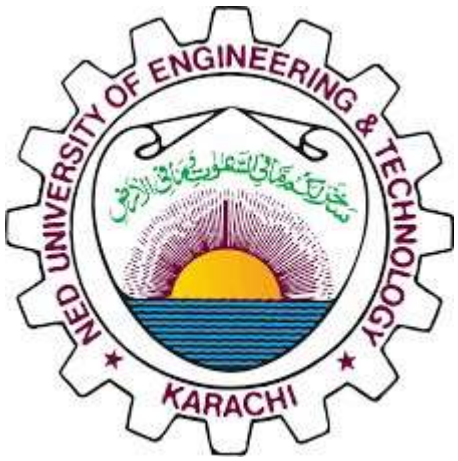


NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Formal Methods in Software Engineering

Formal Specification Document for

"Car Rental System"

Group

- Saqlain Ahmad (SE-21037)
- Maqsood Ahmed (SE-21040)

Table of Contents

1. CAR RENTAL SYSTEM……2
2. 4 + 1 Architectural View 4
   2.1. Logical View 4
   2.2. Process View 5
   2.3. Physical View 6
   2.4. Development View 6
   2.5. +1 Scenarios 6
3. The complete specification of *Car Rental System* 7
4. Java Implementation 9
5. Testing Class 16

CAR RENTAL SYSTEM

DESCRIPTION: The Car Rental System project is a comprehensive console-based application implemented in Java. Aimed at facilitating the rental process for both customers and rental service providers, this system offers a user-friendly interface for seamless interaction. Developed with an emphasis on simplicity and functionality, the program integrates essential features such as user authentication, detailed car model information retrieval, rental fee calculation, and the generation of comprehensive customer invoices.

Requirements: Design and implement a software system which should fulfill the following:

1. Users must be able to log in securely with a password to access the Car Rental System.
2. The system should provide users with a list of available car models to choose from.
3. Users should be able to input their name and select a car model, specifying the number of days they wish to rent the car.
4. Upon successful completion of the rental process, the system must generate a detailed invoice, including customer information, selected car details, rental duration, and total rental amount.

Specifications:

- The Car Rental System should be designed to run on multiple platforms, ensuring compatibility with various operating systems.
- Rental details and car information should be stored persistently to prevent data loss and allow for future reference.
- The system's user interface should be intuitive and easy to interact, facilitating a smooth user experience.
- The system should implement robust security measures, such as password, to protect user data and maintain the confidentiality of sensitive information.
- The architecture should be scalable to accommodate future expansions, such as adding more car models or extending functionalities.

Functionalities:

Car Model Details Display:

- Display detailed information about each car model, including specifications and features, when the user selects a particular model.

Rental Fee Calculation:

- Calculate the rental fee based on the selected car model and the number of days the user wishes to rent the car.

Invoice Printing:

- Provide functionality to print or display the generated invoice for the user.

Administrator Access:

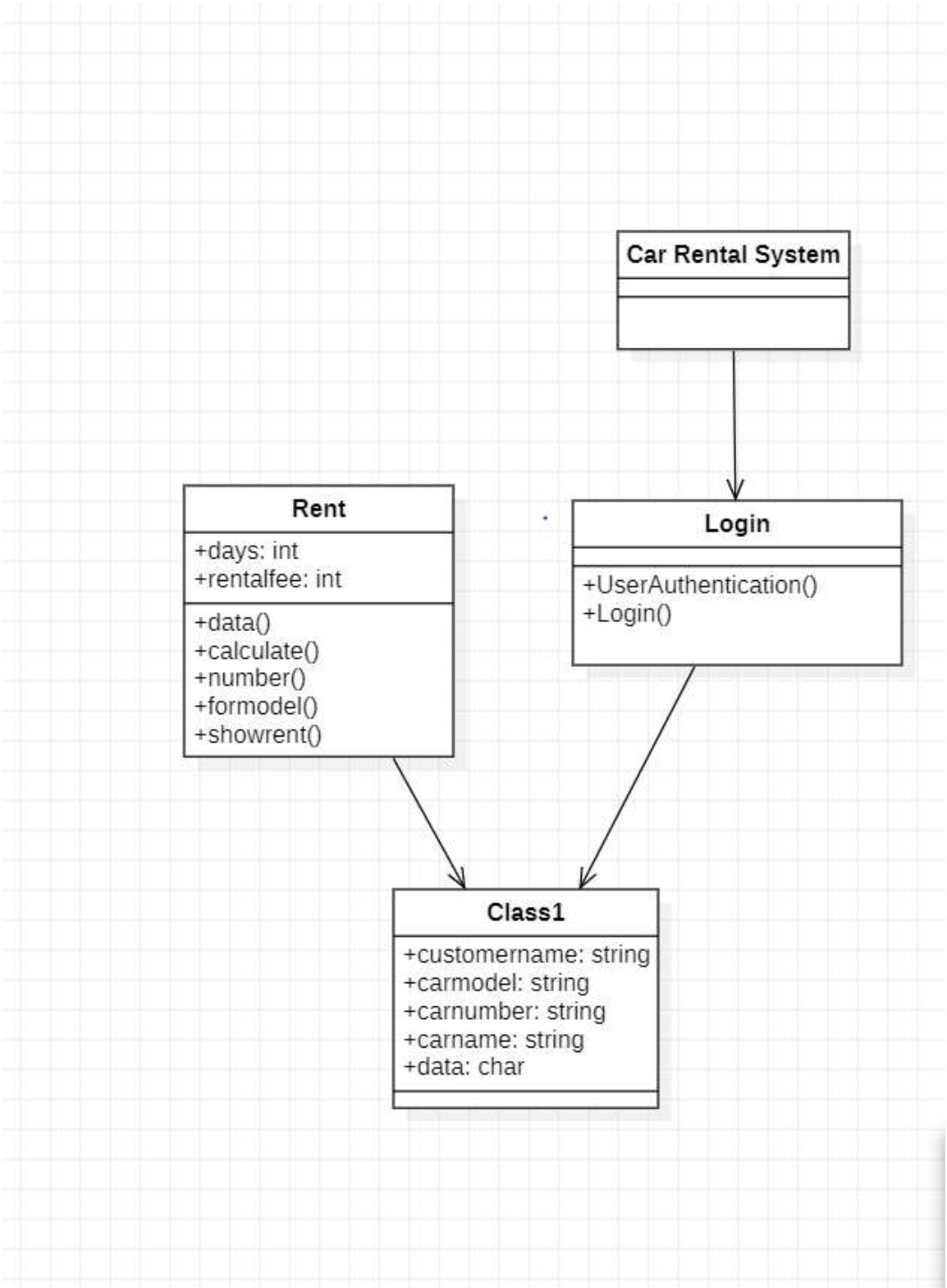
- Implement administrator functionalities for managing the car inventory, ensuring an organized and up-to-date system.

Constraints:

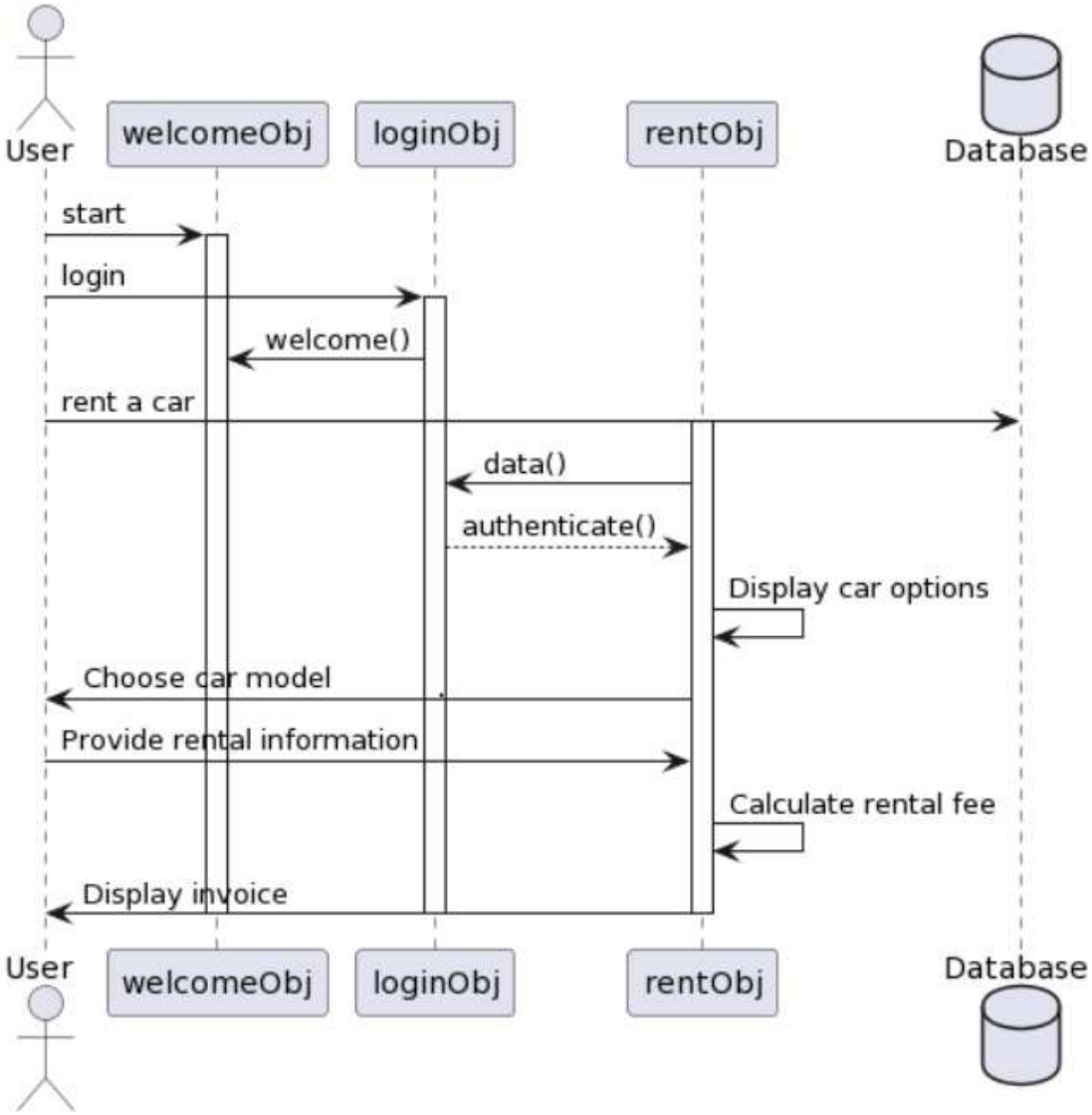
- The system must adhere to data privacy regulations, and any data collected from users should be handled in compliance with relevant laws.
- The system's performance should be optimized to handle a reasonable number of simultaneous users without significant degradation in response times.

1. 4 + 1 Architectural View

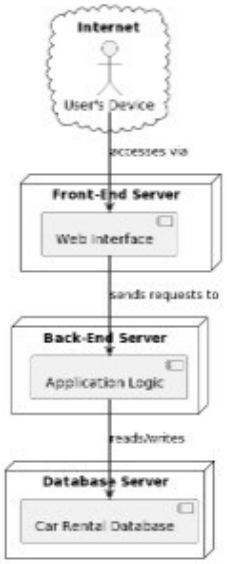
1.1. Logical View



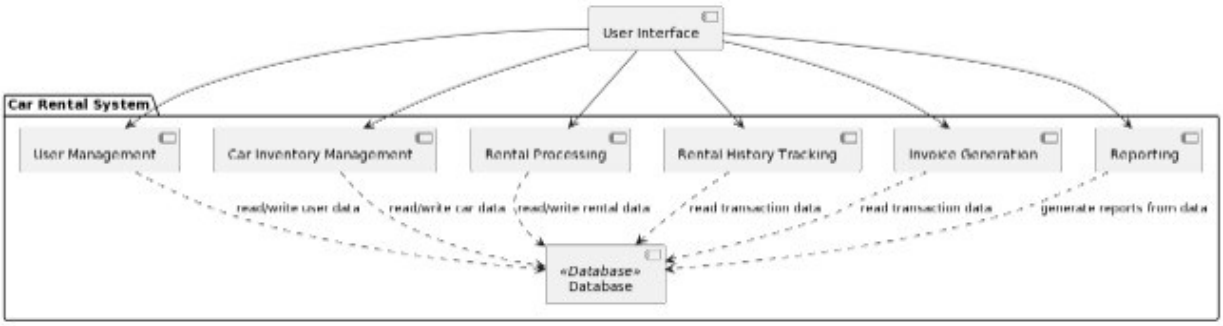
1.2. Process View



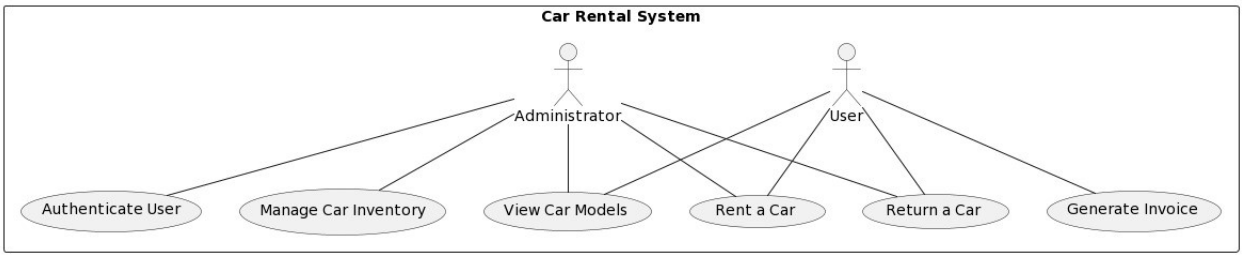
1.3. Physical View



1.4. Development View



1.5. +1 Scenarios



2. The VDM specification of *Car Rental* System

Types
Price = ℝ;
RentalDays = ℝ;
Carmodels = char;

values
MAX_PRICE: Price = 10000.0;
MIN_PRICE: Price = 500.0;

state *CarRentalSystem* **of**
Credentials: [credentials]
Carmodels: [model]
RentalDays: [days]
invoicedetails: [details]

-- CarModel and RentalDays must not be nill and rentaldays must be greater than 0
inv mk-CarRentalSystem (cr,cm,r,i) (inRange(cm) ∨ = **nil**) ∧ (inRange(r) ∨ = **nil**) ∧ (r > 0)

-- CarModel and RentalDays must be undefined when the system is initialized
init mk-CarRentalSystem (cr,cm,r,i) cm = nil ∧ r = **nil**;

end

functions
inRange:(val: Price) result: B
pre True
post RESULT ⇔ 500.0≤ val ≤ 10000.0;

-- This function will calculate the rental fee
CalculateRentalFee: CarModel * RentalDays result: price
pre true
post RESULT = availableCars(CarModel) * RentalDuration;

operations

Login: (cr:Credentials)

ext wr Credentials: [Credentials]

pre

IsValidCredentials(username, password) ∧ loggedInUser = nil

post

loggedInUser = username;

DisplayAvailableCars: (model:carmodel)

ext rd carmodels: [models]

pre

IsUserLoggedIn() ∧ availableCars()

post

true;

SelectCar: (model:CarModel)

ext rd carmodels: [models]

pre

IsUserLoggedIn() ∧ availableCars()∧ model in set availableCars() ∧ selectedCar = nil

post

selectedCar = model;

SelectRentalDays: (rentdays:days)

ext wr RentalDays: [days]

pre

IsUserLoggedIn() ∧ selectedCar ≠ nil ∧ days > 0

post

rentalDays = days;

GenerateInvoice: (indetails:details)

ext wr invoicedetails: [details]

pre

IsUserLoggedIn() ∧ selectedCar ≠ nil ∧ rentalDays ≠nil

post

generatedInvoice = invoicedetails;

end CarRentalSystem

3. Java Implementation

Code:

```
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

class Customer {
    public String customerName;
    public String carModel;
    public String carNumber;
    public String carName;
    public char data;
}

class Rent extends Customer {
    public int days = 0;
    public int rentalFee = 0;

    public void inputData() {
        login();
        Scanner scanner = new Scanner(System.in);
        System.out.print("\t\t\t\tPlease Enter your Name: ");
        customerName = scanner.nextLine();
        System.out.println();

        do {
            System.out.println("\t\t\t\tPlease Select a Car");
            System.out.println("\t\t\t\tEnter 'A' for HONDA");
            System.out.println("\t\t\t\tEnter 'B' for TOYOTA");
            System.out.println("\t\t\t\tEnter 'C' for SUZUKI");
            System.out.println("\t\t\t\tEnter 'D' for KIA");
            System.out.println("\t\t\t\tEnter 'E' for CHANGAN");
            System.out.println();
            System.out.print("\t\t\t\tChoose a Car from the above options: ");
            carModel = scanner.nextLine();
            System.out.println("-----------------------------------------------
-------------------------");

            if ("A".equalsIgnoreCase(carModel)) {
                System.out.println("You have chosen HONDA");
                displayCarDetails("A.txt");
                sleep(2000);
            } else if ("B".equalsIgnoreCase(carModel)) {
```

```
                System.out.println("You have chosen TOYOTA");
                displayCarDetails("B.txt");
                sleep(2000);
            } else if ("C".equalsIgnoreCase(carModel)) {
                System.out.println("You have chosen SUZUKI");
                displayCarDetails("C.txt");
                sleep(2000);
            } else if ("D".equalsIgnoreCase(carModel)) {
                System.out.println("You have chosen KIA");
                displayCarDetails("D.txt");
                sleep(2000);
            } else if ("E".equalsIgnoreCase(carModel)) {
                System.out.println("You have chosen CHANGAN");
                displayCarDetails("E.txt");
                sleep(2000);
            } else {
                System.out.println("Invalid Car Model. Please try again!");
            }
        } while (!("A".equalsIgnoreCase(carModel) ||
"B".equalsIgnoreCase(carModel) || "C".equalsIgnoreCase(carModel)
                || "D".equalsIgnoreCase(carModel) ||
"E".equalsIgnoreCase(carModel)));

        System.out.println("-----------------------------------------------------
----------------------");
        System.out.println("Please provide the following information:");
        System.out.print("Number of days you wish to rent the car: ");
        days = scanner.nextInt();
        System.out.println();
    }

    public void calculate() {
        sleep(1000);
        clearScreen();
        System.out.println("Calculating rent. Please wait......");
        sleep(2000);

        if ("A".equalsIgnoreCase(carModel)) {
            rentalFee = days * 5000;
        } else if ("B".equalsIgnoreCase(carModel)) {
            rentalFee = days * 4000;
        } else if ("C".equalsIgnoreCase(carModel)) {
            rentalFee = days * 3000;
        } else if ("D".equalsIgnoreCase(carModel)) {
            rentalFee = days * 2000;
```

```
        } else if ("E".equalsIgnoreCase(carModel)) {
            rentalFee = days * 3500;
        }
    }

    public void displayCarDetails(String fileName) {
        try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void displayInvoice() {
        System.out.println("\n\t\t                        Car Rental - Customer
Invoice                   ");
        System.out.printf("\t\t\t| Invoice No. :%18s |%n", "SE-313");
        System.out.printf("\t\t\t| Customer Name:%18s |%n", customerName);
        number();
        formodel();
        System.out.printf("\t\t\t| Car Name:%22s |%n", carName);
        System.out.printf("\t\t\t| Car Number:%20s |%n", carNumber);
        System.out.printf("\t\t\t| Number of days :%18s |%n", days);
        System.out.printf("\t\t\t| Your Rental Amount is :%12s |%n", rentalFee);
        System.out.printf("\t\t\t| Advanced :%23s |%n", "0");
        System.out.println("\t\t\t
_____________________________________________________\n");
        System.out.printf("\t\t\t| Total Rental Amount is :%14s  RS |%n",
rentalFee);
        System.out.println("\t\t\t
_____________________________________________________\n");
        System.out.println(" ");
        System.out.println("\t\t\tYou are advised to pay up the amount before the
due date.");
        System.out.println("\t\t\tOtherwise penalty fee will be applied");

        int f;
        pause();
        clearScreen();
```

```
        try (BufferedReader reader = new BufferedReader(new
FileReader("thanks.txt"))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void number() {
        switch (carModel.toUpperCase()) {
            case "A":
                carNumber = "AGY-625";
                break;
            case "B":
                carNumber = "TCF-101";
                break;
            case "C":
                carNumber = "AKW-518";
                break;
            case "D":
                carNumber = "CRZ-567";
                break;
            case "E":
                carNumber = "KJL-589";
                break;
        }
    }

    public void formodel() {
        switch (carModel.toUpperCase()) {
            case "A":
                carName = "HONDA";
                break;
            case "B":
                carName = "TOYOTA";
                break;
            case "C":
                carName = "SUZUKI";
                break;
            case "D":
                carName = "KIA";
                break;
```

```
            case "E":
                carName = "CHANGAN";
                break;
        }
    }

    public void sleep(int milliseconds) {
        try {
            Thread.sleep(milliseconds);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public void pause() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Press Enter to continue...");
        scanner.nextLine();
    }

    public void clearScreen() {
        System.out.print("\033[H\033[2J");
        System.out.flush();
    }

    private void login() {
        String pass = "";
        char ch;
        System.out.println("\n\n\t\t\t\t\t\t       CAR RENTAL SYSTEM \n\n");
        System.out.println("\n\t\t\t\t\t       PREPARED BY: \n\n");
        System.out.println("\n\n\t\t\t\t\t\t       MAQSOOD AHMED (SE-21040) &
SHAIKH SAQLAIN AHMAD (SE-21037)\n\n");
        System.out.println("\n\n\t\t\t\t\t\t       SUBMITTED TO: \n\n");
        System.out.println("\n\t\t\t\t\t\t        PROFESSOR. MUSTAFA LATIF FMS(SE-
313) \n\n");
        System.out.println("\t\t\t\t\t------------------------------");
        System.out.println("\n\t\t\t\t\t\t\t     LOGIN \n");
        System.out.println("\t\t\t\t\t------------------------------\n\n");
        System.out.print("\t\t\t\t\tEnter Password: ");

        Scanner scanner = new Scanner(System.in);
        ch = scanner.next().charAt(0);
        pass = pass + ch;

        if (pass.equals("s")) {
```

```
            System.out.println("\n\n\n\t\t\t\t\t\tAccess Granted! \n");
            pause();
            clearScreen();
        } else {
            System.out.println("\n\n\t\t\t\t\t\t\tAccess
Aborted...\n\t\t\t\t\t\t\tPlease Try Again\n\n");
            pause();
            clearScreen();
            login();
        }
    }
}

class Welcome {
    public void welcomeMessage() {
        displayAsciiArt("welcome.txt");
        sleep(1000);
        System.out.println("\nStarting the program. Please wait...");
        sleep(1000);
        System.out.println("Loading up files...");
        sleep(1000);
        clearScreen();
    }

    public void displayAsciiArt(String fileName) {
        try (BufferedReader reader = new BufferedReader(new
FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                System.out.println(line);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void sleep(int milliseconds) {
        try {
            Thread.sleep(milliseconds);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public void clearScreen() {
```

```
        System.out.print("\033[H\033[2J");
        System.out.flush();
    }
}
```

Output:

```
1. Rent a Car
2. Exit
Choose an option: 2
Closing the program! Good bye :)
PS C:\Users\saqla>
```

4. Testing Class

Code:

```
public class CarRentalSystem {
    //menu driven test class
    public static void main(String[] args) {
        Welcome welcome = new Welcome();
        welcome.welcomeMessage();

        Scanner scanner = new Scanner(System.in);
        boolean exitProgram = false;

        while (!exitProgram) {
            System.out.println("1. Rent a Car");
            System.out.println("2. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    Rent rent = new Rent();
                    rent.inputData();
                    rent.calculate();
                    rent.displayInvoice();
                    break;
                case 2:
                    System.out.println("Closing the program! Good bye :)");
                    exitProgram = true;
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }

        scanner.close();
    }
}
```

Output:

```
                        CAR RENTAL SYSTEM


                        PREPARED BY:


                        MAQSOOD AHMED (SE-21040) & SHAIKH SAQLAIN AHMAD (SE-21037)


                        SUBMITTED TO:


                         PROFESSOR. MUSTAFA LATIF FMS(SE-313)

                    ------------------------------
                              LOGIN
                    ------------------------------

                    Enter Password: s


                         Access Granted!
```

```
                                  Please Enter your Name: saqlain

                                  Please Select a Car
                                  Enter 'A' for HONDA
                                  Enter 'B' for TOYOTA
                                  Enter 'C' for SUZUKI
                                  Enter 'D' for KIA
                                  Enter 'E' for CHANGAN

                                  Choose a Car from the above options: A
--------------------------------------------------------------------------------
You have chosen HONDA
```

```
-----------------------------------------------------
Please provide the following information:
Number of days you wish to rent the car: 6
```

```
Calculating rent. Please wait......

                              Car Rental - Customer Invoice
                         | Invoice No. :            SE-313 |
                         | Customer Name:           saqlain |
                         | Car Name:                 HONDA |
                         | Car Number:             AGY-625 |
                         | Number of days :                5 |
                         | Your Rental Amount is :       25000 |
                         | Advanced :                      0 |
                          ____________________________________________________

                         | Total Rental Amount is :        25000  RS |
                          ____________________________________________________


                        You are advised to pay up the amount before the due date.
                        Otherwise penalty fee will be applied
Press Enter to continue...
```

5. Test Cases

Test Case Id	Functionality	Description	Expected Outcome	Actual Outcome	Status
1	Login	User login with valid credentials	Successful login	Logined Successfully	Pass
2	Login	User login with invalid credentials	Login failure	Invalid id or password	Pass
3	Car Model	Display list of available car models	List of models displayed	List displayed	Pass
4	Car Model	Select a car model and input rental duration	Model selected and duration entered	Model selected and duration	Pass
5	Rental Process	Calculate rental fee for a specified duration	Correct rental fee calculated	Fee calculated	Pass
6	Invoice	Generate invoice after successful rental	Detailed invoice generated	Invoice generated	Pass