****

**National University of Computer and Emerging Sciences**

Object Oriented Programming (OOP)

**“Final Project”**

**STUDENT NAME:** Huzaifa Anjum

**REGISTRATION No:**  20i-2319

**DEGREE PROGRAM:** BSSE

**Section:** G

**SUBJECT NAME:** OOP

**DATE OF SUBMISSION:** December 20, 2021

**SUBMITTED TO:**  Mr. Bilal Khalid Dar

**STUDENT SIGNATURE:**

\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**MARKS:**

\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**REMARKS:**

\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**TEACHERS SIGNATURE:**

\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **Project Information**

**Functionalities or Features implemented in Project**

In this Project I have implemented all the following functionalities :

* **Functionalities of ADMIN**

1. **Login and can Logout**
2. **Add Local Flights in Schedule**
3. **Add International flights in Schedule**
4. **Remove and Block Local Flights in Schedule**
5. **Remove and Block International flights in Schedule**
6. **Have Info of all registered user and can verify VISA**
7. **Change his username and passwords**

* **Functionalities of Registered User (Passenger)**

1. **Signup and Login and can Logout**
2. **Check flight details local and international**
3. **Book flights Local and International (if admin verify VISA)**
4. **Can Check Balance**
5. **Change his username and passwords**
6. **Can see all his bookings**
7. **Cancel Reservation (25% Deduction)**

* **Functionalities of Unregistered User (Passenger)**

1. **Can only See Flight Schedule**

**--------------------------------------------------------------------------------------**

**OOP concept used in Project**

In this Project I have implemented all the following functionalities :

1. **Inheritance**

Person (which have common features)which is Parent class and its two child classes because there are two type of persons who are using this app. One is Admin and other is User means Passenger. These two are child classes of person and have some distinct attributes and functions.

Flights which is parent class and it has two child classes International and Local.

International flight have some extra attributes like country, Boolean indirect and indirect route. And local flight have attribute city. And all other attributes are same between these two types of flights so we have done inheritance in it.

1. **Abstraction**

We have made an abstract class Booking which have 2 unimplemented method double calculatecost(); and double calculateTax(); And implemented these two methods in the extended classes localbooking and interbooking. Because calculating cost and tax is different in both classes so we just declared these methods in abstract class Booking and implementation and body of these methods are in extended classes.

1. **Polymorphism**

We have done method overriding in localbooking and interbooking.

1. **Association**

* **Composition (Strong Association)**
* I have made two array list (list of objects) of local and international flights in Flight schedule(FlightreservationMenu).
* I have made two array list of localBooking and internBooking in User.

**How I have done composition:**

I have done composition by making these array list final by final keyword and initialized it in constructor of the class by the keyword of new.

**Reasons of doing Composition:**

Because if we delete a user account all his/her booking details will also be deleted

And if we delete Schedule then all flight details will also be deleted.

* **Aggregation (Weak Association)**
* I have made array list of User in the Jframe class SignUp and all the users who signup will be stored in that arraylist.
* I have made array list of Admin in AdminLogin Jframe class

And stored some hardcoded admin in it.

* I have made one object of local and international flights in localbooking and interbooking.

**How I have done aggregation:**

I have done aggregation by making object of a class in other class without using keyword final and initializing them where they are declared and in booking objects are initialized by equalizing the object came from the parameter of the constructor.

**Reasons of doing Composition:**

Because destroying bookingdetails or destroying users account doesn’t destroy flights or destroying signup class doesn’t destroy all users. The users can exist independently.

1. **Static Keyword**

* I have declared the object of flightreservationmenu in admin as static.
* I have declared the array list of admin and user as static.
* I also have initialized some variable like serialnumber as static.

**Reasons of using static:**

Because I want these attributes or object same in all classes and they are being used in many different classes.

If I don’t declare them as static then their values will be changed or become null when called them in some other classes.

# **Some Screenshots**

|  |
| --- |
|  |

# **Project Code**

**import** java.util.ArrayList;

**import** java.util.Random;

**import** java.util.\*;

**import** java.awt.BorderLayout;

**import** java.awt.EventQueue;

**import** javax.swing.JFrame;

**import** javax.swing.JPanel;

**import** javax.swing.JPasswordField;

**import** javax.swing.JRadioButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JSpinner;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**import** javax.swing.ListSelectionModel;

**import** javax.swing.SpinnerNumberModel;

**import** javax.swing.border.EmptyBorder;

**import** javax.swing.border.LineBorder;

**import** javax.swing.border.MatteBorder;

**import** javax.swing.table.DefaultTableModel;

**import** java.awt.Color;

**import** javax.swing.JLabel;

**import** java.awt.Font;

**import** javax.swing.ButtonGroup;

**import** javax.swing.DefaultComboBoxModel;

**import** javax.swing.JButton;

**import** javax.swing.JCheckBox;

**import** javax.swing.JComboBox;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.MouseAdapter;

**import** java.awt.event.MouseEvent;

**import** java.util.ArrayList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

AdminLogin.*addSomeAdmins*();

SignUp.*addsomeUsers*();

Admin.*loc*();

Admin.*inter*();

MainMenu main =**new** MainMenu();

main.setVisible(**true**);

}

}

**class** Person {

**protected** **int** id;

**protected** String Name;

**protected** String CNIC;

**protected** **int** age;

**protected** String Gender;

**protected** String userName;

**protected** String password;

**protected** **boolean** login\_status=**false**;

**public** Person(){

}

**public** Person(**int** id, String name,String userName ,String cNIC,String gender ,**int** age, String password) {

**this**.id = id;

Name = name;

CNIC = cNIC;

**this**.age = age;

Gender = gender;

**this**.userName = userName;

**this**.password = password;

**this**.login\_status = **false**;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** Name;

}

**public** **void** setName(String name) {

Name = name;

}

**public** String getCNIC() {

**return** CNIC;

}

**public** **void** setCNIC(String cNIC) {

CNIC = cNIC;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getGender() {

**return** Gender;

}

**public** **void** setGender(String gender) {

Gender = gender;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

**public** **boolean** isLogin\_status() {

**return** login\_status;

}

**public** **void** setLogin\_status(**boolean** login\_status) {

**this**.login\_status = login\_status;

}

}

**class** User **extends** Person{

**private** **boolean** visa=**false**;

**private** String passport\_no;

**private** **double** balance;

**final** ArrayList <localBooking> Localbooked;

**final** ArrayList <interBooking> Interbooked;

**public** User(**int** id, String name, String userName, String cNIC, String gender, **int** age, String password,String passport) {

**super**(id, name, userName, cNIC, gender, age, password);

**this**.passport\_no=passport;

**this**.setBalance(1000000);

Localbooked=**new** ArrayList<>();

Interbooked=**new** ArrayList<>();

}

**public** **boolean** isVisa() {

**return** visa;

}

**public** **void** setVisa(**boolean** visa) {

**this**.visa = visa;

}

**public** String getPassport\_no() {

**return** passport\_no;

}

**public** **void** setPassport\_no(String passport\_no) {

**this**.passport\_no = passport\_no;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** **void** Book\_Local(Local flight ,**int** bseats,**int** eseats) {

localBooking lb=**new** localBooking(flight,bseats,eseats);

lb.cost=lb.calculateCost();

lb.GovtTAX=lb.calculateTax();

lb.ammountToPay=lb.TotalBill();;

Localbooked.add(lb);

balance=balance-lb.ammountToPay;

}

**public** **void** Book\_Inter(International flight ,**int** bseats,**int** eseats) {

interBooking ib=**new** interBooking(flight,bseats,eseats);

ib.cost=ib.calculateCost();

ib.GovtTAX=ib.calculateTax();

ib.ammountToPay=ib.TotalBill();;

Interbooked.add(ib);

balance=balance-ib.ammountToPay;

}

**public** **boolean** cancelLocal(**int** id) {

**for**(**int** i=0;i<Localbooked.size();i++) {

**if**(Localbooked.get(i).booking\_id==id) {

**if**(Localbooked.get(i).isCancelled==**false**) {

Localbooked.get(i).isCancelled=**true**;

**double** refund=Localbooked.get(i).ammountToPay\*75/100;

balance=balance+refund;

**return** **true**;

}

}

}

**return** **false**;

}

**public** **boolean** cancelinter(**int** id) {

**for**(**int** i=0;i<Interbooked.size();i++) {

**if**(Interbooked.get(i).booking\_id==id) {

**if**(Interbooked.get(i).isCancelled==**false**) {

Interbooked.get(i).isCancelled=**true**;

**double** refund=Interbooked.get(i).ammountToPay\*75/100;

balance=balance+refund;

**return** **true**;

}

}

}

**return** **false**;

}

}

**class** Admin **extends** Person{

**public** Admin() {

id=2319;

Name="admin";

age=21;

Gender="Male";

userName="admin";

password="admin";

login\_status=**false**;

}

**public** Admin(**int** id,String name,**int** age,String user,String pass) {

**this**.id=id;

**this**.Name=name;

**this**.age=age;

**this**.userName=user;

**this**.password=pass;

login\_status=**false**;

}

**static** FlightReservationMenu *FM*=**new** FlightReservationMenu();

**public** **void** addLocal(String airport, String northSouth, **double** distance, **int** time\_Hours, String fdate, String ftime,String atime,

**int** seat\_economy, **int** seat\_bussiness, String city) {

Local lflight=**new** Local(*FM*.*localSerialNo*,airport,northSouth,distance,time\_Hours,fdate,ftime,atime,seat\_economy,seat\_bussiness,city);

*FM*.localFlights.add(lflight);

*FM*.*localSerialNo*++;

}

**public** **void** addInternational(String airport, String northSouth, **double** distance, **int** time\_Hours, String fdate,

String ftime,String atime,**int** seat\_economy, **int** seat\_bussiness, String country,**boolean** isitIndirect,String route) {

International iflight=**new** International(*FM*.*InterSerialNo* ,airport, northSouth, distance, time\_Hours, fdate,ftime,atime,seat\_economy,seat\_bussiness,country,isitIndirect,route);

*FM*.interFlights.add(iflight);

*FM*.*InterSerialNo*++;

}

**public** **void** removelocal(**int** serialno,**int** id) {

**for**(**int** i=0;i<*FM*.localFlights.size();i++) {

**if**(*FM*.localFlights.get(i).getId()==id) {

*FM*.localFlights.remove(i);

}

}

**for**(**int** j=serialno;j<*FM*.localFlights.size();j++) {

*FM*.localFlights.get(j).setSerialNo(j);;

}

*FM*.*localSerialNo*--;

}

**public** **void** removeinternational(**int** serialno,**int** id) {

**for**(**int** i=0;i<*FM*.interFlights.size();i++) {

**if**(*FM*.interFlights.get(i).getId()==id) {

*FM*.interFlights.remove(i);

}

}

**for**(**int** j=serialno;j<*FM*.interFlights.size();j++) {

*FM*.interFlights.get(j-1).setSerialNo(j);;

}

*FM*.*InterSerialNo*--;

}

**public** **void** blocklocalflight(**int** id) {

**for**(**int** i=0;i<*FM*.localFlights.size();i++) {

**if**(*FM*.localFlights.get(i).getId()==id) {

*FM*.localFlights.get(i).setOpen(**false**);;

}

}

}

**public** **void** blockinterflight(**int** id) {

**for**(**int** i=0;i<*FM*.interFlights.size();i++) {

**if**(*FM*.interFlights.get(i).getId()==id) {

*FM*.interFlights.get(i).setOpen(**false**);

}

}

}

**public** **void** openlocalflight(**int** id) {

**for**(**int** i=0;i<*FM*.localFlights.size();i++) {

**if**(*FM*.localFlights.get(i).getId()==id) {

*FM*.localFlights.get(i).setOpen(**true**);;

}

}

}

**public** **void** openinterflight(**int** id) {

**for**(**int** i=0;i<*FM*.interFlights.size();i++) {

**if**(*FM*.interFlights.get(i).getId()==id) {

*FM*.interFlights.get(i).setOpen(**true**);

}

}

}

**static** **void** loc() {

Local l1=**new** Local(*FM*.*localSerialNo*,"Lahore","North",300,2,"20 / 12 / 2021","3:20 am","5:20 am",40,20,"Islamabad");*FM*.*localSerialNo*++;

Local l2=**new** Local(*FM*.*localSerialNo*,"Lahore","South",700,4,"21 / 12 / 2021","1:30 am","5:30 am",20,20,"Karachi");*FM*.*localSerialNo*++;

Local l3=**new** Local(*FM*.*localSerialNo*,"Karachi","South",700,4,"22 / 12 / 2021","12:00 am","4:00 am",30,30,"Lahore");*FM*.*localSerialNo*++;

Local l4=**new** Local(*FM*.*localSerialNo*,"Peshawar","North",300,2,"23 / 12 / 2021","3:20 am","5:20 am",30,20,"Quetta");*FM*.*localSerialNo*++;

Local l5=**new** Local(*FM*.*localSerialNo*,"Quetta","South",300,2,"24 / 12 / 2021","8:50 pm","10:50 pm",40,20,"Karachi");*FM*.*localSerialNo*++;

Local l6=**new** Local(*FM*.*localSerialNo*,"Islamabad","North",900,5,"25 / 12 / 2021","7:20 pm","11:20 pm",40,20,"Karachi");*FM*.*localSerialNo*++;

*FM*.localFlights.add(l1);

*FM*.localFlights.add(l2);

*FM*.localFlights.add(l3);

*FM*.localFlights.add(l4);

*FM*.localFlights.add(l5);

*FM*.localFlights.add(l6);

}

**static** **void** inter() {

International i1 =**new** International(*FM*.*InterSerialNo*,"Islamabad","South",2000,3,"20 / 12 / 2021","3:20 am","6:20 am",30,20,"India",**false**,**null**);*FM*.*InterSerialNo*++;

International i2 =**new** International(*FM*.*InterSerialNo*,"Karachi","South",3000,4,"23 / 12 / 2021","8:50 pm","12:50 am",40,20,"China",**false**,**null**);*FM*.*InterSerialNo*++;

International i3 =**new** International(*FM*.*InterSerialNo*,"Lahore","North",4000,6,"21 / 12 / 2021","1:30 am","7:30 am",30,20,"Dubai",**true**,"UAE");*FM*.*InterSerialNo*++;

International i4 =**new** International(*FM*.*InterSerialNo*,"Peshawar","North",1500,3,"22 / 12 / 2021","12:00 am","3:00 am",30,20,"Turkey",**true**,"China");*FM*.*InterSerialNo*++;

International i5 =**new** International(*FM*.*InterSerialNo*,"Quetta","South",3000,5,"24 / 12 / 2021","8:10 pm","1:10 am",30,20,"India",**false**,**null**);*FM*.*InterSerialNo*++;

International i6 =**new** International(*FM*.*InterSerialNo*,"Islamabad","South",1000,1,"25 / 12 / 2021","7:20 pm","8:20 pm",30,20,"Oman",**true**,"Turkey");*FM*.*InterSerialNo*++;

*FM*.interFlights.add(i1);

*FM*.interFlights.add(i2);

*FM*.interFlights.add(i3);

*FM*.interFlights.add(i4);

*FM*.interFlights.add(i5);

*FM*.interFlights.add(i6);

}

}

**abstract** **class** Booking {

**protected** **int** booking\_id;

**protected** **int** Bseats;

**protected** **int** Eseats;

**protected** **double** cost;

**protected** **double** GovtTAX;

**protected** **double** ammountToPay;

**protected** **boolean** isCancelled;

**public** Booking(**int** bseats, **int** eseats) {

**this**.Bseats = bseats;

**this**.Eseats=eseats;

**this**.booking\_id=RandomIdGenerate();

}

**public** **abstract** **double** calculateCost() ; //One abstract Method

**public** **abstract** **double** calculateTax(); //second abstract method

**public** **double** TotalBill() {

ammountToPay=calculateCost()+calculateTax();

**return** ammountToPay;

}

**public** **int** RandomIdGenerate() {

Random rand=**new** Random();

**int** id=rand.nextInt(100)+999;

**return** id;

}

}

**class** localBooking **extends** Booking {

Local bookflight;

**public** localBooking( Local bf,**int** bseats,**int** eseats) {

**super**(bseats, eseats);

**this**.bookflight=bf;

}

**public** **double** calculateCost() {

**int** bussiness=bookflight.time\_Hours\*10000\*Bseats;

**int** economy=bookflight.time\_Hours\*8000\*Eseats;

cost= bussiness+economy;

**return** cost;

}

@Override

**public** **double** calculateTax() {

GovtTAX=cost\*5/100;

**return** GovtTAX;

}

}

**class** interBooking **extends** Booking {

International bookflight;

**public** interBooking( International bf,**int** bseats,**int** eseats) {

**super**(bseats, eseats);

**this**.bookflight=bf;

}

**public** **double** calculateCost() {

**int** bussiness=bookflight.time\_Hours\*10000\*Bseats;

**int** economy=bookflight.time\_Hours\*8000\*Eseats;

cost= bussiness+economy;

**return** cost;

}

@Override

**public** **double** calculateTax() {

GovtTAX=cost\*10/100;

**return** GovtTAX;

}

}

**class** Flights {

**protected** **int** SerialNo;

**protected** **int** id;

**protected** String Airport;

**protected** String NorthSouth;

**protected** **double** distance;

**protected** **int** time\_Hours;

**protected** String fdate;

**protected** String ftime;

**protected** String arrivalTime;

**protected** **int** seat\_economy;

**protected** **int** seat\_bussiness;

**protected** **boolean** isOpen;

**public** Flights(**int** srNo,String airport, String northSouth, **double** distance, **int** time\_Hours, String fdate,

String ftime, String atime ,**int** seat\_economy, **int** seat\_bussiness) {

**this**.SerialNo=srNo;

**this**.id=random();

**this**.Airport = airport;

**this**.NorthSouth = northSouth;

**this**.distance = distance;

**this**.time\_Hours = time\_Hours;

**this**.fdate = fdate;

**this**.ftime = ftime;

**this**.arrivalTime=atime;

**this**.seat\_economy = seat\_economy;

**this**.seat\_bussiness = seat\_bussiness;

**this**.isOpen = **true**;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getAirport() {

**return** Airport;

}

**public** **void** setAirport(String airport) {

Airport = airport;

}

**public** String getNorthSouth() {

**return** NorthSouth;

}

**public** **void** setNorthSouth(String northSouth) {

NorthSouth = northSouth;

}

**public** **double** getDistance() {

**return** distance;

}

**public** **void** setDistance(**double** distance) {

**this**.distance = distance;

}

**public** **int** getTime\_Hours() {

**return** time\_Hours;

}

**public** **void** setTime\_Hours(**int** time\_Hours) {

**this**.time\_Hours = time\_Hours;

}

**public** String getFdate() {

**return** fdate;

}

**public** **void** setFdate(String fdate) {

**this**.fdate = fdate;

}

**public** String getFtime() {

**return** ftime;

}

**public** **void** setFtime(String ftime) {

**this**.ftime = ftime;

}

**public** String getArrivalTime() {

**return** arrivalTime;

}

**public** **void** setArrivalTime(String arrivalTime) {

**this**.arrivalTime = arrivalTime;

}

**public** **int** getSeat\_economy() {

**return** seat\_economy;

}

**public** **void** setSeat\_economy(**int** seat\_economy) {

**this**.seat\_economy = seat\_economy;

}

**public** **int** getSeat\_bussiness() {

**return** seat\_bussiness;

}

**public** **void** setSeat\_bussiness(**int** seat\_bussiness) {

**this**.seat\_bussiness = seat\_bussiness;

}

**public** **boolean** isOpen() {

**return** isOpen;

}

**public** **void** setOpen(**boolean** isOpen) {

**this**.isOpen = isOpen;

}

**public** **int** getSerialNo() {

**return** SerialNo;

}

**public** **void** setSerialNo(**int** serialNo) {

SerialNo = serialNo;

}

**public** **int** random() {

Random random=**new** Random();

**int** num=random.nextInt(2000)+1000;

**return** num;

}

}

**class** International **extends** Flights{

**private** String Country;

**private** String indirectRouteCountry;

**private** **boolean** isIndierct;

**public** International(**int** srNo,String airport, String northSouth, **double** distance, **int** time\_Hours, String fdate,

String ftime,String atime ,**int** seat\_economy, **int** seat\_bussiness, String country, **boolean** isItindirect, String route) {

**super**(srNo,airport, northSouth, distance, time\_Hours, fdate, ftime, atime,seat\_economy, seat\_bussiness);

**this**.Country=country;

**this**.setIndierct(isItindirect);

**this**.setIndirectRouteCountry(route);

}

**public** String getCountry() {

**return** Country;

}

**public** **void** setCountry(String country) {

Country = country;

}

**public** String getIndirectRouteCountry() {

**return** indirectRouteCountry;

}

**public** **void** setIndirectRouteCountry(String indirectRouteCountry) {

**this**.indirectRouteCountry = indirectRouteCountry;

}

**public** **boolean** isIndierct() {

**return** isIndierct;

}

**public** **void** setIndierct(**boolean** isIndierct) {

**this**.isIndierct = isIndierct;

}

}

**class** Local **extends** Flights{

**private** String City;

**public** Local(**int** srNo,String airport, String northSouth, **double** distance, **int** time\_Hours, String fdate, String ftime,

String atime,**int** seat\_economy, **int** seat\_bussiness, String city) {

**super**(srNo, airport, northSouth, distance, time\_Hours, fdate, ftime, atime,seat\_economy, seat\_bussiness);

**this**.City=city;

}

**public** String getCity() {

**return** City;

}

**public** **void** setCity(String city) {

City = city;

}

}

**class** FlightReservationMenu {

**static** **int** *localSerialNo*=0;

**static** **int** *InterSerialNo*=0;

**final** ArrayList <Local> localFlights ;

**final** ArrayList <International> interFlights ;

**public** FlightReservationMenu() {

**this**.localFlights = **new** ArrayList <>() ;

**this**.interFlights = **new** ArrayList <>();

}

}

**class** Formats {

**static** **public** **boolean** PasswordFormat(String pass) {

**boolean** specialcheck=**false**;

**boolean** emptycheck=**false**;

**boolean** numcheck=**false**;

**boolean** capitalcheck=**false**;

**boolean** smallcheck=**false**;

String special = "!@#$%&\*()'+,-./:;<=>?[]^\_`{|}";

String num= "0123456789";

String capital="ABCDEFGHIJKLMNOPQRSTUVWXYZ";

String small="abcdefghijklmnopqrstuvwxyz";

**char**[] passChar=pass.toCharArray();

String[] p=**new** String[pass.length()];

**for**(**int** i=0;i<pass.length();i++) {

p[i]=Character.*toString*(passChar[i]);

}

**if**(!pass.isEmpty()) {

emptycheck=**true**;

}

**for**(**int** i=0;i<pass.length();i++) {

**if**(special.contains(p[i])) {

specialcheck= **true**;

}

**if**(num.contains(p[i])) {

numcheck= **true**;

}

**if**(capital.contains(p[i])) {

capitalcheck= **true**;

}

**if**(small.contains(p[i])) {

smallcheck= **true**;

}

}

**if**(smallcheck && capitalcheck && numcheck && specialcheck && emptycheck) {

**return** **true**;

}

**return** **false**;

}

**static** **public** **boolean** Passwordlength(String pass) {

**if**(pass.length()>=8) {

**return** **true**;

}

**return** **false**;

}

**static** **public** **boolean** Passwordmatch(String pass,String confirm) {

**if**(pass.equals(confirm)) {

**return** **true**;

}

**return** **false**;

}

**static** **public** **boolean** Namelength(String name) {

**if**(name.length()<37 && name.length()>2) {

**return** **true**;

}

**return** **false**;

}

**static** **public** **boolean** cniclength(String cnic) {

**if**(cnic.length()==13) {

**return** **true**;

}

**return** **false**;

}

**static** **public** **boolean** Passportlength(String passport) {

**if**(passport.length()==9) {

**return** **true**;

}

**return** **false**;

}

**static** **boolean** cnicavailable(ArrayList<User> user,String cnic) {

**for**(User x: user) {

**if**(x.CNIC.equalsIgnoreCase(cnic)) {

**return** **false**;

}

}

**return** **true**;

}

**static** **boolean** cnicFormat(String cnic){

**try** {

**long** temp=Long.*parseLong*(cnic);

**return** **true**;

}**catch**(Exception E) {

**return** **false**;

}

}

**static** **boolean** passportFormat(String passport){

**try** {

**long** temp=Long.*parseLong*(passport);

**return** **true**;

}**catch**(Exception E) {

**return** **false**;

}

}

**static** **boolean** NameFormatFun(String Name) {

String specialCharactersString = "!@#$%&\*()'+,-./:;<=>?[]^\_`{|}0123456789";

**char**[] NameChar=Name.toCharArray();

String[] n=**new** String[Name.length()];

**for**(**int** i=0;i<Name.length();i++) {

n[i]=Character.*toString*(NameChar[i]);

}

**for**(**int** i=0;i<Name.length();i++) {

**if**(specialCharactersString.contains(n[i])) {

**return** **false**;

}

}

**if**(Name.isEmpty()) {

**return** **false**;

}

**return** **true**;

}

**static** **boolean** usernameAvailable(ArrayList<User> user,String UserN) {

**for**(User x: user) {

**if**(x.userName.equalsIgnoreCase(UserN)) {

**return** **false**;

}

}

**return** **true**;

}

}

**class** MainMenu **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

MainMenu frame = **new** MainMenu();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** MainMenu() {

setTitle("New-PAK Airline Flight System (NPAFS)");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JButton signB = **new** JButton("SIGN UP");

signB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SignUp s=**new** SignUp();

s.setVisible(**true**);

dispose();

}

});

signB.setBounds(573, 177, 232, 48);

contentPane.add(signB);

JButton userloginB = **new** JButton("USER LOGIN");

userloginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(isanyoneLoggedin()) {

UserLoginSuccess uls =**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

**else** **if**(!isanyoneLoggedin()) {

UserLogin ul=**new** UserLogin();

ul.setVisible(**true**);

dispose();

}

}

});

userloginB.setBounds(573, 316, 232, 48);

contentPane.add(userloginB);

JButton adminloginB = **new** JButton("ADMIN LOGIN");

adminloginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al =**new** AdminLogin();

**if**(al.*admin*.login\_status==**true**) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

dispose();

}

**else** {

al.setVisible(**true**);

dispose();

}

}

});

adminloginB.setBounds(573, 451, 232, 48);

contentPane.add(adminloginB);

JButton withoutloginB = **new** JButton("CONTINUE WITHOUT LOGGING IN");

withoutloginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

FlightDetails fd =**new** FlightDetails();

fd.setVisible(**true**);

dispose();

}

});

withoutloginB.setBounds(573, 583, 232, 48);

contentPane.add(withoutloginB);

JLabel lblHomePage = **new** JLabel("HOME PAGE");

lblHomePage.setForeground(Color.***BLUE***);

lblHomePage.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblHomePage.setBounds(592, 93, 197, 63);

contentPane.add(lblHomePage);

}

**boolean** isanyoneLoggedin() {

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).login\_status==**true** ) {

**return** **true**;

}

}

**return** **false**;

}

}

**class** AccountSettingAdmin **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField uTF;

**private** JPasswordField passTF;

**private** JPasswordField confTF;

**private** JPasswordField oldTF;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

AccountSettingAdmin frame = **new** AccountSettingAdmin();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** AccountSettingAdmin() {

setTitle("Admin Account Settings");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 524, 699);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("ADMIN ACCOUNT SETTINGS");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(61, 48, 325, 37);

contentPane.add(lblNewLabel);

JLabel msgL = **new** JLabel(" ");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

msgL.setBounds(61, 483, 325, 14);

contentPane.add(msgL);

JLabel msg2L = **new** JLabel(" ");

msg2L.setForeground(Color.***RED***);

msg2L.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

msg2L.setBounds(61, 519, 325, 14);

contentPane.add(msg2L);

uTF = **new** JTextField();

uTF.setEditable(**false**);

uTF.setBounds(159, 175, 227, 20);

contentPane.add(uTF);

uTF.setColumns(10);

passTF = **new** JPasswordField();

passTF.setEditable(**false**);

passTF.setBounds(159, 325, 227, 20);

contentPane.add(passTF);

confTF = **new** JPasswordField();

confTF.setEditable(**false**);

confTF.setBounds(159, 374, 227, 20);

contentPane.add(confTF);

JLabel unameL = **new** JLabel("New Username:");

unameL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

unameL.setBounds(46, 178, 103, 14);

contentPane.add(unameL);

JLabel lblNewPassword = **new** JLabel("New Password:");

lblNewPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblNewPassword.setBounds(46, 328, 103, 14);

contentPane.add(lblNewPassword);

JLabel lblReenter = **new** JLabel("re-Enter:");

lblReenter.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblReenter.setBounds(46, 377, 103, 14);

contentPane.add(lblReenter);

oldTF = **new** JPasswordField();

oldTF.setBounds(159, 436, 227, 20);

contentPane.add(oldTF);

JLabel lblOldPassword = **new** JLabel("Old Password: ");

lblOldPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblOldPassword.setBounds(46, 439, 103, 14);

contentPane.add(lblOldPassword);

JRadioButton rb1 = **new** JRadioButton("Change Username: ");

rb1.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(rb1.isSelected()) {

uTF.setEditable(**true**);

}

**if**(!rb1.isSelected()) {

uTF.setEditable(**false**);

}

}

});

rb1.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

rb1.setBackground(Color.***ORANGE***);

rb1.setBounds(123, 116, 171, 23);

contentPane.add(rb1);

JRadioButton rb2 = **new** JRadioButton("Change Password: ");

rb2.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(rb2.isSelected()) {

passTF.setEditable(**true**);

confTF.setEditable(**true**);

}

**if**(!rb2.isSelected()) {

passTF.setEditable(**false**);

confTF.setEditable(**false**);

}

}

});

rb2.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

rb2.setBackground(Color.***ORANGE***);

rb2.setBounds(123, 269, 171, 23);

contentPane.add(rb2);

JButton confirmB = **new** JButton("Confirm");

confirmB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

msgL.setText(" ");

msg2L.setText(" ");

AdminLogin al =**new** AdminLogin();

**boolean** passformat=Formats.*PasswordFormat*(passTF.~~getText~~());

**boolean** passlength=Formats.*Passwordlength*(passTF.~~getText~~());

**boolean** passmatch=**false**;

**if**(passTF.~~getText~~().equals(confTF.~~getText~~())){

passmatch=**true**;

}

**if**(!rb1.isSelected() && !rb2.isSelected()) {

msgL.setText("select atleast 1 button to change account details");

}

**if**(rb1.isSelected()) {

**if**(oldTF.~~getText~~().equals(al.*admin*.getPassword())) {

al.*admin*.setUserName(uTF.getText());

msgL.setText("UserName Updated");

DetailsUpdatedBox dup =**new** DetailsUpdatedBox();

dup.setVisible(**true**);

}

**else** {

msgL.setText("incorrect Old Password");

}

}

**if**(rb2.isSelected()) {

**if**(!passformat || !passlength) {

msg2L.setText("minimum 8 character with upper lower num and symbols");

}

**else** **if**(!passmatch) {

msg2L.setText("password doesn't Match Try Again");

}

**else** **if**(passmatch && passformat && passlength) {

**if**(oldTF.~~getText~~().equals(al.*admin*.getPassword())) {

al.*admin*.setPassword(passTF.~~getText~~());

DetailsUpdatedBox dup =**new** DetailsUpdatedBox();

dup.setVisible(**true**);

msg2L.setText("Password Updated");

}

**else** {

msgL.setText("incorrect Old Password");

}

}

}

}

});

confirmB.setBounds(159, 559, 89, 23);

contentPane.add(confirmB);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(159, 608, 89, 23);

contentPane.add(backB);

}

}

**class** AccountSettingUser **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField uTF;

**private** JPasswordField passTF;

**private** JPasswordField confTF;

**private** JPasswordField oldTF;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

AccountSettingUser frame = **new** AccountSettingUser();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** AccountSettingUser() {

setTitle("Passenger Account Settings");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 495, 699);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Passenger ACCOUNT SETTINGS");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(61, 48, 325, 37);

contentPane.add(lblNewLabel);

JLabel msgL = **new** JLabel(" ");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

msgL.setBounds(61, 483, 325, 14);

contentPane.add(msgL);

JLabel msg2L = **new** JLabel(" ");

msg2L.setForeground(Color.***RED***);

msg2L.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

msg2L.setBounds(61, 519, 325, 14);

contentPane.add(msg2L);

JLabel unameL = **new** JLabel("New Username:");

unameL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

unameL.setBounds(46, 178, 103, 14);

contentPane.add(unameL);

JLabel lblNewPassword = **new** JLabel("New Password:");

lblNewPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblNewPassword.setBounds(46, 328, 103, 14);

contentPane.add(lblNewPassword);

JLabel lblReenter = **new** JLabel("re-Enter:");

lblReenter.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblReenter.setBounds(46, 377, 103, 14);

contentPane.add(lblReenter);

uTF = **new** JTextField();

uTF.setEditable(**false**);

uTF.setBounds(159, 175, 196, 20);

contentPane.add(uTF);

uTF.setColumns(10);

passTF = **new** JPasswordField();

passTF.setEditable(**false**);

passTF.setBounds(159, 325, 196, 20);

contentPane.add(passTF);

confTF = **new** JPasswordField();

confTF.setEditable(**false**);

confTF.setBounds(159, 374, 196, 20);

contentPane.add(confTF);

oldTF = **new** JPasswordField();

oldTF.setBounds(159, 436, 196, 20);

contentPane.add(oldTF);

JLabel lblOldPassword = **new** JLabel("Old Password: ");

lblOldPassword.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblOldPassword.setBounds(46, 439, 103, 14);

contentPane.add(lblOldPassword);

JRadioButton rb1 = **new** JRadioButton("Change Username: ");

rb1.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(rb1.isSelected()) {

uTF.setEditable(**true**);

}

**if**(!rb1.isSelected()) {

uTF.setEditable(**false**);

}

}

});

rb1.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

rb1.setBackground(Color.***ORANGE***);

rb1.setBounds(123, 116, 171, 23);

contentPane.add(rb1);

JRadioButton rb2 = **new** JRadioButton("Change Password: ");

rb2.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(rb2.isSelected()) {

passTF.setEditable(**true**);

confTF.setEditable(**true**);

}

**if**(!rb2.isSelected()) {

passTF.setEditable(**false**);

confTF.setEditable(**false**);

}

}

});

rb2.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

rb2.setBackground(Color.***ORANGE***);

rb2.setBounds(123, 269, 171, 23);

contentPane.add(rb2);

JButton confirmB = **new** JButton("Confirm");

confirmB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

msgL.setText(" ");

msg2L.setText(" ");

**boolean** unameavailable=Formats.*usernameAvailable*(SignUp.*u*, uTF.getText());

**boolean** passformat=Formats.*PasswordFormat*(passTF.~~getText~~());

**boolean** passlength=Formats.*Passwordlength*(passTF.~~getText~~());

**boolean** passmatch=**false**;

**if**(passTF.~~getText~~().equals(confTF.~~getText~~())){

passmatch=**true**;

}

**if**(!rb1.isSelected() && !rb2.isSelected()) {

msgL.setText("select atleast 1 button to change account details");

}

**if**(rb1.isSelected()) {

**if**(unameavailable) {

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).login\_status==**true** ) {

**if**(SignUp.*u*.get(i).getPassword().equals(oldTF.~~getText~~())) {

SignUp.*u*.get(i).setUserName(uTF.getText());

msgL.setText("UserName Updated");

DetailsUpdatedBox dup =**new** DetailsUpdatedBox();

dup.setVisible(**true**);

}

**else** {

msgL.setText("incorrect Old Password");

}

}

}

}

**else** {

msgL.setText("Account already Exisits on this UserName");

}

}

**if**(rb2.isSelected()) {

**if**(!passformat || !passlength) {

msg2L.setText("minimum 8 character with upper lower num and symbols");

}

**else** **if**(!passmatch) {

msg2L.setText("password doesn't Match Try Again");

}

**else** **if**(passmatch && passformat && passlength) {

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).login\_status==**true** ) {

**if**(SignUp.*u*.get(i).getPassword().equals(oldTF.~~getText~~())) {

SignUp.*u*.get(i).setPassword(passTF.~~getText~~());

msg2L.setText("Password Updated");

DetailsUpdatedBox dup =**new** DetailsUpdatedBox();

dup.setVisible(**true**);

}

**else** {

msg2L.setText("incorrect Old Password");

}

}

}

}

}

}

});

confirmB.setBounds(159, 559, 89, 23);

contentPane.add(confirmB);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

backB.setBounds(159, 615, 89, 23);

contentPane.add(backB);

}

}

**class** addInter **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

addInter frame = **new** addInter();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** addInter() {

setTitle("ADMIN");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Add International Flights");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(464, 58, 326, 46);

contentPane.add(lblSuccessfullyLoggedIn);

JLabel airportL = **new** JLabel("AIRPORT(PAKISTAN):");

airportL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

airportL.setBounds(352, 115, 130, 14);

contentPane.add(airportL);

JLabel destL = **new** JLabel("DESTINATION COUNTRY: ");

destL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

destL.setBounds(352, 218, 137, 14);

contentPane.add(destL);

JLabel seatsL = **new** JLabel("SEATS:");

seatsL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

seatsL.setBounds(352, 266, 86, 14);

contentPane.add(seatsL);

JLabel distL = **new** JLabel("DISTANCE:");

distL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

distL.setBounds(352, 305, 130, 14);

contentPane.add(distL);

JLabel timeL = **new** JLabel("TIME:");

timeL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeL.setBounds(352, 353, 130, 14);

contentPane.add(timeL);

JLabel fdateL = **new** JLabel("FLIGHT DATE:");

fdateL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

fdateL.setBounds(352, 409, 130, 14);

contentPane.add(fdateL);

JLabel ftimeL = **new** JLabel("FLIGHT TIME:");

ftimeL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

ftimeL.setBounds(352, 476, 130, 14);

contentPane.add(ftimeL);

JComboBox airportCB = **new** JComboBox();

airportCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Islamabad", "Karachi", "Peshawar", "Quetta", "Lahore"}));

airportCB.setBounds(513, 115, 121, 22);

contentPane.add(airportCB);

JComboBox nsCB = **new** JComboBox();

nsCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"North", "South"}));

nsCB.setBounds(680, 115, 121, 22);

contentPane.add(nsCB);

JComboBox arrivalCB = **new** JComboBox();

arrivalCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "India", "China", "USA", "Turkey", "UAE", "Nepal", "Oman", "Syria", "Afghanistan", "Iran", "UK", "Brazil", "Egypt", "Hongkong", "Ireland", "Malayisa", "Yemen"}));

arrivalCB.setBounds(513, 214, 121, 22);

contentPane.add(arrivalCB);

JSpinner ecoS = **new** JSpinner();

ecoS.setModel(**new** SpinnerNumberModel(0, 0, 50, 1));

ecoS.setBounds(579, 263, 55, 20);

contentPane.add(ecoS);

JSpinner busS = **new** JSpinner();

busS.setModel(**new** SpinnerNumberModel(0, 0, 50, 1));

busS.setBounds(746, 263, 55, 20);

contentPane.add(busS);

JLabel economyL = **new** JLabel("Economy:");

economyL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

economyL.setBounds(494, 266, 86, 14);

contentPane.add(economyL);

JLabel lblBussiness = **new** JLabel("Bussiness:");

lblBussiness.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblBussiness.setBounds(655, 266, 86, 14);

contentPane.add(lblBussiness);

JLabel lblInKm = **new** JLabel("in KM");

lblInKm.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblInKm.setBounds(567, 305, 86, 14);

contentPane.add(lblInKm);

JSpinner disS = **new** JSpinner();

disS.setModel(**new** SpinnerNumberModel(100, 100, 100000, 1));

disS.setBounds(488, 302, 55, 20);

contentPane.add(disS);

JComboBox indirectCB = **new** JComboBox();

indirectCB.setEnabled(**false**);

indirectCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "India", "China", "USA", "Turkey", "UAE", "Nepal", "Oman", "Syria", "Afghanistan", "Iran", "UK", "Brazil", "Egypt", "Hongkong", "Ireland", "Malayisa", "Yemen"}));

indirectCB.setBounds(513, 160, 121, 22);

contentPane.add(indirectCB);

JComboBox hourCB = **new** JComboBox();

hourCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

hourCB.setBounds(488, 472, 55, 22);

contentPane.add(hourCB);

JComboBox minCB = **new** JComboBox();

minCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59"}));

minCB.setBounds(612, 472, 55, 22);

contentPane.add(minCB);

JComboBox daynightCB = **new** JComboBox();

daynightCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "am", "pm"}));

daynightCB.setBounds(728, 472, 55, 22);

contentPane.add(daynightCB);

JComboBox dayCB = **new** JComboBox();

dayCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"}));

dayCB.setBounds(488, 414, 55, 22);

contentPane.add(dayCB);

JComboBox monthCB = **new** JComboBox();

monthCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

monthCB.setBounds(612, 414, 55, 22);

contentPane.add(monthCB);

JComboBox yearCB = **new** JComboBox();

yearCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "2021", "2022", "2023"}));

yearCB.setBounds(728, 414, 55, 22);

contentPane.add(yearCB);

JLabel timeformat = **new** JLabel("( Hour / Min / ampm)");

timeformat.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat.setBounds(832, 476, 159, 14);

contentPane.add(timeformat);

JLabel timeformat\_1 = **new** JLabel("( Day / Month / Year )");

timeformat\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat\_1.setBounds(832, 418, 159, 14);

contentPane.add(timeformat\_1);

JLabel inhourL = **new** JLabel("in Hours");

inhourL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

inhourL.setBounds(567, 353, 86, 14);

contentPane.add(inhourL);

JLabel lblArrivalTime = **new** JLabel("ARRIVAL TIME:");

lblArrivalTime.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblArrivalTime.setBounds(352, 535, 114, 14);

contentPane.add(lblArrivalTime);

JComboBox arhourCB = **new** JComboBox();

arhourCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

arhourCB.setBounds(488, 531, 55, 22);

contentPane.add(arhourCB);

JComboBox arminCB = **new** JComboBox();

arminCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59"}));

arminCB.setBounds(612, 531, 55, 22);

contentPane.add(arminCB);

JComboBox ardaynightCB = **new** JComboBox();

ardaynightCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "am", "pm"}));

ardaynightCB.setBounds(728, 531, 55, 22);

contentPane.add(ardaynightCB);

JLabel timeformat\_2 = **new** JLabel("( Hour / Min / ampm)");

timeformat\_2.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat\_2.setBounds(832, 535, 159, 14);

contentPane.add(timeformat\_2);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setBounds(873, 115, 185, 14);

contentPane.add(msgL);

JLabel msg2L = **new** JLabel("");

msg2L.setForeground(Color.***RED***);

msg2L.setBounds(655, 218, 367, 14);

contentPane.add(msg2L);

JLabel msg3L = **new** JLabel("");

msg3L.setForeground(Color.***RED***);

msg3L.setBounds(488, 447, 326, 14);

contentPane.add(msg3L);

JLabel msg4L = **new** JLabel("");

msg4L.setForeground(Color.***RED***);

msg4L.setBounds(488, 506, 289, 14);

contentPane.add(msg4L);

JLabel msg5L = **new** JLabel("");

msg5L.setForeground(Color.***RED***);

msg5L.setBounds(680, 160, 367, 14);

contentPane.add(msg5L);

JLabel msg6L = **new** JLabel("");

msg6L.setForeground(Color.***RED***);

msg6L.setBounds(488, 567, 289, 14);

contentPane.add(msg6L);

JSpinner timeS = **new** JSpinner();

timeS.setModel(**new** SpinnerNumberModel(1, 1, 30, 1));

timeS.setBounds(488, 350, 55, 20);

contentPane.add(timeS);

JRadioButton indirectRB = **new** JRadioButton("Indirect");

indirectRB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(indirectRB.isSelected()) {

indirectCB.setEnabled(**true**);

}

**if**(!indirectRB.isSelected()) {

indirectCB.setEnabled(**false**);

}

}

});

indirectRB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

indirectRB.setBackground(Color.***ORANGE***);

indirectRB.setBounds(345, 160, 137, 23);

contentPane.add(indirectRB);

JButton addB = **new** JButton("ADD");

addB.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

}

});

addB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** airportcheck = **false**;

**boolean** indirectcheck=**false**;

**boolean** samecountrycheck=**false**;

**boolean** arrivalcheck = **false**;

**boolean** datecheck = **false**;

**boolean** timecheck = **false**;

**boolean** notindirect=**false**;

**boolean** arrTimecheck=**false**;

String airp = **null**;

String norSouth = **null**;

**double** dist;

**int** hours;

String fdate = **null**;

String ftime = **null**;

String arrivalTime=**null**;

**int** bseat;

**int** eseat;

String country = **null**;

**boolean** isitIndirect=**false**;

String indirectcountry=**null**;

**if**(!airportCB.getSelectedItem().toString().equals("-")) {

airportcheck=**true**;

}

**if**(!indirectRB.isSelected()) {

notindirect = **true**;

}

**else** **if**(indirectRB.isSelected() && !indirectCB.getSelectedItem().toString().equals("-")) {

indirectcheck = **true**;

}

**if**(!arrivalCB.getSelectedItem().toString().equals("-")) {

arrivalcheck = **true**;

}

**if**(!dayCB.getSelectedItem().toString().equals("-") && !monthCB.getSelectedItem().toString().equals("-") && !yearCB.getSelectedItem().toString().equals("-") ) {

datecheck=**true**;

}

**if**(!hourCB.getSelectedItem().toString().equals("-") && !minCB.getSelectedItem().toString().equals("-") && !daynightCB.getSelectedItem().toString().equals("-") ) {

timecheck=**true**;

}

**if**(!arhourCB.getSelectedItem().toString().equals("-") && !arminCB.getSelectedItem().toString().equals("-") && !ardaynightCB.getSelectedItem().toString().equals("-") ) {

arrTimecheck=**true**;

}

**if**(airportcheck && arrivalcheck && indirectcheck) {

**if**(!indirectCB.getSelectedItem().toString().equals(arrivalCB.getSelectedItem().toString())) {

samecountrycheck=**true**;

}

}

**if**(!airportcheck) {

msgL.setText("Select airport first");

}

**else** {

msgL.setText(" ");

}

**if**(!arrivalcheck) {

msg2L.setText("Select destination city first");

}

**else** {

msg2L.setText(" ");

}

**if**(!indirectRB.isSelected()) {

msg5L.setText(" ");

}

**else** **if**(!indirectcheck) {

msg5L.setText("select route country");

}

**else** **if**(!samecountrycheck) {

msg5L.setText("this cannot be an indirect flight");

}

**else** {

msg5L.setText(" ");

}

**if**(!datecheck) {

msg3L.setText("Select Date first");

}

**else** {

msg3L.setText(" ");

}

**if**(!timecheck) {

msg4L.setText("Select time first");

}

**else** {

msg4L.setText(" ");

}

**if**(!arrTimecheck) {

msg6L.setText("Select time first");

}

**else** {

msg6L.setText(" ");

}

norSouth=nsCB.getSelectedItem().toString();

dist=Double.*parseDouble*(disS.getValue().toString());

hours=Integer.*parseInt*(timeS.getValue().toString());

bseat=Integer.*parseInt*(busS.getValue().toString());

eseat=Integer.*parseInt*(ecoS.getValue().toString());

**if**(airportcheck) {

airp=airportCB.getSelectedItem().toString();

}

**if**(arrivalcheck) {

country=airportCB.getSelectedItem().toString();

}

**if**(datecheck) {

String day=dayCB.getSelectedItem().toString();

String month=monthCB.getSelectedItem().toString();

String year=yearCB.getSelectedItem().toString();

fdate=day.concat(" / ").concat(month).concat(" / ").concat(year);

}

**if**(timecheck) {

String hour=hourCB.getSelectedItem().toString();

String min=minCB.getSelectedItem().toString();

String ampm=daynightCB.getSelectedItem().toString();

ftime=hour.concat(" : ").concat(min).concat(" ").concat(ampm);

}

**if**(arrTimecheck) {

String hour=hourCB.getSelectedItem().toString();

String min=minCB.getSelectedItem().toString();

String ampm=daynightCB.getSelectedItem().toString();

arrivalTime=hour.concat(" : ").concat(min).concat(" ").concat(ampm);

}

**if**(indirectcheck && samecountrycheck) {

indirectcountry=indirectCB.getSelectedItem().toString();

isitIndirect=**true**;

}

**if**(airportcheck && arrivalcheck && (isitIndirect ||notindirect) && datecheck && timecheck && arrTimecheck) {

AdminLogin al=**new** AdminLogin();

al.*admin*.addInternational(airp, norSouth, dist, hours, fdate, ftime,arrivalTime ,bseat, eseat, country,isitIndirect,indirectcountry);

InterAdded ia=**new** InterAdded();

ia.setVisible(**true**);

dispose();

}

}

});

addB.setBounds(494, 609, 173, 23);

contentPane.add(addB);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(533, 661, 89, 23);

contentPane.add(backB);

}

}

**class** AddLocal **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

AddLocal frame = **new** AddLocal();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** AddLocal() {

setTitle("ADMIN");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(395, 0, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Add Local Flights");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(453, 61, 326, 46);

contentPane.add(lblSuccessfullyLoggedIn);

JLabel airportL = **new** JLabel("AIRPORT:");

airportL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

airportL.setBounds(352, 118, 130, 14);

contentPane.add(airportL);

JLabel destL = **new** JLabel("DESTINATION CITY: ");

destL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

destL.setBounds(352, 176, 137, 14);

contentPane.add(destL);

JLabel seatsL = **new** JLabel("SEATS:");

seatsL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

seatsL.setBounds(352, 231, 86, 14);

contentPane.add(seatsL);

JLabel distL = **new** JLabel("DISTANCE:");

distL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

distL.setBounds(352, 283, 130, 14);

contentPane.add(distL);

JLabel timeL = **new** JLabel("TIME:");

timeL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeL.setBounds(352, 339, 130, 14);

contentPane.add(timeL);

JLabel fdateL = **new** JLabel("FLIGHT DATE:");

fdateL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

fdateL.setBounds(352, 392, 130, 14);

contentPane.add(fdateL);

JLabel ftimeL = **new** JLabel("FLIGHT TIME:");

ftimeL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

ftimeL.setBounds(352, 446, 130, 14);

contentPane.add(ftimeL);

JComboBox airportCB = **new** JComboBox();

airportCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Islamabad", "Karachi", "Peshawar", "Quetta", "Lahore"}));

airportCB.setBounds(492, 118, 121, 22);

contentPane.add(airportCB);

JComboBox nsCB = **new** JComboBox();

nsCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"North", "South"}));

nsCB.setBounds(685, 118, 121, 22);

contentPane.add(nsCB);

JComboBox arrivalCB = **new** JComboBox();

arrivalCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Quetta", "Islamabad", "Karachi", "Peshawar", "Lahore"}));

arrivalCB.setBounds(492, 172, 121, 22);

contentPane.add(arrivalCB);

JSpinner ecoS = **new** JSpinner();

ecoS.setModel(**new** SpinnerNumberModel(0, 0, 50, 1));

ecoS.setBounds(579, 228, 55, 20);

contentPane.add(ecoS);

JSpinner busS = **new** JSpinner();

busS.setModel(**new** SpinnerNumberModel(0, 0, 50, 1));

busS.setBounds(751, 228, 55, 20);

contentPane.add(busS);

JLabel economyL = **new** JLabel("Economy:");

economyL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

economyL.setBounds(492, 231, 86, 14);

contentPane.add(economyL);

JLabel lblBussiness = **new** JLabel("Bussiness:");

lblBussiness.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblBussiness.setBounds(655, 231, 86, 14);

contentPane.add(lblBussiness);

JLabel lblInKm = **new** JLabel("in KM");

lblInKm.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblInKm.setBounds(567, 283, 86, 14);

contentPane.add(lblInKm);

JSpinner disS = **new** JSpinner();

disS.setModel(**new** SpinnerNumberModel(100, 100, 100000, 1));

disS.setBounds(492, 280, 51, 20);

contentPane.add(disS);

JComboBox hourCB = **new** JComboBox();

hourCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

hourCB.setBounds(492, 442, 55, 22);

contentPane.add(hourCB);

JComboBox minCB = **new** JComboBox();

minCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59"}));

minCB.setBounds(612, 442, 55, 22);

contentPane.add(minCB);

JComboBox daynightCB = **new** JComboBox();

daynightCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "am", "pm"}));

daynightCB.setBounds(728, 446, 55, 22);

contentPane.add(daynightCB);

JComboBox dayCB = **new** JComboBox();

dayCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31"}));

dayCB.setBounds(492, 388, 55, 22);

contentPane.add(dayCB);

JComboBox monthCB = **new** JComboBox();

monthCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

monthCB.setBounds(612, 388, 55, 22);

contentPane.add(monthCB);

JComboBox yearCB = **new** JComboBox();

yearCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "2021", "2022", "2023"}));

yearCB.setBounds(728, 388, 55, 22);

contentPane.add(yearCB);

JLabel lblArrivalTime = **new** JLabel("ARRIVAL TIME:");

lblArrivalTime.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

lblArrivalTime.setBounds(352, 502, 114, 14);

contentPane.add(lblArrivalTime);

JComboBox arhourCB = **new** JComboBox();

arhourCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}));

arhourCB.setBounds(492, 498, 55, 22);

contentPane.add(arhourCB);

JComboBox arminCB = **new** JComboBox();

arminCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59"}));

arminCB.setBounds(612, 498, 55, 22);

contentPane.add(arminCB);

JComboBox ardaynightCB = **new** JComboBox();

ardaynightCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "am", "pm"}));

ardaynightCB.setBounds(728, 498, 55, 22);

contentPane.add(ardaynightCB);

JLabel timeformat\_2 = **new** JLabel("( Hour / Min / ampm)");

timeformat\_2.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat\_2.setBounds(832, 502, 159, 14);

contentPane.add(timeformat\_2);

JLabel msg5L = **new** JLabel("");

msg5L.setForeground(Color.***RED***);

msg5L.setBounds(492, 534, 289, 14);

contentPane.add(msg5L);

JLabel timeformat = **new** JLabel("( Hour / Min / ampm)");

timeformat.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat.setBounds(832, 450, 159, 14);

contentPane.add(timeformat);

JLabel timeformat\_1 = **new** JLabel("( Day / Month / Year )");

timeformat\_1.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

timeformat\_1.setBounds(832, 392, 159, 14);

contentPane.add(timeformat\_1);

JLabel inhourL = **new** JLabel("in Hours");

inhourL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

inhourL.setBounds(567, 339, 86, 14);

contentPane.add(inhourL);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setBounds(876, 118, 185, 14);

contentPane.add(msgL);

JLabel msg2L = **new** JLabel("");

msg2L.setForeground(Color.***RED***);

msg2L.setBounds(655, 176, 367, 14);

contentPane.add(msg2L);

JLabel msg3L = **new** JLabel("");

msg3L.setForeground(Color.***RED***);

msg3L.setBounds(492, 421, 326, 14);

contentPane.add(msg3L);

JLabel msg4L = **new** JLabel("");

msg4L.setForeground(Color.***RED***);

msg4L.setBounds(490, 473, 289, 14);

contentPane.add(msg4L);

JSpinner timeS = **new** JSpinner();

timeS.setModel(**new** SpinnerNumberModel(1, 1, 30, 1));

timeS.setBounds(492, 336, 55, 20);

contentPane.add(timeS);

JButton addB = **new** JButton("ADD");

addB.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

}

});

addB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** airportcheck = **false**;

**boolean** arrivalcheck = **false**;

**boolean** departure\_destination = **false**;

**boolean** datecheck = **false**;

**boolean** timecheck = **false**;

**boolean** arrTimecheck=**false**;

String airp = **null**;

String norSouth = **null**;

**double** dist;

**int** hours;

String fdate = **null**;

String ftime = **null**;

String arrivalTime=**null**;

**int** bseat;

**int** eseat;

String city = **null**;

**if**(!airportCB.getSelectedItem().toString().equals("-")) {

airportcheck=**true**;

}

**if**(!arrivalCB.getSelectedItem().toString().equals("-")) {

arrivalcheck = **true**;

}

**if**(!dayCB.getSelectedItem().toString().equals("-") && !monthCB.getSelectedItem().toString().equals("-") && !yearCB.getSelectedItem().toString().equals("-") ) {

datecheck=**true**;

}

**if**(!hourCB.getSelectedItem().toString().equals("-") && !minCB.getSelectedItem().toString().equals("-") && !daynightCB.getSelectedItem().toString().equals("-") ) {

timecheck=**true**;

}

**if**(!arhourCB.getSelectedItem().toString().equals("-") && !arminCB.getSelectedItem().toString().equals("-") && !ardaynightCB.getSelectedItem().toString().equals("-") ) {

arrTimecheck=**true**;

}

**if**(airportcheck && arrivalcheck) {

**if**(!airportCB.getSelectedItem().toString().equals(arrivalCB.getSelectedItem().toString())) {

departure\_destination=**true**;

}

}

**if**(!airportcheck) {

msgL.setText("Select airport first");

}

**else** {

msgL.setText(" ");

}

**if**(!arrivalcheck) {

msg2L.setText("Select destination city first");

}

**else** **if**(!departure\_destination) {

msg2L.setText("Departure and Destination city shall not be same");

}

**else** {

msg2L.setText(" ");

}

**if**(!datecheck) {

msg3L.setText("Select Date first");

}

**else** {

msg3L.setText(" ");

}

**if**(!timecheck) {

msg4L.setText("Select time first");

}

**else** {

msg4L.setText(" ");

}

**if**(!arrTimecheck) {

msg5L.setText("Select time first");

}

**else** {

msg5L.setText(" ");

}

norSouth=nsCB.getSelectedItem().toString();

dist=Double.*parseDouble*(disS.getValue().toString());

hours=Integer.*parseInt*(timeS.getValue().toString());

bseat=Integer.*parseInt*(busS.getValue().toString());

eseat=Integer.*parseInt*(ecoS.getValue().toString());

**if**(airportcheck) {

airp=airportCB.getSelectedItem().toString();

}

**if**(arrivalcheck && departure\_destination) {

city=arrivalCB.getSelectedItem().toString();

}

**if**(datecheck) {

String day=dayCB.getSelectedItem().toString();

String month=monthCB.getSelectedItem().toString();

String year=yearCB.getSelectedItem().toString();

fdate=day.concat(" / ").concat(month).concat(" / ").concat(year);

}

**if**(timecheck) {

String hour=hourCB.getSelectedItem().toString();

String min=minCB.getSelectedItem().toString();

String ampm=daynightCB.getSelectedItem().toString();

ftime=hour.concat(" : ").concat(min).concat(" ").concat(ampm);

}

**if**(arrTimecheck) {

String hour=hourCB.getSelectedItem().toString();

String min=minCB.getSelectedItem().toString();

String ampm=daynightCB.getSelectedItem().toString();

arrivalTime=hour.concat(" : ").concat(min).concat(" ").concat(ampm);

}

**if**(airportcheck && arrivalcheck && departure\_destination && datecheck && timecheck && arrTimecheck) {

AdminLogin al=**new** AdminLogin();

al.*admin*.addLocal(airp, norSouth, dist, hours, fdate, ftime,arrivalTime, bseat, eseat, city);

LocalAdded la=**new** LocalAdded();

la.setVisible(**true**);

dispose();

}

}

});

addB.setBounds(492, 586, 173, 23);

contentPane.add(addB);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(533, 661, 89, 23);

contentPane.add(backB);

}

}

**class** AdminLogin **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField nameTF;

**private** JPasswordField passTF;

**static** ArrayList<Admin> *ad*=**new** ArrayList<>();

**static** Admin *admin*=**new** Admin();

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

AdminLogin frame = **new** AdminLogin();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** AdminLogin() {

setTitle("ADMIN LOGIN");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JLabel userL = **new** JLabel("ADMIN USERNAME:");

userL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

userL.setBounds(467, 219, 143, 14);

contentPane.add(userL);

JLabel passL = **new** JLabel("PASSWORD:");

passL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

passL.setBounds(467, 309, 143, 14);

contentPane.add(passL);

JLabel lblUserLoginMenu = **new** JLabel("ADMIN LOGIN MENU");

lblUserLoginMenu.setForeground(Color.***BLUE***);

lblUserLoginMenu.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblUserLoginMenu.setBounds(563, 110, 231, 33);

contentPane.add(lblUserLoginMenu);

nameTF = **new** JTextField();

nameTF.setBounds(667, 216, 213, 20);

contentPane.add(nameTF);

nameTF.setColumns(10);

passTF = **new** JPasswordField();

passTF.setBounds(667, 306, 213, 20);

contentPane.add(passTF);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setBounds(669, 349, 211, 14);

contentPane.add(msgL);

JButton loginB = **new** JButton("LOGIN");

loginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(nameTF.getText().equals(*admin*.userName) && passTF.~~getText~~().equals(*admin*.password)) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

*admin*.login\_status=**true**;

dispose();

}

**else** {

msgL.setText("Incorrect UserName or Password");

}

}

});

loginB.setBounds(669, 424, 132, 33);

contentPane.add(loginB);

JButton MainB = **new** JButton("Main Menu");

MainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainB.setBounds(669, 542, 132, 33);

contentPane.add(MainB);

}

**static** **void** addSomeAdmins() {

Admin admin2=**new** Admin(1769,"Wajahat",20,"admin2","admin2");

Admin admin3=**new** Admin(222,"Rayed",20,"admin3","admin3");

Admin admin4=**new** Admin(333,"Moiz",19,"admin4","admin4");

Admin admin5=**new** Admin(444,"Umar",22,"admin5","admin5");

Admin admin6=**new** Admin(555,"Haider",21,"admin6","admin6");

*ad*.add(admin2);

*ad*.add(admin3);

*ad*.add(admin4);

*ad*.add(admin5);

*ad*.add(admin6);

}

}

**class** AdminLoginSuccess **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

AdminLoginSuccess frame = **new** AdminLoginSuccess();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** AdminLoginSuccess() {

setTitle("ADMIN");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("ADMIN Successfully Logged in");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(494, 104, 326, 63);

contentPane.add(lblSuccessfullyLoggedIn);

JButton AddLocalB = **new** JButton("Add Flight Local");

AddLocalB.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

AddLocal al=**new** AddLocal();

al.setVisible(**true**);

dispose();

}

});

AddLocalB.setBounds(284, 217, 217, 43);

contentPane.add(AddLocalB);

JButton addInterB = **new** JButton("Add Flight InterNational");

addInterB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

addInter ai=**new** addInter();

ai.setVisible(**true**);

dispose();

}

});

addInterB.setBounds(284, 329, 217, 43);

contentPane.add(addInterB);

JButton removeLocalB = **new** JButton("Delete or Block Local Flight");

removeLocalB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockLocal dbl=**new** DeleteBlockLocal();

dbl.setVisible(**true**);

dispose();

}

});

removeLocalB.setBounds(284, 447, 217, 43);

contentPane.add(removeLocalB);

JButton removeInterB = **new** JButton("Delete or Block International Flight");

removeInterB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockInter dbi=**new** DeleteBlockInter();

dbi.setVisible(**true**);

dispose();

}

});

removeInterB.setBounds(284, 572, 217, 43);

contentPane.add(removeInterB);

JButton MainB = **new** JButton("Main Menu");

MainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m =**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainB.setBounds(720, 447, 224, 43);

contentPane.add(MainB);

JButton logoutB = **new** JButton("Logout");

logoutB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al =**new** AdminLogin();

al.*admin*.setLogin\_status(**false**);

dispose();

LogoutBox lb=**new** LogoutBox();

lb.setVisible(**true**);

}

});

logoutB.setBounds(720, 572, 224, 43);

contentPane.add(logoutB);

JButton settingB = **new** JButton("Account Settings");

settingB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AccountSettingAdmin as=**new** AccountSettingAdmin();

as.setVisible(**true**);

dispose();

}

});

settingB.setBounds(720, 329, 217, 43);

contentPane.add(settingB);

JButton userdetailsB = **new** JButton("Users Details");

userdetailsB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

allUsersDetails aud=**new** allUsersDetails();

aud.setVisible(**true**);

dispose();

}

});

userdetailsB.setBounds(720, 217, 217, 43);

contentPane.add(userdetailsB);

}

}

**class** allUsersDetails **extends** JFrame {

**private** JPanel contentPane;

**private** JTable userTable;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

allUsersDetails frame = **new** allUsersDetails();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** allUsersDetails() {

setTitle("All Users Details");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel localdetailsL = **new** JLabel("Users Details");

localdetailsL.setForeground(Color.***BLUE***);

localdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

localdetailsL.setBounds(515, 61, 326, 46);

contentPane.add(localdetailsL);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

msgL.setBounds(449, 419, 412, 24);

contentPane.add(msgL);

String[] heading= {"User ID","Name","UserName","CNIC","Gender","Age","Passport no","Visa (status)"};

String [][] users2D =**new** String[SignUp.*u*.size()][8];

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**for**(**int** j=0;j<8;j++) {

String id=String.*valueOf*(SignUp.*u*.get(i).id);

String age=String.*valueOf*(SignUp.*u*.get(i).age);

String visaverified=Boolean.*toString*(SignUp.*u*.get(i).isVisa());

users2D[i][0]=id;

users2D[i][1]=SignUp.*u*.get(i).Name;

users2D[i][2]=SignUp.*u*.get(i).userName;

users2D[i][3]=SignUp.*u*.get(i).CNIC;

users2D[i][4]=SignUp.*u*.get(i).Gender;

users2D[i][5]=age;

users2D[i][6]=SignUp.*u*.get(i).getPassport\_no();

users2D[i][7]=visaverified;

}

}

DefaultTableModel modeltable = **new** DefaultTableModel(users2D,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

userTable = **new** JTable(modeltable);

scrollPane.setViewportView(userTable);

userTable.setFillsViewportHeight(**true**);

userTable.setSurrendersFocusOnKeystroke(**true**);

userTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

userTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JButton verifyB = **new** JButton("Verify Visa");

verifyB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(userTable.getSelectedRow()==-1) {

msgL.setText("Please Select a User first");

}

**else** {

String userid=userTable.getValueAt(userTable.getSelectedRow(), 0).toString();

**int** idtoGiveVisa=Integer.*parseInt*(userid);

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).id==idtoGiveVisa) {

SignUp.*u*.get(i).setVisa(**true**);

VisaVerified vv=**new** VisaVerified();

vv.setVisible(**true**);

dispose();

}

}

}

}

});

verifyB.setBounds(530, 475, 166, 38);

contentPane.add(verifyB);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(554, 568, 117, 38);

contentPane.add(backB);

}

}

**class** BalanceBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

BalanceBox frame = **new** BalanceBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** BalanceBox() {

setTitle("Account Balance");

setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel balL = **new** JLabel("Balance(rs): ");

balL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

balL.setForeground(Color.***BLUE***);

balL.setBounds(10, 77, 140, 52);

contentPane.add(balL);

String userbalance = **null**;

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).login\_status==**true** ) {

userbalance= String.*valueOf*(SignUp.*u*.get(i).getBalance());

}

}

JLabel balance = **new** JLabel("");

balance.setForeground(Color.***BLUE***);

balance.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

balance.setBounds(188, 77, 236, 52);

contentPane.add(balance);

balance.setText(userbalance);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** BookInterFlight **extends** JFrame {

**private** JPanel contentPane;

**private** JTable table;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

BookInterFlight frame = **new** BookInterFlight();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** BookInterFlight() {

setTitle("Book International Flight");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel interdetailsL = **new** JLabel("Book International Flights");

interdetailsL.setForeground(Color.***BLUE***);

interdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

interdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(interdetailsL);

JLabel msg2L = **new** JLabel("");

msg2L.setForeground(Color.***RED***);

msg2L.setFont(**new** Font("Tahoma", Font.***BOLD***, 18));

msg2L.setBounds(502, 483, 371, 31);

contentPane.add(msg2L);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 18));

msgL.setBounds(502, 407, 371, 31);

contentPane.add(msgL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Indirect","in Between Country","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] searched2d =**new** String[SearchInternational.*searchedinter*.size()][15];

**for**(**int** i=0;i<SearchInternational.*searchedinter*.size();i++) {

**for**(**int** j=0;j<15;j++) {

String serialno=String.*valueOf*(SearchInternational.*searchedinter*.get(i).SerialNo);

String flightid=String.*valueOf*(SearchInternational.*searchedinter*.get(i).id) ;

String distance=String.*valueOf*(SearchInternational.*searchedinter*.get(i).distance);

String Hours=String.*valueOf*(SearchInternational.*searchedinter*.get(i).time\_Hours);

String Eseats=String.*valueOf*(SearchInternational.*searchedinter*.get(i).seat\_economy);

String Bseats=String.*valueOf*(SearchInternational.*searchedinter*.get(i).seat\_bussiness);

String indirectbool=Boolean.*toString*(SearchInternational.*searchedinter*.get(i).isIndierct());

String isopen=Boolean.*toString*(SearchInternational.*searchedinter*.get(i).isOpen);

searched2d[i][0]=serialno;

searched2d[i][1]=flightid;

searched2d[i][2]=SearchInternational.*searchedinter*.get(i).Airport;

searched2d[i][3]=SearchInternational.*searchedinter*.get(i).NorthSouth;

searched2d[i][4]=indirectbool;

searched2d[i][5]=SearchInternational.*searchedinter*.get(i).getIndirectRouteCountry();

searched2d[i][6]=SearchInternational.*searchedinter*.get(i).getCountry();

searched2d[i][7]=distance;

searched2d[i][8]=Hours;

searched2d[i][9]=SearchInternational.*searchedinter*.get(i).fdate;

searched2d[i][10]=SearchInternational.*searchedinter*.get(i).ftime;

searched2d[i][11]=SearchInternational.*searchedinter*.get(i).arrivalTime;

searched2d[i][12]=Eseats;

searched2d[i][13]=Bseats;

searched2d[i][14]=isopen;

}

}

DefaultTableModel intertable = **new** DefaultTableModel(searched2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 155, 1350, 218);

contentPane.add(scrollPane);

table = **new** JTable(intertable);

scrollPane.setViewportView(table);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

table.setFillsViewportHeight(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setSurrendersFocusOnKeystroke(**true**);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SearchInternational si=**new** SearchInternational();

si.setVisible(**true**);

si.*searchedinter*.clear();

dispose();

}

});

backB.setBounds(554, 647, 117, 38);

contentPane.add(backB);

JRadioButton bussinessB = **new** JRadioButton("Bussiness");

bussinessB.setForeground(Color.***BLACK***);

bussinessB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

bussinessB.setBackground(Color.***ORANGE***);

bussinessB.setBounds(460, 445, 109, 23);

contentPane.add(bussinessB);

JRadioButton economyB = **new** JRadioButton("Economy");

economyB.setForeground(Color.***BLACK***);

economyB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

economyB.setBackground(Color.***ORANGE***);

economyB.setBounds(667, 445, 109, 23);

contentPane.add(economyB);

ButtonGroup bb=**new** ButtonGroup();

bb.add(economyB);

bb.add(bussinessB);

JButton btnNewButton = **new** JButton("Confirm Booking");

btnNewButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** buttoncheck=**false**;

**if**(table.getSelectedRow()==-1) {

msgL.setText("Select a Flight to Book");

}

**else** {

msgL.setText(" ");

}

**if**(!bussinessB.isSelected() && !economyB.isSelected()) {

msg2L.setText("Select your seat class");

}

**if**(bussinessB.isSelected() || economyB.isSelected()) {

buttoncheck=**true**;

msg2L.setText(" ");

}

**if**(table.getSelectedRow()!=-1 && buttoncheck) {

**int** row=table.getSelectedRow();

**int** col=1;//for getting id

String id\_in\_String=table.getValueAt(row, col).toString();

**int** id\_to\_buy=Integer.*parseInt*(id\_in\_String);

International flight\_to\_buy;

**int** buss=0;

**int** econ=0;

**if**(bussinessB.isSelected()) {

buss=1;

}

**if**(economyB.isSelected()) {

econ=1;

}

**for**(**int** i=0;i<Admin.*FM*.interFlights.size();i++) {

**if**(Admin.*FM*.interFlights.get(i).getId()==id\_to\_buy) {

**for**(**int** j=0;j<SignUp.*u*.size();j++) {

**if**(SignUp.*u*.get(j).login\_status==**true**) {

**if**(SignUp.*u*.get(j).isVisa()==**true**) {

flight\_to\_buy=Admin.*FM*.interFlights.get(i);

Admin.*FM*.interFlights.get(i).setSeat\_bussiness(Admin.*FM*.interFlights.get(i).getSeat\_bussiness()-buss);

Admin.*FM*.interFlights.get(i).setSeat\_economy(Admin.*FM*.interFlights.get(i).getSeat\_economy()-econ);

SignUp.*u*.get(j).Book\_Inter(flight\_to\_buy, buss, econ);

SuccessfullyPurchased sp=**new** SuccessfullyPurchased();

sp.setVisible(**true**);

dispose();

}

**else** {

VisaMsg vm=**new** VisaMsg();

vm.setVisible(**true**);

}

}

}

}

}

}

}

});

btnNewButton.setBounds(533, 557, 159, 37);

contentPane.add(btnNewButton);

}

}

**class** BooklocalFlight **extends** JFrame {

**private** JPanel contentPane;

**private** JTable searchTable;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

BooklocalFlight frame = **new** BooklocalFlight();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** BooklocalFlight() {

setTitle("Book Local Flight ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel localdetailsL = **new** JLabel("Book Local Flights");

localdetailsL.setForeground(Color.***BLUE***);

localdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

localdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(localdetailsL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] searched2d =**new** String[SearchLocal.*searchedlocal*.size()][13];

**for**(**int** i=0;i<SearchLocal.*searchedlocal*.size();i++) {

**for**(**int** j=0;j<13;j++) {

String serialno=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).SerialNo) ;

String flightid=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).id) ;

String distance=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).distance);

String Hours=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).time\_Hours);

String Eseats=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).seat\_economy);

String Bseats=String.*valueOf*(SearchLocal.*searchedlocal*.get(i).seat\_bussiness);

String isopen=Boolean.*toString*(SearchLocal.*searchedlocal*.get(i).isOpen);

searched2d [i][0]=serialno;

searched2d [i][1]=flightid;

searched2d [i][2]=SearchLocal.*searchedlocal*.get(i).Airport;

searched2d [i][3]=SearchLocal.*searchedlocal*.get(i).NorthSouth;

searched2d [i][4]=Admin.*FM*.localFlights.get(i).getCity();

searched2d [i][5]=distance;

searched2d [i][6]=Hours;

searched2d [i][7]=SearchLocal.*searchedlocal*.get(i).fdate;

searched2d [i][8]=SearchLocal.*searchedlocal*.get(i).ftime;

searched2d [i][9]=SearchLocal.*searchedlocal*.get(i).arrivalTime;

searched2d [i][10]=Eseats;

searched2d [i][11]=Bseats;

searched2d [i][12]=isopen;

}

}

DefaultTableModel localtable = **new** DefaultTableModel(searched2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

searchTable = **new** JTable(localtable);

scrollPane.setViewportView(searchTable);

searchTable.setFillsViewportHeight(**true**);

searchTable.setSurrendersFocusOnKeystroke(**true**);

searchTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

searchTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JRadioButton bussinessB = **new** JRadioButton("Bussiness");

bussinessB.setBackground(Color.***ORANGE***);

bussinessB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

bussinessB.setForeground(Color.***BLACK***);

bussinessB.setBounds(475, 475, 109, 23);

contentPane.add(bussinessB);

JRadioButton economyB = **new** JRadioButton("Economy");

economyB.setForeground(Color.***BLACK***);

economyB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

economyB.setBackground(Color.***ORANGE***);

economyB.setBounds(661, 475, 109, 23);

contentPane.add(economyB);

ButtonGroup bb=**new** ButtonGroup();

bb.add(economyB);

bb.add(bussinessB);

JLabel msgL = **new** JLabel("");

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 18));

msgL.setForeground(Color.***RED***);

msgL.setBounds(492, 415, 371, 31);

contentPane.add(msgL);

JLabel msg2L = **new** JLabel("");

msg2L.setForeground(Color.***RED***);

msg2L.setFont(**new** Font("Tahoma", Font.***BOLD***, 18));

msg2L.setBounds(492, 505, 371, 31);

contentPane.add(msg2L);

JButton btnNewButton = **new** JButton("Confirm Booking");

btnNewButton.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** buttoncheck=**false**;

**if**(searchTable.getSelectedRow()==-1) {

msgL.setText("Select a Flight to Book");

}

**else** {

msgL.setText(" ");

}

**if**(!bussinessB.isSelected() && !economyB.isSelected()) {

msg2L.setText("Select your seat class");

}

**if**(bussinessB.isSelected() || economyB.isSelected()) {

buttoncheck=**true**;

msg2L.setText(" ");

}

**if**(searchTable.getSelectedRow()!=-1 && buttoncheck) {

**int** row=searchTable.getSelectedRow();

**int** col=1;//for getting id

String id\_in\_String=searchTable.getValueAt(row, col).toString();

**int** id\_to\_buy=Integer.*parseInt*(id\_in\_String);

Local flight\_to\_buy;

**int** buss=0;

**int** econ=0;

**if**(bussinessB.isSelected()) {

buss=1;

}

**if**(economyB.isSelected()) {

econ=1;

}

**for**(**int** i=0;i<Admin.*FM*.localFlights.size();i++) {

**if**(Admin.*FM*.localFlights.get(i).getId()==id\_to\_buy) {

flight\_to\_buy=Admin.*FM*.localFlights.get(i);

Admin.*FM*.localFlights.get(i).setSeat\_bussiness(Admin.*FM*.localFlights.get(i).getSeat\_bussiness()-buss);

Admin.*FM*.localFlights.get(i).setSeat\_economy(Admin.*FM*.localFlights.get(i).getSeat\_economy()-econ);

**for**(**int** j=0;j<SignUp.*u*.size();j++) {

**if**(SignUp.*u*.get(j).login\_status==**true**) {

SignUp.*u*.get(j).Book\_Local(flight\_to\_buy, buss, econ);

SuccessfullyPurchased sp=**new** SuccessfullyPurchased();

sp.setVisible(**true**);

dispose();

}

}

}

}

}

}

});

btnNewButton.setBounds(524, 559, 159, 37);

contentPane.add(btnNewButton);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SearchLocal sl=**new** SearchLocal();

sl.setVisible(**true**);

dispose();

}

});

backB.setBounds(558, 625, 99, 31);

contentPane.add(backB);

}

}

**class** CancelBookedFlight **extends** JFrame {

**private** JPanel contentPane;

**private** JTable LocalTable;

**private** JScrollPane scrollPane;

**private** JTable table;

**private** JScrollPane scrollPane2;

**private** JLabel msgL;

**private** JLabel msg2L;

**private** JButton localCancelB;

**private** JButton interCancelB;

**private** JButton backB;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

CancelBookedFlight frame = **new** CancelBookedFlight();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** CancelBookedFlight() {

setTitle("Cancel Booked Flights Menu");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(434, 11, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Cancel Booked Flights (Local)");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(483, 61, 366, 63);

contentPane.add(lblSuccessfullyLoggedIn);

String [][] Lbooked2d=**null**;

String [][] Ibooked2d=**null**;

String[] heading= {"Booking ID","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Seat Class","Cost","Tax","TotalPaid (pkr)","Booking Status"};

String[] interheading= {"Booking ID","Flight id","Departure","Airport Branch","Destination","inbetween country","Distance(km)","Hours","Date","Departure Time","Arrival Time","Seat Class","Cost","Tax","TotalPaid (pkr)","Booking Status"};

**for**(**int** x=0;x<SignUp.*u*.size();x++) {

**if**(SignUp.*u*.get(x).login\_status) {

Lbooked2d =**new** String[SignUp.*u*.get(x).Localbooked.size()][15];

**for**(**int** i=0;i<SignUp.*u*.get(x).Localbooked.size();i++) {

**for**(**int** j=0;j<15;j++) {

String BookingId=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).booking\_id);

String flightid=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.id);

String distance=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.distance);

String Hours=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.time\_Hours);

String cost=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).cost);

String tax=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).GovtTAX);

String totalPaid=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).ammountToPay);

String seatClass = **null**;

**if**(SignUp.*u*.get(x).Localbooked.get(i).Bseats==1) {

seatClass="Bussiness";

}

**else** **if**(SignUp.*u*.get(x).Localbooked.get(i).Eseats==1) {

seatClass="Economy";

}

String bookingstatus = **null**;

**if**(SignUp.*u*.get(x).Localbooked.get(i).isCancelled==**false**) {

bookingstatus="Booked";

}

**else** **if**(SignUp.*u*.get(x).Localbooked.get(i).isCancelled) {

bookingstatus="Cancelled";

}

Lbooked2d [i][0]=BookingId;

Lbooked2d [i][1]=flightid;

Lbooked2d [i][2]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.Airport;

Lbooked2d [i][3]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.NorthSouth;

Lbooked2d [i][4]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.getCity();

Lbooked2d [i][5]=distance;

Lbooked2d [i][6]=Hours;

Lbooked2d [i][7]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.fdate;

Lbooked2d [i][8]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.ftime;

Lbooked2d [i][9]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.arrivalTime;

Lbooked2d [i][10]=seatClass;

Lbooked2d [i][11]=cost;

Lbooked2d [i][12]=tax;

Lbooked2d [i][13]=totalPaid;

Lbooked2d [i][14]=bookingstatus;

}

}

}

}

DefaultTableModel localtable = **new** DefaultTableModel(Lbooked2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

**for**(**int** x=0;x<SignUp.*u*.size();x++) {

**if**(SignUp.*u*.get(x).login\_status) {

Ibooked2d=**new** String[SignUp.*u*.get(x).Interbooked.size()][16];

**for**(**int** i=0;i<SignUp.*u*.get(x).Interbooked.size();i++) {

**for**(**int** j=0;j<15;j++) {

String BookingId=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).booking\_id);

String flightid=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.id);

String distance=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.distance);

String Hours=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.time\_Hours);

String cost=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).cost);

String tax=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).GovtTAX);

String totalPaid=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).ammountToPay);

String seatClass = **null**;

**if**(SignUp.*u*.get(x).Interbooked.get(i).Bseats==1) {

seatClass="Bussiness";

}

**else** **if**(SignUp.*u*.get(x).Interbooked.get(i).Eseats==1) {

seatClass="Economy";

}

String bookingstatus = **null**;

**if**(SignUp.*u*.get(x).Interbooked.get(i).isCancelled==**false**) {

bookingstatus="Booked";

}

**else** **if**(SignUp.*u*.get(x).Interbooked.get(i).isCancelled) {

bookingstatus="Cancelled";

}

Ibooked2d [i][0]=BookingId;

Ibooked2d [i][1]=flightid;

Ibooked2d [i][2]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.Airport;

Ibooked2d [i][3]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.NorthSouth;

Ibooked2d [i][4]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.getCountry();

Ibooked2d [i][5]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.getIndirectRouteCountry();

Ibooked2d [i][6]=distance;

Ibooked2d [i][7]=Hours;

Ibooked2d [i][8]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.fdate;

Ibooked2d [i][9]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.ftime;

Ibooked2d [i][10]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.arrivalTime;

Ibooked2d [i][11]=seatClass;

Ibooked2d [i][12]=cost;

Ibooked2d [i][13]=tax;

Ibooked2d [i][14]=totalPaid;

Ibooked2d [i][15]=bookingstatus;

}

}

}

}

DefaultTableModel intertable = **new** DefaultTableModel(Ibooked2d,interheading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 114, 1330, 142);

contentPane.add(scrollPane);

LocalTable = **new** JTable(localtable);

scrollPane.setViewportView(LocalTable);

LocalTable.setFillsViewportHeight(**true**);

LocalTable.setSurrendersFocusOnKeystroke(**true**);

LocalTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

LocalTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JLabel lblBookedFlightsinternational = **new** JLabel("Booked Flights (International)");

lblBookedFlightsinternational.setForeground(Color.***BLUE***);

lblBookedFlightsinternational.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblBookedFlightsinternational.setBounds(434, 347, 382, 45);

contentPane.add(lblBookedFlightsinternational);

scrollPane2 = **new** JScrollPane();

scrollPane2.setBounds(10, 392, 1328, 157);

contentPane.add(scrollPane2);

table = **new** JTable(intertable);

scrollPane2.setViewportView(table);

table.setSurrendersFocusOnKeystroke(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setFillsViewportHeight(**true**);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

msgL.setBounds(328, 267, 413, 29);

contentPane.add(msgL);

msg2L = **new** JLabel("");

msg2L.setForeground(Color.***RED***);

msg2L.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

msg2L.setBounds(367, 560, 413, 29);

contentPane.add(msg2L);

localCancelB = **new** JButton("Confirm Cancel");

localCancelB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(LocalTable.getSelectedRow()==-1) {

msgL.setText("Firstly Select a booking to cancel");

}

**else** **if**(LocalTable.getSelectedRow()!=-1) {

**int** row=LocalTable.getSelectedRow();

**int** col=0;//for getting booking id

String id\_in\_String=LocalTable.getValueAt(row, col).toString();

**int** id\_to\_Cancel=Integer.*parseInt*(id\_in\_String);

**for**(**int** j=0;j<SignUp.*u*.size();j++) {

**if**(SignUp.*u*.get(j).login\_status==**true**) {

**if**(SignUp.*u*.get(j).cancelLocal(id\_to\_Cancel)) {

CancelRefundBox crb=**new** CancelRefundBox();

crb.setVisible(**true**);

dispose();

}

}

**else** {

msgL.setText("Already Cancelled");

}

}

}

}

});

localCancelB.setBounds(482, 307, 113, 29);

contentPane.add(localCancelB);

interCancelB = **new** JButton("Confirm Cancel");

interCancelB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**if**(table.getSelectedRow()==-1) {

msg2L.setText("Firstly Select a booking to cancel");

}

**else** **if**(table.getSelectedRow()!=-1) {

**int** row=table.getSelectedRow();

**int** col=0;//for getting booking id

String id\_in\_String=table.getValueAt(row, col).toString();

**int** id\_to\_Cancel=Integer.*parseInt*(id\_in\_String);

**for**(**int** j=0;j<SignUp.*u*.size();j++) {

**if**(SignUp.*u*.get(j).login\_status==**true**) {

**if**(SignUp.*u*.get(j).cancelinter(id\_to\_Cancel)) {

CancelRefundBox crb=**new** CancelRefundBox();

crb.setVisible(**true**);

dispose();

}

**else** {

msg2L.setText("Already Cancelled");

}

}

}

}

}

});

interCancelB.setBounds(483, 600, 113, 29);

contentPane.add(interCancelB);

backB = **new** JButton("Go Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

backB.setBounds(483, 666, 113, 29);

contentPane.add(backB);

}

}

**class** CancelRefundBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

CancelRefundBox frame = **new** CancelRefundBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** CancelRefundBox() {

setTitle("Flight Cancelled");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 597, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** MatteBorder(1, 1, 1, 1, (Color) **new** Color(0, 0, 0)));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight has been cancelled and ammount refunded");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(26, 34, 523, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

CancelBookedFlight cbl=**new** CancelBookedFlight();

cbl.setVisible(**true**);

dispose();

}

});

okB.setBounds(249, 193, 89, 23);

contentPane.add(okB);

JLabel lblYouCannotTravell = **new** JLabel("with 25% deduction you can check your balance");

lblYouCannotTravell.setForeground(Color.***BLUE***);

lblYouCannotTravell.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblYouCannotTravell.setBounds(36, 97, 523, 52);

contentPane.add(lblYouCannotTravell);

}

}

**class** CurrentBookedFlights **extends** JFrame {

**private** JPanel contentPane;

**private** JTable LocalTable;

**private** JScrollPane scrollPane;

**private** JTable table;

**private** JScrollPane scrollPane2;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

CurrentBookedFlights frame = **new** CurrentBookedFlights();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** CurrentBookedFlights() {

setTitle("Booked Flights");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Booked Flights (Local)");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(524, 88, 262, 63);

contentPane.add(lblSuccessfullyLoggedIn);

String [][] Lbooked2d=**null**;

String [][] Ibooked2d=**null**;

String[] heading= {"Booking ID","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Seat Class","Cost","Tax","TotalPaid (pkr)","Booking Status"};

String[] interheading= {"Booking ID","Flight id","Departure","Airport Branch","Destination","inbetween country","Distance(km)","Hours","Date","Departure Time","Arrival Time","Seat Class","Cost","Tax","TotalPaid (pkr)","Booking Status"};

**for**(**int** x=0;x<SignUp.*u*.size();x++) {

**if**(SignUp.*u*.get(x).login\_status) {

Lbooked2d =**new** String[SignUp.*u*.get(x).Localbooked.size()][15];

**for**(**int** i=0;i<SignUp.*u*.get(x).Localbooked.size();i++) {

**for**(**int** j=0;j<15;j++) {

String BookingId=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).booking\_id);

String flightid=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.id);

String distance=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.distance);

String Hours=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).bookflight.time\_Hours);

String cost=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).cost);

String tax=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).GovtTAX);

String totalPaid=String.*valueOf*(SignUp.*u*.get(x).Localbooked.get(i).ammountToPay);

String seatClass = **null**;

**if**(SignUp.*u*.get(x).Localbooked.get(i).Bseats==1) {

seatClass="Bussiness";

}

**else** **if**(SignUp.*u*.get(x).Localbooked.get(i).Eseats==1) {

seatClass="Economy";

}

String bookingstatus = **null**;

**if**(SignUp.*u*.get(x).Localbooked.get(i).isCancelled==**false**) {

bookingstatus="Booked";

}

**else** **if**(SignUp.*u*.get(x).Localbooked.get(i).isCancelled) {

bookingstatus="Cancelled";

}

Lbooked2d [i][0]=BookingId;

Lbooked2d [i][1]=flightid;

Lbooked2d [i][2]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.Airport;

Lbooked2d [i][3]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.NorthSouth;

Lbooked2d [i][4]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.getCity();

Lbooked2d [i][5]=distance;

Lbooked2d [i][6]=Hours;

Lbooked2d [i][7]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.fdate;

Lbooked2d [i][8]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.ftime;

Lbooked2d [i][9]=SignUp.*u*.get(x).Localbooked.get(i).bookflight.arrivalTime;

Lbooked2d [i][10]=seatClass;

Lbooked2d [i][11]=cost;

Lbooked2d [i][12]=tax;

Lbooked2d [i][13]=totalPaid;

Lbooked2d [i][14]=bookingstatus;

}

}

}

}

DefaultTableModel localtable = **new** DefaultTableModel(Lbooked2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

**for**(**int** x=0;x<SignUp.*u*.size();x++) {

**if**(SignUp.*u*.get(x).login\_status) {

Ibooked2d=**new** String[SignUp.*u*.get(x).Interbooked.size()][16];

**for**(**int** i=0;i<SignUp.*u*.get(x).Interbooked.size();i++) {

**for**(**int** j=0;j<15;j++) {

String BookingId=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).booking\_id);

String flightid=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.id);

String distance=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.distance);

String Hours=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).bookflight.time\_Hours);

String cost=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).cost);

String tax=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).GovtTAX);

String totalPaid=String.*valueOf*(SignUp.*u*.get(x).Interbooked.get(i).ammountToPay);

String seatClass = **null**;

**if**(SignUp.*u*.get(x).Interbooked.get(i).Bseats==1) {

seatClass="Bussiness";

}

**else** **if**(SignUp.*u*.get(x).Interbooked.get(i).Eseats==1) {

seatClass="Economy";

}

String bookingstatus = **null**;

**if**(SignUp.*u*.get(x).Interbooked.get(i).isCancelled==**false**) {

bookingstatus="Booked";

}

**else** **if**(SignUp.*u*.get(x).Interbooked.get(i).isCancelled) {

bookingstatus="Cancelled";

}

Ibooked2d [i][0]=BookingId;

Ibooked2d [i][1]=flightid;

Ibooked2d [i][2]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.Airport;

Ibooked2d [i][3]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.NorthSouth;

Ibooked2d [i][4]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.getCountry();

Ibooked2d [i][5]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.getIndirectRouteCountry();

Ibooked2d [i][6]=distance;

Ibooked2d [i][7]=Hours;

Ibooked2d [i][8]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.fdate;

Ibooked2d [i][9]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.ftime;

Ibooked2d [i][10]=SignUp.*u*.get(x).Interbooked.get(i).bookflight.arrivalTime;

Ibooked2d [i][11]=seatClass;

Ibooked2d [i][12]=cost;

Ibooked2d [i][13]=tax;

Ibooked2d [i][14]=totalPaid;

Ibooked2d [i][15]=bookingstatus;

}

}

}

}

DefaultTableModel intertable = **new** DefaultTableModel(Ibooked2d,interheading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

LocalTable = **new** JTable(localtable);

scrollPane.setViewportView(LocalTable);

LocalTable.setFillsViewportHeight(**true**);

LocalTable.setSurrendersFocusOnKeystroke(**true**);

LocalTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

LocalTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JLabel lblBookedFlightsinternational = **new** JLabel("Booked Flights (International)");

lblBookedFlightsinternational.setForeground(Color.***BLUE***);

lblBookedFlightsinternational.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblBookedFlightsinternational.setBounds(484, 391, 382, 45);

contentPane.add(lblBookedFlightsinternational);

scrollPane2 = **new** JScrollPane();

scrollPane2.setBounds(10, 460, 1328, 191);

contentPane.add(scrollPane2);

table = **new** JTable(intertable);

scrollPane2.setViewportView(table);

table.setSurrendersFocusOnKeystroke(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setFillsViewportHeight(**true**);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

backB.setBounds(577, 662, 107, 34);

contentPane.add(backB);

}

}

**class** DeleteBlockInter **extends** JFrame {

**private** JPanel contentPane;

**private** JTable table;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

DeleteBlockInter frame = **new** DeleteBlockInter();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** DeleteBlockInter() {

setTitle("Delete/Block International Flight ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel interdeleteL = **new** JLabel("Delete/Block International Flight");

interdeleteL.setForeground(Color.***BLUE***);

interdeleteL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

interdeleteL.setBounds(493, 67, 326, 46);

contentPane.add(interdeleteL);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

msgL.setBounds(444, 413, 417, 30);

contentPane.add(msgL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Indirect","in Between Country","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] inter2d =**new** String[Admin.*FM*.interFlights.size()][15];

**for**(**int** i=0;i<Admin.*FM*.interFlights.size();i++) {

**for**(**int** j=0;j<15;j++) {

String serialno=String.*valueOf*(Admin.*FM*.interFlights.get(i).SerialNo);

String flightid=String.*valueOf*(Admin.*FM*.interFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.interFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.interFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_bussiness);

String indirectbool=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isIndierct());

String isopen=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isOpen);

inter2d[i][0]=serialno;

inter2d[i][1]=flightid;

inter2d[i][2]=Admin.*FM*.interFlights.get(i).Airport;

inter2d[i][3]=Admin.*FM*.interFlights.get(i).NorthSouth;

inter2d[i][4]=indirectbool;

inter2d[i][5]=Admin.*FM*.interFlights.get(i).getIndirectRouteCountry();

inter2d[i][6]=Admin.*FM*.interFlights.get(i).getCountry();

inter2d[i][7]=distance;

inter2d[i][8]=Hours;

inter2d[i][9]=Admin.*FM*.interFlights.get(i).fdate;

inter2d[i][10]=Admin.*FM*.interFlights.get(i).ftime;

inter2d[i][11]=Admin.*FM*.interFlights.get(i).arrivalTime;

inter2d[i][12]=Eseats;

inter2d[i][13]=Bseats;

inter2d[i][14]=isopen;

}

}

DefaultTableModel intertable = **new** DefaultTableModel(inter2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 155, 1350, 218);

contentPane.add(scrollPane);

table = **new** JTable(intertable);

scrollPane.setViewportView(table);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

table.setFillsViewportHeight(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setSurrendersFocusOnKeystroke(**true**);

JButton deleteB = **new** JButton("Delete International Flight");

deleteB.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

}

});

deleteB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(table.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to delete");

}

**else** {

String srno\_inString=table.getValueAt(table.getSelectedRow(), 0).toString();

String id\_inString=table.getValueAt(table.getSelectedRow(),1).toString();

**int** serial=Integer.*parseInt*(srno\_inString);

**int** idtodelete=Integer.*parseInt*(id\_inString);

al.*admin*.removeinternational(serial,idtodelete);

interdeletebox idb=**new** interdeletebox ();

idb.setVisible(**true**);

dispose();

}

}

});

deleteB.setBounds(225, 524, 222, 38);

contentPane.add(deleteB);

JButton blockB = **new** JButton("Block International Flight");

blockB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(table.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to Block");

}

**else** {

String id\_inString=table.getValueAt(table.getSelectedRow(),1).toString();

**int** idtoBlock=Integer.*parseInt*(id\_inString);

al.*admin*.blockinterflight(idtoBlock);;

interblockbox ibb=**new** interblockbox();

ibb.setVisible(**true**);

dispose();

}

}

});

blockB.setBounds(549, 524, 212, 38);

contentPane.add(blockB);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess ();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(578, 619, 161, 38);

contentPane.add(backB);

JButton openB = **new** JButton("Open International Flight");

openB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(table.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to Block");

}

**else** {

String id\_inString=table.getValueAt(table.getSelectedRow(),1).toString();

**int** idtoopen=Integer.*parseInt*(id\_inString);

al.*admin*.openinterflight(idtoopen);;

interopenbox ito=**new** interopenbox();

ito.setVisible(**true**);

dispose();

}

}

});

openB.setBounds(879, 524, 212, 38);

contentPane.add(openB);

}

}

**class** DeleteBlockLocal **extends** JFrame {

**private** JPanel contentPane;

**private** JTable LocalTable;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

DeleteBlockLocal frame = **new** DeleteBlockLocal();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** DeleteBlockLocal() {

setTitle("Delete/Block Local Flight");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel localdelL = **new** JLabel("Local Flight Delete / Block Menu");

localdelL.setForeground(Color.***BLUE***);

localdelL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

localdelL.setBounds(449, 73, 393, 46);

contentPane.add(localdelL);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

msgL.setBounds(449, 419, 412, 24);

contentPane.add(msgL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] local2d =**new** String[Admin.*FM*.localFlights.size()][13];

**for**(**int** i=0;i<Admin.*FM*.localFlights.size();i++) {

**for**(**int** j=0;j<13;j++) {

String serialno=String.*valueOf*(Admin.*FM*.localFlights.get(i).SerialNo) ;

String flightid=String.*valueOf*(Admin.*FM*.localFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.localFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.localFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_bussiness);

String isopen=Boolean.*toString*(Admin.*FM*.localFlights.get(i).isOpen);

local2d [i][0]=serialno;

local2d [i][1]=flightid;

local2d [i][2]=Admin.*FM*.localFlights.get(i).Airport;

local2d [i][3]=Admin.*FM*.localFlights.get(i).NorthSouth;

local2d [i][4]=Admin.*FM*.localFlights.get(i).getCity();

local2d [i][5]=distance;

local2d [i][6]=Hours;

local2d [i][7]=Admin.*FM*.localFlights.get(i).fdate;

local2d [i][8]=Admin.*FM*.localFlights.get(i).ftime;

local2d [i][9]=Admin.*FM*.localFlights.get(i).arrivalTime;

local2d [i][10]=Eseats;

local2d [i][11]=Bseats;

local2d [i][12]=isopen;

}

}

DefaultTableModel localtable = **new** DefaultTableModel(local2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

LocalTable = **new** JTable(localtable);

scrollPane.setViewportView(LocalTable);

LocalTable.setFillsViewportHeight(**true**);

LocalTable.setSurrendersFocusOnKeystroke(**true**);

LocalTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

LocalTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JButton deleteB = **new** JButton("Delete Local Flight");

deleteB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(LocalTable.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to delete");

}

**else** {

String srno\_inString=LocalTable.getValueAt(LocalTable.getSelectedRow(), 0).toString();

String id\_inString=LocalTable.getValueAt(LocalTable.getSelectedRow(),1).toString();

**int** serial=Integer.*parseInt*(srno\_inString);

**int** idtodelete=Integer.*parseInt*(id\_inString);

al.*admin*.removelocal(serial,idtodelete);

flightdeletedBox fdb=**new** flightdeletedBox();

fdb.setVisible(**true**);

dispose();

}

}

});

deleteB.setBounds(264, 520, 166, 38);

contentPane.add(deleteB);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLoginSuccess als=**new** AdminLoginSuccess ();

als.setVisible(**true**);

dispose();

}

});

backB.setBounds(562, 614, 117, 38);

contentPane.add(backB);

JButton blockB = **new** JButton("Block Local Flight");

blockB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(LocalTable.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to Block");

}

**else** {

String id\_inString=LocalTable.getValueAt(LocalTable.getSelectedRow(),1).toString();

**int** idtoBlock=Integer.*parseInt*(id\_inString);

al.*admin*.blocklocalflight(idtoBlock);;

flightblockedbox fbb=**new** flightblockedbox();

fbb.setVisible(**true**);

dispose();

}

}

});

blockB.setBounds(538, 520, 166, 38);

contentPane.add(blockB);

JButton openB = **new** JButton("Open Local Flight");

openB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AdminLogin al=**new** AdminLogin();

**if**(LocalTable.getSelectedRow()==-1) {

msgL.setText("Please Select a flight to Block");

}

**else** {

String id\_inString=LocalTable.getValueAt(LocalTable.getSelectedRow(),1).toString();

**int** idtoopen=Integer.*parseInt*(id\_inString);

al.*admin*.openlocalflight(idtoopen);;

localopenbox lob=**new** localopenbox();

lob.setVisible(**true**);

dispose();

}

}

});

openB.setBounds(815, 520, 166, 38);

contentPane.add(openB);

}

}

**class** DetailsUpdatedBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

DetailsUpdatedBox frame = **new** DetailsUpdatedBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** DetailsUpdatedBox() {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 521, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Account Details Updated Successfully");

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setBounds(56, 75, 425, 49);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OK");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

dispose();

}

});

okB.setBounds(170, 175, 128, 31);

contentPane.add(okB);

}

}

**class** flightblockedbox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

flightblockedbox frame = **new** flightblockedbox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** flightblockedbox() {

setTitle("Flight Blocked Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Blocked Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockLocal dbl=**new** DeleteBlockLocal();

dbl.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** flightdeletedBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

flightdeletedBox frame = **new** flightdeletedBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** flightdeletedBox() {

setTitle("Flight Deleted Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Deleted Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockLocal dbl=**new** DeleteBlockLocal();

dbl.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** FlightDetails **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

FlightDetails frame = **new** FlightDetails();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** FlightDetails() {

setTitle("Flight Schedule Details ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(384, 0, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Flight Details");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(521, 64, 326, 46);

contentPane.add(lblSuccessfullyLoggedIn);

JLabel infoL = **new** JLabel("You need to login to Book Flight");

infoL.setForeground(Color.***RED***);

infoL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

infoL.setBounds(425, 121, 492, 46);

contentPane.add(infoL);

JButton localB = **new** JButton("Local Flight Details");

localB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

LocalDetailsTable ldt=**new** LocalDetailsTable();

ldt.setVisible(**true**);

dispose();

}

});

localB.setBounds(493, 199, 188, 37);

contentPane.add(localB);

JButton interB = **new** JButton("International Flight Details");

interB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

InterDetailsTable idt =**new** InterDetailsTable();

idt.setVisible(**true**);

dispose();

}

});

interB.setBounds(493, 335, 188, 37);

contentPane.add(interB);

JButton backB = **new** JButton("Back");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

backB.setBounds(493, 466, 188, 37);

contentPane.add(backB);

}

}

**class** InterAdded **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

InterAdded frame = **new** InterAdded();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** InterAdded() {

setTitle("Successfully Added");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JButton addB = **new** JButton("Add another");

addB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

addInter ai=**new** addInter();

ai.setVisible(**true**);

}

});

addB.setBounds(249, 192, 137, 23);

contentPane.add(addB);

JButton MainB = **new** JButton("Main Menu");

MainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainB.setBounds(60, 192, 130, 23);

contentPane.add(MainB);

JLabel TittleL = **new** JLabel("International Flight Added Successfully");

TittleL.setForeground(Color.***BLUE***);

TittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 18));

TittleL.setBounds(33, 81, 391, 34);

contentPane.add(TittleL);

}

}

**class** interblockbox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

interblockbox frame = **new** interblockbox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** interblockbox() {

setTitle("Flight Blocked Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Blocked Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockInter dbi=**new** DeleteBlockInter();

dbi.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** interdeletebox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

interdeletebox frame = **new** interdeletebox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** interdeletebox() {

setTitle("Flight Deleted Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Deleted Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockInter dbi=**new** DeleteBlockInter();

dbi.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** InterDetailsTable **extends** JFrame {

**private** JPanel contentPane;

**private** JTable table;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

InterDetailsTable frame = **new** InterDetailsTable();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** InterDetailsTable() {

setTitle("International Flight Schedule Details ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel interdetailsL = **new** JLabel("International Flight Details");

interdetailsL.setForeground(Color.***BLUE***);

interdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

interdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(interdetailsL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Indirect","in Between Country","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] inter2d =**new** String[Admin.*FM*.interFlights.size()][15];

**for**(**int** i=0;i<Admin.*FM*.interFlights.size();i++) {

**for**(**int** j=0;j<15;j++) {

String serialno=String.*valueOf*(Admin.*FM*.interFlights.get(i).SerialNo);

String flightid=String.*valueOf*(Admin.*FM*.interFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.interFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.interFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_bussiness);

String indirectbool=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isIndierct());

String isopen=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isOpen);

inter2d[i][0]=serialno;

inter2d[i][1]=flightid;

inter2d[i][2]=Admin.*FM*.interFlights.get(i).Airport;

inter2d[i][3]=Admin.*FM*.interFlights.get(i).NorthSouth;

inter2d[i][4]=indirectbool;

inter2d[i][5]=Admin.*FM*.interFlights.get(i).getIndirectRouteCountry();

inter2d[i][6]=Admin.*FM*.interFlights.get(i).getCountry();

inter2d[i][7]=distance;

inter2d[i][8]=Hours;

inter2d[i][9]=Admin.*FM*.interFlights.get(i).fdate;

inter2d[i][10]=Admin.*FM*.interFlights.get(i).ftime;

inter2d[i][11]=Admin.*FM*.interFlights.get(i).arrivalTime;

inter2d[i][12]=Eseats;

inter2d[i][13]=Bseats;

inter2d[i][14]=isopen;

}

}

DefaultTableModel intertable = **new** DefaultTableModel(inter2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 155, 1350, 218);

contentPane.add(scrollPane);

table = **new** JTable(intertable);

scrollPane.setViewportView(table);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

table.setFillsViewportHeight(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setSurrendersFocusOnKeystroke(**true**);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

FlightDetails fd=**new** FlightDetails();

fd.setVisible(**true**);

dispose();

}

});

backB.setBounds(554, 568, 117, 38);

contentPane.add(backB);

}

}

**class** interopenbox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

interopenbox frame = **new** interopenbox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** interopenbox() {

setTitle("Flight Opened Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Opened Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockInter dbi=**new** DeleteBlockInter();

dbi.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** LocalAdded **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

LocalAdded frame = **new** LocalAdded();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** LocalAdded() {

setTitle("Successfully Added");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JButton addB = **new** JButton("Add another");

addB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AddLocal al=**new** AddLocal();

al.setVisible(**true**);

}

});

addB.setBounds(249, 192, 137, 23);

contentPane.add(addB);

JButton MainB = **new** JButton("Main Menu");

MainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainB.setBounds(60, 192, 130, 23);

contentPane.add(MainB);

JLabel TittleL = **new** JLabel("Local Flight Added Successfully");

TittleL.setForeground(Color.***BLUE***);

TittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 18));

TittleL.setBounds(75, 77, 326, 34);

contentPane.add(TittleL);

}

}

**class** LocalDetailsTable **extends** JFrame {

**private** JPanel contentPane;

**private** JTable LocalTable;

**private** JScrollPane scrollPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

LocalDetailsTable frame = **new** LocalDetailsTable();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** LocalDetailsTable() {

setTitle("Local Flight Schedule Details ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel localdetailsL = **new** JLabel("Local Flight Details");

localdetailsL.setForeground(Color.***BLUE***);

localdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

localdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(localdetailsL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] local2d =**new** String[Admin.*FM*.localFlights.size()][13];

**for**(**int** i=0;i<Admin.*FM*.localFlights.size();i++) {

**for**(**int** j=0;j<13;j++) {

String serialno=String.*valueOf*(Admin.*FM*.localFlights.get(i).SerialNo) ;

String flightid=String.*valueOf*(Admin.*FM*.localFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.localFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.localFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_bussiness);

String isopen=Boolean.*toString*(Admin.*FM*.localFlights.get(i).isOpen);

local2d [i][0]=serialno;

local2d [i][1]=flightid;

local2d [i][2]=Admin.*FM*.localFlights.get(i).Airport;

local2d [i][3]=Admin.*FM*.localFlights.get(i).NorthSouth;

local2d [i][4]=Admin.*FM*.localFlights.get(i).getCity();

local2d [i][5]=distance;

local2d [i][6]=Hours;

local2d [i][7]=Admin.*FM*.localFlights.get(i).fdate;

local2d [i][8]=Admin.*FM*.localFlights.get(i).ftime;

local2d [i][9]=Admin.*FM*.localFlights.get(i).arrivalTime;

local2d [i][10]=Eseats;

local2d [i][11]=Bseats;

local2d [i][12]=isopen;

}

}

DefaultTableModel localtable = **new** DefaultTableModel(local2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

LocalTable = **new** JTable(localtable);

scrollPane.setViewportView(LocalTable);

LocalTable.setFillsViewportHeight(**true**);

LocalTable.setSurrendersFocusOnKeystroke(**true**);

LocalTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

LocalTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

FlightDetails fd=**new** FlightDetails();

fd.setVisible(**true**);

dispose();

}

});

backB.setBounds(554, 568, 117, 38);

contentPane.add(backB);

}

}

**class** localopenbox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

localopenbox frame = **new** localopenbox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** localopenbox() {

setTitle("Flight Opened Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Opened Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(73, 79, 285, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

DeleteBlockLocal dbl=**new** DeleteBlockLocal();

dbl.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** LogoutBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

LogoutBox frame = **new** LogoutBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** LogoutBox() {

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JButton HomeB = **new** JButton("HOME");

HomeB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

HomeB.setBounds(139, 167, 129, 30);

contentPane.add(HomeB);

JLabel lblNewLabel = **new** JLabel("Account Logged Out");

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 17));

lblNewLabel.setBounds(115, 71, 204, 37);

contentPane.add(lblNewLabel);

}

}

**class** PassportInstruction **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

PassportInstruction frame = **new** PassportInstruction();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** PassportInstruction() {

setTitle("Passport Help");

setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel balL = **new** JLabel("Passport and Visa Information");

balL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

balL.setForeground(Color.***BLUE***);

balL.setBounds(63, 11, 324, 52);

contentPane.add(balL);

JLabel step1 = **new** JLabel("-->Enter 9 digit Passport Number.");

step1.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

step1.setBounds(41, 89, 324, 23);

contentPane.add(step1);

JLabel step2 = **new** JLabel("-->It will be verified by admin.");

step2.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

step2.setBounds(41, 124, 324, 23);

contentPane.add(step2);

JLabel step3 = **new** JLabel("-->if valid then your visa will be declared verified.");

step3.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

step3.setBounds(41, 159, 324, 23);

contentPane.add(step3);

JLabel step4 = **new** JLabel("-->Then you can enjoy international flights.");

step4.setFont(**new** Font("Tahoma", Font.***BOLD***, 12));

step4.setBounds(41, 193, 324, 23);

contentPane.add(step4);

}

}

**class** SearchInternational **extends** JFrame {

**private** JPanel contentPane;

**private** JTable table;

**private** JScrollPane scrollPane;

**public** **static** ArrayList<International> *searchedinter* =**new** ArrayList<>();

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

SearchInternational frame = **new** SearchInternational();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** SearchInternational() {

setTitle("Search International Flight");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel interdetailsL = **new** JLabel("Search International Flights");

interdetailsL.setForeground(Color.***BLUE***);

interdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

interdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(interdetailsL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Indirect","in Between Country","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] inter2d =**new** String[Admin.*FM*.interFlights.size()][15];

**for**(**int** i=0;i<Admin.*FM*.interFlights.size();i++) {

**for**(**int** j=0;j<15;j++) {

String serialno=String.*valueOf*(Admin.*FM*.interFlights.get(i).SerialNo);

String flightid=String.*valueOf*(Admin.*FM*.interFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.interFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.interFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.interFlights.get(i).seat\_bussiness);

String indirectbool=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isIndierct());

String isopen=Boolean.*toString*(Admin.*FM*.interFlights.get(i).isOpen);

inter2d[i][0]=serialno;

inter2d[i][1]=flightid;

inter2d[i][2]=Admin.*FM*.interFlights.get(i).Airport;

inter2d[i][3]=Admin.*FM*.interFlights.get(i).NorthSouth;

inter2d[i][4]=indirectbool;

inter2d[i][5]=Admin.*FM*.interFlights.get(i).getIndirectRouteCountry();

inter2d[i][6]=Admin.*FM*.interFlights.get(i).getCountry();

inter2d[i][7]=distance;

inter2d[i][8]=Hours;

inter2d[i][9]=Admin.*FM*.interFlights.get(i).fdate;

inter2d[i][10]=Admin.*FM*.interFlights.get(i).ftime;

inter2d[i][11]=Admin.*FM*.interFlights.get(i).arrivalTime;

inter2d[i][12]=Eseats;

inter2d[i][13]=Bseats;

inter2d[i][14]=isopen;

}

}

DefaultTableModel intertable = **new** DefaultTableModel(inter2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 155, 1350, 218);

contentPane.add(scrollPane);

table = **new** JTable(intertable);

scrollPane.setViewportView(table);

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

table.setFillsViewportHeight(**true**);

table.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

table.setSurrendersFocusOnKeystroke(**true**);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

backB.setBounds(554, 647, 117, 38);

contentPane.add(backB);

JComboBox fromCB = **new** JComboBox();

fromCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Islamabad", "Karachi", "Peshawar", "Quetta", "Lahore"}));

fromCB.setBounds(389, 460, 161, 22);

contentPane.add(fromCB);

JLabel lblNewLabel = **new** JLabel("From City:");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblNewLabel.setBounds(389, 419, 161, 22);

contentPane.add(lblNewLabel);

JComboBox toCB = **new** JComboBox();

toCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "India", "China", "USA", "Turkey", "UAE", "Nepal", "Oman", "Syria", "Afghanistan", "Iran", "UK", "Brazil", "Egypt", "Hongkong", "Ireland", "Malayisa", "Yemen"}));

toCB.setBounds(655, 460, 161, 22);

contentPane.add(toCB);

JLabel lblToCountry = **new** JLabel("To Country:");

lblToCountry.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblToCountry.setBounds(655, 419, 161, 22);

contentPane.add(lblToCountry);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

msgL.setBounds(505, 516, 268, 22);

contentPane.add(msgL);

JButton searchB = **new** JButton("Search");

searchB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

String fromcity=fromCB.getSelectedItem().toString();

String toCountry=toCB.getSelectedItem().toString();

**if**(fromcity.equals("-") || toCountry.equals("-")) {

msgL.setText("First select cities and country to search");

}

**else** {

**for**(**int** i=0;i<Admin.*FM*.interFlights.size();i++) {

**if**(Admin.*FM*.interFlights.get(i).Airport.equals(fromcity) && Admin.*FM*.interFlights.get(i).getCountry().equals(toCountry)) {

*searchedinter*.add(Admin.*FM*.interFlights.get(i));

BookInterFlight bif=**new** BookInterFlight();

bif.setVisible(**true**);

dispose();

}

**else** {

msgL.setText("No such Flight Available");

}

}

}

}

});

searchB.setBounds(523, 574, 178, 38);

contentPane.add(searchB);

}

}

**class** SearchLocal **extends** JFrame {

**private** JPanel contentPane;

**private** JTable LocalTable;

**private** JScrollPane scrollPane;

**public** **static** ArrayList<Local> *searchedlocal* =**new** ArrayList<>();

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

SearchLocal frame = **new** SearchLocal();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** SearchLocal() {

setTitle("Search Local Flight ");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(417, 0, 492, 63);

contentPane.add(tittleL);

JLabel localdetailsL = **new** JLabel("Search Local Flights");

localdetailsL.setForeground(Color.***BLUE***);

localdetailsL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

localdetailsL.setBounds(493, 67, 326, 46);

contentPane.add(localdetailsL);

JComboBox fromCB = **new** JComboBox();

fromCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Islamabad", "Karachi", "Peshawar", "Quetta", "Lahore"}));

fromCB.setBounds(376, 453, 161, 22);

contentPane.add(fromCB);

JComboBox toCB = **new** JComboBox();

toCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "Islamabad", "Karachi", "Peshawar", "Quetta", "Lahore"}));

toCB.setBounds(658, 453, 161, 22);

contentPane.add(toCB);

JLabel lblNewLabel = **new** JLabel("From City:");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblNewLabel.setBounds(376, 420, 161, 22);

contentPane.add(lblNewLabel);

JLabel lblToCity = **new** JLabel("To City:");

lblToCity.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

lblToCity.setBounds(658, 420, 161, 22);

contentPane.add(lblToCity);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setFont(**new** Font("Tahoma", Font.***BOLD***, 14));

msgL.setBounds(493, 510, 268, 22);

contentPane.add(msgL);

String[] heading= {"Serial#","Flight id","Departure","Airport Branch","Destination","Distance(km)","Hours","Date","Departure Time","Arrival Time","Economy Available","Business Available","Open Status"};

String [][] local2d =**new** String[Admin.*FM*.localFlights.size()][13];

**for**(**int** i=0;i<Admin.*FM*.localFlights.size();i++) {

**for**(**int** j=0;j<13;j++) {

String serialno=String.*valueOf*(Admin.*FM*.localFlights.get(i).SerialNo) ;

String flightid=String.*valueOf*(Admin.*FM*.localFlights.get(i).id) ;

String distance=String.*valueOf*(Admin.*FM*.localFlights.get(i).distance);

String Hours=String.*valueOf*(Admin.*FM*.localFlights.get(i).time\_Hours);

String Eseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_economy);

String Bseats=String.*valueOf*(Admin.*FM*.localFlights.get(i).seat\_bussiness);

String isopen=Boolean.*toString*(Admin.*FM*.localFlights.get(i).isOpen);

local2d [i][0]=serialno;

local2d [i][1]=flightid;

local2d [i][2]=Admin.*FM*.localFlights.get(i).Airport;

local2d [i][3]=Admin.*FM*.localFlights.get(i).NorthSouth;

local2d [i][4]=Admin.*FM*.localFlights.get(i).getCity();

local2d [i][5]=distance;

local2d [i][6]=Hours;

local2d [i][7]=Admin.*FM*.localFlights.get(i).fdate;

local2d [i][8]=Admin.*FM*.localFlights.get(i).ftime;

local2d [i][9]=Admin.*FM*.localFlights.get(i).arrivalTime;

local2d [i][10]=Eseats;

local2d [i][11]=Bseats;

local2d [i][12]=isopen;

}

}

DefaultTableModel localtable = **new** DefaultTableModel(local2d,heading){

**public** **boolean** isCellEditable(**int** row, **int** col) {

**return** **false**;

}

};

scrollPane = **new** JScrollPane();

scrollPane.setBounds(10, 162, 1330, 218);

contentPane.add(scrollPane);

LocalTable = **new** JTable(localtable);

scrollPane.setViewportView(LocalTable);

LocalTable.setFillsViewportHeight(**true**);

LocalTable.setSurrendersFocusOnKeystroke(**true**);

LocalTable.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

LocalTable.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

JButton backB = **new** JButton("BACK");

backB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

backB.setBounds(554, 640, 117, 38);

contentPane.add(backB);

JButton searchB = **new** JButton("Search");

searchB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

String fromcity=fromCB.getSelectedItem().toString();

String tocity=toCB.getSelectedItem().toString();

**if**(fromcity.equals("-") || tocity.equals("-")) {

msgL.setText("First select cities to search");

}

**else** {

**for**(**int** i=0;i<Admin.*FM*.localFlights.size();i++) {

**if**(Admin.*FM*.localFlights.get(i).Airport.equals(fromcity) && Admin.*FM*.localFlights.get(i).getCity().equals(tocity)) {

*searchedlocal*.add(Admin.*FM*.localFlights.get(i));

BooklocalFlight blf=**new** BooklocalFlight();

blf.setVisible(**true**);

dispose();

}

**else** {

msgL.setText("No such Flight Available");

}

}

}

}

});

searchB.setBounds(523, 559, 178, 38);

contentPane.add(searchB);

}

}

**class** SignUp **extends** JFrame {

**private** JPanel contentPaneU;

**private** JTextField nameTF;

**private** JTextField cnicTF;

**private** JPasswordField passwordTF;

**private** JPasswordField conPassTF;

**private** JTextField usernameTF;

**static** **int** *userid*=1;

**static** ArrayList<User> *u*=**new** ArrayList<>();

**private** JTextField passportTF;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

SignUp frame = **new** SignUp();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** SignUp() {

setTitle("SIGNUP");

setBackground(Color.***WHITE***);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPaneU = **new** JPanel();

contentPaneU.setBackground(Color.***ORANGE***);

contentPaneU.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPaneU);

contentPaneU.setLayout(**null**);

JLabel Signup = **new** JLabel("PASSENGER SIGN UP");

Signup.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 18));

Signup.setBounds(541, 29, 242, 35);

contentPaneU.add(Signup);

JLabel NameL = **new** JLabel("Name: ");

NameL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

NameL.setBounds(301, 92, 123, 14);

contentPaneU.add(NameL);

JLabel CNICL = **new** JLabel("CNIC: ");

CNICL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

CNICL.setBounds(301, 196, 123, 14);

contentPaneU.add(CNICL);

JLabel GenderL = **new** JLabel("Gender:");

GenderL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

GenderL.setBounds(301, 310, 123, 14);

contentPaneU.add(GenderL);

JLabel AgeL = **new** JLabel("Age: ");

AgeL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

AgeL.setBounds(301, 366, 123, 14);

contentPaneU.add(AgeL);

JLabel PassL = **new** JLabel("Password: ");

PassL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

PassL.setBounds(301, 423, 123, 14);

contentPaneU.add(PassL);

JLabel conPassL = **new** JLabel("Confirm Password: ");

conPassL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

conPassL.setBounds(301, 481, 123, 14);

contentPaneU.add(conPassL);

JLabel cnichelp = **new** JLabel("(13 digit without -)");

cnichelp.setBounds(497, 224, 161, 14);

contentPaneU.add(cnichelp);

nameTF = **new** JTextField();

nameTF.setBounds(497, 92, 266, 20);

contentPaneU.add(nameTF);

nameTF.setColumns(10);

cnicTF = **new** JTextField();

cnicTF.setToolTipText("");

cnicTF.setColumns(10);

cnicTF.setBounds(497, 193, 266, 20);

contentPaneU.add(cnicTF);

JRadioButton maleB = **new** JRadioButton("Male");

maleB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

maleB.setBackground(Color.***ORANGE***);

maleB.setBounds(497, 306, 78, 23);

contentPaneU.add(maleB);

JRadioButton femaleB = **new** JRadioButton("Female");

femaleB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

femaleB.setBackground(Color.***ORANGE***);

femaleB.setBounds(595, 306, 85, 23);

contentPaneU.add(femaleB);

JRadioButton customB = **new** JRadioButton("Custom");

customB.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

customB.setBackground(Color.***ORANGE***);

customB.setBounds(689, 306, 109, 23);

contentPaneU.add(customB);

ButtonGroup gen=**new** ButtonGroup();

gen.add(customB);

gen.add(maleB);

gen.add(femaleB);

JComboBox ageCB = **new** JComboBox();

ageCB.setModel(**new** DefaultComboBoxModel(**new** String[] {"-", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59", "60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70", "71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81", "82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92", "93", "94", "95", "96", "97", "98", "99", "100"}));

ageCB.setBounds(497, 362, 55, 22);

contentPaneU.add(ageCB);

passwordTF = **new** JPasswordField();

passwordTF.setBounds(497, 420, 266, 20);

contentPaneU.add(passwordTF);

conPassTF = **new** JPasswordField();

conPassTF.setBounds(497, 475, 266, 20);

contentPaneU.add(conPassTF);

JLabel msg1 = **new** JLabel("");

msg1.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg1.setForeground(Color.***RED***);

msg1.setBounds(786, 92, 310, 14);

contentPaneU.add(msg1);

JLabel msg2 = **new** JLabel("");

msg2.setForeground(Color.***RED***);

msg2.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg2.setBounds(819, 143, 310, 14);

contentPaneU.add(msg2);

JLabel msg3 = **new** JLabel("");

msg3.setForeground(Color.***RED***);

msg3.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg3.setBounds(819, 196, 310, 14);

contentPaneU.add(msg3);

JLabel msg4 = **new** JLabel("");

msg4.setForeground(Color.***RED***);

msg4.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg4.setBounds(819, 310, 310, 14);

contentPaneU.add(msg4);

JLabel msg5 = **new** JLabel("");

msg5.setForeground(Color.***RED***);

msg5.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg5.setBounds(819, 426, 371, 14);

contentPaneU.add(msg5);

JLabel msg6 = **new** JLabel("");

msg6.setForeground(Color.***RED***);

msg6.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg6.setBounds(819, 481, 310, 14);

contentPaneU.add(msg6);

JLabel msg7 = **new** JLabel("");

msg7.setForeground(Color.***RED***);

msg7.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg7.setBounds(819, 546, 310, 14);

contentPaneU.add(msg7);

JLabel msg8 = **new** JLabel("");

msg8.setForeground(Color.***RED***);

msg8.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg8.setBounds(819, 366, 310, 14);

contentPaneU.add(msg8);

JLabel passportL = **new** JLabel("Passport no: ");

passportL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

passportL.setBounds(301, 252, 123, 14);

contentPaneU.add(passportL);

passportTF = **new** JTextField();

passportTF.setToolTipText("");

passportTF.setColumns(10);

passportTF.setBounds(497, 249, 266, 20);

contentPaneU.add(passportTF);

JLabel passporthelp = **new** JLabel("(9 digit without -)");

passporthelp.setBounds(497, 280, 161, 14);

contentPaneU.add(passporthelp);

JLabel msg9 = **new** JLabel("");

msg9.setForeground(Color.***RED***);

msg9.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

msg9.setBounds(828, 252, 310, 14);

contentPaneU.add(msg9);

JCheckBox understandCB = **new** JCheckBox("I Understand all the terms and conditions");

understandCB.setBackground(Color.***ORANGE***);

understandCB.setBounds(301, 546, 271, 23);

contentPaneU.add(understandCB);

JLabel UserL = **new** JLabel("USERNAME: ");

UserL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

UserL.setBounds(301, 143, 123, 14);

contentPaneU.add(UserL);

usernameTF = **new** JTextField();

usernameTF.setColumns(10);

usernameTF.setBounds(497, 143, 266, 20);

contentPaneU.add(usernameTF);

JButton SignupB = **new** JButton("SIGNUP");

SignupB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** nameformat;

**boolean** namelength;

**boolean** useravailable;

**boolean** cniclength;

**boolean** cnicavailable;

**boolean** passportlength;

**boolean** passportformat;

**boolean** cnicformat;

**boolean** gendercheck = **false**;

**boolean** agecheck = **false**;

**boolean** passlength;

**boolean** passformat;

**boolean** passMatch;

String gender = **null**;

**int** age = 0;

String name = **null**;

String pass = **null**;

String userName = **null**;

String cnic = **null**;

String passport=**null**;

**if**(understandCB.isSelected()) {

namelength=Formats.*Namelength*(nameTF.getText());

nameformat=Formats.*NameFormatFun*(nameTF.getText());

useravailable=Formats.*usernameAvailable*(*u*, usernameTF.getText());

cniclength=Formats.*cniclength*(cnicTF.getText());

cnicformat=Formats.*cnicFormat*(cnicTF.getText());

cnicavailable=Formats.*cnicavailable*(*u*, cnicTF.getText());

passportlength=Formats.*Passportlength*(passportTF.getText());

passportformat=Formats.*passportFormat*(passportTF.getText());

passlength=Formats.*Passwordlength*(passwordTF.~~getText~~());

passformat=Formats.*PasswordFormat*(passwordTF.~~getText~~());

passMatch=Formats.*Passwordmatch*(passwordTF.~~getText~~(), conPassTF.~~getText~~());

**if**(!namelength) {

msg1.setText("Name must be less than 37 characters");

}

**else** **if**(!nameformat){

msg1.setText("Name must not contain special characters or numbers");

}

**else** {

msg1.setText(" ");

}

**if**(!useravailable) {

msg2.setText("Account already exists on this username");

}

**else** **if**(useravailable) {

userName=usernameTF.getText();

msg2.setText(" ");

}

**if**(cnicTF.getText().isEmpty()) {

msg3.setText(" ");

}

**else** **if**(!cniclength) {

msg3.setText("CNIC must be of 13 digit");

}

**else** **if**(!cnicformat){

msg3.setText("CNIC must contain number only");

}

**else** **if**(!cnicavailable){

msg3.setText("Account already exists on this cnic");

}

**else** {

msg3.setText(" ");

}

**if**(!passportlength) {

msg9.setText("Passport number must be of 9 digit");

}

**else** **if**(!passportformat){

msg9.setText("Passport must contain number only");

}

**else** {

msg9.setText(" ");

}

**if**(ageCB.getSelectedItem().equals("-") ) {

msg8.setText("select age plz");

}

**else** {

agecheck=**true**;

String a=(String) ageCB.getSelectedItem();

age=Integer.*parseInt*(a);

msg8.setText(" ");

}

**if**(maleB.isSelected()) {

gender="M";

gendercheck=**true**;

}

**else** **if**(femaleB.isSelected()) {

gender="F";

gendercheck=**true**;

}

**else** **if**(customB.isSelected()) {

gender=customB.getText();

gendercheck=**true**;

}

**if**(!gendercheck) {

msg4.setText("Please select your Gender");

}

**else** {

msg4.setText(" ");

}

**if**(!passlength) {

msg5.setText("please enter minimum 8 character");

}

**else** **if**(!passformat){

msg5.setText("password should contain atleast 1 capital,small letter, number and special character ");

}

**else** {

msg5.setText(" ");

}

**if**(!passMatch) {

msg6.setText("Password Doesn't Match");

}

**else** {

msg6.setText(" ");

}

**if**(namelength && nameformat) {

name=nameTF.getText();

}

**if**(passMatch && passformat && passlength) {

pass=passwordTF.~~getText~~();

}

**if**(cniclength && cnicformat && cnicavailable) {

cnic=cnicTF.getText();

}

**if**(passportlength && passportformat) {

passport=passportTF.getText();

}

**if**(namelength && nameformat && passlength && passformat && passMatch && cniclength && cnicformat && cnicavailable && gendercheck && agecheck && useravailable && passportlength && passportformat) {

User temp =**new** User(*userid*,name,userName,cnic,gender,age,pass,passport);

*userid*++;

*u*.add(temp);

SuccessfulSignupBox ssb=**new** SuccessfulSignupBox();

ssb.setVisible(**true**);

dispose();

}

}

**else** {

msg7.setText("Check the box to continue");

}

}

});

SignupB.setBounds(499, 604, 138, 23);

contentPaneU.add(SignupB);

JButton MainB = **new** JButton("Main Menu");

MainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainB.setBounds(497, 669, 140, 23);

contentPaneU.add(MainB);

JButton instruction = **new** JButton("?");

instruction.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

PassportInstruction pi=**new** PassportInstruction();

pi.setVisible(**true**);

}

});

instruction.setBounds(773, 248, 45, 23);

contentPaneU.add(instruction);

}

**static** **public** **void** addsomeUsers() {

User u1=**new** User(*userid*,"Huzaifa","ranahuzaifa","3650247203719","Male",21,"Huzaif@111","090078601");*userid*++;

User u2=**new** User(*userid*,"Umar","byumar","1234567890123","Male",22,"Carbon12>","121212121");*userid*++;

User u3=**new** User(*userid*,"Wajahat","iamwajahat","9876543210876","Male",20,"Xylafer1<","555666777");*userid*++;

User u4=**new** User(*userid*,"Moiz","mueezism","1112223334445","Male",19,"Imaan\*1","123456789");*userid*++;

*u*.add(u1);

*u*.add(u2);

*u*.add(u3);

*u*.add(u4);

}

}

**class** SuccessfullyPurchased **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

SuccessfullyPurchased frame = **new** SuccessfullyPurchased();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** SuccessfullyPurchased() {

setTitle("Flight Purchased Successfully");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Flight Purchased Successfully");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(41, 79, 317, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SearchLocal.*searchedlocal*.clear();

SearchInternational.*searchedinter*.clear();

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

});

okB.setBounds(156, 181, 89, 23);

contentPane.add(okB);

}

}

**class** SuccessfulSignupBox **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

SuccessfulSignupBox frame = **new** SuccessfulSignupBox();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** SuccessfulSignupBox() {

setTitle("Successfuly Signed Up");

setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

setBounds(100, 100, 438, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel balL = **new** JLabel("Successfully Signed Up");

balL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

balL.setForeground(Color.***BLUE***);

balL.setBounds(82, 46, 324, 52);

contentPane.add(balL);

JLabel lblNowYouCan = **new** JLabel("Now you can Login");

lblNowYouCan.setForeground(Color.***BLACK***);

lblNowYouCan.setFont(**new** Font("Tahoma", Font.***BOLD***, 16));

lblNowYouCan.setBounds(117, 97, 268, 25);

contentPane.add(lblNowYouCan);

JButton loginB = **new** JButton("Login");

loginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

loginB.setBounds(148, 150, 89, 32);

contentPane.add(loginB);

}

}

**class** UserLogin **extends** JFrame {

**private** JPanel contentPane;

**private** JTextField userTF;

**private** JPasswordField passTF;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

UserLogin frame = **new** UserLogin();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** UserLogin() {

setTitle("USER LOGIN");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JLabel userL = **new** JLabel("USERNAME:");

userL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

userL.setBounds(504, 219, 106, 14);

contentPane.add(userL);

JLabel passL = **new** JLabel("PASSWORD:");

passL.setFont(**new** Font("Tahoma", Font.***BOLD***, 11));

passL.setBounds(504, 309, 106, 14);

contentPane.add(passL);

userTF = **new** JTextField();

userTF.setBounds(669, 216, 211, 20);

contentPane.add(userTF);

userTF.setColumns(10);

passTF = **new** JPasswordField();

passTF.setBounds(669, 306, 211, 20);

contentPane.add(passTF);

JLabel msgL = **new** JLabel("");

msgL.setForeground(Color.***RED***);

msgL.setBounds(669, 349, 211, 14);

contentPane.add(msgL);

JButton loginB = **new** JButton("LOGIN");

loginB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**boolean** canLogin=**false**;

canLogin=canLogin(userTF.getText(),passTF.~~getText~~());

**if**(canLogin) {

UserLoginSuccess uls=**new** UserLoginSuccess();

uls.setVisible(**true**);

dispose();

}

**else** {

msgL.setText("Incorrect email or password");

}

}

});

loginB.setBounds(669, 424, 132, 33);

contentPane.add(loginB);

JLabel lblUserLoginMenu = **new** JLabel("USER LOGIN MENU");

lblUserLoginMenu.setForeground(Color.***BLUE***);

lblUserLoginMenu.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblUserLoginMenu.setBounds(563, 110, 231, 33);

contentPane.add(lblUserLoginMenu);

JButton MainMenuB = **new** JButton("Main Menu");

MainMenuB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m=**new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

MainMenuB.setBounds(669, 566, 132, 33);

contentPane.add(MainMenuB);

}

**public** **boolean** canLogin(String uname,String pass) {

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).userName.equals(uname) && SignUp.*u*.get(i).password.equals(pass) ) {

SignUp.*u*.get(i).setLogin\_status(**true**);

**return** **true**;

}

}

**return** **false**;

}

}

**class** UserLoginSuccess **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

UserLoginSuccess frame = **new** UserLoginSuccess();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** UserLoginSuccess() {

setTitle("USER");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(0, 0, 1400, 800);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel tittleL = **new** JLabel("New-PAK Airline Flight System (NPAFS)");

tittleL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

tittleL.setBounds(467, 30, 492, 63);

contentPane.add(tittleL);

JLabel lblSuccessfullyLoggedIn = **new** JLabel("Successfully Logged in");

lblSuccessfullyLoggedIn.setForeground(Color.***BLUE***);

lblSuccessfullyLoggedIn.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblSuccessfullyLoggedIn.setBounds(544, 104, 262, 63);

contentPane.add(lblSuccessfullyLoggedIn);

JButton BookInterB = **new** JButton("Reserve Flight (International)");

BookInterB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SearchInternational si=**new** SearchInternational();

si.setVisible(**true**);

dispose();

}

});

BookInterB.setBounds(355, 311, 190, 42);

contentPane.add(BookInterB);

JButton btnAccountSettings = **new** JButton("Current Booked Flights");

btnAccountSettings.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

CurrentBookedFlights cbf=**new** CurrentBookedFlights();

cbf.setVisible(**true**);

dispose();

}

});

btnAccountSettings.setBounds(355, 562, 190, 42);

contentPane.add(btnAccountSettings);

JButton btnCancelReservation = **new** JButton("Cancel Reservation");

btnCancelReservation.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

CancelBookedFlight cbl=**new** CancelBookedFlight();

cbl.setVisible(**true**);

dispose();

}

});

btnCancelReservation.setBounds(355, 445, 190, 42);

contentPane.add(btnCancelReservation);

JButton btnLogout = **new** JButton("Logout");

btnLogout.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**for**(**int** i=0;i<SignUp.*u*.size();i++) {

**if**(SignUp.*u*.get(i).login\_status==**true** ) {

SignUp.*u*.get(i).setLogin\_status(**false**);

}

}

LogoutBox lb=**new** LogoutBox();

lb.setVisible(**true**);

dispose();

}

});

btnLogout.setBounds(725, 562, 190, 42);

contentPane.add(btnLogout);

JButton btnAccountSettings\_1 = **new** JButton("Account Settings");

btnAccountSettings\_1.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

AccountSettingUser asu=**new** AccountSettingUser();

asu.setVisible(**true**);

dispose();

}

});

btnAccountSettings\_1.setBounds(725, 311, 190, 42);

contentPane.add(btnAccountSettings\_1);

JButton mainB = **new** JButton("Main Menu");

mainB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

MainMenu m = **new** MainMenu();

m.setVisible(**true**);

dispose();

}

});

mainB.setBounds(725, 445, 190, 42);

contentPane.add(mainB);

JButton balanceB = **new** JButton("Check Balance");

balanceB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

BalanceBox bb=**new** BalanceBox();

bb.setVisible(**true**);

}

});

balanceB.setBounds(725, 207, 190, 42);

contentPane.add(balanceB);

JButton BookLocalB = **new** JButton("Reserve Flight(Local)");

BookLocalB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

SearchLocal sl=**new** SearchLocal();

sl.setVisible(**true**);

dispose();

}

});

BookLocalB.setBounds(355, 207, 190, 42);

contentPane.add(BookLocalB);

}

}

**class** VisaMsg **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

VisaMsg frame = **new** VisaMsg();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** VisaMsg() {

setTitle("Visa not Verified");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

setBounds(100, 100, 559, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** MatteBorder(1, 1, 1, 1, (Color) **new** Color(0, 0, 0)));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel lblNewLabel = **new** JLabel("Your Passport doesn't have stamp of VISA");

lblNewLabel.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblNewLabel.setForeground(Color.***BLUE***);

lblNewLabel.setBounds(42, 44, 491, 52);

contentPane.add(lblNewLabel);

JButton okB = **new** JButton("OKAY");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

dispose();

}

});

okB.setBounds(210, 207, 89, 23);

contentPane.add(okB);

JLabel lblYouCannotTravell = **new** JLabel("You cannot travell Internationaly");

lblYouCannotTravell.setForeground(Color.***BLUE***);

lblYouCannotTravell.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

lblYouCannotTravell.setBounds(75, 97, 491, 52);

contentPane.add(lblYouCannotTravell);

}

}

**class** VisaVerified **extends** JFrame {

**private** JPanel contentPane;

/\*\*

\* Launch the application.

\*/

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

VisaVerified frame = **new** VisaVerified();

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the frame.

\*/

**public** VisaVerified() {

setTitle("Visa Verified Successfully");

setDefaultCloseOperation(JFrame.***HIDE\_ON\_CLOSE***);

setBounds(100, 100, 450, 300);

contentPane = **new** JPanel();

contentPane.setBackground(Color.***ORANGE***);

contentPane.setBorder(**new** LineBorder(**new** Color(0, 0, 0)));

setContentPane(contentPane);

contentPane.setLayout(**null**);

JLabel balL = **new** JLabel("Visa Verified");

balL.setFont(**new** Font("Tahoma", Font.***BOLD*** | Font.***ITALIC***, 20));

balL.setForeground(Color.***BLUE***);

balL.setBounds(141, 64, 166, 52);

contentPane.add(balL);

JButton okB = **new** JButton("OK");

okB.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

allUsersDetails aud=**new** allUsersDetails();

aud.setVisible(**true**);

dispose();

}

});

okB.setBounds(163, 161, 89, 30);

contentPane.add(okB);

}

}

**==\*-- THE END --\*==**