



Faculty of Science

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

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 in the last week of classes 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 2-4 (2 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 Git (e.g. GitHub or BitBucket). Essentially, your submission will consist of access (e.g. credentials) to your Git repository. The instructor will then use this access to 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 in this course. Consideration will be given to projects whose functionality is very different from samples and those developed in labs and assignments in this course. When evaluating your project, I will consider this as requiring extra work. More work done often equates to a higher grade.

It is acceptable if you want to do a project related to industry. If someone you know wants an application developed, and it lets you demonstrate the skills you've learned in this course, then you can use it for your project (even if you plan to sell that web application when you are finished). Please keep in mind that nothing your prospective buyer says or does will affect the due date or expectations that I have for this project. No matter what happens, this project is due when it is due, my expectations will be based on the content of this course, and I will expect a certain degree of professionalism and production-readiness.

The instructor will be open to alternatives to complete, stand-alone applications. For example, if you wanted to create a Minecraft mod, it could be acceptable. The other rules will still apply, for the most part (e.g. including course concepts). Any alternative project is done at your own risk, this is merely to let you know it is possible.

Basic Requirements

It is your job to incorporate as many course concepts into your project as possible. I would say that at a minimum, your project should have a sophisticated user interface, use files for input and output, and use sockets for communication. Other features you may include could be 2D (or even 3D) graphics. In addition, you should also pay attention to the non-functional requirements discussed in the lectures (e.g. adherence to recommended programming/documentation practices, maintainable design, usable interfaces). Both functional (course concepts) and non-functional requirements will be considered in your evaluation.

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.

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 colour 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 concepts from this course.

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.

How to Submit

To submit this project, please push all your work to your Git repository, and submit access to that into the drop box for this assignment on Blackboard. 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. Most groups will use GitHub, and thus can merely grant permissions to the instructor's GitHub account (randyfortier).

As many students may not use an identifiable username in GitHub (or BitBucket), please ensure that there is a README.md file in the root folder of your project. In this file, include the full names and GitHub (or BitBucket) usernames for each group member. If there are any special instructions for getting your project up and running (e.g. admin credentials), be sure to include instructions in the README.md file as well.

Evaluation

For 10 out of 20 of the allocated project marks, 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 cleanliness
 - e.g. Proper indentation, element/class naming
- Adherence to best practices for design, development
 - e.g. Variable/function naming, comments, error checking
- Amount of work done
 - Not including learning the technology

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
- Copying of (any amount of) work from another student, past or present, without proper citation
- Allowing your own work to be copied by a fellow student