Help Document PA6+Final Project

This document will go over the functionality that should have been achieved in PA6 as well as provide hints for the final project.

Catching up with PA6 Requirements:

1.Addcustomer

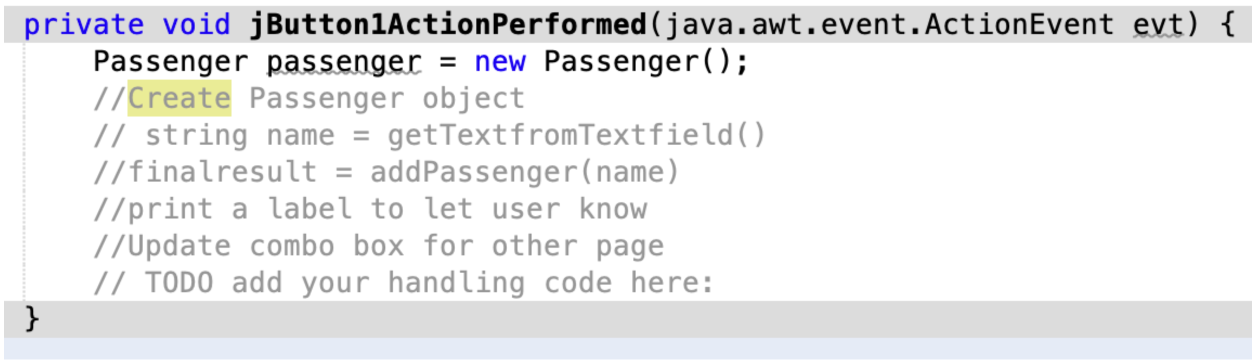
2.AddBooking

3.StatusWaitlist

4.StatusFlightDay

IMPORTANT:

Before we begin please be advised it is best practice to write your code logic under the action listeners for the BUTTONS only on each page this will cause far less problems and help you complete your work smoothly, example below writes the COMPLETE code for adding a passenger under action listener for add passenger BUTTON:

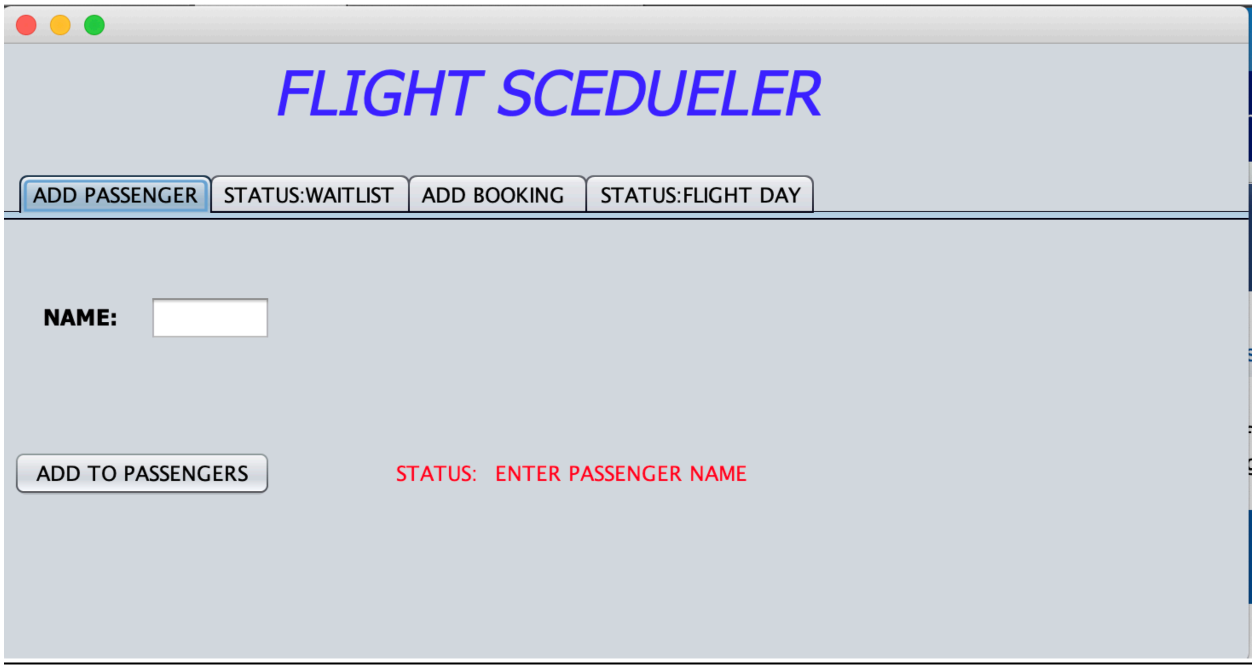


```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    Passenger passenger = new Passenger();
    //Create Passenger object
    // string name = getTextfromTextfield()
    //finalresult = addPassenger(name)
    //print a label to let user know
    //Update combo box for other page
    // TODO add your handling code here:
}
```

Same should be followed for all other functionalities.

1.ADD-CUSTOMER:

- **Required: “name”**



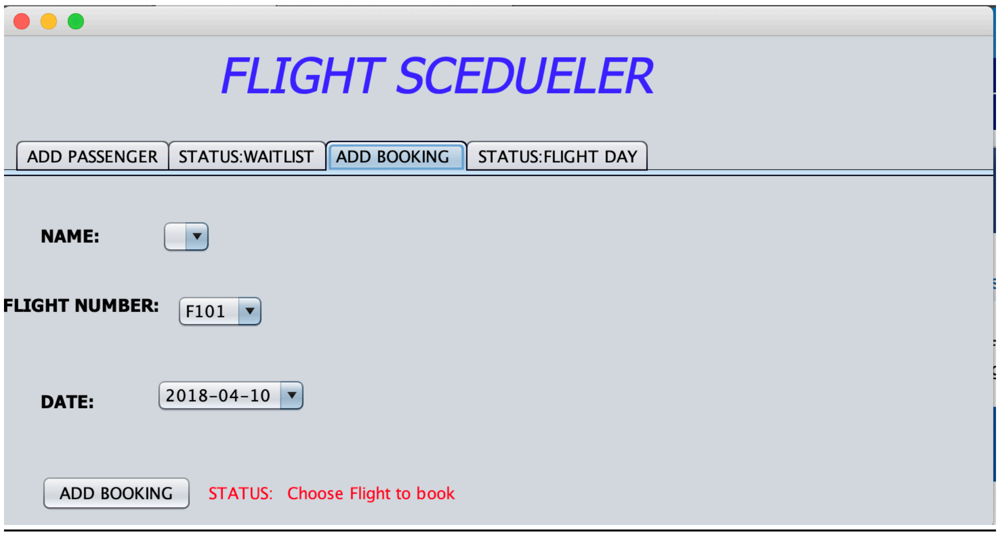
STATUS: ENTER PASSENGER NAME

Code under “add button action listener”:

```
private void add-ToPassenger-Button(java.awt.event.ActionEvent evt) {
        Passenger passenger = new Passenger();
        //Create Passenger object
        // string name = getTextfromTextfield()
        //finalresult = addPassenger(name)
        //print a label to let user know
        //Update combo box for other page
  }
```

2.ADD-PASSENGER:

- **Required:**
  - **Functional Add passenger tab**
  - **Name from combo box1**
  - **Flight number (preloaded to database)**
  - **Date of flight (preloaded to database)**



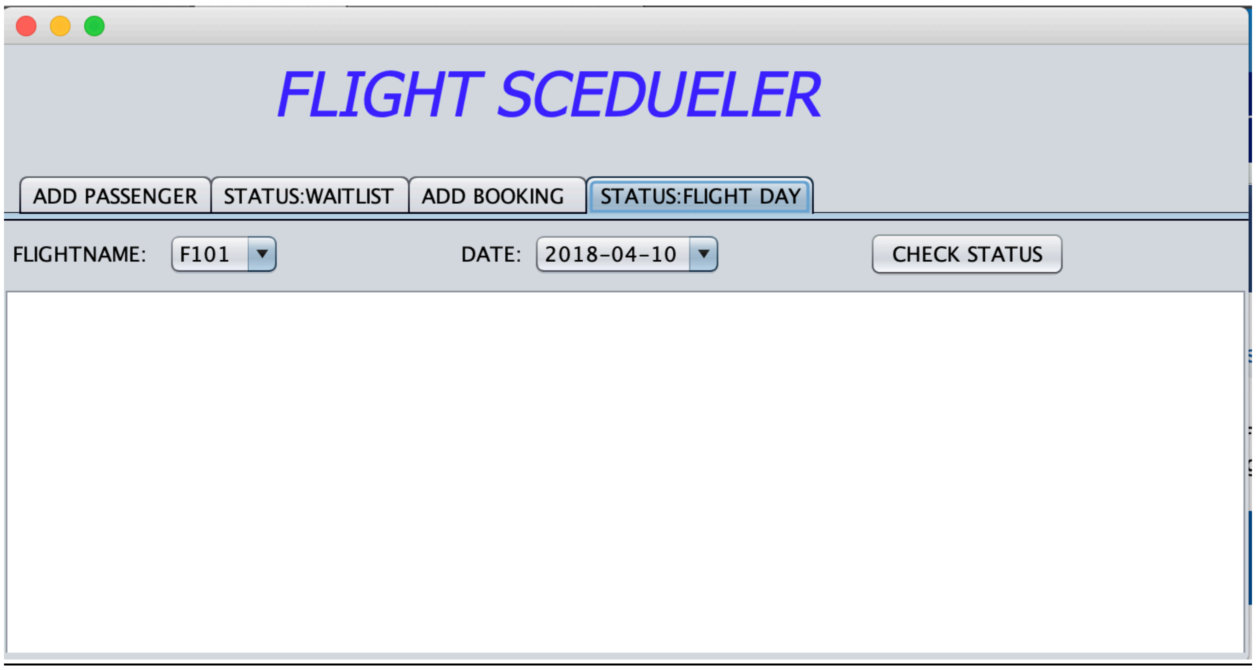
STATUS: Choose Flight to book

Code under "add booking button action listener":

```
private void Add-Booking-ButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        //name =getnamefromcombobox()
        //flight = getflightnamefromcombobox()
        //date = getDatefromCombo()
        //Timestamp timestamp = new
java.sql.Timestamp(Calendar.getInstance().getTime().getTime());
        //create waitlist item
        //create bookingitem
        //createflightitem
        //seats taken =  booking.getSeatsBooked(flightName, day);
        //seats left = flight.getLeftSeats(flightName);
        //if (taken = left){waitlist else book)
        //print label for the action
        }
```

3.FLIGHT-DAY:

- **Required:**
  - **Working addbooking tab**
  - **Flightname**
  - **Flightdate**

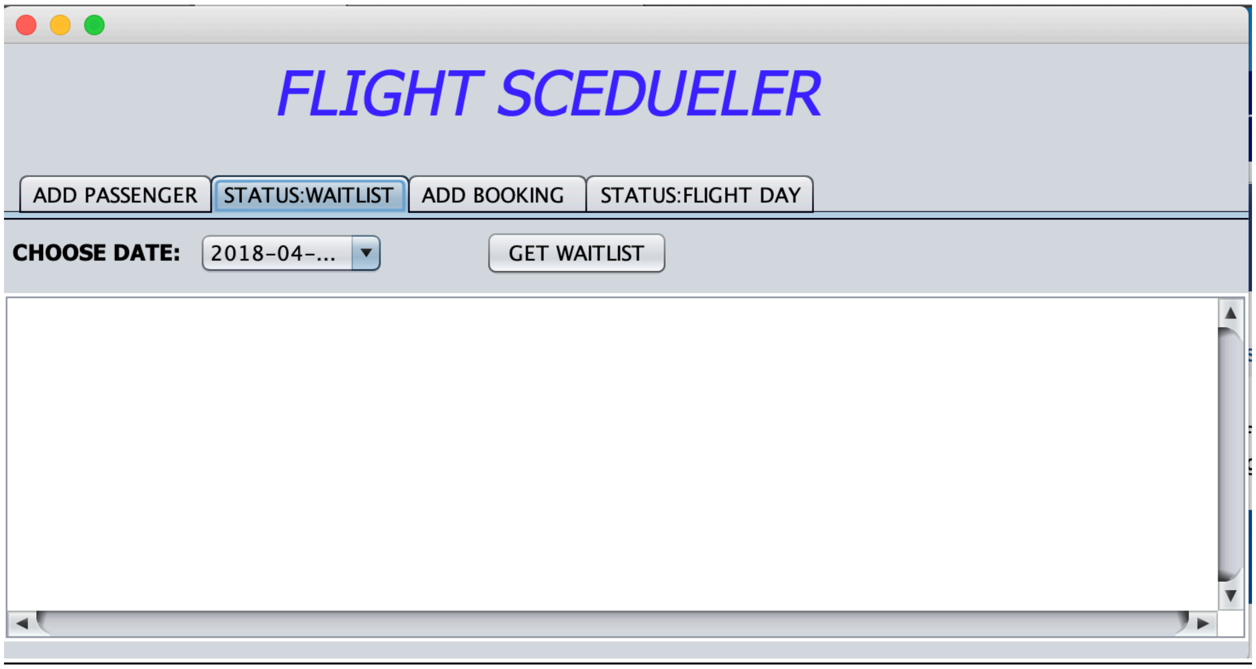


Code under "check status button action listener":

```
private void Flight-DayStatus-ButtonActionPerformed(java.awt.event.ActionEvent
evt) {
   //create booking object
   //date = getdatefromcombo
   //name = getflightnamefromcombo
   //create list
   //result = book.sql-query-to-get-all-booked
   //all results to be looped into the list
   //textbox.settext(results)
   }
```

4.WAITLIST

- **Required:**
  - **Date**
  - **Working booking tab**



Code under "get waitlist button action listener":

```
private void get-waitlist-button(java.awt.event.ActionEvent evt) {
   //create wait list item
   //date = getdatefromCombo
   //create List
   //result = sql-query-to-get-customers-on-waitlist-table
   //loop all result to list
   //textarea.settext(result)
   }
```

Hints for Final Project

Final Project Requirements:

1.AddFlight

2.AddDate

3.StatusCustomer

4.Cancel

5.Drop

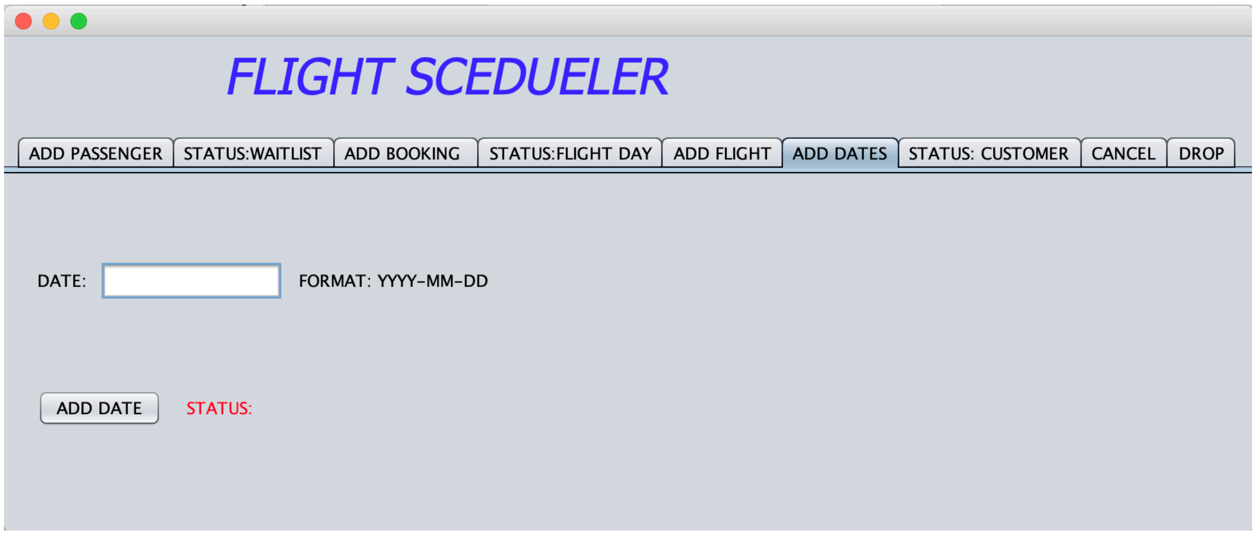
IMPORTANT:

1.All pictures below are suggested templates

2.Truncate your database of any preloaded data Final requires NO preloaded data.

1.<u>ADD-DATE:</u>

- **Required:**
  - **Date text field**
  - **Add button**
  - **Status label**

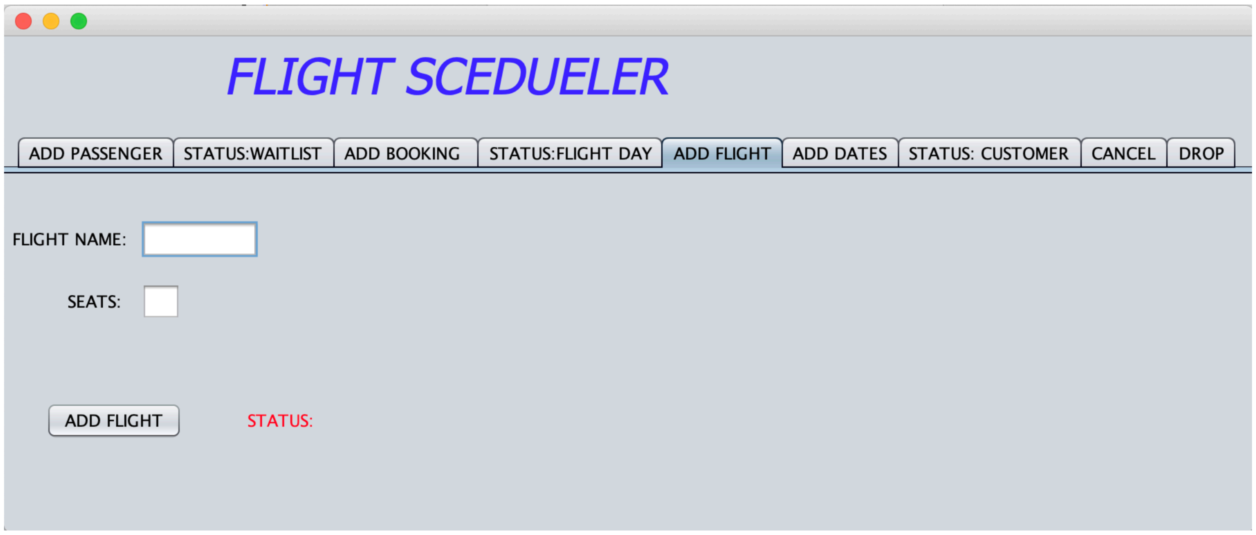


<u>Hints for Code under "add Date button action listener":</u>

1.my sql table requires only one column
2.Apply same idea as adding passenger name
3.error detection
4.I want the status label to inform my user of the action (fail/success)

2.ADD-FLIGHT:

- **Required:**
    - **Flight name text field**
    - **Seats on flight**
    - **Add button**
    - **Status label**

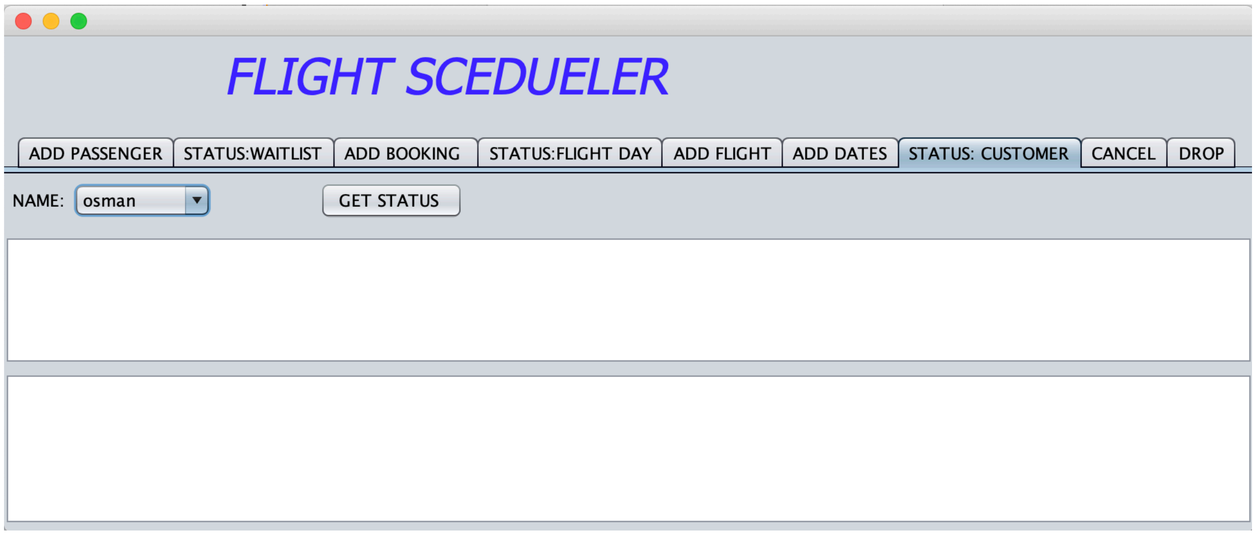


Hints for Code under "add flight button action listener":

1.my sql table requires only two columns
2.Apply same idea as adding passenger name
3.error detection
4.I want the status label to inform my user of the action (fail/success)

3.CUSTOMER-STATUS

- **Required:**
    - **Name of passenger**
    - **Button**
    - **Two boxes maybe? (waitlist & confirmed flight)**

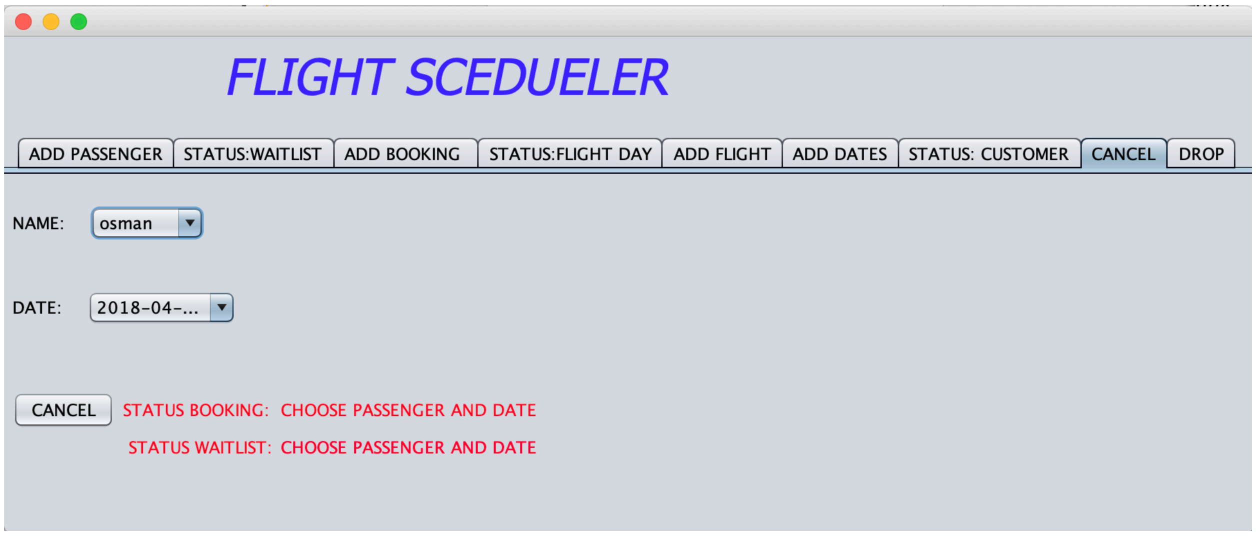


Hints for Code under "get status button action listener":

1.I want to know if my customer is on a waitlist or/and flight
2.I should be simultaneously querieng and displaying data from two sql tables
3.Two objects using two sql queries for the same name of passenger (waitlistquery/bookingquery)
3.i should print my data in a consistent manner

4.CANCEL

- **Required:**
  - **Name of passenger**
  - **date**
  - **cancel button**
  - **one label each for waitlist/booking**



STATUS BOOKING: CHOOSE PASSENGER AND DATE
STATUS WAITLIST: CHOOSE PASSENGER AND DATE

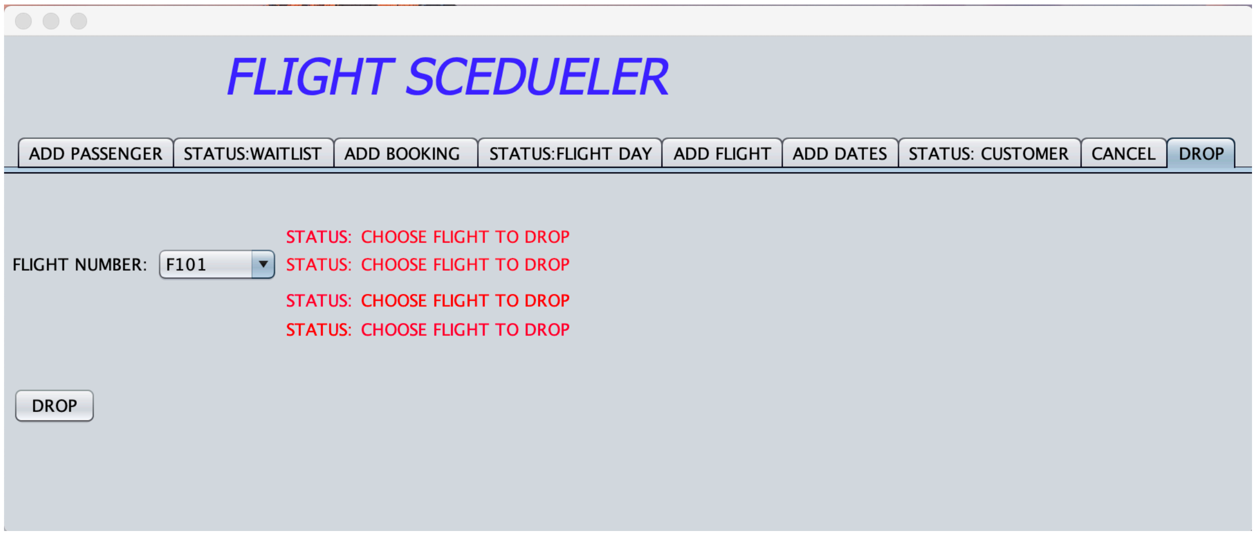
<u>Hints for Code under “cancel button action listener”:</u>

1.When I hit cancel I must do two things:
- remove passenger for this date from any booking
- remove passenger for this date from any waitlist

2.Must once again run two parallel queries with arguments from the combo boxes
3.Upon cancel I must delete said query from both table if it exists in either.

5.DROP

- **Required:**
    - **Name of flight**



STATUS: CHOOSE FLIGHT TO DROP
STATUS: CHOOSE FLIGHT TO DROP
STATUS: CHOOSE FLIGHT TO DROP
STATUS: CHOOSE FLIGHT TO DROP

<u>Hints for Code under "cancel button action listener":</u>

1.What I intend to do:
- Delete selected fligth from databse
- Delete anyone on waitlist on anyday for said flight.
- Check for passengers booked on this flight:
    - If yes I must book them on the next free flight date
    - Show all labels for this

2.This seems a lot, but if one chronologically checks for these conditions they should be able to do fine
3.An idea to check the next available flight for rebooking is to have flight names in an array and loop and check for the first one that has seats left using a sql query function (getSeatsLeft) if yes break and book person on this, else go on till end.
4. a great way would be to draw a picture and solve the rebooking process on paper first before just writing code!