UNIVERSITY of LIMERICK

OLLSCOIL LUIMNIGH

Department of Electronic & Computer Engineering

EE4023 - DISTRIBUTED SYSTEMS

Final Project – Tic Tac Toe

Team Members:	**Ciaran Carroll**	**16181492**
	Eoin Watkins	**16187539**
Team Name:	**Brown Bread**	
Course:	**BSc in Computer Systems**	
Academic Year:	**2019/20**	

Final Project – Tic Tac Toe

Table of Contents

1. Introduction and Planning

1.1 INTRODUCTION

Our final project was focused on the game, Tic Tac Toe.

There were two main parts to the project which were two interfaces:

- One was to be written as a Java desktop application.
- The other a PHP driven website.

Both interfaces needed to work with the webservice which we were provided (TTTWebService) and with the version running in the lab on the demonstration day. The interfaces had to be able to talk to each other and with the interfaces of other project groups. SOAP would be used which is an XML-based protocol for accessing web services over HTTP.

The main requirements of the interfaces were:

- A main screen that loads when the interface starts offering access to create an account or login using existing credentials.
- A login screen to enter username and password.
- A registration screen to create a new account.
- Once logged in, a menu from where you would find:
  - A button to access a score system that will keep track of the number of wins, losses and draws the player has.
  - A button to access a leader board, showing the wins, losses and draws for each registered user.
  - A button to create a new game.
  - A table showing a list of open games that you could join
  - A mechanism for using this table to join an existing game.
- Threads to control who could make the next turn.

1.2 PLANNING

As a group with two members we decided it made the most sense for each of us to focus on one of either the PHP or Java interfaces and then combine efforts for the report. It was decided that Ciaran would focus on the PHP Client Interface and Eoin would work on the Java Client Interface. We would then come together when we felt we had created interfaces which could communicate with one another.

2. PHP Client - Ciaran

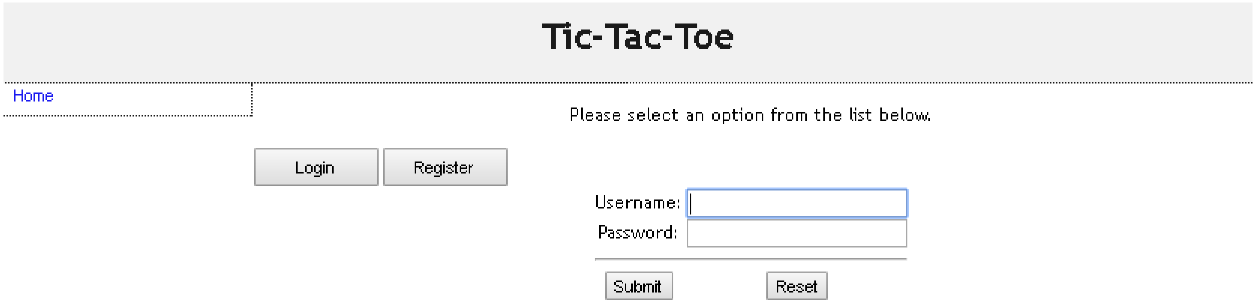
2.1 DESIGN

- HTML and JQuery would be used to provide the front-end of the PHP Client.
- “JQuery is a lightweight, "write less, do more", JavaScript library.”[1]
- This would be styled by a Cascading Style Sheet.
- PHP would be used to provide session variables and to communicate with the Web Service.
- JQuery would be used to update the interfaces after they are loaded and to call the PHP files and receive responses from these calls.
- These calls were made possible by AJAX.
- “AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology helps us to load data from the server without a browser page refresh.” [2]. https://www.tutorialspoint.com/jquery/jquery-ajax.htm
- Timers would be used update the game board, open games and to control who could make a move.
- All pages would be styled using a Cascading Style Sheet.

2.2 IMPLEMENTATION

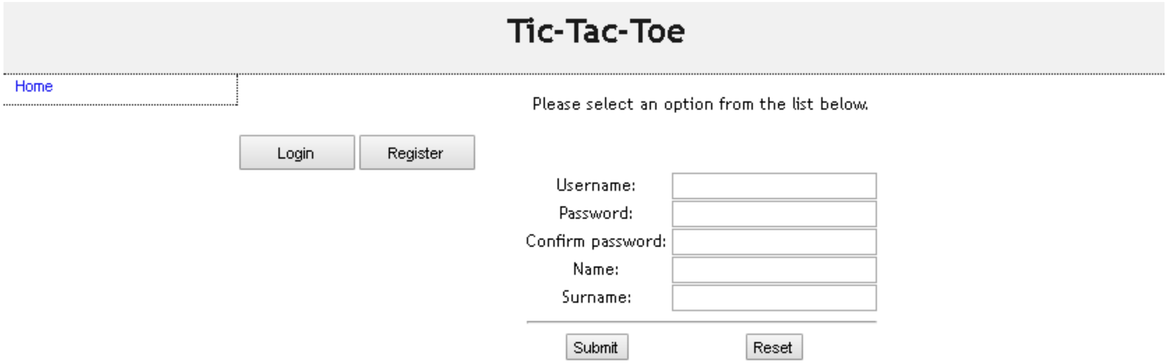
- Project Structure:
    - > **Tic-Tac-Toe**
        - game.php
        - index.php
        - leaderboard.php
        - logout.php
        - main.php
        - score.php
        - styles.css
        - **> action**
            - checksquare.php
            - checkwin.php
            - deletegame.php
            - getboard.php
            - getgamestate.php
            - getleaderboard.php
            - initialise.php
            - joingame.php
            - login.php
            - newgame.php
            - register.php
            - rejoingame.php
            - showopengames.php
            - takesquare.php
            - wsdl.php
        - > **images**
            - x.png
            - o.png

- UI and Code Snippets



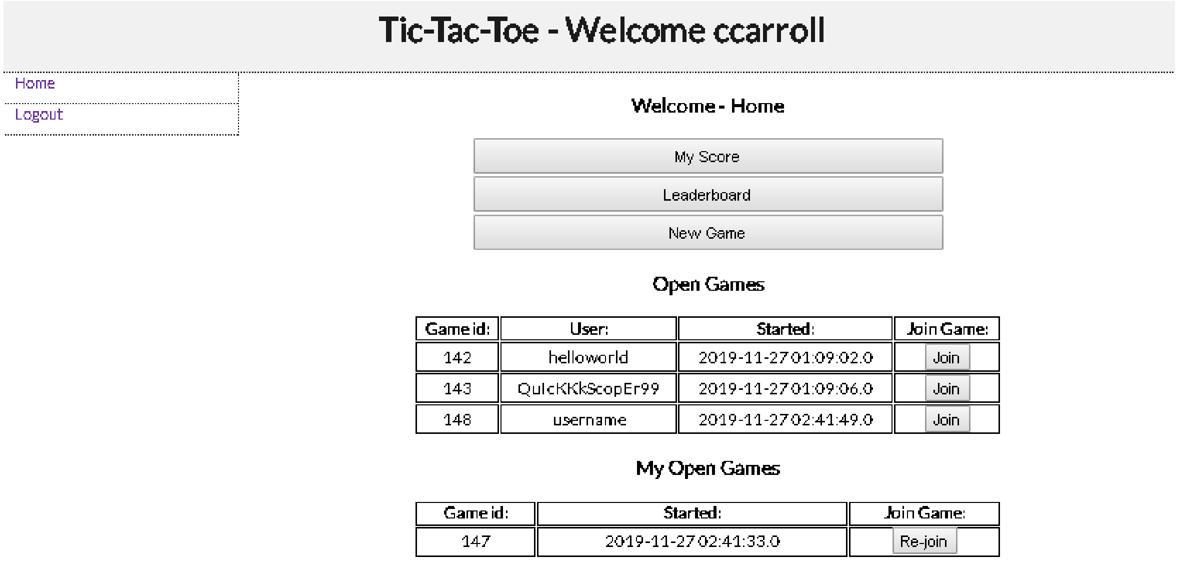
Please select an option from the list below.

- index.php, Login form.



Please select an option from the list below.

- index.php, Registration form.



Tic-Tac-Toe - Welcome ccarroll

Welcome - Home

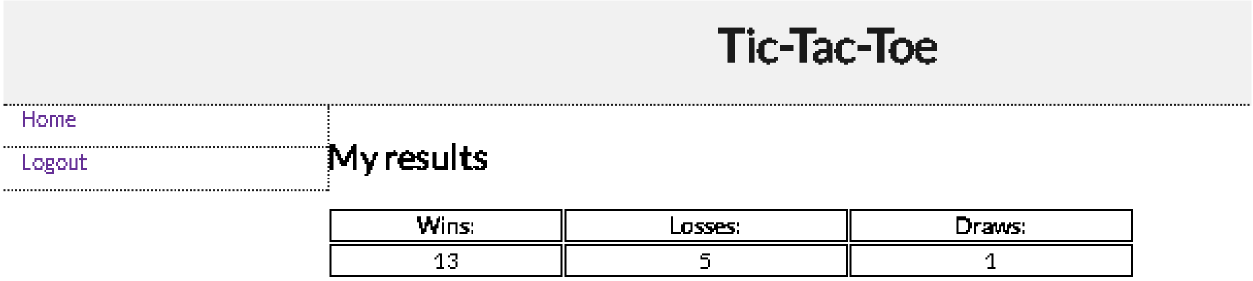
Open Games

Game id:	User:	Started:	Join Game:
142	helloworld	2019-11-27 01:09:02.0	Join
143	QuIcKKkScopEr99	2019-11-27 01:09:06.0	Join
148	username	2019-11-27 02:41:49.0	Join

My Open Games

Game id:	Started:	Join Game:
147	2019-11-27 02:41:33.0	Re-join

- main.php.

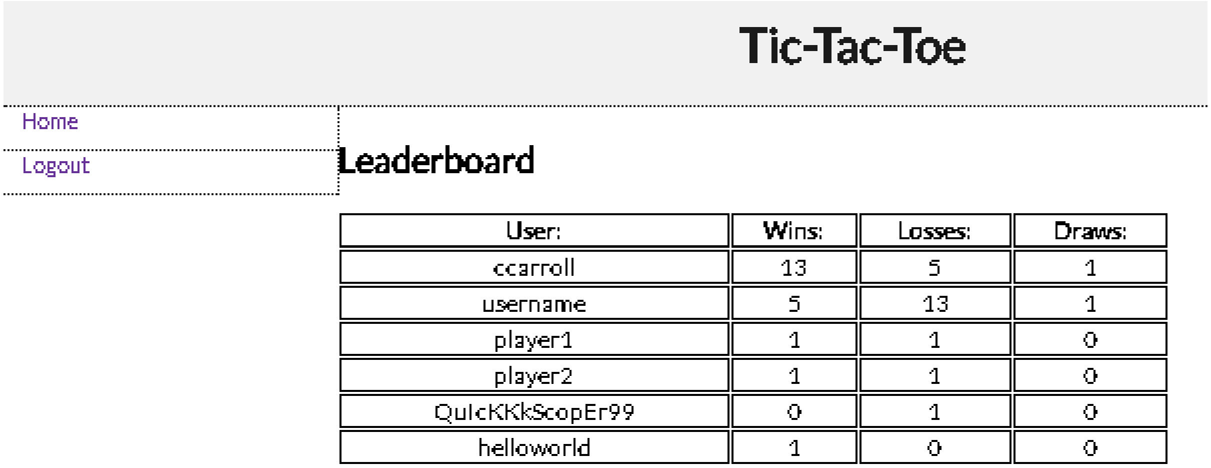


Tic-Tac-Toe

My results

Wins:	Losses:	Draws:
13	5	1

- score.php.

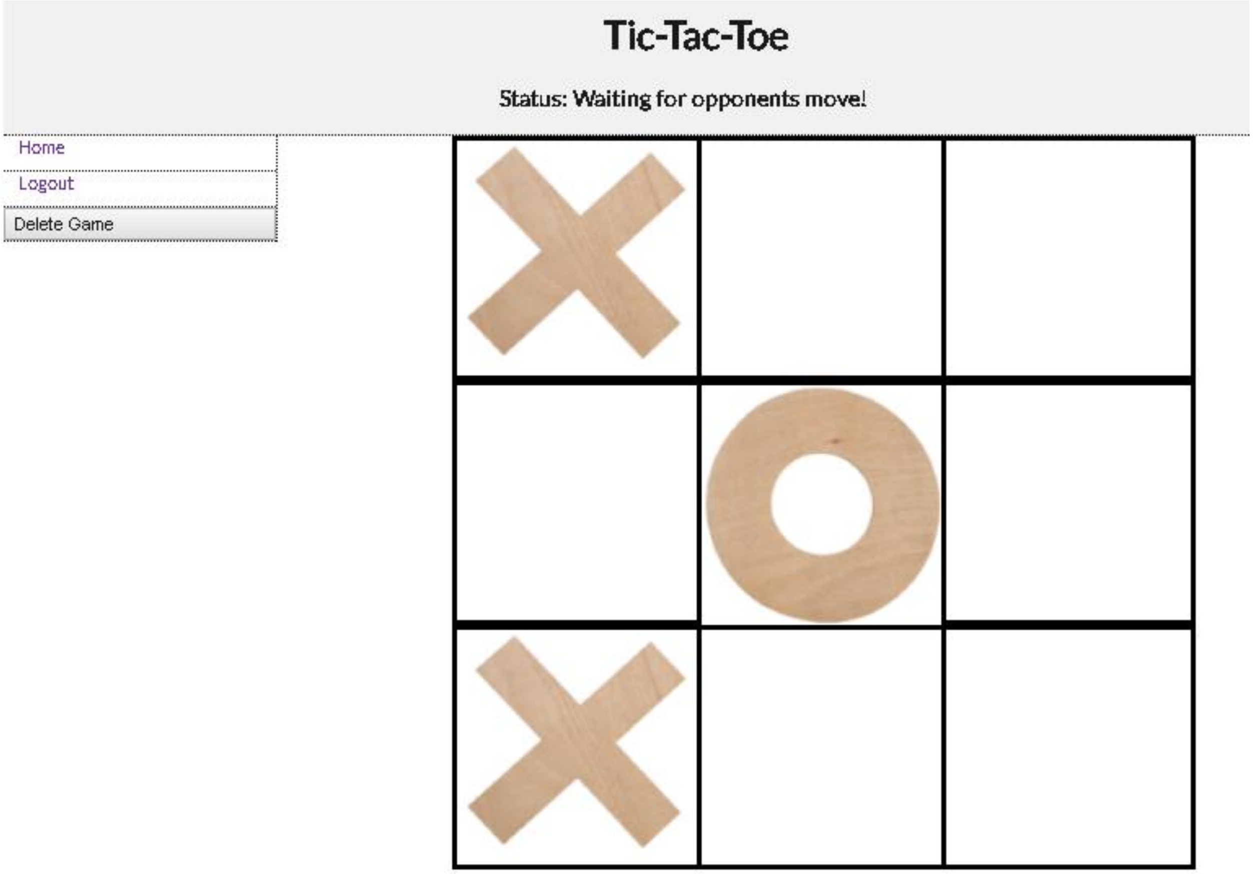


Tic-Tac-Toe

Leaderboard

User:	Wins:	Losses:	Draws:
ccarroll	13	5	1
username	5	13	1
player1	1	1	0
player2	1	1	0
QuIcKKkScopEr99	0	1	0
helloworld	1	0	0

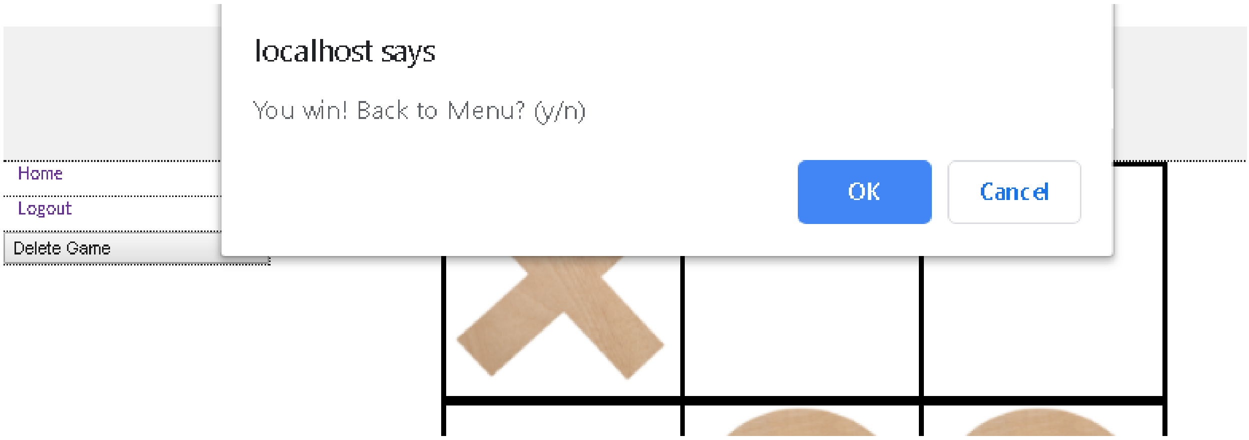
- leaderboard.php.



Tic-Tac-Toe

Status: Waiting for opponents move!

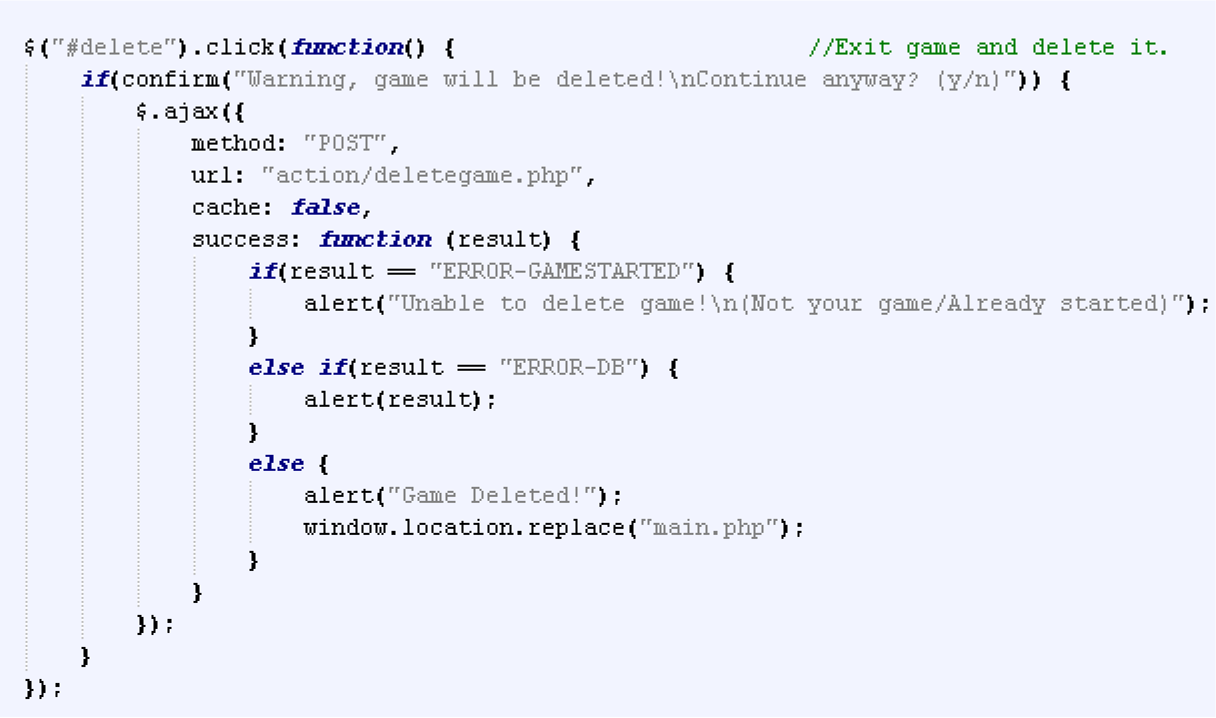
- game.php.



localhost says

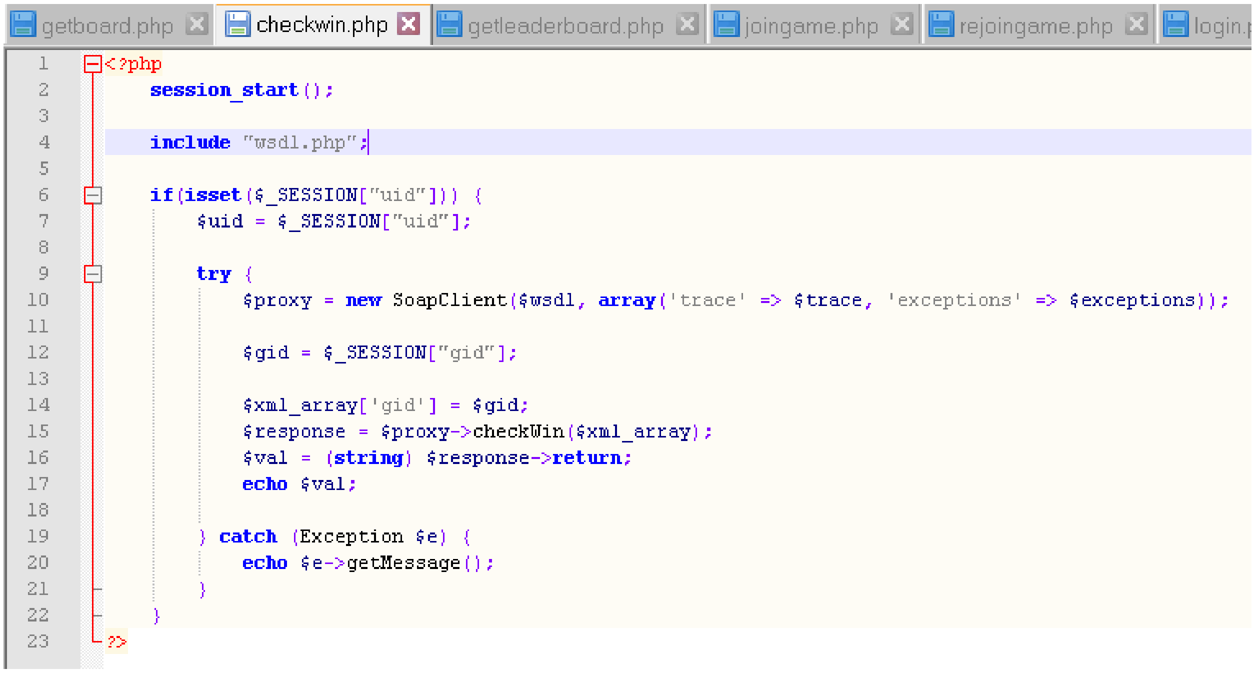
You win! Back to Menu? (y/n)

- Example of alerts.



```
$("#delete").click(function() {                         //Exit game and delete it.
    if(confirm("Warning, game will be deleted!\nContinue anyway? (y/n)")) {
        $.ajax({
            method: "POST",
            url: "action/deletegame.php",
            cache: false,
            success: function (result) {
                if(result == "ERROR-GAMESTARTED") {
                    alert("Unable to delete game!\n(Not your game/Already started)");
                }
                else if(result == "ERROR-DB") {
                    alert(result);
                }
                else {
                    alert("Game Deleted!");
                    window.location.replace("main.php");
                }
            }
        });
    }
});
```

- Example button function with ajax call.



```
<?php
    session_start();

    include "wsdl.php";

    if(isset($_SESSION["uid"])) {
        $uid = $_SESSION["uid"];

        try {
            $proxy = new SoapClient($wsdl, array('trace' => $trace, 'exceptions' => $exceptions));

            $gid = $_SESSION["gid"];

            $xml_array['gid'] = $gid;
            $response = $proxy->checkWin($xml_array);
            $val = (string) $response->return;
            echo $val;

        } catch (Exception $e) {
            echo $e->getMessage();
        }
    }
?>
```

- Example of PHP interaction with Web Service.

3. Java Client - Eoin

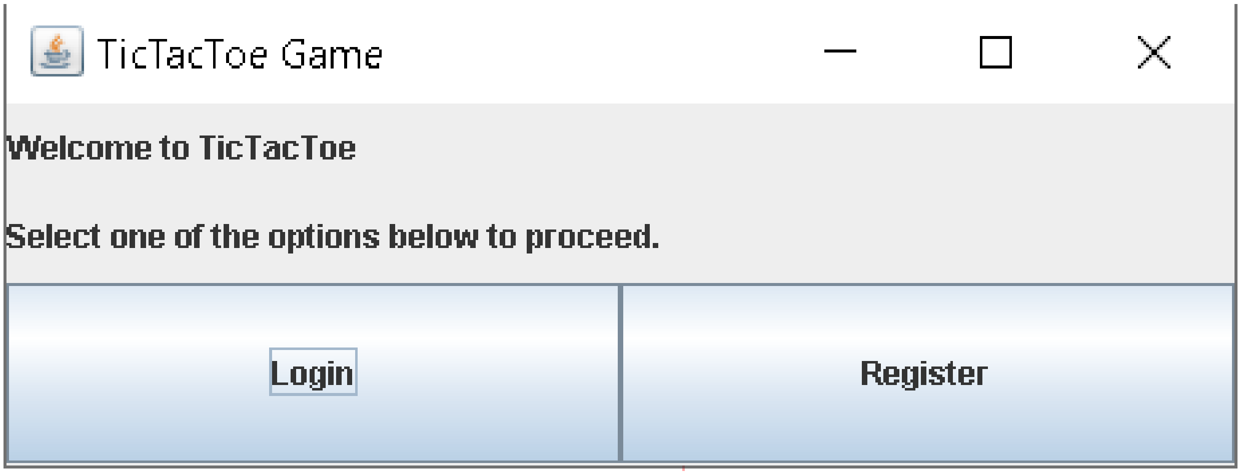
3.1 DESIGN

- Client would be implemented as a netbeans project.
- Java Swing Frames would be used to create the user interface.
- The design would hold two parent windows which held their own child windows.
- Firstly, the start window which would make the login or registration windows visible for the user.
- If successful, the user would then be transferred to the menu window.
- This would hold buttons which would make the score, leader board and game windows visible.
- Exiting these windows would return the user to the menu window.
- Data would be transferred between windows by passing it in the parameter list when creating a new frame.

3.2 IMPLEMENTATION

- Project Structure:
  - > **Tic-Tac-Toe**
    - TicTacToeGame.java
    - LoginFrame.java
    - RegisterFrame.java
    - MenuFrame.java
    - ScoreFrame.java
    - LeaderboardFrame.java
    - GameFrame.java
    - NewGame.java

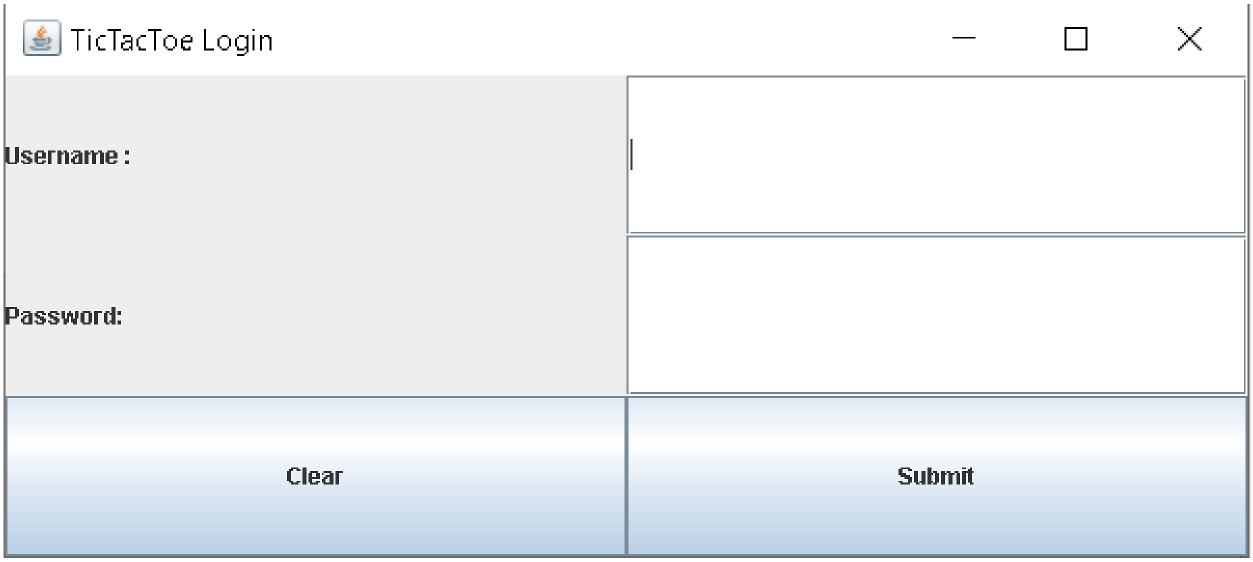
- UI and Code Snippets:



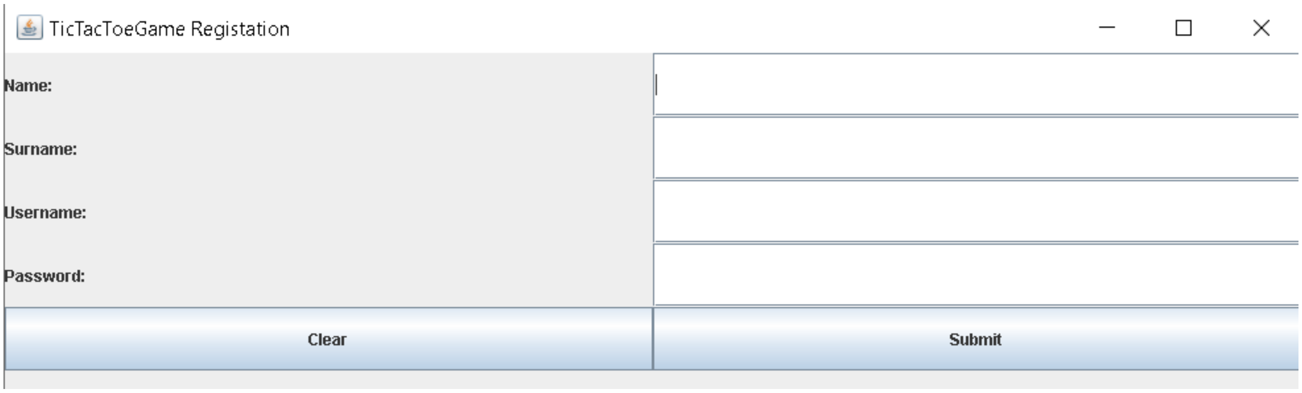
Welcome to TicTacToe

Select one of the options below to proceed.

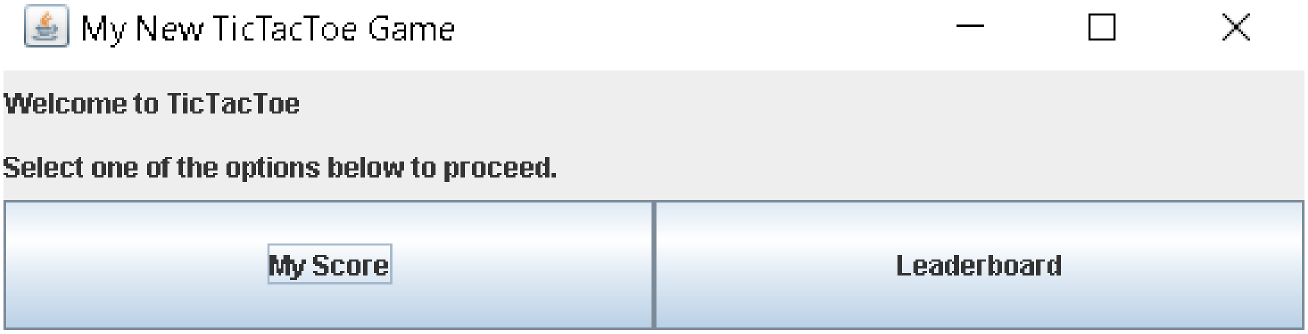
  - TicTacToeGame.java.



  - LoginFrame.java.



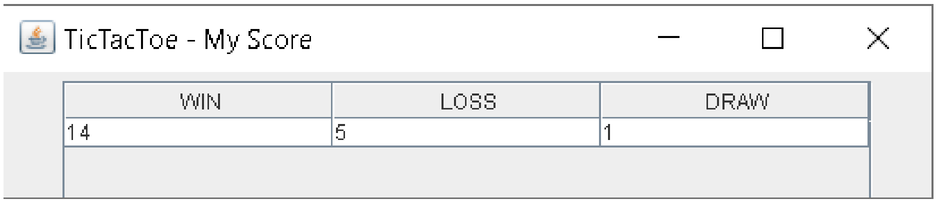
- RegisterFrame.java.



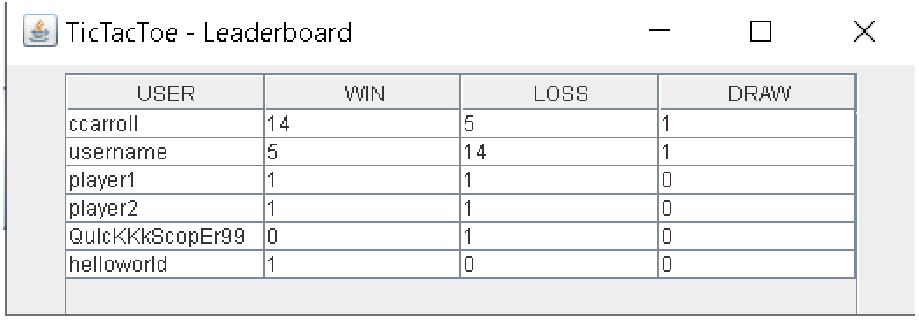
Welcome to TicTacToe

Select one of the options below to proceed.

- MenuFrame.java.

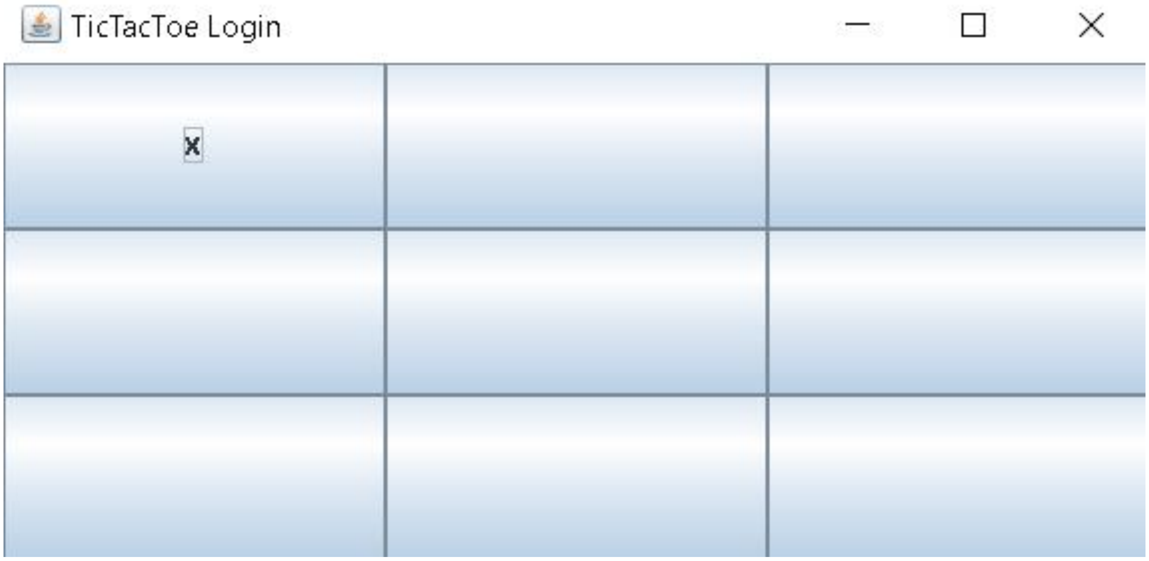


WIN	LOSS	DRAW
14	5	1

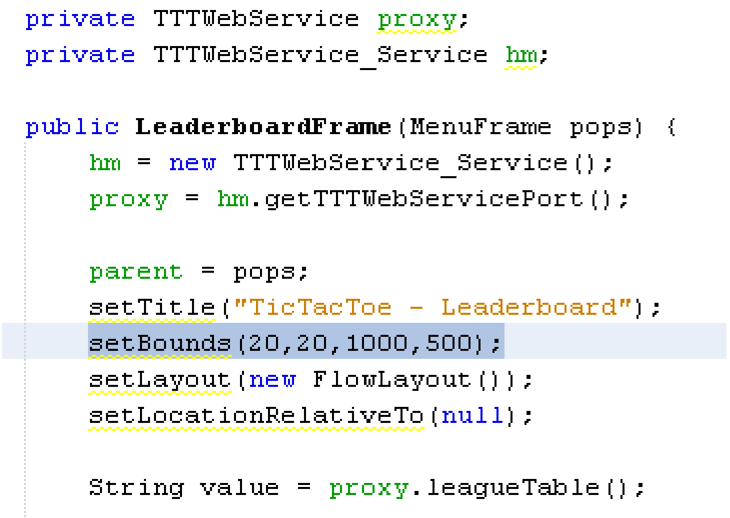


USER	WIN	LOSS	DRAW
ccarroll	14	5	1
username	5	14	1
player1	1	1	0
player2	1	1	0
QuIcKKkScopEr99	0	1	0
helloworld	1	0	0

- ScoreFrame.java and LeaderboardFrame.java.



- GameFrame.java



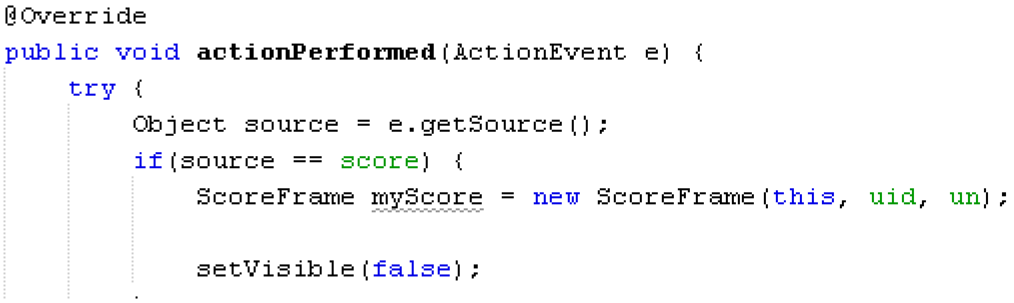
```
private TTTWebService proxy;
private TTTWebService_Service hm;

public LeaderboardFrame(MenuFrame pops) {
    hm = new TTTWebService_Service();
    proxy = hm.getTTTWebServicePort();

    parent = pops;
    setTitle("TicTacToe - Leaderboard");
    setBounds(20,20,1000,500);
    setLayout(new FlowLayout());
    setLocationRelativeTo(null);

    String value = proxy.leagueTable();
```

- Example of connecting to Web Service and calling its method.



```
@Override
public void actionPerformed(ActionEvent e) {
    try {
        Object source = e.getSource();
        if(source == score) {
            ScoreFrame myScore = new ScoreFrame(this, uid, un);

            setVisible(false);
```

- Example of creating child frame and passing data as parameters.

4. Testing and Conclusion

4.1 TESTING

- Testing for each client was implemented iteratively. We would test each part as we worked and would only move onto the next part when the previous was functioning as required. Use cases were kept in mind while running these tests in order to make our systems as enjoyable for the user as possible.
- We then tested our clients against the projects of other groups to make sure they were able to communicate with other peoples designs.

4.2 CONCLUSION

- Overall, we found controlling who could make a move the most difficult aspect of the project but also the most rewarding when seen working.
- If we were to do the project again, we would spend more time on the design of the system focusing on use cases and having a well-drawn out System Architecture for each client so as not to waste too much time updating the layout during the implementation phase.