CSC207/B07 Introduction to Software Design
Fall 2013 – Project Phase I

Logistics

- **Due date:** 10:00pm Thursday 17 October 2013
- **Group size:** Pair. In Phase 0, you chose your partner for this phase. (Submitting this alone is not an option; if you do, you will receive a mark of 0.)

Overview

In Phase I of the project, you will review a client request for a software program and create a design of a solution.

Learning Goals

By the end of this phase, you should have:

- practised performing analysis of software specifications
- practised designing an object-oriented software system
- used CRC cards as an aid to the object-oriented software design process
- used a shared version control repository to support collaborative software development
- experienced some of the joys (and frustrations) of working with others and consequently developed some strategies to facilitate productive team interaction

The Client Request

A company has contacted you about building a program for them. Here is the description.

> We need an Android application to collect patient information and perform triage for patients who arrive at our hospital emergency room (ER). When a patient arrives at the ER, a nurse records their personal data (name, birthdate, and health card number), symptoms, and vital signs. The program should prioritize patients into categories of urgency based on the hospital policy (see below). As the patient waits, the nurse may check and record the symptoms and vitals periodically to re-categorize. The program should report on whether the patient is improving or worsening over time. When a patient has been seen by a doctor, the nurse makes a record of that.

Hospital Policy

The hospital policy for categorizing urgency involves assigning 1 point for each of the following:

- age < 2 years
- temperature ≥ 39.0 degrees Celsius
- blood pressure (measured in systolic/diastolic):
  - systolic ≥ 140 or diastolic ≥ 90
- heart rate ≥ 100 or ≤ 50

Based on points, patients are categorized as follows:

- Urgent (3-4 points)
- Less Urgent (2 points)
- Non Urgent (1 point)

Feature List

After consulting with your requirements engineering experts and after further communication with the client, you arrived at the following feature list:

- Nurses can launch the triage application, which loads saved data, if it exists.
- Nurses can save all collected data.
- Nurses can record individual patient data (name, birthdate, and health card number), along with the patient's arrival time at the hospital.
- 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 and update a text description of a patient's symptoms at a particular time, retaining older descriptions.
- 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, birthdate and health card number) who have not yet been seen by a doctor categorized and ordered by decreasing urgency according to hospital policy.
- Nurses can access a list of patients (name, birthdate, and health card number) who have not yet been seen by a doctor ordered by arrival time (least recent to most recent).

Task 1 — CRC Model

With your partner, create a CRC Model for the problem described. Instead of handing in your index cards, create a `pdf` file named `crc_phaseI.pdf` in your **team's** subversion repository in folder `PI`. This file should contain a collection of CRC cards, and nothing else. Please, make sure your file is easy to read, the figures are not too small, etc. You may want to look at a sample `pdf` file linked from the Project page of the course website.

Keep in mind that you are designing the **backend**, not GUI screens. None of the CRC cards should mention buttons, text fields, or other graphical components. In later phases of the project, you will implement both the backend and a GUI that will invoke the backend methods, but for now you are just designing the backend.

Task 2 — Design Description

Describe all data structures used by your classes. In addition, describe the format of the file(s) in which the application stores its data.

You may present your description in paragraphs or bullet points, as long as you use complete sentences. Imagine that the person reading the description is a colleague or manager who has technical knowledge but is not intimately familiar with your project. Your description should be no more than one page in length, using single spacing.

The design description will be evaluated on several components:

- the design (Is it a good design? Does it take into account all of the requirements?)
- the explanation (Do you explain all necessary parts of the design? Are your arguments justified?)
- the writing (Is the writing grammatically correct and free of spelling errors? Are the ideas clearly presented?)

Submit your descriptions in a plain text file named `design_phaseI.txt`.

Marking

All of these items affect your grade:

- CRC Model and Design Description
  - 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.
  - The quality of the written description and the explanation of the design.
    - Do you explain all necessary parts of the design? Are your arguments justified?
    - Is the writing grammatically correct and free of spelling errors? Are the ideas clearly presented?

Checklist

Have you...

- used your team repository and not your individual repository to submit your work?
- submitted `crc_phaseI.pdf`?
- submitted `design_phaseI.txt`?
- verified that your changes were committed using `svn list` and `svn status`?