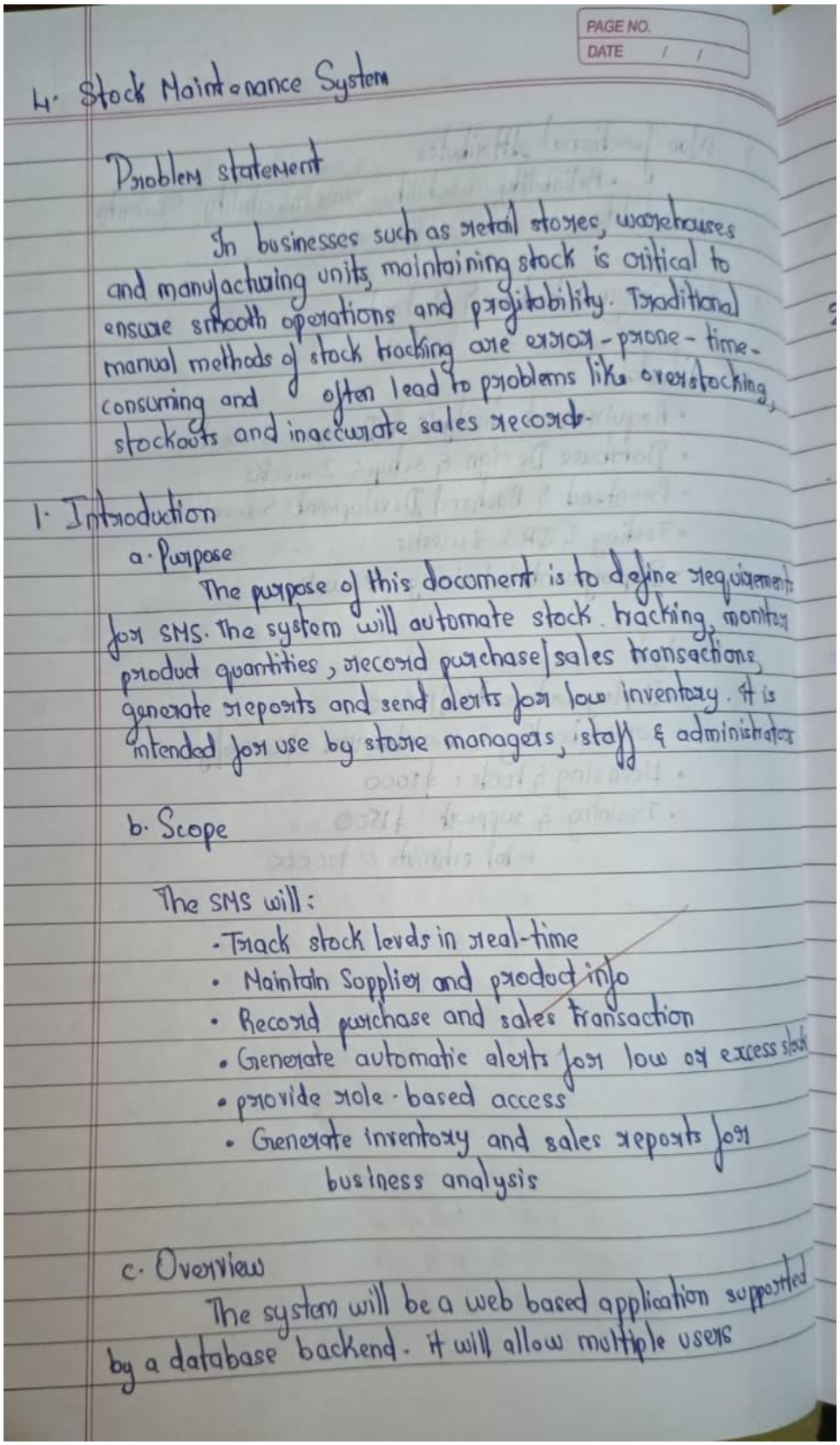
OOM LAB

SRS Document

Stock Maintenance System



4. Stock Maintenance System

Problem statement

In businesses such as retail stores, warehouses and manufacturing units, maintaining stock is critical to ensure smooth operations and profitability. Traditional manual methods of stock tracking are error-prone-time-consuming and often lead to problems like overstocking, stockouts and inaccurate sales records.

1. Introduction

a. Purpose

The purpose of this document is to define requirements for SMS. The system will automate stock tracking, monitor product quantities, record purchase/sales transactions, generate reports and send alerts for low inventory. It is intended for use by store managers, staff & administrators

b. Scope

The SMS will:

- Track stock levels in real-time
- Maintain Supplier and product info
- Record purchase and sales transaction
- Generate automatic alerts for low or excess stock
- provide role-based access
- Generate inventory and sales reports for business analysis

c. Overview

The system will be a web based application supported by a database backend. It will allow multiple users

PAGE NO.
DATE / /

to log in securely, update stock records and access real-time information. The application will integrate with barcode scanners or POS systems for ease of use.

2. General Description
   - Users: Admin, Store, Managers, staff
   - System Environment: web based interface, relational database
   - Assumptions: Users have valid login Credentials and basic computer skills.
   - Dependencies: stable Internet connection, barcode Scanning devices, server hosting.

3. Functional Requirements
   a. User Management
      - login / logout
      - Role based access
   b. Product & supplier Management
      - Add, update, and delete product details
      - Maintain Supplier info
   c. Stock Operations
      - Record incoming / outgoing stock
      - Update stock levels.
   d. Alerts & Notifications
      - Low stock alerts
      - Overstock alerts
      - Expiry Notifications.
   e. Reports & Analysis
      - Daily / weekly / monthly reports
      - Sales & purchase history
      - stock validation reports

4. Inter

4. Interface Requirements
   a. User Interface :
      - Dashboard for managers/admins
      - Mobile friendly staff interface.
   b. Hardware interface
      - Support barcode scanners & printers
      - standard PC/Mobile device compatibility
   c. Software interface :
      - Database : MySQL
      - Frontend : HTML/CSS/Java
      - Backend : Node.js/python

5. Performance Requirements
   - System must support 500+ concurrent users
   - Inventory updates must reflect in 22 seconds
   - Handle database of 500,00+ products.

6. Design Constraints
   - Must comply with data security policies
   - System should be cross platform
   - Should support data backup & recovery

7. Non-functional Requirements/Attributes
   - Reliability, usability, Maintainability, security and scalability.

8. Pre-liminary Schedule & Budget
   a. Schedule
      - Requirement Analysis & planning : 2weeks
      - Database & System Design : 2 weeks
      - Development : 6 weeks

PAGE NO.
DATE : / /

- Testing and Quality A : 3 weeks
- Deployment & training : 2 weeks

b. Budget (Estimated) :

- Development cost : $18000
- Server Hosting & Maintenance : $2500/yr
- Licenses & tools : $1000
- training & support : $1500

total estimated budged : $23000