UNIVERSITY OF SASKATCHEWAN

Department of Computer Science
176 Thorvaldson Building
110 Science Place, Saskatoon, SK, S7N 5C9, Canada
Telephine: (306) 966-4886, Facimile: (306) 966-4884

Assignment 5

Graphical User Interfaces

Date Due: November 18, 2019, 11:55pm **Total Marks: 70**

General Instructions

- **This assignment is individual work.** You may discuss questions and problems with anyone, but the work you hand in for this assignment must be your own work.
- **Assignments are being checked for plagiarism.** We are using state-of-the-art software to compare every pair of student submissions.
- **Make sure your name and student number appear at the top of every document you hand in.** These conventions assist the markers in their work. Failure to follow these conventions will result in needless effort by the markers, and a deduction of grades for you.
- **Assignments must be submitted to Moodle.**
- **Moodle will not let you submit work after the assignment deadline.** It is advisable to hand in each answer that you are happy with as you go. You can always revise and resubmit as many times as you like before the deadline; only your most recent submission will be graded.
- **Programming questions must be written in Java.**
- **Non-programming questions must be submitted as either** `.txt` **or** `.pdf` **files.** If other file types are used, you will receive a grade of zero for that question.

Preamble

- 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.
- It is highly recommended that you use the classes provided in the Assignment 5 folder as your starter code.
  - All the command classes have been refactored in the provided code.
  - The `Patient` class has a new method called `getDoctors()` which allows you to get all the doctors associated with a patient
- The entity and container packages should be unchanged.
- 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 called `GuiHospitalSystem` 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.

Question 0 (2 points):

Purpose: Git for Bonus Marks **(optional)**

Degree of Difficulty: Easy

As a reminder, you have the opportunity to gain some bonus marks by using git/version control on this assignment. The choice to use git/version control is up to you. This is an incentive program - using git will result in bonus marks, not using git will have no effect on your final grade. You can receive up to 2% extra on your final grade (but you cannot exceed a grade of 100% in the course). Additional details and instructional videos can be found in the folder titled "Git for Bonus Marks" on moodle.

What to Hand In

- Your git-log of Assignment 5

There is a separate submission folder on moodle "Assignment 5 Git Log" for you to upload your git log. You must upload your git log here if you want to bonus marks. Uploading it to the regular Assignment 5 submission folder will result in your git log not being graded.

Evaluation

2pts git logs will be given a grade from 0-2 where:

- 0 indicates that your attempt at using was not good enough to be considered for bonus marks
- 1 indicates an attempt was made but you need to improve the number of commits or the content of your commit messages
- 2 indicates a non-trivial use of git

Question 1 (50 points):

Purpose: Practising GUI design and Implementation.

Degree of Difficulty: Tricky

When the application first starts, a window should appear for the user to enter the data to create a ward (if you downloaded everything correctly this should already be done). 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.**

Modify the system so that after submission of the ward information, a different window opens (the main menu window). You need to design and implement this window. This window should have the following four components:

1. This button will initiate operations on patients of the system. If it is pressed, a patient operations window appears (this is the window that presently opens with after the submission of ward information).
2. This button will initiate operations on the doctors of the system. If it is pressed, a new window (the doctor operations window) appears. You will need to design and implement this window. It should meet the following specification:
   - A button 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. You will also have to design and implement this doctor add window.
   - A text field for the entry of a doctor's name. When a name is input and the Enter key is pressed, a new window (a doctor information window) is opened that displays the doctor's name and specialty (if any). The doctor's window should the following components:
     - A TextField and button to add a patient to the doctor's list of patients.
     - A TextField and button 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.
     - A TextField and button to remove a patient from the current doctors list of patients.
     - A button to close the window.
   - A `JOptionsPane` that lists all the doctors in the system. For simplicity, they can be listed in the `JOptionsPane` with default formatting obtained from the dictionary.
   - A button to close the window.
3. This button is used to 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
4. This button is used 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).

What to Hand In

- A file titled `A5.jar` of your complete system.
- A zip folder titled `A5Q1.zip` that contains all the GUI classes you created

Be sure to include your name, NSID, student number and course number at the top of all documents.

Evaluation

10 pts For the jar file (should successfully run and be stand alone)

10 pts For each new window

Question 2 (20 points):

Purpose: Practising External Documentation

Degree of Difficulty: Easy

Once you have completed the questions above, create a file called `A5_documentation.pdf` that includes all the internal documentation for the refactored system. External documentation should include:

- A description of how to run your system. What class should be invoked, what method?
- The status of your assignment. What is working and what is not working? What is tested and what is not tested? If it is only partially working, the previous point should have described how to run that part or parts that work. For the part or parts not working, describe how close they are to working. For example, some of the alternatives for how close to working are (i) nothing done; (ii) designed but no code; (iii) designed and part of the code; (iv) designed and all the code but anticipate many faults; or (v) designed and all the code but with a few faults; (vi) working perfectly and thoroughly tested.
- Instead of a UML diagram, include the following diagrams (these can be hand drawn and scanned):
  - A diagram showing 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.
  - A diagram showing how you laid out the window for the doctor operations. It should show how you used GUI components in order to place the widgets in the window in appropriate places.
  - A diagram showing how you laid out the window for adding a new doctor. It should show how you used GUI components in order to place the widgets in the window in appropriate places.
  - A diagram showing how you laid out the window for the doctor information. It should show how you used GUI components in order to place the widgets in the window in appropriate places.

What to Hand In

- A file titled `A5_documentation.pdf`

Be sure to include your name, NSID, student number and course number at the top of all documents.

Evaluation

4 pts For the first two required sections.

4pts For each required diagram.

Additional information

Window Design: For the new windows you will create, the components can be arranged in any pleasing fashion. Don?t spend too much time making it fancy, but it should be decent. They will have to match the design diagrams you submit as part of your external documentation.

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. 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.