Project 1: The "Personal Directory" Application

Analysis, Design, and Implementation

CISC 3115 Section TY3
Introduction To Modern Programming Techniques
Fall 2018

1 Objective

The project is to help students achieve the following learning objectives:

- to apply object-oriented thinking to analyze a problem;
- to design a Java application consisting of multiple classes;
- to use UML class diagram to present the analysis of the problem and the overall design of the classes;
- to be able to apply inheritance and polymorphism in the design and implementation of the classes and their methods;
- to be able to apply association/aggregation/composition in the design and implementation of the classes;
- to be able to handle exceptions;
- to be able to use simple text/character file I/O;
- to be able to use *class variables*, *instance variables*, *class methods*, *instance methods*;
- to be able to use *flow control* structures in writing class and instance methods;
- to be able to use *constructors*;
- to be able to design a *text-based* application;
- to be able to use *command line arguments* to control a behavior of an application;
- to be able to locate the Java API and third-party Java library documentation and use the documentation to support the development;
- and to be able to function in a team with support of a set of modern software development and productivity tools, such as, `git`, and `Github`.

2 Problem Description

A 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; and
7. removing quick messaging setup

Among the above functionalities, functionalities 1, 2, and 3 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/GlJXI29A`

3.2 Analysis

In this phase, carefully examine entities in the problem and map the entities to classes, and design the relationship among the classes.

3.2.1 Deliverable

The team shall deliver a UML class diagram showing the classes and their relationship. In each class, show data fields and public methods.

3.2.2 Remark

As a part of the requirement, the relationships of the classes must include both inheritance and association.

3.2.3 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 implement the application based on the analysis and the initial design. In this phase, the team may need to adjust their initial design.

3.3.1 Deliverable

The team shall build the application, and complete a UML diagram of the final design. In addition, the team shall verify that the application function as specified, and validate how the application meet the requirements described in Section 2.

3.3.2 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, October 30, and the design phase is due 3:40PM, November 8.