**2015-16**

**Ashwin Joisa, Daivat Bhatt, Sagar B. Manjunath**

**Sri Kumaran Children’s Home**

**XII A**

**ATCAS – Air Traffic Control & Administration System**



# Acknowledgement

**We take this opportunity to thank all those who have helped us in completing this project.**

**We would like to thank Mrs. Kavitha for giving the opportunity to develop this project and for her wonderful assistance and encouragement.**

**This project would not have been completed without constant support of our parents.**

# Index

1. Certificate 1
2. Acknowledgement 2
3. Need for the project 4
4. About the project 5
5. Project requirements 6
6. Project flow diagram 7
7. Class diagram 8
8. Function description 9
9. Source code 11
10. Output screenshots 35
11. Scope for improvement 38
12. References 39

# Need For the Project:

This project was designed to simulate and understand the functioning and management of an airport. It helps us understand the management of an airport in complete automation. All the services of an airport would be completely managed by a computer program making human input very little and efficiency very high.

# About the Project:

The salient features of the project are:

* Complete automation – Everything on the airport is completely managed by the program. The program can run infinitely without any human input or intervention.
* Know what’s going on – The program keeps a list of all the activities that have occurred so far and notifies the user when he/she wishes to get updated. The number of notifications is also shown.

Find planes quickly – The search option helps the user to easily find the plane he/she is looking for. Quick and elaborate flight details make it user friendly

# Project Requirements

Hardware:

RAM – 256 MB

HDD - 512 MB

Pentium 4 Processor

Software:

Dev C++

# Project Flow Diagram

Exit

Search by Flight Status

Search by Destination

Search by Air Company

Gate

Status

Flight

Status

Baggage Claim

Status

Search

Notifications

Menu

Title Page

Start

# Class diagram

**Class gate:**

|  |  |
| --- | --- |
| Member name | Data type |
| gatename  airplaneno  status  get\_gatename(char c[])  get\_airplaneno(char c[])  get\_status(char c[]) | char [20]  char [7]  char [30]  **Class airplane:**  void  void  void |

|  |  |
| --- | --- |
| Member name | Data type |
| flight\_no  status  destination  obj\_tofile()  statusmod()  freer()  initcounter()  get\_flight\_no(char c[])  get\_status(char c[])  get\_destination(char c[]) | char [7]  char [25]  char [50]  void  void  void  void  void  void  void |

**Class baggage\_claim:**

|  |  |
| --- | --- |
| Member name | Data type |
| cname  airplaneno  status  get\_name(char c[])  get\_airplaneno(char c[])  get\_status(char c[]) | char [20]  char [7]  char [30]  void  void  void |

# Function description

1. gate::get\_gatename(char c[]) – Copies the gatename to another character array passed as a parameter.
2. gate::get\_airplaneno(char c[]) – Copies the airplaneno to another character array passed as a parameter.
3. gate::get\_status(char c[]) – Copies the status of the gate to another character array passed as a parameter.
4. baggage\_claim:: get\_name(char c[]) – Copies the companyname to another character array passed as a parameter.
5. baggage\_claim::get\_airplaneno(char c[]) – Copies the airplaneno to another character array passed as a parameter.
6. baggage\_claim::get\_status(char c[]) – Copies the status of the baggage claim to another character array passed as a parameter.
7. airplane()– Constructor that initializes the details of the plane entering the airport.
8. airplane::obj\_tofile()–Writes the details of the initialized plane to the file.
9. airplane::statusmod()–Periodically changes the status of the plane (for instance, from boarding to taxiing).
10. airplane::freer()–Searches for gates and baggage claims which are under maintenance and after some time, makes them available for use.
11. airplane::initcounter()–Initializes the variable called counter to 0.
12. airplane::get\_flight\_no(char c[]) – Copies the gatename to another character array passed as a parameter.
13. airplane::get\_status(char c[]) – Copies the status to another character array passed as a parameter.
14. airplane::get\_destination(char c[]) – Copies the destination of the gate to another character array passed as a parameter.
15. View\_Notifications() – Displays notifications.
16. disp\_menu() – Displays the menu.
17. gate\_status() – Displays gate status.
18. baggage\_claim\_status() – Displays the baggage claim status.
19. airplane\_status() – Displays the airplane status.
20. search() – calls functions search\_by\_aircompany(), search\_by\_status() and search\_by\_destination().
21. search\_by\_aircompany() – Searches for planes by aircompany.
22. search\_by\_status() – Searches for planes by status.
23. search\_by\_destination() – Searches for planes by destination.

# Source Code

#include <iostream>

#include <graphics.h>

#include <Windows.h>

#include <math.h>

#include <string.h>

#include <cstdlib>

#include <ctime>

#include <ctype.h>

#include "class\_airplane\_and\_queue.h"

using namespace std;

void initcounter();

void main\_page();

void start();

void disp\_menu();

void View\_Notifications();

void baggage\_claim\_status();

void gate\_status();

void airplane\_status();

void search();

void Search\_by\_aircompany();

void Search\_by\_destination();

void Search\_by\_status();

int main()

{

srand(time(0));

initcounter();

initwindow(1000, 700, "ATCAS");

main\_page();

disp\_menu();

system("pause");

}

void main\_page()

{

readimagefile("Cover.bmp", 0, 0, 1000, 700);

int a = getch();

}

#include <iostream>

#include <string.h>

#include <time.h>

#include <ctime>

#include <cstdlib>

#include <Windows.h>

#include <graphics.h>

#include <fstream>

using namespace std;

int counter=0;

void initcounter()

{

char str[100];

int i=0;

ifstream ifile("notifications.txt");

while(ifile)

{

ifile.getline(str, 100, '\n');

if(ifile.eof())

break;

i++;

}

ifile.close();

if(i>17)

counter=17;

else

counter=i;

}

class gate

{

public:

char gatename[20], airplaneno[7], status[30];

public:

void get\_gatename(char c[])

{

strcpy(c, gatename);

}

void get\_airplaneno(char c[])

{

strcpy(c, airplaneno);

}

void get\_status(char c[])

{

strcpy(c, status);

}

};

class baggage\_claim

{

public:

char cname[20], airplaneno[7], status[30];

public:

void get\_name(char c[])

{

strcpy(c, cname);

}

void get\_airplaneno(char c[])

{

strcpy(c, airplaneno);

}

void get\_status(char c[])

{

strcpy(c, status);

}

};

class airplane:public gate, public baggage\_claim

{

public:

airplane()

{

int x, y;

y=rand()%10+1;

x=rand()%9000+1000;

strcpy(status, "ARRIVING");

switch(y)

{

case 1:

flight\_no[0]='A';

flight\_no[1]='I';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 2:

flight\_no[0]='L';

flight\_no[1]='U';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 3:

flight\_no[0]='A';

flight\_no[1]='A';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 4:

flight\_no[0]='S';

flight\_no[1]='A';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 5:

flight\_no[0]='F';

flight\_no[1]='R';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 6:

flight\_no[0]='E';

flight\_no[1]='M';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 7:

flight\_no[0]='T';

flight\_no[1]='A';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 8:

flight\_no[0]='B';

flight\_no[1]='A';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 9:

flight\_no[0]='C';

flight\_no[1]='P';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

case 10:

flight\_no[0]='C';

flight\_no[1]='A';

flight\_no[2]=char(x%10+48);

x/=10;

flight\_no[3]=char(x%10+48);

x/=10;

flight\_no[4]=char(x%10+48);

x/=10;

flight\_no[5]=char(x%10+48);

flight\_no[6]='\0';

break;

}

x=int(rand()%10)+1;

switch(x)

{

case 1:

strcpy(destination, "New Delhi");

break;

case 2:

strcpy(destination, "Mumbai");

break;

case 3:

strcpy(destination, "Chennai");

break;

case 4:

strcpy(destination, "Kolkatta");

break;

case 5:

strcpy(destination, "London");

break;

case 6:

strcpy(destination, "Singapore");

break;

case 7:

strcpy(destination, "New York");

break;

case 8:

strcpy(destination, "Barcelona");

break;

case 9:

strcpy(destination, "Paris");

break;

case 10:

strcpy(destination, "Rio De Janeiro");

break;

}

}

public:

char flight\_no[7], status[25], destination[50];

public:

void obj\_tofile()

{

int x=rand()%100;

int ctr=0;

airplane airp;

ifstream ifile("flights.dat");

while(ifile)

{

ifile.read((char\*)&airp,sizeof(airp));

if(ifile.eof())

break;

ctr++;

}

ifile.close();

if(x<=50 && ctr<14)

{

airplane obj;

ofstream ofile("flights.dat", ios::app);

ofile.write((char\*)&obj, sizeof(obj));

ofile.close();

}

}

void statusmod()

{

char str1[5];

str1[0]='(';

str1[3]=')';

char strin[100];

airplane p;

int i;

ofstream note("notifications.txt", ios::app);

ifstream ifile("flights.dat");

ofstream ofile("flights2.dat");

while(ifile)

{

strcpy(strin, "Flight ");

ifile.read((char\*)&p,sizeof(p));

if(ifile.eof())

break;

strcat(strin, p.flight\_no);

int x=rand()%100;

if(!(strcmp(p.status,"ARRIVING")))

{

if(x>=60)

{

gate g;

bool found=false;

strcpy(p.status,"TAXIING FROM ARRIVAL");

strcat(strin," has TAXIED FROM ARRIVAL.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

ifstream ifile2("gates.dat");

ofstream ofile2("gates2.dat");

while(ifile2)

{

ifile2.read((char\*)&g,sizeof(g));

if(ifile2.eof())

break;

if((strcmp(g.status,"AVAILABLE")==0)&&(found==false))

{

strcpy(g.status,"IN USE");

strcpy(g.airplaneno,p.flight\_no);

found=true;

}

ofile2.write((char\*)&g,sizeof(g));

}

ofile2.close();

ifile2.close();

remove("gates.dat");

rename("gates2.dat","gates.dat");

}

}

else if(strcmp(p.status,"TAXIING FROM ARRIVAL")==0)

{

if(x>=60)

{

strcpy(p.status,"DEBOARDING");

strcat(strin," has DEBOARDED.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

baggage\_claim b;

ifstream ifile2("bclaim.dat");

ofstream ofile2("bclaim2.dat");

bool found=false;

while(ifile2)

{

ifile2.read((char\*)&b,sizeof(b));

if(ifile2.eof())

break;

if(strcmp(b.status,"AVAILABLE")==0&&(found==false))

{

strcpy(b.status,"IN USE");

strcpy(b.airplaneno,p.flight\_no);

found=true;

}

ofile2.write((char\*)&b,sizeof(b));

}

ifile2.close();

ofile2.close();

remove("bclaim.dat");

rename("bclaim2.dat","bclaim.dat");

}

}

else if(strcmp(p.status,"DEBOARDING")==0)

{

if(x>=92)

{

strcpy(p.status,"REFUELLING");

strcat(strin," has REFUELLED.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

}

}

else if(strcmp(p.status,"REFUELLING")==0)

{

if(x>=77)

{

baggage\_claim g;

strcpy(p.status,"BOARDING");

strcat(strin," has FINISHED BOARDING.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

bool found=false;

int x;

ifstream ifile2("bclaim.dat");

ofstream ofile2("bclaim2.dat");

while(ifile2)

{

x = rand()%20+1;

ifile2.read((char\*)&g,sizeof(g));

if(ifile2.eof())

break;

if((strcmp(g.airplaneno,p.flight\_no)==0)&&(found==false))

{

if(x<=5)

{

strcpy(g.status,"UNDER MAINTENANCE");

strcpy(g.airplaneno,"");

}

else

{

strcpy(g.status,"AVAILABLE");

strcpy(g.airplaneno,"");

}

found=true;

}

ofile2.write((char\*)&g,sizeof(g));

}

ofile2.close();

ifile2.close();

remove("bclaim.dat");

rename("bclaim2.dat","bclaim.dat");

}

}

else if(strcmp(p.status,"BOARDING")==0)

{

if(x>=70)

{

gate g;

strcpy(p.status,"TAXIING FOR DEPARTURE");

strcat(strin," has TAXIED FOR DEPARTURE\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

bool found=false;

int x;

ifstream ifile2("gates.dat");

ofstream ofile2("gates2.dat");

for(int i=0;i<14;i++)

{

x =int(rand()%20)+1;

ifile2.read((char\*)&g,sizeof(g));

if(ifile2.eof())

break;

if((strcmp(g.airplaneno,p.flight\_no)==0)&&(found==false))

{

if(x<=5)

{

strcpy(g.status,"UNDER MAINTENANCE");

strcpy(g.airplaneno,"");

}

else

{

strcpy(g.status,"AVAILABLE");

strcpy(g.airplaneno,"");

}

found=true;

}

ofile2.write((char\*)&g,sizeof(g));

}

ofile2.close();

ifile2.close();

remove("gates.dat");

rename("gates2.dat","gates.dat");

}

}

else if(strcmp(p.status,"TAXIING FOR DEPARTURE")==0)

{

if(x>=80)

{

strcpy(p.status,"HOLD SHORT RUNWAY");

strcat(strin," is HOLDING SHORT OF RUNWAY.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

}

}

else if(strcmp(p.status,"HOLD SHORT RUNWAY")==0)

{

if(x>=60)

{

strcpy(p.status,"DEPARTED");

strcat(strin," has DEPARTED.\n");

if(counter<17)

{

counter++;

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

settextstyle(8,0,2);

outtextxy(575,70,str1);

}

}

}

else if(strcmp(p.status,"DEPARTED")==0)

continue;

ofile.write((char\*)&p,sizeof(p));

for(int k=0;k<strlen(strin);k++)

{

if(strin[13]!=' ')

break;

note<<strin[k];

}

}

note.close();

ofile.close();

ifile.close();

remove ("flights.dat");

rename("flights2.dat","flights.dat");

}

void freer()

{

baggage\_claim b;

gate g;

int x = rand()%99+1;

ifstream ifile1("gates.dat");

ofstream ofile1("gates2.dat");

ifstream ifile2("bclaim.dat");

ofstream ofile2("bclaim2.dat");

for(int i=0;i<14;i++)

{

ifile2.read((char\*)&b,sizeof(b));

ifile1.read((char\*)&g,sizeof(g));

if(!(strcmp(g.status,"UNDER MAINTENANCE"))&&(x<=40))

strcpy(g.status,"AVAILABLE");

if(!(strcmp(b.status,"UNDER MAINTENANCE"))&&(x>=60))

strcpy(b.status,"AVAILABLE");

ofile1.write((char\*)&g,sizeof(g));

ofile2.write((char\*)&b,sizeof(b));

}

ofile2.close();

ifile2.close();

ofile1.close();

ifile1.close();

remove("gates.dat");

rename("gates2.dat","gates.dat");

remove("bclaim.dat");

rename("bclaim2.dat","bclaim.dat");

}

void initcounter()

{

counter=0;

}

public:

void get\_flight\_no(char c[])

{

strcpy(c, flight\_no);

}

void get\_status(char c[])

{

strcpy(c, status);

}

void get\_destination(char c[])

{

strcpy(c, destination);

}

};

airplane airp;

void start()

{

bool click = false;

int x\_co, y\_co;

clearmouseclick(WM\_LBUTTONDOWN);

while(true)

{

airp.obj\_tofile();

int x=rand()%100;

if(x<=10)

airp.statusmod();

airp.freer();

getmouseclick(WM\_LBUTTONDOWN, x\_co, y\_co);

if(x\_co>=95 && x\_co<=572 && y\_co>=77 && y\_co<=125)

View\_Notifications();

else if(x\_co>=95 && x\_co<=572 && y\_co>=177 && y\_co<=225)

airplane\_status();

else if(x\_co>=95 && x\_co<=520 && y\_co>=277 && y\_co<=325)

gate\_status();

else if(x\_co>=95 && x\_co<=755 && y\_co>=377 && y\_co<=425)

baggage\_claim\_status();

else if(x\_co>=95 && x\_co<=260 && y\_co>=477 && y\_co<=525)

search();

else if(x\_co>=95 && x\_co<=208 && y\_co>=577 && y\_co<=625)

exit(0);

}

}

void disp\_menu()

{

settextstyle(8,0,2);

char str1[5];

str1[0]='(';

str1[3]=')';

str1[2]=counter%10+48;

str1[1]=counter/10+48;

str1[4]='\0';

cleardevice();

outtextxy(575,70,str1);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 77, 572, 125);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 177, 572, 225);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 277, 520, 325);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 377, 755, 425);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 477, 260, 525);

setfillstyle(SOLID\_FILL, rand()%15+1);

bar(95, 577, 208, 625);

settextstyle(8, 0, 5);

outtextxy(100, 80, "View Notifications");

outtextxy(100, 180, "View Flight Status");

outtextxy(100, 280, "View Gate Status");

outtextxy(100, 380, "View Baggage Claim Status");

outtextxy(100, 480, "Search");

outtextxy(100, 580, "Exit");

settextstyle(8, 0, 2);

start();

}

void View\_Notifications()

{

cleardevice();

airp.initcounter();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to main menu.");

settextstyle(8, 0, 2);

setcolor(WHITE);

char c[2], \*str;

int i = 0;

str[0] = '\0';

c[1] = '\0';

ifstream ifile("notifications.txt");

while(ifile)

{

ifile.getline(str, 100, '\n');

if(ifile.eof())

break;

outtextxy(100, 100+i, str);

i+=30;

if(i>=500)

break;

}

ifile.close();

ofstream ofile("notifications.txt", ios::out);

ofile.close();

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

disp\_menu();

else

choice = '\0';

}

}

void gate\_status()

{

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to main menu.");

settextstyle(8, 0, 2);

setcolor(WHITE);

int i=0;

gate obj;

char str[7],str2[7];

ifstream ifile("gates.dat");

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(350, 40, "Gate Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Gate No.");

outtextxy(450, 100, "Flight Number");

outtextxy(700, 100, "Status");

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(440, 80, 440, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

while(ifile)

{

char x[20], y[7], z[30];

ifile.read((char\*)&obj, sizeof(obj));

if(ifile.eof())

break;

obj.get\_gatename(x);

obj.get\_airplaneno(y);

obj.get\_status(z);

obj.get\_airplaneno(str2);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i,2,170+i);

outtextxy(200, 150+i, x);

outtextxy(450, 150+i, y);

if(!(strcmp(z, "AVAILABLE")))

setcolor(GREEN);

else if(!(strcmp(z, "IN USE")))

setcolor(RED);

else

setcolor(YELLOW);

outtextxy(700, 150+i, z);

setcolor(WHITE);

i+=35;

}

ifile.close();

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

disp\_menu();

else

choice = '\0';

}

}

void baggage\_claim\_status()

{

char str2[7];

char str[7];

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to main menu.");

settextstyle(8, 0, 2);

setcolor(WHITE);

int i=0;

baggage\_claim obj;

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(300, 40, "Baggage Claim Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(440, 80, 440, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

ifstream ifile("bclaim.dat");

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Baggage claim No.");

outtextxy(450, 100, "Flight Number");

outtextxy(700, 100, "Status");

while(ifile)

{

char x[20], y[7], z[30];

ifile.read((char\*)&obj, sizeof(obj));

if(ifile.eof())

break;

obj.get\_name(x);

obj.get\_airplaneno(y);

obj.get\_status(z);

obj.get\_airplaneno(str2);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i,2,170+i);

outtextxy(200, 150+i, x);

outtextxy(450, 150+i, y);

if(!(strcmp(z, "AVAILABLE")))

setcolor(GREEN);

else if(!(strcmp(z, "IN USE")))

setcolor(RED);

else

setcolor(YELLOW);

outtextxy(700, 150+i, z);

setcolor(WHITE);

i+=35;

}

ifile.close();

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

disp\_menu();

else

choice = '\0';

}

}

void airplane\_status()

{

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to main menu.");

settextstyle(8, 0, 2);

setcolor(WHITE);

int i=0;

airplane obj;

setcolor(WHITE);

char str[7],str2[7];

ifstream ifile("flights.dat");

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(350, 40, "Flight Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(390, 80, 390, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Flight Number");

outtextxy(400, 100, "Status");

outtextxy(700, 100, "Destination");

while(ifile)

{

char w[7], x[25], z[50];

ifile.read((char\*)&obj, sizeof(obj));

if(ifile.eof())

break;

obj.get\_flight\_no(w);

obj.get\_status(x);

obj.get\_destination(z);

obj.get\_flight\_no(str2);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i, 2, 170+i);

outtextxy(200, 150+i, w);

outtextxy(400, 150+i, x);

outtextxy(700, 150+i, z);

i+=35;

}

ifile.close();

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

disp\_menu();

else

choice = '\0';

}

}

void search()

{

char ch;

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to main menu.");

settextstyle(8, 0, 2);

setcolor(WHITE);

settextstyle(8, 0, 5);

outtextxy(100, 100, "1. Search by aircompany");

outtextxy(100, 250, "2. Search by destination");

outtextxy(100, 400, "3. Search by status");

settextstyle(8, 0, 2);

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

disp\_menu();

else if(choice == '1')

Search\_by\_aircompany();

else if(choice == '2')

Search\_by\_destination();

else if(choice == '3')

Search\_by\_status();

else

choice = '\0';

}

}

void Search\_by\_aircompany()

{

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to Search.");

settextstyle(8, 0, 2);

setcolor(WHITE);

outtextxy(100, 100, "a. American Airlines");

outtextxy(100, 150, "b. Air India");

outtextxy(100, 200, "c. British Airways");

outtextxy(100, 250, "d. Air Canada");

outtextxy(100, 300, "e. Cathay Pacific");

outtextxy(100, 350, "f. Emirates");

outtextxy(100, 400, "g. Air France");

outtextxy(100, 450, "h. Lufthansa");

outtextxy(100, 500, "i. Singapore Airlines");

outtextxy(100, 550, "j. Thai");

char str[3];

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

search();

else if(choice == 'a')

strcpy(str, "AA");

else if(choice == 'b')

strcpy(str, "AI");

else if(choice == 'c')

strcpy(str, "BA");

else if(choice == 'd')

strcpy(str, "CA");

else if(choice == 'e')

strcpy(str, "CP");

else if(choice == 'f')

strcpy(str, "EM");

else if(choice == 'g')

strcpy(str, "FR");

else if(choice == 'h')

strcpy(str, "LU");

else if(choice == 'i')

strcpy(str, "SA");

else if(choice == 'j')

strcpy(str, "TA");

else

{

choice = '\0';

continue;

}

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(350, 680, "Press 'Backspace' to go back to Search by aircompany.");

settextstyle(8, 0, 2);

setcolor(WHITE);

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(350, 40, "Flight Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(390, 80, 390, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Flight Number");

outtextxy(400, 100, "Status");

outtextxy(700, 100, "Destination");

airplane a;

ifstream ifile("flights.dat");

int i=0;

while(ifile)

{

ifile.read((char\*)&a,sizeof(a));

if(ifile.eof())

break;

if((a.flight\_no[0]==str[0])&&(a.flight\_no[1]==str[1]))

{

setcolor(WHITE);

char str[7],str2[7];

strcpy(str2, a.flight\_no);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i, 2, 170+i);

outtextxy(200, 150+i, a.flight\_no);

outtextxy(400, 150+i, a.status);

outtextxy(700, 150+i, a.destination);

i+=35;

}

}

ifile.close();

char s = '\0';

while(s == '\0')

{

s = getch();

if(s == '\b')

Search\_by\_aircompany();

else

s = '\0';

}

}

}

void Search\_by\_destination()

{

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to Search.");

settextstyle(8, 0, 2);

setcolor(WHITE);

outtextxy(100, 100, "a. New Delhi");

outtextxy(100, 150, "b. Mumbai");

outtextxy(100, 200, "c. Chennai");

outtextxy(100, 250, "d. Kolkata");

outtextxy(100, 300, "e. London");

outtextxy(100, 350, "f. Singapore");

outtextxy(100, 400, "g. New York");

outtextxy(100, 450, "h. Barcelona");

outtextxy(100, 500, "i. Paris");

outtextxy(100, 550, "j. Rio De Janeiro");

char str[50];

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

search();

else if(choice == 'a')

strcpy(str, "New Delhi");

else if(choice == 'b')

strcpy(str, "Mumbai");

else if(choice == 'c')

strcpy(str, "Chennai");

else if(choice == 'd')

strcpy(str, "Kolkata");

else if(choice == 'e')

strcpy(str, "London");

else if(choice == 'f')

strcpy(str, "Singapore");

else if(choice == 'g')

strcpy(str, "New York");

else if(choice == 'h')

strcpy(str, "Barcelona");

else if(choice == 'i')

strcpy(str, "Paris");

else if(choice == 'j')

strcpy(str, "Rio De Janeiro");

else

{

choice = '\0';

continue;

}

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(350, 680, "Press 'Backspace' to go back to Search by Destination.");

settextstyle(8, 0, 2);

setcolor(WHITE);

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(350, 40, "Flight Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(390, 80, 390, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Flight Number");

outtextxy(400, 100, "Status");

outtextxy(700, 100, "Destination");

airplane a;

char temp[50];

ifstream ifile("flights.dat");

int i=0;

while(ifile)

{

ifile.read((char\*)&a,sizeof(a));

if(ifile.eof())

break;

strcpy(temp,a.destination);

if(strcmp(temp,str)==0)

{

setcolor(WHITE);

char str[7],str2[7];

strcpy(str2, a.flight\_no);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i, 2, 170+i);

outtextxy(200, 150+i, a.flight\_no);

outtextxy(400, 150+i, a.status);

outtextxy(700, 150+i, a.destination);

i+=35;

}

}

ifile.close();

char s = '\0';

while(s == '\0')

{

s = getch();

if(s == '\b')

Search\_by\_destination();

else

s = '\0';

}

}

}

void Search\_by\_status()

{

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(525, 680, "Press 'Backspace' to go back to Search.");

settextstyle(8, 0, 2);

setcolor(WHITE);

outtextxy(100, 100, "a. ARRIVING");

outtextxy(100, 150, "b. TAXIING FROM ARRIVAL");

outtextxy(100, 200, "c. DEBOARDING");

outtextxy(100, 250, "d. REFUELLING");

outtextxy(100, 300, "e. TAXIING FOR DEPARTURE");

outtextxy(100, 350, "f. HOLD SHORT RUNWAY");

outtextxy(100, 400, "g. DEPARTED");

char str[25];

char choice = '\0';

while(choice == '\0')

{

choice = getch();

if(choice == '\b')

search();

else if(choice == 'a')

strcpy(str, "ARRIVING");

else if(choice == 'b')

strcpy(str, "TAXIING FROM ARRIVAL");

else if(choice == 'c')

strcpy(str, "DEBOARDING");

else if(choice == 'd')

strcpy(str, "REFUELLING");

else if(choice == 'e')

strcpy(str, "TAXIING FOR DEPARTURE");

else if(choice == 'f')

strcpy(str, "HOLD SHORT RUNWAY");

else if(choice == 'g')

strcpy(str, "DEPARTED");

else

{

choice = '\0';

continue;

}

cleardevice();

settextstyle(8, 0, 1);

setcolor(LIGHTBLUE);

outtextxy(350, 680, "Press 'Backspace' to go back to Search by Status.");

settextstyle(8, 0, 2);

setcolor(WHITE);

settextstyle(8, 0, 4);

setcolor(YELLOW);

outtextxy(350, 40, "Flight Status");

settextstyle(8, 0, 2);

setcolor(WHITE);

setcolor(YELLOW);

line(80, 80, 950, 80);

line(80, 135, 950, 135);

line(80, 650, 950, 650);

line(80, 80, 80, 650);

line(190, 80, 190, 650);

line(390, 80, 390, 650);

line(690, 80, 690, 650);

line(950, 80, 950, 650);

setcolor(WHITE);

outtextxy(90, 100, "Logo");

outtextxy(200, 100, "Flight Number");

outtextxy(400, 100, "Status");

outtextxy(700, 100, "Destination");

airplane a;

ifstream ifile("flights.dat");

int i=0;

while(ifile)

{

ifile.read((char\*)&a,sizeof(a));

if(ifile.eof())

break;

if(strcmp(a.status,str)==0)

{

setcolor(WHITE);

char str[7],str2[7];

strcpy(str2, a.flight\_no);

str[0]=str2[0];

str[1]=str2[1];

str[2]='.';

str[3]='b';

str[4]='m';

str[5]='p';

str[6]='\0';

readimagefile(str, 90, 150+i, 2, 170+i);

outtextxy(200, 150+i, a.flight\_no);

outtextxy(400, 150+i, a.status);

outtextxy(700, 150+i, a.destination);

i+=35;

}

}

ifile.close();

char s = '\0';

while(s == '\0')

{

s = getch();

if(s == '\b')

Search\_by\_status();

else

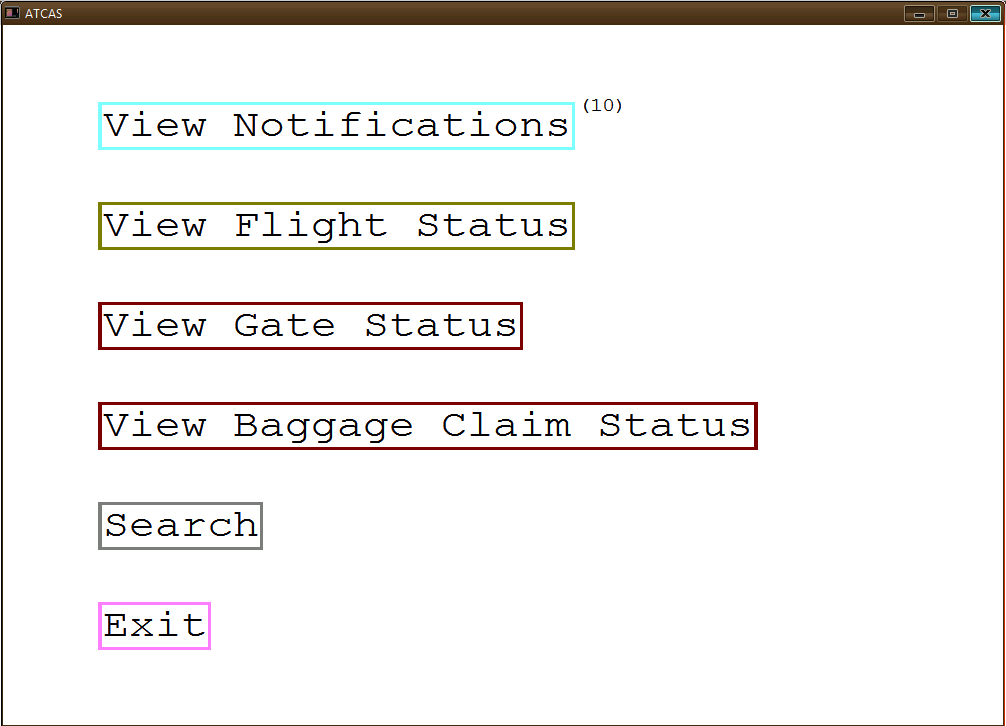
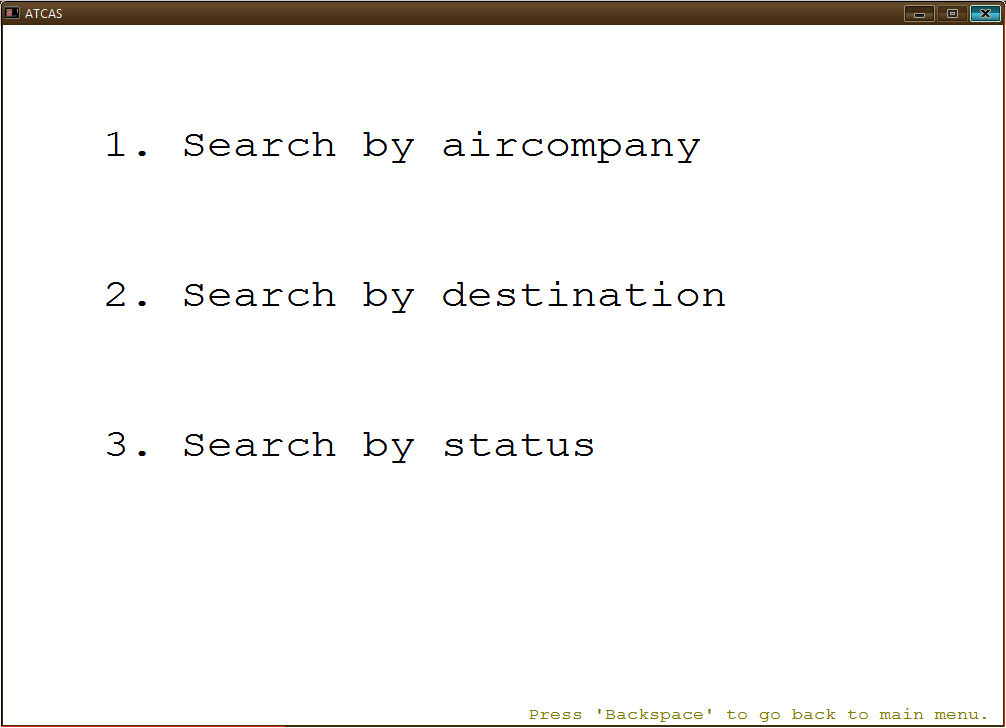
s = '\0';

}

}

}

# Output Screenshots:



# 

# 

# Scope for improvement

1. Airplane allotment could be done manually.
2. Quality of display of the airport screens and menus could be improved.
3. More real-life scenarios could be added.
4. More details could be added.
5. More facilities could be added.

# References:

[www.wikipedia.org](http://www.wikipedia.org)

[www.cplusplus.com](http://www.cplusplus.com)

[www.stackoverflow.com](http://www.stackoverflow.com)