Name: Dung Ngo

Due: 02/12/2021

**Introduction:**

This lab assignment introduces the basic concepts of using RMI (remote method invocation), including the setup of the Election interface, Server/Client classes. Many Client objects can send requests to a Server by invoking its remote methods. Remote methods are considered an intermediate between Clients and a Server by registering themselves in the registry object. To run Client and Server, the registry must be run in the same directory as the other two.

**Additional requirements**

1. Ensure that each user votes only once

I use a static variable as a fixed-length array of class Server to store the ID of voters who have voted. The system will automatically create a voter’s ID. If voters have voted, their ids have been stored and thus are not allowed to vote again. Furthermore, since this is a static variable, many remote objects of Sever class can all share this same variable.

2. The voting records remain consistent even when it is accessed concurrently by multiple clients.

I use a hash table whose key and value are a candidate’s name and his/her total votes, respectively. Since the hash table is synchronized, this makes the hash table better and safer for a threaded application like this scenario which involves many clients and one server.

3. All votes are safely stored even when the server process crashes.

For every update to the hash table, I will write that result to a local file called store.txt. In this case, if the server crashes unexpectedly, its results are still stored in the file. And whenever the server is backed up again, it can restore its data by reading the file.

Below is a screenshot of my program (1 Server and 2 Clients that send requests concurrently to the server)

A picture containing text, screenshot, monitor

Description automatically generated

Figure 1