“League of Legends” is a highly competitive information based game, where every bit of possible information is vital to increase a player’s chances of winning. BlitzPlus aims to be an analytics website which provides players of the game “League of Legends” access to data and statistics which would otherwise be unavailable, thus hoping to increase a player’s chances of winning.

Part 1: Software Architecture

Given the current user stories of what the team has decided to do, what would be the resulting software architecture? The software architecture elements are:
1. What external data sources will your system be accessing?
Riot Developer API for League of Legends. Through this API we will be able to fetch all the statistics regarding summoners and games.

2. Software components: the selected Web stack showing major software components that comprise your solution. These will include both components that need to be developed and third-party components (e.g. web browser).

Frontend (The Website)

- **HTML** to create the elements on the webpage
- **CSS** to change the appearance of the HTML elements
- **Javascript** to manipulate HTML and CSS and to make the website interactive.
- **Chrome** to load and test our website.
- **Python3** to run our frontend server.

We realised that the **Riot API** isn’t able to provide all the functionality that we want for our website so we need a backend to scrape the necessary data from a third-party website. We believe that this is the optimal solution since we are limited in our API calls per second and do not have the resources/time to build algorithms that analyse thousands of games. For our backend, we will use the **Django** web framework since it is written in python and supports MVC patterns which keeps the user interface and business logic layers separated. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. We will use the **Scrapy** web crawler to scrape the data off other websites (such as **Op.gg** which perform a similar goal**)** since it is also written in python.

3. Relating choices to components: decide which language should be used for which component of the software architecture.

HTML, CSS and Javascript are the cornerstones of modern web browsing so they are obviously the most appropriate tools to use for our website. Since the Riot API returns JSON (Javascript Object Notation) response messages which can be handled with Javascript. We are going to use Javascript and not a typescript because some of our members are unfamiliar with front-end development (Javascript) and none have used a typescript before. We believe it will be more beneficial if we use a language that some members already know since they can teach the other members and we are under time constraints in the development stage (roughly 2-3 weeks?).

For the backend components we are obviously going to use Python to write our web crawler with Scrapy and to implement the Django web framework.

4. The choice of an implementation/technology or framework: teams must decide now on a suitable language or combination of languages to implement a prototype. This will be largely determined by the Web stack but at the same time you can make variations

As mentioned before, we will be using the Django web framework to make it as easy as possible to implement our web application.

- Javascript - will be used to handle the JSON objects that is passed through to the web application and used to animate the web elements on our website.
- HTML5 - HTML5 is the standard markup language for creating web pages and will be
- used to generate the basic elements on the website.
- CSS - CSS is a stylesheet language that will be used to style the elements on the webpage in order to make it look aesthetically appealing and professional
- Python - is a scripting language to write web applications with clean, readable, and maintainable code base. Developers have to use Python web frameworks such as Django to accelerate development and deployment of custom web applications.

5. The choice of a platform: decide on machine or machines requirements (Linux, Windows etc.) for the final system.
Our website should work on Linux since we are developing it on CSE's VLAB. However, our website should work on a majority of popular browsers (Firefox, Chrome, etc.) regardless of the operating system.

6. Make a summary of the key benefits/achievements of your architectural choices

Software architecture is an extremely important aspect of any software project. The decisions made within this 'BlitzPlus' project to include specific technologies and software architectural structures were based on what was most beneficial to the team of developers, to ensure the project was delivered on time and to the highest standards. Our choices were based on the following key aspects:

- Familiarity with the chosen architecture and technologies
- Scalability
- Maintainability

- Reusability

Using an architecture that was already familiar to the developers was the main consideration during this stage of the ‘BlitzPlus’ project. The major benefit of using an architecture that was familiar to the developers was the project’s Time to Market. The strict time constraints to design, implement and produce this project meant that there was no additional time for developers to research or to learn new technologies. Additionally, using a familiar architecture meant that the developers were able to bring their pre-existing knowledge to this project, meaning there is a solid foundation for the developers to begin working from.

The project’s framework was decided using the Model-View-Controller (MVC) application design model, which is common in modern user interface software projects. There was a strong emphasis throughout the decision making process of which architecture, technologies and frameworks to use within the ‘BlitzPlus’ project around the reusability and maintenance of the code base. The purpose of the MVC model is for reusability and maintaining code, utilising modular components and low coupling throughout a project. Ultimately, this lead to the decision to include Django within this project as it will ensure developers are able to separate the user interface and the business logic layers, benefitting the project’s reusability and maintainability.

Furthermore, the decision to use a web crawler, Scrappy, also benefits this project in terms of scalability, maintainability and performance. Instead of having to spend time and resources developing and implementing our own algorithms to analyse thousands of games and other large datasets. Not having to develop our own algorithms and relying on existing software also benefits this project because it is easier to maintain existing software than it is developing your own. The decision to use Scrappy ultimately means the reduction in the complexity of the code and it will be easier for developers to find bugs or anomalies within their code as they are initially using the existing tested software and code to build on top of.

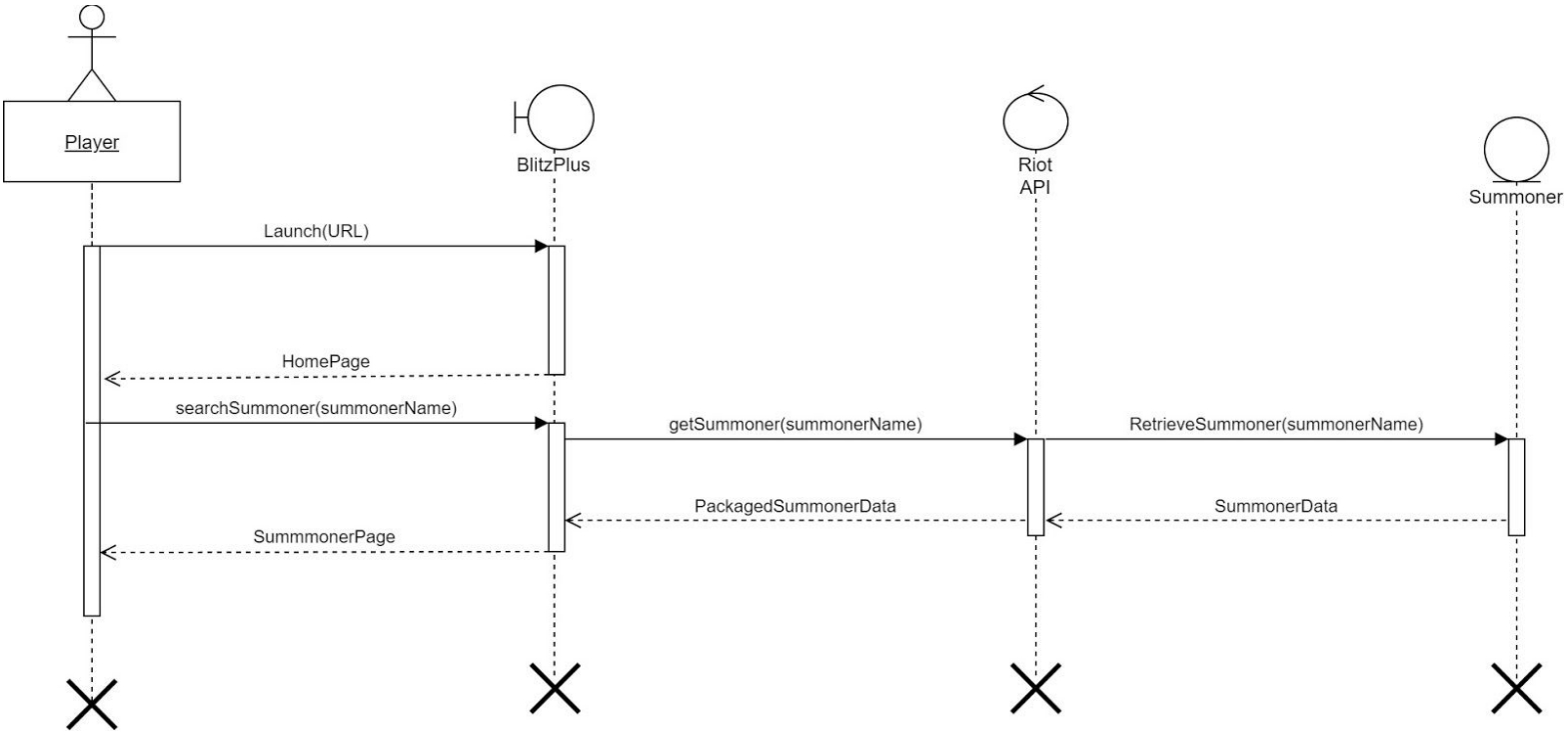
Part 2: Initial Software design

1. The updated list of stories or use cases must first be presented.

Feature: Players can search statistics and results from a summoner’s games (1) (Flynn)

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 winrate.

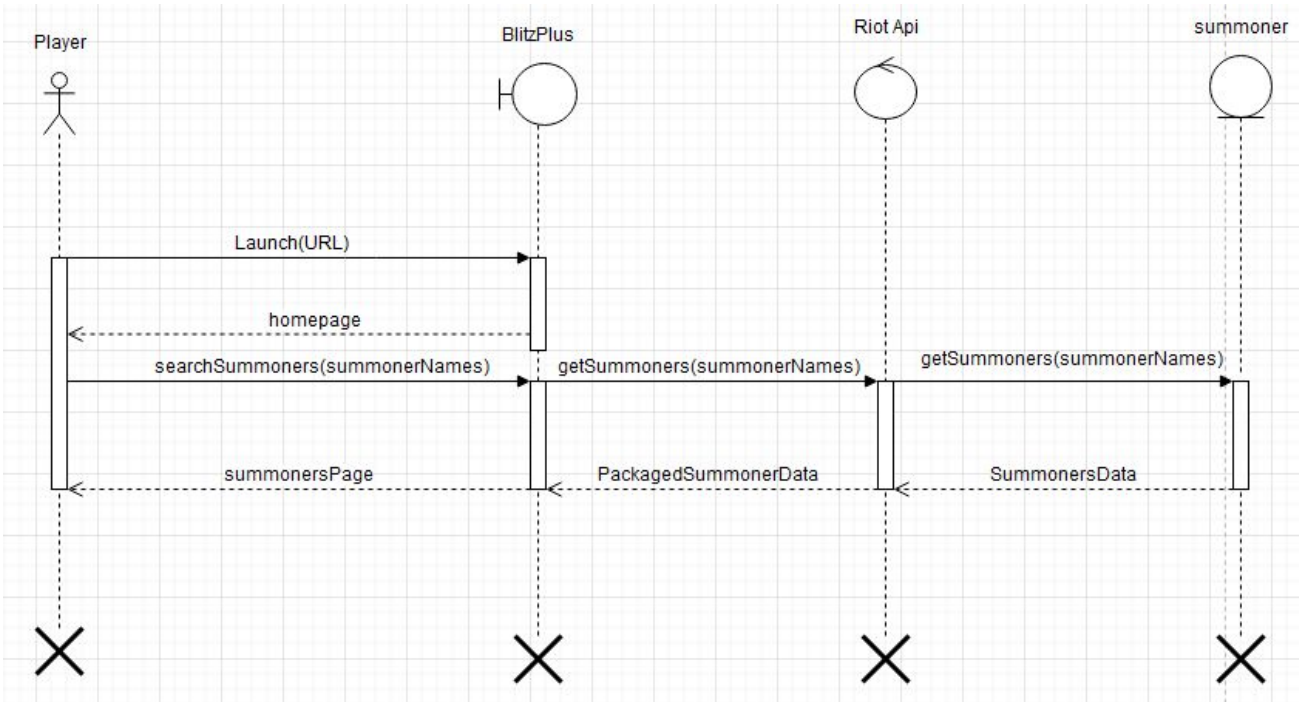


****

Feature: Players can search and compare statistics and results from other summoner's games (2) (Frank)

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 games 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 dashboard with a comparison of match history, rank and winrate.

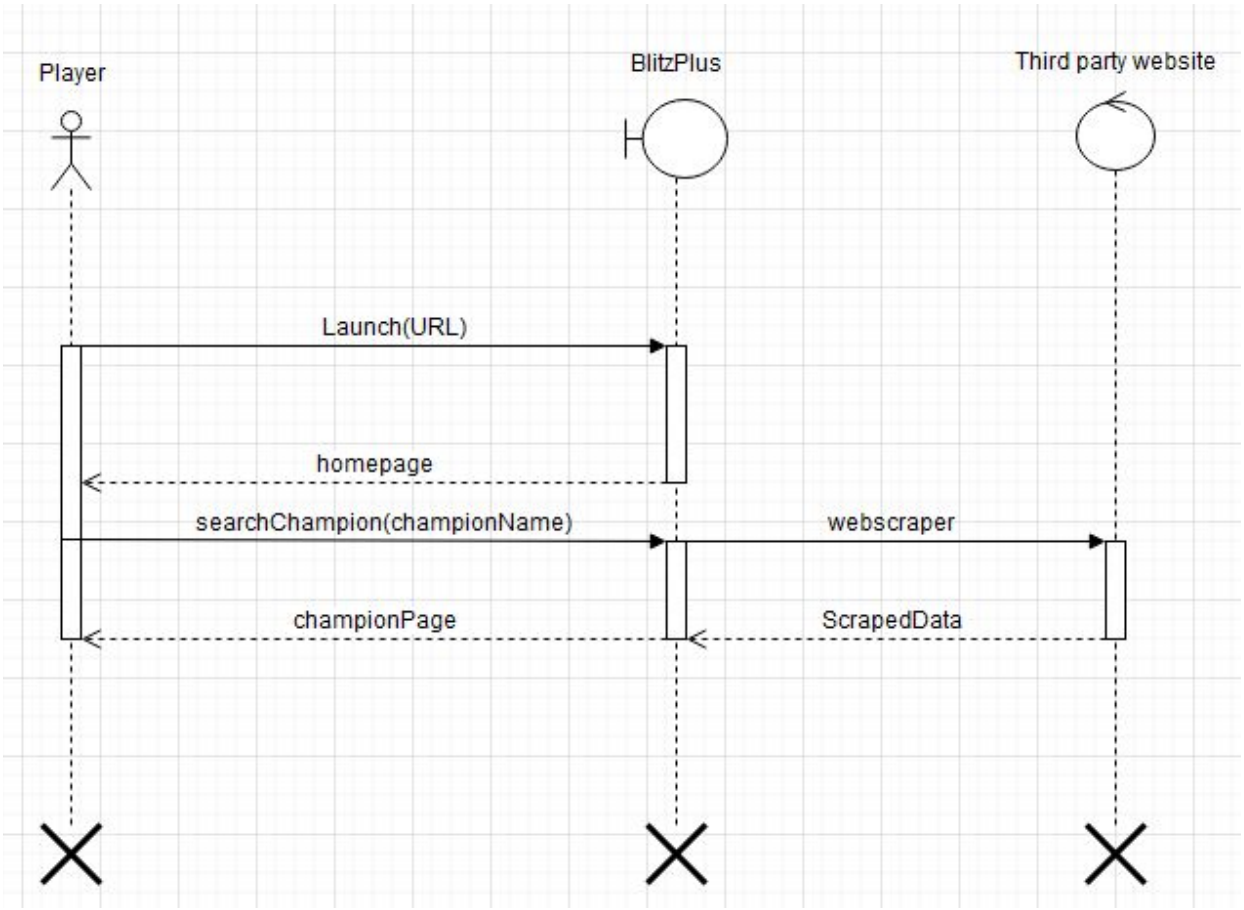


****

Feature: Players can find out what the most popular items/runes are on each champion (3) (Frank)

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 winrate.
(Runes are relatively minor enhancements for a champions 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.

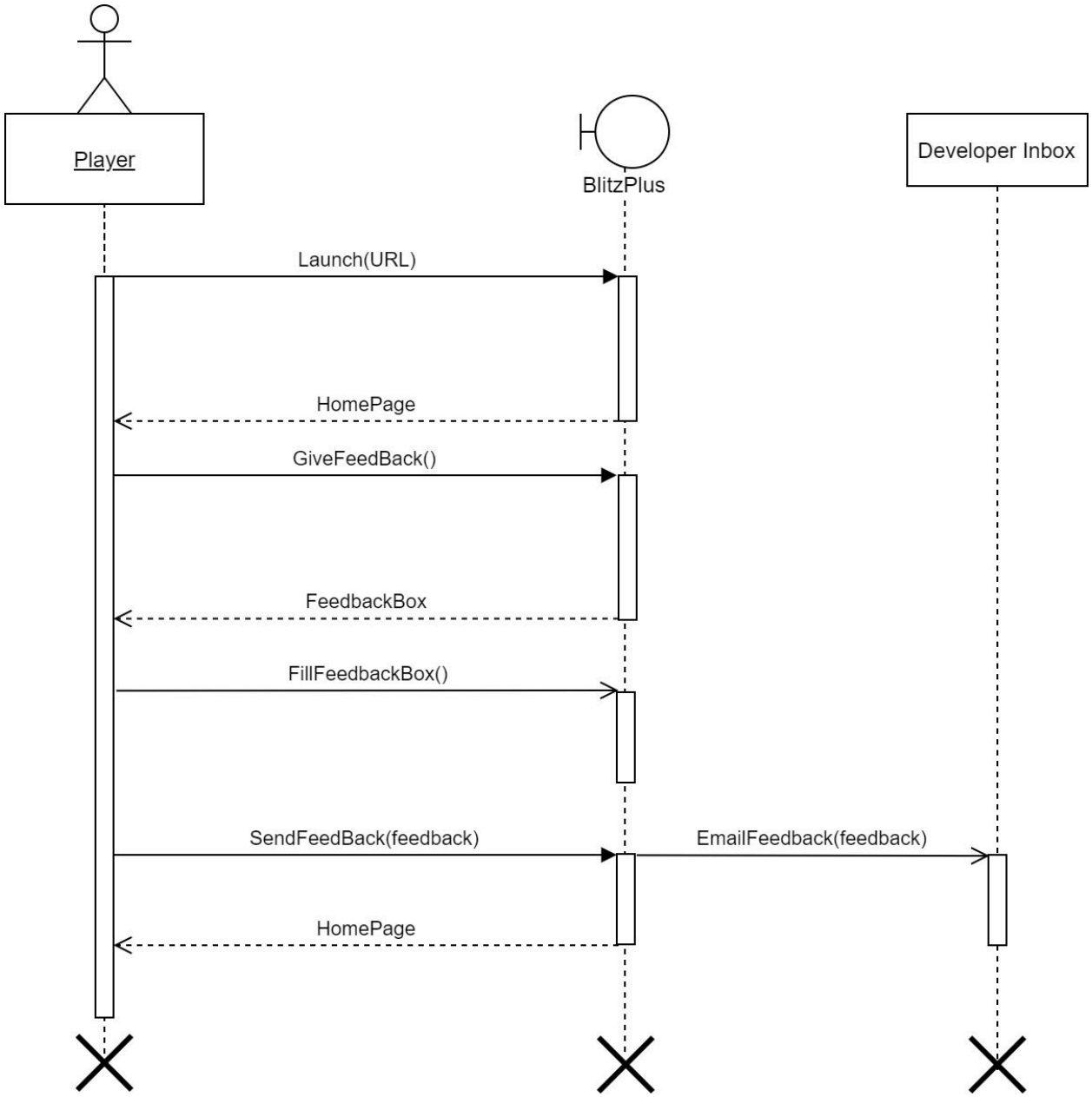


****

Feature: Players can provide feedback on the website (4) (Flynn)

As a developer I want a website where all players and fans can go and provide feedback on the website so that I gain new ideas and make improvements in the next iteration of the website.

Scenario: Provide feedback on the website
Given I am on the BlitzPlus home page
When I click on the “Give Feedback” button
Then I should see a text box pop up where I can type my feedback.
When I click “Send”, my feedback will be sent to the developers.



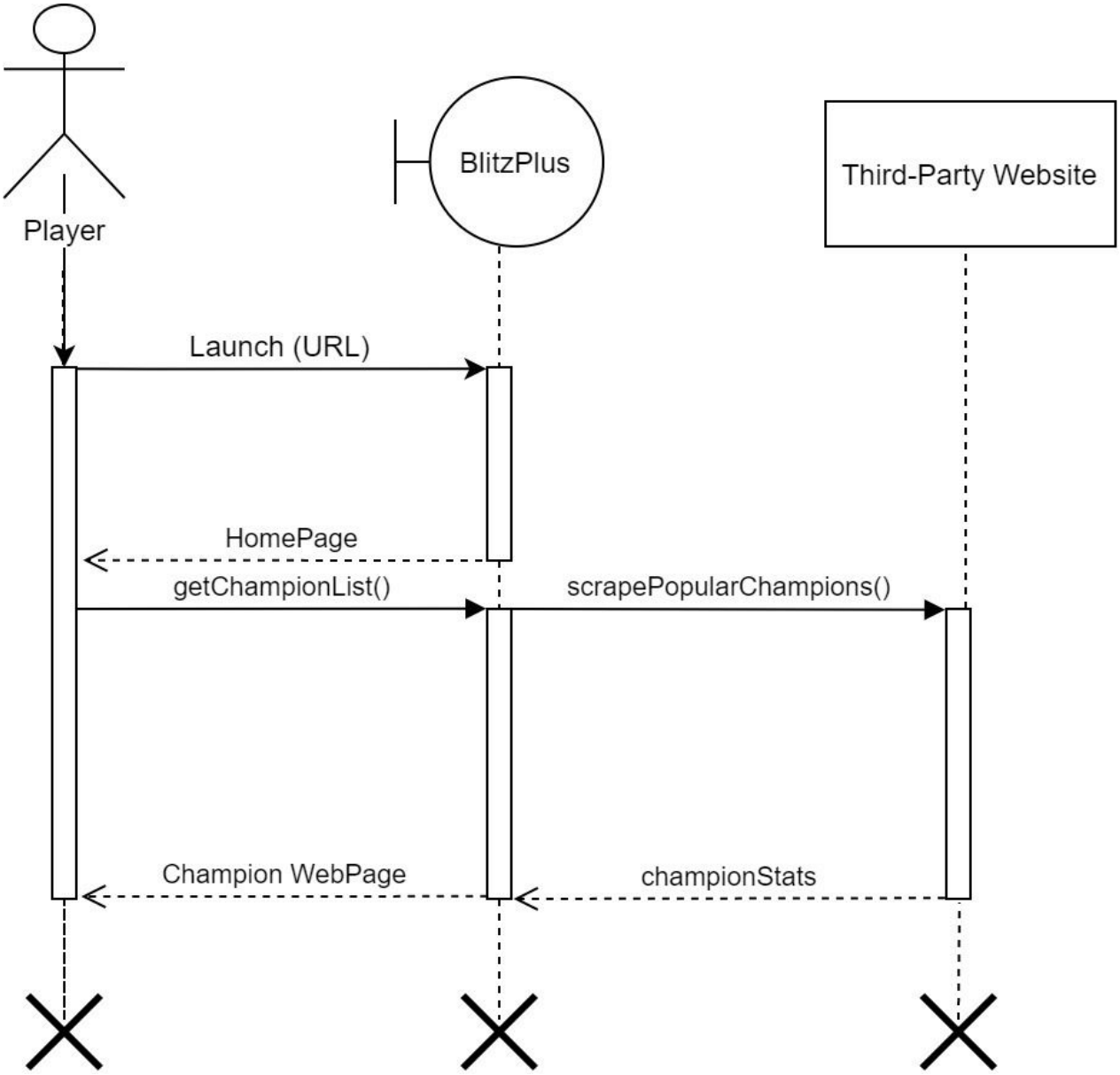
****

Feature: Players can find out what the most popular champions are in current meta-game. (5)(Ben)

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 WR/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

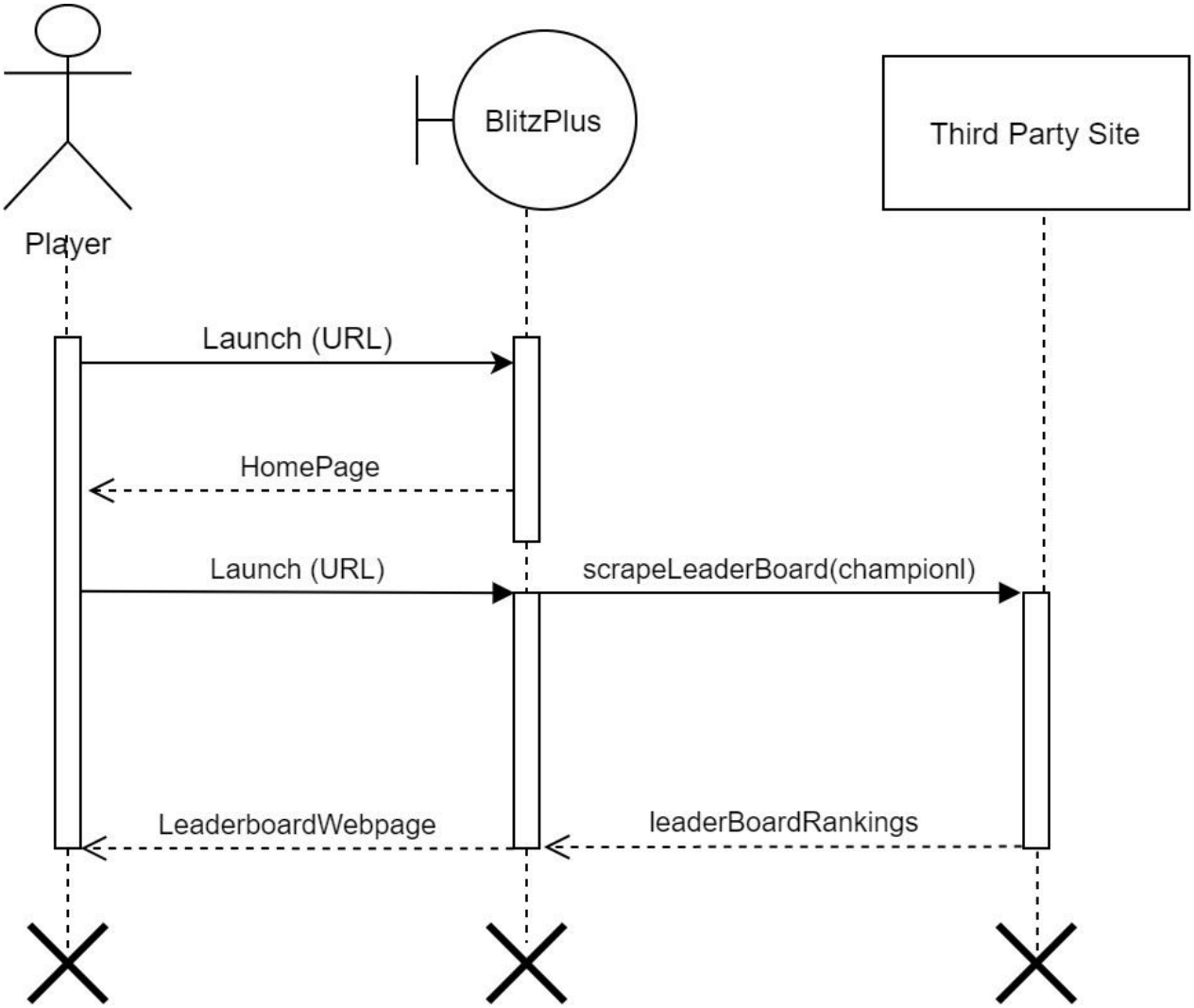


****

Feature: Players can find out which summoners are the best at particular champions (6) (Ben)

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: Identify which summoners are the best at particular champions (have highest winrate/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



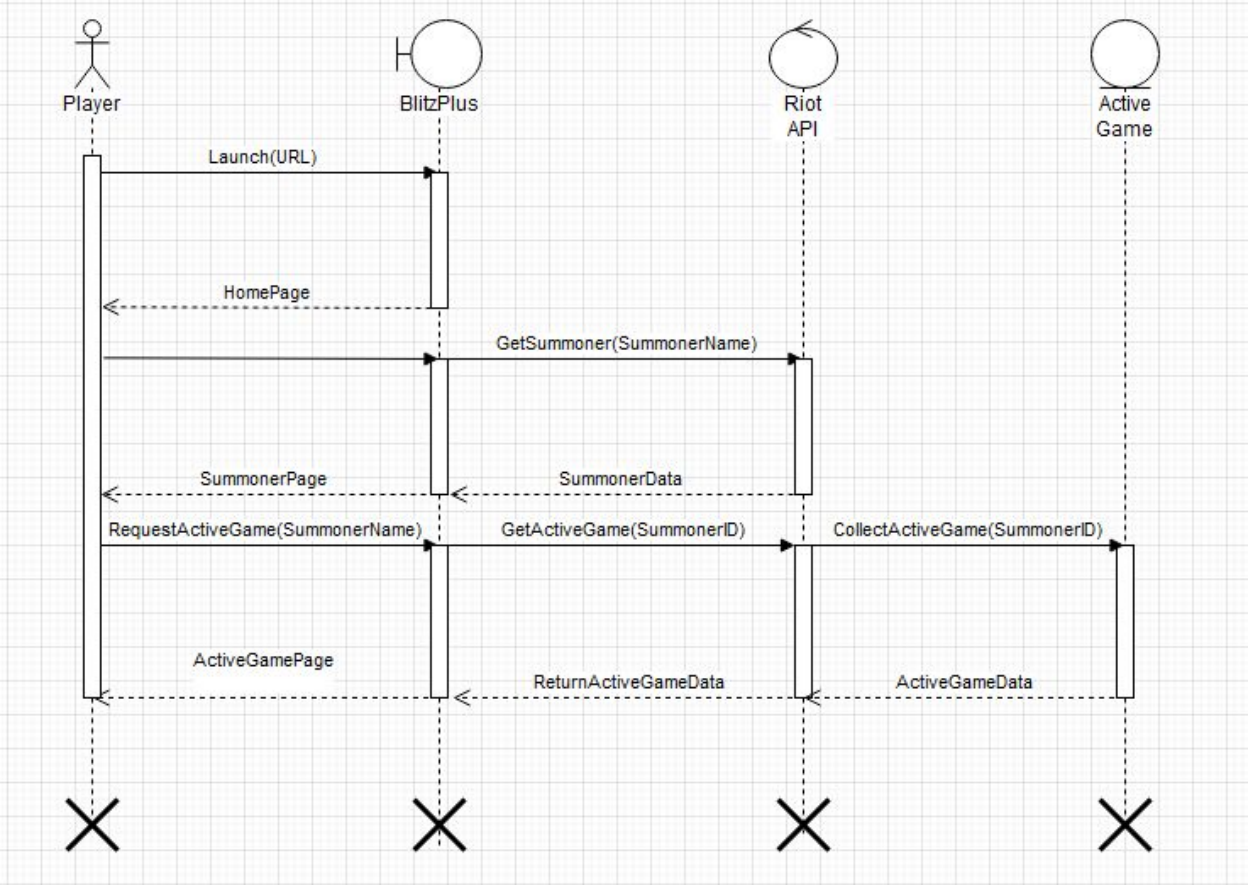
****

Feature: Players can find out the statistics of all the players in my current game. (7) (Edwin)

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



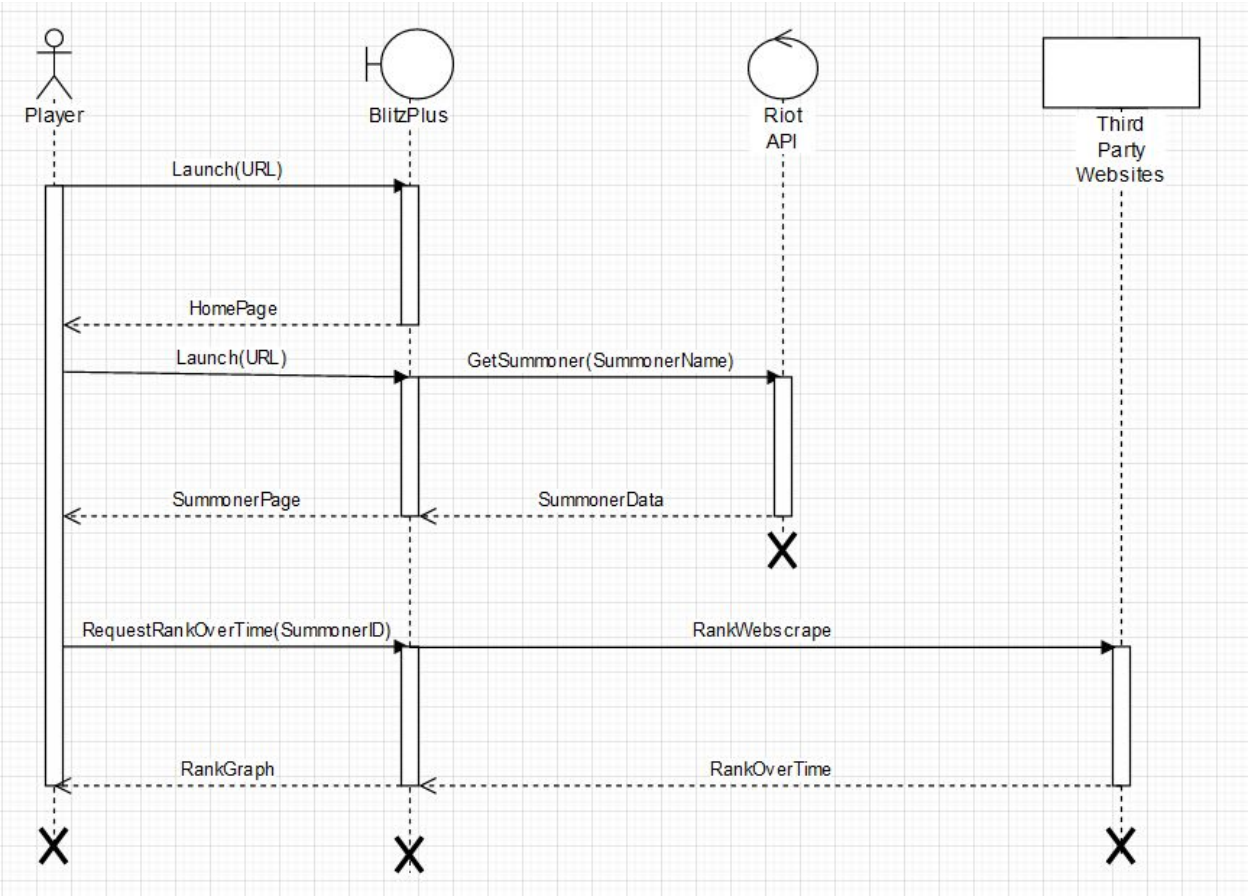
****

Feature: Players can see a player’s performance overtime. (8) (Edwin)

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



2. Then include one sequence or interaction diagram for each use case. You can use UML sequence diagram definition: http://en.wikipedia.org/wiki/Sequence_diagram. Each box in a sequence diagram should correspond to a component in your architecture.