Software Requirements Specification

Prepared by: *Team MedInnov*

Date: *6th September 2024*

Table of Contents

---

1. Introduction

1.1 Purpose

This SRS is supposed to define the software requirements for the Drug Inventory and Supply Chain Tracking System. The system should be able to ensure that procurement, distribution, and monitoring of drugs in healthcare institutions are effective, thus ensuring the right quantity of the right product at the right place and at the right time.

1.2 Scope

This will smoothen the entire inventory management of drugs and the supply chain by:

- Improving procurement efficiency using predictive analytics.
- Tracking and ranking vendor performance.
- Monitoring drug usage and wastage.
- Dashboards for real-time insight.
- Using OCR to track sales data.

This SRS document covers the functional and non-functional requirements, system architecture, technical specifications, and testing plans for the prototype system.

1.3 Definitions, Acronyms, and Abbreviations

- **OCR:** Optical Character Recognition
- **ML:** Machine Learning
- **AI:** Artificial Intelligence
- **API:** Application Programming Interface
- **SRS:** Software Requirements Specification
- **UI:** User Interface
- **UX:** User Experience

1.4 References

- **Problem Statement:** Government of NCT of Delhi, Drug Inventory Management Challenge, 2024
- **Django Documentation:** https://docs.djangoproject.com

2. System Overview

The Drug Inventory and Supply Chain Tracking System is a web-based application intended for healthcare institutions and suppliers of drugs. It will manage the inventory of medicine and track the supply chain, using AI/ML models to forecast demand, monitor consumption, and rank the vendors.

3. Functional Requirements

3.1 Data Collection (OCR)

- It is in this regard that the system will capture data on drug sales by digitizing handwritten or printed documents using OCR.

3.2 Predictive Analysis

- The system shall project future drug demand based on past trends.
- With AI-powered models studying seasonal trends, outbreaks, and local healthcare statistics, predictions will be more accurate.

3.3 Vendor Ranking

- Ranking of vendors based on operational performance metrics such as delivery time, order accuracy, and contractual adherence.

3.4 Drug Usage Monitoring

- It shall monitor the consumption of drugs in various hospitals and medical institutions.
- Reports on drugs that are most in use will be generated.

3.5 Wastage Analysis

- The system shall carry out analysis of wastage by highlighting areas of overstocking or poor management of drugs.
- It will also propose ways of inventory optimization, thereby reducing wastage.

3.6 Dashboard & Reporting

- It must be able to deliver real-time insights through an interactive dashboard.
- Users will be able to gain insight into consumption of drugs, demand forecasts, vendor rankings, and wastage reports.

3.7 User Management

- There shall be a role-based access control feature within the system where admins, managers, and users can do different functions.

4. Non-Functional Requirements

4.1 Performance

- The system should support at least 100 concurrent users without performance degradation.
- The predicted models should provide the output in 1-2 minutes.

4.2 Security

- The system shall provide a role-based access control.
- All data should be encrypted during transport and storage.

4.3 Scalability

- It should be able to scale to support new healthcare institutes in different regions.

4.4 Usability

- The interface should be easy to use, where users do not need much training.

5. System Design

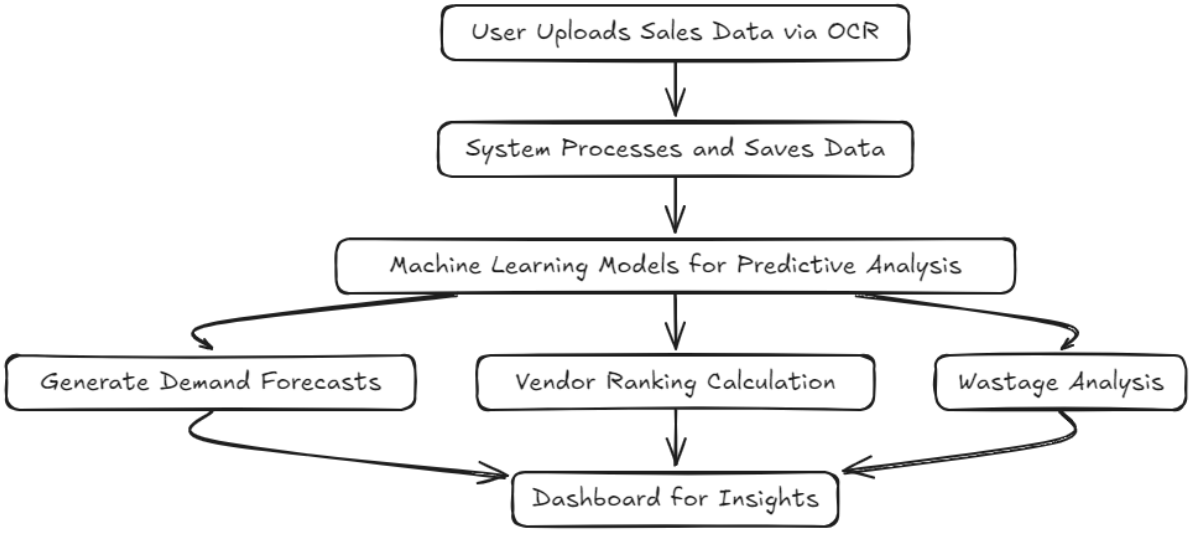
5.1 Architecture

- The system will follow a three-tier architecture:
    - Frontend: Developed using HTML/CSS/JavaScript and Angular for dynamic rendering.
    - Backend: Built with Django for managing APIs, data processing, and business logic.
    - Database: PostgreSQL for structured data storage and fast querying.

5.2 Interfaces

- OCR API: Used for uploading and digitizing documents containing sales data.
- ML Models: Used for predictive analysis, vendor ranking, and wastage analysis.
- Database Interfaces: Will interact with a relational database to store sales, inventory, and other related data.

6. Data Flow Diagrams



7. Technical Specifications

7.1 Software Components

- **Django:** Web framework for backend processing.
- **React/JavaScript:** Frontend framework for building user interfaces.
- **PostgreSQL:** Database system for managing structured data.
- **Python:** Language for implementing AI/ML models and data analysis tools.

- **Scikit-learn:** Libraries for building ML models.
- **Tesseract OCR:** Used for processing sales documents.

7.2 Hardware Requirements

- Server with a minimum of 8 GB RAM and 100 GB of storage.
- Cloud services for hosting and scaling (AWS or equivalent).

8. Validation and Testing

8.1 Unit Testing

- Unit tests will be written for each individual module (OCR, ML Models, Vendor Ranking, etc.).

8.2 Integration Testing

- The system will undergo integration testing to ensure all components (OCR, data processing, ML, etc.) work seamlessly together.

8.3 User Acceptance Testing (UAT)

- UAT will be conducted with a sample of healthcare institutions to gather feedback on usability and functionality.

9. Appendix

- **Glossary:** Definitions of key terms used in this SRS document.
- **References:** Links to resources and documentation used during development.