

CSC468: Introduction to Cloud Computing

Valorant Statistics Tracker and Prediction Generator

Team 3

Members:

Ryan Sayre

Maxwell Mendenhall

Tobyn Sitar

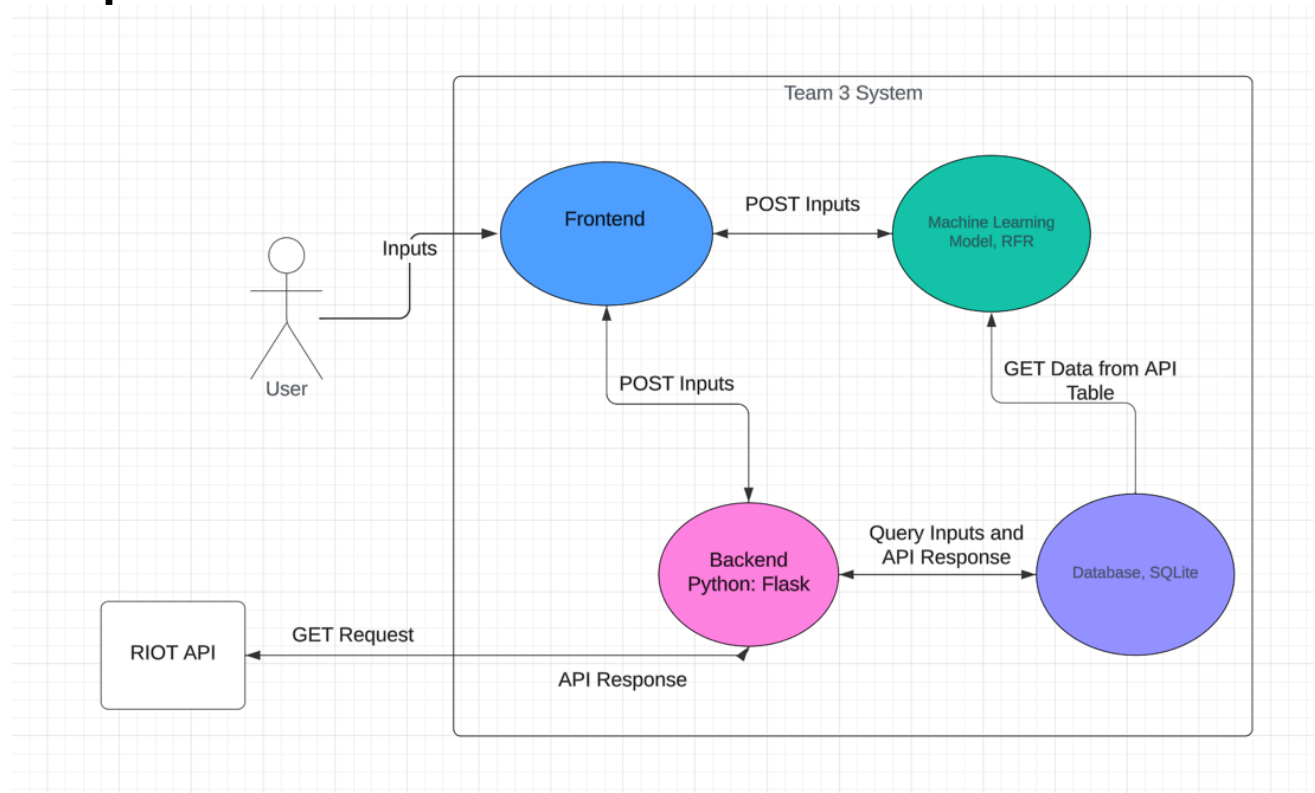
Katherine McCarthy

Project Summary

Our project aims to revolutionize the statistics tracking service available for the popular competitive first-person shooter Valorant. While there are sites such as Tracker.gg and U.gg, we plan to create a new application that provides statistics specifically for Valorant's unique mechanics. With our project, we want to revitalize the way users track their own or others' statistics. One of the main features we want to implement is comparison, in which you can have a side-by-side comparison of your stats compared to another player. With this feature, we plan to ramp up the competition between friends, or prove to an enemy that you are the better Valorant player. We also plan to have an accurate system, where you can see accurate statistics that automatically update the second you finish your match. Furthermore, with our project, we want to implement a user-friendly hub for all your favorite topics regarding Valorant. This homepage will show patch notes, a breakdown of best team compositions and characters, a way to find and connect with others who are looking to play the game, and a way to upload and talk with others about gameplay highlights and opinions. We also plan to implement a machine learning algorithm to predict future performance statistics, based on the last batch of games played. Essentially, we want our model to factor in the most recent statistics into our algorithm to determine predictions for future matches. We want our project to be the best community website for Valorant players, both competitive and casual.

With this, however, we need to consider some limitations. One of the most important limitations is the timeframe. With all the ideal features and components we want to implement into our tracker, time will be critical to completion of our goals. The 5-month time limit given to us is ideal for a lot of work, but with other commitments and classes, providing the adequate time limit to complete our project will be difficult. Another factor that could potentially limit us is the lack of hardware. Fortunately, some of us have complete desktop computers fully equipped for the work, but some members may not be as fortunate. We will do our best to utilize resources like CloudLab effectively and continuously assess the best options to continue making forward progress with the project. Our main source for completing this will be GitHub, so we can remotely commit and add to the project from our separate spaces while working on the different components of our project collectively, and we plan on meeting regularly to work on the project together. We are very excited to begin working through this project and hope that we can achieve our goals at the conclusion of the course.

Chapter 1



Full Stack:

Front end: Web UI

Back-end: Worker, Machine Learning Model

Database: SQLite

The vision of our project starts with the user, who will access our Web UI by URL. Our Web UI will have a welcome page where users input their own userID for the PC game Valorant, made by Riot Games. The user can optionally create an account that will save relevant information into an account holders database. The account feature will be used for user convenience and to avoid the need for users to input redundant information. Once the userID is entered, a backend worker program will retrieve the necessary data from Riot Developer Portal, which contains the APIs for all user data and statistics. The desired statistics and data will be filtered and organized into a machine-readable format. Here we will apply a Machine-Learning Model which will use the data and statistics to perform a prediction on the outcome and statistics of a current match based on data trends. This can be performed using the Riot Developer Portal, which also records data for active matches. Once the Machine Learning Model is applied to the data, the backend worker will retrieve the statistics and predictions from the model and display it to the user through the Web UI. We want our project to provide insightful predictions that other statistic tracking services do not provide to make ours the most popular among the Valorant player base.

Chapter 2

Web UI

For this program, we want a clean, user-friendly UI. Ideally, we want to utilize tabs in order to direct users to their preferred method of tracking, whether that be recent games, aim statistics, the ability to see recently played with or most played with users, etc. Essentially, we want the program to run the same way as other statistics trackers, but stylized to our preferred game of choice, and with more features not commonly found on other sites. The website will be clean, with no outside influences (ADs), and have an aesthetic appearance to it, allowing users to frequent the site. Furthermore, if time permits, we plan to take our application to the IOS and Android stores for users to take with them on the go. We also want to implement a system for users to login, connect their RIOT account, and track their personal stats. Our priority is making sure the system works for everyone, not just a specified group of players that know their way around complex online gaming trackers. We plan to use HTML, Canva, etc. In order to build the UI and implement js and bootstrap. We are not 100% certain how these will interact, but plan to have the program work much like other statistics trackers in its finished stage. We will continue to work together to implement our desired features while researching the best methods or software to use to help us in this project.

Database

For our database, we are planning to use SQLite, as it is lightweight and already built into Python. We have done research on this, and SQLite fits all we want to build into our tracker, without the heavyweight aspect. Because our team is most comfortable using Python, and SQLite has a python package built into it, we have decided that currently we will work with that to complete the database related functions of our project. We also looked into using MySQL, but for a project of our scale we have currently deemed it not necessary, but we will re-evaluate our progress regularly and assess whether or not software changes must be made

Backend

For the backend development leaning towards Python, because of Python's wide spread of support for different databases and APIs. It will also be compatible with SQL based databases and have good integration with SQLite which we plan to use.

Performance Prediction

This part of the project is a primary feature of the application. Competitive players often want to see how they will perform in a match and often look up the stats of other opponents to see how they will play. Taking the heavy lifting off the players and letting them use external software to predict this is the primary goal of the feature (this obviously relies on the fact that the user has a type of consistent performance throughout their games).

Many ML models will be tested on the data before we decide on the final model. The models and not limited to are RFR (random forest regressor) and Basic Feedforward Neural Network. Stacking multiple models also to see the cost outcome will be considered.

A big part of machine learning and the part that usually takes up the most time is feature engineering. We will take the data given by the Api's and engineer the data to come up with more features that we think will be relevant in helping predictions. This step-in machine learning

is usually trial and error so an exact number of features we will not be able to provide until the end.

The language this feature will be built on top of is python. Using python there are many libraries that aid in the development of machine learning like Scikit-learn and TensorFlow. We can build an API on top of flask (python web framework) to expose the model endpoints for the other microservices. The features will then be transferred over web protocols and the data will be passed into the model via query parameters with the web server responding with the prediction.