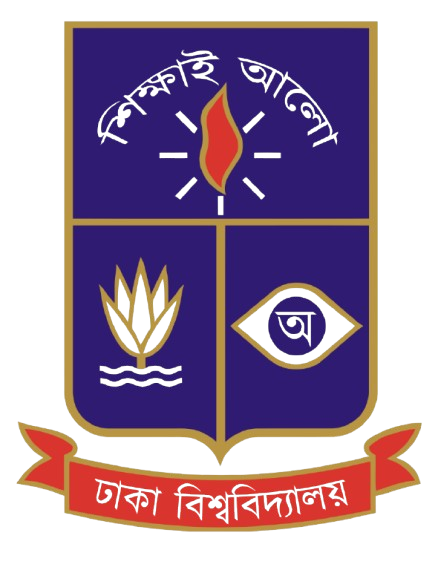
# CSE 2112: Object-Oriented Programming Lab

Department of Computer Science & Engineering 

# 

# **Project Report** on

**Cafeteria Management, DU**

Submitted To

Dr. Chowdhury Farhan Ahmed, Professor

Md. Redwan Ahmed Rizvee, Lecturer

Submitted By

Group No: 10

**Suraya Jannat Mim (Roll: 17)**

**Anisha Tabassum (Roll: 19)**

**Jubayer Ahmed Sojib (Roll: 23)**

**Tamal Kanti Sarker (Roll: 39)**

Table of Contents

1. Introduction 3

1.1 Project Overview 3

1.2 Objectives 3

1.3 Features 4

1.4 Tools 4

1. Design & Implementation 5

2.1 User Interface 5

2.2 Workflow 15

2.3 Implementation of OOP Principles 16

1. Conclusion 17

3.1 Challenges 17

3.2 Future Plans 17

3.3 Repositories 18

1. Introduction

1.1 Project Overview

The **Cafeteria Management,DU** is a Java-based application designed to streamline cafeteria operations at the Science Complex, University of Dhaka.

Despite the presence of important departments, there is currently no cafe or canteen in the Science Complex. To address this, we have designed a system that provides secure login access for different users (Teachers, Students, Admin, and Employees) and offers functionalities for ordering food, managing cafeteria inventory, tracking sales, and ensuring secure user authentication.

1.2 Objectives

The objectives of our project are:

* To develop a secure and efficient cafeteria management system for the Science Complex, University of Dhaka, providing a much-needed café service.
* To implement role-based authentication for Teachers, Students, Admin, and Employees to ensure secure access.
* To enable online food ordering and purchase tracking for a seamless user experience.
* To automate inventory and sales management for better cafeteria operation and financial tracking.
* To enhance user security and convenience with password recovery and a structured employee approval system.

1.3 Features

This system consists of three main roles:

* Admin:

◆ Approves new employees

◆ Updates product availability and pricing

◆ Monitors daily and total income

◆ Processes and tracks customer orders

◆ Also approves employee logins to prevent unauthorized

access.

* Customer (Students & Teachers):

◆ View the menu;

◆ Order food;

◆ Leave reviews;

◆ Check their purchase history;

◆ Update personal details on their profile;

* Employee:

◆ Can perform all tasks like an admin, except adding

employees to the system.

1.4 Tools

* IDE: IntelliJ
* Language: JAVA
* Framework: JavaFX
* Database: MySQL
* Testing: Manual testing for the overall system functionality.

2. Design & Implementation

2.1 User Interface

This system consists of three main roles:

* Admin
* Customer
* Employee

Front Page:

This is the front page or landing page where the user can decide whether they want to sign up or log in.

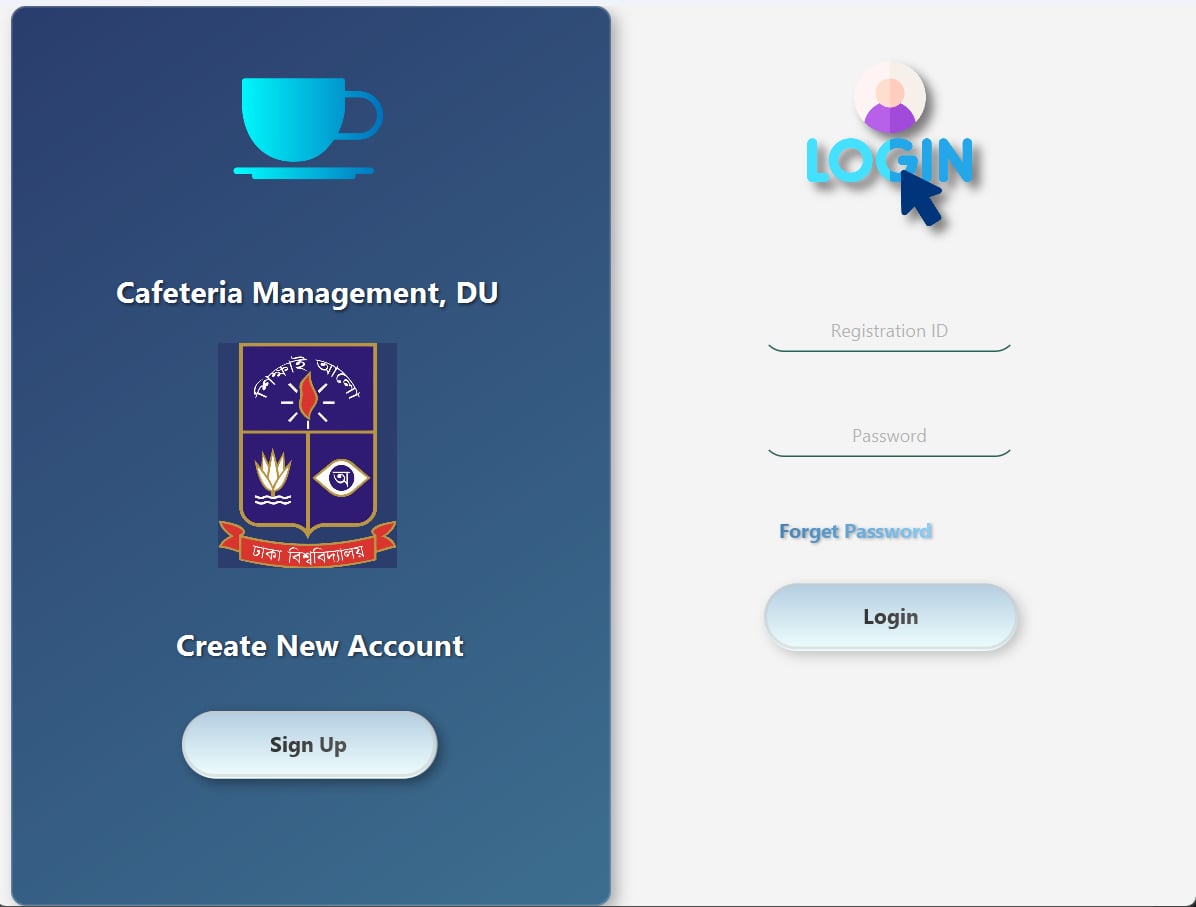


Fig: Front Page/Login Page

SignUp & Login Page:

Users can sign up for an account by providing the necessary information. This information will be stored in the database for their profile and confirmation. Students and teachers need their unique registration ID and password to log in easily. Employees need the admin's confirmation, along with their registration ID and password, to log in. The admin has their own unique ID and password.

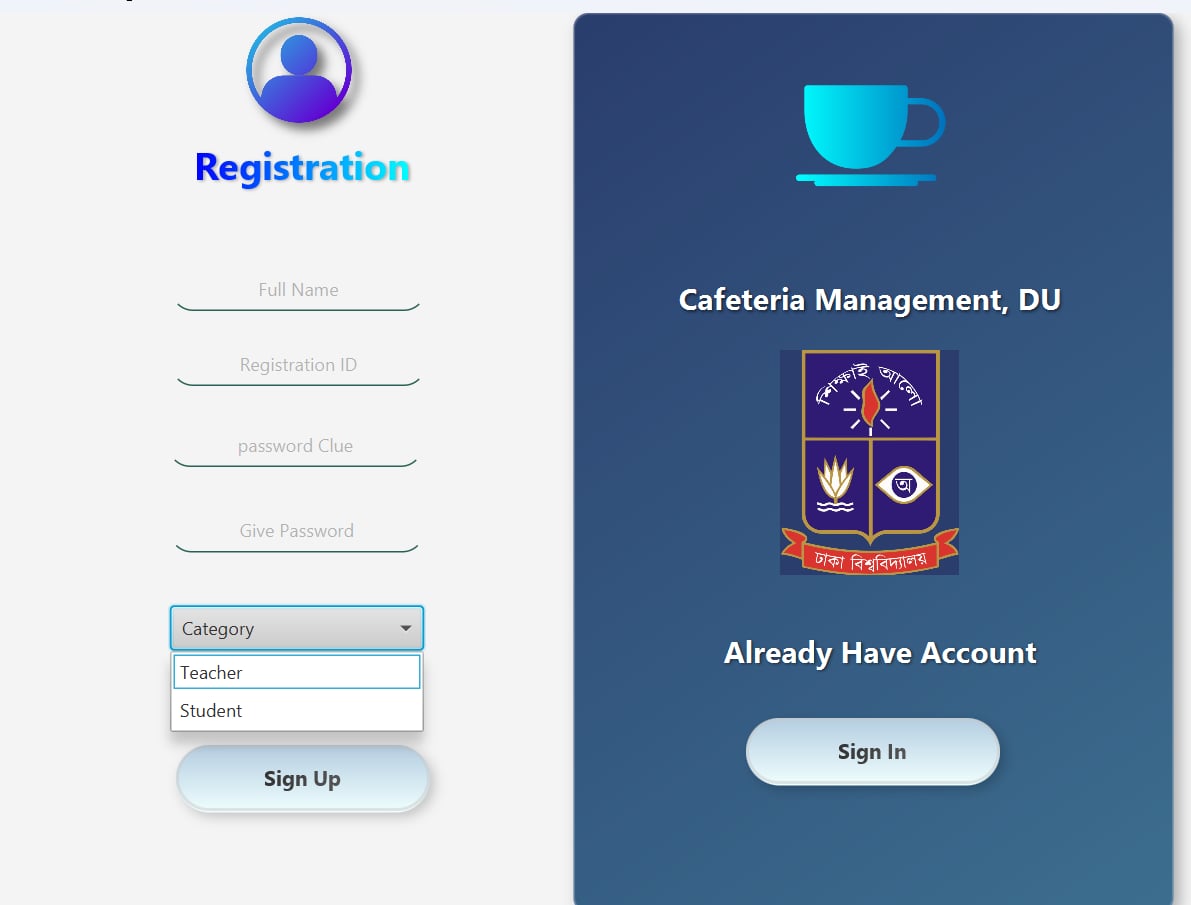


Fig: SignUp Page

Admin:

The admin page has 6 features, including their *Profile*. These are the *Dashboard*, *Inventory*, *Menu*, *Customer*, and *Employee*.

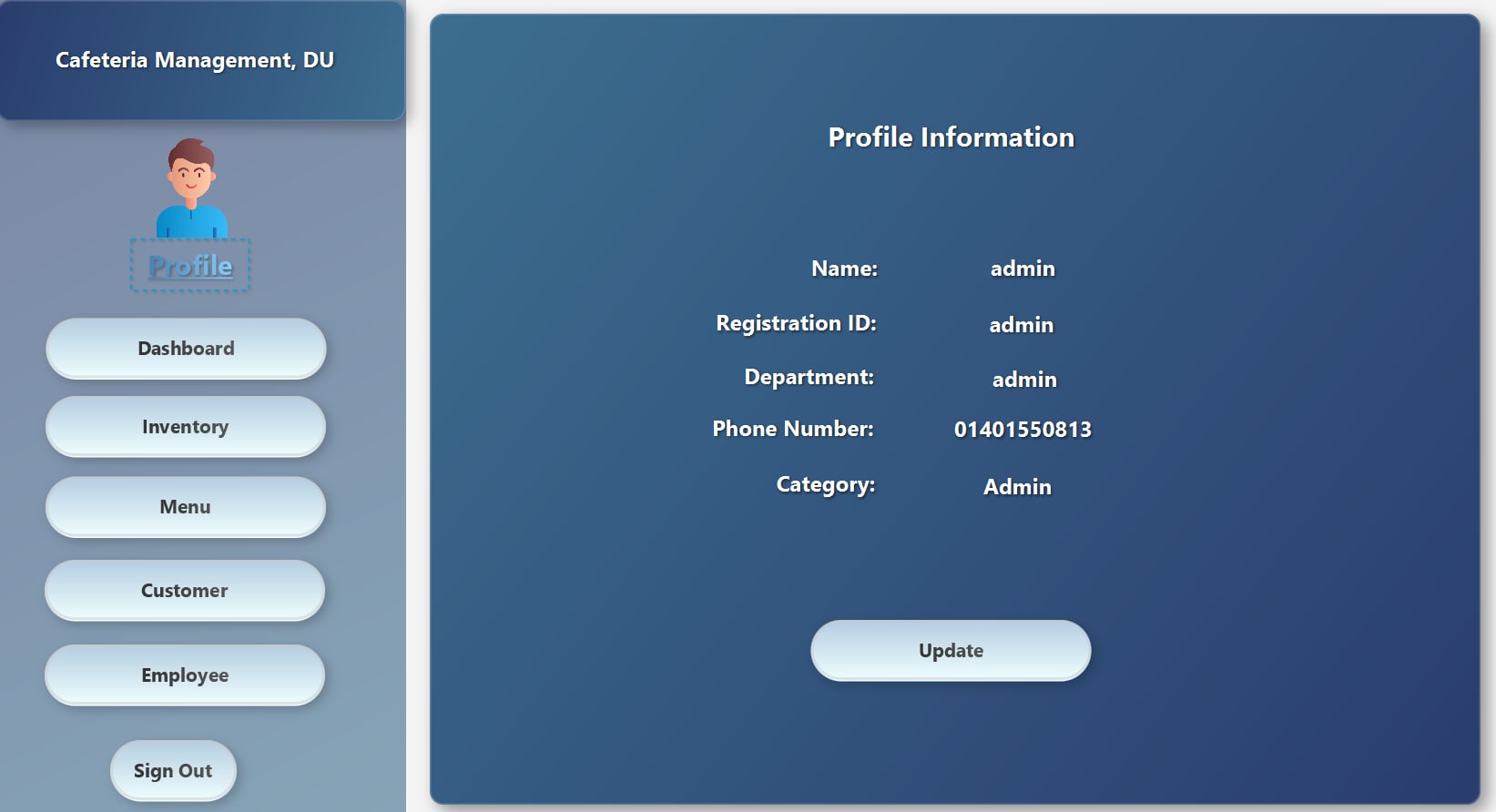


Fig: Profile

He can also update his profile in the *Profile* section. This is actually **common** for all three types of users (**admin, customer, and employee**).

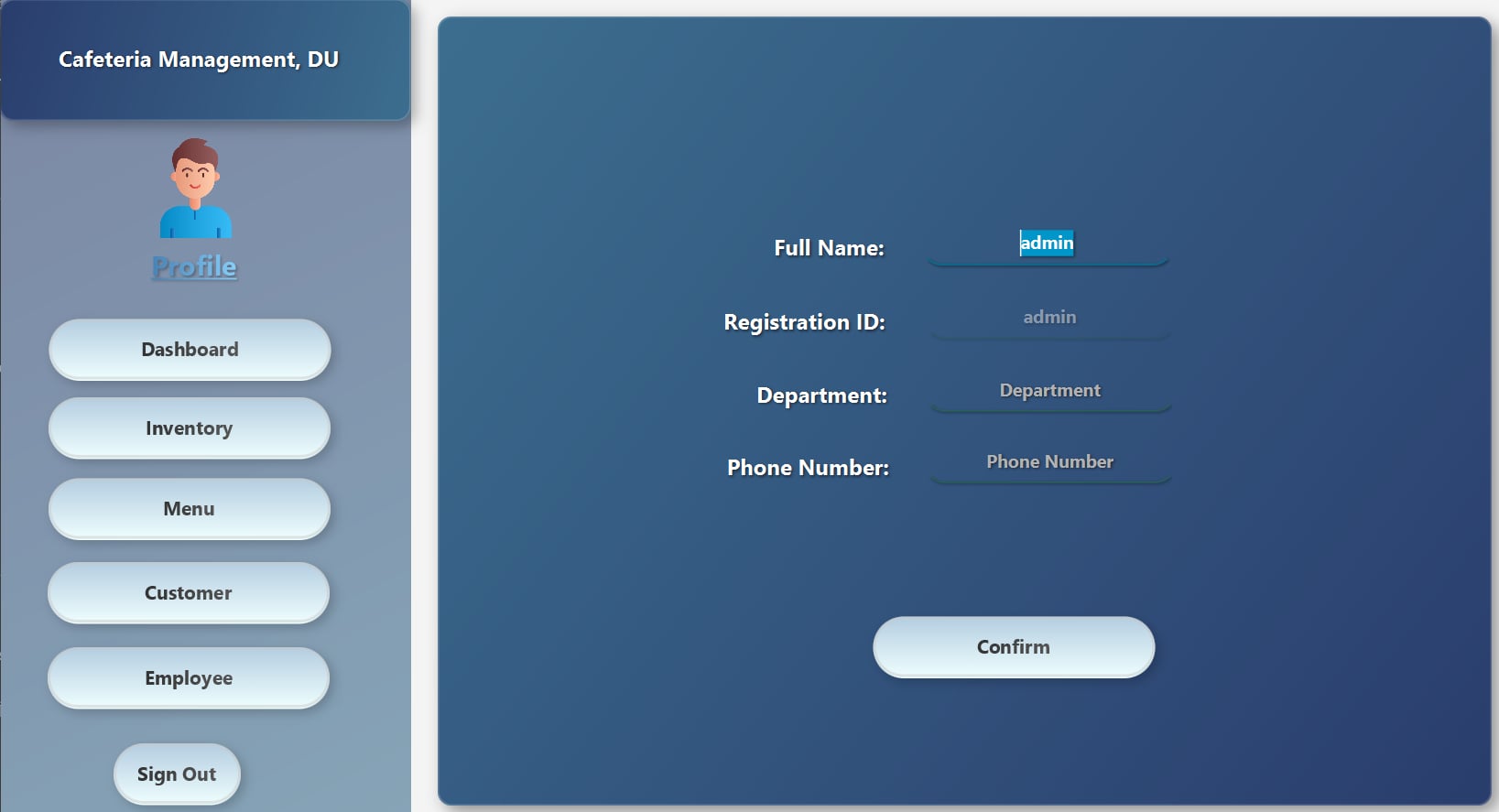


Fig: Update Profile

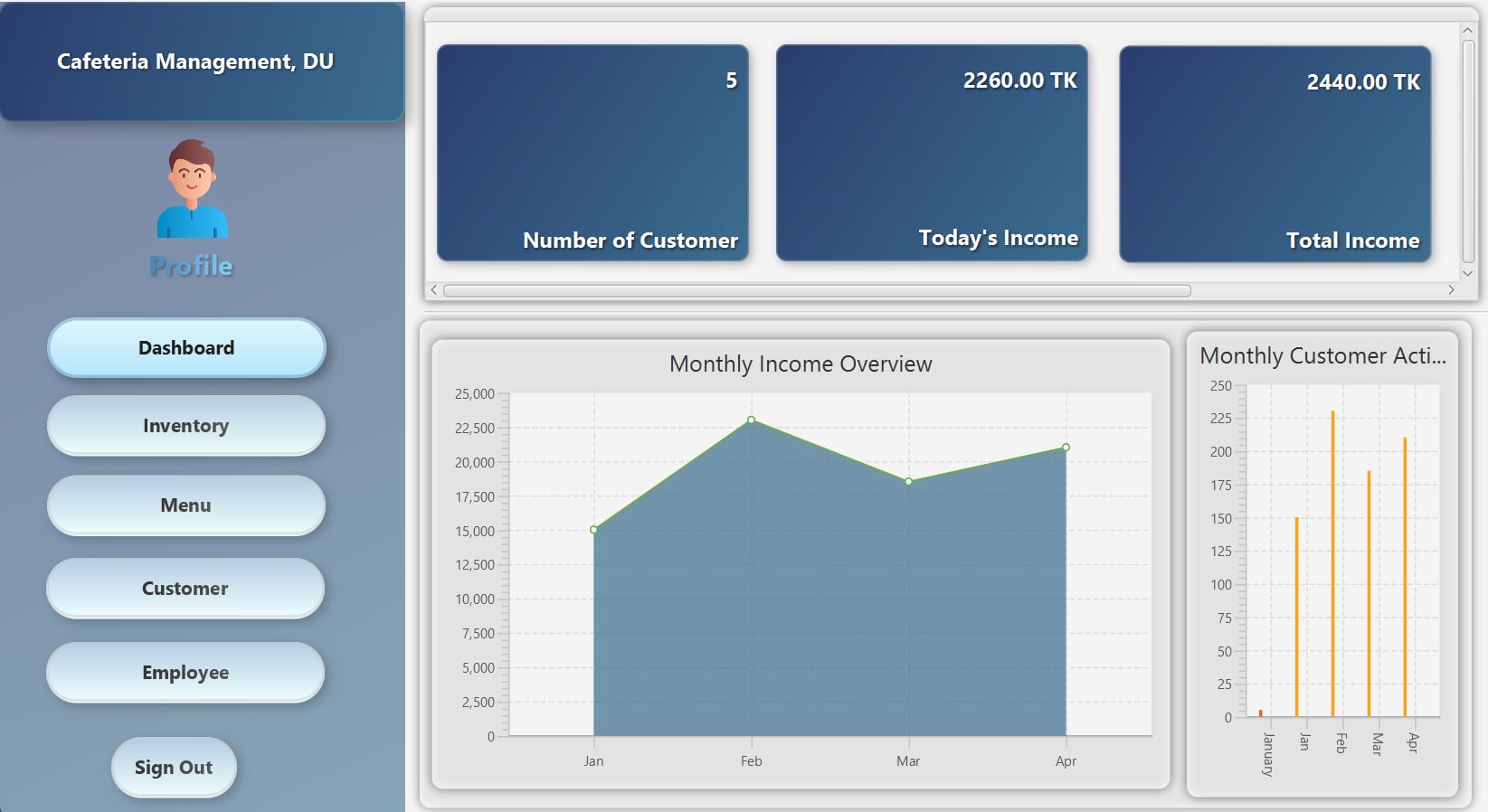


Fig: Dashboard (Admin)

The admin can keep track of income and sold products on the *Dashboard*.

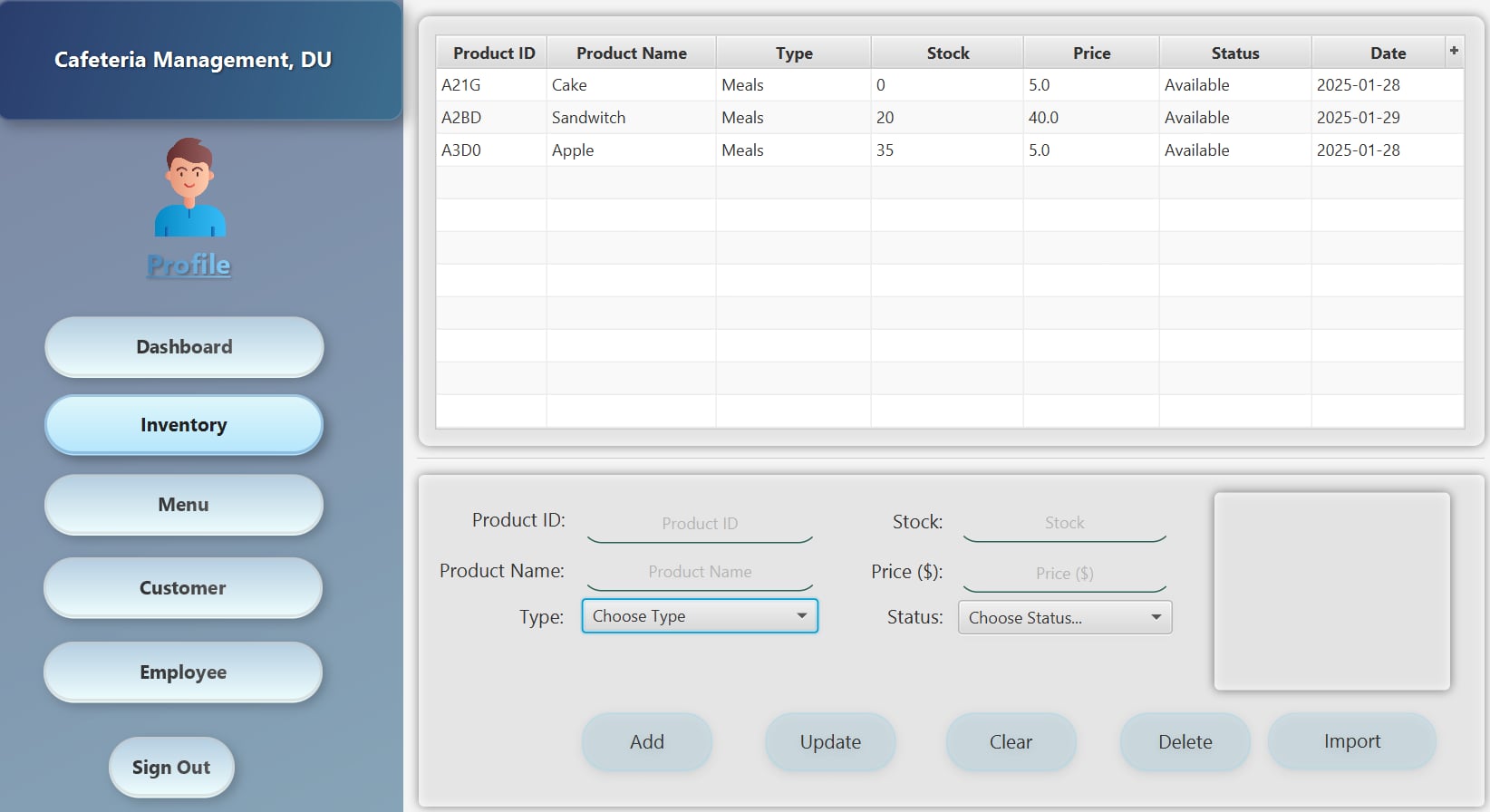


Fig: Inventory (Admin)

Admin can add products, check their availability and unavailability, update product information, and set their prices in *Inventor*y.

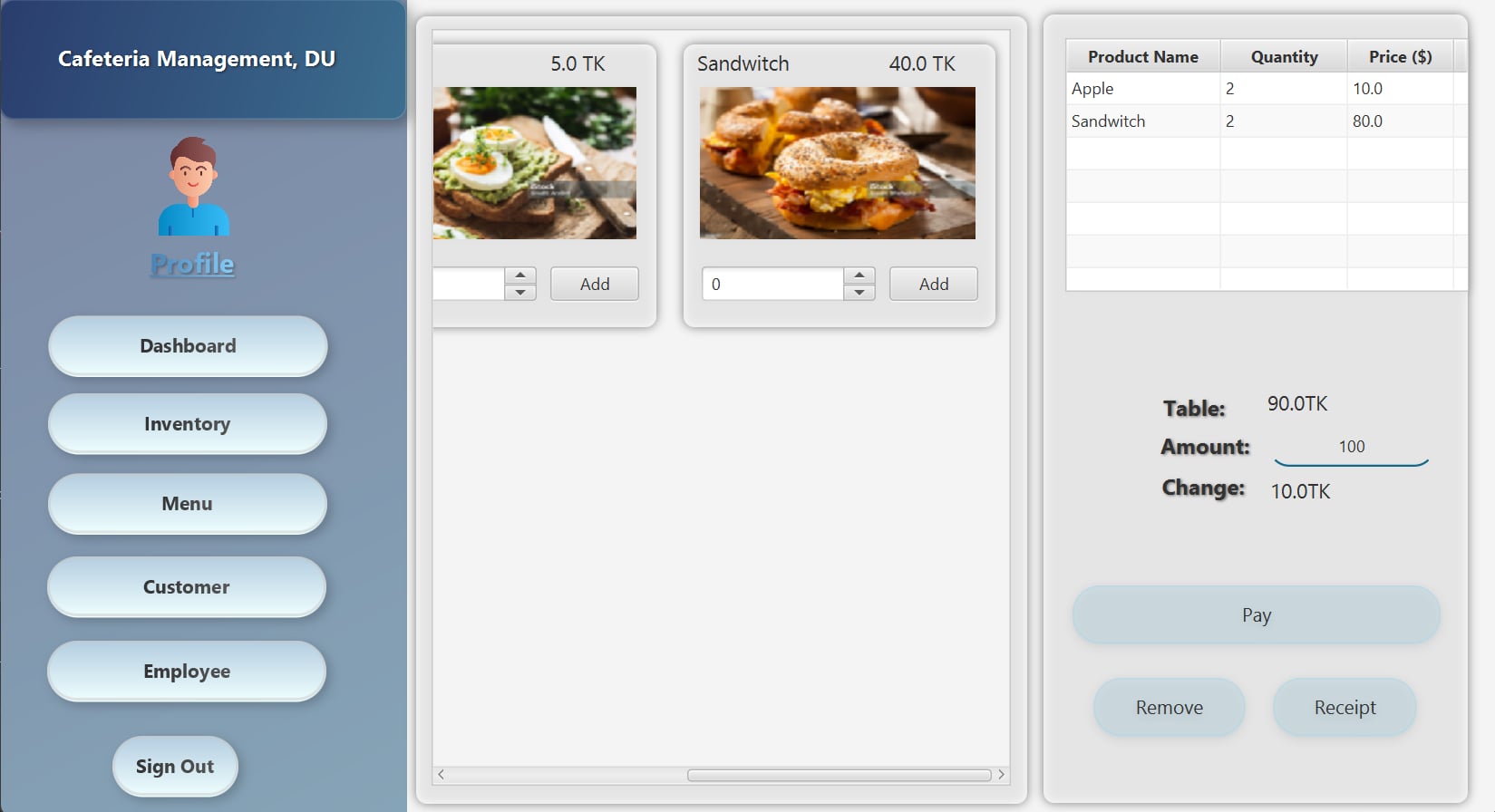


Fig: Menu (Admin)

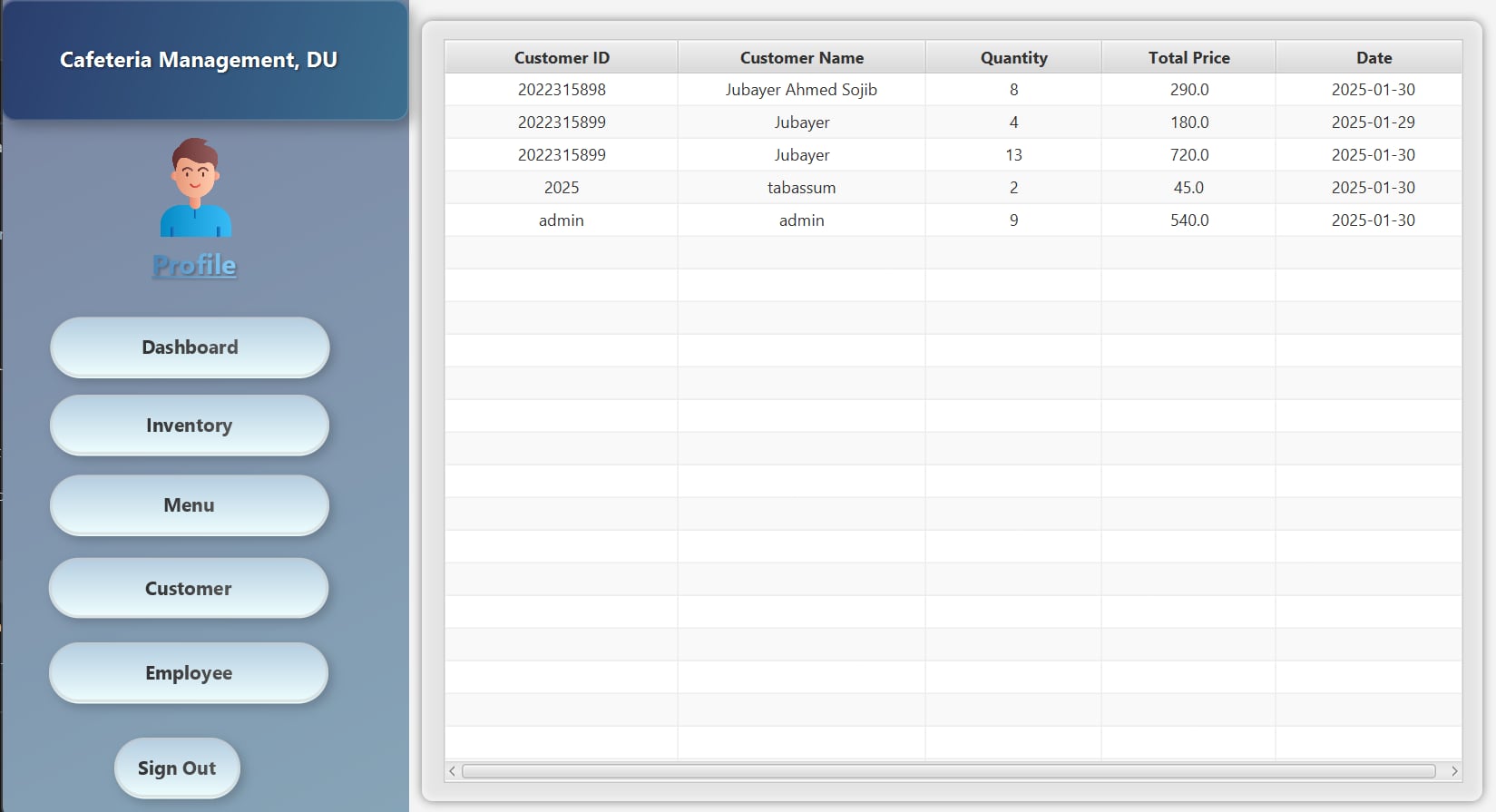


Fig: Customer (Admin)

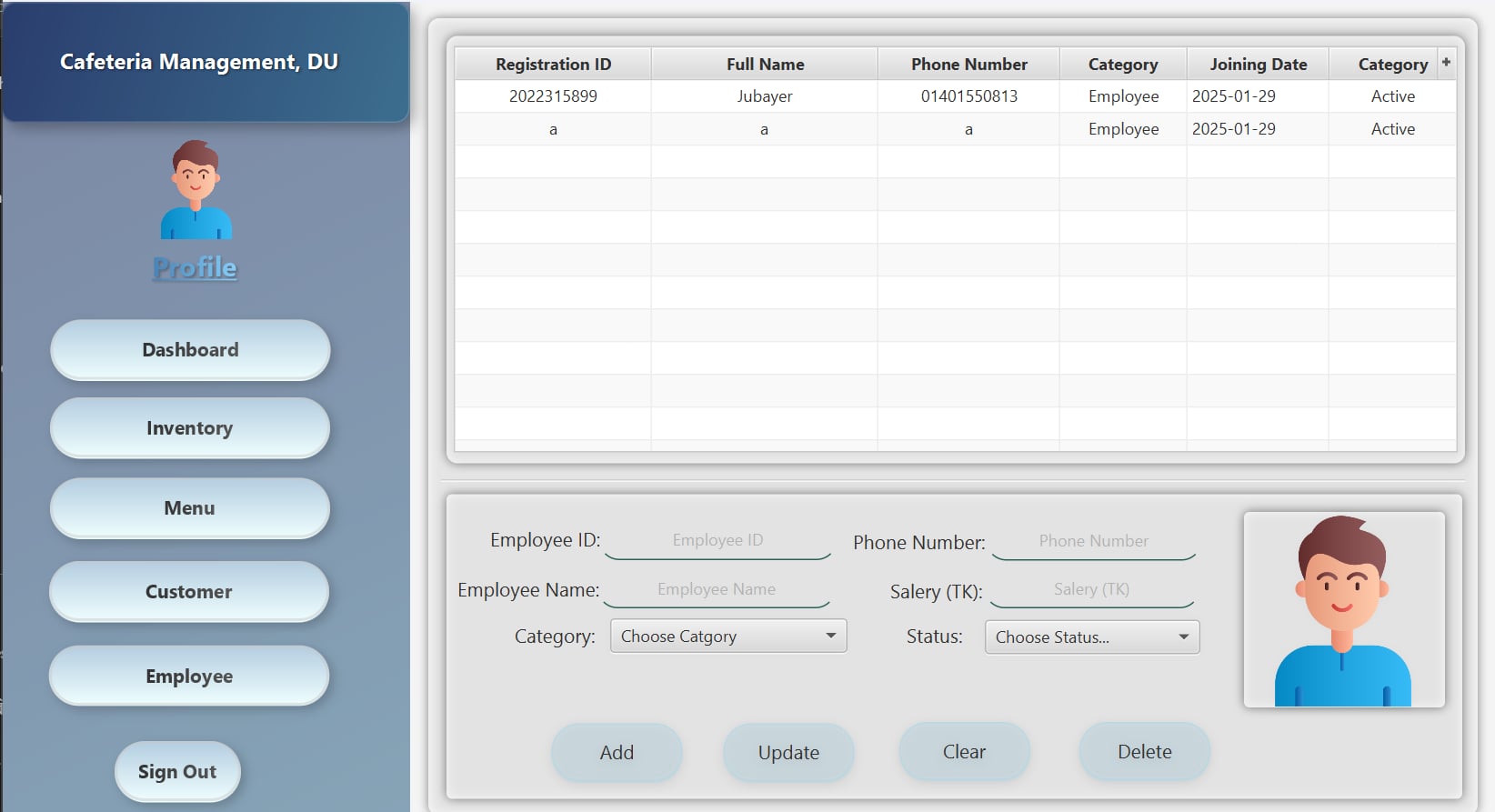


Fig: Employee (Admin)

The admin can check the menu on the *Menu* after adding or removing products from the *Inventory*. He can view customers' purchase history on the *Customer* and add or remove employees on the *Employee*.

Customer (Student & Teacher):

The customer page has *Profile*, *Menu*, *Product review* and *About Us* section as features.

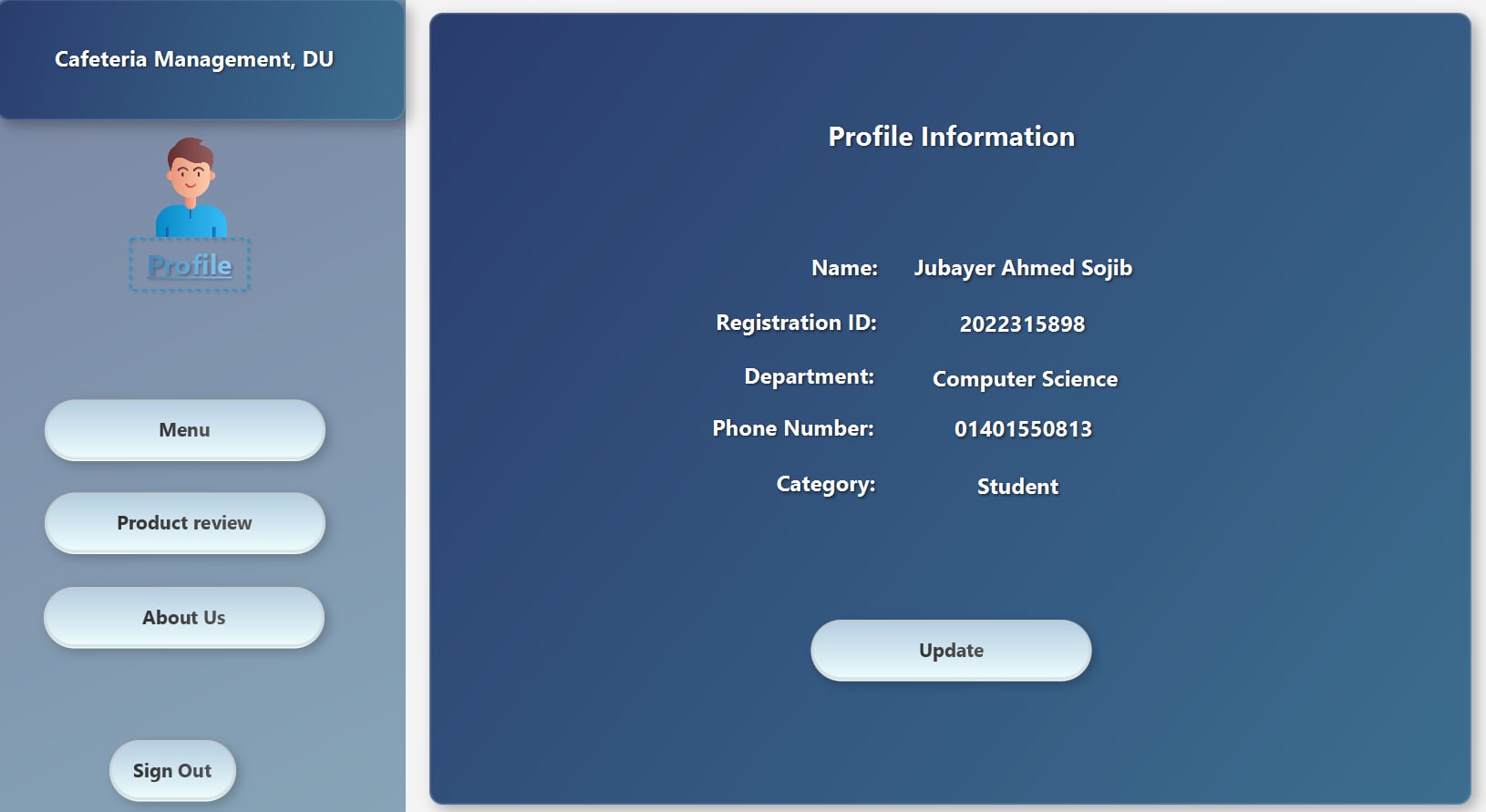


Fig: Profile (Customer)

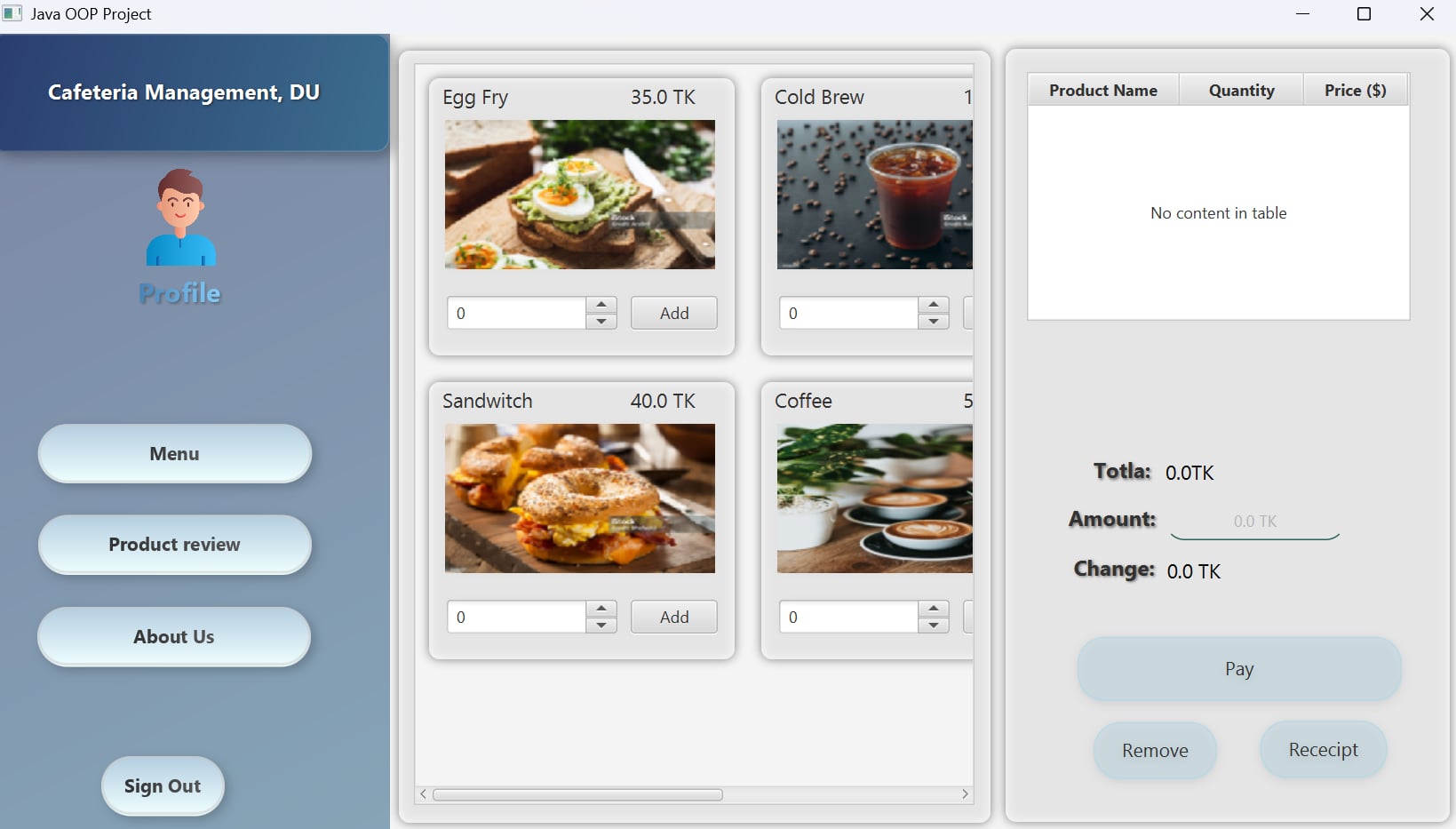


Fig: Menu(Customer)

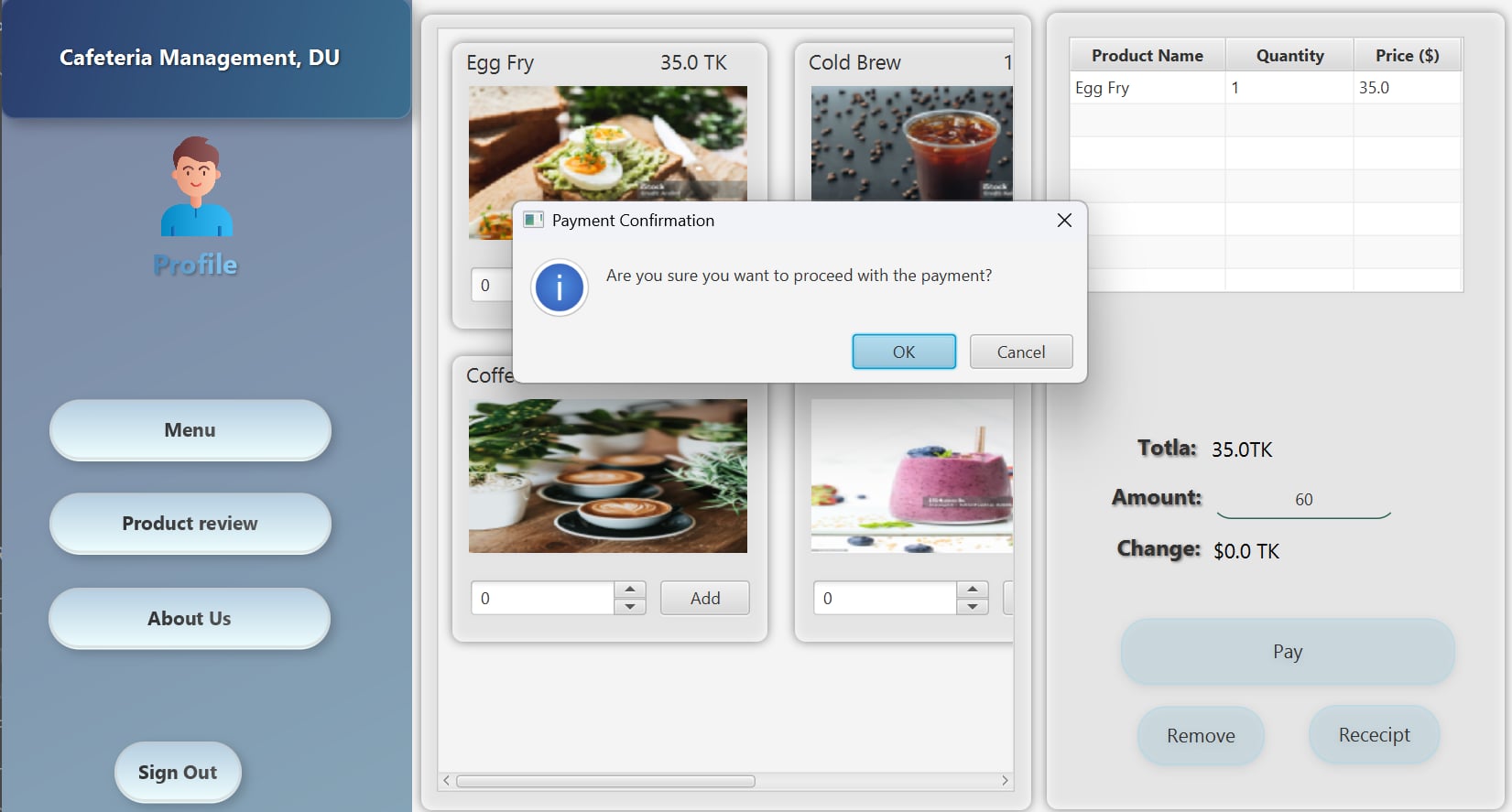
Customers can check the *Menu* and add items to their cart.

Fig: Payment

A confirmation will be sent after payment if you have sufficient balance in your account. The remaining balance(change) will then be calculated.

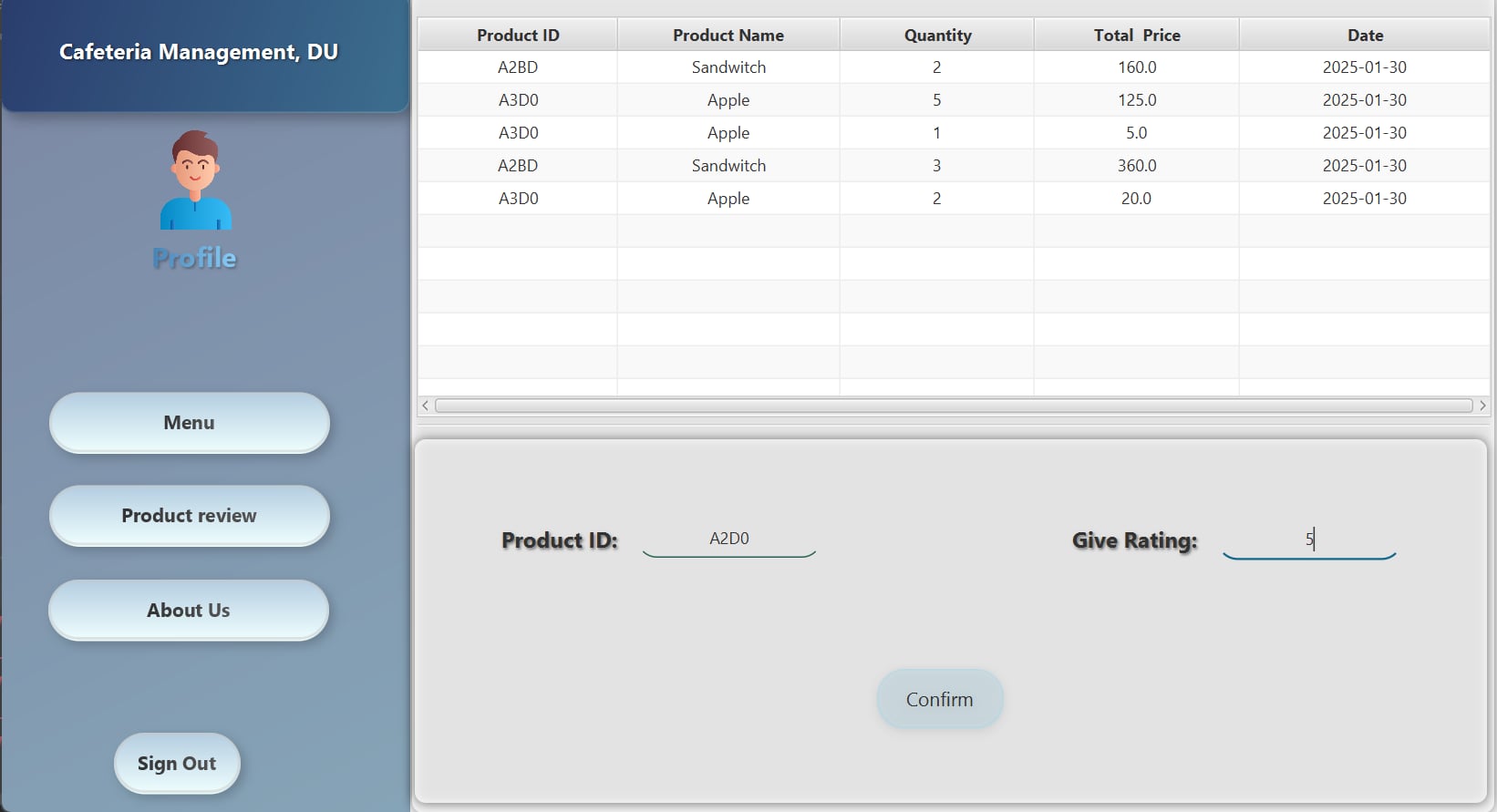


Fig: Product review

Customers can review the product here in *Product review*.



Fig: About Us

Customers can learn about this cafe in *About Us*. They can view the location and contact the cafe via phone.

Employee:

Employees in this system play a crucial role in managing daily operations efficiently. They have access to various features that assist them in handling customer interactions and maintaining the cafeteria's workflow.

Employees can update their personal details through *profile* management and access the *dashboard* to view important statistics such as daily sales and *inventory* status. They are responsible for managing the *inventory* by adding, updating, or removing food items from the menu. Additionally, they can track *customer* orders and purchase history to ensure smooth service. While employees have significant control over the system, they cannot add or remove other employees, a privilege reserved for the admin. However, they can perform almost all other administrative tasks necessary for efficient cafeteria management.

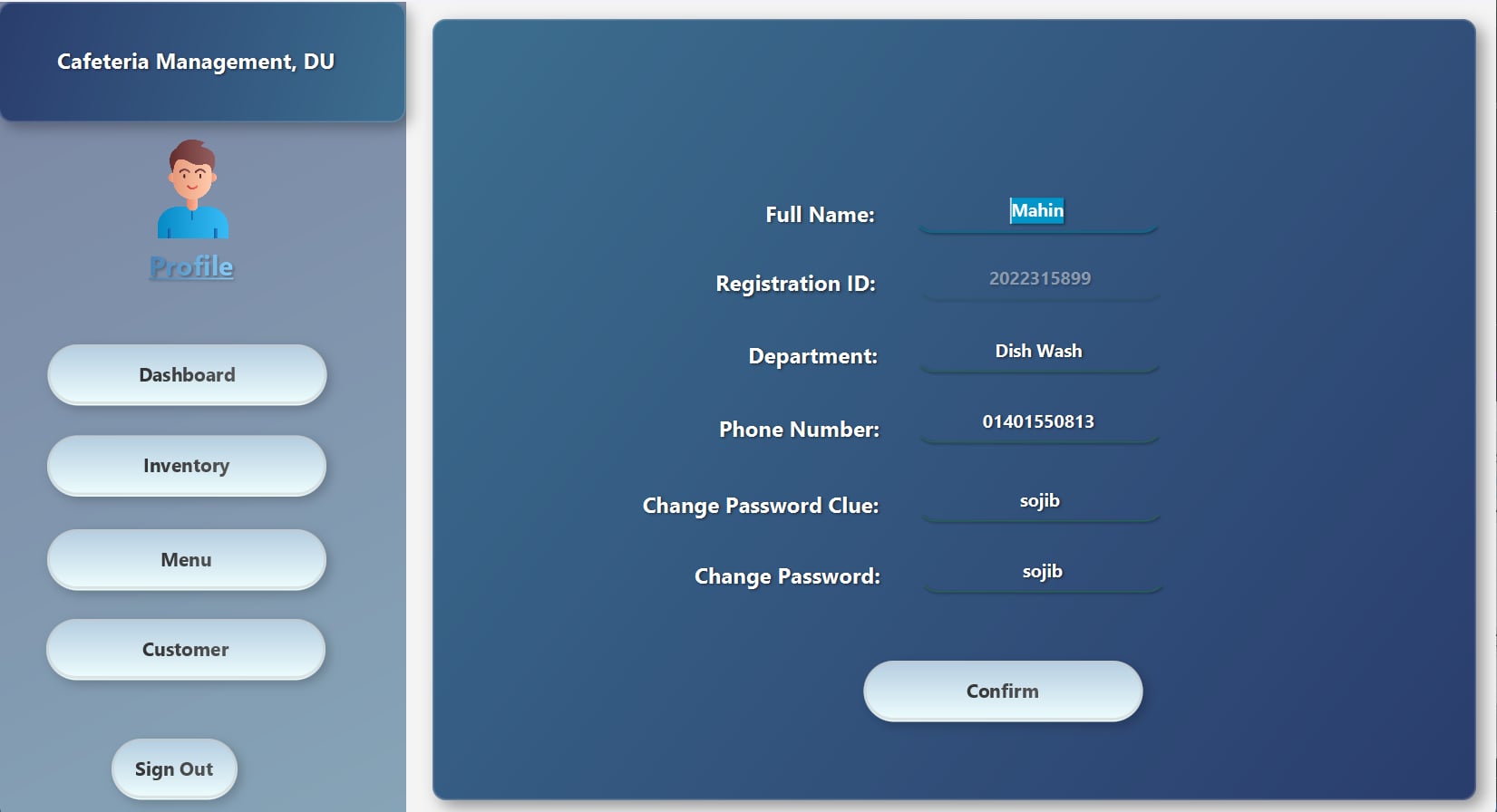


Fig: Profile (Employee)

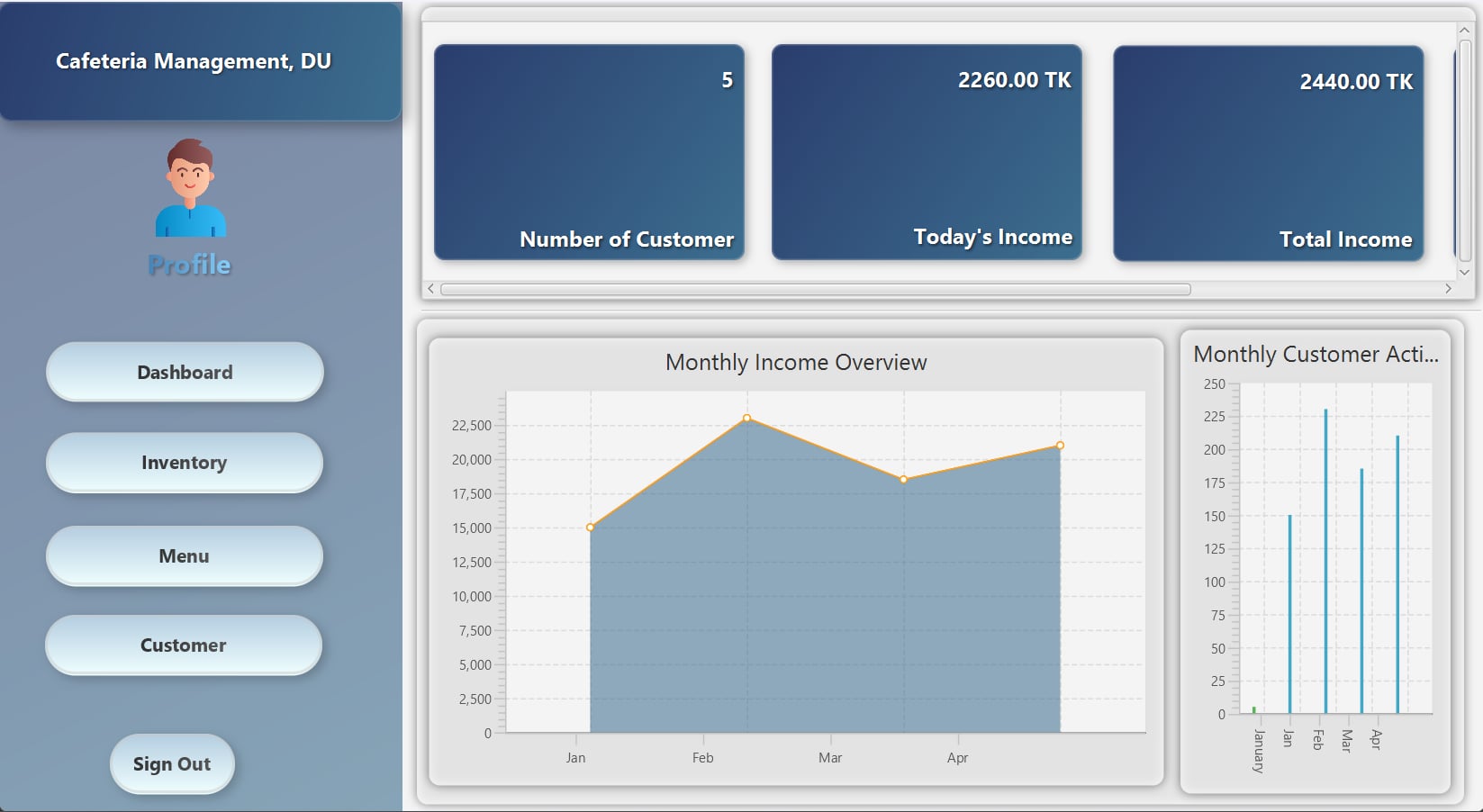


Fig: Dashboard (Employee)

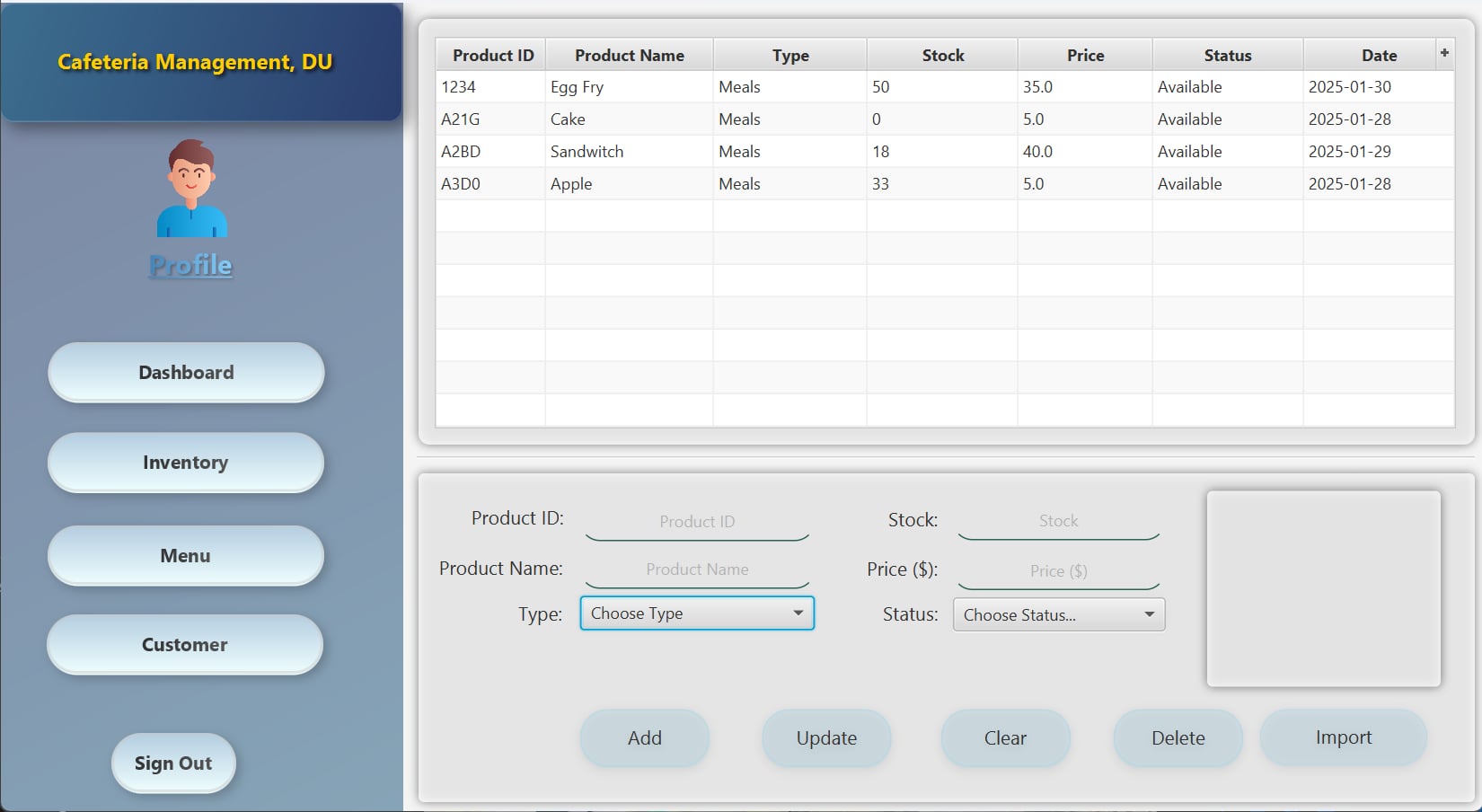


Fig: Inventory (Employee)

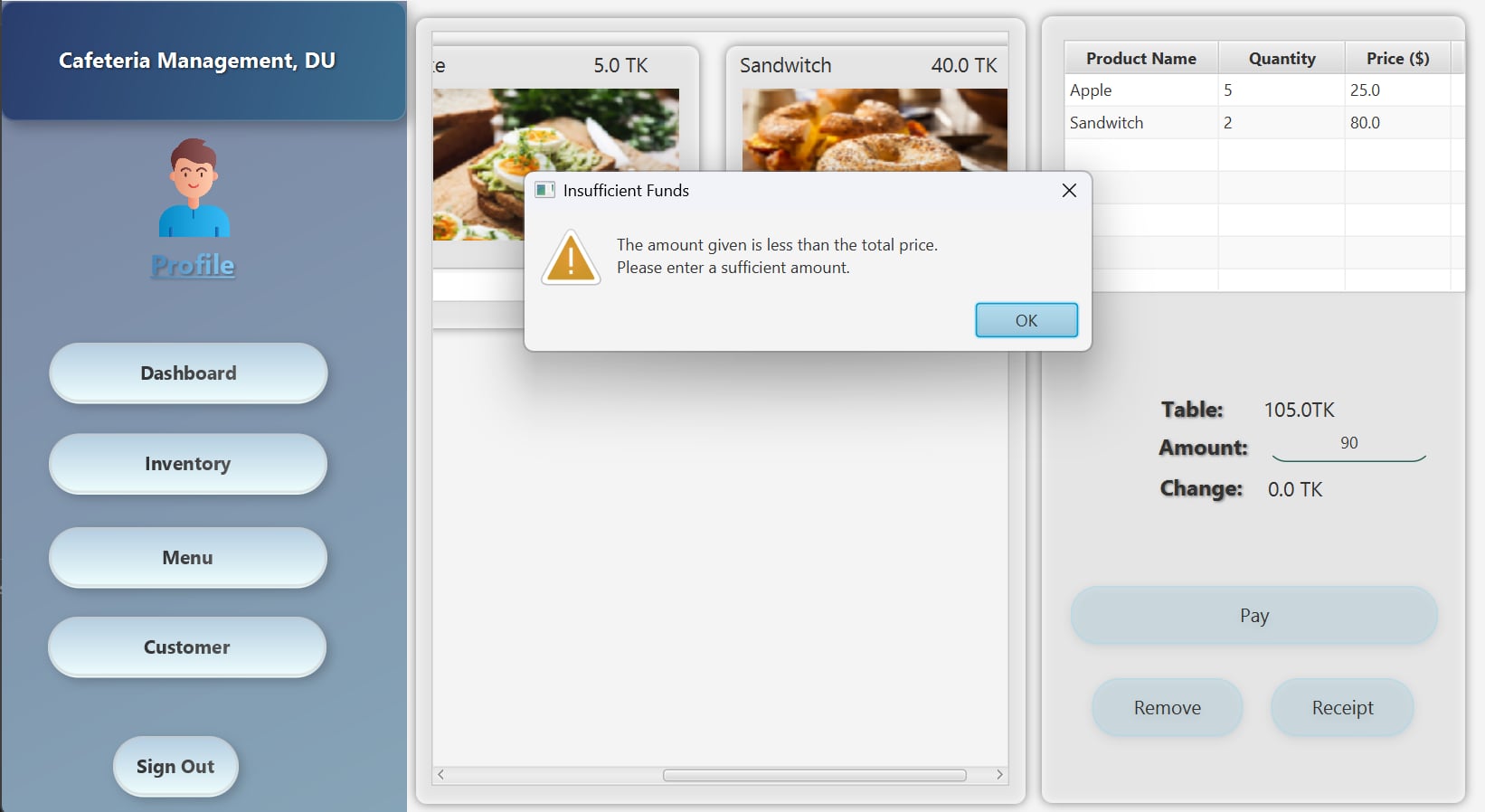


Fig: Menu (Employee)

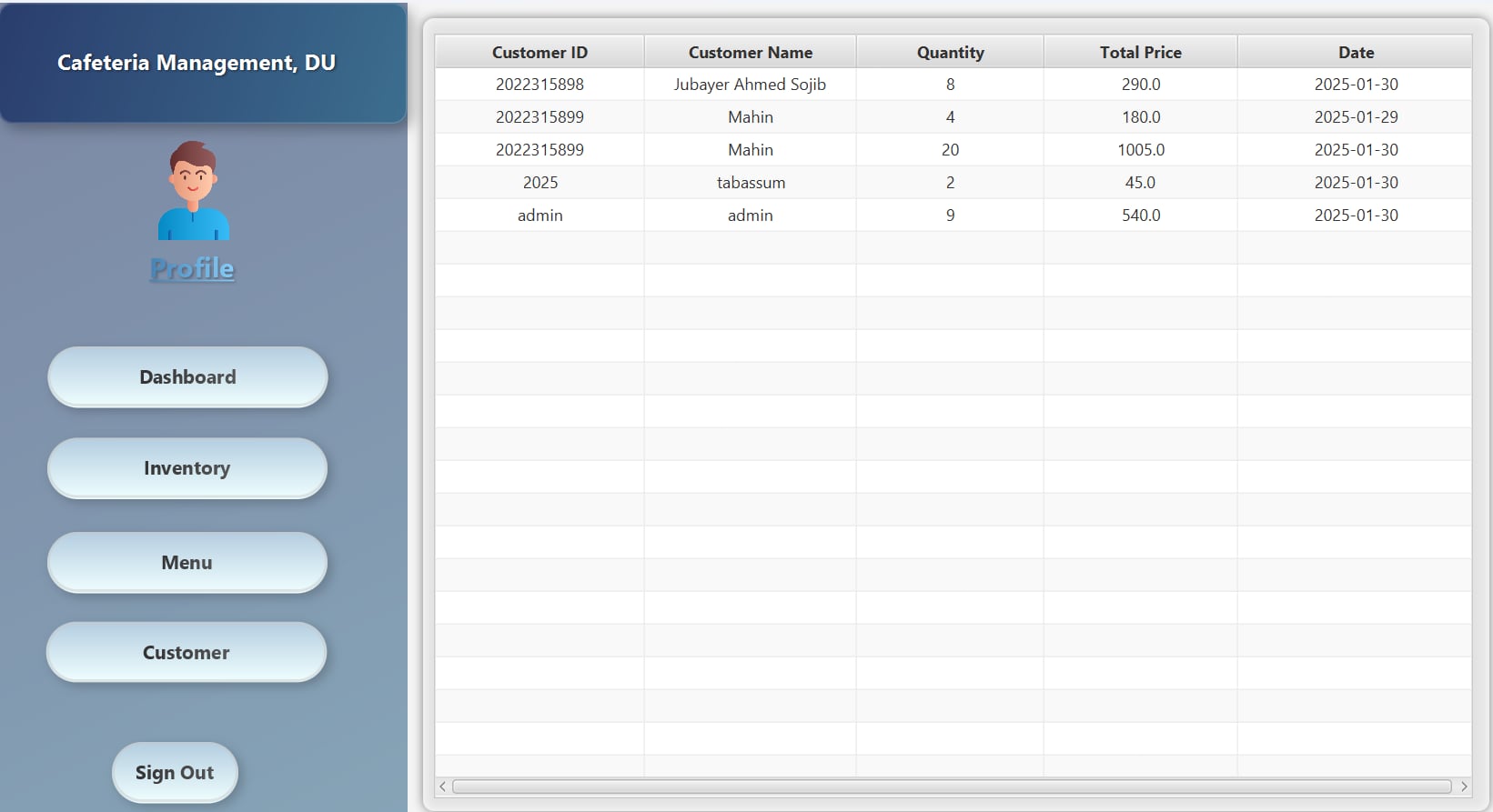
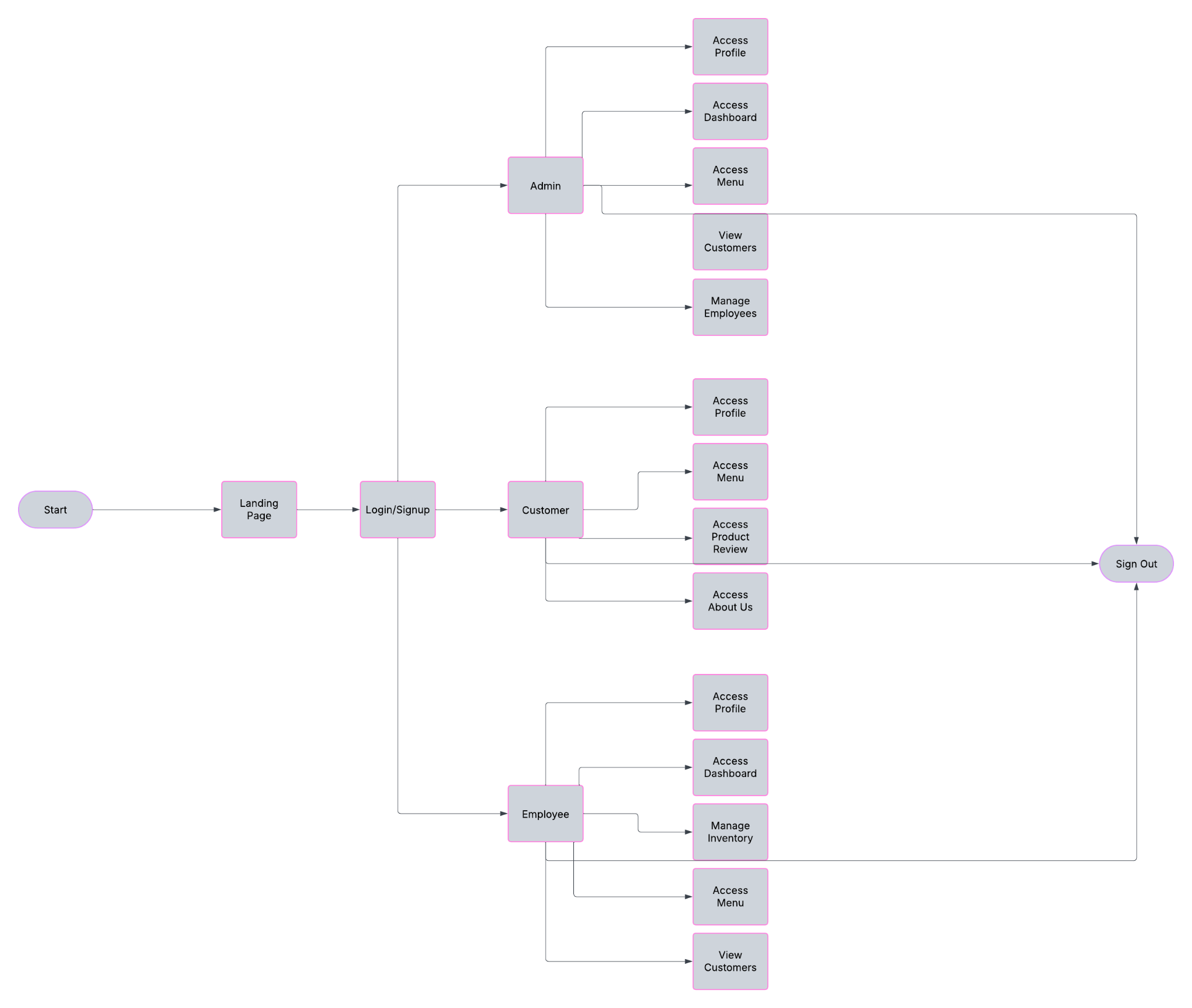


Fig: Customer (Employee)

2.2 Workflow



2.3 Implementation of OOP Principles

Our *Cafeteria Management, DU* is built using Object-Oriented Programming (OOP) principles to ensure scalability, maintainability, and modularity.

* Encapsulation: User data, menu items, and transactions are securely managed through well-defined classes with restricted access.
* Inheritance: The system follows a structured hierarchy, where Admin, Employee, Teacher, and Student inherit common attributes from a User superclass.
* Polymorphism: Different user types interact with the system through a unified interface while performing distinct roles.
* Abstraction: Complex operations like order processing, authentication, and data retrieval are encapsulated within specialized classes, simplifying implementation.

3. Conclusion

3.1 Challenges

In the early stages, we aimed to develop a complete cafe management system that would provide both the admin and customer panels with an easy and reliable platform. However, it is currently limited to one café in DU. Despite this, we have applied our OOP knowledge and refined our software design skills, tackling real-world challenges and devising solutions along the way.

3.2 Future Plans

We plan to integrate a chatbot into the *About Us* section to provide users with instant information about the cafeteria. Additionally, we aim to enhance the system’s usability by optimizing it for mobile devices, ensuring a seamless experience across all platforms. However, achieving these goals will require significant development and refinement.

3.3 Repositories

Github Repository: <https://github.com/Clear20-22/Java-Project>

YouTube Video Link: <https://www.youtube.com/watch?v=Bo50VhnS6qc&t=13s>