

# COMP0004 Coursework Report

Kang Shen

March 31, 2023

In this project I implemented a Java web application for healthcare institutions to store and manage their patient records. This application is capable of storing, accessing and modifying patient records and can therefore improve the work efficiency.

## 1 Features

This application implements the following features for managing patient records:

- The application stores each patient record as one list of items including their information and profile picture.
- The application stores all the patient records using one list for each patient.
- Patient records can be created, deleted, renamed and viewed.
- Information within a patient record can be viewed, added, deleted or edited.
- While viewing the patient list, each patient name is linked to the list of patient information, so patient's record can be accessed by directly clicking on the patient name in the patient list.
- A search function is implemented so that patient records can be queried by searching for relevant information such as ID or Birth Date.
- Patient records are automatically saved to a .csv file, so that the user does not have to manually load or save the information to a file.
- When a patient record or information inside it is modified, the changes are immediately written to the .csv file.

## 2 UML Class Diagram

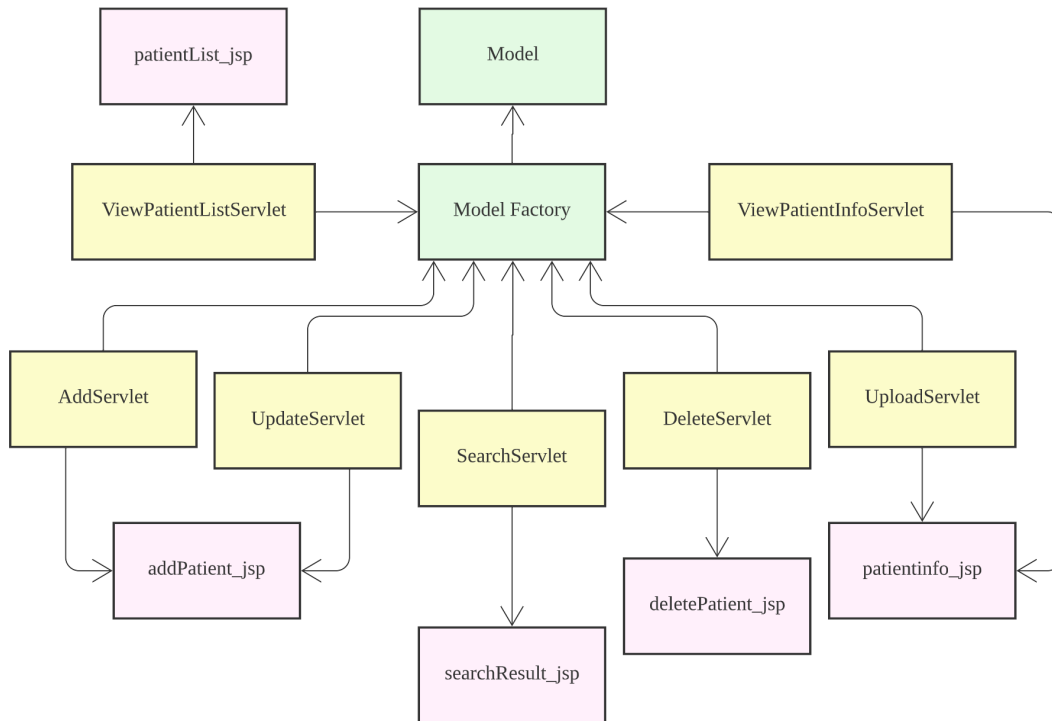


Figure 1: UML Class Diagram

This UML Class Diagram describes the class relations in this application. The overall design of the application follows the Model-View-Controller (MVC) pattern and in this UML diagram colouring is used to help distinguish between the classes. Green is used to represent the Model classes, Red is used to represent the View classes and Yellow is used to represent the Controller classes.

### 3 Application Design

In this application, Java servlets are used to handle user requests. When the user clicks on a link or button, the request is passed to a Java servlet which then forwards the request to a relevant Java Server Page (JSP) which generates the actual web page. The Java servlet retrieves the model reference via the Singleton pattern from the ModelFactory class and calls methods on the model to carry out the actions required by the request. The servlet then forwards the request to a relevant JSP where the results are displayed.

The Model-View-Controller pattern is strictly followed in the development process of this application. The ModelFactory acts as the Model in this case as it returns the reference to the model via the Singleton pattern. The JSPs act as the View as they are responsible for displaying the final result pages. The Servlets are Controller in this case because they receive the user request, carry out the actions accordingly, and finally forward the results to the JSP.

In the development process of this application, good OO design practice is achieved. All data manipulations on adding, deleting or editing patient records are stayed within the model class. The servlets carry out the actions through calling the model class instead of directly accessing the records. The Java Server Pages are used properly for displaying the result page from the Java servlets. Moreover, private fields, for example patient list, and private methods, such as delete patient profile pictures are used in the model class which prevents any potential illegal access from outside the class.

To conclude, the overall quality of this application is good because it implements many useful functions that helps efficiently managing patient records. Although further work could be done on the user interface, all the functions provided are clear to understand and use.