|  |  |
| --- | --- |
|  | **2013 Project Battleship** |
|  | Keremcan Cakar  Lyubomir Atanasov  Yohandi Widjaja  Yingkai Yu |

|  |
| --- |
| **[design document ]** |

Table of Contents

[Functional Requirements 2](#_Toc359571737)

[Non-Functional Requirements 3](#_Toc359571738)

[1. Initial phase 3](#_Toc359571739)

[2. Placing Ship 3](#_Toc359571740)

[3. Starting to Battle 4](#_Toc359571741)

[4. Shooting and Ending the game 4](#_Toc359571742)

[Architecture Design 5](#_Toc359571743)

[Class Diagram 7](#_Toc359571744)

[7](#_Toc359571745)

[Sequence Diagrams 8](#_Toc359571746)

[1. Initial phase sequence diagram. 8](#_Toc359571747)

[2. Shooting and endgame sequence diagram. 8](#_Toc359571748)

[3. Start battle sequence diagram 10](#_Toc359571749)

[4. Placing ship sequence diagram 10](#_Toc359571750)

[User interface design 11](#_Toc359571751)

# Functional Requirements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number | Functional Requirements | Must | Should | Could | Would be better |
| 1 | Client connect to the server | X |  |  |  |
| 2 | A game can be started when two client connected | X |  |  |  |
| 3 | Client connect to the server with user name and IP address |  | X |  |  |
| 4 | A game can be started when two client connected and pressed “Ready” button |  | X |  |  |
| 5 | Client is able to put ship on the battle field | X |  |  |  |
| 6 | Client is able to rotate the ships, sets it horizontal or vertical |  | X |  |  |
| 7 | Program should check if two ships overlap | X |  |  |  |
| 8 | Player can shoot to the other player by clicking enemy’s battlefield grid | X |  |  |  |
| 9 | Player will receive message after shoot a ship (success or not) | X |  |  |  |
| 10 | The other player’s ship will be shown if a shoot successful |  | X |  |  |
| 11 | The player should see which place he has shoot | X |  |  |  |
| 12 | Player will receive a message if he wins or loses | X |  |  |  |
| 13 | Players can chat with each other |  | X |  |  |
| 14 | Players can login to a portal first and choose one of the players to be the opponent and invite him |  |  | X |  |

# 

# Non-Functional Requirements

1. The language we use to build the desired program should not affect the functionality.(c# is the language ,we are going to run the program )
2. It should be possible to run the program on Microsoft Windows operating system with .NET Framework installed.
3. There are five types of battle ships.
4. The users should connect to the same intranet

**Use cases**

### Initial phase

GOAL: To start the game

ACTOR: The users.

PRE-CONDITION: Two client programs (login window) are running.

MSS:

1. User-1 type in his name and click “find players”
2. System displays message “waiting for another player” in user-1’s screen
3. User-2 type in his name and click “find players”
4. System displays message “waiting for another player” in user-2’s screen
5. Two clients are connected, both login forms become battlefield forms and an information shows on both screens

EXTENSION:

5a. Both users close the programs, use case ended.

5b. One of the users close the program, another user still waits for other players.

### Placing Ship

GOAL: To place the ship

ACTOR: The users.

PRE-CONDITION: use case no. 1 has been completed successfully

MSS:

1. User click the picture of the ship that he/she wants to place
2. The user clicks on the grid.
3. The system adds the chosen ship on the grid (the middle of the ship is on the clicked cell, it may take several cell depending on the ship)

EXTENSION:

1a. user cancels the action, use case ended.

2a. the location that user chose is occupied by another ship, system display message and ship is not placed.

POST CONDITION: The ship is placed on the grid

### Starting to Battle

GOAL: To start the battle

ACTOR: The users.

PRE-CONDITION: Ships have already been placed

MSS:

1. User1 confirms that he has finished placing the ships.

2. System marks User1 as ready.

3. User2 confirms that he has finished placing the ships.

4. System marks User2 as ready.

5. System starts the battle after count down.

EXTENSION:

5a. if one user disconnects during count down he loses the game. Option for redraw is shown.

POST CONDITION: The battle has started.

### Shooting and Ending the game

GOAL: Shooting to enemy ships, checking if game has ended

ACTOR: The users

PRE-CONDITION: Game has started, It's user's turn.

MSS:

1. First User clicks on a cell on enemy area to shoot.
2. Server sends coordinates of this shoot to second user.
3. Second user checks if first one hit something.
4. Second user sends result to server.
5. Server sends result to first user.
6. Server says to second user that now it's his/her turn.

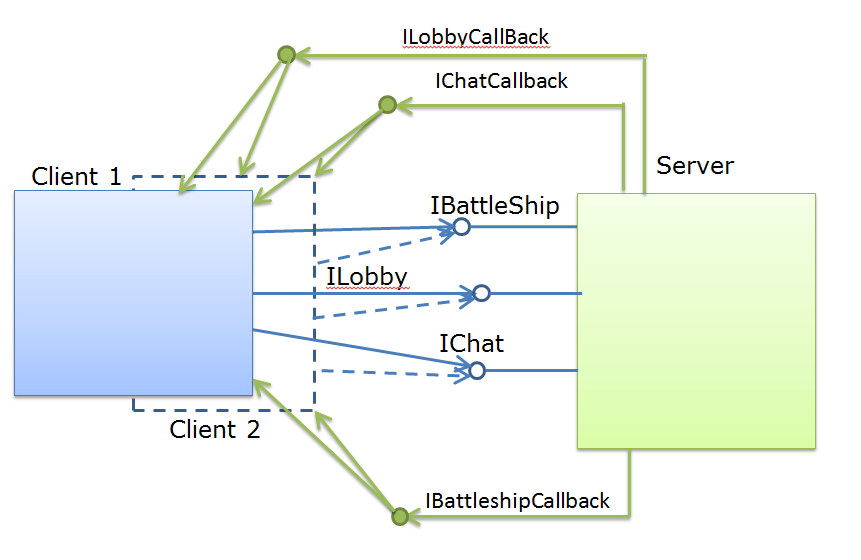
EXTENSION:

3a. If it's a hit and if all ship parts got destroyed, server says each player that game has ended.

3b. If it's a hit and if there are still at least one ship part which didn't got hit, it's first user's turn again

3c. If it's a miss it's second user's turn.

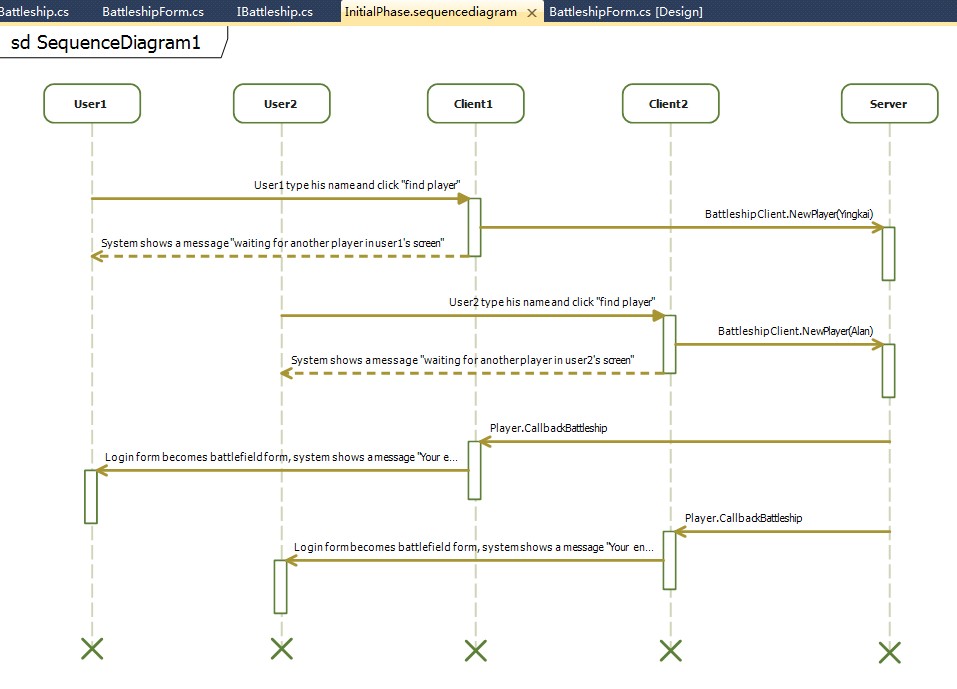
# Architecture Design



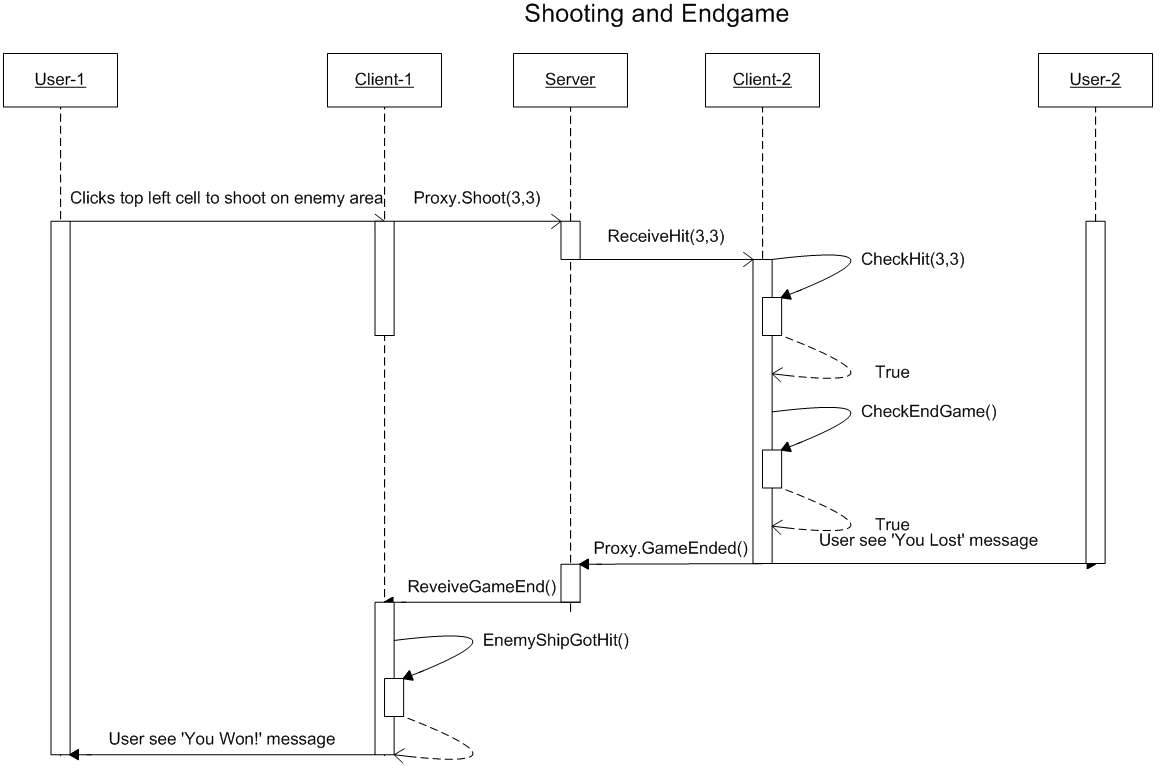
# Class DiagramD:\ClassDiagramClient.pngD:\ClassDiagramService.png

# Sequence Diagrams

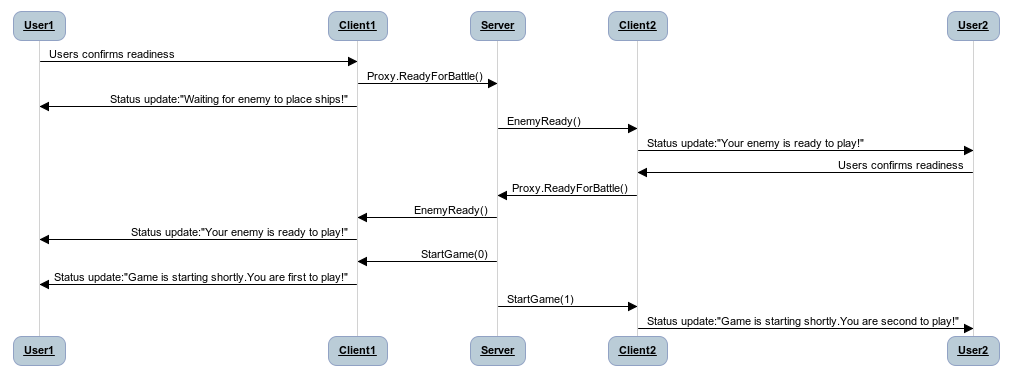
# 1. Initial phase sequence diagram.

****

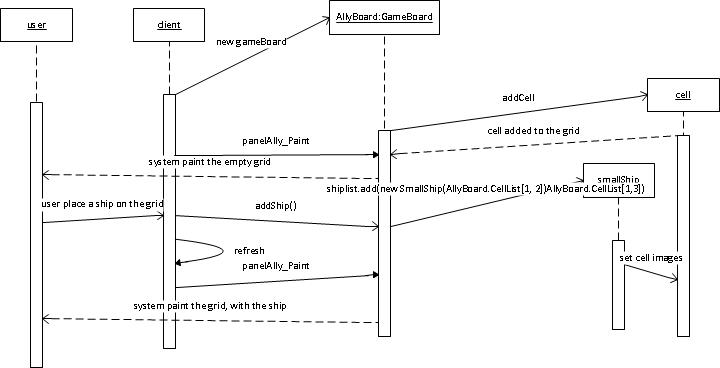
# 2. Shooting and endgame sequence diagram.



# 3. Start battle sequence diagram



# 4. Placing ship sequence diagram



# User interface design

