

## **Technical & Functional Specification**

Student name: Kateris Martinez

Class: 2<sup>nd</sup> year Web design and front end

Project title: TicTacToe,

Technical & Functional Specification

Submission date: March 16, 2017

Assessing Tutors: Scott O'Malley

## Technical & Functional Specification

At the begin, I was thinking that do and app would be very easy, just a little of code to get some principal functions and that it. I do never touch "IONIC" before or hear about it, so as soon I get the information about the tool I start looking on internet, I did find "**Ionic Creator**" and was a very nice tool to design on it, however I must use "**Ionic CLI**" to do my app. And this was a little more complicated, the difference between both is that while with Ionic Creator you have 2 options to design your app:

1. As a simple "mockups take and draw" and After that download the code.
2. You can use the mockups options and do some specific changes with the HTML option on the system.

While with Ionic CLI you most code all as the old school and install all the necessary tools on your laptop, the good notice here is that with CLI I found more precision on my design and I felt that I did learn more.

I did try to familiarize myself with the most commons keys to be used, and after that I did start my design and following the Ionic Tutorial. I must say that all this is something that is easy say on 2 lines but took 5 days for me understand how works Ionic CLI before can start design on it.

After understand Ionic I was looking for the **APIs** than can be useful for my app on the platform <https://market.mashape.com/>. My tree options where:

- **Facebook profile pic viewer** from Codefreak
- **Tic Tact Toe from** Stujo
- **phone validation**

I was thinking on use Facebook API or Phone validation to get the information necessary to "login" a user, However the priority was do the game and for that reason I did choose the second option.

**Tic tac toe** from stujo, This API make my life most easy, because I just need configurated it and add some extra code to do the main characteristic on my game works, This API work as a probability machine where you bring an action and them you get and answer from the server. So, I Must configurated the answer to be reflected on and specific position of the table game and I did this using the grid system and adding a value to each grid.

The next step was done the configuration to get print “X” or “O” depends of the player to do my life easy I assigned to the machine the “O” movement on the function “**machineMove**” To the human interaction we use the function “**move**” This mean that each time that we have an interaction than is not coming from the server the game must print “X” on the table.

“**getStatus**” is a very important function, with this we send the answer to the server of which position are free on the table and on this way, the API can analyse the next move. With this function, we can also check if all the boxes on the table where occupied with a character, on this way we can know when the game is over and send the Alert correspondent with the function “**ShowAlert**”.

There is 3 different kind of alert:

1. Game Win, Play again
2. Game Loose, play again
3. Game over, play again

Each alert was configurated with the “checkWinner” or “checkIfOver” function depends of the situation.

**checkIfOver:** movements counter, Limit the movements to 9 and when the limit is reached send the alert **Game over**.

**checkWinner:** Every time the player moves, we check if he/she/it has won and them send the alert. **You Win / You lose**. The function for this return a true or false result after check the possible positions where a player must be localized to can win a game.

“**Initialize**” is the function responsible for start each game, this function is call each time that the user press the button of “play” when open the app, win, lose or get the “over” answer on a game.

