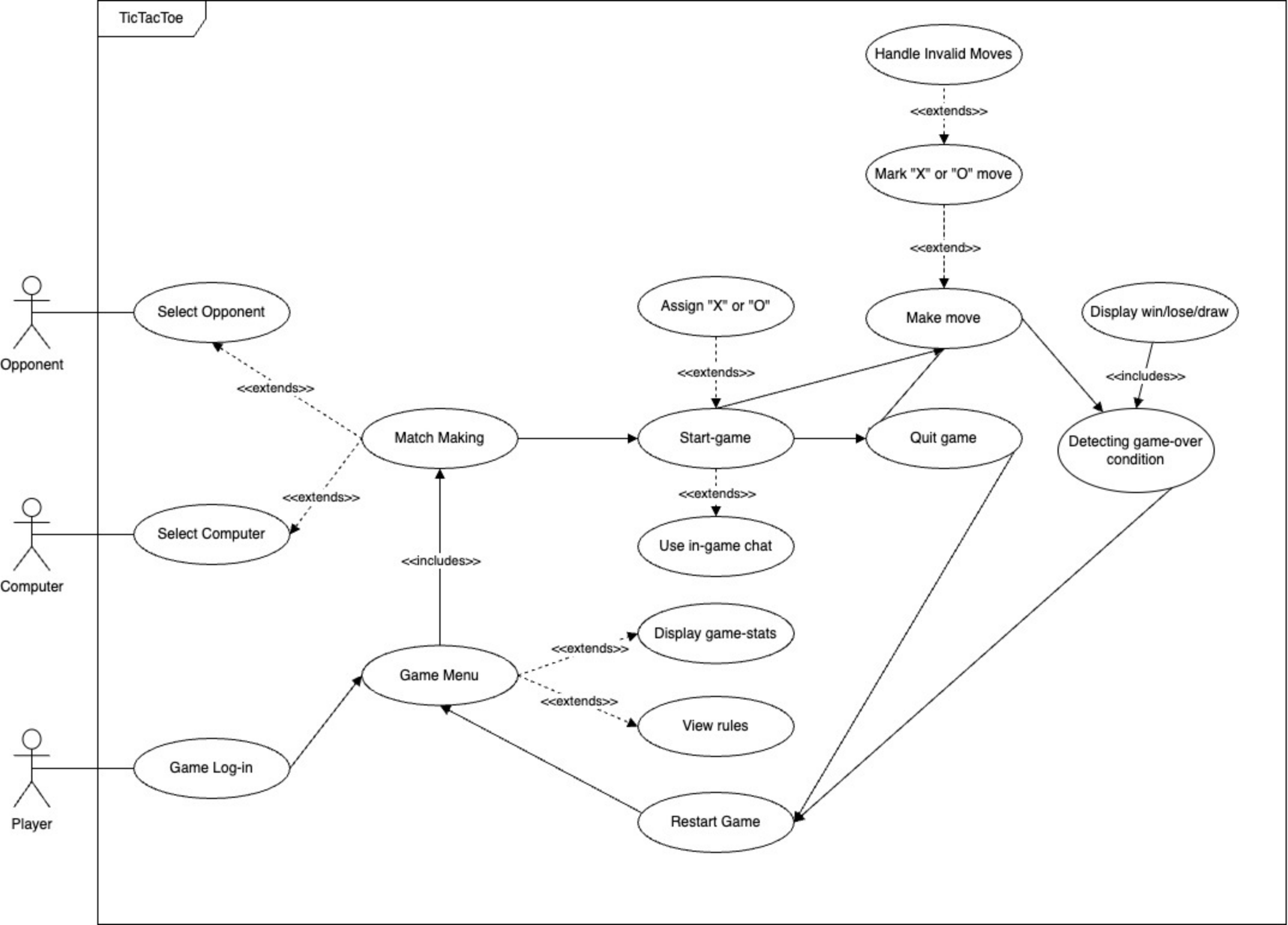

TicTacToe
Handle Invalid Moves
<<extends>>
Mark "X" or "O" move
<<extend>>
Opponent
Select Opponent
Assign "X" or "O"
Make move
Display win/lose/draw
<<extends>>
<<includes>>
<<extends>>
Match Making
Start-game
Quit game
Detecting game-over condition
<<extends>>
<<extends>>
Computer
Select Computer
Use in-game chat
<<includes>>
Display game-stats
<<extends>>
Game Menu
<<extends>>
View rules
Player
Game Log-in
Restart Game

Use Case Descriptions (Tic-Tac-Toe Game)

Use Case 1: Start a new tic-tac-toe game

Iteration: 1, start a new tic-tac-toe game

Primary actor: Player

Goal in context: To start a new session of tic-tac-toe game

Preconditions:

- The player must be successfully logged into the system
- Gui already exists

Trigger:

- Player clicks on the “new game” button

Scenario:

- The player clicks on the new game button.
- Server connection establishes
- Match making takes place
- System initializes a new game board
- System assigns either “X” or “O” to the player
- System waits for the first move from the player

Exceptions:

- If the system is unable to initialize a new game.
- Not enough players on the platform
- Matchmaking fails
- System runs into an unexpected error

Priority:

- High

When available:

- After final testing and review has been done

Frequency of Use:

- Multiple times per session

Channel to actor:

- Graphical User Interface

Secondary actors:

- AI, computer and second player

Channels to secondary actors:

- Matchmaking, server connection

Open Issues:

- How long should the matchmaking system wait before trying?

Use Case 2: Make a move

Iteration: depends on how many moves have previously been made

Primary actor: Player

Goal in context: To successfully make a move on the tic-tac-toe board

Preconditions:

- The game session must be active
- There should be an opponent player
- Multiplayer connection should successfully be established
- There should be blocks on the board to make a move

Trigger:

- Player wants to select a cell on the board

Scenario:

- Player selects an empty cell.
- The system validates the move is in accordance with the rules
- The system recognizes the player's mark ("X" or "O")
- The system checks for a win, draw or lose condition
- The system switches to the opponent player for their move

Exceptions:

- If the system loses connection mid-game, the game "enters" a waiting state.

Priority:

- High

When available:

- After final testing and review

Frequency of Use:

- Multiple times per game

Channel to actor:

- GUI

Secondary actors:

- Opponent player, computer or AI

Channels to secondary actors:

- Matchmaking, server connection and GUI

Open Issues:

- Should there be a time limit per move before system detects for any inactivity.

Use Case 3: Handle invalid moves

Iteration: 2, first move is usually valid because of empty cells

Primary actor: Player

Goal in context: To prevent a player from making an invalid move

Preconditions:

- A game session is already active
- The player must attempt to make a move

Trigger: The player attempts an invalid move

Scenario:

- The player selects a cell that is already occupied with their cursor
- The system recognises the cell is occupied
- The system doesn't allow the player to make a move
- The system displays a message "cell already occupied" when cursor is hovered on an occupied cell

Exceptions:

- If the player doesn't select any move at all

Priority:

- Medium

When available:

- After final testing and review

Frequency of Use:

- Occasionally (when player tries to make an invalid move)

Channel to actor:

- GUI

Secondary actors:

- Opponent player, computer or AI

Channels to secondary actors:

- Matchmaking, server connection and GUI

Open Issues:

- Should the system display an error message prompt if the player repeatedly makes an invalid move.

Use Case 4: View Rules

Iteration: 1

Primary actor: Player

Goal in context: To click a button "Rules" to view rules to the tic-tac-toe game

Preconditions: None

Trigger: Player wants to view game rules

Scenario:

- Player selects "Rules" option from the main menu.
- The system opens a dialogue box with rules for the tic tac toe game
- The player closes the dialogue box with "x" button

Exceptions:

- If the rules page fails to load, should the system display an error message

Priority:

- Low

When available:

- After final testing and review

Frequency of Use:

- Rare

Channel to actor:

- GUI

Secondary actors:

- System

Channels to secondary actors:

- None

Open Issues:

- Should the players be able to view rules during an active game session
- If a player views rule during an active game session how long should the system wait before next turn or inactivity.

Use Case 5: Detect condition for game over

Iteration: depends on the number of moves

Primary actor: Player

Goal in context: To determine when the game ends

Preconditions:

- A game session must be active
- A player played a move

Trigger: A move is made by the player

Scenario:

- Player makes a move.
- System checks whether a player has three consecutive matching cells in a row
- If no win condition and the board is full, system declares a draw

- If no win condition but the board has empty cells, next player takes a turn.
- If win condition detected the system declares the player has won the game.

Exceptions:

- If an error encounter while the system is checking win condition, the system retries.

Priority:

- High

When available:

- After final testing and review

Frequency of Use:

- Very frequent

Channel to actor:

- GUI

Secondary actors:

- Opponent player, computer or AI

Channels to secondary actors:

- Matchmaking or GUI

Open Issues:

- How should stats be handled if game crashes while checking a win condition.

Use Case 6: Display a winner/lose or draw

Iteration: Depend on moves during an active game session

Primary actor: Player

Goal in context: The display game results whether win/lose or draw

Preconditions: The game must have ended in win/lose or draw

Trigger: The system detects for a win/lose or a draw condition

Scenario:

- The player makes a move.
- The system checks whether the player won/lost or ended in draw.
- The system opens a dialogue box and displays “Player X wins”, “Player O wins”

or draw.

Exceptions:

- If any error occurs while displaying message, the system retries showing the message.
- What if the system shows wrong results.

Priority:

- High

When available:

- After final testing and review

Frequency of Use:

- Once per every tic-tac-toe game

Channel to actor:

- GUI

Secondary actors:

- System, opponent player

Channels to secondary actors:

- GUI

Open Issues:

- Should the system save results to be viewed later?

Use Case 7: Quit Game

Iteration: 1

Primary actor: Player

Goal in context: To exit an ongoing game session

Preconditions: A game session must be active

Trigger: Player clicks on the quit game button

Scenario:

- The player clicks on the quit game button
- The system shows a prompt asking the player if they are sure they want to quit the game

- The player confirms their selection
- If player selects no, the game resumes
- If player selects yes, the game returns to main menu

Exceptions:

- If the game crashes or an unexpected error occurs

Priority:

- Medium

When available:

- After final testing and review

Frequency of Use:

- Occasionally

Channel to actor:

- GUI

Secondary actors:

- Opponent player

Channels to secondary actors:

- Server connection, Matchmaking

Open Issues:

- How the stats should be handled if a player exits mid-game?

Use Case 8: Restart Game

Iteration: Depends on how many games played so far

Primary actor: Player

Goal in context: To restart a game after finishing one

Preconditions:

- A game session must have come to an end
- There should be a restart option available to the player

Trigger: Player clicks on the restart game button

Scenario:

- Player finishes an existing game session
- The system opens a dialogue box with two prompts restart game or quit game
- The player clicks on restart game
- The system reconnects matchmaking
- The player enters a new game with new opponent

Exceptions:

- If the system cannot reconnect match making
- if the system fails to restart a game

Priority:

- High

When available:

- After final testing and review

Frequency of Use:

- Frequently

Channel to actor:

- GUI

Secondary actors:

- System, opponent player

Channels to secondary actors:

- Server connection

Open Issues:

- What if the player wants to restart the game but wants to play with the same opponent?

Use Case 9: Use In-Game Chat

Iteration: 1

Primary actor: Player

Goal in context: To send and receive messages from your opponent using in game chat

Preconditions:

- The player should be in a multiplayer session
- A tic tac toe game should be active
- The player should have an online multiplayer opponent

Trigger: The player sends a message

Scenario:

- The player types a message in the chat window
- The system sends the message to the client server
- The client server forwards the message to the opponent
- The opponent receives the message in their chat window

Exceptions:

- If the chat server encounters an error or is down
- If the opponent disconnects the messages are in queue until they reconnect

Priority:

- Medium

When available:

- After final testing and review

Frequency of Use:

- Frequently

Channel to actor:

- GUI

Secondary actors:

- Opponent player

Channels to secondary actors:

- Server connection

Open Issues:

- What if the player or opponent uses offensive language should they be given a warning or be banned

Use Case 10: Display Stats

Iteration: 1

Primary actor: Player

Goal in context: To view a player's game statistics

Preconditions:

- The player should be in main menu
- The player should have had at least played 1 game

Trigger: The player clicks on the view statistics button

Scenario:

- The player clicks on the view statistics button on main menu
- The player views the statistics according to the games
- The player can filter the statistics according to the game
- the player closes the view statistics button

Exceptions:

- If the stats fail to load

Priority:

- Medium

When available:

- After final testing and review

Frequency of Use:

- Occasionally

Channel to actor:

- GUI

Secondary actors:

- System database

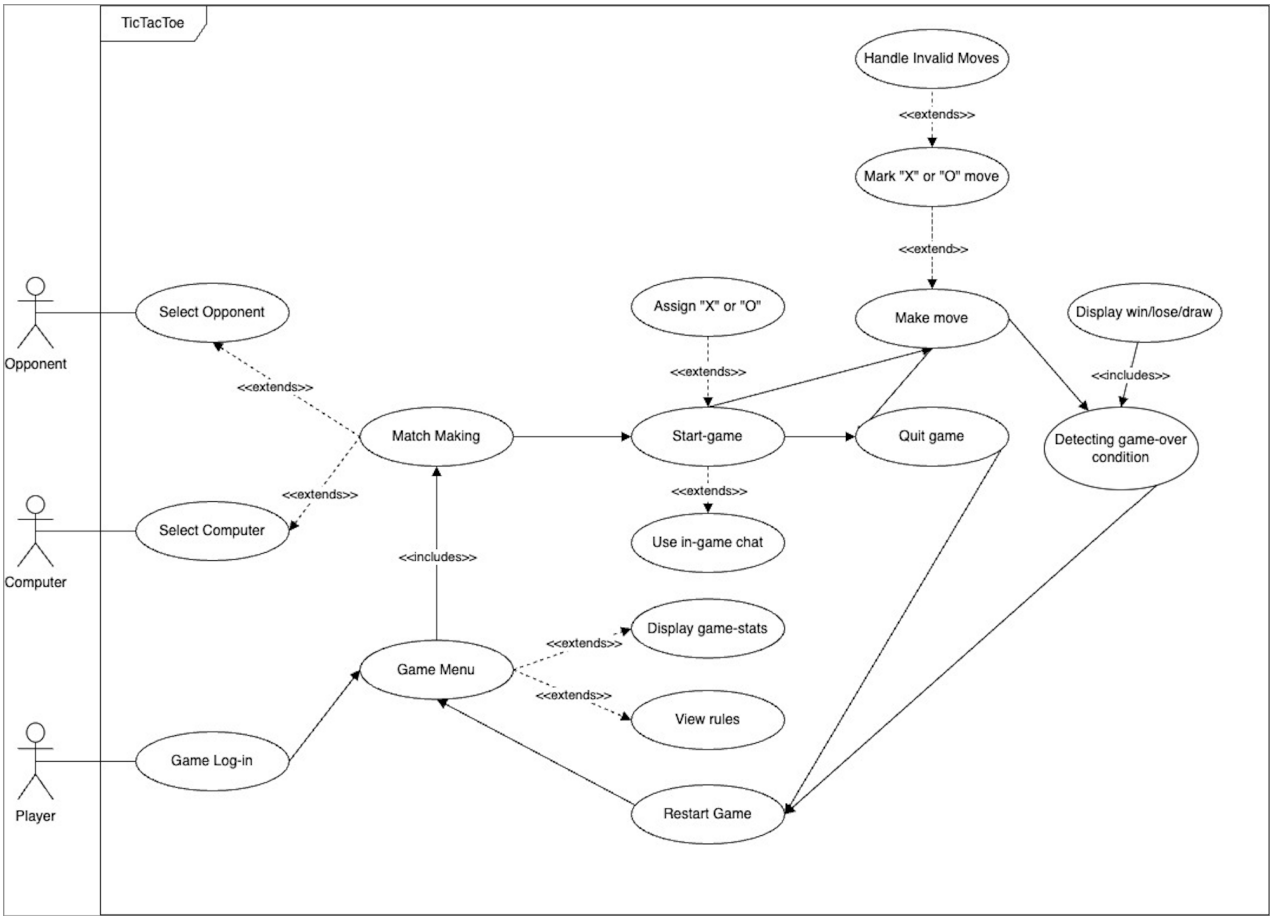
Channels to secondary actors:

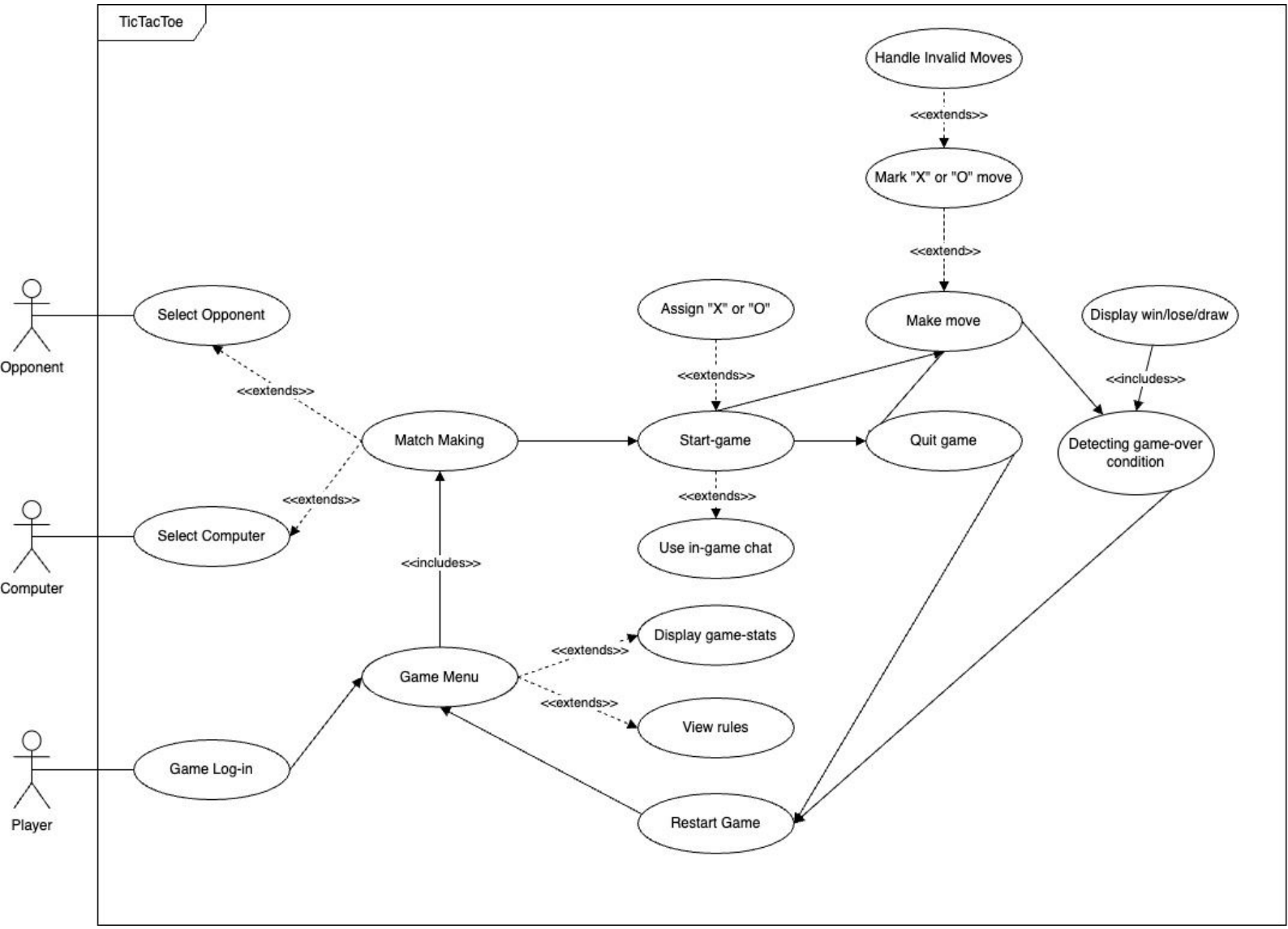
- Remote storage

Open Issues:

- Should the system allow the stats to be reset.

Use Case Diagram:




TicTacToe
Handle Invalid Moves
<<extends>>
Mark "X" or "O" move
<<extend>>
Opponent
Select Opponent
Assign "X" or "O"
Make move
Display win/lose/draw
<<extends>>
<<includes>>
<<extends>>
Match Making
Start-game
Quit game
Detecting game-over condition
<<extends>>
<<extends>>
Computer
Select Computer
Use in-game chat
<<includes>>
Display game-stats
<<extends>>
Game Menu
<<extends>>
View rules
Player
Game Log-in
Restart Game