4. Stock Maintainance System

Problem Statement: It's to ensure that stock levels are optimized that can help reduce cost, increase profitability, and improve customer's satisfaction. It's to ensure that there's no overstocking and loss of revenue.
It should be easy, scalable and customizable to meet the unique needs of businesses.
It should integrate with other systems too.

SRS

1. Introduction

1.1. Purpose: To define the functional and non-functional requirements req for an effective maintainance system. It serves as an outline to improve the operations that can enhance future needs.

1.2. Scope: Overall working and ensure the system is running as smooth as possible. The complexity of the system will affect the development time.

1.3 Overview: Application is developed to optimize the stock maintainance. So that the procss can take place more efficiently and improve the guest satisfaction and increased revenue.

2. General Description

- This system helps business manage their inventory by automating the process of tracking the stock level and reduce the loss. It typically includes stock tracking, order management, purchase mgmt, and reporting. Allows user business to track inventory at multiple locations, setup alert for low stock levels and generate reports on stock movement
- Overall motive of this system is to reduce costs and increase revenue

3- Functional Requirements

- Stock Tracking: system should enable businesses to track inventory and keep records of accurate amount.
- Purchase Management: should be able to manage the purchase orders including creating and tracking it.
- Reporting: should provide comprehensive reporting features including stock levels, sales, trends, etc.
- Inventory control, Integration, Customisation, others are also the requirements.

4. Interface Requirement

- The system should be user friendly and intuitive and require minimal training.

- System should provide dashboard that gives a quick overview of inventory data.
- Search function should be provided to enable find.
- Filtering and sorting options that enable user organize inventory.
- Mobile accessibility and integration.

5. Performance Requirements

- System should be able to perform all functions quickly even managing large amount of data
- System should be able to handle large amount of data as it grows
- System should be reliable & available.
- should have a robust security feature.
- Integration with other systems.
- Data accuracy and Response-time.

6. Design constraints:

Hardware constraints: It should be neatly integrated with the already built infrastructure at different locations

Software limitations: constraints on amount of data that can be processed and stored.

Security: to adhere to secure & safe storage of data

7. Non functional requirements

- Usability: should be easy to use and intuitive, even for non-technical users.
- Accessibility: should be accessible to authorized personnels.
- System should be fast and responsive
- Should be secure, protecting inventory data from unauthorized access.
- System should be reliable and available all time.
- Should be able to handle increasing amount of data.

8. Schedule and Budget

Planning: 20-30 days
Development: 3 months
Testing: 1 month
Deployment: 15 days