University of Dhaka

Department of Computer Science and Engineering

Project Report

Object Oriented Programming (CSE-2112)

Project Name

FoodiePal -(The Ultimate Restaurant Management System)

Submitted to

- Dr. Md Ibrahim
- Md. Ashraful Islam

Submitted by

- Abrar Eyasir
  Roll: FH-12

- Dipankor Roy
  Roll: JH-28

- Abdullah Ashik

Roll: SH-32

Introduction:

FoodiePal is a Java and JavaFX-based project designed to help restaurant owners and managers maintain their cash registers and accounts. The application allows the user to enter the details of the restaurant's menu, prices, and orders, which are then stored in a database. The system also allows the user to generate bills, calculate discounts, and generate reports on sales, profits, and expenses. With FoodiePal, restaurant owners can streamline their accounting processes, reduce errors, and increase efficiency. The project is ideal for small to medium-sized restaurants looking for a simple and user-friendly way to manage their finances.

Requirement Analysis:

Requirement analysis is an important step in the software development process that helps to ensure that the end product meets the needs and expectations of the stakeholders. In the case of the FoodiePal project, the following requirements should be analyzed:

Functional Requirements:

- The application should allow the user to add, edit, and delete menu items, including their prices and descriptions.
- The system should support different types of discounts such as percentages or fixed amounts.
- The system should allow the user to generate bills for customers and print them out.
- The application should be able to calculate the total sales and profits for a given period.

- The system should be able to generate reports on sales, expenses, and profits.

Non-Functional Requirements:

- The application should be user-friendly and easy to navigate.
- The system should be reliable and accurate.
- The application should be responsive and able to handle a large number of transactions simultaneously.
- The system should be secure and protect sensitive information such as customer details and financial data.

User Requirements:

- The application should be accessible to users with different levels of technical expertise.
- The system should be customizable, allowing users to configure settings such as tax rates and currency formats.
- The system should have a fast response time and minimize wait time for customers.
- The system should provide users with real-time updates on inventory levels and order status.

Operational Requirements:

- The system should be compatible with different operating systems such as Windows, MacOS, and Linux.
- The application should be scalable and able to accommodate future growth in the business.
- The system should have a backup and recovery mechanism in case of data loss or system failure.
- The application should be easy to install and maintain.

System Design

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system.
The system design of the FoodiePal project can be broken down into several components:

User Interface:

The user interface should be designed to be user-friendly and easy to navigate. It should consist of screens for adding/editing/deleting menu items, generating bills, and viewing reports. The user interface should also have options for configuring settings such as tax rates and discounts.

Database:

The system should have a database for storing menu items, customer information, and sales data. The database should be designed to be scalable and efficient, allowing for quick retrieval and storage of data.

Payment Gateway:

The system should have a payment gateway integration to allow customers to pay for their orders online through Bkash, Nagad, Rocket, etc. The payment gateway should be secure and reliable, ensuring the safety of customer payment information.

Reporting:

The system should have reporting capabilities to generate reports on sales, expenses, and profits. The reports should be customizable, allowing users to specify the date range and type of report to be generated.

Security:

The system should be designed with security in mind, protecting sensitive data such as customer information and financial data. The system should implement security measures such as encryption and user authentication to ensure that only authorized users can access the system.

Performance:

The system should be designed to handle a large number of transactions simultaneously, ensuring that the application does not crash or slow down during peak hours. The system should be scalable and designed to handle future growth in the business.

Maintenance:

The system should be designed to be easy to maintain, with options for regular updates and bug fixes. The system should also have a backup and recovery mechanism in case of data loss or system failure.

Class Hierarchy

The possible hierarchy of FoodiePal project is given below.

MenuItem: This class represents a menu item, which includes the name, description, price, and any other relevant information.

OrderItem: This class represents an item ordered by a customer, which includes the menu item, quantity, and any special instructions.

Order: This class represents an order placed by a customer, which includes one or more order items, the total cost, and any discounts applied.

Customer: This class represents a customer who places an order, which includes the customer's name, address, and contact information.

Cashier: This class represents a cashier who processes orders, which includes the cashier's name, ID, and login credentials.

Report: This class represents a report generated by the system, which includes information on sales, expenses, and profits.

Database: This class represents the database used to store information on menu items, orders, customers, and cashiers.

PaymentGateway: This class represents a payment gateway used to process customer payments.

UserInterface: This class represents the user interface used to interact with the system, which includes screens for adding/editing/deleting menu items, generating bills, and viewing reports.

SecurityManager: This class represents a security manager used to ensure the security of the system, which includes encryption and user authentication.

Use-Case diagram

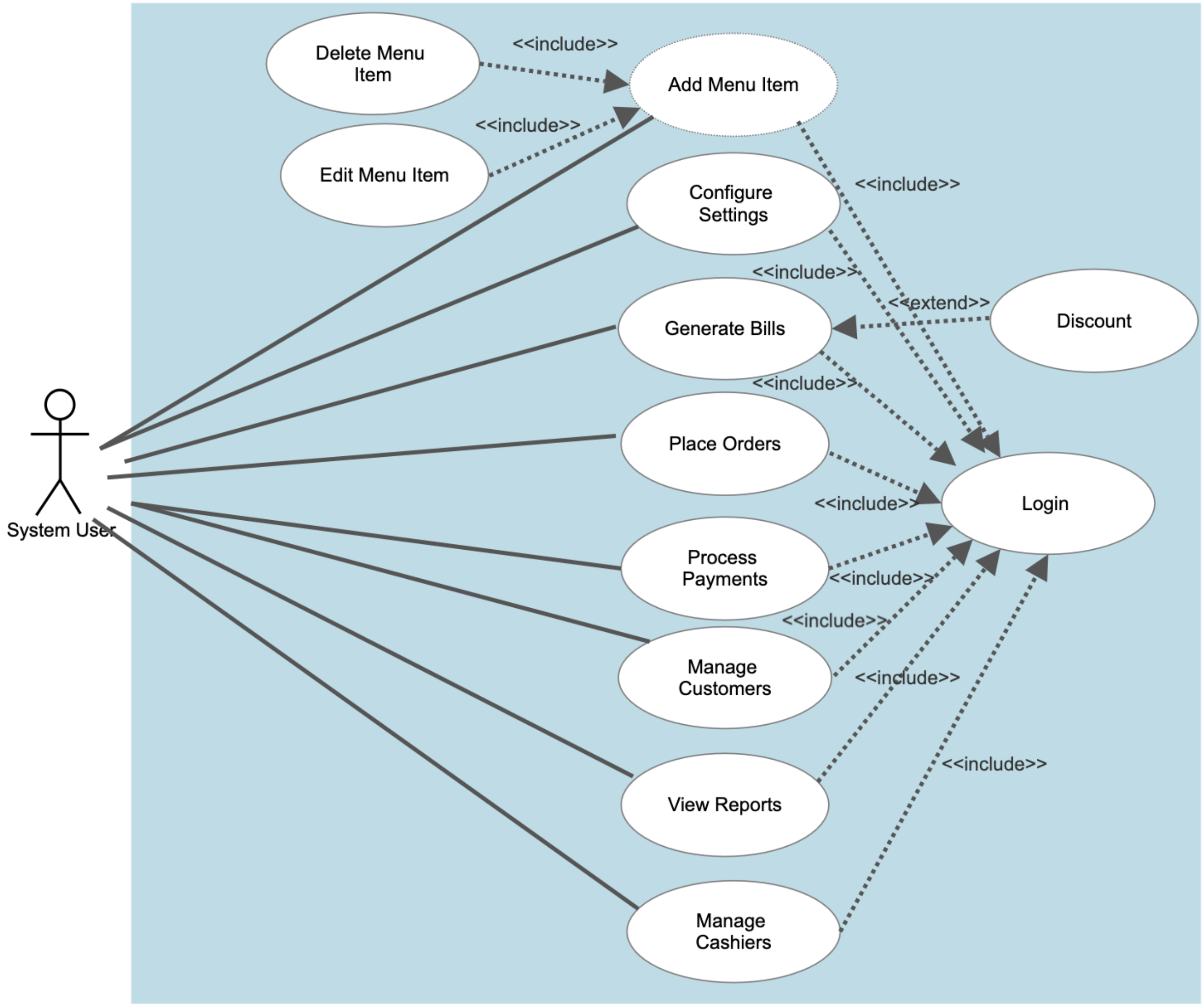

Figure: Use-Case Diagram of FoodiePal

UML diagram

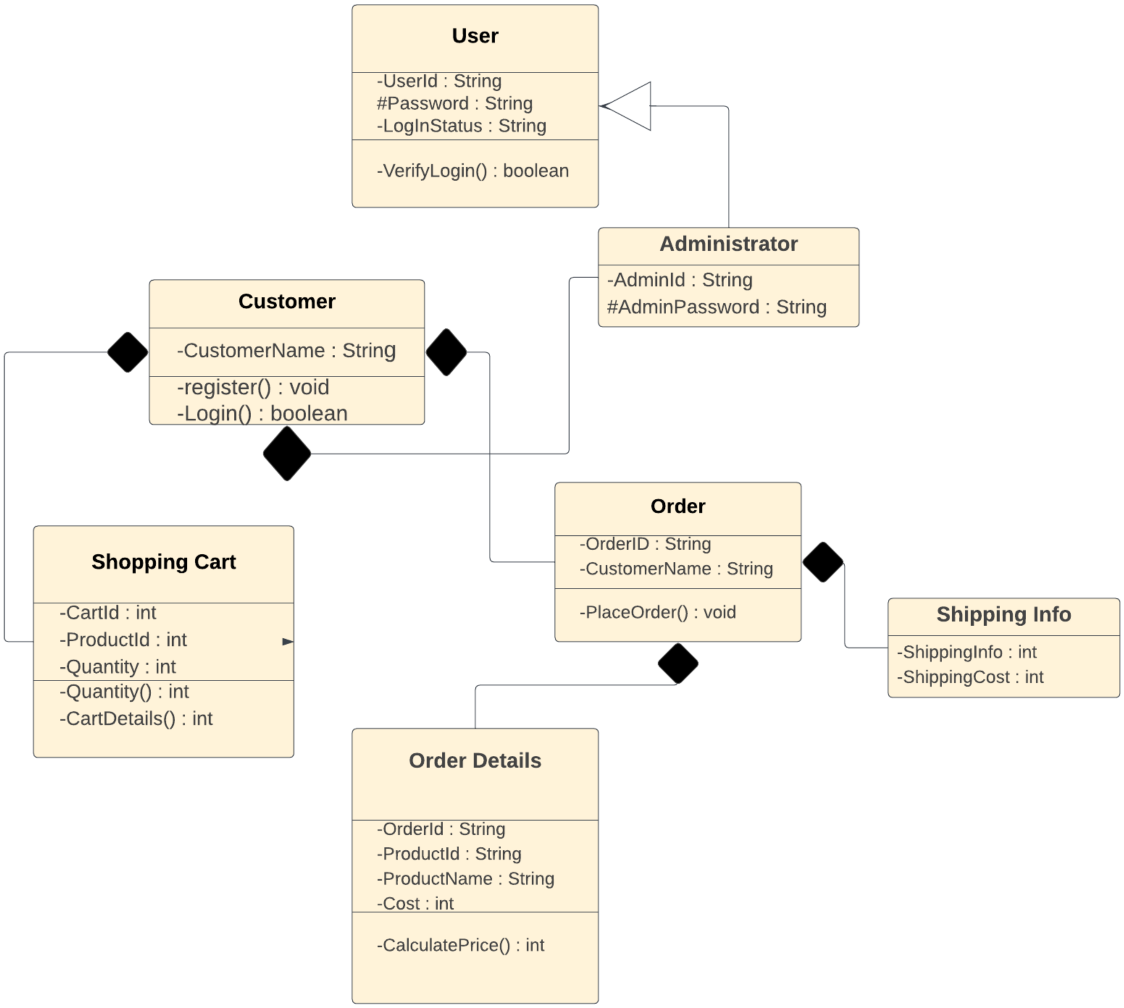

Figure: UML Diagram of FoodiePal

Conclusion:

The FoodiePal project is a Java and JavaFX-based system designed to manage the cashier and accounting functions of a restaurant. The project involves requirements analysis, system design, and implementation of various features such as menu item management, order processing, payment processing, reporting, security, and user interface design. The system is designed to be user-friendly, scalable, and efficient, providing a reliable and secure platform for restaurant owners and cashiers to manage their operations. The success of the Foodipal project will depend on the effective implementation of the various components and adherence to best practices in software development, including testing, documentation, and maintenance. With the right approach, the Foodipal project has the potential to enhance the efficiency and profitability of restaurants by streamlining accounting and cashier functions.