2014

Team Eevee

01/01/2014

Integration Project Architecture & Platforms

Table des matières

[I. About Project 2](#_Toc400542793)

[A. Diagram WBS 2](#_Toc400542794)

[B. Environment 2](#_Toc400542795)

[C. Goal 3](#_Toc400542796)

[II. Risks & Solutions 3](#_Toc400542797)

[III. Activities Management 4](#_Toc400542798)

[A. Skills sheets 4](#_Toc400542799)

[B. Gant Chart 5](#_Toc400542800)

[1. Initial Gantt 5](#_Toc400542801)

[2. Final Gantt 5](#_Toc400542802)

[C. Technical choices 6](#_Toc400542803)

[1. Web Supervisor 6](#_Toc400542804)

[2. Application Server 6](#_Toc400542805)

[3. Mobile Client 6](#_Toc400542806)

[4. Database server 6](#_Toc400542807)

[IV. Quality 7](#_Toc400542808)

Dop Eevee

# About Project

## Diagram WBS

## Environment

This project has an academic purpose. The most important part is to make technologies choices across benchmarks for a mobile platform project on which we had to mix golf and tourism in a game.

## Goal

On this project we will see how important are the technologies choices and how they can impacts the length of the different tasks. They can also increase the challenge but it’s a good way to push our limits and discovering new API and/or technologies. There are more and more connected devices so increasing our knowledge about today’s technologies is very important.

# Risks & Solutions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Quite likely | Probable | Very Likely | Workaround |
| Absent Member |  |  |  | Adapt the Gantt and anticipated workload |
| Undervalued Task |  |  |  | Quickly assign a new resource to avoid brook effect |
| Wrong interpretation |  |  |  | Taking time to understand subject and ask questions |
| Material problem |  |  |  | Having multiple copies |
| Bad Technical choices |  |  |  | Evaluate if remaining time allows to switch |

# Activities Management

## Skills sheets

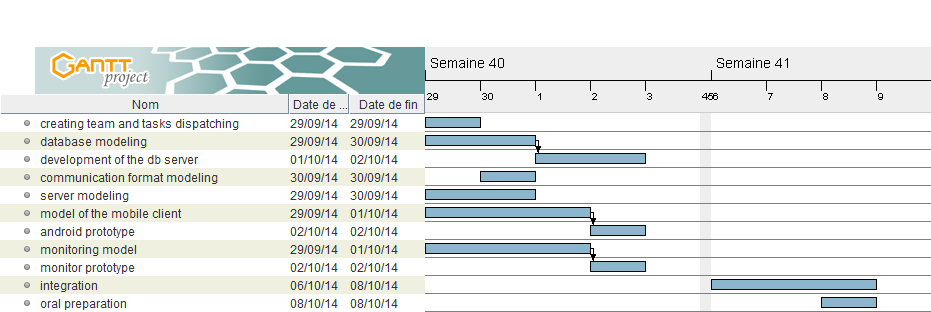
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C++** | **Android** | **Web** | **Data Base** | **Project Management** | **Design and Modeling** | **Communication Http** |
| Dan Castro Lopez |  |  |  |  |  |  |  |
| Mathias Da Costa |  |  |  |  |  |  |  |
| Julien Loève |  |  |  |  |  |  |  |
| Justine Sabbatier |  |  |  |  |  |  |  |
| Thomas Prak |  |  |  |  |  |  |  |
| After Project | | | | | | | |
| Dan Castro Lopez |  |  |  |  |  |  |  |
| Mathias Da Costa |  |  |  |  |  |  |  |
| Julien Loève |  |  |  |  |  |  |  |
| Justine Sabbatier |  |  |  |  |  |  |  |
| Thomas Prak |  |  |  |  |  |  |  |

Legend

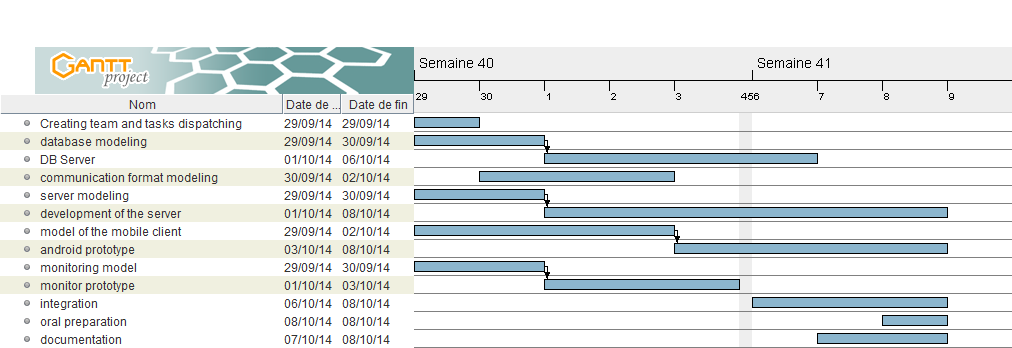
|  |  |  |
| --- | --- | --- |
| Novice | Intermediate | Good |
|  |  |  |

## Gant Chart

### Initial Gantt



### Final Gantt



We divided some task on subtasks to clarify the priorities. For some tasks, we didn’t planned enough time because we didn’t realised that some tasks were harder than expected.

## Technical choices

### Web Supervisor

HTML5, CC3, Jquery has been used with Bootstrap to make interfaces. As said Jquery, we write less and do more in a short lapse of time.

Json was used for following teachers objective’s with Ajax for more effective loading.

### Application Server

The application server needs to communicate with both clients, supervisor and DB server. So it must integrate a server part and a client part.

We chose to use the C++ language and to work with Microsoft Visual Studio. As the instructions asks us to run the application server on a Windows system it appears to be a good integration solution. Using this native language for the required platform gives good performance and require nothing more than the program itself to run.

Moreover, we found a very interesting and open-source Microsoft project named “Casablanca” which is a C++ REST SDK. The sources are available at CodePlex, MSDN or NuGet library. It is designed as a modern asynchronous C++ API made for cloud-based client-server communication. It integrates many features (classes) like HTTP client/server, URI, asynchronous streams, WebSockets client, oAuth. It also facilitate the manipulation of JSON objects with a lot of specific functions which are directly linked to the HTTP client/server classes. As the project require to use JSON in http queries the Casablanca SDK score one more point. Considering the academic part of the project, the application server responsible, with the project manager agreement, chose to spend more time learning how to use this complex but complete SDK which could be used in the future.

/\* Application server technical documentation

* Install Visual Studio 2013
* Open the “Application Server” solution
* Go to NuGet Packages and install the C++ REST SDK (Casablanca) package
* Check and eventually update the DB URI (in Database class)
* Build and run the solution
* The server is working and listening at all hosts on port 34568 \*/

### Mobile Client

### Database server

For the DBMS (Data Base Management System) we chose to use MySQL mostly for the possibility to use the type “Point” like that we can use the DBMS for some calculus like the distance between the golf ball and the nearest POI (Point Of Interest).

To communicate with the database we decided to make a web service in php. This script interpret the Json provided by the Application Server, make the appropriate SQL request and generate a Json given as a reply.

This database was filled with informations from several “office du tourisme”.

# Quality

To evaluate the quality of the application, we can identified standards as:

* Response time
* Ergonomics
* Intuitive use
* Data consummation
* Application size
* Fluidity