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 21, 2014

Table of contents

Introduction **3**

1 Overview and context **4**

2 Goals **5**
2.1 Redmine installation 5
2.2 System testing 6
2.3 Bugs fixing 7

3 Softwares installation **8**
3.1 Researches 8

4 Software Testing **9**
4.1 Acceptance testing 10
4.2 Regression testing 12

5 JMeter **14**
5.1 Research 14
5.2 Implementation 14
5.3 Results 15

6 Bugfix **17**

7 Conclusion **18**

8 Bibliographic reference **19**

Introduction

Introduire le sujet et l'articulation du rapport (2-3 paragraphes)

I worked during six months with the company named Pentila Nero, located in Chambéry. During this internship I had many tasks to do for the company. In this report I will first present the company, then I will describe the three main tasks I had to do : Softwares intallation, testing and debugging.

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 develop 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 easier, to oraganize easier sport activities , la gestion of schools restaurants... 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.

They all 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. I won't describe this task in details in my report.

2 Goals

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

My tasks were related to differents fields such as installing new software for the team of developers, finding new bugs and reporting them into an application then fixing bugs. In this part I will present these tasks separatly. For each of them, I will develop the demand the company, the limit they fix, and finally present my solution.

2.1 Redmine installation

They are a team of 4 employees but they didn't have a tool for project management. They wanted a software that allow them to report bugs and keep a track of their work. I had to install a software called Redmine.
Redmine is a software for project management and bug-tracking. I had to install this application in order to allow the team to track bugs in their app. It was very useful for them to have an access into Redmine directly from their IDE (Eclipse).
It has several useful plugins available. My task were to find some plugins that fit the developers demands. Redmine is a open source software and is easy to install and use. First I installed Redmine into a web page. Then I did some research about all the plugins in order to choose the most usefull, the goal was to make developers works faster. At last, I set up Redmine's environment to allow developers to us it as easyly as possible.

2.2 System testing

The application was almost functionnal but it had some small bugs. The company needed to test it over and over. For example, some services were totally functionnal on Firefox but didn't work so well on Internet Explorer. I had to test all the services of the website, to report the bugs.
The company had one document with the an acceptance test plan for some services of the application. First of all I did some acceptance testing in order to continue and finish to write de document. Then I did some regression testing.

I looked at the existing software for functionnal tests. One could be very useful for the company, it's called JMeter. The other stagiare installed it and register some scenarios into it.

2.3 Bugs fixing

Once the bugs tracking done, I had to fix some bugs. The company gave me a document that explain how to install the source code on your OS.

I had to follow it in order to have access to all their codes and their classes. To do that, I decided to first understand how the application work exactly. So I made some researches about Tomcat, Liferay, Solr and ldap and then install everything.

I had to correct some bugs very diffenrents at each other. Some about Tomcat, some about ldap but most of them was Javascript, Html or Css. That's why I followed some tutorials about all of these web programming languages. I looked at the navigator debuggers for js, html and css.

3 Softwares installation

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

In this part I will discuss the diffenrents softwares I have installed for the company. Firstofall I will focus on the research I have done. Then the details of the installation will be detailed.

3.1 Researches

4 Software Testing

The goal of testing software is to provide informations about the quality of the 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 themself. 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 descibe 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.

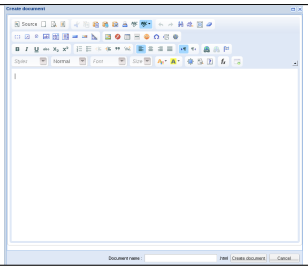
Individual software modules are combined and tested as a group. All the services can be test together, as it would be for the customers. The purpose of these tests is to "proove" that the application works fine and doesn't have big bugs. So it consists of verifying functional, performance and reliability 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 functionalitie that I needed to know. Then I could write the technical report easier.

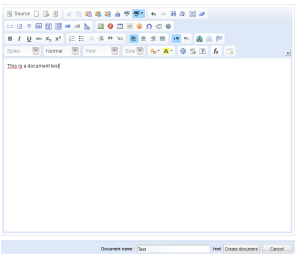
I will give you a detailed example that I wrote in this report. It is about the wysiwyg editor. It is used to create some web document. In this system in which the user can view something very similar to the end result while the document is being created.

b
b
b
b
b
b

- 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 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		

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 beahaviour accoding to me. But then, if the comments was some kind of bugs or evolutions, I used Redmine to report them. My descriptions has 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 scenario 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 changes 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.

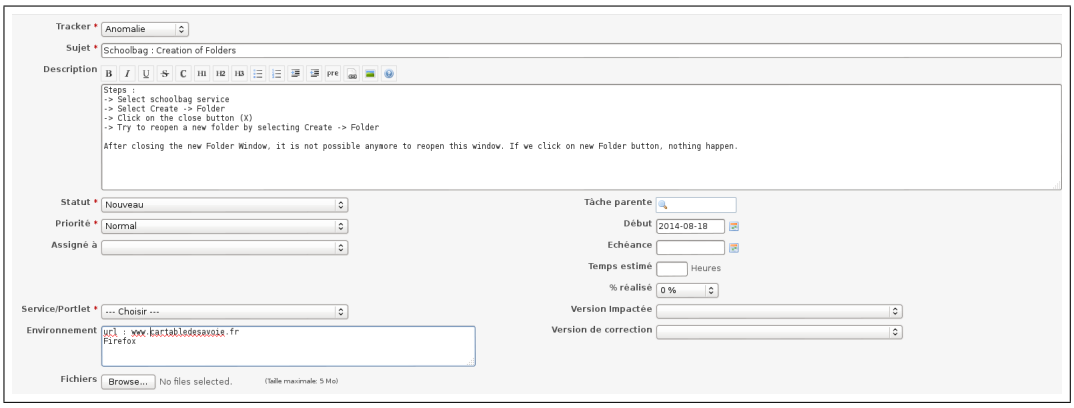
The tests I have done are called black box testing because I didn't know anything yet about the interior workings of the application, about the source code. The main advandage of this testing method is that it clearly separates user's perspective from the developer's perspective. My role was to be similar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

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 run 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 report the bug.

To report anything I notice about a step, 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, such as 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 had 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 :



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.

5 JMeter

I had to choose an application who fitted the needs of our company : a load testing application open source for measuring the performance of its website.

5.1 Research

After doing some research, only two applications held my attention : Jmeter and Gatling.

I have chosen Jmeter even though it was difficult to choose. Indeeed, I have read both good and bad commentary on the net. Jmeter is older than Gatling, hence we can find more tutorial on the net. Moreover, I have read that it is easier to make graphics results on Jmeter, and I was asked to do some graphics analysis.

Apache JMeter is a pure Java application that can be used as a load testing tool for analyzing and measuring the performance of a variety of services. It was originally designed for testing Web Applications but has since expanded to other test functions.

5.2 Implementation

Installing jmetter was not very complex. I just downloaded the release and verified that my environment met the requirements (as java 6 or some optional jars (for jdbc, jms..)). Then it was possible to run the application.

I needed to prepare some scenarios. Here is an exemple of scenario I registered :

1. click on this link
2. create a file
3. and so on.

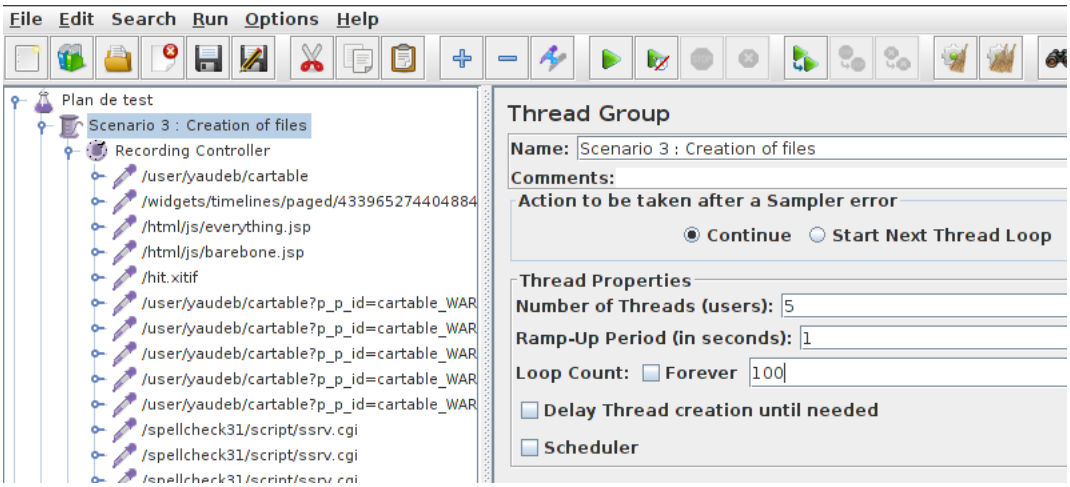
I wrote a document with some scenario to execute.

In order to register scenario, I needed to first configure the browser to use the jmetter proxy and then I recorded some script tests into jmeter. On Jmeter, we need a Test Script Recorder, in order to record every step of our scenario we do on the website.

We also need to add a Thred Group inside the Test Plan, where the scenario will be recorded thanks to the Recording Controller. And also an HTTP Request Defaults that we configured : set the web server and the port. Finally, we can start recording.

5.3 Results

Once the scenario is recorded, we start to test with the Thread Group. The following graphism is an example of what we can have after starting the test. The scenario consist in creation of files. We can see all the steps in the tree on the left part. And we can configure some parameters in the right part.



We set the number of Threads (ex:5) and Loop count (ex:100). As we can see inside the Recording Controller, a lot of different steps have been recorded. The thred will execute all of them as many times as it is asked to.

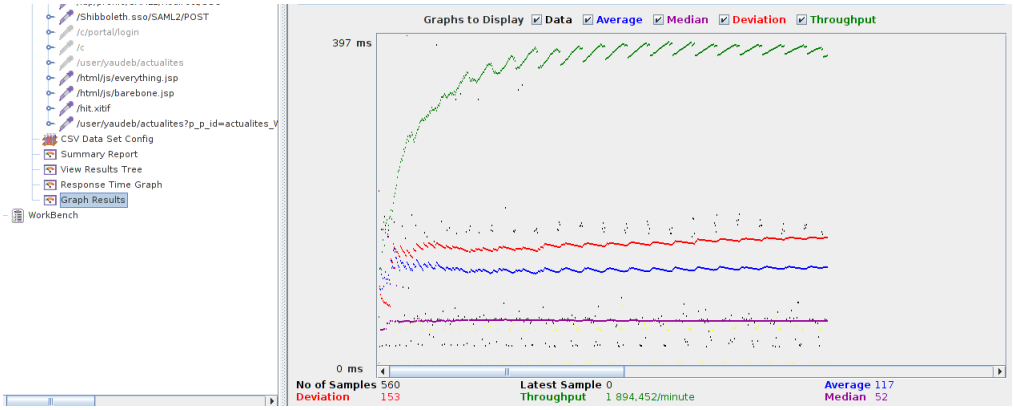
We create a summary report to see the results of the execution.

As we can see, it displays every error in each step, the throughput and so on.

We can also execute some tests with differents users. We write in a file.csv the login and password of some users and add in the application a CSV Data Set Config where we can enter the variables names:LOGIN and MDP (password) for example.

Then we add the variables inside step of the test. We can see in the tab, near username and password we enter the variables.

It is possible to view some graphics result on Jmeter:



6 Bugfix

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

7 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)

8 Bibliographic reference

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