

CSC207/B07 Introduction to Software Design

Fall 2013 – Project Phase III

Logistics

- **Due date:** 10:00pm Thursday 28 November 2013
- **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:

- Physicians can launch the triage application and log in using a username and password, which loads saved data, if it exists. (In our, unrealistic, implementation, you are allowed to simply store usernames and passwords in a file on the device¹.)
- Physicians can save all collected data.
- Using the health card number, physicians can look up a patient's record, which contains all data recorded about that patient.
- 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:

- Nurses can launch the triage application and log in using a username and password, which loads saved data, if it exists. (In our, unrealistic, implementation, you are allowed to simply store usernames and passwords in a file on the device¹.)
- Nurses can save all collected data.
- Nurses can record individual patient data (name, birth date, and health card number), along with the patient's arrival time at the hospital.

¹Requirements for passwords file:

1. In the PIII directory, commit the plaintext password file, `passwords.txt`, that you will be using for authentication.
2. In your app, use the default internal storage location for storing the password file:
`this.getApplicationContext().getFilesDir()`

- Nurses can record and update a patient's vital signs (temperature, blood pressure, and heart rate) at a particular time, retaining older values.
- Nurses can record the date and time when a patient has been seen by a doctor.
- Using the health card number, nurses can look up a patient's record, which contains all data recorded about that patient.
- 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.

Task I — 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).

Task II — Implementing the Android Application

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

- Minimum Required SDK: API 8: Android 2.2 (Froyo)
- Target SDK: API 17: Android 4.2 (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.² *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!

²See lecture slides for some example meeting minutes.

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 find 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

Any student who does not submit their evaluations on time will receive a mark of 0 on this phase of the project. The evaluations are due Monday 2 December at 11:59 p.m.

You will be filling out and submitting a peer evaluation activity on CATME. This form will rate all team members, including yourself, on contributing to the team's work (contributing a sufficient amount of work, contributing work of good quality, being on time, helping teammates) and interacting with teammates (showing interest in teammates' ideas and contributions, asking teammates for feedback and using their suggestions to improve, making sure teammates stay informed and understand each other, providing encouragement and enthusiasm to the team).

These are meant to be private: each team member will submit these separately, and you are not required to 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.

Marking

All of these items affect your grade:

- CRC Model
 - The modularity of the design, and the degree to which it is reusable and extensible.
 - The degree to which the design meets the requirements.
 - The use of OO concepts, such as encapsulation and inheritance.
- The appropriate use of files and data structures.
- Functionality and usability of the application:
 - all functions from the feature list implemented
 - easy to use application, intuitive navigation
- Javadoc:
 - required for 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 Java naming conventions
 - indentation
 - consistency
 - white space
- 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
- Quality of the software development process:
 - the file **meetings.txt** must be committed according to the schedule
 - the contents of the repository and the state of the code must match the contents of the file **meetings.txt**

- Subversion commit history:
 - participation by all team members
 - frequent commits over an extended period of time
 - appropriate commit logs
- Peer evaluation
 - To view the evaluation criterion, see the CATME online evaluation form (www.catme.org).

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 5%). 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. In order to receive a grade, you must indicate, in the README file, what (if any) enhancement your team feels it has made.

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 `crc_phaseIII.pdf`?
- committed **all** of your project files, i.e. the entire Android project directory?
- committed a README file for your TA?
- committed `meetings.txt`?
- verified that your changes were committed using `svn list` and `svn status`?
- before Monday 2 December at 11:59 p.m.: submitted your team evaluation forms using CATME?