



Palestine Technical University – Kadoorie
College of Engineering and Technology
Department of Computer Systems Engineering

Course name:

Software Engineering

Project title:

**DESIGNING CANTEEN MANAGEMENT SYSTEM FOR
STUDENTS AT PALESTINE TECHNICAL UNIVERSITY –
KADOORIE IN TULKARM**

By:

Hanan abuzainab - 202111195

Marah Badran - 202110360

Raneen Atwe - 202110343

Sara Zerie - 202110075

Supervisor:

Dr. Osama Hamed

Tulkarm, Palestine

27 May 2024

Abstract

The Canteen Management System (CMS) is designed to streamline the operations of a canteen by providing an integrated platform for managing orders, inventory, customer interactions, and feedback. This system automates the process of placing orders, processing payments, tracking inventory, and managing staff. The CMS encompasses several key entities, including Canteen, Customer, Worker, Item, Order, Feedback, and Category, which are interconnected to ensure efficient workflow and enhanced customer experience. The CMS aims to enhance the overall user experience by providing quick and reliable canteen services.

Customers can browse the available items, place orders, and provide feedback on the services received. Orders are processed by the workers, who are assigned by the canteen, ensuring that the food preparation and delivery are managed efficiently. The system keeps track of inventory, categorizing items to facilitate easy updates and management. Feedback from customers is collected and analyzed to continually improve service quality.

The system's architecture, as depicted in the Entity-Relationship (ER) diagram, ensures a robust and scalable solution that can handle the dynamic needs of a canteen. Each entity in the system is defined with specific attributes and relationships to other entities, ensuring comprehensive coverage of all aspects of canteen management. The CMS aims to improve operational efficiency, enhance customer satisfaction, and provide valuable insights for continuous improvement in canteen services.

This application will help canteen workers to get an idea of the orders consumed by customers we need an application like this to manage this huge and unorganized crowd of customers.

Users will be able to place orders digitally, and customers will be happy after using this application because it is easy to use and saves time and effort.

TABLE OF CONTENT

Abstract.....	II
Table of Content.....	III
CHAPTER1:Introduction.....	1
1.1 Purpose.....	1
1.2 Scope.....	1
1.3 Definitions, Acronyms, and Abbreviations.....	1
1.4 Overview.....	2
1.5 Team Work.....	2
CHAPTER 2: OVERALL DESCRIPTION.....	3
2.1 Product Perspective.....	3
2.2 Product Features.....	3
2.3 User Class and Characteristics.....	4
2.4 Operating and Environment.....	6
2.5 Design and Implementation Constraints.....	6
2.6 Assumption Dependencies.....	7
CHAPTER 3:System Features(Specific Requirements).....	8
3.1 External Interface Requirements.....	8
3.2 Functional Requirements.....	9
3.3 Behavior Requirements.....	11
CHAPTER 4:External Diagrams Requirements.....	13
4.1 Use Case Diagram.....	13
4.2 Sequence Diagram.....	15
CHAPTER 5:Nonfunctional Requirements.....	18
5.1 Performance Requirements.....	18
5.2 Safety Requirements.....	19
5.3 Security Requirements.....	19
5.4 Software Quality Attributes.....	19
CHAPTER 6:User Interfaces.....	20
6.1 Customer Interfaces.....	20
6.2 Employee Interfaces.....	22
References.....	24

Chapter 1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to detail the requirements for the University Canteen Automation System (CMS). This document aims to provide a comprehensive guide for the development team to design, build, and implement a mobile application that automates the canteen operations at the university. The CMS will streamline the process of ordering food, managing payments, and gathering feedback, thereby enhancing the overall experience for students and staff.

1.2 Scope

The CMS is a mobile application developed using the Flutter framework, intended to operate on both Android and iOS devices. It is designed to facilitate various canteen operations, including digital food ordering, payment processing using university account credits, and feedback collection. The system will benefit students by allowing them to place orders conveniently and benefit canteen staff by simplifying order management and menu updates. The primary goal is to improve efficiency, reduce errors, and enhance user satisfaction within the university's canteen services.

1.3 Definitions, Acronyms, and Abbreviations

CMS: Canteen Management System

UI: User Interface

API: Application Programming Interface

DBMS: Database Management System

Flutter: An open-source UI software development kit created by Google

iOS: iPhone Operating System

SI: software interfaces

1.4 Overview

This document is organized into the following sections:

Introduction: Provides an overview of the purpose, scope, and structure of the document.

Overall Description: Describes the general factors that affect the product and its requirements.

System Features: Details the specific features and functionalities required in the UCAS.

External Interface Requirements: Defines the user, hardware, software, and communication interfaces.

Other Requirements: Includes performance, security, and other system attributes

References: providing essential context and resources.

1.5 Team Work

Names	Work
Hanan Abuzainab	Overall Description, , Introduction ,Interfaces design.
Marah Badran	External Interface Requirements, Abstract, Table of Contact.
Raneen Atwe	Nonfunctional Requirements, Abstract, Cover page..
Sara Zerie	System Features, Introduction, References.

Chapter 2. OVERALL DESCRIPTION

2.1 Product Perspective

The CMS is a standalone application that integrates with the university's existing account management system. It will provide a user-friendly interface for students to place orders, make payments, and provide feedback, while also helping canteen staff manage orders and menus efficiently.

2.2 Product Features

The major features of the CMS database system as shown in below entity relationship model (ER model)

ER- Diagram

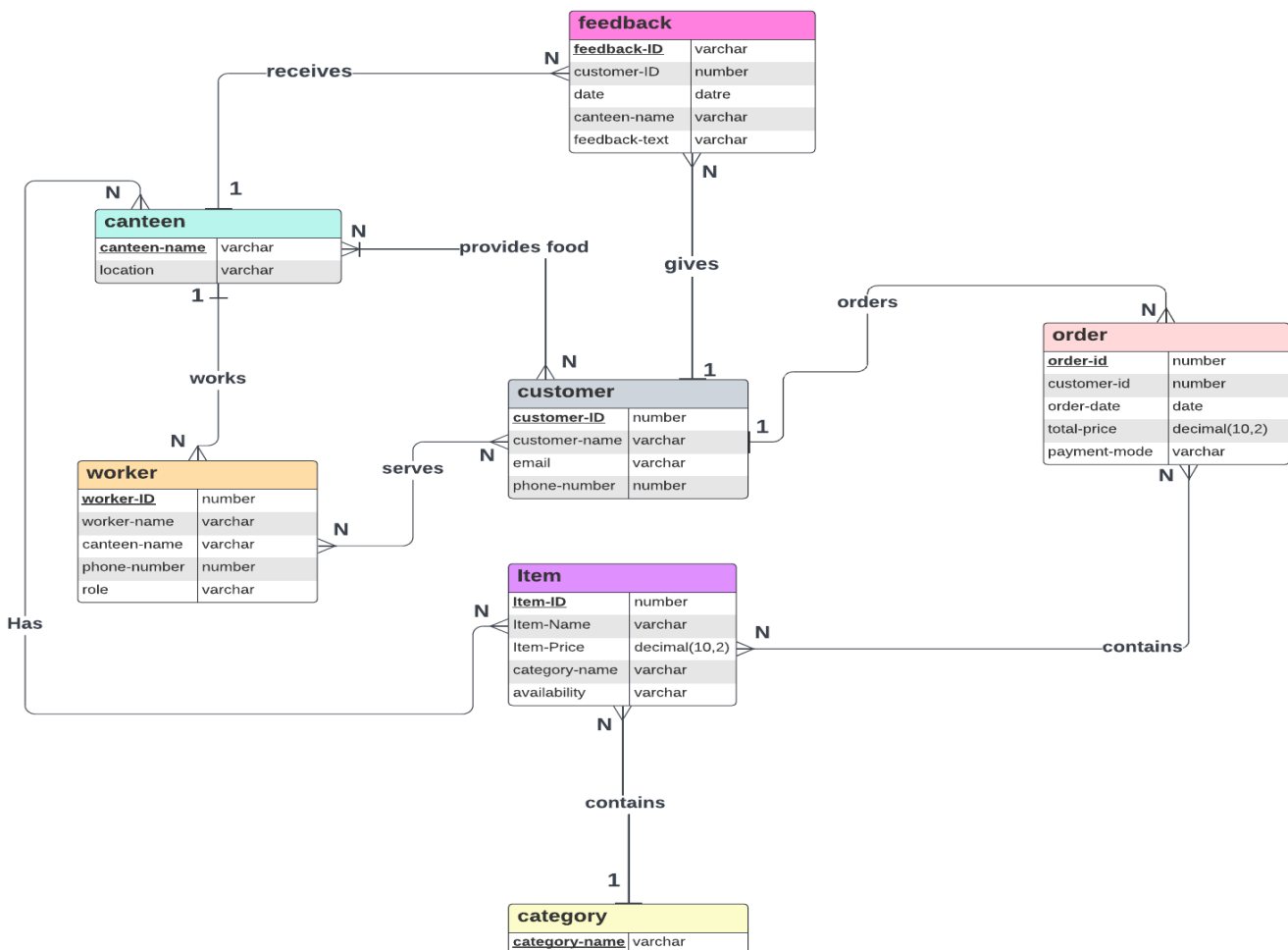


Fig 2.1 : ERD model

2.3 User Class and Characteristics

The Canteen Management System (CMS) will support two types of user privileges: Customers (Students) and Employees (Canteen Staff). Customers will have access to customer functions, while employees will have access to both customer and canteen management functions.

Customer (Student)

Customers are the primary users of the CMS. They are students who can place orders, make payments, and provide feedback. Customers can perform the following functions:

make a new order: Order individual items from the menu.

Order Confirmation: Review and confirm the order.

Cancel an Existing Order: Cancel an order before it is processed by the canteen.

View Order History: Check past orders and their details, including status, items, and payment information.

Provide Feedback: Submit feedback on orders and overall canteen service, including ratings and comments.

Employee (Canteen Staff)

Employees are responsible for managing orders, updating menu items, and viewing feedback. They have access to both customer and management functions. Employees can perform the following functions:

All Customer Functions.

Manage Orders:

View All Orders: See a list of all orders placed by customers.

Update Order Status: Change the status of orders (e.g., pending, in progress, completed).

View Customer Order Details: Get detailed information about customer orders, including items ordered, and payment status.

Manage Feedback:

View All Feedback: See feedback submitted by customers.

Respond to Feedback: Address customer concerns or comments if needed.

Administrative Functions

Manage Menu

Add/Delete Menu Items: Update the menu by adding new items or removing existing ones.

Update Item Details: Modify details of menu items, such as name, price, and description.

Update Item Availability: Change the availability status of items (e.g., available, out of stock).

Manage Canteen Information

Add/Delete Canteens: Add new canteen locations or remove existing ones.

Update Canteen Details: Modify details of canteen locations, such as name and address.

View Reports

Daily Sales Report: Generate a report of total sales for a specific day.

Order Statistics: View statistics on orders, such as the number of orders placed, popular items, and peak order times.

Customer Activity: Get a list of active customers and their order history

2.4 Operating Environment

The University Canteen Automation System (CMS) will operate as a mobile application developed using Flutter, making it accessible on both Android and iOS devices. The system will require the following components:

Client Side:

Mobile devices running on Android or iOS operating systems.

The Flutter app will be installed on these devices, providing a responsive and intuitive user interface.

Server Side:

A backend server running a web server to handle requests from the mobile app.

A Database Management System (DBMS) to store and manage data, such as user information, orders, menu items, and feedback.

Client-Side Requirements

Supported Operating Systems:

Android: Version 6.0 (Marshmallow) and above.

iOS: Version 11.0 and above.

Development Framework:

Flutter: The mobile app will be developed using the Flutter framework, ensuring a consistent and high-quality user experience across both Android and iOS platforms.

Server-Side Requirements

Web Server: A web server (e.g., Apache, Nginx) to serve API requests from the mobile app.

Database: A robust DBMS (e.g., MySQL, PostgreSQL, MongoDB) to manage and store application data.

APIs: RESTful APIs or GraphQL to facilitate communication between the mobile app and the server.

2.5 Design And Implementation Constraints

Compliance with University IT Security Policies: The system must adhere to the university's IT security policies, ensuring the protection of sensitive data and compliance with relevant regulations.

Integration with University Account Management System: The system must integrate seamlessly with the university's existing account management system for authentication and payment processing.

Technology Stack: The system will be developed using the following specific technology stack:

Frontend: Flutter framework for cross-platform mobile app development.

Backend: Node.js or Django for server-side processing.

Database: PostgreSQL or MongoDB for data storage.

APIs: RESTful APIs or GraphQL for communication between the mobile app and the server.

Hosting: Cloud-based hosting services such as AWS, Google Cloud, or Azure for scalability and reliability.

2.6 Assumption Dependencies

Active University Accounts: It is assumed that students have active university accounts with sufficient credits to make purchases through the canteen system.

University IT Infrastructure: The university's IT infrastructure must support the necessary integrations, including authentication services, payment gateways, and data synchronization.

Staff Training: It is assumed that canteen staff will receive adequate training to use the system effectively, including managing orders, updating menu items, and handling customer feedback.

Internet Access: Reliable internet access is available for both students and canteen staff to ensure real-time data synchronization and seamless operation of the mobile app.

Chapter3. System Features (Specific Requirements)

3.1 External Interface Requirements

3.1.1 User Interfaces

User interfaces in a canteen management system play a pivotal role in streamlining operations and enhancing user experience. These interfaces serve as the bridge between users and the system, offering intuitive navigation and efficient interaction. A well-designed interface enables canteen staff to manage orders, track inventory, and handle transactions seamlessly, thus optimizing workflow and minimizing errors. Meanwhile, for customers, an intuitive interface facilitates easy menu browsing, order placement, and payment processing, enhancing their overall dining experience. Whether through desktop applications, mobile apps, or self-service kiosks, user interfaces in canteen management systems prioritize functionality, accessibility, and user-friendly design to ensure smooth operations and customer satisfaction.

UI-1: The CMS screen displays shall conform to the Process Impact Internet Application User Interface Standard, Version 1.0 [4].

3.1.2 Hardware Interfaces

No hardware interfaces have been identified.

3.1.3 Software Interfaces

Software interfaces in a canteen management system are the linchpin of efficient communication and seamless interaction between different components of the system. These interfaces act as gateways for various software modules to exchange data and commands, enabling the system to function as a cohesive unit. For instance, the interface between the inventory management module and the ordering system ensures that stock levels are accurately reflected in real-time, preventing shortages or overstocking. Similarly, the interface between the payment processing module and the customer database facilitates secure transactions and personalized services. These software interfaces are meticulously designed to adhere to industry

standards, promote interoperability, and accommodate future expansions or integrations. Through robust and well-defined interfaces, a canteen management system can effectively automate tasks, improve accuracy, and enhance overall operational efficiency.

SI-1: Canteen Management System

SI-1.1: The CMS shall transmit the quantities of food items ordered to the CMS through

a programmatic interface.

SI-1.2: The CMS shall poll the CMS to determine whether a requested food item is available.

SI-1.3: When the CMS notifies the CMS that a specific food item is no longer available,

the CMS shall remove that food item from the menu for the current date.

3.1.4 Communications Interfaces

Communications interfaces are the vital conduits that enable seamless exchange of data and commands within a canteen management system, facilitating efficient coordination and operation. These interfaces encompass various communication protocols and technologies that link different components of the system, including hardware devices, software modules, and external services. For example, a communication interface between the kitchen display system and the order management software ensures that incoming orders are promptly relayed to kitchen staff for preparation. Similarly, interfaces between point-of-sale terminals and payment gateways facilitate secure transactions and real-time payment processing. Moreover, interfaces with external systems such as inventory suppliers or delivery services streamline procurement and logistics operations. By leveraging robust communication interfaces, a canteen management system can optimize workflow, enhance accuracy, and deliver a seamless dining experience for both staff and customers.

3.2 Functional Requirements

Functional requirements for a canteen management system typically include features and capabilities that directly contribute to the system's primary functions and objectives. Here are some common functional requirements:

1. **Menu Management :** The system should allow canteen administrators to create, update, and manage the menu, including adding new items, setting prices, and specifying ingredients.
2. **Order Management:** Users should be able to place orders for food items through various channels such as self-service kiosks, mobile apps, or in-person at the counter. The system should process these orders efficiently, ensuring accuracy and timeliness.
3. **Inventory Management:** The system should track inventory levels of food items in real-time, automatically updating quantities as orders are processed and ingredients are used. It should generate alerts when inventory levels are low to facilitate timely restocking.
4. **Payment Processing:** The system should support various payment methods such as cash, credit/debit cards, mobile wallets, and meal cards. It should securely process payments, generate receipts, and maintain transaction records.
5. **User Authentication and Authorization:** The system should authenticate users (both customers and staff) and provide appropriate access levels based on roles and permissions. This ensures that only authorized personnel can perform certain functions such as managing menus or accessing financial data.
6. **Reporting and Analytics:** The system should generate reports on various aspects of canteen operations, such as sales trends, inventory turnover, and customer preferences. These reports help in decision-making and performance evaluation.
7. **Integration with External Systems:** The system should be able to integrate with other systems such as accounting software for financial management, inventory management systems for procurement, and student/staff databases for user authentication.
8. **Kitchen Display System (KDS):** For efficient kitchen operations, the system should integrate with a KDS to display orders in real-time, prioritize them based on urgency, and mark them as completed once prepared.
9. **Customer Feedback and Loyalty Programs:** The system should provide mechanisms for collecting customer feedback on food quality and service. Additionally, it may support loyalty programs to incentivize repeat business.
10. **Accessibility and Usability:** The system should be user-friendly and accessible to customers and staff of varying technical proficiency. This includes intuitive interfaces, clear instructions, and support for multiple languages if necessary.

These functional requirements collectively ensure that the canteen management system operates smoothly, efficiently manages resources, and meets the needs of both customers and staff.

3.3 Behaviour Requirements

A Use Case View for a canteen management system would outline the various interactions between the system and its users, as well as between different components within the system itself. Here's a simplified example:

1. Actor: Customer

- **Use Case: Place Order**
 - **Description:** The customer selects items from the menu, adds them to the cart, and proceeds to checkout.
 - **Subflows:**
 - View Menu
 - Add Item to Cart
 - Remove Item from Cart
 - Checkout

2. Actor: Cashier

- **Use Case: Process Order**
 - **Description:** The cashier receives orders from customers, enters them into the system, calculates the total amount, and accepts payment.
 - **Subflows:**
 - View Orders
 - Enter Order
 - Calculate Total
 - Accept Payment

3. Actor: Chef/Kitchen Staff

- **Use Case: Prepare Food**
 - **Description:** The chef receives orders from the system, prepares the food items, and updates the status when ready for pickup.
 - **Subflows:**
 - View Orders
 - Prepare Food
 - Update Status

4. Actor: Manager

- **Use Case: Manage Inventory**
 - **Description:** The manager monitors and updates inventory levels, adds new items, removes discontinued items, and sets reorder thresholds.
 - **Subflows:**

- View Inventory
- Add Item
- Remove Item
- Update Item Quantity

5. Actor: System

- **Use Case:** Generate Reports
 - **Description:** The system generates various reports such as sales reports, inventory reports, and customer feedback reports for analysis.
 - **Subflows:**
 - Generate Sales Report
 - Generate Inventory Report
 - Generate Feedback Report.

Chapter 4. External Diagrams

Requirements

4.1 Use Case Diagram

A use case is a description of how end users will use a software code .It describes a task or a series of tasks that users will accomplish using the software and includes the responses of the software to user actions.

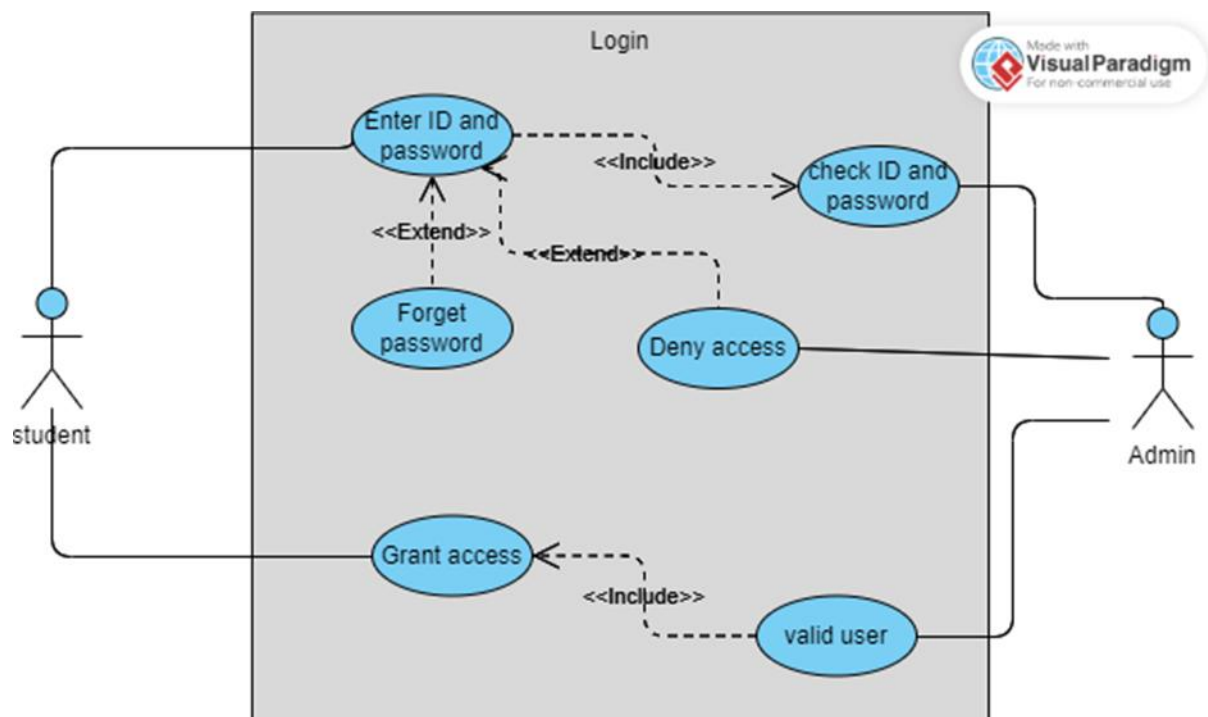


Fig 4.1.1 Use Case Login Page

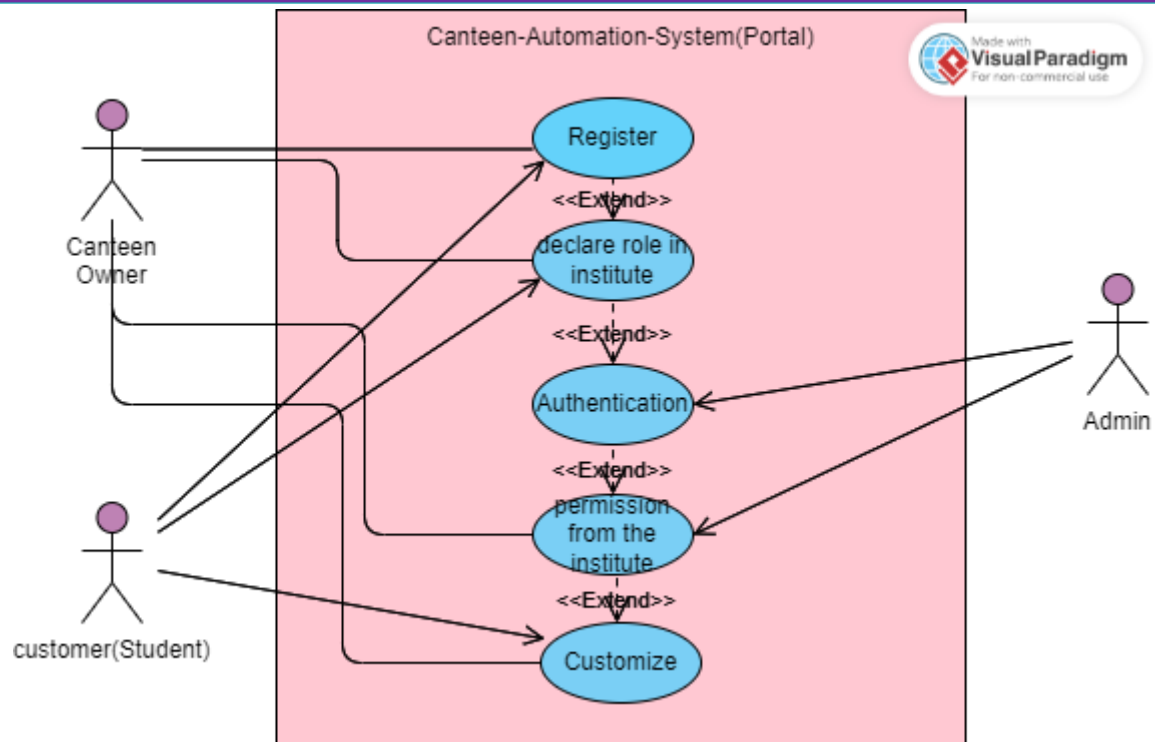
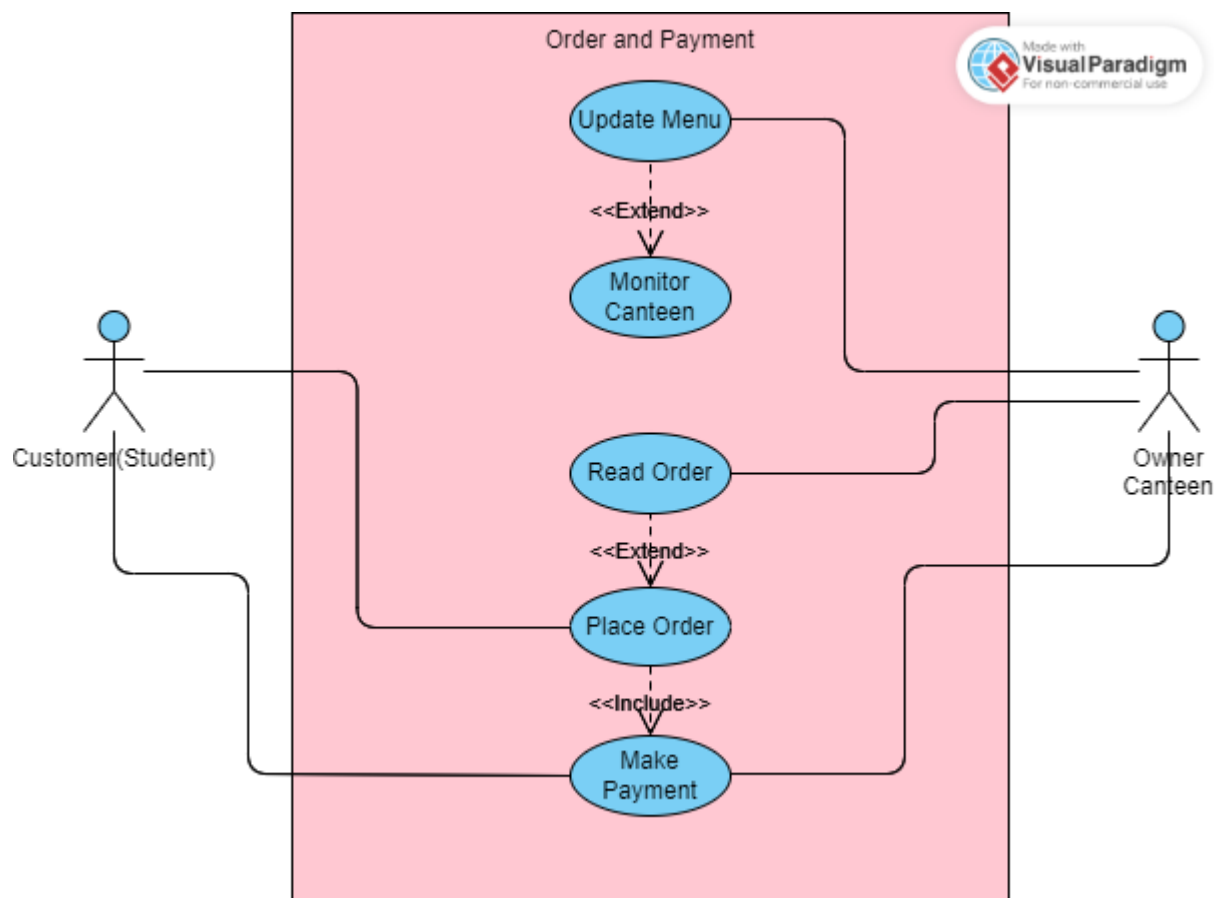


Fig4.1.2 Use Case Diagram for Registration



1.4.3 Use Case Diagram for Order and Payment

4.2 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order .It is a construct of Message Sequence Chart .Sequence diagrams are sometimes called event diagrams ,event scenarios ,and timing diagrams .

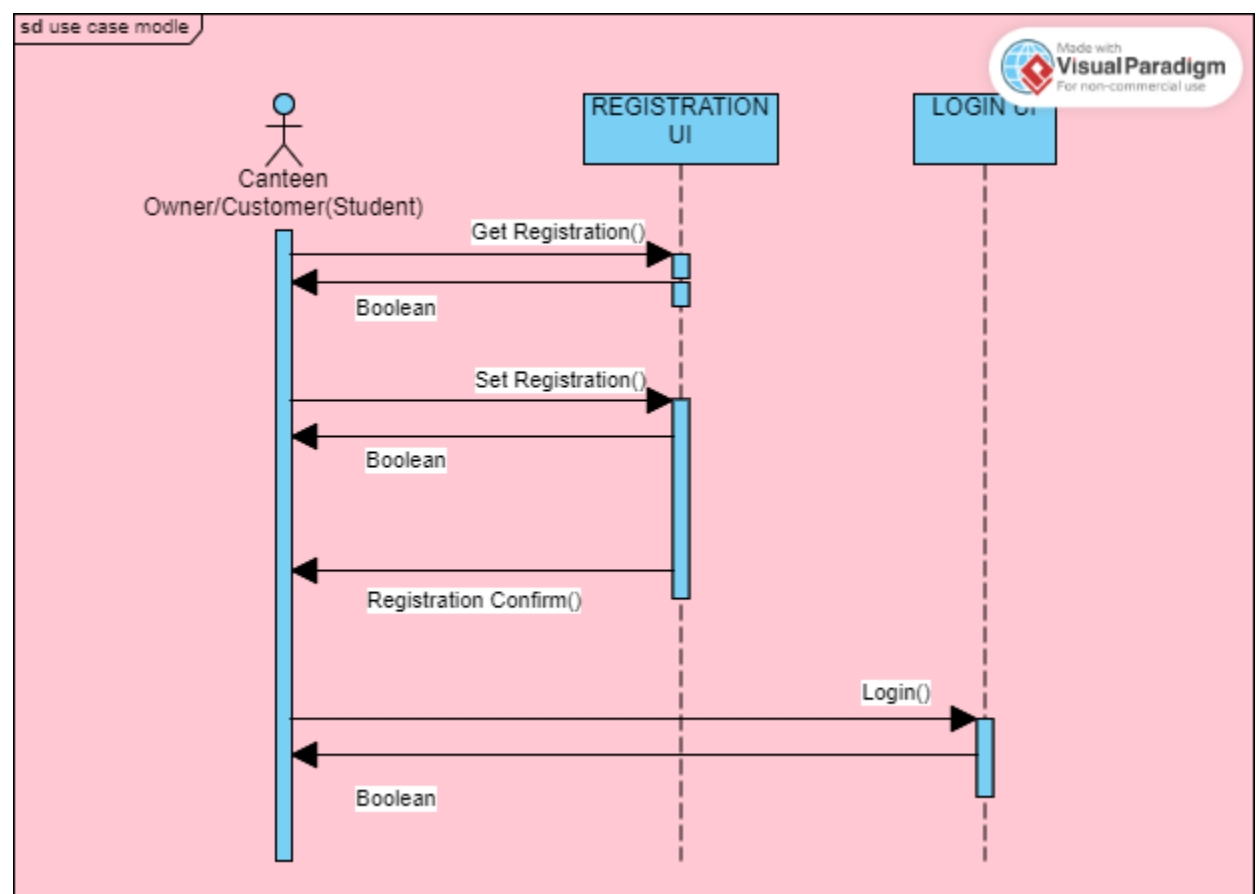


Fig 4.2.1 Sequence Diagram for Registration

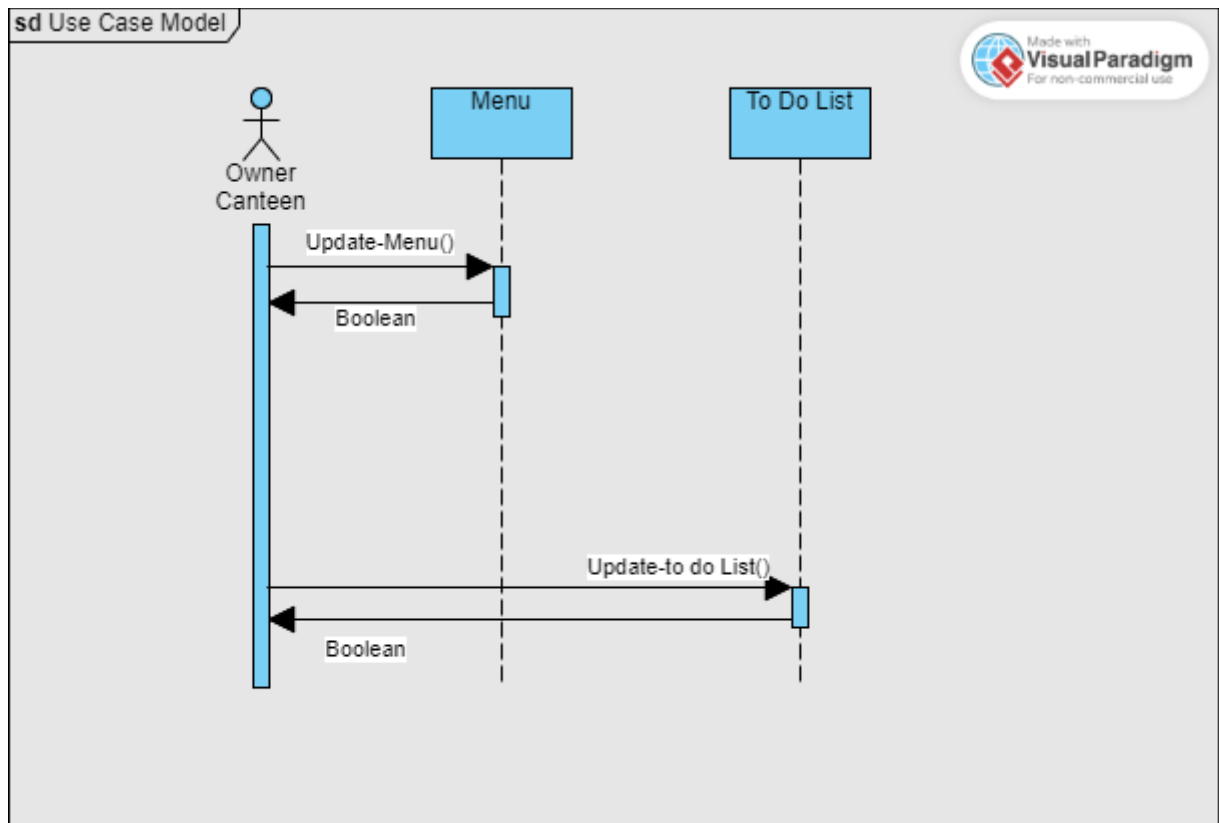


Fig 4.2.2 Sequence Diagram for Updating Menu

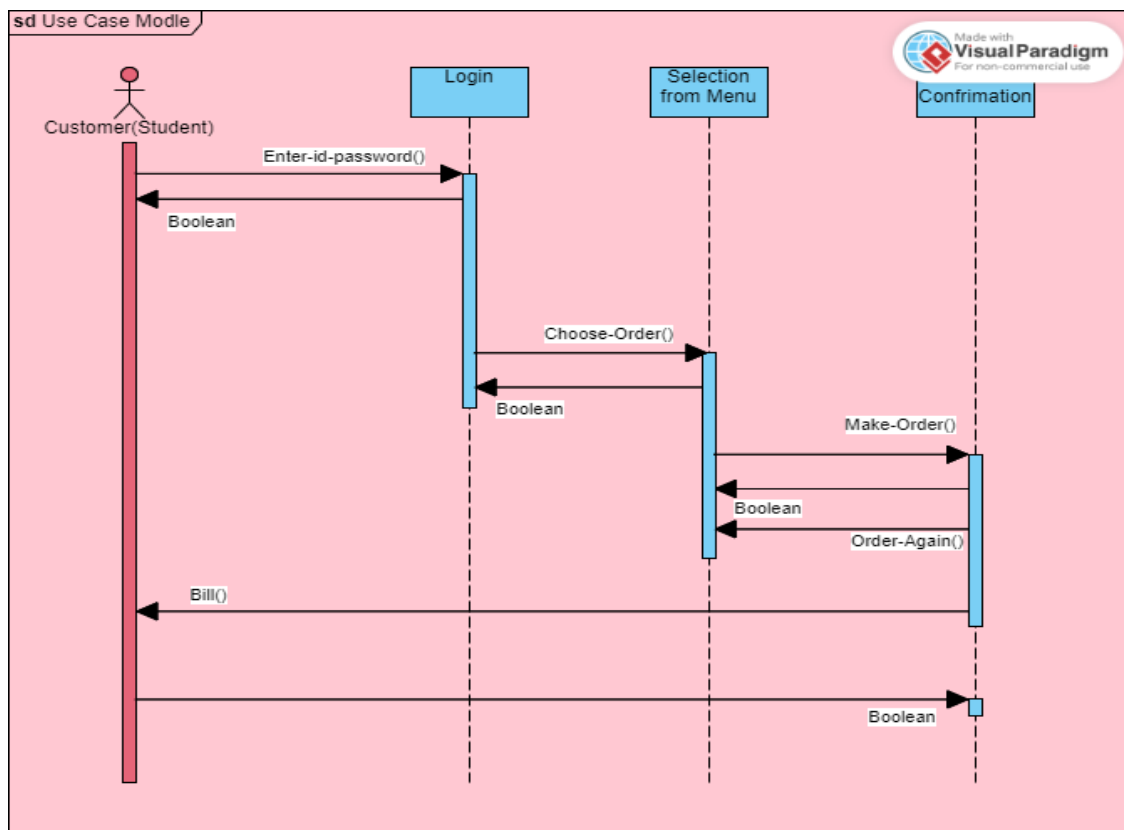


Fig 4.2.3 Sequence Diagram of Placing Order

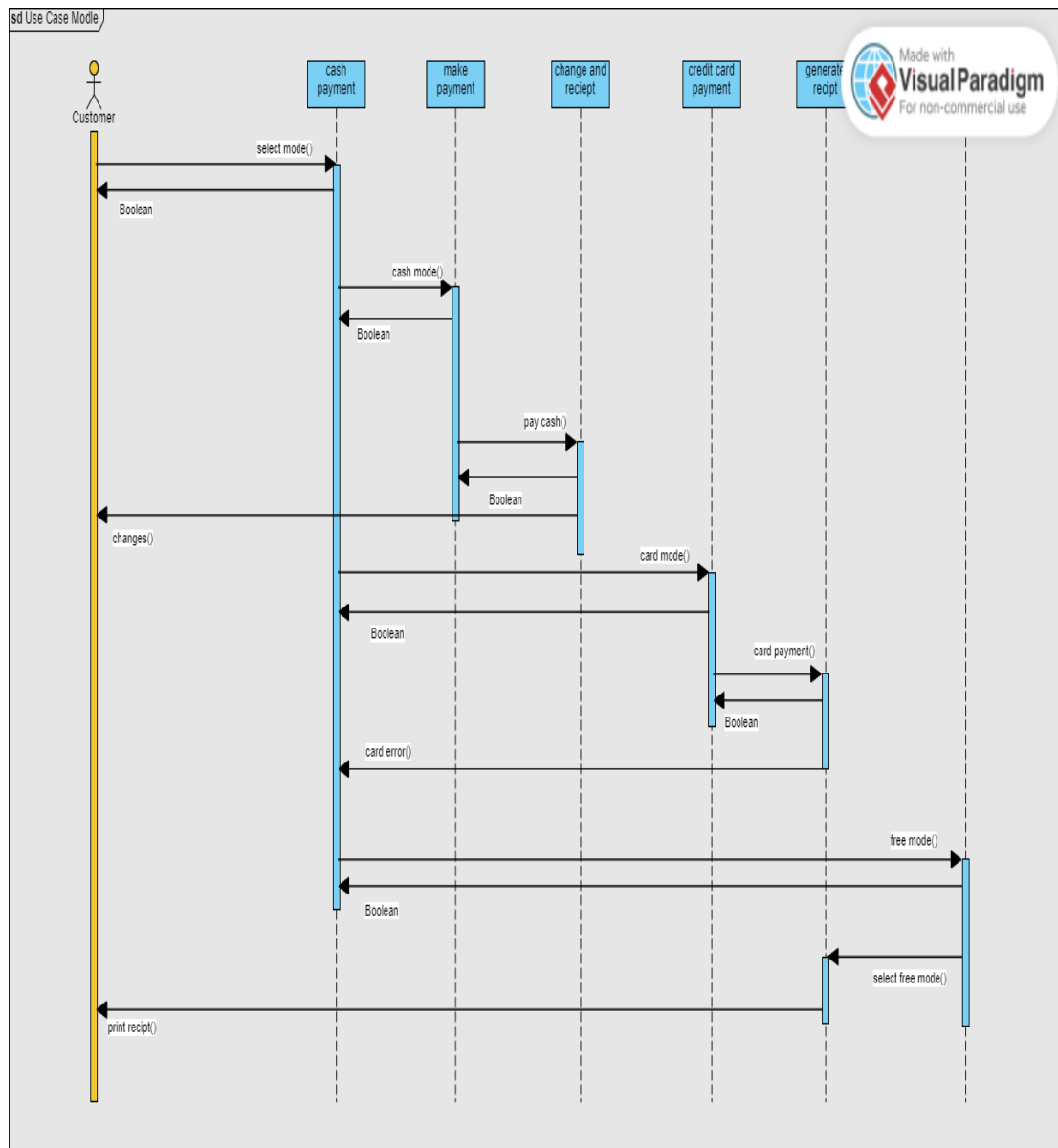


Fig 4.2.3 Sequence Diagram for Payment

Chapter 5. Nonfunctional Requirements

5.1 Performance Requirements

The system must be interactive, and the delays involved must be less. So, in every response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds. In case of opening databases, sorting questions and evaluation there are no delays and the operation is performed in less than 2 seconds for opening, sorting, computing, posting 95% of the files. Also, when connecting to the server the delay is based editing on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds for sake of good communication.

- **Safety:** Information transmission should be securely transmitted to server without any changes in information.
- **Reliability:** As the systems provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.
- **Availability:** If the internet service gets disrupted while sending information to the server, the information can be sent again for verification.
- **Security:** The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security. Hence, security is provided from unwanted use of recognition software.
- **Usability:** As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

- **Scalability:** The system is able to accommodate growth in the number of users and amount of data. It is also capable of handling increased transaction volumes.
- **Maintainability:** The system is easy to update and maintain. It has a modular design that allows components to be easily replaced. It also contains clear and detailed documentation.

5.2 Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Refer to any external policies or regulations that state safety issues that affect the product's design or use.

5.3 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product.

5.4 Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible.

At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

- **Availability :**The Cafeteria Ordering System shall be available to users on the corporate Intranet and to dial-in users 99.9% of the time between 5:00am and midnight local time and 95% of the time between midnight and 5:00am local time.

- **Robustness :**If the connection between the user and the system is broken prior to an order being either confirmed or canceled, the Cafeteria Ordering System shall enable the user to recover an incomplete order.

Chapter 6. User Interfaces

6.1 Customer Interfaces

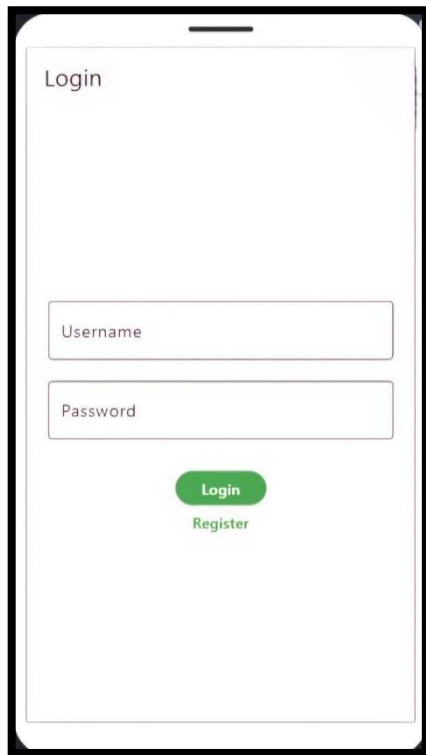


Fig6.1: Login Screen

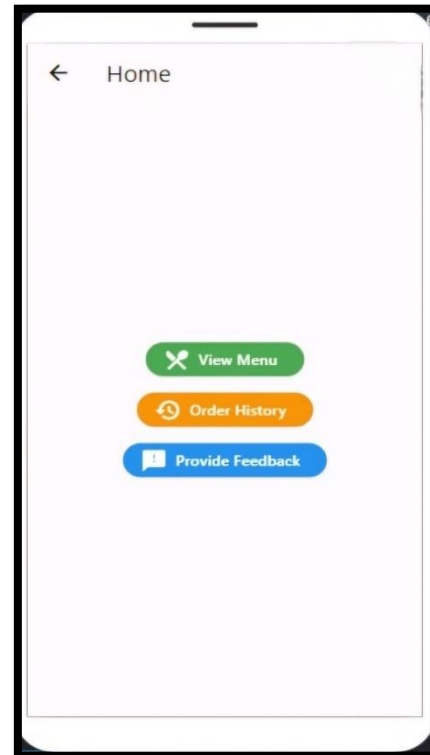


Fig6.2: Home Screen

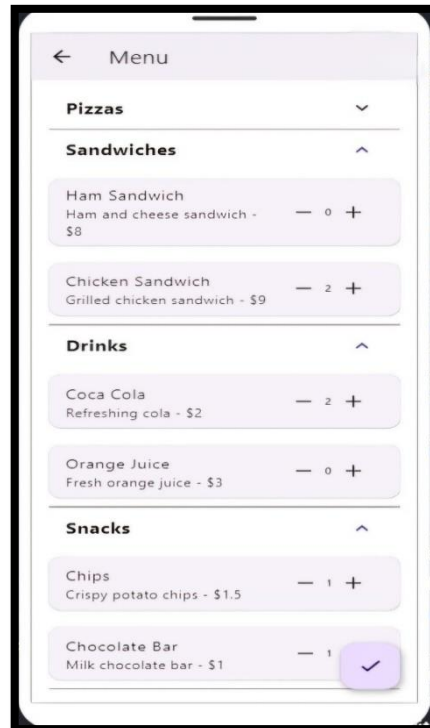


Fig6.3: Menu Screen

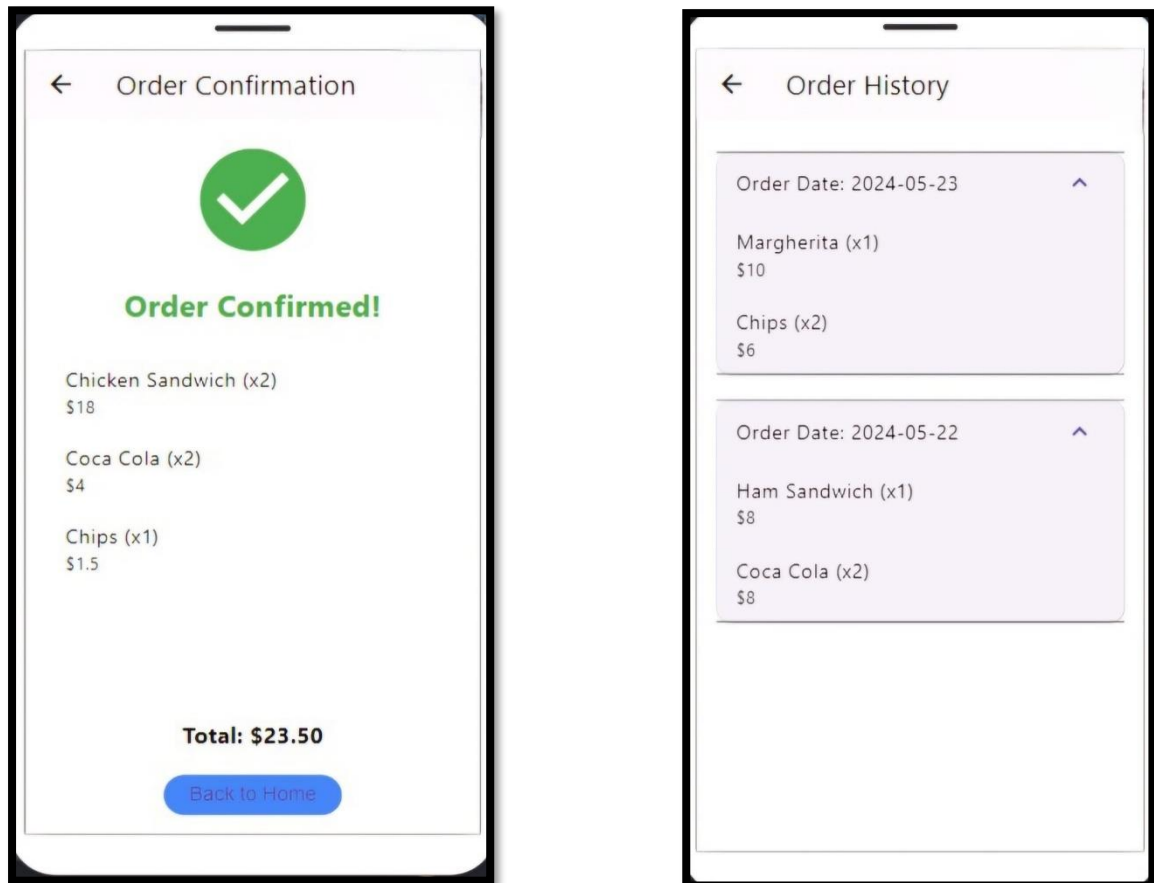


Fig6.4:

Order Confirmation
Fig6.5: Order

Screen
History Screen

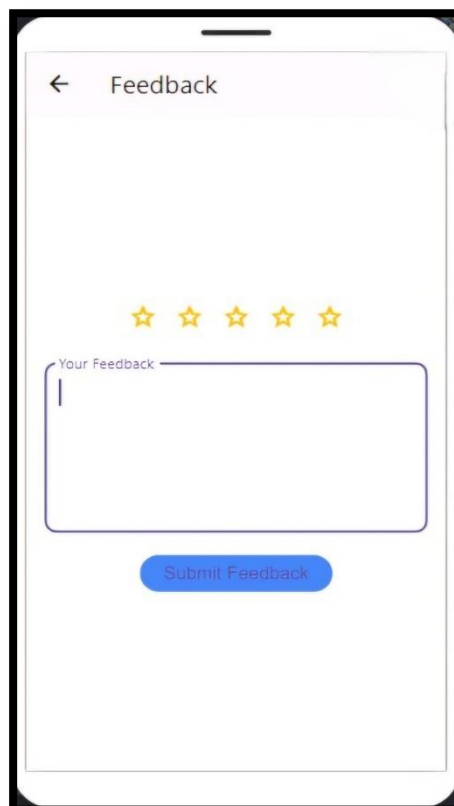


Fig6.6: Feedback Screen

6.2 Employee interfaces

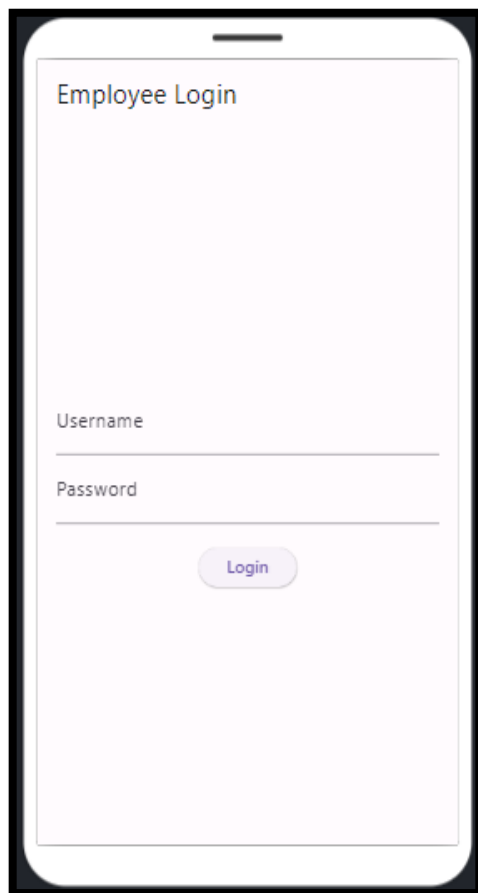


Fig6.7:Employee Login Screen

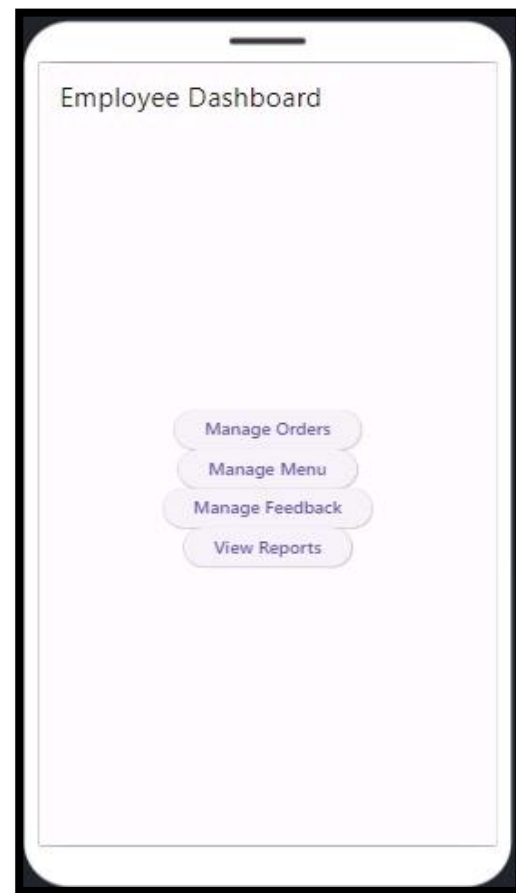
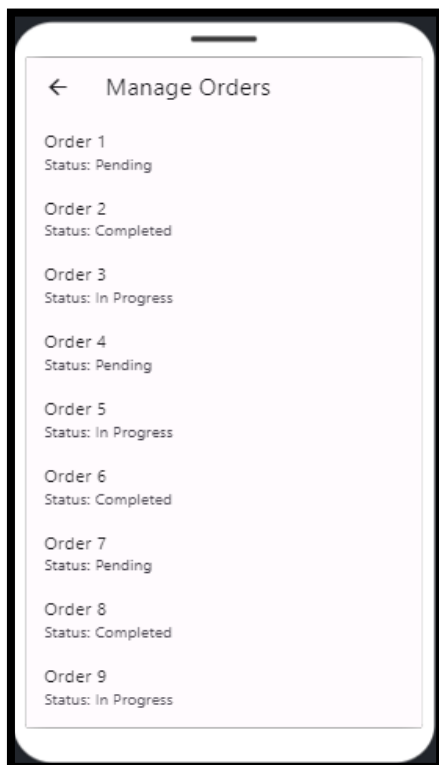


Fig6.8: Employee Dashboard Screen



Manage Order Screen



Fig6.10: Manage menu Screen



Fig6.11: Manage Feedback Screen

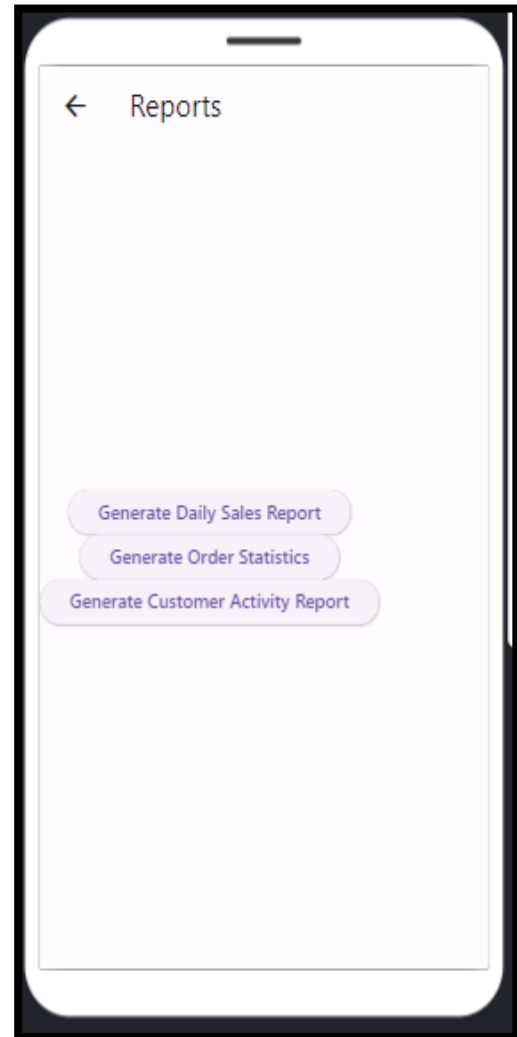


Fig6.12: Reports Screen

References:

1. IEEE SRS Template: The IEEE (Institute of Electrical and Electronics Engineers) provides a standard template for SRS documents. we find this template easily by searching for "IEEE SRS Template". It provides a comprehensive structure for documenting software requirements. Research Papers: Explore academic databases such as IEEE Xplore (<https://ieeexplore.ieee.org/>) and ACM Digital Library (<https://dl.acm.org/>) for research papers on software requirements engineering. Sometimes these papers include examples of SRS documents or discuss methodologies for creating them.
2. Online Resources and Tutorials: There are many online resources and tutorials available that provide guidance on creating SRS documents. Websites like Tutorialspoint, Guru99, and TechBeamers offer tutorials on writing SRS documents and provide examples for various systems, including canteen management and food ordering systems.
3. Academic Websites: Explore the websites of universities and colleges, particularly their computer science or software engineering departments. They often host project reports and documentation as part of student projects. Examples include MIT OpenCourseWare (<https://ocw.mit.edu/>) and Stanford University's course materials (<https://online.stanford.edu/courses>).
4. This is the link of set of projects of acanteen systems <https://github.com/topics/canteen-mangement-system>
5. Books on Software Engineering: Books on software engineering include chapters or sections on requirements engineering and provide examples of SRS documents. textbooks like "Software Engineering: A Practitioner's Approach" by Roger S. Pressman or "Software Engineering" by Ian Sommerville.
6. LinkedIn: LinkedIn is a professional networking platform where professionals share knowledge and resources.
7. YouTube: Search for tutorials on YouTube