

---

# **CSC340: Software Engineering Software Requirements Specification (SRS)**

**GAFF: A Gaming App for Finding Friends**

**11-28-2022**

**Version 4**

**By: Jessica Frank, Alec Droegemeier, Alex Wesley**

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>1. Introduction</b>	<b>3</b>
1.1 Document conventions	3
1.2 Definitions, Acronyms, and Abbreviations	3
<b>2. General Description</b>	<b>3</b>
2.1 Product perspective	3
2.2 Product features	4
2.3 User class and characteristics	4
2.4 Operating environment	4
2.5 Constraints	4
2.6 Assumptions and dependencies	4
<b>3. Functional Requirements</b>	<b>5</b>
3.1 Primary	5
3.2 Secondary	5
3.3 Developer Assignments	5
<b>4. Technical Requirements</b>	<b>6</b>
4.1 Operating System & Compatibility	6
4.2 Interface requirements	6
<b>5. Non-Functional Requirements</b>	<b>7</b>
5.1 Security requirements	7
5.2 Software quality attributes	7
5.3 Process Requirements	8
<b>Appendix A: Use Cases</b>	<b>9</b>
Use-Case Model	9
Use-Case Descriptions	10
Use-Case Scenarios	10
Use-Case Responsibilities	12
<b>Appendix B: Software Architecture</b>	<b>13</b>
Architecture Diagrams	13
Database Relational Schema	14
<b>Appendix C: Software Design</b>	<b>15</b>
State Machine Diagrams	15
UML Class Diagram	18
<b>Appendix D: Presentation Scenarios</b>	<b>19</b>
Player: Log In	19
Player: View Game List	19
Player: View Game Details	20

Player: Join/Leave Game	20
Player: View Profile	21
Moderator: Add New Game	22
Moderator: Change Game Details	23
Admin: Views System Logs	24
Admin: Promote/Demote a User	24

# 1. Introduction

*Author: Jessica Frank*

## 1.1 Document conventions

The purpose of this Software Requirements Document (SRD) is to detail the functionality and requirements of our project GAFF (Gaming App for Finding Friends). This document will describe client-oriented requirements that include how our application will look and perform, as well as developer-oriented requirements that include functional and performance-related requirements.

## 1.2 Definitions, Acronyms, and Abbreviations

Term	Definition
API	An abbreviation for Application Programming Interface, it is a method of communicating with another application to retrieve or send data.
HTTP	HTTP is a protocol commonly used for communicating with websites.
MySQL	MySQL is the database tool we will use to store user and game data.
Material Design Lite	Material Design Lite is a library of components that lets you easily add style and design to web pages.
Spring	Spring is the Java development framework that we are using to build our application.

# 2. General Description

*Author: Jessica Frank*

## 2.1 Product perspective

The goal of this project is to create a web app for video game players looking to find friends to play with. Games are an activity that is often more fun with friends, but can be hard to find people that play the type of games you like. Our website aims to fix this problem by introducing players to others who like the same sort of games they do. This application will let users browse games by various types and genres, view the list of players that are playing a game, and introduce themselves to other players through profiles.

## 2.2 Product features

The product will have several features to give users a better experience.

- Player Profiles will make the experience of finding other players easier by displaying individual information about users.
- Game Search will allow players to sort our site's games by genre and category to make finding games on our site easier.
- Game Addition is a method for moderators to keep the site up-to-date by adding new and upcoming games.
- Administrator Log gives admins a useful tool to keep track of the site's use.
- Role Transitions is our site's tool for admins to promote and demote moderators.

## 2.3 User class and characteristics

Our website will have three user categories. All users are expected to know how to use a web browser and navigate a website.

- Player
  - Characteristics: Players are expected to have some general knowledge about video games, such as video game genres and platforms.
  - Developer: Alec Droegemeier
- Moderator
  - Characteristics: Moderators are expected to have in-depth knowledge about video games, since they will be responsible for adding new games to the site. They should be aware of games that Players are interested in which haven't been added to the site yet.
  - Developer: Jessica Frank
- Administrator
  - Characteristics: Administrators are expected to have enough video game knowledge to analyze how well the site is working. They will be responsible for promoting Players to Moderators and demoting Moderators if necessary.
  - Developer: Alex Wesley

## 2.4 Operating environment

This software is a web application that will run on an Apache server. It is designed to be viewable in any modern web browser on any operating system.

## 2.5 Constraints

To limit user error when selecting game categories, we will create a menu of genres and other categories for the user to choose from.

## 2.6 Assumptions and dependencies

This website uses a database to store user and game information, so it depends on the web server connecting to the database for proper functionality. Parts of the site use the RAWG API to retrieve information about games, and those sections may not function properly if the API is not working.

# 3. Functional Requirements

*Author: Jessica Frank*

## 3.1 Primary

- FR1: The system will allow any user to view a list of game categories. Once the user selects a category, they will be able to see a list of games in that category.
- FR2: The system will allow the user to select individual games in order to view more details. These details will include the game's description, platforms the game is available on, and a list of users on the site who play that game.
- FR3: The system will give each user a profile that other users can view when browsing the site. The database stores information about the user such as a bio, and/or links to third party applications such as Steam, Discord, Twitter, etc which are saved in their profile. This profile can be accessed through the list of users displayed on a game's individual page.
- FR4: The system will allow moderators to add new games to the database of games that users are playing.
- FR5: The system will allow moderators to edit details of games currently listed on the site.
- FR6: The system will allow administrators to view a log of actions that have been performed on the site.
- FR7: The system will allow administrators to change a user's permissions between the Player and Moderator role.

## 3.2 Secondary

- FR8: The system will store a database of user information to support logins, monitor user permissions, and store profile information.
- FR9: The system will store a database of game details to support the primary requirements that depend on information about games.
- FR10: The system will use the RAWG API to automatically give suggestions for moderators when they add a new game to the system.

## 3.3 Developer Assignments

The project description requires each functional requirement to be assigned to a developer. This is the current assignment of project requirements.

Jessica Frank	Alec Droegemeier	Alex Wesley
FR4: Moderator Game Addition FR5: Moderator Game Edits FR9: Game Database FR10: API Suggestions	FR1: Viewing Game Lists FR2: Viewing Game Details FR3: User Profiles FR8: User Database	FR6: Administrator Log FR7: Permission Editing

## 4. Technical Requirements

*Author: Jessica Frank*

### 4.1 Operating System & Compatibility

This application is designed to run on any device with the capability of running the Apache web server and MySQL database that the site depends on. Any device with a web browser is capable of viewing the website once the application is running on an appropriate device.

### 4.2 Interface requirements

#### 4.2.1. User Interfaces

This web application will use Material Design Lite components for a clean and consistent style throughout the application. This application will have several interfaces for different functions. All interfaces contain a navigation bar at the top of the page. This navigation bar has additional elements if the user has logged in as a moderator or administrator, which are not visible to regular users. These additional sections lead to moderator-only and administrator-only sections of the application.

Interface 1. The landing screen is the first page a user accesses when they access the website. It contains information about the website. If a user is not logged in, it will redirect them to the login page. This screen contains links to pages a user has permission to access.

Interface 2. The login screen contains a form with a username and password field for the user to sign in with. If the user's information is correct, they will be navigated to the game list screen. After a successful login, a user will gain access to sections of the site that users with their role (player, moderator, or admin) are supposed to have access to.

Interface 3. The game list screen contains two main sections. The first is a list of selectable video game genres, and the second section shows a list of games. When one of the genres is selected, the second section shows a list of games found in our site's database that match the genre. Each of the games has a brief description and can be clicked on to continue to a page with more details about that game.

Interface 4. The game details screen contains a section with a description and details of the selected game. This screen also lists users on our site that play the game. Each of the users can be clicked on to navigate to their profile page.

Interface 5. The profile screen displays information about a user, such as games they play and any contact information they have shared.

Interface 6. The game addition screen is only visible to moderators. There will be a text box where the moderator enters the name of the game they wish to add to the site. They can also enter an API key to autofill the next section of the form. Once they submit the name and API key, they will receive suggestions for filling out the rest of the form, which contains sections for the game's platforms, genre, and other information that will be added to the game's information in the site's database. If there was no API key submitted, the moderator will have to fill out the form without autofilled details. Once all of the fields are filled and the moderator submits the form, the game will be added to the database.

Interface 7. The game editing screen is only visible to moderators. This field will be similar to the game addition screen, except that all fields will be pre-filled with the information currently in the application's database. When the moderator submits the form on this page, any changes in the game's information will be saved to the database.

Interface 8. The promotion and demotion page is only accessible to administrators. This page contains a text box which is used to type in the user's name and two selectable options to promote them to a moderator or demote them to a player.

Interface 9. The site history page is only visible to administrators. This page will contain a log of recent events on the site so that administrators can keep track of how the site is performing and whether users will need to be promoted or demoted.

#### **4.2.2. Hardware Interfaces**

The system for this project is a demo that is designed and tested to run locally on a personal computer, so supported devices are any computer that is able to locally run Apache and MySQL servers. No other hardware is required.

#### **4.2.3. Communications Interfaces**

The application uses HTTP to communicate with the RAWG API for game information, and the MySQL protocol for connection to the database.

#### **4.2.4. Software Interfaces**

This project uses Spring Boot for front-end development of the application, and MySQL as our database management system.

## **5. Non-Functional Requirements**

*Author: Jessica Frank*

### **5.1 Security requirements**

NFR1: The system will only be usable by authorized users.

NFR2: Only authorized moderators will be able to add or edit game data.

NFR3: Only authorized administrators will be able to edit user roles and permissions.

### **5.2 Software quality attributes**

#### **5.3.1. Availability**

This application will be hosted on a local web server. To connect to the server, users will need to navigate to the website using a web browser.

#### **5.3.2. Correctness**

In order to fix any incorrect information on our site, moderators will be able to edit the details and descriptions of games in the database.

#### **5.3.3. Maintainability**

Administrators will be able to promote users if more moderators are needed to keep up with games being added to the site.

#### **5.3.4. Reusability**

This service will have one template for storing game information and one template for user information, so each of those templates will need to be reusable for any type of video game and user, respectively.

#### **5.3.5. Security**



Although this website will not store sensitive information, it will still contain contact information for many users. For that reason, we will require users to be logged in before they can access most of the site. In addition, users must be an authenticated moderator or administrator to access game information editing and user permission editing sections of the service.

## **5.3 Process Requirements**

### **5.3.1. Development Process Used**

We will be using the Scrum methodology to have a more flexible process that is able to react to unexpected changes, issues, or delays.

### **5.3.2. Time Constraints**

This project must be finished before the presentations at the end of the Fall 2022 semester. The application must have a working prototype by the 18th of October, and a presentation of the final product must be ready by the 15th of November.

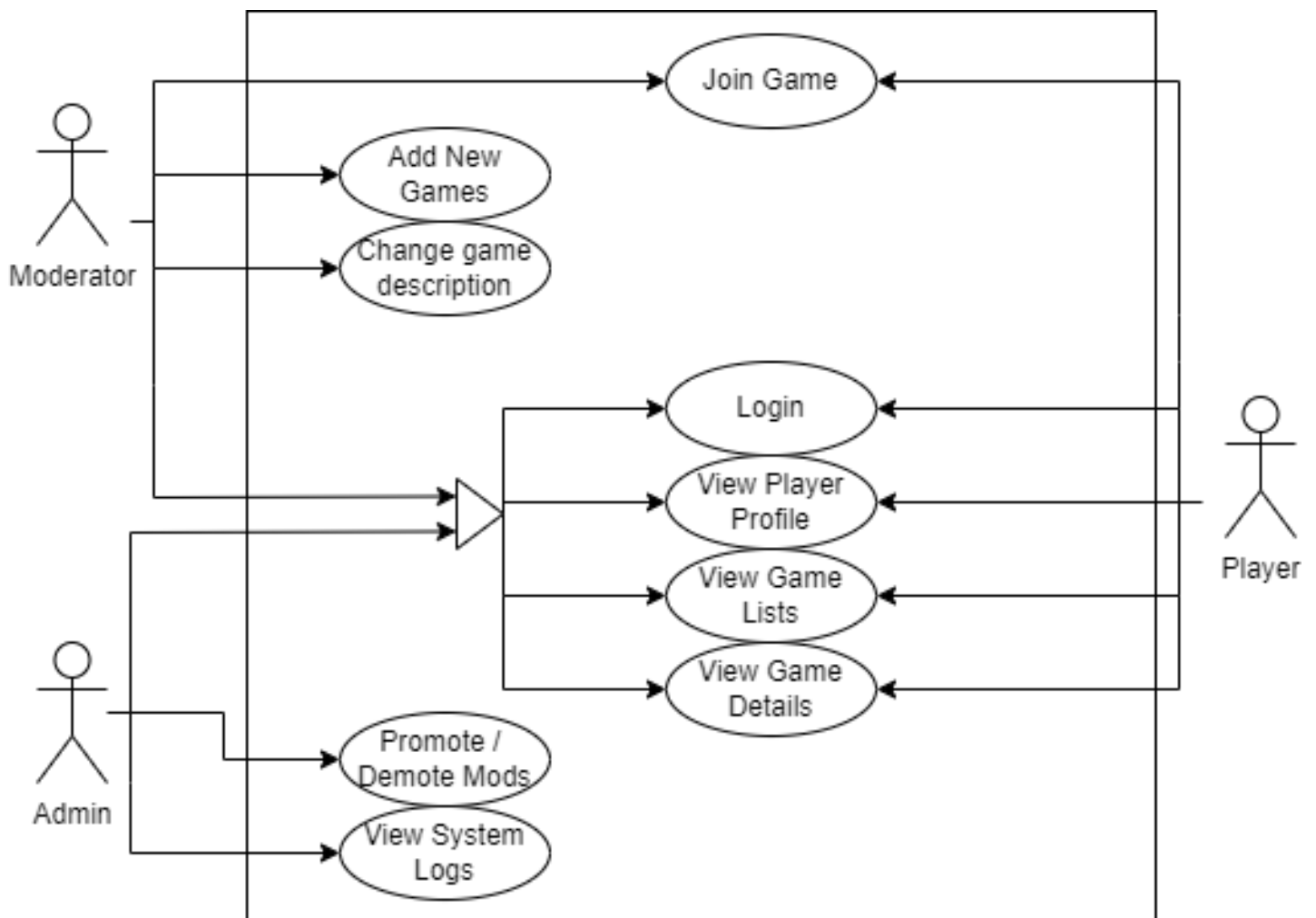
### **5.3.3. Cost and Delivery Date**

This is a student project, and all tools used are freely available at the time this project is created. The final deliverable must be completed before the instructor's deadline.

# Appendix A: Use Cases

*Authors: Jessica Frank (Use Case Scenarios), Alec Droegemeier (Use Case Model, Descriptions, and Responsibilities), Alex Wesley (Use-Case Scenarios)*

## Use-Case Model



## Use-Case Descriptions

**Login-** All users should be able to log in to the website, players will have access to the main website where they will be able to view game lists, game details, and other players profiles.

**Promoting/Demoting Moderators-** Admins will have full control over the website and are the only ones who have authority to give or take away the moderator role from the user.

**View System logs-** Admins will be able to view the system logs and see what changes have been made to the website.

**Add new games-** Moderators are in charge of adding new games to the website for players to be able to find new friends in.

**Change game description-** Moderators can change the description of a game to better describe the game, or change incorrect platforms or genres.

**View player profiles-** All logged-in users will have access to view the list of players that are interested in specific games.

**View game lists-** All logged-in users will be able to see a list of games that are available in each genre.

**View game details-** All logged-in users can view the details of each game.

**Join game-** All logged-in users can select any game and connect to the list of people playing the game.

## Use-Case Scenarios

### A) Login

- i) **Initial State** - A user is viewing the login screen
- ii) **Normal Function** - All types of users can enter their username and password in the form on the login screen.
- iii) **Possible Error** - A user's login fails because the username or password does not match the stored information for the player. The user will not be redirected to the site's main page.
- iv) **Completion Status** - The user will be redirected to the site's main page, and can then view the different sections of the site that they are able to navigate to.

### B) Promoting/Demoting Moderators

- i) **Initial State** - An admin is logged into their account and can view the main admin panel where they will be able to make changes on the website.
- ii) **Normal Function** - Admin will be responsible for promoting and demoting moderators.
- iii) **Possible Error** - The admin can demote or promote the wrong users. If this happens, the admin will be able to change the users permissions back to what they were.
- iv) **Completion Status** - The demotion/promotion successfully happened and that user now has the correct privileges.

#### C) View System Logs

- i) **Initial State** - An admin is logged into their account and can view the main admin panel which will have access to the system logs.
- ii) **Normal Function** - The admin will be able to check system logs and see recent changes made to the website and determine if the changes made are appropriate.
- iii) **Possible Error** - If a user is not logged into the admin account they will not be able to access system logs.
- iv) **Completion Status** - The admin is successfully logged in and can view the system log to determine if the website is running properly.

#### D) Add New Games

- i) **Initial State** - A moderator is logged into their account and can navigate to the moderator dashboard where they can create a new game.
- ii) **Normal Function** - The moderator can choose to create a new game, and give it the correct parameters. The site will attempt to automatically fill out the game's details using the RAWG API.
- iii) **Possible Error** - The moderator can create a new game that does not exist. If the API does not find any details on a game, the moderator will have to manually input the game's details.
- iv) **Completion Status** - The game is successfully added to the list of games that players will have access to.

#### E) Change Game Description

- i) **Initial State** - A moderator is logged into their account and can navigate to the moderator dashboard where they can update game descriptions.
- ii) **Normal Function** - The moderator can choose a game that needs to be adjusted and can edit the description.
- iii) **Possible Error** - A moderator can change the description of the wrong game. If this happens, they can edit the game again to change the game's details back to the correct information.
- iv) **Completion Status** - The game has successfully been updated with a new description.

#### F) View Game Lists

- i) **Initial State** - The player is logged into an account and can select the games list button.
- ii) **Normal Function** - The list of games available will be displayed to the player. Players will have the option to view only a certain category of games.
- iii) **Possible Error** - If the player attempts to view a category that does not exist, no games will be shown in the list that is displayed.
- iv) **Completion Status** - The list of games is displayed properly and allows the player to view each game to learn more details about the game and the other players interested in that game.

#### G) View Player Profile

- i) **Initial State** - The player is logged into an account and can navigate to the list of games to find players who play similar games.

- ii) **Normal Function** - The user can view a player's profile directly from the games list and can see information about the user whose profile they are viewing, such as a bio and links to third party applications like Steam or Discord.
- iii) **Possible Error** - Changed URL to a user id that does not exist.
- iv) **Completion Status** - The correct player's profile is displayed and the links work.

#### H) View Game Details

- i) **Initial State** - The player is logged into an account and has navigated to the game list.
- ii) **Normal Function** - The player can select a game from the game list screen. They will then view details about the game, like its genre and description. The player can also see a list of players interested in the game.
- iii) **Possible Error** - The player attempts to view a game that doesn't exist. To prevent this, the site will only lead users to view a game's details by selecting an option from a list instead of typing a name that could be misspelled.
- iv) **Completion Status** - The player can successfully view the game's details.

#### I) Join Game

- i) **Initial State** - The player is logged into an account and has navigated to a specific games page.
- ii) **Normal Function** - The player can select a game from the game list screen. They will then view details about the game, and can decide if they want to join the game's list of players. The player will also have the option to leave the game's list of players.
- iii) **Possible Error** - The player could join a game that they did not mean to. There is a button to disconnect from the game if the player does not want to be a part of a game's list of players.
- iv) **Completion Status** - The player was successfully added to the list of players for that game.

### Use-Case Responsibilities

Jessica Frank	Alec Droegemeier	Alex Wesley
Login	View Game Lists	View System Logs
Add New Games	View Game Details	Promoting/Demoting Moderators
Change Game Description	View Player Profile	
	Join/Leave Game	

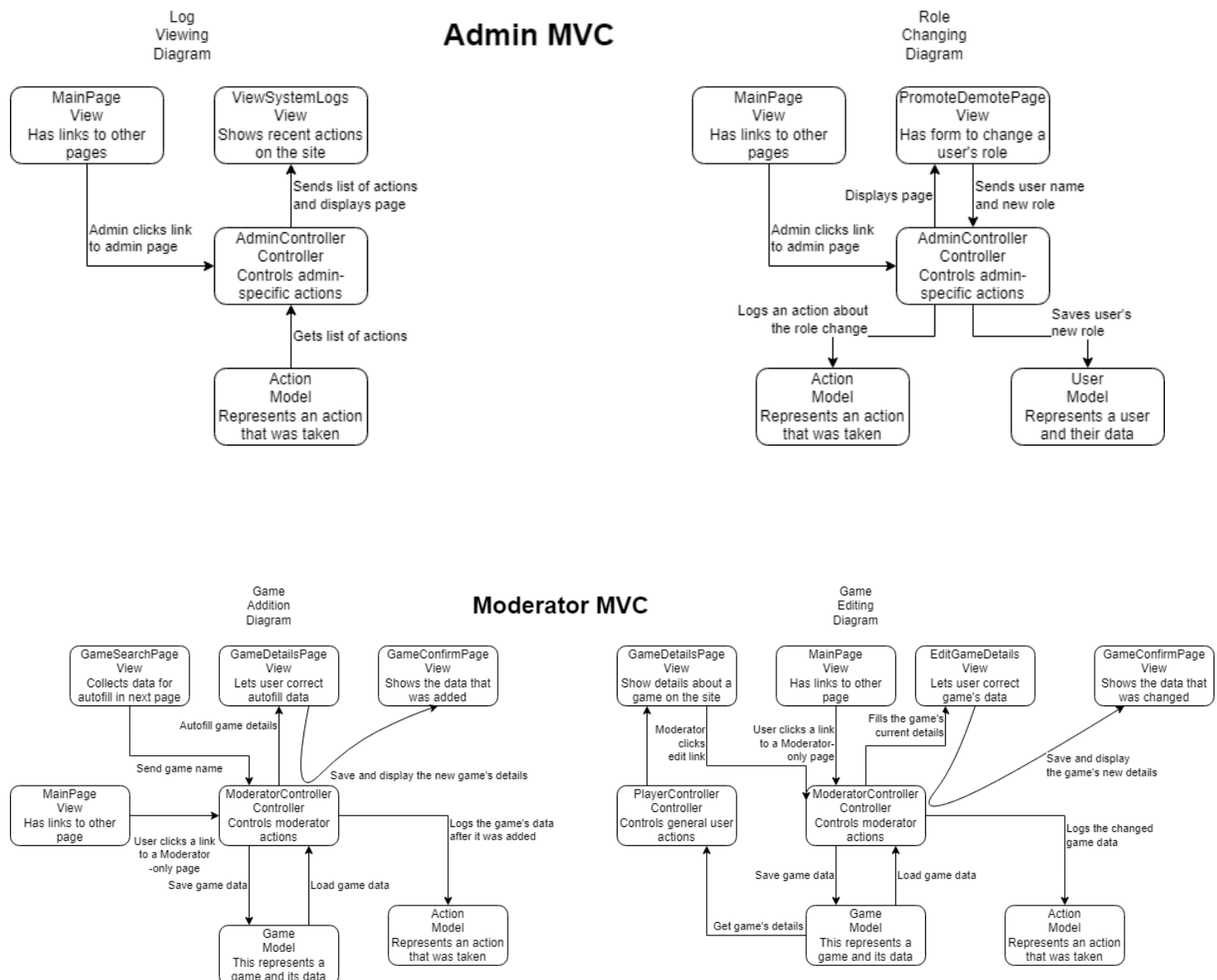
# Appendix B: Software Architecture

Author: Jessica Frank (All Diagrams), Alec Droegemeier (Helped with Player Diagram)

## Architecture Diagrams

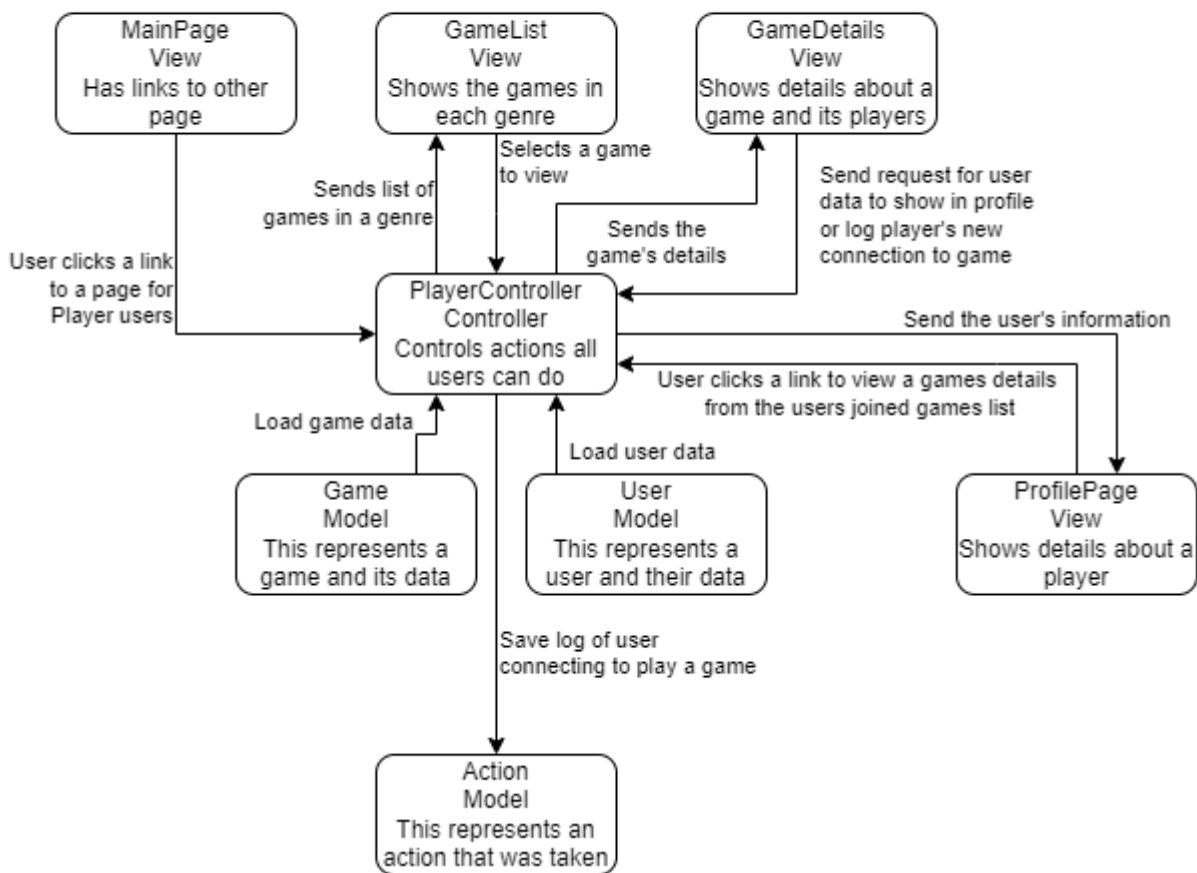
Since we were not given an example of how the architecture diagrams should be written, the MVC diagram below has been broken down into sections for different actors and use cases to improve readability.

Note - this architecture diagram does not include classes such as the SecurityConfig, DemoApplication, DemoController, and the Login Page because their connections are controlled by Spring and not the application itself. For more details on those classes please refer to the UML Class Diagram section.

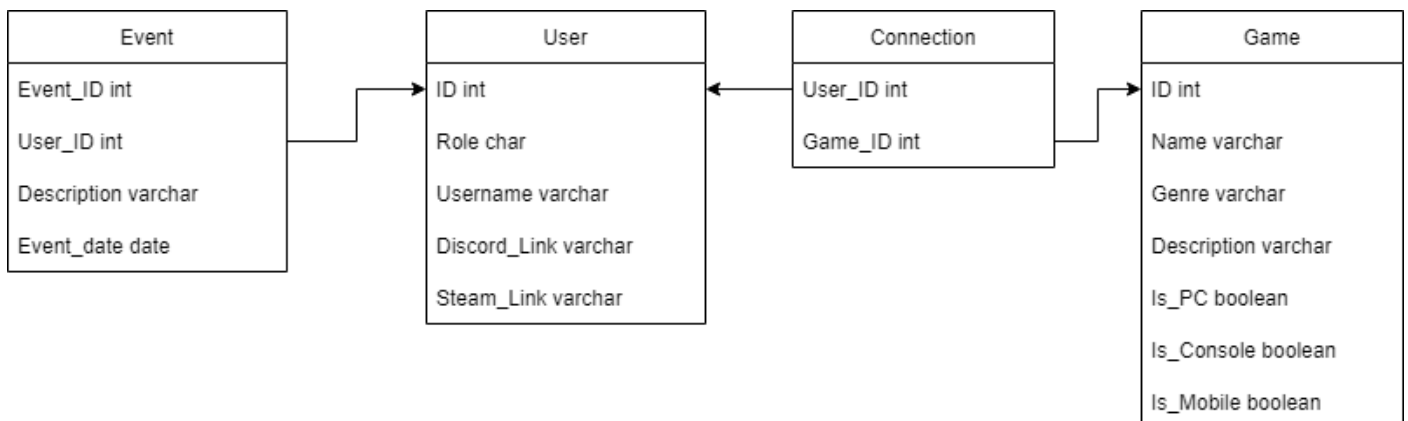


# Player MVC

Viewing game lists, game details, and profiles



## Database Relational Schema

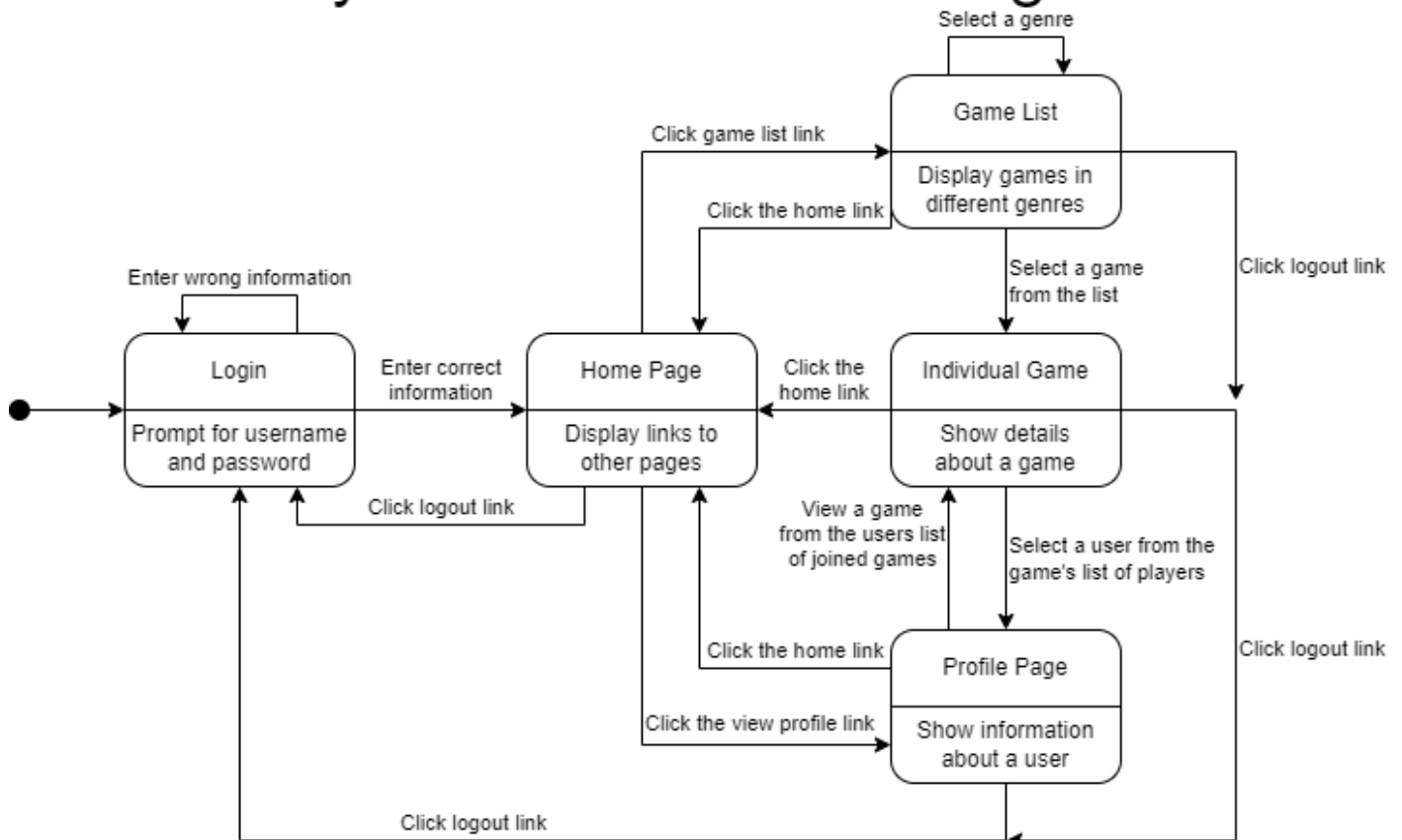


# Appendix C: Software Design

Author: Jessica Frank (All Diagrams), Alec Droegemeier (Helped with Player Diagram), Alex Wesley (Helped with UML Class Diagram)

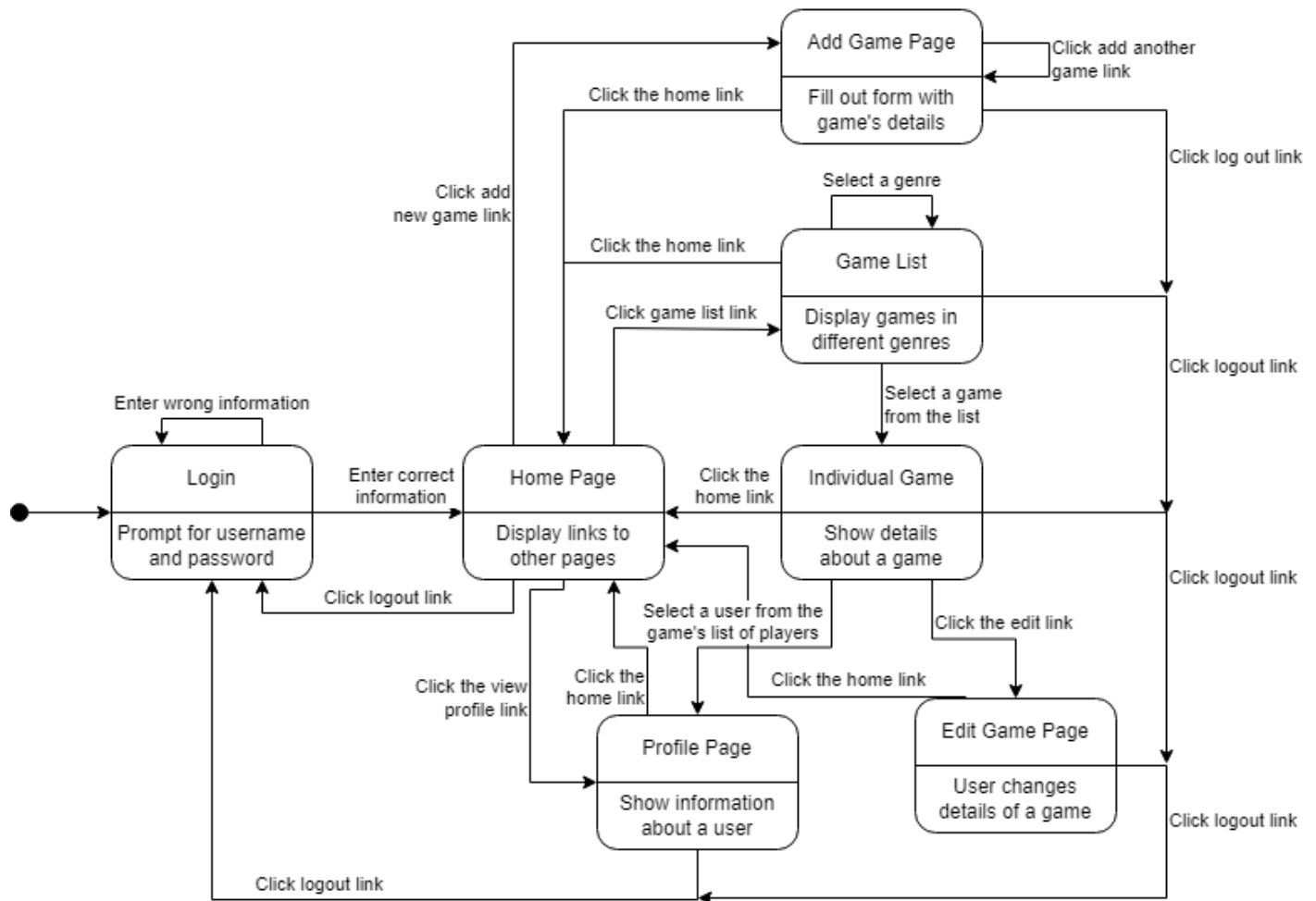
## State Machine Diagrams

### Player State Machine Diagram

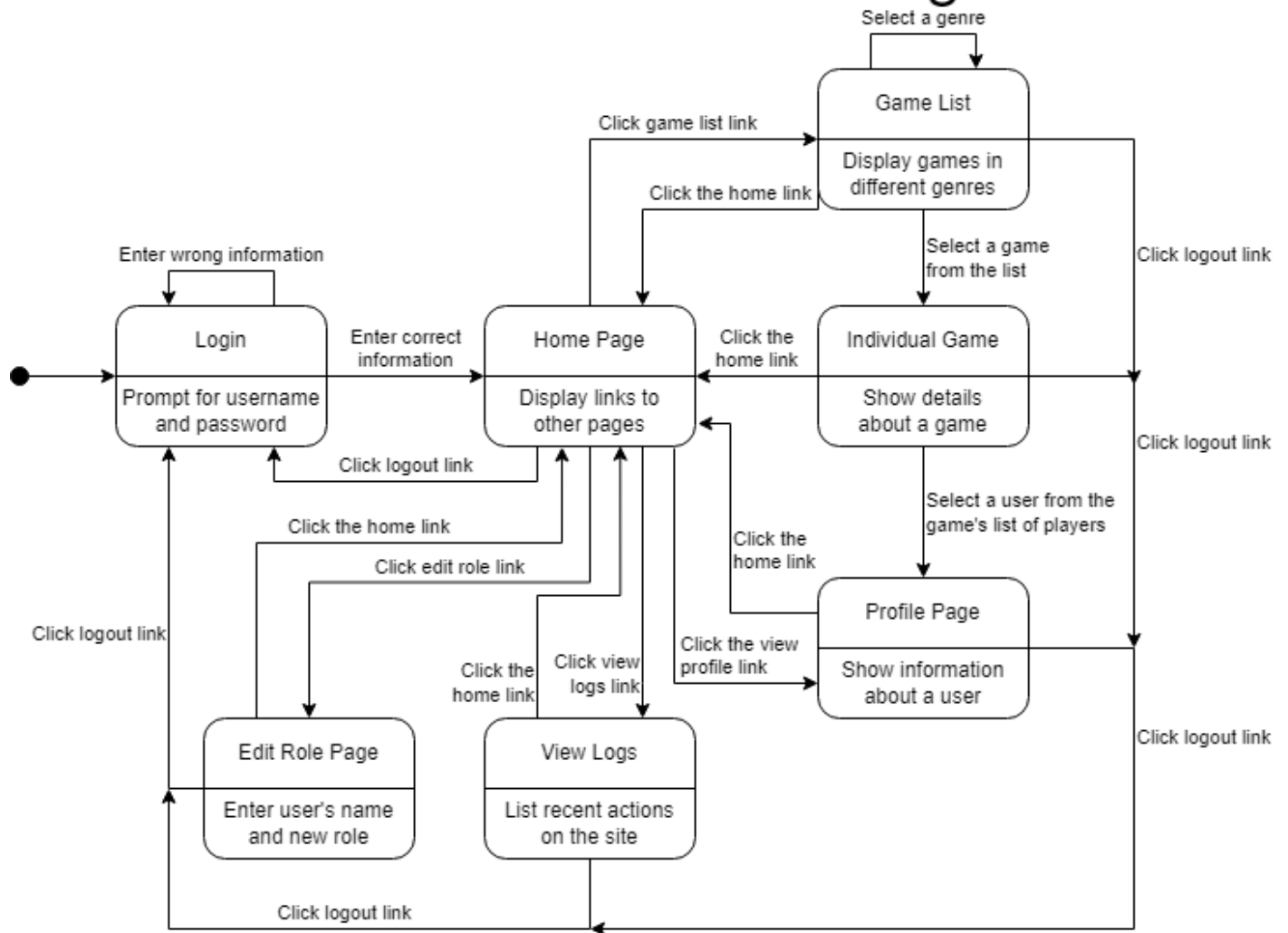




# Moderator State Machine Diagram



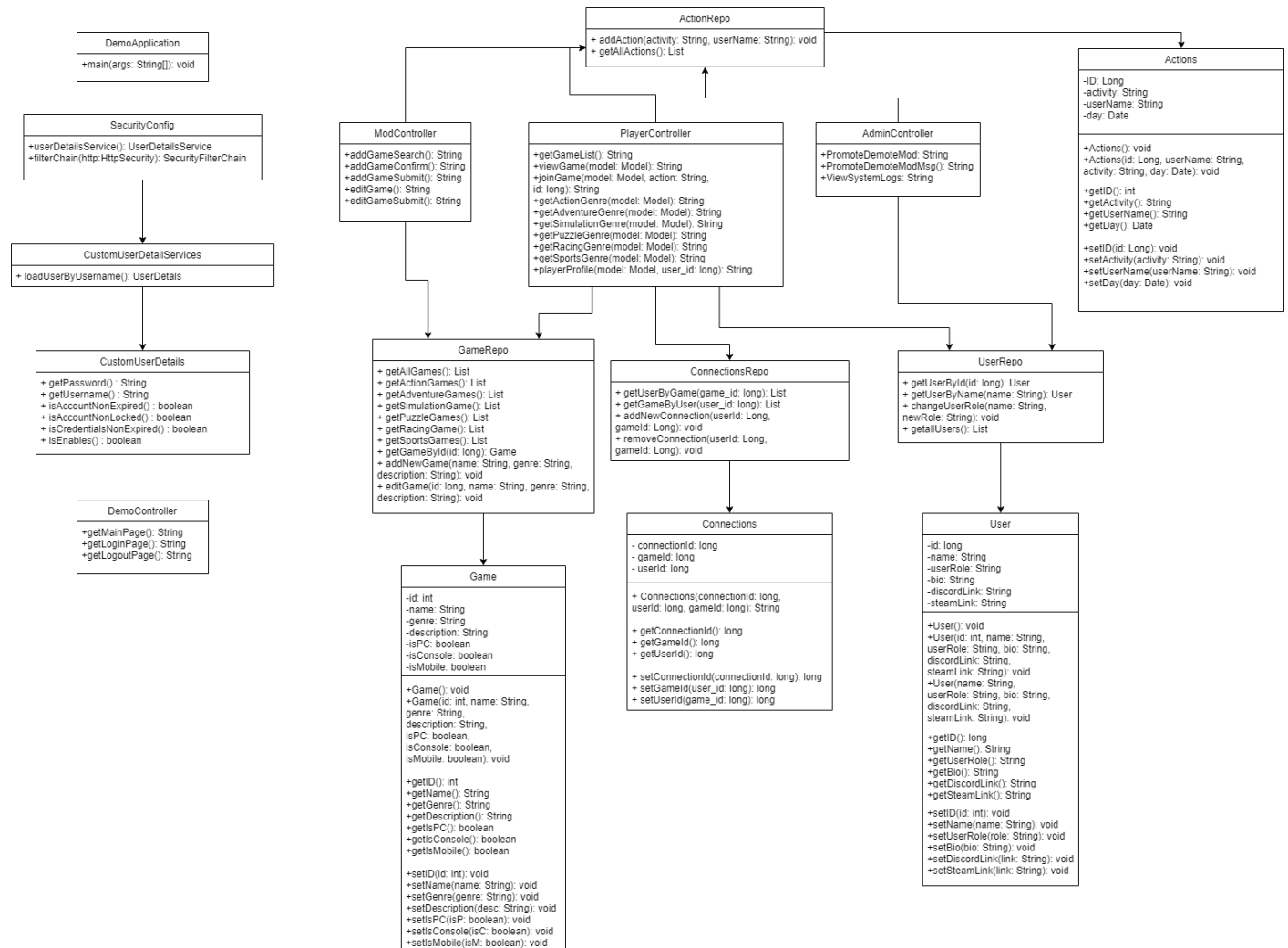
# Admin State Machine Diagram



# UML Class Diagram

Note:

All controller class methods that return a String are used by Spring to display a page on the website. Because UML Class diagrams are intended to show only classes, and in order to simplify this diagram, the individual HTML page files are not included below.



# Appendix D: Presentation Scenarios

*Authors: Jessica Frank (Moderator Scenarios), Alex Wesley (Admin Scenarios), Alec Droegemeier (Player Scenarios and Screenshots)*

Since a user's role can be changed by an admin, different users will be referred to using different numbers. For example, the moderator named M2 does not indicate that there are two moderators, and is only used to show that they are a different user than the player P1.

## Player: Log In

1. Player P1 navigates to the web page
2. P1 logs in with valid credentials

# Login

Username

Alec

Password

••••

LOGIN

## Player: View Game List

1. P1 is logged in and navigates to the home page
2. P1 clicks the game list button which will redirect to the list of created games
3. (Optional) P1 can view games by genre by clicking the respective genre button

Game List			
	ADVENTURE	SIMULATION	PUZZLE RACING
	<div>Rocket League</div> <div>High speed car soccer!</div> <div>VIEW GAME DETAILS</div>	<div>Terraria</div> <div>2D Platformer where you can build your own legacy!</div> <div>VIEW GAME DETAILS</div>	<div>Valorant</div> <div>5v5 Tactical FPS with abilities</div> <div>VIEW GAME DETAILS</div>

## Player: View Game Details

1. P1 is logged in and navigate to the home page
2. P1 views the game list
3. P1 selects the view game details button on game G1 that they wish to learn more about

GAFF Home

Log Out

## Counter Strike

Description: 5v5 Tactical FPS

Available Platforms: PC Console Mobile

Genre: Action

[Edit game](#)

### Player List

JOIN GAME LEAVE GAME

Jessica

Alec

Alex

## Player: Join/Leave Game

1. P1 is logged in and navigate to the home page
2. P1 views the game list
3. P1 selects the view game details button for G1
4. P1 can select Join or Leave to connect to the G1's list of players

### Player List

JOIN GAME

LEAVE GAME

Jessica

Alec

## Player: View Profile

1. P1 is logged in and navigate to the home page
2. P1 views the game list
3. P1 selects the view game details button for G1
4. P1 can select specific users connected to G1 to view their profile
5. P1 logs out

GAFF Home

# Alec

Hi I'm Alec, nice to meet you!

Steam: <https://steamcommunity.com/id/23516134/>

Discord: discord#6762

## Joined Games

### The Witcher 3: Wild Hunt

action role-playing game with a third-person perspective.

[VIEW GAME DETAILS](#)

### Counter-Strike

5v5 Tactical FPS

[VIEW GAME DETAILS](#)

### Rocket League

Car soccer!

[VIEW GAME DETAILS](#)

## Moderator: Add New Game

1. Moderator M2 logs in.
2. M2 navigates to the add game page, and enters the name of game G2 that's not currently on the site and a valid RAWG API key. M2 submits the form.
3. M2 sees that the second page of the form has autofilled details about G2. (These details are retrieved from the RAWG API.) M2 writes a description of G2 and submits the game.
4. (Optional) M2 navigates back to the game list page to see that the game has been added.
5. M2 sees the confirmation message showing that G2 was added to the database.



# Add New Game

Game Name

Witcher 3

Moderator Key Number

.....

SUBMIT

GAFF Home

Log Out

## Add New Game

"Witcher 3" game data from RAWG API

### Details

Game name  
The Witcher 3: Wild Hunt

Action role-playing game with a third-person perspective

### Genre

- ☐ Action
- ☒ Adventure
- ☐ Simulation
- ☐ Puzzle
- ☐ Racing
- ☐ Sports

### Platforms

- ☒ PC
- ☒ Console
- ☐ Mobile

SUBMIT

# Your Game has been Added

Name: The Witcher 3: Wild Hunt

Description: action role-playing game with a third-person perspective.

Genre: Adventure

Platforms: PC Console

[Add another game](#)  
[Home](#)

## Moderator: Change Game Details

1. Since M2 is still logged in, they navigate to the home page.
2. M2 clicks the link to the game list page.
3. M2 finds game G2 in the list of games on the game list page, and clicks the link to view details about G2.
4. M2 clicks the edit game link on the G1 game details page. M2 sees that this form is automatically filled with the current details in the database for G1.
5. M2 changes the description of G1, and submits the form.
6. M2 sees the message confirming that G1 was added.
7. (Optional) M2 navigates back to the game list page to see that the game description has changed.
8. M2 logs out.

### Edit Game Details

Current details for: Counter Strike

Details	Genre	Platforms
<div>Game name Counter Strike</div> <div>Better game</div> <div></div>	<div><input checked="" type="radio"/> Action</div> <div><input type="radio"/> Adventure</div> <div><input type="radio"/> Simulation</div> <div><input type="radio"/> Puzzle</div> <div><input type="radio"/> Racing</div> <div><input type="radio"/> Sports</div>	<div><input checked="" type="checkbox"/> PC</div> <div><input checked="" type="checkbox"/> Console</div> <div><input checked="" type="checkbox"/> Mobile</div>

SUBMIT

## Your Game has been Edited

Name: Counter Strike  
Description: 5v5 Tactical FPS  
Genre: Action  
Platforms: PC Console Mobile

[Home](#)



## Admin: Views System Logs

1. Admin A3 logs in.
2. Admin A3 navigates to the system logs page and views the system logs for the app.

# Recent Activity

Activity	User	Time
Alex joined Counter Strike	PLY	2022-11-22
Jessica joined Simulation	MOD	2022-11-22
Admin left Rocket League 2	ADM	2022-11-22
Admin joined Rocket League 2	ADM	2022-11-22
Valorant was added as a(n) Action game	Moderator	2022-11-11

## Admin: Promote/Demote a User

1. Admin A3 is already logged in, so they go back to the home page.
2. Admin A3 navigates to the Promotion/Demotion page.
3. Admin A3 promotes player P1 and P1 is now a moderator.
4. Admin A3 logs out.

# Promotion and Demotion

Enter the name of the user you want to Promote or Demote

Enter user's Name

☒ Player

☐ Moderator

SUBMIT