CMPT 270 – 2016-2017 Term 01

Assignment 5:
Graphic User Interfaces

Submission Deadline: Wednesday November 30 2016 at 11:55pm – Submit via Moodle!

Description

The objective of this assignment is to add a partial GUI front end to Assignment 4. It is will only be a partial front end, as it would take too much time to do a good job of the whole front end.

Deliverables

***** Upload .java files, .class files and .txt/.pdf files for documentation within one .zip file *****

Also upload an executable JAR file for your system. Check that your JAR file successfully executes before submitting it, as it is easy to have problems with the manifest file. Include the source files (as well as executable files) in your JAR file. The addition of the source files is not the default, so you need to select it. By default, the jar file should run the GUI program to be written.

We will run you JAR file to verify that they run as you state. You should also include a ZIP file that includes:

(i) your .java files

(ii) your external documentation

(iii) a listing of the new classes (just the new startup class, and the GUI classes)

(iv) screen shots showing all of the GUI components

The markers will give feedback on your work.

Graphical User Interface specifications

The entity and container packages should be unchanged. The classes in the commands package should not be changed, although you might need to add a new command. All these classes should be imported from my version of Assignment 4. The FlightReservationSystem class and the interfaces package will not be used, but should be included and should continue to fully contain a working command-line version of the program (three layer architecture allows both command-line and GUI versions to co-exist nicely with not much duplication). A new class will be added to the startup package to start up the event-driven application. This should be a small class to create the main window (JFrame) and make it visible. The other new classes are to be placed in a new package called 'gui'.

When the project first starts, a window should appear and give the user six options. Five of the options will be buttons to press, while the sixth will be a prompt and a text field. Each of these options will be discussed.

The first button will be one to initiate the task of adding a flight to the system. When it is pressed, a new window should appear. In this window, there should be prompts and fields for the entry of the flight's number, width and capacity. Provide a Submit button that will cause the data to be collected from the fields and passed to the command to add the flight. If the command fails, the error message should appear in the window, and the window remains visible for the user to correct the information. If the command is successful, the window should disappear, so that the user is returned to the main menu.

The second button is for the addition of a new passenger. Its affect will be very similar to the add flight button: a new window will appear for the entry of the data for the new passenger. This time, instead of a Submit button, the command to add a passenger should be created and executed whenever the user presses the Enter key in either one of the fields. If the addition of a passenger fails, the error message should appear in the window for the user to correct the data. If the addition was successful, the window to add a passenger should remain visible (with the fields blank) for the user to enter another passenger. The user can close the window via the X in the upper right corner, or you can supply a close button (the close button is optional).

The third and fourth buttons are to display the all passengers in the system, and to display all the flights in the system. In both cases, when the button is pressed, a new window should appear with the information. It should have a 2-D field to display the information, and scroll bars should be provided in case there is too much information for the field. This can be done by

```
JTextArea myTextArea = new JTextArea(textToBeDisplayed);
myTextArea.setEditable(false);  // the user is not allowed to edit the text area
JScrollPane myPane = new JScrollPane(myTextArea);
```

and put myPane into a suitable JPanel, or directly into a JFrame. The preferred size of the scroll pane may want to be set to a suitable size if it is placed in a JPanel. If a scroll pane is directly added into a JFrame, it will take on the size of the available window. Don't worry about formatting the text in a fancy way.

The fifth button is to terminate the project. The project can also be terminated by the X in the right hand corner of the main window. However, selecting the X in another window should just close that window (the default action for that (built in) button).

The sixth option in the main window are a prompt and a text field to handle an existing flight. When the user enters a number in the field (presumably the number of a flight) and presses the Enter key, a new window should appear. The flight number should appear in the title bar of the window, and a line near the top should also show the flight number and the remaining capacity of the flight. If the number is not a valid

number of a flight, the window should just show an appropriate error message. On the other hand, for a valid flight number, this window will show information about the flight, and provide the option to do any of the operations on the flight. However, such a window is too complex for the time available for you to do this assignment. Therefore, a class (`HandleFlightFrame`) will be given to you that handles this window. You should not need to change it, but you will need to design your code so that the frame is created and made visible at the appropriate time.
For the initial window, the six options can be arranged in any pleasing fashion. Don't spend too much time making it fancy, but it should be decent. The same applies to the other windows, although they are simpler (except for the `HandleFlightFrame` which is more complex).

Additional Guidelines

External Documentation

For external documentation, include the following:

(i) A description of how to execute your project.

(ii) The status of your assignment.

This is an individual assignment. You are encouraged to discuss the general concepts of classes, types, containers, etc. with you classmates, but the specific details of the Flight Reservation System in this assignment should be done completely individually. Students that copy / share work will be penalized.