Object Oriented Programming

Assignment

Bachelor of Computer Application

Submitted by

Name: Aryan Bhandari **Sap Id. 500123247**

Submitted to Dr. Muthu Kumar

Project Report: Customer Relationship Management (CRM) System

---

1. Introduction

1.1 Project Overview

The Customer Relationship Management (CRM) System is designed to facilitate seamless interactions between administrators, users, and employees in an e-commerce environment. The system allows users to browse products, place and manage orders, while administrators oversee inventory and order processing. Employees are responsible for handling pending orders and customer management.

1.2 Objectives

- Develop a structured CRM system with role-based access.
- Allow users to browse products and place orders efficiently.
- Enable administrators to manage inventory and orders.
- Provide employees with tools to track and process orders.

---

2. System Architecture

2.1 Technology Stack

- **Programming Language:** Java
- **Development Environment:** VS Code
- **Data Structures:** ArrayLists for storing products, orders, and users
- **Input Method:** Scanner (for user input via command line interface)

2.2 System Components

1. **Person (Base Class):** Stores common attributes such as name, email, and phone.
2. **Product Class:** Stores product details such as ID, name, price, and stock quantity.
3. **Order Class:** Manages orders placed by users and tracks order status.
4. **Authentication Interface:** Defines login and logout methods for different roles.
5. **Admin Class:** Manages product inventory and order tracking.
6. **User Class:** Allows users to browse products, place orders, cancel orders, and update profiles.
7. **Employee Class:** Processes pending orders and manages customer details.
8. **CRM Class:** Implements the main logic and menu-driven user interaction.

---

3. Features and Functionalities

3.1 Admin Features

- **Login/Logout:** Secure login for administrators.
- **View Products:** Display available products in the inventory.
- **Add Product:** Add new products to the inventory.
- **Remove Product:** Delete products from the inventory.
- **View Orders:** Display all orders placed by users.

3.2 User Features

- **Login/Logout:** Secure login for users.
- **View Products:** Browse available products.
- **Place Order:** Select products and place an order.
- **View Order History:** Review past and current orders.
- **Cancel Order:** Cancel pending orders.
- **Update Profile:** Modify personal details such as name, email, and phone.

3.3 Employee Features

- **Login/Logout:** Secure login for employees.
- **View Pending Orders:** Check pending orders for processing.
- **Process Order:** Mark pending orders as completed.
- **View Customers:** Display registered user details.

---

4. Implementation Details

4.1 Product Management

- Products are stored in an ArrayList<Product>.
- Admins can add and remove products dynamically.
- Users can view available products before placing an order.

4.2 Order Processing

- Users select products to place an order.
- Orders are assigned unique orderId values.
- Employees process pending orders and update order statuses.

4.3 Authentication System

- Each role (Admin, User, Employee) implements the Authentication interface.
- Separate login and logout mechanisms ensure controlled access.

---

5. Testing and Results

5.1 Test Cases

Test Case	Expected Result	Actual Result
Admin adds a product	Product successfully added	Passed
User places an order	Order is created and stored	Passed
Employee processes an order	Order status changes to 'Completed'	Passed
User cancels a pending order	Order status changes to 'Cancelled'	Passed

5.2 Performance Evaluation

- The system efficiently manages products and orders.
- Role-based authentication ensures controlled access.
- User interactions through command-line menus are smooth and responsive.

---

6. Conclusion

6.1 Key Achievements

- Developed a functional CRM system with three roles (Admin, User, Employee).
- Implemented product management, order placement, and processing.
- Provided a user-friendly menu-driven interface.

6.2 Future Enhancements

- Integrate a database for persistent storage.
- Develop a graphical user interface (GUI) for better user experience.
- Implement payment gateway integration for real transactions.

---

7. References

- Java Documentation: https://docs.oracle.com/en/java/

- Object-Oriented Programming Concepts
- Data Structures and Algorithms in Java