**Software Design and Architecture**

Project Deliverable 2



Saim Alam 21I-1210

Sami Irshad 21I-1103

Abad Naseer 20I-1815

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| CafeConnect | 04-03-2023 | - | 1.0 |
| CafeConnect | 20-03-2023 | Advised by TA | 1.1 |

Contents

[Work Division Table 3](#_Toc130229562)

[Project Scope 3](#_Toc130229563)

[Actor-Goal List 4](#_Toc130229564)

[Use Case Diagram 5](#_Toc130229565)

[High Level Use Cases 6](#_Toc130229566)

[Extended Use Cases 10](#_Toc130229567)

[Wireframes in Figma 29](#_Toc130229568)

[GitHub Repository 29](#_Toc130229569)

## Work Division Table

|  |  |
| --- | --- |
| **Name** | **Contribution** |
| Saim Alam | Scope, Actor-Goal List, Use Cases for Processing Returns, Placing Orders, Account Handling, Returning Food, Analyzing Sales, Generating Daily Receipt, Handling Student Information, and Managing the Administrators, Extended Use Cases for Place Order, Analyze Sales, Generate Receipt |
| Sami Irshad | Scope, Use Cases for Processing Sales, Tax Deduction, and Paying for Food, Extended Use Cases for Process Sales, Process Returns, Manage Cashiers |
| Abad Naseer | Use Cases for Paying Employees, Updating Inventory, and Paying Rent, Extended Use Cases for Handle Student, Handle Account, Pay Employees |

## Project Scope

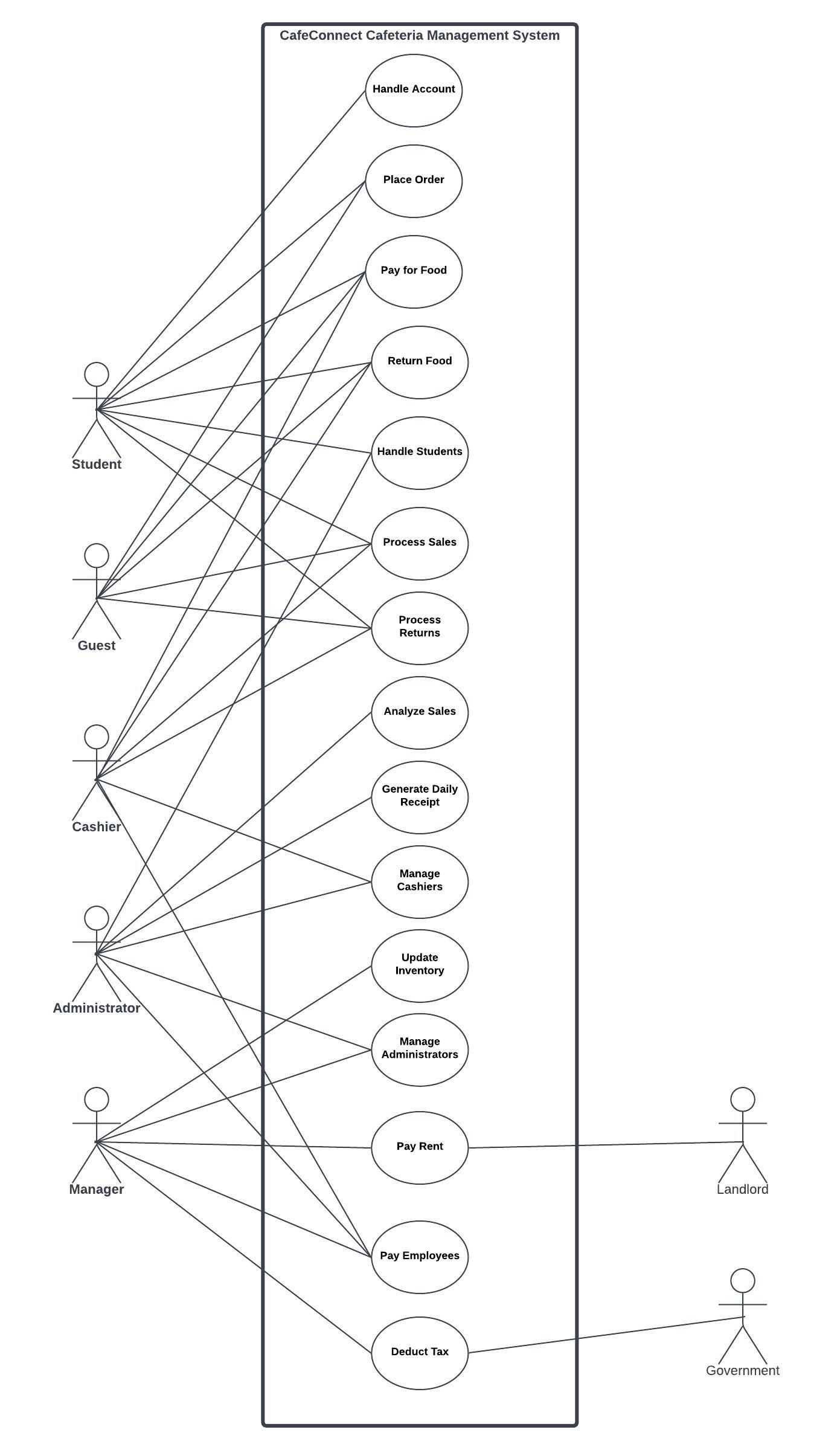
The scope of the proposed cafeteria management project is to develop a Java/JavaFX-based GUI system that will allow students, guests, cashiers, administrators, and a manager to interact with the system. The system will enable students and guests to order food, manage their accounts, pay for food, and return food items. The cashier will be able to process sales, handle returns, and manage cash in and cash out. The administrator will be able to manage the cashiers, analyze sales, generate daily sales receipts, and view inventory. The manager will be able to manage the inventory, the administrators' information, pay rent, and pay both the cashiers and the administrators. The system's key use cases will include managing cashiers, processing sales and returns, analyzing sales, updating inventory, generating daily receipts, handling student information, managing administrators, paying rent, and paying employees. The system will be developed to provide an efficient, reliable, and user-friendly interface that will meet the needs of all the actors involved in the cafeteria management process.

## Actor-Goal List

|  |  |
| --- | --- |
| Actor | Goal |
| Student | Place an order for food  Manage their account/profile  Return food item  Pay for food |
| Guest | Place an order for food  Pay for food  Return food item |
| Cashier | Process sales  Handle returns  Cash in and cash out |
| Administrator | Manage cashiers (add/modify/delete)  Manage student details (add/modify/delete)  Analyze sale(s)  Generate daily sale receipt  View inventory |
| Manager | Manage inventory  Manage administrator information (add/modify/delete)  Pay rent  Pay employees |
| Landlord | Receive rent |
| Government | Collect tax on sale of items |

Each actor has their own specific goals they want to achieve using the system, which highlights the importance of having a comprehensive cafeteria management system that caters to the needs of all stakeholders. By identifying and addressing these needs, the cafeteria management system can improve overall efficiency, accuracy, and customer satisfaction.

## Use Case Diagram



## High Level Use Cases

|  |  |
| --- | --- |
| High Level Use case: Manage Cashiers | |
| Use Case | Manage Cashiers |
| Actors | Administrator, Cashier |
| Type | Secondary |
| Description | The Administrator manages the Cashiers by adding, modifying, or deleting their information in the system based on their performance. |

|  |  |
| --- | --- |
| High Level Use case: Process Sales | |
| Use Case | Process Sales |
| Actors | Student, Guest, Cashier |
| Type | Primary, Primary, Secondary |
| Description | The Cashier processes a sale by recording the items purchased by a Student or Guest and calculating the total cost, then accepting payment and providing change. |
| High Level Use case: Process Returns | |
| Use Case | Process Returns |
| Actors | Student, Guest, Cashier |
| Type | Primary, Primary, Secondary |
| Description | The Cashier processes a return by accepting returned items, verifying the reason for the return, and issuing a refund or credit. |
| High Level Use case: Place Order | |
| Use Case | Place Order |
| Actors | Student, Guest |
| Type | Primary |
| Description | The Student or Guest places an order for food by selecting items from the menu, specifying quantities, and providing payment. |
| High Level Use case: Account Handling | |
| Use Case | Account Handling |
| Actors | Student |
| Type | Primary |
| Description | The Student organizes and handles their account by updating their personal details, viewing their purchase history, and checking their account balance. |
| High Level Use case: Pay for Food | |
| Use Case | Pay for Food |
| Actors | Student, Guest |
| Type | Primary |
| Description | The Student or Guest pays for their food to the cashier by providing payment in the form of cash or other accepted payment methods. |
| High Level Use case: Return Food | |
| Use Case | Return Food |
| Actors | Student, Guest |
| Type | Primary |
| Description | The Student or Guest returns food to the cashier by providing a reason for the return, verifying the purchase details, and receiving a refund or credit. |
| High Level Use case: Analyze Sales | |
| Use Case | Analyze Sales |
| Actors | Administrator |
| Type | Secondary |
| Description | The Administrator analyzes sales by generating reports that summarize the total sales, individual sales, and sales by item, student, or guest. |
| High Level Use case: Update Inventory | |
| Use Case | Update Inventory |
| Actors | Manager |
| Type | Secondary |
| Description | The Manager updates the inventory by adding or removing items, updating quantities, and checking inventory levels to ensure adequate stock. |
| High Level Use case: Generate Daily Receipt | |
| Use Case | Generate Daily Receipt |
| Actors | Administrator |
| Type | Secondary |
| Description | The Administrator generates a daily receipt by compiling a summary of all the sales made in the day, including item descriptions, quantities, and total cost. |
| High Level Use case: Handle Student Information | |
| Use Case | Handle Student Information |
| Actors | Administrator, Student |
| Type | Secondary, Primary |
| Description | The Administrator handles student information by adding, modifying, or deleting their details, such as name, ID number, and contact information. |
| High Level Use case: Manage Administrators | |
| Use Case | Manage Administrators |
| Actors | Manager, Administrator |
| Type | Secondary |
| Description | The Manager manages Administrators by adding, modifying, or deleting their information in the system. |
| High Level Use case: Pay Rent | |
| Use Case | Pay Rent |
| Actors | Manager, Landlord |
| Type | Secondary, Offstage |
| Description | The Manager pays rent by making the necessary payments to the landlord or property owner according to the terms of the rental agreement. |
| High Level Use case: Pay Employees | |
| Use Case | Pay Employees |
| Actors | Manager, Cashier, Administrator |
| Type | Secondary |
| Description | The Manager pays employees by making the necessary payments to the Cashiers and Administrators according to their salaries and payment schedules. |
| High Level Use case: Tax Deduction | |
| Use Case | Tax Deduction |
| Actors | Manager, Government |
| Type | Secondary, Offstage |
| Description | The government of Pakistan has decided to tax a specific percentage of the cost of a product. The Manager pays the government with the tax amount accumulated from the products over the month. |

## Extended Use Cases

|  |  |
| --- | --- |
| Use Case Name | Place Order |
| Scope | CafeConnect Cafeteria Management System |
| Level | User Goal |
| Primary Actor | Student/Guest |
| Stakeholders and Interests | * Student/Guest   Want to order food quickly and easily  Cashier  Wants to receive accurate and complete order information   * Administrator   Wants to manage inventory and analyze sales data |
| Preconditions | • The user is logged in to the system.  • The user has selected the "Place Order" option from the main menu.  • The user has a valid payment method associated with their account. |
| Post conditions | • The order has been processed and recorded in the system.  • The inventory has been updated to reflect the items sold.  • The user has received a confirmation of their order. |
| Main Success Scenario | |  |  | | --- | --- | | Actor Action | System response | | 1. The user selects the "Place Order" option from the main menu. |  | |  | 2. The system displays a menu of available items for the user to select from | | 3. The user selects the items they want to order and adds them to their cart. |  | |  | 4. The system displays the user's order summary, including the items, quantities, and total cost. | | 5. The user reviews their order and makes any necessary changes or corrections. |  | | 6. The user selects their payment method and confirms their order. |  | |  | 7. The system processes the payment and records the order in the system. | |  | 8. The system updates the inventory to reflect the items sold. | |  | 9. The system generates and displays order confirmation message | |
| Extensions | • If the user attempts to order an item that is out of stock, the system will notify them and prompt them to make a different selection.  • If the user encounters any issues with their payment method, the system will prompt them to select a different payment method or contact customer support for assistance.  • If the user needs to make changes or cancel their order after it has been placed, they can do so within a specified timeframe before the order is processed and prepared by the Cashier. |

|  |  |
| --- | --- |
| Use Case Name | Analyze Sales |
| Scope | CafeConnect Cafeteria Management System |
| Level | Subfunction |
| Primary Actor | Administrator |
| Stakeholders and Interests | * Administrator   Wants to analyze sales data to make informed decisions about inventory, promotions, and pricing. |
| Preconditions | The administrator has logged in to the system  The administrator has access to sales data and made informed decisions based on the analysis |
| Post conditions | • The Administrator has analyzed the sales data and made informed decisions based on the analysis. |
| Main Success Scenario | |  |  | | --- | --- | | Actor Action | System Response | | 1. The Administrator selects the "Analyze Sales" option from the main menu. |  | |  | 2. The system displays a dashboard of sales data for the specified time period, including total sales, average sales per day, and sales by item or category. | | 3. The Administrator can filter and sort the data to focus on specific items or time periods. |  | |  | 4. The system generates visualizations of the sales data, such as graphs or charts, to help the Administrator identify trends and patterns. | | 5. The Administrator can use the sales data to make informed decisions about inventory management, promotions, and pricing. |  | |
| Extensions | • If the sales data is incomplete or inaccurate, the Administrator can request additional information or corrections from the Cashiers or other relevant stakeholders.  • If the sales data reveals unexpected or anomalous patterns, the Administrator can investigate further to identify the root causes and take appropriate action.  • If the Administrator needs to share the sales data with other stakeholders, such as the Manager or the Cashiers, they can generate and distribute reports or visualizations as needed. |

|  |  |
| --- | --- |
| Use Case Name | Generate Receipt |
| Scope | CafeConnect Cafeteria Management System |
| Level | Sub function |
| Primary Actor | Administrator |
| Stakeholders and Interests | * Administrator   Wants to generate accurate daily receipts for accounting purposes   * Cashier   Wants to ensure that sales are accurately recorded and processed |
| Preconditions | • The system has recorded all sales transactions for the day.  • The Administrator has logged in to the system. |
| Post conditions | • A daily sales receipt has been generated and saved in the system. |
| Main Success Scenario | |  |  | | --- | --- | | Actor Action | System Response | | The Administrator selects the "Generate Daily Receipt" option from the main menu. |  | |  | The system retrieves all sales transactions for the day and displays them in a report. | | The Administrator reviews the report for accuracy and completeness. |  | |  | The system generates a daily sales receipt based on the report and saves it in the system. | | The Administrator reviews the daily sales receipt to ensure that it accurately reflects the day's sales transactions. |  | | The Administrator prints the daily sales receipt and files it for accounting purposes. |  | |
| Extensions | • If the Administrator finds any errors or discrepancies in the sales transactions, they can modify or correct the transactions before generating the daily sales receipt.  • If the system encounters any errors or issues while generating the daily sales receipt, it will notify the Administrator and prompt them to resolve the issue before generating the receipt. |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Handle Account | |
| Scope: | CafeConnect Cafeteria Management System | |
| Level: | User Goal | |
| Primary Actor: | User | |
| Stakeholders and interests: | -  - | **User: User wants to handle his/her account to get update his profile and account setting.**  **Cashier: wants to see the get the matching profile of the user.** |
| Preconditions: | 1. 2. | User is successfully Logged in.  System is connected to the CafeConnect Cafeteria Management System. |
| Postconditions: | 1. 2. | Users': are successfully updated his profile.  Users’: are successfully updated his account information. |
|  | 3. Users’ complete account are successfully updated. | |
| Main Success Scenario: | |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Actor Action | | | | System Response | | | | | | 1. | | | Admin starts updating his/her profile. | |  | | |  | | 2. | | | User selects his/her Handle Account info. | |  | | |  | |  | | |  | | 3. | | | User is provided with profile and account information.  . | | 4. | | | User selects profile. | |  | | |  | |  | | |  | | 5. | | | User is provided with his name, address and CNIC. | | 6. | | | User enters his details. | |  | | |  | | 7. User selects “Update”. | | | | |  | | |  | |  | | | | |  | | |  | |  | | | | | 8. | | | User is updated with his name, contact, address and CNIC by the system. | | 9. User selects his account setting. | | | | |  | | |  | |  | | | | | 10. | | | User is provided with user id, password. | | 11. User selects change password. | | | | |  | | |  | | 12. User enters new password and re-enters his/her new password. | | | | |  | | | . | |  | | | | | 13. System verifies the new password with re-entered password. | | |  | | 14. User selects update account info. | | | | |  | | |  | |  | | | 15. | | | System updates the user account and. Returns a success message to the User. | | | | 16. User goes back to the dashboard. | | |  | | |  | | | | |
| Extensions: | 1. If the User does not exist:    1. System will return an error message.    2. User tries with correct ID. 2. If the user tries to enters details (Name, CNIC) in wrong format:    1. User is given an error message.    2. System displays correct format for entries. 3. If the user password does not match with the new password:    1. User is notified with the wrong error message “New password does not match conform password”.   System enters correct password. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Handle students | |
| Scope: | CafeConnect Cafeteria Management System | |
| Level: | User Goal | |
| Primary Actor: | 1. | Administrator 2. Student |
| Stakeholders and interests: | **-** | **Admin:** wants to add/delete/modify students in the system. He wants to make sure every student has successfully added/deleted/modified. He/she enters his/her correct information like name, CNIC, ID. |
|  | **-** | **Student:** wants to see his account through login. He/she want to make sure his/her has correct information and details. |
| Preconditions: | 1. 2. | Administrator is successfully Logged in.  Actor is connected to the User Management System. |
| Postconditions: | 1. | Student is added in the system. |
| Main Success Scenario: | |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Actor Action | | | System Response | | | | 1. | | Admin selects the “Student” |  | |  | |  | |  | 2. | | Admin is provided with name, CNIC, Student ID, contact info fields. | | 3. | | Admin enters the students name, CNIC, Student ID, contact info, |  | |  | |  | | 4. Admin selects the “Add”. |  | |  | |  | |  |  | | 5. Student is added into the system. | | 6. Admin goes back to the dashboard. | | |  | |  | | |
| Extensions: | 1. If the user tries to enters details (Name, CNIC) in wrong format:    1. User is given an error message.    2. System displays correct format for entries. | |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Pay Employees | |
| Scope: | CafeConnect Cafeteria management system. | |
| Level: | User Goal | |
| Primary Actor: | 1. Manage | |
| Stakeholders and interests: | **-** | **Manage:**   * Wants to pay the salaries of his employee. |
|  | **-** | **Cashiers:**   * Wants to get the same salary at the end of the month as committed. * Administrator: Wants to get the same salary at the end of the month as committed. |
| Preconditions: |  |  |
|  | 1. | The Manager has the necessary authorization and funds to make payments. |
|  | 2. | The Manager has access to the employee list containing their salaries and payment schedules. |
| Postconditions: |  | The Cashiers and Administrators receive their payments according to their agreed-upon salary and payment schedule. |
|  |  | |
|  |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Main Success Scenario: | |  |  |  |  | | --- | --- | --- | --- | | Actor Action | | System Response | | | 1. | The reviews the employee list to confirm the salaries and payment schedules for cashiers and administrator |  | | | 2. | The employee initiates the payment process by sending the necessary funds to the employees | . |  | |  |  | 3. System displays a successful message “Funds transferred” |  | |  |  | 4. | Employee account is deposited by the system. | | 5. | The Cashiers and Administrator receive the payment. |  |  | |  |  | 6. | System displays a message “Funds deposited”. | | 7. Confirm that the amount is accurate. |  | 8. | System notifies Supervisor and Team Members of project changes and approval | | 9. | If there are any discrepancies, the Cashiers or Administrators inform the Manager for resolution |  | | | 10. | The Manager updates the employee salary list to reflect the payment made. |  | | |  |  | 11. System shows a successful message “Payment Updated” | | | 12. The manager returns to the dashboard. |  |  | | | |
| Extensions: |  | 1. If the Manager encounters any issues with making the necessary payments, such as insufficient funds or technical issues. 2. the Manager contacts the Finance Department to resolve the issue before proceeding with the payment. |
|  |  |  |

|  |  |
| --- | --- |
| High Level Use case: Process Sale | |
| Use Case | Process Sale |
| Scope | Cafe Connect System |
| Level | User Goal |
| Primary Actor | Cashier |
| Stakeholder and interests | Cashier: Wants accurate, fast entry, and no payment errors, as cash drawer shortages are deducted from his/her salary.  Customer: Wants purchase and fast service with minimal effort. Wants easily visible display of entered items and prices. Wants proof of purchase to support returns.  Government Tax Agencies: Want to collect tax from every sale. May be multiple agencies, such as national, state, and county. |
| precondition | Cashier is identified and authenticated. |
| Success Guarantee | Sale is saved. Tax is correctly calculated. Accounting and Inventory are updated. Commissions recorded. Receipt is generated. |
| Main Success Scenario | * Customer arrives at POS checkout with goods and/or services to purchase. * Cashier starts a new sale. * Cashier enters item identifier. * System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules. Cashier repeats steps 3-4 until indicates done. * System presents total with taxes calculated. * Cashier tells Customer the total, and asks for payment. * Customer leaves with receipt and goods (if any). |
| Extension | At any time, System fails:   * Cashier restarts System, logs in, and requests recovery of prior state. * System reconstructs prior state   Invalid item ID (not found in system):   * System signals error and rejects entry. * Cashier responds to the error.   There is a human-readable item ID (e.g., a numeric UPC):   * Cashier manually enters the item ID. * System displays description and price. * Invalid item ID: System signals error. * Cashier tries alternate method. |
| Special Requirement | * Somehow, we want robust recovery when access to remote services such the inventory system is failing. |
| Technology and data variation list | Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.  Item identifier may be any UPC, EAN, JAN, or SKU coding scheme. |

|  |  |
| --- | --- |
| High Level Use case: Manage Cashiers | |
| Use Case | Manage Cashier |
| Scope | Cafe Connect System |
| Level | User Goal |
| Primary Actor | Administrator |
| Stakeholder and interests | Administrator: Administrator can add, remove and update the profile of Cashier and view the complete activity of the cashier.  Manager: Manager also keep track that how may cahier created by the administrator and also view details. |
| precondition | Administrator account created by Manager. |
| Success Guarantee | Add, Update and remove the data of the Cashier Properly. |
| Main Success Scenario | * Manager create an Administrator Account. * Administrator can be added, update and remove Cashier from system. * Manager can view All the Cashier. * Administrator also Perform change in Cashier account. |
| Extension | If Administrator account not exist: Manger creates a profile of Administrator and when administrator account created it manage the cashier. |

|  |  |
| --- | --- |
| High Level Use case: Return Process | |
| Use Case | Return Process |
| Scope | Cafe Connect System |
| Level | User Goal |
| Actor | Customer |
| Stakeholder and interests | Cashier: Cashier will be to return all the buy items.  Customer: Customer can be able to return any item. |
| precondition | Sale record present in system. |
| Success Guarantee | System inventory update properly and update all sale record again. |
| Main Success Scenario | * If Customer want to remove any item they go to cahier. * Cashier check that this item sold against customer id. * If item sold by customer. * Cashier performed return process. * All the sale record updated. * Inventory also updated. * Customer get back |
| Extension | * If System down then cahier check the receipt and return the item. * Customer will not need to wait when system run. * Cashier, perform it return later when system came in run. |
| Special Requirement | If sale record not present return process cannot be performed by cashier. |
| Technology and data variation list | Item identifier entered by bar code laser scanner (if bar code is present) or keyboard.  Item identifier may be any UPC, EAN, JAN, or SKU coding scheme. |

## Wireframes in Figma

<https://www.figma.com/file/XXrgzM67EikSAK1IJbzpXf/Untitled?node-id=0%3A1&t=6dBL83GC1u0tlql4-1>

## GitHub Repository

<https://github.com/31304/SDA-Project.git>