Revision History

Date	Version	Description	Author
05/ April/ 2020	1.0	Software Architecture Document generated, Introduction, purpose and scope of the document	
05/ April/ 2020	1.1	Problem Definition, Project Features, User Stories, Project Constraints	
07/ April/ 2020	1.2	Design and Code Inspection	
07 / April / 2020	1.3	Conceptual design	
09/ April / 2020	1.4	Design Alternatives	
11/ April/ 2020	1.5	Problem Definition	
11/ April / 2020	1.6	Diagrams and Figures	
13/ April/ 2020	1.7	Detailed Design	
13/ April/ 2020	1.8	Detailed Design	
16 / April / 2020	1.9	Conceptual Design	
17 / April / 2020	2.0	Problem Definition	
20 / April / 2020	2.1	Key achievements	
21 / April / 2020	2.2	Team Appraisal	
25/ April/ 2020	2.3	Finalised Report	

Table of Contents

Fat Donkey Software Architecture and Design Report

1. Introduction

Competitive video gaming has rapidly emerged in popularity over the last decade and is becoming one of the fastest-growing industries all around the world. One of the staple titles in this space is the highly popular video game “League of Legends” made by developer “Riot Games”. The game involves two teams of five competing with each other to destroy the enemy team’s base. While this premise may seem quite simple, the developers of this game have devised a vast amount of strategies and techniques that players can utilise which have varying degrees of effectiveness.

We have developed a third-party data analytics website designed to be used when players are playing the game, to improve their performance and win rate.

We believe there is value in our project as the game appeals to both casual and competitive players, reaching 8 million concurrent players daily and generating $1.5 billion USD in revenue for the developer ‘Riot Games’ in 2019. Our application will be of great use to a large portion of the player base as it provides access to features not available through the official client.

1.1 Purpose

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

1.2 Scope

This Software Architecture and Design Report provides an architectural overview of the BlitzPlus System. The BlitzPlus System is being developed to support competitive online gamers of one of the world’s most popular video games, *“League of Legends*”.

The purpose of the document is to provide information on the completed software architectural requirements and design of the BlitzPlus web application that is required to provide players of the game with data sourced from the official ‘*Riot Games*’ API showing the best and most effective strategies that players can utilise to improve their in-game performance and consequently, increase their chance of winning. This document includes the software requirements, design specification and any relevant information to explain how the conceptual and detailed design meets the completed requirements through the use of textual descriptions, diagrams and tables. This document:

- Highlights what the problems were from a software development perspective
- Define solutions to overcome these problems and achieve successful completion of the project
- Outline the conceptual and detailed design of this project, including the initial requirement analysis and design alternatives.

The intended audience of this document is to all of the stakeholders for a project that involves one of the world’s biggest online video games. More specifically, software developers, previous and current online video game players of the *League of Legends* as well as Riot Games, the original developer of the game.

1.3 Definitions, Acronyms and Abbreviations

Champions - playable characters within the game of League of Legends.

Summoners - refers to the human player that plays the game.
Runes - are relatively minor enhancements for a champion's abilities that are selected in the pregame screen. (There are a variety of options for runes depending on the champion)
Items - bought during the game using gold that is accumulated by the player and can greatly enhance how powerful a champion is.

1.4 Problem Definition

We have been required to design and build a web application that can help users navigate through information sources.

The Fat Donkey team chose to design and implement a website a popular online video game, "*League of Legends*", to make vital information, currently not available to users within the game's client, available and easily accessible to them to enhance their in-game experience. The project's objective is to present statistics relating to summoners, champions, items and runes, sourced using the data from the game's API and through web scraping.

2. Requirements Analysis

This document presents the Requirements Analysis as a series of Project Features based on the User Stories collated by the Fat Donkey Software Development team. This document also details the project constraints and technical limitations that impacted the project.

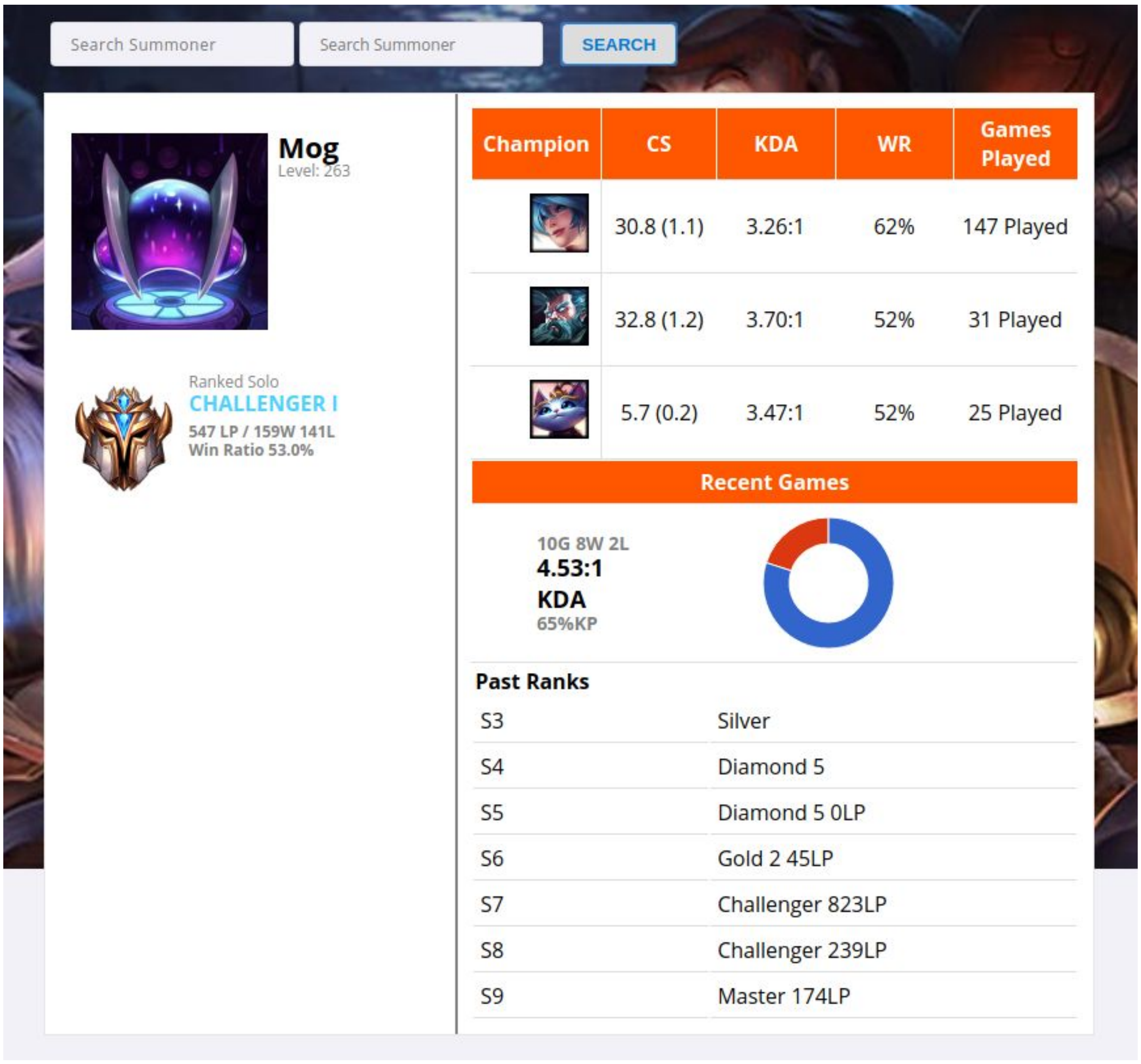
2.1. Project Features

1. Players can search for statistics and results from a summoner's games.
2. Players can search and compare statistics and results from other summoner's games.
3. Players can find out what the most popular items/runes are on each champion.
4. Players can find out what the most popular champions are in current meta-game.
5. Players can find out which summoners are the best at particular champions.
6. Players can find out the statistics of all the players in my current game.
7. Players can see a player's performance over time.

2.2. User Stories

1. As a League of Legends player (henceforth known as "summoners") I want to have a website that contains all the statistics and results from every game I play so that I can see my performance in each game.

   Scenario: Search statistics and results from summoner's own games
   Given: I am on the BlitzPlus home page
   And: I type my summoner name in the summoner name search bar
   When: I click Search
   Then: I should see my summoner dashboard with match history, rank and win rate.



Mog
Level: 263

Ranked Solo
CHALLENGER I
547 LP / 159W 141L
Win Ratio 53.0%

Champion	CS	KDA	WR	Games Played
	30.8 (1.1)	3.26:1	62%	147 Played
	32.8 (1.2)	3.70:1	52%	31 Played
	5.7 (0.2)	3.47:1	52%	25 Played

Recent Games

10G 8W 2L
4.53:1
KDA
65%KP

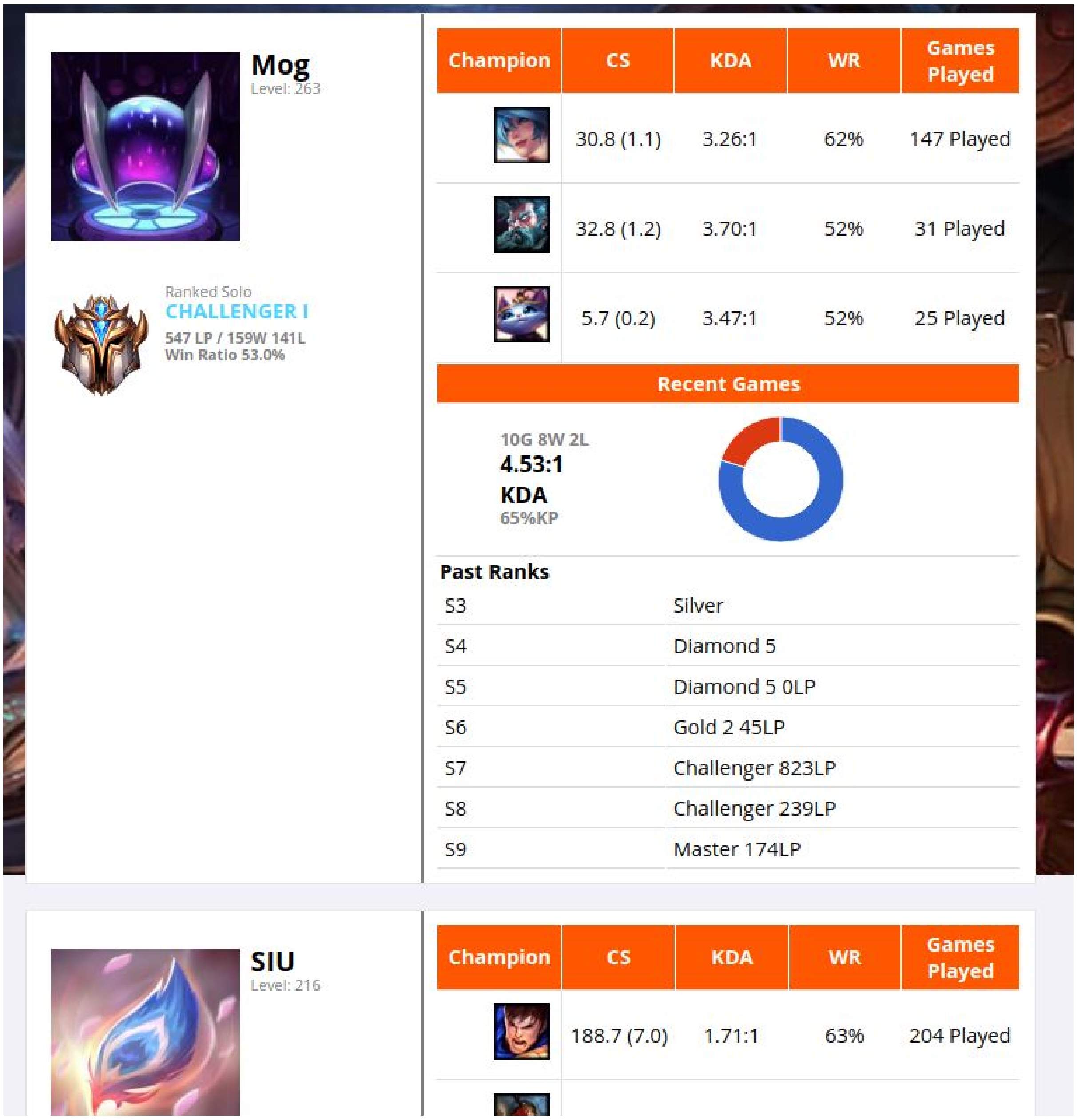
Past Ranks

S3	Silver
S4	Diamond 5
S5	Diamond 5 0LP
S6	Gold 2 45LP
S7	Challenger 823LP
S8	Challenger 239LP
S9	Master 174LP

A live example from the BlitzPlus web application of User Story / Feature 1 - Screenshot displays statistics of player

2. As a fan of the game, I want a website that allows me to view the different stats of other summoners and allows me to compare them to my own so that I know what part of my gameplay I need to improve.

   Scenario: Compare statistics and results from summoner's own game to other summoners.
   Given: I am on the BlitzPlus home page
   And: I type my summoner name along with another in the summoner name search bar.
   When: I click Search
   Then: I should see two dashboards with a comparison of match history, rank and win rate.



Mog
Level: 263

Ranked Solo
CHALLENGER I
547 LP / 159W 141L
Win Ratio 53.0%

Champion	CS	KDA	WR	Games Played
	30.8 (1.1)	3.26:1	62%	147 Played
	32.8 (1.2)	3.70:1	52%	31 Played
	5.7 (0.2)	3.47:1	52%	25 Played

Recent Games

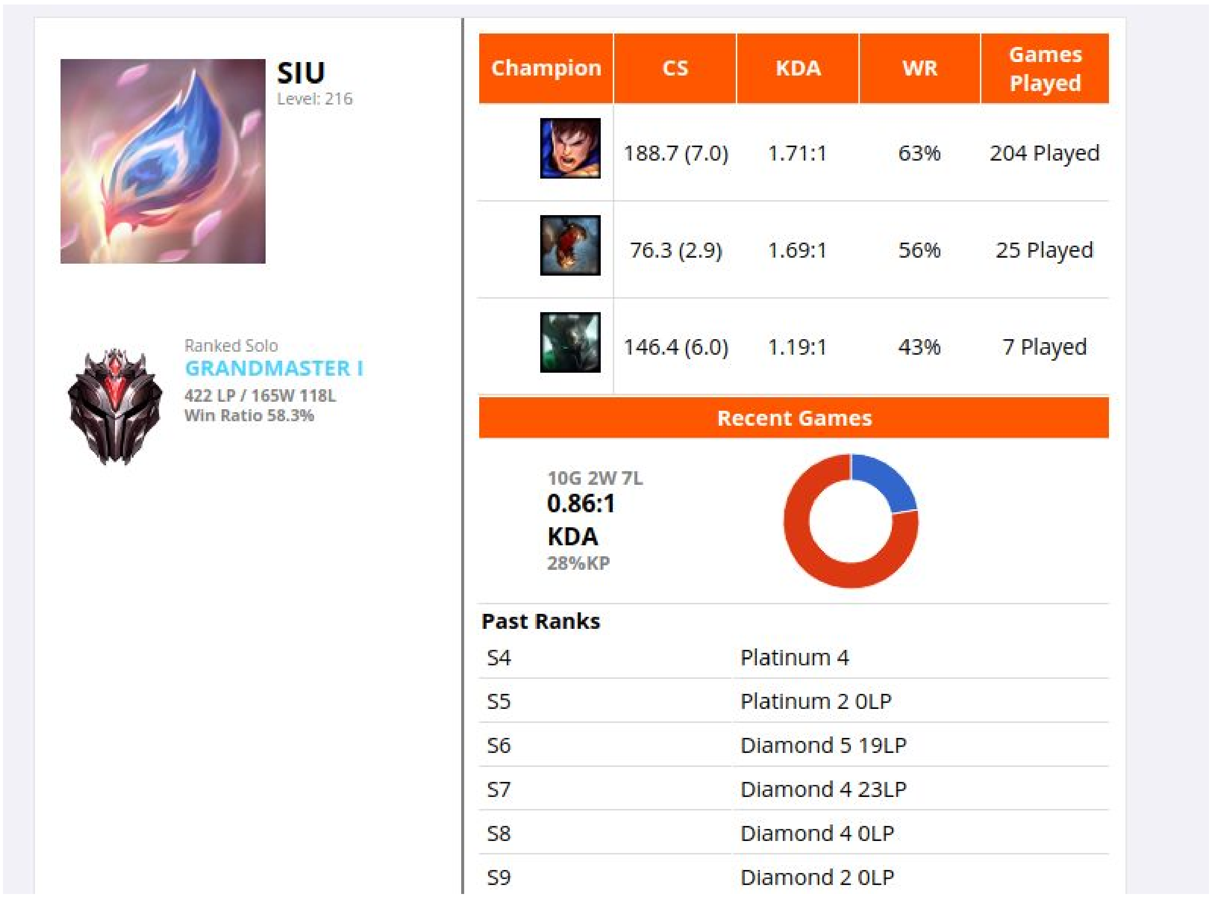
10G 8W 2L
4.53:1
KDA
65%KP

Past Ranks

S3	Silver
S4	Diamond 5
S5	Diamond 5 0LP
S6	Gold 2 45LP
S7	Challenger 823LP
S8	Challenger 239LP
S9	Master 174LP

SIU
Level: 216

Champion	CS	KDA	WR	Games Played
	188.7 (7.0)	1.71:1	63%	204 Played


SIU
Level: 216

Ranked Solo
GRANDMASTER I
422 LP / 165W 118L
Win Ratio 58.3%

Champion	CS	KDA	WR	Games Played
	188.7 (7.0)	1.71:1	63%	204 Played
	76.3 (2.9)	1.69:1	56%	25 Played
	146.4 (6.0)	1.19:1	43%	7 Played

Recent Games

10G 2W 7L
0.86:1
KDA
28%KP

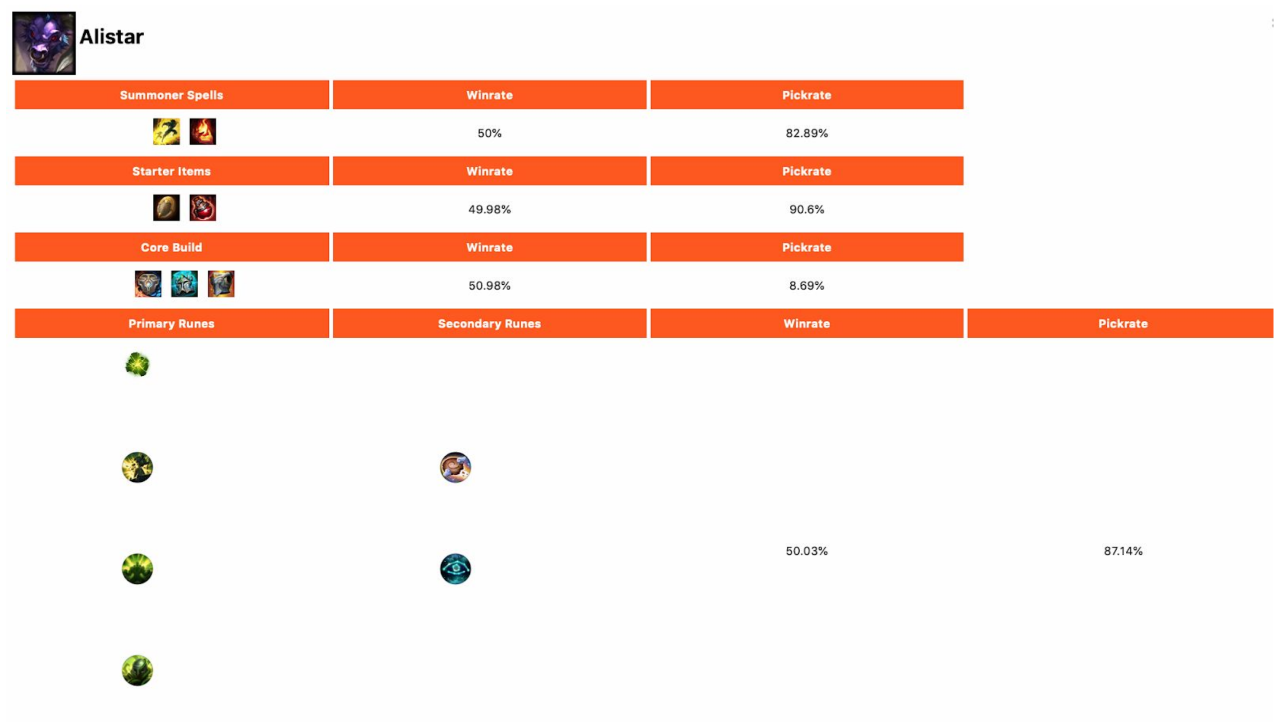
Past Ranks

S4	Platinum 4
S5	Platinum 2 0LP
S6	Diamond 5 19LP
S7	Diamond 4 23LP
S8	Diamond 4 0LP
S9	Diamond 2 0LP

Figure 2.3 A live example from the BlitzPlus web application of User Story / Feature 2 - Screenshot shows 2 player profiles displayed simultaneously. **(Notice that 2 player profiles can be seen on the same page and you can search 2 summoners at a time)**

3. As a summoner, I want to know what the most popular and most successful item/rune combinations are on the champions that I play so that I can maximise my performance and improve my win rate.
(Runes are relatively minor enhancements for a champion's abilities that are taken in the pregame screen and items are bought during the game and can greatly enhance a champion's ability.)

Scenario: View the most popular and most successful rune/ item combinations taken for each of the different playable champions
Given: I am on the BlitzPlus home page
And: I type a Champion's name in the Champion Search Bar
When: I click search
Then: I should see a dashboard showing the different rune/ item combinations along with their win rates.



Alistar

Summoner Spells	Winrate	Pickrate
	50%	82.89%

Starter Items	Winrate	Pickrate
	49.98%	90.6%

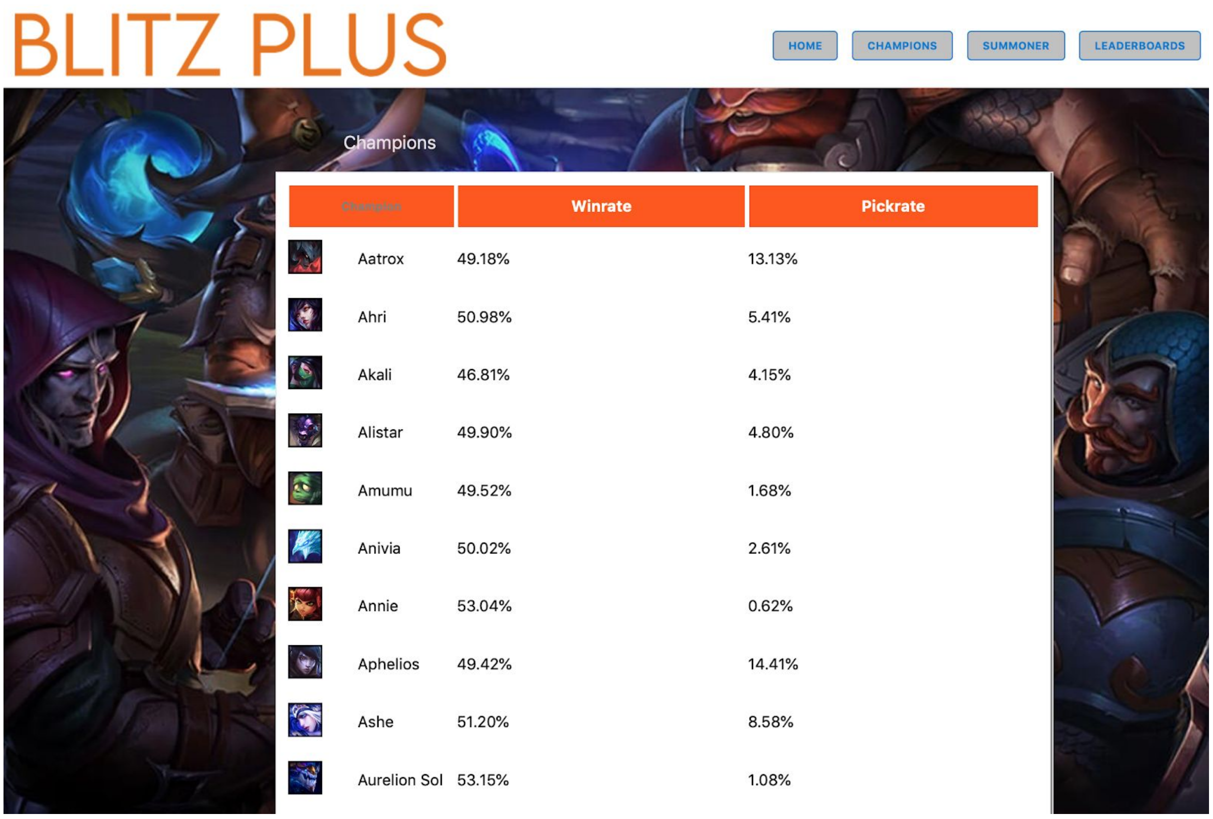
Core Build	Winrate	Pickrate
	50.98%	8.69%

Primary Runes	Secondary Runes	Winrate	Pickrate
		50.03%	87.14%

A Live example from the BlitzPlus web application of User Story / Feature 3 - Screenshots shows runes, items, spells to take on a champion and their corresponding statistics.

4. As a summoner, I want to find out what champions are the most popular/have the highest win rate. So that I know how to play against them and/or play them myself to improve my winrate.

   Scenario: Find out which Champions have the highest win rate/ pick rate in the game currently.
   Given: I am on my summoner dashboard
   When: I click on "Leaderboards"
   Given: I am on the Leaderboards page
   When: I click on Champions (in screen)
   Then: I can see the best players at a particular champion

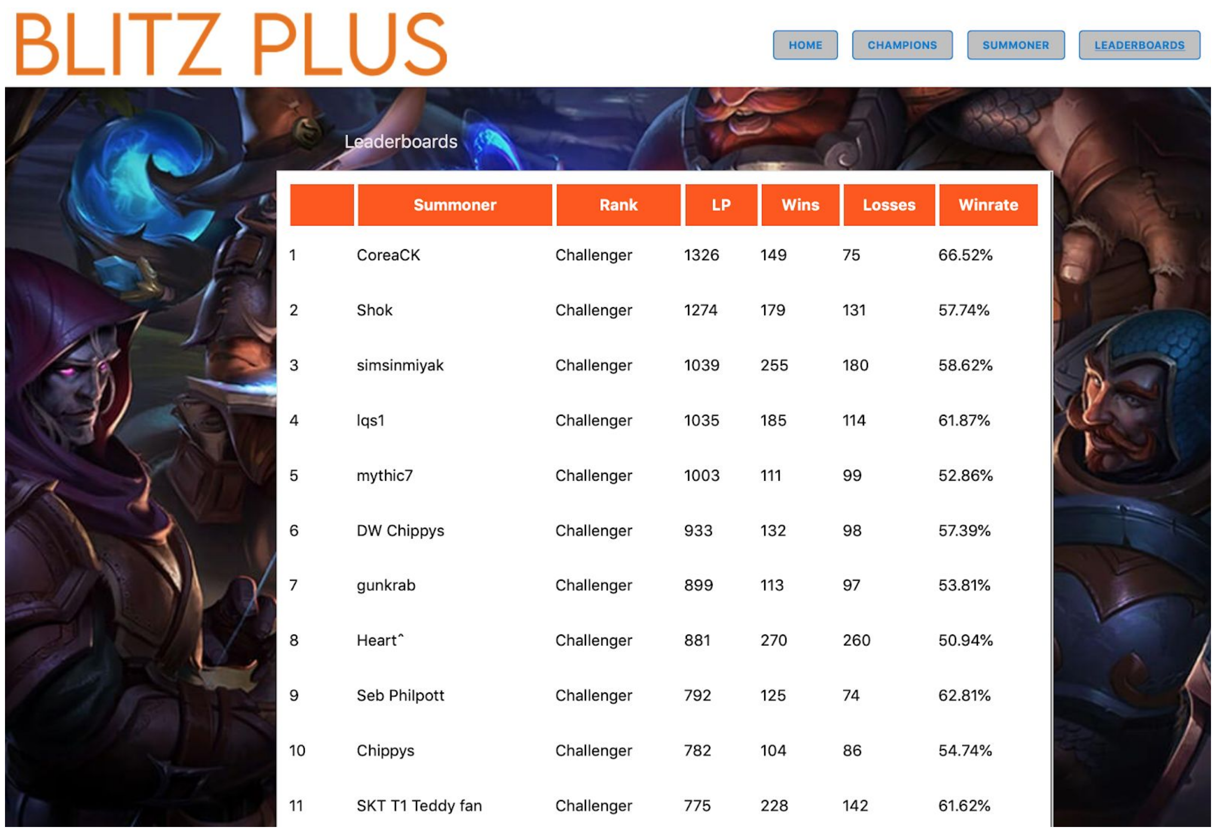


Champion		Winrate	Pickrate
	Aatrox	49.18%	13.13%
	Ahri	50.98%	5.41%
	Akali	46.81%	4.15%
	Alistar	49.90%	4.80%
	Amumu	49.52%	1.68%
	Anivia	50.02%	2.61%
	Annie	53.04%	0.62%
	Aphelios	49.42%	14.41%
	Ashe	51.20%	8.58%
	Aurelion Sol	53.15%	1.08%

A live example from the BlitzPlus web application of User Story / Feature 4 - Table of champions and their winrate/pickrate. Table can be sorted on pickrate and winrate. Clicking on a row will display more info about that champion.

5. As a summoner, I want to find out who is the best at a particular champion So that I can look up their gameplay and learn and improve from them.

   Scenario: Identity which summoners are the best at a particular Champion (have the highest win rate/ most played games on that Champion).
   Given: I am on my summoner dashboard
   When: I click on the “Champions” menu
   Then: I should be on the Champions overview page that displays the Champion Rankings

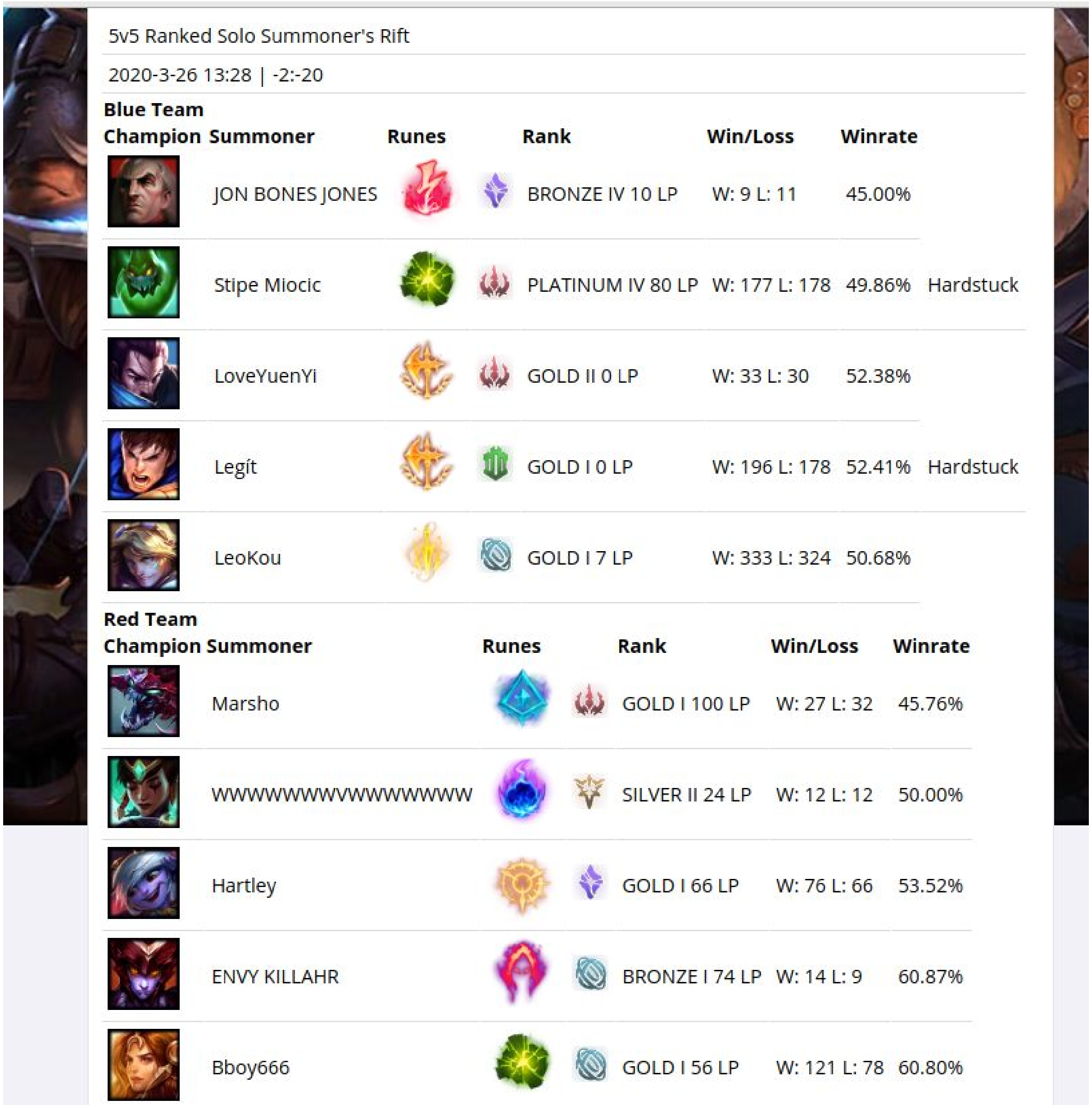


	Summoner	Rank	LP	Wins	Losses	Winrate
1	CoreaCK	Challenger	1326	149	75	66.52%
2	Shok	Challenger	1274	179	131	57.74%
3	simsinmiyak	Challenger	1039	255	180	58.62%
4	lqs1	Challenger	1035	185	114	61.87%
5	mythic7	Challenger	1003	111	99	52.86%
6	DW Chippys	Challenger	933	132	98	57.39%
7	gunkrab	Challenger	899	113	97	53.81%
8	Heart^	Challenger	881	270	260	50.94%
9	Seb Philpott	Challenger	792	125	74	62.81%
10	Chippys	Challenger	782	104	86	54.74%
11	SKT T1 Teddy fan	Challenger	775	228	142	61.62%

A live example from the BlitzPlus web application of User Story / Feature 5 - Leaderboard of summoners and their champions.

6. As a summoner, I want to find out the skill levels of players in my current game, So I can change my game strategy accordingly to play to win.

   Scenario: I am in a game and I want to have an advantage over other players in terms of knowledge so I can win
   Given: I am on my summoner dashboard
   And: I am in-game
   When: I click on “Active Game” button
   Then: I should be able to see an overview of my teammates and enemies statistics in a table


5v5 Ranked Solo Summoner's Rift

2020-3-26 13:28 | -2:-20

Blue Team

Champion	Summoner	Runes		Rank	Win/Loss	Winrate	
	JON BONES JONES			BRONZE IV 10 LP	W: 9 L: 11	45.00%	
	Stipe Miocic			PLATINUM IV 80 LP	W: 177 L: 178	49.86%	Hardstuck
	LoveYuenYi			GOLD II 0 LP	W: 33 L: 30	52.38%	
	Legít			GOLD I 0 LP	W: 196 L: 178	52.41%	Hardstuck
	LeoKou			GOLD I 7 LP	W: 333 L: 324	50.68%	

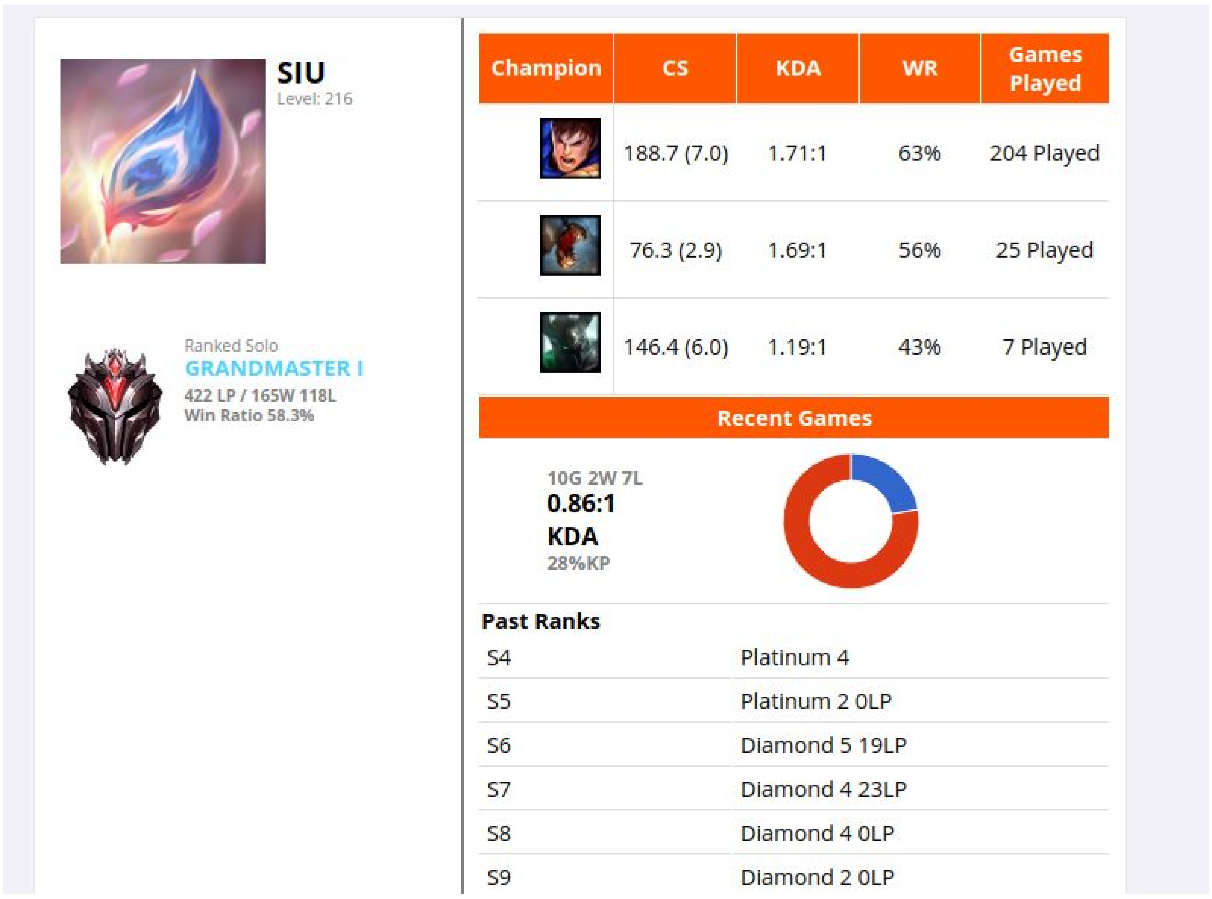
Red Team

Champion	Summoner	Runes		Rank	Win/Loss	Winrate
	Marsho			GOLD I 100 LP	W: 27 L: 32	45.76%
	WWWWWWWWWWWWWWWW			SILVER II 24 LP	W: 12 L: 12	50.00%
	Hartley			GOLD I 66 LP	W: 76 L: 66	53.52%
	ENVY KILLAHR			BRONZE I 74 LP	W: 14 L: 9	60.87%
	Bboy666			GOLD I 56 LP	W: 121 L: 78	60.80%

A live example from the BlitzPlus web application of User Story / Feature 6 - Live game feature showing the statistics of all players in the game and what champions/runes they picked.

7. As a summoner, I want to track certain statistics of mine and their changes over a period of time so I can track my improvement (this will most likely just be ranked rating)

   Scenario: I am a player dedicated to improving and I want to have a page where I can easily see my stats over a period of time to see improvement
   Given: I am on the BlitzPlus home page
   And: I type my summoner name in the summoner name search bar
   When: I click Search
   Then: I should see my summoner dashboard with a line graph indicating stats over a period of time


SIU
Level: 216

Ranked Solo
GRANDMASTER I
422 LP / 165W 118L
Win Ratio 58.3%

Champion	CS	KDA	WR	Games Played
	188.7 (7.0)	1.71:1	63%	204 Played
	76.3 (2.9)	1.69:1	56%	25 Played
	146.4 (6.0)	1.19:1	43%	7 Played

Recent Games

10G 2W 7L
0.86:1
KDA
28%KP

Past Ranks

S4	Platinum 4
S5	Platinum 2 0LP
S6	Diamond 5 19LP
S7	Diamond 4 23LP
S8	Diamond 4 0LP
S9	Diamond 2 0LP

A live example from the BlitzPlus web application of User Story / Feature 7 - This feature was implemented on the summoner page. Users can see past ranks on that page.

2.3. Project Constraints

This project requires a number of software engineers to work together in order to successfully complete the design and implementation of an operational web application. As such, there is a wide range of constraints, detailed below that will impact this project, that includes, scheduling, operational and technical factors.

Technical

There are several different technical constraints placed on the project due to technical limitations or feasibility. The team has had to compromise where possible to fit within different technical and non-technical constraints placed on the team during the project.

The Riot API wasn't able to provide all the functionality that we wanted for our website so we needed a backend to scrape the necessary data from a third-party website. Additionally, software architecture components were selected in accordance with what the majority of team members were already familiar with and not what may be most optimal to deliver this project.

Scheduling

A time constraint of 10 weeks has been placed on the project. The team had only 4 weeks to deliver a minimum viable product, this meant that the Fat Donkey team would only be able to test and experiment with their actual software design and implementation from week 6 after they had successfully canvassed all stakeholders to develop their user stories and brainstorm the features of the project. Another major scheduling constraint placed on the Fat Donkey team will be their availability and ability to work collaboratively with one another on this project due to the current global pandemic forcing everyone to remain under a federal government lockdown, causing team members to work in isolation and individual to one another.

Operational

The application had to be web-based and viewable using common web browsers. The application had to be designed to provide a good user experience for discovering and interacting with one or more information archives.

3. Design

This document details the Design and Code aspects of the project as a series of textual descriptions and diagrams outlining the conceptual design process, design alternatives and the detailed design. This document also depicts evidence of our functional web application, featuring screenshots of the final product and it's features.

3.1 Conceptual Design

The conceptual design and the software components behind the BlitzPlus web application can be found in Figure 1.1 (below). The frontend of the project was constructed using:

- HTML to create the elements of the webpage
- CSS to present the webpage in an aesthetic manner
- Javascript to add interactivity and movement to our HTML pages.

These choices of language for our frontend had been made simply because HTML, CSS and Javascript are cornerstones of modern web design and were also very familiar to most of our team members as we had prior experience in these languages, allowing for us to spend less time "learning" the language and more time constructing the frontend for our app. The Riot API also returns JSON objects which can be handled with Javascript.

Initially the Fat Donkey development team planned to have the BlitzPlus web application designed and built around the Django framework, however, during the production phase of the project we moved away from Django in favour of Flask. Flask is a python module that allowed us to run a webserver needed to run dynamic websites. The decision was also made because Django was found to be a backend framework that is most commonly used to help developers to build large and complex web applications. Whereas Flask accelerated development of simple web applications by providing the required frontend functionality. The fact that we didn't end up having a database and backend, made the choice of using Flask very clear. Additionally, the specification of this project was not to make a highly complex software program but rather to build a web application that could help users to navigate through information sources and therefore the team's priorities were focused more on the project's user interface and the consumer's user experience whilst using the BlitzPlus web application.

As a team we decided to use a web scraper known as scrapy to collect and compile statistics relating to the game. This is mostly due to the shortcomings of our given API, which failed to provide certain data that would allow us to solve our requirements because it did not provide statistical analysis of the game such as the winrate and pickrate of champions. We considered doing the analysis ourselves but it was too much of a task and required large-scale data analysis. This issue was identified during a critical stage of the development process and ultimately led to the decision to alter the design of the project and shift from a system with its own database to one that scraped third party sites for the data that we needed.

Software Architecture Diagram of the BlitzPlus project

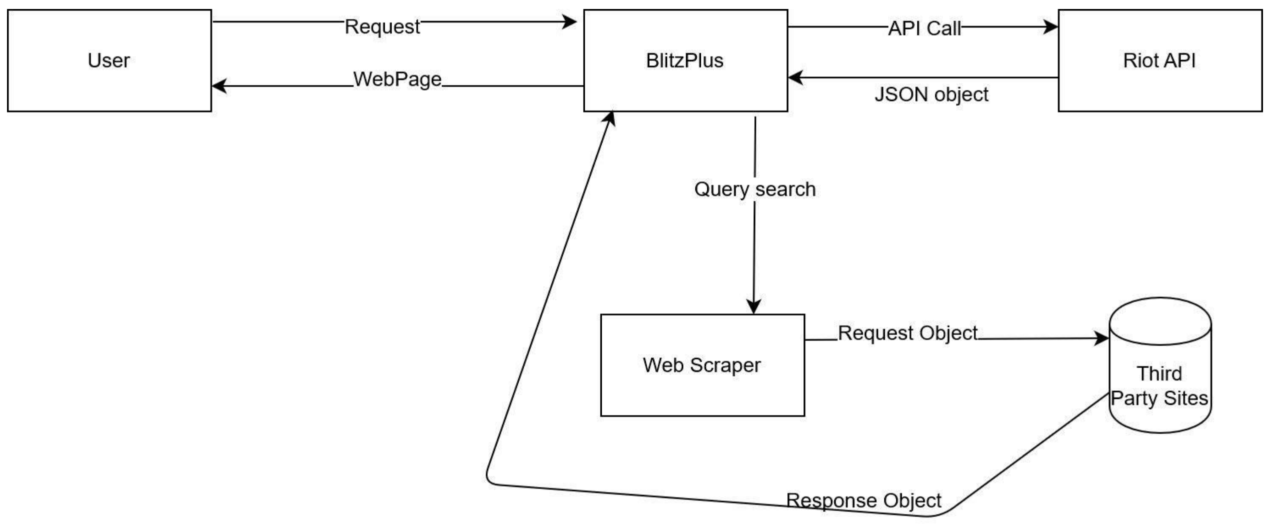


Figure 1.1: Diagram representing the software architecture and components utilised during this project

The Sequence and Interaction Diagrams that show how elements within the web application will interact for each of the use cases can be found (in the Figures 1.2 -1.9) below.

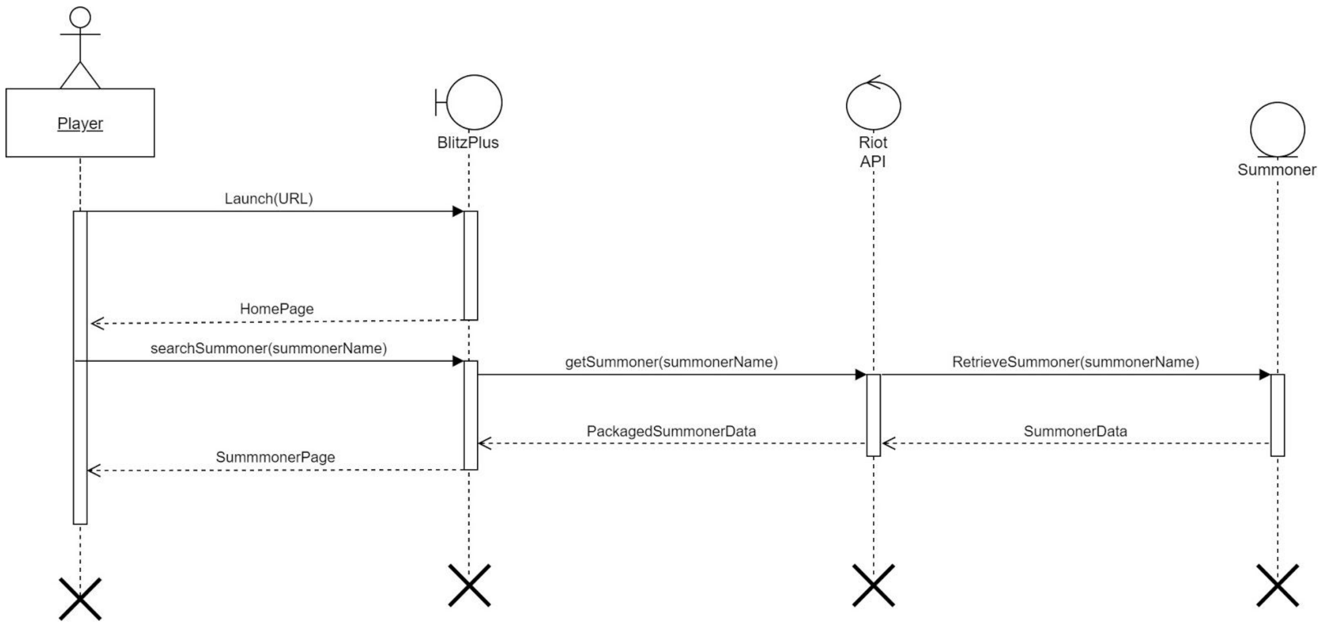


Figure 1.2: Diagram representing the sequence and interaction diagram for User Story 1.

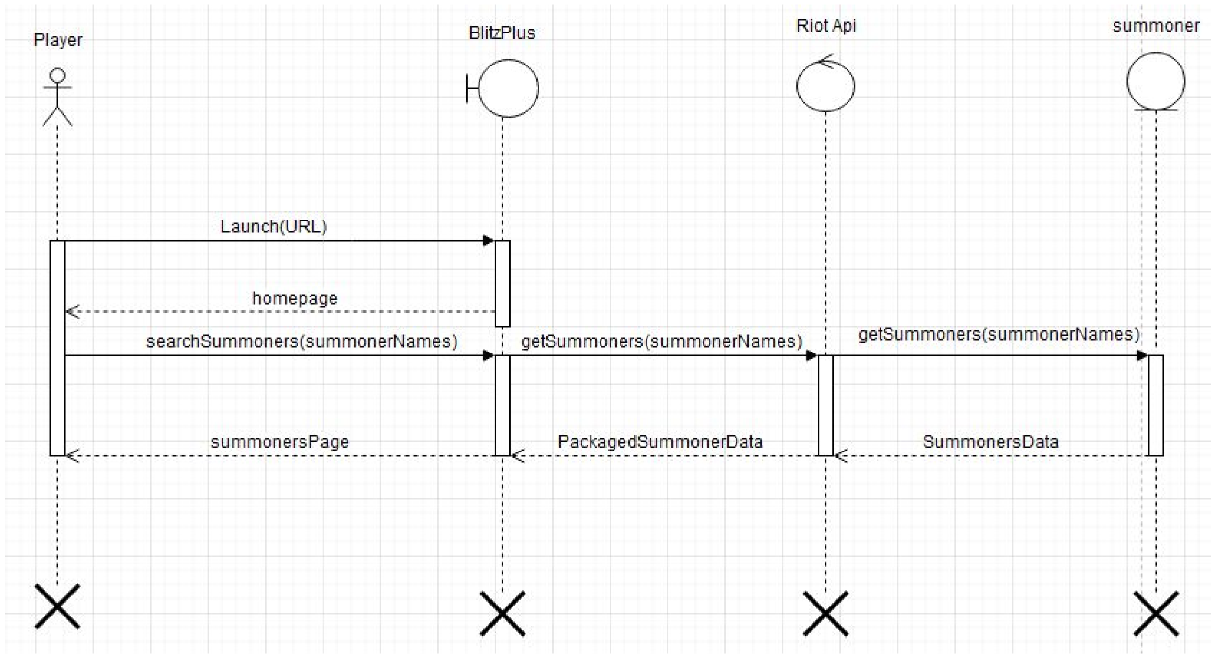


Figure 1.3: Diagram representing the sequence and interaction diagram for User Story 2.

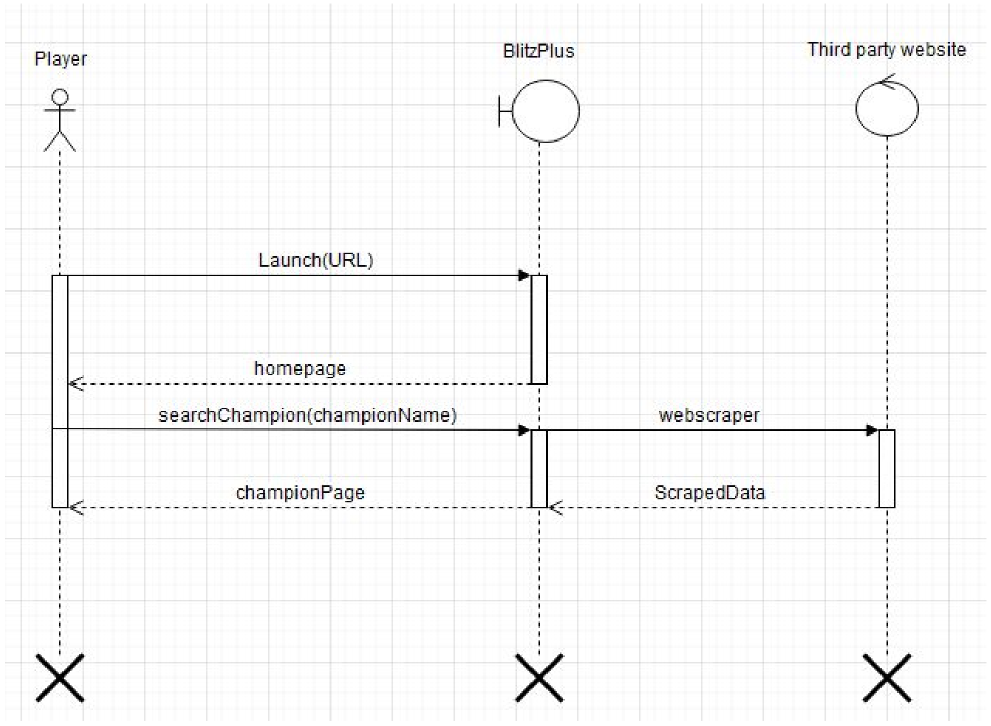


Figure 1.4: Diagram representing the sequence and interaction diagram for User Story 3.

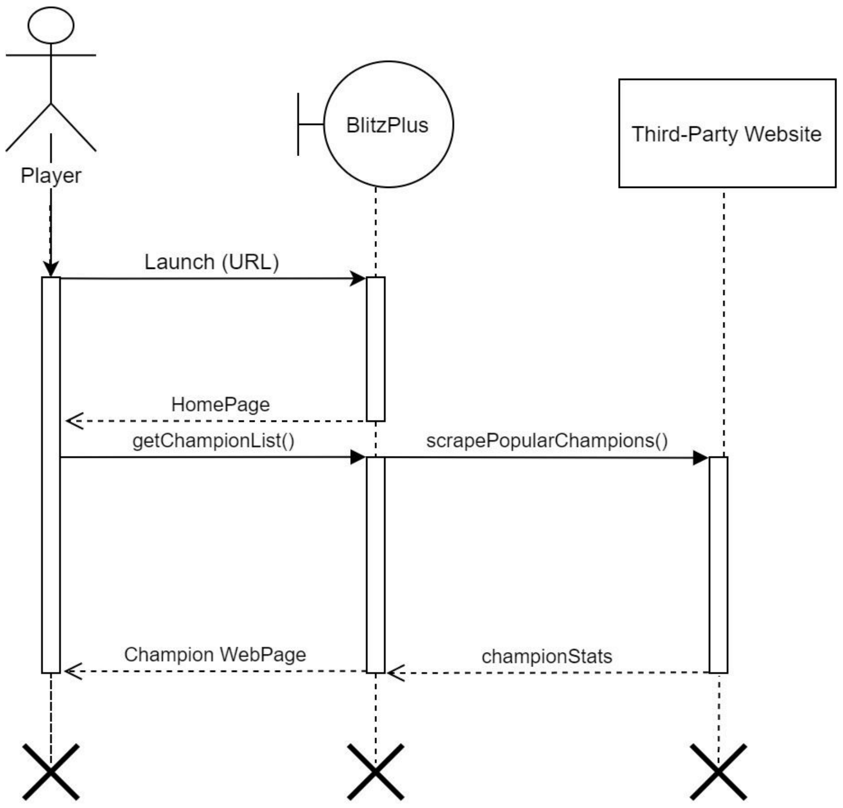


Figure 1.5: Diagram representing the sequence and interaction diagram for User Story 4.

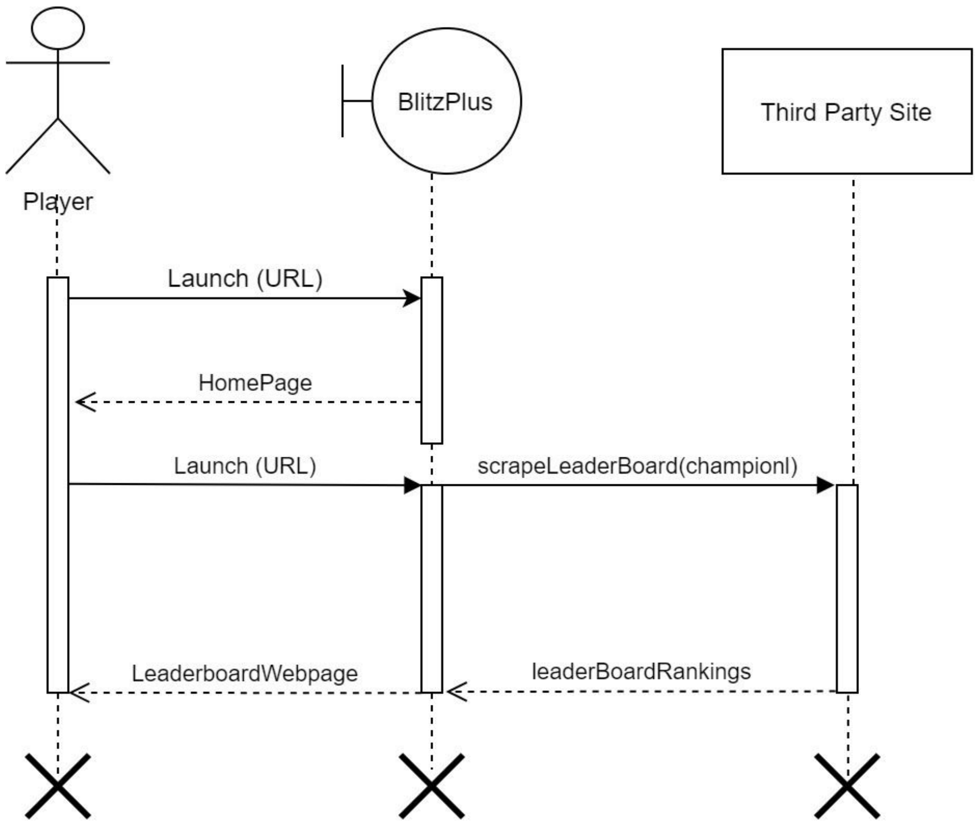


Figure 1.6: Diagram representing the sequence and interaction diagram for User Story 5.

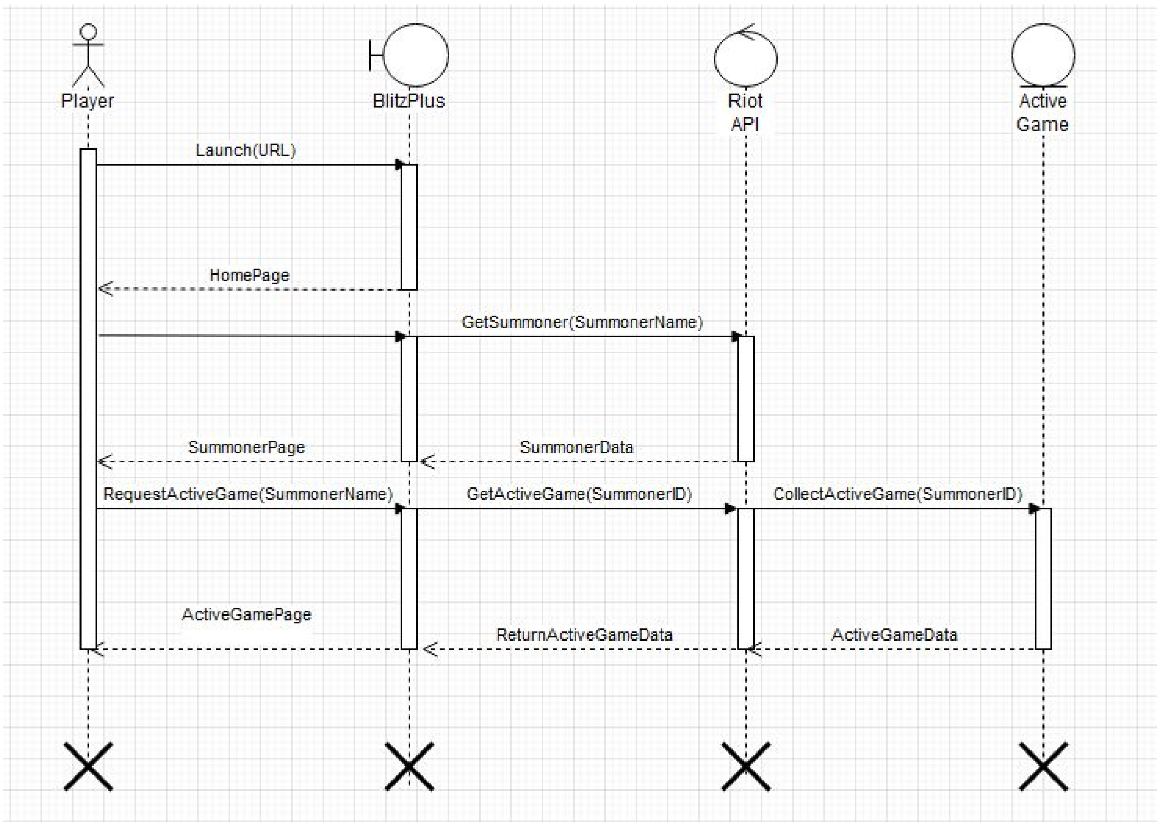


Figure 1.7: Diagram representing the sequence and interaction diagram for User Story 6.

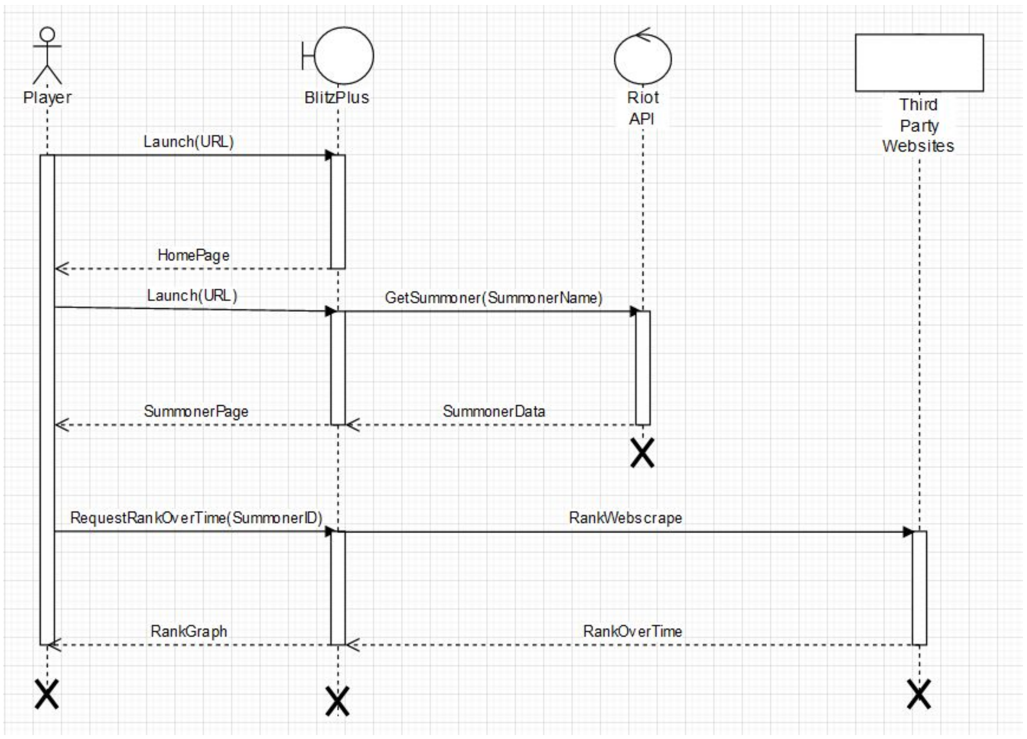


Figure 1.8: Diagram representing the sequence and interaction diagram for User Story 7.

3.2 Design Alternatives

During this project, the Fat Donkey Developers also discussed integrating highlights packages and in-game footage of each Champion and Summoner, to enhance the consumers experience whilst using the web application. To incorporate this idea into the project, a second API would need to be integrated into the BlitzPlus project. The API of choice was the popular '*Twitch API*', because it is able to provide relevant data in an easy, consistent and reliable way, as opposed to other similar API's that would have been less reliable. The '*Twitch API*' allows for a substantial data transfer of 800 points (tokens) per minute and similar to our already existing, '*Riot API*', it returns JSON objects which we would be able to easily integrate into our project (as detailed in *Figure 2.0* below). Once the Controller (BlitzPlus) made the API call to the '*Riot API*' and the query search to the web scraper to collect all of the in-game statistics and results of each Champion and Summoner, the Controller (BlitzPlus) would then make the another API call to the 'Twitch API' to gather all of that Champion and Summoner's in-game highlights to be displayed on their player profile. Once all of this information had been gathered and collated it would be sent to the client-side technologies to package it together in a friendly and easy to view manner for the user. Unfortunately, during the development phase of this project, there was a major update to the '*Twitch API*' which resulted in this additional feature having to be removed. Prior to the update the additional API was not causing any major issues, however, once the update occurred, the decision to scrap this idea was due to severe time constraints and ensuring that the project was able to successfully deliver what it outlined from its initial design and to meet the project specifications of designing and building a web application that can help users to navigate through information sources.

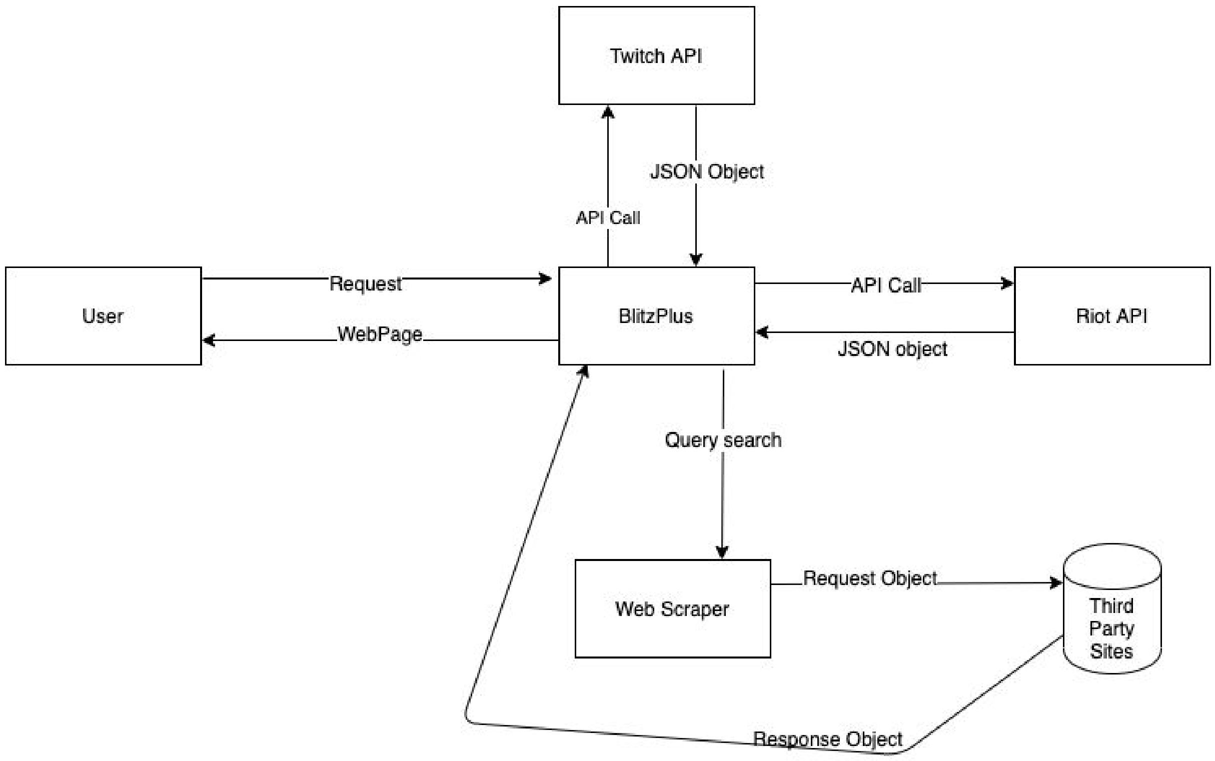


Figure 2.0: MVC Component Diagram of Design Alternative including Twitch API

3.3 Detailed Design and Summary

This section explains in detail the different components of the final design accompanied with diagrams, screen captures and code examples.

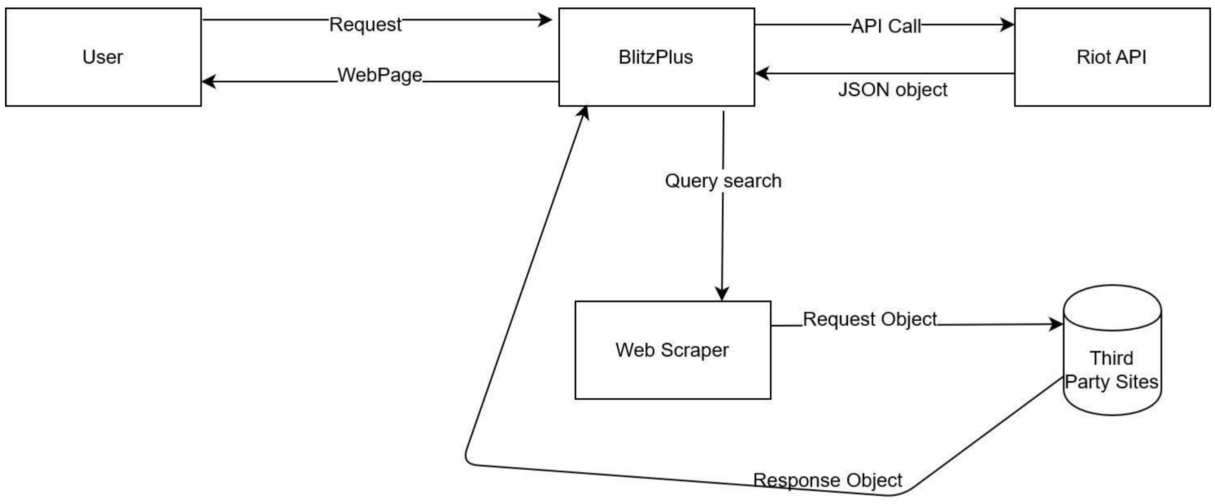


Figure 2.1 BlitzPlus final software architecture design

Figure 2.1 (above) depicts the BlitzPlus web application's final software architecture and the components within the application.

During the design process, there was a decision made to streamline the web application and remove the 'Feedback' feature. The decision to remove the 'Feedback' feature (detailed in Figure 1.5) was ultimately made because it did not align with this application's objectives and would not benefit the consumers user experience. The BlitzPlus web application was designed to highlight vital information and make it easily accessible to enhance the in-game experience of the playing community. Once the production phase of the project began, there was a realisation amongst the Fat Donkey development team that this feature would not benefit the playing community because the feedback they provided would be relating to their gameplay experience and not the statistics they would be viewing on the BlitzPlus web application. Additionally, players already have the ability to contact the development team responsible for the maintenance and improvement of the '*League of Legends*' game within the client application. This coupled with the time restriction on delivering this project resulted in the 'Feedback' feature having to be removed from the final design of this product.

4. Key Achievements

Overall the BlitzPlus project was a success and the Fat Donkey team are proud to have been able to deliver a system for the world's largest online video game and make vital information, currently not available to users within the game's client available and easily accessible to them to enhance their in-game experience. The biggest satisfaction from working together on this project was knowing we created an important and relevant extension to one of the world's most diverse online communities by creating a connection between players and enthusiasts where they can openly connect, contribute and analyse vital in-game information to be used during their next gaming experience.

We were able to successfully implement all the planned features of our website. Our prototype is easy to navigate and very intuitive to use. It also has a distinct and aesthetically pleasing appearance. The team improved and consolidated their frontend web development skills and the use of webscrapers.

5. Team Organisation and Appraisal

Initially the team worked together on soliciting and analysing the requirements of the system and writing the relevant documentation (Deliverable 1 and 2). When it came to developing the website and working on the final deliverables, team members specialised in specific tasks as follows:

- Flynn was the team leader, implemented the functionality of the website and finalised the report and speech.
- Edwin improved and customized the appearance of the website and improved our code design.
- Ben wrote the final report and speech by extracting parts of previous deliverables and updating them
- Frank produced the final demonstration video and wrote the presentation slides

Throughout the development of website, the team as a whole discussed the design of the website and possible features

An improvement we could have made on this project was to have a better developed prototype for deliverable 3 as we did not score very well in that assessment. The development team did not expect that a rather substantial prototype was needed for that deliverable since there was only a week between when deliverable 2 and 3 was due. In retrospect, the team should have started the development earlier or used a wireframe/website builder.

Another area of improvement was the fact that we should have used a frontend framework such as React instead of using plain javascript. Using a framework would have accelerated our development and allowed us to expand the functionality of our website. The main reason we did not use a frontend framework was because we had only one week to learn the new language and develop a substantial prototype. Better foresight and time management would have helped us in this regard.

Overall, we believe the Fat Donkey team came together quickly and worked tirelessly to deliver this project through strict time constraints and increasingly straining working conditions due to the current state of the world.

Video demo:
https://drive.google.com/file/d/1F2lOlMFRpnfelgAYp6pYw8cYT5d2dpwy/view?fbclid=IwAR2OvI6vI8qgpg2cGuUHoSCmGMvIRKW3qGhjR_06xr6lLjdA6i8GuaRiPYs