Redesign Report

R05922130 王瀚磊

1. 簡介

   在這兩次的作業中，我接採取一樣的做法。先想好怎麼寫後先創建好每一個 class 檔案，並且在每個檔案中都用 TODO 的方式先寫好每個 class 應該做些什麼事情後再開始時做，在第一次作業耗費了四個小時寫完，第二次作業耗費兩個小時完成。

2. 變動說明：

   A. 在第一次作業中，我把所有 sensor category 皆視為同一個 class，僅利用其中一個 attribute 存取不同 sensor 之 category name 以及 name。且第一次作業我將所有的 database 在收到 input 後隨即把所有的 factor 讀取出來存在 list 之中，之後再做 monitor 的時候才一個一個叫出來處理。

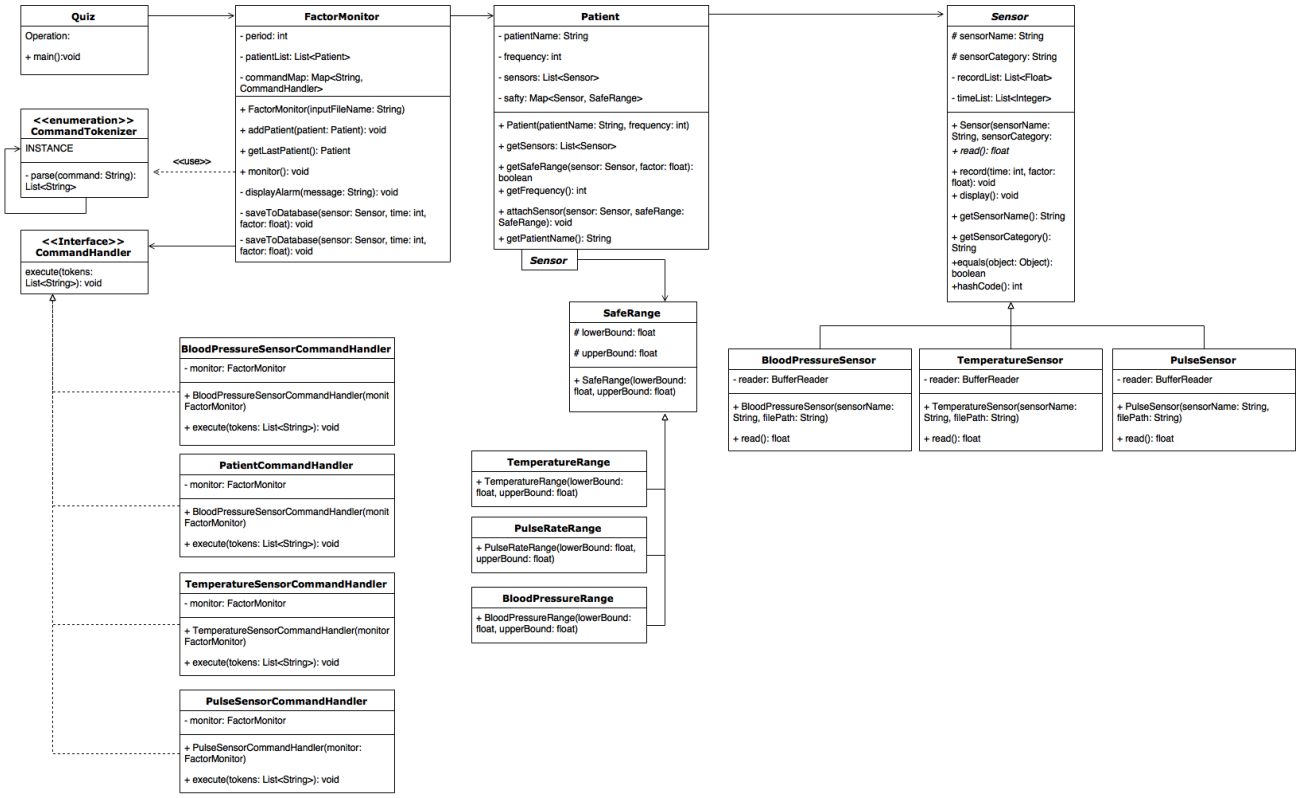
   B. 在第二次作業改進中，我將不同 sensor category 皆創立其專屬之 class，且我也將 sensor 讀取 database factor 的方式改為 monitor 時才一行一行讀取。另一個比較不同的地方是 saferange 也創立其專屬之 abstract class，並由不同 sensor category 之 saferange class 去繼承它。

   C. 這次改動要求我們照著老師所提供的 class diagram 寫程式，然而我發現其中有些部分並沒有寫得很清楚，因此造成在 implement 之中有些困難，必須再根據自己的想法揣測哪些 class 需要增加哪些 operation，或是一些參數。最後的結果如圖一所示。

3. 心得

   兩次的架構其實變化不多，也因為再第一次架構的設計時有多想一下，因此在接下來的變動中並不會造成太大的困難。這是我第二次利用所學的 design pattern 做一些實際例題，我所應用的包涵 command pattern 以及 singleton pattern。我利用 command pattern 處理所有 input 之 command type，例如:patient, BloodPressureSensor…等等，並且利用 singleton 創立一個專門拿來處理 command tokenizer 之部分。

   在這次作業中我更加深刻的體悟到利用一些 design pattern 撰寫 code 時，可以更多元的考慮一些 over design 的部分，例如未來要是增加一些 command 只要再多建立幾個 class 即可。然而這邊的 tokenizer 我有點不確定寫成 singleton 是否是對的，因為最後只有在 FactorMonitor 這個 class 中有使用到，除非未來的改動中有造成其他 class 也會用到，否則我認為我這樣的設計是不好的。



圖一、程式改進後之 Class Diagram