



Institut National des Sciences Appliquées et de Technologie

UNIVERSITÉ DE CARTHAGE

Projet de Fin d'Études

Filière : Génie Logiciel

Réalisation et mise en place d'une solution Big Data au sein d'une plate-forme multi-could

Présenté par

Anis BARKAOUI

Encadrant INSAT : Mme SFAXI Lilia
Encadrant ENTREPRISE : M. LOOMIS Charles

Présenté le : --/--/2015

JURY

M. President FLEN (Président)
Mme. Rapporteur FLENA (Rapporteur)

Année Universitaire : 2014/2015

Remerciements

Merci à tous ! Bonne journée

Table des Matières

Liste des Figures	iii
Liste des Tableaux	iv
Résumé	v
Abstract	vi
Introduction Générale	1
I Présentation et cadre du projet	2
1 Présentation de l'entreprise 'SixSq'	2
2 Cloud Computing et Big Data	2
3 Méthodologie de travail	2
II Plate-formes de gestion des applications cloud	3
1 État de l'art	3
2 La plate-forme SlipStream	4
3 Comparaison	4
III Conception de la solution Big Data au sein de SlipStream	5
1 Besoins fonctionnels	5
2 Diagrammes	5
3 Architecture	5
IV Réalisation	6
1 Outils et langages utilisés	6
2 Présentation de l'application	6
Conclusion Générale et Perspectives	8
Bibliographique	9
Annexe : Remarques Diverses	10

Liste des Figures

II.1 État de l'art	3
------------------------------	---

Liste des Tableaux

Résumé

Ceci est le résumé en français de votre projet. Il devra être plus détaillé que le résumé se trouvant dans le verso de votre rapport.

Abstract

This is the english abstract of your project. It must be longer and presented in more details than the abstract you write on the back of your report.

Introduction Générale

Pour écrire un bon rapport [1] de projet en informatique, il existe certaines règles à respecter. Certes, chacun écrit son rapport avec sa propre plume et sa propre signature, mais certaines règles restent universelles [2].

La Table de matière est la première chose qu'un rapporteur va lire. Il faut qu'elle soit :

- Assez détaillée¹. En général, 3 niveaux de numéros suffisent ;
- Votre rapport doit être réparti en chapitres équilibrés, à part l'introduction et la conclusion, naturellement plus courts que les autres ;
- Vos titres doivent être suffisamment personnalisés pour donner une idée sur votre travail. Éviter le : « Conception », mais privilégier : « Conception de l'application de gestion des ... » Même s'ils vous paraissent longs, c'est mieux que d'avoir un sommaire impersonnel.

Une introduction doit être rédigée sous forme de paragraphes bien ficelés. Elle est normalement constituée de 4 grandes parties :

1. Le contexte de votre application : le domaine en général, par exemple le domaine du web, de BI, des logiciels de gestion ?
2. La problématique : quels sont les besoins qui, dans ce contexte là, nécessitent la réalisation de votre projet ?
3. La contribution : expliquer assez brièvement en quoi consiste votre application, sans entrer dans les détails de réalisation. Ne pas oublier qu'une introduction est censée introduire le travail, pas le résumer ;
4. La composition du rapport : les différents chapitres et leur composition. Il n'est pas nécessaire de numéroter ces parties, mais les mettre plutôt sous forme de paragraphes successifs bien liés.

1. Sans l'être trop

Chapitre I

Présentation et cadre du projet

Plan

1	Présentation de l'entreprise 'SixSq'	2
2	Cloud Computing et Big Data	2
3	Méthodologie de travail	2

Introduction

Une étude théorique [3] peut contenir l'une et/ou l'autre de ces deux parties :

1 Présentation de l'entreprise 'SixSq'

2 Cloud Computing et Big Data

On va présenter les domaines : Cloud et Big Data

3 Méthodologie de travail

On va présenter la méthodologie Scrum + Une comparaison avec les autres méthodologies + Outlis utilisés

Conclusion

La conclusion est en général sans numérotation, et n'apparaît pas dans la table des matières.

Chapitre II

Plate-formes de gestion des applications cloud

Plan

1	Présentation de l'entreprise 'SixSq'	2
2	Cloud Computing et Big Data	2
3	Méthodologie de travail	2

Introduction

Une étude théorique [3] peut contenir l'une et/ou l'autre de ces deux parties :

1 État de l'art

C'est une étude assez détaillée sur ce qui existe sur le marché ou dans la littérature (d'où le terme état de l'art), qui permet de répondre à la problématique. L'idée ici est de faire un comparatif entre les solutions existantes, mais surtout d'analyser le résultat de cette comparaison et de dire pourquoi ne sont-elles pas satisfaisantes pour répondre à votre problématique.



Figure II.1 – État de l'art

2 La plate-forme SlipStream

On va présenter la plate-forme SlipStream

3 Comparison

On peut faire via un tableau comparatif une comparaison entre notre plate-forme 'SlipStream' et les autres plate-formes existent sur le marché.

Conclusion

La conclusion est en général sans numérotation, et n'apparaît pas dans la table des matières.

Chapitre III

Conception de la solution Big Data au sein de SlipStream

Plan

1	État de l'art	3
2	La plate-forme SlipStream	4
3	Comparison	4

Introduction

motivation de l'application (valorisation)

1 Besoins fonctionnels

2 Diagrammes

on va présenter les digrammes : usecase + sequence //a discuter!!!

3 Architecture

Conclusion

Faire ici une petite récapitulation du chapitre, ainsi qu'une introduction du chapitre suivant.

Chapitre IV

Réalisation

Plan

1	Besoins fonctionnels	5
2	Diagrammes	5
3	Architecture	5

Introduction

Ce chapitre porte sur la partie pratique

1 Outils et langages utilisés

L'étude technique peut se trouver dans cette partie, comme elle peut être faite en parallèle avec l'étude théorique (comme le suggère le modèle 2TUP). Dans cette partie, il faut essayer de convaincre le lecteur de vos choix en termes de technologie. Un état de l'art est souhaité ici, avec un comparatif, une synthèse et un choix d'outils, même très brefs.

2 Présentation de l'application

Il est tout à fait normal que tout le monde attende cette partie pour coller à souhait toutes les images correspondant aux interfaces diverses de l'application si chère à votre cœur, mais abstenez vous ! Il FAUT mettre des imprime écrans, mais bien choisis, et surtout, il faut les scénariser : Choisissez un scénario d'exécution, par exemple la création d'un nouveau client, et montrer les différentes interfaces nécessaires pour le faire, en expliquant brièvement le comportement de l'application. Pas trop d'images, ni trop de commentaires : concis, encore et toujours.

Évitez ici de coller du code : personne n'a envie de voir le contenu de vos classes. Mais vous pouvez insérer des snippets (bouts de code) pour montrer certaines fonctionnalités [3][2], si vous en avez vraiment besoin. Si vous voulez montrer une partie de votre code, les étapes d'installation ou de configuration, vous pourrez les mettre dans l'annexe.

Conclusion

Voilà.

Conclusion Générale et Perspectives

im anis cal

C'est l'une des parties les plus importantes et pourtant les plus négligées du rapport. Ce qu'on ne veut pas voir ici, c'est combien ce stage vous a été bénéfique, comment il vous a appris à vous intégrer, à connaître le monde du travail, etc.

Franchement, personne n'en a rien à faire, du moins dans cette partie. Pour cela, vous avez les remerciements et les dédicaces, vous pourrez vous y exprimer à souhait.

La conclusion, c'est très simple : c'est d'abord le résumé de ce que vous avez raconté dans le rapport : vous reprenez votre contribution, en y ajoutant ici les outils que vous avez utilisé, votre manière de procéder. Vous pouvez même mettre les difficultés rencontrées. En deuxième lieu, on y met les perspectives du travail : ce qu'on pourrait ajouter à votre application, comment on pourrait l'améliorer.

Bibliographique

- [1] LILIA SFAXI AND SOUHEIB YOUSFI. *Pour bien écrire un rapport*. Département Math-info (2015). [1](#)
- [2] MR. LATEX. Débuter avec Latex. www.latex.com, (2008). [En ligne; consulté le 19-Juillet-2008]. [1](#), [6](#)
- [3] SOUHEIB YOUSFI AND LILIA SFAXI. *Rapport Latex*. Département Math-info (2015). [2](#), [3](#), [6](#)

Annexe : Remarques Diverses

- Un rapport doit toujours être bien numéroté ;
- De préférence, ne pas utiliser plus que deux couleurs, ni un caractère fantaisiste ;
- Essayer de toujours garder votre rapport sobre et professionnel ;
- Ne jamais utiliser de je ni de on, mais toujours le nous (même si tu as tout fait tout seul) ;
- Si on n'a pas de paragraphe 1.2, ne pas mettre de 1.1 ;
- TOUJOURS, TOUJOURS faire relire votre rapport à quelqu'un d'autre (de préférence qui n'est pas du domaine) pour vous corriger les fautes d'orthographe et de français ;
- Toujours valoriser votre travail : votre contribution doit être bien claire et mise en évidence ;
- Dans chaque chapitre, on doit trouver une introduction et une conclusion ;
- Ayez toujours un fil conducteur dans votre rapport. Il faut que le lecteur suive un raisonnement bien clair, et trouve la relation entre les différentes parties ;
- Il faut toujours que les abréviations soient définies au moins la première fois où elles sont utilisées. Si vous en avez beaucoup, utilisez un glossaire.
- Vous avez tendance, en décrivant l'environnement matériel, à parler de votre ordinateur, sur lequel vous avez développé : ceci est inutile. Dans cette partie, on ne cite que le matériel qui a une influence sur votre application. Que vous l'ayez développé sur Windows Vista ou sur Ubuntu n'a aucune importance ;
- Ne jamais mettre de titres en fin de page ;
- Essayer toujours d'utiliser des termes français, et éviter l'anglicisme. Si certains termes sont plus connus en anglais, donner leur équivalent en français la première fois que vous les utilisez, puis utilisez le mot anglais, mais en italique ;
- Éviter les phrases trop longues : clair et concis, c'est la règle générale !

Rappelez vous que votre rapport est le visage de votre travail : un mauvais rapport peut éclipser de l'excellent travail. Alors prêtez-y l'attention nécessaire.


