Project Phase 1

```
Due: Monday 16 November 2020 at 11 pm
Last modified: Friday 16 October 2020
```

Specification

You will create a program that allows people at a conference (examples: a tech conference, a TED-style conference, an employment fair, etc.) to communicate with each other in very specific ways. In Phase 0, you already designed a program that will allow Attendees to sign up for events and message each other. Now, you will change the class names to something more applicable so that the same design will become part of your Phase 1 submission. Alternatively, you can design Phase 1 independently of your Phase 0 ideas.

By the end of Phase 2, you will have an app that allows Organizers of a conference to plan the event, allows Speakers at the event to learn the information they need to speak and communicate with the Attendees of their presentation, allows Attendees to sign up for events and network with each other, and more.

For Phase 1, your User Interface (UI) will be text only. This involves printing menu items and user prompts on the screen and reading the user's response from the keyboard.

The objective of Phase 1 is to allow you to try out the ***SOLID principles*** of design and the ***clean architecture*** organization. In Phase 2, you will be asked to extend your program in various ways. This will give you a sense of how much your program followed these principles. Keep in mind possible extension that you might want to implement in Phase 2.

If your mark is higher on Phase 2 than on Phase 1, we will replace your Phase 1 mark with your Phase 2 mark. In order to receive a passing grade on Phase 1, you must submit a program that compiles and runs.

Users

Organizers of the conference should be able to enter rooms into the system and create speaker accounts. They should also be able to schedule the speakers to each speak in one or more rooms at different times so that at most one speaker speaks in each room at any given time and each room has at most one speaker at any given time. You can assume that all talks are 1 hour long and happen between 9 am and 5 pm. This will change for Phase 2. Also, in Phase 2, the Organizer will have the ability to cancel or reschedule an event before it happens. You may want to consider this when you create your initial design.

Speakers at the conference should be able to log in and see a list of talks that they are giving. They should also be able to message all Attendees who are signed up for a particular event, at once or individually.

Attendees should be able to log in and see a schedule of events for which they can sign up. They should be able to sign up for events, cancel their enrolment in an event, see the schedule of the events for which they signed up, send messages to and receive messages from other Attendees, and message with Speakers.

Login System

Every user should be able to log into the system and interact with a menu of options that are specific to their type of user.

Schedule System

It should be possible for an Organizer to schedule each speaker to give a talk at specific times in specific rooms, without double-booking a speaker (scheduling them to speak two places at the same time) or double-booking a room (scheduling two people to speak in the same room at the same time). This can be done automatically by your system, by giving the user enough information to avoid such conflicts, **or** restricting the choices an Organizer can make.

Sign-Up System

Attendees will be able to browse the talks and decide which ones they want to see. They will also be able to sign up for the talk. You can assume that each room has a capacity of 2 people besides the speaker. For Phase 2, this will change. But for now, it makes it easier to test your program. Attendees should not be able to sign up for an event that is already full.

Messaging System

Speakers should be able to send a message that automatically goes to all Attendees of their talk or multiple talks that they gave.

Organizers should be able to send a message to all speakers, one specific speaker, all Attendees, or one specific Attendee.

Attendees should be able to message other Attendees or Speakers. Speakers should be able to respond to a specific Attendee.

Saving and Persistence of Information

When your program finishes running, information inside the program will be deleted. If you want to be able to use that information the next time you run your program, you will have to store that info outside of your program and have your program read it back in. Good ways to do this are by storing information in a `.txt` file, a `.csv` file, or a `.ser` file.

Do not use a database for Phase 1. If you know how to connect your program with a database, you can do this for Phase 2 if you want. But we will only mark the way it interacts with your program and not the data base itself. So including a database in your Phase 2 submission will not earn you extra marks.

What You Should Submit

- Your code, including Javadoc (https://q.utoronto.ca/courses/180703/files/9604138/download?wrap=1) (https://q.utoronto.ca/courses/180703/files/9604138/download?download_frd=1) for anything that is not private.
- Any configuration files that are needed to run your program
- a readme.txt file that contains any instructions we need in order to run your program and try out the required functionality. The readme.txt file should be located directly inside the phase1 folder in your group repository
- a design.pdf file that contains a uml diagram of your program. If it is too large and complicated, you are welcome to create a series of diagrams in files called design1.pdf, design2.pdf, etc. that contain uml diagrams for different parts of your program. These files should also be directly inside your phase1 folder.

Getting Started

Designate one group member to do the following:

- Log into MarkUs and click on the phase1 repository url. This will either do nothing or result in an error message. ***This is fine***. The act of clicking on the link will create the repository. Copy the link.
- Clone your repository, using that link. The folder name for your repository should be group_0XXX where the X's should be replaced by your group number. You can find your group number from your repo's url.
- In IntelliJ, if you have a project open, go to the "File" menu and select "close project".
- From the welcome screen, choose "Create New Project" and make sure that it is located inside your phase1 folder in your group repository.
- Add, commit, and push this new project.

Everyone else in your group should clone the repository. The new project should exist in everyone else's copy of the phase1 folder. Make sure that any classes or interfaces that you create go in the "phase1/src" folder.

Tests

Test Driven Development is a method for coding where you decide what each method should do before implementing it. Then you write tests to make sure each method works. As you implement the methods, you can run them against the tests that you created to make sure they do what you intended. **This is optional, but an excellent way to get started, after you finish your CRC cards.**

Feel free to ask questions about the project before/after lecture, during **office hours (https://q.utoronto.ca/courses/180703/pages/office-hours-and-requests)**, on the **message board (http://piazza.com/utoronto.ca/fall2020/csc207/home)**, or you can request an appointment through the Office Hours page. When you ask a question on the message board, the entire class can benefit from hearing the answer. If your question involves discussing your code, an appointment is better, so that you do not get accused of leaving your code for others to plagiarize. Any subset of your group, from one member to everyone, is welcome to attend an appointment.