Kaisel BOT

Discord bot for Displaying Game Statistics

Martin Wong, Lucas Lee, Leo Chung, Cesar Figueroa Socarras

Github: Kaisel Bot Website

Table of Contents

Project Overview

For our project, we are designing and implementing a website and Discord bot that will display statistics for the popular multiplayer games *League of Legends* and *VALORANT*. Discord is a software messaging app that is popular with youth and is mainly used as a place to hang out on the web (Discord (software) - Wikipedia). In addition, we also want to integrate the Twitch API to display the top streamer and popularity for both games. If there is any unfamiliarity with Twitch, it is the most popular streaming platform mainly targeted for gaming. This project will divide the web application and the discord bot into 2 phases.

Phase 1: Web Application

Our first phase will be focused on our web application which includes React.js, HTML/CSS and client-side JavaScript. In this part, we are committed, but not limited to, the following:

- Building the framework and prototype of the website using Figma, vector graphics tool used to create mockups
    - About Figma (Wikipedia)
- Use web development and design languages to program the website. HyperText Markup Language (HTML) and Cascading Style Sheets will be the primary base of the physical design of the website along with JavaScript frameworks to program the behavior of the website.
- Within the website, we plan to have the following:
    - A home page displaying the games our Discord bot is connected to and the current top streamer. In addition, hyperlinks to our other pages will be found such a command list and information page about the bot
    - A statistics page for users to search up statistics for their desired game and account
    - A contact page displaying the members of the project
    - A 404 page in case a user types the wrong link

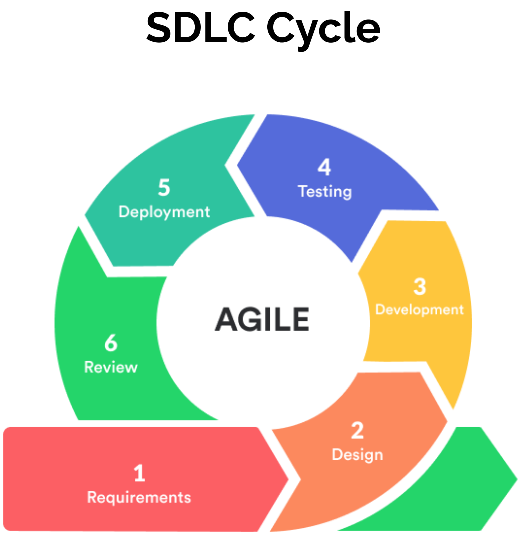
Phase 2: Discord Bot

Our second phase is the optional pure backend content which includes our Discord bot. In DIscord, bots can be used when a user types in a specific message that corresponds with the bot's commands. We plan to implement the following commands:

- Help: Notifies the user of the commands they can use and how to use them
- Stats: Gets the stats of a user from a specific game. The stats includes, but is not limited to, the username, rank, and win rate of that chosen user
- Online: Will check if a Twitch user is currently streaming
- Compare: Brings up two users and compares their stats

Backup Plan

If we are put into a situation where there is not enough time, we will make sure we at least have our web application working as it will meet all the requirements for this project.



The Software Development Life Cycle (SDLC) we have chosen is Agile. Agile is the perfect fit for both the team and for our project plan for the following reasons.

First, as post-secondary students, we are constantly subjected to numerous changes regarding our education and plan our daily lives around that. As a result, while we may have a general idea of what to expect in each course, there could come a time where we are unable to achieve the goals we initially set out for ourselves. Therefore, preparing our plans based on a flexible SDLC will allow us to accommodate and adjust to changes in both the project and in our daily lives.

Secondly, our project is ambitious. We aim to build both a website and Discord in the span of a month and a half (Second half of February and all of March). Being able to incorporate a project such as this one in such a short amount of time, including our daily responsibilities, is time extensive and desires that we are familiar with the languages we are using and the interfaces that are applied to those languages. This requirement is one that only some of the members on our team possess, while others need to be trained. Taking into account the time needed to plan for the project, train individuals, and build both website and Discord bot, it is imperative that our SDLC centers on flexibility. As a result, Agile accommodates the time needed to implement all three requirements listed above.

Based on the two reasons listed above, the team has concluded that an agile framework is the optimal solution to our situation.

User Stories

Riot Games

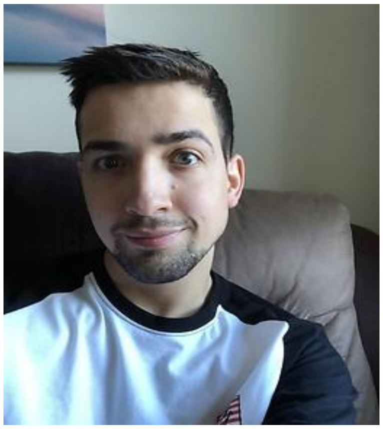
(League of Legends and Valorant)

User Persona: Wei Yong

Personal information

Name: Wei Yong
Age: 20 years old
Born: Vancouver, BC
Description: Wei is a 20 year old professional gamer, in a team called sentinels. He started playing Minecraft when he was around 12 years old but then moved on to first-person shooters at the age of 15. Started playing CS:GO professionally at the age of 18 and signed with Cloud9. He first attempted streaming in November 2015 but didn't really get into it until November 2017. He received the Twitch partnership in July 2018.

User Persona: John Elmo



Personal information

Name: John Elmo
Age: 29 years old
Born: California, US
Description: John, a 29 year old gamer, travels around the world playing League of Legends. John loves to play games ever since when he was younger, first playing Pac-Man then Super Smash Bros and then League of Legends. He has won 3 world Championships and 2 tournaments.

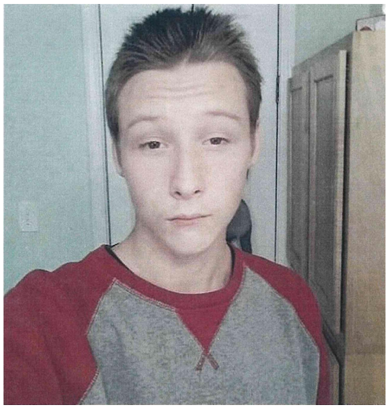
Twitch

User Persona: Victoria Gray

Personal information

Name: Victoria Gray
Age: 30 years old
Born: New York, NY
Description: Victoria is a streamer who mainly plays video games. At first she was uploading random game content. Only started streaming on Twitch around 2019 after a friend convinced her to try it out and grew in popularity around 2021 when she started playing League Of Legends.

User Persona: Willy Carter



Personal information

Name: Willy Carter
Age: 16 years old
Born: Las Vegas, US
Description: A 16 year old guy named Willy. He watches a lot of streams on twitch, his favorite person to watch is loltyler1. He likes to play games like league of legends and valorant in his free time. As he watches all the streams he relaxes while watching and it relieves his stress.

Tech Stack

The following items listed below are the languages and frameworks we plan to implement during the course of the project:

- ReactJS
- HTML/CSS/JS
- MongoDB
- Github - hosting
- Nodejs -> for Discord.js
- ExpressJS

For the website, we will have HTML, CSS and JavaScript.. In addition to HTML, CSS, and JS, we also will specifically implement ReactJS to create a graphic user interface (GUI) and MongoDB to store the stats of each game's users. NodeJS will be used when developing the Discord bot and the backend of our website. Finally, Github will host the website.

API USAGE

Since we are making both a website and Discord bot that displays League and Valorant stats plus any Twitch streamers that play the associated games, we have chosen the Riot Games API for League of Legends and Valorant, the Twitch, and Discord API. Each API will have the following uses:

Riot Games API

- Username
- Current rank
- Kill/Death/Assist ratio (K/DA)
- Favorite role/champion

Twitch

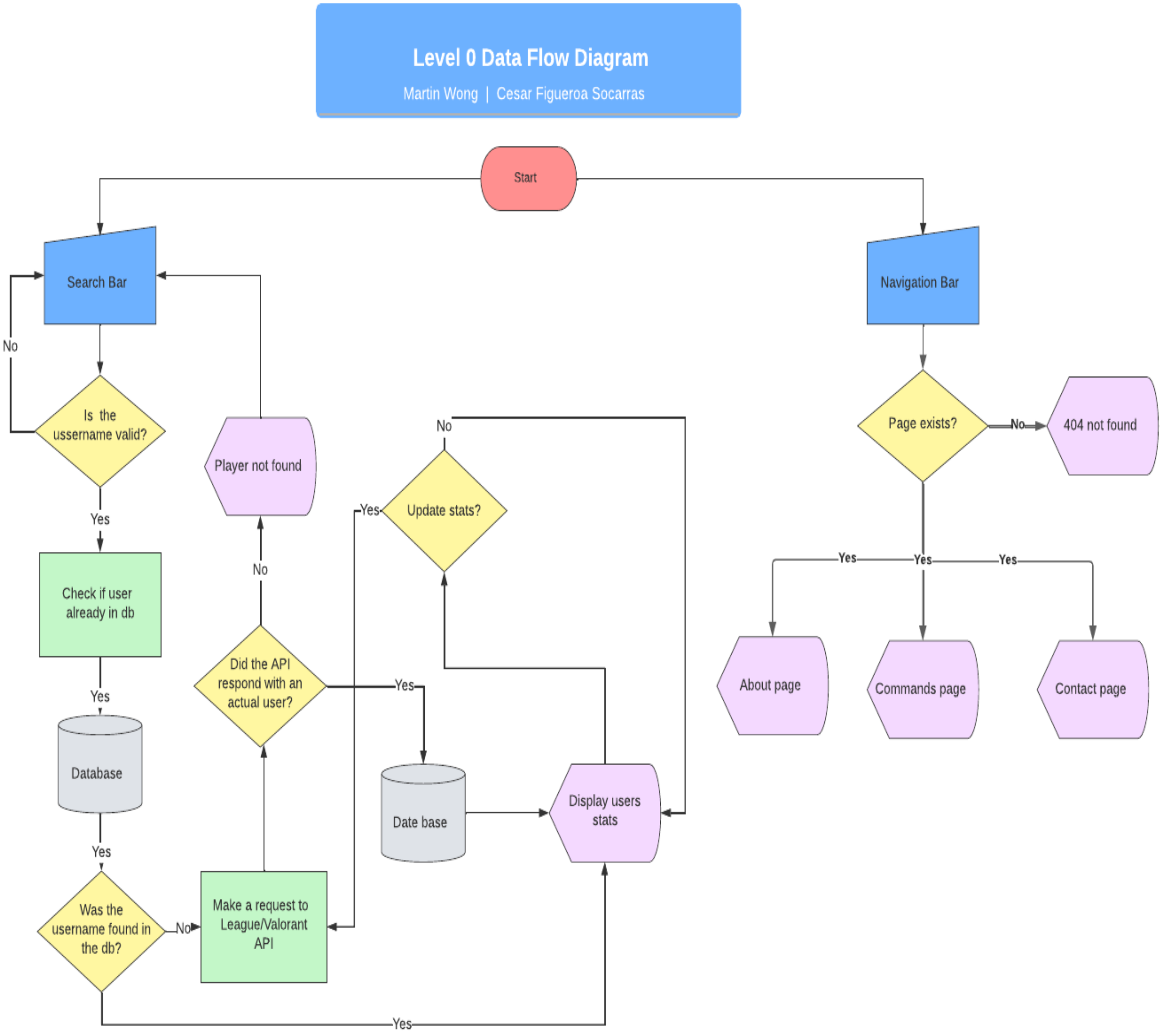
- Top (1-3) streamer(s) for each game
- Current views on Twitch for game/rank
- Allow user to search for streamers/channels playing their chosen game

Discord

- Display stats from the Riot Games API
- Display Twitch streamers playing both games
- Display Custom commands

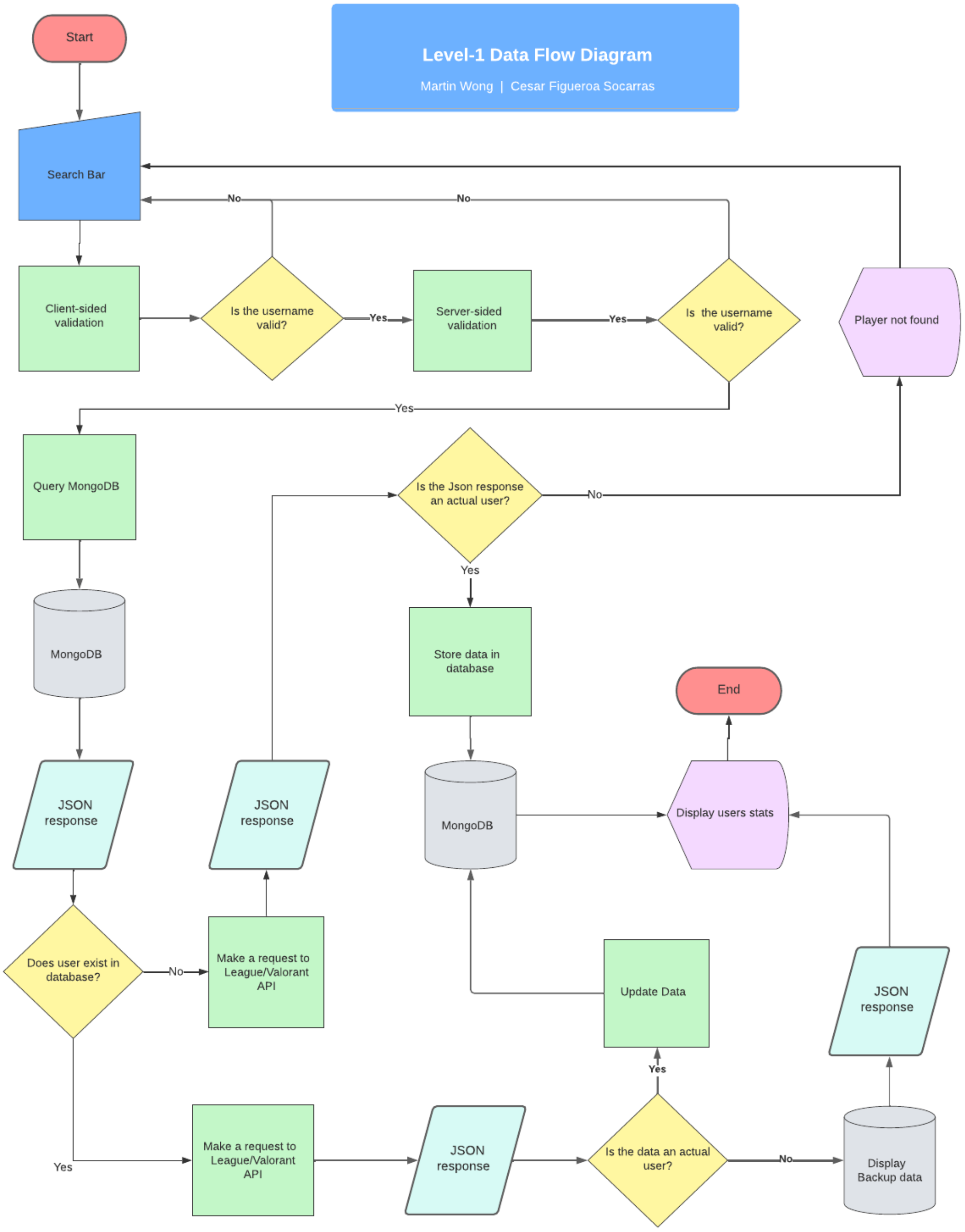
Level-0 Data Flow Diagram

Website: Searching for League/Valorant Usernames



Level-1 Data Flow Diagram

Searching for League/Valorant Usernames



WBS (Work breakdown structure)

Schedule

The image shown below is a general overview of our project structure. In the first portion of February, we will begin preliminary planning of the project. After that, we enter the last portion of February and first half of March. This is when the website will start to be developed and finished. Once that is complete, we will spend the second half of March testing the website and work on the Discord bot.

February (1-19)	February (20-28) and March (1-12)	March(12-31)	April
1.0 WBS	2.0 Features	3.1 Start working on bot	Report & Presentation April 2nd
1.1 Wireframe	2.1 Build content and styling of the website	3.2 Implement APIs (Twitch & Riot Games)	
1.2 Tech Stack	2.2 Get hosting to work	3.3 Testing	
1.3 Data flow Diagrams	2.3 Need javascript logic (Buttons/Forms)	3.4 Keep updating website with new features	
1.4 Website structure	2.4 Implement API		
1.5 Report	2.5 Server-sided scripting needs to be done		
1.6 Practice presentation, at least 1 day before	2.6 Website should be done		

Task Allocation

Each member of the team played an important role in the planning phase of the project. Listed below are the tasks that everyone was assigned and/or contributed to:

Cesar

- API researcher
- Created data flow diagrams
- Decided on features
- Decided on the tech stack

Leo

- Assisted on wireframe sketching
- Wrote user stories

Lucas

- Assisted on the powerpoint (presentation)
- Report co-writer and editor
- WBS editor
- Wireframe designer (not sketcher)

Martin

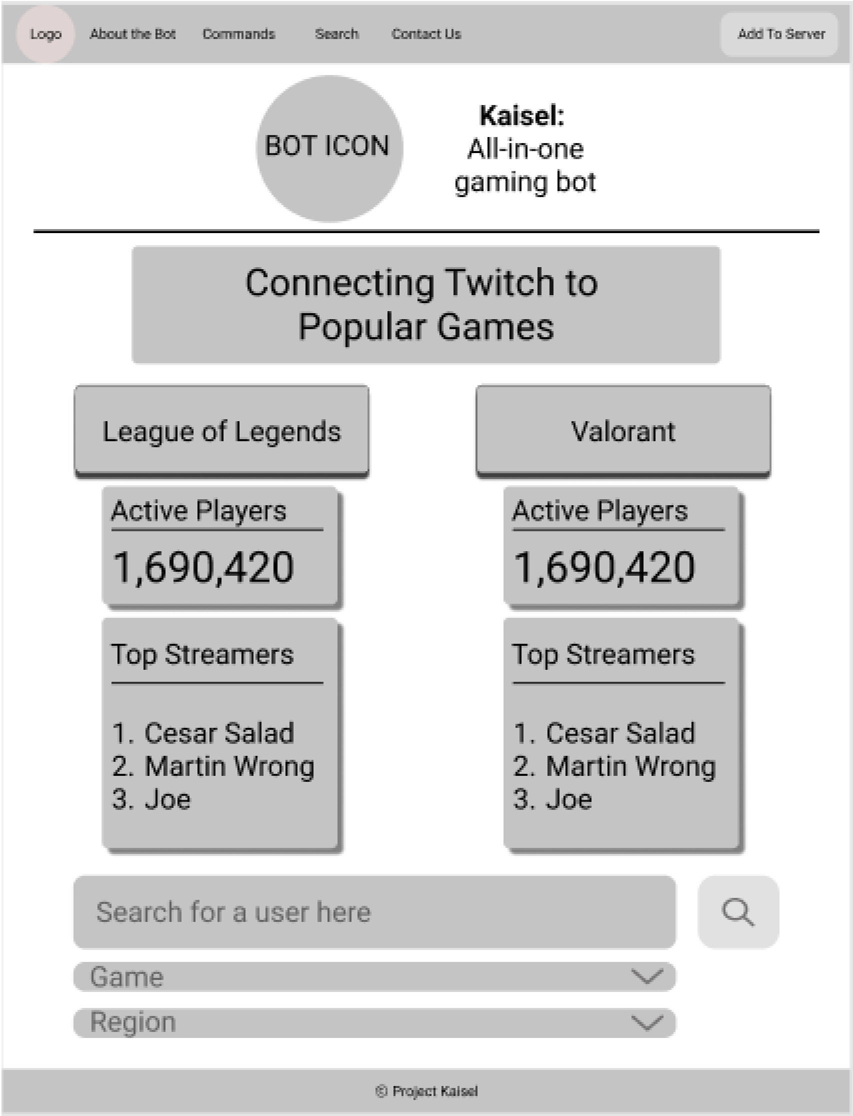
- Assisted wireframe sketching
- Created data flow diagrams
- Created WBS
- Created powerpoint (presentation)
- Decided on the tech stack
- Decided on website structure
- Original writer of the report

Report and Presentation Breakdown

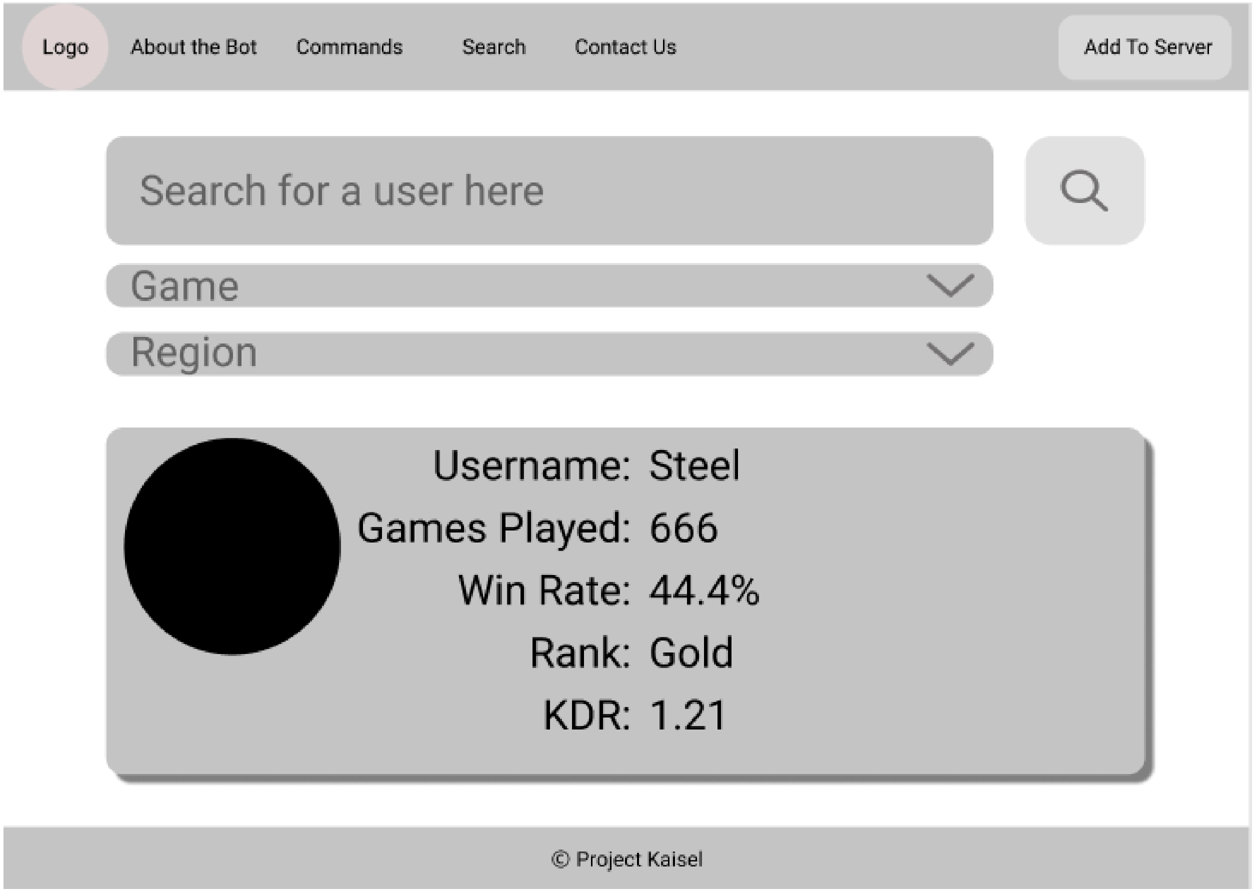
Building the structure of the report was left to Martin Wong and Lucas Lee. Martin built the general framework of the report and listed all of the work we have done thus far into the document while Lucas worked off of what was listed, added additional explanations, and edited the existing framework. As for the presentation, Martin built the majority of the slides pertaining to the defining aspects of our project.

Wireframes

Home Page



Statistics Page



Username: Steel

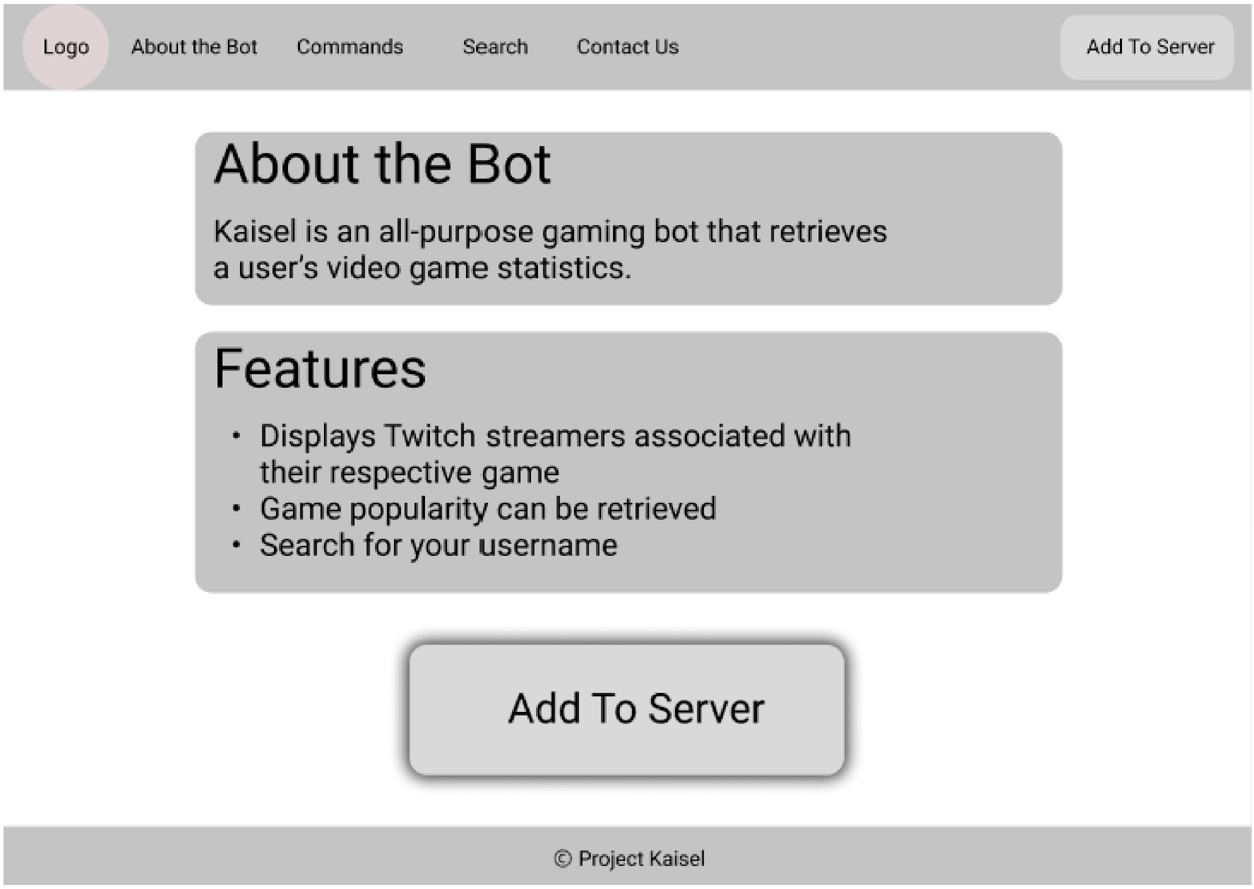
Games Played: 666

Win Rate: 44.4%

Rank: Gold

KDR: 1.21

About Page



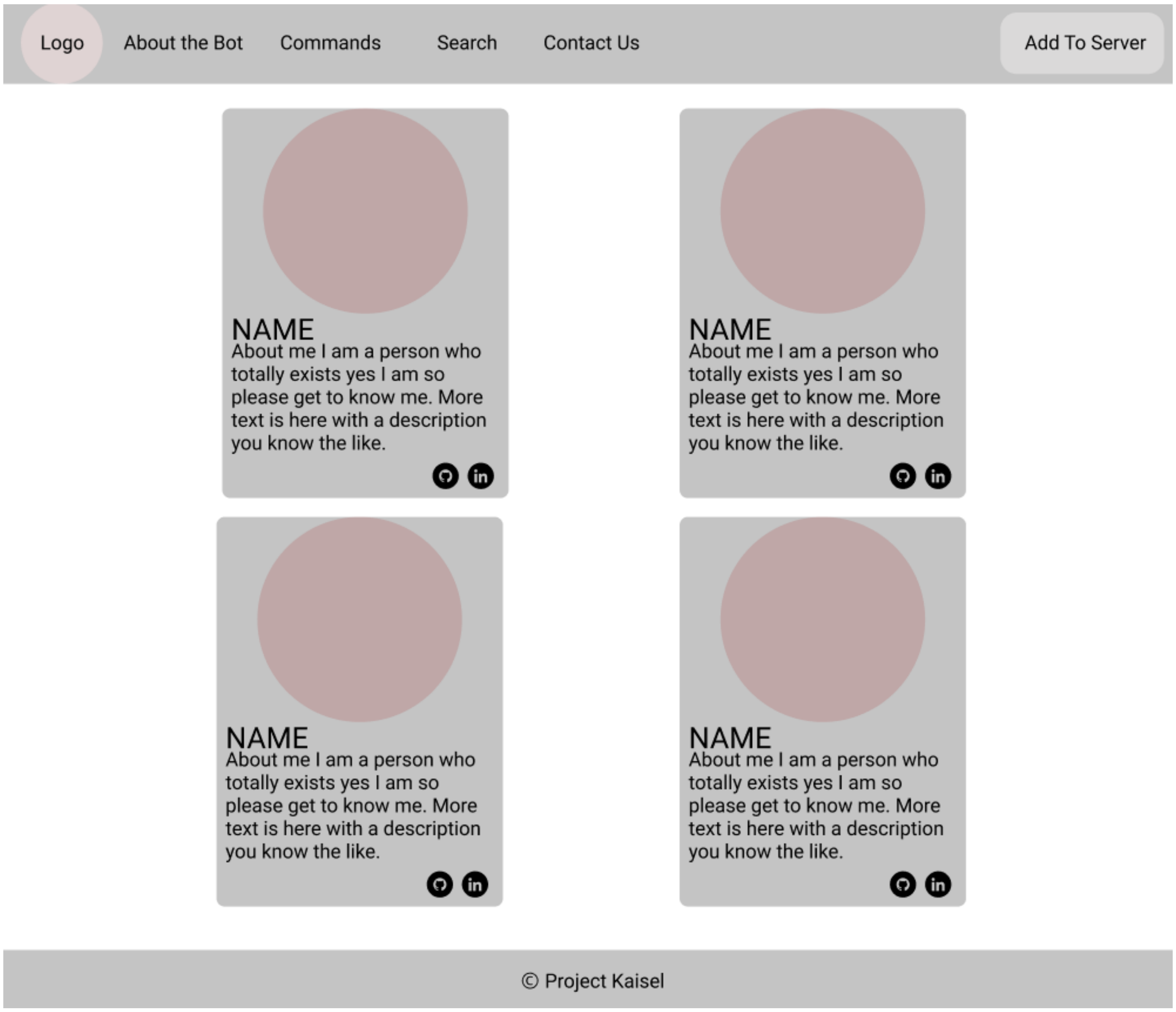
About the Bot

Kaisel is an all-purpose gaming bot that retrieves a user’s video game statistics.

Features

- Displays Twitch streamers associated with their respective game
- Game popularity can be retrieved
- Search for your username

Contact Page



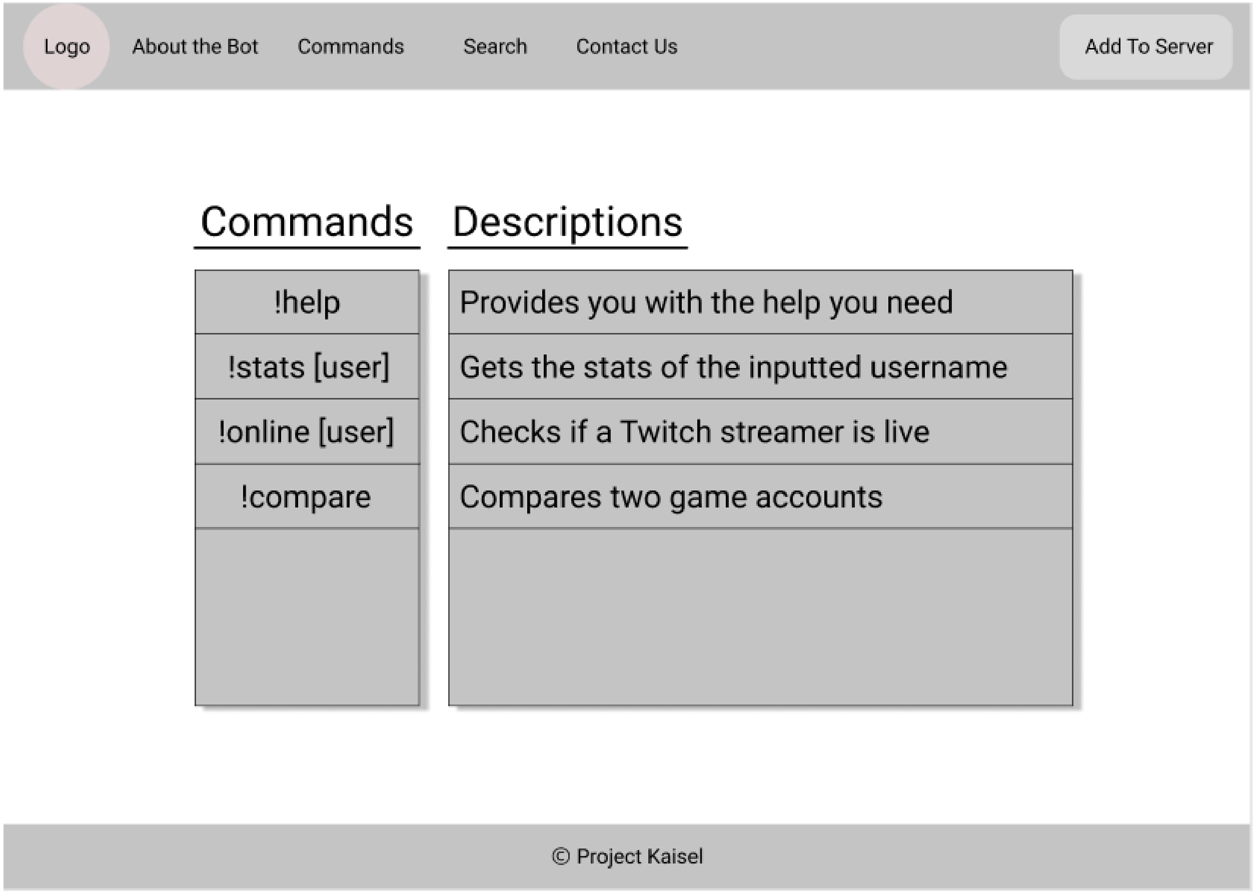
NAME
About me I am a person who totally exists yes I am so please get to know me. More text is here with a description you know the like.

NAME
About me I am a person who totally exists yes I am so please get to know me. More text is here with a description you know the like.

NAME
About me I am a person who totally exists yes I am so please get to know me. More text is here with a description you know the like.

NAME
About me I am a person who totally exists yes I am so please get to know me. More text is here with a description you know the like.

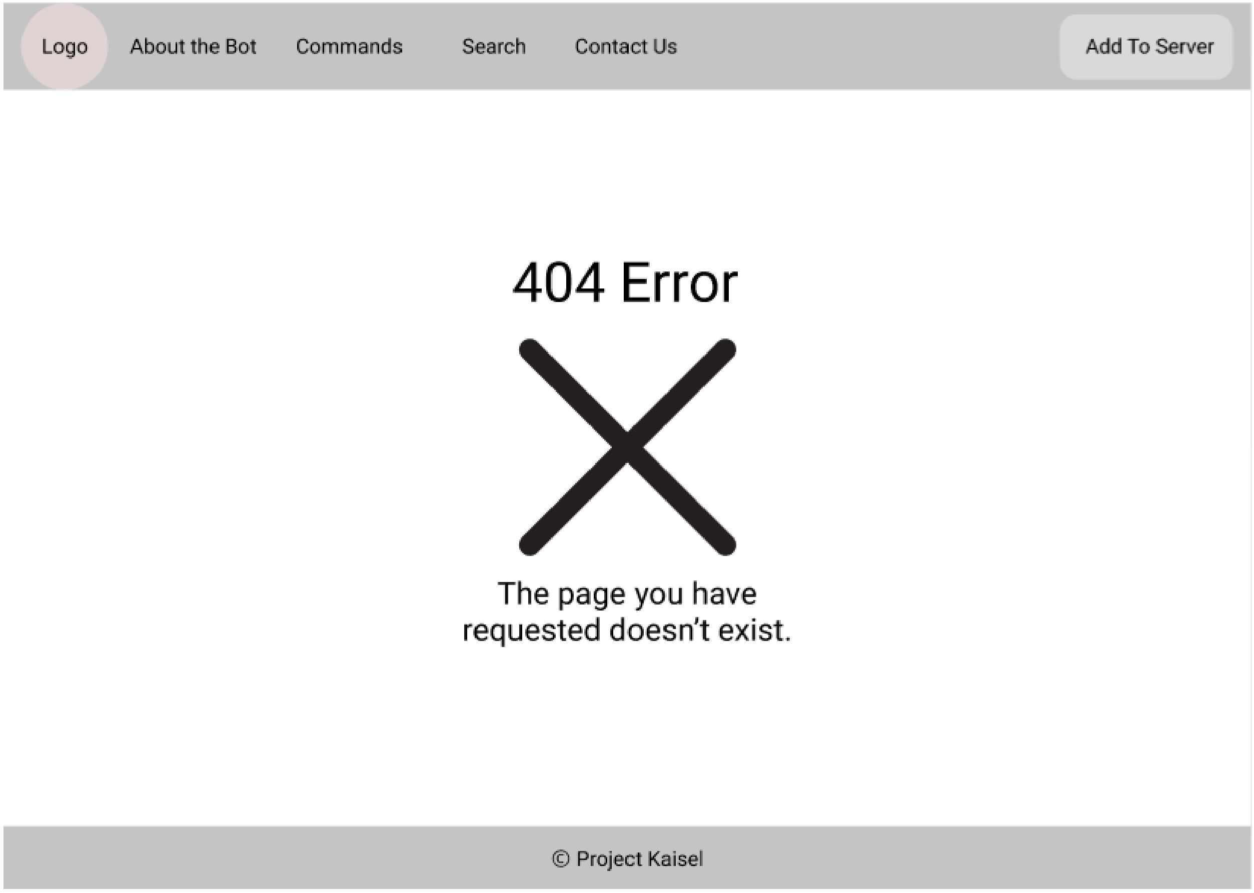
Commands Page



Commands	Descriptions
!help	Provides you with the help you need
!stats [user]	Gets the stats of the inputted username
!online [user]	Checks if a Twitch streamer is live
!compare	Compares two game accounts

© Project Kaisel

404 Page



Additional Links to Our Work:

Github: [Frontend Kaisel Bot]
Prototypes and Mockups: Website Prototype in Figma

ADDITIONAL NOTES

ROLES AND TASKS

Cesar

- Website design / Server-sided scripting

Leo

- Website design / Front-end

Lucas

- Website design / Front-end

Martin

- Website design / Server-sided scripting

Website Structure

- index.html
  - about our bot
  - how to use bot
  - commands
- stats.html -> op.gg (basically)
- status.html -> status of bot (online/offline)
- contact.html

Meetings

Every Friday night at 8PM. Additional meetings can be scheduled if needed.