

Milestone 1

Software Engineering Project: AI-powered Inventory Management System for a Pharmacy.

Identification of Users

Based on the organizational structure, the system will cater to three distinct user tiers:

Primary Users

Direct users of the application.

- Staff Members: Managing day-to-day inventory, processing billing, and restocking alerts.
- Patients: Engaging with the chat interface, booking appointments, and submitting prescriptions.

Secondary Users

Indirect users of the application.

- Admin: Overseeing high-level analytics and managing registered doctor schedules.
- Medicine Suppliers: sales analytics and order placements alerts.

Tertiary Users

Does not use the application at all, but is affected by it.

- Registered Doctors: External practitioners receiving appointments via the platform.

User Research Interview

These are the following pain points faced by the business:

- No digital inventory management system.
- Hand-written bills.
- Patients have problems reading the prescription.
- Patients confused about alternatives to the prescribed medicine.
- Centralized way to track specific medicine requests.
- No data analytics about sales and revenue.

- Re-stock reminders for staff members.
- Book appointments of registered doctors.

You can watch or listen to the following recording of Mr. Saha from Saha Pharmacy using the provided link. Upload the recording to [Team 65](#) drive.

SMART User Stories

User Management

Story: As a staff member, I want to login to the system, so that I can quickly create bills and search for inventory levels for products.

- Specific: Login authentication system for pharmacy.
- Measurable: Should take less than 2 minutes for user registration.
- Achievable: Viable within a one week sprint.
- Relevant: Only give access to staff members.
- Time-bound: Complete by the end of Sprint 2.

Inventory Management System

Story: As a staff user, I want to know the stock of a particular medicine, so that I can complete the order in under a minute.

- Specific: Query inventory database for stock information.
- Measurable: Order takes under a minute.
- Achievable: Feasible within a 2-week sprint.
- Relevant: Decreases order time and improves product tracking.
- Time-bound: Complete by the end of Sprint 4.

Payment System

Story: As a customer, I want to pay for my medication using digital wallet (UPI) so that I can complete my transaction securely and conveniently.

- Specific: Integrate a third-party payment gateway to process encrypted transactions.
- Measurable: Transaction success rate of 99.9% and processing time under 5 seconds.
- Achievable: Standard API integration feasible within a 2-week sprint.
- Relevant: Essential for revenue generation and user convenience.
- Time-bound: Fully functional by the end of Sprint 5.

Billing System

Story: As a cashier, I want the system to automatically generate a tax-compliant invoice so that I can provide the customer with a record of their purchase.

- Specific: Auto-calculate totals, GST/VAT, and discounts to generate a downloadable PDF.
- Measurable: Invoice generation takes less than 2 seconds after payment confirmation.
- Achievable: Uses existing database hooks; manageable for the backend team.
- Relevant: Ensures legal compliance and provides professional documentation.
- Time-bound: Deployment ready by the end of Sprint 6.

Prescription Reading System

Story: As a pharmacist, I want to upload a photo of a handwritten prescription so that the system can automatically extract the drug names and dosages.

- Specific: Implement an OCR using LLM to parse medical text from images.
- Measurable: Achieve at least 85% accuracy in text extraction.
- Achievable: High complexity; requires using a pre-trained ML model or API.
- Relevant: Drastically reduces manual data entry time and potential human error.
- Time-bound: Beta version completed by the end of Sprint 8.

Medicine Recommendation System

Story: As a pharmacist, I want the system to suggest generic alternatives for expensive brands so that I can offer more affordable or available options to patients.

- Specific: Create a logic-based engine that matches active ingredients (APIs) across the inventory.
- Measurable: Return at least two generic alternatives for any "Brand Name" query (if available).
- Achievable: Relies on a well-structured medicine database.
- Relevant: Improves customer satisfaction and increases inventory turnover.
- Time-bound: Feature to be live by the end of Sprint 7.

Data Analytics Dashboard

Story: As a pharmacy manager, I want to view a visual summary of monthly sales and top-selling medicines so that I can make informed restocking decisions.

- Specific: Develop a dashboard with bar charts and line graphs showing sales trends.
- Measurable: Dashboard must refresh with 100% accurate data upon page load.
- Achievable: Feasible using a frontend charting library (like Chart.js or D3).
- Relevant: Critical for strategic planning and financial oversight.
- Time-bound: Finalized and polished by the end of Sprint 9.

Stock Reminder System

Story: As a store owner, I want to receive an automated alert when any medicine falls below its minimum threshold so that I never run out of essential stock.

- Specific: Setup an automated email/SMS notification trigger based on real-time inventory counts.
- Measurable: Notification is sent within 10 minutes of the stock hitting the "Low" threshold.
- Achievable: Simple trigger logic within the existing database.
- Relevant: Prevents lost sales and ensures patients have access to life-saving meds.
- Time-bound: Implemented by the end of Sprint 5.

Appointment Booking System

Story: As a patient, I want to book a consultation with a pharmacist online so that I can avoid waiting in a physical queue.

- Specific: A calendar-based interface for selecting available 15-minute time slots.
- Measurable: The entire booking process must be completed in 4 clicks or fewer.
- Achievable: Standard CRUD operations; well within the team's capabilities.
- Relevant: Optimizes staff time and improves the overall patient experience.
- Time-bound: Ready for UAT (User Acceptance Testing) by the end of Sprint 10.