

Assignment 5:

Graphic User Interfaces

Submission Deadline: 20 November 2017 at 11:55pm – Submit via Moodle!
(Note – late submissions will be accepted until 23 Nov. at 11:55pm without deduction, but no submissions will be accepted after that time)

Description

The objective of this assignment is to add a partial graphical user interface (GUI) front end to Assignment 4. It is will only be a partial GUI so that this assignment is not too long.

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.

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 annotate the .java files to provide written feedback on your work.

Graphical User Interface specifications

The `entity` and `container` packages should be unchanged. Although, the method `getDoctors()` was added to the class `Patient` so that the doctors of a `Patient` could be accessed in order to display them in a GUI window. You should not need to make any changes for your part. The classes in the `commands` package should not be changed, although if needed, new commands can be added. The `HospitalSystem` class and the `interfaces` package will not be used. A new class has been added to the `startup` package to start up the event-driven application. The new classes that you write are to be placed in the package `gui`, along with the GUI classes supplied to you.

When the project first starts, a window should appear for the user to enter the data to create a ward. It has fields (with prompts) for the entry of the ward name, the first bed label, and the last bed label. In addition, there is a submit button to submit the information to the system. After the submission, another window will appear. In the version of the system given to you to start with, the window to appear is the one for the operations dealing with patients (see the **Patient operations window** section below for more detail). This window has four options. The first one is to add a new patient. The second option has a text field for the entry of a health number. When one is entered and the Enter key is pressed, a new window (a patient's window) is opened with information on the patient, and operations on the patient. The third option is to list all the patients currently in the system. The fourth option is to close this window. This part of the GUI system is given to you to use.

You need to modify the system so that after submission of the ward information, a different window opens (the main menu window). It should give the user four buttons to press, as explained below:

The first button of the main menu window will be one to initiate operations on patients of the system. If it is pressed, a patient operations window appears. This is the same window as presently opens with after the submission of ward information.

The second button of the main menu window will be one to initiate operations on the doctors of the system. If it is pressed, a new window (the doctor operations window) appears with four options. The first option is to add a new doctor. This option should create a new window (doctor add window) where the data for a new doctor is entered, and a button is pressed to submit the data to the system to add the new doctor. The second option of the doctor operations window has a text field for the entry of a doctor's name. When a name is entered and the Enter key is pressed, a new window (a doctor's window) is opened with name and specialty (if any) of the doctor. The doctor's window should have three options, and an exit button. The first option is to add a patient to the patients of this doctor. The second is to access a specific patient of the doctor. This option should open the same patient's window as accessing a specific patient from the patient operations window, as mentioned in the third paragraph back. The third option of the doctor's window is to remove a patient from the patients of this doctor. The exit button of the doctor's window should cause this window to close. Note that the doctor's window does not display all the patients of the doctor, as it is anticipated that normally a doctor would have too many to list the all. This is in contrast to a patient's window where all the doctors of a patient are listed. The third option of the doctor operations window lists all the doctors in the system. For simplicity, they can be listed in a `JOptionsPane` with default formatting obtained from the dictionary. The last option in the doctor operations window is a button to close the window. Note to be careful to distinguish between the doctor operations window and a doctor's window, as they are two different windows

The third button of the main menu window is to display the information on the ward. It will list each of the beds. If the bed is empty, there will be a field to enter the health number of a patient to be entered into the bed. If the bed is occupied, there will be a button to cause the patient to be removed from the bed. The bottom of this window will have a button to close the window. The ward windowing part of the

GUI system is also given to you. In the version of the system given to you, it can be accessed by pressing the assign button in a patient's window.

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

Additional Information

For the main menu window, the four 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 that you develop.

As given to you, there are two main methods that can be used to start the system. The first main method is in the class for the frame of the patient operations window, the `PatientOpsFrame` class. The second main method is in the `GuiHospitalSystem` class, and it first creates the window to enter the **Ward** information, and then again opens the patient operations window. As stated above, you are to change this so that the main menu window opens after the submission from the ward creation.

Patient operations window : This window has options to add a new patient to the system, access a specific patient, list all patients, and close the window. If a specific patient is accessed, the window that appears has the patient's information. This window is called a patient's window, as distinct from the patient operations window just discussed. A patient's window allows certain operations on the patient. In particular, if the patient is not presently in a bed, an assign button is in the window that opens the ward window. In the ward window, the user can assign the patient a bed. Returning to the patient's window, if the patient is already in a bed, then a remove button exists to remove the patient from the bed. The patient's windows lists all doctors of the patient (each of which can be removed), and has a textfield to add a doctor to the patient's list. However, in the version of the classes as given to you, there will be no way to add doctors to the system, so you cannot add a doctor to the list of doctors of the patient. Of course, you will be correcting this problem. It is suggested that you first get a basic version of the window working for the main menu, and have it appear after the ward creation window. In the main menu window, first get the button working for the patient operations window. Next, the button for the ward window can be done, and the button to exit the system. After these are working, then start working on the doctor operations window and the button of the main menu to show it. You can start with only the close button in the doctor operations window, and then add the others. Note that the doctor operations window will be very similar to the patient operations window, and the classes for the latter are given to you. Then, you need to build the window to add a doctor, and the window that appears when a specific doctor is accessed. It is suggested that initially you don't worry about error situations.

Additional Guidelines

Submission

You are expected to submit the jar file for your system to Moodle. Be sure to specify to include the source code in the jar file (the default is not to include it). The JAR file is to be executable. We will run your project, so ensure that the JAR file properly executes - you should check it before submitting it. In addition, you are to hand in a zip file that includes (i) all of your .java files, and (ii) the external documentation for the assignment.

External Documentation

For external documentation, include the following:

- (i) A description of how to execute your project.
- (ii) The status of your assignment.
- (iii) Diagram 1: The first diagram should show how you laid out the main menu window. It should show how you used GUI components in order to place the widgets in the window in appropriate places. It is a diagram showing what the components do, not the code to lay out the window. It will be similar to the one drawn in class.
- (iv) Diagram 2: The second diagram should be how you laid out the window for adding a new doctor.

Marking (Total 40 marks)

- (5) External documentation
- (4) the JAR file successfully runs stand alone
- (24) GUI
 - (4) handling errors
 - (3) Readable code

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.