# Problem

# Background Information

The game of tic tac toe essentially consists of “X” and “O” and the objective of the game is to form a line of “X’s” or “O’s” in horizontal, vertical or diagonal form to win the game. We intend to develop a playable game of TIC TAC TOE, which would be delivered in 3 iterations. The first iteration would provide a GUI for the game, the second iteration would provide with a two player game where users can complete with each other and the third deliverable tends to deliver a game which can be played against the computer at different levels. The game would be developed for both Android and desktop using java platform.

# **Deliverable 1**

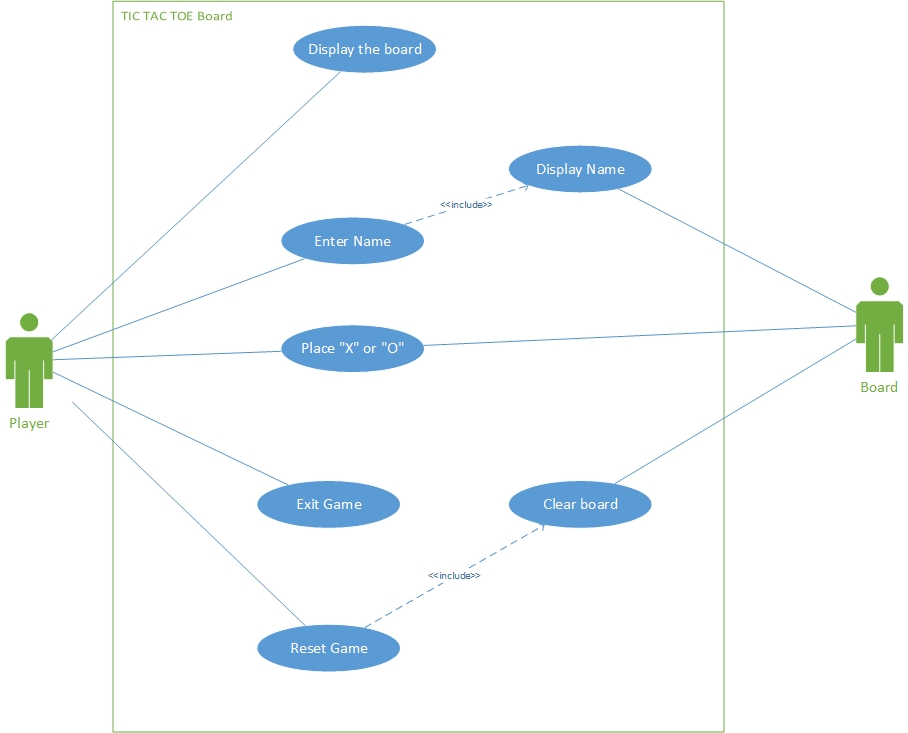
## Functional Requirements

* User can start a new game.
* System should display a 3\*3 board to user.
* If user clicks on any square on a 3\*3 board it should display “X” first and secondly “O” and then vice versa for further clicks.
* Users can enter their name and can reset the board.
* User can exit the game anytime.

## Non Functional Requirements

* **Learnability**: The system should be easy to learn for user as it will include help option.
* **Responsiveness:** The system should be ready to respond to a user’s input no matter how frequently user clicks on square board.
* **Usability**: As it automatically displays the symbols by clicking on any box so it’s easy to play.
* **Modifiability:** It should be easy to make any changes in the system without needing to be rebuilt.
* **Understandability:** User should easily comprehend what system does.

## Use Case Diagram///no board



## Use Case Scenarios

|  |  |
| --- | --- |
| Use Case ID | UC1 |
| Use Case Name | Display the board |
| Primary Actor | Player |
| Pre Conditions | Game should be downloaded in the system |
| Success Guarantee (Postconditions) | Game board is displayed to user |
| Main Success Scenario | 1. The user click on start button  2. System displays the game board |

|  |  |
| --- | --- |
| Use Case ID | UC2 |
| Use Case Name | Place “X” or “O” |
| Primary Actor | Player |
| Secondary actor | Game Board |
| Pre Conditions | There should be available space for user to click |
| Success Guarantee (Postconditions) | “X” or “O” is displayed to user |
| Main Success Scenario | 1. The user starts the game  2. The game board is displayed on the screen  3. The user clicks on any of the 9 spaces on the board  4. The user click will display either “X” or “O” |
| Extensions (Alternative Scenario) | 3.a. There is no available space on the board, so the board will be reset. |

|  |  |
| --- | --- |
| Use Case ID | UC3 |
| Use Case Name | Reset the game |
| Primary Actor | Player |
| Secondary actor | Game Board |
| Pre Conditions | 1. Game should be downloaded in the system 2. Game board should be displayed to the user |
| Success Guarantee (Postconditions) | Clear game board should be displayed to the player |
| Main Success Scenario | 1. The user clicks reset button. 2. System clears the game board. |

# **Deliverable 2**

## Functional Requirements

* User should be able to start a new game.
* The system should be able to maintain the scoreboard of the players playing the game
* The system should allow the user to play the game in tournament mode for best of 3 or best of 5 games.
* The system should show the status and the turn of each player for each move.
* The player should be able to enter their names, which would be displayed on the screen.
* As soon as first player is able to make a single line of ‘X’ horizontally, vertically or diagonally or second player is able to make a single line of ‘O’ horizontally, vertically or diagonally, a dialogue box should pop up which should display who is the winner of the game.

## Non Functional Requirements

* **Reliability –** The system should give proper responses as per the events and should not hang up or crash during the game.
* **Performance**: System should be fast enough to display the user selection and it should not involve the delay of more than 1.5 seconds.
* **Scalability**: Tic tac toe can support all the future advancements keeping all the previous functionality.
* **Testability**: As no major functionality is involved therefore it is easy to test.
* **Portability**: System can run on different versions of the android.

## Use Case Diagram

## Use Case Scenarios

|  |  |
| --- | --- |
| Use Case ID | UC1 |
| Use Case Name | Play game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The game should be installed in the mobile phone. |
| Success Guarantee (Postconditions) | The game runs successfully. |
| Main Success Scenario | 1. The user clicks anywhere on the square board.  2. ‘X’ will be displaced for the first move of the first player and ‘O’ for the second player.  3. Then, the moves will be displaced alternatively. |
| Extensions (Alternative Scenario) | 1a. The user makes a wrong move. |

|  |  |
| --- | --- |
| Use Case ID | UC2 |
| Use Case Name | Exit game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | Game starts successfully. |
| Success Guarantee (Postconditions) | Exit successfully and display clear game board. |
| Main Success Scenario | 1. Player 1 and player 2 play the game against each other.  2. When one player wins the game or game draws then  the system display exitand start a new game option.  3. The player chooses to exit the game by clicking exit button.  4. System clears the game board. |
| Extensions (Alternative Scenario) | 3(a). Exit button is disabled. |

|  |  |
| --- | --- |
| Use Case ID | UC3 |
| Use Case Name | Win game |
| Primary Actor | Tic Tac Toe Player 1, Player 2 |
| Pre Conditions | Players start playing the game. |
| Success Guarantee (Postconditions) | Winning message is displayed on the screen. |
| Main Success Scenario | 1. The user clicks anywhere on the game board.  2. ‘X’ will be displaced for the first move of the first player and ‘O’ for the second player.  3. Then, the moves will be displaced alternatively.  4. One player successfully places his marks in horizontal, vertical or diagonal rows.  5. That player wins the game  6. The game ends and wining message is displayed. |
| Extensions (Alternative Scenario) | . 5 a. Game continues even after the winning.  5b. Winning message not displayed. |

|  |  |
| --- | --- |
| Use Case ID | UC4 |
| Use Case Name | New game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The game successfully ends. |
| Success Guarantee (Postconditions) | The new clear game board is displayed.  . |
| Main Success Scenario | 1. Players start playing the game.  2. Game successfully completed win or draw.  3. Then the user can again start a new game by clicking on the start button.  4. The new game board is displayed. |
| Extensions (Alternative Scenario) | 3(a). User click on start button before the game ends. |

# **Deliverable 3**

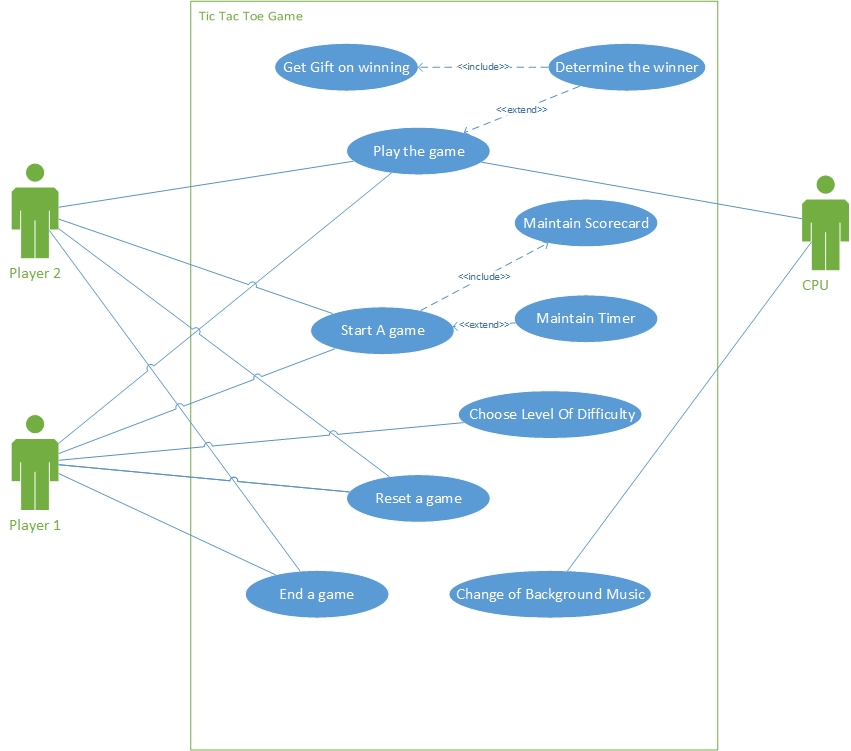
## Functional Requirements

* The heuristics in a game can be made time restricted for every level of the game.
* The CPU must be able to play the game against a human player.
* The game must have different levels of difficulty to play against the CPU.
* The system maintains a scoreboard for each session.
* The background music for playing against the CPU should be different from the background music when playing against another player.

## Non Functional Requirements

1. **Performance**: System should be fast enough to display the user selection and it should not involve the delay of more than 2 seconds.
2. **Reliability**: System should deliver the correct results despite of certain amount of failures. System should work with a reliability of 99 percent, which means in a day player can fail to play no more than 1 % ,or approximately 15 minutes.
3. **Scalability**: System should support the increase use and support the advancements.
4. **Usability**: User should be able to use the system without any difficulty. As it automatically displays the symbols by clicking on any box so it’s easy to play.
5. **Maintainability:** This relates to the ease at which your system finds bugs and fixes them.

## Use Case Diagram// calculate next move



## Use Case Scenarios

|  |  |
| --- | --- |
| Use Case ID | UC1 |
| Use Case Name | Play against the CPU |
| Primary Actor | Tic Tac Toe Player |
| Secondary Actor | Game Board |
| Pre Conditions | The user should chose to play against CPU |
| Success Guarantee (Postconditions) | The game should have a result |
| Main Success Scenario | 1. The user chooses their level of difficulty against the CPU.  2. The players make their moves against the CPU.  2. The CPU counter attacks the player based on the difficulty.  4. The game/tournament ends with a definitive result. |
| Extensions (Alternative Scenario) | 2a. The player can reset or end the game at any move. |

|  |  |
| --- | --- |
| Use Case ID | UC2 |
| Use Case Name | Maintain a scoreboard |
| Primary Actor | Tic Tac Toe Player |
| Secondary Actor | Game Board |
| Pre Conditions | The user should successfully start the game |
| Success Guarantee (Postconditions) | The games should have a result |
| Main Success Scenario | 1. The user chooses to play games against human opponents/CPU.  2. The scorecard is update after each game successfully ends during each session of play.  3. The scorecard is reset once the user exits the game. |
| Extensions (Alternative Scenario) | 2a. The player can reset or end the game at any move. |

|  |  |
| --- | --- |
| Use Case ID | UC3 |
| Use Case Name | Setting timer for the game |
| Primary Actor | Tic Tac Toe Player 1 , Player 2 |
| Pre Conditions | The player should select the timer option |
| Success Guarantee (Postconditions) | The new clear game board is displayed with timer for each player.  . |
| Main Success Scenario | 1. Players start playing the game.  2. Player has to make each move within a specified amount of time.  3. Each player makes a move until the game ends as a tie, win/loss.  4. The score adds to the player’s scoreboard. |
| Extensions (Alternative Scenario) | 2(a). Player does not make his/her move within the specified amount of time.  (b) The player instantly loses the game.  (c) The score add’s to the player’s scoreboard. |