François Lemarchand
C00151108

15/04/2012

Project Report

Project Manager App

Supervisor : Paul Barry

Acknowledgements

I would like to thank Paul Barry for supervising the project and also Joseph Kehoe for helping me to get a proper idea of what the product should be.

Table of Content

1-Introduction 4

2-Final product description 5

Web application

Android application

3-Conformance to Specification and Design 9

4-Learning 10

5-Review 11

6-References 12

1-Introduction

Despite the stoppage of my first project in December, I came up with the idea of developing a project manager which would be accessible via web browsers or Android application. [AND] The project is not really original but it was still quite ambitious as I only started in January and this left me only four and a half months to go through the whole development process. I started coding the web application in February and finished it at the beginning of April while the whole Android app was coded during the second week of April.

This document will point out all the problems encountered during the developments and the differences between the design and the final product.

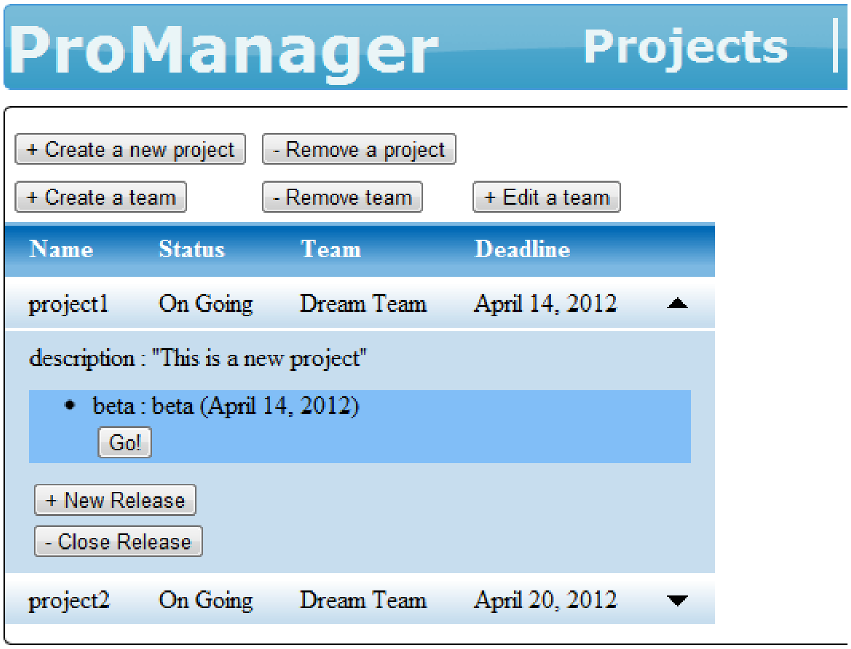
2-Final product description

Web application

First of all, the web application can be accessed by different types of accounts which are administrators, managers and normal users (such as developers). Administrators are able to create new companies, understanding that each company has its own workspace and cannot see members and projects of others companies. Administrators are also able to create new managers and simple users for any company.

Then, we have managers, they can see all the projects of their company and have all the rights on all the projects, not only on the ones they have created. They can create new manager and new users for their own company. They have an important role as they can create new projects, new releases, new sprints and new tasks, matching a team for each project. Even if they can change the members of a team at any time, it is not possible to assign a project to a new team.

To finish, simple users can browse through the projects in which their team is involved. Then,they can only play with the tasks.



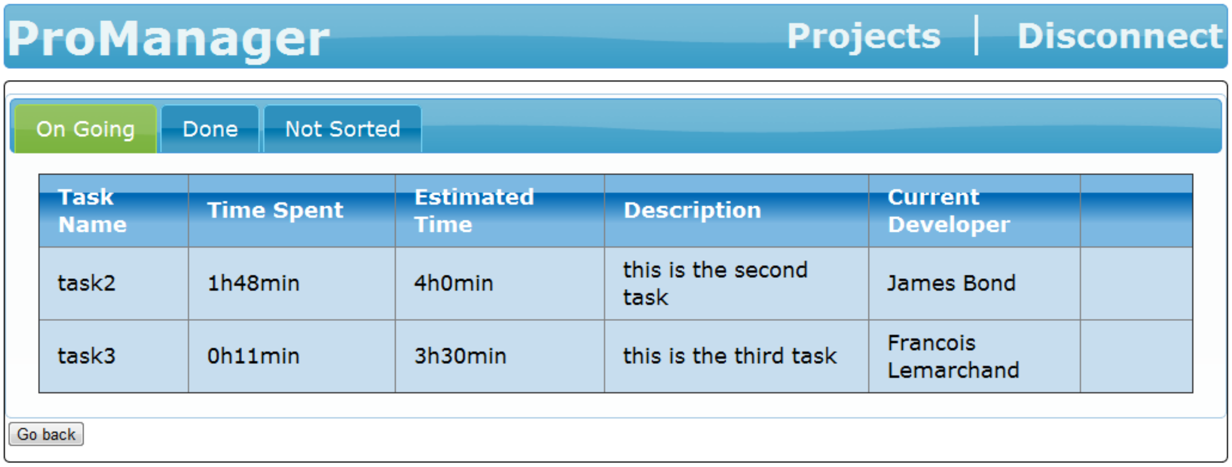
Name	Status	Team	Deadline
project1	On Going	Dream Team	April 14, 2012
project2	On Going	Dream Team	April 20, 2012

description : "This is a new project"

- beta : beta (April 14, 2012)

Main page permitting to manage projects, releases and teams

Those tasks are the key point of the tool, developers can lock a task to define it as “on going” but this will only mean that they are going to work on this task, this is a way to “book” a task. Once a task is locked, no one can have access to it, except managers who can unlock a task at any moment as long as the developer is not working on it. Developers are considered as working on a task when they use the timer feature, this feature create a session variable representing the start time, this one will be used to determine for how long the developer was working when they will stop the timer. A session variable sounded quite appropriate for this feature as if a developer forgets to stop the timer, the session variable will be destroyed and the developer won’t be able to validate 2 days later which would add 48 hours of work to the task even though the developer was not working of this time.



Task Name	Time Spent	Estimated Time	Description	Current Developer	
task2	1h48min	4h0min	this is the second task	James Bond	
task3	0h11min	3h30min	this is the third task	Francois Lemarchand	

Page displaying all the tasks from a developer's point of view

Using the logs which tell who work on a task and for how long, charts are generated and show if we are behind the schedule.

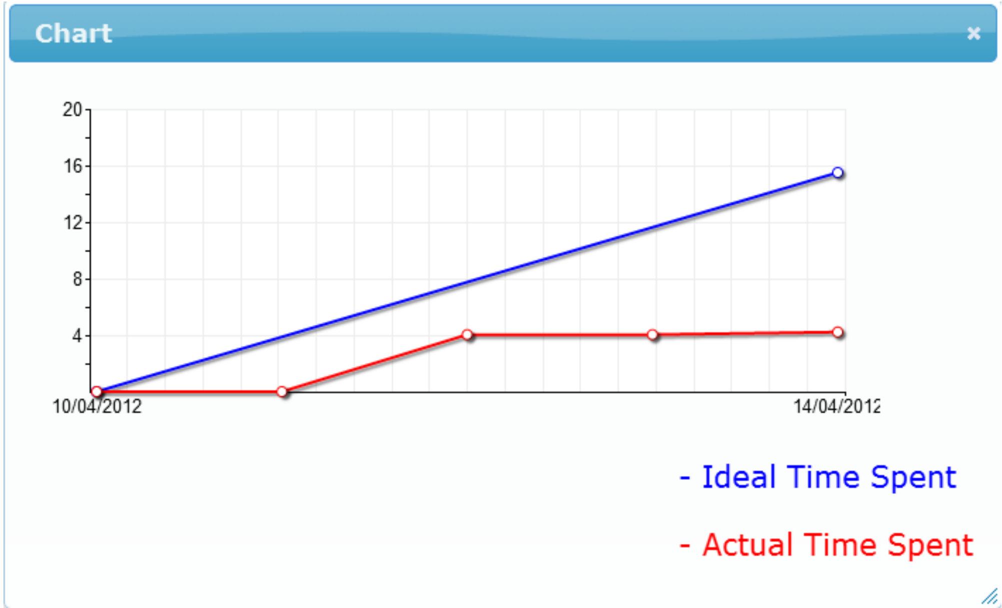
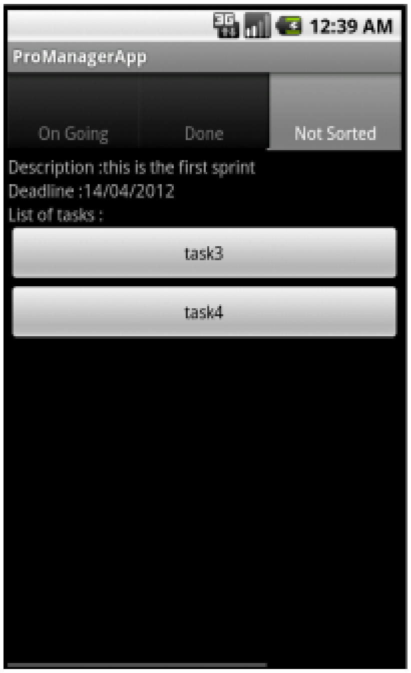


Chart showing that the project is running behind the schedule

Android application

The phone application is made to be used when we can't connect to the Internet from a computer in case we are on a train for example. The Android application has been developed essentially for developers as it permits to go through the list of projects until the list of tasks (as seen on the picture below), lock a task and exploit the timer feature as well. However, the timer feature does not work in the same way as on the web application, simply because we cannot save the start time on the phone as the application or the phone could be restarted. The start time is then saved in the database on our server.



Description :this is the first sprint

Deadline :14/04/2012

List of tasks :

Task screen on the Android App

3-Conformance to Specification and Design

The final result is quite close to what has been designed. All the main features has been implemented even though the usability could be improved, by using more Ajax for example. The web application perfectly matches with what I wanted to do but the Android application is a bit different from what I expected as I had not coded in Java for Android for 2 years. I first wanted to retrieve all the information from the first screen in which we would have logged on and then, build a screen with all the projects, releases, sprints and then, a third screen representing each sprint, with all the tasks sorted and to finish, a last screen in which we would have had all the details about a certain task.

But I finally have a login screen and then, I get the project list from the server and build a screen with the list, and then do the same thing for releases and sprints. However, the part permitting to manage the tasks is exactly as expected.

Plus, I didn't implement the "versioning" of the projects as the amount of data transmitted between the phone and the server for each update is really insignificant.

4-Learning

Even though I already had an experience with every single technology that I used, which has been done on purpose as I had to start a new project after the stoppage of my first project, this permitted me to improve my knowledge of JavaScript [JS] and Ajax [AJAX], but also update my knowledge of Java for Android as I had not coded for 2 years. I was quite surprised by all the new stuff that have been implemented in a few years, such as the GUI builder which is really easy to use and helped me to understand the XML [XML] used to build an Android interface.

Plus, the project was maybe more complex that I would have thought. It is something to develop a web application or an Android application communicating with a server. But it’s a brand new thing to develop a web application and an Android application that will work in parallel, using the same data, because we have to avoid any conflict between the two applications.

Finally, I had to work under pressure because I was only starting a new project while my classmates were already coding. This kind of situation can often in professional life so, nevertheless, it’s a good experience.

5-Review

Globally, I am really happy with the final product as all the main features have been implemented. However, I am conscious that a lot of ameliorations can be brought to those features. One of the biggest problems is that we cannot edit project information, even if I don't think a deadline should be changed, this could be handy.

Errors handling should also be improved, even though forms are checked, it will not display any errors if a mistake is made but the user will be redirect to a previous page.

Plus, supposing that a task would need several developers, this could be a problem because only one developer can work on a task, one solution could be to create a copy of the task for each extra developer. Nevertheless, this is not totally wrong because it forces managers to create tasks that are well determined.

I found only one bug in the web application, when it generates a chart for a sprint and then, for another sprint, the axis stay and the new axis are drawn over so this looks quite messy. I tried several methods to clean the canvas or apply a white rectangle on the canvas before drawing the new chart, but none of them has worked.

About the Android application, there is plenty of things that could be added such as a way to save the login, because for the moment, we have to log in every single time we launch the application which can be a bit annoying.

Despite those issues, I would have liked to modify my code and use setting files to make the applications more portable. By portable, I mean that I would have liked to create a package with the Google App Engine application [GAE] and the Android app, and then, anyone could upload the application on their own Google App Engine server. For the moment, the URL of the website on which the Android app get its information is hard-coded, so the application would need to be recompiled if the web application is uploaded on another server.

As said previously, I already had an experience with every single technology that I used so I was sure that they were perfect for this kind of project. Python [PYTH] is really more efficient than Java when dealing with Google App Engine. Plus, Python permitted me to use Django nonrel and a web framework is more than welcomed for an application of this size. Then, I would have liked to try CoffeeScript [CFFS] but there was no proper justification for that as the bits of JavaScript that I wrote were using jQuery [JQRY] libraries so I didn't need to write any complicated algorithms in JavaScript.

Then, it can be a surprise that I used Java for my Android application as I used Python for the web part, but it is just that I don't think Python for Android is mature enough, a lot of people uses Java for Android so it grows way faster and the community is being really big.

If I should start again, I would change only one thing and it would be that during my design, I would think more about the way to develop two applications that run in parallel rather than one web application and an Android app that do the same job. Even though I'm happy with the way I've solved problems while writing the Android application, I didn't think of a lot of problems before, because I thought of my web application as the main application and then, made it maybe less open than it should have been.

To conclude, the product is ready to be used but a few changes would make it user-friendlier.

6-References

[AJAX] Wikipedia, *Ajax (programming)* , http://en.wikipedia.org/wiki/Ajax_(programming)

[AND] Google, *Android*, http://www.android.com/

[CFFS] Jeremy Ashkenas, *CoffeeScript*, http://coffeescript.org/

[GAE] Google, *Google App Engine,* https://developers.google.com/appengine/

[JQRY] The jQuery Foundation, *jQuery: The Write Less, Do More, Javascript Library*, http://jquery.com/

[JS] Wikipedia, *JavaScript*, http://en.wikipedia.org/wiki/JavaScript

[PYTH] Python Software Foundation, *Python Programming Language - Official Website,* http://www.python.org/

[XML] Wikipedia, *XML*, http://en.wikipedia.org/wiki/XML