

Software Requirements Specification

Version 1.0

Pharmaceutical Management System

Team 10

Member Name	Member Roll #	Primary Responsibility
Ghayyur Ali	12L-4276	General Manager
Usama Saqib	15L-4064	Salesman
Usman Majied	15L-4033	Branch Manager
Muhammad Faiq	15L-4266	GUI

Table of Contents

Table of Contents	ii
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. Functional Requirements	2
2.1.....	2
2.2.....	3
2.3.....	4
2.4.....	5
2.5.....	6
2.6.....	7
2.7.....	8
2.8.....	9
2.9.....	10
2.10.....	11
2.11.....	12
2.12.....	13
3. Nonfunctional Requirements	14
3.1 Performance Requirements	14
3.2 Security Requirements	14
Appendix A: Glossary.....	15
Appendix B: Analysis Models.....	16
1. USE Case Diagram:	16
Class Diagram:	17
ER Diagram:.....	18
Data Flow Diagram:.....	19

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 **Java¹ 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. Users will be able to sign up and there will be a system generated user too. Users will be able to create branches, add/remove sales man in a branch, and display branch statistics of every branch. Branch will be able to request for more stock as well as search inventory and sell medicine. Each user will have 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

<https://www.java.com/en/download/>
<http://www.cclpharma.com/>
<https://www.selmorpharma.com/>

2. Functional Requirements

2.1

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

2.2

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 Managers home page.
Alternate Course of Action (Incorrect/Incomplete Information)		
S#	Actor Action	System Response
3		System Displays error message.
Go To 3		

2.3

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

2.4

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.
----------	--	---

2.5

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

2.6

Identifier		UC-6
Name		Remove Sales Man
Summary		Branch Manager removes a Sales Man from his Branch.
Priority		Medium
Actors		Branch Manager
Pre-condition(s)		UC-1
Post-condition(s)		A Sales Man is removed.
Typical Course of Action		
S#	Actor Action	System Response
1	Branch Manager selects “Remove Sales Man” button.	
2		System will redirect to “Sales Man ID” Page.
3	Branch Manager will select the sales man to be removed from a list.	
4		System removes sales man from database.
5		System redirects back to Branch Managers home page.

2.7

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.

2.8

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 Managers 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		

2.9

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

2.10

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

2.11

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

2.12

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

3. Nonfunctional Requirements

3.1 Performance Requirements

1. System will be efficient and optimized for better performance. Turnaround time will be as low as possible.
2. As this is a management software so our top priority is on time query execution.
3. There shouldn't be any delay in our database transactions.
4. Each branch should have a local storage for database.
5. System should must have compatible Java installed.
6. Results should be displayed in real time.
7. 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.

3.2 Security Requirements

1. Each user has a separate account on our software.
2. Users should use their desired passwords for their account safety.
3. Security will be the main concern. All users of our system will have an authorized access to the system.
4. 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

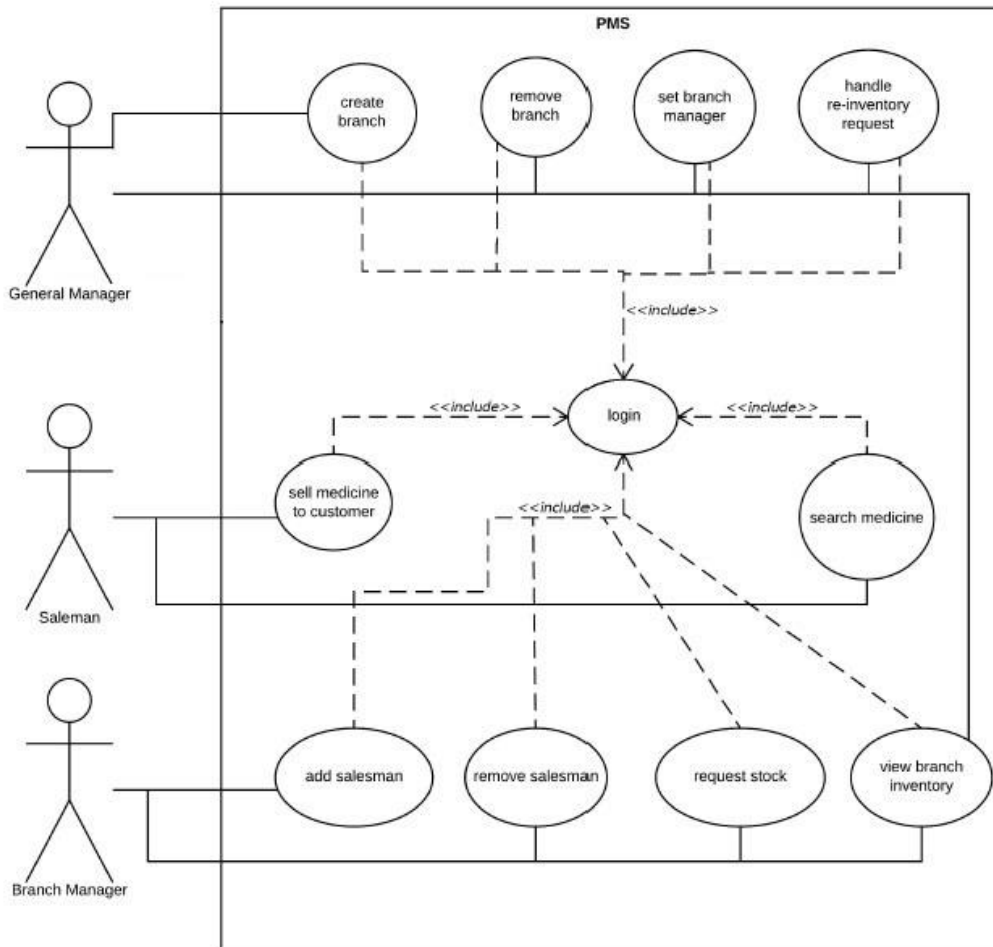
DB = Database

PMS = Pharmaceutical Management System

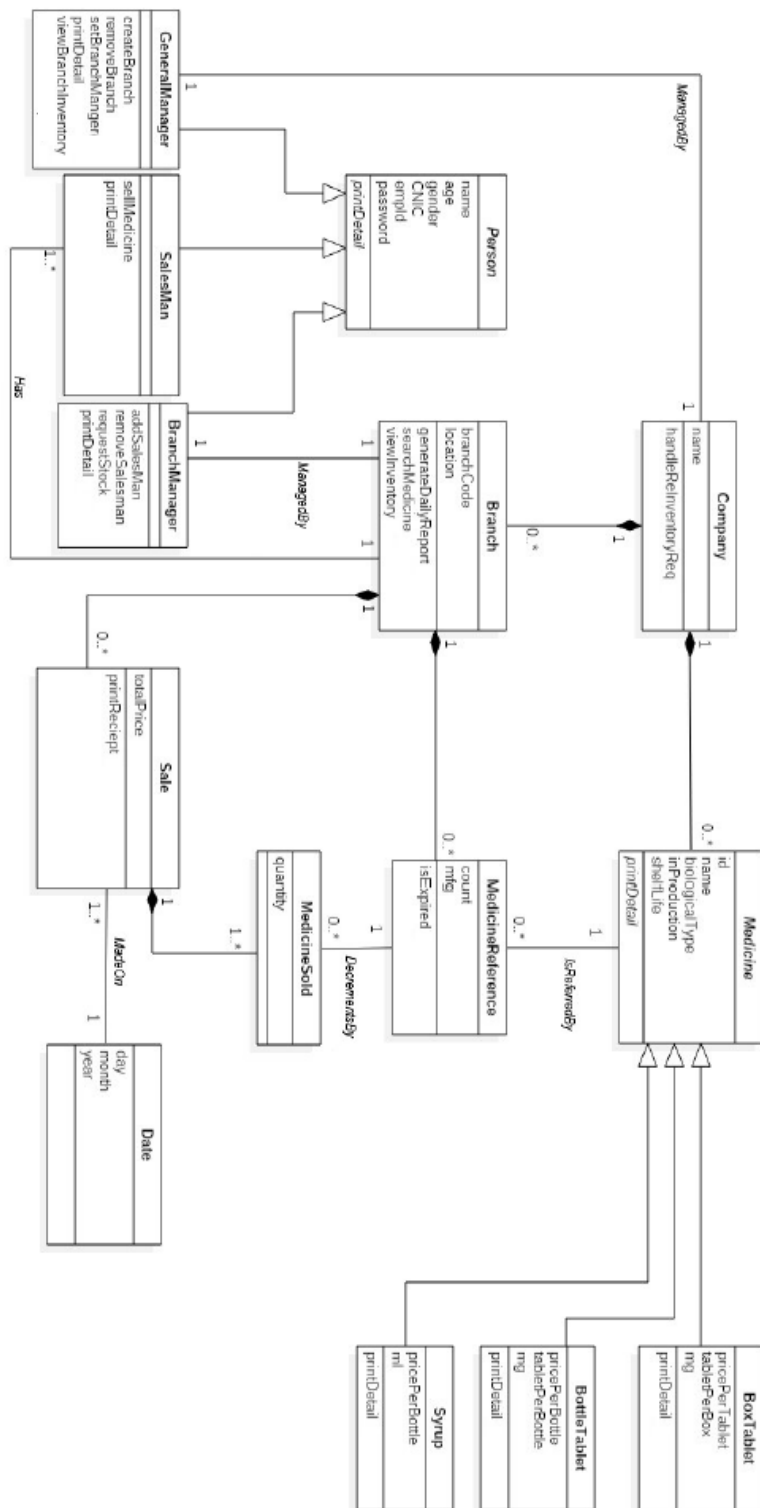
ER = Entity Relationship

Appendix B: Analysis Models

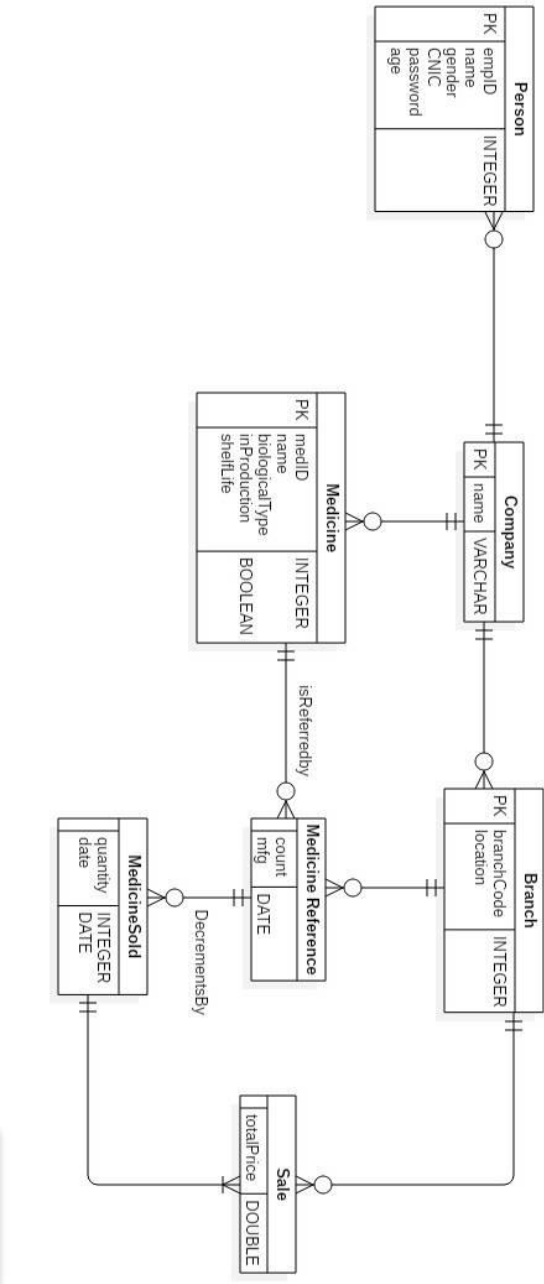
1. USE Case Diagram:



Class Diagram:

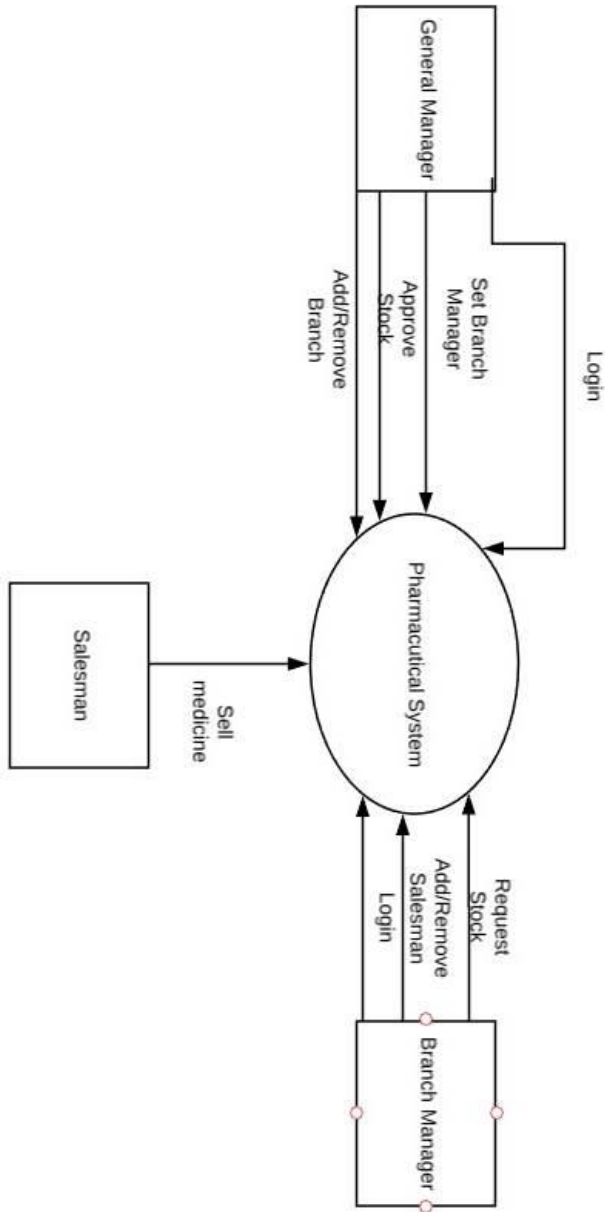


ER Diagram:



Data Flow Diagram:

Context Level:



Final Level:

