Table of Contents

[**1. Introduction** 2](#_Toc141261632)

[***1.1 Purpose*** 2](#_Toc141261633)

[***1.2 Scope*** 2](#_Toc141261634)

[***1.3 Definitions, Acronyms, and Abbreviations*** 2](#_Toc141261635)

[**1.4 References** 2](#_Toc141261636)

[**2. Overall Description** 2](#_Toc141261637)

[***2.1 Product Perspective*** 2](#_Toc141261638)

[***2.2 Features*** 2](#_Toc141261639)

[***2.3 User Classes and Characteristics*** 2](#_Toc141261640)

[***2.4 Operating Environment*** 3](#_Toc141261641)

[**3. Design and Implementation Constraints** 3](#_Toc141261642)

[**4.Functional Requirements** 3](#_Toc141261643)

[***4.1. Product Management*** 3](#_Toc141261644)

[***4.2. Transaction Processing*** 3](#_Toc141261645)

[***4.3. User Management*** 3](#_Toc141261646)

[***4.4. Reporting and Analytics*** 3](#_Toc141261647)

[***4.5. Error Handling*** 3](#_Toc141261648)

[***4.6. System Configuration*** 4](#_Toc141261649)

[***4.7. Backup and Restore*** 4](#_Toc141261650)

[***4.8. Integration*** 4](#_Toc141261651)

[**5. Non-Functional Requirements** 4](#_Toc141261652)

[***5.1 Performance*** 4](#_Toc141261653)

[***5.2 Security*** 4](#_Toc141261654)

[***5.3 Usability*** 4](#_Toc141261655)

[***5.4 Scalability*** 4](#_Toc141261656)

[**6. Appendices** 5](#_Toc141261657)

Software Requirements Specification (SRS) for Point of Sale (POS) System

## **1. Introduction**

### ***1.1 Purpose***

The purpose of this document is to define the requirements and functionalities of the Point of Sale (POS) System.

### ***1.2 Scope***

The POS System is designed to handle the management of products and user transactions in a retail environment. It includes the classes Product and User, each with specific attributes and methods.

### ***1.3 Definitions, Acronyms, and Abbreviations***

POS – Point of Sale

### **1.4 References**

chatGPT

## **2. Overall Description**

***2.1 Product Perspective***

The context of the Point of Sale (POS) system lies within the larger system of a retail environment. The POS system serves as a critical component in the overall retail management ecosystem, facilitating the day-to-day operations related to product sales, inventory management, and customer transactions.

### ***2.2 Features***

* Product management.
* Transaction processing
* Reports and analytics
* User management.
* Error handling.
* Data storage.
* Offline mode.

### ***2.3 User Classes and Characteristics***

Supports two types of user:

1. Admin

Has the following privileges:

* Adding products.
* Updating product prices and quantity.
* Viewing Reports and Analytics.
* Creating new user accounts.

1. Employee

Has the following privileges:

* Selling products.
* Viewing available products.
* Change password.

***2.4 Operating Environment***

The system is compatible with all Operating System and Architecture (x64 or x86)

## **3. Design and Implementation Constraints**

The command line interface may deter user since it is difficult to use.

## **4.Functional Requirements**

***4.1. Product Management***

The system shall allow users to add a new product to the inventory with attributes: id, name, quantity, price, and type. The system shall provide the capability to update the price and quantity of an existing product. The system shall prevent users from adding a product with the same id as an existing product (ensuring uniqueness).

***4.2. Transaction Processing***

The system shall enable users to initiate a sale by selecting products from the inventory and specifying their quantities. The system shall calculate and display the total cost of the selected products in real-time during the transaction. The system shall update the product quantity in the inventory after a successful sale. The system shall generate and print a detailed receipt for each completed transaction, including itemized information and the total amount.

***4.3. User Management***

The system shall provide authentication and access control for different user roles (e.g., cashiers, managers) to ensure data security. The system shall allow authorized users to log in and log out of the POS system. The system shall track and maintain an audit log of user activities, including login attempts, sales, and product updates.

***4.4. Reporting and Analytics***

The system shall generate daily, weekly, and monthly sales reports for business analysis. The system shall provide insights into top-selling products and their performance over time

***4.5. Error Handling***

The system shall handle and display appropriate error messages for common scenarios such as invalid inputs, insufficient stock, etc. The system shall gracefully recover from unexpected errors and prevent data loss.

***4.6. System Configuration***

The system shall allow administrators to configure system settings, such as tax rates, currency format and discount.

***4.7. Backup and Restore***

The system shall periodically back up the data to prevent data loss in case of system failure. The system provides a mechanism to restore the system to a previous backup if needed.

### ***4.8. Integration***

The system shall be capable of integrating with external hardware (e.g., barcode scanners, receipt printers) and payment gateways.

## **5. Non-Functional Requirements**

### ***5.1 Performance***

Reliable in terms of speed and accuracy.

### ***5.2 Security***

Encrypt file to prevent unauthorized access.

### ***5.3 Usability***

Command line-based interface with appropriate messages to enhance usability and friendliness.

### ***5.4 Scalability***

The system is easily scalable to meet customer needs.

And can support any retail business.

## **6. Appendices**

Available Classes.

|  |
| --- |
| Product |
| id: string  name: string  quantity: int  price: double  type: string |
| addProduct(): void  sellProduct(): void  updateProductPrice(): void  updateProductQuantity(): void  printReceipt(): void  analyzeSales(): void  searchProduct(): void |

|  |
| --- |
| User |
| username: string  password: string |
| getPassword(): string  setPassword(): string  setUsername(): string  getUsername(): string  createAccount(): void  changePassword(): void  login(): void |

|  |
| --- |
| Admin |
| role: string |
| displayRole(): void |

|  |
| --- |
| Employee |
| role: string |
| displayRole(): void |