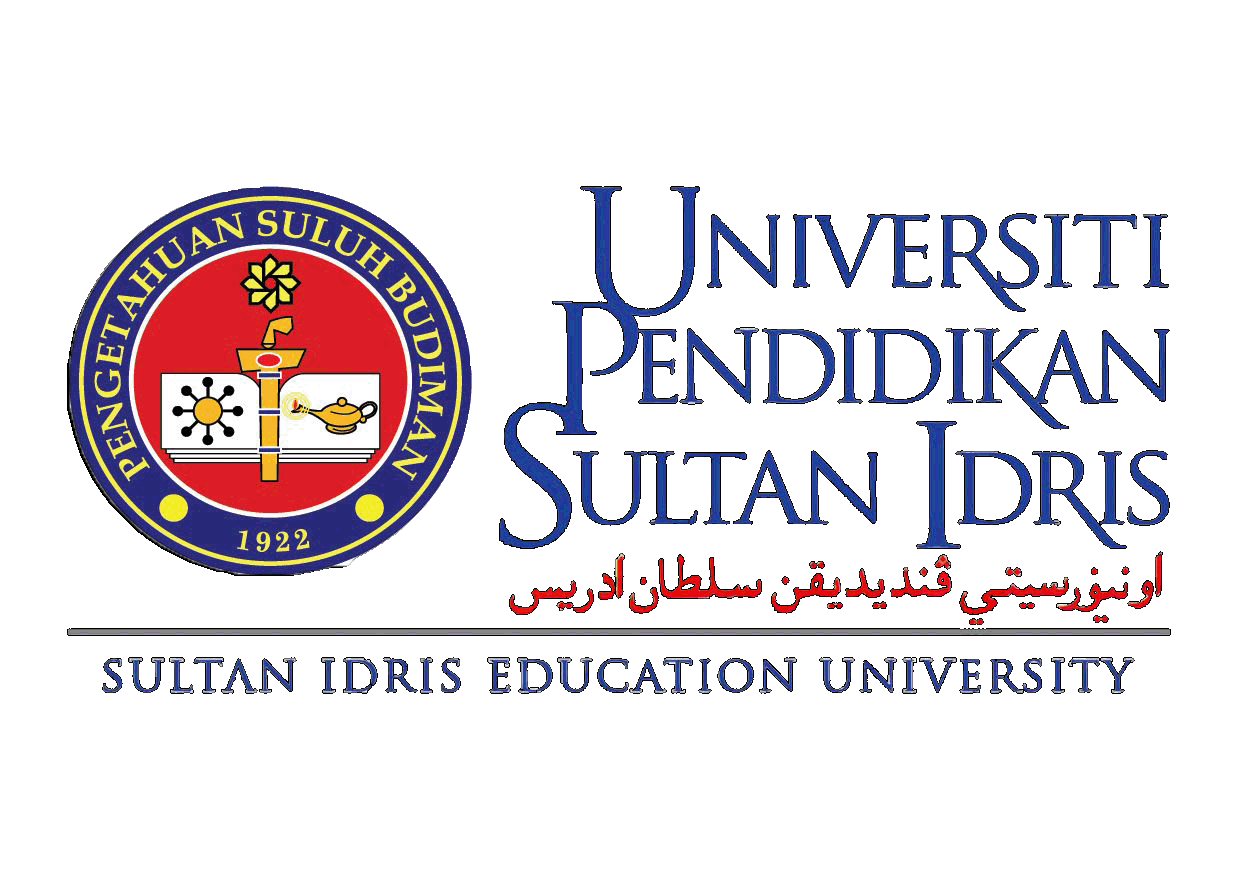
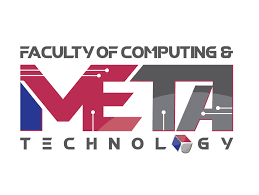
****

# **Assignment 3: Final Project**

**(Bus ticketing Online Booking System)**

|  |  |
| --- | --- |
| **SIYAM MD RAYSUL ISLAM** | **E20231032649** |
| **AHMAD HAZIQ IRSYAD BIN AHMAD HANAFI** | **E20231032527** |
| **NURI SAADAN BIN KHAULID** | **E20231031683** |
| **MUHAMMAD ADAM BIN MD SUHAIZAM** | **E20231032528** |

**Faculty of Computing and Meta Technology, Universiti Pendidikan Sultan Idris**

**Asas Pengaturcaraan Objek (DKS 1073), Diploma in Computer Science (Internet Computing)**

**Puan Asmara binti Alias**

**3 May 2024**

Contents

[**Assignment 3: Final Project** 1](#_Toc165368245)

[Chapter 1: Flow chart of the program 3](#_Toc165368246)

[Chapter 2: Class Diagram with attributes and method 4](#_Toc165368247)

[2.1 Class Diagram 4](#_Toc165368248)

[2.3 Attributes And methods 5](#_Toc165368249)

[Chapter 3: Program explanation with Screenshots and labels 7](#_Toc165368250)

[Chapter 4: Program Code 14](#_Toc165368251)

[1. Main Code: **“BookingSystem.java”** 14](#_Toc165368252)

[2. Additional TXT files 26](#_Toc165368253)

[A. seats,Adik.txt 26](#_Toc165368254)

[B. seats,transN.txt 26](#_Toc165368255)

[C. seats,adam.txt 27](#_Toc165368256)

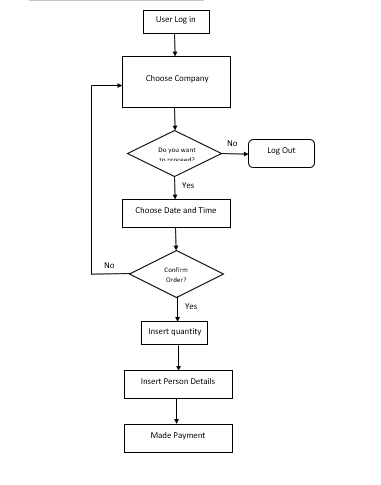
[D. seats,sani.txt 27](#_Toc165368257)

[E. dates.txt 28](#_Toc165368258)

[F. user\_data.txt 29](#_Toc165368259)

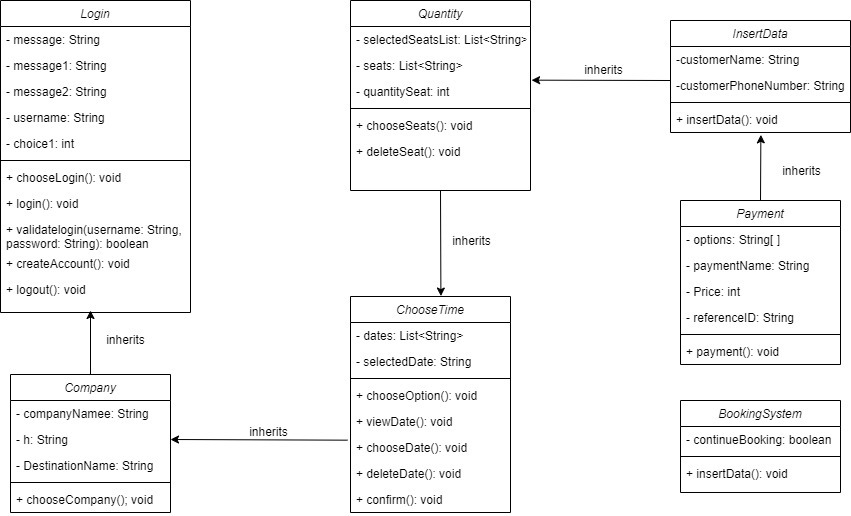
# Chapter 1: Flow chart of the program

Title: **“Bus ticketing Online Booking System”**



# Chapter 2: Class Diagram with attributes and method

## 2.1 Class Diagram



Title: **“Bus ticketing Online Booking System”**

## 2.3 Attributes And methods

**Class Login:**

* Attributes :
  + **message**: String (stores the welcome message)
  + **username**: String (stores the username)
  + **choice1**: int (stores the choice made by the user)
* Methods :
  + **chooseLogin()**: Presents options for login or account creation.
  + **login()**: Handles the login process.
  + **validateLogin(String username, String password)**: Validates the login credentials.
  + **createAccount()**: Handles the account creation process.
  + **logout()**: Handles the logout process.

**Class Company extends Login:**

* Attributes :
  + **companyName**: String (stores the selected company name)
  + **DestinationName**: String
* Methods :
  + **chooseCompany()**: Allows the user to select a bus company and destination.

**Class ChooseTime extends Company:**

* Attributes :
  + **dates**: List<String> (stores available dates)
  + **selectedDate**: String (stores the selected date)
* Methods :
  + **chooseOption()**: Presents options for viewing dates or choosing a date.
  + **viewDates()**: Displays available dates.
  + **chooseDate()**: Allows the user to choose a date for booking.
  + **deleteDate()**: Deletes the selected date from the list.
  + **confirm():** void (Confirm the option)

**Class Quantity extends ChooseTime:**

* Attributes :
  + **selectedSeatsList**: List<String> (stores selected seats)
  + **seats**: List<String> (stores available seats)
  + **quantitySeat**: int (stores the quantity of selected seats)
* Methods :
  + **chooseSeats()**: Allows the user to choose seats for booking.
  + **deleteSeat()**: Deletes the selected seats from the list.

**Class InsertData ectends Quantiity:**

* Attributes **:**
* customerName: String(stores the information of customer’s name)
* customerPhoneNumber: String(stores the information of customer’s Phone Number)
* Methods :
  + **insertData()**: Allows the user to input their name and phone number.

**Class Payment extends InsertData:**

* Attributes :
* options: String[ ]
* paymentName: String
* Price: Int
* referenceID: String
* Methods :
  + **payment()**: Allows the user to select a payment method and calculates the total price.

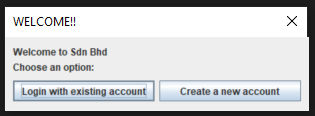
**Class BookingSystem:**

* Methods :
  + **main(String[] args)**: Entry point of the program. Handles the overall booking process.

# Chapter 3: Program explanation with Screenshots and labels

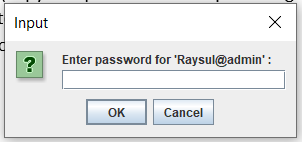
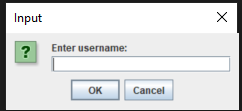
**1.** When the program is opened, it will display a popup message with a welcome banner and company name.

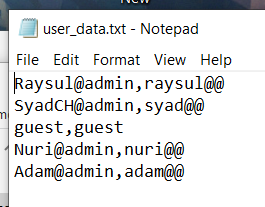
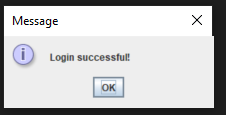
* Upon program start, the user is presented with a dialog box containing two options:
  + "Login with existing account"
  + "Create a new account"

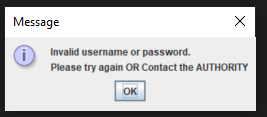


**A.** If the user selects "Login with existing account":

* The program prompts the user to enter their username and password.

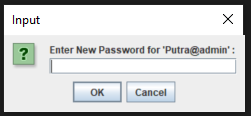
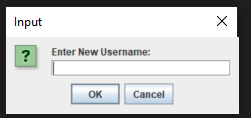


* It validates the login credentials by checking them against the data stored in the "user\_data.txt" file.
* If the credentials are valid, a success message is displayed, and the user is logged in.
* If the credentials are invalid, an error message is displayed, and the program exits.

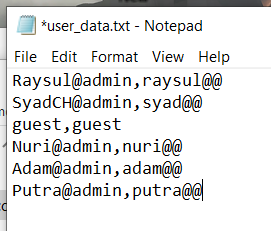


**B.** If the user selects "Create a new account":

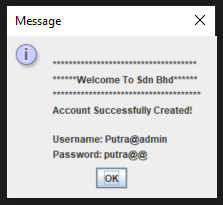
* The program prompts the user to enter a new username and password.



* It saves the new account information in the "user\_data.txt" file.

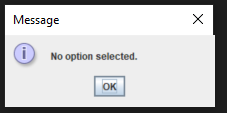


* A success message with the newly created account details is displayed.

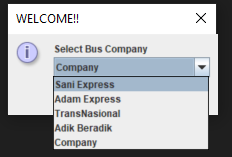
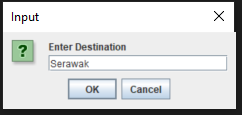


* After successfully creating an account, the program will then open the main welcome menu again to proceed with login.

**C.** If the user exits the program with the cross(X) sign above, the program will show that no option is selected and the program shuts down.

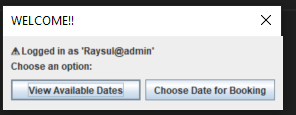


**2.** Choose Bus Company:

* After logging in the user is prompted to select a bus company from a list of options.
* They are also asked to enter the destination.
* The program records the selected bus company and destination.

**3.** Once the bus company and destination are selected, the user is presented with two options:

* "View Available Dates"
* "Choose Date for Booking"



**A.** If the user selects "View Available Dates":

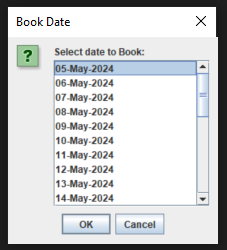
* The program displays a list of available dates for booking from the data stored in the "dates.txt" file.
* After the showing the available dates the

program will go back to choose two options

again to preserve continuity.

**B.** If the user selects "Choose Date for Booking":

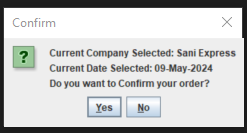
* The program prompts the user to select a date for booking.



* After the program is done the date will automatically be deleted from txt file for the next user to choose different date.

**4.** After selecting the date, the program confirms the user's booking details, including the selected bus company, destination, and date.

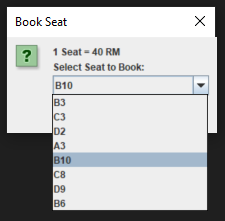
A. The user is asked to confirm their booking.



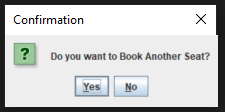
* If confirmed, the program proceeds

B. If the user doesn’t confirm the order the program will be prompted back to choosing the company part.

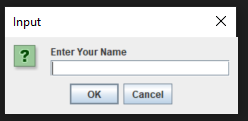
**5.** Next, the user is prompted to select seats for booking.

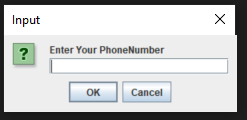


* Every company have their unique seats, the program will read data from txt file of the company selected.
* They can choose multiple seats and continue until they are done.

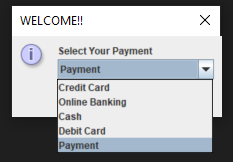


* The program records the selected seats and calculates the total price.
* After the program is finished, the selected seats will be deleted from the txt file.

**6.** After selecting seats, the user is asked to enter their name and phone number.



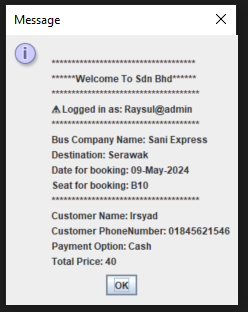
**7.** Finally, the user is prompted to select a payment method from a list of options.



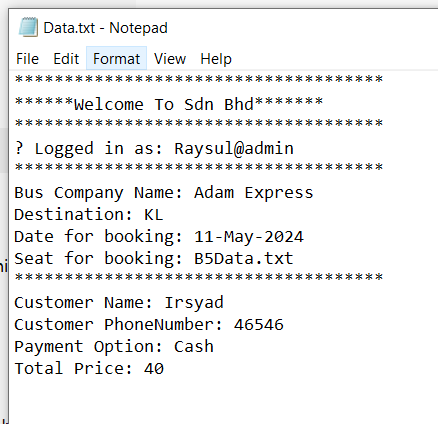
* The program calculates the total price based on the selected seats and displays it along with the chosen payment method.

**8.** Output:

* The booking details, including user information, selected seats, payment method, and total price, are displayed in a message dialog.



* Additionally, the booking details are saved to a file named "Data.txt".



# Chapter 4: Program Code

## 1. Main Code: **“BookingSystem.java”**

import java.io.PrintWriter;

import java.io.IOException;

import javax.swing.\*;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.util.\*;

class Login{

    String message="";

    String message1="";

    String message2="";

    String username;

    int choice1;

        {message+="\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

        message+="\n\*\*\*\*\*\*Welcome To Sdn Bhd\*\*\*\*\*\*\*";

        message+="\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";}

        void chooseLogin(){

            String[] options = {"Login with existing account", "Create a new account"};

        int choice = JOptionPane.showOptionDialog(null, "Welcome to Sdn Bhd"+"\nChoose an option:", "WELCOME!!", JOptionPane.DEFAULT\_OPTION, JOptionPane.PLAIN\_MESSAGE, null, options, options[0]);

        switch (choice) {

            case 0:

                login();

                break;

            case 1:

                createAccount();

                chooseLogin();

                break;

            default:

                JOptionPane.showMessageDialog(null,"No option selected.");

                System.exit(0);

                break;

        }

        }

        void login(){

            boolean loggedIn = false;

            while (!loggedIn) {

                username = JOptionPane.showInputDialog(null, "Enter username: ");

                String password = JOptionPane.showInputDialog(null, "Enter password for '"+ (username) +"' : ");

                // Validate login credentials

                try {

                    if (username == null || password == null || username.isEmpty() || password.isEmpty()) {

                        JOptionPane.showMessageDialog(null, "Username and password cannot be empty. Please try again.");

                    } else if (validateLogin(username, password)) {

                        JOptionPane.showMessageDialog(null,"Login successful!");

                        message+= "\n⚠ Logged in as: "+(username);

                        message+="\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

                        loggedIn = true;

                    } else {

                        JOptionPane.showMessageDialog(null,"Invalid username or password. \nPlease try again OR Contact the AUTHORITY");

                        System.exit(0);

                        break;

                    }

                } catch (IOException e) {

                    System.err.println("Error reading user data: " + e.getMessage());

                    JOptionPane.showMessageDialog(null,"Error reading user data.\n Please Contact the AUTHORITY");

                    }

                }

            }

        // Method to validate login credentials

        private boolean validateLogin(String username, String password) throws IOException  {

            try (BufferedReader reader = new BufferedReader(new FileReader("user\_data.txt"))) {

                String line;

                while ((line = reader.readLine()) != null) {

                    String[] parts = line.split(",");

                    String storedUsername = parts[0];

                    String storedPassword = parts[1];

                    if (username.equals(storedUsername) && password.equals(storedPassword)) {

                        return true;

                    }

                }

            } catch (IOException e) {

                System.err.println("Error reading user data: " + e.getMessage());

            }

            return false;

        }

        void createAccount() {

            message2="";

            message2= message;

            boolean accountCreated = false;

            while (!accountCreated) {

                String username = JOptionPane.showInputDialog(null, "Enter New Username: ");

                String password = JOptionPane.showInputDialog(null, "Enter New Password for '" + username + "' : ");

                if (username == null || password == null || username.isEmpty() || password.isEmpty()) {

                    JOptionPane.showMessageDialog(null, "Username and password cannot be empty. Please try again.");

                } else {

                    try (FileWriter writer = new FileWriter("user\_data.txt", true)) {

                        writer.write("\n" + username + "," + password);

                        message2+="\nAccount Successfully Created!";

                        message2+="\n";

                        message2+="\nUsername: "+(username);

                        message2+="\nPassword: "+(password);

                        accountCreated = true;

                    } catch (IOException e) {

                        System.err.println("Error appending data to file: " + e.getMessage());

                        JOptionPane.showMessageDialog(null, "An error occurred. Please try again.");

                    }

                    JOptionPane.showMessageDialog(null,message2);

                }

            }

        }

        void logout() {

            int choice = JOptionPane.showConfirmDialog(null, "Do you want to continue?", "Logout", JOptionPane.YES\_NO\_OPTION);

            if (choice == JOptionPane.NO\_OPTION) {

                JOptionPane.showMessageDialog(null,"Logged out successfully!");

                System.exit(0);

            }

        }

}

class Company extends Login{

    String h;

    String companyName;

    void chooseCompany(){

        message1="";

        String[] options ={"Sani Express","Adam Express","TransNasional","Adik Beradik","Company"};

        companyName= (String) JOptionPane.showInputDialog(null,"Select Bus Company","WELCOME!!",JOptionPane.INFORMATION\_MESSAGE,null,options,options[4]);

        String DestinationName= JOptionPane.showInputDialog(null,"Enter Destination");

        message1+="\nBus Company Name: "+(companyName);

        message1+= "\nDestination: "+(DestinationName);

        if (companyName==options[0]) {h= "seats,sani.txt";}

        else if (companyName==options[1]) {h= "seats,adam.txt";}

        else if (companyName==options[2]) {h= "seats,transN.txt";}

        else if (companyName==options[3]) {h= "seats,Adik.txt";}

    }

}

class ChooseTime extends Company{

    List<String> dates = new ArrayList<>();

    String selectedDate;

    void chooseOption(){

        String[] options = {"View Available Dates", "Choose Date for Booking"};

    int choice = JOptionPane.showOptionDialog(null, "⚠ Logged in as '"+(username)+"' \nChoose an option:", "WELCOME!!", JOptionPane.DEFAULT\_OPTION, JOptionPane.PLAIN\_MESSAGE, null, options, options[0]);

    switch (choice) {

        case 0:

            viewDates();

            chooseOption();

            break;

        case 1:

            chooseDate();

            deleteDate();

            break;

        default:

            JOptionPane.showMessageDialog(null,"No option selected.");

            break;

    }

    }

    void viewDates() {

        dates.clear(); // Clear the dates list before displaying them

        try (BufferedReader reader = new BufferedReader(new FileReader("dates.txt"))) {

            String line;

            while ((line = reader.readLine()) != null) {

                dates.add(line);

            }

        } catch (IOException e) {

            System.err.println("Error reading dates data: " + e.getMessage());

        }

        StringBuilder dateM = new StringBuilder("Available Dates:\n");

        for (String date : dates) {

            dateM.append(date).append("\n");

        }

        JOptionPane.showMessageDialog(null, dateM);

        dates.clear(); // Clear the dates list after displaying them

    }

    void chooseDate() {

        try (BufferedReader reader = new BufferedReader(new FileReader("dates.txt"))) {

            String line;

            while ((line = reader.readLine()) != null) {

                dates.add(line);

            }

        } catch (IOException e) {

            System.err.println("Error reading dates data: " + e.getMessage());

        }

        String[] datesArray = dates.toArray(new String[0]);

        selectedDate = (String) JOptionPane.showInputDialog(null, "Select date to Book:", "Book Date", JOptionPane.QUESTION\_MESSAGE, null, datesArray, datesArray[0]);

        message1+= "\nDate for booking: "+(selectedDate);

    }

    void deleteDate(){

        if (selectedDate != null) {

            dates.remove(selectedDate);

            try (PrintWriter writer = new PrintWriter(new FileWriter("dates.txt"))) {

                for (String date : dates) {

                    writer.println(date);

                }

            } catch (IOException e) {

                System.err.println("Error writing dates data: " + e.getMessage());

            }

        }

    }

    void confirm() {

        choice1 = JOptionPane.showConfirmDialog(null, "Current Company Selected: "+(companyName)+"\nCurrent Date Selected: "+(selectedDate)+"\nDo you want to Confirm your order?", "Confirm", JOptionPane.YES\_NO\_OPTION);

    if (choice1==JOptionPane.YES\_OPTION) {message=message+message1;

    }

    }

}

class Quantity extends ChooseTime {

    List<String> selectedSeatsList = new ArrayList<>();

    List<String> seats = new ArrayList<>();

    int quantitySeat;

    void chooseSeats() {

        boolean selectingSeats = true;

        try (BufferedReader reader = new BufferedReader(new FileReader(h))) {

            String line;

            while ((line = reader.readLine()) != null) {

                seats.add(line);

            }

        } catch (IOException e) {

            System.err.println("Error reading Seats data: " + e.getMessage());

        }

        String[] seatsArray = seats.toArray(new String[0]);

        while (selectingSeats) {

            String selectedSeat = (String) JOptionPane.showInputDialog(null, "1 Seat = 40 RM \nSelect Seat to Book:", "Book Seat", JOptionPane.QUESTION\_MESSAGE, null, seatsArray, seatsArray[0]);

            if (selectedSeat != null) {

                selectedSeatsList.add(selectedSeat);

                int choice = JOptionPane.showConfirmDialog(null, "Do you want to Book Another Seat?", "Confirmation", JOptionPane.YES\_NO\_OPTION);

                if (choice == JOptionPane.NO\_OPTION) {

                    selectingSeats = false;

                }

            } else {

                selectingSeats = false;

            }

        }

        for (String seat : selectedSeatsList) {

            message += "\nSeat for booking: " + seat;

        }

        quantitySeat = selectedSeatsList.size();

    }

    void deleteSeat() {

        for (String selectedSeat : selectedSeatsList) {

            seats.remove(selectedSeat);

        }

        selectedSeatsList.clear(); // Clear the list after deletion

        try (PrintWriter writer = new PrintWriter(new FileWriter(h))) {

            for (String seat : seats) {

                writer.println(seat);

            }

        } catch (IOException e) {

            System.err.println("Error writing seats data: " + e.getMessage());

        }

    }

}

class InsertData extends Quantity {

    void insertData(){

        String customerName= (String) JOptionPane.showInputDialog(null,"Enter Your Name");

        String customerPhoneNumber= JOptionPane.showInputDialog(null,"Enter Your PhoneNumber");

        message+="\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

        message+="\nCustomer Name: "+(customerName);

        message+="\nCustomer PhoneNumber: "+(customerPhoneNumber);

    }

}

class Payment extends InsertData {

    void payment(){

        String[] options ={"Credit Card","Online Banking","Cash","Debit Card","Payment"};

        String paymentName= (String) JOptionPane.showInputDialog(null,"Select Your Payment","WELCOME!!",JOptionPane.INFORMATION\_MESSAGE,null,options,options[4]);

        int Price= quantitySeat\*40;

        message+="\nPayment Option: "+(paymentName);

        message+= "\nTotal Price: "+(Price);

    }

}

class BookingSystem{

    public static void main(String[] args) throws IOException

    {

        Payment i= new Payment();

        PrintWriter out = new PrintWriter("Data.txt");

        boolean continueBooking = true;

        i.chooseLogin();

        while (continueBooking) {

            i.chooseCompany();

            i.logout();

            i.chooseOption();

            i.confirm();

            if (i.choice1 == JOptionPane.NO\_OPTION) {

                continueBooking = true;

            }

            else{continueBooking=false;}

        }

        i.chooseSeats();

        i.deleteSeat();

        i.insertData();

        i.payment();

        out.print(i.message);

        JOptionPane.showMessageDialog(null,i.message);

        out.close();

        System.exit(0);

    }

}

## 2. Additional TXT files

### A. seats,Adik.txt

A3

B4

C10

D2

B3

D10

D7

B9

D4

D1

C4

### B. seats,transN.txt

A3

B4

C10

D2

B3

D10

D7

B9

D4

D1

C4

### C. seats,adam.txt

B1

C1

C3

B3

D10

D5

D1

D7

B9

D4

C4

### D. seats,sani.txt

B3

C3

D2

C8

D9

B6

D10

D7

D1

B9

D4

C4

### E. dates.txt

05-May-2024

06-May-2024

07-May-2024

08-May-2024

09-May-2024

10-May-2024

11-May-2024

12-May-2024

13-May-2024

14-May-2024

16-May-2024

17-May-2024

18-May-2024

19-May-2024

20-May-2024

21-May-2024

22-May-2024

23-May-2024

24-May-2024

25-May-2024

26-May-2024

27-May-2024

28-May-2024

29-May-2024

31-May-2024

### F. user\_data.txt

Raysul@admin,raysul@@

SyadCH@admin,syad@@

guest,guest

Nuri@admin,nuri@@

Adam@admin,adam@@