**Mini Project Report on**



Inventory Management System [IMS]



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Divya 2019477**

**Arnav Dobriyal 2019466**

**Vinayak Pokhriyal 2019495**

***Under the Mentorship of***

**Mr. Jyotir Moy Chatterjee**

**Assistant Professor**

**A logo of a graphic era

Description automatically generated**

**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**June-2024**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction. | **3-4** |
| Chapter 2 | SRS. | **5-20** |
| Chapter 3 | Result. | **21-25** |
| Chapter 4 | Conclusion. | **26** |
| Chapter 5 | References. | **27** |

1. **Introduction**

Inventory Management System project, an innovative solution designed to streamline inventory processes and enhance efficiency for businesses of all sizes. In today's dynamic market, effective inventory management is crucial for ensuring smooth operations and maximizing profitability. Traditional inventory management methods are often time-consuming and prone to errors, leading to inefficiencies and lost revenue. Our project seeks to address these challenges by leveraging technology to develop a comprehensive inventory management system that automates key processes and provides real-time visibility into inventory levels.

The Inventory Management System project is a collaborative effort between our team and XYZ Corporation. The project aims to develop a user-friendly platform that allows businesses to track inventory levels, manage orders, and optimize stocking strategies. By integrating advanced features such as barcode scanning, automated alerts, and customizable reporting, our system will enable businesses to reduce stockouts, minimize excess inventory, and improve overall inventory control. With a focus on scalability and flexibility, our system will cater to the unique needs of each business, helping them achieve operational excellence and drive growth in today's competitive market.

* 1. **Motivation of Work**

The Inventory Management System project aims to revolutionize inventory processes by leveraging technology to streamline operations and enhance efficiency for businesses. By providing a user-friendly platform that automates key processes, tracks inventory levels, manages orders, and optimizes stocking strategies, the system helps businesses reduce stockouts, minimize excess inventory, and improve overall inventory control. With features such as barcode scanning, automated alerts, and customizable reporting, the system not only enhances operational efficiency but also improves customer satisfaction by ensuring products are consistently in stock and orders are fulfilled promptly. Additionally, by embracing digital inventory management solutions, businesses can gain a competitive advantage, align with market trends, and future-proof their operations in today's dynamic market.

* 1. **Objective of the project**

The primary objective of the Bank Account Management System project is to address the challenges and develop a modern, secure, and user-friendly online banking platform that delivers a superior customer experience while improving operational efficiency for banks.

The Bank Account Management System project will involve the following key activities:

* Collaborate with stakeholders, including business owners and inventory managers, to gather and document requirements for the inventory management system.
* Design the architecture, user interface, and functionalities of the system based on the gathered requirements and industry best practices.
* Develop the inventory management system using modern technologies and programming languages, ensuring scalability, security, and usability.
* Conduct rigorous testing to identify and fix any bugs or issues in the system, ensuring that it meets quality standards and performs reliably under various conditions.
* Deploy the inventory management system in a production environment and provide training to users on how to effectively use the system.
* Provide ongoing maintenance and support services to ensure the smooth operation of the inventory management system and address any issues or enhancements as needed.
  1. **Summary**

In summary, the Inventory Management System project offers businesses a strategic opportunity to enhance operational efficiency, improve customer satisfaction, and gain a competitive edge in today's dynamic market. By automating key inventory processes and providing real-time visibility into inventory levels, the system helps businesses reduce costs, minimize stockouts, and optimize stocking strategies. Investing in this system enables businesses to streamline operations, meet customer demands, and position themselves for long-term success and growth in an increasingly digital and competitive business landscape

1. .**Software Requirement Specification for Inventory Management System**

# 2. Introduction

## 2.1 Purpose

The purpose of this document is to provide a comprehensive overview of the inventory management software, outlining its functionalities for tracking inventory, replenishing stock, optimizing warehouse layouts, detailing product quality, monitoring expiration dates, and managing authorization levels for both administrators and customers.

## 2.2 Document Conventions

This document adheres to the IEEE template for System Requirement Specification Documents.

## 2.3 Intended Audience and Reading Suggestions

* Retailers and Business Owners: Those involved in using inventory and warehouse operations as a customer.
* Warehouse Managers and Staff: Individuals responsible for day-to-day inventory management tasks.
* Software Developers and Engineers: Those interested in further developing or customizing the inventory management system.

## 2.4 Product Scope

The inventory management software is designed to streamline inventory tracking, stock replenishment, warehouse layout optimization, product quality management, expiration date monitoring, and user authorization. Users can efficiently organize and manage their inventory data, analyze trends, and ensure product quality and compliance.

## 2.5 References

1. [www.mysql.com](http://www.mysql.com/)
2. [www.php.com](http://www.php.com/)
3. [www.apache.org](http://www.apache.org/)
4. [www.stackoverflow.com](http://www.stackoverflow.com/)

# 3. Overall Description

## 3.1 Product Perspective

The inventory management system operates as a standalone application designed to streamline inventory-related tasks for businesses, retailers, and warehouse managers. It interfaces with a database to store and retrieve inventory data, user information, and system configurations. While it primarily functions independently, it can integrate with other systems such as accounting software for seamless business operations.

## 3.2 Product Functions

Authentication and Authorization: Allows users to authenticate with the system and grants appropriate permissions based on their role (admin or customer).

Inventory Management: Enables users to add, edit, view, and delete inventory items, including details such as name, expiry date, quantity, size, and type.

User Management: Admin users can manage customer details, including name, phone number, and email, and delete customers from the system if necessary.

Stock Replenishment: Allows both owners and customers to replenish stock by adding new items to the inventory.

Expiration Monitoring: Provides functionalities to monitor expiration dates of inventory items, notifying users of approaching expirations.

User Interface: Offers an intuitive interface for easy navigation and utilization of system functionalities.

## 3.3 User Classes and Characteristics

Admin Users: Typically business owners or warehouse managers who have full access to system functionalities. They are responsible for managing inventory, user accounts, and overall system configurations.

Customer Users: Individuals or employees who interact with the system to monitor and manage inventory stock levels. They have limited access compared to admin users, primarily focusing on viewing and replenishing inventory.

## 3.4 Operating Environment

The system is designed to operate in various environments, including desktop computers, laptops, and servers. It is compatible with commonly used operating systems such as Windows, macOS, and Linux. Users can access the system through modern web browsers, ensuring flexibility and accessibility across different devices.

## 3.5 Design and Implementation Constraints

Hardware Requirements: The system requires a minimum hardware configuration of a 500 MHz CPU, 128 MB RAM, and an OpenGL 1.2 compatible graphics card. Users handling larger inventories may require hardware upgrades for optimal performance.

Database Compatibility: The system relies on a compatible database management system (DBMS) for storing and retrieving data. It should be compatible with popular DBMS such as MySQL, PostgreSQL, or SQLite.

Web Framework Dependency: The system is built using the FastAPI framework, which imposes certain constraints on development and deployment. Developers need to adhere to the framework's conventions and best practices during implementation.

## 3.6 User Documentation

Comprehensive user documentation accompanies the system, providing detailed instructions on installation, configuration, and usage. It includes tutorials, FAQs, troubleshooting guides, and system requirements to assist users in effectively utilizing the system's functionalities.

## 3.7 Assumptions and Dependencies

Internet Connectivity: The system assumes users have access to stable internet connectivity to interact with the web-based interface and communicate with the backend server.

Browser Compatibility: Users are expected to use modern web browsers compatible with HTML5, CSS3, and JavaScript for optimal performance and functionality.

Database Connectivity: The system relies on a functioning database server accessible to the application for storing and retrieving data. Proper database configurations and connectivity are assumed for seamless operation.

User Training: Users are assumed to receive basic training on system usage and functionalities to maximize productivity and minimize errors during operation. Training materials and support channels are provided as part of user documentation.

**3.7 Methodology Used**

Prototype modeling for an inventory management system involves creating a preliminary version of the system to visualize its features and functionalities. This prototype helps in identifying and addressing potential issues early, gathering user feedback, and refining system requirements. It allows stakeholders to interact with the system, providing insights into usability and workflow efficiency. By iterating on the prototype based on feedback, developers can enhance the design and functionality, ensuring the final system effectively manages inventory levels, tracks stock movements, and provides accurate reporting. Ultimately, this approach leads to a more user-friendly and efficient inventory management solution.

# 4. External Interface Requirements

## 4.1 User Interfaces

Login Page: Provides a form for users to enter their credentials (username and password) for authentication.

Signup Page: Allows new users to create an account by providing necessary information such as name, phone number, email, and password.

Dashboard: Upon successful authentication, users are presented with a dashboard tailored to their role (admin or customer). It provides access to various functionalities and displays relevant information such as inventory details or customer information.

Inventory Management Interface: Enables users to add, edit, view, and delete inventory items. It includes forms for inputting item details and tables for displaying existing inventory items.

Customer Management Interface: Available only to admin users, it facilitates the management of customer details, including name, phone number, and email. Admins can view customer information and delete customers if necessary.

Replenishment Interfaces: Separate interfaces are provided for owners and customers to replenish stock by adding new items to the inventory.

Expiration Monitoring Interface: Displays expired items for both owners and customers, notifying users of items that require attention.

## 4.2 Hardware Interfaces

CPU: The system requires a minimum of a 500 MHz CPU to ensure smooth operation.

RAM: A minimum of 128 MB RAM is necessary to handle system processes and data storage.

Graphics Card: An OpenGL 1.2 compatible graphics card is recommended for optimal rendering of graphical elements in the user interface.

## 4.3 Software Interfaces

Database Management System (DBMS): The system interfaces with a compatible DBMS for storing and retrieving inventory data, user information, and system configurations. Popular options include MySQL, PostgreSQL, or SQLite.

Web Framework: Built using the FastAPI framework, the system interfaces with FastAPI's modules and libraries for handling HTTP requests, routing, and template rendering.

Frontend Technologies: Utilizes HTML, CSS, and JavaScript for building user interfaces and enhancing user interactions in the web-based interface.

External Libraries: May incorporate external libraries or modules for additional functionalities such as authentication, data validation, or visualization.

## 4.4 Communications Interfaces

HTTP Protocol: Communication between the client (user's web browser) and the server (FastAPI application) occurs over the HTTP protocol. HTTP requests and responses facilitate data exchange and interaction with the system.

RESTful API: The system may implement a RESTful API for communicating with external services or integrating with other systems. API endpoints allow external applications to interact with inventory data and system functionalities.

WebSocket (Optional): WebSocket communication may be implemented for real-time updates or notifications, such as expiration alerts or inventory changes. This facilitates seamless communication between the client and server for dynamic updates.

# 5. System Features

Authentication and Authorization:

Allows users to securely log in to the system using their username and password.

Implements role-based access control (RBAC) to grant appropriate permissions based on user roles (admin or customer).

Inventory Management:

Enables users to add, edit, view, and delete inventory items.

Captures item details such as name, expiry date, quantity, size, and type for effective inventory tracking.

Provides functionalities for owners and customers to manage their respective inventories.

User Management:

Admin users can manage customer details, including name, phone number, and email.

Facilitates adding new customers and deleting existing customer accounts if necessary.

Stock Replenishment:

Allows owners and customers to replenish stock by adding new items to the inventory.

Ensures seamless addition of items to maintain optimal stock levels and prevent shortages.

Expiration Monitoring:

Monitors expiration dates of inventory items to prevent stockouts and minimize wastage.

Notifies users of approaching expirations, enabling timely actions such as restocking or disposal.

User Interface:

Provides an intuitive and user-friendly interface for easy navigation and interaction.

Offers clear visual representations of inventory data and system functionalities to enhance usability.

Ensures consistent user experience across different devices and screen sizes.

Error Handling and Reporting:

Implements robust error handling mechanisms to gracefully handle exceptions and unexpected scenarios.

Provides informative error messages to guide users in resolving issues and performing corrective actions.

Enables users to report encountered bugs or issues for prompt resolution by the development team.

Dashboard:

Upon login, presents users with a personalized dashboard tailored to their role (admin or customer).

Displays relevant information such as inventory details, customer information, and system notifications.

Offers quick access to commonly used functionalities and tasks, enhancing user productivity.

Customer-Specific Functionalities:

Provides customer-specific functionalities such as viewing their inventory, replenishing stock, and managing expiration dates.

Ensures customers have access only to relevant features based on their role and permissions.

Owner-Specific Functionalities:

Offers additional functionalities for owners, including viewing total inventory, managing customer details, and monitoring expiration dates.

Empowers owners with comprehensive tools for efficient inventory management and business operations.

These system features collectively enable efficient inventory management, user authentication, and seamless interaction with the system, catering to the diverse needs of administrators, customers, and other stakeholders.

# 6. Other Nonfunctional Requirements

## 6.1 Performance Requirements

The inventory management software is optimized to function seamlessly across different systems, ensuring efficient inventory management without performance hiccups. It requires a minimum of a 500 MHz CPU, 128 MB RAM, and an OpenGL 1.2 compatible graphics card to support essential tasks such as tracking inventory, replenishing stock, optimizing warehouse layouts, managing product quality, monitoring expiration dates, and authorizing users. However, performance may fluctuate depending on database size and task complexity. Users handling larger inventories or complex operations might need hardware upgrades for better performance.

## 6.2 Safety Requirements

To safeguard inventory data, the software undergoes regular updates and employs a comprehensive bug tracking system. Users are encouraged to report any encountered bugs or issues, which the development team addresses in subsequent updates. This proactive bug management approach minimizes the risk of data loss and ensures software reliability and security.

## 6.3 Security Requirements

The inventory management software ensures security through role-based access control (RBAC), which regulates user permissions and limits access to authorized individuals. Users are granted specific authorization levels to carry out tasks like managing permissions and accessing detailed product quality information.

## 6.4 Software Quality Attributes

The inventory management software prioritizes user-friendliness, catering to various user groups, including retailers, business owners, warehouse managers, staff, software developers, and engineers. It offers a range of features from basic inventory tracking to advanced warehouse layout optimization, ensuring usability regardless of technical expertise. The software boasts an intuitive interface for easy navigation and utilization of functionalities. However, users are advised to familiarize themselves with inventory management basics and software features to maximize its capabilities.

## 6.5 Business Rules

Business rules outline the operational guidelines and constraints that govern the behavior and functionality of the Inventory Management System. These rules define specific actions, permissions, and restrictions based on various inventory-related scenarios and user roles. The following business rules are specified:

Authentication and Authorization Rules: Users must authenticate with valid credentials (username and password) to access the system. Role-based access control (RBAC) governs user permissions, granting access levels based on roles such as admin or customer. Admin users have full access to system functionalities, including inventory management and user management. Customer users have limited access, primarily to view their inventory, replenish stock, and view expiration dates. Deleting items or customers requires appropriate authorization levels.

Inventory Management Rules: Users can add items to the inventory, specifying details such as name, expiry date, size, quantity, perishable status, and type. Items can be replenished by both owners and customers, with different interfaces for each. Inventory items can be displayed, edited, and deleted by authorized users. Owners have additional functionalities such as viewing total inventory, managing customer details, and monitoring expiration dates. Customers can only view their own inventory items and replenish stock.

Data Management Rules: Inventory data must be stored securely in the database. Customer information, including name, phone number, email, and password, is collected during signup and stored securely. Items in the inventory are associated with the user who added them.

Error Handling Rules: Proper error messages should be displayed for invalid login attempts, failed operations, or system errors. Error handling mechanisms should prevent unauthorized access or data manipulation.

User Interface Rules: The user interface should be intuitive and user-friendly, guiding users through various functionalities. Clear navigation paths should be provided for users to access different sections of the system. Forms should include validation to ensure correct input and prevent errors. Relevant information, such as inventory details or customer information, should be presented clearly and logically.

# 7. Other Requirements

Appendix A: Glossary

This glossary provides definitions for terms, acronyms, and abbreviations used throughout the Software Requirements Specification (SRS) document to ensure clarity and consistency in interpretation:

SRS: Software Requirements Specification - A document that specifies the functional and non-functional requirements of a software system.

DBMS: Database Management System - A software system used to manage and organize databases.

API: Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other.

UI: User Interface - The graphical or textual interface through which users interact with a software system

RBAC: Role-Based Access Control - A security model that restricts system access based on user roles and permissions.

OTP: One-Time Password - A unique password that is valid for only one login session or transaction.

TLS/SSL: Transport Layer Security/Secure Sockets Layer - Protocols that provide secure communication over a computer network

GDPR: General Data Protection Regulation - A regulation in EU law on data protection and privacy.

PCI DSS: Payment Card Industry Data Security Standard - A set of security standards designed to ensure that all companies that accept, process, store, or transmit credit card information maintain a secure environment.

ISO 27001: International Organization for Standardization 27001 - A standard for information security management systems

SOC 2: Service Organization Control 2 - A framework for managing customer data based on five "trust service principles": security, availability, processing integrity, confidentiality, and privacy.

HIPAA: Health Insurance Portability and Accountability Act - A regulation that provides data privacy and security provisions for safeguarding medical information.

API: Application Programming Interface - A set of rules and protocols that allows different software applications to communicate with each other.

TBD: To Be Determined - Indicates that a specific detail or requirement is yet to be finalized.

UI: User Interface - The graphical or textual interface through which users interact with a software system

GUI: Graphical User Interface - A type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators

API: Application Programming Interface - A set of protocols, tools, and definitions that allows different software applications to communicate with each other

UX: User Experience - The overall experience of a user when interacting with a product or system, including ease of use, efficiency, and satisfaction.

HTML: Hypertext Markup Language - The standard markup language for creating web pages and web applications.

CSS: Cascading Style Sheets - A style sheet language used for describing the presentation of a document written in HTML or XML.

Appendix B: Analysis Models

1. Use Case Diagram: -

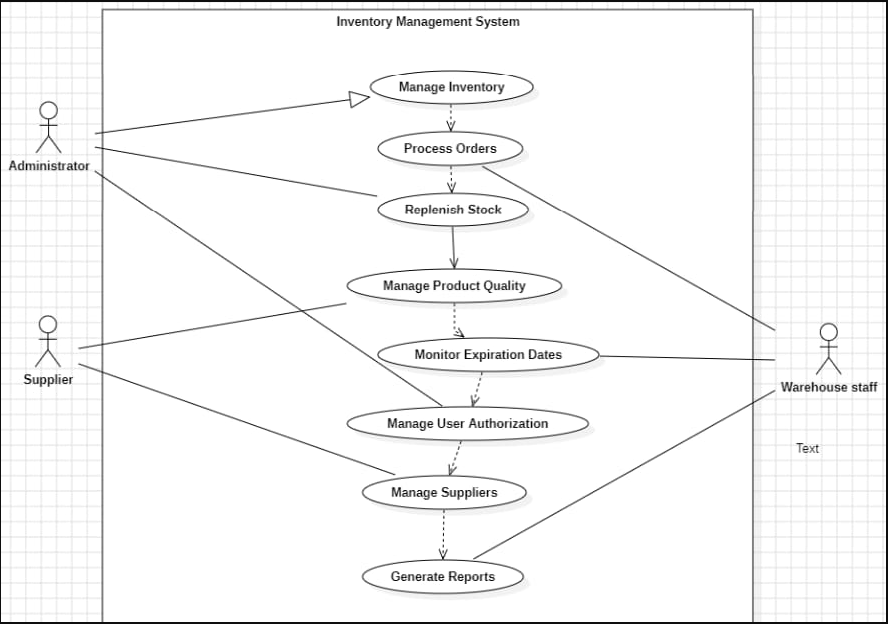
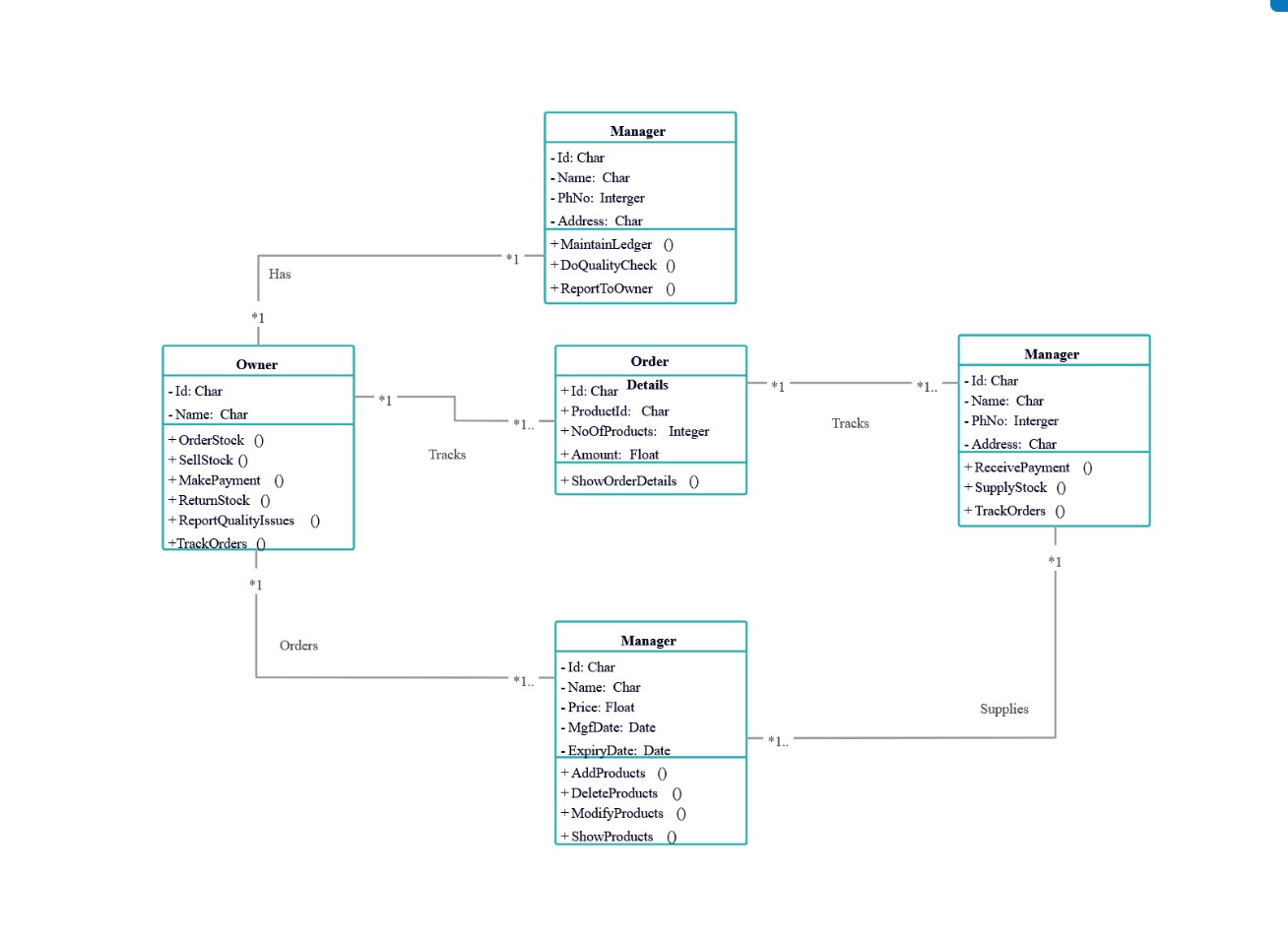


Fig 1: Use case Diagram for IMS

A use case diagram for a Inventory Management System illustrates interactions between users and the system, depicting actions like creating an account, generating reports , managing supliers. It provides a high-level overview of how users interact with the system's functionalities to manage their inventory efficiently.

2. Class Diagram: -

Fig [2]: Class Diagram for IMS

The class diagram for a Inventory Management System outlines the system's structure, depicting classes such as Manager, Owner, Order, and along with their attributes and relationships. It illustrates how these classes interact to manage inventory, handle tracks, and maintain user and administrative functionalities within the system.

1. Sequence Diagram: -

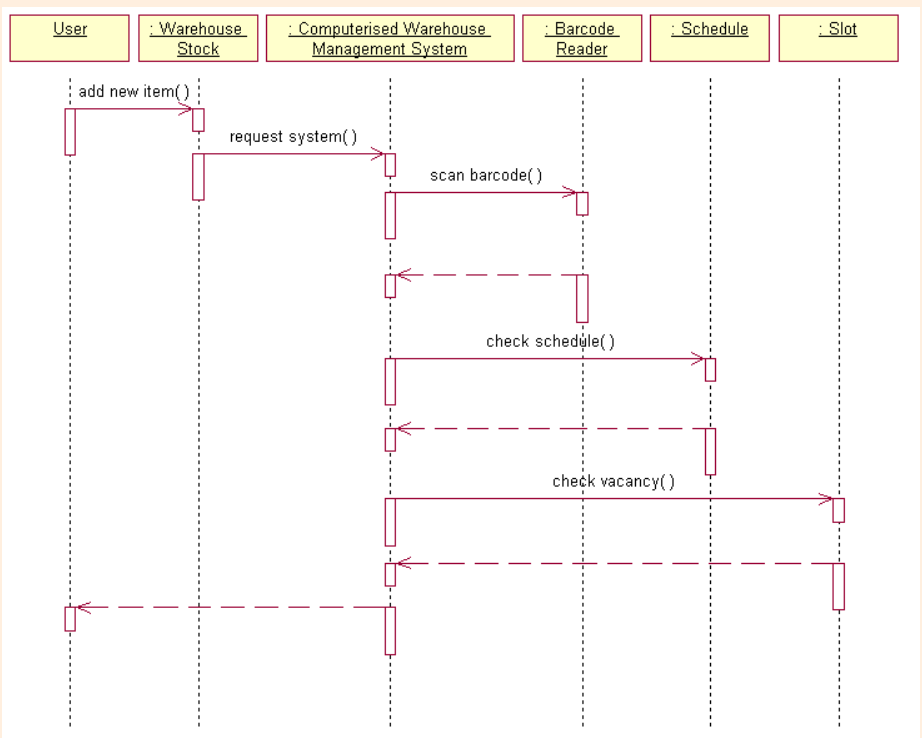


Fig [3]: Sequence Diagram for IMS

The sequence diagram for the Inventory Management System visualizes the chronological order of interactions between actors and the system, illustrating steps like check barcode, transaction initiation, and database updates. It demonstrates the flow of control and data between various components, showcasing how the system processes user requests and updates inventory information accordingly

1. Entity Relationship Diagram: -

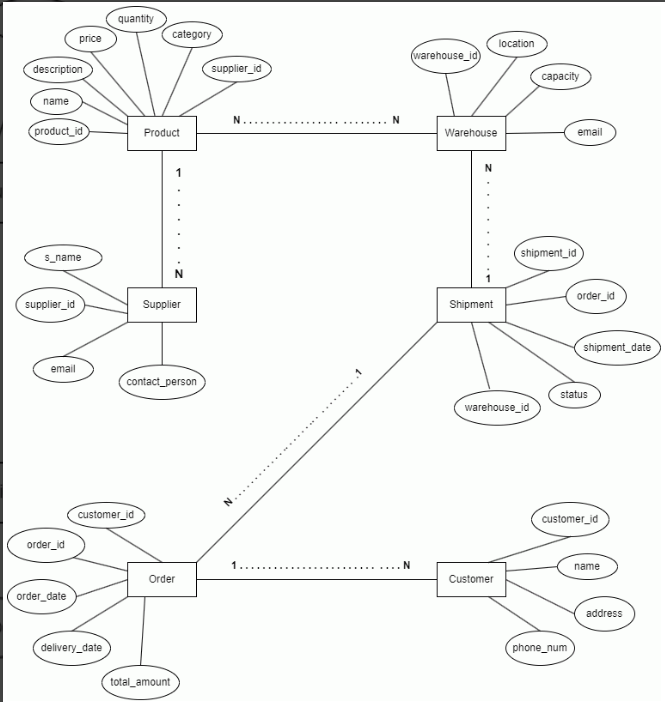


Fig [4]: E-R diagram for IMS

**3. RESULT**

**3.1 Sample Code: -**

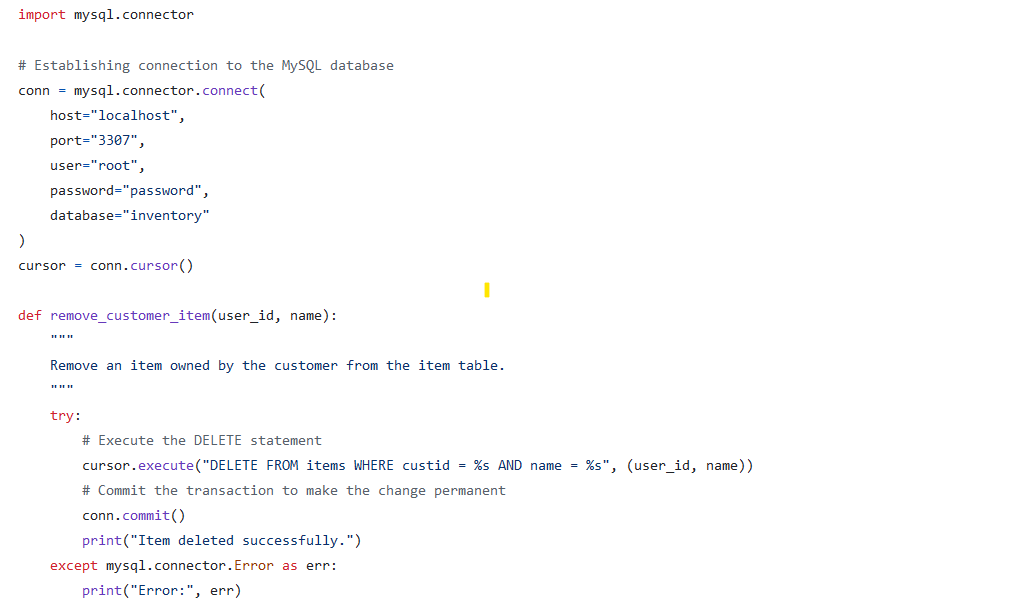
1. **Fastapi Inventory Management Code:**

****

**Fig [3.1]**

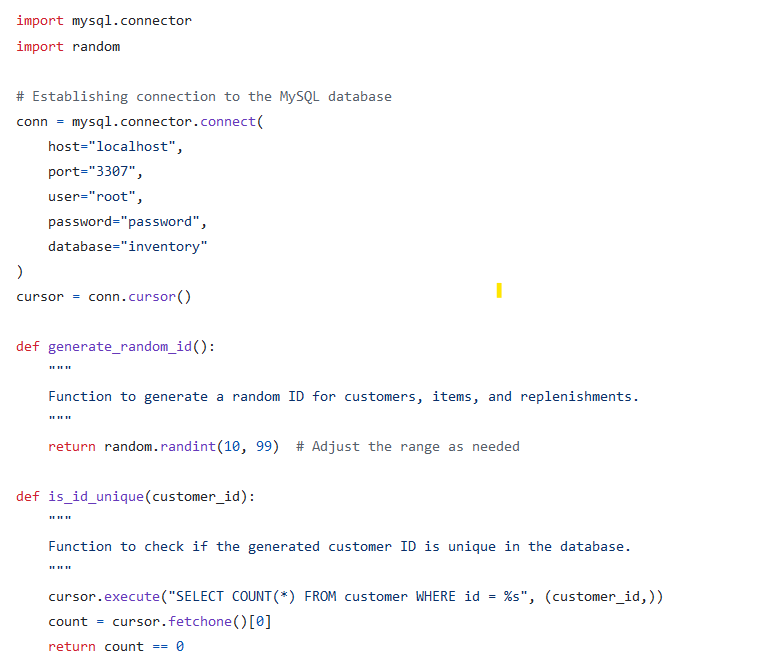
This FastAPI application is a simple inventory management system that allows users to sign up, log in, and manage inventory items. Users can add, view, and delete items, view customer details, and manage replenishments and expired items. The application uses FastAPI for the web framework, Jinja2Templates for rendering HTML templates, and SQLite for the database. It also includes session management to track the logged-in user. The code follows a modular structure with separate functions for database operations, authentication, and route handling. Overall, this application provides a foundation for a functional inventory management system that can be further expanded and customized for specific business needs.

1. **Deletion Code:**

****

**Fig [3.3]**

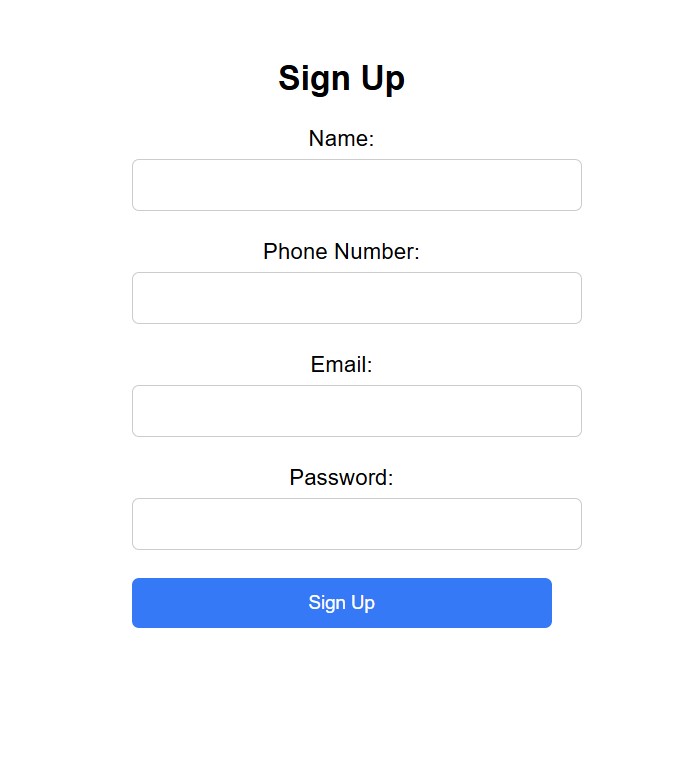
1. **Insertion Dashboard:**



**Fig [3.4]**

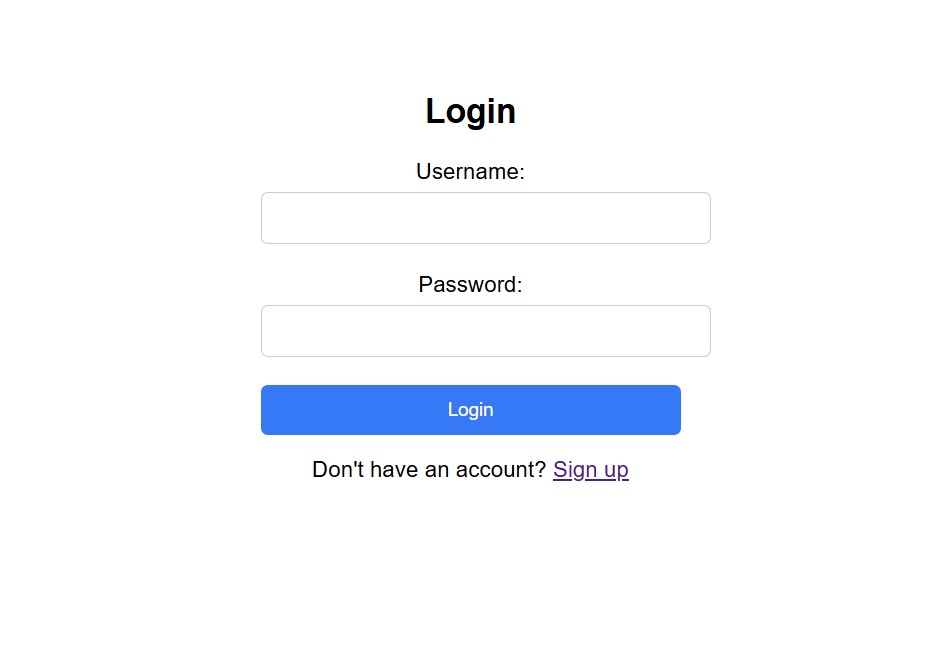
**3.2 Sample Form: -**

1. **Sign up form :**



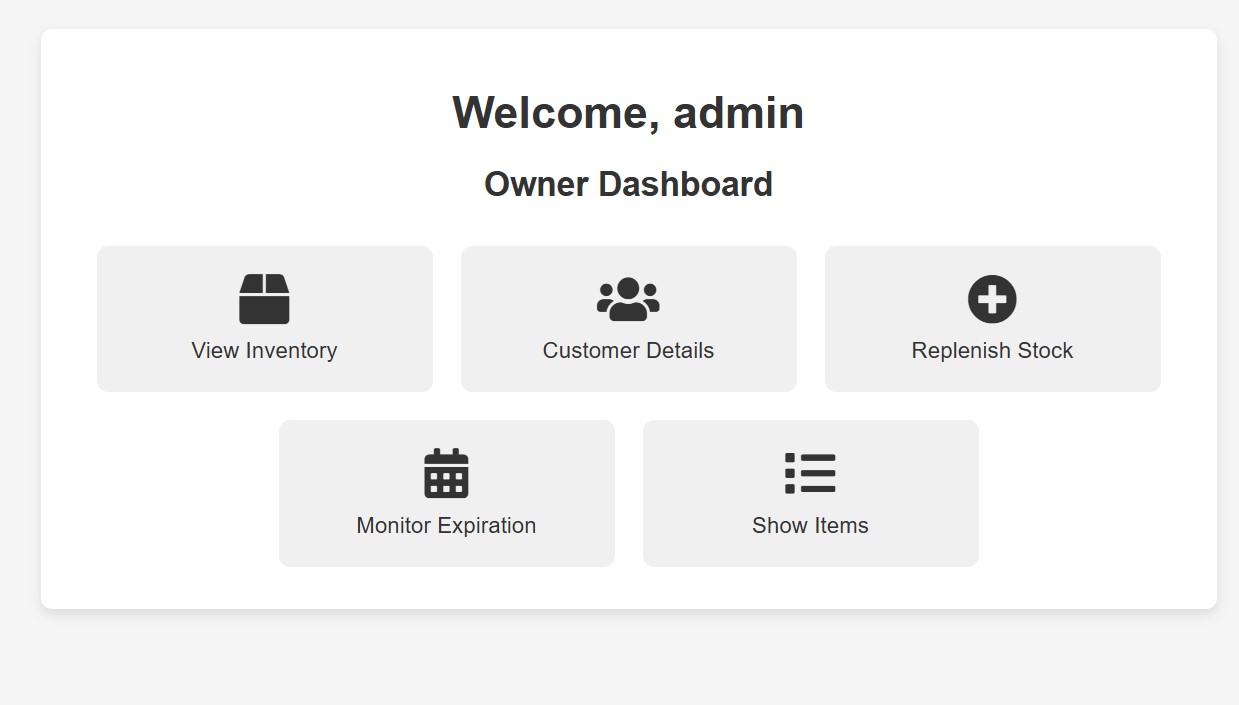
**Fig [3.5]**

1. **Log in form :**

**­­­­­**

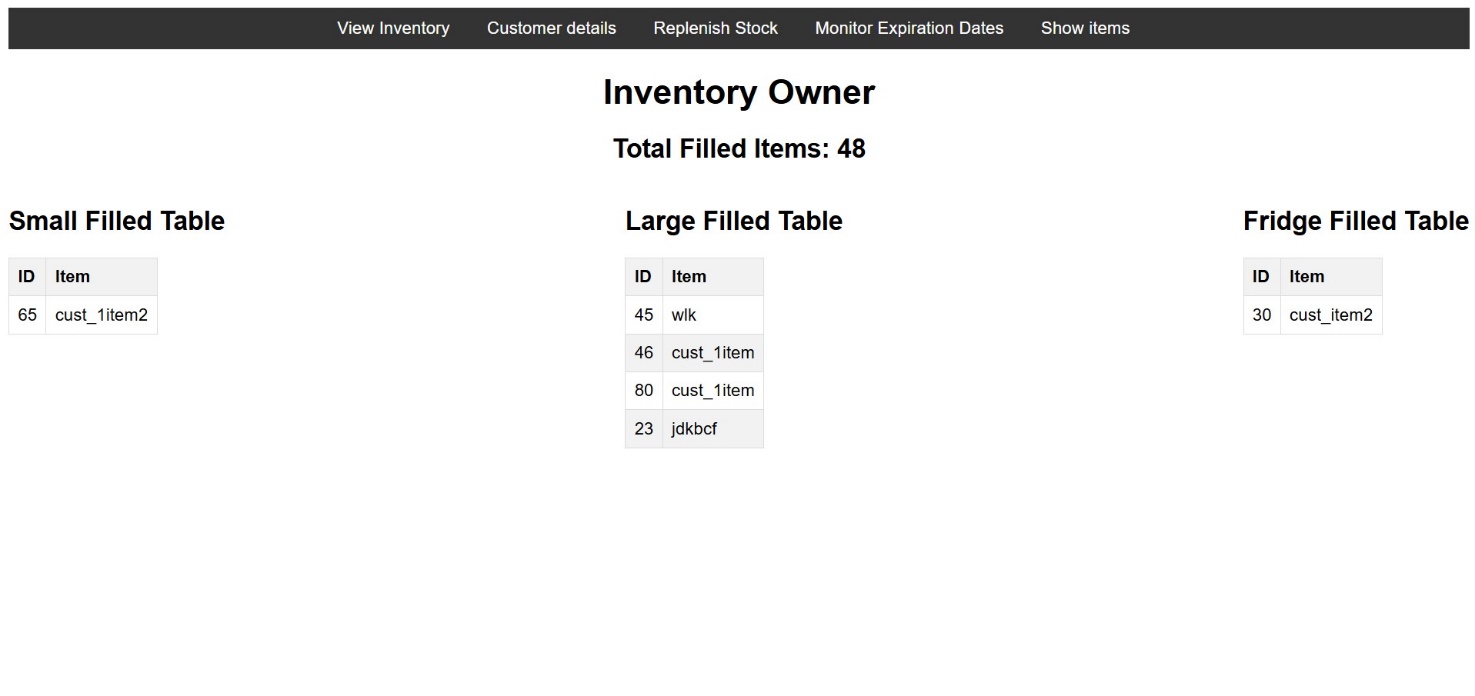
**Fig [3.6]**

1. **Owner Dashboard:**



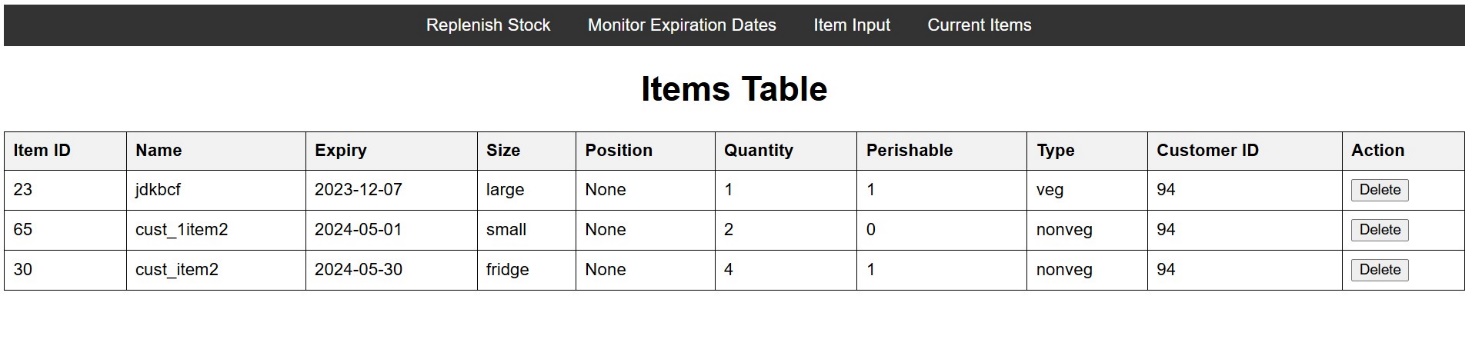
**Fig [3.7]**

1. **Inventory \_Owner :**



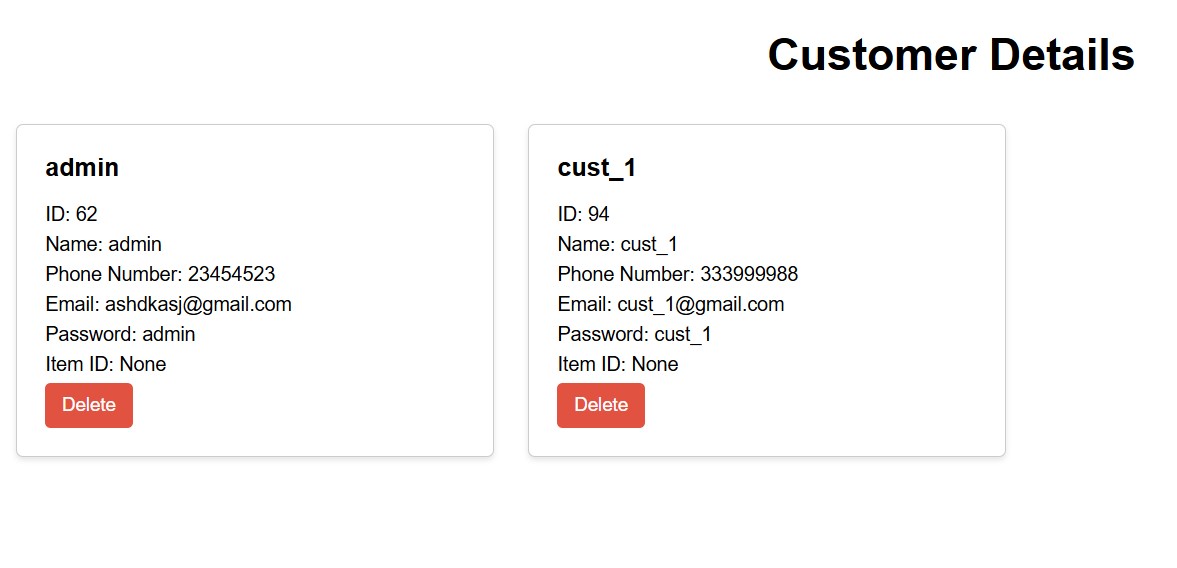
**Fig[3.8]**

1. **Item Table :**



**Fig [3.9]**

1. **Customer Details :**



**Fig [3.10]**

**4.Conclusion**

The System Requirements Specification (SRS) for a Inventory Management System is a critical document that outlines the functional and non-functional requirements of the software system. This document serves as a foundation for the design, development, and testing phases of the project, ensuring that the final system meets the needs and expectations of both the bank and its customers. In this conclusion, we will highlight the key takeaways and emphasize the importance of a well-defined SRS for such a complex and sensitive domain. First and foremost, the SRS for a Inventory Management System has been designed with a strong emphasis on security and data integrity. The confidentiality, availability, and integrity of customer data are of paramount importance in the banking industry. This document provides a detailed outline of the security measures, user access controls, and encryption standards that will be implemented to safeguard sensitive information. By doing so, it ensures that customer trust is maintained and regulatory compliance is upheld. Another crucial aspect of the SRS is its comprehensive coverage of the system's functionality. It defines the various features and modules that the Banking Management System will offer, including customer account management, transaction processing, loan management, reporting, and administrative tools. This level of detail enables developers to understand the full scope of the project and minimizes misunderstandings or misinterpretations during the development process.

In conclusion, the System Requirements Specification for a Inventory Management is an indispensable document that lays the foundation for the successful development and deployment of a sophisticated and secure banking software solution. It encompasses the critical aspects of functionality, security, performance, and user experience, all of which are essential in the inventory industry. A well-defined SRS reduces ambiguity, minimizes risks, and serves as a roadmap for developers and stakeholders. By following the guidelines and requirements set forth in this document, the bank can look forward to a robust, secure, and customer-centric Inventory Management System that will not only meet their immediate needs but also adapt to future challenges and opportunities in the ever-evolving world of banking.

**5.Reference**

1. **Websites :**
2. <https://www.geeksforgeeks.org/>
3. <https://www.researchgate.net/>
4. <https://www.slideshare.net/>
5. <https://www.studocu.com/>
6. <https://en.wikipedia.org/>
7. **Tutorials**
8. MySQL video tutorials

<https://www.youtube.com/watch?v=DxoRUmW44JE>

1. Python tutorials