

# EH102 Basic EMR Project Specification

---

## Overview

You are to develop a very basic EMR system with the following functionality:

- Create a new patient or user and input their details
- List a summary of all patients and users in the system
- View all the details of a specific patient
- Find patients based on their name
- Delete a specific patient or user
- Password protected logins for users

## Framework

You are provided with the main application Java class (see *JEMR.java*) – this is essentially a driver which will run your EMR, provided that it implements the BasicEMR interface (see *BasicEMR.java*).

```
public interface BasicEMR {  
    public String getWelcomeMessage();  
    public boolean load();  
    public boolean save();  
    public boolean authenticateUser(String login, String password);  
    public void addPatient();  
    public void addUser();  
    public void listPatients();  
    public void listUsers();  
    public void viewPatient(int id);  
    public void findPatients(String name);  
    public void deletePerson(int id);  
}
```

You can then modify the main method of *JEMR* so that it instantiates your new EMR class, e.g.

```
JEMR app = new JEMR(new MyNewEMR());
```

## Data Storage

Your EMR system should store the user and patient records in two separate Comma Separated Values (CSV) files. For example the patients file will contain lines like

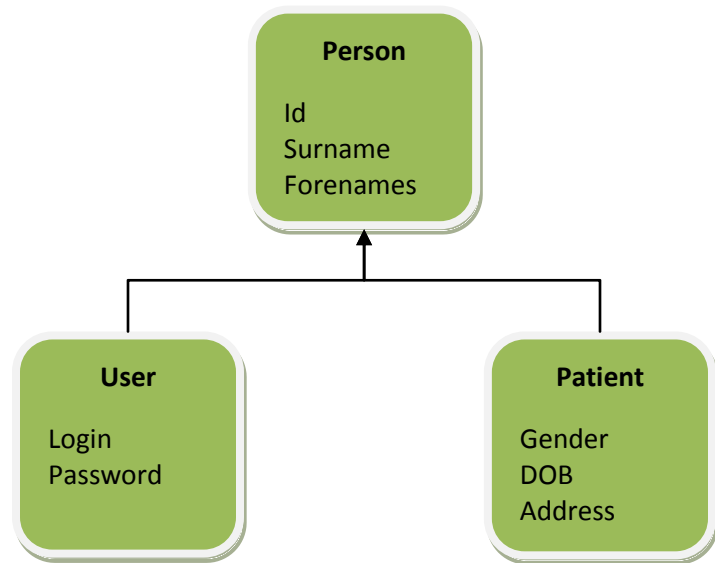
```
1,SEYMOUR,Rowan Patrick,MALE,28/05/1981,Remera  
2,DERIGGI,John,MALE,05/01/1980,Kimihurura
```

Because the CSV format uses commas to separate values, it is important that the values do not contain commas themselves. These should be removed from fields such as address, before saving to the file. You can double-click on any CSV file to open it in OpenOffice.

## Data Model

Your EMR should be based on this data model, though you may add additional fields if you want. Fields should be properly encapsulated with getter and setter methods.

The users and patients can be stored in separate collections within your EMR.



## Additional Requirements

1. Use *SimpleDateFormat* to parse inputted dates, and handle the parse exception if the format isn't correct
2. Use *TreeSet* to store users and patients. You will need to implement the interface *Comparable* on your Person class, but that will mean that all persons are automatically sorted in the *TreeSet*.

## Extra Credit

1. Modify the *JEMR* class and *BasicEMR* interface so that find operation takes two parameters – the field to search in, and the value to search for, e.g.  

```
find address Remera
```
2. Add a field to User to show if a user is an administrator. Your EMR should then limit some commands (such as adding new users) to only those users who are administrators.
3. Something special

## Demo

You are provided with an example EMR system, which has been packaged into a JAR file. You can run it by typing:

```
java -jar jemr.jar
```

Ensure that the files *users.csv* and *patients.csv* are in the same directory as the jar file.

You will need to login using as "admin", password "test". Once logged in, if you type "help", it will show a list of available commands. You should try these to see how they should be implemented.