UE22CS341A: Software Engineering

Case Study

Unit 1 Deliverable

To adapt the SRS for an Inventory Management System (IMS) for Kirana stores using the Agile development model, the structure will focus on iterative development, continuous feedback, and evolving requirements. Agile emphasizes collaboration, customer feedback, and flexibility to changes, ensuring the final product is closely aligned with user needs.

Software Requirements Specification (SRS) Document for Inventory Management System (IMS) for Kirana Stores (Agile Model)

1. Introduction

1.1 Purpose

The purpose of this document is to provide an overview of the requirements for the Inventory Management System (IMS) tailored for Kirana stores, using the Agile development model. This system is intended to facilitate the efficient management of inventory, track stock levels, monitor expiry dates, manage sales, and provide analytics, all through a user-friendly web interface. The document outlines initial requirements, user stories, and the development approach to be followed in Agile sprints.

1.2 Scope

The Inventory Management System aims to replace traditional, manual inventory management practices in Kirana stores with a streamlined, automated system that integrates seamlessly into daily operations. The scope of the project includes managing product details, sales tracking, stock monitoring, expiry date alerts, supplier management, and generating reports. The system will evolve through multiple iterations based on feedback from store owners and users, ensuring it meets their evolving needs.

1.3 Definitions, Acronyms, and Abbreviations

- **IMS**: Inventory Management System
- **UI**: User Interface
- **API**: Application Programming Interface
- **SKU**: Stock Keeping Unit
- **Sprint**: A set period during which specific work must be completed and made ready for review

1.4 Overview

This document is structured into sections detailing the Agile approach, user stories, product backlog, and system features. It will be continuously updated as new requirements emerge and development progresses through iterative sprints.

2. Agile Development Approach

2.1 Agile Methodology

The Agile development process for the IMS will follow the Scrum framework, emphasizing collaboration, flexibility, and customer feedback. The project will be broken down into several sprints, each lasting 2-4 weeks. At the end of each sprint, a potentially shippable product increment will be reviewed by stakeholders to gather feedback and adjust the product backlog accordingly.

2.2 Sprint Planning

Each sprint will begin with a planning session where the team will define the goals, select user stories from the product backlog, and estimate the effort required to complete them. The outcome of the sprint planning is a sprint backlog containing all tasks to be completed during the sprint.

2.3 Daily Standups

Daily standup meetings will be held to ensure transparency and effective communication among team members. Each team member will report on what they accomplished the previous day, what they plan to accomplish today, and any impediments they are facing.

2.4 Sprint Review and Retrospective

At the end of each sprint, a sprint review will be conducted to demonstrate the completed work to stakeholders and gather feedback. This will be followed by a sprint retrospective to discuss what went well, what could be improved, and how to enhance productivity in future sprints.

3. Overall Description

3.1 Product Vision

The IMS for Kirana stores aims to provide an efficient, user-friendly solution for inventory management, enabling store owners to easily manage stock levels, track sales, monitor product expiry dates, and gain insights into store performance. The vision is to replace outdated manual systems with a modern digital solution that integrates seamlessly into the fast-paced environment of a Kirana store.

3.2 Key Features (Initial Backlog)

1. **Product Management**:
   - Create, update, delete, and view product records with details like name, category, price, stock levels, and expiry dates.
2. **Stock Monitoring**:
   - Track stock levels in real-time, update stock based on sales, and notify users when stock falls below a threshold.
3. **Sales Management**:
   - Record sales transactions, update inventory, and provide daily, weekly, and monthly sales analytics.
4. **Expiry Date Tracking**:
   - Monitor product expiry dates and provide alerts for soon-to-expire items.
5. **Supplier Management**:
   - Manage supplier information and automate ordering processes based on inventory levels.
6. **User Management**:
   - Role-based access control to restrict and manage user access to different parts of the system.
7. **Reporting and Analytics**:
   - Generate reports on sales, stock levels, and product turnover rates.

3.3 User Stories

- **As a store owner**, I want to add new products to the inventory, so that I can keep track of all items in the store.
- **As a cashier**, I want to scan products and process sales quickly, so that customer wait times are minimized.
- **As a store manager**, I want to receive alerts for low-stock items, so that I can reorder them before they run out.
- **As a supplier**, I want to receive automated purchase orders, so that I can fulfill store restocking needs efficiently.

3.4 Assumptions and Dependencies

- A stable internet connection is assumed for real-time data synchronization.
- Regular feedback from store owners and employees will be provided to improve the system continuously.
- The system depends on modern web technologies such as HTML, CSS, Next.js, SQL, and Auth0 for secure authentication.

4. System Features

4.1 Product Management

User Story:

"As a store owner, I want to manage product records, so that I can keep inventory data up-to-date."

Functional Requirements:

- The system shall allow the user to add new products with details like name, SKU, price, category, and expiry date.
- The system shall enable users to update or delete product details as needed.

4.2 Stock Monitoring

User Story:

"As a store manager, I want to monitor stock levels in real-time, so that I can restock products before they run out."

Functional Requirements:

- The system shall automatically update stock levels based on sales transactions.
- The system shall provide alerts when stock falls below a specified threshold.

4.3 Sales Management

User Story:

"As a cashier, I want to record sales transactions quickly, so that customer wait times are minimized."

Functional Requirements:

- The system shall record each sale transaction and update inventory in real-time.
- The system shall provide options to print receipts and view transaction history.

4.4 Expiry Date Tracking

User Story:

"As a store owner, I want to track product expiry dates, so that I can avoid selling expired goods."

Functional Requirements:

- The system shall monitor expiry dates for all perishable products.
- The system shall notify users of products nearing their expiry date.

4.5 Supplier Management

User Story:

"As a supplier, I want to receive automated purchase orders, so that I can fulfill restocking needs efficiently."

Functional Requirements:

- The system shall maintain a database of suppliers with contact details and supply history.
- The system shall automate purchase orders based on predefined rules and stock levels.

4.6 User Management

User Story:

"As an admin, I want to manage user roles and access, so that sensitive information is protected."

Functional Requirements:

- The system shall implement role-based access control.
- The system shall support secure user authentication using Auth0.

4.7 Reporting and Analytics

User Story:

"As a store owner, I want to see reports on sales and stock levels, so that I can make informed business decisions."

Functional Requirements:

- The system shall generate daily, weekly, and monthly reports on sales and inventory turnover.

- The system shall provide visual analytics to help identify sales trends and inventory needs.

5. Non-Functional Requirements

5.1 Performance Requirements

- The system shall respond to user inputs within 1 second.
- The system shall handle concurrent access by multiple users.

5.2 Security Requirements

- The system shall use HTTPS for secure data transmission.
- The system shall require user authentication and role-based access control.

5.3 Usability Requirements

- The system shall be intuitive and user-friendly, requiring minimal training.
- The system shall be accessible on both desktop and mobile devices.

5.4 Reliability Requirements

- The system shall have an uptime of 99.9%.
- The system shall include error-handling mechanisms for failed transactions.

5.5 Scalability Requirements

- The system shall be able to scale to support multiple stores under the same owner.

6. Backlog and Iteration Planning

6.1 Product Backlog

- Add/Edit/Delete Products
- Real-time Stock Monitoring
- Sales Transaction Module
- Expiry Date Alerts
- Supplier Database Management
- Automated Purchase Orders
- User Authentication and Role Management

- Report Generation and Analytics

6.2 Sprint Backlog

Sprint 1:
- Set up basic infrastructure and environment.
- Implement product management module.
- Develop user authentication using Auth0.

Sprint 2:
- Implement stock monitoring and expiry date tracking.
- Integrate sales transaction module.
- Develop basic reporting features.

Sprint 3:
- Enhance reporting and analytics features.
- Integrate supplier management and automated ordering.
- Improve UI/UX based on feedback from sprint reviews.

7. Conclusion

The Inventory Management System (IMS) for Kirana stores will be developed using the Agile model to ensure flexibility and responsiveness to user needs. By focusing on iterative development and continuous feedback, the system will evolve to provide a robust solution tailored to the dynamic environment of Kirana stores.

This Agile SRS document provides a comprehensive approach to developing the Inventory Management System for Kirana