**Software Requirements Specification**

## **For**

**Point of Sales System**

### **Version 0.1**

### 

### **Prepared By**

### **Group Name: Name**

Yasir Zafar 23L-3029

Tawseen Majid 23L-3100

Eishal Fatima 23L-3027

Raza Ali 23L-3021

Ryyan Choudhary 23L-2542

**Instructor:**  Kiran Khurshid

**Course:** Introduction to Software Engineering

**Project Domain:** Finance

**Teaching Assistant**: Areeba Jamil

**Date:** 14th March 2024

# 

# Table of Contents

### Introduction 3

* 1. Product Scope 3

### Overall Description 4

* 1. Product Overview 4
  2. Product Functionality 5
  3. Assumptions and Dependencies 6

### Specific Requirements 6

* 1. Functional Requirements 7
  2. Use Case Model 8
  3. Use Case Specification 9

### Other Non-functional Requirements 20

### **Revsions**

| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| --- | --- | --- | --- |
| Final Draft  No. 1 | Yasir Zafar | First Version of the SRS document, may be subject to minor changes in the future | 13/03/24 |

# 

# 1. Introduction

Welcome to the Software Requirements Specification (SRS) for the Point of Sale System developed in C++, utilising the Qt framework and SQL. This document provides a comprehensive overview of the project, outlining its scope, objectives, and benefits.

## 1.1 Product Scope

The Point of Sales System, is a comprehensive solution designed to revolutionise businesses' sales operations. With a focus on optimising efficiency and providing valuable features, the system offers a myriad of benefits to its users.

### Benefits

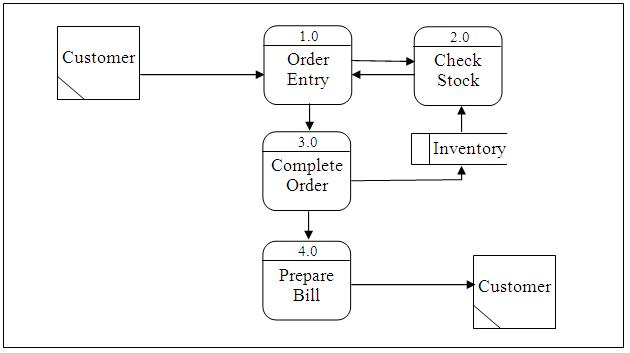
* **Seamless Transaction Processing:** The system ensures swift and error-free transactions, minimising waiting times for customers and reducing the likelihood of errors in order processing.
* **Dynamic Inventory Management:** Real-time updates and intelligent forecasting contribute to optimal inventory levels, preventing stockouts and minimising excess stock. This dynamic approach fosters better resource utilisation.
* **Advanced Reporting and Analytics:** In-depth reporting tools provide actionable insights into sales trends, customer behaviour, and product performance. This empowers businesses to make informed decisions, identify growth opportunities, and optimise their strategies.
* **Customer Relationship Management (CRM):** The system incorporates CRM functionalities, enabling businesses to build and nurture customer relationships. Personalised promotions, loyalty programs, and targeted communication enhance customer satisfaction and retention.
* **Multi-Platform Accessibility:** The systems’ compatibility across various platforms ensures flexibility in its usage, allowing businesses to adapt to changing environments and user preferences.

# 2. Overall Description

## 2.1 Product Overview

Our Point of Sales (POS) system is a cutting-edge addition to the realm of retail solutions, acting as a standalone product designed to revolutionise sales operations. This innovative software is not merely a successor but a pioneer in its own right, introducing advanced features and capabilities that set it apart in the market.

The POS system is positioned as a key component within the broader landscape of retail management systems. It functions as a self-contained entity, adept at seamlessly integrating with existing systems or operating independently. To illustrate its interaction with the environment, consider a simplified diagram



## 2.2 Product Functionality

* **Transaction Processing:** Facilitate seamless and accurate processing of sales transactions.
* **Inventory Management:** Provide real-time tracking and management of inventory levels.
* **Reporting and Analytics:** Generate comprehensive reports and analytics for informed decision-making.
* **Customer Relationship Management (CRM):** Implement CRM functionalities for personalised customer interactions.
* **Multi-Platform Accessibility:** Ensure compatibility across various platforms for flexibility in usage.
* **Database Integration:** Integrate with a robust database for efficient data storage and retrieval.
* **User Authentication and Authorization:** Implement secure user authentication and authorization mechanisms.
* **Product Search and Information Retrieval:** Enable users to search for products and retrieve relevant information efficiently.
* **Sales History Tracking:** Maintain a history of sales transactions for auditing and analysis purposes.

## 2.3 Assumptions and Dependencies

### Assumptions:

1. **Stable Operating Environment:** The successful operation of the system assumes a stable and reliable computing environment, free from frequent disruptions or hardware failures.
2. **User Training:** It is assumed that end-users will undergo sufficient training to familiarise themselves with the POS system's functionalities and operation.
3. **Data Integrity:** The system assumes that the data inputted by users is accurate and valid, ensuring the integrity of the information processed by the POS system.

### Dependencies:

1. **Database Management System (DBMS):** The project is contingent on the availability and compatibility of the designated SQL DBMS, ensuring efficient data storage, retrieval, and management with adherence to SQL standards.
2. **Development Libraries and Frameworks:** The system relies on the correct functioning and availability of the selected development libraries and frameworks, especially those related to Qt and C++, for the development and execution of the application.
3. **Hardware Components:** The system is dependent on the availability and proper functioning of hardware components such as servers, workstations, and peripheral devices, for optimal performance and functionality.

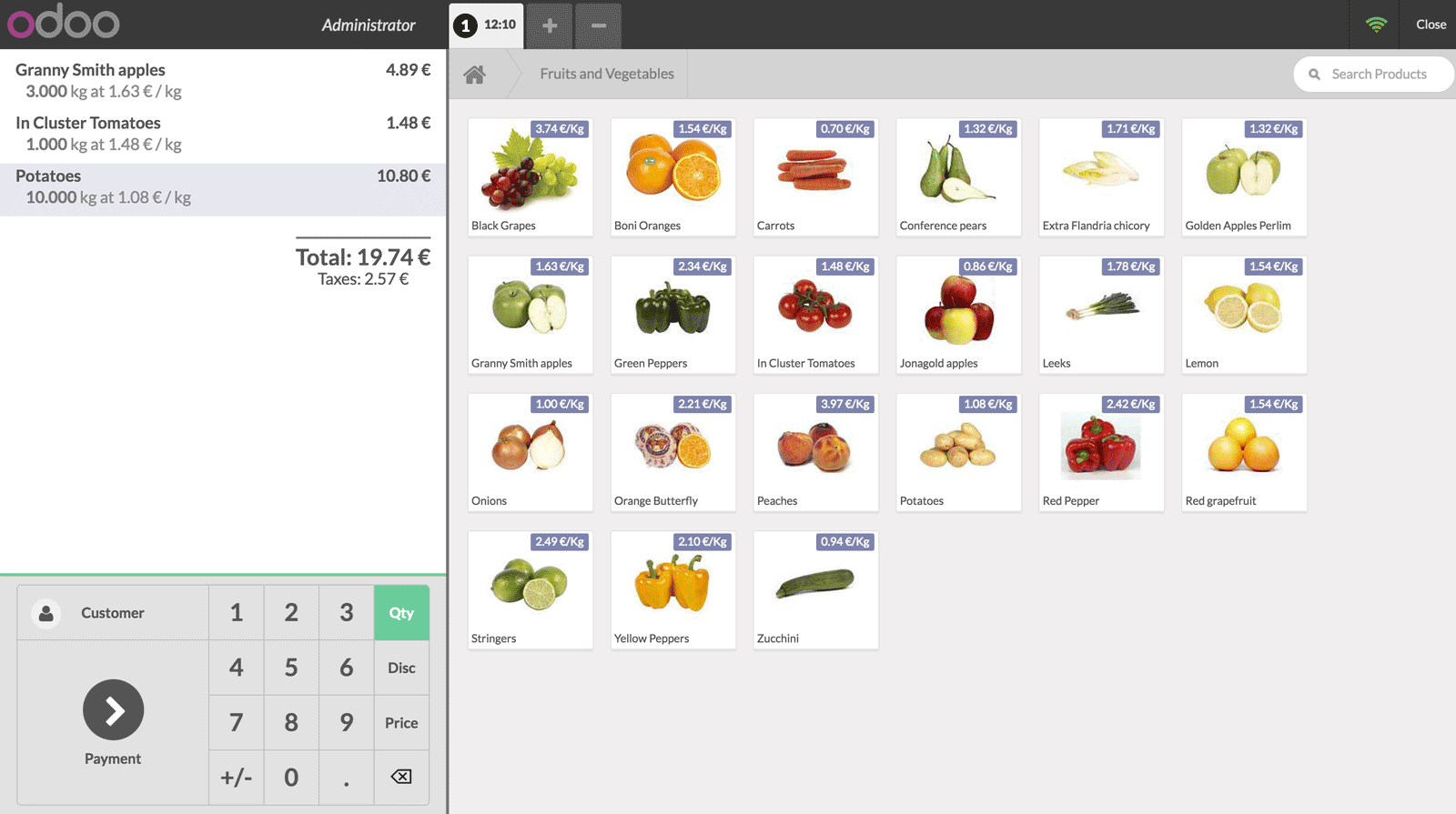
# 3 Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The Point of Sales (POS) system interface is optimised for standard mouse and keyboard navigation, ensuring practical and efficient interactions. Key features include:

* **Mouse Navigation:** Users point and click to navigate, select products, and process transactions swiftly.
* **Keyboard Input:** Convenient data entry for quantities, pricing adjustments, and customer details using a standard keyboard.
* **Menu Navigation:** Logically organized menus with dropdowns, sliders, and input fields for efficient user interaction.
* **Visual Feedback:** Immediate confirmation through highlights, messages, and colour-coded indicators for successful actions.
* **Efficient Controls:** Streamlined interface design prioritizes user control and responsiveness during POS operations.



## 3.2 Functional Requirements

#### 3.2.1 F1: Transaction Processing

* The system shall facilitate the initiation, processing, and completion of sales transactions, ensuring accuracy and efficiency.

#### 3.2.2 F2: Inventory Management

* The system shall provide real-time tracking and management of inventory levels, including additions, removals, and updates to the product catalogue.

#### 3.2.3 F3: Reporting and Analytics

* The system shall generate comprehensive reports and analytics, allowing users to analyse sales trends, track product performance, and make informed business decisions.

#### 3.2.4 F4: Customer Relationship Management (CRM)

* The system shall incorporate CRM functionalities, enabling businesses to manage customer information, track interactions, and implement personalised promotions.

#### 3.2.6 F5: Database Integration

* The system shall integrate seamlessly with a SQL-based Database Management System (DBMS) for efficient storage, retrieval, and management of transaction and inventory data.

#### 3.2.8 F6: Product Search and Information Retrieval

* The system shall enable users to search for products using various criteria, providing relevant information such as pricing, availability, and detailed product specifications.

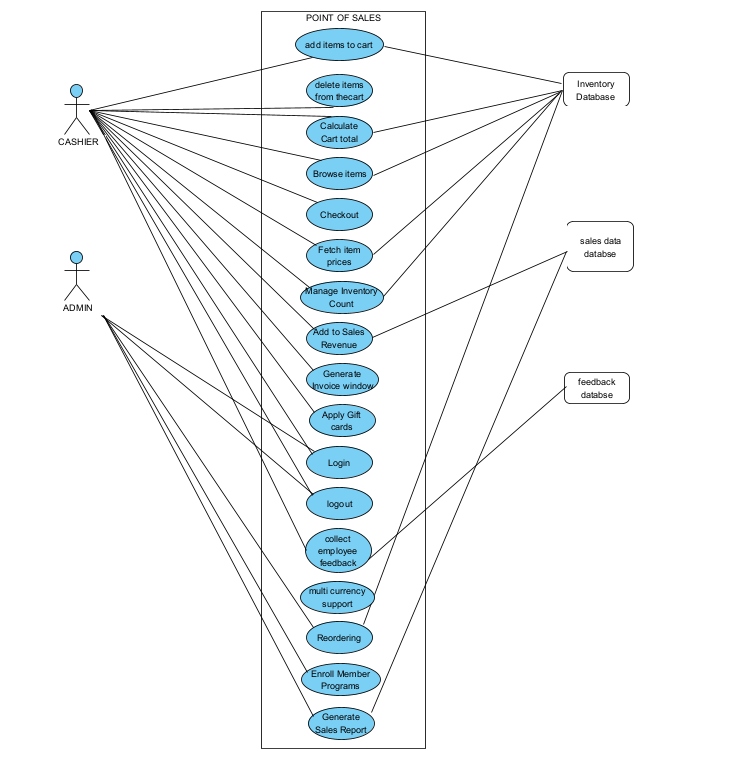
#### 3.2.9 F7: Sales History Tracking

* The system shall maintain a comprehensive history of sales transactions, allowing for auditing, analysis, and historical reporting.

#### 3.2.10 F8: User Management

* The system shall allow administrators to manage user accounts, including the creation, modification, and deactivation of user profiles.

## 3.3 Use Case Model



### 3.3.1 UC1 - Adding Items to Cart

**Author:** Ryyan Choudhary

**Purpose:** Selecting an item and adding it to the list of items in the cart.

**Actors:**

* Cashier

**Preconditions:**

* Admin or cashier has successfully logged in.
* The list of items is opened.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Admin or cashier successfully logs into the system.*
2. **Open List of Items:**
   * *The actor opens the list of items available for selection.*
3. **Select and Add to Cart:**
   * *The actor selects an item from the list and adds it to the cart.*

**Post Conditions:**

* The cart now exists with items added.

### 3.3.2 UC2 - Checking Out

**Author:** Ryyan Choudhary

**Purpose:** Finalise purchase to pay for items.

**Actors:**

* Cashier

**Preconditions:**

* Admin or cashier has successfully logged in.
* Items are selected and added to the cart.
* The billing screen is opened.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Admin or cashier successfully logs into the system.*
2. **Select Items to Add to Cart:**
   * *The actor selects items to add to the cart.*
3. **Open Billing:**
   * *The actor opens the billing screen to finalise the purchase.*

**Post Conditions:**

* Billing and payment can now go through.

### 3.3.3 UC3 - Browsing Items

**Author:** Ryyan Choudhary

**Purpose:** Being able to search for keywords or browse the list of items so we can add them to the cart.

**Actors:**

* Cashier

**Preconditions:**

* Admin or cashier has successfully logged in.
* The items menu is opened.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Admin or cashier successfully logs into the system.*
2. **Open Items Menu:**
   * *The actor opens the items menu to browse the list.*
3. **Look for Items:**
   * *The actor searches for items using keywords or browses the list to find specific items.*

**Post Conditions:**

* Specific items can now be found.

### 3.3.4 UC4 - Deleting Items from Cart

**Author:** Ryyan Choudhary

**Purpose:** Being able to add or remove items if in a cart.

**Actors:**

* Cashier

**Preconditions:**

* Admin or cashier has successfully logged in.
* The cart has at least one item in it.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Admin or cashier successfully logs into the system.*
2. **Open Items Menu:**
   * *The actor opens the items menu to add items to the cart.*
3. **Add Items to Cart:**
   * *The actor adds items to the cart.*
4. **Remove an Item:**
   * *The actor removes an item from the cart.*

**Post Conditions:**

* Unneeded items will be removed from the cart.

### 3.3.5 UC5 - Deleting Items from Cart

**Author:** Ryyan Choudhary

**Purpose:** Being able to add or remove items if in a cart.

**Actors:**

* Cashier

**Preconditions:**

* Admin or cashier has successfully logged in.
* The cart has at least one item in it.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Admin or cashier successfully logs into the system.*
2. **Open Items Menu:**
   * *The actor opens the items menu to add items to the cart.*
3. **Add Items to Cart:**
   * *The actor adds items to the cart.*
4. **Remove an Item:**
   * *The actor removes an item from the cart.*

**Post Conditions:**

* Unneeded items will be removed from the cart.

### 

### 3.3.6 UC6 - Fetch Item Prices

**Author:** Yasir Zafar

**Purpose:** The primary objective of this use case is to retrieve the prices of items within the Point of Sales (POS) system, enabling accurate transaction processing and providing up-to-date information to actors.

**Actors:**

* Cashier

**Preconditions:**

* The system is initialised and operational.
* The cashier is logged into the POS system.

**Flow of Events - Basic Flow:**

1. **Cashier Requests Item Prices:**
   * *Cashier initiates the request for item prices from the system.*
2. **System Retrieves Prices:**
   * *The POS system communicates with the Inventory System to fetch the current prices of items.*
3. **Display Item Prices:**
   * *The POS system displays the retrieved item prices on the actor interface for the cashier.*

**Post Conditions:**

* Item prices are successfully displayed on the POS interface.
* The cashier has access to updated and accurate pricing information for all items.

### 3.3.7 UC7 - Apply Discount

**Author:** Yasir Zafar

**Purpose:** This use case aims to apply discounts to items during a transaction, facilitating promotional offers or adjustments in pricing.

**Actors:**

* Cashier

**Preconditions:**

* The system is initialised and operational.
* The cashier is logged into the POS system.
* Items are added to the transaction.

**Flow of Events - Basic Flow:**

1. **Cashier Requests Discount:**
   * *Cashier initiates the request for applying a discount to specific items in the transaction.*
2. **System Retrieves Discount Details:**
   * *The POS system communicates with the Discount Management System to fetch applicable discount details.*
3. **Apply Discount to Items:**
   * *The POS system applies the relevant discount to the selected items in the transaction.*
4. **Update Transaction Total:**
   * *The system recalculates the total amount, reflecting the applied discount.*

**Post Conditions:**

* The discount is successfully applied to the selected items in the transaction.
* The transaction total is updated to reflect the discounted amount.

### 3.3.8 UC8 - Generate Sales Report

**Author:** Yasir Zafar

**Purpose:** The primary objective of this use case is to generate a comprehensive sales report, providing valuable insights into transaction trends and performance.

**Actors:**

* Admin

**Preconditions:**

* The system is initialised and operational.
* The admin is logged into the POS system.
* Previous transactions are stored in the system.

**Flow of Events - Basic Flow:**

1. **Admin Initiates Report Generation:**
   * *Admin triggers the request to generate a sales report from the system.*
2. **System Retrieves Sales Data:**
   * *The POS system communicates with the Reporting System to fetch relevant sales data.*
3. **Generate and Display Sales Report:**
   * *The POS system processes the sales data and generates a comprehensive report, displaying it on the admin's interface.*

**Post Conditions:**

* The sales report is successfully generated and displayed on the admin's interface.
* The admin has access to insights into transaction trends and performance.

### 3.3.9 UC9 - Enrol Member Program

**Author:** Yasir Zafar

**Purpose:** This use case is designed to enrol customers into the membership program, providing them with benefits and enhancing customer engagement.

**Actors:**

* Cashier

**Preconditions:**

* The system is initialised and operational.
* The cashier is logged into the POS system.

**Flow of Events - Basic Flow:**

1. **Cashier Initiates Member Enrollment:**
   * *Cashier triggers the request to enrol a customer into the membership program.*
2. **System Collects Customer Information:**
   * *The POS system prompts the cashier to collect necessary customer information for enrollment.*
3. **Communicate with the Membership System:**
   * *The POS system communicates with the Membership System to enrol the customer and obtain membership benefits.*
4. **Provide Membership Information:**
   * *The system provides the cashier with membership details to share with the customer.*

**Post Conditions:**

* The customer is successfully enrolled in the membership program.
* The cashier has access to membership information to share with the customer.

### 3.3.10 UC10 - Manage Inventory Count

**Author:** Tawseen Majid

**Purpose:** Manage and manipulate the inventory count of every product after a completed billing.

**Actors:**

* Cashier

**Preconditions:**

* The actor adds the products they want to purchase in the cart and proceeds to checkout.

**Flow of Events - Basic Flow:**

1. **Search for Products:**
   * *Once the actor proceeds to checkout, each product in the cart is searched for in the inventory database.*
2. **Fetch Inventory Count:**
   * *The matching inventory count is fetched from the database.*
3. **Update Inventory Count:**
   * *The required value is subtracted from the inventory count to give the new quantity of the item in the inventory.*

**Post Conditions:**

* The updated inventory count for the respective items is appended to the inventory database.
* A prompt for reordering is generated if the quantity falls below a certain point.

### 3.3.11 UC11 - Add to Sales Revenue

**Author:** Tawseen Majid

**Purpose:** Make necessary changes to the total sales revenue after each billing.

**Actors:**

* Cashier

**Preconditions:**

* The actor adds the products they want to purchase in the cart and proceeds to checkout.

**Flow of Events - Basic Flow:**

1. **Calculate Total Billing Amount:**
   * *Once the actor proceeds to checkout, the total amount of the bill is calculated by adding the prices of every added item together with their desired quantity.*
2. **Fetch Sales Revenue:**
   * *The total revenue is fetched from the sales data database.*
3. **Update Sales Revenue:**
   * *The calculated billing amount is added to give the new sales revenue after every sale.*

**Post Conditions:**

* The updated sales revenue is appended to the sales data database.
* A flag is activated once the sales target is achieved, adding to the discount of all following orders.

### 3.3.12 UC12 - Reordering Products

**Author:** Tawseen Majid

**Purpose:** Send a prompt for reordering of an item after its quantity falls below a certain level.

**Actors:**

* Cashier

**Preconditions:**

* The inventory counts of each item are updated after every completed billing.

**Flow of Events - Basic Flow:**

1. **Check Inventory Counts:**
   * *After each update in the item inventory counts, if the updated count falls below a certain level, a prompt for reordering is generated.*

**Post Conditions:**

* For as long as the reorder flag remains activated, after each billing containing the flagged product, the admin is prompted to reorder.
* If reordered, update the restocked value in the inventory database.

### 3.3.13 UC13 - Generate Invoice Window

**Author:** Tawseen Majid

**Purpose:** Generate the summary invoice for the cart.

**Actors:**

* Cashier

**Preconditions:**

* Actor adds items to the cart and checks out.

**Flow of Events - Basic Flow:**

1. **Fetch Individual Prices:**
   * *The individual prices of added items are fetched.*
2. **Calculate Total Invoice:**
   * *The total invoice is generated with the total both inclusive and exclusive of tax.*

**Post Conditions:**

* The next billing session is initiated.

### 3.3.14 UC14 - Logging In/Out

**Author:** Raza Ali

**Purpose:** Depicting whether the actor is an admin or a cashier to differentiate the function’s accessibility.

**Actors:**

* Admin
* Cashiers

**Preconditions:**

* App is open.

**Flow of Events - Basic Flow:**

1. **App Opened:**
   * *The application is opened.*
2. **Login Screen Shown:**
   * *The actor is directed to the login screen.*

**Post Conditions:**

* Actors logged in as either Admin or Cashier.

### 3.3.15 UC15 - Applying Gift Cards

**Author:** Raza Ali

**Purpose:** Applying Discount Code to apply a discount.

**Actors:**

* Cashier

**Preconditions:**

* Cashier is logged in.
* List of Items is open.

**Flow of Events - Basic Flow:**

1. **Login:**
   * *Cashier successfully logs into the system.*
2. **Open Cart:**
   * *The actor opens the cart.*
3. **Press Apply Gift Card:**
   * *Cashier presses the option to apply for a gift card.*

**Post Conditions:**

* Discount is added.

### 3.3.16 UC16 - Collect Employee Feedback

**Author:** Eishal Fatima

**Purpose:** The primary objective of this use case is to collect cashier feedback within the Point of Sale (POS) system, enabling businesses to gather insights and improve cashier satisfaction.

**Actors:**

* Cashier

**Preconditions:**

* The system is initialized and operational.
* The cashier is logged into the POS system.

**Flow of Events - Basic Flow:**

1. **Cashier Provides Feedback:**
   * *After completing a transaction, the cashier prompts the customer to provide feedback on their experience.*
2. **System Records Feedback:**
   * *The POS system records the feedback provided by the customer.*
   * *Feedback may include ratings, comments, or specific suggestions.*
3. **Store Feedback Data:**
   * *The POS system stores the collected feedback data in the database for further analysis.*

**Post Conditions:**

* Cashier feedback is successfully recorded in the POS system.
* The business can use customer feedback to identify areas for improvement and enhance overall customer satisfaction.

### 3.3.17 UC17 - Provide Multi-Currency Support

**Author:** Eishal Fatima

**Purpose:** The primary objective of this use case is to enable transactions in multiple currencies within the Point of Sales (POS) system, accommodating international customers and providing accurate pricing conversions.

**Actors:**

* Employee

**Preconditions:**

* The system is initialised and operational.
* The employee is logged into the POS system.
* The employee requests to view and select items for purchase in the standard currency, to have the total bill displayed in a different currency.

**Flow of Events - Basic Flow:**

1. **Cashier Requests Multi-Currency Display:**
   * *The customer informs the cashier of their desire to transact in a different currency.*
2. **Cashier Enables Multi-Currency Display:**
   * *The cashier selects the option within the POS system to enable multi-currency display for the customer's transaction.*
3. **System Retrieves Exchange Rates:**
   * *The POS system communicates with an external exchange rate service or database to fetch the current exchange rates for the currency selected by the customer.*
4. **System Converts Prices and Total Bill:**
   * *As the cashier continues to add items to the shopping cart, the POS system automatically converts the prices of each item and calculates the total bill based on the selected currency and retrieved exchange rates.*

**Post Conditions:**

* The transaction is completed with the shopping cart contents and total bill displayed in the currency the customer selects
* The cashier receives a receipt showing the transaction details, including the total bill, in the currency the customer selects.

# 

# 4 Other Non-functional Requirements

## 4.1 Performance Requirements

**4.1.1 Response Time:**

* The system shall respond to user interactions within 2 seconds, ensuring prompt and efficient user engagement.

**4.1.2 Database Query Performance:**

* Database queries shall execute within 5 seconds to maintain responsive interactions and real-time data access.

## 4.2 Usability Requirements

**4.2.1 Intuitive User Interface:**

* The user interface shall be intuitive, minimising the learning curve for new users.

**4.2.1 Error Handling:**

* Error messages shall be clear and concise, guiding users to understand and resolve issues promptly.

## 4.4 Reliability Requirements

**4.4.1 System Uptime:**

* The system shall have a minimum uptime of 99.9%, ensuring continuous availability for users.

**4.4.2 Data Integrity:**

* Data stored in the system shall have a backup and recovery mechanism to prevent data loss and ensure reliability.