CSE241: Object-Oriented Programming

Youssef Wael Youssef
23P0283

Ali Ibrahim Ahmed 3lwy
23P0217

Youssef Lotfy El-sayed
23P0216

Eyad Mohamed El-tanany
23P0207

Desktop E-commerce Application - Project Report

Faculty of Engineering, Ain Shams University

CSE241: Object-Oriented Programming

1st Semester 2024/2025

Table of Contents

1. Project Overview

This project implements a Desktop E-commerce Application using Java. The application demonstrates Object-Oriented Programming (OOP) principles in the backend and features a user-friendly graphical user interface (GUI). It provides functionalities for customer management, product browsing, cart operations, and administrative control.

2. Problem Description

The goal was to design and develop a fully functional e-commerce system with two primary roles:
1. Customer: Browses products, adds items to the cart, and places orders.
2. Admin: Manages products, views customers and orders, and performs CRUD operations.

The system also incorporates validation, multi-threading for real-time updates, and basic networking for admin-user communication.

3. Analysis and Design

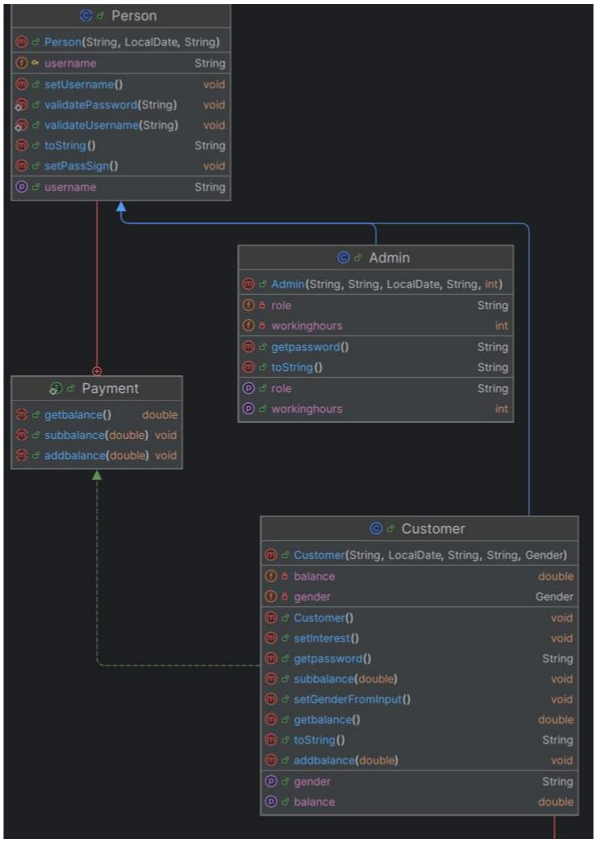
UML Diagrams

UML diagrams illustrate the relationships between classes such as Admin, Customer, Product, Cart, and Order.

Class Relationships and Hierarchies

The project employs abstraction through interfaces and inheritance. Key classes include:

- Admin and Customer: Derived from a base Person class.
- Product and Category: Classes represent items in the e-commerce system.
- Cart and Order: Classes handle user transactions and order tracking.
- Database: Centralized static storage for all data entities.

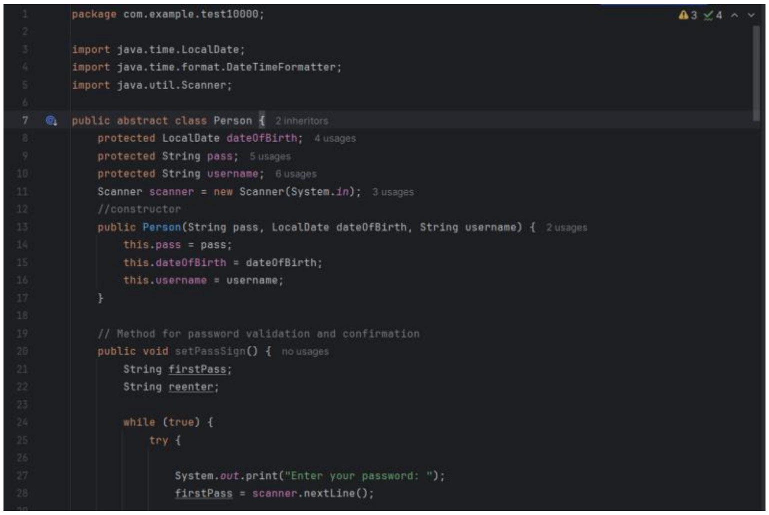


4. Implementation

Milestone 1: OOP Foundations

Key Features:

1. Admin and Customer Profiles
   - Attributes: username, password, date of birth, etc.
2. Product and Category Management
   - Admin CRUD operations for products and categories.
3. Cart and Order Management
   - Customers can add items to their cart and place orders.
4. Validation
   - Comprehensive data validation across the application.
5. Database
   - ArrayList for storing entities like users and products.



```java
package com.example.test10000;

import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.Scanner;

public abstract class Person {
    protected LocalDate dateOfBirth;
    protected String pass;
    protected String username;
    Scanner scanner = new Scanner(System.in);
    //constructor
    public Person(String pass, LocalDate dateOfBirth, String username) {
        this.pass = pass;
        this.dateOfBirth = dateOfBirth;
        this.username = username;
    }

    // Method for password validation and confirmation
    public void setPassSign() {
        String firstPass;
        String reenter;

        while (true) {
            try {

                System.out.print("Enter your password: ");
                firstPass = scanner.nextLine();
```

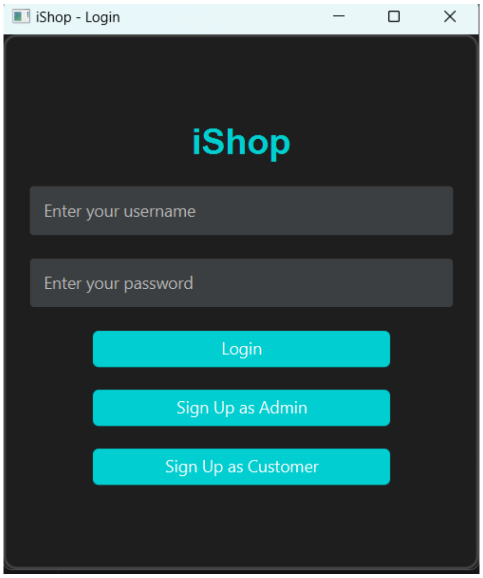
Milestone 2: GUI, Multi-threading, and Networking

1. GUI: Developed using JavaFX, featuring screens for:

   - User registration and login.

   - Product browsing and cart management.

   - Admin dashboard for managing users, products, and orders.

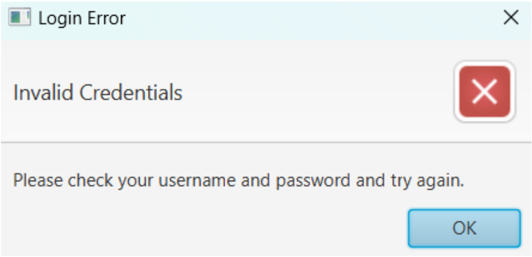


5. Testing

Valid and invalid test cases were implemented for:

- User registration and login validation.
- Cart operations (e.g., adding unavailable items).
- Admin CRUD operations.

Test cases demonstrated the robustness of the system under edge cases.



Invalid Credentials

Please check your username and password and try again.

6. Limitations

1. No integration with an external database system; data is stored in static memory.
2. Limited scalability for handling a large number of concurrent users.
3. Basic networking features implemented; advanced functionality like real-time notifications is absent.

7. Future Enhancements

1. Database Integration: Migrate static data storage to an SQL-based database.
2. Enhanced GUI: Add features like product search, sorting, and filtering.
3. Advanced Networking: Incorporate real-time notifications and live customer support.
4. Scalability: Optimize the application for handling more users and data.