DHA Suffa University

Department of Computer Science

Final Year Project



**Automatic Inventory Management System**

**S29F24**

**Software Requirements Specifications**

Submitted by

**Team Lead: Rimsha younus(cs211237)**

**Team Member: Summaya Abid (cs211240)**

**Team Member: Khunsha Noman (cs211248)**

**Team Member: Muhammad Mohsin (cs211202)**

Supervisor(s)

Co-Supervisor Name (Miss Raazia Sosan)

Supervisor Name (Sir Conrad)

**Fall 2024**

**Document Sign off Sheet**

### **Document Information**

|  |  |
| --- | --- |
| **Project Title** | Automatic Inventory Management System |
| **Project Code** | S29F24 |
| **Document Name** | Software Requirements Specifications |
| **Document Version** | <1.0> |
| **Document Identifier** | FYP01- S29F24 |
| **Document Status** | Initial draft |
| **Author(s)** | Rimsha, Summaya, Khunsha, Mohsin |
| **Approver(s)** | <Conrad D silva> |
| **Issue Date** | <Date of issuance of this document> |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 13/1/2025 | 1.0 | initial document | Rimsha, Summaya, KhMohsin, Khunsha |
|  |  |  | Khunsha, Mohsin |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Signature** | **Date** |
| Rimsha Younus | Team Lead |  |  |
| Summaya Abid | Team Member 2 |  |  |
| Khunsha Noman | Team Member 3 |  |  |
| M.Mohsin | Team Member 4 |  |  |
| Conrad D Silva | Supervisor |  |  |
| Miss Raazia | Co-Supervisor |  |  |
| Sir Saad | Project Coordinator |  |  |

**Definition of Terms, Acronyms, and Abbreviations**

*[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.]*

|  |  |
| --- | --- |
| **Term** | **Description** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **Table of Contents**

[**1.** Introduction 8](#_Toc4677804)

[1.1 Purpose of Document 8](#_Toc4677805)

[1.2 Intended Audience 8](#_Toc4677806)

[1.3 Abbreviations 8](#_Toc4677807)

[**2.** Overall System Description 9](#_Toc4677808)

[2.1 Project Background 9](#_Toc4677809)

[2.2 Problem Statement 9](#_Toc4677810)

[2.3 Project Scope 9](#_Toc4677811)

[2.4 Not In Scope 9](#_Toc4677812)

[2.5 Project Objectives 9](#_Toc4677813)

[2.6 Stakeholders & Affected Groups 9](#_Toc4677814)

[2.7 Operating Environment 9](#_Toc4677815)

[2.8 System Constraints 9](#_Toc4677816)

[2.9 Assumptions & Dependencies 9](#_Toc4677817)

[**3.** External Interface Requirements 10](#_Toc4677818)

[3.1 Hardware Interfaces 10](#_Toc4677819)

[3.2 Software Interfaces 10](#_Toc4677820)

[3.3 Communications Interfaces 10](#_Toc4677821)

[**4.** System Functions / Functional Requirements 11](#_Toc4677822)

[4.1 System Functions 11](#_Toc4677823)

[4.2 Use Cases 12](#_Toc4677824)

[4.2.1 List of Actors 12](#_Toc4677825)

[4.2.2 List of Use Cases 12](#_Toc4677826)

[4.2.3 Use Case Diagram 13](#_Toc4677827)

[4.2.4 Description of Use Cases 13](#_Toc4677828)

[**5.** Non - Functional Requirements 16](#_Toc4677829)

[5.1 Performance Requirements 16](#_Toc4677830)

[5.2 Safety Requirements 16](#_Toc4677831)

[5.3 Security Requirements 16](#_Toc4677832)

[5.4 Reliability Requirements 16](#_Toc4677833)

[5.5 Usability Requirements 16](#_Toc4677834)

[5.6 Supportability Requirements 16](#_Toc4677835)

[5.7 User Documentation 16](#_Toc4677836)

[**6.** References 17](#_Toc4677837)

# **Introduction**

.

## **Purpose of Document**

## The purpose of this document is to provide a comprehensive and detailed analysis of the **Inventory Management System** (IMS), outlining its core features, functionalities, and system requirements. This document is aimed at multiple stakeholders, including developers, users, project managers, and system administrators, to offer a clear understanding of the system's design and objectives

## **Intended Audience**

It is primarily intended for:

* **Jury Members**: To understand the objectives, structure, and design of the project.
* **Project Managers & Developers**: To follow the development process and guidelines.
* **End Users (SMB Owners/Managers)**: To understand the system’s features and how it will benefit their business.

The purpose of this project is to create an efficient Inventory Management System that simplifies managing sales, purchases, products, categories, and reports while generating invoices and bills.

# Overall System Description

**2.1 Project Background**

In today’s fast-paced business environment, small and medium-sized businesses (SMBs) often struggle with managing their inventory, sales, billing, and reporting processes efficiently. Many businesses still rely on outdated, manual methods for tracking inventory and managing orders, which can lead to errors, inefficiencies, and a lack of real-time insights.

The **Automatic Inventory Management System (AIMS)** aims to address these challenges by offering a fully integrated platform that streamlines key business operations, such as inventory tracking, sales management, billing, and reporting. The system uses artificial intelligence (AI) to enhance decision-making, automate repetitive tasks, and provide actionable insights, enabling businesses to make data-driven decisions.

The platform is designed to be highly customizable and scalable, catering to the unique needs of SMBs. AIMS offers a suite of modules, including Sales, Billing, Inventory, and Reporting, each tailored to improve operational efficiency, reduce manual errors, and promote growth. With AI-powered suggestions, businesses can optimize inventory levels, predict demand trends, and automate financial reporting, ultimately enhancing productivity and profitability.

Managing inventory manually can lead to inefficiencies, errors, and delays. This project aims to automate the inventory management process, offering streamlined operations and robust reporting.

**2.2 Problem Statement**

 **Slow and Error-prone**: Manual inventory management takes time and is prone to mistakes, causing incorrect stock records and delays.

 **No Instant Data**: Businesses have trouble getting real-time information, making it hard to manage stock and make quick decisions.

 **Slow Order Processing**: Inefficient inventory management causes delays in processing orders, affecting customer satisfaction and business performance.

.

**2.3 Project Scope**

### **In Scope:**

The **Automatic Inventory Management System (AIMS)** will include the following key features to help businesses manage their inventory effectively:

* **Inventory Management**: Track stock levels, manage product orders, and update inventory in real-time.
* **Sales and Purchase Management**: Record sales and purchase transactions, with automatic inventory updates.
* **Reports**: Generate detailed reports on inventory, sales, and purchase history.
* **AI-Powered Suggestions**: Use AI to analyze past data and give recommendations to optimize inventory, predict trends, and improve decision-making.
* **Chatbot**: Chatbot integration to provide answers to users queries and questions
* **Alerts**: Receive notifications when stock levels are low to avoid running out of products or overstocking etc
* **User Management**: Allow different user roles (admin, manager, staff) to access features based on their responsibilities.

### **Out of Scope:**

The following features will not be included in **AIMS**:

* **E-commerce Functionality**: The system will not support online store features like product catalogs, shopping carts, or payment processing.
* **Advanced Accounting Tools**: AIMS will not offer detailed financial tools such as tax calculations or profit and loss statements.
* **Multi-location Management**: The system will track inventory but will not manage multiple locations or warehouses.
* **Mobile Application**: AIMS will be a web-based application only and will not have a native mobile app for iOS or Android devices.
* **Enterprise-Level Features**: AIMS is designed for small and medium-sized businesses (SMBs) and will not include features required by large enterprises, like multi-department coordination or global supply chain management.

**2.5 Project Objectives**

* **Automate Inventory Tracking**: Streamline and automate the process of tracking inventory levels, product orders, and stock updates to ensure accurate and real-time data.
* **Provide Detailed Reporting and Analysis**: Offer in-depth reports on inventory performance, sales, and purchase history to help businesses make data-driven decisions.
* **Facilitate Role-Based Access and Actions**: Enable different user roles (such as admin, manager, and staff) to access specific system features based on their responsibilities, ensuring secure and efficient operations.
* **Enhance Operational Efficiency**:  
  Automate key business processes, such as sales tracking, billing, and order processing, to reduce administrative overhead and improve the speed and accuracy of daily operations.
* **Improve Decision-Making with AI Integration**:  
  Incorporate AI-driven insights to predict trends, suggest reorder points, enabling businesses to make proactive decisions and reduce stockouts or overstocking.
* **Enhance User Experience**:  
  Focus on creating an intuitive and user-friendly interface that reduces training time and increases adoption among users, from store staff to top-level management.

**2.6 Stakeholders & Affected Groups**

 **Stakeholders**:

* **Business Owners**: Focused on overall system performance, efficiency, and scalability.
* **Admins**: Manage system settings, user roles, and ensure smooth operation.

 **Users**:

* **Staff Members**: Handle sales, inventory updates, and report generation, relying on accurate, real-time data.

**2.7 Operating Environment**

The system will run on the following components:

* **Web Browsers**: The platform will be accessed through modern web browsers (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge, Safari) for user interactions.
* **SQL Database**: The system will use an SQL database to store and manage critical data, including inventory, sales transactions, and user information.
* **Django Framework**: The backend will be developed using the Django framework, handling system operations, user management, and interactions with the database.

**2.8 System Constraints**

The system will run on the following components:

1. **Internet Connection**:  
   A stable internet connection is required for users to access the platform and ensure its functionality. A reliable internet connection is essential for smooth operation, real-time updates, and the accurate syncing of data across all users and devices. In case of an unstable connection, users may experience delays in data updates or interruptions in accessing certain features.
2. **Database Storage**:  
   The system’s performance and data storage capacity will be determined by the available storage in the database. The database will store all critical information, including inventory data, sales records, and user details. The storage limit is dependent on the database provider and the server's physical capacity. As the system scales and data increases, additional storage may be needed to maintain performance and prevent data overload. Regular database management, including optimizing queries and archiving old data, will be required to ensure efficient operation and avoid hitting storage limits

**2.9 Assumptions & Dependencies**

* **The system depends on Django for backend functionality.**

The **Django framework** will handle backend operations, including user authentication, data processing, and managing interactions between the front-end and the database.

* **Relies on SQL for database management.**

The system will rely on an **SQL-based database** to store and manage critical data such as inventory, sales transactions, and user information. SQL will ensure efficient data retrieval and support complex queries.

**3. External Interface Requirements**

**3.1 Hardware Interfaces**

* **Server with minimum 8 GB RAM and 500 GB storage.**

The system will be hosted on a server with a minimum of **8 GB RAM** and **500 GB storage**. This configuration will ensure sufficient memory and storage for handling system operations, data management, and supporting multiple users simultaneously. The server will store the database, handle backend processes, and ensure smooth performance even with increasing data and traffic.

* **User devices: PCs or laptops with modern web browsers.**

Users can access the system via **PCs or laptops** equipped with modern web browsers (e.g., **Google Chrome**, **Mozilla Firefox**, **Microsoft Edge**, **Safari**). These devices must have stable internet access to interact with the platform and perform tasks like inventory management, sales tracking, and report generation.

**3.2 Software Interfaces**

* **Django framework.**

The backend of the system will be developed using the **Django framework**, which will manage core functionalities like user authentication, data processing, and interactions with the database. Django’s built-in features will ensure secure, scalable, and maintainable backend operations for tasks such as inventory tracking, sales management, and generating reports.

* **SQL Database.**

The system will use an **SQL database** to store all critical data, including inventory records, sales transactions, and user information. SQL ensures efficient data retrieval, supports complex queries, and guarantees data integrity. The relational database structure allows for smooth management and relationships between various data points in the system.

* **Integration with Excel for report generation.**

The system will integrate with **Excel** to facilitate the generation of detailed, customizable reports. This integration will allow users to export data from the platform into Excel format for further analysis or sharing. It will support reports on inventory performance, sales, and other key metrics, offering users flexibility in handling data outside of the system.

**3.3 Communications Interfaces**

* **Local area network or internet for user access**.

The system will require either a **Local Area Network (LAN)** or an **internet connection** to provide users with access. A **LAN** can be used in environments where the system is deployed within a closed network, allowing users within the same physical location (e.g., office or warehouse) to access the platform. Alternatively, an **internet connection** will enable users to access the system remotely from any location, ensuring flexibility and ease of use. Both options will ensure that users can interact with the system in real time, enabling tasks such as inventory management, sales tracking, and reporting

# System Functions / Functional Requirements

**4.1 System Functions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref #** | **Function** | **Category** | **Attribute** | **Details & Constraints** |
| R1.1 | Record sales and purchases | Evident | System Response time | Data displayed within 3 seconds. |
| R1.2 | Update inventory levels on transactions | Hidden | Concurrent user load | Supports up to 50 users concurrently |
| R1.3 | Generate invoices and bills | Evident | Response time | Invoice generated within 5 seconds. |

**4.2 Use Cases**

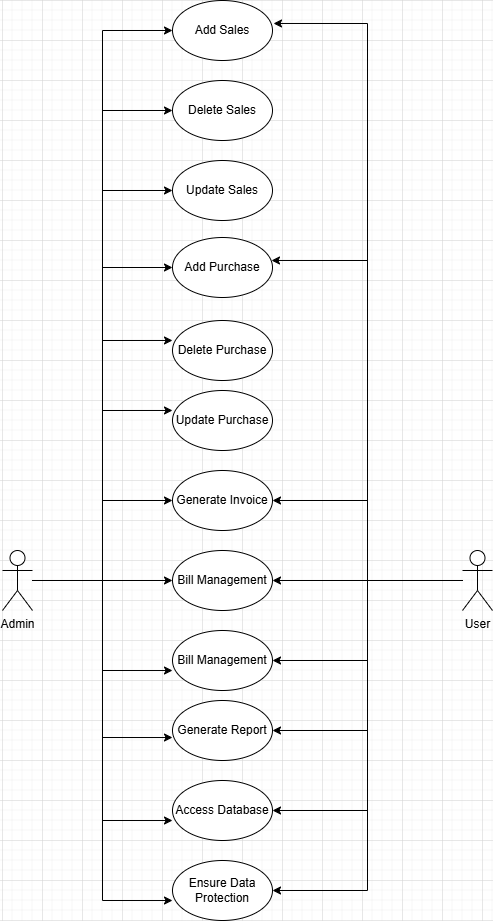
**4.2.1 List of Actors**

* **Admin:** Manages all system configurations and reports.
* **Staff:** Adds records for sales, purchases, and inventory.

**4.2.2 List of Use Cases**

* Add sale records.
* Manage purchase entries.
* Track inventory levels.
* Generate invoices and reports.

### Use Case Diagram



### Description of Use Cases

### Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| **1**. This use case begins when a customer arrives at the Point of Sale checkout with items to purchase. |  |
| **2**. The cashier records each item. | **3**. Determines the item price and adds the item information to the running sales transaction. |
| **4**. Repeats for all items until the sale is complete. | **5**. Updates the total price, including any applicable taxes. |
| **6**. The cashier confirms the sale is ready for payment. |  |
| **7**. Customer selects payment type: |  |
| - a. If cash payment, see section "Pay by Cash". |  |
| - b. If credit payment, see section "Pay by Credit". |  |
|  | **8**. Logs the completed sale. |
|  | **9**. Updates inventory levels. |
|  | **10**. Generates a receipt. |
| **11**. Cashier gives the receipt to the customer. |  |
| **12**. The customer leaves with the items purchased. |  |

### Alternative Course

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| Step 2: Invalid item identifier entered. | System indicates an error and prompts for re-entry. |
| Step 7: Customer cannot pay. | System cancels the sales transaction. |

### Section: Pay by Cash

#### Typical Course of Events

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| Customer makes a cash payment. |  |
| Cashier records the cash tendered. | System presents the balance due back to the customer, if any. |
| Cashier deposits received cash and provides balance. |  |

**Alternative Course**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| Step 1: Customer does not have sufficient cash. | Cashier cancels the sale or initiates another payment method. |

# Non - Functional Requirements

### **4.1 Performance Requirements**

 **Concurrent Users**:  
The system must be capable of handling up to **50 concurrent users** without experiencing performance degradation. This includes smooth navigation, fast response times, and uninterrupted operations for all active users.

 **Page Load Time**:  
All web pages must load within **3 seconds** under normal operating conditions. This ensures a seamless user experience, particularly for critical functions like inventory management, sales transactions, and report viewing.

 **Inventory Updates**:  
Changes to inventory levels, such as stock additions, reductions, or adjustments, must be reflected in the database within **5 seconds** of completing a transaction. This ensures real-time accuracy for inventory data and prevents discrepancies.

 **Report Generation**:  
Generating detailed reports (e.g., sales, purchases, inventory summaries) must be completed within **10 seconds**. This ensures that users can quickly access insights and data without delays, even when the system is handling a substantial amount of data.

### **4.2 Safety Requirements**

 **Automatic Data Backups**:  
The system must include a feature for automatic data backups at regular intervals to prevent data loss in case of hardware failures or unexpected errors.

 **Input Validation**:  
All user inputs must undergo strict validation to ensure they are valid and free of harmful content, protecting the system from potential errors or security risks.

 **Low Inventory Warnings**:  
The system should display warnings or notifications whenever inventory quantities fall below a predefined minimum threshold, helping users avoid stockouts and maintain optimal inventory levels.

### **4.3 Security Requirements**

 **Role-Based Access Control**:  
The system must implement role-based access control (RBAC) to ensure that only authorized users can perform specific actions based on their roles (e.g., admin, manager, or staff).

 **Data Encryption**:  
All sensitive data, such as customer and vendor information, must be encrypted both during storage and transmission to protect against unauthorized access and breaches.

 **Strong Password Policies**:  
The system must enforce strong password policies for all user accounts, requiring a combination of uppercase and lowercase letters, numbers, and special characters to enhance security.

### **4.4 Reliability Requirements**

 **High Uptime**:  
The system must maintain at least **99.5% uptime**, ensuring minimal disruptions to business operations and consistent availability for users.

 **Automatic Recovery**:  
Automatic recovery mechanisms should be in place to restore the system to a stable state quickly after unexpected crashes or failures, minimizing downtime.

 **Stress Testing for Critical Features**:  
Critical features like transaction processing and inventory updates must undergo rigorous testing to ensure they can handle peak loads without performance issues.

### **4.5 Usability Requirements**

 **User-Friendly Interface**:  
The system interface must be intuitive and straightforward, allowing new users to become proficient with minimal training, requiring no more than **1 hour**.

 **Dashboard Insights**:  
A comprehensive dashboard should display key metrics, including inventory status, sales trends, and purchase history, providing users with quick and actionable insights.

 **Clear Error Messages**:  
Error messages must be easy to understand and provide actionable guidance to help users resolve issues efficiently (e.g., "Invalid product ID. Please check and re-enter.").

### **4.6 Supportability Requirements**

 **Spreadsheet Integration**:  
The system must support exporting and importing data in popular spreadsheet formats, such as **Excel**, to facilitate reporting and data analysis.

 **Error Logging**:  
Comprehensive logging mechanisms must be implemented to record errors and system activities, enabling easier troubleshooting and support for system maintenance.

### **4.7 User Documentation**

1. **User Manual**:  
   A detailed user manual must be provided, explaining all major functionalities such as adding products, managing inventory, and generating invoices.
2. **Embedded Help Tooltips**:  
   Help tooltips should be integrated within the application to guide users on key operations, ensuring quick and accessible assistance during usage.
3. **Admin Documentation**:  
   Comprehensive admin documentation should cover instructions for tasks such as creating user accounts, setting role-based permissions, and performing system backups.
4. **FAQs Section**:  
   A frequently asked questions (FAQs) section should be included, addressing common issues and providing solutions to enhance user self-service.

.

# References

List References