

NSF-TUES Award # 1245036
Collaborative Education: Building a Skilled V&V Community
For use in Code Inspection Exercise
Used with Permission of author: **Kaitlyn Sprole**

```
//Airlines Reservation System

#include <iostream>
#include <string>
#include <fstream>
using namespace std;
class airline
{
public:
void start_system();
int menu();
void mod_flight();
void mod_reserv();
void mod_empl();
void report();
void exit_system();
private:
struct employee_node
{
string e_fname, e_lname;
int e_type, e_id;
employee_node *next_employee;
};
struct reservation_node
{
string r_fname, r_lname, phone;
int r_type;
reservation_node *next_reservation;
};

struct flight_node
{
string o_city, d_city, date, time, number;
flight_node *prev, *next;
employee_node *add_employee;
reservation_node *add_reservation;
};

flight_node *first, *last, *current, *save;
reservation_node *r_first, *r_last, *r_current, *r_save, *r_prev;
employee_node *e_first, *e_last, *e_current, *e_save, *e_prev;
};
int main()
{
airline air;//struct air of type airline

int selection;

air.start_system(); //start system
selection = air.menu();//main menu
while (selection!=5)
```

```

{
if (selection==1)
{
air.mod_flight();//modify flight
}
else if (selection==2)
{
air.mod_reserv();//modify reservation
}
else if (selection==3)
{
air.mod_empl();//modify employee
}
else if (selection==4)
{
air.report();//report
}
selection = air.menu();
//end while loop
}
air.exit_system();
return 0;
}

/*****
FUNCTION: int airline::menu()

Create menu for user. This menu will prompt the user
to enter an integer value between 1 and 4 to select which
menu he wants to enter. If the user selects option 5 the
program will end
*****/

int airline::menu()
{

int selection;
cout<<"AIRLINE MENU"<<endl<<endl;
cout<<"1. Add/Modify Flight Information"<<endl;
cout<<"2. Add/Modify Reservation Information"<<endl;
cout<<"3. Add/Modify Employee Information"<<endl;
cout<<"4. Report Section"<<endl;
cout<<"5. Exit System"<<endl<<endl;
cout<<"Please enter selection> ";
cin>>selection;
cout<<endl;
return selection;
}

/*****
FUNCTION: int airline::start_system()

```

This function reads in all flight, employee, and reservation information in from the data file "flightfile.dat". It also creates a data file for each flight for employees called E####.dat, and reservations R###.dat where #### is the number of the flight

*****/

```
void airline::start_system()
{
    ifstream infile,infile2;
    int i;
    char filename[10];

    first=new flight_node;
    first->prev=0;
    first->next=0;
    last=first;

    infile.open("flightfile.dat");

    while(!infile.eof() &&!infile.fail()){

        current=new flight_node;
        current->next=0;
        current->prev=last;
        last->next=current;
        last=current;

        infile>>current->number>>current->o_city>>
            current->d_city>>current->date>>current->time;

        filename[0]='E';
        for(i=1;i<=4;i++) filename[i]=current->number[i-1];
        filename[5]='.';
        filename[6]='d';
        filename[7]='a';
        filename[8]='t';
        filename[9]='\0';
        infile2.open(filename);

        while(!infile2.eof() &&!infile2.fail()){

            e_current = new employee_node;

            infile2>>e_current->e_id>>e_current->e_fname>>
                e_current->e_lname>>e_current->e_type;
                //if this is the first employee
                if (current->add_employee == 0)
            {
                current->add_employee = e_current;
                e_current-> next_employee = 0;

            }//end if
            else
            {
                e_save = current->add_employee;
                e_last = current->add_employee;
                while (e_last != 0)
                {
                    e_save = e_last;
                    e_last = e_last->next_employee;
                }
            }
        }
    }
}
```

```

        }//end while
        e_save->next_employee = e_current;
        e_current->next_employee = 0;
    }//end else
}//end employee while

infile2.close();

filename[0]='R';
infile2.open(filename);

while(!infile2.eof() && !infile2.fail()){

    r_current = new reservation_node;

    infile2>>r_current->phone>>r_current->
r_fname>>r_current->r_lname>>r_current->r_type;
    //if this is the first reservation
    if (current->add_reservation == 0)
    {
        current->add_reservation = r_current;
        r_current-> next_reservation = 0;

    }//end if
    //if this is not the first reservation
    else
    {
        r_save = current->add_reservation;
        r_last = current->add_reservation;
        while (r_last != 0)
        {
            r_save = r_last;
            r_last = r_last->next_reservation;
        }//end while
        r_save->next_reservation = r_current;
        r_current->next_reservation = 0;
    }//end else

}//end reservation while

infile2.close();

}//end flight while

infile.close();

}//end start_system

/*****
FUNCTION: int airline::mod_flight()

```

This function reads in an integer value between 1 to 4 to enable the user to add a flight, delete a flight, modify a flight or return to the main menu. This function also carries out the desired action of the user.

```

*****/

```

```

void airline::mod_flight()
{
int selection1;

cout<<"FLIGHT INFORMATION MENU"<<endl<<endl;
cout<<"1. Add Flight"<<endl;
cout<<"2. Delete Flight"<<endl;
cout<<"3. Modify Flight"<<endl;
cout<<"4. Return"<<endl<<endl;
cout<<"Please enter selection> ";
cin>>selection1;
cout<<endl;
while (selection1 !=4)
{
if(selection1 == 1)
{
//add flight
current= new flight_node;
cout<<"Flight Number (4 digits): ";
cin>>current->number;
cout<<"Orignation City: ";
cin>>current->o_city;
cout<<"Destination City: ";
cin>>current->d_city;
cout<<"Date: ";
cin>>current->date;
cout<<"Time: ";
cin>>current->time;
cout<<endl<<endl;
current->add_reservation = 0;
current->add_employee = 0;
current -> next = 0;
current ->prev = last;
last->next = current;
last = current;

}
else if(selection1 == 2)
{
//delete flight
char del='n';
//-----
current = first->next;
cout<<"Flights: "<<endl;
while (current != 0)
{
    cout<<current->number<<endl;
    current = current->next;
};
cout<<endl;
//-----
current = first->next;
string deleteflight;
cout<<"Enter flight number to delete flight: ";
cin>>deleteflight;
while (current != 0)

```

```

{
    if (current ->number == deleteflight)
        save = current;
        current = current-> next;
}
current = save;

        cout<<"Are you sure you want to delete this ";
        cout<<"flight?(y or n): ";
        cin>>del;
        cout<<endl;
        if (del == 'y')
        {
            current->prev->next = current->next;
            if (current->next ==0)
                last = current->prev;
            else
                current->next->prev = current->prev;
            delete current;
        }

}

else if(selection1 == 3)
{
//modify flight
int mod_selection;
string modifyflight;
//-----
current = first->next;
cout<<"Flights: "<<endl;
while (current != 0)
{
    cout<<current->number<<endl;
    current = current->next;
};
cout<<endl;
//-----
cout<<"Enter flight number to modify flight: ";
cin>> modifyflight;
cout<<endl;
current = first ->next;
while (current != 0)
{
    if (current-> number == modifyflight)
        save = current;
        current = current-> next;
}
}
//end while
    current = save;
    cout<<"1. Modify Origination City"<<endl;
    cout<<"2. Modify Destination City"<<endl;
    cout<<"3. Modify Date"<<endl;
    cout<<"4. Modify Time"<<endl;
    cout<<"Please enter selection> ";
    cin>>mod_selection;
    cout<<endl;

```

```

        if (mod_selection == 1)
        {
            string new_o_city;
            cout<<"Enter new origination city :";
            cin>>new_o_city;
            current->o_city = new_o_city;
        }
        else if (mod_selection == 2)
        {
            string new_d_city;
            cout<<"Enter new destination city :";
            cin>>new_d_city;
            current->d_city = new_d_city;
        }
        else if (mod_selection == 3)
        {
            string new_date;
            cout<<"Enter new date :";
            cin>>new_date;
            current->date = new_date;
        }
        else if (mod_selection == 4)
        {
            string new_time;
            cout<<"Enter new time :";
            cin>>new_time;
            current->time = new_time;
        }
        cout<<endl;
    }

    cout<<"FLIGHT INFORMATION MENU"<<endl<<endl;
    cout<<"1. Add Flight"<<endl;
    cout<<"2. Delete Flight"<<endl;
    cout<<"3. Modify Flight"<<endl;
    cout<<"4. Return"<<endl<<endl;
    cout<<"Please enter selection> ";
    cin>>selection1;
    cout<<endl;
}
}

/*****
FUNCTION: int airline::mod_reserv()

This function reads in an integer value between 1 to 4 to
enable the user to add a reservation, delete a reservation,
modify a reservation or return to the main menu. This function
also carries out the desired action of the user.
*****/

void airline::mod_reserv()
{
    int selection2;
    cout<<"RESERVATION INFORMATION MENU"<<endl<<endl;
    cout<<"1. Add Reservation"<<endl;

```



```

        }//end while
        r_save->next_reservation = r_current;
        r_current->next_reservation = 0;
    }//end else
}//end if
current=current->next;
}//end while
}//end add employee if
else if(selection2 == 2)
{
string delresf;

current = first->next;
cout<<"Flights: "<<endl<<endl;
while (current != 0)
{
    cout<<current->number<<endl;
    current = current->next;
};
cout<<endl;
    //search for flight
    cout<<"Enter flight number to delete reservation from: ";
    cin>>delresf;
    current = first->next;
    while (current != 0 && current->number != delresf)
    {
        current = current->next;
    }//end while
    r_current = current->add_reservation;
    r_prev = 0;
    string delres;
    cout<<"Enter last name to delete reservation: ";
    cin>>delres;
    cout<<endl;
    //if first reservation;
    if (r_current == current->add_reservation)
    {
        current->add_reservation = r_current->next_reservation;
    }//end if first employee
else{
//search for employee
    while (r_current != 0 && r_current->r_lname != delres)
    {
        r_prev = r_current;
        r_current = r_current->next_reservation;
    }//end while
    //if last reservation
    if (r_current->next_reservation == 0)
    {
        r_prev->next_reservation = 0;
        delete r_current;
        r_current = current->add_reservation;
    }//end else if last employee
    //if reservation in the middle
else
    {
        r_prev->next_reservation = r_current->next_reservation;

```

```

        r_current = current->add_reservation;
    } //end else if in middle
} //end else
} //end delete
else if(selection2 == 3)
{
string modresf;
current = first->next;
cout<<"Flights: "<<endl<<endl;
while (current != 0)
{
    cout<<current->number<<endl;
    current = current->next;
};
cout<<endl;

//search for flight
cout<<"Enter flight number to modify reservation type: ";
cin>>modresf;
cout<<endl;
current = first->next;
while (current != 0 && current->number != modresf)
{
    current = current->next;
} //end while
cout<<current->number<<endl;
r_current = current->add_reservation;
string modres;
cout<<"Enter last name to modify reservation type: ";
cin>>modres;
cout<<endl;
while (r_current->r_lname != modres) //search for reservation
{
    r_current = r_current -> next_reservation;
}
if (r_current->r_lname == modres)
{
    int new_r_type;
    cout<<"Enter new employee type (1 = Coach,";
    cout<<" 2 = Business, 3 = First Class): ";
    cin>>new_r_type;
    cout<<endl;
    r_current->r_lname = new_r_type;
}
r_current = current->add_reservation;

}
cout<<"RESERVATION INFORMATION MENU"<<endl<<endl;
cout<<"1. Add Reservation"<<endl;
cout<<"2. Delete Reservation"<<endl;
cout<<"3. Modify Type of Seat for Reservation"<<endl;
cout<<"4. Return"<<endl<<endl;
cout<<"Please enter selection> ";
cin>>selection2;
cout<<endl;

```

```

}
}

/*****
FUNCTION: int airline::mod_empl()

This function reads in an integer value between 1 to 4 to
enable the user to add an employee, delete an employee,
modify an employee's information or return to the main menu. This
function also carries out the desired action of the user.
*****/

void airline::mod_empl()
{
int selection3;
cout<<"RESERVATION EMPLOYEE MENU"<<endl<<endl;
cout<<"1. Add Employee"<<endl;
cout<<"2. Delete Employee"<<endl;
cout<<"3. Modify Type of Employee"<<endl;
cout<<"4. Return"<<endl<<endl;
cout<<"Please enter selection> ";
cin>>selection3;
cout<<endl;
while (selection3 !=4)
{
if(selection3 == 1)
{
current = first->next;
string addemployee;
//-----
current = first->next;
cout<<"Flights: "<<endl<<endl;
while (current != 0)
{
cout<<current->number<<endl;
current = current->next;
};
cout<<endl;
//-----
current = first->next;4
cout<<"Enter flight number to add employee to: ";
cin>>addemployee;
cout<<endl;
while (current!=0)
{
if (addemployee == current->number)
{
e_current = new employee_node;
cout<<"First Name: ";
cin>>e_current->e_fname;
cout<<"Last Name: ";
cin>>e_current->e_lname;
cout<<"Type of Employee (1 pilot, 2 Co-Pilot, 3 Flight Attendant): ";
cin>>e_current->e_type;
cout<<"ID Number: ";
cin>>e_current->e_id;

```

```

cout<<endl;
//if this is the first reservation
if (current->add_employee == 0)
{
    current->add_employee = e_current;
    e_current-> next_employee = 0;

}

//end if
else
{
    e_save = current->add_employee;
    e_last = current->add_employee;
while (e_last != 0)
{
    e_save = e_last;
    e_last = e_last->next_employee;
}

//end while
e_save->next_employee = e_current;
e_current->next_employee = 0;
}

//end else
}

//end if
current=current->next;
}

//end while
}

//end add employee if
else if(selection3 == 2)
{
    string delempf;

current = first->next;
cout<<"Flights: "<<endl<<endl;
while (current != 0)
{
    cout<<current->number<<endl;
    current = current->next;
};
cout<<endl;
//search for flight
cout<<"Enter flight number to delete employee from: ";
cin>>delempf;
cout<<endl;
current = first->next;
while (current != 0 && current->number != delempf)
{
    current = current->next;
}

//end while
cout<<current->number<<endl;
e_current = current->add_employee;
e_prev = 0;
int delemp;
cout<<"Enter employee ID to delete employee: ";
cin>>delemp;
cout<<endl;
//if first employee;
if (e_current == current->add_employee
    && e_current->e_id == delemp)
{
    current->add_employee = e_current->next_employee;
}
}
}

```

```

        }//end if first employee

else
{
//search for employee
while (e_current != 0 && e_current->e_id != delempt)
{
    e_prev = e_current;
    e_current = e_current->next_employee;
}//end while
cout<<e_current->e_id<<endl;
cout<<e_prev->e_id<<endl;
//if last employee
if (e_current->next_employee == 0)
{
    e_prev->next_employee = 0;
    delete e_current;
    e_current = current->add_employee;
}//end else if last employee
else
{
    e_prev->next_employee = e_current->next_employee;
    e_current = current->add_employee;
}//end else if in middle
}//end else
}
else if(selection3 == 3)
{
    string modemp;
    current = first->next;
    cout<<"Flights: "<<endl<<endl;
    while (current != 0)
    {
        cout<<current->number<<endl;
        current = current->next;
    };
    cout<<endl;
    //search for flight
    cout<<"Enter flight number to modify employee type: ";
    cin>>modemp;
    cout<<endl;
    current = first->next;
    while (current != 0 && current->number != modemp)
    {
        current = current->next;
    }//end while
    cout<<current->number<<endl;
    e_current = current->add_employee;
    int modemp;
    cout<<"Enter employee ID to modify employee type: ";
    cin>>modemp;
    cout<<endl;
    //search for employee
    while (e_current->e_id != modemp)
    {
        e_current = e_current -> next_employee;
    }
}

```

```

        if (e_current->e_id ==modemp)
        {
            int new_e_type;
            cout<<"Enter new employee type (1 = Pilot, 2 = ";
            cout<<"Co-Pilot, 3 = Flight Attendant): ";
            cin>>new_e_type;
            cout<<endl;
            e_current->e_type = new_e_type;
        }
        e_current = current->add_employee;
    }
    cout<<"EMPLOYEE INFORMATION MENU"<<endl<<endl;
    cout<<"1. Add Employee"<<endl;
    cout<<"2. Delete Employee"<<endl;
    cout<<"3. Modify Type of Employee"<<endl;
    cout<<"4. Return"<<endl<<endl;
    cout<<"Please enter selection> ";
    cin>>selection3;
    cout<<endl;

}
}

/*****
FUNCTION: int airline::report()

This function reads in an integer value between 1 to 7 to
determine what information to print. If the user selects
option 8 he will return to the main menu.
*****/

void airline::report()
{
    int
    selection4;
    cout<<"FLIGHT REPORT"<<endl<<endl;
    cout<<"1. Flight Master Information"<<endl;
    cout<<"2. Print All Reservations for Desired Flight"<<endl;
    cout<<"3. Print All Employees on Desired Flight"<<endl;
    cout<<"4. Print All Pilots and What Flights They Work On"<<endl;
    cout<<"5. Print All Flights A Specific Employee Works On"<<endl;
    cout<<"6. Print All Flights Going To Or Coming From A Specific City"<<endl;
    cout<<"7. Print All Flights On A Specific Date"<<endl;
    cout<<"8. Return"<<endl<<endl;
    cout<<"Please enter selection> ";
    cin>> selection4;
    cout<<endl;
    while (selection4 != 8)
    {
        if(selection4 == 1)
        {
            current = first-> next;
            while(current != last)
            {
                cout<<"Flight Number:"<<current->number<<endl;
                cout<<"Origination City:"<<current->o_city<<endl;
            }
        }
    }
}

```

```

cout<<"Destination City:"<<current->d_city<<endl;
cout<<"Date:"<<current->date<<endl;
cout<<"Time:"<<current->time<<endl<<endl;
current = current->next;
}
//end while
if (current == last)
{
cout<<"Flight Number: "<<current->number<<endl;
cout<<"Origination City: "<<current->o_city<<endl;
cout<<"Destination City: "<<current->d_city<<endl;
cout<<"Date: "<<current->date<<endl;
cout<<"Time: "<<current->time<<endl<<endl;
}
//end if
} //end report1

else if (selection4 == 2)
{
string printreserv;
cout<<"Enter flight number to print reservation list: "<<endl;
cin>>printreserv;
current = first->next;
while (current != 0)
{
    if(current->number == printreserv)
    {
        r_current = current->add_reservation;
        while (r_current != 0)
        {
            cout<<"RESERVATIONS:"<<endl<<endl;
            cout<<"First Name: "<<r_current->r_fname<<endl;
            cout<<"Last Name: "<<r_current->r_lname<<endl;
            cout<<"Type: "<<r_current->r_type<<endl;
            cout<<"Phone Number: "<<r_current->phone<<endl<<endl;
            r_current= r_current->next_reservation;
        }
    } //end while
} //end if
    current = current->next;

} //end while
} //end else if

else if(selection4 == 3)
{
string printempl;
cout<<"Enter flight number to employee list: ";
cin>>printempl;
cout<<endl;
current = first->next;
while (current != 0)
{
    if(current->number == printempl)
    {
        e_current = current->add_employee;
        while (e_current != 0)

```

```

        {
            cout<<"EMPLOYEES:"<<endl<<endl;
            cout<<"First Name: "<<e_current->e_fname<<endl;
            cout<<"Last Name: "<<e_current->e_lname<<endl;
            cout<<"Type: "<<e_current->e_type<<endl;
            cout<<"ID number: "<<e_current->e_id<<endl<<endl;
            e_current= e_current->next_employee;

            }//end while
        }//end if
        current = current->next;

}

}

else if (selection4 == 4)
{
    current = first->next;

    while (current != 0)
    {
        e_current = current-> add_employee;
        while (e_current != 0)
        {
            if (e_current->e_type == 1)
            {
                cout<<"Pilot Name: "<<e_current->e_fname<<" "
                    <<e_current->e_lname<<endl;
                cout<<"Flight Number: "<<current->number<<endl<<endl;
            }
            e_current = e_current-> next_employee;
        }
        current = current->next;
    }
}

else if(selection4 == 5)
{
    int searchid;
    cout<<"Enter employee id to look up: ";
    cin>>searchid;
    cout<<endl;
    cout<<"Flights Employee "<<searchid<<" Works On: "<<endl;
    current = first->next;
    while (current != 0)
    {
        e_current = current-> add_employee;
        while (e_current != 0)
        {
            if (e_current->e_id == searchid)
            {
                cout<<current->number<<endl;
            }
            e_current = e_current-> next_employee;
        }
        current = current->next;
    }
}

cout<<endl;

```



```

}
else if (selection4 == 6)
{
    string searchcity;
    cout<<"Enter city to view flights to and from city: ";
    cin>>searchcity;
    cout<<endl<<endl;
    cout<<"Flights To City: "<<endl<<endl;
    current = first-> next;
    while (current != 0)
    {
        if (current->d_city == searchcity)
            cout<<current->number<<endl;
            current = current-> next;
    }
    cout<<endl<<endl;
    cout<<"Flights From City: "<<endl<<endl;
    current = first-> next;
    while (current != 0)
    {
        if (current->o_city == searchcity)
            cout<<current->number<<endl;
            current = current-> next;
    }
    cout<<endl<<endl;
}
else if (selection4 == 7)
{
    string searchdate;
    cout<<"Enter date to view flights: ";
    cin>>searchdate;
    cout<<endl;
    current = first-> next;
    while (current != 0)
    {
        if (current->date == searchdate)
            cout<<current->number<<endl;
            current = current->next;
    }
}
else if (selection4 == 8)
{
}
cout<<"FLIGHT REPORT"<<endl<<endl;
cout<<"1. Flight Master Information"<<endl;
cout<<"2. Print all reservations for desired flight"<<endl;
cout<<"3. Print all employees on desired flight"<<endl;
cout<<"4. Print All Pilots and What Flights They Work On"<<endl;
cout<<"5. Print All Flights A Specific Employee Works On"<<endl;
cout<<"6. Print All Flights Going To Or Coming From A Specific City"<<endl;
cout<<"7. Print All Flights On A Specific Date"<<endl;
cout<<"8. Return"<<endl<<endl;
cout<<"Please enter selection> ";
cin>>selection4;
cout<<endl;
} //end while

```

```
}//end report
```

```
/*  
*****  
FUNCTION: int airline::start_system()
```

```
This function reads in all flight, employee, and reservation  
information in from the data file "flightfile.dat". It  
also creates a data file for each flight for employees  
called E####.dat, and reservations R###.dat where #### is  
the number of the flight
```

```
*****/  
*/
```

```
void airline::exit_system()
```

```
{
```

```
    ofstream outfile,outfile2;
```

```
    int i;
```

```
    char filename[10];
```

```
    if(first!=last){
```

```
        //WRITE FLIGHT INFORMATION TO DATA FILE
```

```
        current=first->next;
```

```
        outfile.open("flightfile.dat");
```

```
        while(current!=0){
```

```
            if(current->next!=0)
```

```
            {
```

```
                outfile<<current->number<<" "<<current->o_city<<" "
```

```
                <<current->d_city<<" "<<current->date<<" "<<current->
```

```
>time<<endl;
```

```
            }
```

```
            else
```

```
            {
```

```
                outfile<<current->number<<" "<<current->o_city<<" "
```

```
                <<current->d_city<<" "<<current->date<<" "<<current->
```

```
>time;
```

```
            }
```

```
        //WRITE EMPLOYEE INFORMATION TO ITS DATA FILE (E####.dat)
```

```
        e_current=current->add_employee;
```

```
        if(e_current!=0){
```

```
            filename[0]='E';
```

```
            for(i=1;i<=4;i++) filename[i]=current->number[i-1];
```

```
            filename[5]='.';
```

```
            filename[6]='d';
```

```
            filename[7]='a';
```

```
            filename[8]='t';
```

```
            filename[9]='\0';
```

```
            outfile2.open(filename);
```

```
            e_current= current->add_employee;
```

```
            while(e_current!=0){
```

```
                if(e_current->next_employee!=0)
```

```
                {
```

```
                    outfile2<<e_current->e_id<<" "<<e_current->
```

```
>e_fname<<" "
```

```

        <<e_current->e_lname<<" "<<e_current-
>e_type<<endl;
    }
    else
    {
        outfile2<<e_current->e_id<<" "<<e_current-
>e_fname
        <<" "<<e_current->e_lname<<" "<<e_current-
>e_type;
    }

    e_current=e_current->next_employee;

} //end employee while

outfile2.close();

} //end if employee
//WRITE RESERVATION INFORMATION TO ITS DATA FILE
(R####.dat)
r_current=current->add_reservation;
if(r_current!=0){
    filename[0]='R';
    for(i=1;i<=4;i++) filename[i]=current->number[i-1];
    filename[5]='.';
    filename[6]='d';
    filename[7]='a';
    filename[8]='t';
    filename[9]='\0';
    outfile2.open(filename);

    r_current=current->add_reservation;

    while(r_current!=0){
        if(current->add_reservation!=0)
        {
            outfile2<<r_current->phone<<" "<<r_current-
>r_fname<<
                " "<<r_current->r_lname<<"
"<<r_current->r_type<<endl;
        }
        else
        {
            outfile2<<r_current->phone<<" "<<r_current-
>r_fname<<
                " "<<r_current->r_lname<<" "<<r_current-
>r_type;
        }

        r_current=r_current->next_reservation;

    } //end reservation while

    outfile2.close();

} //end if reservation

```

```
        current=current->next;
    }//end flight while
    outfile.close();
}//end if
}
```