SOFE 4790U: Distributed Systems (Fall 2023)

Instructor: Dr. Ahmed Badr
Adapted from content developed by Dr. Q.H. Mahmoud

Individual Programming Assignment #1

The objective of this individual programming assignment is to get a flavour of the effort involved in designing and developing client/server applications. You will practice designing and developing an innovative client-server application of your choice using Java sockets (I recommend using TCP sockets, but you may use UDP datagrams if you wish).

The Task
Design and develop a novel, usable, and useful client/server application of your choice. It must accomplish something useful and has some novel features. The server must provide at least 2 significant functionalities or services for clients to use, and the application must have 2 novel features.

Guidelines

- Your application must consist of a multi-threaded server and a client, both coded in Java, and it cannot be an HTTP proxy server.
- You must use the Java programming language.
- You must use sockets; applications that make use of high-level APIs (such as URL, URIConnection, etc) will not be accepted.
- Your application should continue to handle clients' requests until it is manually terminated, or you have a UI for run/shutdown.

Important Notes

- **Deadline**: Assignment#1 must be demoed and submitted **by 11:59pm (night) on Thursday, October 19th**. *No extensions, so plan accordingly*.
- Your solution must be designed and developed by yourself (your own work).
- While students are encouraged to discuss the assignment and general ideas for solutions, each student must design and develop his/her own solution and code. No code sharing is allowed, and no two or more students can have the same application. **JPlag will be used for detecting code similarity**.
- The assignment will be assessed based on the grading rubrics provided on page 2 of this document.

Submission Guidelines (note the 2-step submission)

1) **Source & class files, and a README file**: Submit your assignment solution source code (*.java) and bytecode (*.class) files, along with a README file on GitHub **by 11:59pm (night) on Thursday, Oct 19th** as per the following instructions:
    a. Create a GitHub account (https://github.com/join)
    b. Register for GitHub Devpack in order to get private repositories (https://education.github.com/pack)
    c. Go to the following link for Assignment 1: https://classroom.github.com/a/GLUlXlws
    d. If you have not used GitHub before, go through this one-hour tutorial: https://lab.github.com/githubtraining/introduction-to-github

e. Your submission must include a README file with a brief description (one paragraph) of the application, and instructions on how to run your application.
f. If your applications require any resource files, make sure you include them in your submission on GitHub.

2) **Report**: Submit your assignment report through Canvas by **11:59pm (night) on Thursday, Oct 19th** (look under Home -> Assignments –> Assignment #1 Submission). Your report must be in PDF or Word and must include:
   a. One full-page (approx. 500 words) detailing your application idea, novel feature(s), challenges and solutions. You may include one clear diagram but no screenshots of the application.
   b. A description of the tests you have run to demonstrate the functionality of your application. You must describe the actions with screenshots, and clearly demonstrate this was done by you on your own laptop (e.g., show command-line prompt with your account name).

Grading Rubrics

Item (%)	**Excellent (Full mark)**	**Good (75%)**	**Satisfactory (50%)**	**Unsatisfactory (25%)**	**Zero (zero)**
Report (20)	Clearly documented and well organized with novel features, sample runs with description & screen shots, challenges and solutions.	Readable but not well organized or missing parts.	Documentation is minimal, but clear sample run.	Documentation is minimal, with no sample run.	Non-existent.
Usefulness and usability (20)	Useful and intuitive to use.	Nothing special.	Requires a manual to use.	Not useful or usable.	Non-existent.
Novel features (20)	Creative and offers novel functionalities.	One novel feature.	Nothing special.	Cannot be considered as novel features.	Non-existent.
Functionality (20)	Fully functional with no errors or warning.	Functional but nothing special and sometimes no response.	Basic functionality beyond code covered in class.	Error messages during run.	Doesn't compile or run.
Source code (20)	Follows coding standards (name, date, title, meaningful variable names, whitespaces, etc.) and code is fully documented.	Readable source code. Doesn't follow coding standards.	Spaghetti code.	Code provided is incomplete or does not make sense.	No source code provided or the link to the source code is not accessible.