National Institute of Applied Sciences and Technology

CARTHAGE UNIVERSITY

Graduation Project

Specialty : **Software Engineering**

A flexible data analysis library for hybrid pixel detectors

Presented by

Bechir Braham

INSAT Supervisor : **Dr. Guesmi Ghada**
Company Supervisor : **Dr. Froejdh Erik**

Presented on : **30/09/2024**

JURY

M. President FLEN (President)
Ms. Reviewer FLENA (Reviewer)

Academic Year : 2023/2024

Acknowledgements

Thank you all!

Table of Contents

List of Figures

List of Tables

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.

General Introduction

Pour écrire un bon rapport [Gre93] 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 [Gre93].

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

- Assez détaillée [1]. 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

Part I

Partie 1

Chapter I

Theoretical Study

Introduction

Une étude théorique [Knu] 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 I.1 – État de l'art

2 Étude de l'existant

Elle est en général réalisée quand on va développer un module supplémentaire sur un logiciel existant, ou si on va modifier une application existante. L'étude de l'existant consiste à expliquer ce qui existe déjà dans votre environnement de travail.

Conclusion

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

Part II

Partie 2

Chapter II

Design

Introduction

La partie conception de l'application ***n'est pas toujours obligatoire***. En effet, quand notre travail consiste en une étude théorique, ou une mise en place d'un système par exemple, il est inutile voire obsolète de faire un diagramme de classes ou de séquence.
Quand il s'agit de développement, par contre, la partie conception s'impose.

1 Recommadations

En général, il faut suivre les règles suivantes :

- Choisir une méthodologie de travail : un processus unifié, une méthode agile;

2 Diagrammes

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

- Les diagrammes doivent être clairs, lisibles et bien expliqués, sans pour autant nous submerger de détails. Des explications trop longues deviennent ennuyeuses;
- Si un diagramme est trop grand, vous pouvez le diviser, le représenter sous forme de plusieurs diagrammes, ou vous abstraire de certains détails. Si c'est impossible, imprimez le sur une grande page (A3), quitte à la plier ensuite. Le plus important est que tous les mots soient lisibles.

2.1 Diagramme de Séquence

Un diagramme de séquence :

- Représente un scénario possible qui se déroule dans un cas d'utilisation. Vous n'êtes donc pas obligés de montrer tous les cas d'exécution possibles;
- Représente l'interaction entre les objets : donc normalement, toutes les instances définies dans un diagramme de séquences doivent correspondre à des classes qui se trouvent dans le diagramme des classes;
- Il existe parfois des dizaines de diagrammes de séquences possibles. Choisissez certains d'entre eux à mettre dans le rapport (2 ou 3). Privilégiez les diagrammes les plus importants (et non, l'authentification n'en fait pas partie!).

2.2 Diagramme de Classes

Un diagramme de classes :

- Doit être fidèle à l'architecture logicielle choisie. Si vous utilisez le MVC, alors les trois couches doivent être représentées dans le diagramme de classes grâce aux packages;
- Les stéréotypes sont fortement conseillés. Si vous développez une application web, n'hésitez pas à utiliser les stéréotypes de la figure II.1 :

Figure II.1 – Les stéréotypes

- Attention à ne pas confondre classes et tables : évitez la tentation de mettre des id partout.

Conclusion

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

Chapter III

Implementation

Introduction

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

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 coeur, 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 [Knu68][Dir81], 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.

2.1 Exemple de tableau

Vous pouvez utiliser une description tabulaire d'une éventuelle 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

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

}
```

3 Remarques sur la bibliographie

Votre bibliographie doit répondre à certains critères, sinon, on vous fera encore et toujours la remarque dessus (et parfois, même si vous pensez avoir tout fait comme il faut, on peut vous faire la remarque quand même : chacun a une conception très personnelle de comment une

bibliographie devrait être).

- Une bibliographie dans un bon rapport doit contenir plus de livres et d'articles que de sites web : après tout c'est une biblio. Privilégiez donc les ouvrages reconnus et publiés pour vos définitions, au lieu de sauter directement sur le premier article wikipedia;
- Les éléments d'une bibliographie sont de préférence classés par ordre alphabétique, ou par thèmes (et ordre alphabétique pour chaque thème);
- Une entrée bibliographique doit être sous la forme suivante :
  - Elle doit contenir un identifiant unique: représenté soit par un numéro [1] ou par le nom du premier auteur, suivi de l'année d'édition [Kuntz, 1987];
  - Si c'est un livre : Les noms des auteurs, suivi du titre du livre, de l'éditeur, ISBN/ISSN, et la date d'édition;
  - Si c'est un article : Les noms des auteurs, le titre , le journal ou la conférence, et la date de publication;
  - Si c'est un site web ou un document électronique : Le titre, le lien et la date de consultation;
  - Si c'est une thèse : nom et prénom, titre de la thèse, université de soutenance, année de soutenance, nombre de pages;
  - Exemples :

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

    [*Anderson*, 1998] ANDERSON P.JF. Checklist of criteria used for evaluation of metasites. [en ligne]. Université du Michigan, Etats Unis. Site disponible sur : http://www.lib.umich.edu/megasite/critlist.html.(Page consultée le 11/09/1998).
  - Dans le texte du rapport, on doit obligatoirement citer la référence en faisant appel à son identifiant, juste après avoir utilisé la citation. Si ceci n'est pas fait dans les règles, on peut être accusé de plagiat.

Conclusion

Voilà.

Conclusion and Perspectives

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.

Bibliography

[Dir81] Paul Adrien Maurice Dirac. *The Principles of Quantum Mechanics*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.

[Gre93] George D. Greenwade. "The Comprehensive Tex Archive Network (CTAN)". In: *TUGBoat* 14.3 (1993), pp. 342–351.

[Knu] Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/~uno/abcde.html.

[Knu68] Donald E. Knuth. *The Art of Computer Programming*. Four volumes. Seven volumes planned. Addison-Wesley, 1968.

Appendix : Miscellaneous remarks

- 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 !

APPENDIX : MISCELLANEOUS REMARKS

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.