|  |  |  |
| --- | --- | --- |
| DESIGN DOCUMENT | January 1  2015 | |
| LUDO WCF APPLICATION | | GROUP 6: Todor Tsekov Ivana Raykova Zair Thiel |



Table of Contents

[1. Introduction 3](#_Toc418252136)

[2. Architecture Design 4](#_Toc418252137)

[3. Description of Interfaces 6](#_Toc418252138)

[4. Class Diagram for client 7](#_Toc418252139)

[5. Class diagram for Service 8](#_Toc418252140)

[6. Sequence Diagrams for MUST use-cases 9](#_Toc418252141)

[6.1 Roll Die 9](#_Toc418252142)

[6.2 Place Token 10](#_Toc418252143)

[6.3 In Game Chat 10](#_Toc418252144)

[6.4 Move Piece 11](#_Toc418252145)

[6.5 Remove Piece 12](#_Toc418252146)

[6.6 Choose Color 13](#_Toc418252147)

[6.7 Create Game 14](#_Toc418252148)

[6.8 Start Game 15](#_Toc418252149)

[6.9 Invite Player 16](#_Toc418252151)

# Introduction

A software design document (SDD) is a written description of a software product, that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. An SDD accompanies a class diagram with pointers to detailed feature specifications of smaller pieces of the design, sequence diagrams and explanation of the classes, methods, etc. in the project. Practically, a design document is required to coordinate a large team under a single vision. A design document needs to be a stable reference, outlining all parts of the software and how they will work. The document is commanded to give a fairly complete description, while maintaining a high-level view of the software.

The goal of this project is to build a Ludo Application. Ludo is a board game for 2-4 players. Every player is assigned 4 pieces, a dice and one of 4 possible starting areas. The goal is to get all your pieces around the game board to the finish line. Players take turns throwing the dice, which decides how far a piece can be moved. If you get a 6 you get an extra turn or you may enter another staged token to its starting square. When a player lands a piece on top on an opponent’s piece, the opponent’s piece is sent to the starting area. The general strategy is to make sure that your own pieces are not send to your starting area, and at the same time try to send the pieces of your opponents to their starting area. The rolls of a die control the swiftness of the tokens, and entry to the finishing square requires a precise roll from the player. The first to bring all their tokens to the finish wins the game.

Now you will see our application in more detailed and more structured way. With explanation of every class, so it can be clear what every method is doing and how our classes (structure) are connected.

# Architecture Design

LudoServer(app)

LudoClient(app)

IGame

Client 1

ILogin

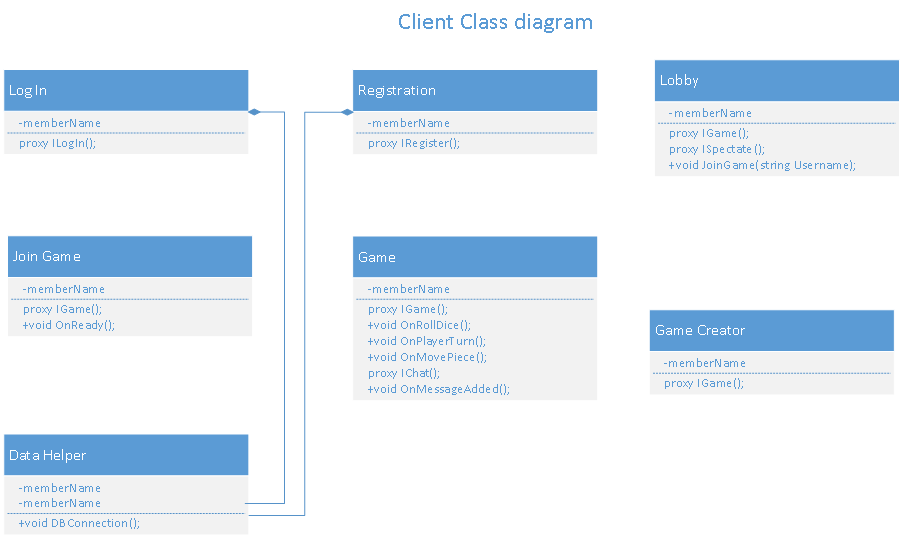
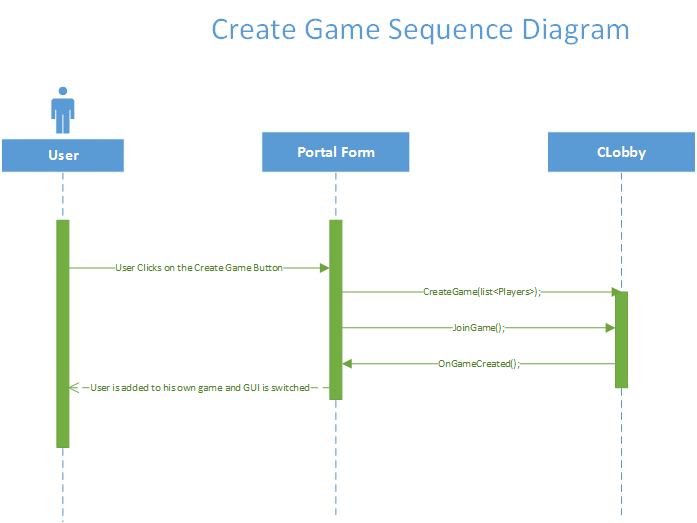
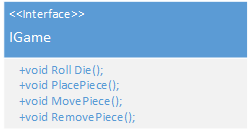
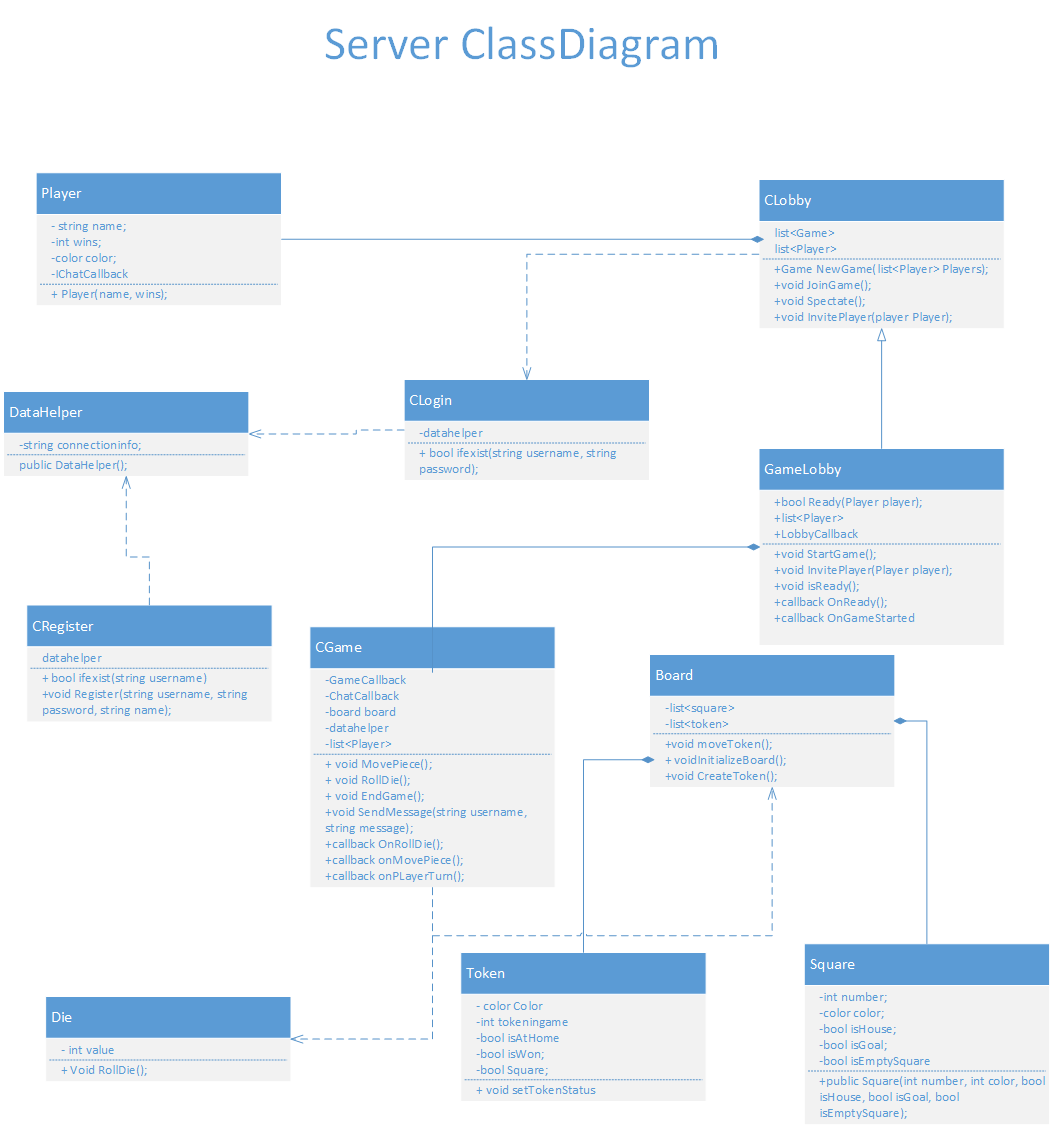
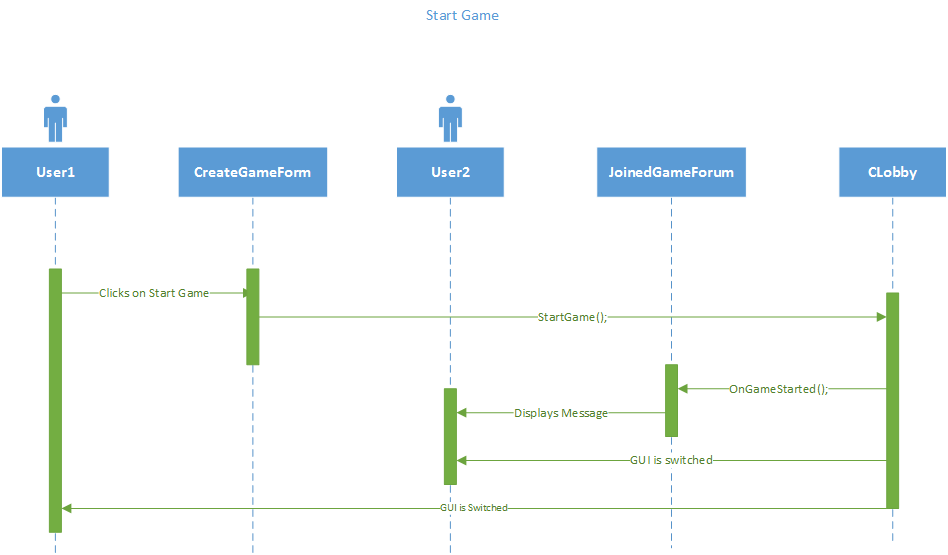
IChat

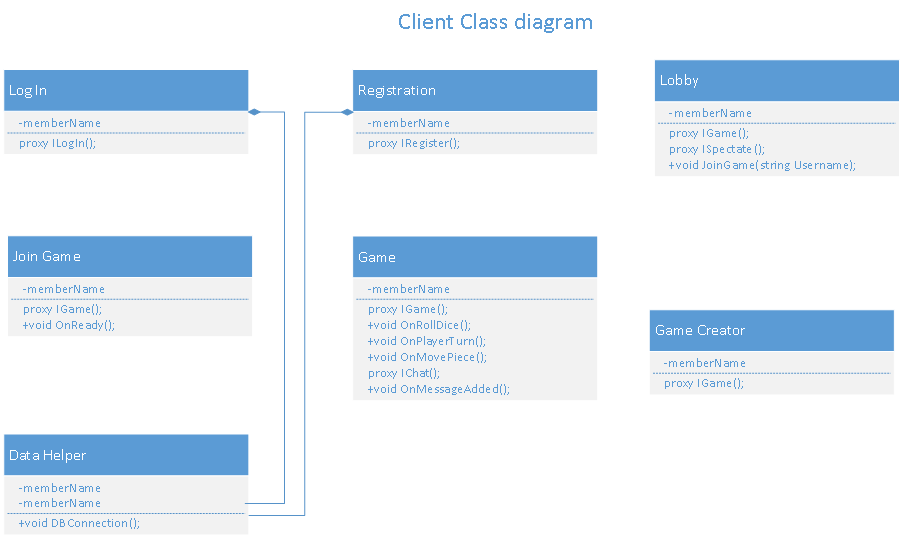
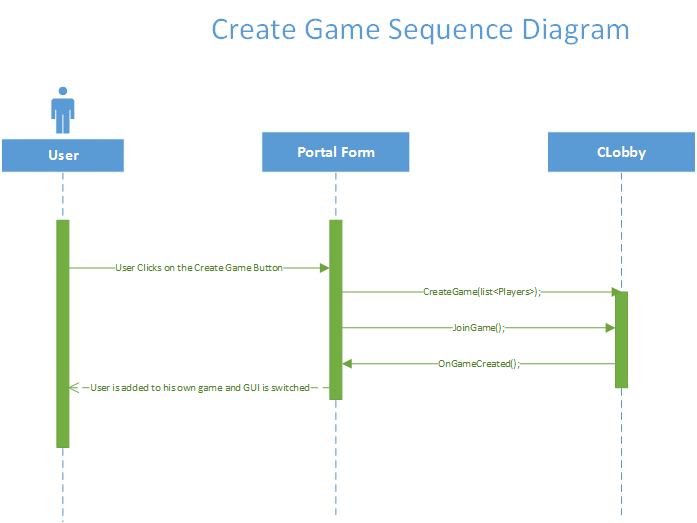
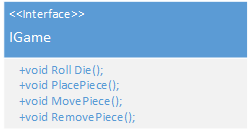
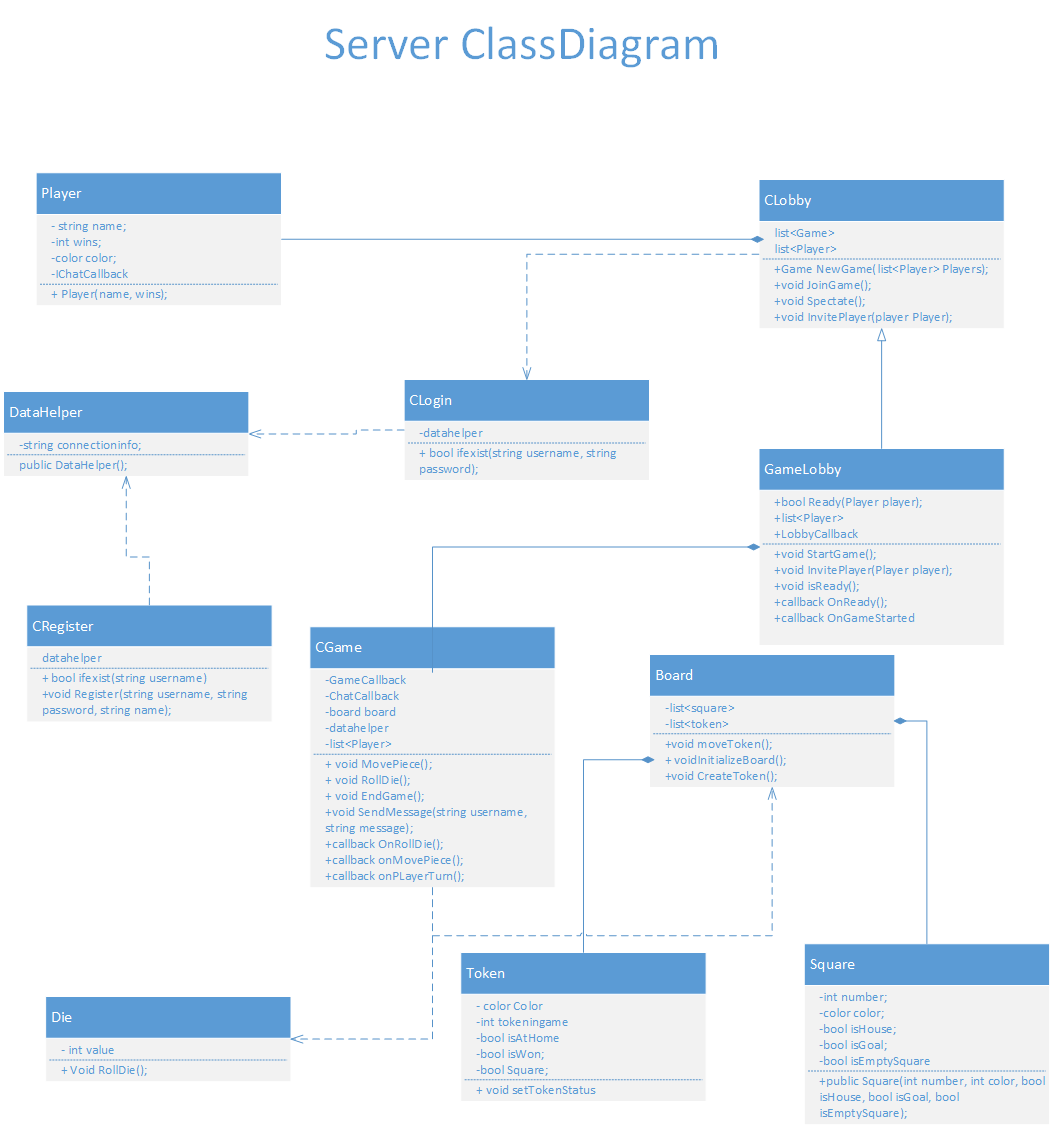
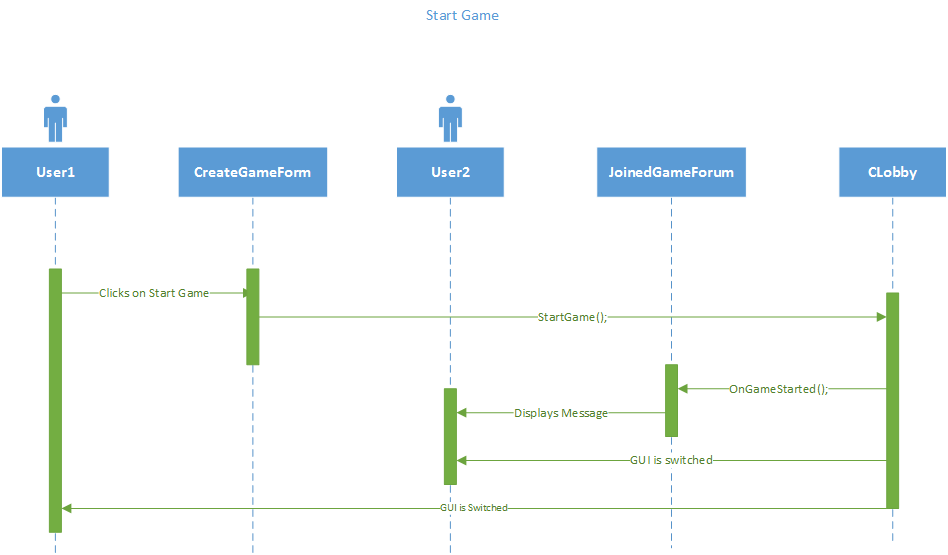
IRegister

ISpectate

ILobby

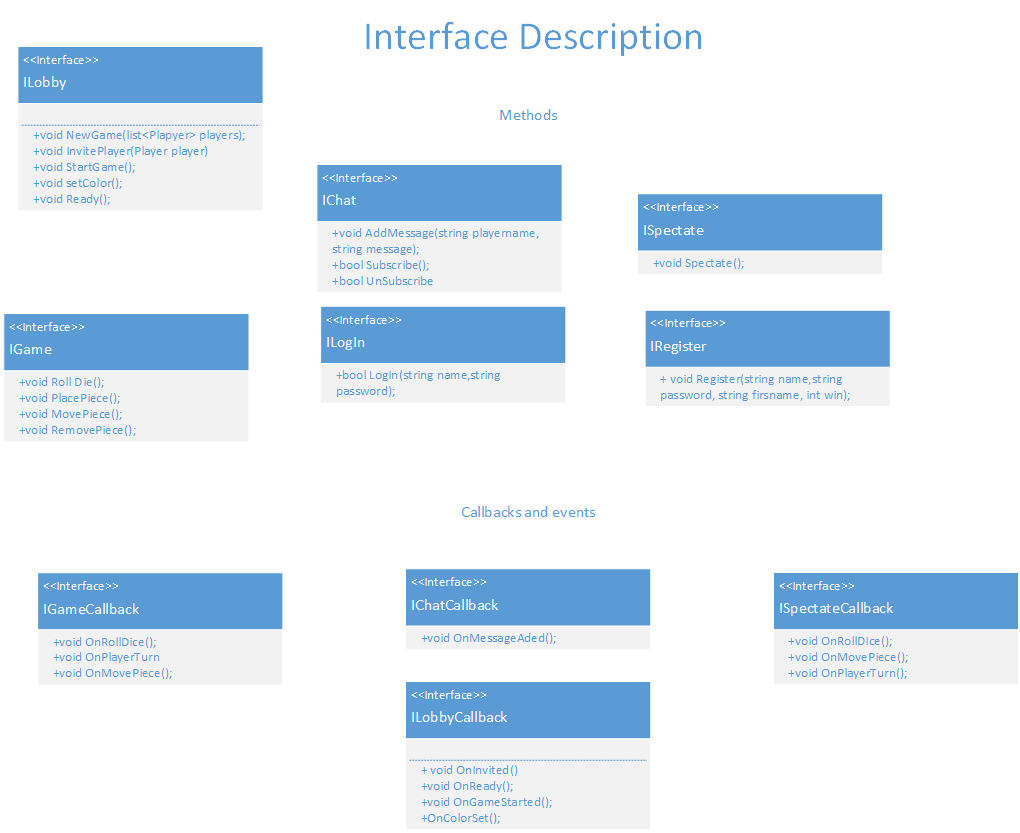
IChatCallback

ISpectateeCallback

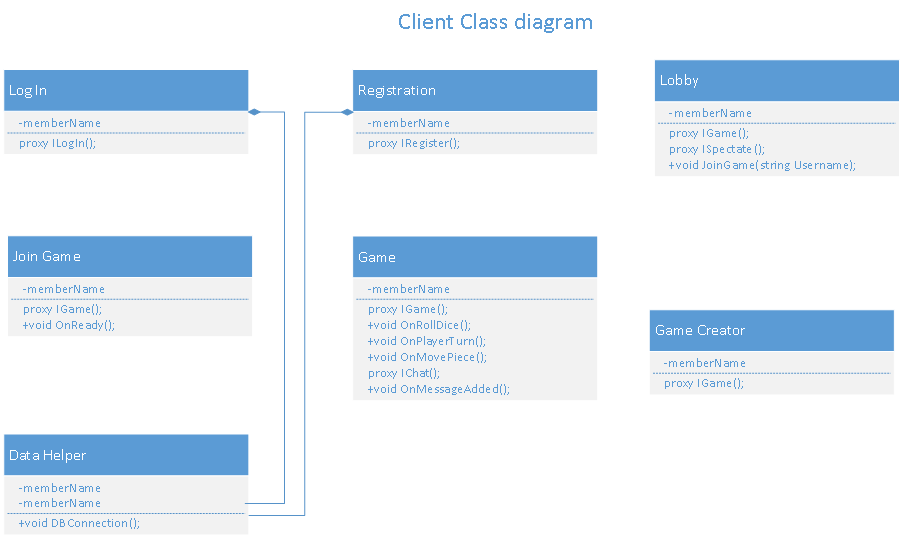
ILobbyCallbackeCallback

Client 2

# Description of Interfaces



# Class Diagram for client

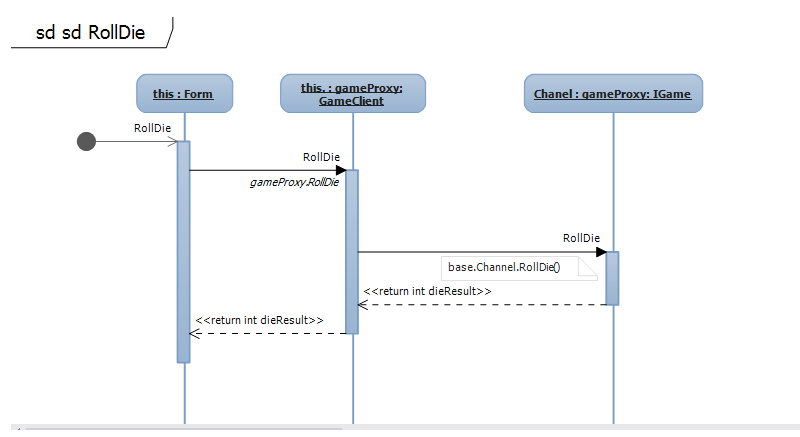


# Class diagram for Service

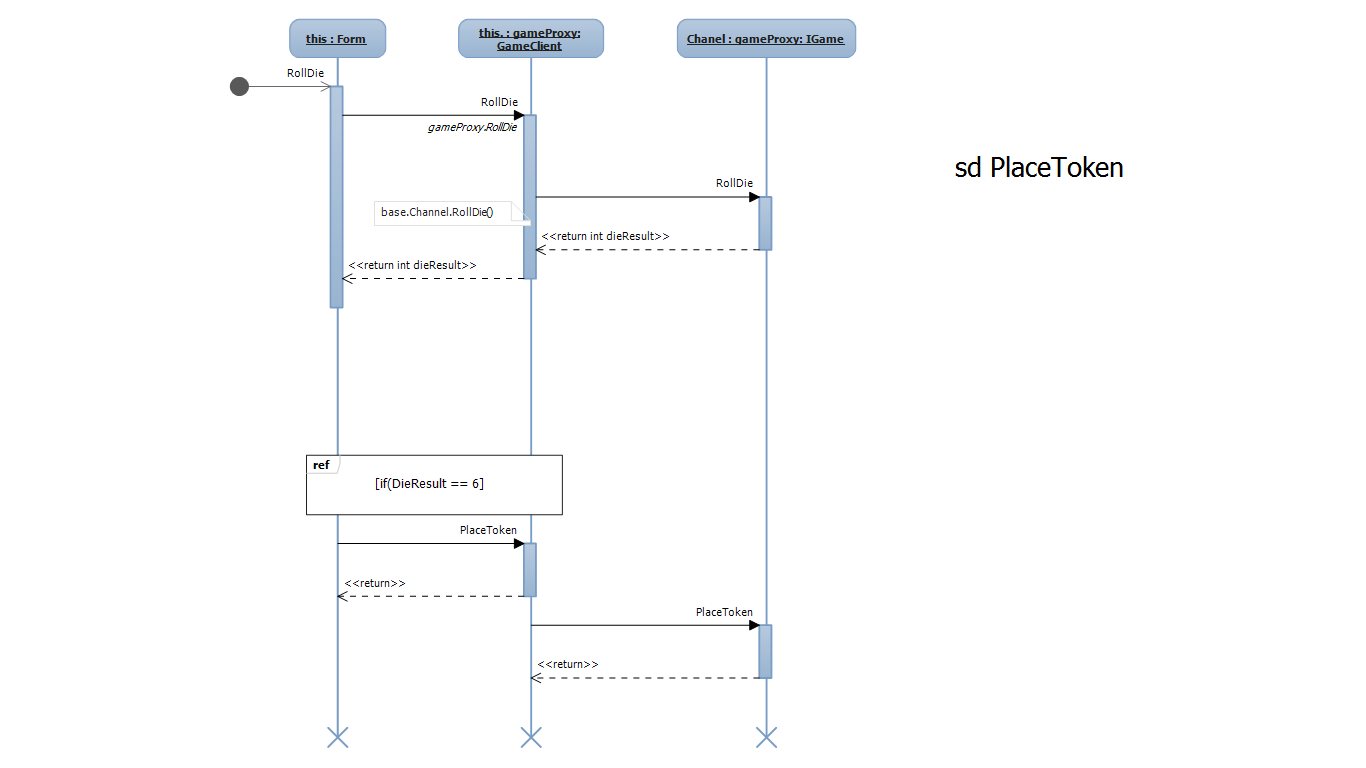


# Sequence Diagrams for MUST use-cases

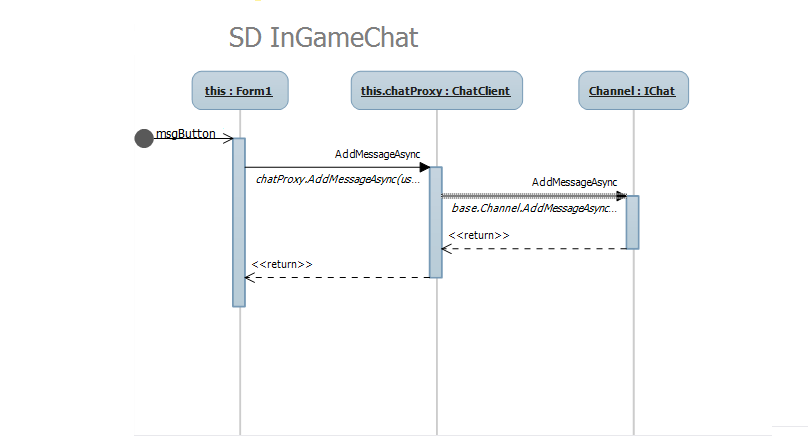
## Roll Die



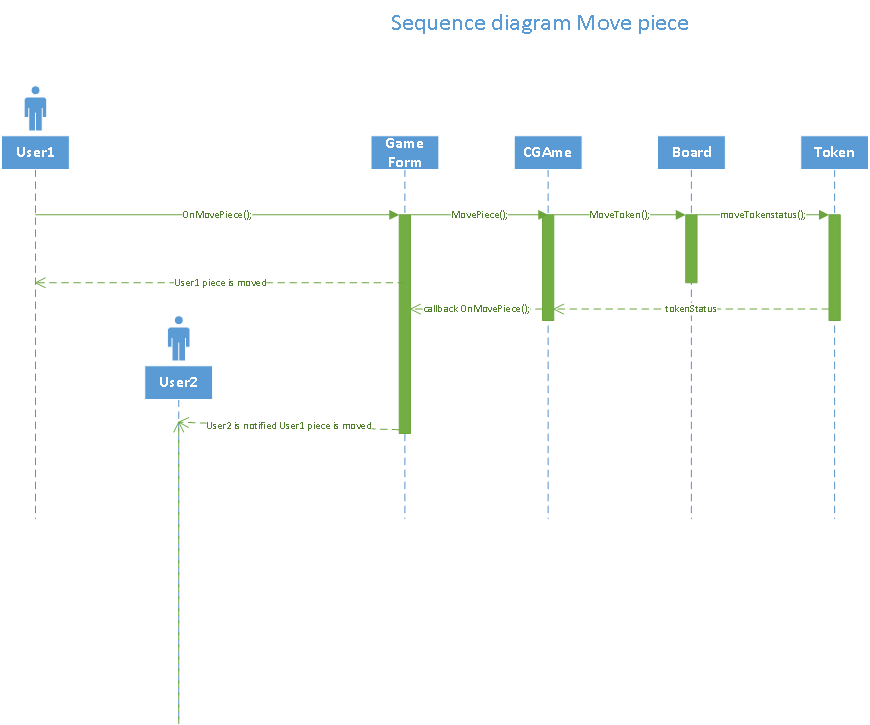
## 6.2 Place Token



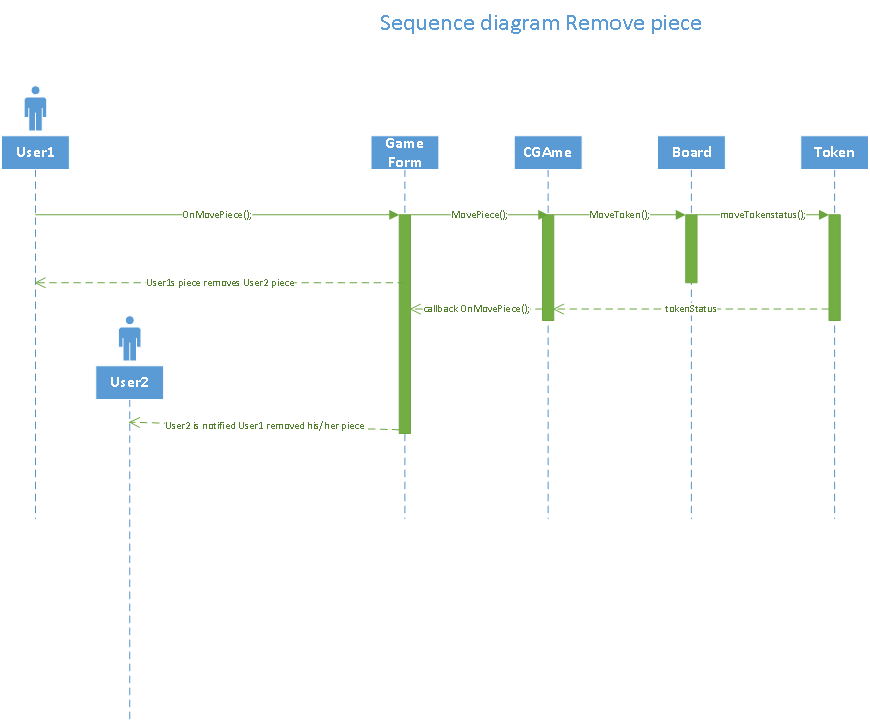
## 6.3 In Game Chat



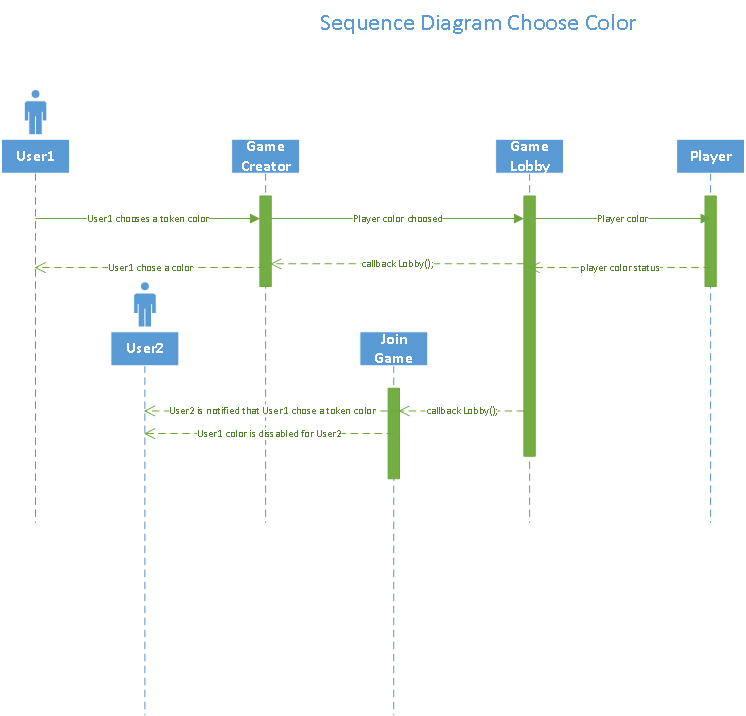
## 6.4 Move Piece



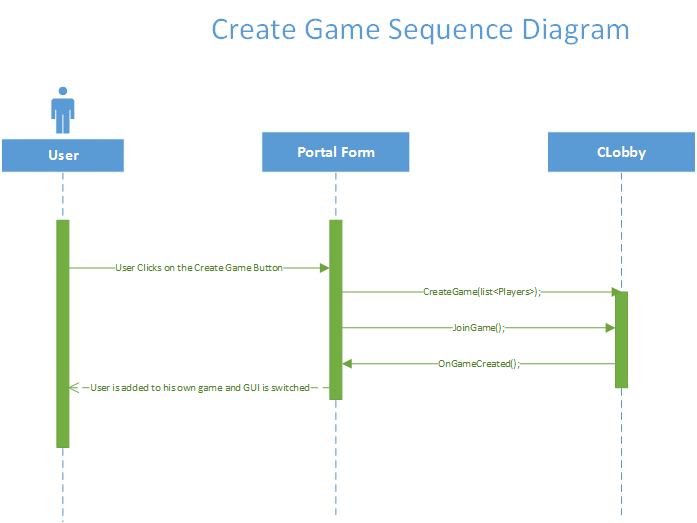
## 6.5 Remove Piece



## 6.6 Choose Color



## 6.7 Create Game



## 6.8 Start Game

## J:\MDW\Design Document\StartGame Sequence Diagram.png

## 6.9 Invite Player

## J:\MDW\Design Document\Invite Player Sequence Diagram.png