**EIE4108 Distributed Systems and Cloud Computing**

**Assignment 1**

**Name: Sze Kin Sang**

**Student No.: 19062606D**

To implement the code for assignment 1, we need to modify the *run()* and *actionPerformed()* functions(at least) in order to allow the code to run properly.

Screen Capture of the program output

Graphical user interface, application

Description automatically generated

Screen Capture of the *run()* function

Text

Description automatically generated

For this section, when the thread is created and started, it will stay inside a while loop to execute the time ticking function. The thread will increment the clock and update the text fields every seconds.

When the thread is interrupted, the while loop will be broken and thus the thread will be killed.

Screen Capture of the *actionPerformed()* function

Text

Description automatically generated

This section contains the action taken by the code when the respective button is pressed. To start a thread, we need to create and start it manually and it will keep running in the background.

When the stop button is pressed, the program will check if the thread is initialized or not, then interrupt the thread to kill it, and set the clockThread parameter to null.

When the reset button is pressed, the program will also check if the thread is initialized or not, then interrupt the thread to kill it. Since the thread is killed, to restart the counting, we also need to create and start the thread manually again. The code for creating and starting the thread can be placed inside the *reset()* function, but for simpler presentation, I placed it here.

Since multi-threading allows program to perform tasks in the background without interrupting the main program, the performance of the program can be improved by having task running in background at the same time while user can still interact with the GUI interface and buttons without any delay.