LOVELY PROFESSIONAL UNIVERSITY

# TRANSPORT ENQUIRY SYSTEM USING JAVA

A Project

Submitted in partial fulfilment of the

Requirement for the award of the

Degree Of

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRICAL AND ELECTRONIC & COMMUNICATION ENGINEERING**

By

Abhishek Kumar Mourya
Manish Kumar Sah & Shivam  Verma

**UNDER THE GUIDANCE OF**

Surbhi Ma'am

# CERTIFICATE

This is to certify that Dissertation titled  Multi-Face Detection using machine learning that is being submitted by Abhishek Kumar Mourya, Manish Kumar Sah & Shivam  Verma is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bona-fide work done under my guidance. The contents of this dissertation, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Surbhi Ma'am
Assistant Professor
School of Electronics and Electrical Engineering
Lovely Professional University, Phagwara Punjab.

# ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude towards my guide, Kriti Rawal, who gave his full support in the working of this dissertation work with his stimulating suggestions and encouragement all the time. She has always been a source of ideas.

I am very thankful to my guide Surbhi for her readiness to show me the right path. Above her highly intelligent and tactful mind is her warm and inspiring heart, which has always inspired and given a thrust to do the work as a master no matter however small the task may be. Her inspiration has made me accomplish my task with ease.

I am thankful to management of Lovely Professional University for giving me an opportunity to carry out studies at the university and providing the proper infrastructure like internet facility which is pool of vast knowledge to work in.

At last but not the least my gratitude towards my parents, I would like to thank God for the strength that keep me standing and for the hope that keep me believing that this report would be possible.

# DECLARATION

I Abhishek Kumar Mourya, Manish Kumar Sah & Shivam Verma students of B.Tech (EEE) under the Department of Electronics and Electronic Engineering of Lovely Professional University, Punjab, hereby declare that all the information furnished in this dissertation report is based on my intensive research and is genuine. This dissertation is done to the best of my knowledge and does not contain any part of my work that has been submitted for the award of my degree either of this university or any other university without proper citation.

Abhishek Kumar Mourya : 12015994
Manish Kumar Sah : 12013598
Shivam Verma : 12015055

## ABSTRACT

Now this is the age of speed. Everything happens in the speed of supersonic. The data can be transferred at the speed of light in the digital medium, can travel in the supersonic speed, hence three is a need of information inflow in the same speed. Here is one such need of information fast enough. We have experienced in waiting to a transport terminals for transport controllers to get the information about the transport facility. We encounter so many times there will be no person for providing these information which significantly wastes the time just to know whether there is any facility or not. Here is one solution for such a problem which lessens the human intervention in providing such information in the transport terminals.

# INDEX

Project Contents:

# CHAPTER 1:
# INTRODUCTION

The Transport Enquiry System is a real-time train information system used in the transportation industry. This system allows passengers to access information about train schedules, delays, and cancellations in real-time. Additionally, it helps transportation authorities to track train movements and ensure the safety and efficiency of train operations.

The Nation Train Enquiry System (NTES) is an example of a Transport Enquiry System that uses data analytics to provide real-time information about train schedules and delays [3]. NTES is widely used in India and has become a crucial tool for passengers and transportation authorities alike.

The development of such a system would require the integration of various technologies such as data analytics, artificial intelligence, and machine learning. The system can be designed to provide a seamless experience for passengers by integrating with mobile applications and digital signage systems.
Transportation authorities can also use the system to monitor and optimize transportation operations.

Fig 1 Transport Enquiry System

Overall, the development of an automated Transport Enquiry System can greatly benefit the transportation industry by improving efficiency and enhancing the passenger experience.

### 1.1.2. The Categories of Transport Enquiry System

The categories of a Transport Enquiry System might include:

1. Transportation modes: This category includes information about different modes of transportation, such as buses, trains, planes, and taxis.

2. Routes: This category includes information about different transportation routes, including their starting and ending points, stops along the way, and the duration of the trip.

3. Schedules: This category includes information about the timing of transportation services, such as the departure and arrival times of buses and trains.

4. Fares: This category includes information about the costs of transportation services, including ticket prices, discounts, and any applicable fees.

5. Booking and reservations: This category includes information about how to book and reserve transportation services, including online booking, phone reservations, and in-person booking options.

6. Real-time information: This category includes information about the current status of transportation services, such as delays, cancellations, and any other unexpected changes.

7. Customer support: This category includes information about how to contact customer support for any issues or questions related to transportation services.

These categories may vary depending on the specific Transport Enquiry System and its intended use.

## 1.1.3. Inheritance in Java

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The idea behind inheritance in Java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and felds of the parent class. Moreover, you can add new methods and felds in your current class also.

Inheritance represents the IS-A relationship which is also known as a parent-child relationship.

*Why use inheritance in java*
o For Method Overriding (so runtime polymorphism can be achieved).
o For Code Reusability.

*Terms used in Inheritance*

- Class: A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.
- Sub Class/Child Class: Subclass is a class which inherits the other class. It is also called a derived class, extended class, or child class.
- Super Class/Parent Class: Superclass is the class from where a subclass inherits the features. It is also called a base class or a parent class.
- Reusability: As the name specifes, reusability is a mechanism which facilitates you to reuse the felds and methods of the existing class when you create a new class. You can use the same felds and methods already defned in the previous class.

## *1.1.4. File Handling*

File handling can be an important aspect of implementing a Transport Enquiry System, as it can help to store and manage the real-time information that is collected and used by the system.

For example, the system may collect and store information about transportation schedules, delays, cancellations, and other important details related to the movement of people and goods. This information can be stored in a database or file system, which can then be accessed by the system to provide real-time updates to passengers and transportation authorities.

In addition to storing real-time information, file handling can also be used to store historical data about transportation operations. This data can be used to analyze trends, identify areas for improvement, and optimize transportation operations.

Overall, file handling is an important aspect of implementing a Transport Enquiry System, as it can help to ensure that real-time information is stored and managed effectively. By leveraging file handling techniques, transportation authorities can improve the efficiency and effectiveness of their operations, leading to a better travel experience for passengers.

### 1.1.5. EXCEPTION HANDLING

Exception handling in the context of the Transport Enquiry System in Java refers to the process of handling unexpected or exceptional events that may occur during the execution of the program.

Java has built-in support for exception handling through the try-catch-finally block. When an exception is thrown within a try block, the catch block is executed, allowing the program to gracefully handle the exception and prevent crashes. The finally block is then executed, allowing the program to clean up resources and finalize operations.

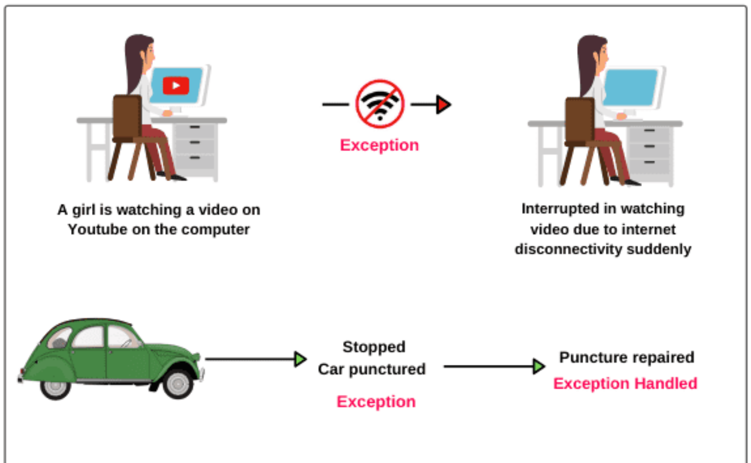

A girl is watching a video on Youtube on the computer
Exception
Interrupted in watching video due to internet disconnectivity suddenly
Stopped
Car punctured
Exception
Puncture repaired
Exception Handled

Fig: Realtime Example of Exception Handling

Specifically, in the context of the Transport Enquiry System, exception handling can help to prevent the system from crashing in the event of unexpected errors, such as invalid user input or server connectivity issues. By implementing proper exception handling techniques, the system can continue to function reliably and provide accurate information to users.

It is important to note that while exception handling can prevent program crashes, it does not necessarily resolve the underlying issues causing the exceptions. Developers should also aim to address the root causes of exceptions to ensure the long-term reliability and performance of the Transport Enquiry System.

### 1.1.6. HASHMAP AND HASHTABLE

HashMap and Hashtable store key and value pairs in a hash table. When using a Hashtable or HashMap, we specify an object that is used as a key and the value that you want to be linked to that key. The key is then hashed, and the resulting hash code is used as the index at which the value is stored within the table. Now let us discuss with the help of an example.

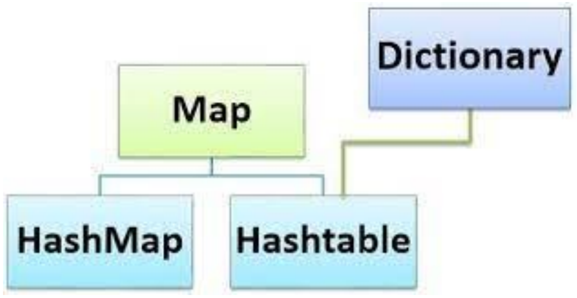


Figure 5 hash map and hash table

### HASHMAP VS HASHTABLE

- HashMap is non-synchronized. It is not thread-safe and can't be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.

- HashMap allows one null key and multiple null values whereas Hashtable doesn't allow any null key or value.

# CHAPTER 2. EXPERIMENTAL RESULTS

*Code implementation-*

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


class Admin {
```

```
private String username;
private String password;

public Admin(String username, String password) {
    this.username = username;
    this.password = password;
}

// getters and setters for the fields

public boolean authenticate(String username, String password) {
    return this.username.equals(username) && this.password.equals(password);
}

public void addData(Map<Integer, String[]> transport,int id) {
    Scanner input=new Scanner(System.in);
    String s1=input.nextLine();
    String s2=input.nextLine();
    String s3=input.nextLine();
    transport.put(id, new String[] {s1, s2, s3});
}
public void editData(Map<Integer, String[]> transport,int id,int choiceedit) {
    System.out.println("edit");

    Scanner input=new Scanner(System.in);

    if(choiceedit==1){
        System.out.println("ENTER NEW DATA TO THE TRANSPORT NO="+id);
        String k1;
        k1=input.nextLine();
        String s2=input.nextLine();
        String s3=input.nextLine();

        String[] currentValue = transport.get(id);
        if (currentValue != null) {
            transport.put(id, new String[] {k1, s2, s3});
        }
        System.out.println("VALUES UPDATED SUCCESSFULLY");
    }
    else if(choiceedit==2){
        String[] value=transport.get(id);
        System.out.println("ENTER NEW ID=");
        int key=input.nextInt();
        transport.put(key,value);
        transport.remove(id);
        System.out.println("ID UPDATED SUCCESSFULLY");

    }

}

public void deleteData(Map<Integer, String[]> transport,int id) {
    transport.remove(id);
    System.out.println("REMOVED THE ELEMENT");

}
```

```
}


// Base class for all users
class User {
    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }
}

// Class for registered users
class RegisteredUser extends User {
    public static final HashMap<String, String[]> users = new HashMap<>();

    static {
        try {
            BufferedReader reader = new BufferedReader(new FileReader("user.txt"));
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 2) {
                    String key = parts[0];
                    String[] value = new String[]{parts[1]};
                    users.put(key, value);
                }
            }
            reader.close();
            System.out.println("Dictionary read from file");
        } catch (IOException e) {
            System.out.println("Error reading file");
        }
    }

    public RegisteredUser(String username, String password) {
        super(username, password);
    }

    public static boolean authenticate(String username, String password) {
        // Check if the user exists in the dictionary and the password is correct
        return users.containsKey(username) && users.get(username)[0].equals(password);
    }

    public static void addUser(String username, String password) {
        if (users.containsKey(username)) {
            System.out.println("Username already exists");
            return;
        }
        users.put(username, new String[]{password});
        try (BufferedWriter bw = new BufferedWriter(new FileWriter("user.txt", true))) {
            bw.write(username + "," + password);
            bw.newLine();
            System.out.println("New user created");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

}

// Class for new users
class NewUser extends User {
    public NewUser(String username, String password) {
        super(username, password);
        RegisteredUser.addUser(username, password); // Add the new user to the dictionary
    }
}


interface displayfunc{
    void showdata( Map<Integer, String[]> transport );
    void userdisplay(Map<Integer, String[]> transport,int key);
}

class display implements displayfunc{
    public  void showdata( Map<Integer, String[]> transport ){
        for (Map.Entry<Integer, String[]> entry : transport.entrySet()) {
            int key = entry.getKey();
            String[] value = entry.getValue();
            System.out.println("TRANSPORT ID: " + key + "   DETAILS: " + Arrays.toString(value));
        }
    }
    public void userdisplay(Map<Integer, String[]> transport,int key){
        if (transport.containsKey(key)) {
            String[] value = transport.get(key);
            System.out.println("DETAILS:"+Arrays.toString(value));
        } else {
            System.out.println("Key not found in map.");
        }

    }

}


public class finalproject {
    public static void readdata(Map<Integer, String[]> transport, String filename) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader(filename));
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length >= 4) {
                    Integer key = Integer.parseInt(parts[0]);
                    String[] value = new String[]{parts[1], parts[2], parts[3]};
                    transport.put(key, value);
                }
            }
            reader.close();
            System.out.println("Dictionary read from file");

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

    public static void userreaddata(Map<String, String[]> userdetail) {
        try {
            BufferedReader reader = new BufferedReader(new FileReader("user.txt"));
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 2) {
                    String key = parts[0];
                    String[] value = new String[]{parts[1]};
                    userdetail.put(key, value);
                }
            }
            reader.close();
            System.out.println("Dictionary read from file");
        } catch (IOException e) {
            System.out.println("Error reading file");
        }
    }

    public static void userdeleteData(Map<String, String[]> userdetail, String id) {
        userdetail.remove(id);
        System.out.println("REMOVED THE ELEMENT");
    }

    public static void userappenddata(Map<String,String[]> userdetail){
        try (BufferedWriter bw = new BufferedWriter(new FileWriter("user.txt", true))) {
            for (String key : userdetail.keySet()) {
                String[] values = userdetail.get(key);
                String line = key + "," + values[0];

                // check if the line already exists in the file
                boolean lineExists = false;
                try (java.util.Scanner scanner = new java.util.Scanner(new java.io.File("user.txt"))) {
                    while (scanner.hasNextLine()) {
                        String existingLine = scanner.nextLine();
                        if (existingLine.equals(line)) {
                            lineExists = true;
                            break;
                        }
                    }
                }

                // write the line to the file if it doesn't already exist
                if (!lineExists) {
                    bw.write(line);
                    bw.newLine();
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    public static void appenddata(Map<Integer,String[]> transport,String filename){
        try (BufferedWriter bw = new BufferedWriter(new FileWriter(filename, true))) {
            for (Integer key : transport.keySet()) {
                String[] values = transport.get(key);
                String line = key + "," + values[0] + "," + values[1]+ "," + values[2];

                // check if the line already exists in the file
                boolean lineExists = false;
                try (java.util.Scanner scanner = new java.util.Scanner(new java.io.File(filename))) {
                    while (scanner.hasNextLine()) {
                        String existingLine = scanner.nextLine();
                        if (existingLine.equals(line)) {
                            lineExists = true;
                            break;
                        }
                    }
                }

                // write the line to the file if it doesn't already exist
                if (!lineExists) {
                    bw.write(line);
                    bw.newLine();
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    public static void cleardata(String filename){
        try {
            FileOutputStream fos = new FileOutputStream(filename);
            fos.write(new byte[0]);
            fos.close();
         } catch (IOException e) {
            System.out.println("An error occurred while clearing the file: " + e.getMessage());
         }
    }


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

        Map<Integer, String[]> train = new Hashtable<>();
        Map<Integer, String[]> bus = new Hashtable<>();
```

```
        Map<Integer, String[]> airplane = new Hashtable<>();

        readdata(train,"train.txt");
        readdata(bus,"bus.txt");
        readdata(airplane,"airplane.txt");

        Map<String, String[]> userdetail = new Hashtable<>();
        userreaddata(userdetail);

        display d=new display();

        boolean choiceadimnuser=true;
        while(choiceadimnuser){
           System.out.println("=================ENTER YOUR CHOICE===============\n 1
TO LOGIN AS ADMIN \n 2 TO LOGIN AS USER
\n=========================================== ");

           int opt;
           opt=input.nextInt();
           input.nextLine();
           if(opt==1){
              Admin a1=new Admin("admin","abc123");
              System.out.println("===========<<ENTER ADMIN USER NAME AND
PASSWARD:>>===========");
              String uname=input.nextLine();
              String passward=input.nextLine();
              boolean loginpass=(a1.authenticate(uname, passward));
              if(loginpass==true){

                 boolean conti=true;
                 while(conti){

                    System.out.println("==============<<SELECT>>=============\n 1: TO ADD
DATA \n 2: TO EDIT DATA \n 3: TO DELETE DATA \n 4: TO DELETE USER \n OR ANY OTHER
KEY TO EXIT \n======================================");

                    int choice=input.nextInt();
                    input.nextLine();
                    if(choice==1){
                       System.out.println("==================<<ADDING
DATA>>===================");
                       System.out.println("==============<<SELECT>>=============\n 1: FOR
TRAIN \n 2: FOR BUS \n 3: FOR AIRPORT \n======================================");
                       int select=input.nextInt();
                       if(select==1){
                          System.out.println("========<<ENTER TRAIN_NUMBER, NAME, DATE
AND TIME>>=========");
                          int id=input.nextInt();
                          a1.addData(train,id);
                          appenddata(train,"train.txt");

                       }
                       else if(select==2){
                          System.out.println("========<<ENTER BUS_NUMBER, NAME, DATE AND
TIME>>=========");
                          int id=input.nextInt();
                          a1.addData(bus,id);
                    appenddata(bus,"bus.txt");
                }
                else if(select==3){
                    System.out.println("========<<ENTER AIRPLANE_NUMBER, NAME, DATE
AND TIME>>=========");
                    int id=input.nextInt();
                    a1.addData(airplane,id);
                    appenddata(airplane,"airplane.txt");
                }
                else{
                    System.out.println("Wrong choice");
                }

            }
            else if(choice==2){
                System.out.println("==================<<EDITING
DATA>>===================");
                System.out.println("==============<<SELECT>>=============\n 1: FOR
TRAIN \n 2: FOR BUS \n 3: FOR AIRPORT \n=====================================");
                int select=input.nextInt();
                if(select==1){
                    System.out.println("==================<<ENTER ID
NUMBER>>===============");
                    int id=input.nextInt();
                    System.out.println("============<<ENTER>>=========\n 1: IF YOU
WANT TO EDIT THE DETAILS AND \n 2: IF YOU WANT TO EDIT THE
ID\n=========================");
                    int choiceedit=input.nextInt();
                    if (train.containsKey(id)) {
                        a1.editData(train,id,choiceedit);
                      } else {
                        System.out.println(id + " is train number is not present in data base");
                      }
                    cleardata("train.txt");

                   appenddata(train,"train.txt");
                }
                else if(select==2){
                    System.out.println("==================<<ENTER ID
NUMBER>>===============");
                    int id=input.nextInt();
                    System.out.println("============<<ENTER>>=========\n 1: IF YOU
WANT TO EDIT THE DETAILS AND \n 2: IF YOU WANT TO EDIT THE
ID\n=========================");
                    int choiceedit=input.nextInt();
                    if (bus.containsKey(id)) {
                        a1.editData(bus,id,choiceedit);
                      } else {
                        System.out.println(id + " is bus number is not present in data base");
                      }
                    cleardata("bus.txt");
                     appenddata(bus,"bus.txt");
                }
                else if(select==3){
                    System.out.println("==================<<ENTER ID
NUMBER>>===============");
                    int id=input.nextInt();
                System.out.println("============<<ENTER>>=========\n 1: IF YOU
WANT TO EDIT THE DETAILS AND \n 2: IF YOU WANT TO EDIT THE
ID\n=========================");
                int choiceedit=input.nextInt();
                if (airplane.containsKey(id)) {
                  a1.editData(airplane,id,choiceedit);
                 } else {
                  System.out.println(id + " is airplane number is not present in data base");
                 }
                 cleardata("airplane.txt");
                appenddata(airplane,"airplane.txt");
              }
              else{
                System.out.println("Wrong choice");
              }
            }
            else if(choice==3){
              System.out.println("==================<<DELETING
DATA>>===================");
              System.out.println("==============<<SELECT>>=============\n 1: FOR
TRAIN \n 2: FOR BUS \n 3: FOR AIRPORT \n=====================================");
              int select=input.nextInt();
              if(select==1){
                System.out.println("========<<ENTER TRAIN_NUMBER>>=========");
                int id=input.nextInt();
                a1.deleteData(train,id);
                 cleardata("train.txt");
                appenddata(train,"train.txt");

              }
              else if(select==2){
                System.out.println("========<<ENTER BUS_NUMBER>>=========");
                int id=input.nextInt();
                a1.deleteData(bus,id);
                cleardata("bus.txt");
                appenddata(bus,"bus.txt");
              }
              else if(select==3){
                System.out.println("========<<ENTER
AIRPLANE_NUMBER>>=========");
                int id=input.nextInt();
                a1.deleteData(airplane,id);
                cleardata("airplane.txt");
                appenddata(airplane,"airplane.txt");
              }
              else{
                System.out.println("WRONG CHOICE");
              }
            }
            else if(choice==4){
              System.out.println("=======ENTER THE USER ID======");
              String userid=input.nextLine();
              userdeleteData(userdetail, userid);
              cleardata("user.txt");
              userappenddata(userdetail);
            }
            else{
                    conti=false;

                  }
               }
            }

            else{
               System.out.println("========<<WRONG USER NAME AND
PASSWARD>>=========");
            }
            if(loginpass==true){
               System.out.println("===<<SELECT 1,2,3 YOU WANT IF YOU WANT TO SEE THE
RECORDS OF TRAIN BUS AIRPLANE RESPECTIVELY:>>==");
               int s=input.nextInt();
               if (s==1){
                  System.out.println("===========<<ALL TRAIN DETAILS>============");
                  d.showdata(train);
               }
               else if(s==2){
                  System.out.println("============<<ALL BUS DETAILS>>=============");
                  d.showdata(bus);
               }
               else if(s==3){
                  System.out.println("============<<ALL BUS DETAILS>>=============");
                  d.showdata(airplane);
               }
               else{
                  System.out.println("===========<<WRONG CHOICE>>=============");
               }

            }
         }
         else if(opt==2){
            boolean continuser=true;
            while(continuser){
               System.out.println("==========ENTER 1 TO LOGIN USER ACCOUNT AND 2 TO
CREATE ACCOUNT ===========");
               int c = input.nextInt();
               input.nextLine();
               if(c==1){
                  System.out.println("=====ENTER USER NAME AND PASSWORD======");
                  String username = input.nextLine();
                  String password = input.nextLine();
                  if (RegisteredUser.authenticate(username,password)) {
                     System.out.println("Authentication successful");
                     System.out.println("===<<SELECT 1,2,3 YOU WANT IF YOU WANT TO SEE
THE RECORDS OF TRAIN BUS AIRPLANE RESPECTIVELY:>>== ");
                     int cho=input.nextInt();
                     input.nextLine();
                     if(cho==1){
                        System.out.println("=======ENTER THE TRANSPORT ID=======");
                        int tid=input.nextInt();
                        d.userdisplay(train,tid);
                     }
                     else if(cho==2){
                        System.out.println("=======ENTER THE TRANSPORT ID=======");
                        int tid=input.nextInt();
                        d.userdisplay(bus,tid);
                    }
                    else if(cho==3){
                        System.out.println("=======ENTER THE TRANSPORT ID=======");
                        int tid=input.nextInt();
                        d.userdisplay(airplane,tid);
                    }
                    else{
                        System.out.println("===========WRONG CHOICE=============");
                    }

                } else {
                    System.out.println("Authentication failed");
                }

            }
            else if(c==2){
                System.out.println("=====ENTER USER NAME AND PASSWORD======");
                String newUsername = input.nextLine();
                String newPassword = input.nextLine();
                NewUser newUser = new NewUser(newUsername, newPassword);
            }
            else{
                System.out.println("=====WRONG CHOICE=====");
            }
            System.out.println("======ENTER 1 IF YOU WANT TO EXIT USER
BLOCK======");
            int a=input.nextInt();
            if(a==1){
                continuser=false;
            }
        }

    }
    else{
        System.out.println("===========<<WRONG CHOICE>>============= ");
    }
    System.out.println("=========WOULD YOU LIKE TO EXIT OR
CONTINUE==========\n #####ENTER 1 IF YOU WANT TO EXIT.");
    int s=input.nextInt();
    if(s==1){
        choiceadimnuser=false;
    }

  }


 }

}
```

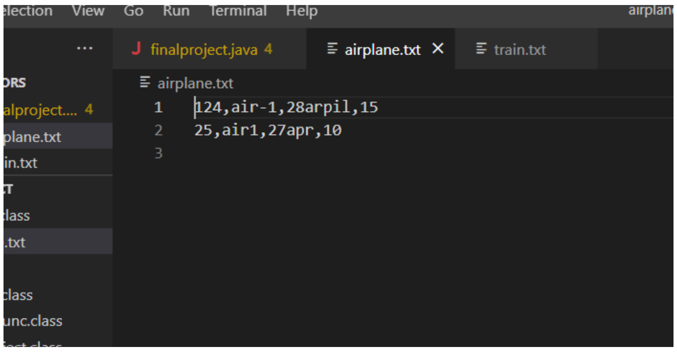
## ➢ *OUTPUT FOR ADMIN LOG IN AND TASKS*

```
Dictionary read from file
=================ENTER YOUR CHOICE===============
 1 TO LOGIN AS ADMIN
 2 TO LOGIN AS USER
==========================================
1
===========<<ENTER ADMIN USER NAME AND PASSWARD:>>===========
admin
abc123
==============<<SELECT>>=============
 1: TO ADD DATA
 2: TO EDIT DATA
 3: TO DELETE DATA
 4: TO DELETE USER
 OR ANY OTHER KEY TO EXIT
=====================================
1
==================<<ADDING DATA>>===================
==============<<SELECT>>=============
 1: FOR TRAIN
 2: FOR BUS
 3: FOR AIRPORT
=====================================
3
========<<ENTER AIRPLANE_NUMBER, NAME, DATE AND TIME>>=========
124
air-1
```

```
27april
16
==============<<SELECT>>=============
 1: TO ADD DATA
 2: TO EDIT DATA
 3: TO DELETE DATA
 4: TO DELETE USER
 OR ANY OTHER KEY TO EXIT
=====================================
2
==================<<EDITING DATA>>===================
==============<<SELECT>>=============
 1: FOR TRAIN
 2: FOR BUS
 3: FOR AIRPORT
=====================================
3
==================<<ENTER ID NUMBER>>===============
124
============<<ENTER>>=========
 1: IF YOU WANT TO EDIT THE DETAILS AND
 2: IF YOU WANT TO EDIT THE ID
==========================
1
edit
ENTER NEW DATA TO THE TRANSPORT NO=124
air-1
28arpil
15
```

```
VALUES UPDATED SUCCESSFULLY
==============<<SELECT>>=============
 1: TO ADD DATA
 2: TO EDIT DATA
 3: TO DELETE DATA
 4: TO DELETE USER
 OR ANY OTHER KEY TO EXIT
=====================================
```

- ***File updated successfully added new train number & details***



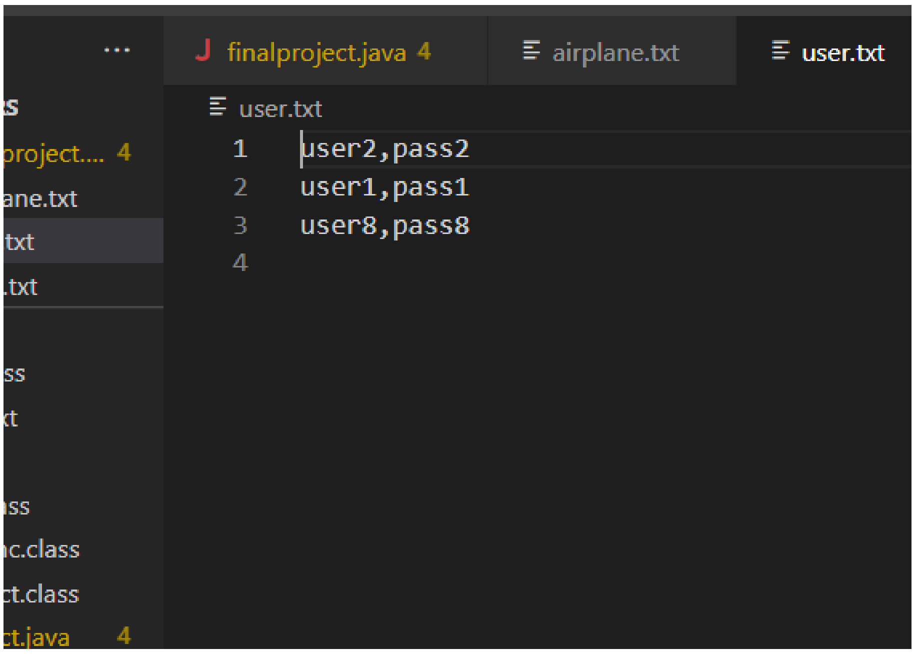
```
124,air-1,28arpil,15
25,air1,27apr,10
```

- ***OUTPUT FOR USER LOGIN AND USAGES***

```
=================ENTER YOUR CHOICE===============
 1 TO LOGIN AS ADMIN
 2 TO LOGIN AS USER
==========================================
2
==========ENTER 1 TO LOGIN USER ACCOUNT AND 2 TO CREATE ACCOUNT ===========
1
=====ENTER USER NAME AND PASSWORD======
user1
pass1
Dictionary read from file
Authentication successful
===<<SELECT 1,2,3 YOU WANT IF YOU WANT TO SEE THE RECORDS OF TRAIN BUS
AIRPLANE RESPECTIVELY:>>==
1
=======ENTER THE TRANSPORT ID=======
65
DETAILS:[train-A, 28april, 15]
======ENTER 1 IF YOU WANT TO EXIT USER BLOCK======
2
```

```
==========ENTER 1 TO LOGIN USER ACCOUNT AND 2 TO CREATE ACCOUNT ===========
2
=====ENTER USER NAME AND PASSWORD======
user8
pass8
New user created
======ENTER 1 IF YOU WANT TO EXIT USER BLOCK======
2
==========ENTER 1 TO LOGIN USER ACCOUNT AND 2 TO CREATE ACCOUNT ===========
1
=====ENTER USER NAME AND PASSWORD======
user8
pass8
Authentication successful
===<<SELECT 1,2,3 YOU WANT IF YOU WANT TO SEE THE RECORDS OF TRAIN BUS
AIRPLANE RESPECTIVELY:>>==
1
=======ENTER THE TRANSPORT ID=======
65
DETAILS:[train-A, 28april, 15]
======ENTER 1 IF YOU WANT TO EXIT USER BLOCK======
```

- ***File updated successfully added new userid and passward***



```
user.txt
user2,pass2
user1,pass1
user8,pass8
```

# CHAPTER 3.

# LEARNING OUTCOME

The Transport Enquiry System can have several learning outcomes, both for passengers and transportation authorities. Here are a few potential examples:
For passengers:

- Improved travel planning: With real-time information about transportation schedules, delays, and cancellations, passengers can better plan their travel and avoid unnecessary waiting or delays.
- Enhanced safety: By providing up-to-date information about transportation operations, the Transport Enquiry System can help passengers make informed decisions about their safety, such as avoiding crowded trains or buses.
- Better travel experience: By providing accurate and reliable information about transportation operations, the Transport Enquiry System can help improve the overall travel experience for passengers, leading to increased satisfaction and loyalty.

For transportation authorities:

- Improved operations: By collecting and analyzing real-time information about transportation operations, the Transport Enquiry System can help transportation authorities optimize their operations, leading to improved efficiency and reduced costs.
- Enhanced safety and security: By monitoring transportation operations in real-time, the Transport Enquiry System can help transportation authorities identify potential safety or security risks and take appropriate action to mitigate them.
- Improved decision-making: By providing transportation authorities with accurate and timely information about transportation operations, the Transport Enquiry System can help support better decision-making, leading to more effective transportation policies and strategies.

Overall, the Transport Enquiry System can have several learning outcomes, both for passengers and transportation authorities. By providing real-time information about transportation operations, the system can help improve the efficiency, safety, and overall experience of transportation for all stakeholders involved.

# CHAPTER 4.

# CONCLUSION &FUTURE SCOPE

The design and implementation details of a simple Voice-Based Public Transport Enquiry System is presented in this paper. Unlike other SMS based systems and call enquiry based systems, in which the user needs to send SMS in predefined formats or he needs to call a number to get the information about his/her travel, this system is very simple to use and more efficient. Moreover, there is no requirement of human power like in the enquiry desks. New stations and busses can be added easily and the details are readily available to the users of the system. It doesn't require much cost to establish the system and, as no SMS gateway is required and no persons are required round the clock for running the system.

A big organization like Government running public transport service can provide toll-free numbers to the user and make the user happy with free calls to the system.

**However, there is still room for improvement and future development of the Transport Enquiry System. Here are a few potential areas for future scope:**

- Integration with other transportation systems: The Transport Enquiry System could be integrated with other transportation systems, such as traffic management systems or smart city infrastructure, to provide a more comprehensive view of transportation operations.
- Use of advanced analytics and AI: By leveraging advanced analytics and artificial intelligence techniques, the Transport Enquiry System could provide more accurate and personalized information to passengers, as well as help transportation authorities optimize their operations and reduce costs.
- Expansion to new transportation modes: The Transport Enquiry System could be expanded to include new transportation modes, such as ride-sharing or micro-mobility services, to provide a more comprehensive view of transportation options to passengers.

## REFERENCES

- www.javatpoint.com.
- www.w3schools.com.
- http://www.apachefriends.org
- https://opus.govst.edu/cgi/viewcontent.cgi?article=1167&context=capstones
- https://getliner.com/en/picked-by-liner/reader-mode?url=https%3A%2F%2Fwww.altexsoft.com%2Fblog%2Ftravel%2Fairport-technology-management-operations-software-solutions-and-vendors%2F
- https://ijcrt.org/papers/IJCRT2005317.pdf

***Thank you***