Amira Mohamed (219311869)

Addressing the reported bugs:

There was no bug reported for my user stories. All the test worked as expected.

Refactoring:

VitalSignsDisplayGUI

Separation of Input and Display Panels:

I separated the input field and the display area into two separate panels, named “inputPanel” and “displayPanel”. The “inputPanel” contains the patientID label, patientID field, and the submit button. This separation improves the readability of the interface when new input is entered. With this separation, when we clear out the display panel’s components using “displayPanel.removeAll()”, the input panel remains intact, providing a clearer user experience and allowing for further input without disruption.

Removal of Main Method:

I removed the main method from this class, making it accessible and runnable only from another class, specifically is the “PatientGUI” class. This change adheres to the Single Responsibility Principle (SRP) by delegating the responsibility of application startup to a dedicated class.

Addition of Return Button:

I added a return button to allow users to navigate back to the patient portal, enhancing the overall user experience. This addition provides convenient way for the user to return to the main patient portal, facilitating seamless interaction with other features in the patient portal after retrieving the vital signs.

These refactoring enhance the overall readability and maintainability of the class by improving the organization of its component and delegating responsibilities appropriately.

PatientGUI and DischargePatientGUI

Refactor in PatientGUI:

- Modified the “PatientGUI” class to check if there are any patients in the hospital before enabling the discharge patient functionality.
- Displays an informative message if there no patient in the hospital to discharge.
- I ensured that each method have one specific functionality that it is addressing and added comments to explain what each method is for.

PatientGUI has undergone refactoring to enhance the "Display All Patients" feature, now specifically displaying patients assigned to the physician rather than all patients in the hospital. Additionally, the "Get Patient by ID" functionality has been tailored to retrieve information solely for patients assigned to the physician.

Refactoring in DischargePatientGUI:

- I modified the code so that only patients under the physician get displayed in the drop-down menu.
- I modified that code that it checks if there are still patients left in the hospital. If there are, they will be added to the list of choices. If not, the discharge functionality is disabled.
- If all patients get discharge, the code will display a message indicating that no patient left in the hospital and the discharge button get disabled.

These refactorings enhance code quality by implementing checks to ensure that discharge patient functionality is only enabled when there are patients in the hospital. This approach improves user experience and prevents error or confusion when trying to perform action without the presence of patients.

VitalSignsGUI

- In this class, I refactored the patient ID validation method to check if the patient is in the database based on their ID.
- Previously, the method was only checking whether a correct ID is entered (i.e. not a negative number), but now it confirms that the patient exists in the hospital.
- I added a “clear” method to clear the test fields after it get recorded.
- I refactored the patient ID validation method to check if the patient is in the database based on their ID. The method was only checking whether a correct ID is entered (i.e. not a negative number), but now it confirms that the patient exists in the hospital.

By this refactoring, we can confirm that the patient is in the hospital before recording their vital signs.