Practical – 2

AIM: Prepare Software Requirement Specification (SRS) document for chosen application.

Record Book

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to specify the software requirements for an Inventory and Transaction Management System. It describes the system's intended features, functionalities, and constraints, and serves as a guide for design, development, and validation.

1.2 Scope

This system will allow administrators and users to manage inventory, track transactions, handle billing, and maintain customer and user records. Major modules include:

- User Management
- Item Management
- Customer Management
- Transaction Tracking
- Billing & History

1.3 Intended Audience

- Developers
- Testers
- Project Managers
- Stakeholders

1.4 Definitions, Acronyms, Abbreviations

- **CRUD** – Create, Read, Update, Delete
- **SRS** – Software Requirements Specification
- **ER** – Entity Relationship
- **DFD** – Data Flow Diagram

2. Overall Description

2.1 Product Perspective

This is a standalone system intended to digitize and streamline inventory and transaction processes. It interacts with databases to store user, item, transaction, and billing data.

2.2 Product Functions

- Register, search, and manage users and customers.
- Add, edit, and remove inventory items.
- Generate and track bills for customers.
- Monitor incoming and outgoing transactions.
- Maintain historical data for analysis and backup.

2.3 User Classes and Characteristics

- **Admin**: Full access to user, item, and transaction modules.
- **Record Users**: Limited access to manage transactions and customer details.

2.4 Operating Environment

- Web-based or desktop application
- Supported on Windows/Linux/Mac
- Relational database like MySQL or PostgreSQL

2.5 Design and Implementation Constraints

- The system must adhere to security standards (e.g., encrypted passwords).
- All sensitive operations require authentication.

2.6 User Documentation

User manuals and system help will be provided for navigation and operations.

3. System Features

3.1 User Management

- Register, search, and edit users
- Store contact info, DOB, and addresses
- Differentiate user roles

3.2 Item Management

- Add new items with name, stock, unit, prices, and image
- View low stock and all stock reports
- Edit/delete items

3.3 Customer Management

- Add/edit/delete customer profiles
- Link customers with transactions and bills

3.4 Transactions

- Record transactions: amount given, received, time limit, description
- Track balances and payment history
- List of incoming and outgoing amounts

3.5 Billing

- Generate bills with item, quantity, and date
- Search previous bills
- Download bill history

4. External Interface Requirements

4.1 User Interfaces

- Login and registration forms
- Dashboards for item/transaction/customer management
- Modal forms for CRUD operations

4.2 Hardware Interfaces

- Not hardware-dependent; standard computing device required

4.3 Software Interfaces

- Database system (e.g., MySQL)
- Backend language (e.g., Java, Python, PHP)
- Frontend framework (e.g., React, Angular, or plain HTML/CSS)

4.4 Communication Interfaces

- May include REST APIs for integration or mobile app support in future

5. Non-Functional Requirements

5.1 Performance Requirements

- System must respond to CRUD operations within 2 seconds

5.2 Safety Requirements

- Auto-logout after inactivity
- Frequent data backups

5.3 Security Requirements

- Role-based access control
- Encrypted passwords
- Secure login sessions

5.4 Maintainability & Scalability

- Modular code architecture
- Easily extendable to add new modules

6. Other Requirements

- System should log all actions for audit purposes
- Data import/export functionality is desirable

7. Appendices

- **A. Diagrams**: ER, DFD, Class, Activity, Sequence Diagrams
- **B. Glossary**: CRUD, Transaction, Stock, User Roles
- **C. Tools Used**: Draw.io for diagrams, MySQL for database (assumed)