Medicare Center Management System

for

Habarakada Medicare Center

R.A.L.U. Rupasingha

2021

Medicare Center Management System

for

Habarakada Medicare Center

A dissertation submitted for the Degree of Master of Information Technology

R.A.L.U. Rupasingha
University of Colombo School of Computing
2021

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: R.A.L.U. Rupasingha
Registration Number: 2018MIT070
Index Number: 18550702

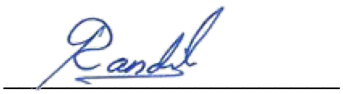
Signature: Date: 28-11-2021

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

Ms. R.A.L.U. Rupasingha

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:
Supervisor Name:



Signature: Date: 30 - 11 - 2021

Abstract

Habarakada Medicare Centre (HMC) is one of the well-established rather a patient-based regional medicare center treating outpatients. For 10 years, all the functionalities of the medicare center have been maintained by a traditional manual system.

The ultimate aim of the Medicare Center Management System is to automate the existing traditional manual systems with the support of current technology. In places with a huge patient base, patients have to stay in long queues, plenty of time is spent waiting for the doctor's consultation, getting medicine from the pharmacy, and paying the bill. Moreover, both the doctor and the patient are disadvantaged due to the absence of channeling history with details of previously prescribed medicines and the patient's medical records. The pharmacist is also unaware of the stockout or overstocking issue regarding pharmaceutical drugs. Thus, the Medicare Center Management System can be a solution to resolve these issues. After the installation of such a system, the patient can get an appointment online. In addition, the time wasted on queues, unavailability of electronic access to patient's treatment history, and improper medicine stock maintenance will be diminished by the enablement of the Medicare Center Management System for Habarakada Medicare Centre. The system can manage the patients' registration, online appointments, keeping details of the prescription and medical records, medicine details, and billing information which the medicare center administration can rely upon. The system is developed using the CodeIgniter framework using PHP, MySQL, and Bootstrap.

It is expected that after implementing the system, it will enhance the quality of service in order to generate a better patient experience. This system is user-friendly and flexible for users to deliver real conceivable benefits to Habarakada Medicare Centre.

Acknowledgment

First and foremost, I would like to thank the academic/non-academic staff of the University of Colombo School of Computing for the support and guidance provided throughout the Master's Program and also the Lecture panel of the Master's program of UCSC for guidance and support.

Special thanks go to the supervisor Dr. B H R Pushpananda, for the advice and guidance given throughout the project. I am also really thankful for the help and guidance provided by project coordinator Dr. (Ms.) H N D Thilini.

Last but not least, I would like to thank all my family members and friends who were with me throughout this project, giving me strength and never letting me go without advice and support whenever required.

Table of Contents

Table of Figures

List of Tables

List of Acronyms

CSS	Cascading Style Sheets
ER	Entity Relationship
GB	Gigabyte
GHz	Gigahertz
HMC	Habarakada Medicare Center
HTML	Hypertext Markup Language
MCMS	Medicare Center Managements System
MVC	Model View Controller
MYSQL	Structured Query Language
PHP	Hypertext Preprocessor
RAM	Random Access Memory
RDBMS	Relational Database Management System
SSD	Solid-state drive UI User interface
UI	User Interface
UML	Unified Modeling Language
UX	User Experience
WAMP	Windows, Apache, MySQL, and PHP

Chapter – 01 Introduction

1.1. Project Overview

The medicare centers, a facility that specializes in outpatient care; are operated both by the government and private sectors. Many Medicare centers are primarily concerned with providing primary care services to patients in the surrounding area. Habarakada Medicare Centre (HMC) is one of the more patient-centered regional medicare centers that treat outpatients. However, for the past ten years, all of the functions of this medicare center have been maintained using a traditional method. The Medicare Center Management System (MCMS) is being developed for Habarakada Medicare Center as a solution to the deficiencies of the current system.

The Medicare Center Management System (MCMS) for Habarakada Medicare Centre (HMC), of which the primary goal of development is to automate the current manual system of the medicare center, is a web-based system. The MCMS is designed to address a wide range of medicare center management processes which are patient management, doctor management, online appointment management, invoice/bill management, pharmacy management, and medical services, etc.

Registration of patient information is the system's initial and most significant function. Thus, according to the doctor's availability, an appointment to consult the respective doctor can be made either online or by going to the center in person. The next task is to select patients and provide medication advice. While this process is being completed, the doctor will be adding pertinent prescriptions and medical reports to pertinent patients. Another useful function is controlling the medicine stock by registering medicines. This will enable the possibility of maintaining the desired medicine stock without going out of stock or overstock as the reorder-level alerts and management reports will be considered when maintaining the medicine stock. The next function is to generate a more precise automated bill/invoice. These are the significant tasks of MCMS.

Through the installation of MCMS, the difficulty of manual maintenance of records, waste of time, rework, paperwork and the burden of file storage can be reduced. The computerization/ automation of the medicare center management system will help enhance

patients' and the medicare Staff's satisfaction with the most effective and efficient service. After implementing the MCMS, the quality of the medicare center will be upgraded to one of the best of its kind.

1.2. Motivation

In Sri Lanka, most of the medicare centers use a manual system to manage the functionalities of the medicare centers. The Habarakada Medicare Center (HMC) is also operated as a manual system. Thus, in the current system, a patient ticket or a number is issued at the counter after the patient comes in person. Medication and treatment will be prescribed by the doctor when the patient's turn comes. After that, that patient takes medicine from the pharmacist of the Habarakada Medicare Center. In this center, issuing patient numbers, billing system, managing medicine stocks, and every functionality is done manually. The current system is very time-consuming and ineffective.

The current system has many problems due to it being a manual system. There is no proper doctor reserved system therefore, the patients have to wait for a long time until their turn comes. It is a time-consuming task.

There is no patient registering system therefore, the doctor and the patient are struggling with the history of channeling. Most of the time doctors want to refer to the previous prescription but the patient may not have it and also the patient may not have their previous medical reports either. If there is any automated system to record a patient's medical history, it will be more efficient and worthwhile for both patient and doctor (Chandresh Prasad, 2017)

In the payment counter if there is no automated billing system then both parties involved in the transaction have to face accuracy problems with the bill.

Since in the pharmacy counter there is no stock management system, then most of the time the pharmacist has to face the lacking medicine stock problem.

Due to these problems, the quality of the Habarakada Medicare Center is going down therefore the current system should be transformed to automate the procedures.

1.3. Objectives

Following are the objectives of the proposed web-based system,

- Facilitate to reduce time-consuming, paper works of the patients and the staff of medicare center
- Digitalize the medical records (the doctor prescription and lab reports)
  By digitalizing the medical records,
    - Retrieving the doctor prescription and lab reports
    - Facilitate to eliminate the data misplacements of patients
- Facilitate to reduce errors in processes which are the doctor channeling, the billing, and pharmacy management
- Accurate data retrieval from the database
    - Checking the patients' profiles, checking doctor availability via doctor schedule, and checking available drugs
    - Generating the various type of reports accurately
        - The patient's profiles report
        - The reserved patient's information report (Count)
        - The current drugs stock report
        - The lacking drugs report
- Maintain the best pharmacy management system without any lack of drugs stock
- Maintain the accurate invoice/ bill management system
- Facilitate to increase data security (the patients' medical records)
- Provide a better patient experience by the implementation of the web-based system

1.4. Background of the Study

The medicare center management system (MCMS) is integrated software that handles the various clinical processes of the medicare centers. It oversees the smooth operation of healthcare as well as administrative, medical, legal, and financial controls. The MCMSs could be a unique system for certain institutions, chains of clinics, state hospitals, or even international medical organizations. The existing local and international systems that are similar to the one being considered are discussed in the background chapter.

The MCMS enhances the efficiency of health service delivery, minimizing paper waste, improving patient care, and giving better information to patients and doctors around the world. (Pollak and Lorch, 2007). The basic features of other existing systems and the newly added features of this proposed system are described in Table 1.1: Features Comparison.

Table 1.1: Features Comparison

Basic Features of other existing MCMS	**Newly added features of the MCMS of HMC**
Management of Patients: Organize and handle all inpatients and outpatients.	Computerized patient profile
Medical Records: Keep records of almost everything, including consultations, vital signs, lab findings, and image analysis.	Make appointments via the web: there is an option to view past medical records also
Medical Forms: Use a form template in all departments to save the time of processing data.	Storing the patients' medical records (prescription and medical reports).
Schedules: Keep track of patient appointments, Medic Waiting Lists, and Inpatient Schedules.	Computerized pharmacy management with reorder level management
Medical Billing and Claims: Extract invoices from medical records to create invoices.	Computerized Invoice/ bill management and automated reports generating

Both inpatients and outpatients are treated in most medicare centers that have MCMSs. However, this MCMS has been developed to treat outpatients only. The MCMS function's general purpose is to collect, manage, and use data to enhance the health care and medical services of the medicare center. The MCMS should be very effective and patient-centric.

1.5. Scope of the study

The Medicare Centre Management System facilitates the development and improvement of the medicare center's efficiency and operational quality.

This proposed system is based on a web-based management system. This system handles a lot of secured data of patients therefore, “It is possible to have high-level information security for the system to have the right information to the right person at the right time to the right place” (Bourgeois and Bourgeois, 2014). Because of sensitive information of patients in the medicare center the high-level protection requirements and needs are required for security. Furthermore, a web-based MCMS improves information and resource availability, accessibility, and usability.

The doctor, receptionist, pharmacy counter clerk, and customer are the system users. The related MCMS modules describe under in scope.

1.5.1. In-Scope

- The perspective of the doctor
    - Recording prescription and providing it to the patient

      Creating a comprehensive database to record the medical records of the registered patients as an image of the medical report and the prescription. These digitalized medical reports can effectively be used by the system users to avoid misplacement of the medical reports.

    - Retrieve the patient records by doctor

      The existing patients’ new doctor channeling details can be searched and added. The patient’s medical records access will be made available to the doctor and because the sensitive data in the Medicare Center Management System is secure, it does not harm the privacy and confidentiality of the patient.

- The perspective of the receptionist
    - Registering a patient as the initial process

    The initial registration will be required for further processes such as for the management of patients, management of patients' appointments, setting up of up the doctor availability, and attending to queue handling.

- The perspective of the pharmacy counter clerk
    - Automated the pharmacy process and the related report generation
        - Managing the medicine stock without having lack of medicine stock level
        - Registering medicine stock
        - Managing to reorder level of medicine stock
        - Generating reports to identify the stock level.

    - Automated the billing process and the related report generation

    The pharmacy counter clerk handles pharmacy processes and invoice/billing processes.

- The perspective of the patient
    - Check the doctor's availability and create an appointment for doctor channeling. The patient has the privilege of making the doctor's appointment via a web-based system and view their medical records. The Registered patients can check their past prescriptions and medical reports.

1.5.2. Out-Scope

- Due to practical scenarios, patients are not allowed to make payments online
- The appointments cannot be canceled by the patient over the internet.
- Ordering Drugs from suppliers can be added as future developments
- Keeping track of supplier details also can be added as future developments. (Not included in this stage)
- No Medicare Center Network (can be expanded as future developments)
- Only one doctor and two staff members are working here.

1.6. Feasibility Study

A feasibility study is intended to determine whether or not the proposed system should be implemented by determining whether or not the system is practical and feasible (Kenton, 2021). For this system, the feasibility study is done by discussing and interviewing with the owner of Habarakada Medicare Centre.

- Technical Feasibility

  This entails determining the proper software and hardware to satisfy the user's requirements. In this project, compatibility is fundamental. PHP and MYSQL SERVER work well together. The output speed is great, the response time is getting quicker, and the result is delivered almost instantly. PHP was chosen for its ease of use, well-defined database interface, and well-defined MYSQL SERVER handshaking, as well as its debugging capabilities. MYSQL SERVER was chosen for its capacity to manage large amounts of data, as well as its security, solid RDBMS, and backup and recovery capabilities.
- Operational Feasibility

  At present all of the operations are done manually. As a result, throughput and reaction time is excessive. After the web-based system is introduced, the manual operations must be eliminated and the web-based system with all essential hardware must be established. It will assist in removing all of the manual system's overhead. The system is operationally practical since it is simple to use for end-users. It effectively requires fundamental knowledge of the Windows operating system. All system interactors are willing to use the computerized system and all have adequate knowledge to handle the system therefore it will be not a difficult task to train for the computerized system.
- Economic Feasibility

  Once the proposed system is operational, the medical center will reap great benefits. Many great benefits will be enjoyed by the medical center when the proposed system is operational. Thus, considering the benefits of the new system, investing is not without consequences.
- Organization Feasibility

  After implementing the proposed system, if the system owner is willing to have new features, there is no barrier to have them.

1.7. Structure of the Dissertation

1.7.1. Background

The Background Chapter reviews an analysis of the requirements and existing software that is similar to the proposed system. An analysis of functional and non-functional requirements of MCMS is included. There is a history and nature of a system that has been thoroughly described in the literature. It identifies features of existing systems and the gap between the proposed system with the existing systems.

1.7.2. Design

This chapter contains the design for the proposed system. The ER, the proposed system's high-level architecture diagram, and UML diagrams are used in this chapter to determine the system's operations and procedures.

1.7.3. Evaluation

Because this suggested system is an implementation project, it includes a critical assessment of the system. It examines if the project's objectives were met, and if not, why. This chapter describes the evaluation and testing of system interactors.

1.7.4. Conclusion

This chapter concludes with a realistic evaluation of MCMS, presents the conclusions gained, and recommendations for future development. This chapter indicates the summary of the results of the proposed project, identifies the deficiencies of the proposed project, how to solve problems and deficiencies through future work.

Chapter – 02 Background

2.1. Introduction

In this chapter, the existing system of HMC will be discussed with emphasis on its drawbacks. The purpose of the proposed MCMS is to evaluate and understand the system requirements by focusing on functional and non-functional requirements. Furthermore, this chapter describes the background of the proposed application, a comparison of features with the existing present system, design strategies, development strategies, technologies, hardware, and software of similar systems, how the proposed system overcomes the issues existing in the current system and suggests improvements to the proposed application.

2.2. The Existing System

The existing system of HMC is a manual system that can be characterized by many less-effective processes. In the current system patients come and take the patient ticket (Number) from the counter. When the relevant patient's turn comes, the doctor will prescribe the medicine for the patient. After that, that patient goes to the pharmacy counter and takes medicine from the pharmacist of the Habarakada Medicare Center. In this center issuing patient numbers, billing system, managing medicine stocks, and every other function is done manually.

HMC and the patients who come to the HMC faced many difficulties in that existing system, which results in wasting of time, cost, and effort for both HMC and its potential patients. This led to the need of improving the current manual system to be more efficient and user-friendly. The overall view of the existing system is illustrated in Figure 2.1: Use Case Diagram for the Existing System.

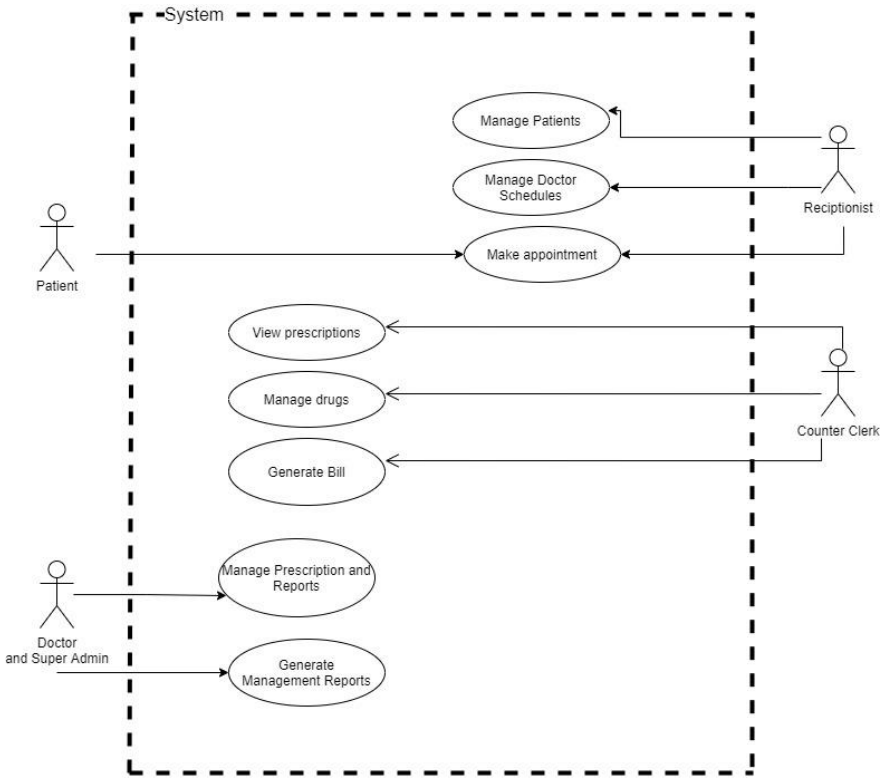


Figure 2.1: Use Case Diagram for the Existing System

2.3. Cons of Current System

- Since making appointments is done in manual logbooks, retrieval of information to respond to queries is slow and inefficient
- The waiting time is high as the patients have to wait until their turn comes
- Issues of the patient queue
- The HMC staff has to manually record the patient details with the appointments, which could lead to errors and miscommunications
- Data loss is probable due to records being maintained manually
- There is no method to handle historical data about the patients' history
- Unable to identify drugs unavailable due to improper medicine management
- The errors are possible due to manual bill/ invoice generation
- Time-consuming and difficult to produce reports

  Due to the above difficulties, the HMC developed an MCMS for HMC.

2.4. Requirement Analysis

Requirement Analysis is one of the major stages in the software development life cycle. Before transforming a manually driven process into a computerized system it is very important to gather knowledge about the problem source and user requirements. Following methods are used to gather requirements,

- Observation
- Inspection of existing documentation and existing system
- Interviews

The most common types of system requirements are,

- Functional Requirements
- Non-functional Requirements

2.4.1. Functional Requirements

The Medicare Centre Management System contains various processes, registering a patient as the initial process, recording prescription and providing it to the patient, creating a comprehensive database to record medical information of the registered patients, recording the medical test report to the relevant patient account, report generating and automated the billing process and report generation (Sanjana, 2019). Detailed functionalities of each user role are listed out below.

- Functional Requirement of the receptionist
    - Registering a patient
        - Adding patients
        - Deleting patients
        - Modifying patients Information
        - View patients
    - Reserving time
        - Update available time
        - Generating the report of the patients

- Functional Requirement of the pharmacy counter clerk
    - Billing process and Report Generation
        - Add Items
        - Remove items
        - Billing calculations
        - Printing bills and reports
    - Automated the pharmacy process and Report Generation
        - Drug stock management (Add, Delete, and Modify)
        - Reorder level notification
        - Searching drug
        - View stock
        - Stock summary report generation
        - Printing reports

- Functional Requirement of the Doctor
    - Recording the prescriptions and the medical reports
        - Record the prescriptions and the medical reports Deleting prescription
        - Printing the prescriptions and the medical report
        - View the prescriptions and the medical reports
        - Deleting prescriptions and the medical reports
    - Search patients to check the history

- Functional Requirement of Patients
    - Check the availability and reserve the doctor
    - View the patient's profile

2.4.2. Non-Functional Requirements

Following are the non-functional requirements of the MCMS. All the non-functional requirements fall under security, user-friendly, performance, availability, and reliability.

- Security
    - System privileges are different from user to user the highest privileges give to the super administrator of the MCMS.

- User-friendly
    - The system should be more user-friendly. The user interface should be kept simple.
    - The system should be flexible to expand and enhance when required.
    - The system should support all leading web browsers for the patients.

- Performance
    - Less response time for operations of MCMS

- Availability
    - When the patient requires access, the system is available all the time.

- Reliability
    - The degree to which the result of system reports and calculation of the invoice should be accurate.

2.5. Similar Systems

Nowadays most hospitals used Hospital Management System but this system is implemented for the medicare center is exclusively for somewhere that has only one doctor, one receptionist, one pharmacy counter clerk, and many patients. This system is supposed to be developed for regional medicare centers and there is no medicare center network involved here. The following systems were identified as similar to the proposed system of MCMS.

Hospital Health Information Management System (hhims) ("HHIMS Hospital Health Information Management System," 2012) is one of the medicare center management systems in local. It is intended for use by clinical staff and stores clinical details of patients treated and the details are recorded on the system as the patient is examined. When a patient comes in for treatment, hospital staff can refer to previous clinical records, print visit slips or discharge letters for patients, print quarterly health statistics, and prepare infectious disease notifications for local Medical Officers of Health.

Kareo Clinical ("Kareo Clinical | Electronic Medical Software," 2021) is "a fully integrated cloud-based EHR software that lets your medical practice write notes and prescriptions, code encounters, and manage patients. The software was designed by healthcare providers and is easy to set up with no downloads and no hardware required". The features of Kareo Clinical are a comprehensive dashboard, simple charting, easy e-prescribing, convenient eLabs, integrated patient portal, secure messaging, flexible calendar, and agenda overview.

Patient Manager ("Vertikal Systems Patient Manager," 2018) is "a clinic management software that lets the team shared medical records and communicate effectively. The solution connects your medical team, front desk, and cashiers while providing a strong level of security". It has the following features and the Patient Manager Dashboard is illustrated in Figure2.2: Patient Manager Dashboard.

- Patients' management: manage and organize all types of patients.
- Medical records: document everything: consultations, vitals, lab results, imaging.
- Medical scheduling: track patient appointments, medic waiting lists, and inpatient schedules.

- Medical forms: employ form template in all departments for quick data entry.
- Medical billing and claims: generate the invoices by extracting them from the records.

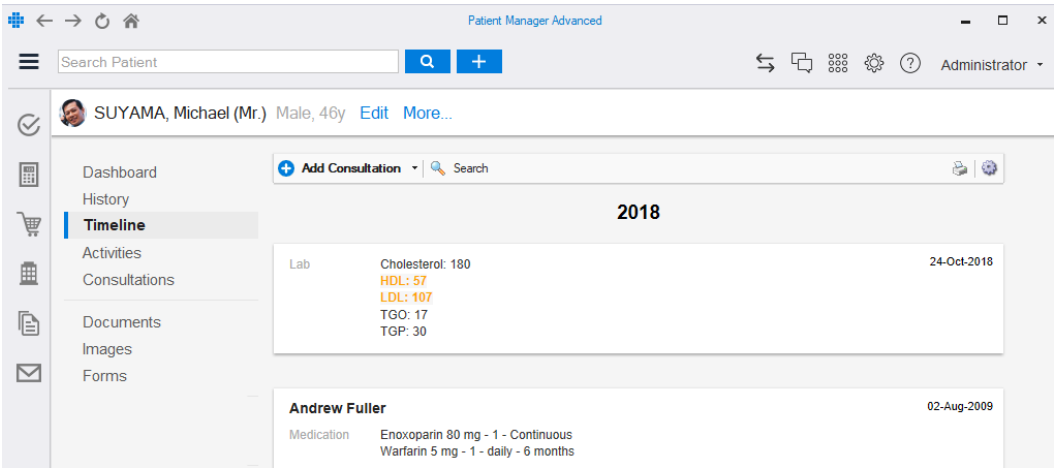


Figure 2.2: Patient Manager Dashboard

hCue ("hCue | Clinic Management Software," 2017) is "a clinic management system providing EMR and EHR capabilities for doctors and patients. This integrated health care platform helps manage doctor appointments, medical records, and pharmacy billing. The solution streamlines clinical workflows by integrating your medical practice and helping connect to patients safely to keep their data secure". The features of hCue are online appointment management, practice management, pharmacy management, register patients, and create notes. The hCue Dashboard is illustrated in Figure2.3: hCue Dashboard.

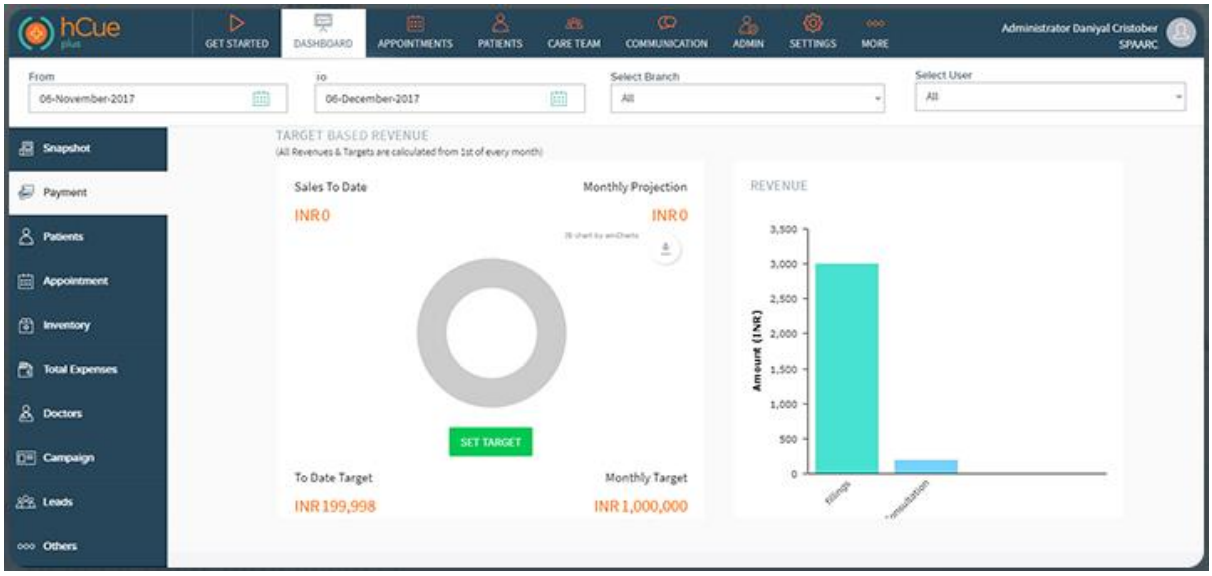


Figure 2.3: hCue Dashboard

2.6. Quality of the Solution

The summary of the comparison of MCMS is described in Table 2.1: Summary of Comparison.

Table 2.1: Summary of Comparison

MCMSs / Feature	Patient Manager	hCue	Kareo Clinical	MCMS for HMC
User friendly	No	Yes	Yes	Yes
Too complex	Yes	Yes	Yes	No
Secured	Yes	Yes	Yes	Yes
Important Report Generation	Yes	Yes	Yes	Yes
Costly	Yes	Yes	Yes	No
Used costly Softwares	Yes	Yes	Yes	No
Inpatient treatment	Yes	Yes	Yes	No
Outpatient treatment	Yes	Yes	Yes	No need
Online appointment	Yes	Yes	Yes	Yes
Doctor availability scheduling	Yes	Yes	Yes	Yes
Prescription and report storing	Yes	Yes	Yes	Yes
Automated invoice generation	Yes	Yes	Yes	Yes
Checking to reorder levels for drug	No	Yes	Yes	Yes

When compared with the existing medicare center management system this proposed system has some better user-friendly features. Many systems are developed for the complex hospital networks but Habarakada Medicare Center is a regional small institution that treats outpatients. Therefore, the system is developed specifically for a single medicare center. Following are the reasons that the proposed system is the most suitable one and this way the proposed system gives solutions to solve the drawbacks of the existing system.

- User friendly

The system interfaces more simply and users can understand and give quick responses to the activities. Both administration staff and patients can work easily.

- The proposed system is developed in PHP and uses the MYSQL database server. For web development, PHP is the most widely used scripting language. It's a server-side application that's free and open-source. MySQL is a Structured Query Language (SQL)-based Relational Database Management System (RDBMS). In addition, it is open source and free.

- More features for administration staff and patients
  - Registering a patient as the initial process of Software Requirement Specification
  - Reserving time
  - Check the availability and reserve the doctor
  - Recording the prescriptions
  - Billing process and Report Generation
  - Automated the pharmacy process and Report Generation
  - Creating a comprehensive database to record medical information of the registered patients

- Handling confidential data in a safe manner

2.7. Related Design Strategies

The waterfall model, a popular variant of the software engineering development life cycle model, may be used to construct most medicare center management systems. It uses a linear and sequential development process (MD Adam Tabriz, 2020).

To design the proposed system, the Iterative Waterfall Methodology was used of which every phase has a feedback link to the preceding phases. These feedback routes allow for the correction of errors that occurred during a previous phase where errors would be recognized later. The feedback channels allow the phase in which errors occurred to be modified, with the modifications reflected in subsequent phases. When the mistakes and errors are identified and corrected during the early stages of the development, the time and energy needed to fix deficiencies could have economized.

Furthermore, the Model View Controller (MVC) architecture is used in this MCMS. The goal of this MVC-based system is to produce the most desired result by being quick, interactive, simple of software development and maintenance, and satisfying to the user. The MCMS system of HMC is then implemented using the PHP programming language and the Codeigniter framework.

2.8. Chapter Summary

The background chapter was dedicated to discussing the existing system of HMC, the drawbacks of the existing system, the system requirements by focusing on functional and non-functional requirements, hardware and software of similar systems with comparison and related design strategies. The design stage of the system development life cycle of MCMS will be discussed in the next chapter.

Chapter – 03 Design

3.1. Introduction

The design architecture of the project, the design of the proposed system using unified modeling language (UML), and the translation of the design into the desired design specifications into implementation, the database table structure, and user interface designs of the modules in MCMS are described in this chapter.

3.2. System Architecture

The proposed system uses MVC (Santosa et al., 2019) and this web-based system is designed for remote and portable use and it has also been devised to facilitate smooth and easy use for the system users. All the collected data can be controlled and managed centrally and users can have access to manipulate and present the data locally. The system uses remote networking technology and multiple concurrent users at remote sites of care. The system architecture of the web-based medicare center management system is shown in Figure 3.1: System Architecture.

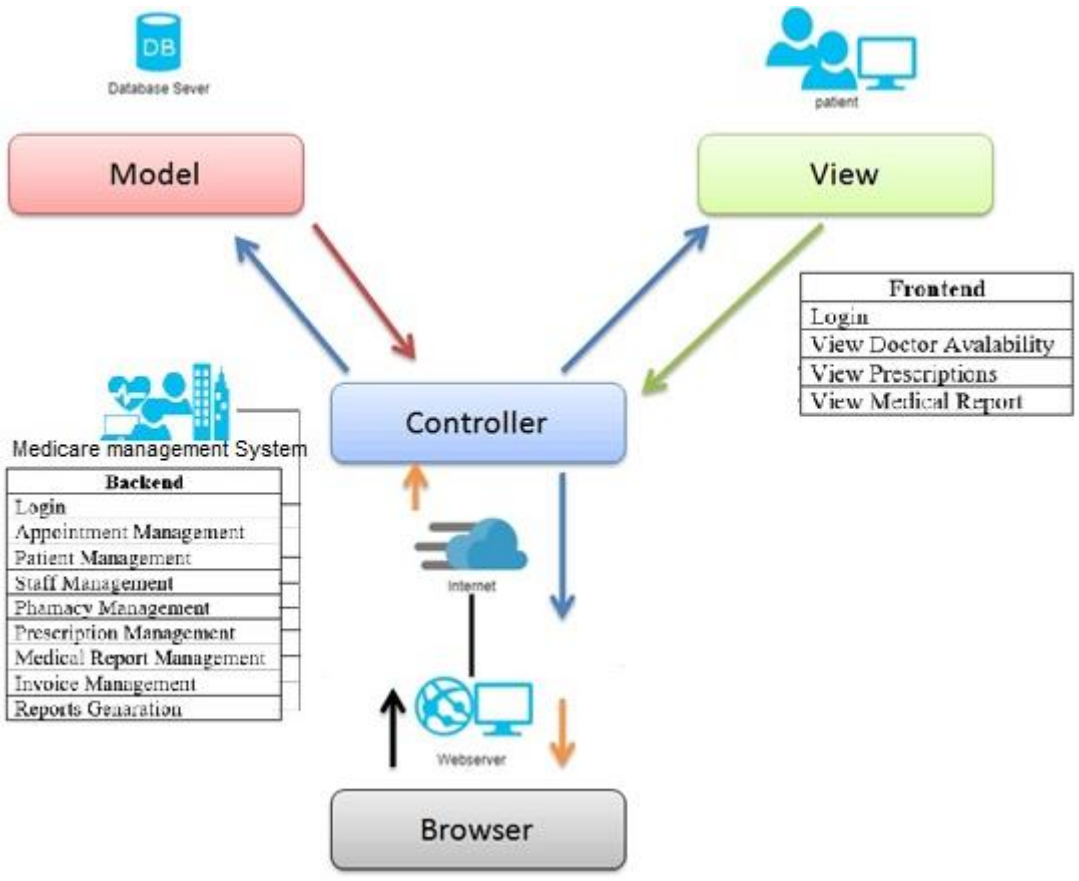


Figure 3.1: System Architecture

In MVC architecture, the application data and functionality, as well as the connection to the database, supporting the software application's dynamic data structure are controlled by the model.

The view is the visual representation of data in MVC architecture, such as forms, tables, diagrams, and charts. This section has taken care of the functionality that interacts directly with the user's graphical user interfaces (GUI).

In MVC architecture, the controller who is the link between the model and the view receives the inputs and fetches or updates data from the model to create a view and then sends data from the model to the view, which the user may see on the interface.

3.3. UML Diagrams

3.3.1. Use Case Diagram of the System

The use case diagram of the system is represented in Figure 3.2: Use case diagram of the System. The patients can make an appointment online and after that, every process is managed by the doctor and the counter clerk. The processes include the management of appointments, patients, medical records, prescriptions, drugs, and invoices/bills.

The actors must be logged into the system to do many functionalities. When designing a use case diagram, use cases were drawn for each category and those are described below.

- Use cases related to the patient
- Use cases related to the doctor
- Use cases related to the receptionist
- Use cases related to the pharmacy counter clerk

The doctor has another role in the system other than his role as a professional doctor, that is the role of super administrator. The super administrator is the superior who manages the whole system and he has all privileges to do interface in the entire process.

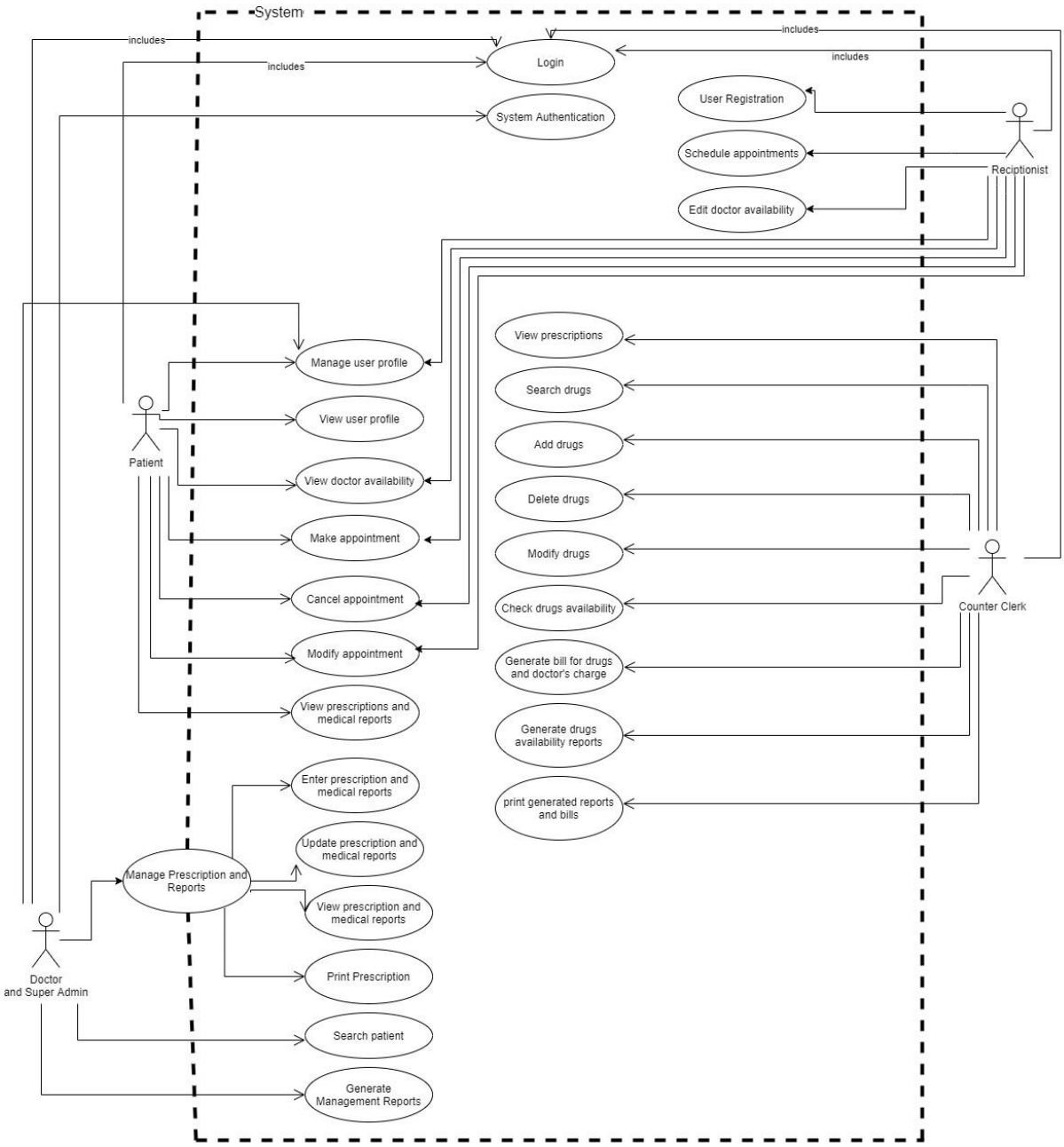


Figure 3.2: Use Case Diagram of the System

3.3.1.1 Use Case Scenarios

Some of the use case descriptions which are related to the above use case diagram are illustrated in Tables 3.1 – 3.5.

Table 3.1: Use Case Scenario – Login

Use-Case Name	Login
Use-Case ID	01
Use-Case Description	Doctor, Patient, Receptionist and Pharmacy counter clerk Login
Primary business actor	Doctor, Patient, Receptionist and Pharmacy counter clerk
Pre-Condition	Internet connection is important to access the system
Main flow	1. Doctor, Patient, and Counter staff should visit the “Medicare center management system”. 2. System will ask User's name and password. 3. Doctor, Patient, and Counter staff should give his/her username and password 4. System does the authentication. 5. Home page will display.
Alternative flow	3 (a) Authorization Fails. 4 (a) 1. A message will be displayed and that indicates “Incorrect User name or Password”. 4 (a) 2. Doctor, Patient, and Counter staff will be allowed to re-type User name and Password.
Post-condition	Doctor, Patient, and Counter staff will access his functions

Table 3.2: Use Case Scenario – Update Profile

Use-Case Name	Update profile
Use-Case ID	02
Use-Case Description	Update profile
Primary business actor	Doctor, Patient, Pharmacy counter clerk
Pre-Condition	Internet connection is important to access the system. The doctor, patient, and counter staff should log in.
Main flow	1. Doctor, Patient, and Counter staff should click the "update profile" option. 2. Then Doctor, Patient, and Counter staff can update his/her profile 3. System does the authentication 4. "New profile information" will be stored.
Alternative flow	
Post-condition	Doctor, Patient, and Counter staff can access the system with his new information

Table 3.3: Use Case Scenario – Add New Drug

Use-Case Name	Add new drug
Use-Case ID	03
Use-Case Description	Add new drug
Primary business actor	Pharmacy counter clerk
Pre-Condition	Internet connection is important to access the system Counter staff should log in.
Main flow	1. Counter staff should select the "drug" button which appears on his/her home page 2. Then he/she should select the "Add drug" button. 3. Then he/she should add new drug details such as generic name, expiry date and, minimum required amount. 4. Then he/she should click the "Add" button.
Alternative flow	
Post-condition	

Table 3.4: Use Case Scenario – Generate Reports

Use-Case Name	Generate Reports
Use-Case ID	04
Use-Case Description	Prepare drug report, drug reorder level report, daily appointments report, and current patient report
Primary business actor	Doctor
Pre-Condition	Internet connection is important to access the system. The doctor should log in.
Main flow	1. Select required reports 2. View
Alternative flow	
Post-condition	The requested reports will be viewed accordingly

Table 3.5: Use Case Scenario – Make an Appointment

Use-Case Name	Make an appointment
Use-Case ID	05
Use-Case Description	Make an appointment online
Primary business actor	Patient and Receptionist
Pre-Condition	Internet connection is important to access the system The doctor should log in.
Main flow	This use case starts when an actor wishes to place an appointment online. 1. View available time. 2. Actors can select the time slot
Alternative flow	
Post-condition	an appointment has been made.

3.3.2. ER Diagram

The entity-relationship diagram of the system is represented in Figure 3.3: ER Diagram.

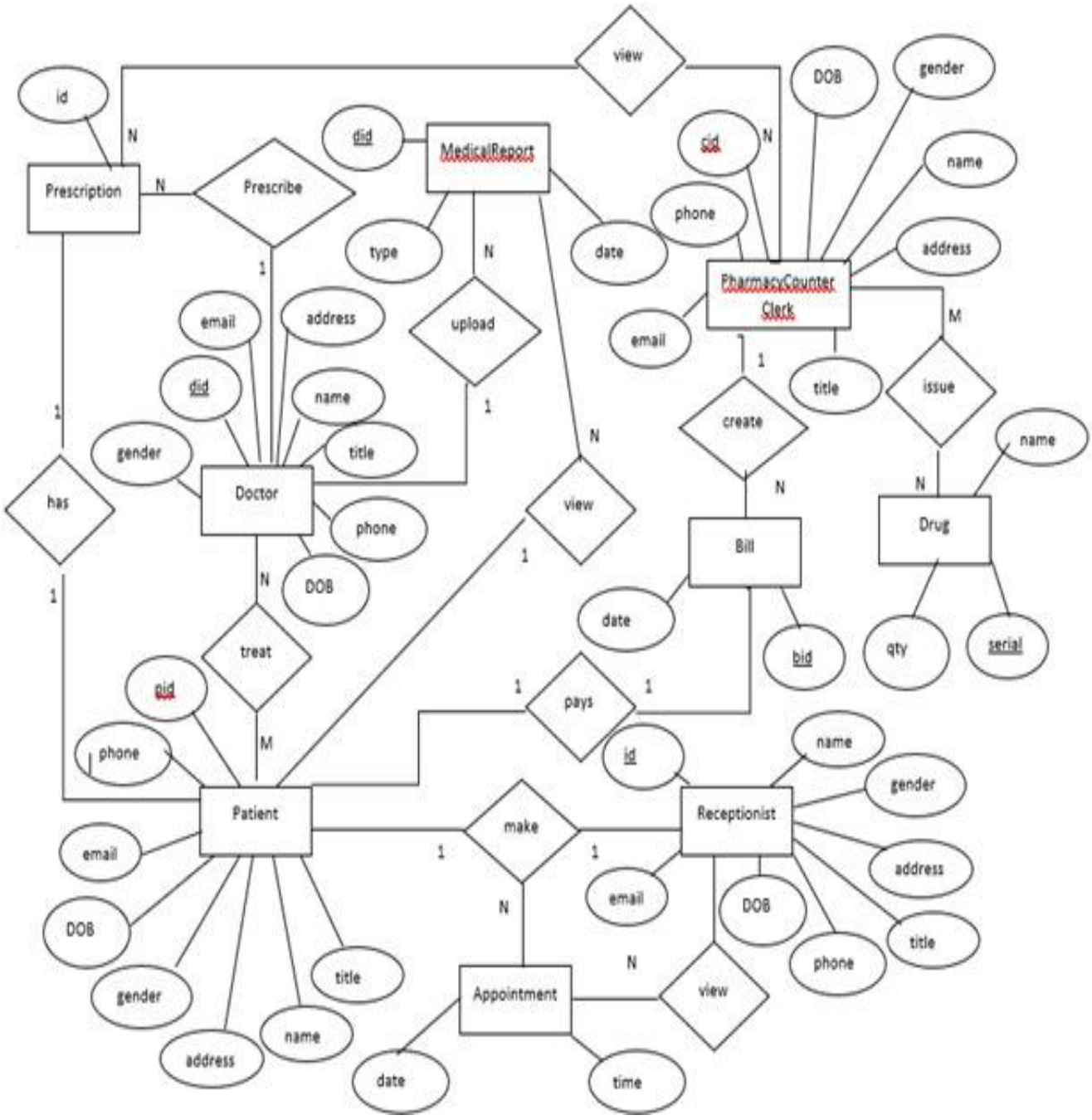


Figure 3.3: ER Diagram

3.3.3. Class Diagram of the System

The class diagram of the system is shown in Figure3.4: Class Diagram of the System. The user class is inherited by the classes Doctor, Counter staff, and patient respectively. All the other classes contain relevant information accordingly.

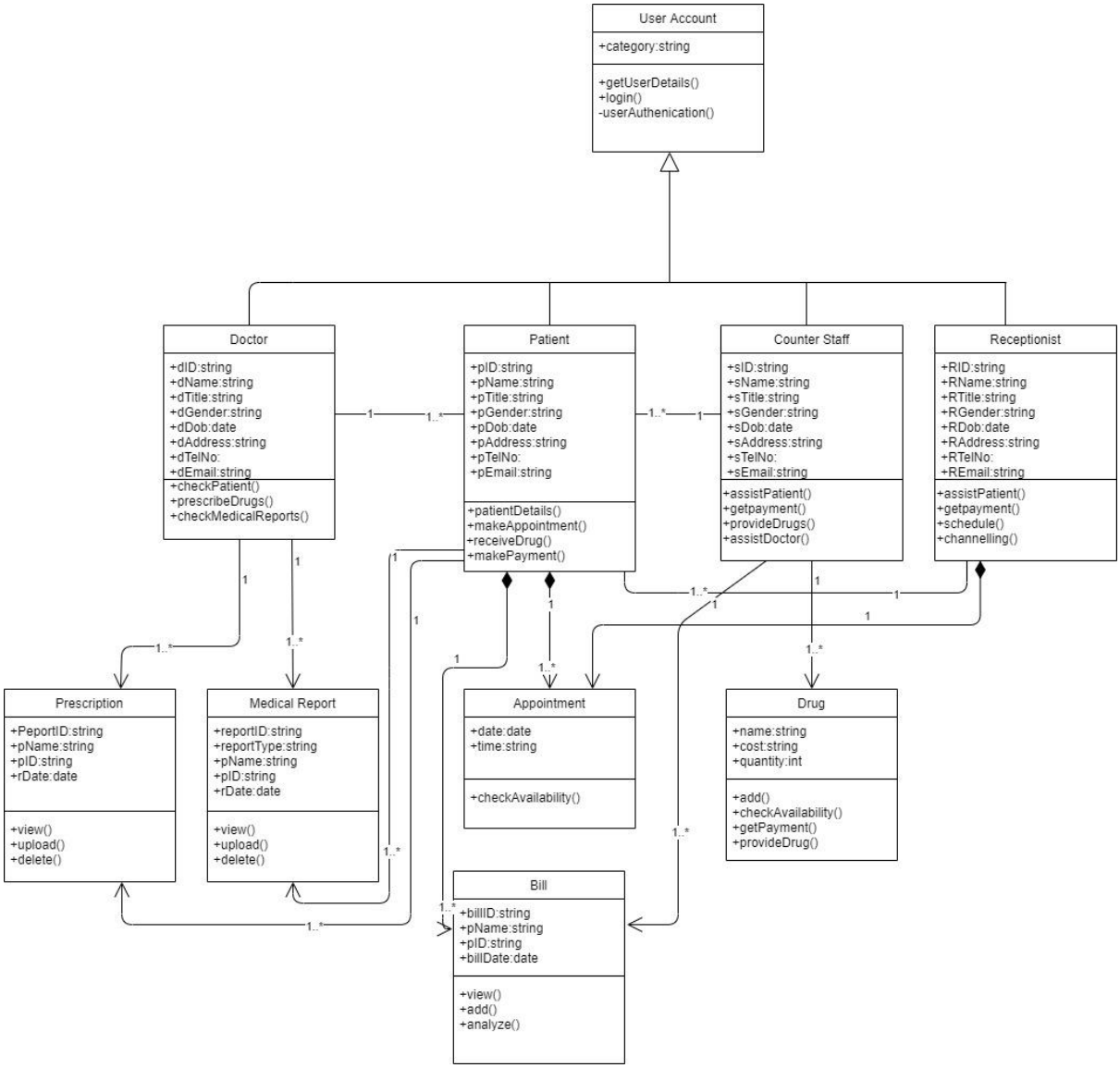


Figure 3.4: Class Diagram of the System

3.4. Database Table Structures

The database of the web-based medicare center management system will be implemented using Structured Query Language (MYSQL server). The table structures of the database are shown in Figure 3.5: Database Table Structure.

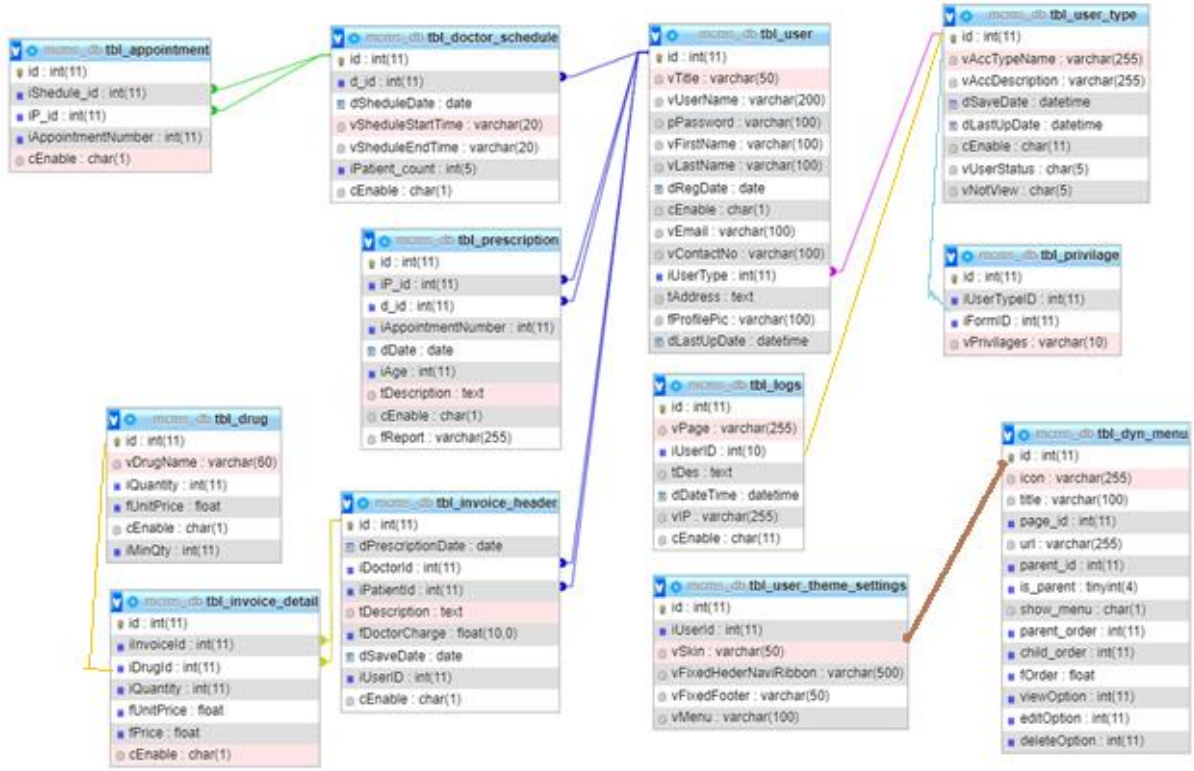


Figure 3.5: Database Table Structure

3.5. User Interface Design

The objective of a good UI should be to make client and customer interactions straightforward, instinctive, and efficient. HMC has to serve the main objective of UI design efficiently. Further, the UI design must ensure a convenient interchange of information between medicare providers and patients.

In addition, the medicare application design must highlight the details of a medical application. Besides, it should guide the users and prompt the next action. When you have the right UI design, the app can be an effective tool for lead generation. Moreover, it can turn out to be the main product of the medicare center.

Also, UI/UX design promotes the success of our clients. It is critical to consider the clients' requirements and design accordingly. ("UI Design Tips For Healthcare Industry," 2020). The following UI design elements were adapted to the MCMS of HMC.

- Easy of understanding navigation
- Well-positioned buttons and the search bar
- The straightforwardness of steps required to complete the task
- Attractiveness

The above-mentioned features have been implemented to the MCMS of HMC. "Appendix A: User Manual" can be referred to the UI designs and the user flow of the system.

3.6. Chapter Summary

The design chapter was dedicated to discussing the design architecture of the project, the design of the proposed system using unified modeling language (UML), the database table structure, and user interface designs. The implementation stage of the system development life cycle of MCMS will be discussed in the next chapter.

Chapter – 04 Implementation

4.1. Introduction

The implementation chapter is a strictly defined combination of how the system is developed which includes the details of the tools and techniques used to implement this system such as the frameworks, programming languages, scripting languages, database systems, etc. Additionally, it includes major codes segment also.

4.2. Implementation Environment

4.2.1. Hardware

- Intel Core i3 Processor
- 4 GB RAM
- 64-Bit Operating System
- 100 GB Hard Disk Drive
- Dual Monitors 1366 X 768

4.2.2. Software

- WAMP server

  WAMP is made available under an open-source license. It supports a significant portion of the functionality of the most expensive and powerful database packages. WAMP makes use of a standard implementation of the well-known MYSQL data language. WAMP is compatible with a wide range of operating systems and languages, including PHP.

- PHP
  - Easy to start with PHP
  - Easy to use
  - Integrated database support

- Notepad++

  The code was done with Notepad++ text editor. Its user-friendly UI and shortcode keys make the completion of this task quicker.

- HTML 5

  HTML is a mark-up language that is commonly used in web browsers to structure content. HTML 5 is the most recent iteration of HTML. It has numerous new capabilities and is compatible with all modern browsers, including iOS. HTML5 has a lot of multimedia and graphics capability.

- CSS 3

  CSS3 is the most recent update to the CSS family. Drop shadows, rounded edges, gradients, and transitions are just a few of the new features. In addition, it cuts down on coding time and supports a variety of browsers.

4.3. Reused Components

- Bootstrap

  Bootstrap (Nathan Segal, 2016) is an open-source front-end framework that includes HTML, CSS, and JavaScript tools for developing responsive websites. Bootstrap is easily customizable, and its mobile-first approach aids in the creation of responsive websites for all types of device browsers. It makes developing the front-end of the system more appealing and efficient because it includes many components such as menus, alert boxes, and JavaScript plugins. When developing a website with the CodeIgniter framework, Bootstrap is a very useful tool for building the front end.

- Codeigniter Framework

  Codeigniter is a robust open-source PHP-driven framework designed for developers who require a simple and elegant toolkit for creating multi-featured web applications. Model-View-Controller (MVC) design pattern is used by Codeigniter to separate application logic from presentation logic and business logic. Since the presentation is distinct from the PHP scripting, the webpages require less scripting.

4.4. Related Technologies

Modern web-based technology is offered in almost all medicare fields. The majority of hospitals are converting and establishing a digital front for all their major operations to get closer to the booming digital market. Even though in today's world, the information flow is very fast, that speed and quality cannot be equally attributed to the regional medicare center or the hospitals. Online connectivity is now a must for all well-organized and well-managed establishments. It is desirable that all the medicare centers are converted to web-based ones so as to uplift the entire system to be enriched in more efficiency and effectiveness.

- When considering the related technologies of total web solutions that are available for a medicare center, most of them have been developed using PHP and JavaScript through content management systems. PHP is a much-preferred option over JavaScript. Using a framework like CodeIgniter is more beneficial considering factors such as user-friendliness, security, and the support to build large dynamic web applications. CodeIgniter is lightweight and high speed during the implementation. The proposed system was developed using PHP.

- Nowadays MCMSs use many comprehensive database servers like PostgreSQL. it is an open-source, feature-rich database that is perceived as a go-to solution for performing complicated, high-volume data operations. MYSQL is a much-preferred option over other database servers. Considering the reliability, high volume data handling, and not complicated data handling, MYSQL is the most preferred one to use as the database server.

- The proposed system is web-based and the patient can access it on any smart device and it is compatible with iOS and Android or any OS. The developer can use Intel i3 or a better processor with 4GB RAM for the development.

- Many web-based medicare center management platforms eliminate the need for paper prescriptions in medical centers and propose E-Medical Management, which would improve patient management efficiency, doctor schedule management, and provide universal access to patient data anywhere in the

hospital. One of the primary technologies used in the proposed system is electronic data handling.

- Many Hospitals use the Hospital Operating System, Clinic Operating System similar to iOS and Android or any OS, they are standardized software. “Most medicare center management systems cover healthcare regulatory and revenue cycle compliance with auto-update. Highly customizable software with interoperability and health exchange features built-in the system”(Wilson, 2020). This proposed system is not used any specific operating system and has not used revenue cycle compliance as an additional feature since this is a regional and small-scale medicare center.

- In the health care industry, they adopt several new technologies to have the best customer compliance. These include the integration of barcodes, integration of biometric authorization, token-based treatment facility, integration with image capture equipment, accessibility through mobile, editable treatment plans, assistant doctor registration, multi-location accessibility, creating and updating patient education material, flexibility to update bills, etc (Uy Raymonde Charles Y. et al., 2015). As a future development, these technologies can be added to get value addition to HMC.

4.5. Major Code Segment

Major code segments in the MCMS are discussed below in detail. The code segment of the login is illustrated in Figure 4.1: Login Code.

```
79          $this->load->model('model_users');
80
81          $userDetail = $this->model_users->get_user_id();
82
83
84          $data = array(
85
86              'oba_userbackendsession' => $userDetail["id"],
87
88              'oba_is_logged_inbackendsession' => 1,
89
90              'oba_vUserNamebackendsession' => $userDetail["vUserName"],
91
92              'oba_iUserTypeBackendsession' => $userDetail["iUserType"],
93
94              'oba_iUserTypeNamebackendsession' => $userDetail["vAccTypeName"],
95
96              'oba_vFirstNamebackendsession' => $userDetail["vFirstName"],
97
98              'oba_vLastNamebackendsession' => $userDetail["vLastName"],
99
00              'oba_DynMenuSelect' => '1'
01
02          );
03
04          $this->session->set_userdata($data);
05
```

Figure 4.1: Login Code

The code segment of getting drug reorder level is illustrated in Figure 4.2: Drug Reorder Level.

```
30      public function view_min_drug() {
31          $data = array();
32          if (empty($data))
33              $data['saveStatus'] = 'V';
34
35              $this->load->model('admin_model');
36
37              $sql_data = "SELECT
38              tbl_drug.id,
39              tbl_drug.vDrugName,
40              tbl_drug.iQuantity,
41              tbl_drug.fUnitPrice,
42              tbl_drug.cEnable,
43              tbl_drug.iMinQty
44              FROM
45              tbl_drug
46              where tbl_drug.iQuantity < tbl_drug.iMinQty
47              ORDER BY
48              tbl_drug.vDrugName ASC";
49
50
51          $data['list_data'] = $this->common_model->populate_drop_down($sql_data);
52
53
54              $this->load->view('adminpanel/header_view');
55
56              $this->load->view('adminpanel/masterdata/min_drug_report_view',$data);
57              $this->load->view('adminpanel/footer_view');
58      }
```

Figure 4.2: Drug Reorder Level

The code segment of common save is illustrated in Figure 4.3: Save.

```
58      public function save_data($tbl_name) {
59          $fields = $this->db->list_fields($tbl_name);
60
61          $data = array();
62          foreach ($fields as $field) {
63              $f_string = "";
64              $f_string = $field[0];
65
66              if ($field !== "id")
67                  $data[$field] = $this->input->post($field, TRUE);
68
69              if ($f_string === 'p')
70                  $data[$field] = md5($this->input->post($field, TRUE));
71
72              if ($f_string === 'f') {
73
74                  $filevalidate = $_FILES[$field]['name'];
75
76                  if ($filevalidate != '') {
77
78                      $imagename = $this->doupload($field);
79
80                      $img = $imagename['upload_data']['file_name'];
81
82
83                      $data[$field] = $img;
84                  }
85              }
86          }
```

Figure 4.3: Save

Refer “Appendix B: “Major Code Segments” for some of the other important coding of the system.

4.6. Chapter Summary

The implementation chapter was included the details of the tools and techniques used to implement this system such as the frameworks, programming languages, scripting languages, database systems, and major codes. The next chapter discusses the testing and evaluation section of the system development life cycle of MCMS.

Chapter – 05 Testing and Evaluation

5.1. Introduction

After the implementation of the system, testing and evaluation was carried out with patients and staff of HMC. The aim of conducting testing and evaluation was to determine the quality and accuracy of the system, how users behave on different modules and identify areas of further development. In this chapter, we have presented the testing plan, the test results, and the evaluation of the results.

5.2. Related Software Testing Types

All testing types include black-box testing and white-box testing. The following testing methods were used in the system to evaluate and maintain the quality of the product.

- Unit Testing

  The system is made up of various components, each of which must be tested separately. Each of these modules is handled as a distinct entity and tested separately. It will define a set of scenarios that will be tested in a black-box setting. In these scenarios, there will be a series of inputs and projected outputs. Without realizing how or where the inputs are processed, the tester will interact with the system's user interface by providing inputs and evaluating outcomes. The actual users will most likely follow the approach since it separates the user's perspective from the developer's perspective.

- Integrated Testing

  The previously tested units will be combined and retested as a group. In order to integrate and test modules as a group, a variety of methods are employed. The smallest tested portion of the system is defined as a unit in this aspect. Integration testing which can uncover problems with software component interfaces before they cause problems in real-world program functioning is used to identify any discrepancies between related components.

- System Testing

  System testing looks for issues in both individual components and the entire system. The observed behavior is the result of system testing when a component or a system is tested. System testing is done on the entire system, either to meet system requirements, functional requirements, or both. During system testing, the design and behavior of the system, as well as the patient's expectations, are investigated.

- Acceptance Testing

  The system's implementation will be followed by user acceptance testing. The manual procedure runs concurrently with the preliminary implementation step, and the system will be thoroughly tested by the end-user. At this point, most minor modifications will be captured.

5.3. Test Plans

The Test Plan is intended to specify the scope, approach, resources, and schedule for all HMC MCMS testing activities. The plan specified the products to be tested, the features to be tested, the types of testing to be performed, the testing staff, the resources and timeline needed to complete the testing, and the risks associated with the plan. Test cases must be identified before the test plan can be carried out (Hamilton, 2021).

5.4. Test Cases

A test case is a series of actions performed to validate a certain feature or function of the system, and it often includes test steps, test data, preconditioning, and postconditioning built to verify any need (“Sample Test Case Template with Test Case Examples,” 2021). Test cases are necessary for both developer-led system testing and user acceptability testing to keep track of which tests had been completed and to inform users about the regions that needed to be tested. The following section depicts the test cases and their accompanying test results.

5.4.1. Functional Test Cases

Some of the functional test cases in the system are illustrated in Table 5.1 and 5.2 represent and Refer “Appendix C: “Test Case Documents” for some of the other important test cases of the system.

Table 5.1: Login Related Test Cases

Test Case ID	Functionality	Procedure	Expected Results	Test Data	Status
Login related Test cases					
01.	Log in to the System as an Admin	Go to the login page Enter User Name Enter Password Click on Login Button	Users should be able to login to the system ONLY admin privileges should be displayed	Username: lakmini Password: admin	Passed
02.	Log in to the System as a patient	Go to the login page Enter User Name Enter Password Click on Login Button	Users should be able to login to the system ONLY patient privileges should be displayed	Username: nilanga Password: Nilanga	Passed
03.	Unsuccessful Login	Go to the login web page Enter User Name Enter Password Click on Login Button	Invalid User Name Valid Password	Username: lakmini Password: admin123	Passed
04.	Verify Admin user can logout from the system	Login as an Admin Click Logout Button	Admin should be able to log out of the system		Passed
05.	Verify patient can logout from the system	Log in as a patient Click Logout Button	The customer should be able to log out of the system		Passed

Table 5.2: Functionalities in Logged Patient

Test Case ID	Functionality	Procedure	Expected Results	Status
Functionalities in logged patient				
06.	Dashboard	Click dashboard menu	Direct to the dashboard	Passed
07.	Go to make Appointment	Click Doctor Channel Click Make Appointment	Direct to Doctor Availability	Passed
08.	Load make appointment	Click book now	Direct to add an appointment	Passed
09.	Add appointment	Enter Doctor Enter Schedule date Enter Schedule time Schedule patient Click submit	Make new appointment	Passed
10.	Load My Profile	Click profile	Direct to my profile	Passed
11.	Load Prescription	Click prescription	Direct to prescription	Passed

5.4.2. Non-functional Test Cases

Non-functional tests are just as essential as functional tests in terms of client satisfaction. These tests are intended to assess the application's readiness based on non-functional parameters that are never addressed in functional testing. Security and usability testing are some of the non-functional testings. The examples of the non-functional test cases in the system are represented in Tables 5.3 and 5.4.

Table 5.3: Non-Functionalities in Security_ Password Encryption

Test Case ID	Non-Functionality	Procedure	Expected Results	Test Data	Status
Password Encryption					
01.	Verify whether the passwords are stored encrypted in the database	User registration and verify the password field in the database.	Passwords should be stored encrypted in the DB.		Passed

Table 5.4: Non-Functionalities in Usability_ View

Test Case ID	Non-Functionality	Procedure	Expected Results	Test Data	Status
View					
02.	Verify whether the available schedules are displayed to the patient	Go to make an appointment	View the available schedules		Passed
03.	View patient List	Click view patient List	Display patient List		Passed

5.5. Test Results

After the development process was completed, the test cases were executed to assess how many of the test cases passed and whether the system was stable. The test result of the tests that were conducted is shown in Table 5.5: Test Result. It contains a summary of the areas that were tested and test results were obtained after the test cases were executed.

Table 5.5: Test Result

Functions	Description	Executed % Test Cases	Passed % Test Cases	Pending % Test Cases	Priority % Test Cases
Login	Whole Processes	100%	100%	0	High
Add		100%	100%	0	High
Modify		100%	100%	0	High
Delete		100%	100%	0	High
View		100%	100%	0	High
Search		100%	100%	0	High
Print		100%	100%	0	High

5.4. User Evaluation

The primary goal of the evaluation process is to decide whether it accomplishes the ideas presented in the initial overview, solves the problems that were identified, and assesses the quality of the system. Furthermore, the evaluation process will also assess the software development methodology that was used throughout the development of the framework, the usefulness of the technologies and tools used, and the accuracy of the estimations, and the usefulness of the client reviews.

Questionnaires are the primary data collection tool in surveys, which will be used to obtain information from end-users. The options and weights provided of the evaluation form are listed in Table 5.6: Likert Scale for Evaluation Form. The evaluation form to get feedback from the users is shown in Table 5.7: Evaluation Form.

Table 5.6: Likert Scale for Evaluation Form

Likert Option	Value
Very Poor	1
Poor	2
Satisfactory	3
Good	4
Excellent	5

Table 5.7: Evaluation Form

Medicare Center Management System Evaluation Form							
	Please select the category that best describes your role.						
(i).	Patient (ii). System User (iii). Admin						
	Parameter	1	2	3	4	5	Remarks
01.	Appearance						
	User interfaces						
	Colors						
	Fonts						
02.	Usability						
	Navigation was easy between the screens						
	Menus and options were easy to understand						
	Data validation was helpful						
	The requested output results are successful						
03	Functionality						
	Registration & Login Function						
	Add new, Modify & Delete Function						
	Search Function						
	Display details of the function						
	Make Appointment Function						
	Reorder Levels						
	Print function						
	Bill Generation						
04	Performance						
	Page load time						
	Response time						
	Total						

5.5. Evaluation Result

The evaluation results of the evaluation form are shown in Table 5.8: Evaluation Results and it is represented as pie charts in Figure 5.1: Evaluation Results Using Pie Charts.

Table 5.8: Evaluation Results

Parameter	Likert scale count									
	1	%	2	%	3	%	4	%	5	%
Appearance	0	0%	3	15%	5	25%	9	45%	3	15%
Usability	0	0%	1	5%	4	20%	10	50%	5	25%
Functionality	2	10%	5	25%	6	30%	6	30%	1	5%
Performance	0	0%	4	20%	7	35%	8	40%	1	5%

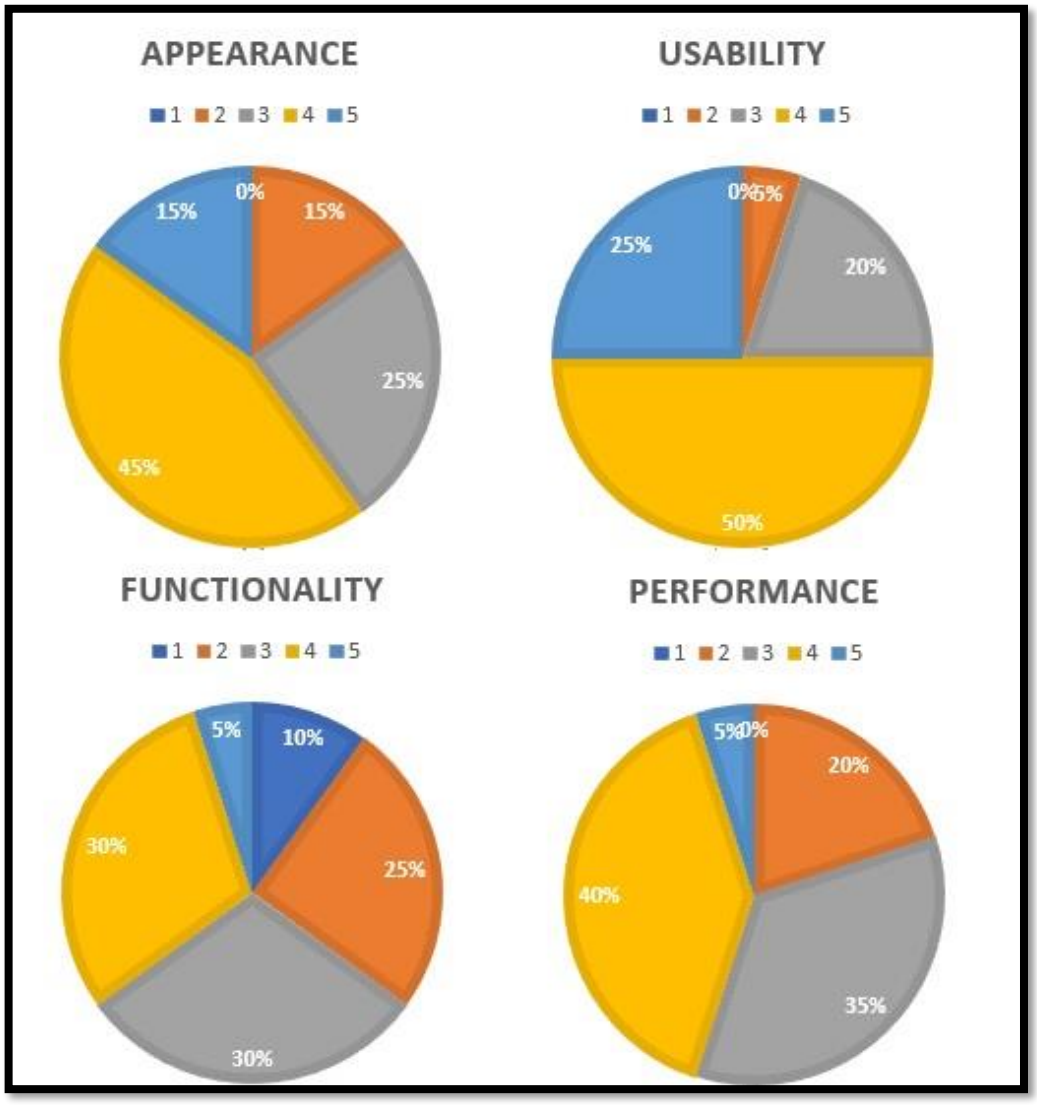


Figure 5.1: Evaluation Results Using Pie Charts

5.6. Critical Evaluation

According to the results obtained from the Evaluation Form of Medicare Center Management System, 45% of the users have rated the appearance as good and 15% of the users have rated the appearance as excellent. If the rates are above at least satisfactory level, it can be considered as pass. If so total pass rate of appearance is 85%.

The feedback for the usability is marked with 50% of users finding it good and another 20% finding it satisfactory and 25% finding it excellent. If the rates are above at least satisfactory level, it can be considered as pass. If so total pass rate of usability is 95%.

30% of the users feel the functionality is good and 5% of the users feel the functionality is excellent. If the rates are above at least satisfactory level, it can be considered as pass. If so total pass rate of functionality is 65%.

40% of the users rating the performance as a good level. 35% are rating the performance as satisfactory level and 5% are rating the performance as excellent level. If the rates are above at least satisfactory level, it can be considered as pass. If so total pass rate of performance is 80%.

According to the final rates of the above parameters, the developed system is at a satisfactory level in the real-world scenario.

5.7. Chapter Summary

The testing and evaluation chapter was dedicated to discussing the testing plan, test cases, the test results, and the evaluation of the results by considering user feedback. The next chapter discusses the overall conclusion of the system with the future enhancement of the system.

Chapter – 06 Conclusion

6.1. Introduction

This chapter includes the overall summary of the MCMS of HMC. In this chapter, the problems encountered, the lessons learned, how you met the original objectives of the project, and potential future work are discussed.

6.2. Problems encountered

Following were some of the problems encountered during this project.

- The requirement gathering step was the most crucial part of the development process. Because of the current pandemic situation in the country, physically visiting HMC to collect information has been a significant problem. As a result, making voice and data calls are the contact methods to HMC.
- The medicare center functionalities are not familiar to me. To overcome the internet sources are referred.
- Adapting to the MVC architecture of CodeIgniter was a big challenge. Hence, the time taken to study the framework was longer than expected.

6.3. Lessons learned

A few sets of problems were encountered at the developing stages and found suitable solutions to overcome them. Some of them are design level and some of the requirements need to change slightly.

- Online tutorials are a good way to learn new technologies and it is also important to follow the articles & speeches by experienced developers of that technology to understand the depth of the technologies and how to implement them.
- The most important lesson learned from this project was the significance of planning and implementing the project and adapting to changes in the timeline. Throughout the project, there were many deviations from the agreed timeline due to the feedback.

- Build good relationships with the clients, especially through regular communication and by sharing the progress regularly with the client through workable prototypes.
- Many of the problems were solved easily by conducting interviews and meetings with the industry experts. One such meeting was fruitful for me, as it helped to create a standard set of data to be collected from the customer and avoid the collection of unwanted data.

6.4. Critical Conclusion

According to the results obtained from the result of the evaluation, the total pass rate of appearance can be considered as 85%. The total pass rate of usability is 95%. The total pass rate of functionality is 65%. The total pass rate of performance is 80%. Therefore, according to the final rates of the parameters, the developed system is at a satisfactory level in the real-world scenario.

This developed system is a web-based system therefore it facilitated to reduce time-consuming, paper works of the patients and the staff of the medicare center. All the doctor prescriptions and lab reports can be stored in a digitized manner, therefore, making it easy to retrieve the data and eliminate misplacement of patients' data. There is a zero-error concept in doctor channeling, billing, and pharmacy management. The generated reports are more valued to take long-term and short-term decisions. Due to the pharmacy management system, the pharmacist can maintain drugs stock effectively.

By considering the user evaluation, all can identify the system provide a better user experience for all users of MCMS of HMC. It represents most of the objectives of the MCMS of HMC were achieved via this developed system.

6.5. Future Enhancements

Concerning future enhancements, a significant priority will be placed to improve the performance and usability of the website according to the feedback received. To improve the performance in the future, we are looking forward to using high-level techniques.

In addition, the online payment feature is not yet implemented on this website. Therefore, the focus will be placed to introduce a payment platform to accommodate users to make payments through Credit & Debit cards with ease.

Following are some other enhancements and features to be incorporated in the future.

- E-mail alerts to the pharmacist to acknowledge the reorder level
- Online payment
- Integration of channeling made to online calendars such as Google Calendar
- Handling medical history of family members
- Integration of ordering drugs from suppliers
- Keeping track of supplier details

List of References

Century, I. of M. (US) C. on A. the H. of the P. in the 21st, 2002. The Health Care Delivery System, The Future of the Public's Health in the 21st Century. National Academies Press (US).

Chandresh Prasad, 2017. Hospital Management System proposal.

Chueh, H.C., Barnett, G.O., 1994. Client-server, distributed database strategies in a health-care record system for a homeless population. J. Am. Med. Inform. Assoc. 1, 186–198.

Clinic Management Software [WWW Document], n.d. . App Dev. URL https://www.apptunix.com/solutions/clinic-management-software/ (accessed 5.28.21).

Hamilton, T., 2021. Test Plan Template: Sample Document with Web Application Example. URL https://www.guru99.com/test-plan-for-project.html (accessed 8.30.21).

hCue | Clinic Management Software [WWW Document], 2017. . Softw. Connect. URL https://softwareconnect.com/clinic-management/hcue/ (accessed 5.28.21).

HHIMS Hospital Health Information Management System [WWW Document], 2012. URL https://sites.google.com/a/hhims.org/hhims/home (accessed 5.28.21).

Kareo Clinical | Electronic Medical Software [WWW Document], 2021. . Softw. Connect. URL https://softwareconnect.com/ehr/kareo-ehr/ (accessed 5.28.21).

Kenton, W., 2021. Considering a New Venture? Consider a Feasibility Study [WWW Document]. Investopedia. URL https://www.investopedia.com/terms/f/feasibility-study.asp (accessed 9.1.21).

MD Adam Tabriz, 2020. The Waterfall Project Management Methodology [WWW Document]. Medium. URL https://medium.datadriveninvestor.com/the-waterfall-project-management-methodology-4d45c2025010 (accessed 5.28.21).

Nathan Segal, 2016. 10 Common Uses of Bootstrap [WWW Document]. HTML Goodies. URL https://www.htmlgoodies.com/html5/10-common-uses-of-bootstrap/ (accessed 8.30.21).

Pollak, V.E., Lorch, J.A., 2007. Effect of electronic patient record use on mortality in End Stage Renal Disease, a model chronic disease: retrospective analysis of 9 years of prospectively collected data. BMC Med. Inform. Decis. Mak. 7, 38. https://doi.org/10.1186/1472-6947-7-38

Sample Test Case Template with Test Case Examples [WWW Document], 2021. URL https://www.softwaretestinghelp.com/test-case-template-examples/ (accessed 8.30.21).

Sanjana, 2019. The Functional and Nonfunctional Requirement for HMS [WWW Document]. URL https://mocdoc.in/blog/the-functional-and-nonfunctional-requirement-for-hms (accessed 5.28.21).

Santosa, I., Kurniawan, F., Rahmanita, E., Rachmad, A., 2019. Application of Model View Controller Architecture in Hospital Inventory Logistic Management. https://doi.org/10.2991/icst-18.2018.72

UI Design Tips For Healthcare Industry, 2020. . CronJ. URL https://www.cronj.com/blog/ui-design-graphic-design-tips-for-healthcare-industry/ (accessed 7.17.21).

Uy Raymonde Charles Y., Kury, F.P., Fontelo, P.A., 2015. The State and Trends of Barcode, RFID, Biometric and Pharmacy Automation Technologies in US Hospitals. AMIA. Annu. Symp. Proc. 2015, 1242–1251.

Vertikal Systems Patient Manager [WWW Document], 2018. . Softw. Connect. URL https://softwareconnect.com/clinic-management/patient-manager/ (accessed 5.28.21).

Wilson, B., 2020. Should You Develop on iOS or Android for Healthcare Mobility? DevPro J. URL https://www.devprojournal.com/technology-trends/mobility/should-you-develop-on-ios-or-android-for-healthcare-mobility/ (accessed 5.28.21).

Appendix A: User Manual

A.1 Home page

The home page of the system is illustrated in Figure A.1: Home page.

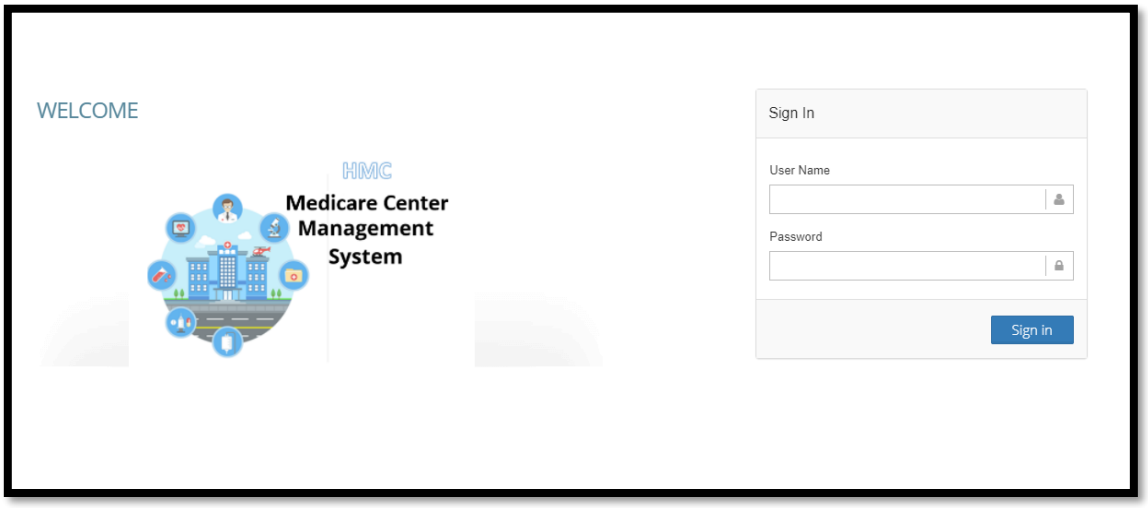


Figure A.1: Home page

A.2 Admin Dashboard

The Admin Dashboard of the system is illustrated in Figure A.2: Admin Dashboard.

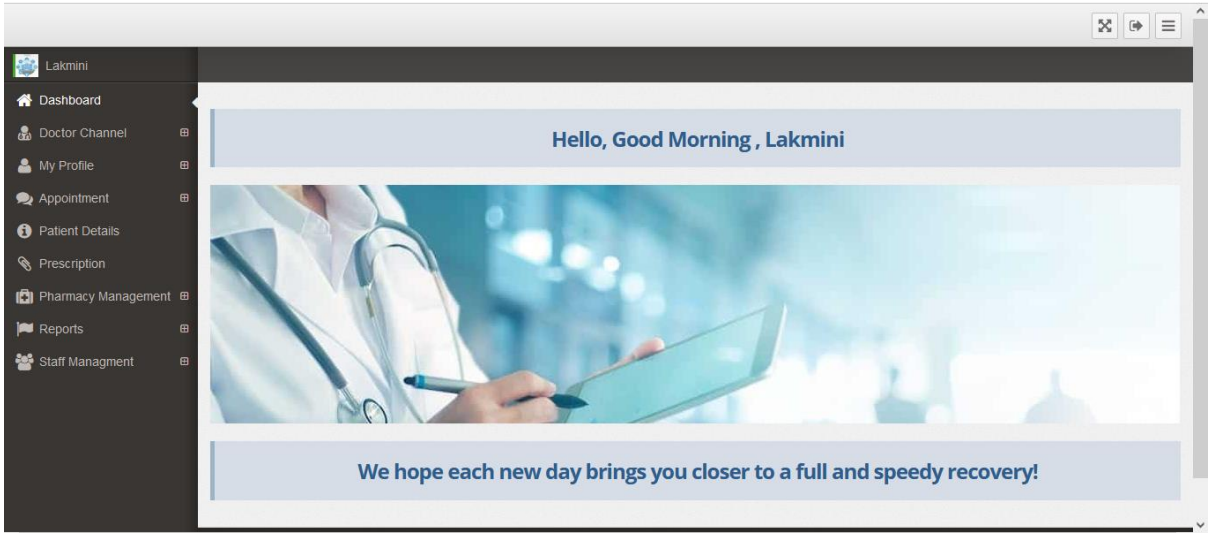


Figure A.2: Admin Dashboard

A.3 Patient Dashboard

The Patient Dashboard of the system is illustrated in Figure A.3: Patient Dashboard.

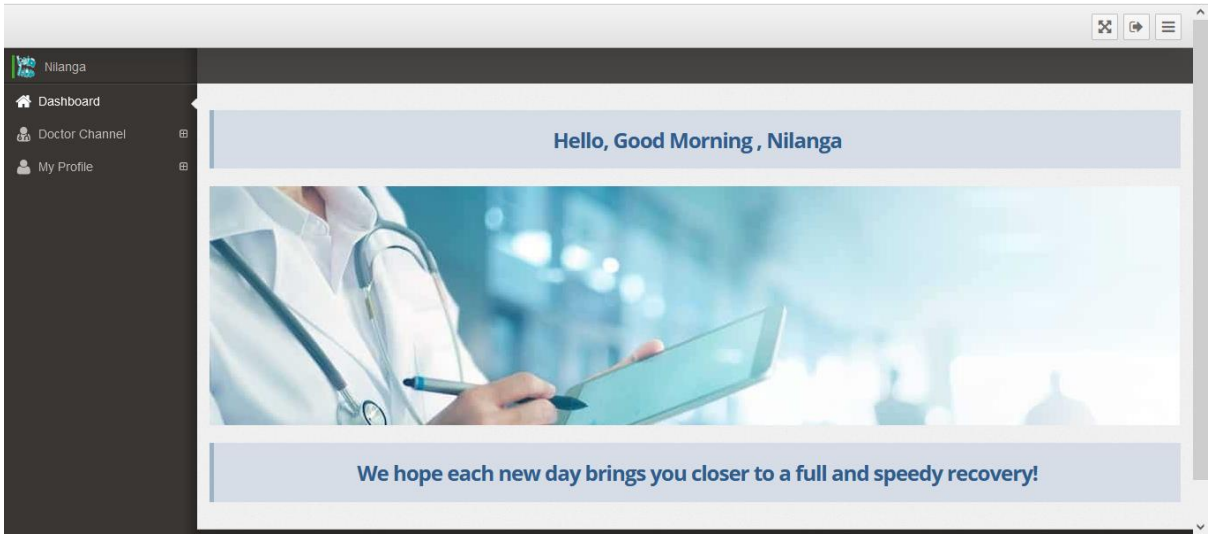


Figure A.3: Patient Dashboard

A.4 Doctor Dashboard

The Doctor Dashboard of the system is illustrated in Figure A.4: Doctor Dashboard.

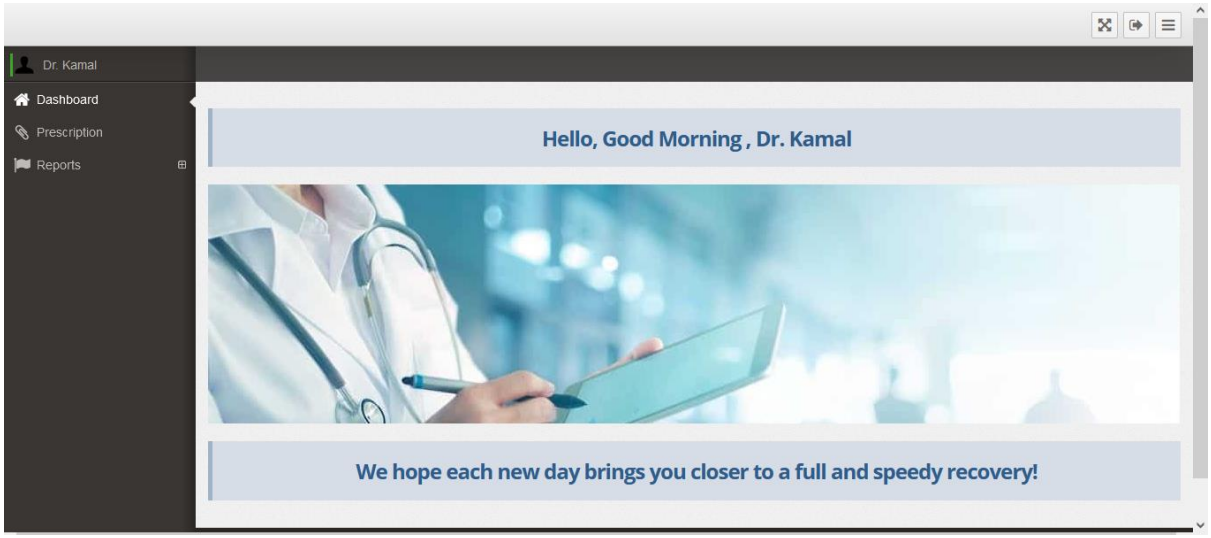


Figure A.4: Doctor Dashboard

A.5 Receptionist Dashboard

The Receptionist Dashboard of the system is illustrated in Figure A.5: Receptionist Dashboard.

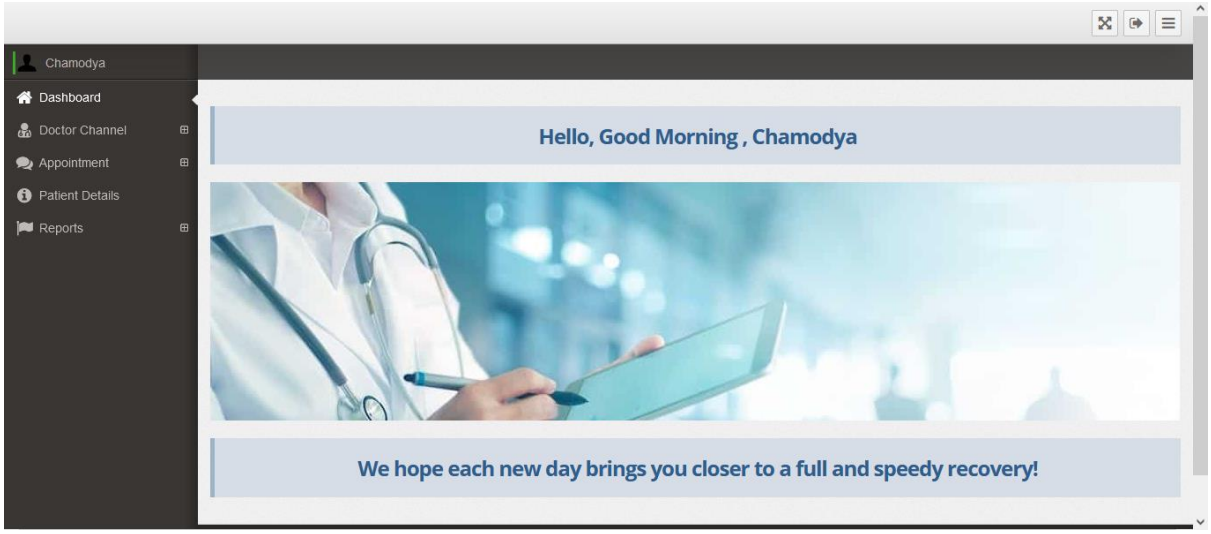


Figure A.5: Receptionist Dashboard

A.6 Pharmacy Counter Clerk Dashboard

The Pharmacy Counter Clerk Dashboard of the system is illustrated in Figure A.6: Pharmacy Counter Clerk Dashboard.

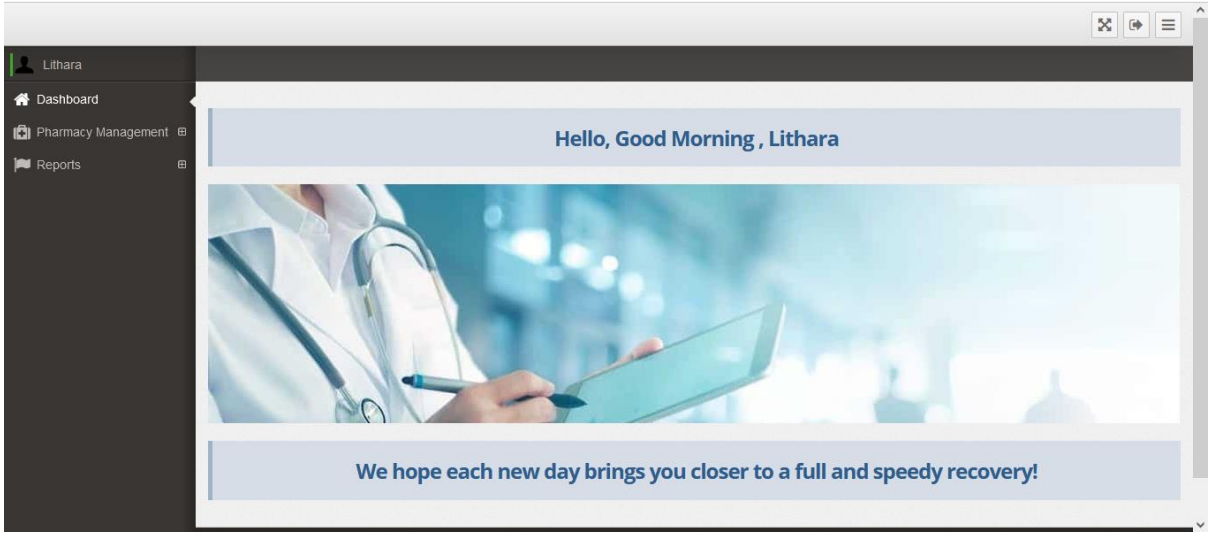


Figure A.6: Pharmacy Counter Clerk Dashboard

A.7 User Role

Add new roles and view all members are illustrated in Figure A.7: User Roles. Admin has privileges to this.

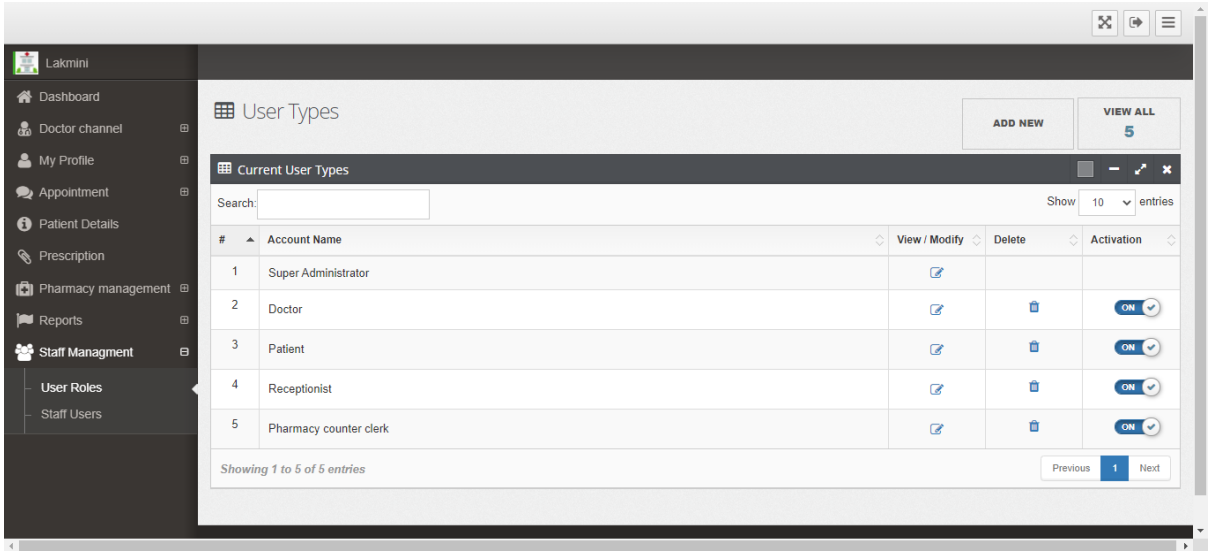


Figure A.7: User Roles

A.8 Staff Users

Add new members and view all are illustrated in Figure A.8: Staff Users. The modification and deletions are applicable. Admin has privileges to this.

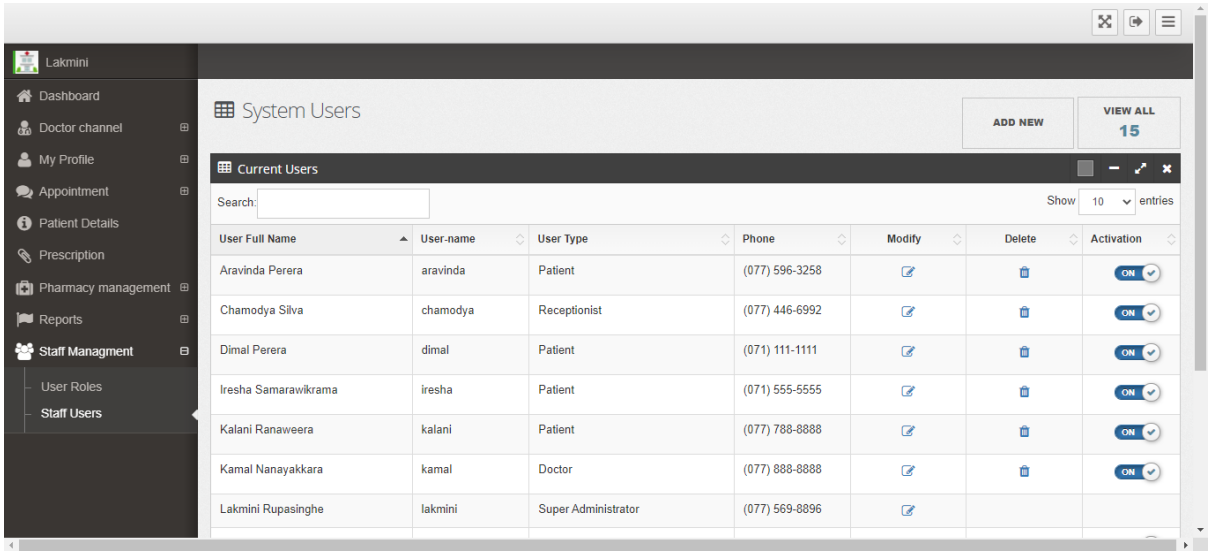


Figure A.8: Staff Users

A.9 Availability Management

Schedule the doctor, modify and delete the doctor's availability are shown in Figure A.9: Availability Management. The receptionist has privileges to do this.

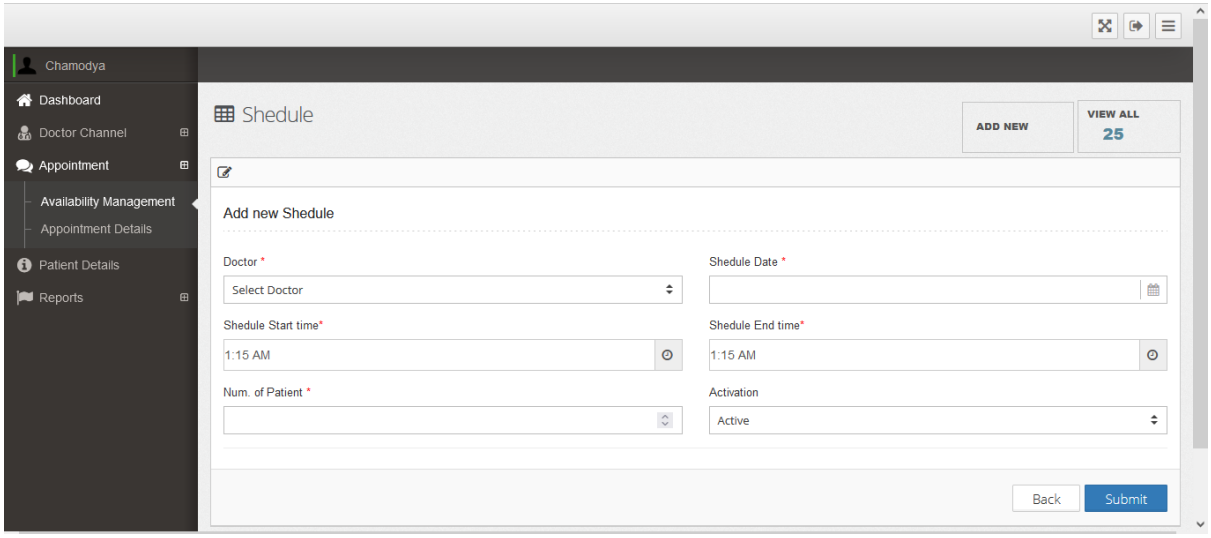


Figure A.9: Availability Management

A.10 Appointment Details

The receptionist has privileges to do this. Appointments details are shown in Figure A.10: Appointment Details.

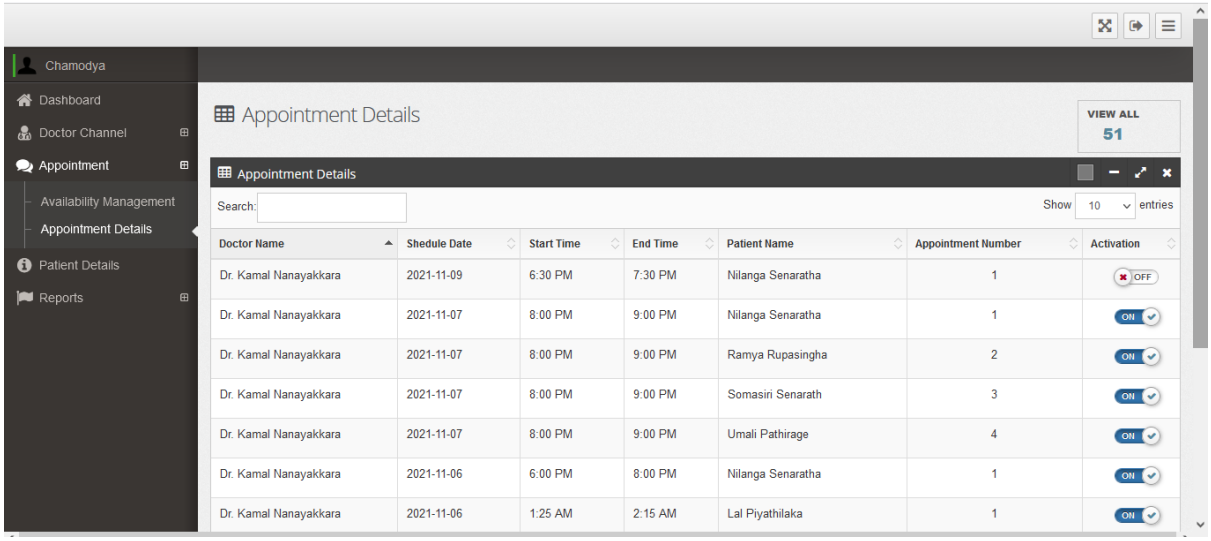


Figure A.10: Appointment Details

A.11 Doctor Channel at the Patient and Receptionist Level

Make an appointment is shown in Figure A.11: Make Appointment and it is done by the patient or receptionist.

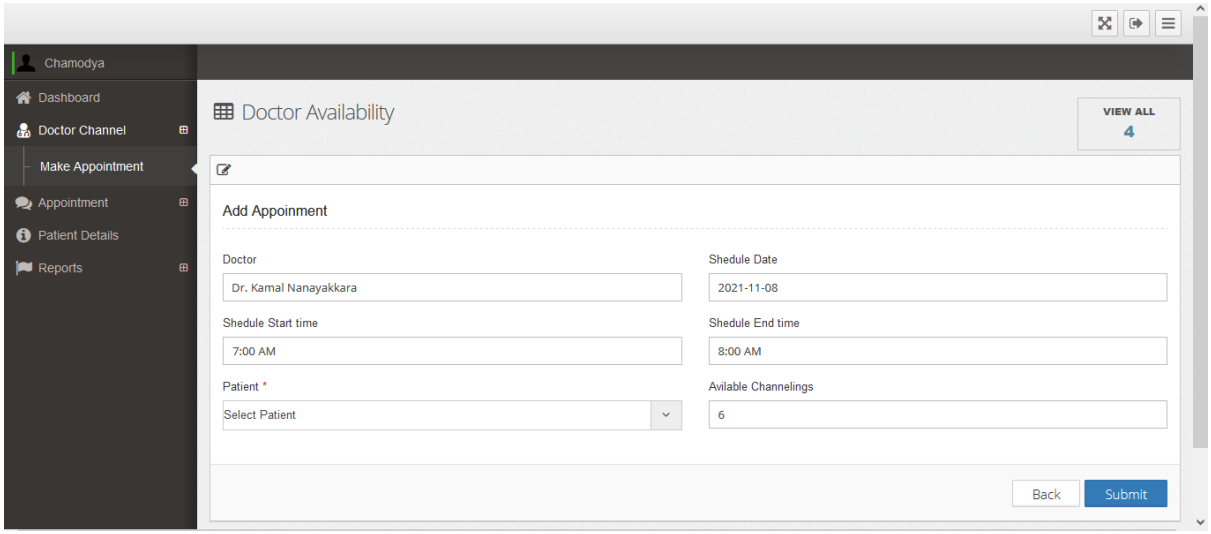


Figure A.11: Make Appointment

A.12 Register Patient

The receptionist has the privilege to register new patients using the registered patient interface. It is shown in Figure A.12: Register Patient.

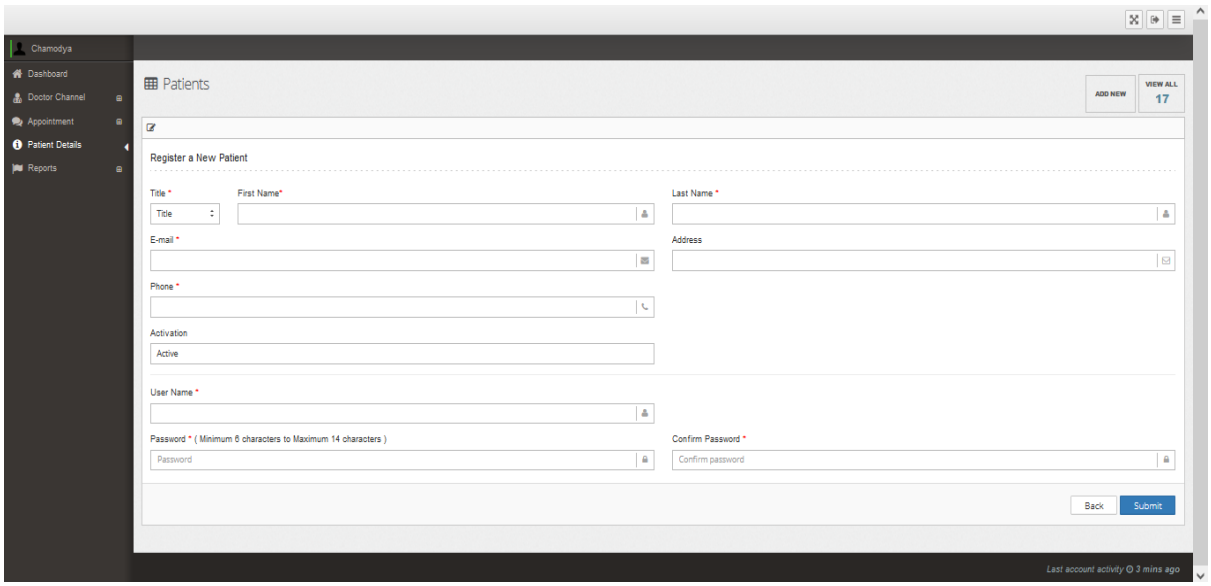


Figure A.12: Register Patient

A.13 Patients List

The receptionist has the privilege to check the patient list using the patient list interface. It is shown in Figure A.13: Patient List.

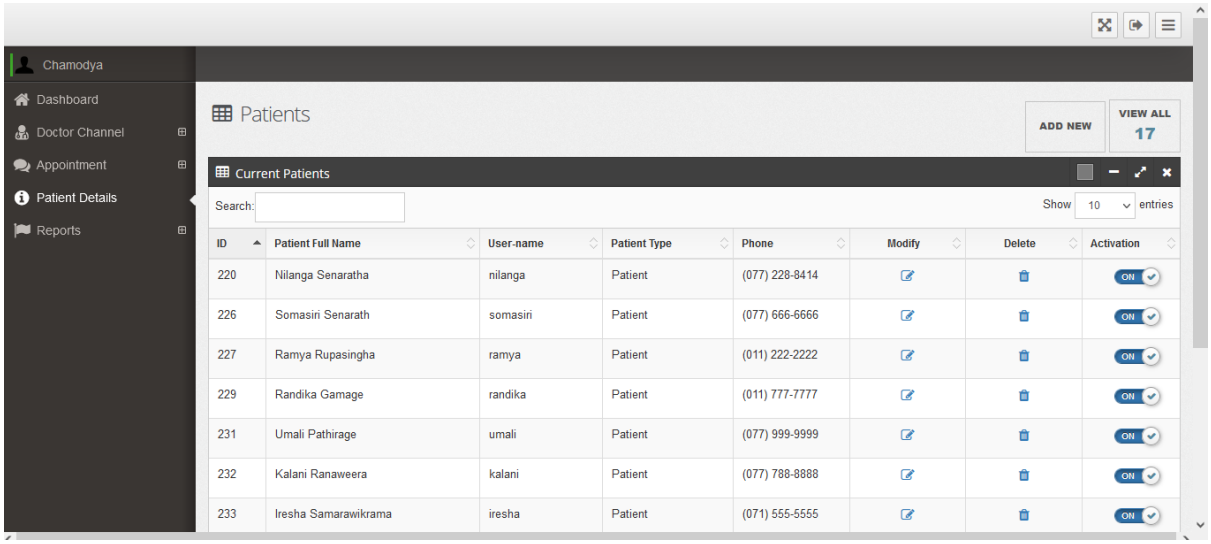


Figure A.13: Patients List

A.14 Prescriptions of My Profile

The patients have the privilege to check their prescriptions using the My Profile tab. It is shown in Figure A.14: Prescription of my profile.

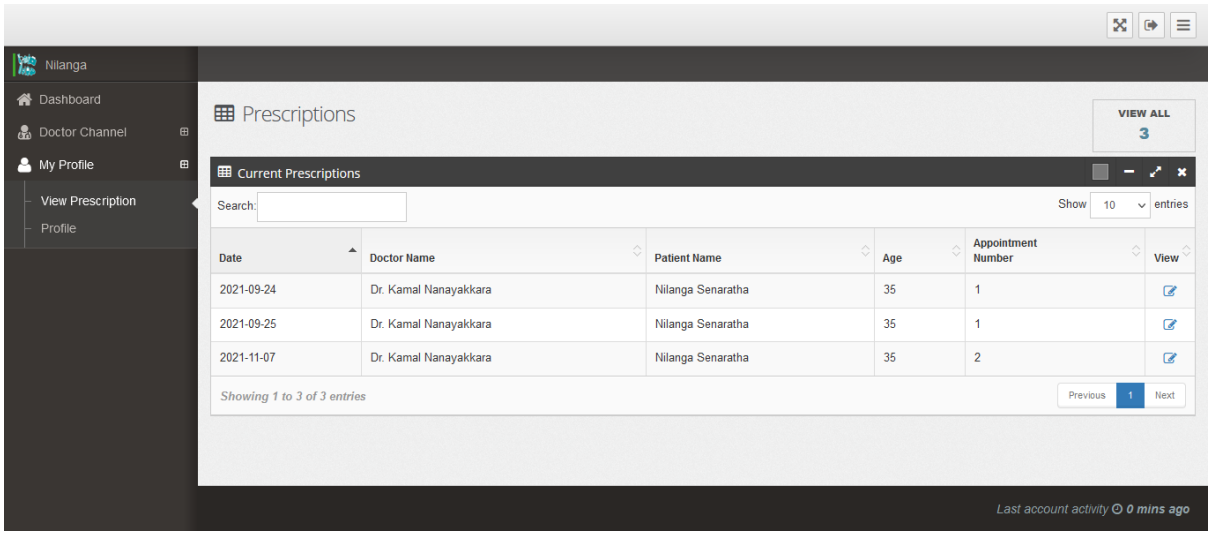


Figure A.14: Prescription of My Profile

A.15 My Profile

The patients have the privilege to view and change passwords in this module using the My Profile tab. It is shown in Figure A.15: My Profile.

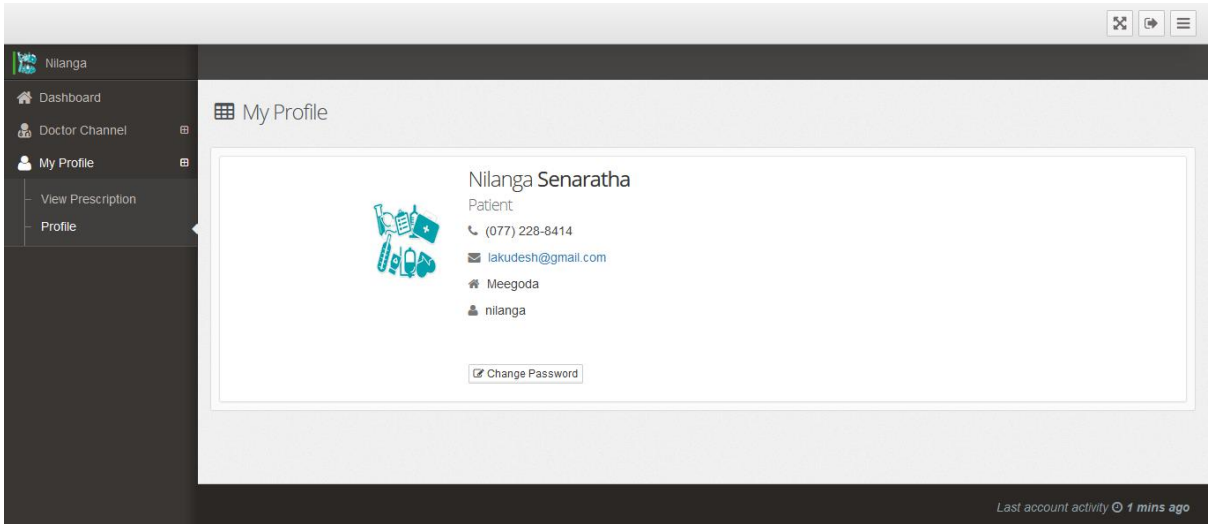


Figure A.15: My Profile

A.16 Add Prescription

The doctor has the privilege to make prescriptions. Register new prescription is shown in Figure A.16: Add Prescription.

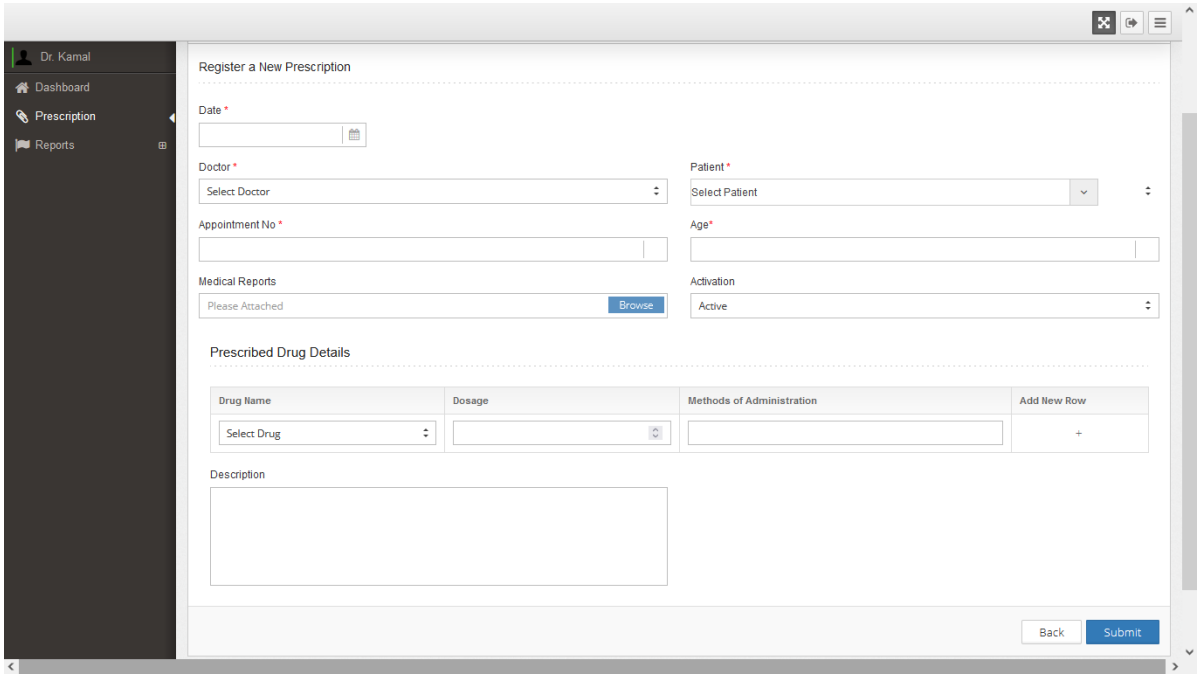


Figure A.16: Add Prescription

A.17 Add Drugs

The pharmacy counter clerk has the privilege to add medicines using the Drug interface. It is shown in Figure A.17: Add Drugs.

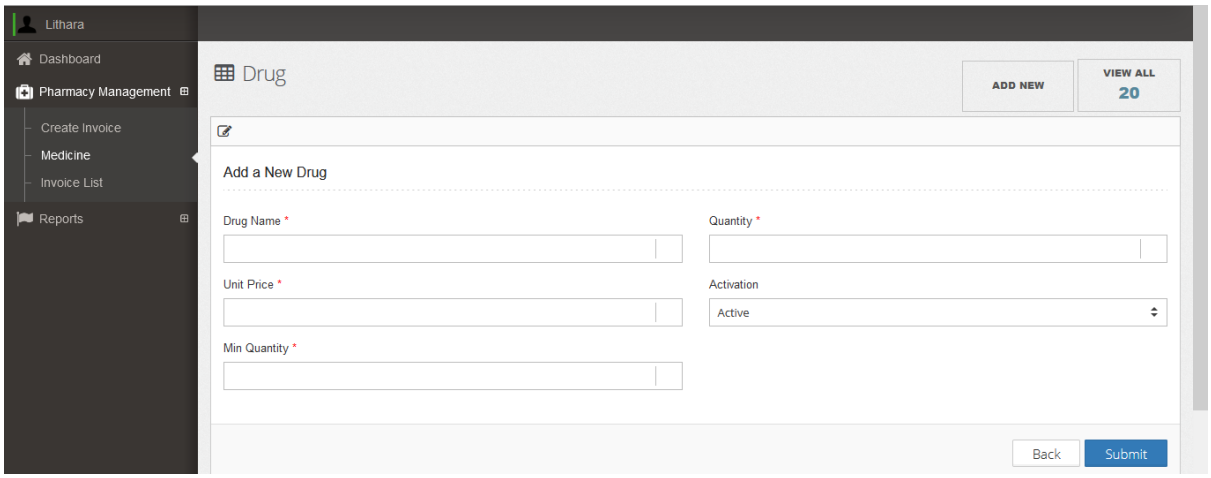


Figure A.17: Add Drugs

A.18 Add Invoice

The invoice of the patient can be prepared using the Prescription Details. The invoice preparation is illustrated in Figure A.18: Add Invoice.

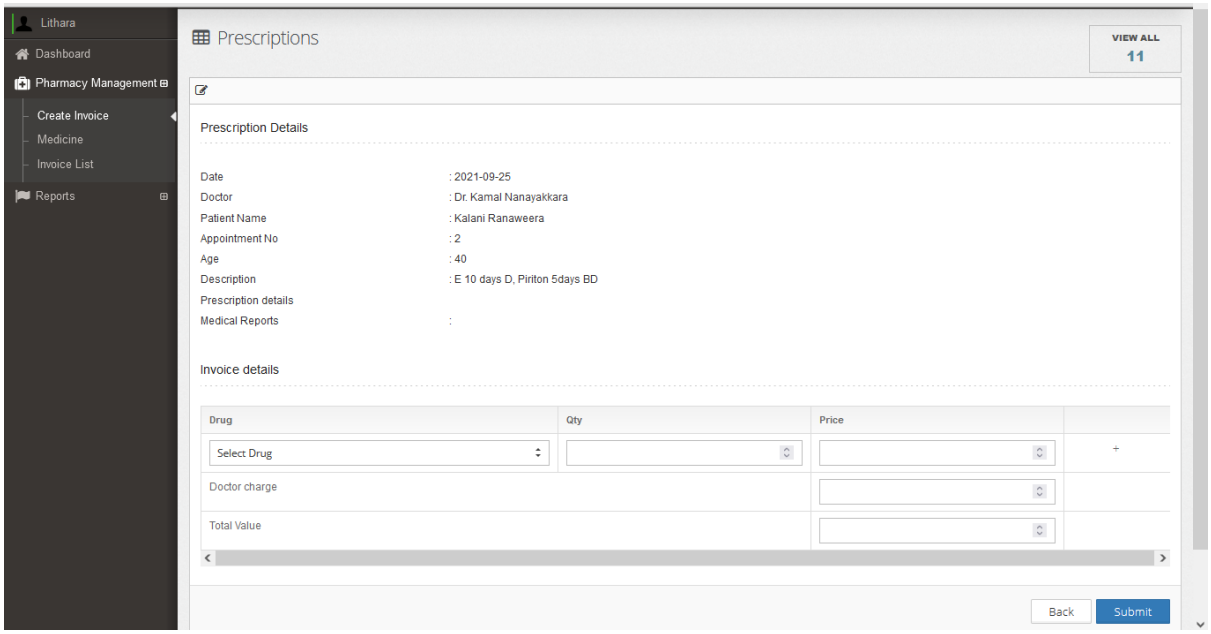


Figure A.18: Add Invoice

A.19 Invoice List

The pharmacy counter clerk has the privilege to check the list of invoices. It is shown in Figure A.19: Invoice List.

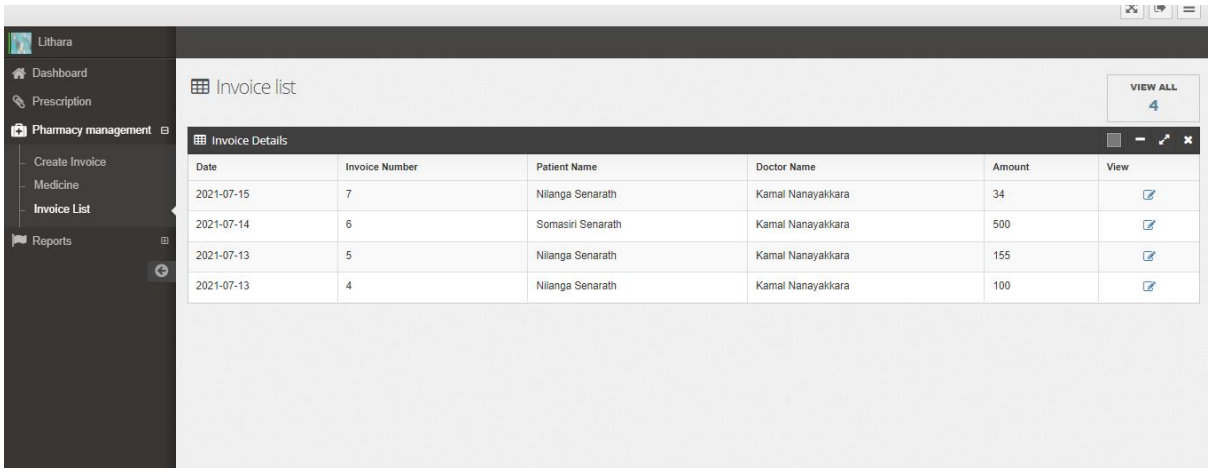


Figure A.19: Invoice List

Appendix B: Major Code Segments

B.1 Invoice Preparation

The invoice preparation code is used to make the invoice. The section is illustrated in Figure B.1: Invoice Preparation.

```
        $this->db->trans_start();

        if ($save_status != 'E') {

            $this->db->insert('tbl_invoice_header', $data);
            $insert_id = $this->db->insert_id();
            $iDrugID=$this->input->post('iDrugID', TRUE);
            $iQty=$this->input->post('iQty', TRUE);
            $dPrice=$this->input->post('dPrice', TRUE);
            for ($x = 0; $x < count($iDrugID); $x++) {

                $Drugandprice=$iDrugID[$x];
                $Drugandprice_array=explode("_",$Drugandprice);
                $drugID=$Drugandprice_array[0];
                $qtyprice=$Drugandprice_array[1];
                $price=$qtyprice*$iQty[$x];
                    $data2 = array(
                        'iInvoiceId' => $insert_id,
                        'iDrugId' => $drugID,
                        'iQuantity' => $iQty[$x],
                        'fUnitPrice' => $qtyprice,
                        'fPrice' => $price,
                        'cEnable'=>'Y'
                    );

                $updatesql="update tbl_drug set iQuantity=iQuantity-$iQty[$x] where id='$drugID'";
                $query = $this->db->query($updatesql);
                $this->db->insert('tbl_invoice_detail', $data2);
                $insert_id = $this->db->insert_id();
            }

        } else {
```

Figure B.1: Invoice Preparation

B.2 Invoice Calculation

The invoice calculation code is illustrated in Figure B.2: Invoice Calculation.

```
function cal_val(){
    var rowCount = $('#happy tr').length;
    var row_count=rowCount-3;
    var tot=0;
    for (var i = 0; i < row_count; i++) {
        var qty = $('#Qty_'+i).val();
        var value = $('#DrugID_'+i).val();
        if(qty!='' && qty!='undefined' && value!='' && qty!='undefined' && qty!='Select Drug'){
            const myArr = value.split("_");
            var price= myArr[1];
            $('#Price_'+i).val(qty*price);
            tot=tot+(qty*price);
        }
    }
    var dcharge = $('#dcharge').val();
    $('#totPrice').val(Number(dcharge)+Number(tot));
}
```

Figure B.2: Invoice Calculation

Appendix C: Test Case Documents

C.1 Functionalities in Logged Receptionist

The functional test case of the receptionist in the system is illustrated in Table C.1: Functionalities in Logged Receptionist.

Table C.1: Functionalities in Logged Receptionist

Test Case ID	Functionality	Procedure	Expected Results	Status
Functionalities in logged Receptionist				
12.	Dashboard	Click dashboard menu	Direct to the dashboard	Passed
13.	Go to make Appointment	Click Doctor Channel Click Make Appointment	Direct to Doctor Availability	Passed
14.	Load make appointment	Click book now	Direct to add an appointment	Passed
15.	Add appointment	Enter Doctor Enter Schedule date Enter Schedule time Schedule patient Click submit	Make new appointment	Passed
16.	Load Availability management	Click appointment Click Availability management	Direct to Schedule	Passed
17.	Load appointment details	Click appointment Click appointment details	Direct to appointment details	Passed
18.	Modify schedule	Click modify If wants, Enter Doctor Enter Schedule date Enter Schedule time Schedule patient Click submit	Saved modified one	Passed
19.	Delete Schedule	Click delete	Delete schedule	Passed
20.	Load patient	Click patient Details Click patients	Load Patients	Passed
21.	View all Patient	Click view all	Load patients list	Passed

C.2 Functionalities in Logged Doctor

The functional test case of the doctor in the system is illustrated in Table C.2: Functionalities in Logged Doctor.

Table C.2: Functionalities in Logged Doctor

Test Case ID	Functionality	Procedure	Expected Results	Status
Functionalities in logged Doctor				
22.	Dashboard	Click dashboard menu	Direct to the dashboard	Passed
23.	Go to make prescription	Click prescription	Direct to prescription	Passed
24.	Load prescription	Click prescription Add new	Direct to add prescription	Passed
25.	Add prescription	Enter Date Enter Doctor Enter patient Age Enter description Add medical reports Click submit	Make new prescription	Passed
26.	Load Availability management	Click appointment Click Availability management	Direct to Schedule	Passed
27.	Load All prescription details	Click View all	Direct to prescription details	Passed
28.	Modify prescription	Click modify If wants, Enter Date Enter Doctor Enter patient Age Enter description Add medical reports Click submit	Saved modified one	Passed
29.	Delete prescription	Click delete	Delete prescription	Passed

C.3 Functionalities in Logged Pharmacy Counter Clerk

The functional test case of the pharmacy counter clerk in the system is illustrated in Table C.3: Functionalities in Logged Pharmacy Counter Clerk.

Table C.3: Functionalities in Logged Pharmacy Counter Clerk

Test Case ID	Functionality	Procedure	Expected Results	Status
Functionalities in logged Pharmacy Counter Clerk				
30.	Dashboard	Click dashboard menu	Direct to the dashboard	Passed
31.	Go to View prescription	Click prescription	Direct to prescription	Passed
32.	Load prescription	Click prescription View all	Direct to prescriptions	Passed
33.	Load Drug	Click Medicine	Direct to Drug	Passed
34.	Add Drug	Enter Name Enter Qty Enter price Reorder level Click submit	Add new drug	Passed
35.	Load All Drug details	Click View all	Direct to Drug details	Passed
36.	Modify Drug	Click modify If wants, Enter Name Enter Qty Enter price Reorder level Click submit	Saved modified one	Passed
37.	Delete Drug	Click delete	Delete Drug	Passed
38.	Load Prescription to got to a new invoice	Click create invoice	Direct to prescription	Passed
39.	Load Create invoice	Click create invoice	Direct to create invoice	Passed
40.	Create invoice	Enter drug name, qty, price, dr charge and submit	New invoice created	Passed
41.	Modify invoice	Click modify If wants, Enter drug name, qty, price, dr charge and submit	Saved modified one	Passed
42.	Generate Drug Report	Click report Click Drug report	Direct report	Passed

C.4 Functionalities in Logged Supper Admin

The functional test cases of super admin in the system are illustrated in Table C.4: Functionalities in Logged Super Admin.

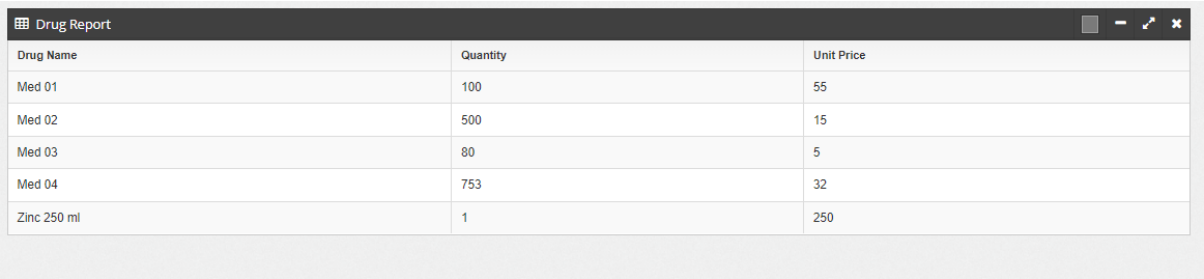
Table C.4: Functionalities in Logged Super Admin

Test Case ID	Functionality	Procedure	Expected Results	Status
Functionalities in logged Super admin				
43.	Dashboard	Click dashboard menu	Direct to the dashboard	Passed
44.	Access all privileges	Click menus	Direct to all modules	Passed
45.	Load user roles	Click user roles	Direct to system users	Passed
46.	Add user roles	Click add Enter name role Password Username Give privileges etc submit	Added new role	Passed
47.	Modify role	Enter modify Enter name role Password Username Give privileges etc submit	Modified	Passed
48.	Delete Role	Click delete	Deleted	Passed
49.	View role	Click View all	Viewed all roles	Passed
50.	Load Staff user	Click staff uses	Direct staff users	Passed
51.	Add staff	Click add Enter whole fields	Added new staff	Passed
52.	Modify staff	Click modify Enter whole fields, if wants	Modified	Passed
53.	Delete staff	Click delete	Deleted staff	Passed

Appendix D: Management Reports

D.1 Drugs Report

This can be used for long-term decision-making regarding the drugs. It is illustrated in Figure D.1: Drugs Report.



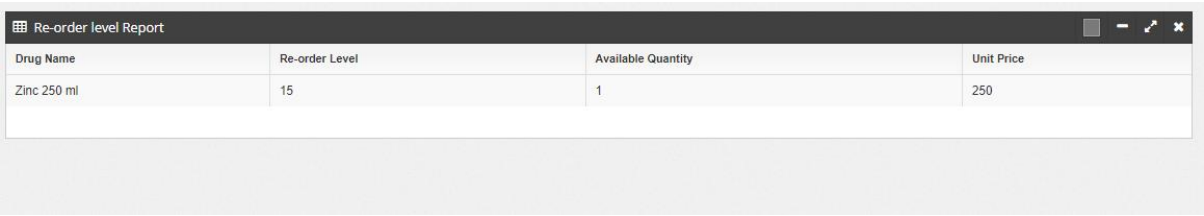
Drug Report

Drug Name	Quantity	Unit Price
Med 01	100	55
Med 02	500	15
Med 03	80	5
Med 04	753	32
Zinc 250 ml	1	250

Figure D.1: Drugs Report

D.2 Drugs Reorder Level Report

This can be used for long-term decision-making to keep effective drugs stocks without shortages. It is illustrated in Figure D.2: Drugs Reorder Level Report.



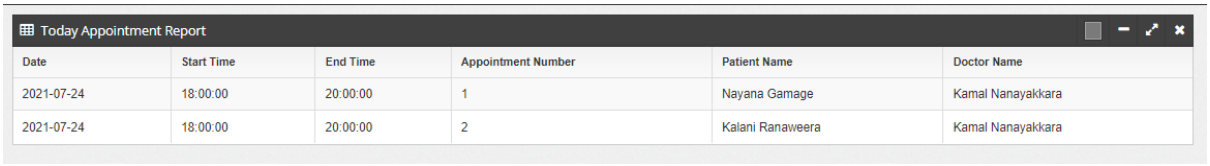
Re-order level Report

Drug Name	Re-order Level	Available Quantity	Unit Price
Zinc 250 ml	15	1	250

Figure D.2: Drugs Reorder Level Report

D.3 Today Appointments Report

This can be used for short-term decision-making to check today's patient queue effectively. This report is described in Figure D.3: Today Appointments Report.



Today Appointment Report

Date	Start Time	End Time	Appointment Number	Patient Name	Doctor Name
2021-07-24	18:00:00	20:00:00	1	Nayana Gamage	Kamal Nanayakkara
2021-07-24	18:00:00	20:00:00	2	Kalani Ranaweera	Kamal Nanayakkara

Figure D.3: Today Appointments Report

D.4 Patients Report

This can be used for long-term decision-making to keep records to identify the patient base. This report is described in Figure D.4: Patients Report.

Patients Report

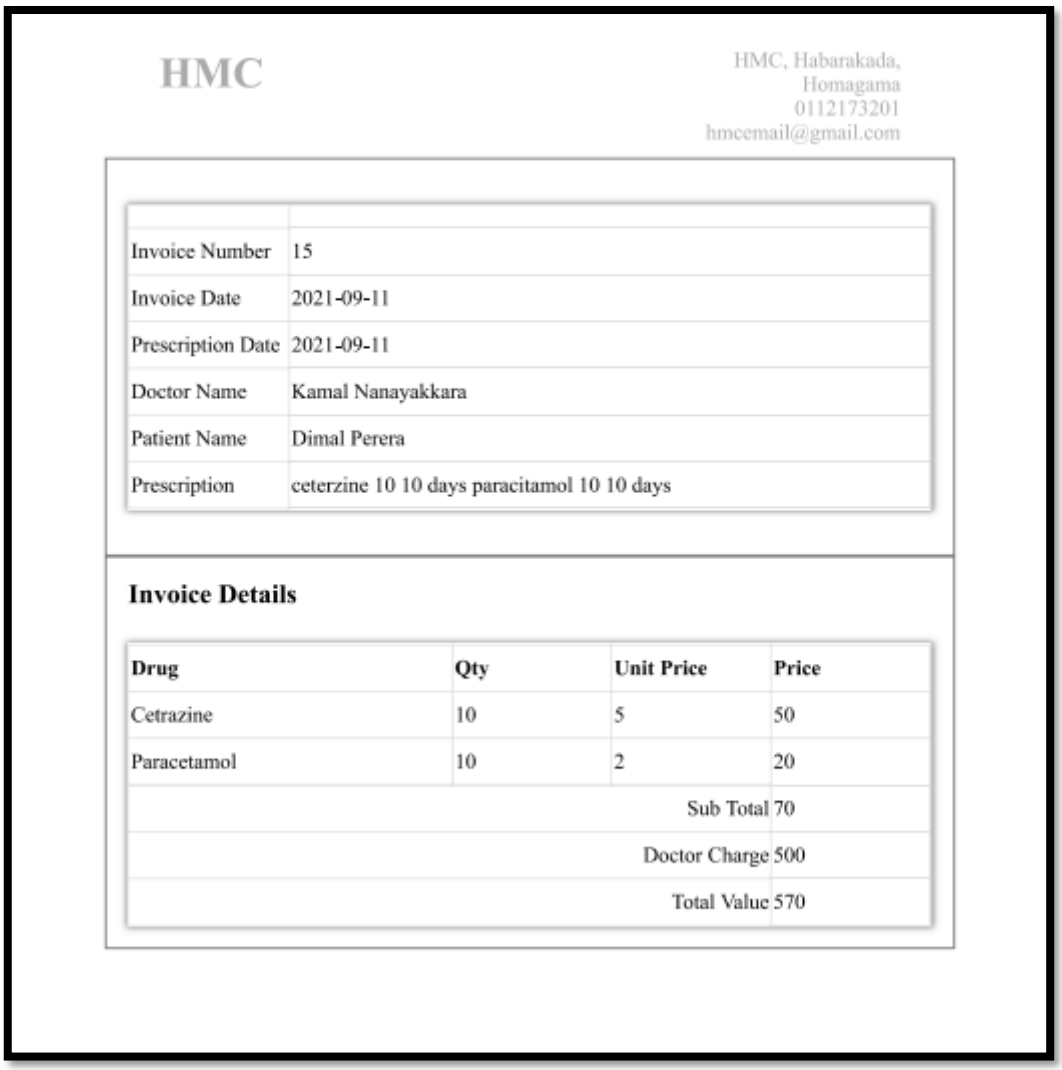
Patient Full Name	User-name	Patient Type	Phone
Aravinda Perera	aravinda	Patient	(077) 596-3258
Dimal Perera	dimal	Patient	(071) 111-1111
Iresha Samarawikrama	iresha	Patient	(071) 555-5555
Kalani Ranaweera	kalani	Patient	(077) 788-8888
Nayana Gamage	nayana	Patient	(077) 111-1111
Nilanga Senarath	nilanga	Patient	(077) 228-8414
Piyumali Jayasooriya	piyumali	Patient	(011) 244-4444
Ramya Rupasingha	ramya	Patient	(011) 222-2222
Randika Gamage	randika	Patient	(011) 777-7777
Somasiri Senarath	somasiri	Patient	(077) 666-6666
Umali Pathirage	umali	Patient	(077) 999-9999

Figure D.4: Patients Report

Appendix E: System Reports

E.1 Invoice

The invoice which the pharmacist gave to the patient. This is shown in Figure E.1: Invoice



HMC

HMC, Habarakada,
Homagama
0112173201
hmcemail@gmail.com

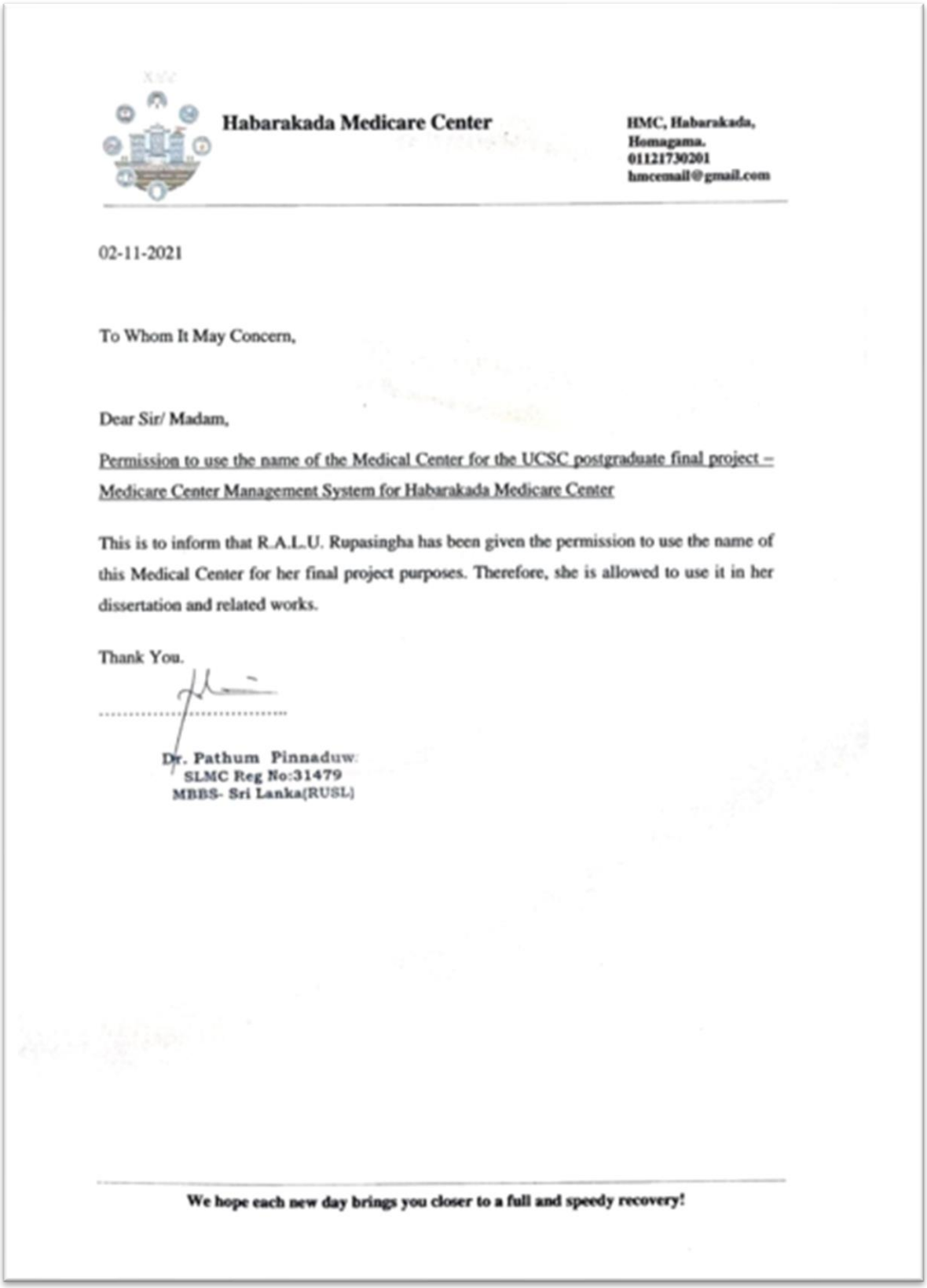
Invoice Number	15
Invoice Date	2021-09-11
Prescription Date	2021-09-11
Doctor Name	Kamal Nanayakkara
Patient Name	Dimal Perera
Prescription	ceterzine 10 10 days paracitamol 10 10 days

Invoice Details

Drug	Qty	Unit Price	Price
Cetrazine	10	5	50
Paracetamol	10	2	20
		Sub Total	70
		Doctor Charge	500
		Total Value	570

Figure E.1: Invoice

Appendix F: Medicare Center Consent



Habarakada Medicare Center

HMC, Habarakada,
Homagama.
01121730201
hmcemail@gmail.com

02-11-2021

To Whom It May Concern,

Dear Sir/ Madam,

Permission to use the name of the Medical Center for the UCSC postgraduate final project – Medicare Center Management System for Habarakada Medicare Center

This is to inform that R.A.L.U. Rupasingha has been given the permission to use the name of this Medical Center for her final project purposes. Therefore, she is allowed to use it in her dissertation and related works.

Thank You.

.....

Dr. Pathum Pinnaduwa
SLMC Reg No:31479
MBBS- Sri Lanka(RUSL)

We hope each new day brings you closer to a full and speedy recovery!

Figure F.1: Medical Center Consent Letter

Appendix G: Gannett Chart

No	Project Activity	Duration (Weeks)	2020																2021																											
			Sep				Oct				Nov				Dec				Jan				Feb				Mar				Apr				May				June				July			
			1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Planning	22																																										
2	Requirement and Analysis	22																																										
3	Designing	20																																										
4	Implementation	21																																										
5	Testing	20																																										
6	Project Closure	1																																										

Figure G.1: Gannett Chart