IACM
H-9, SOUTH EX.-I
NEW DELHI

CENTRE CODE: 1503

PROJECT
ON
RAILWAY RESERVATION SYSTEM

SUBMITTED BY:
RAJIW KUMAR
ROLL NO.: 510731861
2007-2010

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

OF SIKKIM MANIPAL UNIVERSITY, INDIA

SIKKIM-MANIPAL UNIVERSITY OF HEALTH,

MEDICAL AND TECHNOLOGICAL SCIENCES
DISTANCE EDUCATION WING
SYNDICATE HOUSE, MANIPAL-576 104

STUDENT DECLARATION

I here by declare that the project report entitled

“RAILWAY RESERVATION SYSTEM”

submitted in partial fulfillment of the requirements for the degree of ***Bachelor of science in information technology*** *to Sikkim-Manipal University, India, is my original work and not submitted for the award of any other degree, diploma, fellowship, or any other similar title or prizes.*

PLACE:	*(RAJIW KUMAR)*
DATE:	*REG.NO.:510731861*

EXAMINER’S CERTIFICATION

The project report of Mr. RAJIW KUMAR ***“RAILWAY RESERVATION SYSTEM”*** *is approved and is acceptable in quality and form.*

Internal Examiner External Examiners

UNIVERSITY STUDY CENTRE CERTIFICATE

This is to certify that the project report entitled ***"RAILWAY RESERVATION SYSTEM"*** *submitted in partial fulfillment of the requirements for the degree of Bachelor of science in information technology of Sikkim-Manipal University of Health, Medical and technological sciences*

Mr. RAJIW KUMAR has worked under my supervision and guidance and that no part of this report has been submitted for the award of any other degree, diploma, fellowship or other similar titles or prizes and that the work has not been published in any journal or magazine.

Reg. No.: 510731861 ***CERTIFIED***

ACKNOWLEDGEMENT

I am grateful to all those, who made this project possible and helped me to complete it within time.

Last but not least I extend my thanks to all staff members and friends for their help and invaluable support to me.

TABLE OF CONTENTS

Annexure – III

List of Tables

Annexure – IV

List of Figures

List of Abbreviations

SDM	Systems Development Method
IT	Information Technology
PERT	Program Evaluation Review Technique
C++	C Plus Plus
OPP	Object Oriented Programming
RTTI	Runtime Type Identification
EOL	End Of Line
DFD	Data Flow Diagram
ER	Entity Relationship
ERD	Entity Relationship Diagram

INTRODUCTION

INTRODUCTION

Today, IT industry is an industry which brings changes in the working environment of our society in a convenient and efficient way. IT industry is flourishing and has its scope in relatively all the fields. We are in an era of change and can do away with all the obstacles of the current prevailing system. The existing manual system of Railways is very difficult to handle as the various operations like reservation of tickets, cancellation of ticket and queries generated by the passengers take enough time in executing. While studying and analyzing the working of this system, there were many problems that occurred. These problems are: -

- Manual system is a time consuming system, i.e. manual system takes a large amount of time for processing queries generated by the passengers.
- There is a possibility of duplicity of data in manual system. Due to this, there is also a possibility of inconsistency.
- Manual system is difficult to operate.
- It is more costly.
- It uses more man power.

OBJECTIVE

OBJECTIVES OF THE PROJECT

The main objective of the **RAILWAY RESERVATION SYSTEM** is to perform all the functions or operations accurately and correctly. It overcomes all the problems that we have in our existing system. Our existing system is as follows:

- **Duplication of data: -** Duplication of data means that the same record was repeated a number of times. Due to this, our database becomes lengthy and difficult to maintain. If there is duplication of data, then there is a possibility of inconsistency of data. Due to the above problem, it is very difficult to handle the database. For example, if a passenger generates a query and there is duplication of data, it is impossible to processing the passenger‘s query.
- **Update Problem**:- The updation problem is further categorized into three category:-

- **Insertion Problem**: - . It doesn’t enforce any constraint over the user while inputting data in the table. The user may enter irrelevant, false information in the table. That result inaccuracy, inconsistency of record. In our new computerized record keeping system, domains are specified that enforces the user to input valid data. For example, the user can enter the incorrect data in the table, which may lead to irrelevant or false data.

- **Manipulation Problem**: - In the existing system, there is a possibility of duplication of records. The manipulation of record is successful if it is done to all the duplicate records. For example, if a record is entered in the table twice, and there is only one record is updated and then there is a problem in query handling.

- **Deletion problem**: - If the customer want to cancel his/her booking, want to returns the ticket than it is deleted from the confirm table and stored in the cancellation table. And if the table having duplicate records than all the duplicate records have been deleted and the practical implementation of this is not possible.

- Apart of duplication problem and updation problem, there is another problem that is, our existing system is very slow. The process of inserting, updating or deleting of records is limited to speed of operator. And maintaining registers or papers is very difficult there is need of extra storage space and person who maintain the registers of records. There are more chances of loses of records during handling of records.

- **Computerized system**: - The problems associated with the existing system, approach to develop a new computerized system that overcomes all the problems associated with our existing manual system. Apart from this, The system has a user friendly environment where the end user can perform all the activities that are related to online reservation system such as reservation of tickets, cancellation of tickets, enquiries etc.
The proposed system has the following objectives to be achieved.

✓ User Friendly Environment.

✓ Less Space.

✓ Fast Retrieval.

✓ Easy to Operate.

✓ Accuracy.

✓ Report generation etc

PROBLEM SELECTION

PROBLEM SELECTION

The existing system in the Railway is manual where all the work is done manually by entering the records in the various files and folders.

It is quite cumbersome to proceed through the manual system because Reservations, Query related rail schedule as well as Cancellation of tickets takes ample time. Maintaining these files requires lot of space as well lot of man power. Also there is a possibility of duplicacy of data and data inconsistency.

Therefore the requirements of the project are as follows:

- Greater storage capacity.
- Greater processing speed of input and output, i.e. the query and other operations completed in less and less time.
- More reliable and consistent procedure to eliminate errors in handling day to day processing.

PROJECT MONITORING SYSTEM

PROJECT MONITORING SYSTEM

After studying my project and analyzing it, I decided to precede it using the SPIRAL model. I used the spiral model because the spiral model promotes quality assurance through prototyping at each stage in systems development.

SPIRAL MODEL: - The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model, it is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for large, expensive and complicated projects.

The steps in the spiral model can be generalized as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
2. A preliminary design is created for the new system.
3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4. A second prototype is evolved by a fourfold procedure:
    1. evaluating the first prototype in terms of its strengths, weaknesses, and risks;
    2. defining the requirements of the second prototype;
    3. planning and designing the second prototype;
    4. Constructing and testing the second prototype.

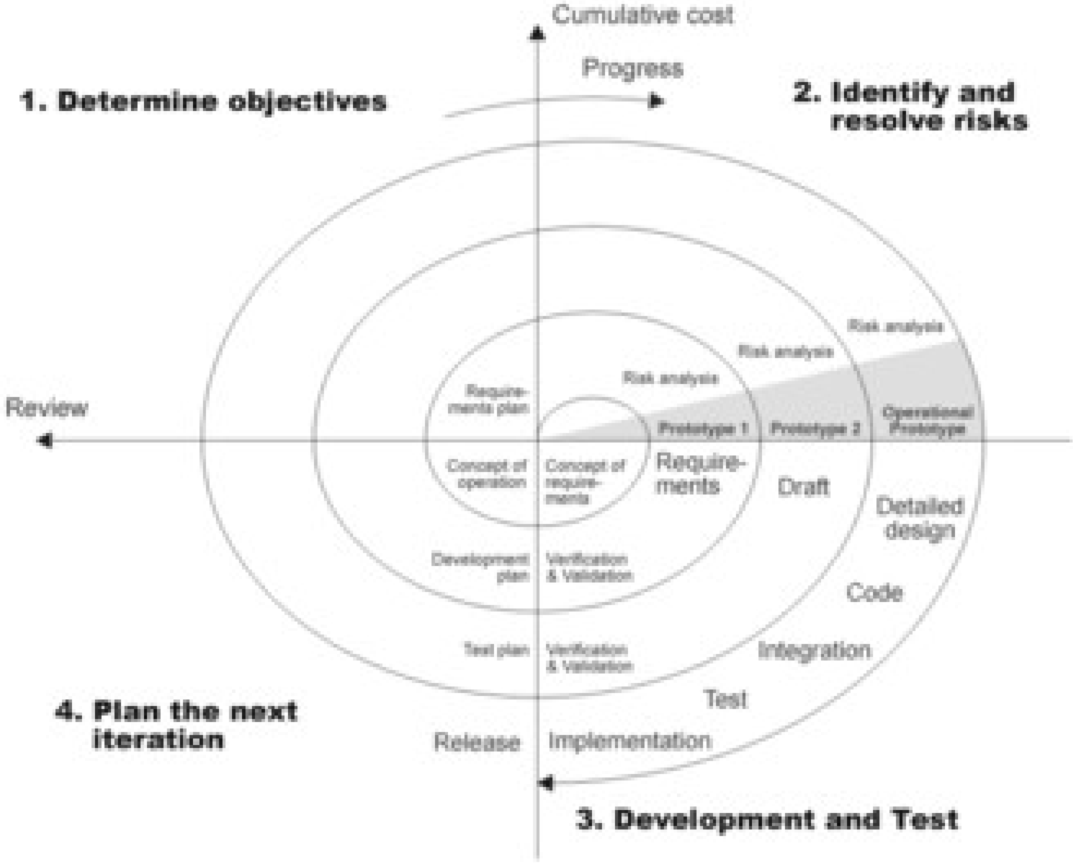


Figure 1: - SPIRAL MODEL

As I am using this model, I evaluated the duration of this project will take to get completed finally. For this I used PERT chart.

PERT CHART: -

A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. PERT stands for *Program Evaluation Review Technique*. A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered *nodes* (either circles or rectangles) representing events, or milestones in the project linked by labeled *vectors* (directional lines) representing tasks in the project.

The direction of the arrows on the lines indicates the sequence of tasks.

→ Arrow is used to represents task.

○ Circle represents the beginning or completion of task.

The PERT chart process includes following steps: -

1) Identifies the specific activities and milestone.
2) Determine the proper sequence of the activities.
3) Construct a network diagram.
4) Estimate the time required for each activity.
5) Determine the critical path.
6) Update the PERT chart as the process progresses.

PERT CHART

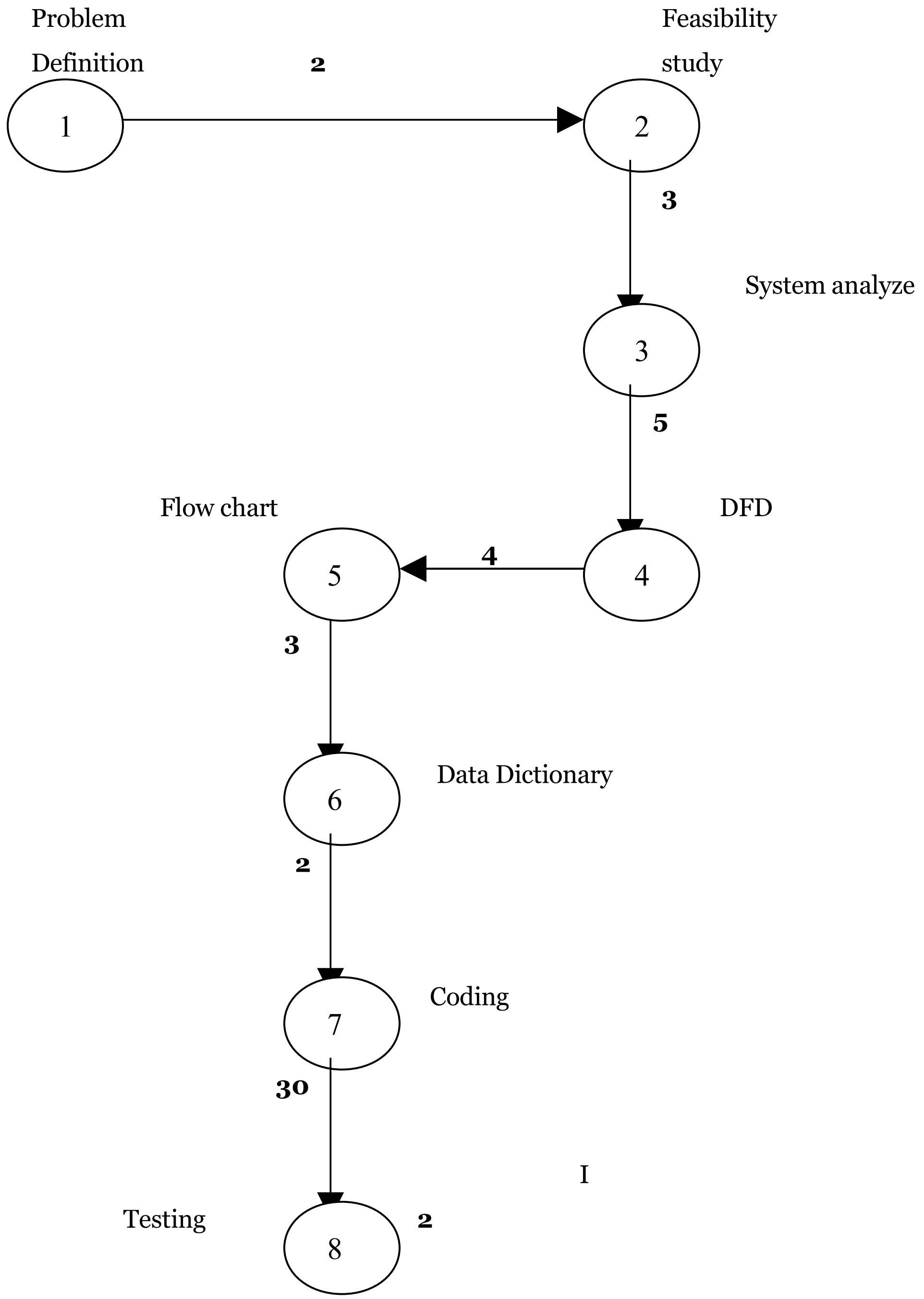


Figure 2: - PERT chart

Actual Process

S.No	Phase	Time Estimate in days	Actual Time
1.	Problem Definition	2	3
2.	Feasibility study	3	6
3.	System analysis and design		
3.1.	Flow Chart	3	4
3.2.	DFD	4	5
3.3.	Data Dictionary	2	3
3.4.	Complete documentation	8	10
4.	Coding	30	8
5.	Testing	2	4

Total Number of Hours= **54days* 5 hours**

= 270 hrs.

<u>Methodology Used</u>: -

The methodology I will be using in my project is C++ with OOP's concept.

C++ ("C plus plus") is a general-purpose programming language. It is regarded as a middle-level language, as it comprises a combination of both high-level and low-level language features. It was developed by Bjarne Stroustrup in 1979 at Bell Labs as an enhancement to the C programming language and originally named "C with Classes". It was renamed to C++ in 1983. It is a statically typed free-form multi-paradigm language supporting procedural programming, data abstraction, object-oriented programming, generic programming and Runtime Type Identification (RTTI). Since the 1990's, C++ has been one of the most popular commercial programming languages.

History of C++: -

The c++ programming language was developed at AT&T Bell Laboratories in the early 1980's by Bjarne Stroustrup. Stroustrup began work on *C with Classes* in 1979. The idea of creating a new language originated from Stroustrup's experience in programming for his Ph.D. thesis. Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level to be suitable for large software development. When Stroustrup started working in AT&T Bell Labs, he had the problem of analyzing the UNIX kernel with respect to distributed computing. Remembering his Ph.D. experience, Stroustrup set out to enhance the C language with Simula-like features. C was chosen because it was general-purpose, fast, portable and widely used.

As the C++ language evolved, a standard library also evolved with it. The first addition to the C++ standard library was the stream I/O library which provided facilities to replace the traditional C functions such as printf and scanf. Later, among the most significant additions to the standard library, was the Standard Template Library.

C++ continues to be used and is still one of the preferred programming languages to develop professional applications. The language has gone from being mostly Western, to attracting programmers from all over the world.

<u>Features introduced in C++:</u> -

Compared to the C language, C++ introduced extra features, including declarations as statement, function-like casts, new/ delete, reference type, inline functions, default arguments, function overloading, namespaces, classes(including all class-related features such as inheritance, member function, virtual function, abstract classes, and constructors), operator overloading, templates, the :: operator, exception handling, and runtime type identification.

OOP's concept with c++ language acts as a front-end as well as back-end. Back-end deals with the file handling concept of the c++. Basically the file handling concept uses data files to store the data. Thus the data files are the files that store data pertaining to a specific application, for later use. The data files can be stored in two ways:

1) **Text files**: - A text file stores information in ASCII characters. In text files, each line of text is terminated, with a special character known as EOL (End of Line) character.
2) **Binary**: - A binary file is just a file that contains information in the same format in which the information is held in memory. In binary file, there is no delimiter for a line. As a result, binary files are faster and easier for a program to read and write than text files.

To open a file, there are various opening modes. The file mode describes how a file is to be used: to read from it, to write to it, to append it, and so on. The various files modes are as follows: -

OPENING MODES	DESCRIPTION
ios::in	Open file in input mode for reading.
ios::out	Open file in output mode for writing.
ios::app	Open file in output mode for writing the new content at the end of the file without removing the previous contents of the files.
ios::ate	Open file in output mode for writing the new content at the current position of the pointer without removing the previous contents of the files.
ios::trunc	Open file in output mode for writing the new content at the beginning of the file removing the previous contents of the files.
ios::nocreate	The file is not created. The operation takes place in existing file. If the file is not found an error occurs.
ios::noreplace	The existing file is not overwritten. The operations take place on existing file. If the file is not found an error occur.
ios::binary	Opens the file in binary mode reading not a character but reading/writing whatever the binary value is stored in the file.

Basic concepts of Object Oriented Programming: -

The basic concepts of Object Oriented Programming are as follows: -

1. **Objects: -** Objects are the basic runtime entities in an object oriented system.
2. **Classes: -** Classes are the user defined data type and behave like a built in type of a programming language.
3. **Data abstraction: -** It means consider only those attributes or aspects of objects which are important for a particular class.
4. **Data encapsulation: -** The purpose of wrapping similar objects in a single unit is called encapsulation. It is done to secure objects from external world and misuse.
5. **Inheritance: -** Process of acquiring properties of one class to another class without physical declaration of variables is called inheritance.
6. **Polymorphism: -** When a single object or method behaves different in different location called polymorphism. It is done by using two types of overloading –
   - i. Function overloading
   - ii. Operator overloading
7. **Dynamic binding: -** Binding refers to the linking of a procedural call to the code to the executed in response to the call.
8. **Message passing: -** An object oriented program consists of a set of objects that communicate with each other. The process of communication between the objects is known as message passing.

In my project “Railway Reservation System”, I, tried to use or implement all these features of OOPS up to my level best.

SYSTEM STUDY

SYSTEM STUDY

System Study is to measure that how beneficial or practical the development of an information system will be to an organization. It is also known as the Feasibility study.

FEASIBILITY STUDY

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, it is important that it is conducted completely and that no fundamental errors of judgement are made.

The purpose of feasibility study is to determine whether the requested project is successfully realizable. There are three aspects of feasibility study, namely

(a) Technical feasibility

(b) Economic feasibility

(c) Operational feasibility

- ## Operational feasibility

Operational feasibility must determines how the proposed system will fit in with the current operations and what, if any, job restructuring and retraining will be needed to implement the system.

The analyst should determine:

- Whether the system can be used if it is developed and implemented?
- Will there be resistance from users that will cripple the possible application benefit?

- How well the solution will work in the railway management and how the end-users and staff members of the railway feel about the system. This people oriented test measures the urgency of the problem or the acceptability of a solution i.e. Is the problem worth solving?

PIECE is used as the basis for analyzing the urgency of problem or the effectiveness of a solution.

PIECE stands for:

P – Performance

I – Information

E – Economics

C – Control

E – Efficiency

- ## Technical feasibility

The purpose of assessing technical feasibility is to gain an understanding of the organization's ability to construct the proposed system. Technical feasibility determines whether the technology needed for the proposed system is available and how it can be integrated with in the organization. Technical evaluation must also assess whether the staff have the technical expertise to assimilate and use the new technology.

➢ Whether the project can be carried out with the existing equipments?

The existing system is manual system where all the work is done manually. So this project is new technology.

The technical feasibility centers on the existing computer system and to what extent it can support the proposed system.

- **Economic feasibility**

The purpose of assessing economic feasibility is to identify the financial benefits and costs associated with the development project. Economic feasibility is often known as *cost-benefits analysis.*

To carry out an economic feasibility study, it is necessary to estimate actual money values against any purchase or activities needed to implement the project. It is also necessary to assess money value against any benefits that will accrue from a new system created by the project.

There are mainly two categories/analysis to determine economic feasibility: -

- Cost category
- Benefit category

COST-BENEFITS ANALYSIS

Cost Categories

Equipment cost	It includes various items of computing equipment associated with the work.
Operating cost	It includes the expense to run the system. Operating cost depends on the amount of time taken for a process i.e., it includes the cost associated with the day to day operation of the system.
Personnel cost	It includes the salaries and wages of analyst, programmers, operators, consultants etc. Salary may be on hourly basis or the entire salary for the duration of the project.
Material cost	It includes cost of stationary, paper, ribbons, floppies, CD's etc.
Conversion cost	It includes that of designing new forms and procedures, expenditure to prepare the project for using the new system.

Benefit Categories

Performance	The criteria emphasize whether the new system has resulted in a marked improvement in the accuracy in access to information. Does the new system provide easier and secure access to the authorized user.
Minimizing cost	If the new system is efficient with minimum error, reduction of staff is a benefit that should be measured and included in cost benefit analysis

Software and Hardware Requirement Specification

Hardware: -

- Intel Pentium III processor
- Minimum 128 MB RAM
- Recommended 256 MB RAM
- Minimum hard disk 40GB

Software: -

- Windows operating system
- C++ complier compatible with the Windows OS

SYSTEM ANALYSIS

SYSTEM ANALYSIS

It is the most important phase of the system development cycle.The analysis phase is the detailed understanding of all important facts of the business area under investigation. The relationship of the various system components among themselves and with environment are studied and understood. This requires data collection from a variety of sources. Analysis involves deep study of the task. System analysis is the process of gathering and interpreting facts diagnose the problems, defining the goals, design the constraints and using the facts improve the system. The aim of system analysis to identify the boundaries of the system and their sub-systems and interface between sub-systems and systems.

- **INFORMATION REQUIRED: -**
    - What is being done in the project?
    - How is it being done?
    - What are the problems that may arise?
    - If a problem arises, how will it be solved?
    - What could cause such a problem?

For this I have collected the data by consulting various departments, persons working in different departments. Then I discussed various issues with my project incharge and other staff members. They helped me to find the requirements of my project and also help me in identify the reasons for the problem that has occurred and the preventive measures to avoid them.

Hence from all the discussion, I perceived the knowledge that what is going in railway management system and how it is maintained. The discussion with them also helped me in extracting the ideas that what actually is required in the system.

SYSTEM DESIGN

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 & entity-relationship diagrams.
- Defines the data requirements with a data dictionary.
- Writes program specifications.
- Identifies and orders any hardware and software that the system design phase would need.

DATA FLOW DIAGRAMS

Data flow diagrams provide a logical model of the system and show the flow of data and the flow of logic involved. The Data Flow Diagram (DFD) clarifies system requirements and identifies major transformations that will become programs in system design. With only four symbols, you can use data flow diagrams to represent both physical and logical information systems. Data flow diagrams (DFD's) are not as good as flowcharts for not very useful for depicting purely logical information flows. In fact, flowcharting has been criticized by proponents of structured analysis and structured design because it is too physically oriented. Data flow diagram shows how data travels from one point to another point in the diagram. The flow is shown as an arrowed line with the arrowhead showing the direction of flow.

The data flow is given a simple and meaningful descriptive name such as order details or customer payments etc. The data flow may move from an external entity to a process, from one process to another process and from one process to external entity. The Data Flow Diagram (DFD) clarifies system requirements and identifies major transformations that will become programs in system design. It is the starting point of system design that decomposes the requirements specifications down to the lowest level of detail.

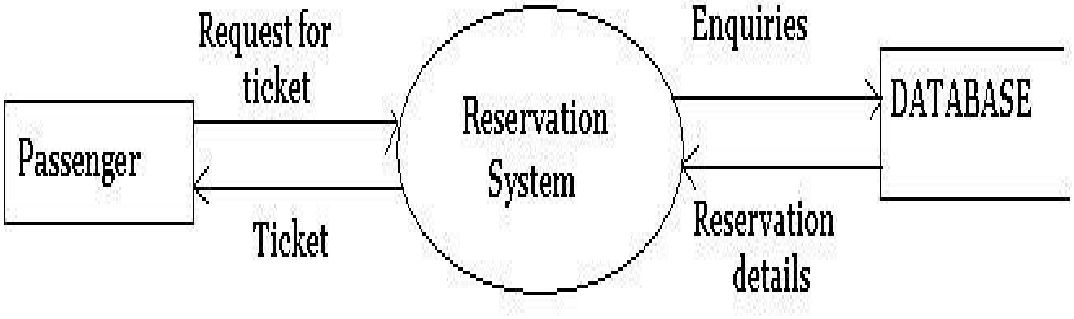


Figure 3: - Data Flow Diagram

ZERO -LEVEL DFD

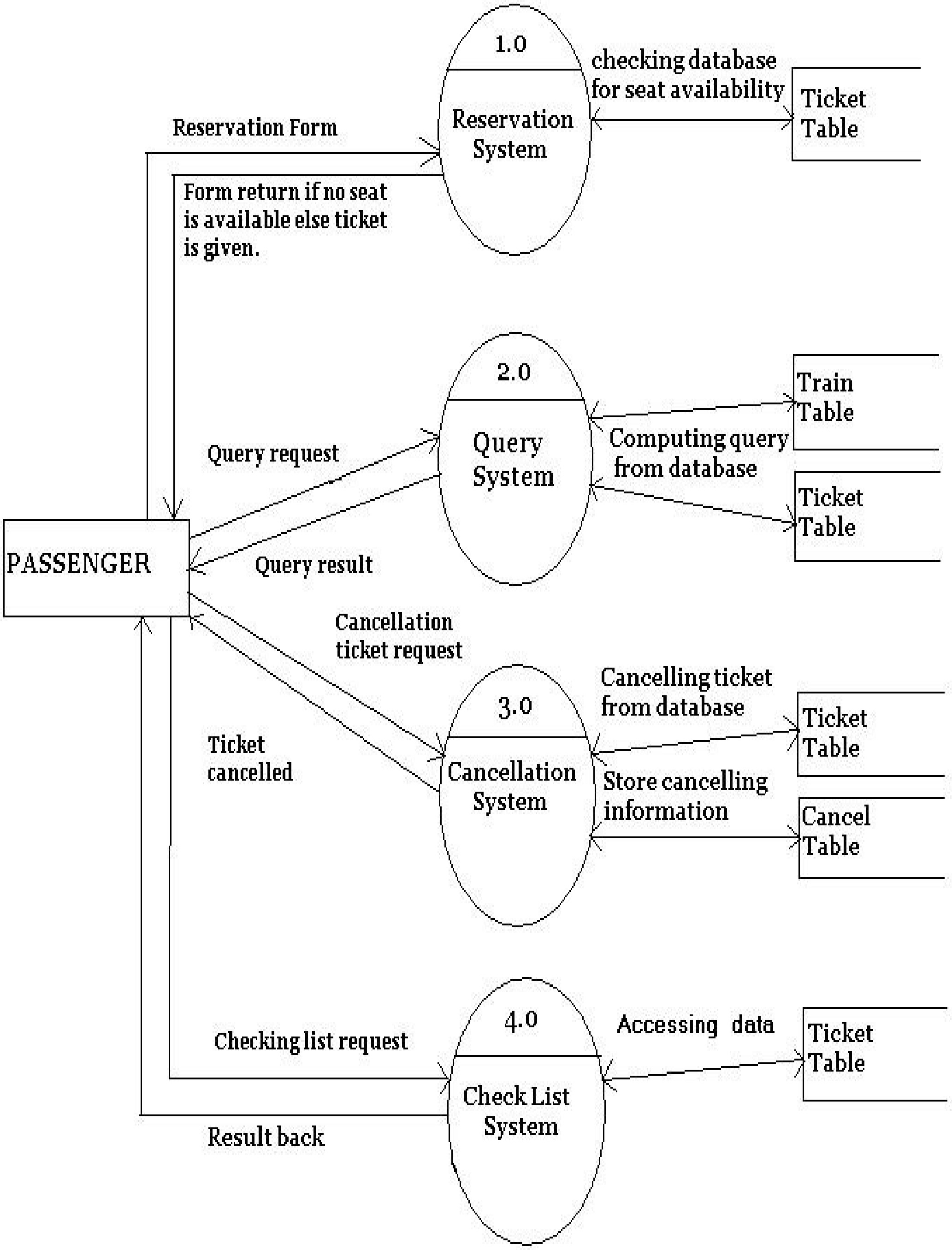


Figure 4: - Zero Level DFD for reservation

1 -LEVEL DFD

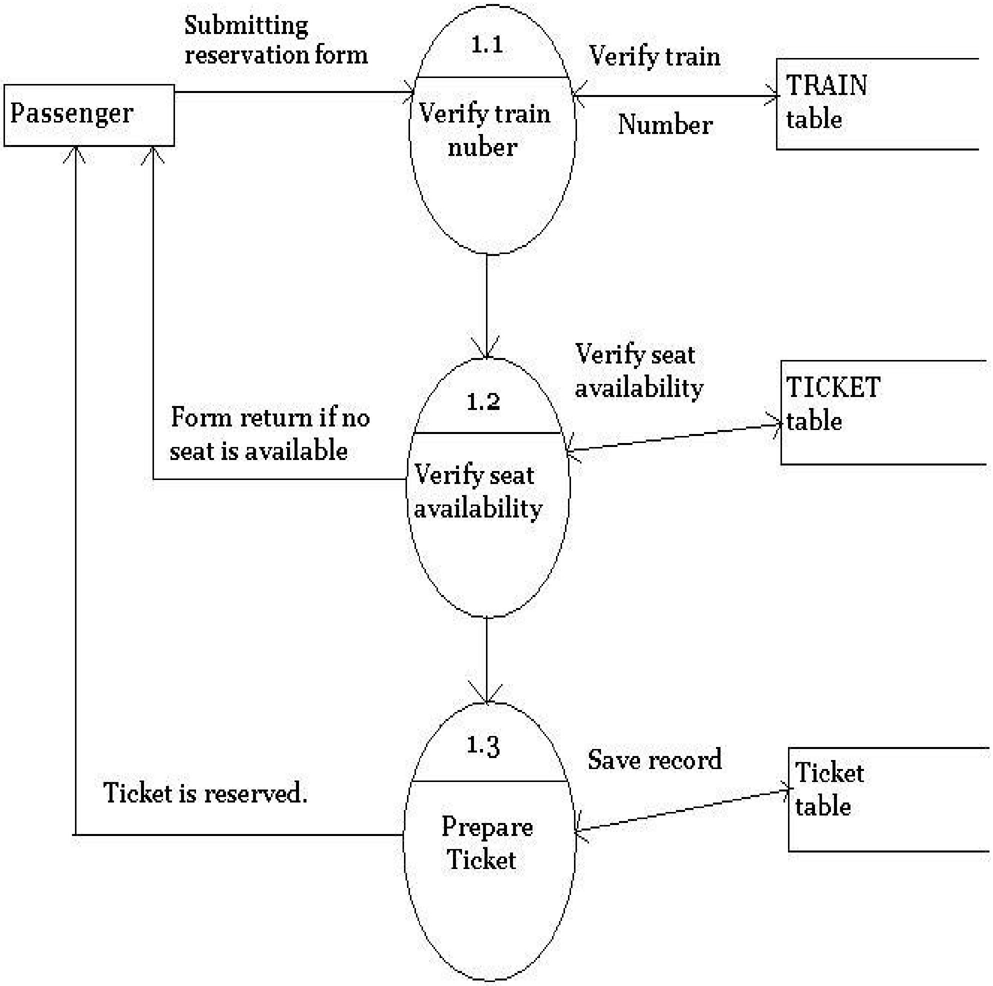


Figure 5: - Level 1 DFD for Process1.0

1 -LEVEL DFD

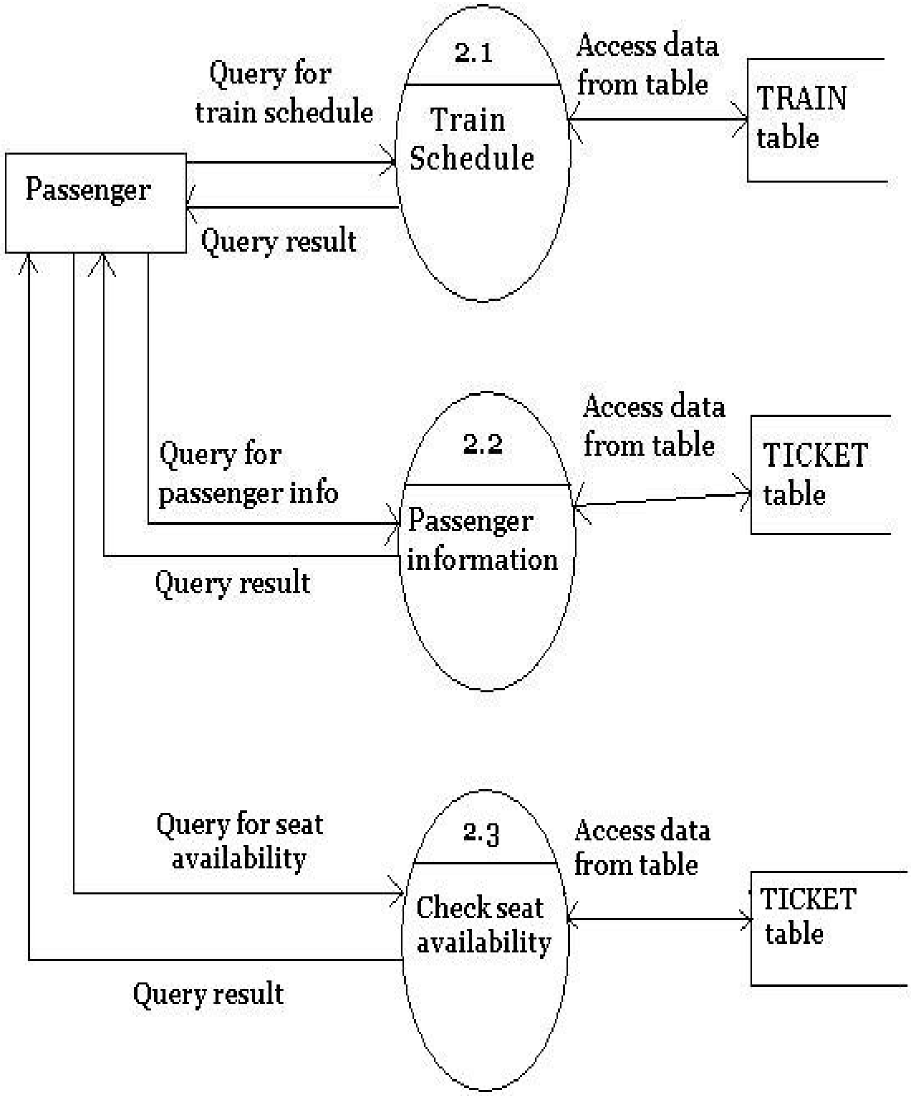


Figure 6: - Level 1 DFD for Process2.0

1 -LEVEL DFD

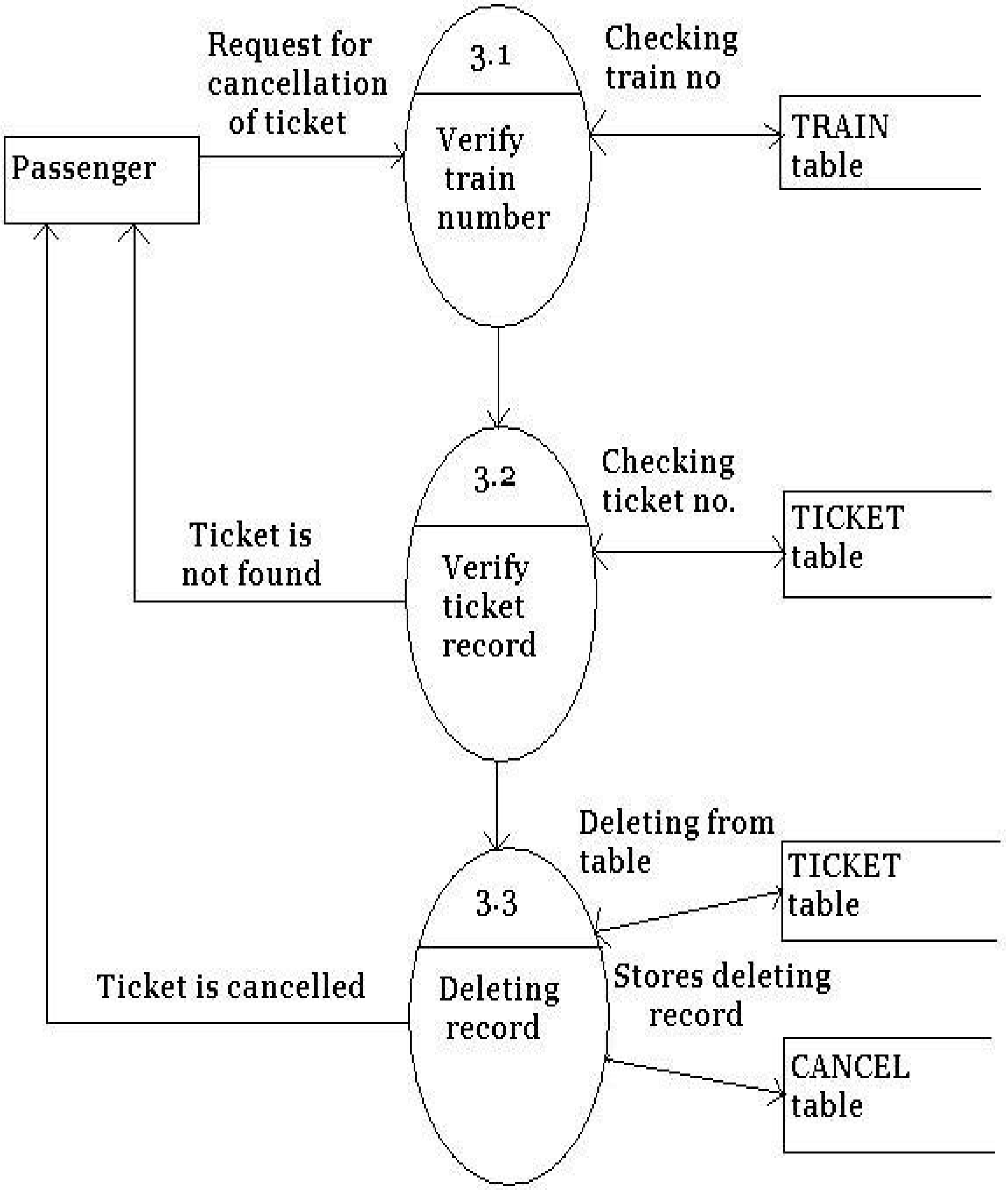


Figure 7: - Level 1 DFD for Process3.0

1 -LEVEL DFD

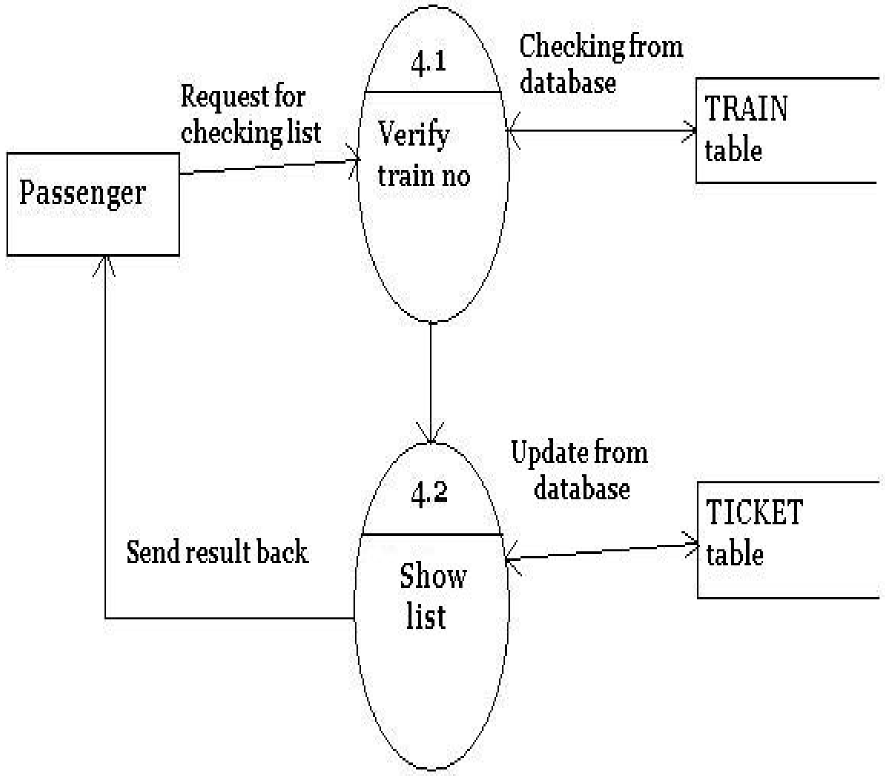


Figure 8: - Level 1 DFD for Process4.0

FLOW CHART (MENU)

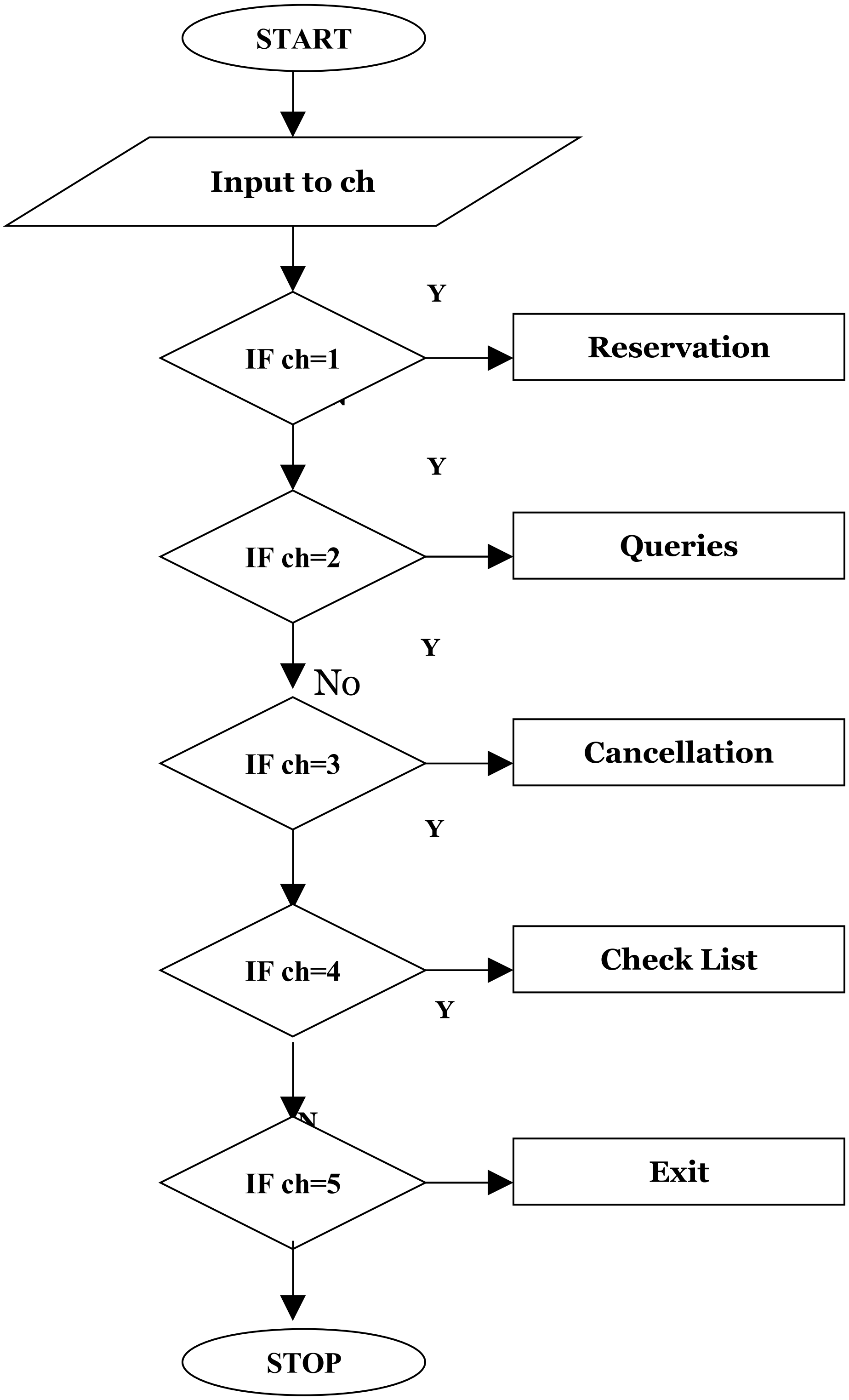


Figure 9: - Flowchart of main menu

Flow chart of RESERVATION

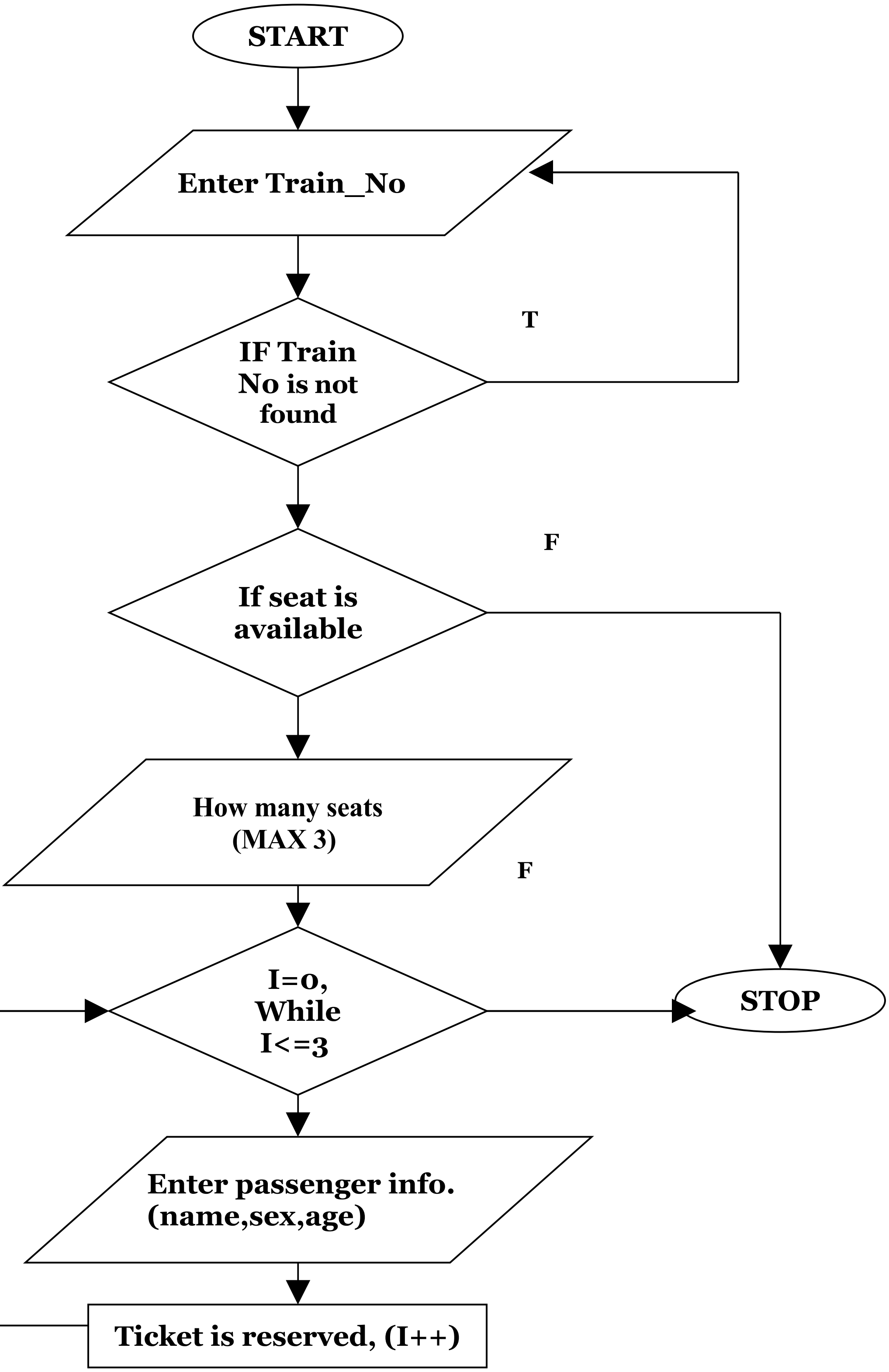


Figure 10: - Flowchart of RESERVATION option

Flow chart of QUERIES

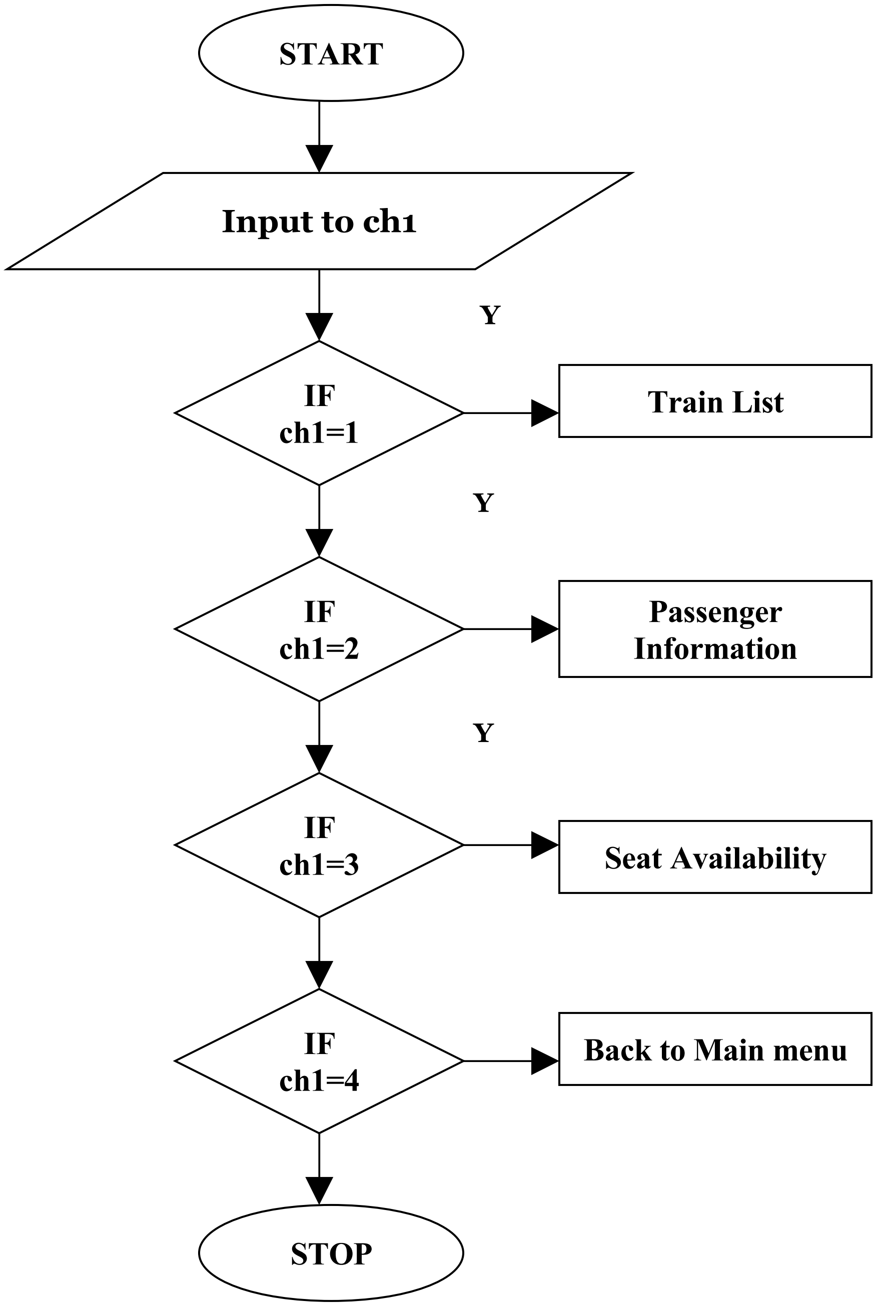


Figure 11: - Flowchart of QUERIES option

Flow chart of CANCELLATION

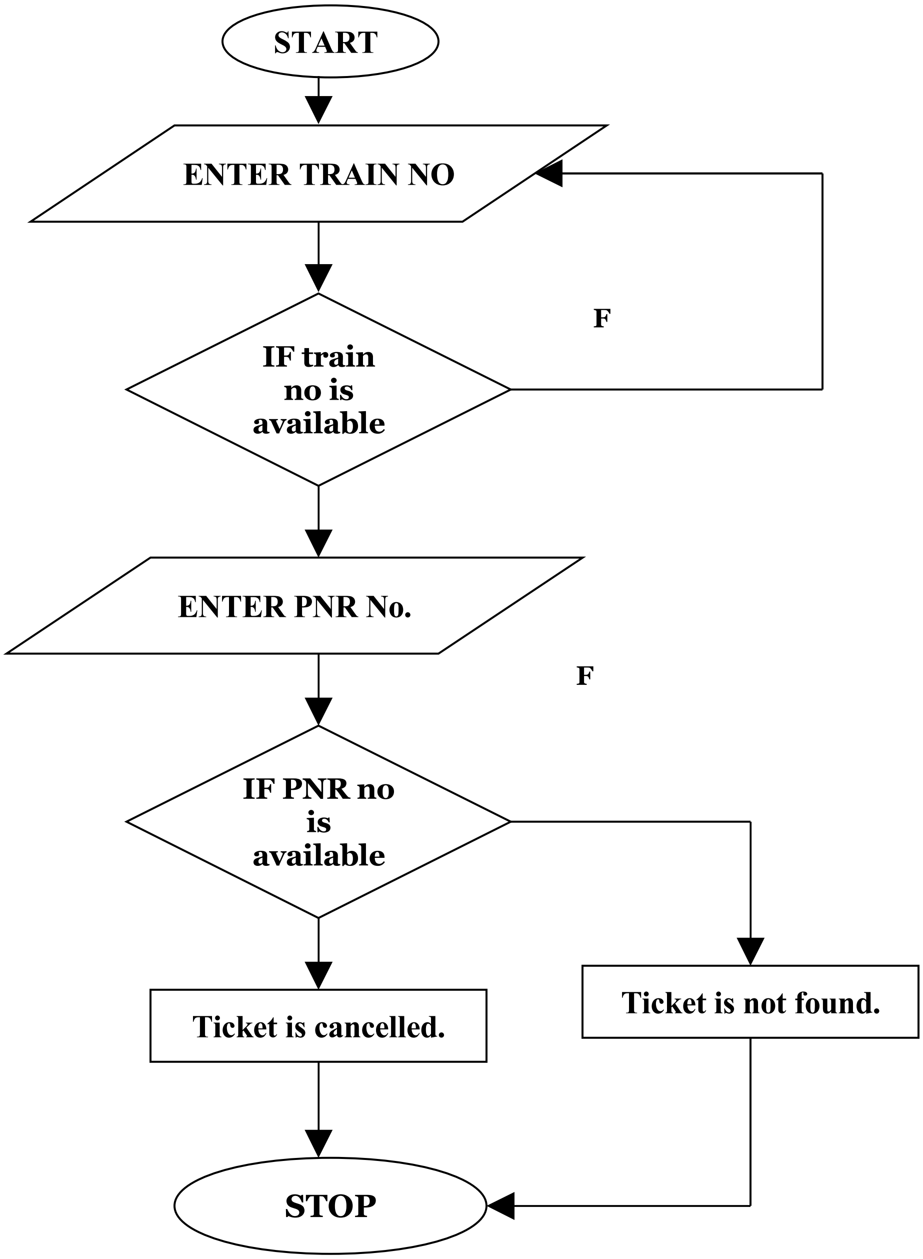


Figure 12: - Flowchart of CANCELLATION option

Flow chart of CHECK LIST

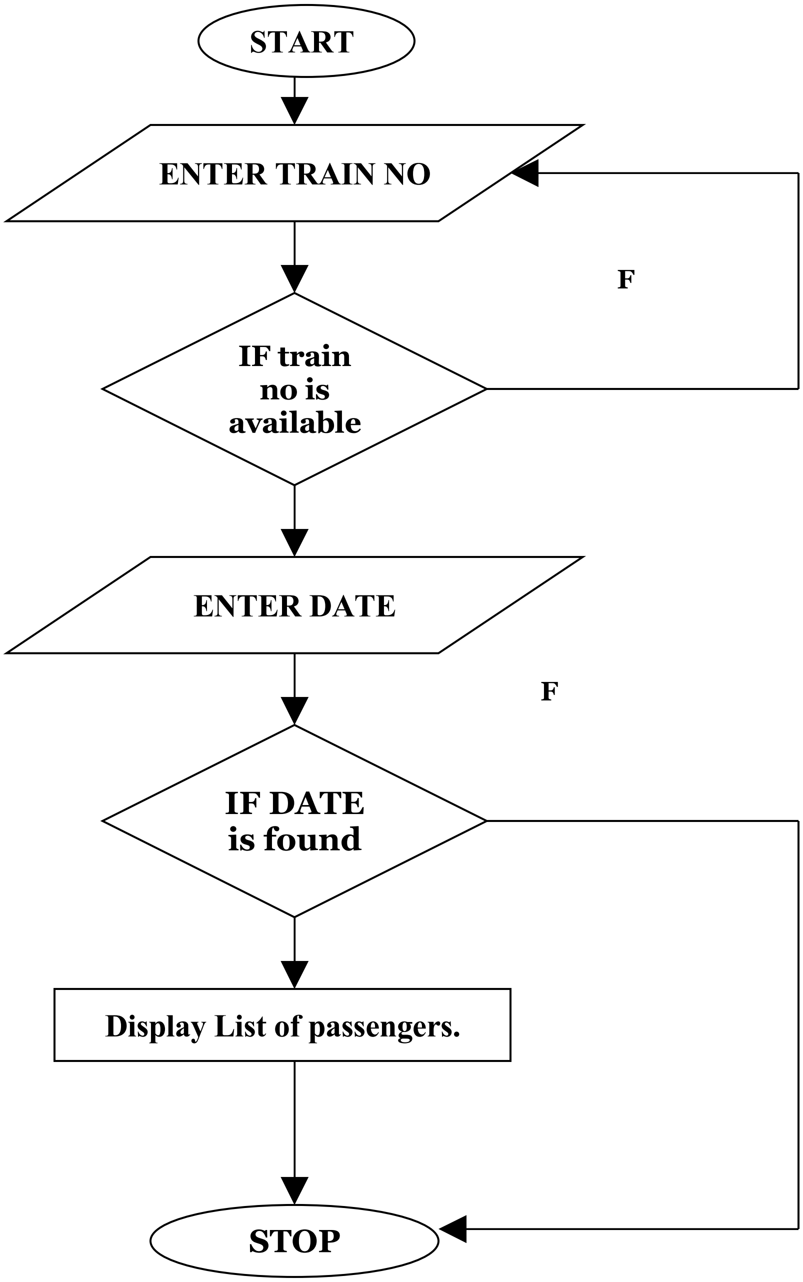


Figure 13: - Flowchart of CHECK LIST option

ENTITY-RELATIONSHIP DIAGRAM

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. The other tool for problem analysis is the Entity-Relationship Diagram, often called as ER Diagram. It is a detailed logical representation of the data for an organization and uses three main constructs, i.e. Data Entities, Relationships and their Associated Attributes.

1. **Entities: -** An entity is a person, place, thing or event of interest to the organization and about which data are captured, stored or processed. For example, an employee is an entity.

2. **Relationship: -** An association of several entities in a entity-relation model is called relationship.

3. **Attributes:** Each entity type has a set of attributes associates with it. An attribute is a property of characteristics of an entity that is of interest to the organization. We use an initial Capital Letters, followed by Lowercase Letters in naming an attribute

ERD
[Entity Relationship Diagram]

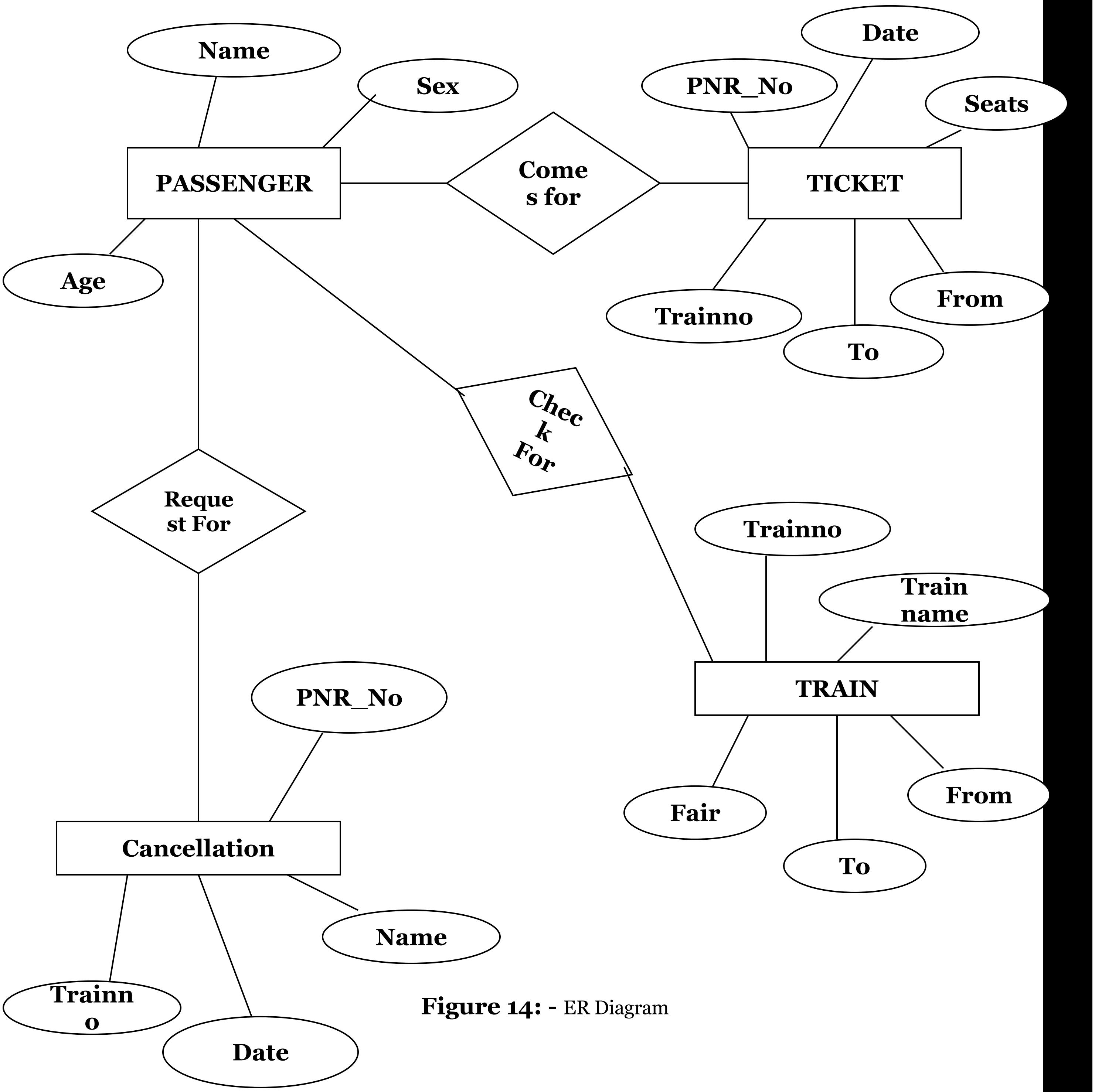


Figure 14: - ER Diagram

After brief analysis of DFD's and E-R diagrams, i found that there are basically three tables, namely, "Train.dat", "Ticket.dat" and "Cancel.dat".

Train.dat: - This table records the information about the train. For example, train number, train name, No. of seats in the train, and source & destination address of the train.

Ticket.dat: - This table records the information about the passenger's details and ticket details. For example, it records the PNR no., train number, date of journey, name of passenger, passenger age & sex, source and destination address and fair of ticket.

The tables of the project "Railway reservation system" are as follows: -

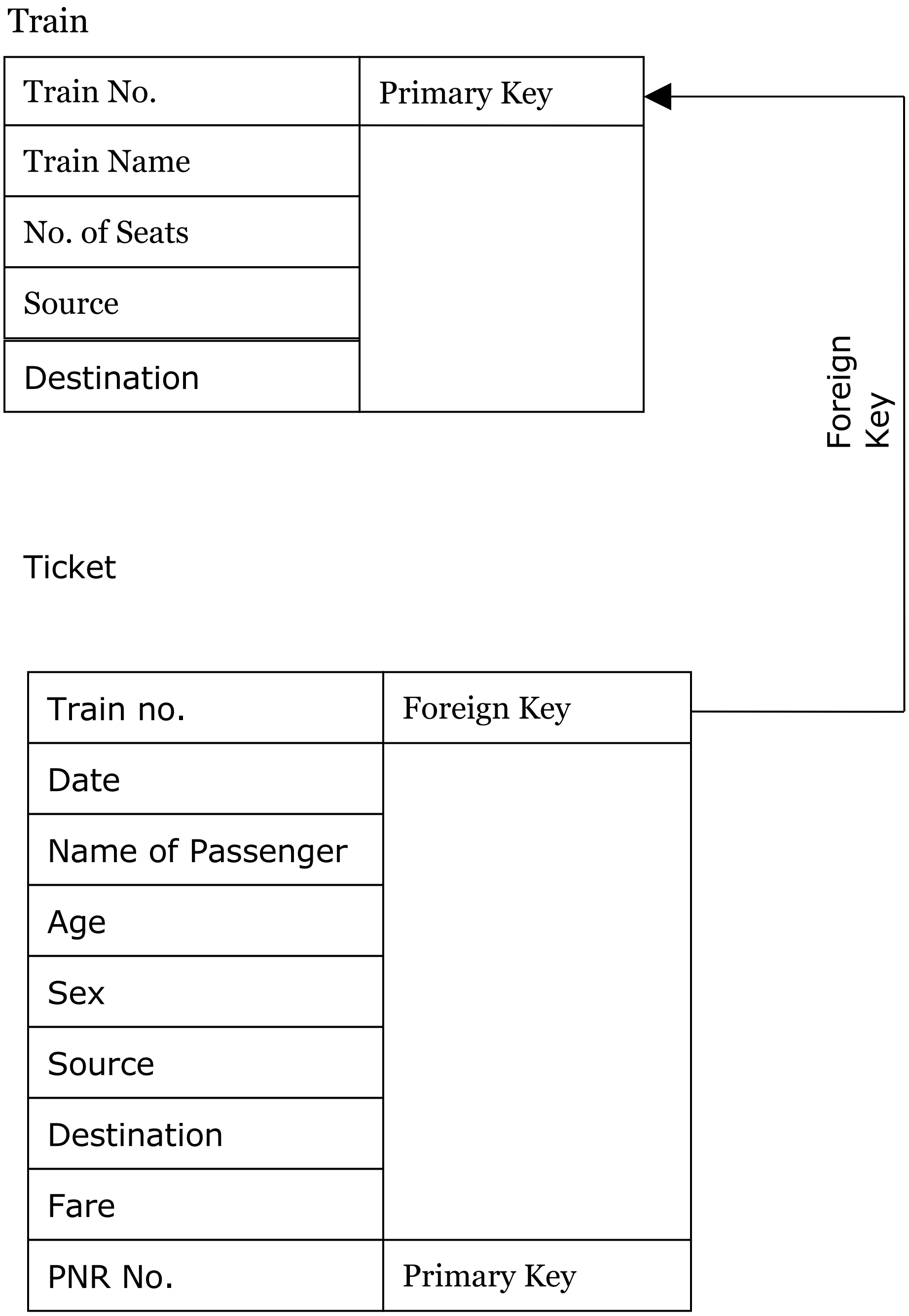
Train.dat

Train no.	Int
Train name	Char
No. of seats	Int
Source	Char
Destination	Char

Ticket.dat

Train no.	Int
Date of journey	Int
Name of passenger	Char
Age	Int
Sex	Char
Source	Char
Destination	Char
PNR no.	Int

There are two tables in my project as discussed earlier, the relationship between these two tables is as follows: -



Train

Train No.	Primary Key
Train Name	
No. of Seats	
Source	
Destination	

Ticket

Train no.	Foreign Key
Date	
Name of Passenger	
Age	
Sex	
Source	
Destination	
Fare	
PNR No.	Primary Key

DOCUMENTATION

DOCUMENTATION

```
/*                               RAILWAY
                                    RESERVATION
                                             SYSTEM

                                 Made by:-  Pradeep Kumar Patel
                                 Course:- BCA (Bachelor in computer application)
                                 */



#include<iostream.h>
#include<dos.h>
#include<conio.h>
#include<string.h>
#include<stdio.h>
#include<fstream.h>
#include<ctype.h>
#include<process.h>
#include<graphics.h>



            char tno[5];
            char pfrom[15],pto[15];
            int ts;
            int totaltickets=0;
/**************************************************************************
           THIS CLASS CONTAINS ALL THE FUNCTIONS RELATED TO PROJECT MENU'S
                                 AND BACKGROUND DESIGN
**************************************************************************/
```

```
class railway
{
        public :
        void sketch();
        void menu();
        void distext();
        void query();
        void train_design();
};
/************************************************************************************
 //   THIS CLASS CONTAINS ALL THE FUNCTIONS RELATED TO DATE
************************************************************************************/
class DATE
{
        public:
        void INCREASE_DATE(int,int,int);
        void CURRENT_DATE();
        int day,month,year;
};
/************************************************************************************
//   THIS CLASS CONTAINS ALL THE FUNCTIONS RELATED TO TRAIN
************************************************************************************/
class TRAIN
{
        public:
        char train_no[5],train_name[20],train_source[20];
        int train_seats,fair;
        char train_desti[20];
        int check(int r);
        void DISPLAY_DESIGN();
        void ADD_TRAIN(char[],char[],int,char[],char[],int);
        void ADDITION();
        void TRAIN_LIST();
        int FOUND_TRAIN(char[]);
};
/*******************************************************************************
        THIS FUNCTION CONTAINS ALL THE FUNCTIONS RELATED TO TICKET LIKE
                  RESERVATION, CANCELATION ETC.
*******************************************************************************/
class TICKET
{
        int pnr_no,kms;
        char name[26],sex,tr_no[5],t_name[30];
        int age,fair;
        int dd,mm,yy;
        char from[15],to[15];
        public:
        void RESERVATION();
        void CANCELLATION();
        void CHECK_LIST();
        void PASS_INFO();
        void AVAILABILITY();
};

                TICKET t;
/*******************************************************************************
           FUNCTION TO GET THE CURRENT SYSTEM DATE
*******************************************************************************/
void DATE::CURRENT_DATE()
{
        date d;
        getdate(&d);
        day=d.da_day;
        month=d.da_mon;
        year=d.da_year;
}
/*******************************************************************************
                FUNCTION TO EXTEND THE DATE BY ONE
*******************************************************************************/
```

```
void DATE::INCREASE_DATE(int dday,int dmonth,int dyear)
{
        static int m[] = {31,29,31,30,31,30,31,31,30,31,30,31};
        dday++;
        if((dday>m[dmonth-1])||(dyear%4!=0 && dmonth==2 && dday>28))
        {
                dday=1;
                dmonth++;
        }
        if(dmonth>12)
        {
                dmonth=1;
                dyear++;
        }
        day = dday;
        month = dmonth;
        year = dyear;
}
/*************************************************************************************
                Function to create Front background page.
*************************************************************************************/
void railway :: sketch()
{
        setbkcolor(BLACK);
        setcolor(YELLOW);
        settextstyle(11, HORIZ_DIR, 3);
        outtextxy(0,468," Copyright @laxman thakur");
        outtextxy(560,460, "I/R");
        outtextxy(535,468, "Reservation");
}
/*************************************************************************************
                function to display the running train
************************************************************************************/
void TRAIN :: DISPLAY_DESIGN()
{
```

```
railway rail;
rail.sketch();
setcolor(YELLOW);
arc(292,102,0,360,92);
arc(292,102,0,360,68);
gotoxy(27,6);
cout<<"I";
gotoxy(29,4);
cout<<"N";
gotoxy(34,2);
cout<<"D";
gotoxy(40,2);
cout<<"I";
gotoxy(45,4);
cout<<"A";
gotoxy(47,6);
cout<<"N";
rectangle(200,100,223,100);
rectangle(360,100,383,100);
gotoxy(27,8);
cout<<"R";
gotoxy(29,10);
cout<<"A";
gotoxy(31,11);
cout<<"I";
gotoxy(37,12);
cout<<"L";
gotoxy(43,11);
cout<<"W";
gotoxy(45,10);
cout<<"A";
gotoxy(47,8);
cout<<"Y";
setcolor(YELLOW);
settextstyle(7, HORIZ_DIR, 5);
```

```
        outtextxy(245,50, "I/R");
        settextstyle(1, HORIZ_DIR, 1);
        outtextxy(235,110, "RESERVATION");
        setcolor(WHITE);
        outtextxy(130,200, "COME FOR A SAFE & HAPPY JOURNEY");
        delay(3000);
        setcolor(YELLOW);
        rectangle (100, 280, 550, 350);
        delay(500);
        setcolor(WHITE);
        settextstyle(7, HORIZ_DIR, 3);
        outtextxy(240,285,"WELCOME TO");
        delay(600);
        outtextxy(123,320,"RAILWAY RESERVATION SYSTEM");
        delay(600);
        gotoxy(28,25);
        cout<<"Please wait";
        setcolor(WHITE);
        rectangle(308,386,450,398);
        delay(500);
        setcolor(RED+YELLOW);
        for(int i=1;i<138;i++)
        {
        rectangle(310+i,388,311+i,395);
        delay(30);
        }
        delay(100);

}
/**************************************************************************
        FUNCTION TO CREATE THE MAIN MENU OF THE PROJECT
**************************************************************************/
void railway::menu()
{
        cleardevice();
        int d1,m1,y1;
        DATE dt;
        dt.CURRENT_DATE();
        d1=dt.day;
        m1=dt.month;
        y1=dt.year;
        gotoxy(18,8);
        cout<<"DATE:- "<<d1<<"/"<<m1<<"/"<<y1<<endl;
        rectangle (130, 100, 450, 380);
        gotoxy(25,7);
        cout<<" RAILWAY RESERVATION SYSTEM ";
        setcolor(YELLOW);
        settextstyle(7, HORIZ_DIR, 4);
        outtextxy(210,135, "M E N U");
        setcolor(WHITE);
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(220,190, "1. Reservation");
        outtextxy(220,210, "2. Queries");
        outtextxy(220,230, "3. Cancellation");
        outtextxy(220,250, "4. Check List");
        outtextxy(220,270, "5. Exit");
}
/*************************************************************************
        FUNCTION TO CREATE QUERY MENU
*************************************************************************/
void railway::query()
{
        rectangle (130, 100, 450, 380);
        gotoxy(25,7);
        cout<<" RAILWAY RESERVATION SYSTEM ";
        setcolor(YELLOW);
        settextstyle(1, HORIZ_DIR, 1);
        delay(1000);
        outtextxy(150, 130, "WELCOME TO RAILWAY QUERIES");
        setcolor(WHITE);
        delay(500);
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(220,190, "1. Train List");
        outtextxy(220,210, "2. Passenger Info");
        outtextxy(220,230, "3. Seat Availability");
        outtextxy(220,250, "4. Back to main nenu");
}
/************************************************************************
        THIS FUNCTION ADDS THE GIVEN DATA TO THE TRAIN FILE
************************************************************************/
void TRAIN::ADD_TRAIN(char tno[5],char tname[20],int tseats,char tsource[20],char
tdesti[20],int pfair)
{
        fstream file;
        file.open("TRAIN.dat",ios::out|ios::app);
        strcpy(train_no,tno);
        strcpy(train_name,tname);
        train_seats=tseats;
        strcpy(train_source,tsource);
        strcpy(train_desti,tdesti);
        fair=pfair;
        file.write((char *)this,sizeof(TRAIN));
        file.close();
}
/**************************************************************************
 //THIS FUNCTION GIVES THE DATA TO ADD RECORDS IN THE TRAIN FILE
**************************************************************************/
void TRAIN :: ADDITION()
{
        fstream file;
        file.open("TRAIN.DAT",ios::in);
        if(!file.fail())
                return;
        file.close();
        ADD_TRAIN("4121","RAJDHANI EXPRESS",5,"MUMBAI","GOA",550);
        ADD_TRAIN("4123","TAJ EXPRESS",1000,"PUNJAB","AGRA",750);
        ADD_TRAIN("4125","GOMTI EXPRESS",1600,"BANGALORE","DELHI",500);
        ADD_TRAIN("4127","KALKA MAIL",1000,"INDORE","JAMMU",450);
        ADD_TRAIN("4129","KALINDI EXPRESS",600,"KOLKATTA","DELHI",800);
        ADD_TRAIN("4130","JAIPUR MAIL",1200,"DELHI","JAIPUR",250);
        ADD_TRAIN("9002","BIHAR EXPRESS",2200,"DELHI","BIHAR",300);
        ADD_TRAIN("4466","HIMACHAL EXPRESS",600,"DELHI","UNA",180);
        ADD_TRAIN("9988","SHIMLA MAIL",400,"DELHI","SHIMLA",200);
        ADD_TRAIN("8884","A.P. EXPRESS",900,"DELHI","HYDERABAD",850);
}
//***********************************************************************
                //FUNTION TO DISPLAY THE TRAIN LIST
//***********************************************************************
void TRAIN::TRAIN_LIST()
{
        cleardevice();
        setcolor(YELLOW);
        rectangle(5,43,630,44);
        rectangle(5,65,630,66);
        rectangle(5,43,6,245);
        rectangle(5,245,630,246);
        rectangle(630,43,631,245);
        rectangle(80,43,81,245);
        rectangle(225,43,226,245);
        rectangle(330,43,331,245);
        rectangle(430,43,431,245);
        rectangle(540,43,541,245);
        fstream file;
        file.open("TRAIN.DAT",ios::in);
        gotoxy(30,1);
        cout<<"LIST OF THE TRAINS";
        gotoxy(30,2);
        cout<<"******************";
        gotoxy(2,4);
        cout<<"TRAIN NO.";
        gotoxy(14,4);
        cout<<"TRAIN NAME";
        gotoxy(30,4);
        cout<<"NO.OF SEATS";
        gotoxy(46,4);
        cout<<"FROM";
        gotoxy(60,4);
        cout<<"TO";
        gotoxy(70,4);
        cout<<"FAIR";
        int row=6;
        while(file.read((char *)this,sizeof(TRAIN)))
        {
                gotoxy(4,row);
                cout<<train_no;
                gotoxy(12,row);
                cout<<train_name;
                gotoxy(33,row);
                cout<<train_seats;
                gotoxy(45,row);
                cout<<train_source;
                gotoxy(57,row);
                cout<<train_desti;
                gotoxy(70,row);
                cout<<fair;
                row++;
        }
        gotoxy(1,row);
        file.close();
}
/***************************************************************************
          // THIS FUNCTION RETURNS ONE IF GIVEN TRAIN NO. FOUND
***************************************************************************/
int TRAIN:: FOUND_TRAIN(char tno[5])
{
        fstream file;
        file.open("TRAIN.DAT",ios::in);
        int found=0;
        while (file.read((char*)this,sizeof(TRAIN)))
        {
                if(!strcmp(tno,train_no))
                {
                        found=1;
                        break;
                }
        }
        file.close();
        return found;
}
/**************************************************************************
        FUNCTION TO CHECK SEAT AVAILABILITY
**************************************************************************/
void TICKET::AVAILABILITY()
{
        st1:
        clrscr();

        cleardevice();
        char tno[5];
        railway rail;
        TRAIN train;
        train.TRAIN_LIST();
        cout<<"\n\n Press X to EXIT.";
        cout<<"\n ENTER THE TRAIN NUMBER:-> ";
        cin>>tno;
        if (tno[0]=='x' || tno[0]=='X')
                return;
        if(!train.FOUND_TRAIN(tno))
        {
                settextstyle(12, HORIZ_DIR, 1);
```

```
            outtextxy(180,460, "TRAIN NUMBER IS INVALID");
            delay(1000);
            goto st1;
    }
    st2:
    clrscr();
    cleardevice();
    int d1,m1,y1,d2,m2,y2,d3,m3,y3,d4,m4,y4;
    int d,m,y;
    rectangle (130, 100, 450, 380);
    gotoxy(25,7);
    cout<<" RAILWAY RESERVATION SYSTEM ";
    DATE dt;
    dt.CURRENT_DATE();
    d1=dt.day;
    m1=dt.month;
    y1=dt.year;
    gotoxy(18,9);
    cout<<"THE TRAIN WILL WE GOING ON THE";
    gotoxy(18,10);
    cout<<"FOLLOWING DATES :-";
    gotoxy(18,12);
    cout<<"1:"<<d1<<"/"<<m1<<"/"<<y1<<endl;
    dt.INCREASE_DATE(d1,m1,y1);
    d2=dt.day;
    m2=dt.month;
    y2=dt.year;
    gotoxy(18,13);
    cout<<"2:"<<d2<<"/"<<m2<<"/"<<y2<<endl;
    dt.INCREASE_DATE(d2,m2,y2);
    d3=dt.day;
    m3=dt.month;
    y3=dt.year;
    gotoxy(18,14);
    cout<<"3:"<<d3<<"/"<<m3<<"/"<<y3<<endl;
```

```
dt.INCREASE_DATE(d3,m3,y3);
d4=dt.day;
m4=dt.month;
y4=dt.year;
gotoxy(18,15);
cout<<"4:"<<d4<<"/"<<m4<<"/"<<y4<<endl;
gotoxy(18,17);
cout<<"5:"<<"PREVIOUS MENU";
gotoxy(18,18);
cout<<"6:"<<"MAIN MENU";
int sno=0;
gotoxy(18,20);
cout<<"Please enter your choice:-> ";
cin>>sno;
if(sno<1||sno>6)
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
}
switch(sno)
{
        case 1 : d=d1;
                      m=m1;
                      y=y1;
                      break;
        case 2 : d=d2;
                      m=m2;
                      y=y2;
                      break;
        case 3 : d=d3;
                      m=m3;
                      y=y3;
                      break;
        case 4 : d=d4;
                        m=m4;
                        y=y4;
                        break;
        case 5 : goto st1;
                //break;
        case 6 : return;
                //break;
        default:break;
    }
    int count=0,total=0,seat=0,found=0;
    char name[30],fr[20],tt[20];
    fstream file;
    fstream fout;
    fout.open("TRAIN.DAT",ios::in);
    while(fout.read((char*)this,sizeof(TRAIN)))
    {
        if(strcmp(tr_no,tno))
        {
                found=1;
                seat=train.train_seats;
                strcpy(name,train.train_name);
                strcpy(fr,train.train_source);
                strcpy(tt,train.train_desti);
        }
    }
    fout.close();
    file.open("TICKET.DAT",ios::in);
    while(file.read((char*)this,sizeof(TICKET)))
    {
        if((dd==d&&mm==m&&yy==y)&&!strcmp(tr_no,tno))
        {
                count++;
        }
    }
```

```
clrscr();
cleardevice();
rectangle (130, 100, 450, 380);
gotoxy(18,8);
cout<<"DATE:- "<<d<<"/"<<m<<"/"<<y;
gotoxy(18,10);
cout<<"Train No: "<<tno;
gotoxy(18,11);
cout<<"Train Name: "<<name;
gotoxy(18,12);
cout<<"From: "<<fr;
gotoxy(18,13);
cout<<"To  : "<<tt;
rectangle (160, 250, 420, 350);
settextstyle(7, HORIZ_DIR, 1);
outtextxy(190,225, "SEAT AVAILABILITY");
rectangle (160, 290, 420, 291);
rectangle (240, 250, 241, 350);
rectangle (337, 250, 338, 350);
gotoxy(22,17);
cout<<"Total No.";
gotoxy(22,18);
cout<<"of Seats";
gotoxy(32,17);
cout<<"No.of seats";
gotoxy(33,18);
cout<<"Reserved";
gotoxy(44,17);
cout<<"Available";
gotoxy(24,20);
cout<<seat;
gotoxy(36,20);
cout<<count;
total=seat-count;
gotoxy(47,20);
```

```
         cout<<total;
         file.close();
         settextstyle(12, HORIZ_DIR, 1);
         outtextxy(180,460, "PLEASE ENTER A KEY TO CONTINUE......");
         getch();

         if(!found)
         {
                settextstyle(12, HORIZ_DIR, 1);
                outtextxy(180,460, "TICKET NO IS NOT FOUND");
                delay(1000);
                goto st2;
         }
}

//*************************************************************
                //THIS FUNCTION DISPLAYS PASSENGER INFORMATION
//*************************************************************
void TICKET ::PASS_INFO()
{
        st1:
        clrscr();
        cleardevice();
        char tno[5];
        railway rail;
        TRAIN train;
        train.TRAIN_LIST();
        cout<<"\n\n Press X to EXIT.";
        cout<<"\n ENTER THE TRAIN NUMBER:-> ";
        cin>>tno;
        if (tno[0]=='x' || tno[0]=='X')
                return;
        if(!train.FOUND_TRAIN(tno))
        {
                 settextstyle(12, HORIZ_DIR, 1);
                 outtextxy(180,460, "TRAIN NUMBER IS INVALID");
                 delay(1000);
                 goto st1;
        }
        st2:
        clrscr();
        cleardevice();
        int d1,m1,y1,d2,m2,y2,d3,m3,y3,d4,m4,y4;
        int d,m,y;
        rectangle (130, 100, 450, 380);
        gotoxy(25,7);
        cout<<" RAILWAY RESERVATION SYSTEM ";
        DATE dt;
        dt.CURRENT_DATE();
        d1=dt.day;
        m1=dt.month;
        y1=dt.year;
        gotoxy(18,9);
        cout<<"THE TRAIN WILL WE GOING ON THE";
        gotoxy(18,10);
        cout<<"FOLLOWING DATES :-";
        gotoxy(18,12);
        cout<<"1:"<<d1<<"/"<<m1<<"/"<<y1<<endl;
        dt.INCREASE_DATE(d1,m1,y1);
        d2=dt.day;
        m2=dt.month;
        y2=dt.year;
        gotoxy(18,13);
        cout<<"2:"<<d2<<"/"<<m2<<"/"<<y2<<endl;
        dt.INCREASE_DATE(d2,m2,y2);
        d3=dt.day;
        m3=dt.month;
        y3=dt.year;
        gotoxy(18,14);
        cout<<"3:"<<d3<<"/"<<m3<<"/"<<y3<<endl;
```

```
dt.INCREASE_DATE(d3,m3,y3);
d4=dt.day;
m4=dt.month;
y4=dt.year;
gotoxy(18,15);
cout<<"4:"<<d4<<"/"<<m4<<"/"<<y4<<endl;
gotoxy(18,17);
cout<<"5:"<<"PREVIOUS MENU";
gotoxy(18,18);
cout<<"6:"<<"MAIN MENU";
int sno=0;
gotoxy(18,20);
cout<<"Please enter your choice:-> ";
cin>>sno;
if(sno<1||sno>6)
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
}
switch(sno)
{
        case 1 : d=d1;
                    m=m1;
                    y=y1;
                    break;
        case 2 : d=d2;
                    m=m2;
                    y=y2;
                    break;
        case 3 : d=d3;
                    m=m3;
                    y=y3;
                    break;
```

```
                case 4 : d=d4;
                                m=m4;
                                y=y4;
                                break;
                case 5 : goto st1;
                        //break;
                case 6 : return;
                        //break;
                default:break;
        }
        int pnr=0,found=0,kms;
        gotoxy(18,21);
        cout<<"Enter PNR No. of the ticket:-> ";
        cin>>pnr;
        fstream file;
        file.open("TICKET.DAT",ios::in);
        while(file.read((char*)this,sizeof(TICKET)))
        {
                if((dd==d&&mm==m&&yy==y)&&!strcmp(tr_no,tno)&&pnr==pnr_no)
                {
                        found=1;
                        clrscr();
                        cleardevice();
                        rectangle (30, 85, 500, 85);
                        rectangle (30, 85, 30, 300);
                        rectangle (500, 85, 500, 300);
                        rectangle (30, 300, 500, 300);
                        rectangle (30, 105, 500, 105);
                        setcolor(YELLOW);
                        settextstyle(12, HORIZ_DIR, 1);
                        outtextxy(35,90,"INDIAN RAILWAY");
                        outtextxy(380,90,"HAPPY JOURNEY");
                        gotoxy(8,8);
                        cout<<"PRN NO."<<"\t\tDATE OF JOURNEY"<<"\t\tTRAIN NO.";
                        gotoxy(8,9);
```

```
            cout<<pnr_no<<"\t\t "<<dd<<"/"<<mm<<"/"<<yy<<"\t\t
"<<tr_no;
            gotoxy(8,11);
            cout<<"TRAIN NAME: \t"<<t_name;;
            gotoxy(8,13);
            cout<<"BOARDING PLACE "<<"\t\tDESTINATION";
            gotoxy(8,14);
            cout<<"  "<<from<<"\t\t\t"<<to;
            gotoxy(8,16);
            cout<<"PASSENGER NAME: "<<name;
            gotoxy(8,17);
            cout<<"PASSENGER AGE : "<<age;
            gotoxy(8,18);
            cout<<"SEX          : "<<sex<<"\t\t\t    FAIR: "<<fair;
            gotoxy(1,25);
            cout<<"Press any key to continue...";
            getch();
            rail.menu();
        }
    }
    file.close();
    if(!found)
    {
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "TICKET NO IS NOT FOUND");
        delay(1000);
        goto st2;
    }
}
//**********************************************************************
        //FUNCTION TO RESERVE TICKET FOR A PASSENGER
//**********************************************************************
void TICKET::RESERVATION()
{
    st1:
```

```
clrscr();
cleardevice();
TRAIN train;
railway rail;
train.TRAIN_LIST();
cout<<"\n Press r to RETURN.";
cout<<"\n Enter the train no:-> ";
cin>>tno;
if(tno[0]=='r' || tno[0]=='R')
        return;
if(!train.FOUND_TRAIN(tno))
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "TRAIN NUMBER IS INVALID");
        delay(1000);
        goto st1;
}
st2:
clrscr();
cleardevice();
int d1,m1,y1,d2,m2,y2,d3,m3,y3,d4,m4,y4,valid;
int d,m,y;
rectangle (130, 100, 450, 380);
gotoxy(25,7);
cout<<" RAILWAY RESERVATION SYSTEM ";
DATE dt;
gotoxy(18,9);
cout<<"YOU CAN RESERVE THE SEAT ON";
gotoxy(18,10);
cout<<"THE FOLLOWING DATES: ";
dt.CURRENT_DATE();
d1=dt.day;
m1=dt.month;
y1=dt.year;
gotoxy(18,12);
```

```
cout<<"1:  "<<" "<<d1<<"/"<<m1<<"/"<<y1<<endl;
dt.INCREASE_DATE(d1,m1,y1);
d2=dt.day;
m2=dt.month;
y2=dt.year;
gotoxy(18,13);
cout<<"2:  "<<" "<<d2<<"/"<<m2<<"/"<<y2<<endl;
dt.INCREASE_DATE(d2,m2,y2);
d3=dt.day;
m3=dt.month;
y3=dt.year;
gotoxy(18,14);
cout<<"3:  "<<" "<<d3<<"/"<<m3<<"/"<<y3<<endl;
dt.INCREASE_DATE(d3,m3,y3);
d4=dt.day;
m4=dt.month;
y4=dt.year;
gotoxy(18,15);
cout<<"4:  "<<" "<<d4<<"/"<<m4<<"/"<<y4<<endl;
gotoxy(18,17);
cout<<"5:  "<<" PREVIOUS MENU";
gotoxy(18,18);
cout<<"6:  "<<" MAIN MENU";
int sno=0;
gotoxy(18,20);
cout<<"Please select your choice:-> ";
cin>>sno;
if(sno<1||sno>6)
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
}
switch(sno)
```

```
        {
                case 1 : d = d1;
                        m = m1;
                        y = y1;
                        break;
                case 2 : d = d2;
                        m = m2;
                        y = y2;
                        break;
                case 3 : d = d3;
                        m = m3;
                        y = y3;
                        break;
                case 4 : d = d4;
                        m = m4;
                        y = y4;
                        break;
                case 5 : goto st1;
                case 6 : return;
                default : break;
        }
        int pnr=0,count=0,seat=0,seats=0;
        AA:
        //      TO GET THE TOTAL SEATS IN THE TRAIN
        fstream fout1;
         fout1.open("TRAIN.DAT",ios::in);
         while(fout1.read((char*)this,sizeof(TRAIN)))
         {
                if(strcmp(tr_no,tno))
                {
                        seat=train.train_seats;
                }
         }
         fout1.close();
//      TO COUNT THE NO. OF SEATS RESERVED
```

```
            fstream fout,file;
            fout.open("TICKET.DAT",ios::in);
             while(fout.read((char*)this,sizeof(TICKET)))
             {
                        if((dd==d&&mm==m&&yy==y)&&!strcmp(tr_no,tno))
                        {
                          count++;
                        }
             }
//          TO CHECK THAT SEAT IS AVAILABLE OR NOT.
//          IF THE SEAT IS NOT AVAILABLE THEN IT SHOWS MESSAGE
//                      ELSE IT CONTINUES.......
             if(count>=seat)
             {
             settextstyle(12, HORIZ_DIR, 1);
             outtextxy(180,460, "THERE IS NO SEAT IN THIS TRAIN");
             delay(2000);
             rail.menu();
             }
             clrscr();
             cleardevice();
             rectangle (130, 100, 450, 380);
             gotoxy(25,7);
             cout<<"                    ";
             settextstyle(7, HORIZ_DIR, 1);
             outtextxy(200,80, "RESERVATION FORM");
             gotoxy(18,10);
             cout<<"HOW MANY SEATS YOU WANT";
             gotoxy(18,11);
             cout<<"TO BE RESERVE: ->";
             cin>>seats;
             if(seats>=4)
             {
             settextstyle(12, HORIZ_DIR, 1);
             outtextxy(140,460, "U CAN ONLY RESERVE 3 SEATS......");
```

```
 delay(2000);
 rail.menu();
 }
 else
 if(seats>(seat-count))
 {
 settextstyle(12, HORIZ_DIR, 1);
 outtextxy(140,460, "THERE ARE NOT ENOUGH NO. Of SEATS IN TRAIN");
 delay(2000);
 rail.menu();
 }
 else
 {
 for(int i=1;i<=seats;i++)
 {
file.open("TICKET.DAT",ios::in|ios::binary);
while (file.read((char*)this,sizeof(TICKET))!=0)
{
        if((dd==d && mm == m && yy==y) && strcmp(tr_no,tno)==0)
        {
                pnr=pnr_no;
        }
}
file.close();
pnr++;
totaltickets++;
char pname[4][26],psex[4],ch='y',tr_name[30];
int pkms,page[4],pfair;
clrscr();
cleardevice();
rectangle (130, 100, 450, 380);
gotoxy(25,7);
cout<<"                    ";
settextstyle(7, HORIZ_DIR, 1);
outtextxy(200,80, "RESERVATION FORM");
```

```
gotoxy(18,10);
cout<<"PASSENGER NAME: ";
gotoxy(18,12);
cout<<"AGE: ";
gotoxy(18,14);
cout<<"SEX(M/F): ";
gotoxy(37,10);
cout<<"->";
gotoxy(40,10);
gets(pname[i]);
st3:
gotoxy(37,12);
cout<<"->";
gotoxy(40,12);
cout<<"  ";
gotoxy(40,12);
cin>>page[i];
if(page[i]<=5)
{

        gotoxy(18,25);
        cout<<"NO RESERVATION IS REQUIRED UPTO AGE 5";
        delay(1300);
        gotoxy(18,25);
        cout<<"                                        ";
        goto st3;

}
st4:
gotoxy(37,14);
cout<<"->";
gotoxy(40,14);
cout<<"  ";
gotoxy(40,14);
cin>>psex[i];
```

```
if(psex[i]=='m'||psex[i]=='M'||psex[i]=='f'||psex[i]=='F')
{
	settextstyle(12, HORIZ_DIR, 1);
	outtextxy(180,460, "PRESS A KEY TO CONTINUE.....");
	delay(1000);
	goto st5;
}
else
{
	gotoxy(18,25);
	cout<<"PLEASE ENTER THE CORRECT CHARACTER";
	delay(1300);
	gotoxy(18,25);
	cout<<"                                  ";
	goto st4;
}
st5:
if(tno[3]=='1')
 {
 strcpy(tr_name,"RAJDHANI EXPRESS");
 strcpy(pfrom,"MUMBAI");
 strcpy(pto,"GOA");
 pfair=550;
 }
 else
 if(tno[3]=='3')
 {
 strcpy(tr_name,"TAJ EXPRESS");
 strcpy(pfrom,"PUNJAB");
 strcpy(pto,"AGRA");
 pfair=750;
 }
 else
 if(tno[3]=='5')
 {
```

```
strcpy(tr_name,"GOMTI EXPRESS");
strcpy(pfrom,"BANGLORE");
strcpy(pto,"DELHI");
pfair=500;
}
else
if(tno[3]=='7')
{
strcpy(tr_name,"KALKA MAIL");
strcpy(pfrom,"INDORE");
strcpy(pto,"JAMMU");
pfair=450;
}
else
if(tno[3]=='9')
{
strcpy(tr_name,"KALINDI EXPRESS");
strcpy(pfrom,"KOLKATTA");
strcpy(pto,"DELHI");
pfair=800;
}
else
if(tno[3]=='0')
{
strcpy(tr_name,"JAIPUR MAIL");
strcpy(pfrom,"DELHI");
strcpy(pto,"JAIPUR");
pfair=250;
}
else
if(tno[3]=='2')
{
strcpy(tr_name,"BIHAR EXPRESS");
strcpy(pfrom,"DELHI");
strcpy(pto,"BIHAR");
```

```
pfair=300;
}
else
if(tno[3]=='4')
{
strcpy(tr_name,"A.P. EXPRESS");
strcpy(pfrom,"DELHI");
strcpy(pto,"HYDERABAD");
pfair=800;
}
else
if(tno[3]=='6')
{
strcpy(tr_name,"HIMACHAL EXPRESS");
strcpy(pfrom,"DELHI");
strcpy(pto,"HIMACHAL");
pfair=180;
}
else
if(tno[3]=='8')
{
strcpy(tr_name,"SHIMLA MAIL");
strcpy(pfrom,"DELHI");
strcpy(pto,"SHIMLA");
pfair=200;
}
if(!train.check(totaltickets))
{
        goto st2;
}
st6:
clrscr();
cleardevice();
rectangle (30, 85, 500, 85);
rectangle (30, 85, 30, 300);
rectangle (500, 85, 500, 300);
rectangle (30, 300, 500, 300);
rectangle (30, 105, 500, 105);
setcolor(YELLOW);
settextstyle(12, HORIZ_DIR, 1);
outtextxy(35,90,"INDIAN RAILWAY");
outtextxy(380,90,"HAPPY JOURNEY");
gotoxy(8,8);
cout<<"PRN NO."<<"\t\tDATE OF JOURNEY"<<"\t\tTRAIN NO.";
gotoxy(8,9);
cout<<pnr<<"\t\t "<<d<<"/"<<m<<"/"<<y<<"\t\t  "<<tno;
gotoxy(8,11);
cout<<"TRAIN NAME: \t"<<tr_name;;
gotoxy(8,13);
cout<<"BOARDING PLACE "<<"\t\tDESTINATION";
gotoxy(8,14);
cout<<"  "<<pfrom<<"\t\t\t"<<pto;
gotoxy(8,16);
cout<<"PASSENGER NAME: "<<pname[i];
gotoxy(8,17);
cout<<"PASSENGER AGE : "<<page[i];
gotoxy(8,18);
cout<<"SEX          : "<<psex[i]<<"\t\t\t  FAIR: "<<pfair;
cout<<"\n\n\n";
cout<<"Do you want to save ticket(y/n):-> ";
cin>>ch;
switch(ch)
{
case 'y':
case 'Y':
        file.open("TICKET.DAT",ios::app);
        t.pnr_no=pnr;
        t.kms=pkms;
        strcpy(t.t_name,tr_name);
        strcpy(t.name,pname[i]);
```

```
                strcpy(t.tr_no,tno);
                strcpy(t.from,pfrom);
                strcpy(t.to,pto);
                t.sex=psex[i];
                t.age=page[i];
                t.fair=pfair;
                t.dd=d;
                t.mm=m;
                t.yy=y;
                file.write((char*)&t,sizeof(TICKET));
                file.close();
                settextstyle(12, HORIZ_DIR, 1);
                outtextxy(180,460, "RECORD IS SAVED.........");
                delay(2000);
                break;
        case 'n':
        case 'N':
                rail.menu();
                break;
        default:
                settextstyle(12, HORIZ_DIR, 1);
                outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
                delay(1000);
                goto st6;
    }
    }
    }
}
//************************************************************************
        //FUNCTION TO CANCEL RESERVED TICKET OF A PASSENGER
//***********************************************************************
void TICKET::CANCELLATION()
{
        st1:
        clrscr();
```

```
cleardevice();
char tno[5];
TRAIN train;
railway rail;
train.TRAIN_LIST();
cout<<"\n Press r to RETURN";
cout<<"\n\n Enter the train no:-> ";
cin>>tno;
if(tno[0]=='r' || tno[0]=='R')
        return;
if(!train.FOUND_TRAIN(tno))
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "TRAIN NUMBER IS INVALID");
        delay(1000);
        goto st1;
}
st2:
clrscr();
cleardevice();
int d1,m1,y1,d2,m2,y2,d3,m3,y3,d4,m4,y4;
int d,m,y;
rectangle (130, 100, 450, 380);
gotoxy(25,7);
cout<<" RAILWAY RESERVATION SYSTEM ";
DATE dt;
dt.CURRENT_DATE();
d1=dt.day;
m1=dt.month;
y1=dt.year;
gotoxy(18,9);
cout<<"THE TRAIN WILL BE GOING ON THE";
gotoxy(18,10);
cout<<"FOLLOWING DATES:-";
gotoxy(18,12);
```

```
cout<<"1:"<<d1<<"/"<<m1<<"/"<<y1<<endl;
dt.INCREASE_DATE(d1,m1,y1);
d2=dt.day;
m2=dt.month;
y2=dt.year;
gotoxy(18,13);
cout<<"2:"<<d2<<"/"<<m2<<"/"<<y2<<endl;
dt.INCREASE_DATE(d2,m2,y2);
d3=dt.day;
m3=dt.month;
y3=dt.year;
gotoxy(18,14);
cout<<"3:"<<d3<<"/"<<m3<<"/"<<y3<<endl;
dt.INCREASE_DATE(d3,m3,y3);
d4=dt.day;
m4=dt.month;
y4=dt.year;
gotoxy(18,15);
cout<<"4:"<<d4<<"/"<<m4<<"/"<<y4<<endl;
gotoxy(18,17);
int sno=0;
cout<<"5:"<<"PREVIOUS MENU.";
gotoxy(18,18);
cout<<"6:"<<"MAIN MENU.";
gotoxy(18,20);
cout<<"Please enter your choice:-> ";
cin>>sno;
if(sno<1||sno>6)
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
}
switch(sno)
```

```
{
        case 1: d=d1;
                m=m1;
                y=y1;
                break;
        case 2: d=d2;
                m=m2;
                y=y2;
                break;
        case 3: d=d3;
                m=m3;
                y=y3;
                break;
        case 4: d=d4;
                m=m4;
                y=y4;
                break;
        case 5: goto st1;
        case 6: return;
        default : break;

}
int pnr=0,found=0;
char ch;
gotoxy(18,22);
cout<<"Enter PNR NO. of the ticket";
gotoxy(18,23);
cout<<"to be cancelled:-> ";
cin>>pnr;
fstream file,temp;
file.open("TICKET.DAT",ios::in);
temp.open("TEMP.DAT",ios::out);
while(file.read((char*)this,sizeof(TICKET)))
{
        if((dd==d && mm==m && yy==y) && !strcmp(tr_no,tno) && pnr==pnr_no)
        {

                found=1;
                cleardevice();
                rectangle (30, 85, 500, 85);
                 rectangle (30, 85, 30, 300);
                 rectangle (500, 85, 500, 300);
                 rectangle (30, 300, 500, 300);
                 rectangle (30, 105, 500, 105);
                 setcolor(YELLOW);
                 settextstyle(12, HORIZ_DIR, 1);
                 outtextxy(35,90,"INDIAN RAILWAY");
                 outtextxy(380,90,"HAPPY JOURNEY");
                 gotoxy(8,8);
                 cout<<"PRN NO."<<"\t\tDate of Journey"<<"\t\tTrain No.";
                 gotoxy(8,9);
                 cout<<pnr_no<<"\t\t "<<dd<<"/"<<mm<<"/"<<yy<<"\t\t
"<<tr_no;
                 gotoxy(8,11);
                 cout<<"TRAIN NAME: "<<t_name;
                 gotoxy(8,13);
                 cout<<"Boarding Place "<<"\t\tDestination";
                 gotoxy(8,14);
                 cout<<"  "<<from<<"\t\t\t"<<to;
                 gotoxy(8,16);
                 cout<<"NAME: "<<name;
                 gotoxy(8,17);
                 cout<<"AGE : "<<age;
                 gotoxy(8,18);
                 cout<<"SEX : "<<sex<<"\t\t\t   FAIR: "<<fair;;

                do
                {
                        gotoxy(8,25);
                  //    clreol();
                cout<<"Do you want to cancel ticket(y/n):-> ";
                cin>>ch;
            }while(toupper(ch)!='Y' && toupper(ch)!='N');
        }
        else
        temp.write((char*)this,sizeof(TICKET));

    }
    file.close();
    temp.close();
    if(!found)
    {
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
    }
    if(toupper(ch)=='N')
        goto st2;
    file.open("TICKET.DAT",ios::out);
    temp.open("TEMP.DAT",ios::in);
    while(temp.read((char*)this,sizeof(TICKET)))
    file.write((char*)this,sizeof(TICKET));
    file.close();
    temp.close();
    settextstyle(12, HORIZ_DIR, 1);
    outtextxy(180,460, "RECORD IS DELETED........");
    delay(1000);
    rail.menu();
}
//************************************************************************
                    //FUNCTION TO DISPLAY LIST OF PASSENGERS
//************************************************************************
void TICKET ::CHECK_LIST()
{
```

```
st1:
clrscr();
cleardevice();
char tno[5];
TRAIN train;
railway rail;
train.TRAIN_LIST();
cout<<"\n Press r to RETURN";
cout<<"\n\n Enter the train no:-> ";
cin>>tno;
if(tno[0]=='r' || tno[0]=='R')
        return;
if(!train.FOUND_TRAIN(tno))
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "TRAIN NUMBER IS INVALID");
        delay(1000);
        goto st1;

}
st2:
clrscr();
cleardevice();
int d1,m1,y1,d2,m2,y2,d3,m3,y3,d4,m4,y4;
int d,m,y;
rectangle (130, 100, 450, 380);
gotoxy(25,7);
cout<<" RAILWAY RESERVATION SYSTEM ";
DATE dt;
dt.CURRENT_DATE();
d1=dt.day;
m1=dt.month;
y1=dt.year;
gotoxy(18,9);
cout<<"THE TRAIN WILL BE GOING ON THE";
```

```
gotoxy(18,10);
cout<<"FOLLOWING DATES :-";
gotoxy(18,12);
cout<<"1:"<<d1<<"/"<<m1<<"/"<<y1<<endl;
dt.INCREASE_DATE(d1,m1,y1);
d2=dt.day;
m2=dt.month;
y2=dt.year;
gotoxy(18,13);
cout<<"2:"<<d2<<"/"<<m2<<"/"<<y2<<endl;
dt.INCREASE_DATE(d2,m2,y2);
d3=dt.day;
m3=dt.month;
y3=dt.year;
gotoxy(18,14);
cout<<"3:"<<d3<<"/"<<m3<<"/"<<y3<<endl;
dt.INCREASE_DATE(d3,m3,y3);
d4=dt.day;
m4=dt.month;
y4=dt.year;
gotoxy(18,15);
cout<<"4:"<<d4<<"/"<<m4<<"/"<<y4<<endl;
gotoxy(18,17);
cout<<"5:"<<"PREVIOUS MENU";
gotoxy(18,18);
cout<<"6:"<<"MAIN MENU";
int sno=0;
gotoxy(18,20);
cout<<"Please enter your choice:-> ";
cin>>sno;
if(sno<1||sno>6)
{
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE PROVIDE CORRECT INPUT");
        delay(1000);
        goto st2;
}
switch(sno)
{
        case 1: d=d1;
                m=m1;
                y=y1;
                break;
        case 2: d=d2;
                m=m2;
                y=y2;
                break;
        case 3: d=d3;
                m=m3;
                y=y3;
                break;
        case 4: d=d4;
                m=m4;
                y=y4;
                break;
        case 5: goto st1;
        case 6: return;
        default: break;
}
clrscr();
cleardevice();
fstream fout;
fout.open("TRAIN.DAT",ios::in);
while(fout.read((char*)this,sizeof(TRAIN)))
{
        if(strcmp(tr_no,tno))
        {

                strcpy(t_name,train.train_name);
                strcpy(from,train.train_source);
                strcpy(to,train.train_desti);
        }
}
fout.close();
 setcolor(YELLOW);
 rectangle(60,40,530,41);
 rectangle(60,85,530,86);
 rectangle(60,115,530,116);
 rectangle(60,40,61,440);
 rectangle(60,439,530,440);
 rectangle(150,85,151,440);
 rectangle(330,85,331,440);
 rectangle(420,85,421,440);
 rectangle(529,40,530,440);
fstream file;
file.open("TICKET.DAT",ios::in);
gotoxy(25,1);
cout<<"   LIST OF PASSENGERS";
gotoxy(10,2);
cout<<"Date:"<<d <<"/"<<m<<"/" <<y;
gotoxy(10,4);
cout<<"TRAIN NO: "<<tno<<"\t   TRAIN NAME: "<<t_name;
gotoxy(10,5);
cout<<"FROM    : "<<from<<"\t      TO        : "<<to<<endl;
gotoxy(10,7);
cout<<"PNR NO.";
gotoxy(23,7);
cout<<"PASSENGER NAME";
gotoxy(45,7);
cout<<"AGE";
gotoxy(58,7);
cout<<"SEX";
int row=9,found=0;
while(file.read((char*)this,sizeof(TICKET)))
{
        if((dd==d && mm==m && yy==y) && !strcmp(tr_no,tno))
        {
                found=1;
                gotoxy(14,row);
                cout<<pnr_no;
                gotoxy(25,row);
                cout<<name;
                gotoxy(46,row);
                cout<<age;
                gotoxy(60,row);
                cout<<sex;
                row++;
                if(row==25)
                {
                    getch();
                    for(int i=9;i<24;i++)
                    {
                        gotoxy(1,i);
                        clreol();
                    }
                    row=9;
                }
        }
}
file.close();
if(!found)
{
        settextstyle(7, VERT_DIR, 2);
        outtextxy(250,200, "NO RESERVATION");
        settextstyle(12, HORIZ_DIR, 1);
        outtextxy(180,460, "PLEASE ENTER A KEY TO CONTINUE......");
        getch();
        goto st2;
}
settextstyle(12, HORIZ_DIR, 1);
```

```
        outtextxy(180,460, "PLEASE ENTER A KEY TO CONTINUE......");
        getch();
        rail.menu();
}
//*************************************************************
         // THIS FUNCTION HANDLES THE SEAT NUMBER OF THE TRAIN
//*************************************************************
int TRAIN :: check(int r)
{
        fstream file;
        file.open("TRAIN.DAT",ios::in);
        delay(100);
        file.open("TRAIN.DAT",ios::in);
        while(file.read((char*)this,sizeof(TRAIN)))
        {
                if(strcmp(train_no,tno)==0)
                {
                        int a = train_seats;
                        a -= r;
                        file.flush();
                }
        }
        file.read((char*)this, sizeof(TRAIN));
        file.close();
        getch();
        return 1;
}
//*******************************************************************
 //MAIN FUNCTION WHICH DISPLAYS MAIN MENU & CALLS ALL OTHER FUNCTIONS
//*******************************************************************
void main()
{
        char ch,ch1,ch3;
        TICKET ti;
        TRAIN  train;
```

```
railway rail;
int gd=DETECT,gm;
initgraph(&gd, &gm,"c:\\tc\\bgi");
train.ADDITION();
train.DISPLAY_DESIGN();
cc:
clrscr();
cleardevice();
rail.sketch();
gotoxy(30,10);
rail.menu();
gotoxy(23,22);
cout<<"Enter your choice:-> ";
ch=getche();
if(ch=='1')
{
        clrscr();
        cleardevice();
        ti.RESERVATION();
        goto cc;
}
else
if(ch=='2')
{
        pp:
        clrscr();
        cleardevice();
        rail.sketch();
        rail.query();
        gotoxy(23,22);
        cout<<"Enter your choice:-> ";
        ch3=getche();
        if(ch3=='1')
        {
                clrscr();
```

```
		cleardevice();
		train.TRAIN_LIST();
		settextstyle(12, HORIZ_DIR, 1);
		outtextxy(180,460, "PRESS A KEY TO CONTINUE........");
		getch();
		goto cc;
	}
	else
	if(ch3=='2')
	{
		clrscr();
		cleardevice();
		ti.PASS_INFO();
		getch();
		goto cc;
	}
	else
	if(ch3=='3')
	{
	clrscr();
	cleardevice();
	ti.AVAILABILITY();
	goto cc;
	}
	else
	if(ch3=='4')
	{
	goto cc;
	}
	else
	{
	gotoxy(25,25);
	cout<<"YOU HAVE ENTERED WRONG CHOICE...";
	delay(1000);
	goto pp;
```

```
        }

    }
    else
    if(ch=='3')
    {
        clrscr();
        cleardevice();
        ti.CANCELLATION();
        delay(500);
        goto cc;
    }
    else
    if(ch=='4')
    {
        clrscr();
        cleardevice();
        ti.CHECK_LIST();
        goto cc;
    }
    else
    if(ch=='5')
    {
        settextstyle(12, HORIZ_DIR, 1);
        setcolor(YELLOW);
        outtextxy(180,460,"THANKS FOR VISITING");
        delay(1000);
        exit(0);
    }
    else
    {
        clrscr();
        cleardevice();
        gotoxy(3,3);
        cout<<"You have entered Wrong choice....";
        cout<<"\n   Do u want to continue(Y/N):-> ";
        cin>>ch1;
        if(ch1=='y'||ch1=='Y')
        {
            goto cc;
        }
        else
        {
            settextstyle(7, HORIZ_DIR, 4);
            setcolor(YELLOW);
            outtextxy(140,180,"THANKS FOR VISITING");
            delay(1000);
        }
    }
    closegraph();

}
/****************************************************************************
*****************************************************************************
            END OF PROJECT
****************************************************************************
****************************************************************************/
```

SYSTEM TESTING

SYSTEM TESTING

System testing is a process of executing a program to locate an error. Testing is the verification and validation activity for the software product. It involves checking processes, such as inspection and reviews, at each stage of the software process from user requirements definition to program development. During system testing, the system is used experimentally to ensure that the software does not fail, i.e. it will run according to its specification and in the way users expect to it.

Testing is a four stage process where system components are tested, and the integrated system is tested and finally the system is tested with the customer's data. The four types of testing are as follows:

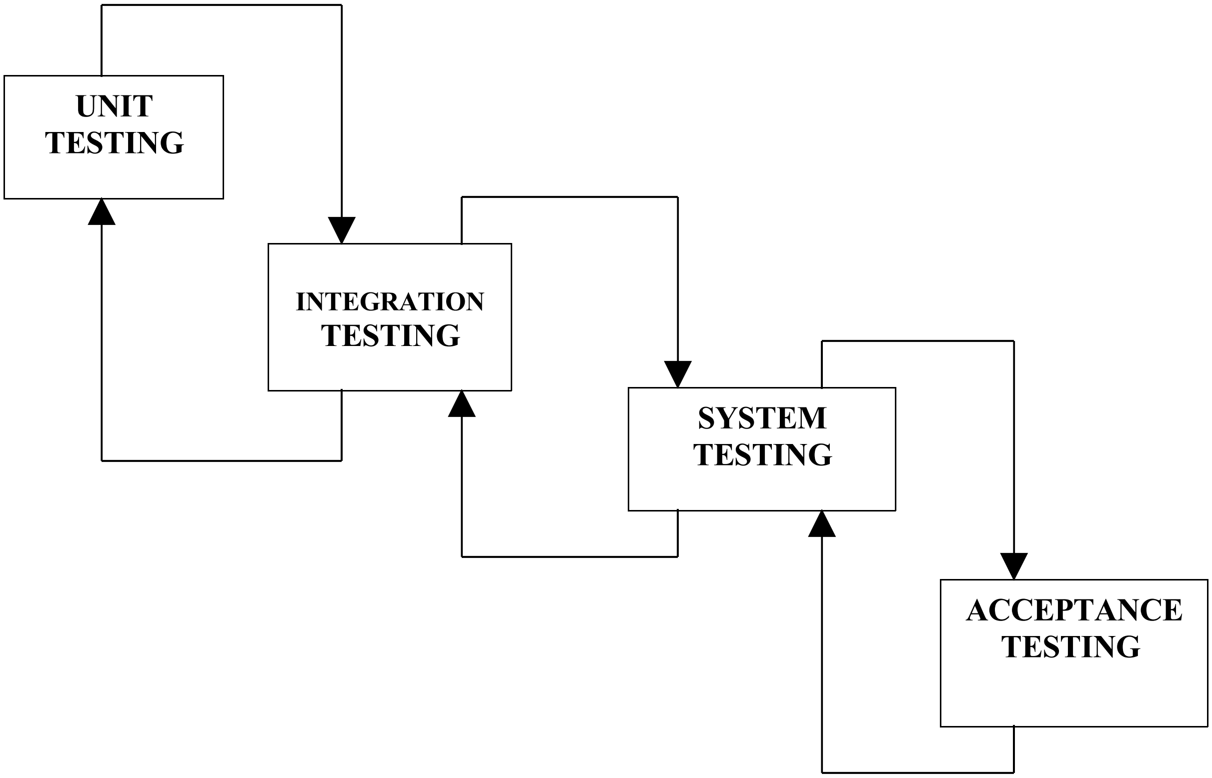


Figure 15: - Testing Process

The following are the stages in testing process:

- **Unit Testing: -**

Unit testing is a method of testing that verifies the individual units of source code are working properly. A unit is the smallest testable part of an application. Using component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. Tests of data flow across a module interface were carried out for each module of the software. I.e. unit testing ensures that functions or procedures are operating correctly. Each component is tested individually.

In my project "Railway Reservation System", I have done unit testing on various modules. For example, in RESERVATION () function of class TICKET, when user enter the train number, then firstly it checks the train record and search the train number. Then, if the train number is found, then the process is continuing, else it will shows a message, **"TRAIN NUMBER IS INVALID".** This message is shown whenever the user enters the wrong train number or entering any character value.

```
cout<<"\n Enter the train no:-> ";
        cin>>tno;
        if(tno[0]=='r' || tno[0]=='R')
                return;
        if(!train.FOUND_TRAIN(tno))
        {
                settextstyle(12, HORIZ_DIR, 1);
                outtextxy(180,460, "TRAIN NUMBER IS INVALID");
                delay(1000);
                goto st1;
        }
```

- **Black Box Testing: - Black box testing** takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure.

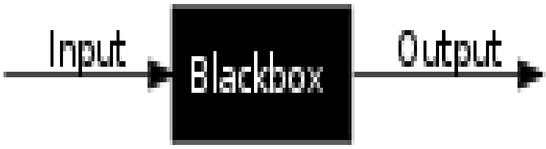


Figure 16: - Black box testing diagram

- **<u>White Box Testing</u>: -** White box uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. In electrical hardware testing, every node in a circuit may be probed and measured. While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units during integration, and between subsystems during a system level test. Though this method of test design can uncover an overwhelming number of test cases, it might not detect unimplemented parts of the specification or missing requirements, but one can be sure that all paths through the test object are executed.

- **<u>Integration Testing</u>: -**

Integration testing is a process of combining multiple modules systematically for conducting tests in order to find errors in the interface between modules. The goal here is to see if the modules can be integrated properly. Hence, the emphasis is on testing the interfaces between modules. The integration testing follows the following strategies: -

- Incremental strategy.
- Non-incremental strategy.
- Mixed strategy.

In my project "Railway Reservation System", I have done integration testing on various modules. For example, when I combined two modules/functions CANCELLATION()

and AVAILABILITY() of class TICKET, then they were working properly. When the user cancels the ticket from the database, then the AVAILABILITY() function shows that the ticket is cancelled.

Note: ***Refer page no.*** ***52-58 for AVAILABILITY() function and 74-79 for CANCELLATION() function.***

- **System Testing: -** In the system testing, the entire software system is tested. The reference document for this purpose is the requirements document, and the objective is to see if the software meets its requirements. A number of different types of testing are used at this stage. They are follows:
    - **Functional Testing: -** It is concerned with functionality rather than the implementation of the program. In functional testing the structure of the program is not considered.
    - **Performance Testing: -** It deals with quality related issues like security test, reliability test etc. System testing is done by the developers and before the product is given to customer for use.

In my project," Railway Reservation System", I have done the system testing, and I found that there was no error in the project and it is running accurately & correctly.

- **Acceptance Testing: -** After the system testing is completed successfully by the developers. Acceptance testing is done at the customer's end. It is the customer and the end user who now designs the test cases. Acceptance testing is performed with realistic data of the client to demonstrate that is software is working satisfactory.

IMPLEMENTATION

IMPLEMENTATION

System implementation is the final stage of development. It consists of installing hardware, programs, collecting data and organizing people to interact with and run the system.

In the implementation phase, user actually starts using the system. This phase therefore involves training the users for using the system and also providing them friendly documentation to refer to.

Implementation can be done in two ways. One way is by implementing the new system along with the old system and makes them run in parallel. The other method is to replace the entire system. In large organizations, the new system can be implemented in certain areas as a pilot project and if satisfactory results are obtained, it can be implemented to other areas also.

CONCLUSION

CONCLUSION

This software is a database project with all the basic capabilities a database should have. This application software is about railway reservation and it records and maintains records about the rails, booking and the passengers. Records for railways and passenger are saved in separate files and can be used to delete or modify records in them. The objective of the project is to perform all the functions or operations accurately. It overcomes all the problems that we have in our existing system.

SCOPE OF THE PROJECT

SCOPE OF THE PROJECT

Today, IT industry is known as word of magic. No doubt it brings a universal change in society. The existing manual system is very tedious, therefore this system is used to get the existing system more efficient and fast.

The proposed system aims to develop a system that helps to manage all drawbacks of the existing system. The system has a user friendly environment where the end user can perform all the activity. Even if the user has little knowledge of computers he/she can easily operate.

The proposed system has the following objectives to be achieved.

- ✓ User Friendly Environment.
- ✓ Less Space.
- ✓ Fast Retrieval.
- ✓ Easy to Operate.
- ✓ Accuracy.
- ✓ Cost Reduction.

REFERENCES

REFERENCES

WEB REFERENCES

- http://en.wikipedia.org
- http://google.com

BOOKS REFERENCES

- **Object Oriented Programming with C++** by E Balagurusamy Tata Mc Graw Hill India.
- **System Analysis And Design** by Elias M.Awad
- BOOK: Object Oriented Programming in Turbo C++
  AUTHOR: Robert Lafore
  PUBLISHER: Galgotia Publications pvt. Ltd.
- BOOK: Computer Sciences with C++
  AUTHOR: Sumita Arora
  PUBLISHER: DHANPAT RAI & CO.
- BOOK: Structured System Analysis and Design (A Level)
  AUTHOR: Dr. Madhulika Jain
  PUBLISHER: BPB PUBLICATIONS
- BOOK: Software Engineering
  AUTHOR: Bali-Bali
  PUBLISHER: S. K. KATARIA & SONS