



Railway Reservation System

Peddi Oliva Joicy and Vidiyala Prathyusha

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

March 29, 2021

RAILWAY RESERVATION SYSTEM

By

P.OlivaJoicy H.No 160618733035
V.Prathyusha H.No 160618733058

STUDENT
INDIA

ABSTRACT

The railway reservation system facilitates the passengers to enquiry about the trains available on the basis of source and destination, booking and cancellation of tickets, enquiry about the status of the booked ticket, etc. The aim of case study is to design and develop a data base maintaining records of different trains, train status and passengers. This project contains introduction to the railways reservation system. It is the computerized system of reserving the seats of train seats in advance. It is mainly used for a long route. Online reservation has made the process for the reservation of seats very much easier than ever before.

In our country India, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Railway reservation system, has described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization it better utilization of resources. Administrator of the project, with the help of a password, can enter new train record, display all train records, modify train records and delete train records. The record of train includes its number, name, source, destination, and days on which it is available, whereas record of train status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.

CHAPTER 1

1. INTRODUCTION

Railway reservation system is developed for to automate the railways reservation system. It includes modules required to successfully operate railways reversion process smoothly. It has train master to add modified train information, train schedule to enter train journey details include all the station name, arrival time and departure time. It includes automatic fair calculation as per the distance between two stations. Reservation module consists of automatic seat number and coaches no allocation system. Daily schedule for updating of not conform seat and coach no. All master like train master, Train schedule, reservation fees, cancellation fees, charges can be modified individually from front end and changes reflect in all modules immediately. Therefore proposed “Railway reservation system” has been designed to automate the process of railway for ticket reservation and back office activities. System can make the daily activities efficient and provide the fast response.

The “Railway reservation system” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for a particular need of a company to carry out operations in a smooth and effective manner. The application is reduced has much as possible to avoid errors while entering the data. It also provides error messages while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user friendly. Railway reservation system, has described, can lead to error free, and secure, reliable and fast managing system. It can assist the user to concentrate on the other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of ticket, train, customer, seat, payment. Every Railway reservation system has different train needs, therefore we designed exclusive employee management systems that are adopted to your managerial requirements.

1.1 PURPOSE

The purpose of the source is to describe the railway reservation system which provides the train timing details, reservation, billing and cancellation on various types of reservation mainly,

- Conform reservation for conform seat.
- Reservation against cancellation.
- Waiting list reservation.
- Online reservation.
- Tatkal reservation.

1.2 SCOPE

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of the past year perfectly and vividly. It also helps in current all works related to railway reservation system. It will be also reduced the cost of collecting the management and collection procedure will go on smoothly.

1.3 STUDY OF EXISTING SYSTEM

In the existing system the exams are done only manually but in proposed system we have to computerize the exams using this application.

- Lack of security of Data.
- More man power.

- Time consuming.
- Consumes large volume of pure work
- Needs manual calculations.
- No direct role for the higher officials.

Existing railway reservation system not having feature of displaying appropriate messages for events and errors which occur while accessing the system. Proper validation and session has not been setup by which it system not able to identify and differentiate users, so cannot be used appropriately for multi user environment. This railway reservation system do not able to provide the details of customers transactions and their payment mode along with date time and processing charge details. Making changing in information and perform administrative task was not possible while using existing system.

It does not able to provide availability details to chargers 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 the processing and charges details.

1.4 PROPOSED SYSTEM

The aim of the proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work.

- Security of data.
- Ensure data accuracy's
- Proper control of the higher officials.
- Minimize manual data entry.
- Minimum time needed for the various processing.
- Greater efficiency.

- Better service.
- User friendliness and interactive.

CHAPTER 2

2. REQUIREMENTS

2.1 SOFTWARE REQUIREMENT

Software	Description
Windows	Operating System
Turbo C++ 4.5	For execution of the program
MS Word	For presentation of the output

2.2 HARDWARE REQUIREMENT

Hardware	Description
Ram	256MB
Hard Disk	20 GB
Processor	Pentium III

Monitor	14.4"
Keyboard	104 keys

CHAPTER 3

3. SYSTEM DESIGN

In the system design process, the primary objective is to identify user requirements and to build a system that satisfies these requirements. Basically, the design phase concentrates on "how" the system is developed. Design describes a final system and the process by which it is developed. It refers to the technical specification that will apply in implementing the candidate system. The design phase is the second important steps in the system development life cycle.

During this phase the analyst does the following:

- Schedules design activities.
- Works with the user to determine the different data inputs to the system.
- Draws the model of new system, using data flow diagram & entityrelationship diagrams.
- Defines the data requirements with a data dictionary.
- Writes program specifications.
- Identifies and orders any hardware and software that the system design phase

3.1 ER DIAGRAM

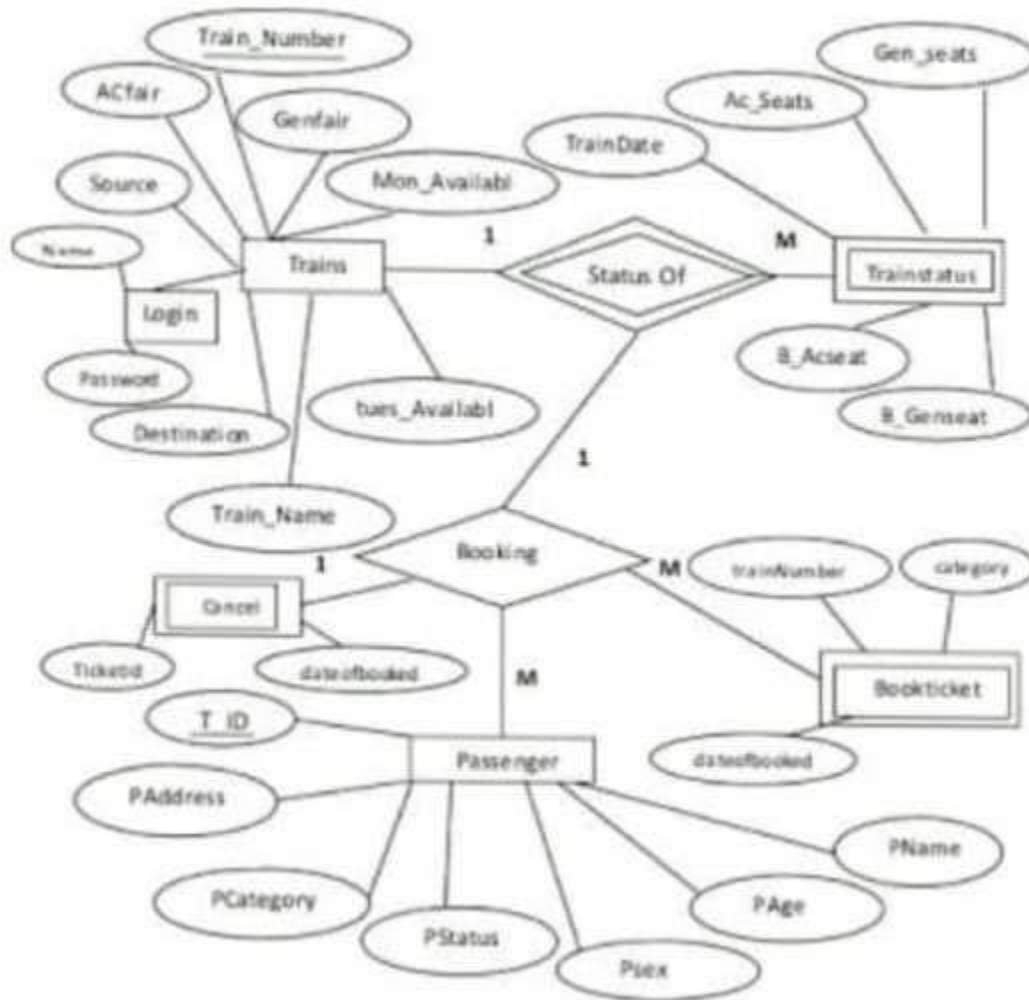


Fig 3.1 ER Diagram

3.2 USE CASE DIAGRAM

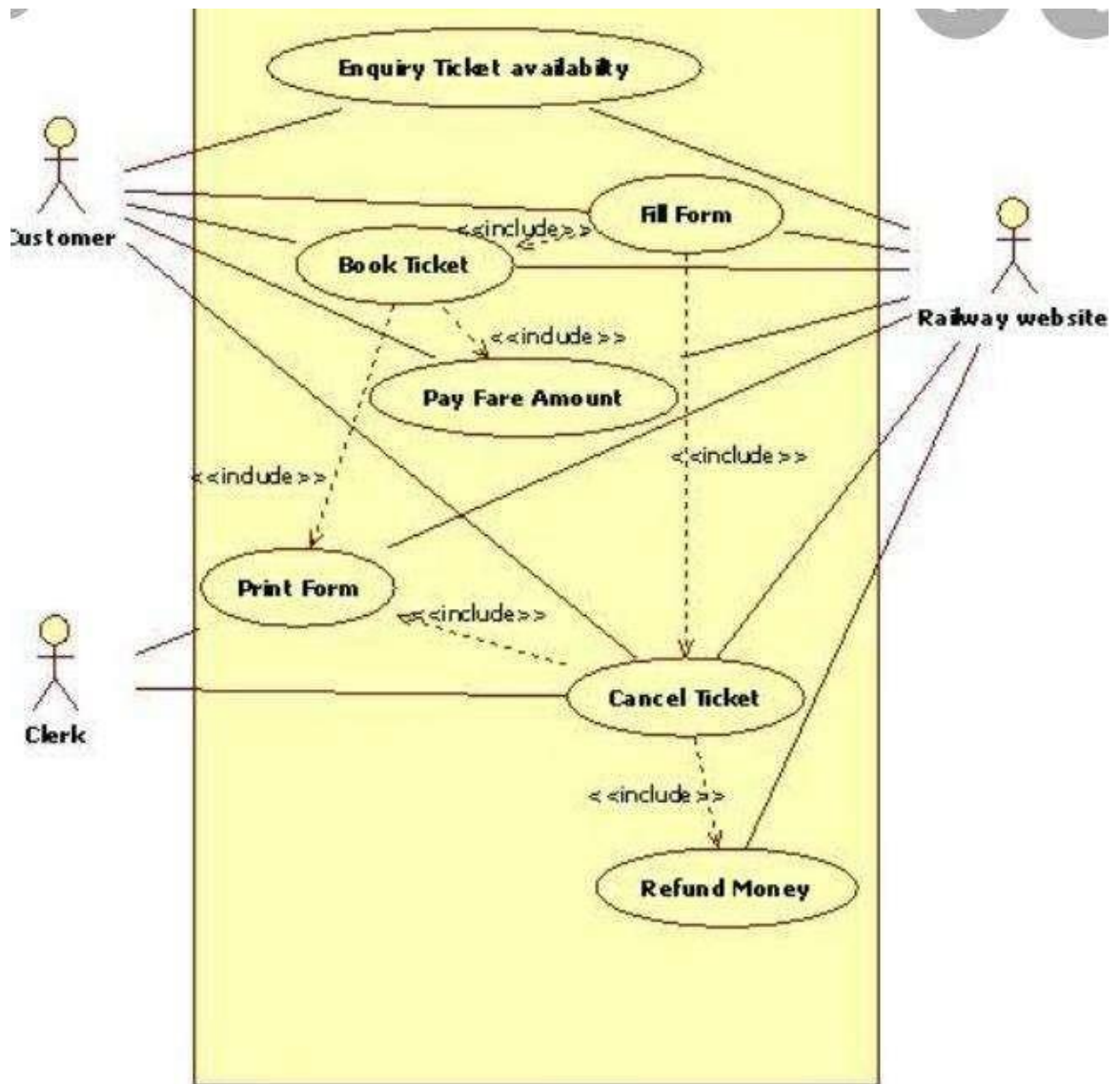


Fig 3.2 Use Case Diagram

CHAPTER 4

4. CODING

```
#include<conio.h>
#include<stdio.h>
#include<process.h
>
#include<string.h>
class pn { public:
int pnr;
}x;
void star(); // FUNCTION TO PRINT STARS ON //
SCREEN
class train { public:
int tno,ac,sleeper,sc,time,dist,fac,fsleeper,fsc; char from[100],to[100],name[100];
public:
void modify(); // FUNCTION TO MODIFY TRAIN
// RECORDS
void del(); // FUNCTION TO DELETE TRAIN
// RECORDS
void display(); // FUNCTION TO DISPLAY TRAIN
// RECORDS
void acp(); // FUNCTION TO ACCEPT TRAIN
// RECORDS
void check(); // FUNCTION TO CHECK AVAILABILITY
void update(int); // FUNCTION TO UPDATE NO. OF
// SEATS AFTER CANCELLATION
int ret() // ACCESSOR FUNCTION TO RETURN TRAIN
//NUMBER
{ return
tno;
```

```

}
char *f1()    // ACCESSOR FUNCTION TO RETURN
              //STARTING POINT
{ return
from;
}
char *t1()    // ACCESSOR FUNCTION TO RETURN
              //DESTINATION
{
return to;
}
}t;    // END OF CLASS
void admin();// ADMIN SECTION TO MANIPULATE TRAIN
          // RECORDS
void main()// STARTING MAIN FUNCTION }
{ int n;
while(1
) {
clrscr();
star();
gotoxy(30,2);cout<<" WELCOME TO INDIAN RAILWAYS";
gotoxy(5,5); cout<<" ENTER 1 TO CHECK AVAILABILITY ";
gotoxy(5,6); cout<<" ENTER 2 TO BOOK TICKETS";
gotoxy(5,7); cout<<" ENTER 3 FOR CANCELLATION";
gotoxy(5,8); cout<<" ENTER 4 FOR ADMIN ACCESS";
gotoxy(5,9); cout<<" ENTER 5 TO EXIT";
gotoxy(5,14);cout<<" ENTER YOUR CHOICE    -    ";
cin>>n; clrscr();
switch (n) { case
1 : t.check();

```

```

getch();
break; case 2 :
b.get();
getch();
break; case 3 :
b.cancel();
getch();
break;
case 4 : admin();
            break;
case 5 : exit(0);
default: clrscr();
star();
            gotoxy(10,10);cout<<"        WRONG CHOICE";
            getch();
} // END OF SWITCH
} // END OF WHILE
} // END OF MAIN
void train::display() // FUNCTION TO DISPLAY TRAIN
                    //RECORD
{
star();
gotoxy
(5,3);c
out<<"
TRAIN
NUMB
ER

"<<tno

```

```

;
gotoxy
(5,4);c
out<<"
TRAIN
NAME
"<<name;
gotoxy(5,5);cout<<" STARTING POINT"<<from;

gotoxy(5,6);cout<<" DESTINATION                               "<<to;
gotoxy(5,7);cout<<" NUMBER OF AC TICKETS AVAILABLE"<<ac;
gotoxy(5,8);cout<<" PRICE OF EACH AC TICKET           Rs."<<fac;
gotoxy(5,9);cout<<" NUMBER OF SLEEPER TICKETS AVAILABLE"<<sleeper;
gotoxy(5,10);cout<<" PRICE OF EACH SLEEPER CLASSTICKET   Rs.
                "<<fsleeper;
gotoxy(5,11);cout<<" NUMBER OF SECOND CLASS TICKETS AVAILABLE "<<sc;
gotoxy(5,12);cout<<" PRICE OF EACH SECOND CLASSTICKET   Rs.
                "<<fsc; gotoxy(5,13);cout<<" DURATION OF JOURNEY (hrs)"<<time;
gotoxy(5,14);cout<<" TOTAL DISTANCE (km)"<<dist; getch();
}
void train::acp() // FUNCTION TO ACCEPT TRAIN
                // RECORDS
{
clrscr();
star();
gotoxy(5,3);cout<<" ENTER TRAIN NUMBER";cin>>tno;
gotoxy(5,4);cout<<" ENTER TRAIN NAME";gets(name);
gotoxy(5,5);cout<<" ENTER STARTING POINT";gets(from);
gotoxy(5,6);cout<<" ENTER DESTINATION";gets(to);
gotoxy(5,7);cout<<" ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE

```

```

        ";cin>>ac;
gotoxy(5,8);cout<<" ENTER PRICE OF EACH AC TICKETRs. ";cin>>fac;
gotoxy(5,9);cout<<" ENTER NUMBER OF SLEEPER TICKET TO BE MADE
AVAILABLE";cin>>sleeper; gotoxy(5,10);cout<<" ENTER PRICE OF EACH SLEEPER
TICKET    Rs.
";cin>>fsleeper;
gotoxy(5,11);cout<<" ENTER NUMBER OF SECOND CLASS TICKETS TO BE
MADE AVAILABLE";cin>>sc; gotoxy(5,12);cout<<" ENTER PRICE OF EACH
SECOND CLASS TICKET  Rs.
";cin>>fsc;
gotoxy(5,13);cout<<" ENTER DURATION OF JOURNEY (hrs) ";cin>>time;
gotoxy(5,14);cout<<" ENTER TOTAL DISTANCE (km) ";cin>>dist;
}    // END OF FUNCTION

```

```

void train::modify() // FUNCTION TO MODIFY TRAIN
//RECORD
{ clrscr();
star();
fstream
rr;
rr.open("train.dat",ios::in|ios::out|ios::binary);
int id,f=0,n=0,p;
gotoxy(5,3);cout<<" ENTER TRAIN NUMBER OF TRAIN TO
        BE MODIFIED    -    ";
cin>>id;

while(rr.read((char*)&t,sizeof t))
{
n++;

```

```

if(id==t.ret())
{
acp();
p=(n-
1)*(siz
eof t);
rr.seek
p(p,ios
::beg);
rr.write
((char*
)&t,siz
eof t);
f=1; }
}
if(f==0
)
{
gotoxy(10,6);cout<<" SORRY RECORD NOT FOUND !!";
} rr.close();
} //END OF FUNCTION
void train::check() // FUNCTION TO CHECK TICKET
// AVAILABILITY
{
star(); char
T[100],f[100]; int
flag=0;
gotoxy(10,5);cout<<" ENTER THE STARTING POINT - "; gets(f);

```



```

gotoxy(10,8);cout<<" ENTER THE DESTINATION      -
      "; gets(T); fstream rr;
rr.open("train.dat",ios::in|ios::binary);
while(rr.read((char*)&t,sizeof t))
{
if( strcmpi(f,t.f1())==0 && strcmpi(T,t.t1())==0 )
{
clrscr();
star();
flag=1;

t.display();
}
if(flag==0)
{
gotoxy(10,14);cout<<" SORRY NO AVAILABLE TRAINS FOUND ";
}
}

```

```

void book::get()      // FUNCTION TO BOOK
TICKETS{ int n=1; while(n==1)
{      // WHILE 1
clrscr();
star();
gotoxy(33,3);cout<<" WELCOME TO BOOKING MENU";
gotoxy(5,6);cout<<" ENTER 1 TO BOOK TICKETS";
gotoxy(5,8);cout<<" ENTER 2 TO GO BACK";

```

```

gotoxy(5,14);cout<<" ENTER YOUR CHOICE    -
        "; cin>>n; if(n==1)
{      //      IF
1 clrscr(); star();
int flag=0;
gotoxy(10,3);cout<<" ENTER THE BOARDING POINT - "; gets(pfrom);
gotoxy(10,6);cout<<" ENTER THE DESTINATION - "; gets(pto);
fstream rr;
rr.open("train.dat",ios::in|ios::binary);
fstream pas;
pas.open("passenger.dat",ios::app|ios::binary);

rr.seekg(0,ios::beg);
while(rr.read((char*)&t,sizeof t)
{      // WHILE 2
if( strcmpi(pfrom,t.f1())==0 && strcmpi(pto,t.t1())==0 )
{
clrscr();
star();
flag=1;
t.display();
}
} // END OF WHILE 2

rr.close();
if(flag==0)
{
gotoxy(10,14);cout<<" SORRY NO AVAILABLE TRAINS FOUND    ";
getch();
}

```

```

}
// STARTING THE BOOKING PROCESS
if(flag==1)
{
    // IF 2
    int f1=0;
    clrscr();
    star();
    gotoxy(5,5);cout<<" ENTER THE TRAIN NUMBER OF 3 if (f1==0)
    {
    gotoxy(10,10);cout<<" SORRY YOU HAVE ENTERED INCORRECT TRAIN
    NUMBER
    ";
    getch();
    }
} // END OF IF 2
pas.close();
rr.close();
} // END OF IF 1
} // END OF WHILE 1
} // END OF FUNCTION

```

```

int book::assign() // FUNCTION TO CALCULATE AND
// ASSIGN BOOKING DETAILS
{
    int
    n=0,p;
    fstream rr;
    rr.open("train.dat",ios::in|ios::out|ios::binary);
    rr.seekg(0,ios::beg);
    while(rr.read((char*)&t,sizeof t))

```

```

{ n++;
if(no==t.ret()
)
{
if(strcmpi(clas,"ac")==0)
{
amt=tickets*t.fac; p=(n-
1)*sizeof (t);
rr.seekp(p,ios::beg);
t.ac=t.ac-tickets;
if(t.ac<0)
{ return 0; } else
rr.write((char*)&t,sizeof
t);
} else
if(strcmpi(clas,"sleeper")==0)
{
amt=tickets*t.fsleeper;
p=(n-1)*sizeof (t);
rr.seekp(p,ios::beg);
t.sleeper=t.sleeper-tickets;
if(t.sleeper<0) return 0;
else
rr.write((char*)&t,sizeof t);
} else {
amt=tickets*t.fsc;
p=(n-1)*sizeof (t);
rr.seekp(p,ios::beg);

```

```

t.sc=t.sc-tickets; if(t.sc<0)
return 0; else
rr.write((char*)&t,sizeof t);
}
break; }

} fstream
b1;
b1.open("booking.dat",ios::app|ios::binary)
; fstream rr2;
rr2.open("pnr.dat",ios::in|ios::binary);
if(!rr2) { rr2.close();
rr2.open("pnr.dat",ios::in|ios::app|ios::binary);
rr2.seekp(0,ios::beg);
x.pnr=1000;
rr2.write((char*)&x,sizeof x);
rr2.close(); } else { int d;
rr2.seekg(0,ios::beg);
while(rr2.read((char*)&x,sizeof
x))
{ d=x.pnr;
}
rr2.close();
fstream r2;
r2.open("pnr.dat",ios::app|ios::binary);
x.pnr=++d;
r2.write((char*)&x,sizeof x);
r2.close();
}
b.pnr=x.pnr;

```

```

b1.write((char*)&b,sizeof
b); b1.close(); return 1;
} //END OF FUNCTION

void book::show()    // FUNCTION TO DISPLAY BOOKING
                    //DETAILS

{ clrscr();
star();
gotoxy(33,2);cout<<" TICKET";
gotoxy(5,4); cout<<" PNR NUMBER      -      "<<pnr;
gotoxy(5,5); cout<<" TRAIN NUMBER    -      "<<no;
gotoxy(5,6); cout<<" TOTAL PASSENGERS -      "<<tickets;
gotoxy(5,7); cout<<" TOTAL AMOUNT    -      "<<amt;
gotoxy(5,8); cout<<" BOARDING POINT -      "<<pfrom;
gotoxy(5,9); cout<<" DESTINATION      -      "<<pto;
gotoxy(8,23);cout<<" YOUR TICKET HAS BEEN RESERVED";
}                    // END OF FUNCTION

```

CHAPTER 5

5. TESTING

Testing of software can be done both automation and manual testing method. But it's totally depends the project budget associated with projects and which testing method will be benefited to the project. This article provides basic information about manual testing and automation testing.

Automation Testing

The method takes automation tool support to execute the test cases is known as automation testing. Automation testing good for large projects.

It is a method which uses automation tools to nm tests that repeat predefined actions.

Manual Testing

Manual testing is a method used by software developers to run tests manually. There are many manual testing types which are come out manually as well as automatically these are

Black Box Testing

It is a testing method to test functions and requirements of the system. It does not test the internal part of the system.

White Box Testing

It is a testing method based on the information of the internal logic of the applications and also known as glass box testing.

Integration Testing

Integrated modules testing method to verify joint functionality after integration-modules are initially code modules, individual applications, client and server applications on a network etc.

CHAPTER 6

6. OUTPUT SCREENS

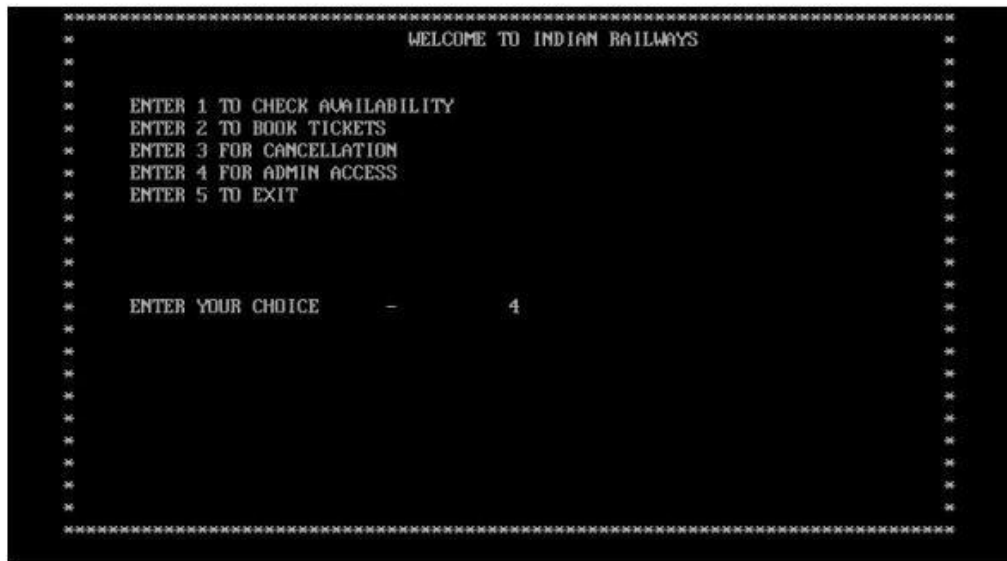
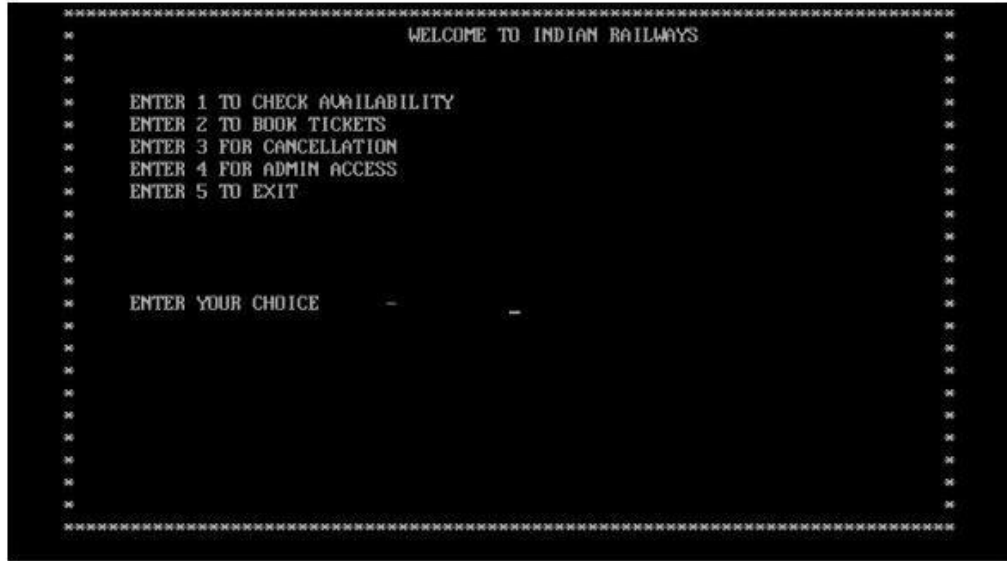


Fig 6.1 welcome page


```

*****
*
*   ENTER TRAIN NUMBER                1002
*   ENTER TRAIN NAME                   JAN SHATABDI
*   ENTER STARTING POINT               HARIDWAR
*   ENTER DESTINATION                 AMRITSAR
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 80
*   ENTER PRICE OF EACH AC TICKET     Rs. 1000
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 90
*   ENTER PRICE OF EACH SLEEPER TICKET Rs. 850
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 80
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 700
*   ENTER DURATION OF JOURNEY (hrs)   6
*   ENTER TOTAL DISTANCE (km)        411
*   ENTER MORE RECORDS ? (Y/N)      -   Y
*
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                1003
*   ENTER TRAIN NAME                   KASHMIR EXPRESS
*   ENTER STARTING POINT               KOLKATA
*   ENTER DESTINATION                 JAMMU
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 70
*   ENTER PRICE OF EACH AC TICKET     Rs. 2200
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SLEEPER TICKET Rs. 1700
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 60
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 1300
*   ENTER DURATION OF JOURNEY (hrs)   26
*   ENTER TOTAL DISTANCE (km)        2588
*   ENTER MORE RECORDS ? (Y/N)      -   N
*
*
*
*
*
*
*
*
*
*
*
*****

```

Fig 6.4 Entering Details for Jan Shatabdi & Kashmir Express

CHAPTER 7

7. CONCLUSION

The main of developing reservation system is to provide all information that is required by the users. User friendliness is a must that is the user must get the details without complicated searching procedures .Other important requirements of software are data security, extensibility and maintainability.

CHAPTER 8

8. REFERENCES

1. <http://www.google.com/>
2. <http://en.wikipedia.org>
3. Computer science with C++ by Sumita Arora