

# **CS 320 Course Project - Software Design Document**

**for**

## **Online Multiplayer Board Game**

**Prepared by**

**Group Name: Wolverines**

**Ryan Paulos  
Damon Willingham  
Yousef Kitali  
Jingyu Cao**

**11354677  
11747649  
11536847  
11696853**

**ryan.paulos@wsu.edu  
damon.willingham@wsu.edu  
Yousef.Kitali@wsu.edu  
Jingyu.Cao@wsu.edu**

**Date: 11/20/2020**

<b>CONTENTS</b>	<b>II</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 PROJECT OVERVIEW	1
1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	1
1.3 REFERENCES AND ACKNOWLEDGMENTS	1
<b>2 ACTIVITY DIAGRAM(S) JINGYU</b>	<b>2</b>
2.1 Activity Diagram 1, Macro	2
2.2 Activity Diagram 2, Homepage	3
2.2 Activity Diagram 3, Gameplay Loop	4
<b>3 CLASS DIAGRAM(S) RYAN/YOUSEF</b>	<b>5</b>
3.1 Model-View-Controller+	5
3.2 Context Diagram	6
<b>4 BEHAVIORAL DIAGRAM(S) DAMON</b>	<b>7</b>
4.1 LOGIN SEQUENCE DIAGRAM	7
4.2 REGISTRATION SEQUENCE DIAGRAM	8
4.3 GAME QUEUE SEQUENCE DIAGRAM	9
<b>APPENDIX A - GROUP LOG</b>	<b>10</b>

# 1 Introduction

## 1.1 Project Overview

Our product is intended to meet the needs of the small, yet global network of competitive tic tac toe enthusiasts. Our platform will include persistent aliases through registered accounts, statistics tracking on game performance, and a matchmaking game queue.

- Section 2 contains activity diagrams illustrating the actions user's may perform.
- Section 3 contains a class diagram illustrating the framework upon which the program code will be built.
- Section 4 contains event-driven, behavioral sequence diagrams that show the Login, Registration, and Queue/Play Game events.

## 1.2 Definitions, Acronyms, and Abbreviations

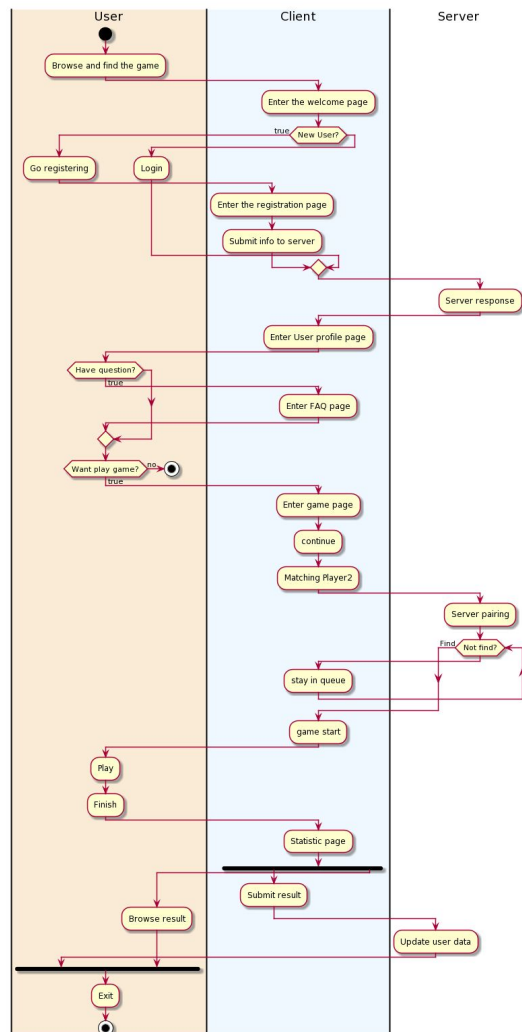
FAQ	Frequently Asked Questions
UML	Universal Modeling Language

## 1.3 References and Acknowledgments

PlantUML, "Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams.," *PlantUML.com*. [Online]. Available: <https://plantuml.com/>. [Accessed: 22-Nov-2020].

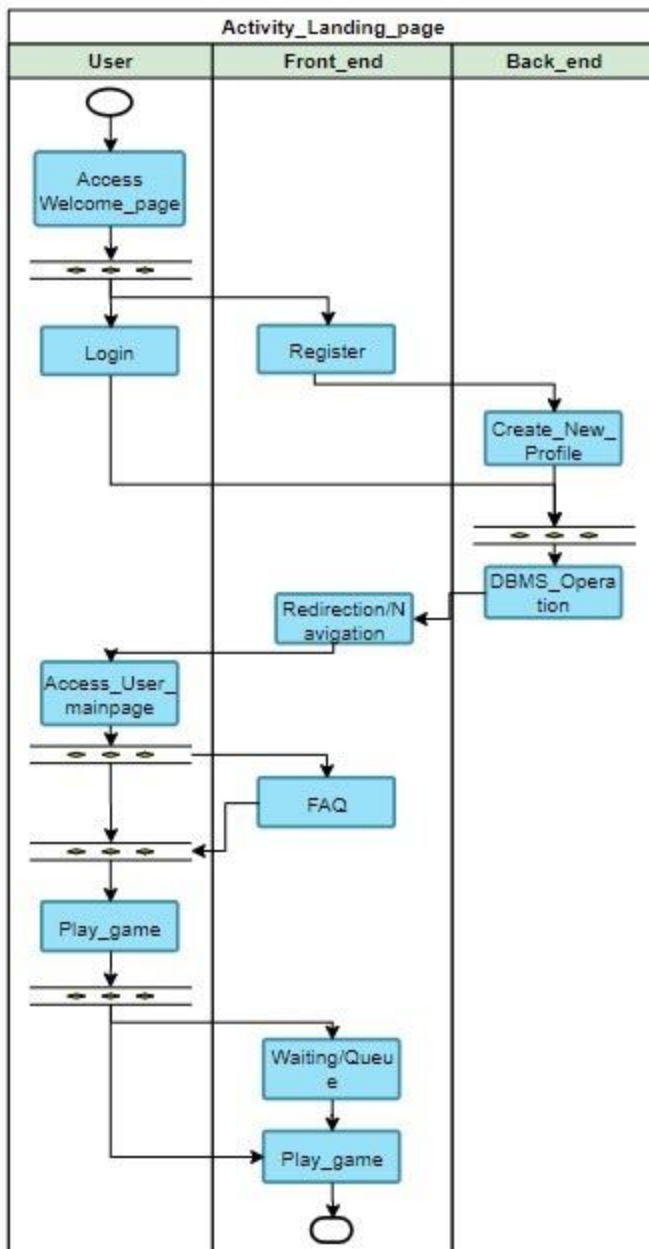
## 2 Activity Diagram(s)

### 2.1 Activity Diagram 1, Macro



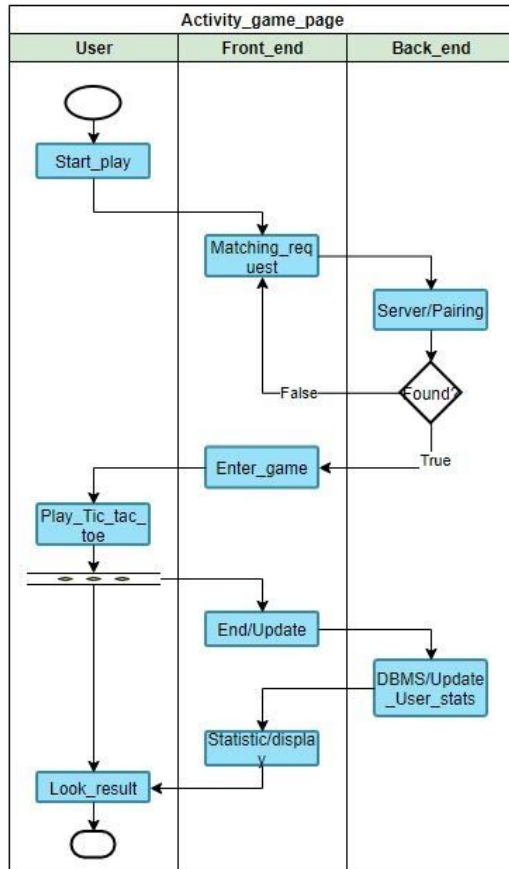
This diagram encapsulates the full set of possible paths a user can take when utilizing our platform.

## 2.2 Activity Diagram 2, Homepage



Broad level diagram showcasing the process pathway for navigating to the homepage/landing page. User's must initially log in or register a new account, after which they are taken to their homepage. From there the player can view the FAQ, which explains the rules of the game, review their current stats, or begin the process of finding a match.

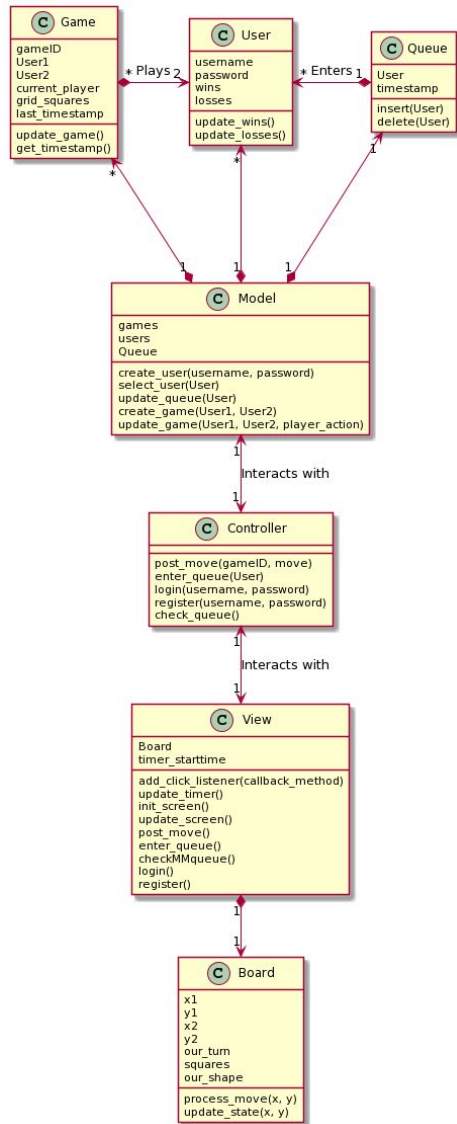
## 2.3 Activity Diagram 3, Gameplay Loop



Activity diagram 3 provides more specific details on the gameplay loop process. Once player's have logged in they are free to enter a match. From here the system will begin searching for an opponent. The game will play out, the results will post to the server, and then the player can reenter matchmaking, closing the core gameplay loop.

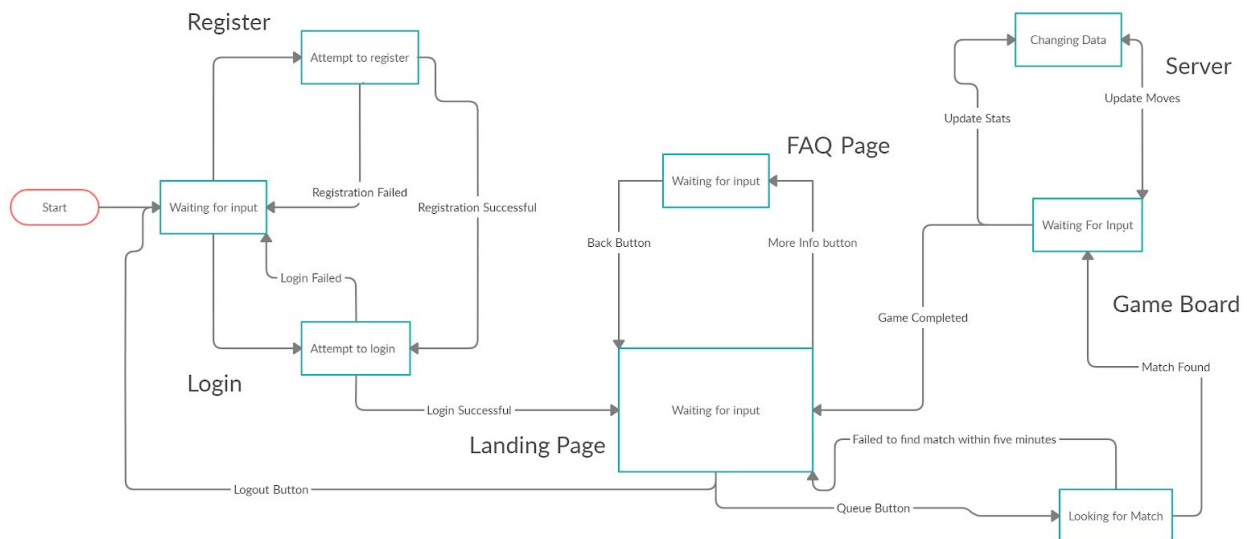
## 3 Class Diagram(s)

### 3.1 Model-View-Controller+



Game	Responsible for tracking the game state of active games.
User	Basic class for storing user information.
Queue	Data structure to facilitate the matchmaking process. A list of players currently seeking a game.
Model	Responsible for tracking state site-wide. Provides the interface for retrieving or modifying state data.
Controller	Liason between the View and Model. Acts as the server-side interface for the client.
View	Relays visual information to the user while providing input data to the server. Acts as the client-side interface
Board	Data structure used locally to track the state of the game and its relation to the physical characteristics of the user's screen.

### 3.2 Context Diagram

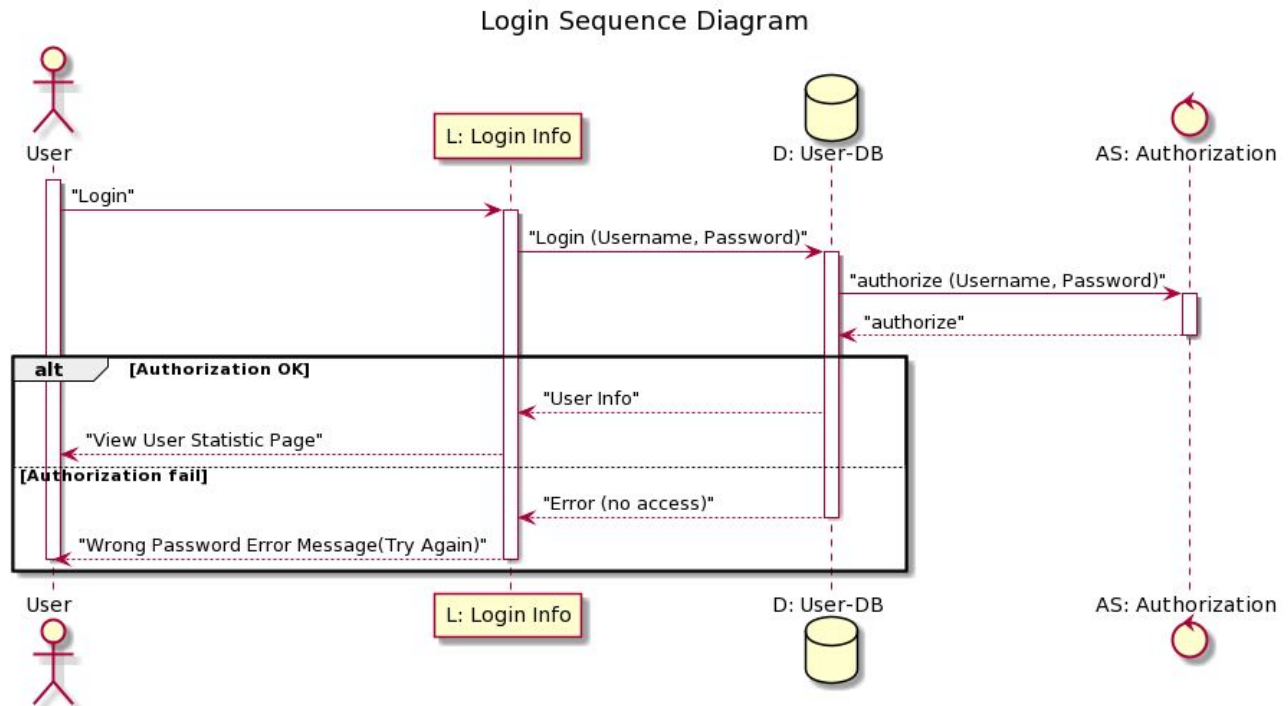


Visual representation of the events facilitated by the classes showcased in the Model-View-Controller+ diagram



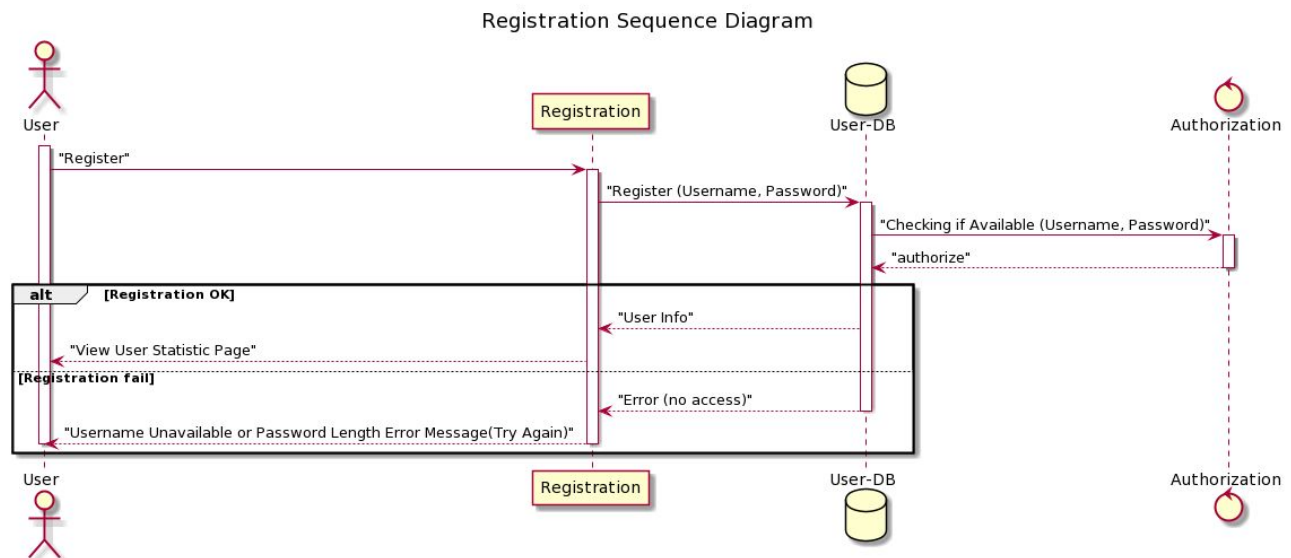
## 4 Behavioral Diagram(s)

### 4.1 Login Sequence



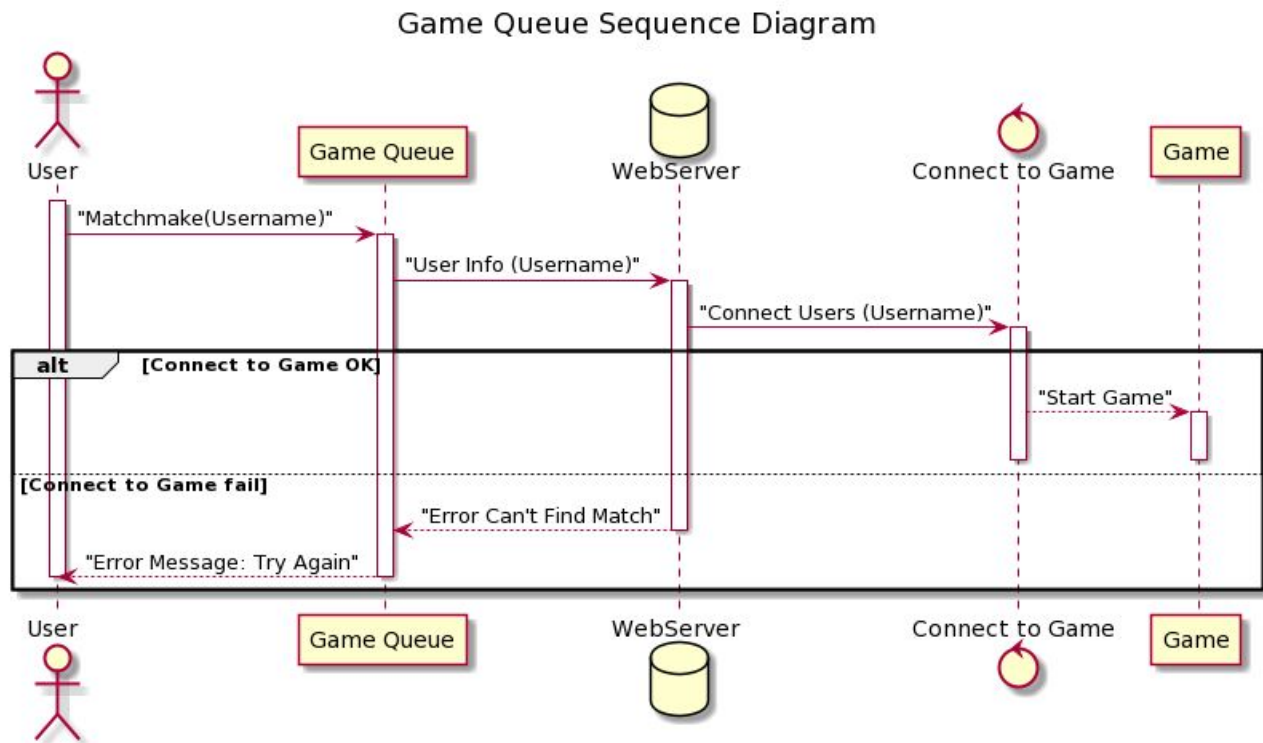
1. The User enters a username and password into the Login() function.
2. The Login() function passes the username and password to the authorization function. The authorization function checks if the username is already in use and if the password matches the password associated with the username key inside the User-DB.
3. If the username is not registered or the password is incorrect then the authorization function returns false, displaying an error message to the User.
4. If the Authorization function passes then the User is sent to their User Statistic Page.

## 4.2 Registration Sequence



1. The User enters a username and password into the Registration() function.
2. The Registration() function passes the username and password to the authorization function.
3. The authorization function checks if the username is already in use and if the password is an adequate length.
4. If the username is not registered then the authorization function returns true, sending the User to the Statistic Page.
5. If the Authorization function fails then the User is given an error message saying either the "Username is Already in Use" or "Password Invalid".

### 4.3 Game Queue Sequence



1. Once the User presses the "Matchmake" button, the Matchmake function is passes the Username to the Game Queue.
2. The Game Queue passes the User Info to the WebServer.
3. The WebServer then tries to match two Users to the game.
4. If the WebServer is able to connect two Users then the game starts.
5. Else if the WebServer fails to connect two Users an Error message displays.

## Appendix A - Group Log

10/20/2020 – Ryan and Yousef conducted the initial group meeting. Ryan and Yousef reviewed the requirements for project milestone 1, developed a methodology for developing the SRS document, and determined a general agenda for the following meeting on 10/27/2020.

10/27/2020 - Ryan, Yousef, and Damon met to develop an outline for the SRS document. In this meeting we divided the document into three parts, one for each of us to complete. The process of incorporating each section will commence at the next meeting. The sections were assigned as follows:

Ryan: Section 1  
Yousef: Section 2  
Damon: Section 3

11/3/2020 - Attended by Ryan, Yousef, and Damon. Reviewed and integrated our individual sections into the first draft of a completed SRS document.

11/17/2020 4:00PM- Group Meeting to discuss how the work for the Software Design Document would be split up. Section 2: Activity Diagram was assigned to Jingyu Cao, Section 3: Class Diagram was assigned to Ryan Paulos and Yousef Kitali, and Section 4: Behavioral Diagram was assigned to Damon Willingham.

11/18/2020-11/21/2020- Throughout the week we each worked on our assigned sections while using Discord to communicate. We used Discord to share our ideas for diagrams and get feedback on diagrams.