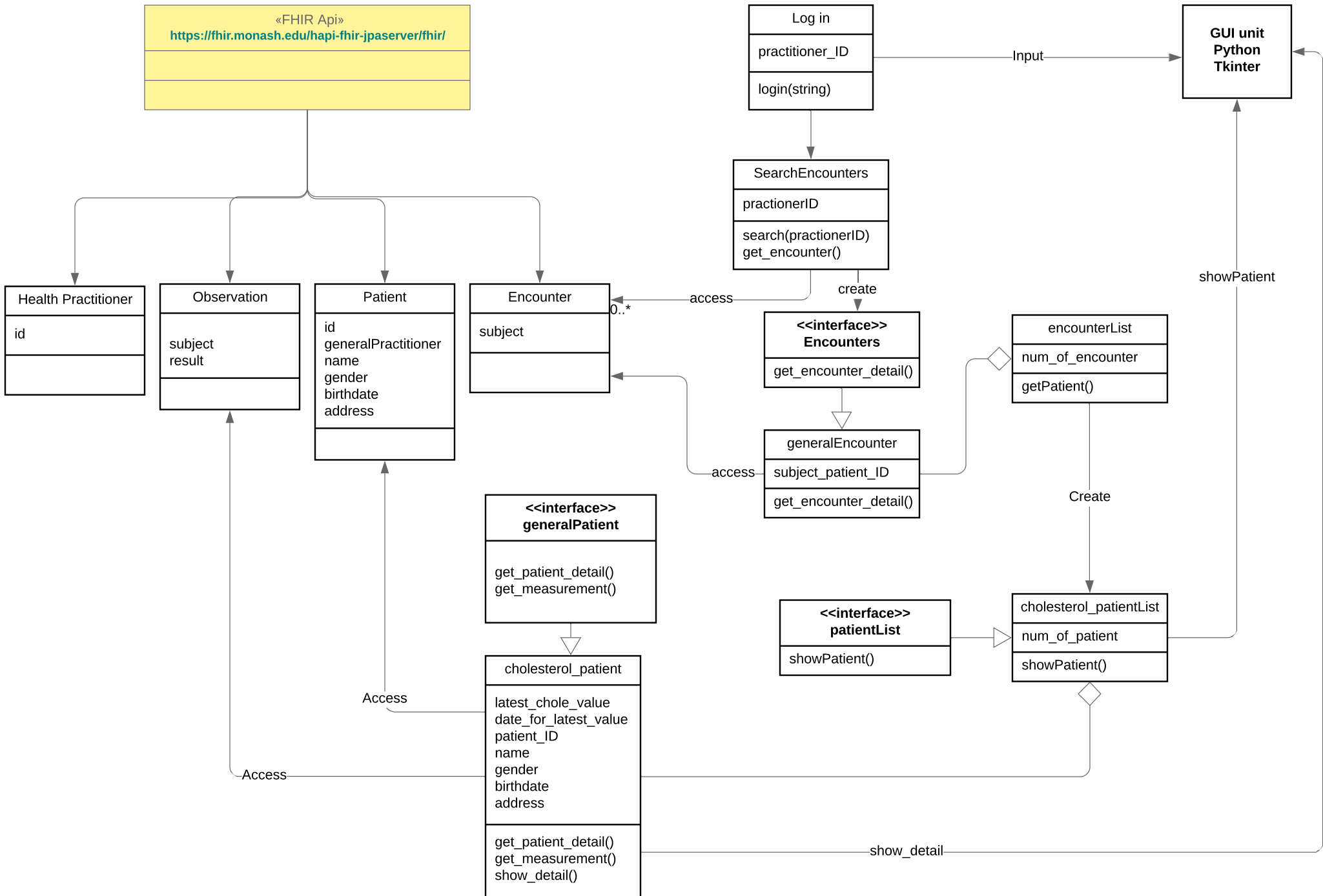


FIT3077 Assignment 2

Junxuan Liang | May 20, 2020



Design principles

- Open/Close principle
 - Interfaces are used for the encounters and patients to be included in the patient list and encounter list classes. In that way, the program can be easily extended if different kinds of patients and encounters are willing to be monitored without modifying the patient list and encounter list classes.
- Liskov Substitutability Principle
 - There are two subtypes in the design which are “generalEncounter” being a subtype of encounter and “cholesterol_patient” being a subtype of generalPatient. These two interfaces have methods to get detail from the API which can be done by their subtypes as well. For “generalPatient” interface it has a method to get different measurement which can also be done by the “cholesterol_patient” class to get the cholesterol value for a patient.
- Dependency Inversion Principle
 - In the design, using of interface allows the connection between encounter list and encounters, patient list and patient to be not directly. Thus, patient list and encounter list do not need to worry about the details about encounters and patients classes which allows easy modification in the future.
- Acyclic Dependencies Principle
 - No dependencies between packages form a cycle in the design.
- Stable Dependencies Principle
 - Most classes in the design are stable as they only depend on one or two classes. Thus a good thing when modification are needed for extending the program as we can limit the number of modifications.

Design patterns

- Iterator
 - An iterator is used to look encounter entries when a practitioner ID is entered as there is a limit for the FHIR API on accessing the number of items at once. The iterator can help to get rid of the problem.