CSC340: Software Engineering
Software Requirements Specification (SRS)

GAFF: A Gaming App for Finding Friends

10-22-2022

Version 3

By: Jessica Frank, Alec Droegemeier, Alex Wesley

Table of Contents

1. Introduction

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

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

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 allow a user to create a profile that other users can view when browsing the site. The user can choose to add a bio, and/or link to third party applications such as Steam, Discord, Twitter, etc to 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 Player, Moderator, and Administrator.

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

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.

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 wrong, an error message will be displayed. If the user's information is correct, they will be navigated to the game list screen.

Interface 3. The game list screen contains two main sections. The first is a list of selectable video game genres. 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 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. Once they enter the name, 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 need to be filled out for the game to be added to the site. 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 user permissions page is only visible to administrators. This page will contain a list of the site's users. When the administrator selects one of the users, they will see a dropdown where they can change the user's rank between Player (normal user), Moderator, and Administrator.

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

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

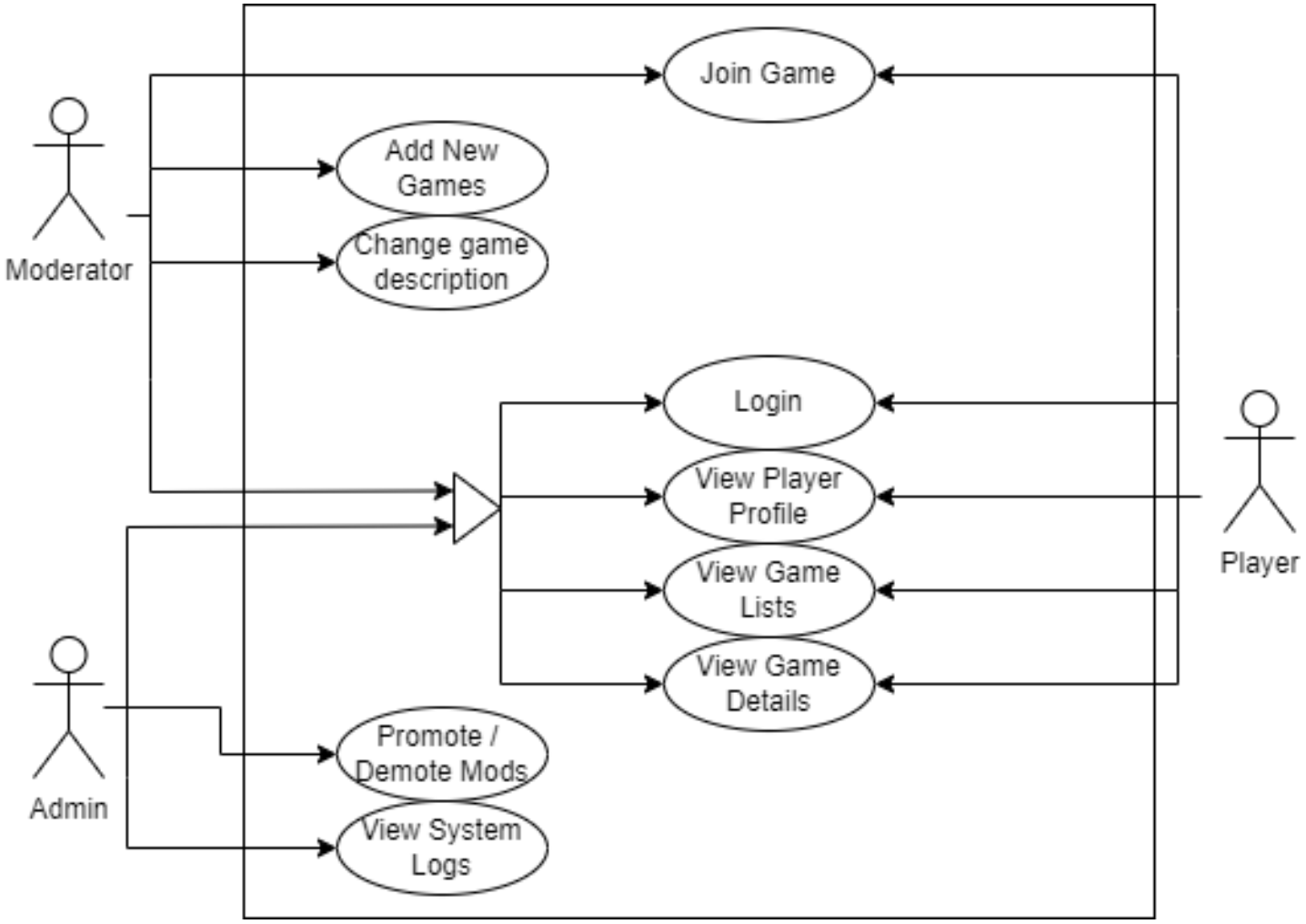
This project must be finished before the presentations at the end of the Fall 2022 semester. The application must have a working prototype by the 18th of October, and a presentation of the final product must be ready by the 15th of November.

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

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, and add it to the correct genre.

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 players and moderators 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 see a message informing them that their login information was incorrect.
iv) **Completion Status -** The user can view the different sections of the site that they can now 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.
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 chooses 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 there are links to third party applications such as Steam or Discord.
iii) **Possible Error -** Viewed the wrong profile.
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 games list of players.

iii) **Possible Error -** The player could join a game that they did not mean to, but there is also a button to disconnect from the game if the player does not want to be a part of that list.

iv) **Completion Status -** The player was successfully added to the list of players for that game.

Use-Case Responsibilities

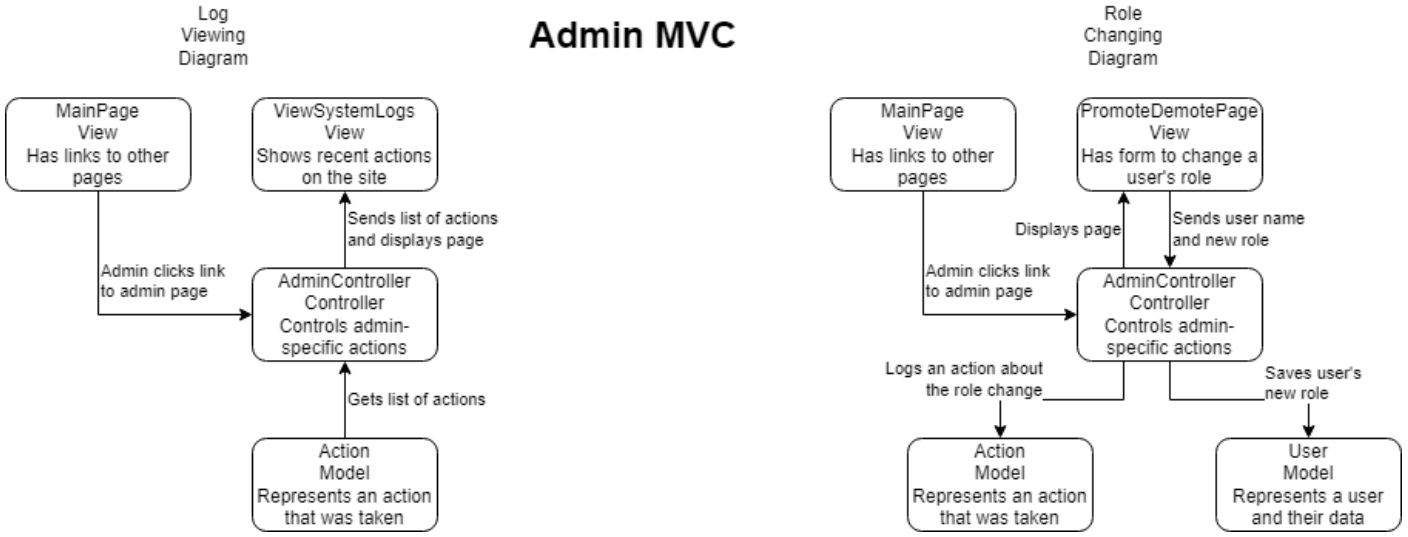
Jessica Frank	Alec Droegemeier	Alex Wesley
Login	View Game Lists	View System Logs
Add New Games	View Game Details	Promoting/Demoting Moderators
Change Game Description	View Player Profile	
	Join Game	

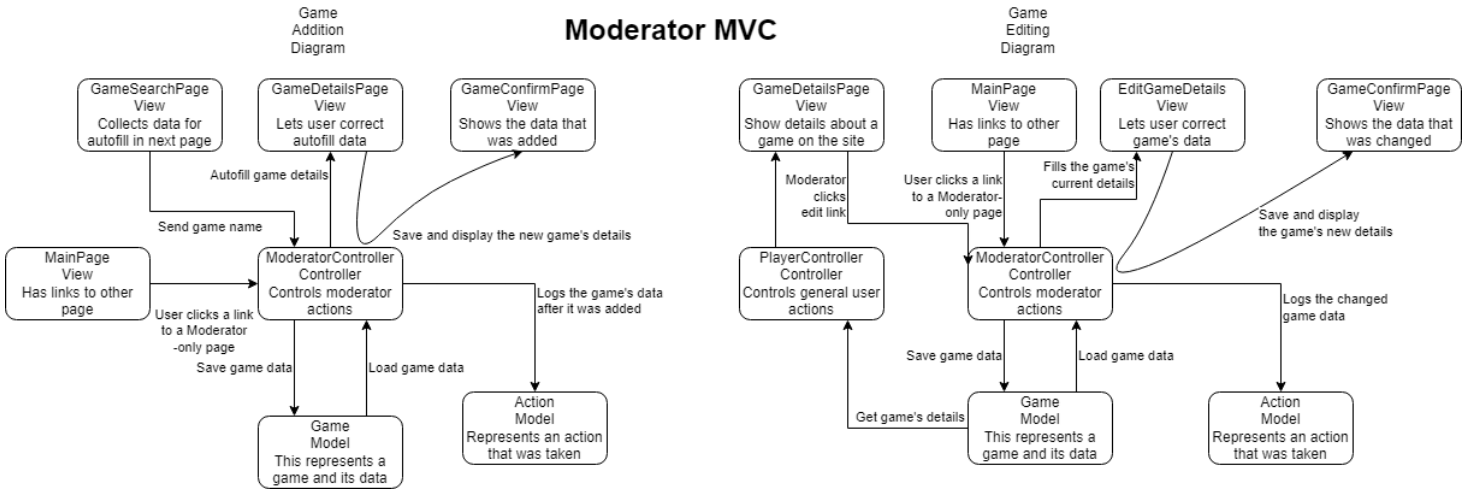
Appendix B: Software Architecture

Architecture Diagrams

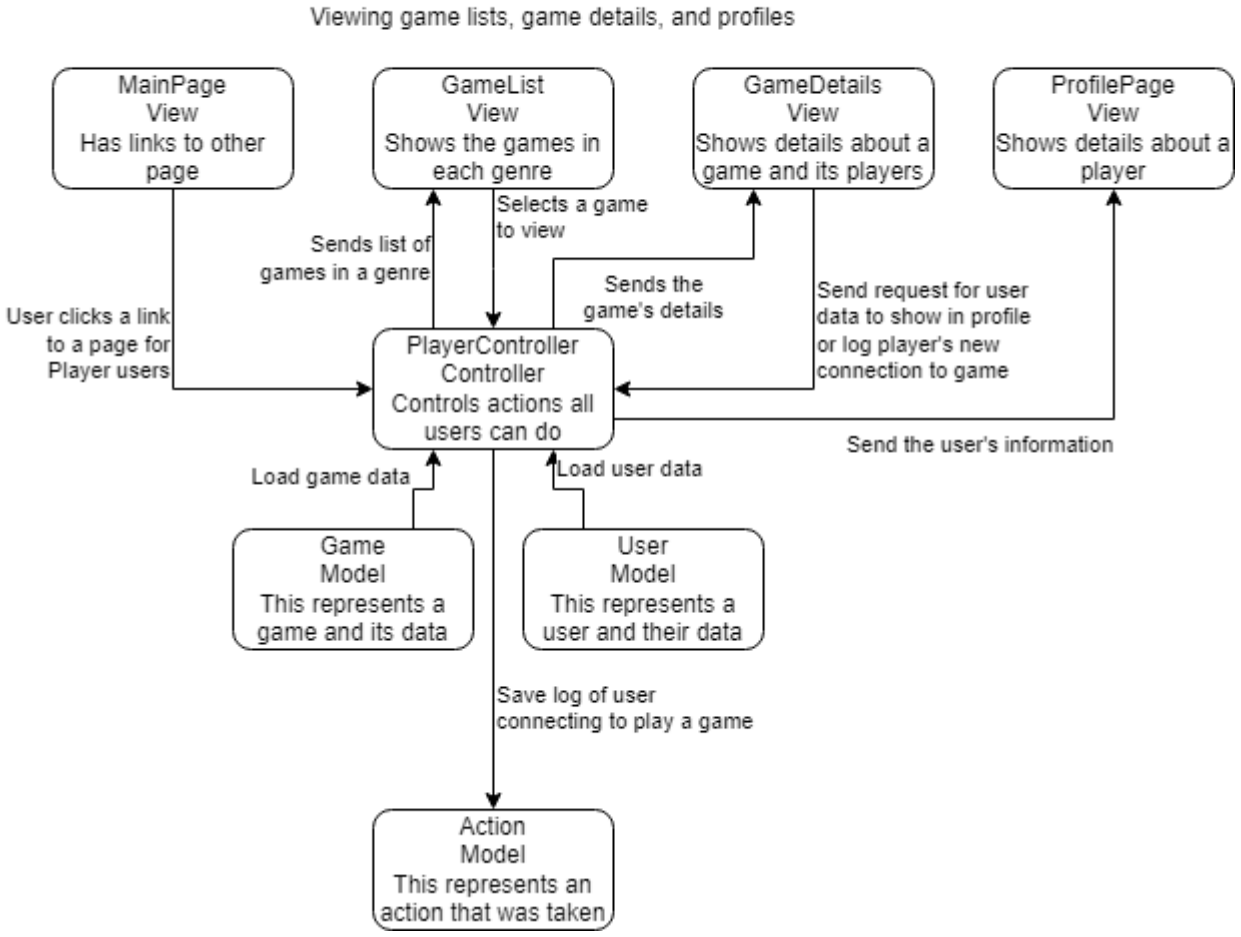
Since we were not given an example of how the architecture diagrams should be written, the MVC diagram below has been broken down into sections for different actors and use cases to improve readability.

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.

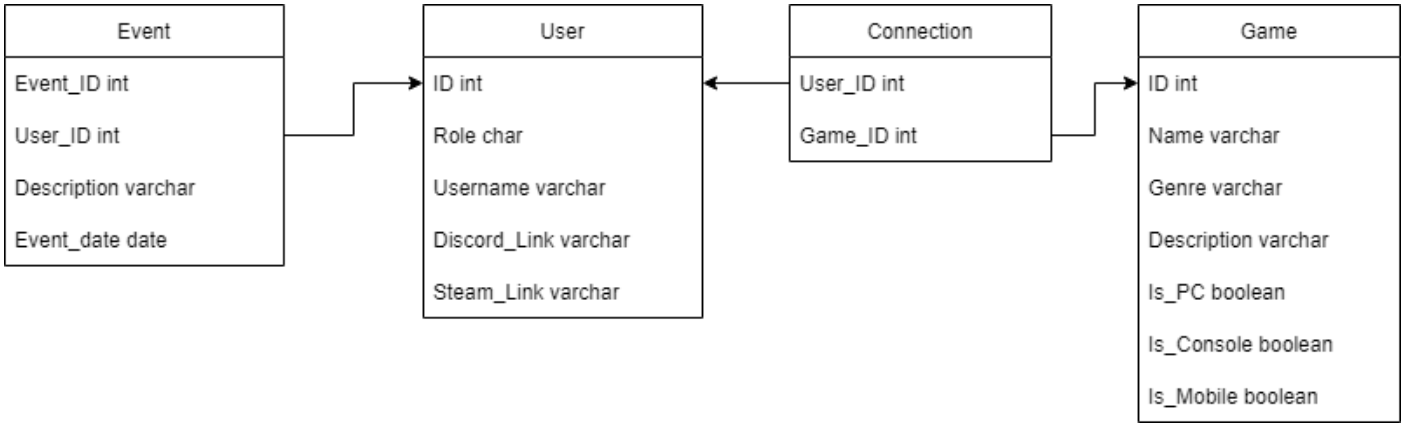




Player MVC

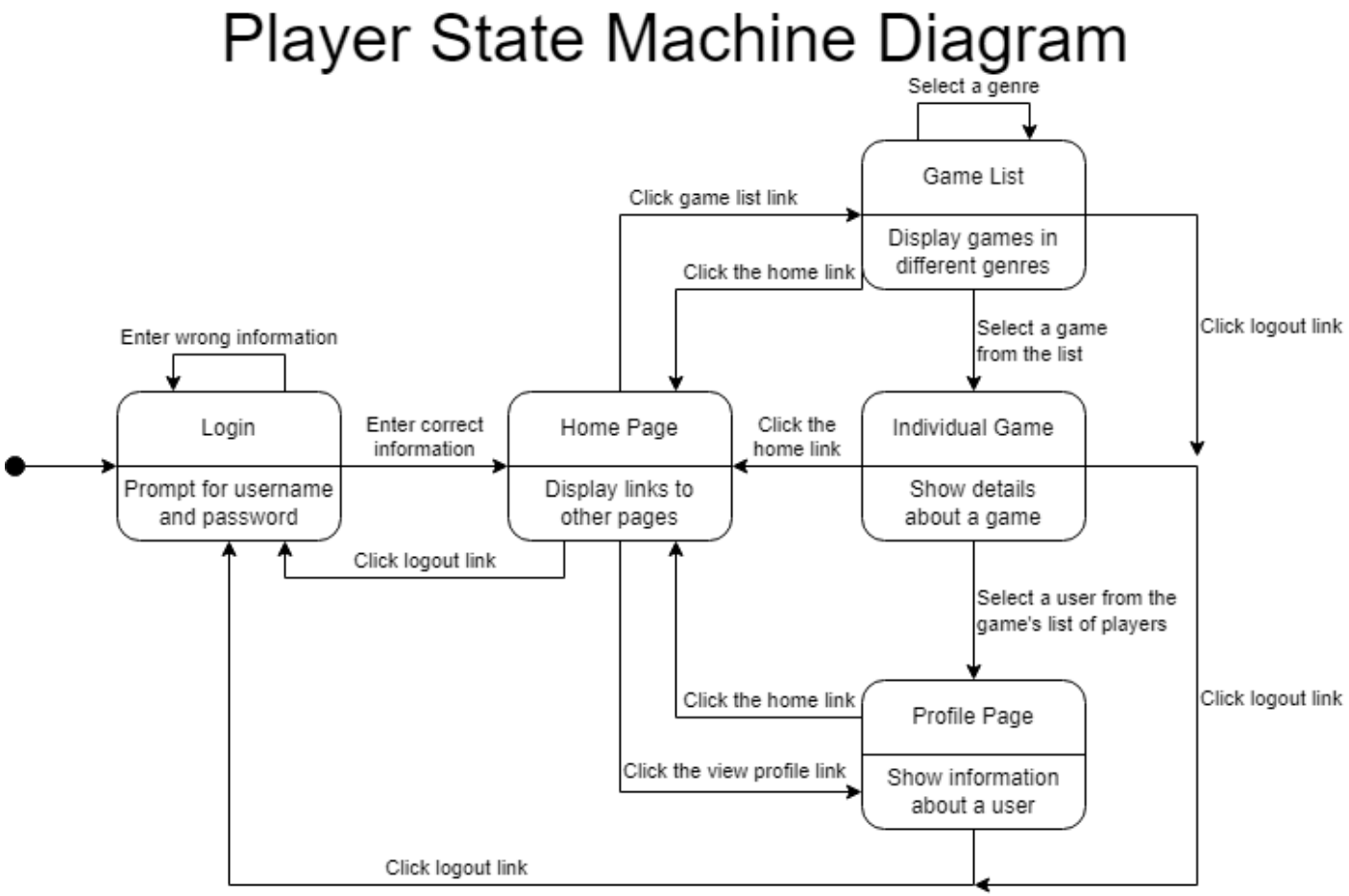


Database Relational Schema

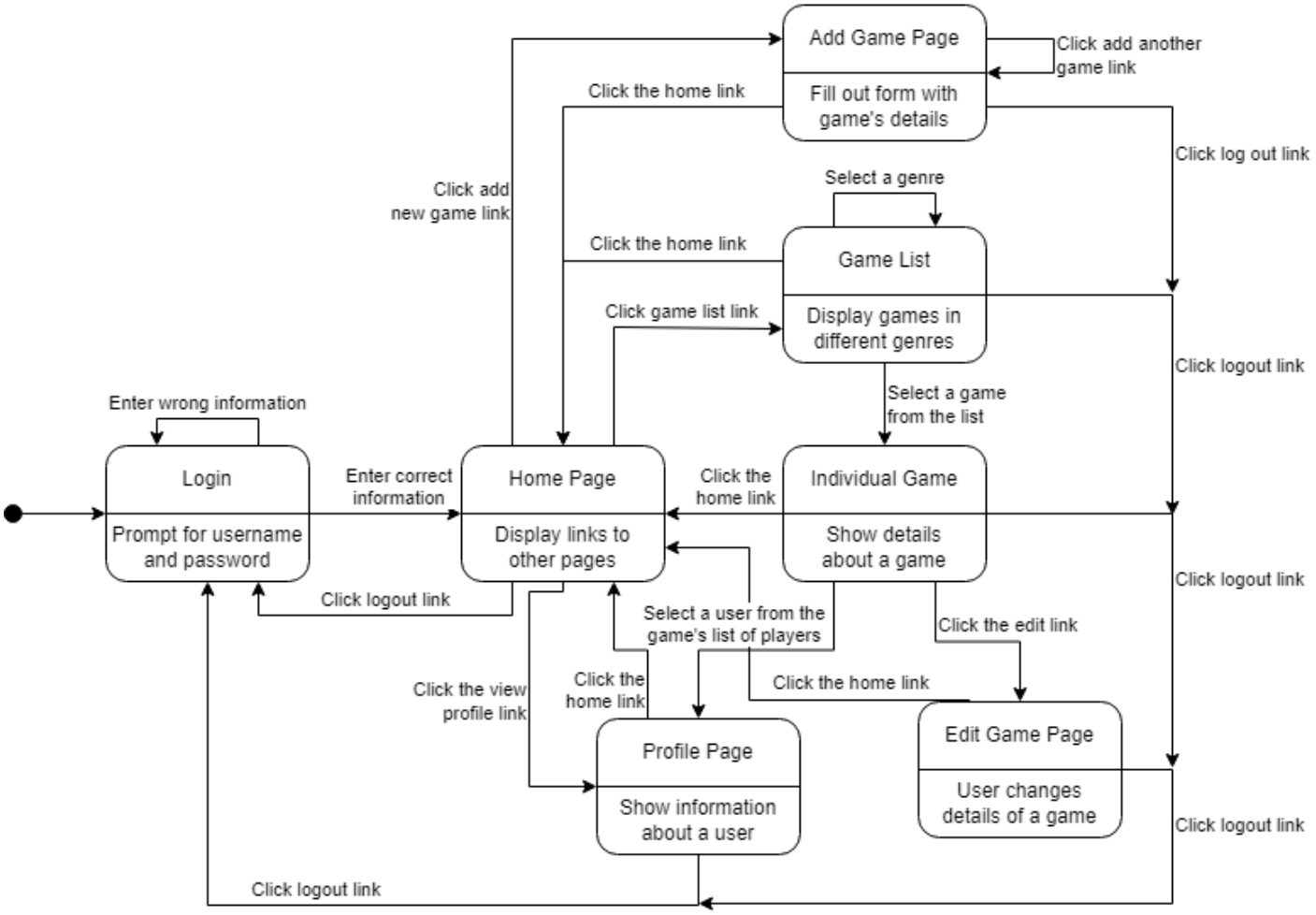


Appendix C: Software Design

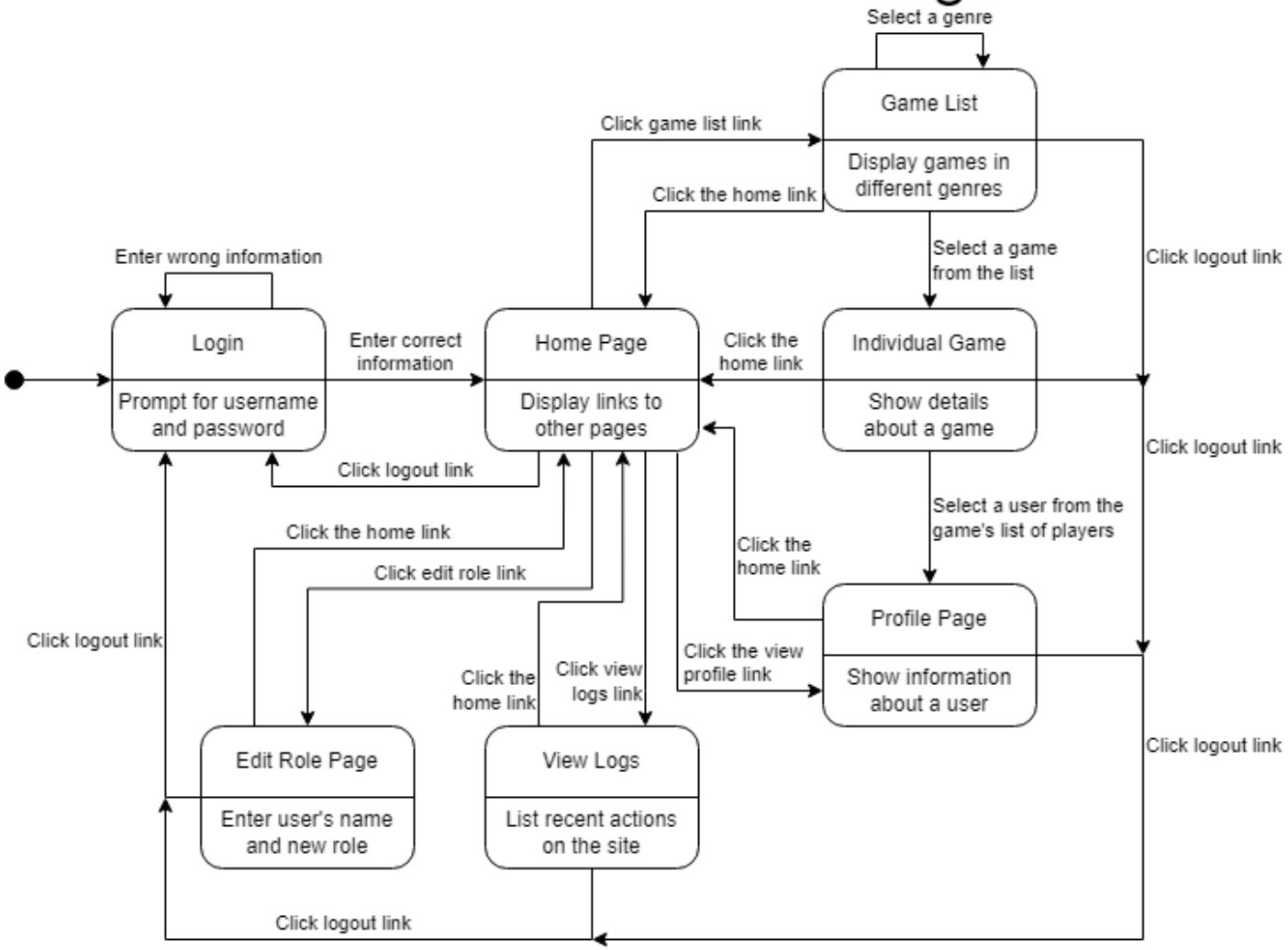
State Machine Diagrams



Moderator State Machine Diagram



Admin State Machine Diagram



UML Class Diagram

Notes:

1. 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.
2. This diagram represents the current plan for our site's classes. Depending on the time this document is viewed, the diagram below may not represent the current code since several of the classes are still being developed.