Software Requirements Specification

Version 2.0

Pharmaceutical Management System

Team 10

Member Name	Member Roll #	Primary Responsibility
Ghayyur Ali	12L-4276	Documentation
Usama Saqib	15L-4064	Development
Usman Majied	15L-4033	Development
Muhammad Faiq	15L-4266	Documentation

Table of Contents

1. Introduction 1
1.1 Product 1
1.2 Scope 1
1.3 Business Goals 1
1.4 Document Conventions 1
1.5 References 1
2 Overall Description 2
2.1 Product Features 2
2.2 User Classes and Characteristics 3
2.3 Operating Environment 4
2.4 Design and Implementation Constraints 4
2.5 Assumptions and Dependencies 4
3. Functional Requirements 5
3.1 5
3.2 6
3.3 7
3.4 8
3.5 10
3.6 11
3.7 12
3.8 13
3.9 14
3.10 15
3.11 16
3.12 17
4. Nonfunctional Requirements 18
4.1 Performance Requirements 18
4.2 Security Requirements 18
Appendix A: Glossary 19
Appendix B: Analysis Models 20
1. USE Case Diagram: 20
2. Class Diagram: 21
3. ER Diagram: 22
4. Data Flow Diagram: 23
4.1 Context Level: 23
4.2 Final Level: 24
Appendix C: Design 25
1. Component Diagram 25

1. Introduction

1.1 Product

Our project is of a large-scale pharmaceutical company which is looking to computerize its operations management. The company has several branches and is looking to move on from the paper system. Our **Java Based Desktop Application** will allow the company to handle all of its pharmacy branches remotely as well as keep track of the inventory of each branch.

1.2 Scope

The Software provide users to handle all of their pharmacy branches remotely as well as keep track of the inventory of each branch. System will provide means for users to sign up and there will be a system generated user too. System will provide users to create branches, add/remove salesman in a branch, and display branch statistics of every branch. System will allow user to request for more stock as well as search inventory and sell medicine. System will provide each user certain functionalities as per their need.

1.3 Business Goals

Pharmaceutical chains are spread everywhere. Our goal is to computerize the slow paperwork pharmacies have to deal with and provide a very easy yet feature filled user interface to pharmaceutical companies. A Single GM will be able to manage all branches, medicines requests and BM. Other users of our software will be able to benefit from it by all means possible. Software is not just limited to one company, rather it'll have features that are in demand from many pharmaceutical companies.

1.4 Document Conventions

Bold characters exhibit significant information.

1.5 References

Pharmaceuticals, C. (2018). *CCL Pharmaceuticals - Healthy, Happy Life*. Retrieved from CCL Pharmaceuticals: http://www.cclpharma.com/

Pharmacy, S. (2018). *Selmore Pharmacy*. Retrieved from Selmorepharma.com: https://www.selmorepharma.com/

Tech, A. (2014). *Non Functional Requirements*. Retrieved from Aakash Tech Support Docs: https://aakashtechsupportdocs.readthedocs.io/en/latest/nonfunc.html

2 Overall Description

2.1 Product Features

PMS is an all in one management system for pharmaceuticals that houses a large number feature set that's not limited to a single user but many users including General Manager, Branch Manager and Salesman. Following are the product features.

2.1.1 Signup

System will provide general managers the means to create an account for branch managers. Branch managers will have the functionality to add salesman.

2.1.2 Login

System will provide users to log in to their accounts at any time.

2.1.3 General Manager Feature Set

i. System will provide the means for general manager to add branches.
ii. System will provide the means for General Manager to create an account for the branch managers
iii. System will provide the means for General Manager to assign the Branch Manager to each branch.
iv. System will provide the means for General Manager to remove and modify branch manager and branch data.
v. System will provide the means for General Manager to view requests for stock.

2.1.4 Branch Manager Feature Set

i. System will provide the means for Branch Manager to add salesman.
ii. System will provide the means for Branch Manager to modify details of a salesman.
iii. System will provide the means for Branch Manager to remove a salesman.
iv. System will provide the means for Branch Manager to request for stock and forward it to GM.

2.1.5 Salesman Feature Set

i. System will provide the means for Salesman to view and search the inventory.
ii. System will provide the means for Salesman to sell medicine and generate bill.

2.2 User Classes and Characteristics

System defines three user classes including General Manager, Branch Manager and Salesman. Following are the pertinent characteristics of each user class.

2.2.1 General Manager

General Manager handles the significant part of the project. GM is responsible for adding branches and assigning branch manager to each branch. Furthermore, GM can modify the branches and their branch managers. GM also entertains the stock related requests that are generated by the branch managers.

2.2.2 Branch Manager

Branch manager handles the less favored work compared to GM. BM is responsible for managing salesmen of their particular branch. BM also monitors the stock of each medicine and sends the request for stock to GM.

2.2.3 Salesman

Salesman is the least favored user class among GM and BM classes. Salesman can view inventory and search it. Salesman can sell medicine and generate bill.

2.3 Operating Environment

PMS does not require a high-end hardware or software related environment to operate on. If a platform caters the following software and hardware then our system can operate on it easily.

2.3.1 Hardware Platform

i. Core 2 Duo 2.4 GHz Processor
ii. 2 GB RAM
iii. 500MB Hard drive space

2.3.2 Software Platform

i. Windows 7/8/10 32/64-bit OS
ii. Java 8 platform with SQL

2.4 Design and Implementation Constraints

i. In PMS, we have incorporated Java FXML for GUI development. The programmers might find it tedious to integrate it with java source code.
ii. We are using SQL server as our database which will not affect the performance and efficiency of the application.
iii. No hardware limitations.

2.5 Assumptions and Dependencies

i. We have assumed that the target host should be running a suitable OS with JAVA installed.
ii. We have assumed that the user has a sufficiently sound understanding of our system and inputs correct data to the required fields, although many checks are there but we have assumed the user knows what they are doing.

3. Functional Requirements

3.1

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Actor Login</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">An actor logins to the application.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">General Manager, Branch Manager, Salesman</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">None</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">Actors Home Screen Is Displayed.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Actor Selects “login” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System redirects to “Login” page.</td></tr>
<tr><td><b>3</b></td><td>Actor will enter the employee ID and password.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System redirects to Actors designated home page.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action (Wrong Employee ID/Password)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>3</b></td><td></td><td>System will give an error message.</td></tr>
<tr><td colspan="3">Go To 3</td></tr>
</table>

3.2

Identifier	UC-2	
Name	Create Branch	
Summary	General Manager will create a new pharmacy branch.	
Priority	High	
Actors	General Manager	
Pre-condition(s)	UC-1	
Post-condition(s)	New Branch will be created.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	General Manager selects "Add New Branch" button.	
2		System will redirect to "Input Form" page.
3	The General Manager will enter the details for the branch to be created.	
4		System will add new branch to database.
5		System redirects back to General Manager's home page.
Alternate Course of Action (Incorrect/Incomplete Information)		
S#	**Actor Action**	**System Response**
3		System Displays error message.
Go To 3		

3.3

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-3</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Remove Branch</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">General Manager will remove a pharmacy branch.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">High</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">General Manager</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">A branch will be deleted.</td></tr>
<tr><th colspan="3">Typical Course of Action</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>1</b></td><td>General Manager selects “Remove Branch” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System will redirect to “Select Branch” page.</td></tr>
<tr><td><b>3</b></td><td>The General Manager will select the branch to be removed from a list of available branches.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System will remove the selected branch from its database.</td></tr>
<tr><td><b>5</b></td><td></td><td>System redirects back to General Manager’s home page.</td></tr>
<tr><th colspan="3">Alternate Course of Action (Manager Presses Go Back Button)</th></tr>
<tr><th>S#</th><th>Actor Action</th><th>System Response</th></tr>
<tr><td><b>2</b></td><td></td><td>System redirects back to General Manager’s home page.</td></tr>
</table>

3.4

Identifier	UC-4
Name	Set Branch Manager
Summary	General Manager will select a branch manager.
Priority	High
Actors	General Manager
Pre-condition(s)	UC-1
Post-condition(s)	Branch will have a branch manager.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	General Manager selects “Set Branch Manager” button.	
2		System will redirect to “Create Manager” page.
3	The General Manager will enter the details for the branch manager to be created.	
4		System will create a branch manager.
5		System will redirect to “Select Branch” page.
6	The General Manager selects the branch to assign branch manager to.	
7		System will assign created branch manager to specified branch.
8		System redirects back to General Manager’s home page.

Alternate Course of Action (Incorrect/Incomplete Info)

S#	**Actor Action**	**System Response**
3		System Displays error message.

Go To 3

Alternate Course of Action (Branch already has a branch manager)

S#	**Actor Action**	**System Response**
6		System displays prompt if branch manager is to be replaced.
7	General Manager selects yes or no option.	
8		System updates or skips set

		branch manager according to General manager’s selection.
9		System redirects back to General Manager’s home page.

3.5

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-5</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Add Salesman</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Branch Manager adds a Salesman to his Branch.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Branch Manager</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2">New Salesman is added.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Branch Manager selects “Add new Salesman” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System will redirect to “Salesman Info” Page.</td></tr>
<tr><td><b>3</b></td><td>Branch Manager will enter details for salesman to be added.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System will create a salesman.</td></tr>
<tr><td><b>5</b></td><td></td><td>System adds salesman to branch.</td></tr>
<tr><td><b>6</b></td><td></td><td>System redirects back to Branch Managers home page.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action (Incorrect/Incomplete Information)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>3</b></td><td></td><td>System Displays error message.</td></tr>
<tr><td colspan="3">Go To 3</td></tr>
</table>

3.6

<table>
<tr><td colspan="2"><b>Identifier</b></td><td>UC-6</td></tr>
<tr><td colspan="2"><b>Name</b></td><td>Remove Salesman</td></tr>
<tr><td colspan="2"><b>Summary</b></td><td>Branch Manager removes a Salesman from his Branch.</td></tr>
<tr><td colspan="2"><b>Priority</b></td><td>Medium</td></tr>
<tr><td colspan="2"><b>Actors</b></td><td>Branch Manager</td></tr>
<tr><td colspan="2"><b>Pre-condition(s)</b></td><td>UC-1</td></tr>
<tr><td colspan="2"><b>Post-condition(s)</b></td><td>A Salesman is removed.</td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Branch Manager selects “Remove Salesman” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System will redirect to “Salesman ID” Page.</td></tr>
<tr><td><b>3</b></td><td>Branch Manager will select the salesman to be removed from a list.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System removes salesman from database.</td></tr>
<tr><td><b>5</b></td><td></td><td>System redirects back to Branch Managers home page.</td></tr>
</table>

3.7

Identifier	UC-7
Name	Request Stock
Summary	Branch Manager issues requests for additional medicine stock.
Priority	High
Actors	Branch Manager
Pre-condition(s)	UC-1
Post-condition(s)	General Manger is displayed the request.

Typical Course of Action

S#	**Actor Action**	**System Response**
1	Branch Manager selects "Request Medicine" button.	
2		System will redirect to "Medicine Info" Page.
3	Branch Manager will select the Medicine names and corresponding quantities from a list.	
4		System adds request to database.
5		General manager gets a message to approve or reject request.
6		System redirects back to Branch Managers home page.

3.8

Identifier	UC-8	
Name	Handle Re-inventory Request	
Summary	General Manager will accept or reject a re-inventory request from a branch	
Priority	High	
Actors	General Manager	
Pre-condition(s)	UC-1, UC-7	
Post-condition(s)	Branch is granted or denied new stock of medicine.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1		System Displays “Re-Inventory Request” message on General Managers Home Page.
2	General Manager Selects “Re-Inventory Request” message.	
3		System Displays the request details and redirects to “Handle Request” page.
4	General Manager chooses “Accept”.	
5		System adds the required medicine and quantities to the branch.
6		System Displays “Request Accepted” message on Branch Managers Home Page.
7		System redirects back to General Manager’s home page.
Alternate Course of Action (Manager Chooses Reject)		
S#	**Actor Action**	**System Response**
6		System Displays “Request Rejected” message on Branch Managers Home Page.
Go To 7		

3.9

Identifier	UC-9	
Name	Sell Medicine to Customer	
Summary	Salesman sells medicine to a customer.	
Priority	High	
Actors	Salesman	
Pre-condition(s)	UC-1	
Post-condition(s)	A Sales is made.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	Salesman selects “Sell Medicine” button.	
2		System redirects to “Sell Medicine” page.
3	Salesman selects medicines and quantities from list of available medicines according to customer request.	
4		System adds Sale to database.
5		System Updates branch inventory.
6		System redirects to Salesman homepage.
Alternate Course of Action (Required Medicine Quantity is not available)		
S#	**Actor Action**	**System Response**
4	Salesman informs customer.	
5	Salesman Cancels Sale.	
6		System displays “Low Stock” message to branch manager.
Go To 6		

3.10

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-10</td></tr>
<tr><td><b>Name</b></td><td colspan="2">Search Medicine</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Salesman searches for a medicine in inventory</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Salesman</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2"></td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Salesman selects “search Medicine” option.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System redirects to “search” page</td></tr>
<tr><td><b>3</b></td><td>Salesman inputs medicine name</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System displays the search result</td></tr>
<tr><td><b>5</b></td><td>Salesman selects “home page” option</td><td></td></tr>
<tr><td><b>6</b></td><td></td><td>System redirects to homepage</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action (Medicine Name does not exist)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>4</b></td><td></td><td>System displays “Medicine does not exist” message to Salesman.</td></tr>
<tr><td colspan="3">Go To 6</td></tr>
</table>

3.11

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-11</td></tr>
<tr><td><b>Name</b></td><td colspan="2">View Inventory</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">Branch Manager Views the current inventory of his branch.</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">Branch Manager</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2"></td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>Branch Manager selects “View Inventory” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System Displays the current inventory.</td></tr>
<tr><td><b>3</b></td><td>Branch Manager Selects “Home Page” button.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System Redirects to Home Page.</td></tr>
</table>

3.12

<table>
<tr><td><b>Identifier</b></td><td colspan="2">UC-12</td></tr>
<tr><td><b>Name</b></td><td colspan="2">View Branch Report</td></tr>
<tr><td><b>Summary</b></td><td colspan="2">General Manger views the report of sales of a selected branch</td></tr>
<tr><td><b>Priority</b></td><td colspan="2">Medium</td></tr>
<tr><td><b>Actors</b></td><td colspan="2">General Manager</td></tr>
<tr><td><b>Pre-condition(s)</b></td><td colspan="2">UC-1</td></tr>
<tr><td><b>Post-condition(s)</b></td><td colspan="2"></td></tr>
<tr><td colspan="3"><b>Typical Course of Action</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>1</b></td><td>General Manager selects “View Branch Report” button.</td><td></td></tr>
<tr><td><b>2</b></td><td></td><td>System Displays redirects to “Get Branch Report” page.</td></tr>
<tr><td><b>3</b></td><td>General Manager enters code of Branch for which to view report.</td><td></td></tr>
<tr><td><b>4</b></td><td></td><td>System Displays report of the selected branch.</td></tr>
<tr><td><b>5</b></td><td>General Manager Selects “Home Page” button</td><td></td></tr>
<tr><td><b>6</b></td><td></td><td>System Redirects to General Manager’s home page.</td></tr>
<tr><td colspan="3"><b>Alternate Course of Action (Incorrect Branch Code)</b></td></tr>
<tr><td><b>S#</b></td><td><b>Actor Action</b></td><td><b>System Response</b></td></tr>
<tr><td><b>3</b></td><td></td><td>System Displays error message.</td></tr>
<tr><td colspan="3">Go To 3</td></tr>
</table>

4. Nonfunctional Requirements

4.1 Performance Requirements

i. The system must be interactive and the delays involved must be less. In every action-response of the system, there are no immediate delays.
ii. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds. There shouldn't be any delay in our database transactions.
iii. In case of opening databases, sorting questions and evaluation there are no delays and the operations are performed in less than 2 seconds for opening, sorting and computing for 95% of the files.
iv. Each branch should have a local storage for database.
v. Form validation should not require users to re-enter correctly filled fields. Forms filled by users must be validated in such a way so that the user only has to refill the incorrect field.

4.2 Security Requirements

i. The main security concern is for users account hence proper login mechanism should be used to avoid hacking.
ii. System will provide each user a separate account on our software.
iii. System will allow users to use their desired passwords for their account safety.
iv. System will provide all users to have an authorized access to the system.
v. Moreover, this software is by design encapsulated. Information of each user is hidden from other user. Privacy concerns are dealt carefully.

Appendix A: Glossary

UC = Use Case

GM = General Manager

DB = Database

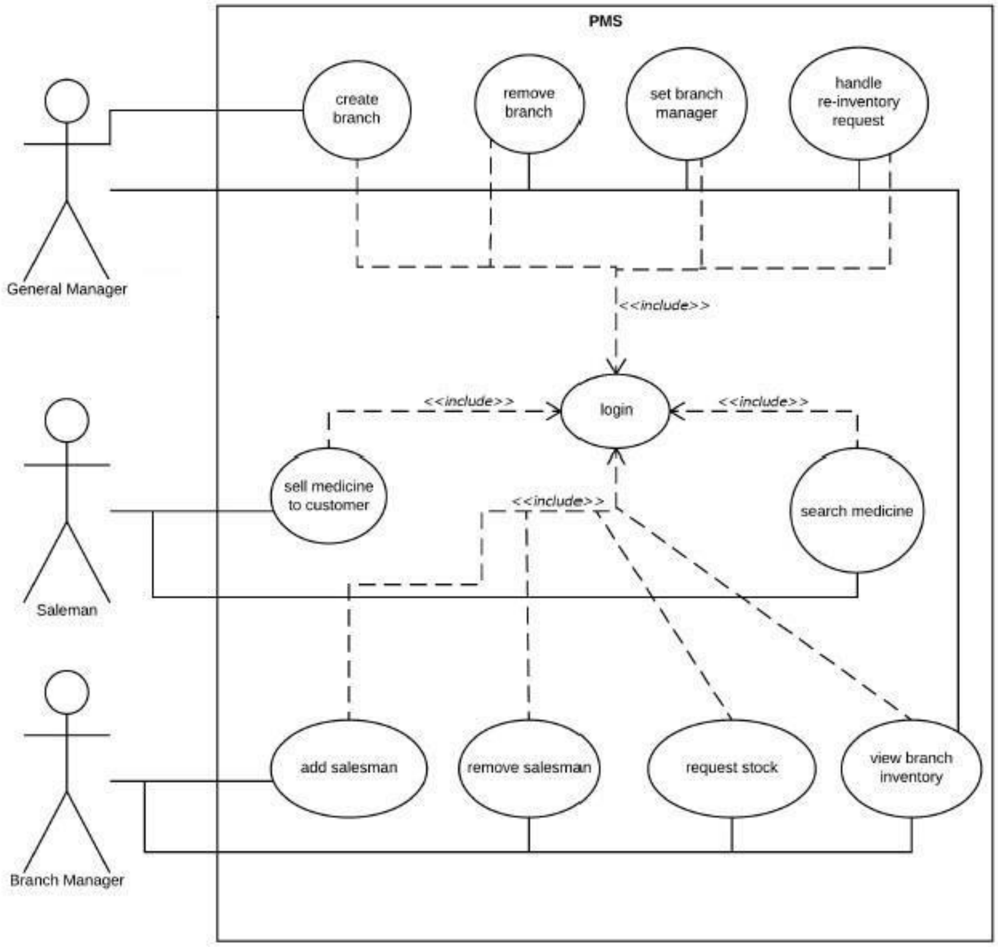
PMS = Pharmaceutical Management System

ER = Entity Relationship

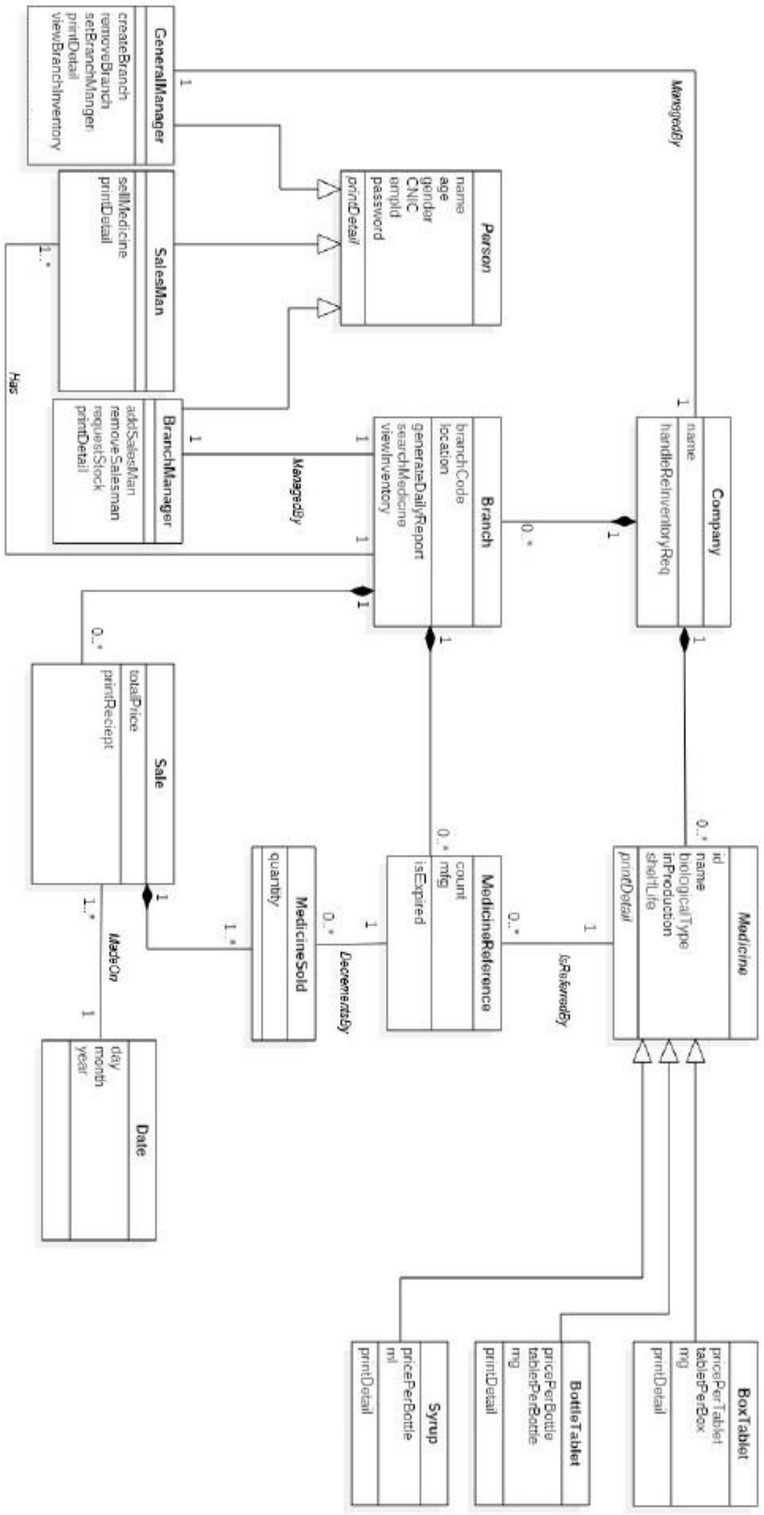
BM = Branch Manager

Appendix B: Analysis Models

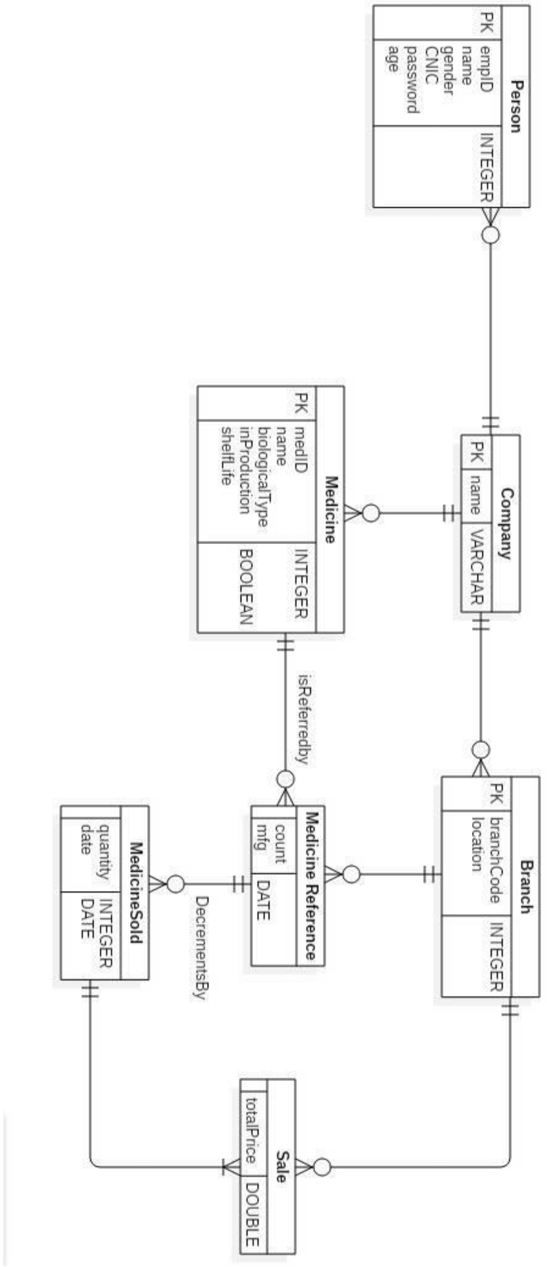
1. USE Case Diagram:



2. Class Diagram:

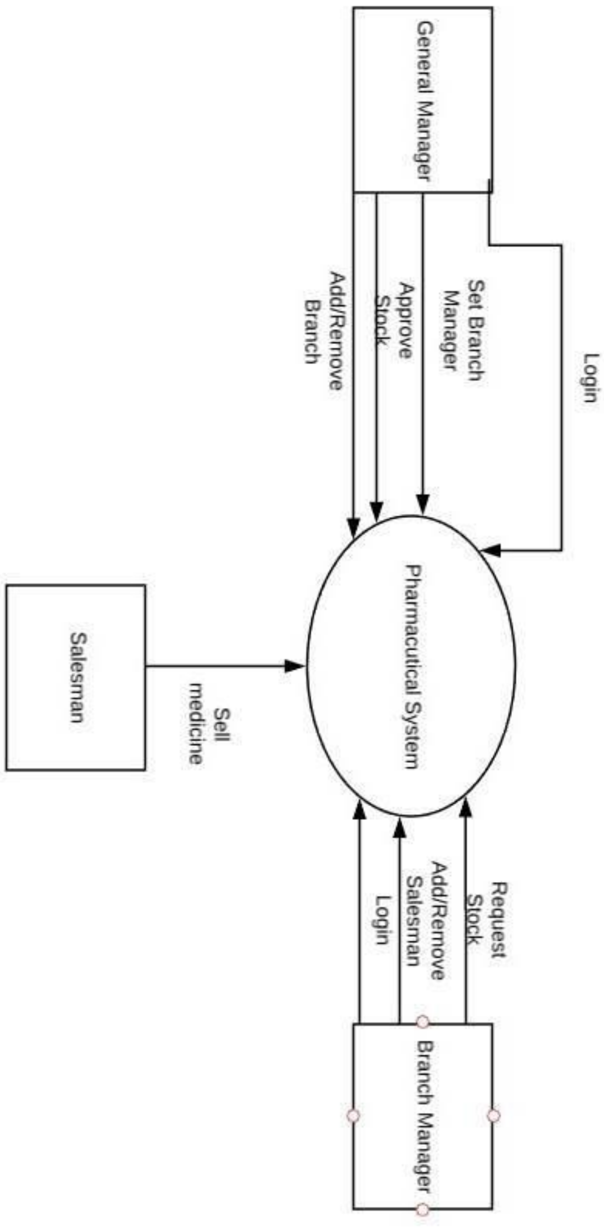


3. ER Diagram:

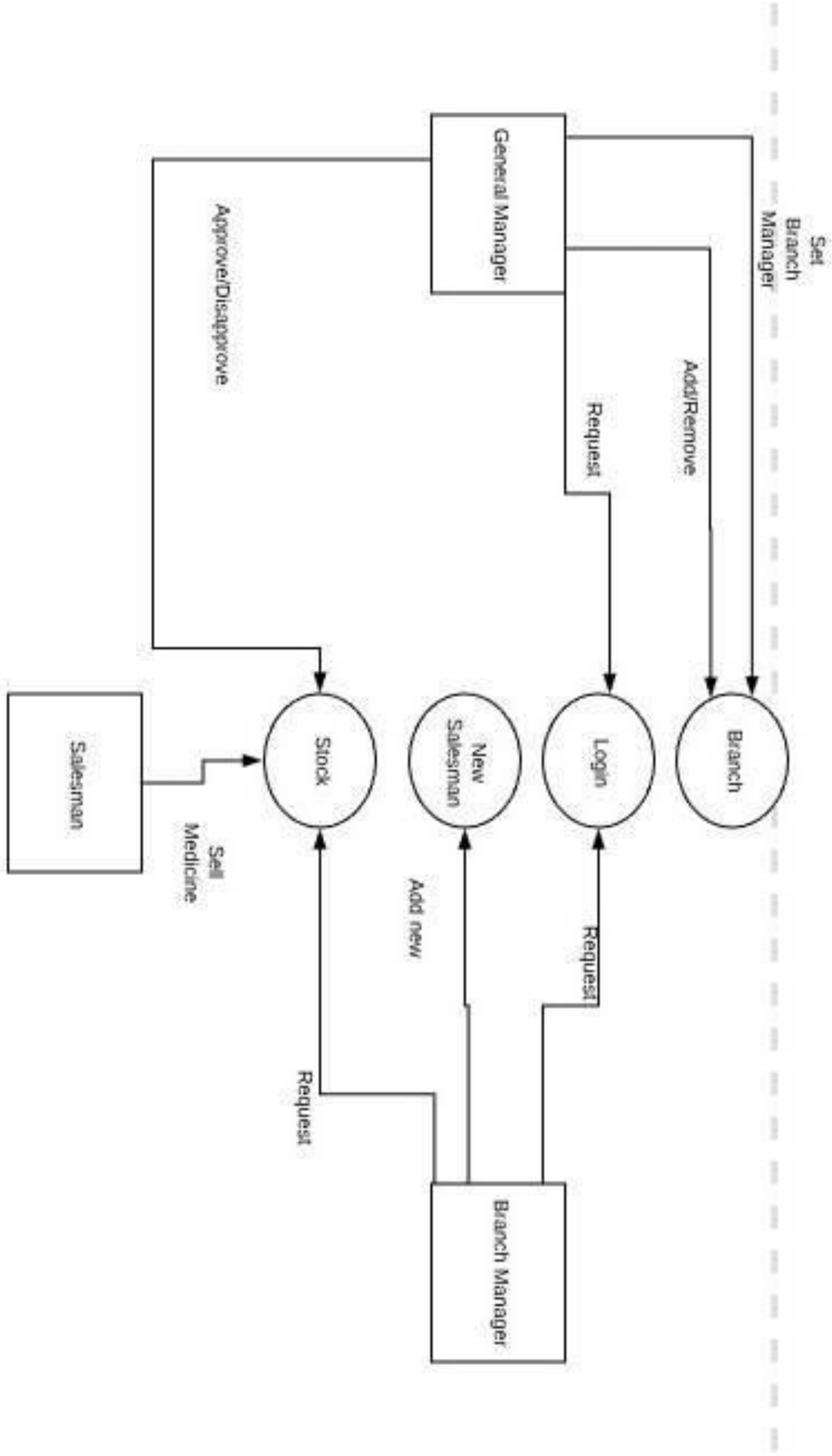

Person
PK
empID
name
gender
CNIC
password
age
INTEGER
Company
PK
name
VARCHAR
Medicine
PK
medID
name
biologicalType
inProduction
shelfLife
INTEGER
BOOLEAN
isReferredby
Branch
PK
branchCode
location
INTEGER
Medicine Reference
count
mfg
DATE
MedicineSold
quantity
date
INTEGER
DATE
DecrementsBy
Sale
totalPrice
DOUBLE

4. Data Flow Diagram:

4.1 Context Level:

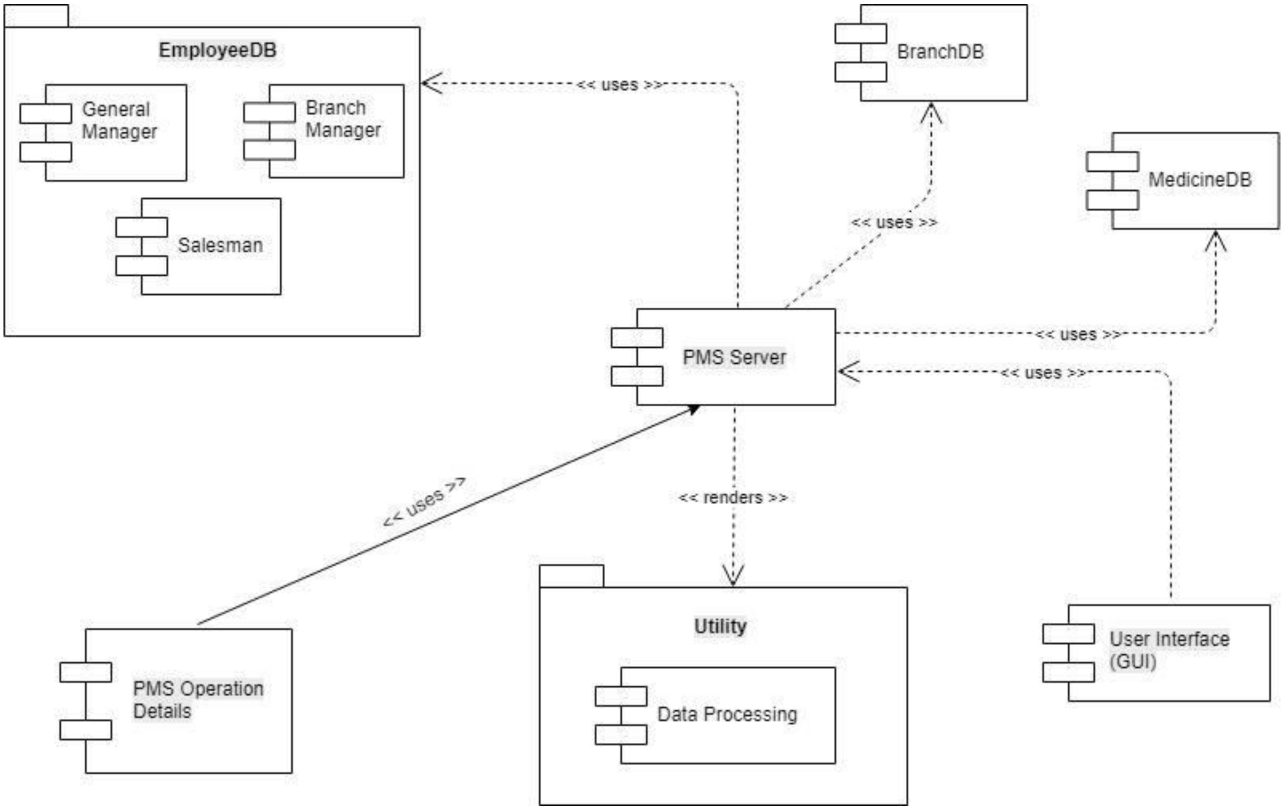


4.2 Final Level:



Appendix C: Design

1. Component Diagram


EmployeeDB
General Manager
Branch Manager
Salesman
<< uses >>
BranchDB
<< uses >>
MedicineDB
PMS Server
<< uses >>
<< uses >>
<< uses >>
<< renders >>
PMS Operation Details
Utility
Data Processing
User Interface (GUI)