

League of Legends application

Project Plan

Students: David Byron
 William Hall
 Yu-Ching Ho
 Louis Magand

Contents

1. Overview
2. Objectives
3. General Approach
4. Contractual Aspects
5. Scheduling
6. Resource Requirements
7. Personnel
8. Risk Management
9. Evaluation Methods
10. Appendix

1. Overview

Our team is planning to make an application which will benefit League of Legend players. It is going to be a simple offline app that shows which champions you play in each lane.

This app is very personal to the user since the user can select which attributes and lane(s) he wants for every champion he plays.

Our set milestones are:

- The design of the app.
- The planning, charts, diagrams and documentation.
- The database construction.
- The programming linking the database to the GUI.

Any constraints we may have:

We may have some copyright issues with getting data from a website with user-generated statistics - lolcounter.com

2. Objectives

Our objective for this app is show the champion pool of each person.

There are a lot of different champions and for some people, they play only a few champions for each lane and master them completely.

While for other people, they can play a wide range of champions - a jack of all trades - and so it is hard to keep track which champions they can play.

Or there are people who are in-between: they have a preferred lane they wish to play in, can play a wide range of champions in that lane; however in other lanes, can only play a small number of champions.

The app will have four screens - each easy to navigate and self-explanatory and behind the scenes, an SQL database for saving all the information.

The user can select a champion to edit its attributes and decide which lane(s) the champion will be played in. The user can then go to another screen and see for each lane, all the champions that the user plays for each lane.

That is our main objective. If we have time and desire it, we will add a champion-counter option.

When the user selects which lane they want to play in, the user will see all the champions they play in that lane. If the user is playing against a particular champion, the user can select that champion in the search box, and all the counters for that champion will remain while the rest of the champions are filtered out; then they can refine the search of which champion they want to play by selecting desired attributes.

Our outcomes for this project are:

Being able to select a champion - edit its attributes and select which lane(s) the champion will go on, and both save successfully.

If the user wishes, default values for each champion will be automatically organised.

If we add in the champion-counter option - when the user selects a counter champion, which champions they selected for that lane, will be filtered out for the best counter.

3. General Approach

Technological and managerial approaches to the work:

We tried to divide the work fairly among our group members to try and ensure that everyone has equal workload and attempting to stop anyone from failing to pull their weight. We will go into greater detail about all the tasks in a later section.

Everyone was involved in creating the Project Plan document. This was done so that everyone knew the work that needs to be done as well what work they will be responsible for completing.

Preparation:

Yu-Ching Ho made wireframe sample screens showing the UI design of our app (available in Appendix).

Louis Magand made the Gantt Chart and the PERT chart with Microsoft Project.

David Byron and William Hall created the rough template of the project plan and everyone contributed and collaborated to write it.

Conception:

William and David will make a **Use Case** diagram for the database and the user. It's a diagram that shows any kind of interaction that the user has with the interface.

William and Yu-Ching will create a **Sequence** diagram. It shows how processes operate with one another and in what order. It's like a use case diagram but more in depth.

Everybody will make the **class** diagram and the **function list**. The class diagram is a very important diagram for the programmer because it describes the structure of a system with all the classes that is used within the program.

Louis and Yu-Ching will help in all the diagram conception and will make a skeleton of the database.

Programming Stage 1 (View):

We said “View” because in this section we will program the look of the program - (ie - the interface and GUI.)

Yu-Ching will construct the GUI application with Android Studio.

Louis and William will create the initial design of interface and begin coding it.

The team will test the interface with some class tests. The class tests will replace the database for the test.

Programming Stage 2 (Model):

We will then move onto programming the main model of the program.

Yu-Ching will construct the database using the skeleton previously created.

Louis will link the interface to the database and connect it up with the relevant code.

The team will work together to code the fourth screen and finish by linking it with the database.

Testing and Marketing:

This will be the final phase of the project.

Everyone will work together to first make the **alpha** and then the **beta** test. The testing process will be outlined in more detail later in this document.

David and Yu-Ching will be behind the advertising, an introductory YouTube video and finally, the upload on the Android PlayStore.

It will be important for group members to be flexible and willing to support each other depending on our workload for other classes. We will endeavour to communicate to make sure we have a plan B should one of our team members have issues with or be unable to complete an assigned piece of work. This document itself is a good example of how we can work well as a team - with different members working on different sections at the same time. We hope to continue this level of collaboration in our other work.

We decided to program with Android Studio because Yu-Ching, Louis and David have some courses in this. Louis’s previous experience with Java would also be useful here.

4. Contractual Aspects

The application will be standalone but would definitely profit from linking up with some similar applications and websites such as lolcounter.com. This would improve the visibility of the app and help creating a larger userbase for testing.

Similarly, linking up with other free League of Legends apps such as lolnexus can help create a broader experience for the user. User information and market research data could be shared between the creators and a network of like-minded creators could be built.

An agreement with Riot Games (the game's company) themselves could see us use the official Riot Games API which would provide us with a wealth of information straight from the source which could potentially be used in the construction of our database as well as for research into user trends.

We have decided on Android Studio as our development environment. This IDE should provide us the flexibility to create a functional app that will be available to the largest user-base possible.

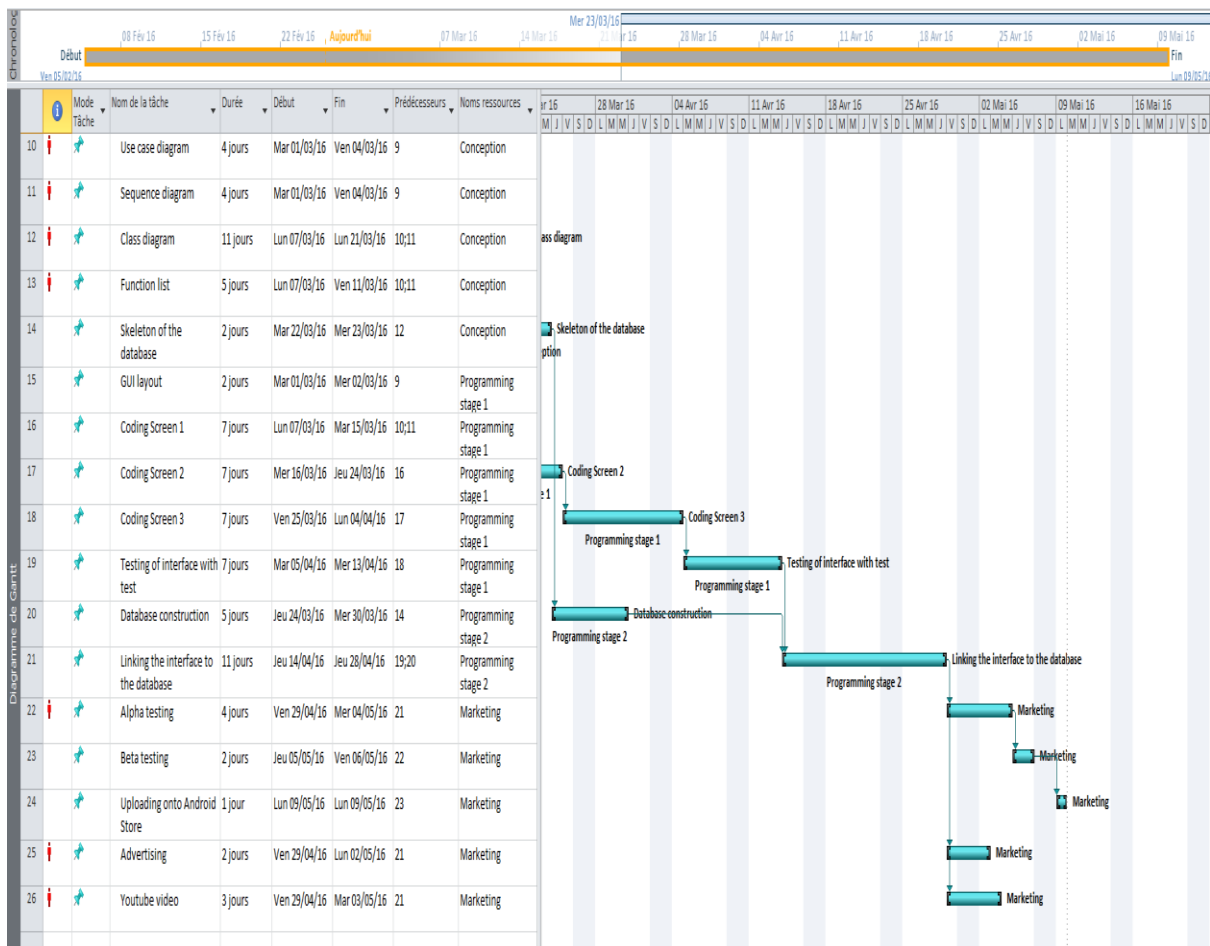
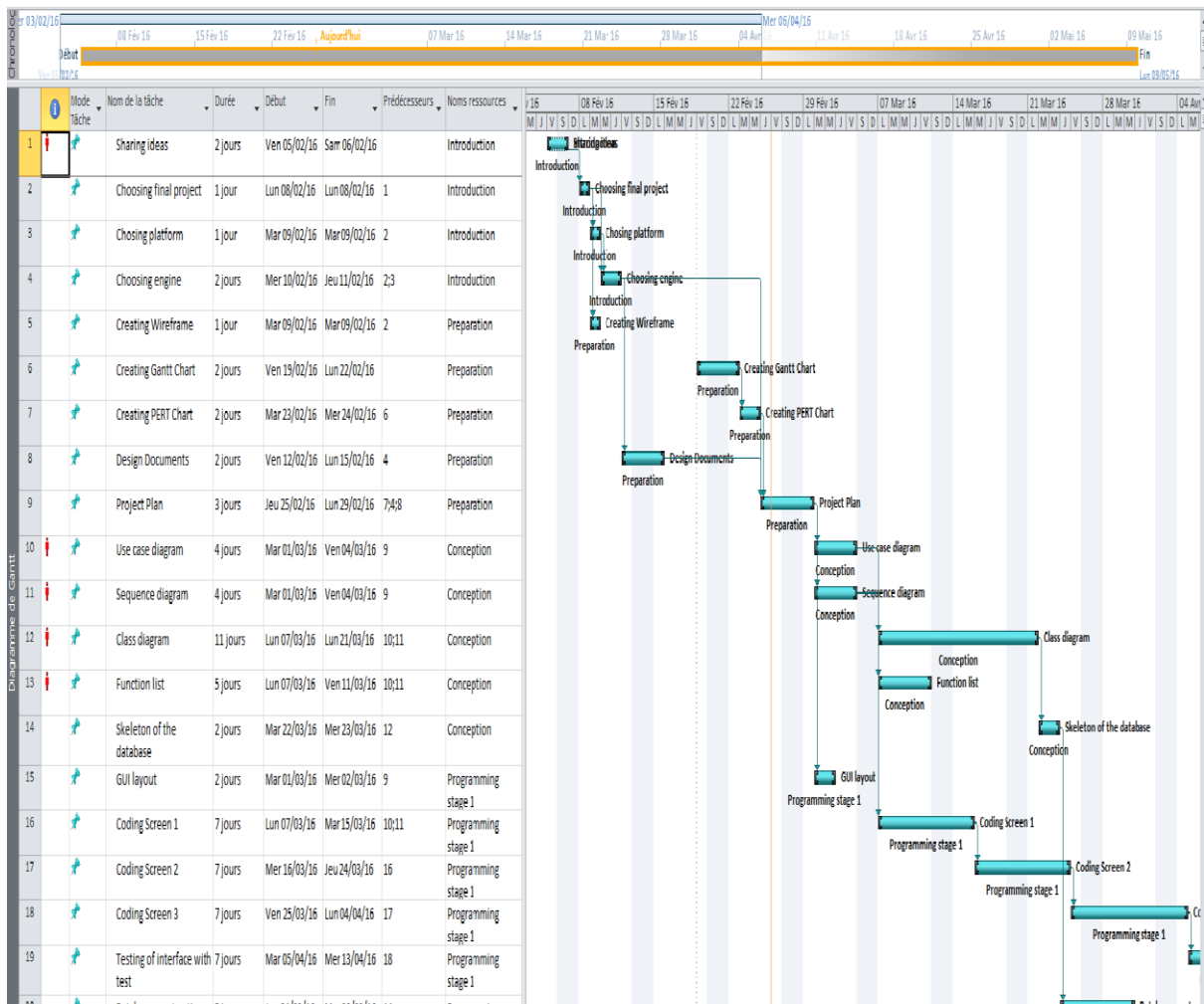
There is also the potential to expand the app to other similar games like **Dota 2** or **Smite**. Similar to this we could try to expand our app to IOS which would help expand the user base. The only potential problem that could interfere with our app would be potential copyright issues.

It would be important to discuss potential copyright/legal issues with our potential clients and partners to make sure we develop a good reputation as we take our first step into the industry.

5. Scheduling

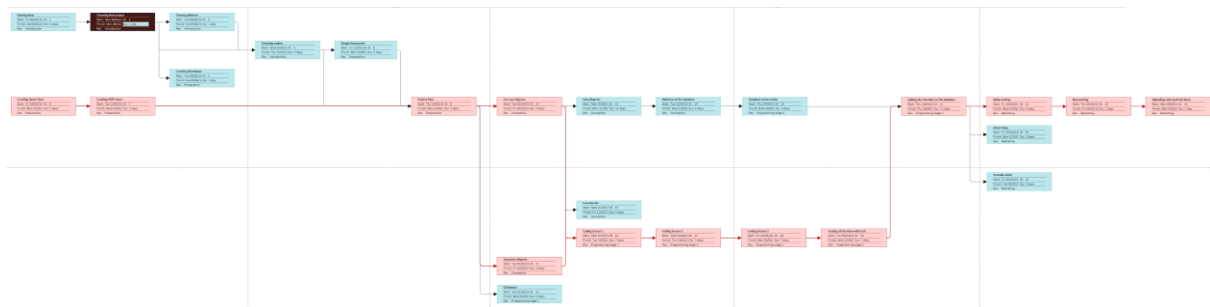
As shown in part 3, the work was divided into six important parts. Using these six parts as a basis, we created GANT and PERT charts (with samples below). The full charts are submitted with this document.

GANT Chart (Detailed chart in submitted folder):



PERT Chart:

The red line represents any delays encountered (Critical path). You can find the picture in the folder for ease of movement.



6. Resource Requirements

There are no monetary costs associated with the tasks as the IDE we are using, Android Studio is free to use. It is possible there would be costs sometime down the line in the future if we were to expand the app or enter into some sort of agreement with a client.

The main resource would be the actual man hours required to complete the project. In this case of course, we will be working for free but it could be something important to consider in the future.

7. Personnel

The team is made up of a group of 4 Students from UWS, Paisley.

Louis Magand:

French student. Computing Science, he knows how to code in Java and started Programming Native Application (Android) as same time than Yu-Ching and David.

Has also studied MySQL.

William Hall:

Computer Games Technology student, has knowledge in programming in C++ and Java. Also attempting to learn some android programming to help in programming the app and knows how to complete documentation fairly well.

David Byron:

Computer Games Development student. Has some knowledge in various game creation software such as GameMaker, Unreal and Unity. Can also code simple programs in Java and JavaScript and is attempting to improve his level in those programming languages in his spare time. Comfortable in the creation and maintenance of various kinds of document.

Yu-Ching Ho:

Computer Games Development student. Also has some experience in various game creation software. Strong networker - has useful contacts that should prove useful to our project. Comfortable in some design software (for creation of UI elements, wireframes etc.)

8. Risk Management

The main issue with any complex software project tends to be time. We have a strict deadline in 8 weeks' time and it is important to use that time as efficiently as possible. It will be important for the team to communicate often and discuss progress on their current workload. In the areas where our skills overlap, we will be able to support each other and give a second opinion on a piece of work.

We also have members with specialist skills on the team who may have to tackle important pieces of work alone, and we will be sure to give them a more flexible deadline and support them by covering work in other more general areas.

It is always easy to stretch a vision too far - something that's especially common in software development. The nature of our application means that it has a wide scope for further add-ons and expansions. While there is some functionality we would hope to add, time-allowing, we must be sure to have the core of the app in place before doing so.

Before we begin the main development of the project, we must be certain our chosen development environment (Android Studio) has the tools required to create what we need. We will do some research on the capabilities of the IDE to make sure it can handle the demands of our app.

9. Evaluation Methods

Our project is primarily a software project and so we will be taking an iterative approach at each stage in development to ensure that:

- a) The software is functioning properly
- b) That it is coded and structured in the most efficient way possible.

We will try to follow our project plan as closely as possible so that each project member is always active and has something to work on. UI Design for sub-screens of the app for example will be created while the programming for the main screen is in progress.

As our areas of expertise overlap to some extent, we will be able to have team members perform checks and testing on completed work and make changes where necessary.

If possible, we would also like to send out an early prototype of the app to friends and perhaps interested players on the LoL forums. We would then be able to change some design aspects based on user feedback.

We have incorporated two large testing phases into our project plan: Alpha and Beta.

In the Alpha stage, the overall build of the app is still somewhat flexible and changes can be made based on our own in-house testing. It is here we will make final decisions on the project in preparation for Beta.

By the Beta stage we should have a working prototype off the app and we should be able to supply the build to tester volunteers. While we await feedback, it will be time to clean up the app and deliver the final polished application.

10. Appendix

