Développeur technologie Java J2E / Ajax OS Debian
Host company : Pentila Nero
Jury members : Christophe Rippert and Véronique Dudley-Beguin

Second year intern-ship

Mathilde Andre

Chambéry, the August 25, 2014

Table of contents

Introduction

Within the studies at the Enginiering school ENSIMAG in Grenoble, every student has to attend a two months internship within a company of his choice. I worked with the 10 years old company named Pentila, located in Chambéry. This internship hold my attention because it was about web development which is a field I wanted to know better. The application that the company develops used modern dynamic web technology such as Tomcat Server, ldap and j2EE programming.

Pentila is a company who develops a digital workspace inspired from the elcetronic schoolbag, it has the same goals and functionalities. It aims to facilitate the flow of informations between teachers, students and parents.

During this internship I had many tasks to do for the company. I split them into 3 main areas. They would be the development of my report. I will first discribe the company within the context of my work. Then I will focus of the main tasks I had to do. First, I will explain the softwares installations I did for the company, why and how I did them. I will then move on the testing area and the differents kind of testing I did. Finally I will explain how I fix the bugs found during the testing part. The last section would be about my feelings and outcome learning of this internship.

1 Overview and context

Présenter le contexte dans lequel vous avez effectué votre stage : description de l'entreprise en lien avec le stage : organisation, enjeux pour l'entreprise, qui vous à encadré (une personne, une équipe, ...) (1 page maximum). Le «copier/coller» des sites web des entreprises est à proscrire !

Pentila is a small company with 4 employees. They develops an application which is an electronic schoolbag. It is already used in some schools, especialy in Rhone Alpes.

This application called ENT Pentila Nero has several services individual and collectives. It allows the educationnal community to communicate easily, to organize sport activities, to manage schools restaurants... It also alows users to communicate in a very easy way. Parents can know their children's homeworks as well as their schedule. Students can hand in homework to their teachers directly on the website. It makes relation between school, parents and students easier. One of the go

All the developers work on MAC computer. I had a computer with only Windows install on it. That's why I had to install Debian OS, and all the tools to make the application work on my computer. According to me, this part is not very interesting because it is only installations and downloading on the net that's why I won't describe it in details in my report.

Présenter les objectifs du stage afin que vos évaluateurs cernent bien la limite entre l'existant et votre contribution réelle : (environ 5 pages)

- le travail à réaliser ou le problème à résoudre
- l'état de l'art des solutions existantes et des contraintes fixées par l'entreprise
- votre solution motivée à partir de l'analyse ci dessus

2 Goals

My tasks were related to differents fields such as installing new software for the team of developers, finding bugs and evolution, reporting them into an application and finally fixing some of them.

In this part I will present these tasks separatly. For each of them, I will develop the demand of the company, the limit they fix, and finally present my solution.

2.1 Redmine installation

As I said before, they are a team of 4 developers in the company. They all work with an IDE for j2EE development : Eclipse. They also use SVN, a software versionning and source code management system. They now needed a tool for project management. They wanted a software that would allow them to report bugs in a easy way and keep a track of their work. Indeed, they were in a stage where they had to test the application over and over. So they would need to write about bugs they found, assigned them to somebody in order to be solved. In a nutshell, they needed to keep track of the bugs. After some research, they found that Redmine application was a good project management tool, that's why I had to install this software.

Redmine is a tool for project management and bug-tracking. It is quite easy to begin with the software, even for a non technical person. Moreover, it is very scalable thanks to the plugins we can install with it.

The developers wanted to have an access into Redmine directly from their IDE (Eclipse), in order to know very easily what bug they had to fix or change the statue of an issue.

Redmine software has several useful plugins available. My task were to find some plugins that fit the developers demands I described before. Redmine is an open source software and is easy to install and use. First I installed Redmine into a web page shared by the company. Then I did some research about all the plugins in order to choose the most useful, the goal was to make developers work faster. At last, I set up Redmine's environment to allow developers to use it as easily as possible.

2.2 System testing

The company had some code that they didn't develop by themselves that's why they didn't know if the application was working exactly as expected but it was almost functionnal. In order to be sure it has as few bugs as possible, they were in a stage where they have to test the application as deeply as possible. I will give you an example of problems a lot of developers encountered : it is the compatibility of different browsers. When you develop a web application, you always have to install as many browsers as possible in order to test your application on them all. On the Pentila application, some services were very well designed and totally functionnal on Firefox but on Internet Explorer, the design was awfull, we couldn't read all the sentences of a post-it for example or some pictures weren't displayed.

The company had one document with an acceptance test plan for some services of the application. This document was not finished at all but it was very useful in order to continue the testing stage later. My task was first to finish this document, I had to write all the functionalities of every services into the document. Once done, I would have to test all the services of the website, and report what I found into the Redmine application.

In order to accomplish this task, first of all I did some acceptance testing to continue and finish to write the document. Then I did some regression testing. After each new version of the application, I test the services by following the steps of the acceptance test plan.

Finally, I looked at the existing softwares for functionnal tests in order to do automated tests. One could be very useful for the company, it's called JMeter. The other intern installed it and register some scenarios into it. At the end, I helped him to install a tool useful to use with JMeter sorftware, it is the jconsole that we can use with Tomcat server. I will describe a bit how I did it later in this report.

2.3 Bugs fixing

Once the bugs tracking done, the next stage was to fix the bugs reported. The bugs were related to different areas of the application such as Tomcat, ldap or programming bugs. Most of them was CSS, Html, Javascript or java EE bugs. Here is an example of bug I had to solve. On the Articles service, we could see all the articles posted by your group or by the headteacher, but the user could also write a new article for a group and add a media into it. When we clicked on the add Media button, the user could choose a file from his local computer or a file from his pigeonhole or school bag services. The file has to be with a good extension. But here is the problem : some names of files from the pigeonhole and schoolbag was displayed twice in the list.

I was asked to fix some of the bugs such as the one described above. In order to do that, I needed to have an access to the source code of the application and make it work on my computer. The company gave me a document that explain how to install the source code on OS Debian. I had to follow the steps described in it, in order to have access to all their codes and their classes.

The first step was to understand how the application worked, and then install the working environment on my computer. So I made some researches about Tomcat, Liferay, Solr and ldap and then installed everything. Once the environment working, I started to correct some bugs. I quickly understood that the bugs assigned to me would be related to different areas. That's why I followed some tutorials about all of these web programming languages. I needed to know at least how was CSS and Javascript working. I already know quite well Html.

In order to correct the bugs assigned to me, I first had to reproduce them, then I had to look at the code source to find in which class the bug was related. I looked at the navigator debugger tool of Firefox for example. This tool prooves to be very helpful to figure out what was the cause of the bug. I will describe in details the different steps to correct bugs of the application later in the Bugfix part of this report.

Décrire votre travail (environ 6 pages)

- architecture de votre solution (vision haut niveau)
- implémentation de votre solution (détail technique) ou de la partie la plus intéressante si la place manque

3 Softwares installation

In this part I will discuss the different softwares I have installed for the company. As I said in the previous part, the company needed a tool for project managing and asked me to install Redmine's software. Firstofall I will focus on the researches I have done and the plugins I finally installed. Then the details of the installation will be presented. To finish I will explain how I set up the environment and how team members can now work with this new software.

3.1 Researches

I dind't know a lot about project managing so I had to look what was Redmine for and what this application will allow the developers to do. Redmine is a powerful web based project managing which allow bugs tracking. It allows the users to create a collection of issues and then see a Gant chart given by the start and end dates of issues. You can also look at a calendar to see what happened on a particular day. The best thing about Redmine is that it is very easy to use. When working on a issue for example you can update it in one click or add notes to it very quickly. It is possible to add a lot of functionalities to this application by intalling some plugings into it.

This open-source software is written in Ruby and to install it we need to have the right versions of Ruby and Rails, depending of the redmine version we want.

The developers team needed to keep a track of building versions of their application. They also needed to have access to Redmine from Eclipse in order to see and change some issues from their IDE for example. I read about all the plugings for Redmine and I finally chose some of them. Here is a quick presentation of these plugins :

Jenkins.

This plugin was quite easy to choose, it fills exactly the need of the company. This application provides an easy to use continuous integration system. It is an application that monitors executions of repeated jobs such as bulding, testing a software project. It has 3 main features :

- It allows a team to share common information easily.
- It executes automatically compilation, testing without human intervention.
- It keeps a track of previous productions and allow us to see their development.

Mylin for eclipse connection.

Here too, the plugin fits exactly the need of the company. It is the only one that allow users to have a connection to Redmine from Eclipse. Mylin is the task and application lifecycle management framework for Eclipse. It allows to visualize tasks from Redmine repository and it has a connector to Jenkins.
It helps a developer to work efficiently with many different tasks (such as bugs, problem reports or new features). In a nutshell, it improves their productivity by reducing searching, scrolling, and navigation.

Ldap authentification.

Ldap is a protocol that allow us to access and maintain directory services. So we can access to some informations about the users of a network over TCP/IP protocol. With the ldap authentification into redmine, the users don't need to create a Redmine's account but they can directly access into their redmine's project by using their ldap password and login. It just makes the register part into Redmine easier and faster.

3.2 Implementation

I installed Redmine in ssh on an other machine. The installation was not very hard. I needed first to install Ruby and Rails, and MySQL. I created an empty database and an user, then I set up the database connection configuration. To have an access to Redmine we need to run a server. But it's a bit annoying to start manually the server every time we restart the computer. That's why I searched how we can start a server automatically when we power up the computer. I add a daemon that starts the Redmine's server, and as long as the computer is on, we can directly access to our Redmine instance.

Then I installed Jenkins plugin, meanwhile I learnt about SVN in order to add jobs into Redmine. SVN is a software versioning and revision control system, it allows developers to maintain current and historical versions of files such as source code, web pages, and documentation.

The hardest part was to integrate Mylin into Eclipse. I needed a while to make it work but finally I just used the generic web connector of Eclipse.

The integration of ldap authentification into Redmine was straightforward. I just needed to configure the authentification mode with ldap data.

Once evrything installed, I just had to set up the environment in order to allow the team's members to use it directly and easily.

3.3 Project creation

The last step was to create an instance for the project and look how to set up the environment in order to be nicest to use.

I created a new project in which I add some members with their roles such as manager, developer or rapporteur. Then I looked at the issues service. In this service it is possible to add some issues such as bugs. It is probably the service that the developers will most use. So I set up the environment by adding different types of issues such as bugs, evolution, installation, development and assistance. I also add some new issue statuses for example. When we create a new issue, we first choose the type of the issue, then its status can be "New" or "To be covered". The issue is assigned to a project member who can later change the issue status from "New" to "Solved" for example. I also add that it is mandatory to choose in which portlet the issue has been noticed. Issues can be nested within other issues, and also linked to each other.

I wrote some wiki pages related to the project. I add a jenkins page into the project in order to have an access to Jenkins directly from the project. Once the Redmine's environment set up, every member of the project could add some issue, look at the issue assigned to them in order to fix them later.

4 Software Testing

The goal of testing software is to provide informations about the quality of a product. It allows developers to know about potential bugs but it also allows the business to appreciate more the product. The testeur write a document with all the functionalities of the application and give it to the customers.

The company has a quite big application and the developers have some code that they didn't develop themselves. They need to do a lot of software testing, in order to know about bugs in their application but also about not convenient functionalities. We distinguish two kind of testing issues such as bugs and evolutions. This step of software development is essential and is quite long and boring. But thanks to this part, the customers will have a nice application, easy to use and without obvious bugs. My task was to do system testing. In order to accomplish it, I did black box testing. I didn't know anything about the internal stucture and implementation of the application and I tested functional parts of the website.

I had do to typical testing such as testing the functionalities on the website. But, I also had to test more complicated things such as the coherence between teacher, parents and students accounts. For example, in the homework notebook service, if a teacher add an homework for a class, all his sudents should see it as well as the sudent's parents.

But doing some tests without writing anything about them is not very useful for the future of the company. That's why my first task was to test all the services of the application and report all the functionalities of every services on a document test. I learnt how to write lisable tests documents with functionalities plan and it is not so obvious.

In this part of the report, I will describe my different tasks in software testing. I separate software testing in two sections that represent my work for the company: Acceptance testing and Resgression testing. So I will first describe these two kind of testing and explain what I did exactly. In the last part I will focus on a software that I used a bit for testing : JMeter.

4.1 Acceptance testing

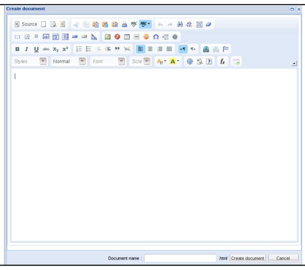
I named this section Acceptance testing but I will more talk about how to write a technical validation report than testing.

Individual software modules are combined and tested as a group. All the services can be test together, as it would be for the customers. The purpose of these tests is to "proove" that the application works fine and doesn't have big bugs. So it consists of verifying functional, performance and reliability requirements. In order to do that, the company first needed a document that explain the different functionalities of the website. This paper will be very useful for the future users of the application. My first task was to test all the functionalities of the application and report them all in a technical validation report. This document is a measuring tool which will be very useful to do regression tests. As the application is suposed to work in all web browser that the customers would be likely to use, I had to test the functionalities on these web browsers. I finally did some tests on three of them : Firefox, Google chrome and Internet Explorer.

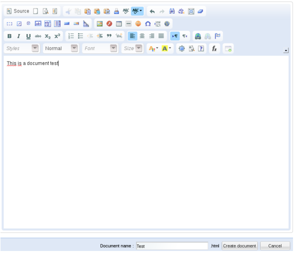
The first step was to test the functionalities and write them all in a technical validation report. In order to do so, I looked in details every services of the application to understand well what they were supposed to do. As an example, the website had a school bag service and a pigeonhole service. In the school bag service you could add some documents, oranize all your papers in differents folders and so on. Then you could drop off some of them in the pigeonhole of an other user. All the services had some tricky functionalities that I needed to know. Then I could write the technical report easier.

I will give you a detailed example that I wrote in this report. It is about the wysiwyg editor. It is used to create some web pages. In this system the user can view something very similar to the end result while the document is being created.

Access to the service

Name of the stage	Description	Expected result	Comments
Select Schoolbag service then the Create menu -> Document	Opening of the editor Wysiwyg	 Create document Source Styles Normal Font Size Document name : .html Create document Cancel	

Create a document

Name of the stage	Description	Expected result	Comments
1. Fill a description of a document 2. Choose a name : test 3. Select create document	Display of the new document in the list : test.html	 This is a document test Document name : Test .html Source Styles Normal Font Size Create document Cancel	

In the report, I created one section for each service. Then I accompanied each functionality with a formal description of the actions to perform, the eventual input data and their expected output. In the comments column, I wrote when the functionality didn't work well or had a strange or not convenient beahaviour accoding to me. But then, if the comments was some kind of bugs or evolutions, I used Redmine to report them. My descriptions have to be as high level as possible in order to be understable for the customers.

As a result of my work, the company had a useful document. It is now used to do regression testing as I will explain in the next part. And it could be used to do automated tests later, all the scenarios to test are already written. And of course it would be given to the customers. It is quite easy to evaluate the quality of the document, it has to be easy to read for the regression tests that will consist in following all the steps detailed into it.

4.2 Regression testing

It is a kind of software testing that consists to uncover new softwares bugs or regression. These tests are done after each change such as bugs fixing or new configurations settings made in the production instance. Indeed, we can manage to fix a bug but meanwhile a new bug can appear. So regression testing aim to discover these kind of new bugs, it helps to determine whether a change in one part of the software affects other parts of the software.

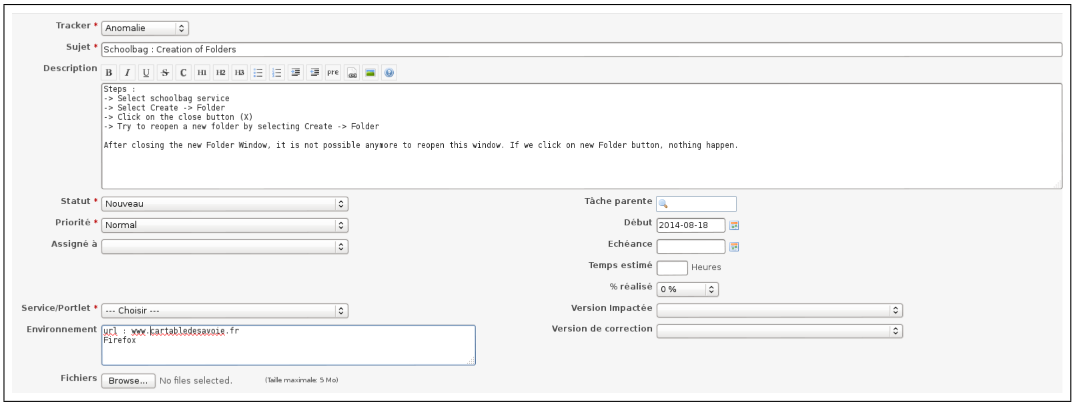
The tests I have done are called black box testing because I didn't know anything yet about the interior workings of the application, about the source code. The main advandage of this testing method is that it clearly separates user's perspective from the developer's perspective. My role was to have a beahaviour as similar as possible to the one of the future users.

The following is an example of test I had to do, which one lead to a bug I have found : On the school bag service when we clicked on the new Folder button, then on the close button. Then we couldn't click again on the new Folder button. The button was still displayed but when we clicked on it nothing would happen. I created a new Redmine issue to report this bug.

Each time the team published changement in the production instance, I had to do some regression testing. In order to do these tests, I used the technical report that I wrote before. I ran set of test-cases by folowing the steps discribed in the document. For each test, I compare the results obtained with the expected results. If there is a correct match for the test, no new bugs had appeared for this functionality. If not, I reported the bug.

To report what I noticed during a specific test, I used Redmine and created an issue. In case of a correct match it can happen that the functionality tested is less convenient for the users than before. In this case I had to create an other kind of issue in Redmine, an evolution issue. These issues are usually less urgent to do. In each issues (bugs and evolutions), I described with precision what I detected. It is very important to know in which environment the bug have been found such as Windows or Debian, Firefox, Google chrome or Internet Explorer as well as all the stages needed to reproduce the bug.

Here is an example of a creation of Redmine issue :



Tracker * Anomalie

Sujet * Schoolbag : Creation of Folders

Description

```
Steps :
-> Select schoolbag service
-> Select Create -> Folder
-> Click on the close button (X)
-> Try to reopen a new folder by selecting Create -> Folder

After closing the new Folder Window, it is not possible anymore to reopen this window. If we click on new Folder button, nothing happen.
```

Statut * Nouveau

Priorité * Normal

Assigné à

Service/Portlet * --- Choisir ---

Environnement

```
url : www.cartabledesavoie.fr
Firefox
```

Fichiers Browse... No files selected. (Taille maximale: 5 Mo)

Tâche parente

Début 2014-08-18

Echéance

Temps estimé Heures

% réalisé 0 %

Version Impactée

Version de correction

All the redmine issues was then assigned to one member of the team developers and fix by him. I will talk about this in the part called Bugfix.

4.3 JMeter and some tools

Part about JConsole and JMeter (if enough space)

Expliquer les résultats obtenus et analyser leur cohérence (environ 2 pages)

- plateforme de test mise en place, quelles métriques pour évaluer l'efficacité de votre solution
- l'adéquation avec les attentes de l'entreprise
- les perspectives ouvertes

5 Bugfix

In this part I will talk about my plan to fix bugs that some team's members assigned to me.

6 Conclusion

Faire un bilan personnel du stage : (1 page) Présenter les obstacles, points durs les plus importants et les moyens entrepris pour les résoudre ; les compétences interpersonnelles acquises en entreprise

Conclure en résumant ce qui a été effectué durant le stage (2-3 paragraphes)

7 Bibliographic reference

Références bibliographiques le cas échéant ; une documentation technique déjà rédigée pourra être jointe en annexe.