Design Document

Shihan Wang, Zhiwei Zhuge, Ali Hamzehei, George Vasileiadis



Contents

[Introduction: 2](#_Toc404803224)

[Architecture diagram: 2](#_Toc404803225)

[3](#_Toc404803226)

[The interfaces: 3](#_Toc404803227)

[IPortal: 3](#_Toc404803228)

[IGame: 3](#_Toc404803229)

[IGameCallBack: 3](#_Toc404803230)

[Service side: 3](#_Toc404803231)

[String CheckNumber([]int) 4](#_Toc404803232)

[Bool InvitePlayer(string player, string opponent) 4](#_Toc404803233)

[Void StartGame() 4](#_Toc404803234)

[List<> GetLogedPlayers() 4](#_Toc404803235)

[Bool LoginPlayer(string username, string password) 4](#_Toc404803236)

[Bool RegisterPlayer(string username, string password) 4](#_Toc404803237)

[Client side: 4](#_Toc404803238)

[List<> AvailablePlayers() 4](#_Toc404803239)

[String ChatMessage(string player, string message) 4](#_Toc404803240)

[Int GuessCounter() 4](#_Toc404803241)

[String Winner() 4](#_Toc404803242)

[Class Diagram: 4](#_Toc404803243)

[Sequence Diagrams: 5](#_Toc404803244)

[Login: 5](#_Toc404803245)

[Register: 6](#_Toc404803246)

[Start Game: 6](#_Toc404803247)

# 

# Introduction:

This is the design document for our bulls and cows project. In this document you will find the architecture and the class diagram of the bulls and cows game, Also you will find a short description of the methods we will be using.

# Architecture diagram:

Here you can see the architecture diagram of the bulls and cows game. The clients communicate with the two interfaces as shown below. And there is one Callback interface IGameCallBack.

# 

# The interfaces:

## IPortal:

The Iportal interface of the bulls and cows game is the interface that has the most methods, thus has a lot of important functionalities. The Iportal is in charge of registering users and signing them in so they can reach the Lobby. In the Lobby they need to be able to invite other players and the Iportal has a job for setting the opponents. Also, updating scores is part of the Iportal. Of course a user needs to be able to leave the Lobby so there is also a log out function to do that.

## IGame:

The IGame interface of the bulls and cows game contains three methods. It has a method so that users can send each other a message in the form of a chat. There is a method for count a guess, so that users could know how many guess is left, and this method calls a callback so that the boards are updated. Of course there is also a method for quitting a game, making it possible for users to leave mid game.

## IGameCallBack:

# Service side:

## String CheckNumber([]int)

This method checks the number the player has entered and returns a string indicating how many digits are correct or at the wrong position.

## Bool InvitePlayer(string player, string opponent)

This method is called when a player invites an opponent to start a match. It returns true if the opponent accepts the game or false otherwise.

## Void StartGame()

When both players agree to start a match this method is called.

## List<> GetLogedPlayers()

This method returns a list with all the players that are currently logged in.

## Bool LoginPlayer(string username, string password)

It is called when a player logs into the lobby.

## Bool RegisterPlayer(string username, string password)

This method saves the player’s username and password to the database.

# Client side:

## List<> AvailablePlayers()

It returns a list of players that are free to start a game.

## String ChatMessage(string player, string message)

This method returns which player send what message in the lobby chat.

## Int GuessCounter()

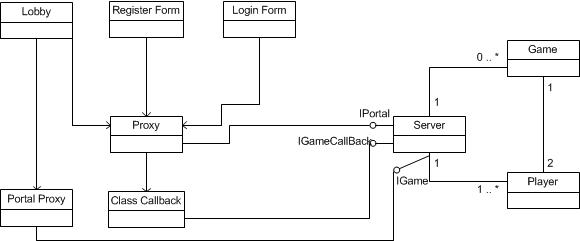
This method returns how many times a player has guessed the number in a game.

## String Winner()

This method returns the winner of the game.

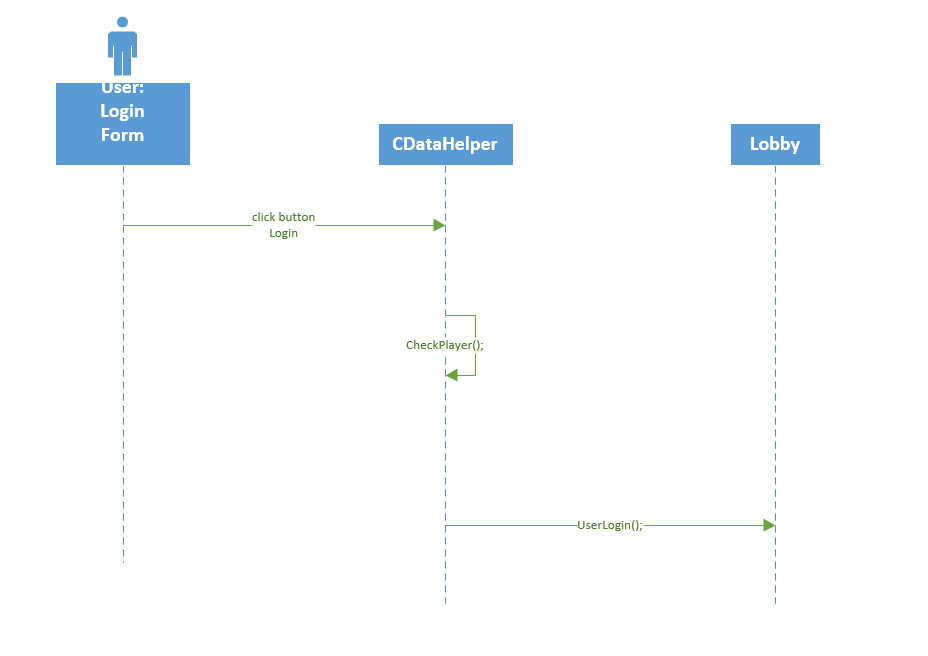
# Class Diagram:

The bulls and cows game is set up in a way that the forms go through a different class proxy, and those connect to the interfaces (Iportal, IGame). There is only one callback interface, and that is the IGameCallback. This callback interface is implemented by the classCallback. On the server side we have a Server that implements the two interfaces: IPortal and IGame.

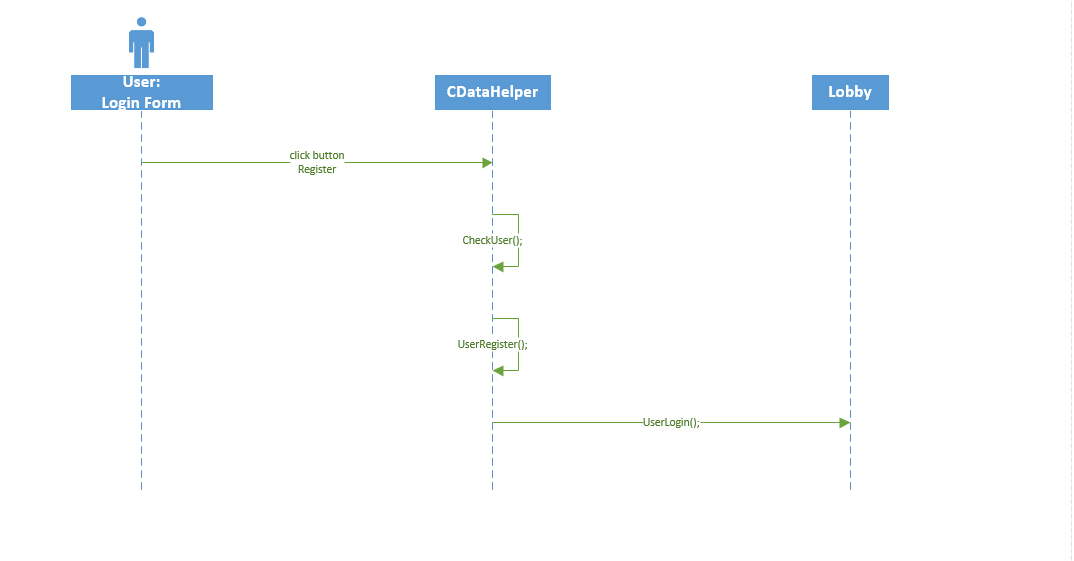


# Sequence Diagrams:

## Login:



## Register:



## Start Game:

