Project 2: The Enhanced "Personal Directory" Application

Analysis, Design, and Implementation

CISC 3115 Section TY3
Introduction To Modern Programming Techniques
Fall 2018

1 Objective

In addition to the objectives in Project 1, the project is to help students achieve the following learning objectives:

- to understand interface and abstract classes and methods including a few commonly used Java interfaces (e.g., `Comparable`, `Comparator`);
- to use effectively Java Collections Framework APIs (e.g., `Collection`, `List`, `Set`, `Map`, and various subtypes of these interfaces).

2 Problem Description

A **large** student organization intends to develop a Personal Directory application to ease management and communication of the members. The Personal Directory contains personal entries for the members that consist of undergraduate students, graduate students, and faculty/industry mentors/advisors, and supports the following functionality:

1. finding personal address book entries;
   (a) finding entries using first name;
   (b) finding entries using last name;
   (c) finding entries using member type (graduate, undergraduate, academic mentor, industry mentor);
   (d) finding student's entries using mentor's name (given that a student can have at least one mentor);
2. pinging mentees from a mentor;
3. adding a personal address book entry;
4. editing a personal address book entry;
5. deleting personal address book entries;
6. setting up quick messaging to member;
7. removing quick messaging setup; and
8. listing entries ordered first by lastname, then by firstname

Among the above functionalities, functionalities 1, 2, 3, and 8 are mandatory, and the rest is optional for this project. The "pinging" is interpreted as a "flag" being placed on a mentee's entry, and the mentee can view who is pinging her or him, and can then choose to remove the flag.

3 Tasks

We divide the project into two major phases *Analysis* and *Design*:

1. Project Setup: setting up a Git repository for the team project
2. Analysis: analysis of the problem and preliminary design of the classes, and
3. Design: design, implementation, and testing the application.

3.1 Project Setup

- Each team shall elect a project coordinator for this project. The coordinator has the following responsibility,
  - to accept the assignment invitation via the Github Classroom;
  - to inform team members that the project set-up is ready; and
  - to schedule meetings and coordinate the members' efforts.
- The team members shall accept the invitation and clone the project repository. The collaboration and project development continues.

  The project assignment invitation is at

  `https://classroom.github.com/g/srBbWkHM`

3.2 Analysis

In this phase, *carefully* examine the problem and the programs you designed and wrote for project 1, and consider whether you can improve the design.

3.2.1 Deliverable

The team shall deliver a UML class diagram showing the classes, interfaces, and their relationship. The UML diagram should include essential Java classes and interfaces you are using (See Section 1).

3.2.2 Submission

Create a `doc` directory in the project repository, and add the UML diagram to the directory. The UML diagram can be a photo of the diagram although preferably a graph produced using some professional tool, such as, Microsoft Visio.

You may obtain a copy of Microsoft Visio free of charge from the Department's Microsoft Imagine Premium subscription at,

`https://goo.gl/ccQo2c`

3.3 Design

The team shall revise and enhance the application built in Project 1. One important aspect of this phase is to select most appropriate data structures.

3.4 Testing for Performance

It is required that the team should write a class that tests search efficiency, for which, you may randomly generate significant number of entries and do a search, and measure the wall-clock time, as illustrated in the sample programs given in the class.

3.4.1 Deliverable

The deliverable is the application built and enhanced. If the final product deviates from the initial design, include an updated UML diagram of the final design.

3.4.2 Submission

Create a `src` directory in the project repository, and add the source code to the repository.

Create a `test` directory in the project repository, and add a description and test results of the tests conducted.

4 Final Remark

To ensure success of the project and to meet the learning objectives, the team should examine the learning objectives, and reflect on how the team's analysis, design, and implementation help each member of the team meet the learning objectives. The team may need to reiterate the project to ensure that the final deliverable exhibits evidence that the learning objectives are met.

5 Submission Deadline

All submissions are done via `git` to the `git` repository hosted on the Github.

The analysis phase is due 3:40PM, December 6, and the design phase is due 12:00 noon, December 13.