Web Based Report & Invoice Generating System

For

Hope Diagnostics Medical Laboratory Service

K.L.Viraj Prasanna

Registration No: R110691

Index No: 1106912

Supervised by

Mr. Chamira Dharmawijaya

17-February-2018

This dissertation is submitted in partial fulfilment of the requirement of the

Degree of Bachelor of Information Technology (external) of the

University of Colombo School of Computing

Declaration

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

Signature of Candidate: ..... Date: 06/11/2017

Name of Candidate: K.L.V.Prasanna

Countersigned by:

Signature of Supervisor:..... Date: 06/11/2017

Name of Supervisor: Mr. Chamira Dharmawijaya

ABSTRACT

Hope Diagnostics Medical Laboratory Service is providing a great service to people who live in Panadura area by doing their medical laboratory testings accurately. Currently they have nine centres to collect samples and collected samples are brought to Panadura main laboratory for testings.

Hope Diagnostics has been using Ms Excel sheets for report printing and a computerized system only for the payment transactions. The process involves entering details and results of each patient one by one to the excel sheet. This can be time consuming, leading to decreased productivity, data redundancy, lack of accuracy and etc. In order to eliminate this issue, and to improve their service for community, the administration has decided to implement a web-based solution.

The web-based report and invoice generating system is designed and to be implemented to overcome those problems mentioned above through automating their processes. This system is mainly designed for automating the data manipulation.

Proposed system is developed based on the Rational Unified Process(RUP) methodology. This methodology is selected since the proposed system has iterative functions. RUP is an appropriate software development life cycle methodology for systems like this proposed system.

This web based system is basically designed by using the Unified Modelling Language (UML). PHP Hypertext Pre-processor (PHP) [7] [8] for scripting, MySQL [7][8] for database handling and Apache as the local server are built in WAMP server for web development and html, [7] CSS [7], JavaScript [7] [8] for web designing and Windows as the platform are to be used for the system development. Hope this designed web-based report and invoice generating system for Hope diagnostics medical laboratory service will ensure a quality service for the company and to the community.

ACKNOWLEDGEMENT

I am very much grateful to the external degrees centre of University of Colombo School of Computing for giving me a great opportunity to carry out an individual project to obtain the Bachelor of Information Technology and guiding me to succeed in my project.

I would like to thank the academic staff of Matrix Institute of Information Technology for giving me the academic knowledge throughout the course. My special thank goes to my supervisor Mr. Chamira Dharmawijaya for guiding me throughout the project. I am grateful for his great support to succeed my project.

Many thanks to Mr. Jeewantha Hasaranga and his staff for their supportive hand to succeed my project. Finally, a heartfelt gratitude to everyone, especially to my wife and family who guided and supported me as well as encouraged me in every possible way throughout the Bachelor of Information Technology degree program.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

BIT -Bachelor of Information Technology

HTML -Hyper Text Markup Language

CSS - Cascading Style Sheets

Ajax - Asynchronous JavaScript and XML.

JSON- JavaScript Object Notation

PDF - Portable Document Format

UML- Unified Modelling Language

ERD- Entity Relationship Diagram

RUP- Rational Unified Process

CHAPTER 01: INTRODUCTION

1.1. INTRODUCTION TO HOPE DIAGNOSTICS

Hope Diagnostics was established on 09th of October 2009, determined to provide a quality service for their customers. It has risen to be the leading medical laboratory service provider in Panadura area. This private business was empowered by Business Names Ordinance (Chapter 149) Incorporated by Business Names Statute No 4 of 1990.

Continuous progressive development due to time to time implemented scattered master plans of the major objective led them to climb the ladder of success within a very short period of time while penetrating to all the customer segments by being so flexible. Overcoming all challenges in a way of going ahead towards the success with an innovative solution was always one of their major key features for their success. Currently, they are fully flourished, established and equipped sophisticatedly in their category of representation with a warm heart always ready to serve their customers

As an area based flagship clinical diagnostic metrological establishment with a sole mission, Hope Diagnostics has stated their vision and mission as follows.

Vision

Our Vision is a Hope for, Accuracy, Quality, Charity, Affordability, Flexibility, Reliability and Traceability in You.

Mission

Our Mission is a Hope in you through hard work Dedication Innovation Extra Miles Understanding & Endurance

As a leading clinical metrological service provider in the area, Hope Diagnostics are working hard and dedicating to provide the best service to their customers and fulfil their vision as well as mission.

1.2. MOTIVATION FOR THE PROJECT

Hope diagnostics currently has nine centres for sample collecting whereby the employees of Hope diagnostics collect & transport the samples to the Panadura main laboratory to be tested. Although they have computerized system, they use it only for payment transactions. Following test completion, the results are entered in to an excel sheet format to generate printed medical reports which are then distributed to the nine centres.

Administration requires monthly reports on income, the tests most requested, etc. Currently, this is carried out manually by entering details in books followed by printing reports and delivering them to each centre themselves. This process is time consuming at most times and report delivering can be delayed due to several reasons such as traffic on the road. This has become a major concern for Hope diagnostics.

In some critical situations, doctors may need to check their patients' tests history, such as sugar, cholesterol etc. and the patient might have lost his/her previous reports. Therefore, it is very important to have records at least of regular patients of the medical centre. Therefore, providing patients with their previous reports may be an advantage for Hope Diagnostics to increase their number of customers.

Nowadays most industries are using internet and web applications to proceed their businesses productively. Hope Diagnostics can also make their service more productive by automating their processes and develop their business by providing new facilities to the customer.

As a final year student of Bachelor of Information Technology Degree Programme in University of Colombo School of Computing, I am eager to apply my acquired knowledge during my course period, in to developing an application to overcome Hope Diagnostics' difficulties while facing the challenges which arise towards a developer.

1.3. OBJECTIVES OF THE PROJECT

The main objective of this project is to support Hope Diagnostics to make their service more productive by automating their processes and develop their businesses by providing more facilities to their customers.

Avoid difficulties when getting payments and report printing.

Currently the patients' payments and report printing are processed separately, resulting repeat entering of customer/patient details. This project aims to facilitate them to make those processes easy by interconnecting those processes.

Develop the website to view medical report online private and confidentially
Hope to facilitate patients to view and take print outs of their laboratory reports via online to avoid delay.

Facilitate regular users to register and maintain their accounts to keep their past results safely.
Regular patients can register and maintain their separate accounts to keep their past results safely and print their reports if needed.

Facilitate patients to be notified when their results are ready.
When the results are ready, an auto generated email will be sent to patient notifying them that his/her report is ready and can be viewed online.

Facilitate patients to get cost estimation before come to the lab.
The prices of tests will appear on the website and patients can get estimation for their tests online. Packages provided by the client will be added under this.

Facilitate admin to take admin reports
Admin will be provided with the facility to generate reports on relevant areas as the admin requires. Such as,

- How much tests have been done within a month.
- What are the most demanded tests months.

Facilitate doctors to log in to view results of patients
Doctors will be provided a login to the system and they can view their patients' results and the regular patients' history. If a doctor needs a medical analysis of a regular patient, he will be facilitated for that purpose.

Facilitate patients to channel doctors online
As an additional service, hope diagnostics likes to provide e-channelling service for their patients. Through this option, hope diagnostics hope to fulfil their customer needs well.

1.4. SCOPE

The scope for this project contains,

- Two sections should be maintained for payment transaction and report printing. Two tables should be interconnected using a foreign key to make report printing easy.
- Patients should be able to register online, edit their details, view their resulting reports, keep previous reports and print their reports themselves. This is divided in to two sections according to patient type. Regular Patients and Random Patients. Regular patients should have some more advantages. So they can keep records of their previous results.
- The patients should be able to be notified when their reports are ready. For that function, an automated email should be delivered to the patient.
- The patients should be able to estimate their cost before coming lab and they should have options for payments. And when a deal is made by a patient, the admin should be notified with an automated email.
- Doctors who collaborate with Hope Diagnostics should have doctor's login and they should be able to login their account and check the reports of patients. If they want to get an analysing on particular test, they should have the options for their analysing.
- Admin should be able to login to the system and take reports on particular subjects such as, the income in a particular month, test details which are done in a particular month, etc
- And the patient should have the option to channel a doctor of the medical centre.

1.5. STRUCTURE OF THE DISSERTATION

The dissertation has been contained six chapters with fully detailed and it is providing great idea about the project.

This chapter has been included about the Hope diagnostics and a rough idea about the project which is done for Hope Diagnostics.

CHAPTER 02-ANALYSIS

When an IT project has to been started, the first thing a developer do is analysing. In this stage, all subject areas related to the project have been discussed and analysed and the functional and non-functional requirements will be identified which are helpful for designing the system.

CHAPTER 03-DESIGN

After all the analyses, the next job is to design the system according to user requirements. In this chapter, the used designing approach, database and user interface designing [5] will be discussed. And some diagrams which illustrate the functionality of system will be included in this chapter.

CHAPTER 04-IMPLEMENTATION

The development procedure of the system is included in this chapter. All coding of modules, user interfaces, creation of databases have been included in this chapter.

CHAPTER 05-EVALUATION

Testing procedure and results of it will be discussed in this chapter. Unit and integration tests, errors found, how they are corrected, and functionalities that satisfy the all user requirements in accepted quality have been mentioned in this chapter.

CHAPTER 06 –CONCLUSION

This chapter will be concluded the all final details of project. And this contains the lessons we learnt and information about future developments of the system.

CHAPTER 2: ANALYSIS

2.1. INTRODUCTION

This stage, all subject areas related to the project have been discussed and analysed and the functional and non-functional requirements will be identified which are helpful for designing the system. Requirement identifying and analysis is a very important stage and therefore it should be done neatly.

2.2. FACT GATHERING TECHNIQUES

There were several techniques used for fact gathering from different stakeholder perspectives. Frequently used fact gathering methods are,

- Interviews
- Document reviews
- Observations

Both open ended and close ended interviews were conducted with company administration and employees and it was helpful to capture new problems which they have to face.

Reviewing of currently used excel sheets, user manuals of machines, and documents with the details gathered from above interviews helped to get a good idea and a better understanding about the current work process.

And finally all of those details helped to observe a better picture of working process of the company.

2.3. ANALSING THE CURRENT SYSTEM

I carried out some analysing parts through the facts gathered from certain fact gathering techniques from company administrations and employment. Current procedures which carried out at Hope Diagnostics are done at three ends.

1. Doctor's End
2. Patient End
3. Company End

Current working process can be shown in a use case diagram like below. (Fig 2.1).

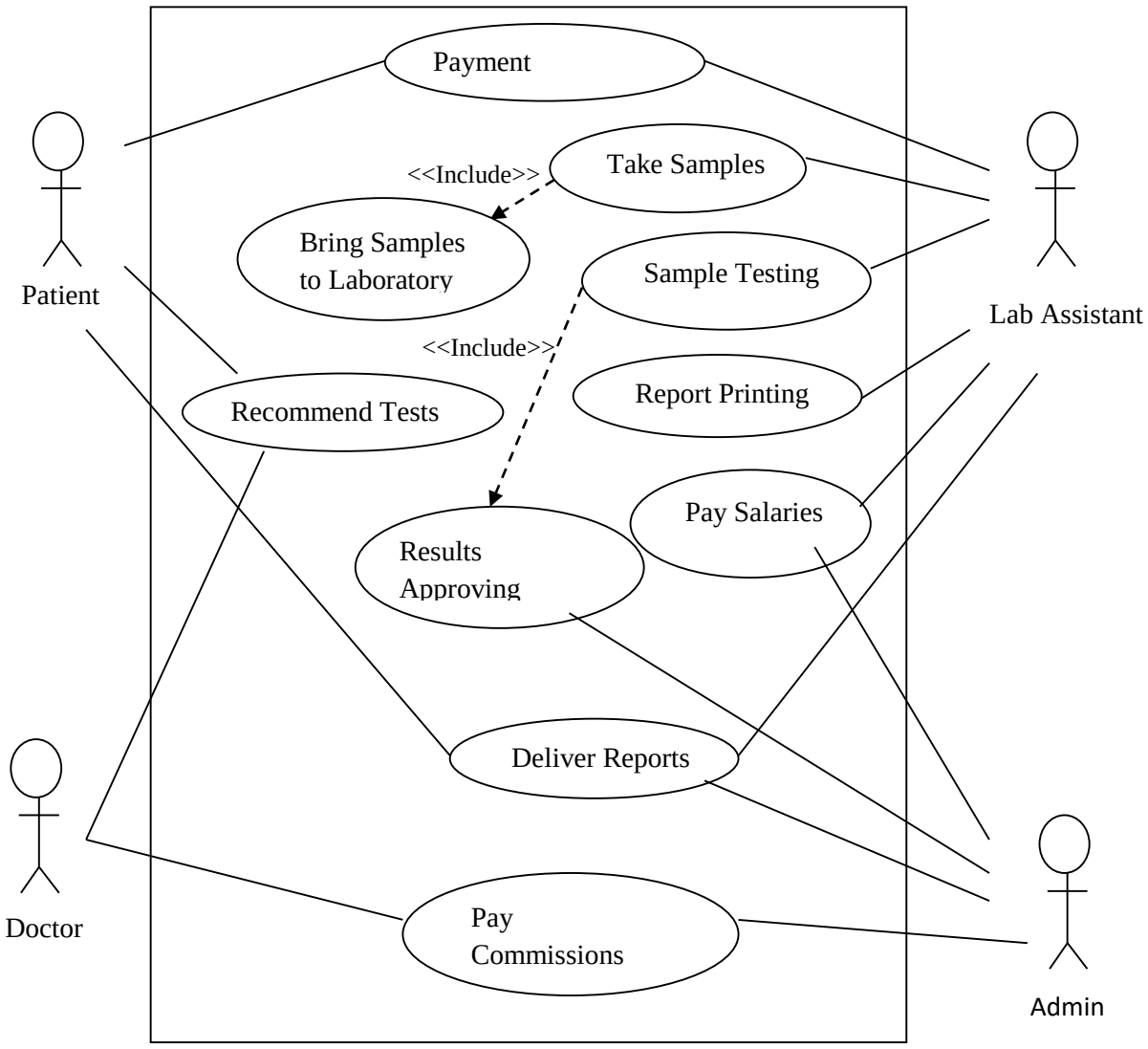


Fig 2.1: Use case diagram for existing system

2.3.1 Doctor's End

Doctors examine their patients and recommend necessary laboratory tests for their patients. And a considerable commission will be payed to doctors by company end to encourage them for a recommendation of Hope Diagnostics to their patients

2.3.2 Patient End

With or without doctor recommendations, patients come to Hope Diagnostics or a sample collecting centre, pay a fair price and do their medical laboratory tests. And patients will have their test results as a printed paper as soon as possible.

2.3.3 Company End

Hope Diagnostics is a laboratory which has enough tools, equipment and well trained employees to provide better service to the community. Currently it has nine centres including head branch to collect samples. All samples are brought to the head branch by lab assistants. Then all samples are tested by well-trained lab assistants by using standard equipment as well as microscope. So each sample is tested twice and gets the most appropriate results. Then all results are sent to administration and they will be approved by the Medical Laboratory Technician (MLT) and send back for report printing. After report printing, each report is delivered to relevant centre by relevant lab assistants or administration.

Payment transactions with patients are carried out by each lab assistant at the relevant centre and all money will be brought to administration at the end of the day. Salaries for employees are payed to them at the end of each month. And a considerable commission is payed to each doctor who works where the centres are established. And establishments' rentals, payments for reagent and other payments are carried out by administration at the end of each month.

Current MS excel sheet format for report printing and system user interface for payment transactions are showed in below. (Fig 2.2).

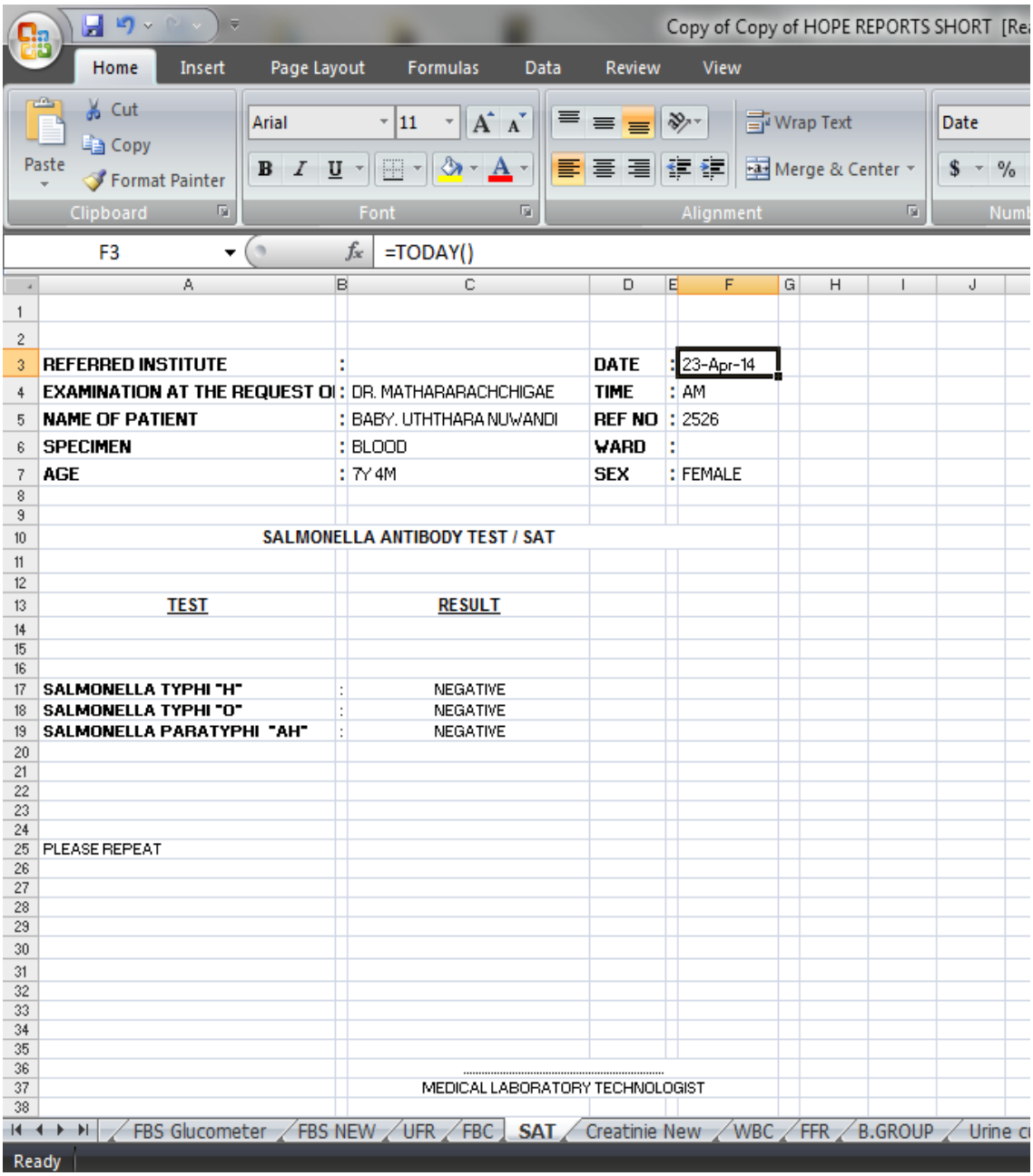


Fig 2.2: Current MS excel sheet format for report printing

Following figure 2.3 illustrates the user interface of the current system for payment transactions.

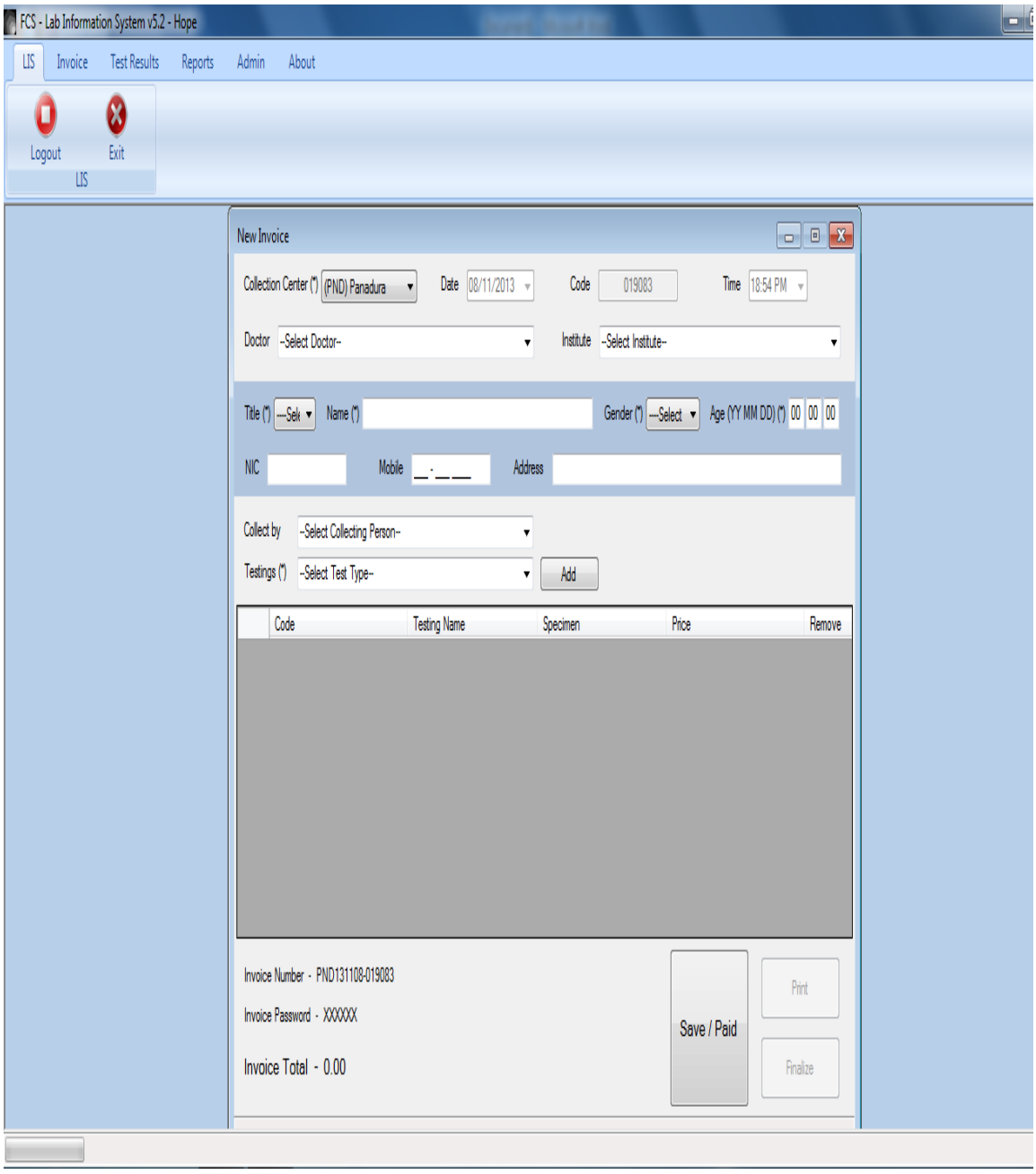


Fig 2.3: Current system user interface for payment transactions

2.4. FUNCTIONAL REQUIREMENTS

I have recognized the functions those should be performed by the proposed Web Based Report and Invoice Generating System for Hope Diagnostics Medical Laboratory Service. Following are the functional requirements those I recognized.

Company End

- Automating data transferring between invoice generating and report printing.
- Automating data processing and calculations

Patient End

- Keeping separate login accounts for each regular patient to keep their previous reports.
- Facilitate patients to be notified when their reports are ready and to be able to view online.
- Automating calculation to get cost estimation
- Facilitate patients to channel doctors who are working at the Medical Centre where the head branch established.

Doctors End

- Keeping separate login accounts for each registered doctors to view patient's results
- Automating analysing of particular patient's results.

2.5. NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements are explaining the important properties which the proposed system should have to perform with maximum efficiency for its users. These requirements may affect at the whole system. So failing to accomplish non-functional requirements may make whole system unsuccessful. The key non-functional requirements which the system should meet are listed below.

- **Security and privacy**

  Medical laboratory reports of a person are highly private and confidential documents. Therefore, those kinds of documents should be able to see only authorized people. And the system handles important data of company business processes and patients. So the system should have highly secured and optimum privacy.

- **Accuracy and consistency**

  This is another key non-functional requirement of the system. The purpose of automating is to increase accuracy and consistency. Therefore, the system should be accurate and consistent when the data manipulating.

- **Reliability**

  The system should be able to execute properly at any circumstance. The unexpected errors or mistakes cannot affect at the process of system and the results.

- **Availability**

  The system should be available for all users (company, doctors, and patients) at any time.

- **User friendliness**

  The current employment and the considerable amount of patients are not too much familiar with working in a computerized environment. Therefore, providing easiness of accessing is a major requirement of the system.

2.6 EXISTING SIMILAR SYSTEMS

It couldn't find exactly similar type of systems for the proposed web solution in the web. But it is able to find some system similar to some part of the proposed system.

E-health care is a medical lab reporting system for managing patient details, generate testing reports, and a SMS management system. It had few unnecessary function as well as it is lack of few required functions of the proposed web based system. Figure 2.4 and figure 2.5 show the user interfaces of E-health care lab reporting system.

The Inventory is a simple inventory management system for inventory management of a simple stock. Client needed a simple inventory management system to be attached to the proposed system for his store keeping. Figure 2.6 is a simple user interface of "The Inventory" system.

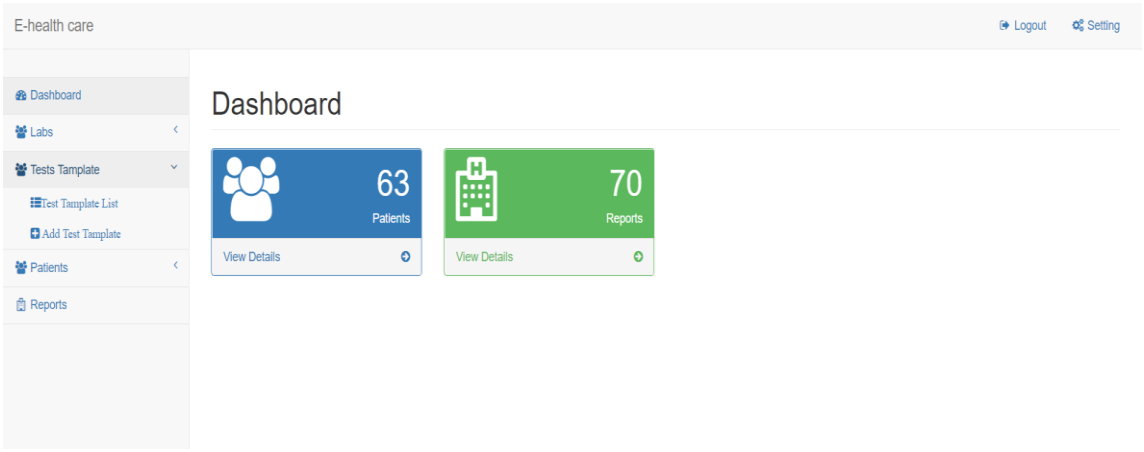


Fig 2.4: Interfaces of E-health care Medical Laboratory System

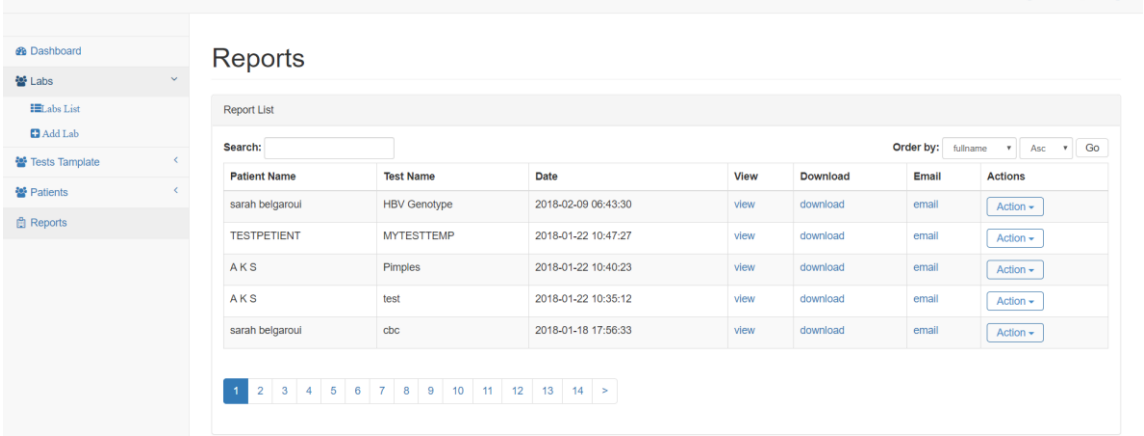


Fig 2.5: Interfaces of E-health care Medical Laboratory System

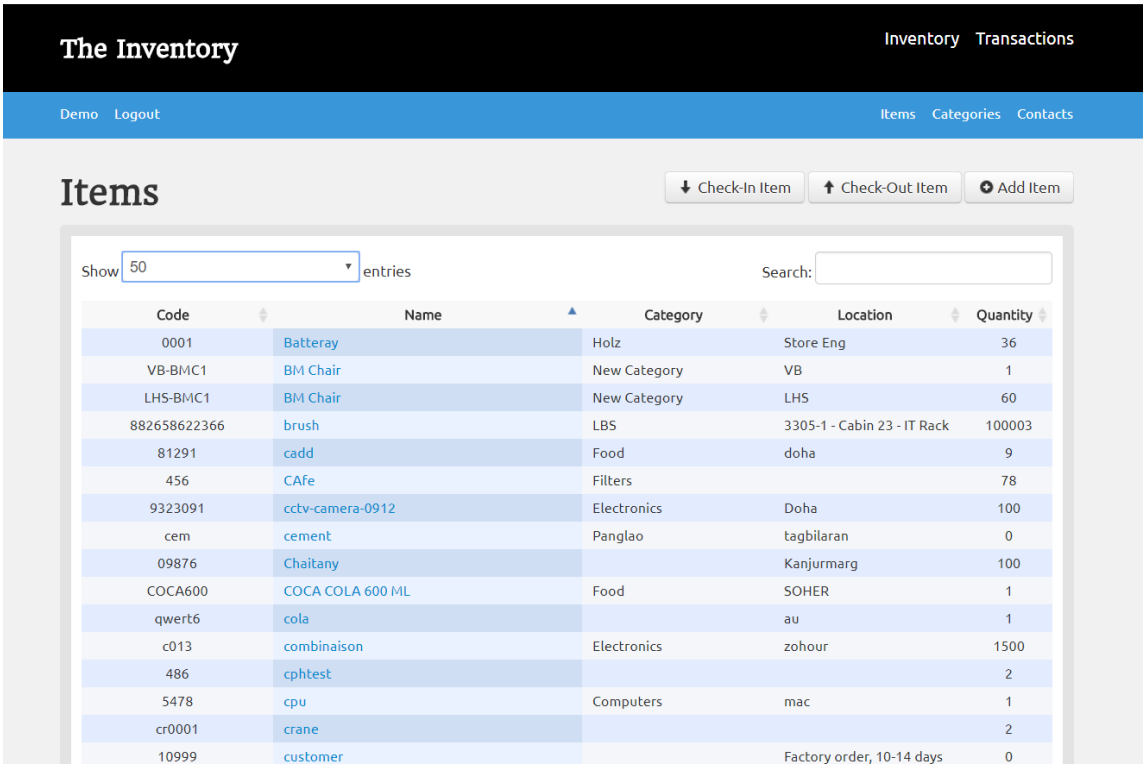


Fig 2.6: Interface of "The Inventory" inventory management System

2.7. METHODOLOGY FOR THE PROPOSED SYSTEM

Proposed system is developed using Rational Unified Process(RUP) [9] because RUP is an iterative software development process framework. Even though client has some initial requirements, it seemed to add some function to the system later. And system has to be developed as separated modules, payment transactions, result report printing and administrative backend, inventory management, patients end. All these modules have been developed iteratively. Therefore, RUP is appropriate for this system as the SDLC methodology.

There are four main phases of RUP lifecycle. Figure 2.7 shows the life cycle parts according to the phases.

- Inception Phase

  In this phase, system will be scoped based on initial costing and budget.

- Elaboration Phase

  This phase is where the project starts to take shape. Problem domain analysis is made and the architecture of the project gets its basic form in this phase.

- Construction Phase

  Functions and features of the system are developed in this phase. System is built in this phase.

- Transition Phase

  Developed system is delivered to its end users will be done in this phase. The product is also checked against the quality level set in the inception phase.

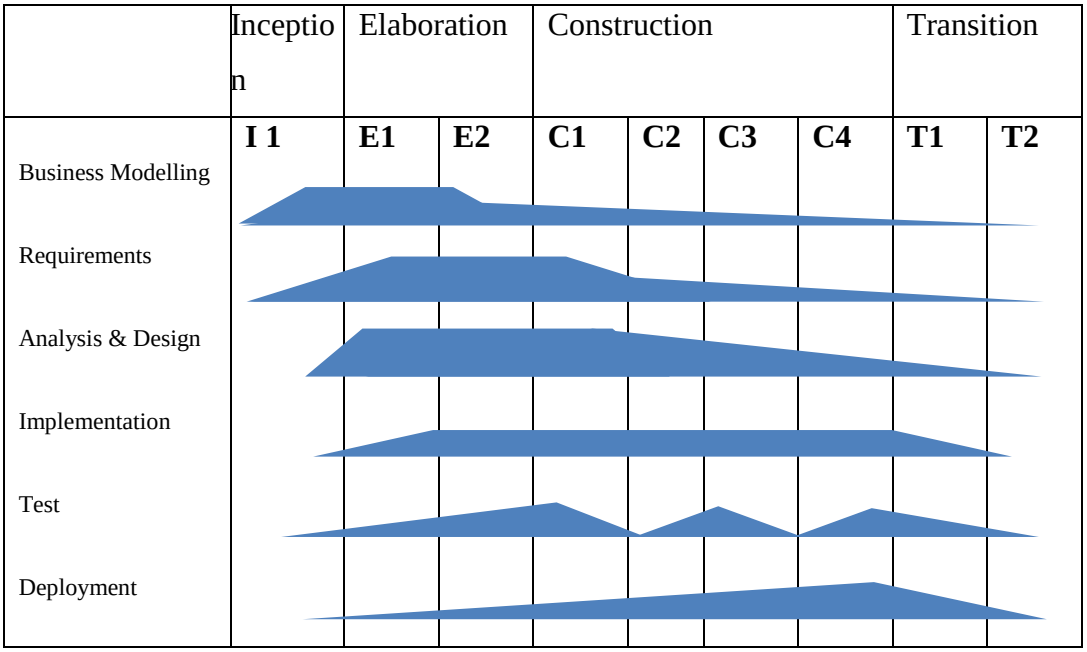


Fig 2.7: RUP Methodology Diagram for The Proposed System

CHAPTER 3: DESIGN

3.1. INTRODUCTION

After the requirements are gathered in the analysing stage, next stage is gathered requirement turn in to the exact system. At the designing stage, the system is constructing imaginary using designing techniques for developing the system. This design depicts a high level view of the system.

3.2. ALTERNATE SOLUTIONS TO THE SYSTEM

Following mentioned alternate solutions can be used for the proposed system.

- Standalone System
  As simplifying the transactions & result reports generating is the main target of this system, we could have proposed a standalone system for the client. But the client needed to facilitate his patients via online and he had a view of some future enhancements, therefore a standalone system wasn't selected.
- Implementing the proposed system in a Private network.
  Although this method has some security wise advantages, since we have to deal with public, it was not suitable to use a private network.

3.3. REASONS TO SELECT A WEB BASED SYSTEM

Following is why a web based solution is selected for the proposed system.

- Mainly client needed to facilitate general public to view their test results online and facilitate regular patients to keep their results records individually. Therefore, it is decided a web based solution to be developed.
- Through a web based solution, client can be facilitated to access his systems from anywhere without coming to the office. Therefore, a system user can do particular tasks via online.
- It is not needed an additional cost to implement a web based system since existing infrastructure can be used.

- Since web based systems are platform independent, users can use any platform to access the system.
- Since we can access real time updates through a web based systems, doctors can login to the system and check history and present situations of patients individually.

3.4. OBJECT ORIENTED DESIGN USING THE UML

Object-oriented design [1] is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design [1]. Object oriented designing involve designing object classes and relationships between these classes. The objects in the system and their interaction are defined by those classes.

3.4.1. USE CASE DIAGRAM

A use case diagram [2] at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

The high level use case diagram [2] for the proposed Web Based Report and Invoice Generating System for Hope Diagnostics Medical Laboratory Service is shown in following figure 3.1. Detailed use case diagrams of some modules are mentioned in APPENDIX B.

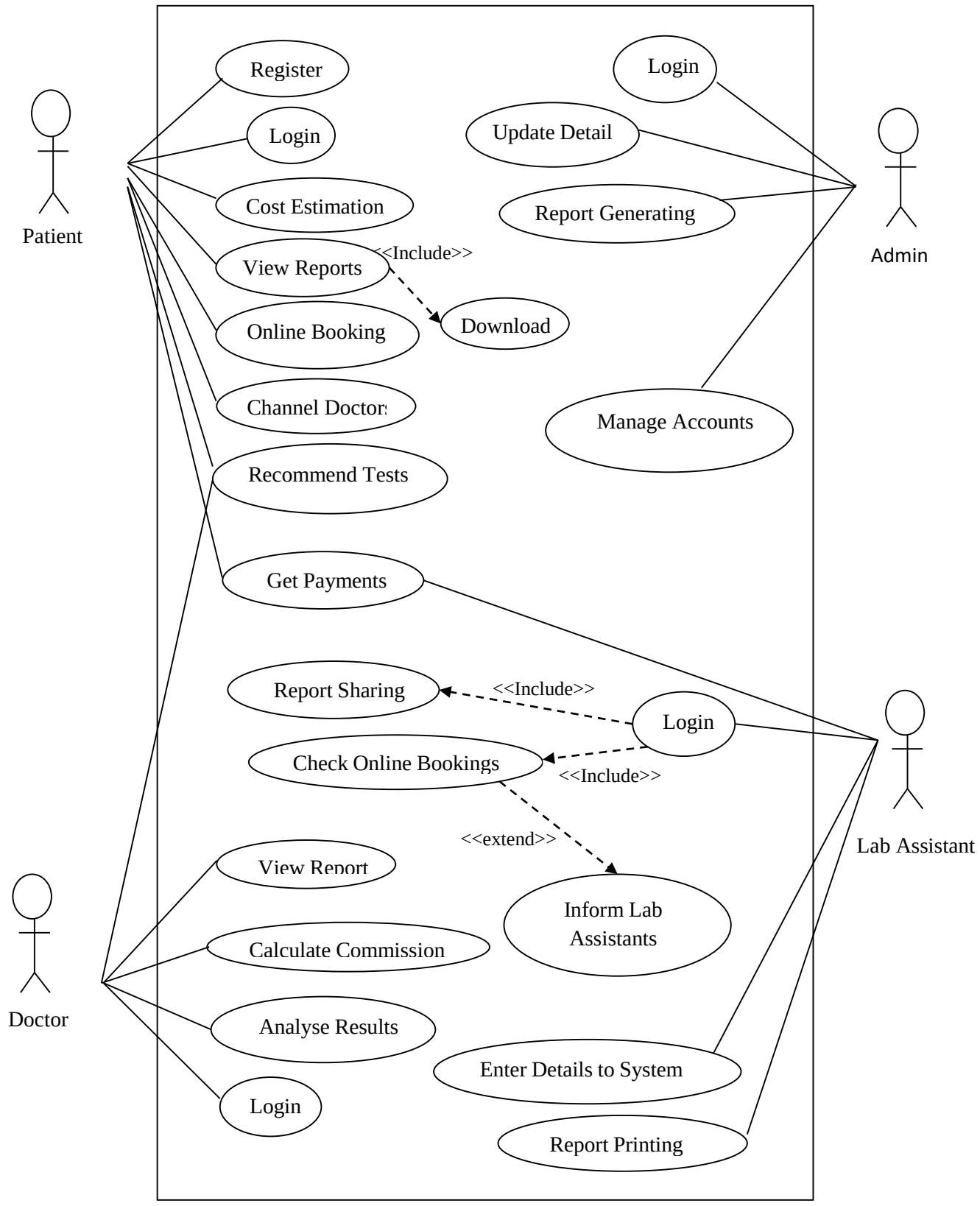


Fig 3.1: Use Case Diagram for Web Based Report and Invoice Generating System

3.4.2. CLASS DIAGRAM

In software engineering, a class diagram [3] in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. [3] Figure 3.2 demonstrates the class diagram for proposed system.

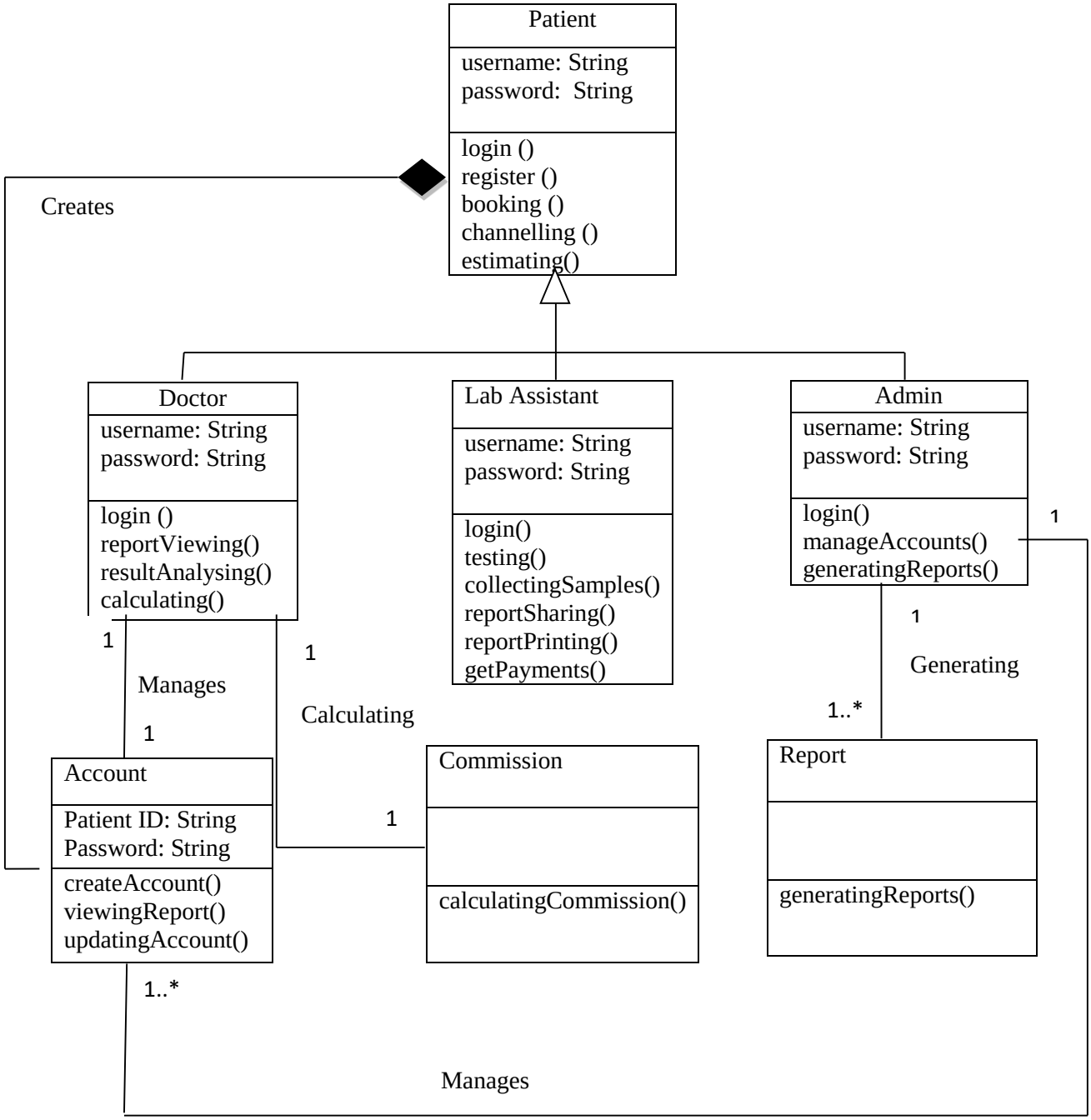


Fig 3.2: Class diagram for Web Based Report and Invoice Generating System

3.4.3 SEQUENCE DIAGRAM

Sequence diagrams illustrate the interaction among classes in terms of exchanging the messages over time. It is a good method to validate and explain various runtime scenarios. This can be used to show how a system will function and to obtain responsibilities a class may need to have within the process of modelling the new system. Sequence diagrams are also called as event diagrams.

Sequence diagram for the user access granting and login module. (Figure 3.3)

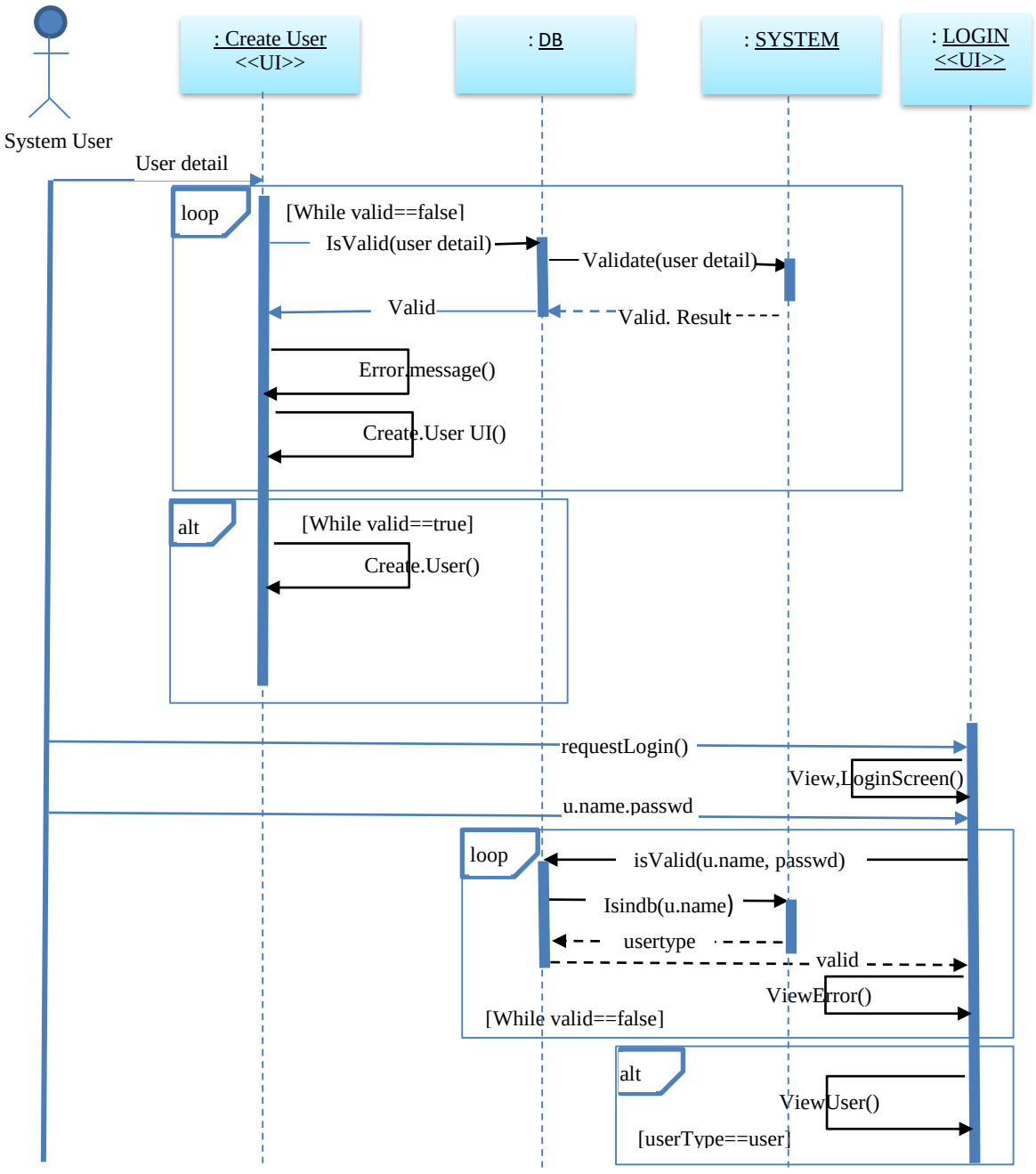


Fig 3.3: Sequence diagram for Login & Add user module of System

Sequence diagram of Payment transaction and Test results printing (Figure 3.4).

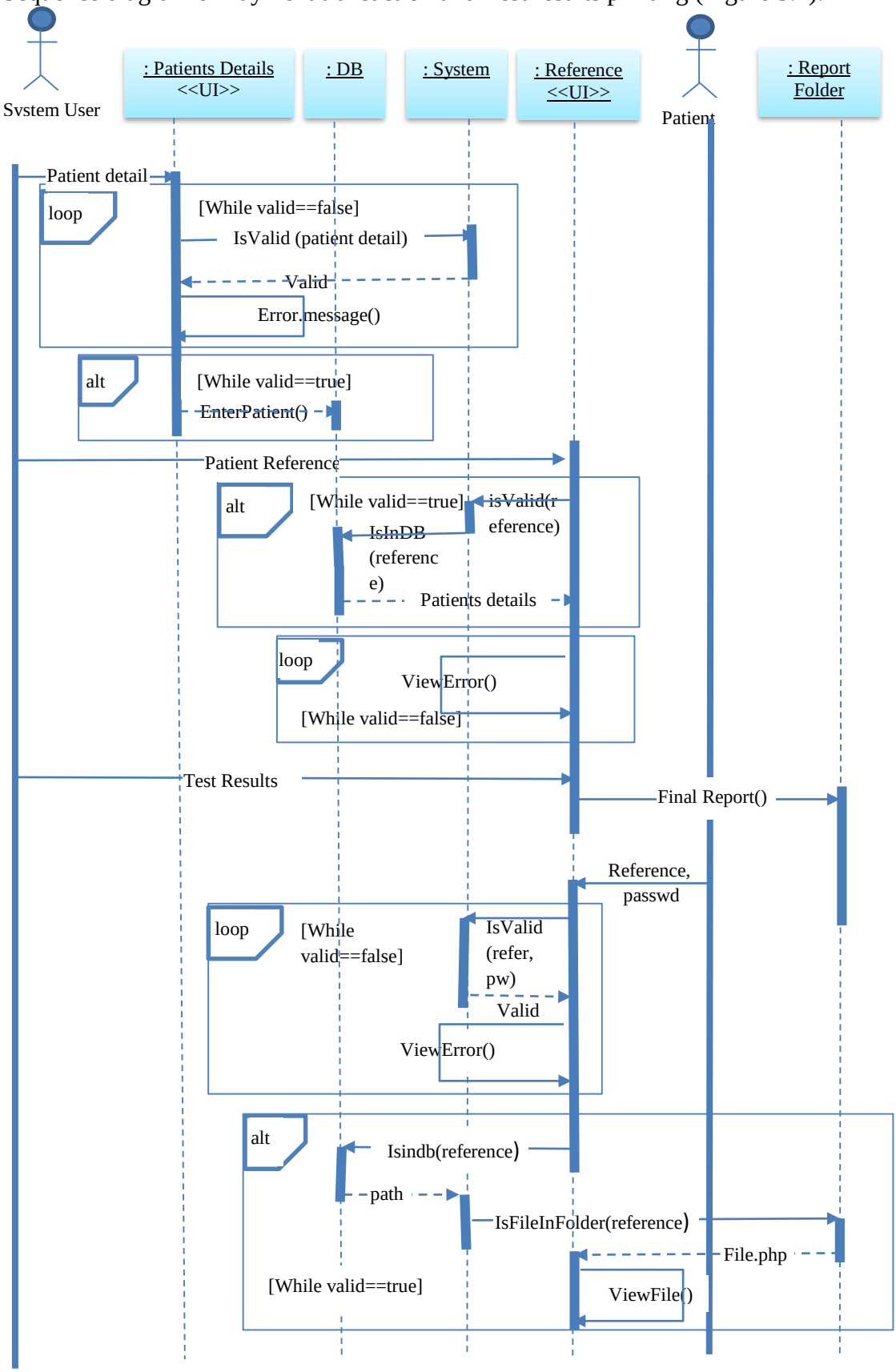


Fig 3.4: Sequence diagram for Payment Transaction, Results Printing & online Report Viewing modules of the System

3.4.4 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. Following figure 3.5 illustrate the activity diagram of proposed system.

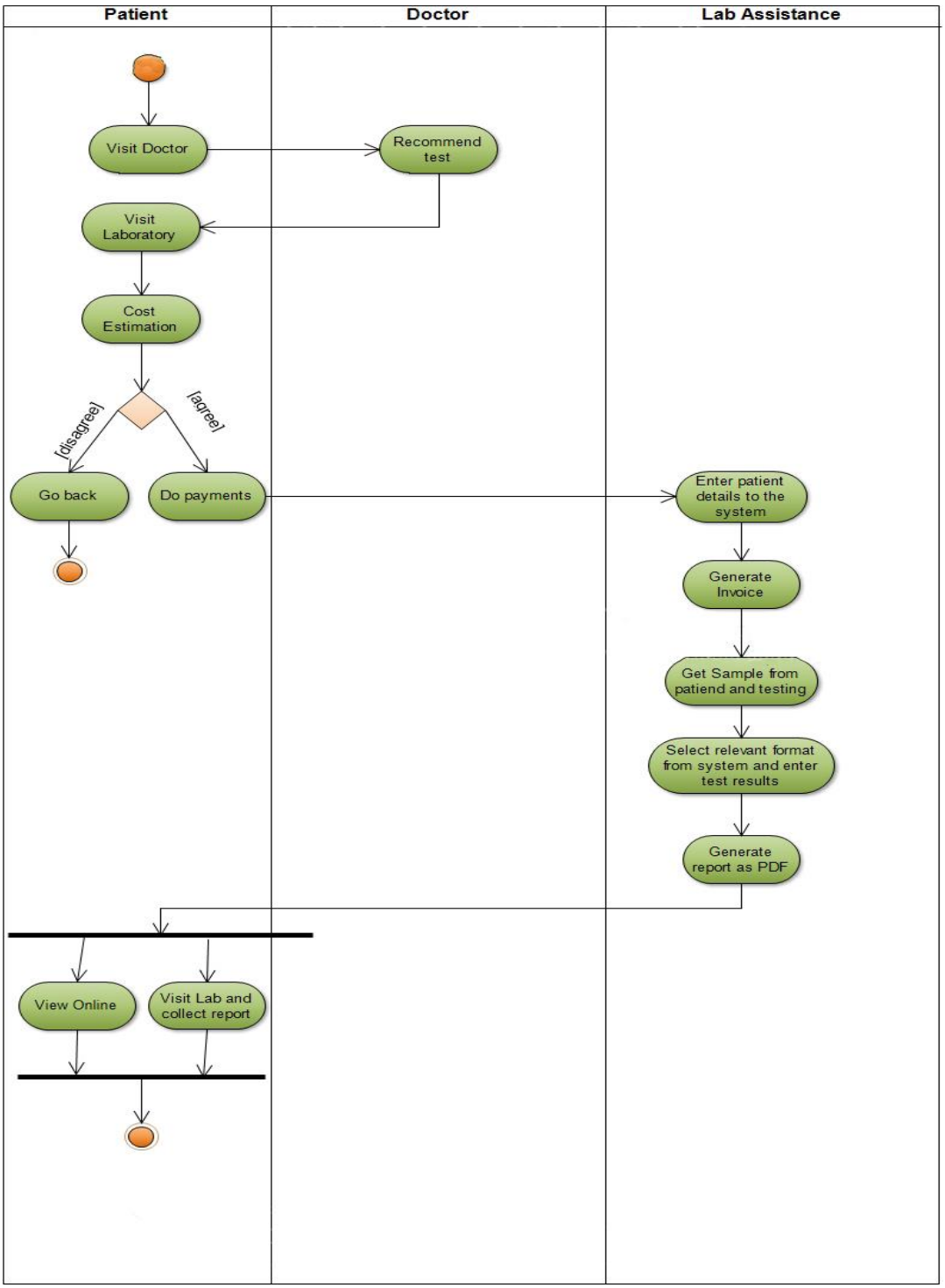


Fig 3.5: Activity Diagram for Web Based Report and Invoice Generating System

3.5. DATABASE DESIGNING

Database design [4] is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design [4] can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design [4] could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS). [4]

Following figure 3.6 is the Entity Relationship Diagram (ERD) for the proposed web based system.

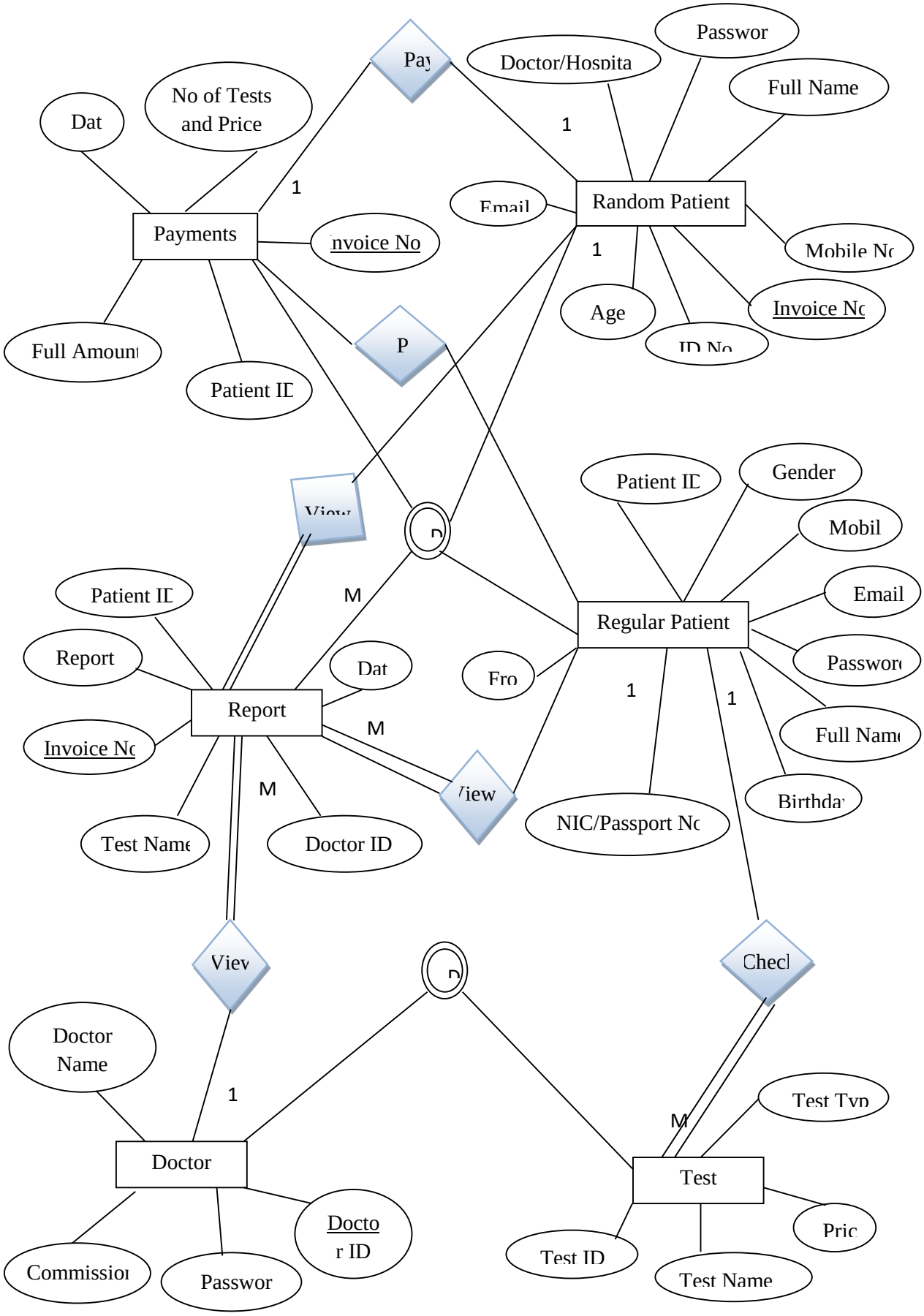


Figure 3.6: ER Diagram for Web Based Report and Invoice Generating System

3.6. USER INTERFACE DESIGNING

User interface design [5] (UID) or user interface engineering is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction. The goal of user interface design [5] is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals what is often called user-centred design.

Good user interface design [5] facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

When designing a user interface for a web based public support system, there are some critical facts we have to consider. Due to the lack of IT knowledge of majority of public users we have to maintain these facts in a proper level.

- Simple User Interface.

  Maintaining a clean and clear user interface will be helpful to every user to carry out their task easily.

- Proper Navigation

  Proper and easy navigation between every page is important to access system easily.

- Familiarity

  Developing interfaces, terminology, and flow in a familiar way to the users is important.

3.6.1. HOME PAGE

Fig 3.7.: Home page of the web site

A simple design is used for Home page and the rest pages. It will help to increase user friendliness. And to reduce the loading time, the pages are linked in the home page. (Fig 3.7)

3.6.2. ABOUT US PAGE

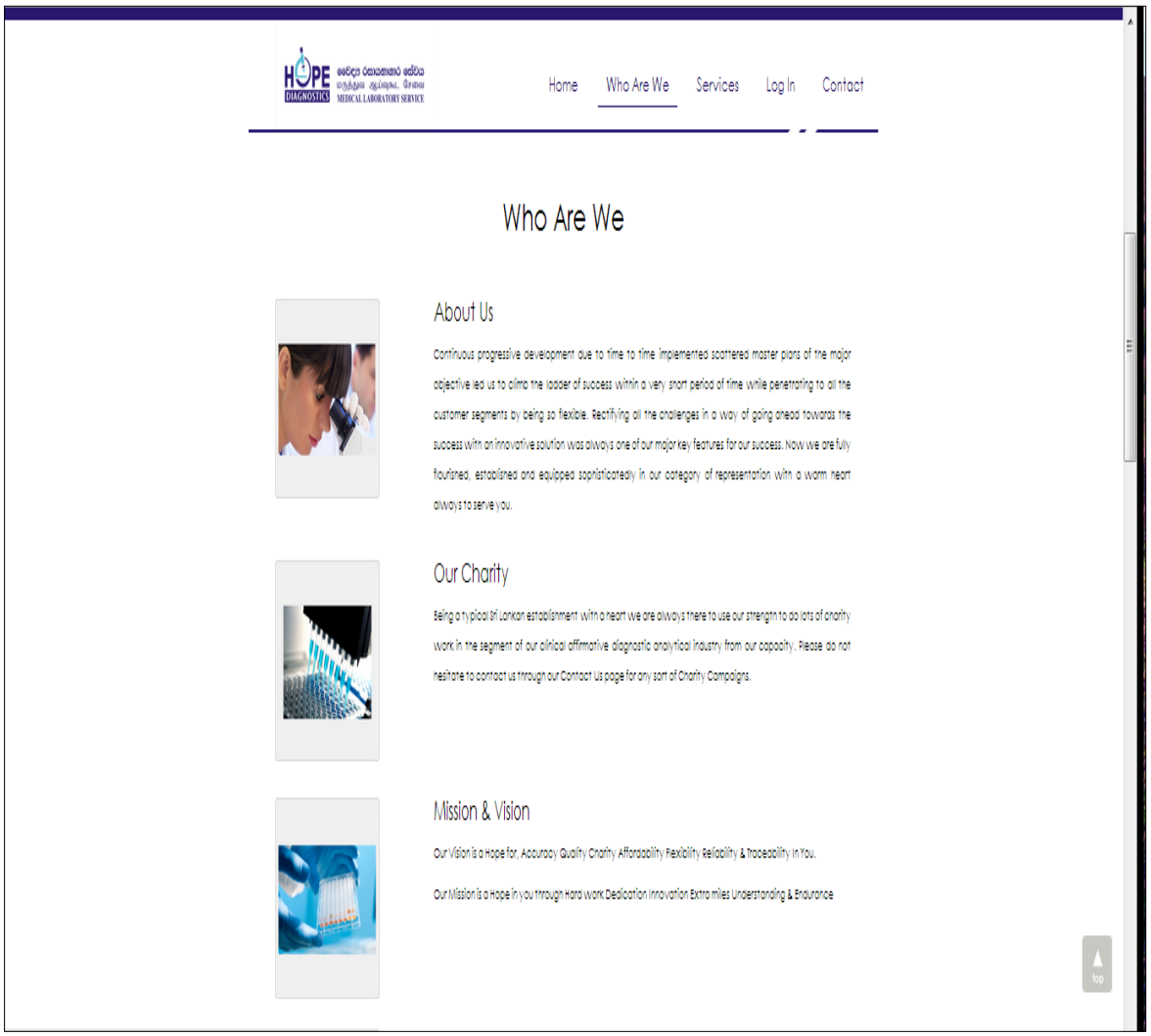


Fig 3.8: About Us page of the web site

Client wanted to have an overlook of his laboratory services in his web site. Therefore, a little description of the laboratory and their services is put in the about us page. (Fig 3.8).

3.6.3. LOGIN AND SIGN UP PAGE

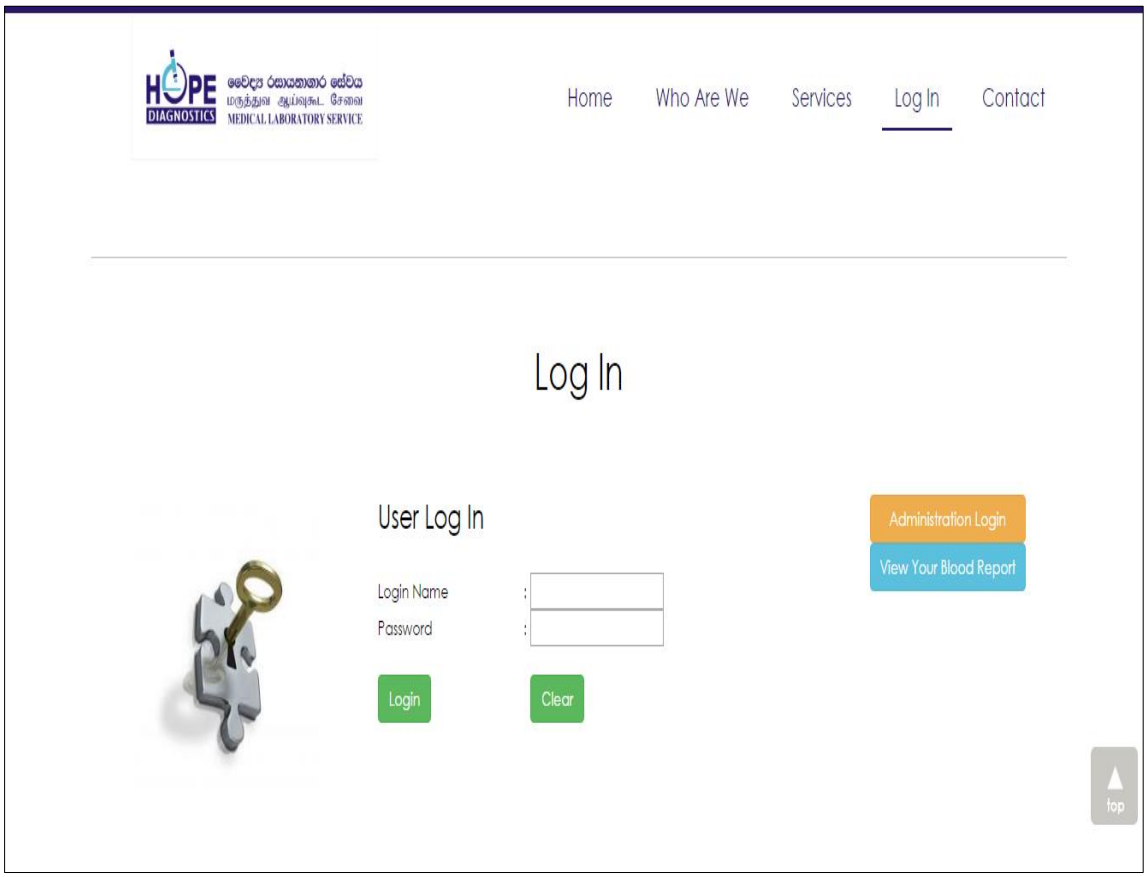


Fig 3.9: Login Interface for patients & Login terminal for backend

Figure 3.9 shows login interface for patients and the login terminal for the administrative backend. Following figures will describe the administrative backend.

3.7. BACKEND LOGIN

3.7.1. STAFF LOGIN INTERFACE

The figure 3.10 illustrates the login interface for admin and staff. Staff members have to login in to the system using their user names and passwords for carrying out any task of backend. Error messages regarding user login attempts will be shown by pop up messages.

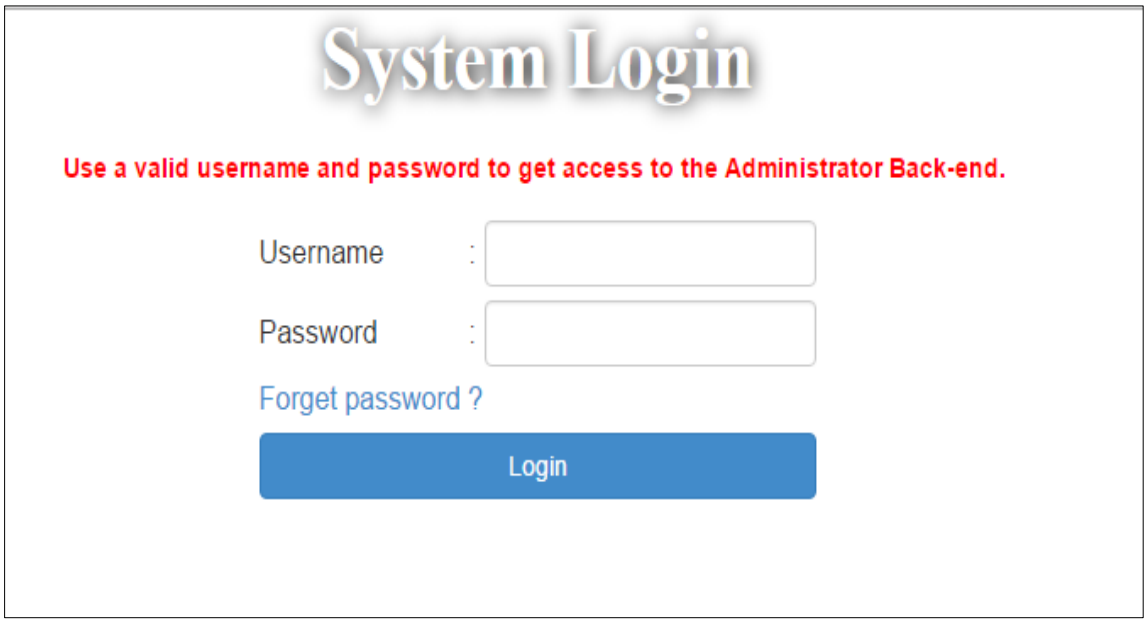


Fig 3.10: Backend login interface

And also authorised doctors can use this interface to login their account and view the test results reports of their patients.

3.7.2. ADMINISTRATOR HOME PAGE

The backend home page user interface for system administrators has been shown below figure 3.11. It has been customized according to the client requirements to facilitate admin to do all admin tasks. I was free to decide the interface design therefore I used this template for the interface.

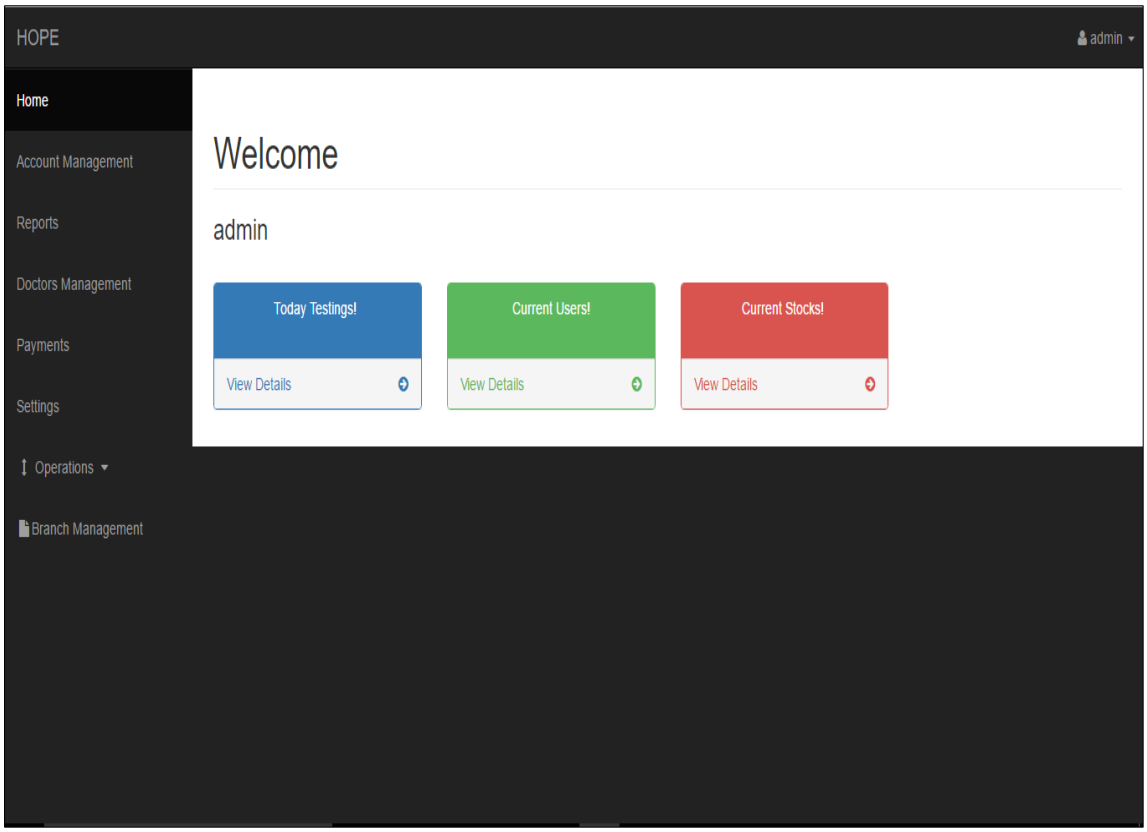


Fig 3.11: Home page of Admin account

The navigation bars in left side of the page will facilitate admin to do all the administrative and other tasks while some status relation to the lab will be showing middle of the page.

3.7.3 INTERFACE FOR ADD USERS

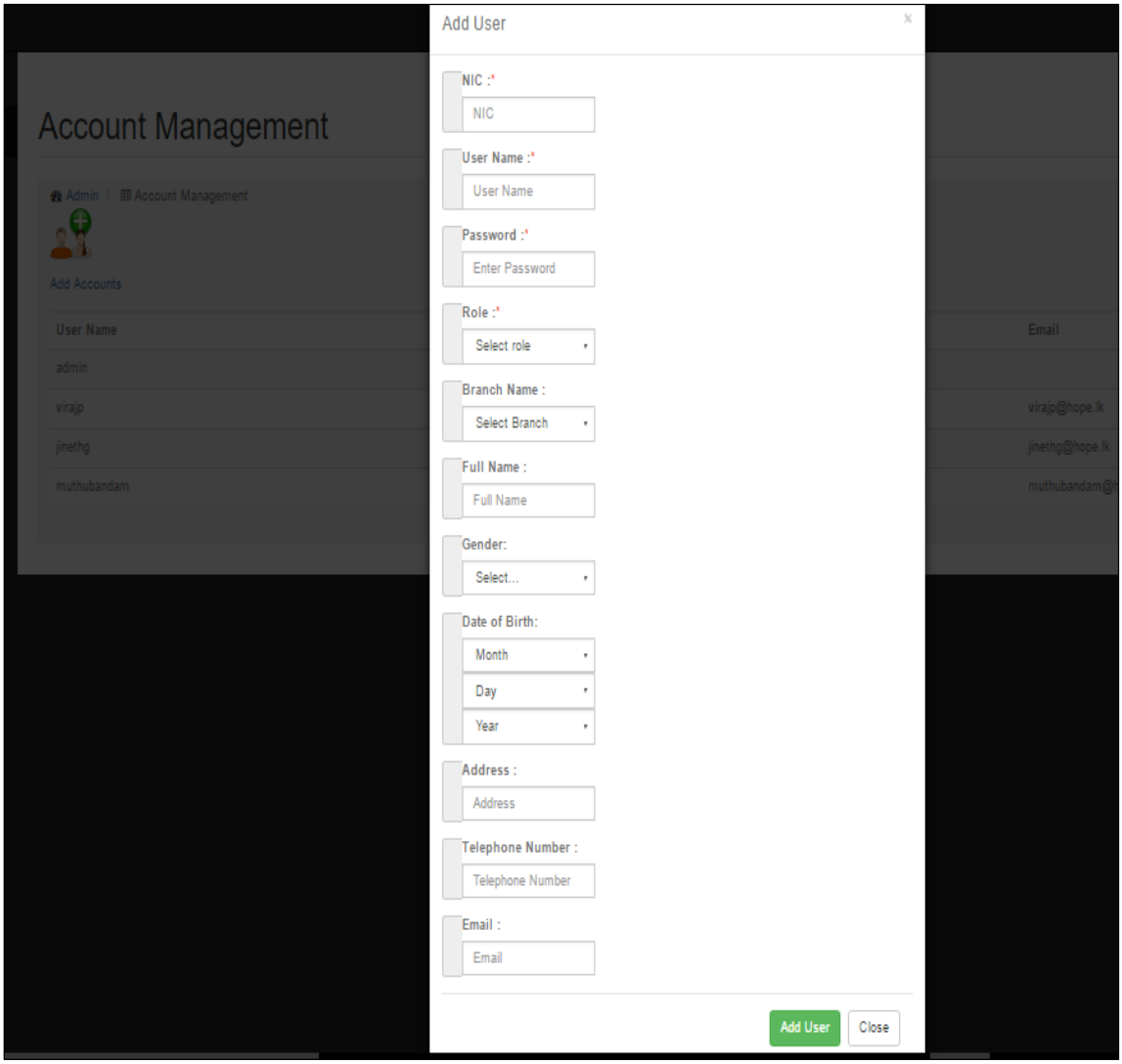


Fig 3.12: Add user interface

This interface (Fig 3.12) is used to add a system user. There are four types of users have been introduced to the system. They are admin users, senior employee, employee and doctors.

3.7.4 INTERFACE FOR SHOW, DELETE AND UPDATE USERS

Following figure 3.13 illustrate the interface of viewing current user accounts of backend system. From this interface administrator is allowed to edit delete the user details.

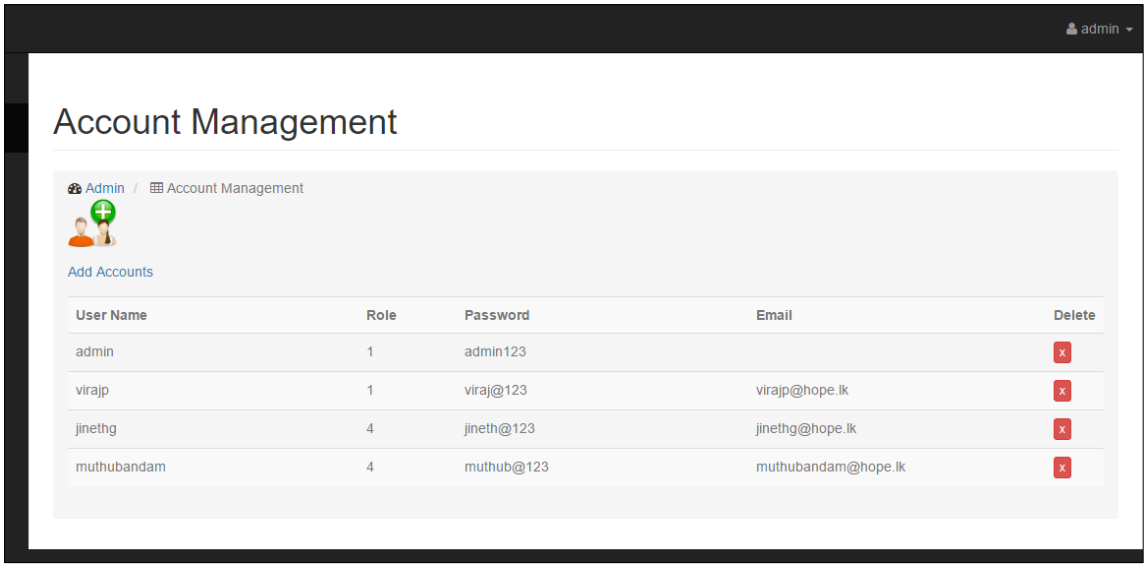


Fig 3.13: Accounts Management Page

3.7.5 INTERFACE FOR REPORT PRINTING PROCESS

Following figure 3.14 shows the option to retrieve data of patients from payment table to form of report printing.

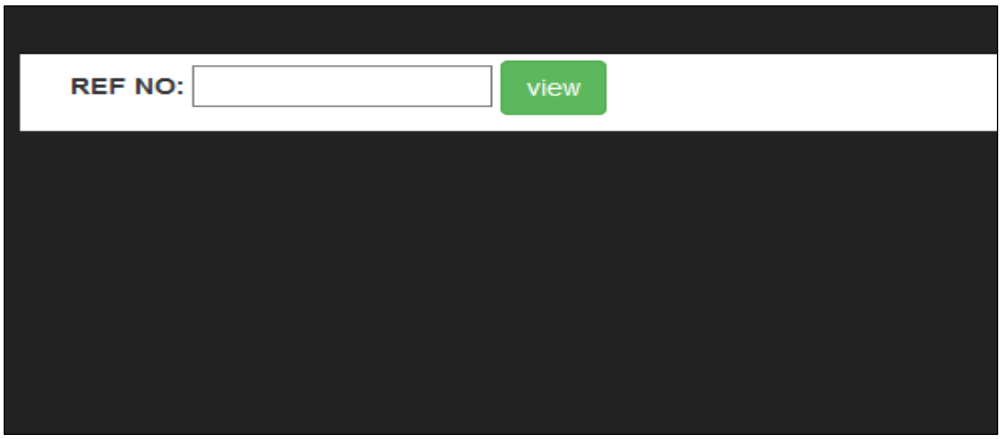


Fig 3.14: Data Retrieving Page

Page will be displayed as below figure 3.15 after data retrieved from the payment table.

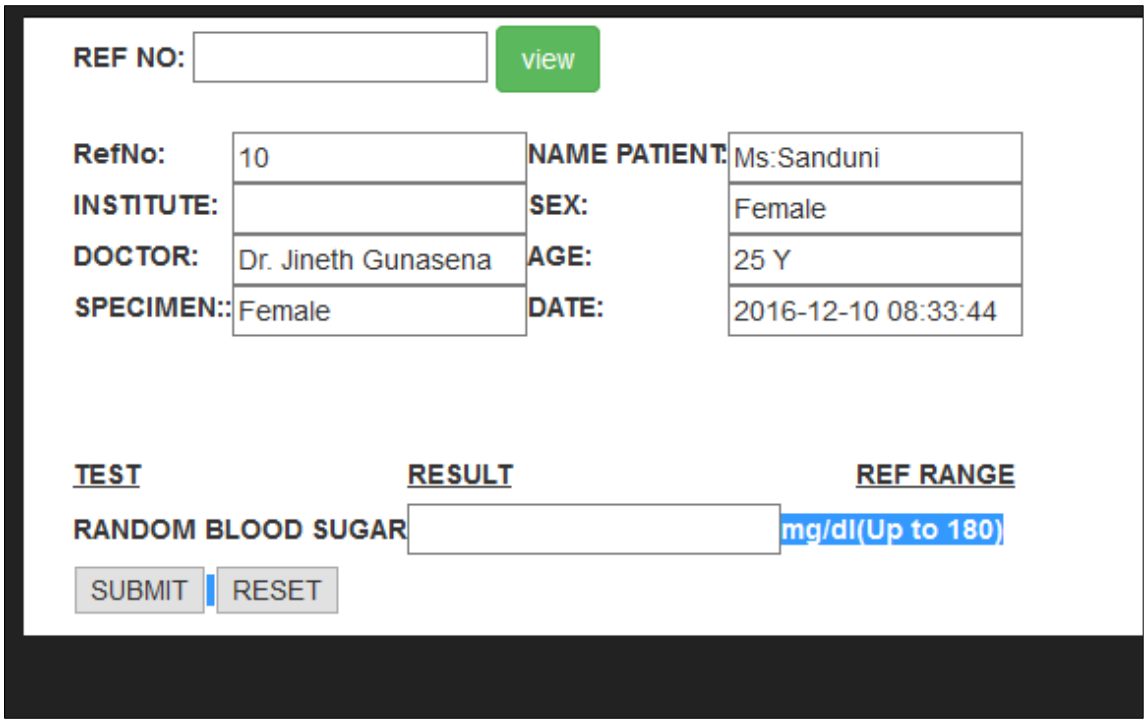
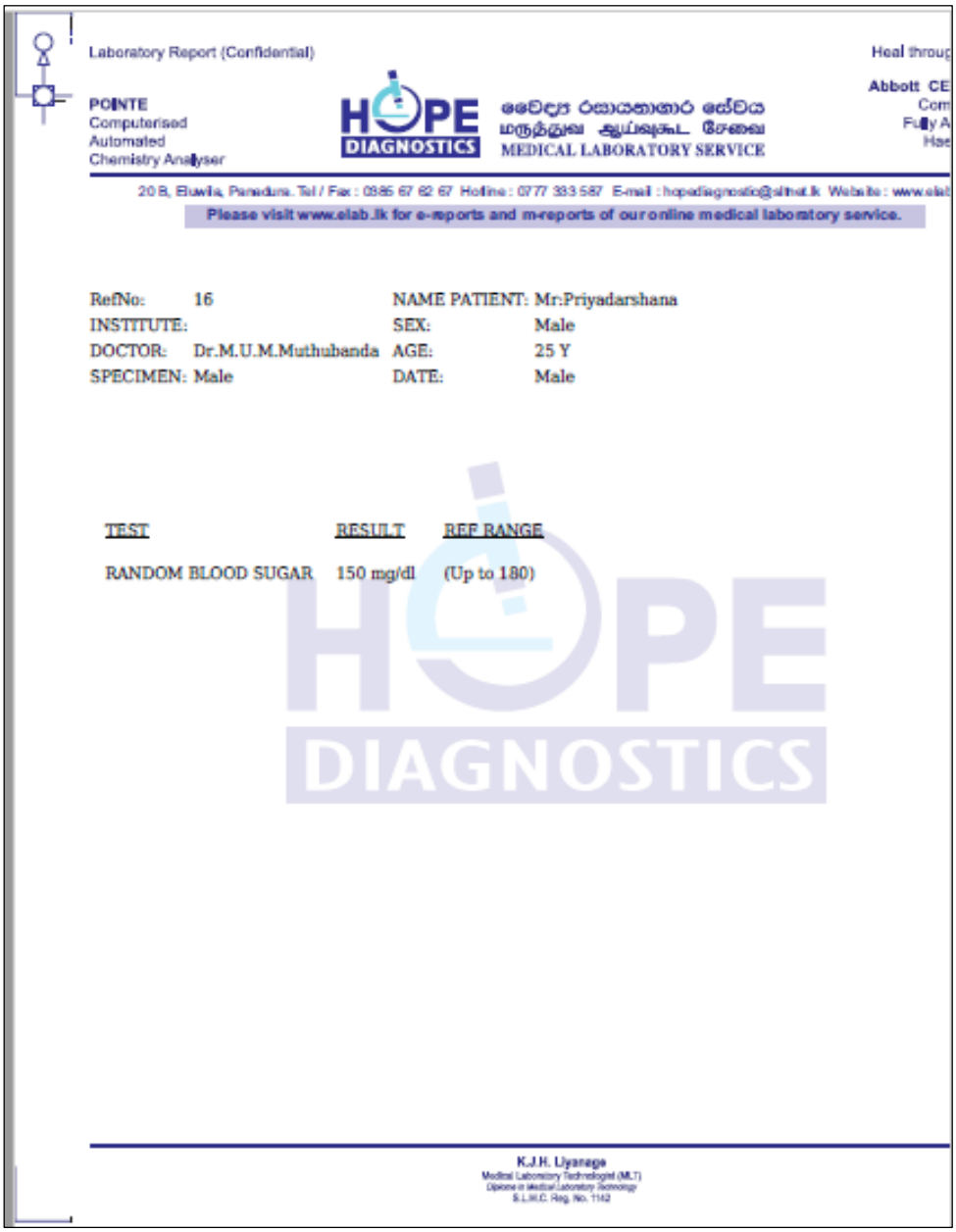


Fig 3.15: After Data Retrieving Page

After the data are submitted, all data will pass to an intermediate page and give us preview of the report. Then we can get the report as a pdf document and it will be displayed as below figure 3.16.



Laboratory Report (Confidential)

Heal throug

POINTE
Computerised
Automated
Chemistry Analyser

HOPE DIAGNOSTICS

වෛද්‍ය රසායනාගාර සේවය
மருத்துவ ஆய்வுகூட சேவை
MEDICAL LABORATORY SERVICE

Abbott CE
Com
Fully A
Has

20 B, Eluwila, Panadura. Tel / Fax : 0385 67 62 67 Hotline : 0777 333 587 E-mail : hopediagnostic@slt.net.lk Website : www.elab

Please visit www.elab.lk for e-reports and m-reports of our online medical laboratory service.

RefNo:	16	NAME PATIENT:	Mr:Priyadarshana
INSTITUTE:		SEX:	Male
DOCTOR:	Dr.M.U.M.Muthubanda	AGE:	25 Y
SPECIMEN:	Male	DATE:	Male

TEST	RESULT	REF RANGE
RANDOM BLOOD SUGAR	150 mg/dl	(Up to 180)

HOPE DIAGNOSTICS

K.J.H. Liyanage
Medical Laboratory Technologist (MLT)
Diploma in Medical Laboratory Technology
S.L.M.C. Reg. No. 1142

Fig 3.16: Result PDF Document Page

3.7.6 INTERFACE FOR INVOICE GENERATING PROCESS

Following figure 3.17 is illustrating the interface for entering patients details before generate an invoice. All the required fields are validated by using jQuery and PHP

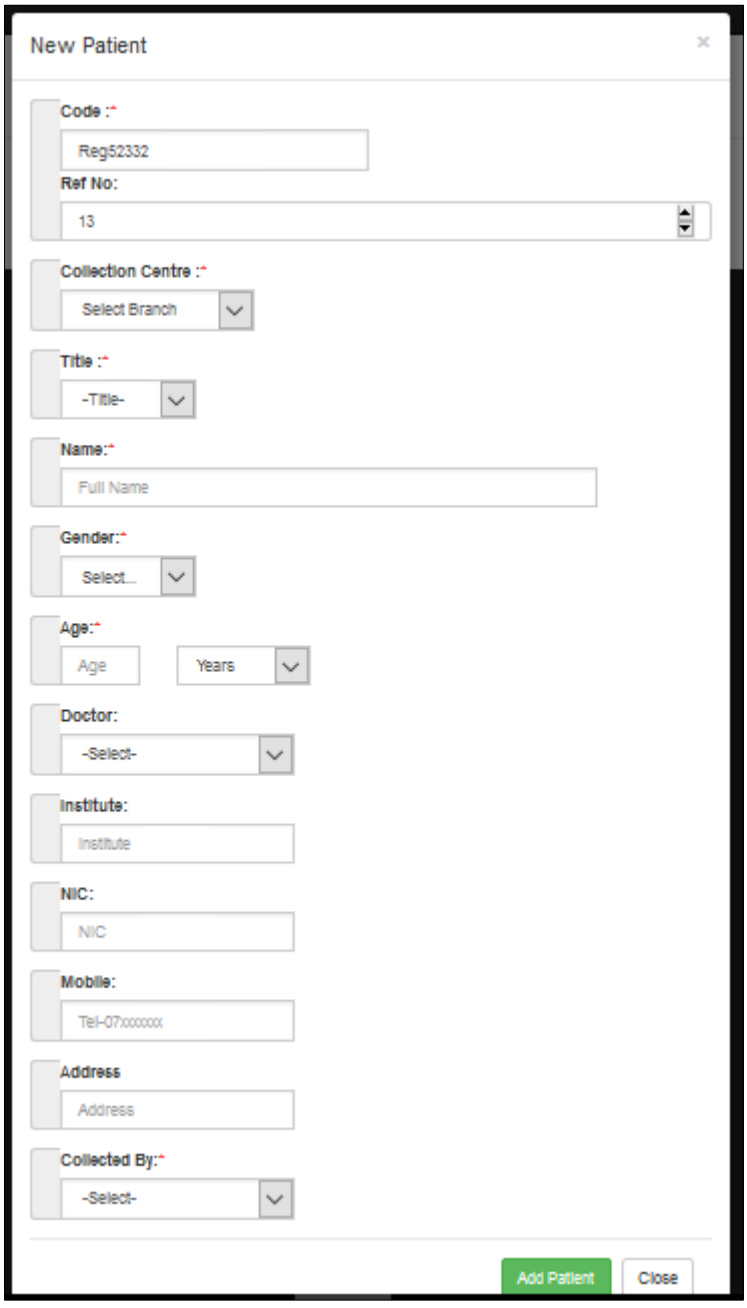


Fig 3.17: Adding New Patient Details Page

Figure 3.18 is the check box to select relevant testings at the payment. Sometimes it can be only one testing and sometimes it can be multiple testings.

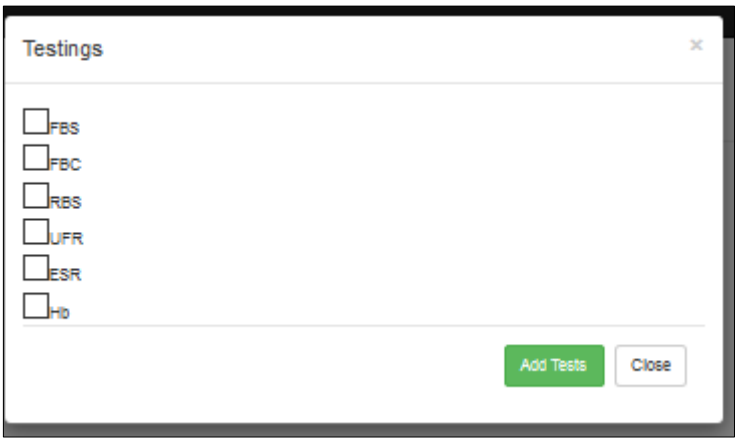


Fig 3.18: Adding Testing with Patient

After all the selected testings are added, the total amount will be calculated and displayed with the details of the patient as below figure 3.19.

Fig 3.19: After Generating an Invoice

3.7.7 INTERFACE FOR STOCKS MANAGEMENT PROCESS

Following figure 3.20 is the user interface design of the stock management. A simple interface is used for the stock management since the client has a small store of required items of laboratory.

Fig 3.20: Stocks Management Page

CHAPTER 04-IMPLEMENTATION

Software implementation encompasses all the post-sale processes involved in something operating properly in its environment, including analysing requirements, installation, configuration, customization, running, testing, systems integrations, user training, and delivery and making necessary changes. The word "deployment" is sometimes used to mean the same thing. [6]

4.1. SOFTWARE ENVIRONMENT

4.1.1. LANGUAGES AND TECHNOLOGIES USED FOR DEVELOPING THE SYSTEM.

Following languages and technologies have been used to develop the web based system.

- HTML (HyperText Mark-up Language)
  HTML [7] has been used to design the user interfaces of the web based system.

- CSS (Cascading Style Sheets)
  CSS [7] was used to design the web pages of the web based system.

- Java Scripts
  Java script [7] [8] has been used control the browser, and for validation of the system.

- PHP (Hypertext Pre-processor)
  PHP [7] [8] was used as the backend scripting language of the system. It is specific language for web based system development.

4.1.2. SOFTWARES AND DEVELOPMENT TOOLS

- Microsoft Windows 7 64 – bit OS
- WAMP Server version 2.2 including
  - ✓ Apache version 2.2.21
  - ✓ PHP version 5.3.10
  - ✓ MySQL version 5.5.20
- Notepad++
- Adobe Photoshop CS4

4.2. HARDWARE ENVIRONMENT

The system was developed using a notebook computer under following hardware environment.

- Processor – Intel(R) Core(TM) i3-2350M CPU @2.30GHz 2.30 GHz
- Installed memory (RAM) – 4.00 GB
- Hard Disk capacity ((HDD) – 500 GB
- Monitor – 17 inches' screen size with 1366 x 768 resolution

4.3. MOST IMPORTANT SOURCE CODES.

Database Access Control.

```
$dbusername = "root";
$dbpassword = "";
$dbhostname = "localhost";
$dbdatabase = "hope";

$dbhandle = mysql_connect($dbhostname, $dbusername, $dbpassword) or die ("Could not connect to database");

$selected = mysql_select_db($dbdatabase, $dbhandle);
```

Fig 4.1: Database Connection Code

Above code (Fig 4.1) implements the connection to the database of the system.

Administration Login Source Code

This module validates users and their passwords and redirect to the particular user homepage according to the privileges.

Also this source code is used to define sessions for specific users and avoid duplicate logins. (Figure 4.2)

```
// avoiding duplicate login
if (isset($_SESSION['signed_in']) && $_SESSION['signed_in'] == true) {
    $error = 'You are already signed in, <a href="logout.php">sign out</a> if you want.';
```

```
if (isset($_POST['LOGIN'])) {
 if(($_POST["ALUSERNAME"]&&$_POST["ALPASSWORD"])!=null) {
// assign post data from login interface to variables
    $myusername = $_POST["ALUSERNAME"];
    $mypassword = $_POST["ALPASSWORD"];
```

```
// Removing escapes special characters & selecting matching record

   $query = "SELECT * FROM user_management WHERE UserName='" . mysqli_real_escape_string($dbhandle,$myusername)."' and
   Password='" . mysqli_real_escape_string($dbhandle,$mypassword)."'";
```

```
$row = mysqli_fetch_array($result);
// check whether the user is active user or not and  create Sessions using login Information
if ($row['status'] == 1) {
    $_SESSION['NIC'] = $row['NIC'];
    $_SESSION['UserName'] = $myusername;
    $_SESSION['Role'] = $row['Role'];

    $_SESSION['SignedIn'] = true;
} else {
    $error = "You have no privileges to access this system .Please contact your system admin";
}
```

Fig 4.2: Login to the Backend Implementation Code

Source code for Login Out

All global session variables will be removed and destroyed using this LOGOUT module. Figure 4.3 is the source code for logout module.

```php
<?php

session_start();
// remove all session variables
session_unset();
// destroy the session
session_destroy();
header("location:adminlogin.php");
//close db connection
mysql_close($dbhandle);
?>
```

Fig 4.3: Log Out Source Code

Following figure 4.4 is showing the source code for querying the testings done within the particular day.

```php
<?php
include '..\dbconnect.php';
if(isset($_POST['todaytestings'])){
$da=date("Y-m-d");
//query testing details
$testqry = mysqli_query($dbhandle, "SELECT * FROM patients_testings where Date='$da' ")or die(mysqli_error($dbhandle));
//disply table headerss
echo"<div class='table-responsive'>
<table  class='table table-striped'>
<tr><th>Date</th>
<th>Ref No</th>
<th>Testings</th>
</tr>";
//displaying results
if(mysqli_num_rows($testqry)>0){

    while($rowqry = mysqli_fetch_assoc($testqry)){
        $myString = $rowqry['testings'];
            $myArray = explode(',', $myString);
        echo"<tr><td>".$rowqry['Date']."</td>";
        echo"<td>".$rowqry['ref_no']."</td>";
        for($i=0; $i<sizeof($myArray); $i++){
            $namequery = mysqli_query($dbhandle, "SELECT * FROM testings WHERE TestId='$myArray[$i]' ")or die(mysqli_error($dbhandle));
            while($rownameqry = mysqli_fetch_assoc($namequery)){
                echo"<td>".$rownameqry['TestName']."</td>";
            }
        }
    }
}
}
else {
    echo"<tr>
    <td colspan='4'>Data not Found</td>
    </tr>";
    }
    echo"</table>";
    ?>
```

Fig 4.4: Take a report of Today Testing Code

The source code showing below figure 4.5 is used to highlight the items which are going out of stock from the inventory.

```php
<?php
//displaying items going out of stocks
$CRITICAL=10;
$sql2=mysqli_query($dbhandle, "select * from inventory where qtyleft<='$CRITICAL'");
while($row2=mysqli_fetch_array($sql2))
{
echo '<li style="color:red;">'.$row2['item'].' going out of stock</li>';
}
?>
```

Fig 4.5: Take a report of going out of stocks Code

Following figure 4.6 is illustrating the source code from inventory page to get the current status of the inventory.

```php
<?php
//displaying current inventory
$da=date("Y-m-d");
$sql=mysqli_query($dbhandle, "SELECT * FROM inventory");
$i=1;
while($row=mysqli_fetch_array($sql)){
    $id=$row['id'];
    $date=$da;
    $item=$row['item'];
    $qtyleft=$row['qtyleft'];
    $qty_dispatch=$row['qty_dispatch'];
    if($i%2){
    ?>
        <tr id="<?php echo $id; ?>" class="edit_tr">
    <?php } else { ?>
        <tr id="<?php echo $id; ?>" bgcolor="#f2f2f2" class="edit_tr">
    <?php } ?>
        <td class="edit_td">
        <span class="text"><?php echo $da; ?></span>
        </td>
        <td>
        <span class="text"><?php echo $id; ?></span>
        </td>
        <td>
        <span class="text"><?php echo $item; ?></span>
        </td>
        <td>
        <span class="text"><?php echo $qtyleft; ?></span>
        </td>
        <td>
        <span class="text"><?php echo $qty_dispatch; ?></span>
        </td>
        </tr>
        <?php
        $i++;
        }
    ?>
```

Fig 4.6: Showing Current State of Stock Code

4.4. REUSED MODULES AND LIBRARIES

To maintain a quality software in an efficient manner, following modules, libraries and codes were reused while developing this system.

- Bootstrap Templates

  Bootstrap Templates are used to customize the user interfaces of relevant users. It is a free and open source tool for web designing which has HTML [7] and CSS [7] based in build modules for forms, tables, buttons, etc. It is used in user interface to have high quality and responsive web design.

- mPDF Library

  mPDF is a PHP [7] [8] library which helps to generate PDF files. mPDF was used to generate blood result reports.

- jQuery Java Script [7] [8] Library
  jQuery is a JavaScript library which helps to manipulate documents, event handling easily and fast. jQuery is used in this system specially for validation and ajax calls.

CHAPTER 05 – EVALUATION

The evaluation [6] of performance, from the perspectives of both developers and users, of complex systems of hardware and software. Modern computer-based information systems have become increasingly complex because of networking, distributed computing, distributed and heterogeneous databases, and the need to store large quantities of data. People are relying increasingly on computer systems to support daily activities. When these systems fail, significant breakdowns may ensue.

A computer system can fail in two major ways. First, functional failure occurs when the system fails to generate the correct results for a set of inputs. For example, if an information system fails to retrieve records that match a set of keywords, or if an air-missile tracking system fails to distinguish between a friendly and enemy missile, a functional failure has occurred. Second, performance failure occurs when the system operates correctly but fails to deliver the results in a timely fashion. For example, if an information system takes a longer time than users are willing to wait for the records they requested, the system is said to fail performance-wise even though it may eventually retrieve the correct set of records. Also, if the air-missile tracking system fails to detect an enemy missile in sufficient time to launch a counterattack, the system manifests performance failure.

Therefore, in designing a computer system it is necessary to guarantee that the end product will display neither functional nor performance failure. It is then necessary to predict the performance of computer systems when they are under design and development, as well as to predict the impact of changes in configurations of existing systems. This requires the use of predictive performance models. [7]

5.1. TEST PLAN

This section describes the levels of tests that take place during development: integration, system security, and user acceptance tests, and the planning that is needed. The test environment is described in terms of milestones, schedules, and resources needed to support testing.

5.1.1 Objectives

The main objective of test plan is to confirm whether the developed system fulfil the requirement of Hope Diagnostics Medical Laboratory Services.

5.1.2 Scope

This section describes the projected boundaries of the planned tests. Include a summary of any constraints imposed on the testing, whether they are because of a lack of specialized test equipment, or constraints on time or resources.

Following functions are tested.

- Staff user login
- User management
- Doctors Management
- Patient user login
- Invoice generating
- Test report printing

Non Functional Requirements to be tested.

- Security
- Browser Compatibility
- Platform Compatibility

5.2. TESTING TECHNIQUES

5.2.1. BLACK BOX TESTING

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance.

5.3. TEST PROCESS

- Requirements analysing
  The requirements were identified and analysed by using use cases designed for the system. Observations are taken from interviewing the clients & employees.

- Preparing test cases
  After requirements were analysed, based on them, the test cases will be designed.

- Executing Test cases
  Test cases will be tested based on the designed test case using test data. The results will be prompted on Appendix E.

- Defect Tracking and Reporting
  Errors found in the execution process will be fixed and verified. They will be tracked and retested.

5.4. TEST TYPES

- Integration Testing
  Integration testing can be carried out when all the modules are integrated as a group. Purpose of this testing is to make sure each modules are working properly after integrated them.

- System Testing
  System testing will be carried out after integration testing. Purpose of this is to evaluate end user requirements.

- Acceptance Testing
  Acceptance testing will be carried out after system testing by administration and employees of the Hope diagnostics.

- Regression Testing
  To make sure the changes made to fix issues found & future enhancements will not affect at current system.

5.5. DELIVERABLES

Test reports with all the results of test cases will be delivered.

5.6. TEST RESULTS

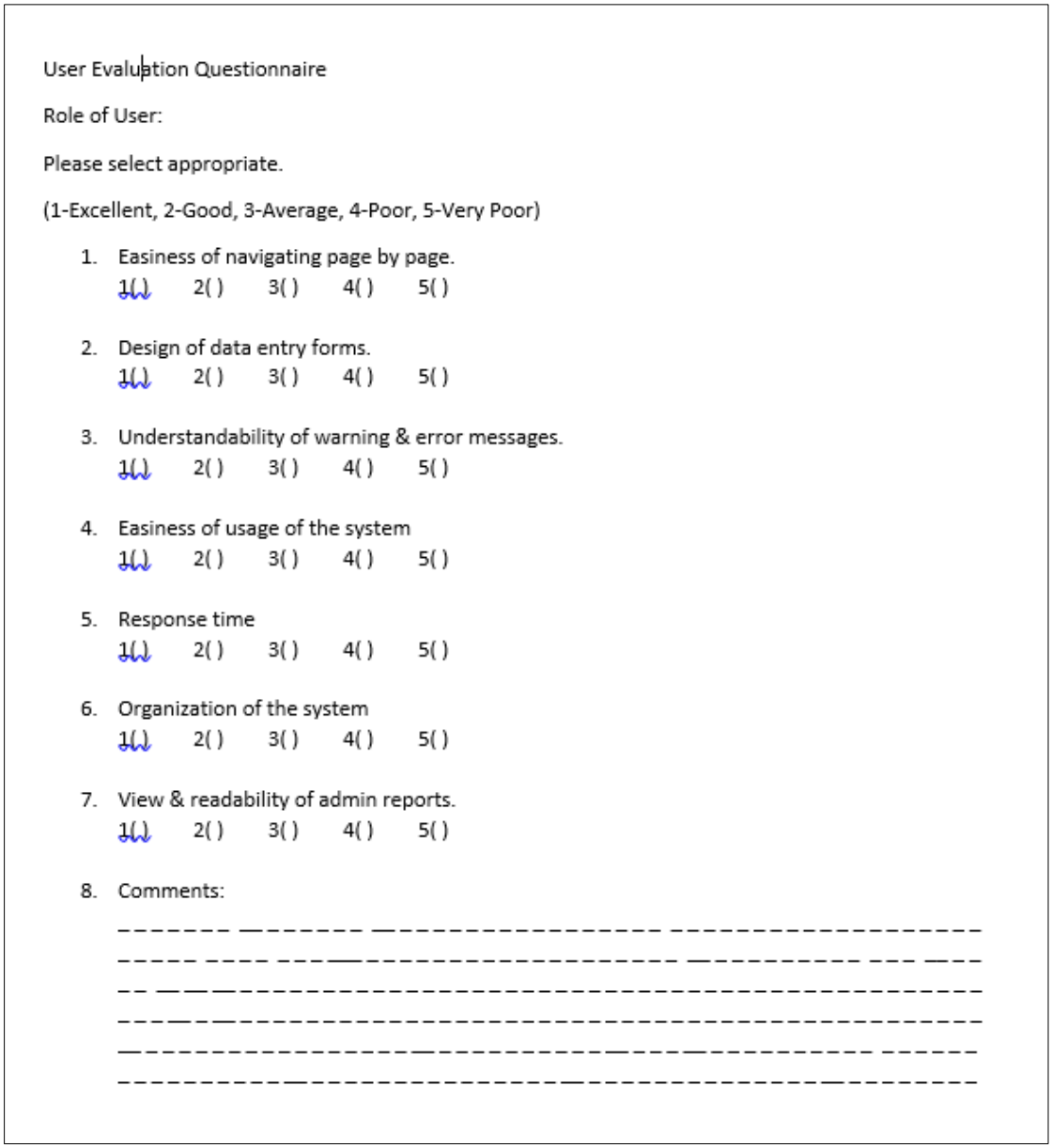
Following table shows the results of testing at user login interface of final testing stage. All test results will be mentioned under Appendix E.

Module		User Login Interface			
Tested Area		Checking user logins			
Test case Description					
ID	Test Case	Expected Result	Actual Result	Priority	Status
1	Kept both fields blank and click Login button	Display the warning message		Medium	Pass
2	Enter valid user name and click Login button without entering password	Display the warning message		Medium	Pass
3	Enter valid password and click Login button without entering user name	Display the warning message		Medium	Pass
4	Enter invalid user name and password and click Login button	Display the warning message		High	Pass
5	Enter valid username name and password and click Login button	Successful login with correct privileges		High	Pass

Table 5.1: Test results of User login

5.7 USER EVALUATION

To get feedback from end users, a user evaluation questionnaire will be given to end users of the system. In this system client is taken as administrator and other employees are taken as normal system users. After getting feedbacks, they will be evaluating and summarized. Following figure 5.1 is the questionnaire given to the system users.



User Evaluation Questionnaire

Role of User:

Please select appropriate.

(1-Excellent, 2-Good, 3-Average, 4-Poor, 5-Very Poor)

1. Easiness of navigating page by page.
   1( ) 2( ) 3( ) 4( ) 5( )
2. Design of data entry forms.
   1( ) 2( ) 3( ) 4( ) 5( )
3. Understandability of warning & error messages.
   1( ) 2( ) 3( ) 4( ) 5( )
4. Easiness of usage of the system
   1( ) 2( ) 3( ) 4( ) 5( )
5. Response time
   1( ) 2( ) 3( ) 4( ) 5( )
6. Organization of the system
   1( ) 2( ) 3( ) 4( ) 5( )
7. View & readability of admin reports.
   1( ) 2( ) 3( ) 4( ) 5( )
8. Comments:

Figure 5.1: User Evaluation Questionnaire

As mentioned above, to summarize user evaluation data, 5 marks are given for the every "very good" answer and gradually decrement 1 by 1 to other answers as every "very poor" answer is given 1 mark. Then get the sum of values of the users' answers and get the percentage. Following figure 5.2 is illustrating the user evaluation results.

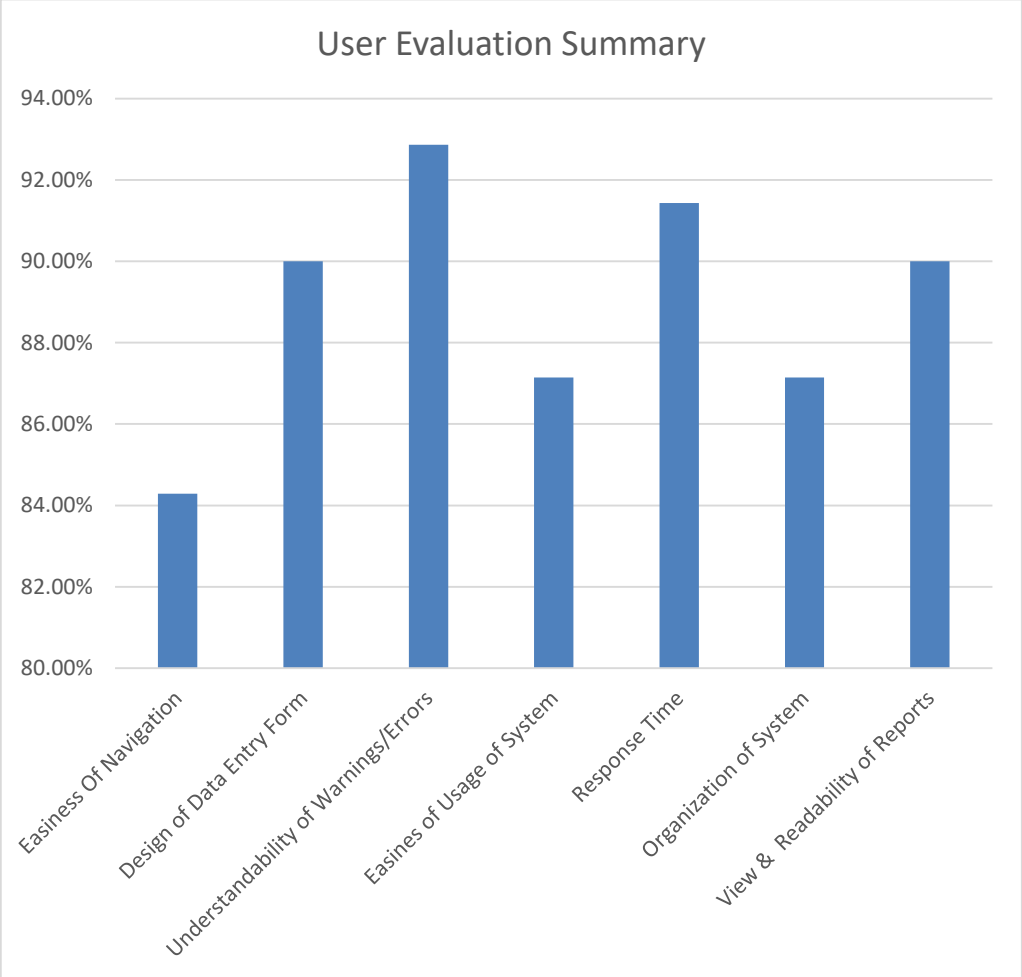


Figure 5.2: User Evaluation Results

CHAPTER 06 – CONCLUSION

6.1 CRITICAL ASSESSMENT OF THE PROJECT

Web Based Report and Invoice Generating System for Hope Diagnostics Medical Laboratory Service was developed for the purpose of improving efficiency and accuracy of data manipulating and making easy of report handling of the company. The success of the project is depending on the satisfaction of the client.

It is important whether the developed system is the best output for the requirement of the client or not within the selected scope. The user satisfaction is in a great stand which highlights the satisfaction of the client in to his expected level.

The new system meets many advantages over the previous system. Such as it makes easy to print blood reports and the close relationship with patients etc. I had to face some problems while developing the system.

I was new to this field as well as I haven't worked in a project like this before. Therefore, I had to adapt to situations as soon as possible. When I started this project, lack of my knowledge and skills in this field was a big problem for me. And it was hard to make appointments with client because of his busy day to day career. And the lack of IT knowledge of the staff in Hope Diagnostics was an important issue I had to face.

The most important thing is that I could be able to experience the situations while working on a project. It helps me to manage the time while working on a project. And it enhanced my programming knowledge and taught me to get backups of the system frequently. And it enhanced my communication and presenting skills since I had to work with the people in a business environment.

6.2 PROBLEMS ENCOUNTERED

- The client environment was new to me. Therefore, I have to collect their requirements very carefully.
- Lack of knowledge of designing and programming tools and designed patterns
- Lack of client's programming literacy.

6.3 LESSONS LEARNT

The most valuable thing is the lessons I learned and experiences I gained throughout this project. It was a challenge to fulfil all requirements. But after I started it under the well-planned guidance of my supervisor I could able to finish the project successfully.

- I learnt to work on a given time frame and adjust my works according to it.
- I could be able to sharpen my programming knowledge.
- During evaluation phase I learned how to test my work and correct my errors
- I could be able to improve my interpersonal skills while working with client environment specially while requirement gathering.

6.4 ACHIEVED OBJECTIVES

Achieved objectives are mentioned below.

- Automated data processing between payment transaction & blood report printing.
- Blood reports can be viewed and downloaded online confidentially through website.
- Regular users can register through website and maintain their past blood reports.
- Regular patients can get cost estimation through online.
- Administration can take reports over the testings done by them and the stocks they have.
- Doctors can login to their interface and view patients' blood reports.

6.5 FUTURE ENHANCEMENT

This system is fulfilling many requirements of the end users in different ways. But few enhancements and facilities are to be added to the system to facilitate end users with some new features to make easy for them.

- Patients to be able to channel doctors of medical centre online.
- Patients to be make payments online and make appointments online.
- Patients to be notified by SMS when their blood reports are ready.

REFERENCES

[1]: Wikipedia, *Object oriented design*. [Online]Available: http://en.wikipedia.org/wiki/Object-oriented_design [Accessed: 5th May 2017]

[2]: Wikipedia, *Use case diagrams*. [Online] Available: http://en.wikipedia.org/wiki/Use_Case_Diagram [Accessed: 10th May 2017]

[3]: Wikipedia, *Class Diagrams*. [Online] Available: http://en.wikipedia.org/wiki/Class_diagram [Accessed: 10th May 2017]

[4]: Wikipedia, *Database Design*. [Online] Available: http://en.wikipedia.org/wiki/Database_design [Accessed: 10th May 2017]

[5]: Wikipedia, *User Interface Design*. [Online] Available: http://en.wikipedia.org/wiki/User_interface_design [Accessed: 10th May 2017]

[6]: Encyclopaedia, *System Evaluation*. [Online] Available: http://encyclopedia2.thefreedictionary.com/Computer-system+evaluation [Accessed: 8th August 2017]

[7]: W3schools, *html, css, php, javascripts, mysql*. [Online] Available: https://www.w3schools.com/ [Accessed: 8th August 2016]

[8]: Stack overflow, *php, javascript, mysql*. [Online] Available: https://stackoverflow.com/questions [Accessed: 8th April 2017]

[9] : Wikipedia, Rational Unified Process [Online] Available: https://en.wikipedia.org/wiki/Rational_Unified_Process [Accessed: 10th May 2017]

APPENDIX A

SYSTEM DOCUMENTATION

The documentation provides step by step instructions on how to install the system properly, and specifications can be used as a further development of the web based report and invoice generating system which was made. Also this document includes the requirements needed to operate the system software and tools that will be useful for possible improvements to the system in the future

Minimum Hardware Requirements

- Pentium IV or higher processor
- 512MB or above Random Access Memory
- 160GB or higher Hard Disk
- 32MB VGA card

Minimum Software Requirements

- Windows 7 or higher operating system
- Wamp Server version 2.5 including
  - ✓ PHP version 5.5.12
  - ✓ Apache version 2.4.9
  - ✓ MySQL Client version 5.6.17
  - ✓ phpMyAdmin 4.1.14
- Google Chrome 13.0.782.112m or Internet Explorer 10.0.9200.16721 or Mozilla Firefox 39.0 or higher web browser

Following additional software can be used to do further development or extension of the system.

- Notepad++
- Adobe Photoshop

Setting up Web Based Report and Invoice Generating System for Development

These are the steps to be carried out for setting up Web based report and invoice generating system for Development.

1. Required software and tools installation
2. Database installation
3. System installation
4. Setting up Database connection

Required Software Installations

Installing Wamp Server

- Download WampServer form http://www.wampserver.com/en/ and install it.
- Give C:\ drive as installation path of the computer.
- Please follow the installation guide provided in the above website before installing Wamp on your computer to avoid any conflicts.
- Run the Wamp Server and open your web browser and type localhost or 127.0.0.1 in the address bar and press enter. If the Wamp Server is installed properly, you could see the Wamp server home page on your browser window.

Installing Web browser

- The user has to install one of above mentioned web browser or upgrade them from their official web sites.

Database Installation

- Click on phpmyadmin link under “Your Aliases”. In the phpmyadmin window select “Databases” tab and create a new database named “hope”
- Next go to the “Import” tab and click on “Choose File” and select the CD-ROM Drive: \Database\ hope.sql.
- Click “Go” button to finish the database installation.

System Installation

- Find “hope” folder which is located in CD-ROM Drive and copy entire folder in to C:\wamp\www location.

Set Up Database Connection

- There is a file named “dbconnect.php’ in the location hope\dbconnect.php and it can be used to change the user name and password when it is required. Figure A.1 is showing the source code of the file.

```
$dbusername = "root";
$dbpassword = "";
$dbhostname = "localhost";
$dbdatabase = "hope";

$dbhandle = mysql_connect($dbhostname, $dbusername, $dbpassword) or die ("Could not connect to database");

$selected = mysql_select_db($dbdatabase, $dbhandle);
```

Figure A.1: Code for Database Connection

In this source code it shows default user name and blank password and if it is required to change them at any circumstance, modify the “root” username and blank password respectively. And do same in ‘phpmyadmin’ page “users” tab.

Launching the System

- Open wamp Server and check whether it starts all services by the green colour wamp server icon on the system tray.

- Now type http://localhost/hope/index.php on the address bar in your web browser to launch the system.

APPENDIX: B

DESIGN DOCUMENTATION

Use case diagrams with detailed information

The use case diagram [2] for the system is mentioned in chapter 3 and it will be described section by section here. Figure B.1 is illustrating the use case diagram for login module.

Login Module

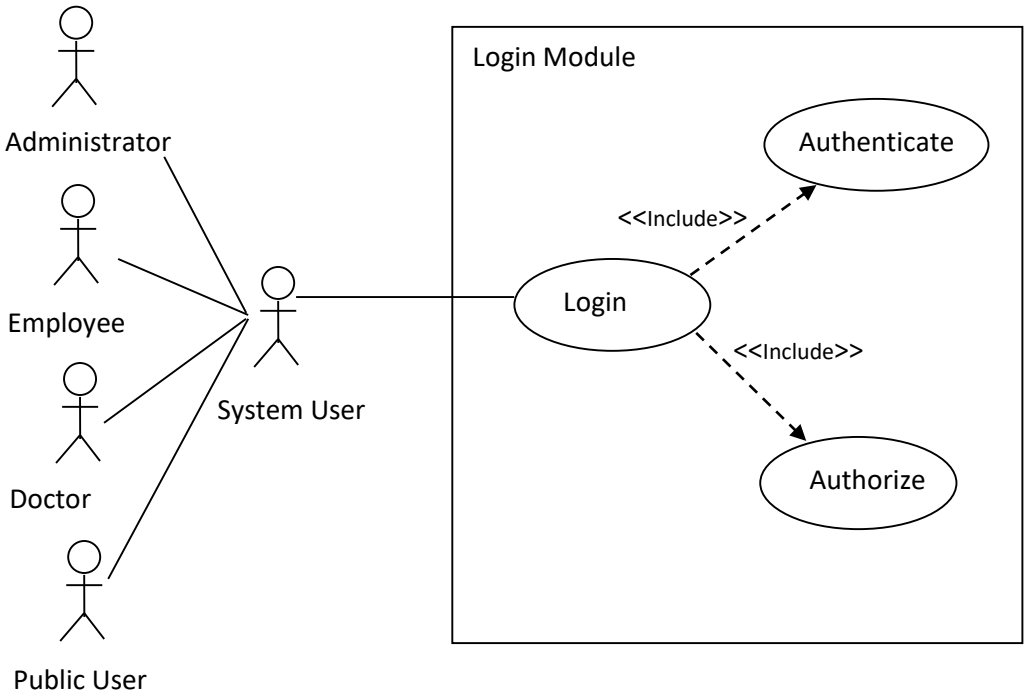


Figure B.1: Use case diagram for the Login module

Use Case	Login
Actors	System Users
Description	
Staff members can log into the backend & public users can log in to their account using their correct user names & passwords	
Pre-Conditions	
Having privileges to the system or registered as regular patient.	
Flow of Events	
1. Entering Username & password. 2. User Authentication. 3. User Authorization	
Post-Condition	
System users can log in to the System	

Table B.1: Use case Description for the login module

Accounts Management Module

All the backend user accounts of the system are managed by the administrator of the Hope Diagnostics. Following figure B.2 is showing the use case diagram of account management.

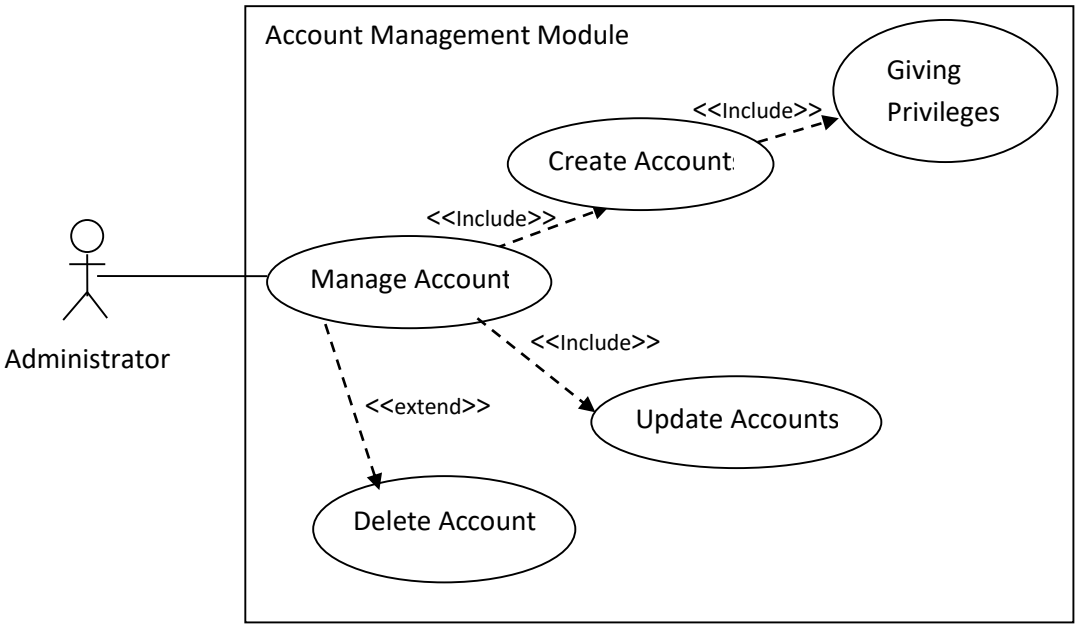


Figure B.2: Use case diagram for the Accounts Management module

Use Case	Account Management
Actors	System Administrator
Description	
Only System Administrator can add new accounts, update & delete existing accounts of the system	
Pre-Conditions	
Login to the system as Administrator	
Flow of Events	
1. Navigate to account management. 2. Add new user accounts to the system. 3. Update or Delete users in the system when necessary.	
Post-Condition	
New users can log in to the system with their privileges, updated users can use system with their new privileges, deleted users can't access the system.	

Table B.2: Use case Description for the Account Management module

Payment Transaction Module

All payment transactions are done live between Lab assistants and patients. Figure B.3 is the use case diagram for payment transaction module.

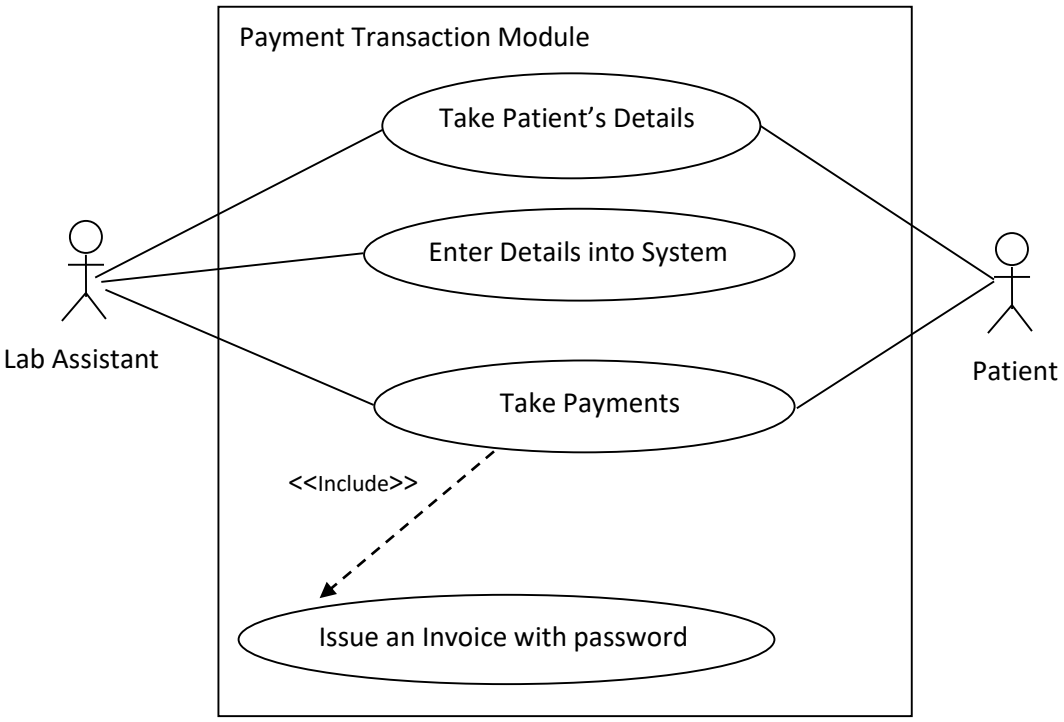


Figure B.3: Use case diagram for the Payment Transaction module

Use Case	Payment Transaction Module
Actors	Patients & Backend Users except doctors
Description	
Backend user will enter patients' details into the system to make easy for results printing	
Pre-Conditions	
Login to the system as a privileged user	
Flow of Events	
1. Navigate to payments. 2. Entering patient's personal details. 3. Select necessary testings. 4. Calculate total price and take payments. 5. Issue an invoice.	
Post-Condition	
Patient's personal details can be retrieved when making results reports. Patients could be able to view the result reports online with the reference and password in the Invoice	

Table B.3: Use case Description for the Payment Transaction module

APPENDIX C

USER DOCUMENTATION

User Manual provides guidance for use in the system and the functions of the system user from Web Based Report and Invoice Generating System. Administration, senior employees, employees and doctors can access backend system.

Since there are four categories, in this document describes administrator user levels with user mainly because it has almost all functions and system permissions. For other functions other than administrator has deal with will describe on the point of relevant user.

Login Module

The figure C.1 shows the login interface of the backend system. The user should provide correct username and password to access the system. If any user forgets his password, he has to go contact administrator of the system to reset his password.

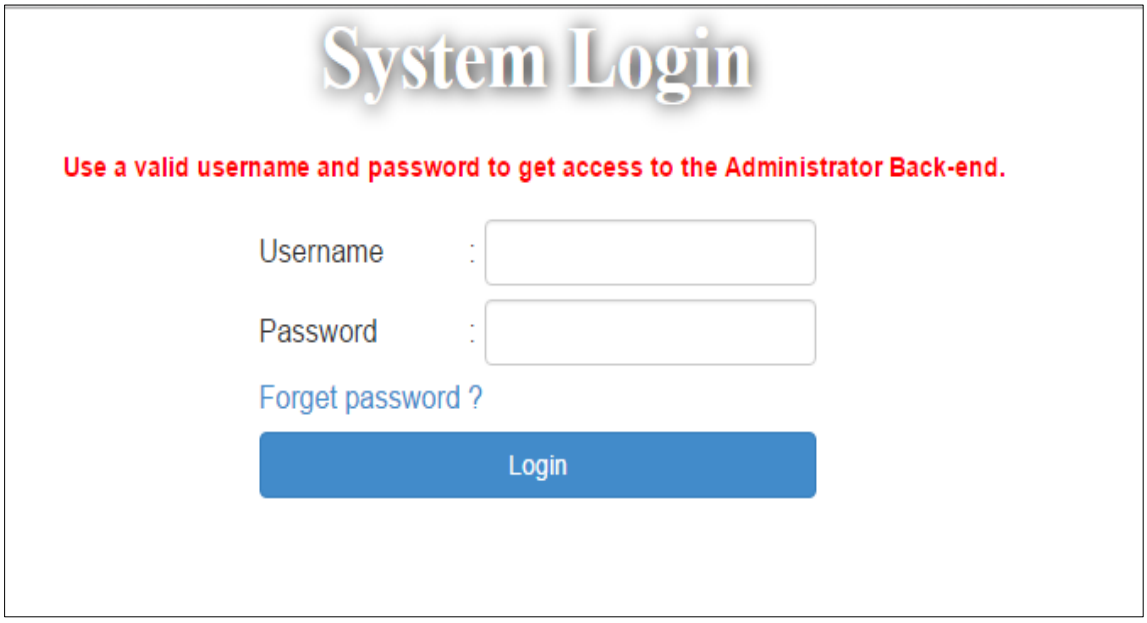


Figure C.1: Login module

Home Page

After a successful login, the user will be redirected to the appropriate home page based on their roles and the granted permission. The following figure C.2 shows homepage of Admin

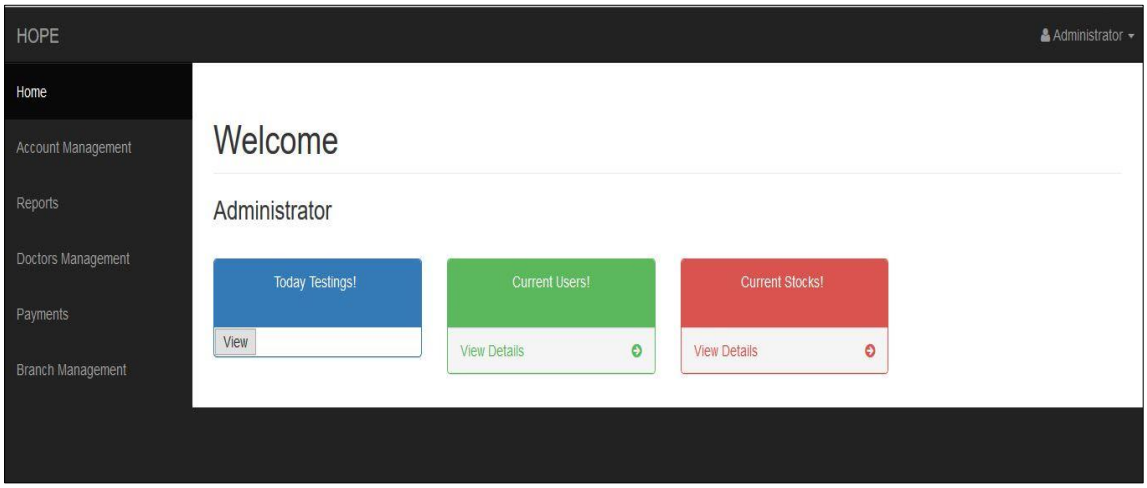


Figure C.2: Backend Home Page

Account Management Page

Information of the users can be diplayed on the Account Mangement module and when it required to delete a user from system, it needs to click on 'Cross' button on the "Delete" column. The figure C.3 demonstrates the interface of account management.

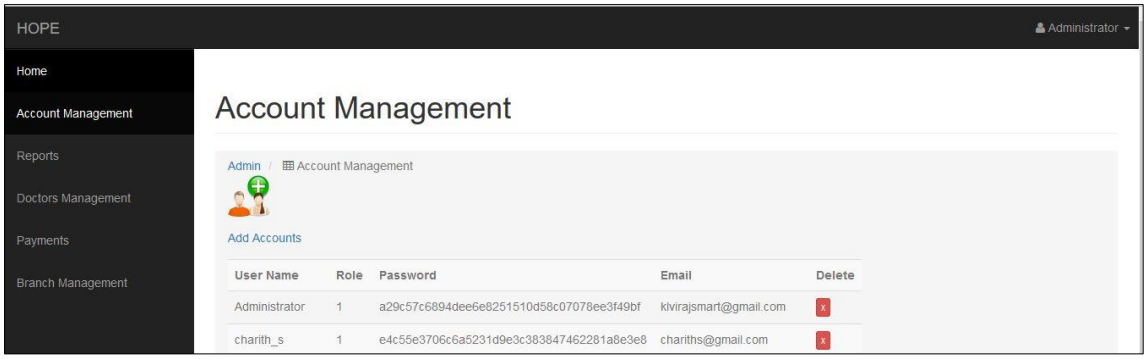


Figure C.3: Account Management Module

Add Account Page

By clicking on the button "Add Accounts" on the module for Account Management, new user can be created. There would be a form as figure C.4 to be filled all required fields, and then click "Add User",and then a new user can easily add to the system.

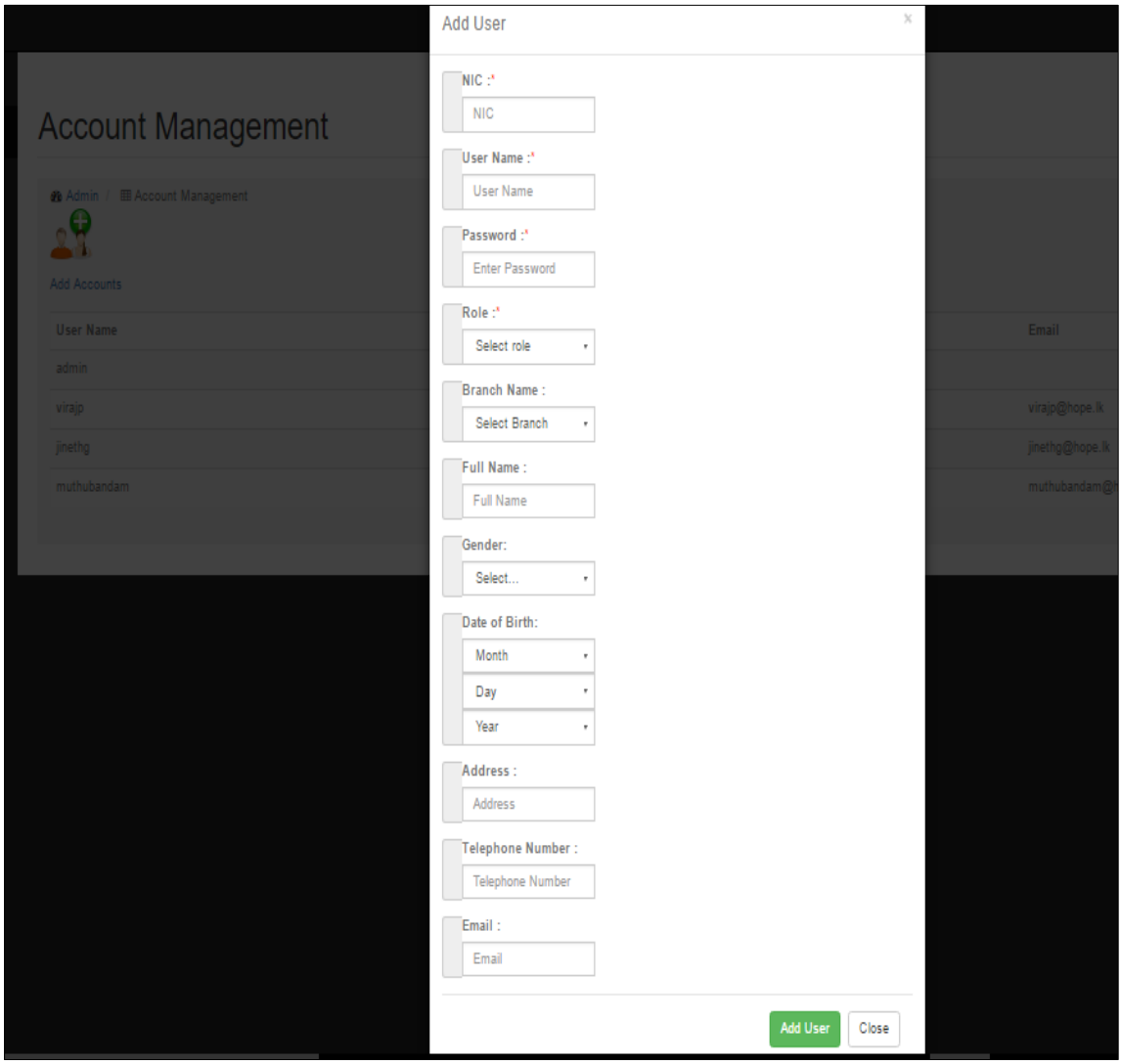


Figure C.4: Add Account Module

New Invoice Page

New Invoice can generate by clicking on the ‘Enter patient’ button. Then you have to enter correct details of the patient in correct format in to a form as figure C.5. Then you can select testings those you have to done by clicking on ‘Testings’ button. After all get the payment and click on the ‘Print’ button to print the new invoice.

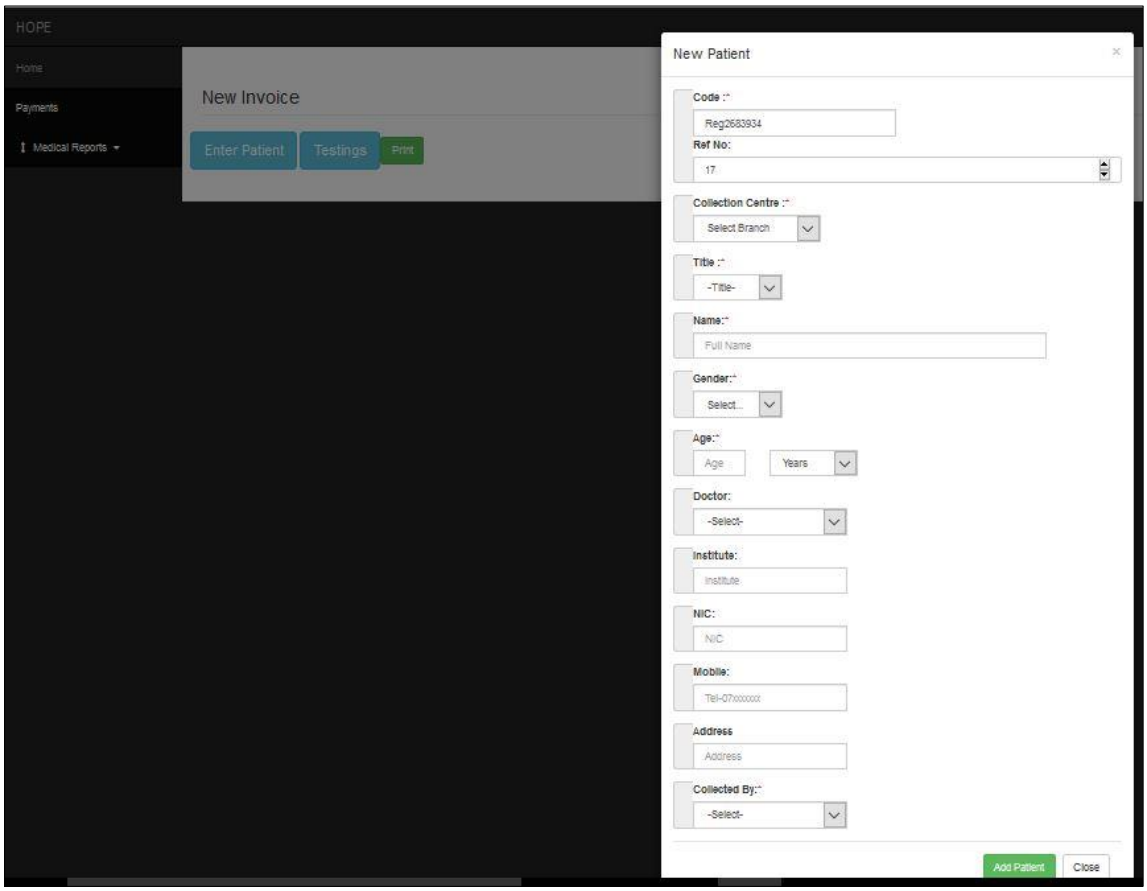


Figure C.5: Create a New Invoice Module

Blood Reports Printing Page.

Navigate to ‘Reports’ tab and select the relative Testing by clicking on the tab. Then enter the relative reference number into the box as in figure C.6 and click on ‘View’ button.

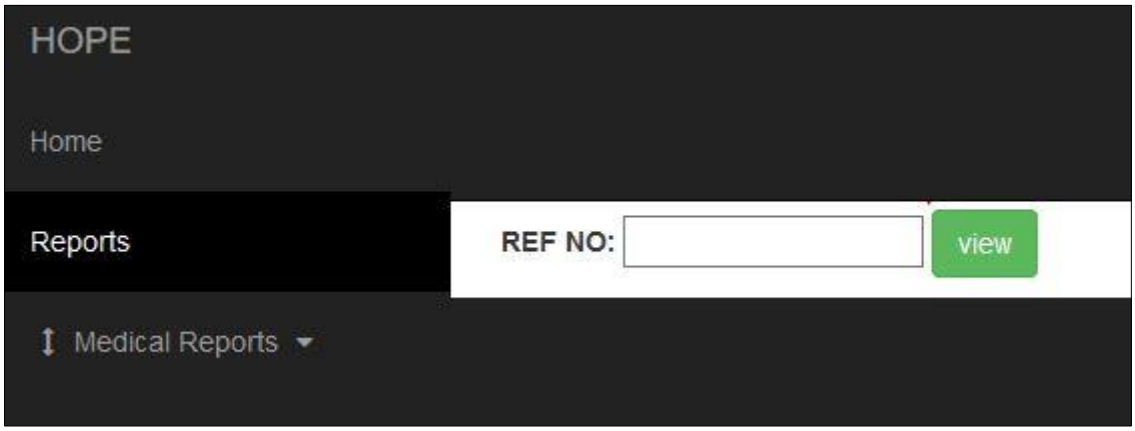


Figure C.6: Selecting A Test & Retrieving Data Page

Then all the information relative to the reference number will be retrieved and displayed as below figure C.7.

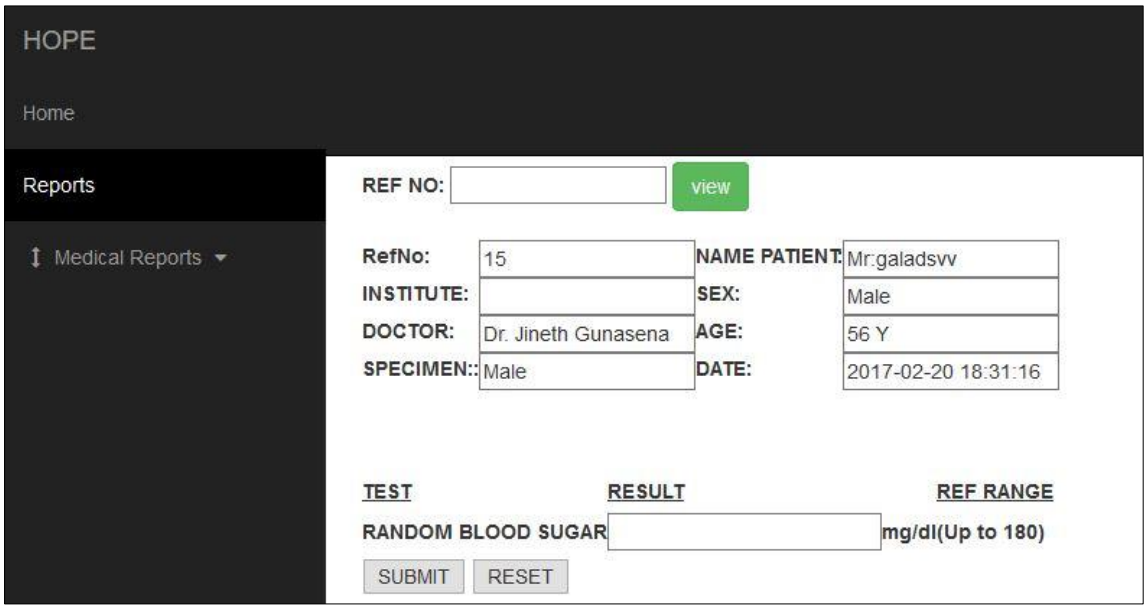


Figure C.7: The page after retrieving data of specific patient

Then click on ‘Submit’ button & click on the ‘Get PDF’ to generate the pdf file as in figure C.8.

Then print the report.

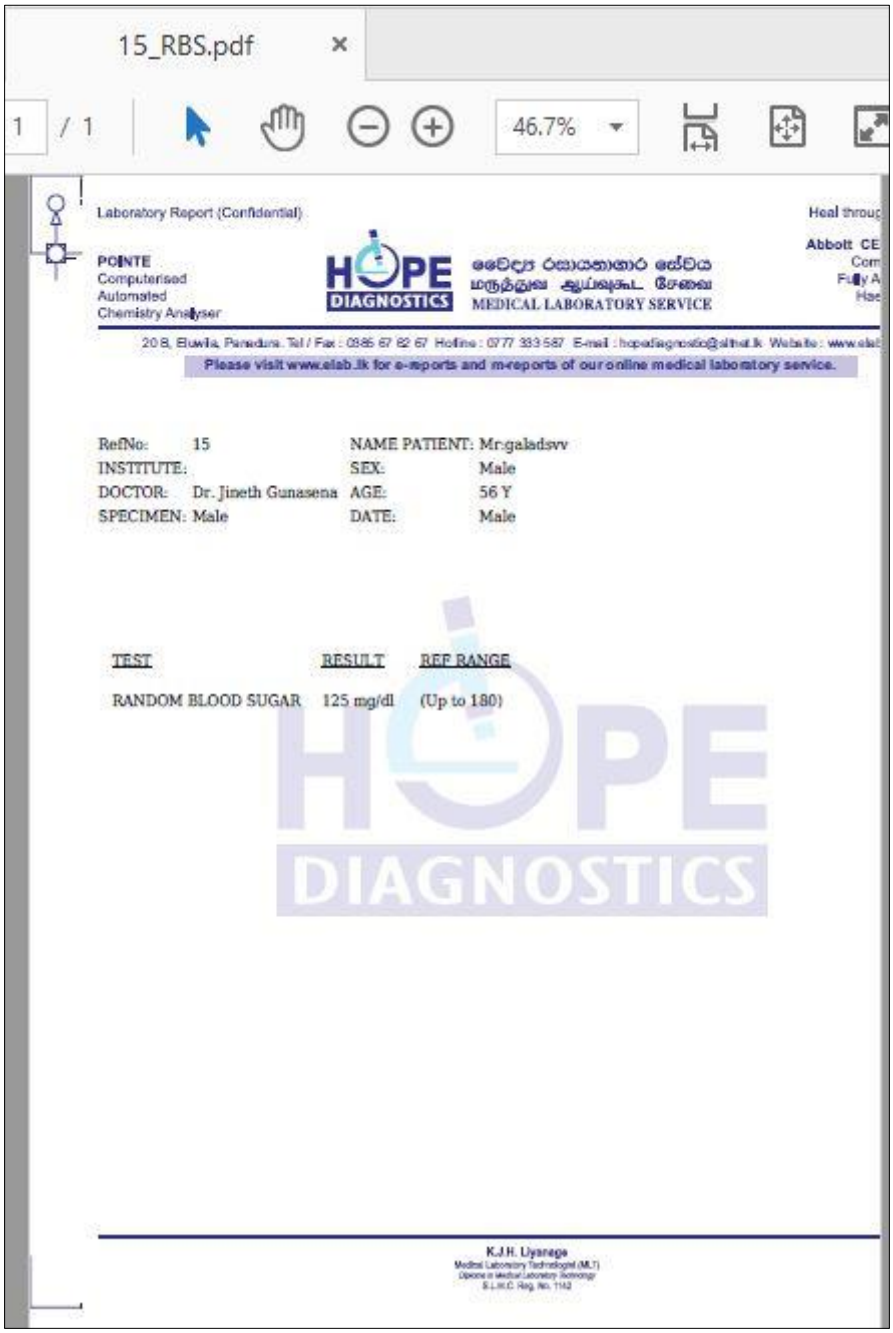


Figure C.8: Final Report PDF

The pdf file named with reference number & testing code will be saved in a local folder as showing in figure C.9.

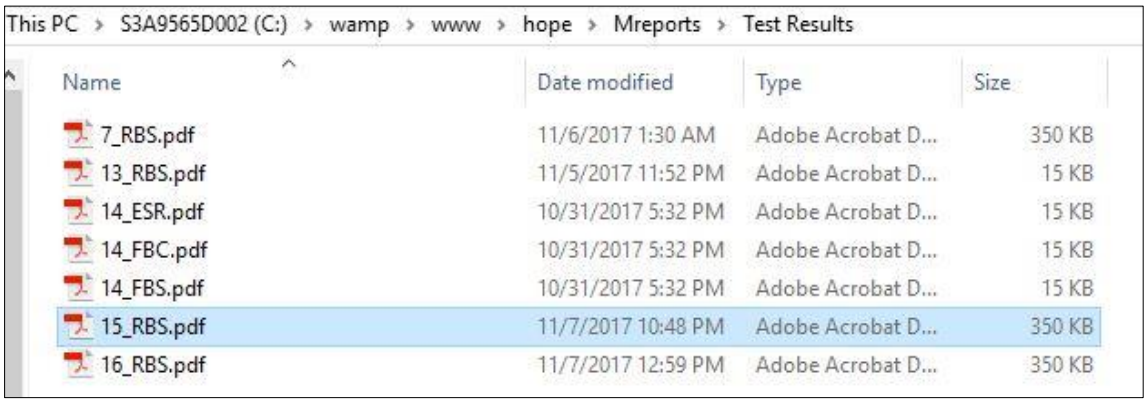


Figure C.9: PDF saved Location

Stocks Page.

Navigate to the ‘Home’ page and click on current stocks ‘View Details’ button as shown in figure C.10.

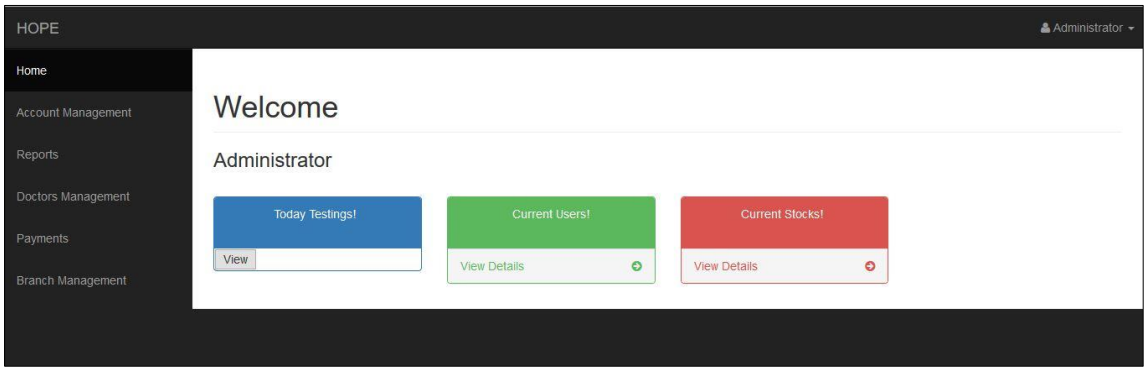


Figure C.10: Navigating to Stock Page

Then you will be directed to the stock page as illustrated in figure C.11. You can proceed all the things relevant to the stock on this page.

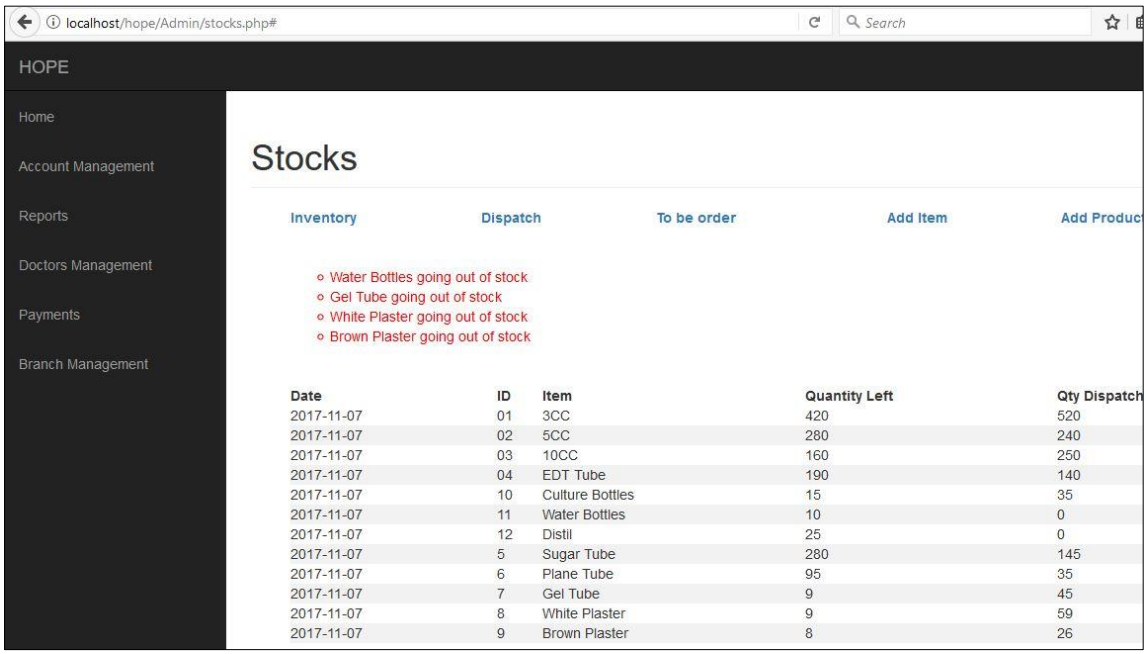


Figure C.11: Stock Module

Register Public User Page

Public users (regular patients) can register via the website by entering correct informatin into a form as below figure C.12.

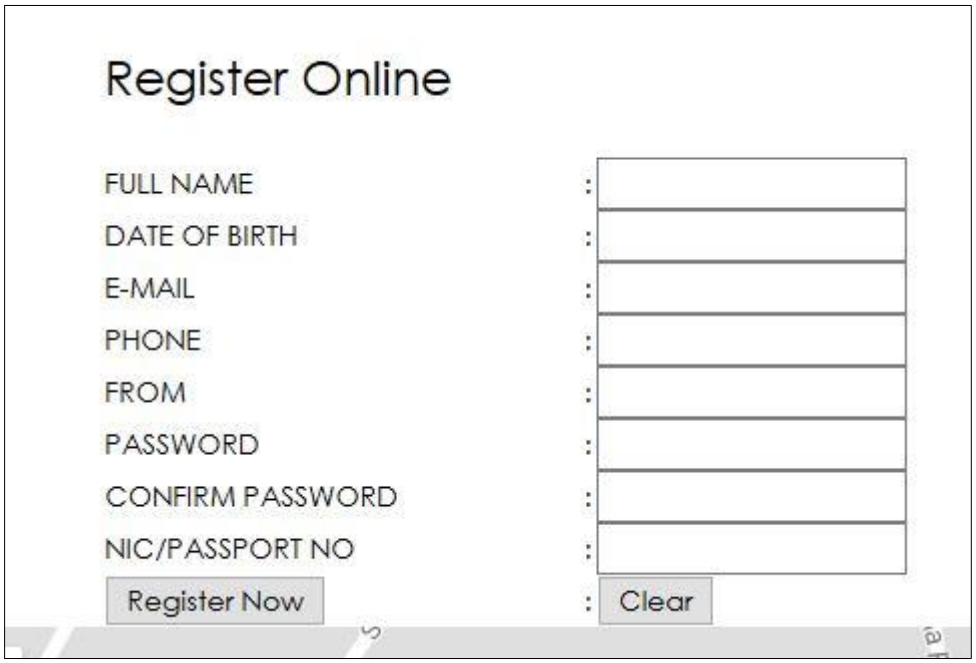


Figure C.12: Public user register module

Public User Login.

Registered public users can login to their accounts by entering correct login details in to the boxes as shown in figure C.13.

Figure C.13: Public User Login module

Online Viewing Page.

Any patient(not only the registered patients) can view their results online by providing correct details into boxes shown below figure C.14.

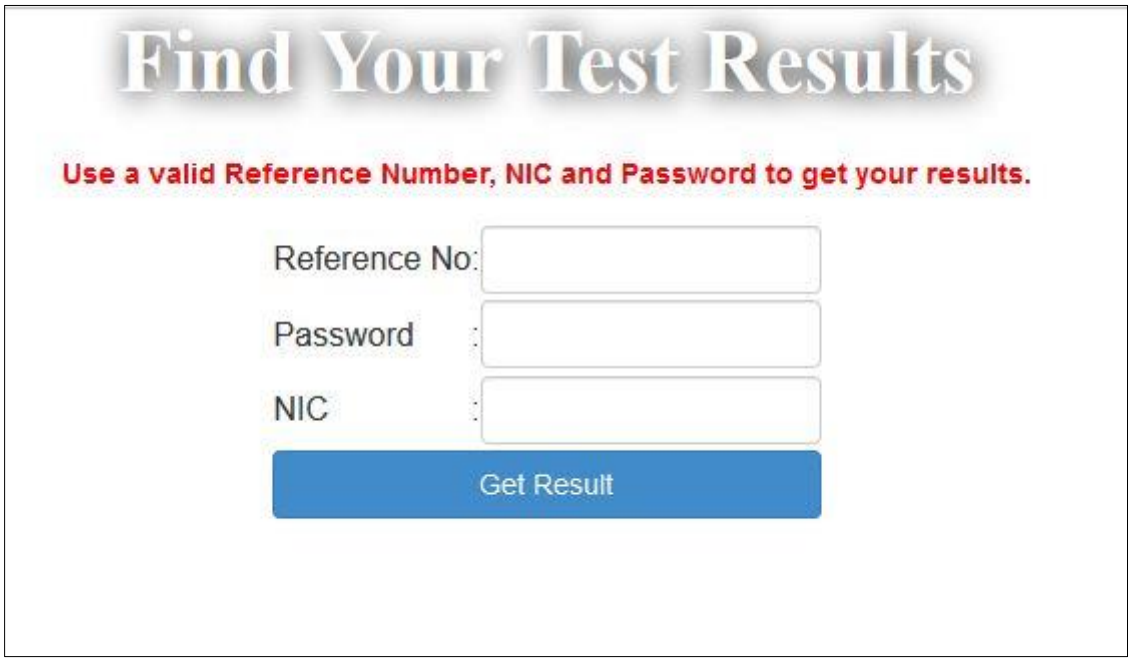


Figure C.14: Online Blood Report Viewing Module

APPENDIX D

MANAGEMENT REPORTS

The web based report & invoice generating system letting administrator and other parties to generate some management reports. Therefore, management can get some managerial decisions according to these reports.

Administrator can view day to day testing details directly click on view today testings. It will give a view of testings done by each patient with reference number. (Figure D.1)

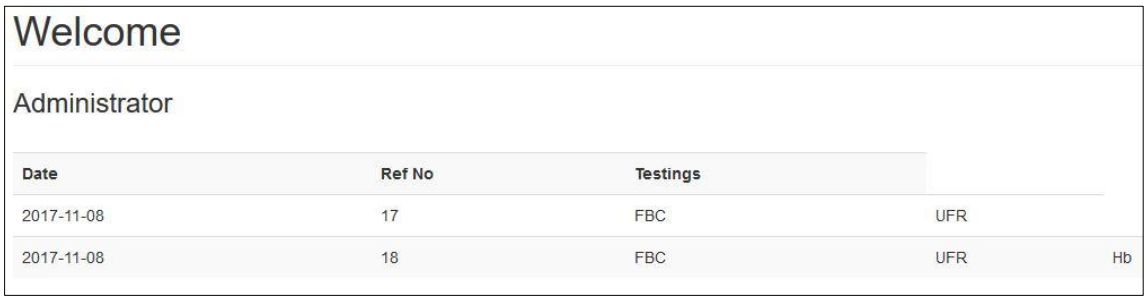


Figure D.1: Report of Today Testing Viewing Module

Administrator also can view dispatch details according to a specific time period. It will give an output of records for the dispatching day by day & item by item. (Figure D.2)

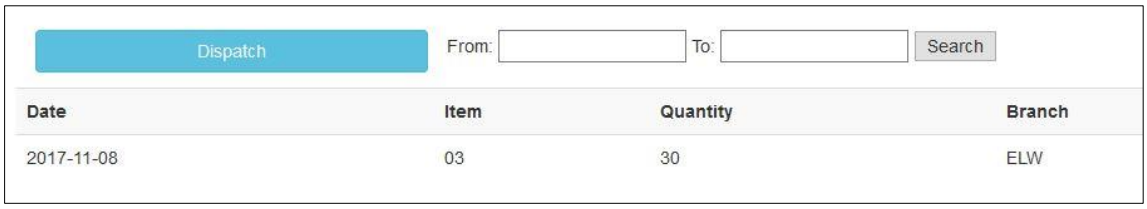


Figure D.2: Report of Dispatching Viewing Module

If administrator wants to see the current status of the stocks he can go to stocks page and get a report of current amount of all the items. And administrator will be notified the item going out of stock by indicating red highlighted. (Figure D.3)

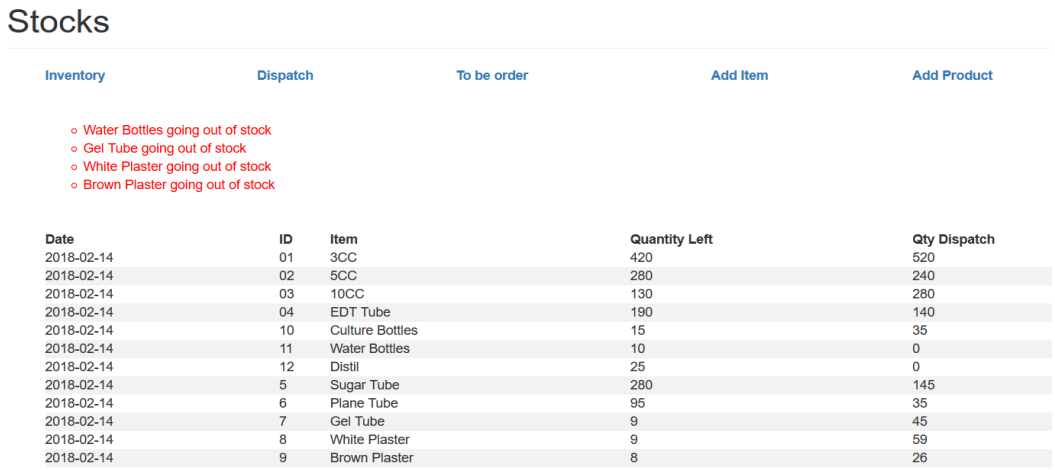


Figure D.3: Report of Current State

Administrator has been faschilated to get reports about testings based on following.

- Given Time Period
- Doctor
- Branch
- Testing

NAVIGATION TO MANAGEMENT REPORT

Following figure D.4 shows the navigation to the page to get management reports.

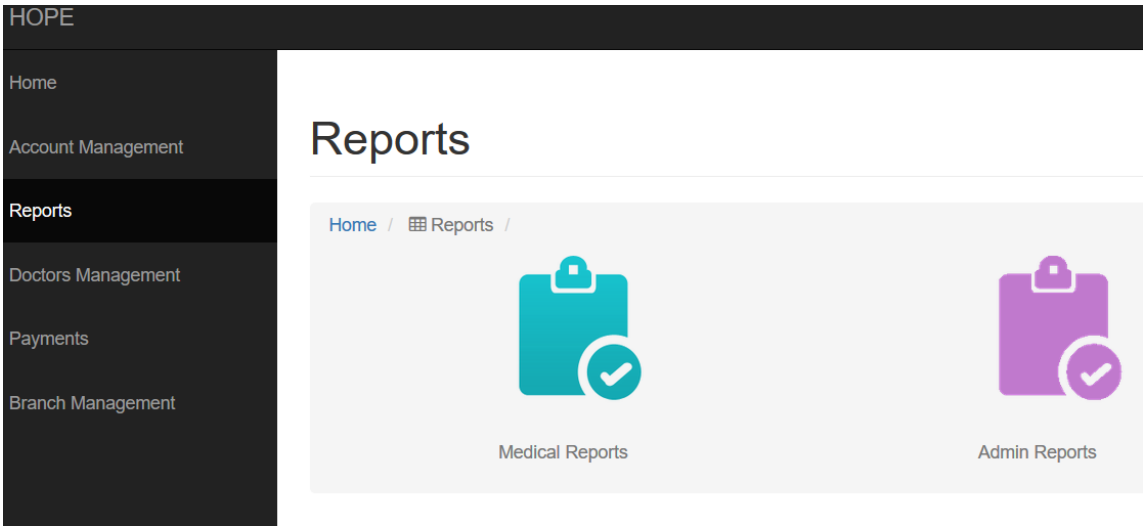


Figure D.4: Navigation to Management Report

Here we can give the conditions of queries based on the requirments. (Figure D.5)

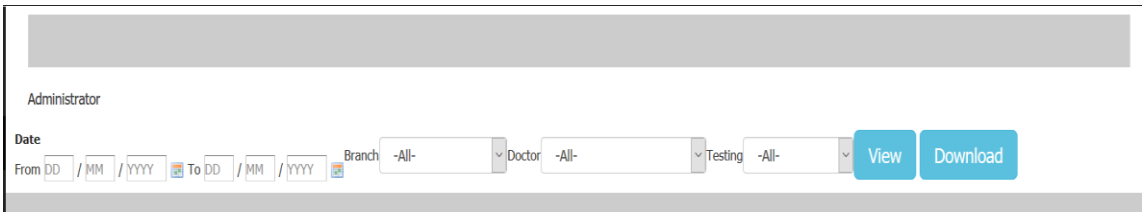


Figure D.5: Query Tabs for conditions

After the conditions are given, we can view the results as illustrate in figure D.6.

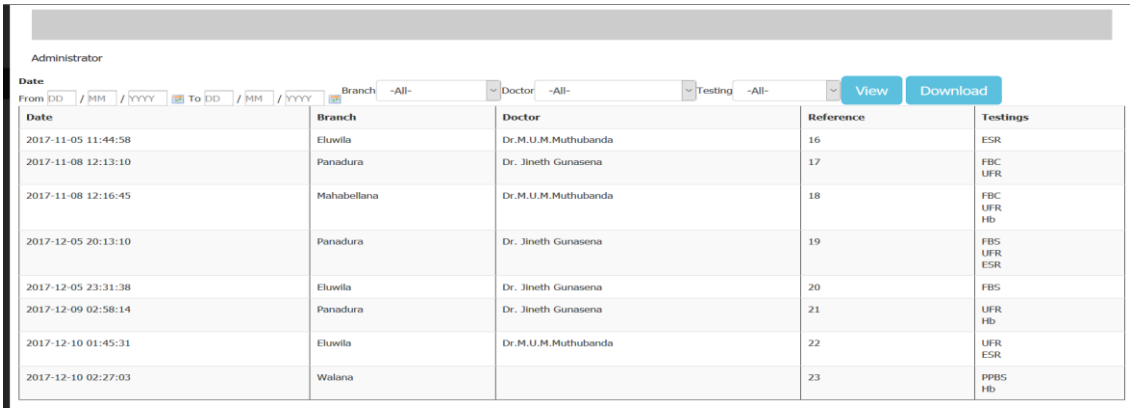


Figure D.6: Result Interface of Management Report

Administrator can export the reports he needs, in to a pdf file as below. (Figure D.7 & Figure D.8)

HOPE DIAGNOSTICS
වෛද්‍ය රසායනාගාර සේවය
மருத்துவ ஆய்வுகூட சேவை
MEDICAL LABORATORY SERVICE

Date: 12/02/2018

Report Name: Testings

Period: 01/11/2017: 31/12/2017

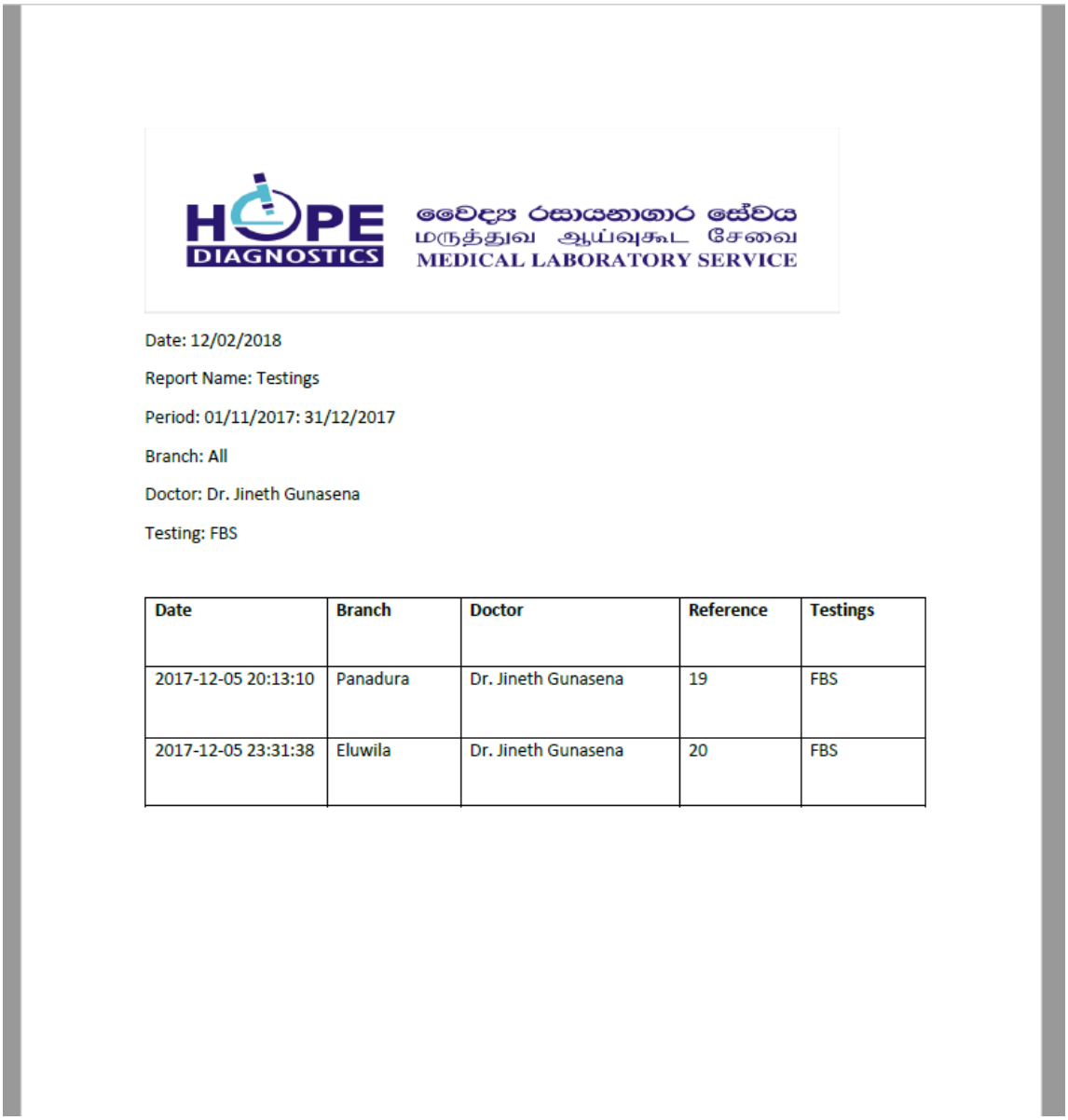
Branch: All

Doctor: All

Testing: All

Date	Branch	Doctor	Reference	Testings
2017-11-05 11:44:58	Eluwila	Dr.M.U.M.Muthubanda	16	ESR
2017-11-08 12:13:10	Panadura	Dr. Jineth Gunasena	17	FBC UFR
2017-11-08 12:16:45	Mahabellana	Dr.M.U.M.Muthubanda	18	FBC UFR Hb
2017-12-05 20:13:10	Panadura	Dr. Jineth Gunasena	19	FBS UFR ESR
2017-12-05 23:31:38	Eluwila	Dr. Jineth Gunasena	20	FBS
2017-12-09 02:58:14	Panadura	Dr. Jineth Gunasena	21	UFR Hb
2017-12-10 01:45:31	Eluwila	Dr.M.U.M.Muthubanda	22	UFR ESR
2017-12-10 02:27:03	Walana		23	PPBS Hb

Figure D.7: Exported Result Interface of Management Report 1



HOPE DIAGNOSTICS

වෛද්‍ය රසායනාගාර සේවය
மருத்துவ ஆய்வுகூட சேவை
MEDICAL LABORATORY SERVICE

Date: 12/02/2018

Report Name: Testings

Period: 01/11/2017: 31/12/2017

Branch: All

Doctor: Dr. Jineth Gunasena

Testing: FBS

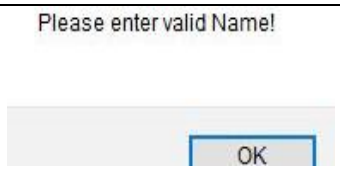
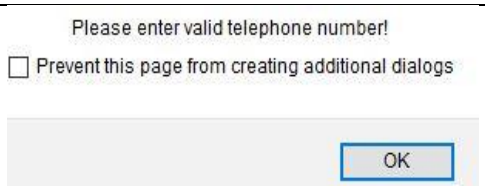
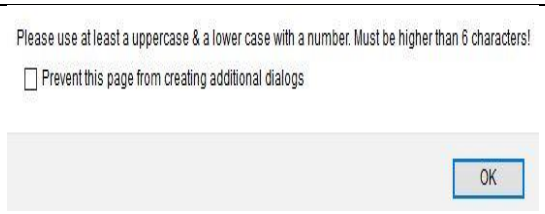
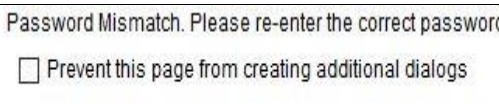
Date	Branch	Doctor	Reference	Testings
2017-12-05 20:13:10	Panadura	Dr. Jineth Gunasena	19	FBS
2017-12-05 23:31:38	Eluwila	Dr. Jineth Gunasena	20	FBS

Figure D.8: Exported Result Interface of Management Report 2

APPENDIX E

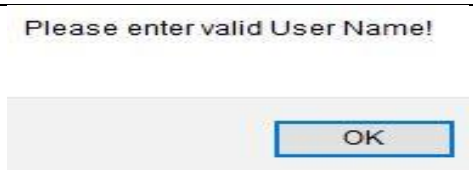
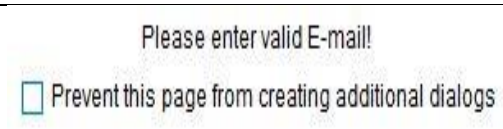
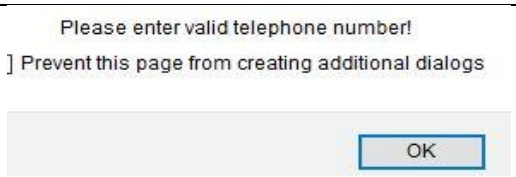
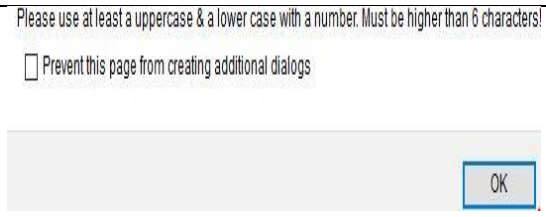
Test Results

All test cases of final testing are mentioned below table.

Module Name	Public user registration & login				
Test Area	Register online & Login				
Pre-Condition	Enter valid information				
Test case description					
Case No	**Test Case**	**Results Expecting**	**Results Got**	**Priority**	**Status**
1	Register a user without a proper name	Display proper warning message.		Medium	Pass
2	Register a user without a valid email	Display proper warning message.		High	Pass
3	Register a user without a proper phone number	Display proper warning message.		Medium	Pass
4	Register a user without a password according to criteria	Display proper warning message.		High	Pass
5	Confirrm password is not matching	Display proper warning message.		High	Pass
6	Enter an existing email or NIC	Display proper warning message.		High	Pass
7	Enter correcrt details & Register	Display proper succesful message & new record should be added.		High	Pass

8	Enter Correct Details & Login	Redirect to profile page	Redirected	High	Pass

Table E.1: Testing Public User Register.

Module Name	Manage Users				
Test Area	Add/Edit/Delete System Users & Login				
Pre-Condition	Login as Administrator Navigate to Account Management				
Test case description					
Case No	**Test Case**	**Results Expecting**	**Results Got**	**Priority**	**Status**
1	Add a user without a proper name	Display proper warning message.		Medium	Pass
2	Add a user without a proper username	Display proper warning message.		High	Pass
3	Add a user without a valid email	Display proper warning message.		High	Pass
4	Add a user without a proper phone number	Display proper warning message.		Medium	Pass
5	Add a user without a password according to criteria	Display proper warning message.		High	Pass
6	Enter an existing email or NIC	Display proper warning message.		High	Pass

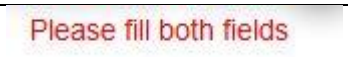
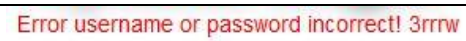
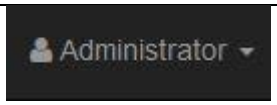
7	Enter correcrt details & Register	Display proper succesful message & new record should be added.		High	Pass
8	Edit user nformation & save	Display proper succesful message		High	Pass
9	Click on Delete button	Display proper warning message.		High	Pass
10	Click on Ok of Warning Message	Display proper succesful message		High	Pass
11	Login without filling all fields	Display proper warning message.		High	Pass
12	Login without correct Login details	Display proper warning message.		High	Pass
13	Login with Correct details	Direct to Backend profile page		High	Pass

Table E.2: Testing Account Management & Login.

APPENDIX F

CODE LISTING

This appendix consists an overview of codes those are used to develop this web based system.

PDF online viewing.

Getting entered relevent details, validate them & directing to the correct pdfs.

Following figure F.1 demonstrates the source code for validation of patients reference details.

```php
<?php

session_start();
require_once 'dbconnect.php';
$error = '';
$output ='';

if (isset($_POST['result'])){
    if(($_POST["reference"]&&$_POST["password"]&&$_POST["nic"])!=null){
        $reference = $_POST["reference"];
        $password = $_POST["password"];
        $nic = $_POST["nic"];

        $query = "SELECT * FROM patients_details WHERE RefNo='" . mysqli_real_escape_string($dbhandle,$reference)."' and
        CODE='" . mysqli_real_escape_string($dbhandle,$password)."' and NIC='" . mysqli_real_escape_string($dbhandle,$nic)."'";
        $result = mysqli_query($dbhandle,$query);

        $count = mysqli_num_rows($result);
        if ($count == 1){
            $query_t = "SELECT * FROM patients_testings WHERE ref_no='" . mysqli_real_escape_string($dbhandle,$reference)."'";
            $result_t = mysqli_query($dbhandle,$query_t);
            $count_t = mysqli_num_rows($result_t);
            if ($count_t == 1){
                $row_t = mysqli_fetch_array($result_t);
                $testing= $row_t['testings'];
                $testing_array = explode(",",$testing);
            if (sizeof($testing_array)>=2){
                $path="Mreports/Test Results/";
                foreach (glob($path.$reference."_*.pdf") as $filename) {
                 $fname=trim($filename,$path);

                  echo '<a href="'.$filename.'">'.$fname.'</a><br />';
            }
            }
            else{
                $error ="No Files found.";
```

Figure F.1: Code for the Blood Report Online Viewing

New account registering codes.

Source code for getting entered details, validate them according to criteria, match with current records & inserting as a new record is shown in figure F.2 below.

```php
<?php

$nic = '';
$password = '';
$fullname = '';
$dob = '';
$from = '';
$telephone = '';
$email = '';

if (isset($_POST['register'])){
$fullname = $_POST["FULLNAME"];
$dob = $_POST["DOB"];
$email = $_POST["EMAIL"];
$telephone = $_POST["PHONE"];
$from = $_POST["FROM"];
$password = sha1($_POST["PASSWORD"]);
$nic = $_POST["NIC"];

//check for duplicate NICs
$sql1 = "SELECT * FROM public_users WHERE nic_passp='" . mysqli_real_escape_string($dbhandle,$nic)."'";
$query1 = mysqli_query($dbhandle, $sql1);
$row1 = mysqli_num_rows($query1);
//check for duplicate user names
$sql2 = "SELECT * FROM public_users WHERE email='" . mysqli_real_escape_string($dbhandle,$email)."'";
$query2 = mysqli_query($dbhandle, $sql2);
$row2 = mysqli_num_rows($query2);

$last= "SELECT * FROM public_users ORDER BY index2 DESC LIMIT 1";
    $querylast = mysqli_query($dbhandle, $last);
    $rowlast=mysqli_fetch_assoc($querylast);
    $index=$rowlast['index2']+1;
```

Figure F.2: Code for the public new user registering

Figure F.3 is showing the source code for alerting messages according to criterias pre-defined with help of java script.

```php
if ($row1 != 0){
?>
    <script type="text/javascript">
    alert ("The user you entered is already exist");
    </script>;
<?php
    }elseif ($row2 != 0){
?>
    <script type="text/javascript">
    alert ("The user you entered is already exist");
    </script>;
<?php
    }else{

    $sql3 = "INSERT INTO public_users VALUES ('$index','$fullname','$dob','$email','$telephone','$from','$password','$nic')";

        if(mysqli_query($dbhandle, $sql3)){
?>
        <script type="text/javascript">
        alert ("New User Added Successfully!");
        </script>;
<?php
        }else{
        echo "<div class='alert alert-danger'>Error: " . $sql3 . "<br>" . mysqli_error($dbhandle) . "<br><a href='../index.html'>Back</a> </div> ";
        }


    }

}
```

Figure F.3: Code for Validating details & Adding New user to the Database

Figure F.4 below demonstrates a part of JQuery code of validating entering details in to forms.

```
<script type="text/javascript">

    function validate_name(){
        var onlyLetters = /^[a-zA-Z\s]+$/;
        var fullname = $('#FULLNAME').val();
        if(onlyLetters.test(fullname)==false){
            alert('Please enter valid Name!');
            return false;
        }
    }
```

Figure F.4: Code for jQuery validating part

Source code for retriewing data & sending them to create the pdf document is illustrating following figure F.5.

```
<?php
if(isset($_POST['submit'])){
 $ref_no=$_POST['RefNo'];
 $_SESSION['ref_no']=$ref_no;
 $_SESSION['test']="RBS";
if($ref_no!=null){
 $sql="SELECT * FROM patients_details WHERE RefNo = $ref_no";
 $result = mysqli_query($dbhandle,$sql);
 $row = mysqli_fetch_array($result);
 //echo $row['NameOfPatient'];
 //echo "Ref no: ". $ref_no;
 $sql1 = "SHOW COLUMNS FROM patients_details ";
 $query1 = mysqli_query($dbhandle, $sql1);
while ($row1 = mysqli_fetch_assoc($query1)) {
    $topics[] = $row1['Field'];
}
 $sql2 = "SELECT * FROM doctors_details WHERE NIC ='".$row['DOCTOR']."'";
 $query2 = mysqli_query($dbhandle,$sql2);
 $row2 = mysqli_fetch_assoc($query2);
 //print_r($topics);
 echo "<br/>";
 $_SESSION['html1']="";
 echo "<table><tr><td>";
 $_SESSION['html1'].="<br /><br /><br /><br /><br /><br /><br /><br /><br /><table><tr><td>";
 //$_SESSION['html1'].='<div class="row">';
 //$_SESSION['html1'].='<div class="col-md-6">';
 //$_SESSION['html1'].="<table>";
 echo "<table>";
 $_SESSION['html1'].="<table>";
```

Figure F.5: Code for Retriewing data & sending data to create the pdf

According to the codings shown in figure F.6 the data will be passed to a session variable in order to create PDF.

```
echo "<tr>";
$_SESSION['html1'].= "<tr>";
echo "<td><label>".str_replace("_"," ",$topics[0]).":</label></td>";
$_SESSION['html1'].="<td><label>".str_replace("_"," ",$topics[0]).":</label></td>";

echo '<td><input type="text" name="" id="" value="'.$row['RefNo'].'" /> </td>';
$_SESSION['html1'].= '<td>'.$row['RefNo'].'</td>';
echo "</tr>";
$_SESSION['html1'].="</tr>";


echo "<tr>";
$_SESSION['html1'].= "<tr>";
echo "<td><label>".str_replace("_"," ",$topics[9]).":</label></td>";
$_SESSION['html1'].="<td><label>".str_replace("_"," ",$topics[9]).":</label></td>";

echo '<td><input type="text" name="" id="" value="'.$row['INSTITUTE'].'" /> </td>';
$_SESSION['html1'].= '<td>'.$row['INSTITUTE'].'</td>';
echo "</tr>";
$_SESSION['html1'].="</tr>";
```

Figure F.6: Code for sending data in to a session for the PDF

From the source code shown in below figure F.7 the data from the session will be proscessed and view a final preview before create pdf.

```
<form action ="test.php" method="POST">
<label><input type="text" name="rand_blood_sugar" id="rand_blood_sugar" />mg/dl </label>
</td>
    <td>
    <label>(Up to 180)</label>
    </td></tr>

   <tr><td>

<input name="view" type="submit" value="SUBMIT" />
<input name="reset" type="reset" value="RESET" />
</form>
<?php
echo "</td></tr>
  </table>";
```

Figure F.7: Code for Pass the data to the final page

The figure F.8 demonstrate the source code for outputting a PDF document using the mpdf library. The all data inserted in to the session will be manipulated to a pdf file by using below code.

```php
<?php
session_start();

include 'MPDF60/mpdf.php';
require_once __DIR__ . '\MPDF60\vendor\autoload.php';
$html1=$_SESSION['html1'];
$ref_no=$_SESSION['ref_no'];
$test=$_SESSION['test'];
$mpdf=new mPDF();
//$mPDF->SetColumns(2);
$mpdf->SetImportUse();
$page = $mpdf->SetSourceFile('Labreport.pdf');
$tpl = $mpdf->ImportPage($page);
$mpdf->UseTemplate($tpl);
$mpdf->SetDisplayMode('fullpage');
$mpdf->WriteHTML("$html1");
//$mpdf->AddColumn();
//$mpdf->WriteHTML("Next coloumn");

$mpdf->Output('Test Results/'.$ref_no.'_'.$test.'.pdf', 'F')
?>
```

Figure F.8: Code for Creating PDF

Figure F.9 demonstrates the retriewing data from a table and display on a page. This source code is used in several times to display data from a table.

```php
<?php
include '..\dbconnect.php';
$sql = "SELECT * FROM user_management";
$result = mysqli_query($dbhandle, $sql);
echo"
        <div class='table-responsive'>
            <table  class='table table-striped'>
                <tr>
                    <th width='30%'>User Name</th>
                    <th width='10%'>Role</th>
                    <th width='30%'>Password</th>
                    <th width='30%'>Email</th>
                    <th width='5%'>Delete</th>
                </tr>";
if(mysqli_num_rows($result)>0){
    while($row = mysqli_fetch_array($result)){
        echo"<tr><td>".$row['UserName']."</td>";
                echo"<td class='role' data-id1=".$row['NIC']." contenteditable>".$row['Role']."</td>";
                echo"<td class='password' data-id2=".$row['NIC']." contenteditable>".$row['Password']."</td>";
                echo"<td class='email' data-id3=".$row['NIC']." contenteditable>".$row['Email']."</td>";
                echo"<td><button name='btn_delete' id='btn_delete' data-id4=".$row['NIC']." class='btn btn-xs btn-danger btn_delete'>x</button></td></tr>";
    }
}
else{
    echo"<tr>
                    <td colspan='4'>Data not Found</td>
                </tr>";
}
echo"</table>
            </div>";
?>
```

Figure F.9: Code for Showing User Table of System

Figure F.10 below is the coding for deleting a user in the user accounts table.

```php
<?php
include '..\dbconnect.php';
$sql = "DELETE FROM user_management WHERE NIC = '".$_POST["NIC"]."'";
if (mysqli_query($dbhandle, $sql)){
     echo 'Data Deleted';
}

?>
```

Figure F.10: Code for Deleting User from System

Following figure F.11 demonstrates the source code for dispatching items from stocks to branches & update tables accordingly.

```php
<?php
$date = '';
$id = '';
$item = '';
$branch = '';
$qty = '';

if (isset($_POST['submitdis'])){
    $date = $_POST["date"];
    $item = $_POST["item"];
    $branch = $_POST["branch"];
    $qty = $_POST["qty"];

    $sqldisp = "SELECT * FROM inventory WHERE id='" . mysqli_real_escape_string($dbhandle, $item)."'";
    $querydisp = mysqli_query($dbhandle, $sqldisp);
    while($rowdisp = mysqli_fetch_array($querydisp)){
        if ($qty>$rowdisp['qtyleft']-8){
            echo "Exceed the available amount";

        }
        else{
            $sqldisp1= "INSERT INTO dispatch VALUES ('$date','$item','$qty','$branch')";
            if(mysqli_query($dbhandle, $sqldisp1)){

                $sqldisp2 = "UPDATE inventory SET qtyleft = qtyleft - $qty WHERE id = $item";
                $sqldisp3 = "UPDATE inventory SET qty_dispatch=qty_dispatch+$qty WHERE id = $item";
                if(mysqli_query($dbhandle, $sqldisp2) && mysqli_query($dbhandle, $sqldisp3))
                {
                    echo "Items Dispatched";
                    //header("location:stocks.php");
                }
            }

        }
}}
header("location:stocks.php");
?>
```

Figure F.11: Code for Updating Stock table after a dispatch

Admin accounts are fascilated to have summay report of testings done on the day once they logged in. Figure F.12 illustrates the source code for taking a report of 'today testings' done by tha Lab.

```php
<?php
include '..\dbconnect.php';
if(isset($_POST['todaytestings'])){
$da=date("Y-m-d");
//query testing details
$testqry = mysqli_query($dbhandle, "SELECT * FROM patients_testings where Date='$da' ")or die(mysqli_error($dbhandle));
//disply table headerss
echo"<div class='table-responsive'>
<table  class='table table-striped'>
<tr><th>Date</th>
<th>Ref No</th>
<th>Testings</th>
</tr>";
//displaying results
if(mysqli_num_rows($testqry)>0){

    while($rowqry = mysqli_fetch_assoc($testqry)){
        $myString = $rowqry['testings'];
            $myArray = explode(',', $myString);
        echo"<tr><td>".$rowqry['Date']."</td>";
        echo"<td>".$rowqry['ref_no']."</td>";
        for($i=0; $i<sizeof($myArray); $i++){
            $namequery = mysqli_query($dbhandle, "SELECT * FROM testings WHERE TestId='$myArray[$i]' ")or die(mysqli_error($dbhandle));
            while($rownameqry = mysqli_fetch_assoc($namequery)){
                echo"<td>".$rownameqry['TestName']."</td>";
            }
        }
    }
}
}
else {
    echo"<tr>
    <td colspan='4'>Data not Found</td>
    </tr>";
    }
    echo"</table>";
    ?>
```

Figure F.12: Code for taking a report of Testings done Today

Administrator can takes a report of items dispatching to branches for a specific time period. Following figure F.13 shows the source code for generating those reports.

```php
<?php
include '..\dbconnect.php';
    if(isset($_POST['submitqry'])){
      $a=$_POST['from'];
      $b=$_POST['to'];

      $resultqry = mysqli_query($dbhandle, "SELECT * FROM dispatch where date BETWEEN '$a' AND '$b'");


      echo"<div class='table-responsive'>
          <table  class='table table-striped'>
              <tr><th>Date</th>
                      <th>Item</th>
                      <th>Quantity</th>
                      <th>Branch </th>

                      </tr>";
      if(mysqli_num_rows($resultqry)>0){
      while($rowqry = mysqli_fetch_assoc($resultqry))

      {

          echo"<tr><td>".$rowqry['date']."</td>";
          echo"<td>".$rowqry['id']."</td>";
          echo"<td>".$rowqry['qty']."</td>";
          echo"<td>".$rowqry['branch']."</td></tr>";
      }

    }
    }
      else {
          echo"<tr>
                  <td colspan='4'>Data not Found</td>
              </tr>";

      }
      echo"</table>";

      ?>
```

Figure F.13: Code for taking a report of dispatching for specific period

Followng figure F.14 is demonstrating the codes for taking Managements reports on Testing according to conditions. Conditions are applied for dates, branches, test types and doctors.

```php
require_once '../dbconnect.php';


if (isset($_POST['viewreport'])) {
$from_month=$_POST["fdate_1"];
$from_day  =$_POST["fdate_2"];
$from_year =$_POST["fdate_3"];
$to_month=$_POST["tdate_1"];
$to_day  =$_POST["tdate_2"];
$to_year =$_POST["tdate_3"];
$branch =$_POST["branch"];
$doctor =$_POST["doctor"];
$testing =$_POST["testing"];
//$testing1=$testing.",";

$from_date=$from_year."-".$from_month."-".$from_day;
$to_date=$to_year."-".$to_month."-".$to_day;

if (!empty($from_day) && !empty($to_day)){

    //$sqlbetdate = "SELECT * FROM patients_details WHERE Date BETWEEN '$from_date 00:00:00' AND  '$to_date 23:59:59'";
    $sqlbetdate = "SELECT * FROM patients_details AS d INNER JOIN patients_testings AS t ON d.RefNo = t.ref_no WHERE d.Date BETWEEN '$from_date 00:00:00' AND  '$to_date 23:59:59'";

    if($branch<>''){
        $sqlbetdate .= " AND d.BRANCH='$branch'";
    }
    if($doctor<>''){
        $sqlbetdate .= " AND d.DOCTOR='$doctor'";
    }
    if($testing<>''){
        $sqlbetdate .= " AND (t.testings LIKE '$testing,%' OR t.testings LIKE '%,$testing,%' OR t.testings LIKE '%,$testing' )";
    }

    //echo $sqlbetdate;
    echo "<table border='1' class='table table-striped'><tr><th>Date</th><th>Branch</th><th>Doctor</th><th>Reference</th><th colspan='9'>Testings</th></tr>";
    $result=mysqli_query($dbhandle,$sqlbetdate);
```

```php
    $result=mysqli_query($dbhandle,$sqlbetdate);

        if(count($result)>0){

        foreach($result as $row){
            echo "<tr><td>";
            echo $row['DATE']."<br/>";
            echo "</td><td>";
            //echo $row['BRANCH']."<br/>";
            $sqlbranch = "SELECT * FROM branch_details WHERE Branch_ID='".$row['BRANCH']."'";
            $resultbranch=mysqli_query($dbhandle,$sqlbranch);
            foreach($resultbranch as $rowbranch){
            echo  $rowbranch['Branch_Name']."<br/>";
            }
            echo "</td><td>";
            //echo $row['DOCTOR']."<br/>";
            $sqldoc = "SELECT * FROM doctors_details WHERE NIC='".$row['DOCTOR']."'";
            $resultdoc=mysqli_query($dbhandle,$sqldoc);
            foreach($resultdoc as $rowdoc){
            echo  $rowdoc['DocName']."<br/>";
            }
            echo "</td><td>";
            echo $row['RefNo']."<br/>";

            echo "</td><td>";
            if($testing<>''){
            $sqltypdate = "SELECT * FROM testings WHERE TestId='$testing'";
            $resulttyp=mysqli_query($dbhandle,$sqltypdate);
            if(count($resulttyp)>0){

            foreach($resulttyp as $rowtyp){
            echo  $rowtyp['TestName']."<br/><hr>";
            echo "</td><td>";
            }}
            }else{
                //echo $row['testings']."<br/>";
                foreach(explode(",",$row['testings']) as $testing_1){
                //echo $testing_1."<br />";
                $sqltypdate = "SELECT * FROM testings WHERE TestId='$testing_1'";
            $resulttyp=mysqli_query($dbhandle,$sqltypdate);
            if(count($resulttyp)>0){

            foreach($resulttyp as $rowtyp){
            echo  $rowtyp['TestName']."<br/>";
            }}
            }

            }
```

Figure F.14: Code for taking reports of testing according to conditions specific period

APPENDIX G: CLIENT CERTIFICATE

Laboratory Report (Confidential)

Heal through a Hope

POINTE
Computerised
Automated
Chemistry Analyser

HOPE DIAGNOSTICS

වෛද්‍ය රසායනාගාර සේවය
மருத்துவ ஆய்வுகூட சேவை
MEDICAL LABORATORY SERVICE

Abbott CELL - DYN
Computerised
Fully Automated
Haematology
Analyser

20 B, Eluwila, Panadura. Tel / Fax : 0385 67 62 67 Hotline : 0777 333 587 E-mail : hopediagnostic@sltnet.lk Website : www.elab.lk

Please visit www.elab.lk for e-reports and m-reports of our online medical laboratory service.

Project Examination Board.
University of Colombo School of Computing
No 17, Swarna Road, Colombo 06.
November 05, 2017

Dear Sir/Madam,

Letter of Certification

Mr. Viraj Prasanna, who is following the final year of Bachelor of Information Technology (BIT) at University of Colombo School of computing (UCSC) has successfully developed a Web Based Report and Invoice Generating System for Hope Diagnostics Medical Laboratory Service to fulfil the final year requirement of BIT degree.
He has successfully analysed the current state of our laboratory and developing us a good system which will fulfil our requirements. Hopefully this system will make our activities efficient and accurate and serve general public in more effective manner.
This Web Based Report and Invoice Generating System automate the data manipulating and help to print blood report easily, thus it facilitates a much advanced and smoother flow in day to day activities of the staff. Finally, I do hereby recommend the Web Based Report and Invoice Generating System for Hope Diagnostics Medical Laboratory Service since it is proven tool for our tasks and I'm satisfied with it.

Yours Sincerely,

.....
Jeewantha Hasaranga,
Medical Laboratory Technologist,
Hope Diagnostics Medical Laboratory Service,
Panadura.

K.J.H. Liyanage
Medical Laboratory Technologist (MLT)
Diploma in Medical Laboratory Technology
S.L.M.C. Reg. No. 1142

Figure G.1: Client Certificate

GLOSSARY

jQuery

jQuery is a JavaScript library designed to simplify the client-side scripting. jQuery is the most popular JavaScript in use today. jQuery is a free, open-source software

mPDF

mPDF is a PHP [7] [8] library which helps to generate PDF files. It is a free library and popular for using to generate server side pdf formats.

Prototype

A prototype is used to demonstrate concepts, test the design options and learn more about the problems and its possible solutions at the initial stage.

INDEX