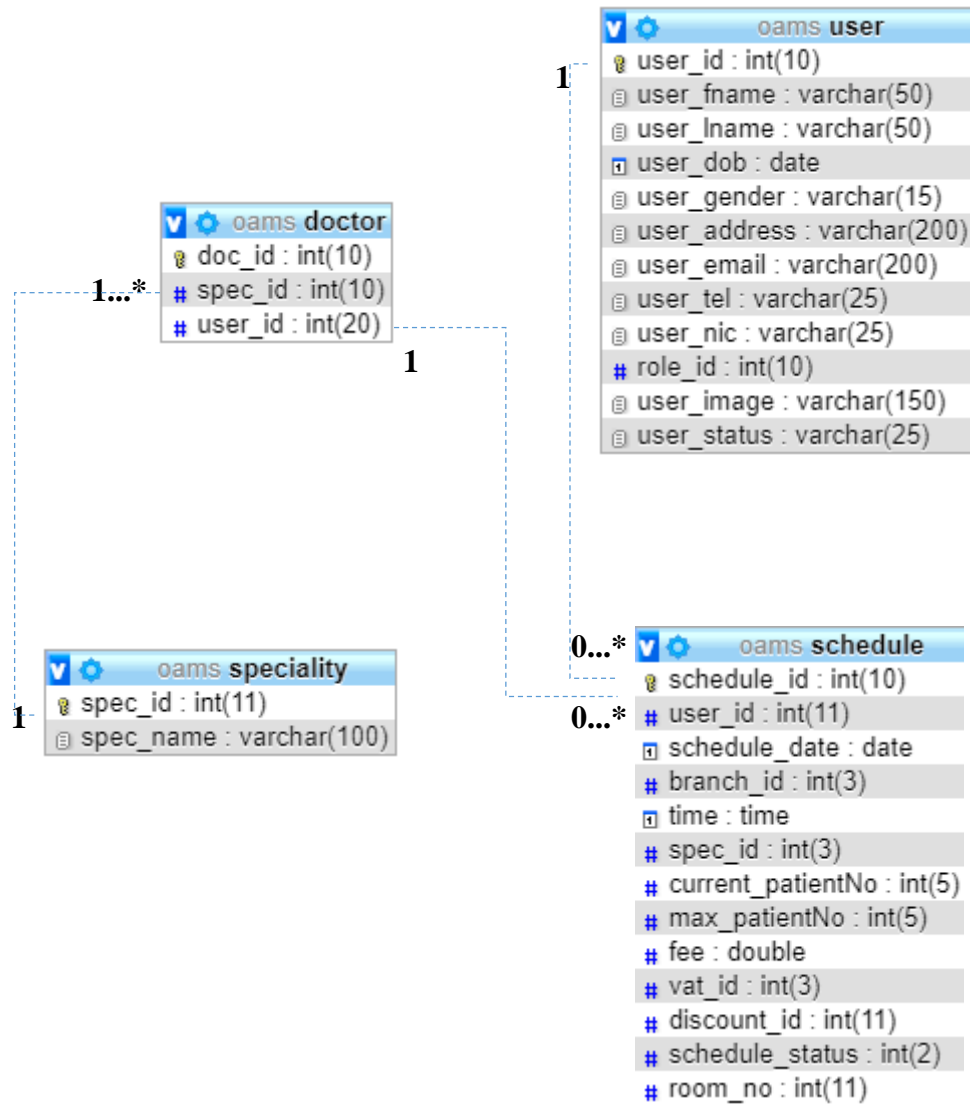


Figure B.8 - Database table structure for few tables

APPENDIX C – USER DOCUMENTATION

User documentation provides guidance to the users those who wish to use the system. To get the maximum use of the system users need to identify all the features and functions of the system. This section contains the main functions of the system and how to use them. System has four user roles. They are Administrator, Cashier, Doctor and patient. Some features are limited to some users according to their privileges.

Login to the system

Users need to enter correct username and password to log into the system. If login is unsuccessful an error message will be displayed on login screen. After entering correct username and password user will be redirect to system dashboard. Figure C.1 illustrates the login screen of the system.

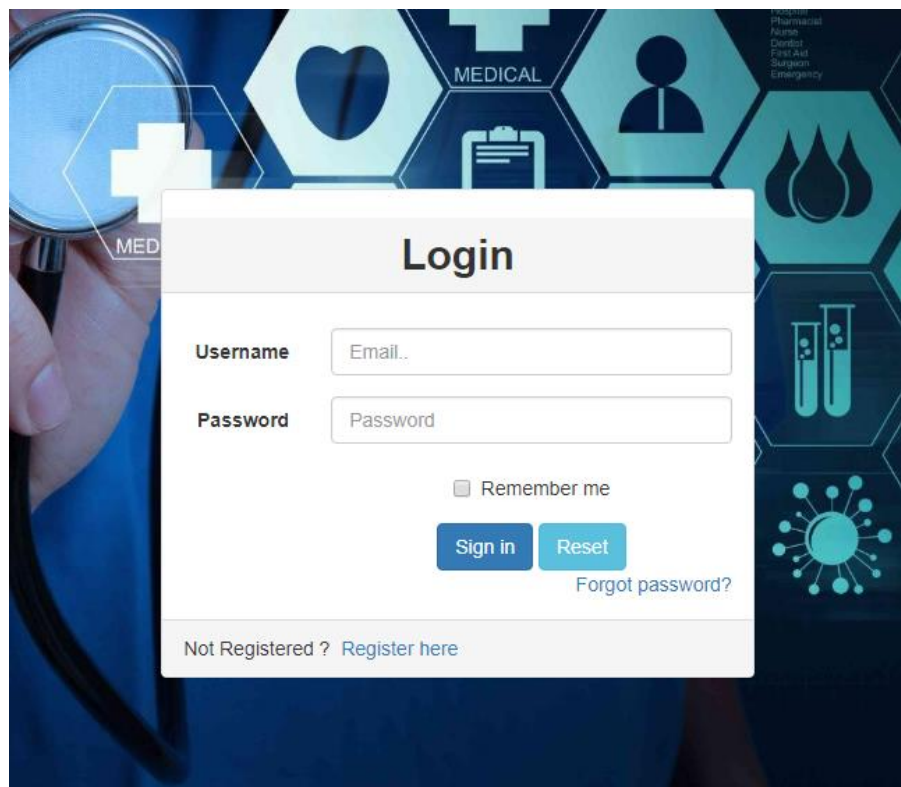


Figure C.1 - Login screen of the system

After successfully login into the system user can see system dashboard and navigation links. Master modules available to admin users only. Figure C.2 illustrates the system dashboard.

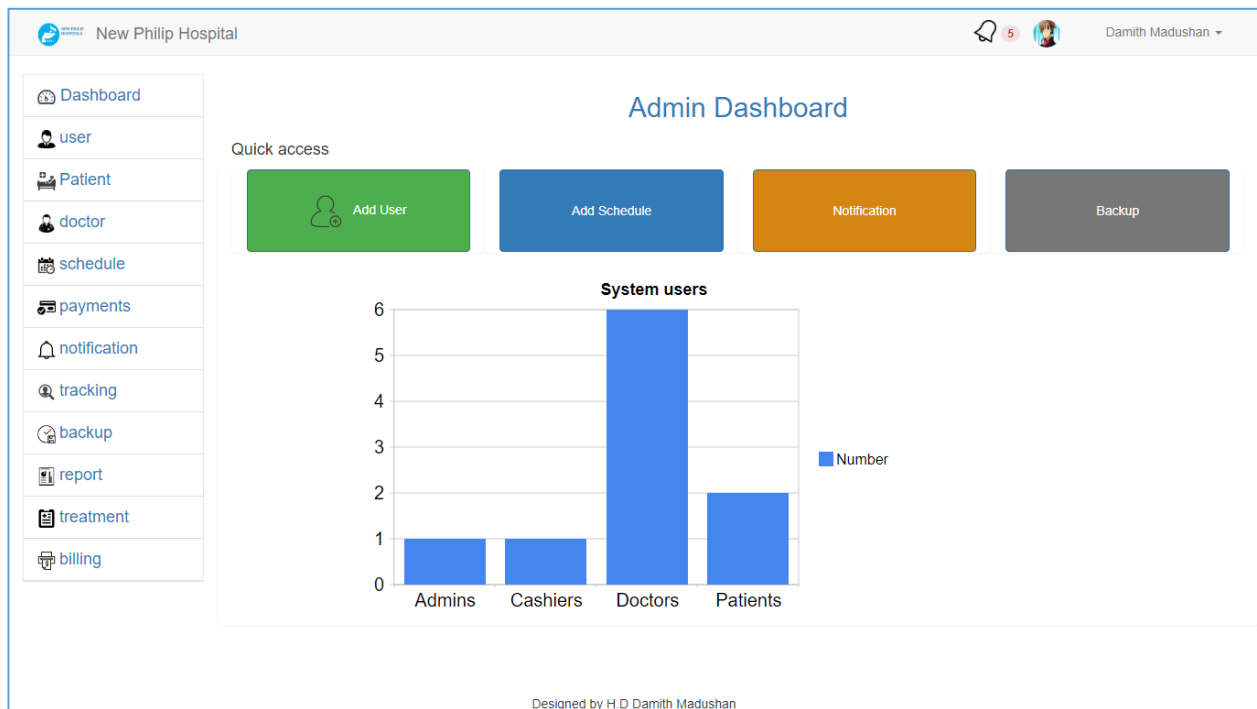


Figure C.2 - System dashboard

- Accessible modules are visible left side of the dashboard.
- Users can logout from the system by clicking on the logout link on top right side of the screen. Users will automatically logout from the system within 5 seconds.

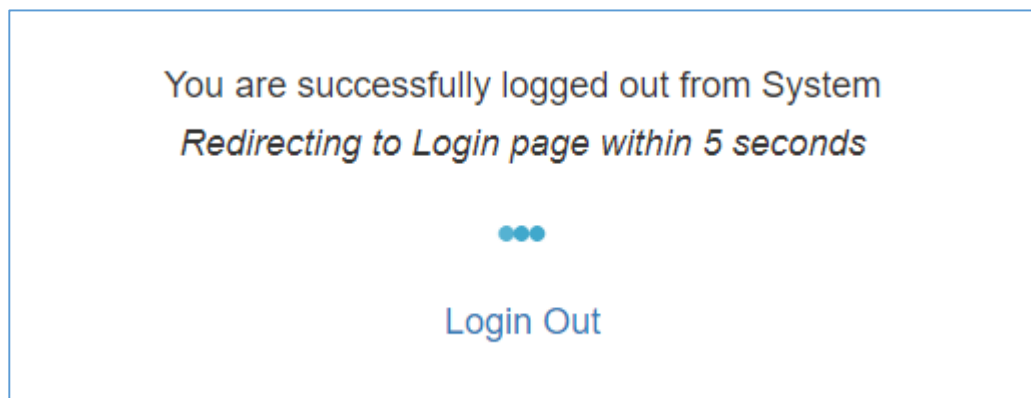


Figure C.3 - Logout screen

- Notifications are shown on top of the screen.



Figure C.4 - Notifications for users

Add a user

Admin can add a user by clicking on the add user button on user management module.

The 'Add User' form contains the following fields and controls:

- First Name: Text input field
- Last Name: Text input field
- Email: Text input field labeled 'Email Address'
- Gender: Radio buttons for 'Male' and 'Female'
- DOB: Text input field labeled 'mm/dd/yyyy'
- NIC: Text input field
- Telephone No: Text input field
- Address: Text input field
- Role: Dropdown menu labeled 'Select a Role'
- Password: Text input field labeled 'User password'
- Confirm Password: Text input field labeled 'User password'
- Image: File upload button labeled 'Choose File' with 'No file chosen' text

Buttons: Submit (blue), Reset (red)

Footer: Designed by H.D Damith Madushan

Figure C.5 - Add user interface

- All fields, are need to fill correctly. If an invalid input is detected an error message will be displayed.

Ex: **Please select a Gender**

Figure C.6 - Verification example

- After selecting a user role relevant functionalities will be displayed. Admin can add or remove functionalities for a user.

Ex:

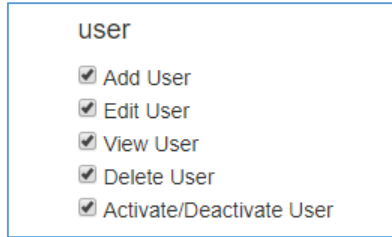


Figure C.7 - User functionalities example

- If selected role was doctor, speciality field will automatically activate.
- User can upload their image when registering. If no image was selected default image will be used as user image.
- Admin can update, view, activate or deactivate user profiles.




	sahan	Fernando	1982-09-28	Male	Panadura	sh@example.com	+94776543458	826545658V	Patient	Active	Action ▾
	Nalin	santha	1985-09-19	Male	Mathugama	nalin@oams.com	+94776567324	856787455V	Doctor	Active	Action ▾
	Sahan	Fernando	1988-08-05	Male	Galle 08	sahan@oams.com	+94776543458	886545658V	Doctor	Active	View Edit Deactive

Figure C.8 - Update, view, activate or deactivate user profiles

Put an appointment

- After successfully login patients can go to payments page and click on add an appointment button.

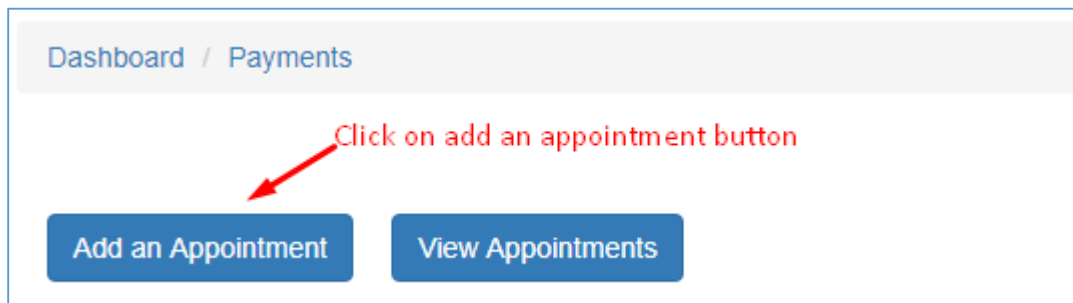


Figure C.9 - Add an appointment

- After clicking on add an appointment button user will redirect to session search page.

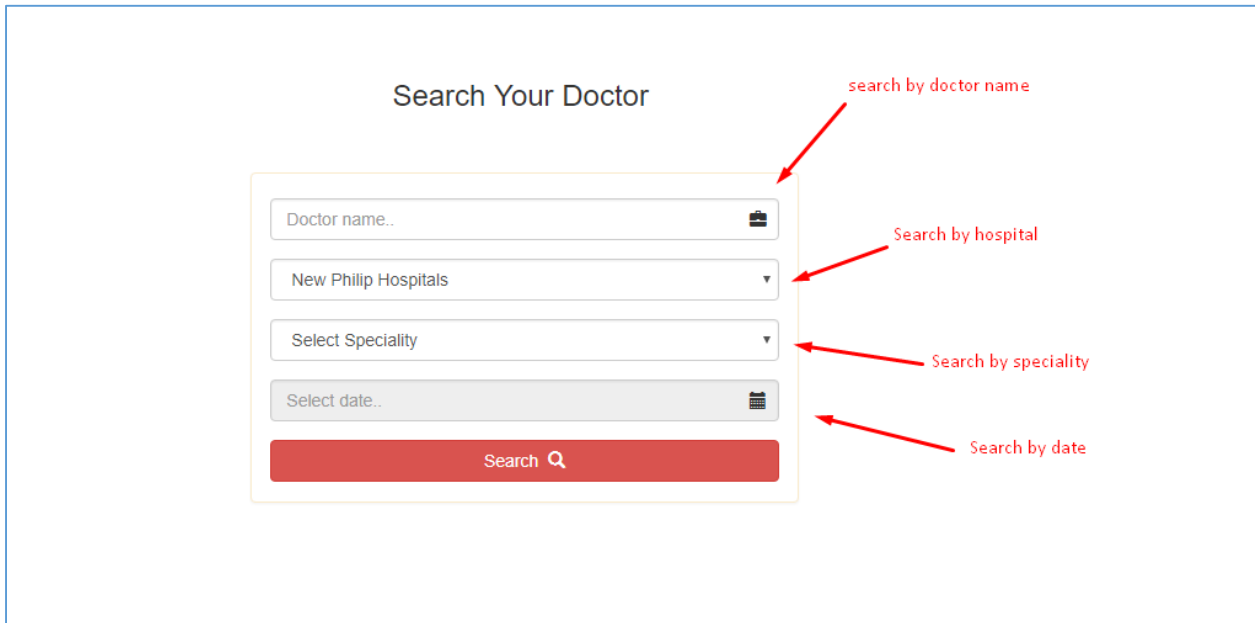


Figure C.10 - Search for an appointment

Users can search by doctor name, Hospital name, speciality, date or combination of those options. Selecting old dated are automatically disabled when search for a doctor session.

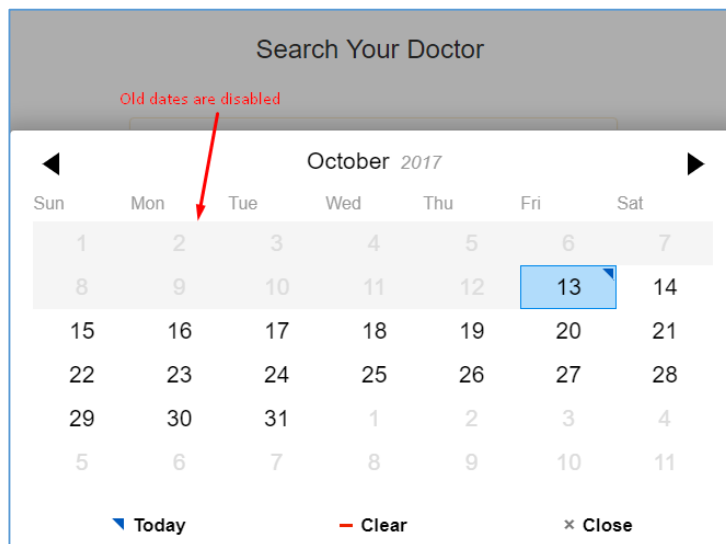


Figure C.11 - Disable old dates

Patient need to fill at least one search field to search for a session. Otherwise error message will be displayed.

Please select a doctor name or a speciality

Figure C.12 - Search error message

After selecting search options and click search user will redirect to channel your doctor page. User can select a doctor form that page.



Channel Your Doctor			
Doctor Name	Schedule date	Speciality	Channel
Dr. Nalin santha	2017-10-13	Neurologist	 Channel

Figure C.13 - Channel your doctor

After selecting a session patient will be redirect to appointment booking page.

Channel Your Doctor



Dr. Nalin santha
Neurologist

Next Number : 1

Session Date : 2017-10-13

Session time : 8:00 AM

Hospital : New Philip Hospitals

Fee : Rs. 1,650.00

Status : Available


 Book

Figure C.14 - Appointment booking

After click on Book button user will be redirect to PayPal payment gateway. Users can confirm payment and pay the relevant fee. Upon successful payment user will be redirect to system and a

success message will be displayed. An appointment notification will be sent to user mobile and copy of receipt will be sent to user email.

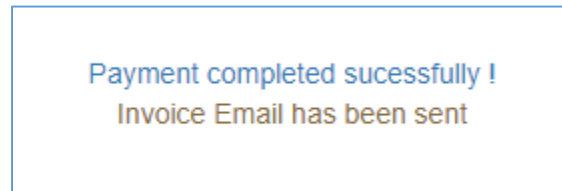


Figure C.15 - Payment completed message

- Patients can view appointments by clicking on View appointments button.

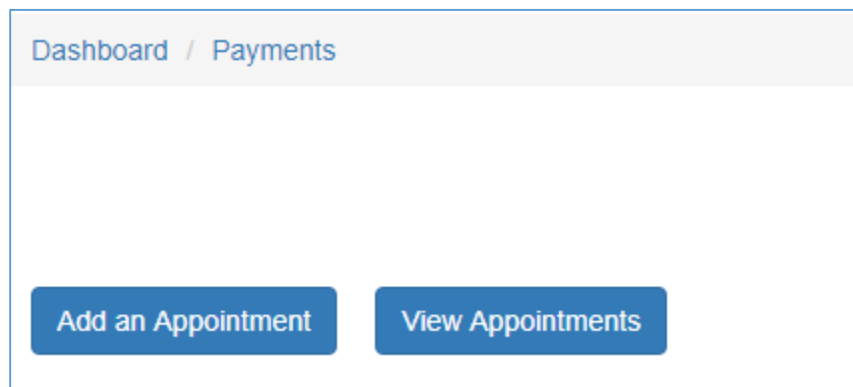


Figure C.16 - View appointments

- Then user will be redirected to view appointments page. Here users can view all appointments made by the user. Users can print receipt for an appointment and view prescription for a particular appointment.

Dashboard / Payments / View Appointments

Show entries Search:

Appointment No	Doctor Name	Speciality	Date	Time	Amount(Rs)	Print	View Prescription
1	Nalin santha	Neurologist	2017-10-13	8:41 AM	1,650.00	Print	View Prescription
1	Sahan Fernando	Neurologist	2017-10-03	9:38 AM	1,200.00	Print	View Prescription
3	Ruwan Fernando	surgeon	2017-10-01	12:08 AM	1,200.00	Print	View Prescription
6	Sahan Fernando	Neurologist	2017-09-29	6:22 PM	1,400.00	Print	View Prescription
1	Nalin santha	Neurologist	2017-09-26	8:30 AM	1,400.00	Print	View Prescription
2	Nalin santha	Neurologist	2017-09-26	5:11 PM	1,400.00	Print	View Prescription
3	Nalin santha	Neurologist	2017-09-26	5:17 PM	1,400.00	Print	View Prescription

Figure C.17 - View appointments page

- Click on print button to print receipt

New Philip Hospital

INVOICE

No: 225, Galle Rd, Kalutara, Sri Lanka.
 Hotline : 034 2 222888 Fax : 034 2 2223550
 24 Hours Service

Invoice NO: NP-4332324

Patient Name : S.H Kapila

Doctor Name : Dr . Nalin santha

Speciality : Neurologist

Appointment Date : 2017-11-08

Room No : 3

Time : 10:30 AM

Appointment No : 1

Doctor Fee : 1500

Vat : 15 %

Discount : 0 %

Total payment : Rs. 1,725.00

Cashier Name : Damith Madushan Invoice Date : November-08-2017 10:31:58 am

Figure C.18 - Invoice for appointment

APPENDIX D - MANAGEMENT REPORTS

The system allows admins to generate well documented reports. The system generates daily and monthly reports which help to make decisions and get overall idea about system status.

Daily income report

Figure D.1 shows daily income report


 NEW PHILIP HOSPITALS		Today's income				
						Date : 2017/10/13
Doctor Name	Speciality	No of patients	Vat Percentage	Discount percentage	Gross Revenue(Rs)	Net Revenue(Rs)
Nalin santha	Neurologist	1	10	0	1,650.00	1,350.00
Nimal perera	Cardiologist	1	10	0	1,540.00	1,260.00
Nisal Fernando	Cardiologist	2	10	0	3,300.00	2,700.00
Ruwan Fernando	surgeon	1	10	0	1,760.00	1,440.00
Sahan Fernando	Neurologist	1	10	0	1,540.00	1,260.00
Total					9,790.00	8,010.00

Figure D.1 - Daily income report

Figure D.2 shows types of users and how many users are there for each type.

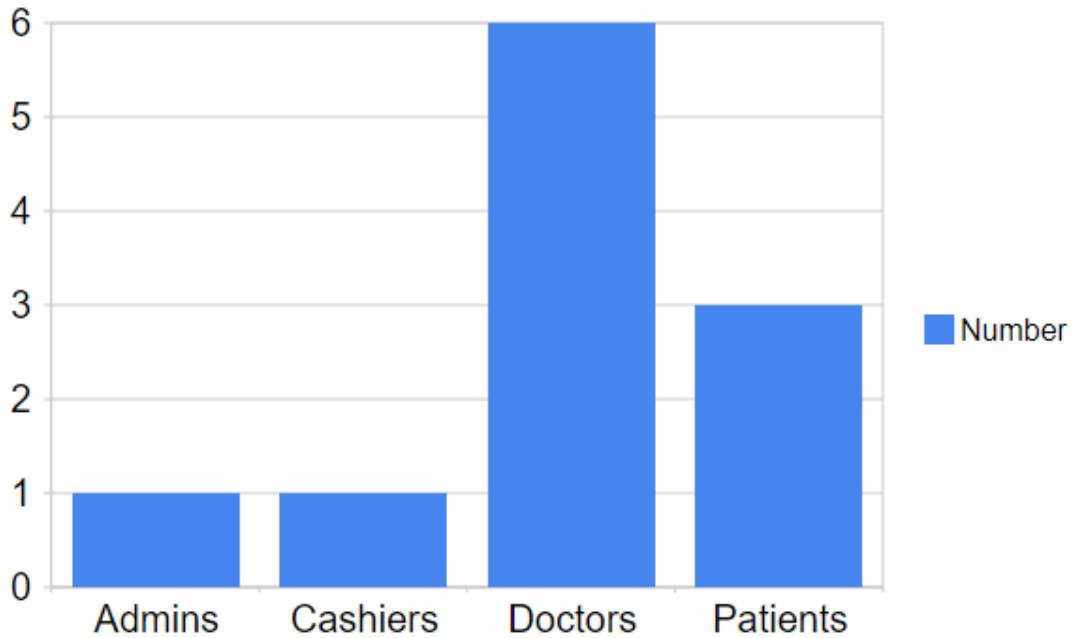


Figure D.2 - Report for system users by role

Figure D.3 shows generating detailed overall user report in pdf format

User Report									
First Name	Last Name	DOB	Gender	Address	Emal	Tel	NIC	Role	Status
Damith	Madushan	1990-08-14	male	kalutara	sinhalasms@gmail.com	+94770387494	906546545V	Admin	Active
Kamal	dayan	1991-09-04	male	colombo 4	kamal@oams.com	+94777777435	915674325V	Cashier	Active
Nalin	santha	1985-09-19	Male	Mathugama	nalin@oams.com	+94776567324	856787455V	Doctor	Active
Nimal	perera	1990-08-29	Male	Colombo 05	nimal@oams.com	+94767687678	906787455V	Doctor	Active
Nisal	Fernando	1985-09-14	Male	Kurunagala	nisal@oams.com	+94776567876	855676545V	Doctor	Active
Nisan	Harsha	1984-10-11	Male	Rathnapura	nisan@oams.com	+94770387494	846545658V	Patient	Active
Priyal	Kumara	1989-09-13	male	katukurunda	prial@oams.com	+94777777675	897546545V	Patient	Active
Priyan	Fonseka	1989-09-11	Male	Panadura	priyan@oams.com	+94777656435	894356786V	Doctor	Active
Ruwan	Fernando	1990-09-04	Male	panadura	ruwan@oams.com	+94777777324	906546546V	Doctor	Active
sahan	Fernando	1982-09-28	Male	Panadura	sh@example.com	+94776543458	826545658V	Patient	Active
Sahan	Fernando	1988-08-05	Male	Galle 08	sahan@oams.com	+94776543458	886545658V	Doctor	Active

Figure D.3 - Detailed overall user report

Figure D.4 shows income report within last 7 days

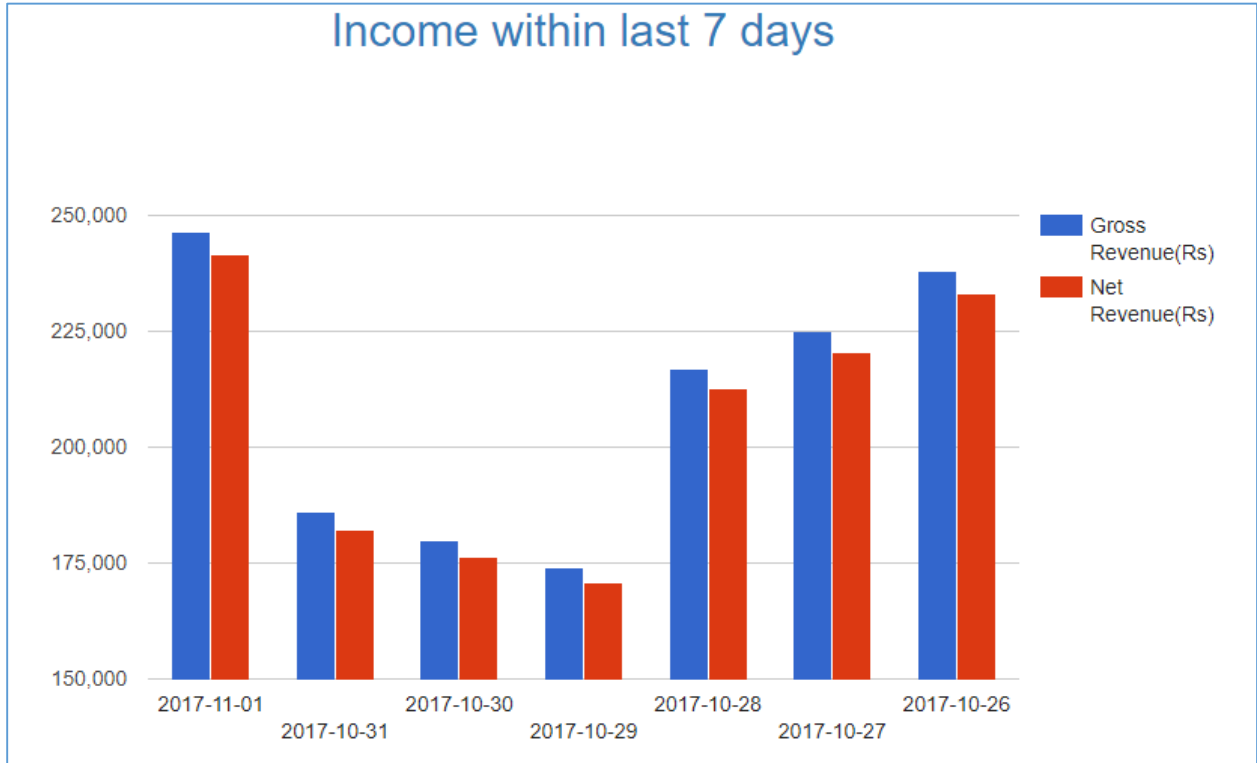





Figure D.4 - Income within last 7 days

APPENDIX E – TEST RESULTS

Table E.1 shows test results for user management

No	Description	Actual Output	Status
1	Enter all fields correctly	Successfully register	Pass
2	Blank username	 Please fill out this field.	Pass
3	Gender field is blank	Please select a Gender	Pass
4	Under age	Under Age	Pass
5	Existing email	Existing Email Address	Pass
6	Invalid email	Invalid email	Pass
7	Empty address	 Please fill out this field.	Pass
8	Empty role	Role Name is empty	Pass
9	Empty speciality	 Please select an item in the list.	Pass
10	Invalid NIC	NIC is invalid	Pass
11	NIC and DOB are not matching	DOB and NIC are not matching	Pass
12	Short password	password length is less than 8 characters	Pass

13	Confirm password is not matching		Pass
14	Activate user		Pass
15	Deactivate user		Pass
16	View user		Pass
17	Edit user		Pass

Table E.1 - Test results for user management

Table E.2 shows test results for notification management

No	Description	Actual Output	Status
1	Admin gets message through website		Pass

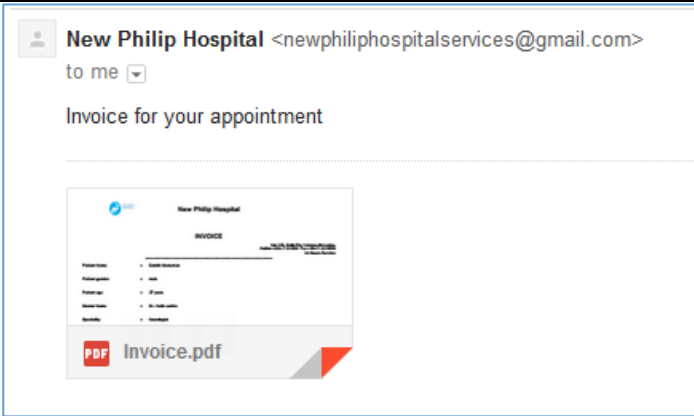
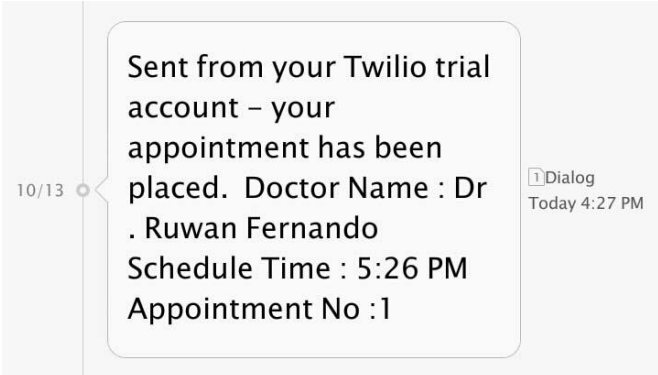
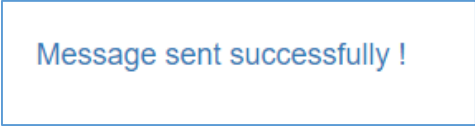
<p>2</p>	<p>User put an appointment using system</p>	 <p>New Philip Hospital <newphiliphospitalservices@gmail.com> to me ▾ Invoice for your appointment</p> <p>Invoice.pdf</p>  <p>Sent from your Twilio trial account – your appointment has been placed. Doctor Name : Dr . Ruwan Fernando Schedule Time : 5:26 PM Appointment No :1</p> <p>10/13</p> <p>Dialog Today 4:27 PM</p>	<p>Pass</p>
<p>3</p>	<p>Users and admin send messages within system</p>	 <p>Message sent successfully !</p>	<p>Pass</p>

Table E.2 - Test results for notification management

APPENDIX F – CODE LISTINING

This section is to enlighten the reader more about the code structure of the system. Please refer the CD-ROM for the complete system code.

Controller code for add schedule module

```
<?php
include '../common/session.php';
require_once '../common/db_connection.php';
include '../model/schedulemodel.php';
$she_obj = new schedule();
$status = $_REQUEST['status'];
switch ($status) {
    case 'Add':
        $doc_user_id = $_POST['doc_name'];
//convert time to 24 hour format
        $time = date("G:i:s", strtotime($_POST['timepicker']));
        $spec_id = $_POST['speciality'];
        $Max_pNo = $_POST['MaxPatientNo'];
        $hospital = $_POST['hospital'];
        $fee = $_POST['fee'];
        $date = $_POST['date'];
        $vat = $_POST['vat'];
        $discount = $_POST['discount'];
        if ($doc_user_id !== "" & $time !== "" & $spec_id !== "" & $Max_pNo !== "" & $Max_pNo >= 1
& $hospital !== "" & $fee !== "" & $fee >= 1 & $date !== "" & $vat !== "" & $discount !== "") {
```

```

        $she_obj->AddSchedule($doc_user_id, $date, $hospital, $time, $spec_id, $Max_pNo, $fee, $vat,
$discount);

        $msg = "Schedule added successfully !";

        $msg1 = base64_encode($msg);

        header("Location:../view/schedule.php?msg=$msg1");

    } else {

        $msg = "Check input values again";

        $msg1 = base64_encode($msg);

        header("Location:../view/addschedule.php?msg=$msg1");

    }

    break;

case 'update':

    $doc_user_id = $_POST['doc_name'];

//convert time to 24 hour format

    $time = date("G:i:s", strtotime($_POST['timepicker']));

    $spec_id = $_POST['speciality'];

    $Max_pNo = $_POST['MaxPatientNo'];

    $hospital = $_POST['hospital'];

    $fee = $_POST['fee'];

    $date = $_POST['date'];

    $vat_id = $_POST['vat'];

    $discount_id = $_POST['discount'];

    $schedule_id = $_REQUEST['schedule_id'];

    if ($doc_user_id !== "" & $time !== "" & $spec_id !== "" & $Max_pNo !== "" & $Max_pNo >= 1
& $hospital !== "" & $fee !== "" & $fee >= 1 & $date !== "" & $vat !== "" & $discount !== "") {

        $she_obj->updateSchedule($doc_user_id, $date, $hospital, $time, $spec_id, $Max_pNo, $fee,
$vat_id, $discount_id, $schedule_id);

```

```

    $msg = "Schedule updated successfully !";
    $msg1 = base64_encode($msg);
    header("Location:../view/schedule.php?msg=$msg1");
} else {
    $msg = "Check input values again";
    $msg1 = base64_encode($msg);
    header("Location:../view/addschedule.php?msg=$msg1");
}
break;
case 'cancel':
    $schedule_id = $_REQUEST['schedule_id'];
    $she_status = '0';
    $she_obj->cancelOrActiveSchedule($she_status, $schedule_id);
    $msg = "Schedule cancelled !";
    $msg1 = base64_encode($msg);
    header("Location:../view/schedule.php?msg=$msg1");
    break;
case 'active':
    $schedule_id = $_REQUEST['schedule_id'];
    $she_status = '1';
    $she_obj->cancelOrActiveSchedule($she_status, $schedule_id);
    $msg = "Schedule activated !";
    $msg1 = base64_encode($msg);
    header("Location:../view/schedule.php?msg=$msg1");
    break;
}

```

?>

Code for view appointments

<body>

```
<div class="container-fluid">
```

```
<div class="row">
```

```
<!-- header start -->
```

```
<?php include '../common/header.php'; ?>
```

```
<!-- header end -->
```

```
<div class="container-fluid">
```

```
<!--page navigation bar start -->
```

```
<nav>
```

```
<div class="nav col-md-12" >
```

```
<ol class="breadcrumb">
```

```
<li><a href="dashboard.php">Dashboard</a></li>
```

```
<li><a href="payments.php">Payments</a></li>
```

```
<li><a href="viewappointments.php">View Appointments</a></li>
```

```
</ol>
```

```
&nbsp;</div>
```

```
</nav>
```

```
<!-- page navigation Navigation end -->
```

```
<div class="col-md-12">
```

```
<!-- add appointment -->
```

```
<div class="row">
```

```
<div class="col-md-12">&nbsp;</div>
```

```
</div>
```

```
<!-- appointment table start -->
```



```

<div class="row">
  <div class="col-md-12">
    <table id="myTable" class="display stripe hover" width="100%" cellspacing="0">
      <thead>
        <tr>
          <th>Appointment No&nbsp;</th>
          <th>Doctor Name&nbsp;</th>
          <th>Speciality&nbsp;</th>
          <th>Date&nbsp;</th>
          <th>Time&nbsp;</th>
          <th>Amount(Rs)&nbsp;</th>
          <th>Print&nbsp;</th>
          <th>View Prescription&nbsp;</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <?php while ($appointments = $result->fetch_assoc()) { ?>
            <td><?php echo $appointments['app_no']; ?></td>
            <td><?php echo $appointments['user_fname'] . " " .
$appointments['user_lname']; ?></td>
            <td><?php echo $appointments['spec_name']; ?></td>
            <td><?php echo $appointments['schedule_date']; ?></td>
            <td><?php echo date("g:i A", strtotime($appointments['time'])); ?></td>
            <td> <?php echo number_format($appointments['amount'],2); ?> </td>
            <td>
              <a
href="../controller/printappointmentcontroller.php?schedule_id=<?php echo
$appointments['schedule_id']; ?>" target="_blank">

```

```

                <button name="download" id="download" class="btn btn-default
glyphicon glyphicon-print">&nbsp;Print</button>
            </a>
        </td>
    </td>
    <td>
        <a href="viewprescription.php?status=view&puser_id=<?php echo
$user_id; ?>&schedule_id=<?php echo $appointments['schedule_id']; ?>&app_no=<?php echo
$appointments['app_no']; ?>">
            <button name="addpres" id="addpres" class="btn btn-
primary">View Prescription</button>
        </a>
    </td>
</tr>
<?php } ?>
</tbody>
</table>
</div>
</div>
<!-- payment table end -->
</div>
</div>
<div class="col-md-12 footer text-center " >
    <?php include './common/footer.php'; ?>
</div>
</div>
<script type="text/javascript" charset="utf8"
src="../../DataTables/dataTables.bootstrap4.min.js"></script>
<script type="text/javascript" charset="utf8"
src="../../DataTables/jquery.dataTables.min.js"></script>
<script type="text/javascript" charset="utf8"
src="../../DataTables/dataTables.buttons.min.js"></script>

```

```
<script type="text/javascript">
    $(document).ready(function () {
        $('#myTable').DataTable();
    });
</script>
</body>
```

APPENDIX G - CLIENT CERTIFICATE

New Philip Hospitals (Pvt) Ltd.



7th October 2017

The Coordinator – BIT Degree Programme,
University of Colombo School of Computing,
University of Colombo,
Colombo,


Dear Madam/Sir,

Re: Online Appointment Management System for New Philip Hospitals (Pvt) Ltd

This is to confirm that Mr.H.D.Damith Madushan has successfully analyzed designed and developed a Online Appointment Management System for New Philip Hospitals (Pvt) Ltd in partial fulfillment of the requirement for the degree of Bachelor of Information Technology.

This letter of certification is issued on the request of Mr.H.D.Damith Madushan

New Philip Hospitals (Pvt) Ltd



Chris Fernando
Director

225, Galle Road, Kalutara South, Sri Lanka. Tel: +94 34-2222888, 2224941
Fax: +94 34 2223550 E-mail: director@philiphospitals.com
Web : www.philiphospitals.com

GLOSSARY

AJAX – AJAX is used to send and retrieve data without reloading web pages

Apache – A free and open source web server

Bootstrap – A front end web development framework

DataTables – powerful jQuery plugin for creating table listings and adding interactions to them.

Encryption Algorithm - is a mathematical method used to transform Information (plain text) into an unreadable form (cipher text).

Firewall - A network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules.

Graphical User Interface – Allows users to interact with system

JavaScript - Powerful and popular language for programming on the web.

jQuery – A JavaScript library which simplifies the JavaScript coding

PHP – An open source server-side scripting language

SHA1 – A hashing function which produces 160 bit hash value

UML – A modeling language for object oriented development

INDEX

A

Accuracy vi, 10
AJAX..... xi, 31, 90
Apache web server..... iii, 61

B

billing.....6
BIT..... i, iv
Black box Testing viii, 42

C

Class diagramvi, ix, 13, 20
CODE LISTINING84
CONCLUSION viii, 4, 53
Critical assessments..... viii, 54
CSS..... xi, 31

D

DBMS..... xi, 56
DESIGN DOCUMENTATION62
Development tools vii, 30

E

Efficiency..... vi, 9, 55
ER..... xi
EVALUATION vii, 4, 42

F

Fact finding techniques.....v, 6
functional.....v, 4, 5, 6, 7, 9, 42, 53, 55

G

GUI..... xi

H

HTML..... xi, 31, 32

I

IMPLEMENTATION..... vii, 4, 29
Interfacevi, vii, ix, xi, 23, 24, 26, 27, 90

L

Login vii, xi, 25, 44, 48, 49, 50, 70

M

MVC..... ix, xi, 16, 35, 36
MySQL..... iii, 29, 30, 33, 37, 59, 61

N

Network Architecture vii, 34
non-functional requirements 4, 5, 6, 9

O

Object-oriented design.....13, 55

P

Payments..... 2, 7, 63
PDF..... xi
PHP..... iii, xii, 29, 30, 31, 32, 33, 57, 59, 90
Privacy vi, 10
Prototyping Model..... vi, 14, 56

R

Rapid Application Developmentvi, xii, 13, 14
Rational Unified Process iii, 13, 56
REFERENCES.....55
Reliability vi, 10

Requirements gathering and analyzing5

S

Security v, 9

Sequence diagram ix, 13, 19, 66, 67

Software Process Models vi, 13

SQL xii

SYSTEM DOCUMENTATION59

T

Test casex, xi, 44, 45, 46, 47, 48

Test Results..... viii, xi, 48, 50, 51

U

UML xii, 13, 90

Usability v, 9, 55

Use case diagram ix, 13, 62, 63, 64, 65

User acceptance testing viii, 4, 52

User interface design23, 56

V

Validation..... vii, 40, 41, 42

W

Waterfall Model..... vi, 14

White Box Testing..... viii, 43