LinkedList.java

```
public class LinkedList {
    Node first;
    Node current;
    Node last;

    public LinkedList() {
        first = current = last = null;
    }

    public boolean isEmpty() {
        return (first == null);
    }

    public void insertAtFront(Object insertItem) {
        Node newNode = new Node(insertItem);

        if (isEmpty()) {
            first = newNode;
            last = newNode;
        } else {
            newNode.next = first;
            first = newNode;
        }
    }

    public void insertAtBack(Object insertItem) {
        Node newNode = new Node(insertItem);

        if (isEmpty()) {
            first = newNode;
            last = newNode;
        } else {
            last.next = newNode;
            last = newNode;
        }
    }

    public Object removeFromFront() {
        Object removeItem = null;
        if (isEmpty()) {
            return removeItem;
        }
        removeItem = first.data;

        if (first == last) {
            first = null;
            last = null;
        } else {
            first = first.next;
```

```
    }
    return removeItem;
}

public Object removeFromBack() {
    Object removeItem = null;

    if (isEmpty()) {
        return removeItem;
    }
    removeItem = last.data;

    if (first == last) {
        first = null;
        last = null;
    } else {
        current = first;
        while (current.next != last) {
            current = current.next;
        }
        last = current;
        last.next = null;
    }

    return removeItem;
}

public Object getFirst() {
    if (isEmpty()) {
        return null;
    } else {
        current = first;
        return current.data;
    }
}

// Get the next node pointed by current node
public Object getNext() {
    if (current == last) {
        return null;
    } else {
        current = current.next;
        return current.data;
    }
}

// Get the size of the list
public int getSize() {
    Node current = first;
    int size = 0;
    while (current != null) {
```

```
            size++;
            current = current.next;
        }

        return size;
    }

    public void removeAtIndex(int index) {
        if (index < 0) {
            System.out.println("Invalid index");
            return;
        }

        if (index == 0) {
            first = first.next;
            return;
        }

        Node current = first;
        Node prev = null;
        int count = 0;

        while (current != null && count < index) {
            prev = current;
            current = current.next;
            count++;
        }

        if (current == null) {
            System.out.println("Index out of range");
            return;
        }

        prev.next = current.next;
    }

    public int indexOf(PurchasementDetail pd) {
        int index = 0;
        Node current = first;

        while (current != null) {
            if (current.data.equals(pd)) {
                return index;
            }
            current = current.next;
            index++;
        }

        return -1; // Element not found
    }
}
```

Node.java

```
public class Node {
    Object data;
    Node next;

    public Node() {
        data = null;
        next = null;
    }

    public Node(Object data) {
        this.data = data;
        next = null;
    }
}
```

Queue.java

```
import java.util.LinkedList;

public class Queue<E> {
    LinkedList<E> list;

    public Queue() {
        list = new LinkedList<E>();
    }

    public int size() {
        return list.size();
    }

    public boolean isEmpty() {
        return list.isEmpty();
    }

    public void enqueue(E item) {
        list.addLast(item);
    }

    public E dequeue() {
        return list.removeFirst();
    }

    public E front() {
        return list.getFirst();
    }
}
```

Client.java

```
public class Client {

    private String name;
    private String noIC;
    private String noPhone;
    private String Email;
    private CarPackage cp;

    public Client() {
        name = "";
        noIC = "";
        noPhone = "";
        Email = "";
    }

    public void setDetails(String nm, String IC, String np, String email, double bp, int aChoice,
char bk) {
        name = nm;
        noIC = IC;
        noPhone = np;
        Email = email;
        cp = new CarPackage (bp, aChoice, bk);
    }

    public void setName(String nm) {
        name = nm;
    }
    public void setNoIC(String IC) {
        noIC = IC;
    }
    public void setNoPhone(String np) {
        noPhone = np;
    }
    public void setCustEmail(String Cemail) {
        Email = Cemail;
    }

    public String getName() {
        return name;
    }
    public String getNoIC() {
        return noIC;
    }
    public String getNoPhone() {
        return noPhone;
    }
    public String getCustEmail() {
```

```
        return Email;
    }

    public String toString() {
        return "Name: " + name + "\nIC: " + noIC + "\nPhone: " + noPhone + "\nEmail: " +
Email;
    }

    public double totalPayment(int size, char hasBodykit) {

        return (cp.calcPayment(size) + cp.calcBodykitFee(hasBodykit));
    }
}
```

CarPackage.java

```
public class CarPackage {
    protected double basePrice;
    protected int amountPreference;
    protected char hasBodykit;

    public CarPackage(double bp, int ap, char bk) {
        basePrice = bp;
        amountPreference = ap;
        hasBodykit = bk;
    }

    public double getBasePrice() {
        return basePrice;
    }
    public int getAmountPreference() {
        return amountPreference;
    }
    public char getHasBodykit() {
        return hasBodykit;
    }
    public double calcBodykitFee(char hasBodykit) {
        double bodykitFee = 0.0;
        if (hasBodykit == 'Y' || hasBodykit == 'y') {
            bodykitFee = 800.00;
        }
        return bodykitFee;
    }

    public double calcPayment(int size) {
        double totalPrice = basePrice;

        if (size >= 2) {
            totalPrice = totalPrice - 500 ;
        } else if (size < 2) {
```

```
            totalPrice = totalPrice;
        }
        return totalPrice;
    }
}
```

Car.java

```
public class Car {
    private String carName;
    private double price;
    private int yearMade;
    private double mileage;
    private String condition;

    public Car() {
        carName = "";
        price = 0.0;
        yearMade = 0;
        mileage = 0.0;
        condition = "";
    }

    public Car(String nm, double p, int y, double m, String c) {
        carName = nm;
        price = p;
        yearMade = y;
        mileage = m;
        condition = c;
    }

    public String getCarName() { return carName; }
    public double getPrice() { return price; }
    public int getYearMade() { return yearMade; }
    public double getMileage() { return mileage; }
    public String getCondition() { return condition; }

    public String toString() {
        return String.format("%-20s RM %-10.2f Year: %-4d Mileage: %-8.1f km Condition: %-
10s",
                carName, price, yearMade, mileage, condition);
    }
}
```

PurchasementDetail.java

```
public class PurchasementDetail {

    private String pickupName;
    private String carColor;
    private String rimColor;
    private String purchaseDate;
    private String desiredLocation;


    public PurchasementDetail() {
        pickupName = "";
        carColor = "";
        rimColor = "";
        purchaseDate = "";
        desiredLocation = "";
        pickupName = "";
    }

    public PurchasementDetail(String pickupN, String color, String rim, String date, String
location) {
        pickupName = pickupN;
        carColor = color;
        rimColor = rim;
        purchaseDate = date;
        desiredLocation = location;
    }

    public void setPickupName(String pn) { pickupName = pn; }
    public void setCarColor(String cc) { carColor = cc; }
    public void setRimColor(String rc) { rimColor = rc; }
    public void setPurchaseDate(String pd) { purchaseDate = pd; }
    public void setDesiredLocation(String dl) { desiredLocation = dl; }


    public void setDetails(String pn,String cc, String rc, String pd, String dl ) {
        pickupName = pn;
        carColor = cc;
        rimColor = rc;
        purchaseDate = pd;
        desiredLocation = dl;
    }

    public String getPickupName() { return pickupName; }
    public String getCarColor() { return carColor; }
    public String getRimColor() { return rimColor; }
    public String getPurchaseDate() { return purchaseDate; }
    public String getDesiredLocation() { return desiredLocation; }
```

```
    public String toString() {
        return "\nName of the person that pickup the car: " + pickupName +
                "\nCar Color: " + carColor +
                "\nRim Color: " + rimColor +
                "\nPurchase Date: " + purchaseDate +
                "\nDesired Location: " + desiredLocation;

    }
}
```

CarDealerLL.java

```
import java.io.*;
import java.util.*;

public class CarDealerLL {
    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            Client cl = new Client();
            LinkedList PurchasementLL = new LinkedList();
            LinkedList CarLL = new LinkedList();

            FileReader fr = new FileReader("C:\\Users\\user\\Documents\\CarDetail.txt");
            BufferedReader br = new BufferedReader(fr);
            FileWriter wr = new
FileWriter("C:\\Users\\user\\Documents\\PurchasementDetails.txt");
            PrintWriter pw = new PrintWriter(wr);
            String strLine;

            while ((strLine = br.readLine()) != null) {
                try {
                    String[] data = strLine.split(",");
                    if (data.length != 5) {
                        throw new IllegalArgumentException("Invalid data format: " + strLine);
                    }

                    String carName = data[0].trim();
                    double price = Double.parseDouble(data[1].trim());
                    int yearMade = Integer.parseInt(data[2].trim());
                    double mileage = Double.parseDouble(data[3].trim());
                    String condition = data[4].trim();

                    Car cr = new Car(carName, price, yearMade, mileage, condition);
                    CarLL.insertAtBack(cr);

                } catch (Exception e) {
                    System.err.println("Error parsing car details: " + strLine + ". Skipping this line.");
                }
```

```
        }

System.out.println("╔═══════════════════════════════════════════════════════════════
═══════╗");
        System.out.println("║                                              ║");
        System.out.println("║    Welcome to HUA Refurbished Car Dealership     ║");
        System.out.println("║                                              ║");

System.out.println("╚═══════════════════════════════════════════════════════════════
═══════╝");
        System.out.println("--------------------------------------");
        System.out.println("🌍  Your Car, Anywhere in Malaysia  🌍");
        System.out.println("--------------------------------------");
        System.out.println("✓ 100+ branches nationwide.");
        System.out.println("✓ Make deals anytime, anywhere with your preference and
versatility.");
        System.out.println("✓ Many options for customer's main plan and backup plan.");
        System.out.println("✓ Perfect for personal or gift purposes.");
        System.out.println("--------------------------------------");
        System.out.print("\nPlease Enter your Personal Detail  ");
        System.out.print("\n\nEnter Name: ");
        String name = sc.nextLine();
        System.out.print("Enter IC Number: ");
        String noIC = sc.nextLine();
        System.out.print("Enter Phone Number: ");
        String noPhone = sc.nextLine();
        System.out.print("Enter Email: ");
        String Email = sc.nextLine();
        System.out.print("\nCustomer that provides many purchasement plan will ease our
dealer work ");
        System.out.print("\n!! RM500 DISCOUNT will be given if there is more than 1  !! ");
        System.out.print("\nEnter Number Of Purchasement Plan: ");
        int amountChoice = sc.nextInt();

        PurchasementDetail[] PD = new PurchasementDetail[amountChoice];

        for (int i = 0; i < amountChoice; i++) {
           PD[i] = new PurchasementDetail();
           System.out.print("\nPurchasement Plan " + (i + 1));
           System.out.print("\nName of the person that pickup the car: ");
           String pickupName = sc.next();
           System.out.print("Enter Car Color: ");
           String carColor = sc.next();
           sc.nextLine();
           System.out.print("Enter Rim Color: ");
           String rimColor = sc.nextLine();
           System.out.print("Enter Purchase Date [dd/mm/yyyy]: ");
           String purchaseDate = sc.nextLine();
           System.out.print("Enter Desired Purchase Location: ");
           String desiredLocation = sc.nextLine();
```

```
            PD[i].setDetails(pickupName, carColor, rimColor, purchaseDate, desiredLocation);
            PurchasementLL.insertAtBack(PD[i]);
        }
        Car cr = (Car) CarLL.getFirst();


        System.out.println("Available cars:");
        while (cr != null) {
            System.out.println(cr.toString());
            cr = (Car) CarLL.getNext();
        }

        System.out.print("\nChoose your car: ");
        String chosenCar = sc.nextLine();
        System.out.print("Do You Want the Car Bodykit Package (Y/N): ");
        char hasBodykit = sc.nextLine().charAt(0);
        double basePrice = 0.0;

        cr = (Car) CarLL.getFirst();


        while (cr != null) {
            if (cr.getCarName().equalsIgnoreCase(chosenCar)) {
                basePrice = cr.getPrice();
                break;
            }
            cr = (Car) CarLL.getNext();
        }

        if (basePrice == 0.0) {
            System.out.println("Car not found. Please choose a valid car.");
        }

        cl.setDetails(name, noIC, noPhone, Email, basePrice, amountChoice, hasBodykit);

        System.out.print("\n\n0. End program"
                + "\n1. Calculate total payment"
                + "\n2. View personal details"
                + "\n3. Change personal information"
                + "\n4. View Purchasement Plan details"
                + "\n5. Change Purchasement Plan details"
                + "\n6. Search Purchasement Plan details by name of the person that pickup for
Confirmation "
                + "\nChoose your option: ");
        int option = sc.nextInt();

        double finalPayment = 0.0;

        while (option != 0) {
            if (option == 1) {
```

```
                finalPayment = cl.totalPayment(PurchasementLL.getSize(), hasBodykit);
                System.out.print("Total Payment: RM" + finalPayment + "\n");
            } else if (option == 2) {
                System.out.print(cl.toString());
            } else if (option == 3) {
                System.out.print("\n1 Name" + "\n2 IC Number" + "\n3 Phone Number" + "\n4
Email"
                        + "\nPlease select info to change: ");
                int editOption = sc.nextInt();

                if (editOption == 1) {
                    System.out.print("Enter new name: ");
                    name = sc.next();
                    cl.setName(name);
                } else if (editOption == 2) {
                    System.out.print("Enter new IC number: ");
                    noIC = sc.next();
                    cl.setNoIC(noIC);
                } else if (editOption == 3) {
                    System.out.print("Enter new phone number: ");
                    noPhone = sc.next();
                    cl.setNoPhone(noPhone);
                } else if (editOption == 4) {
                    System.out.print("Enter new email: ");
                    Email = sc.next();
                    cl.setCustEmail(Email);
                }

            } else if (option == 4) {
                for (int i = 0; i < amountChoice; i++) {

                    System.out.println("\nPurchasement Plan " + (i + 1) + PD[i].toString() + "\n");
                }

            } else if (option == 5) {
                System.out.print("Enter Name of the person that pickup to change
Purchasement Plan detail : ");
                String pName = sc.next();

                PurchasementDetail pd = (PurchasementDetail) PurchasementLL.getFirst();//

                while (pd != null) {
                    if (pd.getPickupName().contains(pName)) {
                        System.out.print("1 - Name of the person that pickup the car\n2 - Car
Color\n3 - Rim Color\n4 - Purchase Date\n5 - Desired Location\nChoose information to
change: ");
                        int editPref = sc.nextInt();

                        if (editPref == 1) {
                            System.out.print("Enter new name of the person that pickup the car : ");
                            String pickupName = sc.next();
```

```
                    pd.setPickupName(pickupName);
                }
                if (editPref == 2) {
                    System.out.print("Enter new Car Color you Preferred : ");
                    String carColor = sc.next();
                    pd.setCarColor(carColor);
                } else if (editPref == 3) {
                    System.out.print("Enter new Rim Color you Preferred : ");
                    String rColor = sc.next();
                    pd.setRimColor(rColor);
                } else if (editPref == 4) {
                    System.out.print("Enter new purchase date: ");
                    String pDate = sc.next();
                    pd.setPurchaseDate(pDate);
                } else if (editPref == 5) {
                    System.out.print("Enter new desired location: ");
                    String DLoc = sc.next();
                    pd.setDesiredLocation(DLoc);
                }
            }
            pd = (PurchasementDetail) PurchasementLL.getNext();

        }
    }
    else if (option == 6) {
            System.out.print("Enter name of the person that pickup the car for Final
Confirmation : ");
            String pName = sc.next();
            PurchasementDetail pd = (PurchasementDetail)
PurchasementLL.getFirst();

            while (pd != null) {
                if (pd.getPickupName().contains(pName)) {
                    System.out.println(pd.toString());
                }
                pd = (PurchasementDetail) PurchasementLL.getNext();
            }
        }

        System.out.print("\n\n0. End program"
                + "\n1. Calculate total payment"
                + "\n2. View personal details"
                + "\n3. Change personal information"
                + "\n4. View Purchasement Plan details"
                + "\n5. Change Purchasement Plan details"
                + "\n6. Search Purchasement Plan details by name of the person that
pickup for Confirmation "
                + "\nChoose your option: ");
        option = sc.nextInt();
    }
```

```
        pw.println("\n" + "-".repeat(80));
        pw.println("                HUA Refurbished Car Dealership     ");
        pw.println("-".repeat(80));
        pw.println("\n🎯 Client Details 🎯");
        pw.println("-".repeat(80));
        pw.println("Name: " + name);
        pw.println("IC: " + noIC);
        pw.println("Phone: " + noPhone);
        pw.println("Email: " + Email);
        pw.println("\nChosen Car: " + chosenCar);
        pw.println("\n🚗 Car Purchasement Plan Detail List 🚗");
        pw.println("-".repeat(80));
        pw.println(String.format("%-20s %-15s %-15s %-20s %-20s",
        "Name for Pickup", "Car Color", "Rim Color", "Purchase Date", "Location"));
        pw.println("-".repeat(80));

PurchasementDetail pd = (PurchasementDetail) PurchasementLL.getFirst();

while (pd != null) {
    pw.println(String.format("%-20s %-15s %-15s %-20s %-20s",
            pd.getPickupName(), // Added Name for Pickup
            pd.getCarColor(),
            pd.getRimColor(),
            pd.getPurchaseDate(),
            pd.getDesiredLocation()));
    pd = (PurchasementDetail) PurchasementLL.getNext();
}


        pw.println("\n💰 Total Payment 💰");
        pw.println("-".repeat(80));
        pw.println("Total Payment: RM" + finalPayment);
        pw.println("-".repeat(80));

        System.out.println("========================================");
        System.out.println("           Payment Invoice             ");
        System.out.println("========================================");
        System.out.println(cl.toString());
        System.out.println("========================================");
        System.out.println("         Chosen Car :" + chosenCar);
        System.out.println("========================================");
        System.out.println("Total Price : RM" + finalPayment);
        System.out.println("========================================");

        sc.close();
        pw.close();
    } catch(FileNotFoundException fe){
        System.out.println(fe.getMessage());
    } catch(Exception e){
        System.err.println("Error: " + e.getMessage());
```

```
            } finally{
                System.out.println("\nThank you for dealing from us!");
                System.out.println("\nWe look forward to your arrival in our shop for further
business ");
            }


        }
    }
```

CarDealerQ.java

```
import java.io.*;
import java.util.*;

public class CarDealerQ {

    public static void main(String[] args) {
        try {
            Scanner sc = new Scanner(System.in);
            Scanner sc1 = new Scanner(System.in);
            Client cl = new Client();
            Queue<PurchasementDetail> PurchasementQ = new Queue<>();
            Queue<Car> CarQ = new Queue<>();
            Queue<PurchasementDetail> PurchasementTemp = new Queue<>();
            Queue<Car> CarTemp = new Queue<>();
            Car obj;
            PurchasementDetail obj1;


            FileReader fr = new FileReader("C:\\Users\\user\\Documents\\CarDetail.txt");
            BufferedReader br = new BufferedReader(fr);
            FileWriter wr = new
FileWriter("C:\\Users\\user\\Documents\\PurchasementDetails.txt");
            PrintWriter pw = new PrintWriter(wr);

            String strLine;

            while ((strLine = br.readLine()) != null) {
                try {
                    String[] data = strLine.split(",");
                    if (data.length != 5) {
                        throw new IllegalArgumentException("Invalid data format: " + strLine);
                    }

                    String carName = data[0].trim();
                    double price = Double.parseDouble(data[1].trim());
                    int yearMade = Integer.parseInt(data[2].trim());
```

```
                double mileage = Double.parseDouble(data[3].trim());
                String condition = data[4].trim();

                Car car = new Car(carName, price, yearMade, mileage, condition);
                CarQ.enqueue(car);
            } catch (Exception e) {
                System.err.println("Error parsing car details: " + strLine + ". Skipping this line.");
            }
        }


System.out.println("╔══════════════════════════════════════════════════════
════════╗");
        System.out.println("║                                                  ║");
        System.out.println("║    Welcome to HUA Refurbished Car Dealership     ║");
        System.out.println("║                                                  ║");

System.out.println("╚══════════════════════════════════════════════════════
════════╝");
        System.out.println("--------------------------------------");
        System.out.println("🌍  Your Car, Anywhere in Malaysia  🌍");
        System.out.println("--------------------------------------");
        System.out.println("✓ 100+ branches nationwide.");
        System.out.println("✓ Make deals anytime, anywhere with your preference and
versatility.");
        System.out.println("✓ Many options for customer's main plan and backup plan.");
        System.out.println("✓ Perfect for personal or gift purposes.");
        System.out.println("--------------------------------------");
        System.out.print("\nPlease Enter your Personal Detail  ");
        System.out.print("\n\nEnter Name: ");
        String name = sc.nextLine();
        System.out.print("Enter IC Number: ");
        String noIC = sc.nextLine();
        System.out.print("Enter Phone Number: ");
        String noPhone = sc.nextLine();
        System.out.print("Enter Email: ");
        String Email = sc.nextLine();
        System.out.print("\nCustomer that give many provides many purchasement plan will
ease our dealer work ");
        System.out.print("\n!! RM500 DISCOUNT will be given if there is more than 1 !! ");
        System.out.print("\nEnter Number Of Purchasement Plan: ");
        int amountChoice = sc.nextInt();

        PurchasementDetail[] PD = new PurchasementDetail[amountChoice];

        for (int i = 0; i < amountChoice; i++) {
            PD[i] = new PurchasementDetail();
            System.out.print("\nPurchasement Plan " + (i + 1));
            System.out.print("\nName of the person that pickup the car: ");
            String pickupName = sc.next();
```

```
    System.out.print("Enter Car Color: ");
    String carColor = sc.next();
    sc.nextLine();
    System.out.print("Enter Rim Color: ");
    String rimColor = sc.nextLine();
    System.out.print("Enter Purchase Date [dd/mm/yyyy]: ");
    String purchaseDate = sc.nextLine();
    System.out.print("Enter Desired Purchase Location: ");
    String desiredLocation = sc.nextLine();

    PD[i].setDetails(pickupName,carColor, rimColor, purchaseDate, desiredLocation);
    PurchasementQ.enqueue(PD[i]);
}

System.out.println("Available cars:");
while (!CarQ.isEmpty()) {
    obj = CarQ.dequeue();
    System.out.println(obj.toString());
    CarTemp.enqueue(obj);
}

while (!CarTemp.isEmpty()) {
    CarQ.enqueue(CarTemp.dequeue());
}

System.out.print("\nChoose your car: ");
String chosenCar = sc.nextLine(); // Use nextLine to capture the full car name
System.out.print("Do You Want the Car Bodykit Package (Y/N): ");
char hasBodykit = sc.nextLine().charAt(0);
double basePrice = 0.0; // Updated to handle car not found scenario


while (!CarQ.isEmpty()) {
    obj = CarQ.dequeue();
    if (obj.getCarName().equalsIgnoreCase(chosenCar)) {
        basePrice = obj.getPrice();
    }
    CarTemp.enqueue(obj);
}

while (!CarTemp.isEmpty()) {
    CarQ.enqueue(CarTemp.dequeue());
}

if (basePrice == 0.0) {
    throw new Exception("Selected car not found in inventory.");
}

cl.setDetails(name, noIC, noPhone, Email, basePrice, amountChoice, hasBodykit);

System.out.print("\n\n0. End program"
```

```
                + "\n1. Calculate total payment"
                + "\n2. View personal details"
                + "\n3. Change personal information"
                + "\n4. View Purchasement Plan details"
                + "\n5. Change Purchasement Plan details"
                + "\n6. Search Purchasement Plan details by name of the person that pickup for
Confirmation "
                + "\nChoose your option: ");
        int option = sc.nextInt();

        double totalPayment = 0.0;

        while (option != 0) {
            if (option == 1) {
                totalPayment = cl.totalPayment(PurchasementQ.size(), hasBodykit);
                System.out.print("Total Payment: RM" + totalPayment + "\n");
            } else if (option == 2) {
                System.out.print(cl.toString());
            } else if (option == 3) {
                System.out.print("\n1 Name" + "\n2 IC Number" + "\n3 Phone Number" + "\n4
Email"
                        + "\nPlease select info to change: ");
                int editOption = sc.nextInt();

                if (editOption == 1) {
                    System.out.print("Enter new name: ");
                    name = sc.next();
                    cl.setName(name);
                } else if (editOption == 2) {
                    System.out.print("Enter new IC number: ");
                    noIC = sc.next();
                    cl.setNoIC(noIC);
                } else if (editOption == 3) {
                    System.out.print("Enter new phone number: ");
                    noPhone = sc.next();
                    cl.setNoPhone(noPhone);
                } else if (editOption == 4) {
                    System.out.print("Enter new email: ");
                    Email = sc.next();
                    cl.setCustEmail(Email);
                }

            } else if (option == 4) {
                for (int i = 0; i < amountChoice; i++) {
                    System.out.println("\nPurchasement Detail " + (i + 1) + PD[i].toString() + "\n");
                }
            } else if (option == 5) {
                System.out.print("Enter Name of the person that pickup to change
Purchasement detail : ");
                String pName = sc.next();
```

```
                while (!PurchasementQ.isEmpty()) {
                    obj1 = PurchasementQ.dequeue();
                    if (obj1.getPickupName().contains(pName)) {
                        System.out.print("1 - Name of the person that pickup the car\n2 - Car
Color\n3 - Rim Color\n4 - Purchase Date\n5 - Desired Location\nChoose information to
change: ");
                        int editPref = sc.nextInt();
                        if (editPref == 1) {
                            System.out.print("Enter new name of the person that pickup the car :  ");
                            String pickupName = sc.next();
                            obj1.setPickupName(pickupName);
                        } else if (editPref == 2) {
                            System.out.print("Enter new Car Color you Preferred : ");
                            String carColor = sc.next();
                            obj1.setCarColor(carColor);
                        } else if (editPref == 3) {
                            System.out.print("Enter new Rim Color you Preferred : ");
                            String rColor = sc.next();
                            obj1.setRimColor(rColor);
                        } else if (editPref == 4) {
                            System.out.print("Enter new purchase date: ");
                            String pDate = sc.next();
                            obj1.setPurchaseDate(pDate);
                        } else if (editPref == 5) {
                            System.out.print("Enter new desired location: ");
                            String DLoc = sc.next();
                            obj1.setDesiredLocation(DLoc);
                        }
                    }
                    PurchasementTemp.enqueue(obj1);
                }
                while (!PurchasementTemp.isEmpty()) {
                    PurchasementQ.enqueue(PurchasementTemp.dequeue());
                }
            } else if (option == 6) {
                System.out.print("Enter name of the person that pickup the car for Confirmation
: ");
                String pName = sc.next();
                while (!PurchasementQ.isEmpty()) {
                    obj1 = PurchasementQ.dequeue();
                    if (obj1.getPickupName().contains(pName)) {
                        System.out.println(obj1.toString());
                    }
                    PurchasementTemp.enqueue(obj1);
                }
                while (!PurchasementTemp.isEmpty()) {
                    PurchasementQ.enqueue(PurchasementTemp.dequeue());
                }
            }

            System.out.print("\n\n0. End program"
```

```
                + "\n1. Calculate total payment"
                + "\n2. View personal details"
                + "\n3. Change personal information"
                + "\n4. View Purchasement details"
                + "\n5. Change Purchasement details"
                + "\n6. Search Purchasement details by name of the person that pickup for
Confirmation "
                + "\nChoose your option: ");
            option = sc.nextInt();
        }

        pw.println("\n" + "-".repeat(80));
        pw.println("                HUA Refurbished Car Dealership     ");
        pw.println("-".repeat(80));
        pw.println("\n🎯 Client Details 🎯");
        pw.println("-".repeat(80));
        pw.println("Name: " + name);
        pw.println("IC: " + noIC);
        pw.println("Phone: " + noPhone);
        pw.println("Email: " + Email);
        pw.println("\nChosen Car: " + chosenCar);
        pw.println("\n🚗 Car Purchasement Detail List 🚗");
        pw.println("-".repeat(80));
        pw.println(String.format("%-20s %-15s %-15s %-20s %-20s",
              "Name for Pickup", "Car Color", "Rim Color", "Purchase Date", "Location"));
        pw.println("-".repeat(80));

        while (!PurchasementQ.isEmpty()) {
            obj1 = PurchasementQ.dequeue();
            pw.println(String.format("%-20s %-15s %-15s %-20s %-20s",
                    obj1.getPickupName(), // Added Name for Pickup
                    obj1.getCarColor(),
                    obj1.getRimColor(),
                    obj1.getPurchaseDate(),
                    obj1.getDesiredLocation()));
            PurchasementTemp.enqueue(obj1);
        }

        while (!PurchasementTemp.isEmpty()) {
            PurchasementQ.enqueue(PurchasementTemp.dequeue());
        }


        pw.println("\n💰 Total Payment 💰");
        pw.println("-".repeat(80));
        pw.println("Total Payment: RM" + totalPayment);
        pw.println("-".repeat(80));
```

```
            System.out.println("=======================================");
            System.out.println("            Payment Invoice            ");
            System.out.println("=======================================");
            System.out.println(cl.toString());
            System.out.println("=======================================");
            System.out.println("         Chosen Car :" +chosenCar        );
            System.out.println("=======================================");
            System.out.println("Total Price : RM" + totalPayment);
            System.out.println("=======================================");

            sc.close();
            pw.close();
        } catch (FileNotFoundException fe) {
            System.out.println(fe.getMessage());
        } catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
        } finally {
            System.out.println("\nThank you for dealing from us!");
            System.out.println("\nWe look forward to your arrival in our shop for further business
");
        }
    }
}
```