

# **Integrated Hospital Information System**

**G D S R Nanayakkara  
2019**



# **Integrated Hospital Information System**

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

**G D S R Nanayakkara  
University of Colombo School of Computing  
2019**



## 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: G D S R Nanayakkara

Registration Number: 2016/MIT/047

Index Number: 16550477

---

Signature:

Date: 04.10.2019

This is to certify that this thesis is based on the work of Mr. G D S R Nanayakkara under my supervision.

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

Certified by:

Supervisor Name: Dr H A Caldera

---

Signature:

Date:

## Abstract

This dissertation focuses on the experience gained by developing a hospital information system software.

Accurate, reliable and timely available information is vital in any industry. It allows people to make effective decisions as and when required. Healthcare is one of a significant industry as it directly interacts with human lives and, a hospital which is a key component of the healthcare sector requires to have such information when cooperating with patient care.

A study was carried out to identify the issues faced in the daily routine of a Sri Lankan hospital and found out that most of the processes are carried manually. Especially patient medical records maintained in manual documents made it difficult to access and retrieve information swiftly. Further, there are few standalone software systems operate in some departments of the hospital but those systems do not share information with other departments and results in duplication of important information.

Therefore, it was decided to develop an integrated hospital information system which intended to automate and streamline the patient care processes and distribute information among medical staff and patients.

Waterfall, the most traditional software development method was used as the development methodology. The requirement of the system was fixed hence the waterfall method was more suited for the project. Laravel PHP web framework was used to build the system. Laravel is based on model-view-controller (MVC) architecture which helps in improving the performance, allows better documentation, and has multiple built-in functionalities.

The web-based, user-friendly interface of the system made an easy to handle system for any user and enhanced the workload efficiency and improved the documentation and communication. The evaluation resulted in a favorable response from stakeholders but also suggested a few alterations to the system. Nevertheless, it is proved that the integrated hospital information system enhanced the productivity of a hospital in the most effective and efficient manner.

# Table of Content

<b>Abstract</b> .....	4
<b>List of Tables</b> .....	8
<b>List of Abbreviations</b> .....	9
<b>Acknowledgment</b> .....	10
<b>1. Introduction</b> .....	11
<b>1.1. Introduction</b> .....	11
<b>1.2. Motivation</b> .....	12
<b>1.3. Objectives</b> .....	13
<b>1.4. Scope of the Project</b> .....	14
<b>1.5. Non-Functional Requirements</b> .....	16
<b>1.6. Work Breakdown Structure</b> .....	17
<b>1.7. Structure of the Dissertation</b> .....	19
<b>1.8. Summary</b> .....	20
<b>2. Background</b> .....	21
<b>2.1. Evolution of Hospital Information Systems</b> .....	21
<b>2.2. Review of Similar Hospital Information Systems</b> .....	23
<b>2.3. Summary</b> .....	31
<b>3. Methodology</b> .....	32
<b>3.1. Software Development Life Cycle</b> .....	32
<b>3.2. Use Case Diagram</b> .....	34
<b>3.3. Sequence Diagram</b> .....	38
<b>3.4. Database Schema Diagram</b> .....	40
<b>3.5. Entity Relationship Diagram</b> .....	42
<b>3.6. Operating Environment</b> .....	43
<b>3.7. Summary</b> .....	45
<b>4. Evaluation</b> .....	46
<b>4.1. Evaluation Approach</b> .....	46
<b>4.2. Evaluation Results</b> .....	47
<b>4.3. Analysis of Results</b> .....	47
<b>4.4. Lessons Learnt and Failure to achieve objectives</b> .....	54
<b>4.5. Summary</b> .....	55

**5. Conclusion ..... 56**

**5.1. Summary of Results..... 56**

**5.2. Future Enhancements ..... 56**

**References ..... 57**

**Appendix A: Project Evaluation Form..... 58**

**Appendix B: Functional Specification ..... 60**

**Appendix C: Technical Specification ..... 76**

**Appendix D: User Manual ..... 90**

## List of Figures

Figure 1.1 Work Breakdown Structure.....	17
Figure 2.1 HospitalRun System.....	23
Figure 2.2 OpenHRS system.....	24
Figure 2.3 Vertikal System.....	25
Figure 2.4 eHospital System.....	26
Figure 2.5 Caresoft Hospital Information System.....	27
Figure 3.1 Patient Use Case Diagram.....	33
Figure 3.2 Doctor Use Case Diagram.....	34
Figure 3.3 Laboratory Use Case Diagram.....	35
Figure 3.4 Pharmacy Use Case Diagram.....	35
Figure 3.5 Nurse Use Case Diagram.....	36
Figure 3.6 Admin Use Case Diagram.....	36
Figure 3.7 Create New Patient Sequence Diagram.....	37
Figure 3.8 Prescribed Laboratory Test Sequence Diagram.....	38
Figure 3.9 Database Schema Diagram.....	40
Figure 3.10 Entity Relationship Diagram.....	41
Figure 3.11 Operating Environment.....	43
Figure 4.1 Look And Feel of The System.....	47
Figure 4.2 Ease Of Use of The System.....	48
Figure 4.3 Reliability of The System.....	49
Figure 4.4 Security of The System.....	50
Figure 4.5 Recommend to A Friend.....	51

## List of Tables

Table 4.1 Evaluation Results.....	46
Table 4.2 Look And Feel of The System.....	47
Table 4.3 Ease Of Use of The System.....	48
Table 4.4 Reliability of The System.....	49
Table 4.5 Security of The System.....	50
Table 4.6 Recommend To A Friend.....	51



## List of Abbreviations

HIS	Hospital Information System
OPD	Out Patient Department
HMS	Hospital Management System
MVC	Model View Controller
WBS	Work Breakdown Structure
OSS	Open Source Software
CSS	Closed Source Software
SDLC	Software Development Life Cycle
AI	Artificial Intelligence
CRM	Customer Relationship Management
SMS	Short Message Service
GUI	Graphical User Interface
UI	User Interface
UML	Unified Modelling Language
ERD	Entity Relationship Diagram
FTP	File Transfer Protocol
PHP	Hypertext Preprocessor
HTTP	Hypertext Transport Protocol
HTML	Hypertext Mark-up Language

## Acknowledgment

I owe my deepest gratitude to my supervisor Dr. Amith Caldera of University of Colombo School of Computing, without his continuous encouragement and support this project would hardly have been completed.

I would also like to thank the experts who were involved in the requirement gathering and validation phases of this project. Without their passionate participation and input, the development and completion of the system could not have been successfully conducted.

Finally, I must express my very profound gratitude to my family and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of development of the system and writing this thesis. This accomplishment would not have been possible without them.

# 1. Introduction

Introduction chapter describes the motivation for selecting the project, objectives, scope, work breakdown structure, non-functional requirements of the project and finally the structure of the dissertation document.

## 1.1. Introduction

In the modern business world, it is vital for the right information at the right time. It is evident that in the last couple of decades attempts has been made to develop systems which make information more precise, readily available and easily accessible throughout an organization. Encyclopaedia Britannica defines an information system as an integrated set of components for collecting, storing, and processing data and for providing information, knowledge, and digital products [1]. The development and use of information system is a modern trend which is primarily concerned with the collection, process, and dissimilation of useful information that directs an organization for better planning, better decision making and ultimately the better results.

In an organizational context, information became more like a basic resource like labor, material, and money. Information is the binding element that holds an organization together. On the contrary, since it is intangible, information is quite different for physical resources and is often difficult to interpret and utilize inefficiently in order to achieve desirable outcomes from the organization. Nevertheless, it is an integral part of an organization and should be properly managed to achieve the goals and objectives of the organization. Hence, it is important to acknowledge the importance of information in organizational performance.

Today, almost all industries use various kind of information systems. In the health sector, particularly hospitals around the world use different types of information systems to carry out and manage their activities. It benefits both patients and medical professionals in various aspects for instance easy access to patient medical records, efficient distribution of medical reports, easy coordination between different departments, etc. However, in Sri Lanka, medical information systems are not in use for all the areas of a hospital or they are not integrated with each other.

## 1.2. Motivation

According to various studies carried out around the world, it is revealed that paper-based medical records are far from an effective solution when timely information with respect to a patient's medical history can make the difference between life and death. Further medical records stored on papers, made it hard to coordinate and distribute accurate information regarding patient between different medical teams. Inturn it results in delays of the decision-making process.

A doctor usually writes the patient's diagnosis and prescriptions manually on paper. A patient has to preserve and take all documents when visiting the doctor next time. Also, a doctor has to go through the patient's medical history by paper records if it is required to decide the treatment which is obviously an inefficient, ineffective and time-consuming process.

During a medical emergency or in the event of an accident, patients are rushed to hospitals. Often, the medical staff is finding it difficult to discover the medical history of the patient and results delays in treatment.

A patient must visit the hospital pharmacy to purchase the prescribed medicines. Usually, a token is given and then the patient has to wait till the order getting prepared which is a time-consuming process. Further, when performing medical tests at laboratories, a patient has to provide the samples on one day and later on another day they have to visit the hospital again to collect the reports.

For some serious illness's patients are recommended to do an organ transplant. which is often a tedious task to find a suitable donor. Similarly, some people are willing to donate organs but there is no proper mechanism to register and notify such persons.

Above all observations suggested the importance of a single software system which could cater to different stakeholders of a hospital. Therefore, it was encouraged to design and develop an integrated information system for a hospital.

### 1.3.Objectives

The project “Integrated Hospital Information System” was designed to achieve the following objectives.

- Increase electronic medical records coverage of a hospital.
- Ensure availability of timely updated and accurate information for patient care.
- Improve quality and efficiency in service delivery, governance, accountability and effective use of resources of a hospital.
- Improve the capacity of health authorities to detect emerging and re-emerging diseases and patterns to take necessary preventive actions.

## 1.4.Scope of the Project

- Facilitate to keep patient medical history in a smart card/bar code
  - Patients are registered in the system with basic medical details such as blood group, allergies, medical history, etc and provided a smart card/barcode along with a username and password to access the system.
  - Patients will be registered with their national identity card number and upon successful registration, the user will be notified via an SMS. user has to produce the smart card/bar code in every action he/she interacts with the hospital (eg: bill payments, laboratory services, pharmacy services, etc)
- Track patient medical history and monitor progress of illnesses
  - Each time a patient visits a doctor, smart card/bar code will be scanned by the support staff member and the patient's medical records will automatically be displayed on the doctor's computer screen. New diagnosis details and prescription details will be recorded in the system.
- Remind patients for upcoming events (Eg: appointments and vaccinations )
  - Some patients (e.g. pregnant mothers, babies, diabetic patients, etc) are required to meet the doctor regularly on a schedule.
  - On such occasions patient, doctor or support staff member can set up the forthcoming appointment/event in the system.
  - Due dates will be notified via either email or SMS alert.
- Link with hospital pharmacy and alert patient when the prescription is ready to collect drugs.
  - The doctor will enter the prescription on the system, and a patient can decide whether to purchase medicine from the hospital pharmacy or outside. If the patient decides to purchase from the hospital pharmacy, upon submission it will direct to the hospital pharmacy and once the medicines are ready to collect, the patient will receive an SMS with bill amount.
  - Otherwise patient can get a doctor signed printout of the prescription.
- Allow patients to get their lab reports online
  - Once the lab reports are ready, the patient will get an SMS and he/she will also be able to view the reports online.
  - Reports will link to patients records as well.

- Develop donor/requester module
  - This module will facilitate both blood and human organ donations and requests.
  - Both donor and requesters have to register in the system with some medical details such as blood group, certain diseases history, organs willing to donate, etc
  - As and when there is a request, the system will automatically match the request details with the donor details and send all suitable donors an email/SMS.

## 1.5. Non-Functional Requirements

- Performance requirements

Transaction time: less than 15 seconds (does not include back-end response times)

Network connectivity speed: 1 Mbps

- Security

The data collected and maintained through the application are of a sensitive nature. Therefore, proper safeguards should be followed and access needs to be controlled.

- Software quality attributes

- Availability:

Architecture/Design to support 99% Availability.

System downtime: 30 min per day and uptime the remaining.

- Scalability

100 concurrent https/http requests.

- Interface

The human-computer interfaces should comply with standard graphical user interface design and development best-practices of human-computer interaction. Software interfaces will conform to notions of modularity, consistency, and standardization.

Need to support Web 2.0 standards to increase usability.

- Usability

Any user proficient in using a web browser should be able to use the web application without much difficulty.

- Reliability

The system will be based on proven programming models, web servers and object-oriented concepts. Hence, it should be dependable, robust and safe to the extent of the maturity of these state of the art technologies.



## 1.6. Work Breakdown Structure

A work breakdown structure (WBS) is a key project deliverable that organizes the project work into manageable sections. The Project Management Body of Knowledge (PMBOK) defines the work breakdown structure as a "deliverable oriented hierarchical decomposition of the work to be executed by the project team."

The work breakdown structure visually defines the scope into manageable portions that a project team can understand, as each level of the work breakdown structure provides further definition and detail.

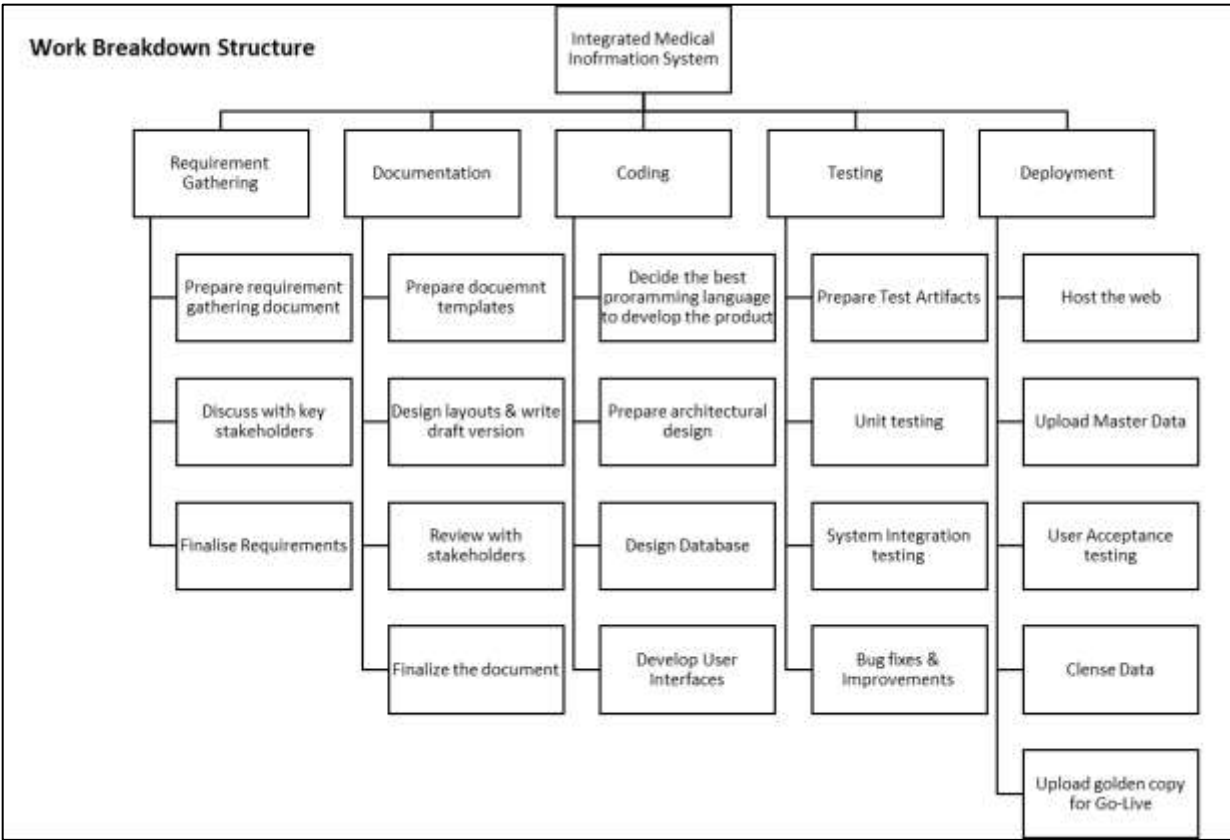
Figure 1.1 work breakdown structure displays the WBS of the hospital information system. Requirement gathering, documentation, coding, testing, and deployment were the main components of the work breakdown structure of the project and represents the first level of the structure.

According to Figure 1.1, requirement gathering was subdivided into three levels namely prepare requirement gathering document, discuss with key stakeholders and finalize requirements. Documentation component expanded into four sublevels which comprised of preparation of functional specification, screen designs, review and finalize with stakeholders.

Coding segment comprised of activities related to coding the system. Selection of coding language, system architecture, design user interfaces and design of database structures included in this section. System testing section depicted the unit testing and integration testing related activities.

Finally, the deployment segment categorized the activities related to the deployment of the Hospital Information System at a client site. User acceptance testing, cleanse and upload master data and the go-live of the system were the important activities of this segment.

Thus, Figure 1.1 work breakdown structure provided a comprehensive breakdown of the project work.



*Figure 1.1 Work breakdown structure*

## 1.7. Structure of the Dissertation

Chapter 2 discussed the evolution of health information systems, review of similar systems and finally discussed the importance of the proposed solution.

Analysis of the problem, functional and non-functional requirements with use case diagram, sequence diagram, database schema diagram, and entity relationship diagram were discussed in Chapter 3. Further, it discussed the operating environment of the system.

Chapter 4 presents the evaluation techniques used to evaluate the product and it further analyses the feedback received from evaluators. It also discussed the lessons learned during the project and object which were unable to achieve.

Final Chapter provides the conclusion with the summary of results and expected future enhancements of the hospital information system.

## 1.8. Summary

Introduction chapter explained the reasons for selecting the project and issues or problems which are trying to solve by the system, objectives of the projects were also explained in this chapter. Both functional and non-functional requirements were explained, and components of the work breakdown structure were explained. To conclude the chapter, the remaining chapters of the dissertation document was briefly described.

## 2. Background

Background chapter discusses the evolution of the hospital information system and then explains existing similar information systems and the advantage of proposed systems against the existing systems.

### 2.1. Evolution of Hospital Information Systems

Health information management is defined as the collection and analysis of healthcare data to provide information for health care decisions involving patient care, institutional management, health care policies, planning, and research.

The earliest forms of medical records were narratives written by ancient Greeks to document successful cures, share observations about symptoms and outcomes, and teach others who provided medical advice through these case studies.

As healthcare advanced, physicians realized that the best way to continue improving diagnosing and treating illnesses was to carefully document observations and actions while treating patients – and share this information as a way to teach other health professionals.

As early as 1600, physicians offered advice on how to present information in a medical record, but it wasn't until 1928 that the American College of Surgeons (ACOS) took steps to standardize the growing number of medical records by establishing the American Association of Record Librarians (AARL). "Record librarians" was the term used because early medical records were documented on paper [2].

Standardization of medical records and growth of complete record-keeping continued from the 1920s through the 1960s, but records were paper-based. The development of computers presented the opportunity to maintain records electronically, but the expense of purchasing and maintaining a mainframe, and the expense associated with storage of data, meant that only the largest organizations could use technology to handle medical records. The 1960s also saw the introduction of Medicare and Medicaid, which required nurses to collect data to document care for reimbursement. While computers were increasingly used for accounting and billing functions, the use of computers to collect and manage medical records was not common [3].

In the 1970s, as computers became smaller, software designed to support clinical functions for pharmacy, clinical laboratory, patient registration, and billing began to increase. The

disadvantage of these health information systems was their department-specific functions – they were not accessible by other departments.

The first attempt at a total, integrated health records system was implemented in a gynecology unit at the University Medical Center in Burlington, Vermont in 1971. Based on the problem-oriented medical record, the system was patient-oriented – all disciplines included in care made notes in the record to provide an overview of care to see the relationship between conditions, treatments, costs, and outcomes.

Because personal computers and widespread health-related software applications had grown in popularity, hospital information technology (IT) staff were tasked with the responsibility to integrate multiple, disparate systems. As network solutions were developed, IT departments were able to connect financial and clinical systems – for limited functions.

As competition in healthcare created a consolidation of individual hospitals to form health systems, the need for integration grew. Technology advances gave hospitals access to computing systems that could share information across disparate systems to set the stage for data-sharing [4].

The importance of integrated electronic health records (EHRs) to enable providers to make better decisions grew, and more hospitals and physicians implemented them to reduce the incidence of medical error by improving the accuracy and clarity of medical records.

Increased focus on value-based care as opposed to fee-based care and a drive to improve patient outcomes propel the growing accumulation of data to support clinical as well as operational decisions in health care.

Just as clinicians in the 1920s understood the importance of previous health records as learning tools that would improve outcomes, healthcare professionals leverage data to enhance care on a larger scale — using tools that analyze population health data.

## 2.2. Review of Similar Hospital Information Systems

It was found that standalone hospital-section (eg: OPD, pharmacy, laboratory, etc) management systems are available and used in some of the leading private hospitals as well as few government hospitals of Sri Lanka. However, most of those systems were operating in isolation and hence the users did not get the full benefits of those systems.

Following are a few examples,

- District Hospital, Dompe

Recently the e-Hospital concept was introduced to the district hospital of Dompe [5] by the health ministry of Sri Lanka with collaboration with Information and Communication Technology Agency of Sri Lanka (ICTA). The first hospital of its kind in Sri Lanka, this pioneering healthcare institution has embraced the concept of digital health, which means that it now uses information technology to improve the efficiency, competency, and quality of the services it offers. It facilitates patient registrations, records their medical history, medicine details, queue management, etc.

- Asiri Laboratories, Colombo

Allows patients to view their lab reports online. System generated user name and password are provided with the laboratory bill.

- Lanka Hospitals, Colombo

Does register a few categories of patients (eg: pregnant mothers, babies born in the hospital, etc) in their system. Bar code printed card issued however it does not record medical history rather contain patient's personal details.

There is a number of open-source and bespoke or off the shelf software solutions available in the market to use in hospitals.

For example :

### Open Source

- HospitalRun

HospitalRun is an open-source project [6], incubated and led by CURE International's technology team, aimed at delivering modern software for healthcare facilities in the developing world. It utilizes modern browser-based software design and a user-friendly interface to create an intuitive system. It is also architected to deal with the issues of Internet reliability in a developing world context and is designed to allow records to be securely carried outside of the hospital for mobile clinic activities.



*Figure 2.1 HospitalRun System*

- OpenMRS

OpenMRS is both a software and a community. As software, it serves as an electronic medical record system (EMR) originally designed for developing countries. Through its open-source community, it has grown into a medical informatics platform used on every continent.



OpenMRS is a software platform and a reference application which enables the design of a customized medical records system with no programming knowledge (although medical and systems analysis knowledge is required) [7].

It is a common platform upon which medical informatics efforts in developing countries can be built. The system is based on a conceptual database structure which is not dependent on the actual types of medical information required to be collected or on particular data collection forms and so can be customized for different uses.

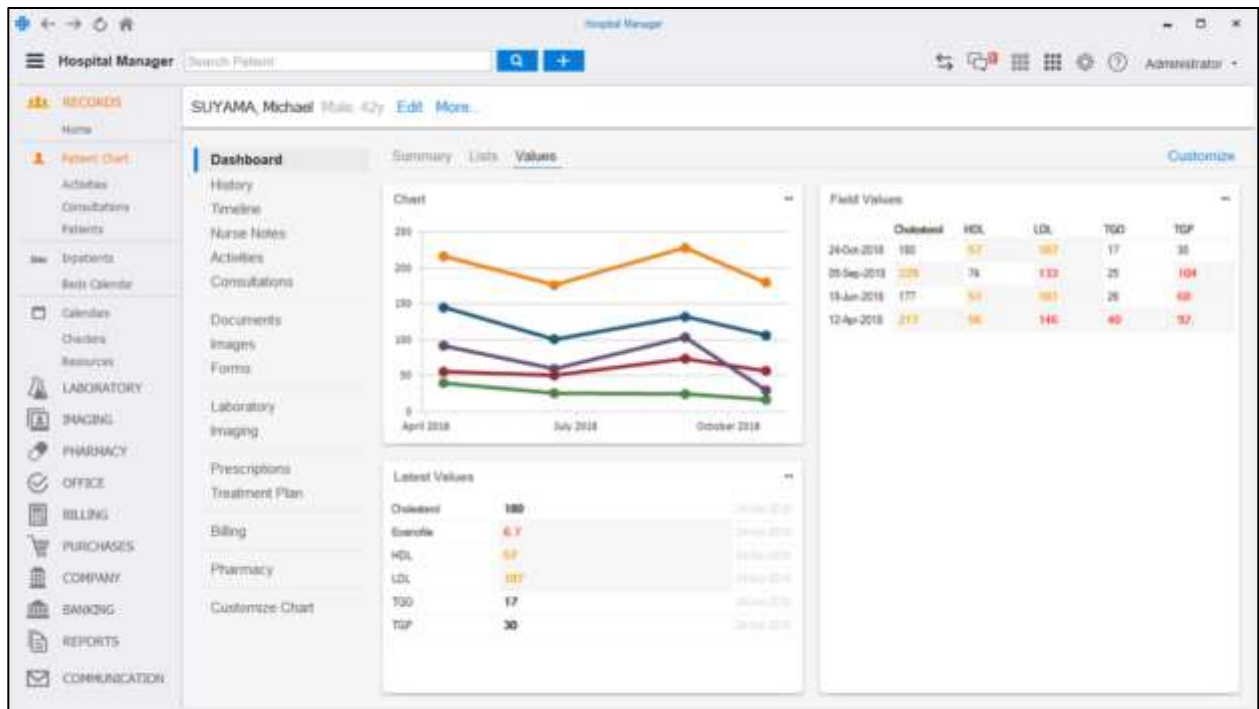


*Figure 2.2 OpenMRS System*

## Off the Shelf Software

- VERTICAL Systems

Designed for clinics and hospitals, it is an on-premise medical billing solution that helps manage patient records, billing, scheduling and financials on a centralized platform [8]. It also supports document management, medical scheduling, medical workflows, medical billing and claims, inventory management, ward management, nurse module, laboratory, and pharmacy.



*Figure 2.3 Vertikal System*

- eHospital System

eHospital Systems is a customizable, comprehensive and integrated hospital management system designed to manage all hospital operations [9]. Hospitals, it can be used by healthcare facilities, multi-specialty clinics, and medical practitioners. The customizable alert system sends the text and email reminders and improves the quality of patient care.

The software is user-friendly and with no error that is usually associated with handwriting. Pulling up server or cloud information is now easily done with new technological computer systems, yielding optimal performance. Patients can find doctors and book an online appointment based on specialty, rating, fees, and availability. Organizing doctor schedules, collating patient notes, and handling payment are effortless. It is developed based on the best practices around the world.



*Figure 2.4 eHospital System*

- Caresoft Hospital Information System

Caresoft – hospital information system is application software that is used for complete automation of hospitals enabling enhancement inpatient care, resource and information management leading to superior clinical and financial outcomes [10]. It helps health care enterprises in delivering enhanced patient care, optimizing resource management using Business Intelligence (BI) tools. It also helps in lowering the cost of operations and facilitates effective decision making.

Key Features:

- Patient Information Management
- Appointment Management
- Electronic medical records
- Billing and Revenue Cycle Management
- Doctor Schedule and Appointments
- Modality (Diagnostic) Schedule and Appointments
- In-patient Management System
- Lab Information System



*Figure 2.5 Caresoft Hospital Information System*

### **Comparison of alternative design strategies**

Open-source software (OSS) refers to the software which uses the code freely available on the Internet. The code can be copied, modified or deleted by other users and organizations. As the software is open to the public, the result is that it constantly updates, improves and expands as more people can work on its improvement [11].

Closed source software (CSS) is opposite to open source software and means the software which uses the proprietary and closely guarded code. Only the original authors of software can access, copy, and alter that software. Users are not purchasing the software but only pay to use it [11].

- **Price Policy**

Open source often referred to as a free of cost software. It can, however, have costs for extras like end-user support, additional services or added functionality. Thus, you may still pay for a service with open-source software. Closed source software is usually a paid software. The costs can vary depending on the complexity of the software.

- Security

The code of open source software can be viewed, shared and modified by the community, which means anyone can fix, upgrade and test the broken code. The bugs are fixed quickly, and the code is checked thoroughly after each release. However, because of availability, the source code is open for hackers as well.

On the contrary, closed source software can be fixed only by a vendor. If something goes wrong with the software, Customer has to send a request and wait for the answer from the support team. Solving the problem can take much longer than compared to Open Source.

- Usability

Usability is a painful subject of open-source software. User guides are written for developers rather than to layperson users. Also, these manuals are failing to conform to the standards and structure.

For closed source software usability is one of the merits. Documentation is usually well-written and contains detailed instructions.

It is apparent that each of the platform types has its own philosophy, methodology, advantages, and disadvantages. Therefore there is no univocal option as it depends on business needs.

When comparing both open source and off-the-shelf hospital information systems, it was evident that open source software solutions provided a basic features such as patient management, appointment scheduling, and pharmacy management while off-the-shelf software's offer more options, for instance, document management, medical scheduling, medical workflows, medical billing and claims, inventory management, ward management, nurse module, laboratory, and pharmacy.

Further following differences were identified among the open-source and off-the-shelf health information systems.

- Open-source software's were free of charge whereas the off the shelf software's were priced and price varies according to required modules. Further separate charges are applicable for support services and user licenses.

- Open-source software provides a large amount of flexibility and freedom to change the software without restriction. This innovation, however, may not be passed on to all users and it is debated whether customized changes to the original source code can limit the future support and growth of the software. Unlike open-source, proprietary software also attracts larger amounts of research and development to regularly offer new products and upgrades.
- Usability is often a major area of criticism for open source software because the technology is generally not reviewed by usability experts and caters to developers rather than the vast majority of new users. For off-the-shelf software, usability is a high selling point due to expert usability testing for a more targeted audience. User manuals are also provided for immediate reference and quick training, while support services help to maximize the use of the software.
- Off-the-shelf software is generally seen as more secure because it is developed in a controlled environment by a concentrated team with a common direction. Whereas open source software is not always developed in a controlled environment, therefore, those might vulnerable to security threats.

However, none of the ready-made solutions are provided an integrated solution as per the proposed solution and none of them failed to provided reminders for upcoming vaccination or appointments and organ donor/requester facilities. The proposed system would integrate the information systems used in different sections of a hospital and hence it will be convenient to users when performing their tasks and find information using a single system.

The proposed solution clearly had the upper hand compared to the existing information systems designed for hospital management.

## 2.3. Summary

This chapter explained how the hospital information systems evolved over the years. It explained how the necessity arose for information systems and how the systems grew gradually. This chapter then focused on available similar information systems. It discussed both open source and off the shelf software, their advantages and disadvantages and finally the competitive edge of the proposed system over the existing systems.

### 3. Methodology

Methodology chapter elucidates the software development best practices used for product development. Software development life cycle used for the project, relevant unified modeling language (UML) diagrams and components of the operating environment of the system are elaborated throughout the chapter.

#### 3.1. Software Development Life Cycle

The waterfall model is a popular software development process. In “The Waterfall” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. That is any phase in the development process begins only if the previous phase is completed.

The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of requirement gathering and analysis, system design, implementation, testing, deployment of system and maintenance.

Waterfall model is recommended to be applied if the project requirements are clearly defined. Functional and non-functional requirements were agreed and finalized when the project was selected and therefore the waterfall model was used as the guidance to develop the product.

#### **Sequential phases of the waterfall model with respect to the hospital information system**

- **Requirements:** The first phase involved understanding what needed to be designed and what should be its function and purpose. Here, the specifications of the input and output or the final product are studied and marked. Therefore, a doctor, patient and a director of a private hospital were contacted and discussed their experience, processes, requirements, and suggestions.
- **System Design:** The requirement specifications from the first phase were studied and system design was prepared. System design helped in specifying hardware and system requirements and helped in defining overall system architecture.



- **Implementation:** With inputs from system design, the system was first developed in small programs, which were integrated with the next phase. Each unit was developed separately and tested for its functionality. Patient module, doctor module, lab module, etc were developed as individual modules and unit testing was carried out.
- **Integration and Testing:** All the units developed in the implementation phase were integrated into a single system after testing of each module. The designed software went through constant integration testing to find out if there were any flaw or errors. Testing was done so that the client does not face any problem during the installation of the software.
- **Deployment of System:** Once the functional and non-functional testing was successfully completed, the product was deployed in the customer environment.
- **Maintenance:** It is expected to make modifications to the system or an individual component to alter attributes or improve performance after deployment of the product. These modifications are expected to arise either due to change requests initiated by the customer, or defects uncovered during live use of the system.

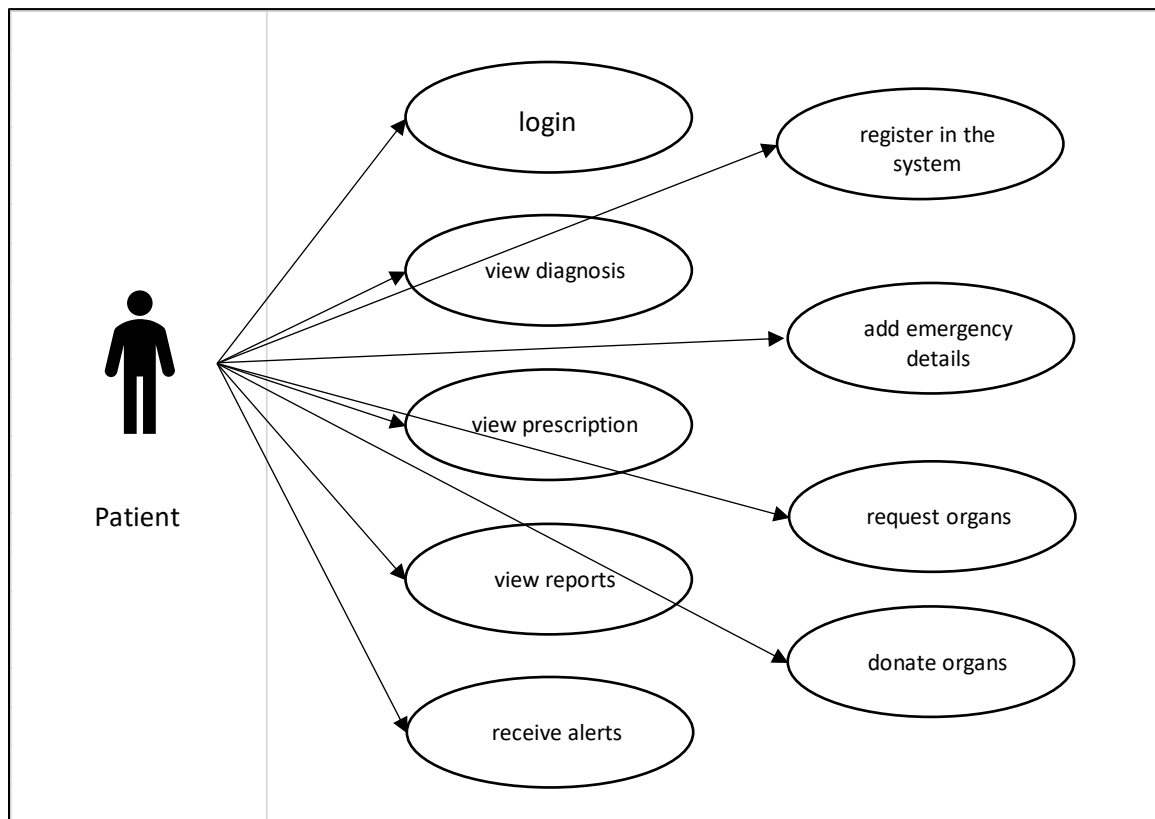
### 3.2. Use Case Diagram

A use case diagram is a graphic representation of the interactions between the elements of a system [12]. That is, it is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior and not the exact method of making it happen.

A key concept of use case modeling is that it helps to design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior. It summarizes some of the relationships between use cases, actors, and systems

Following use case diagrams display the use cases drew for each module. Six actors were identified, and their behavior was illustrated in each diagram.

#### 3.2.1. Patient



*Figure 3.1 Patient Use Case Diagram*

Activities of a patient-user are defined in Figure 3.1 patient use case diagram. A patient would register in the system and then update the system with doner status. View diagnosis and prescription details, view reports and receive alerts for different events were also activities that permitted to a patient-user.

### 3.2.2. Doctor

A doctor user would be able to login to the system, view appointments and then would also be able to write diagnosis and prescriptions for a patient. As per Figure 3.2 doctor use case diagram, a doctor would be able to request laboratory reports and set upcoming event alerts as well.

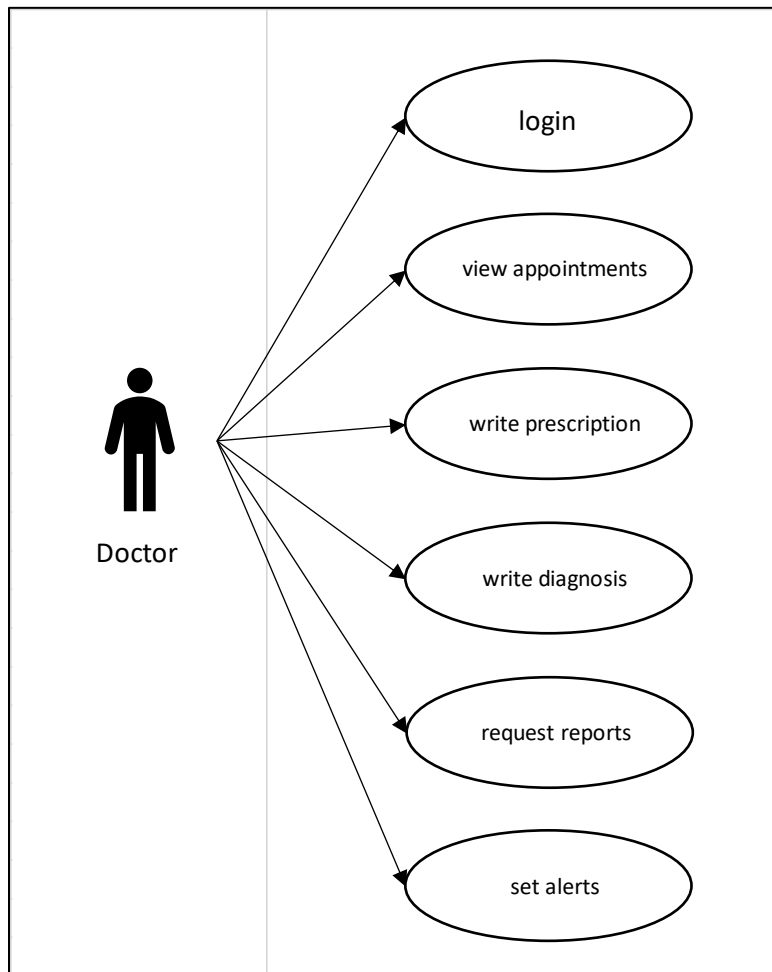
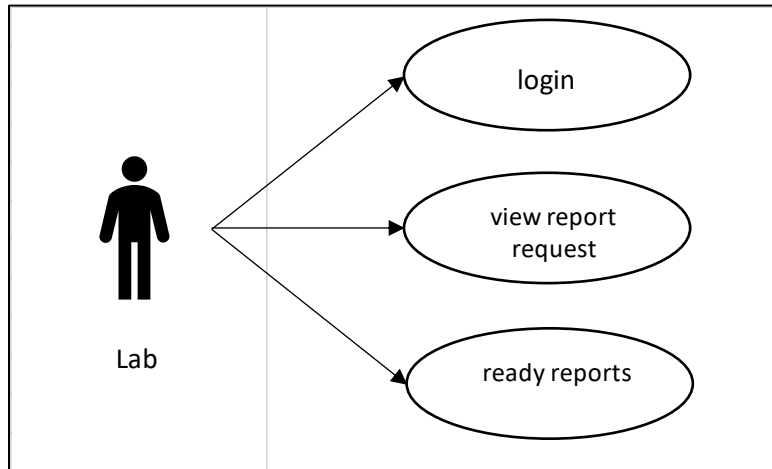


Figure 3.2 Doctor Use Case Diagram

### 3.2.3. Laboratory

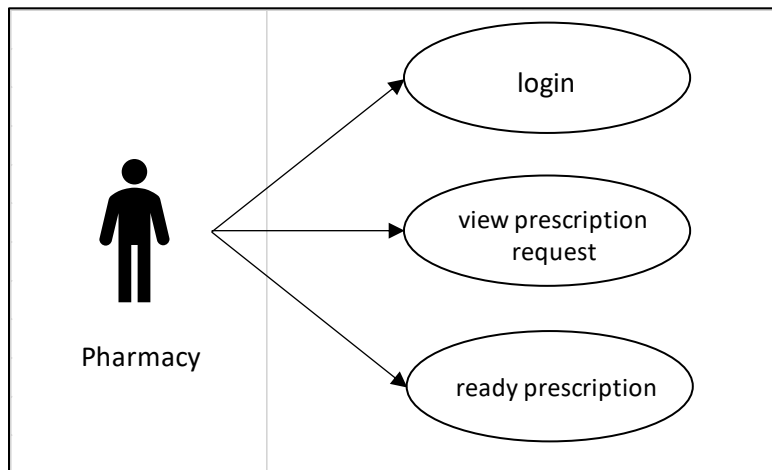
Figure 3.3 laboratory use case diagram explained the actions of a laboratory user. The user would be able to login to the system, view requested reports and then once the reports are ready to collect, alert the user.



*Figure 3.3 Laboratory Use Case Diagram*

### 3.2.4. Pharmacy

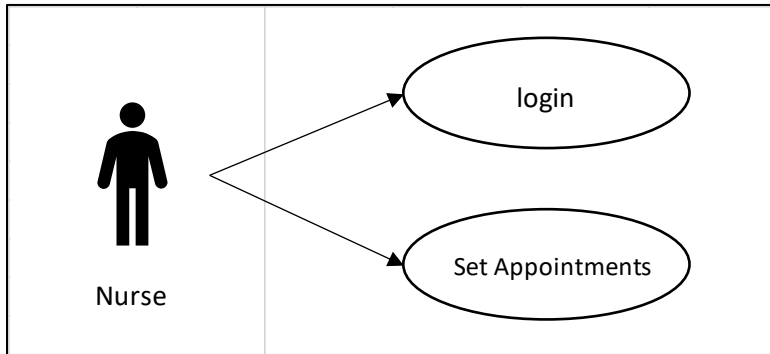
Pharmacy-user acts similar to a laboratory user. As per Figure 3.4 pharmacy use case diagram, the user would be able to login to the system, view prescriptions and then once the drugs are ready to collect, alert the user.



*Figure 3.4 Pharmacy Use Case Diagram*

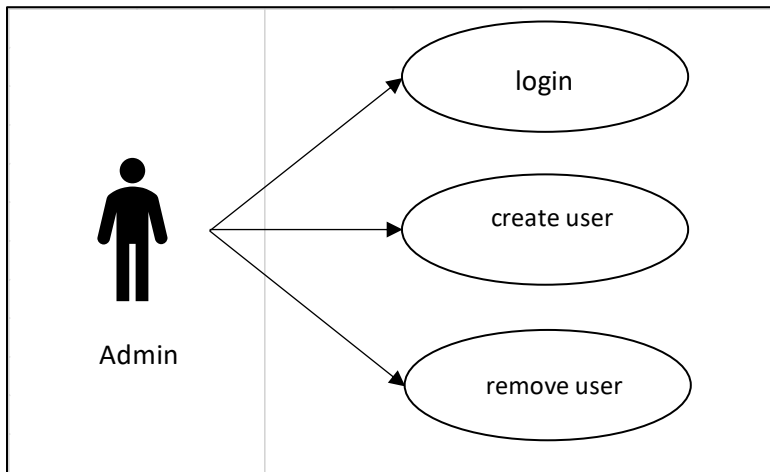
### 3.2.5. Nurse

A nurse-user would log in to the system and would create appointments for doctors



*Figure 3.5 Nurse User Case Diagram*

### 3.2.6. Administrator



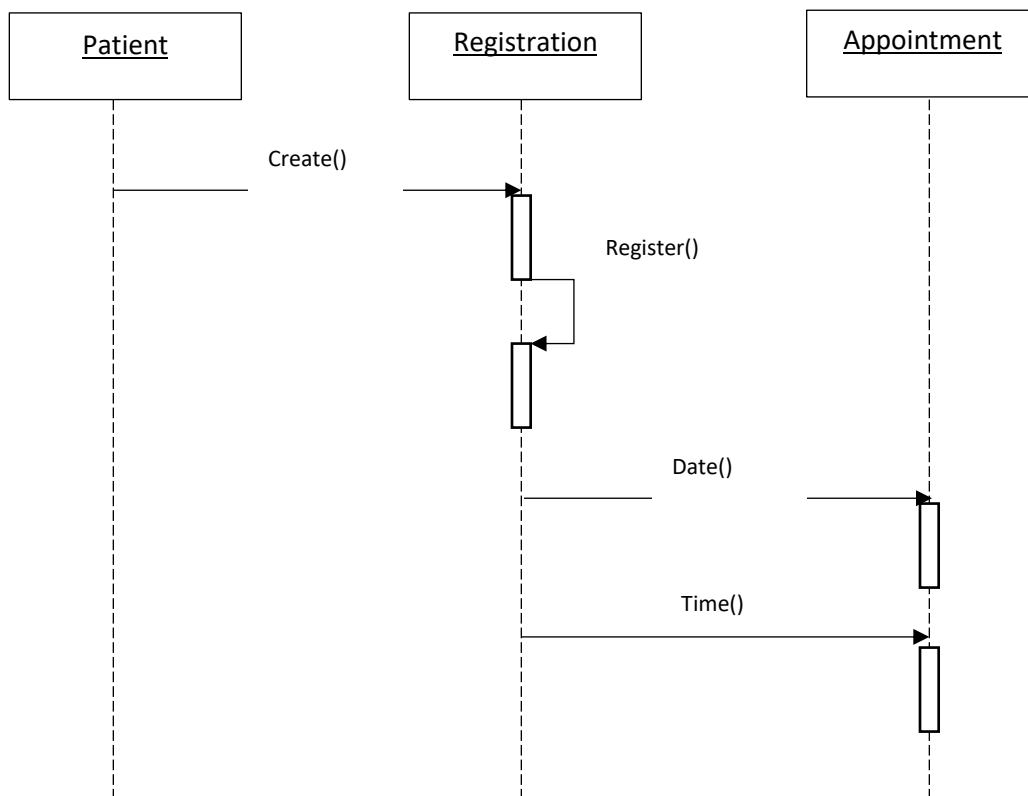
*Figure 3.6 Admin Use Case Diagram*

Figure 3.6 elucidates the actions permitted to the system administrator. Admin-user would create other users in the system and would also be able to remove users including patients.

### 3.3. Sequence Diagram

The sequence diagram is used primarily to show the interactions between objects in the sequential order that those interactions occur. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

- Create new patient and make an appointment

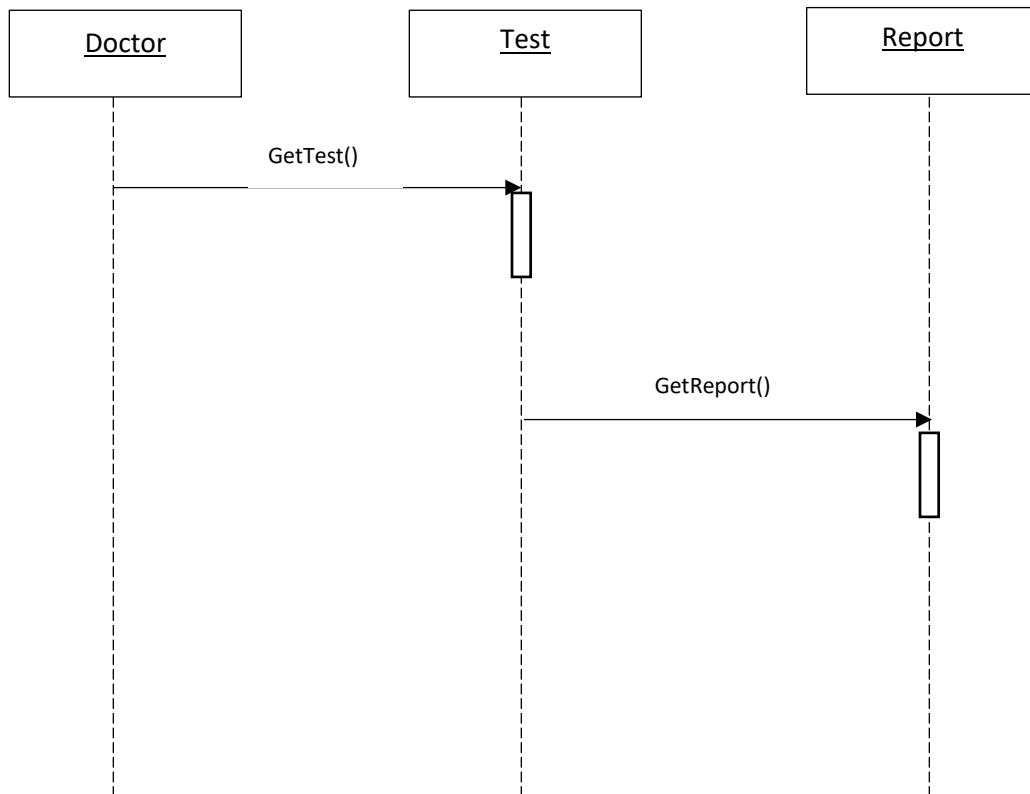


*Figure 3.7 Create New Patient Sequence Diagram*

Above Figure 3.7 create new patient sequence diagram illustrates the time sequence diagram as and when a new patient walks into the hospital. A patient will be registered in the system and a system-generated bar code will be issued. An appointment to visit a doctor with date and time will also be entered into the system.

- Prescribed Laboratory Test

Following Figure 3.8 prescribed laboratory test sequence diagram depicts the sequence diagram of the prescribed laboratory test report. A doctor recommends a test to be done. If the patient decides the test to be done from the hospital's laboratory, then it will appear in the laboratory module. Results will be shown in the patient's module under the doctor prescription.



*Figure 3.8 Prescribed Laboratory Test Sequence Diagram*

### 3.4. Database Schema Diagram

A database schema is a visual and logical architecture of a database created on a database management system. It provides a graphical view of the entire database architecture and structure. It provides a means for logically grouping and displaying database objects such as tables, fields, functions, and relations.

The system was designed with multiple database tables as shown in the Figure 3.9 database schema diagram. Patient details are stored inpatient database table where patient\_id is the primary key. Similarly, all other database tables were created with suitable primary keys. Relationships between each table were established.



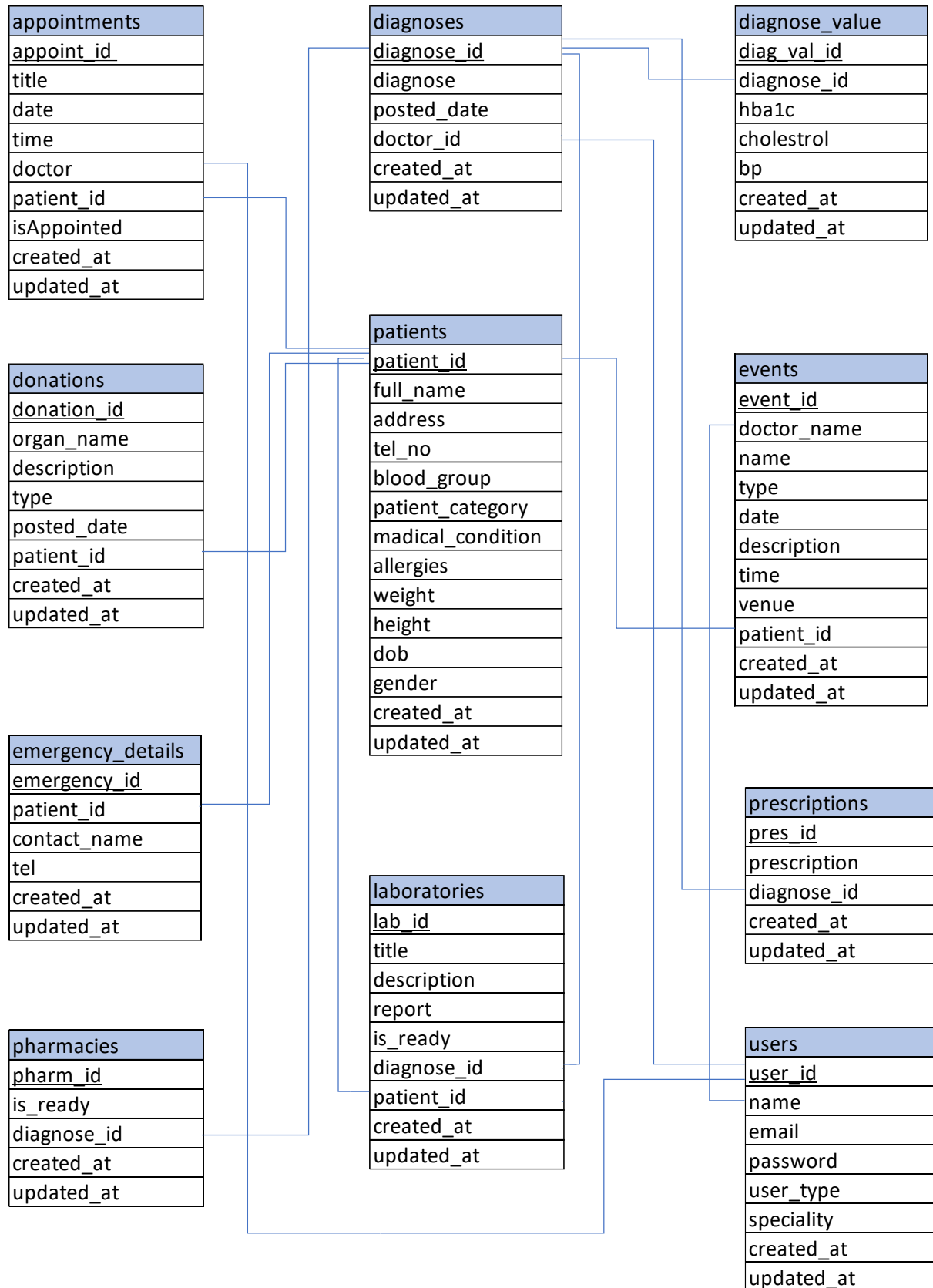
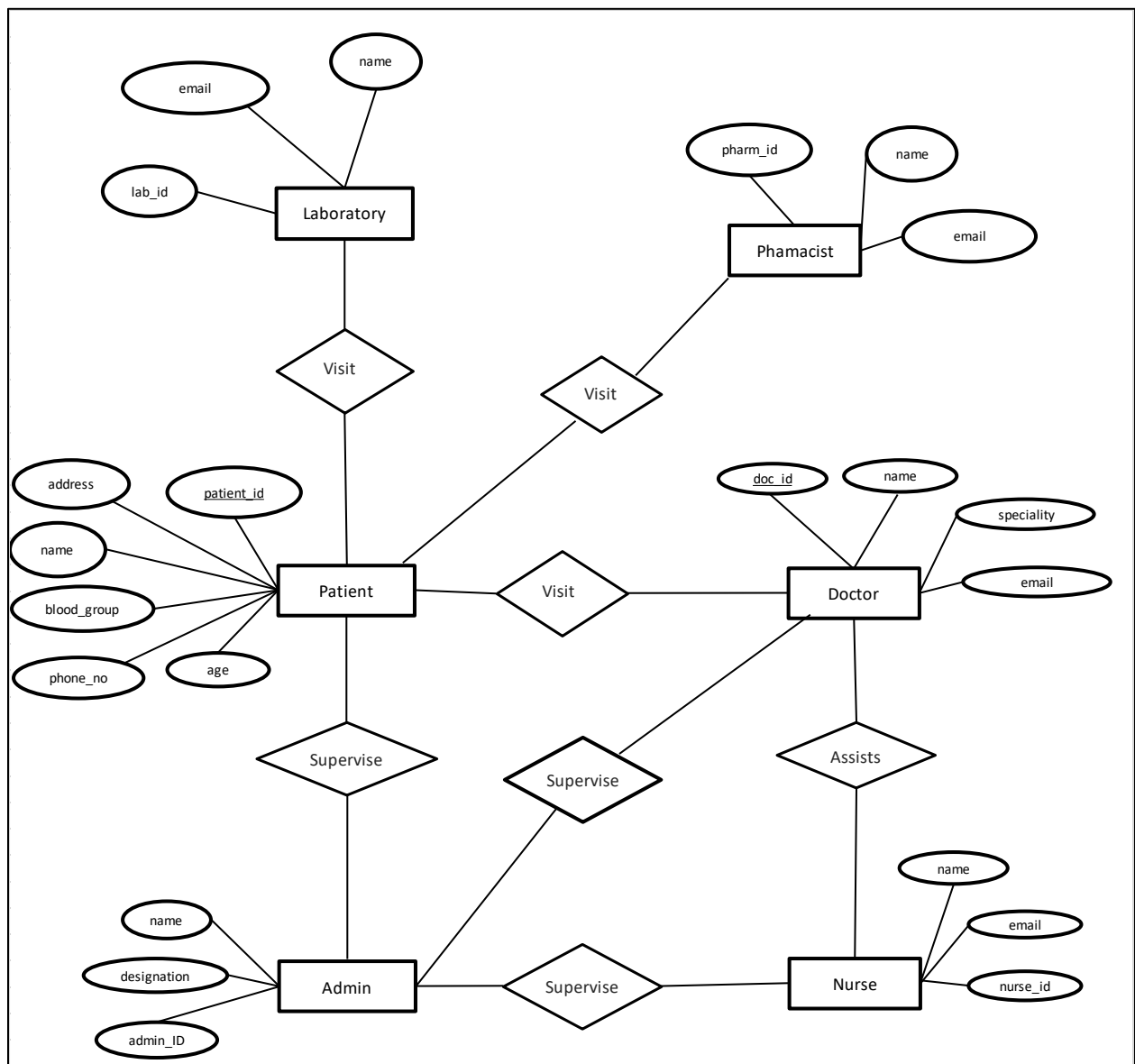


Figure 3.9 Data Base Schema Diagram

### 3.5. Entity Relationship Diagram

An Entity-Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system [13]. The entity-relationship diagram displays the relationships of entity sets stored in a database.

An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.



*Figure 3.10 Entity Relationship Diagram*

## 3.6. Operating Environment

### 3.6.1. PHP

PHP (*Hypertext Preprocessor*) is a widely-used open source general-purpose scripting language which is especially suited for web development and can be easily embedded into HTML code. Therefore, PHP was used as the scripting language for the development of the product. It is recommended to use PHP 7.0 or higher version for the application.

### 3.6.2. Apache

A PHP script is run on Apache HTTP web server. Apache HTTP Server is a freely-available open-source web server software package. Apache provides a secure, efficient and extensible server that allows HTTP services in sync with the current HTTP standards. Apache version 2.4.x required to be used with the system.

### 3.6.3. The backend of the application

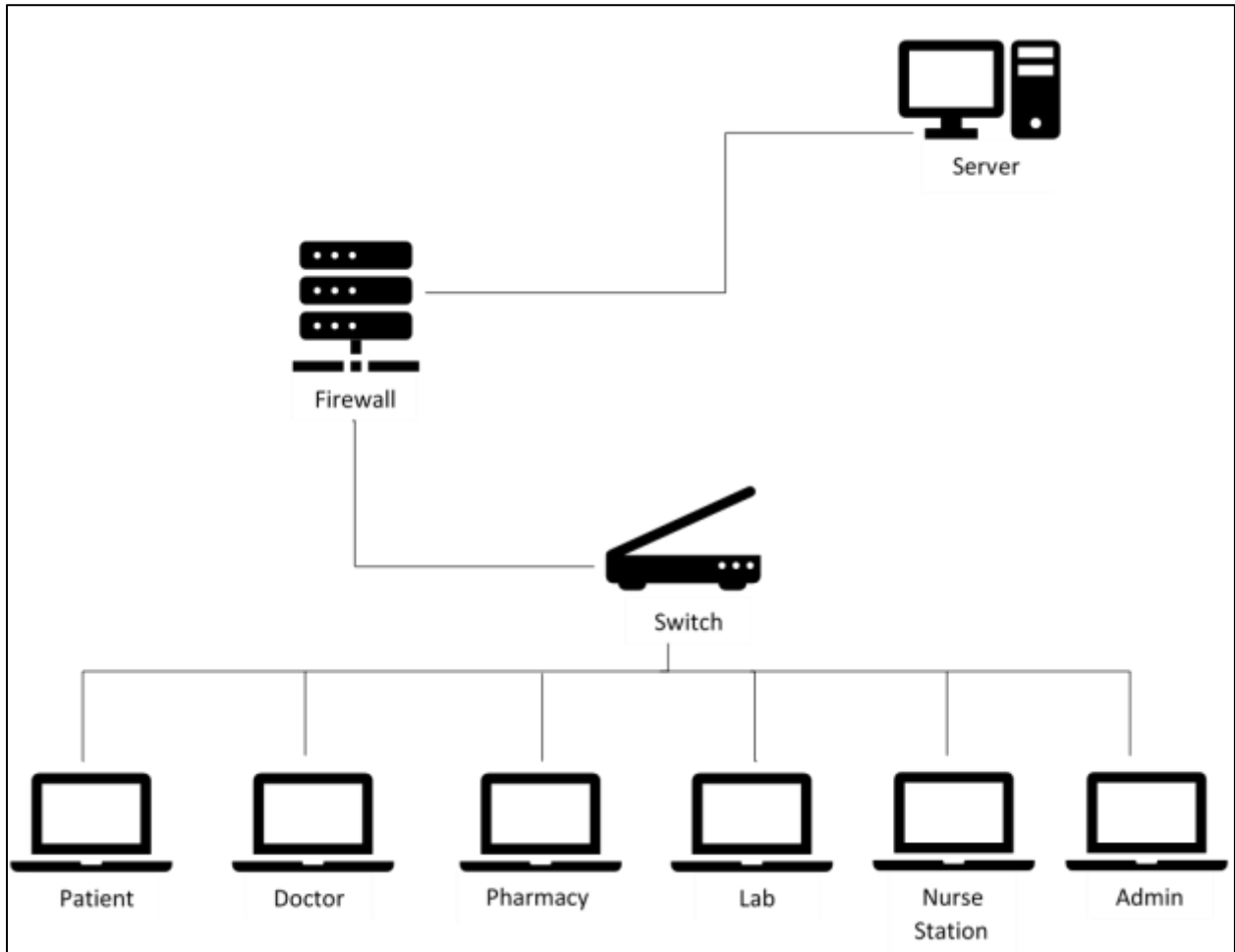
Database of the system uses MySQL. MySQL is an oracle-backed open-source relational database management system (RDBMS) based on Structured Query Language (SQL). It is most often associated with web applications and online publishing. The system was tested with MySQL version 8.0

The system currently supports Microsoft Windows environment only.

### 3.6.4. Disaster Recovery and System Backup

It is recommended to place a disaster recovery plan for the system. A daily system backup is required to be taken and stored in a separate location. The disaster recovery server is required to be located in a separate location and it is required to be accessible in the event of failure of the regular server.

Figure 3.11 operating environment diagram displays a basic operating environment of the system. A server should be connected to the firewall which provides network security. Users of the system such as a doctor, pharmacist, lab assistant, etc would be connected to the system via a router.



*Figure 3.11 Operating Environment*

### 3.7. Summary

This chapter explained the waterfall software development life cycle used as the guidance to develop the project and how its stages were used during the project. Use case diagram and time sequence diagram were explained as those helped in designing the system. The database diagram which consists of database tables and their relationships was also discussed in the chapter. Thereafter the entity-relationship diagram of the system was illustrated. Finally, the operating environment where the hardware and networking setup of the system was explained.

## 4. Evaluation

Evaluation chapter discusses the criteria used to measure the success of the system. The methods used to obtain the user feedback and the results of the responses are explained. Suggestions for the improvement of the system and lessons learned throughout the project are also contained in this chapter.

User evaluation was carried out mainly to identify how satisfied the users are with the system. The decision to continued with the project or not would depend on critical evaluation. Heuristic evaluation is an informal method of identifying the problems associated with the user interface.

Objectives of the evaluation include assessing the relevance, effectiveness, efficiency, impacts, and sustainability of the project and its activities. Objectives also include what challenges were faced during the implementation of the project, important lessons learned and recommendations for the future project implementation.

### 4.1. Evaluation Approach

The system was demonstrated to a selected group of twenty individuals comprised of patients, health sector personnel and a hospital administrator. Thereafter they were asked to use the system and provide the feedback on the given questionnaire. (Refer: Appendix A).

The questionnaire was developed based on the following principles.

- Problems and Needs (Relevance)
- Achievement of Purpose (Effectiveness)
- Sound management and value for money (Efficiency)
- Achievement of wider effects (Impact)
- Likely continuation of achieved results (Sustainability)

## 4.2. Evaluation Results

Summary of feedback of the evaluators was displayed in Table 4.1 evaluation results.

No	Object	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
1	Look & Feel	9	7	2	2	0
2	Ease of use	8	6	1	3	2
3	Reliability	8	7	3	1	1
4	Security	5	9	3	2	1
5	Value for money	4	7	5	3	1
6	Recommend to friend	7	5	8	0	0

*Table 4.1 Evaluation Results*

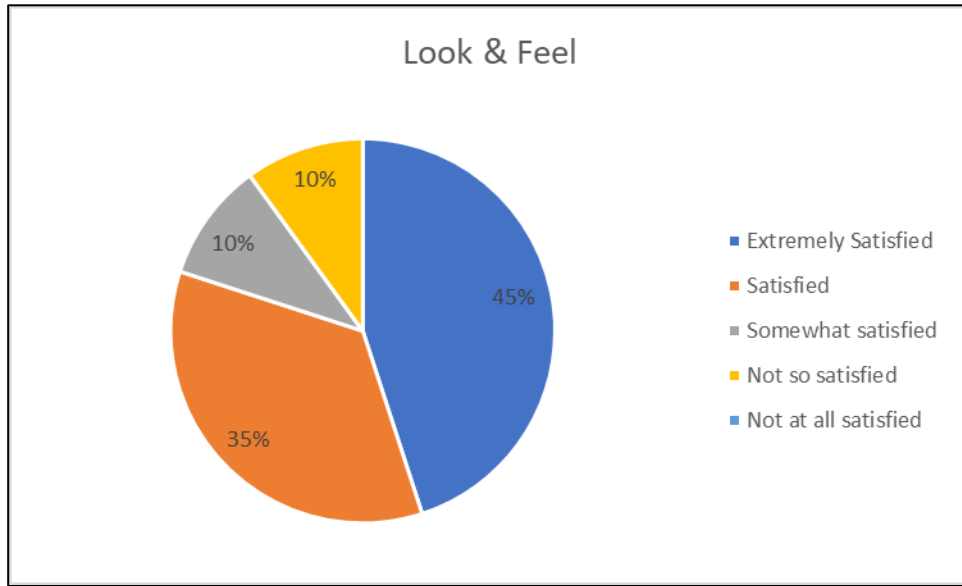
## 4.3. Analysis of Results

Feedback received from the user evaluation questionnaire were counted and summarized visually using pie charts.

A pie chart displays data, information, and statistics in an easy-to-read 'pie-slice' format with varying slice sizes illustrates how much of one data element exists. The bigger the slice, the more of that particular data was gathered. The evaluation helped to identify whether the developed system had achieved the project objectives or not.

### 4.3.1. Look and Feel of the system

The pleasantness of the system is assessed with the following pie chart in Figure 4.1 look and feel of the system.



*Figure 4.1 Look and feel of the system*

### Results

	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
Look and Feel	9	7	2	2	

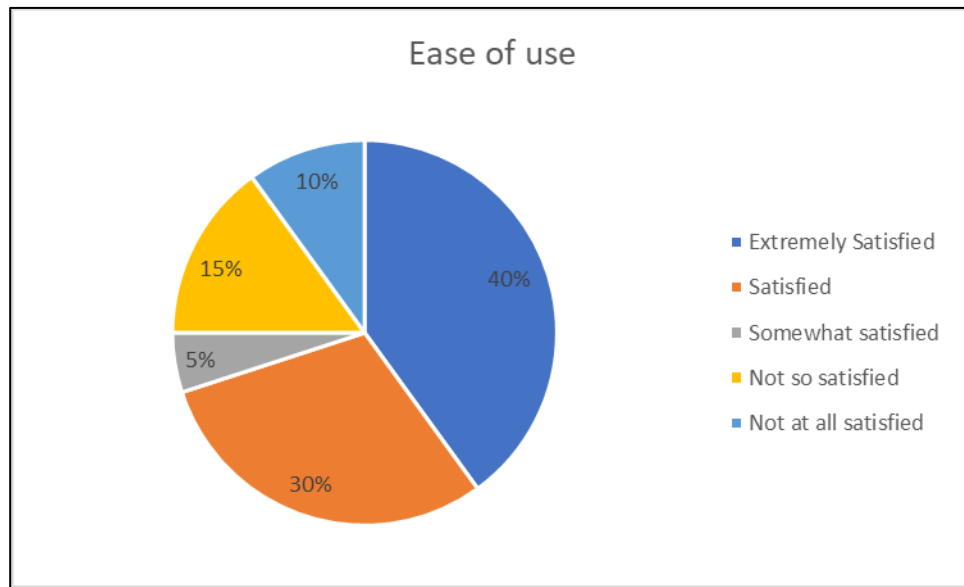
*Table 4.2 Look and feel of the system*

According to the Figure 4.1 look and feel of the system, 90% of users were satisfied with the user interface of the system while 10% of users, that is just two users as per the Table 4.2 look and feel of the system were not so satisfied with the look and feel of the system. Therefore, the majority of the users were satisfied with the system hence we could conclude that the design of the system was satisfactory.



### 4.3.2. Ease of use of the System

User-friendliness of the system was measured from this question. Logical order, navigation, and simplicity of understanding the system were the quantified. Results were illustrated in Figure 4.2 ease of use of the system and Table 4.3 ease of use of the system.



*Figure 4.2 Ease of use of the system*

### Results

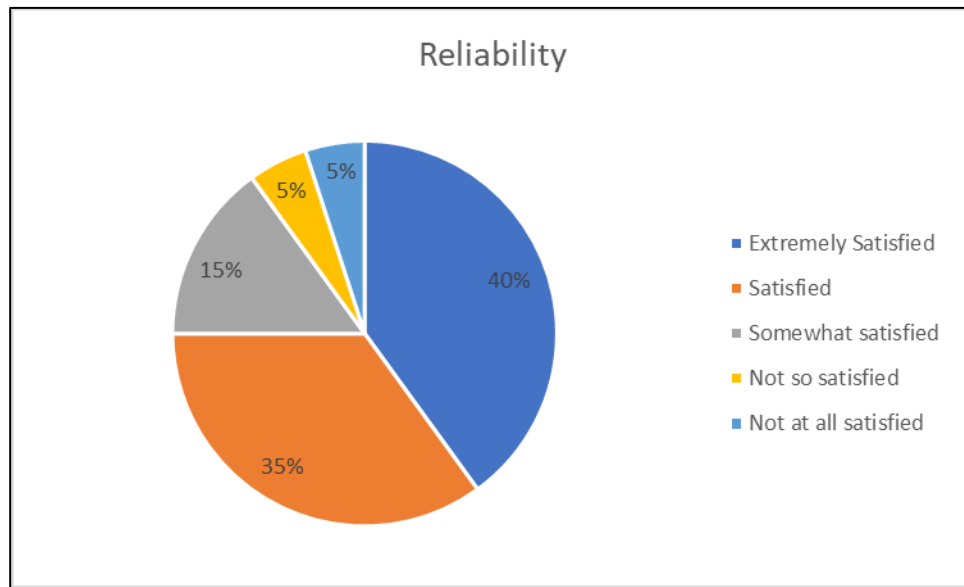
	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
Ease of use	8	6	1	3	2

*Table 4.3 Ease of use of the system*

As per Figure 4.2 ease of use of the system, 25% of users were not satisfied with the user-friendliness of the system while the rest of the users found that the system was user-friendly. It was found that the main reason for the unsatisfactory was some users were using the system for the first time without proper training.

### 4.3.3. Reliability of the System

Assessing the trustworthiness or the consistent performance of the system was the objective of this question and the results are graphically represented in the below Figure 4.3 reliability of the system.



*Figure 4.3 Reliability of the system*

### Results

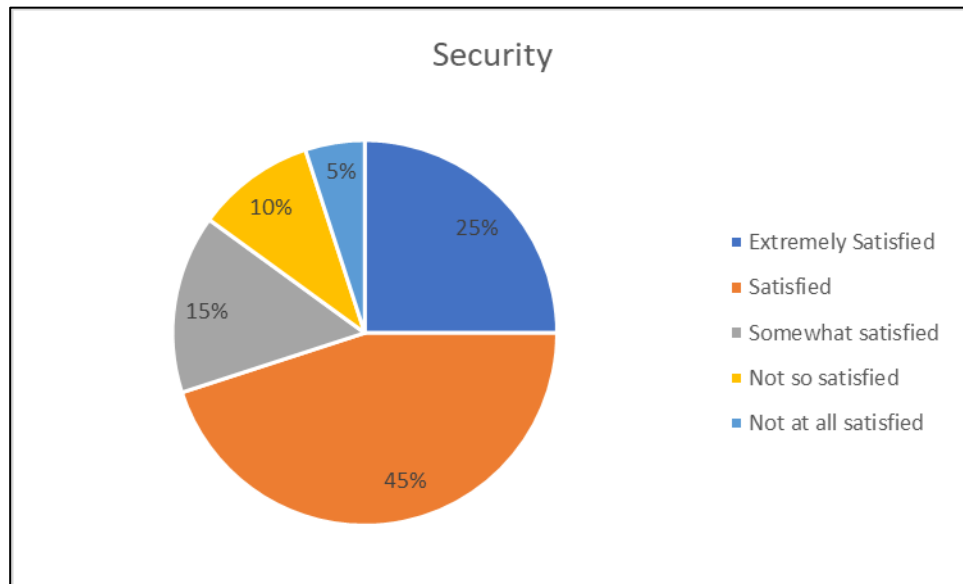
	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
Reliability	8	7	3	1	1

*Table 4.4 Reliability of the system*

Majority of users i.e. about 90%, felt confident about the system however Table 4.4 reliability of the system showed that two users were not satisfied with the reliability of the system but they were not mentioned reasons for their decision.

#### 4.3.4. Security of the System

Users were explained about the security of the system. An authorization matrix was maintained and access to the system modules will be based on the user role. For instance, a patient will not be able to access to doctor’s dashboard. Though the authorization was not implemented to the existing system, the user opinion was obtained for future references. User responses were summarized in the below Figure 4.4 security of the system and Table 4.5 security of the system.



*Figure 4.4 Security of the system*

#### Results

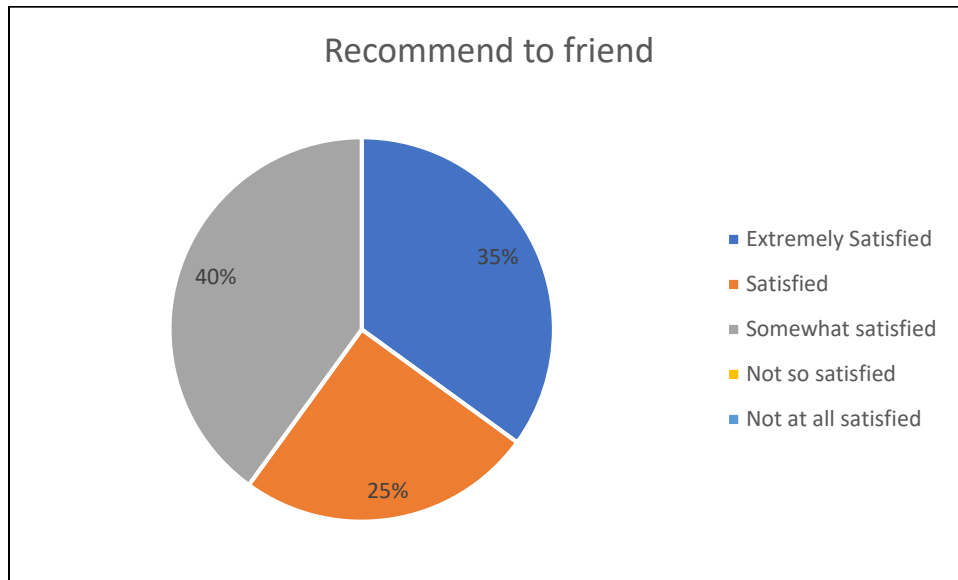
	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
Security	5	9	3	2	1

*Table 4.5 Security of the system*

According to the pie chart shown in Figure 4.4 security of the system, it was revealed that 85% of the users were satisfied with the proposed authorization and other security measures of the system. As of the Table 4.5 security of the system, three users were unsatisfied with the security measures. It is found that those users were satisfied with user-based authorization roles but wanted more system-wise security measures.

#### 4.3.5. Recommend to friend

The objective of this question was to get an idea of whether users will encourage their friends and family to enrol the system and results would be beneficial in the event of releasing the product to market. User feedback is illustrated in the pie chart given in the below Figure 4.5 recommend to a friend and actual results are shown in Table 4.6 recommend to a friend.



*Figure 4.5 Recommend to a friend*

#### Results

	Extremely Satisfied	Satisfied	Somewhat satisfied	Not so satisfied	Not at all satisfied
Recommend to friend	7	5	8	0	0

*Table 4.6 Recommend to a friend*

As shown in both Figure and Table 4.6, the results were very satisfactory. All users were responded favourably as they would recommend the system to their friends and family.

#### **4.3.6. Thoughts on how to improve the system**

Users were given a free space to write down their thoughts on how to improve the system. Following were their significant suggestions.

- Integrate an appointment system or interface with existing appointment systems eg: eChannelling, Doc990
- Facility to prepare medical bills
- Introduce ward management module
- Inventory management module for stores, pharmacy, laboratory, etc
- Tele-doctor facility. Ability to get the doctor option via online, video conferencing, over the telephone, etc
- Deliver drugs to the door-step facility
- Facility to manage complaints
- Insert chatbot
- More reports and statistics

Further discussions and analysis of suggestions would help to identify the essential features of the system. It is expected to gradually include those identified features to the system with future releases.

#### 4.4. Lessons Learnt and Failure to achieve objectives

This was the first-ever development of this scale performed by the developer. He had not any previous experience of developing a complete software product. Further, the knowledge of PHP was limited to basic commands and operations. Therefore, the biggest challenge was to develop the system while learning PHP language.

It was tough to conduct development of this scale while learning. Was stuck on some stages and had to seek the help of either online or friends. Yet, it was managed to get the system completed overcoming all obstacles.

Two deliverables related to the following objectives of the project were unable to achieve. Nevertheless, the rest of the deliverables were accomplished successfully.

- Increase electronic medical records coverage of hospitals
  - › Facilitate to keep patient medical history in a smart card/barcode

The system was designed to retain a patient's medical history and it was supposed to record medical history in a smart card or bar code. The hardware requirement (smart cards, bar code readers and bar code printers) prevent the implementation of the objective. However, the system is capable of recording patient history and it may be possible to deploy in a future release.

- Improve quality and efficiency in service delivery, governance, accountability and effective use of resources of a hospital.
  - › SMS Alerts

The system was supposed to send SMS alerts to remind patients about forthcoming appointments. , The requirement of SMS gateway which was supposed to purchase was prevented implementation of SMS alerts. Yet, email alerts were configured and deployed successfully. SMS alerts could be deployed with a future commercial release of the product.

## 4.5. Summary

The chapter started with the explanation of the base of project evaluation. Results of the user feedback were illustrated and analyzed with relevant tables and charts. Thereafter the chapter emphasizes the system improvements suggested by the users. Finally, the lessons learned by the developer throughout the project development and objectives which were unable to achieve were revealed in the evaluation chapter.

## 5. Conclusion

This chapter presents a summary of the results of the project and future improvements.

### 5.1. Summary of Results

A web-based hospital information system was developed based on the requirements of different stakeholders of a hospital. It comprised of six modules namely patient, doctor, laboratory, pharmacy, nurse and administrator. Each module has its own set of features, for instance, email alerts, reports, etc. and comprehensively integrated with each other. Evaluators of the system confirmed that the software was up to the expected standards. Foremost of the benefits of the system are as follows:

- Automation of the system improves the efficiency
- The user-friendly graphical user interface provided easy to use system
- Effectively overcome the delays in communication
- System security, data security, and reliability are the striking features

### 5.2. Future Enhancements

The system is open for future enhancements. The existing system did not facilitate financial transactions, therefore, the immediate improvement would modify the system to connect with payment gateway and equip with the financial module.

The system should also need to be integrated with different laboratory systems used in the hospital to get the laboratory reports directly without manual intervention. The existing system is capable of integration with external systems.

eChannelling & DOC990 are popular online appointment creation service providers in Sri Lanka. Therefore, the hospital information system should be integrated with such service providers which would be convenient to users.

The hospital management system would also provide an inventory management system for departments like pharmacy, laboratory, etc which would eliminate the requirement of maintaining separate software for the same.

These enhancements would provide the hospital management system a distinctive advantage over its competitors in the market once the system is commercially launched.



## References

1. Vladimir Zwass. Information System. [Online]. Available : <https://www.britannica.com/topic/information-system>
2. Morris F Collen M.D. Origins of Medical Informatics [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1307150/?page=1>
3. The History of Health Information Management – from then to now. [Online]. Available: <https://liaison.opentext.com/blog/2017/05/02/history-health-information-management-now/>
4. Joshua R Vest and Larry D Gamm. Health information exchange: persistent challenges and new strategies. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2995716/>
5. Digital Health Project. Information and Communication Technology Agency (ICTA) – Sri Lanka. [Online]. Available: <https://www.icta.lk/projects/digital-health/>
6. HospitalRun. Free, easy-to-use software for developing world healthcare. [Online]. Available: <http://hospitalrun.io/>
7. OpenMRS. A free, open-source electronic medical record platform. [Online]. Available: <https://openmrs.org/>
8. VERTICAL Systems. Medical Software for Clinics and Hospitals. [Online]. Available: <http://www.ehr-medical-software.com/index.htm>
9. eHospital System. [Online]. Available: <https://www.adroitinfosystems.com/products/ehospital-systems>
10. Caresoft. [Online]. Available: <https://caresoft.co.in/>
11. Khrystyna Oliinyk. 5 Differences Between Open Source and Closed Source Software. [Online]. Available: <https://api2cart.com/business/5-differences-between-open-source-and-closed-source-software/>
12. What is use case diagram. [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>
13. What is an entity relationship diagram. [Online]. Available: <https://www.lucidchart.com/pages/er-diagrams>

## Appendix A: Project Evaluation Form

# Project Evaluation Form

### Instructions for filling out this form:

This form is an aid for the identification of strengths and weaknesses in the performance of the Integrated Hospital Information System. Please be as objective when filling out this form. I will use this information to reinforce the software's strengths and to find ways to eliminate weaknesses.

1. How satisfied you are with the look and feel of this software (User Interface)?

- Extremely Satisfied
- Satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

2. How satisfied you are with the software's ease of use?

- Extremely Satisfied
- Satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

3. How satisfied you are with the reliability of this software?

- Extremely Satisfied
- Satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

4. How satisfied you are with the security of this software?

- Extremely Satisfied
- Satisfied
- Somewhat satisfied
- Not so satisfied
- Not at all satisfied

5. How likely is it that you would recommend this software to a friend or family member?

- Extremely likely
- Likely
- Somewhat likely
- Not so likely
- Not at all likely

6. Do you have any thoughts on how to improve this software?

## Appendix B: Functional Specification

# Integrated Hospital Information System

## Functional Specification

## 1. Purpose

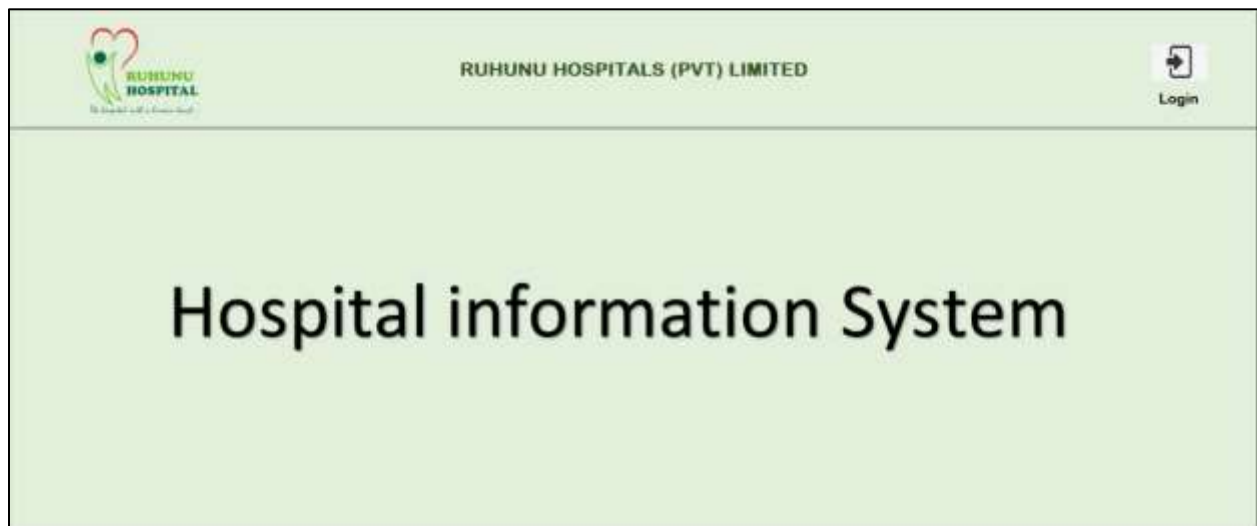
The purpose of this document is to present the functional specification of the Integrated Hospital Information System. Design of the screens, navigation through screens and functions of each screen are presented in this document.

## 2. System Features

Please note that screen design, content, and navigation are subjected to change during the development of the system

### 2.1. Main Page

The main page of the system should be as follows



*Figure 2.1 Home Page*

- Click on “Login” icon should direct the user to login page

## 2.2. Login Page

A registered user should be able to enter his/her credentials and login to the system

### New User

- A new patient-user should be able to register himself to the system
- Staff users should be created by the Administrator

The screenshot shows the login page for Ruhunu Hospitals (Pvt) Limited. The page has a light green background. At the top left is the logo for Ruhunu Hospital, which includes a heart and a caduceus. To the right of the logo, the text "RUHUNU HOSPITALS (PVT) LIMITED" is displayed. Below this, a dark green horizontal bar contains the text "Health Information System". The main content area is light green and features the word "Login" in a large, bold, dark green font. Below "Login" are two input fields: "User Name" and "Password". The "User Name" field is empty, and the "Password" field contains a series of asterisks. To the right of these fields is a green button with the text "SIGN IN". Below the "Password" field is a checkbox labeled "Remember me". Below the checkbox are two links: "Forget password ? [Click here](#)" and "[Sign up](#)".

*Figure 2.2 Login Page*

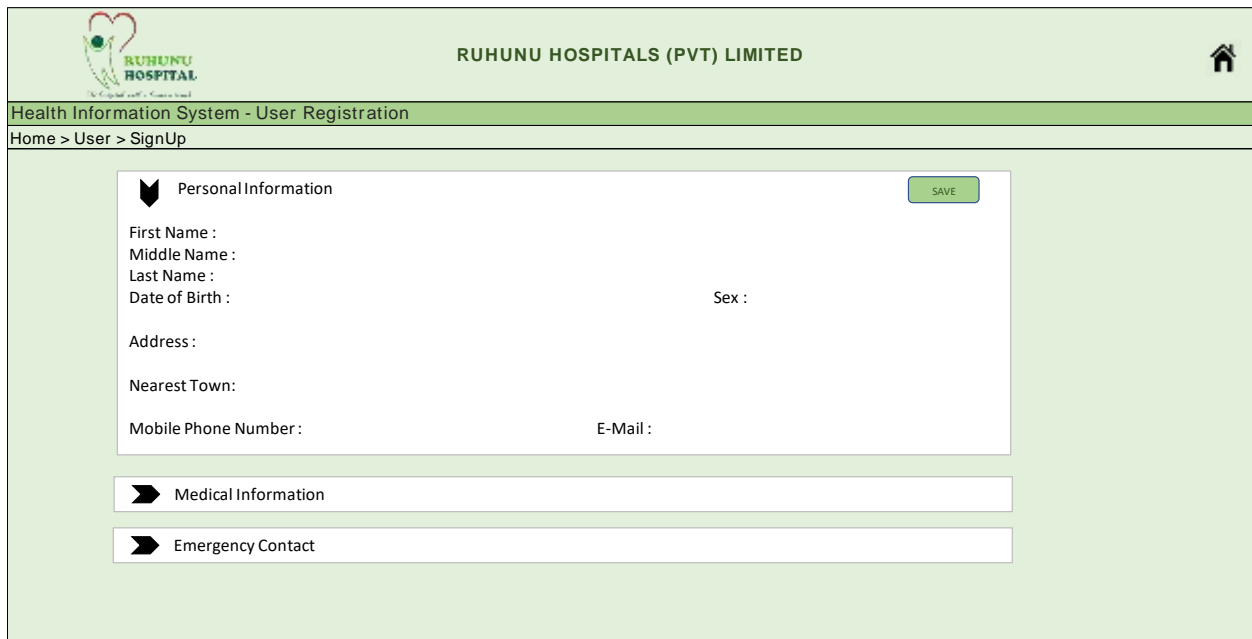
## 2.3. Sign-Up Page

- Click on Sign up icon in the login page would result below user registration screen



*Figure 2.3.1 Sign up Page*

- Print bar code button will print bar code in an external material
- Click on Personal Information will open the following screen
- Click on the SAVE button will save the data



*Figure 2.3.2 Personal Information*

- Click on Medical Information will open the following screen
- Click on the SAVE button will save the data

*Figure 2.3.3 Medical Information*

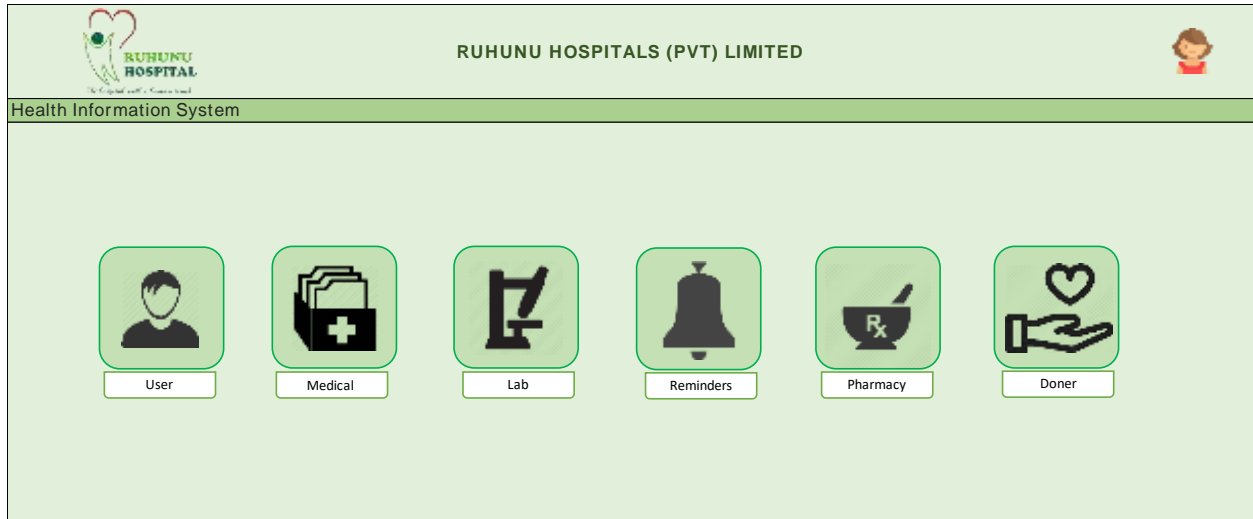
- Click on Emergency Contact will open the following screen
- Click on the SAVE button will save the data

*Figure 2.3.4 Emergency Contact*



## 2.4.Home Page


Once the user successfully logged in to the system following home page will appear




*Figure 2.4 Home Page*

Following explanations are on the assumption of a patient logged in to the system

## 2.5. User Page

- Click on  icon in the Home page would results screens drew under 2.3 Sign-up page
- User Should be able to change/update data.

## 2.6. Medical Page

- Click on  icon in the Home page would result below screen
- This will display the medical prescription details of the user

RUHUNU HOSPITALS (PVT) LIMITED


Health Information System - Medical Records

Date  To

Doctor

Execute

*Figure 2.6.1 Selection Screen*

- User will select a date range and doctor name and press execute
- Date range - mandatory / Doctor name – Optional
- The following results screen will appear
- It will be a Display only screen
- Click on  icon will pop-up any attached documents


RUHUNU HOSPITALS (PVT) LIMITED

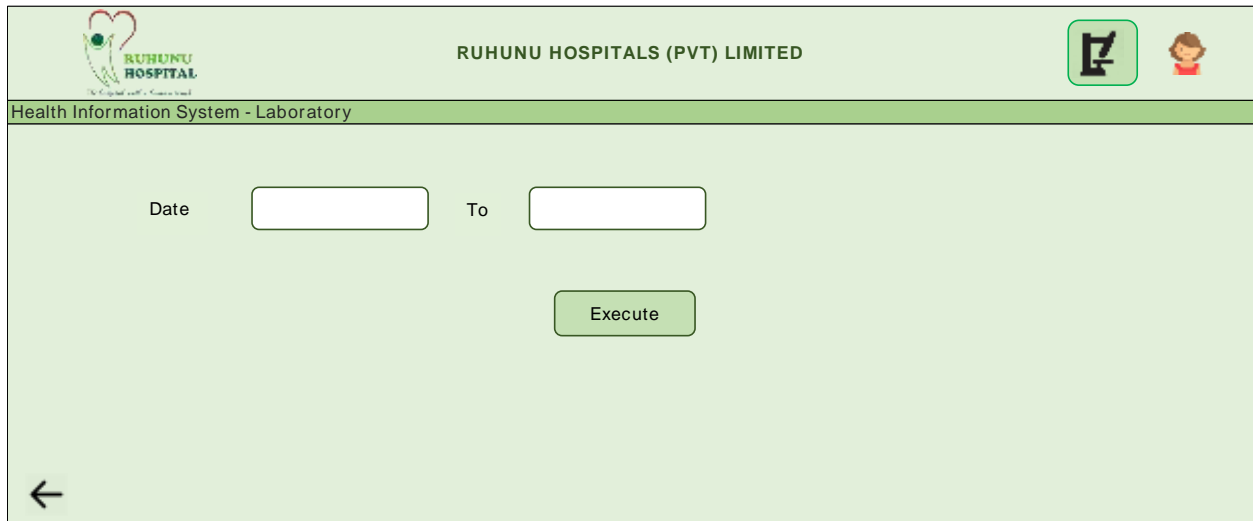
Health Information System - Medical Records

Date	Doctor	Description
21.10.2017	Dr. G D E S Jayawardena	The symptoms themselves may be difficult to identify. An occasional headache may be just a headache, or it could be a symptom of a larger problem.

*Figure 2.6.2 Results Screen*


## 2.7.Lab Page

- Click on  icon in the Home page will result below screen
- This will display the laboratory test report details of the user



The screenshot shows a web interface for 'RUHUNU HOSPITALS (PVT) LIMITED'. The page title is 'Health Information System - Laboratory'. It features two text input fields for 'Date' and 'To', and a green 'Execute' button. A back arrow is located in the bottom left corner.


*Figure 2.7.1 Selection Screen*

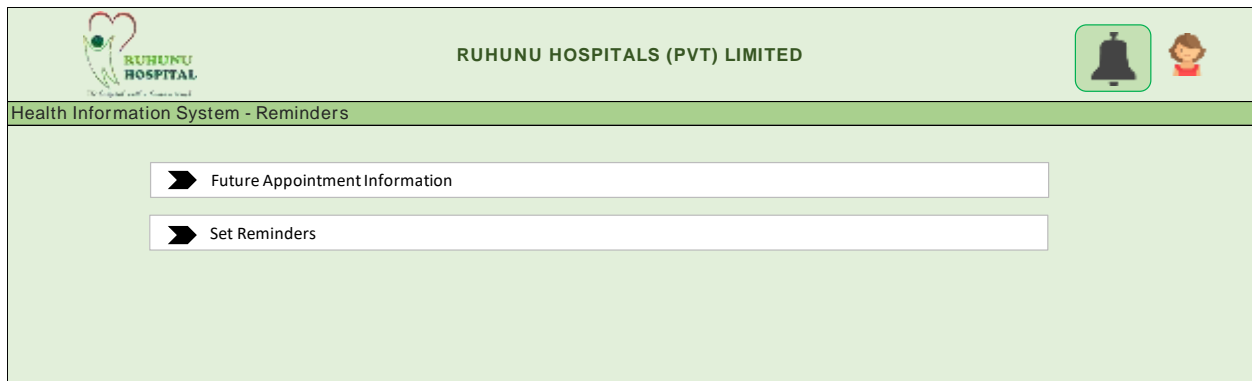
- User will select a date range and press execute
- The following results screen will appear
- It will be a Display only screen, but the user will be able to download if required
- Click on  icon will pop-up any attached documents
- If the user wishes to get the reports done from outside, then the user should be able to attach such reports in the system.
- There could be following types
  - Reports
  - X-Rays
  - Scan



*Figure 2.7.2 Results Screen*

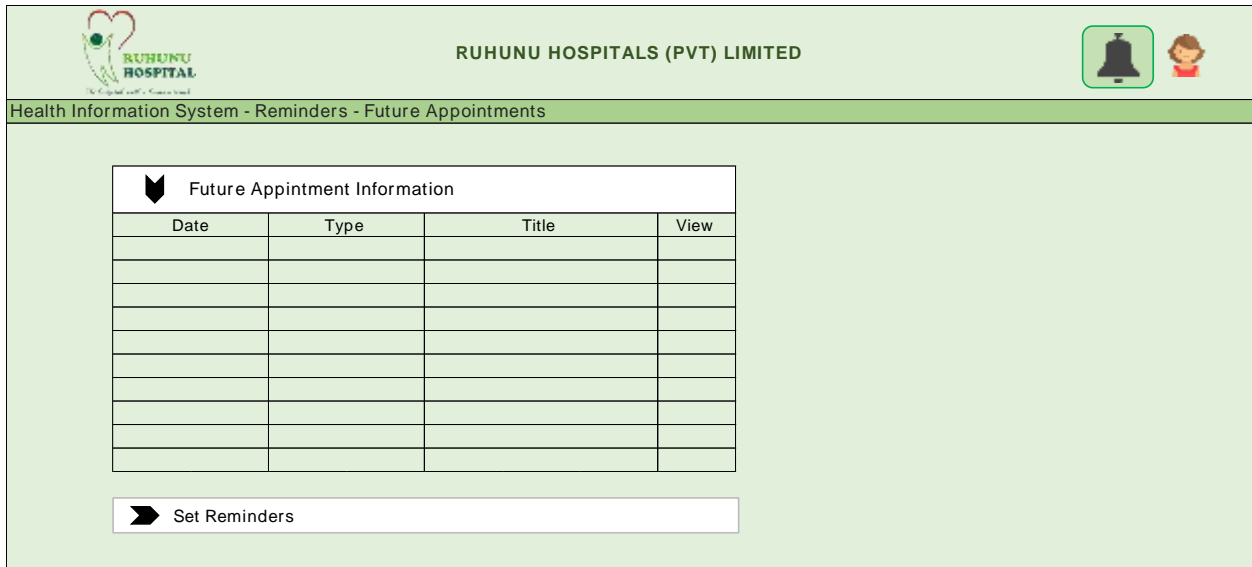
## 2.8.Reminders

- Click on  icon in the Home page will result below screen
- This screen will display the forthcoming appointments and users are also able to set reminders



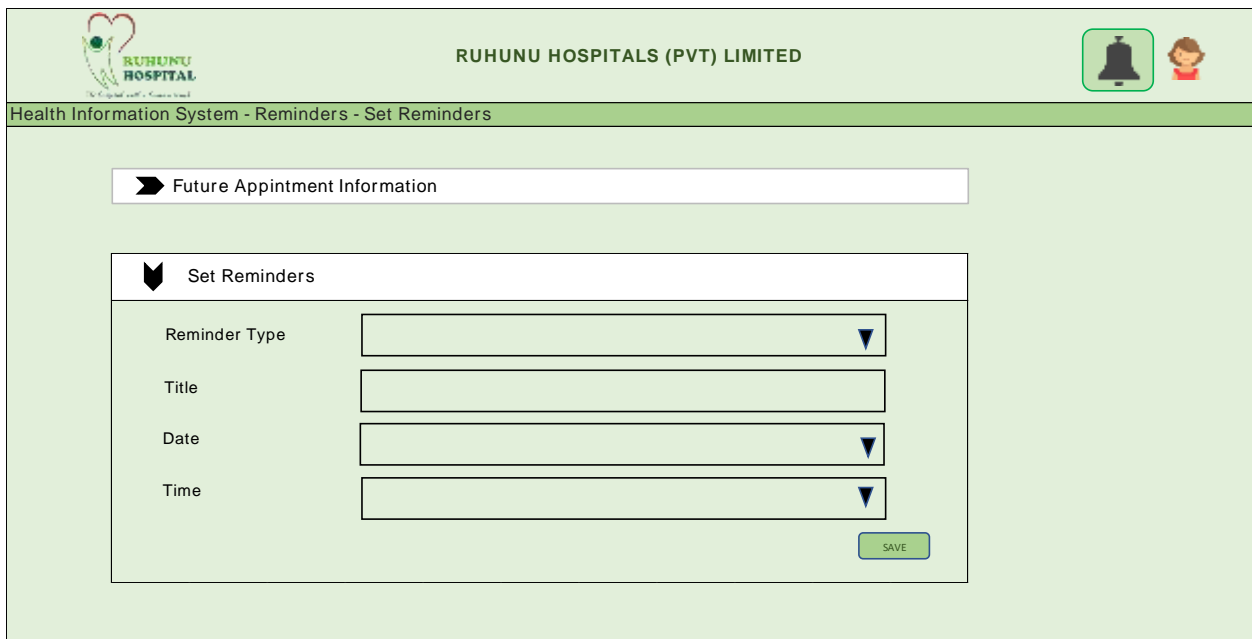
*Figure 2.8.1 Reminders Screen*

- Initially following appointment types will be conFigured
  - Doctor Appointments
  - Vaccination
  - Medical Test appointments




*Figure 2.8.2 Future appointment information*

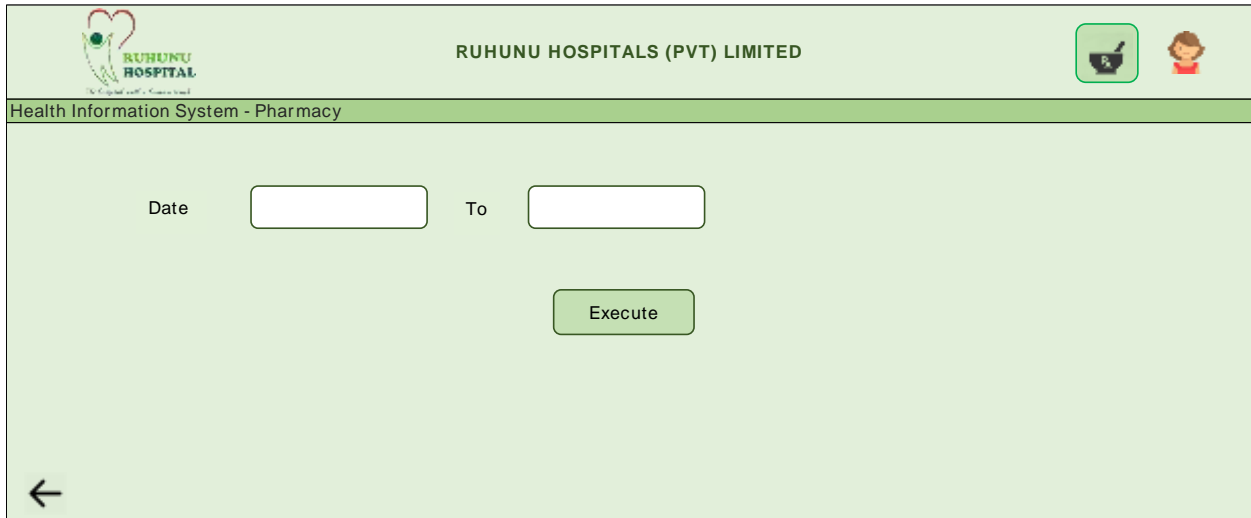
- User will be able to set reminders themselves
- Click on the “Save” button would save data
- Upon the due date, an SMS will be delivered to the given mobile number



*Figure 2.8.3 Set Reminders*

## 2.9. Pharmacy

- Click on  icon in the Home page will results pharmacy screen



RUHUNU HOSPITALS (PVT) LIMITED

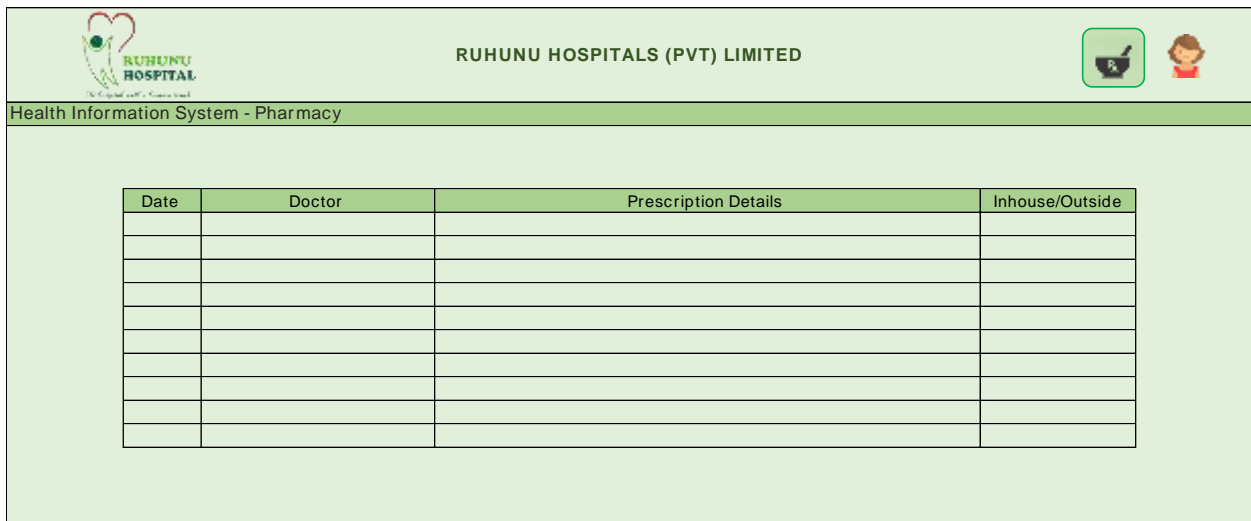
Health Information System - Pharmacy

Date  To

Execute

*Figure 2.9.1 Selection Screen*

- User will enter a date range and press the “execute” button
- Results screen would be as follows
- It will display whether the drugs were issued from Hospital pharmacy or not




RUHUNU HOSPITALS (PVT) LIMITED

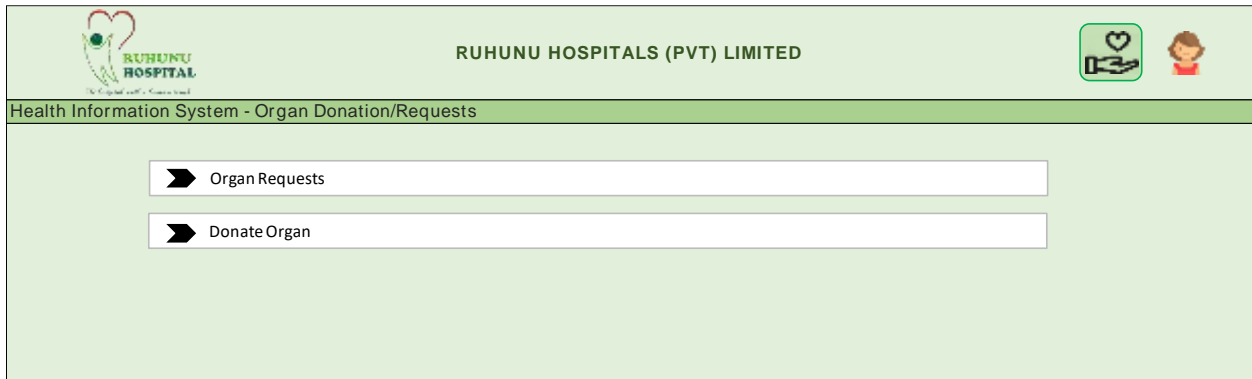
Health Information System - Pharmacy

Date	Doctor	Prescription Details	Inhouse/Outside

*Figure 2.9.2 Results Screen*

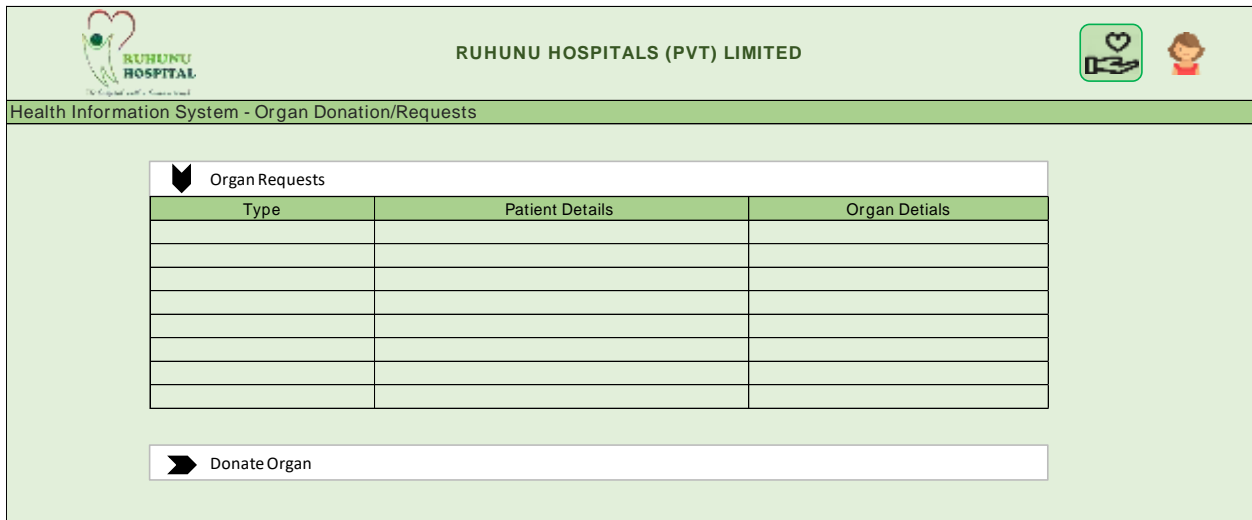
## 2.10. Organ Donations /Requests

- Click on  button will result the organ donations/requests screen



*Figure 2.10.1 Organs donation/requests Screen*

- Organ requests will list down the existing organ requirements



*Figure 2.10.2 Organs requests Screen*

- User will be able to mention the organ donation details using the “Donate organ” option

The screenshot displays the 'Donate Organ' interface. At the top, the header includes the hospital logo and name 'RUHUNU HOSPITALS (PVT) LIMITED' and the system title 'Health Information System - Organ Donation/Requests'. On the right, there are icons for a heart and a person. The main area features a sidebar with 'Organ Requests' and 'Donate Organ'. The 'Donate Organ' section contains three input fields: 'Organ Type' (dropdown), 'Blood Group' (dropdown), and 'Contact Details' (text input). A 'SAVE' button is located at the bottom right of the form.

*Figure 2.10.3 Donate organ Screen*

## 2.11. Doctor

When a doctor logged into the system his home page will be as of below



*Figure 2.11 Doctor Screen*

- The doctor will be able to view his personal details
- Appointments for the day
- Organ donor requests and donations



## 2.12. Laboratory

- Laboratory user will have following home page
- User will be able to search for laboratory requests
- New/Open, Completed, postponed statuses will be used to segregate requests

RUHUNU HOSPITALS (PVT) LIMITED

Health Information System - Laboratory

Open  Completed  Postpone

Date  To

Execute

*Figure 2.12.1 Selection Screen*

- Following results screen will appear according to the selection parameters
- User will be able to post results when it is ready

RUHUNU HOSPITALS (PVT) LIMITED

Health Information System - Laboratory

Date	Status	Type	Report Description	Patient ID

*Figure 2.12.2 Results Screen*

## 2.13. Pharmacy

- Pharmacy user will have following home page
- User will be able to search for drug requests
- New/Open, Completed, postponed statuses will be used to segregate drug requests

RUHUNU HOSPITALS (PVT) LIMITED

Health Information System - Pharmacy

Open  Completed  Postpone

Date  To

Execute

*Figure 2.13.1 Selection Screen*

- Following results screen will appear according to the selection parameters
- User will be able to post results when it is ready

RUHUNU HOSPITALS (PVT) LIMITED

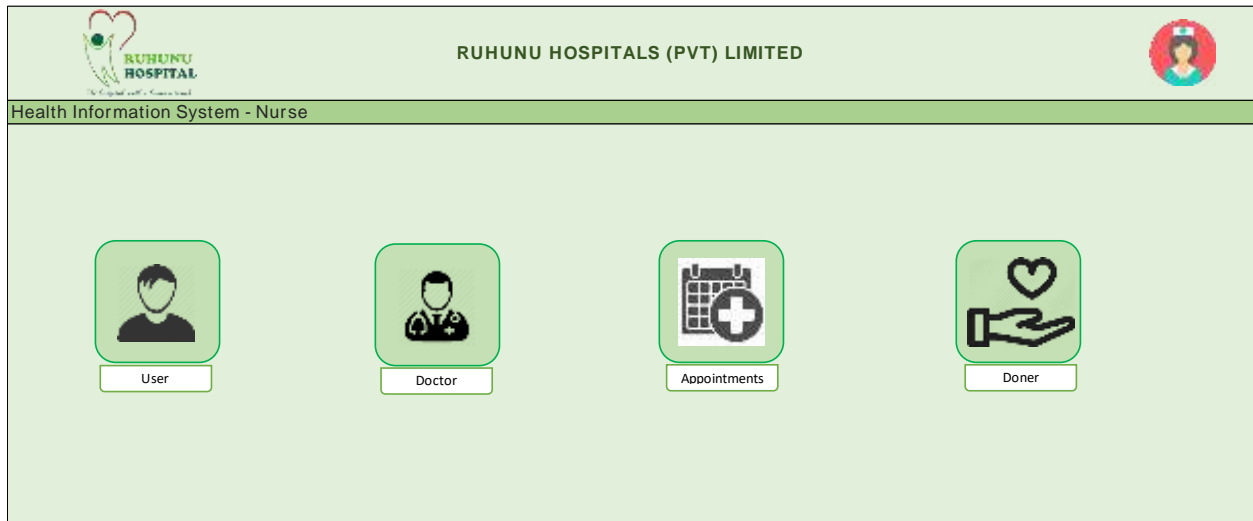
Health Information System - Pharmacy

Date	Status	Doctor Name	Prescription	Patient ID

*Figure 2.13.2 Results Screen*

## 2.14. Nurse

- Following screen will be the home screen for Nurse users



*Figure 2.14 Nurse Screen*

- The nurse will be able to view patient, doctor and organ donor/request details
- Set appointments for doctors
- Register new patients

## Appendix C: Technical Specification

# Integrated Hospital Information System

## Technical Specification

## 1. Purpose

The purpose of this document is to present the technical specification of the Integrated Hospital Information System.

## 2. Operation Environment

### 2.1. Software Requirement

- ASP.Net 4.0 or higher
- EXT.Net 1.2 or higher
- SAP .NET Connector 3.0 or higher
- Office Web Components 2003 or higher
- Fusion Charts 3.2.4 or higher
- IIS 7.0 or higher
- Email Server
- Windows Server R2 (64 Bit)

### 2.2. Hardware Requirement

The hardware requirement is to be analyzed and implemented by the client.

### 2.3. Web Browser Requirement

Hospital Information system requires following web browsers for it to run:

- Google Chrome V50+
- Firefox 60+

### 3. General Development Standards

- Depending on the data type all the input fields should be validated with appropriate messages
- Depending on the character length all input fields should be limited to hold only the maximum characters allowed in the respective data field. Ex. A text box to input First Name. The respective data fields type is 'varchar(100)'. The text field that takes the First Name as the input should not allow the user to type more than 100 characters in it.
- Invalid character validation. Ex. Want to validate special characters such as a <html>, <script>
- Accessibility: Users who are given access to the Hospital Information System

### 4. Code Extractions

Important code extractions are listed below

#### 4.1. Manage Users

- Model – User
- Controller – AdminController

```
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show()
{
    $users = new User();
    $filterParams = [];

    if(request()->has('filter_enabled') && request('filter_enabled') === 'true'){
        if(request()->has('user_name') && request('user_name') !== null){
            $users = $users->where('name', 'like', '%'.request('user_name').'%');
            $filterParams['user_name'] = request('user_name');
        }
        if(request()->has('email') && request('email') !== null){
            $users = $users->where('email', request('email'));
            $filterParams['email'] = request('email');
        }
        if(request()->has('user_type') && request('user_type') !== null){
            $users = $users->where('user_type', request('user_type'));
        }
    }
}
```

```

        $filterParams['user_type'] = request('user_type');
    }
}

return view('admin.pages.admin_dashboard',[
    'users'=>$users->get(),
    'filterParams' => $filterParams
]);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store()
{
    $data = request()->validate([
        'name' => 'required|string',
        'user_type'=> 'required',
        'email' => 'required|email',
        'password' => 'required|string|min:6',
        'speciality' => ""
    ]);
    $data['password'] = Hash::make($data['password']);
    $data['email_verified_at'] = Carbon::now();
    User::create($data);
    return redirect()->back()->with('message','User created successfully');
}

/**
 * Delete a user from storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function delete($id)
{
    User::find($id)->delete();
    return redirect()->back()->with('message','User removed successfully');
}
}

```

## 4.2.Add Diagnosis

- Model – Diagnose
- Controller – DiagnoseController

```
/**
 * Store Diagnose and Prescription
 * @param Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function store(Request $request){
    $validatedData = $request->validate([
        'diagnose' => 'required',
        'prescription' => 'string',
    ]);
    $diagnoseValues = [];
    DB::beginTransaction();
    try {
        $result=Diagnose::create([
            'diagnose'=>$validatedData['diagnose'],
            'patient_id'=>request('patient_id'),
            'posted_date'=>Carbon::now(),
            'doctor_id'=>Auth::id()
        ]);
        Prescription::create([
            'prescription'=>$validatedData['prescription'],
            'diagnose_id'=>$result['id']
        ]);
        if (request('pharmacy_type') === '0' ){
            Pharmacy::create([
                'diagnose_id'=>$result['id'],
                'is_ready'=>false,
            ]);
        }
        if(request()->has('cholesterol') && request('cholesterol') !== null){
            $diagnoseValues['cholesterol'] = request('cholesterol');
        }
        if(request()->has('bp') && request('bp') !== null){
            $diagnoseValues['bp'] = request('bp');
        }
        if(request()->has('hba1c') && request('hba1c') !== null){
            $diagnoseValues['hba1c'] = request('hba1c');
        }
        if(count($diagnoseValues) > 0){
            $diagnoseValues['diagnose_id'] = $result['id'];
            $diagnoseValues['created_at'] = Carbon::now();
            $diagnoseValues['updated_at'] = Carbon::now();
            DiagnoseValue::create($diagnoseValues);
        }
    }
    DB::commit();
    return redirect()->back()->with('diagnose_id',$result['id'])->withSuccess('Diagnose Added Successfully !')
        ->with('active_tab', 'add_diagnose');
```



```

}catch (Exception $exception){
    DB::rollback();
    return redirect()->back()->withErrors('Something went wrong !');
}

```

### 4.3. Make Donations

- Model – Donations
- Controller – DonationController

```

/**
 * Show Donation page
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show(){
    return view('patient.pages.organ',[
        'donations'=>Donation::where('patient_id',Patient::getPatient(Auth::id()))['id']->get()
    ]);
}

/**
 * Store Donations
 * @param Request $request
 * @return mixed
 */
public function store(Request $request){
    $validatedData = $request->validate([
        'organ_name' => 'required|string',
        'description' => 'required|string',
        'type'=>'required'
    ]);
    $validatedData['posted_date']=Carbon::now();
    $validatedData['patient_id']=Patient::getPatient(Auth::id())['id'];

    Donation::create($validatedData);
    return Redirect::back()->withSuccess('Donation request submitted successfully !');
}

/**
 * Show Donation page
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function showDonations(){
    return view('patient.pages.organ',[
        'donations'=>Donation::where('patient_id',Patient::getPatient(Auth::id()))['id']->where('type','Donate')->get()
    ]);
}

/**
 * Show Donation page
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View

```

```

*/
public function showRequests(){
    return view('patient.pages.organ',[
        'donations'=>Donation::where('patient_id',Patient::getPatient(Auth::id()))['id']->where('type','Request')->get()
    ]);
}

```

## 4.4. Add Appointments

- Model – Appointment
- Controller – AppointmentController

```

* Display the specified resource.
*
* @param int $id
* @return \Illuminate\Http\Response
*/
public function show()
{
// $unreg_appointments = UnregisteredAppointment::where('doctor',Auth::user()->name)
// ->where('date',Carbon::today()->get());
return view('doctor.pages.all_appointments',[
    'appointments' => Appointment::getAppointmentByDate(Carbon::now()),
    'date'=>Carbon::now()->toDateString()
// 'unreg_appointments'=>$unreg_appointments
]);
}

/**
* Store a newly created resource in storage.
*
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store()
{
    $data = request()->validate([
        'title' => 'required|string',
        'time' => 'required',
        'date' => 'required|date',
        'doctor' => 'required|string',
        'patient_id' => 'required'
    ]);
    $data['isAppointed'] = '1';
    $response = Appointment::updateOrCreate(['date'=>$data['date'],
        'patient_id'=>Patient::where('id',$data['patient_id']->value('id')), $data);
    return $response->wasRecentlyCreated == true ? redirect()->back()->with('message','Appointed Successfully') :
        redirect()->back()->with('message','Appointment Made Before');
}

```

```

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function storeUnregistered()
{
    $data = request()->validate([
        'name' => 'required|string',
        'title' => 'required|string',
        'time' => 'required',
        'date' => 'required|date',
        'doctor' => 'required|string',
    ]);
    $data['isAppointed'] = '1';
    $response = UnregisteredAppointment::updateOrCreate(['date'=>$data['date'],'name'=>$data['name']], $data);
    return $response->wasRecentlyCreated == true ? redirect()->back()->with('message', 'Appointed Successfully') :
        redirect()->back()->with('message', 'Appointment Made Before');
}
/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}

/**
 * Filter Appintment by Date
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function filter(){
    $date=request('date_picker');
    return view('doctor.pages.all_appointments',[
        'appointments' => Appointment::getAppointmentByDate($date),
        'date'=>$date
    ]);
}

```

## 4.5. Add Emergency Details

- Model – EmergencyDetail
- Controller – EmergencyDetailController

```
/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $data=EmergencyDetail::where('patient_id',Auth::id()->get());
    return view('patient.pages.my_profile',['details'=>$data]);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store()
{
    $data = request()->validate([
        'contact_name' => 'required|string',
        'tel' => 'required|numeric',
    ]);
    $data['patient_id']=Auth::id();
    EmergencyDetail::updateOrCreate(['patient_id'=>Auth::id()],$data);
    return redirect()->route('show_patient_myprofile', ['active_tab' => 'emergency_contact']);
}
```

## 4.6.Add Event

- Model – Event
- Controller – EventController

```
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
```

```

public function store(Request $request)
{
    $data=request()->validate([
        'name' => 'required|string',
        'type' => 'required|string',
        'date' => 'required|date',
        'description' => 'required',
        'time' => 'required',
        'venue' => 'required',
        'patient_id' => 'required'
    ]);
    $data['doctor_name'] = Auth::user()->name;
    Event::create($data);
    $user = User::find((int)request('patient_id'));
    $user->notify(new NewEventAdded());
    return redirect()->back()->with('active_tab', 'add_events')->withSuccess('Event Added Successfully !');
}

```

## 4.7. Request Lab Reports

- Model – Laboratory
- Controller – LabReportController

```

/**
 * Request lab reports
 * @param Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function store(Request $request){

    if(request('diagnose_id') === null){
        return redirect()->back()->with('active_tab', 'request_lab_report')
            ->withErrors('You have to add Diagnose record first !');
    }

    $validatedData = $request->validate([
        'report-title' => 'required',
        'report-description' => 'string|required',
    ]);
    Laboratory::create([
        'title'=>$validatedData['report-title'],
        'description'=>$validatedData['report-description'],
        'is_ready'=>(int)0,
        'patient_id'=>$request->patient_id,
        'diagnose_id'=>request('diagnose_id'),
    ]);
    return redirect()->back()->with('diagnose_id',request('diagnose_id'))
        ->with('message',$validatedData['report-title'].' report requested successfully !')
        ->with('active_tab', 'add_events');
}

```

## 4.8. Add and Update Patients

- Model – Patient
- Controller – PatientDetailController

```
/**
 * Show the patient detail view
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show(){
    return view('patient.pages.patient_details',[
        'patient_categories'=>config('constances.patient_categories'),
    ]);
}
/**
 * Store Patient details
 * @param Request $request
 * @return view
 */
public function store(Request $request){
    $validatedData = $request->validate([
        'full_name' => 'required|max:255',
        'address' => 'required|string',
        'tel_no' => 'required|numeric',
        'gender' => 'required|string',
        'blood_group'=>'required|string',
        'dob' => 'required',
        'height' => 'required|string',
        'weight' => 'required|string',
        'patient_category' => 'required|string',
        'allergies' => 'required|string',
        'medical_condition' => 'required|string'
    ]);
    $validatedData['user_id']=Auth::id();
    $validatedData['patient_category']=config('constances.patient_categories')[\request('patient_category')];

    $patient_detail=Patient::create($validatedData);
    return Redirect::route('show_patient_myprofile');
}
```

## 4.9. Download Lab Reports

- Model – Laboratory
- Controller – CompletedRequestsController

```
/**
 * show completed request view
 * @param $report_id
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function show($report_id){

    return view('lab.pages.completed_request',[
        'diagnose'=>Diagnose::getSingleReport($report_id),
        'report'=>Laboratory::where('id',$report_id)->first()
    ]);
}
/**
 * download report
 * @param $report_id
 * @return \Symfony\Component\HttpFoundation\BinaryFileResponse
 */
public function download($report_id){
    $report=Laboratory::find($report_id);
    $pathToFile = storage_path('app/public/upload/' . $report->report);
    return response()->download($pathToFile);
}
```

## 4.10. Upload Lab Reports

- Model – Laboratory
- Controller – SingleRequestsController

```
/**
 * Upload report
 * @param Request $request
 * @return mixed
 */
public function update(Request $request){

    $validatedData = $request->validate([
        'report' => 'required',
    ]);
    $file=$validatedData['report']->getClientOriginalName();
    $validatedData['report']->storeAs('public/upload',$file);
}
```

```

$validatedData['report']=$file;
$validatedData['is_ready']=(int)1;

$report=Laboratory::find(request('report_id'));
$report->update($validatedData);
return Redirect::back()->withSuccess('Report Uploaded Successfully !');

}

```

## 4.11. Upload Prescriptions

- Model – Pharmacy
- Controller – DiagnoseController

```

/**
 * Store Diagnose and Prescription
 * @param Request $request
 * @return \Illuminate\Http\RedirectResponse
 */
public function store(Request $request){
    $validatedData = $request->validate([
        'diagnose' => 'required',
        'prescription' => 'string',
    ]);
    DB::beginTransaction();
    try {
        $result=Diagnose::create([
            'diagnose'=>$validatedData['diagnose'],
            'patient_id'=>request('patient_id'),
            'posted_date'=>Carbon::now(),
            'doctor_id'=>Auth::id()
        ]);
        Prescription::create([
            'prescription'=>$validatedData['prescription'],
            'diagnose_id'=>$result['id']
        ]);
        if (request('pharmacy_type') === '0' ){
            Pharmacy::create([
                'diagnose_id'=>$result['id'],
                'is_ready'=>false,
            ]);
        }
        DB::commit();
        return redirect()->back()->with('diagnose_id',$result['id'])->withSuccess('Diagnose Added Successfully !');
    }catch (Exception $exception){
        DB::rollback();
        return redirect()->back()->withErrors('Something went wrong !');
    }
}

```



## 4.12. Download Prescriptions

- Model – Patient
- Controller – PDFController

```
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function getPDF($id)
{
    $prescription = Diagnose::where('diagnoses.id', $id)
        ->join('patients', 'diagnoses.patient_id', 'patients.id')
        ->join('users', 'diagnoses.doctor_id', 'users.id')
        ->join('prescriptions', 'diagnoses.id', 'prescriptions.diagnose_id')
        ->select(
            'diagnoses.posted_date as prescribed_date',
            'diagnoses.diagnose',
            'prescriptions.prescription',
            'prescriptions.id as prescription_id',
            'users.name as doctor_name',
            'patients.full_name as patient_name',
            'patients.dob as patient_dob'
        )->first();

    $diagnoseValues = DiagnoseValue::where('diagnose_id', $id)->first();

    $pdf = PDF::loadView('patient.pages.prescription',[
        'prescription' => $prescription,
        'diagnoseValues' => $diagnoseValues
    ]);
    return $pdf->download('prescription.pdf');
}
```

## Appendix D: User Manual

# Integrated Hospital Information System

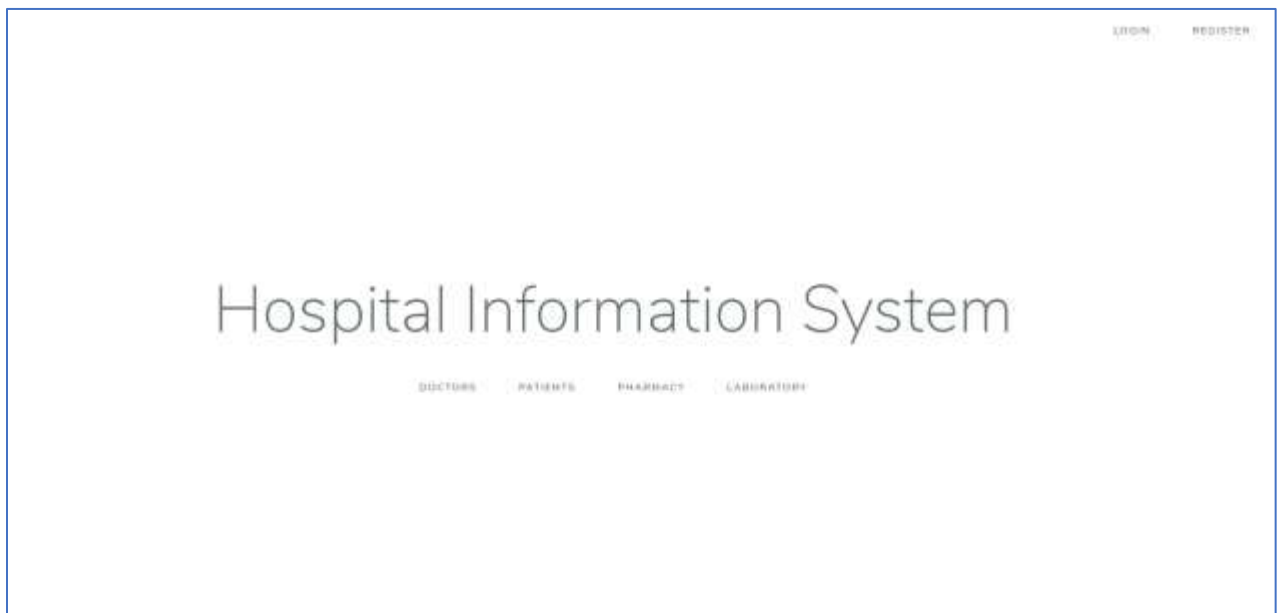
## User Manual

## 1. Purpose

The purpose of this document is to assist users to use the Hospital Information System without difficulties.

## 2. Home Page

- Home page of the system



- Login Screen

- > Click on the “Login” button will result below screen

No.	Action	ROC	Value
(01)	<b>Login:</b> Email Address: Password: Remember Me	C C O	Enter registered email and password Tick if want to remember user name and password

### 3. Patient

- Register

- › Click on “REGISTER” icon on the top right-hand corner of the ‘Home’ page



- › The results screen will as below

A screenshot of a web form titled 'Hospital System - Patients Registration'. Below the title is a sub-header 'User Registration Form'. The form contains four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Each field is followed by a blue 'Register' button.

No.	Action	ROC	Value
(01)	<b>Patient Registration:</b> Name: Email address: Password: Confirm Password:	C	Enter suitable details

- › Press “Register” button – Following screen will appear

## Patient Details Page

Full Name

Address

Contact No  Gender  Blood Group

Date of Birth  Height (m)  Weight (kg)  Patient Category

Allergies

Medical Condition

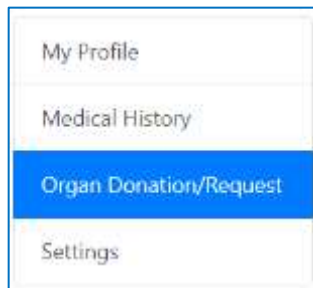
[Save Details](#)

No.	Action	ROC	Value
(02)	<b>Patient Details:</b> Full Name: Address: Contact No: Gender: Blood Group: Date of Birth: Height(m): Weight(kg): Patient Category: Allergies: Medical Condition:	C	Enter suitable details

- > Press the “Save Details” button
- > User will direct to “Patient Dashboard” page
- > Click on “Emergency details” tab and enter emergency contact details

No.	Action	ROC	Value
(03)	<b>Patient Details:</b> Contact Name: Contact No:	C	Enter suitable details

- › Click the “Save or Update” button
- Organ donation/request registration
  - › Select ‘Organ Donation/Request’ from the menu on the left-hand side of the screen.



**Organs**

All Donations Requests

Donation/Request ID	Organ Name	Description	Type	Posted Date
---------------------	------------	-------------	------	-------------

---

**Organ Donations/Requests**

Organ Name :

Type :

Description :

No.	Action	ROC	Value
(04)	<b>Organ Donation/Requests:</b> Organ Name: Type: Description	C	Type: select Donation or Request

- > Click the "Save" button.
- > A successful data entrance will result in the following message

**Donation request submitted successfully !**

- > Click on "All", "Donations", "Requests" buttons will filter data accordingly

**Organs**

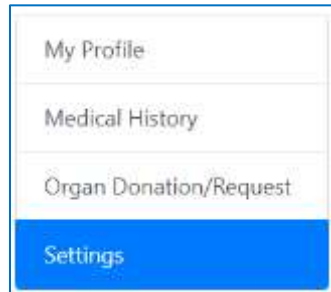
All Donations Requests

Donation/Request ID	Organ Name	Description	Type	Posted Date
3	Heart	Please take my heart	Donate	2019-05-23 05:29:56

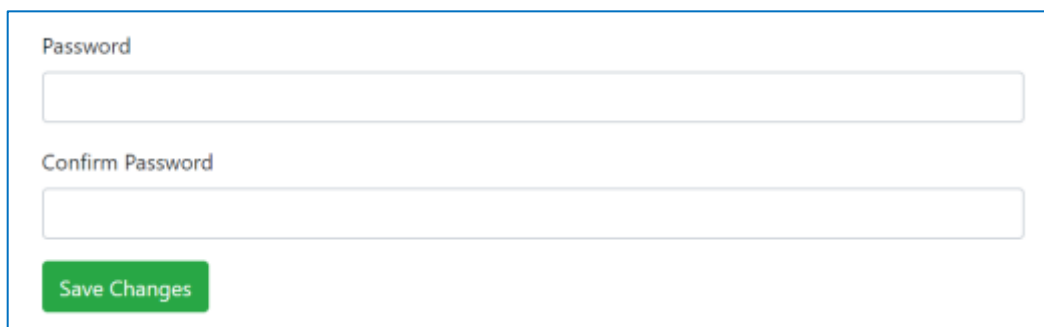


- **Reset Password**

- › Select “Settings” from the menu on the left-hand side of the screen.

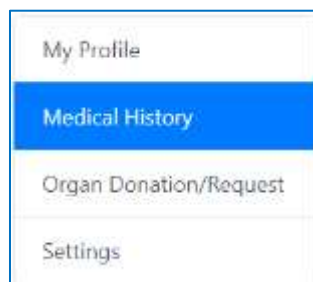


- › Enter a new password and click on “Save Changes” button.

A form with two input fields. The first is labeled "Password" and the second is labeled "Confirm Password". Below the fields is a green button labeled "Save Changes".

- **Medical History**

- › Select “Medical History” from the menu on the left-hand side of the screen.



- > Results screen – Patient Dashboard would be as follows

**Patient Dashboard**

Diagnoses | Prescriptions | Self Uploaded Reports | Lab Reports | Events | Statistic Reports

### Diagnose History

Doctor Name	Diagnose	Posted Date	
<b>Dr. Wasantha Madurawala</b>	Runny nose, Headache mild fever cough	2019-07-07 13:25:46	<a href="#">More</a>
<b>Dr. Dinushan Kaluarachchi</b>	Belly pain, vomit	2019-07-07 13:42:50	<a href="#">More</a>
<b>Dr. Wasantha Madurawala</b>	rewnwqrg	2019-09-04 13:36:01	<a href="#">More</a>
<b>Dr. Wasantha Madurawala</b>	US Embassy responds to Wimal's allegations	2019-09-06 11:07:37	<a href="#">More</a>
<b>Dr. Wasantha Madurawala</b>	All the world's cricketers walked into a bar, The LOLs, the chatter, and the occasional mini-brawls, they were endless. And the news cycle was never the same again!	2019-09-07 18:48:02	<a href="#">More</a>
<b>Dr. Wasantha Madurawala</b>	Twice now in international cricket, he has taken four wickets in four balls. Of the 100 hat-tricks in international cricket, he has a 5% share.	2019-09-07 18:57:31	<a href="#">More</a>

- > List down the doctor channeling history in the “Diagnoses” tab.
- > Click on the “More” button will display the details of the particular selected Doctor visit.
- > If a user decides to get the medical reports from an external laboratory, the user will be able to attach the same
  - Click on “Choose File” button to search documents from the computer and attach the report
  - Click on “Upload Report” button will save data.

### Diagnose

---

**Doctor Name :** Madurawi **Posted Date :** 2019-05-23 05:38:23  
**Diagnose :** headache, cold and high fever

### Prescription

---

paracetamol 3 times/day ,1/52  
 drug 2 - 3 times/day ,1/52  
 drug 3 - 3 times/day ,1/52  
 Status : Not yet

### Attached Reports

---

Report ID	Title	Description	Status	
5	Report 1	Please do the test 1 & test 2	Not Ready	<a href="#" style="background-color: #27ae60; color: white; padding: 5px 10px; border-radius: 3px;">Download Report</a>

#### Attach a report

Report Title

Upload your report

No file chosen

[Upload Report](#)

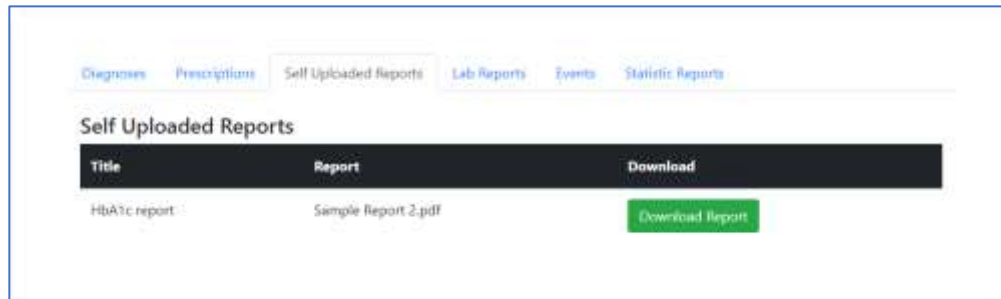
- Prescriptions tab → Displays the prescription history. “Download Prescription” button will allow the user to download the prescription.

Diagnoses
Prescriptions
Self Uploaded Reports
Lab Reports
Events
Statistic Reports

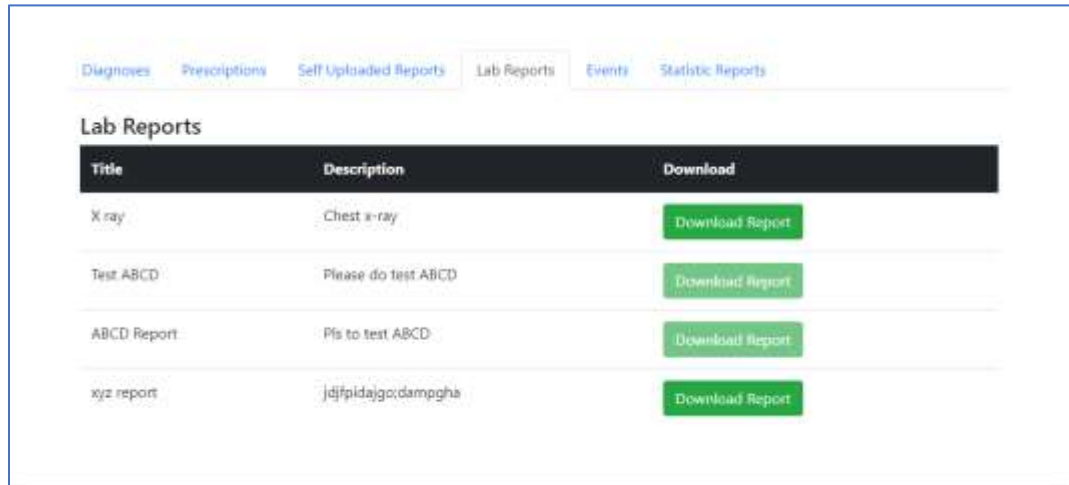
### Prescriptions

Diagnose	Prescription	Doctor	Date	Download
Runny nose, Headache mild fever cough	Paracetamol - 3 times per day - after meal - 1/52 Augmentine 3 times per day - after meal - 1/52 Cough syrup 2 times per day - after meal - 1/52	Dr. Wasantha Madurawala	Jul 7, 2019	<a href="#" style="background-color: #e74c3c; color: white; padding: 5px 10px; border-radius: 3px;">Download Prescriptions</a>

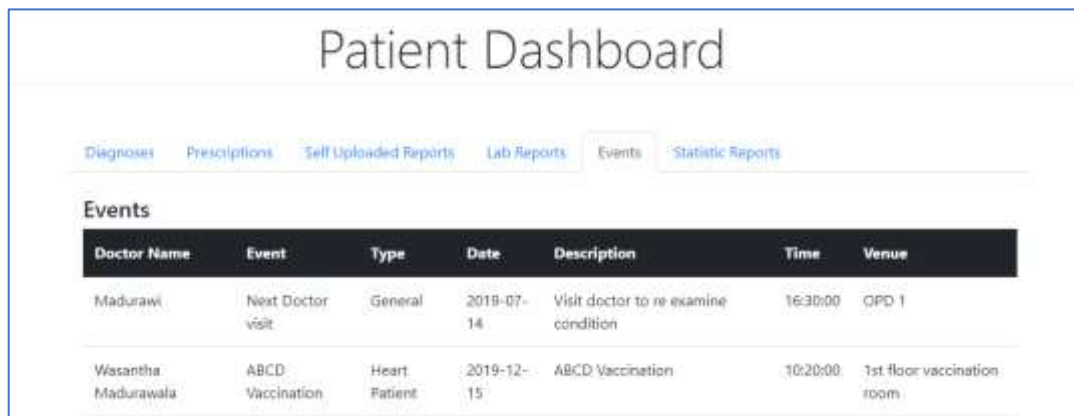
- › Self Uploaded Reports tab → Displays the user uploaded report to the system. If required, the user will also be able to download the report



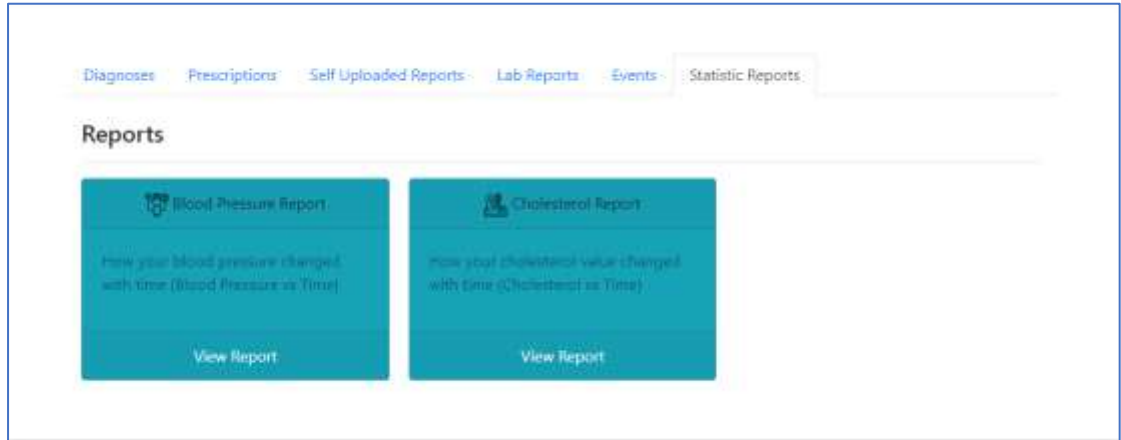
- › Lab Reports tab → Consists of laboratory reports uploaded by the hospital laboratory. User will be able to download the report by click on the “Download Report” button.



- › Events Tab → will display upcoming events set by a Doctor



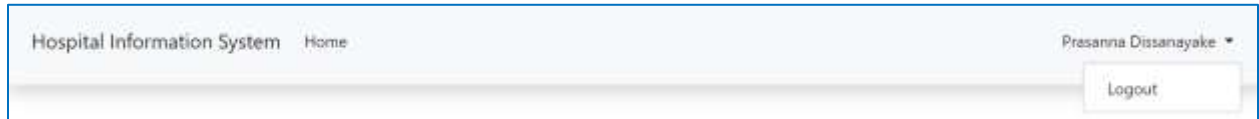
- Static Reports Tab → This tab display reports based on the patient category  
 Example: A diabetic patient would be able to view two reports namely HbA1C and Cholesterol while a heart patient would view Blood Pressure and Cholesterol report.



View report: Click on a report then select the required date range and click on the “Search” button



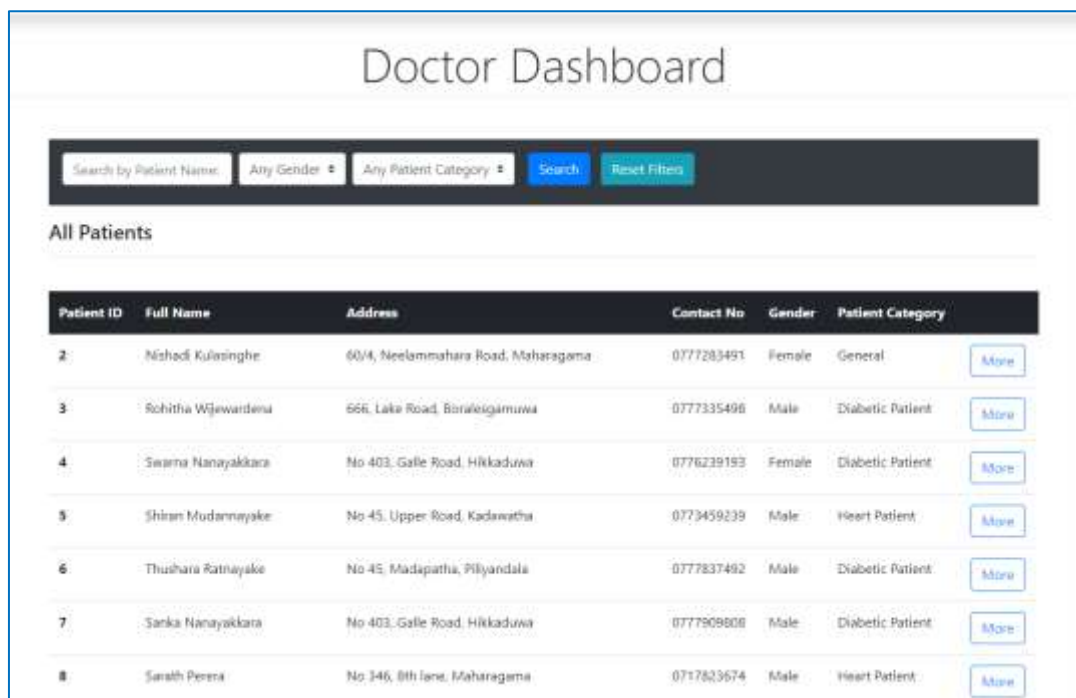
- Logout
  - > A click on the user name will pop out the “Logout” button



## 4. Doctor

- Doctor Dashboard

- > Log into the system with the given user name and password.



The screenshot displays the 'Doctor Dashboard' interface. At the top, there is a search bar with the following fields: 'Search by Patient Name', 'Any Gender' (dropdown), 'Any Patient Category' (dropdown), a 'Search' button, and a 'Reset Filters' button. Below the search bar, the text 'All Patients' is displayed. A table lists eight patients with columns for Patient ID, Full Name, Address, Contact No, Gender, and Patient Category. Each row also includes a 'More' button.

Patient ID	Full Name	Address	Contact No	Gender	Patient Category	
2	Nishodi Kulesinghe	60/4, Neelammahara Road, Maharagama	0777283491	Female	General	<a href="#">More</a>
3	Rohitha Wijewardena	666, Lake Road, Boralesgamuwa	0777335496	Male	Diabetic Patient	<a href="#">More</a>
4	Swarna Nanayakkara	No 403, Galle Road, Hikkaduwa	0776239193	Female	Diabetic Patient	<a href="#">More</a>
5	Shirsh Mudannayake	No 45, Upper Road, Kadawatha	0773459239	Male	Heart Patient	<a href="#">More</a>
6	Thushara Ratnayake	No 45, Madapatha, Piliyandala	0777837492	Male	Diabetic Patient	<a href="#">More</a>
7	Sanka Nanayakkara	No 403, Galle Road, Hikkaduwa	0777909808	Male	Diabetic Patient	<a href="#">More</a>
8	Sarath Perera	No 346, 8th Lane, Maharagama	0717823674	Male	Heart Patient	<a href="#">More</a>

- > A doctor would be able to search patients by following parameters
  - Patient Name
  - Gender
  - Patient Category
- > Click on the “Search” button would filter the list and “Reset Filters” button would clear filters
- > Click on the “More” button next to a patients’ name, will result below screen

### Jeewana Premaratne's Profile

<b>Patient Name :</b> Jeewana Premaratne	<b>Patient Category :</b> Heart Patient
<b>Address :</b> No 203, Thalagama Road, Battaramulla	<b>Blood Group :</b> O+
<b>Age :</b> 32 Years	<b>Weight(kg) :</b> 70
<b>Gender :</b> Male	<b>Height(m) :</b> 1.56
<b>Allergies :</b> amoxicillin (Moxatag), ampicillin, penicillin	
<b>Medical Condition :</b> Kidney disease	

---

[Add Diagnose](#)   [Request Lab Reports](#)   [Patient Diagnose History](#)   [Add Events](#)

Diagnose :

Prescription :

Cholesterol :       Blood Pressure :

Get Medicine From :

▼

[Save](#)

- > Add Diagnosis tab → Doctor could diagnosis and prescription. Enter values for 'Cholesterol' and 'Blood Pressure' [Note: value fields would be changed based on patient category]
- > Select either 'Hospital Pharmacy' or 'External Pharmacy' from the drop-down.
- > Click on the "Save" button will save data
- > Request Lab Report tab → allows the doctor to request laboratory reports

### Jeewana Premaratne's Profile

<b>Patient Name :</b> Jeewana Premaratne	<b>Patient Category :</b> Heart Patient
<b>Address :</b> No 203, Thalagama Road, Battaramulla	<b>Blood Group :</b> O+
<b>Age :</b> 32 Years	<b>Weight(kg) :</b> 70
<b>Gender :</b> Male	<b>Height(m) :</b> 1.56
<b>Allergies :</b> amoxicillin (Moxatag), ampicillin, penicillin	
<b>Medical Condition :</b> Kidney disease	

---

[Add Diagnose](#)   [Request Lab Reports](#)   [Patient Diagnose History](#)   [Add Events](#)

Report Title :

Report Description :

[Request Report](#)



- › Patient Diagnosis History Tab → Displays the previous doctor visits of the patient

**Jeewana Premaratne's Profile**

**Patient Name :** Jeewana Premaratne  
**Address :** No 203, Thalangama Road, Battaramulla  
**Age :** 32 Years  
**Gender :** Male  
**Allergies :** amoxicillin (Moxatag), ampicillin, penicillin  
**Medical Condition :** Kidney disease

**Patient Category :** Heart Patient  
**Blood Group :** O+  
**Weight(kg) :** 70  
**Height(m) :** 1.56

[Add Diagnose](#) [Request Lab Reports](#) [Patient Diagnose History](#) [Add Events](#)

Doctor Name	Diagnose	Posted Date	
Dr. Wasantha Madurawala	Runny nose, Headache mild fever cough	2019-07-07 13:25:46	<a href="#">More</a>
Dr. Dinushan Kaluarachchi	Belly pain, vomit	2019-07-07 13:42:50	<a href="#">More</a>

- › Add the Events Tab section → allows entering future events for a patient
- › Fill the required details and click on the “Submit Event” button will save data

**Jeewana Premaratne's Profile**

**Patient Name :** Jeewana Premaratne  
**Address :** No 203, Thalangama Road, Battaramulla  
**Age :** 32 Years  
**Gender :** Male  
**Allergies :** amoxicillin (Moxatag), ampicillin, penicillin  
**Medical Condition :** Kidney disease

**Patient Category :** Heart Patient  
**Blood Group :** O+  
**Weight(kg) :** 70  
**Height(m) :** 1.56

[Add Diagnose](#) [Request Lab Reports](#) [Patient Diagnose History](#) [Add Events](#)

Event Name  Patient Type  Date

Event Description

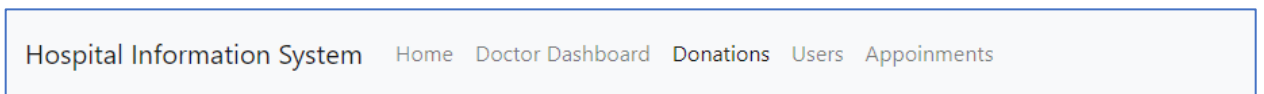
Time  Venue

[Submit Event](#)

No.	Action	ROC	Value
(01)	<b>Add Event:</b> Event Name: Patient type: Date: Event description: Time Venue:	C O C C C O	Enter data as applicable

- Donations

- › Click on ‘Donations’ in the header panel, will result in the following screen



- › Displays all donations and requests for Organs

**All Donations/Requests**

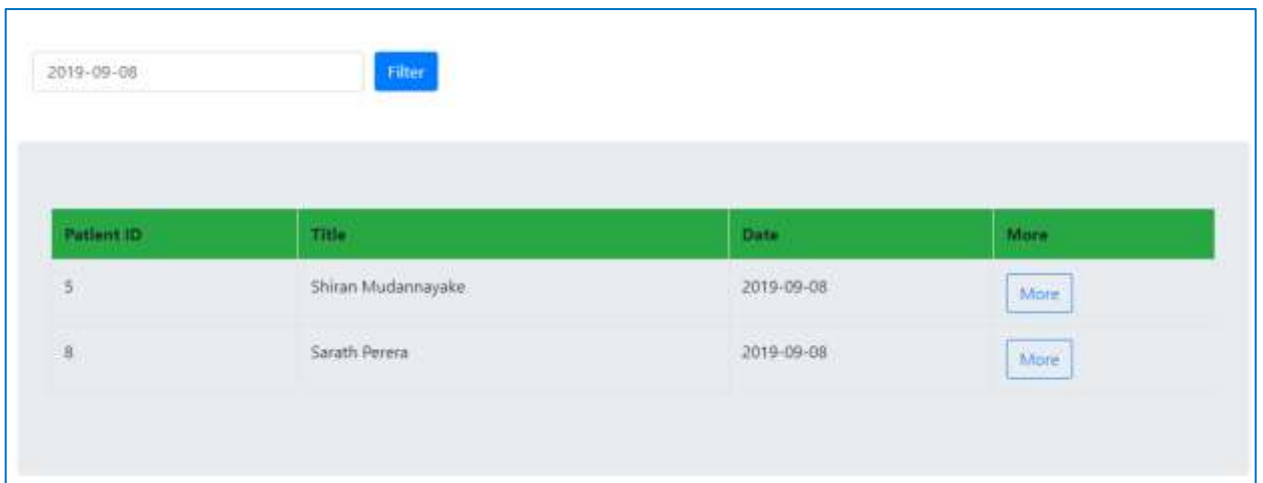
Donation/Request ID	Patient Name	Organ Name	Description	Posted Date	Type
1	Jeewana Premaratne	Heart	would like to donate	2019-07-06 03:02:15	Donate
2	Nishadi Kulasinghe	Brain	need a brain	2019-07-06 14:34:31	Request
3	Sarath Perera	Kidney	Urgently required a O+ kidney	2019-09-07 17:48:12	Request
4	Sanka Nanayakkara	Brain	Would like to donate brain	2019-09-07 17:48:55	Donate
5	Jeewana Premaratne	Kidney	would like to donate a kidney for a suitable patient	2019-09-07 17:49:52	Donate
6	Nishadi Kulasinghe	Eye	Agree to donate eyes	2019-09-07 17:50:59	Donate

- Appointments

- › Click on 'Appointments' in the header panel will result in the following screen

Hospital Information System   Home   Doctor Dashboard   Donations   Users   Appointments

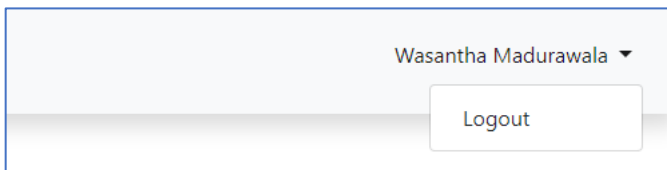
- › User will be able to filter appointments for a date
- › Click on the "More" button will direct to patient's profile



The screenshot shows a web interface for appointments. At the top, there is a search bar containing the date "2019-09-08" and a blue "Filter" button. Below this is a table with a green header and two data rows. Each row has a "More" button in the rightmost column.

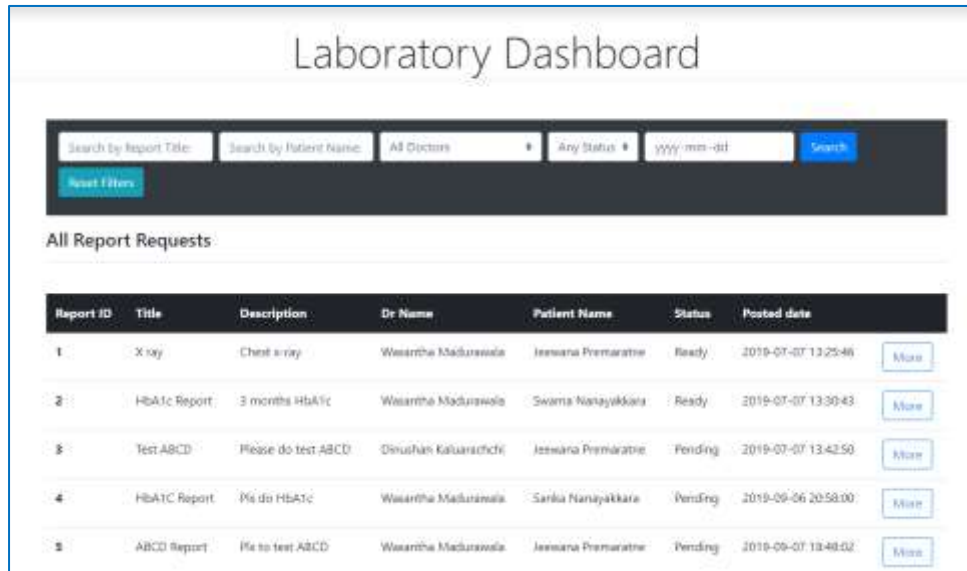
Patient ID	Title	Date	More
5	Shiran Mudannayake	2019-09-08	More
8	Sarath Perera	2019-09-08	More

- Click on the user name will pop out the "Logout" button and will enable the user to sign out from the system



## 5. Laboratory

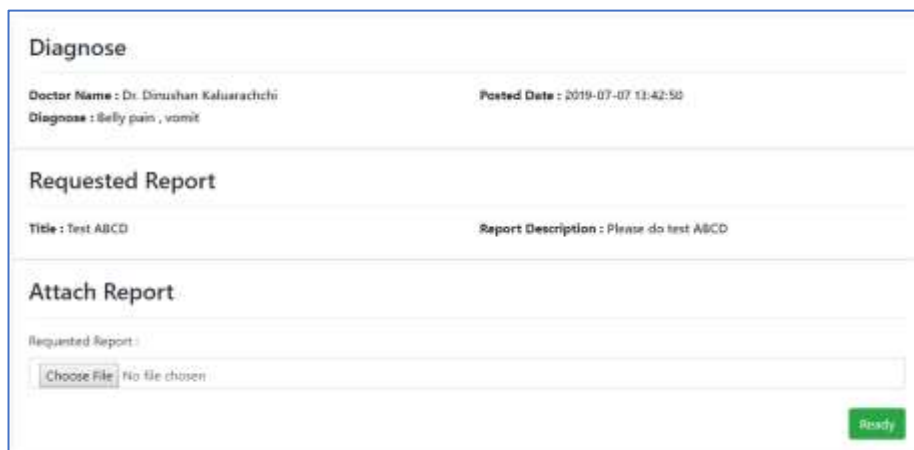
- A successful login to the system will result in the following dashboard screen



The screenshot shows the 'Laboratory Dashboard' interface. At the top, there is a search bar with fields for 'Search by Report Title', 'Search by Patient Name', 'All Doctors', 'Any Status', and a date field 'yyyymm-dd'. A 'Search' button is on the right, and a 'Reset Filters' button is on the left. Below the search bar, the section is titled 'All Report Requests'. It contains a table with the following data:

Report ID	Title	Description	Dr Name	Patient Name	Status	Posted date	More
1	X ray	Chest x ray	Wasantha Madurawala	Jeevana Premaratne	Ready	2019-07-07 13:25:46	More
2	HbA1c Report	3 months HbA1c	Wasantha Madurawala	Swarna Nanayakkara	Ready	2019-07-07 13:30:43	More
3	Test ABCD	Please do test ABCD	Dinushan Kaluarachchi	Jeevana Premaratne	Pending	2019-07-07 13:42:50	More
4	HbA1c Report	File do HbA1c	Wasantha Madurawala	Sarika Nanayakkara	Pending	2019-06-06 20:58:00	More
5	ABCD Report	File to test ABCD	Wasantha Madurawala	Jeevana Premaratne	Pending	2019-09-07 10:48:02	More

- › Reports can be filtered by the following options
  - Report title
  - Patient name
  - Doctor name
  - Report status
  - Date
- › Click on the “More” button will result in the following screen



The screenshot shows the 'Diagnose' screen. It displays the following information:

- Diagnose**
- Doctor Name : Dr. Dinushan Kaluarachchi
- Posted Date : 2019-07-07 13:42:50
- Diagnose : Belly pain , vomit
- Requested Report**
- Title : Test ABCD
- Report Description : Please do test ABCD
- Attach Report**
- Requested Report :
- Choose File | No file chosen
- Ready

- › Click on “Choose file” to browse relevant report and attach to the request
- › Click on the “Ready” button will save the report and status will change to “Ready”
- Click on ‘Users’ button in the header panel will display the following screen. It allows users to reset their password

Hospital Information System   Home   Laboratory Dashboard   Users

Name

Email address  
  
We'll never share your email with anyone else.

Password

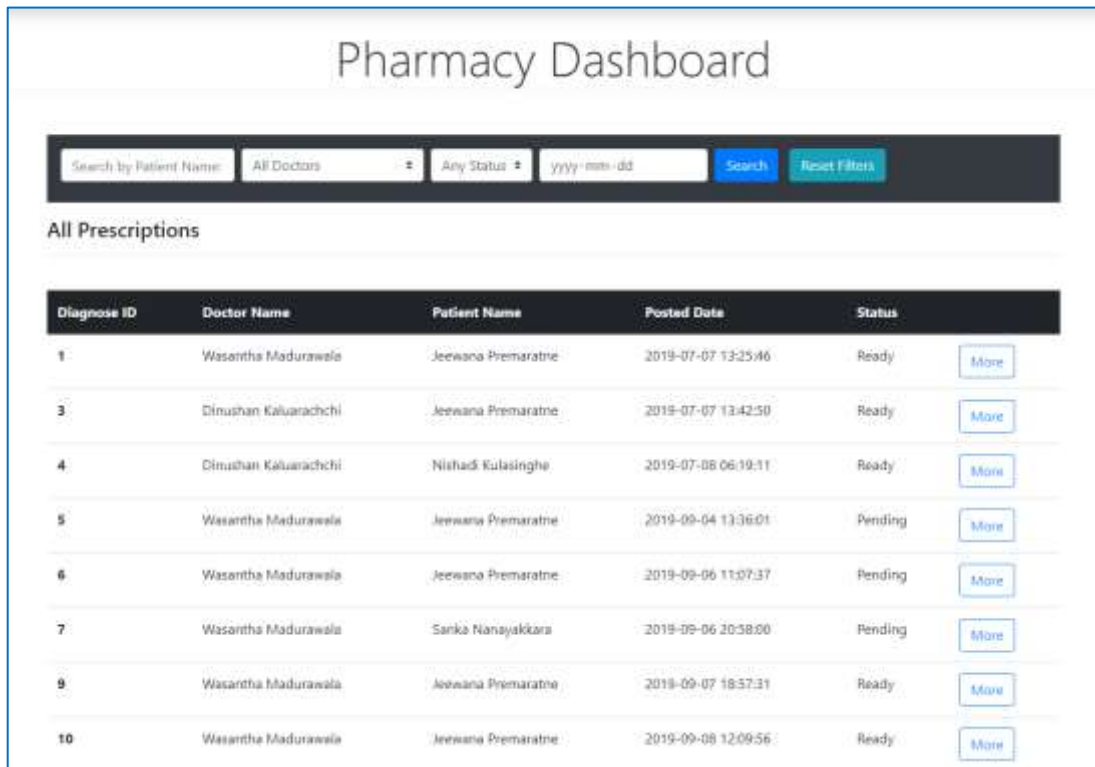
- Click on the user name will pop out the “Logout” button to sign out from the system

Lab ▼

Logout

## 6. Pharmacy

- A successful login to the system will result following Pharmacy Dashboard screen
- If a patient decides to take the prescribed drugs from the hospital pharmacy, such requests will appear in the following dashboard



The screenshot displays the Pharmacy Dashboard interface. At the top, there is a search bar with the following fields: "Search by Patient Name", "All Doctors" (dropdown), "Any Status" (dropdown), and "yyyy-mm-dd" (date input). There are "Search" and "Reset Filters" buttons. Below the search bar, the heading "All Prescriptions" is visible. The main content is a table with the following columns: "Diagnose ID", "Doctor Name", "Patient Name", "Posted Date", and "Status". Each row also includes a "More" button.

Diagnose ID	Doctor Name	Patient Name	Posted Date	Status	
1	Wasantha Madurawala	Jeevana Premaratne	2019-07-07 13:25:46	Ready	<a href="#">More</a>
3	Dinushan Kaluarachchi	Jeevana Premaratne	2019-07-07 13:42:50	Ready	<a href="#">More</a>
4	Dinushan Kaluarachchi	Nishad Kulasinghe	2019-07-08 06:19:11	Ready	<a href="#">More</a>
5	Wasantha Madurawala	Jeevana Premaratne	2019-09-04 13:36:01	Pending	<a href="#">More</a>
6	Wasantha Madurawala	Jeevana Premaratne	2019-09-06 11:07:37	Pending	<a href="#">More</a>
7	Wasantha Madurawala	Sanka Nanayakkara	2019-09-06 20:58:00	Pending	<a href="#">More</a>
9	Wasantha Madurawala	Jeevana Premaratne	2019-09-07 18:57:31	Ready	<a href="#">More</a>
10	Wasantha Madurawala	Jeevana Premaratne	2019-09-08 12:09:56	Ready	<a href="#">More</a>

- › Requests can be filtered by using the following options
  - Patient name
  - Doctor name
  - Status
  - Date

- › Click on the “More” button will result in the following screen

**Jeewana Premaratne's Profile**

---

**Patient Name :** Jeewana Premaratne  
**Address :** No 203, Thalangama Road, Battaramulla  
**Age :** Years  
**Gender :** Male  
**Allergies :** amoxicillin (Moxatag), ampicillin, penicillin  
**Medical Condition :** Kidney disease

**Patient Category :** Heart Patient  
**Blood Group :** O+  
**Weight :** 70  
**Height :** 1.56

---

**Diagnose**

---

**Doctor Name :** Wasantha Madurawala  
**Diagnose :** Runny nose. Headache mild fever cough  
**Posted Date :** 2019-07-07 13:25:46

---

**Prescription**

---

Paracetamol - 3 times per day - after meal - 1/52  
Augmentine 3 times per day - after meal - 1/52  
Cough syrup 2 times per day - after meal - 1/52

[Ready](#)

- › Once the drugs are ready to collect, click on the “Ready” button

- Click on ‘Users’ button in the header panel will display the following screen. It allows users to reset their password

Hospital Information System   Home   Pharmacy Dashboard   Users

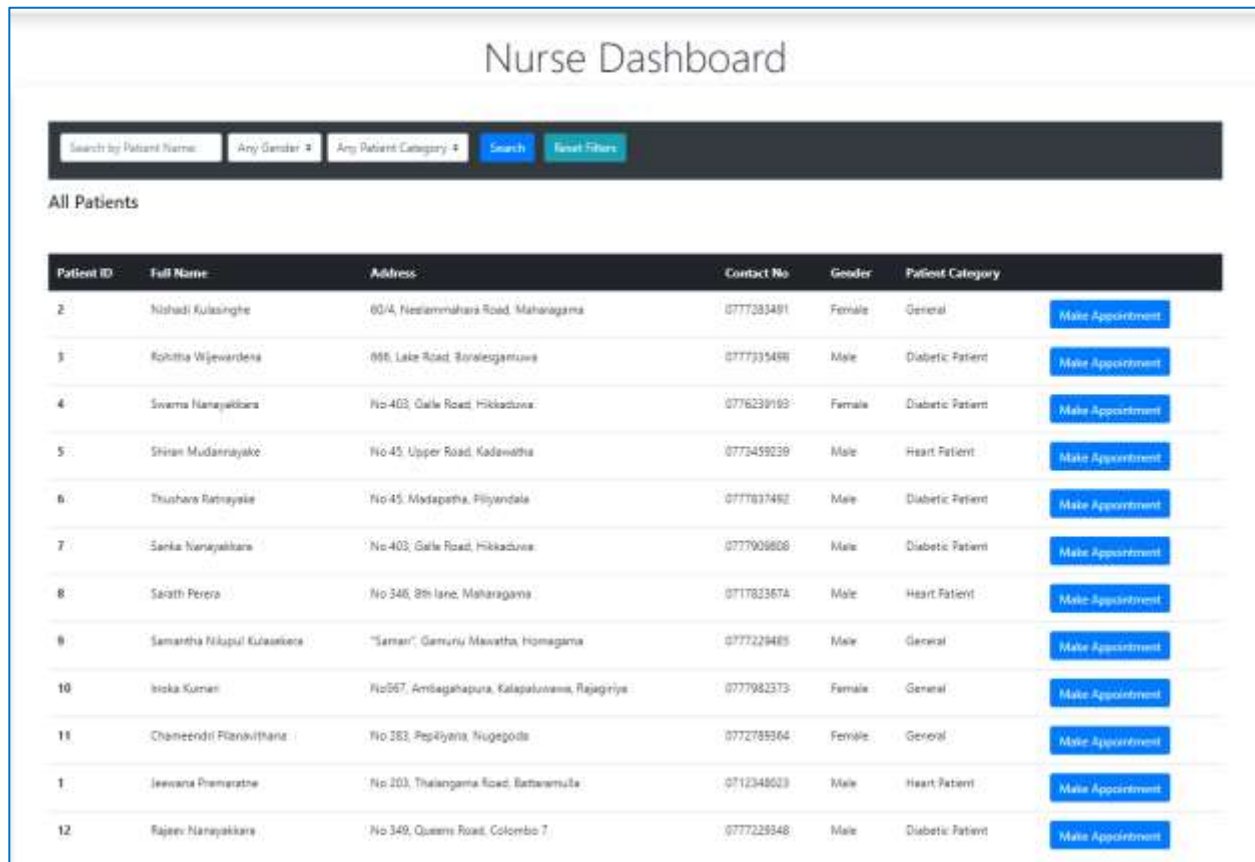
- Click on the user name will pop out the “Logout” button to sign out from the system

Pharmacy ▾

Logout

## 7. Nurse

- A successful login to the system will result following 'Nurse Dashboard' screen
- The result screen will list down all registered patients



The screenshot displays the 'Nurse Dashboard' interface. At the top, there is a search bar with three input fields: 'Search by Patient Name', 'Any Gender', and 'Any Patient Category'. Below the search bar, the text 'All Patients' is visible. The main content is a table with the following columns: Patient ID, Full Name, Address, Contact No, Gender, Patient Category, and a 'Make Appointment' button for each row.

Patient ID	Full Name	Address	Contact No	Gender	Patient Category	Make Appointment
2	Nishadi Kulasinghe	60/A, Neralmmahara Road, Maharagama	0777333491	Female	General	Make Appointment
3	Kotilla Wijewardena	696, Lake Road, Soralogampawa	0777335498	Male	Diabetic Patient	Make Appointment
4	Swarna Nanayakkara	No 403, Gale Road, Hikkadulla	0776239193	Female	Diabetic Patient	Make Appointment
5	Shirani Mudanayake	No 45, Upper Road, Kadawatha	0775459239	Male	Heart Patient	Make Appointment
6	Thushara Ratnayake	No 45, Madapatha, Piliyandala	0777637462	Male	Diabetic Patient	Make Appointment
7	Sanka Nanayakkara	No 403, Gale Road, Hikkadulla	0777909808	Male	Diabetic Patient	Make Appointment
8	Sarath Perera	No 346, 8th lane, Maharagama	07717833674	Male	Heart Patient	Make Appointment
9	Samantha Nilupul Kulasiri	"Saman", Gemunu Mawatha, Homagama	0777229485	Male	General	Make Appointment
10	Inoka Kumari	No 567, Ambagahapura, Kalapaluzuwewa, Rajagiriya	0777942373	Female	General	Make Appointment
11	Chameendri Ranawithana	No 383, Peraliyana, Nugegoda	0772785964	Female	General	Make Appointment
1	Jeevana Premaratne	No 203, Thalagama Road, Battaramulla	0712348623	Male	Heart Patient	Make Appointment
12	Rajeev Nanayakkara	No 349, Queens Road, Colombo 7	0777228948	Male	Diabetic Patient	Make Appointment

- Patients can be filtered by the following parameters
  - Patient name
  - Gender
  - Patient category



- Click on the “Make Appointment” button will result in the following screen

Making an appointment for Shiran Mudannayake

Patient Name : Shiran Mudannayake  
 Address : No 43, Upper Road, Baddegama  
 Age : 40 Years  
 Gender : Male

Patient Category : Heart Patient  
 Blood Group : D+  
 Weight(kg) : 90  
 Height(m) : 1.78

Search by Doctor Name: Search by Doctor Speciality Search Reset Filters

Name	Speciality	
Dr. Wasantha Madurawa	Cardiologist	Select Doctor
Dr. Dinushan Kaluarachchi	Physician	Select Doctor
Dr. Dinusha Ganage	Physician	Select Doctor
Dr. Priyankara Jayawardena	Physician	Select Doctor
Dr. Chinthaka Athukorala	ENT Surgeon	Select Doctor

Appointment Title :  
 Give a Title / Name for appointment

Appointment Time :  
 08:14 AM

Appointment Date :  
 2019-00-38

Make Appointment Reset

- A doctor can be searched by name or speciality.

No.	Action	ROC	Value
(01)	<b>Add New appointment:</b> Select Doctor: Appointment Title: Time: Date:	C C C C	Select doctor from the list

- › Click on the “Make Appointment” button will save an appointment for the selected doctor
- › The “Reset” button will clear-out the data entered

## 8. Administrator

- A successful login to the system will result following 'Admin Dashboard' screen

### Admin Dashboard

Add new user

Name  
Enter name

User Type  
Doctor

Speciality  
Enter Doctor Speciality

Email address  
Enter email  
We'll never share your email with anyone else.

Password  
Password

Submit

---

Search by Username   Search by Email   All User Types   Search   Reset Filters

Name	Email	User Type	Remove
Wasantha Madurawala	maduravi@gmail.com	DOCTOR	Remove User
Pharmacy	pharmacy@gmail.com	PHARMACY	Remove User
Admin	admin@gmail.com	ADMIN	Remove User
Jeevana Premaratne	sanku08@gmail.com	PATIENT	Remove User

- **Create New Users**

- › Doctor

The screenshot shows a form titled "Add new user" with the following fields and elements:

- Name:** A text input field with the placeholder text "Enter name".
- User Type:** A dropdown menu with "Doctor" selected.
- Speciality:** A text input field with the placeholder text "Enter Doctor Speciality".
- Email address:** A text input field with the placeholder text "Enter email". Below this field is a small text note: "We'll never share your email with anyone else."
- Password:** A text input field with the placeholder text "Password".
- Submit:** A blue button labeled "Submit".

- › Other users – Nurse, Pharmacy, Laboratory

The screenshot shows a form titled "Add new user" with the following fields and elements:

- Name:** A text input field with the placeholder text "Enter name".
- User Type:** A dropdown menu with "Nurse" selected.
- Email address:** A text input field with the placeholder text "Enter email". Below this field is a small text note: "We'll never share your email with anyone else."
- Password:** A text input field with the placeholder text "Password".
- Submit:** A blue button labeled "Submit".

No.	Action	ROC	Value
(01)	<b>Add New user:</b> Name: User type: Specialty: Email address: Password:	C C R C C	User type: Doctor/Nurse/Lab/Pharmacy Specialty: Applicable only for user type - "Doctor"

> Click on the "Submit" button will save data

- All Users

- > Displays all users created in the system
- > Filtering can be done by User name, Email address or user type
- > "Remove user" button will erase users from the system

Name	Email	User Type	Remove
Wasantha Madurawala	maduravi@gmail.com	DOCTOR	<a href="#">Remove User</a>
Pharmacy	pharmacy@gmail.com	PHARMACY	<a href="#">Remove User</a>
Admin	admin@gmail.com	ADMIN	<a href="#">Remove User</a>
Jeevana Premaratne	senka08@gmail.com	PATIENT	<a href="#">Remove User</a>
Nishadi Kulesinghe	nishadi@gmail.com	PATIENT	<a href="#">Remove User</a>
Rohitha Wijewardena	rohitha@gmail.com	PATIENT	<a href="#">Remove User</a>
Swarna Gunasekera	swarna@gmail.com	PATIENT	<a href="#">Remove User</a>
Dinushan Kaluarachchi	dinushan@gmail.com	DOCTOR	<a href="#">Remove User</a>
Shiran Mudannayake	shiran@gmail.com	PATIENT	<a href="#">Remove User</a>
Lab	lab@gmail.com	LAB	<a href="#">Remove User</a>
Nurse	nurse@gmail.com	NURSE	<a href="#">Remove User</a>
Thushara Ratnayake	thushara@gmail.com	PATIENT	<a href="#">Remove User</a>
Dinusha Gamage	dinusha@gmail.com	DOCTOR	<a href="#">Remove User</a>
Priyanika Jayawardena	priyanika@gmail.com	DOCTOR	<a href="#">Remove User</a>
Sarath Perera	senka08@yahoo.com	PATIENT	<a href="#">Remove User</a>
Samantha Kulasekara	samantha@gmail.com	PATIENT	<a href="#">Remove User</a>
Inoka Kumari	inoka@gmail.com	PATIENT	<a href="#">Remove User</a>