Software Requirements Specification (SRS) for Billing System

# **Introduction**

* 1. **Purpose**  
     The purpose of this document is to specify the requirements for the Billing System in a department store. This system is designed to automate the billing process, manage product inventory, track customer transactions, and generate reports. The document outlines the system’s functionalities, performance, interface, and other critical factors necessary for development.
  2. **Scope**  
     The Billing System will be used in a department store to handle day-to-day sales operations, including product management, customer management, billing, and inventory control. The system will also generate various reports for sales and inventory analysis. It will be accessible by store staff with varying levels of permissions based on their roles (e.g., Admin, Staff).

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **GUI**: Graphical User Interface
* **DBMS**: Database Management System
* **CRUD**: Create, Read, Update, Delete

**1.4 References**

* Java Documentation: [Oracle Java Documentation](https://docs.oracle.com/en/java/)
* MySQL Documentation: [MySQL Documentation](https://dev.mysql.com/doc/)
* Software Engineering by Ian Sommerville, 10th Edition

**1.5 Overview**  
This document is structured to provide a comprehensive overview of the system, including detailed functional and non-functional requirements, system models, and interface descriptions.

# **Overall Description**

**2.1 Product Perspective**  
The Billing System is a standalone application that will integrate with the department store’s existing IT infrastructure. It is designed to replace manual billing processes, thereby reducing errors and improving efficiency. The system will interact with a backend database to store and retrieve data.

**2.2 Product Functions**

* **User Authentication**: Secure login and role-based access control.
* **Product Management**: Add, update, delete, and view product details.
* **Customer Management**: Manage customer information and purchase history.
* **Billing**: Generate and print bills for customers, apply discounts, and calculate totals.
* **Inventory Management**: Track stock levels and generate alerts for low stock.
* **Report Generation**: Produce sales and inventory reports for analysis.

**2.3 User Classes and Characteristics**

* **Admin**: Has full access to the system, including user management, product management, and report generation.
* **Staff**: Has limited access, primarily to the billing and inventory management modules.

**2.4 Operating Environment**

* **Operating System**: Windows/Linux/MacOS
* **Database**: MySQL or SQLite
* **Programming Language**: Java
* **Hardware**: Standard PC with at least 4 GB RAM, 100 GB HDD

**2.5 Design and Implementation Constraints**

* The system must be developed using Java for the front-end and SQL for the back-end.
* The system should be scalable to handle large amounts of data without performance degradation.
* The application must ensure data security, particularly for customer and transaction data.

**2.6 Assumptions and Dependencies**

* The users have basic knowledge of using a computer and can navigate a GUI.
* The system depends on a stable database connection for real-time operations.
* The store’s network infrastructure must support client-server communication.

# **System Features**

**3.1 User Authentication**  
**3.1.1 Description**  
The system will require users to log in with a username and password. User roles (Admin or Staff) will determine the level of access to system features.

**3.1.2 Functional Requirements**

* **REQ-1.1**: The system shall validate user credentials against the database.
* **REQ-1.2**: The system shall allow Admin users to manage user accounts.
* **REQ-1.3**: The system shall log user activities for audit purposes.

**3.2 Product Management**  
**3.2.1 Description**  
The system will allow users to add, update, delete, and view product information, including product name, category, price, and stock quantity.

**3.2.2 Functional Requirements**

* **REQ-2.1**: The system shall allow Admin users to add new products to the inventory.
* **REQ-2.2**: The system shall allow Admin and Staff users to update existing product details.
* **REQ-2.3**: The system shall allow Admin users to delete products from the inventory.
* **REQ-2.4**: The system shall display a list of products with search and filter options.

**3.3 Billing**  
**3.3.1 Description**  
The billing module will generate bills for customers, apply discounts, and calculate totals based on selected products.

**3.3.2 Functional Requirements**

* **REQ-3.1**: The system shall allow users to select products and quantity for billing.
* **REQ-3.2**: The system shall calculate the total bill amount, including applicable taxes and discounts.
* **REQ-3.3**: The system shall generate a printable bill receipt.

**3.4 Inventory Management**  
**3.4.1 Description**  
The inventory management module will track stock levels and provide alerts when stock is low.

**3.4.2 Functional Requirements**

* **REQ-4.1**: The system shall update stock levels automatically after each transaction.
* **REQ-4.2**: The system shall generate alerts for products with stock levels below a predefined threshold.

**3.5 Report Generation**  
**3.5.1 Description**  
The report generation module will provide sales and inventory reports for analysis.

**3.5.2 Functional Requirements**

* **REQ-5.1**: The system shall generate daily, weekly, and monthly sales reports.
* **REQ-5.2**: The system shall generate inventory status reports.
* **REQ-5.3**: The system shall allow reports to be exported in PDF and Excel formats.

# **External Interface Requirements**

**4.1 User Interfaces**

* The system will have a GUI developed using Java Swing or JavaFX.
* The main interface will include menus for navigating between different modules (e.g., Billing, Product Management, Reports).
* Forms will be used for data entry, with validation for mandatory fields.

**4.2 Hardware Interfaces**

* The system will interface with standard input devices (keyboard, mouse) and output devices (monitor, printer).

**4.3 Software Interfaces**

* The system will interface with a MySQL database for data storage and retrieval.
* The system will generate reports that can be exported to third-party applications like Excel and PDF viewers.

**4.4 Communications Interfaces**

* The system will communicate with the database over TCP/IP if deployed in a client-server environment.

# **Other Non-Functional Requirements**

**5.1 Performance Requirements**

* The system should be able to handle up to 10,000 products and 1,000 transactions per day without performance degradation.
* The system should load the product list within 3 seconds for up to 1,000 items.

**5.2 Safety Requirements**

* The system should ensure data integrity by using transactions to prevent data loss in case of a failure during data operations.
* Regular backups should be scheduled to prevent data loss.

**5.3 Security Requirements**

* User passwords should be stored in the database using encryption.
* The system should enforce role-based access control to restrict unauthorized access to certain features.

**5.4 Software Quality Attributes**

* **Usability**: The system should have an intuitive interface, making it easy for users to perform tasks with minimal training.
* **Reliability**: The system should have an uptime of 99.9% and handle errors gracefully without crashing.
* **Maintainability**: The system’s codebase should be modular and well-documented to facilitate easy updates and maintenance.

**5.5 Business Rules**

* All products must have a unique identifier (Product ID).
* Only Admin users can delete products or modify user roles.
* Discounts applied during billing must not exceed 50% of the total bill.

# **Other Requirements**

**6.1 Database Backup**

* The system should automatically back up the database daily and store the backup in a secure location.

**6.2 System Documentation**

* The system should include a user manual and technical documentation covering installation, configuration, and troubleshooting.

**6.3 Training**

* Training sessions should be provided for store staff to familiarize them with the system's features

# **Requirements Traceability Matrix (RTM)**

The RTM ensures that all requirements are covered by design, development, and testing activities. Below is a simplified RTM example for the Billing system:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | Requirement Description | Source | Design Document | Development | Test Case ID | Status | Comments |
| R1 | The system must support multiple payment methods. | Business Analyst | Design Doc V1 | Dev Team A | TC01 | In Progress |  |
| R2 | The system should calculate and apply discounts. | Business Analyst | Design Doc V1 | Dev Team B | TC02 | Completed |  |
| R3 | The system must generate detailed receipts. | Business Analyst | Design Doc V2 | Dev Team C | TC03 | Not Started |  |
| R4 | The system should update inventory in real-time. | Business Analyst | Design Doc V3 | Dev Team A | TC04 | In Progress |  |
| R5 | The system must provide sales reporting. | Business Analyst | Design Doc V2 | Dev Team B |  |  |  |