



**Atal Bihari Vajpayee  
Indian Institute of Information Technology  
and Management (ABV-IIITM), Gwalior**  
(An Institute of National Importance, Ministry of Education, Government of India)

## **SOFTWARE ENGINEERING**

# **Software Requirement Specification Pharmacy Management System**

Submitted to:

Dr.Santosh Singh Rathore

Group Members:

MARAM VIGNESH(2021 BCS-039)

NARRA ABHIGNA(2021 BCS-048)

PATHLAVATH SUDEENDRA(2021 BCS-052)

RAVI JWALANA(2021 BCS-056)

# Table of Contents

## 1. Introduction

1.1. Purpose

1.2. Scope of the project

1.3. Glossary

1.4. References

## 2. Overall Description

2.1 System environment

2.2 Functional requirement specification

    2.2.1 Super admin use case

    2.2.2 System user use case

    2.2.3 Shopkeeper use case

    2.2.4 Customer use case

2.3 User Characteristics

2.4 Non functional requirement specification

## 3. Requirements specification

3.1 External interface requirements

3.2 Functional requirements

3.3 Detailed non-functional requirements

# 1. Introduction

A pharmacy management system is a type of management that is intended to boost efficiency, safety, and accuracy in a pharmacy. It is a computer-based system that aids pharmacists in managing inventories, costs, and other factors including medical safety. The system will also provide a report with a list of products that will expire after a certain date before the product itself does. For a set period, such as every month, it also requires manual entry when new batches of medications arrive and when they are moved out of the pharmacy. This involves gathering information about the medications, such as their expiration date, date of purchase, how many of each type are still in stock, and where they are located within the pharmacy. In the pharmacy, a manual system is now in use. Each drug that is on hand in the pharmacy needs to be manually checked by the pharmacist. This typically results in errors as the pharmacist's workload rises.

The pharmacy has a very big customer base as a result of its size and high level of customer care. These clients typically visit the drugstore for medicines is growing at the moment, which adds to the pharmacists already hearvices after they leave work. The number of consumers visiting increases the burden.

It is challenging for the pharmacist in this situation to serve consumers quickly. In the meantime, the pharmacist must guarantee consumer pleasure with their services to maintain them. The problems cause delays in providing services to customers, which slows down sales and increases the chance of losing important clients in the long run.

Writing the order requires a significant amount of time since the pharmacist must review the stock balance and make an educated guess as to how much to order based on the figures .

Medicines should not be used after their expiration dates.

This project work will notify the pharmacist of medications that are ready to expire, prohibiting the sale of those medications and offering a remedy to the above mentioned issues.

## 1.1 Purpose

The goal of this project is to provide software for a pharmaceutical store's efficient management that can accomplish the following goals: supplying information on drug stock figures to ensure efficient law enforcement maintaining an accurate database by giving users the choice to update their stock of medications. ensuring effective service and activity monitoring in order to increase system efficiency.

- 1) to manage the pharmacy's drug inventory as best as possible by keeping an eye on the flow of medications there.
- 2) to guarantee that access is restricted to a certain degree based on functionality and role.
- 3) to make sure the interface is simple to use.
- 4) to be able to produce a report in a certain amount of time

## 1.2 Scope of project

The focus of this project is on a pharmacy's operations, which include enhancing access to care in the estate and nearby communities, decreasing hospital and long-term care admissions, improving health outcomes, and making the best use of available resources. A computer-based management system is required to

increase a pharmacy's efficiency, and it is a crucial component of any contemporary, ever-evolving society.

## 1.3 Glossary

1. Database: It is a group of tables which stores the information of users.
2. Customer: The person who avails the services of the website.
3. HTTP: HyperText Transfer Protocol
4. Software Requirement Specification: A document that completely describes all the functions of a proposed system and the constraints under which it must operate. For example, this document.

## 1.4 References

1.Wikipedia for pharmacy management system:

Pharmacy management system - Wikipedia

2.See <http://www.psnc.org.uk/pages/itsuppliers.html>

3.Borggren L. Computers in pharmacy: improve efficiency and productivity. Comput Healthcare.

1984;5(7):39–40.

For reference of data model and implementation

## 2. Overall Description

### 2.1 System environment

Hardware:

1. A computer system featuring a display, hard disc, RAM, and processor
2. A barcode scanner to read drug labels
3. A printer for receipts and medicine labels
4. A point-of-sale (POS) system or cash register for handling transactions
5. A data protection backup system

Software:

1. Operating system (e.g., Windows, Linux)
2. Database management system (e.g., MySQL, Oracle)
3. Pharmacy management software (e.g., PioneerRx, PrimeRx)
4. Anti-virus and anti-malware software
5. Web browser (e.g., Chrome, Firefox) for accessing online resources

Network:

Internet connection for accessing online resources and carrying out electronic transactions Local area network (LAN) for tying together computers and other equipment inside the pharmacy

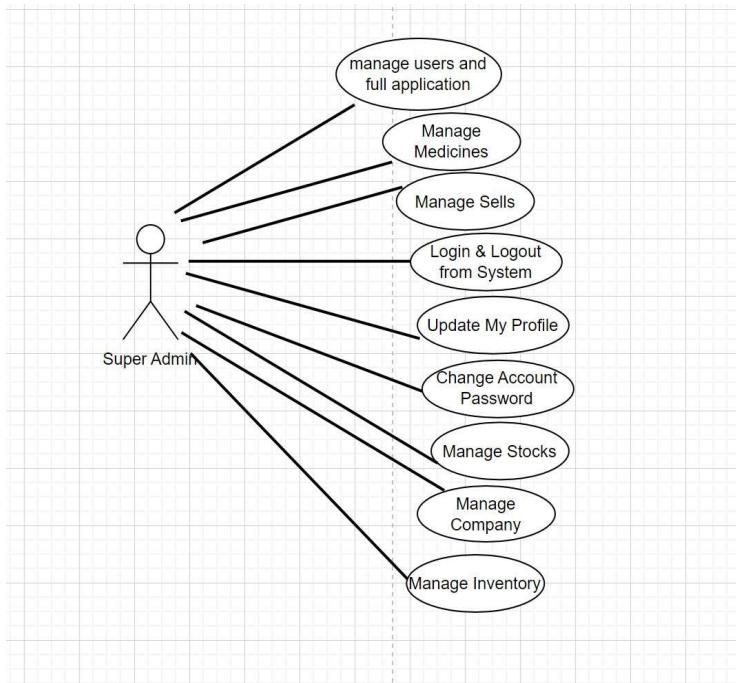
Using a Virtual Private Network (VPN) to access the system from a secure distance. The system environment must also comply with any applicable legal mandates, such as HIPAA (Health Insurance Portability and Accountability Act), which protects patient data.

## 2.2 Functional requirement specification



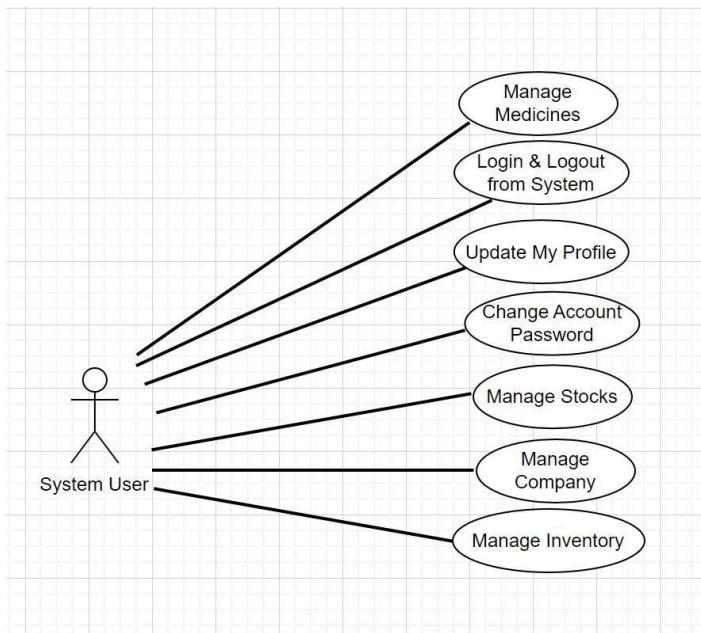
### 2.2.1 Super admin use case

Super admin has the following set of use cases:



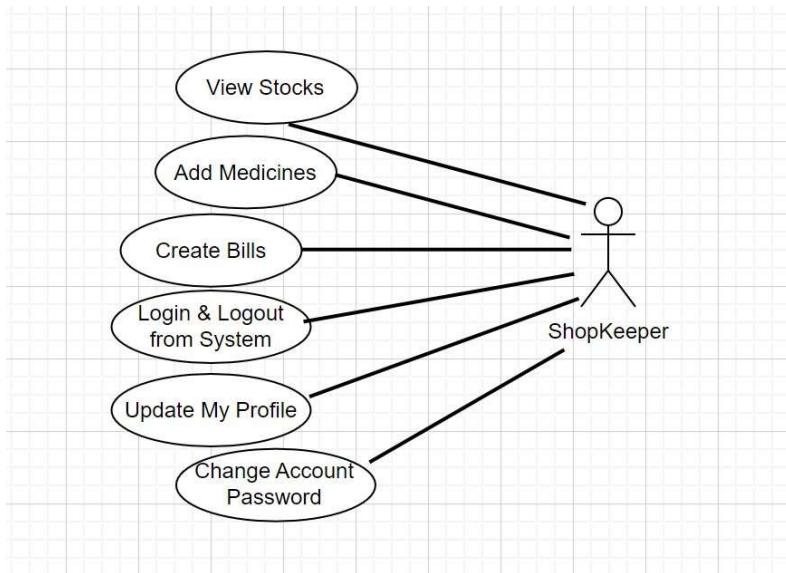
## 2.2.2 System user use case

System user has the following set of use cases:



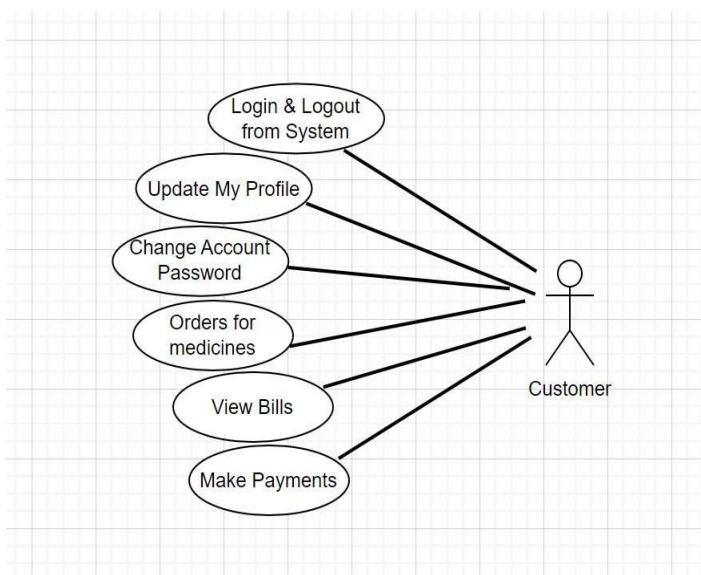
### 2.2.3 Shopkeeper use case

Shopkeeper has the following set of use cases:



### 2.2.4 Customers use case

Customers have the following set of use cases:



## **2.3 User Characteristics**

Different users can utilize this feature with restricted access. For simple management, access to certain functionalities can be restricted for different users. There are four categories for this authentication, notably Super admin, system user, shopkeeper, customers.

### **Super admin:**

The system admin can manage pharmacy, medicines, stocks, company, inventory, sales, users, and full pharmacy management system operations.

### **System user:**

The system user can manage pharmacy, medicines, stocks, company, inventory, sales.

### **Shopkeeper:**

The shopkeeper can add medicines, create bills, print bills, collect payments and view stocks.

### **Customers:**

The customer can order for medicines, view their bills and make payments for their purchase.

## **2.4 Non-Functional Requirement Specification**

**Performance:** The system should be able to process many transactions without noticeably slowing down. Many users should be allowed to use the system at once.

**Reliability:** Any downtime or system failure should be kept to a minimum, and the system should be extremely dependable. To prevent data loss, the system needs to have a backup and recovery system.

**Usability:** The system ought to have an intuitive user interface and be simple to use. Clear instructions and help features should be included in the system, which should be created with the user in mind.

**Security:** The system needs to be very secure, with the right safeguards in place to prevent unwanted access to private information. Any potential security threats should be able to be recognised and avoided by the system.

**Scalability:** The system must be scalable in order to accommodate growing data storage and load demands without suffering performance deterioration.

**System compatibility** refers to how well the hardware, software, and operating system work together.

**Accessibility:** The system should be compliant with accessibility standards and accessible to users with impairments.

**Maintainability:** The system needs to be simple to keep up to date, with clear instructions and an easy-to-use interface for troubleshooting.

**Integration:** The system must be able to work with other computer programmes, such as inventory or accounting software.

**Data Integrity:** The system must guarantee data integrity and provide the necessary safeguards against data loss or corruption. To prevent data loss, the system needs backup and recovery measures in place.

### 3. Requirements specification

#### 3.1 External interface requirements

**Electronic Prescriptions:** Doctor's offices and other healthcare providers should be able to send electronic prescriptions to the pharmacy management system.

**Health Information Exchanges:** In order to communicate and receive patient health information, the pharmacy management system should be able to connect with HIEs.

**Verification of Insurance and Benefits:** In order to confirm patient insurance coverage and eligibility for prescription coverage, the pharmacy management system should be able to communicate with insurance providers and benefit managers.

**Drug Databases:** To ensure proper prescription doses, interactions, and drug use reviews, the pharmacy management system should be able to connect with drug databases.

**Point-of-Sale Systems:** To track sales, inventory, and revenue, the pharmacy management system should be able to communicate with point-of-sale (POS) systems.

**Electronic Medical Records (EMRs):** To keep precise patient data and prescription histories, the pharmacy management system should be able to interact with EMRs.

**Pharmacy Benefit Managers:** In order to obtain data on drug costs, formularies, and coverage, the pharmacy management system should be able to communicate with PBMs.

**Electronic Prior Authorization:** To speed up the prior authorisation process for pharmaceuticals that need insurance company preapproval, the pharmacy management system should be able to interact with electronic prior authorization (ePA) systems.

**Medical Billing and Claims Processing:** In order to submit claims for drug reimbursement, the pharmacy management system should be able to connect with medical billing and claims processing systems.

**Patient Portals:** In order to give patients access to their prescription histories, refill requests, and other medication-related information, the pharmacy management system should be able to communicate with patient portals.

## 3.2 Functional requirements

### 3.2.1 Manage users and full application

The system should allow for the creation, modification, and deletion of user accounts. It should also provide various levels of access control for different user roles, such as a pharmacist, a salesperson, or an administrator.

### **3.2.2 Manage medicines**

The system should allow for the management of medicines, including adding, modifying, and deleting their details, such as name, description, dosage, and price.

### **3.2.3 Manage sales**

The system should allow for the management of sales transactions, including creating, modifying, and deleting sales orders and invoices.

### **3.2.4 Manage stocks**

The system should allow for the management of inventory and stock levels of medicines, including adding, modifying, and deleting stock details, such as batch numbers, expiry dates, and quantities.

### **3.2.5 Manage company**

The system should allow for the management of the company, including adding, modifying, and deleting the company details, such as name, address, and contact information.

### **3.2.6 Manage inventory**

The system should allow for the management of the inventory, including adding, modifying, and deleting inventory details, such as current stock levels, reorder levels, and minimum stock levels.

### **3.2.7 Login and logout from system**

The system should allow for secure login and logout, ensuring user privacy and security.

### **3.2.8 Update my profile**

The system should allow users to view and update their personal profile information, such as name, contact information, and address.

### **3.2.9 Change account password**

The system should allow users to change their account passwords to ensure security.

### **3.2.10 View stocks**

The system should allow users to view the current stock levels of medicines, including details such as batch numbers, expiry dates, and quantities.

### **3.2.11 Add medicines**

The system should allow authorized users to add new medicines to the inventory, including details such as name, description, dosage, and price.

### **3.2.12 Create bills**

The system should allow authorized users to create bills for sales transactions, including details such as customer information, medicines purchased, and prices.

### **3.2.13 Orders for medicine**

The system should allow authorized users to place orders for medicines with suppliers, including details such as the name of the medicine, the quantity ordered, and the expected delivery date.

### **3.2.14 View bills**

The system should allow authorized users to view bills for sales transactions, including details such as customer information, medicines purchased, and prices.

### 3.2.15 Make payments

The system should allow authorized users to make payments for bills and purchase orders, including details such as payment amount, date, and method of payment.

## 3.3 Detailed non-functional requirements

### 3.3.1 Logical Structure of the Data

User: This entity stands in for a system user account and includes attributes like the user ID, name, password, role, and contact details.

Medicine: This entity in the system represents a drug and includes features including the drug's ID, name, dose, cost, and stock level.

Sales: This entity, which in the system represents a sales transaction, has features including the sale ID, customer details, date, and total price.

The stock entity, which has features including the stock ID, drug ID, batch number, expiration date, quantity, and cost, represents the system's inventory of pharmaceuticals.

Company: This entity, which comprises attributes like the company ID, name, address, and contact information, represents the company information in the system.

Order: This object is a representation of a supplier-placed purchase order for medications and includes features including the order ID, supplier details, medicine ID, quantity, and anticipated delivery date.

**Bill:** This entity, which in the system represents a bill for a sales transaction, has attributes such as a bill ID, sales ID, customer information, medicine information, date, and total price.

**Payment:** This object represents a payment made in relation to a bill or a purchase order and includes features like the payment ID, the bill ID or order ID, the payment amount, the date, and the mode of payment.

### **3.3.2 Security**

Any software application needs to be secure, but pharmacy management systems are especially vulnerable since they handle private data like customer information, medication inventories, and financial transactions. The security of a pharmacy management system can be ensured by the security methods listed below:

Install access control measures to make sure that only individuals with the proper authorization can access and alter system data. User login, authorization, and permission settings can be used to do this.

**Safe Login:** Use two-factor authentication, strong password guidelines, and secure communication protocols (like HTTPS) to protect the user's login information from unauthorized access.

Employ encryption measures to safeguard sensitive data, including client information, drug inventories, and financial activities. Both data at rest and in transit can be encrypted.

In the event of a system failure or data breach, make sure data is not lost by implementing backup and recovery procedures. To guarantee that the data can be recovered, routine backups can be taken.

Implementing an audit trail will allow you to keep track of user actions within the system, including failed login attempts, data changes, and system adjustments. As a result, unwanted access and data breaches may be easier to spot and stop.

Security testing: Do routine security testing and vulnerability analyses to find and fix system security flaws. By doing so, the system can be made safer against potential threats.

Frequent Updates: To ensure that known security vulnerabilities are fixed, keep the system and all its components up to date with the most recent security patches and upgrades.