

League Champion Pool

Written collectively by:
William Hall, Louis Magand, Yu-Ching Ho, David Byron
University of the West Scotland



Table of Contents

Summary	02

Introduction	02

Project Plan and Approach	03

Design/Background	05

Development of Prototype	06

Code Development	08

User Guide	08

Implementation & Evaluation	09

Conclusion	10

Appendix	11

Summary

This report will breakdown our approach to developing an application based on the MOBA (Multiplayer Online Battle Arena) game: League of Legends. There was a multitude of things that had to be taken into consideration while creating this application. This included our project planning, approach, design/background, implementation and evaluation. So throughout this report these subjects and more will be deconstructed and explored to elaborate on our attempt at creating an application.

Introduction

This report was written as part of the required coursework for the Computing Project module. However it is also very valuable in allowing the group to keep track of almost all the important pieces of information with little effort needed to keep it all together and to prevent any loss of data. The objective of this report is simply to keep track of the progress made throughout the project's timeline, this includes showing some of the interface designs that were considered before the final design was agreed on. It also includes explaining the creation of the app, the approach taken in making the app and the feedback received from the internal testing performed as well as the small external testing that took place.

The purpose and use of the app we created is to be a kind of personalised tracker for the MOBA game known as League of Legends. This app allows the user to track all of the lanes and roles that they enjoy playing each individual champion, so at a glance the user can determine who they want to play each game without the need to minimize their game window, which is not always possible depending on the system that is being used to play the game. In addition to this the app also tracks the "standard" roles and lanes each champion typically plays in, meaning even if the player is unfamiliar with certain champions they can have a basic knowledge of what type of damage they do and the role they are most likely to fill within the game.

The method of investigation that was used:

There are a variety of resources available such as the multitude of websites available, android studio for coding the app itself as well as first hand knowledge that can be obtained from the numerous people that each group member knows who plays the game. Following on from this the sources we used for this project was the [League of Legends Wiki](#), the [Android Studio website](#) for documentation and the knowledge the group had on certain subjects.

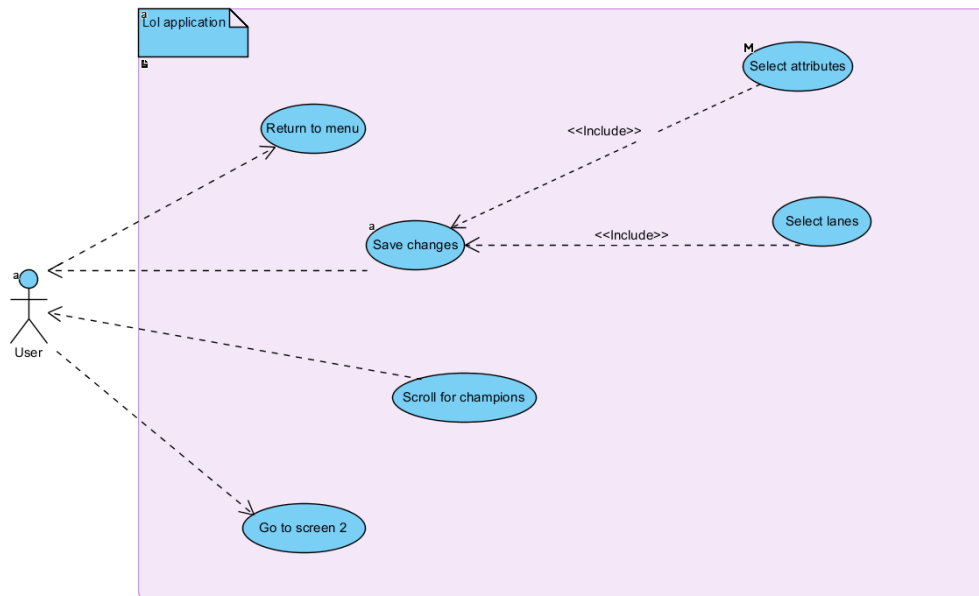
However, there are always limitations. The limitations within this project mostly comes down to group members completing their given work on time and being able to make any group meetings. The other limitations we face is the ability to use the desired software for the app, as well as the physical limitations of the software itself.

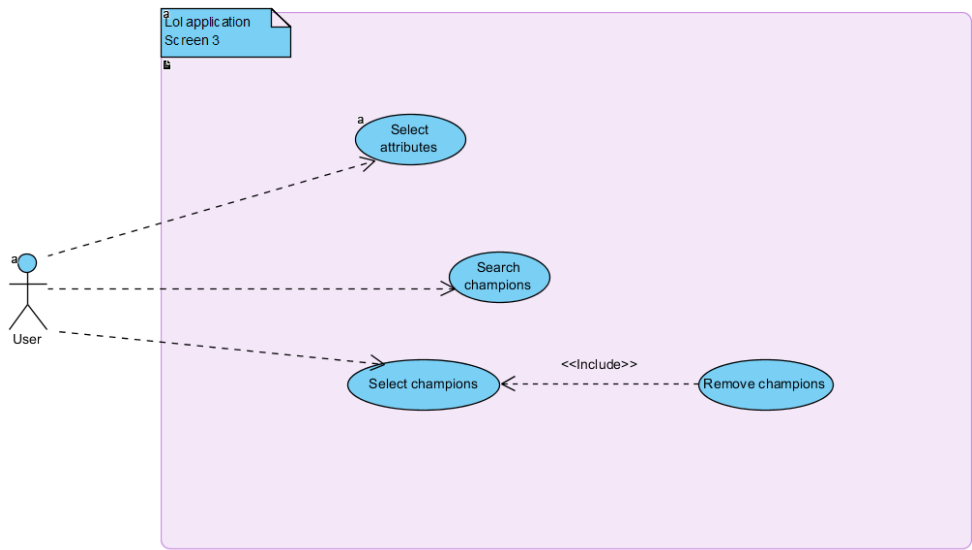
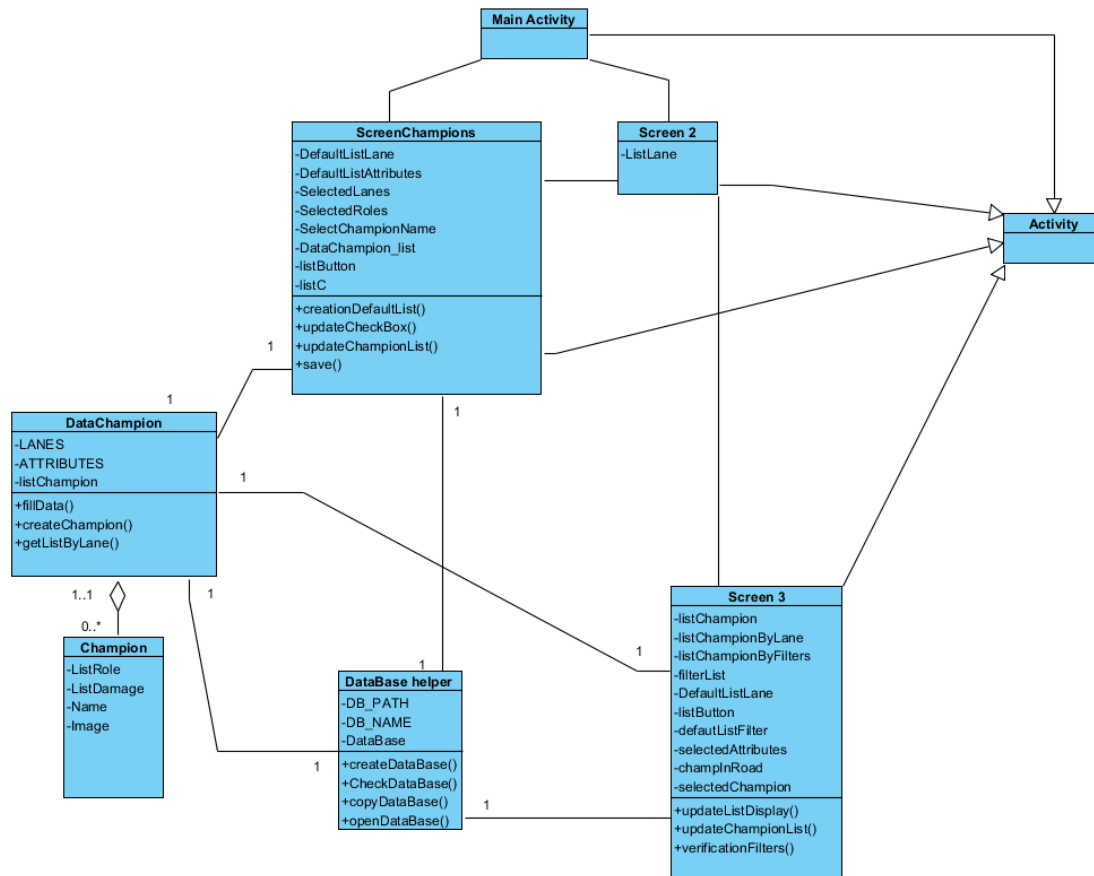
Project Plan and Approach

The approach we decided was best for completing this project was a simple one. The group started by creating a simple plan for the objective of the application we wanted to create, with this plan in place we developed it further to include several milestones for the project, these included the design of the app, the planning, charts, diagrams and documentation, the database construction, the programming linking the database to the GUI. Any constraints and limitations that could arise were then noted and revised to see how likely there were to affect the project. After this was completed a rough timescale was created, using PERT and GANT charts, to give everyone an idea about what had to be done and when. This was primarily to keep all group members busy and prevent someone skimping on any work.

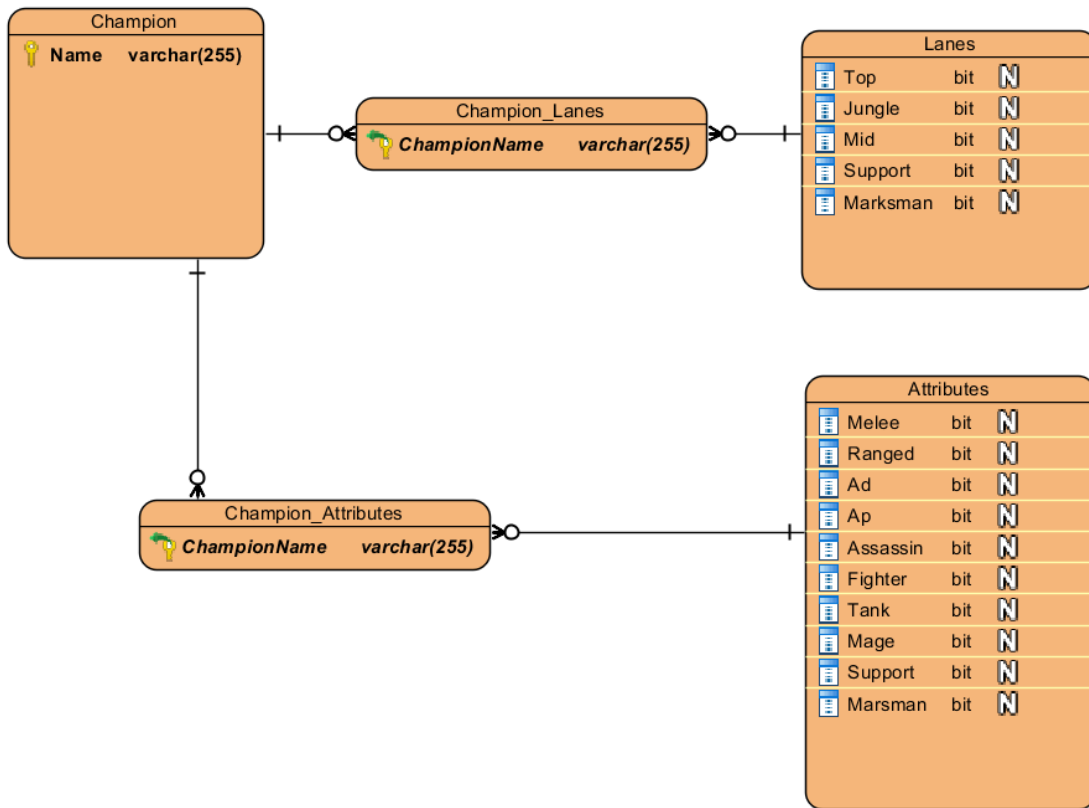
With these aspects in place, we moved on to assessing the resources that we would require for the project as well as the risks that could hinder our plans. In addition to this the evaluation methods that would be best suited to our goal, with this in mind we came to an agreement on primarily internal testing with some external testing performed once the application was properly functioning.

Here are some use-case diagrams (with the class diagram) that we used for the project plan:





For the database, we made a diagram in order to help us to build the SQL database:



Design/Background

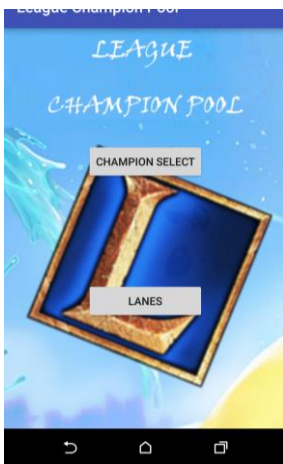
With the project plan complete, we moved on to the preparation for developing the application. For this we began by creating a use-case diagram - which shows any kind of interaction the user has with the interface - and using the base idea we had for the app to determine roughly how we wanted the app to work. Once this had been completed, a wireframe was implemented as a basic construct of the layout we thought would best fit the application. With these completed, before any coding was attempted, we used a sequence diagram - which shows how processes operate with one another and in what order. It is like a use-case diagram but more in-depth, and a class diagram and a function list was also needed. The class diagram is very important for the programmer because it describes the structure of a system with all the classes that is used within the program.

Now that the basis for almost everything that the group wanted to implement in the app, the actual code and GUI (Graphical User Interface) for the application could start being developed. The GUI was started first as it was a necessary component for the remainder of the coding to be based around, because if it wasn't then it would be impossible to test the interactions, as the code can't affect what doesn't exist. So once all of GUI was in place, the initial version main body of the code could be written. Then it simply became a task of updating the GUI and the code for any changes or updates that were implemented as the development progressed.

With the initial version of the code written there is now a basis for the database that will store all the information required for the application can be created. The database was an integral part of ensuring our app worked in the way we desired, we decided to create this database on SQL lite which was capable of implementing the database the way we needed it with relative ease. With the database functioning the code for the application could be expanded and any initial code improved so the app can interact correctly with the new database that had been implemented.

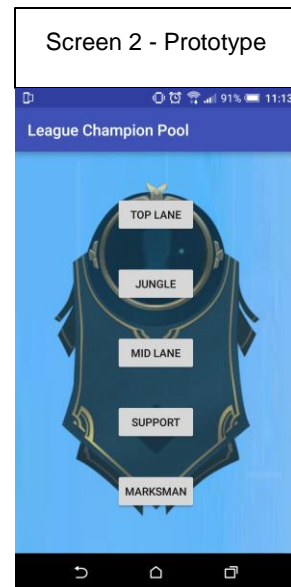
Development of Prototype

With the code, the GUI and the database all in production and near a working state we could actually have the beginnings of our app. One of the initial prototypes of the app was a somewhat simplistic design that we aimed to keep easily accessible.



Screen 1 - Prototype

The first screen of the app (seen on the left) acted as a title screen for the application, it allows a new user to edit the champion list to start their own personalised index of the champions they enjoy playing and in what role or for returning users, to add new champions to their existing lists. The other option on this screen is for lanes, which is for users who have chosen where they like playing the specific champions they own to filter through all of their champions, so that they can inspect which champions they play in each desired lane.



Following on from this, the second screen (seen on the right) is reached from the lanes button from screen one and acts as a kind of middle-man between the user and their personalised index of champions. This screen is there for the user to be able to select which lane of champions they want to browse without the need for dropdown boxes that can be easily missed at a glance.



Screen 3 - Prototype

Once the user has selected the lane they want, it takes them to the next screen (screen 3 - prototype). This screen is the area that keeps track of the various lanes that the user has set for the champions they want so that they can easily scroll through the champions they enjoy playing for their desired lane before a match starts. Being able to do this stops any need to minimize the game window in order to check things before the game begins.

Returning to the first screen, by using the champion list button the user is taken to the final screen used by the app (seen on the right). This screen is the main screen the user will use for setting up their personal lists of the champions they enjoy playing. As this screen contains all of the champions as well as their 'standard' attributes, which allows the user to have an idea on what a champion is supposed to do even if

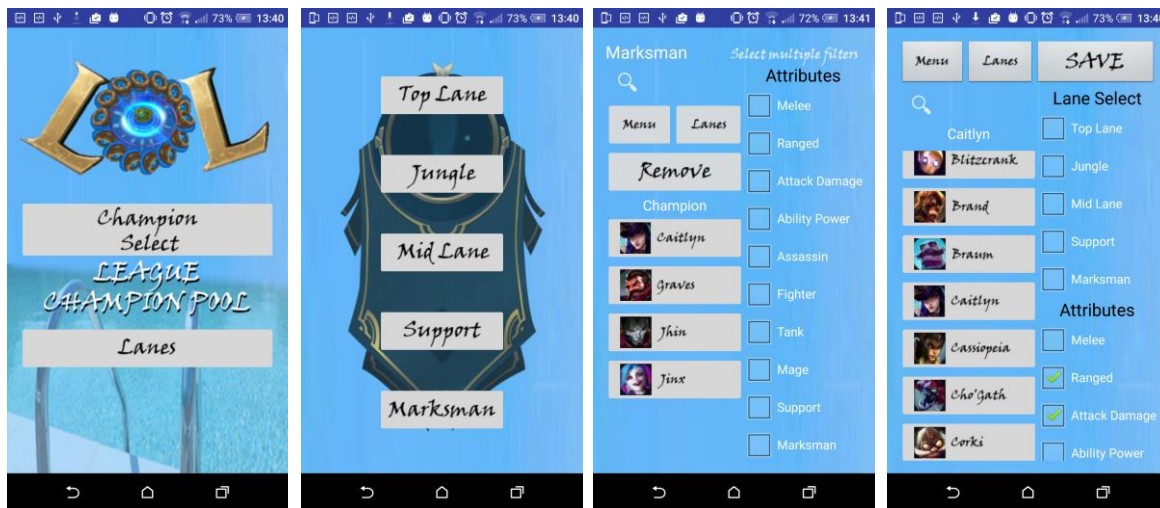
they haven't played them before. This is useful to both new players and experienced players as it means the user needs to remember less information on over 100 different champions that are in the game, some of which are played considerably less than others.

Screen 4 - Prototype



After continuing to improve and modify our code and GUI we eventually ended up with a better looking aesthetic compared to our original prototype.

Our finalised design looks like this:



Code Development

For all of the project, we didn't take anything from the Internet except for help regarding the DatabaseHelper class, from this [website](#). This code had everything that we needed for the application - the creation of a database if it doesn't exist, but if it does exist, it will be copied but then when we made the database, we could open it at any time - so we took this code and modified it for our application.

The most complicated and sizable file is the DataChampion class, this class has a DatabaseHelper attribute, so it will deal with lot of functions with SQL query. The first important function is fillData(). It fills an array list of Champion with all the champions (their attributes and their lanes) from the database. We also have createChampion() and updateChampion() which update his array list of lanes and array list of attributes. And finally, the function getListBylane(), it change a array list of champions in a array list of champions with a specified lane.

The other two important classes are ScreenChampion and ScreenFilters. In both files, we made a little trick for the search view. In the onCreate(), if the user wrote something in the search view, the variable "listC" is created which is another champion list with all the champions with the searched text in their names, and the screen updates with the new list. For the other functions, they do what their names said, for example updateCheckBox(), it updates checkboxes.

User Guide

When you launch the application, you arrive at the main screen. Here you can select two options, the first is "Champion Select" and the second is "Lanes".

In "Champion Select", you can modify individual champion's attributes and their lanes, return to the menu, go to the "Lanes", search for a specific champion or save your choices.

In "Lanes", you can select a lane which move you to a different screen. Like "Champion Select", you can go to the menu, return to "Lanes" to select a different lane, search for a champion, remove a champion, filter out champions with different attributes, and once selecting a champion a textbox will display the name of the selected champion.

How to modify the attributes or the lanes of a specific champion in "Champion Select"

You click on the champion you want to modify and the name of the champion will show up in a textbox to avoid selection errors. The checkboxes on the right-hand side of the screen will update with the default or saved attributes. For example you can click on the top lane checkbox if you want the specific champion in top lane. After checked the box, you have to click on the "Save" button to save the changes in the database and the same thing applies to the attributes section. If you forget to save, all changes when leaving the "Champion Select" page will be lost.

How to see all my champions for a specific lane in “Lanes”

When you are in “Lanes”, you can select a lane. When you have selected, the program searches in the database all the champions with the lane that you have selected. Then on the left-hand side of the screen, you can see a list with all your champions with the specified lane.

How to filter my champions with specific attributes in “Lanes”

In the selected lane page, you can filter your champions with different attributes. For this to happen, you check which attributes you want and the champion’s list will update. For example, if you want to see all your champions that you have with the melee attribute, click on the checkbox “melee” at the right of the screen and it will update with all your champions with the melee attributes.

How to remove a champion for a specific lane in “Lanes”?

In the ‘Lanes’ option, you have the possibility to remove a champion from the selected lane. If you see a champion that you don’t want to appear in whichever lane, you can click on the champion (the name will appear in the text box just below the remove button) and delete it from this lane by clicking on the “remove” button.

Implementation & Evaluation

Having completed the prototype of the application and the initial testing phase, we began to think about how we could test the app on a wider scale and potentially implement it into the android market, the Google Play Store. It is a fairly simple process to upload an application to any android phone, and so we began to circulate the app among family, friends and fellow students to get some feedback from fresh eyes.

We began with a pre-release evaluation of the user interface design. As an Android app, it was of vital importance that our software was both functional and intuitive. Our clean and simple design allowed our users to easily come to grips with how each screen functioned and navigate around the app successfully.

The beta testers feedback on the usability of the application was mostly favourable. It was pleasant to look at, streamlined, efficient and easy to learn. The scope of the app was simple, but League players noted that it was a boon for them to have our app at hand as they played the game or even as something to use as a planning tool.

The application will not be released for public consumption until later, so the full scope of feedback will not be available for analysis in this report. That will be the true acid test, but signs are good that we will see further favourable reviews.

We managed to avoid any potential legal issues by getting in touch with Riot Games (developer of the game) and finding out what names and images were copyrighted material. We found out that the League of Legends logo itself was a protected trademark but the images and names of the champions in the game were not. We have since created our own custom logo and verified with Riot that our application was within the boundaries of their copyright.

There is only one problem in using this app - that it is designed solely for League of Legends players. Since we are marketing it towards League players only however, it will not be an issue.

In the summary in the Google Play Store, it will explain how the attributes and lanes can be edited. This is what we plan to put as our summary, in addition to some eye-catching screenshots:

“This app allows you to put each champion you play into your favourite lanes and roles!

Every champion is sorted by its default attributes but the attributes can be edited to suit how you play!

Remember to save each champion individually!”

Conclusion

As a group, we feel we have learned a lot from our work on this module and hit an important first milestone: our first software release.

This not only will be an excellent thing to add to our CVs and portfolios, it also provided valuable insights into the software development process and the way projects are planned and broken down. We will be able to use this experience to expand our knowledge of effective group work and project management. Overall this has been an interesting experience working in a somewhat small group and actually producing a working application that will be published on the android store, so hopefully this should give us some knowledge on how an actual development process within the industry is conducted.

Appendix

There is a javadoc attached to the report. You can find it in javadoc.zip. This expands on every section of code used within the application, which furthers what was spoken about in this document.

Wireframes:

The very first time users use this app, a pop-up box will appear with:

"Do you want the champions to be sorted automatically to pre-set lanes?"

Yes - will be set automatically

No - user will manually set the lanes

Main screen. First page that users will see they open the app.

TITLE - Name of our app

SCREEN 1 - Screen 1. Users can put champions into their lanes and can edit the attributes of each champion.

SCREEN 2 - Screen 2. Users can select the "Lane" they want to play

Animations? - Barely anything. Box changing colour when selected.
Moving between screens.
Removing a champion.

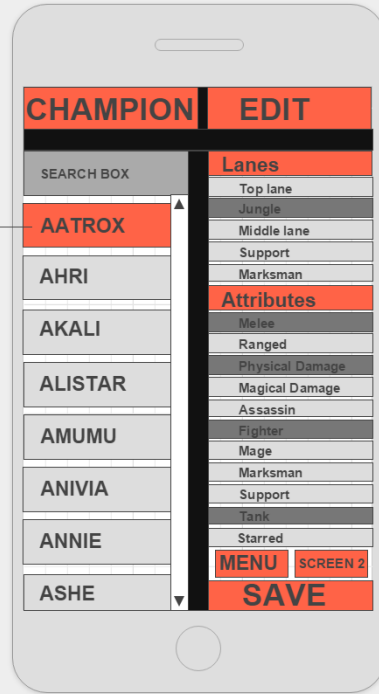
Art? -
Nope. We're just using the faces of champions so we can go to the champion wallpaper and get the face off of that.

Sound? -
Just sound effects. No need for background music.

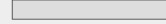
Copyright? -
Need to check.
Specially from lolcounter if we are adding the counter section.

SCREEN 1

A champion's box will appear as below:

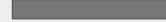


Colour of unselected



(Say if user ticked "Yes".
Pre-set lanes and attributes that can be edited)

Colour of selected



If "Mage" is selected, "Magical Damage" is
always auto-selected, but not vice-versa.

If "Marksman" is selected, "Physical Damage"
is always auto-selected, but not vice-versa.

SCREEN 2

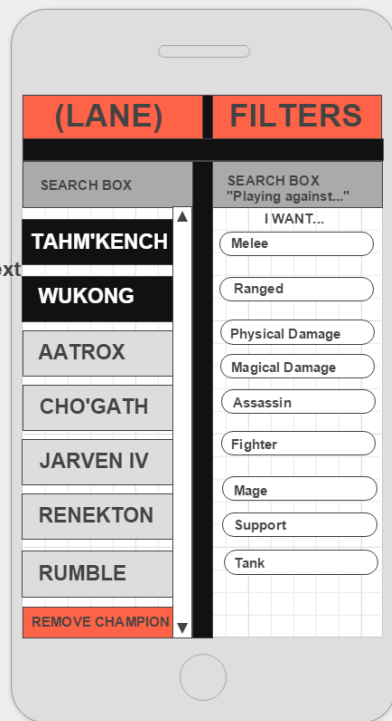


SCREEN 3

Imagine, the starred champions are in a bright yellow box, or has a star next to their name.

If a champion is "starred", the champion will show up at the top of the list as the user's "favourite champion(s)"

If there are multiple "starred" champions, they will arrange alphabetically.



Your selected champions will appear on the left-hand-side.

They will organise "starred" first, then alphabetically.

BONUS!!!!!!!!!!!!

You can narrow down which champion you want to play selecting which filter you want.

If you use the search box "Playing against" and type in a specific champion. Your champions on the left-hand-side will start narrowing down on which ones are the best counter-champions.

Once the counter-champions show up, you can select one, you can narrow the champions down even more by selecting any filters you want.

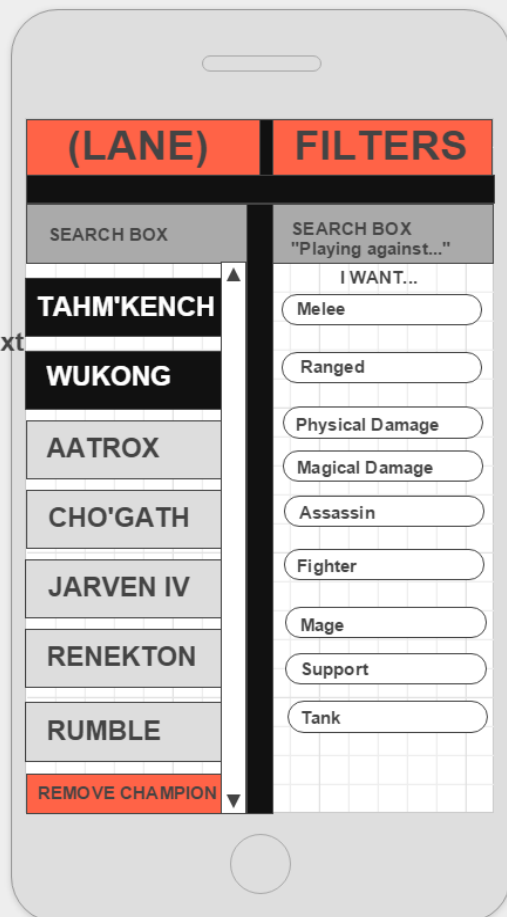
<-- Obviously, this will be spaced out better

SCREEN 3

Imagine, the starred champions are in a bright yellow box, or has a star next to their name.

If a champion is "starred", the champion will show up at the top of the list as the user's "favourite champion(s)"

If there are multiple "starred" champions, they will arrange alphabetically.



Your selected champions will appear on the left-hand-side.

They will organise "starred" first, then alphabetically.

BONUS!!!!!!!!!!!!

You can narrow down which champion you want to play selecting which filter you want.

If you use the search box "Playing against" and type in a specific champion. Your champions on the left-hand-side will start narrowing down on which ones are the best counter-champions.

Once the counter-champions show up, you can select one, you can narrow the champions down even more by selecting any filters you want.

<-- Obviously, this will be spaced out better

Sequence Diagram:

