2) Stock ~~Management~~ Maintenance System:

1. Introduction

1.1 Purpose of this document: This document details the requirements for Stock ~~[illegible]~~ Maintenance System

1.2 Scope of this document: in this, Stock details, prices, requirements and maintenance are outlined

1.3 Overview: Stock Maintenance System is designed to simplify the process of managing inventory for business. It automates the following processes: stock & order tracking. It sends reminders when a product is on the verge of selling out. Helps users track and return their orders with ease

2. General Description:

The system updates the stock of each and every item after each and every purchase. It ~~provides~~ generates alerts when the stock of ~~it~~ items falls below a certain value. It automatically generates a receipt of each and every transaction. Provides the user with real time updates of their orders thus enabling easy tracking

3. Functional Requirements:

- Inventory tracking: The system keeps a real-time record of stock levels, locations and item statuses and generates
- ~~[illegible]~~
~~[illegible]~~ alerts if the stock falls below required value

- Prediction: The system can predict when stock has to be replenished.
- Search: It enables users to search for the required products
- Display details: It displays details of each and every product with its amount, price description.

4. User Interface Requirements:
- User interface: user dashboard with account access to use the system
- System Interface: Integrated with systems to check availability, price and details of the products regularly

5. Performance Requirements:
- Maximum allowed error rate should be less than 1%
- It should the memory usage should not exceed 100Mb
- It should take not more than 2 secs to respond to each user activity
- It should handle 10000 concurrent users.

6. Design constraints:
- It should be compatible with Linux and windows
- Database should be used to keep track of items

7. Non Functional Attributes:
- Security: it should implement end to end encryption for

each and every user.
- Reliability: failure rate should be less than 0.001%.
- Scalability: It should be able to handle increasing load of users and their respective transactions.

8. Preliminary Schedule and Budget:

Schedule:

Requirements gathering: 1 month
Design phase: 1 month
Development Phase: 5 months
Testing Phase: 3 months

Budget: $100,000
Requirements gathering: $20,000
Design phase: $10,000
Development phase: $40,000
Testing Phase: $30,000