



JAVA PROJECT

RAILWAY RESERVATION SYSTEM

By-

Hari Om Topal

189301189

CSE-A

Manipal University Jaipur

Overview

This is a simple railway reservation program in JAVA where you can book, cancel and view tickets. You can also search for particular passenger number, check status of reserved tickets etc.

Functions

1. Login/Register
2. Book Ticket
3. Cancel Ticket
4. Display Reservation Chart
5. Search Passenger Number
6. Display unbooked tickets
7. Display Full Train Table
8. View Transaction History

Features

- **Object Oriented**

The program has been divided into a number of different functions ,which helps increase the reusability of the code.

- **File Handling**

File Handling has been used extensively to store and manage data even when the program is terminated.

- **Secure**

Private access specifier has been used wherever necessary to provide greater protection against data misuse.

- **Exception Handling**

Multiple Exceptions have been handled in order to allow the program to function smoothly without any interruptions.

- Inheritance

The program has been divided into separate classes with each class providing separate functionality.

- Easy to Edit

The use of file handling makes it easier to add/remove/edit the Train Schedules by directly making changes to the TrainTable.txt without the requirement of understanding the source code

Source Code

OutputHandler.java

```
import java.io.*;

class OutputHandler {
    void writeToFile(String s1, String s2) {
        try {

            File f1 = new File("ID.txt");
            FileWriter fw = new FileWriter(f1, true);
            BufferedWriter bw = new BufferedWriter(fw);
            bw.newLine();
            bw.write(s1 + " " + s2);
            bw.close();
            fw.close();
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

IDCheck.java

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

class IDCheck extends OutputHandler {
    String userID, password;
```

```

int ID() {
    try {
        Scanner sc = new Scanner(System.in);
        File f1 = new File("ID.txt");
        BufferedReader br = new BufferedReader(new FileReader(f1));
        System.out.println("Enter USER ID and PASSWORD");
        String s;
        userID = sc.next();
        password = sc.next();
        int match = 2;
        while ((s = br.readLine()) != null) {
            String[] words = s.split(" ");
            for (int i = 0; i < 2; i++) {
                if (words[0].equals(userID)) {
                    if (words[1].equals(password)) {
                        match = 0;
                    } else {
                        match = 1;
                    }
                }
            }
        }
        if (match == 0)
            System.out.println("ID MATCH FOUND, Logging in ...");
        else if (match == 1)
            System.out.println("Either USER ID of PASSWORD is
incorrect ");
        else if (match == 2)
            System.out.println("NOT FOUND");

        return (match);
    } catch (IOException e) {

        System.out.println(e);
    }
}

```

```
        return (2);
    }

    RegisterUser() {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USER ID and PASSWORD");
        userID = sc.next();
        password = sc.next();
        writeToFile(userID, password);
        System.out.println("ACCOUNT CREATED SUCCESSFULLY ");

    }

    void writeToFile(String s) {
        try {

            File file = new File(userID + ".txt");
            FileWriter fw = new FileWriter(file, true);
            BufferedWriter bw = new BufferedWriter(fw);
            bw.newLine();
            bw.write(s);
            bw.close();
            fw.close();
        } catch (IOException e) {
            System.out.println(e);
        }
    }
}
```

RailReserve.java

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

public class RailReserve extends IDcheck {
    Date d1 = new Date();
    InputStreamReader isr = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(isr);
    int pno[] = new int[275];
    String name[] = new String[275];
    String phno[] = new String[275];
    int age[] = new int[275];
    double amount;
    int pcount = 0;
    int pnum = 1;
    int linecount = 0;
    int row = 0;
    int col = 0;
    String[][] TrainTable;
    int trainid;
    int trno;

    RailReserve() throws Exception {
        initialiseTable();
    }

    private void printTable(int i) {
        System.out.printf("%-20s %-20s %-20s %-20s %-20s %-20s %-20s %-20s\n",
            TrainTable[i][0], TrainTable[i][1],
            TrainTable[i][2], TrainTable[i][3], TrainTable[i][4],
            TrainTable[i][5], TrainTable[i][6],
            TrainTable[i][7], TrainTable[i][8]);
        System.out.println();
    }
}
```

```

private void initialiseTable() throws Exception {
    linecount = 0;
    FileInputStream fstream = new FileInputStream("TrainTable.txt");
    BufferedReader br = new BufferedReader(new
InputStreamReader(fstream));
    String s;
    int i = 0;
    int j = 0;
    while ((s = br.readLine()) != null) {
        linecount++;
    }
    fstream.getChannel().position(0);
    br = new BufferedReader(new InputStreamReader(fstream));
    col = 9;
    row = linecount;
    TrainTable = new String[row][col];

    while ((s = br.readLine()) != null) {
        String[] line = s.trim().split("\t");
        for (j = 0; j < 9; j++) {
            TrainTable[i][j] = line[j];
        }
        i++;
    }
}

public void doMenu() throws Exception {
    int cho = 0;
    do {
        System.out.println("\f");
        doHeading();
        System.out.println("1.Book Ticket");
        System.out.println("2.Cancel Ticket");
        System.out.println("3.Search Passenger");
        System.out.println("4.Reservation Chart");
    }
}

```



```
System.out.println("5.Display Unbooked Tickets");
System.out.println("6.Display Full Train Table");
System.out.println("7.View Transaction History");
System.out.println("8.Exit");
System.out.println("Please enter your choice");
cho = Integer.parseInt(br.readLine());
switch (cho) {
case 1:
    doBook();
    break;
case 2:
    doCancel();
    break;
case 3:
    doSearch();
    break;
case 4:
    doDispList();
    break;
case 5:
    doDispUnbooked();
    break;
case 6:
    doDispList();
    break;
case 7:
    doHistory();
    break;
case 8:
    doExit();
    break;
default:
    System.out.println("Invalid choice");
}
} while (cho != 8);
}
```

```

        private void doHeading() throws Exception {

System.out.println("#####
###");

        System.out.println("*****Railway Reservation
System*****");

System.out.println("#####
###");

        }

        private void showTrain(String src, String dest) {
            for (int i = 0; i < row; i++) {
                if (i == 0) {
                    printTable(i);
                }
                if (TrainTable[i][2].equalsIgnoreCase(src) &&
TrainTable[i][3].equalsIgnoreCase(dest)) {
                    printTable(i);
                }
            }
            System.out.println();
        }

        private void doBook() throws Exception {
            System.out.println("Please enter the SOURCE & DESTINATION");
            String src = br.readLine();
            String dest = br.readLine();
            showTrain(src, dest);
            System.out.println("Enter Train No.");
            trno = Integer.parseInt(br.readLine());
            trainid = trno;
            System.out.println("Please enter no. of tickets");
            int t = Integer.parseInt(br.readLine());
            int ticketAvailable = 0;
            if (Integer.parseInt(TrainTable[trno][7]) >= t) {
                ticketAvailable = 1;
            }
        }
    }
}

```

```

    }
    if (ticketAvailable == 1) {
        for (int i = 0; i < t; i++) {
            pno[pcount] = pnum;
            System.out.println("Please enter your name");
            name[pcount] = br.readLine();
            System.out.println("Please enter your age");
            age[pcount] = Integer.parseInt(br.readLine());
            System.out.println("Please enter your phno");
            phno[pcount] = br.readLine();
            pcount++;
            pnum++;
            System.out.println("Ticket successfully booked");
            String booked = TrainTable[trno][6];
            String unbooked = TrainTable[trno][7];
            int b = Integer.parseInt(booked);
            int u = Integer.parseInt(unbooked);
            b = b + 1;
            u = u - 1;
            TrainTable[trno][6] = Integer.toString(b);
            TrainTable[trno][7] = Integer.toString(u);
        }
        amount = Integer.parseInt(TrainTable[trno][8]);
        printBill();
        displayBill();
        System.out.println("\nPlease Pay Rs. " + t * amount);
    }
    if (ticketAvailable == 0) {
        System.out.println("The TRAIN is fully booked .");
    }
}

private void doCancel() throws Exception {
    int t_pno[] = new int[275];
    String t_name[] = new String[275];
    String t_phno[] = new String[275];

```

```

int t_age[] = new int[275];
int t_cl[] = new int[275];
int t_pcount = 0;
int passengerFound = 0;
System.out.println("Please enter your PNR no.");
int p = Integer.parseInt(br.readLine());
for (int i = 0; i < pcount; i++) {
    if (pno[i] != p) {
        t_pno[t_pcount] = pno[i];
        t_name[t_pcount] = name[i];
        t_phno[t_pcount] = phno[i];
        t_age[t_pcount] = age[i];
        t_pcount++;
    } else {
        passengerFound = 1;
        System.out.println("Please collect refund of Rs." + 0.70 *
Integer.parseInt(TrainTable[trno][8]));
        lString booked = TrainTable[trno][6];
        String unbooked = TrainTable[trno][7];
        int b = Integer.parseInt(booked);
        int u = Integer.parseInt(unbooked);
        b = b - 1;
        u = u + 1;
        TrainTable[trno][6] = Integer.toString(b);
        TrainTable[trno][7] = Integer.toString(u);
        writeToFile("Refund Initiated towards PNR No. - " + p);
        writeToFile("Refund Amount = " + 0.70 * amount);
    }
}
if (passengerFound == 1) {
    pno = t_pno;
    name = t_name;
    age = t_age;
    phno = t_phno;
    pcount = t_pcount;
    System.out.println("Ticket Successfully Cancelled");
}

```

```

        writeToFile("Ticket Successfully Cancelled");
    } else if (passengerFound == 0) {
        System.out.println("PASSENGER Not Found");
    }
}

private void doDispList() throws Exception {
    System.out.println(
        "*****\tTRAIN
TABLE\t*****");
    for (int i = 0; i < linecount; i++) {
        printTable(i);
    }
}

private void doSearch() throws Exception {
    int passengerFound = 0;
    System.out.println("Please enter passenger no. to search");
    int p = Integer.parseInt(br.readLine());
    for (int i = 0; i < pcount; i++) {
        if (pno[i] == p) {
            System.out.println("Details Found-");
            passengerFound = 1;
            System.out.println("PNR NO - " + pno[i]);
            System.out.println("NAME - " + name[i]);
            System.out.println("Ph. No. - " + phno[i]);
            System.out.println("AGE - " + age[i]);
        }
    }
    if (passengerFound == 0)
        System.out.println("No such passenger");
}

private void doDispUnbooked() throws Exception {
    System.out.println("No. of Booked & Unbooked Tickets-");
    for (int i = 0; i < linecount; i++) {

```

```

        System.out.printf(" %-20s %-20s %-20s %-20s",
TrainTable[i][0], TrainTable[i][1], TrainTable[i][6],
        TrainTable[i][7]);
        System.out.println();

    }

}

private void doExit() throws Exception {
    System.out.println("*****");
    System.out.println("MADE BY - \nName : Hari Om Topal\nClass :
CSE-A\nRoll No. : 36");
    System.out.println("*****");
    updateTable();
}

private void updateTable() throws Exception {
    FileOutputStream fstream = new FileOutputStream("TrainTable.txt");
    BufferedWriter br = new BufferedWriter(new
OutputStreamWriter(fstream));
    for (int i = 0; i < linecount; i++) {
        for (int j = 0; j < 9; j++) {

            br.write(TrainTable[i][j] + "\t");
        }
        br.newLine();
    }
    br.close();
}

private void doLogin() throws Exception {
    int reset = 1;
    while (reset == 1) {
        System.out.println("SELECT \n1.Registered User \n2.New User");
        int choice = Integer.parseInt(br.readLine());
    }
}

```

```

        if (choice == 1) {
            if (ID() == 0) {
                reset = 0;
                doMenu();
            } else {
                System.out.println("press 1 to retry , 0 to
continue");
                reset = Integer.parseInt(br.readLine());
            }
        } else if (choice == 2) {
            RegisterUser();
            doLogin();
        } else {
            System.out.println("Invalid Choice \n");
            doLogin();
        }
    }
}

private void printBill() {
    Date d1 = new Date();
    writeToFile("*****");
    String s = ("\nLogged In AT " + d1);
    writeToFile(s);
    int i = 0;
    while (pno[i] != 0) {
        writeToFile("PNR NO. - " + Integer.toString(pno[i]));
        writeToFile("NAME - " + name[i]);
        writeToFile("AGE - " + Integer.toString(age[i]));
        writeToFile("PHONE NO - " + phno[i]);
        s = ("Train No - " + trainid + " 1- " +
(TrainTable[trainid][1]) + " FROM " + (TrainTable[trainid][2])
            + " TO " + (TrainTable[trainid][3]));
        writeToFile(s);
        writeToFile("Amount = Rs. " + amount);
        I++;
    }
}

```

```

    }
}

private void displayBill() {
    System.out.println("*****");
    System.out.println("Bill Generated On " + d1);
    int i = 0;
    while (pno[i] != 0) {
        System.out.println("PNR NO. - " + Integer.toString(pno[i]));
        System.out.println("NAME - " + name[i]);
        System.out.println("AGE - " + Integer.toString(age[i]));
        System.out.println("PHONE NO - " + phno[i]);
        System.out.println("Train No - " + trainid + " - " +
            (TrainTable[trainid][1]) + " FROM "
                + (TrainTable[trainid][2]) + " TO " +
            (TrainTable[trainid][3]));
        System.out.println("Amount = Rs. " + amount);
        i++;
    }
}

private void doHistory() throws Exception {
    File f1 = new File(userID + ".txt");
    BufferedReader br = new BufferedReader(new FileReader(f1));
    String s;
    while ((s = br.readLine()) != null) {
        System.out.println(s);
    }
    br.close();
}

public static void main(String args[]) {
    try {
        RailReserve obj = new RailReserve();
        obj.doLogin();
    }
}

```



```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Output

```
PS D:\TUTS\OOPJ LAB\Railway Reservation System> cd "d:\TUTS\OOPJ LAB\Railway Reservation System\" ; if ($?) { javac RailReserve.java } ; if ($?) {
```

```

1.Registered User
2.New User
2
Enter USER ID and PASSWORD
ABCD
1234
ACCOUNT CREATED SUCCESSFULLY
SELECT
1.Registered User
2.New User
1
Enter USER ID and PASSWORD
ABC
123
NOT FOUND
press 1 to retry , 0 to continue
1
SELECT
1.Registered User
2.New User
1
Enter USER ID and PASSWORD
ABCD
1234
ID MATCH FOUND, Logging in ...

```

```

#####
*****Railway Reservation System*****
#####
1.Book Ticket
2.Cancel Ticket
3.Search Passenger
4.Reservation Chart
5.Display Unbooked Tickets
6.Display Full Train Table
7.View Transaction History
8.Exit
Please enter your choice
█

```

8.Exit
Please enter your choice

6

TRAIN TABLE								
TRAIN NO.	TRAIN NAME	SOURCE	DESTINATION	TIME OF DEPARTURE	TIME OF ARRIVAL	BOOKED	UNBOOKED	FARE
1	Orient Express	Ahmedabad	Delhi	12:00	20:00	2	48	499
2	Chennai Express	Chennai	Delhi	05:00	15:00	1	74	699
3	Shatabdi	Pune	Bangalore	07:00	20:00	1	29	399

*****Railway Reservation System*****
#####

- 1.Book Ticket
- 2.Cancel Ticket
- 3.Search Passenger
- 4.Reservation Chart
- 5.Display Unbooked Tickets
- 6.Display Full Train Table
- 7.View Transaction History

8.Exit
Please enter your choice

1

Please enter the SOURCE & DESTINATION
ahmedabad

TRAIN NO.	TRAIN NAME	SOURCE	DESTINATION	TIME OF DEPARTURE	TIME OF ARRIVAL	BOOKED	UNBOOKED	FARE
1	Orient Express	Ahmedabad	Delhi	12:00	20:00	2	48	499

Enter Train No.

1

Please enter no. of tickets

2

Please enter your name

Hari Om

Please enter your age

20

Please enter your phno

9887984512

Ticket successfully booked

Please enter your name

HARI 2

Please enter your age

21

Please enter your phno

21
 Please enter your phno
 8754321586
 Ticket successfully booked

 Bill Generated On Sun Nov 17 16:19:28 IST 2019
 PNR NO. - 1
 NAME - Hari Om
 AGE - 20
 PHONE NO - 9887984512
 Train No - 1 - Orient Express FROM Ahmedabad TO Delhi
 Amount = Rs. 499.0
 PNR NO. - 2
 NAME - HARI 2
 AGE - 21
 PHONE NO - 8754321586
 Train No - 1 - Orient Express FROM Ahmedabad TO Delhi
 Amount = Rs. 499.0
 Please Pay Rs. 998.0

 *****Railway Reservation System*****
 #####
 1.Book Ticket
 2.Cancel Ticket
 3.Search Passenger
 4.Reservation Chart
 5.Display Unbooked Tickets
 6.Display Full Train Table
 7.View Transaction History
 8.Exit
 Please enter your choice

4

			TRAIN TABLE			*****		
TRAIN NO.	TRAIN NAME	SOURCE	DESTINATION	TIME OF DEPARTURE	TIME OF ARRIVAL	BOOKED	UNBOOKED	FARE
1	Orient Express	Ahmedabad	Delhi	12:00	20:00	4	46	499
2	Chennai Express	Chennai	Delhi	05:00	15:00	1	74	699
3	Shatabdi	Pune	Bangalore	07:00	20:00	1	29	399

 *****Railway Reservation System*****
 #####

Ph. No. - 8754321586
AGE - 21

```
#####
*****Railway Reservation System*****
#####
1.Book Ticket
2.Cancel Ticket
3.Search Passenger
4.Reservation Chart
5.Display Unbooked Tickets
6.Display Full Train Table
7.View Transaction History
8.Exit
Please enter your choice
2
Please enter your PNR no.
2
Please collect refund of Rs.349.29999999999995
Ticket Successfully Cancelled
```

```
#####
*****Railway Reservation System*****
#####
1.Book Ticket
2.Cancel Ticket
3.Search Passenger
4.Reservation Chart
5.Display Unbooked Tickets
6.Display Full Train Table
7.View Transaction History
8.Exit
Please enter your choice
5
No. of Booked & Unbooked Tickets-
```

TRAIN NO.	TRAIN NAME	BOOKED	UNBOOKED
1	Orient Express	3	47
2	Chennai Express	1	74
3	Shatabdi	1	29

```
#####
*****Railway Reservation System*****
```

```
#####
*****Railway Reservation System*****
#####
```

- 1.Book Ticket
- 2.Cancel Ticket
- 3.Search Passenger
- 4.Reservation Chart
- 5.Display Unbooked Tickets
- 6.Display Full Train Table
- 7.View Transaction History
- 8.Exit

Please enter your choice

5

No. of Booked & Unbooked Tickets-

TRAIN NO.	TRAIN NAME	BOOKED	UNBOOKED
1	Orient Express	4	46
2	Chennai Express	1	74
3	Shatabdi	1	29

```
#####
*****Railway Reservation System*****
#####
```

- 1.Book Ticket
- 2.Cancel Ticket
- 3.Search Passenger
- 4.Reservation Chart
- 5.Display Unbooked Tickets
- 6.Display Full Train Table
- 7.View Transaction History
- 8.Exit

Please enter your choice

3

Please enter passenger no. to search

2

Details Found-

PNR NO - 2

NAME - HARI 2

Ph. No. - 8754321586

AGE - 21

```
#####
```

3

Shatabdi

1

29

```
#####
*****Railway Reservation System*****
#####
1.Book Ticket
2.Cancel Ticket
3.Search Passenger
4.Reservation Chart
5.Display Unbooked Tickets
6.Display Full Train Table
7.View Transaction History
8.Exit
Please enter your choice
Please enter your choice
7
```

```
*****
```

```
Logged In AT Sun Nov 17 16:21:28 IST 2019
PNR NO. - 1
NAME - Hari Om
AGE - 20
PHONE NO - 9887984512
Train No - 1 l- Orient Express FROM Ahmedabad TO Delhi
Amount = Rs. 499.0
PNR NO. - 2
NAME - HARI 2
AGE - 21
PHONE NO - 8754321586
Train No - 1 l- Orient Express FROM Ahmedabad TO Delhi
Train No - 1 l- Orient Express FROM Ahmedabad TO Delhi
Amount = Rs. 499.0
Refund Initiated towards PNR No. - 2
Refund Amount = 349.29999999999995
Refund Amount = 349.29999999999995
Refund Amount = 349.29999999999995
Ticket Successfully Cancelled
```

```
#####
*****Railway Reservation System*****
#####
1.Book Ticket
```

```
#####
*****Railway Reservation System*****
#####
1.Book Ticket
2.Cancel Ticket
3.Search Passenger
4.Reservation Chart
5.Display Unbooked Tickets
6.Display Full Train Table
7.View Transaction History
8.Exit
Please enter your choice
8
*****
MADE BY -
Name : Hari Om Topal
Class : CSE-A
Roll No. : 36
*****
```

Conclusion

This project deals with reserving tickets for the trains which is heck these days. So in order to make it convenient to travel we can book, cancel, display reservation chart, search passenger number and display unbooked tickets. This makes the travel smooth and convenient. This also helps in checking the details afterwards.

Limitations

- Does not provide the functionality to make changes to the Train Table from directly within the program
- Does not have SuperUser/Admin access , it works strictly for ticket booking and checking schedules
- Does not have the functionality to search passengers by name.
- Does not have functionality to see the transaction history for a particular range of time.