

Software Requirement Specification

# Venom



Fall 2023

Version 1.0

Joshua Zhu (Project Manager)

Anthony Arocha (Assistant Project Manager)

Nathaniel Wallace (Software Engineer)

Juan Trujillo (Software Engineer)

Armando Magallon (Software Engineer)

Marissa Bonilla (Software Engineer)

Mario Perez (Software Engineer)

## Table of Contents

1. Introduction .....	3
1.1. Purpose .....	3
1.2. Scope .....	3
1.3. Definitions, Acronyms, and Abbreviations .....	4
1.4. References .....	4
1.5. Overview .....	4
2. Overall Description .....	5
2.1. Product Perspectives .....	5
2.1.1. System Interfaces (Deployment Diagram) .....	5
2.1.2. User Interfaces .....	5
2.1.3. Software Interfaces .....	6
2.1.4. Memory .....	7
2.1.5. Operation .....	7
2.2. Product Function (Case Diagram) .....	7
2.3. User Characteristics .....	8
2.4. Constraints .....	8
2.5. Assumptions and Dependencies .....	8
3. Specific Requirements .....	9
3.1. External Interface Requirements .....	9
3.1.1. User Interfaces .....	9
3.1.2. Hardware Interfaces .....	9
3.1.3. Software Interfaces .....	9
3.1.4. Communication Interfaces .....	9
3.2. Functional Requirements .....	9
3.2.1. Homepage .....	10
3.2.2. Login .....	10
3.2.3. Reset Your Password .....	10
3.2.4. Create Account .....	11
3.2.5. About Us .....	11
3.2.6. Search Results .....	12
3.2.7. Game Page .....	12
3.2.8. Accounts Page .....	13
3.2.9. Inbox Page .....	13
3.2.10. Game Tracker Page .....	13
3.2.11. Comments .....	13
3.3. Performance Requirements .....	13
3.4. Design Constraints .....	14
3.5. Software System Attributes .....	14
3.5.1. Reliability .....	14
3.5.2. Maintainability .....	14
3.5.3. Ease of Use .....	14

# 1. Introduction

## 1.1. Purpose

The purpose of this Software Requirement Specification is to outline the requirements and goals of the Venom game hosting web application. The requirements and goals in this document reflect those discussed with the client Jennifer Jin. These requirements and goals will have been met and tested by the end of the Fall semester of 2023 for the CSE-4550 course.

## 1.2 Scope

The Venom game hosting web application allows users to access a wide range of games that can be played in the browser they currently are using. Regular users will be able to create an account where they can favorite their like games, keep their progress and high scores in others, and comment on games that they like. Users who want to upload their own game creations will be considered as developers and will have the ability to upload their games onto the web application, as well as track how much traffic the game will receive. Finally, to maintain control and moderate the web application, there will be web application administrators who will approve of the games that are being uploaded as well as moderate the comment sections.

Prototype 1: The first prototype will consist of a web application that allows users to create their own accounts, a functioning home, game and about us page as well as a search bar to help search for specific games. The web application will also have a way to upload games, onto the web application.

Prototype 2: The second prototype will consist of the actual accounts and the functionality of those accounts. The player accounts will be able to favorite their games, comment and keep their high scores, while the developer accounts will be able to upload their games and monitor the traffic each games receive. Meanwhile, the admin accounts will be able to approve of the uploaded games and moderate the comment sections of the games.

## 1.3 Definitions, Acronyms, and Abbreviations

Algorithm: A finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation.

Cloud Firestore: A flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform.

Computer: A machine that can be instructed to carry out sequences of arithmetic or logical operations automatically via computer programming. These include desktops and laptops.

CSS: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

Firebase: A platform developed by Google for creating mobile and web applications.

Homepage: The primary page of the web application.

HTML: Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.

HTTP: Hypertext Transfer Protocol. A session layer protocol for distributed, collaborative, hypermedia information systems.

Internet Browser: A software application for accessing information on the World Wide Web.

Javascript: A programming language that conforms to the ECMAScript specification.

Real-time Web Database: A database system that uses real-time processing to handle workloads whose state is constantly changing.

SRS: This document, the Software Requirements Specification (SRS), that illustrates the functional and non-functional requirements of the 4-Wheel Drive Used Car Dealership web application.

UID: Unique Identifier used by Firebase Authentication. Each user account created is given a UID.

User: Someone who uses the web application.

User Credentials: A user's username and password used to sign into their account.

Web Application: An application software that runs on a web server. Web applications are accessed by the user through a web browser with an active internet connection. The term will also be used synonymously with website.

## 1.4 References

IEEE Software Engineering Standard Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specification", October 20, 1998.

"Welcome to Flask." *Welcome to Flask - Flask Documentation (2.3.x)*, flask.palletsprojects.com/en/2.3.x/. Accessed 30 Sept. 2023.

Github

Firebase

## 1.5 Overview

The following information provided in this document is intended to provide information about the web application as well as discuss in depth about the details of use, functionality, and any future plans for the Venom Game Hosting web application.

The second section of this document will give an overview of the website and how it works by explaining the integrated software system and the constraints of said system.

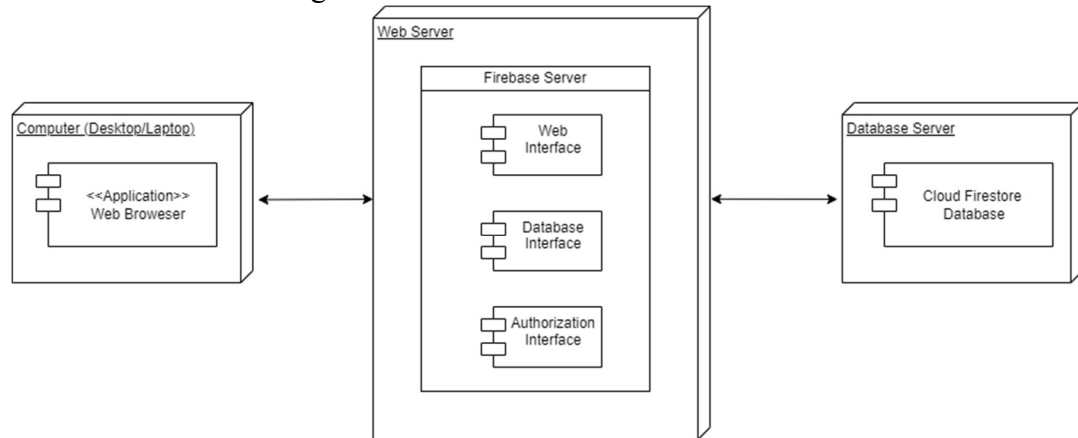
The third section will go into detail about the requirements to run the website. It will provide the specific components required to run the software and the website.

## 2. Overall Description

### 2.1. Product Perspective

#### 2.1.1. System interfaces (Deployment Diagram)

The major components of this website include the Firebase web server that contains all of the web, database and authorization interface, the Cloud Firestore Database which will provide a real-time database, and finally the computer that the user themselves will be using.



#### 2.1.2. User Interfaces

##### Guest/Signed Out User

Once arriving at the Venom Game Hosting website, users will automatically be placed on the home page. The home page will display the logo on the top left of the web page, as well as buttons to lead to different web pages, such as different game genres, an About Us page, the Login/Signup pages, as well as a search bar for users who want to find a specific game. Underneath the header, the rest of the page will be filled out with game hyperlinks, that will direct users to the individual game pages, in a list format. For each game genre that is listed, the individual genre pages will have the games in list form similar to how the home page games are formatted. The About Us page will display information about our company, our values and business practices and contact information. The search bar will be there if a user wants to find a specific game. Currently it is hashed out to only return the exact game that it is told to find, however later on can be modified to include similar search results. The Login and Signup pages are individually different, but take in similar information. The Login page will display the company's logo as well as a field for the username/phone number and password. There will also be a link to a password reset page if the user is to forget their password and/or wants to reset it. It will also have a hyperlink that will link the user to the Signup page if they do not have an

existing account. The Signup page will have a similar layout as the Login page, and will contain fields to fill out their information such as email, password and phone number. The Signup page is also where new users will tell us if they want a developer account with access to upload their own games.

#### Signed In – Player Account

The player account will have all the same features as a guest user, but will have an additional Account Settings page. On the Account Settings page, they will be able to see game that they have favorited, as well as change their password, and change their avatars. As a signed in player, the user will also be allowed to comment on games that they have/are playing so they can leave feedback for the developers or just commenting on how much they enjoy the game.

#### Signed In – Developer Account

Developers will have all the same amenities as the player account, meaning that they can still play games, set favorites and leave comments, but with additional abilities that help them with uploading their games. The main feature would be to upload their games onto the website, but also they would be able to monitor the traffic on each individual game to see which one is doing the best. As there is a procedure to uploading the game, developers may need to discuss issues with administrators, so they will also have an inbox so that communication can happen between the two parties.

#### Signed In – Administrator Account

Administrator accounts will have the ability to delete and moderate the comment section of each game as well as approve whether a game can be uploaded onto the website. As stated before, they are in charge of approving upload requests, so they also have an inbox so they can communicate with the developers.

### 2.1.3. Software Interfaces

Visual Studio Code is used to work on the web application's source code. Firebase Server is used to handle HTTP requests sent from a user's computer. The Firebase Server runs using an Occupancy Web Server which includes two interfaces:

Authentication - which stores users emails, usernames, and passwords and assigns each created account with a UID.

Cloud Firestore Database - a real-time web database which stores and organizes user and vehicle information

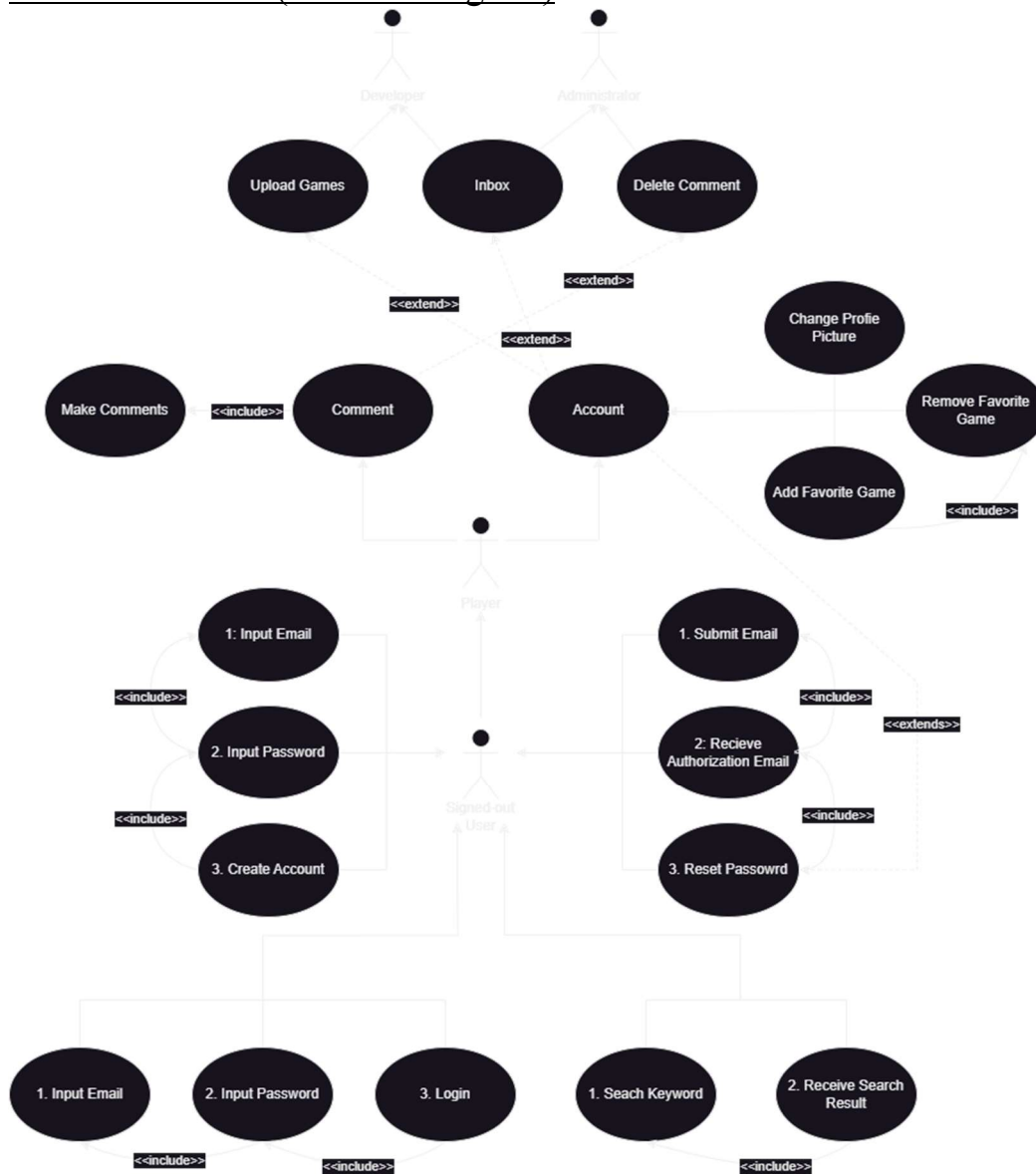
#### 2.1.4. Memory

The Firebase Server limits hosted web applications to 1GB of storage, 10GB downloads, and 100 simultaneous connections. This limitation is a result of Firebase Server being a free service for anyone to use.

#### 2.1.5. Operation

The Firebase Server will be required to be active during all activities performed on the web application. Firebase is a service provided by Google Inc. that runs active servers 24/7 with adequate data protection.

### 2.2. Product Function (use case diagram)



This web application will allow users to interact with the game hosting website by playing, searching, and commenting on games. In addition Developers will be able to upload games, and Administrators can delete comments and approve uploads.

### 2.3. User Characteristics

The use case diagram shows that users will be able to play games and navigate through most of the website's pages. Users who want to search for games can use the search bar to find the games they want, then navigate to the respective game page and play the game. Signed-out users will not be able to comment on games like signed in users can, but they will be able to create their own accounts by navigating to the sign up page. By inputting their email, phone number and password, they can create an account, and also use that information to sign in next time. Users who are classified as developers can also upload their games onto the website once it has been approved by the administrators, and can keep in contact through the inboxes in their accounts. The intended demographic of the Venom Game Hosting website is for all people who wish to play games without having to download it onto their computer. It would be a quick and easy way for users, old and young, to navigate through the pages and play the games they want.

### 2.4. Constraints

Some issues are that we need to learn new technologies such as using the Firebase Server, Cloud Firestore Database, and some team members using a Git repository for the first time.

### 2.5. Assumptions and Dependencies

We will assume that the user will have a working computer and internet when trying to access the website. They will also have to be competent at navigating to our website. Lastly, because the website heavily relies on Firebase and Cloud Firestore, we assume that both will be functional at all times.



## 3. Specific Requirements

### 3.1. External Interface Requirements

#### 3.1.1. User interfaces

The user interface is made up of graphical elements and texts on the web application. A more in-depth showing of each respective page and it's interface will be in Section 3.2.

#### 3.1.2. Hardware Interfaces

The user will need to have access to a computer with a monitor, keyboard, and mouse to be able to interact with the web application and the games on it.

#### 3.1.3. Software Interfaces

The Firebase Server will communicate with the Venom website to get and save the information from the individual users by using the Cloud Firestore's database.

#### 3.1.4. Communication Interfaces

The Firebase Server must be able to communicate with the individual user's web browser and computer to be able to send and receive the necessary information.

### 3.2. Function Requirements

To keep in mind, as it is still the early stages of development, these screenshots are subject to change. As you will see, the theme and color choice are still in discussion but the following are the functions that will definitely be present at the end of the project and be fully functioning. There will also be hand drawn pictures. These are the layouts and blueprints of what we plan on making that specific page look like. Some sections unfortunately will not have any pictures due to this being early stages, however as mentioned before, they will be updated as we move along.

### 3.2.1. Homepage

If a user is not logged in then the web application's, homepage will be the first to load.



Figure 1: Homepage

### 3.2.2. Login

If the user clicks on the “Login” button in Figure 1, they will be brought to the Login Screen (as shown below). This will allow the user to sign into their account using their email or phone number and their password.

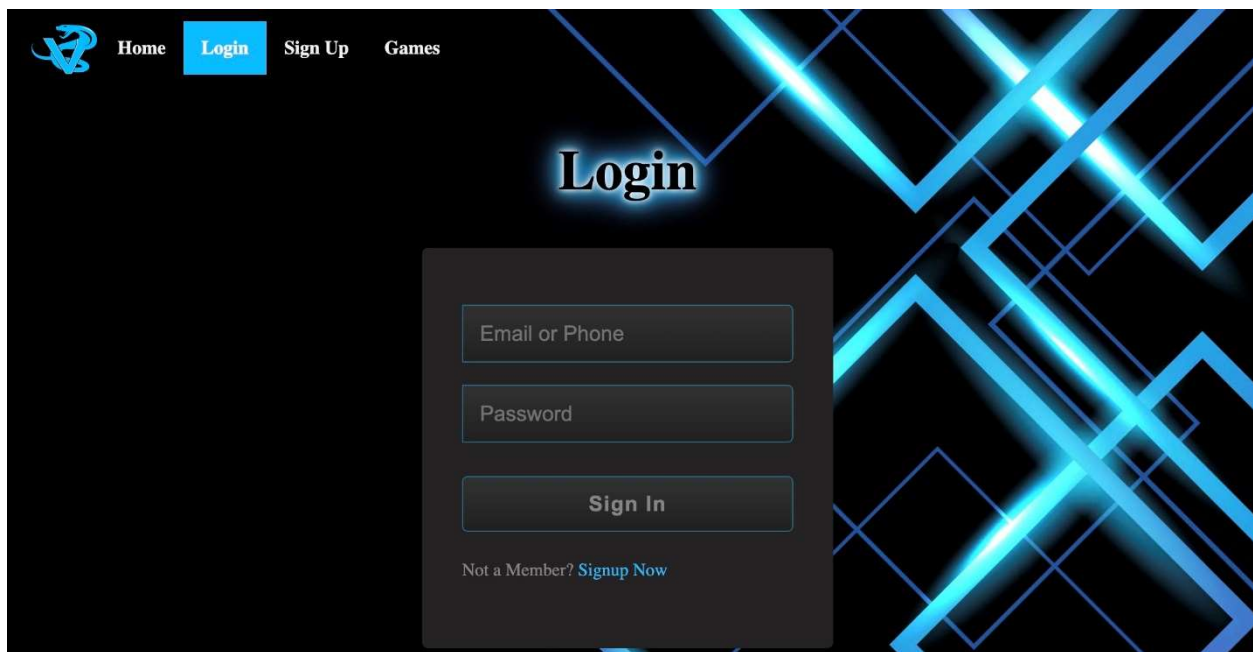


Figure 2: Login

### 3.2.3. Reset Your Password

If the user were to forget their password, they can click on the “Forgot your password?” link. This will bring the user to the Reset Your Password screen where they can submit their email. A verification email will be sent to the user and then through this, the user can create their new password.

### 3.2.4. Create Account

If the user does not already have an account, they can click on the “Signup now” link in Figure 2 or the “Sign Up” button in Figure 1 & 2. This will bring the users to the sign up screen where they can submit their email, phone number, and their other information. This is also where users can identify if they are a developer or not.

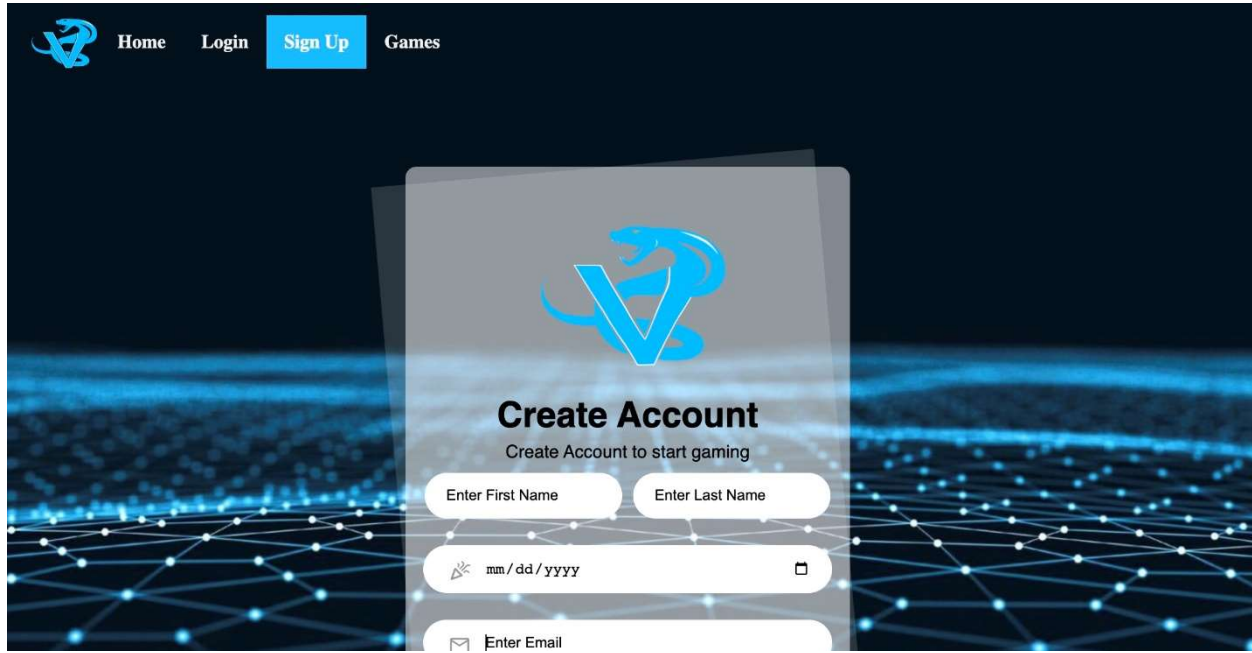


Figure 3: Sign up page

### 3.2.5. About Us

When the user clicks on the “About Us” link, the user will be brought to the About us page. This page displays information about Venom as a game hosting website and company.

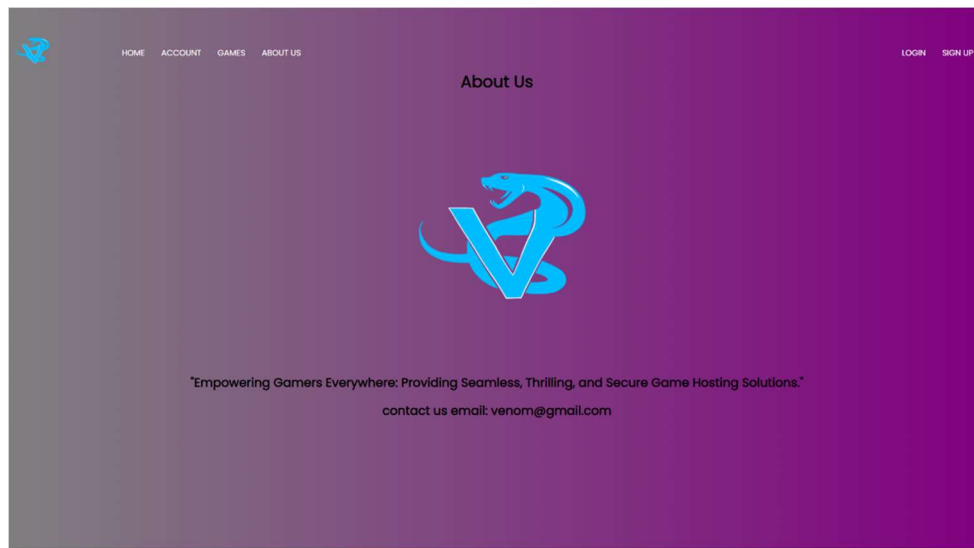


Figure 4: About Us Page

### 3.2.6. Search Results

Once the user inputs a game title into the search bar, the page will display the game that is requested and the user can click on the link to start playing the game on the game page.

### 3.2.7. Game Page

The game page is the most important page on the web application as it is where most of the users will be spending their time on. On this page, the user will be able to play their game in a windowed format and have instructions and comments below the window. Towards the side of the window will also be a section for us to put the game suggestions or the most recent games uploaded.



Figure 5: Game Page

### 3.2.8. Accounts Page

The accounts page will be different depending on the type of account; however the overall structure will not change. Users will be able to change their profile pictures and have access to different things like favorite games, inbox, or game trackers.

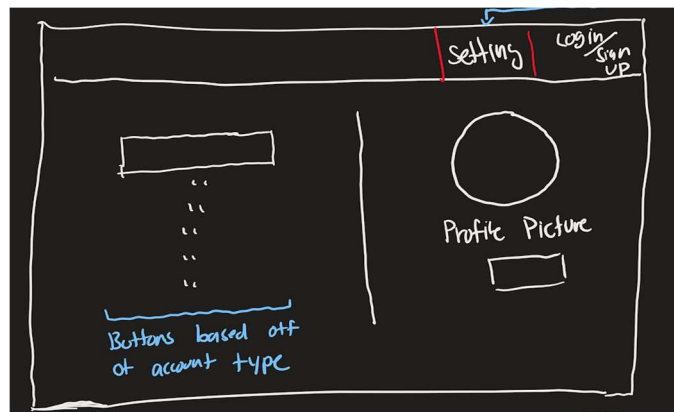


Figure 6: Accounts Page

### 3.2.9. Add Game Page

This page will be a section of the developer's account page. This page's function is to allow the developer upload their games for the administrators to review before it is implemented onto the website.

### 3.2.10. Inbox Page

This page is a section of both the developer and administrator account pages. The page is to allow communication to happen between the developer and administrator, especially when the developer is trying to upload of their games.

### 3.2.11. Game Tracker Page

This page is a section of the developer to track the amount of traffic that their game gets. This will provide vital information to the developer to see what kind of game the audience likes but also make changes if need be.

### 3.2.12. Comments

This last required function is not an actual page, but will be a part of every game page throughout the website. This functions allows the players to not only communicate on the game but also communicate to the developers uploading the game. The administrator will moderate the comment sections and will have the ability to delete comments as they see fit.

## 3.3. Performance Requirements

The Firebase server must be able to update the player's information in real time as to provide the users with up to date information about their games, comment, and account settings.

Additionally, the search algorithm that is implemented must work in a timely manner and provide the correct results.

### 3.4. Design Constraints

The user's computer must have an internet browser and be connect to the internet in order to access and use our web application. The internet will be used to send and receive information from the Cloud Firestore database.

### 3.5. **Software System Attributes**

#### 3.5.1. Reliability

Multiple measures will be take in order to ensure that the web application is reliable and secure. As long as the web application is reliable, it allows the user to have a smooth experience using the application. Secure our application is also important as we would have access to users' email information.

#### 3.5.2. Maintainability

The software provided will also have documentation attached to provide solid and simple maintainability and the ability to make modifications. Future features can be added, so the development has allowed for such areas and capabilities in the application to be had.

#### 3.5.3. Ease of Use

The web application is designed to be easy and intuitive for all users to navigate. Users will easily be able to go from one page to another as well as backtrack if necessary. Regardless of what happens, the user will always have access to the home bar which will have the major pages on it, as well as the home page. This will make sure that no user will ever be lost in the path of different pages.