Final Report

Project Description

For our project, we built an ordering system application for a restaurant. This application consists of three main parts: the user interface, the backend code, and the database.

The **user interface (UI)** was developed using Java Swing. It features separate panels for customers and admins. Customers can view the restaurant menu, add items to their cart, place orders, and make payments. Admins can log in and manage the menu by adding, editing, or deleting items.

The **backend code** contains eight separate classes:

- **Admin** – manages admin login and menu control
- **Customer** – holds customer information and reservation details
- **MenuItem** – represents individual items on the menu
- **Order** – tracks customer orders
- **OrderItem** – represents each item within an order
- **Restaurant** – the main class that launches the GUI and manages app flow
- **Test** – used for initial testing
- **User** – serves as a base class for both admin and customer accounts

The original database contains six tables, including ***admin***, ***customer***, ***user***, ***payment***, ***order,*** and ***menuItem***.These are used to store login credentials, customer details, menu items, orders, and payment information. However in our project we ended up only using the **Admin and MenuItem tables**.

User Guide

To run the application, users need to:

1. Make sure the Java Development Kit (JDK) is installed.
2. Launch the application by running the `Restaurant` class in an IDE like IntelliJ or Eclipse.
3. Make sure the MySQL database is running and connected properly. The connection settings (URL, username, password) should be updated in the backend before launching the app.

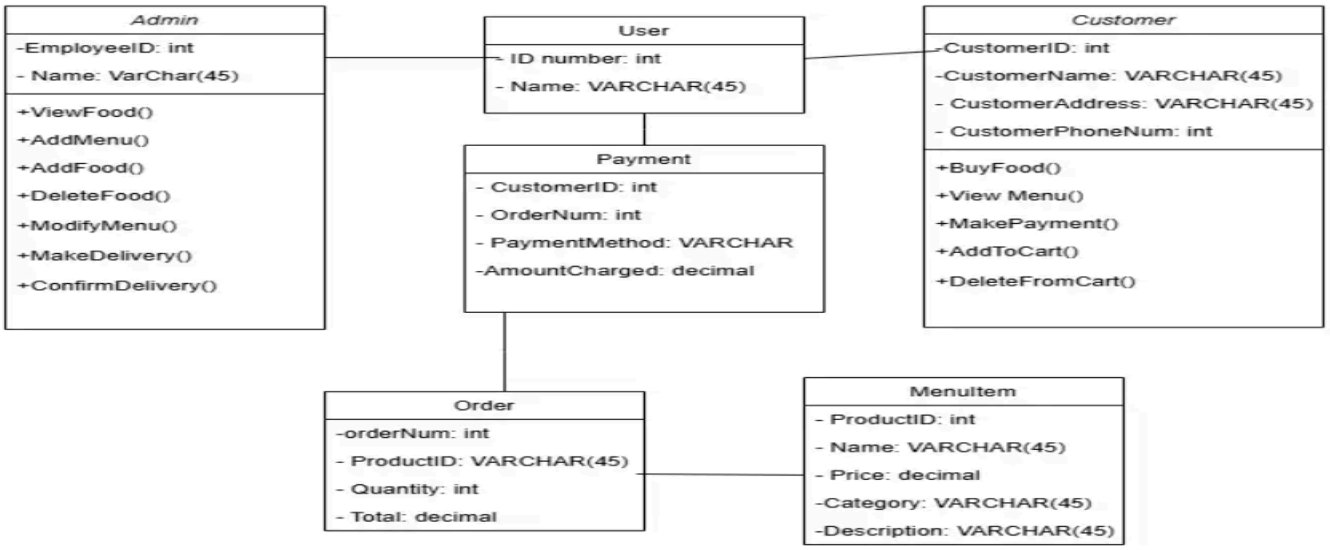
Customer View:

- When the app starts, customers see the restaurant menu.
- They can click on menu items to add them to their cart.
- Once finished, they can view their cart, see the total, and proceed to payment.

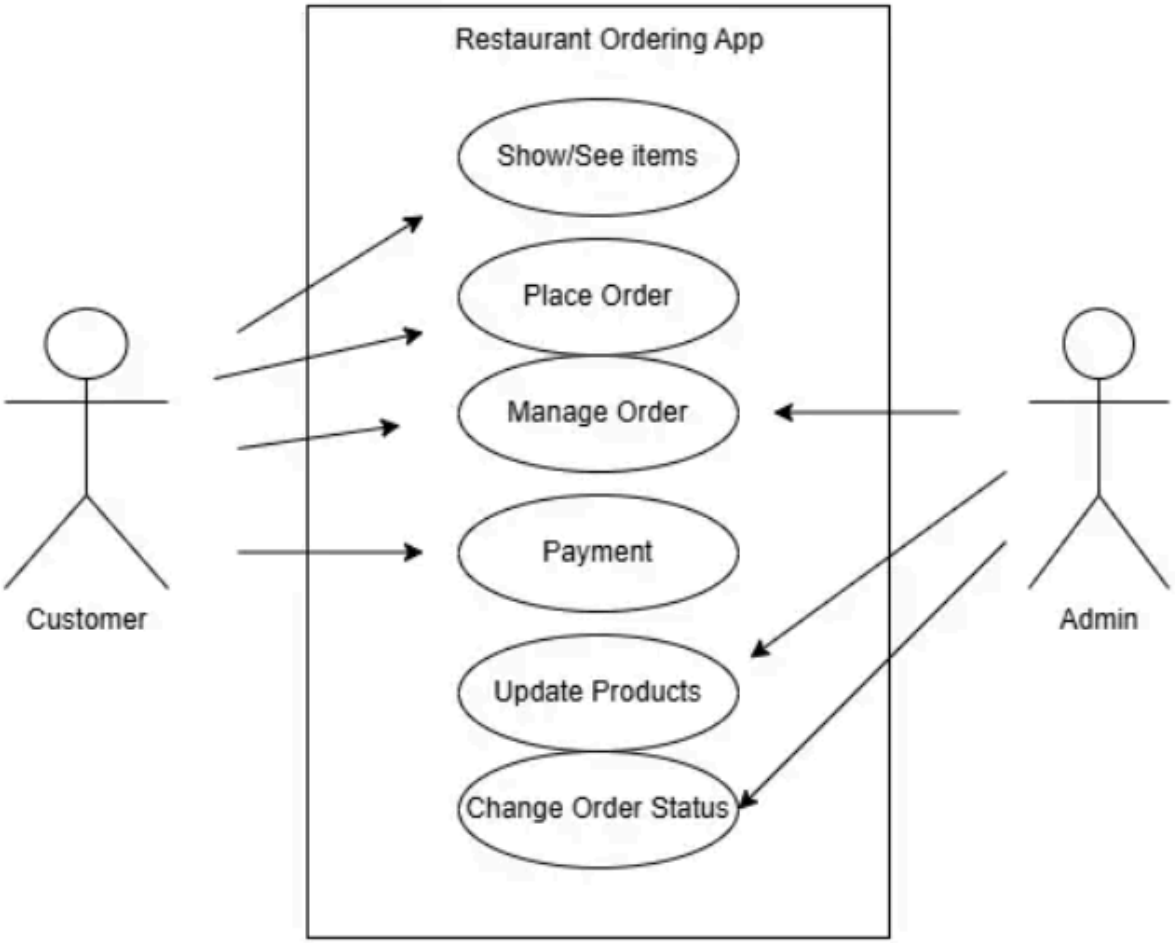
Admin View:

- Admins can log in using a username and password stored in the database.
- The admin's credentials are "***admin***" and the password is "***password123***".
- After logging in, they're taken to the admin panel where they can:
    - Load the current menu
    - Add new items
    - Edit existing ones
    - Delete items

UML Diagrams



Use Case Diagrams



Testing Overview

For testing purposes, we created a test class where we used informal unit testing to test our entire project. The test class manually goes through every class and tests every method within the classes. The following is what we tested:

- **Admin class:** Constructing an admin, getName function, getEmployeeID function, setEmployeeID, and then printing the updated employee ID.
- **Customer class:** Constructing a customer, getName function, getCustomerAddress function, getCustomerPhone function, setCustomerAddress function, and printing the updated address.
- **Menu Item class:** Constructing a menu item, getName function, getPrice function, setName function, and printing the updated name.
- **Order class:** Construction a new order, addItem function, and printing out the order.
- **Order Item class:** Constructing an orderItem object, getItem function, getQuantity function, setQuantity function, printing updated quantity, and printing the total price.
- **User class:** Constructing a user object, getName function, getIDNumber function, setName function, printing the updated name, setIDNumber function, and printing the updated ID number.

Known Bugs

- If the database connection fails (e.g., wrong credentials), the app does not display a user-friendly error message
- Customers can add the same item multiple times, but there's no quantity field to indicate how many of each item they want.

Possible Feature Additions

- Add a **login and account system** for customers so they can track past orders.
- Allow **online order tracking** and **email confirmations**.
- Add **quantity selectors** to the cart.
- Improve the **payment system** to support different simulated payment method
- Include **analytics** on best-selling items for the admin dashboard.