

Homework 3

Due date:

Sunday 7/10 at 11:59 pm

Implement the following 3 use cases

1. Manage Travel Agency

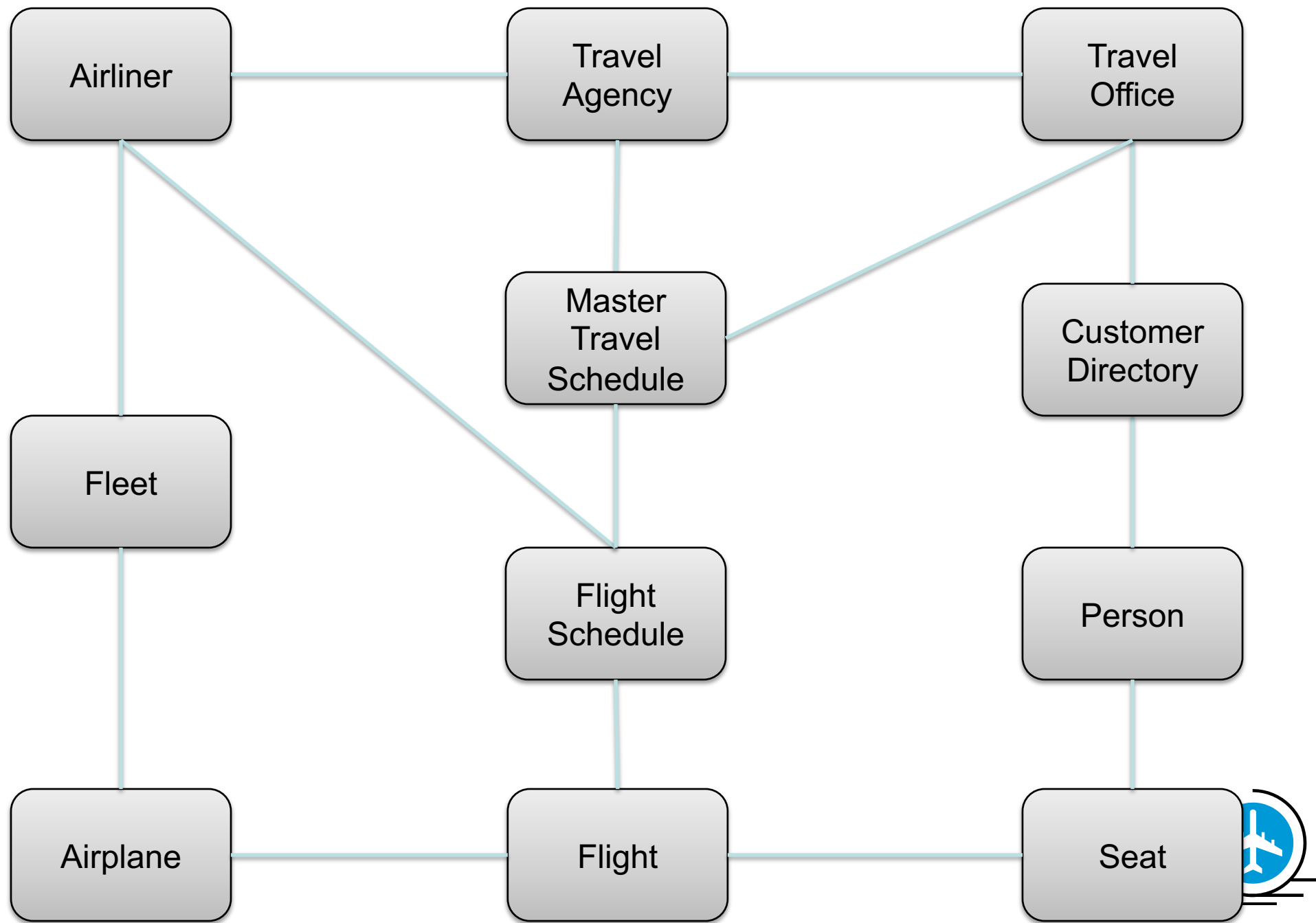
1. Create new airliner
2. View airliner profiles

2. Airlines Manage flight schedules

1. Manage airliner flight schedules

3. Manage customers

1. Search for best flights (all flights, across all airlines)
2. Book customer reservation on a flight



Objectives

- Specify the business problem
 - Multi-retailer and multi-airliner inventory system
- Specify and implement user interfacing processes
 - Identify key stakeholder
 - Identify stakeholder user processes
- Expand the object model to support the travel system

Need to know

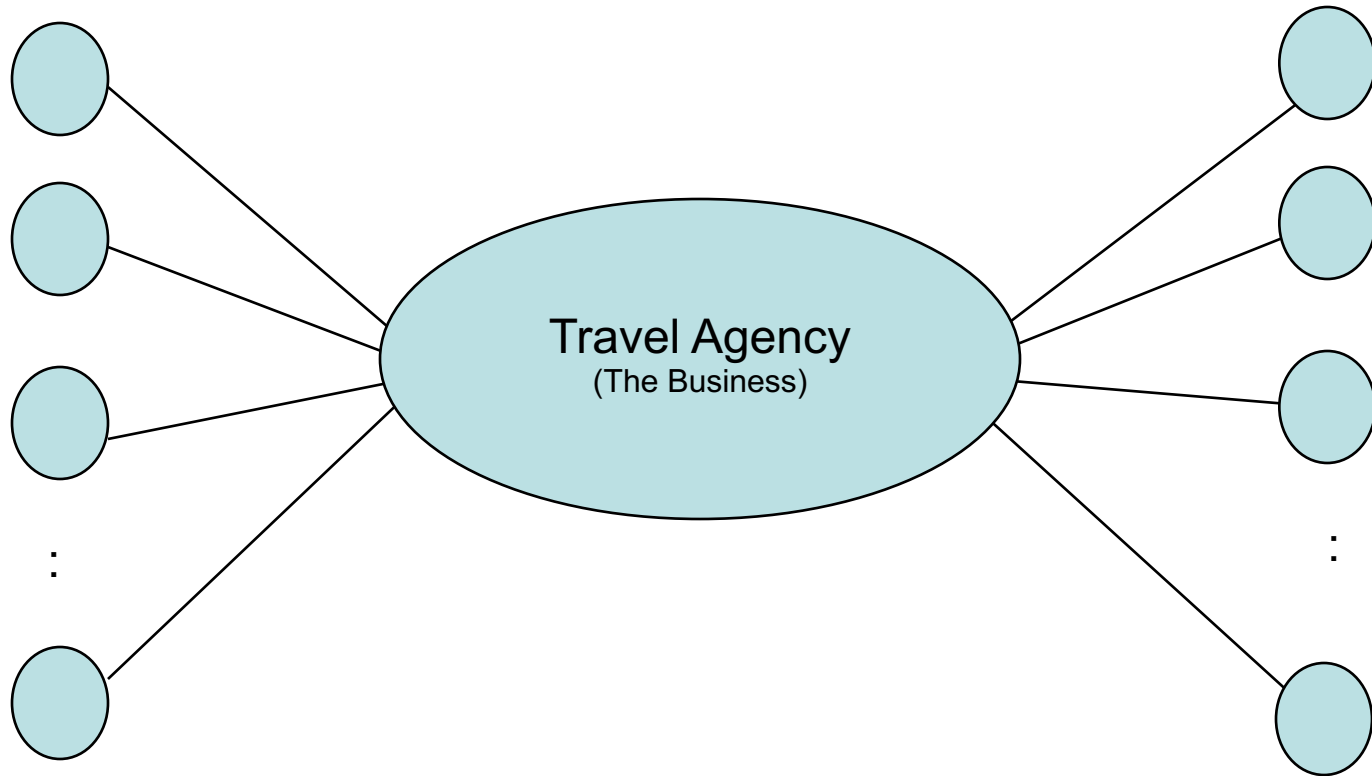
- How to specify processes in swing?
 - Cardlayout
- How to expand the object model
 - Add Travel Agency class, Airliner, Airliner directory, customer, and customer directory, and user directory
- How to link the expanded object model to the user processes defined in swing?

The Business Model

The Travel Agency View

Customers

Airliners



Key Travel Agency Use Case

1. Manage Travel Agency
 1. Create new airliner
 2. View airliner profiles
2. Manage customers
 1. Search for best flights (all flights, across all airlines)
 2. Book customer reservation on a flight

Airliner: Manage flight schedule

Manage airliner flight schedules

- List all flights
- Add flight to flight schedule
 - Morning, day, and evening
 - From location
 - To location
- Select and update flight information
- Select and cancel flight

Steps

- Define new project
- Define a business package under src
- Define `UserInterface` package
 - define subpackages for each user process
- New `JFrame` (`TravelAgencyMain`)
- Include splitpane
- Add `jpanel` to the right side of the split pane
 - Set the layout manager to absolute layout
- Add `JPanel` to the left-side of the splitpane
 - Set the layout manager as the cardlayout
 - Rename the `JPanel` as the `CardSequenceJPanel`
- Define stakeholder action buttons on the right-hand side of `jpanel` of the splitpane

The Object Model Supports the following

- Business (travel agency)
 - Name
 - Search for flights across airliner flight schedules
 - Airliner Directory
 - Airliner
 - User
 - Flight Schedule
 - » Flight
 - » Add flight, find and update flight, select and cancel flight

Scenario I: Manage Airlines

- User chooses to manage airlines
- System displays screen with a table of list of airlines and with the following options:
 - Add a new airline
 - System displays a new airline screen
 - User enters airline information and hits ok or cancel to go back
 - Select/Browse a airline
 - System retrieves selected airline and displays airline information.
 - If a user chooses to update airline information system takes the user to an update airline screen where the user can update airline information and returns to airline view screen with updated information

How to define a sequence of screens?

- Add a subpackage under the `UserInterface`. call it `ManageAirlines`
- Define three `JPanel`s under the `ManageAirlines` subpackage
 - `ManageAirlinesJPanel`
 - Add a `JLabel` on the `JPanel` canvas and rename it as “Manage Airlines”
 - `CreateNewAirlinerJPanel`
 - `ViewAirlinerJPanel`

ManageAirlinersJPanel

- Redefine the arguments on the constructor so it accepts two arguments

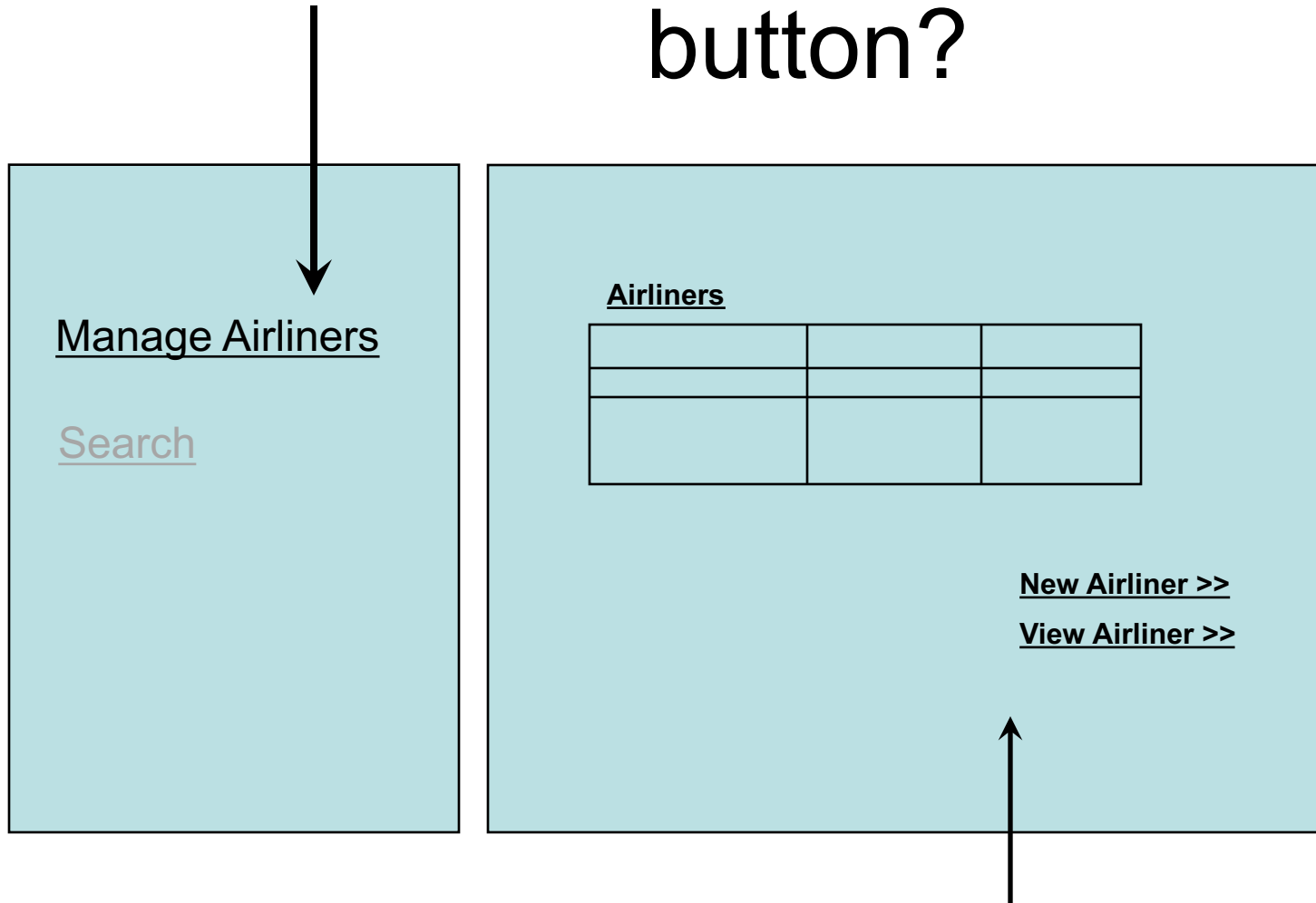
Class ManageAirlinerJPanel extends JPanel {

AirlinerDirectory airlinerdirectory;
javax.swing.JPanel CardSequenceJPanel;

```
public ManageAirlinerJPanel(javax.swing.JPanel spane, AirlinerDirectory sd) {  
    CardSequenceJPanel = spane;  
    airlinerdirectory = sd;  
    :  
<Display the airliner list in the supplier table here>  
    :  
    initComponents();  
}
```

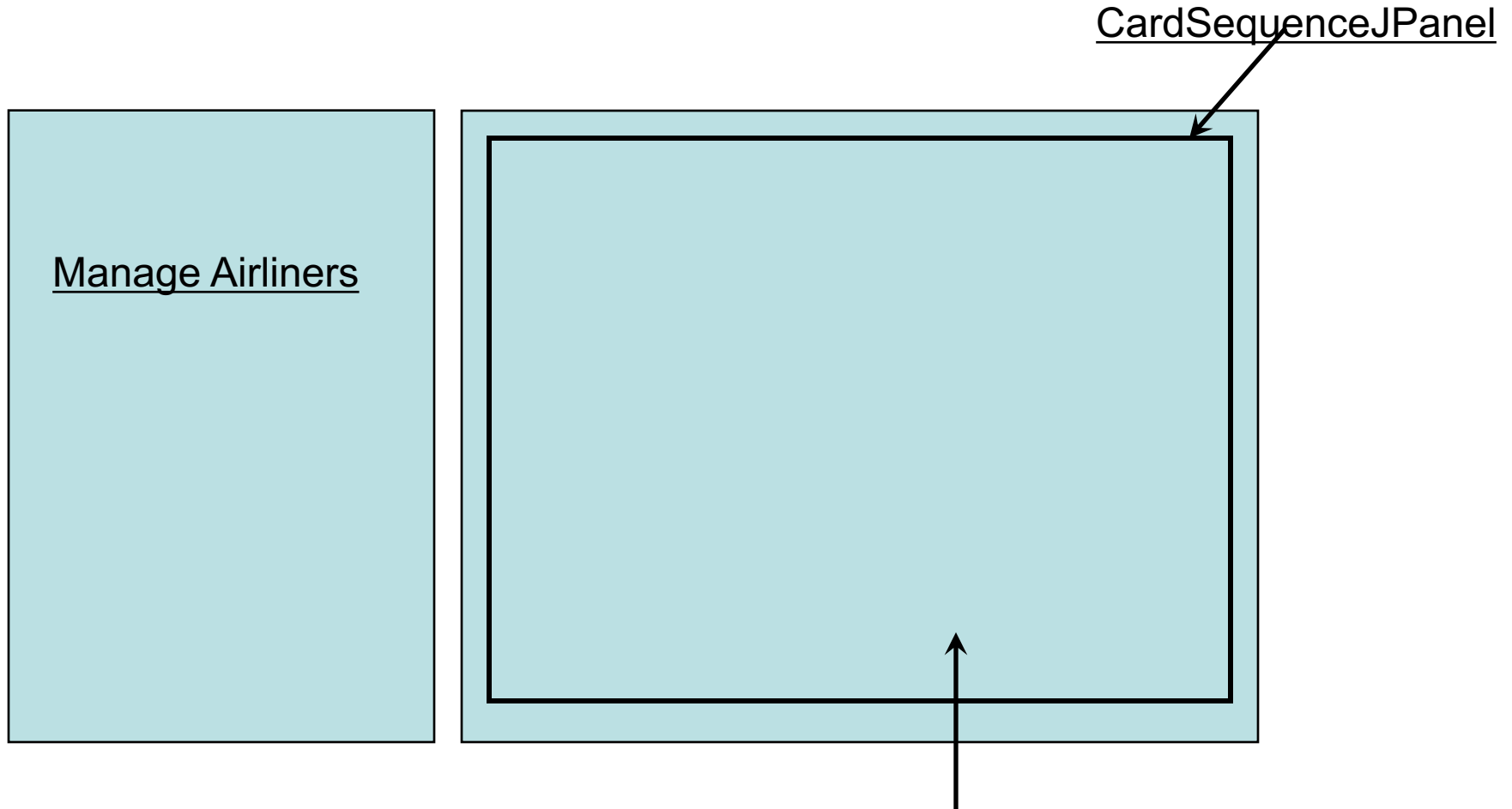
Note: Must import AirlinerDirectory from the business side

What Happens when the user presses the Manage Airlines button?



Show the ManageAirlinerJPanel in this area

The Splitpane inside the JFrame



Must insert the ManageAirlinesJPanel inside the CardSequenceJPanel

How to create and insert an instance of ManageAirlinesJPanel?

Manage Airlines

Assume travelagency is an instance of the TravelAgency class

```
AirlinerDirectory ad = travelagency.getAirlinerDirectory();
```

```
CardSequenceJPanel.removeAll();
```

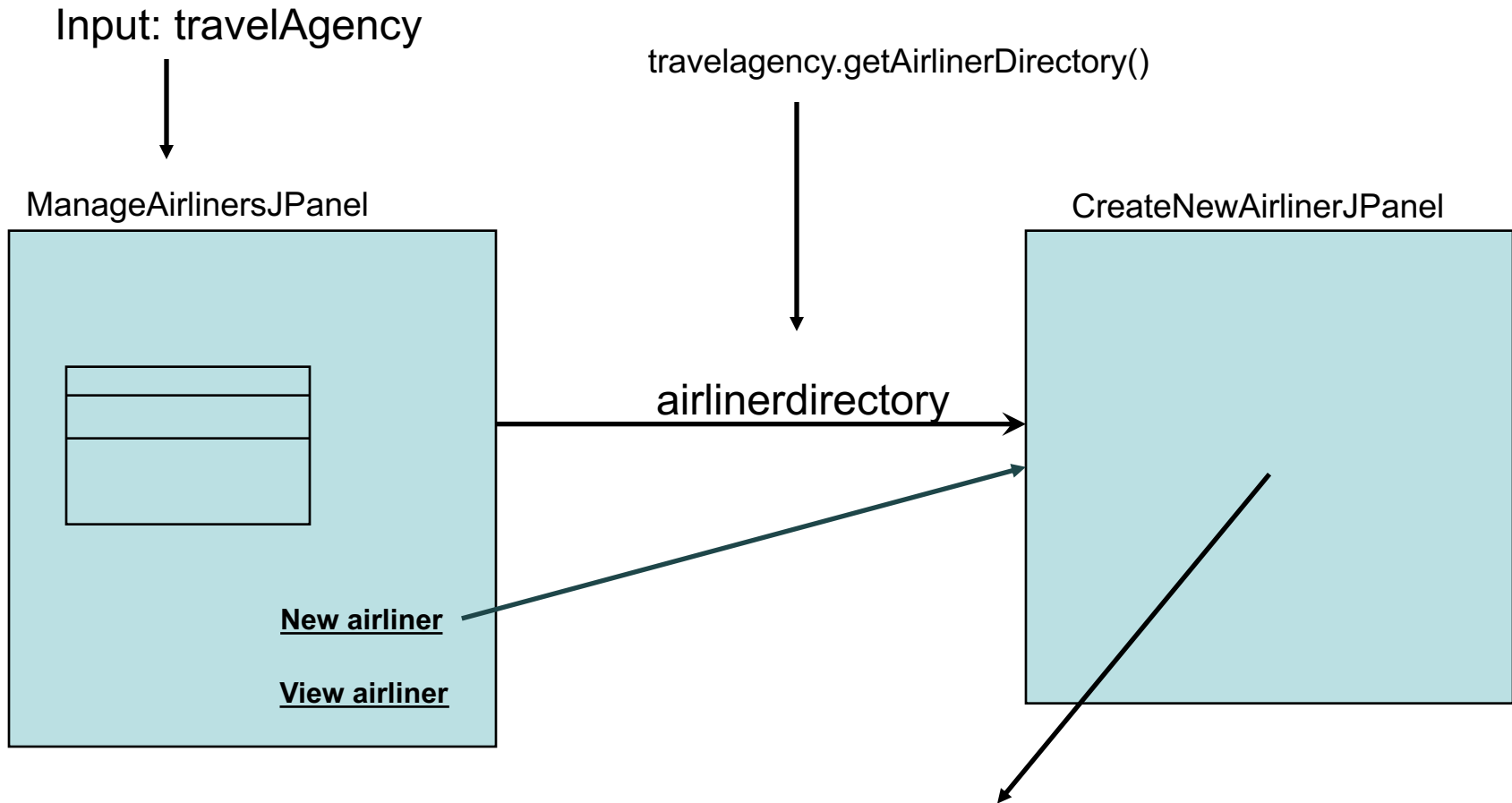
```
ManageAirlinesJPanel msjp = new ManageAirlinesJPanel (CardSequencePanel, ad);
```

```
CardSequencePanel.add("airlinersjpanel",msjp); //any name will do
```

```
((java.awt.CardLayout)CardSequenceJPanel.getLayout()).next(CardSequencePanel);
```

Insert this code as part of the perform action event associated with

How to passing parameters?



Uses airlinerdirectory object to create an airliner:

```
Airliner s = new airlinerdirectory.newAirliner();
```

Passing Parameters

Assume travelagency is an instance of the TravelAgency class

```
AirlinerDirectory sd = travelagency.getAirlinerDirectory();
```

```
CardSequenceJPanel.removeAll();
```

```
ManageTravelAgencyJPanel msjp = new ManageTravelAgencyJPanel (CardSequencePanel, sd);
```

```
CardSequencePanel.add("airlinersjpanel",msjp); //any name will do
```

```
((java.awt.CardLayout)CardSequenceJPanel.getLayout()).next(CardSequencePanel);
```

Passing the airliner directory as a parameter after it getting extracted from travelagency Object. The cardjpanel is also passed.

How to display the airliner table?

```
//Clear airliner table
```

```
int rc = airlinerjtable.getModel().getRowCount()-1;
    while(rc>=0) {
        ((DefaultTableModel) airlinerjtable.getModel()).removeRow(rc);
        rc = rc - 1;
    }
```

```
ArrayList<Airliner> airlinerlist =airlinerdirectory.getAirlinerList();
```

```
For (airliner aliner: airlinerlist){
```

```
Object[] row = new Object[5];
```

```
row[0] = aliner.getName();
```

```
row[1] = aliner.getAddress();
```

```
Row[2] = aliner.getTotalFlightsPerDay();
```

```
:
```

```
:etc
```

```
Row[4] = aliner; // this is an extra entry so you have a way to retrieving
```

```
    // the actual airliner upon user selection of the corresponding row.
```

```
//add row to supplier table
```

```
((DefaultTableModel) airlinerjtable.getModel()).addRow(row);
```

```
}
```

How to extract the user selected airliner from the airliner table?

- Define an action perform on the airliner table
- Insert the following code to isolate the selected airliner object

How to retrieve selected row?

```
int r = airlinerTable.getSelectionModel().getLeadSelectionIndex();  
  
    if (r > -1) {  
        Airliner aliner = (Airliner) airlinerTable.getValueAt(r, 4); // the 4th column  
                                                                    // is where we save the supplier object  
    }
```

The Airliner Use Case

Airliner name:

Go

Manage Flights

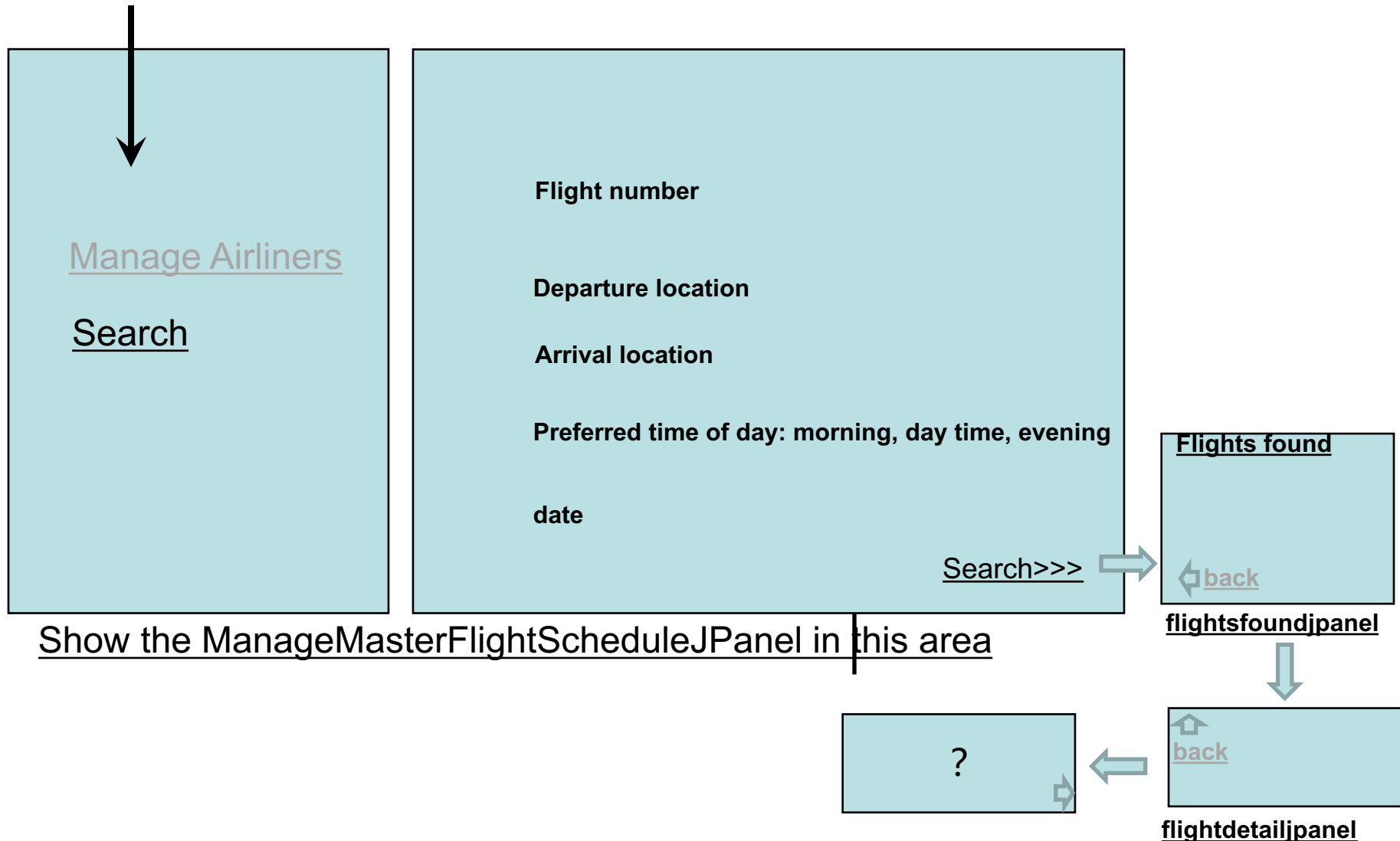
Flight schedule

New flight >>

View Flight >>

Show the ManageAirlinerFlightScheduleJPanel in this area

How to handle travel agency search for flights through the masterschedule



How to pass selected airliner?

