National Institut of Applied Sciences and Technology

UNIVERSITY OF CARTHAGE

Graduation Project

Speciality : **GL**

Flouci Developers API

Presented By

Marouane ELKAMEL

University Supervisor : **Mr SELLAOUTI Aymen**
Enterprise Supervisor : **Mr KALLEL Anis**

Presented the : –/–/**2019**

JURY

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

Annee Universitaire : 2018/2019

Remerciements

Merci e tous ! Bonne journee

Table des Matieres

Liste des Figures

Liste des Tableaux

Resume

Ceci est le resume en franeais de votre projet. Il devra etre plus detaille que le resume 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 Generale

Pour ecrire un bon rapport [1] de projet en informatique, il existe certaines regles e respecter. Certes, chacun ecrit son rapport avec sa propre plume et sa propre signature, mais certaines regles restent universelles [2].

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

— Assez detaillee [1]. En general, 3 niveaux de numeros suffisent ;

— Votre rapport doit etre reparti en chapitres equilibres, e part l'introduction et la conclusion, naturellement plus courts que les autres ;

— Vos titres doivent etre suffisamment personnalises pour donner une idee sur votre travail. eviter le : e Conception e, mais privilegier : e Conception de l'application de gestion des ... e Meme s'ils vous paraissent longs, c'est mieux que d'avoir un sommaire impersonnel.

Une introduction doit etre redigee sous forme de paragraphes bien ficeles. Elle est normalement constituee de 4 grandes parties :

1. Le contexte de votre application : le domaine en general, par exemple le domaine du web, de BI, des logiciels de gestion ?
2. La problematique : quels sont les besoins qui, dans ce contexte le, necessitent la realisation de votre projet ?
3. La contribution : expliquer assez brievement en quoi consiste votre application, sans entrer dans les details de realisation. Ne pas oublier qu'une introduction est censee introduire le travail, pas le resumer ;
4. La composition du rapport : les differents chapitres et leur composition. Il n'est pas necessaire de numeroter ces parties, mais les mettre plutet sous forme de paragraphes successifs bien lies.

1. Sans l'etre trop

Chapitre I

etude Theorique

Plan

Introduction

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

1 etat de l'art

C'est une etude assez detaillee sur ce qui existe sur le marche ou dans la litterature (d'oe le terme etat de l'art), qui permet de repondre e la problematique. L'idee ici est de faire un comparatif entre les solutions existantes, mais surtout d'analyser le resultat de cette comparaison et de dire pourquoi ne sont-elles pas satisfaisantes pour repondre e votre problematique.

Figure I.1 – etat de l'art

2 etude de l'existant

Elle est en general realisee quand on va developper un module supplementaire sur un logiciel existant, ou si on va modifier une application existante. L'etude de l'existant consiste e expliquer ce qui existe deje dans votre environnement de travail.

Conclusion

La conclusion est en general sans numerotation, et n'apparaet pas dans la table des matieres.

Chapitre II

Conception

Plan

Introduction

La partie conception de l'application ***n'est pas toujours obligatoire***. En effet, quand notre travail consiste en une etude theorique, ou une mise en place d'un systeme par exemple, il est inutile voire obsolete de faire un diagramme de classes ou de sequence.
Quand il s'agit de developpement, par contre, la partie conception s'impose.

1 Recommadations

En general, il faut suivre les regles suivantes :

— Choisir une methodologie de travail : un processus unifie, une methode agile ;

2 Diagrammes

Il faut Bien choisir les diagrammes adequats pour votre application. En general, les diagrammes obligatoires sont les diagrammes de cas d'utilisation, de classe et de sequence. Vous pouvez ajouter en plus le diagramme qui vous semble pertinent : par exemple, pour une application sur plusieurs tiers, il est interessant de montrer le diagramme de deploiement ;

— Les diagrammes doivent etre clairs, lisibles et bien expliques, sans pour autant nous submerger de details. Des explications trop longues deviennent ennuyeuses ;

— Si un diagramme est trop grand, vous pouvez le diviser, le representer sous forme de plusieurs diagrammes, ou vous abstraire de certains details. Si c'est impossible, imprimez

le sur une grande page (A3), quitte e la plier ensuite. Le plus important est que tous les mots soient lisibles.

2.1 Diagramme de Sequence

Un diagramme de sequence :

— Represente un scenario possible qui se deroule dans un cas d'utilisation. Vous n'etes donc pas obliges de montrer tous les cas d'execution possibles ;

— Represente l'interaction entre les objets : donc normalement, toutes les instances definies dans un diagramme de sequences doivent correspondre e des classes qui se trouvent dans le diagramme des classes ;

— Il existe parfois des dizaines de diagrammes de sequences possibles. Choisissez certains d'entre eux e mettre dans le rapport (2 ou 3). Privilegiez les diagrammes les plus importants (et non, l'authentification n'en fait pas partie !).

2.2 Diagramme de Classes

Un diagramme de classes :

— Doit etre fidele e l'architecture logicielle choisie. Si vous utilisez le MVC, alors les trois couches doivent etre representees dans le diagramme de classes grece aux packages ;

— Les stereotypes sont fortement conseilles. Si vous developpez une application web, n'hesitez pas e utiliser les stereotypes de la figure II.1 :

Figure II.1 – Les stereotypes

— Attention e ne pas confondre classes et tables : evitez la tentation de mettre des id partout.

Conclusion

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

Chapitre III

Realisation

Plan

Introduction

Ce chapitre porte sur la partie pratique ainsi que la bibliographie.

1 Outils et langages utilises

L'etude technique peut se trouver dans cette partie, comme elle peut etre faite en parallele avec l'etude theorique (comme le suggere le modele 2TUP). Dans cette partie, il faut essayer de convaincre le lecteur de vos choix en termes de technologie. Un etat de l'art est souhaite ici, avec un comparatif, une synthese et un choix d'outils, meme tres brefs.

2 Presentation de l'application

Il est tout e fait normal que tout le monde attende cette partie pour coller e souhait toutes les images correspondant aux interfaces diverses de l'application si chere e votre coeur, mais abstenez vous ! Il FAUT mettre des imprime ecrans, mais bien choisis, et surtout, il faut les scenariser : Choisissez un scenario d'execution, par exemple la creation d'un nouveau client, et montrer les differentes interfaces necessaires pour le faire, en expliquant brievement le comportement de l'application. Pas trop d'images, ni trop de commentaires : concis, encore et toujours.

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

2.1 Exemple de tableau

Vous pouvez utiliser une description tabulaire d'une eventuelle comparaison entre les travaux existants. Ceci est un exemple de tableau : Tab III.1.

Tableau III.1 – Tableau comparatif

	Col1	**Col2**	**Col3**	**Col4**
Row1		X		
Row2	X			
Row3	X	X	X	X
Row4	X		X	X
Row5	X		X	X
Row6	X		X	X
Row7	X		X	
Row8	X	X	X	

2.2 Exemple de Code

Voici un exemple de code Java, avec coloration syntaxique III.1.

Listing III.1 – Helloworld Java

```
  public class HelloWorld {
  //la methode main
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }

}
```

3 Remarques sur la bibliographie

Votre bibliographie doit repondre e certains criteres, sinon, on vous fera encore et toujours la remarque dessus (et parfois, meme si vous pensez avoir tout fait comme il faut, on peut vous faire la remarque quand meme : chacun a une conception tres personnelle de comment une bibliographie devrait etre).

- Une bibliographie dans un bon rapport doit contenir plus de livres et d'articles que de sites web : apres tout c'est une biblio. Privilegiez donc les ouvrages reconnus et publies pour vos definitions, au lieu de sauter directement sur le premier article wikipedia ;
- Les elements d'une bibliographie sont de preference classes par ordre alphabetique, ou par themes (et ordre alphabetique pour chaque theme) ;
- Une entree bibliographique doit etre sous la forme suivante :
  - Elle doit contenir un identifiant unique : represente soit par un numero [1] ou par le nom du premier auteur, suivi de l'annee d'edition [Kuntz, 1987] ;
  - Si c'est un livre : Les noms des auteurs, suivi du titre du livre, de l'editeur, ISBN/ISSN, et la date d'edition ;
  - Si c'est un article : Les noms des auteurs, le titre , le journal ou la conference, et la date de publication ;
  - Si c'est un site web ou un document electronique : Le titre, le lien et la date de consultation ;
  - Si c'est une these : nom et prenom, titre de la these, universite de soutenance, annee de soutenance, nombre de pages ;
  - Exemples :

    [*Bazin*, 1992] BAZIN R., REGNIER B. Les traitements antiviraux et leurs essais therapeutiques. Rev. Prat., 1992, 42, 2, p.148-153.

    [*Anderson*, 1998] ANDERSON P.JF. Checklist of criteria used for evaluation of metasites. [en ligne]. Universite du Michigan, Etats Unis. Site disponible sur : http ://www.lib.umich.edu/megasite/critlist.html.(Page consultee le 11/09/1998).
  - Dans le texte du rapport, on doit obligatoirement citer la reference en faisant appel e son identifiant, juste apres avoir utilise la citation. Si ceci n'est pas fait dans les regles, on peut etre accuse de plagiat.

Conclusion

Voile.

Conclusion Generale et Perspectives

C'est l'une des parties les plus importantes et pourtant les plus negligees du rapport. Ce qu'on ne veut pas voir ici, c'est combien ce stage vous a ete benefique, comment il vous a appris e vous integrer, e connaetre le monde du travail, etc.
Franchement, personne n'en a rien e faire, du moins dans cette partie. Pour cela, vous avez les remerciements et les dedicaces, vous pourrez vous y exprimer e souhait.
La conclusion, c'est tres simple : c'est d'abord le resume de ce que vous avez raconte dans le rapport : vous reprenez votre contribution, en y ajoutant ici les outils que vous avez utilise, votre maniere de proceder. Vous pouvez meme mettre les difficultes rencontrees. En deuxieme lieu, on y met les perspectives du travail : ce qu'on pourrait ajouter e votre application, comment on pourrait l'ameliorer.

Bibliographique

[1] Lilia Sfaxi and Souheib Yousfi. *Pour bien ecrire un rapport*. Departement Math-info (2015). 1

[2] Mr. Latex. Debuter avec Latex. www.latex.com, (2008). [En ligne ; consulte le 19-Juillet-2008]. 1, 7

[3] Souheib Yousfi and Lilia Sfaxi. *Rapport Latex*. Departement Math-info (2015). 2, 7

Annexe : Remarques Diverses

— Un rapport doit toujours etre bien numerote ;
— De preference, ne pas utiliser plus que deux couleurs, ni un caractere fantaisiste ;
— Essayer de toujours garder votre rapport sobre et professionnel ;
— Ne jamais utiliser de je ni de on, mais toujours le nous (meme 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 e quelqu'un d'autre (de preference qui n'est pas du domaine) pour vous corriger les fautes d'orthographe et de franeais ;
— Toujours valoriser votre travail : votre contribution doit etre bien claire et mise en evidence ;
— 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 differentes parties ;
— Il faut toujours que les abreviations soient definies au moins la premiere fois oe elles sont utilisees. Si vous en avez beaucoup, utilisez un glossaire.
— Vous avez tendance, en decrivant l'environnement materiel, e parler de votre ordinateur, sur lequel vous avez developpe : ceci est inutile. Dans cette partie, on ne cite que le materiel qui a une influence sur votre application. Que vous l'ayez developpe 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 franeais, et eviter l'anglicisme. Si certains termes sont plus connus en anglais, donner leur equivalent en franeais la premiere fois que vous les utilisez, puis utilisez le mot anglais, mais en italique ;
— eviter les phrases trop longues : clair et concis, c'est la regle generale !

Rappelez vous que votre rapport est le visage de votre travail : un mauvais rapport peut eclipser de l'excellent travail. Alors pretez-y l'attention necessaire.