Module 2 Assignment

Description

In this module, you will create a JavaFX user interface that edits a SeatReservation.

First Steps

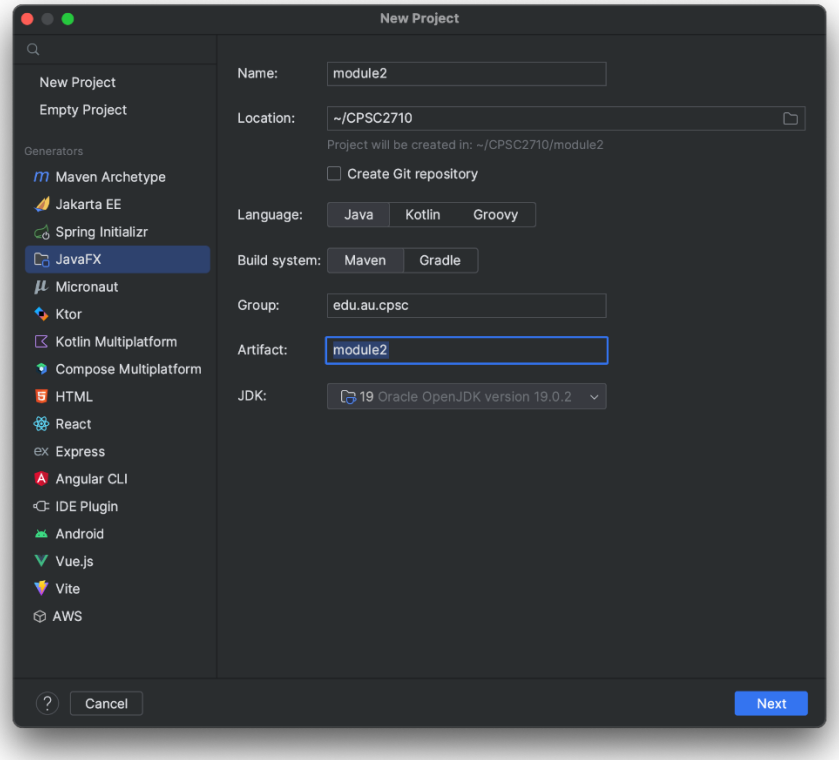
Using the command line, in your solutions directory:

1. 1. Switch to the main branch and merge in your module1 branch.
2. 2. Create (and switch to) a branch named module2.

If you haven't already, apply for a student license for IntelliJ. Create a new IntelliJ project using the JavaFX template. Name the project module2 and place it inside your solutions directory.
IMPORTANT: disable the 'create git repository' checkbox since you've already created a repository.

Here's what my 'new project' dialog looked like:

[Image of new project dialog would go here]



Delete the HelloController.java and helloview.fxml files. Rename HelloApplication.java to SeatReservationApplication.java. Copy SeatReservation.java from module1 into your module2 project's primary package directory (src > main > java > edu > au > cpsc > module2).

Changes to SeatReservation

Add a private int instance variable named numberOfBags with correctly-named getter and setter methods. Add a private boolean instance variable named flyingWithInfant with a getter named isFlyingWithInfant() and two setter-ish methods: makeFlyingWithInfant() -- sets i-var to true, makeNotFlyingWithInfant() -- sets i-var to false.

Update the toString() method adding these instance variables.

The Application Window

Create a graphical user interface to edit a SeatReservation. For each of the six i-vars in this class, add a control to display its value.

Use a CheckBox for the flyingWithInfant field, TextFields for flightDesignator, firstName and lastName and a DatePicker for the flightDate. Place a row or column with Cancel and Save buttons at the top, bottom, left, or right in the window.

Your layout should be as follows:

- • BorderPane
  - o Center: GridPane
    - □ Labels in the first column
    - □ Input controls in the second column
  - o Bottom: HBox (alignment of children set to top right)
    - □ Two buttons

In addition to the six controls, add a TextField labeled 'Number of passengers'. Make this TextField un-editable. Set its default contents to '1'.

Hooking Things Up

Create an instance variable called seatReservation of type SeatReservation in your application class. In your application's start() method, construct an instance and store it in this instance variable. Set all of seatReservation's instance variables so that they contain reasonable values.

Create a method called updateUI() that takes the contents of the instance variables of seatReservation and displays them in your controls. For example, you'll have several lines similar to this:

flightDesignatorField.setText(seatReservation.getFlightDesignator());

Call updateUI() from your start() method as almost the last thing that start() does, right before it makes the stage visible. At this point when you run your application it should display the flight information that you set up in start().

Add an event handler to the 'flying with infant' CheckBox. If that CheckBox becomes checked, display the number 2 in the 'Number of passengers' field. Display a 1 there when the CheckBox becomes unchecked.

When the Save button is clicked, populate the seatReservation instance variable by calling setters using the values in the input controls as arguments. Be sure to catch IllegalArgumentException and display an error message if it occurs. After populating the object, display it on the console. Exit the application using Platform.exit().

When the Cancel button is clicked, display the message 'Cancel clicked' on the console and exit the application using Platform.exit().

Submitting

Submit this assignment on Canvas. Paste the URL of your GitHub repository into the supplied input. This will be the same URL that you submitted in Module 1. Note that Canvas will try to display a preview of it but that preview may show an error since the repository is private. Don't worry, as long as your URL is correct, everything is OK.