**Final Technical Report**

**Multiplayer Tic-Tac-Toe Gaming Application**



University of Windsor

March 28th, 2017

**0360656-1-2017 – Advanced Systems Programming - Winter Semester**

**GROUP 21**

By -

Vatsal Jani   ( 104672987 )

           Aditya Taneja ( 104604869 )

           Raghavnender Yadav ( 104696310 )(

Prabhleen Singh  ( 104683866 )

104696310)

Table of Contents

[1. Abstract 1](#_Toc478509040)

[Vision and Goal 1](#_Toc478509041)

[2. Introduction 1](#_Toc478509042)

[How the system Works 1](#_Toc478509043)

[Scope of Project 2](#_Toc478509044)

[Risk Management 2](#_Toc478509045)

[3. Requirement Analysis 2](#_Toc478509046)

[Functional requirements 2](#_Toc478509047)

[Use Case story 3](#_Toc478509048)

[4. Design 5](#_Toc478509049)

[Setting up the version control management system (GitHub) 5](#_Toc478509050)

[5. Prototype and Design !!--- EVERY SNAP NEEDS TO BE EXPLAINED ---!!! 6](#_Toc478509051)

[6. Future Scope of the Multiplayer Tic-Tac-Toe Project 11](#_Toc478509052)

[Technical tools used 11](#_Toc478509053)

[7. Conclusion 11](#_Toc478509054)

[8. Acknowledgement 12](#_Toc478509055)

[9. Reference 12](#_Toc478509056)

# Abstract

The project “Multiplayer Tic-Tac-Toe game” focuses upon the implementation of a client-server architecture where there are multiple clients (Players) and a single server monitoring the clients. **Risks** overcame in achieving the above stated goals are as follows –

1. Learning Android Development to develop the client side.
2. Timely acquiring the skills to implement Socket to enable the conglomeration of the client and the server side.
3. Implementing the client and server sides together to form a client-server architecture.
4. Removal of various probable bugs and errors to assure software quality within the initial and final time-box.

## Vision and Goal

* This project implements a multiplayer distributed gaming application based on a client-server architecture.
* The server side shall be accepting multiple clients as participants and their related processes.
* The client side will be participating through a separate machine. The client side has been implemented in the C language.

It will be usable through an Android device. The server side has been developed in C language.

# 2.Introduction

## How the system Works

The system is an interactive application that would start the server and also ask the Player(clients) to join the server and enter the player name. The system will then initialise the game for Player 1 and put the game on “Wait” status till Player 2 joins from an Android device or running client program from command line on the other side.

When the server gets two clients it will start the game. First the server will send the players their opponents name and then will act as mediator to send moves to each clients.

Example:

If the player1 enters its move in the first column then the server will let the player2 know that player1 has made a move and now its player2 turn to make the move.

## Scope of Project

This project scope would focus on the match between two Players. Furthermore, tournaments can be added to organize several matches between different players in a proper hierarchical manner of victorious players in each match. The team made use of the **GitHub** as the software version control method.

Managing the risks tend to maximize our changes to have a successful project so that a plan on how to manage them can also be built.

The risk list is as below -

## Risk Management

* Project was developed and deployed in Android Studio . Team had to invest few hours to get accustomed to the AS IDE and have an acumen command over it.
* There were insufficient helpful resources and technical support available to meet our needs.
* With time, all developers became familiar with the basic features of GitHub.
* Schedule was very tight once most technologies were new to everyone but all planned functionalities were developed and deployed in the presented prototype.

During the first meeting, many issues were found. Most of them were directly related to the fact that the schedule is very time bound and most of the technologies and development environment were completely new to the team of developers. In order to minimize these risks, all members of the team had to focus on learning and getting the initial development set up.

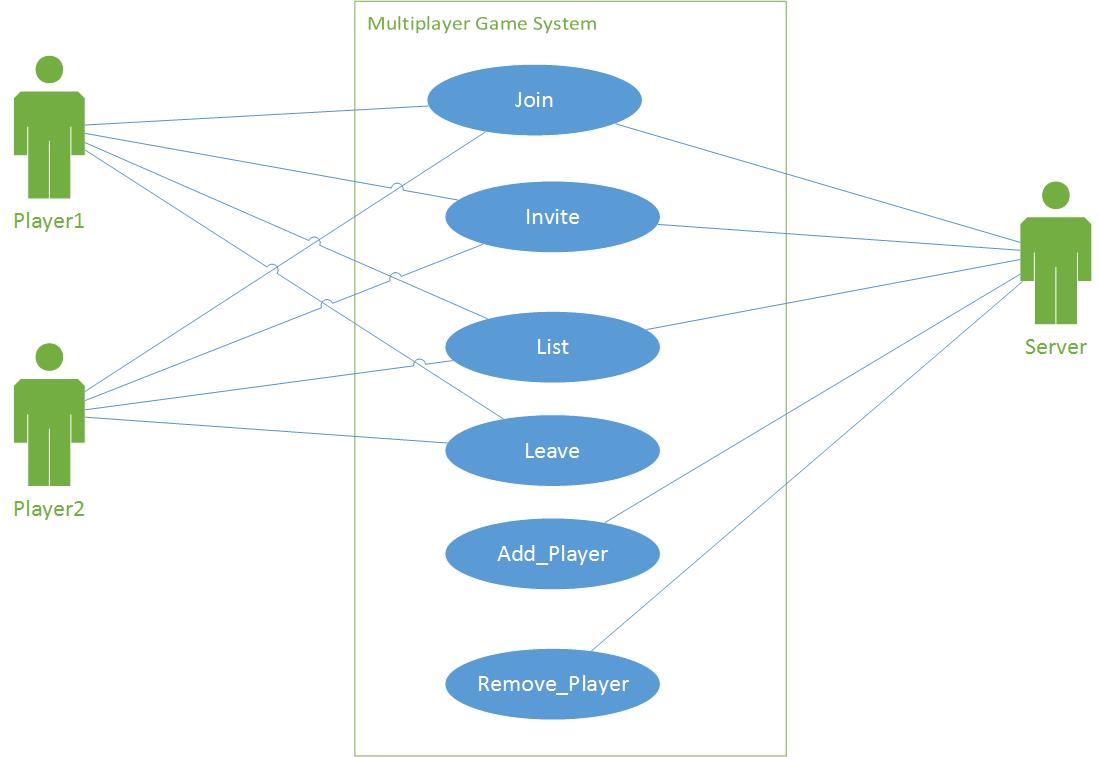
# 3.Requirement Analysis

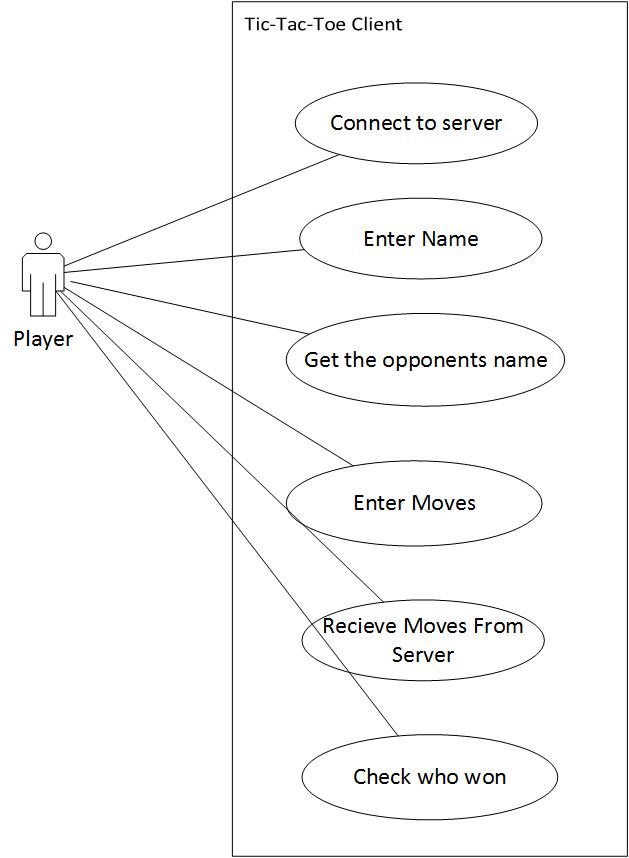
## Functional requirements

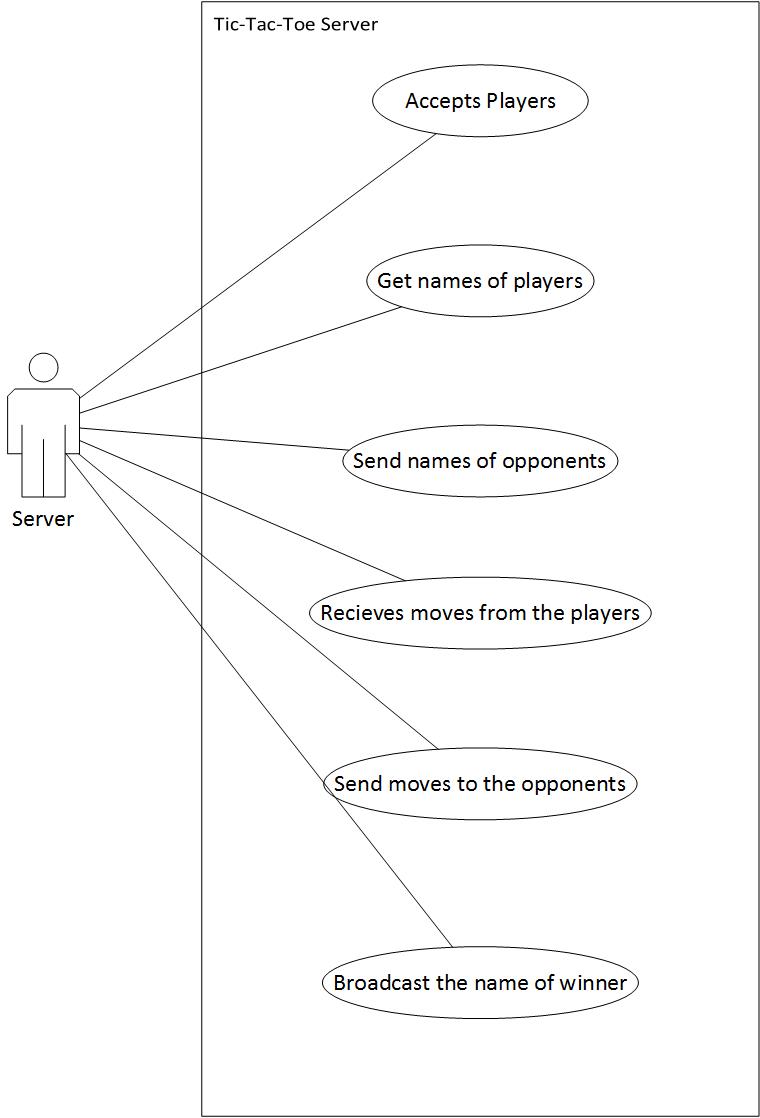
* + Android Studio that is an application for Android application development that allows the programmers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure that relates to developing and launching the application.
  + Use Case story

Player 1 is a real-time user that gains access to the system by logging in and the system immediately initiates a game waiting for another player.

The system now gets the information (name of the player). When the information is retrieved from Player 1, the system puts the game on “Wait” status, waiting for Player 2 to join the game.







**Glossary**

A list and dictionary is extremely important in order to guarantee that everyone involved in the project will have a clear understanding of some of the key words.

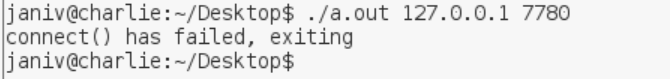
1. **Connect:-** Clients will connect to the server using ip address and port number.
2. **Accept Clients:-** Server will accept two clients at a time and start the game.
3. **Send and Receive:-** Clients will send their names to the server. Now server will send the name of the opponent to the respective clients. Also every client will send their moves to the opponent via server.
4. **Announce the Winner:-**When the game is over, server will announce the winner to both players**.**
5. **GitHub –** Remote repository based on Git from source code and version control.

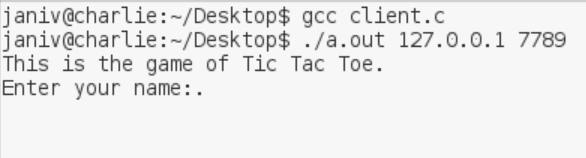
# 4. Design

## Setting up the version control management system (GitHub)

GitHub is a version control management system, which is used to maintain the source code and allow various operation such as versioning, verifying changes, tracking issues etc. This tool helped us in working simultaneously on the same source code.

An important part of our Agile software development process was to guarantee that all developed code was tested before merged into GitHub repository. First of all we designed the server in such a manner that it should just facilitate the interaction between the two players in simple manner. So we kept the server functionalities as low as possible. Then we design the client machine in which we added the most of the code. For c client we developed different self developed functions like game initialization, win-check function to check if the player wins or not and for android client we used the same concept using classes. Everytime we wrote the code we checked it by passing the dummy values and it was a manual test.



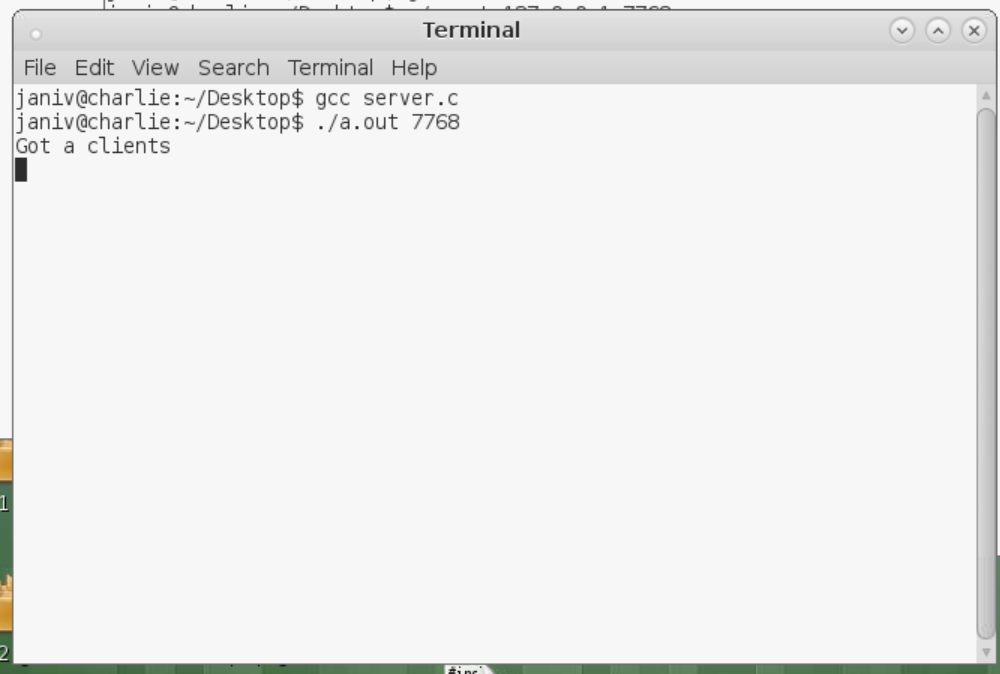


In the first screenshot shown above we tried to connect to the server using a wrong port number and the error message was displayed correctly. When we passed the true value the game starts and asks the player to enter his/her name, which is shown in the second screenshot. This is how we tested our project each time we updated the code.

# 5.Prototype and Design

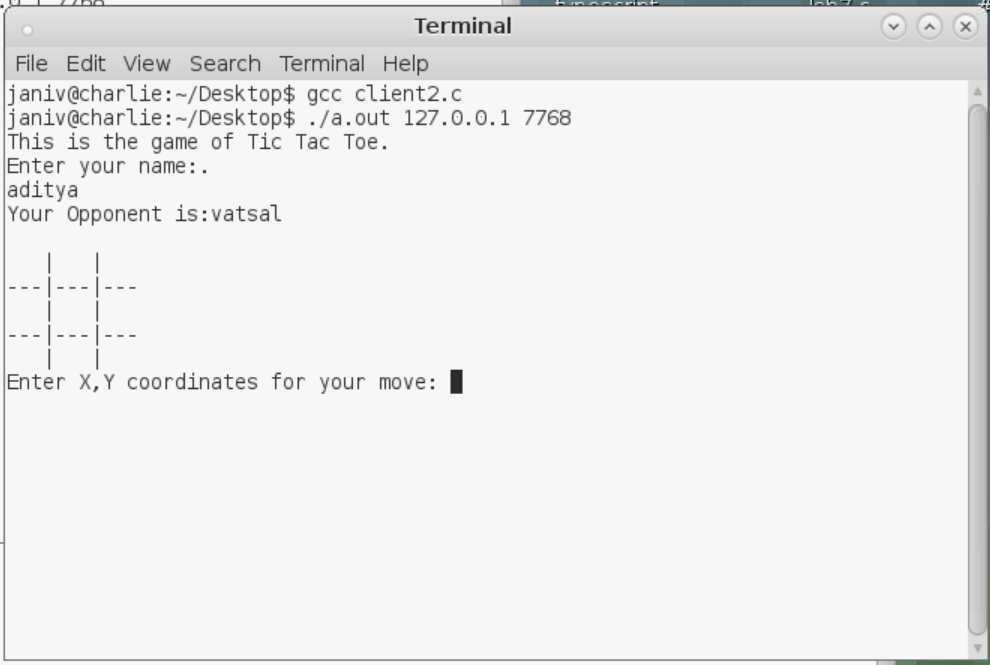
The following are the screenshots of a working sample of our project. The screenshots are arranged in the ordered of which the user can perform the activities.

**STARTING THE SERVER**



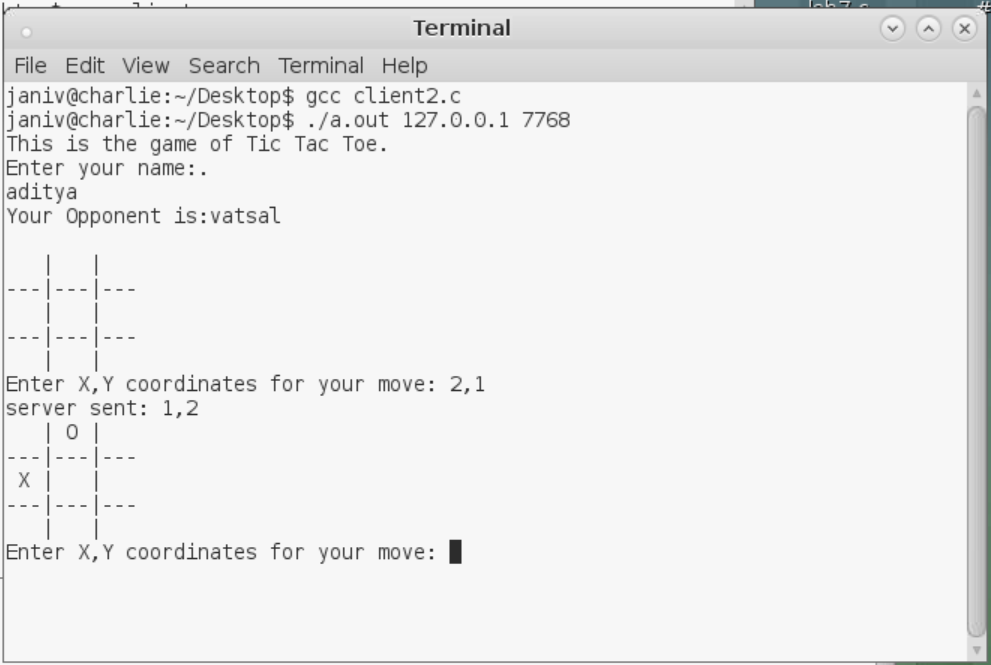
* To start the server just run the server file and pass the port number on which you want your server to run.
* This is the screen shot of server when it gets 2 clients.

**STARTING THE CLIENT**

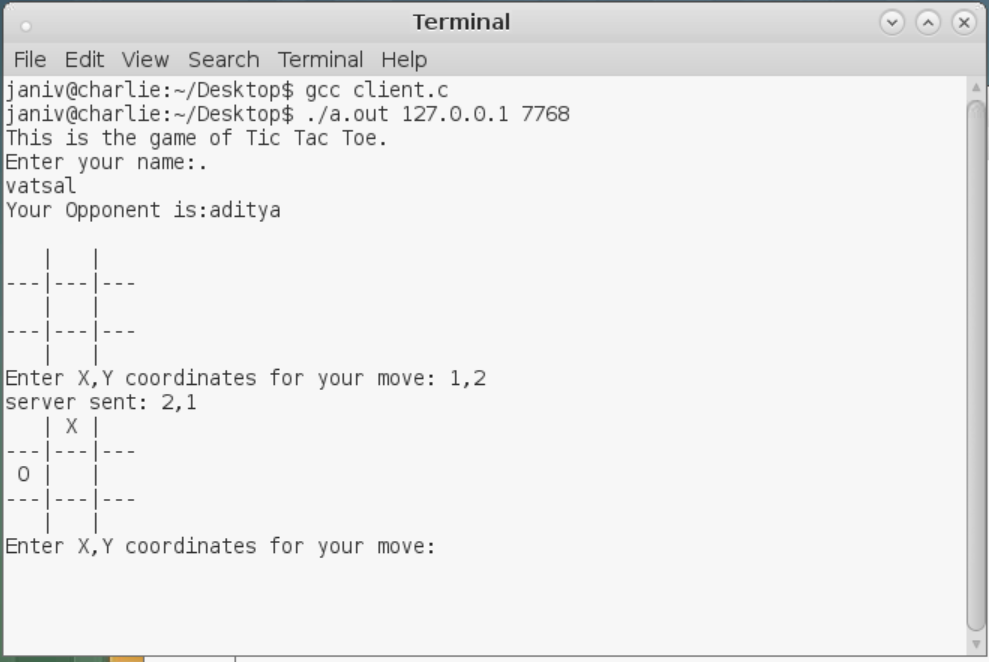


* To start the client just run the client file and pass the port number on which server is running and also the ip address.
* When the client will connect to the server, it will ask the player to enter the name. Once both the players enter the name , their names will be sent to their opponents as opponents name via server.

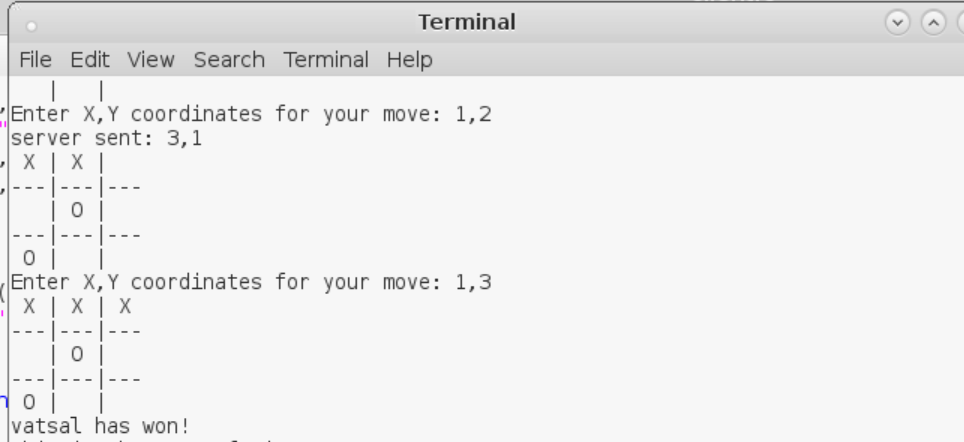
**SERVER READS CLIENT’S MOVES**



* When the player enters its move, it is sent to the server and server will pass that move to its opponent. This way server facilitates the whole game.

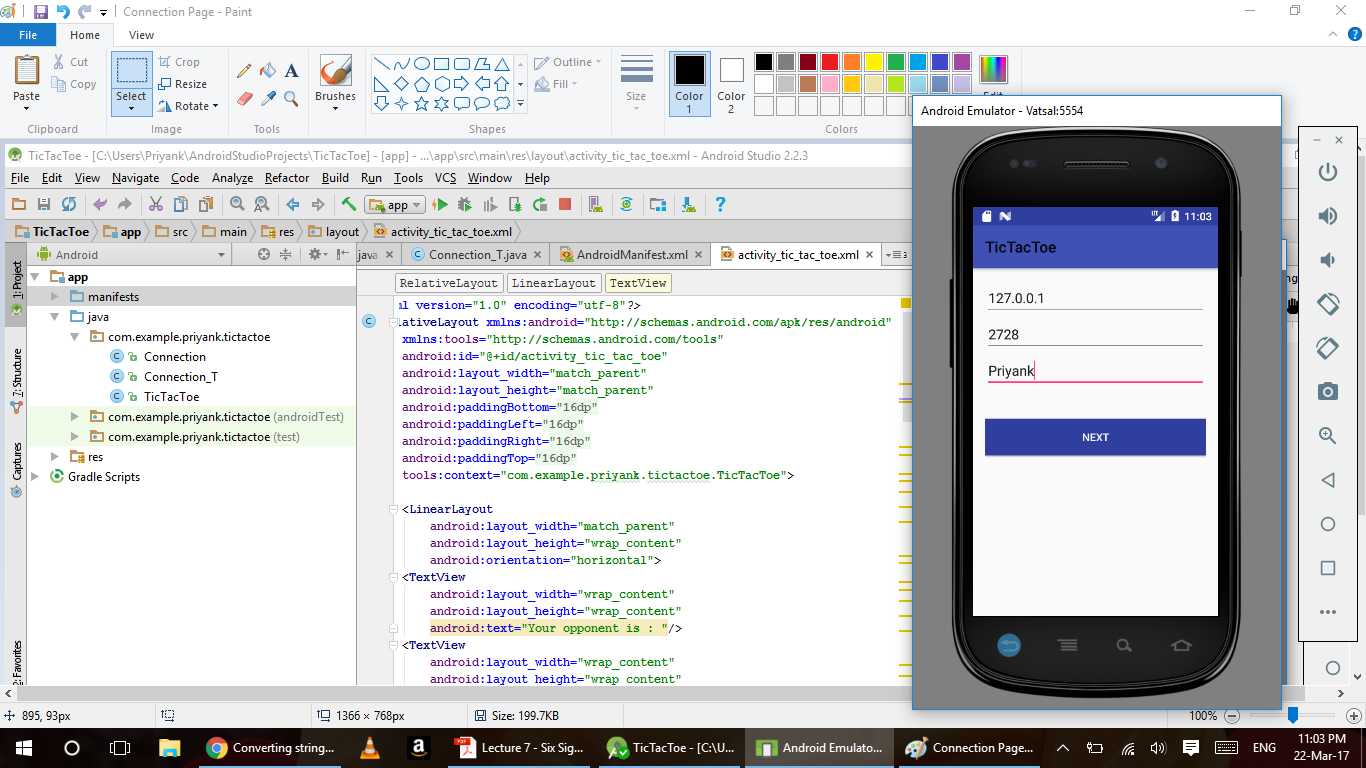


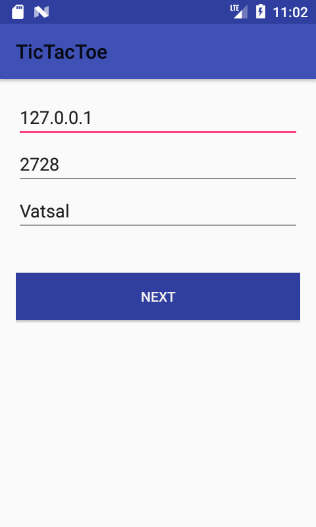
**SERVER WILL BROADCASTS THE WINNER OF THE GAME**



* At the end the server will broadcast to the players about the winner.

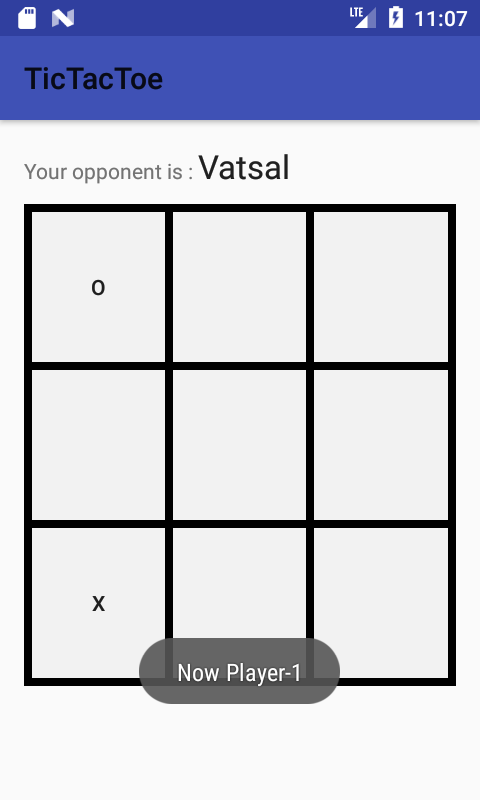
**ANDROID CLIENT MAIN PAGE**





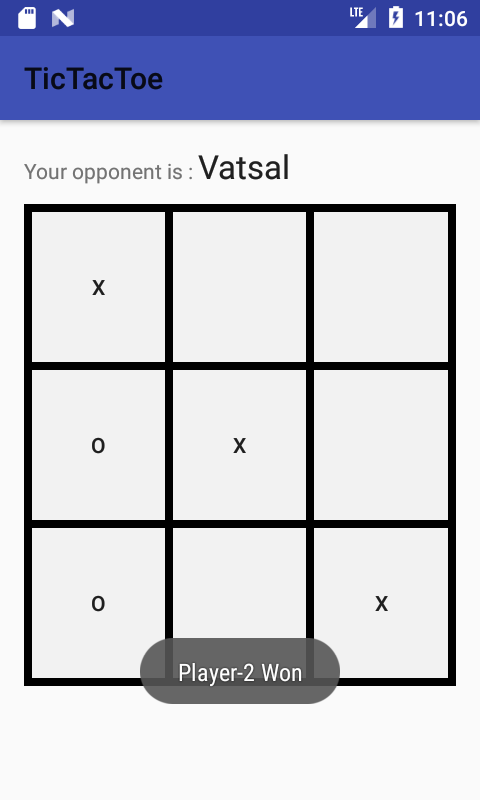
**🡨----------ANDROID CLIENT GAME PAGE**

* This is the starting page of android application. The player needs to enter the ip address, port number and his/her name.



* The game will start on the successful connection with server. The players can enter there moves and there opponents moves are also displayed on the game board. To develop the game board we have used buttons and set them in a grid.

**ANDROID CLIENT WINNER ANNOUNCEMENT**



* Similar to the c client the android client also broadcast the results at the end of the game.

# 6.Future Scope of the Multiplayer Tic-Tac-Toe Project

In the near future, the project may include tournaments where multiple Players can take part till the Final Match between two best scoring players takes place. This shall be done by a hierarchical B-Tree implementation of Players, thereby increasing the scope of the application to a great extent.

## Technical tools used

The following technical tools were used in the implementation of the project:

1. Android Studio 2.3.
2. Star UML.
3. GitHub repository (for Software Version Control).
4. SSH.

# 7.Conclusion

Along development, we faced several server errors while deploying and testing our system in Android Studio. Application would not start up or show internal errors that did not mean anything to us as developers but after comprehensive testing we were able to perform proper debugging to ensure smooth functionality across the system.

The project prototype is an application that can be improvised in terms of its GUI to make it look more appealing to the players.

Assessment of Product Quality, Dependability and Security –.

Product Quality of the Multiplayer Tic-Tac-Toe application was ensured by rigorous testing of the front-end and back-end functionalities.

Since, dependability of any software is directly related to the quality of the software product, we have made sure that the quality of the software application is optimum and up-to the mark.

We all at team **Group 21 for Multiplayer Tic-Tac-Toe application** collaborated and contributed equally to create this project, while, sometimes focusing more on each other’s strengths. Our extensive teamwork and collaboration aided in the timely execution and implementation of the project with optimum quality, performance, and dependability.

# 8.Acknowledgement

1. [The GitHub Logo](https://github.com/) © 2017 GitHub Inc. All rights reserved.
2. Android Studio is the official IDE for Android. It is designed to accelerate your development and help you create the highest-quality apps for every Android device.
3. © 2014-2016 MKLab, Co. All rights reserved.

# 9.References

1. <http://www.waleedkhan.com/tic-tac-toe-online-c-socket-programming/>
2. <https://developer.android.com/reference/java/net/Socket.html>
3. <http://www.cppforschool.com/project/tic-tac-toe-project.html>
4. Github - Adityataneja94/Project1