An online version could be found at https://www.patmonitor.online/

FIT3077 Software Engineering Architecture and design S1 2020

Assignment 3 Design Rationale

Team JustLookAtMyInterface
Khai Dinh Nguyen - 29010691
Matthew Hoang Viet Pham - 29667437

1. Data Source

From Assignment 2, we incorporated the Dependency Inversion Principle by creating a DataSource interface. By creating an abstract interface for the Store to depend on, we were able to introduce a hinge point that allows the source of the data to change.

In addition this allows us to employ the Open/Closed Principle, as the `DataSource` interface allowed us to easily extend functionality to include the fetching of blood pressures. By modelling the Data Model classes closely to their representation on the FHIR server in assignment 2, we were able to reuse `Observation` to hold our blood pressure readings.

In regards to Package Level Design Principles, the Common Reuse Principle was considered when creating the `DataModel` package. As these classes model elements from a medical record system and contain many composition relationships, these classes will be reused together.

The Patient how contains two kinds of clinical data: Cholesterol level & blood pressure, so we decided to **generalize** them into a new class: `ClinicalData<T>`. This has 3 properties:

- Whether it is monitored
- Whether it is loading
- The fetched data

By doing this, we maintain a stable interface for the views to display the clinical data. This also makes it easier to add new kind of clinical data (e.g. vital signs, etc..)

2. Data Store (Redux Store)

There is nothing new about the store's structure. We just altered the actions to match with the new `DataModel` design.

3. UI Views

The view class `BloodPressureHistoryCard` is reusable which takes a patient as input. This makes it easy for `BloodPressureHistoryMonitor` to display multiple patient history view simultaneously.