



## FINAL PROJECT REPORT

### OBJECT ORIENTED PROGRAMMING - FALL 2025

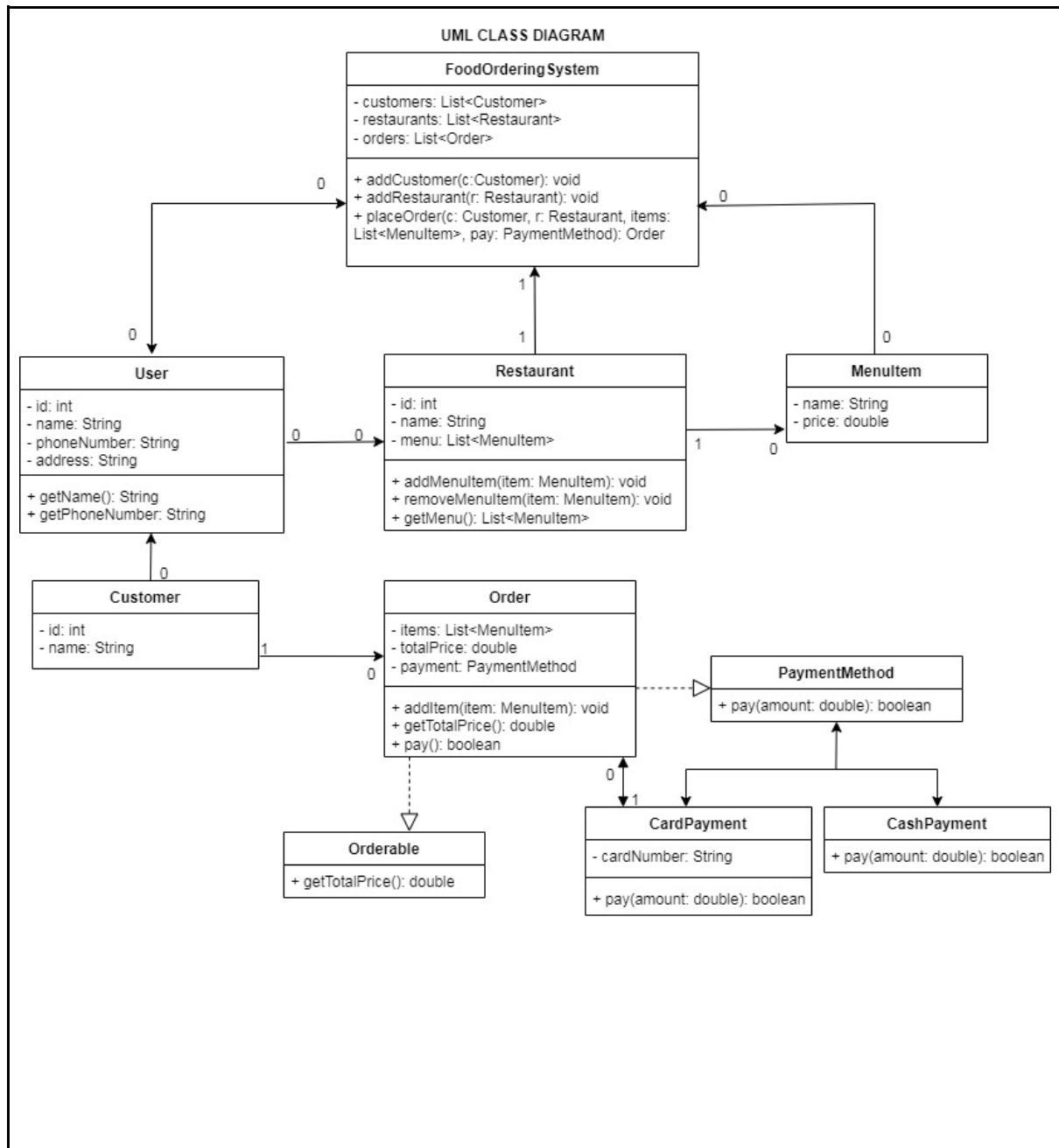
<b>Name, Surname</b>	Issa Konaté
<b>Student ID</b>	220303974
<b>Name of the Project</b>	Online Food Ordering System

### PROJECT DESCRIPTION

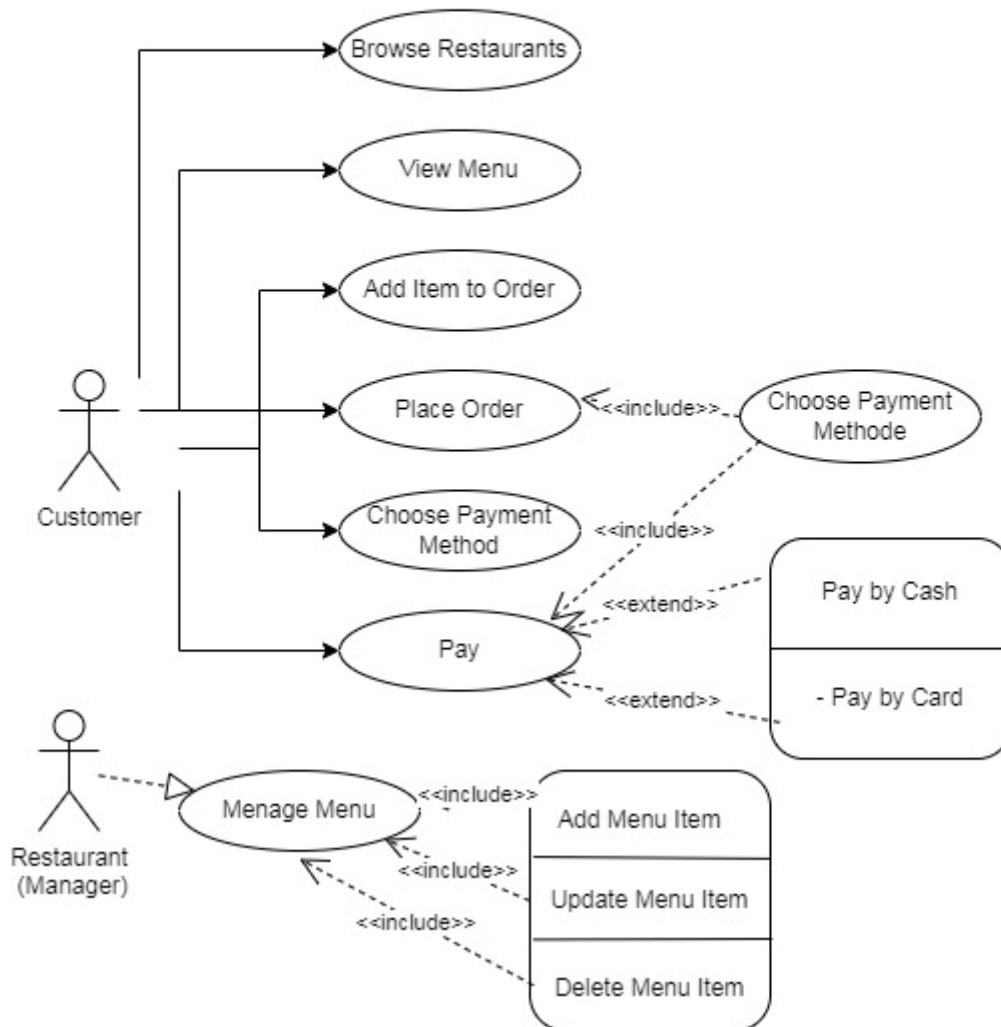
The Online Food Ordering System is a console-based Java application designed to simulate the basic workflow of ordering food from restaurants. The system allows customers to browse restaurants, view menus, add menu items to an order, and complete payments using different payment methods. It aims to model real-world food ordering platforms in a simplified and object-oriented way.

The main users of the system are customers and restaurant managers. Customers can place orders and choose payment methods, while restaurant managers can manage menu items by adding, updating, or deleting them. The project focuses on applying Object Oriented Programming principles such as encapsulation, inheritance, polymorphism, and interfaces, while also using JUnit testing and Git/GitHub to simulate a real software development workflow.

# UML DIAGRAMS



# UML USE CASE DIAGRAM



## SYSTEM DESIGN

<b>Number of classes</b>	8
<b>Number of interfaces</b>	2
<b>Number of abstract classes (if any)</b>	0

### Short explanation of your classes:

<b>ClassName 1</b>	FoodOrderingSystem Acts as the main controller of the application. It manages customers, restaurants, and orders, and coordinates the order placement process.
<b>ClassName 2</b>	User A base class that stores common user information such as name, phone number, and address.
<b>ClassName 3</b>	Customer Extends User. Represents a customer who can place food orders from restaurants.
<b>ClassName 4</b>	Restaurant Stores restaurant information and manages a list of menu items. Provides methods to add, remove, and retrieve menu items.
<b>ClassName 5</b>	MenuItem Represents a food item with a name and price.
<b>ClassName 6</b>	Order Represents a customer's order. Stores selected menu items, calculates the total price, and processes payment.

<b>ClassName 7</b>	PaymentMethod (Interface) Defines a common pay(amount) method for different payment strategies.
<b>ClassName 8</b>	CardPayment / CashPayment Concrete implementations of the PaymentMethod interface that handle different payment types.

**Explain how did you accomplish the following:**

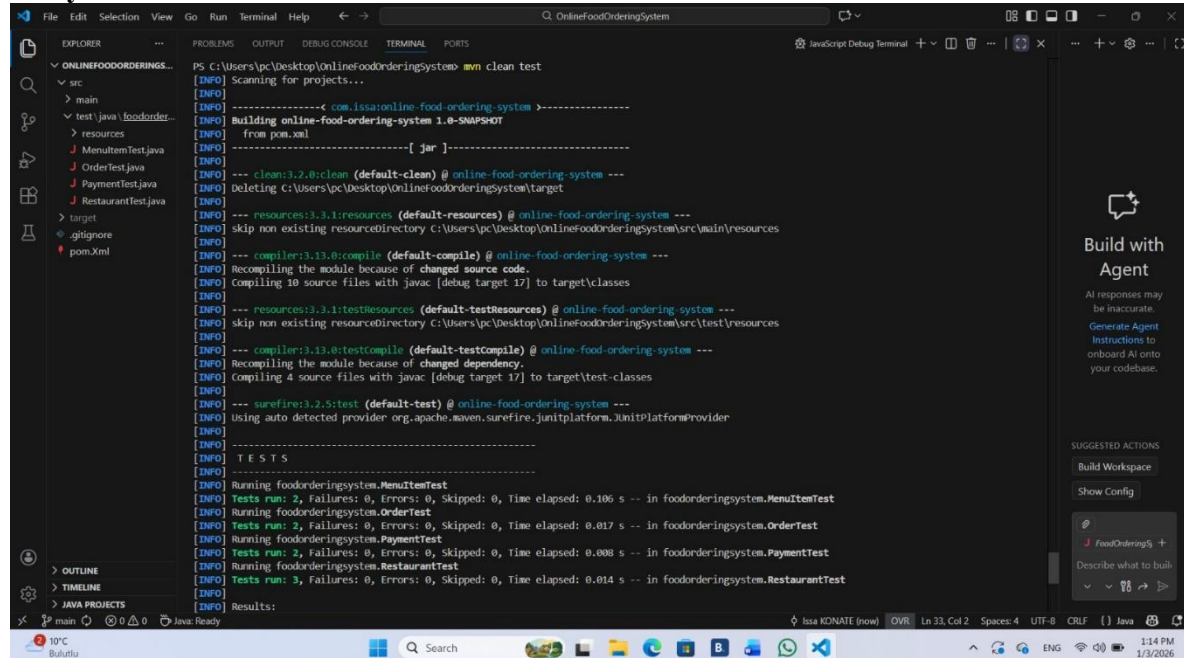
<b>Encapsulation</b>	Encapsulation is achieved by declaring class fields as private and accessing them through public getter and setter methods. This protects data from direct external access and ensures controlled modification.
<b>Inheritance</b>	Inheritance is implemented by having the Customer class extend the User class. This allows reuse of common attributes like name and phone number while adding customer-specific behavior.
<b>Polymorphism</b>	Polymorphism is used in the payment system. The PaymentMethod interface allows the program to process payments using different implementations (CashPayment or CardPayment) without changing the order logic.
<b>Interfaces</b>	Interfaces are used to define contracts. PaymentMethod enforces a common payment behavior, while Orderable defines pricing behavior, making the system flexible and extensible.

# UNIT TESTING

How many classes you've tested?

4

Put your JUnit screenshot here



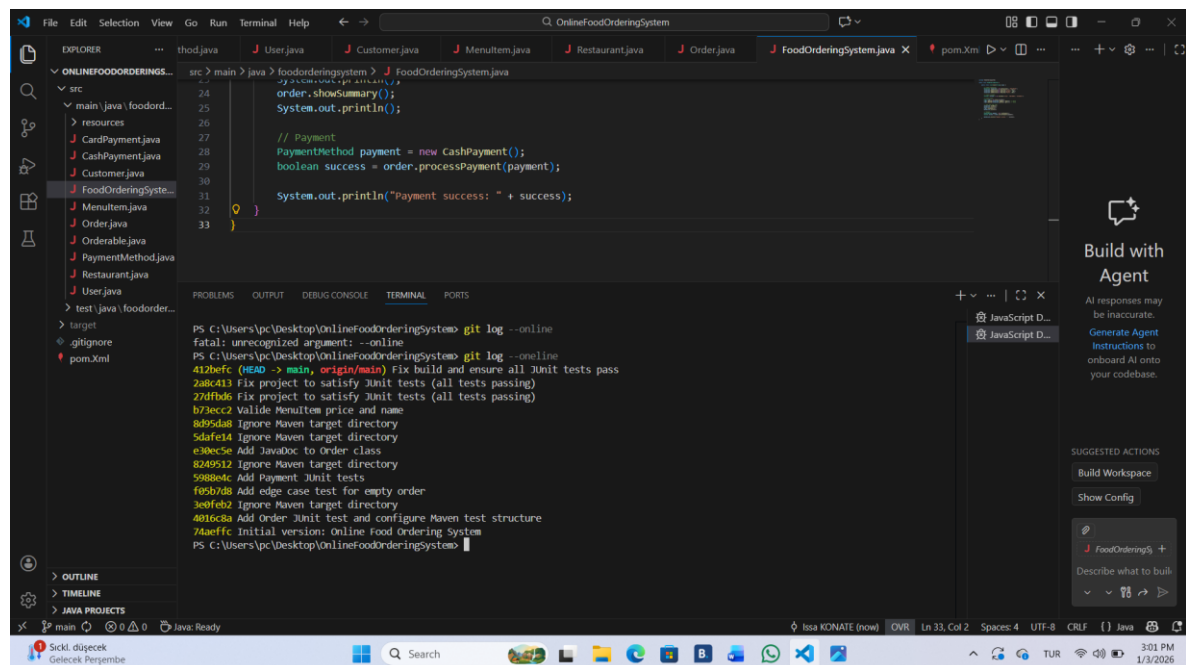
```
PS C:\Users\pc\Desktop\OnlineFoodOrderingSystem> mvn clean test
[INFO] Scanning for projects...
[INFO]
[INFO] ----- com.issaonline:online-food-ordering-system -----
[INFO] Building online-food-ordering-system 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO]
[INFO] ----- [ jar ] -----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ online-food-ordering-system ---
[INFO] Deleting C:\Users\pc\Desktop\OnlineFoodOrderingSystem\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ online-food-ordering-system ---
[INFO] skip non existing resourceDirectory C:\Users\pc\Desktop\OnlineFoodOrderingSystem\src\main\resources
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ online-food-ordering-system ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 10 source files with javac [debug target 17] to target\classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ online-food-ordering-system ---
[INFO] skip non existing resourceDirectory C:\Users\pc\Desktop\OnlineFoodOrderingSystem\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ online-food-ordering-system ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 4 source files with javac [debug target 17] to target\test-classes
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ online-food-ordering-system ---
[INFO] Using auto detected provider org.apache.maven.surefire.junitplatform.JUnitPlatformProvider
[INFO]
[INFO] ----- T E S T S -----
[INFO]
[INFO] Running foodorderingsystem.MenuItemTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.106 s -- in foodorderingsystem.MenuItemTest
[INFO] Running foodorderingsystem.OrderTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 s -- in foodorderingsystem.OrderTest
[INFO] Running foodorderingsystem.PaymentTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.008 s -- in foodorderingsystem.PaymentTest
[INFO] Running foodorderingsystem.RestaurantTest
[INFO] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 s -- in foodorderingsystem.RestaurantTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 9, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] BUILD SUCCESS
[INFO] Total time: 3.100 s
[INFO] Finished at: 2026-01-02T12:34:56+03:00
[INFO]
PS C:\Users\pc\Desktop\OnlineFoodOrderingSystem>
```

Figure X: Successful execution of JUnit tests using Maven (mvn clean test)

# GIT & GITHUB

<b>Github Repository Link:</b>	<a href="https://github.com/tenakosai90/OnlineFoodOrderingSystem">https://github.com/tenakosai90/OnlineFoodOrderingSystem</a>
<b>Github Projects Link:</b>	<a href="https://github.com/users/tenakosai90/projects/2/views/1">https://github.com/users/tenakosai90/projects/2/views/1</a>

Add a screenshot of your **git log –online** here



The screenshot shows an IDE with a project named 'OnlineFoodOrderingSystem'. The Explorer panel on the left shows the project structure, including 'src/main/java/foodord...' and 'resources'. The main editor displays 'FoodOrderingSystem.java' with the following code:

```
23: public void showSummary() {
24:     order.showSummary();
25:     System.out.println();
26: }
27: // Payment
28: PaymentMethod payment = new CashPayment();
29: boolean success = order.processPayment(payment);
30: System.out.println("Payment success: " + success);
31: }
32: }
33: }
```

The terminal window at the bottom shows the output of the command 'git log --online'. The output is as follows:

```
PS C:\Users\pc\Desktop\OnlineFoodOrderingSystem> git log --online
fatal: unrecognized argument: --online
PS C:\Users\pc\Desktop\OnlineFoodOrderingSystem> git log --online
412b6fc (HEAD -> main, origin/main) Fix build and ensure all JUnit tests pass
2a8c411 Fix project to satisfy JUnit tests (all tests passing)
27dfbdc Fix project to satisfy JUnit tests (all tests passing)
b73ecc2 Valide MenuItem price and name
8d95da8 Ignore Maven target directory
5d4fe14 Ignore Maven target directory
e30ec5e Add JavaDoc to Order class
8249512 Ignore Maven target directory
5988b4c Add Payment JUnit tests
f05b7d8 Add edge case test for empty order
3e0feb2 Ignore Maven target directory
4016c8a Add Order JUnit test and configure Maven test structure
74aeffc Initial version: Online Food Ordering System
PS C:\Users\pc\Desktop\OnlineFoodOrderingSystem>
```

The IDE also shows a 'Build with Agent' sidebar on the right, which includes a 'Generate Agent Instructions' button and a 'SUGGESTED ACTIONS' section with buttons for 'Build Workspace' and 'Show Config'.