

Tutor: MINXIAN XU
Team Name: DouTu Team
Team Members:
Lixuan DING 927096
Xinwei LUO 948602
Lei REN 950214
Min XUE 897082

Scrabble Game

1. Introduction

This project is about designing a Scrabble Game which allows multi-players to play at the same time and one winner will be ultimately got at the end of the game. This game is actually a word game in which each player will take turns to add one character at a time and get a score based on the word they generate. Whether the score can be added or not is according to other players' voting. Once a player disagrees with the submitted word, the score will not be given to the corresponding player. Some bonuses are designed for the word which includes the character in the blue grid. The game is designed for many users to play simultaneously, hence one player can invite other users who are involved in the system to join the game. Other users who log in the system later can only watch it without executing any operations. The client-server architecture is applied for achieving all those functions and they will be connected through the socket. Also, by implementing multi-threads, various users can join the game and perform different operations concurrently. TCP is chosen to guarantee reliable communication with which any failure or exception will be caught without affecting the system. In addition, other specific protocols are also designed for exchanging messages. All functions shown above will be achieved and demonstrated via GUI for both client and server.

2. Graphical User Interfaces(GUI)

2.1. Login

This Login GUI(Graphical User Interfaces) is intended to help users check whether the connection between server and client is successful or not. In addition, once entering a wrong port number or invalid input, then users would receive an error message as a reminder. The preset port number is 2332.



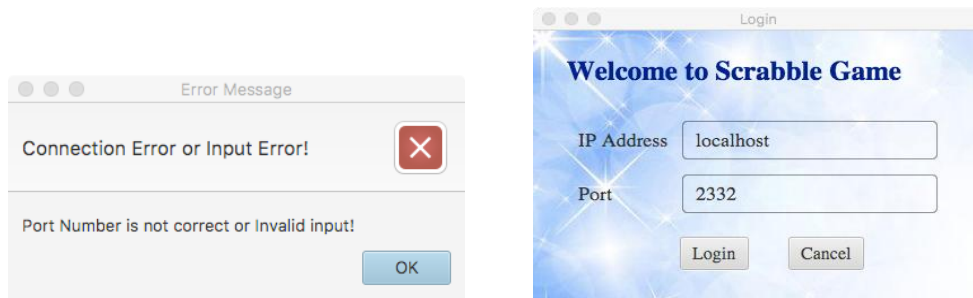


Figure 1: Login GUI (Graphical User Interfaces)

2.2. IdentifyUsername

This IdentifyUsername GUI (Graphical User Interfaces) is applied to validate usernames. The username refers to an identification. Various users have their specific username. That is to say, repeated usernames cannot be permitted in the project.

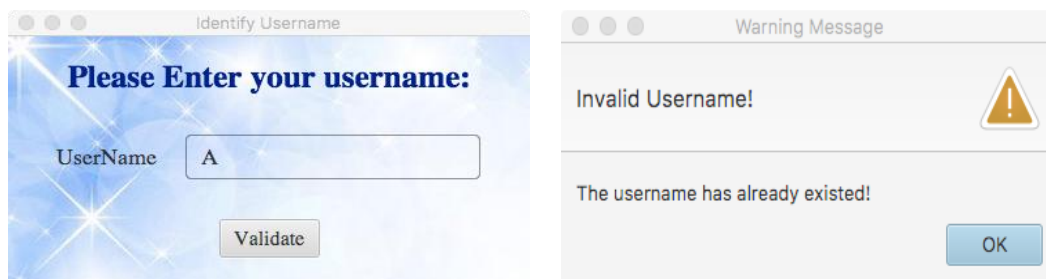


Figure 2: IdentifyUsername GUI (Graphical User Interfaces)

2.3. MainPage

This MainPage GUI (Graphical User Interfaces) aims to display online users. All the players who join the game would be shown in the User Status area. A user can invite others to join the game. By clicking button "Start", a new game will be processed. Once a game started, other users cannot invite anyone else or start a new game. In other words, a user can only choose the watch mode. Button "Refresh" is used to update online users. Moreover, a user who clicks button "Quit" would quit the whole game. Then this user's name would be removed from the list of online users.

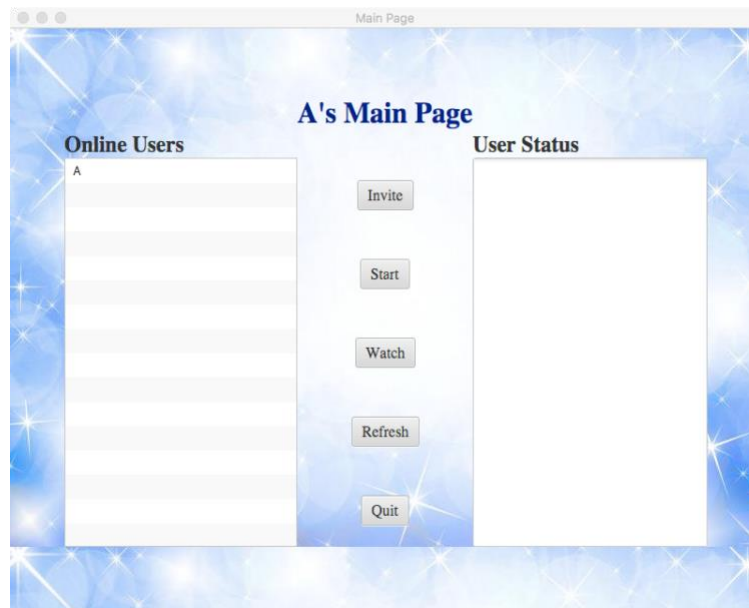


Figure 3: MainPage GUI (Graphical User Interfaces)

2.4. Game

This Game GUI(Graphical User Interfaces) is used to present the whole process of scrabble game. If it is the turn of one player, then he can do a series of operations. A player can only drag one character at a time. Whether choosing this character or not is by clicking separately button "Confirm" and "Undo". After confirming a character, a word read from left to right and a word read from top to bottom are displayed in the Choose A Word area. Furthermore, by clicking button "Submit", other players would receive a voting window. For a player, button "Pass" is applied to pass his turn.



Figure 4: Game GUI (Graphical User Interfaces)

3. System Components

- GUI(Graphical User Interfaces): assist users to implement the whole process of the game in a visual way.
- Architecture: a client-server architecture, which allows multiple users to do a series of operations.

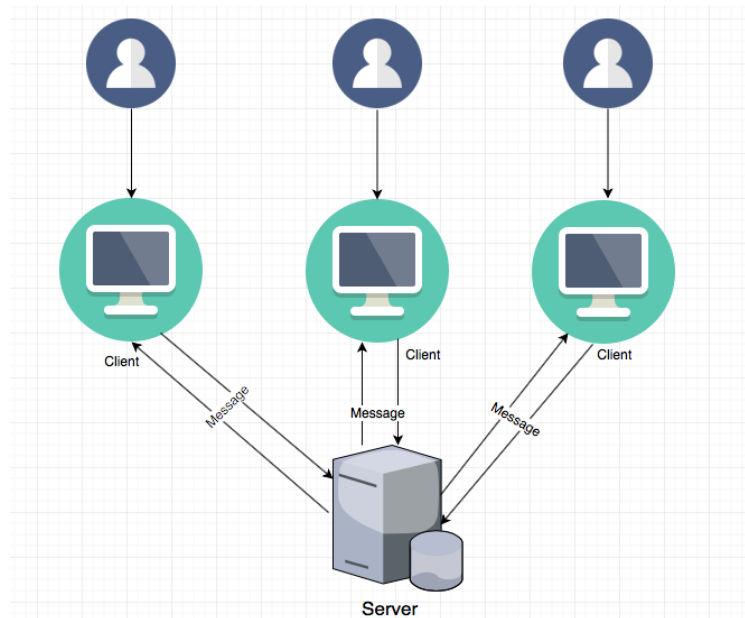


Figure 5: Architecture of the Project

- Sockets: by using sockets, the connection between the server and client can be established. Then, through invoking sockets on both sides, the client could connect to the server and exchange messages between each other.
- Protocol:
 1. The first protocol is TCP. It provides an essential and reliable connection between server and client.
 2. The second protocol is message commands between server and socket. When the server or client receives a specific command, it will carry out corresponding operations. "|" is an extremely key symbol to split every part of the command.
 - ◇ When a client wants to start a game, after clicking button "Start", a message "startstart|" will be sent to the server. Once the server receives this message, then it will make some analysis and judgements and broadcast "startstart|" to all clients that take part in the game. The user who receives this reply would open his own "Game" GUI.
 - ◇ When client A wants to invite client B to take part in a game, after choosing the name of client B then clicking button "Invite", a message "invite|B" will be sent the server. Once the server receives this message, then method "invite()" will be executed. A reply message "invite|A"(A invites you to start a game) will be sent back to client B.
 3. The third protocol is about starting or closing a socket. For instance, while the server executed, it establishes a socket to listen to an incoming connection. Then, when a client establishes a socket that connects the server, the server will output that "XXX-

Client connection accepted".

- Thread: Multi-thread architecture. Each thread of server offers the service of one corresponding client.

4. Results

- MainPage

A user should at least choose one player to invite, otherwise, a warning message would be displayed as the following alter window.

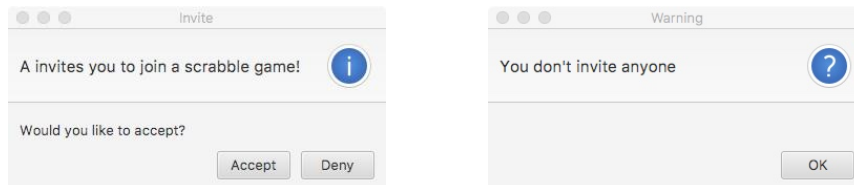


Figure 6: Invite Players

Once a game has been started, other users cannot invite anyone else or start a new game. Otherwise, after clicking button "Invite" or "Start", different warning windows would pop up.

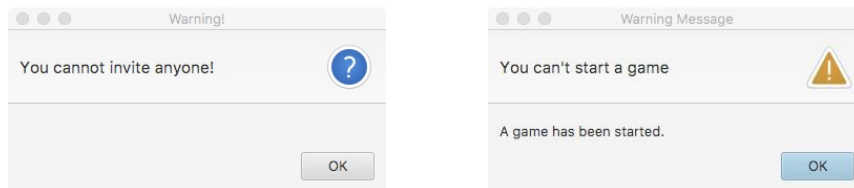


Figure 7: Game Already Started

All users who have logged in the system will be shown on the Online Users panel with their usernames.

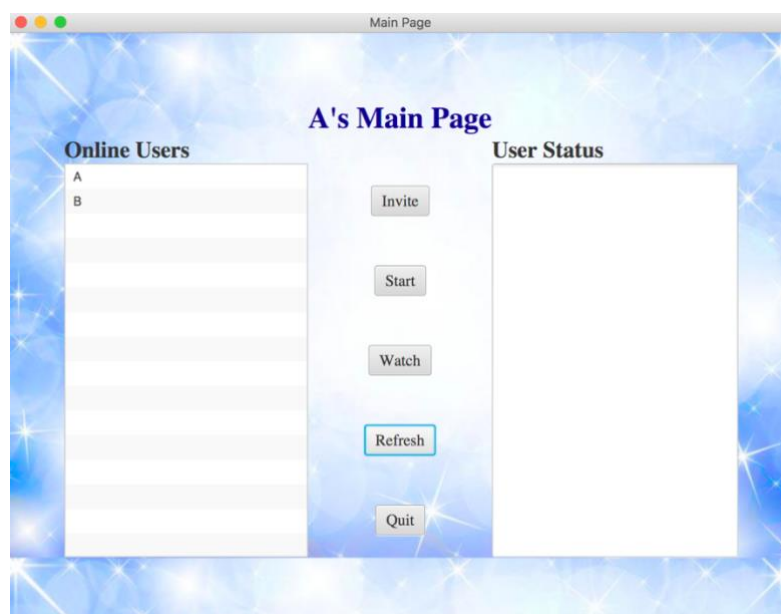


Figure 8: Online users

Once a player wants to invite others to join the game, the "Refresh" button is available for updating the Online Users list and choose invited users from it.

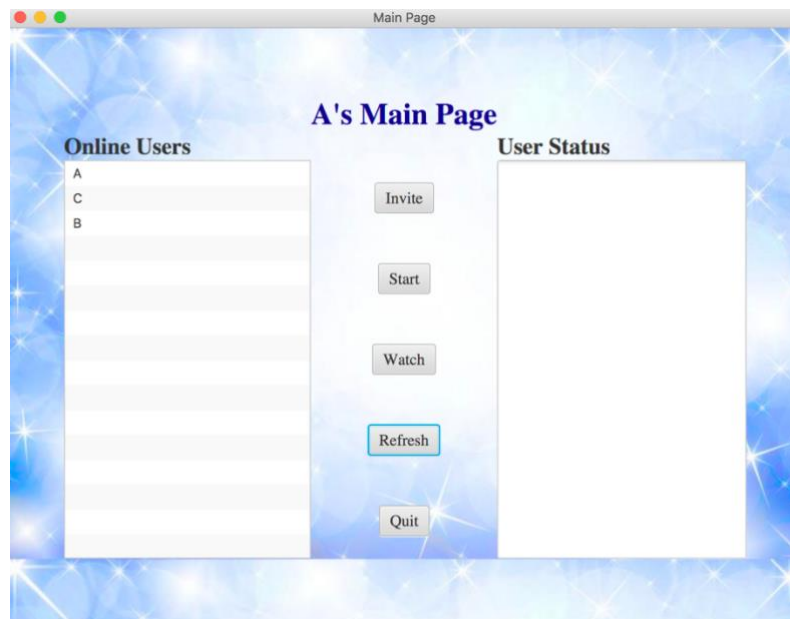


Figure 9: Update Online Users

If a user accepts the invitation, the corresponding username will be added to the User Status and available for all users who agree to join the game. Once all invited users agree to join, the game can start immediately. Otherwise, the inviter can press the "Start" button whenever he wants without waiting for all users' responses.

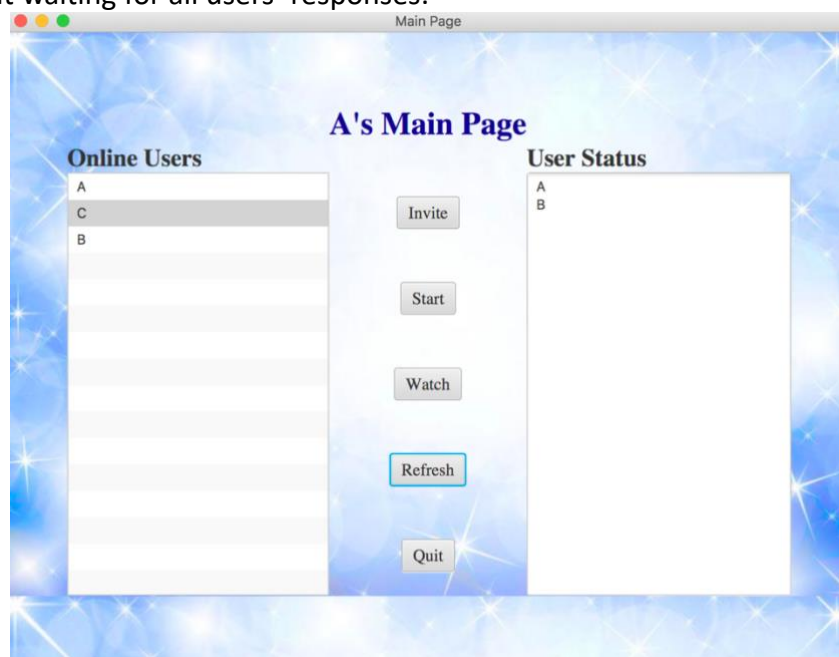


Figure 10: User Status

- Game

Once a game starts, all players will take turns to add a character each time. If it is the turn for

one player, he can undo his operation after dragging a character into the game grid. The final choice will be determined after click button "Confirm". Then a word from left to right and a word from top to bottom will be generated according to the location of this final chosen character. Everyone joining the game should follow the rule, otherwise, a warning window will pop up for reminding.

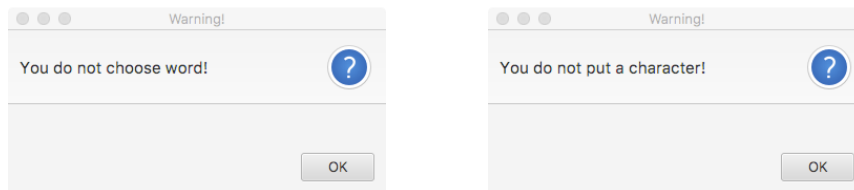


Figure 11: Choose A Word Error

When a player chooses a word to submit, it will be highlighted on the grid and could be seen by all players.



Figure 12: Highlight Chosen Word

If the chosen word is agreed by other players, the corresponding score will be given to the player and the updated score will be demonstrated on the Players' Score panel.



Figure 13: Update Scores

If a player does not want to add a character in his turn, he can press the "Pass" button. Once all players choose to pass, the game will end and the ranking window will be shown below. Then, all players go back to the MainPage and the function of invitation or start the game is also available.



Figure 14: Pass the Round

If a player wants to quit during the game, a "Quit" button is accessible for choosing. Once pressing, a confirmation window will pop up and if the player chooses "Yes", a ranking window will be shown and the whole game will be over.

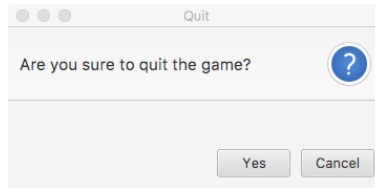


Figure 15: Quit Game

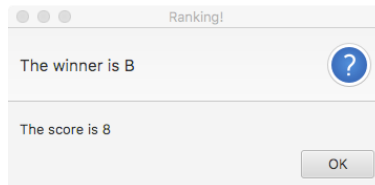


Figure 16: Ranking

5. UML Diagram

Server:

- Server: data stored in the form of Hashtable and ArrayList. Establish a connection and always listen to an incoming message from the client.
- Connection: resolve all the message received from clients. Handle these command then proceed to broadcast or unicast request to certain clients.

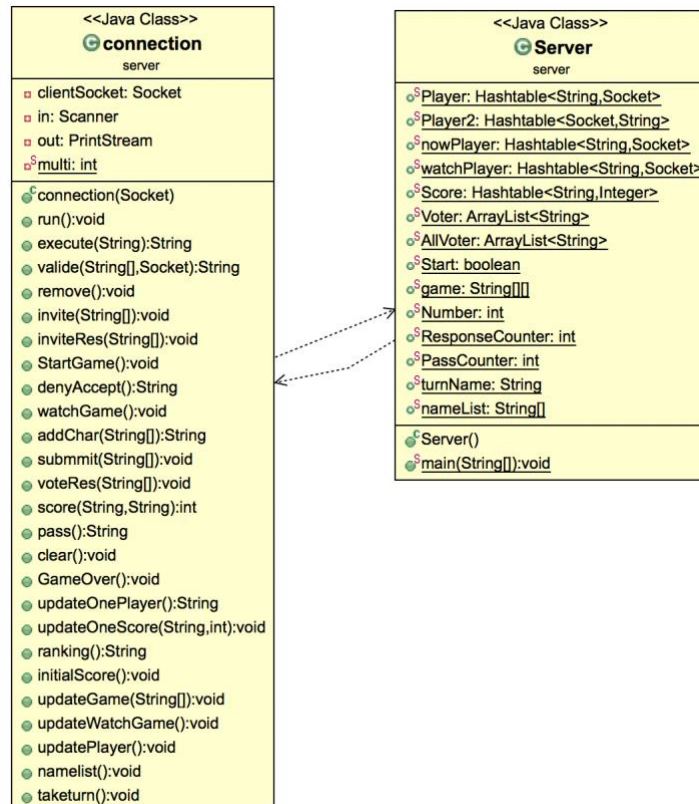


Figure17: UML of Server

Client:

- Login: enter the specified port number to make a connection between the client and the server.
- IdentifyUsername: validate whether the username has already existed in the list of online users.
- MainPage: display all the online users. Besides, all the successful invited users would be displayed in the User Status area.
- Game: the whole process of scrabble game, including choosing a character, choosing a word for players to vote, passing one's turn, etc.
- MessageListener: read a message from the server, then execute the related operations, like jumping into an alert window, writing a message to the server, etc.

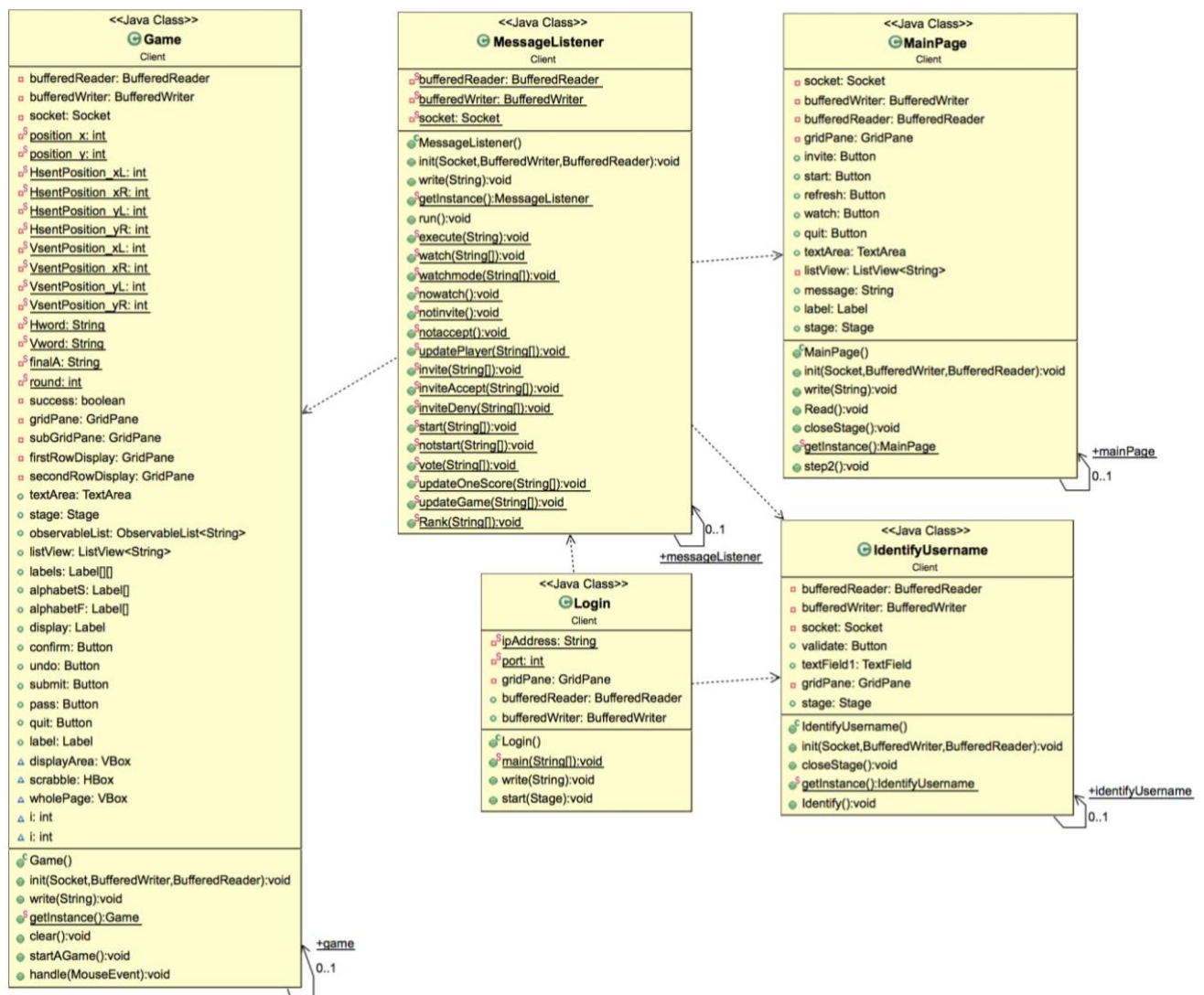


Figure18: UML of Client

6. Data Structure

- Hashtable

In the project, online users and game players are stored in the form of Hashtable. The reason

for choosing the Hashtable data structure is that this type of data structure can assure thread safe. In addition, an associative array abstract data type and a structure that can map keys to values are put into use.

As the following figure shown, there are two types of players. "Player" and "Player2" refers to online users whose specified sockets and name are a one-to-one correspondence. The "nowPlayer" is served as all the players who join the game. As for "watchPlayer", it refers to someone that has to choose watch mode instead of starting a new game when one game is in the process. "Score" is intended to store current scores of all the game players during one round of the game.

```
public static Hashtable<String, Socket> Player = new Hashtable<String, Socket>();
public static Hashtable<Socket, String> Player2 = new Hashtable<Socket, String>();
public static Hashtable<String, Socket> nowPlayer = new Hashtable<String, Socket>();
public static Hashtable<String, Socket> watchPlayer = new Hashtable<String, Socket>();
public static Hashtable<String, Integer> Score = new Hashtable<String, Integer>();
```

Figure19: Hashtable Data Structure

- ArrayList

In the project, anyone who has the qualification to vote for a chosen word would be stored in the form of ArrayList. Standard Java arrays are of a fixed length, which always leads to ArrayIndexOutOfBoundsException. On the contrary, the ArrayList data structure supports dynamic arrays to solve this kind of problem. That is why the ArrayList data structure is applied in the project.

The below figure shows that voters are classified as two types, "Voter" and "AllVoter". These two types of voters are indicators to record whether a game player vote or not.

```
public static ArrayList<String> Voter = new ArrayList<String>();
public static ArrayList<String> AllVoter = new ArrayList<String>();
```

Figure20: ArrayList Data Structure

- 20 * 20 Game Array

The 20*20 game array is used to store characters in corresponding positions. Game interface synchronization for all game players and people who join the watch mode can be ensured by broadcasting this array.

7. Critical Analysis

7.1. Pros

Advantages of this concurrent multi-clients scrabble game are as follows:

- By beautifying and editing, all the GUIs(Graphical User Interfaces) satisfy the requirement of User Interface friendly.
- TCP(Transmission Control Protocol) is a connection-oriented framework. Thus, the connection between client and server is more reliable than UDP(User Datagram Protocol). Furthermore, by a particular port, the TCP protocol would validate incoming data during a specified computer process.
- Excellent exception handling. A dialog would pop up while an exception is caught. For instance, after terminating the server, clients would receive an error message to

remind them that connection error happens.

- All the GUIs(Graphical User Interfaces) update timely. In other words, all users are given a low-delay, concurrent user experience.

7.2. Bonus

Certain creative ideas are pointed out in the project as bonuses, which are listed below:

- In the process of designing game, more interesting rules are implemented, like bonus score set in specific positions(four corners of the 20 * 20 grid).
- Watch mode is executed. To elaborate, once one game is in the process, other users can merely to watch this game instead of starting a new game by themselves.

7.3. Improvement

Admittedly, there are a series of advantages of this project. However, some improvements need to be implemented in the future project:

- Optimized scrabble game will be applied in the future. That is to say, all the online users can play multiple games at the same time.
- For the game design part, players can choose a word by directly clicking on the 20 * 20 grid and more fancy game rules will be implemented in the future.

8. Contribution

- Lixuan DING(927096)

For the coding part, the structure of server was sketched and executed. Data structures, such as Hashtable and ArrayList were used in the project. Code integration was ensured during the process of implementing the project.

For the documentation part, details about Data Structure part was described by using various vivid examples.

- Xinwei LUO(948602)

For the coding part, protocols between server and client for data exchanges were devised. While coming across some bugs, corresponding strategies were pointed out to handle these problems timely.

For the documentation part, System Components part was represented point by point clearly to list all the key components related to the project.

- Lei REN(950214)

For the coding part, all classes related to client were written, such as Login, IdentifyUsername, MainPage, Game, MessageListener and etc.. By writing the file "background.css", beautifying all the GUI(Graphical User Interfaces) were achieved.

For the documentation part, GUI(Graphical User Interfaces), UML Diagram of client part were explained in details and the whole document was formatted.

- Min XUE(897082)

For the coding part, GUIs(Graphical User Interfaces) of the whole project were sketched and designed. Basic functions of clients were proposed. The structure of the system was applied and protocols between the server and client for exchanging message were implemented. After testing the project, a variety of exceptions were handled.

For the documentation part, Project Architecture and UML Diagram of server and client were drawn. Introduction part and Results part were elaborated.