# CSC207 Fall 2014: Project Phase III

## Logistics

- Due date: 10:00pm Sunday 23 December 2014
- Group size: Four. In this phase of the project you continue to work in your team from Phase II.

## Overview

In Phase III of the project you will complete the implementation of the Android application. You will also
receive some updated requirements from your customer, which may require you to re-examine the design of your software.

## Learning Goals

By the end of this phase, you should have:

- Practised dealing with changing software requirements in the course of the software development project
- Worked closely with your teammates to re-evaluate and possibly update your design of a software system
- Produced a working Android application that implements your software design and corresponds to user requirements

## Updated Requirements

As nearly always happens, while you were working on the application, your client realized that they need more features. In particular, they realized that physicians will need to use the application as well. You have the following additional requirements for your Feature List:

1. Nurses and physcians can launch the triage application and log in using a username and password, which loads saved data, if it exists. In our, unrealistic, implementation, all username passwards are stored in the file *passwords.txt* that we give you.
2. Nurses can create new patient records and record individual patient data (name, birth date, and health card number)
3. Nurses can record the date and time when a patient has been seen by a doctor.
4. Nurses can access a list of patients (name, birth date and health card number) who have not yet been seen by a doctor categorized and ordered by decreasing urgency according to hospital policy.
5. Using the health card number, physicians can look up a patient's record, which contains all data recorded about that patient.
6. Physicians can record prescription information (name of the medication and instructions) for a given patient. (Notice that this information becomes part of the patient's record.)

The following features from the original Feature List are still part of the requirements:

1. Nurses can launch the triage application, which loads saved patient records from the given file *patient_records.txt*.
2. Nurses can look up existing patients in the records based on their health card number and view individual patient data (name, birthdate, and health card number).
3. Nurses can create a new visit record based on the patient's arrival time at the hospital
4. Nurses can update a patient's visit record with vital signs (temperature, blood pressure, and heart rate) at a particular time, retaining older values.
5. Nurses can save patient visit records with all the relevant collected data
6. Nurses can view previous visit records for a given patient (if they exist). Previous records may have been created in previous sessions of the application. For each visit record they can view what data was collected at the time (temperature, blood pressure, and heart rate).

# Task 1: Software Design

The additional requirements will likely have an effect on your system design. With your teammates, discuss what changes you need to make to your design from Phase II of the project in light of the new requirements. Create a file crc_phaseIII.pdf, following the same format you used in Phase II, and commit this file to the directory PIII of your team repository.

In contrast to Phase II of the project, in this Phase your CRC model should also include CRC cards for the front-end classes (the activities).

Note that your design should cover all application features from Phases II and III.

# Task 2: Developing the Android Application

### Android Version

When setting up your Android Application Project, you must select:

- Minimum Required SDK: API 14: Android 4.0 (Ice Cream Sandwich)
- Target SDK: API 18: Android 4.3 (Jelly Bean)
- Compile with: API 18: Android 4.3 (Jelly Bean)

### Feature List for this Phase

The Feature List for this Phase of the Project contains all of the features listed above.

### The Software Development Process

Your team should meet regularly while working on the project. We have two types of meetings | planning
meetings and status meetings.
For **planning meetings**, you need to meet twice: once in the beginning of the project and once mid-way
through the project phase. During a planning meeting, the team will (a) recap on the current state of the

project (if mid-way meeting), (b) decide on a set of tasks the team will accomplish before the next planning meeting, and (c) decide who will perform which tasks.
For the **status meetings**, the team will meet at least once a week, in addition to the planning meetings.
During these meetings, each member will report on (a) what (s)he has accomplished since the last meeting, (b) what (s)he plans to accomplish before the next meeting, and (c) if there are any problems/obstacles that prevent him/her from making progress.
To demonstrate the software development process the team followed, you need to **maintain a plain text file** called **meetings.txt**, where the team will record all meeting minutes.[1] On the day of each meeting, commit this file into your team repository. The contents of this file must match the state of the rest of your repository!

### The end of this project phase

At the end of this project phase, your team should have a working version of an Android application that
implements every feature on the above feature list. You should, of course, have Javadoc comments for all your code.
Please include a plain text file README in your repository telling your TA where to nd your Phase III project. You must tell your TA everything they need to know about your project to help them run your application and navigate your work.

# Task 3: Team member and self evaluations

You will be filling out and submitting an evaluation form for each of your team members, **including yourself**. This form will rate team members on citizenship: did they attend meetings, did they participate, did they do what they said they were going to do, and so on.

Your Phase III will receive a grade, and these evaluations will be used to make individual adjustments to that grade. **Peer evaluations are meant to be private: each team member will hand these in separately, and you should not show each other your forms.** In the case of serious disagreement, or if you request it, we will hold a team meeting to discuss the results, but we will **never** reveal individual ratings.

The evaluations must be submitted electronically. You will receive instructions about how to do this during the second week of Phase III. The deadline for submitting them is **48 hours after the deadline for Phase III.**

**IMPORTANT: Any student who does not submit their evaluations on time will receive a mark of 0 for Phase III.**

[1] See lecture slides for some example meeting minutes.

# Marking

All these items affect your grade. Note that the rubric is the same as in Phase II, but this time it will be enforced much less leniently.

- Final version of the design:
  - good modular design, with reusability and extensibility in mind
  - good use of OO concepts, such as encapsulation and inheritance
  - no redundant or repeated code
  - good use of files and data structures
  - well-written design description files
- Functionality and usability of the application:
  - all functions from the backlog implemented
  - easy to use application, intuitive navigation
- Javadoc:
  - required for classes, interfaces, methods and instance and static variables
  - must have a period at the end of every sentence
  - must use @*param* and @*return* tags
  - must use good English
- Coding Style:
  - must follow the Java naming conventions
  - indentation
  - consistency
  - whitespace
- Quality of the *README* file
  - it must take the TA less than 2 minutes to read your *README* file and understand how to run and use your application
- Proper use of version control:
  - commits by all team members
  - frequent commits over an extended period of time (as opposed to lump commits in the end)
  - meaningful and appropriate commit messages
- Good team-work:
  - Meetings and minutes must be committed according to schedule
  - Well-attended planning and status meetings, documented with minutes
  - the contents of the repository and the state of the code must match the contents of the minutes files
  - Peer evaluation

### Bonus Marks for enhancements:

- A significant enhancement, such as a great user interface or use of a database, will be awarded bonus marks (up to 25% of this phase's mark). Neither of these features constitute a

requirement for the project. The bonus marks will **only** be awarded if **all** of the requirements have been fulfilled **correctly**. In order to receive a grade, you must indicate, in the README file, what (if any) enhancement your team feels it has made.

**Look at the next page for the checklist**

# Checklist

Have you...

- used your new team repository and not your individual repository and not your repository from phase I to submit your work?
- committed **all** of your design files (CRC models and design description files), including your individual designs and the final design?
- submitted your team's meeting minutes?
- submitted **all** of your project files?
- submitted a *README* file for your TA?
- verified that everything was properly committed using *svn list* and *svn status*?
- submitted your team evaluation forms?