AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

SCHOOL OF COMPUTING

MARADANA SRIJA

CH.SC.U4CSE24126

OBJECT ORIENTED PROGRAMMING (23CSE111)

LAB RECORD

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING, CHENNAI

BONAFIDE CERTIFICATE

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24126 – MARADANA SRIJA*** in **"Computer Science and Engineering"** is a Bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on

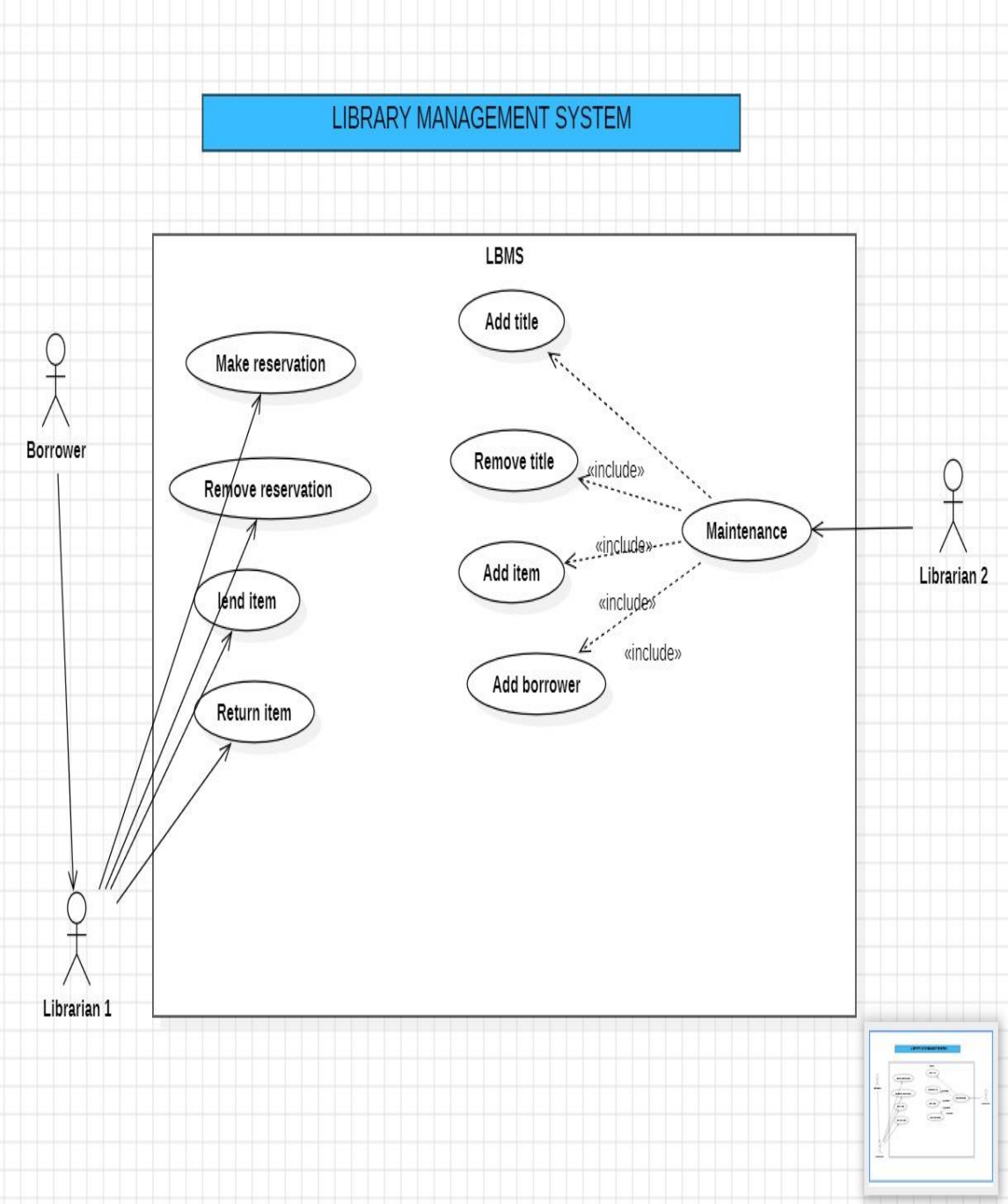
Internal Examiner 1 Internal Examiner 2

INDEX

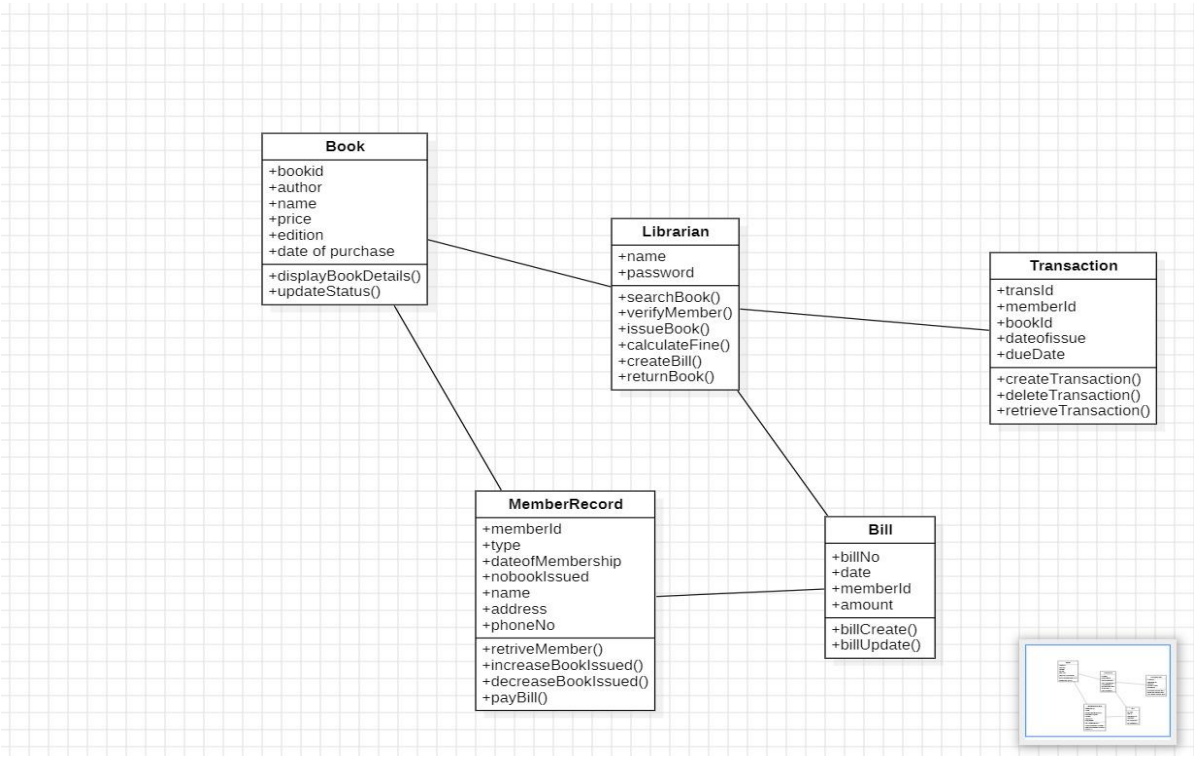
UML DIAGRAMS

1. LIBRARY MANAGEMENT SYSTEM

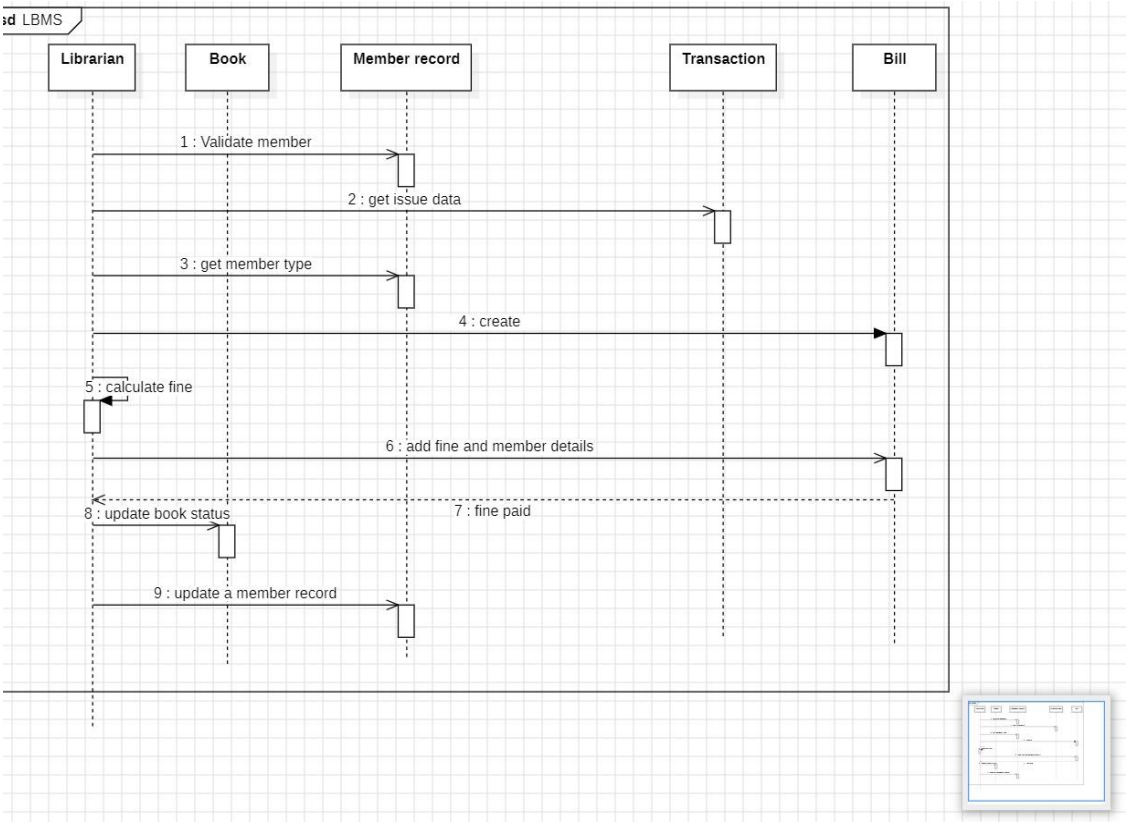
1.a) Use Case Diagram:



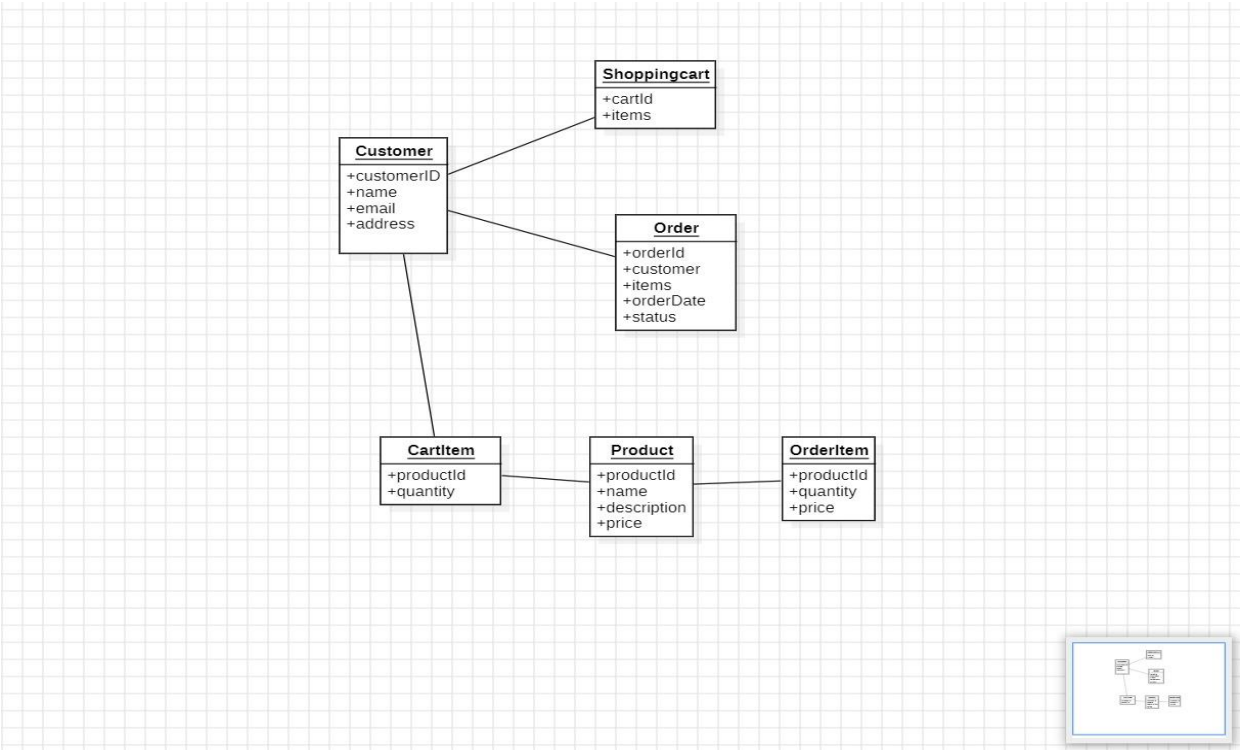
1.b) Class Diagram:



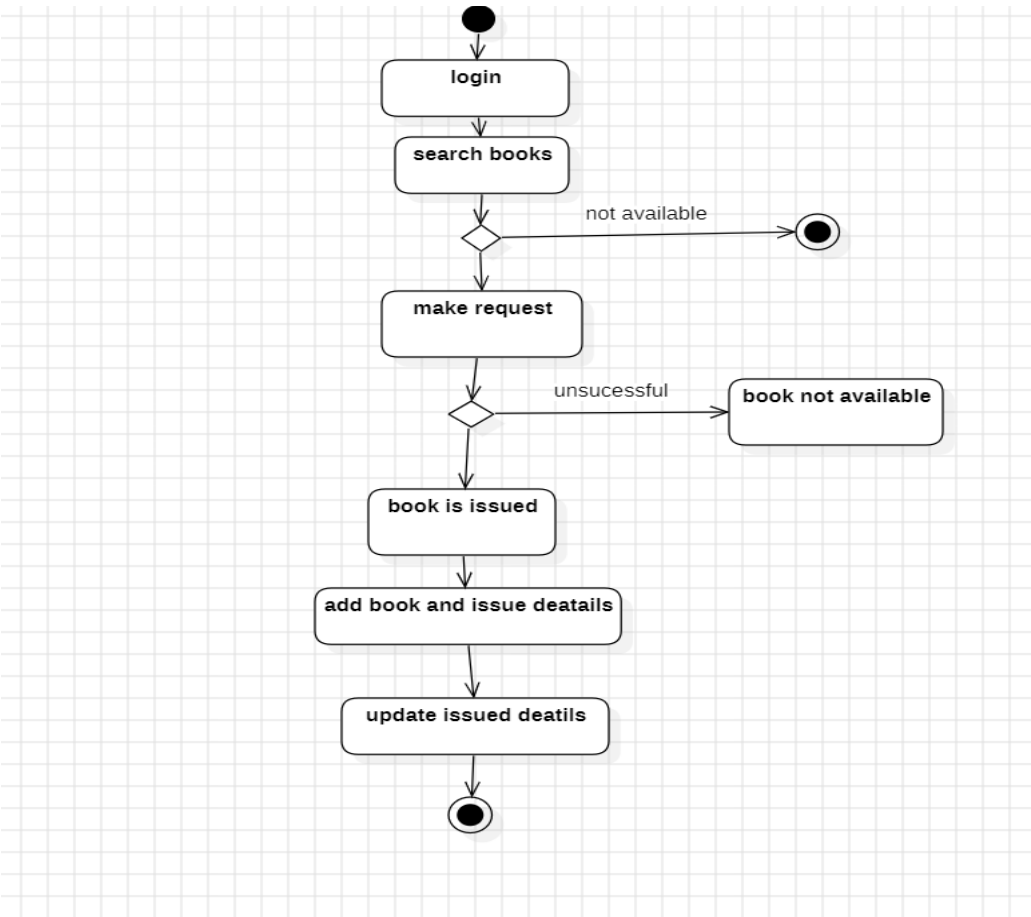
1.c) Sequence Diagram:



1.d) Object Diagram:

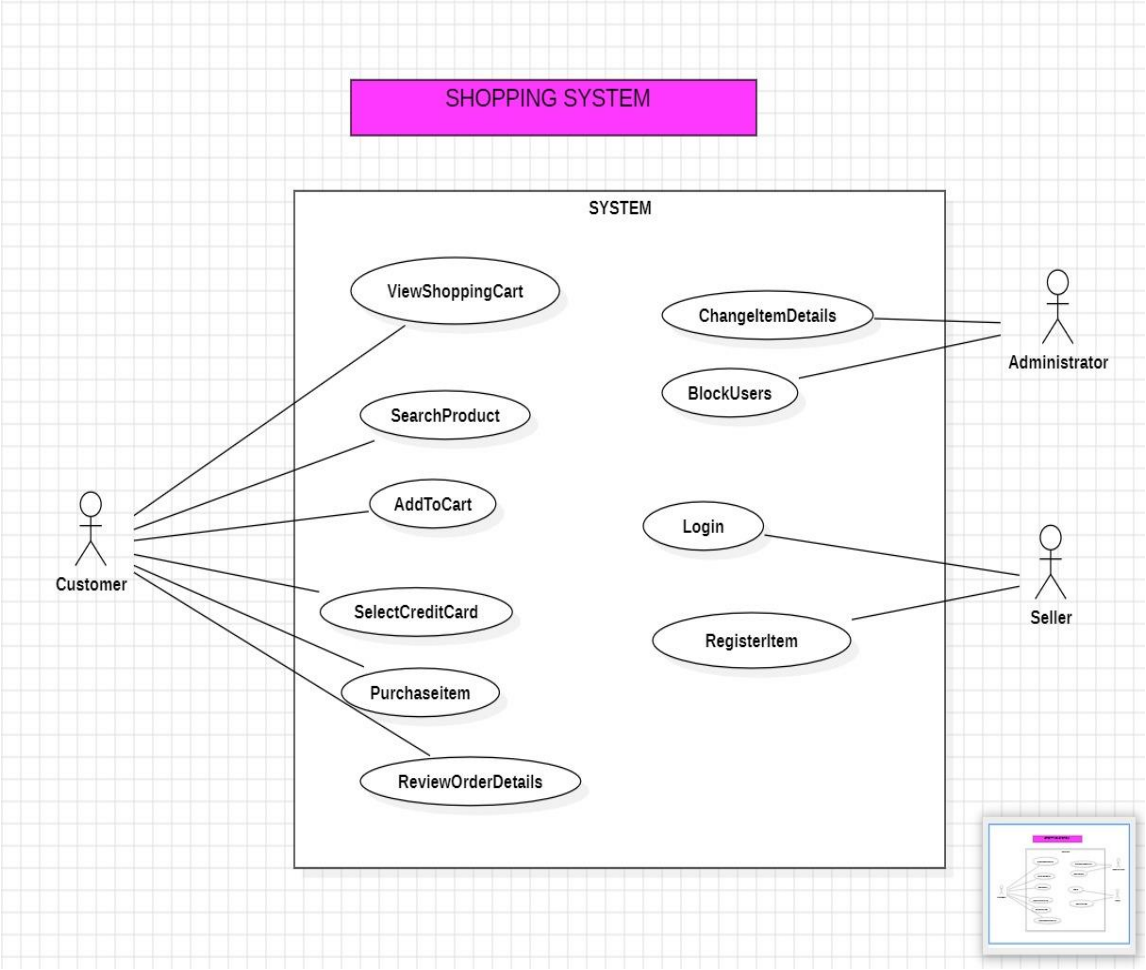


1.e) State-Activity Diagram:

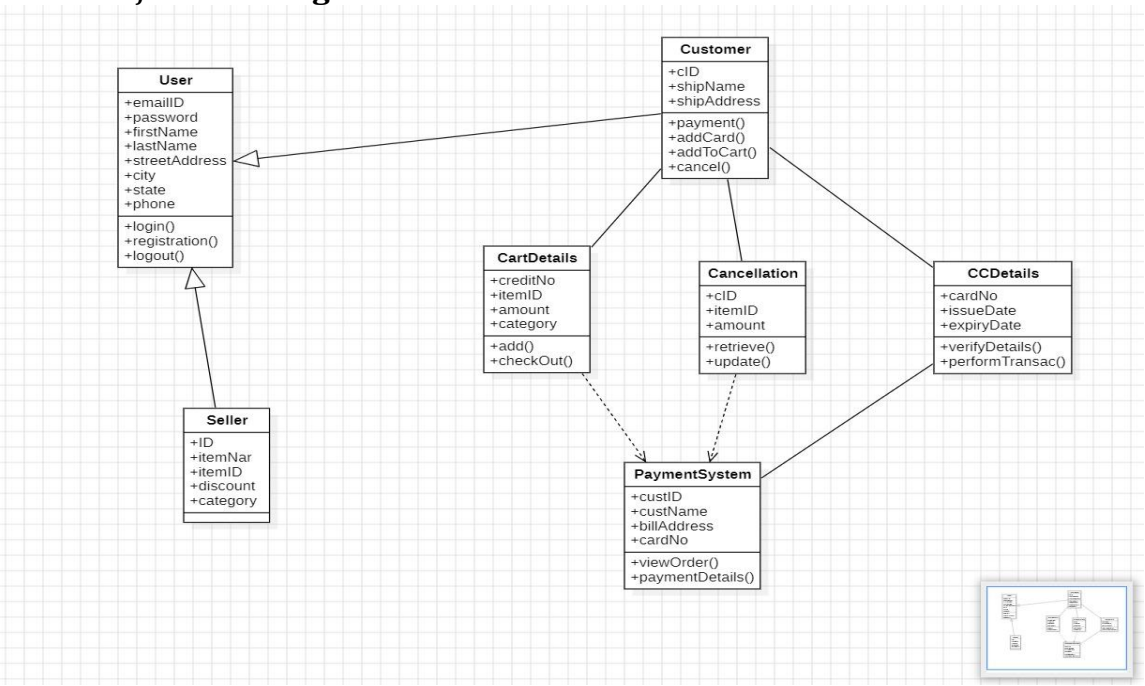


2. ONLINE SHOPPING SYSTEM

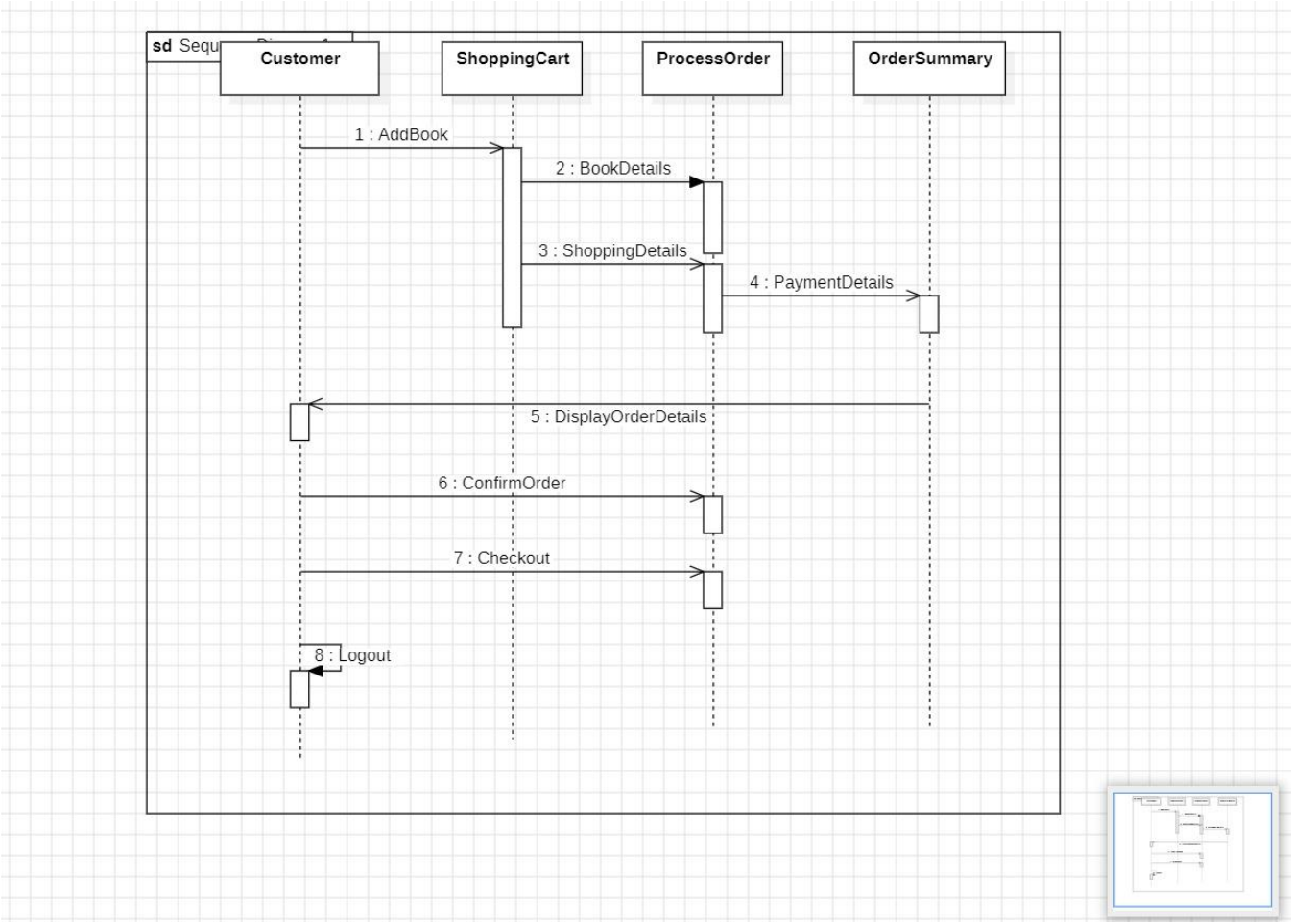
2.a) Use Case Diagram:



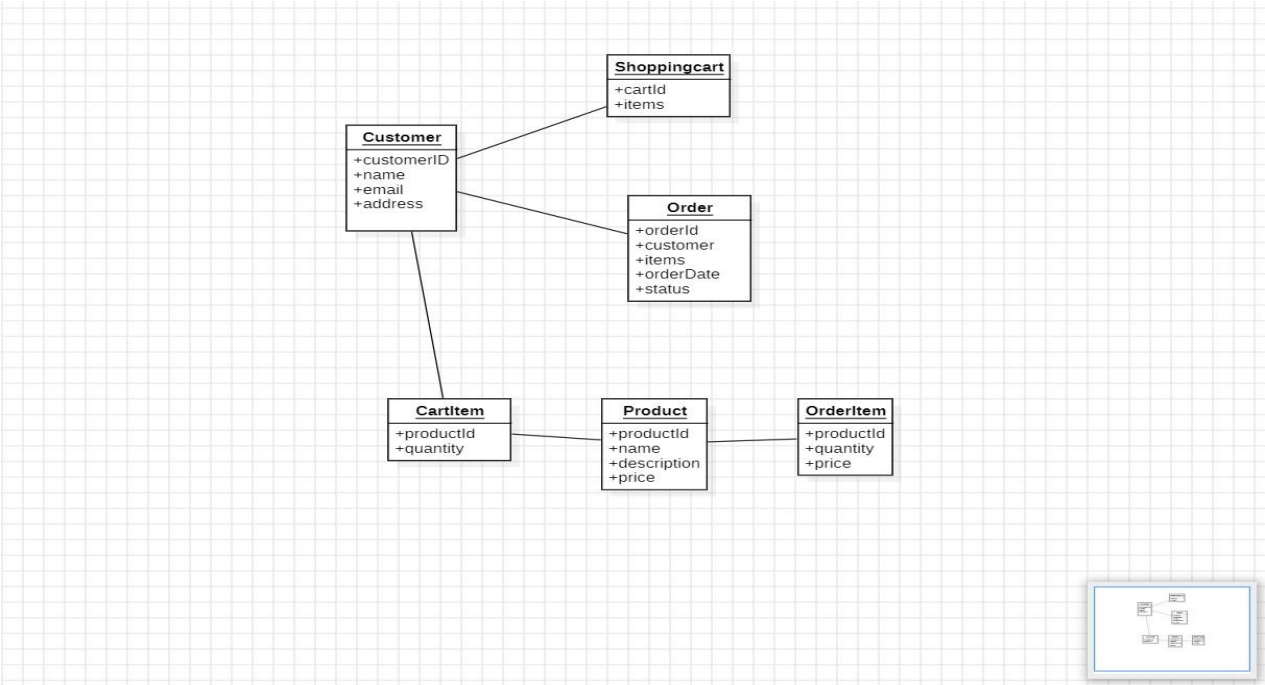
2.b) Class Diagram:



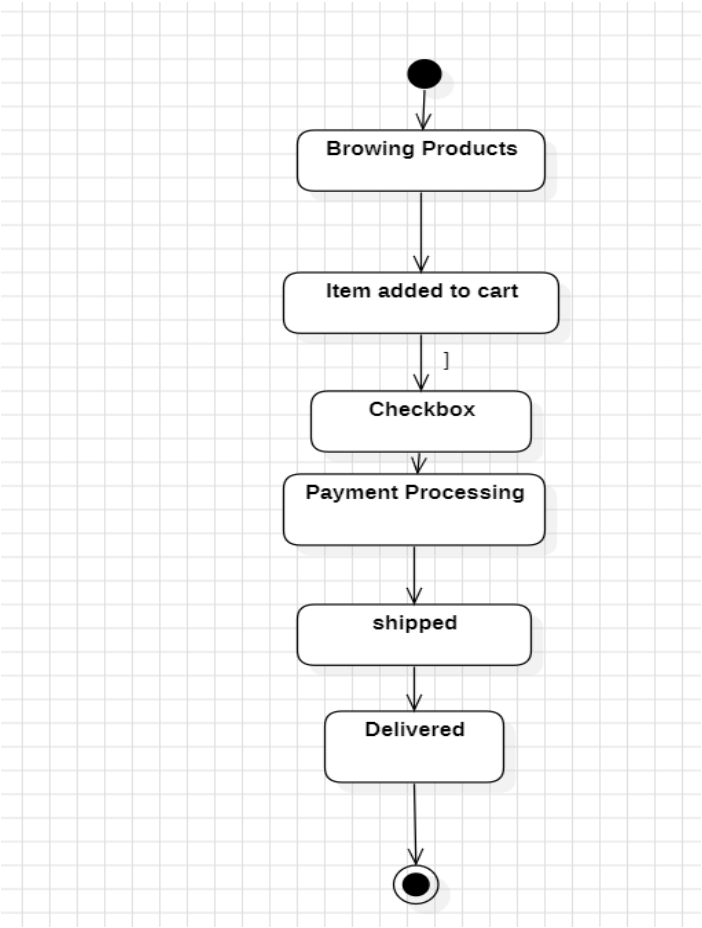
2.c) Sequence Diagram:



2.d) Object Diagram:



2.e) State-Activity Diagram:

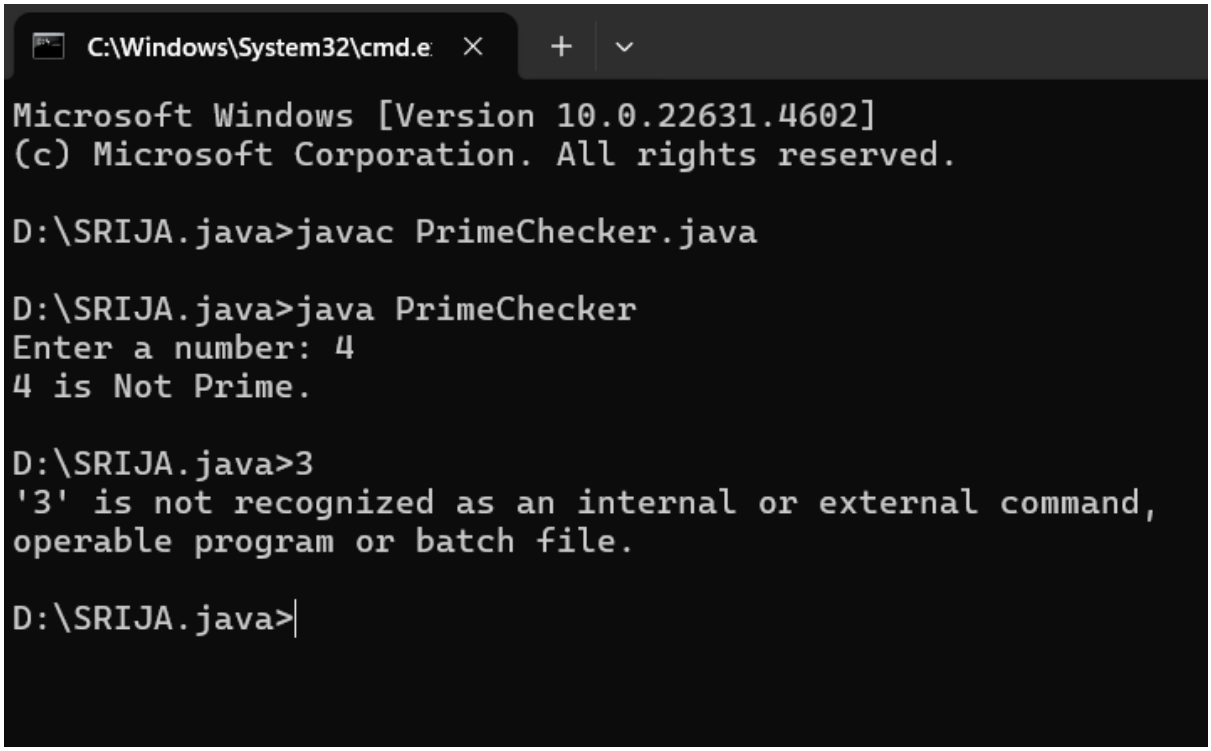


3. Basic Java Programs

1)Java Program to Check if a Number is Prime
CODE:

```
import java.util.Scanner;

public class PrimeChecker {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.print("Enter a number: ");
      int num = scanner.nextInt();
      boolean isPrime = true;

      if (num <= 1) {
         isPrime = false;
      } else {
         for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0) {
               isPrime = false;
               break;
            }
         }
      }

      if (isPrime)
         System.out.println(num + " is Prime.");
      else
         System.out.println(num + " is Not Prime.");

      scanner.close();
   }
}
```

OUTPUT:



2) Find the Greatest Common Divisor (GCD) Using a Class

CODE:

```
import java.util.Scanner;

class GCDCalculator {
   int a, b;

   void inputNumbers() {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter first number: ");
      a = scanner.nextInt();
      System.out.print("Enter second number: ");
      b = scanner.nextInt();
   }

   void findGCD() {
      int num1 = a, num2 = b;
      while (num2 != 0) {
         int temp = num2;
         num2 = num1 % num2;
         num1 = temp;
      }
      System.out.println("GCD: " + num1);
   }
}

public class GCDApp {
   public static void main(String[] args) {
      GCDCalculator gcd = new GCDCalculator();
      gcd.inputNumbers();
      gcd.findGCD();
```

```
    }
}
```

OUTPUT:

```
D:\SRIJA.java>javac GCDApp.java

D:\SRIJA.java>java GCDApp
Enter first number: 10
Enter second number: 12
GCD: 2

D:\SRIJA.java>
```

3) Convert Celsius to Fahrenheit

CODE:

```
import java.util.Scanner;

class TemperatureConverter {
    double celsius;

    void inputTemperature() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter temperature in Celsius: ");
        celsius = scanner.nextDouble();
    }

    void convert() {
        double fahrenheit = (celsius * 9/5) + 32;
        System.out.println("Temperature in Fahrenheit: " + fahrenheit);
    }
}

public class TemperatureApp {
    public static void main(String[] args) {
        TemperatureConverter temp = new TemperatureConverter();
        temp.inputTemperature();
        temp.convert();
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac TemperatureApp.java

D:\SRIJA.java>java TemperatureApp
Enter temperature in Celsius: 40
Temperature in Fahrenheit: 104.0

D:\SRIJA.java>
```

4) Student Grading System for Multiple Students (While Loop)

Problem: Allow multiple students to enter their marks using a **while loop**, and print their grades.

CODE:

```
import java.util.Scanner;

class MultipleStudentsGrading {
   void assignGrades() {
      Scanner scanner = new Scanner(System.in);
      char choice;

      while (true) {
         System.out.print("Enter student marks: ");
         int marks = scanner.nextInt();

         if (marks >= 90)
            System.out.println("Grade: A");
         else if (marks >= 80)
            System.out.println("Grade: B");
         else if (marks >= 70)
            System.out.println("Grade: C");
         else if (marks >= 60)
            System.out.println("Grade: D");
         else
            System.out.println("Grade: F");

         System.out.print("Do you want to enter another student's marks? (y/n): ");
         choice = scanner.next().charAt(0);
         if (choice == 'n' || choice == 'N')
            break;
      }
   }
}

public class MultiStudentGradeApp {
   public static void main(String[] args) {
      MultipleStudentsGrading msg = new MultipleStudentsGrading();
      msg.assignGrades();
   }
```

}

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac MultiStudentGradeApp.java

D:\SRIJA.java>java MultiStudentGradeApp
Enter student marks: 20
Grade: F
Do you want to enter another student?s marks? (y/n): yes
Enter student marks: 56
Grade: F
Do you want to enter another student?s marks? (y/n): no

D:\SRIJA.java>
```

5) Grading System for Multiple Subjects (Do-While Loop)

Problem: Calculate the **average marks and grade** for a student across multiple subjects using a **do-while loop**.

CODE:

```
import java.util.Scanner;

class SubjectGrading {
   void calculateAverageGrade() {
      Scanner scanner = new Scanner(System.in);
      int totalMarks = 0, subjectCount = 0, marks;
      char choice;

      do {
         System.out.print("Enter subject marks: ");
         marks = scanner.nextInt();
         totalMarks += marks;
         subjectCount++;

         System.out.print("Do you have more subjects? (y/n): ");
         choice = scanner.next().charAt(0);
      } while (choice == 'y' || choice == 'Y');

      double average = (double) totalMarks / subjectCount;
      System.out.println("Average Marks: " + average);

      if (average >= 90)
         System.out.println("Final Grade: A");
      else if (average >= 80)
         System.out.println("Final Grade: B");
      else if (average >= 70)
         System.out.println("Final Grade: C");
      else if (average >= 60)
         System.out.println("Final Grade: D");
      else
         System.out.println("Final Grade: F");
```

```
    }
}

public class SubjectGradingApp {
    public static void main(String[] args) {
        SubjectGrading sg = new SubjectGrading();
        sg.calculateAverageGrade();
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac  SubjectGradingApp.java

D:\SRIJA.java>java  SubjectGradingApp
Enter subject marks: 67
Do you have more subjects? (y/n): 50
Average Marks: 67.0
Final Grade: D

D:\SRIJA.java>
```

6) **Employee Payroll System**

Problem: Create classes for **Employee, Salary, and Payroll**. Implement methods to calculate salary based on working hours, generate pay slips, and apply bonuses.

CODE:

```
class Employee {
    String name;
    int empId;
    double hourlyRate;

    Employee(String name, int empId, double hourlyRate) {
        this.name = name;
        this.empId = empId;
        this.hourlyRate = hourlyRate;
    }

    double calculateSalary(int hoursWorked) {
        return hoursWorked * hourlyRate;
    }

    void displayInfo() {
        System.out.println("Employee: " + name + ", ID: " + empId);
    }
}

class Payroll {
    static void generatePaySlip(Employee employee, int hoursWorked) {
```

```
        double salary = employee.calculateSalary(hoursWorked);
        System.out.println("Pay Slip for " + employee.name);
        System.out.println("Hours Worked: " + hoursWorked);
        System.out.println("Total Salary: $" + salary);
    }
}

public class PayrollApp {
    public static void main(String[] args) {
        Employee emp1 = new Employee("Alice Johnson", 1001, 20);
        emp1.displayInfo();
        Payroll.generatePaySlip(emp1, 40);
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac PayrollApp.java

D:\SRIJA.java>java PayrollApp
Employee: Alice Johnson, ID: 1001
Pay Slip for Alice Johnson
Hours Worked: 40
Total Salary: $800.0

D:\SRIJA.java>
```

7) Airline Reservation System

Problem: Create classes for **Flight, Passenger, Ticket, and Booking**. Implement methods to book tickets, cancel reservations, and display flight details.

CODE:

```
import java.util.ArrayList;

class Flight {
    String flightNumber;
    String destination;

    Flight(String flightNumber, String destination) {
        this.flightNumber = flightNumber;
        this.destination = destination;
    }

    void displayFlightDetails() {
        System.out.println("Flight: " + flightNumber + " to " + destination);
    }
}
```

```
class Passenger {
   String name;
   int age;

   Passenger(String name, int age) {
      this.name = name;
      this.age = age;
   }

   void displayPassengerInfo() {
      System.out.println("Passenger: " + name + ", Age: " + age);
   }
}

class Ticket {
   Flight flight;
   Passenger passenger;

   Ticket(Flight flight, Passenger passenger) {
      this.flight = flight;
      this.passenger = passenger;
   }

   void displayTicketDetails() {
      flight.displayFlightDetails();
      passenger.displayPassengerInfo();
   }
}

public class AirlineReservationApp {
   public static void main(String[] args) {
      Flight flight = new Flight("AI123", "New York");
      Passenger passenger = new Passenger("John Smith", 30);

      Ticket ticket = new Ticket(flight, passenger);
      ticket.displayTicketDetails();
   }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac AirlineReservationApp.java

D:\SRIJA.java>java AirlineReservationApp
Flight: AI123 to New York
Passenger: John Smith, Age: 30

D:\SRIJA.java>
```

8) **University Management System**

Problem: Create classes for **Department, Faculty, Student, and Exam**. Implement methods to assign faculty to courses, conduct exams, and manage student records.

CODE:

```
import java.util.ArrayList;

class Department {
   String name;
   ArrayList<String> courses = new ArrayList<>();

   Department(String name) {
      this.name = name;
   }

   void addCourse(String course) {
      courses.add(course);
   }

   void displayDetails() {
      System.out.println("Department: " + name);
      System.out.println("Courses Offered: " + courses);
   }
}

class Faculty {
   String name;
   String department;

   Faculty(String name, String department) {
      this.name = name;
      this.department = department;
   }

   void displayInfo() {
      System.out.println("Faculty: " + name + ", Department: " + department);
   }
```

```
}

class Student {
   String name;
   int id;

   Student(String name, int id) {
      this.name = name;
      this.id = id;
   }

   void displayInfo() {
      System.out.println("Student: " + name + ", ID: " + id);
   }
}

public class UniversityApp {
   public static void main(String[] args) {
      Department csDept = new Department("Computer Science");
      csDept.addCourse("Java");
      csDept.addCourse("Data Structures");

      Faculty prof = new Faculty("Dr. Brown", "Computer Science");
      Student student = new Student("Alice", 101);

      csDept.displayDetails();
      prof.displayInfo();
      student.displayInfo();
   }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac UniversityApp.java

D:\SRIJA.java>java UniversityApp
Department: Computer Science
Courses Offered: [Java, Data Structures]
Faculty: Dr. Brown, Department: Computer Science
Student: Alice, ID: 101

D:\SRIJA.java>
```

9) **Hotel Booking System**

Problem: Create classes for **Hotel, Room, Guest, and Reservation**. Implement methods to check room availability, book rooms, and manage guest details.

CODE:

```
import java.util.ArrayList;

class Room {
   int roomNumber;
   boolean isBooked;

   Room(int roomNumber) {
      this.roomNumber = roomNumber;
      this.isBooked = false;
   }

   void bookRoom() {
      if (!isBooked) {
         isBooked = true;
         System.out.println("Room " + roomNumber + " booked successfully.");
      } else {
         System.out.println("Room " + roomNumber + " is already booked.");
      }
   }

   void displayRoomInfo() {
      System.out.println("Room " + roomNumber + " - " + (isBooked ? "Booked" : "Available"));
   }
}

class Guest {
   String name;
   int guestId;

   Guest(String name, int guestId) {
      this.name = name;
      this.guestId = guestId;
   }

   void displayGuestInfo() {
      System.out.println("Guest: " + name + ", ID: " + guestId);
   }
}

public class HotelBookingApp {
   public static void main(String[] args) {
      Room room1 = new Room(101);
      Room room2 = new Room(102);

      Guest guest1 = new Guest("Alice Johnson", 1);

      guest1.displayGuestInfo();
      room1.displayRoomInfo();
      room1.bookRoom();
      room1.displayRoomInfo();
   }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac HotelBookingApp.java

D:\SRIJA.java>java HotelBookingApp
Guest: Alice Johnson, ID: 1
Room 101 - Available
Room 101 booked successfully.
Room 101 - Booked

D:\SRIJA.java>
```

10) **Online Shopping System**

Problem: Define classes for **User, Product, Order, and Payment**. Implement methods to browse products, place orders, process payments, and track shipments.

CODE:

```
import java.util.ArrayList;

class Product {
   String name;
   double price;

   Product(String name, double price) {
      this.name = name;
      this.price = price;
   }

   void displayProduct() {
      System.out.println("Product: " + name + " - Price: $" + price);
   }
}

class Order {
   ArrayList<Product> products = new ArrayList<>();

   void addProduct(Product product) {
      products.add(product);
   }

   void displayOrderDetails() {
      System.out.println("Order Details:");
      double total = 0;
      for (Product p : products) {
         p.displayProduct();
         total += p.price;
      }
      System.out.println("Total: $" + total);
```

```
    }
}

public class ShoppingApp {
    public static void main(String[] args) {
        Product p1 = new Product("Smartphone", 700);
        Product p2 = new Product("Headphones", 50);

        Order order = new Order();
        order.addProduct(p1);
        order.addProduct(p2);
        order.displayOrderDetails();
    }
}
```

OUTPUT:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

D:\SRIJA.java>javac ShoppingApp.java

D:\SRIJA.java>java ShoppingApp
Order Details:
Product: Smartphone - Price: $700.0
Product: Headphones - Price: $50.0
Total: $750.0

D:\SRIJA.java>
```