Software Requirements Specification

for

Point of Sale Simulator, Release 1.0

Version 1.0 draft 1

Prepared by Agile forces

Process Impact

September 14, 2023

Ganta Srilalitha PES1UG21CS200
Harshitha Krishna R PES1UG21CS227
Indhu R PES1UG21CS237 Jeshwanth
Krishnama Naidu PES1UG21CS246

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Agile Forces	14/09/23	initial draft	1.0 draft 1

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the POS(Point of Sale)- Simulator. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Project Scope

The Point of Sale (POS) Simulator is designed to provide a comprehensive training and testing platform for retail and business.A Point of Sale (POS) system is a critical component for retailers, both small and large, as it serves as the central hub for completing sales transactions and managing various aspects of their business operations.

POS Simulator helps the retailers to store the following information with every purchase made by customers: a unique transaction number assigned to every transaction, the items purchased and their prices, the date and time of the transaction and the total price of all the items bought.

For the database of items, the system will only store information about the products offered in the store. The product type may be groceries or accessories, the system shall be able to store the unique product identification number, the product name, product type, the price the item was bought, and the selling price, stock of product for each item in the database.

In generating reports on sales, the system can show detailed reports on the sales made on a daily basis. The user can view the report any time. All transactions are stored by the system for future reference. Each transaction has the items purchased, payment method used , and the total cost of purchase.

1.3 Product Features

1. Process Sales Transactions:

   -The system shall allow cashiers to input items, calculate the total price, apply discounts, and finalize sales transactions.

   -It shall support various payment methods, including cash, credit cards, and mobile payments.

   -The system shall generate itemized receipts for customers.

2. Manage Product Inventory:

   -        The system shall maintain a database of products, including their names, prices, and quantities in stock.

   -        It shall allow authorized users to add, modify, or remove products from the inventory.

3. Generate Sales Reports:

   -        The system shall generate daily transaction reports, including total revenue, top-selling products, and sales trends.

   -        Reports can be visually interpreted using graphs and charts.

1.4 References

1. [POS Device Simulator SPDH HPDH, card reader, CHIP and NFC (neapay.com)](#)

2. [(38) Software Requirements Specification for Point Of Communication Sale System | mahmoud almomani - Academia.edu](#)

3. [POS SRS Document.docx - Topic: Doc Code 1.0 Point Of Sale Co Logo Effective date](#)

29/12/2011 IT Department Version 1.1 kappa statistics Software | Course Hero

4. (38) SRS Document for Point Of Sale | Davies Michael - Academia.edu

5. Srs Example 2010 Group2 | PDF | Databases | Mobile App (scribd.com)

6. Point-Of-Sale (POS) System | PDF | Point Of Sale | Electronic Data Interchange (scribd.com)

2. Overall Description

2.1 Product Perspective

The POS (Point of Sale) System is a novel system designed to modernize and streamline the current manual and paper-based transaction processes in retail establishments. This transition marks a significant shift from traditional methods. As depicted in Figure 1 , the system outlines external entities and interfaces. It is crucial to note that this system's evolution will span multiple phases, with the ultimate goal of integrating with online ordering services from various local restaurants and credit/debit card authorization services in future releases.

2.2 User Classes and Characteristics

Cashier:

- Cashiers are the primary users of the POS Simulator, responsible for conducting sales transactions.
- Cashiers may have varying levels of experience and skill, ranging from novice to expert.
- Cashiers may vary in experience and skill levels.
- They are limited to performing transactions. They can view product information, add products to the order ,edit the order, such as modifying quantities or removing items, and cancel orders if necessary.
- Cashiers require a User-friendly interface for quick and efficient transaction processing.
- POS Simulator should support various payment methods (cash, credit card, mobile payments).

Administrator:

- An administrator is responsible for configuring the system, managing product inventory, and generating daily transactions and sales reports.
-Administrators can view total sales of each product, and unique invoice generated at each counter along with cashier information at each counter.
- The simulator should offer administrative tools and dashboards with features for configuring the system and generating reports. The interface should be designed to accommodate their responsibilities effectively.

Customer:

-Customers are end-users who make purchases of items using various payment methods.

-Select products they want to purchase.
-Choose their preferred payment method (e.g., cash, credit card, mobile payment).
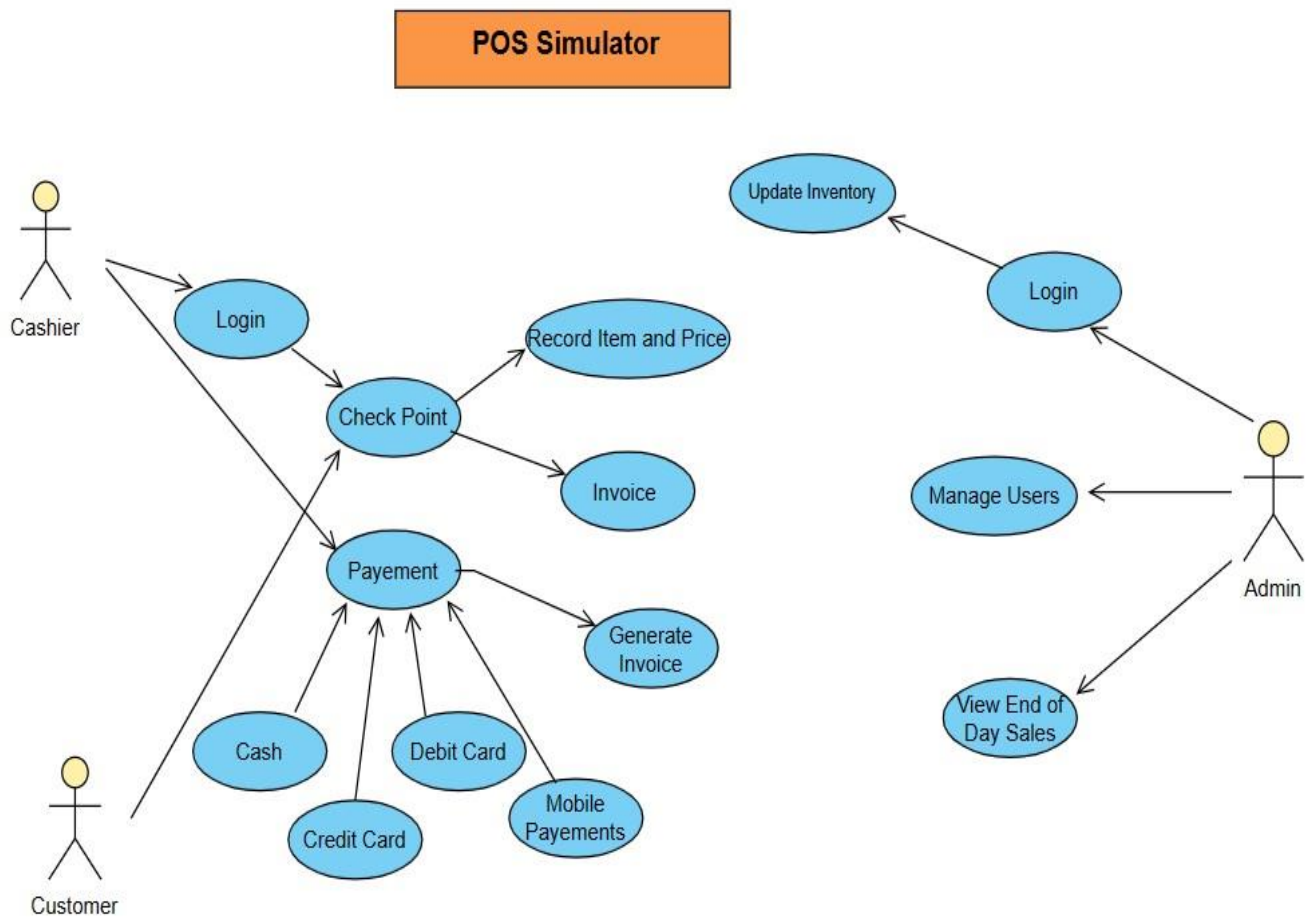
USE CASE DIAGRAM

Figure 1

2.3 Operating Environment

OE-1: The POS Simulator shall run on Windows operating systems.
OE-2: The application requires a minimum screen resolution of 1280x768 pixels.

OE-3: It is compatible with modern hardware configurations, including barcode scanners and receipt printers.

2.4 Design and Implementation Constraints

CO-1:   Compliance with Company's POS Development Standards
CO-2:   Use of the Company's Approved Database Engine.
CO-3:   Conformance to the Relevant Payment Industry Standards
CO-4:   Language and Platform Compatibility
CO-5:   Usability and Accessibility Standards
CO-6:   Integration with Existing Systems
CO-7:   Compliance with Regulatory Requirements
CO-8:   Data Privacy and Security
CO-9:   Maintenance and Supportability

2.5 User Documentation

UD-1:   The system shall provide comprehensive online help documentation accessible from within the application.

UD-2:   The documentation shall include guides for both cashiers and administrators on using the POS Simulator effectively

2.6 Assumptions and Dependencies

AS-1:   The POS Simulator assumes that users have basic computer literacy skills.

AS-2:   The POS Simulator assumes to work on Windows Systems and Google browser.

DE-1:   The functionality of the POS Simulator relies on modifications made to the Payment System to accommodate payment requests related to simulated transactions within the POS Simulator.

DE-2:   The operation of the POS depends on changes being made in the Inventory System to update the availability of items as orders are accepted.

3. System Features

3.1 Process Sales Transactions

3.1.1   Description and Priority
In the POS Simulator, authorized users, primarily cashiers, have the ability to process sales transactions for customers. This feature allows for the efficient and accurate handling of customer purchases using various payment methods. Users can create,

edit, and finalize sales orders, ensuring a seamless shopping experience. Additionally, this functionality includes real-time calculation of transaction totals, including applicable taxes and discounts, ensuring precision and customer satisfaction. This feature is central to the core functionality of the POS Simulator, as it emulates the primary purpose of a real-world point-of-sale system – facilitating sales transactions accurately and efficiently.

3.1.2 Stimulus/Response Sequences

Stimulus: Cashier begins a new sales transaction.
Response: The system initializes a new transaction and awaits product input.

Stimulus: Cashier scans or enters product information.
Response: The system adds the product to the transaction,reduces the stock count, updates the total, and displays the itemized list.

Stimulus: Cashier applies discounts or promotions.
Response: The system adjusts the total price based on the applied discounts provided if any discounts exist.

Stimulus: Cashier finalizes the transaction and selects a payment method.
Response: The system prompts for payment, processes the payment, and generates a receipt.

3.2 Functional Requirements

3.2.1 Retrieve name and price of products

Name	retrieve name and price of product
Description	
Input	Barcode
Output	Name and price of product
Action	Access backend catalog system using barcode, find and retrieve product description
Pre-condition	Valid barcode (consistent with standard + corresponding product exists)
Post-condition	Price and name of product available

3.2.2 Handle payment

Name	Handle payment
Description	
Input	Amount to be paid, cash received

Output	Change to be given
Action	If result ok, Add amount to be paid to total amount of cash
Pre condition	Cash received >= amount to be paid
Post condition	Amount of cash (after payment) = amount of cash (before payment) + amount to be paid

3.2.3 Deduce stock amount

Name	deduce stock amount
Description	
Input	Barcode
Output	Stock amount after this sale
Action	Deduce one from stock amount of product
Pre-condition	Valid code, at least one product in stock
Post-condition	Product.stockAmount_after = Product.stockAmount_before -1

3.2.4 Check Stock Inventory

Name	check stock inventory
Description	
Input	Barcode
Output	stock amount after adding product
Action	adding product to the stock
Pre-condition	Demand for the product should exist along with limited stock
Post-condition	Product.stockamtt_after=Product.stockamtt_before+amt of stock brought

3.2.5 ViewEnd-Of-Day-Sales

Name	view End-of-day sales
Input	date
Output	sales happened on that particular day
Action	Read barcode on coupon , find id of coupon, retrieve name of
Pre-condition	view stock inventory before the sales has been made
Post-condition	view stock inventory after the sales has been mad

4. External Interface Requirements

4.1 User Interfaces

UI-1: The user interface of the POS Simulator should provide a graphical interface for users(cashiers, administrators) to perform various tasks within the system, such as processing sales transactions, managing inventory, and generating reports.

UI-2: The simulator should offer administrative tools and dashboards with features for configuring the system and generating reports. The interface should be designed to accommodate the user responsibilities effectively.

UI-3: Cashiers require a user-friendly interface for quick and efficient transaction processing.

UI-4: POS Simulator should support for various payment methods(cash,credit card, mobile payments)

4.2 Hardware Interfaces

Hardware interfaces for a Point of Sale (POS) Simulator typically involve the physical connections and devices that interact with the software to facilitate sales transactions.

Barcode Scanner:
- A barcode scanner is used to scan product barcodes for quick and accurate item identification.

Receipt Printer:
- A receipt printer is used to print customer receipts, providing a physical record of the transaction.

Payment Terminal (Card Reader):
- A payment terminal or card reader is used to process credit card and debit card payments. It can be connected via USB or other interfaces.

Touchscreen Display:
- A touchscreen display is the primary interface for cashiers to interact with the POS software. It allows for item selection, input, and order processing.

Customer Display:
- A customer display is a secondary display that shows customers the items being rung up and the total cost.

4.3 Software Interfaces

SI-1:   **InventorySystem**

SI-1.1:   The POS shall transmit the quantities of items ordered to the Inventory System through a programmatic interface.

SI-1.2:   The POS Simulator shall query the Inventory System to determine the availability of specific items.

SI-1.3:   When the Inventory System notifies the POS Simulator that an item is out of stock or back in stock, the POS Simulator shall update its menu accordingly.

SI-2:   **Payment Gateway/Processor**
              The POS shall communicate with the Payroll System through a programmatic interface for the following operations:

SI-2.1:   The POS Simulator shall communicate with a Payment Gateway or Processor through a programmatic interface to process various payment methods, such as credit card payments or mobile payments.

SI-2.2:   The POS Simulator shall interact with the Payment Gateway to initiate refunds when necessary, such as for returned items or customer dispute deduction.

SI-3:   **Authentication System**
SI-3.1:   The POS Simulator shall use a programmatic interface to authenticate users, such as cashiers or managers, before allowing access to certain functions or sensitive operations.

SI-4:   **Receipt Generation System**

SI-4.1:   The POS Simulator shall interface with a Receipt Generation System to generate digital or physical receipts for customers after a transaction is completed.

SI-5:   **Accounting Software**

SI-5.1:   The POS Simulator shall provide transaction data to an accounting or financial software system for record-keeping and reporting purposes.

SI-6:   **Reporting and Analytics**

SI-6.1:   The POS Simulator may provide data to a reporting and analytics system for generating sales reports, inventory analysis, and other business insights.

SI-7:   **External APIs:**

SI-7.1:   The POS Simulator may integrate with external APIs for services like online ordering, delivery, or online payment services

4.4   Communications Interfaces

CI-1:   Confirmation Emails

CI-1.1: Confirmation Email to Customer.The POS Simulator shall send an email to the customer to confirm the details of their transaction, including items purchased, total amount, payment method, and transaction timestamp.

CI-1.2: Confirmation Email to Cashier.The POS Simulator shall send an email to the cashier or store personnel to confirm the successful completion of the customer's transaction.

CI-2:   Error and Issue Reporting Emails

CI-2.1: The POS Simulator shall send an email to designated personnel or support staff to report any errors or issues encountered during a transaction, such as payment processing errors or inventory discrepancies.

CI-2.2: Customer Support Email.The POS Simulator shall provide a mechanism for customers to report problems with their transactions via email, and it shall send these reports to a designated customer support email address.

CI-3:   Admin-Cashier Communication

CI-3.1: Emergency Alerts.Administrators shall have the capability to send emergency alerts or urgent messages to cashiers in case of critical situations or issues that require immediate attention.

CI-3.2: Shift Scheduling and Changes.The POS Simulator shall enable administrators to communicate cashier shift schedules and any modifications to the schedule.

CI-3.3: System Updates.Administrators shall use the messaging system to inform cashiers about system updates, maintenance windows, or software changes affecting POS operations.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1:   **Response Time:**

PE-1.1:
    POS Transaction Response Time:The POS Simulator shall respond to cashier actions, such as item scanning, payment processing, and receipt generation, within 2 seconds or less to provide a snappy user experience.

PE-1.2: Inventory Query Response Time: Inventory queries to check item availability shall be completed within 1 second to prevent delays during item scanning.

PE-2: **Transaction Throughput:**

PE-2.1: Transaction Throughput Rate:The POS Simulator shall support a minimum of 30 transactions per minute to accommodate peak business hours without significant performance degradation.

PE-3: **Scalability:**

PE-3.1: Scalability for High-Volume Days:The POS Simulator should be scalable to handle increased transaction volumes during holidays or special events, with a minimum
capacity of 50 transactions per minute.

PE-4: **Load Handling:**

PE-4.1: Load Handling Stress Test:The POS Simulator shall undergo stress testing to ensure it can handle 150% of the expected peak load without system crashes or performance degradation.

PE-5: **Concurrent Users:**

PE-5.1: Simultaneous Users: The POS Simulator shall support at least 10 concurrent cashier sessions to accommodate multiple cashiers working simultaneously during busy periods.

PE-6: **Database Performance:**

PE-6.1: Database Query Time:Database queries related to transaction data retrieval and storage shall execute within 1 second to maintain transaction processing speed.

PE-7: **Error Handling :**

PE-7.1: Error Recovery Time:The POS Simulator shall handle and recover from errors gracefully, with a maximum downtime of 10 seconds in case of a system failure.

PE-8: **Reporting Performance:**

PE-8.1: Reporting Generation Time:The POS Simulator shall generate daily sales reports within 5 minutes of the end of the business day to provide timely business insights.

PE-9: **Data Backup:**

PE-9.1: Data Backup Speed:Backup processes for transaction data shall be completed within 15 minutes of initiation to ensure data integrity and availability.

PE-10: **Offline Mode:**

PE-10.1: Offline Mode Transaction Handling:In the event of a network failure, the POS Simulator shall continue to process transactions in offline mode, with the ability to synchronize data when the network is restored.

5.2 Safety Requirements

SR-1:   The POS Simulator shall be designed and tested to minimize system failures and crashes, crashes, ensuring uninterrupted service during business hours

SR-2:   The system shall implement data validation and verification mechanisms to ensure that transactions and data entry are accurate, preventing errors in financial records

SR-3:   .Implement regular data backup procedures to protect against data loss and provide

mechanisms for data recovery in case of system failures.

SR-4:   .Ensure that users (cashiers and administrators) receive proper training on how to use the POS Simulator safely and effectively to minimize errors and accidents .

SR-5:   The system shall provide clear and user-friendly error messages to guide users in the event of a system error, preventing confusion and potential mistakes.

SR-6:   Define and communicate emergency procedures for system failures or data breaches, including steps to mitigate risks and restore service.

5.3 Security Requirements

SE-1:   The system shall implement role-based access control (RBAC) to restrict access to specific functionalities and data based on user roles (e.g., cashiers, administrators, patrons).

SE-2:   All sensitive data transmitted between the POS Simulator and external systems, such as payment information and personal data, shall be encrypted using industry-standard encryption protocols (e.g., TLS/SSL).

SE-3:   If the POS Simulator handles credit card transactions, it shall comply with PCI DSS standards to protect cardholder data during storage, transmission, and processing.

SE-4:   The system shall have a clear incident reporting process, and all security incidents, breaches, or suspicious activities shall be reported and investigated promptly.

SE-5:   If the POS Simulator integrates with external systems or services (e.g., payment gateways), these third-party systems shall meet security and compliance standards.

SE-6: The system shall have a disaster recovery plan in place to ensure business continuity in case of unexpected events, such as hardware failures or natural disasters.

SE-7:   The POS Simulator shall comply with applicable data protection and privacy regulations, such as GDPR or HIPAA, depending on the nature of the data it handles.

SE-8:   Personnel responsible for using or maintaining the POS Simulator shall receive security training and awareness programs to recognize and mitigate security threats

5.4 Software Quality Attributes

Usability: The POS Simulator should have an intuitive and user-friendly interface to facilitate easy navigation and use by cashiers and other personnel.

Performance: The POS Simulator should respond promptly to user actions, ensuring quick item scanning, transaction processing, and receipt generation.The system should be able to handle increased transaction volumes during peak hours or special events without significant performance degradation.

Reliability: The POS Simulator should be available for use during business hours with minimal downtime. It should have mechanisms in place to recover gracefully from errors or system failures without data loss.

Security: Sensitive data, such as payment information, should be encrypted during transmission and storage.Role-based access control (RBAC) should restrict access to authorized personnel only.The system should undergo regular security testing to identify and address vulnerabilities.

Portability: The POS Simulator should be designed to run on various hardware and operating systems to accommodate different setups.

Maintainability: The system should be designed with modular components to facilitate easy updates, maintenance, and bug fixes.Comprehensive documentation should be available for users and administrators to assist with troubleshooting and system maintenance.

Compatibility: The POS Simulator should be compatible with external systems, such as payment gateways and inventory management systems, to ensure seamless operations.

Compliance: The system should adhere to legal and industry-specific regulations, such as PCI DSS for payment card data security.

Performance Efficiency: The system should use hardware resources efficiently, minimizing CPU and memory usage to ensure optimal performance.

Error Handling:The POS Simulator should handle errors gracefully, providing clear error messages and recovery options to users.

Reporting and Analytics: The system should provide reporting and analytics features to help businesses analyze sales data and make informed decisions

Appendix A: Data Dictionary

user
authentication   user_id+ username+password

user_id   = * integer representing a unique identifier for a user, 8 bit long *

username

                     = * string representing the login username for userauthentication,30
                        character alphanumeric*
password   = * hashed string representing user's password for secure
                        authentication *

role   =   (Customer|Cashier|Administrator)

                     = product id+price+product name+barcode+category+quantity
product
information

product id   = * integer serving as a unique identifier for a product*

price   = * decimal value indicating the price of the product*

product name   = *string representing the name of a product, alphanumeric
                     characters*

barcode   =* string containing a product's barcode or SKU*

category   =(food|clothing|electronics)

quantity   =*integer representing the quantity*
stock_quantity   = * integer indicating the quantity of the product available in stock*

transaction   =transaction id + cashier counter +timestamp +subtotal +payment
                     method + status

transaction id   =*integer serving as a unique identifier for a transaction,20 bit,
alphanumeric   characters   *

cashier counter   = * integer identifying the counter processing the transaction, max 2 bit long*

timestamp   =* A date and time representing when the transaction occurred, format
                    MM/DD/YYYY*

subtotal   =*decimal value indicating the subtotal amount of the transaction*

payment method   = (cash | mobile payments | card)

status   = (Completed | Refunded |Voided)

                 = customerid + name +phone
customer
                  * An integer serving as a unique identifier for a customer.*
customerid

name   = * string representing the customer's f name.*

phone   *integer representing the customer's phone number.*
admin view   = user authentication +report id+ report date + analytics tools

report id   =*Unique identifier for viewing invoices at each counter , alphanumeric
                  characters*

report date   =   *Date for which the sales report is generated, format
                  YYYY/MM/DD*

analytics tools =   (bar graphs | pie charts)