

# **Smart Computerized Pharmacy Management System**

**I.H.M Kalpa Chathuranga  
2020**



# **Smart Computerized Pharmacy Management System**

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

**I.H.M Kalpa Chathuranga**  
**University of Colombo School of Computing**  
**2020**



## Declaration

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

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

Student Name: I.H.M Kalpa Chathuranga

Registration Number: 2014/MIT/010

Index Number: 14550109

---

Signature:

Date:

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

Mr./Ms. I.H.M Kalpa Chathuranga

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

Certified by:

Supervisor Name: Dr. K.L Jayaratne

---

Signature:

Date:

# List of Content

List of Content .....	ii
List of Figures .....	iv
List of Tables .....	v
List of Abbreviations .....	vi
Chapter 1.....	1
Introduction.....	1
1.1 Overview .....	1
1.2 Project Objective.....	2
1.3 Scope of the project.....	3
Chapter 2.....	7
Review of Similar Systems and Technologies .....	7
2.1 Background .....	7
2.2 Computerized pharmacy management system .....	8
2.3 Literature Review .....	8
Summary .....	14
Chapter 3.....	15
Analysis and Design .....	15
3.1 Disadvantages of manual process .....	15
3.2 Proposed System.....	15
3.3 Requirements Gathering Methods .....	16
3.4 Functional Requirements.....	18
3.5 Nonfunctional Requirements .....	19
3.6 Alternative Solution Evaluation.....	20
3.7 Initial use case Diagram .....	22
3.9 Description of the system users .....	23
3.10 Entity Relationship Diagram .....	24
3.11 Sequence diagrams.....	25
3.12 Activity Diagrams .....	26
3.13 Design Decisions .....	28
3.14 System Interfaces .....	28
Chapter 4.....	32
Implementation.....	32
4.1 Software Background.....	32

4.2 Usage of Design Patterns .....	34
4.3 Development Tools .....	35
4.4 User Interface Design.....	35
4.5 Reusable Codes.....	36
4.6 Program List .....	37
Chapter 5.....	42
Evaluation and Testing .....	42
5.1 Test Plan.....	42
5.2 Test Plan of the System.....	47
5.3 Test Cases.....	48
5.4 User Experience Evaluation .....	49
Chapter 6.....	52
Conclusion and Future Work .....	52
6.1 Project evaluation .....	52
6.2 Problems Encountered.....	53
6.3 Lessons Learned .....	54
6.4 Achievements .....	54
6.5 Future Developments .....	55
References.....	57
Appendix A – System Configuration Documentation.....	58
Appendix B - Use case narratives .....	60
Appendix C – Test cases.....	66
Appendix D: User Evaluation Form.....	78

## List of Figures

Figure 1.1 - Smart Computerized Pharmacy Management System.....	02
Figure 1.2 - Module of the proposed system.....	04
Figure 3.1 - Use case diagram.....	22
Figure 3.2 – Entity Relationship diagram .....	25
Figure 3.3 - Sequence diagram for user login .....	26
Figure 3.4 - Activity diagram for user login .....	27
Figure 3.5 - User Login Interface .....	29
Figure 3.6 - Add User Interface.....	30
Figure 3.7 - Add Medicine Interface.....	31
Figure 3.8 - List of Medicine Interface.....	31
Figure 4.1- Input validation for Add medicine inventory form.....	36
Figure 4.2- DataContext.cs .....	37
Figure 4.3 - Add user model and model validations.....	38
Figure 4.4 - Add user view.....	39
Figure 4.5 - Add user controller.....	40
Figure 4.6 - _Layout.cshtml.....	41
Figure 5.1 - Overall user satisfaction level .....	50
Figure 5.2 - User satisfaction levels of system performance.....	51
Figure 5.3 - User friendliness vs overall user satisfaction .....	51

## List of Tables

Table 3.1 - Use case narrative for Administrator login .....	23
Table 3.2 - System Privilege .....	28
Table 4.1 - Recommended Hardware Requirements.....	33
Table 4.2 - Recommended Software Requirements.....	33
Table 5.1 - Unit test for User login.....	43
Table 5.2 - Unit test for add new medicine Brand.....	44
Table 5.3 - Test plan .....	47
Table 5.4 - Test case for Administrator log in .....	48
Table 5.5 - Test case for Create supplier orders .....	49
Table A.1 - Recommended Hardware Requirements.....	58
Table A.2 - Recommended Software Requirements.....	58
Table B.1 - Use case narrative for administrator log in .....	60
Table B.2 - Use case narrative for Add user to the System .....	60
Table B.3 - Use case narrative for create supplier order.....	61
Table B.4 - Use case narrative for Generate reports.....	62
Table B.5 - Use case narrative for manage medicine.....	62
Table B.6 - Use case narrative for manage pharmacy items .....	63
Table B.7 - Use case narrative for manage suppliers .....	64
Table C.1 - Test case for User Login .....	66
Table C.2 - Test case Add new medicine brand .....	67
Table C.3 - Test case Administrator Login.....	67
Table C.4 - Test case for Create Supplier Order .....	68
Table C.5 - Test case for Add users by Administrator .....	69
Table C.6 - Test case for Edit users by Administrator .....	71
Table C.7 - Test case Add supplier by Administrator.....	72
Table C.8 - Test case for Add New Invoice.....	74
Table C.9 - Test case for Add Medicine Inventory .....	75
Table C.10 - Test case for Add Customer.....	76

## List of Abbreviations

ASP	Active Server Pages
ER	Entity Relationship
My SQL	My Structured Query Language
OS	Operating System
RAM	Random Access Memory
HDD	Hard Disk Drive
MVC	Model-View-Controller
SMS	Short Message Service
UML	Unified Modeling Language
CSS	Cascading Style Sheet
HTML	Hypertext Markup Language
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphics
PDF	portable document format
SVG	Scalable Vector Graphics



## **Abstract**

In typical pharmacy use paper-based inventory management, Supplier management and sales management modules even they use some point of sales system to manage the customer sales. Most of the point of sales processes only provide a solution to track the sales and create sales invoices. An alternative method can be used to maximize the profit by automating the routine manual process. To do this web based or standalone application can be used. Since most of the process are interconnected with each other web-based solution has the edge when it comes to automatic process management

The purpose of this project is to develop a web based Smart computerized pharmacy management system for Gamage Pharmacy which is in Homagama. Currently they are handling daily routine work of the pharmacy by using manual processes.

The proposed pharmacy management system will improve the manual process done by pharmacy staff. The system will have inventory management system to manage their stocks and keep track of the inventory and make suggestion to order stocks automatically. This process will significantly improve their current manual ordering process which includes of writing the orders by going through stock balances manually and make rough estimate of the amount to order based on stock figures. Drugs are not supposed to use after the expiration period. This makes the manual tracking and ordering the stocks more tiresome process which the system will automatically detect and inform the users. The system will replace manual paper-based pharmacy management process with computerized management process.

The system was developed on ASP.Net Core Framework using C# as the computer language. The development process will follow Model-View-Controller architecture. Interfaces were designed using existing html templates by using JavaScript, jQuery and CSS. The system runs in IIS web server and MySQL used as the database in windows environment.

## **Acknowledgment**

First, I would like to thank the Directors and all the staff members of the University Of Colombo School Of Computing for giving us all the support and all the studding materials.

This dissertation could not have been written without help from Dr. K.L Jayaratne, my supervisor in charge who helped me with various aspects. So, I would like to extend my gratitude to Dr. Jayaratne for all the help he provided during the project.

I would like to extend my gratitude to Mr. Suranga Rasika Gamage, who is the owner of Gamage Pharmacy Homagama who help and support during the requirement gathering and providing me all the details I needed to get a clear understanding about the pharmacy related processes. Also, I need to extend my gratitude to the staff members of Gamage pharmacy who spent time for providing me with required details. And I would like to offer my gratitude for all those who supported me in many ways during the completion of the project. Finally, I would like to convey my heartfelt gratitude to my family and friends who encouraged me to complete this project successfully.

# Chapter 1

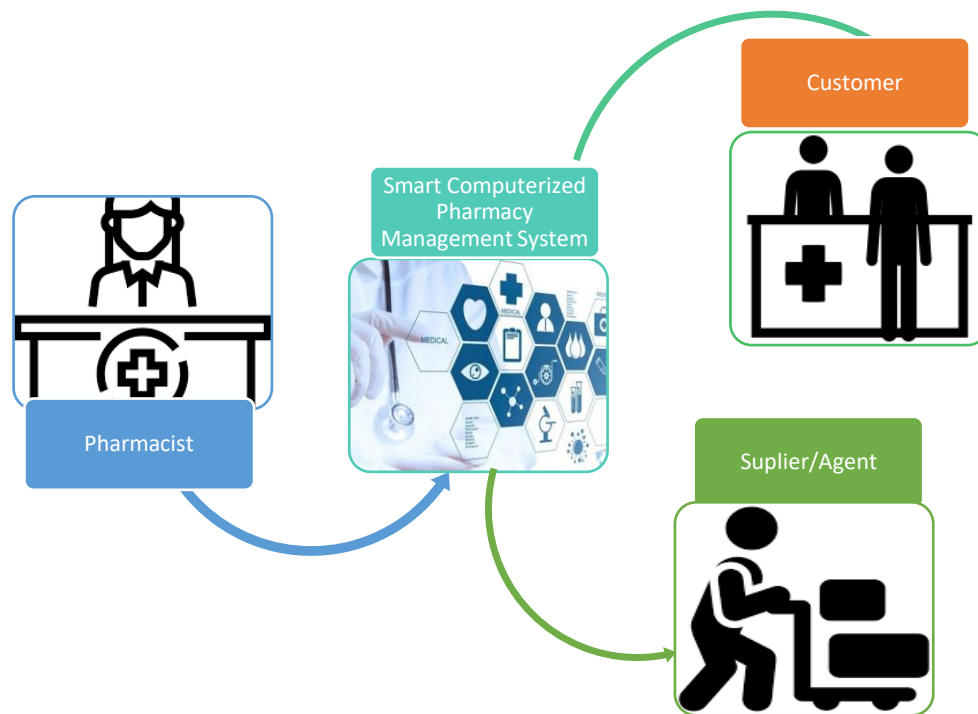
## Introduction

### 1.1 Overview

Gamage Pharmacy is a pharmacy located in Homagama. They are currently handling daily routine of their pharmacy and managing daily processes manually. When it comes to the pharmacy management keeping track of inventories with regards to the drugs in the store, quantity of drugs available, the expiry dates and their usage manually is a tedious and difficult task. This project is based on automating the above-mentioned processes. The pharmacist has to manually order drugs to restock the already reducing stocks. Since the pharmacist cannot order their supplies from there vendors online now, All the supplier orders done manually via agents. They do not have an inventory management system to manage their stocks and they must keep track of the inventory and order stocks manually. This ordering process takes significant amount which includes of writing the orders by going through the stock balances manually and make rough estimate of the amount to order based on stock figures. Drugs are not supposed to use after the expiration period. This makes the manual tracking and ordering the stocks more tiresome process. They have some employees working there and they do not have an employee management system to keep track of their employee details. Pharmacy Management with manual paper record is a hassle in this era.

Since above mentioned tasks handled by pharmacist manually it is hectic for pharmacist to conduct above manual tasks every day. So, in my project I am going to address above mentioned main issues when it comes to manual work. The objective of this project is to reduce manual work and automate most of these manual processes.

## 1.2 Project Objective



*Figure 1.1 Smart Computerized Pharmacy Management System*

The objective is to develop a solution which will automate manual process done by Pharmacist and improve the timing and accuracy of current processes. Proposed system will keep track of stock related information and analyze those data to identify fast moving medicines, the availability of the stocks and making suggestions based on medicine sales to improve the efficiency of sales process.

One of the main objectives is to find out which manual tasks are mostly used, and which areas need more focus when it comes to normal pharmacy related daily routine work. So, that pharmacist can focus more on improving the sales and gain more customer satisfactions instead of wasting time on work which can be automatically done. This will increase efficiency of the ordering and sales processes as well as reduce the maintenance cost and wastage. This will give more control to pharmacist so that they can utilize their time effectively.

This solution is mainly focused on a manual process now and the objective is to make that process automatic. When it comes to stock keeping historical data is important to identify future stock requirements and actions that needs to be taken to optimize the ordering process. The proposed system will also act as a central stock management system for pharmacist to monitor and analyze existing stocks to make future orders in more robust manner.

Following are the brief of main objectives,

- Provide an autonomous smart computerized solution which can be used to increase the efficiency of day to day stock and sales management process and increase availability of customer requirements.
- Provide a solution to reduce maintenance cost and time of unwanted Manual processes.
- Provide a solution to manage pharmacist related daily work
- Provide a stocks and sales history tracking and analyzing system which can provide predictions based on stock counts and sales
- Design a solution which can speed up day to day work of pharmacist

### 1.3 Scope of the project

The scope can be broadened with the time. Since my initial intention is to automate manual processes which handles by pharmacist which includes in daily work processes.

#### Addressing the time constraints

When considering time constraints, I should produce a solution so that pharmacy staff can reduce the day to day manual work time using a solution like this.

When doing day to day work pharmacy staff spend lots of time on manual tasks like keeping track of stocks, sales records, expiration details of sales items and catering customers at the same time. Monitoring and analyzing those kinds of processes automatically is very valuable for pharmacists. So, I will develop a solution to monitor and analyze stocks and sales and make proactive suggestions to improve the ordering and sales process

#### Addressing the performance constraint

In a pharmacy, there can be so many variables and factors that hinders the efficiency. Daily routine tasks which can be affect the sales as well as the customer satisfactions. Aim is to find out manual process which takes lot of effort.

By using this system, I will try to monitor sales and stock related areas and generate automatic alerts and status mails/messaged about a specific area and send them to relevant people.

Modules of the proposed software solution/Service,

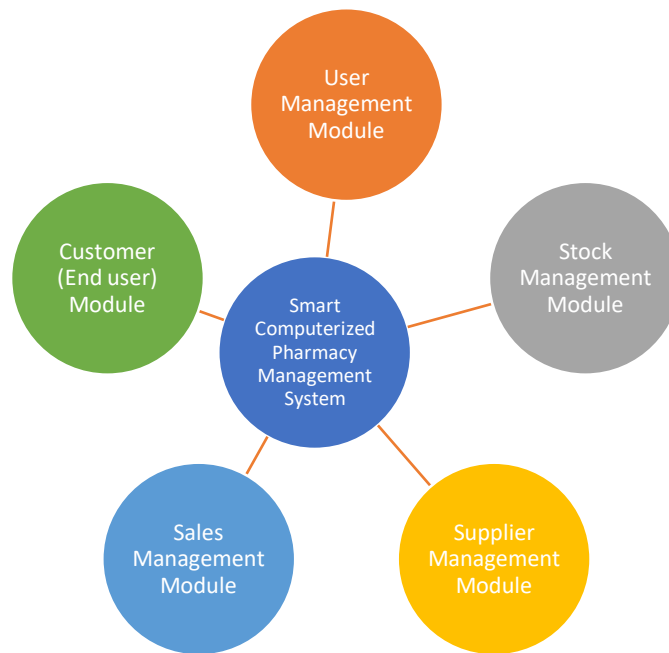


Figure 1.2: Module of the proposed system

Smart Computerized Pharmacy Management system will consist with 5 major modules. Stock management, Supplier management and Sales management will include a functionality to gather data and analyse them to generate proactive predictions.

- User Management Module

This module provides the functionality for pharmacist to carry out day to day work. Users of the system, namely pharmacist and pharmacy staff must be provided the following functionality:

- Create users.
- Manage users.
- Assign security roles to users
- User Access control

- Stock Management Module

This module provides the functionality for pharmacist to carry out stock related work. This function helps pharmacist to identify lack of stocks as well as the soon to be expired stocks. The function is intelligent enough to automatically check the current stock and make predictions on the historical data and provide pharmacist suggestion to manage the stocks in

more robust way. Users of the system, namely pharmacist and pharmacy staff must be provided the following functionality:

- Add, edit or remove medicine
- Add, edit or remove stocks
- Analyse stock and suggestion making
- Automatic Notifications
  - Pharmacist notification
  - Agent notification
- Report Generations
  - History report
  - Analytic reports
  - Summary reports
  
- Supplier Management Module

This module provides the functionality for pharmacist to carry out supplier related information. This will facilitate the system user to search most suitable suppliers and make orders by analysing historical data based on supply time, quantity and quality.

- Create supplier account
- Manage supplier Accounts
- Generate supplier invoice
- Report Generations
  - Supplier order report generations
- Analyse Supplier data and predict suggestions
  
- Sales Management Module

This module provides the functionality for pharmacist to carry out sales related information. This will facilitate the system user to search most suitable suppliers and make orders by analysing historical data based on supply time, quantity and quality.

- Add, Edit or Remove Orders
- Sales Analyzation
- Provide sales improvement suggestions

- Manage customer Feedback
- Customer (End user) Module

This module provides the functionality for customers to register the in this web-based solution and place orders for over-the-counter medicine or other non-pharmaceutical sales items.

- Create user account
- Edit user account
- Search catalogue
- Place Orders
- Check order status
- Give Feedback



## Chapter 2

### Review of Similar Systems and Technologies

This chapter gives a brief introduction about the project background. Here describes about traditional pharmacy management methodologies and existing alternate pharmacy management systems that used.

#### 2.1 Background

At present, many pharmacies have chosen to focus on efficient stock management and convenient service to customers while offering a good customer satisfaction by providing quality products with high availability. Until recently, most of pharmacy related tasks manually executed by pharmacists, but there are many disadvantages to this process, including the inconvenience of the pharmacist needing to keep manual stock counts, manual stock analyzations and customers must wait longer time to get the medicine they ordered. Pharmacist have the utmost responsibility

Pharmacy services has become a mandatory industry of a medical systems these days. Delivering medicine to customers is not the only pharmacy related activities that a pharmacy management system automates. It will automate most of the crucial activities of pharmacists starting from stock management to customer purchase management. The web-based solution is very convenient for pharmacist because they have visibility into the items in hand, price and extremely simplified navigation for inventory management.

Web based Pharmacy management systems are used by many pharmacies all over the world. Some of those systems are GOFRUGAL (*Gofrugal, 2020*), Meditab Intelligent medical Software (*Meditab, 2020*), CoverMyMeds platform (*Covermymeds, 2020*), RxMaster Pharmacy System (*Rx Master, 2020*) etc. Online medication ordering is possible via mobile applications also now a days and it is the new trend. Few of the above-mentioned systems also having mobile applications. Some of the core features in most of the pharmacy management systems are discussed in literature review section under automated computerized pharmacy management section.

## 2.2 Computerized pharmacy management system

A pharmacy management system is referred to as a set of detail methods that is being used in handling the pharmacy related activities such as invoice entry, inventory management, expiry management, sales billing, reordering, scheduled medicine index. All the above activities can be computerized or done manually. By computerizing these processes, it helps the pharmacist to run their businesses smoothly.

In general pharmacy management systems consists of data entry, data retrieval and inventory monitoring options, monitoring and tracking medicine dispensing patterns and generating reports based on above functionalities. The Pharmacy management system can be defined as a computerized system that is being used by pharmacist to execute their daily activities efficiently and allow them to monitor inventory automatically. The system will automatically analyze the inventory using the usage patterns to make suggestion for ordering medications beforehand.

Automated pharmacy management in Pharmaceutical industry refers to the pharmacy related daily routine activities automatically manage through applying various types of technologies such as internet and many others. Pharmacy Management Systems are successful when it is applied at pharmacies in many other countries. The usage of the automated management technology is proven to benefit most of the investors.

## 2.3 Literature Review

In the pass decades, the rapid growth of automatic systems and computer technology did a great impact for how people mange certain business-related activities. At present, this technology also leads different kind industries to change their entire management aspect. Pharmaceutical industry is one of the industries in the market that apply these technologies into their business processes that assist them to be much more convenience and efficient.

Various systems were established in market for the purpose of Pharmacy management. Some of the existing systems are explained below.

### **Traditional Pharmacy management systems**

Traditional pharmacy management systems used in most small and medium scale pharmacies. The process includes manual medicine stock ordering and purchasing, inventory control, expiry management and sales billing. One of the most widely used traditional pharmacy management system is the conventional paper-based system. In this system all records are stored on paper. The main drawback of this system is papers can get easily lost or damaged. There is also wastage of money, time and paper. Paper-based systems do not provide any form of dynamicity. Even a small change requires the re-print of entire inventory lists. Also, large amount of human efforts are required, this system is not work properly because it has some error and from a pharmacists point of view it is time consuming.

### **Automated Computerized Pharmacy Management Systems**

In order to reduce service cost and enhance user experiences, some of the pharmacies have invested in the automated pharmacy managements systems. The pharmacy management system used to perform varies activities from inventory management to sales billing. After purchasing stocks, the pharmacist should insert the new stock data into the system in system where PC was set up. The system will manage all the inventory management activities automatically. When Customer purchase medicine the inventory automatically updates, and the data stored for further analyzation of stocks and make reminders for new stock orderings. The system was also informing the pharmacist about the availability of the medicines. If a certain brand of medicine was not available, then pharmacist will be notified by the system. After serving the medicine order, bill was generated at the cash counter as per customer order. The management had full authority to access all details of the customer which are fed into the system.

With the improvement in the computer and communication technology, various systems were launched in market for the purpose of computerization of the Pharmacy management systems.

Computerized Pharmacy Management Systems mainly focus on inventory store operations and automatic management of the inventory flow with suppliers and customers. When it comes to supplier end computerized system covers some main operations. Some of those operations are receiving information from suppliers about the new medicine arrivals, alert suppliers to replenish the decreasing stocks, placing online orders and returning expired medicine stocks.

Pharmacy Management Systems use pre fed data to monitor and control the inventory, automatically generate alerts for minimum quantity of each drug when the stocks reach minimal levels.

At present there are two different categories of pharmacy managements systems available in the market. Those are inpatient pharmacy management systems and outpatient pharmacy management systems. The functionalities provided by these systems are varied from one to another. In most pharmacy management systems, there are some core features and functionalities which is needed to perform the pharmacist's duties. Some of the core functionalities are,

### **Dispensing process and prescription management**

In a Pharmacy the most associated process is dispensing of the medicine. Dispensing process starts when the pharmacist receives the prescription from the patient(inpatient/outpatient) or prescriber. This process divides into few different stages,

Data Entry stages - Pharmacist or technician entered the details of the received prescription to the system.

Pre check stage – Before feed the data into the system pharmacist review the prescribed have potential drug interactions, appropriate dosage or transcription errors.

Fill Stage: After counting and verified by hand Medication are dispensed during this stage.

Checking Stage: Pharmacist confirmed that the filled prescription has the right medication and dosage instructions.

### **Clinical information management**

Pharmacies use Patient basic information like age, address, phone number, allergies. Also they record known health conditions, prescriber information and other necessary details which are related to patient care management.

The pharmacies also keep records of patient's previous medication records as historical data so they can avoid negative drug interactions.

For the patients who takes multiple medication for different illnesses, pharmacists should organize the medication on a respective schedule reducing the medicine interactions by synchronizing the medication.

### **Inventory Management**

There are many systems that provides inventory management features. These systems help pharmacist to rearrange items, return unused stocks and at the same time it helps to keep a clean inventory which will ultimately save money on medicine products which are not moves form the shelves. Some pharmacy management system supports electronic data exchange (EDI) between pharmacies and suppliers to circulate the information on fast moving medicines and the price changes.

### **Usage-Based Ordering**

There are systems that facilitate the ordering process of medication by automatically monitoring the stock counts are lower for a respective medication based on the dispensing patterns.

### **Pricing and Billing**

For any business financial management is crucial for that business. It is same with Pharmacy industry as well. The pharmacy management systems should have the capability to identify and manage the finance flow of the business by providing real time feedback to the owners.

### **Reporting**

Pharmacy management systems facilitate report generations using historical data to provide during audit processes, documenting patient details and these reports can also be utilized in business intelligence practices and decision makings.

Most of these systems specialized to manage medicines. But when it comes to retail pharmacies are selling other retails items. So, a customized pharmacy management system should be able to manage other item inventories while mainly focusing the medicine dispensing. Also, the system should be robust and reliable. At the same time, it should be easier to use for a daily user. To overcome the limitations of currently available systems, solution is a smart online

pharmacy management system using web-based application. The main objective of this project is to develop a smart online pharmacy management system which can be used in any pharmacy.

The entire system can be cost effective and significant amounts of time and money can be saved by implementing it. Human errors in billing and delivering medication would be totally eliminated, the wastage will be minimized due to smart inventory handling and ordering process, the upcoming trends in the market can be identified beforehand and orders can be placed to replenish the stocks in advance which will ultimately increase the sales and the revenue collection.

After searching throughout the internet and reading some publications, the following systems have similar functionalities to the proposed system,

### **GOFRUGAL Pharmacy Management System**

GoFrugal point of sales system is a hybrid solution that helps retailers across various industries. One of them is GoFrugal's pharmacy management software which can be used in single & chain stores which can provide information for strategic decision-making. The system can be deployed on premises in windows platform. This system has many features and Some of those features are listed below,

- Cash Management
- Point of Sale (POS)
- Billing & Invoicing
- Billing Management
- MIS Reports
- Alerts/Notifications
- Document Printing
- Barcode Generation
- Transaction management
- Discount system
- Automatic Blocking of Expired Items
- Expiry and Near Expiry

### **RxMaster Pharmacy System**

RxMaster pharmacy software provides pharmacy management system with many features including pharmacy business management tools and pharmacy point-of-sale. Some of those features are listed below,

- Workflow
- Pharmacy Inventory Control
- Documentation and Billing
- Prescription Pricing
- Prescriber Validation and Verification
- Data Management Services
- Data Analytics/Business Intelligence
- Barcode Scanning
- Long-Term Care Software
- Revenue Accounting
- Prescription Pricing
- Prescription Synchronization

### **CoverMyMeds Platform**

CoverMyMeds is a fully integrated software designed for pharmacists to make day-to-day workings of their pharmacies. It is a web-based solution that enables pharmacists and prescribers to manage the provision of medications and healthcare plans for patients which provides electronic prior authorization functionality. This system has many features starting from inventory management to expense control. Some of those features are,

- Document Management
- Inventory Management
- Billing & Invoicing
- E-Prescribing
- Patient Portal
- Management Information System Reports
- Records Management
- Expense management
- Prescription Management

- Multi-Store Management
- Insurance Processing
- Prescription Processing

When it comes to pharmacy management solutions the inventory management and point of sales management is the main areas most focus was given. The proposed system was based on the same focus area and improve that focus area to provide a solution to small to medium scale pharmacies. To design a system which provides more value to small and medium scale pharmacies it is crucial to focus on minimizing the wastage of medication by monitoring the inventory automatically. In the same time, the proposed system should automate daily pharmacy related manual work which this project was based on.

## Summary

Proposed system is about automating the manual process of the pharmacy and provide efficient and effective system to manage the pharmacy related actives. This is a web-based system, and this is applicable in any pharmacy.



## Chapter 3

### Analysis and Design

Chapter three briefly describes the disadvantages of manual processes and advantages about proposed system over the main functions that addressed by it. It also includes a description about nonfunctional requirements, hardware and software requirements. Also, it describes the methods and tools that used to develop the proposed system. Furthermore, it represents primary UML diagram techniques and description of the system users as well as system reports that generated for above mentioned users.

#### 3.1 Disadvantages of manual process

- Pharmacist must spend more time on maintaining the medicine stocks
- Medicine stocks get expired due to lack of management
- High waiting time of customer and decrease customer satisfaction.
- High labor cost.
- Less accurate data.
- Difficulty in using the data for data mining and decision making.
- High paper-based log keeping.
- Lesser security of information.
- No automatic report generation.

#### 3.2 Proposed System

Propose system is a Smart Computerized Pharmacy Management System for Gamage Pharmacy (a web-based system). But this system is applicable in any pharmacy related shop. Main advantage of this system is that it greatly simplifies the pharmacy management processes for both the pharmacist and the owner. When pharmacist login to this system, he/she is presented with an interactive and up-to-date pharmacy related data, complete with all available options to carry out his/her day to day work and dynamically notifying the pharmacist regarding the new trends and stock related data based on the selected options. This provides instant interactive visual based feedback of the current situation of the pharmacy stocks and

ensures that items which needs to order. Also, pharmacist can generate reports using this system if they need to make important decisions related to pharmacy management.

Benefits:

- Minimize the number of manual tasks of the pharmacist.
- Minimize manual paper-based logging.
- Reduce labor cost involved.
- Minimize miscalculations, since it is a smart computerized system.
- Increase efficiency and effectiveness of the pharmacy staff.
- Only authorized users have access to the relevant information.
- Data and information Security.
- Support business decision making process.
- Regulate privacy and confidentiality and stop improper disclosure of information.

### 3.3 Requirements Gathering Methods

For requirement gathering process various approaches available for requirement gathering such as Interviews, Questionnaires, User Observation and Analyzing Existing Documents. Methods used for requirement gathering of proposed system are explained below.

Interviews

There are two type of interviews: individual and group interviews. Individual (One-on-one) interviews are the most common technique for collecting requirements and it is one of the main techniques of gathering requirements. The person who gather requirements should identify the stakeholders that needs to be interviewed. The stakeholders can be users who interact with the current or new system, management staff or other people involved in the system.

In comparison to individual interviews, Group interviews are similar in most ways, except that more than one person is interviewed. To get optimal results from Group interviews the participants should be on the same level or position. Any participant can suggest an idea that is overlooked by other members of the group which can lead to more thoughts and discussions and thus provide more information about relevant topic. The interviewer can evaluate which topics are more relevant for the system design and check weather if it is generally agreed or generally differ. The major disadvantage of this is the planning phase. Since there can be more

than one person involved in these interviews, the process can be time consuming. To gather requirements the proposed system individual interviews conducted with pharmacy staff in Gamage pharmacy to get the information regarding current manual system. Group interviews conducted with main pharmacist and the owners of the pharmacy to gather requirements from the administration level. Non-functional requirements and information on current manual process was gathered by these interviews. Furthermore, some of the suppliers has been interviewed to get more detailed information on supplier management and handling process of the proposed system.

### User Observation

The analyst may observe the current users in their working environment to get better understanding of the manual processes. Observing users is helpful for the analyst to get a full grasp of the user's interactions with the current system. It is found that sometime the current users may find difficult in explaining what exactly their regular tasks consist of and their requirements may be differ from the initial descriptions of the problems. User observation helps in cases like these which will assist analyst by providing a clear picture of requirements. By user observation analysts can validate data that had been previously collected.

For the proposed system, user observation has been done for the pharmacy staff of Gamage pharmacy to obtain a clear and better understanding of the current manual processes. Also user observation helped to gather more detailed information for the proposed system. During the user observation phase more useful functionalities and features was identified which should be available in the proposed system for the pharmacy staff.

### Analyzing Existing Documents

Existing document analyzation is a valuable technique in requirement gathering. It can support other techniques by providing vital information. Observing the current processes and any documentation helps the designer or analyst to understand the business, or the situation of the current manual system. By referring the documents, the system users who interact with the system can be identified. Which helps the analyst to prepare questions for detailed discussions or questionnaires prepared for relevant stakeholders, in order to gather additional details and requirements.

Documents related to current manual process such as inventory receipts, payment receipts, order receipts, documents related to order details etc. analyzed to gather information and requirements for the proposed system.

### 3.4 Functional Requirements

The structure of the system can be divided into five main logical components:

- User Management Module - Manage Systems users by adding, modifying and removing users, Granting permission levels.
- Stock Management Module – Manage stocks related information, generate reports and Notifications, automatically monitor stocks to identify stock conditions
- Supplier Management Module – Manage Supplier information, compare supplier prices, notify suppliers on supply requests, provide supplier reports.
- Sales Management Module – Manager sales related information, provide sales reports, upcoming sales trends and notify daily sales reports to owner
- Customer (End user) Module – Manage customer data, provide customer medication history, provide ordering functionality on over the counter medicine.

#### **User Management Module**

This module provides the functionality for pharmacist to carry out day to day work. Users of the system, namely pharmacist and pharmacy staff must be provided the following functionality:

- Create users.
- Manage users.
- Assign security roles to users
- User Access control

#### **Stock Management Module**

This module provides the functionality for pharmacist to carry out stock related work. This function helps pharmacist to identify lack of stocks as well as the soon to be expired stocks. The function is intelligent enough to automatically check the current stock and make predictions on the historical data and provide pharmacist suggestion to manage the stocks in

more robust way. Users of the system, namely pharmacist and pharmacy staff must be provided the following functionality:

- Add, edit or remove medicine
- Add, edit or remove stocks
- Analyze stock and suggestion making
- Automatic Notifications
  - Pharmacist notification
  - Agent notification
- Report Generations
  - History report
  - Analytic reports
  - Summary reports

### **Supplier Management Module**

This module provides the functionality for pharmacist to carry out supplier related information. This will facilitate the system user to search most suitable suppliers and make orders by analyzing historical data based on supply time, quantity and quality.

- Create supplier account
- Manage supplier Accounts
- Generate supplier invoice
- Report Generations
- Supplier order report generations
- Analyze Supplier data and predict suggestions

## **3.5 Nonfunctional Requirements**

### **Usability**

The proposed system should address the requirements of the pharmacy staff and it should facilitate all the above-mentioned functional requirements without hindering the regular work processes. It should access relevant information and produce relevant statistics about the all the covered functionalities. System should make better performance at peak time.

### Availability

The proposed system is a web based solution, which is hosted in web server .It is an online solution so any user can logged into the system using a computer anywhere in the world at any time, and the database backups should be kept periodically depend on the owner requirement and system should secure confidential data.

### Access Control

The proposed system must provide proper access control mechanism for system users and administrators. Administrators should be able to manage all the user accounts. Only Administrators should have proper way to access administrative information.

### Modifiable

In the proposed system authorized personals can add data, modify data or remove data. Also access administrative data without any issues. Other users depend on their security level will be restricted doing any additional operations which are not authorized for their security level.

### Platform compatibility

The proposed system is a web-based solution which can be access through any web browser software. Since it is a web-based solution it is not platform dependent and operable in any operating system.

### Flexibility

The proposed system must be flexible so that it can cater any kind of pharmacy related manual task and any kind of workload. The system should be able to send notifications based on analyzed data and handle supplier orders. The system should be extendable to cater any future requirement.

## 3.6 Alternative Solution Evaluation

Develop using ASP.NET technology

ASP.NET can be used to develop proposed system. .NET is a developer platform made up of tools, programming languages, and libraries for building many different types of applications. ASP.Net is a strong and powerful framework for building web apps and services with .NET and C# programming language. ASP.NET MVC is an open source web development framework from Microsoft that provides a Model View Controller architecture.

ASP.NET MVC is a .NET implementation model-view controller concept which offers an alternative to ASP.net web forms for building web applications. It is a part of the .Net platform for building, deploying and running web apps. You can develop web apps and website with the help of HTML, CSS, jQuery, JavaScript, etc. In ASP.NET MVC developers can work in separate layers for data handling, presentation and business logic. It will result in much more web-style development, with cleaner separation of code and cleaner output

But when it comes to drawbacks development tools for ASP.NET are not available for free as well. Also, it has higher write and read performance in MS SQL server rather than oracle database. ASP.NET limits customizing interfaces (limits developing own JavaScript solutions, limits advanced designs, etc.) and due to this it produces very unmanageable HTML output. After if has been generated it is highly difficult to change even if someone want to work with it.

### Develop as a Desktop Application

The proposed system needs to be shared with different kind of users which can be located in same or different places. To cater that another option was building a desktop application that all the users can users can connect through a network

Most of the time desktop applications remain reliable and trustworthy. It is because they are not connected to any networks and restricted to a physical location. Since they are restricted to a physical location hence there is usability constraint on desktop application. Desktop applications do not require an Internet connection. The main drawback of desktop application is that it cannot be accessed without a computer with already installed aforesaid desktop application. So desktop applications are less exposed to security risks than web applications. The admin user has full control over the desktop applications so it can be protected from numerous vulnerabilities.

For a visible disadvantage desktop application are to be installed separately on each and every computer whereas web applications are hosted in one place for anyone to access from any computer.

After a thorough consideration on the usable solutions for proposed system development, web application approach was selected considering the usability and accessibility aspects. Users will be able to access the web application via internet and do the work without any hassle. The Proposed system will develop using ASP.NET MVC, Bootstrap and Angular because it is more reliable in deployment.

### 3.7 Initial use case Diagram



Figure 3.1: Use case diagram



### 3.8.1 Use case narratives

Table 3.1 shows the use case description for Administrator log in to the system.

Table 3.1: Use case narrative for Administrator login

<b>ID: U01</b>	<b>Actors: Administrator</b> <b>Title: Log in</b>
Description:	Relevant Administrator comes to log in page Logged in using Username and password.
Preconditions:	Administrator has a valid username and password. User should be in login page
Main Flows:	Enter a valid username. Enter valid password. User log in to the system.
Post Conditions:	User Log in successfully and redirect to Main page
Alternative paths:	User enters invalid username then an incorrect username message will be displayed. User enters an invalid password then incorrect password message will be displayed. User enters only username then please insert valid password message will be displayed. User enters only password then username password mismatch message will be displayed. Administrator should re-enter correct log in details or terminate the process.

Refer Appendix B for more use case narratives.

### 3.9 Description of the system users

#### Pharmacist:

This is one of the main users of the proposed system. This user will use the proposed system to manage the pharmacy related tasks. Order stocks, analyze stocks, manage medicine information etc.

#### Main functions

- Manage Inventory

- Manage Stocks
- Make payments
- Manage/View Medicine
- Generate Bills
- Analyze stocks
- Generate Reports

Customer:

Persons who is the third party who use this system to order items from the pharmacy.

Main functions

- Search items
- Order items
- Make Payments

Administrator:

Persons who have authority to manage user accounts and access permissions.

Main functions

- All administrator related functions
- Manage users accounts
- Manage system access permissions

### 3.10 Entity Relationship Diagram

Object Oriented Programming methodology will be used to develop the proposed system. So, entity relationship diagram drawn to identify all the required database tables and connections to get a clear understanding about the proposed system. Figure 3 will describe the initial entity relationship diagram

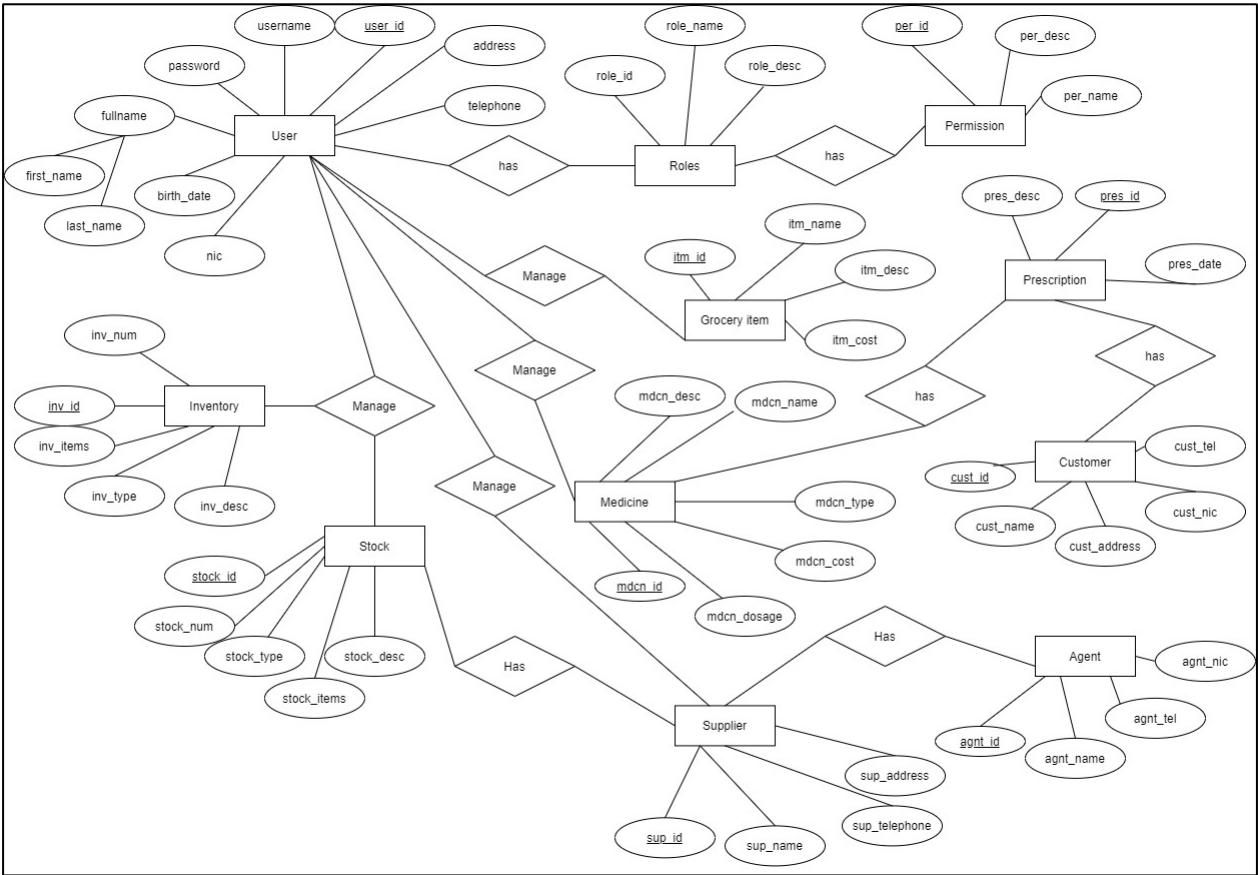


Figure 3.2: Entity Relation diagram

### 3.11 Sequence diagrams

Following diagram (figure 4) shows the sequence diagrams for login user

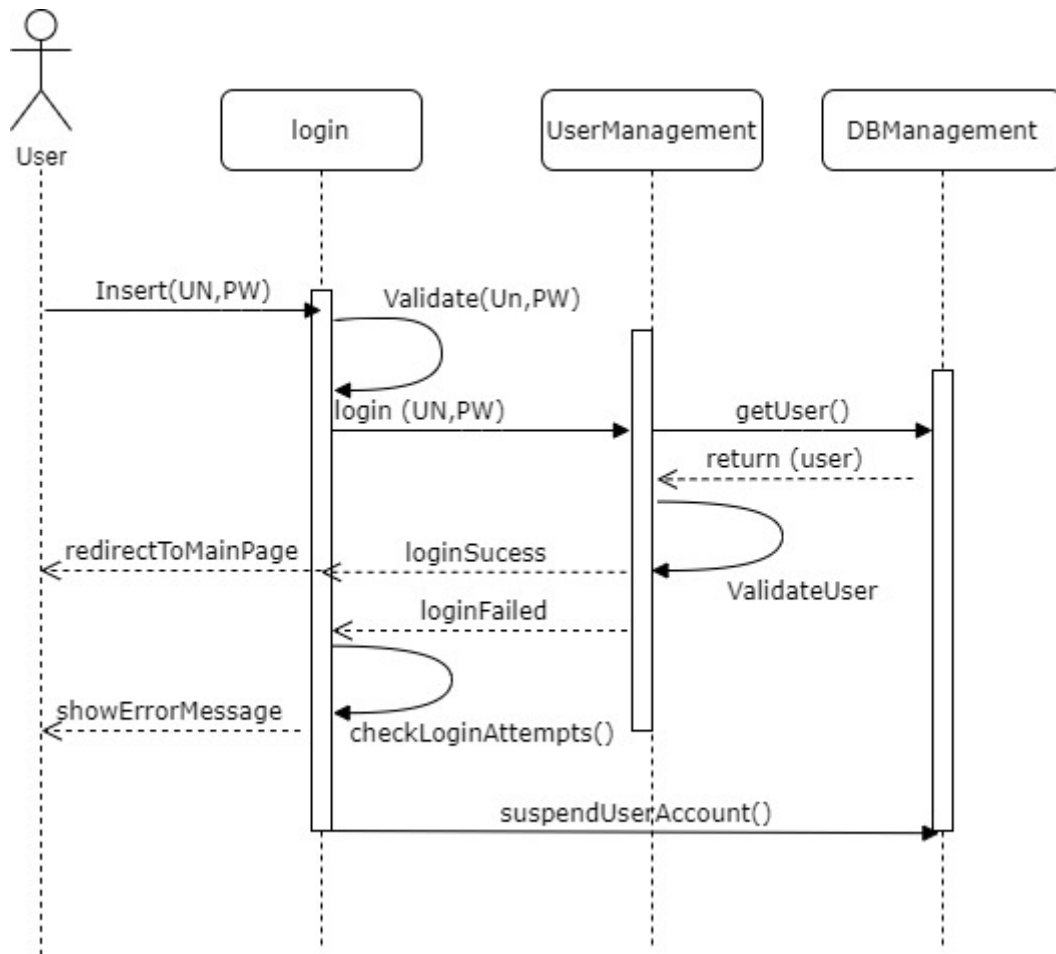


Figure 3.3: Sequence diagram for user login

### 3.12 Activity Diagrams

The following Activity diagram (Figure 5) describes functionality that occurred when a user try to log-in.

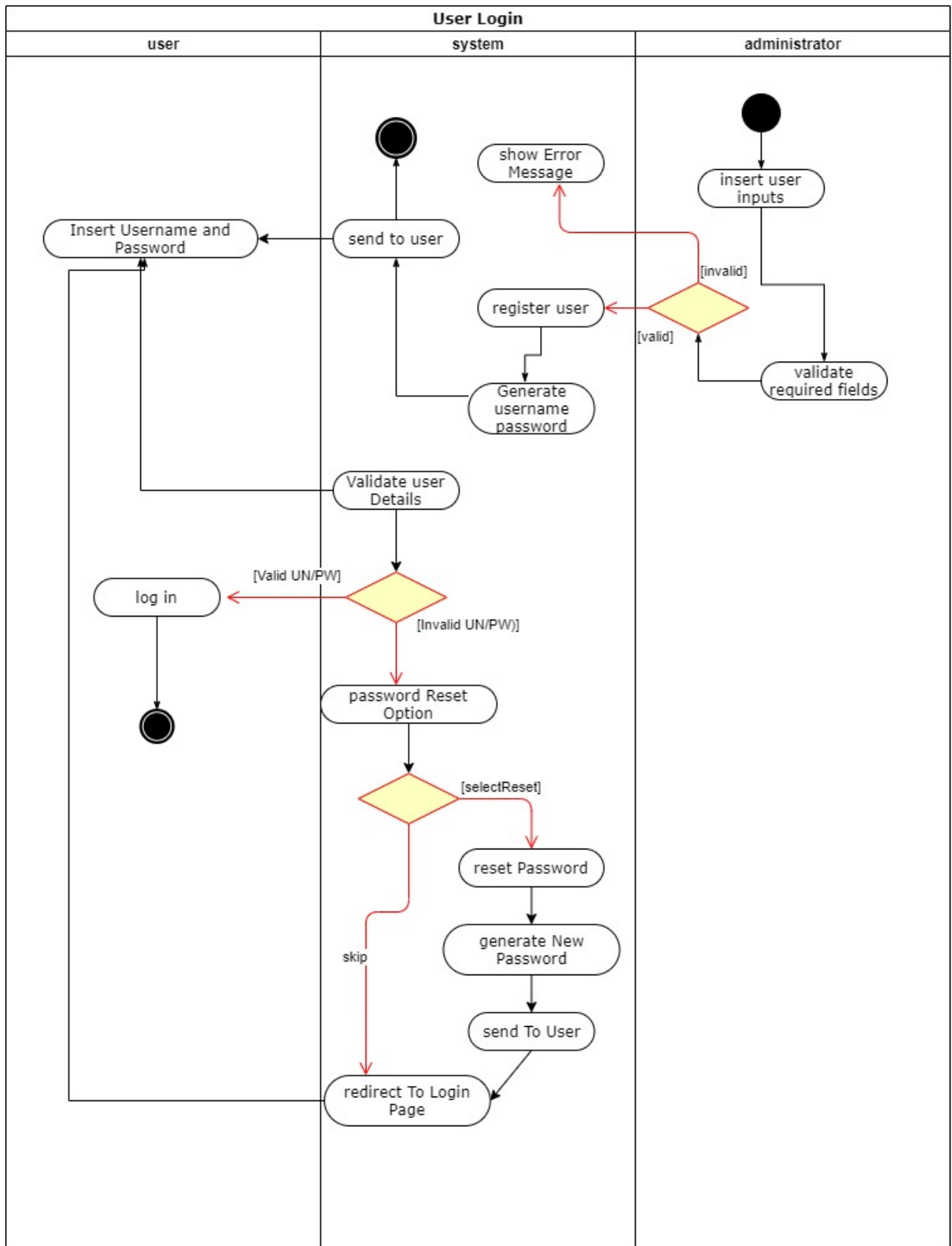


Figure 3.4: Activity diagram for user login

### 3.13 Design Decisions

#### Interface design decisions

- Return standard error messages in the similar modules.
- Validation messages displayed nearby the relevant controller.
- Confirmation should be mandatory when executing a Administrative Action
- To make the messages clear and informative, messages should be in camel case.
- Write code comments in whenever necessary.
- Modularization should be handled based on the functionality.
- Use of common controllers and objects optimally.

#### Database design decisions

Following table describes (Table 3) system privileges of each user. Administrators can access the database with most privileges. Pharmacist has privileges to execute day to day activities. customers have least access to the database.

*Table 3.2: System Privileges*

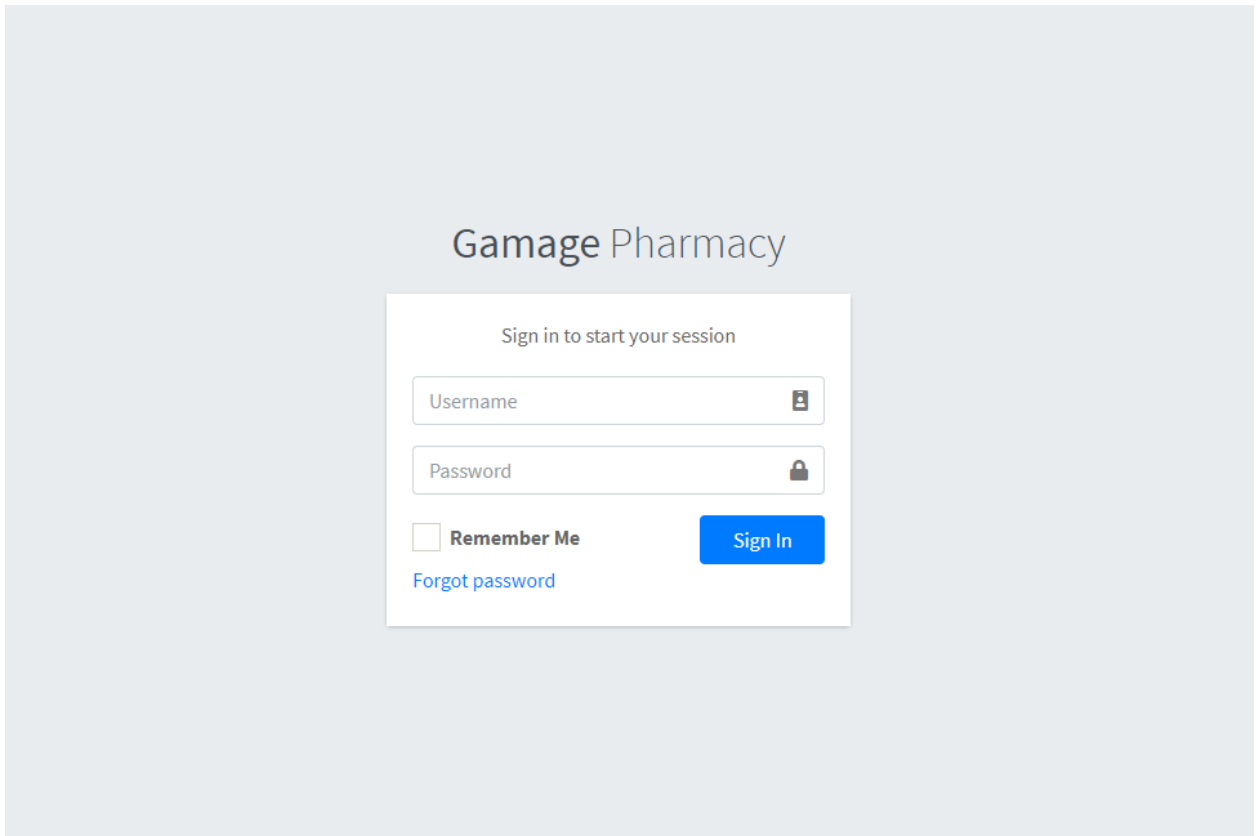
User Category/Actor	Self-change password	View All Data	View Own Profile	Update Others Profile	Update Own Profile
Administrator	YES	YES	YES	YES	YES
Pharmacist	NO	YES	YES	NO	Limited

### 3.14 System Interfaces

Few interfaces of the system are mentioned below.

Below figure (figure 3.5) shows the login page of the developed system.

## Login Page



*Figure 3.5: User login Interface*

Below figure 3.6 shows Add user Page

Add User Home / Add User

Enter Details— ×

**User Role**

**Username**

**Password**

**First Name**

**Last Name**

**Nation Identitiy Card Number**

**Date of Birth**

**Email**

**Address**

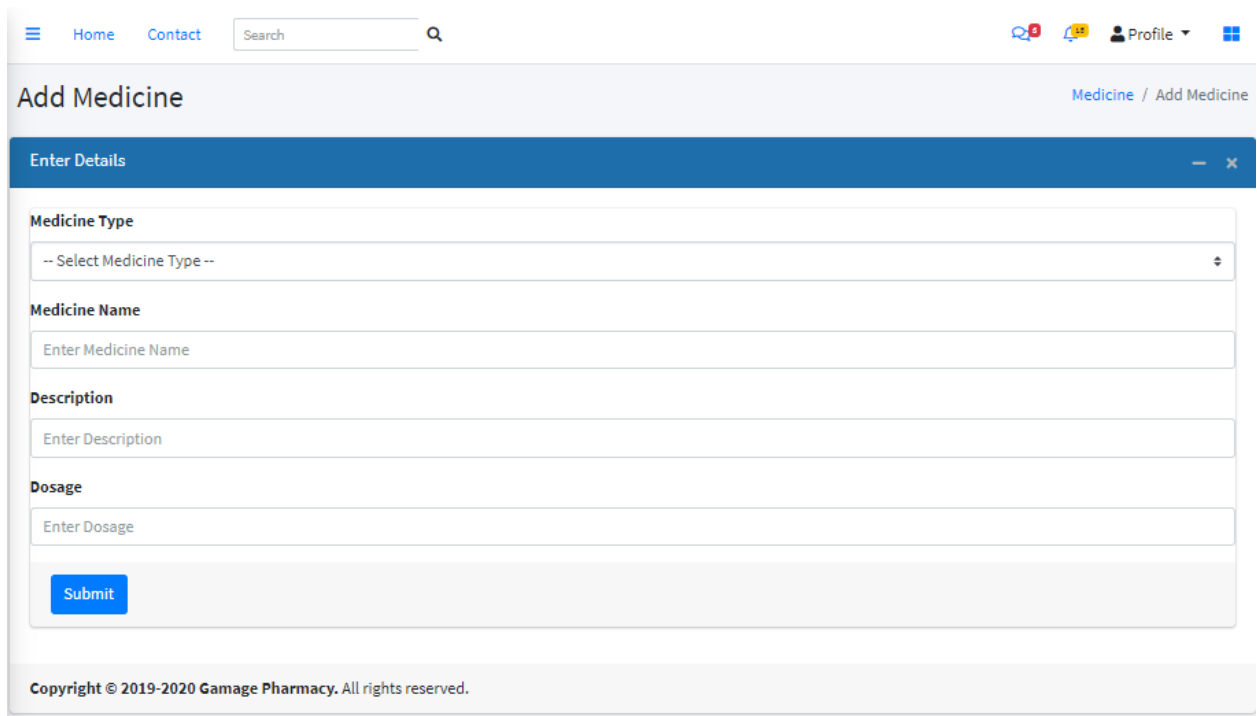
**City**

**Contact Number**

Figure 3.6: Add User Interface

Below figure 3.7 shows Add Medicine





Home Contact Search

Profile

## Add Medicine

Medicine / Add Medicine

Enter Details

**Medicine Type**

-- Select Medicine Type --

**Medicine Name**

Enter Medicine Name

**Description**

Enter Description

**Dosage**

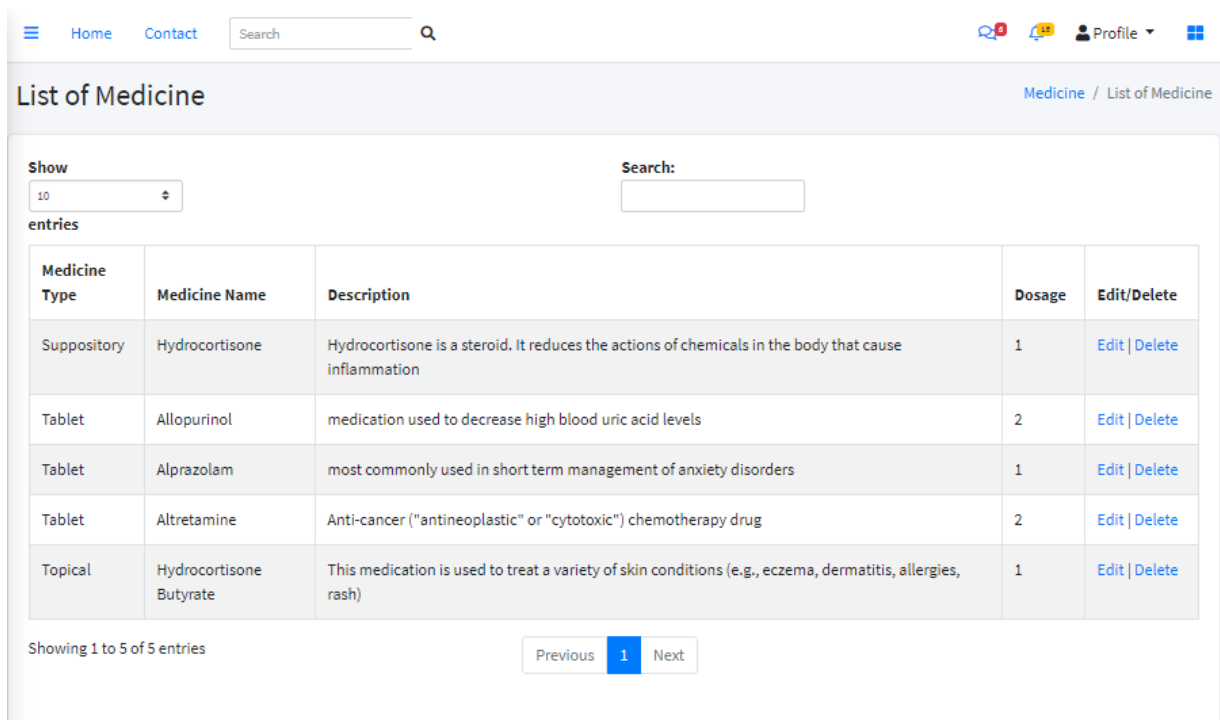
Enter Dosage

Submit

Copyright © 2019-2020 Gamage Pharmacy. All rights reserved.

Figure 3.7: Add Medicine Interface

Below figure 3.8 shows List of medicine Page



Home Contact Search

Profile

## List of Medicine

Medicine / List of Medicine

Show: 10 Search:

entries

Medicine Type	Medicine Name	Description	Dosage	Edit/Delete
Suppository	Hydrocortisone	Hydrocortisone is a steroid. It reduces the actions of chemicals in the body that cause inflammation	1	<a href="#">Edit</a>   <a href="#">Delete</a>
Tablet	Allopurinol	medication used to decrease high blood uric acid levels	2	<a href="#">Edit</a>   <a href="#">Delete</a>
Tablet	Alprazolam	most commonly used in short term management of anxiety disorders	1	<a href="#">Edit</a>   <a href="#">Delete</a>
Tablet	Altretamine	Anti-cancer ("antineoplastic" or "cytotoxic") chemotherapy drug	2	<a href="#">Edit</a>   <a href="#">Delete</a>
Topical	Hydrocortisone Butyrate	This medication is used to treat a variety of skin conditions (e.g., eczema, dermatitis, allergies, rash)	1	<a href="#">Edit</a>   <a href="#">Delete</a>

Showing 1 to 5 of 5 entries

Previous 1 Next

Figure 3.8: List of Medicine Interface

## Chapter 4

### Implementation

This chapter evaluates the development environment of the system and which tools and technologies used to develop the proposed system. Also, this describes computer language that used for development, the followed standards and development tools used. Apart from that some of the used major codes, development modules, reused codes and components are discussed here. Also, this chapter contains the methods and ways of interfaces implementation, system validates user inputs and how this system generates reports.

#### 4.1 Software Background

##### C#

C# is a general-purpose, multi-paradigm programming language encompassing strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented, and component-oriented programming disciplines.

##### ASP.NET Core MVC

The ASP.NET Core MVC framework is a lightweight, open source, highly testable presentation framework optimized for use with ASP.NET Core. ASP.NET Core MVC provides a patterns-based way to build dynamic websites that enables a clean separation of concerns

##### MySql

MySQL is an free and open-source relational database management system. This system is developed and maintained by Oracle. MySql database works in client/server or embedded systems. So, a decision has been made to use this database solution to create the database of the proposed system.

##### Jquery

jQuery is a small, fast and feature rich JavaScript library. It was designed to simplify HTML document tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software with and user-friendly API which works across many browsers.

##### Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

#### Chart.js

Is an open-source free and opensource charting JavaScript library that can be used to manipulate data-driven animated charts. Chart.js can be used with any web scripting language like ASP, .NET, JSP, ColdFusion, JavaScript, and Ruby on Rails etc. Chart.js renders in HTML5 and is widely covers great rendering performance across all major web browsers. All systems graphical repots generated via Chart.js charts.

### 4.1.1 System Implementation

Hardware and software requirements and other technical requirements needed for the implementation of the developed system is explained below.

#### Hardware Requirements

*Table 4.1-Recommended Hardware Requirements*

<b>Recommended Hardware Requirements</b>	
<b>Processor</b>	Intel Pentium Core i5 or higher
<b>RAM</b>	8 GB or higher
<b>HDD Space</b>	15 GB free hard disk space or higher
<b>Internet</b>	Dedicated connection to the internet

#### Software Requirements

*Table 4.2-Recommended Software Requirements*

<b>Recommended Software Requirements</b>	
<b>Operating system</b>	Windows OS
<b>integrated development environment</b>	Visual Studio 2017 IDE
<b>Database</b>	MySQL 8 with MySQL workbench 8.0.19 (64 bit)
<b>Web server</b>	Microsoft IIS
<b>Reporting</b>	Fusion Charts
<b>Web browser</b>	Google Chrome, Mozilla Firefox

## **Pre installations required**

1. Install Visual Studio 2017 IDE
2. Install MySQL package which installs MySQL Server 8.0 and MySQL Workbench 8.0.19.

## **Restore application database**

1. Open MySQL Workbench.
2. Log in to MySQL work bench as root user.
3. Go to Server Menu
4. Select data import from Server menu
5. Select backup file
6. Go to Import progress tab
7. Click start Import

## 4.2 Usage of Design Patterns

Model-View-Controller (MVC) software design pattern was mainly used to design and implement the proposed solution. It is the commonly used design pattern for developing user interfaces which divides the logic behind into three interconnected elements. MVC contains three main elements,

### **Model**

Model Represent the structure of the data in the table. Model is used by controller to store or retrieve data from the database. In simple words Model represent data. For an example proposed system has User management module, for this the model should be the User. User model file contains the attributes like Name, Age, Telephone etc. of the User.

### **View**

View in Model-View-Controller is simply a user interface. It displays model data to the user and enable users to modify that data. In ASP .NET MVC view uses HTML, CSS and some special syntax which makes it easier to communicate with model and controller. For an example for User Management module the Add, Edit and Delete user interfaces act as the interfaces which enable system users manipulate data.

### **Controller**

The Controller handles the user request using the user input from the view. Using the view user raises and HTTP request which analyzed by the controller and process it with help of model.

After processing Controller returns the appropriate view as a response. Single Controller can use one or more models at a given time to return a response. In user management module user controller handles the request sent to or from Add, Edit and Delete user interfaces. It provides a mechanism to manipulate user inserted data for storage purpose.

### 4.3 Development Tools

#### **Visual Studio 2017**

Visual Studio is an IDE (integrated development environment) used in computer programming which use .NET open source platform and it used mostly to develop ASP .NET applications. Visual Studio is the IDE release from Microsoft which contains a base workspace and it can be extend and customize use developers own extensions. It provides tools and services for projects of any size or complexity

#### **Draw.io – Diagrams.net**

draw.io is completely free online diagram editor built around Google Drive, that enables users to create flowcharts, UML, entity relation, network diagrams, mockups and more.

#### **Ms Project**

It is a project management software that developed by Microsoft Corporation to manage and develop project plans.

#### **My SQL Workbench**

It is free and user-friendly software to database carry out database operations. It provides a platform to develop and design MySQL databases.

### 4.4 User Interface Design

When it comes to user interface designing, the key objective should be to produce a user interface which is easy to understand by the common user. It should be efficient and user-friendly interface which enables user to interact with the machine. With a minimum user input it should be able to produce required output. Taking use of a free template design this system was designed with consistent color themes and simple error handling to follow the user interface design rules. Common users usually make mistakes while using the system, the

system itself should be able to handle and minimize those mistakes at the same time it should be easier for users to understand the issue and provide details how to solve the issue. The system should be able to recover from most of the common user introduces errors.

So, during the designing phase developer followed relevant user interface design principals to increase the user-friendliness of the system to minimize confusion about system features. The proposed system was designed to provide instant error messages (Figure 4.1) and feedback if user makes most of the pre identified common mistakes.

The screenshot shows a web form titled "Add Medicine Inventory" with a breadcrumb "Home / Add Medicine Inventory". The form is titled "Enter Details" and contains several input fields with validation errors:

- Supplier:** A dropdown menu with "-- Select Supplier --" and the error message "The value '' is invalid."
- Medicine Type:** A dropdown menu with "-- Select Medicine Type --" and the error message "The value '' is invalid."
- Medicine Name:** A dropdown menu with "-- Select Medicine --" and the error message "The value '' is invalid."
- Medicine Brand:** A dropdown menu with "-- Select Medicine Brand --" and the error message "The value '' is invalid."
- Description:** A text input field with "Enter Description..." and the error message "Please Enter Description".
- Manufacturer:** A text input field with "Enter Manufacturer" and the error message "Please Enter Manufacturer".
- Inventory Date:** A date input field with "mm/dd/yyyy" and the error message "The value '' is invalid."
- Manufacture Date:** A date input field with "mm/dd/yyyy" and the error message "The value '' is invalid."
- Expiration Date:** A date input field with "mm/dd/yyyy" and the error message "The value '' is invalid."
- Quantity:** A text input field with "Enter Quantity" and the error message "The value '' is invalid."
- Total Cost (Rs.):** A text input field with "Enter Cost" and the error message "The value '' is invalid."

A "Submit" button is located at the bottom left of the form.

Figure 4.1-Input validation for Add medicine inventory

## 4.5 Reusable Codes

When designing systems developers have usually reuse sections of codes, functions, procedures and templates. Reusing codes will save valuable time and resource. Also, it is an advantage to use of codes that have already been created and used in some form within another development process which will reduce redundancy. The main objective is to reuse parts of a computer program written at earlier product development phase which can be or should be used during the implementation of other programs.

Datacontext.cs model which located in Models in this solution, which will create database connection using set of parameters (Server name, database name, username, password) from the connection string in appsettings.json file is the most reusable code used in this solution.

## 4.6 Program List

In this section there are some programs codes related to execute main functions and validations of the developed system.

In below section it has a sample of code how one feature was developed using MVC architecture. In this section to develop a certain functionality developers first need a connection to the database using appsettings.json and DataContext.cs with SQL “ConnectionStrings”.

Model component stores data and its related logic. Model represents data which being handled between controller component. When controller class will retrieve data from the database, the model will manipulate that data and send back to the database. Model responds to requests from the View and it also responds to the instructions from controller,

Below figure (figure 4.2) contains part of DataContext.cs model file which implement the connection between the database to insert, update, delete or view data from the database.

Database connection string details are in appsettings.json file which is an input to this code.



```
1 using Microsoft.EntityFrameworkCore;
2 using Microsoft.Extensions.Configuration;
3 using System.IO;
4
5 namespace SmPharmaMgt.Models
6 {
7     public class DataContext : DbContext
8     {
9         protected override void OnConfiguring(DbContextOptionsBuilder optionBuilder)
10        {
11            var builder = new ConfigurationBuilder().SetBasePath(Directory.GetCurrentDirectory()).AddJsonFile("appsettings.json");
12
13            var configuration = builder.Build();
14            optionBuilder.UseMySQL(configuration["ConnectionStrings:DefaultConnection"]);
15        }
16
17        public DbSet<User> Users { get; set; }
18
19        public DbSet<UserRole> UserRole { get; set; }
20
21        public DbSet<Cities> Cities { get; set; }
22
23        public DbSet<Medicine> Medicines { get; set; }
24
25        public DbSet<MedicineType> MedicineTypes { get; set; }
```

Figure 4.2- DataContext.cs

Figure 4.3 shows the Model and model validation code for the validations of AddUser form.

```

namespace SmPharmaMgt.Models
{
    [Table("user")]
    17 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
    public class User
    {
        [Key]
        2 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public int user_id { get; set; }

        [Required(ErrorMessage = "Please enter Username")]
        [Remote("IsUsernameTaken", "User", HttpMethod = "POST", ErrorMessage = "Username already exists in database.", AdditionalFields = "email,nic")]
        6 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string username { get; set; }

        [Required(ErrorMessage = "Please enter password")]
        [DataType(DataType.Password)]
        [StringLength(100, ErrorMessage = "Password \"(0)\" must have {2} character", MinimumLength = 8)]
        [RegularExpression(@"^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@$!%*?&])[A-Za-z\d@$!%*?&]{6,}$", ErrorMessage = "Password must contain: Minimum 8 characters atleast 1 UpperCase Alphabet,
        5 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string password { get; set; }

        [Required(ErrorMessage = "Please enter Firstname")]
        3 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string firstname { get; set; }

        [Required(ErrorMessage = "Please enter Lastname")]
        2 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string lastname { get; set; }

        [Required(ErrorMessage = "Please enter NIC")]
        [RegularExpression(@"^[0-9]{9}[x|X|v|V][0-9]{12}$", ErrorMessage = "Please enter Correct National identitiy card value")]
        [Remote("IsUsernameTaken", "User", HttpMethod = "POST", ErrorMessage = "National identitiy card Number already exists in database.", AdditionalFields = "username,email")]
        1 reference | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string nic { get; set; }

        [DataType(DataType.Date)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode = true)]
        [Required(ErrorMessage = "Please enter Date of Birth")]
        1 reference | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public DateTime dob { get; set; }

        0 references | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string dobFormatted { get { return string.Format("{0:dd/MM/yyyy}", dob); } }

        [Required(ErrorMessage = "Please enter Email")]
        [EmailAddress(ErrorMessage = "Invalid Email Address")]
        [Remote("IsUsernameTaken", "User", HttpMethod = "POST", ErrorMessage = "Email already exists in database.", AdditionalFields = "username,nic")]
        1 reference | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string email { get; set; }

        [Required(ErrorMessage = "Please enter Address")]
        1 reference | kalckl-admin, 2 hours ago | 1 author, 1 change | 0 exceptions
        public string address { get; set; }
    }
}

```

Figure 4.3-Adduser model and model validations

Figure (figure 4.4) shows the AddUser.html view page which creates a add user form of the website



```

@model SmPharmaIgt.Models.User
@{
    ViewData["Title"] = "AddUser";
}

<!DOCTYPE html>
<html>
<head>...</head>
<body>
    <!-- Content Wrapper. Contains page content -->
    <div class="content-wrapper">
        <!-- Content Header (Page header) -->
        <section class="content-header">...</section>
        <!-- Main content -->
        <section class="content">
            <!-- Default box -->
            <div class="card">
                <div class="card-header">...</div>
                <div class="card-body">
                    <div class="card card-primary">
                        <!-- /.card-header -->
                        <!-- form start -->
                        <form role="form" method="post" asp-controller="user" asp-action="adduser">
                            <div class="form-group">
                                <label for="selectrole">User Role</label>
                                <select class="custom-select" id="selectrole" asp-for="role_id" asp-items="@((List<SelectListItem>)ViewBag.role)" >
                                    <option value="">-- Select User Role --</option>
                                </select>
                                <span asp-validation-for="role_id" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Username</label>
                                <input type="text" asp-for="username" class="form-control" placeholder="Enter Username">
                                <span asp-validation-for="username" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label for="password">Password</label>
                                <input type="password" id="password" name="password" asp-for="password" class="form-control" placeholder="Enter Password">
                                <span asp-validation-for="password" class="text-danger"></span>
                                <br />
                                <input type="password" id="password_confirm" name="password_confirm" class="form-control" placeholder="Confirm Password" oninput="check(this)">
                                <span asp-validation-for="password" class="text-danger"></span>
                                <script>...</script>
                                <span id="result"></span>
                            </div>
                            <div class="form-group">
                                <label>First Name</label>
                                <input type="text" asp-for="firstname" class="form-control" placeholder="Enter First Name">
                                <span asp-validation-for="firstname" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Last Name</label>
                                <input type="text" asp-for="lastname" class="form-control" placeholder="Enter Last Name">
                                <span asp-validation-for="lastname" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Nation Identity Card Number</label>
                                <input type="text" asp-for="nic" class="form-control" placeholder="Enter NIC number">
                                <span asp-validation-for="nic" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Date of Birth</label>
                                <input type="date" asp-for="dob" class="form-control" placeholder="Enter Date of Birth">
                                <span asp-validation-for="dob" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Email</label>
                                <input type="text" asp-for="email" class="form-control" placeholder="Enter Email">
                                <span asp-validation-for="email" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Address</label>
                                <input type="text" asp-for="address" class="form-control" placeholder="Enter Address">
                                <span asp-validation-for="address" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>City</label>
                                <br />
                                <select class="custom-select" id="selectcity" asp-for="city_id" asp-items="@((List<SelectListItem>)ViewBag.city)">
                                    <option value="">-- Select City --</option>
                                </select>
                                <span asp-validation-for="city_id" class="text-danger"></span>
                            </div>
                            <div class="form-group">
                                <label>Contact Number</label>
                                <input type="text" asp-for="contactno" class="form-control" placeholder="Enter Mobile Number">
                                <span asp-validation-for="contactno" class="text-danger"></span>
                            </div>
                        </form>
                    </div>
                </div>
                <div class="card-footer">
                    <button type="submit" class="btn btn-primary" value="Save">Submit</button>
                </div>
            </div>
            <div class="card-footer">
                <strong>Copyright &copy; 2019-2020 Gamage Pharmacy.</strong> All rights reserved.
            </div>
        </div>
    </div>
</div>
</div>
</body>
</html>

```

Figure 4.4-AddUser form

Figure (figure 4.5) shows the UserController.cs view page which contains the methods for add user function of the website

```

}
0 references | 0 requests | 0 exceptions
public IActionResult Index(string searchString)
{
    List<User> lst = new List<User>();
    lst = db.Users.ToList();
    foreach(var usr in lst)
    {
        usr.rolename = GetUserRolesbyid(usr.role_id);
        usr.city = GetUserCitybyId(usr.city_id);
    }
    ViewBag.users = lst;
    return View();
}

[HttpGet]
[Route("AddUser")]
0 references | 0 requests | 0 exceptions
public IActionResult AddUser()
{
    var roles = GetUserRoles();
    ViewBag.role = roles;
    var cities = GetCities();
    ViewBag.city = cities;
    return View("AddUser");
}

[HttpPost]
[Route("AddUser")]
[ValidateAntiForgeryToken]
0 references | 0 requests | 0 exceptions
public IActionResult AddUser(User user, IFormFile imagepath)
{
    var roles = GetUserRoles();
    ViewBag.role = roles;
    var cities = GetCities();
    ViewBag.city = cities;

    if (ModelState.IsValid)
    {
        var password = EncodeString.Encrypt(user.password.ToString());
        user.password = password;
        if (imagepath != null)
        {
            var ext = Path.GetExtension(imagepath.FileName);
            int intIdt = db.Users.Max(u => u.user_id) + 1;
            var filepathstore = Path.Combine("~/wwwroot/Images/" + imagepath.FileName);
            var filepath = Path.Combine(he.WebRootPath, "Images", Path.GetFileName(imagepath.FileName));
            imagepath.CopyTo(new FileStream(filepath, FileMode.Create));
            user.imagepath = filepathstore;
        }
        db.Users.Add(user);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View("AddUser");
}
}

```

Figure 4.5-AddUser form

Figure 4.6 shows part of the content of `_Layout.cshtml` view which used as the main layout template of this web site. This the template used to design the main interface of the system which includes common content of web pages.

```

@using Microsoft.AspNetCore.Http
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>@ViewData["Title"] - Pharmacy Management System</title>

  <!-- Font Awesome -->
  <link href="~/theme/plugins/fontawesome-free/css/all.min.css" rel="stylesheet" />
  <!-- Ionicons -->
  <link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
  <!-- overlayScrollbars -->
  <link rel="stylesheet" href="~/theme/dist/css/adminlte.min.css">
  <!-- Google Font: Source Sans Pro -->
  <link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">
</head>
<body class="hold-transition sidebar-mini">
  <div class="wrapper">
    <!-- Navbar -->
    <nav class="main-header navbar navbar-expand navbar-white">
      <!-- Left navbar links -->
      <ul class="navbar-nav">...</ul>
      <!-- SEARCH FORM -->
      <form class="form-inline ml-3">...</form>
      <!-- Right navbar links -->
      <ul class="navbar-nav ml-auto">...</ul>
    </nav>
    <!-- /.navbar -->
    <!-- Main Sidebar Container -->
    <aside class="main-sidebar sidebar-dark-primary elevation-4">...</aside>

    @RenderBody()
    <footer class="main-footer">...</footer>

    <!-- Control Sidebar -->
    <aside class="control-sidebar control-sidebar-dark">...</aside>
    <!-- /.control-sidebar -->
  </div>

  <!-- jQuery -->
  <script src="~/theme/plugins/jquery/jquery.min.js"></script>
  <!-- Bootstrap 4 -->
  <script src="~/theme/plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
  <!-- AdminLTE App -->
  <script src="~/theme/dist/js/adminlte.min.js"></script>
  <!-- AdminLTE for demo purposes -->
  <script src="~/theme/dist/js/demo.js"></script>
  <!-- DataTables -->
  <script src="~/theme/plugins/datatables/jquery.dataTables.js"></script>
  <script src="~/theme/plugins/datatables/jquery.dataTables.min.js"></script>
  <script src="~/theme/plugins/datatables-bs4/js/dataTables.bootstrap4.js"></script>
  <!-- page script -->
  <script>...</script>
  @RenderSection("Scripts", required: false)
</body>
</html>

```

Figure 4.6 \_Layout.cshhtml

## Chapter 5

### Evaluation and Testing

When it comes to evaluation of the system, the evaluation and testing should be carried out throughout the whole system development life cycle and should use better evaluation techniques. Full system test is used to evaluate the developed system functionalities are appropriate and satisfied the proposed system requirements.

This chapter evaluates how the testing of proposed system carried out and the errors that found and identified solutions for those errors. Furthermore, this chapter evaluates the carried-out test plans, test process and the expected output and the actual behavior of the system.

Software Testing is a process of executing functionality of a software application to evaluate that process to identify software bugs. Testing the system will explore whether the developed software met the set of requirements and identifies any bugs in the application. The identified bugs will be fixed ensuring a quality product. In simple words it is the process of executing the system process in order to identify any errors, breaches or missing requirements which differs to the actual requirements.

Most of the times the testing boundaries are not sufficient to guarantee the quality of the product. Also finding the design defects are rather difficult. Certain quality protocol and rules must be laid down to maintain the quality of the software solution which will impact the effectiveness of the software solution.

More testing can be used to improve the quality of the software solution. Also, this will improve verification, validation and reliability estimations. In software testing there are many tests that can be used to evaluate and test the software. Few examples for test types are Black box Testing, White Box Testing, Unit Testing, Integration testing and System testing.

#### 5.1 Test Plan

##### **Unit testing**

Individual units/ components of a software are tested during the unit testing. The main purpose of unit testing is to validate each unit of the system to check whether the system is working as designed and implemented.

A Class is the smallest testable part in object-oriented methodology, Unit a part of an application which can be execute and evaluate separately. Application Codes needs to be more reusable in order to unit testing possible. When codes are modular it is easier to reuse. Completion of each functionality unit testing will be carried out by the developer.

Each functionality or module of the system is tested by developer to identify the bugs before integrating all submodules to the main system. Few sample unit tests carried out for the proposed system are mentioned below.

Table 5.1 – unit test for user login

Test Case Scenario		User Login		
Test Case Scenario Description		This Test case Test the User Login for the System		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Sign in button with Empty Fields	Validate Empty fields in login	Show error message. “Please enter Username/Password “	Pass
2	Only Insert username value click Sign in button	Validate login without password	Show error message “Please Enter Valid Password”	Pass
3	Only Insert password value click Sign in button	Validate login without username	Show error message “Please Enter Valid Username”	Pass
4	Insert valid username and invalid password value and click submit.	Validate login with invalid password	Show error message “Please Enter Valid Username/Password!”	Pass
5	Insert invalid Username and valid Password value and click submit.	Validate login with invalid username	Show error message “Please Enter Valid Username/Password!”	Pass
6	Try to access Homepage page without logged in.	To validate unauthorized access to the system	Redirect to login page	Pass
7	Insert valid username and password.	To validate login with valid login information	Redirect to relevant home page	Pass
8	Click log out button after logged in	To check whether successfully logged out from the system	Redirect to login Page	Pass

9	Try to access Homepage page	To check the session was terminated properly	Redirect to login Page	Pass
---	-----------------------------	--	------------------------	------

Table 5.2 – unit test for add new medicine brand

Test Case Scenario		Add New Medicine Brand		
Test Case Scenario Description		This Test case test Adding new medicine brand		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Submit button with Empty Fields	Validate Empty fields on submit	Show error messages for all the Fields “Please Select Medicine“ “Please Enter Brand Name” “Please Enter Cost”	Pass
2	Only Select Medicine and click Submit button	Validate empty Brand name and Cost	Show error messages for all the Fields “Please Enter Brand Name” “Please Enter Cost”	Pass
3	Only Insert Brand name and click Submit button	Validate empty Medicine name and Cost	Show error messages “Please Select Medicine” “Please Enter Cost”	Pass
4	Only Insert Cost and click Submit button	Validate empty Medicine name and Brand name	Show error messages “Please Select Medicine” “Please Enter Brand Name”	Pass
5	Select medicine value, insert Brand name and Insert invalid value to cost field and click submit button	Validate Cost Field for number value	Show error message “Please enter valid Cost”	Pass
6	Insert duplicate Brand name and click submit	To validate inserting duplicate values	Show error message “Brand name already Exist for This Medicine”	Pass
7	Enter valid values to Medicine name,	To validate successful	Redirect to Brand Table	Pass

	Brand Name and Cost	brand addition to the system		
--	---------------------	------------------------------	--	--

Refer appendix c for more test cases.

### **Black box testing**

Black box Testing also known as Behavioral testing is a software testing method only based on the functionalities of the system without considering its internal structure or processing. Blackbox testing can be carried out for each functionality individually and for the full application when its complete. This method can be applied to every level of software testing. The objective is to check whether the systems meets the requirements.

Black box testing done for each sub system of the proposed system after completing each. Input the relevant data through the forms in sub system and check whether it provides the expected outputs. And the bugs that were prevailing in some forms of the sub systems were fixed. Black box testing was useful to identify the below issues in the developed system.

- Incorrect or missing functions
- Interface errors
- Errors in data structures
- Errors external database access
- Performance errors
- Initialization and termination errors

### **White box testing**

This testing phase is also known as open box Testing, Code-Based Testing or Structural testing is a testing method which the design and the implementation is known to the tester. The tester inputs to follow through process and the code to determines the appropriate outputs. Developers can use white box testing during the development phase of the system. Also, after the system is fully completed full code walk through can be carried out to find any existing errors.

Using white box testing methods, the following tests were made on the developed system.

- To check the truth and falsity of the values all the logical decisions were checked during the development phase
- To check paths within a functionality all the independent paths have been tested. Also, checked all the case structures to ensure cases were selected and executed in the developed system. The bugs found during the testing period in the code base were fixed.

### **Integration Testing**

During Integrations software testing phase the individual functions are combined and tested as a single system. This can be carried out during the development process where each individual functionality is tested against another functionality after they were integrated to the system.

Integration testing carried out to makes sure that individual functionalities are functioning properly when connected.

Below testing scenarios were used to test the system during the integration testing.

- When combined check whether the functionalities are according to standards and gives the expected results. Also check the functionalities are modular and developed individually.
- After combining and connecting the functions within the central system there can be some errors during the data transfers between functions which can result inconsistency in results. To detect those issues integration tests are performed in all interfaces.

### **System Testing**

System testing methodology is used when the system is fully integrated and completed. This testing method will integrate hardware and software system to make sure that the system fulfills expected objectives.

During the system testing period more complex test cases were used to test the developed system. These test cases will be created according to actual business requirements which were collected from the actual system users. During this period the system will tested by pharmacist, Owner and the pharmacy staff for a period and monitored for any flaws. Few test staff accounts were created, to perform the daily pharmacy related activities and check how the new system



handle during normal working scenario. Using actual testers to test the system is a more effective to find bugs and issues in the system.

## 5.2 Test Plan of the System

Considering all aspects of the system test cases were designed and executed in a manner to find out the flaws in the system. In the first run sub module vice tests has been carried out. Then the full system was tested end to end.

Table 5.3 – Test Plan

Modules	User/role	Events
User Management	Administrator	Login
		Create User
		Manage Users
		Assign Security Roles
		User Access Control
		Generate Reports
Stock Management	Administrator/Pharmacist	Add, edit and remove medicine
		Add, edit or remove stocks
		Report Generations
	System	Analyse stock and suggestion making
		Automatic Notifications
Supplier Management	Administrator/Pharmacist	Create supplier account
		Generate supplier invoice
		Report Generations
	System	Analyse Supplier data suggestion making
Sales Management	Administrator/Pharmacist	Add, Edit or Remove Orders
		Sales Analyzation
		Manage customer Feedback
	System	Provide sales improvement suggestions
Customer Module	Customer	Create user account
		Edit user account
		Search catalogue

		Place Orders
		Check order status
		Give Feedback

### 5.3 Test Cases

Following table shows test case for Administrator log in.

*Table 5.4 - Test case for administrator log in*

Test Case Scenario		Administrator Login		
Test Case Scenario Description		This Test case Test the Administrator Login for the System		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Sign in button with Empty Fields	Validate Empty fields in login	Show error message. "Please enter Username/Password "	Pass
2	Only Insert username value click Sign in button	Validate login without password	Show error message "Please Enter Valid Password"	Pass
3	Only Insert Password value click Sign in button	Validate login without username	Show error message "Please Enter Valid Username"	Pass
4	Insert valid username and invalid password value and click submit.	Validate login with invalid password	Show error message "Please Enter Valid Username/Password!"	Pass
5	Insert invalid username and valid password value and click submit.	Validate login with invalid username	Show error message "Please Enter Valid Username/Password!"	Pass
6	Try to access Homepage page without logged in.	To validate unauthorized access to the system	Redirect to login page	Pass
7	Insert valid username and password.	To validate login with valid login information	Redirect to relevant home page, Check full access to the system without limitation	Pass
8	Click log out button after logged in	To check whether successfully logged out from the system	Redirect to login Page	Pass

9	Try to access Homepage page	To check the session was terminated properly	Redirect to login Page	Pass
---	-----------------------------	--	------------------------	------

Following table shows test case for Create Supplier orders.

*Table 5.5 - test case for Create Supplier orders.*

Test Case Scenario		Create Supplier Order		
Test Case Scenario Description		This Test case Test the Create Supplier Order function		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click submit with Empty fields	Validate empty fields in login	Show error messages for all the Fields “Please enter valid values “	Pass
2	Only select supplier with empty other fields and click submit button	Validate empty fields in login	Show error messages for all other fields “Please enter valid values “	Pass
3	Select Supplier, Items and keep the Quantity fields empty and click submit button	Validate empty fields in login	Show error messages for all other fields “Please enter valid values “	Pass
4	Insert all the data correctly & click submit.	Validate Successful supplier order creation	Redirect to relevant Supplier order table	Pass

Refer appendix c for more test cases.

## 5.4 User Experience Evaluation

User experience evaluation or user experience assessment is a way uncover how an actual system user perceives a system before, during and after using it. Using user experience evaluation reports developers can get an overview of how actual user is satisfied with the developed system. User evaluation reports includes feedback for functional and non-functional user requirements which is important to successfully complete the project and finalize the developed system.

Some statistics related to user evaluation report is explained below.

Below figure (figure 5.1) shows the overall user satisfaction level according to evaluation report results. 60% of the users extremely satisfied, 30% of the users are very satisfied and 10% of the users are somewhat about the developed system.

### Overall User Satisfaction Level

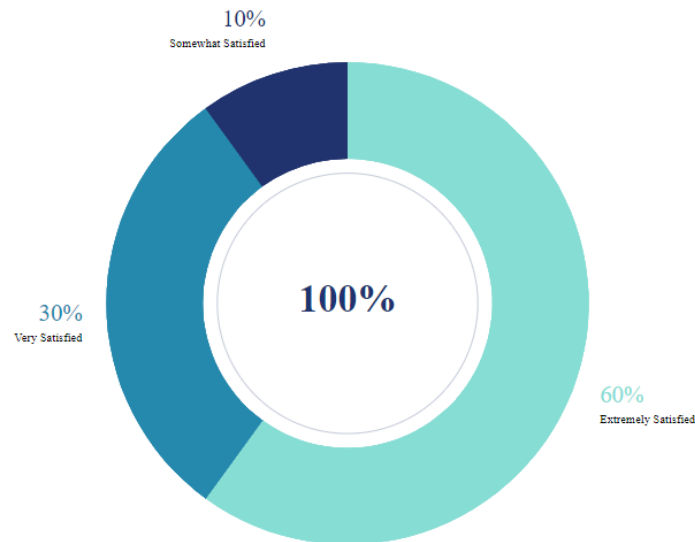


Figure 5.1 Overall user satisfaction level

Below figure (figure 5.2) shows the user satisfaction level for the system performance according to the user evaluation. From the users 60% of the users extremely satisfied, 20% of the users are very satisfied and 20% of the users somewhat satisfied with the system performance.

## User Satisfaction Level for System Performance

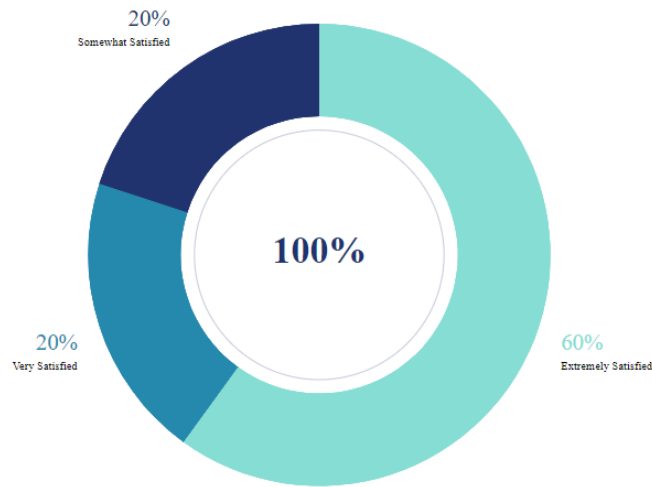


Figure 5.2 –User satisfaction levels of system performance

Below graph (figure 5.3) shows the level of overall user satisfaction level with related to the user friendliness of the developed system. More users are satisfied with the user friendliness of the system when compared to the overall satisfaction level.

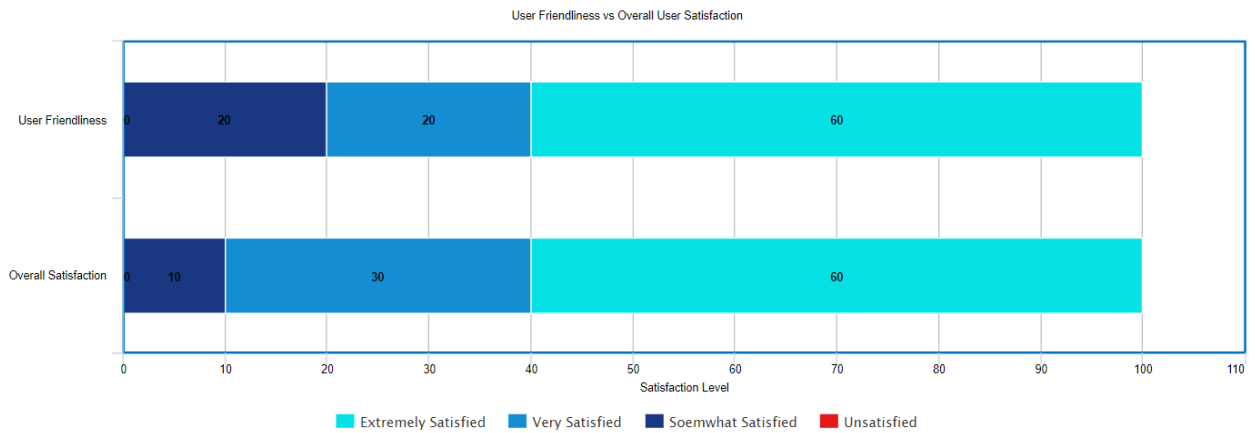


Figure 5.3- User friendliness vs overall user satisfaction

Refer Appendix D for User Evaluation Form.

## Chapter 6

### Conclusion and Future Work

The main objective of this project was to explore a concept to design a reliable, efficient and user friendly online smart pharmacy management system for Gamage Pharmacy. By using the system, the intention was to achieve more cost effective and time saving approach. Since this is an automated system this will reduce manual work which needs to be carried out every day. In the same time this reduce operational costs and improve transparency of the process.

This chapter includes the project evaluation, problems encounter during the system development, lessons learned during the project and possible future improvements for the system.

#### 6.1 Project evaluation

System should meet all the requirements of the actual users and should facilitate all the objectives mentioned in the first chapter of this document. To call it a effective system it should add values to the business.

I was able to identify main requirement of web based Smart pharmacy management system, after thorough evaluation of the problem domain. At the start of this project first phase was to gather requirements from pharmacy staff and the owner. This is to identify all the usage of the proposed system.

Using the pharmacy staff and analyzing existing system documents, author was able to get clear understanding about the pharmacy management process. Also, several alternative solutions were considered during the system implementation. After evaluating all this, C# and ASP .NET framework had been selected as the most suitable development tools to implement this solution.

One of the main objectives was to reduce amount of time spent to manage the day to day pharmacy related work. By doing so the system must eliminate the unnecessary paperwork and should improve the speed of the manual process by saving valuable time for the pharmacy staff. Supplier management should be handles without any hassle and orders should be placed in

time to replenish the stocks. Also, the system needs to reduce the wastage of the medicine by handling the demand effectively during certain periods.

Initially a literature survey and background analysis were carried out to have through idea about real world technological concepts. During the requirement gathering process, functional and non-functional requirements of the systems were identified. Also, the hardware and other requirements of the system were identified during the process.

Object-oriented methods have been followed to design the system. During this phase Entity relationship diagrams and various UML diagrams were used to get a complete view of the system processes. All the software tools were used to develop final software system is described in Chapter four.

The proposed system was designed and implemented by using advanced and latest development tools and technologies to improve the user-friendliness of the system. Report generation was designed to facilitate faster and accurate report generation in numerical or graphical formats. More test cases and test automation tools used while implementing in order to minimize the system issues and bugs.

During the implementation of the proposed system author was able to acquire through knowledge and experience about the software development life cycle. Also, the author was able to gain self-confidence on completing a development project successfully on a given time frame.

## 6.2 Problems Encountered

- At requirement gathering phase author had limited understanding about pharmacy management process. However, during the implementation phase author was able to enhance the knowledge on pharmacy management process.
- During the requirement gathering process there was some difficulties on identifying correct and needed requirements without having an end to end understand on pharmacy management process.
- Due to lack of experience on used development languages and tools more time must spent on learning before the implementation. Author had to learn some technologies from the beginning which was crucial in developing system.
- Due to above point when it comes to the development phase managing time was the main concern. Author had to spend more time developing module and functionalities.

The time spent on development exceeded previously planned time. Hence planned time frame rescheduled to complete system.

- During the requirement analysis and development phase it was decided to remove selected functionalities and added some critical needed functionalities to the system.
- Due to lack of technical expertise Interface designing was challenge, decided to use a template model to add some user-friendliness to the system.

### 6.3 Lessons Learned

Self-learning was used most of the time during the development process in order to complete the system. Author had to learn about the software development life cycle, development methodologies, designing methodologies and development technologies.

During the project development phase author acquired sound knowledge and experience about software system implementation, programming languages and techniques. After the self-learning curve author gained an excellent knowledge of tools which were used in the development process. During the implementation phase most developments were carried out by trial and error.

Using the previously gained knowledge on testing methodologies and project management the author was able to get a better understanding on how to execute the testing and how to walk through the project life cycle.

From writing this dissertation Author gained more knowledge about the documentation methods. Extra effort and time were needed to improve the writing skill of the author which ultimately improved authors writing skill and method.

In the development phase suitable software, hardware and development environments has to selected by author. Which improves the multitasking capabilities. Author learned may new technologies to make the project a success.

### 6.4 Achievements

One of the main objectives of this project was to automate the manual pharmacy related routine work. By implementing the proposed system will prove the pharmacy staff a robust solution



which will reduce the unnecessary time spent during work hours. The system will automate the whole pharmacy management process and to provide effective management on stocks reducing the wastage. This will increase the profit margin by a considerable amount. Without the unnecessary time spent on daily routines the pharmacy staff will be able to provide a better service to their customers efficiently which will ultimately increase the customer satisfaction level. Since the stocks can be managed and ordered using the system without any hassle current phone call-based ordering can be stopped which will save money and time for the owners.

Since the inventory control happens automatically and stock information will be displayed via alerts owners can order medicine before had and replenish the stocks before they were out of stock.

One of the major achievements of this project is, event thought the smart automated pharmacy management system developed to considering Gamage Pharmacy Homagama, the same solution can be modified to use in any kind of pharmacy to meet their requirement since all the common functionalities for the pharmacy management are present in this solution. Proposed system can be customized to suite any pharmacy since it was developed with that flexibility.

Using the system will reduce manual work like document keeping for each transaction as well as keeping historical information for future evaluation. By automating those processes the system will reduce the time consume for each task and generate reports which can be utilize for future management decision makings.

This system will increase the productivity and efficiency of the pharmacy and at the same time this will have a positive impact to the profitability of the pharmacy. Since most of the business process were automated by the system the pharmacy can be operate with reduced manual labor which will ultimately lead the business to grow further.

## 6.5 Future Developments

- Current system uses cash on delivery for supplier orders Since that is used by most suppliers who already deals with the pharmacy. But online payment method should be introduced to the system to keep better monitoring on transactions to increase the transparency of the routine transactions.

- When using SMS communication functionalities, it was recommended to use paid SMS gateway to send and receive system generated SMS messages
- Mobile application is the new trend when it comes to management systems. Since it will provide a better management on the go. The current system can be used as a web site using the mobile phone but recommends developing a mobile application to manage the pharmacy.

## References

[1]"GOFRUGAL POS Software for Retail, Restaurant & Distribution", *Gofrugal.com*, 2020. [Online]. Available: <https://www.gofrugal.com/>. [Accessed: 09- May- 2020].

[2] M. Inc., "Empowering Providers with Innovative Healthcare Solutions | Meditab", *Meditab.com*, 2020. [Online]. Available: <https://www.meditab.com/>. [Accessed: 09- May- 2020].

[3]"Helping Patients Get the Medication They Need | CoverMyMeds", *Covermymeds.com*, 2020. [Online]. Available: <https://www.covermymeds.com/>. [Accessed: 09- May- 2020].

[4]"Home - Rx Master", *Rx Master*, 2020. [Online]. Available: <https://rxmaster.com/>. [Accessed: 09- May- 2020].

# Appendix A – System Configuration Documentation

## System Documentation

Title : Smart Computerized Pharmacy Management System  
Version : 1.0  
System : Web application

This documentation provides information regarding the technical requirements on system configuration, installation and other technical information for system administrators, developers, and anyone who wants to improve the system or customize the system. Please refer to the recommended hardware requirements (Table A.1) and software requirements (Table A.2) for the system.

## Hardware Requirements

Recommended Hardware Requirements	
Processor	Intel Core i3 or higher
RAM	8 GB or higher
HDD Space	15 GB free hard disk space or higher
Internet	Dedicated connection to the internet

*Table A. 1 Recommended Hardware Requirements*

## Software Requirements

Recommended Software Requirements	
Operating system	Windows OS
Eclipse IDE	Visual Studio 2017 or higher
Database	MySQL 8.0 with MySQL workbench 8.0.19 or higher
Web server	IIS
Web browser	Google Chrome, Mozilla Firefox

*Table A. 2 Recommended Software Requirements*

## Pre installations required

1. Install Visual Studio 2017 IDE
2. Install MySQL package which installs MySQL Server 8.0 and MySQL Workbench 8.0.19.

## Restore application database

1. Open MySQL Workbench.
2. Log in to MySQL work bench as root user.

3. Go to Server Menu
4. Select data import from Server menu
5. Select backup file
6. Go to Import progress tab
7. Click start Import

## Appendix B - Use case narratives

Following table (Table B.1) describes use case narrative for customer log in.

<b>ID: U01</b>	<b>Actors: Administrator</b> <b>Title: Log in</b>
Description:	Relevant Administrator comes to log in page Logged in using Username and password.
Preconditions:	Administrator has a valid username and password. User should be in login page
Main Flows:	Enter a valid username. Enter valid password. User log in to the system.
Post Conditions:	User Log in successfully and user redirect to Main page
Alternative paths:	User enters invalid username then an incorrect username message will be displayed. User enters an invalid password then incorrect password message will be displayed. User enters only username then please insert valid password message will be displayed. User enters only password then username password mismatch message will be displayed. Administrator should re-enter correct log in details or terminate the process.

*Table B.1 Use case narrative for Administrator log in*

Following table (Table B.2) describes use case narrative for how Administrator add users to the system.

<b>ID: U02</b>	<b>Actors: Administrator</b> <b>Title: Manage user accounts</b>
Description:	Relevant Administrator add, edit or delete users to/from the system
Preconditions:	Administrator has a valid username and password. Administrator should be in logged in to the system
Main Flows:	<i>Add User to the system</i> Administrator should fill all required fields with valid values Administrator should select User type Email should have “@” sign and “.” to be a valid email

	Password must contain: Minimum 8 characters at least 1 Uppercase Alphabet, 1 Lowercase Alphabet, 1 Number and 1 Special Character
	Passwords should match with confirm password field
	Mobile number should contain ten digits
	NIC number should have a valid format
	Redirect to User List upon submit
	<i>Edit User</i>
	Administrator enters username in search field
	Select relevant user from search results and click edit button
	Administrator should fill all required fields with valid values
	All previous conditions should be satisfied when entering values
	Redirect to User List upon submit
	<i>Delete User</i>
	Administrator enters username in search field
	Select relevant user from search results and click delete button
	Display confirmation messages
	Click confirm
	Redirect to User List upon submit
Post Conditions:	User information will be added to the user table User information will be edit or delete from the User table
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages
	Try to search value not in the database then display empty result view
	Click another user link when updating user profile then discard changes

Table B. 2 Use case narrative for Add User to the System

Following table (Table B.3) describes use case narrative for create supplier order.

<b>ID: U03</b>	<b>Actors: Administrator/Pharmacist</b> <b>Title: Create Supplier order</b>
Description:	Describes how Administrator or pharmacist create supplier order
Preconditions:	Administrator or pharmacist should be logged in to the system Administrator or Pharmacist is in Add Supplier Order Page
Main Flows:	Add Supplier Order to the System
	Administrator or Pharmacist should fill all required fields with valid values (Supplier ID etc.)
	Administrator or Pharmacist should add items to the list

	Administrator or Pharmacist should add Quantity to respective item
	Click Submit Button
Post Conditions:	Supplier invoice generated for the supplier order
	Order details updates in the order table
	SMS sent to supplier regarding the order number
	Email Sent to supplier with the copy of supplier order
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages

Table B.3 Use case narrative for create supplier order

Following table (Table B.4) describes use case narrative for Generate Report.

<b>ID: U04</b>	<b>Actors: Administrator/Pharmacist</b> <b>Title: Generate reports</b>
Description:	Describes how Administrator/Pharmacist generate system related reports
Preconditions:	User logged into the system
	User is in generate report page
	There should be at least one data record in database
Main Flows:	User enters relevant details required for the report
	Click view button
	Display report.
Post Conditions:	User can click export option as pdf or jpg image format
	Input values can be added again to generate new report
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages

Table B.4 Use case narrative for Generate reports

Following table (Table B.5) describes use case narrative for manage medicine

<b>ID: U02</b>	<b>Actors: Administrator/Pharmacist</b> <b>Title: Manage Medicine</b>
Description:	Relevant User add, edit or delete <i>Medicine</i> to/from the system
Preconditions:	Administrator has a valid username and password.
	Administrator should be in logged in to the system
Main Flows:	<i>Add Medicine to the system</i>
	Administrator should fill all required fields with valid values
	Administrator should select <i>Medicine</i> type
	Administrator should enter <i>Medicine</i> Name



	Administrator should enter <i>Medicine</i> Description
	Administrator should enter <i>Medicine</i> Dosage
	Redirect to <i>Medicine</i> List upon submit
	<i>Edit Medicine</i>
	Administrator enters <i>Medicine name</i> in search field
	Select relevant <i>Medicine</i> from search results and click edit button
	Administrator should fill all required fields with valid values
	All previous conditions should be satisfied when entering values
	Redirect to <i>Medicine</i> List upon submit
	<i>Delete Medicine</i>
	Administrator enters <i>Medicine</i> name in search field
	Select relevant <i>Medicine</i> from search results and click delete button
	Display confirmation messages
	Click confirm
	Redirect to <i>Medicine</i> List upon submit
Post Conditions:	<i>Medicine</i> information will be added to the user table <i>Medicine</i> information will be edit or delete from the <i>Medicine</i> table
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages
	Try to search value not in the database then display empty result view
	Click another <i>Medicine</i> link when updating user profile then discard changes

Table B.5 Use case narrative for manage medicine

Following table (Table B.6) describes use case narrative for manage pharmacy items

<b>ID: U02</b>	<b>Actors: Administrator/Pharmacist</b> <b>Title: Manage Pharmacy Items</b>
Description:	Relevant User add, edit or delete pharmacy item to/from the system
Preconditions:	User has a valid username and password. User should be in logged in to the system
	User should have valid administrator privileges to add, edit or delete items
Main Flows:	<i>Add Pharmacy Item to the system</i>
	User should fill all required fields with valid values

	User should select <i>Pharmacy Item</i> type from the dropdown
	<i>Pharmacy Item size should be select from dropdown</i> Choose default if items available in same size.
	<i>Cost value should be Number</i>
	Redirect to <i>Pharmacy Items List</i> upon submit
	<i>Edit Pharmacy Item</i>
	User enters <i>Pharmacy Item Name</i> in search field
	Select relevant <i>Pharmacy Item</i> from search results and click edit button
	User should fill all required fields with valid values
	All previous conditions should be satisfied when entering values
	Redirect to <i>Pharmacy Item List</i> upon submit
	<i>Delete Pharmacy Item</i>
	User enters <i>Pharmacy Item</i> name in search field
	Select relevant <i>Pharmacy Item</i> from search results and click delete button
	Display confirmation messages
	Click confirm
	Redirect to <i>Pharmacy Item List</i> upon submit
Post Conditions:	<i>Pharmacy Item</i> information will be added to the user table <i>Pharmacy Item</i> information will be edit or delete from the <i>Medicine</i> table
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages
	Try to search value not in the database then display empty result view
	Click another <i>Pharmacy Item</i> link when updating user profile then discard changes

Table B.6 Use case narrative for manage pharmacy items

Following table (Table B.7) describes use case narrative for manage suppliers

<b>ID: U02</b>	<b>Actors: Administrator</b> <b>Title: Manage Suppliers</b>
Description:	Relevant Administrator add, edit or delete <i>supplier</i> to/from the system
Preconditions:	Administrator has a valid username and password. Administrator should be in logged in to the system
Main Flows:	<i>Add supplier to the system</i>

	Administrator should fill all required fields with valid values
	Administrator should select supplier type
	Email should have “@” sign and “.” to be a valid email
	Mobile number should contain ten digits
	Landline number should contain ten digits
	Fax number should contain ten digits
	Redirect to <i>supplier</i> List upon submit
	<i>Edit supplier</i>
	Administrator enters <i>supplier</i> name in search field
	Select relevant <i>supplier</i> from search results and click edit button
	Administrator should fill all required fields with valid values
	All previous conditions should be satisfied when entering values
	Redirect to <i>supplier</i> List upon submit
	<i>Delete supplier</i>
	Administrator enters <i>supplier</i> name in search field
	Select relevant user from search results and click delete button
	Display confirmation messages
	Click confirm
	Redirect to <i>supplier</i> List upon submit
Post Conditions:	<i>Supplier</i> information will be added to the supplier table <i>Supplier</i> information will be edit or delete from the supplier table
Alternative paths:	Leave required fields empty and click submit then display an error message
	Enter invalid value as an input then system display error messages
	Try to search value not in the database then display empty result view
	Click another <i>supplier</i> link when updating user profile then discard changes

Table B.7 Use case narrative for manage suppliers

## Appendix C – Test cases

Following table (Table C.1) describes test case for User Login

Test Case Scenario		User Login		
Test Case Scenario Description		This Test case Test the User Login for the System		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Sign in button with Empty Fields	Validate Empty fields in login	Show error message. <i>“Please enter Username/password “</i>	Pass
2	Only Insert username value click Sign in button	Validate login without password	Show error message <i>“Please Enter Valid Password”</i>	Pass
3	Only Insert Password value click Sign in button	Validate login without username	Show error message <i>“Please Enter Valid Username”</i>	Pass
4	Insert valid username and invalid password value and click submit.	Validate login with invalid password	Show error message <i>“Please Enter Valid Username/Password!”</i>	Pass
5	Insert invalid username and valid password value and click submit.	Validate login with invalid username	Show error message <i>“Please Enter Valid Username/Password!”</i>	Pass
6	Try to access Homepage page without logged in.	To validate unauthorized access to the system	Redirect to login page	Pass
7	Insert valid username and password.	To validate login with valid login information	Redirect to relevant home page	Pass
8	Click log out button after logged in	To check whether successfully logged out from the system	Redirect to login Page	Pass
9	Try to access Homepage page	To check the session was terminated properly	Redirect to login Page	Pass

*Table C: 1 Test case for User Login*

Following table (Table C.2) describes test case for Add New Medicine Brand

Test Case Scenario		Add New Medicine Brand		
Test Case Scenario Description		This Test case Test Adding New Medicine Brand		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Submit button with Empty Fields	Validate Empty fields on submit	Show error messages for all the Fields “Please Select Medicine “ “Please Enter Brand Name” “Please Enter Cost”	Pass
2	Only Select Medicine and click Submit button	Validate empty Brand name and Cost	Show error messages for all the Fields “Please Enter Brand Name” “Please Enter Cost”	Pass
3	Only Insert Brand name and click Submit button	Validate empty Medicine name and Cost	Show error messages “Please Select Medicine” “Please Enter Cost”	Pass
4	Only Insert Cost and click Submit button	Validate empty Medicine name and Brand name	Show error messages “Please Select Medicine” “Please Enter Brand Name”	Pass
5	Select medicine value, insert Brand name and Insert invalid value to cost field and click submit button	Validate Cost Field for number value	Show error message “Please enter valid Cost”	Pass
6	Insert duplicate Brand name and click submit	To validate inserting duplicate values	Show error message “Brand name already Exist for This Medicine”	Pass
7	Enter valid values to Medicine name, Brand Name and Cost	To validate successful brand addition to the system	Redirect to Brand Table	Pass

Table C.2 Test case Add new medicine brand

Following table (Table C.3) describes test case for Administrator Login

Test Case Scenario	Administrator Login
--------------------	---------------------

Test Case Scenario Description		This Test case Test the Administrator Login for the System		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click Sign in button with Empty Fields	Validate Empty fields in login	Show error message. <i>“Please enter Username/Password “</i>	Pass
2	Only Insert username value click Sign in button	Validate login without password	Show error message <i>“Please Enter valid password”</i>	Pass
3	Only Insert Password value click Sign in button	Validate login without username	Show error message <i>“Please Enter valid username”</i>	Pass
4	Insert valid username and invalid password value and click submit.	Validate login with invalid password	Show error message <i>“Please Enter Valid Username/Password!”</i>	Pass
5	Insert invalid username and valid password value and click submit.	Validate login with invalid username	Show error message <i>“Please Enter Valid Username/Password!”</i>	Pass
6	Try to access Homepage page without logged in.	To validate unauthorized access to the system	Redirect to login page	Pass
7	Insert valid username and password.	To validate login with valid login information	Redirect to relevant home page, Check full access to the system without limitation	Pass
8	Click log out button after logged in	To check whether successfully logged out from the system	Redirect to login Page	Pass
9	Try to access Homepage page	To check the session was terminated properly	Redirect to login Page	Pass

Table C.3 Test case Administrator Login

Following table (Table C.4) describes test case for Create Supplier Order

Test Case Scenario		Create Supplier Order		
Test Case Scenario Description		This Test case Test the Create Supplier Order function		
Test Case Number	Test Case Description	Objective	Result	Status

1	Click submit with Empty fields	Validate empty fields in Create Supplier Order	Show error messages for all the Fields <i>“Please enter valid values “</i>	Pass
2	Only select supplier with empty other fields and click submit button	Validate empty fields in Create Supplier Order	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
3	Select Supplier, Items and keep the Quantity fields empty and click submit button	Validate empty fields in Create Supplier Order	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
4	Insert all the data correctly & click submit.	Validate Successful supplier order creation	Redirect to relevant Supplier order table	Pass

Table C.4 Test case for Create Supplier Order

Following table (Table C.5) describes test case for Add users by Administrator

Test Case Scenario		Add users by Administrator		
Test Case Scenario Description		This Test case Test Add users by Administrator		
Test Case Number	Test Case Description	Objective	Result	Status
1	Insert already existing username as username	Prevent adding similar usernames	Show error messages username already exist	Pass
2	Insert username with special characters and click submit	To check if system able to face sql injection	Show error messages <i>“Username has invalid characters”</i>	Pass
3	Keep username blank with all other required fields filled with valid values and click submit	Validate empty fields	Show error messages <i>“Please enter valid Username “</i>	Pass
4	Insert first name and last name as numerical value with all other required fields filled with valid values and click submit	Avoid numerical values in name field	Show error messages <i>“Please enter valid characters “</i>	Pass
5	Insert first name and last name fields as special characters	To avoid special character	Show error messages <i>“Please enter valid characters “</i>	Pass

	with all other required fields filled with valid values and click submit	values in name field		
6	Add new username name value with all other required fields filled with valid values	To validate fields	Records added to the database and redirect to List of user's tables	Pass
7	Insert contact no as numerical value with all other valid field filled and click submit	To validate numeric field	Records added to the database and redirect to List of user's tables	Pass
8	Insert invalid contact no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	Show error message <i>"Please enter valid Number"</i>	Pass
9	Insert invalid Email with all other required fields filled with valid values and click submit	To validate email field	Show error message <i>"Please enter valid Email"</i>	Pass
10	Insert valid Email with all other valid field filled and click submit	To validate email field	Records added to the database and redirect to List of user's tables	Pass
11	Insert invalid NIC Number with all other required fields filled with valid values and click submit	To validate NIC field	Show error message <i>"Please enter valid NIC Number"</i>	Pass
12	Insert valid NIC Number with all other required fields filled with valid values and click submit	To validate NIC field	Records added to the database and redirect to List of user's tables	Pass
10	Insert Password without any special character with all other required fields filled with valid values and click submit	To validate password field	Show error message <i>"Password must contain: Minimum 8 characters at least 1 Uppercase Alphabet, 1 Lowercase Alphabet, 1 Number and 1 Special Character"</i>	Pass



11	Insert Valid Password with all other valid field and click submit	To validate password field	Record added to the table	Pass
12	Do not pick date of birth from the date time picker	To validate empty field	Show error message “ <i>Please enter valid Date of Birth</i> “	Pass
13	pick date of birth from the date time picker	To validate empty field	Record added to the table	Pass
14	Do not pick city from city dropdown	To validate empty field	Show error message “ <i>Please Select City</i> “	Pass
15	pick city from city dropdown	To validate empty field	Record added to the table	Pass

Table C.5 Test case for Add users by Administrator

Following table (Table C.6) describes test case for Edit users by Administrator

Test Case Scenario		Edit user by Administrator		
Test Case Scenario Description		This Test case Test Edit users by Administrator		
Test Case Number	Test Case Description	Objective	Result	Status
1	Try to Change Username	Prevent changing username	Show error messages “ <i>Username cannot be changed</i> ”	Pass
3	change first name and last name as numerical value	Avoid numerical values in name field	Show error messages “ <i>Please enter valid characters</i> “	Pass
4	Change first name and last name fields as special characters	To avoid special character values in name field	Show error messages “ <i>Please enter Valid characters</i> “	Pass
5	Submit the changed value(s) except username with all other required fields filled with valid values	To validate fields	Records updated in the database and redirect to List of user’s tables	Pass
6	Insert contact no as numerical value with all other valid field and click submit	To validate numeric field	Records added to the database and redirect to List of user’s tables	Pass
7	Insert invalid contact no as numerical value	To validate numeric field	Show error message “ <i>Please enter Valid Contact Number</i> “	Pass

8	Insert invalid Email	To validate email field	Show error message “ <i>Please enter Valid Email</i> “	Pass
9	Insert valid Email with all other valid field and click submit	To validate email field	Records added to the database and redirect to List of user’s tables	Pass
10	Insert invalid NIC Number	To validate NIC field	Show error message “ <i>Please enter valid NIC Number</i> “	Pass
11	Insert valid NIC Number with all other valid field and click submit	To validate NIC field	Records added to the database and redirect to List of user’s tables	Pass
12	Insert Password without any special character	To validate password field	Show error message “ <i>Password must contain: Minimum 8 characters at least 1 Uppercase Alphabet, 1 Lowercase Alphabet, 1 Number and 1 Special Character</i> ”	Pass
13	Edit Valid Password with all other valid field and click submit	To validate password field	Record added to the table	Pass
14	Delete date of birth from the date time picker	To validate empty field	Show error message “ <i>Please enter valid Date of Birth</i> “	Pass
15	pick date of birth from the date time picker	To validate empty field	Record added to the table	Pass
16	Do not pick city from city dropdown	To validate empty field	Show error message “ <i>Please Select City</i> “	Pass
17	pick city from city dropdown	To validate empty field	Record added to the table	Pass

Table C.6 Test case for Edit users by Administrator

Following table (Table C.7) describes test case for Add supplier by Administrator

<b>Test Case Scenario</b>		Add supplier by Administrator		
<b>Test Case Scenario Description</b>		This Test case Test Add supplier by Administrator		
Test Case Number	Test Case Description	Objective	Result	Status

1	Insert already inserted supplier registration number	Prevent adding similar supplier	Show error messages supplier already exist	Pass
2	Insert Supplier name with special characters and click submit	To check if system able to face sql injection	Show error messages <i>“Supplier name has invalid characters”</i>	Pass
3	Keep username blank with all other required fields filled with valid values and click submit	Validate empty fields	Show error messages <i>“Please enter valid Supplier Name “</i>	Pass
4	Insert supplier name as numerical value with all other required fields filled with valid values and click submit	Avoid numerical values in name field	Show error messages <i>“Please enter valid characters “</i>	Pass
5	Insert supplier name as special characters with all other required fields filled with valid values and click submit	To avoid special character values in supplier name field	Show error messages <i>“Please enter valid characters “</i>	Pass
6	Add new registration number with all other required fields filled with valid values	To validate fields	Records added to the database and redirect to List of supplier’s tables	Pass
7	Insert mobile. Landline and fax no as numerical value with all other valid field filled and click submit	To validate numeric field	Records added to the database and redirect to List of user’s tables	Pass
8	Insert invalid mobile contact no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	Show error message <i>“Please enter valid Number “</i>	Pass
9	Insert invalid landline contact no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	To validate numeric field	Pass

10	Insert invalid fax no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	To validate numeric field	Pass
11	Insert invalid Email with all other required fields filled with valid values and click submit	To validate email field	Show error message <i>“Please enter valid Email “</i>	Pass
12	Insert valid Email with all other valid field filled and click submit	To validate email field	Records added to the database and redirect to List of user’s tables	Pass
13	Do not pick supplier type with other required fields filled with valid values and click submit	To validate supplier type, drop down field	Show error message <i>“please Select a supplier type “</i>	Pass

Table C.7 Test case for Add supplier by Administrator

Following table (Table C.8) describes test case for Add New Invoice

Test Case Scenario		Add New Invoice		
Test Case Scenario Description		This Test case Test the Add New Invoice function		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click submit with Empty fields	Validate empty fields in Add Invoice	Show error messages for all the Fields <i>“Please enter valid values “</i>	Pass
2	Insert customer name as numerical value with all other required fields filled with valid values and click submit	To avoid special character values in customer name field	Show error messages <i>“Please enter valid characters “</i>	Pass
3	Insert valid customer name with all other valid data with empty date field	Validate empty fields in Add Invoice	Set date as the current date	Pass
4	Add item to the invoice	Validate add new invoice item	New empty row added to the invoice item list	Pass
5	Enter character in item field	Validate automatic item search function	Drop down list of items appear when entering characters	Pass

6	Try to Insert invalid item in item field	Validate non existing items	No values selected to inventory number dropdown field and all other fields gets disabled	Pass
7	Insert valid item with no available quantity	Validate empty inventory	No values selected to inventory number dropdown field and all other fields gets disabled	Pass
8	Insert valid item with available quantity	Validate empty inventory	Inventory numbers selected to inventory number dropdown field and all other fields gets disabled	Pass
9	Select inventory number	Validate auto fill of available quantity, Expiry date and unit price fields	Available quantity, expiry date and unit price select to relevant fields	Pass
10	Insert Quantity higher than available quantity	Validate quantity field against available quantity	Show error message <i>"Insert Valid Quantity"</i>	Pass
11	Insert few items to the invoice	Validate autofill of subtotal, total discount, tax, Total fields	Calculated values autofill to relevant fields	Pass
12	Insert all the data correctly & click submit.	Validate Successful Add Invoice function	Print Invoice and refresh the invoice page	Pass

Table C.8 Add New Invoice

Following table (Table C.9) describes test case for Add Medicine Inventory

Test Case Scenario		Add Medicine Inventory		
Test Case Scenario Description		This Test case Test the Add Medicine Inventory function		
Test Case Number	Test Case Description	Objective	Result	Status
1	Click submit with Empty fields	Validate empty fields in Add Medicine inventory	Show error messages for all the Fields <i>"Please enter valid values"</i>	Pass

2	Only select supplier with empty other fields and click submit button	Validate empty fields in Add Medicine inventory	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
3	Only select Medicine Type with empty other fields and click submit button	Validate empty fields in Add Medicine inventory	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
4	Only select Medicine Name with empty other fields and click submit button	Validate empty fields in Add Medicine inventory	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
5	Only select Medicine Brand with empty other fields and click submit button	Validate empty fields in Add Medicine inventory	Show error messages for all other fields <i>“Please enter valid values “</i>	Pass
6	Try to Insert quaintly as alphabetical characters	Validate empty Add Medicine inventory	Text box does not support for alphabetical characters	Pass
6	Try to Insert cost field as alphabetical characters	Validate empty Add Medicine inventory	Text box does not support for alphabetical characters	Pass
4	Insert all the data correctly & click submit.	Validate Successful Add medicine Inventory function	Redirect to relevant Medicine Inventory table	Pass

Table C.9 Add Medicine Inventory

Following table (Table C.10) describes test case for Add Customer

Test Case Scenario		Add Customer		
Test Case Scenario Description		This Test case Test Add Customer		
Test Case Number	Test Case Description	Objective	Result	Status
1	Insert first name and last name as numerical value with all other required fields filled with valid values and click submit	Avoid numerical values in name field	Show error messages <i>“Please enter valid characters “</i>	Pass
2	Insert first name and last name fields as special characters with all other required fields filled with valid values and click submit	To avoid special character values in name field	Show error messages <i>“Please enter valid characters “</i>	Pass

6	Insert invalid NIC Number with all other required fields filled with valid values and click submit	To validate fields	Show error message <i>“Please enter valid NIC Number “</i>	Pass
7	Insert already existed NIC Number with all other required fields filled with valid values and click submit	To validate customer already registered	Show error message <i>“NIC Already exists “</i>	Pass
8	Insert invalid contact no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	Show error message <i>“Please enter valid Number “</i>	Pass
9	Insert valid contact no as numerical value with all other required fields filled with valid values and click submit	To validate numeric field	Records added to the database and redirect to List of user’s tables	
11	Insert invalid Email with all other required fields filled with valid values and click submit	To validate email field	Show error message <i>“Please enter valid Email “</i>	Pass
12	Insert valid Email with all other valid field filled and click submit	To validate email field	Records added to the database and redirect to List of user’s tables	Pass
13	Empty city filed with all other required fields filled with valid values and click submit	To validate city field	Show error message <i>“Please select a valid City “</i>	Pass
14	Insert all required fields filled with valid values and click submit	To validate add new customer function	Records added to the database and redirect to List of user’s tables	Pass

*Table C.10 Add Customer*

## Appendix D: User Evaluation Form

### User Evaluation Form for Smart computerized Pharmacy Management system

*(These details are required for system evaluation purposes only and will not be disclosed)*

1. Personal details

Name .....  
Job role .....  
Email .....

*(Please mark the relevant box)*

2. What is your level of computer and internet literacy?

**Average**    **Basic**    **Advanced**

3. How satisfied are you with the **functionalities** of the system?

**Extremely satisfied**    **Very Satisfied**    **Somewhat Satisfied**     
**Unsatisfied**

4. How Satisfied are you with the system **performance**?

**Extremely satisfied**    **Very Satisfied**    **Somewhat Satisfied**     
**Unsatisfied**

5. How Satisfied are you with the **reliability** of this system?

**Extremely satisfied**    **Very Satisfied**    **Somewhat Satisfied**     
**Unsatisfied**

6. How Satisfied are you with this system's **ease of use**?

**Extremely satisfied**    **Very Satisfied**    **Somewhat Satisfied**     
**Unsatisfied**

7. How Satisfied are you with the **user friendliness** of this system?

**Extremely satisfied**    **Very Satisfied**    **Somewhat Satisfied**     
**Unsatisfied**

8. Do you have any thought on how to improve this system?

Comment:

.....  
.....