**Requirements Document**

**Table of Contents**

[Modification History 3](#_Toc444295994)

[Domain Knowledge 4](#_Toc444295995)

[Glossary 4](#_Toc444295996)

[Interview with Client/Questionear/… 5](#_Toc444295997)

[Functional Requirements 7](#_Toc444295998)

[Use Cases 7](#_Toc444295999)

[Use Case 1 7](#_Toc444296000)

[Use Case 2 7](#_Toc444296001)

[Use Case 3 8](#_Toc444296002)

[Use Case 4 8](#_Toc444296003)

[Use Case 5 9](#_Toc444296004)

[Use Case 6 10](#_Toc444296005)

[Use Case 7 10](#_Toc444296006)

[Use Case 8 11](#_Toc444296007)

[Use Case 9 11](#_Toc444296008)

[Non-Functional Requirements 13](#_Toc444296009)

[Cost Constraints 13](#_Toc444296010)

[Reliability 13](#_Toc444296011)

[Time Constraints 13](#_Toc444296012)

# Modification History

* 2/23/2016 Original product design and documentation was modified to keep track of each player’s history and display upon user’s request.
* Modified the PowerPoint presentation which has to be re-designed due to lack of understanding and clarifying the material.
* 02/26/2016 Edited for consistency and clarity

# Domain Knowledge

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| 6x6 Tic Tac Toe | A game that is played with two players, X and O, who take turns marking spaces usually in a 6x6 grid. |
| Graphical User Interface | Human-Computer Interface that uses windows icons and menus which can be manipulated by user input. |
| Platform | The hardware and support software which a program is intended to operate. |
| Client/Server System | The relationship between processes running on separate machines. A client initiates the dialog by sending a request to the server. |
| 4-in-a-row | When a player succeeds in placing four marks in a horizontal, vertical, or diagonal row. |
| MySQL | An application database that manages data and allows fast storage and retrieval of that data. |

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| 3T | Tic-Tac-Toe |
| GUI | Graphical User Interface |
| SPMP | Software Project Management Plan |
| SQA | Software Quality Assurance |
| 3T FC | Tic Tac Toe Flow Chart |
| TC | Test Cases |
| AI | Artificial Intelligence |

## Interview with Client

**Location:** S708OneMainStreet

**Date:** 2/8/2015

**Time:** 1:30 PM

**Attendees:** Yuan Shengli

Project Management Committee (Eddie, Carrie, Jason, Ibra)

**Description:**

*Question 1*: How is the application going to work?

*Response*: The application will be a computer 6x6x4 tic-tac-toe game. The game will be played on a 6x6 grid, two players (one may be a computer) will take turns to place a game piece of their choice. The winner is the player with the most 4 stones in a horizontal, vertical, or diagonal row. A user can either play as a guest or as a registered use. The computer will keep track of each player’s history. If requested, the computer will display the history of the selected players. Users may select the level of skill and who goes first.

*Question 2*: What steps did you take to develop it?

*Response:* I took the classic Tic-tac-toe model, so there was not much of a concept to develop, but I added the extra element of extra playing rows. I wanted to add a sense of excitement to the game.

*Question 3:* What are the parameters of the game?

*Response:* Due to the nature of the 6x6 game board, there are more spaces available to be played on, but the rules of the 4x4 game board are still in play. By having this relationship between the game board and rules, it allows for the player to strategically plan his/her moves in more ways.

*Question 4:* What problems do you believe you’ll encounter whilst creating the application?

*Response:* I believe that there may be an issue in the deployment of the application itself, and though I believe that it will not be an issue, there is still the possibility. Other than that, I do not know of any foreseeable issues.

# Functional Requirements

## Use Cases

### Use Case 1

|  |  |
| --- | --- |
| Goals of actor | To play the game during downtime, to be entertained and to win the game. |
| Tasks | The game user starts a game. The system should create a game for the player so he/she can choose an opponent or play against the PC also allowing the user to pick the difficulty level and start a new game. |
| Preconditions | Clean game |
| Exceptions | If a game is already in process, the application will show an error message. |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 2

|  |  |
| --- | --- |
| Goals of actor | Register the user name |
| Task | Register user name to log into the game and be able to play. |
| Preconditions | No registered user name |
| Exceptions | Display error message if user name is already taken. |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 3

|  |  |
| --- | --- |
| Goals of actor | Log in to the game |
| Task | Enter the user name to log into the game |
| Preconditions | User has to be registered in order to log in |
| Exceptions | Display error message if user name is invalid |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 4

|  |  |
| --- | --- |
| Goals of actor | Quit game |
| Task | Quits the game |
| Preconditions | A game has to be running and/or ended in order to cancel or quit the application |
| Exceptions | Can’t quit the application if there is no game in process |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 5

|  |  |
| --- | --- |
| Goals of actor | Load New Game |
| Task | Starts a new game after the previous game has ended |
| Preconditions | A game has to have ended or canceled in order to load a new game form |
| Exceptions | New game is loaded only after application is started or after a game has either ended or has been canceled |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 6

|  |  |
| --- | --- |
| Goals of actor | Single Player |
| Task | It will use the single player vs AI whilst playing the game |
| Preconditions | Can only be used in a system playing against the AI |
| Exceptions | Single player is only applicable against to AI |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 7

|  |  |
| --- | --- |
| Goals of actor | Difficulty Level |
| Task | Assigns a timeline in which a moves needs to be made in order to increase the difficulty of the game. |
| Preconditions | User has to set the difficulty level in order to play with either an advance, medium or easy setting. |
| Exceptions | Can be used in a multi-player setting or against the AI |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 8

|  |  |
| --- | --- |
| Goals of actor | Password Reset |
| Task | Resets the password |
| Preconditions | User forgot or wishes to change the password |
| Exceptions | User has to have a user name and be registered in order to be able to have a password and be able to reset it |
| Variation of action interactions |  |
| System change/production |  |

### Use Case 9

|  |  |
| --- | --- |
| Goals of actor | Score Display |
| Task | Display the number of winnings and loses of the players. |
| Preconditions | Game has to have ended in order for the screen to display scores |
| Exceptions | No score will be available unless a game has been played and a game has been decided. |
| Variation of action interactions |  |
| System change/production |  |

# Non-Functional Requirements

## Cost Constraints

No cost constraints are foreseeable in the near future. Due to this lack of constraint, we are sure no cost will be generated and also no downtime will be expected for this project.

We are using GitHub and Drobox to keep track of our documentation and source code. A log file is generated to show all of changes and commits on the master branch. The branch has been replicated as a mirror branch where developers can make changes and submit their code to be reviewed before being merged with the main branch.

## Reliability

The software should be usable, without any operational faults, for the period of time between its deployment and the following three years.

## Time Constraints

Most meetings took place either after CS 3420 class or in Google Hangouts, Skype calls, and email in order to accommodate all group members with a schedule where we could all meet without running into any time constraints whilst building the application and filling out the required documents for the project.