



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

Dedication

Merci e tous! Bonne journee

Table des Matieres

Liste des Figures	iv
Liste des Tableaux	v
Resume	vi
Abstract	vii
General Introduction	1
I Project Scope	2
1 Host company presentation	2
1.1 Presentation of Kaoun	2
1.2 Presentation of Flouci	3
2 Project overview	4
2.1 Project Context	4
2.2 Study of the existing	4
2.3 Project goals	4
3 Methodology	5
3.1 Agile methodology	5
3.2 SCRUM methodology	5
3.3 Lean software development	5
II Conception	7
1 Recommadations	7
2 Diagrammes	7
2.1 Diagramme de Sequence	8
2.2 Diagramme de Classes	8
III Realisation	9
1 Outils et langages utilises	9
2 Presentation de l'application	9
2.1 Exemple de tableau	10
2.2 Exemple de Code	10

3	Remarques sur la bibliographie	11
	General Conclusion	13
	Bibliographique	14
	Annexe : Remarques Diverses	15

Liste des Figures

I.1	Kaoun logo	3
I.2	Flouci logo	4
I.3	state of the art	6
II.1	Les stereotypes	8

Liste des Tableaux

III.1 Tableau comparatif	10
------------------------------------	----

Summary

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.

General Introduction

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 detaillee¹. 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 estensee 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

Project Scope

Plan

1	Host company presentation	2
1.1	Presentation of Kaoun	2
1.2	Presentation of Flouci	3
2	Project overview	4
2.1	Project Context	4
2.2	Study of the existing	4
2.3	Project goals	4
3	Methodology	5
3.1	Agile methodology	5
3.2	SCRUM methodology	5
3.3	Lean software development	5

Introduction

This chapter is dedicated to the presentation of our project's general scope. This will include an introduction of the host company Kaoun and it's main product Flouci. Also we will give an overview of the developer API project. After that, we will describe the chosen methodology that we followed during the realization of our project.

1 Host company presentation

In The first section of the report, we will introduce Kaoun, The host company that made the project possible.

1.1 Presentation of Kaoun

Kaoun is a new FinTech company that builds reliable infrastructure for payments and credits in Tunisia, and whose mission is to enable all individuals and businesses to access

financial services using any phone, anywhere, anytime.

Kaoun's first product, Flouci, is a mobile and web payment application built on top of a unique decentralized inter and intrabank infrastructure that allows instant transactions for peer to peer transfers and merchant payments. Kaoun plans to work with governments, traditional banks, mobile operators, and microfinance institutions to fix the lag between technology adoption and financial inclusion by reducing the barriers to entry for the unbanked and the underbanked.



Figure I.1 – Kaoun logo

1.2 Presentation of Flouci

Flouci is the first wallet designed to innovate mobile payment in Tunisia. It serves as a quick, easy, and convenient way to open a bank account, send and receive money, and pay different merchants in-person or online, all from within the app.

- **Open an account :**

In order to create a Flouci account, you either link your Flouci wallet to an existing bank account or follow the step-by-step KYC (Know Your Customer) guide to create an account with one of our partner financial institutions. Once you have your wallet and your QR code, you can start sending and receiving money and paying merchants using your phone.

- **Send and receive money :**

Once you create a Flouci account, you can send and receive money to and from anybody that has a flouci account/wallet. It's easy, cheap, and secure. And the best part is that it's practically instant. All you have to do is enter their number or scan their personal QR code to access their profile. You then enter the amount and confirm. You both then receive a confirmation SMS and you're done.

- **Pay merchants :**

Through Flouci, you have access to a wide range of partner merchants across Tunisia.

You can pay through the app by just scanning the QR code shown on the counter of the merchant. No waiting lines, no more looking for change or realizing you forgot your wallet at home.



Figure I.2 – Flouci logo

2 Project overview

In this section, we will present the developer API project context, we will analyze some of the existing mobile payments API's. And finally, we will indicate our project goals.

2.1 Project Context

Flouci in it's first version made it possible for users to pay merchants in simple steps and without the need of cash. However the market is rapidly shifting toward online e-commerce sites with companies like Jumia, Tayara and many other introducing their solutions in Tunisia.

In it's current implementation Flouci is not able to integrate into any form of online payments due to the lack of it's implementation.

Facing this problem and a fast moving e-commerce market the company had to move toward implementing a solution for developers.

The API should make it possible for developers to link their Flouci wallets to the their e-commerce sites and add flouci as an easy and instant payment method.

2.2 Study of the existing

2.3 Project goals

To bring developers to the Flouci world and allow e-commerce owners to introduce a mobile payment solution Kaoun has decided to build it's own developer API from scratch and provide an easy way to accept payments in few simple steps. This presented the opportunity for us to turn this problem into the main objective of our graduation project.

By the end of our project, we need to achieve these goals :

- **Create Account :**

Any developer should be able to create a Flouci developer account from the web platform, basic informations are needed to open an account.

Also it should be possible to use an existing Flouci account and switch it to developer mode.

- **Create App and link Flouci wallet :**

The web interface should offer the possibility to create an App and link an existing wallet to the specific configuration. A two steps verification system should be put in place to verify ownership of the wallet. To verify transaction developer could choose between two modes :

- **Active mode :** Activate an endpoint to verify transactions by id.

- **Passive mode :** Configure a webhook to receive transactions info once validated.

A unique token is generated for each app to allow client integration.

- **Integrate Flouci client :**

Once the App is configured the developer can easily add flouci as a payment method with the token provided.

3 Methodology

3.1 Agile methodology

3.2 SCRUM methodology

3.3 Lean software development

Conclusion

Une etude theorique [3] peut contenir l'une et/ou l'autre de ces deux parties : 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 a expliquer ce qui existe deja dans votre environnement de travail.

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

C'est une etude assez detaillee sur ce qui existe sur le marche ou dans la litterature (d'où le terme etat de l'art), qui permet de repondre a 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 répondre e votre problématique.



Figure I.3 – state of the art

Chapitre II

Conception

Plan

1	Recommandations	7
2	Diagrammes	7
2.1	Diagramme de Sequence	8
2.2	Diagramme de Classes	8

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 Recommandations

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 représenter 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 grace 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

1	Outils et langages utilises	9
2	Presentation de l'application	9
2.1	Exemple de tableau	10
2.2	Exemple de Code	10
3	Remarques sur la bibliographie	11

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 souhaite 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 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 : représente 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 metastases. [en ligne]. Université du Michigan, États 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

Voile.

General Conclusion

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 écrire 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](#), [10](#)
- [3] SOUHEIB YOUSFI AND LILIA SFAXI. *Rapport Latex*. Departement Math-info (2015). [5](#), [10](#)

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.


