Middleware: Ludo game

Group 4

Jean Chan  
Loic Motheu  
Hristo Tarnev

Design document

Contents

[Architecture diagram 3](#_Toc418247198)

[Description of Interfaces 4](#_Toc418247200)

[Methods 4](#_Toc418247201)

[ILudo 4](#_Toc418247202)

[IChat 4](#_Toc418247203)

[Callbacks/Events 4](#_Toc418247204)

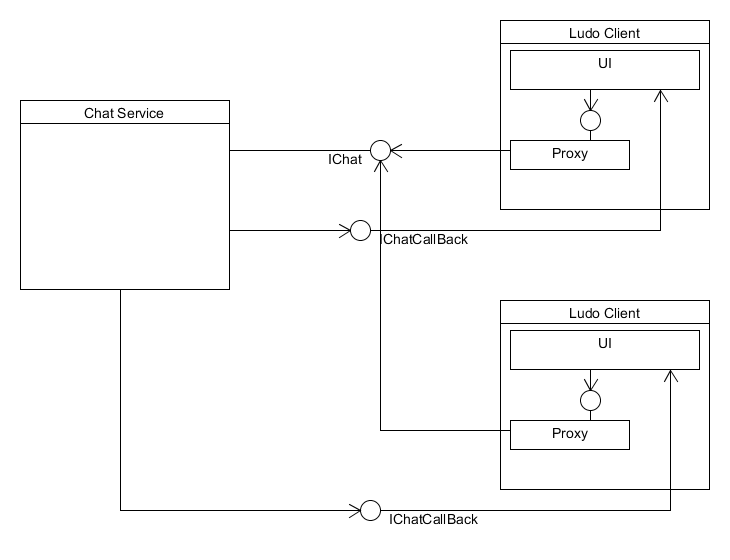
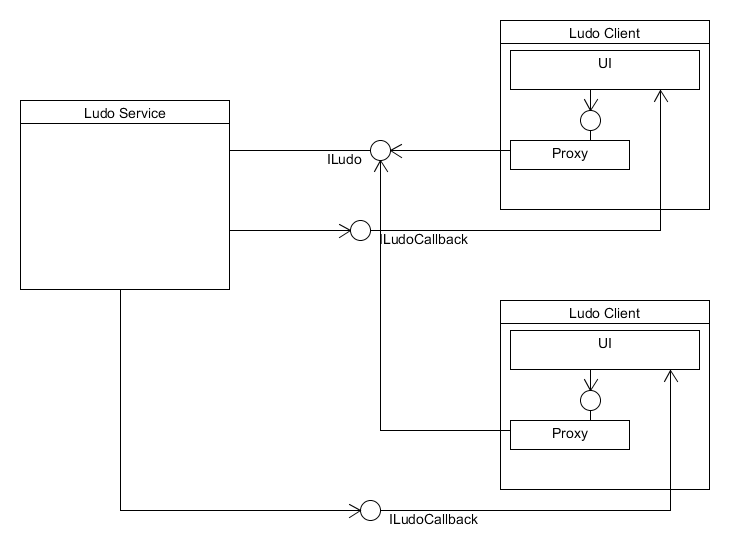
[ILudoCallBack 4](#_Toc418247205)

[IChatCallback: 4](#_Toc418247206)

[Class diagram 5](#_Toc418247207)

[Sequence diagrams 6](#_Toc418247208)

# Architecture diagram



# Description of Interfaces

## Methods

### ILudo

*This interface handles all of the player actions during a game.*

Void InitializeBoard() that places all tokens into base and lets player 1 start.

Void RollDice(): Generates a random number between 1 and 6.

Void MoveToken(Token token): Moves a desired token x amount of squares where x is the random generated dice roll.

Void PlaceToken(Token): When a user rolls a 6, he may place a token on the starting square.

Bool PlayerReady(Player player): Marks a player as ready to play

Void ChooseColor(Player player): Allows a player to choose a color.

Bool RegisterPlayer(string email, string password): Adds a player to the list of all players.

Bool CheckUserInfo(string email, string password): Checks if player exists in the system.

### IChat

*Handles the chatting between players*

Void SendMessage(String message, List<Player> players): Can be both public and private, by sending a string message to all players available in the current game, or to a specific player.

## Callbacks/Events

### ILudoCallBack

*Handles the notification of the changes within the game to other players.*

Void PlayerTurn(Player player): Indicates who’s player turn it is.

Void TokenMoved(Token token): When a player has moved a token, now all the other players get notified of the position of that token.

Void StartGame(List<Player> players): Needs a list of players who are ready to be able to start the game.

Void TokenOut(Token token): Signals all players, when a token has exited the board(either by force, or by placing it at the finish square).

Void DiceRolled(int number): Indicates the number of the dice when rolled.

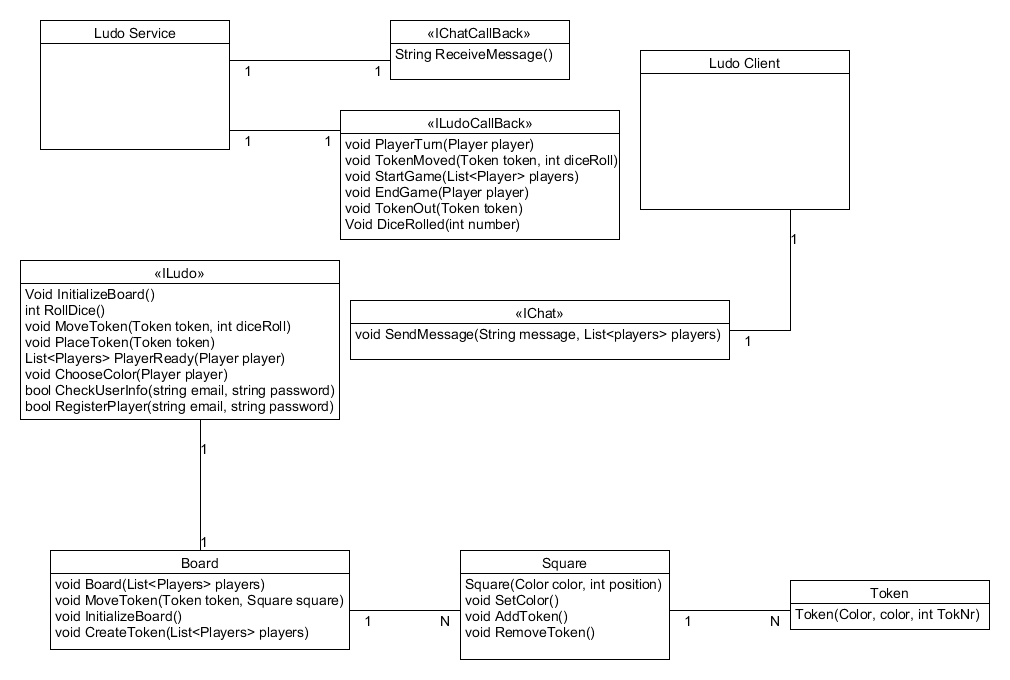
### IChatCallback:

*Handles the notifying the other players of the chat messages.*

Void SendMessage(string message): Returns the message written by other player/s.

# Class diagram

Note: We were unsure whether we needed to have a class diagram for the overall project, of for just the WCF components



# Sequence diagrams

