Purpose:

The project is a user-created trivia game that can be played with at least 4 players on a local network. The server operator can add questions to the trivia game by formatting questions into a text file and uploading the text file to the program. If a user answers a question correctly, then they will receive a point. The server operator can control the time interval between questions and the general pace of the game by clicking on the “send question” button as fast or as slow as he/she wants.

When the server operator uploads the questions, he or she will have the option to modify the questions in the program, and do a final look over before he/she decides to click “finalize”. At which point, the questions can no longer be modified. When the server operator presses the “send question” button, the next question in the list of questions is fetched, and sent to the client. And that process repeats until there are no more questions left, at which point, the server sends the player scores to the clients to see who got the most points. At this point, the clients’ buttons are disabled, and they can no longer participate in the game. In order to start a new game, the users and server operator must restart the program.

High Level Entities:

The data access layer constitutes a .txt file containing questions and answers. This is a bank of questions and answers prepared so that the server can send them to the Client.
The Controller Object is the Server class. It is in charge loading question and answers, accepting connections with clients and sending information to each client. Both the server and clients have their own GUIs and displays information respectively.

Low Level Entities:

The Server will load these questions and answers using FileReader and BufferReader. Once the questions and answers have been loaded, the server can then proceed to send them to all the connected clients. On the client side, each client’s GUI will display a multiple choice interface and the client will be prompted to choose an answer. The answer is sent back to the Server and the Server will update the score for the clients. After all the questions are asked, the game ends.

Benefits, assumptions, risks/issues:

Benefits:

- Modularity. The project structure allows for ease of understanding and allows for quick last minute changes with minimum refactoring
- This flexible design allows for multiple collaborators to add on to the project.

Risks:

- Minimally engaging game mechanics