Mobile Application Development Coursework Report

Rory Harrison
40296172@napier.ac.uk
Edinburgh Napier University - Mobile Application Development (SET08114)

Abstract

The aim of this coursework was to design, implement and evaluate a prototype for a mobile application for the Android platform using the Android SDK. We were given relatively free reign on our choice of application, I chose to focus my application LoLAnalytics around statistics from the game League of Legends.

Keywords – Mobile Application Development, Android, Java, api, json, Edinburgh Npaier University, Napier, Rory Harrison, LoLAnalytics

1 Introduction

LoLAnalytics is a statistical analysis android mobile application with it's key focus being on the ever changing meta of League of Legends. The target audience is experienced players who want stats presented in a comprehensive way that aids interpretation and understanding. The app retrieves live statistics such as win rates, pick rates, ban rates, and I have implemented creative design and graphics to display the stats. I initially wanted to create LoLAnalytics because of inspiration from a website called http://op.gg. Op.gg presents a high volume of detailed statistics and tools for players to know as much about the game as possible in order to get an edge over their opponents. However op.gg is not available as a mobile app and despite there being some League of Legends stats apps already on Google Play they have nowhere near the amount of detail and content as op.gg. The goal of the app is to provide players information to help them make decisions about the game, by letting them see which characters are performing well and which ones aren't. LoLAnalytics is currently comprised of a grid of all current characters, users can select any specific character to pull up an activity with statistics and information about them.

2 Software Design

2.1 Activity Diagram

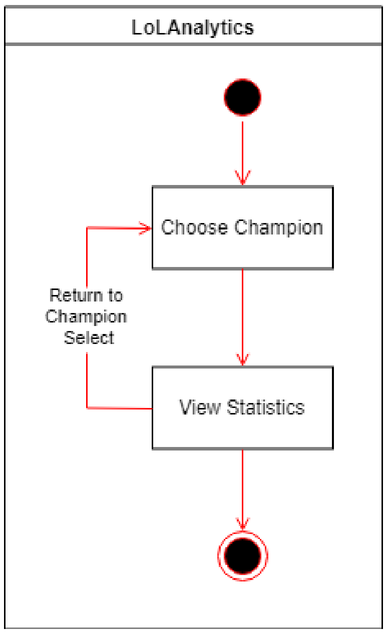


Figure 1: Activity Diagram

I decided to keep the user behaviour flow of my app simple, highlighting the statistics and not confusing users with complicated menus. This is important because often the user may be in a position where they are queuing for or have just joined a match lobby(a time limited situation) and quickly want to look up stats about a character and see how their current standings are before deciding to play as them.

2.2 Class Diagram

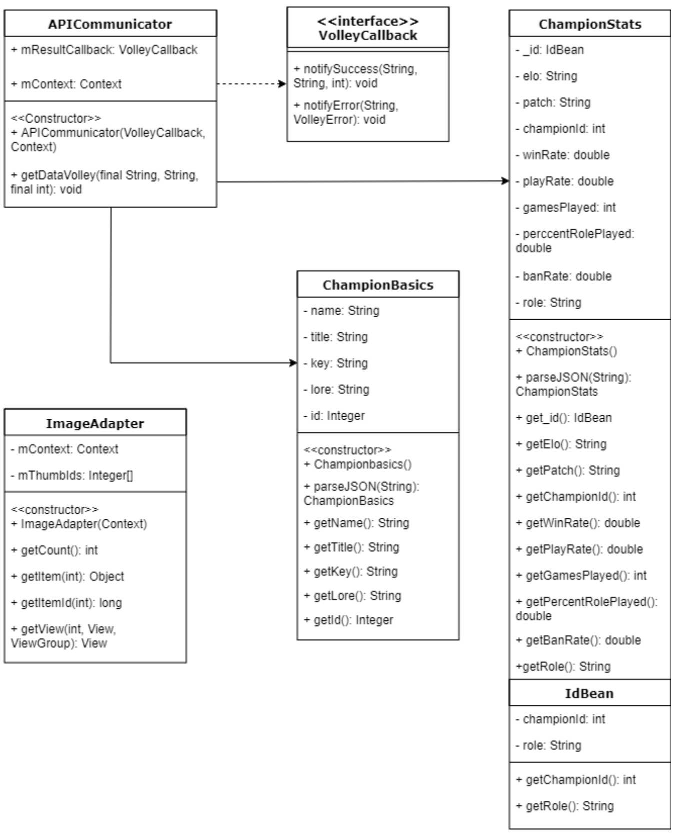


Figure 2: Class Diagram

<u>**Web Classes**</u>

APICommunicator The APICommunicator class is where all the API calls are made and serialised. I use two libraries to help with the requests the first being Android Volley. Volley takes a URL pairs it with a request type and deals with the call. I had originally coded a functional async task that handled the http get requests and was called directly from the championPage activity however this got very messy when carrying out multiple calls in a timely fashion so I decided to use Volley to manage the calls. The second library I used is called google Gson and I use it to serialise the raw JSON string responses into custom java objects, this meant that once I had the calls setup and the responses being serialised into objects using the data was much more convenient.

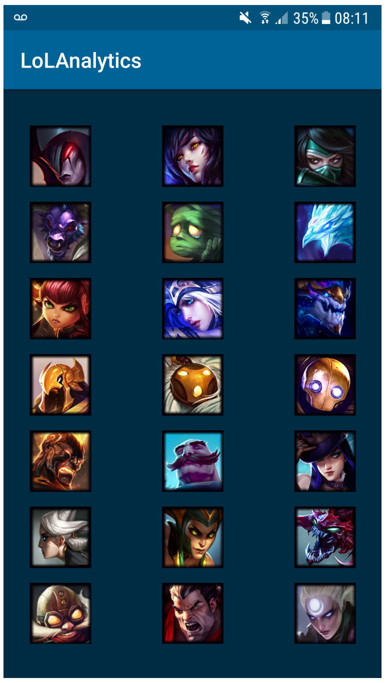
VolleyCallBack This is an interface I created so I could access the responses inside the championPage activity making displaying the statistics easier.

ChampionBasics and ChampionStats These are custom objects for holding the two different API call responses, they have variables that map directly to fields in the JSON responses and methods to retrieve the values. Gson manages the conversions between Json and the these objects.

ImageAdapter This class is used to support the Grid Layout that displays all of the champion icons. It creates new imageviews for each champion icon resource id in an array the adapter holds.

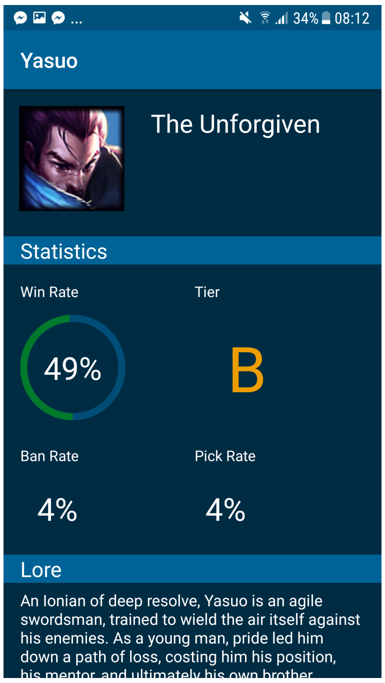
3 Implementation

3.1 Champions Menu



The champion menu where users can choose which champion to see stats about

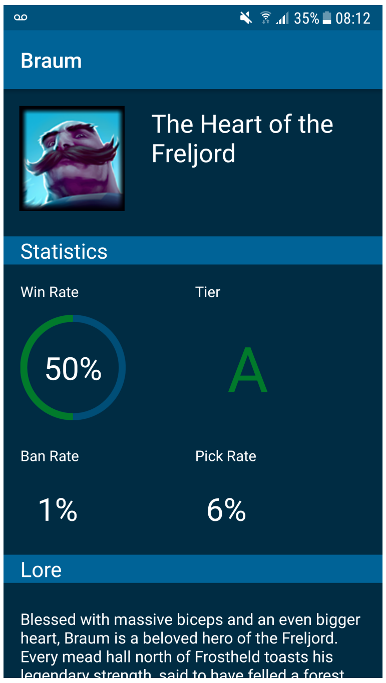
3.2 Yasuo Stats



An Ionian of deep resolve, Yasuo is an agile swordsman, trained to wield the air itself against his enemies. As a young man, pride led him down a path of loss, costing him his position, his mentor and ultimately his own brother

Example: Stats for the champion Yasuo

3.3 Braum Statistics


Braum

The Heart of the Freljord

Statistics

Win Rate 50%

Tier A

Ban Rate 1%

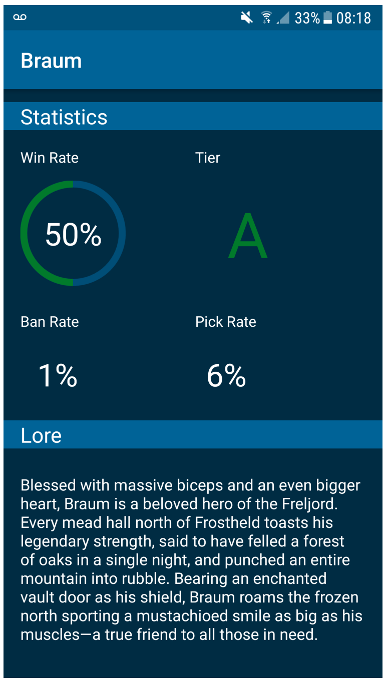
Pick Rate 6%

Lore

Blessed with massive biceps and an even bigger heart, Braum is a beloved hero of the Freljord. Every mead hall north of Frostheld toasts his legendary strength, said to have felled a forest

Example: Stats for the champion Braum

3.4 braum Lore


Braum

Statistics

Win Rate 50%

Tier A

Ban Rate 1%

Pick Rate 6%

Lore

Blessed with massive biceps and an even bigger heart, Braum is a beloved hero of the Freljord. Every mead hall north of Frostheld toasts his legendary strength, said to have felled a forest of oaks in a single night, and punched an entire mountain into rubble. Bearing an enchanted vault door as his shield, Braum roams the frozen north sporting a mustachioed smile as big as his muscles—a true friend to all those in need.

Example: Lore for the champion braum

3.5 Icon

The icon for the app

4 Evaluation

Comparing my app to the original idea Although my app may not have all of the functionality that I want it to eventually have I think that I have made good progress creating a good back-end for managing various API requests and have managed to implement the core statistics that players desire. However it does still have a lot of functionality that could be added to further enhance the app.

Comparing to http://op.gg op.gg does provide a much more extensive set of statistics as my app is only in the prototype stages however I think my app has one advantage of being very straight to the point. It's easy to find the most important stats and not get lost in irrelevant data that could end up.

Possible Improvements I could improve the app by making calls to riot games' api more robust, sometimes their servers get overloaded or go down and I could further implement measures to work around these issues but these events are quite rare. I also plan to go on to add more functionality to the app, adding in a new menu screen(alongside the current one) that is split up into 5 tabs each one showing a list of champions in order of descending win rate where each tab only has champions of one of the 5 roles and having these lists also link to the specific champion pages. I would like to add more legacy data from previous patches of the game to show if a champion's stats are trending upwards, downwards or just for users to see how much certain changes to the game affected each champion.

5 Resources

APIs: developer.riotgames.com/, api.champion.gg/

Images: Champion Icons - League of Legends, Riot Games

Libraries: google gson:2.8.0, android Volley:1.0.0