I. SYSTEM SPECIFICATIONS AT A GLANCE

INTRODUCTION:

This program is designed to display flight details. The program also assists users to book and cancels flight tickets. The program for this project make extensive use of several features of C++ including classes, nested classes, files, user defined functions and algorithmic programming.

The quintessential purpose of this project is to enable easier and more efficient access and tracking of the flight details by the system. Just as in an airport, all changes, update & transactions are automated; this program tries to mimic all "behind the scene" flight transactions in the real time airport.

Classes:

This program uses object-oriented programming to work with two main classes to achieve the required objectives of creating a program to enable airline reservation to its users.

1. Passenger
2. Airline

Nested classes:

An important feature of this program is that it uses nested classes effectively to encapsulate different kinds of data.

Modules:

Modules deal with:

1. Adding a new flight
2. Modifying flight details
3. Listing of flight
4. Listing all flights to a destination
5. Deleting of flight

6. Booking on tickets
7. View a ticket
8. Delete a ticket
9. List of all passenger of flight

The project will be divided into various modules to enable easier understanding of the source code.

FILES

Since all objects of person class or with airplane class, file management is simplified using a single file.

Data is also retrieved easily

Function of each class as well as Stream class work together to retrieve data. There is an adequate scope for the modification of flight schedules using these functions

NAME OF FILE USED; flight.dat

Objects stored in file:

1. Airlines
2. Passenger (INSIDE RESPECTIVE AIRLINE OBJECTS)

Table 1: Structure of flight.dat file

Fno	n	dt	Cost	Des	D	M	Y	Sb	Passengers [10]

REQUIREMENT ANALYSIS:

INPUT /OUTPUT REUIREMENTS

SL No	Function name	Input Data	Output Data
1.	getdata()	Inputs passenger details like name, age, sex, nationality	Write the passenger details to ‘flight.txt’
2	showdata()	-	Displays details of the passengers
3	addf()	Flight details like flight number, name, destination, cost, date and departure time	Appends the flight details to ‘flight.txt’ file
4	displayf()	-	Displays details of the flight
5.	bookticket()	If seat is available this assigns seat number and ticket number	Updates the total number of seats booked for a particular flight
6.	displayt(int)	Passes the index of seat number	Displays a particular ticket
7.	displayf()	-	Displays the details of flight
8.	displaypass()	-	Displays the details of a passenger
9.	cancelticket(int)	Passes the index of seat number	Resets the passenger details with initial values
10.	header()	-	Displays the project title with graphics

11.	returnfno()	-	Accessor funtion that returns fno.
12.	returndes()	-	Returns the destination
13.	getticket()	-	Accessor function that returns ticket
14.	intval(int)	Integer to be validated	If integer is valid, it return the number else returns 0
15.	strval(char[])	Character array to be validated	If string is valid it returns 1, else returns 0

HARDWARE REQUIREMENTS

Processors: Intel Pentium 4 CPU

CPU Speed: 2.67 GHz

RAM: 512 MB

Hard Disk Memory: 80 GB

Cache: 512 KB

Software requirements

Operating system: Microsoft Windows XP Professional or Higher

Software: Turbo C++ IDE

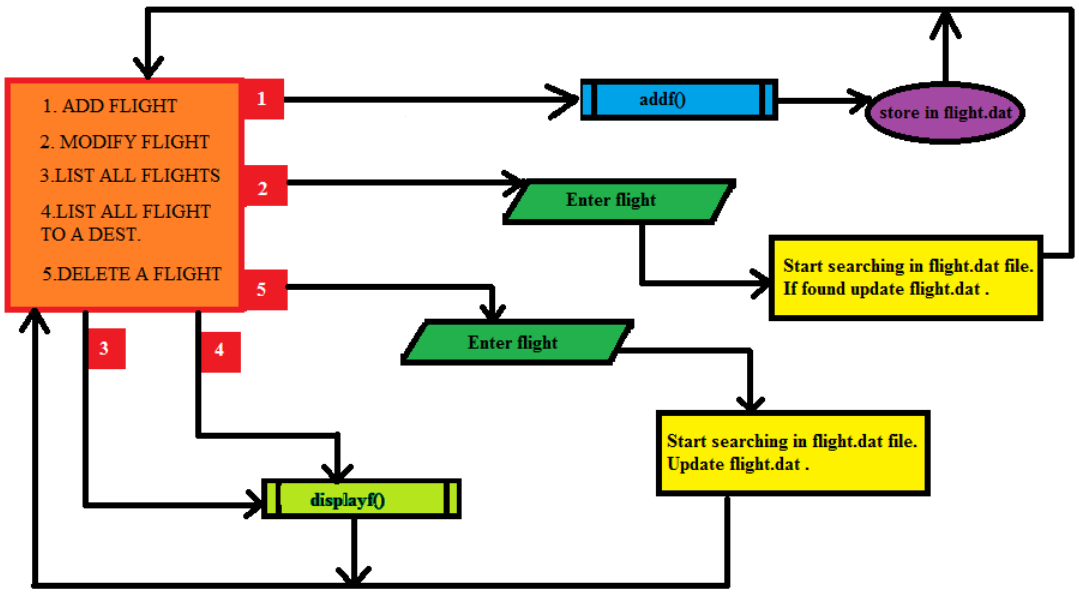
II. SYSTEMS SPECIFICATIONS

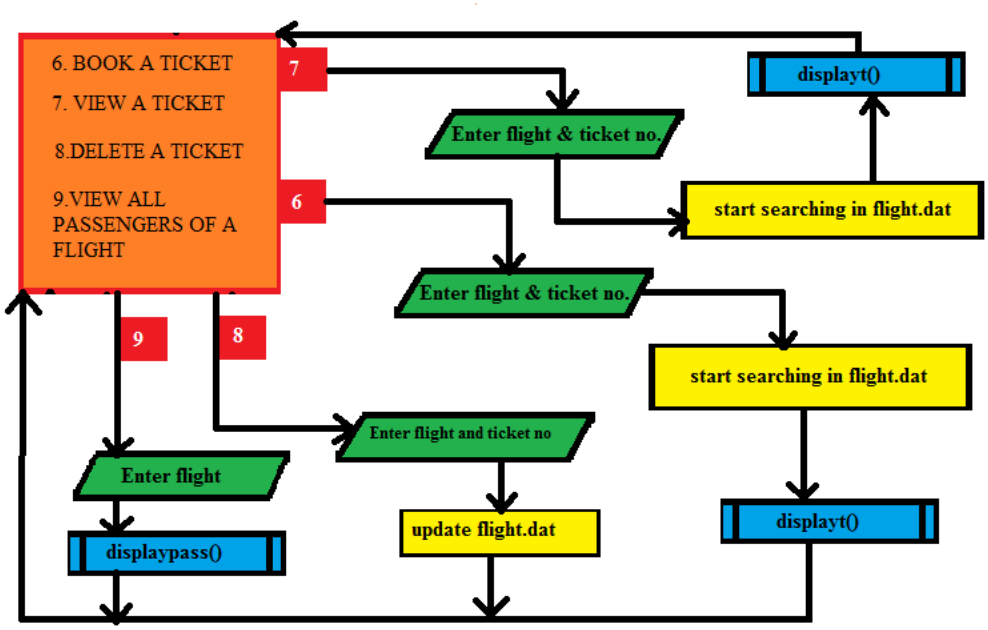
SYSTEM DESIGN

The system is designed to enter all flight details like carrier name, flight number, arrival time & departure time. This system built to capture the personal details of the passenger and book flight tickets and it is also assumed that only administrator of the system will be using the system for all the functionalities like entering the flight details, booking tickets, etc.

It is designed to work on all systems with windows OS 2000 / XP and later. It is designed to be a single user system. The validation of the system is done through the testing of the systems

The object oriented programming concepts have been used for the development of the system the coding is done through C++ and the file based approach being used.





DATA DESIGN

Ifstream anf ofstream class have been used to capture and store all the details of the system like passenger details, flight details etc.

CLASS: passenger

Data members:

S.NO	NAME	TYPE	SIZE	DESCRIPTION
1.	age	int	2	Age of the passenger
2.	nationality[30]	char	30	Nationality of passenger
3.	nm[30]	char	30	Name of passenger
4.	ticket	int	2	Ticket number of passenger
5.	sex[5]	char	5	Sex of passenger
6.	seatn	int	2	Seat number of passenger

Member functions:

1) getdata() : Used for getting the details of the passenger as input

2) showdata(): Used for displaying the details of passenger upon request.

3) reset(): Used to reset data members with initial values

.

CLASS: Airline

Data members:

S.No	NAME	TYPE	SIZE	DESCRIPTION
1.	n[30]	char	30	Name of flight
2.	cost	float	4	Travel cost
3	dt	int	2	Departure time (in hours)
4.	sb	int	2	No of seats booked
5.	fno	int	2	Flight number
6.	des[30]	char	30	Destination place of flight
7.	S[10]	passenger	-	Seats of flight
8.	d	int	2	Date of the flight
9.	m	int	2	
10.	y	int	2	

Member funtions:

addf() : Used to enter details of flight as input

modify() : Used to modify flight details.

displayf() : Used to display flight details

displayt() : Used to display the ticket

displaypass() : Used to display passenger details

cancelticket() : Used to cancel a ticket.

III. C++ OVERVIEW

C++ is a statically typed, free-form, multi-paradigm, compiled, general purpose programming language. It is regarded as an intermediate-level language, as it comprises a combination of both high-level and low-level features. Developed by Bjarne Stroustrup starting in 1979 at Bell Labs, it adds object oriented features such as classes, and other enhancements to the C programming language. Originally named C with Classes, the language was renamed C++ in 1983

C++ is one of the most popular programming languages and is implemented on a wide variety of hardware and operating system platforms. As an efficient compiler to native code, its application domains include systems software, application software, device drivers, embedded software, high-performance server and client applications, and entertainment software such as video games. Several groups provide both free and proprietary C++ compiler software, including the GNU Project, Microsoft, Intel and Embarcadero Technologies. C ++has greatly influenced many other popular programming languages, most notably C# and Java Other successful languages such as Objective-C use a very different syntax and approach to adding classes to C.

C++ is also used for hardware design, where the design is initially described in C++, then analyzed, architecturally constrained, and scheduled to create a register transfer level hardware description language via high-level synthesis.

The language began as enhancements to C, first adding classes, then virtual functions, operator overloading, multiple inheritance, templates, and exception handling among other features.

IV. Standard template library (STL)

The STL provides a ready-made set of common classes for C++, such as containers and associative arrays, that can be used with any built-in type and with any user-defined type that supports some elementary operations (such as copying and assignment). STL algorithms are independent of containers, which significantly reduces the complexity of the library.

Standard library

The C+ standard library incorporates the C standard library with some small modifications to make it optimized with the C++ language. Another large part of the library is based on the STL. This provides such useful tool
as containers (for example vectors and lists), iterators to provide these containers with array-like access and algorithms to perform operations such as searching and sorting. Further more (multi) maps (associative arrays) and (multi)sets are provided, all of which export compatible interfaces. Therefore it is possible, using template, to write generic algorithms that work with any container or on any sequence defined by iterators. As in C, the features of the library are accessed by using the #include directive to include a standard header. C++ provides 105 standard headers, of which 27 are deprecated.

Language features

C++ inherits most of C's syntax. The following is Bjarne Stroustrup's version of the Hello world program that uses the C++ Standard Library stream facility to write a message to standard output:

```
# include <iostream>
int main()
{
std::cout<<"Hello, world!"\ n;
}
```

Within functions that define a non-void return type, failure to return a value before control reaches the end of the function results in undefined behavior (compilers typically provide the means to issue a diagnostic in such a case). The sole exception to this rule is the main function, which implicitly returns a value of zero.

Operators & operator overloading

C++ provides more than 35 operators, covering basic arithmetic, bit manipulation, indirection, comparisons, logical operations and others. Almost all operators can be overloaded for user-defined types, with a few notable exceptions such as member access (. and.*) as well as the conditional operator. The rich set of overloadable operators is central to using C++ as a domain-specific language. The overloadable operators are also an essential part of many advanced C++ programming techniques, such as smart pointers. Overloading an operator does not change the precedence of calculations involving the operator, nor does it change

the number of operands that the operator uses (any operand may however ignored by the operator, though it will be evaluated prior to execution). Over "& &" and "||" operators lose their short-circuit evaluation property.

Objects

C++ introduces object-oriented programming (OOP) features to
It offers classes, which provide the four features commonly present in OOP (and some non-OOP) languages: abstraction, encapsulation, inheritance
and polymorphism. One distinguishing feature of C++ classes compared to classes in other programming languages is support for deterministic destructors, which in turn provide support for the resource concept.

Encapsulation

Encapsulation is the hiding of information in order to ensure that data structures the developer. C++ provides the ability to define classes and functions as its either public, protected, or private in order to explicitly enforce encapsulation. A and operators are used as intended and to make the usage model more obvious to primary encapsulation mechanisms. Within a class, members can be declared as public member of the class is accessible to any function. A private member is accessible only to functions that are members of that class and to functions and classes explicitly granted access permission by the class ("friends"). A protected member is accessible to members of classes that inherit from the class in addition to the class itself and any friends.

The OO principle is that all of the functions (and only the functions) that access the internal representation of a type should be encapsulated within the type definition C++ supports this (via member functions and friend functions), but does not enforce it: the programmer can declare parts or all of the representation of a type to
be public, and is allowed to make public entities that are not
part of the representation of the type. Therefore, C++ supports not just OO programming,
but other weaker decomposition paradigms, like modular programming
It is generally considered good practice to make all data private or protected, and to make public only those functions that are part of a minimal interface for users of is can hide the details of data implementation, allowing the designer to
later fundamentally change the implementation without changing the interface in any way.

Inheritance

Inheritance allows one data type to acquire properties of other data types
Inheritance from a base class may be declared as public
access specifier determines whether unrelated and derived classes
inherited public and protected members of the base class. Only public inheritance corresponds to what is usually meant by "inheritance". The other two forms are much less frequently used. If the access specifier is omitted, a "class" inherits privately, while a "struct" inherits publicly. Base classes may be declared as virtual: this is called virtual inheritance. Virtual inheritance ensures that only one instance of a base class exists in the inheritance graph, avoiding some of the ambiguity problems of multiple inheritance.

Multiple inheritance is a C++ feature not found in most other languages, allowing a class to be derived from more than one base classes; this allows for more elaborate inheritance relationships. For example, a "Flying Cat" class can inherit from both "Cat" and "Flying Mammal". Some other languages, such as C# or Java, accomplish something similar (although more limited) by allowing inheritance of multiple interfaces while restricting the number of base classes to one (interfaces. unlike classes, provide only declarations of member functions, no implementation or member data). An interface as in C# and Java can be defined in C++ as a class containing only pure virtual functions, often known as an abstract base class or "ABC". The member functions of such an abstract base class are normally explicitly defined in the derived class, not inherited implicitly. C++ virtual inheritance exhibits an ambiguity resolution feature called dominance.

Polymorphism

Polymorphism enables one common interference for many implementations, and for objects to act differently under different circumstances.

C++ supports several kinds of static (compile-time)
and dynamic (run-time) polymorphisms .
Compile-time polymorphism does not allow for certain run-time decisions, while run-time polymorphism typically incurs a performance penalty.

Static polymorphism

Function overloading allows programs to declare multiple function
same name (but with different arguments).
The functions are distinguished by the number or types of their formal parameters. Thus, the same function name can refer to different functions depending on the context in which it is used. The type returned by the function is not used to distinguish overloaded functions and would result in a compile-time error message.

When declaring a function, a programmer can specify for one or more p
a default value. Doing so allows the parameter s with defaults to optionally be omitted when the function is called, in which case the default arguments will be used. When a function is called with fewer arguments than there are declared parameters, explicit arguments are matched to parameters in left-to-right order, with any unmatched parameters at the end of the parameter list being assigned their default arguments. In many cases, specifying default arguments in a single function declaration is preferable to providing overloaded function definitions with different numbers of parameters.

Templates in C++ provide a sophisticated mechanism for writing generic, polymorphic code. In particular, through the Curiously Recurring Template Pattern, it's possible to implement a form of static polymorphism that closely mimics the syntax for overriding virtual functions. Since C++ templates are type-aware and Turing-complete, they can also be used to let the compiler resolve recursive conditionals and generate substantial programs through template meta programming. Contrary to some opinion, template code will not generate a bulk code after compilation with the proper compiler settings.

Compatibility of C and C++

C++ is often considered to be a superset of C. but this is not strictly true.[36] Most C code can easily be made to compile correctly in C++, but there are a few differences that cause some valid C code to be invalid or behave differently in C++

One commonly encountered difference is that C allows implicit conversion from void* to other pointer types, but C++ does not. Another common portability issue is that C++ defines many new keywords, such as new and class, that may be used as identifiers (e.g. variable names) in a C program.

Some incompatibilities have been removed by the 1999 revision of the C standard (C99), which now supports C++ features such as line comment (//) and mixed declarations and code. On the other hand, C99 introduced a number of new features that C++ did not support, such as variable-length arrays, native complex-number types, designated initializers, and compound literals.

Criticism

Due to its large feature set and oft-perceived "strict" syntax, the language is sometimes criticized as being overly complicated and thus difficult to fully master. C++ is sometimes compared unfavorably with more strictly object-oriented languages on the basis that it enables programmers to "mix and match" declarative, functional, generic, modular, and procedural

programming styles with object-oriented programming, rather than strictly enforcing a single style, although C++ is intentionally a multi-paradigm language.

A widely distributed satirical article portrayed Bjarne Stroustrup, interviewed for a 1998 issue of IEEE's Computer magazine, confessing that C+ was deliberately designed to be complex and difficult, weeding out amateur programmers and raising the salaries of the few programmers who could master the language. The FAQ section of Stroustrup's personal website contains a denial and a link to the actual interview.

V. Header files

Graphics.h	setbkcolour()	To change the background colour
	intigraph()	To initialize the graphics driver
	Setcolour()	To set the color of text
	Settextstyle()	To change the style of text
	Outtextxy()	To print on the graphics screen
conio.h	getch()	For getting a character as input
	cprintf()	For printing on screen
	textmode()	To bring the screen to textmode()

	getch()	To get a character as input
	textcolor()	To change the color of the text
	clrscr()	To clear the output screen
	Textbackground()	To set the background color of the text
stdio.h	gets()	For getting a string
	remove()	For deleting a file
	rename()	For renaming a file
string.h	strcpy()	To copy a string to another
	strcmpi()	To compare 2 strings without sensitivity
fstream.h	open()	To open a file
	read()	To read data from a file
	seekg()	To put the file pointer while reading
	seekp()	To put the file pointer while writing
	tellg()	To get the position of file pointer while writing
	write()	To write data into a file
	close()	To close a fila
stdlib.h	atoi()	To convert string to integer
	exit()	To stop program execution
	random()	To get a random number
	randomize()	To generate different random number based on time
ctype.h	isdigit()	To check if a character is a digit
	isalpha()	To check for an alphabet
iomanip.h	setw()	To set the width of the data

VI. File Design Structure

This project is designed for the case of the administrators to handle flight booking and cancelling for the passengers.

Name of File Used: flight.dat
Thus effectively only a single object is stored in the file and the data are,

1. Flight number
2. Flight Name
3. Departure time
4. Cost
5. Destination
6. Day
7. Month
8. Year
9. SB (Seats Booked)
10. S (10) which is the object of passenger class.

VII. Procedure/Function Description

Here are the the functions that are used in the program.

VALIDATING THE DATA

strval()
intval()

ENTERING DETAILS AS INPUT

getdata()
addf()

DISPLAYING DETAILS

showdata()
displayt()
displaypass()

CONSTRUCTOR FUNCTIONS

```
passenger()
{
ticket=0;
strcpy(nm,”null”);
strpy(nationality,”null”);
strpy(sex,”null”);
age=0;
seatn=0
}
Airline()
{
Flightno=0;
Strcmpi(n,NULL);
Strcmpi(des,NULL);
Cost=0;
D=m=y=0;
Dt=0;
}
```

OTHER FUNCTIONS

void bookticket (); For booking a flight ticket
header(): To display project title as header of every page

ACCESSOR FUNCTIONS

```
Char*retname ()
{
Return nm;
}
Int getticket()
{
Return ticket;
}
Int returnfno()
{
Return flightno;
}
Char*returndes()
{
Return des;
}
```

VIII. Program Source Code

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<ctype.h>
#include<graphics.h>
#include<string.h>
#include<stdlib.h>
#include<dos.h>
#include<iomanip.h>
 ifstream fin;                  int i=0;
 ofstream fout;
int strval(char s[]) //to validate strings
{
int k=0;int c=0;
for(int i =0; i<strlen(s); i++)
{

if(!(isalpha(s[i])|| s[i] ==' '))
k++;
```

```
if(s[i]==' ')
c++;


}
if(k !=0||c==strlen(s))
return 0;
else
return 1;
}
int intval(char s[])  //to validate integers
{
int k =0;
for(int i =0; i<strlen(s);i++)
{
if(!(isdigit(s[i])))
k++;
}
if(k!=0)
return 0;
else
{
int n =atoi(s);
return n;
}
}


class passenger
{
int ticket;
char nm[30];
char sex[5];
int age;
char nationality[30];
public:
int seatn;
passenger()  //constructor of class passenger
{
ticket=0;
strcpy(nm,"NULL");
strcpy(sex,"NULL");
strcpy(nationality,"NULL");
```

```
age=0;
seatn=0;
}
void getdata(); //to accept values for data members of class passengers
void showdata()//to diplay the datamembers of class passengers
{ cout<<"\n\t\tFLIGHT DETAILS~\n";
cout<<"\n\t\t------------------------------"<<endl;
cout<<"\n       NAME: ";puts(nm);
cout<<"\n        AGE: "<<age;
cout<<"\n        SEX: ";puts(sex);
cout<<"\nNATIONALITY: ";puts(nationality);
cout<<"\n   SEAT NO.: "<<seatn;
cout<<"\n    TICKETS: "<<ticket;
cout<<"\n------------------------------"<<endl;
}
void reset() // to initialize values of all datamembers to null
{
ticket=0;
strcpy(nm,"NULL");
strcpy(sex,"NULL");
strcpy(nationality,"NULL");
age=0;
seatn=0;
}
char *retname()  //accessor funtion to return nm
{
return nm;
}
int getticket() //accessor funtion to return ticket
{
return ticket;
}
};
void passenger::getdata()
{
randomize();
ticket = random(500-10+1)+10;
cout<<"\n\t\tTICKET: "<<ticket;
L:
cout<<"\n\t\tEnter your name  :  ";
char s[30];
gets(s);
int n = strval(s);
```

```
if(n==1)
strcpy(nm,s);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
getch();
goto L;
}
L2:
cout<<"\n\t\tEnter your age  :  ";
char s1[30];
gets(s1);
age = intval(s1);
if(age==0)
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
getch();
goto L2;
}
L3:
cout<<"\n\t\tEnter your sex (M/F) :  ";
char s2[30];
gets(s2);
int n2= strval(s2);
if(n2 == 1&&(strcmpi(s2,"m")==0 || strcmpi(s2,"f")==0))
strcpy(sex,s2);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
getch();
goto L3;

}
L4:
cout<<"\n\t\tEnter your nationality (ind/other) :  ";
char s3[30];
gets(s3);
int n3=strval(s3) ;
if(n3==1)
strcpy(nationality,s3);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
```

```
getch();
goto L4;

}
}
class airline
{ int fno;
  char n[30];
  int dt;
  float cost;
  char des[30];
  int d,m,y;
  public:
  int sb;
  passenger s[10];
  int getfnum() //to return fno
  {
    return fno;
  }
  airline()//contructor for class airline
 {
   fno=0;
   strcmpi(n,NULL);
   strcmpi(des,NULL);
   cost=0;
   d=m=y=0;
   dt=0;
  }
   void addf();//to accept values of all data members of class airline
   void modify();//to accept values of all dtat members of class airline for modification
   void displayf() //to display all values of datamembers of class airline in tabular
  {

    cout<<endl<<setw(3)<<fno<<setw(18)<<n<<"
"<<setw(4)<<d<<"/"<<setw(3)<<m<<"/"<<setw(5)<<y<<setw(7)<<"
"<<des<<setw(5)<<cost;
  }
  void displayt(int j)//to diplay data members in non tabular form
 {
   cout<<"\t\t";
   cout<<"\n-------------------------------";
   cout<<"\n\t  flight number:"<<fno;
   cout<<"\n\t  flight   name:"<<n;
```

```
  cout<<"\n\t  Passenger name:"<<s[j].retname();
  cout<<"\n\t    Destination:"<<des;
  cout<<"\n\t Departure Time:"<<dt;
  cout<<"\n\tdate of journey:"<<d<<"/"<<m<<"/"<<y;
  cout<<"\n\t    seat number:"<<s[j].seatn;
  cout<<"\n\t  ticket number:"<<s[j].getticket();
  cout<<"\n\t           cost:"<<cost;
  cout<<"\n";
  cout<<"-------------------------------";
 }
 void cancelticket(int j)//to null all values of s[j]
 {
  s[j].reset();
 }
 int returnfno()// accessor function to return fno
 {
   return fno;
 }
 char*returndes() // accessor function to return des
 {
   return des;
 }
 void displaypass();//to diplay all datamembers of containership class
}A;


void airline::modify()
{

cout<<"\n\t\tFlight no: "<<fno;
Z1:
cout<<"\n\t\tEnter flight name:  ";
char q[30];gets(q);
int n4 =strval(q);
if(n4==1)
strcpy(n,q);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
getch();
goto Z1;
}
Z2:
```

```
cout<<"\n\t\tEnter the departure time (24 hours format):  ";
char q1[30];gets(q1);
dt = intval(q1);
if(dt == 0||dt>24)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n\t\t";
getch();
goto Z2;
}
Z3:
cout<<"\n\t\tEnter flight destination:  ";
char q2[30];gets(q2);
int n5 =strval(q2);
if(n5==1)
strcpy(des,q2);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n\t\t";
getch();
goto Z3;
}XX:
cout<<"\n\t\t----------ENTER TIME----------\n\t\t";
Z4:
cout<<"\n\t\t Enter day  : ";
char q4[30];gets(q4);
d = intval(q4);
if(d<1 || d>31)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n\t\t";
getch();
goto Z4;
}
Z5:
cout<<"\n\t\tEnter months  :  ";
char q5[30];gets(q5);
m = intval(q5);
if(m<1|| m >12)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();

goto Z5;
}
```

```
Z6:
cout<<"\n\t\tEnter year(from 2018-2025) :  ";
char q6[30];gets(q6);
y = intval(q6);
if(y<2018 || y>2025)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto Z6;
}
if(!(((m==1 ||m==3||m==5||m==7||m==8||m==10||m==12)&&(d>0 &&
d<=31))||!((m==4||m==6||m==9||m==11)&&(d>0&&d<31))||!(m==2&&(d>0&&d<2
9))))
{cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto XX;
}
Z7:
cout<<"\n\t\tEnter cost  :  ";
char q7[30];gets(q7);
cost = intval(q7);
if(cost == 0)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto Z7;
}
}

void airline::addf()
{
randomize();
fno = random(859-10+1)+10;
cout<<"\n\t\tFlight no: "<<fno;
G1:
cout<<"\n\t\tEnter flight name:  ";
char q[30];gets(q);
int n4 =strval(q);
if(n4==1)
strcpy(n,q);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n";
```

```
getch();
goto G1;
}
G2:
cout<<"\n\t\tEnter the departure time(24 hour format):  ";
char q1[30];gets(q1);
dt = intval(q1);
if(dt == 0|| dt>24)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto G2;
}
G3:
cout<<"\n\t\tEnter flight destination:  ";
char q2[30];gets(q2);
int n5 =strval(q2);
if(n5==1)
strcpy(des,q2);
else
{
cout<<"\n\t\tINVALID INPUT!  Press any key to re-enter\n";
getch();
goto G3;
}XY:
cout<<"\n\t\t----------ENTER TIME----------\n";
G4:
cout<<"\n\t\tEnter day  :  ";
char q4[30];gets(q4);
d = intval(q4);
if(d<1 || d>31)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n\n\t\t";
getch();
goto G4;
}
G5:
cout<<"\n\t\tEnter months  :  ";
char q5[30];gets(q5);
m = intval(q5);
if(m<1|| m >12)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
```

```
getch();

goto G5;
}
G6:
cout<<"\n\t\tEnter year(from 2018-2025):  ";
char q6[30];gets(q6);
y = intval(q6);
if(y<2018 || y>2025)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto G6;
}
if(!(((m==1 ||m==3||m==5||m==7||m==8||m==10||m==12)&&(d>0 &&
d<=31))||((m==4||m==6||m==9||m==11)&&(d>0&&d<31))||(m==2&&(d>0&&d<29)
)))
{cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto XY;
}
G7:
cout<<"\n\t\tEnter cost  :  ";
char q7[30];gets(q7);
cost = intval(q7);
if(cost == 0)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto G7;
}
}
void airline::displaypass()
{
  int u=0;
  for(int b=0;b<10;++b)
  {
   if(s[b].seatn!=0)
   {
    if(u==0)
    {
     cout<<endl<<"\t\t
"<<setw(10)<<"Ticket_No"<<setw(20)<<"Name"<<setw(10)<<"Seat_No";
```

```
     cout<<endl<<"\t\t   ----------------------------------------------";
    }
    ++u;
    cout<<endl<<"\t\t
"<<setw(10)<<s[b].getticket()<<setw(20)<<s[b].retname()<<setw(10)<<s[b].seatn;
   }
  }
}
void bookticket() // to book a ticket
{
int a;
int f1 =0;
A1:
cout<<"\n\t\tEnter flight number  :\n";
char q8[30];gets(q8);
a = intval(q8);
if(a == 0)
{
cout<<"\n\t\tINVALID INPUT! Press any key to re-enter\n";
getch();
goto A1;
} fstream f;
 f.open("flight.dat",ios::in|ios::out|ios::binary);

        if(!f){cout<<"ERROR";exit(1);}
        while(!f.eof())
        {
         int poo = f.tellg();
         f.read((char*)&A,sizeof(A));
         if(f.eof())break;
      if(A.returnfno() == a)
         {
          f1 = 1;
          if(A.sb ==10)
                    {
                     cout<<"\n\n\t\t Seats are full\n";
                       getch();
                       break;

                    }
          else
          {
           for(int j = 0; j<10; j++)
```

```
                {
                 if(A.s[j].seatn==0)
                  {
                   A.s[j].getdata();
                   A.s[j].seatn = j+1;
                   A.sb++;
                   f.seekp(poo);
                   f.write((char*)&A,sizeof(A));
                   cout<<"\n\n\t\t Seat Booked\n";
                   cout<<"\n\t\tPress 'y' key and press enter to book another seat\n";
                   cout<<"\n\t\tPress any other key and press enter to go to menu\n";
                   char ch;
                   cin>>ch;
                   if(ch!='y')
                   break;

                  }
                }
          }
 } }
f.close();
if(f1==0)
cout<<"\n\n\t\t FLIGHT NOT FOUND\n";

}
void header() //to display the default header part of the program
{
clrscr();
textmode(3);
textbackground(GREEN);
textcolor(BLUE);
cprintf( "   \b\b\b\b                                        ");
cprintf( "                                                 ");
cprintf( " AIRLINE RESERVATION SYSTEM                          ");
cprintf( "                                                      ");
textmode(3);
 textbackground(WHITE);
 textcolor(BLUE);
}
 void main()   //main funtions
{
clrscr();
 int ch;
```

```
int gd=DETECT,gm;

initgraph(&gd,&gm,"c:\\tc\\bgi");

setbkcolor(EGA_GREEN);
settextstyle(0,0,4.5);
 rectangle(20,20,620,400);
outtextxy(230,100,"AIRLINE");
outtextxy(30,200,"RESERVATION SYSTEM");

settextstyle(0,0,2.0);
outtextxy(300,440,"Project done by ");
outtextxy(350,460,"Pradeep & Anirudh ");
delay(3500);

textmode(3);
textbackground(WHITE);
textcolor(BLUE);
int n=0;
A2:
header();

gotoxy(15,8);
textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);
cprintf("                    MENU                    ");
textmode(3);
textbackground(WHITE);
textcolor(BLUE);
 gotoxy(15,9);
cprintf("1.Add Flight");
 gotoxy(15,10);
cprintf("2.Modify Flight");
 gotoxy(15,11);
cprintf("3.List all Flights");
 gotoxy(15,12);
cprintf("4.List all flight to a destination");
 gotoxy(15,13);
cprintf("5.Delete a Flight");
 gotoxy(15,14);
cprintf("6.Book a ticket");
```

```
 gotoxy(15,15);
 cprintf("7.View a ticket");
  gotoxy(15,16);
 cprintf("8.Delete a Ticket");
  gotoxy(15,17);
 cprintf("9.List all passengers of a flight");
  gotoxy(15,18);
 cprintf("10.To exit");
 gotoxy(15,19);
 cprintf("ENTER YOUR CHOICE");
 cout<<endl;  cout<<"\t\t\t";
char cha[1];gets(cha);
ch = intval(cha);
if(ch<1 || ch>10)
{
cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

getch();
goto A2;
}
 switch(ch)
{
case 1: clrscr();
      header();
      fout.open("flight.dat",ios::app|ios::binary);
      if(!fout){cout<<"ERROR";exit(1);}
      textmode(3);
 textbackground(EGA_BLUE);
 textcolor(WHITE);
   gotoxy(17,6);
       cprintf("ADDING NEW FLIGHT");
      A.sb=0;
      textmode(3);
 textbackground(WHITE);
 textcolor(BLUE);
      A.addf();
      fout.write((char*)&A,sizeof(A));
         gotoxy(17,6); clrscr();header();
      cprintf("FLIGHT ADDED SUCCESSFULY");
      n++;
      gotoxy(17,7);
      cprintf("Press any key to go to menu ");
                          fout.close();
```

```
      getch();
      goto A2;

case 2:clrscr();
      header();
       if(n==0)                                    {
       cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

       getch();
       goto A2;}
       textmode(3);
 textbackground(EGA_BLUE);
 textcolor(WHITE);              gotoxy(17,7);
       cprintf("MODIFYING FLIGHT");
       textmode(3);
 textbackground(WHITE);
 textcolor(BLUE);
       int c;
       A3:
       cout<<"\n\t\tEnter flight number\n";
       char ert[30];
       gets(ert);
       c = intval(ert);
       if(c==0)
       {
       cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

       getch();
       goto A3;
       }int k =0;
       fin.open("flight.dat",ios::in|ios::binary);
       fout.open("temp.dat",ios::app|ios::binary);
       if(!fin||!fout){cout<<"ERROR";exit(1);}
       while(!fin.eof())
       {
       fin.read((char*)&A,sizeof(A));
       if(fin.eof())break;
       if(A.returnfno() == c)
       {
       k++;
      A.modify();

      fout.write((char*)&A,sizeof(A));
```

```
    }
    else
            fout.write((char*)&A,sizeof(A));

    }
    fout.close();
    fin.close();
    if(k==0)                {
    cout<<"\n\t\tRECORD NOT FOUND\n";
    cout<<"\n\t\tPRESS ANY KEY TO EXIT TO MENU\n";

    getch();
    }
    else
    { remove("flight.dat");
    rename("temp.dat","flight.dat" );
     gotoxy(17,7);   clrscr();
        cprintf("DETAILS MODIFIED SUCCESFULLY");

cout<<endl;
 cout<<"\n\t\t\t Press any key to exit to main menu\n";
       } getch();
    goto A2;
   case 3: clrscr(); header();
    if(n==0)                                {
     cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";
         getch();
     goto A2;}
   textmode(3);
 textbackground(EGA_BLUE);
 textcolor(WHITE);      gotoxy(17,7);
          cprintf("DISPLAYING DETAILS...");
        textmode(3);
 textbackground(WHITE);
 textcolor(BLUE);
        cout<<endl<<setw(3)<<"FNo."<<setw(18)<<"NAME"<<"
"<<setw(12)<<"DATE"<<setw(7)<<" "<<"DEST."<<setw(5)<<"COST";
       fin.open("flight.dat",ios::in|ios::binary);
       if(!fin){cout<<"ERROR";exit(1);}
       while(!fin.eof())
       {
        fin.read((char*)&A,sizeof(A));
        if(fin.eof())break;
```

```
        A.displayf();cout<<endl;
       }

        cout<<"\n\t\tPress any key to go to main\n";
        fin.close();
        getch();
        goto A2;
   case 4:clrscr();  header();
     if(n==0)                                  {
     cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

     getch();
     goto A2;}
       A8:
       cout<<"\n\t\tEnter Destination: \n";
       char o[30];char o1[30];
       gets(o1);
       int w = strval(o1);
       if(w == 1)
       strcpy(o,o1);
       else
       {
       cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n" ;

       getch();
       goto A8;
       }
      textmode(3);
 textbackground(EGA_BLUE);
 textcolor(WHITE);      gotoxy(17,7); clrscr();header();
        cprintf("DISPLAYING DETAILS OF FLIGHTS");
        textmode(3);
 textbackground(WHITE);
 textcolor(BLUE);
              cout<<endl<<setw(3)<<"FNo."<<setw(18)<<"NAME"<<"
"<<setw(12)<<"DATE"<<setw(7)<<" "<<"DEST."<<setw(5)<<"    "<<"COST";
       int u=0;
        fin.open("flight.dat",ios::in|ios::binary);
       if(!fin){cout<<"ERROR";exit(1);}
       while(!fin.eof())
       {
        fin.read((char*)&A,sizeof(A));
        if(fin.eof())break;
```

```
 if(strcmpi(o,A.returndes())==0)
{u++;
A.displayf();
 }
 }
 if(u==0)
 {clrscr();
       header();

 cout<<"\n\t\t RECORD NOT FOUND.\n";

 }
 cout<<"\n\t\tPress any key to go to main\n";
                fin.close();
 getch();
 goto A2;

case 5:   clrscr();header();
  if(n==0)                                {
 cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

 getch();
 goto A2;}
 textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);          gotoxy(17,7);
 cprintf("DELETING FLIGHT");
 textmode(3);
textbackground(WHITE);
textcolor(BLUE);
 int c1;
 Am:
 cout<<"\n\t\tEnter flight number\n";
 char ert1[30];
 gets(ert1);
 c1 = intval(ert1);
 if(c1==0)
 {
 cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

 getch();
 goto Am;
 }
```

```
    int k1 =0;
    fin.open("flight.dat",ios::in|ios::binary);
    fout.open("temp.dat",ios::app|ios::binary);
    if(!fin||!fout){cout<<"ERROR";exit(1);}
    while(!fin.eof())
    {
    fin.read((char*)&A,sizeof(A));
    if(fin.eof())break;
    if(A.returnfno() != c1)
    {
    k1++;


   fout.write((char*)&A,sizeof(A));
   }



   }
   fout.close();
   fin.close();
   if(k1==0)                {
   cout<<"\n\t\tRECORD NOT FOUND\n";
   cout<<"\n\t\tPRESS ANY KEY TO EXIT TO MENU\n";

   getch();
   }
   else
   { remove("flight.dat");
   rename("temp.dat","flight.dat" );
    gotoxy(17,7);   clrscr();
        cprintf("RECORDS DELETED SUCCESFULLY");

cout<<endl;
 cout<<"\n\t\t\t Press any key to exit to main menu\n";
       } getch();
    goto A2;

 case 6: clrscr();    header();
 if(n==0)                                {
    cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

    getch();
    goto A2;}
```

```
    textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);      gotoxy(17,7);
    cprintf("BOOKING TICKETS...\n");
    textmode(3);
textbackground(WHITE);
textcolor(BLUE);
      bookticket();



      getch();
      goto A2;
 case 7: clrscr();    header();
     if(n==0)                                    {
     cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

     getch();
     goto A2;}
      An:
      int c2=0;
     cout<<"\n\t\tEnter flight number\n";
     char ert2[30];
     gets(ert2);
     c2 = intval(ert2);
     if(c2==0)
     {
     cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

     getch();
     goto An;
     }
    A4:
     cout<<"\n\t\tEnter Ticket no.: \n";
     char et[30];
     gets(et);
     int tic = intval(et);
     if(tic==0)
     {
     cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

     getch();
     goto A4;
```

```
    } int km=0;     int kk=0;
    textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);     gotoxy(17,7);   clrscr();
    cprintf("DISPLAYING DETAILS...");
    textmode(3);
textbackground(WHITE);
textcolor(BLUE);
    fin.open("flight.dat",ios::in|ios::binary);
      if(!fin){cout<<"ERROR";exit(1);}
      while(!fin.eof())
      {
       fin.read((char*)&A,sizeof(A));
       if(fin.eof())break;

    if(A.returnfno()== c2)
    {
              kk++;
              for(int n=0; n<10;n++)
              {
                if(tic==A.s[n].getticket())
                     {km++;
                     A.displayt(n);
                      }
              }
    }
    }
    if(km==0)
    {
    cout<<"\n\t\tTICKET NOT FOUND\n" ;
    }
    else if(kk==0)
    cout<<"\n\t\tFLIGHT NOT FOUND\n";
    cout<<"\n\t\tPress any key to go to menu\n ";
                                  fin.close();
    getch();
    goto A2;

 case 8: clrscr();   header();
    if(n==0)                                  {
    cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

    getch();
```

```
      goto A2;}
      textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);      gotoxy(17,7);
      cprintf("CANCEL TICKETS");
      textmode(3);
textbackground(WHITE);
textcolor(BLUE);  int c3;
     Ak:
      cout<<"\n\t\tEnter flight number\n";
      char ert3[30];
      gets(ert3);
      c3 = intval(ert3);
      if(c3==0)
      {
      cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

      getch();
      goto Ak;
      }
     A5:
      cout<<"\n\t\tEnter Ticket no.: \n";
      char tt[30];
      gets(tt);
      int ticc = intval(tt);
      if(ticc==0)
      {
      cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";

      getch();
      goto A5;
      } int  k2=0;  int ky=0;fstream ff;
     ff.open("flight.dat",ios::in|ios::out|ios::binary);

        if(!ff){cout<<"ERROR";exit(1);}
        while(!ff.eof())
        {   int poss= ff.tellg();
         ff.read((char*)&A,sizeof(A));
         if(ff.eof())break;
         if(A.returnfno()== c3)
      {
                  ky++;
                  for(int n=0; n<10;n++)
```

```
                {
     if(ticc==A.s[n].getticket())
     {
     k2++;
     A.cancelticket(n);
     A.sb--;
     ff.seekp(poss);


    ff.write((char*)&A,sizeof(A));
     }

     }
     }
     }


     if(kk==0)
    { cout<<"\n\t\tFLIGHT NOT FOUND\n"; getch();break;}
     if(k2==0)
     {
     cout<<"\n\t\tTICKET NOT FOUND\n" ;getch();break;  }

     cout<<"\n\t\tRECORD DELETED\n";
     cout<<"\n\t\tPress any key to go to menu\n ";
     ff.close();
     getch();
     goto A2;

case 9:clrscr();     header();
    if(n==0)                                {
     cout<<"\n\t\tNO ENTRIES.Press any key to go to menu\n";

     getch();
     goto A2;}
       A6:
       cout<<"\n\t\tEnter Flight no.: \n";
       int b;char b1[30];
       gets(b1);
       b = intval(b1);
       if(b == 0)
                {
       cout<<"\n\t\tINVALID INPUT! press any key to re-enter\n";
```

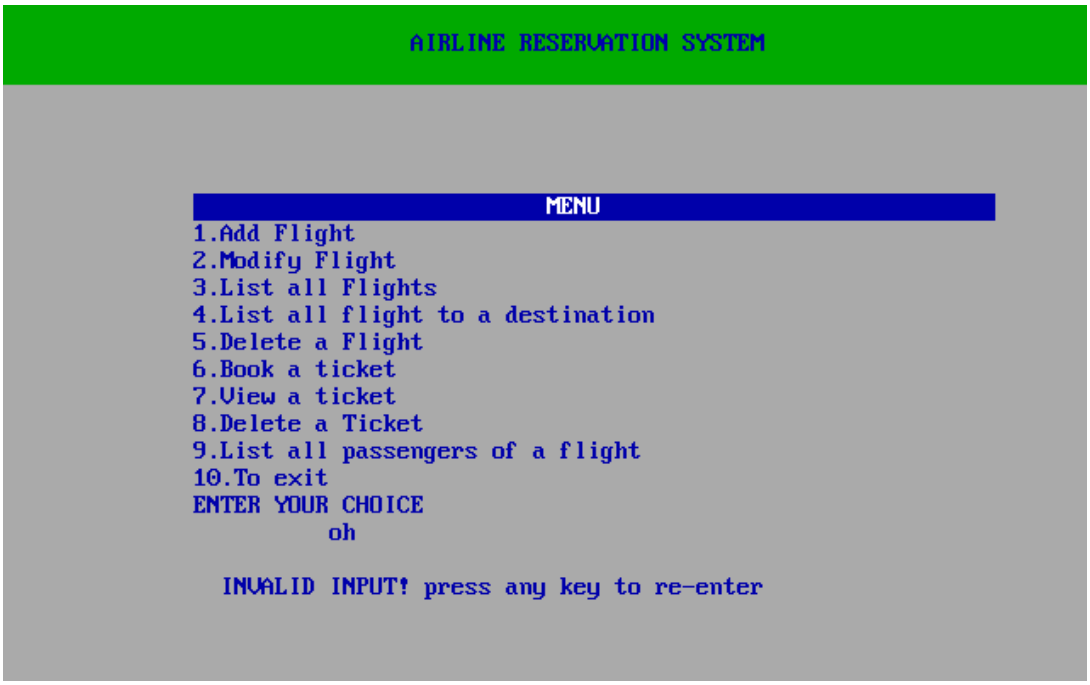
```
        getch();
        goto A6;
        }
       int uu=0;
     textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);     gotoxy(17,7);              clrscr();   header();
       cprintf("DISPLAYING DETAILS OF PASSENGER");
       textmode(3);
textbackground(WHITE);
textcolor(BLUE);
        cout<<endl;int ccc=0;
        fin.open("flight.dat",ios::in|ios::binary);
       if(!fin){cout<<"ERROR";exit(1);}
       while(!fin.eof())
       {
        fin.read((char*)&A,sizeof(A));
        if(fin.eof())break;
       if(b==A.returnfno())
       { uu++;
                if(A.sb!=0)
                { ccc++;
                 A.displaypass();  }
        }
        }
        if(uu==0||ccc==0)
        {clrscr();
              header();

        cout<<"\n\t\tRECORD NOT FOUND.\n";

        }

        cout<<"\n\t\tPress any key to go to main\n";
                                                          fin.close();
        getch();
        goto A2;
case 10:   clrscr();          textmode(3);
textbackground(EGA_BLUE);
textcolor(WHITE);     gotoxy(17,7);
     cprintf("PRESS ANY KEY TO EXIT\n"); textmode(3);
textbackground(WHITE);   remove("flight.dat");
```

```
textcolor(BLUE);
      getch();
      cout<<"\n\t\t...Exitting...\n";   delay(1000);
      exit(0);
      break;
default:
     cout<<"\n\t\tINVALID INPUT! Press any key to go to menu\n";

     getch();
     goto A2;
     }



   }
```

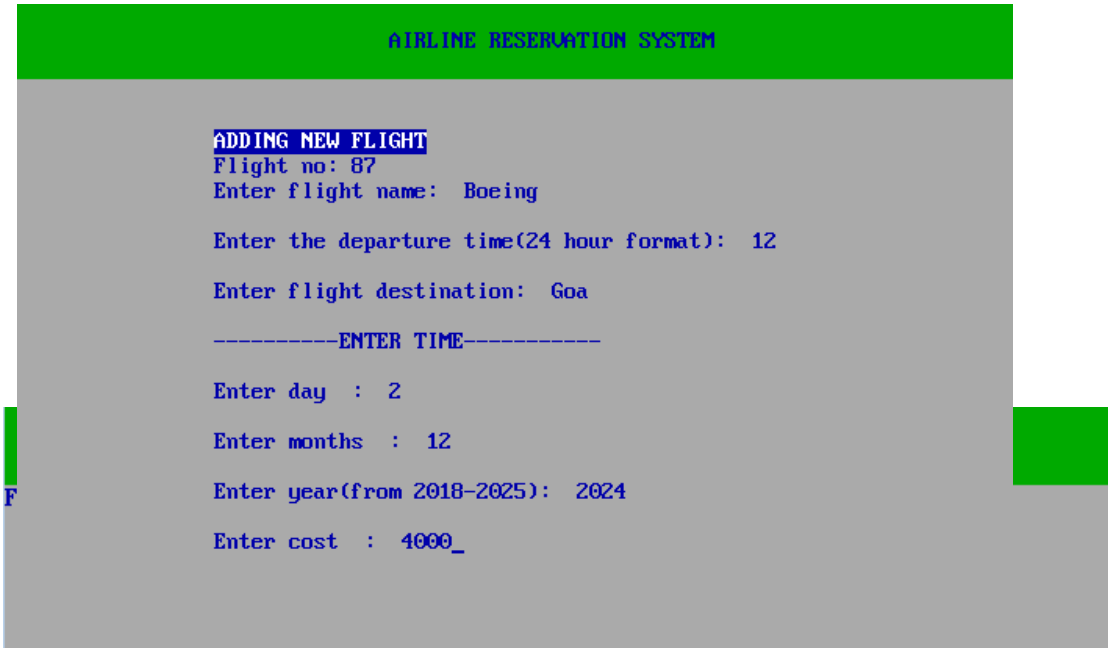
IX. Output Screen

TITLE SCREEN

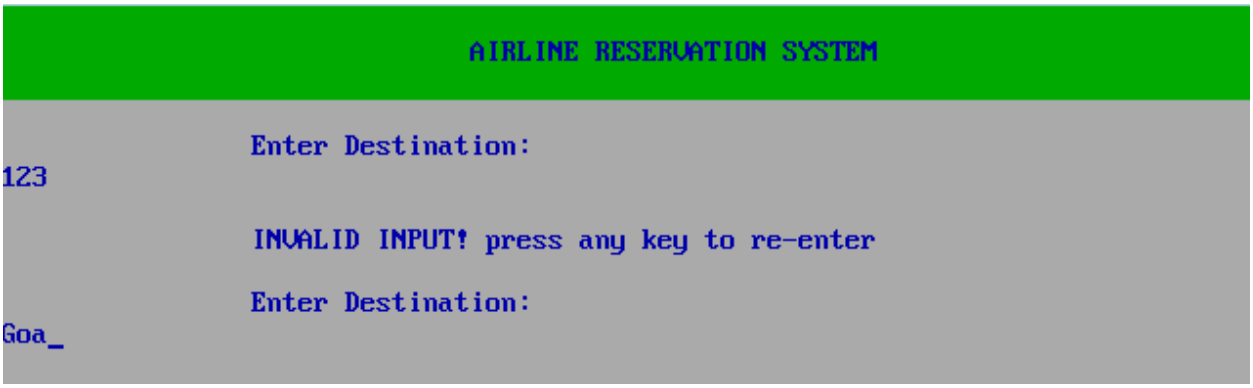
MAIN MENU/ MAIN MENU SCREEN VALIDATION



ADDING A FLIGHT



LIST FLIGHTS TO A DESTINATION

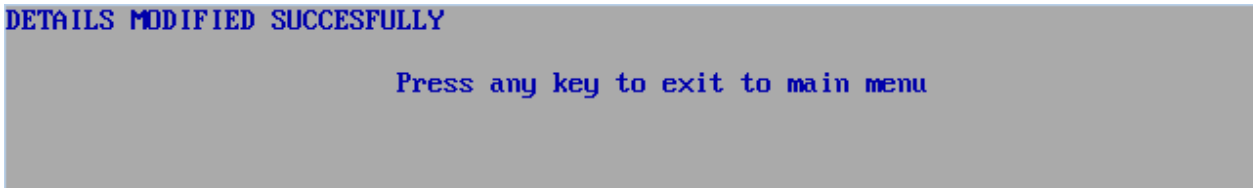


MODIFYING A FLIGHT

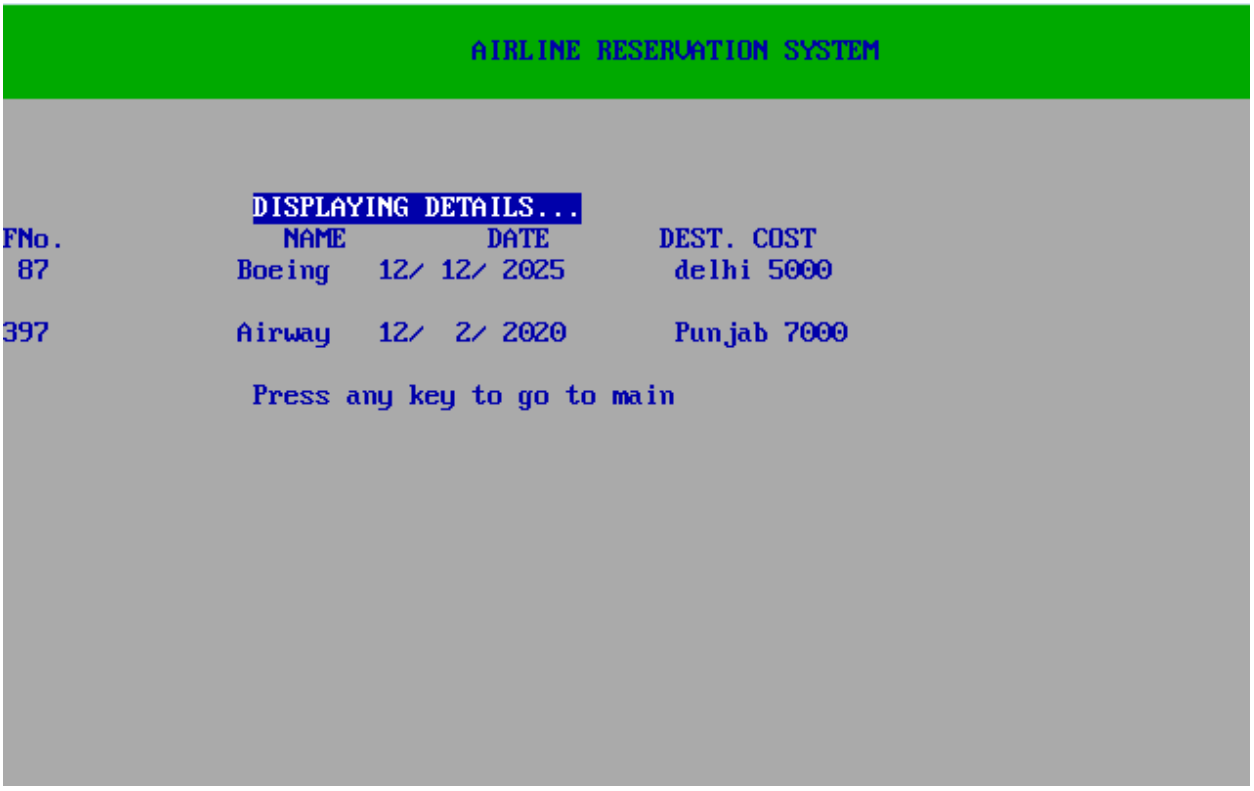

AIRLINE RESERVATION SYSTEM
MODIFYING FLIGHT
Enter flight number
123
RECORD NOT FOUND
PRESS ANY KEY TO EXIT TO MENU

```
MODIFYING FLIGHT
Enter flight number
87

Flight no: 87
Enter flight name:  Boeing

Enter the departure time (24 hours format):  11

Enter flight destination:  12

INVALID INPUT!  Press any key to re-enter

Enter flight destination:  delhi

----------ENTER TIME-----------

 Enter day  : qw

INVALID INPUT! Press any key to re-enter
```

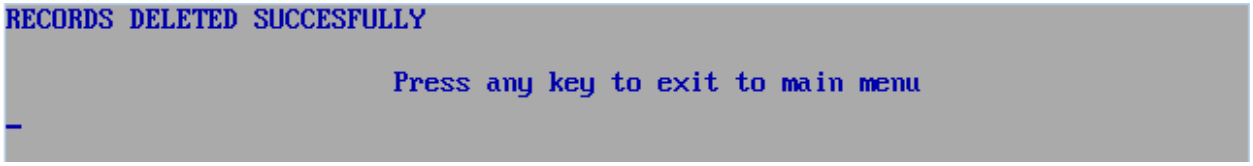
```
INVALID INPUT!  Press any key to re-enter

Enter flight destination:  delhi

----------ENTER TIME-----------

 Enter day  : qw

INVALID INPUT! Press any key to re-enter

 Enter day  : 12

Enter months  :  23

INVALID INPUT! Press any key to re-enter

Enter months  :  12

Enter year(from 2018-2025) :  2026

INVALID INPUT! Press any key to re-enter

Enter year(from 2018-2025) :  2025

Enter cost  :  5000_
```



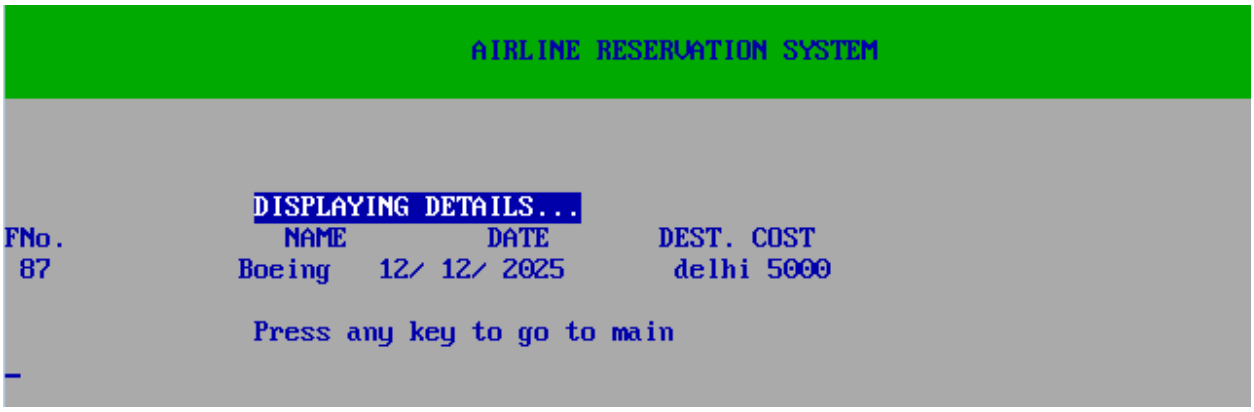
LIST ALL FLIGHTS



DELETING A FLIGHT



BOOKING A TICKET



```
                BOOKING TICKETS...

                Enter flight number  :
87

                TICKET: 89
                Enter your name  :  Pradeep

                Enter your age  :  16

                Enter your sex (M/F) :  M

                Enter your nationality (ind/other) :  ind

                 Seat Booked

                Press 'y' key and press enter to book another seat
```

OBJ

```
                        AIRLINE RESERVATION SYSTEM

                BOOKING TICKETS...

                Enter flight number  :
67

                 FLIGHT NOT FOUND
```

DISPLAY ALL PASSENGERS TO A FLIGHT

```
                        AIRLINE RESERVATION SYSTEM

                Enter Flight no.:
87
```

```
                    AIRLINE RESERVATION SYSTEM
DISPLAYING DETAILS OF PASSENGER

          Ticket_No          Name    Seat_No
          ------------------------------------------
                89           Pradeep        1
        Press any key to go to main
```

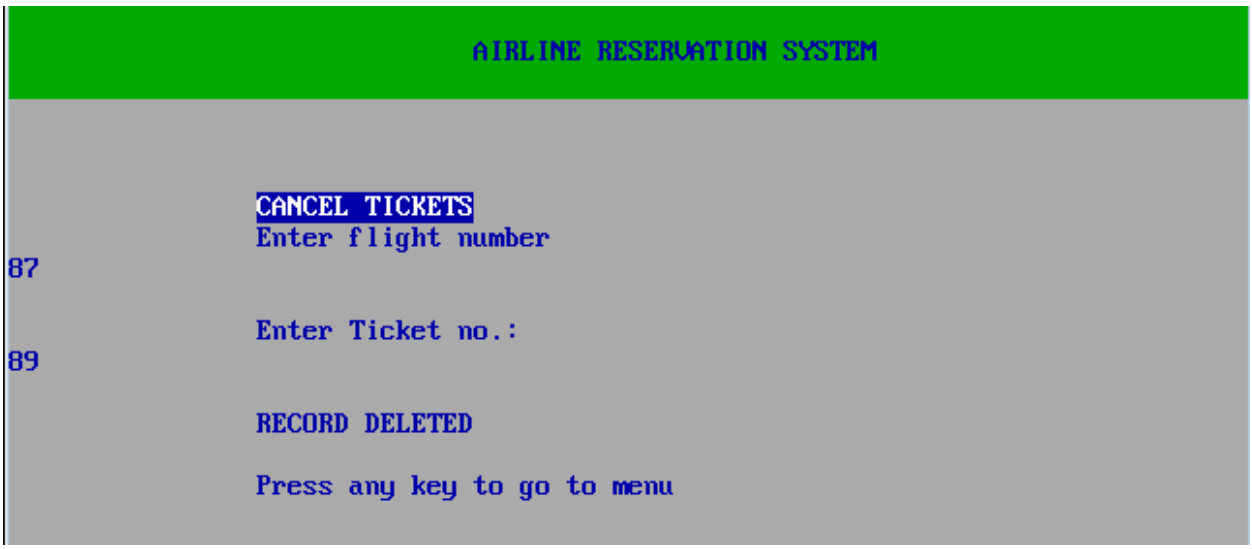
VIEW A TICKET

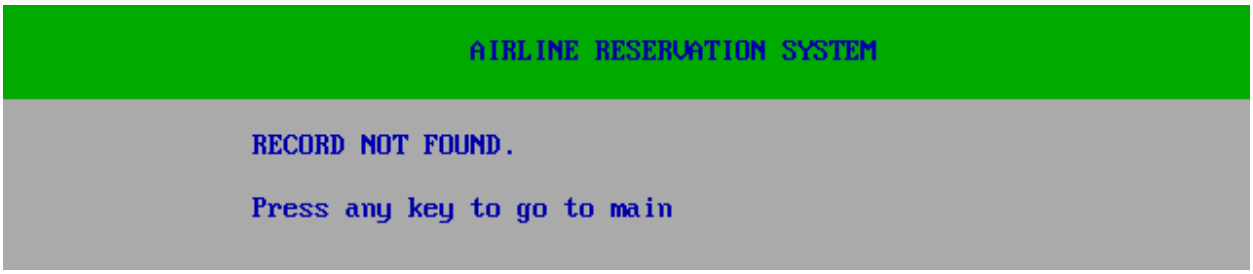
```
                    AIRLINE RESERVATION SYSTEM

                Enter flight number
87

                Enter Ticket no.:
89_
```

```
DISPLAYING DETAILS...
------------------------------
        flight number:87
        flight   name:Boeing
        Passenger name:Pradeep
          Destination:delhi
       Departure Time:11
      date of  journey:12/12/2025
         seat number:1
        ticket number:89
                cost:5000
------------------------------
            Press any key to go to menu
```

DELETING A TICKET

VIEW A TICKET





EXITTING SCREEN