



Online Medicare Medical Portal for Moratuwa Medical Centre

**A dissertation submitted for the Degree of Master
of Information Technology**

**D.M.S.P. Rohana
University of Colombo School of Computing
2018**



Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: D.M.S.P.Rohana

Registration Number: 2014/MIT/054

Index Number: 14550542

Signature:

Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

.....D.M.S.P.Rohana.....

Under my supervision, the thesis has been prepared according to the format stipulated and is of an acceptable standard.

Certified by:

Supervisor Name: Dr. Enosha Hettiarachchi

Signature:

Date:

Abstract

The Moratuwa medical Centre (PVT)LTD has stepped forward significantly in a very short period of time among one of the most popular medical centres and newly emerging medical centres in a Sri Lankan competitive market system.

The company was established in 2012 as a small medical centre with two doctors. Currently, there are more than 50 doctors available for channelling at the Moratuwa medical centre. However, all the operations are handled by an old traditional manual system with paper and phone-based system which has become a barrier for the company growth and for a better coordination.

When dealing with the main operations of the company, there is a better coordination can be assured to the internal and external users, especially for patients, doctors and management of the Moratuwa medical centre by the proposed web-based system. Moreover, a lot of facilities can be provided by the web-based system to the users such as online channelling, view medical history, doctor scheduling leaves and stock control for pharmacy and external user communication, factory expenses handing and machinery maintenance operations.

The system was developed using PHP (PHP Hypertext Processor) open source server-side scripting language. MySQL was used as the database tool for designing the database. Apache web server was chosen to run the system. Adobe Photoshop CS5 was chosen as the interface designing tool in the main stages of the developing.

The Rational Unified Process (RUP) was identified as the most suitable development methodology based on its flexible developing theory and related advantages in comparison to other software methodologies according to the present software engineering theories and by studying the problem domain,

At the end of this development process, the web-based system which was tested by many users of the company was successfully handed over to the client to enhance their company business process as well as to help the company to become best medical organization among the competitors.

Table of Contents

Declaration.....	i
Abstract.....	ii
Table Of Contents.....	iii
List of Figures.....	vi
List of Tables.....	vii
Chapter 1 Introduction.....	1
1.1 Motivations of the project	2
1.2 Aims & The Objectives Of the Project.....	3
1.3 Scope Of the project.....	4
1.4 Outline Of the Chapters.....	6
Chapter 2 Literature Review.....	7
2.1 Similar Systems and the Literature Review	7
2.1.1 OPEN EMR.....	7
2.1.2 EChannling.....	8
2.1.3 Asiri Hospotals.com.....	9
2.2 Design Comparisan.....	10
Chapter 3 Analysis & Design.....	11
3.1 Requirement Analysis	11
3.2 Requirement Gathering Techniques.....	11
3.3 Functional Requirements.....	12
3.4 Non-Functional Requirements.....	13
3.5 Methodology.....	13
3.6 Design Overview.....	13
3.6.1 Use Case Diagrams & Use case Narratives.....	14
3.6.2. Activity Diagrams	15
3.7 Database Designing for the system.....	17
3.8 User Interface Design for the system.....	18
3.8.1 User Login Interface.....	18

3.8.2	User Registration Interface.....	19
3.8.3	Doctor search result interface.....	19
3.8.4	Doctors availability showing interface.....	20
3.8.5	Contact Us Interface.....	20
3.8.6	About Us Interface.....	21
3.8.7	Dashboard for the receptionist.....	21
3.8.8	Dashboard for the doctor.....	22
3.8.9	Dashboard for the patient.....	22
Chapter 4 Implementation.....		23
4.1	Introduction	23
4.2	Implementation Environment.....	23
4.3	System Developed Tools and Technologies	23
4.4	Major Code Segments.....	24
4.4.1	Register Blade.....	25
4.4.2	Index Page.....	26
4.4.3	Password Reset Function of patient.....	27
4.4.4	Public Controller.....	28
4.4.5	User Table Migration	28
4.4.6	Main Routes Of the system.....	28
4.4.7	Contact Us Page.....	29
4.4.8	Appoinments Controller.....	30
4.4.9	Patient Login Controller	31
4.4.10	Patient Forget Password Controller.....	31
Chapter 5 Evaluation.....		32
5.1	User Evaluation & system testing.....	33
5.2	System Test Plan.....	34
5.3	System Test Cases	35
5.3.1	Test Case for User Authentication.....	33
5.3.2	Test Case for Doctor search	34
5.5.3	Test case for managing users.....	34

Chapter 6 Conclusion & Future work.....35

References List.....36

Appendix A-system documentation.....37

Appendix C-user documentation.....39

List of Figures

Figure	Title	Page number
Figure 2.1	Open EMR Main Interface	07
Figure 2.2	Echannling Landing Page	08
Figure 2.3	AsiriHospitals.com Main Page	09
Figure 3.1	Level1 use case Diagram for admin actions	14
Figure 3.2	Activity Diagram for the user Registration in the system	15
Figure 3.3	Activity Diagram for the registered user Login in the system	16
Figure 3.4	Activity Diagram for the admin login and admin activity	16
Figure 3.5	Database Design for the system	17
Figure 3.6	User login interface	18
Figure 3.7	User registration interface	19
Figure 3.8	Doctor search result interface with speciality	19
Figure 3.9	Doctor availability showing interface	20
Figure 3.10	Contact Us Interface	20
Figure:3.11	About Us Interface	21
Figure:3.12	Dashboard for the receptionist	21
Figure:3.13	Dashboard for the doctors	22
Figure:3.14	Dashboard for the patient	22

List of Tables

Figure	Title	Page number
Table 2.1:	System Comparison Metrics	10
Table 3.1	Use case Narratives for admin actions in the system	14
Table 3.2	Use Case Narrative for the Manage contents in the system	15
Table 4.1	Hardware & Software requirement for the system	24
Table 5.1	User Authentication Test cases	35
Table 5.2	Add Product Module Test cases	36
Table 5.3	Manage Users Test cases	37
Table 5.4	Adding product Test cases	38

Chapter 1: Introduction

Moratuwa Medical centre is located near Katubedda Junction on Galle road. People near Moratuwa , panadura and piliyandala area go to this place to channel doctors, to obtain medical testing, and for other health-related requirements such as OPD services and Pharmacy . Consultants with various specialities conduct private practice at Moratuwa Medical Centre. Those who are having issues in channelling doctors inside the city of Colombo visit this place to channel the doctors due to easy accessibility.

The current system at the Moratuwa medical centre totally runs on paper and phone based. People who want to channel the doctors from the Moratuwa medical centre are required to come here to make the payment and book the appointment for channelling doctors. Otherwise, the patient has to give a phone call to book the doctor with the next available number. However, the given number is confirmed only after the payment have been made. Priority gives to the patient who comes to the centre to channel the doctors. There is a limit of phone call bookings numbers per day per doctor. Currently, all the doctor records are at the record book as well as at the document at the receptionist computer at the Moratuwa medical centre.

List of channelling patients' details are entered into the book. A relevant staff member of the Moratuwa medical centre is supposed to give a phone call to the doctor to confirm his arrival at Moratuwa medical Centre on the particular date and time. Patients have to come to the medical centre or give a phone call to the medical centre to get to know of their appointment time with a particular doctor because of the manual system.

Before the patient diagnosis, the doctor is supposed to inquire about the medical history of the patient manually.

After diagnosis, the doctor would give a ticket that contains the dosage and duration of the relevant drugs. The doctor would tell the upcoming appointment dates of the patient especially for the diseases that need to be treated long term.

The patient needs to hand over the ticket to the pharmacy at the Moratuwa medical centre to get the relevant drugs recommended by the doctor.

Doctor fees are calculated after counting each appointment manually. Doctor prices are variable and medical centre charges are fixed. Doctors do not have a method to cross check the number of patients came to them.

When the patient needs to do a particular medical testing from the Moratuwa medical centre, he/she is required to give a phone call and check the availability of the testing facility and also have to come to collect the test reports.

Due to manual system runs on Moratuwa medical centre there is a lot of wastage in time and money.

1.1 Motivations of the project

Using paper and Phone; operations are currently being handled by Moratuwa Medical Centre. But in the purchase of drugs and other products for Pharmacy when needed, a lot of difficulties are being faced. In this situation, a proper idea about the present status of the medicine stock until it runs out is not known to the management and the pharmacist. Delays in buying drugs and related products are being caused due to this. In some cases, misplacement of some valuable stock reports by the management had created big problems when ordering items. When the Managing Director is out of the premises, it had been difficult to obtain and view the details of day to day activities of the medical centre.

A proper and latest method to all the stakeholders at the system such as patients, doctors, medical centre staff and management is an essential part for the future development of the Moratuwa medical centre.

In the current system, all the details are in papers. It has been time-consuming to generate essential reports such as stock on hand, stock levels report by the current system at the medical centre.

Thus, the new system was developed to support the company to handle their main operations successfully by overcoming the above problems, which occurred due to using the paper and phone-based system.

1.2 Aims & The Objectives of the Project

The main purpose of doing this project is to automate the ongoing paper and phone-based system with a view to reduce the wastage of time and money and to enhance the profit of the Moratuwa Medical Centre and to give a quality service to its customers.

The following objectives are also expected to achieve.

- Minimize the paperwork as well as the time wastage for patients as well as the staff of the Moratuwa Medical Centre
- Enhance the quality of service by introducing easy access and easy handling system.
- Enhance the profit with minimized the paper and time wastage
- Provide a transparent system to doctors as well as to the Moratuwa Medical Centre staff, patients and the doctors who are using the new system.
- Use of new technology to automate the manual system of the Moratuwa Medical Centre and keep up with the competition.
- Online appointments at the home, will leads to time and cost reduction.
- The patient can view their medical history, detail expenses for medicines from the medical centre at the particular time period.
- Doctors can view the patient medical history and gain the essential details before prescribing the drugs to the patient.
- Management of the medical centre can view the current situation in the medical centre even when they are out of the medical centre.
- Stock control system at the pharmacy to maintain drugs stock without out of balance and gain essential details for the order the drugs and other pharmacy items.
- Expired drugs can be noted reports generated through inventory control system and can be removed from the system manually.
- Doctors have the system access to mark their availability and will lead to preventing miscommunication problems between doctors and medical centre staff.

1.3 The scope of the Project

The system would be developed using PHP, Laravel Framework, Bootstrap and JavaScript.

The system would contain medical portal as well as the inventory control system for the pharmacy.

Patient (registered user), doctor, management of Moratuwa medical centre (Managing Director), receptionist, pharmacist, cashier, Stock Keeper and the admin are the users of the proposed system.

The patient needs to register with the system online. Successfully registered user(patient) will notify by the email message. Then, the patient can log in to the system using an email address as the username and relevant password. The patient can reset the password in the event of the forgotten password. The registered patient can search doctors available at the moratuwa medical centre with respect to the speciality and available dates. The patient can select a doctor with relevant date and time and place a temporary appointment to the doctor through the new system.

The receptionist of the Moratuwa medical centre is one of a new user of the system, should be created by the admin. The receptionist can log in to the system and search available doctors, temporary appointments pleased by various patients. When the patients came to the moratuwa medical centre who have pleased the online temporary appointment have to make relevant payment to the receptionist and convert it to the permanent appointment. Otherwise, a temporary appointment will remain as a request from the patient. The patient came to the Moratuwa medical centre and search available doctors through the receptionist. Then, the patient can get the permanent appointment with the payment. if the patient is a new user, then the receptionist will give request to admin for add the patient to the system. Patients have to provide the required details to the receptionist. Existing patients at the system will have to provide user id and make payment. The receptionist will have a search box for the existing users. The receptionist will have the reporting facility in the system such as Patient count for each doctor and total payment for each doctor.

Doctors having the system access. Doctors will be created by the admin. Doctors will log in to the system and mark their availability for the period of one-month time. Normal Prescribe method will apply when the new patient came to the doctor. The doctor can enter a prescription using the system. When the user comes to the doctor, then the doctor can view the patient medical history report and enter the prescription. Doctors having a daily patient count and payment details.

After entering the prescription by using the system, doctors have to click on the print button.

Copy of the prescription will print to the cashier and the pharmacist at the pharmacy of the Moratuwa medical centre. When the entering drugs to the prescription, doctors can view the available drugs list at the pharmacy as well as the drugs with out of stock. If the drugs are not available at the pharmacy, the doctor has to write a manual prescription including those drugs and handover to the patient. Doctors can edit the prescription. However, they can't delete the prescription. When printing the prescription, another receipt contained the patient id, doctor id and the date/time will print automatically generated by the system. The doctor has to issue this receipt to the patient.

Pharmacist and the cashier at the pharmacy are also users of the new system at the Moratuwa medical centre. Both users should be created by the admin. After entering the system prescription doctor have to click the print button. Then, the prescription copy can be viewed by the pharmacist as well as the cashier at the pharmacy. The Pharmacist login to the system and view the prescription according to the patient id as well as the doctor id and date. Then the drugs list can be prepared by them.

Patient has to go to the cashier at the pharmacy and pay the total bill according to the drugs Patient have to hand over the prescription to the cashier which is given by the doctor. Then the total bill has to paid by the patient to the cashier. After that, the cashier will issue the system generated payment receipt which contains the paid stamp and cashier signature. Patient has to handover the payment receipt to the pharmacist and collects the drugs.

Stockkeeper at the pharmacy is another admin created user at the new system which the pharmacy inventory system stock keeper has the full access to inventory module at the pharmacy such as add new item, manage items, etc.. Stock keeper having main reporting access of the inventory system such as stock on hand, out of item stock, stock issue at the particular time period, etc...

Managing Director of the Moratuwa medical centre is another admin created a user of the new system. All the reporting access to the new system such as stock reports, patient detail and doctor detail report, etc...can be done by him/her.

Admin user is the main user of the new system. Admin user has full control of the system with all the access privilege to do user management and the system management. All other system users except patient have to be created by the admin.

1.4Outline of the Chapters

Outline Of the chapters of the project is as follows.

Chapter 2- Literature Review

This chapter will contain the Survey of existing similar systems. Issues related to those systems and the solutions that would provide the proposed system. Also contains the technologies that used for the design and development of the proposed system with the references related for this project.

Chapter 3- Analysis & Design

Requirement gathering techniques such as interviews and questionnaires are described in this chapter. How the current system works and what the requirements are for the project (functional requirements and non-functional requirements) are also identified here. UML diagrams such as use case diagrams are drawn and provided to identify unclear requirements and obtain better ideas about the system.

Chapter 4- Implementation

This chapter will include hardware & software requirements, tools and technologies used, as well as existing codes of the system.

Chapter 5- Evaluation

How the implement system should be tested as described in this chapter; such as test cases planning, applying different testing methodologies to test for the accuracy of the system. User acceptance testing is also carried out at this stage

Chapter 6- Conclusion & Future work

Lessons learnt by implementing this system with a brief description of the company and future improvements of the system are discussed in this chapter.

Chapter 2: Literature Review

2.1 Similar Systems and the Literature Review

By studying ongoing famous web-based medical systems, to gain idea about medicine portal as well as the find the problems that having with such systems and how the required functionalities should be presented. Following are a few similar systems that were reviewed to get the idea to build the new system.

2.1.1 OPEN EMR

<http://www.open-emr.org/> [01]

Open source. Patient scheduling, prescriptions, appointments & medical billing contained at this.

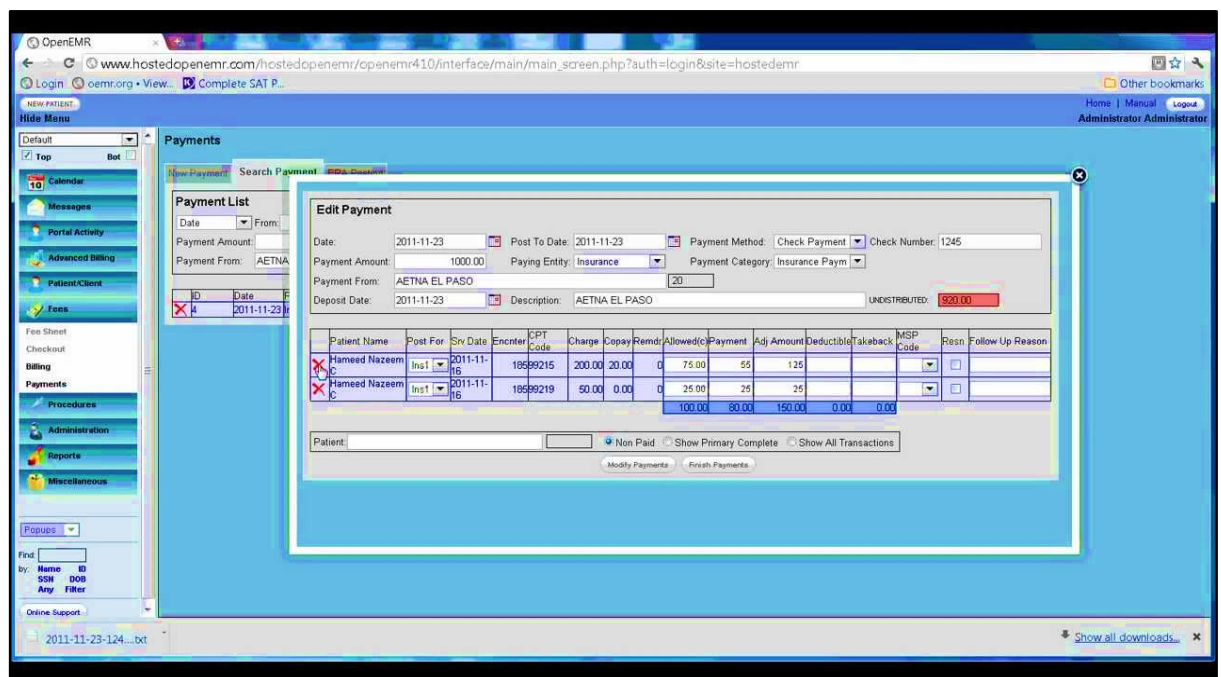


Figure:2.1 Open EMR Main Interface

This is open source system. Doctors do not have access to the system. Does not having pharmacy module with the stock control system. does not have patient record history at the system. Proposed system contains the doctor access to the system as well as the stock control system for the pharmacy and patient records are stored in the system.

2.1.2 EChannelling

<https://www.echannelling.com/Echannelling/index> [02]

Echannelling is the mostly used doctor channelling solution in Srilanka to channel the doctors at the various hospitals. The doctors can be searched by name, by speciality, by a hospital or by date. This is not an open source software solution.

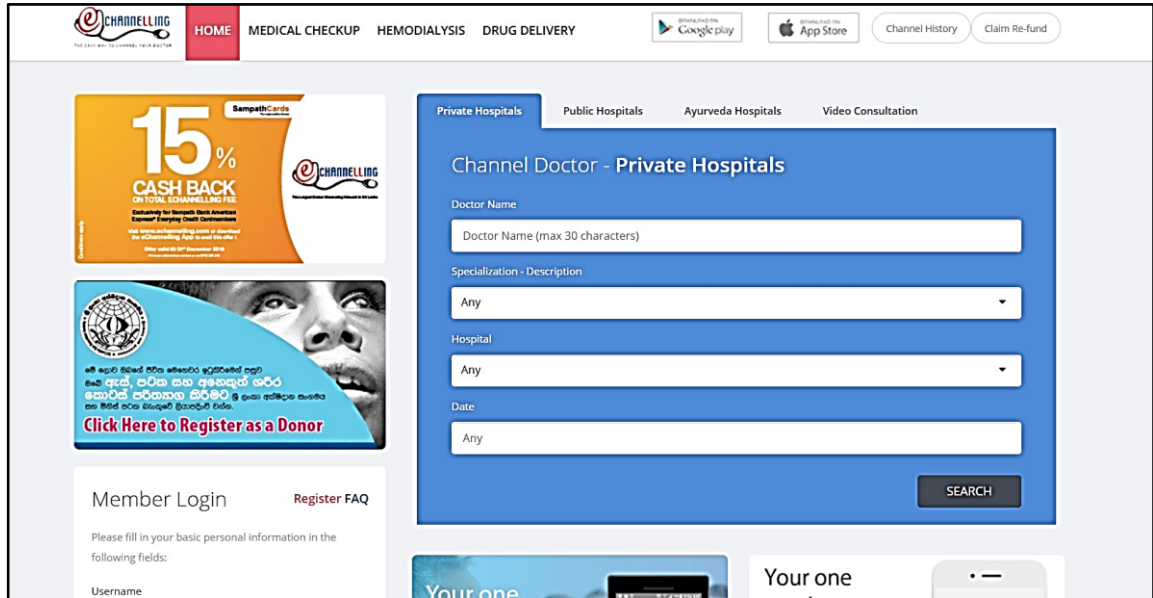


Figure:2.2 Echannelling Landing page

In here system Does not have a patient login or record history. Does not having a patient login to the system and patient record history at the system. The proposed system would have patient login as well as records the patient history at the system.

Doctors do not have system access to the echannelling system. In here doctors do not have the system access. Site admin will do all the updated related to the doctors as well .eg: doctor availability

At the proposed system doctors having access to the system as well as they can log in to the system and mark their availability.

2.1.3 Asiri Hospitals.com

<http://www.asirihospitals.com/> [03]

Asiri hospital group is another major portal used for the channel the doctors by the patients.

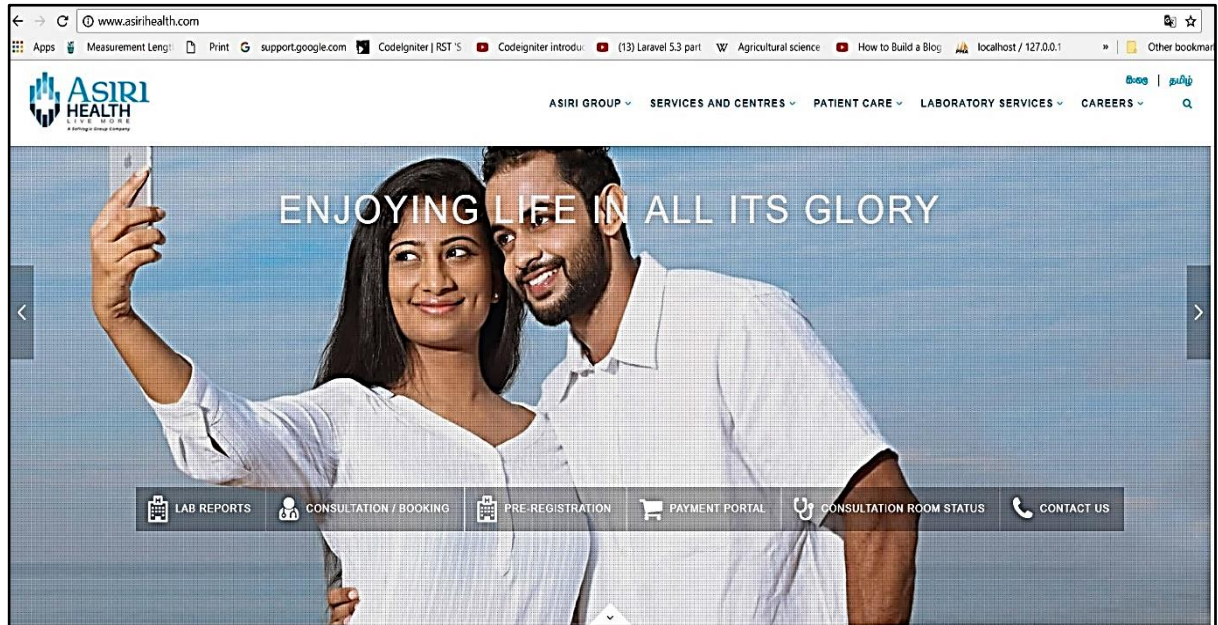


Figure:2.3 AsiriHospitals.com Main page

In here patient can search doctors and make appointments. At the home page, there is a link to consultation booking to search the doctors and place the appointments. Does not have a patient log in and record history stored at the portal.

Does not having a patient login at the system as well as the patient record history.

Also in this system Doctors does not have system access. Site admin has to mark the availability of the doctors because doctors do not have the system access.

The proposed system would contain the patient access, patient record history, as well as the doctor login to the system as well as the prescription, can be generated through the system.

2.2 Design Comparison

System Comparison Matrices

Feature	Open EMR	Echannling.com	Asirihospitals.com	New System
Patient Login	Yes	Yes	No	Yes
Doctor Channeling	Yes	Yes	Yes	Yes
Store Patient Record History	No	No	No	Yes
Doctors Having system access	No	No	No	Yes
Stock Control System	No	No	No	Yes
Open Source	Yes	No	No	No

Table:2.1 System Comparison Matrices

Those systems I have selected for the comparison having some common weaknesses. System access to the doctors, patient medical history & stock control system does not have in these systems. some systems having patient login facility. However, all the systems having doctor channeling function. The new system will cover and give solutions to the weaknesses that I have identified by refereeing those available common systems.

Chapter 3: Analysis & Design

3.1 Requirement Analysis:

Requirement analysis also called requirement engineering is the process of determining user expectations for a new or modified product. These features called requirements must be quantifiable relevant and detailed. [04]

Requirement gathering techniques used to collect requirements, how the current system process is going on, drawbacks of the present system, functional and non- functional requirements of the proposed system.

3.2 Requirement Gathering Techniques:

Gathering of requirements should be properly accomplished before the start of the analysis stage. The inadequate information would cause a waste of time for modifications and lose the opportunity and would cause additional unnecessary overhead.

used the interview as the main requirement gathering technique. Mainly done the interview with the people at the Moratuwa medical centre who handling the current manual process as well as the management of the Moratuwa medical centre. In here, asked the structured question set & noted down each answer.

used Observation as another requirement gathering technique. Went as a visitor to the Moratuwa medical centre and done the observation about all the subprocesses to identify the entire process of the Moratuwa medical centre currently had.

used another technique to the requirement gathering process.it is document review. Most ongoing processes of the Moratuwa medical centre mainly used the various types of documents. Hence document review led to getting a clear idea about the current process.

3.3 Functional Requirements

“In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs.”[05]

The success of this system is not only dependent on the functional requirements; it is also based on the non-functional requirements. It acts as the main role for the success of the system. Following are the major non-functional requirements which were identified at this stage:

- 1) Overall system should be simple as much as possible with the more user-friendly environment.
- 2) Access privileges should be maintained accurately.
- 3) Automated notification generating module should be generated at the accurate time levels.
- 4) System security should be highly secure to prevent unauthorized modification and Access.
- 5) User registration, login, patient medical history, search doctors & make a temporary appointment through the system can be done by the patient.
- 6) Add patient, payment for the appointment & confirm doctor appointment, make a new appointment, show the existing appointment list is the system functions to the receptionist.
- 7) Mark Availability, check patient history detail, patient count report, enter prescription to the patient are the access for doctor users at the system.
- 8) View prescription and enter payment receipt print can be done by the cashier at the pharmacy.
- 9) View prescription and issue drugs from the system done through the system by the pharmacist at the pharmacy.
- 10) Stockkeeper having all the stock control functions.
- 11) Managing Director having all the reporting Facility.
- 12) Admin having the all the system control and user control facility of the system.

3.4 Non-Functional Requirements

Usability

The system would be easy to use by all people categories including the admin, patient, doctor, pharmacist, receptionist, cashier and storekeeper.

The system must be quickly accessible by all peoples whenever needed.

The system shall provide a simple as well as user-friendly interface along with easy navigation methods to access needed information.

Performance

All the functions of the system should be available to the user every time the system is turned on and can browse the portal from any desktop and laptop device as well as any smart devices with an average internet connection. The data which contain in the system would up to date and also make sure those data are accurate.

Reliability

The system should have with minimized downtime capability such as uninterrupted power supply for the hosting server computer and reliable best quality internet connection supply to the server computer.

Security requirements

The system should have with backup and restore mechanism as well as the auto schedule backup system. Those backups can be used to restore crashed system to original system.

3.5 Design Overview

System administrator, Receptionist, Doctor & the patient are the main system users of this system.

In here the use case diagrams along with the use case descriptions, activity diagrams for each user activities and the sequence diagram for the proposed system are included.

3.6.1 Use Case Diagrams & Use case Narratives

Use Case Narrative for User Registration

Use Case Name	User Registration
Actors	New Patient
Purpose	Registering user in the system
Normal Registering Events	(01)Click User Registration (02)Fill out the form & submit (03)Log in to the system
Alternative Registering Method	Create the user by the receptionist

Table 3.1 Use Case Narrative for User Registration

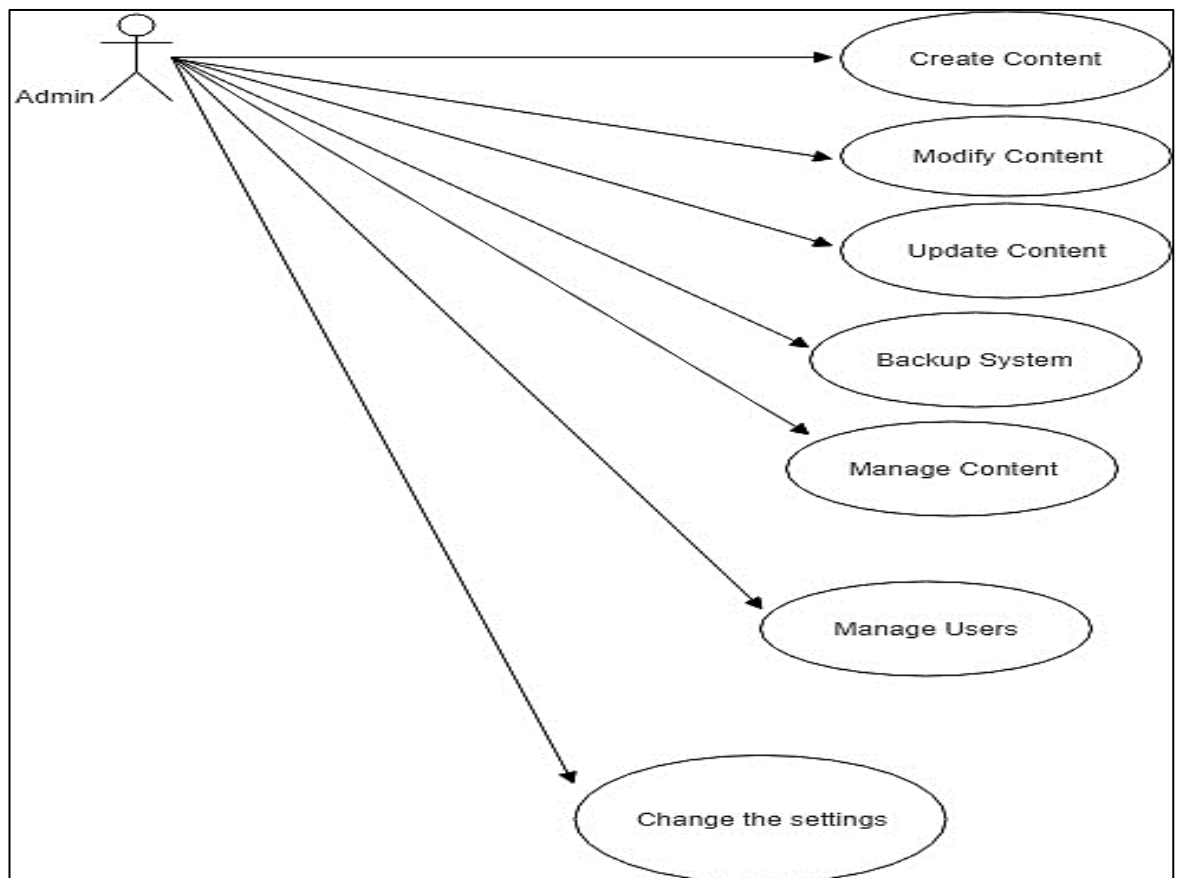


Figure 3.1 Level1 use case Diagram for admin actions

Use Case Narrative for the Manage contents in the system

<i>Use Case Name</i>	Manage content
<i>Actors</i>	Admin
<i>Purpose</i>	Manage the Content of the portal
<i>Related Events</i>	(01) Log in to the system (02) Select the contents. (03) Manage the Content (Edit/Delete/Create) (04) Validate and save the modifications (05) Log Out
<i>Pre-Condition</i>	Needs to Log as administrator by providing correct username and passwords

Table 3.2 Use Case Narrative for the Manage contents in the system

3.6.2 Activity Diagrams

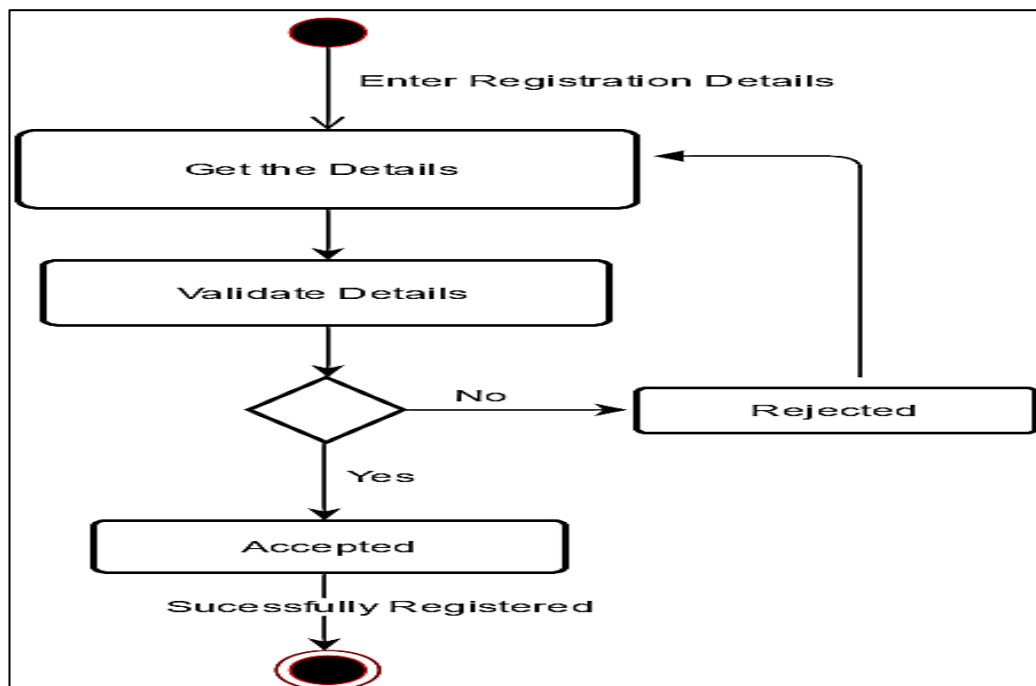


Figure 3.2 Activity Diagram for the user Registration in the system

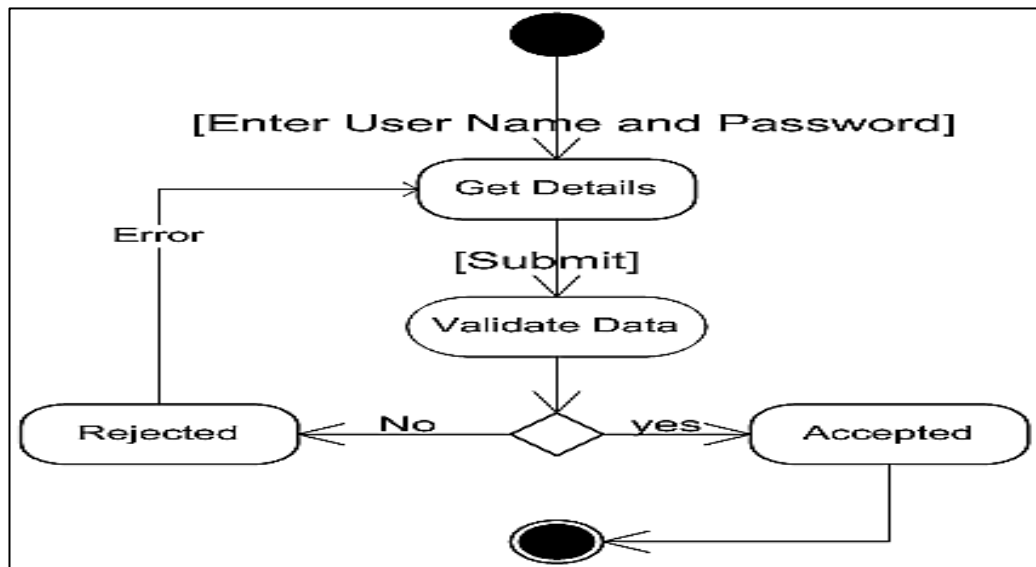


Figure 3.3 Activity Diagram for the registered user Login in the system

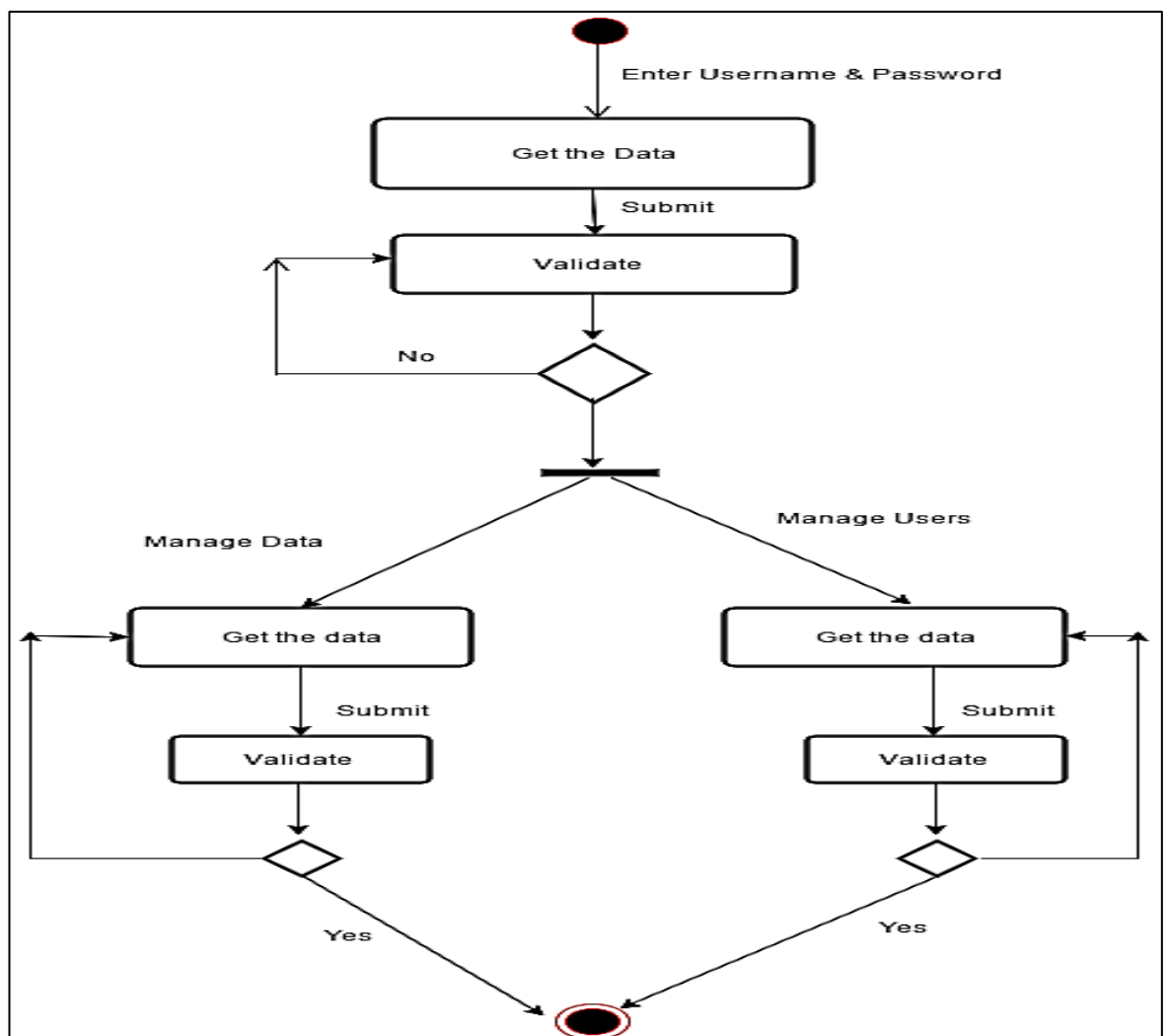


Figure 3.4 Activity Diagram for the admin login and admin activity in the system.

3.7 Database Designing for the system

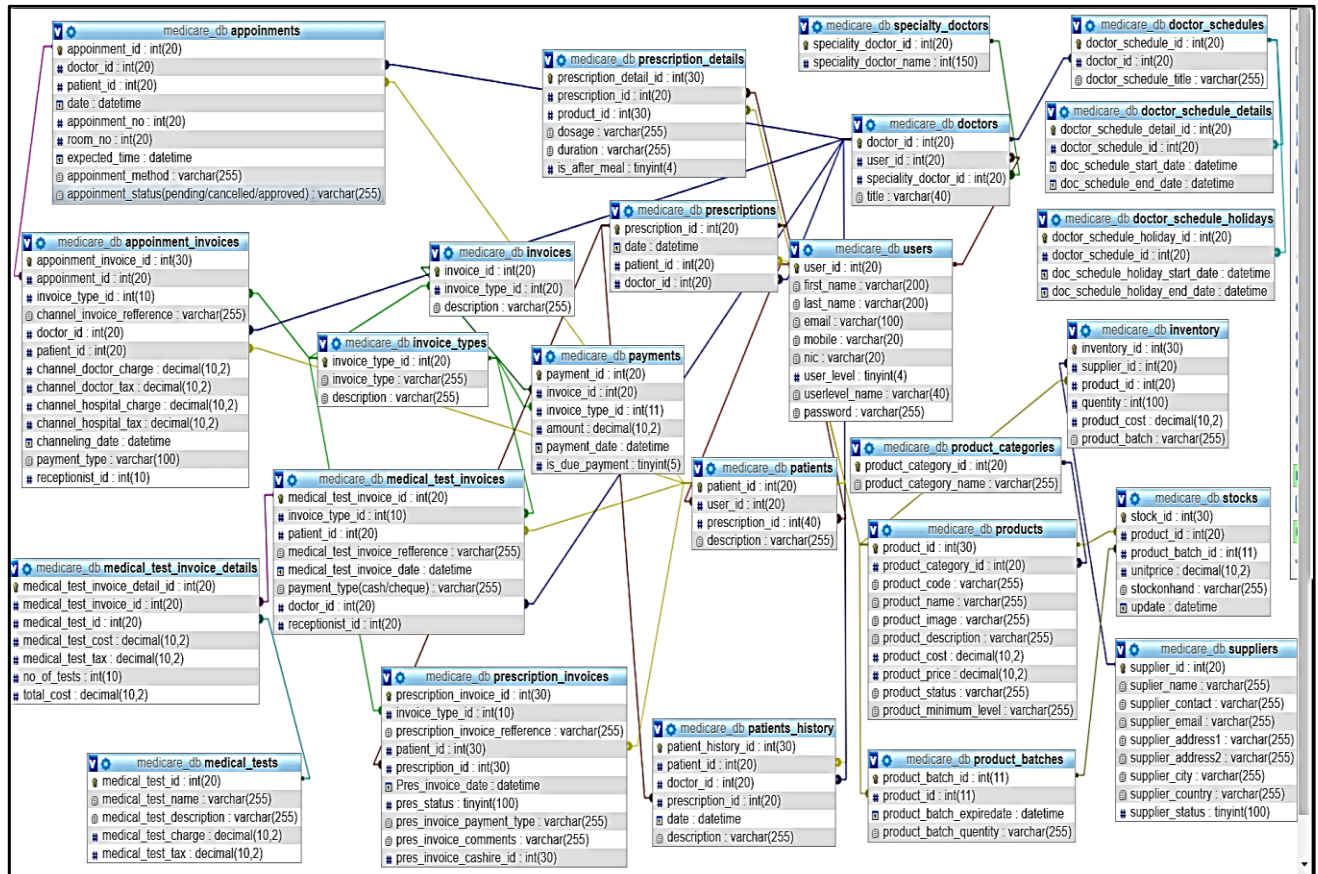


Figure 3.5 Database Design for the system

Valuable data can be kept in a proper order without losing them by a better database design. The ultimate goal can be achieved by the database normalization method. Reduction of data redundancy and keeping consistency of the database is helped by it.

Normalization consists of several normal form stages with different goals. They are First Normal Form (1NF), Second Normal Form (2NF) and Third Normal Form (3NF). Database for the system has normalized up to the Third Normal Form and the diagram is depicted under the figure 3.5

3.8 User Interface Design for the system

Among all the major components of the system, a major role of the system is played by user interfaces. Interactivity in between system and the user is managed by the interface. User friendliness, integrated colour combination and the well-organized components are dependent on it.

Without having a user-friendly interface, interaction with the system becomes hard. Privileges must be set by the System Administrator to users in different ways. Management of those privileges and presenting them effectively to users is helped by the interfaces.

3.8.1 User Login Interface

The system login page which belongs to the developed system is displayed by figure 3.6 In any computerized system the first interface encountered by the user is the login page. Therefore, by designing and handling errors properly a pleasant feeling about the rest of the system can be created within the user.in this system login interface at the landing page along with the doctor search screen.



The screenshot displays the Medicare Medical Portal's user login interface. At the top, a green header bar contains the text "Monday - Saturday, 8am to 10pm" on the left and "Call us now +94 112 655 886" on the right. Below this, a blue navigation bar features the "Medicare Medical Portal" logo and links for "Home", "About Us", and "Contact Us". The main content area is divided into three sections. On the left, a teal-colored "Channel Doctor" form includes fields for "Doctor Name" (with a "Max 30 characters" hint), "Speciality - Description" (a dropdown menu labeled "Select specialty"), and "Date" (with a "mm/dd/yyyy" hint). A "Search" button is located at the bottom right of this form. In the center, there is a photograph of a smiling female doctor in a white coat with a stethoscope. On the right, a white login/register box contains tabs for "Login" and "Register". It includes an "E-Mail Address" field (with an "Email" hint), a "Passwords" section with a "Password" field, a "Remember Me" checkbox, and a "Forgot Password?" link next to a blue "Login" button. At the bottom of the page, two circular icons are shown: one for a "Receptionist" (a person at a desk) and one for a "Doctor" (a person with a stethoscope).

Figure 3.6. user login interface

3.8.2 User Registration Interface

The user needs to register to the system before login. Only the registered users can make an appointment using the system. User registration interface showing at the Figure 3.7

The screenshot shows the 'Medicare Medical Portal' with a header bar containing the text 'Monday - Saturday, 8am to 10pm' and 'Call us now +94 112 655 886'. Below the header, there are navigation links: 'Home', 'About Us', and 'Contact Us'. The main content area is divided into three sections. On the left, there is a 'Channel Doctor' form with fields for 'Doctor Name' (with a hint 'Doctor Name (Max 30 characters)'), 'Specialty - Description' (a dropdown menu showing 'Select specialty'), and 'Date' (with a hint 'mm/dd/yyyy'). A 'SEARCH' button is at the bottom right of this form. In the center, there is a placeholder image of a female doctor. On the right, there is a 'Login' and 'Register' section. It includes a 'Name' field (with a hint 'Last Name'), an 'E-Mail Address' field (with a hint 'Email'), a 'Mobile No.' field (with a hint 'Mobile No.'), a 'NIC' field (with a hint 'NIC'), a 'Password' field (with a hint 'Password'), and a 'Confirm Password' field (with a hint 'Password'). A 'Register' button is at the bottom right of this section.

Figure 3.7 user registration interface

3.8.3 Doctor search result interface

Registered user of the system can be login to the system and search doctors at the Moratuwa medical centre.

search result interface according to the speciality in this system as per Figure 3.8.

The screenshot shows the 'Medicare: Public' interface with a header bar containing 'Login' and 'Register' links. The main content area is titled 'Available Doctors' and displays a list of doctors with their IDs and specialties. Each entry has a 'Book' button next to it. The list includes: 'doc5' (Family Physician), 'doc2' (General Surgen), 'doc3' (Cardiologist), 'doc1' (Gynecologist), and 'doc7' (Radiologist).

Figure 3.8 Doctor search result interface with speciality

3.8.4 Doctors availability showing interface

After the patient selects the doctor from the search result list and clicks channel, then the doctor availability list at the Moratuwa medical centre will display by the system. Doctors availability showing interface showing in Figure 3.9

Medicare: Public Login Register

doc1
Gynecologist

2018-06-29 : 05:00:00	Appointment No: 2 Book Now
2018-06-29 : 08:00:00	Appointment No: 1 Book Now

Figure 3.9 doctor availability showing interface.

3.8.5 Contact Us Interface

Contact Us page interface with contact details, contact us from along with the location of the Moratuwa medical centre google map showing in the Figure 3.10

Monday - Saturday, 8am to 10pm Call Us now +94 112 655 886

Medicare Medical Portal HOME About Us Contact Us

Contact Us

Email:

Subject:

Message:

[send message](#)

LAKSHAPATHIYA NORTH

LAKSHAPATHIYA SOUTH

KADUWAMULLA

RAWATHAWATTA EAST

MORATUMULLA

Moratuwa Medical Center

30 Galle Rd, Moratuwa 10400 0112 655 886

Figure 3.10 Contact Us Interface

3.8.6 About Us Interface

About Us page interface at the figure 3.8.6 contains the links for structure of the Moratuwa medical centre, services available at Moratuwa medical centre, History of Medical centre an image of the medical centre

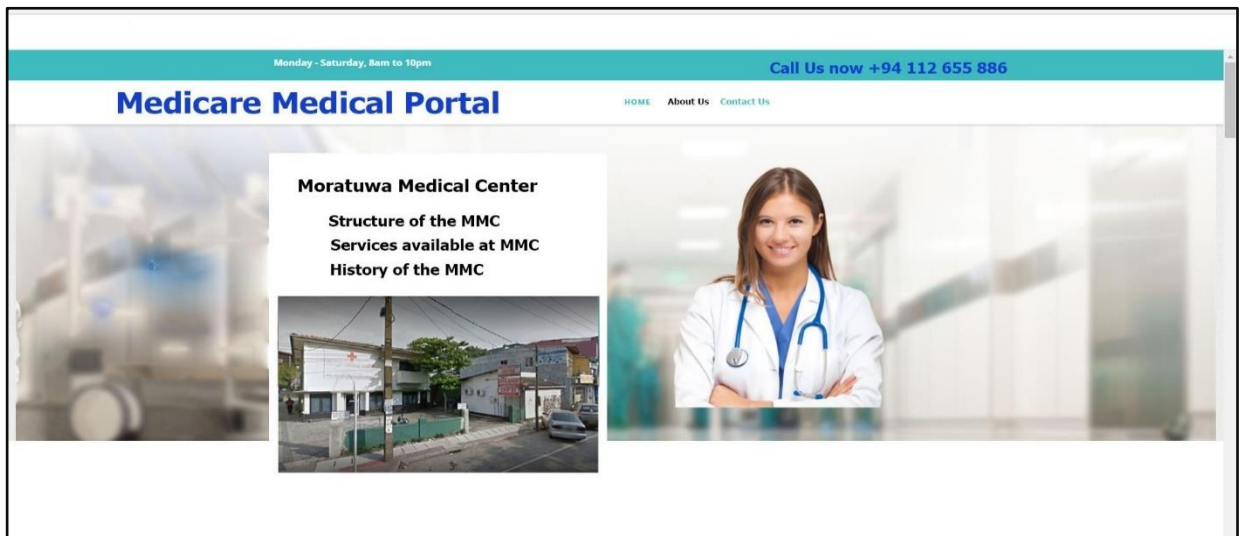


Figure:3.11 About Us Interface

3.8.7 Dashboard for the receptionist

There are 05 main functions at the receptionist dashboard at the system. They are Add patient, add doctor, add receptionist, Make an appointment and Appointment list. This dashboard showing at figure;3.8.7

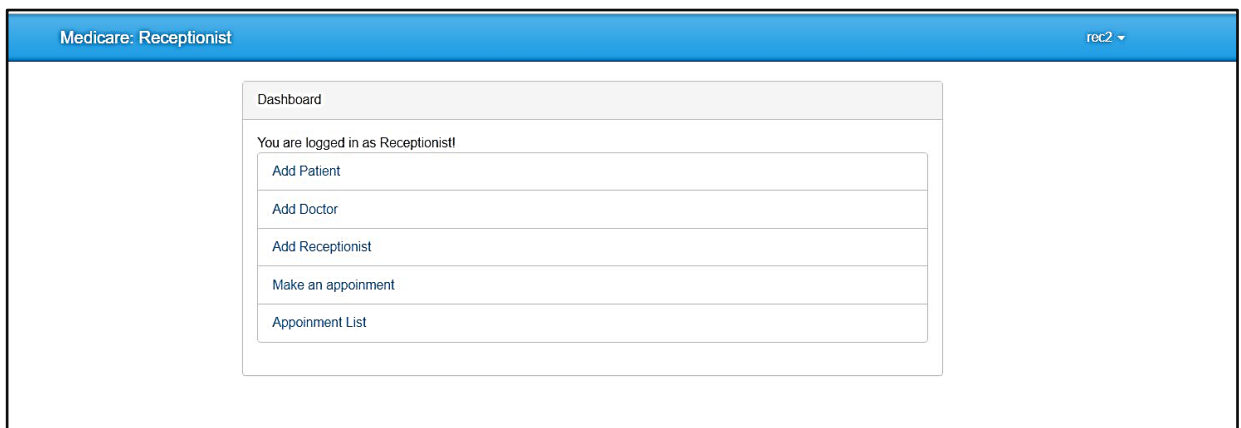


Figure:3.12 Dashboard for the receptionist

3.8.9 Dashboard for the doctor

system registered doctors having figure 3.13 interface in the system with tabs to Appointments Per day, payment per day, Appointment Details, Payment Details and view profile contains at this dashboard.

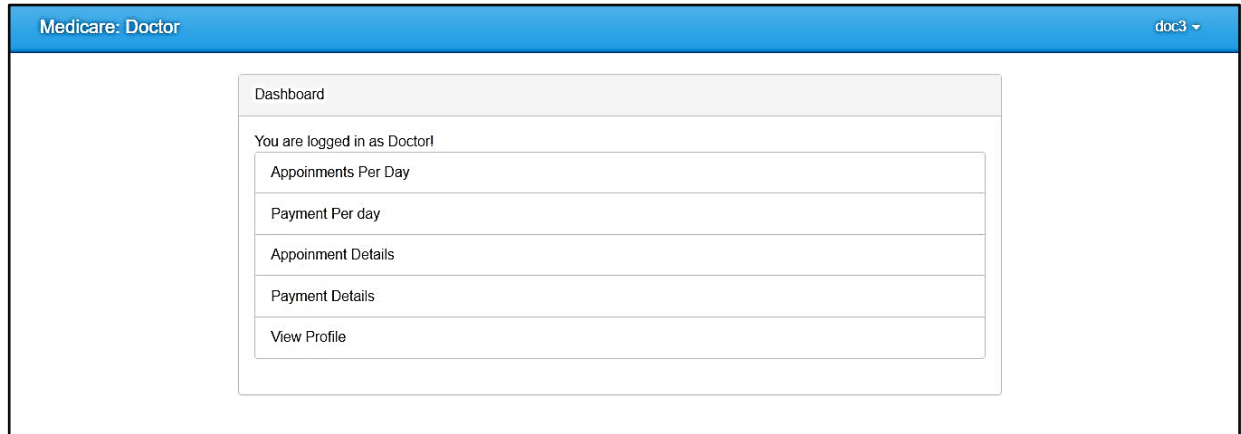


Figure:3.13 Dashboard for the doctors

3.8.10Dashboard for the Patient

system registered Patient having figure 3.14 interface in the system with tabs to Make Appointment, appointment history, edit profile and view profile contains at this dashboard

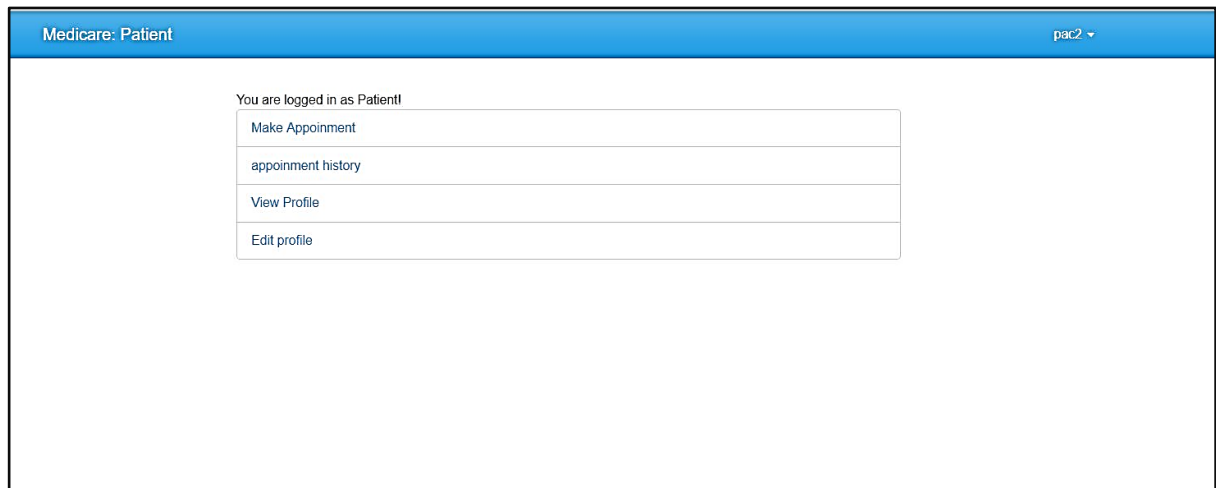


Figure:3.13 Dashboard for the Patient

Chapter 4: Implementation

4.1 Introduction

After completion of the design stage, implementation of the system begins according to what the design stage planned by using adequate tools and techniques. This is an important stage in the software development lifecycle, because this stage builds the system into a working system.

This chapter mainly discusses the implementation environment, the tools and techniques as well as the reusable components used to implement the system. The major code segments have been included to help the process functionalities of the system in this chapter. Code segmentations which were included in the system with a Comment are meant to be used for future improvements

A short description of software, Languages, tools and technologies used to develop the web portal has been indicated below.

4.2 Implementation Environment

The System Implementation Environment can be divided into two categories, Hardware Environment as well as Software Environment. The two major categories are listed below. Those are the minimum requirements for the implement the developed system.

Hardware & Software requirement for the system

Hardware Environment	Software Environment
Intel Core(TM) i5-2410M @ CPU 2.30 GHz	Microsoft Windows 7 or higher version
2GB RAM	PHP Version 5.4.6 or higher
500GB Hard Disk	MySQL Client & Apache
2GB VGA	Latest version of web browser
	Up to date software modules

Table 4.1: Hardware & Software requirement for the system

4.3 System Development Tools and Technologies

When developing the system, the following tools and technologies were used and the following bullet points briefly discuss them:

- Software Ideas Modeler.

Diagram designing had done by using this software.

- PHP

This is a server-side object-oriented scripting language which used when developing the system.

- MySQL

This is the Database software which used to handle the database of the system.

- HTML

This is the basic web-related language and it helps to keep the system structure clear and conscious.

- JavaScript

This is a Scripting language and this helps for client-side validation and to dynamic the system components.

- Ajax

Ajax is a JavaScript-based technology and it supports updating the system components without refreshing the whole system page.

- CSS

This is used for the styling of the system.

- JSON

“JSON is the syntax for storing and exchanging text information. Much like XML. JSON is smaller than XML and faster and easier to parse.” [06]

- jQuery

It is a JavaScript library and used as a reusable component when developing.

- Sublime Text2 (Text Editor).

4.4 Major Code Segments

The main code modules developed in the system have been mentioned below by Briefly describing their functionality.

4.4.1 Register Blade:

Every system user needs to register to the system before login. Patient registration handled by below code segments.


```

1 | @extends('patient.layout.auth')
2 |
3 | @section('content')
4 | <div class="container">
5 |     <div class="row">
6 |         <div class="col-md-8 col-md-offset-2">
7 |             <div class="panel panel-default">
8 |                 <div class="panel-heading">Register</div>
9 |                 <div class="panel-body">
10 |                    <form class="form-horizontal" role="form" method="POST" action="{{ url('/patient/register') }}">
11 |                        {{ csrf_field() }}
12 |
13 |                        <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
14 |                            <label for="name" class="col-md-4 control-label">Name</label>
15 |
16 |                            <div class="col-md-6">
17 |                                <input id="name" type="text" class="form-control" name="name" value="{{ old('name') }}" autofocus>
18 |
19 |                                @if ($errors->has('name'))
20 |                                    <span class="help-block">
21 |                                        <strong>{{ $errors->first('name') }}</strong>
22 |                                    </span>
23 |                                @endif
24 |                            </div>
25 |                        </div>
26 |
27 |                        <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
28 |                            <label for="email" class="col-md-4 control-label">E-Mail Address</label>
29 |
30 |                            <div class="col-md-6">
31 |                                <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}">
32 |
33 |                                @if ($errors->has('email'))
34 |                                    <span class="help-block">
35 |                                        <strong>{{ $errors->first('email') }}</strong>
36 |                                    </span>
37 |                                @endif
38 |                            </div>
39 |                        </div>
40 |
41 |                        <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
42 |                            <label for="password" class="col-md-4 control-label">Password</label>
43 |
44 |                            <div class="col-md-6">
45 |                                <input id="password" type="password" class="form-control" name="password">
46 |
47 |                                @if ($errors->has('password'))
48 |                                    <span class="help-block">
49 |                                        <strong>{{ $errors->first('password') }}</strong>
50 |                                    </span>
51 |                                @endif
52 |                            </div>
53 |                        </div>

```

```

54 |
55 |                    <div class="form-group{{ $errors->has('password_confirmation') ? ' has-error' : '' }}">
56 |                        <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
57 |
58 |                        <div class="col-md-6">
59 |                            <input id="password-confirm" type="password" class="form-control" name="password_confirmation">
60 |
61 |                            @if ($errors->has('password_confirmation'))
62 |                                <span class="help-block">
63 |                                    <strong>{{ $errors->first('password_confirmation') }}</strong>
64 |                                </span>
65 |                            @endif
66 |                        </div>
67 |                    </div>
68 |
69 |                    <div class="form-group">
70 |                        <div class="col-md-6 col-md-offset-4">
71 |                            <button type="submit" class="btn btn-primary">
72 |                                Register
73 |                            </button>
74 |                        </div>
75 |                    </div>
76 |                </form>
77 |            </div>
78 |        </div>
79 |    </div>
80 |</div>
81 |@endsection
82 |
83 |

```

4.4.2 Index Page

Index page of this portal handled by below code segments.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <!-- The above 3 meta tags must come first in the head; any other head content must come "after" these tags -->
8   <title>Medicare Medical Portal</title>
9
10  <!-- Bootstrap -->
11  <link href="css/bootstrap.min.css" rel="stylesheet">
12  <link href="css/styles.css" rel="stylesheet">
13
14  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
15  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
16  <!--[if lt IE 9]>
17    <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
18    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
19  <![endif]-->
20 </head>
21 <body>
22   <div class="main-wrapper">
23     <div class="container">
24       <div class="row">
25         <div class="col-md-6">
26           <div class="open-hours">Monday - Saturday, 8am to 10pm</div>
27         </div>
28         <div class="col-md-6">
29           <div class="phone-no">Call us now +94 112 655 886</div>
30         </div>
31       </div>
32     </div>
33   </div>
34   <div class="navbar navbar-default">
35     <div class="container">
36       <!-- Brand and toggle get grouped for better mobile display -->
37       <div class="navbar-header">
38         <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
39           data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
40           <span class="sr-only">Toggle navigation</span>
41           <span class="icon-bar"></span>
42           <span class="icon-bar"></span>
43           <span class="icon-bar"></span>
44         </button>
45         <a class="navbar-brand" href="#">Medicare Medical Portal</a>
46       </div>
47     </div>
48
49     <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
50       <ul class="nav navbar-nav">
51         <li class="active"><a href="#">Home <span class="sr-only">(current)</span></li>
52         <li><a href="#">About Us</li>
53         <li><a href="#">Contact Us</li>
54       </ul>
55     </div>
56   </div>
57   <!-- Container fluid -->
58   <div class="banner">
59     <div class="container">
60       <div class="row">
61         <div class="col-md-6">
62           <div class="channel-form">
63             <div class="channel-heading">Channel Doctors</div>
64             <form method="post" action="{url('/doctor/search')}">
65               @csrf
66               <div class="form-group">
67                 <label for="doctor-name">Doctor Name</label>
68                 <input type="text" class="form-control" name="doctor_name" id="doctor-name"
69                   placeholder="Doctor Name (Max 30 characters)">
70               </div>
71               <div class="form-group">
72                 <label for="speciality">Speciality - Description</label>
73                 <select id="speciality" type="speciality"
74                   class="form-control" name="speciality_id">
75                   <option value="">Select speciality</option>
76                   @if(count($specialitylist) > 0)
77                     @foreach($specialitylist as $item)
78                       <option value="{{ $item->id }}">{{ $item->specialty_name }}</option>
79                     @endforeach
80                   @endif
81                 </select>
82               </div>
83               <div class="form-group">
84                 <label for="date">Date</label>
85                 <input type="date" class="form-control" name="date" id="date" placeholder="Any">
86               </div>
87               <div class="text-right">
88                 <button type="submit" class="btn btn-default">SEARCH</button>
89               </div>
90             </form>
91           </div>
92         </div>
93       </div>
94       <div class="col-md-6">
95         <div class="row">
96           <div class="col-md-6">
97             
98           </div>
99           <div class="col-md-6">
100            <div class="login-register">
101              <!-- Nav tabs -->
102              <div class="nav nav-tabs">
103                <li role="presentation" class="active">
104                  <a href="#login" aria-controls="home" role="tab" data-toggle="tab">Login</a>
105                </li>
106                <li role="presentation">
107                  <a href="#register" aria-controls="profile" role="tab" data-toggle="tab">Register</a>
108                </li>
109              </div>
110              <!-- Tab panes -->
111              <div class="tab-content">
112                <div role="tabpanel" class="tab-pane active" id="login">
113                  <div style="margin-top: 5px; margin-bottom: 20px;">
114                    <div class="form-horizontal" role="form" method="POST" action="{url('/patient/login')}">
115                      @csrf
116                      <div class="form-group">
117                        <label for="exampleInputEmail1">{{ __('E-Mail Address') }}</label>
118                        <input type="email"
119                          class="form-control" {{ $errors->has('email') ? 'is-invalid' : '' }}
120                          name="email" id="exampleInputEmail1" placeholder="Email"
121                          value="{{ old('email') }}" required autofocus>
122                        @if ($errors->has('email'))
123                          <span class="invalid-feedback">
124                            <strong>{{ $errors->first('email') }}</strong>
125                          </span>
126                        @endif
127                      </div>
128                      <div class="form-group">
129                        <label for="exampleInputPassword1">{{ __('Password') }}</label>
130                        <input type="password" class="form-control" {{ $errors->has('password') ? 'is-invalid' : '' }}
131                          name="password" id="exampleInputPassword1"
132                          placeholder="Password" required>
133                        @if ($errors->has('password'))
134                          <span class="invalid-feedback">
135                            <strong>{{ $errors->first('password') }}</strong>
136                          </span>
137                        @endif
138                      </div>
139                      <div class="form-group row">
140                        <div class="col-md-8 offset-md-8">
141                          <div class="checkbox">
142                            <input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}> {{ __('Remember Me') }}
143                          </div>
144                        </div>
145                      </div>
146                    </div>
147                  </div>
148                </div>
149              </div>
150            </div>
151          </div>
152        </div>
153      </div>
154    </div>
155  </div>
```

4.4.3 Password Reset Function of patient

when the registered patient needs to reset password.it can be an archive from the password reset link. It was handled by the below-attached code segments.

```
1 | extends('patient.layout.auth')
2
3 @section('content')
4 <div class="container">
5     <div class="row">
6         <div class="col-md-8 col-md-offset-2">
7             <div class="panel panel-default">
8                 <div class="panel-heading">Reset Password</div>
9
10                <div class="panel-body">
11                    <form class="form-horizontal" role="form" method="POST" action="{{ url('/patient/password/reset') }}">
12                        {{ csrf_field() }}
13
14                        <input type="hidden" name="token" value="{{ $token }}">
15
16                        <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
17                            <label for="email" class="col-md-4 control-label">E-Mail Address</label>
18
19                            <div class="col-md-6">
20                                <input id="email" type="email" class="form-control" name="email" value="{{ $email or old('email') }}" autofocus>
21
22                                @if ($errors->has('email'))
23                                    <span class="help-block">
24                                        <strong>{{ $errors->first('email') }}</strong>
25                                    </span>
26                                @endif
27                            </div>
28                        </div>
29
30                        <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
31                            <label for="password" class="col-md-4 control-label">Password</label>
32
33                            <div class="col-md-6">
34                                <input id="password" type="password" class="form-control" name="password">
35
36                                @if ($errors->has('password'))
37                                    <span class="help-block">
38                                        <strong>{{ $errors->first('password') }}</strong>
39                                    </span>
40                                @endif
41                            </div>
42                        </div>
43
44                        <div class="form-group{{ $errors->has('password_confirmation') ? ' has-error' : '' }}">
45                            <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
46
47                            <div class="col-md-6">
48                                <input id="password-confirm" type="password" class="form-control" name="password_confirmation">
49
50                                @if ($errors->has('password_confirmation'))
51                                    <span class="help-block">
52                                        <strong>{{ $errors->first('password_confirmation') }}</strong>
53                                    </span>
54                                @endif
55                            </div>
56                        </div>
57                    </form>
58                </div>
59            </div>
60        </div>
61    </div>
62</div>
```

4.4.4 Public controller.

Public search of the doctor appointment details controlled by below code segments.

```
1 | <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Doctor;
6 use App\Specialty;
7 use App\DoctorAvailability;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\DB;
10 use Illuminate\Support\Facades\Input;
11
12 class PublicController extends Controller
13 {
14     public function index()
15     {
16         $specialty = new Specialty;
17         $specialtyList = $specialty->all();
18         return view('index')->withSpecialtyList($specialtyList);
19     }
20
21     public function search(Request $request)
22     {
23         $doctors = new Doctor();
24         $doctors = $doctors->select('doctors.id as d_id', 'doctors.name as d_name', 'specialties.*', 'specialties.id as s_id')
25             ->join('specialties', 'doctors.specialty', 'specialties.id')
26             ->with('DoctorAvailability');
27
28         if($request->input("specialty_id") != ''){
29             $doctors->where('doctors.specialty', '=', $request->input("specialty_id"));
30         }
31
32         if($request->input('doctor_name') != ''){
33             $doctors->where('doctors.name', 'like', "%".$request->input('doctor_name')."%");
34         }
35
36         if($request->input('date') != ''){
37             $doctors->join('doctor_availabilities', 'doctors.id', 'doctor_availabilities.doctor_id')
38             ->where('doctor_availabilities.date', $request->input('date'));
39         }
40
41         $doctors = $doctors->get();
42
43         $date = $request->input('date');
44         return view('public.available_doctors')->withDoctors($doctors)->withDate($date);
45     }
46 }
47
48
49
50
51
```

```

50
51
52 public function selectDoctor(){
53
54     $inputs = Input::get();
55     if($inputs['doctor_id'] != ""){
56         $doctor = new Doctor();
57         $doctor = $doctor::join('specialties', 'doctors.specialty', 'specialties.id')
58             ->where('doctors.id', $inputs['doctor_id']->get());
59
60         $availability = DoctorAvailability::where('doctor_id', $inputs['doctor_id'])
61             ->with('appointments');
62         if($inputs['date'] != ""){
63             $availability->where('date', $inputs['date']);
64         }
65         $availability = $availability->get();
66
67         return view('public.doctor_availability')->withDoctor($doctor)->withAvailability($availability);
68     }else{
69         return redirect()->back();
70     }
71 }
72
73
74
75
76
77
78

```

4.4.5 User Table Migration

User table migration handled by below-attached code segments.

```

1 k?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->string('password');
21             $table->rememberToken();
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33         Schema::dropIfExists('users');
34     }
35 }
36

```

4.4.6 Main routes of the system

Main routes of the system handled by below attached code segments.


```

1 |<?php
2
3 |Route::get('/home', function () {
4 |    return view('receptionist.home');
5 |})->name('home');
6
7 |Route::get('/', function () {
8 |    return view('receptionist.home');
9 |});
10
11 |Route::get('addpatient', function(){
12 |    return view('receptionist.addpatient');
13 |});
14 |Route::post('addpatient', 'PatientController@newPatient');
15
16
17
18 |Route::post('adddoctor', 'DoctorController@addDoctor');
19 |Route::post('adddoctor', 'DoctorController@newDoctor');
20
21 |Route::get('addreceptionist', function(){
22 |    return view('receptionist.addreceptionist');
23 |});
24
25 |Route::get('makeappointment', function(){
26 |    return view('receptionist.makeappointment');
27 |});
28 |Route::get('makeappointment', 'AppointmentController@showmakeanappointmentreceptionistform');
29
30 |Route::post('doctorssearch', 'AppointmentController@doctorSearch');
31
32
33 |Route::post('addreceptionist', 'ReceptionistController@newReceptionist');
34
35 |Route::get('adddoctor', 'ReceptionistController@addDoctor');

```

4.4.7 Contact Us Page

Contact Us Page was controlled by below-attached code segment.

```

1 |<!DOCTYPE html>
2 |<html lang="en">
3 |<head>
4 |    <meta charset="utf-8">
5 |    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 |    <meta name="viewport" content="width=device-width, initial-scale=1">
7 |    <!-- The above 3 meta tags *must* come first in the head, any other head content must come "after" these tags -->
8 |    <title>Medicare Medical Portal</title>
9 |
10 |    <!-- Bootstrap -->
11 |    <link href="css/bootstrap.min.css" rel="stylesheet">
12 |    <link href="css/styles.css" rel="stylesheet">
13 |
14 |    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
15 |    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
16 |    <!--[if lt IE 9]>
17 |    <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
18 |    <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
19 |    <[/if]>
20 |</head>
21 |<body>
22 |    <div class="main-wrapper">
23 |        <div class="top-header">
24 |            <div class="container">
25 |                <div class="row">
26 |                    <div class="col-md-6">
27 |                        <div class="open-hours">Monday - Saturday, 8am to 10pm</div>
28 |                    </div>
29 |                    <div class="col-md-6">
30 |                        <div class="phone-no">Call us now +94 112 655 886</div>
31 |                    </div>
32 |                </div>
33 |            </div>
34 |
35 |            <nav class="navbar navbar-default">
36 |                <div class="container">
37 |                    <!-- Brand and toggle get grouped for better mobile display -->
38 |                    <div class="navbar-header">
39 |                        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
40 |                            <span class="sr-only">Toggle navigation</span>
41 |                            <span class="icon-bar"></span>
42 |                            <span class="icon-bar"></span>
43 |                            <span class="icon-bar"></span>
44 |                        </button>
45 |                        <a class="navbar-brand" href="#">Medicare Medical Portal</a>
46 |                    </div>
47 |
48 |                    <!-- Collect the nav links, forms, and other content for toggling -->
49 |                    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
50 |                        <ul class="nav navbar-nav">
51 |                            <li class="active"><a href="/">Home <span class="sr-only">(current)</span></a></li>
52 |                            <li><a href="#">About Us</a></li>
53 |                            <li><a href="/contact">Contact Us</a></li>

```

```

52         <li><a href="#">About Us</a></li>
53         <li><a href="/contact">Contact Us</a></li>
54     </ul>
55 </div><!-- /.navbar-collapse -->
56 </div><!-- /.container-fluid -->
57
58 <div class="banner">
59     <div class="container">
60         <div class="row">
61             <div class="col-md-8" style="padding-right: 0;">
62                 <div class="contact-form">
63                     <h4 class="contact-heading">Contact Us</h4>
64                     <form>
65                         <div class="form-group">
66                             <label for="cont-email">Email</label>
67                             <input type="email" class="form-control" id="cont-email">
68                         </div>
69                         <div class="form-group">
70                             <label for="subject">Subject</label>
71                             <input type="email" class="form-control" id="subject">
72                         </div>
73                         <div class="form-group">
74                             <label for="message">Message</label>
75                             <textarea name="" id="message" cols="30" rows="1" class="form-control"></textarea>
76                         </div>
77                         <div class="text-right">
78                             <button type="submit" class="btn btn-default">Send Message</button>
79                         </div>
80                     </form>
81                 </div>
82             <div class="col-md-4 map-container" style="padding-left: 0;">
83                 
84             </div>
85         </div>
86     </div>
87 </div>
88 <div class="inner-body">
89     <div class="container">
90         <div class="list-inline list-unstyled">
91             <li style="margin-right: 20px;">
92                 
93                 <span>30, Galle Rd, Moratuwa</span>
94             </li>
95             <li>
96                 
97                 <span>0112 655 886</span>
98             </li>
99         </div>
100     </div>
101 </div>
102 </div>
103 </div>
104

```

4.4.8 Appointments Controller

Doctor search and the appointments controlled by below-attached code segments.

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Specialty;
6  use App\DoctorAvailability;
7  use App\Doctor;
8  use App\Appointments;
9
10 use Illuminate\Http\Request;
11
12 class AppointmentController extends Controller
13 {
14     public function showmakeanappointmentpatientform()
15     {
16         $specialty = new Specialty;
17         $specialtyList = $specialty->all();
18         return view('patient.makeanappointment')->withSpecialtyList($specialtyList);
19     }
20
21     public function showmakeanappointmentreceptionistform()
22     {
23         $specialty = new Specialty;
24         $specialtyList = $specialty->all();
25         return view('receptionist.makeanappointment')->withSpecialtyList($specialtyList);
26     }
27
28     public function doctorSearch(Request $request)
29     {
30         $doctors = new Doctor();
31
32         $doctors = $doctors->join('specialties', 'doctors.specialty' , 'specialties.id')
33         ->with('DoctorAvailability');
34
35         if($request->specialty_id != ''){
36             $doctors->where('doctors.specialty', $request->specialty_id);
37         }
38
39         if($request->input('doctor_name') != ''){
40             $doctors->where('doctors.name', 'like', "%".$request->input('doctor_name')."%");
41         }
42
43         $doctors = $doctors->get();
44
45         return view('receptionist.available_doctors')->withDoctors($doctors);
46     }
47
48 }
49
50

```

4.4.9 Patient Login Controller

Patient Login controlled by below-attached code segments.

```
1 k?php
2
3 namespace App\Http\Controllers\PatientAuth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7 use Illuminate\Support\Facades\Auth;
8 use Hesto\MultiAuth\Traits\LogoutGuard;
9
10 class LoginController extends Controller
11 {
12     /**
13      * Login Controller
14      *
15      * This controller handles authenticating users for the application and
16      * redirecting them to your home screen. The controller uses a trait
17      * to conveniently provide its functionality to your applications.
18      */
19
20     use AuthenticatesUsers, LogoutGuard {
21         LogoutGuard::logout insteadof AuthenticatesUsers;
22     }
23
24     /**
25      * Where to redirect users after login / registration.
26      *
27      * @var string
28      */
29     public $redirectTo = '/patient/home';
30
31     /**
32      * Create a new controller instance.
33      *
34      * @return void
35      */
36     public function __construct()
37     {
38         $this->middleware('patient.guest', ['except' => 'logout']);
39     }
40
41     /**
42      * Show the application's login form.
43      *
44      * @return \Illuminate\Http\Response
45      */
46     public function showLoginForm()
47     {
48         return view('patient.auth.login');
49     }
50
51 }
52
53
```

4.4.10 Patient Forget Password Controller

Patient Forget password will be controlled by below-attached code segments.

```
1 k?php
2
3 namespace App\Http\Controllers\PatientAuth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\SendsPasswordResetEmails;
7 use Illuminate\Support\Facades\Password;
8
9 class ForgotPasswordController extends Controller
10 {
11     /**
12      * Password Reset Controller
13      *
14      * This controller is responsible for handling password reset emails and
15      * includes a trait which assists in sending these notifications from
16      * your application to your users. Feel free to explore this trait.
17      */
18
19     use SendsPasswordResetEmails;
20
21     /**
22      * Create a new controller instance.
23      *
24      * @return void
25      */
26     public function __construct()
27     {
28         $this->middleware('patient.guest');
29     }
30
31     /**
32      * Display the form to request a password reset link.
33      *
34      * @return \Illuminate\Http\Response
35      */
36     public function showLinkRequestForm()
37     {
38         return view('patient.auth.passwords.email');
39     }
40
41     /**
42      * Get the broker to be used during password reset.
43      *
44      * @return \Illuminate\Contracts\Auth\PasswordBroker
45      */
46     public function broker()
47     {
48         return Password::broker('patients');
49     }
50
51 }
52
53
```

CHAPTER 05: EVALUATION

5.1 User Evaluation & system testing

Evaluation is a systematic determination of a subject's merit, worth and significance, using criteria governed by a set of standards. It can assist an organization, program, project or any other intervention or initiative to assess any aim, realizable concept/proposal, or any alternative, to help in decision-making; or to ascertain the degree of achievement or value in regard to the aim and objectives and results of any such action that has been completed. The primary purpose of evaluation, in addition to gaining insight into prior or existing initiatives, is to enable reflection and assist in the identification of future change

This project would test and evaluation was done using a selected set of system users before releasing the final product. system hosted at the free hosting site and link to this temporarily hosted site given to the selected list of peoples with the simple questions.

Main testing idea gain through evaluation of those answers

5.2 System Test Plan

Planning a test case is a very important aspect for developing the system as well as for the Completed system. The test plan should have the ability to test the functionality of the Overall system. By properly testing a system, it can identify the errors which generate from the system and can correct them. The implemented system was tested using different test cases.

Since the development started, the test plan continued by testing the system units.

After completing a system unit, it was completely tested to identify whether it can function according to expectations. Therefore, early detection of the errors was helped by this testing stage.

After performing the system unit testing, next integration testing was done, and this can identify the errors and the required functionality of the units after integration.

System testing was performed as the final stage for the completely developed system to check the functionality.

Software Testing

It's a process of validating and verifying the system in order to identify any gaps, errors or missing requirements in line with actual requirements. Validation is the process of evaluating the final product to check whether the software meets the business needs. Verification is a

process of evaluating the intermediary work products of a software development lifecycle to check if we are on the right track of creating the final product

5.3 System Test Cases

A properly planned test case should have the ability to verify the relevant system component functionality. Therefore, to verify all the system functions there should be properly planned test cases for each and every function. Some test cases designed for major system modules are as follows

5.3.1 Test Case for User Authentication.

The test cases and results of the User Authentication Module are displayed by below table.

Test No	Test Description	Steps to Test	Expected Result	Success
1	Validate User Input Details	Enter correct username Enter correct password Enable account status	Successfully Login into the system	yes
2	Validate User Input Details.	Enter correct username Enter incorrect password Enable account status	Display error message “Incorrect username or password”	yes
3	Validate User Input Details.	Enter incorrect username Enter correct password Enable account status	Display error message “Incorrect username or password”	yes
4	Validate User Input Details.	Enter both username and password incorrectly Enable account status	Display error message “Incorrect username or password”	yes

Table 5.1 User Authentication Test cases

5.3.2 Test Case for Doctor search.

The test cases and results of the Add Product Module are displayed at the below table.

Test Number	Test Description	Steps to Test	Expected Result	Success
1	Search by speciality	Select speciality and click search	Display all the doctors which registered to the selected speciality.	yes
2	Search by Name	Enter patient of name and click search. (without select speciality)	Display all the registered doctors which contain that name.	yes
2	Search by Date	Enter patient of name speciality with date and click search	Display all the registered doctors which contain that name with selected date.	yes
3	Search by Name with Speciality	Enter part of name with speciality and click search.	Display all the registered doctors which contain that name with required speciality.	yes
4	Search not registered doctor	Enter part of name with speciality and click search.	Showing the message of "No registered doctors found"	yes

Table 5.2 Doctor Search Test cases

5.3.3 Test case for managing users

Test Number	Test Description	Steps to Test	Expected Result	Success
1	Validate Personnel Information adding	Add Information with relevant details	Display Message "Successfully inserted User details" with his name	yes
2	Validate Personnel Information adding	Try to add information by keeping all fields in empty	Popup the error message of the first field.	yes
3	Validate Personnel Information adding	Enter all fields without Name correctly.	Popup the error message "Invalid username"	yes
4	Validate Personnel Information adding	Enter all fields without the @sign for Email address .	Popup the error message "missing the @ sign"	yes

5	Validate Personnel Information adding	Enable relevant information fields according to User type	Enable relevant fields by highlighting fields.	yes
6	Validate Personnel Information adding	Enter Incorrect NIC Number	Popup the error message "Invalid NIC Number"	yes
7	Validate Personnel Information adding	Enter existing NIC Number	Display Message with Existing user's name as an Error Message.	yes
8	Validate Create Account	Enter incorrect NIC	Display Message" No User for this Username"	yes
9	Validate Create Account	Enter correct NIC	Display Message with relevant user's first name with the last name and appear Job in the job field	yes

Table 5.2 Manage Users Test cases

Chapter 6: Conclusion and future work

6.1 Future works

created this portal without payment gateway for online channelling. Online payment can be integrating into this system by purchasing payment gateway. this portal design for only one medical centre. It can be introduce many more such medical institutions to this system and enhance the system access.

6.2 Lessons Learned through this project

This final project helps me to build a fully functional system which improves my programming skills. It gives me thorough knowledge and experience in developing a system within both theoretical and practical aspects. It also teaches me to work according to a proper schedule, systematically. Thereby I understand the importance of managing the time to submit the reports, a dissertation on time and experienced to work under pressure. Herewith listed some of the knowledge I gain through this project.

- ✓ Learn to analyse and draw use case diagrams, sequence diagrams, activity diagrams, class diagrams and ER diagrams for a real system.
- ✓ Learn to pre-design user interfaces with the help of free wireframe design tool.
- ✓ Learn to create databases and related functions with phpMyAdmin which improve my

knowledge of databases.

- ✓ Learn to code the system with the help of Laravel framework 4.2 versions which improve my programming skills.
- ✓ Learn that a template can be taken to design the webpage attractively and used AdminLTE template.
- ✓ Learn that by using an IDE it helps to minimize compatibility issues. Therefore used Sublime Text2.
- ✓ Learn to design test cases and test plans during the evaluation phase.

The Reference List

Online Sources

- [01] Open Source EMR [Online]. Available: <http://www.open-emr.org/>
- [02] Echannling [Online]. Available: <https://www.echannelling.com/Echannelling/index>
- [03] Asiri Hospital.com [Online]. Available: <http://www.asirihospitals.com/>
- [04] Requirement Analysis Definition[Online]
Available:<http://searchsoftwarequality.techtarget.com/definition/requirements-analysis>
(01-08-2017)
- [05] Wikipedia Website [online].[http://en.wikipedia.org/wiki/Functional requirement](http://en.wikipedia.org/wiki/Functional_requirement)
(01-08-2017)
- [06] W3Schools Website. <http://www.w3schools.com/json/> (05-11-2017)
- [07] Wikipedia Website <http://en.wikipedia.org/wiki/Evaluation> (04-11-2017)

Appendix A – System Documentation

This documentation consists of a set of steps to show, how to install this System. These steps explain the hardware and software environment which needs to be installed. When installing the system, this documentation can be followed by the interested parties.

1st Step

□ System requirements can be verified according to Table A.1 and Table A.2.

Hardware Requirements

Hardware	Minimum Requirements
Processor	3.0 GHz Intel Processor
Memory	512 MB Memory Capacity
Hard Disk	500GB Capacity
Display	1024*768 Resolution Monitor.
Printer	Inkjet Printer or LaserJet Printer
Internet	ADSL Connection (Minimum Speed 16Mbps)

Table A.1 Hardware Requirements

Software Requirements

Software	Minimum Requirements
Operating System	Windows XP, Windows 7, Windows 8 , Windows10 O/S
XAMPP	XAMPP Version (1.8.2) or Separate Installation of Apache 2.4.4 / MySQL 5.5.32 / PHP 5.4.16 / phpMyAdmin 4.0.4 /Mercury Mail Transport System v4.62
Web Browser	Google chrome(Version 28.0.1500.72 m), Firefox(Version 21.0) Web browser.
Image Editor	Adobe Photoshop CS4 (Need for future Developments)
Code Editor	Sublime Text2 (Need for future Developments)

Table A.2 Software Requirements

2nd Step

Installing XAMPP

- Download and install XAMPP for Windows (refer Table A.2 for the Minimum Version) from <http://www.apachefriends.org>. Give installation path to C:\xampp of the computer.

Installing Web Browsers

- Install Browsers (refer Table A.2 for the Version and Recommended Browsers).

3rd Step

Files Extraction

- Open the CD and copy the OOMS folder and paste it into the directory path “C:\xampp\htdocs”

4th Step

Database Installation

- Open the web browser and type the URL <http://localhost/phpmyadmin/> and enter Username and Password (if you set username and password).
- Create an empty database by providing the name as “**Medicare**” and navigate to the “**Import**” tab and click “**choose file**” button. Then browse the CD and select the “**medicare_db.sql**” file by opening Database folder.
- Then Press “**GO**” button located at the bottom of the page.

Final Step (Launching System)

- Verify the XAMPP is running, go to the “C:\xampp\” and open the control panel and verify whether Apache, PHP, MySQL, Mercury is running.
- Open the installed web browser and type the URL <http://localhost/OOMS> and press “Enter” button to access the system.
- Please refer Appendix-C User Documentation to get the idea about how to operate the system.

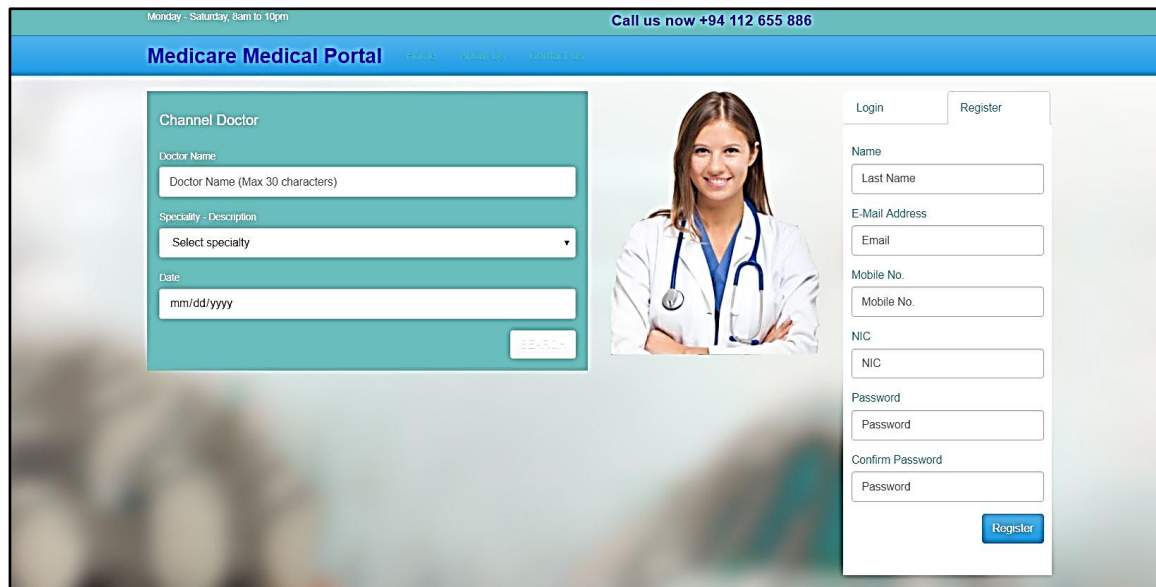
Appendix C - User Documentation

Users who have privileges to access the system about how to navigate within the system and how to use its functions are helped by this document. Eligibility to access the system is available to the factory manager, accountant, administrator, wholesale dealer, raw material supplier and sales representative.

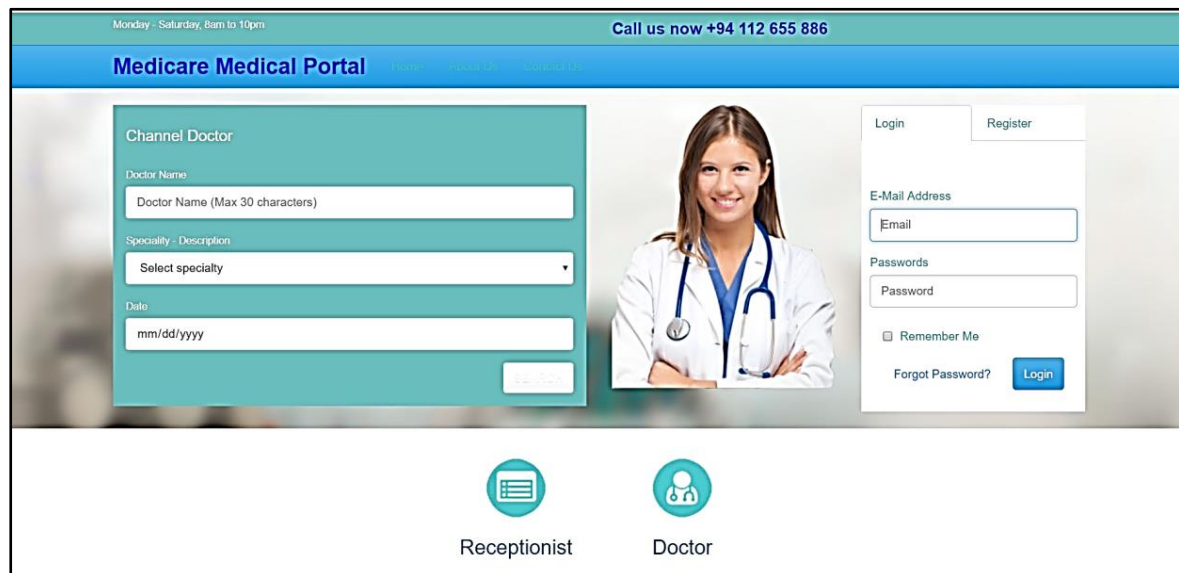
Different user levels are available to each and every member and different profile pages are provided for their management. The administrator's profile which has all the system privileges is described by this document.

Registration page, Login Page and other functionalities

Open the web browser that configured in the installation. Type the UR<http://localhost/Medicare> and navigate. After navigation, the user can find the initial welcome page that is under figure C.1. and user needs to fill the fields and click submit. Email address and password are the compulsory fields to fill. Other fields are optional. Registered users can login to the system using login screen showing under figure C.2



FigureC.1 user registration page.



Monday - Saturday, 8am to 10pm Call us now +94 112 655 886

Medicare Medical Portal


Channel Doctor

Doctor Name

Speciality - Description

Date

Register




Login Register

E-Mail Address


Passwords

☐ Remember Me

Forgot Password? Login



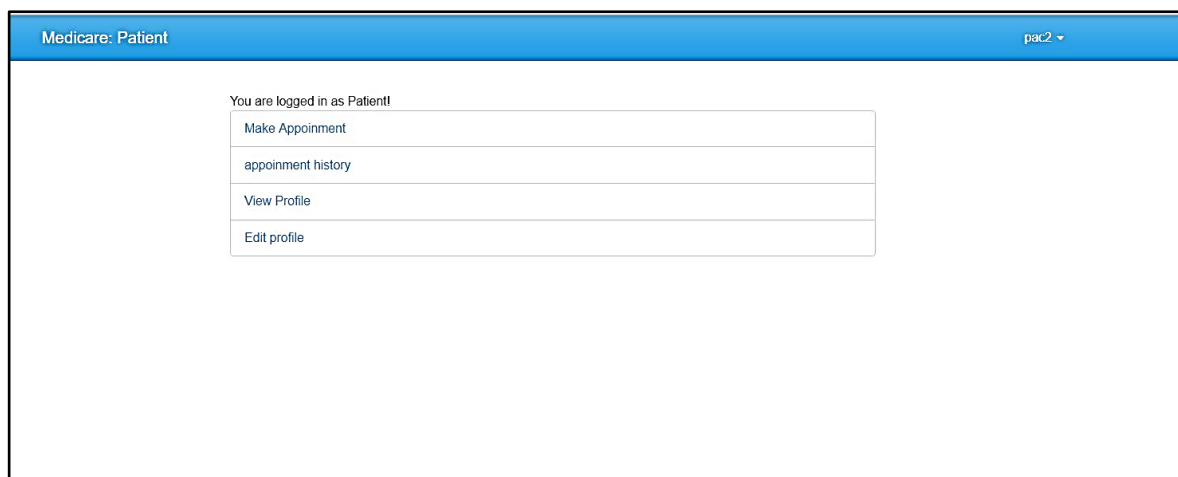
Receptionist



Doctor

FigureC.2 user Login page.

Dashboard for the logged user showing at figure C.3



Medicare: Patient pac2 ▾

You are logged in as Patient!

Make Appointment
appointment history
View Profile
Edit profile

FigureC.3 Dashboard for logged user.

After login to the system user have to click make appointment tab to put the doctor appointment through system.