Project Design

Programming Languages and Tools

In this project, we will be using JavaScript, Node.js and the command line tool npm install. Listed below are some other Framework and packages we would use to make sure the system the plugin interact with runs smoothly between Discord and Steam:

- Discord.js - library used to interact with discord bot
- dotenv - a configuration package to store security tokens

We would also use some tools to help us program the Discord plugin. One of those tools is a Discord developer portal to create a discord bot for servers. This bot can store information from users in the server such as hours played, achievements and ratings of different Steam games. Another tool we would use is SteamAPI. Included in SteamAPI is ISteamUser, ISteamNews, and ISteamUserStats. These tools will help us collect data from the user's Steam account, and Steam in general, for the user to access through the plugin. We will use a command line interface as a tool to test and build the systems we implement for the plugin and discord bot.

Data Types Needed

This Discord plugin will need a few different data types to run successfully. The first data type is input types. These inputs can include:

1. Integers
    a. These integers will hold values of the game ratings on a scale of 1-5. These integers will be stored in the data structure to be displayed on Discord bots for servers or on the user's profile.
2. Strings
    a. These strings will hold the various bot commands to make sure that the Discord bots are storing the correct information and displaying it properly on the Discord server.
3. Chars
    a. These char variables will hold the achievements for each game that is played. Once an achievement is made, a notification is displayed to the user to show they earned an achievement. This is then stored in the user's profile so they can see what achievements they earned and what achievements are left to earn.

The plugin will also need several data structures to the inputs. The application will utilize vectors because of their dynamic array size. The structures will need to expand as more information is collected, so a vector will be the best option.

File Structure

File structure needed to use the discord bot
Node modules holds the framework libraries

Index.js is the main file, an entry point
Other js files are components for the many systems and functions of a functional requirement
Package.json is used to provide dependencies for npm and testing for npm
.env will be used to store configuration

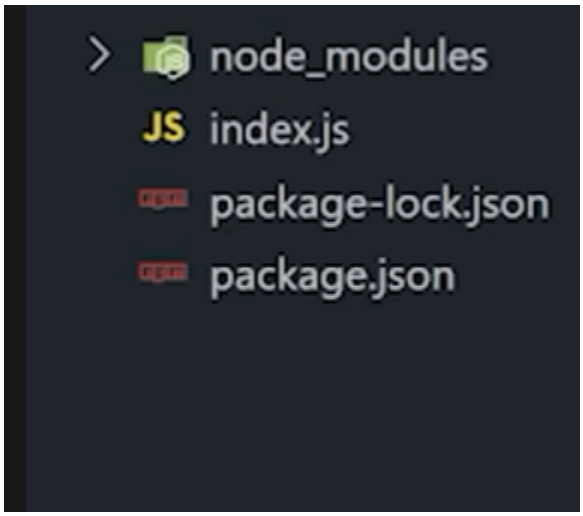

Figure 1: Displays an example of what the file structure for the Discord bot client might look like

Program Interaction and Deployment

The plugin mainly interacts between Discord and Steam. Discord acts as the main interface for the user to communicate with the application. Steam acts as the main source of data the application will utilize. When the user permits, the application will gather game data from Steam, or the most recently played gaming application, store it in the plugin's data structures, and display it through Discord's interface.
In regards to deployment of the systems, the Discord developer portal gives utility in posting an online discord bot listener.
This entity will listen to server-side calls and provide a way the user can interact with the system. Cloud hosting services like Google Cloud or Digital Ocean to store the information collected by each system and a way to ensure each system can be deployed in a live environment.

User Interface

The discord developer portal API provides many of the UI for the discord-bot, this will be needed in order to satisfy how the user interacts with the program. For the case of the requirements of the scenario for a rating system through discord, the modal UI provides the form interface for the User to provide to the rating system. The button component will be used to clarify how the user can submit their input to the system, additionally the button component will add context to many of the scenarios where the user will need to post and get information with an element of the a particular system for example, having a button to get positive or negative reviews for a Steam game. In addition, the User interface for key information components should be designed to visually communicate elements such as displaying Steam titles, playtime per title,

achievements for each title, and some visual representation of a notification for hitting certain playtime hours in the plugin application.
The images below show the possible components of the user interface that will communicate to the user what information a key component should provide.

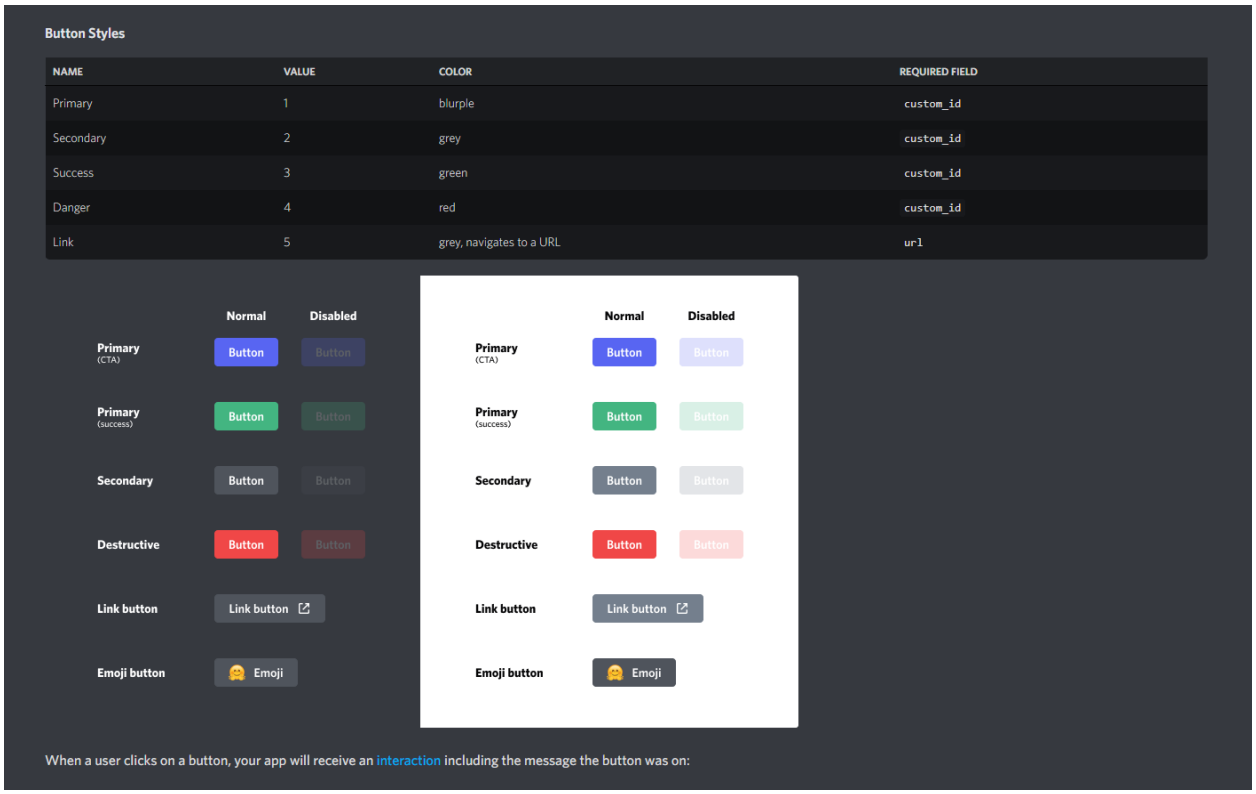

Figure 2: Explains the buttons that would be used for the user in Discord.

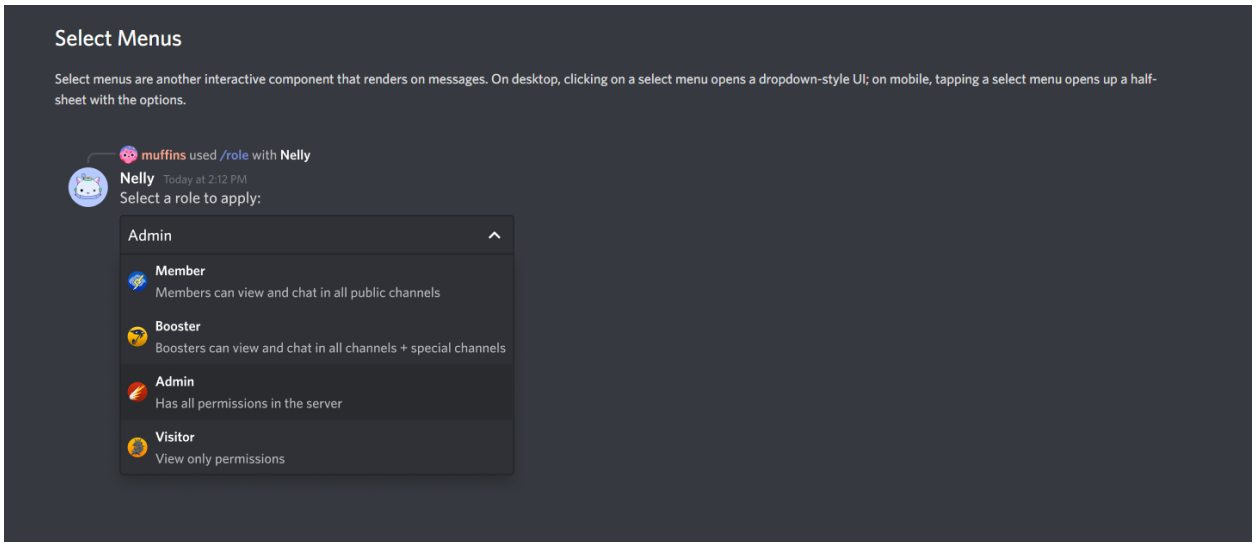

Figure 3: Explains the Select Menus that will be included in servers

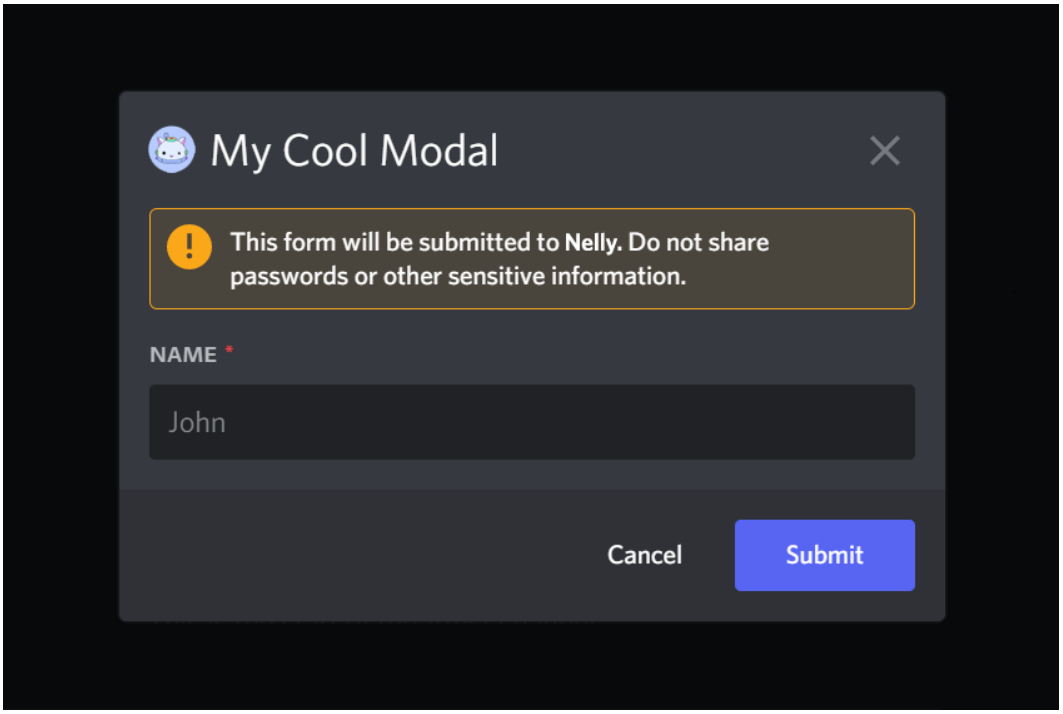

Figure 4: Sample modal that asks for your name. This would be used for recommendation and rating

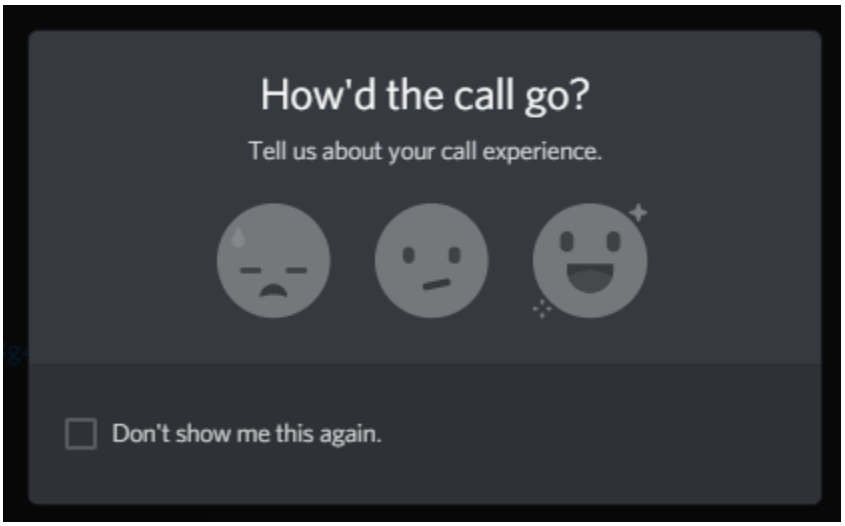

figure 5: UI by discord services to can be repurposed to fit the 5-star rating system

Prototype Development Tasks

- [ ] Implement a way to verify a link between the user's Discord and Steam accounts
- [ ] Create data structures to hold the user's game statistics (achievements, rating, etc.)
- [ ] Develop a bot to respond to user commands for fetching game data.

- [ ] Establish notifications for each component of the plugin (game achievements, hours played, etc.)