CPSC 353 Class project Name:\_\_\_\_\_\_\_Mohammed Alsoheem\_\_\_\_\_\_\_\_\_\_\_\_\_

(Final Submission)

Name of your project: Secure Network Game

URL of github repository for the project: https://github.com/aloma111/GameProject

Names of Team Members:

Mohammed Alsoheem

Moath Almar

Parker Janke

**Original Deliverables**

The purpose of the project is to explore issues in Cybersecurity by implementing a secure network game using multiple Java technologies. The project addresses security concerns by encrypting and decrypting messages and by securing the network connection using the SSL (Secure Socket Layer) security protocol. SSL technology allows users and server to commute in secure connection. The data is encrypted before sending and is decrypted when received. The server presents a certificate (public key certificate) to the user as part of the authentication process. The data is encrypted with this certificate to ensure confidentiality. Once encrypted the messages cannot be modified or viewed by a third party. Java 8 provided the classes that are used to generate the keys (KeyPairGenerator, SecureRamdom, KeyPair, PrivateKey, PublicKey…). The client and server use FSMs (Finite state machines) to maintain the game states. Each group of players has its own state. Each player (in a separated thread) also has the state. The problem of shared data is also solved. Java supports concurrency with mechanisms such as Thread/Runnable, Lock, Synchronized statements, notify and wait. Each class in the project implements a specific task. The design patterns organize the classes and assist with finding and fixing bugs and with code maintenance. The following design patterns are used: Singleton pattern (provide game questions that are from file), State pattern (maintain game state), Command pattern (implement loose coupling in a request-response model), Iterator pattern to iterate through the question list.

**Modifications (if any) to the deliverables and reasons for the modifications**

The project generated the key pair by using keytool command line. It saves time if keys are generated once when the project is started.

The project used singleton pattern, state pattern as design patterns. Other patterns are optional only. They will be added in next phase if any.

**Your documentation must include**

* Instructions for installing components needed to compile and run the project
* Instructions for compiling and running the project
* Instructions for using the software

Follow the following steps to build and run:

1. Download from GitHub

ClientGUI.java

ClientQuestionPanel.java

Configuration.java

GameClient.java

GameClientHandler.java

GameServer.java

Message.java

QuestionGenerator.java

ServerGUI.java

certfile.cer

GamePrivateKey.store

questions.txt

questions/Answer.java

questions/MultiChoiceAnswer.java

questions/MultiChoiceQuestion.java

questions/Question.java

questions/TrueFalseAnswer.java

questions/TrueFalseQuestion.java

2. open terminal or command line and enter the following command to build

javac \*.java

3. Open terminal or command line and enter the following command to run server

java ServerGUI 1234

4. Open terminal or command line and enter the following command to run client

java ClientGUI localhost 1234

java ClientGUI localhost 1234

java ClientGUI localhost 1234

...

**How to use the software**

Server: Click Start server button to run the server

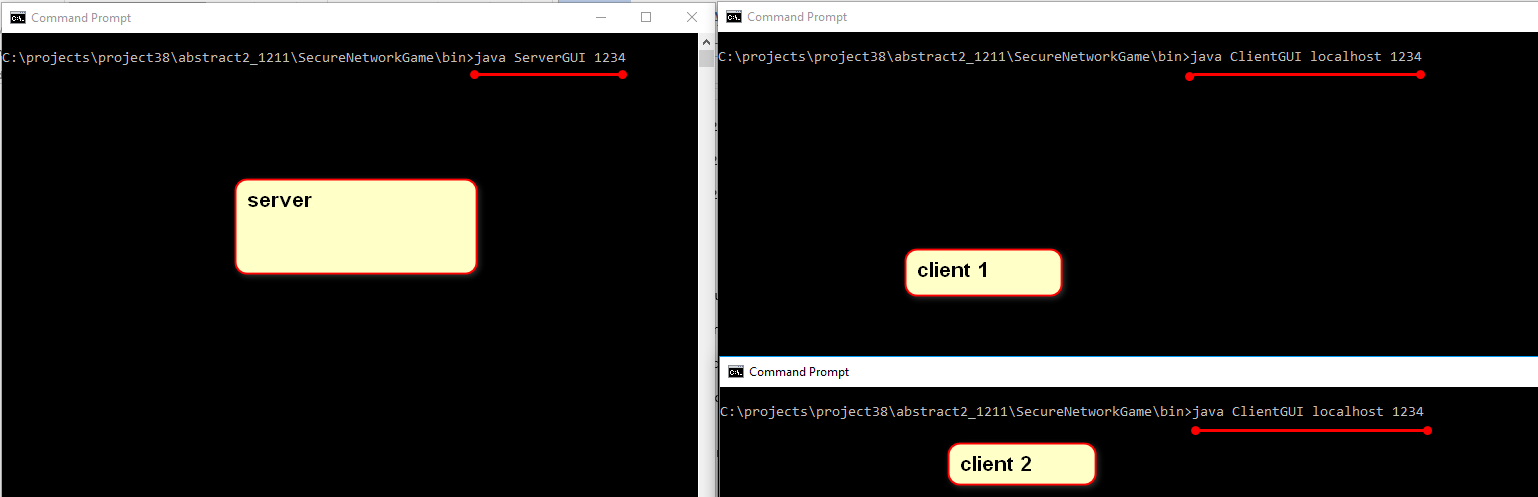
Client: Click Connect to connect to server and answer the question.

The server will return the current score of player, the maximum score, and the number of players.

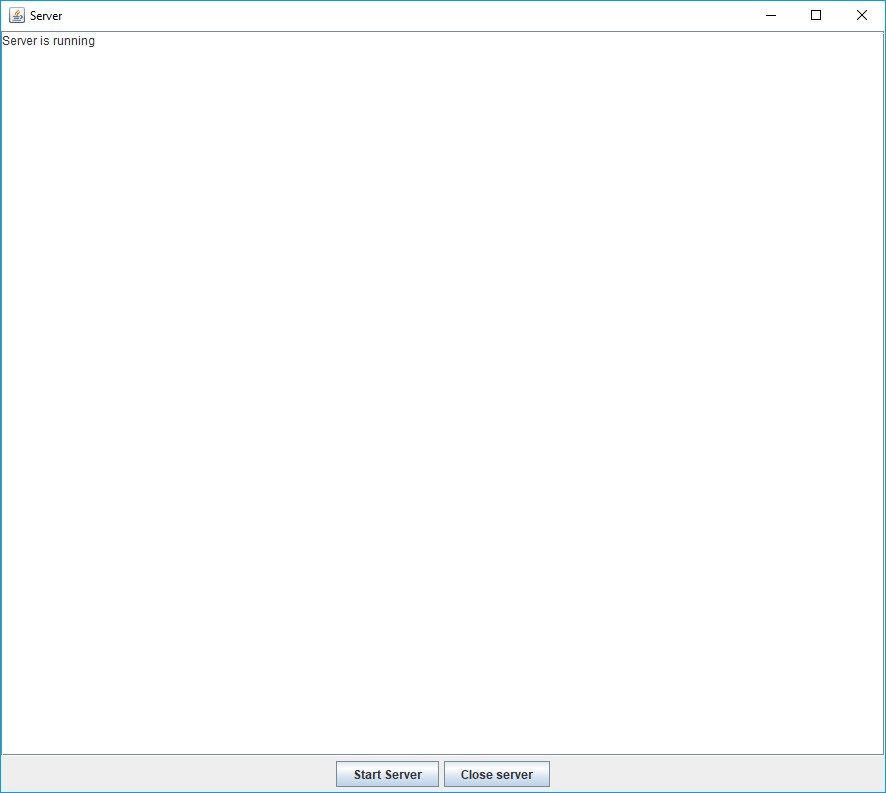
The player that is the first one answers the question will be scored. Correct score is + 10 and incorrect score is -5.

The server will display the next questions to all players when it gets an answer from the fastest player.

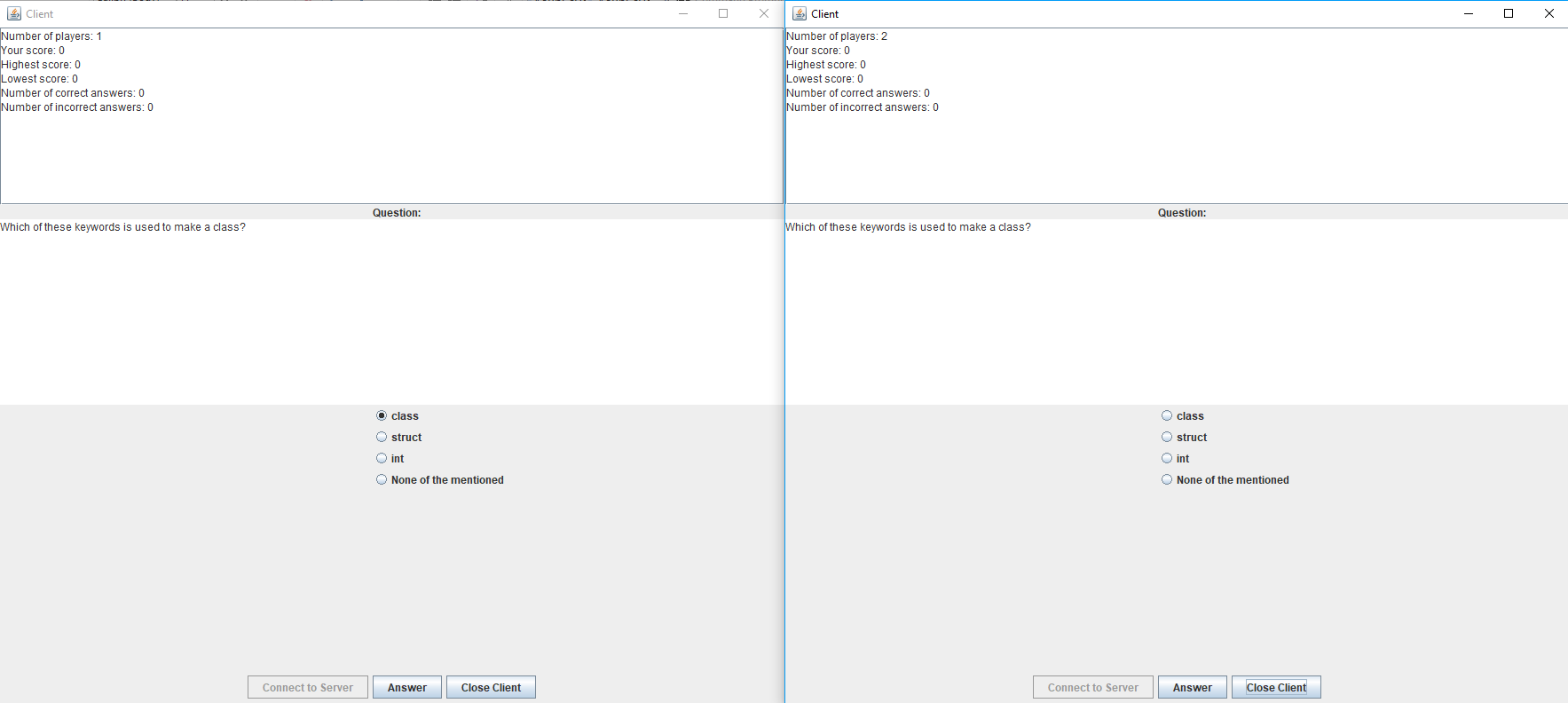
**Screenshot**

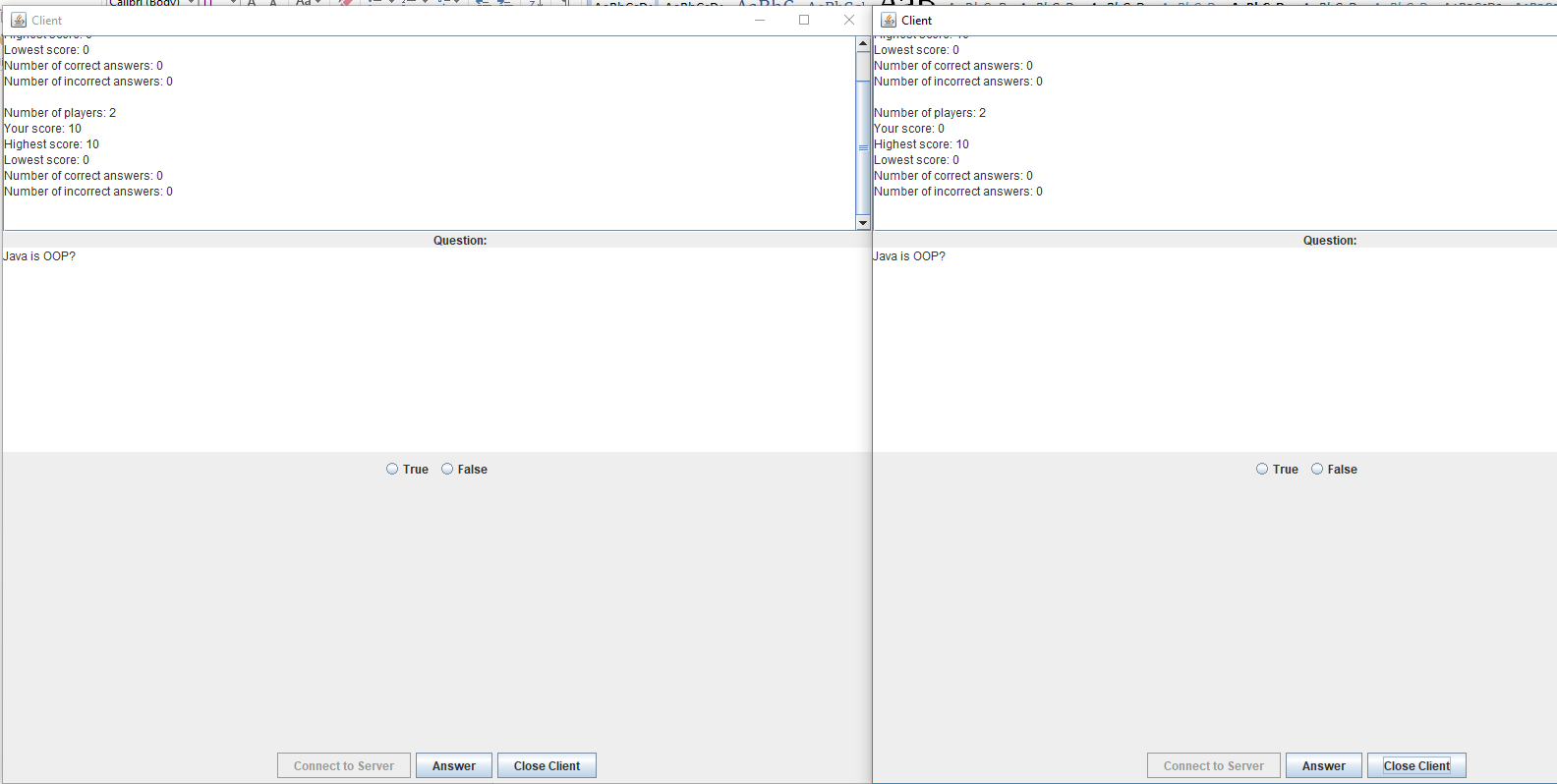


Server



Clients





Contributions by each team member

Each Team Member must post a README-name.md file in the repository listing their contributions to the project and also post this file via Blackboard.

See https://github.com/aloma111/GameProject