GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

RAILWAY RESERVATION SYSTEM

A Report for the Project 1

Submitted by

SUNNY CHAUDHARY
(1713104020)

in partial fulfilment for the award of the degree
of

BACHELOR OF COMPUTER APPLICATION

IN

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the Supervision of

MS. KAMAKSHI GUPTA ASSISTANT PROFESSOR;

MAY- 2020

GALGOTIAS UNIVERSITY

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "**RAILWAY RESERVATION SYSTEM"** is The Bonafde work of **"SUNNY CHAUDHARY(1713104020)"**who carried Out the project work under my supervision**.**

SIGNATURE OF HEAD	**SIGNATURE OF SUPERVISOR**
Dr. MUNISH SHABARWAL,	MS,KAMAKSHI GUPTA.,
PhD (Management), PhD (CS)	Assistant Professor
Professor & Dean,	**School of Computing Science &**
School of Computing Science &	**Engineering**
Engineering	

TABLE OF CONTENTS

1. ABSTRACT

Automatic processing plays vital role for bulk amount of data and even that generating correct results and displaying correct information is one of the challenging task in this technological era. Among various organization our Indian railway network also deals and process bulk data every day. To make the processing task easier and saving correct information into the file system will help this organization to grow more and more and supports in our Indian economy growth. Among various tasks performed within this organization reservation of tickets is also a common task which takes place every day, so to manage this task efficiently, a new computer oriented system is required. In order to maintain privacy and security for each customers, each customers will have a valid login id and password to access their account and using their particular, they will able to make their reservation. Proper payment has been implemented and provided within the system, to make secure payment and keep their payment resources secured. Using the concepts of object oriented programming, this project has been divided into different modules which is to be carried out by classes and methods and final touch given by the objects which are allowed to access their variables and methods of a particular class.

Existing system not having feature of displaying appropriate messages for events and errors which occurs while accessing the system. Proper validation and session has not been set up by which it system not able to identify and differentiate users, so cannot be used appropriately for multi user environment. This system do not able to provide the details of customers transactions and their payment mode along with date, time and processing charges details. Making changes

in information and perform administrative task was not possible while using existing system.

2. INTRODUCTION

2.1 OVERALL DESCRYPTION

In this emerging world of computers, almost all-manual system has switched to automated and computerized system. Therefore, we are developing the software for “Railway Reservation System” to model the present system and to remove the drawbacks of the present system. This project explores how computer technology can be used to solve the problem of user.

This being a big step in terms of improvement in the railway system it is widely accepted across the country. Rather than designing manually, we have made use of computer. Use of computer has solved many problems, which are faced during manual calculation. Once data are fed, it can perform accurate functions. Therefore, to reduce the complexity and efficiency a versatile and an outsourcing railway reservation system has been developed

2.2 PURPOSE -A feasibility study is undertaken to determine the possibility of either improving the existing system or developing a completely new system. This study helps to obtain an overview of the problem and to get rough assessment of whether feasible solutions exist. Since the feasibility study may lead to the commitment of large resources,

2.3 MOTIVATION AND SCOPE

The customers are required to register on the server for getting access to the database and query result retrieval. Upon registration, each user has an account that is essentially the ‘view level’ for the customer. The account contains comprehensive information of the user entered during registration and permits the customer to get access to his/her past reservations, enquire about travel fare and availability of seats, make fresh reservations, and update his account details. Each passenger is allotted a unique PNR no. through which one can access his/her account.

The railway administrator is another member involved in the transactions. The administrator is required to login using a master password, once authenticated as an administrator, one has access and right of modification to all the information stored in the database. This includes the account information of the customers, attributes and statistics of stations, description of the train stoppages and physical description of coaches, all the reservations that have been made. The railway administrator has the right to modify any information stored at the server database.

3. PROPOSED MODEL

While using with this new system, passengers will be provided with two options at this project home page and these two are: - Register and Login. If candidates do not have their account, they can create a new account and after completion allowed to access this system to carry out the task of making their reservation. If existing passengers have their account, they can access it by using their passenger id and password. Passengers will be allowed to search between source and destination and check availability of seats on particular date. Once the particular train will be selected, they can fill the information form along with total number of passengers and select appropriate payment source to make their final payment for making final confirmation of their reservation. Typically, a requirements analyst generates functional requirements after building use cases. However, this may have exceptions since software development is an iterative process and sometime certain requirements are conceived prior to definitin ofthe use case. Both artifacts (use cases documents and requiremets documents) complement each other in a bidirectional process. typcal functional requirement will contain a unique name and number, brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, arequirement through the development of the system

3.1 SYSTEM REQUIRMENT

A Software Requirement Specification (SRS) is a requirements specification for a software system that is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements. Non-functional requirements are requirements that impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

The initial specifications of user requirements may be based on interviews with the database users and on the designers own analysis of the enterprise. The basic issues that the SRS writer(s) shall address are the following:

- Functionality:

  What is the software supposed to do?

- External interfaces.

  How does the software interact with people, the system's hardware, other hardware, and other software?

- Performance.

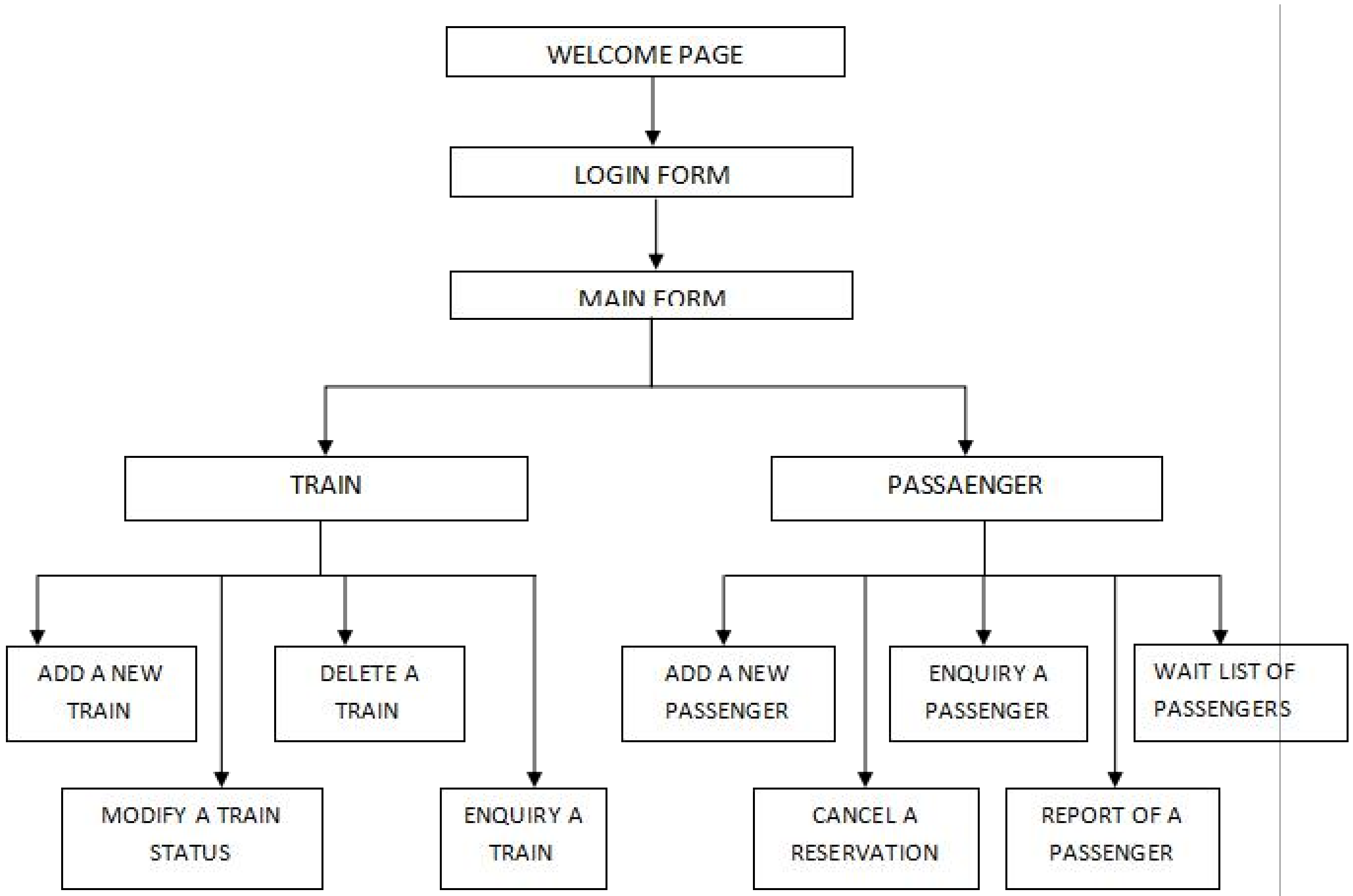
  What is the speed, availability, response time, recovery time symbol

4. EXISTING SYSTEM

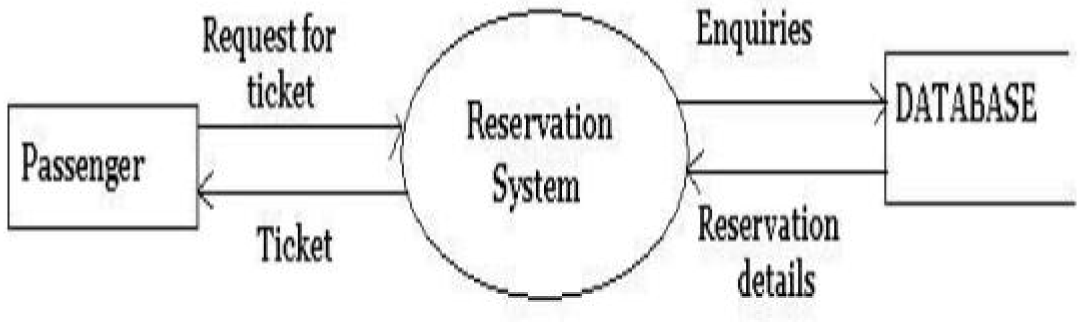
Existing system not having feature of displaying appropriate messages for events and errors which occurs while accessing the system. Proper validation and session has not been set up by which it system not able to identify and differentiate users, so cannot be used appropriately for multi user environment. This system do not able to provide the details of customers transactions and their payment mode along with date, time and processing charges details. Making changes in information and perform administrative task was not possible while using existing system. It does not able to provide availability details to the passengers and system can only be used, when passengers know the train details along with their processing and charges details.

5. DIAGARMS

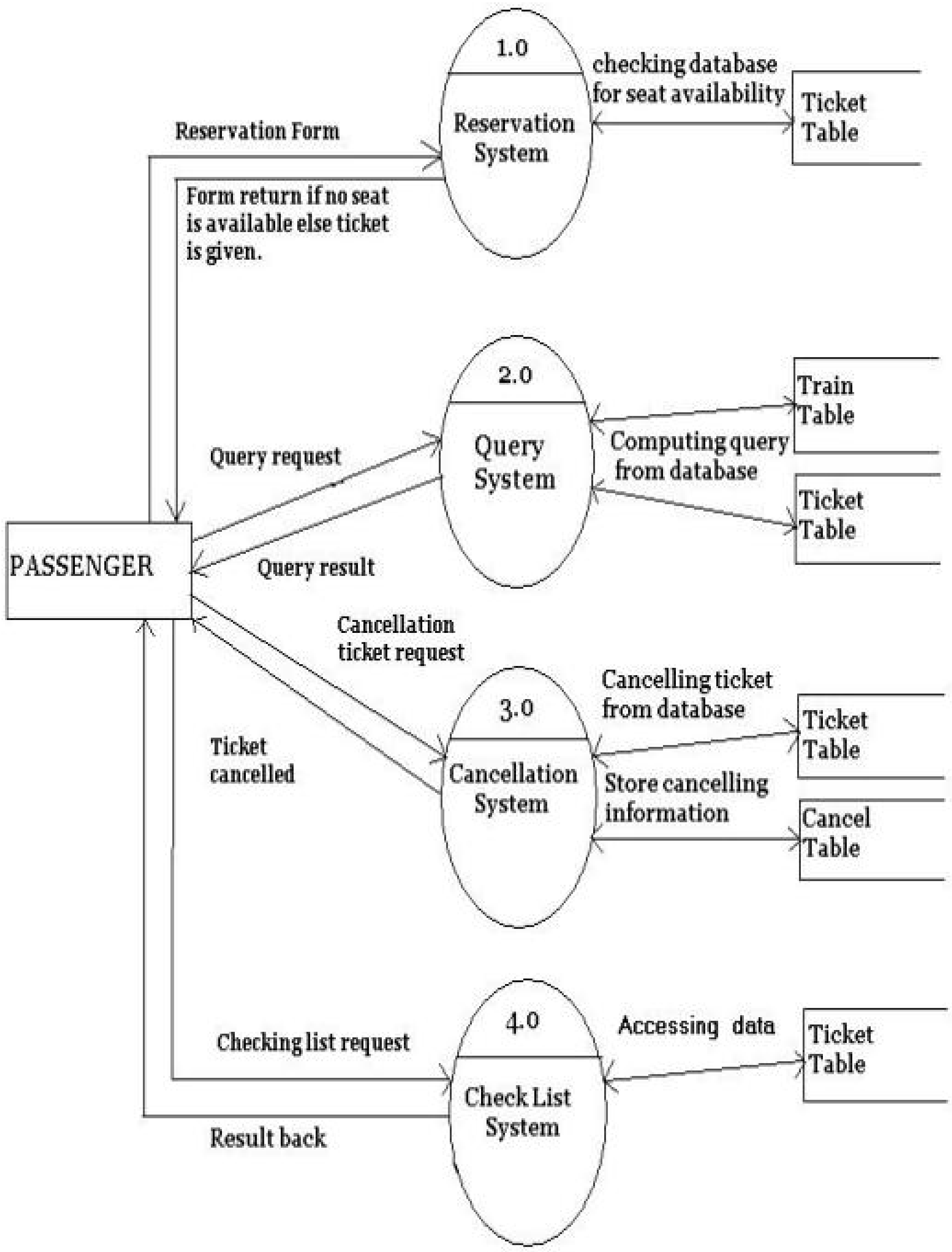
5.1 ER-Diagram

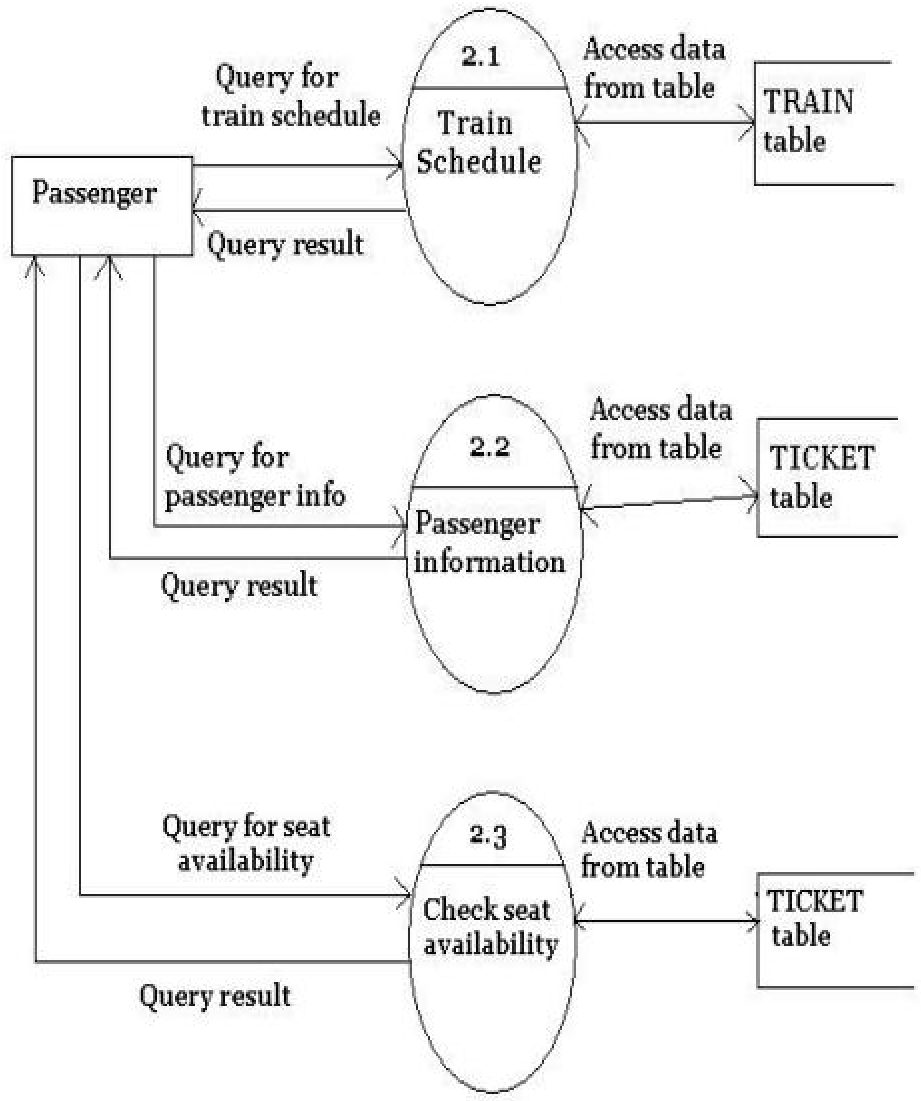


5.2 DATA FLOW DIAGRAM

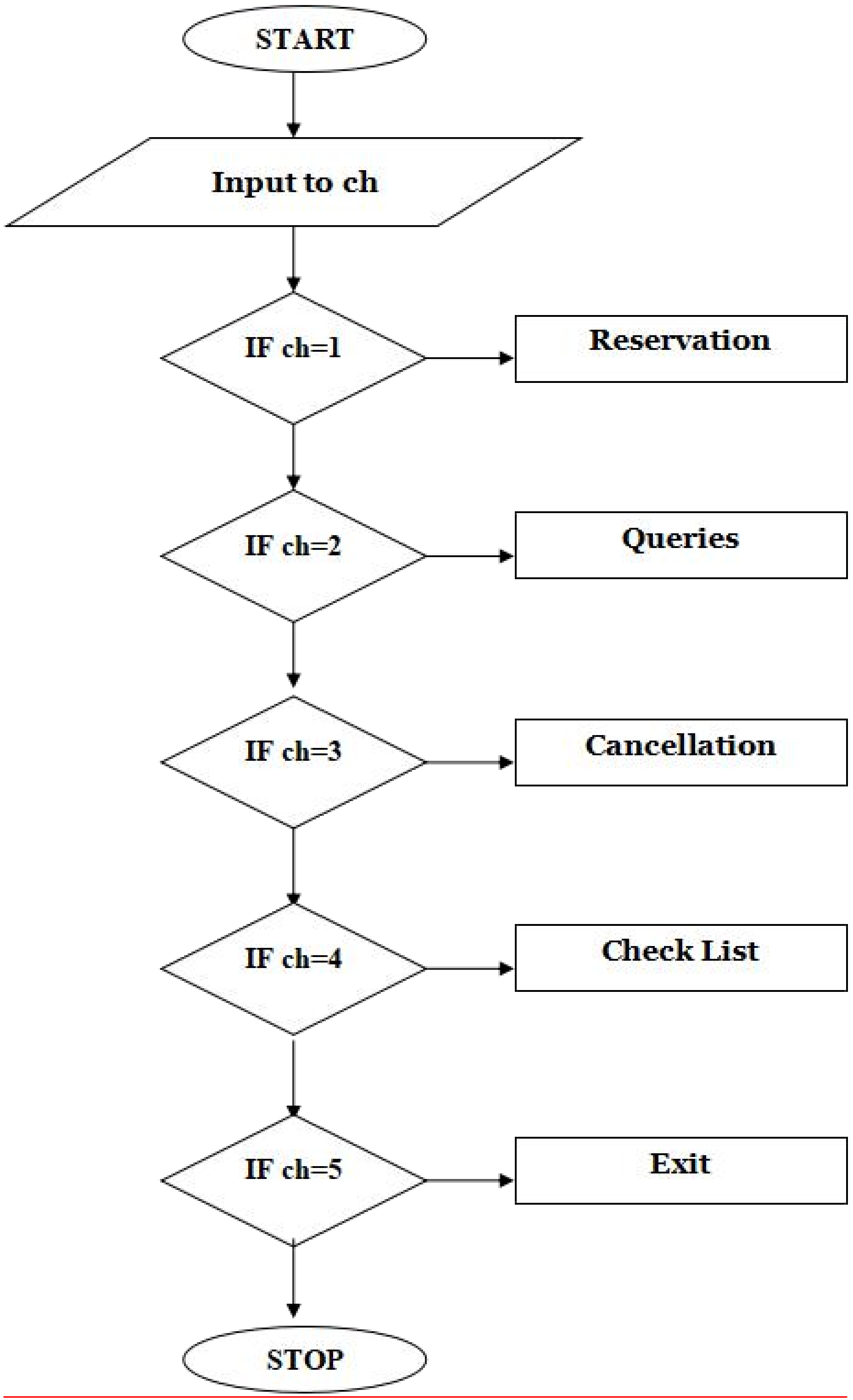


ZERO LEVEL -DFD

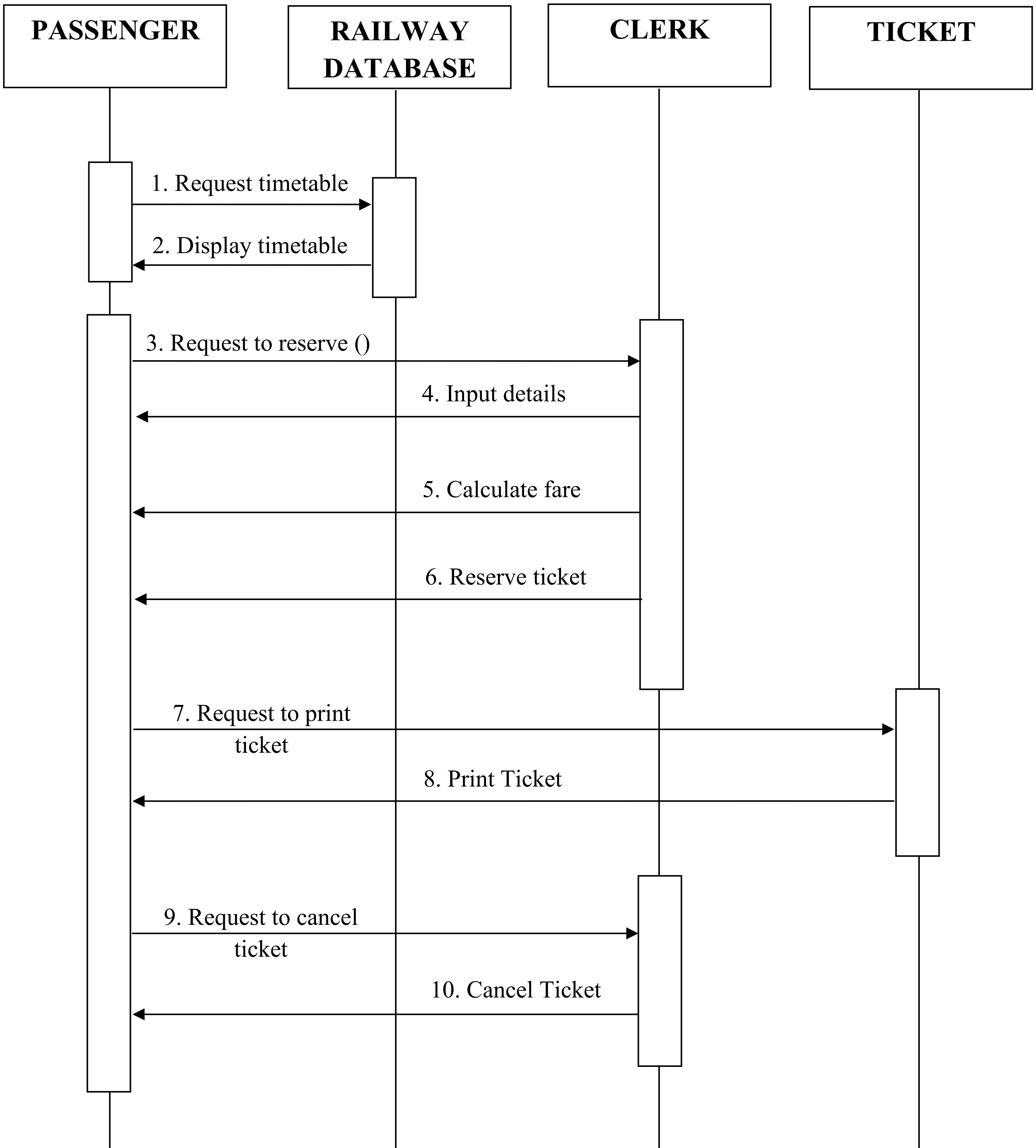


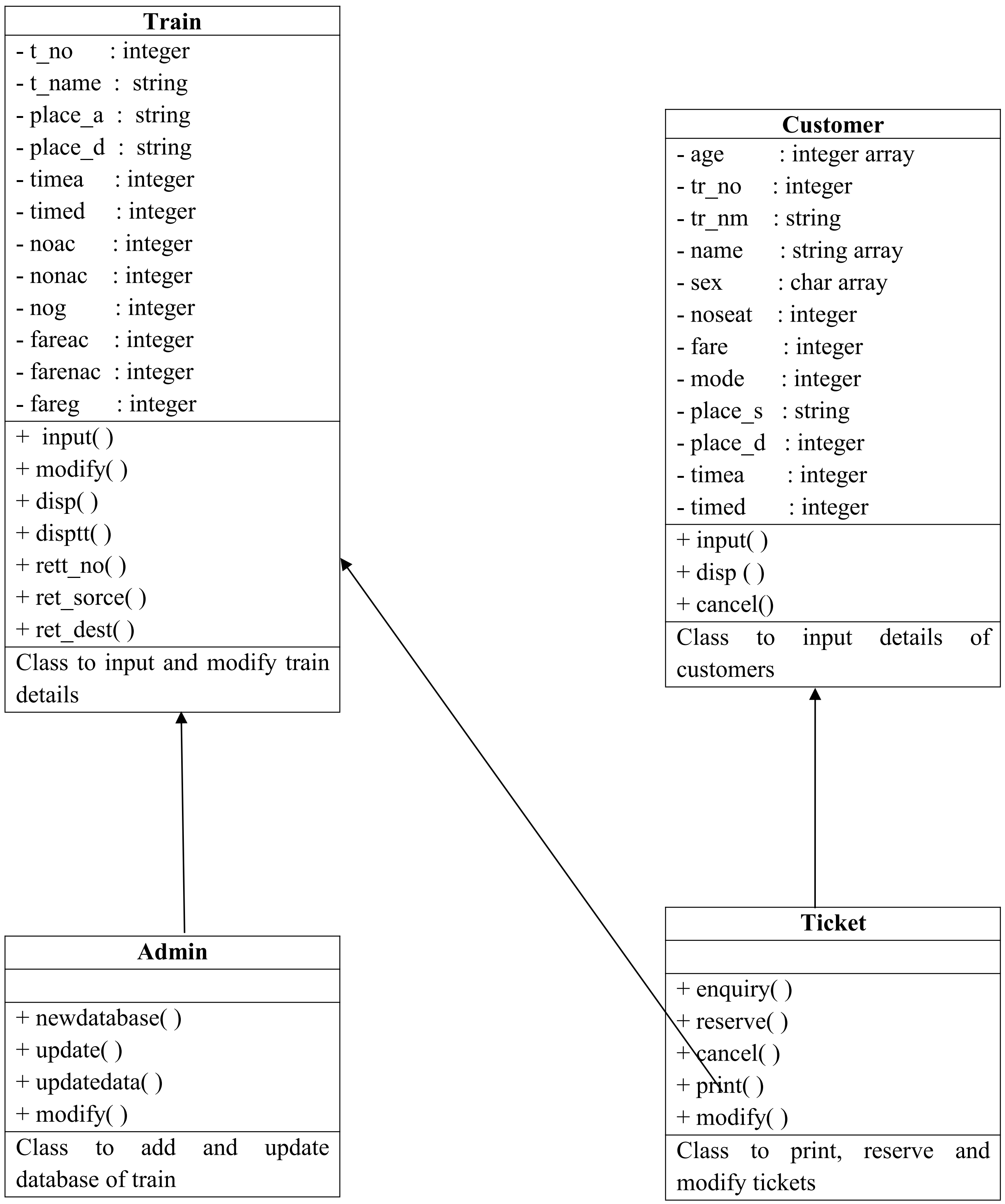

Passenger
Query for train schedule
Query result
2.1
Train Schedule
Access data from table
TRAIN table
Query for passenger info
Query result
2.2
Passenger information
Access data from table
TICKET table
Query for seat availability
Query result
2.3
Check seat availability
Access data from table
TICKET table

FLOW CHART MENU



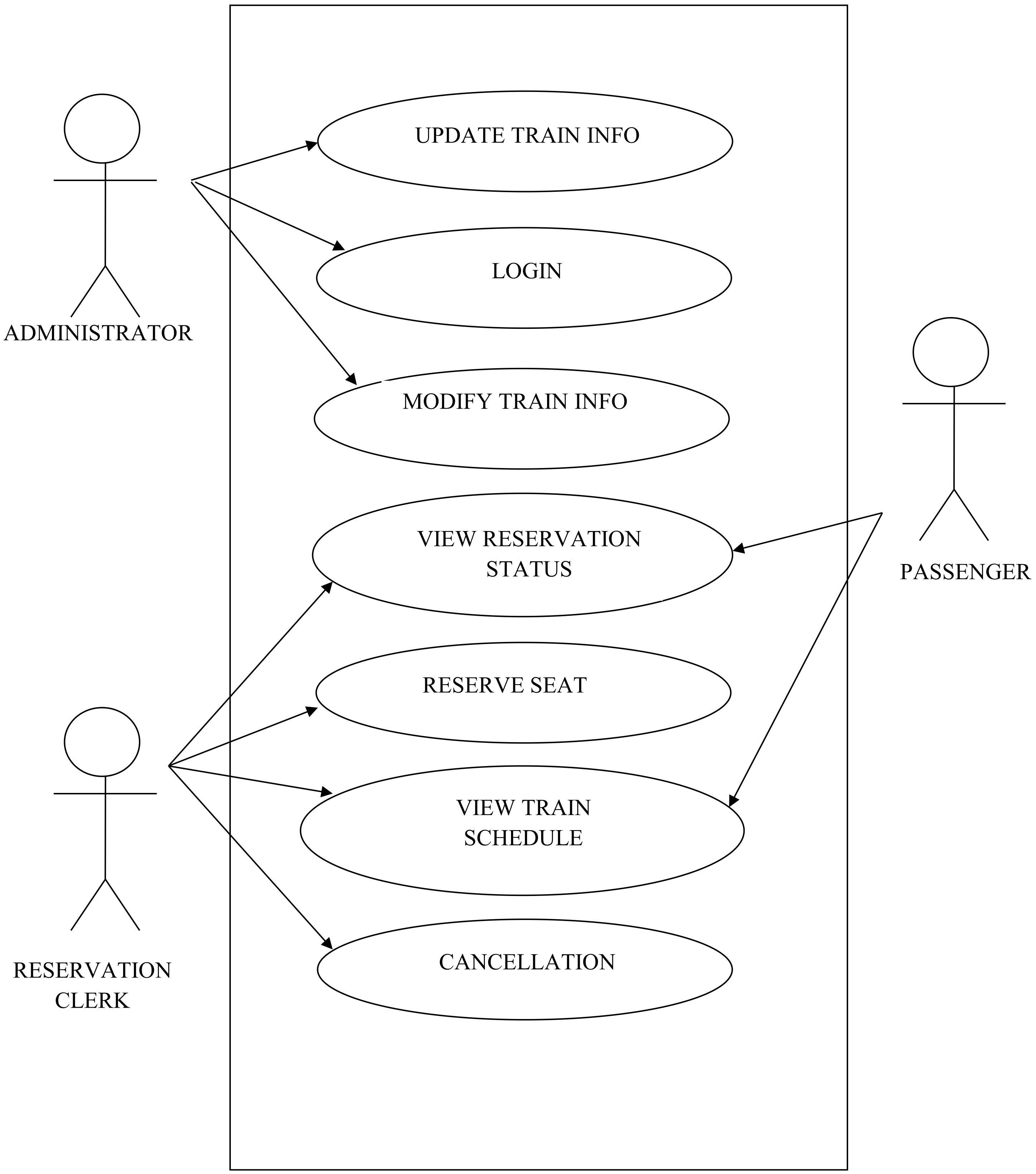
CLASS DIAGRAM



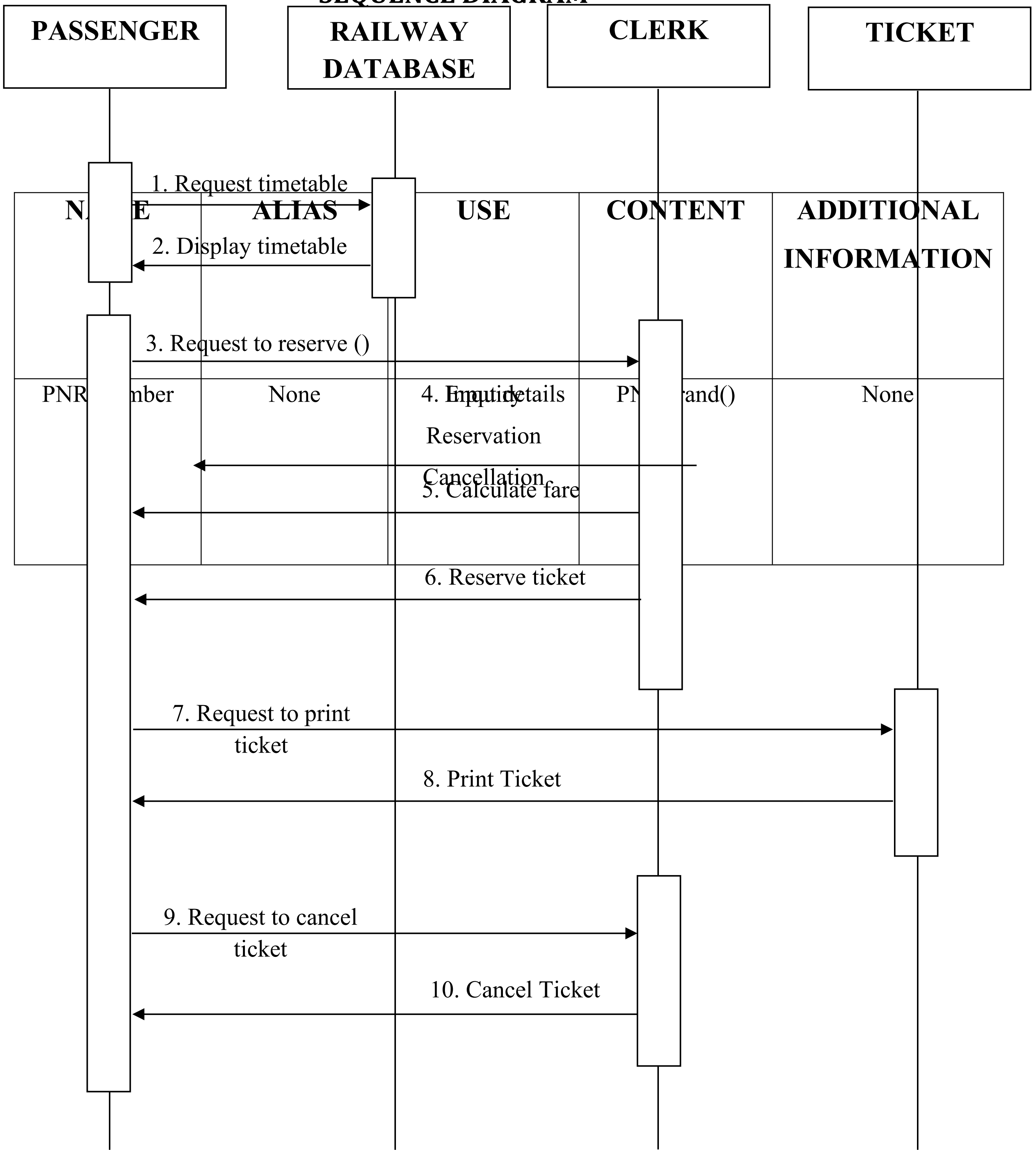


Train
- t_no : integer
- t_name : string
- place_a : string
- place_d : string
- timea : integer
- timed : integer
- noac : integer
- nonac : integer
- nog : integer
- fareac : integer
- farenac : integer
- fareg : integer
+ input( )
+ modify( )
+ disp( )
+ disptt( )
+ rett_no( )
+ ret_sorce( )
+ ret_dest( )
Class to input and modify train details
Customer
- age : integer array
- tr_no : integer
- tr_nm : string
- name : string array
- sex : char array
- noseat : integer
- fare : integer
- mode : integer
- place_s : string
- place_d : integer
- timea : integer
- timed : integer
+ input( )
+ disp ( )
+ cancel()
Class to input details of customers
Admin
+ newdatabase( )
+ update( )
+ updatedata( )
+ modify( )
Class to add and update database of train
Ticket
+ enquiry( )
+ reserve( )
+ cancel( )
+ print( )
+ modify( )
Class to print, reserve and modify tickets

USE CASE DIAGRAM



SEQUENCE DIAGRAM



NAME	ALIAS	USE	CONTENT	ADDITIONAL INFORMATION
PNR number	None	Enquiry Reservation Cancellation	PN[illegible]rand()	None

7. DOCUMENTATION

```
 #include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
#include<iostream.h>
#include<time.h>
#include<iomanip.h>
#include<fstream.h>
char f[10]="f";
char s[10]="s";
int addr,ad,flag,f1,d,m,i,amt;
float tamt;
class login
{
public:
char id[100];
char pass[100];
char *password;
void getid()
{
cout<<"Enter your id:";gets(id);
password=getpass("Enter the password:");
strcpy(pass,password);
}
void displayid()
{
cout<<"Id:";puts(id);
cout<<"Password:";puts(pass);
}
};
class detail
{
public:
int tno;
char tname[100];

char bp[100];
char dest[100];
```

```
int c1,c1fare;
int c2,c2fare;
int d,m,y;
void getdetail()
{
cout<<"Enter the details as follows\n";
cout<<"Train no:";cin>>tno;
cout<<"Train name:";gets(tname);
cout<<"Boarding point:";gets(bp);
cout<<"Destination pt:";gets(dest);
cout<<"No of seats in first class & fare per ticket:";
cin>>c1>>c1fare;
cout<<"No of seats in second class & fare per ticket:";
cin>>c2>>c2fare;
cout<<"Date of travel:";cin>>d>>m>>y;
}
void displaydetail()
{
cout<<tno<<"\t"<<tname<<"\t"<<bp<<"\t"<<dest<<"\t";
cout<<c1<<"\t"<<c1fare<<"\t"<<c2<<"\t"<<c2fare<<"\t";
cout<<d<<"-"<<m<<"-"<<y<<"\t"<<endl;
}
};
class reser
{
public:
int pnr;
int tno;
char tname[100];
char bp[10];
char dest[100];
char pname[10][100];
int age[20];
char clas[10];
int nosr;
int i;
int d,m,y;
int con;
```

```
float amc;
void getresdet()
{
cout<<"Enter the details as follows\n";
cout<<"Train no:";cin>>tno;
cout<<"Train name:";gets(tname);
cout<<"Boarding point:";gets(bp);
cout<<"Destination pt:";gets(dest);
cout<<"No of seats required:";cin>>nosr;
for(i=0;i<nosr;i++)
{
cout<<"Passenger name:";gets(pname[i]);
cout<<"Passenger age:";cin>>age[i];
}
cout<<"Enter the class f-first class s-second class:";
gets(clas);
cout<<"Date of travel:";cin>>d>>m>>y;
cout<<"Enter the concession category\n";
cout<<"1.Military\n2.Senior citizen\n";
cout<<"3.Children below 5 yrs\n4.None\n";
cin>>con;
cout<<"...........END OF GETTING DETAILS...........\n";
}
void displayresdet()
{
cout<<"...............................................\n";
cout<<"...............................................\n";
cout<<"Pnr no:"<<pnr;
cout<<"\nTrain no:"<<tno;
cout<<"\nTrain name:";puts(tname);
cout<<"Boarding point:";puts(bp);
cout<<"Destination pt:";puts(dest);
cout<<"No of seats reserved:"<<nosr;
for(i=0;i<nosr;i++)
{
cout<<"Passenger name:";puts(pname[i]);
cout<<"Passenger age:"<<age[i];
}

cout<<"\nYour class:";puts(clas);
```

```
cout<<"\nDate of reservation:"<<d<<"-"<<m<<"-"<<y;
cout<<"\nYour concession category:"<<con;
cout<<"\nYou must pay:"<<amc<<endl;
cout<<"**********************************************\n";
cout<<".........END OF RESERVATION................\n";
cout<<"**********************************************\n";
}
};
class canc
{
public:
int pnr;
int tno;
char tname[100];
char bp[10];
char dest[100];
char pname[10][100];
int age[20];
int i;
char clas[10];
int nosc;
int d,m,y;
float amr;
void getcancdet()
{
cout<<"Enter the details as follows\n";
cout<<"Pnr no:";cin>>pnr;
cout<<"Date of cancellation:";cin>>d>>m>>y;
cout<<"...........END OF GETTING DETAILS..........\n";
}
void displaycancdet()
{
cout<<"..........................................\n";
cout<<"..........................................\n";
cout<<"Pnr no:"<<pnr;
cout<<"\nTrain no:"<<tno;
cout<<"\nTrain name:";puts(tname);
cout<<"Boarding point:";puts(bp);

cout<<"Destination pt:";puts(dest);
```

```
cout<<"\nYour class:";puts(clas);
cout<<"no of seats to be cancelled:"<<nosc;
for(i=0;i<nosc;i++)
{
cout<<"Passenger name:";puts(pname[i]);
cout<<"passenger age:"<<age[i];
}
cout<<"\nDate of cancellation:"<<d<<"-"<<m<<"-"<<y;
cout<<"\nYou can collect:"<<amr<<"rs"<<endl;
cout<<"***************************************\n";
cout<<".........END OF CANCELLATION............\n";
cout<<"***************************************\n";
}
};
void manage();
void can();
void user();
void database();
void res();
void reserve();
void displaypassdetail();
void cancell();
void enquiry();
void main()
{
clrscr();
int ch;
cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";
cout<<".......WELCOME TO RAILWAY RESERVATION SYSTEM..........\n";
cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";
do
{
cout<<"^^^^^^^^^^^^^^^^^^^^^^^^MAIN MENU^^^^^^^^^^^^^^^^^^^^^^\n";
cout<<"1.Admin mode\n2.User mode\n3.Exit\n";
cout<<"Enter your choice:";
cin>>ch;
cout<<endl;
switch(ch)

{
```

```
case 1:
database();
break;
case 2:
user();
break;
case 3:
exit(0);
}
}while(ch<=3);
getch();
}
void database()
{
char *password;
char *pass="12345678";
password=getpass("Enter the admininistrator password:");
detail a;
fstream f;
int ch;
char c;
if(strcmp(pass,password)!=0)
{
cout<<"Enter the password correctly \n";
cout<<"You are not permitted to logon this mode\n";
goto h;
}
if(strcmp(pass,password)==0)
{
char c;
do
{
cout<<"...........ADMINISTRATOR MENU...........\n";
cout<<"1.Create detail data base\n2.Add details\n";
cout<<"3.Display details\n4.User management\n";
cout<<"5.Display passenger details\n6.Return to main menu\n";
cout<<"Enter your choice:";
cin>>ch;
```

```
cout<<endl;
switch(ch)
{
case 1:
f.open("t.txt",ios::out|ios::binary);
do
{
a.getdetail();
f.write((char *) & a,sizeof(a));
cout<<"Do you want to add one more record?\n";
cout<<"y-for Yes\nn-for No\n";
cin>>c;
}while(c=='y');
f.close();
break;
case 2:
f.open("t.txt",ios::in|ios::out|ios::binary|ios::app);
a.getdetail();
f.write((char *) & a,sizeof(a));
f.close();
break;
case 3:
f.open("t.txt",ios::in|ios::out|ios::binary|ios::app);
f.seekg(0);
while(f.read((char *) & a,sizeof(a)))
{
a.displaydetail();
}
f.close();
break;
case 4:
manage();
break;
case 5:
displaypassdetail();
break;
}
}while(ch<=5);

f.close();
```

```
}
h:
}
void reserve()
{
int ch;
do
{
cout<<"1.Reserve\n2.Return to the main menu\n";
cout<<"Enter your choice:";
cin>>ch;
cout<<endl;
switch(ch)
{
case 1:
res();
break;
}
}while(ch==1);
getch();
}
void res()
{
detail a;
reser b;
fstream f1,f2;
time_t t;
f1.open("t.txt",ios::in|ios::out|ios::binary);
f2.open("p.txt",ios::in|ios::out|ios::binary|ios::app);
int ch;
b.getresdet();
while(f1.read((char *) &a,sizeof(a)))
{
if(a.tno==b.tno)
{
if(strcmp(b.clas,f)==0)
{
if(a.c1>=b.nosr)

{
```

```
amt=a.c1fare;
addr=f1.tellg();
ad=sizeof(a.c1);
f1.seekp(addr-(7*ad));
a.c1=a.c1-b.nosr;
f1.write((char *) & a.c1,sizeof(a.c1));
if(b.con==1)
{
cout<<"Concession category:MILITARY PERSONNEL\n";

b.amc=b.nosr*((amt*50)/100);
}
else if(b.con==2)
{
cout<<"Concession category:SENIOR CITIZEN\n";
b.amc=b.nosr*((amt*60)/100);
}
else if(b.con==3)
{
cout<<"Concession category:CHILDERN BELOW FIVE\n";
b.amc=0.0;
}
else if(b.con==4)
{
cout<<"You cannot get any concession\n";
b.amc=b.nosr*amt;
}
srand((unsigned) time(&t));
b.pnr=rand();
f2.write((char *) & b,sizeof(b));
b.displayresdet();
cout<<"-----------------------------------------------------\n";
cout<<"--------------Your ticket is reserved-----------\n";
cout<<"-----------------End of reservation menu-------\n";
}
else
{
cout<<"**********Sorry req seats not available********\n";

}
```

```
}
else if(strcmp(b.clas,s)==0)
{
if(a.c2>=b.nosr)
{
amt=a.c2fare;
addr=f1.tellg();
ad=sizeof(a.c2);
f1.seekp(addr-(5*ad));
a.c2=a.c2-b.nosr;
f1.write((char *) & a.c2,sizeof(a.c2));
if(b.con==1)
{
cout<<"Concession category:MILITARY PRESONNEL\n";
b.amc=b.nosr*((amt*50)/100);
}
else if(b.con==2)
{
cout<<"Concession category:SENIOR CITIZEN\n";
b.amc=b.nosr*((amt*60)/100);
}
else if(b.con==3)
{
cout<<"Concession category:CHILDERN BELOW FIVE\n";
b.amc=0.0;
}
else if(b.con==4)
{
cout<<"You cannot get any concession\n";
b.amc=b.nosr*amt;
}
f2.write((char *) & b,sizeof(b));
b.displayresdet();
cout<<"---------------------------------------\n";
cout<<"--------Your ticket is reserved--------\n";
cout<<"------------End of reservation---------\n";
}
else
```

```
{
cout<<"********Sorry req no of seats not available*******\n";
}
}
getch();

goto h;
}
else
{
flag=0;
}
}
if(flag==0)
{
cout<<"............Wrong train no......................\n";
cout<<"......Enter the train no from the data base.....\n";
}
f1.close();
f2.close();
getch();
h:
}
void displaypassdetail()
{
fstream f;
reser b;
f.open("p.txt",ios::in|ios::out|ios::binary);
f.seekg(0);
while(f.read((char *) & b,sizeof(b)))
{
b.displayresdet();
}
f.close();
getch();
}
void enquiry()
{

fstream f;
```

```
f.open("t.txt",ios::in|ios::out|ios::binary);
detail a;
while(f.read((char *) & a,sizeof(a)))
{
a.displaydetail();
}
getch();
}
void cancell()
{
detail a;
reser b;
canc c;
fstream f1,f2,f3;
f1.open("t.txt",ios::in|ios::out|ios::binary);
f2.open("p.txt",ios::in|ios::out|ios::binary);
f3.open("cn.txt",ios::in|ios::out|ios::binary);
cout<<"**********CANCELLATION MENU*********\n";
c.getcancdet();
while(f2.read((char *) & b,sizeof(b)))
{
if(b.pnr==c.pnr)
{
c.tno=b.tno;
strcpy(c.tname,b.tname);
strcpy(c.bp,b.bp);
strcpy(c.dest,b.dest);
c.nosc=b.nosr;
for(int j=0;j<c.nosc;j++)
{
strcpy(c.pname[j],b.pname[j]);
c.age[j]=b.age[j];
}
strcpy(c.clas,b.clas);
if(strcmp(c.clas,f)==0)
{
while(f1.read((char *) & a,sizeof(a)))
{

if(a.tno==c.tno)
```

```
{
a.c1=a.c1+c.nosc;
d=a.d;
m=a.m;
addr=f1.tellg();
ad=sizeof(a.c1);
f1.seekp(addr-(7*ad));
f1.write((char *) & a.c1,sizeof(a.c1));
tamt=b.amc;
if((c.d==d)&&(c.m==m))
{
cout<<"You are cancelling at the date of departure\n";
c.amr=tamt-((tamt*60)/100);
}
else if(c.m==m)
{
cout<<"You are cancelling at the month of departure\n";
c.amr=tamt-((tamt*50)/100);
}
else if(m>c.m)
{
cout<<"You are cancelling one month before the date of departure\n";
c.amr=tamt-((tamt*20)/100);
}
else
{
cout<<"Cancelling after the departure\n";
cout<<"Your request cannot be completed\n";
}
goto h;
displaycancdet();
}
}
}
else if(strcmp(c.clas,s)==0)
{
while(f1.read((char *) & a,sizeof(a)))
{

if(a.tno==c.tno)
```

```
{
a.c2=a.c2+c.nosc;
d=a.d;
m=a.m;
addr=f1.tellg();
ad=sizeof(a.c2);
f1.seekp(addr-(5*ad));
f1.write((char *) & a.c2,sizeof(a.c2));
tamt=b.amc;
if((c.d==d)&&(c.m==m))
{
cout<<"You are cancelling at the date of departure\n";
c.amr=tamt-((tamt*60)/100);
}
else if(c.m==m)
{
cout<<"You are cancelling at the month of departure\n";
c.amr=tamt-((tamt*50)/100);
}
else if(m>c.m)
{
cout<<"You are cancelling one month before the date of departure\n";
c.amr=tamt-((tamt*20)/100);
}
else
{
cout<<"Cancelling after the departure\n";
cout<<"Your request cannot be completed\n";
}
goto h;
displaycancdet();
}
}
}
}
else
{
flag=0;

}
```

```
}
h:
if(flag==0)
{
cout<<"Enter the correct pnr no\n";
}
f1.close();
f2.close();
f3.close();
getch();
}
void can()
{
int ch;
do
{
cout<<".................CANCELLATION MENU.........\n";
cout<<"1.Cancell\n2.Return to the main menu\n";
cout<<"Enter your choice:";
cin>>ch;
cout<<endl;
switch(ch)
{
case 1:
cancell();
break;
}
}while(ch==1);
getch();
}
void user()
{
login a;
int ch;
cout<<"*****************************************************\n";
cout<<"***********WELCOME TO THE USER MENU**\n";
cout<<"****************************************************\n";
char *password;

fstream f;
```

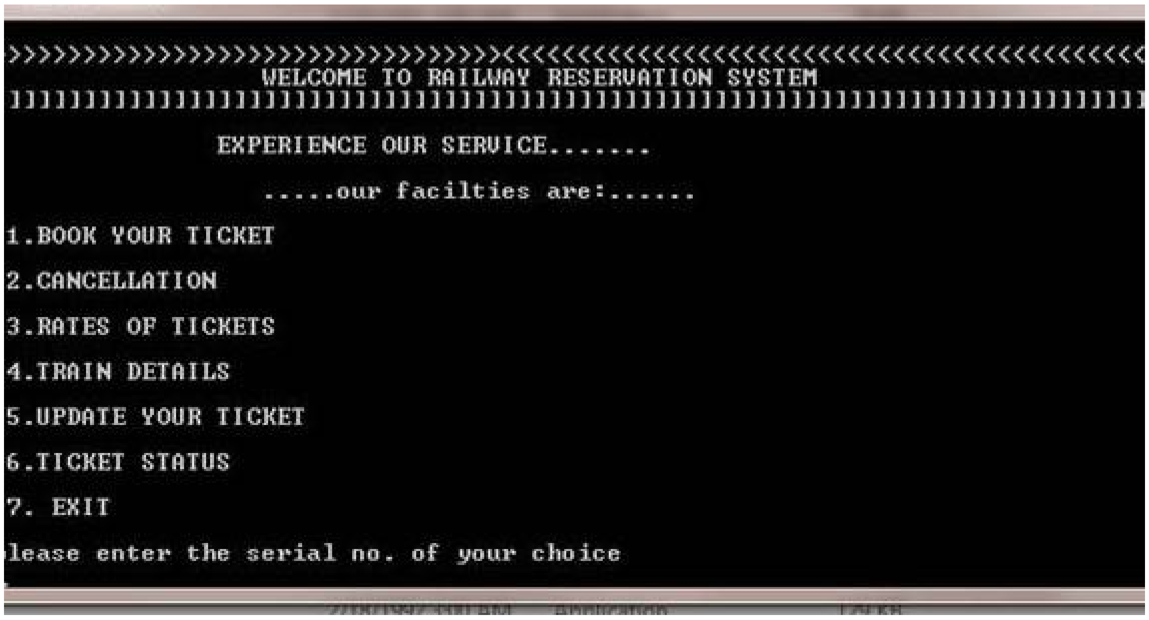
```
f.open("id.txt",ios::in|ios::out|ios::binary);
char id[100];
puts("Enter your id:");gets(id);
password=getpass("Enter your password:");
while(f.read((char *) & a,sizeof(a)))
{
if((strcmp(a.id,id)==0)&&(strcmp(a.pass,password)==0))
{
do
{
cout<<"1.Reserve\n2.Cancell\n3.Enquiry\n4.Return to the main menu\n";
cout<<"Enter your choice:";
cin>>ch;
cout<<endl;
switch(ch)
{
case 1:
reserve();
break;
case 2:
cancell();
break;
case 3:
enquiry();
break;
}
}while(ch<=3);
goto j;
}
else
{
d=1;
}
}
if(d==1)
{
cout<<"Enter your user id and password correctly\n";
}

getch();
```

```
j:
}
void manage()
{
int ch;
fstream f;
char c;
login a;
cout<<".........WELCOME TO THE USER MANAGEMENT MENU........\n";
do
{
cout<<"1.Create id data base\n2.Add details\n";
cout<<"3.Display details\n4.Return to the main menu\n";
cout<<"Enter your choice:";
cin>>ch;
cout<<endl;
switch(ch)
{
case 1:
f.open("id.txt",ios::out|ios::binary);
do
{
a.getid();
f.write((char *) & a,sizeof(a));
cout<<"Do you want to add one more record\n";
cout<<"y-Yes\nn-No\n";
cin>>c;
}while(c=='y');
f.close();
break;
case 2:
f.open("id.txt",ios::in|ios::out|ios::binary|ios::app);
a.getid();
f.write((char *) & a,sizeof(a));
f.close();
break;
case 3:
f.open("id.txt",ios::in|ios::out|ios::binary);

f.seekg(0);
while(f.read((char *) & a,sizeof(a)))
{
a.displayid();
}
f.close();
break;
}
}while(ch<=3);
getch();
}
```

8. RESULT

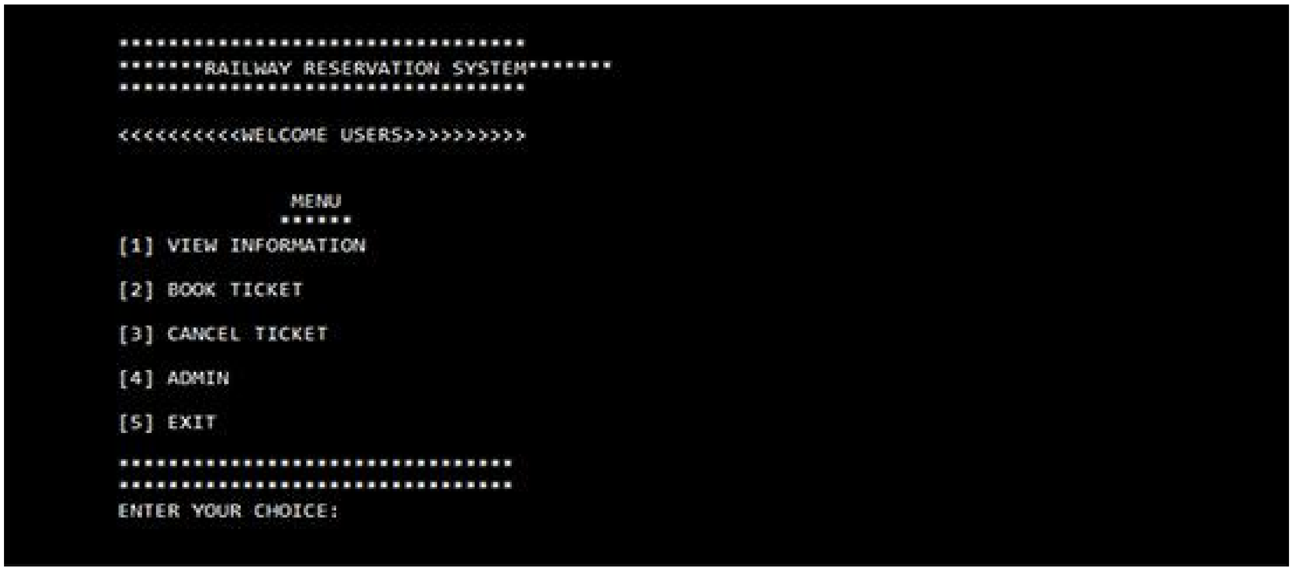
OUTPUT THE RAILWAY RESERVATION SYSTEM

(a).WELOCOME THE INDIAN RAILWAYS



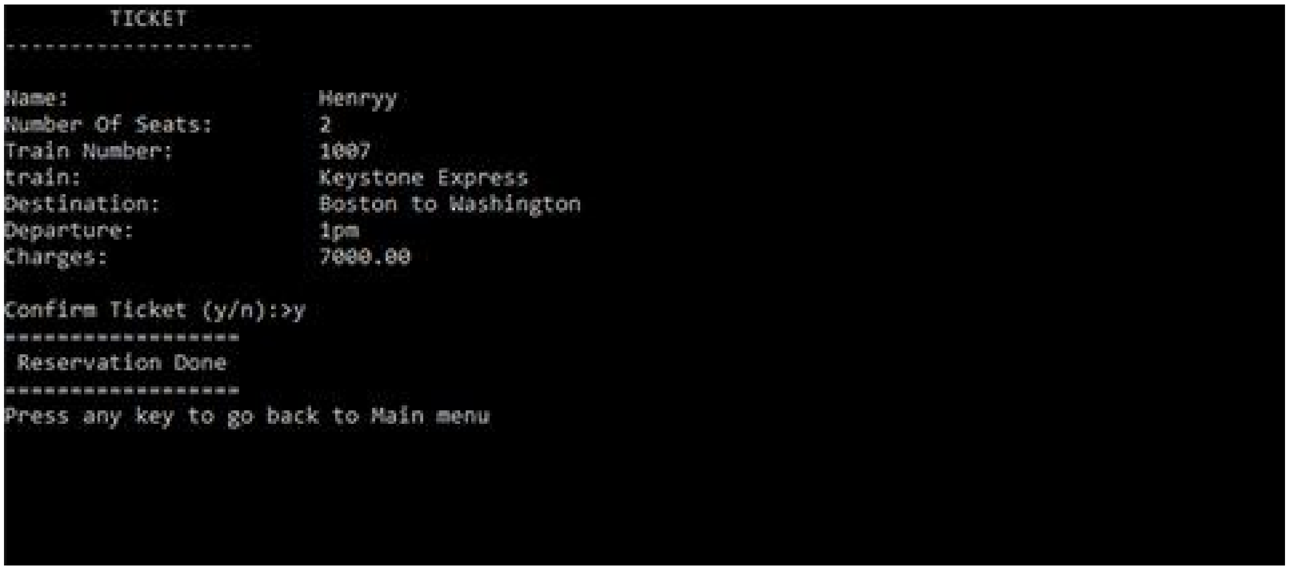
```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
                  WELCOME TO RAILWAY RESERVATION SYSTEM
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]

              EXPERIENCE OUR SERVICE.......

                 .....our facilties are:......

1.BOOK YOUR TICKET

2.CANCELLATION

3.RATES OF TICKETS

4.TRAIN DETAILS

5.UPDATE YOUR TICKET

6.TICKET STATUS

7. EXIT

lease enter the serial no. of your choice
```

(b).CONFIRM THE TICKET BOOKING



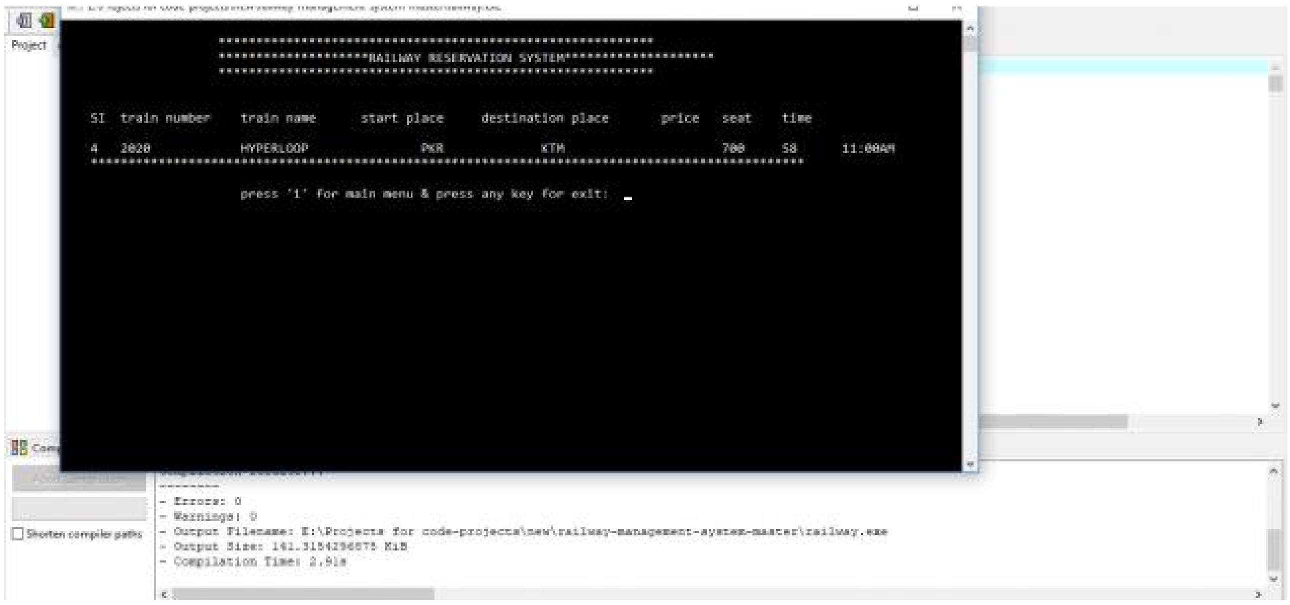
```
*********************************
*******RAILWAY RESERVATION SYSTEM*******
*********************************

<<<<<<<<<<WELCOME USERS>>>>>>>>>>>


              MENU
             ******
[1] VIEW INFORMATION

[2] BOOK TICKET

[3] CANCEL TICKET

[4] ADMIN

[5] EXIT

********************************
********************************
ENTER YOUR CHOICE:
```

(c).ADMIN MODE



```
        TICKET
-------------------

Name:                   Henryy
Number Of Seats:        2
Train Number:           1007
train:                  Keystone Express
Destination:            Boston to Washington
Departure:              1pm
Charges:                7000.00

Confirm Ticket (y/n):>y
==================
 Reservation Done
==================
Press any key to go back to Main menu
```

(d)THE TICKET BOOKING RESERVATION



```
                ***********************************************************
                ********************RAILWAY RESERVATION SYSTEM********************
                ***********************************************************

 SI  train number    train name      start place     destination place       price   seat    time

 4   2020           HYPERLOOP              PKR            KTM                  700     58      11:00AM
 *****************************************************************************************

                   press '1' for main menu & press any key for exit:
```

```
--------
- Errors: 0
- Warnings: 0
- Output Filename: E:\Projects for code-projects\new\railway-management-system-master\railway.exe
- Output Size: 141.3154296875 KiB
- Compilation Time: 2.91s
```

9. LIMITATIONS

Our project meets the following limitations:

1) The software is not able to reserve tickets for more than 10 people per train.
2) The fare allotted for every reservation is independent of Kilometres travelled instead it is set for every mode (AC, Non-AC or General) of each train.
3) The software is made such to carry out reservation in max 15 trains.
4) The software does not support multi-day reservation system, i.e., the reservations cannot be done in advance rather it is carried out for single day.
5) The software does not provide concession in fare rates for children, aged people, armament etc. i.e., the fare identical for all people.
6) The software does not take into consideration the stations falling in between the source and destination station.

10.FUTURE SCOPE

If anyone wants to extend this project then he/she can make an additional database of Train Fare. And database for updated availability of seats which is available after the cancellation of ticket on that specific train etc. He/she can also add some more command buttons in the existing software and extend working of the existing software.

Implementations of this project idea are in industrial use. Hence, this can be used for suggesting improvements in design, performance and greater usability. Apart from the industrial applications, it is a research-oriented project as well, the task of performance evaluation of different database designs, for efficiency, is in this spirit.

11.REFERENCES

- **BOOKS USED:**
  - Object Oriented Programming C++ (E. Balaguruswami)
  - Introducing C++ (Sumita Arora)
  - Software Engineering (Shalini Puri)
  - Software Engineering (Pressman)

- **SITES USED :**
  - www.scribd.com
  - www.irctc.com
  - www.indianrail.com
  - www.wikipedia.org
  - www.yatra.com
  - www.trainenquiry.com