Department of Computer Science & Software Engineering

SOEN 6011 Summer 2016

Requirements Document / Use Case Analysis[1]

Assignment - 2

Professor: - Nicolangelo Piccirilli

Team Name: Triple – T
Group Number - 1
Date of submission – 13th May 2016

Team Leader: NIDHI ARORA
Team Members:

	Name	Student ID	Contribution
1.	Amanjot Kaur Ahluwalia	40011623	Requirements Analysis - Documentation
2.	Arash Arasteh	40000580	Coding Team 1- Documentation
3.	Arash Farkish	27678835	Coding Team 1- Documentation
4.	Basireddy Sandeep Kumar	40016071	Testing - Designing
5.	Dalvir Singh Bains	40012722	Requirements Analysis - Documentation
6.	Nidhi Arora	40014504	Coding Team 2 - Documentation
7.	Parinaz Barakhshan	27675518	Coding Team 1- Documentation
8.	Sarthak Batra	27408978	Coding Team 2 - Documentation

---

[1]

TABLE OF CONTENTS

Part A – Requirements Document

A. Problem

We are Supposed to develop three versions of java application of the Tic-Tac-Toe game upon different platforms like desktop and android. Initially we need to develop a simple GUI over desktop having very basic game functionalities. later , we need to develop a full fledged game on android platform, where two players can play game like real life scenario. Finally, the level of the game has to be enhanced with the addition of heuristic in the game where a player can play game with the computer itself.

The Software life cycle model which will be used in this project is iterative and different kind of tools like Eclipse IDE and Android Studio will be used to accomplish this project.

B. Background information

Deliverable 1 - The platform used will be desktop and at the starting the players will be asked for their name and their symbol preference. As soon as they enter this information, the 3X3 grid will be displayed which is initially empty. Starting the game play, the player 1 will mark his preference over the particular cell he wants too with the selected symbol and then the chance will be given to the other player while switching the symbol. This alternative game play goes on until all the cells are completely filled. Although no algorithm will be applied to check the winner or draw state.

Deliverable 2 - Now the platform used will be android and will be having extended functionalities of the previous deliverable. Here, as the players will be getting chances to mark their preferences in alternative fashion like before while at the backend as soon as 5 moves have been completed the system will be calling a separate algorithm which will seek for the winner(if any) and now this will be checked at every stage until all the cells in the grid are over and the match is draw.

The players win state will be determined if any of the symbol persists in 3 adjoining cells either vertically, horizontally or diagonally. Therefore, the player to whom that symbol belongs to will be the winner.

 Three possible outcomes of the game are-
   ● "Player 1 wins" (the game is over).
   ● "Player 2 wins" (the game is over).
   ● "Match Draw" (the game is over, and it ended in a draw)

Requirements Document / Use Case Analysis(Assignment 2)

Also, If a player exits the game before the end, the other player will be the winner.

Deliverable 3 -For this deliverable we have a choice of implementing the the application in either java or android platform. Here the whole application will be having an addition of a heuristic algorithm which means that the player can now play game with the computer and the system will keep on determining the winner as soon as 5 moves have been conducted.

The heuristics of the game need to be upto the mark so that the player shall be having a challenge while playing the game and the three possible outcomes will be-
Three possible outcomes of the game are-
- "Player 1 wins" (the game is over).
- "Computer wins" (the game is over).
- "Match Draw" (the game is over, and it ended in a draw)

Inputs -

Before game play - player's name and symbol preference ('X' or 'O')

During Game play - Select desired grid and place symbol upon click

Outputs -

Display the final status of the game such as winner (player 1, player 2 or computer) or draw match.

Goal During game play -

Determining the status of the game (win or draw condition).

Requirements for the whole project -

	DELIVERABLE 1		DELIVERABLE 2		DELIVERABLE 3		
Groups							
Lists	FUNCTIONAL REQUIREMENTS	NON FUNCTIONAL REQUIREMENTS	FUNCTIONAL REQUIREMENTS	NON FUNCTIONAL REQUIREMENTS	FUNCTIONAL REQUIREMENTS	NON FUNCTIONAL REQUIREMENTS	
Layers							
LAYER 1	The user shall be able to view a Grid of 3*3 as Game board where if clicked on any empty cell, it should display	Platform specif c: Windows xp + is needed as the operating system. And also JDK should be installed on the system	Winning condition should be checked.The f rst player who places three marks in a horizontal, diagonal or	Platform specif c: Windows xp + is needed as the operating system. And also JDK should be installed on the system	Query player for choosing his competitor to be player or Computer(Game Mode).	Platform specif c : This Deliverable will be implemented using Java platform. Therefore JRE is needed for executing	
	Players can enter their names and choose their symbol preferences	Performance: There is no considerable delay(0.3 Seconds is acceptable for this Iteration) between action performed by	If there are no empty cells and neither player has a winning combination, the game enters draw state.	Performance: There is no considerable delay(0.3 Seconds is acceptable for this Iteration) between	Query players for choosing Game Dif culty Level (Easy, Normal, Hard) which determines the level of	Usability and Learnability: all the information regarding game rules and features will be	
	After the f rst move, the other player places his symbol in any empty cell and so on.Reset button is needed to clean the board like	Documentation of Code is provided so that improving the application features for later deliveries become easier.	To explain the rules of the game or introduce new features to the player. Help option should be provided.	Documentation of Code is provided so that improving the application features for later deliveries become	Providing Administrator Access for viewing the high scores and logs during initial phase of the game.	Performance: Delay of loading the program is less than 0.3 Seconds(acceptable for this application)-	
	Reset button is needed to clean the board like initial state.	Recoverability:in the event of an interruption or a failure, system recovers by resetting the board and the state of the system	3 options for Game sets (set of 1 game , set of 3 games and set of 5 games) should be provided to the user.	Recoverability:in the event of an interruption or a failure, system recovers by resetting the board and the state	Considering 3 options for Game sessions: set of 1 game . set of 3 games and set of 5 games(for both manual	Operationality : when 2 players play dif erent Game Dif culty levels are implemented through dif erent	
	Exit Button is needed if the player wants to exit the game.	Extensibility:the implementation takes future growth into consideration	Calculate the total number of mission accomplishments, draws in set chosen	Extensibility : the implementation takes future growth into consideration.	Saving high scores of all the players based upon the dif culty level they selected.	Documentation: is provided to have easier maintenance of the application .	
			Providing Administrator Access for viewing the high scores and logs		Providing help page for the player to make judgement upon the move he should make to win the game.		
					Calculate the total number of matches won or draws state entered in a session and give the f nal result		
					Playing dif erent background music (within the code based) on the type of game level (easy, medium or		
					As the player wins, Provide him with a gift and save data in the log f le.		

Project Deliverable 01

Narrative

As the game is started, both the players shall be asked for their names and their symbol preference. As soon as this information is entered, the Game board of 3*3 grid will be displayed and then one of the players will be asked to mark his preference over any of the empty cell. As this player marks his choice, the chance to play will be automatically transferred to the other player with the switched symbol and even this player shall be able to make his selection over empty cell only.

This deliverable will only be showing up the basic GUI of the application in desktop platform and the players will be able to mark their selections alternatively over the empty cells only. The winning condition or draw state will not be checked in the first deliverable. The GUI will also be having a Reset button that cleans the grids like the initial state and exit button to finish the game play.

Functional Requirements

REQ#	Requirement Description
REQ1	The user shall be able to view a Grid of 3*3 as Game board where if clicked on any empty cell, it should display the symbol selected by the player respectively.
REQ2	Players should enter their names and choose their symbol preferences
REQ3	After the first move, the other player places his symbol in any empty cell and so on.
REQ4	Reset button is needed to clean the board like initial state.
REQ5	Exit Button is needed if the player wants to exit the game.

- ■ - High Priority
- □ - Medium Priority
- ▢ - Low Priority

Non- Functional Requirements

- **Platform specific:** Windows xp + is needed as the operating system. And also JAVA should be installed on the system so that the application can run.
- **Performance:** There is no considerable delay(0.3 Seconds is acceptable for this Iteration) between action performed by the user and response from system to display the selection.
- **Documentation of Code:** Will be provided so that improving the application features for later deliveries become easier.

- **Recoverability:** In the event of an interruption or a failure, system recovers by resetting the board and the state of the system
- **Extensibility:** The implementation takes future growth and enhancements into consideration.

Project Deliverable 02

Narrative

This Deliverable focuses over android platform as here the application will be working on mobile devices and the will be having similar functionality as of the previous version like 3X3 grid and alternative player chance but will be included with a new algorithm which determines the winner of the game or if the game enters a draw state.

This deliverable will be an extension of deliverable 1 but over android platform. Also, the game will be having an option for the players to select if they want 3 or 5 game set, also the player with maximum number of wins in a set will be the winner as a whole. There will be an additional option to login as administrator to view the highscores of all the players. The Gui will be having options like reset or exit the game.

Functional Requirements

REQ#	Requirement Description
REQ1	Winning condition should be checked.The first player who places three marks in a horizontal, diagonal or vertical line is declared as the winner of the game.
REQ2	If there are no empty cells and neither player has a winning combination, the game enters draw state.
REQ3	To explain the rules of the game or introduce new features to the player. **Help** option should be provided.
REQ4	Calculate the total number of **accomplishments, draws** in set chosen
REQ5	Providing **Administrator Access** for viewing the high scores and logs

- High Priority
- Medium Priority
- Low Priority

Non- Functional Requirements

- **Platform specific:** Android 4.0 and up is needed to execute the program. Developing the program will be done through Android Studio.
- **Usability and Learnability:** Explanation of all the game rules and additional features will be provided using the Help page accessible through help button.
- **Performance:** There is no considerable delay (0.3 Seconds is acceptable for this Iteration) between action performed by the user and response from system to display the selection.
- **Documentation:** All necessary documents will be provided to make improvement in functionality and features at later stage..
- **Scalability:** Scalable UI is needed that works seamlessly across a broad range of Android devices.
- **Portability:** Application can be transferred from one hardware, software or other operational environment to another Android System.
- **Quality Measurement:** Memory Allocated to the application and also processing speed of the application is measured.

Project Deliverable 03

Narrative

The deliverable could either be developed over desktop or android platform. And this deliverable is believed to be an extension of the previous version. As in the previous version, we have two distinct players who make their choices in order to achieve the goal of winning the match. But here, the game will be having a heuristic algorithm by the help of which the player can play game with the automated system. Therefore the need for the second player to play the game will be eliminated and with the addition of the heuristic algorithm, the player will be able to select the level of game(easy, medium or hard) in the initial phase of the game. At the back end, another algorithm will be seeking for the winner of the game or draw state of the game. Hence, there will be three outcomes of the games- player is winner, computer is winner or the match is draw. To enhance the difficulty level of the game, a new concept of time frame constraint will be added in which the player and the computer needs to mark their selection within a particular time frame or else they lose the game, this time frame constraint will be based upon level of the game selected by the player in initial phase.

Even in this deliverable, there will be features like administrator to manage the highscore and logs.

Functional Requirements

REQ#	Requirement Description
REQ1	Query player for choosing his competitor to be player or Computer(**Game Mode**).
REQ2	Query players for choosing Game Difficulty Level (Easy, Normal, Hard) which determines the level of heuristic in the game and the time constraint for the player.
REQ3	Providing Administrator Access for viewing the high scores and logs during initial phase of the game.
REQ4	Considering 3 options for Game sessions: set of 1 game . set of 3 games and set of 5 games(for both manual and automated play) and count the number of wins for each case.
REQ5	Saving high scores of all the players based upon the difficulty level they selected.
REQ6	Providing help page for the player to make judgement upon the move he should make to win the game.
REQ7	Calculate the total number of matches won or draws state entered in a session and give the final result later.
REQ 8	Playing different background music (within the code based) on the type of game level (easy, medium or hard)
REQ9	As the player wins, Provide him with a gift and save data in the log file.

- ■ - High Priority
- ☐ - Medium Priority
- ☐ - Low Priority

Non- Functional Requirements

- **Platform specific** : This Deliverable will be implemented using Java platform. Therefore JRE is needed for executing the application and JDK and Eclipse is used for developing the application in case of desktop version and android studio will be needed to build in android platform.
- **Usability and Learnability**: all the information regarding game rules and features will be provided through the help page accessible through help button. .

Requirements Document / Use Case Analysis(Assignment 2)

- **Performance**: Delay of loading the program is less than 0.3 Seconds (acceptable for this application)-reaction to user click and there is no considerable delay (0.1 Seconds is acceptable for this Iteration) between action performed by the user and response from system.
- **Operationality** : in case 2 users are playing the game, then the operationality will be determined by the time constraint then have to input their choice. While, on the other hand in case of automated play, the heuristic level (easy, medium or hard) will be determined to enhance game play.
- **Documentation:** all information related to coding and implementation will be provided to enable easy maintenance and modification of the application .
-

Part B – Use Case Analysis

Project Deliverable 01

UC#	Use Case Name
UC#1	Start Game
UC#2	Enter Player Name
UC#3	Choose Symbol
UC#4	Player Move
UC#5	Reset Game
UC#6	Exit Game

Use Case Diagram:
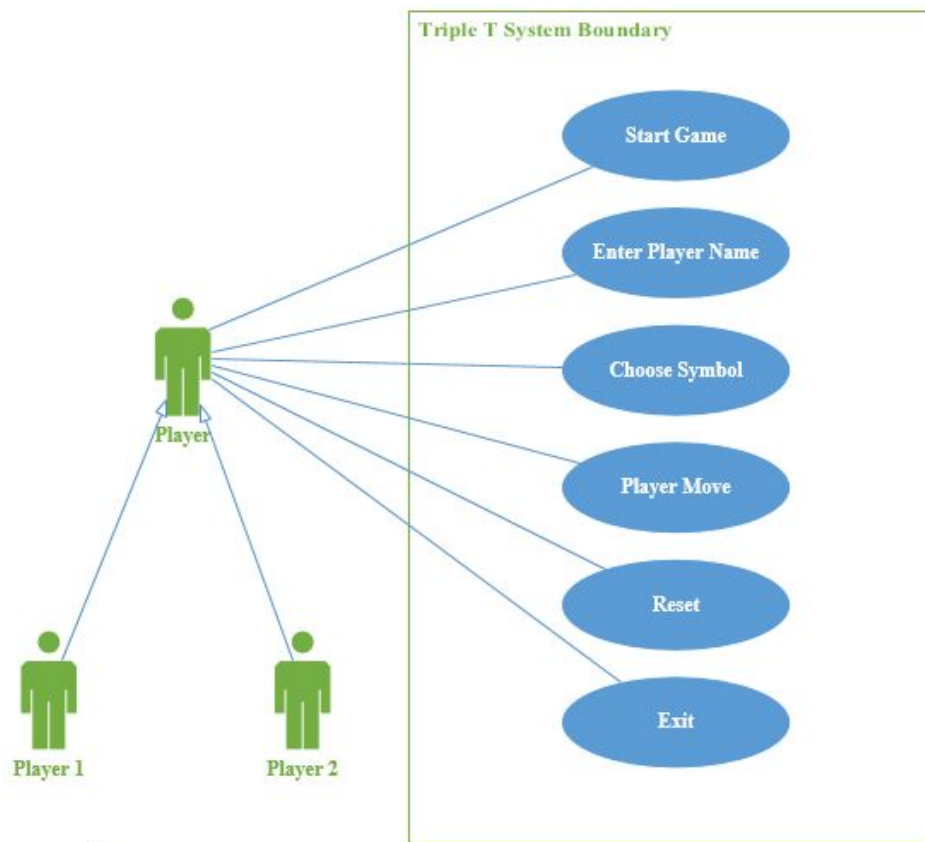
Use Case Scenarios:

Use Case 1: Start Game

Number	1
Name	Start Game
Summary	User starts the game set
Priority	High priority
Preconditions	Software is already installed on the system. JRE is also installed
Postconditions	User enters his name.

Primary Actors	Player 1, Player 2	
Secondary Actors	-	
Trigger	Player runs the application	
Main Scenario	Step	Action
	1	User runs the application
	2	System displays name entry page
Extensions	Step	Branching Action
	2a	System does not load the application
Requirements	REQ #	Description
	REQ1	The user shall be able to view a Grid of 3*3 as Game board where if clicked on any empty cell, it should display the symbol selected by the player respectively.

Use Case 2:Enter Player Name

Number	2	
Name	Enter Player Name	
Summary	User will enter his/her name	
Priority	High priority	
Preconditions	Player has already entered the first interface in which he is being an option of writing his name.	
Postconditions	System asks to choose the symbol.	
Primary Actors	Player 1, Player 2	
Secondary Actors	-	
Trigger	Player enters the first interface of the game.	
Main Scenario	Step	Action

	1	System requests the name
	2	Player enters the name
Extensions	**Step**	**Branching Action**
	2a	If both the players have the same usernames, message pops up saying"this username already exists ,please choose another username"
Requirements	**REQ #**	Description
	REQ2	Players should enter their names and choose their symbol preferences

Use Case 3: Choose Symbol

Number	3	
Name	Choose Symbol	
Summary	Player1 chooses 'X' or 'O'	
Priority	High priority	
Preconditions	Name is entered	
Postconditions	System shows board, Player1 enters the symbol	
Primary Actors	Player	
Secondary Actors	-	
Trigger	Player enters the name	
Main Scenario	**Step**	**Action**
	1	System shows symbol choice option
	2	Player chooses the symbol

Requirements	REQ #	Description
	REQ2	Players should enter their names and choose their symbol preferences

Use Case 4: Player Move

Number	4	
Name	Player Move	
Summary	Players put 'X' or 'O' on the grid	
Priority	High priority	
Preconditions	The user adds his name and symbol and clicks on "play option" to get a view of grid.	
Postconditions	Symbols are on the grids. Cells containing the symbols are deactivated.	
Primary Actors	Player1, Player 2	
Secondary Actors		
Trigger	Player can visualise an empty grid.	
Main Scenario	Step	Action
	1	System views the empty board
	2	Player can click on any empty cell of the board
	3	System shows the symbol related to the player on the selected cell and deactivates that cell
Requirements	REQ #	Description
	REQ3	After the first move, the other player places his symbol in any empty cell and so on.

Use Case 5: Reset Game

Number	5
Name	Reset Game
Summary	Grid cells will be unmarked and Game status is set to null
Priority	Medium priority
Preconditions	At Least one cell of the grid must have been marked
Postconditions	The entire grid gets blank and game status is set to zero
Primary Actors	Player1, Player 2
Secondary Actors	-
Trigger	Player clicks on reset button

Main Scenario	**Step**	**Action**
	1	Player clicks on reset button
	2	System clears the entire grid and will set the Game status to Null
Requirements	**REQ #**	**Description**
	REQ4	Reset button is needed to clean the board like initial state.

Use Case 6: **Exit Game**

Number	6
Name	Exit Game
Summary	Player decides to end game and close the game console
Priority	High priority
Preconditions	Game board is viewed
Postconditions	The game console is closed
Primary Actors	Player1, Player 2
Secondary Actors	-
Trigger	Player clicks the exit button

Main Scenario	Step	Action
	1	Player clicks the exit button
	2	System logs the status of the game and closes the game console
Requirements	**REQ #**	**Description**
	REQ5	Exit Button is needed if the player wants to exit the game.

Project Deliverable 02

UC#	Use Case Name
UC#1	Start Game
UC #2	Enter Player Name
UC #3	Choose Player Symbol
UC#4	Win History
UC#5	Player Move
UC#6	View Game Status
UC#7	Process Game Status
UC#8	Reset Game
UC#9	Exit Game

Use Case Diagram:
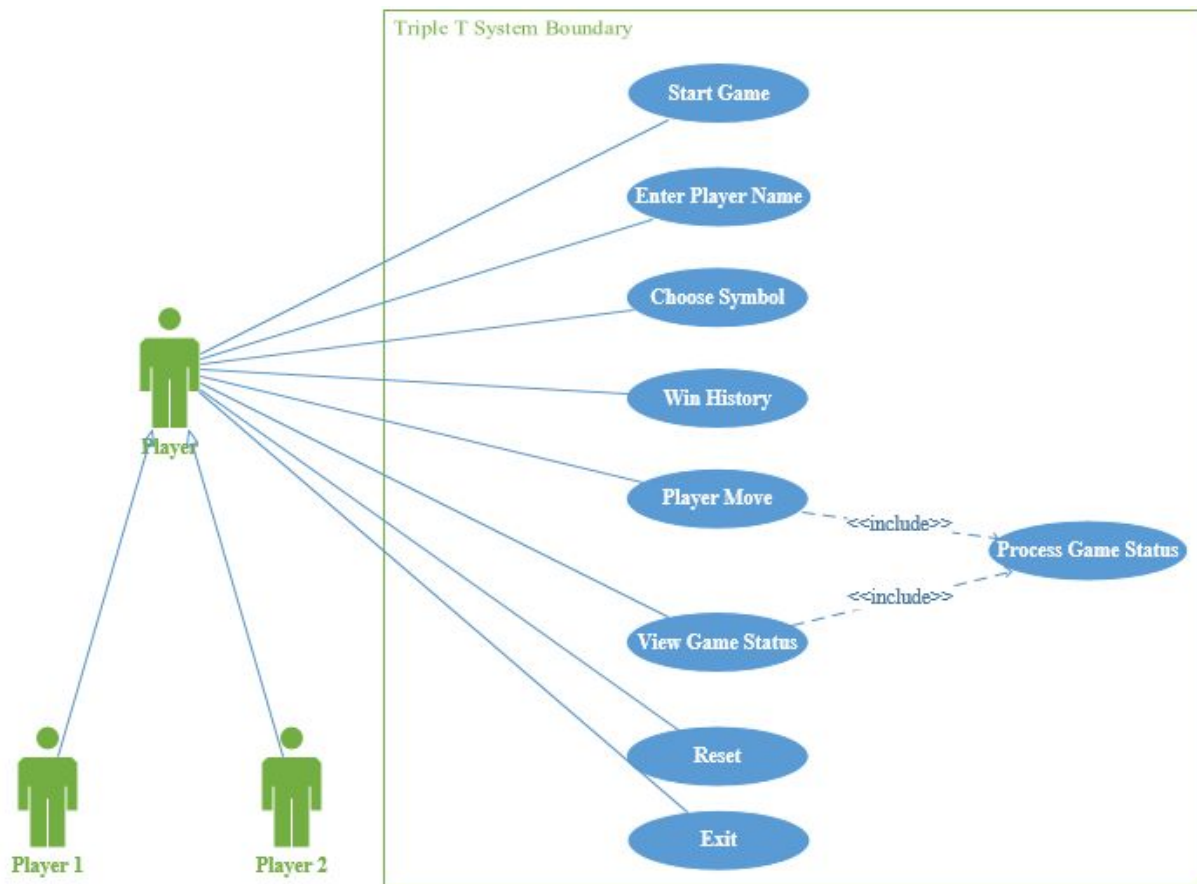
Use Case Scenarios:

Use Case 4: Win History

Number	4
Name	Win History
Summary	The user checks his/her win History in the game compared to the top player.
Priority	Medium priority

Preconditions	Game is finished	
Postconditions	System waits for new user commands.	
Primary Actors	Player1, Player2	
Secondary Actors	-	
Trigger	The player finishes his/her game.	
Main Scenario	**Step**	**Action**
	1	The system calculates the score and compares with the score of the top player and outputs it to the user.
Extensions	**Step**	**Branching Action**
	1.a	System does not have any top score to compare with so the player score will be set as the top score
Requirements	**REQ #**	**Description**
	REQ1	Winning condition should be checked.The first player who places three marks in a horizontal, diagonal or vertical line is declared as the winner of the game.

Use Case 6: View Game Status

Number	6
Name	View game status
Summary	The player wants to know the previous game record like the number of wins and total number of points earned.
Priority	Low Priority
Preconditions	The game has ended.
Postconditions	The game status page is on the display and further marking on the grid is disabled
Primary Actors	Player 1, Player 2
Secondary Actors	-

Trigger	The player selects the game status option	
Main Scenario	**Step**	**Action**
	1	The player chooses to check the game status
	2	Game statistics are displayed to the user
Requirements	**REQ #**	**Description**
	REQ1	Winning condition should be checked.The first player who places three marks in a horizontal, diagonal or vertical line is declared as the winner of the game.

Use Case 7: **Process Game Status**

Number	3	
Name	Process Game Status	
Summary	The player wants to see his past winning record which is number of wins in the game	
Priority	Medium priority	
Preconditions	The player has at least won a single game	
Postconditions	The system displays the player's past winning record	
Primary Actors	Player 1, Player 2	
Secondary Actors	-	
Trigger	The player wins a game and want to retrieve about it	
Main Scenario	**Step**	**Action**
	1	The player selects to preview the winning history
	2	The system displays his number of wins
	3	The player continues the game
Requirements	**REQ #**	**Description**

	REQ1	Winning condition should be checked.The first player who places three marks in a horizontal, diagonal or vertical line is declared as the winner of the game.

Project Deliverable 03
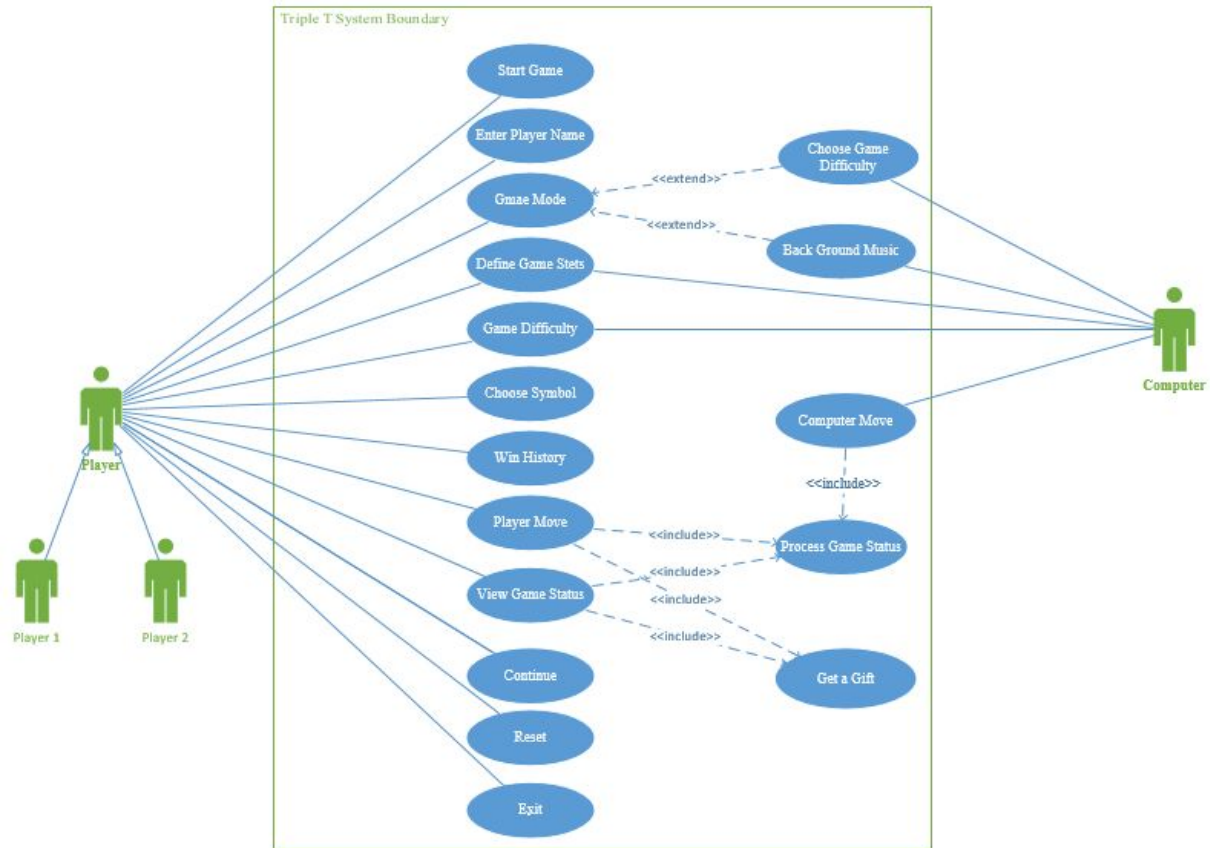
UC#	Use Case Name
UC#1	Start Game
UC #2	Enter Player Name
UC #3	Game Mode
UC#4	Define Game Sets
UC#5	Game Difficulty
UC#6	Choose Symbol
UC#7	Win History
UC#8	Player Move
UC#9	View Game Status
UC#10	Choose Game Difficulty
UC#11	Background Music
UC#12	Computer Move
UC#13	Process Game Status
UC#14	Get Gift
UC#15	Reset Game
UC#16	Exit Game
UC#17	Continue Game Set

Use Case Diagram:

Use Case Scenarios:

Use Case 3: Game mode

Number	3
Name	Game Mode
Summary	Player 1 has an option to select whether to play with human player or with computer
Priority	High priority
Preconditions	The game has already been started

Postconditions	Difficulty selection page is on the screen	
Primary Actors	Player 1, Player 2	
Secondary Actors	Computer <<system>>	
Trigger	The system prompts the user to select the game mode	
Main Scenario	**Step**	**Action**
	1	System displays the available options of game mode to user
	2	User selects his choice
	3	System allocates either player 2 or computer to the game
Requirements	**REQ #**	**Description**
	REQ1	Query player for choosing his competitor to be player or Computer(**Game Mode**).

Use Case 4: Define Game set

Number	4	
Name	Define Game set	
Summary	The	
Priority	High Priority	
Preconditions	Player has chosen Game Mode	
Postconditions	The game started	
Primary Actors	Player 1, Player 2	
Secondary Actors	-	
Trigger	Player wants to have a best of 3 games in a particular set	
Main Scenario	**Step**	**Action**
	1	The player selects the number of sets in the game

	2	System starts the game
Extensions	**Step**	**Branching Action**
	1a	Player clicks on play button and a message pops up to remind the customer to choose the number of sets.
Requirements	**REQ #**	**Description**
	REQ4	Considering 3 options for Game sessions: set of 1 game . set of 3 games and set of 5 games(for both manual and automated play) and count the number of wins for each case.

Use Case 5: Game Difficulty

Number	5	
Name	Choose Game Difficulty	
Summary	The user is able to choose the game difficulty level. The computer checks the level selected by user and play its moves accordingly.	
Priority	High Priority	
Preconditions	Player has chosen the set he wants to play.	
Postconditions	The computer takes turns after having determined difficulty level	
Primary Actors	Player	
Secondary Actors	Computer	
Trigger	The user chooses the game set.	
Main Scenario	**Step**	**Action**
	1	The user chooses the difficulty level.
	2	The computer keeps track of the chosen level.
	3	The game set starts between the players
Extensions	**Step**	**Branching Action**

Requirements Document / Use Case Analysis(Assignment 2)

	1a	Player clicks on play button and a message pops up to remind the customer to choose the difficulty level.
Requirements	**REQ #**	**Description**
	REQ2	Query players for choosing Game Difficulty Level (Easy, Normal, Hard) which determines the level of heuristic in the game and the time constraint for the player.

Use Case 11: Background Music

Number	11
Name	Background Music
Summary	When the player and the computer are opponents in the game then the music plays in the background
Priority	Low priority
Preconditions	In game mode,player 1 chooses to play with computer
Postconditions	The game in progress with music in the background
Primary Actors	Computer<<system>>
Secondary Actors	-
Trigger	The computer enters the game as an opponent

Main Scenario	**Step**	**Action**
	1	The player chooses the mode ,set, and starts the game.
	2	The system plays the music according to the chosen mode(either computer or with human player)
Requirements	**REQ #**	**Description**
	REQ8	Playing different background music (within the code based) on the type of game level (easy, medium or hard).

Use Case 12: Computer Move

Number	12
Name	Computer Move
Summary	Computer acts as a player and makes moves based on a well defined algorithm keeping in mind the chosen difficulty level
Priority	High priority
Preconditions	The player 1 has made a move
Postconditions	The player 1 again gets turn to make a move
Primary Actors	Computer(as player)
Secondary Actors	Computer <<system>>
Trigger	The human player has made his move

Main Scenario	Step	Action
	1	The computer takes its turn to make a move on the grid
	2	System processes the game status and fills a grid
Requirements	REQ #	Description
	REQ1,REQ2	Query player for choosing his competitor to be player or Computer(**Game Mode**). Query players for choosing Game Difficulty Level (Easy, Normal, Hard) which determines the level of heuristic in the game and the time constraint for the player.

Use Case 14: Get a gift

Number	14

Name	Get a gift
Summary	Whenever the user wins a set,the system displays a message to the user with the picture of a gift
Priority	Low Priority
Preconditions	Player has won a particular set of the game
Postconditions	The user continues or quit the game set
Primary Actors	Player-Computer
Secondary Actors	-
Trigger	The game set completes successfully.

Main Scenario	Step	Action
	1	The player wins a game set
	2	The system displays a message indicating who won.
	3	The window showing the image of the gift pops out on the screen
	4	The player restarts the game or chooses to exit.
Requirements	REQ #	Description
	REQ9	As the player wins, Provide him with a gift and save data in the log file.

Use Case 17: **Continue Game Set**

Number	10
Name	Continue game set
Summary	When one set of a game ends, user can choose to play another set of the same game or end the game.

Requirements Document / Use Case Analysis(Assignment 2)

Priority	Medium priority	
Preconditions	One set of a game ends.	
Postconditions	Player is playing new set-System exit the game	
Primary Actors	Player	
Secondary Actors	-	
Trigger	Fishing a set	
Main Scenario	**Step**	**Action**
	1	System displays "continue this game"
	2	Player select "yes"
	3	System starts new set
Extensions	**Step**	**Branching Action**
	2a	Player selects "No"
	3a	System exit the game
Requirements	**REQ #**	**Description**
	REQ4	Considering 3 options for Game sessions: set of 1 game . set of 3 games and set of 5 games(for both manual and automated play) and count the number of wins for each case.