

Team members:

Lokesh R- 1BM20C078

MD Suraj Kumar -1BM20CS079

M Udaya Raj -1BM20CS080

Prashant R Joshi-1BM20CS110

## **Problem Statement:**

The client currently manages their inventory manually, which is time-consuming and prone to errors. The client needs a system to automate inventory management processes, reduce errors, and provide real-time inventory information. The system should enable the client to manage their inventory efficiently and make informed decisions based on accurate data. Therefore, there is a need for a Stock Maintenance System to automate inventory management processes and provide real-time inventory information.

## **Software Requirement Specification(SRS)**

### **1. Introduction:**

- 1.1. **Purpose of this Document:** The purpose of the Stock Maintenance System is to provide an automated solution for managing inventory for a business. The system will help the client save time and effort in managing inventory, and provide accurate data for decision-making. The system aims to automate inventory management processes, reduce errors, and provide real-time inventory information. Additionally, the system should enable the client to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history. Ultimately, the purpose of the Stock Maintenance System is to improve the overall efficiency of inventory management for the client's business.
- 1.2. **Scope of this document** –The scope of the Stock Maintenance System is to manage inventory for the client's business. The system will include functionalities for inventory management such as adding new items, updating stock levels, and generating reports. The system will also provide real-time inventory information to the client, enabling them to make informed decisions. The Stock Maintenance System will be designed to handle a large amount of data and will be able to manage multiple locations of inventory. The system will have a user-friendly interface that allows for easy navigation and will be developed using the latest web technologies. However, the Stock Maintenance System will not include functionalities for financial management or point of sale.

- 1.3. **Overview** – The Stock Maintenance System is a web-based application developed using the latest web technologies. The system will have a user-friendly interface, making it easy for the client to use. The system will automate inventory management processes, reduce errors, and provide real-time inventory information. The system will be developed using an agile methodology, allowing for iterative development and quick deployment of new features. The Stock Maintenance System will enable the client to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history. Ultimately, the Stock Maintenance System aims to improve the overall efficiency of inventory management for the client's business.

## 2 **General description:**

The Stock Maintenance System is an online platform designed to streamline the inventory management processes of businesses. It provides a centralized inventory management system accessible through a web browser, eliminating the need for manual record keeping and reducing errors. The system enables businesses to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history.

The Stock Maintenance System is scalable and can handle a large volume of data. It can manage inventory across multiple locations, making it suitable for businesses with warehouses or stores in different locations. The system is customizable, allowing businesses to tailor the platform to their specific needs.

The system will be developed using the latest web technologies, ensuring that it is secure, reliable, and easy to use. The development process will follow an agile methodology, allowing for quick and iterative deployment of new features. The system will be thoroughly tested to ensure that it meets the client's requirements and is free of bugs.

The Stock Maintenance System will be deployed on a cloud server, making it accessible from anywhere with an internet connection. The system will have a user-friendly interface, making it easy for users to add new items, update stock levels, and generate reports. However, the Stock Maintenance System will not include functionalities for financial management or point of sale.

### 3 **Functional Requirements:**

- **User Management:**  
The system must be able to manage user accounts for staff and customers. Staff accounts must be able to access the management functions of the system, while customer accounts will have limited access to view and manage their own reservations.
- **Reservation Management:**  
The system must be able to manage hotel room reservations, including the ability to check room availability, reserve rooms, modify and cancel reservations, and track payment and deposit information.
- **Room Management:**  
The system must be able to manage hotel rooms, including the ability to assign rooms to guests, track room occupancy, manage room cleaning and maintenance schedules, and track room availability.
- **Billing and Payment Management:**  
The system must be able to manage guest billing and payment information, including the ability to generate invoices, track payments, and generate financial reports.
- **Inventory Management:**  
The system must be able to manage hotel inventory, including the ability to track inventory levels, reorder inventory, and generate reports on inventory usage and availability.
- **Reporting:**  
The system must be able to generate various reports, including financial reports, occupancy reports, inventory reports, and customer feedback reports.

### 4 **Interface Requirements:**

- **User Interface:**  
The user interface of the system should be user-friendly, easy to navigate, and responsive. The interface should provide the user with the ability to manage inventory, purchase orders, sales orders, and generate reports.

- **Browser Compatibility:**  
The system should be compatible with multiple browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- **Mobile Responsiveness:**  
The system should be responsive to mobile devices, enabling users to access the system from their smartphones and tablets.
- **API Integration:**  
The system should be able to integrate with external APIs, such as suppliers, accounting software, and other inventory management systems.
- **Import and Export Data:**  
The system should allow users to import and export data from external sources, such as spreadsheets.
- **Error Handling:**  
The system should be able to handle errors and provide the user with feedback when an error occurs.
- **Language Localization:**  
The system should be able to support multiple languages, enabling users to use the system in their preferred language.
- **Printing:**  
The system should provide the user with the ability to print reports and documents.
- **Help and Support:**  
The system should provide help and support features such as tutorials, user guides, and a knowledge base to assist the user in using the system.

## **5 Performance Requirements:**

- **Response Time:**  
The system should respond to user actions, such as adding or updating inventory items, creating purchase orders, or generating reports, within 3 seconds or less.

- **Concurrent Users:**  
The system should be able to handle concurrent users and maintain the same level of performance, regardless of the number of users accessing the system simultaneously.
- **Scalability:**  
The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance.
- **Data Storage:**  
The system should be able to store large amounts of data, including inventory items, purchase orders, sales orders, and user information, without experiencing performance issues.
- **Backup and Recovery:**  
The system should include backup and recovery features to ensure data is safe and can be restored in the event of a system failure.
- **System Uptime:**  
The system should have a minimum uptime of 99.9%, ensuring that the system is available to users at all times.
- **System Monitoring:**  
The system should be monitored continuously for performance issues and system failures. The monitoring system should notify the administrator immediately in the event of an issue.
- **Load Testing:**  
The system should undergo load testing to ensure that it can handle the expected number of users and inventory items without experiencing performance issues.
- **Network Latency:**  
The system should be designed to minimize network latency, ensuring that users can access the system quickly and efficiently.
- **Compatibility:**  
The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing performance issues.

## 6 Design Constraints:

- **Response Time:**  
The system should respond to user actions, such as adding or updating inventory items, creating purchase orders, or generating reports, within 3 seconds or less.
- **Concurrent Users:**  
The system should be able to handle concurrent users and maintain the same level of performance, regardless of the number of users accessing the system simultaneously.
- **Scalability:**  
The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance.
- **Data Storage:**  
The system should be able to store large amounts of data, including inventory items, purchase orders, sales orders, and user information, without experiencing performance issues.
- **Backup and Recovery:**  
The system should include backup and recovery features to ensure data is safe and can be restored in the event of a system failure.
- **System Uptime:**  
The system should have a minimum uptime of 99.9%, ensuring that the system is available to users at all times.
- **System Monitoring:**  
The system should be monitored continuously for performance issues and system failures. The monitoring system should notify the administrator immediately in the event of an issue.
- **Compatibility:**  
The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing performance issues.

## 7 Non-Functional Attributes:

- **Usability:**  
The system should be designed with usability in mind, ensuring that it is easy to use and intuitive for the end-users. This may include compliance with relevant usability standards and best practices.
- **Security:**  
The system should be designed to ensure data security and prevent unauthorized access. This may include encryption of data, secure authentication and authorization, and compliance with relevant security standards.
- **Performance:**  
The system should be designed to meet performance requirements, such as response time, concurrent user capacity, and system uptime. This may include load testing and optimization.
- **Reliability:**  
The system should be reliable, ensuring that it is available to users at all times and that data is stored and processed correctly. This may include backup and recovery features, error handling, and fault tolerance.
- **Scalability:**  
The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance. This may include the ability to add more servers, load balancing, and clustering.
- **Maintainability:**  
The system should be designed to facilitate maintenance and support activities, such as updates, bug fixes, and user support. This may include modular architecture, version control, and documentation.
- **Portability:**  
The system should be designed to be portable, allowing it to be deployed on different hardware and software configurations. This may include compliance with relevant software standards, and use of platform-independent technologies.

- **Accessibility:**  
The system should be designed to be accessible to users with disabilities, ensuring compliance with relevant accessibility standards. This may include compliance with the Web Content Accessibility Guidelines (WCAG).
- **Compatibility:**  
The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing compatibility issues. This may include compliance with relevant software standards, and use of platform-independent technologies.
- **Interoperability:**  
The system should be designed to integrate with other systems or third-party software, such as accounting software or e-commerce platforms. This may include compliance with relevant integration standards and use of open APIs.

## **8 Preliminary Schedule and Budget:**

- **Project Scope Definition:**  
Duration: 1 week  
Budget: \$5,000
- **Requirements Gathering:**  
Duration: 2 weeks  
Budget: \$10,000
- **System Design and Architecture:**  
Duration: 4 weeks  
Budget: \$20,000
- **Development:**  
Duration: 12 weeks  
Budget: \$60,000
- **Quality Assurance and Testing:**  
Duration: 4 weeks  
Budget: \$20,000



- Deployment and User Training:  
Duration: 2 weeks  
Budget: \$10,000
- Total Duration: 25 weeks
- Total Budget: \$125,000

Note: The above schedule and budget are preliminary and subject to change based on further analysis and project requirements. It is also important to note that the budget may vary depending on the rates of the development team and any third-party tools or services required for the project.