Faculty of Science

Course:	CSCI 2020U: Software Systems Development & Integration
Assessment:	Final project
Topic:	Java, GUI, File I/O, Socket I/O, Multi-threading, Database

Overview

This group project is designed for you to demonstrate the skills that you have learned in this course. The final project that you submit **before 23th March** will be a completed application. *Non-functional requirements, especially those associated with production-readiness, will be considered very important when marking this project.* You are expected to work in **a group of 4-5 (4 is recommended)** when completing this project, but some exceptions can be made in rare cases.

Note: Any projects from individual students will not be accepted, except if special permission has been given by the instructor in advance, since working alone eliminates some of the learning objectives of this assessment.

The entire project will be maintained in a public **Git** repository (e.g. **GitHub** or **BitBucket**). Essentially, your project submission will consist of giving access to your Git repository (make it public; more details, below). The instructor and TAs will download the latest version of your project, along with other information (e.g. commit logs) available through Git.

Detailed Instructions

Choosing a Topic

The project topic is, for the most part, up to you. Therefore, ensure that you choose a project topic that lets you demonstrate the skills you have learned and includes the required components.

Basic Requirements (Objective, 10 marks)

It is your job to incorporate as many course concepts into your project as possible. At a minimum, your project should have:

- a README.md file containing the group member names, their contributions, the repository URL, and any instructions required for using your application (more details, below)
- a working **gradle** build file, that allows the following from the command line:
  - download all required dependencies
  - build the project
  - run the project (a task to run each component)
- a sophisticated user interface
- file input and output

- socket input and output
- multi-threading
- Database

Presentation & Additional Requirements (Subjective: 10 marks)

You should pay attention to the non-functional requirements discussed in the lectures (e.g. adherence to recommended programming/documentation practices, maintainable design, and usable interfaces). Both functional and non-functional requirements will be considered in your evaluation.

When evaluating this project, the instructor will attempt to give a metric to the amount of work involved, considering several important factors (e.g. design, cleanliness of code, code comments, variable/function naming, error checking, usability/user-friendliness/aesthetic). This metric will be affected by the size of your group (i.e. what will be evaluated is the average work done per group member).

As for the **look and feel** of your application, this will be fairly important. You will need to ensure that your project is presentable (line up form fields, give a non-default color scheme, etc.), and pay attention that your user interface is convenient and easy to use. If you incorporate concepts outside of this course (e.g. game engines, 3D graphics, web service APIs) the effort will be counted, but with a reduced weight. Ideally, you should focus mostly on components mentioned.

The Git commit logs will also be used to ensure that work was evenly distributed. If you work together (e.g. pair programming), be sure to indicate which group members were responsible for each commit. If there are no names in the commit message, I will assume the user that submitted the commit was solely responsible for the changes of that commit.

Project Size

The actual size of the project (in terms of the number of activities, or lines of code) will differ from group to group. Ultimately, the factor being considered is how much work appears to have gone into the project. This does not count learning course concepts. Some people take longer to learn course concepts than others, but this does not mean that you worked harder on the project in terms of evaluation. Larger groups should ensure that projects are proportionally larger, to account for the extra people available to work on the project.

Evaluation

For 10 out of 20 of the allocated project marks, as indicated above, students will be evaluated subjectively. The remainder of the marks will be allocated based on whether or not your web application is functional, and whether or not you have demonstrated all of the major course concepts. Factors that will be taken into consideration for the subjective evaluation include:

- User interface
    - Usability (how easy/efficient is your application to use?)
    - e.g. Validation of form input
    - Appearance/look and feel of user interface/graphics
- Code quality
    - e.g. Proper indentation, element/class naming

- Adherence to best practices for design, development
    - e.g. Variable/function naming, comments, error checking

- Presentation

It is not guaranteed that all members of a group will receive the same project mark. The marker will be examining the GitHub/BitBucket commit logs to verify that every group member contributed significantly to the project. If that is not the case, the offending person may receive a much lower mark.

How to Submit

Your project will be maintained on either **GitHub** or **BitBucket**, in a public repository.

It is your responsibility to ensure that all necessary files are included. If a file is missing from the repository, you do not get credit for it.

Every project will contain a README.md file, which will contain three sections:

1. A list of the full names of every group member, and their GitHub/BitBucket usernames
    - Do not include Student IDs in the README.md file
    - Include a brief summary of the contributions of each group member

2. **The repository URL**

3. Any instructions necessary to use your application
    - There should be no special installation requirements, aside from **gradle** dependencies
    - A list of **gradle** tasks for compiling, running
    - If your software has authentication, provide some sample credentials to use for testing

When the project due date approaches, you will submit this **README.md file** to the Project drop box on **Blackboard**. Only one group member should submit, but it is strongly recommended that all group members be present for peace of mind.

Note: Do not make any changes to the remote repository after the project's due date. Failure to comply could result in a deduction of marks.

Note: Any instances of plagiarism will result in the student(s) receiving **a mark of zero for the project**, and further disciplinary action will be taken. Plagiarism includes, but is not limited to:

- Copying of (any amount of) work from the Internet, without proper citation
- Submitting a body of work, cited or not, that is primarily not your own work
- Allowing your own work to be copied by a fellow student
- Copying of (any amount of) work from another student, past or present, without proper citation