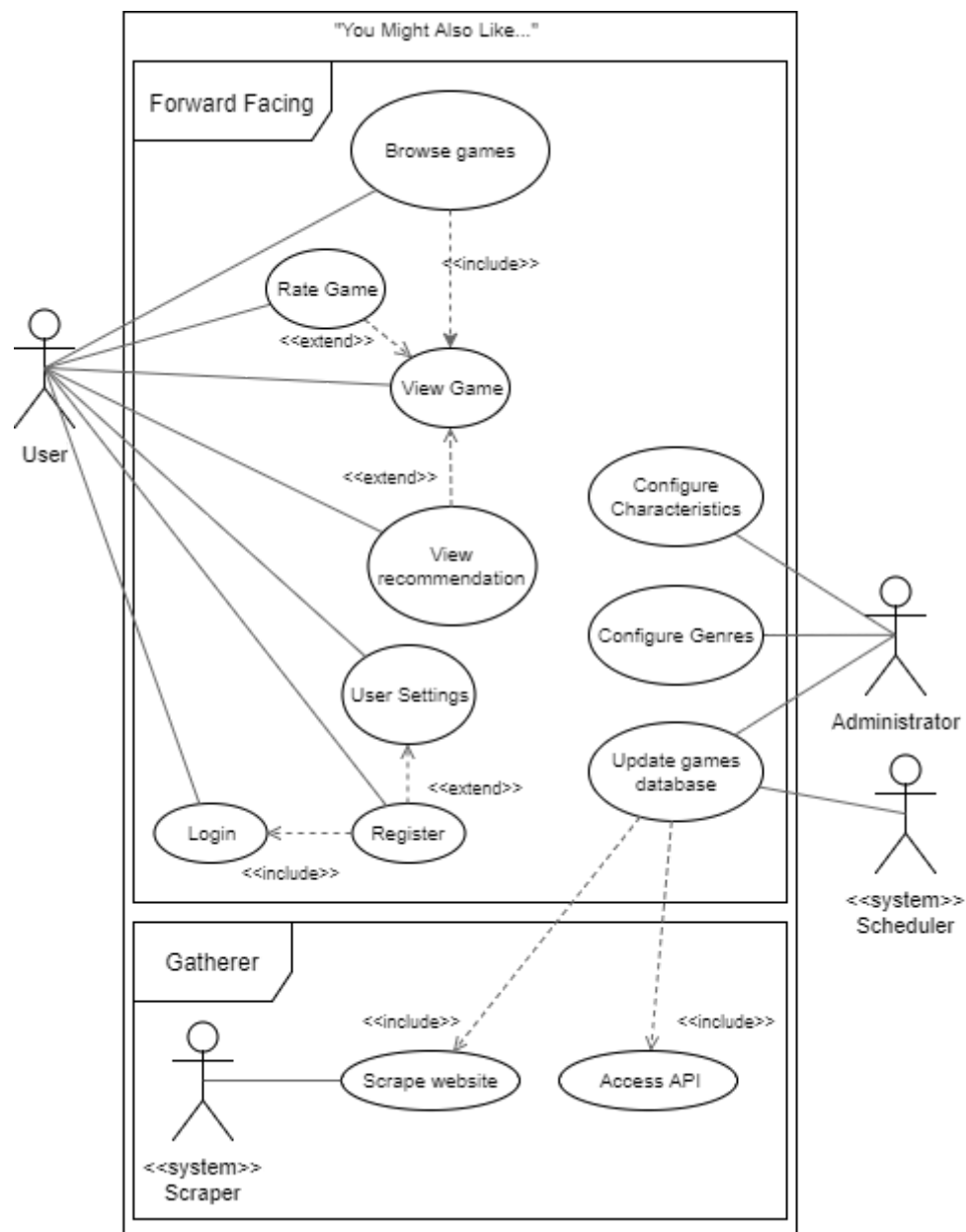# Requirement Model

"You Might Also Like…"

# Use-Case Model

# Use-Case Descriptions

## 1.1. Browse Games

When User
Wants to view the games within the system with the option of filtering or sorting the list - they visit the "browse" page
So that they can see the variety of games that are registered with the system or can see details on a specific game including description, current ratings, posts from other gamers in regards to their feedback to the game, can also rate a game. Further implementation could include ability to purchase a game.
Includes use-case 1.2. (View Game)

## 1.2. View Game

When User
Wants to view a singular game they visit the "view" page
So that they can can view a single game with more detail than is given on the browse page

## 1.3. Rate Game

When <u>authenticated</u> User
Wants to rate a game within the system they visit the view game page
So that the system has knowledge of what kinds of games the User is interested in
Extends use-case 1.3. (View Game)

## 1.4. View Recommendation

When <u>authenticated</u> User
Wants to view a game the system recommends they might enjoy
So that the user can view the current/next game the system recommends to them based on their preferences and how they and other users have rated games and will suggest games that they have not rated which the user may enjoy.
Extends use-case 1.3. (View Game)

## 1.5. Register

When User
Wants to register an account for the service they click register, and fill out the requested information
So that the User is granted an account and the system has the required information about them
Extends use-case 1.6. (User Settings)
Includes use-case 1.7. (Login)

## 1.6. User Settings

When <u>authenticated</u> User
Wants to update their settings/account information they visit the settings page and alter any information/settings they wish - such as genre and game characteristic preferences
So that the User's settings/info is updated and the system reflects this

## 1.7. Login

When User
Wants to login to the system they click login and enter their account details
So that the User is authenticated and can perform actions that require authentication

## 2.1. Update Games Database

When the Scheduler
Triggers based on settings it begins the process of updating the games database
**OR**
When an Administrator
Wants to manually trigger the process of updating the games database they visit the administrator page and trigger the update

So that the system will begin launching scrapers and accessing APIs to grab game lists from external sources
Includes use-cases 2.2. and 2.3. (Scrape Website and Access API)

## 2.2. Scrape Website

When the System
Launches the Scraper the Scraper searches the entirety of the website for relevant data (game info)
So that the system's database can be updated despite the target data not having a public API

## 2.3. Access API

When the System
Wants to update its database using a service that provides a public API they run a series of API calls
So that the system's database can be updated with the new data

## 2.4. Configure Genres

When an Administrator
Wants to blacklist or hide one or many genres they visit the 'genres' section of the admin site
So that blacklisted genres hide games assigned that genre and hidden genres don't appear in a game's list of genres.
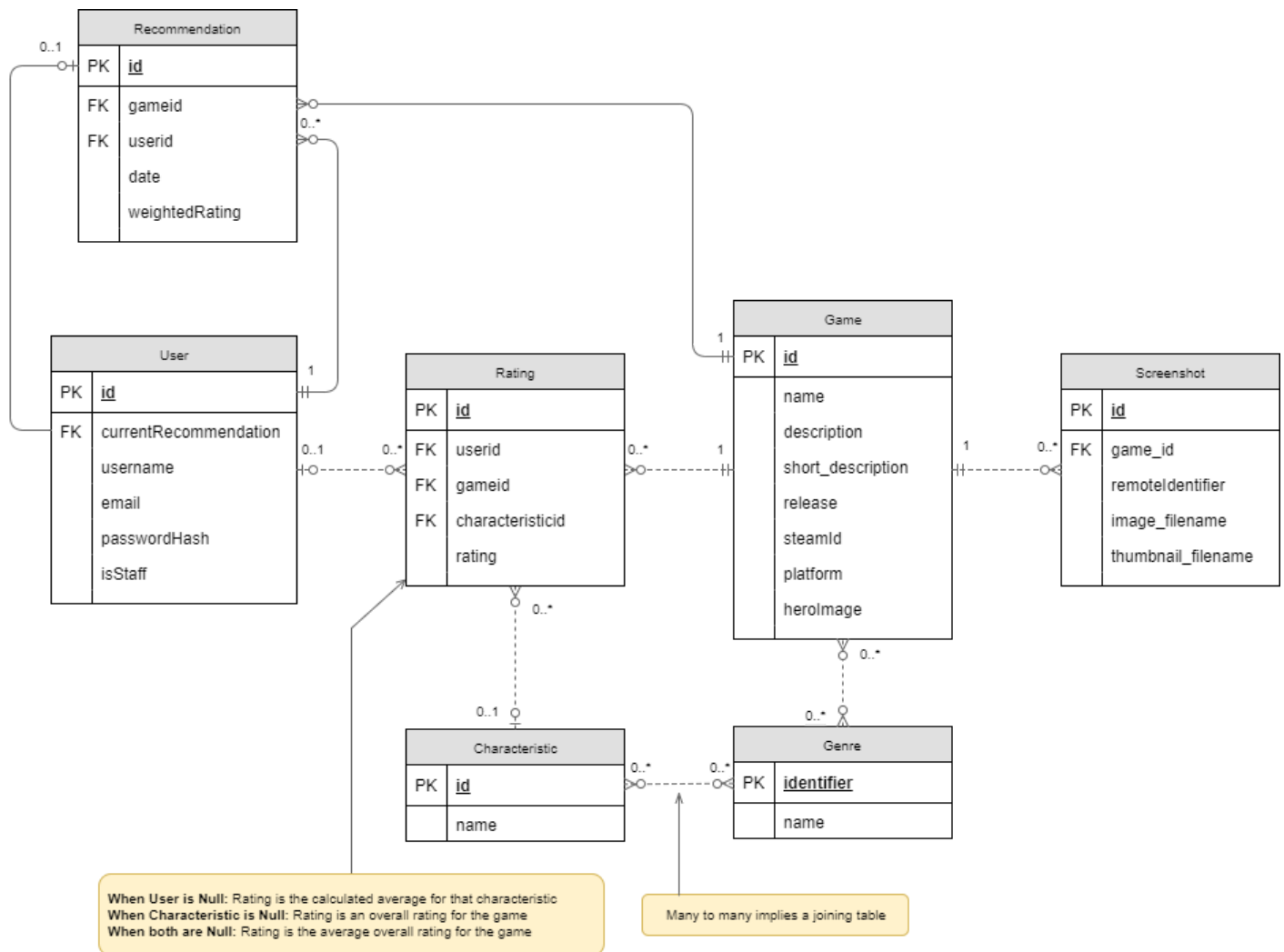
## 2.5. Configure Genre Characteristics
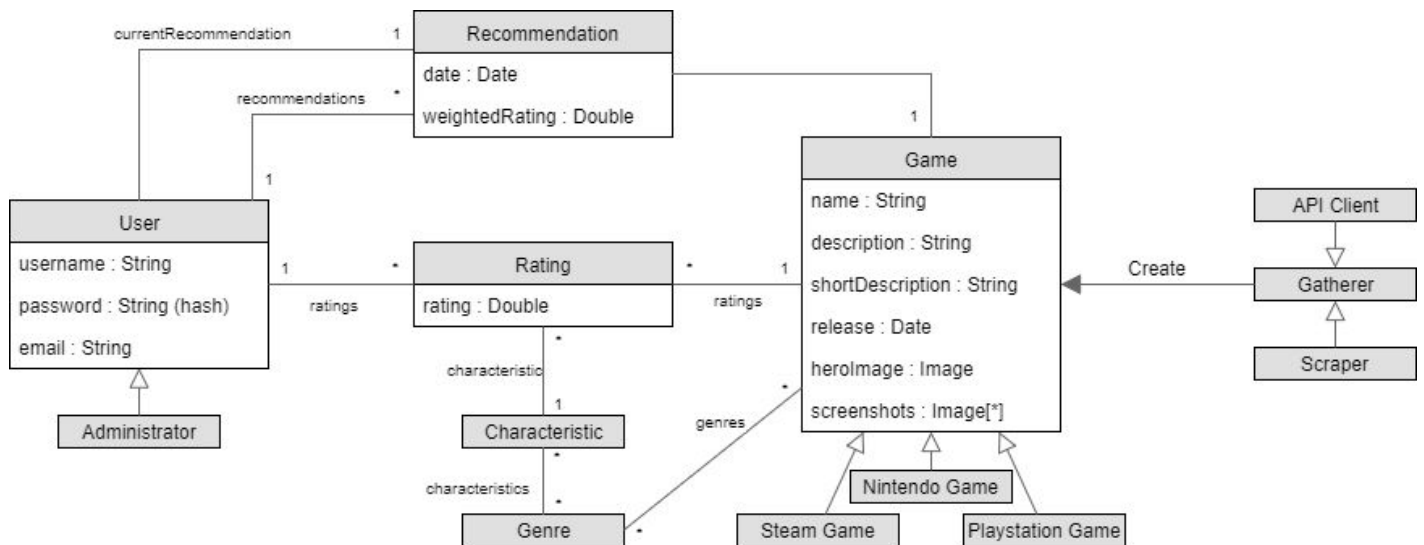
When an Administrator
Wants to specify which characteristics games within any given may be rated on they visit the 'characteristics' section of the admin site
So that the characteristics specified can be used to rate a game based upon by users.

# Database Design



When User is Null: Rating is the calculated average for that characteristic
When Characteristic is Null: Rating is an overall rating for the game
When both are Null: Rating is the average overall rating for the game

Many to many implies a joining table

# Domain Model



# Non-Functional Requirements

Can be found in the separate NFR list on version control