CSE 2050 – Programming in a Second Language

Fall 2023

Homework 5: GUI with PyQt

Total Points: 30

Date Assigned: Wednesday, Nov 8, 2023 **Due Date:** Sunday, Nov 19, 2023

Instructions: Please submit your work on Canvas as a zipped file named `cse2050_your_full_name_hw5.zip`. Make sure to include a main.py file as the driver for testing your program and organize all classes into separate files. You may design your UI using the PyQt Designer or write all of the code by hand. Please ensure that your code obey PEP 8 standards and the best practices discussed in class.

Key Concepts Demonstrated

- Writing GUI programs based on the Objected Oriented Programming (OOP) Paradigm
  - Writing code that uses objects that interact with each other
  - Creating a GUI using PyQt

1. (30 points) **Find me a Doctor**

Given an SQLite database file named `health_first_florida.db` (locate it on Canvas) containing information about medical doctors in Florida, please complete the following:

Write a PyQt program that features a UI that allows the user to navigate the list of doctors in the database as well as to search for a doctor using specialty or name of doctor. Specialty may be a sub-string of a full specialization (e.g., "Oncology" should also return doctors who specialize in "Gynecologic Oncology"). In like manner, the name of the doctor may be partial or full (for example, "John" should also return records for "John Perry").

Requirements:

1. Your program must be written using the OOP methodology. All classes should have a constructor that initializes instance variables. You may use input fields or labels to display information. If you use input fields, the information in the field should be read-only.
2. Searching for a doctor or specialty should open a dialog showing a list of doctors. This list of doctors could be displayed in a combo-box or a list of links. Clicking on a doctor's name should populate the parent UI with information about the doctor.
3. The UI must feature a *next* and *previous* button, which should be used to navigate the list of doctors. The functionality of the *next* button must be coded using the Python `__next__` special method and the status-bar should show the index of the current doctor being displayed.

Please see a picture of the expected UI in the Figure 1 below. Notice the fields and the kind of information expected for each doctor.

Figure 2 shows an example of the dialog populated with results for a given search string. Clicking on a result in the search list and clicking the OK button should update the UI with information about the doctor.

Figure 1: Example UI for the Doctor Search Application

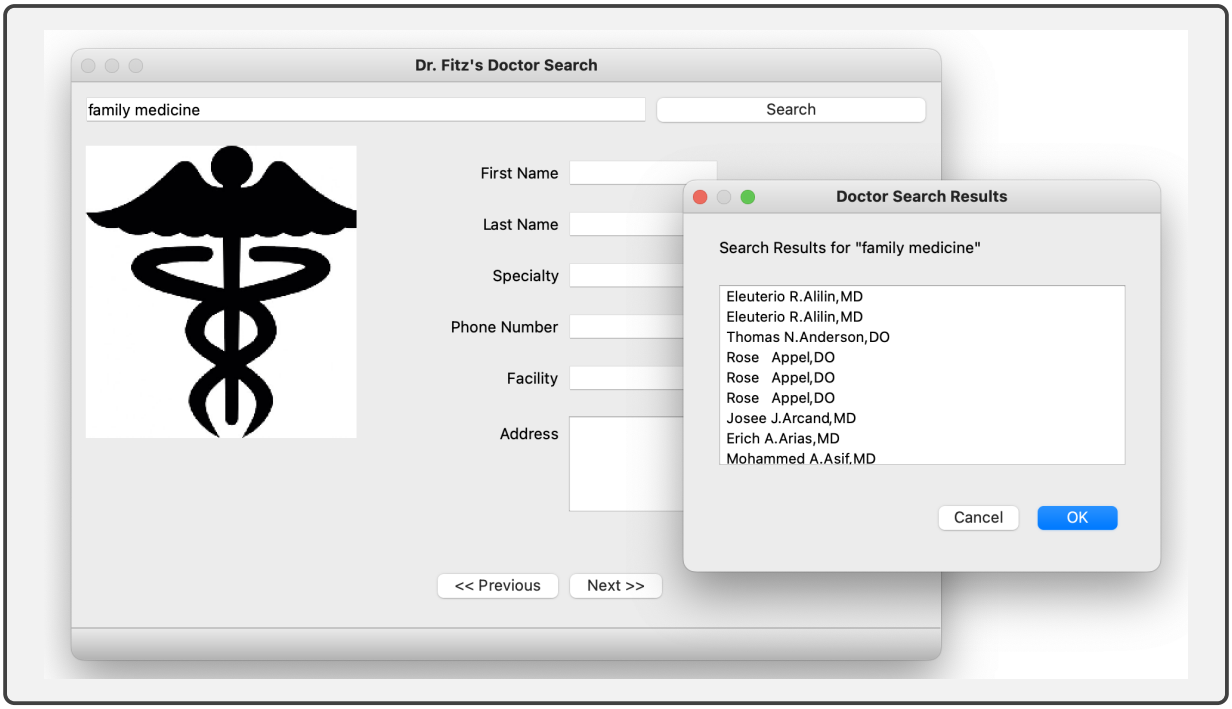

Figure 2: Example Search Results Dialog