

# Project Report

## Pharmacy Management System Group 3S



Software Engineering  
Group Project

Department of Computer Science,  
Faculty of Science ,  
University of Jaffna.

# Pharmacy Management System

By

**Group 3S**

The report submitted for the course of software engineering group project for the degree of

**Bachelor of Science in**

**Computer Science (B.Sc. computer science)**

**Name of the Supervisor**

**Dr. A. Ramanan**

**(Senior lecturer)**



**B.Sc.in Computer Science-2014/2015**

**Department of Computer Science**

**Faculty of Science**

**University of Jaffna.**

**Sri Lanka.**

## I. Executive Summary

Our team is group 3S,

The members of our team are:

S.Jathurshan (2014/CSC/003)

M.Kaushik (2014/CSC/010)

S.Thilakarathna (2014/CSC/030)

C.Lahiru (2014/CSC/034)

A.Tharshikan (2014/CSC/036)

Group mentor:

Senior lecturer Dr.A.Ramanan

We are undergraduate students and following Computer Science at Faculty of Science of University of Jaffna. This report is contained the detailed information of our second year group project. The title of our project is “Pharmacy Management Application”. The aim of our project to provide a good standalone software for medicine stock and billing management with efficient manner.

Why we are choosing this idea to our project? There are many pharmacies surrounding us but there are few pharmacies have the application to manage the stocks and billing. Unfortunately, many pharmacies that they do not have the complete application to manage the pharmacy, and another thing is the keeping their medicine records and sales details manually with large number of files, books and hard copies with large space. If the pharmacy people want to know any particular details of medicine, they must search hole files of the form of hard copy. This method is wasting their own time and, it will make them annoyance and frustrated. And also each medicines have unique brand name, generic name, and different dosages, these kind of information are hard to find manually.

This software saving the human time, we can store the medicine details in proper way, billing and billing records can be secured, and application will give the automatic alert and notifications about the expiry date of the medicine and low stock of medicine.

Main advantages of our application are:

- Reducing human effort, human errors, time and make easy to manage huge database.
- Solve the conflict during the stock entry.
- Easy to maintain the stock details and sales details.
- Do not need larger space comparing to the physical records.
- Secure

The result of our project is complete standalone application for pharmacy management. The application is password protected and all the password is encrypted by md5 hashing algorithm. There are two panels “Admin” and “User”. Admin can access fields of Medicine, Staff, Supplier, Sales, change password. User has restricted access on the application (mainly billing section).

“ATLANTA Pharmacy” is involving in our project, because we implemented the standalone application for the “ATLANTA Pharmacy”. They liked to work with us and they are happy to use our product. We got the real data from them.

We designed and implemented our product as a user friendly. We can access easily. In future, we have a plan to develop our application with barcode reader. It will help to our customers to handle our application easily.

‘Group 3S’ have many challenges during the project period. In our initial stage we do not have a good client because of these we could not get requirements on time, after that we used C# as a programming language to implement our application. This programming language is new to our group members. We learned C# in short time of period for this purpose. Finally, group 3S produce their end product.

## II. Declaration of Authorship

Hereby we, Mr.S.Jathurshan, Mr.M.Kausik, Mr.S.Thilakarathna, Mr.C.Lahiru, A.Tharshikan are members of group 3S certify that, our project is title as “Pharmacy Management Application” and this is completely our own work. The project of “Pharmacy Management Application” was done by group 3S.

This project was not submitted in identical or similar form to another degree program or same degree program.

Date

.....  
Mr.S.Jathurshan

(2014/CSC/003)

.....  
Mr.M.Kausik

(2014/ CSC /010)

.....  
Mr.S.Thilakarathna

(2014/ CSC /030)

.....  
Mr.C.Lahiru

(2014/ CSC /034)

.....  
A.Tharshikan

(2014/ CSC /036)

.....  
Dr.A.Ramanan

Supervisor,

Department of Computer Science,

Faculty of Science.

.....  
Dr.K.Thabotharan

Head of the Department,

Department of Computer Science,

Faculty of Science.

### III. Contribution of each member

#### Mr.S.Jathurshan

He gave the idea of “Pharmacy Management Application” for the group project of software engineering course unit. He designed the whole interface of this application, arranging the button functions of each interface based on functional requirements, took part application testing and he has learned C# programming language in short time of period for this project work. This project report is prepared by him.

He has coded the following interfaces using C# programming language:

- Login interface
- Admin interface
- User interface
- Change password interface
- Add Medicine interface
- Add Staff interface
- Add Supplier interface
- Modify Medicine Details interface
- Modify Staff Details interface
- Modify Supplier Details interface
- Retails Sales Details interface (billing part of the application)

And also coded for validation of Email Id, NIC No, Contact number, Numerical and character validation of textboxes.

Mr.M.Kausik

He has learned C# programming language in short time of period and he has coded the following interfaces of the application:

- Login interface
- Admin interface
- User interface
- Change password interface
- Add Medicine interface
- Add Staff interface
- Add Supplier interface
- Modify Medicine Details interface
- Modify Staff Details interface
- Modify Supplier Details interface
- Retails Sales Details interface (billing part of the application)
- Delete Medicine Details interface
- Delete Staff Details interface
- Delete Supplier Details interface
- View Purchase interface
- View Expiry Date interface
- View Stock interface
- View Supplied Details interface
- View Staff Details interface
- View Sales interface

He has done hashing for password encryption and decryption using md5 hashing algorithm and took part of application testing.

Mr.S.Thilakarathna & Mr.A.Tharshikan

He collected the information from the pharmacy people

- How they buy medicines from the medical agents,
- What kind of medicines that they sell to the customers?
- Details of medicines.
- How they keep the record of new stock, sales, orders and purchases
- How they remove the expiry stocks from the record and return to the agent.

And also he collected some medicine labels and containers and details of manual billing application. He got the functional and non-functional requirements from them.

Mr.C.Lahiru

He good in SQL database and help to connect the SQL server to visual studio platform. He helped to retrieve the data from database with SQL codes, he created database, tables and he entered the data to the tables.

#### IV. Consent of the client

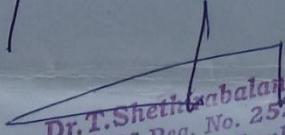
To Dr. T. Shelthabalan  
Head  
Deposit or Contra Side

09/02/17

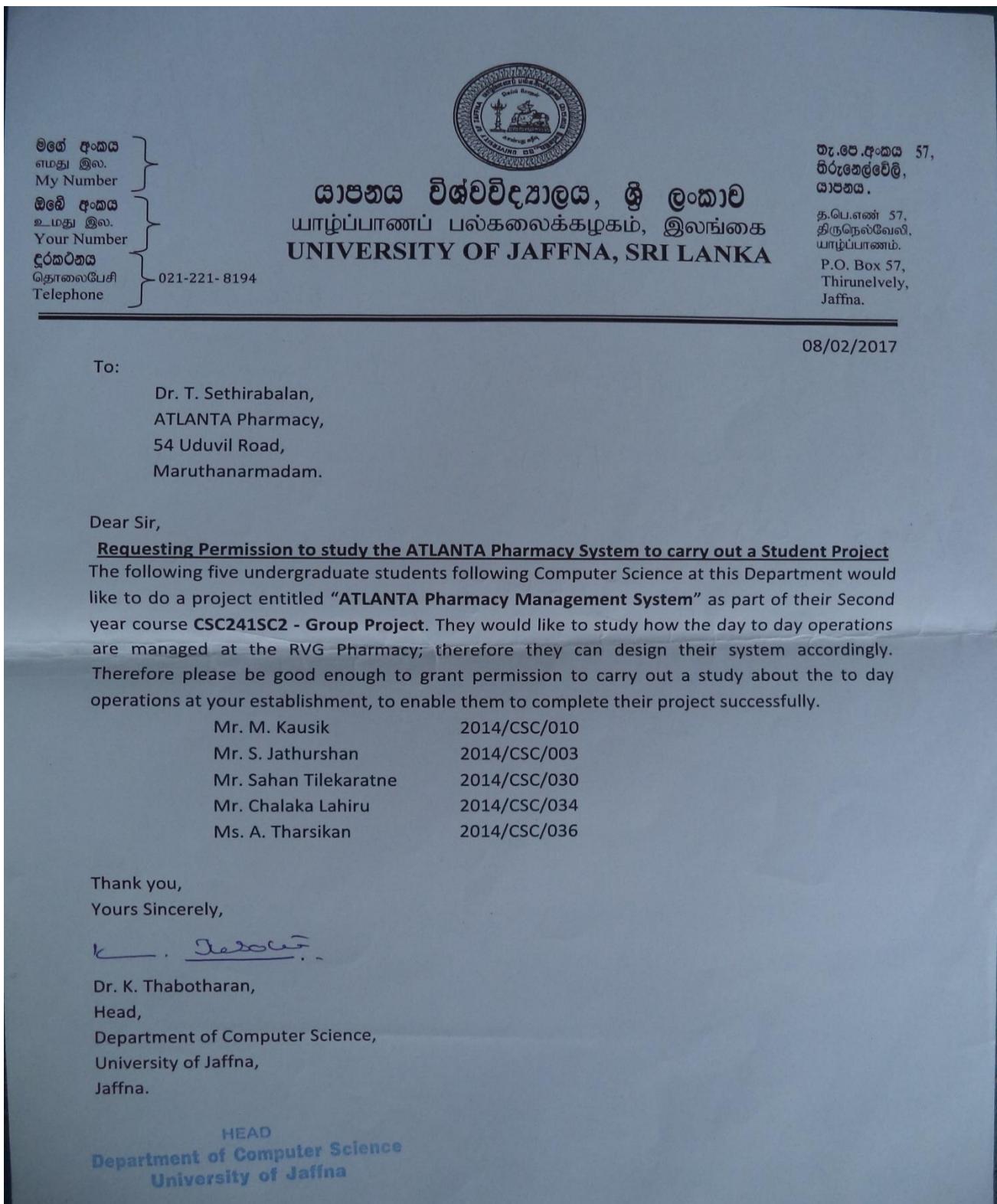
There go for your studies to  
do his project, I remain  
the tie project were us.  
Yours OI.

Thiriyag

Atlanta Hospital  
No.54, Uduvil Road,  
Maruthanarmadam,  
Jaffna  
Tel.021 2242299

  
Dr.T.Shelthabalan (MD)  
SLMC Reg. No. 25246  
Jaffna - Sri Lanka

## V. Letter of approval of the final software



## Table of content

Content	Page no
<b>I. Executive summary</b>	2
<b>II. Declaration of authorship</b>	4
<b>III. Contribution of each members</b>	5
<b>IV. Consent of the client</b>	8
<b>V. Letter of approval of the final software</b>	9
<b>Table of content</b>	10
<b>List of figures</b>	11
<b>List of tables</b>	12
<b>1. Introduction</b>	13
<b>2. Purpose of the project</b>	14
<b>3. Scope of the project</b>	14
<b>4. Over view of the documentation</b>	15
<b>5. Requirement Engineering</b>	16
5.5. Techniques used for requirements gathering	16
5.6. Functional requirements	16
5.7. Non-functional requirements	23
5.8. Domain requirements	23
5.9. Challenges	25
<b>6. Software design</b>	26
6.1. Software design	27
6.2. Software design approach	38
6.3. Context diagram	39
6.4. Interaction diagram	40
6.5. Structural diagram	42
6.6. Behavior diagram	43
6.7. Data modal	51
6.8. Challenges	52
<b>7. Implementation</b>	53
7.1. Tools and techniques and justification	53
7.2. Special features	55
7.3. Challenges	56
<b>8. Testing</b>	57
<b>9. Discussion and challenges</b>	67
<b>10. References</b>	68
<b>11. Appendices</b>	68
11.1. User documentation	68
11.2. Development documentation	73

## List of figures

Figure 5.2.1- Sample details for medicine for real data.	17
Figure 5.2.2- Sample details for medicine for real data.	18
Figure 5.2.3- Sample details for medicine for real data.	19
Figure 5.2.4- Sample details for medicine for real data.	20
Figure 5.2.5- Sample details for medicine for real data.	21
Figure 5.2.6- Sample details for medicine for real data.	22
Figure 6.1.1- login interface	27
Figure 6.1.2- admin interface	27
Figure 6.1.3- user interface	28
Figure 6.1.4- change password interface	28
Figure 6.1.5- add medicine interface	29
Figure 6.1.6- view stock interface	29
Figure 6.1.7- modify medicine interface	30
Figure 6.1.8- view expiry date interface	30
Figure 6.1.9- delete medicine details interface	31
Figure 6.1.10- add staff interface	31
Figure 6.1.11- view staff details interface	32
Figure 6.1.12- delete staff details interface	32
Figure 6.1.13- modify staff details interface	33
Figure 6.1.14- add supplier details interface	34
Figure 6.1.15- view supplier details interface	35
Figure 6.1.16- modify supplier details interface	35
Figure 6.1.17- delete supplier details interface	36
Figure 6.1.18- retail sales details interface	36
Figure 6.1.19- view sales details interface	37
Figure 6.1.20- view purchase details interface	37
Figure 6.2.1.1-software engineering approach.	38
Figure 6.3.1.-context diagram for the system.	39
Figure 6.4.1-sequence diagram for the admin functions	40
Figure 6.4.1-sequence diagram for the user functions	41
Figure 6.5.1-UML class diagram for the system	42
Figure 6.6.1-sequence diagram for login page	43
Figure 6.6.2-sequence diagram for add medicine page	44
Figure 6.6.3-sequence diagram for search medicine page	45
Figure 6.6.4-sequence diagram for delete medicine page	46
Figure 6.6.5-sequence diagram for modify medicine page	47
Figure 6.6.6-sequence diagram for change password page	48
Figure 6.6.7-sequence diagram for billing section page	49
Figure 6.6.8-sequence diagram for reminder page	49
Figure 6.6.9-use-case diagram for the system	50
Figure 6.7.1-ER diagram for the system	51
Figure 8.1-testing technique	57
Figure 11.1.1-how to install the system(step1)	68
Figure 11.1.2-how to install the system (step 2)	69
Figure 11.1.3-how to install the system (step 3)	69
Figure 11.1.4-how to install the system (step 4) and setup the path of the system	70
Figure 11.1.5-how to install the system (step 5)	70
Figure 11.1.6-open the application.	71
Figure 11.1.7-little guide for user	71

**List of tables**

Table: 8.1 – Minimum testing platform requirements	58
Table: 8.2 – add medicine case discussion	59
Table: 8.3 – view medicine case discussion	59
Table: 8.4 – view expiry date case discussion	60
Table: 8.5 – modify medicine case discussion	60
Table: 8.6 – delete medicine case discussion	61
Table: 8.7 – add staff case discussion	61
Table: 8.8 – view staff case discussion	62
Table: 8.9 – modify staff case discussion	62
Table: 8.10 – delete staff case discussion	63
Table: 8.11 – add supplier case discussion	63
Table: 8.12 – view supplier case discussion	63
Table: 8.13 – modify supplier case discussion	64
Table: 8.14 – delete supplier case discussion	64
Table: 8.15 – retails sales case discussion	65
Table: 8.16 – view sales case discussion	65
Table: 8.17 – view purchase case discussion	65
Table: 11.2.1.1-feedback of user	73

## 1 Introduction

The project titled “Pharmacy Management Application”, this is a standalone application for medical and pharmacy information.

This is a user friendly and interactive application for managing both stock and billing which helps in maintain the records of the medicine, the users and store details and also reduce the work of searching the stock details. The main aim of this application is to apply technology is supporting the pharmacist during the data management and reduce human effort on searching and automation of the billing.

The project has been developed on the basis of “stock managing” and it’s “billing process” being presently used in the medical stores for storing and retrieving the available information in the store. The application has auto notification features for low stocks and expiry information of stocks. The expiry alert will be given in current date and the low stock alert will be given, when the particular stock count reach 50. Each medicines and other items in the pharmacy are labelled by unique Id, this method is useful for avoiding the problem during the stock details entry and it is easy for searching.

The user has to get his username and password from the admin by providing the name, address, phone number, Id proof and role. They can get the access to the application. Without the username and password, he cannot get access to the application. The password is encrypted by hashing algorithm(md5).

In this application the users will be specified by their role in the pharmacy and the user can access only few functions (view sales, view stock, view purchase, and retail sales (billing section)). And the admin can access and manage all the features of the application.

## 2 Purpose of the project

The main purpose of the project is replacing the manually maintained record of sales, purchase, billing, stock, staffs and suppliers by automated application. Another thing is to give a good application for pharmacist to avoid their work pressure during the billing period, no one cannot cheat the administration and owner and many small pharmacies do not have application like this, one of the major problem is their economic status, but they wish to use the application for maintaining their pharmacy details.

## 3 Scope of the project

We are introducing the software “Pharmacy Management System” which is implemented for “ATLANTA Pharmacy”. The application is designed simply and easy to access. The major function of the application is maintaining medicine details and billing section and dealing the supplier information. Supplier details are related with the medicine details, by these return of medicines is easy to handle.

Owner(admin), manager, staff (person who are in the counter) are the end users of this application. Owner can access all the functionalities of the application without any restrictions. But, the manager and staffs can access with limited restrictions. Only admin can make any changes to the data. This will help to keep their details secretly.

We hope that, most of the pharmacist like to use our application. We have decided to include the barcode system in future to this application.

## 4 Overview of the document

In documentation overview, we have included the problem, challenges, and technics used in our project.

- First stage: The technics used for requirements gathering, and problems of getting the functional and non-functional requirements, and what are the challenges we faced to get the requirements.
- Second stage: Idea of Software design, software design approach, we illustrated with the diagram to the application, data model and challenges during the designing work.
- Third stage: Implementation work and strategy, tools and techniques and justification of special features.

And also included the references, user and development documentation.

## 5 Requirement Engineering

Our project “Pharmacy Management System” needs requirements to design and implementation of application. Initially we got the requirements from “RVG Pharmacy” and we started our designing work to the application with their requirement. Unfortunately, we could not work with them. But, we were implementing our application with their requirements and we did not have a pharmacy for our project.

After 2 months our team were connected with “ATLANTA Pharmacy”. We showed our application until we have implemented. The Administration of “ATLANTA Pharmacy” were happy to see that. And they gave some additional requirements with few corrections.

Finally, we implemented our standalone application, with requirements which we were getting from the pharmacist.

### 5.1 Techniques used for requirements gathering

First we approached the owner of the pharmacy with the question “what are the problems you are facing in your business in daily basis?”. Our team described them about our idea for them, how to manage their conflicts and problems of pharmacy management and we told them the basic functionality of our application. Addition to that we explained to importance of the system to the pharmacy. After that we requested for the requirements of them. They listed out their requirements, it was easy to reach our complete application. we needed the medicine details because, we want real data to the process. Our client gave us some labels, boxes and container of medicines and gave the approval to the project.

### 5.2 Functional requirements

- Generate report weekly on information about the drugs and it exports the information as output document.
- Store the detailed information about each medicine including actual name formula of medicine.
- When the user searches the item on search bar the related things were displayed in the screen and can select the actual item that the user needs and searching must be multiple searching.
- Give an alert when the medicine outdated.
- Give an alert for the user when medicine stock is low.
- Automated billing system and prepare bill for the sales.
- Managing the details of medicine, staff, supplier and sales.
- Date and time display.



**Figure 5.2.1- Sample details for medicine for real data.**

# PHARMACY MANAGEMENT SYSTEM

44)	LIDOCAINE HYDROCHLORIDE JELLY 4%SP	30ml	oct/15	May/18	Amcica Jelly
45)	SALICYLIC ACID Preservative Free Salicylic Acid 2%	320 doses	100mg	07/13	07/16
46)	Dexamethasone Hydrochloride	0.025%	10ml	Jun/14	May/17
47)	Clorem-G (Antimicrobial Cream)	30gm	07/14	06/17	Clorem-G Cream
48)	Fidapa Cream (Adapalene )	0.1%	10gm	May/14	Apr/17
49)	Pain-zeal	30g			Methyl Salicylate Ointment BP
50)	Aqueous Cream with Glycerine	50g	15/06	17/06	D - Soft cream
51)	Flamox Cream	20g	Dec/2015	Nov/2018	-
52)	Benzyl Benzoate 25% w/v	100 mL	10/01/15	10/01/17	-
53)	Asthelin (Respirator Solution)	15 mL	12/01/14	11/01/16	Solubilized Respirator Soln
54)	Mometasone Furoate Cream USP	5g	08/01/15	02/01/17	Momicon cream
55)	Miconazole Cream BP	15g	07/01/15	06/01/17	Unimed cream
56)	Benzymethotoin Hydrochloride Nasal soln 1% w/v 0.025%	10ml	11/2014	04/2017	Advair Novum (PABA free)
57)	Silver Sulfadiazine Cream USP	25g	12/01/15	11/01/18	Silverch cream
58)	KETOPROFEN	15g	1/7/01/14	1/7/2017	Fastum gel
59)	Antifungal, Antiflammatory	15g	Dec/14	Nov/17	Transcipro.ca cream
60)	Terbinfine Hydrochloride Cream	10g	1/Feb/2014	31/Jan/21/7	TERBINIFINE cream
61)	SILVIRIN	20g	13/11/14	12/11/17	Silver Sulfadia -zinc cream USP

	Name	mf	Ex
1)	SPMC		
2)	Benzhexol Tablets B.P.	10 tablets 2mg	Aug/14 Nov/16 Benzhexol Hydrochloride
3)	HIST-8	10x10 Tablets 8mg	Apr. 2015 Mar 2017 Histidine Dihydrochloride Tablets BP
4)	Walcort	10 strips of 0.5mg 10 tablets	29/04/2015 28/04/2018 Betamethasone Tablets BP
5)	Gastro-resistant Bismocodyl Tablets BP	10x10 rapid Tablets	11/2015 10/2018 BISMOCODYL
6)	Hyoscine Butylbromide Tablets IP	10x10 Tablets 10mg	03/2016 02/2019 CANTEROL
7)	Bromhexine Hydrochloride Tablets BP	10x10 8mg	10/2015 01/2017 Bromhexine Hydrochloride
8)	Diazepam tablets B.P.	100 tablets 5mg	03/2016 02/2019 Diazepam
9)	Dexamethasone Tablets BP	0.5mg	03/2015 02/2018 Dexamethasone
10)	"	1000 Tablets 0.5mg	03/2015 03/2017 "
11)	Diclofenac Tablets BP	10x10 50mg	(01/2015) 09/2018 Diclofenac
12)	Gastro-resistant Diclofenac Tablets BP	10x10 25mg	Mar/2015 Feb/2017 "
13)	"	10x10 50mg	Jul/2015 Jun/2016 "
14)	Slow Diclofenac Tablets BP (Sustained Release)	10 strips of 100mg tablets each	04/2014 03/2017 "
15)	ADIFLAM-50 TAB	30x100	09/2015 02/2018 Diflunisal
16)	Domperidone Tablets BP	2x5 Blister of 10 tablets each	May/2016 Feb/2018 Domperidone
17)	"	10x10 "	"
18)	Doxycycline Tablets	2x50 Tablets 100mg	10/2015 09/2019 Doxycycline
19)	" "	10x10 10 mg	Jan/2015 Dec/2018 "
20)	Doxycycline Capsules BP	10x10 capsules 100mg	07/2015 06/2017 Doxycycline

E	1)	Esomeprazole tablet	10mg	20mg	09/2015	08/2017	Esomeprazole
2)	"	"	50mg	80mg	Apr 15	Mar 17	"
3)	"	"	10x10	20x10	Jul 15	Jul 18	"
4)	Sotalol Erxtra	300	150mg				Etofylline
	Tablets of Etofylline and Theophylline			(Etofylline 150mg) (Theophylline 300mg)	07/15	08/18	
5)							
6)	2)	Glibenclamide	10x10	5mg	16/2/2015	16/2/2020	Glibenclamide Tablets
7)	2)	Gliclazide tablets BP	10x10	80mg	July/2015	June/2017	Gliclazide
8)	2)	Glibenclamide Tablets BP	10x10	6mg	03/2015	02/2018	Glibenclamide
9)	"	"	50	5mg	04/2014	03/2017	"
10)	2)	Gliclazide Tablets BP	10x10	80mg	08/2015	07/2017	Gliclazide
11)	"	"	10 tablets 5 strips of 10 tablets	80mg	Dec/14	Nov/16	"
12)	"	"	10 tablets	80	11/2015	10/2017	"
13)	"	"	10 tablets	50	09/2015	09/2017	"
Amiodarone & Daps							
14)	2)	Beta-blocker Valerenate Cream BP	15g	12/2015	11/2017	Beta-blocker	
15)	2)	Clotrimazole Cream BP, easy, w/w	0.5g	4/15	10/18	Clotrimazole cream	
16)	2)	"	0.5g	07/15	10/17	"	
17)	2)	Ciprofloxacin Hydrochloride Solution	5ml	Jul 15	Jul 17	Ciprofloxacin	
18)	2)	(Eye/Ear drops)					
19)	2)	Aqueous Cream (Emollient, Moisturiser & Soothing Substances)	100g	15.06	17.06	Cetrimide Shampoo Aqueous Cream	
20)	"	"	"	"	"		
21)	2)	Betamethasone Valerenate Cream BP	15g	12/2015	11/2017	Betamethasone	
22)	2)	Clotrimazole Cream BP, easy, w/w	0.5g	4/15	10/18	Clotrimazole cream	
23)	2)	"	0.5g	07/15	10/17	"	
24)	2)	Ciprofloxacin	5ml	Jul 15	Jul 17	Ciprofloxacin	
25)	2)	(Eye/Ear drops)					
26)	2)	Aqueous Cream (Emollient, Moisturiser & Soothing Substances)	100g	15.06	17.06	Cetrimide Shampoo Aqueous Cream	
27)	2)	"	"	"	"		
28)	2)	Betamethasone & Dexamethasone sodium phosphate 0.01% w/v   0.1% w/v	5ml	May/15	Apr/17	Betamethasone	
29)	2)	(Eye/Ear drops)					
30)	2)	Betamethasone and Miconazole Lotion 5ml	05/15	04/17	Betamethasone		
31)	2)	Clotrimazole Eye/Ear Drops 0.3% w/v	10x5ml	04/15	04/18	GRACIN	
32)	2)	Ciprofloxacin Eye/Ear Drops	5ml	04/15	07/18	CIPLOX	

47)	LIDOCAINE HYDROCHLORIDE 10% USP	80ml	oct/15	Mar/18	Amica Jelly
48)	SALBUTAMOL Pressurised Inhalation 60 doses	100mg	07/13	07/16	SALBUTAMOL
49)	Oxytetracycline Hydrochloride 0.05%	10ml	Jun/14	May/17	Oxytetracycline
50)	Clonex G1 (Antimicrobital cream)	30gm	07/14	06/17	Clonex G1
51)	Fidra cream (Adapalene ) 0.1%	10gm	May/14	Apr/17	Fidra cream
52)	Painrelief	30g			Methyl salicylate Balsam EP
53)	Aqueous Cream with Glycerine	50g	15/06	17/06	D - Soften cream
54)	Flomar CREAM	20g	Dec 2015	Nov 2019	-
55)	Benzyl Benzoate 25% w/v	100ml	06/01/15	16/06/17	-
56)	Asthelin (Respirator Solution)	15ml	12/01/14	11/01/16	Salbutamol Respirator soln
57)	Mometasone Furoate Cream USP	6g	03/05	02/017	Memon cream
58)	Miconazole Cream EP	15g	07/01/15	06/01/17	Umbrales cream
59)	Oxytetracycline Hydrochloride Nasal Soln USP 0.025%	30ml	11/02/14	05/2017	Noxses Nasal (nasal spray)
60)	Silver Sulfadiazine Cream USP	20g	10/01/15	11/01/18	Silver cream
61)	KE TOPROFEN	15g	1/7/1014	14/10/19	Fastin gel
62)	Antifungal, Antiflammatory	15g	Dec 16	Nov 17	Transfase cream
63)	Tertbutanis Hydrochloride Cream	10g	1/05/1014	31/Jan/17	TERTBUTANIS cream
64)	SLIVIRIN	30g	13/11/14	12/11/17	Silver Sulphadiazine cream

**Figure 5.2.2- Sample details for medicine for real data.**

## PHARMACY MANAGEMENT SYSTEM

● 9) Metformin Tablets 250	250mg	09/14	06/16	Metformin
● 9) Metoclopramide Tablets 10	10mg	07/15	06/18	Metoclopramide
● 9) Metronidazole tablets 400	400mg	05/15	04/018	Amo-Metronidazole
● 9) Amodia 400	400mg	10/15	9/18	
● 9) Metronidazole tablet Rx form	400mg	7/15	6/18	
● 9) Mifepro 0.25mg	0.25mg	15/7/2015	25/7/2020	
● 9) Mifepro 0.3mg	0.3mg			
● 9) Nifedipine Retard tablets	30mg	07/14	06/17	
● 9) Proprianol tablets	10mg	07/2015	06/2012	
● 9) GRABIN 75	75mg	09/2015	03/2017	
● 9) Pregabalin capsules	75mg			
● 9) Prednisalone tablets	5mg	12/2014	11/2017	
● 9) Ranitidine tablets	150mg	09/15	08/18	
● 9) Benfo benztiamine.	500mg	07/2015	06/2017	
● 9) Recoplus	280mg	11/2016	06/2017	
● 9) Vitamine B complex Folic Acid Vitamin C Zinc	40mg	27/06/2014	07/06/2018	
● 9) Glaxo-medimethimicin				
Dose:- Infant 0.3ml (8 dropperful) 3-4 times daily				
● 9) Calcium lactate tablets	800mg	04/2015	03/2017	
● 9) Lvitone syrup	200ml	11/2015	09/2017	
● 9) Vit B-complex zinc				
● 9) Lysofit syrup	300ml	02/2016	7/2017	
● 9) Vitamin B tablets	29mg	08/2015	4/18	
● 9) Anemidox capsules of vitamins with Iron.	700mg	11/2015	1/2017	

● 9) Delonil Salbutamol	Salbutamol	07-2015	06-2017	100m
● 9) Expectorant Salbutamol Bromhexine Hcl Guaifenesin & Menthol Syrup		08		
● 9) Paediatric Paracetamol	PARACETAMOL 75 mg/5 ml	Jul 2015 - Jun 2018		50ml
● 9) Oral solution BP	120	1-6-275		
● 9) 6-12-475				
● 9) chlorphenamine oral solution				
● 9) Oral solution	-	-	12-2015 11-2017	75ml
● 9) Rp				
● 9) Salbutamol			12-2015 02-2018	75ml
● 9) Oral Solution				
● 9) Briton chlorphenamine bisigalite	03-2016 - 09-2016			750mls
● 9) Expectorant	Expectorant child 6-12 75			

* Cefcoxib 100	100mg	MAY15	MAY17	Cefcoxib
* Cefturoxime Axetil	500mg	OCT15	SEP17	Cefturoxime Axetil
colour - Titatium Dioxide				
* Cefixime 200mg	200mg	MAR15	FEB18	Cefixime Trihydrate
colour - Titanium Dioxide (IVA FUXIM - 200)				
* Cefixime Dispersible 100mg	100mg	FEB16	SAN19	Cefixime Trihydrate
THEOTIX				
* Cefalixin capsules	500mg	MAY15	JUN17	Cefalixin monohydrate
colour - Magnesium Oxide				
* Cefalexin	250mg	JUL14	JUN17	Cefalexin Monohydrate
T THEOFLEX- 250				
* Carbamazepine	200mg	DEC14	NOV19	Carbamazepine
ZEPTOL 200				
* Cefuroxime Axetil	250mg	JUL15	JUN18	Cefuroxime Axetil
colour - Magnesium Oxide				
* Cefuroxime Axetil	250mg	JUL15	JUN18	Cefuroxime Axetil
colour - Magnesium Oxide				

* As directed by physician				
* Clopidogrel	75mg	JAN15	JAN15	Clopidogrel Bisulfate
* Ciproquin 500	500mg	SUM15	AUG18	Ciprofloxacin Hydrochloride
colour - Titanium Dioxide				
* Ciprofloxacin Forte	500mg	MAY15	APR18	Ciprofloxacin Hydrochloride
colour - Titanium Dioxide				
* Ciprofloxacin	250mg	DEC15	NOV17	" "
PRONTOX 500				
* Ciprofloxacin	250mg	MAR15	FEB17	" "
Ultraflox - 250				
* Chlorpheniramine	4mg	APR14	MAR17	Chlorpheniramine Maleate
PIHEN Tablets				
* CINNARIZINE	20mg	JUN15	MAY18	
VERTRON				
* Cetirizine	10mg	SEP15	Aug18	Cetirizine Hydrochloride
CITARD				
colour - Titanium Dioxide				
* Clavacillin	250mg	AUG14	NOV16	Clavacillin Sodium
* Cetirizine	10mg	MAY15	APR15	Cetirizine Dihydrochloride
* Celecoxib	200mg	OCT15	OCT18	Celecoxib
SELECAP-200				
* CELECOXIB	200mg	JUN14	MAY16	CELECOXIB
COLCIBRA				
* Cefuroxime Axetil	250mg	JUL15	JUN18	Cefuroxime Axetil
ZINAXIN				

**Figure 5.2.3- Sample details for medicine for real data.**

● D Peridone Domperidone 10mg/TS 24Times 9-2014 08-2016 Quinacrine 100mg Yellow.	● B Prolonged-release Adrflam. - 12-2012 11-2016 Sunset yellow & titanium dioxide 100mg
● Cloxacillin Sodium	● Decafed-X Eparavant 1-5 mg/ml 10-2015 07-2018 Macroglyc 100ml Bronchobidator
● Mefenamic Acid	● Smaferp Suspension Mefenamic acid - 02-2016 04-2018 - 600ml
● R-Oral Prednisolone	● Cefalexin oral Lecet - 02-2015 07-2017 - 100ml
● Amonoxidin Oral	● Doxidofric Paracetamol Analgesic Oral Suspension - 02-2012 03-2016 - 100ml
● Cephalexin Cephalexin Oral	● Cefotaxim Ceftaz - 02-2015 03-2015 Sunset yellow FCP
● Maskasid Mifadotrate and Simethicone Oral Suspension 100ml	● Broncodilator Oral Suspension (G-2) 10-2015 11-2017 - 100ml
● Broncodilator Salbutamol (S-2) 10-2015 12-2015 11-2017. Oral Solution(G-2) 10-2015 11-2017.	● Altimedimethyldiglycolol Chloroform 10-2015 06-2015 05-2018 - 200ml
● Lubritabs Plus Syrup Mucolytic Bronchobidator	● Col-Kid Bronchodilator - 02-2015 01-2017 Sunset yellow FCP
● Paediatric Kelvin Syrup 10mg/5ml 60ml/250ml	● Expectorant Mucolytic Oral Liquid
● Paracetamol Oral Solution	● Cephalexin Salform - 02-2015 09-2017. Salbutamol & Bithione 100ml Blue caps
● Terbutaline Sulphate Bromhexine Hcl Extro	● Axetol Paracetamol/100mg - 02-2015 02-2016 - 100ml
● Quafenidine & menthol 5%	● Throphylline And Guaiacolfenesin - 01-2016 12-2017 - 500ml
● Cetirizine Zecet Hydrochloride oral	● Pred-kid Prednisolone - 10-2014 09-2017 Earwaxine 60ml
	● Gripe
● Medical Term	
● Nifedipine Extended Release Tablets 20mg Feb/15 JAN/18 Nifedipine 20mg	# UNIFERTE PHARMA- -OFFICIAL LABORATORIES 20mg June/2015 MAY/2017 Omlek* 20
● Niflumacin Tablets 40mg June/16 May/2018 Normal	# MOVCOL - 22/12/2015 23/12/2017 Polyethylene glycol 3350 + Sodium chloride + Sodium bicarbonate + Potassium chloride)
● Nasestatolone Tablets BP 5mg Dec/15 Nov/18 NCNormal-p	# Pantoprazole Tablets Fazel 10mg 12/2015 11/2018 - Pantoprazole Fazel Tablets
● Pantoprazole Sodium delayed release tablets 40mg mag/15	# - 20mg 09/2015 07/2018
● Pantoprazole tablets 20mg Jan/15 May/17 Pantacid 20	# Protagac-HD 10mg osfoids 04/2017
● Paracetamol tablets BP 500mg Aug/17 Regonal	# PANFRED-20 20mg June/2015 May/2017
● Theopas-Sim 500mg Nov/15 Oct/18 Aragonic	# OMMED-20 20mg 12/2015 11/2018
● Calcium Citrate Zinc 1140mg Sep/15 Feb/17 Maxical	# Neoprex® Tablets 500mg 09/2012 08/2015
● Vit D3 & Magnesium 500mg Dec/14 Nov/17 Enzyme-50	# THEOFIX-100DT 100mg 01/05/2013 30/04/2016
● Silodent Citrate Tablets 500mg Dec/14 NECASE	# TAMIGEN-E EYE/EAR DROPS 5ml June/2014 May/2016
● Glugamine, MSM, Multi Vitamin & Mineral Tablets 750mg Nov/15 Sep/17	# CAPTOL 250mg 14-

**Figure 5.2.4- Sample details for medicine for real data.**





### 5.3 Non-functional requirements

- It has the instruction menu to explain users about how to use it.
- It is available based on the user needs (work properly, do sales and purchases efficiently including safe data management of the pharmacy)
- Pharmacy system is password protected to control the system in proper way.
- Data are protected and controlled by the administrator.
- The owner(admin) can handle all the data without any restrictions.
- The users have restrictions to handle the application

### 5.4 Domain requirements

These requirements that reflect the characteristic and features of the “Pharmacy Management System” and this is new functional requirements, constraints on existing requirements or define some specific requirements.

- Our system should be safe, unbreakable and password protected and passwords are stored as encrypted form.
- Data should be integrity.
- System should allow authenticated people.
- Stock, sales, purchase and billing report should be provided to the admin.

#### Modules of the system

- Admin module
- User module
- Manage medicine module
- Manage staff module
- Manage supplier module
- Manage sales module
- Change password module.

Admin can access all the modules and user only access sales module with some restrictions.

#### Hardware Requirement

Minimum hardware and Software Requirements for System Installation  
Hardware:

- CPU - Pentium 4 2.8 GHz
- RAM - 4GB or above

- 40 GB or above hard disk
- Printer

### Software Requirements

- Windows 7, Windows 8, Windows 8.1, Windows 10
- Visual studio tool.
- A good antivirus that can either be McAfee, K7 or A firewall and the most important thing is to keep your antivirus always updated same with your operating system.

## 5.5 Challenges in getting requirements

In the requirement getting we had many problems. Because, we were frustrated with searching the suitable client for our project in Jaffna town. After that we got the clients at other district (Batticaloa and Kegalle), but, we are permitted to searching a client from jaffna town.

We talked with many owner of pharmacy, but, they did not interest to use the computer based system to manage their pharmacy. Because, they used to familiar with their manual records and some of the pharmacies have small business and resources, they told us that is easy to maintain with manual records. Some of them thought the computer system that the hardware and software requirement cost will be very high, and also this technology were new for them. Some other owners got fear to give their medicine and pharmacy details to us.

We have got more than one month of period for a suitable client (RVG Pharmacy) for our project. We implemented our application with their requirements (RVG Pharmacy) until middle stage of our project. Unfortunately, we could not work with them. After 2 months our team were connected with “ATLANTA Pharmacy”. We showed our application until we have implemented. The Administration of “ATLANTA Pharmacy” were happy to see that. And they gave some additional requirements with few corrections.

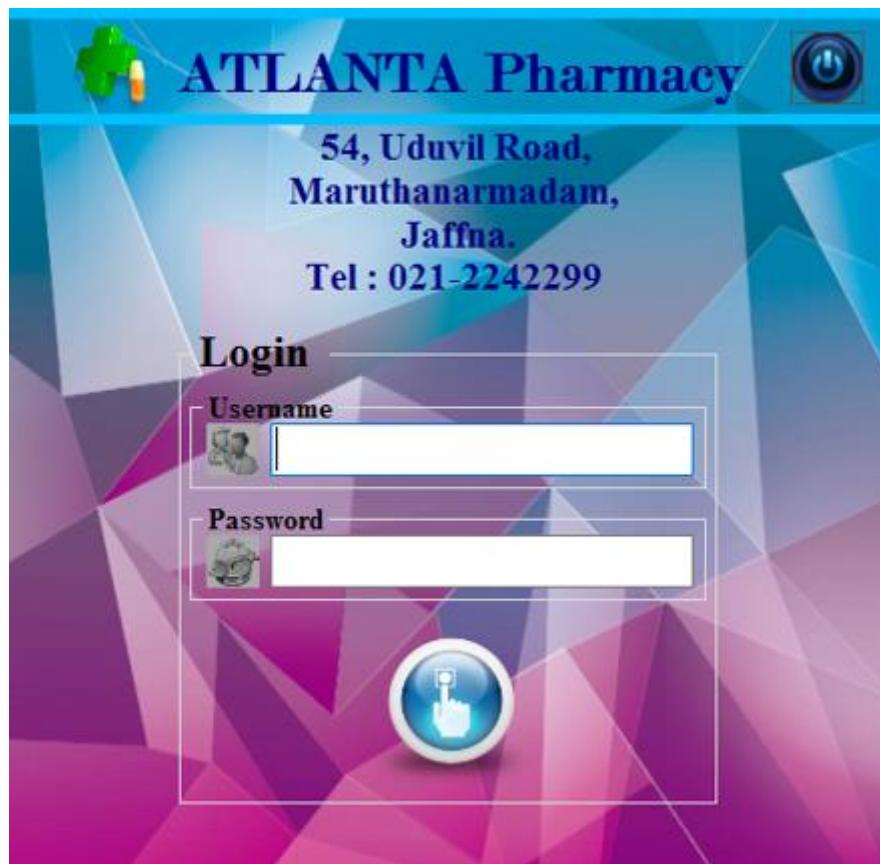
## 6 Software design

This section we expect to analyse and discuss the proposed system for the “Pharmacy Management System”. After the requirement analysis and specification, this section will give a better understanding of the system behaviour and interactions with the diagrams used. First, we want to describe the reader about the functions of the proposed system with detailed understanding.

For this purpose, we have included:

- Software Design.
- Software design approach.
- Context diagram, that defines the boundary between the system.
- Interactive diagram, that describes how a group of objects collaborate with in the use-case.
- Structural diagram, we have showed the class diagram, it is useful for showing the structure of the system.
- Behaviour diagram, here we have showed the sequence diagram to provide better clarity of the sequence of activities, and use-case diagrams to identify the main components of the proposed system.
- Data model, here we showed the ER-diagram to give an idea of database management.

## 6.1 Software design



**Figure 6.1.1- login interface**

This is a common login page for both admin and user.

The admin interface has a teal background with a grid of functional buttons. At the top left is a "Welcome" message and the text "Logged in as : Admin". To the right are two small icons: a gear and a power button. The center displays the current time "12:30:55 AM" and date "4/26/2017". On the right side, there are two tables: "Expired Stock" and "Low Stock", each with columns for mID, mName, and quantity. The "Expired Stock" table shows one row with mID \* and quantity \*. The "Low Stock" table shows four rows: M2 (Panadol, 32), M2 (Cefuroxime Ax..., 1), m61 (amoxilene, 48), and m63 (azithromycin, 41). Below these tables are four main sections: "Medicine", "Staff", "Supplier", and "Sales". Each section contains several sub-options with corresponding icons. For example, the "Medicine" section includes "Add Medicine", "View Stock", "Modify item", and "Expiry Date".

**Figure 6.1.2- admin interface**

After login as a admin this page will appear and it contained main functionalities of this application.

**Figure 6.1.3- user interface**

After login as a admin this page will appear and it contained few functionalities of this application for users.

**Figure 6.1.4- change password interface**

Only admin can access this interface to change both admin and user passwords.



The screenshot shows the 'Add New Medicine' interface. At the top right are icons for back, forward, and search. The title 'Add New Medicine' is centered above a form. The form contains two columns of input fields. The left column includes: Category (dropdown), Medicine Id (text input), Medicine Name (text input), Batch No (text input), Dosage (text input), Manufacturer (text input), Imported By (text input), and Brand Name (text input). The right column includes: Supplier ID (text input), Generic Name (text input), Quantity (text input), Entry Date (date input set to 4/26/2017), Manufacture Date (date input set to 4/26/2017), Expiry Date (date input set to 4/26/2017), Buying Price (text input), and Selling Price (text input).

**Figure 6.1.5- add medicine interface**

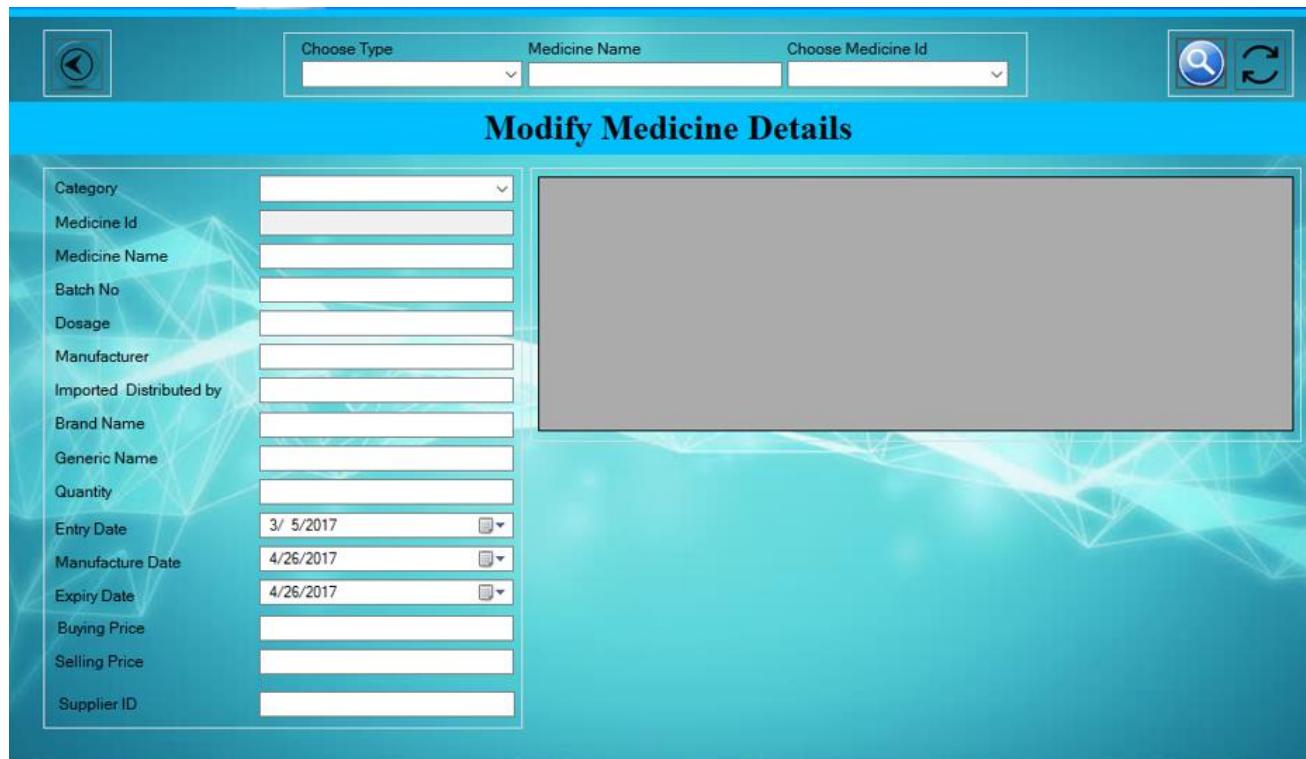
This interface only accessed by admin, user cannot access it



The screenshot shows the 'View Stock' interface. At the top right are icons for back, forward, and search. The title 'View Stock' is centered above a large gray rectangular area representing the stock view. Above this area are two input fields: 'Choose Type' (dropdown) and 'Medicine Name' (text input), followed by a search icon.

**Figure 6.1.6- view stock interface**

This interface can access by both user and admin it will show details of particular medicine.



This screenshot shows the 'Modify Medicine Details' interface. At the top, there are three search/filter fields: 'Choose Type' (dropdown), 'Medicine Name' (text input), and 'Choose Medicine Id' (dropdown). To the right of these are a magnifying glass icon and a refresh/circular arrow icon. Below the header, the title 'Modify Medicine Details' is centered. On the left side, there is a vertical list of medicine details with corresponding input fields. The fields include: Category (dropdown), Medicine Id (text input), Medicine Name (text input), Batch No (text input), Dosage (text input), Manufacturer (text input), Imported Distributed by (text input), Brand Name (text input), Generic Name (text input), Quantity (text input), Entry Date (date input set to 3/ 5/2017), Manufacture Date (date input set to 4/26/2017), Expiry Date (date input set to 4/26/2017), Buying Price (text input), Selling Price (text input), and Supplier ID (text input). The background features a blue gradient with a network-like geometric pattern.

**Figure 6.1.7- modify medicine interface**

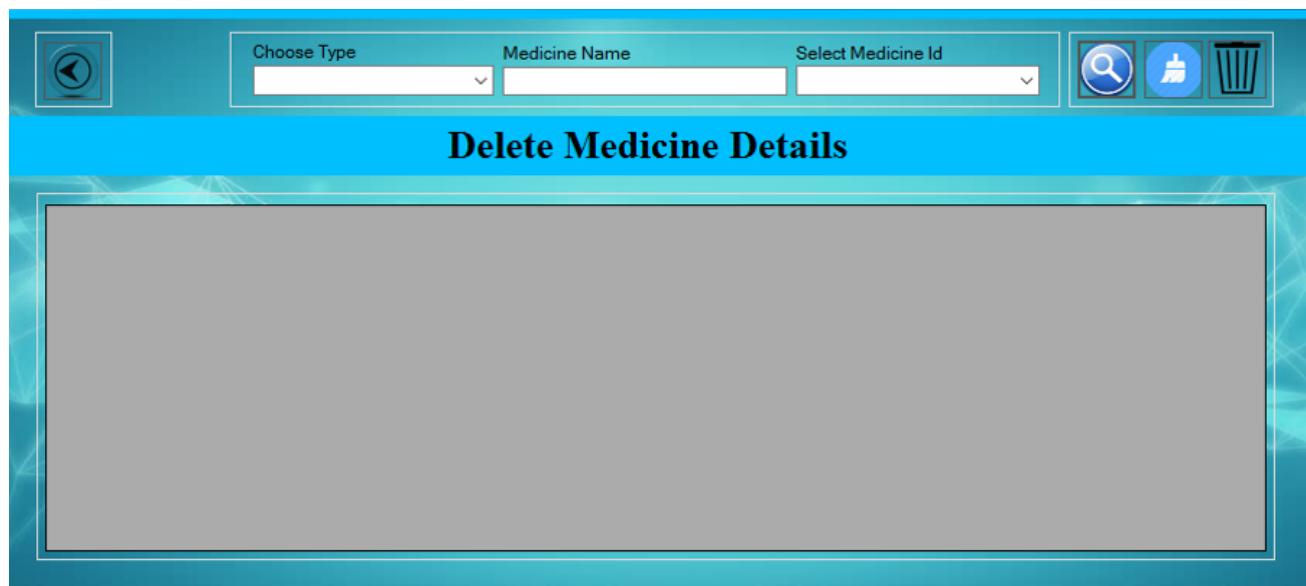
This interface only accessed by admin, user cannot access it



This screenshot shows the 'View Expiry Date' interface. At the top, there is a search/filter field labeled 'Expiry Date before the Date' containing the value '4/26/2017'. To the right of this field are a magnifying glass icon and a trash bin icon. Below the header, the title 'View Expiry Date' is centered. A large gray rectangular area occupies the main body of the page, likely intended for displaying results. The background features a blue gradient with a network-like geometric pattern.

**Figure 6.1.8- view expiry date interface**

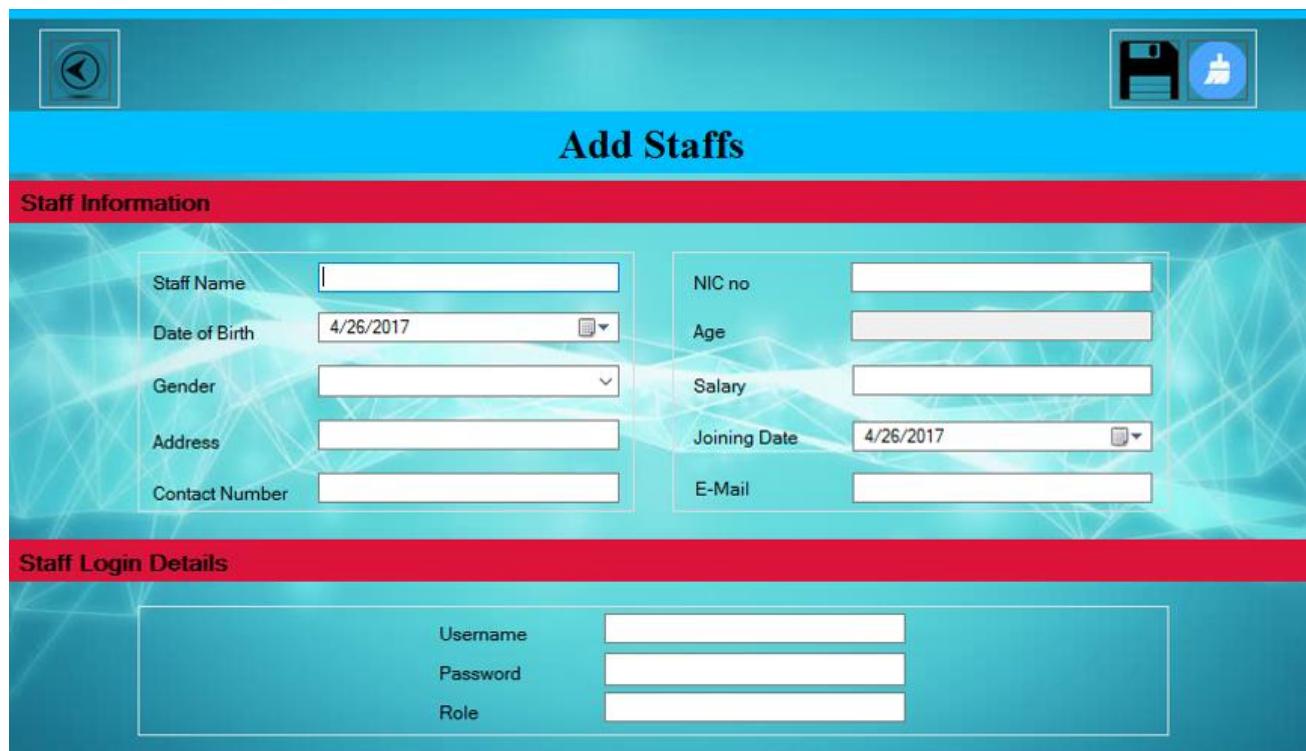
This interface only accessed by admin, user cannot access it,



The screenshot shows the 'Delete Medicine Details' interface. At the top, there are search and filter fields: 'Choose Type' (dropdown), 'Medicine Name' (text input), 'Select Medicine Id' (dropdown), and three icons: a magnifying glass, a trash can, and a clipboard. Below these is a large, empty rectangular area for displaying results.

**Figure 6.1.9- delete medicine details interface**

Admin can delete medicine details in this application.



The screenshot shows the 'Add Staffs' interface. It has two main sections: 'Staff Information' and 'Staff Login Details'. The 'Staff Information' section contains fields for Staff Name, Date of Birth, Gender, Address, Contact Number, NIC no, Age, Salary, Joining Date, and E-Mail. The 'Staff Login Details' section contains fields for Username, Password, and Role.

Staff Information	
Staff Name	<input type="text"/>
Date of Birth	<input type="date" value="4/26/2017"/>
Gender	<input type="text"/>
Address	<input type="text"/>
Contact Number	<input type="text"/>
NIC no	<input type="text"/>
Age	<input type="text"/>
Salary	<input type="text"/>
Joining Date	<input type="date" value="4/26/2017"/>
E-Mail	<input type="text"/>

Staff Login Details	
Username	<input type="text"/>
Password	<input type="text"/>
Role	<input type="text"/>

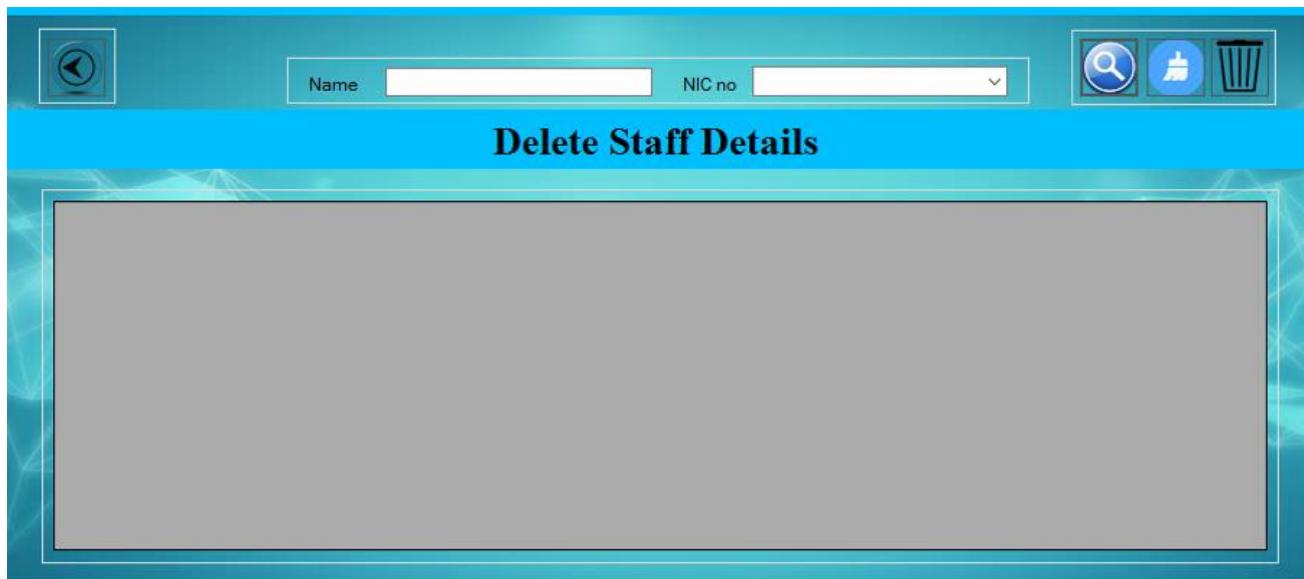
**Figure 6.1.10- add staff interface**

This interface can be accessed by admin he can make change in staff details



**Figure 6.1.11- view staff details interface**

Only admin can access this features.



**Figure 6.1.12- delete staff details interface**

Only admin can access this features and delete staff details.



The screenshot shows the 'Modify Staff Details' interface of a Pharmacy Management System. At the top, there are search fields for 'Staff Name' and 'Staff NIC' with a magnifying glass icon and a refresh/circular arrow icon. Below the title 'Modify Staff Details', there are two columns of input fields. The left column includes fields for Staff Name, Date of Birth (set to Wednesday, April 26, 2017), Gender, Address, Contact Number, NIC no., Age, Salary, Joing Date (set to Wednesday, April 26, 2017), and E-Mail. The right column includes fields for Username, Password, and Role. A large gray rectangular area is positioned to the right of the input fields.

**Figure 6.1.13- modify staff details interface**

Only admin can modify staff details in this application.

The screenshot shows a software application window titled "Add Supplier". At the top left is a back arrow icon, and at the top right are icons for saving (floppy disk) and refreshing (blue circle with a white arrow). The main area is titled "Add Supplier" in bold black font. Below the title is a form with the following fields and input boxes:

Supplier Id	<input type="text"/>
Supplier Name	<input type="text"/>
Company Name	<input type="text"/>
Licence Number	<input type="text"/>
E-Mail	<input type="text"/>
Contact Number	<input type="text"/>
Address	<input type="text"/>

**Figure 6.1.14- add supplier details interface**

Admin can add the supplier details in this system.



The screenshot shows the 'View Supplier Details' interface. At the top, there are search fields for 'Supplier Name' and 'Supplier ID' with a magnifying glass icon for searching. Below the header, the title 'View Supplier Details' is centered. A large gray rectangular area is present, likely for displaying supplier information.

**Figure 6.1.15- view supplier details interface**

Only admin can access this features.



The screenshot shows the 'Modify Supplier Details' interface. At the top, there are search fields for 'Supplier Name' and 'Select Supplier Id' with a magnifying glass and refresh icon. Below the header, the title 'Modify Supplier Details' is centered. On the left, there is a vertical list of form fields with input boxes: Supplier Id, Supplier Name, Company Name, Licence Number, E-Mail, Contact Number, and Address. A large gray rectangular area is present on the right, likely for displaying supplier information.

**Figure 6.1.16- modify supplier details interface**

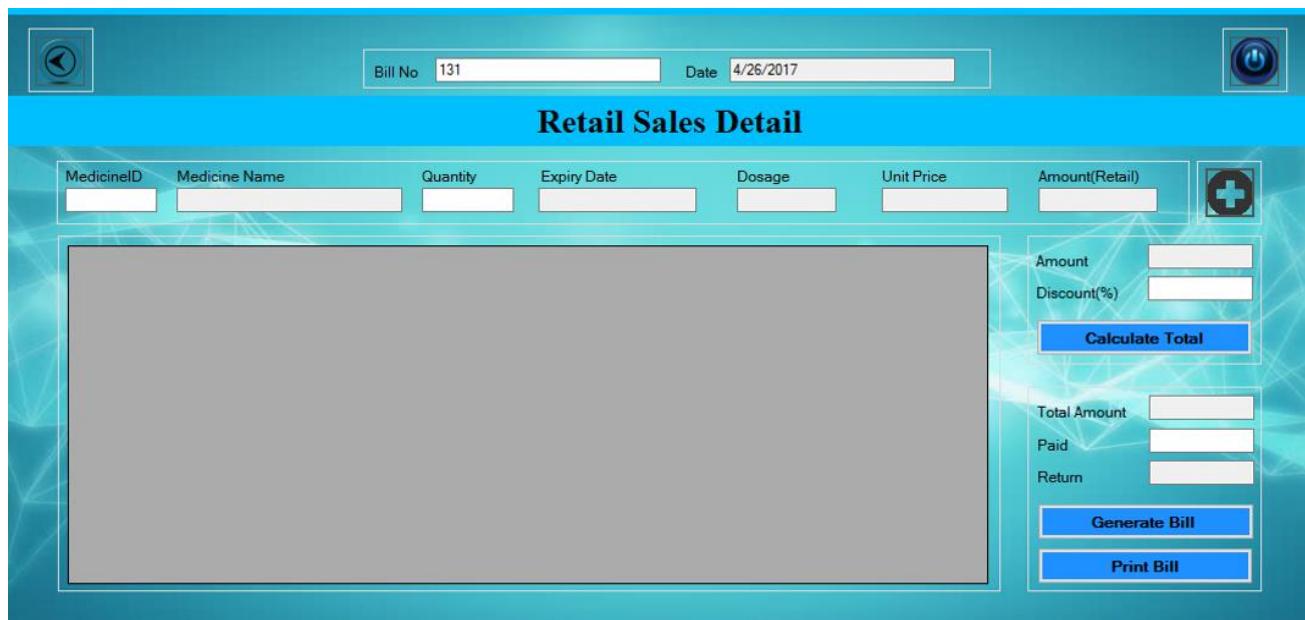
Only admin can access this features.



The interface for deleting supplier details features a blue header bar with the title "Delete Supplier Details". Below the header are two input fields: "Supplier Name" and "Supplier Id", each with a dropdown arrow. To the right of these fields are three icons: a magnifying glass, a blue circular arrow, and a trash can. The main body of the interface is a large gray rectangular area.

**Figure 6.1.17- delete supplier details interface**

Only admin can access this features.



The interface for retail sales details has a blue header bar with the title "Retail Sales Detail". It includes input fields for "Bill No" (131) and "Date" (4/26/2017). Below the header are several input fields: "MedicineID", "Medicine Name", "Quantity", "Expiry Date", "Dosage", "Unit Price", and "Amount(Retail)". To the right of these fields is a blue plus sign icon. The main body of the interface is a large gray rectangular area. On the right side, there are additional input fields for "Amount" and "Discount(%)", a "Calculate Total" button, and another set of fields for "Total Amount", "Paid", and "Return", along with "Generate Bill" and "Print Bill" buttons.

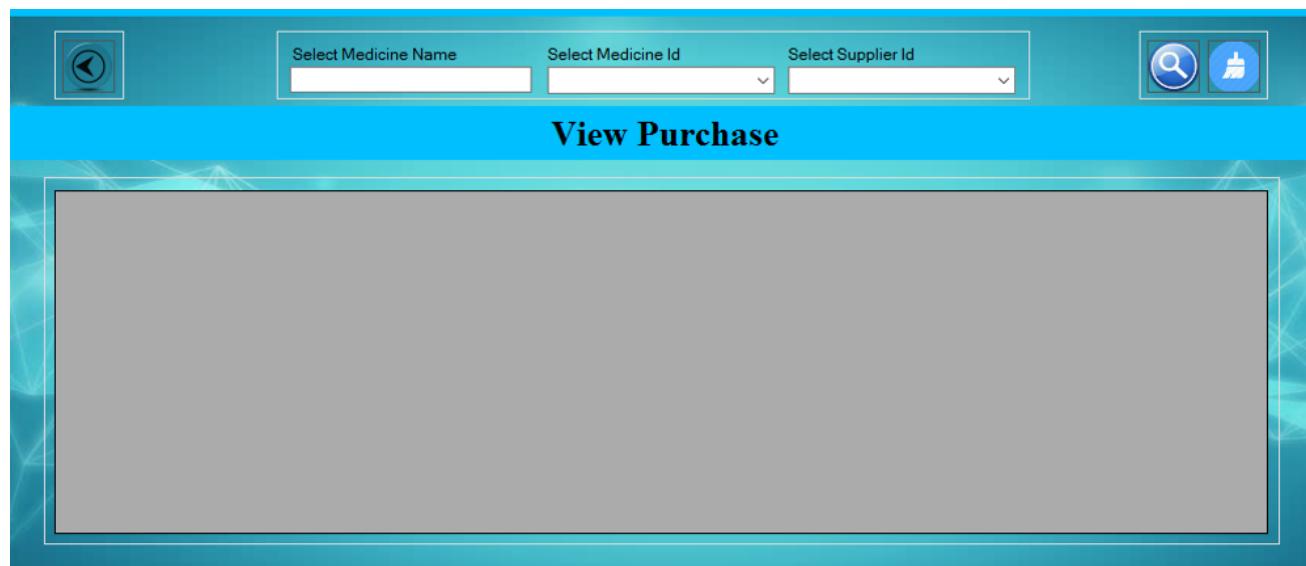
**Figure 6.1.18- retail sales details interface**

Both user and admin can access this features.



**Figure 6.1.19- view sales details interface**

Both user and admin can access this features



**Figure 6.1.20- view purchase details interface**

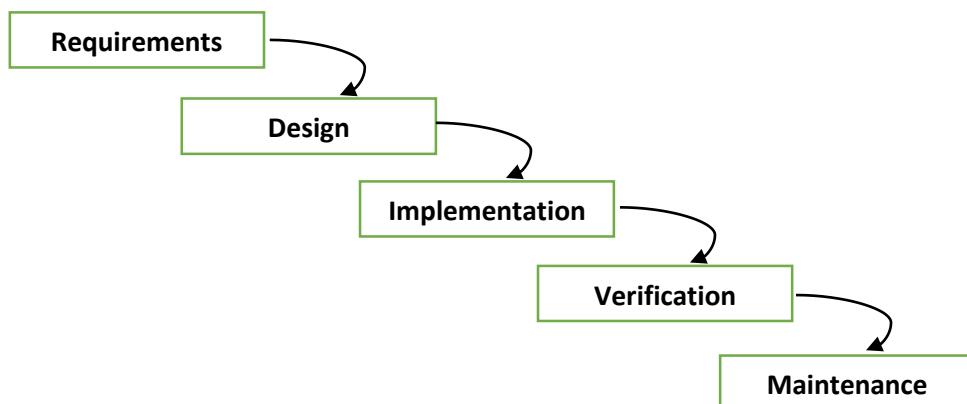
Both user and admin can access this features

## 6.2 software design approach.

In the software design, we first designed the interface with the corresponding functional and non-functional requirements. And here we are using the Top-down approach to design and implement the system.

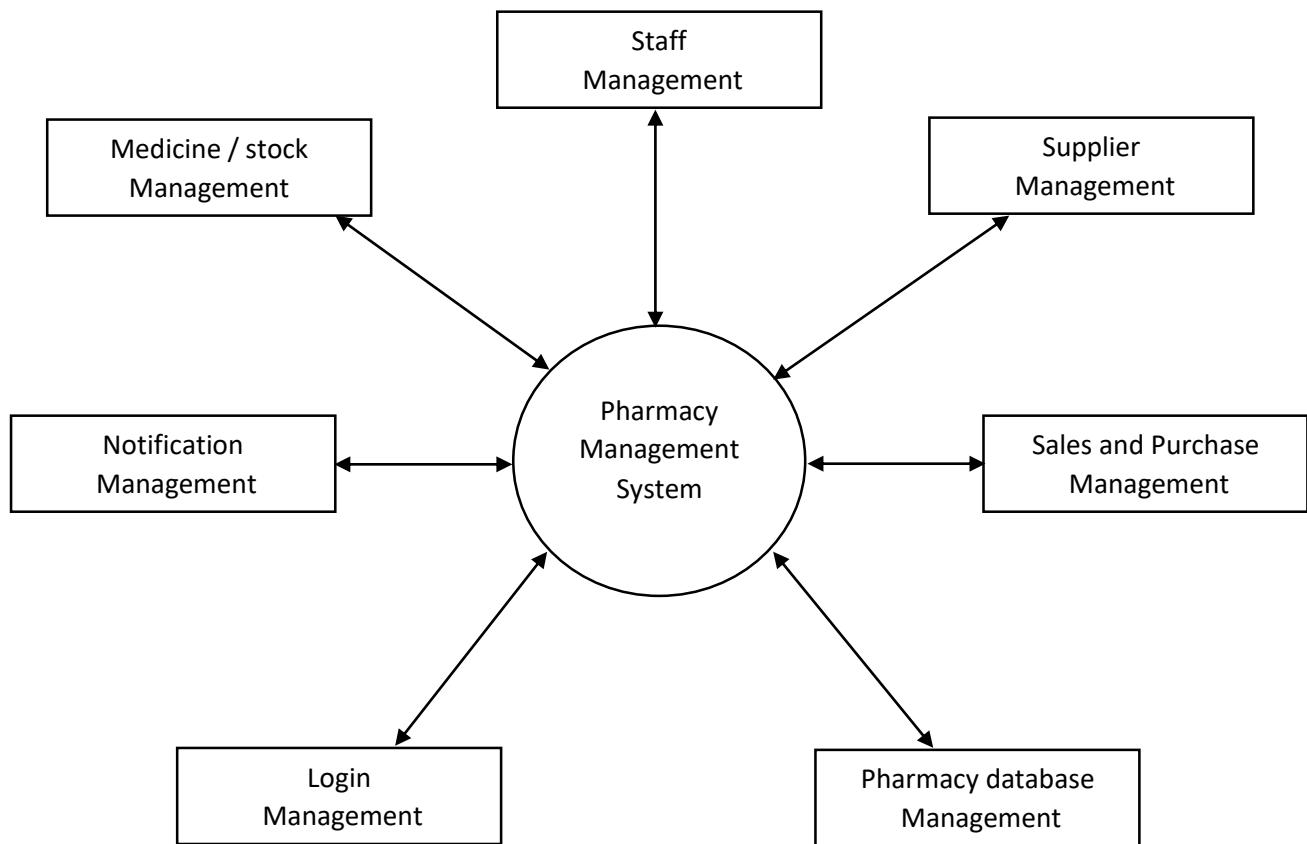
Top-down design approach starts by identifying the major modules of the system. Decomposing them into their lower level modules and iterating until the desired level of detail is achieved. This is stepwise refinement, starting from an abstract design in each step the design is refine to a more concrete level, until we reached a level where no more refinement is needed and the design can be implemented directly.

- C# library functions.
- Case studies based on our project.
- Top-down design and structure chart
- Basic concepts about functions.
- Building programs from existing information.



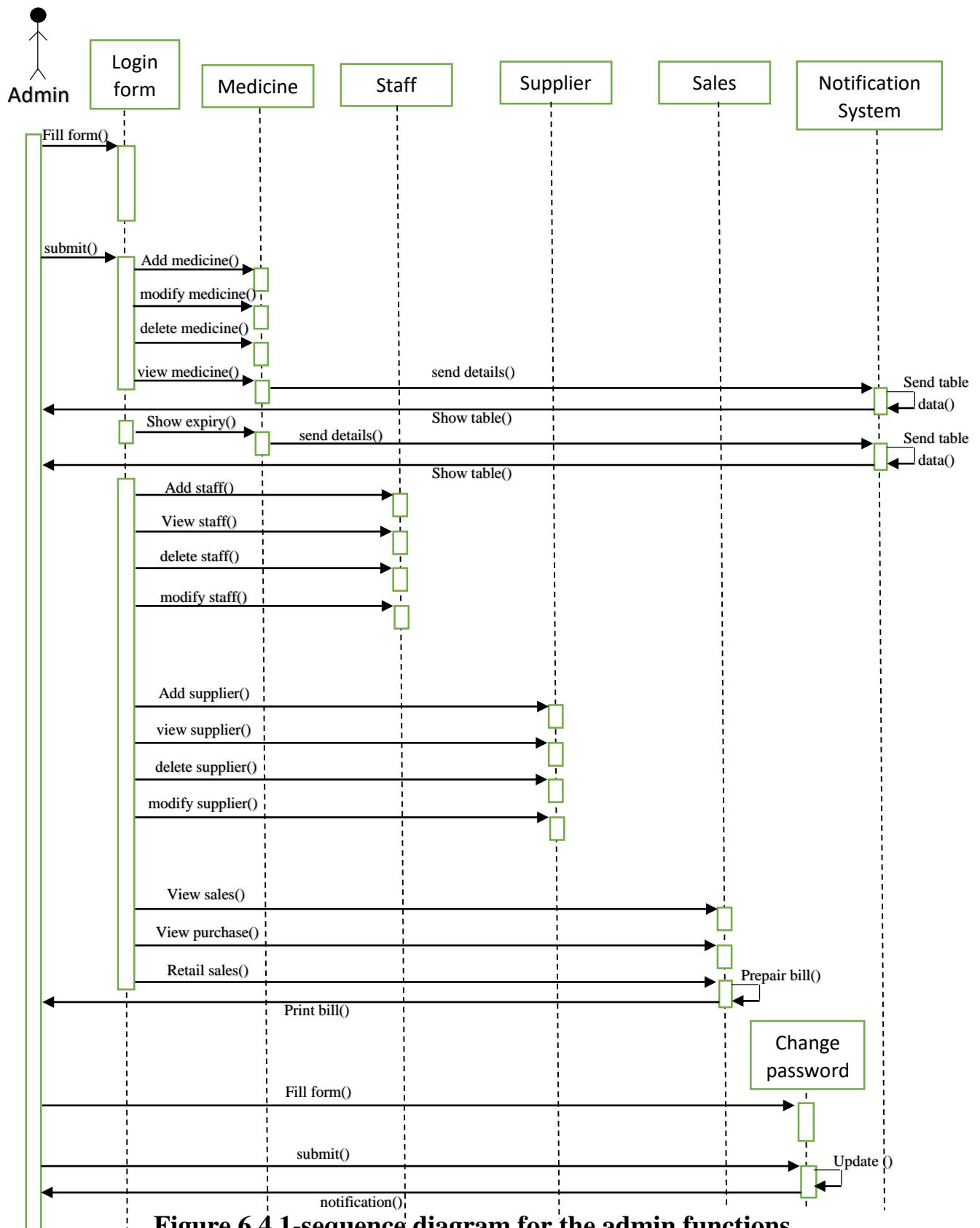
**Figure 6.2.1.1-software engineering approach.**

### 6.3 Context diagram

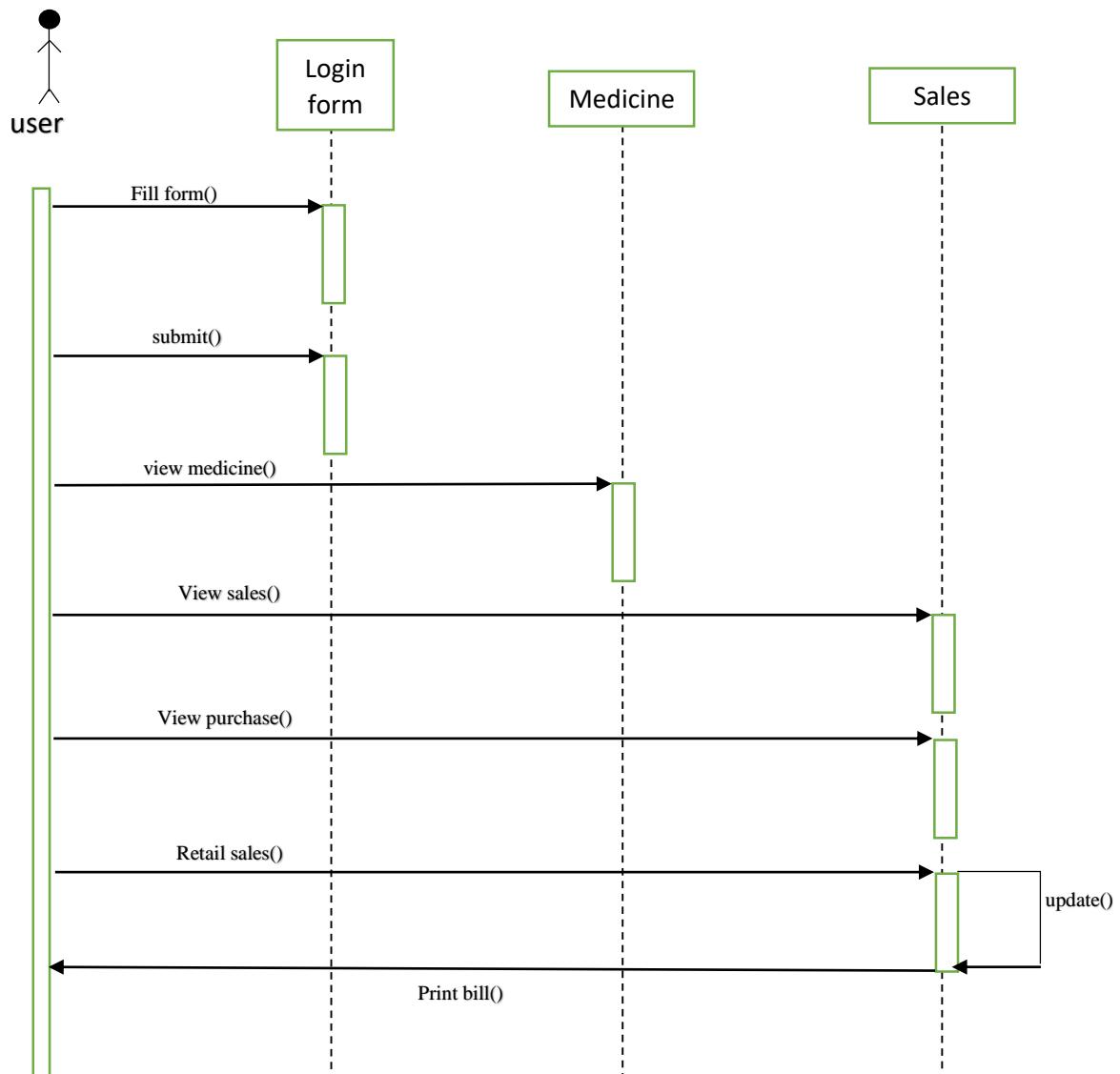


**Figure 6.3.1.-context diagram for the system.**

## 6.4 Interaction diagram



**Figure 6.4.1-sequence diagram for the admin functions**



**Figure 6.4.1-sequence diagram for the user functions**

## 6.5 Structural Diagram

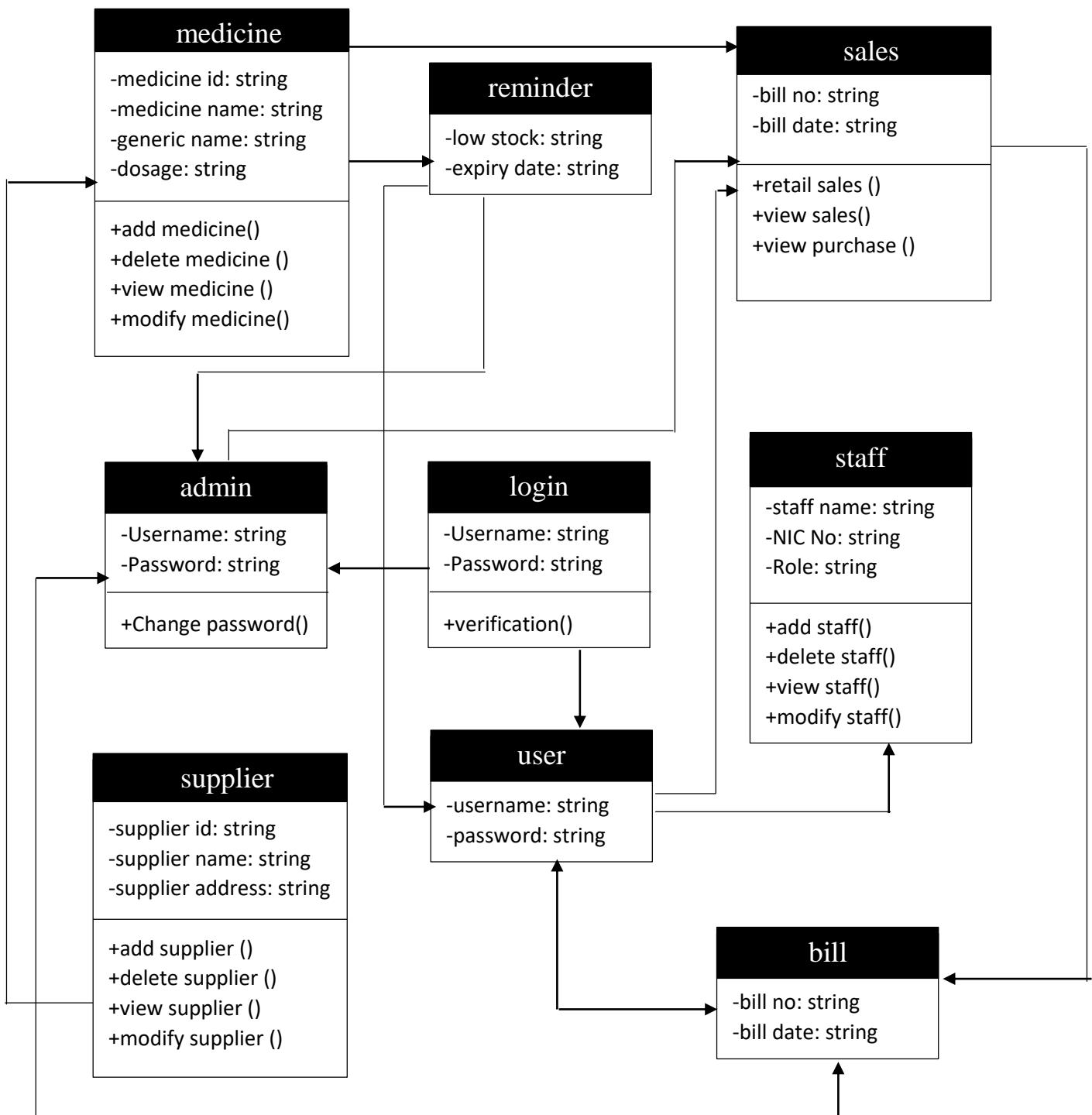


Figure 6.5.1-UML class diagram for the system

## 6.6 Behaviour Diagram

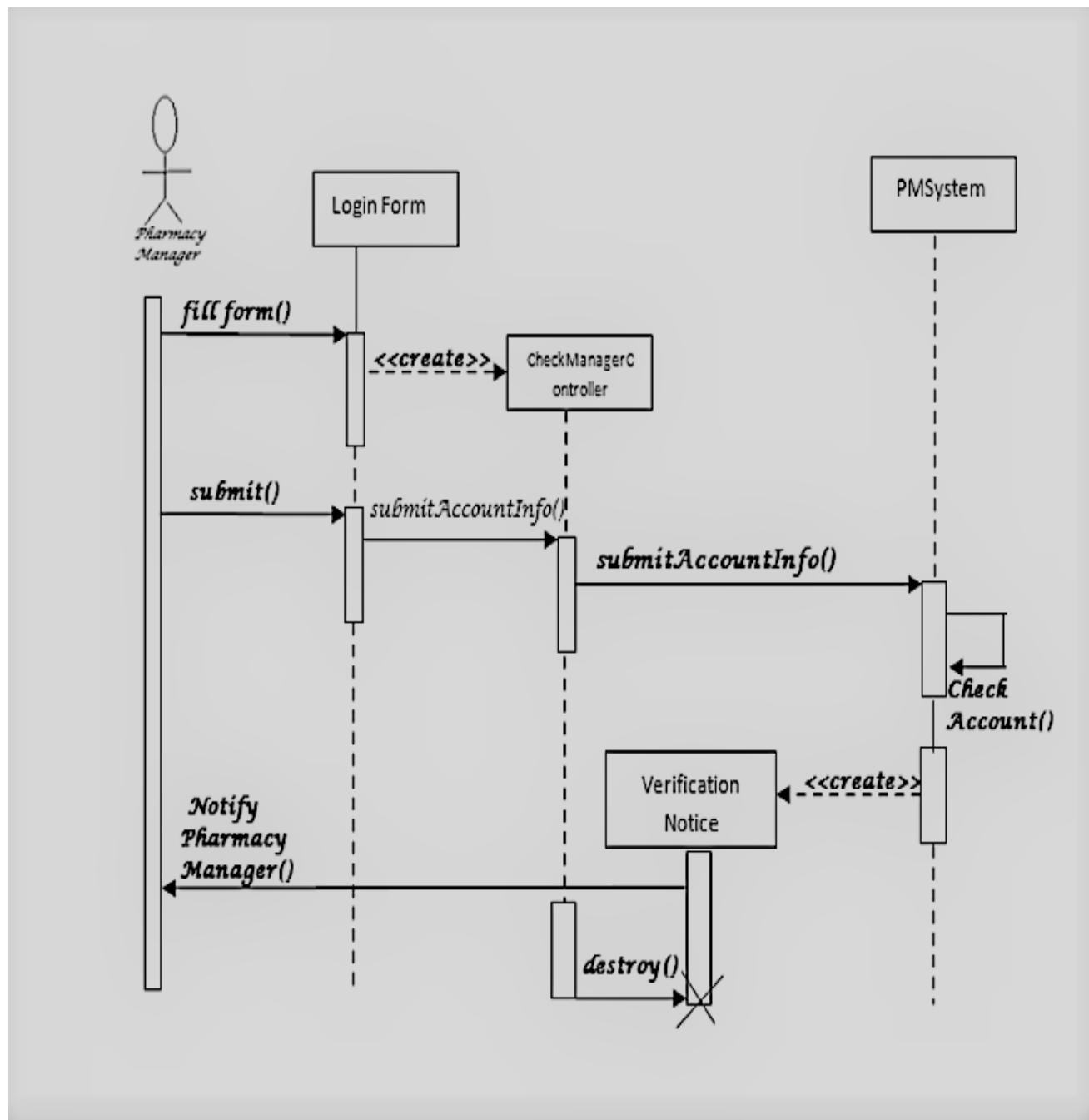


Figure 6.6.1-sequence diagram for login page

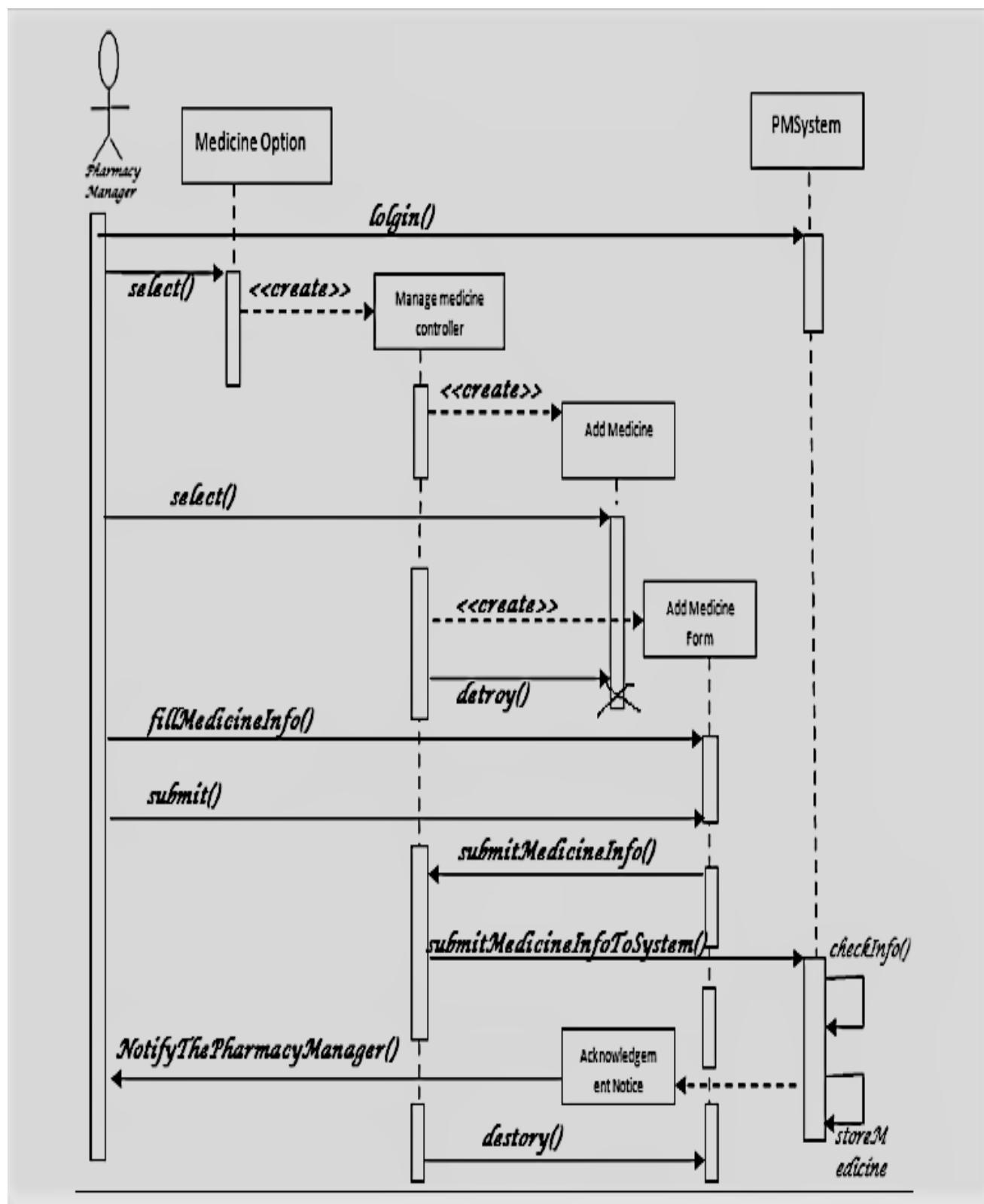


Figure 6.6.2-sequence diagram for add medicine page

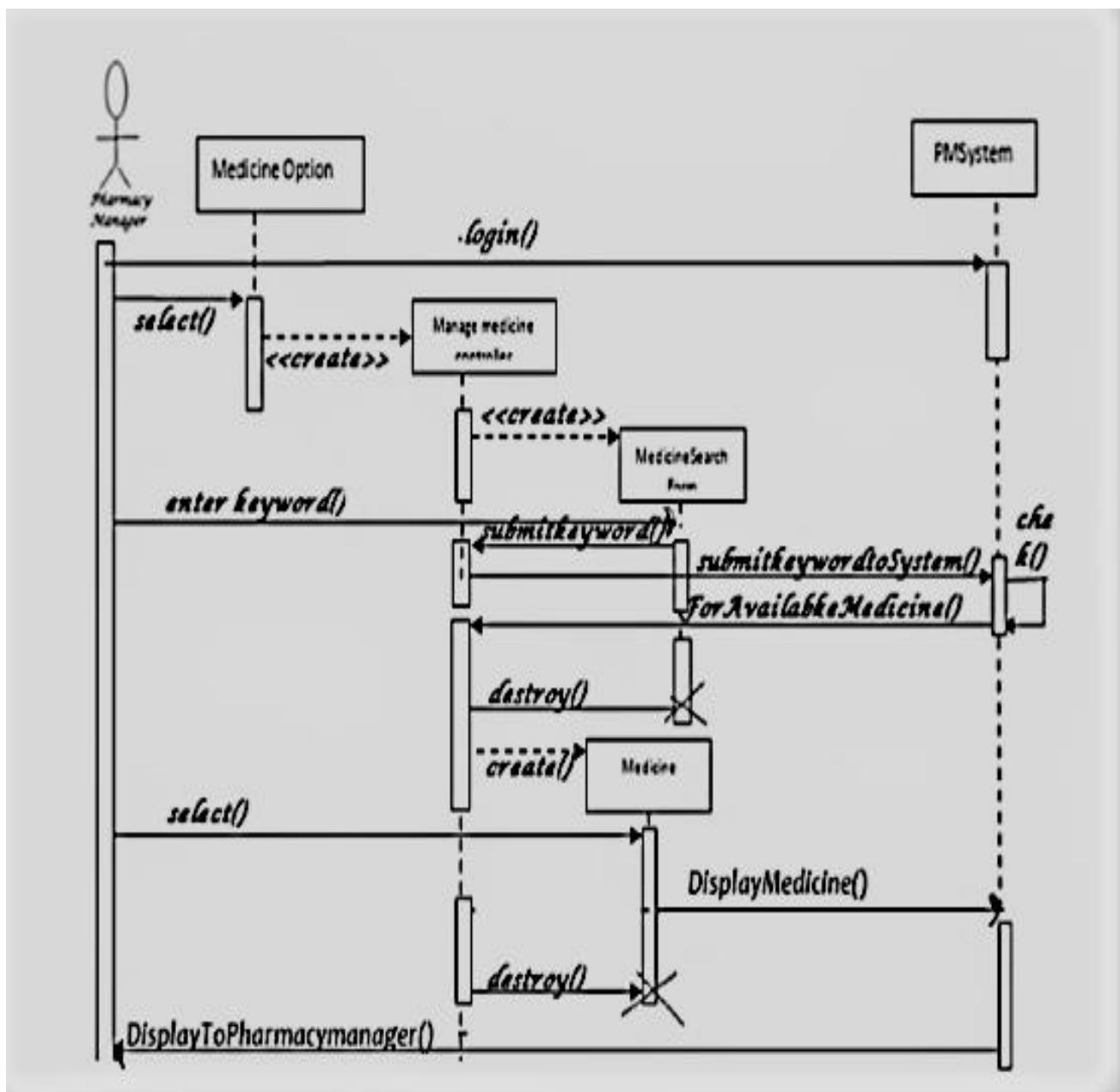


Figure 6.6.3-sequence diagram for search medicine page

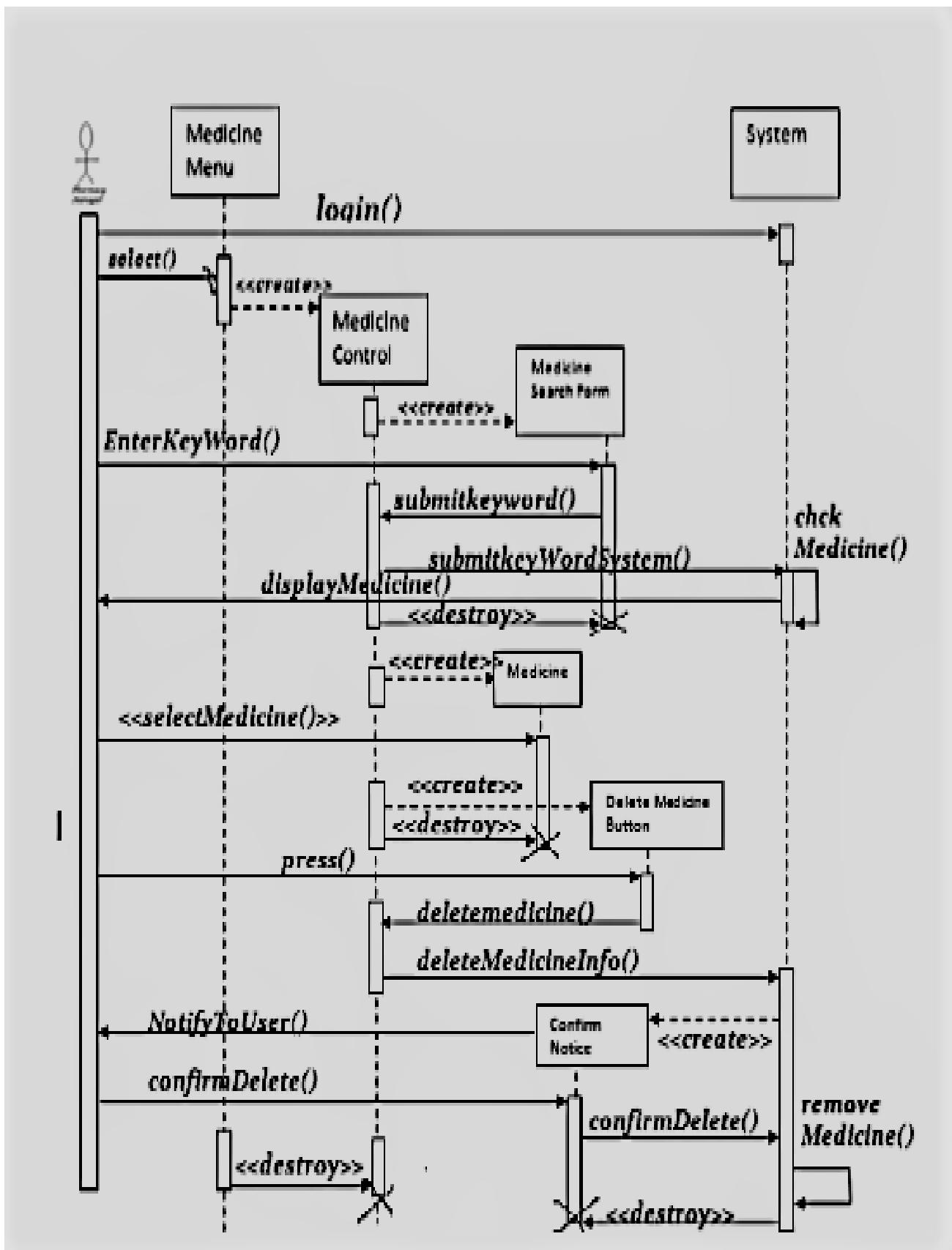


Figure 6.6.4-sequence diagram for delete medicine page

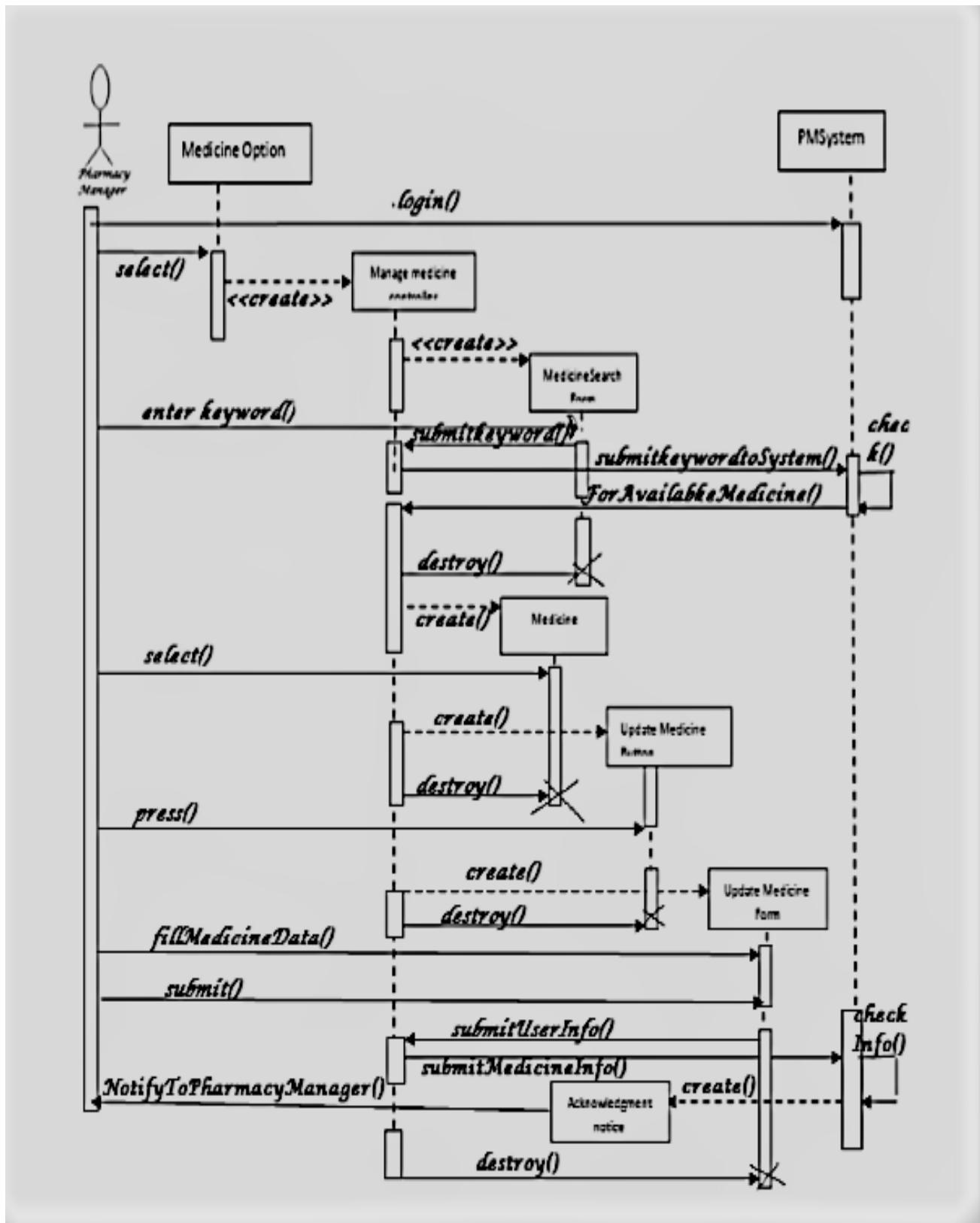


Figure 6.6.5-sequence diagram for modify medicine page

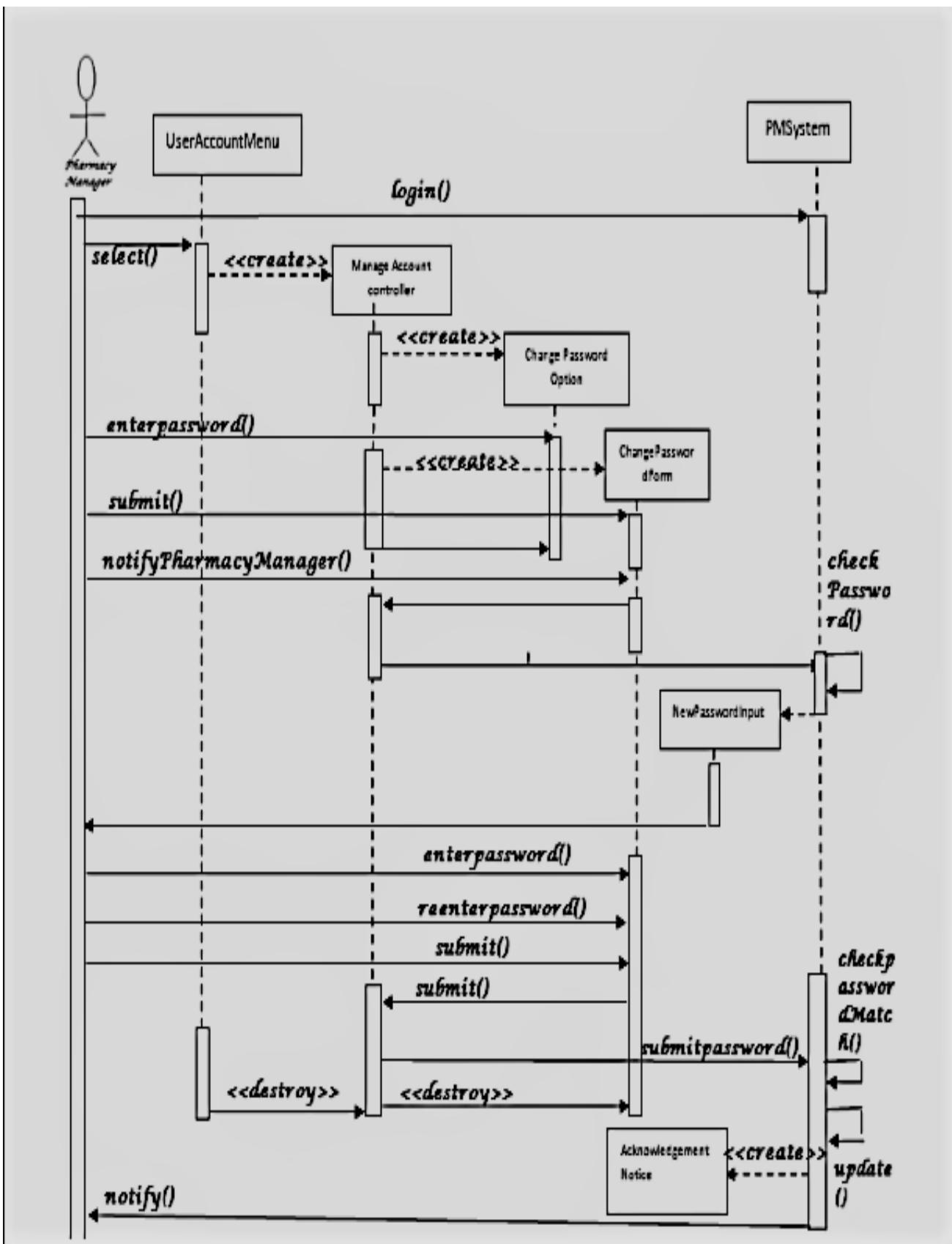


Figure 6.6.6-sequence diagram for change password page

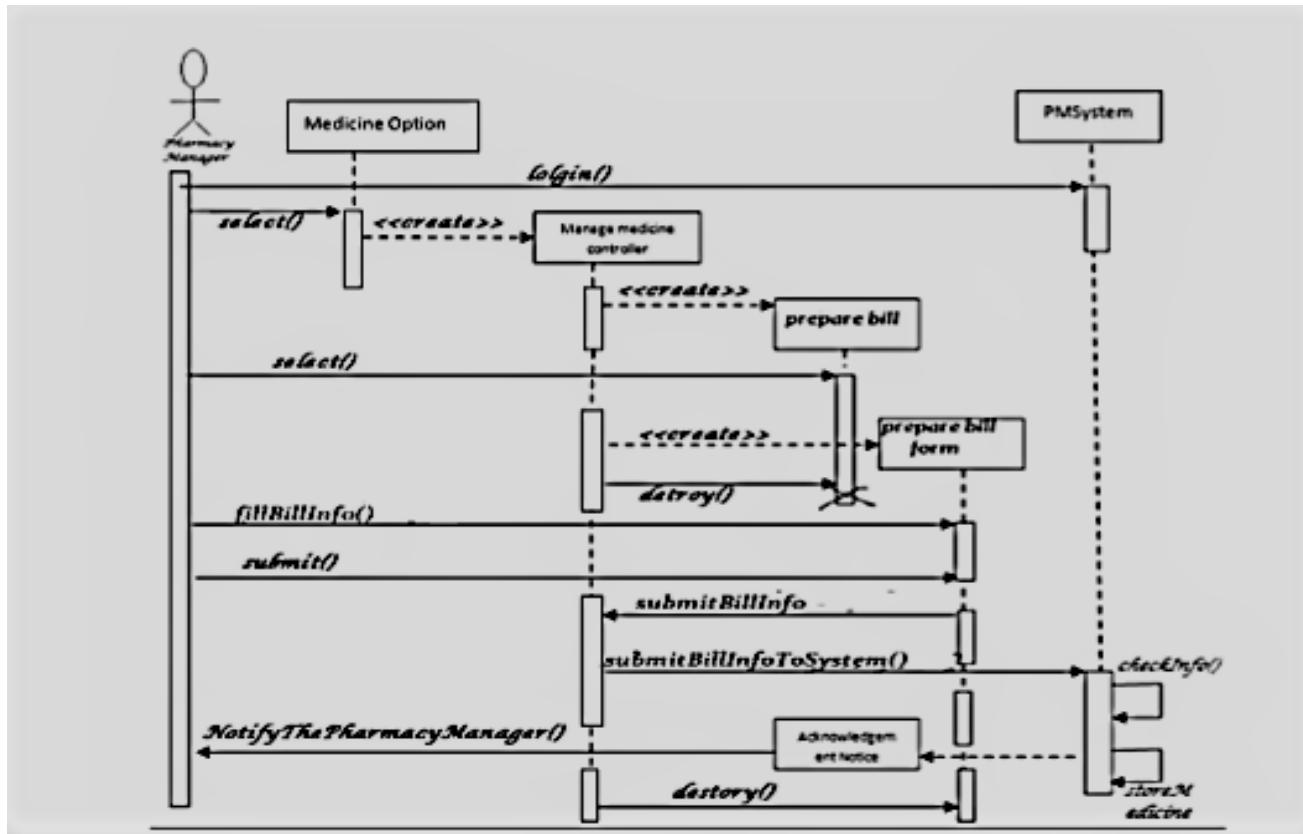


Figure 6.6.7-sequence diagram for billing section page

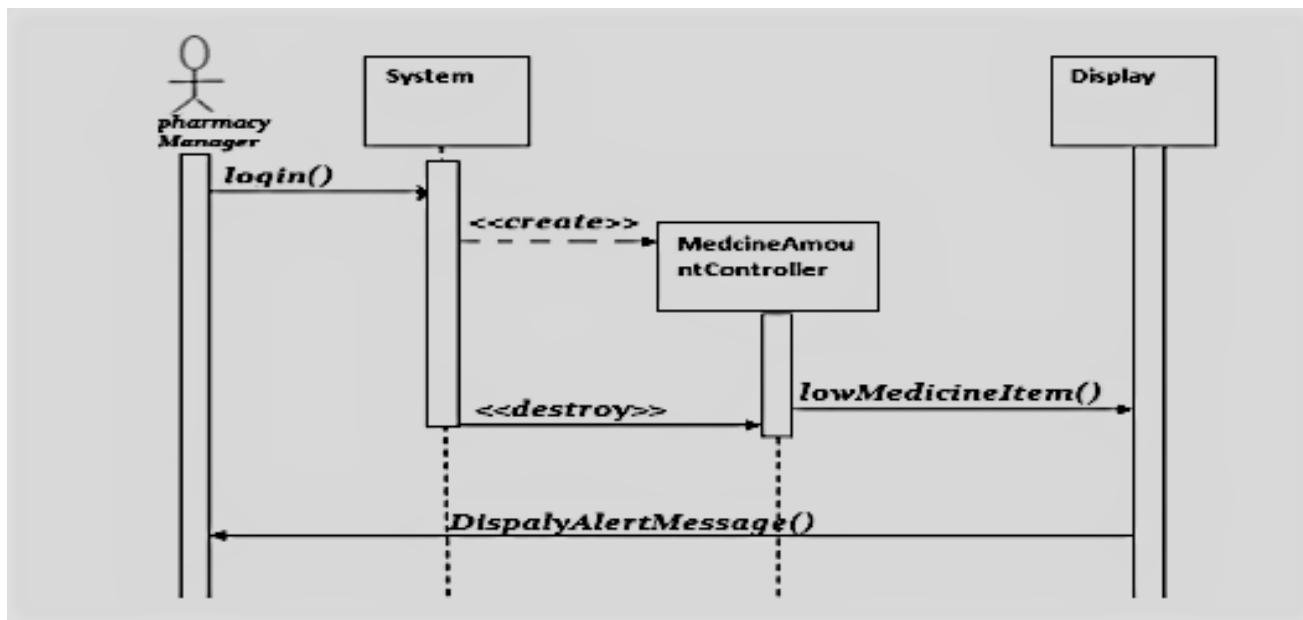
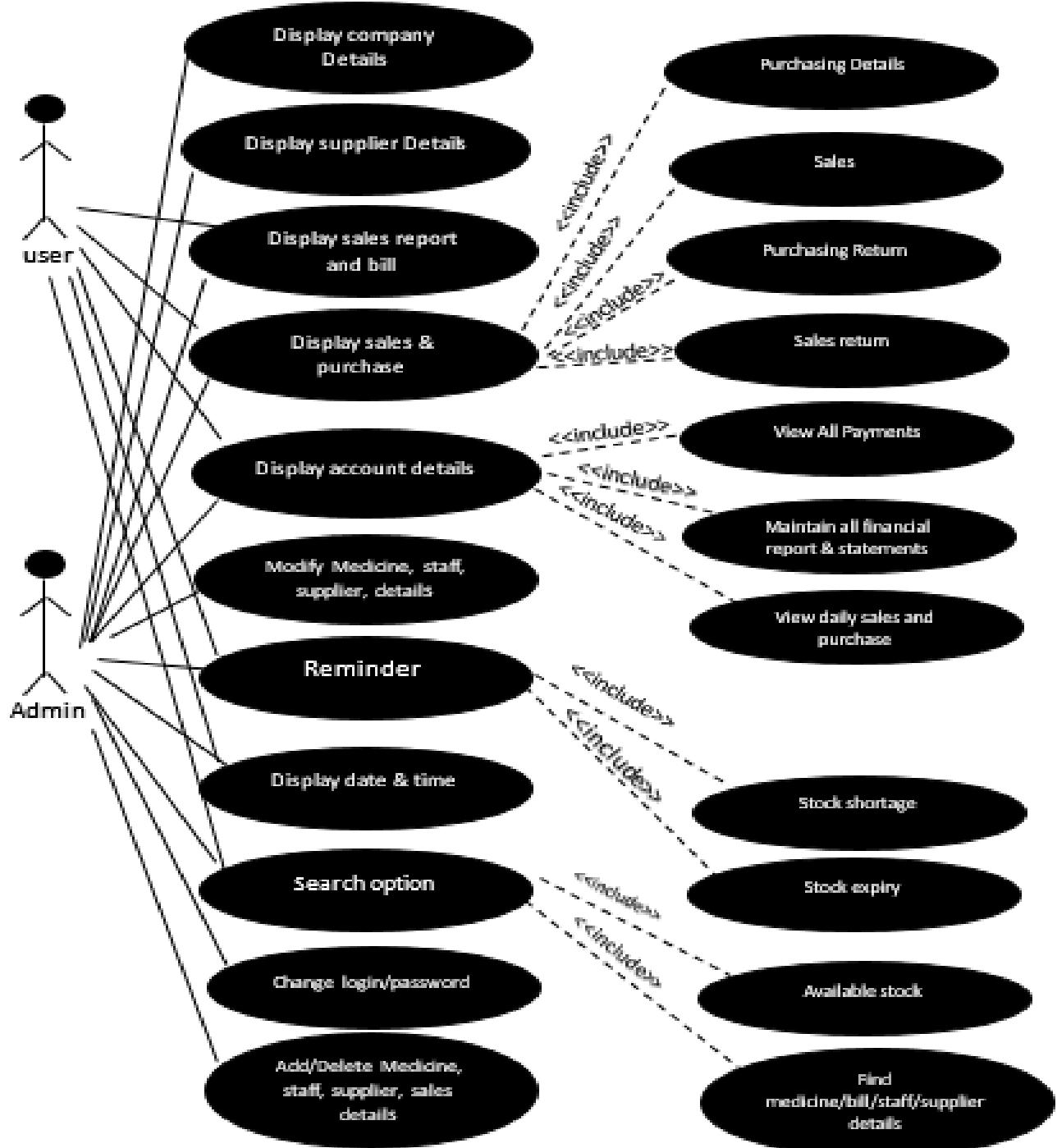


Figure 6.6.8-sequence diagram for reminder page



**Figure 6.6.9-use-case diagram for the system**

## 6.7 Data Modal

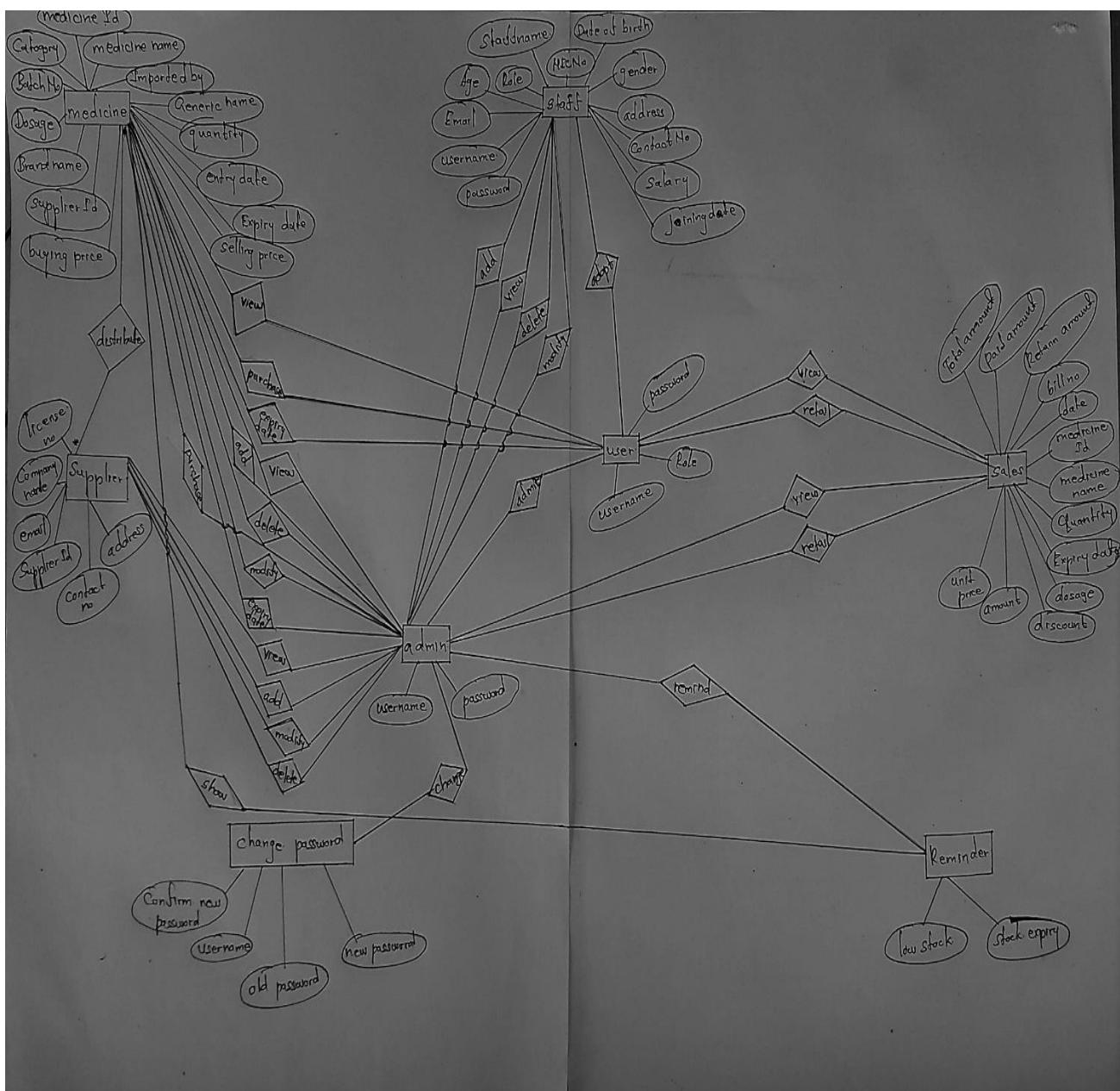


Figure 6.7.1-ER diagram for the system

## 6.8 Challenges we faced in designing

The course of software engineering is new to us and software implementation and system designing are new to our group, in early days we did not hear these kind of terms after that we realised in designing part of the project we have to consider requirements, software design approaches, software design.

First time, we have done a standalone system for a group project in our life time. For this project purpose we learned lot of things in short time and this is little toff to us. But, we managed that, and learned that, what we need for the purpose of our group activity.

In the middle of the implementation process, our client suggested to change some functional requirements of the software design and it caused some change in the application, these changes took our valuable time to redesign our application.

And we spent lot of time to software designing part. Because, we plan to design stylish, good looking and neat software interface.

Mainly, these kind of challenges we faced during the software design. Addition to that some little issues we faced like group management.

## 7 Implementation

This section describe how we implemented our system with the requirements. Here we described about what kind of techniques and tools we used, what are the special features in our application and the challenges we faced during the implementation. This section should give detailed understanding of implementation part.

- The application has two main modules that are admin and user modules. Only admin can make changes or modify the details of medicine, staff, supplier and sales modules. And user access only the sales modules.
- Some of the staffs can act as user with admin's knowledge, only admin can change or create the password in the change password module.
- The application will show the details about the low stock and expired stock in the table in admin module.
- Finally, this application will produce an invoice for each sale, and this system will record each sales and purchases according to the date.

Each and every modules of the application were implemented using above outlines that we explained.

### 7.1 Tools and techniques, and justification

- We have used C# programming language for our designing and implementation process and visual studio as a platform.
- Object-oriented concept is used in this application.
- SQL server is used for connecting database with our application to the process

#### C# programming language

C# is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines.

- The language is intended to be a simple, modern, general-purpose, object-oriented programming language.
- The language is intended for use in developing software components suitable for deployment in distributed environments.
- Portability is very important for source code and programmers, especially those already familiar with C and C++.
- Support for internationalization is very important.

- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.
- Although C# applications are intended to be economical with regard to memory and processing power requirements, the language was not intended to compete directly on performance and size with C or assembly language

## **Microsoft SQL server**

“Microsoft SQL Server” is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications-which may run either on the same computer or on another computer across a network (including the Internet).

## **Object oriented concept**

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. There is significant diversity of OOP languages, but the most popular ones are class-based, meaning that objects are instances of classes, which typically also determine their type.

In our application medicine and sales modules are containing the main functions. In medicine module, add, delete and show expiry are main functionalities and sales module, retail sale is important functionality. And here, following are the coding for these functionality.

## 7.2 Special features

In visual studio 2010

- In built, designing and implementation tools
- Debugger
- Inbuilt C# library for implementation
  - using System.Security.Cryptography;(for hashing password).
  - using System.Text.RegularExpressions;(for email, phone no and NIC no validation).
  - using Pharmacy\_Management\_System.Properties;(for using the properties of project).
  - using System.Collections.Specialized;(for suggestion append of combo box).
  - using System.Configuration;(for system configuration).
- SQL server to connect the database.

Above special features of the visual studio 2010, we have used for our project.

In our application

- According to the special features this application will check all the validation of particular fields for example e-mail, telephone number, NIC No, string and character validations of each textbox and number of inputs of each textbox.
- In billing section, we automatically find the medicine name, dosage, unit price with typing the medicine id and it will automatically calculate the total amount with the discount and balance, finally keep the records of sales.

### 7.3 Challenges in implementation

For the implementation of this application, we already decided to do a standalone system, but we have two choices to choose which programming language we want to use that are java and C#. After our group discussion, we finalised to go for C# programming language. We interested to learn a new programming language. But, some members of our group not ready to learn and two of them learned this language.

Here, we mentioned that the programming language we used for this project is new to us. But, we have already study and familiar with java and C# is little different from java, we have used object-oriented concept to build our application this is our first challenge.

Another thing was version problem of our platform, in early stage we used visual studio 2015 to implementation. Some of issues raised during the implementation that are difficult to connect database, sometimes connected database was not working properly, SQL server problem, it got load to start and some other little issues. Finally, we used visual studio 2010.

Real data of the stock was the challenge to us. We got the requirements to implement but did not get the real data to check.

And many conflicts raised between group members based on coding work during the implementation in finding correct methods to improve our application.

## 8 Testing

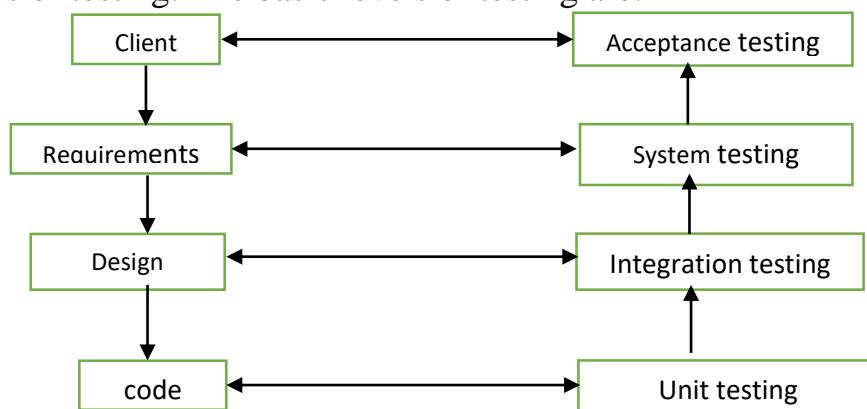
Testing is done in order to align the product with the user. The functionalities and functions are tested in order to point out the defects and errors that were made during the development phases so that the quality of the product and the functionalities of the product are improved. The testing plan defines the items to be tested and the functions are selected bases on the importance of the functions, and the risk of the functions on the user's viewpoint.

The test cases were designed corresponding with the use case descriptions. They were executed manually and the results were recorded. The bugs identified were corrected and tested again.

In the testing process of this system both black box testing and white box testing is being used.

- **White box testing** is mostly carried out while the coding is taking place. Each statement, conditions and decision structures test soon after finishing the coded. The purpose of using white box testing for this system is that White box testing enables to identify the small bugs that could be created by the mistakes of coding and due to the errors in the decision logics, and it enables the developers to prevent those errors before integrating the recently developed contents together with the system.
- **Black box testing** has used once after a particular functionality is completely developed. A set of random values used for the testing and the errors and bugs which are generated is identifies and fixed. The gessoes over user error can be tested at this stage.

In order to uncover the errors, present in different phases we have the concept of levels of testing. The basic levels of testing are:



**Figure 8.1-testing technique**

A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

#### Acceptance testing

Acceptance testing is often done by the customers to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

#### System testing

System testing is the testing to ensure that by putting the software in different environments (eg. Operating system) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

#### Integration testing

Integration testing is testing in which a group of components are combined to produce output, also the interaction between software and hardware is tested in integration testing. If software and hardware components have any relation. It may fall under both white box and black box testing.

#### Unit testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that unit programmer has implemented is producing expected output against given input.

Operating Systems	Windows 10 Pro
Hardware	Intel Core i5 2.3GHz 4GB RAM
Software	Windows 10 Operating System (64 bit)
Debugger and testing tools	visual Studio Debugger

**Table: 8.1 – Minimum testing platform requirements**

<b>Add medicine field</b>			
Test case descriptor	Input data	Expected output	status
Category	Medicine category	Display the successfully added message	pass
Medicine Id	Unique Id		
Medicine name	Medicine name		
Dosage	Dosage of medicine		
Manufacturer	Manufacturer name		
Imported by	Company name		
Brand name	Brand name		
Supplier Id	Unique Id		
Generic name	Unique name		
Quantity	Amount of stock		
Entry date	Date		
Manufacturer date	Date		
Expiry date	Date		
Buying price	Price rupees		
Selling price	Price rupees		
Validate medicine details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.2 – add medicine case discussion**

<b>View medicine field</b>			
Test case descriptor	Input data	Expected output	status
Choose type	Category of medicine	Display the medicine details	pass
Medicine name	Medicine name		
Validate view medicine details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.3 – view medicine case discussion**

<b>View expiry of medicine field</b>			
Test case descriptor	Input data	Expected output	status
Search date	date	Display the medicine details	pass
Validate view medicine details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.4 – view expiry date case discussion**

<b>modify medicine field</b>			
Test case descriptor	Input data	Expected output	status
Choose type	Category of medicine	Display the medicine details	pass
Medicine name	Medicine name		
Medicine Id	Unique Id		
Category	Medicine category		
Medicine Id	Unique Id		
Medicine name	Medicine name		
Dosage	Dosage of medicine		
Manufacturer	Manufacturer name		
Imported by	Company name		
Brand name	Brand name		
Supplier Id	Unique Id		
Generic name	Unique name		
Quantity	Amount of stock		
Entry date	Date		
Manufacturer date	Date		
Expiry date	Date		
Buying price	Price rupees		
Selling price	Price rupees		
Validate view medicine details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.5 – modify medicine case discussion**

<b>Delete medicine field</b>			
Test case descriptor	Input data	Expected output	status
Choose type	Category of medicine	Display the medicine details	pass
Medicine name	Medicine name		
Medicine Id	Unique Id		
Validate view medicine details Steps: 1. Enter all the required fields 2. Unique input	Incorrect data	Display error type	pass

**Table: 8.6 – delete medicine case discussion**

<b>Add staff field</b>			
Test case descriptor	Input data	Expected output	status
Staff name	Full name	Display the successfully added message	pass
Date of birth	date		
Gender	Male/ Female		
Address	Address		
Contact no	number		
NIC No	Unique no		
Age	Age		
Salary	Salary rupees		
Joining date	date		
E-mail	Unique e-mail id		
username	name		
Password	password		
role	Job of staff		
Validate staff details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.7 – add staff case discussion**

<b>View staff field</b>			
Test case descriptor	Input data	Expected output	status
Staff name	Full name	Display the staff details	pass
NIC No	Unique Id		
Validate view staff details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.8 – view staff case discussion**

<b>Modify staff field</b>			
Test case descriptor	Input data	Expected output	status
Staff name	Full name	Display the staff details	pass
NIC No	Unique Id		
Staff name	Full name		
Date of birth	date		
Gender	Male/ Female		
Address	Address		
Contact no	number		
NIC No	Unique no		
Age	Age		
Salary	Salary rupees		
Joining date	date		
E-mail	Unique e-mail id		
username	name		
Password	password		
role	Job of staff		
Validate staff details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.9 – modify staff case discussion**

<b>Delete staff field</b>			
Test case descriptor	Input data	Expected output	status
Staff name	Full name	Display the staff details	pass
NIC No	Unique Id		
Validate view staff details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.10 – delete staff case discussion**

<b>Add supplier field</b>			
Test case descriptor	Input data	Expected output	status
Supplier name	Full name		
Supplier Id	Unique no		
Company name	Company name	Display the successfully added message	pass
License number	Unique no		
E-mail	Unique mail id		
Contact number	number		
address	address		
Validate supplier details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.11 – add supplier case discussion**

<b>View supplier field</b>			
Test case descriptor	Input data	Expected output	status
Supplier name	Full name	Display the staff details	pass
Supplier Id	Unique Id		
Validate view supplier details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.12 – view supplier case discussion**

<b>Modify supplier field</b>			
Test case descriptor	Input data	Expected output	status
Supplier name	Full name	Display the supplier details	pass
Supplier Id	Unique Id		
Supplier name	Full name		
Supplier Id	Unique no		
Company name	Company name		
License number	Unique no		
E-mail	Unique mail id		
Contact number	number		
address	address		
Validate supplier details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.13 – modify supplier case discussion**

<b>Delete supplier field</b>			
Test case descriptor	Input data	Expected output	status
Supplier name	Full name	Display the staff details	pass
Supplier Id	Unique Id		
Validate view supplier details Steps: 1. Enter all the required fields	Incorrect data	Display error type	pass

**Table: 8.14 – delete supplier case discussion**

<b>Retail sales field</b>			
Test case descriptor	Input data	Expected output	status
Bill no	number	Display the successfully added message	pass
Date	date		
Medicine Id	Unique Id		
Quantity	number		
Discount	Number in percentage		
Paid amount	Amount in rupees		
Validate sales details Steps: 1. Enter all the required fields 2. Unique inputs	Incorrect data	Display the error type	pass

**Table: 8.15 – retails sales case discussion**

<b>View sales field</b>			
Test case descriptor	Input data	Expected output	status
Bill no	number	Display the sales details	pass
Date	date		
Validate view sales details Steps: 1. Enter all the required fields	Incorrect data		

**Table: 8.16 – view sales case discussion**

<b>View purchase field</b>			
Test case descriptor	Input data	Expected output	status
Medicine name	name	Display the purchase details	pass
Medicine Id	Unique Id		
Supplier Id	Unique Id		
Validate view purchase details Steps: 1. Enter all the required fields	Incorrect data		

**Table: 8.17 – view purchase case discussion**

<b>Login field</b>			
Test case descriptor	Input data	Expected output	status
Input username	Username	Accept the username and password, display the main field	
Input password	password		pass
Validate login details Steps: 1. Enter all the required fields	Incorrect data	Display error message and stay login page	pass

**Table: 8.18 – login case discussion**

<b>Change password field</b>			
Test case descriptor	Input data	Expected output	status
Input username	Username		
Input old password	Old password	Accept the username and password, display updated message	
Input new password	New password		pass
Input confirm password	New password		
Validate login details Steps: 1. Enter all the required fields	Incorrect data	Display error message and stay change password page	pass

**Table: 8.19 – change password case discussion**

In this section, we have mentioned the test types, which we have followed when implementing the testing procedure and test cases of the system. We have further described test strategies used, test plan, sample test cases and presents the report of the overall test phase.

## 9 Discussions and challenges

In the starting point of our project, the members of group 3S discussed about the topic of the project. But, we could not get a stable decision on project topic and we suggested some project topics to our mentor, that was the application for CD/DVD shop management, health care centre of University of Jaffna and school management system, these topics were already done by our seniors for their group project and finally we got the permission to do a “Pharmacy Management System”, from our mentor.

Our members have already decided to do a standalone system and chosen visual C# as a programming language in the platform of visual studio. And also we planned to do an efficient complete application for our client.

Next challenge raised to find the suitable client for us. We have faced many problems during our project time. Because we needed the requirement to prepare the project proposal, through this we wanted to get approval letter from our university, for start our project work.

Already we included in previous sections, the challenges were faced by us, during the requirement gathering, system designing, and implementation works.

Another way of problem was team unity and cooperation, some of the team members not interested and do not care about the group project in the designing and implantation work. Interested people could not contact other members on needed time to meet client and mentor of our group. This type of problem made us frustrated. But, we manage these problems in better way.

Software engineering techniques used in our project and we learned these techniques in short time of period. In daily basis we have learned new important things of software engineering techniques, how to take of the project and how to guide a group.

And also some issues during the implementation, sometimes our coding is correct by some bugs occurred.

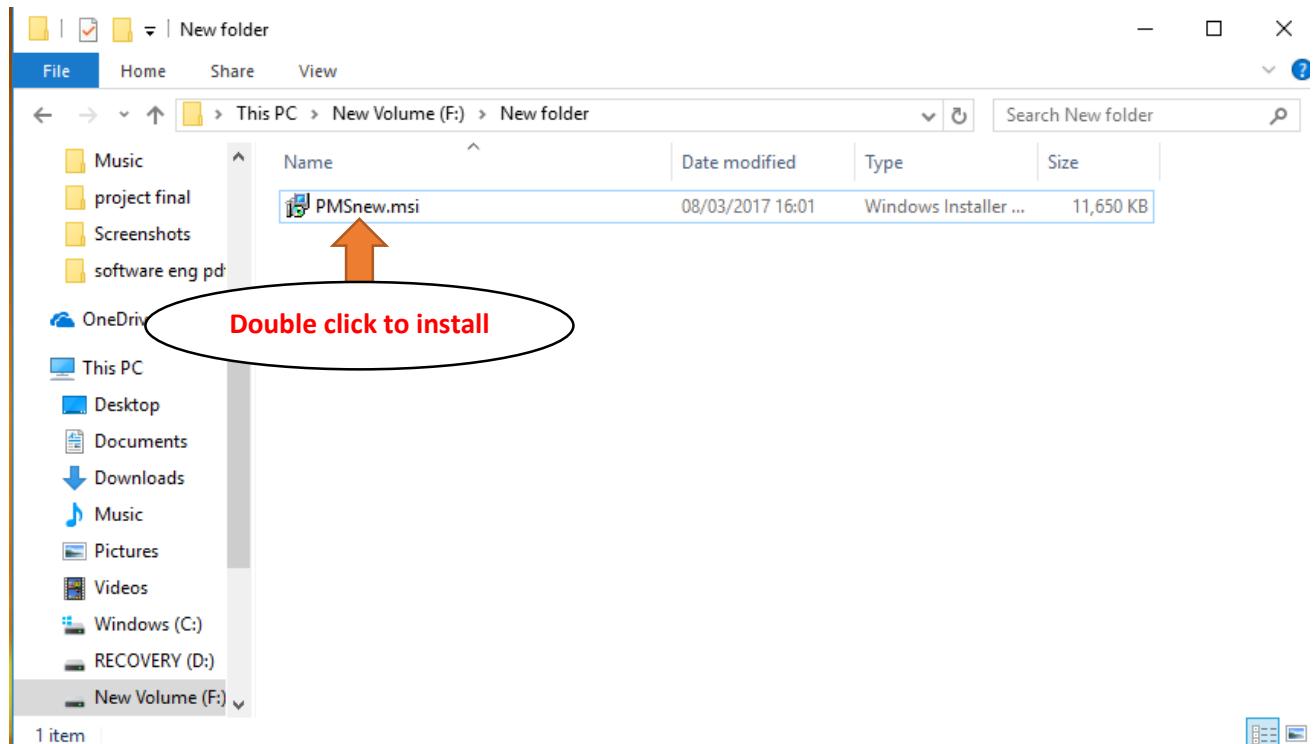
Finally, project report is preparation was the challenge to us. Because, member of group 3S are in their semester vacation at their home town, they could not meet each other, by telephone calls, we managed that problem also.

## 10 References

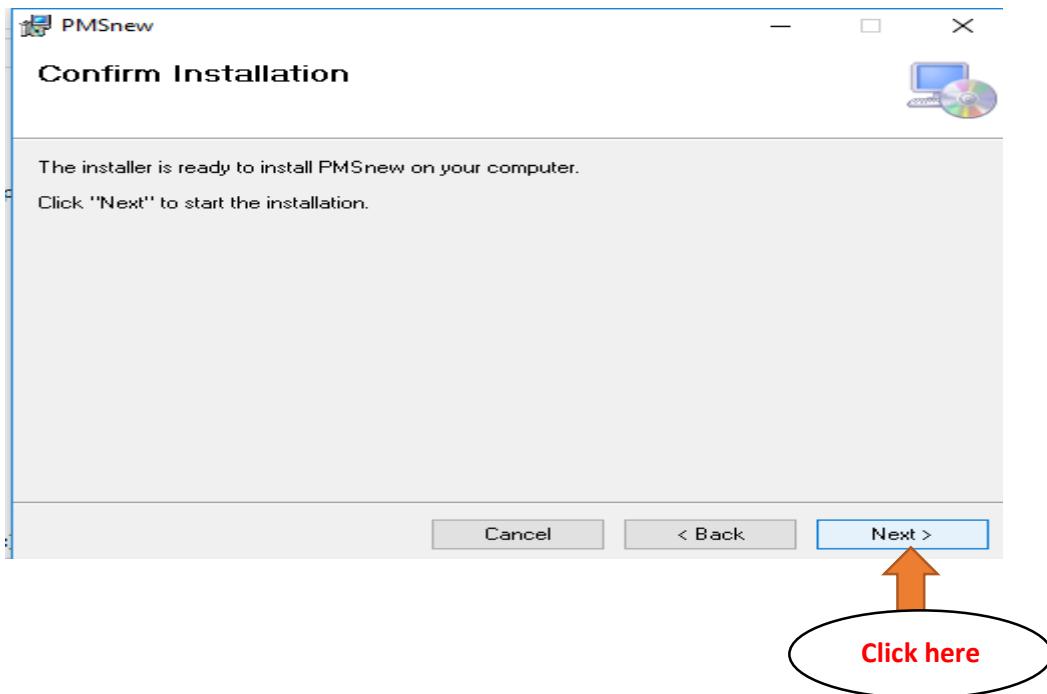
- [www.google.com](http://www.google.com)
- stack overflow.com
- [www.youtube.com](http://www.youtube.com)

## 11 Appendices

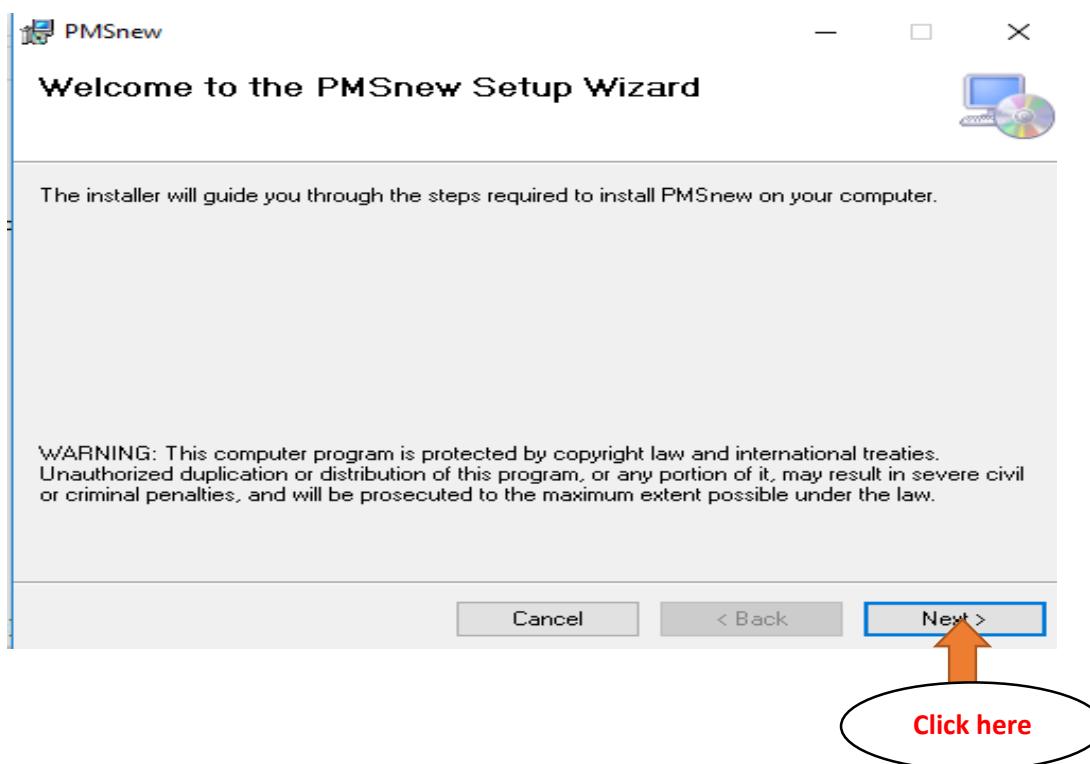
### 11.1 User documentation



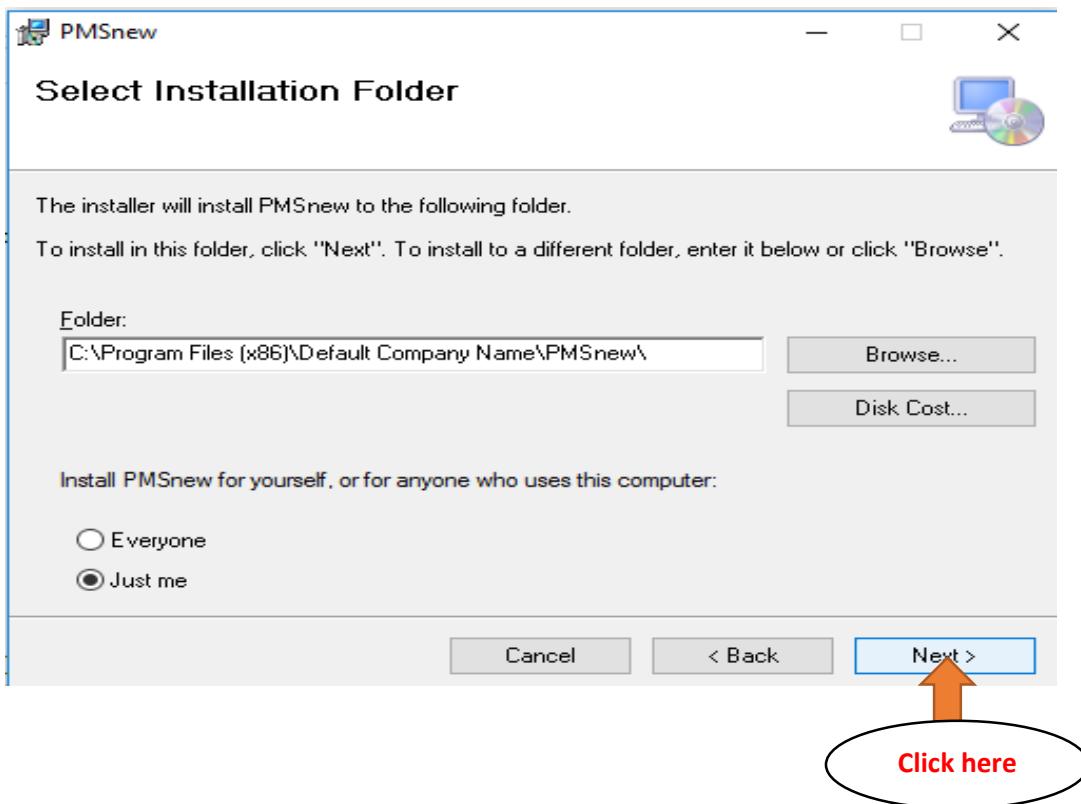
**Figure 11.1.1-how to install the system(step1)**



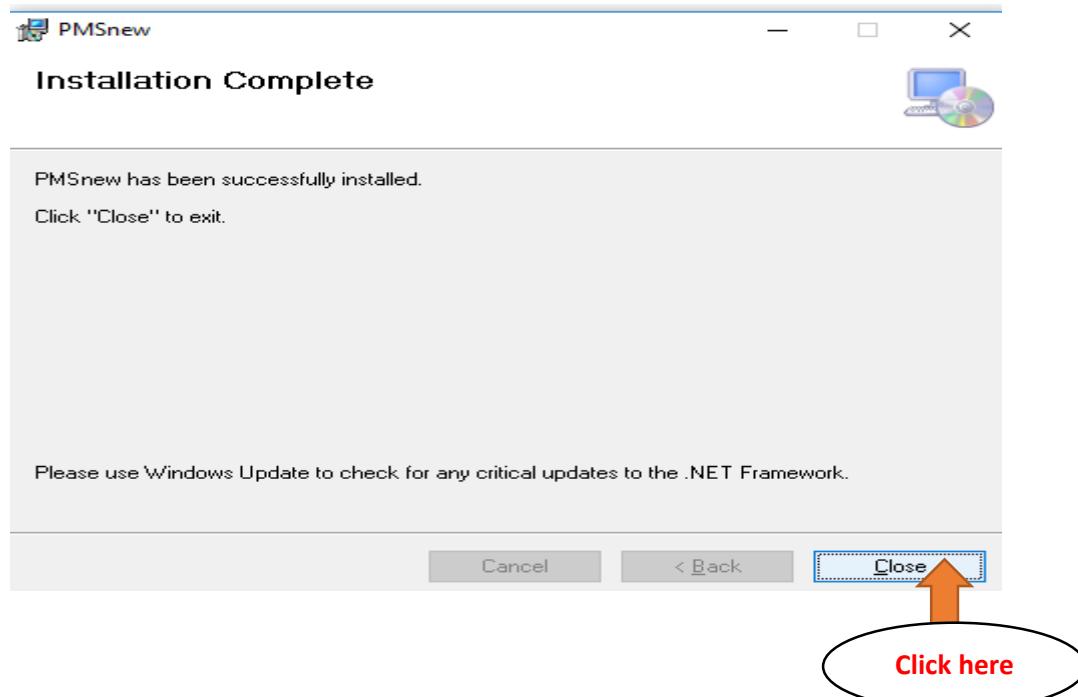
**Figure 11.1.2-how to install the system (step 2)**



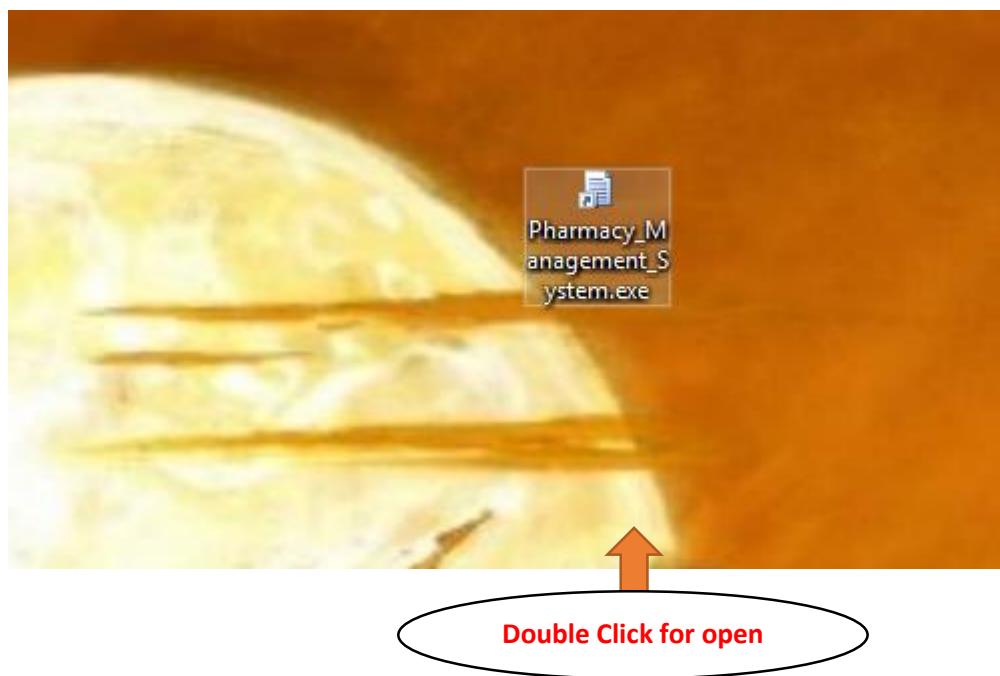
**Figure 11.1.3-how to install the system (step 3)**



**Figure 11.1.4-how to install the system (step 4) and setup the path of the system**



**Figure 11.1.5-how to install the system (step 5)**



**Figure 11.1.6-open the application.**



**Figure 11.1.7-little guide for user**

We ensure that our system is automated and user friendly to use, and the user can access without user guide. And also the functionalities of the system, simply understand anyone. That's why we did not include the detailed user guide here.

## 11.2 Development documentation

### 11.2.1 Questionnaire

The purpose of the questionnaire is to identify and specify functional requirements of our proposed “Pharmacy Management System” to be used by “ATLANTA” pharmacy.

The purpose of the questionnaire

It's basically intended to solicit for information as regards to the pharmacy management, this section briefly describes the open end and close end questionnaire to be answered by the user.

Name:					
Pharmacy:					
What type of technique is the current system	manual		computerized		
Is the system secure	Yes		No		
What is the speed of the system	slow		Medium		Fast
Are you contented with the system	Yes		No		
Multipurpose enough to handle a number of user at a go	minimum		maximum		
User friendly	To admin		To user		To all users
Password for security	Necessary		Not necessary		

**Table: 11.2.1.1-feedback of user**

The following description enables the respondent to give admin/users own opinion about the pharmacy management system in hospital.

Comment on how the system works

.....

What are some of the merits of the system?

.....

Comment on the security of the system

Recommend on the features of the system

You can give feedbacks and more comments here about the system:

## 11.2.2 Source code for the developed system.

### Coding for add medicine

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Pharmacy_Management_System
{
    public partial class Add_Medicine : Form
    {
        SqlConnection con;
        SqlCommand com;
        string sql;
        public Add_Medicine()
        {
            InitializeComponent();
            con = new SqlConnection(@"Data
Source=.\SQLEXPRESS;AttachDbFilename=C:\Pharmacy_Management_System\Pharmacy_Management_System
\Medicine.mdf;Integrated Security=True;User Instance=True");
        }

        private void Add_Medicine_Load(object sender, EventArgs e)
        {
            textBox12.Text = dateTimePicker1.Value.ToShortDateString();

            con.Open();
            SqlCommand cmd0 = new SqlCommand("SELECT mID FROM details_M", con);

            SqlDataReader reader0 = cmd0.ExecuteReader();
            AutoCompleteStringCollection MyCollection0 = new AutoCompleteStringCollection();
            while (reader0.Read())
            {
                MyCollection0.Add(reader0.GetString(0));
            }
            textBox1.AutoCompleteCustomSource = MyCollection0;
            con.Close();

            con.Open();
            SqlCommand cmd = new SqlCommand("SELECT mName FROM details_M", con);

            SqlDataReader reader = cmd.ExecuteReader();
            AutoCompleteStringCollection MyCollection = new AutoCompleteStringCollection();
            while (reader.Read())
            {
                MyCollection.Add(reader.GetString(0));
            }
            textBox2.AutoCompleteCustomSource = MyCollection;
            con.Close();

            con.Open();
            SqlCommand cmd1 = new SqlCommand("SELECT batchNo FROM details_M", con);

            SqlDataReader reader1 = cmd1.ExecuteReader();

```

```

AutoCompleteStringCollection MyCollection1 = new AutoCompleteStringCollection();
while (reader1.Read())
{
    MyCollection1.Add(reader1.GetString(0));
}
textBox3.AutoCompleteCustomSource = MyCollection1;
con.Close();

con.Open();
SqlCommand cmd2 = new SqlCommand("SELECT dosage FROM details_M", con);

SqlDataReader reader2 = cmd2.ExecuteReader();
AutoCompleteStringCollection MyCollection2 = new AutoCompleteStringCollection();
while (reader2.Read())
{
    MyCollection2.Add(reader2.GetString(0));
}
textBox4.AutoCompleteCustomSource = MyCollection2;
con.Close();

con.Open();
SqlCommand cmd3 = new SqlCommand("SELECT manufacturer FROM details_M", con);

SqlDataReader reader3 = cmd3.ExecuteReader();
AutoCompleteStringCollection MyCollection3 = new AutoCompleteStringCollection();
while (reader3.Read())
{
    MyCollection3.Add(reader3.GetString(0));
}
textBox5.AutoCompleteCustomSource = MyCollection3;
con.Close();

con.Open();
SqlCommand cmd4 = new SqlCommand("SELECT importBy FROM details_M", con);

SqlDataReader reader4 = cmd4.ExecuteReader();
AutoCompleteStringCollection MyCollection4 = new AutoCompleteStringCollection();
while (reader4.Read())
{
    MyCollection4.Add(reader4.GetString(0));
}
textBox9.AutoCompleteCustomSource = MyCollection4;
con.Close();

con.Open();
SqlCommand cmd5 = new SqlCommand("SELECT brandName FROM details_M", con);

SqlDataReader reader5 = cmd5.ExecuteReader();
AutoCompleteStringCollection MyCollection5 = new AutoCompleteStringCollection();
while (reader5.Read())
{
    MyCollection5.Add(reader5.GetString(0));
}
textBox10.AutoCompleteCustomSource = MyCollection5;
con.Close();

con.Open();
SqlCommand cmd6 = new SqlCommand("SELECT genericName FROM details_M", con);

SqlDataReader reader6 = cmd6.ExecuteReader();

```

```

        AutoCompleteStringCollection MyCollection6 = new AutoCompleteStringCollection();
        while (reader6.Read())
        {
            MyCollection6.Add(reader6.GetString(0));
        }
        textBox11.AutoCompleteCustomSource = MyCollection6;
        con.Close();
    }

    private void textBox4_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 || e.KeyChar == 109 ||
e.KeyChar == 108 || e.KeyChar == 103 || e.KeyChar == 107 ? false : true;
    }

    private void textBox6_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 ? false : true;
    }

    private void textBox7_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 || e.KeyChar == 46 ? false
: true;
    }

    private void textBox8_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 || e.KeyChar == 46 ? false
: true;
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Login_Admin la = new Login_Admin();
        la.Show();
        this.Hide();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        try
        {

            string x, ab, cd, ef, gh, ij, kl, mn, op, qr, st, uv, xy, mk, im, br, gn, cv;

            x = comboBox1.Text;
            ab = textBox1.Text;
            cd = textBox2.Text;
            ef = textBox3.Text;
            gh = textBox4.Text;
            ij = textBox5.Text;
            im = textBox9.Text;
            br = textBox10.Text;
            gn = textBox11.Text;
            kl = textBox6.Text;
            mn = textBox12.Text;
            op = datePicker2.Text;
            qr = datePicker3.Text;
            st = textBox7.Text;
            uv = textBox8.Text;
            cv = textBox13.Text;
        }
    }
}

```

```

        if ((x == "") || (ab == "") || (cd == "") || (ef == "") || (gh == "") || (ij ==
== "") || (kl == "") || (mn == "") || (op == "") || (qr == "") || (st == "") || (uv == "") ||
(im == "") || (br == "") || (gn == "") || (cv == ""))
    {
        MessageBox.Show("please add the full details", "Warning");
    }
    else
    {

        con.Open();
        sql = "insert into details_M values('"+ x + "','" + ab + "','" + cd +
"','" + ef + "','" + gh + "','" + im + "','" + br + "','" + gn + "','" + ij + "','" + kl +
"','" + mn + "','" + op + "','" + qr + "','" + st + "','" + uv + "','" + cv + "')";
        com = new SqlCommand(sql, con);
        com.ExecuteNonQuery();

        con.Close();
        MessageBox.Show("successfully added", "Add", MessageBoxButtons.OK,
MessageBoxIcon.Information);
    }
}

private void button2_Click(object sender, EventArgs e)
{
    textBox4.Text = "";
    textBox5.Text = "";
    textBox2.Text = "";
    textBox1.Text = "";
    textBox3.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    textBox10.Text = "";
    textBox11.Text = "";
    dateTimePicker1.Text = "";
    dateTimePicker2.Text = "";
    dateTimePicker3.Text = "";
    textBox13.Text = "";
}
}
catch (Exception ex)
{
    MessageBox.Show("something went wrong, try again..!!!");
}
}

private void button2_Click(object sender, EventArgs e)
{
    textBox4.Text = "";
    textBox5.Text = "";
    textBox2.Text = "";
    textBox1.Text = "";
    textBox3.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    textBox10.Text = "";
    textBox11.Text = "";
    dateTimePicker1.Text = "";
    dateTimePicker2.Text = "";
}

```

```

        dateTimePicker3.Text = "";
        textBox13.Text = "";
    }

    private void textBox1_TextChanged(object sender, EventArgs e)
    {

    }

    private void textBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox2.Focus();
        }
    }

    private void textBox2_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox3.Focus();
        }
    }

    private void textBox3_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox4.Focus();
        }
    }

    private void textBox4_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox5.Focus();
        }
    }

    private void textBox5_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox9.Focus();
        }
    }

    private void textBox9_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox10.Focus();
        }
    }

    private void textBox10_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox13.Focus();
        }
    }
}

```

```

        }

    }

    private void textBox11_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox6.Focus();
        }
    }

    private void textBox6_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            dateTimePicker2.Focus();
        }
    }

    private void dateTimePicker1_KeyDown(object sender, KeyEventArgs e)
    {
    }

    private void dateTimePicker2_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            dateTimePicker3.Focus();
        }
    }

    private void dateTimePicker3_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox7.Focus();
        }
    }

    private void textBox7_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            textBox8.Focus();
        }
    }

    private void textBox8_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Enter)
        {
            button1.PerformClick();
        }
    }

    private void dateTimePicker2_ValueChanged(object sender, EventArgs e)
    {
        if (dateTimePicker2.Value > DateTime.Now)
        {
            dateTimePicker2.Value = DateTime.Now;
        }
    }
}

```

```
        MessageBox.Show("select the correct date ..!!", "warning");

    }

}

private void dateTimePicker3_ValueChanged(object sender, EventArgs e)
{
    if (dateTimePicker3.Value < DateTime.Now)
    {
        dateTimePicker3.Value = DateTime.Now;
        MessageBox.Show("select the correct date ..!!", "warning");
    }
}

private void textBox13_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox11.Focus();
    }
}

}
```

## Coding for delete medicine

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Pharmacy_Management_System
{
    public partial class Delete_Medicine_Details : Form
    {
        SqlConnection con;
        SqlCommand com;
        string sql;
        public Delete_Medicine_Details()
        {
            InitializeComponent();
            con = new SqlConnection(@"Data
Source=.\SQLEXPRESS;AttachDbFilename=F:\Pharmacy_Management_System\Pharmacy_Management_System
\Medicine.mdf;Integrated Security=True;User Instance=True");
        }

        private void label1_Click(object sender, EventArgs e)
        {
```

```

        }

e)    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs
{
}

private void Delete_Medicine_Details_Load(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("SELECT mName FROM details_M", con);

    SqlDataReader reader = cmd.ExecuteReader();
    AutoCompleteStringCollection MyCollection = new AutoCompleteStringCollection();
    while (reader.Read())
    {
        MyCollection.Add(reader.GetString(0));
    }
    textBox1.AutoCompleteCustomSource = MyCollection;
    con.Close();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("Select mName from details_M where category = '" +
+ comboBox1.Text + "'", con);

    SqlDataReader reader = cmd.ExecuteReader();
    AutoCompleteStringCollection MyCollection = new AutoCompleteStringCollection();
    while (reader.Read())
    {
        MyCollection.Add(reader.GetString(0));
    }
    textBox1.AutoCompleteCustomSource = MyCollection;
    con.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        if ((textBox1.Text == "") || (comboBox1.Text == ""))
        {
            MessageBox.Show("sorry,please add the full details", "Warning");
        }
        con.Open();

        //com = new SqlCommand("select * from details where CONVERT
        (varchar,expiryDate) ='" + dateTimePicker1.Text + "'", con);
        sql = "select * from details_M where CONVERT(varchar,mID) = '" +
        comboBox2.Text + "'";

        SqlDataAdapter da = new SqlDataAdapter(sql, con);

        DataTable di = new DataTable();
        da.Fill(di);
        dataGridView1.DataSource = di;
        con.Close();
    }
}

```

```

        catch (Exception ex1)
        {
            MessageBox.Show("sorry there are something went wrong..!!!");
        }

    }

    private void button2_Click(object sender, EventArgs e)
    {
        textBox1.Text = "";
        comboBox1.Text = "";
    }

    private void button3_Click(object sender, EventArgs e)
    {
        try
        {
            // con.Open();
            if ((textBox1.Text == "") || (comboBox1.Text == ""))
            {
                MessageBox.Show("sorry,please add the full details", "Warning");
            }
            else
            {

                con.Open();

                DialogResult result = MessageBox.Show("Do you want to Remove the Member ?",
                "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

                if (result == DialogResult.Yes)
                {
                    sql = ("delete from details_M where CONVERT(VARCHAR,mID) ='" +
                comboBox2.Text + "'");

                    com = new SqlCommand(sql, con);

                    com.ExecuteNonQuery();

                    MessageBox.Show("successfully deleted", "Delete", MessageBoxButtons.OK,
                    MessageBoxIcon.Information);

                    textBox1.Text = "";
                    comboBox2.Text = "";
                }
                else
                {
                    MessageBox.Show("not deleted", "Warning", MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
                    textBox1.Text = "";
                    comboBox2.Text = "";

                }
                con.Close();
            }
            //con.Close();
        }

        catch (Exception ex1)
        {
            MessageBox.Show("sorry there are something went wrong..!!!");
        }

    }
}

```

```
private void button4_Click(object sender, EventArgs e)
{
    Login_Admin la = new Login_Admin();
    la.Show();
    this.Hide();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    try
    {
        con.Open();
        com = new SqlCommand("select * from details_M where mName = '" +
textBox1.Text + "'", con);
        SqlDataReader dr;
        dr = com.ExecuteReader();
        while (dr.Read())
        {
            comboBox2.Items.Add(dr["mID"].ToString());
        }
        dr.Close();
        con.Close();
    }
    catch (Exception ex1)
    {
        MessageBox.Show("sorry there are something went wrong, please try
again...!!!");
    }
}
```

## Coding for view expiry

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Pharmacy_Management_System
{
    public partial class View_Expiry_Date : Form
    {
        SqlConnection con;
        SqlCommand com;
        string sql;
        public View_Expiry_Date()
        {
            InitializeComponent();
            con = new SqlConnection(@"Data
Source=.\\SQLEXPRESS;AttachDbFilename=F:\\Pharmacy_Management_System\\Pharmacy_Management_System
\\Medicine.mdf;Integrated Security=True;User Instance=True");
        }
    }
}
```

```

e)
{
}

private void button4_Click(object sender, EventArgs e)
{
    Login_Admin la = new Login_Admin();
    la.Show();
    this.Hide();
}

private void button6_Click(object sender, EventArgs e)
{
    try
    {
        if ((dateTimePicker1.Text == ""))
        {
            MessageBox.Show("sorry,please add the full details", "Warning");
        }
        con.Open();

        //com = new SqlCommand("select * from details where CONVERT
        (varchar,expiryDate) ='" + dateTimePicker1.Text + "'", con);
        sql = "select * from details_M where expiryDate ='" + dateTimePicker1.Text +
        "'";
    }

    SqlDataAdapter da = new SqlDataAdapter(sql, con);

    DataTable di = new DataTable();
    da.Fill(di);
    dataGridView1.DataSource = di;
    con.Close();
}
catch (Exception ex1)
{
    MessageBox.Show("sorry there are something went wrong, please try
again..!!!");
}
}

private void button5_Click(object sender, EventArgs e)
{
    try
    {
        if ((dateTimePicker1.Text == ""))
        {
            MessageBox.Show("sorry,please add the full details", "Warning");
        }
        else
        {

            con.Open();

            DialogResult result = MessageBox.Show("Do you want to Remove the Member
?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                SqlCommand cmd = new SqlCommand("UPDATE details_M SET quantity = 0
where expiryDate ='" + dateTimePicker1.Text + "'", con);
}

```

```
cmd.ExecuteNonQuery();  
  
SqlDataAdapter da = new SqlDataAdapter(sql, con);  
  
DataTable di = new DataTable();  
da.Fill(di);  
dataGridView1.DataSource = di;  
  
MessageBox.Show("successfully deleted", "Delete",  
MessageBoxButtons.OK, MessageBoxIcon.Information);  
  
dateTimePicker1.Text = "";  
}  
else  
{  
    MessageBox.Show("not deleted", "Warning", MessageBoxButtons.OK,  
MessageBoxIcon.Information);  
    dateTimePicker1.Text = "";  
  
}  
con.Close();  
}  
}  
}  
catch (Exception ex1)  
{  
    MessageBox.Show("sorry there are something went wrong, please try  
again..!!");  
}  
}  
}  
}
```

## Coding for retail sales and billing

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using Pharmacy_Management_System.Properties;

namespace Pharmacy_Management_System
{
    public partial class Retail_Sales_Details : Form
    {
        SqlConnection con;
        SqlCommand com,com1;
        String sql,sql1;
        public Retail_Sales_Details()
        {
            InitializeComponent();
            con = new SqlConnection(@"Data
Source=.\SQL EXPRESS;AttachDbFilename=C:\Users\Jathurshan
```

```

Sumaniran\Desktop\Pharmacy_Management_System\Pharmacy_Management_System\Medicine.mdf;Integrated Security=True;User Instance=True");
}

private List<Retail> shoppingCart = new List<Retail>();

private void label6_Click(object sender, EventArgs e)
{

}

private void label15_Click(object sender, EventArgs e)
{
}

private void label5_Click(object sender, EventArgs e)
{
}

private void textBox12_TextChanged(object sender, EventArgs e)
{
    if ((textBox12.Text == ""))
    {
        MessageBox.Show("Sorry,please add the full details", "Warning");
    }

    con.Open();

    com = new SqlCommand("select mName,expiryDate,dosage,sellingPrice from details_M
where CONVERT (varchar,mID) ='" + textBox12.Text + "'", con);
    //sql = "select * from details where CONVERT (varchar,mID) ='" + textBox12.Text +
""";
    SqlDataReader dr;
    dr = com.ExecuteReader();
    while (dr.Read())
    {
        textBox2.Text = dr[0].ToString();
        textBox14.Text = dr[1].ToString();
        textBox4.Text = dr[2].ToString();
        textBox5.Text = dr[3].ToString();

    }
    dr.Close();
    con.Close();

/* SqlDataAdapter da = new SqlDataAdapter(sql, con);

DataTable di = new DataTable();
da.Fill(di);
dataGridView1.DataSource = di;

*/
}

private void textBox3_TextChanged(object sender, EventArgs e)
{
    double anDouble;
    anDouble = Convert.ToInt32(textBox5.Text);
}

```

```

anDouble = Double.Parse(textBox5.Text);
if (!string.IsNullOrEmpty(textBox3.Text) && !string.IsNullOrEmpty(textBox5.Text))
    textBox6.Text = (Double.Parse(textBox3.Text) * anDouble).ToString();
}

private void Retail_Sales_Details_Load(object sender, EventArgs e)
{
    textBox13.Text = dateTimePicker1.Value.ToShortDateString();

    con.Open();

    com = new SqlCommand("select max(convert(int,billNo)) from retails", con);

    SqlDataReader dr2;
    dr2 = com.ExecuteReader();
    String x1 = "";
    while (dr2.Read())
    {
        x1 = dr2[0].ToString();

    }
    dr2.Close();

    double anDouble;
    anDouble = Convert.ToInt32(x1);
    anDouble = Double.Parse(x1);

    anDouble = anDouble + 1;
    textBox1.Text = anDouble.ToString();
    con.Close();

    con.Open();
    SqlCommand cmd = new SqlCommand("SELECT billNo FROM retails", con);

    SqlDataReader reader = cmd.ExecuteReader();
    AutoCompleteStringCollection MyCollection = new AutoCompleteStringCollection();
    while (reader.Read())
    {
        MyCollection.Add(reader.GetString(0));
    }
    textBox1.AutoCompleteCustomSource = MyCollection;
    con.Close();

    con.Open();
    SqlCommand cmd0 = new SqlCommand("SELECT mID FROM details_M", con);

    SqlDataReader reader0 = cmd0.ExecuteReader();
    AutoCompleteStringCollection MyCollection0 = new AutoCompleteStringCollection();
    while (reader0.Read())
    {
        MyCollection0.Add(reader0.GetString(0));
    }
    textBox12.AutoCompleteCustomSource = MyCollection0;
    con.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
}

```

```

        con.Open();

        com = new SqlCommand("select SUM(amount) from retails where billNo ='" +
textBox1.Text + "'", con);
        //sql = "select * from details where CONVERT (varchar,mID) ='" +
textBox12.Text + "'";
        SqlDataReader dr;
        dr = com.ExecuteReader();
        while (dr.Read())
        {
            textBox7.Text = dr[0].ToString();
        }
        dr.Close();

        con.Close();

    }

    catch (Exception ex2)
    {
        MessageBox.Show("this Bill No is already available");
    }

}

private void textBox8_TextChanged(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(textBox7.Text) && !string.IsNullOrEmpty(textBox8.Text))
        textBox9.Text = ((Double.Parse(textBox7.Text)) - ((Double.Parse(textBox7.Text) *
Double.Parse(textBox8.Text)/100))).ToString();
}

private void textBox10_TextChanged(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(textBox9.Text) &&
!string.IsNullOrEmpty(textBox10.Text))
        textBox11.Text = (Double.Parse(textBox10.Text) -
Double.Parse(textBox9.Text)).ToString();
}

private void button2_Click(object sender, EventArgs e)
{
    try
    {
        string ab, cd, ef, gh, ij, kl, mn, mk;

        ij = textBox1.Text;
        kl = textBox13.Text;
        ab = textBox7.Text;
        cd = textBox8.Text;
        ef = textBox9.Text;
        mn = textBox10.Text;
        gh = textBox11.Text;

        if ((ij == "") || (kl == "") || (ab == "") || (cd == "") || (ef == "") || (ef
== "") || (gh == ""))
        {
            MessageBox.Show("please add the full details", "Warning");
        }
    }
}

```

```

        else
        {
            con.Open();
            sql = "insert into purchase values('" + ij + "','" + kl + "','" + ab +
"','" + cd + "','" + ef + "','" + mn + "','" + gh + "')";
            com = new SqlCommand(sql, con);
            com.ExecuteNonQuery();

            con.Close();
            MessageBox.Show("Bill is Finalised", "Add", MessageBoxButtons.OK,
MessageBoxIcon.Information);

            //dataGridView1.Rows.Clear();
        }
    }

    catch (Exception ex1)
    {
        MessageBox.Show("this Bill No is already available");
    }
}

private void textBox7_TextChanged(object sender, EventArgs e)
{
}

private void button4_Click(object sender, EventArgs e)
{
    Login_Admin la = new Login_Admin();
    la.Show();
    this.Hide();
}

private void button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void button6_Click(object sender, EventArgs e)
{
    string x, ab, cd, ef, gh, ij, kl, mn, op, qr, st, uv, xy, mk, im, br, gn;

    ab = textBox1.Text;
    cd = textBox13.Text;
    ef = textBox12.Text;
    gh = textBox2.Text;
    ij = textBox3.Text;
    kl = textBox14.Text;
    mn = textBox4.Text;
    op = textBox5.Text;
    qr = textBox6.Text;

    if (textBox1.Text != "" && ((ab != "") || (cd != "") || (ef != "") || (gh != ""))
|| (ij != "") || (kl != "") || (mn != "") || (op != "") || (qr != ""))
    {
        if (IsValidated())
        {
            Retail item = new Retail()
            {

```

```
mName = textBox2.Text,
quantity = textBox3.Text.Trim(),
dosage = textBox4.Text.Trim(),
mID = textBox12.Text.Trim(),
expiryDate = textBox14.Text.Trim(),
unitPrice = textBox5.Text.Trim(),
amount = textBox6.Text.Trim(),
};

shoppingCart.Add(item);
dataGridView1.DataSource = null;

dataGridView1.DataSource = shoppingCart;

}
}

try
{
    // string x, ab, cd, ef, gh, ij, kl, mn, op, qr, st, uv, xy, mk, im, br, gn;

ab = textBox1.Text;
cd = textBox13.Text;
ef = textBox12.Text;
gh = textBox2.Text;
ij = textBox3.Text;
kl = textBox14.Text;
mn = textBox4.Text;
op = textBox5.Text;
qr = textBox6.Text;

if ((ab == "") || (cd == "") || (ef == "") || (gh == "") || (ij == "") || (kl == ""))
|| (mn == "") || (op == "") || (qr == ""))
{
    MessageBox.Show("please add the full details", "Warning");
}
else
{
    con.Open();
    com = new SqlCommand("select SUM(quantity) from details_M where mID = '" +
textBox12.Text + "'", con);

    SqlDataReader dr2;
    dr2 = com.ExecuteReader();
    String x1 = "";
    while (dr2.Read())
    {
        x1 = dr2[0].ToString();

    }
    dr2.Close();

    double anDouble;
    anDouble = Convert.ToInt32(x1);
    anDouble = Double.Parse(x1);

    double anDouble1;
    anDouble1 = Convert.ToInt32(ij);
    anDouble1 = Double.Parse(ij);
    if (anDouble <= anDouble1)
    {
```

```

        MessageBox.Show("Your available quantity is low ..... ");
    }

    else
    {
        sql = "insert into retails values('"+ ab + "','" + cd + "','" + ef +
        "','" + gh + "','" + ij + "','" + kl + "','" + mn + "','" + op + "','" + qr + "')";
        com1 = new SqlCommand(sql, con);
        com1.ExecuteNonQuery();

        //MessageBox.Show("successfully added", "Add", MessageBoxButtons.OK,
        MessageBoxIcon.Information);

        com = new SqlCommand("select SUM(quantity) from details_M where mID = '" +
        + textBox12.Text + "'", con);

        SqlDataReader dr1;
        dr1 = com.ExecuteReader();
        String x5 = "";
        String Quant = "";
        while (dr1.Read())
        {
            x5 = dr1[0].ToString();

        }
        dr1.Close();

        double anDouble5;
        anDouble5 = Convert.ToInt32(x5);
        anDouble5 = Double.Parse(x5);
        if (!string.IsNullOrEmpty(x5) && !string.IsNullOrEmpty(textBox3.Text))
            Quant = (anDouble5 - Double.Parse(textBox3.Text)).ToString();

        SqlCommand cmd = new SqlCommand("UPDATE details_M SET quantity = '" +
        Quant + "' WHERE mID ='" + textBox12.Text + "'", con);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Details are modified");



        sql1 = "select mID,mName,quantity,expiryDate,dosage,unitPrice,amount from
retails where billNo ='"+ textBox1.Text + "'";

        SqlDataAdapter da = new SqlDataAdapter(sql1, con);

        DataTable di = new DataTable();
        da.Fill(di);
        // dataGridView1.DataSource = di;
    }
    con.Close();

```

```

        }

    }

    catch (Exception ex)
    {
        MessageBox.Show("this Bill No is already available");
    }

}

private bool IsValidated()
{
    return true;
}

private void button7_Click(object sender, EventArgs e)
{

}

private void printPreviewDialog1_Load(object sender, EventArgs e)
{

}

private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    Image image = Resources.atlanta;
    e.Graphics.DrawImage(image, 125, 20, image.Width, image.Height);

    e.Graphics.DrawString("Date: " + DateTime.Now, new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(25, 160));

    e.Graphics.DrawString("Invoice No: " + textBox1.Text.Trim(), new Font("Arial",
12, FontStyle.Regular), Brushes.Black, new Point(25, 190));
    e.Graphics.DrawString("-----", new
Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(25, 235));

    e.Graphics.DrawString("Medicine Name ", new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(30, 255));
    e.Graphics.DrawString("Quantity ", new Font("Arial", 12, FontStyle.Regular),
Brushes.Black, new Point(240, 255));
    e.Graphics.DrawString("Dosage ", new Font("Arial", 12, FontStyle.Regular),
Brushes.Black, new Point(410, 255));
    e.Graphics.DrawString("Unit Price ", new Font("Arial", 12, FontStyle.Regular),
Brushes.Black, new Point(560, 255));
    e.Graphics.DrawString("Total Price ", new Font("Arial", 12, FontStyle.Regular),
Brushes.Black, new Point(710, 255));
    e.Graphics.DrawString("-----", new
Font("Times and Newroman", 12, FontStyle.Regular), Brushes.Black, new Point(25, 270));

    int y = 295;

    foreach (var i in shoppingCart)

```

```

    {

        e.Graphics.DrawString(i.mName, new Font("Arial", 12, FontStyle.Regular),
Brushes.Black, new Point(30, y));
        e.Graphics.DrawString(i.quantity.ToString(), new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(240, y));
        e.Graphics.DrawString(i.dosage.ToString(), new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(430, y));
        e.Graphics.DrawString(i.unitPrice.ToString(), new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(560, y));
        e.Graphics.DrawString(i.mID.ToString(), new Font("Arial", 12,
FontStyle.Regular), Brushes.Black, new Point(710, y));

        y += 30;
    }
    e.Graphics.DrawString("-----", new
Font("Times New Roman", 12, FontStyle.Regular), Brushes.Black, new Point(25, y));

    e.Graphics.DrawString("Total Amount : " + textBox7.Text.Trim(), new
Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(550, y + 30));
    e.Graphics.DrawString("Discount(%) : " + textBox8.Text.Trim(), new
Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(550, y + 60));
    e.Graphics.DrawString("Total Pay : " + textBox9.Text.Trim(), new
Font("Arial", 12, FontStyle.Regular), Brushes.Black, new Point(550, y + 90));
}

public string quantity { get; set; }

public string mName { get; set; }

private void button7_Click_1(object sender, EventArgs e)
{
    printPreviewDialog1.Document = printDocument1;
    printPreviewDialog1.ShowDialog();

    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";
    textBox7.Text = "";
    textBox8.Text = "";
    textBox9.Text = "";
    textBox10.Text = "";
    textBox11.Text = "";
    textBox12.Text = "";
    textBox14.Text = "";

    textBox13.Text = dateTimePicker1.Value.ToShortDateString();
    dataGridView1.DataSource = null;

    con.Open();

    com = new SqlCommand("select max(convert(int,billNo)) from retails", con);
}

```

```

SqlDataReader dr2;
dr2 = com.ExecuteReader();
String x1 = "";
while (dr2.Read())
{
    x1 = dr2[0].ToString();

}
dr2.Close();

double anDouble;
anDouble = Convert.ToInt32(x1);
anDouble = Double.Parse(x1);

anDouble = anDouble + 1;
textBox1.Text = anDouble.ToString();
con.Close();
}

private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 ? false : true;
}

private void textBox8_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 ? false : true;
}

private void textBox10_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = char.IsNumber(e.KeyChar) || e.KeyChar == 8 ? false : true;
}

private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{

}

private void textBox12_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox3.Focus();
    }
}

private void textBox3_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button6.PerformClick();
    }
}

private void textBox8_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button1.PerformClick();
    }
}

```

```

private void textBox10_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        textBox11.Focus();
    }
}

private void textBox11_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
    {
        button2.PerformClick();
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void dataGridView1_MouseDown(object sender, MouseEventArgs e)
{
    try
    {
        if (e.Button == System.Windows.Forms.MouseButtons.Right)
        {
            var hit = dataGridView1.HitTest(e.X, e.Y);
            dataGridView1.Rows[hit.RowIndex].Selected = true;
            contextMenuStrip1.Show(dataGridView1, e.X, e.Y);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("There are some problem..!!!");
    }
}

private void deletItemToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        if ((textBox1.Text == ""))
        {
            MessageBox.Show("Sorry,please add the full details", "Warning");
        }
        else
        {

            con.Open();

            DialogResult result = MessageBox.Show("Do you want to Remove the
Member ?", "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)
            {
                int index = dataGridView1.CurrentCell.RowIndex;

                shoppingCart.RemoveAt(index);
            }
        }
    }
}

```

```

        dataGridView1.DataSource = null;
        dataGridView1.DataSource = shoppingCart;

        sql = ("delete from retails where billNo ='" + textBox1.Text +
        "'");

        com = new SqlCommand(sql, con);

        com.ExecuteNonQuery();

        MessageBox.Show("Successfully deleted", "Delete",
        MessageBoxButtons.OK, MessageBoxIcon.Information);

        // textBox1.Text = "";

        sql1 = "select
mID,mName,quantity,expiryDate,dosage,unitPrice,amount from retails where billNo ='" +
textBox1.Text + "'";

        SqlDataAdapter da = new SqlDataAdapter(sql1, con);

        DataTable di = new DataTable();
        da.Fill(di);
        //dataGridView1.DataSource = di;

        com = new SqlCommand("select SUM(quantity) from details_M where
mID = '" + textBox12.Text + "'", con);

        SqlDataReader dr1;
        dr1 = com.ExecuteReader();
        String x6 = "";
        String Quant1 = "";
        while (dr1.Read())
        {
            x6 = dr1[0].ToString();

        }
        dr1.Close();

        double anDouble6;
        anDouble6 = Convert.ToInt32(x6);
        anDouble6 = Double.Parse(x6);
        if (!string.IsNullOrEmpty(x6) &&
!string.IsNullOrEmpty(textBox3.Text))
            Quant1 = (anDouble6 +
Double.Parse(textBox3.Text)).ToString();

        SqlCommand cmd = new SqlCommand("UPDATE details_M SET quantity =
'" + Quant1 + "' WHERE mID ='" + textBox12.Text + "'", con);
        cmd.ExecuteNonQuery();
        MessageBox.Show("Details are modified");
    }
    else
    {
        MessageBox.Show("not deleted", "Warning", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
        // textBox1.Text = "";
    }
}

```

```
        con.Close();
    }

    catch (Exception ex)
    {
        MessageBox.Show("There are some problem..!!!");
    }
}

private void textBox9_TextChanged(object sender, EventArgs e)
{
}

private void button6_KeyDown(object sender, KeyEventArgs e)
{
    if(e.KeyCode==Keys.Enter)
    {
        textBox12.Focus();
    }
}

private void textBox6_TextChanged(object sender, EventArgs e)
{
}

}
}
```

For other information contact via e-mail

S.Jathurshan :- [s.jathur93@gmail.com](mailto:s.jathur93@gmail.com)

M.Kausik:- [kausik219@gmail.com](mailto:kausik219@gmail.com)

And our blogger Id:- <https://pharmacy3s.blogspot.com>