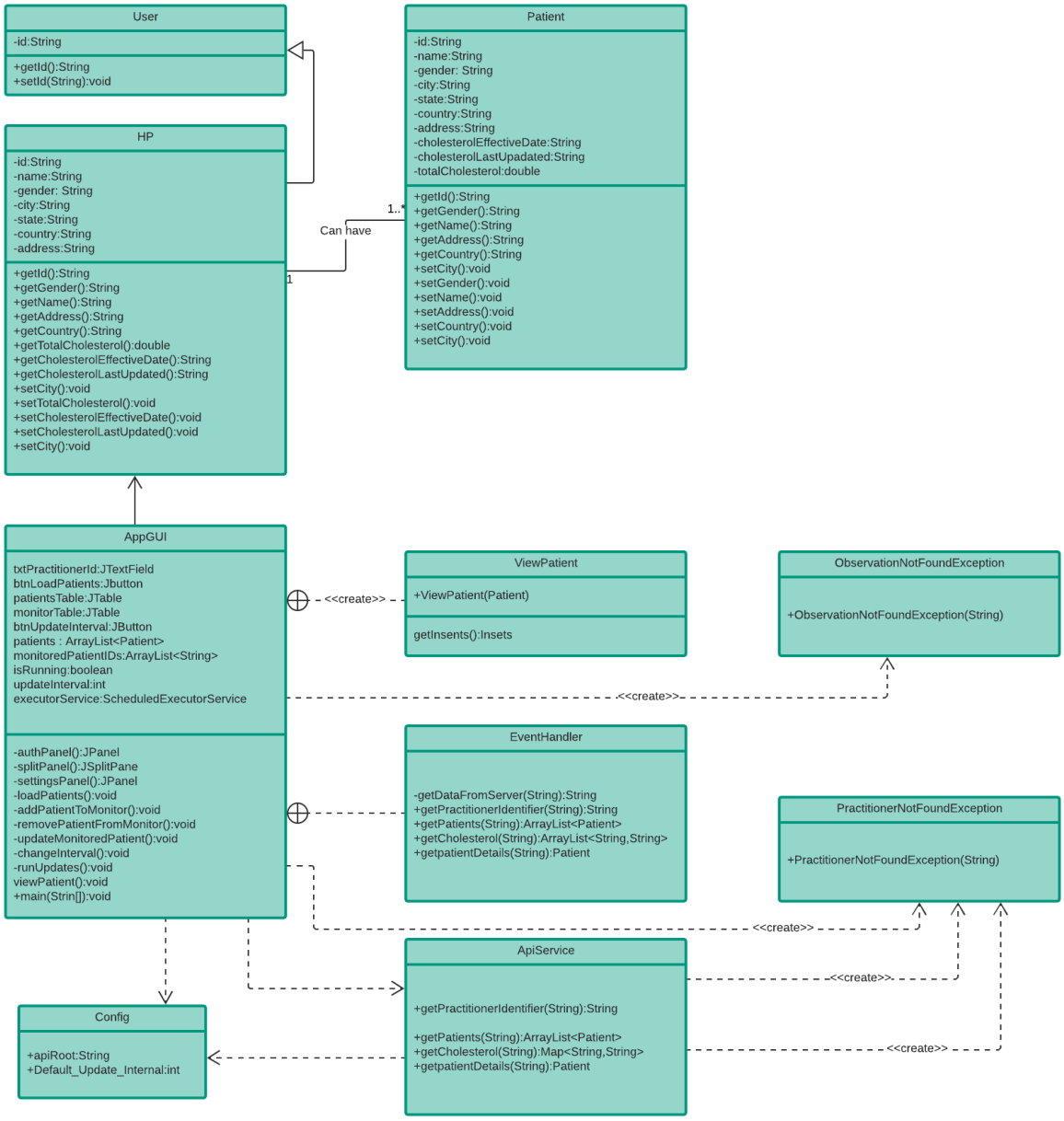
UML CLASS DIAGRAM



SOFTWARE DESIGN

This analysis provides an understanding in terms of design standards chosen to develop an application that helps health practitioners monitor the total level cholesterol of their patients. Design patterns and approaches were implemented to provide with a way to solve issues related to software development using a proven solution.

The adapter approach was implemented on the practitioner to provide a solution on when different user wants to use the application and are not compatible and interfaces cannot interact.

Open close principle: The system was designed in a way that few changes wouldn't need to modify the whole code but add functionalities, this principle saved the team falling into a fragile design. The ApiService class is the code base that shows a good implementation of the open close principle because it can be used to extend code and add functionality by adding classes that inherit methods from the ApiService class.

Top Down approach was implemented when designing the system focusing on the system requirements which helped make a design responsive according to its requirements. The system is split into sub-systems and components. The same is done with each of the sub-system. The main top part is the AppGUI class that has other sub classes and components that are also broken down, this process continues until it gets to the lowest level which are the exception classes in this system.

In order to prevent running very expensive computation the team splitted the patient details and full information only when requested because we assumed that getting patient list using a loop would require multiple calls to the server within a very short time and that is bad engineering.

The team solution offers a highlight consistent modules with minimal coupling and leaves a space to easy adapt on future system requirements which makes it easier to maintain.