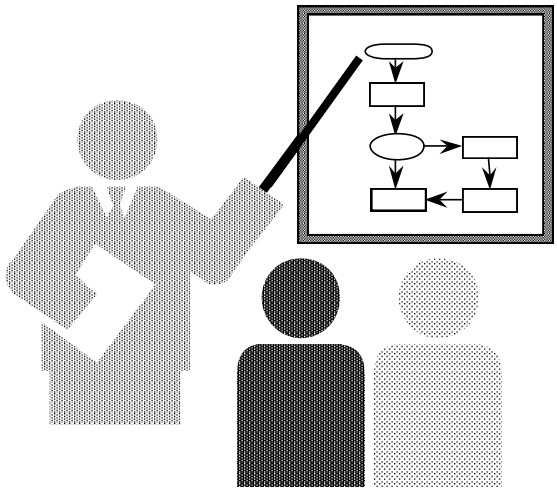
TTT Project
Requirements Document / Use Case Analysis



MAY 13, 2016

Team Members:

Name	ID	Email Address
Shen Li	27913907	shenli2015@yahoo.com
Isha Mehta	27376030	ishakmehta04@gmail.com
Zhihao li	27252331	Lizhihao1109@gmail.com
Akjil Kumar	27711794	reddyakhil24@gmail.com
Chenyang Li	27588801	tomli8chenyangli@gmail.com
Xiangshiyu Li	27320841	lixiangshiyu@gmail.com
Meng Yao	26847005	yaomenghncn@gmail.com
Singh Mehare	40012584	birdevinder1705@gmail.com
Golnoush Lotfi	26753906	golnoush.lotfi@gmail.com

Table of Content

1. Requirements Document 3
   1.1 Problem 3
   1.2 Background information 3
2. Deliverable 1 4
   2.1 Functional Requirements 4
   2.2 Non-functional requirements 4
   2.3 Use Case Diagram 5
   2.4 Use Case Scenario 6
3. Deliverable 2 7
   3.1 Functional Requirements 7
   3.2 Non-functional requirements 8
   3.3 Use Case Diagram 10
   3.4 Use Case Scenario 10
4. Deliverable 3 12
   4.1 Functional Requirements 12
   4.2 Non-functional requirements 13
   4.3 Use Case Diagram 15
   4.4 Use Case Scenario 15

1. Requirements Document

1.1 Problem

Tic-Tac-toe is a simple strategy game, where the aim is to place 3 'X' or 3 'O' vertically, horizontally or diagonally in a row. Once a player enters an 'X' or 'O', other player can place his/her mark on the empty tile and first person to get consecutive 'X' or 'O' in a row wins. The game is developed in Java platform using SWING framework and Android for mobile version of the game. The game should allow two players or a player and a computer player to play the game. The game offers different difficulty levels (beginners, intermediate and advanced) to choose from. The game algorithm should be such that when the advanced level is chosen, it becomes very difficult for the human player to beat the computer player.

The game has three version where first version just includes a board that allows user to click 'X' or 'O' on the board. The second version is the complete game with a human player and is the mobile application. And the third version is the desktop version f the second. The game board includes help, reset and exit button to allow user to perform these functions. The game also stores the scores of players which can be viewed on score board. To implement this we will be using the array list to store the scores obtained by each player. The reason behind using array list is to enable flexible portability of the code for desktop version to mobile version.

1.2 Background information

Tic -Tac-Toe is an ancient strategy game whose rules are quite simple. A two player game, that allows players to place an 'X' or 'O' on the board to play the game. The goal of the game is to get 3 consecutive 'Xs' or 'Os' in a row. The first one to achieve this wins the game.

Despite of its simplicity, Tic-Tac-Toe game is tricky due to number of possible moves and number of possible positions. The player has to make the best move each time to increase its chance of winning. If none of the player succeed in getting 'X' or 'O' in a row the game is a tie.

2. Deliverable 1

2.1 Functional Requirements

ID	Feature	Priority(Critical, Important, Useful)
F1	Players can start a game by clicking the game icon.	Critical
F2	System asks player to choose 'X' or 'O' to display on the game board.	Critical
F3	Players can click the board to place ‘X’ or ‘O’ on the board.	Critical
F4	Players can click the help button for instructions.	Important
F5	System can display the game rules.	Important
F6	Player can exit the game.	Critical
F7	System can show pop-up to confirm exit.	Important
F8	Player can click the reset button to reset the game.	Critical

2.2 Non-functional requirements

Platform requirements

- IDE: Eclipse with ADT v23.2.1(Android Development Kit)
- Windows operating system

Process requirements
Release of PC version by Friday May 20 at 6 pm

Response time Process requirements
The maximum response time between click and reaction must be 16ms

Memory usage
The amount of memory occupied by the application should be between 128 to 512 MB.
Observations done from the performance log during testing.

Reliability
The probability of failure-free software operation is 99%

Availability
The Systems must achieve 99.999% availability

Allowances for maintainability and enhancement

- Good Code Quality
- Less Source Code Defects
- Less Technical Complexity
- Excellent Documentation
- Less Dead Code

Allowances for reusability

- Modules have low module complexity.
- Modules have good documentation (a high number of non-blank comment lines).
- Modules have few external dependencies (low fan-in and fan-out, few input and output parameters).
- Modules have proven reliability (thorough testing and low error rates).

2.3 Use Case Diagram

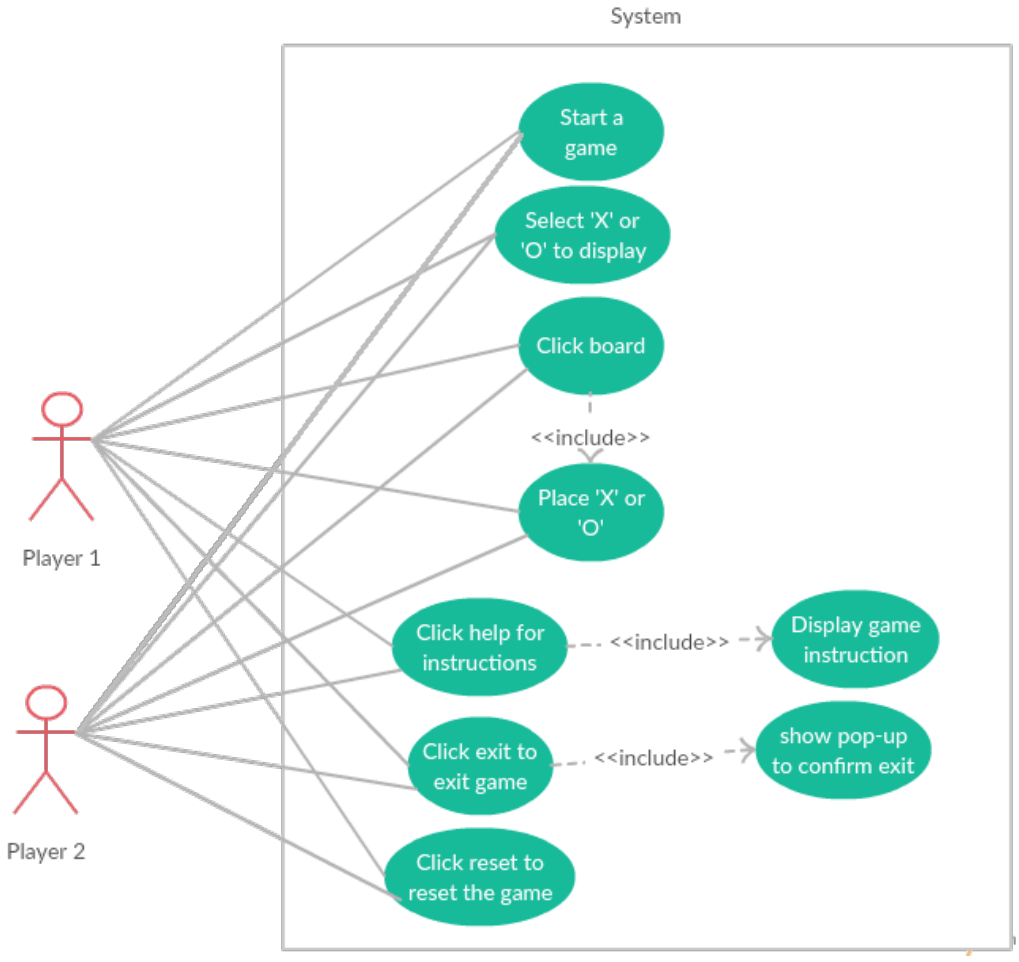


Figure 1. Use case for Deliverable 1

2.4 Use Case Scenario

Start a Game

Identifier	Explanation
Use Case ID	UID-001
Use case name	Start a Game
Description	The players are allowed to click the icon to start the game.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	N/A, Due to Start a Game is the very beginning step, there is no pre-condition.
Basic Flow	1. Player 1 and Player 2 can click the game icon to start the game
Post-condition	The screen-board is displayed and the player 1 and Player 2 are allowed to select 'X' or "O" to display.
Alternate-flow	N/A

Select 'X' or "O" to display

Identifier	Explanation
Use Case ID	UID-002
Use case name	Select 'X' or "O" to display
Description	The players are allowed to select 'X' or "O" to display which will be stored and displayed on the score-board.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	Any player must have started the game.
Basic Flow	1. System asks player to select 'X' or "O" to display. 2. Player 1 select 'X' or 'O'. 3. Player 2 select 'O' or 'X'.

Post-condition	Both of the players will select their own 'X' or 'O' and system will store both of their selection and display.
Alternate-flow	1 a. If player is unable to start the game, the system should inform player about the faulty condition. 3a and 4a. If players cannot select their own 'X' or 'O', system should by default show player 1 and player 2 instead of own 'X' or 'O'.

Click help for instructions

Identifier	Explanation
Use Case ID	UID-003
Use case name	Click help for instructions
Description	The players are allowed to click help button for instructions.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	Any player must have started the game and the help button is displayed on the screen-board
Basic Flow	1. Either the player 1 or player 2 to show the game instruction.
Post-condition	The game instruction is displayed n the screen.
Alternate-flow	1 a. If player is unable to click the help button, the system should inform player about the faulty condition.

3. Deliverable 2

3.1 Functional Requirements

ID	Feature	Priority(Critical, Important, Useful)
F1	Players can start a game	Critical
F2	Players can enter their name.	Critical
F3	The system can display players' name.	Critical
F4	The players can choose a level of game play (Beginners, Intermediate or Advance).	Useful

F5	System asks player to choose 'X' or 'O' to display on the game board.	Critical
F6	System can displays whose turn it is by showing player's name on the board.	Useful
F7	The players can click on the board to place ‘X’ or ‘O’.	Critical
F8	The system marks tile with an 'X' or 'O' once the player clicks on it.	Critical
F9	System will check after each click if any player is winning.	Critical
F10	The system freezes the board if any player wins.	Critical
F11	A player can detect its opponent's move to beat him/her.	Critical
F12	The players can view scores by clicking on the score-board.	Useful
F13	System can display the number of times a player has won.	Useful

3.2 Non-functional requirements

Platform requirements

- Platform: Android OS
- IDE: Eclipse with ADT v23.2.1(Android Development Kit)
- Android XML
- Android 5.0.1(API 21)

Process requirements
Release of Release of Mobile version by Friday May 27 at 6 pm

Response time
The maximum response time between click and reaction must be 10ms

Memory usage
The amount of memory occupied by the application should be between 128 to 512 MB. Observations done from the performance log during testing.

Reliability
The probability of failure-free software operation is 99%

Availability
The Systems must achieve 99.999% availability

Allowances for maintainability and enhancement

- Good Code Quality
- Less Source Code Defects
- Less Technical Complexity
- Excellent Documentation
- Less Dead Code

Allowances for reusability

- Modules have low module complexity.
- Modules have good documentation (a high number of non-blank comment lines).
- Modules have few external dependencies (low fan-in and fan-out, few input and output parameters).
- Modules have proven reliability (thorough testing and low error rates).

3.3 Use Case Diagram

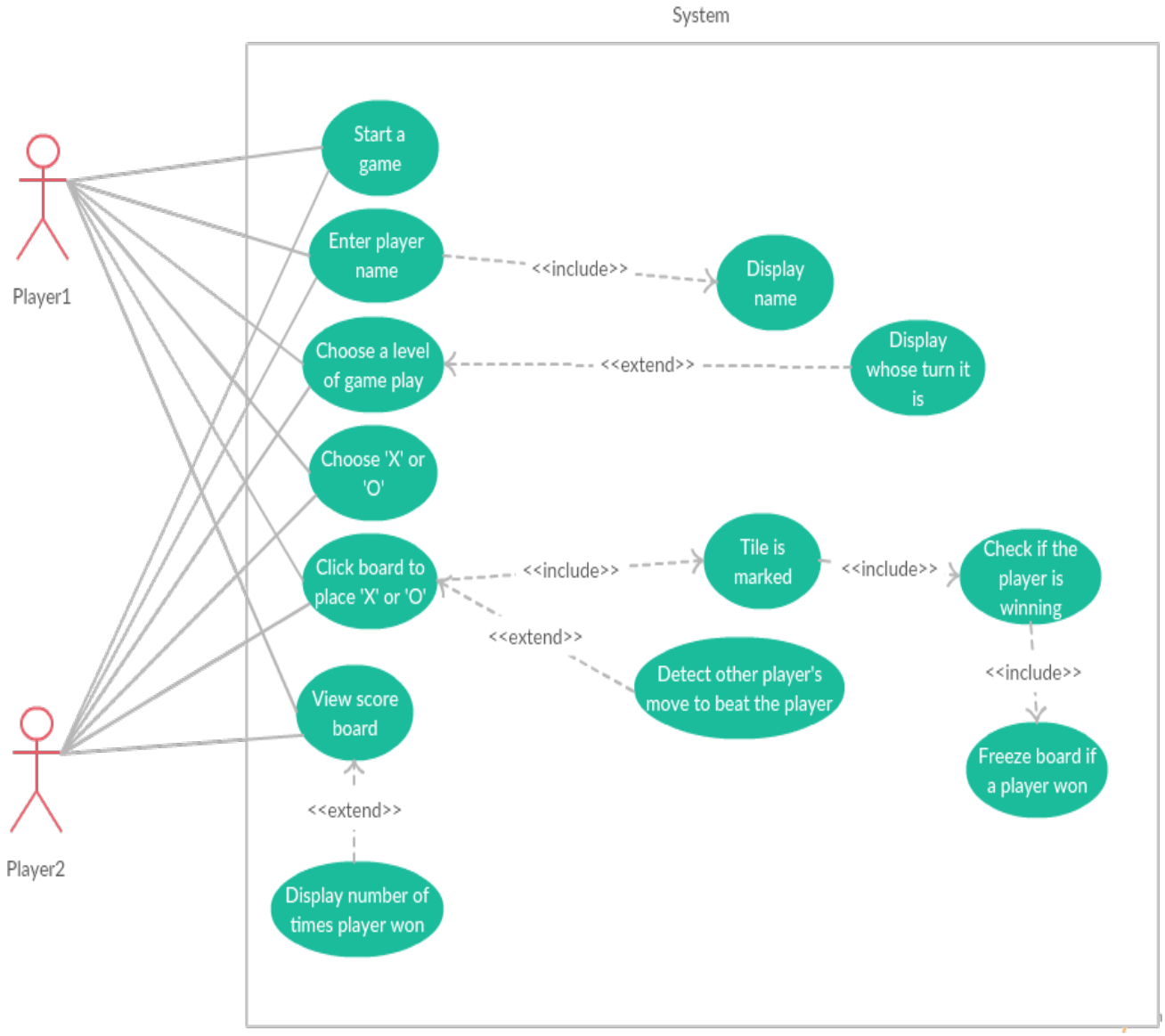


Figure 2. Use case for Deliverable 2

3.4 Use Case Scenario

Enter Player Name

Identifier	Explanation
Use Case ID	UID-001
Use case name	Enter player name

Description	The players are allowed to specify their names which will be stored and displayed on the score-board.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	Any player must have started the game.
Basic Flow	1. Any player starts a game 2. System asks player to enter their name. 3. Player 1 enters the name. 4. Player 2 enters the name.
Post-condition	Both the players name will be stored and displayed.
Alternate-flow	1 a. If player is unable to start the game, the system should inform player about the faulty condition. 3a and 4a. If players cannot enter their name, system should by default show player 1 and player 2 instead of their names.

View Score-board

Identifier	Explanation
Use Case ID	UID-002
Use case name	View score-board
Description	The players are allowed to view the score-board to see score of any players.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	Any player must have started the game.
Basic Flow	1. Player starts the game. 2. Player clicks the view score-board button.
Post-condition	System displays the scores of all the players.
Alternate-flow	1 a. If player is unable to start the game, the system should inform player about the faulty condition. 2a. If system cannot display the scores of the players, the system must provide an appropriate message to the user.

Choose the level of game play

Identifier	Explanation
Use Case ID	UID-003
Use case name	Choose the level of game play
Description	The players are allowed to choose the level of game that we wants to play.
Primary Actor	Player 1 and Player 2
Priority	Critical
Pre-conditions	Any player must have started the game.
Basic Flow	1. Player starts the game. 2. Both players enter their names. 3. Player can select the level of game (beginners, intermediate and advanced)
Post-condition	Based on the level chosen, the game will open up.
Alternate-flow	3 a.If the player is unable to choose the level, or skips that step, the system will by default take beginner's level.

4. Deliverable 3

4.1 Functional Requirements

ID	Feature	Priority(Critical, Important, Useful)
F1	Players can start a game.	Critical
F2	System will start playing the background music.	Important
F3	Player can enter his/her name.	Critical
F4	The system can display the player's name	Critical
F5	System offers different game levels for the user to choose from.	Useful
F6	User can select his opponent (computer or human).	Critical
F7	User is able to choose between X or O to play with.	Critical
F8	If player selects computer as his opponent, then computer can join the game as a player.	Critical

F9	Player can click on the board to place an 'X' or 'O' and play the game	Critical
F10	The system will start the timer on the player's move	Important
F11	The system marks tile with an 'X' or 'O' once the player clicks on it.	Critical
F12	System will check after each click if any player is winning.	Critical
F13	The system freezes the board if any player wins.	Critical
F14	System will issue a gift to the player on winning.	Important
F15	The computer payer can place 'X' or 'O'.	Critical
F16	The computer player can detect player's move to make the best move.	Critical
F17	The player can click the help button for instructions.	Important
F18	The system can display the game instructions.	Important
F19	The player can click the exit button to exit the game.	Critical
F20	The system can show pop-up to confirm the exit.	Important

4.2 Non-functional requirements

Platform requirements

- IDE: Eclipse with ADT v23.2.1(Android Development Kit)
- Windows operating system

Process requirements
Release of Release of AI version by Friday June 3 at 6 pm

Response time
The maximum response time between click and reaction must be 6ms

Memory usage
The amount of memory occupied by the application should be between 128 to 512 MB. Observations done from the performance log during testing.

Reliability

The probability of failure-free software operation is 99%

Availability
The Systems must achieve 99.999% availability

Allowances for maintainability and enhancement

- Good Code Quality
- Less Source Code Defects
- Less Technical Complexity
- Excellent Documentation
- Less Dead Code

Allowances for reusability

- Modules have low module complexity.
- Modules have good documentation (a high number of non-blank comment lines).
- Modules have few external dependencies (low fan-in and fan-out, few input and output parameters).
- Modules have proven reliability (thorough testing and low error rates).

4.3 Use Case Diagram

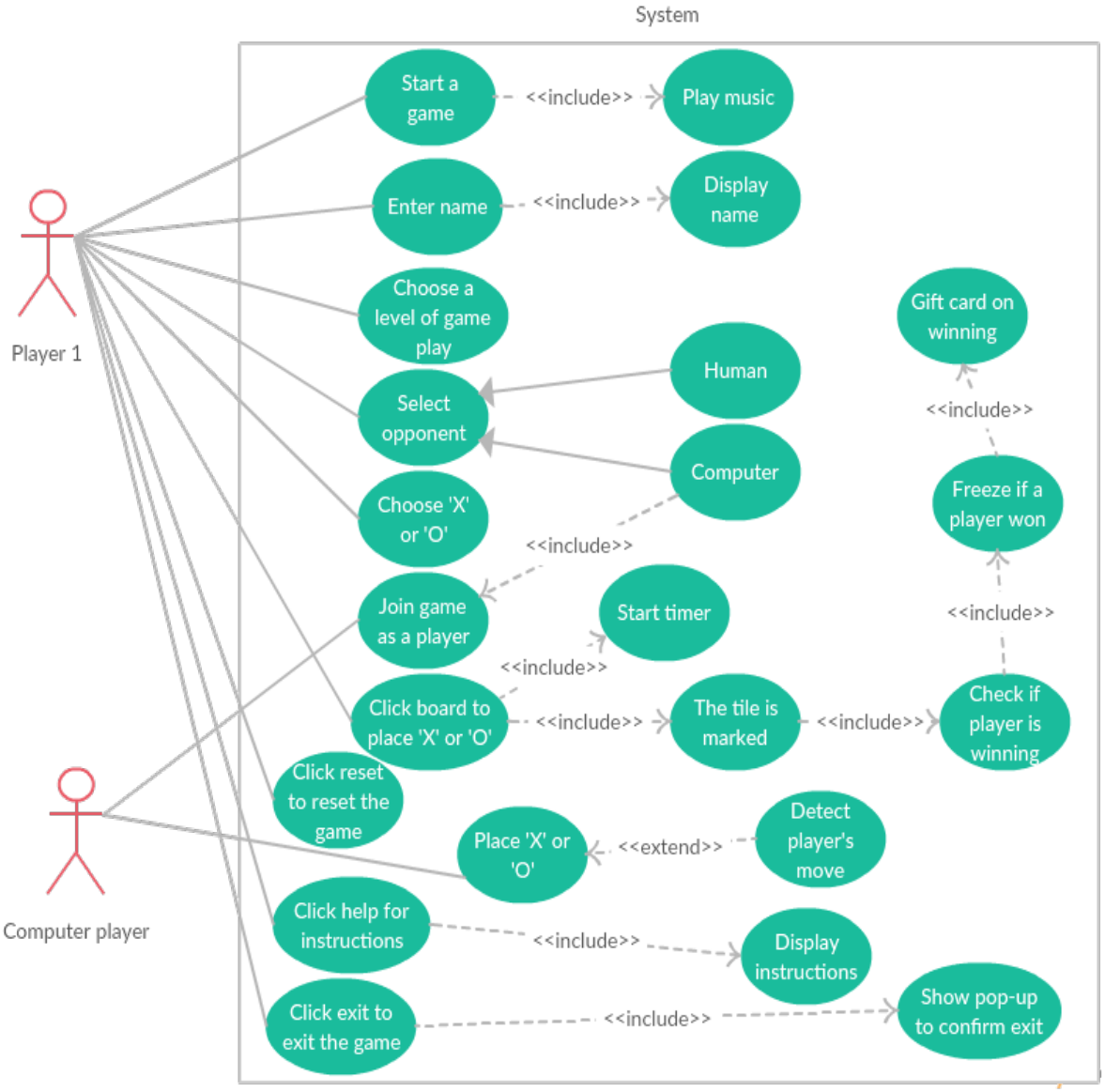


Figure 3. Use case for Deliverable 3

4.4 Use Case Scenario

Select Opponent

Identifier	Explanation
Use Case ID	UID-001
Use case name	Select opponent

Description	The players have the option to select their opponent (e.g. human player or computer).
Primary Actor	Player 1
Priority	Critical
Pre-conditions	1. Players have started the game. 2. Opponent option is available.
Basic Flow	1. Any player starts a game 2. System asks the player to choose their opponent. 3. The player can either click the "human player" button or "computer" button.
Post-condition	The player can start the game with the opponent which he/she has chosen.
Alternate-flow	1. If the player trying to click both options which are human player and computer, the system should jump out a warning window.

Reset game

Identifier	Explanation
Use Case ID	UID-002
Use case name	Reset game
Description	The players have the option to reset the game
Primary Actor	Player
Priority	Critical
Pre-conditions	1. The gaming is running. 2. The player clicks the "reset" button.
Basic Flow	1. Players are playing the game. 2. Once the player wants to reset the game, he/she clicks the reset button.
Post-condition	The game board has been resetting after the player clicks the "reset" button.
Alternate-flow	2a. If the reset button is not functional, then system should show some appropriate message.

Join game

Identifier	Explanation
Use Case ID	UID-003
Use case name	Join game
Description	The computer player can join the game after the human player finishes the selection of the opponent.
Primary Actor	Computer player.
Priority	Critical
Pre-conditions	1. The human player has finished the selection of the opponent. 2. The option must be computer player.
Basic Flow	1. The human player decided to play with computer opponent. 2. The human player clicks the button. 3. The computer players join the game.
Post-condition	The computer player can successfully join the game.
Alternate-flow	The computer player cannot enter the game if human player does not make their option.

Exit game

Identifier	Explanation
Use Case ID	UID-004
Use case name	Exit game
Description	The player can exit the game whenever he/she wants.
Primary Actor	Player.
Priority	Critical
Pre-conditions	1. The player is playing the game. 2. The “exit” button works well.

Basic Flow	1. The player decides to exit the game. 2. Click the “exit” button. 3. Player successfully exit the game
Post-condition	The player can successfully exit the game.
Alternate-flow	2a. If the exit button is not functional, then system should show some appropriate message.