CA314 OO ANALYSIS AND DESIGN

THE GAME OF GO

GROUP 13 - GROUP MEMBERS

Student Name	**Student Number**
Cathal Hughes	15417922
Russell Brady	15534623
Daniel Allen	15361731
Riain Condon	15363466
Mikhail Arkhangelskiy	15443198

PLAGIARISM DECLARATION

TABLE OF CONTENTS

REFINED REQUIREMENTS SPECIFICATION

PROJECT OVERVIEW

The goal of our assignment is to implement an online head-to-head version of the board game Go. It is our intention for the game to be Internet-based, with the users to have the ability to play other random users online. This application will take on a client/server model. Each client will support a singular player, while the server coordinates communication between the two players and the game sequence.

Go is a strategic game where players try to take over as much territory on the board as possible. This is done by placing their team's coloured stones on the board. Opponents' pieces can be captured if they are fully surrounded by the players' pieces.

GAME ELEMENTS

Users are given black or white pieces (Based on experience level, see 1.5). Each piece when placed will take a location in a square on a 19 x 19 chess-styled board. The pieces are fixed in these positions for the full length of the game and cannot be moved unless they are captured by the opponent. In this case, they are removed from the board and their previous square is vacant.

The user interface should also contain a leader board, showing the win-loss record of the user and the top 10 users.

THE GAME SEQUENCE OF EVENTS

Firstly, the user must log onto their Go account. The user will join a game or wait for an opponent to join their game. The more experienced user (Player 1) is given white pieces, and the less is given black pieces (Player 2). Player 2 moves first, placing a black piece on the board. Player 1 then makes their move. Players go back and forward making moves, but also have the option to pass on their turn. This continues until both players pass consecutively.

MOVING AND TAKING TERRITORIES

Users capture their opponents' pieces by completely surrounding the other players' piece. The opponent's piece will be removed from the board and the space will be vacant.

SPECIAL RULES

Experienced User Rule – When a game begins, the more experience player is given white pieces and the lesser is given black pieces. The latter player is given the first move while the more experienced player is given 6.5 points to start with.

Suicide Rule – If there is a vacant space which is completely surrounded by one players' pieces then the other player cannot place their piece in this vacant spot. This is to stop the player automatically losing their piece.

KO Rule – This rule stops a looping situation. A move is illegal if a move will create a position that has previously existed in the game. This stops players "reversing" a capture that has just happened.

WINNING THE GAME

The game ends when both players consecutively pass on their turns. Each territory that a player has captured earns them a point, with bonus points given to the more experienced player. The player with the most points wins.

SCENARIOS

SCENARIO: THE FIRST USER CONNECTS TO THE SERVER

CURRENT SYSTEM STATE

The system state consists of no players connected to the server.

INFORMAL SCENARIO

Player connects to the server. The Player can start a game. The player has to wait for another player to join. He selects his colour as he was first to join.

NEXT SCENARIO

Player waits for opponent to join.

SCENARIO: AN ADDITIONAL USER CONNECTS TO THE SERVER AND JOINS THE GAME

CURRENT SYSTEM STATE

The system state consists of Player 1 waiting another player to connect to the server.

INFORMAL SCENARIO

Player 2 (opponent) joins and is given the other remaining colour.

NEXT SCENARIO

Now that the game has two players the game initialises.

SCENARIO: GAME INITIALISES CURRENT SYSTEM STATE

CURRENT SYSTEM STATE

The system state consists of two players connected to the server waiting for the game to start.

INFORMAL SCENARIO

The board has appeared on screen and whoever is chose Black must make the first move.

NEXT SCENARIO

The player who has the black stones makes a move.

SCENARIO: USER PLACES STONE AND SURRONDS OPPONENTS STONE

CURRENT SYSTEM STATE

The system state consists of two players Player 1 and Player 2. Player 1 has the black stones and so they went first. They are playing on a standard Go board.

INFORMAL SCENARIO

Player 1 surrounds a piece belonging to Player 2 and so that piece is captured and removed from the board.

NEXT SCENARIO

Player 2 surrounds a piece belonging to player 1

SCENARIO: OPPONENT SURRONDS PLAYER 1'S STONE

CURRENT SYSTEM STATE

The system state consists of two. Player 2 wants to capture a stone belonging to player 1.

INFORMAL SCENARIO

Player 2 sees a stone belonging to player 1 that is surrounded on three sides and decides to capture it by placing his stone and surrounding the final open side. Player 1's stone is removed from the board.

NEXT SCENARIO

Player 1 tries to make a move that will return the board to its's previous state

SCENARIO: USER TRIES TO RETURN BOARD TO PREVIOUS STATE WITH A MOVE

CURRENT SYSTEM STATE

The game consists of two players. Player 2 has just captured a stone belonging to player 1 and player 1 wants to capture a stone of player 2's which will return the board to the previous state.

INFORMAL SCENARIO

Player 1 makes the move but is unable to as a dialogue pops up saying how they are breaking the Ko rule. Player 1 is invited to make another move.

NEXT SCENARIO

Player 1 makes another move and ends their turn.

SCENARIO: USER BREAKS THE SUICIDE RULE

CURRENT SYSTEM STATE

The system state consists of two players and the board is now becoming very full.

INFORMAL SCENARIO

The board is now starting to fill up and making a good move is becoming hard. Player 1 places a stone on the board that will immediately be captured if the move was allowed; instead a dialogue appears saying this move is in breach of the suicide rule and the player is invited to make an alternative move or pass the move. Player 1 passes and so Player 2 places a stone on the board.

NEXT SCENARIO

The end of game looms as the board is now quite full and no moves will be worth playing.

SCENARIO: THE GAME ENDS AFTER TWO CONSECUTIVE PASSES

CURRENT SYSTEM STATE

The system state consists of two players. The board is so full that neither player can see a good move.

INFORMAL SCENARIO

Player 1 is unable to make a move and so decides to pass. Player 2 also passes after much deliberation. The game is thus ended and the server prevents anyone making another move. It then calculates the territories.

NEXT SCENARIO

The server then declares the winner to the players

.

PRIMARY CLASS LIST

GAME CLASS

Class Name: Game	ID: 1	Type:
Description:	Associated Use Cases:	
The game class initialises a game of Go between two players.	Player starts a new game Player joins a game of Go Player disconnects from game Player wins a game of Go	
Responsibilities	Collaborators	
Initialising a game	Board Class / Player Class / Stone Class	
Initialising a board	Board Class	
Ending a game	Player Class	
Attributes		
Board	Player 1	
Board Size	Player 2	

Moves made	Game in play
Suicide rule broken	Ko rule broken
Relationships	
Aggregation (has-parts)	
Player (2)	Board
Stone	

BOARD CLASS

Class Name: Board	ID: 2	Type:

Description: Initialises a board and keeps track of the board after every move is made.	Associated Use Cases: Player makes a move on the board Board is initialised at the start of a game
Responsibilities	Collaborators
Update a board after a move	Player Class
Place a stone on board	Stone Class
Make a move	Player Class
Attributes	
Size	Colour
Relationships	
Aggregation (has-parts)	
Stone	

PLAYER CLASS

Class Name: Player	ID: 3	Type:

Description:	Associated Use Cases:
The Player Class makes a move or passes on a move. Player can quit a game.	Player makes a move Player quits a game Player passes on a move Player starts / joins a game.
Responsibilities	Collaborators
Make a move	Board Class, Stone Class
Pass on a move	
Attributes	
ID	Next Move
Stone	
Relationships	
Aggregation (has-parts)	
Stone	
Other Associations	
Server	Board

STONE CLASS

Class Name: Stone	ID: 4	Type:

Description:	Associated Use Cases:
The stone class encapsulates a stone on a board. The stone keeps track of its position and whether it has been captured or not.	Player places stone on a board Player chooses stone colour
Responsibilities	Collaborators
Check if captured	Board Class
Attributes	
Colour	Is Captured
Position	

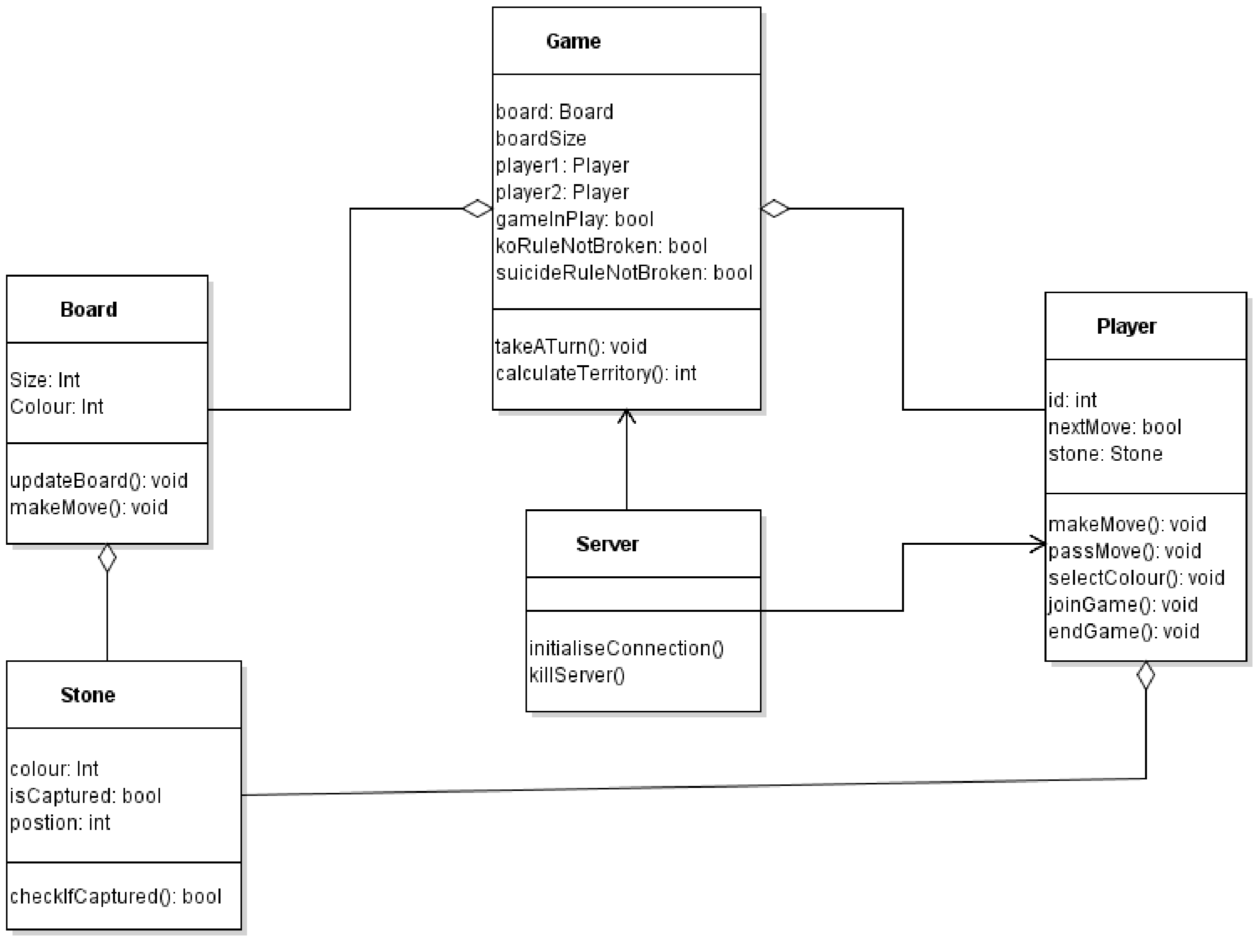
SERVER CLASS

Class Name: Server	ID: 5	Type:
Description: The server class initialises connection between a player and the game and keeps connection during the course of the game. It connects 2 players to a game.		Associated Use Cases: Player starts a new game Player joins a game Player disconnects from game Player waits for an opponent to connect

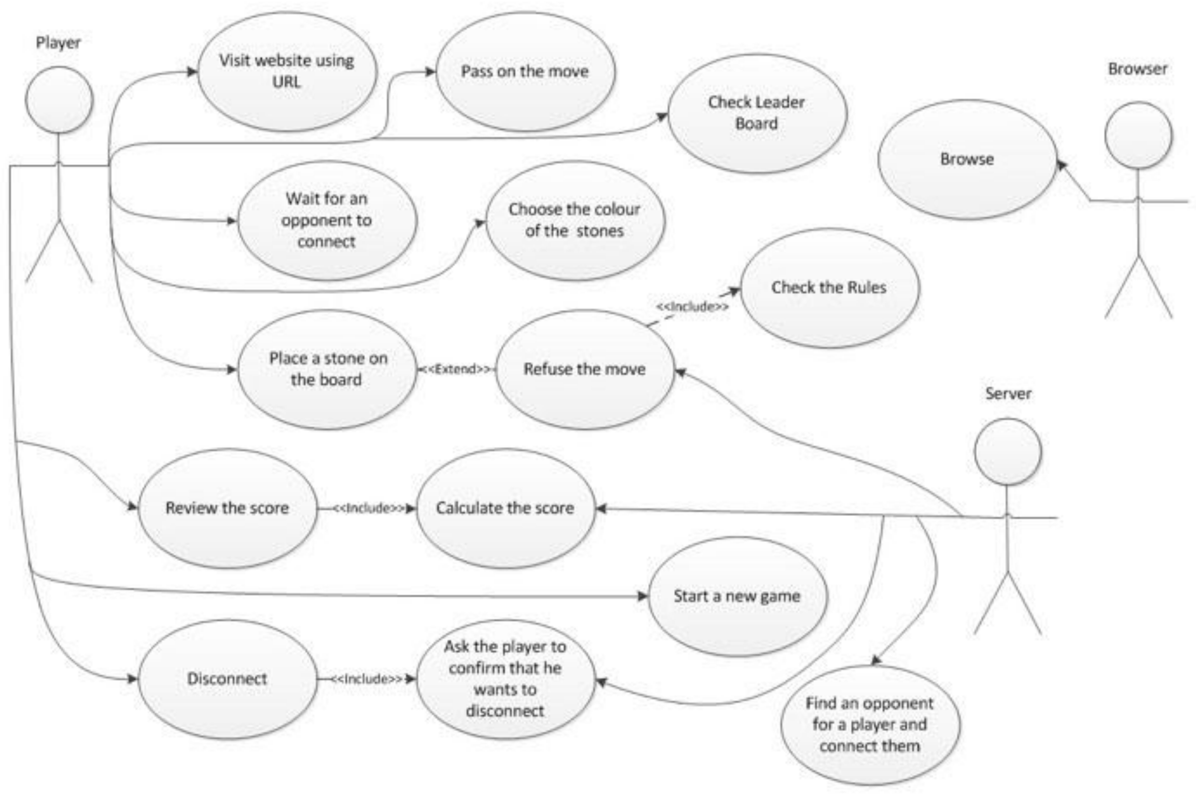
Responsibilities	Collaborators
Initialise connection	Player Class. Game class
Kill connection	Player Class, Game class

Relationships	
Other Associations	
The Game class and Player class rely on the server	

CLASS DIAGRAM



USE CASE DIAGRAM



USE CASE DESCRIPTIONS

USE CASE 1	Start a new game	
Goal in Context	Complete a game of Go against an opponent	
Scope & Level	System, Core.	
Preconditions	User has visited website and has opted to play against an opponent	
Success End Condition	The game is completed and the scores are shown	
Failed End Condition	The game is not completed and the user is returned to the main menu	
Primary, **Secondary Actors**	Player 1 Opponent (Player 2), Game	
Trigger	Player visits website	
DESCRIPTION	**Step**	**Action**
	1	Player 1 visits website using URL
	2	Player opts to play a game of Go against an opponent
	3	Player picks the stone colour they prefer (White or Black)
	4	Player 1 waits while they wait for an opponent to connect.
	5	The opponent connects.
	6	Player 1 is black and makes their first move by selecting the coordinates for the first stone

	7	Player 1 captures a stone by surrounding the stones of player 2 and vice versa
	8	The game ends when both players pass on a move consecutively.
	9	The total score each player is added up and the winner is displayed
EXTENSIONS	**Step**	**Branching Action**
	1a	Website is down
	5a	An opponent fails to connect

USE CASE 2	Quit a game a Go (Disconnect)
Goal in Context	Quit a game midway through a game
Scope & Level	System, Core.
Preconditions	User has visited website and has opted to play against an opponent
Success End Condition	The game is quit halfway by the Player
Failed End Condition	The game is quit but the user is not returned to the main menu.
Primary, **Secondary Actors**	Player 1 Opponent (Player 2), Game
Trigger	Player is connected to the server.

DESCRIPTION	**Step**	**Action**
	1	Player 1 visits website using URL
	2	Player opts to play a game of Go against an opponent
	3	Player picks the stone colour they prefer (White or Black)
	4	Player 1 waits while they wait for an opponent to connect.
	5	The opponent connects.
	6	Player 1 is black and makes their first move by selecting the coordinates for the first stone
	7	Player 1 captures a stone by surrounding the stones of player 2 and vice versa
	8	Before the game is completed player 1 decides to quit the game.
	9	Player 1 is presented with a dialogue asking are they sure and they click yes.
	10	Both players are returned to the main menu.
EXTENSIONS	**Step**	**Branching Action**
	5a	An opponent fails to connect
	10a	The player isn’t returned to the main menu.

USE CASE 3	Player 1 passes on a move
Goal in Context	The player cannot see a good move and so opts to pass.
Scope & Level	System, Core.

Preconditions	User has visited website and has opted to play against an opponent	
Success End Condition	Player 1passes on their move and now it's the opponents move	
Failed End Condition	Player 1 is unable to pass on a move	
Primary, **Secondary Actors**	Player 1 Opponent (Player 2), Game	
Trigger	Player is connected to a server and is mid game	
DESCRIPTION	**Step**	**Action**
	1	Player 1 visits website using URL
	2	Player opts to play a game of Go against an opponent
	3	Player picks the stone colour they prefer (White or Black)
	4	Player 1 waits while they wait for an opponent to connect.
	5	The opponent connects.
	6	Player 1 is black and makes their first move by selecting the coordinates for the first stone
	7	Player 1 captures a stone by surrounding the stones of player 2 and vice versa
	8	The board is now becoming more full and Player 1 is struggling to see a move to play.
	9	Player 1 thus decides to pass on a move and it is now the opponents move

	10	The opponent makes their move.
EXTENSIONS	**Step**	**Branching Action**
	5a	An opponent fails to connect
	10a	The opponents opts to pass also and the game is over.

USE CASE 4	Player checks leaderboard.	
Goal in Context	The player visits website to see how they have done in their previous games.	
Scope & Level	System, Core.	
Preconditions	User has visited website.	
Success End Condition	The player is shown a screen displaying the world leaderboard	
Failed End Condition	The player is unable to see the leaderboard	
Primary, **Secondary Actors**	Player 1 Game	
Trigger	Player visits website	
DESCRIPTION	**Step**	**Action**
	1	Player 1 visits website using URL
	2	Player opts to view leaderboard
	3	Player is shown top 10 players in the world

EXTENSIONS	**Step**	**Branching Action**
	2a	Leaderboard is not visible.

RESULTS OF STRUCTURED WALKTHROUGH

Our team met to discuss the potential errors that may exist in this project and run through the structured walkthrough.
We walked through the specification as a team and came up with any potential errors that could arise or any ambiguities in the game. We kept a list of the issues encountered. For any disagreements the group came to a mutual resolution to the problem. The results of our structured walkthrough include a guide through the Game of Go using Use cases and classes as well as a series of questions and answers to some of the potential problems we identified.

GUIDE THROUGH THE GAME

The user visits the website and decides to start a new game of Go. Player 1 waits while another player joins the game. Once the opponent joins, the server initialises the connection. Both players can make a move by placing a stone on the board or pass on a move. If either player wants to disconnect the server will close the connection and both players will be returned to the main menu. On completing a game both players can initialise a new game or return to the main menu. At the main menu players can check the world leader board or view the rules on the homepage.

QUESTIONS & ANSWERS

Question: What happens if a Player disconnects mid game?

Answer: The users will be returned to the main menu.

Question: What happens if a user disconnects during game initialization?

Answer: The users will be returned to the main menu.

Question: What happens if a user disconnects during waiting for other user to join?

Answer: The user will be returned to the main menu.

Question: What happens if the user that created the game disconnects?

Answer: The users will be returned to the main menu.

Question: Should we allow multiple games to happen on the server?

Answer: Yes.

Question: After a game are both players invited to play another game?

Answer: Yes.

Question: Will a player be informed if they play an invalid move?

Answer: Yes.

GROUP MINUTES AND AGENDA

MEETING 1 - 28/09/17

- All Members present
- Cathal took note of the minutes
- We discussed possible code / bug management system and how we were going to stay in contact during the course of the project - we decided on slack as our means of communication
- Discussed potential programming languages to implement the project in
- We decided to play the game before the next meeting and generate some requirements

MEETING 2 - 02/10/17

- All members present
- Cathal recorded the minutes
- A stand-up took place where every member discussed potential requirements based on having played the game since the last meeting
- Some suggestions were made as to how to add our own twist to the game
- We decided to split the rest of the work for the first phase between members. Each member was given something to work on. Once a draft of everything was done we could then review what was done and re-draft

MEETING 3 - 11/10/17

- All members present
- Cathal recorded the minutes

- Since the last meeting members had been working on their own part of the analysis phase and sharing their drafts online with members and taking feedback. We met up to see how progress was going for everyone and to iron out any problems encountered
- Members finalised what needed to be re-drafted and we decided to meet up early next week to bring the whole thing together once individual parts had been completed

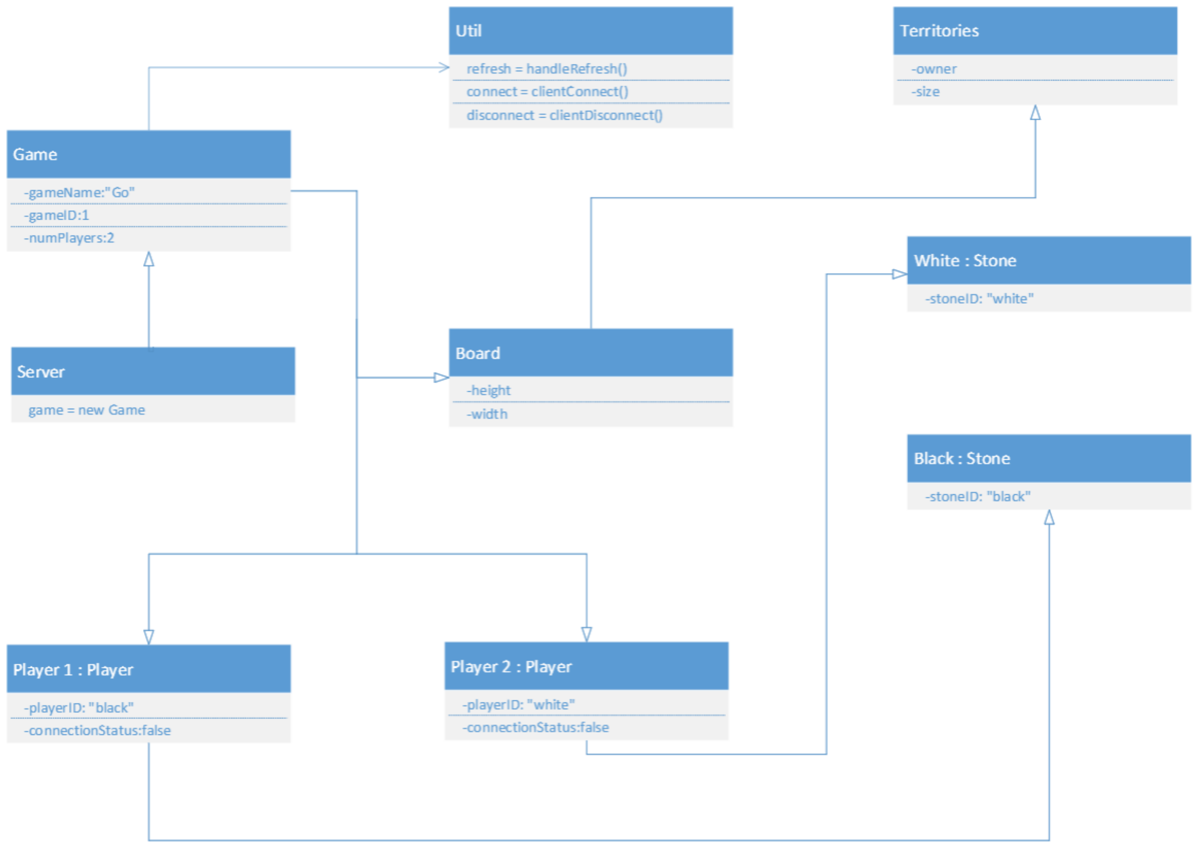
MEETING 4 - 17/10/17

- All members present
- Cathal took note of the minutes
- We brought all completed parts together into one doc to be formatted before our final meeting at the end of the week.

MEETING 5 - 19/10/17

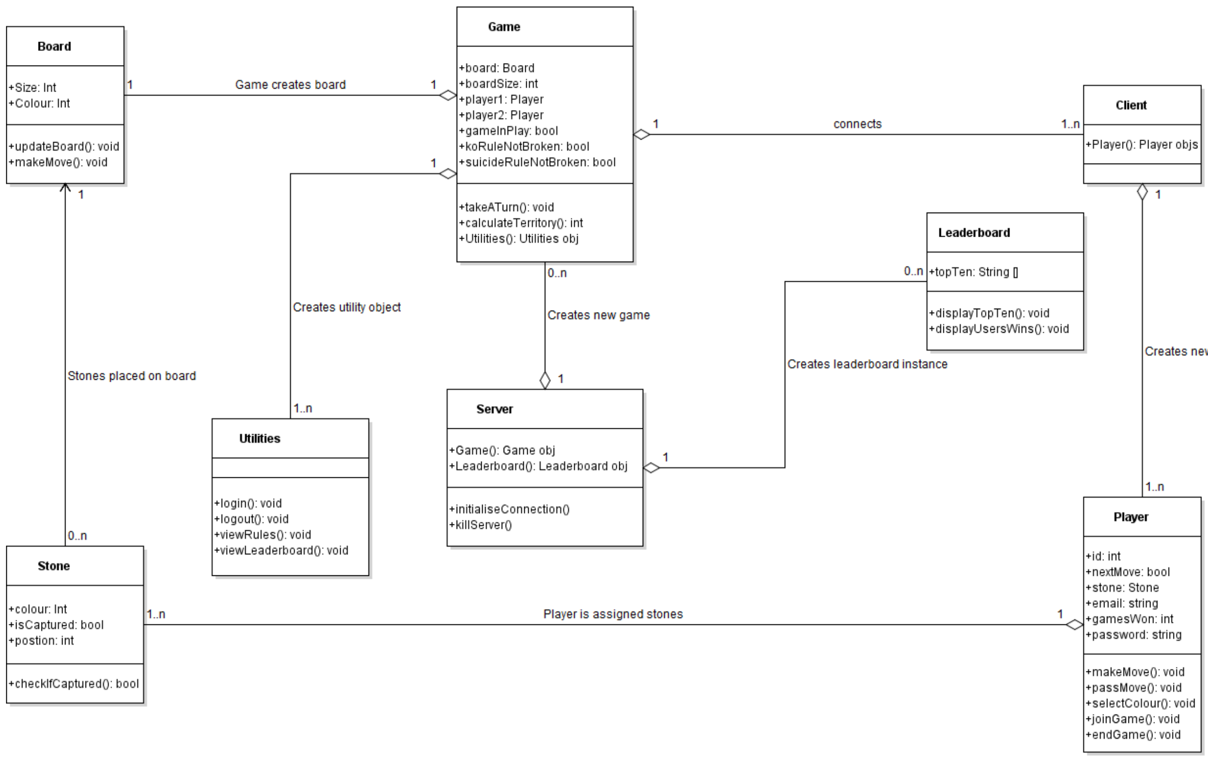
- All members present
- Cathal took note of the minutes
- We updated structured walkthrough in the doc and we reviewed the analysis phase as a whole. All members were happy with the Doc design and it was ready for submission.

OBJECT DIAGRAM

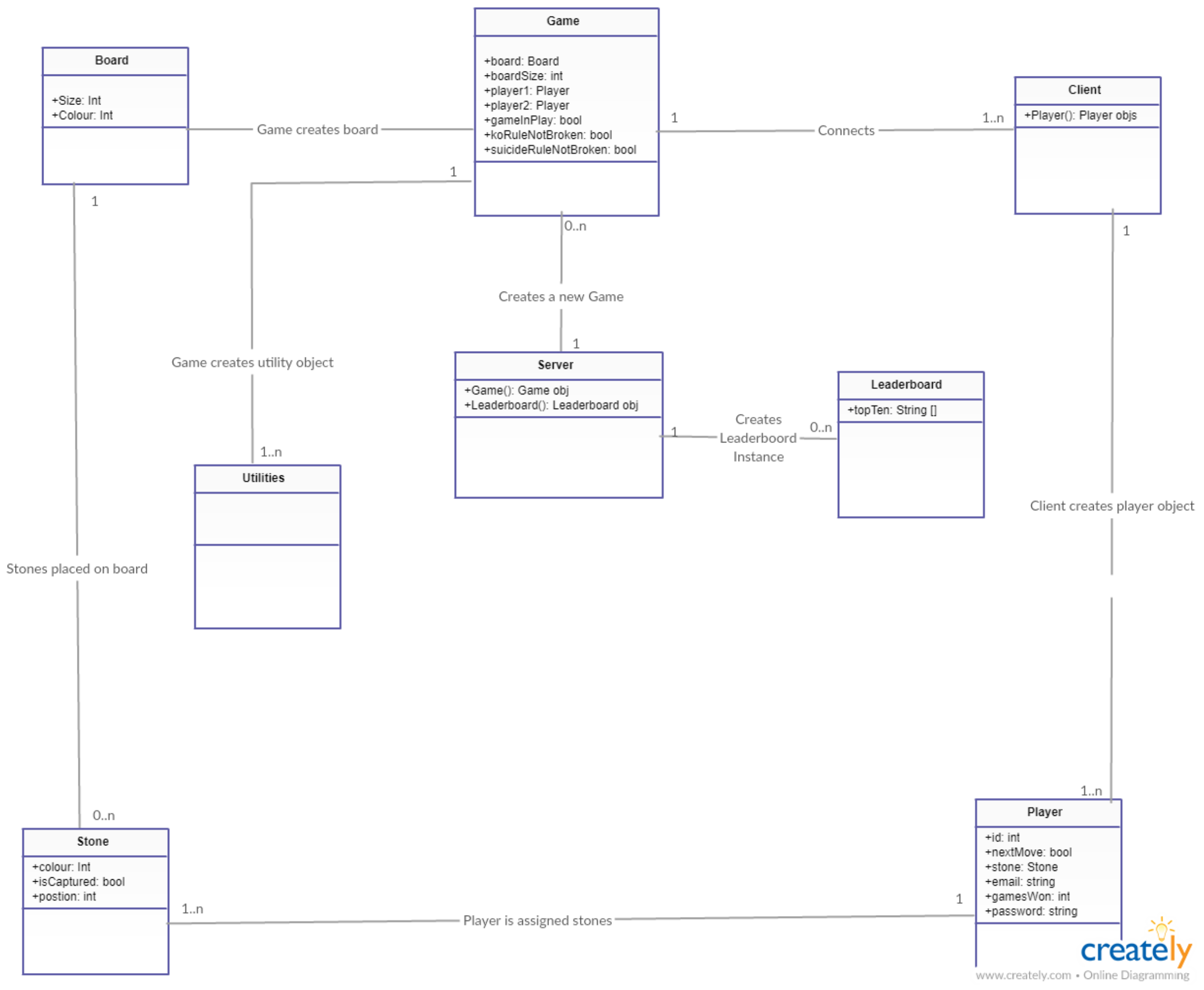


REFINED CLASS DIAGRAMS

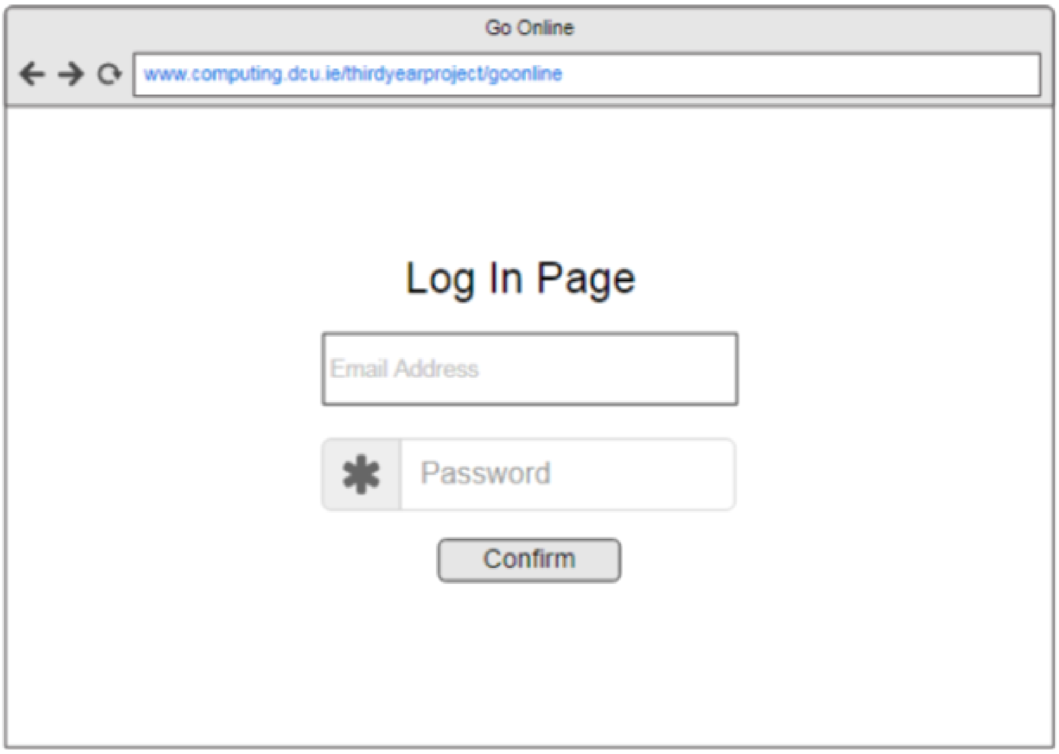
CLASS DIAGRAM

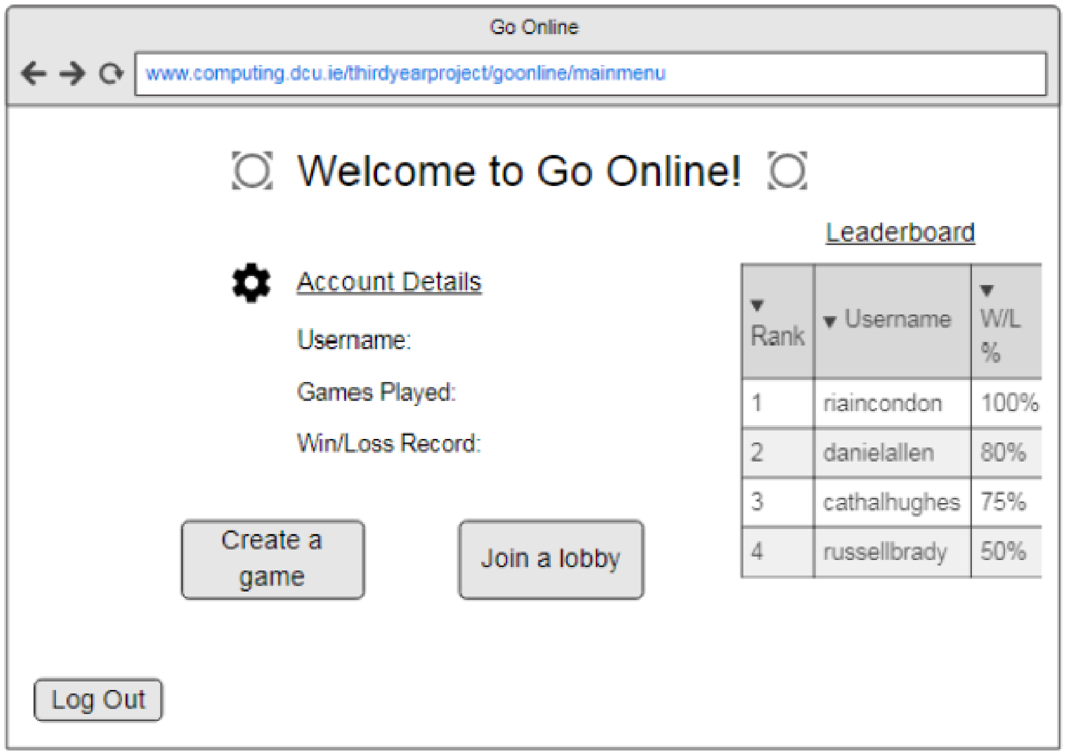


DOMAIN CLASS DIAGRAM

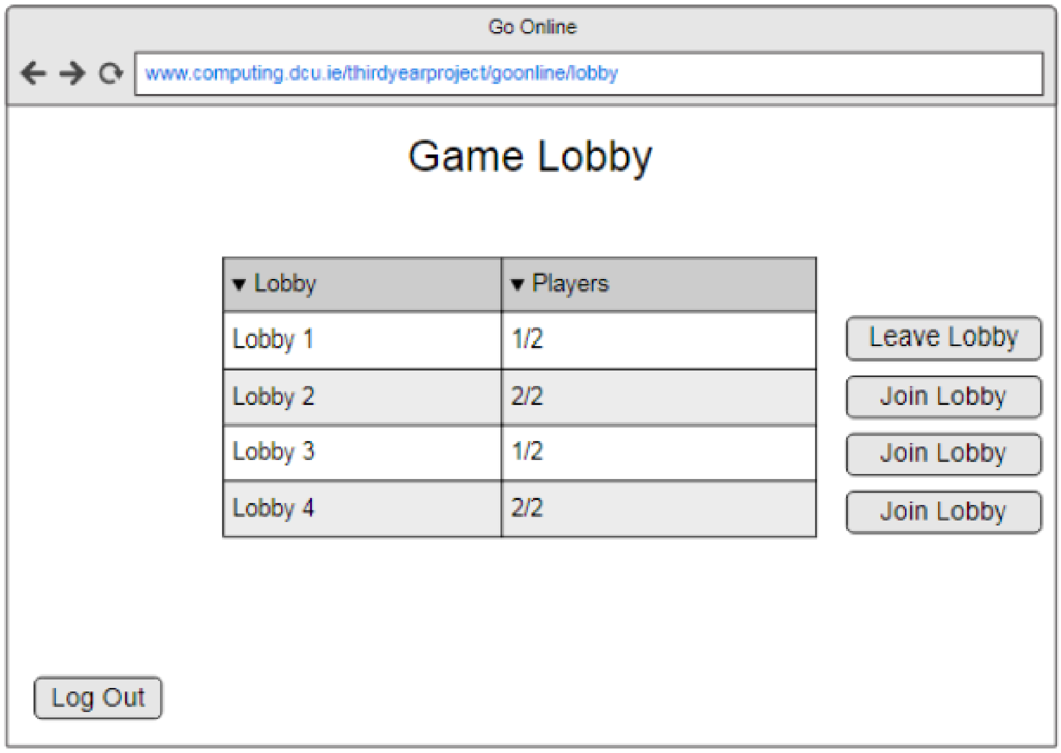


USER INTERFACE MOCK-UPS



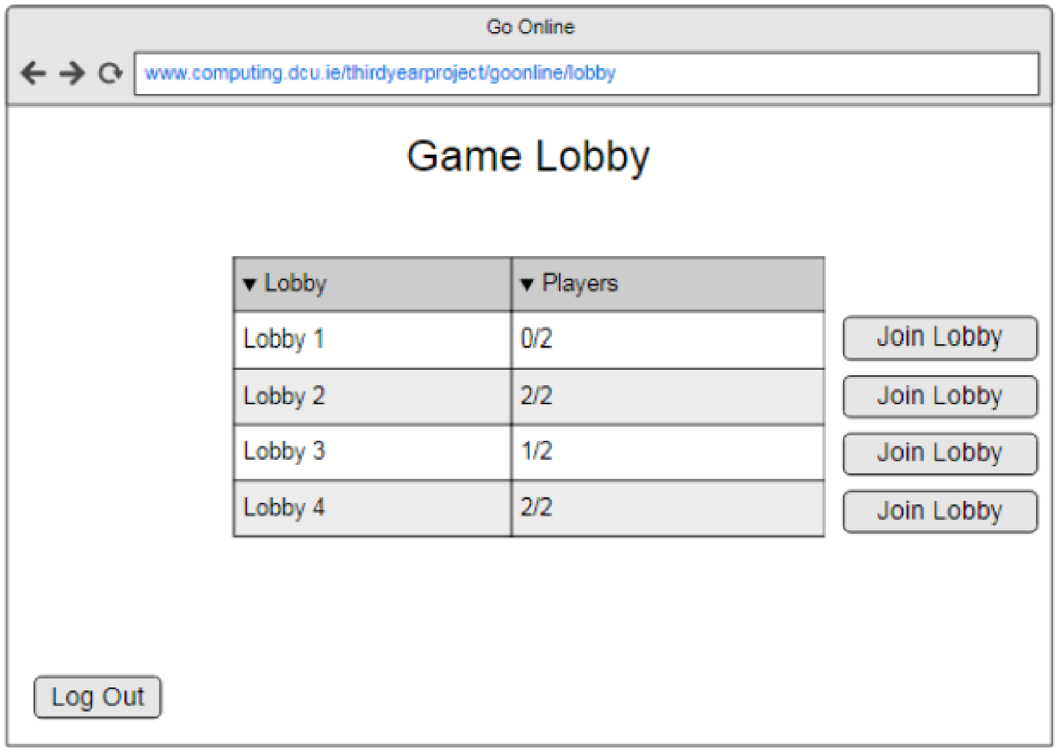


Rank	Username	W/L %
1	riaincondon	100%
2	danielallen	80%
3	cathalhughes	75%
4	russellbrady	50%



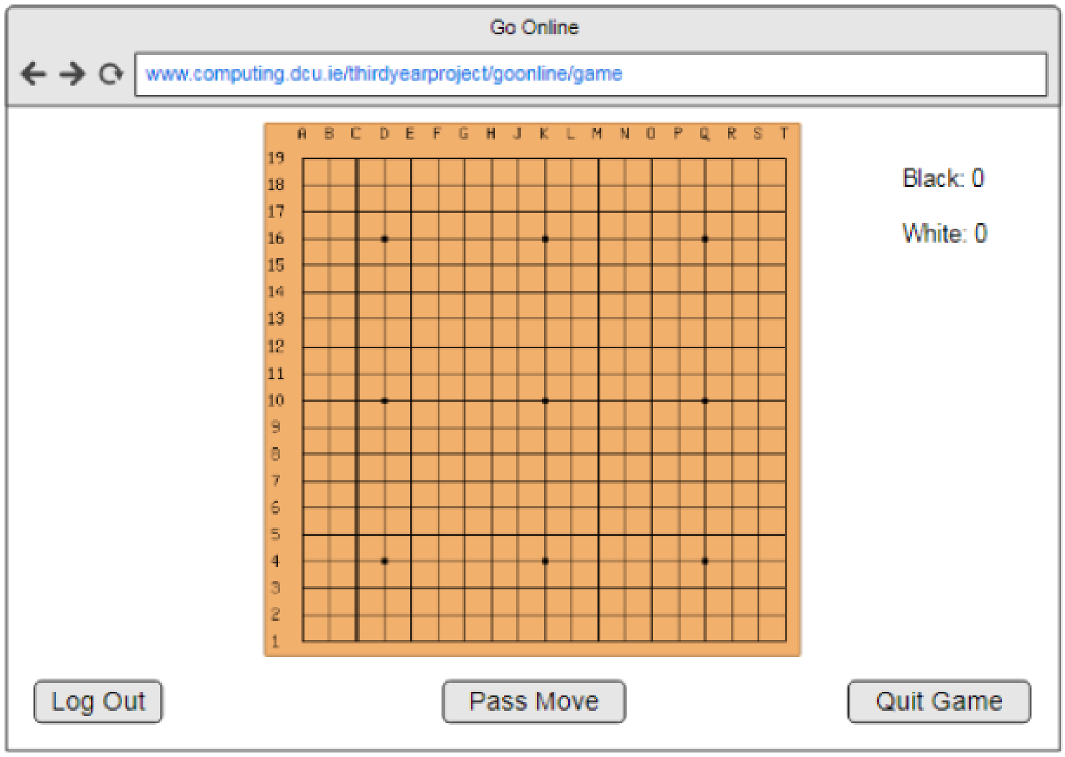
Game Lobby

▾ Lobby	▾ Players
Lobby 1	1/2
Lobby 2	2/2
Lobby 3	1/2
Lobby 4	2/2



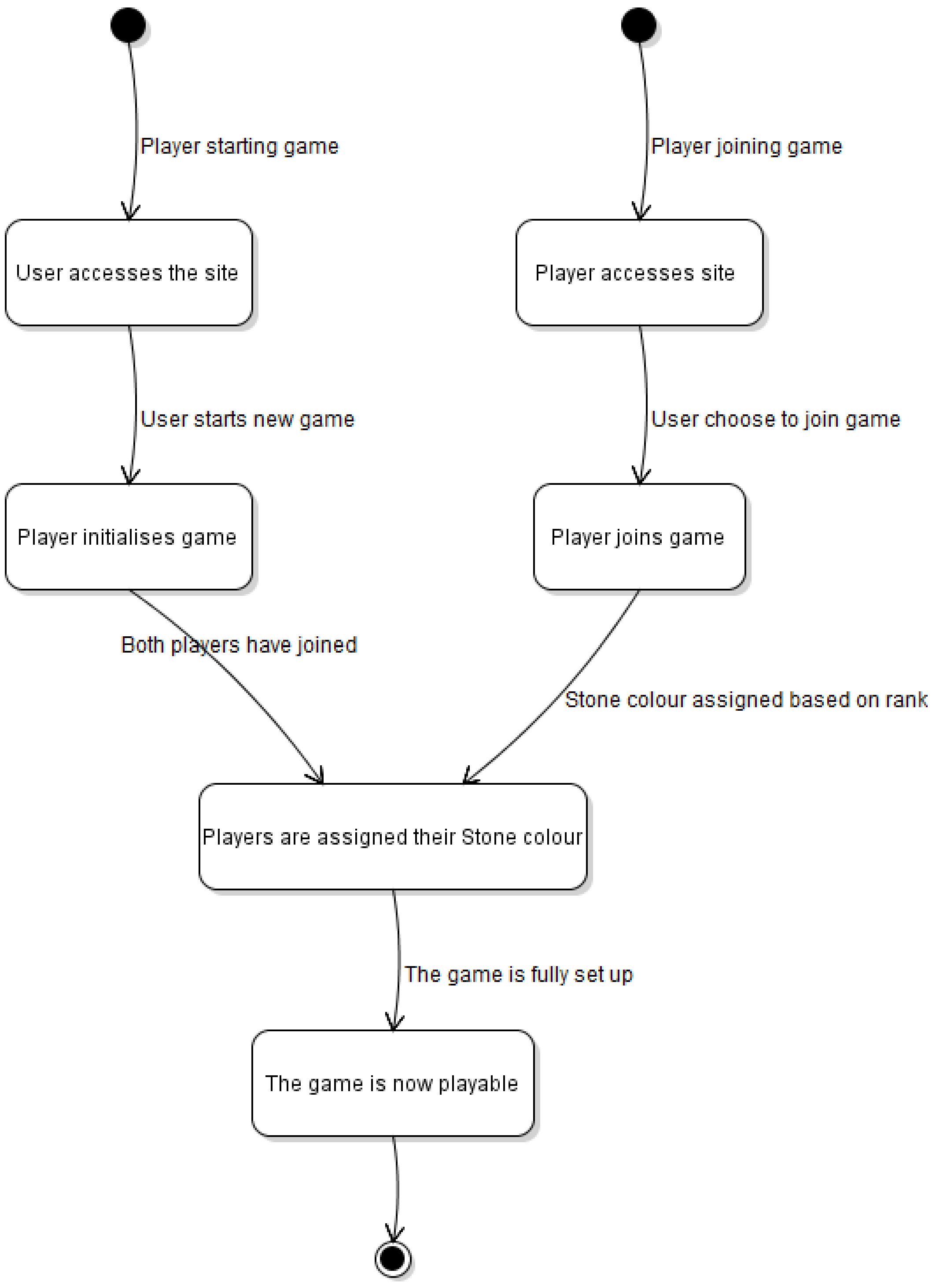
Game Lobby

▾ Lobby	▾ Players
Lobby 1	0/2
Lobby 2	2/2
Lobby 3	1/2
Lobby 4	2/2

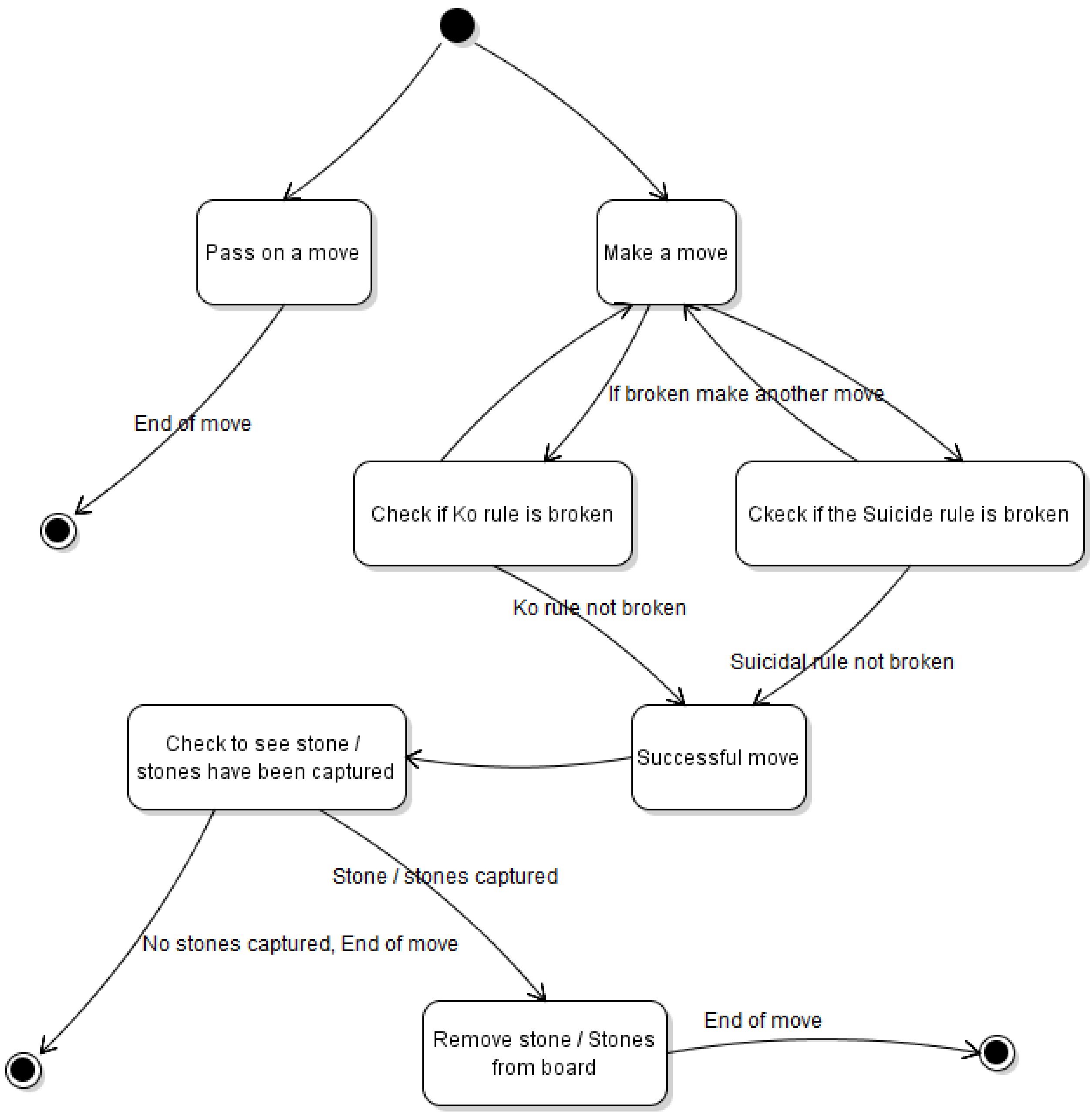

Go Online
www.computing.dcu.ie/thirdyearproject/goonline/game
A B C D E F G H J K L M N O P Q R S T
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
Black: 0
White: 0
Log Out
Pass Move
Quit Game

STATE DIAGRAMS

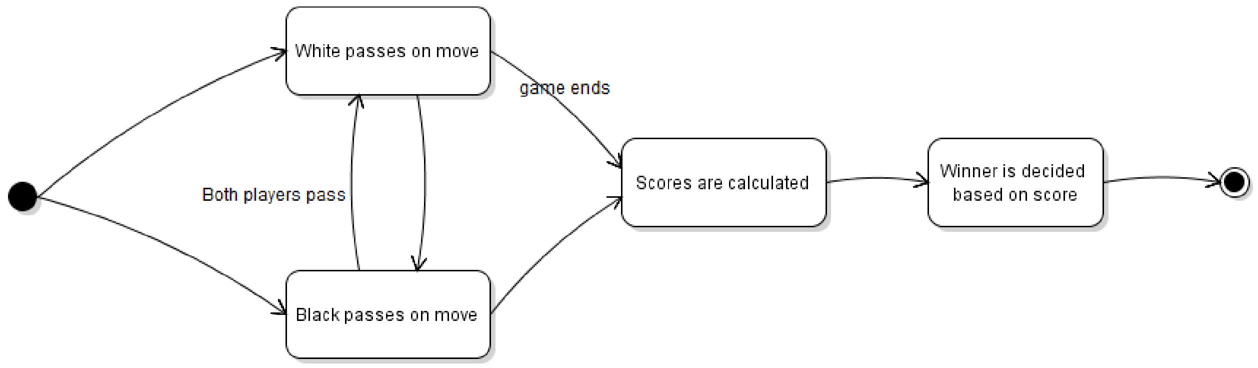
INITIALISE GAME



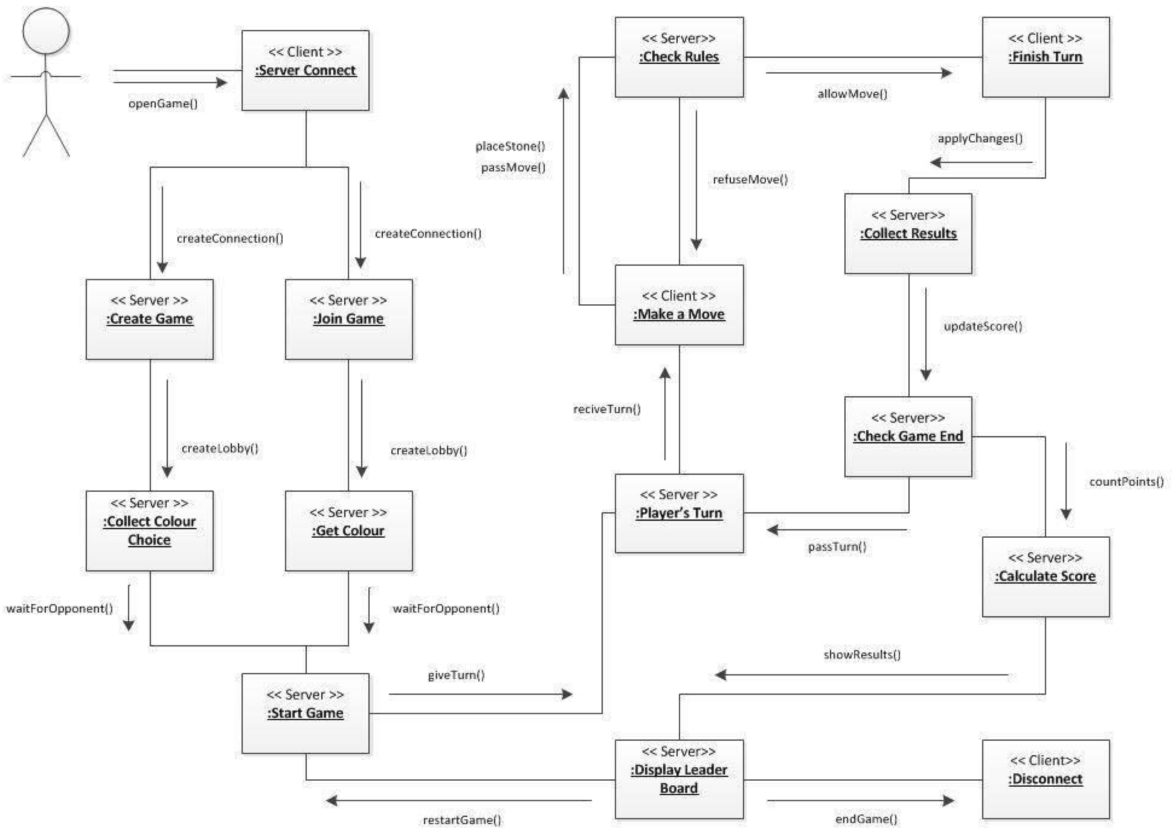
TAKE TURN


Pass on a move
Make a move
End of move
If broken make another move
Check if Ko rule is broken
Ckeck if the Suicide rule is broken
Ko rule not broken
Suicidal rule not broken
Check to see stone / stones have been captured
Successful move
Stone / stones captured
No stones captured, End of move
Remove stone / Stones from board
End of move

WIN GAME

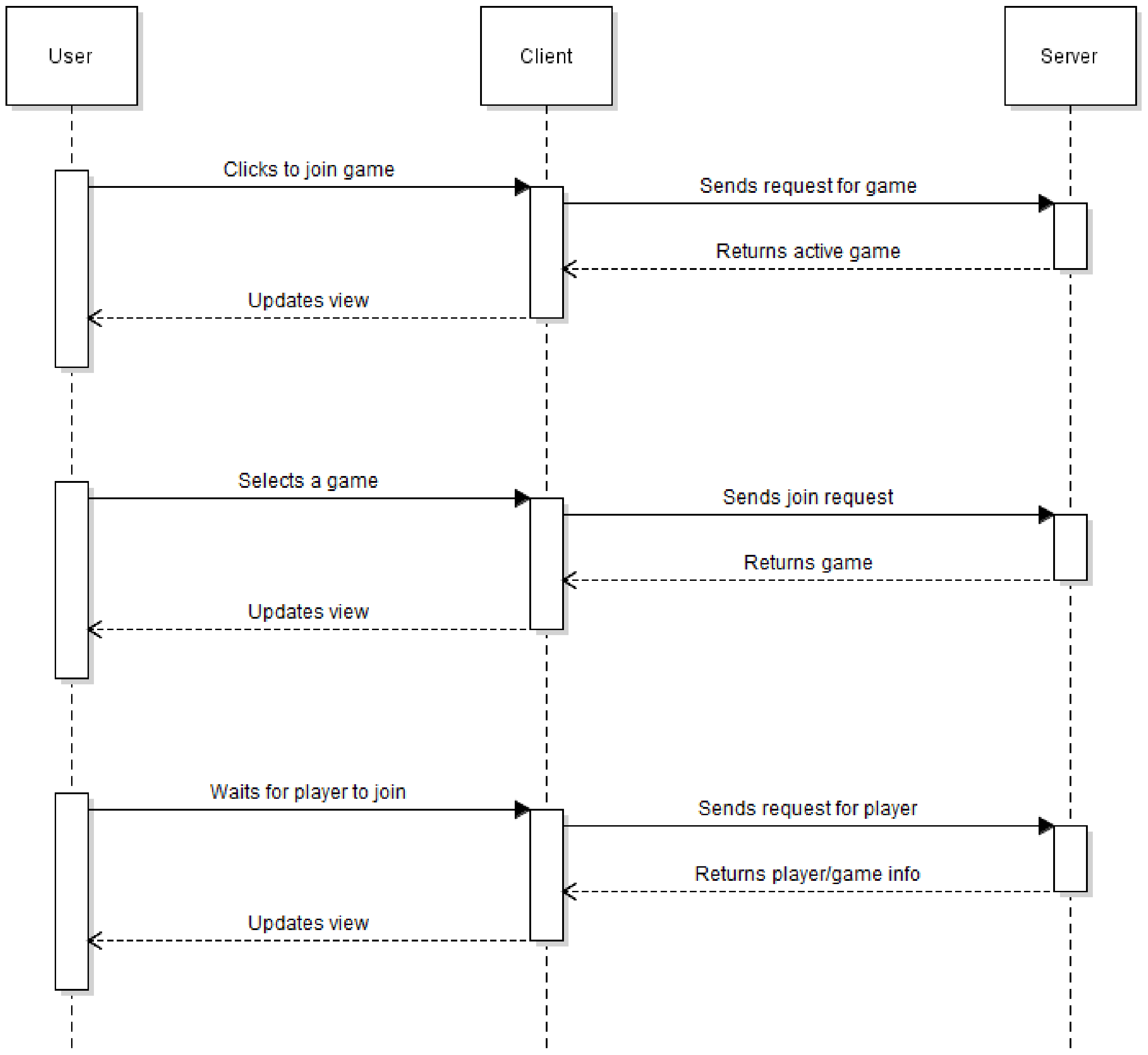


COLLABORATION DIAGRAM

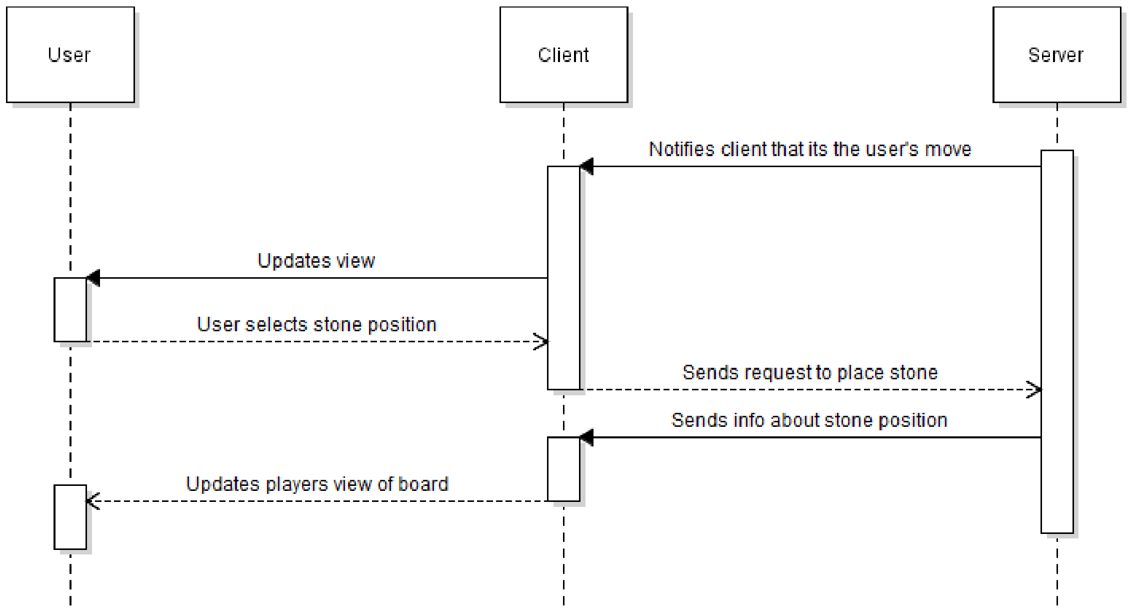


SEQUENCE DIAGRAMS

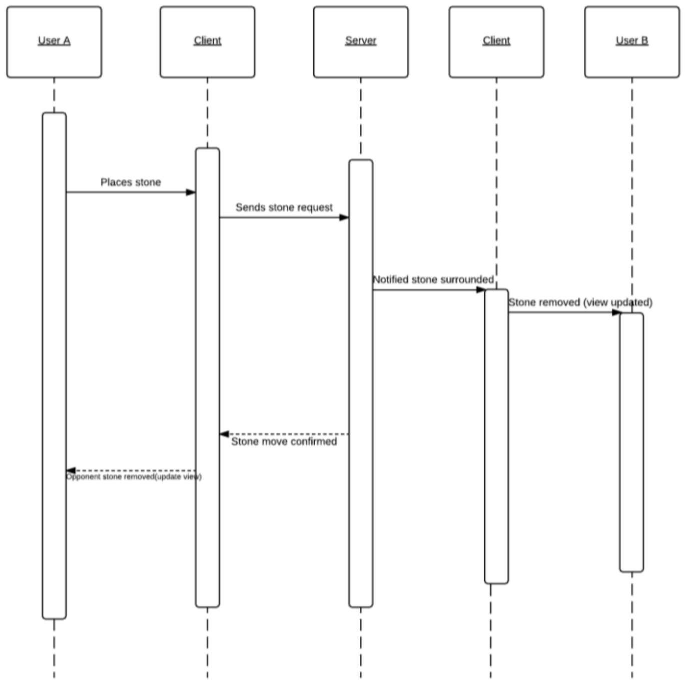
USER JOINS SERVER



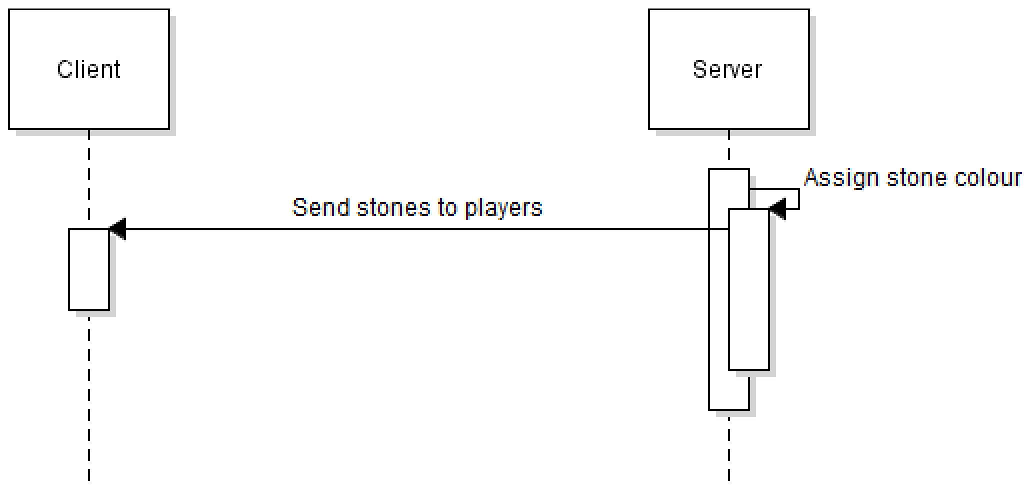
USER PLACES A STONE


User
Client
Server
Notifies client that its the user's move
Updates view
User selects stone position
Sends request to place stone
Sends info about stone position
Updates players view of board

USER CAPTURES STONE


User A
Client
Server
Client
User B
Places stone
Sends stone request
Notified stone surrounded
Stone removed (view updated)
Stone move confirmed
Opponent stone removed(update view)

GAME INITIALISES



CLASS SKELETONS

GAME CLASS

```
class Game
{
        Player player1
        Player player2
        Board board
        Int boardSize
        Bool gameInPlay
        Bool koRuleNotBroken
        Bool suicideRuleNotBroken

        Board() {
                //generates board for game
        }

        Player() {
                //Generates players and their attributes for a game
        }

        takeATurn() {
                //calls the players makeAMove() method
        }

        calculateTerritory() {
                //calculates territory for both players when the game is over
        }

        Utilities() {
                //Creates a utility object based on what the user wants to do, e.g. logoff or login
        }

}
```

CLIENT CLASS

```
class Client {

        Player() {

                //Generate Players for the game

        }

}
```

SERVER CLASS

```
class Server {
        Game() {
                //Creates a game (This will be the main)
        }
        Leaderboard() {
                //This will create an instance of the leaderboard
        }
}
```

UTILITY CLASS

```
class Utility {
        login() {
                //Logs user onto a server
        }
        Logoff() {
                //Logs user off the server
        }
        vewRules() {
                //allows user to view rules
        }
        viewLeaderboard() {
                //allows user to view leaderboard
        }
}
```

PLAYER CLASS

```
class Player {

        Int id

        Bool nextMove

        Stone stone

        String email

        Int gamesWon

        String password

        makeMove() {

                //allows a player to pick a position to place a stone

        }

        passMove() {

                //Allows a player to pass on a move

        }

        joinGame() {

                //allows a player to join a game

        }

        endGame() {

                //allows a player to quit a game

        }

}
```

STONE CLASS

```
class Stone {

        Int colour

        Bool isCaptured

        Int postion
```

```
    checkIfCaptured() {
        //Checks if the stone is captured by checking if its surrounded and returns a boolean
    }
}
```

BOARD CLASS

```
class Board {
    Int Size
    Int Colour
    updateBoard() {
    //Update the board after a stone has been placed or captured
    }
    makeMove() {
        //allows player to place a stone on the board
    }
}
```

LEADERBOARD CLASS

```
class Leaderboard {
    String [] topTen
    displayTopTen() {
        //displays the top ten Go players using our service
    }
    displayUsersWins() {
        //displays the current users wins
    }
}
```

GROUP MINUTES AND AGENDA

MEETING 6 - 24/10/2017

- All members present.
- Cathal took note of the minutes.
- We talked about how the first phase of the project went and how we could improve things.
- We also decided what work had to be done for the next sprint.
- We talked about the next phase of the project and what was expected of us.
- We all took a piece of the next phase and we will collaborate again earl next week.

MEETING 7 - 31/10/2017

- All members present
- Cathal took note of the minutes
- This meeting was short and was just to clarify any issues anyone had regarding the section they were doing for phase 2
- We specified all methods needed for the class skeletons

MEETING 8 - 7/11/2017

- All members present
- Cathal took note of the minutes
- We came together to try and put the doc together for phase 2 and critique each other's work.
- Anything that we thought was right was added to the doc whilst others were told to change or refine their work

MEETING 9 - 13/11/2017

- All members present
- Cathal took note of the minutes
- The final doc is put together after everything was approved by every member.
- We all proof read the doc and decided that it was ready for submission