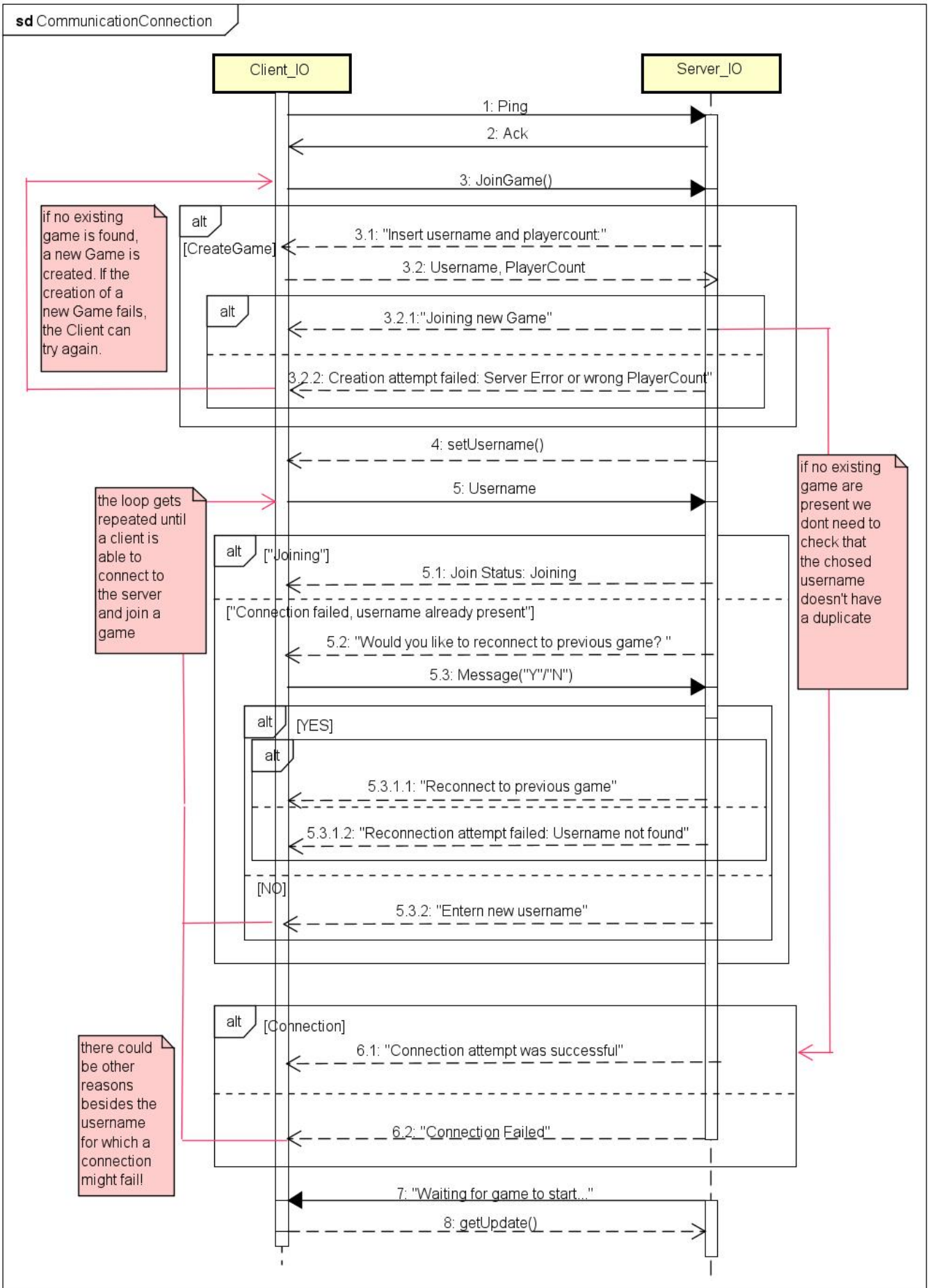# 1. Managing multiple games

1.1 A client Joining a Game

In particular when a game starts and the player is requested to submit a username, if the chosen username already exists, they are asked if they want to change their name or they want to reconnect to an old/existing game. In case they type Y (yes) there's an attempt to said reconnection.

sd CommunicationConnection

Client_IO   Server_IO

1: Ping

2: Ack

3: JoinGame()

alt

[CreateGame]

3.1: "Insert username and playercount:"

3.2: Username, PlayerCount

alt

3.2.1: "Joining new Game"

3.2.2: Creation attempt failed: Server Error or wrong PlayerCount"

if no existing game is found, a new Game is created. If the creation of a new Game fails, the Client can try again.

4: setUsername()

5: Username

alt ["Joining"]

5.1: Join Status: Joining

["Connection failed, username already present"]

5.2: "Would you like to reconnect to previous game? "

5.3: Message("Y"/"N")

the loop gets repeated until a client is able to connect to the server and join a game

alt [YES]

alt

5.3.1.1: "Reconnect to previous game"

5.3.1.2: "Reconnection attempt failed: Username not found"

[NO]

5.3.2: "Entern new username"

alt [Connection]

6.1: "Connection attempt was successful"

6.2: "Connection Failed"

there could be other reasons besides the username for which a connection might fail!

if no existing game are present we dont need to check that the chosed username doesn't have a duplicate

7: "Waiting for game to start..."

8: getUpdate()

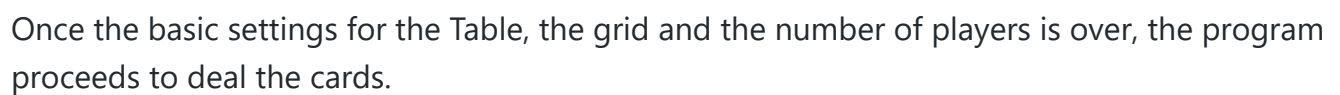## 1.2 Server waiting for Players to Join

The Game starts when the number of players submitted by the Client who started the game is reached.
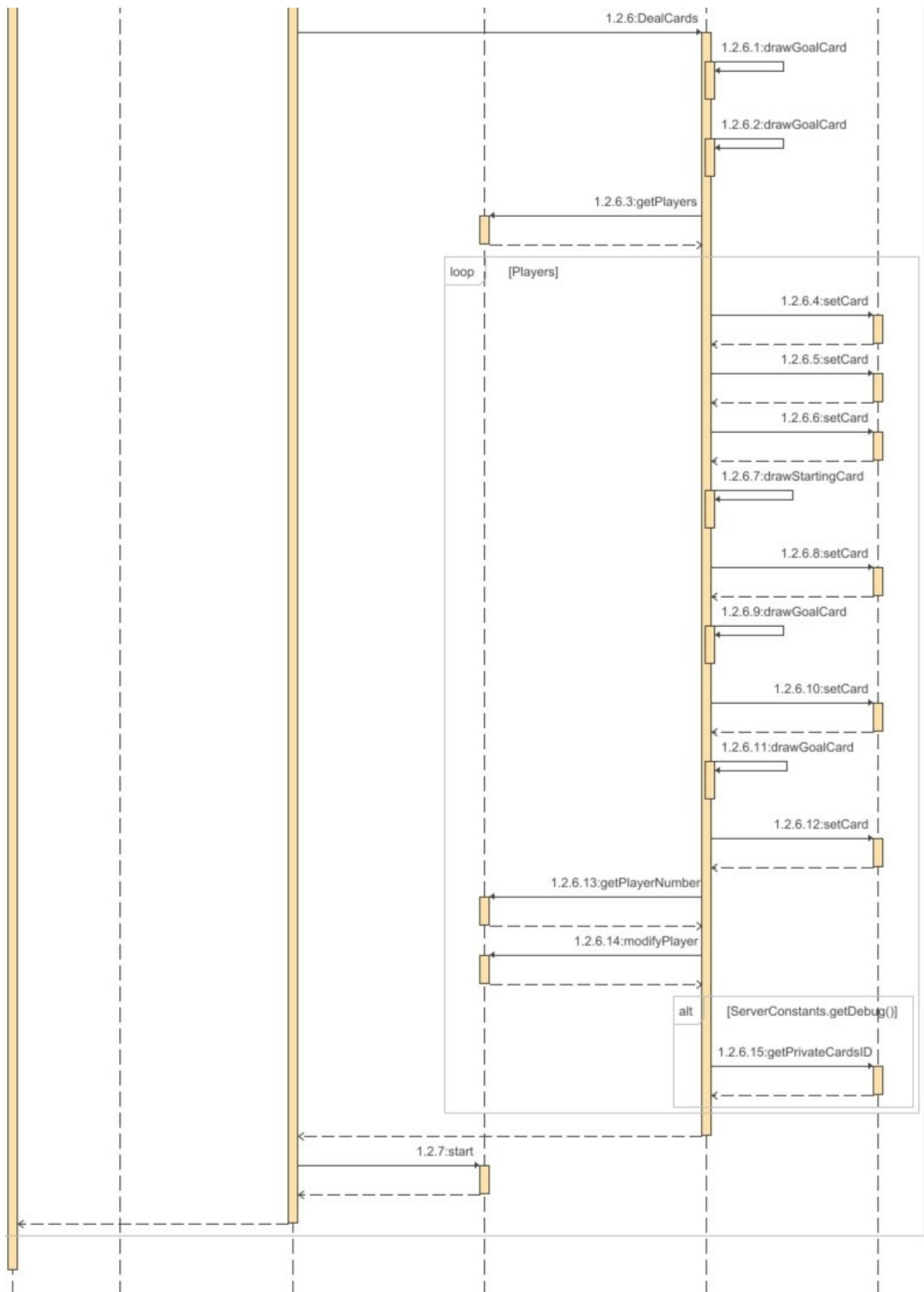

2. Single Game Setup

Starting operations such as setting the space where the players can place the cards and draw the common cards at the beginning (the ones that players are not able to choose).

For a better understanding we are going to show only the Server's point of view, as in this phase he interacts with the Client only for the choice of his personal Goal.

Main | GameServer | MultipleGameManager | Game | Table | Player

main

alt [test]

1.1:<<create>>

1.2:addGame

1.2.1:<<create>>

1.2.1.1:<<create>>

1.2.2:startGameLoop

1.2.2.1:<<create>>

1.2.3:getRelatedTable

1.2.4:AutoFillSpaces

alt [PublicSpacesID[0]==0 && PublicSpacesID[1]==0]

1.2.4.1:end

1.2.4.1.1:end

1.2.4.1.1.1:isGameStarted

1.2.4.1.1.2:getPlayers

1.2.4.1.1.3:getPlayerCount

1.2.5:getRelatedTable

Once the basic settings for the Table, the grid and the number of players is over, the program proceeds to deal the cards.

1.2.6:DealCards

1.2.6.1:drawGoalCard

1.2.6.2:drawGoalCard

1.2.6.3:getPlayers

loop [Players]

1.2.6.4:setCard

1.2.6.5:setCard

1.2.6.6:setCard

1.2.6.7:drawStartingCard

1.2.6.8:setCard

1.2.6.9:drawGoalCard

1.2.6.10:setCard

1.2.6.11:drawGoalCard

1.2.6.12:setCard

1.2.6.13:getPlayerNumber

1.2.6.14:modifyPlayer

alt [ServerConstants.getDebug()]

1.2.6.15:getPrivateCardsID

1.2.7:start

2.1 Personal Goals

The Client has to choose between the two GoalCards that the Game provides to them.
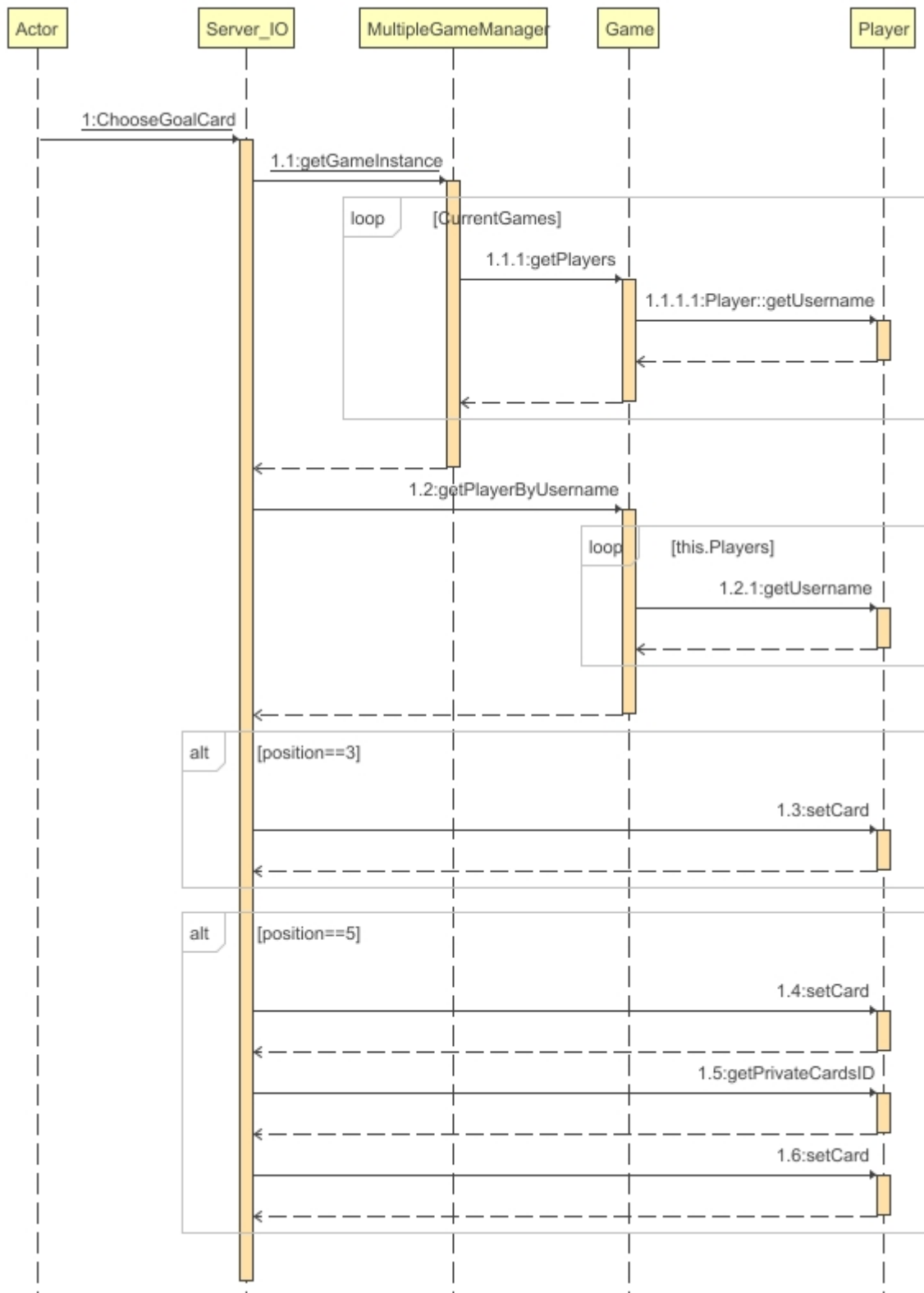

The server randomly picks two GoalCards for each player, and places them in "position 3" and "position 5". Two different scenarios may play out:

- The player DOESN'T choose a card before the end of the Game

    - the player is automatically assigned the card in position 3.

- The Player DOES choose a Goal Card

    - If it's the card in position 3, the card in position 5 gets "consumed";

    - If it's the card in position 5, the card gets relocated to position 3.

    Both ways, the assigned or chosen card will be found at position 3.

3. Change Turns

As soon as the Player has completed the last action on his turn, which is drawing a PlayableCard, in the Draw method itself, the Server calls for "changePlayerTurn()", which changes the number associated with the current player with the next player in order and the boolean "isTurn()" changes from 1 to 0.
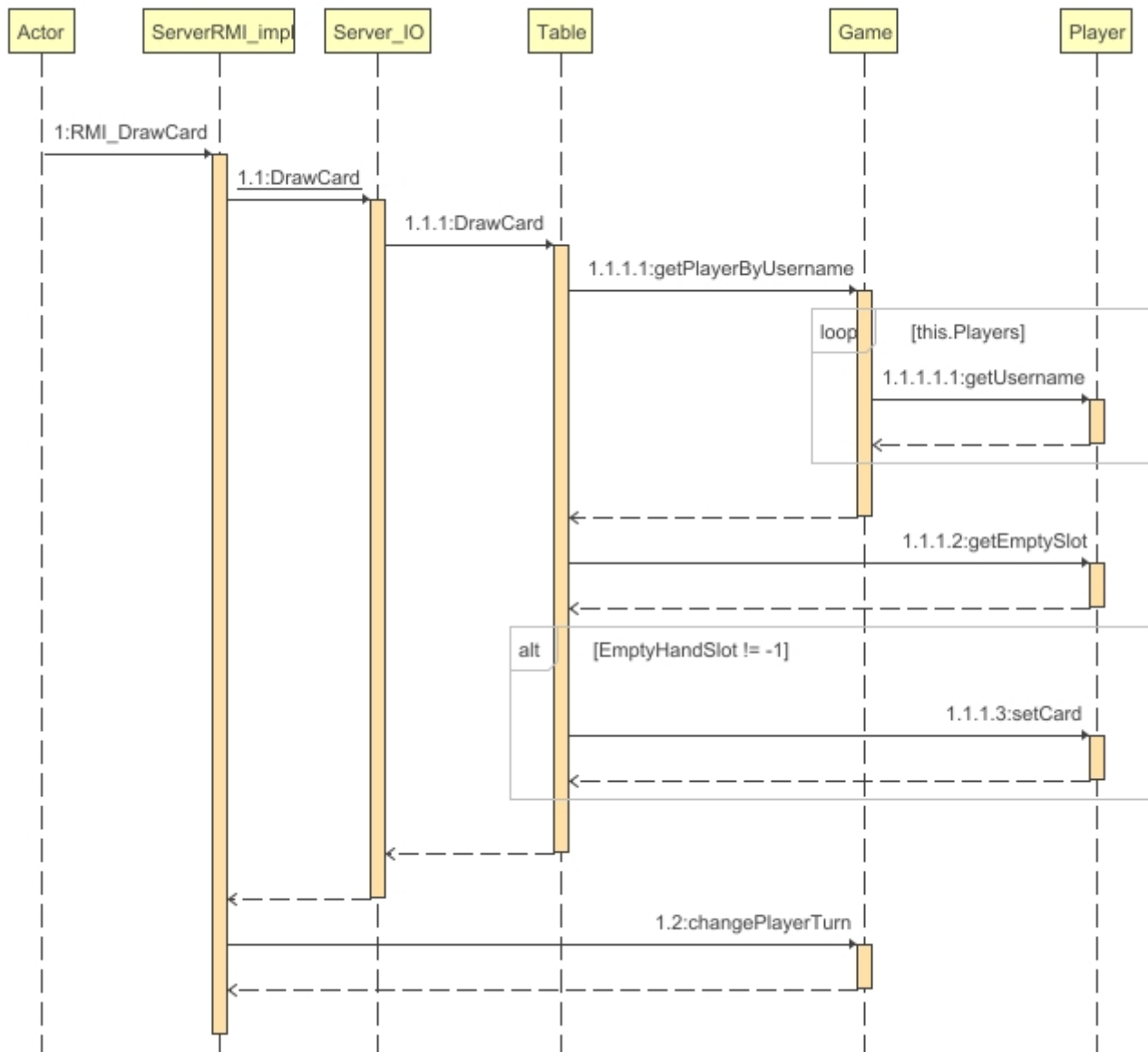
3.1 Player Turns Update

The turns are going to be shown primarily from the Client`s point of you; as each player completes their set of action for a turn, an update package is sent to the server with information such as: what card was placed - and the exact position on the board, what card has been drawn.

sd Players Turn
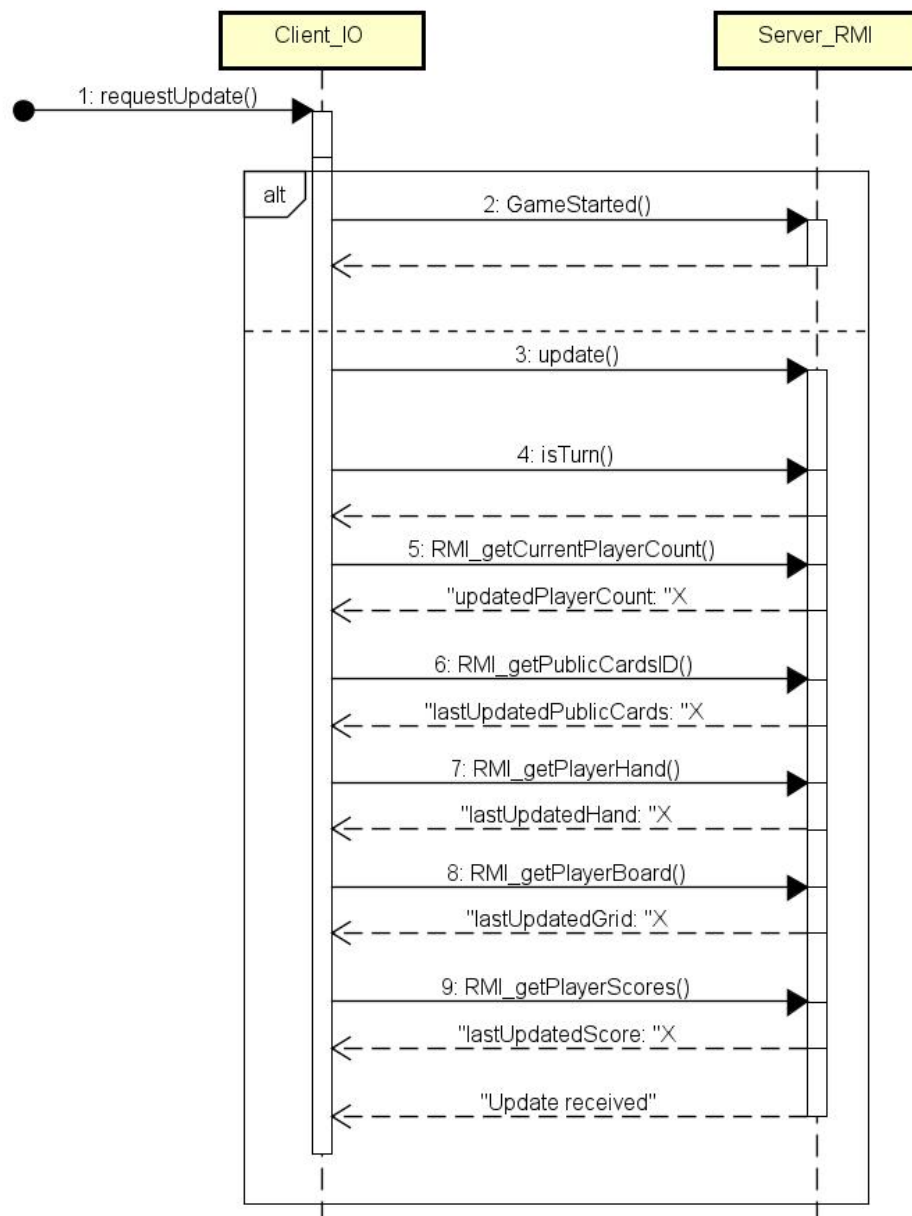
```
                    Client_IO                          Server_RMI

      1: requestUpdate()
  ●───────────────────▶│
                       │
                       │┌──────────────────────────────────────┐
                   alt ││                                        │          ┌──────────────────┐
                       ││        2: GameStarted()                │          │ if the Game is   │
                       ││───────────────────────────────▶│      │          │ yet to start or  │
                       ││◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │      │          │ a players turn   │
                       ││                                        │          │ just took        │
                       ││─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─  │          │ place, the       │
                       ││        3: update()                     │          │ player is        │
                       ││───────────────────────────────▶│      │          │ prevented        │
                       ││                                        │          │ from             │
                       ││        4: isTurn()                     │          │ proceeding       │
                       ││───────────────────────────────▶│      │          │ with any         │
                       ││◀ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │      │          │ action           │
                       ││   5: RMI_getCurrentPlayerCount()       │          └──────────────────┘
                       ││───────────────────────────────▶│      │
                       ││◀ ─ "updatedPlayerCount: "X ─ ─ ─│      │
                       ││      6: RMI_getPublicCardsID()         │
                       ││───────────────────────────────▶│      │
                       ││◀ ─ "lastUpdatedPublicCards: "X ─│      │
                       ││       7: RMI_getPlayerHand()           │
                       ││───────────────────────────────▶│      │
                       ││◀ ─ "lastUpdatedHand: "X ─ ─ ─ ─ │      │
                       ││       8: RMI_getPlayerBoard()          │
                       ││───────────────────────────────▶│      │
                       ││◀ ─ "lastUpdatedGrid: "X ─ ─ ─ ─ │      │
                       ││       9: RMI_getPlayerScores()         │
                       ││───────────────────────────────▶│      │
                       ││◀ ─ "lastUpdatedScore: "X ─ ─ ─ ─│      │
                       ││◀ ─ "Update received" ─ ─ ─ ─ ─ ─│      │
                       │└──────────────────────────────────────┘
```