

REPUBLIQUE DU CAMEROUN

PAIX-TRAVAIL-PATRIE



INSTITUT AFRICAIN D'INFORMATIQUE

« Centre d'Excellence Technologique Paul Biya »

BP: 13719 Yaoundé, Cameroun

Tel: +237 22 72 99 57

Site web: www.iai-cameroun.com

Email: iaicameroun@yahoo.fr

REPUBLIQUE DU CAMEROUN

PAIX-TRAVAIL-PATRIE



RAPIDOS Sarl

BP: 6087 Douala, Cameroun

Tel: +237 683 128 441 / +237 695 400 777

Site web: www.rapidos.cm

Email: contact@rapidos.cm

RAPPORT DE STAGE ACADEMIQUE

THEME: ANALYSE, CONCEPTION ET REALISATION D'UNE PLATEFORME WEB ET MOBILE D'ACHATS ET LIVRAISON DE REPAS A DOMICILE: CAS DE RAPIDOSEAT

Stage effectué du 01^{er} juillet 2022 au 30 septembre 2022 en vue de l'obtention du **Diplôme d'ingénieur des travaux informatiques** option Génie Logiciel

Rédigé par :

TAGAKOU FOGANG Hammel Brayan

Étudiant en 3^{ème} année à l'IAI-CAMEROUN, option Génie Logiciel

Sous l'encadrement :

Académique de :

Mme. BELINGA Estelle

Sous-directeur chargée de la filière Génie Logiciel IAI-Yaoundé

Professionnel de:

M. NKOM TAKEU Guy

Directeur Général de RAPIDOS Sarl

Année académique: 2021-2022



DEDICACE

*À mes
Parents*

REMERCIEMENTS

Nos sincères remerciements s'adressent :

- ❖ Au Représentant Résident de l'IAI-Cameroun « Centre d'Excellence Technologique Paul BIYA » Monsieur **ABANDA Armand Claude**, pour la structure académique qui nous offre une formation et un suivi digne de ce nom ;
- ❖ A Monsieur **TAKEU NKOM Guy Junior**, Directeur général de Rapidos Sarl et encadrant professionnel, pour son encadrement, ses conseils, sa présence, et son dévouement dans le travail ;
- ❖ A Madame **BELINGA Estelle**, Sous-directeur chargée de la filière Génie Logiciel et encadrant académique, pour ses remarques, ses conseils et sa disponibilité ;
- ❖ A mes parents **FOGANG Deschanel** et **FOGANG Yolande**, **DJANGA Stéphane** et **DJANGA Manuella** pour tout le soutien moral et financier ;
- ❖ A Monsieur **NJIPMEGNI TANKEU Franck** qui a été un propulseur dans ma modeste carrière de développeur ;
- ❖ A Monsieur **MBA TAGUEU Franck** qui a été une source de motivation et d'inspiration.
- ❖ A tout le personnel de l'entreprise Rapidos Sarl pour leur présence et leur soutien quotidien tout au long de notre séjour ;
- ❖ A Mme **KAMTCHOUA Jacobine** pour le soutien incommensurable ;
- ❖ A M. **TAGAKOU Jean** pour les multiples conseils ;
- ❖ A Monsieur **SIGNE Juste** pour le soutien quotidien ;
- ❖ A nos frères et sœurs pour leur esprit de solidarité ;
- ❖ A tous nos camarades de l'IAI-Cameroun, qui m'ont soutenu durant ce stage ;

SOMMAIRE

DEDICACE	I
REMERCIEMENTS.....	II
SOMMAIRE	III
LISTE DES FIGURES	IV
LISTE DE TABLEAUX.....	VI
LISTE DES ABREVIATIONS.....	VIII
RÉSUMÉ	IX
ABSTRACT.....	X
INTRODUCTION GENERALE	1
DOSSIER D'INSERTION.....	2
CAHIERS DES CHARGES	14
DOSSIER D'ANALYSE	29
DOSSIER DE CONCEPTION	70
DOSSIER DE RÉALISATION	82
TESTS DE FONCTIONNALITÉS.....	94
GUIDE D'UTILISATION.....	104
CONCLUSION GENERALE.....	115
ANNEXES	A
BIBLIOGRAPHIE.....	B
WEBOGRAPHIE	C
GLOSSAIRE.....	G
TABLE DES MATIERES	H

LISTE DES FIGURES

Figure 1: Localisation Rapidos Sarl.....	6
Figure 2: Organigramme de Rapidos SARL (Source: Rapidos Sarl)	9
Figure 3:Diagramme de gant du projet	26
Figure 4:Diagramme de PERT.....	27
Figure 5: Sprint de la méthodologie Scrum	42
Figure 6:Sprint 1 du projet Rapidos Eat	43
Figure 7:Diagramme de cas d'utilisation global du système	46
Figure 8:Diagramme du cas d'utilisation commander repas	47
Figure 9:Diagramme du cas d'utilisation Gérer Produit.....	48
Figure 10:Diagramme du cas d'utilisation gérer utilisateur	49
Figure 11:Diagramme du cas d'utilisation Gestion des véhicules.....	50
Figure 12:Diagramme du cas d'utilisation gérer son profil	51
Figure 13:Exemple de diagramme de séquence.....	56
Figure 14:Diagramme de séquence de l'Authentification avec JWT token	57
Figure 15:Diagramme de séquence Ajouter un partenaire	58
Figure 16:Diagramme de séquence d'inscription	59
Figure 17:Exemple diagramme d'activité	61
Figure 18:Diagramme d'activité Modifier compte restaurant	62
Figure 19:Diagramme d'activité Passer commande	63
Figure 20:Diagramme d'activité accepter commande.....	64
Figure 21:Diagramme de communication ajouter produit.....	66
Figure 22:Diagramme de communication effectuer payement.....	67
Figure 23:Diagramme de communication effectuer recherche.....	68
Figure 24: Exemple de diagramme de classe.....	74
Figure 25:Diagramme de classe du système	75
Figure 26:Diagramme de package du système	76
Figure 27: Exemple de diagramme d'états-transition	78
Figure 28: Diagramme d'états-transition du système.....	78
Figure 29:representation d'un objet.....	79
Figure 30:Exemple de diagramme d'objet.....	79
Figure 31: Diagramme d'objet du système	80

Figure 32: Architecture MVC	86
Figure 33: Exemple de diagramme de composant	88
Figure 34: Diagramme de composant du système	88
Figure 35: Représentation architecture n-tiers (Source : http://cedric.cnam.fr/~farinone/SMB111/annee1011/architectureNTiers.pdf)	90
Figure 36:Exemple de diagramme de déploiement	91
Figure 37:Diagramme de déploiement du système.....	92
Figure 38: Test d'inscription	98
Figure 39: Test de connexion.....	99
Figure 40: Test de catégories	100
Figure 41:Test de produits	101
Figure 42: Test d'ajout de restaurant	101
Figure 43:Guide d'installation1	107
Figure 44:Guide d'installation2.....	107
Figure 45:Icone de lancement de l'application.....	107
Figure 46:Page d'accueil Rapidos Eat.....	108
Figure 47:Page d'inscription RapidosEat	108
Figure 48:Drawer page RapidosEat	109
Figure 49:Page de connexion RapidosEat	109
Figure 50:Page d'accueil des utilisateurs authentifiés.....	110
Figure 51:Page d'accueil des restaurants	110
Figure 52:Single Product page.....	111
Figure 53:Panier sur RapidosEat	111
Figure 54:Page adresse de livraison.....	112
Figure 55:Page facture	112
Figure 56:Facture PDF RapidosEat	113

LISTE DE TABLEAUX

Tableau 1: Ressources matérielles de Rapidos Sarl (Source: Rapidos Sarl).....	10
Tableau 2: Ressources logiciel Rapidos (Source: Rapidos Sarl)	11
Tableau 3:Ressources matérielles (mercuriale 2022)	21
Tableau 4: Ressources logicielles et abonnements (site officiel des produits)	22
Tableau 5: Ressources humaines (Source : D'après le planning financier de l'entreprise Rapidos)	25
Tableau 6:Estimation du coût du projet (Source : Calcul prévisionnel)	25
Tableau 7:Etat de l'art	32
Tableau 8: critiques et limites de l'existant	34
Tableau 9:Tableau comparatif entre MERISE et UML.....	35
Tableau 10:Tableau de comparaison processus unifiées, source (https://www.studocu.com/row/document/universite-de-tunis/genie-logiciel/quiz-5-5-avril-2020-questions/8012662).....	38
Tableau 11: Formalisme du diagramme de cas d'utilisation (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	44
Tableau 12: Relations du diagramme de cas d'utilisation (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	44
Tableau 13: Formalisme du diagramme de séquence (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	55
Tableau 14: Formalisme diagramme d'activité (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	60
Tableau 15: Formalisme diagramme d'activité (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	65
Tableau 16: Formalisme diagramme de classe (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	74
Tableau 17: Formalisme diagramme de package (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	76
Tableau 18: Formalisme Diagramme d'états-transition (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	77
Tableau 19 : Equipe projet.....	85
Tableau 20:Langages et Framework du projet.....	85

Tableau 21: Formalisme diagramme de composant (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	87
Tableau 22:Formalisme du diagramme de déploiement (Source: http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf).....	91

LISTE DES ABREVIATIONS

AGL : Atelier de Génie Logiciel

API: Application Programming Interface

CSS: Cascading Style Sheets

HTML : Hypertexte Markup Language

IAI : Institut Africaine d’Informatique

IOS: IPHONE Operating System

JDBC: Java DataBase Connectivity

JPA: Java Persistance API

MVC : modèle Vue Contrôleur

SGBD : Système de Gestion de Base de Données

SI : Systèmes d’Information

TIC : Technologie de l’information et de la Communication

UML: Unified Modeling Language

USB: Universal Serial Bus

VPS : Virtual Private Server

RÉSUMÉ

Pendant 3 mois de stage académique, nous avons travaillé assidûment sur le thème qui s'intitule **analyse, conception et réalisation d'une plateforme web et mobile d'achats et de livraison de repas à domicile dans le cas de RapidoSEAT**. Ce thème est une solution brillante en ceci qu'il impacte directement sur la **pandémie Covid-19** qui jusqu'aujourd'hui sévit encore dans le monde. Ainsi dit, ce service en ligne permet le **respect des mesures barrières établies par notre Etat camerounais et s'applique particulièrement sur la distanciation sociale**. Cette plateforme permettra d'améliorer la vie des camerounais et aura pour objectifs de référencer les restaurants et supermarchés sur l'étendue du territoire national, mettre leur repas et produits agro-alimentaires à disposition, donner la possibilité au client de **passer une commande et de se faire livrer à domicile**, et enfin permettre aux partenaires d'obtenir des conseils sur leurs statistiques en entrées et sorties. Son système a donc pour principaux acteurs: **le client, le restaurant ou supermarché, le livreur, et enfin l'administrateur (RAPIDOS)** qui gère tout se dit système. Ce projet regroupe en son sein deux (02) modules à savoir le front-office et le back-office. Nous nous sommes appesantis sur le module front-office, qui regroupe toutes les fonctionnalités accessibles au client. Un projet comme celui-ci fait appel à plusieurs ressources tant matérielles que logicielles et humaines ; ce qui en somme ramène à un budget très important. Pour sa réalisation nous avons utilisé des langages et Framework tel que dart, Java, Flutter comme Framework Dart, et Spring boot comme Framework Java ; pour la modélisation, notre choix à porter sur le langage UML qui se rapproche bien plus de l'utilisateur, à côté duquel la méthode Agile SCRUM pour son côté minutieux qui permet d'aller étape par étape tout en étant sûre de l'avancement au travers des différents tests effectués.

Mots clés : Plateforme ; Pandémie ; Livraison;

ABSTRACT

During 03 months of internship, we were given as theme: **Analysis, Conception and Realization of a web and mobile platform for sales and food delivery at home: case of RapidosEat.** This theme is a brilliant solution because it would have a positive impact on the **Covid-19 pandemic** which is still killing some lives all around the globe. With all this said, this online service **covid barrier measure to limit the rate of contamination** established by our government. This platform shall enable the amelioration of the standard of living of every Cameroonian and shall have as objectives: reference all restaurants and super markets on the national territory of Cameroon, make available food and agri-food products for clients, give the possibility to clients to command and receive their meal at home and finally enable our various partners obtain some advice on their stock (inputs and outputs). Thus, the actors of this system are: **the client, restaurants or supermarkets, the delivery agent and finally the administrator RAPIDOS** who is in charge of all the system. This project has two main modules: front office and the back office. We handled most of the time the front office, who regrouped all the functionalities accessible to the client. A project of this type needs a lot of resources (material, software and labour); thus, given us a very high budget. To realize this project, we used a variety of languages and Frameworks such as dart, Java, Flutter as Dart Framework and Spring boot as a Java Framework; in order to create our various diagrams, we used a modeling the **UML modeling language** closed to the clients, a method for project management such as **SCRUM** which would permit us step by step to ensure the adequate realization with various tests to ensure that all functionalities are well developed.

Key words: Platform; Pandemic; Delivery;

INTRODUCTION GENERALE

Pendant l'année académique 2019-2020 à l'IAI Cameroun, nous avons été dans l'obligation de nous confiner à domicile, étudiants comme membres du personnel de l'institut, suite à la **pandémie Covid-19** qui venait de surgir à l'échelle mondiale. Cette phase de crise étant brusque et inconnue de tous, à causer assez de dégâts notamment la suspension temporaire des cours et a éveillé aussitôt les consciences à rapidement trouver des solutions pour pallier ce problème, et ainsi reprendre de vue les différents objectifs à atteindre. C'est dans cette lancée qu'étant avant tout un établissement de technologie, nous avons continué à poursuivre les cours à distance en téléconférence grâce aux plateformes appropriées en ligne. C'est ainsi que beaucoup d'autres secteurs ont donc opté, comme nous, des solutions au travers des **services en ligne**. C'est dans ce domaine même que notre entreprise d'accueil **Rapidos Sarl** excelle. **Rapidos Sarl** est spécialisée dans le développement web, les services de voyage pour études et même des vacances à l'étranger, les services d'aide à domicile, et est également prestataire d'un service de livraison. **RapidosEat** est donc ce service de livraison sur lequel nous allons nous appesantir tout au long de nos 3 mois de stage sur le thème **analyse, conception, et réalisation d'une plateforme web et mobile d'achats et de livraison de repas en ligne**. Pour présenter notre expérience de stage au sein de Rapidos Sarl, il apparaît pertinent de présenter à titre préalable notre environnement d'accueil, ensuite d'élaborer un cahier de charges qui mettra en exergue les objectifs à atteindre, puis suivra un document d'analyse dans lequel une étude et une critique de l'existant seront effectuées, en plus le dossier de conception, et enfin le dossier de réalisation qui présentera les outils utilisés pour la réalisation de cette solution, et en définitive le guide d'utilisation du produit développé.



DOSSIER D'INSERTION

RESUME

Le dossier d'insertion décrit le déroulement de l'accueil et de l'intégration des stagiaires dans la structure d'accueil, présente l'ensemble des informations recueillis pendant la période d'insertion aussi bien sur l'entreprise que sur la tâche à accomplir pendant la période de stage.

APERÇU

INTRODUCTION

- I. ACCEUIL ET INTÉGRATION
- II. RESENTATION DE LA STRUCTURE
- III. PRESENTATION DU THEME

CONCLUSION



INTRODUCTION

La phase d'insertion est le début de notre stage académique. Dans notre cas, à savoir celui de l'IAI Cameroun, cette phase est d'une durée limite de deux (2) semaines. Au cours de cette insertion, le stagiaire prend contact avec les membres de la structure parmi lesquels son encadreur professionnel, s'informe sur le fonctionnement de l'entreprise et en particulier celui de la cellule informatique. Ce document contient en son sein tout d'abord le thème qui a été affecté au dit stagiaire, ensuite les tâches effectuées durant notre insertion dans l'entreprise **Rapidos Sarl**. Tout au long de ce rapport, il sera question pour nous de mettre en exergue d'une part l'accueil au sein de l'entreprise, et d'autre part sa structure complète à travers une présentation détaillée.

I. ACCEUIL ET INTÉGRATION

1. Accueil

A notre arrivée au sein de l'entreprise **Rapidos** le lundi 04 Juillet 2022 à 8h, nous avons été briefés sur son fonctionnement mais aussi sur l'objectif à atteindre durant les quatre (03) prochains mois de stage qui nous ont été accordés. Après plusieurs séances de sprints chaque matin (réunion de travail mentionnée dans la méthode Agile SCRUM), nous étions fixés sur les projets à réaliser, les différentes technologies et outils que nous devions utiliser ainsi que les horaires de travail. Après cet entretien, nous avons pris part à une réunion sur Zoom avec **M. Bertrand NGANSOP**, le fondateur de **Rapidos Sarl** actuellement domicilié en Allemagne ; Dans le cadre de notre stage académique, **M. NKOM TAKEU Guy** a été désigné comme notre encadreur professionnel.

2. Intégration

Suite à cet accueil chaleureux notre encadreur professionnel nous a fait passer quelques tests de performance pour juger nos différents niveaux afin de combler les vides en connaissances technologiques pour une intégration efficace. Après avoir choisir une technologie Backend et Frontend pour la première semaine de notre stage, il a été question pour nous de réaliser une application CRUD ainsi qu'une authentification simple sans token, ni providers externes. Comme premier test réussi avec succès, nous avons choisi de créer une interface et une base de données en interactions permettant d'afficher, d'ajouter, et de supprimer des produits ainsi que leur prix et description ceci à la fois sur appareil mobile que sur pc avec un navigateur web. Chacun de nous avons créé un compte GitLab lié à son Gmail au format prenom.rapidos@gmail.com Les projets CRUD ont été rendus sur ces différents comptes GitLab avec notre encadreur professionnel modérateur sur tous nos projets GitLab.

II. PRESENTATION DE LA STRUCTURE

1. Historique

Rapidos est une jeune entreprise qui a vu le jour le **02 Août 2020 à Bepanda**, plus précisément sur la rue face **au commissariat 7^{ème} de Douala**. L'idée du promoteur **M. Bertrand** était de mettre sur pied un projet permettant d'améliorer le standard de vie des camerounais, de créer une plateforme appelée **RapidosEat**, pour la livraison de repas cuisinés par les restaurants partenaires. Cette idée fut l'un des premiers services de **Rapidos**, ce projet par sa diversité a incité à s'étendre sur d'autres secteurs d'activités précisément le secteur de

la santé appelé **RapidosCare**, tout en considérant le fait que la santé n'a pas de prix et qu'il pourrait apporter plus ou moins une solution palliative à l'état camerounais en ce qui concerne le secteur médico-social du pays.

2. Missions

La mission principale de **Rapidos** est d'améliorer les conditions de vie des camerounais, de manière spécifique, il a pour missions :

- D'offrir des services de livraison de qualité, rapide et à moindre coût ;
- De donner un accès plus facile à tous produits sur tout le territoire national du Cameroun ;
- D'impacter les secteurs économiques, sociaux, technologiques du pays.

3. Situation géographique

Le siège social de **Rapidos** est situé dans la capitale économique du Cameroun, Douala, plus précisément au quartier dit **CAMTEL Bepanda**, sur la rue face **au commissariat 7^{ème}** dite **nouvelle route 7^{ème}**. L'image ci-dessous est une esquisse du plan de localisation de **Rapidos Sarl** :



Figure 1: Localisation Rapidos Sarl

4. Activités réalisation et partenaires

i. Activités

Rapidos Sarl est un regroupement de plusieurs micro-services qui sont les suivants :

- ❖ Prestataire de service dans le domaine de la santé à travers livraison des médicaments : **RapidosCare** ;
- ❖ Prestataire de service dans la logistique pour vos voyages et études à l'étranger : **RapidosTravel** ;
- ❖ Prestataire de service d'aide à domicile : **RapidosAid** ;
- ❖ Prestataire de service de location de véhicules : **RapidosAuto** ;
- ❖ Prestation de service de livraison des colis : **Rapidos Service** ;
- ❖ Prestataire de service informatique création de sites web, applications, et même formations : **RapidosTech** ;
- ❖ Prestataire de service de livraison en ligne de vos repas favoris : **RapidosEat**. C'est dans cette dernière que nous sommes intervenus ;

ii. Réalisations

De nombreuses réalisations ont été faites par la Startup Rapidos notamment de par ses différentes filiales :

➤ RapidosTech :

- Réalisation des sites client de l'entreprise **Digital Desk**
- Conception et réalisation du site e-commerce pour la boutique **Ma'a Magni Food**
- Refonte du site vitrine du supermarché **Santa Lucia**.
- Conception et réalisation du site e-commerce de l'entreprise **Bricotech**.
- Conception et réalisation du site vitrine la **Financière**.
- Conception et réalisation du site vitrine de l'agence de comptabilité **Audit Vision**.

➤ RapidosEat :

- Assurer la livraison des restaurants comme **Lepop** et **KMC**.
- Signature de partenariat avec plusieurs restaurants et tournes de la ville de Douala

iii. Partenaires

Comme partenaire l'entreprise **Rapidos** en Comme plusieurs et de renommée tant à l'échelle national qu'à l'échelle internationale. Nous pouvons citer :

- **Santa Lucia** : Pionniers de la distribution au Cameroun, Les supermarchés Santa Lucia sont un espace de distribution, de ventes en gros et détails des produits alimentaires, vestimentaires, électroniques, électroménagers, parfumerie et cosmétique, etc.
- **Lepop** : Le restaurant est largement connu pour son poulet frit parmi tous les plats délicieux qu'il sert.
- **K My Choice** : Fast food restaurant
- **Afarik** : Nous voulons permettre l'accès facile à tous, aussi bien aux produits de base qu'aux produits innovants. Nous aspirons à permettre aux petites et moyennes entreprises de tirer parti de l'innovation et de la technologie pour croître et rivaliser plus efficacement dans les économies nationales et mondiales.
- **Le Rosty** : est le restaurant qui offre le meilleur des deux mondes est enfin à votre portée. L'ambiance familiale qu'elle offre tout en servant certaines des meilleures cuisines africaines à un prix abordable est tout simplement remarquable. Chaque fois que vous entrez dans ce restaurant, vous serez submergé par le confort, le charme terre à terre qui y règne, le goût des boissons faites maison et des plats incroyables servis ici.
- **Bricotech Equipements**
- **La financiere** : spécialisés dans le refinancement, refinancement hypothécaire, le financement privé ou encore le prêt en financement de capital afin de cerner les besoins réels de nos clients. Nous vous offrons de précieux conseils financiers ainsi qu'un service pour refaire votre cote de crédit, au besoin.
- **Audit Vision**
- **Digital Desk**
- **Tik Tac Market**
- **Francy Shop**
- **Ma Magui Food**

5. Organigrammes

Le schéma suivant illustre l'organigramme de l'entreprise, Rapidos Eat est le département où j'ai été conduit :

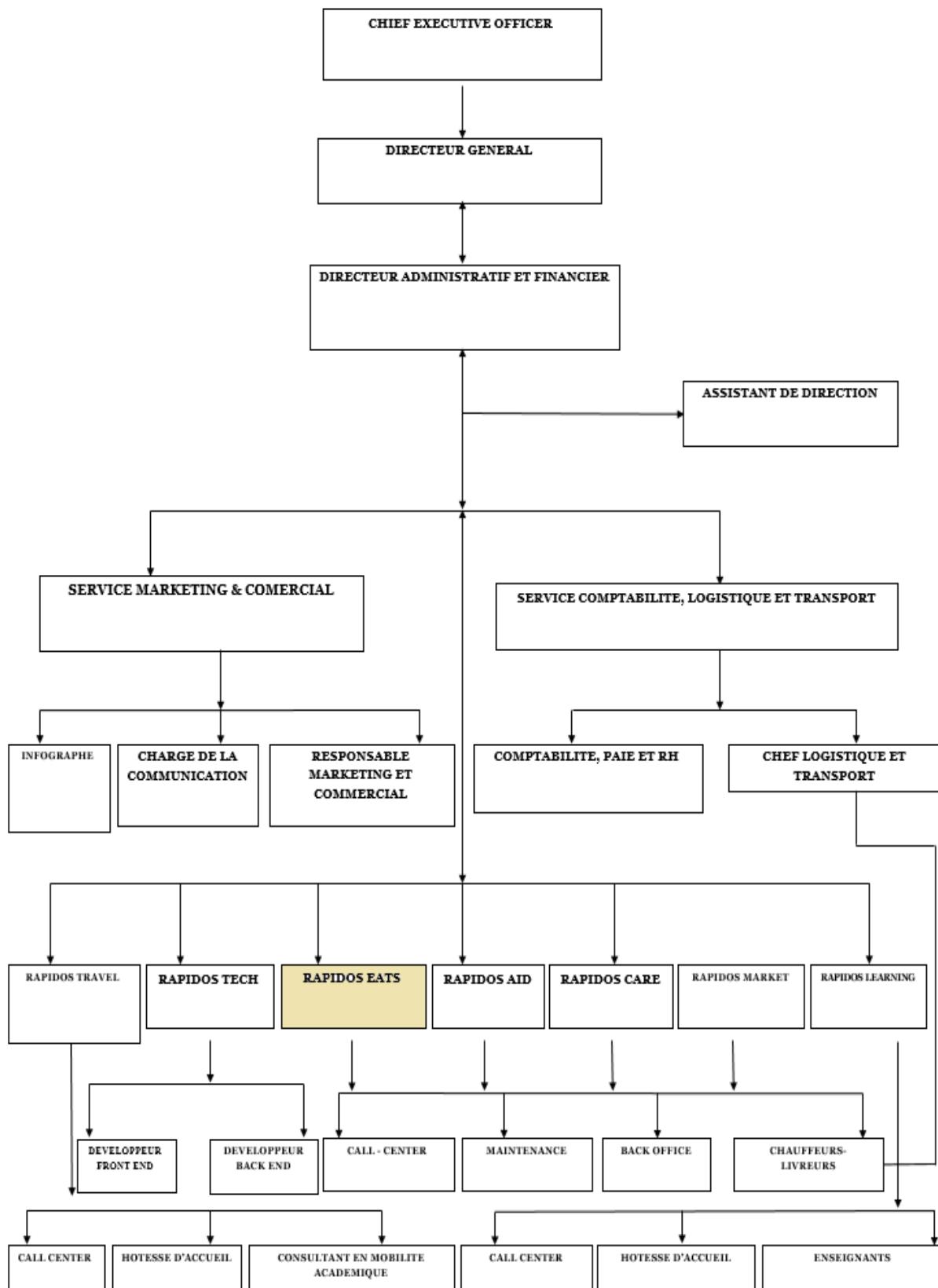


Figure 2: Organigramme de Rapidos SARL (Source: Rapidos Sarl)

6. Ressources de l'entreprise

i. Ressources matérielles de Rapidos Sarl

Dans notre cas, nous nous référerons à l'ensemble des équipements informatiques utilisés pour le traitement automatique, d'envoie, de réception, et/ou de restitution des informations. Il s'agit du parc informatique de **Rapidos Sarl**. Le tableau ci-dessous nous présente ses différentes ressources matérielles.

Tableau 1: Ressources matérielles de Rapidos Sarl (Source: Rapidos Sarl)

Quantité	Nom	Rôle	Caractéristiques
1	Routeur Huawei	Est un équipement matériel informatique qui se charge de diriger et orienter transit des données à travers un réseau.	Peut supporter jusqu'à 32 postes
1	Serveur VPS (Virtual Private Server)	Permet d'offrir aux utilisateurs un accès avec un serveur émulé reposant sur des ressources d'un serveur physique grâce à un hyperviseur.	<ul style="list-style-type: none"> - RAM : 8Go - Processeur 2.4GHz - Système : CentOS
3	HP utilisés comme poste de travail	Terminaux de travail.	<ul style="list-style-type: none"> - RAM : 8Go - Processeur 2.4GHz - Système : Windows 10
3	Switch pour la création de vlan	Il permet de relier en réseau différents équipements informatique entre eux, et contribue aussi à la sécurité du réseau d'entreprise.	25 ports RJ45
4	Pc jouant le rôle de serveur de test	Appareil utilisé pour stocker les fichiers et gérer la base de données de l'entreprise.	<ul style="list-style-type: none"> - RAM : 8Go - Processeur : core i7 2.3Ghz

- HDD : 700Go

ii. Ressources logicielles de Rapidos

Les ressources logicielles représentent l'ensemble des programmes utilisés pour le traitement de l'information, le suivi, la supervision des différents services.

Tableau 2: Ressources logiciel Rapidos (Source: Rapidos Sarl)

Nom	Type	Rôle
IntelliJ IDEA	Outil de développement	Est un environnement de développement intégré nous permettant de développer sur la technologie Java (Spring boot)
Postman	Outil de développement	Permet d'interroger les services web avec des appels http directement depuis une interface graphique
PostgreSQL (pgAdmin)	Outil de développement	Outil d'administration graphique pour PostgreSQL. Qui joue ici le rôle de notre Systèmes de Gestion de Bases de Données (SGBD)
GitLab	Outils de versionning	Systèmes de suivi des versions d'un logiciel
MapBox	Outil de cartographie	Elle fournit les cartes de l'entreprise pour la traçabilité des livraisons
Visual Studio Code	Outil de développement	Éditeur de texte pour saisir le code
Trello	Groupware	Permet de gérer les tâches

Bitrix	Customer Relationship management (CRM)	Permet de communiquer en ligne via messagerie et appel vidéo, de gérer également les tâches.
Gmail	Application de collaboration	Permet la communication pour l'envoi des mails
Google Docs	Suite bureautique Google	Permet de créer et de modifier des documents en ligne et de travailler en équipes, en temps réel.

III. PRESENTATION DU THEME

Pendant l'année académique 2019-2020 à l'IAI Cameroun, nous avons été dans l'obligation de nous confiner à domicile, étudiants comme membres du personnel de l'institut, suite à la pandémie de la Covid-19 qui venait de surgir à l'échelle mondiale. Cette phase de crise étant brusque et inconnue de tous. C'est ainsi que beaucoup d'autres secteurs ont donc opté des solutions au travers des services en ligne. C'est dans ce domaine même que notre entreprise d'accueil Rapidos Sarl excelle. Rapidos Sarl est spécialisée dans le développement web, les services de voyage pour études et même des vacances à l'étranger, les services d'aide à domicile, et est également prestataire d'un service de livraison. **RapidosEat** est donc ce service de livraison sur lequel nous allons nous appesantir tout au long de nos 3 mois de stage sur le thème **analyse, conception, et réalisation d'une plateforme web et mobile d'achats et de livraison de repas en ligne**.

CONCLUSION

Parvenus au terme de cette phase d'insertion, il était question pour nous de prendre connaissance de l'entreprise **Rapidos Sarl** et de l'ensemble de l'équipe avec laquelle nous serons appelés à coopérer tout au long de notre stage académique, se documenter sur l'entreprise, et présenter la structure et le fonctionnement de celle-ci. Cette prise de contact avec de nouveaux logiciels et langage appris ont fait de notre insertion une phase très passionnante, et bien sûr d'ardeur dans le travail. Nous pouvons dire avec certitude que nous nous sommes bien intégrés et avons pu adhérer aux idées de cette entreprise. Nous nous sentons prêt à aborder la plus grande tâche qui est **Analyse, conception et réalisation de la plateforme web et mobile d'achats et de livraison : cas de RapidoSEat**. N'étant qu'une mise en jeu d'introduction aux autres phases, il nous revient maintenant d'élaborer un document d'analyse à savoir le cahier de charges, qui fera l'objet de la prochaine partie.



CAHIERS DES CHARGES

RESUME

Le cahier de charge est un document projet établi de commun accord entre le maître d’ouvrage et le maître d’œuvre, qui permet de définir de manière exhaustive les différentes fonctionnalités à fournir par la future solution. Ce document présente aussi une estimation du coût des différentes ressources à mettre en place pour réaliser le projet.

APERÇU

INTRODUCTION

- I. CONTEXTE ET JUSTIFICATION DU PROJET 17
- II. CIBLE ET BÉNÉFICIAIRES
- III. OBJECTIFS
- IV. EXPRESSIONS DES BESOINS
- V. ESTIMATION DU COUT DU PROJET
- VI. PLANIFICATION DU PROJET
- VII. CONTRAINTES
- VIII. LIVRABLES

CONCLUSION

INTRODUCTION

Grâce au cahier d'insertion, le plus important a été mis en exergue notamment notre thème s'intitulant : **Analyse, conception, et réalisation d'une plateforme web et mobile d'achat et de livraison de repas à domicile**. Le cahier des charges décrit précisément les besoins auxquels le prestataire doit répondre, et organise la relation entre les différents acteurs tout au long du projet. Il fixe les objectifs à atteindre et décrit en même temps les obligations des deux parties, c'est-à-dire le maître d'ouvrage et le maître d'œuvre. De manière brève, il précise les attentes du client sur le nouveau système d'information, les délais à respecter ainsi que les contraintes sur les outils à utiliser. Dans ce cahier des charges, il sera question pour nous de présenter dans un premier temps le contexte qui permet de comprendre pourquoi cette solution est mise sur pied à travers la justification, de plus énoncer les objectifs à atteindre, ensuite énumérer les différentes fonctionnalités de la future plateforme, et enfin planifier le projet tout en évaluant l'aspect financier des ressources mis en jeu.

I. CONTEXTE ET JUSTIFICATION DU PROJET

1. Contexte

En fin d'année Novembre 2019, est survenue un drame en Chine, la pandémie appelée **coronavirus 2019 (Covid-19)** qui s'est propagée assez rapidement dans plus de 170 autres pays suite aux migrations des populations et, devenant ainsi un phénomène mondial et variant. C'est dans ces pays contaminés que nous comptons malheureusement : le Cameroun. Cette pandémie nous a réduit à appliquer plusieurs mesures barrières telles que limiter les contacts directs entre population en fermant les frontières, créer une distanciation sociale et bien d'autres. De cette pandémie on observe une expansion dans le monde numérique car les populations ont bien plus recours à la technologie et aux services en ligne.

2. Justification

C'est de ce contexte que l'on utilise massivement des plateformes numériques et applications mobiles dans l'optique de solliciter tout service pour éviter le regroupement de plus de 50 personnes. C'est dans cette même lancée que naît l'idée de l'entreprise **Rapidos Sarl** : Créer une plateforme web et mobile d'achat et de livraison de repas à domicile : **RapidosEat**, qui permet aux clients de solliciter désormais ses services à distance. Cette plateforme fédère les partenaires, propose leurs différents repas et produits agro-alimentaires, conseille ces restaurants sur les statistiques par rapport à leurs entrées et sorties, et permet avant tout aux clients de passer des commandes en ligne. **RapidosEat** est conçue pour faciliter la vie des camerounais au travers de la dématérialisation de l'accès au produit, aide les personnes ayant des difficultés de déplacement, et contribue à limiter les risques de propagation du **covid-19**.

II. CIBLE ET BÉNÉFICIAIRES

1. Cible

Notre cible principale est **les restaurants** car la mission de l'application étant d'une part d'offrir des services de livraison de qualité, rapide et à moindre coût et d'autre part de donner un accès plus facile à tous produits sur tout le territoire national du Cameroun de ces dernier.

2. Bénéficiaires

Les bénéficiaires représentent ceux qui vont utiliser l'application, du cadre de notre thème il s'agit des acteurs du système qui sont :

- ❖ Les clients;
- ❖ Les restaurants;
- ❖ Les livreurs;
- ❖ Rapidos SARL.

III. OBJECTIFS

1. Objectif général

L'objectif global est d'apporter un **service de livraison de qualité** sur l'étendue du territoire national.

2. Objectifs spécifiques

De manière explicite, les différents objectifs à atteindre au travers de cette plateforme sont les suivants :

- ❖ Référencer les restaurants partenaires sur la plateforme ;
- ❖ Être intermédiaire entre le client et le restaurant ;
- ❖ Permettre au client de passer une commande ;
- ❖ Permettre au client de suivre sa commande ;
- ❖ Permettre au client d'effectuer un paiement via Orange money/MTN mobile money/Compte bancaire ;
- ❖ Permettre au restaurant de prêter leurs produits ;
- ❖ Permettre à Rapidos SARL d'avoir une global des activités de la plateforme ;

IV. EXPRESSIONS DES BESOINS

Lorsque nous parlons de besoins ici, il s'agit des différentes fonctionnalités liées au système. Il existe ainsi des **besoins fonctionnels et des besoins non-fonctionnels**.

1. Besoins fonctionnels

Ils désignent les différentes actions effectuées par un utilisateur dans le système et retournant un résultat donné. Les besoins fonctionnels du système sont regroupés et subdivisés comme tel :

- Gestion des Menus :

- Afficher tous les menus ;
- Commenter un menu ;

- Noter un menu ;
- Consulter un menu ;
- Masquer un menu ;

- **Gestion du panier :**

- Ajouter au panier ;
- Modifier le panier ;
- Vider le panier ;

- **Gestion de la localisation :**

- Consulter adresse de livraison ;
- Ajouter adresse de livraison ;
- Localiser chauffeur Google Mapp (Map box) ;
- Ajouter des moyens de livraisons ;
- Afficher adresse du chauffeur ;
- Afficher temps de livraison ;
- Afficher information du livreur ;
- Afficher suivi de la commande ;
- Afficher suivi du chauffeur ;

- **Système de paiement :**

- Consulter mode de paiement ;
- Choisir paiement à la livraison ;
- Choisir paiement par Orange money/MTN mobile ;
- Choisir paiement par carte bancaire ;

- **Gestion du profil et paramètres :**

- Changer photo de profil ;
- Modifier son profil ;
- Contacter service ;
- Mettre le Mode nuit/ clair ;

- **Login et inscription :**

- Login/connexion avec Google ou Facebook ;
- Créer son compte ;
- Choisir code des pays ;

- Sélectionner mot de passe oublié ;
- Déconnexion ;

2. Besoins non fonctionnels

Ce sont les besoins en matière de performance, de type matériel ou de conception. Nous distinguons comme besoins non-fonctionnels :

- ❖ **Multilinguisme** : la plateforme doit intégrer plusieurs langues pour la diversité culturelle ;
- ❖ **La facilité de maintenance et de réutilisabilité** : le système de notre application doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation ;
- ❖ **La portabilité** : l'application RapidosEat doit être capable de s'exécuter sur différents environnements comme par exemple sur Ios et Android en mobile, Mac ou linux.
- ❖ **La fiabilité** : l'application doit fonctionner de façon cohérente sans erreur et doit être satisfaisante ;
- ❖ **La sécurité** : la plateforme doit s'assurer de la sécurité des transactions effectuées et des informations personnelles du client donc les piliers sont les suivants;
 - **La disponibilité** : la plateforme doit s'assurer que les utilisateurs ayant un compte ont accès aux informations quand ils le demandent, ou dans le temps requis pour son traitement ;
 - **L'intégrité** : elle doit certifier la présence non modifiée ou non altérée des informations, et de la complétude des processus de traitement ;
 - **La confidentialité** : notre application doit respecter surtout la confidentialité des données du client ;
 - **La preuve ou traçabilité** : la plateforme doit garantir que l'émetteur d'une information soit bien identifié et qu'il a les droits et les accès logiques, et que le récepteur identifié soit bien autorisé à accéder à l'information ;
 - **Authentification** : Chaque activité menée sur la plateforme doit être faite par un utilisateur possédant un compte et connecté sur la plateforme à partir de ses identifiants.
- ❖ **Ergonomie** : le système de la plateforme doit avoir un bon visuel et permettre une facilité à l'utilisateur dans la réalisation des différentes tâches pour lesquelles elle a été conçue ;

V. ESTIMATION DU COUT DU PROJET

Pour la réalisation de ce projet nous devons avoir à notre disposition des ressources matérielles, logicielles, et également humaine.

1. Ressources matérielles

Pour la réalisation de cette plateforme, nous avons besoin de ces matériels:

Tableau 3:Ressources matérielles (mercuriale 2022)

Quantité	Matériels	Fonctionnalité	Prix unitaire	Caractéristiques	Prix total
			(FCFA)		(FCFA)
5	HP Pavillon	Terminal de développement	350 000	-8 Go RAM -carte graphique 4Go de mémoire dédiée -écran 20”	1 750 000
1	ROUTEUR DLINK WIRELESS N300 ADSL 2+MODEM ROUTEUR	Est un équipement matériel informatique qui se charge de l'accès internet	60 000	Peut supporter jusqu'à 20 postes -3 ports RG45 -1 port USB -Bande passante 10Mb/	63 250
2	Desktop DELL	Simulation web avec navigateur	500 000	- RAM : 30Go - HDD 500GO - Processeur 5Hz	1 000 000
3	CLE USB ADATA 32GB	Transport de fichier entre terminaux	20 217	32Go de mémoire	60 651

1	Téléphone Android Redmi note 10	Terminal de déploiement mobile	200 000	- RAM : 4 Go - ROM : 128 Go	200 000
1	Téléphone IOS iPhone 11	Terminal de déploiement mobile	290 000	- RAM : 4 Go - ROM : 225 Go	290 000
Total					3 363 901

2. Ressources logicielles et abonnements

Comme programmes et abonnements dont nous avons besoin pour la réalisation pendant un an on a :

Tableau 4: Ressources logicielles et abonnements (site officiel des produits)

Quantité	Logiciels	Type	Rôles	Prix (FCFA)
5	IntelliJ IDEA	Outil de développement	Est un environnement de développement intégré nous permettant de développer sur les technologies Java (Spring boot)	99 234 / an
5	Postman	Outil de développement	Permet d'interroger les services web avec des appels http directement depuis une interface graphique	Version gratuite
5	Postgresql (pgAdmin)	Outil de développement	Outil d'administration graphique pour Postgresql. Qui joue ici le rôle de notre Systèmes de Gestion de Bases de Données (SGBD)	Open source libéral, gratuit
	GitLab	Outils de versionning	Système de suivi des versions d'un logiciel.	Gratuit en ligne

	Google Map	Outil de cartographie	Elle fournit les cartes de l'entreprise pour la traçabilité des livraisons.	Version gratuite
	Visual Studio Code	Outil de développement	Éditeur de texte pour saisir le code.	Version gratuite
	Trello	Outils de gestion de projet	Permet d'organiser un projet, et de suivre les tâches.	Version gratuite en ligne
	Bitrix	Groupware	Permet de communiquer en ligne via messagerie et appel vidéo, de gérer également les tâches.	Gratuit en ligne
	Gmail	Application de collaboration	Permet la communication pour l'envoi des mails	Gratuit
	Google Docs	Suite bureautique Google	Permet de créer et de modifier des documents en ligne et de travailler en équipes, en temps réel.	Gratuit
5	Abonnement annuel Office 365 Personnel	Suite bureautique	Permet de rédiger les comptes rendus Daily Scrum, rédactions des recrutements, divers...	40 699.22 FCFA/ an
5	Abonnement annuel Kaspersky 2017	Antivirus	Protections contre virus, malwares, installation annuelle des firewalls	45000 / an

1	Location annuelle d'une Instance Amazon RDS for Postgresql (2 cœurs, 2 Go RAM, 250 gigas de Stockage)	SaaS	Serveur de base de données Postgresql	3 205.5 / an
1	Location annuelle d'une Instance A0 du Cloud EC2 AWS (1 cœur, 2 go RAM, 250 giga de Stockage)	SaaS	Service proposé par Amazon permettant à des tiers de louer des serveurs sur lesquels exécuter leurs propres applications web	105 000 / an
1	Abonnement annuel AWS Elastic Beanstalk (1 cœur, 2 Go RAM, 850 Giga de Stockage)	SaaS	service d'orchestration offert par Amazon Web Services pour le déploiement d'applications orchestrant différents services d'AWS	410.000 / an
1	Abonnement annuel App Store	Hébergeur application mobile	Magasin d'applications distribué par Apple sur les appareils mobiles fonctionnant sous Ios	64 350 / an
5	Système d'exploitation Windows 10 pro	Système d'exploitation	Système d'exploitation	13 367.5
Total				1 520 589.1 FCFA

3. Ressources humaines

Il s'agit ici des membres tenant part à la réalisation de ce projet ; en d'autres termes de la main d'œuvre indispensable à la réalisation du projet.

Tableau 5: Ressources humaines (Source : D'après le planning financier de l'entreprise Rapidos)

	Main d'œuvre	Quantité	Temps de travail	Salaire	Total
Scrum master	Chef de projet	1	7 mois	350.000	2.450.000 FCFA
Equipe de développement	Analyste/Concepteur	1	5 mois	250.000 FCFA	1.250.000 FCFA
	Programmeur Frontend	1	6 mois	300.000 FCFA	1.800.000 FCFA
	Programmeur Backend	1	6 mois	300.000 FCFA	1.800.000 FCFA
	Développeur	1	5 mois	200.000 FCFA	1.000.000 FCFA
	Testeur	1	3 mois	200 000	600.000
	Designer	1	2 mois	200 000 FCFA	400.000 FCFA
Total					9.300.000 FCFA

Il advient donc qu'en définitive l'estimation du coût du projet se révèle être comme tel :

Tableau 6:Estimation du coût du projet (Source : Calcul prévisionnel)

Types	Prix
Matériels	3 363 901 FCFA
Logiciels et abonnements	1 520 589.1 FCFA
Humain	9 300 000 FCFA
Total provisoire	14 184 490.1 FCFA
Imprévu (15% prix provisoire)	2 127 673.51 FCFA

Coût hors taxe

16 312 163.6 FCFA

De tous ces périssos le coût total pour le projet est de :

16 312 163.6 FCFA

(Seize millions trois cent douze mille cent soixante-trois et six FCFA)

VI. PLANIFICATION DU PROJET

Il s'agit ici de l'organisation des tâches à réaliser sur une période donnée. Elle peut s'observer sur le **diagramme de Gantt** et le **diagramme de Perte**.

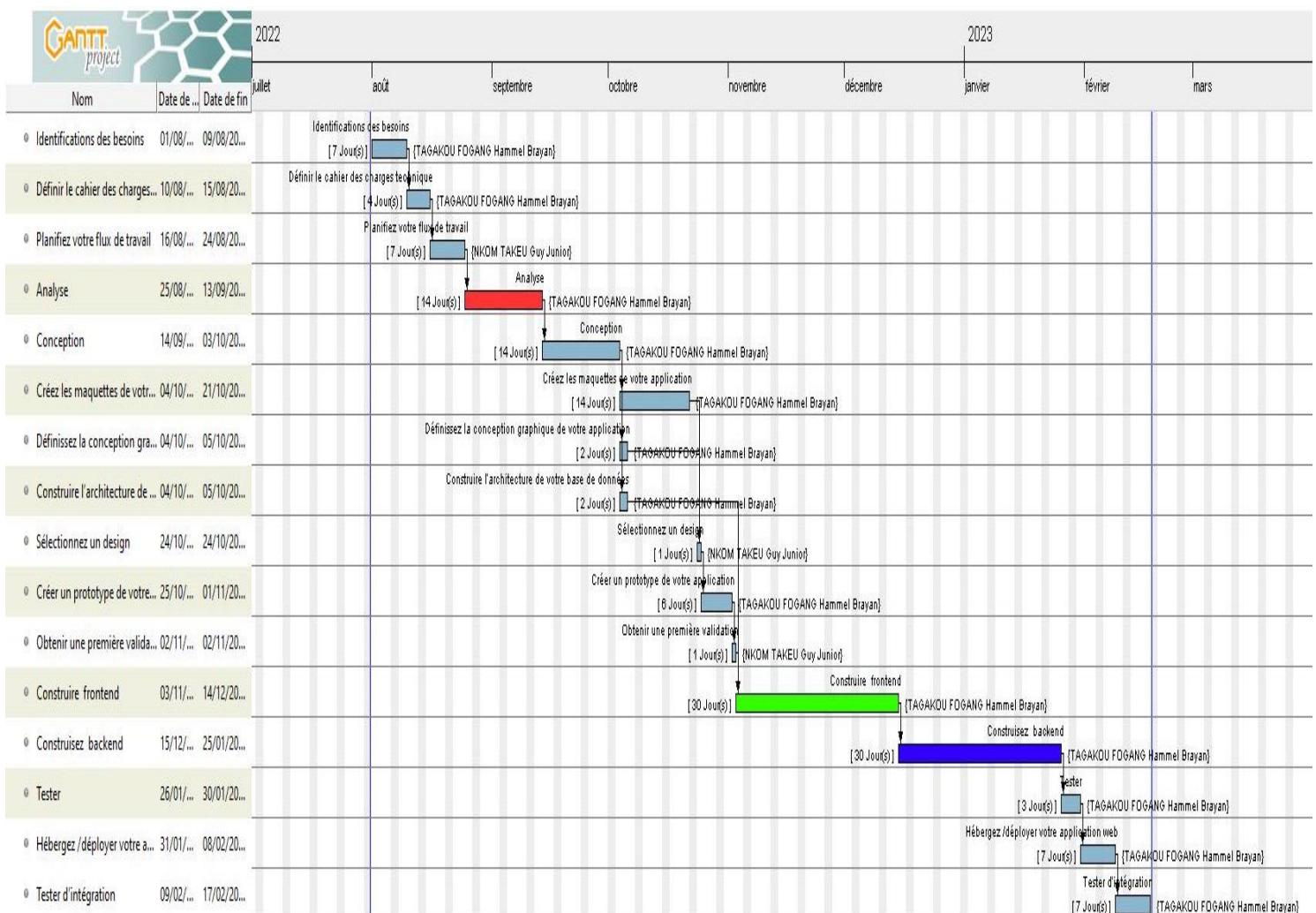


Figure 3:Diagramme de gant du projet

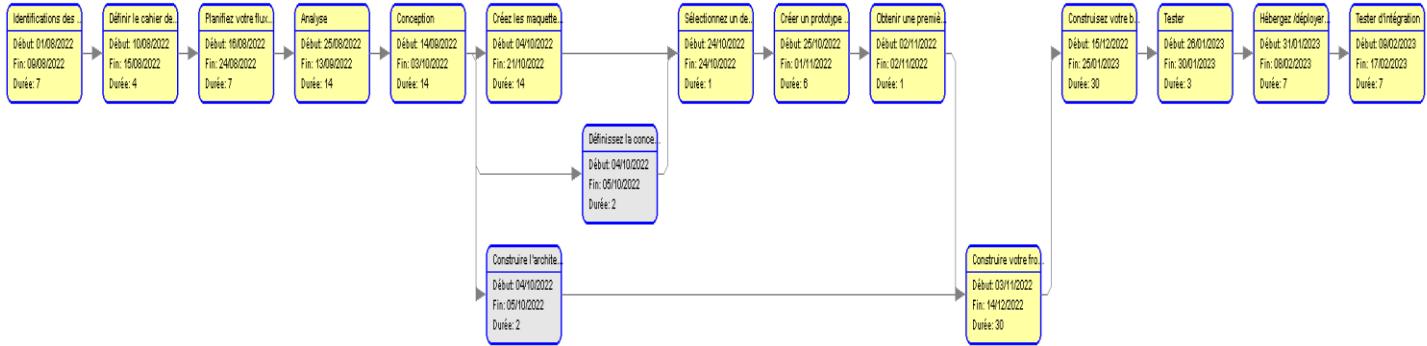


Figure 4:Diagramme de PERT

VII. CONTRAINTES

1. Coût

D'après l'estimation du cout du projet il advient que la réalisation de ce projet en vue des ressources matérielles, logicielles et humaines, s'élève de **16 312 163.6 FCFA**.

2. Délai

D'après le tableau dans « ressources humaines » nous constatons que la réalisation de ce projet va jusqu'à **7 mois** soit **du Vendredi 01 juillet 2022 au Vendredi 03 mars 2023**.

3. Qualité

La plateforme web et mobile offrira rapidité lorsqu'on voudra effectuer toute action dans le système, portabilité de telle manière à ce qu'on puisse l'exécuter sur tout environnement, ergonomie afin qu'on puisse avoir non seulement un design moderne mais également une bonne expérience utilisateur, **stabilité, responsivité et avant tout la sécurité**.

VIII. LIVRABLES

Les éléments qui seront livrés au terme de notre réalisation sont :

- L'application (RAPIDOS, restaurant, client, livreur) ;
- Le code source de notre application ;
- Le dossier d'analyse ;
- Le dossier de conception ;
- Le dossier de réalisation ;
- Le guide d'utilisation de l'application RapidosEat ;

CONCLUSION

Au terme de notre **cahier des charges**, il est visible que nous avons une vue globale mais aussi détaillée de tout ce qu'il faut comme ressources pour la réalisation de ce projet. Ces ressources sont d'ordre **matérielle, logicielle, et humaine**. Toutefois, une estimation du plan financier étant déjà connue, il nous revient maintenant d'approcher la prochaine partie qui est **le dossier d'analyse**



DOSSIER D'ANALYSE

RESUME

Le dossier d'analyse a pour but d'affiner le choix de la méthode à utiliser tout au long du projet. Il permet dans la même lancée de faire comme son nom l'indique une analyse détaillée du projet à mettre en œuvre.

APERÇU

INTRODUCTION

- I. DESCRIPTION DE L'EXISTANT
- II. CRITIQUES ET LIMITES DE L'EXISTANT
- III. PROBLEMATIQUE
- IV. PROPOSITION DE LA SOLUTION
- V. DEMARCHE D'ANALYSE
- VI. CAPTURE DES BESOINS FONCTIONNELS

CONCLUSION

INTRODUCTION

Suite au cahier des charges qui a mis en exergue les détails sur les besoins du futur système, le dossier d'analyse nous apporte une continuité en ceci qui nous permettra de faire une étude de l'existant, faire des critiques et de modéliser la solution. Cette modélisation se fera grâce au langage **UML** version 2.5.1 et suivra les étapes de la méthode de gestion de projet **Agile : Scrum**. La phase d'analyse est considérée comme la fondation pour la réussite d'un projet car bien que la méthode de gestion de projet utilisée soit itérative et incrémentale, la présence d'une erreur dans l'analyse aura des répercussions néfastes dans notre futur système raison pour laquelle, nous avons mis un accent sur l'élaboration minutieuse de cette partie jugée à la fois fragile et délicate ;afin d'éviter les erreurs, minimiser les risques et construire un système qui répondent aux normes des systèmes d'information récents . Par conséquent le squelette de notre dossier d'analyse sera fait autour des diagrammes de cas d'utilisation, séquences, d'activité, communications.

I. DESCRIPTION DE L'EXISTANT

1. L'existant

Le service de livraison à domicile est appliqué depuis de façon routinière dans plusieurs pays étrangers ; mais ici, ce service n'est pas encore tellement diversifié et appliqué à sa juste valeur. Dans le cadre actuel, les livraisons se poursuivent comme suit au Cameroun : soit les **livreurs exerçant en externe ont un pourcentage sur chaque livraison**, soit il existe un **partenariat gagnant-gagnant entre livreur et vendeur** donnant ainsi un pourcentage sur la livraison et la vente du produit. D'autres prestataires de livraison qui ont une application propre à eux, achètent le repas ou produit commandé avant de le faire livrer et récupérer leur dû auprès du client. **Certains restaurants refusent tout partenariat et préfèrent vendre leurs produits** (repas...) de façon physique. Nous avons également des vendeurs qui font dans la restauration et travaillent à leur compte personnel ; ils font des prospections en ligne à travers les réseaux sociaux telles que **Facebook, WhatsApp, Snapchat** afin de cibler leur contact de la toile et élargir le champ de visibilité. Dans ce cas précis, ces vendeurs livrent eux-mêmes ou alors **le client se charge de venir de lui-même récupérer son repas**.

2. L'état de l'art

Les plateformes web et mobile d'achats et de livraison des repas à domicile les plus connus dans ce service dans les pays étrangers sont **Deliveroo, Foodora, Glovo, UberEats...** Au Cameroun, nous avons par exemple **Food hot, Coursier Express, Solex Miam...**

Le tableau suivant nous montre la description et le fonctionnement de quelques-unes de ces applications.

Tableau 7:Etat de l'art

Plateforme	Description	fonctionnement
Glovo 	Entreprise espagnole fondée en 2015 à Barcelone de livraison de repas à domicile via une application mobile dédiée. Il exerce dans neuf pays européens, huit pays africains parmi	Après inscription puis connexion de l'utilisateur, il a accès aux différents restaurants partenaire où il pourra ensuite passer la commande. Une fois le panier rempli, un formulaire est soumis au client afin de renseigner son adresse

	<p>lesquelles la cote d'ivoire et un pays d'Asie.</p> <p>https://glovoapp.com/</p>	<p>de livraison ainsi que le moyen de paiement.</p> <p>Ici la livraison est gratuite mais avec une augmentation de 40% sur les prix des produits initialement remplis par le restaurant ;</p>
Uber Eats 	<p>service de livraison de plats cuisinés lancé par Uber en 2015 et basé à San Francisco, en Californie. Les commandes sont prises via l'application mobile ou le site web d'Uber Eats auprès des restaurants partenaires et sont livrées par des coursiers indépendants (vélo, scooter, etc.).</p> <p>L'entreprise opère dans le monde entier : Amérique du Nord, Amérique du Sud, Europe, Australie, Asie et dans quelques villes d'Afrique</p> <p>https://www.ubereats.com/fr</p>	<p>Les restaurants partenaires sont sélectionnés par la plateforme qui gère le service. Le coût du service est facturé au client final en fonction de la distance de livraison. Par ailleurs la plateforme prend une commission de 30 % en moyenne sur la facture générée par le restaurant partenaire.</p> <p>Les livreurs (en vélo, scooter, automobile, ou autre) employés dans le cadre du service Uber Eats sont rémunérés à la course effectuée et en fonction de la distance de cette dernière.</p> <p>Ce sont des auto-entrepreneurs, parfois soumis à une obligation d'affichage de la marque et de suivi de leur itinéraire</p>
Solex Miam Miam 	<p>Plateforme numérique exerçant au Cameroun qui vous permet de commander et de faire livrer des repas depuis les restaurants de votre région.</p> <p>https://www.solextgv.com/home</p>	<p>Ici, l'entreprise Solex va vers les restaurants et leurs propose des partenariats. Après accord de contrat, le restaurant partenaire est disponible sur la plateforme avec leurs différents produits. Le prix de la livraison est fixe à 1000 FCFA dans la même ville.</p>

II. CRITIQUES ET LIMITES DE L'EXISTANT

Comme différentes zones d'ombre identifiée lors de l'étude de l'existant nous avons :

Tableau 8: critiques et limites de l'existant

Critiques de l'existant	Limites de l'existant
Vue notre étude de l'existant, un client peut passer la commande sur une application tiers comme WhatsApp... avant d'aller récupérer le repas lui-même ou se faire livrer par le restaurant	<ul style="list-style-type: none"> ➤ Non-respect de mesures barrières édictées par le gouvernement camerounais ; ➤ Les clients ne commandent que pendant les heures d'ouvertures des restaurants ; ➤ Les revendeurs de repas peuvent perdre dans la mesure où il pouvait s'agir d'une fausse commande (blague), ce qui est très mauvais financièrement parlant et assez risqué ;
D'après l'état de l'art fait sur l'application Solex Miam Miam, le prix de la livraison est fixe à 1000 FCFA.	<ul style="list-style-type: none"> ➤ Les clients situés près du restaurant paye une somme parfois un peu enlevé vue la distance de livraison ;
Selon l'étude de l'existant, nous avons des prestataires de livraison externe qui sont rémunérer à partir de chaque livraison.	<ul style="list-style-type: none"> ➤ La difficulté pour le livreur à retrouver l'adresse de livraison du client ou à le localiser
D'après l'étude de l'existant, Certains restaurant refuse tout partenariat avec les prestataires externes et préfère gerber transaction et livraison eux même.	<ul style="list-style-type: none"> ➤ Beaucoup de clients sont forcés de se déplacer chaque fois qu'ils veulent acheter un repas. ➤ L'impossibilité pour bon nombre de restaurants d'informer les clients en cas de nouvelles recettes ou repas au menu.

III. PROBLEMATIQUE

Ce rapport de stage s'articule au prisme de la problématique de prévenir et atténuer la propagation de la pandémie covid-19. Il serait donc judicieux pour nous de se poser la question de comment amener la population à limiter le taux de propagation de la pandémie ?

IV. PROPOSITION DE LA SOLUTION

De l'étude de l'existant à sa critique, il en est sorti une problématique donc la solution est une plateforme d'achat et de livraison de repas a domicile.

Cette dite plateforme permettra:

- Aux clients de passer des commandes de gérer leurs compte, repas dans les restaurants de leurs choix, gérer leurs paniers, et effectuer le payement de leurs plateforme ;
- Aux restaurants gérer leurs comptes et produits, recevoir leurs commandes et consulter leurs statistiques de vent ;
- Aux livreurs de gérer leurs compte, recevoir des commandes et de gérer leurs livraisons ;
- A RAPIDOS Sarl de gérer tout l'ensemble du système qu'il possède.

V. DEMARCHE D'ANALYSE

1. Etude Comparative Entre Merise Et UML

Merise est une **méthode d'analyse, de conception et de gestion de projet** informatique.

Cette méthode reste adaptée pour la gestion des projets internes aux organisations, se limitant à un domaine précis.

UML, en français, le Langage de Modélisation Unifié, de l'anglais *Unified Modeling Language (UML)*, est un **langage de modélisation graphique** à base de **pictogrammes** conçu comme une méthode normalisée de **visualisation** dans les domaines du **développement logiciel** et en **conception orientée objet**.

L'UML est une synthèse de langages de modélisation objet antérieurs.

Outre des différences évidentes de natures, buts et d'utilisations (une méthode d'une part et un langage d'autre part), l'aspect graphique de l'ULM est une caractéristique qui le distingue de Merise

Tableau 9:Tableau comparatif entre MERISE et UML

MERISE :	UML :
Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprises est une méthode systémique d'analyse et de conception de systèmes d'information. C'est-	Unified Modeling Language n'est cependant pas une méthode mais plutôt un langage de modélisation objet à qui il faut associer une démarche pour

à-dire qu'elle utilise une approche systémique.	en faire une méthode. c'est le cas de la méthode 2TUP ; RUT et XP.
relationnel	objet
propose de considérer le système réel selon deux points de vue: - Une vue statique (données) - Une vue dynamique (traitements).	propose une approche différente de celle de MERISE en ce sens qu'il associe les données et les traitements. Car avec UML, centraliser les données d'un type et les traitements associés permet de limiter les points de maintenance dans le code et facilite l'accès à l'information en cas d'évolution du logiciel. De plus, UML décrit la dynamique du système d'information comme un ensemble d'opérations attachées aux objets du système.
C'est-à-dire qu'avec la méthode MERISE, nous avons une étude séparée des données et des traitements.	

2. Choix De La Démarche D'analyse

Après cette étude comparative entre les méthodes UML et MERISE, notre choix sera portée sur la méthode **UML**.

UML se définit comme étant un langage de modélisation graphique et textuel destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leurs structures aussi bien que leurs comportements, concevoir des solutions et communiquer des points de vue. L'UML utilise les points forts de ces trois approches pour présenter une méthodologie plus cohérente et plus facile à utiliser. Il représente les meilleures pratiques de création et de documentation des différents aspects de la modélisation des systèmes logiciels et d'entreprise. UML unifie à la fois les notions et les concepts orientés objets. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage. C'est un langage standard de modélisation des systèmes d'informations à objet.

La version D'UML utilisée ici est la version 2.5.1 qui compte treize (13) diagrammes, et une division du système en deux (02) grandes vues :

- La vue statique composée des diagrammes UML structurels qui représente le système physiquement et comporte six (06) diagrammes
 - Diagramme de classe
 - Diagramme de composants

- Diagramme de structures composite
- Diagramme de déploiement
- Diagramme d'objets
- Diagramme de packages
- La vue dynamique constituée des diagrammes UML comportementaux qui représente les interactions effectuées dans le système elle comporte sept (07) diagrammes :
 - Diagramme d'activité
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de séquence
 - Diagramme d'états-transition
 - Diagramme de temps
 - Diagramme de cas d'utilisation

UML étant un langage formel et normalisé, il est un bon support de communication dans l'élaboration d'une solution informatique, sa polyvalence et sa souplesse font de lui un langage universel. Pour développer des solutions, on a besoin de lui associer une méthode générique qui s'attache à ses diagrammes, par exemple la méthode **Agile** il en existe plusieurs : Scrum, XP (EXtreme Programming), RUP (Rational Unified Process), FDD (Feature Driven Development), etc...

3. Etude Comparative Des Processus Unifiés

Un **processus unifiés** est une famille de méthodes de développement de logiciels orientés objets. Elle se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML.

Face à la panoplie de méthodes disponibles, le choix devient assez complexe et difficile. Beaucoup de questions en reviennent toujours à un chef de projet lors du lancement d'un nouveau projet : Comment faire collaborer les différentes équipes ? Comment se rassurer des besoins du client ?

De ce fait nous avons fait une comparaison entre les méthodes de gestion de projet les plus utilisées à savoir 2TUP et Agile Scrum

Tableau 10: Tableau de comparaison processus unifiées, source (<https://www.studocu.com/row/document/universite-de-tunis/genie-logiciel/quiz-5-5-avril-2020-questions/8012662>)

Critères de comparaison	Scrum	RUP	2TUP
Méthodologie	Scrum est une méthode agile dédiée à la ‘gestion de projet’ avec pour principal objectif l’optimisation de la productivité	Rational Unified Process (RUP) est à l’origine du processus unifié et est à ce titre l’implémentation la plus connue. Il s’agit d’un produit et d’une marque de Rational Software, entre-temps acquise par IBM. La méthode est livrée clés en main et est accompagnée d’outils pour guider les équipes dans l’adaptation et exécution du processus.	2TUP est un processus de développement logiciel qui met en œuvre la méthode du processus Unifié. Le 2TUP propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels
Spécification des besoins	La rédaction des spécifications est basée sur l’expression des besoins du client, généralement regroupés et reformulés dans un cahier des charges.	La création : évaluation des risques, architecture, planification. L’élaboration : spécification du besoin, validation, environnement projet La construction : pour chaque	Liste des cas d'utilisation: obtenue à partir du diagramme de contexte dynamique de l’étude préliminaire. Il y a un cas d'utilisation pour chaque

	<p>... Les spécifications fonctionnelles vont donc décrire les aspects métier de l'application, et leur mise en place, leur organisation</p>	<p>production du logiciel et de la documentation support, tests. La transition : test système et utilisateur, correction, déploiement.</p>	<p>message externe entrant. Description des cas d'utilisation: pour chacun des cas d'utilisation trouvé dans le paragraphe précédent, on va documenter ce qui suit</p>
Itération	<p>On constate des variations qui vont d'une semaine jusqu'à 6 semaines. Scrum recommande des itérations (sprints) de 30 jours.</p>	<p>Le développement d'un logiciel se fait souvent de manière itérative jusqu'à conduire à des livrables incrémentaux du système. Chaque itération est considérée comme un livrable et l'on peut revenir sur les phases précédentes et réadapter le logiciel.</p>	<p>Il est nécessaire de désigner les cas d'utilisation pour la première itération. Les autres cas seront traités dans les incrément de développement suivants, selon leur degré d'importance.</p>
Avantages	<ul style="list-style-type: none"> -Compréhension du travail et des tâches à accomplir -Transparence et respect -Deadlines intégrées -Visibilité continue -Focus et flexibilité 	<ul style="list-style-type: none"> - S'assurer de toujours répondre aux besoins des clients en temps et en heure. - Les livrables sont gérés et contrôlés, tout au long du développement. 	<ul style="list-style-type: none"> - Piloté par les cas d'utilisation -Centré sur l'architecture-Toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement,

		-Permet d'utiliser une architecture qui est basée sur les composants. -, Les modifications qui interviennent au cours du projet peuvent être prises en charge et intégrées directement.	
Inconvénients	Peu de documentation écrite. Violation de responsabilité L'équipe ne se prête pas au SCRUM	.Certaines structures peinent à utiliser la méthode RUP, la trouvant moins rapide et adaptable que les méthodes Agile ou SCRUM	L'inconvénient est lorsque les deux branches du cycle en Y vont fusionner. La difficulté va être d'intégrer l'étude fonctionnelle et technique en un seul modèle

4. Choix Du Processus

Après cette étude comparative entre les processus unifiés 2TUP et Agile Scrum, notre choix sera portée sur la méthode de gestion de projet **Agile Scrum**. Scrum ici est d'un atout favorable pour le développement de ce projet car il facilite le développement rapide et efficace des diagrammes UML et ceci en interaction avec les utilisateurs ;

Souvent considéré comme un Framework de gestion de projet **Agile Scrum** est une méthodologie de **gestion des projets** agiles. Il résout ici le problème de périmètre de temps et de coût lors de la gestion de projet. Ici **Scrum** va nous permettre de développer une application. Contrairement aux méthodes non agiles qui consiste à élaborer un plan de travail et expression de besoins définitifs cette méthodologie fonctionne en mode incrémental et itératif cela signifie qu'ici on construit des solutions par itération (Par cas d'utilisation ou par besoin). Pour mieux implémenter cette méthodologie on va se baser sur des boîtes de temps c'est à dire des

événements borner dans le temps (**Sprint**) pour réaliser les cas d'utilisation de notre diagramme de cas d'utilisation UML celui-ci dure généralement 1(un) mois. C'est durant ces différents sprints que nous allons réaliser les incrément de fonctionnalités demandées par les utilisateurs. Un sprint respecte en générale les étapes suivantes :

- **Le Grooming** il s'agit de la phase de la phase d'élaboration des Product backlog qui sont constitués de toutes les demandes d'utilisation et fonctionnalité priorisée.
- **Sprint Planning** qui consiste à la sélection de tous les PBIs (Product backlog items) qui sont les Product backlog les plus demandés par les utilisateurs. Ensuite on décompose ces PBIs en tâches au quelle on accorde une durée en heure
- **Sprint Exécution** : il s'agit de la phase d'exécution des différentes tâches. Et on a une équipe de développement qui réalise les incrément de fonctionnalité. Cette phase est marquée deux évènements majeurs à savoir
 - **Daily Scrum** qui est un petit meeting de moins d'un quart d'heure qui permet de passer en revue ce qui a été développé la veille, ce qui est à développer demain et comment s'y prendre.
 - **Potentially shippable Product increment** il s'agit ici d'émettre des potentiels incrément de fonctionnalité qui sont déjà intégrables dans le projet. C'est à dire possible d'être mise en production. Et qui sera ensuite présenté aux utilisateurs qui donnerons les avis et les accords ou ajoute des nouveau PBIs
- **Sprint Review** Il s'agit ici de faire passer en revue les réalisations d'itération avec l'utilisateur. Soit ajouter supprimer ou améliorer les fonctionnalités du système. Cette étape dure les uns (01) à quatre (04) heures.
- **Sprint Rétrospective** il passe en revue les éléments de la méthodologie Scrum dans l'entreprise. Il s'agit ici plus précisément d'une revue méthodologique. Cette étape dure les uns (01) à quatre (04) heures.

La méthode Scrum tout comme tout autre méthode fait intervenir des acteurs pour garantir son bon fonctionnement lors de la réalisation du projet. Les principaux ici sont les suivants :

- **Les Product Owners** travaillent en étroite collaboration avec le business et l'équipe pour s'assurer que tous savent en quoi consistent les tâches du backlog produit ; Sa

priorité est de s'assurer que l'équipe de développement offre un maximum de valeur ajoutée au business.

- **Le Scrum master** : Il coach l'équipe, les Product Owners et le business sur le processus Scrum et cherchent les façons d'affiner leur pratique en la matière
- **L'équipe de développement Scrum** : il s'agit ici du cœur de la méthode Scrum. Ce sont des équipes qui abattent le travail et font partie du développement du projet.

Le Schéma suivant illustre de façon complète et détaillé le fondement des méthodes Scrum :

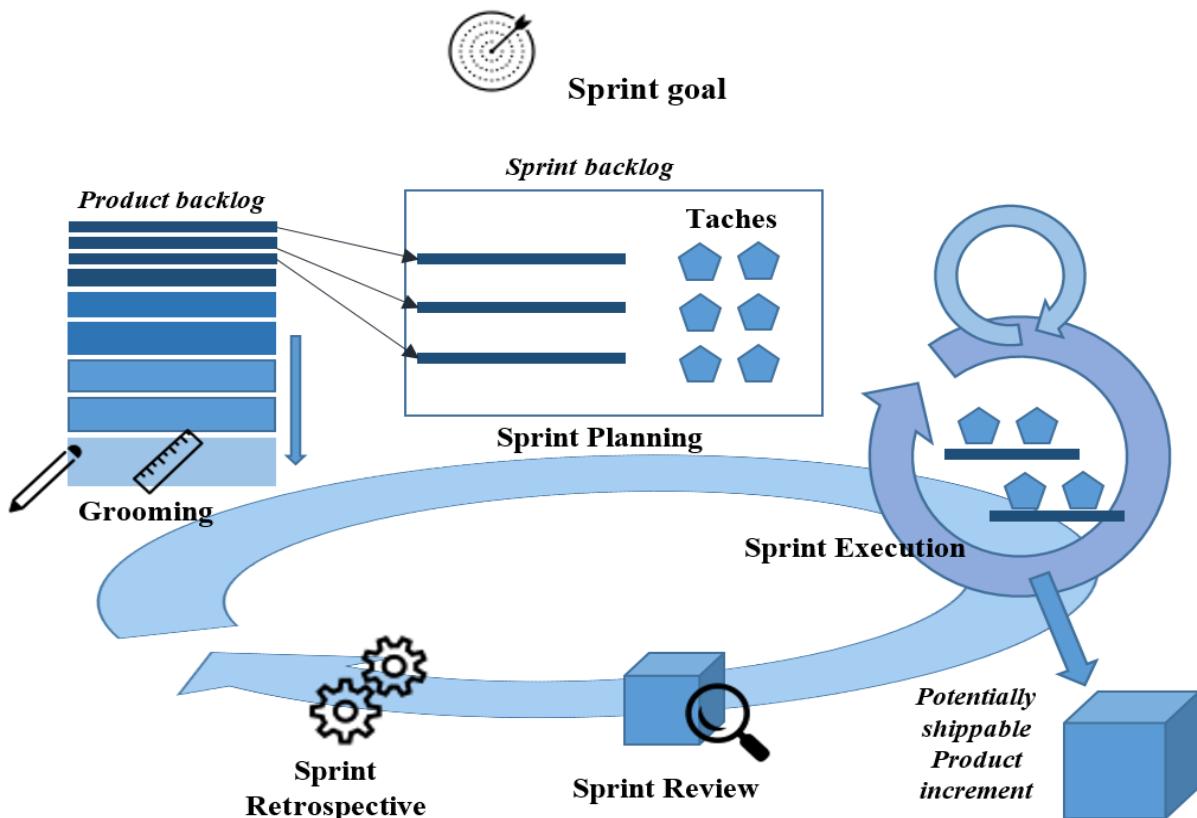


Figure 5: Sprint de la méthodologie Scrum

Plusieurs logiciels nous permettent d'implémenter la méthode Agile Scrum. Notre Choix a été porté sur le logiciel **JIRA**. **Jira Software** fait partie d'une gamme de produits conçus pour aider les équipes de tous types à gérer leur travail. Il fournit des outils de planification et de feuille de route pour permettre aux équipes de gérer les parties prenantes, les budgets et les exigences des fonctionnalités dès le démarrage du projet. L'avantage que ce logiciel propose est que nous pouvons configurer Jira Software de sorte qu'il s'adapte à tout type de projet. Les équipes peuvent se lancer avec un modèle de projet ou créer leur propre workflow personnalisé.

Les tickets Jira, également appelés « tâches », suivent chaque travail qui doit passer par les différentes étapes du workflow jusqu'à son achèvement.

La figure suivante représente une capture d'écran des sprints et backlog du projet Rapido Eat

The screenshot shows a Jira Software interface for the project 'rapidos-eat'. On the left, a sidebar menu includes 'PLANNIFICATION' (Feuille de route, Backlog, Tableau), 'DÉVELOPPEMENT' (Code, Pages de projet, Ajouter un raccourci, Paramètres du projet), and a note about being part of a team. The main area displays 'Tableau Sprint 1' with three columns: 'A FAIRE 2 TICKETS' (Realisation: RE-20), 'EN COURS 1 TICKET' (Conception: RE-19), and 'FINI 2 TICKETS' (Identification des besoins: RE-1, Analyse: RE-1B). A 'Quickstart' sidebar on the right provides links to create a project, personalize the board, and create a ticket, along with a 'Me montrer' button and a note about tickets being individual work items assigned to team members.

Figure 6:Sprint 1 du projet Rapido Eat

VI. CAPTURE DES BESOINS FONCTIONNELS

L'usage de l'outil de modélisation **Astah community version 6.6.4** qui est un AGL (Atelier de Génie Logiciel) nous a permis de modéliser les différents diagrammes nécessaires pour le développement de notre solution.

A. Diagramme de cas d'utilisation

Les **diagrammes de cas d'utilisation** identifient les fonctionnalités fournies par le système, ces fonctionnalités sont appelées ici **cas d'utilisations**, les utilisateurs qui interagissent avec le système (**acteurs**), et les interactions entre ces derniers. C'est une description de l'ensemble des opérations que l'utilisateur pourra effectuer dans le système.

1. Formalisme

Les cas d'utilisations sont représentés par une ellipse contenant le nom du cas d'utilisation. Un acteur et un cas d'utilisation sont mis en relation par une association représentée par **une ligne**.

Tableau 11: Formalisme du diagramme de cas d'utilisation (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Composant du diagramme	Représentation Graphique
Acteur	
Cas d'utilisation	
Association	

Trois types de relations sont pris en charge par la norme UML. Nous pouvons les distinguer dans le tableau ci-dessous :

Tableau 12: Relations du diagramme de cas d'utilisation (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Type de relation	Description	Représentation graphique
Inclusion	Un cas 1 est un inclus dans un cas 2 si l'exécution du cas 1 passe obligatoirement par l'exécution du cas 2	
Extension	On dit qu'un cas 1 étend un autre cas 2 si la réalisation du cas 1 ajoute un ou plusieurs comportements à la réalisation du cas 2.	
Généralisation (héritage)	Un cas 1 hérite d'un cas 2 si le cas 1 en plus de réaliser complètement les actions du cas 2 réalise ou non d'autres nouvelles actions	

2. Identifications des acteurs

Les différents acteurs que nous avons identifiés dans le système sont les suivants :

- Le client ;
- Le Partenaire (Restaurant, supermarché, etc.) ;
- Livreur ;

- API externe ;
- Administrateur (RapidoEat).

3. Quelques cas d'utilisation du système

a. Diagramme de cas d'utilisation global du système

Le diagramme suivant est celui du cas d'utilisation du système faisant intervenir tous les acteurs.

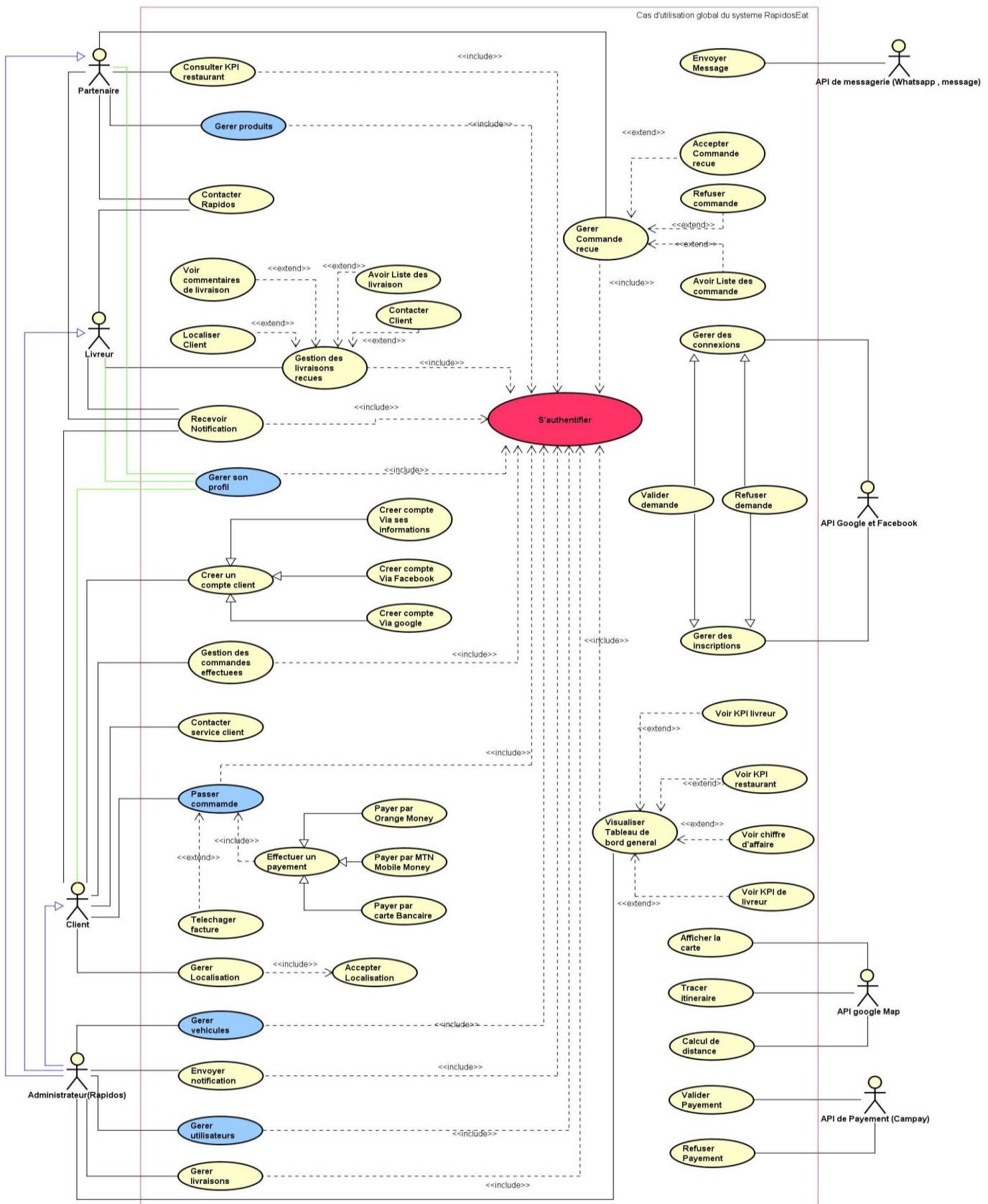


Figure 7:Diagramme de cas d'utilisation global du système

b. Diagramme du cas d'utilisation commander repas

L'activité principale du client est de commander un repas, le diagramme suivant nous donne des détails sur le cas d'utilisation.

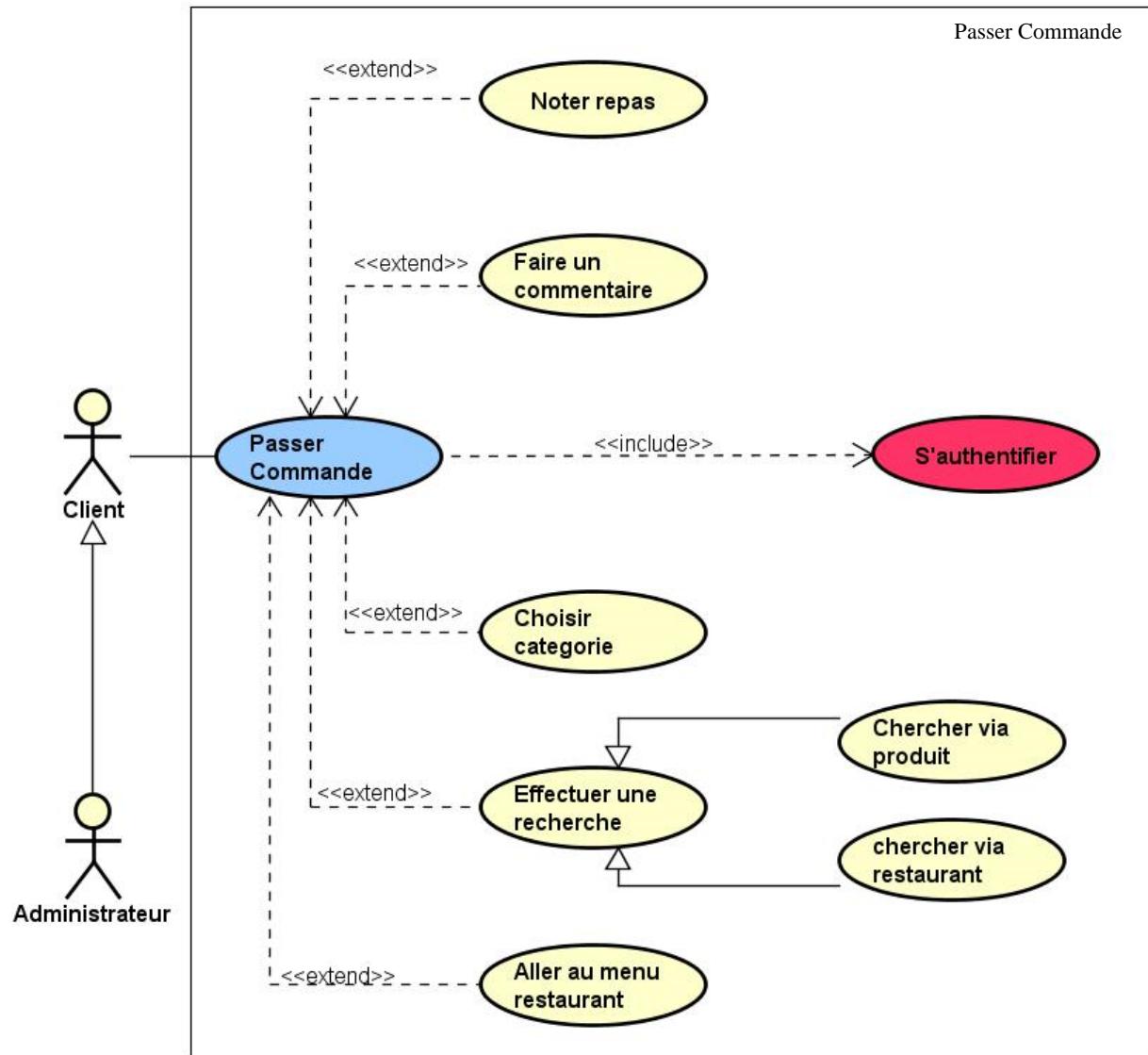


Figure 8:Diagramme du cas d'utilisation commander repas

c. Diagramme du cas d'utilisation Gérer Produit

Le partenaire a la possibilité de manager ses produits sur la plateforme et qui sera ensuite disponible chez le client.

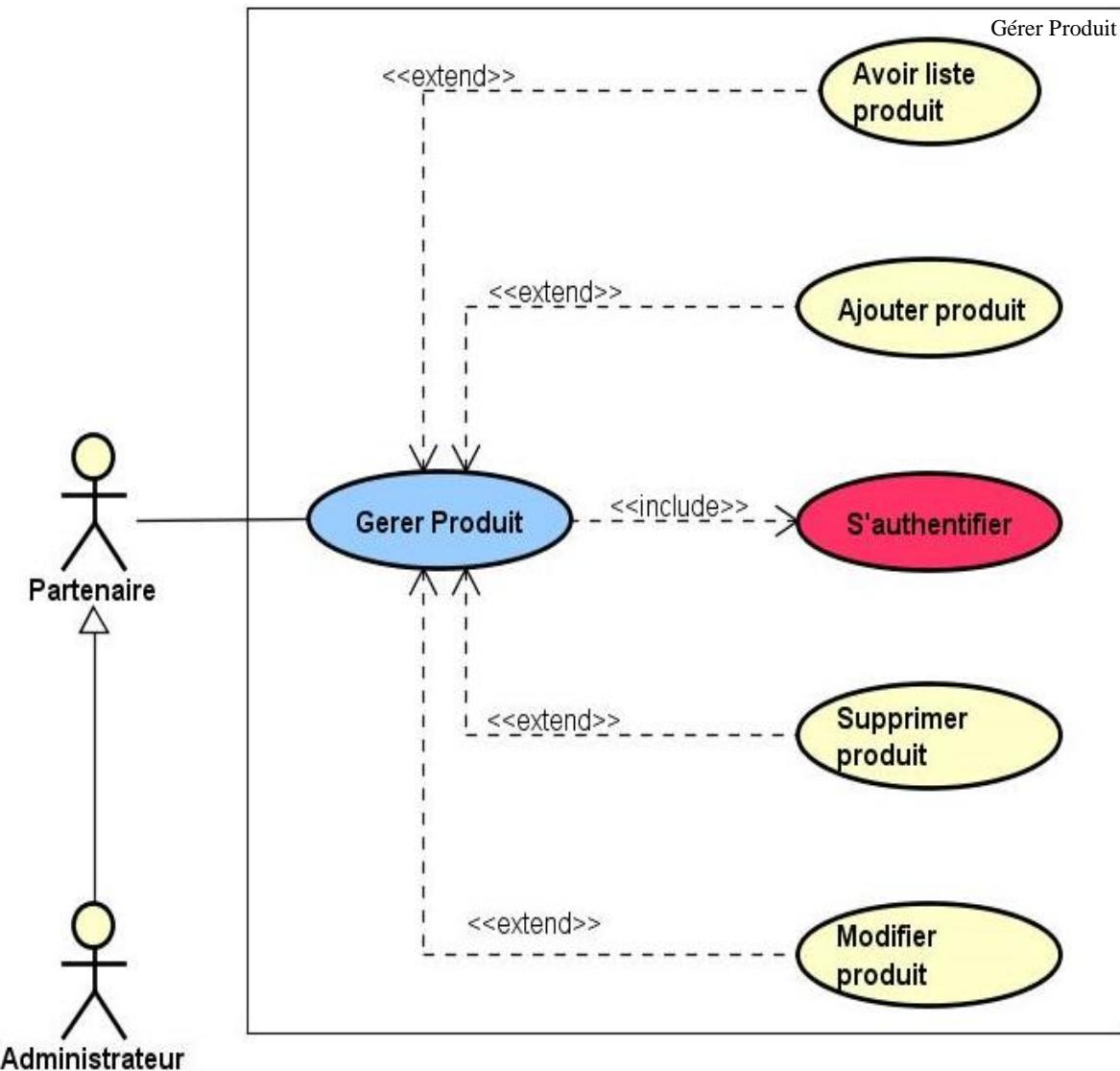


Figure 9:Diagramme du cas d'utilisation Gérer Produit

d. Diagramme du cas d'utilisation gérer utilisateur

L'administrateur à savoir Rapidos aura un total contrôle sur les utilisateurs de l'application à savoir les clients, les partenaires et même les livreurs.

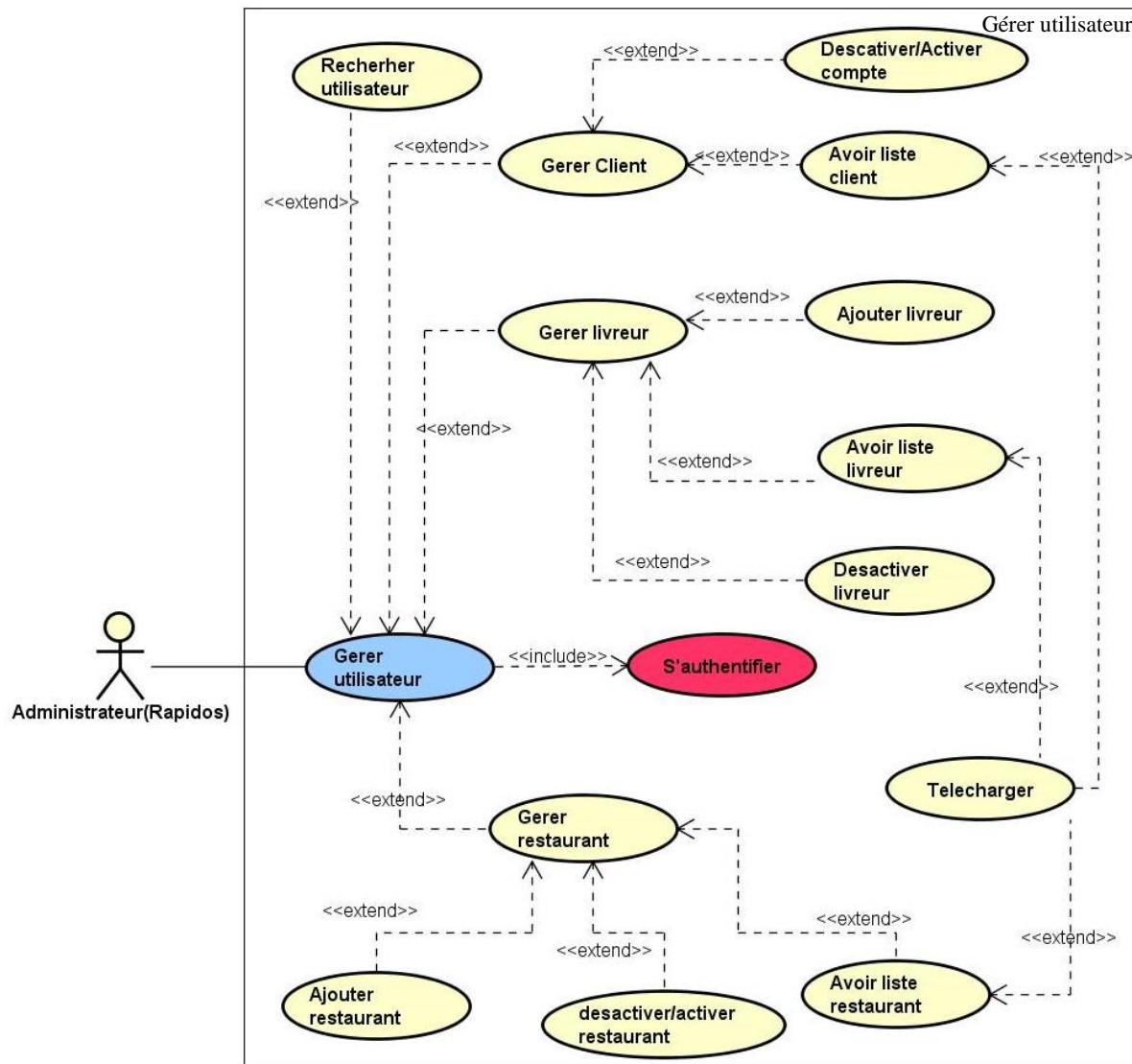


Figure 10:Diagramme du cas d'utilisation gérer utilisateur

e. Diagramme du cas d'utilisation Gestion des véhicules

Dans ce diagramme, nous présentons les différentes opérations de gestion d'un véhicule effectué par le l'administrateur.

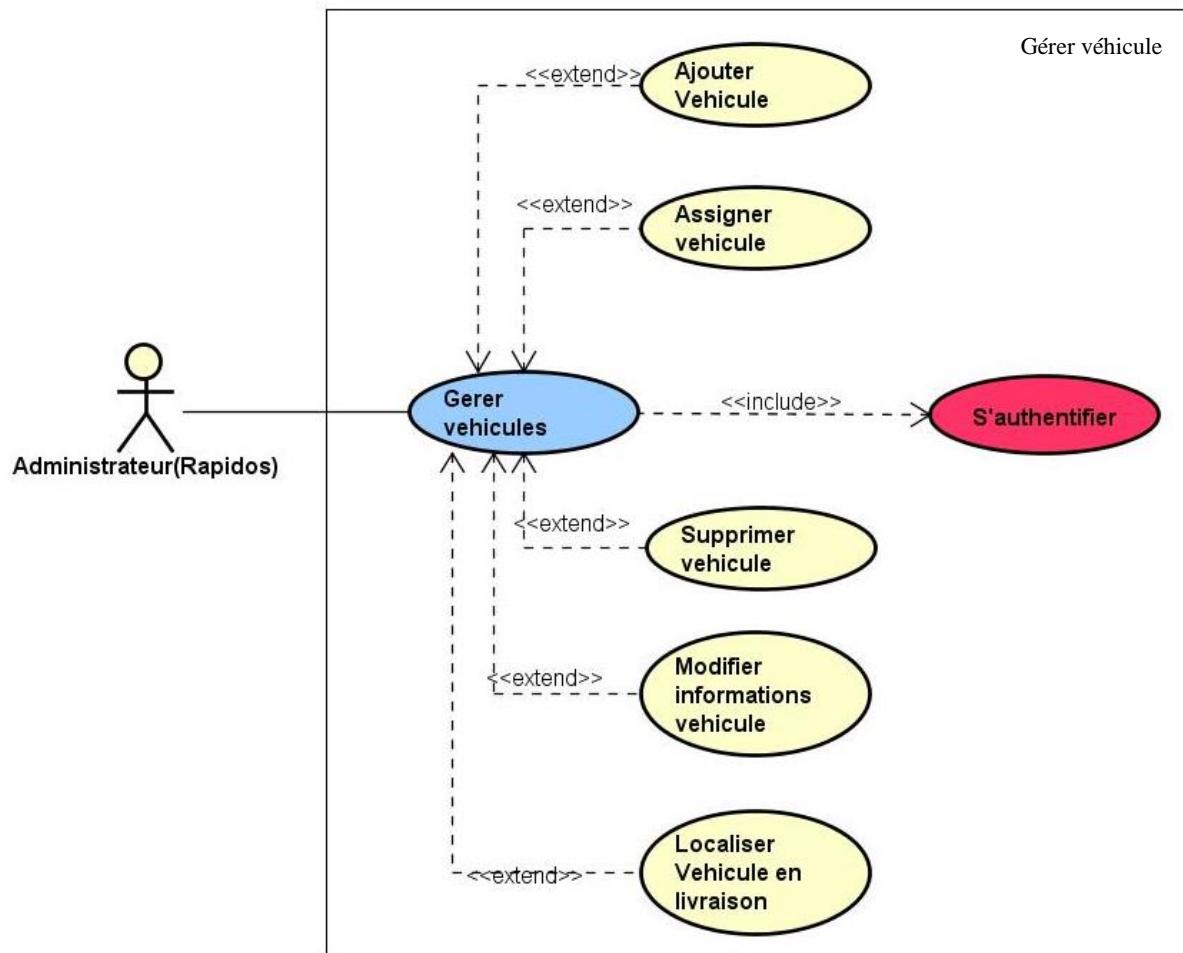


Figure 11:Diagramme du cas d'utilisation Gestion des véhicules

f. Diagramme du cas d'utilisation gérer son profil

Une fois authentifié chaque utilisateur du système aura la possibilité de changer ses informations personnelles.

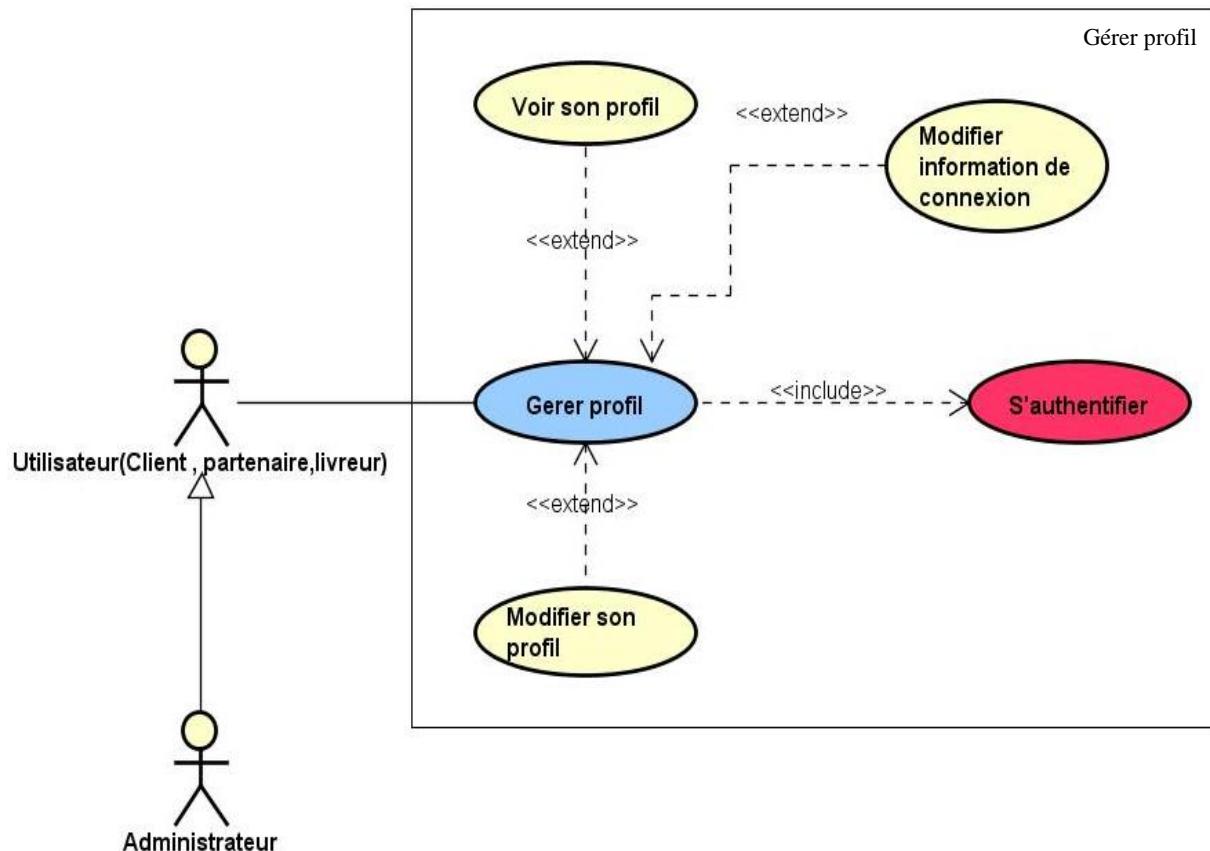


Figure 12:Diagramme du cas d'utilisation gérer son profil

B. Description textuelles de quelques cas d'utilisation

La description textuelle va permettre de préciser un point très important : il y a plusieurs manières d'exécuter le même cas d'utilisation. Un scénario est un mode d'exécution du cas d'utilisation. Voici la description textuelle de quelques cas d'utilisation de notre système:

1. S'inscrire

Titre : S'inscrire

• **Résumé :** ce cas d'utilisation permet aux utilisateurs de s'enregistrer à la plateforme

• **Acteurs :**

-acteur principal : Les clients

• **Date de création :** 15 aout 2022

• **Responsable :** TAGAKOU FOGANG Hammel Brayan

• **Version :** 1.0

• **Préconditions :** le client doit être sur la plateforme

• **Scénario nominal:**

1. l'utilisateur clique sur inscription
2. le système front-end affiche le formulaire d'inscription
3. l'utilisateur saisit ses informations
4. le système front end vérifie la conformité des champs
5. le système front end envoie la requête http POST (information de l'utilisateur) au contrôleur Backend
6. contrôleur Backend Vérifie existence
7. le contrôleur Backend envoie les données à la BD
8. la Bd renvoie le résultat de la requête
9. le contrôleur Backend envoie la réponse http
10. Le système front-end affiche message de succès

• **Scénario alternatif :**

4.a) à l'étape 4 du scénario nominal, l'utilisateur entre des informations non correspondantes ou manquantes

4.b) le système affiche un message d'erreur puis retour à l'étape 2 du scénario nominal.

6.a)A l'étape 6 du scénario nominal, l'utilisateur existe déjà

6.b) le système affiche un message d'erreur puis retour à l'étape 2 du scénario nominal.

• **Scénario d'exception :**

- 3.a) l'utilisateur saisit des identifiants erronés ou manquants à plus de trois reprises
- 3.b) le système part en veille pour 15 secondes puis retour à l'étape 2 du scénario nominal.

Nominal.

- **Post condition de succès :** l'utilisateur a accès à la page de connexion
- **Post condition d'échec :** l'utilisateur n'a pas accès à la plateforme
- **Exigence non fonctionnelle :** la saisie du mot de passe ne doit pas être visible à l'écran

2. Effectuer paiement

- **Titre :** Effectuer paiement
- **Résumé :** ce cas d'utilisation permet aux utilisateurs de payer les produits de son panier
- **Acteurs :**
 - acteur principal :** Les clients
 - acteur secondaire:** API de paiement
- **Date de création :** 15 aout 2022
- **Responsable :** TAGAKOU FOGANG Hammel Brayan
- **Version :** 1.0
- **Préconditions :**
 - Le client s'est authentifié ;
 - Le client possède un panier non vide ;
 - Le client a déjà renseigné son adresse de livraison
- **Scénario nominal:**
 1. Le client valide son adresse de livraison ;
 2. Le système fait une requête de demande de token
 3. Le système front-end affiche la page de paiement
 4. Le client entre le numéro de téléphone
 5. Le système front-end vérifie le numéro de téléphone ;
 6. Le système front-end envoie requête http de paiement
 7. API de paiement envoie la réponse de payement ;
 8. Envoie une requête de payement au contrôleur Backend
 9. Le contrôleur Backend envoie les données de payement à la base de données Postgre
 10. La base de données envoie la réponse de la requête SQL
 11. Le contrôleur Backend envoie la réponse http au service front-end

12. Le système front-end envoie message au client

• **Scénario alternatif :**

- 4.a) à l'étape 4 du scénario nominal, l'utilisateur entre un numéro de téléphone erroné ;
- 4. b) le système affiche un message d'erreur puis retour à l'étape 3 du scénario nominal ;
- 7.a) L'étape 7 du scenario nominal, la réponse de l'API de payement est : Payement FAILED

7.b) Le système front-end affiche message d'erreur a l'utilisateur

• **Scénario d'exception :**

- 2.a) l'utilisateur n'est pas connectée à l'internet
- 2.b) le système front-end envoie message d'erreur

• **Post condition de succès :** l'utilisateur a la page de récapitulatif de la commande

• **Post condition d'échec :** l'utilisateur n'effectue pas la commande et sera pas livre

3. Modifier information du compte

• **Titre :** Modifier information du compte

• **Résumé :** ce cas d'utilisation permet aux utilisateurs de modifier leurs informations sur la plateforme

• **Acteurs :**

- **acteur principal :** L'utilisateur

• **Date de création :** 15 aout 2022

• **Responsable :** TAGAKOU FOGANG Hammel Brayan

• **Version :** 1.0

• **Préconditions :** Etre Authentifier sur la plateforme

• **Scénario nominal:**

1. L'utilisateur clique sur modifier son compte
2. Le Système front end envoie le formulaire de modification
3. L'utilisateur remplit le formulaire
4. Le système front-end vérifie la conformité des champs
5. Le système front-end envoie la requête http de modification au contrôleur back-end
6. Le contrôleur Backend envoie une requête SQL de modification à la Base de données Postgre SQL
7. La base de données envoie la réponse de la requête SQL

8. Le contrôleur Backend envoie la réponse de la requête http
9. Le système front end envoie un message de succès à l'utilisateur

- Scénario alternatif :

4.a) A l'étape 4 du scénario nominal, l'utilisateur entre des informations non correspondantes ou manquantes

4.b) le système affiche un message d'erreur puis retour à l'étape 2 du scénario nominal

- Scénario d'exception :

3.a) l'utilisateur saisit des identifiants erronés ou manquants à plus de trois reprises

3.b) le système part en veille pour 15 secondes puis retour à l'étape 2 du scénario nominal.

- **Post condition de succès :** les informations de l'utilisateur modifie.

- **Post condition d'échec :** les informations de l'utilisateur ne sont pas changées.

- **Exigence non fonctionnelle :** la saisie du mot de passe ne doit pas être visible à l'écran

C. Diagramme de séquence

Les diagrammes de séquences sont des représentations graphiques des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Le diagramme de séquences permet de décrire les interactions entre les objets au sein d'un diagramme des cas d'utilisation.

1. Formalisme

Le diagramme de séquence énumère les objets horizontalement et le temps verticalement. Il modélise l'exécution des différents modèles en fonction du temps. Dans ce diagramme les objets et les acteurs sont énumérés en colonnes avec leur ligne de vie verticale indiquant la durée de vie de l'objet.

Tableau 13: Formalisme du diagramme de séquence (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Composant	Description	Représentation
Objet	Se décrit comme étant l'instance d'une classe	

Acteur	Personne qui interagit et communique avec le système	
Ligne de vie	A ligne de vie représente le déroulement temporel d'un processus	
Activation	Il s'agit ici du temps nécessaire pour qu'un objet ou un acteur accomplisse une tâche, elle indique quand l'objet effectue une action.	
Message (indique les communications entre les objets)	<p>Les messages Synchrones : La réception d'un message synchrone doit provoquer chez le destinataire le lancement d'une de ses méthodes ou opérations. L'expéditeur du message reste bloqué pendant toute l'exécution de la méthode et attend donc la fin de celle-ci avant de pouvoir lancer un nouveau message</p> <p>Les messages asynchrones : ils ne nécessitent pas de réponse du destinataire, avant que l'expéditeur ne continue.</p> <p>Les messages de retour : ils représentent la réponse à un message synchrone</p>	

Exemple de diagramme de séquence :

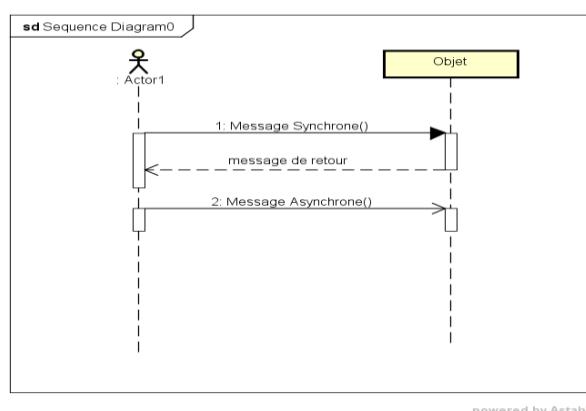


Figure 13: Exemple de diagramme de séquence

2. Quelques diagrammes de séquence

i. Authentification avec JWT token

Ce diagramme présente de façon séquentielle le déroulement du processus d'authentification avec deux **tokens** pour sécuriser les requêtes transitant dans le système.

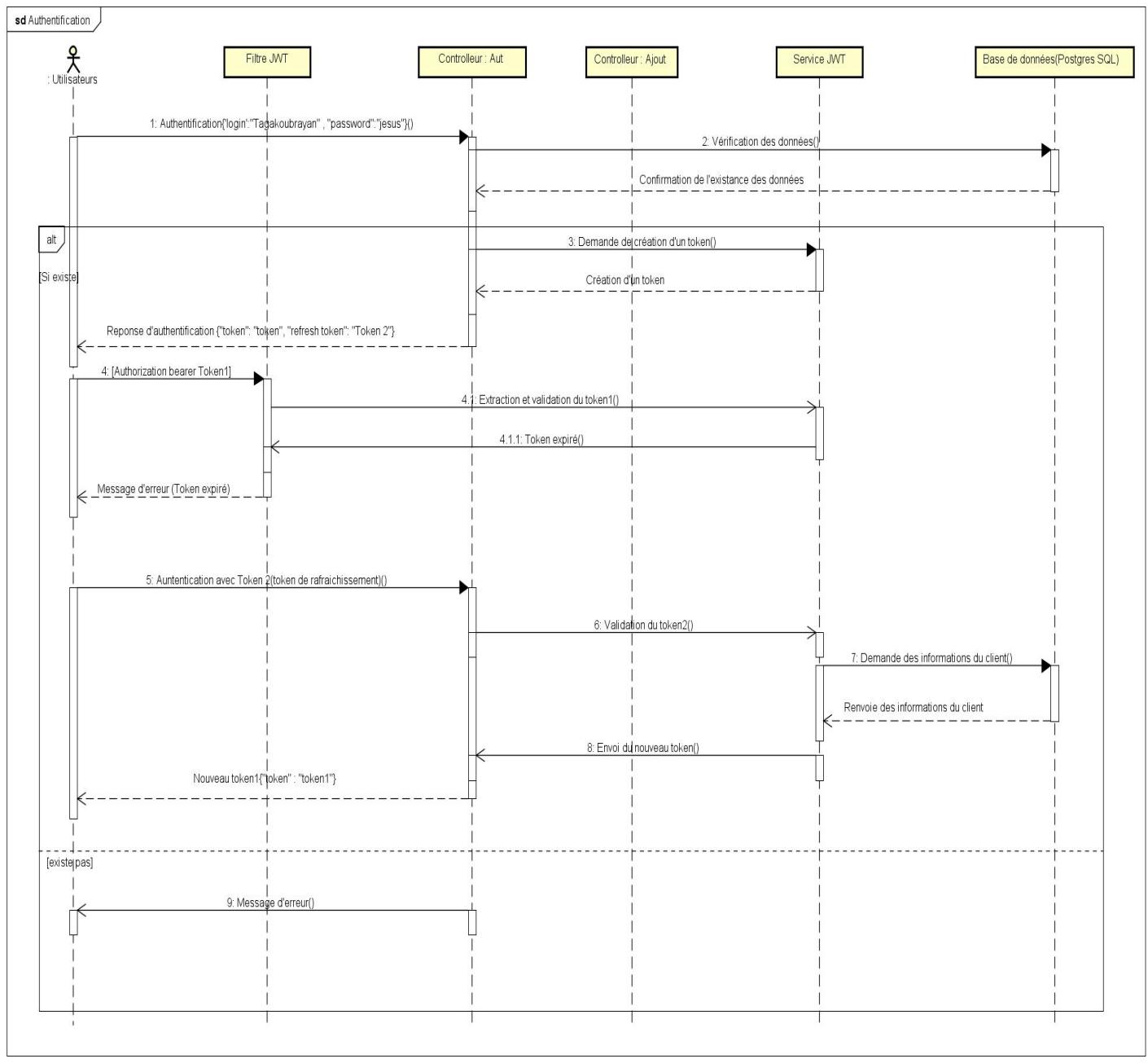


Figure 14:Diagramme de séquence de l'Authentification avec JWT token

ii. Ajouter un partenaire

Le diagramme suivant montre le processus d'ajout d'un partenaire effectué par un administrateur Rapidos.

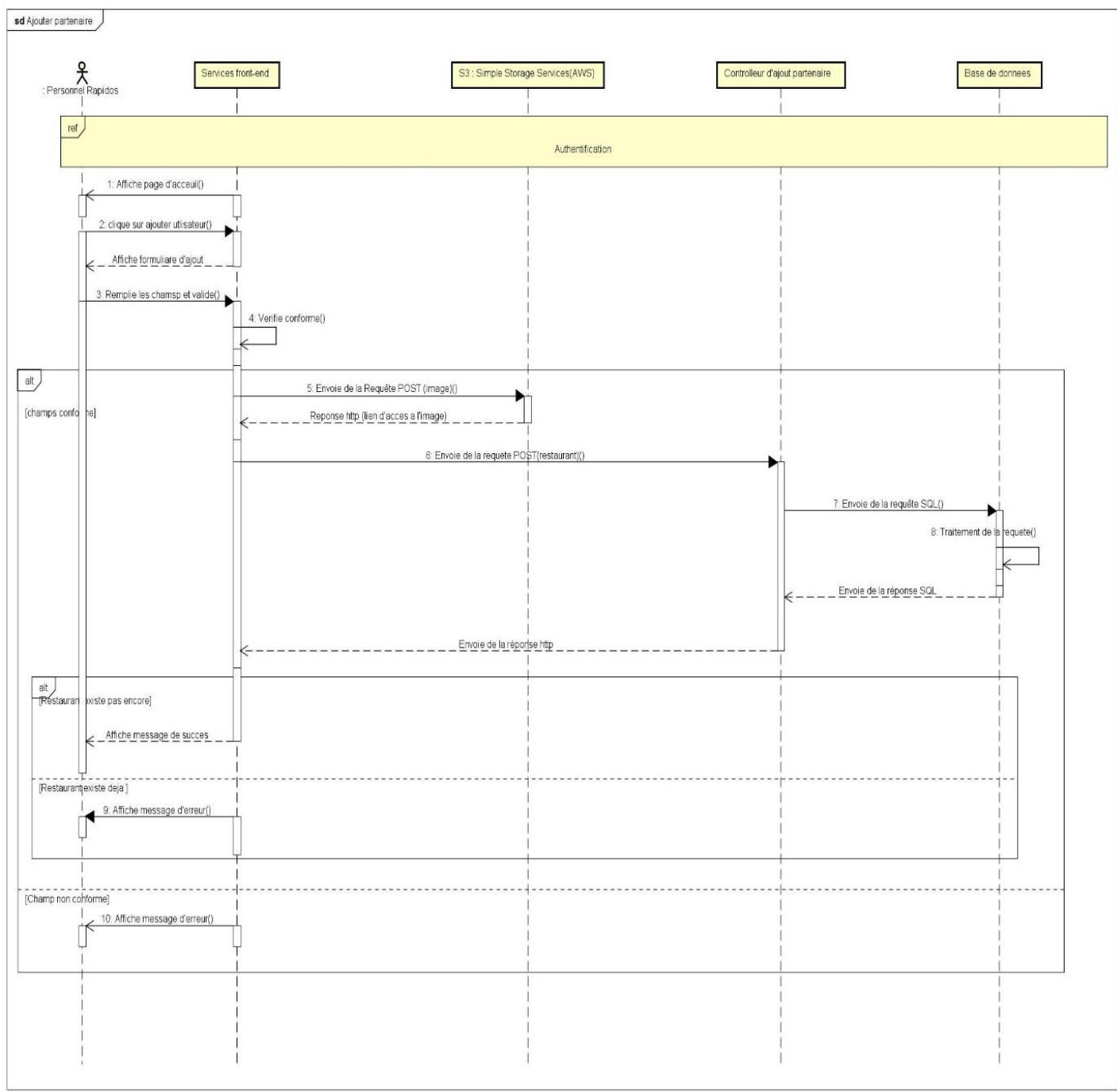
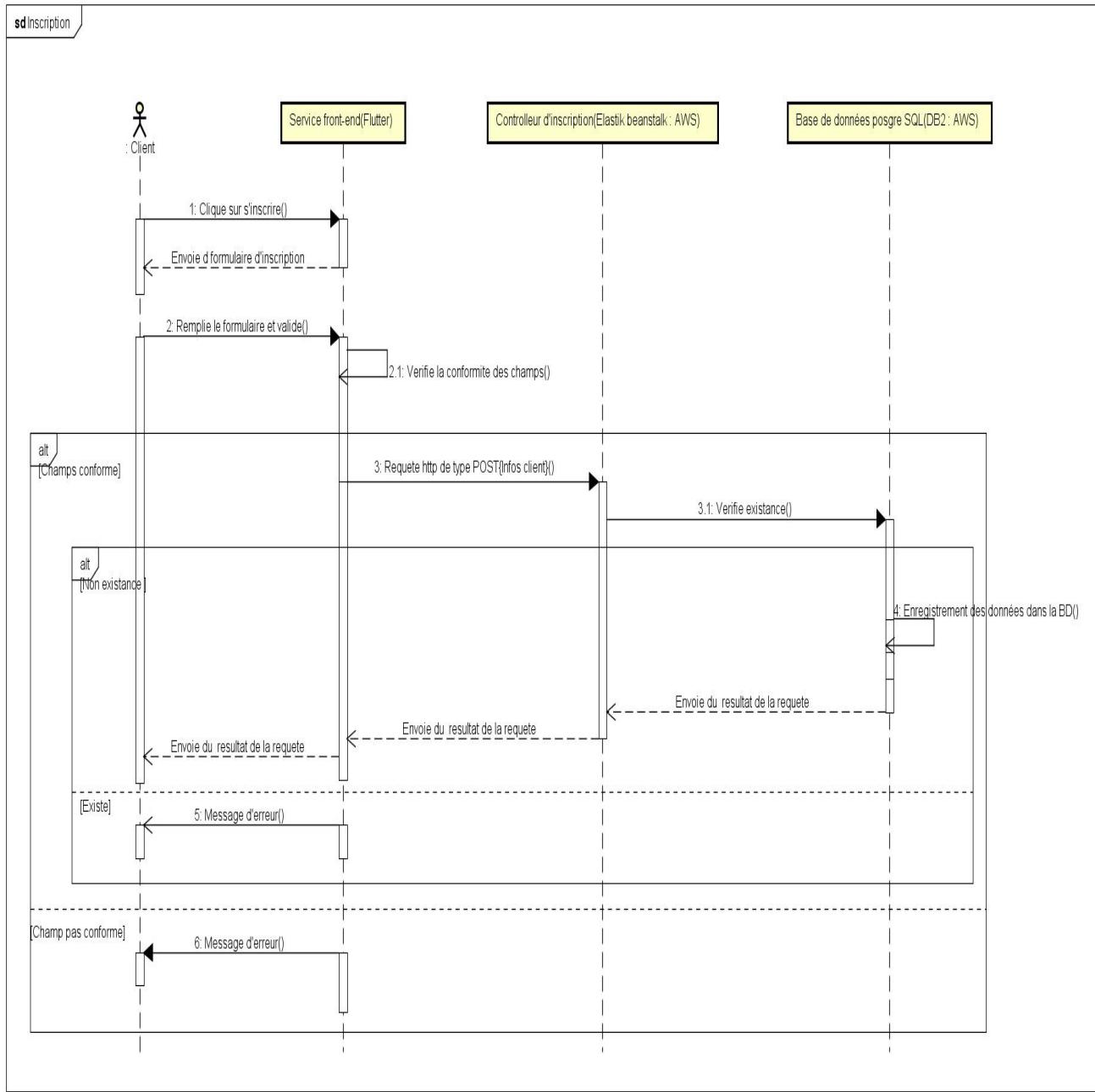


Figure 15:Diagramme de séquence Ajouter un partenaire

iii. Inscription

Ce diagramme montre de manière séquentielle la façon la plus simple et pratique pour le client de s'inscrire à la plateforme **RapidoEat**.



powered by Astah

Figure 16:Diagramme de séquence d'inscription

D. Diagramme d'activité

Les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation.

1. Formalisme

L'usage général du diagramme d'activité permet de mettre l'accent sur les traitements et de formaliser graphiquement la séquence d'actions réalisées dans un cas d'utilisation

Tableau 14: Formalisme diagramme d'activité (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Composant	Description	Représentation
Une action	Rectangle aux coins arrondis qui représente une opération élémentaire et instantanée	
Une transition	Ligne qui représente le passage d'une activité à une autre	
Nœud initial	Cercle plein à partir du quelle un flot débute dans un diagramme d'activité	
Nœud final	Un nœud de fin d'activité : un petit cercle plein marquant l'arrêt l'exécution de l'activité enveloppante s'arrête	
	Un nœud de fin de flot : cercle vide barré d'une croix marquant l'arrêt d'un flot à l'intérieur du diagramme	
Couloir	Les diagrammes d'activités font intervenir les acteurs de chaque activité. Chaque activité sera placée dans une colonne (couloir) qui correspond à l'acteur qui l'effectue	
Barre de synchronisation	Barre horizontale sur laquelle convergent les activités qui sont exécutées en parallèle.	

Exemple de diagramme d'activité

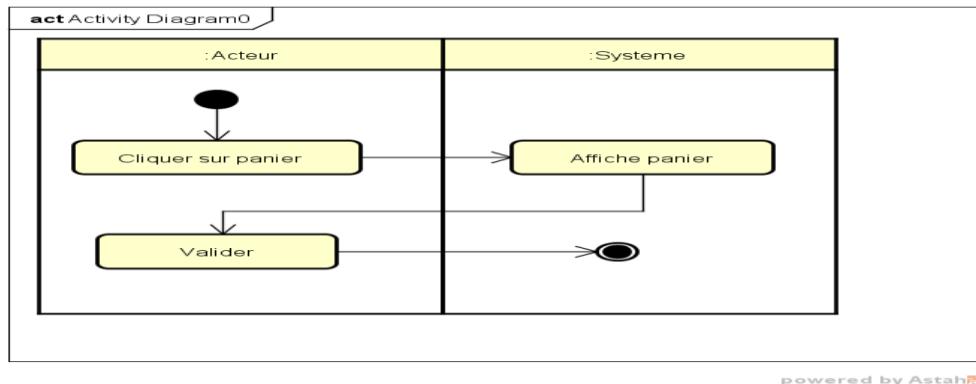


Figure 17: Exemple diagramme d'activité

2. Quelques diagrammes d'activité

i. Modifier son compte(Partenaire).

Ce diagramme présente de manière séquentielle le déroulement de la modification des informations d'un compte partenaire.

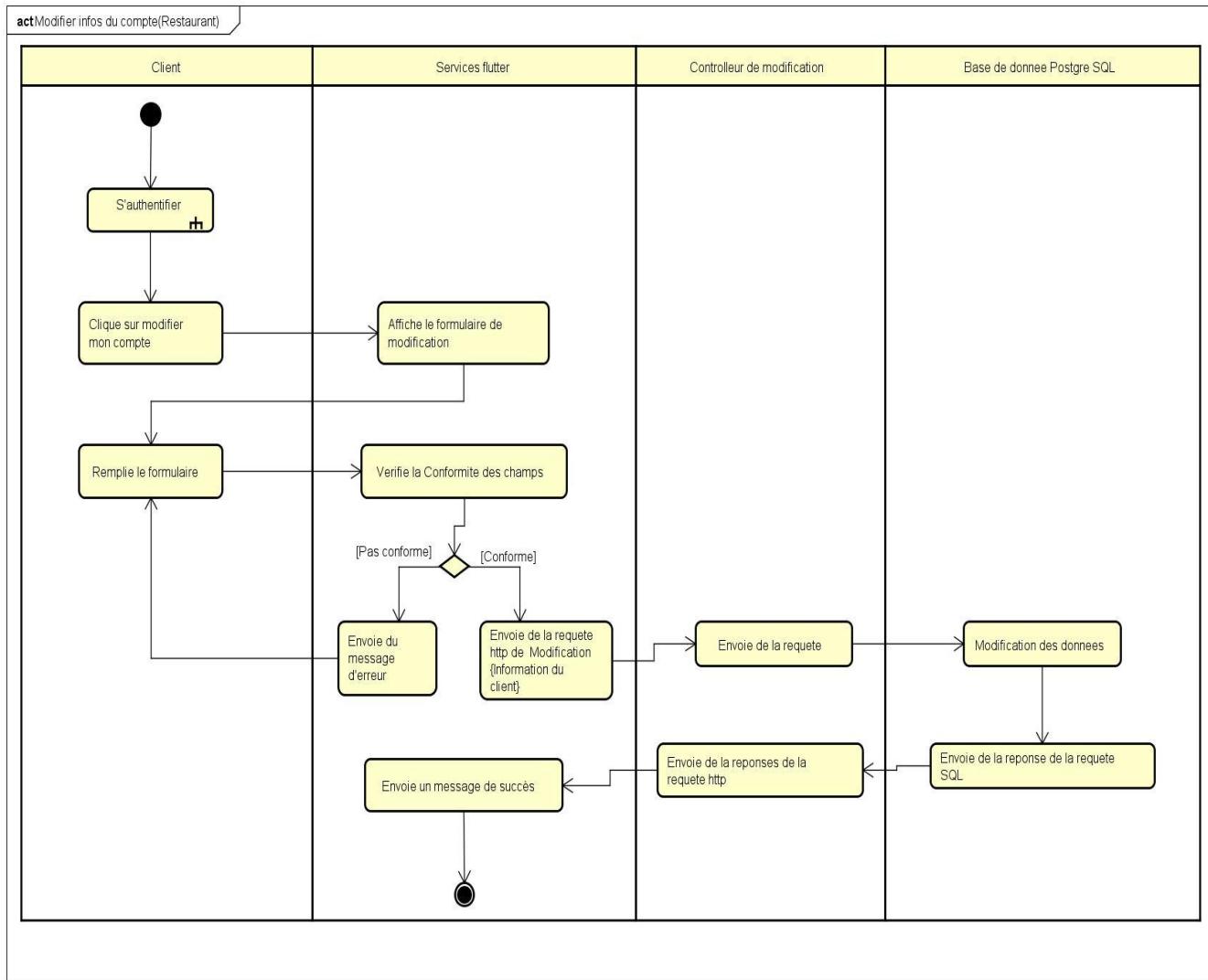
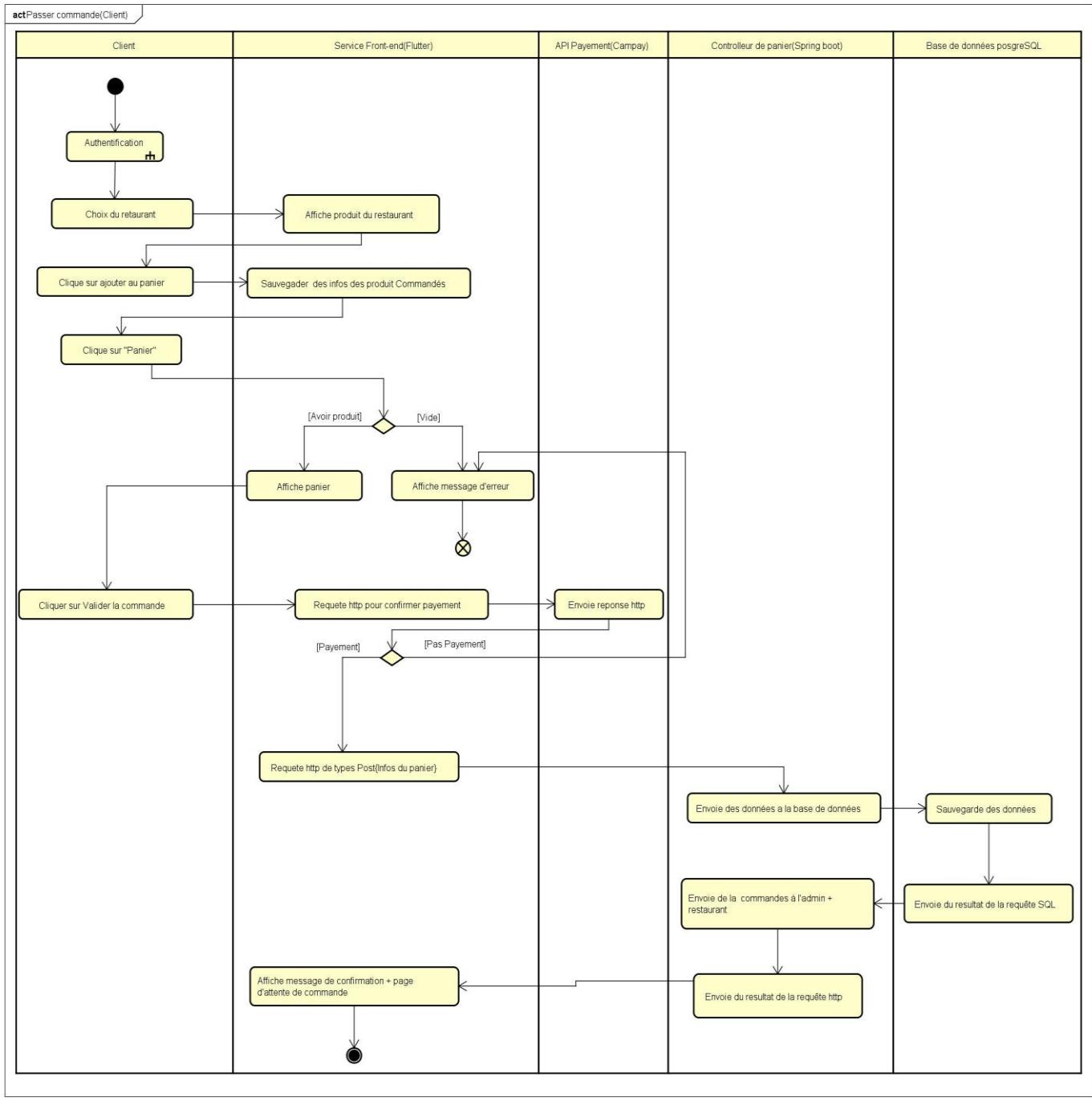


Figure 18:Diagramme d'activité Modifier compte restaurant

ii. Passer commande

Ce diagramme nous montre le déroulement du cas d'utilisation « **Passer commande** »



powered by Astah

Figure 19:Diagramme d'activité Passer commande

iii. Accepter Commande

Toutes les commandes effectuées par les clients seront directement assignées au restaurant sollicité qui devra lui ensuite valider la commande.

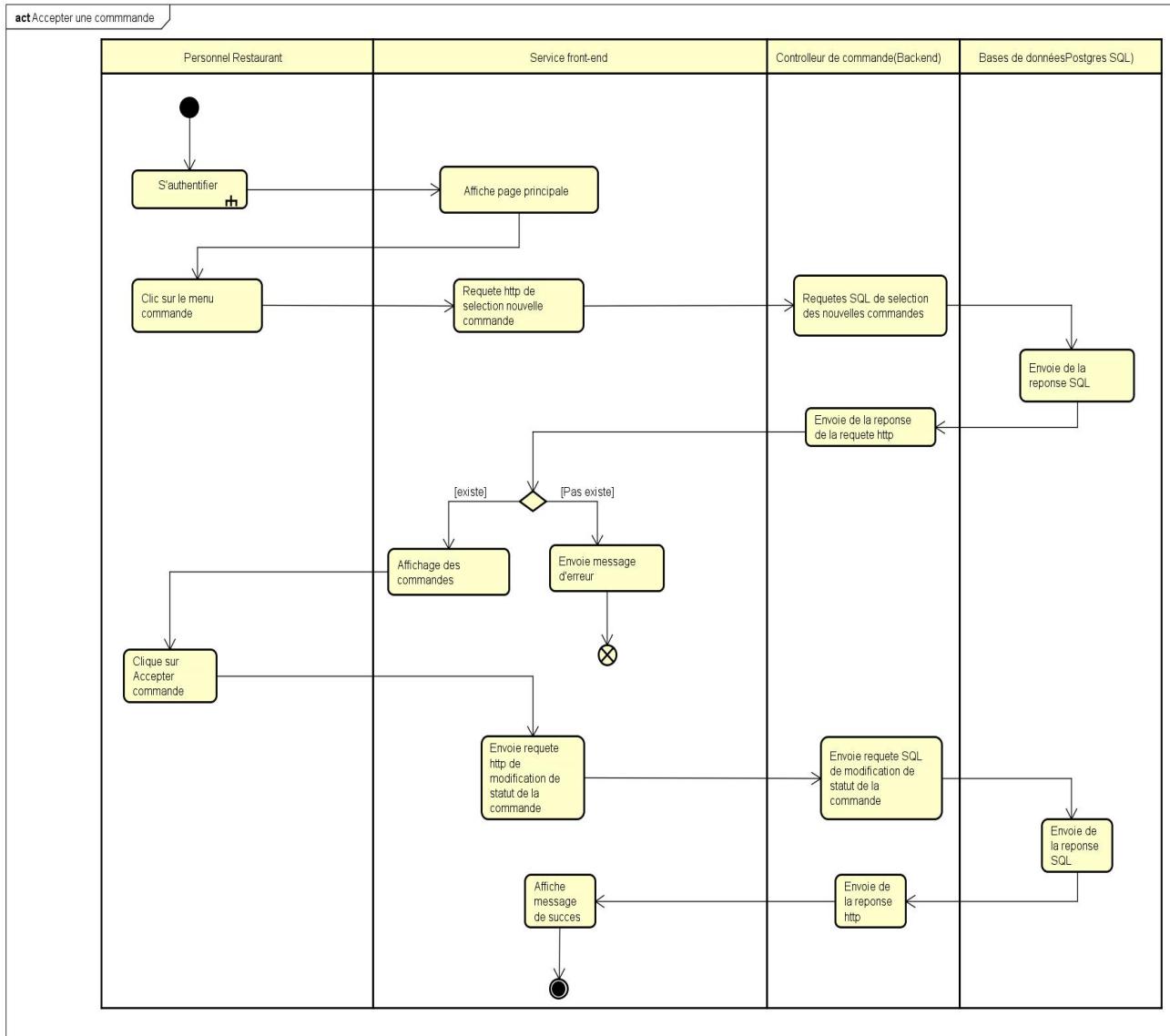


Figure 20:Diagramme d'activité accepter commande

E. Diagramme de communication

Un diagramme de communication dans le langage de modélisation unifié fait référence à un graphique qui représente le flux de messages dans un système. En un mot, il montre comment les parties d'un système interagissent ou, dans ce cas, communiquent entre elles.

1. Formalisme

Comme tout autre diagramme, le diagramme de communication UML comporte également plusieurs composants qui constituent son intégralité. La bonne chose à propos de ce schéma est qu'il peut être réalisé avec seulement quelques composants.

Tableau 15: Formalisme diagramme d'activité (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Composant	Description	Représentation
Objet	Est représentée par un rectangle contenant le nom et le type de l'instance.	
Les connecteurs	Les relations entre les lignes de vie sont appelées connecteurs et se représentent par un trait plein reliant deux lignes de vie.	
Les messages	Désignent une communication particulière entre les lignes de vie, et sont généralement ordonnés selon un ordre de numéro croissant.	

2. Quelques diagrammes de communication

i. Ajouter Produit

Chaque partenaire doit ajouter les produits de son restaurant qui sera ensuite disponible sur l'interface client. Le diagramme suivant montre la communication entre les éléments du système.

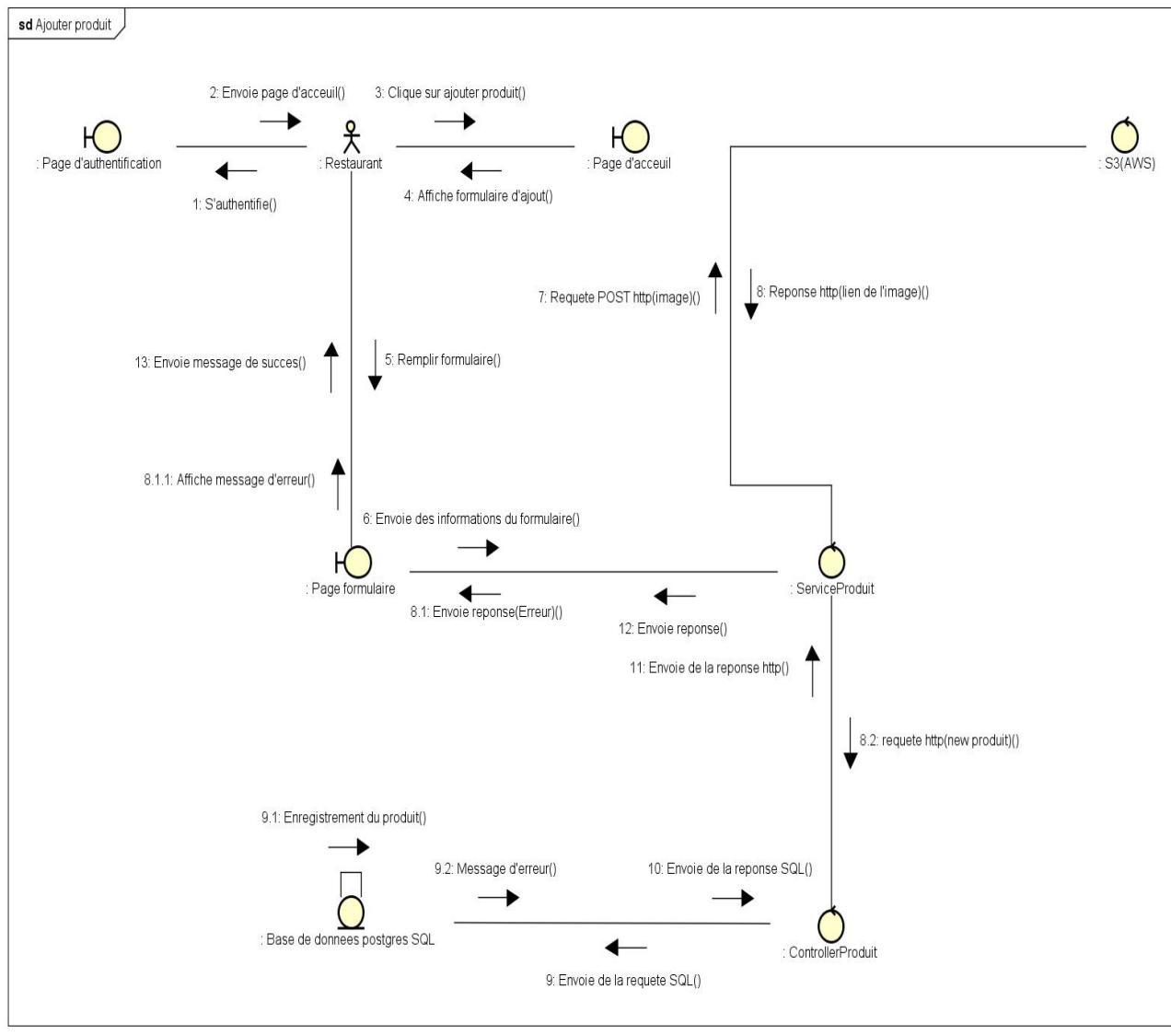


Figure 21:Diagramme de communication ajouter produit

ii. Effectuer paiement

Après avoir validé la commande, l'utilisateur devra effectuer un paiement afin de finaliser sa commande.

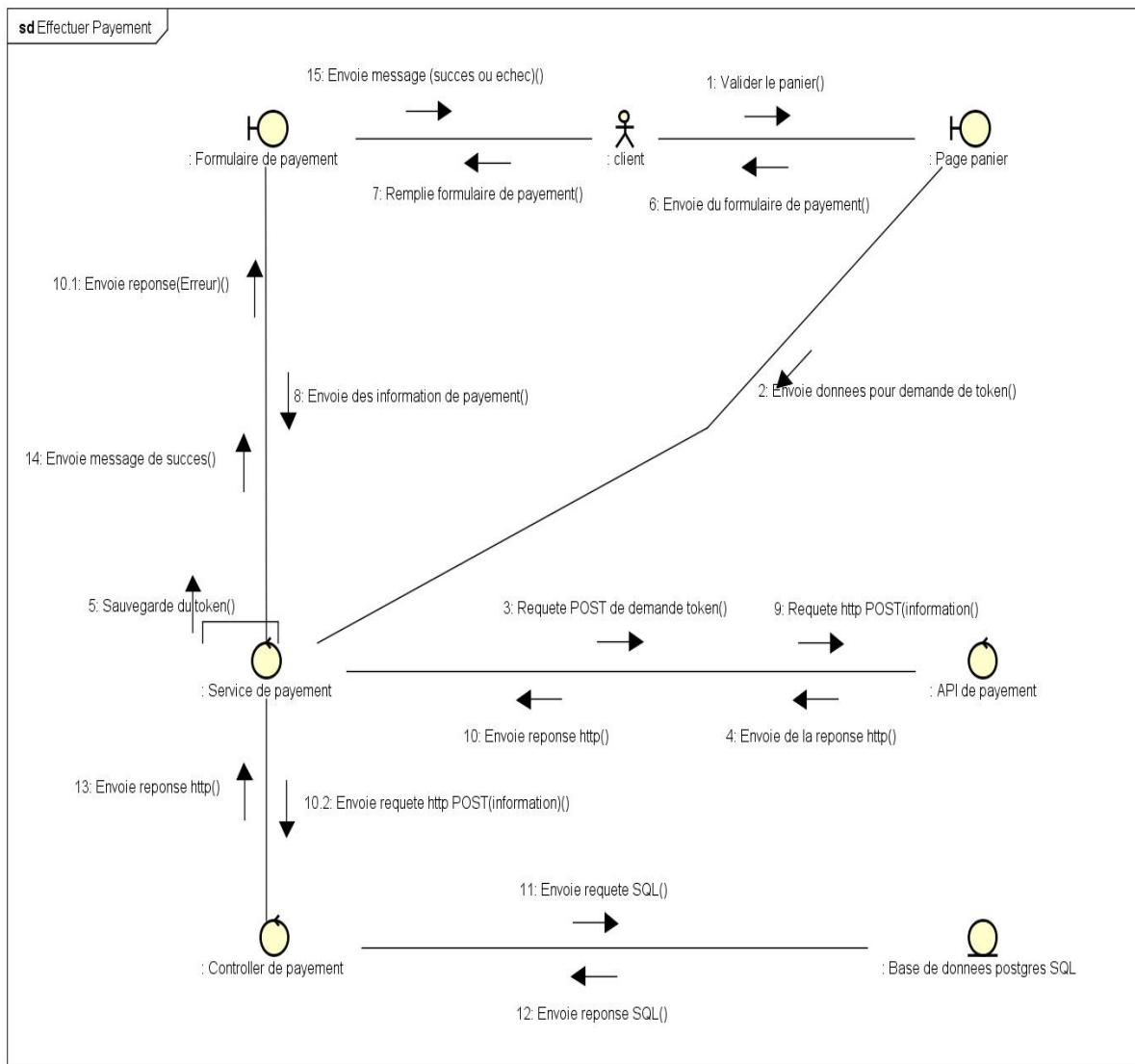
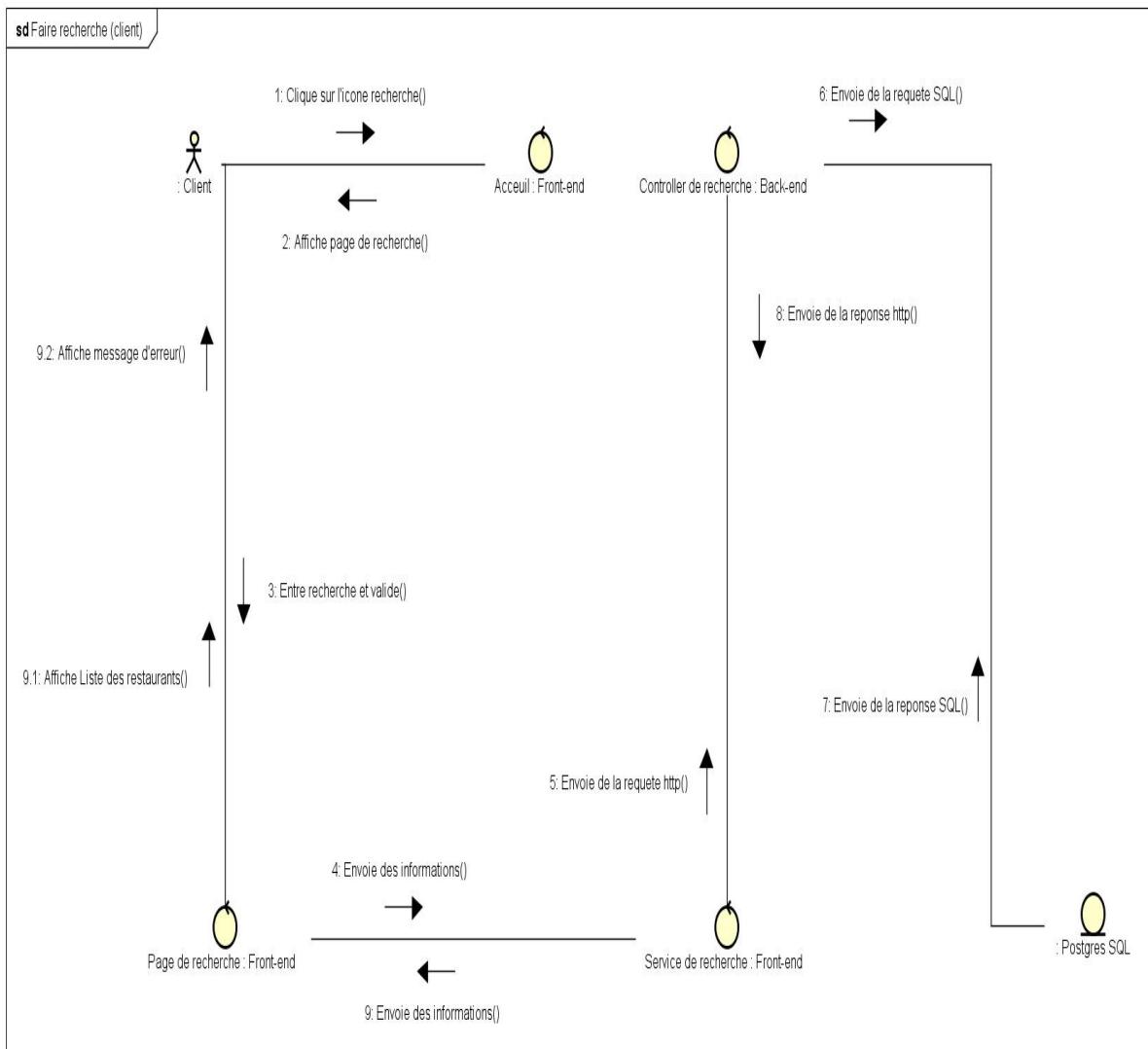


Figure 22:Diagramme de communication effectuer paiement

iii. Rechercher restaurant

Pour commander un repas le client doit peut passer par une recherche du produit ou du restaurant.



powered by Astah

Figure 23:Diagramme de communication effectuer recherche

CONCLUSION

En définitive, ce dossier d'analyse nous a permis d'avoir un aperçu détaillé du système à mettre sur pied. Toujours en exploitant le langage de modélisation UML et la méthode de gestion de projet agile Scrum, nous poursuivrons par le **dossier de conception** dans lequel nous représenterons les diagrammes de la branche technique du projet.



DOSSIER DE CONCEPTION

RESUME

Dans le cycle de vie d'un projet, la conception représente une phase primordiale et déterminante pour produire une application de haute qualité. C'est dans cette partie que nous allons détailler notre choix conceptuel à travers quelque diagramme.

APERÇU

INTRODUCTION

- I. DIAGRAMME DE CLASSE
- II. DIAGRAMME DE PACKAGE
- III. DIAGRAMME D'ETATS-TRANSITION
- IV. DIAGRAMME D'OBJET

CONCLUSION

INTRODUCTION

La phase de conception est en plein cœur dans le processus de développement d'un logiciel car elle est sans doute la phase la plus complexe. La conception bien faite d'un logiciel garantie à la fois les 4 principales fonctions d'un système d'information : collecte, stockage, transformation, et diffusion ; mais aussi une facilité de maintenance et d'utilisation. C'est l'une des raisons majeures pour lesquelles après avoir fait l'analyse d'un système nous allons nous en tenir sur le document de conception du futur système. D'après notre méthode d'analyse UML et notre méthodologie de gestion de projet Scrum nous allons donc rédiger ce document de conception à partir de 4(quatre) diagrammes principaux qui sont le **diagramme de classe**, **le diagramme de paquetages**, **le diagramme d'état transition** et **le diagramme d'objet**.

I. DIAGRAMME DE CLASSE

Le diagramme de classe exprime la structure statique du système de par ses classes et des relations entre ces classes. Ce diagramme fait partie de la vue statique d'UML, car il fait abstraction des aspects temporels et dynamiques. L'intérêt du diagramme de classe est de modéliser les entités du SI (Système d'information). Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par système

1. Formalisme

Le diagramme de classe met en évidence d'éventuelles relations entre ces classes. Le diagramme de classes comporte 6 concepts : **classe, attribut, identifiant, relation, opération, généralisation / spécialisation**. Ce diagramme est constitué des éléments suivants :

- **Classe** : il s'agit de la description abstraite à un ensemble d'objet : elle définit leurs structures, leurs comportements et leurs relations.
- **L'attribut** : Il représente la modélisation d'une information élémentaire représentée par son nom et format.

UML définit 3(trois) niveaux de visibilité pour les attributs :

- **Public (+)** : l'élément est visible par tous les clients de la classe.
 - **Protégé (#)** : l'élément est visible que par les sous classes ou les classes filles.
 - **Privé (-)** : l'élément n'est visible que par les objets de la classe auxquels il appartient.
- **L'identifiant** : c'est un attribut particulier, qui permet de repérer de façon unique chaque objet, instance de la classe.
 - **Multiplicité** : elle définit le nombre d'instance de la classe, elle est définie par un nombre entier ou un intervalle de valeur (1..1, 0..1, 1..*, 0..*).
 - **Associations** : Une association est une relation entre deux classes (association binaire) ou plus (association n-aire), qui décrit les connexions structurelles entre leurs instances.

Il existe plusieurs types de relations dont les plus connues sont les suivants :

- **Généralisation** : Relation d'héritage, dans laquelle les objets de l'élément spécialisé (Classe enfant) peuvent remplacer les objets de l'élément général (classe parent)
- **Agrégation** : utilisé pour les relations de types « contenant / contenu »

- **La composition :** La composition est un cas particulier de l'agrégation dans laquelle la vie des composants est liée à celle des agrégats.

Le tableau suivant nous donne une liste assez détaillée du formalisme du diagramme de classe en un tableau :

Tableau 16: Formalisme diagramme de classe (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Elément	Représentation	
Classe		
Association		
Composition		
Agrégation		
Généralisation		
Multiplicité	$0..*, 0..1, 1..1, 1..*$	
Modificateur d'accès	$+, -, \#, \sim, /$	

Exemple de diagramme de classe

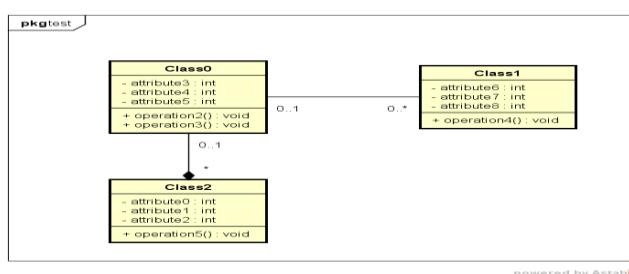


Figure 24: Exemple de diagramme de classe

2. Diagramme de classe du système

Le diagramme ci-dessous représente le diagramme de classe complète de notre futur système.

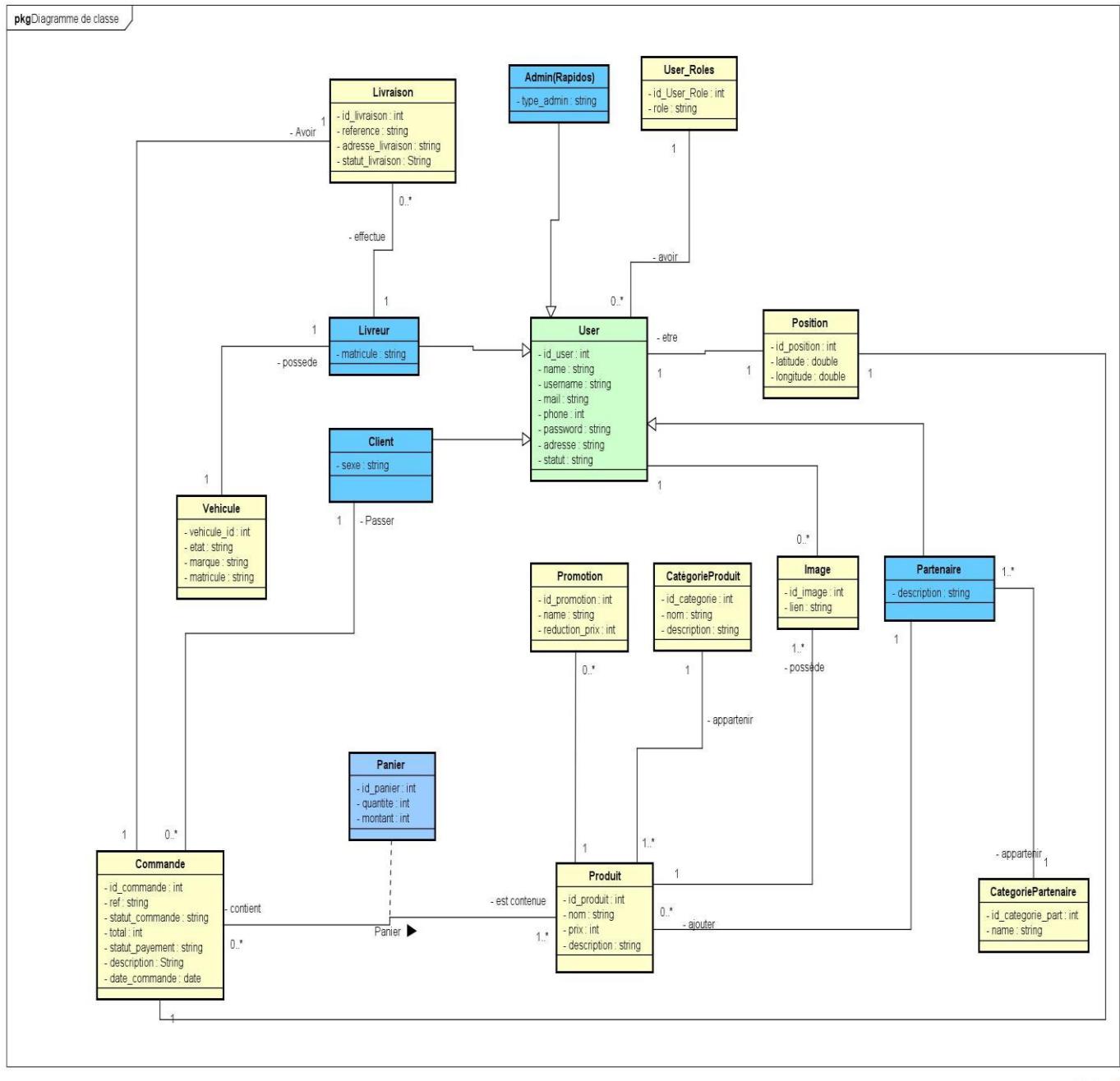


Figure 25:Diagramme de classe du système

II. DIAGRAMME DE PACKAGE

Les diagrammes de package (ou diagramme de paquetages) permettent de décomposer le système en catégories ou parties plus facilement observables appelées « packages ». L'utilisation la plus courante pour les diagrammes de paquetages est d'organiser des

Diagrammes de Cas d'Utilisation et des Diagrammes de Classes. Bien que l'utilisation des Diagrammes de Paquetages ne se limite pas à ces éléments UML.

1. Formalisme

Tableau 17: Formalisme diagramme de package (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Composant	Description	Représentation
Package	Regroupement des cas d'utilisation en catégorie	

2. Diagramme de package du système

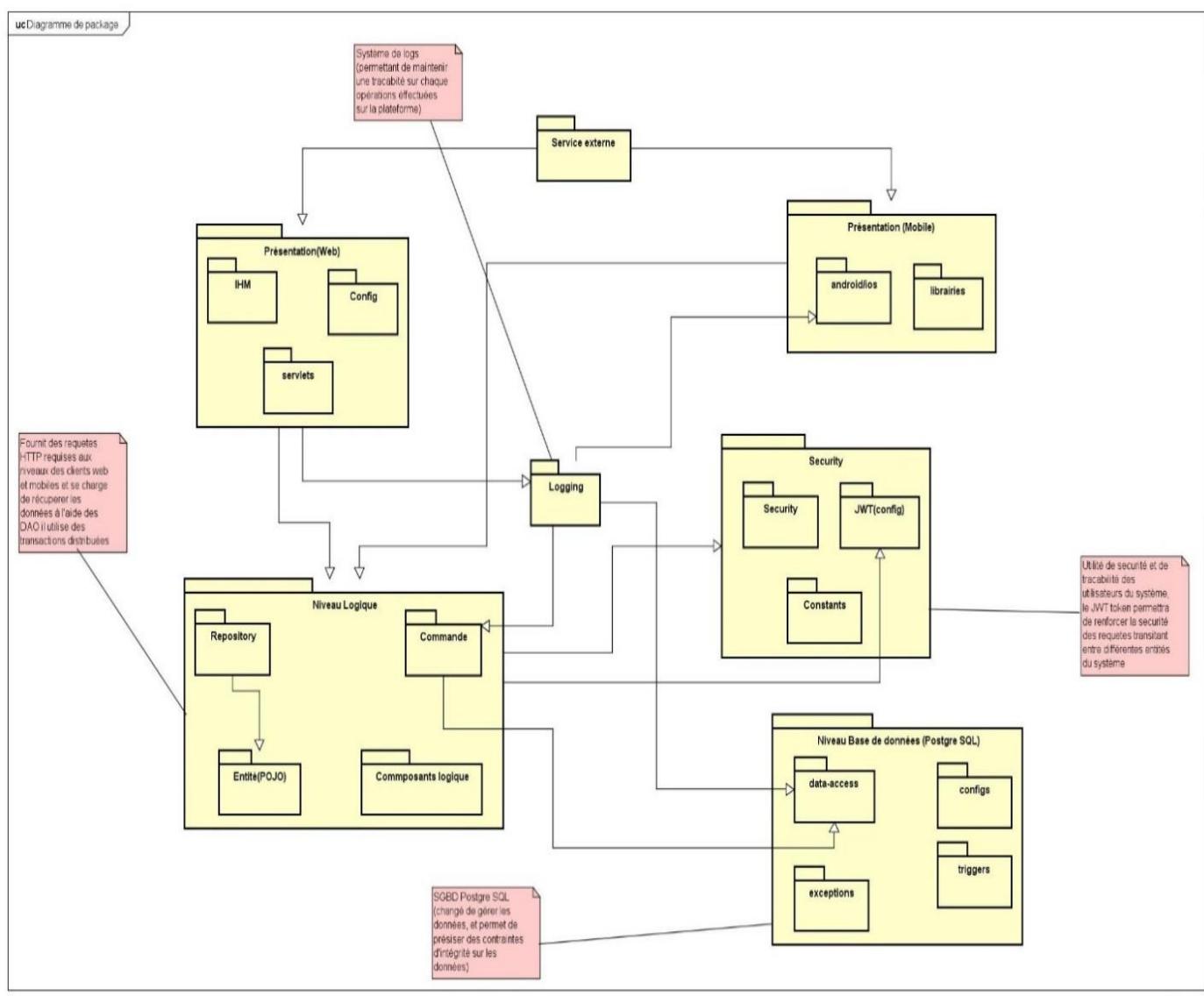


Figure 26:Diagramme de package du système

III. DIAGRAMME D'ETATS-TRANSITION

Le diagramme d'états-transition est comme son nom l'indique décrit les transitions entre les états et les actions que le système ou ses parties réalisent en réponse à un événement. Les diagrammes d'états-transitions permettent justement de spécifier les réactions d'une partie du système à **des événements discrets**. Un événement se produit à un instant précis et est dépourvu de durée.

1. Formalisme

Le diagramme d'états-transition décrit l'état ou le comportement d'un objet à un événement précis du système. A cet effet il est composé des éléments suivants :

Tableau 18: Formalisme Diagramme d'états-transition (*Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>*)

Elément	Description	Représentation
Etat initial	Il s'agit de l'état dans lequel il se trouve lors de sa création. Est représenté par un point noir.	
Etat final	Correspond à une étape où l'objet n'est plus nécessaire dans le système et où il est détruit. Il se représente par un point entouré d'un cercle.	
Etat intermédiaire	Est une situation stable qui possède une certaine durée pendant laquelle un objet exécute une activité ou attend un événement. Il est représenté par un rectangle arrondi contenant son nom.	
Transition	Une transition définit la réponse d'un objet à l'arrivée d'un événement. Elle est représentée par une flèche.	

Exemple de diagramme d'état transition

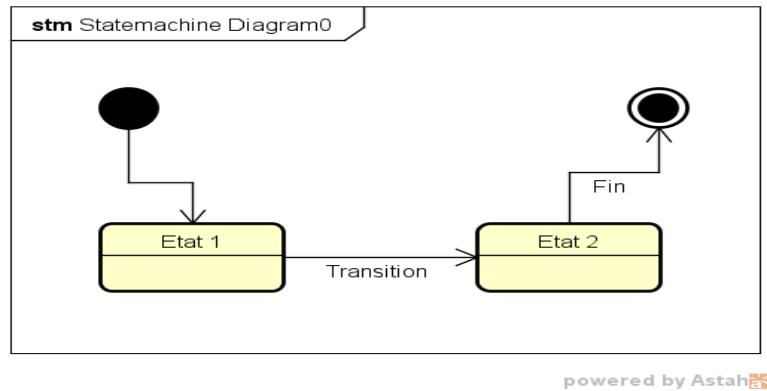


Figure 27: Exemple de diagramme d'états-transition

2. Diagramme d'états-transition du système

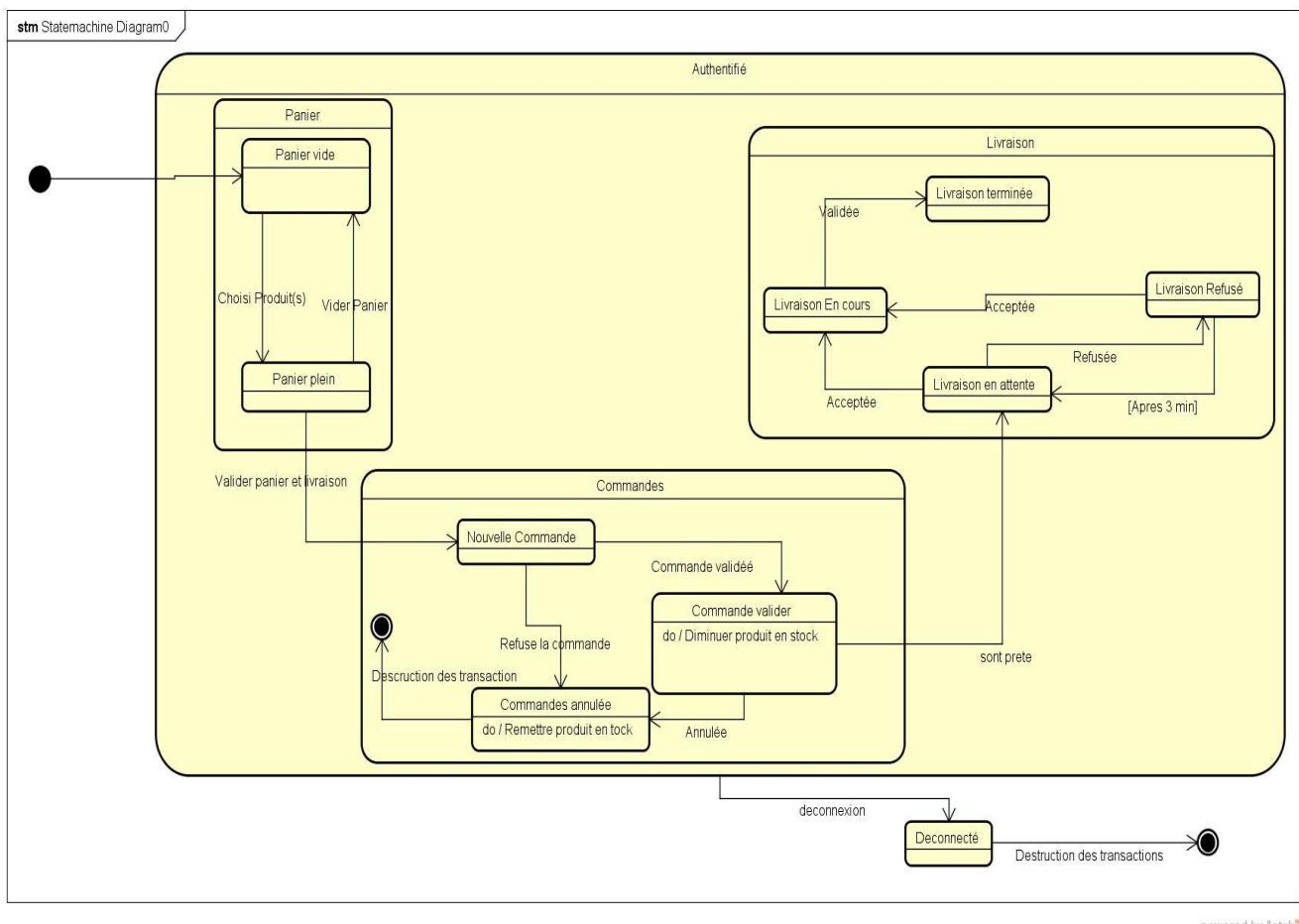


Figure 28: Diagramme d'états-transition du système

IV. DIAGRAMME D'OBJET

Un **diagramme d'objets** UML représente une instance spécifique d'un diagramme de classes à un moment précis. Dans sa représentation visuelle, il est très similaire à un diagramme de classes. Un diagramme d'objets se concentre sur les attributs d'un ensemble d'objets et sur la façon dont ils interagissent les uns avec les autres. A un diagramme de classe correspond une infinité de diagrammes d'objets.

1. Formalisme

Chaque **objet** est représenté dans un **rectangle** dans lequel figure le **nom de l'objet** et éventuellement la valeur de un ou plusieurs de ses attributs. Comme pour la représentation d'une classe, la représentation d'un objet pourra être plus ou moins détaillée.

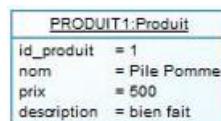


Figure 29:représentation d'un objet

Exemple de diagramme d'objet

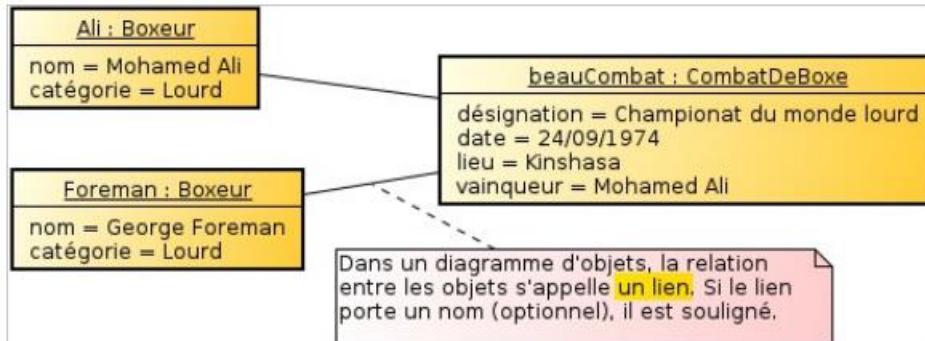


Figure 30:Exemple de diagramme d'objet

Diagramme d'objet du système

Le diagramme d'objet suivant est basé d'abord sur le diagramme de classe su système aussi représente les objets allant de la mise en système des produits par les partenaires passant par la commande par le client jusqu'à la livraison par le livreur.

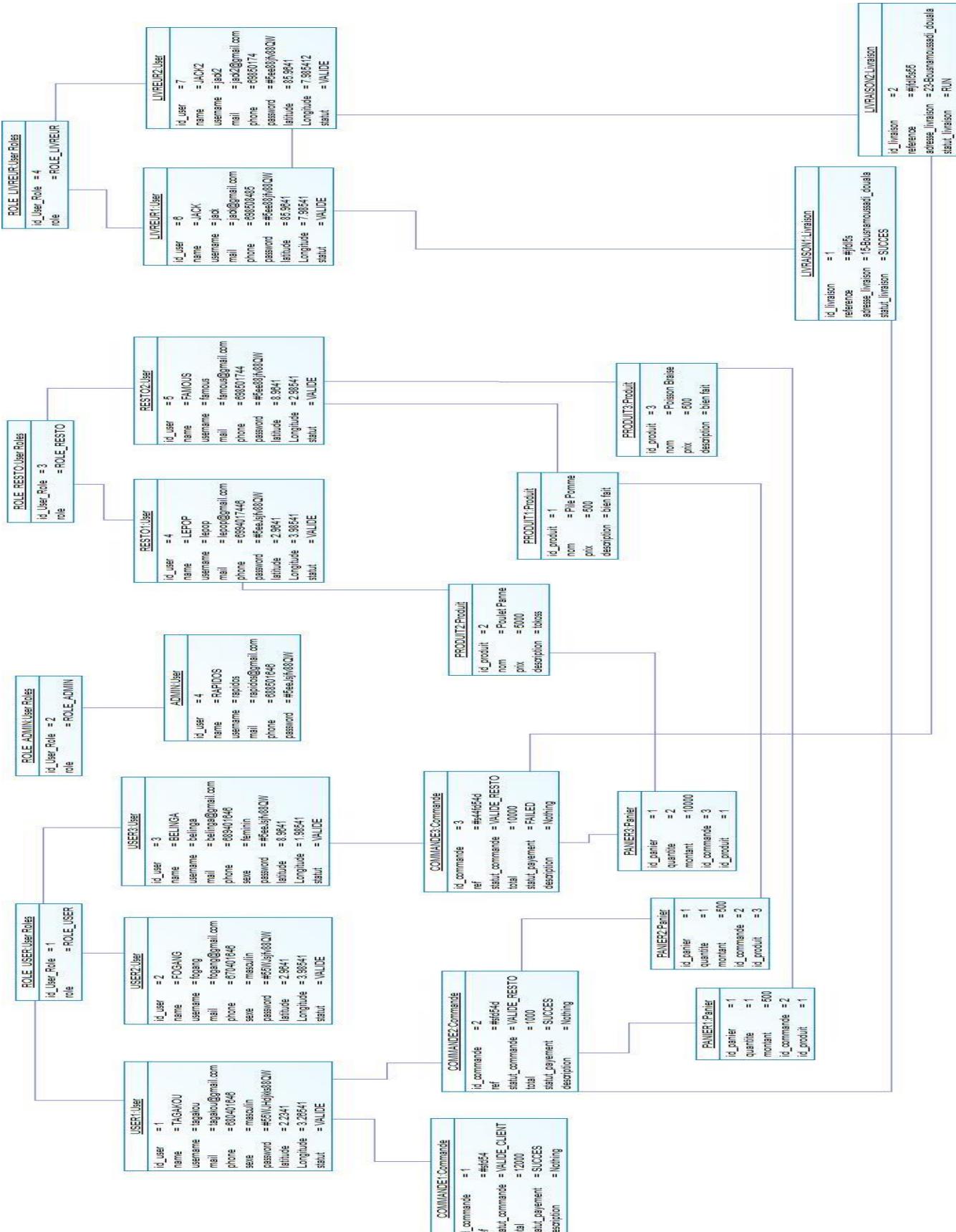


Figure 31: Diagramme d'objet du système du système

CONCLUSION

Le dossier de conception nous a permis de ressortir les données et les composants nécessaires pour la création de notre base de données et l'implémentation de notre application. Les différents éléments modélisés dans cette partie nous ont permis d'avoir une vue globale sur les différents modules de notre application, dès lors, l'étape suivante de notre projet sera la rédaction du **dossier de réalisation** tenant compte des différents éléments modélisés plus haut.



DOSSIER DE RÉALISATION

RESUME

Dans le dossier de réalisation, nous présentons les différents outils nécessaires à la mise en œuvre de la solution. Pour la réalisation du service web et de la plateforme de ventes, nous allons utiliser les technologies récentes en insistant sur l'aspect sécurité, et nous allons montrer les étapes à suivre dans la phase de déploiement.

APERÇU

INTRODUCTION

- I. ÉQUIPE PROJET ET CHOIX DES OUTILS
- II. PHASE D'IMPLÉMENTATION
- III. PHASE DE DÉPLOIEMENT

CONCLUSION

INTRODUCTION

La conception et la modélisation sont des maillons importants dans le processus de développement d'une application car il permet de décrire l'application en tant qu'instrument visible et manipulable. **La réalisation** représente l'aboutissement de tous les dossiers précédents à savoir le dossier d'analyse et le dossier de conception. C'est donc une concrétisation de la solution proposée. Il sera donc question pour nous tout au long de ce dossier de vous présenter l'équipe de développement et le choix des outils qui nous permettront de réaliser l'application, les langages de programmation et Framework utilisés et pour finir les résultats obtenus et quelques commentaires.

I. ÉQUIPES PROJET ET CHOIX DES OUTILS

1. Equipe projet

Le tableau suivant présente les différents membres ayant participé à l'implémentation de ce projet :

Tableau 19 : Equipe projet

Noms et prénoms	Rôles
M. Bertrand NGANSOP	Product Owner
M. NKOM TAKEU Guy Junior	Scrum master
Mme BELINGA Estelle	Equipe de développement Scrum
TAGAKOU brayan	Equipe de développement Scrum

2. Présentation des langages de programmation et Framework

Pour la réalisation de nos applications nous avons utilisé les technologies les plus récentes et surtout sécurisées vue la délicatesse de l'application, ceci pour faciliter la maintenance du système.

Tableau 20:Langages et Framework du projet

Langages et Framework	Rôle
JAVA	Langage de programmation utilisé pour créer des applications clientes lourdes
Dart	Langage de programmation optimisé pour concevoir les applications mobiles de bureau et des serveurs web
SQL/JPA	Langage utilisé pour communiquer avec la base de données
Flutter	Framework Dart utilisé pour le développement de logiciel d'interface utilisateur capable de fonctionner sur plusieurs plateformes.
Java Spring boot	Framework de développement applicatif JAVA recommandé pour le développement d'API.

II. PHASE D'IMPLÉMENTATION

1. Architecture logique de déploiement

Modèle-vue-contrôleur ou **MVC** est un motif d'architecture logicielle destiné aux interfaces graphiques lancées en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : **les modèles, les vues et les contrôleurs**. Le pattern **MVC** permet de bien organiser son code source. Il va vous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de **MVC** est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

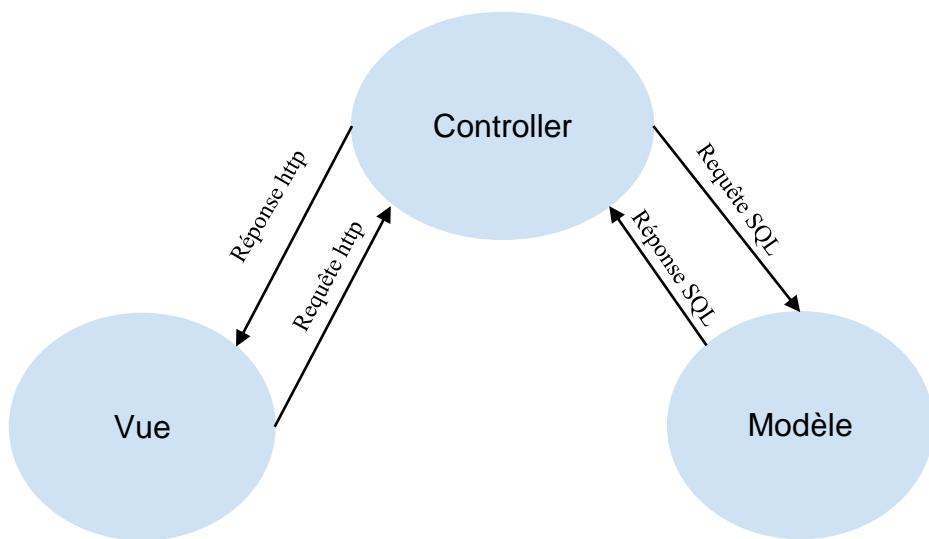


Figure 32: Architecture MVC

La logique applicative est basée sur l'architecture MVC qui est très pratiquée dans les projets. Il est un modèle de conception standard qui est connu de nombreux développeurs. Ici cette architecture nous a servie au niveau de notre back end notamment avec notre technologie spring boot

L'infrastructure MVC inclut les composants suivants :

- **Modèles** : Les objets de modèle sont les parties de l'application qui implémentent la logique du domaine de données de l'application. Souvent, ils récupèrent l'état du modèle et le stockent dans une base de données.
- **Vues** : Les vues sont les composants qui affichent l'interface utilisateur (IU) de l'application. En général, cette interface utilisateur est créée à partir des données du modèle. Il peut s'agir par exemple d'une vue d'édition d'une table affichant des zones de texte, des listes déroulantes et des cases à cocher en fonction de l'état actuel d'un objet.

- Contrôleurs :** Les contrôleurs sont les composants qui gèrent les interventions de l'utilisateur, exploitent le modèle et finalement sélectionnent une vue permettant de restituer l'interface utilisateur. Dans une application MVC, la vue sert uniquement à afficher les informations ; le contrôleur gère les entrées et interactions de l'utilisateur, et y répond. Par exemple, il gère les valeurs de chaîne de requête et les passe au modèle, qui peut à son tour les utiliser pour interroger la base de données

2. Diagramme de composant

Le diagramme de composants est principalement employé pour décrire les dépendances entre divers composants (les dépendances entre fichiers exécutables et les fichiers sources). Le diagramme de composants modélise les composants logiciels utilisés pour implémenter un système et l'association entre ces composants

i. Formalisme

Le diagramme de composant décrit les composants et leurs dépendances dans leurs environnements de réalisation. A cet effet il est composé des éléments suivants :

Tableau 21: Formalisme diagramme de composant (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Elément	Description	Représentation
Les Composants	Un composant représente une entité logicielle d'un système. Un composant est représenté par une boîte rectangulaire, avec deux rectangles dépassant du côté gauche.	
Dépendances	Elles sont utilisées pour modéliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche pointillée, se dirigeant d'un composant donné au composant dont il dépend.	
Interface (Nœud)	Lien entre deux composants logiciels. Il en existe 2 : <ul style="list-style-type: none"> • Interface fournie • Interface requise 	

Exemple de diagramme de composant

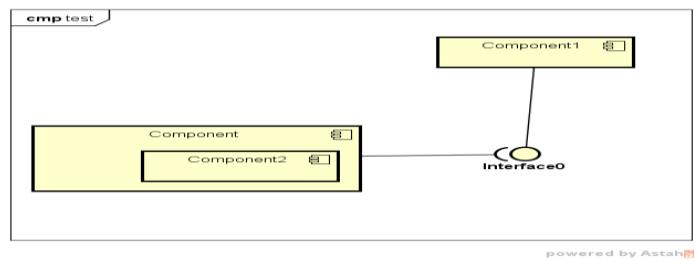


Figure 33: Exemple de diagramme de composant

ii. Diagramme de composant du système

Le diagramme suivant montre la structure des composants logiciels du système.

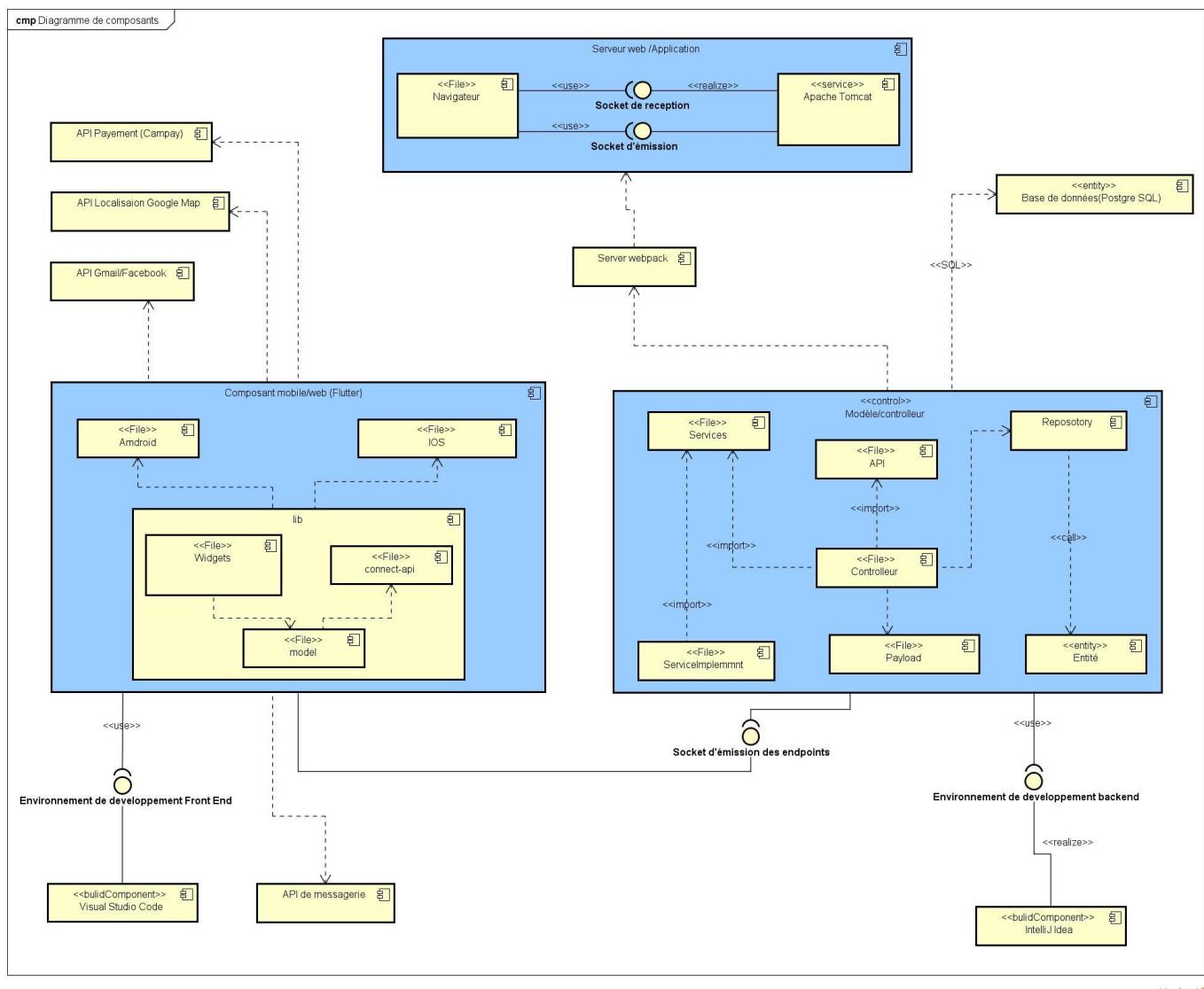


Figure 34: Diagramme de composant du système

3. Aspect sécurité de l'application

i. Sécurité des connections

Pour avoir l'accès à l'application les utilisateurs du front office à savoir les clients, doivent **posséder un compte** et **s'authentifier** en entrant un login et un mot de passe valide

ii. Sécurité du service web et mobile

Les applications à déployer seront muni d'une **sécurité stateless** c'est-à-dire sans état. Un token valide pendant 10(dix) minutes est généré pour chaque utilisateur dudit système et ce renouvelle automatiquement après authentification de l'utilisateur courant du système. Le token est injecté directement dans l'entête http de toutes les requêtes effectuées par l'utilisateur courant.

iii. Sécurité du mot de passe

Dans la base de données les mots de passe sont hachés grâce à notre fonction de hachage **SHA 256**. Et par conséquent les mots de passe seront illisibles même pour les administrateurs du système

iv. Sécurité via loggers

Le **loggers** permet de gérer des messages émis par une application durant son exécution et de permettre leur exploitation immédiate ou a posteriori. Ces messages sont d'ailleurs très utiles lors de la mise au point d'une application ou lors de son exploitation pour comprendre son fonctionnement ou résoudre une anomalie. Ils nous permettent de savoir qui est connecté en temps réel au niveau de notre back-end.

III. PHASE DE DÉPLOIEMENT

1. Architecture Physique de déploiement

Dans le domaine informatique, l'**architecture physique** (également nommée architecture technique) décrit l'ensemble des composants matériels supportant l'application.

On peut séparer notre application en trois parties dont les suivant :

- La couche interface homme machine (Vue mobile et web)
- La couche de traitement (L'API, Java Servlets)
- La couche de gestion de données (SQL, JDBC)

Dans le cadre de notre travail nous utiliserons l'**architecture n-tiers** qui est aussi appelée **architecture distribuée** ou **architectures multi-tiers**. Ceci se justifie à travers le fait

que ce type d'architecture qualifie la distribution d'application entre de multiples services et non la multiplication des niveaux de services

L'avantage de cette distribution est quelle est facilitée par l'utilisation des composants métiers spécialisés et indépendants introduit par les concepts de l'orienté objet. Ses composants rendent un service, si possible générique et clairement identifiées. Ils sont capables de coopérer entre eux en étant implantés sur des machines différentes et hétérogène.

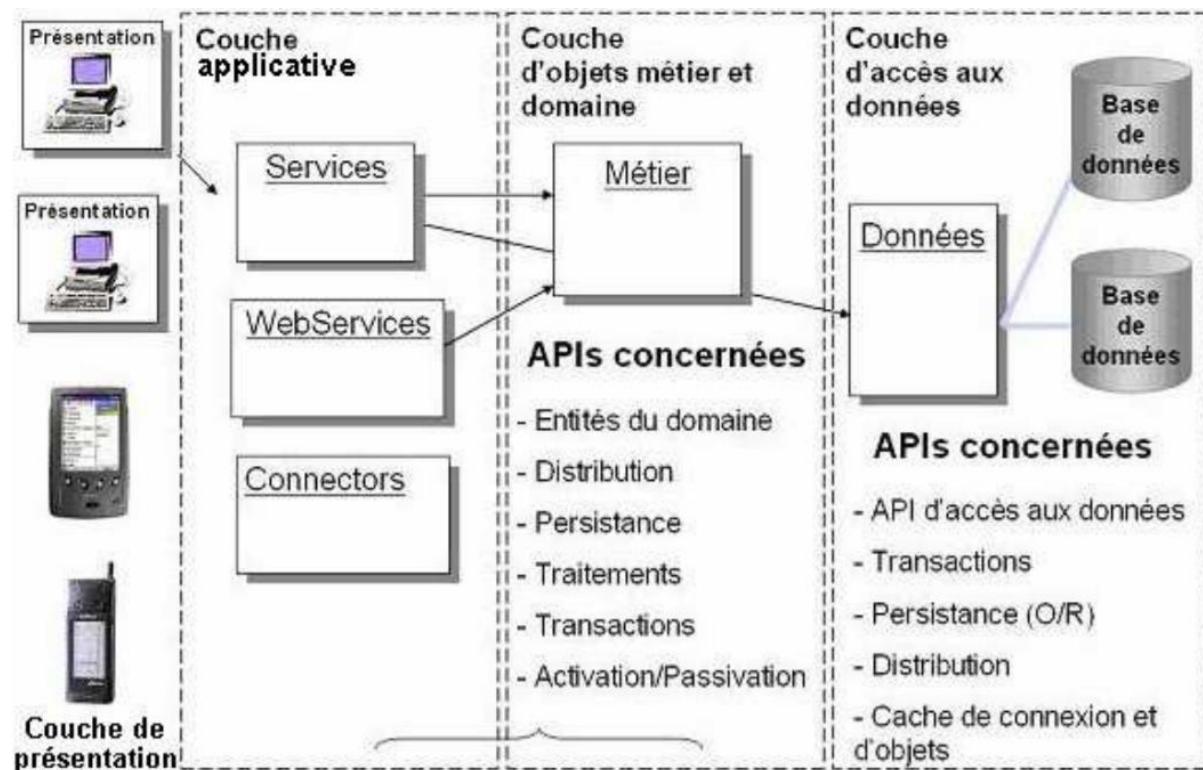


Figure 35: Représentation architecture n-tiers

(Source : <http://cedric.cnam.fr/~farinone/SMB111/annee1011/architectureNTiers.pdf>)

2. Diagramme de déploiement

Le **diagramme de déploiement** est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système. Et la manière dont les composants du système sont répartis ainsi que les relations entre eux

i. Formalisme

Le diagramme de déploiement peut être mis en œuvre dès la phase de conception pour documenter l'architecture physique du système.

Tableau 22: Formalisme du diagramme de déploiement (Source: <http://remy-manu.no-ip.biz/UML/Cours/coursUML7.pdf>)

Elément	Description	Représentation
Les composants	Il représente une entité logicielle du système (Fichier de code source, programmes, documents, fichiers de ressource, etc.). Un composant est représenté par une boîte rectangulaire avec deux rectangles dépassant du côté gauche portant le nom du composant.	
Les nœuds	Un nœud représente un ensemble d'éléments matériels du système. Cette entité est représentée par un cube tridimensionnel.	
Les Associations	Une association, représentée par une ligne pleine entre deux nœuds, indique une ligne de communication entre les éléments matériels.	
Les dépendances	Une dépendance est utilisée pour mobiliser la relation entre deux composants. La notation utilisée pour cette relation de dépendance est une flèche de pointillés.	

Exemple de diagramme de déploiement

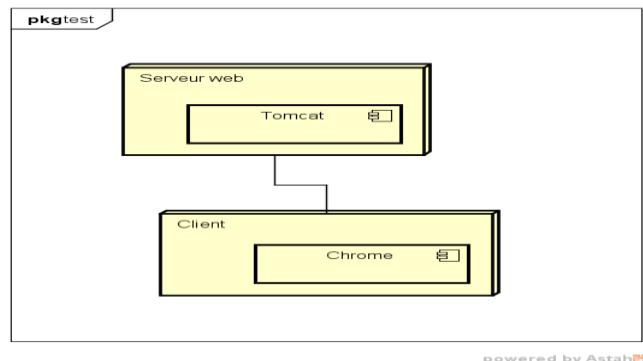


Figure 36: Exemple de diagramme de déploiement

ii. Diagramme de déploiement du système

Le diagramme suivant est une architecture utilisée pour le déploiement de notre application.

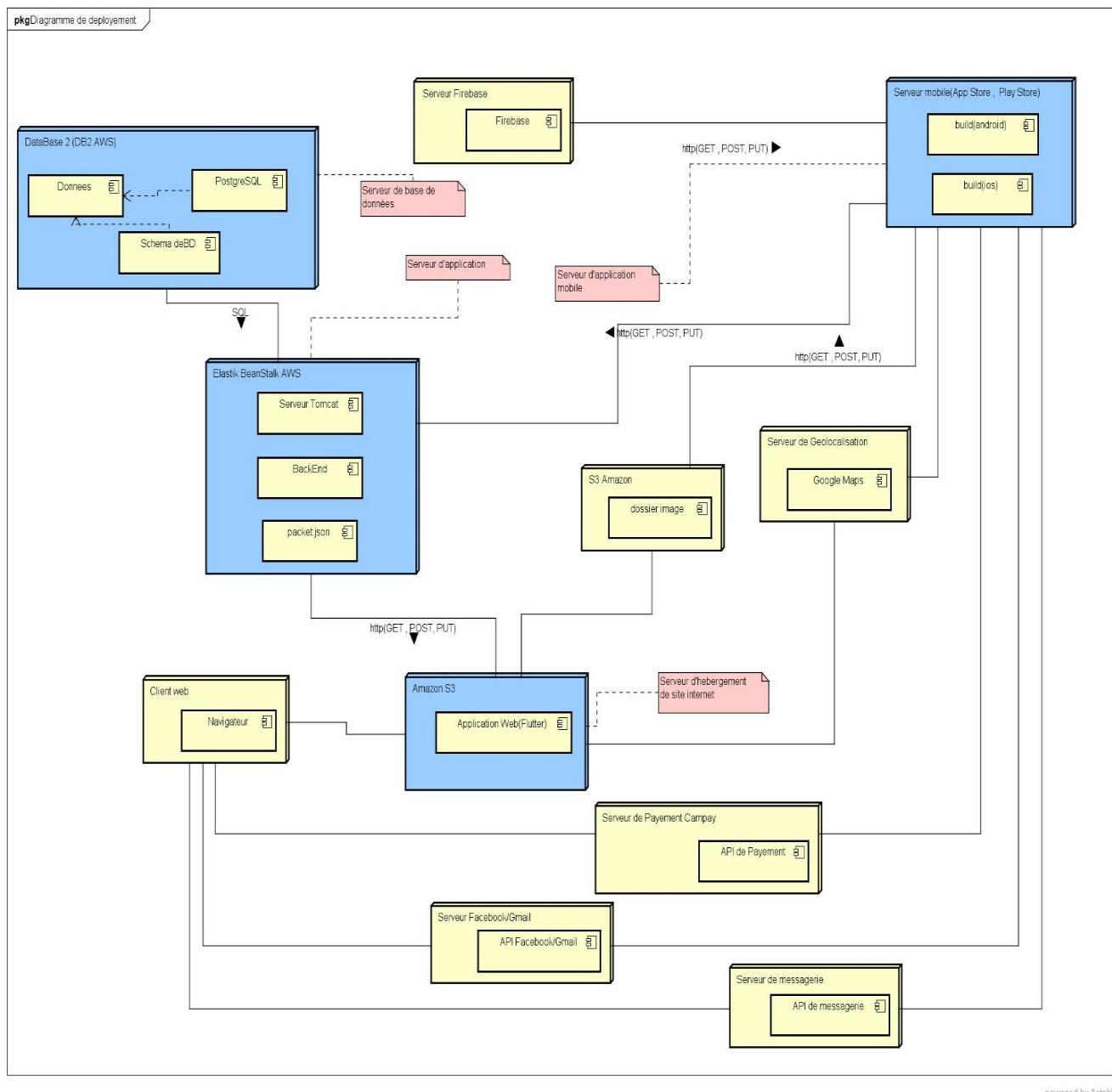


Figure 37: Diagramme de déploiement du système

CONCLUSION

Le dossier de réalisation ainsi finement élaboré nous a permis de solidifier du mieux l'environnement de déploiement de notre futur système. Cette multiplateforme nommée **RapidosEat** va sans doute améliorer l'expérience de d'achat et de livraison de la population en général et même des entreprises et particuliers. Mais pour l'utiliser, il faut faire recours à un **guide d'utilisation**. Pour cela nous consacrerons la dernière partie de notre projet à cette rubrique.



TESTS DE FONCTIONNALITÉS

RESUME

Un **test** désigne une procédure de vérification partielle d'un système. **Son objectif principal est d'identifier un nombre maximal de comportements problématiques du logiciel.** Il permet ainsi, dès lors que les problèmes identifiés seront corrigés, d'en augmenter la qualité.

APERÇU

INTRODUCTION

- I. TESTS FONCTIONNELS
- II. TESTS NON-FONCTIONNELS
- III. TEST DE PERFORMANCE

CONCLUSION

INTRODUCTION

La réalisation du logiciel est une étape très importante dans le cycle de vie d'un logiciel car à la sortie de celle-ci nous avons une application déjà déployable mais pas prête à l'utilisation parce qu'aucune phase de test n'a encore été effectuée. Cette phase de test ici nous permettra de dénicher d'éventuels bogues et erreurs. Il sera question pour nous tout au long de ce document de présenter d'abord les **tests fonctionnels**, ensuite les **tests non fonctionnels** et enfin les **tests de performances**.

I. TESTS FONCTIONNELS

Les tests fonctionnels sont effectués pour vérifier les fonctionnalités et la facilité d'utilisation. Il garantit si toutes les fonctionnalités se comportent comme prévu sans aucun problème. Dans notre cas nous avons utilisé **Postman** pour attaquer l'API qui nous fournit les processus métiers du logiciel.

La requête http lancée possède les parties suivantes :

- Lignes de la requête
 - Méthodes [1]
 - URL [2]
 - Version du protocole utilisé par le client (Dans notre cas la version http /0.1)
- Champs d'entête/autorisation de la requête [3]
- Corps de la requête [4]

A l'envoi de la requête, l'API se charge toujours de renvoyer une **réponse http** [5] possédant un ensemble d'éléments caractéristiques [6] constitués de :

- Le statut code
- Le temps de réponse en milliseconde **ms**
- La taille des données de la réponse et kilobit **KB**

Quelques tests importants ont été réalisés dans notre système et ont été répartis comme suit:

1. Test d'inscription

Ce test consiste à envoyer des données JSON contenant les informations de l'utilisateur à enregistrer.

The screenshot shows the Postman interface with the following details:

- Left Sidebar (Collections):** Shows a collection named "rapidos" containing several API endpoints such as "Ajouter resto", "Ajouter livreur", and "Ajouter client".
- Central Area:**
 - Request URL:** POST <http://rapidoseat-env.eba-adp3mtww.us-east-1.elasticbeanstalk.com/rapidoseat/auth/signup>
 - Method:** POST (highlighted with a red box labeled 1)
 - Body:** JSON (highlighted with a red box labeled 4)
 - Response:** Status: 201 Created Time: 1111 ms Size: 502 B (highlighted with a red box labeled 6)
 - ResponseBody:** (highlighted with a red box labeled 5)

```

1
2 "SUCCESS": true,
3 "message": "Utilisateur enregistre avec succes"
4
      
```

Figure 38: Test d'inscription

2. Test de connexion (Authentification)

Il s'agit d'envoyer les données de connexion afin d'avoir un token qui nous donnera accès à d'autres fonctionnalités à validité grâce à ce jeton.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, Monitors, and History. The main area displays a 'Scratch Pad' workspace titled 'rapido / connexion'. Within this workspace, a POST request is configured to the URL <http://rapidoseat-env.eba-adp3mtww.us-east-1.elasticbeanstalk.com/rapidoseat/auth/signin>. The 'Body' tab is selected, showing the JSON payload:

```

1
2   ...
3     "username": "estelle",
4     ...
5     "password": "estelle"

```

Below the request, the 'Body' tab is active, showing the response status: 200 OK, Time: 771 ms, Size: 740 B. The response body is displayed as:

```

1
2   "accessToken": "eyJhbGciOiJIUzUxMiJ9.eyJzdWJiOiI0IiwiZWFOIjoxNjY0NDM1Mzc0LCJleHAiOjE2NjUwNDAxNzR9.
3     ZAVsCYV3m6tGQDnCh03sxyMN0tduxGKMyA55XScEkB5TJotzW0Kt25atUfSzJN500W613707R42pQ_9Zg",
4     "tokenType": "Bearer",
5     "userRole": "ROLE_CLIENT"

```

Figure 39: Test de connexion

3. Test de catégorie

Ce test consiste à avoir la liste de toutes les catégories de produit de notre base de données retourné sous forme de JSON. Cette requête requiert d'une autorisation de type **Bearer token** afin d'être exécutée.

The screenshot shows the Postman interface with the following details:

- Collection:** rapidos / affiche cat
- Request Type:** GET
- URL:** http://rapidoseat-env.eba-adp3mtww.us-east-1.elasticbeanstalk.com/rapidoseat/categories/tout
- Authorization:** Bearer Token (highlighted in red)
- Body:** (Empty)
- Response:**
 - Status: 200 OK
 - Time: 1307 ms
 - Size: 1.25 KB
 - Content (Pretty):

```

1 {
2   "createdAt": 1664438834.405281000,
3   "updatedAt": 1664438834.405281000,
4   "createdBy": 3,
5   "updatedBy": 3,
6   "categorie_id": 2,
7   "libelle": "Met d'Afrique",
8   "image": "https://images.rapidoseat.s3.amazonaws.com/CategoriesImage/2022/Septembre/Met+d'Afrique+2904529.jpg",
9   "description": "Vos meilleurs plats africain"
10 },
11 [
12   {
13     "createdAt": 1664444103.069050000,
14     "updatedAt": 1664444103.069050000,
15     "createdBy": 5,
16     "updatedBy": 5,
17   }
18 ]

```

Figure 40: Test de catégories

4. Test de produits

Ce test nous permet d'avoir tous les produits de notre plateforme. Cette requête requiert une autorisation de type **Bearer token** afin d'être exécutée.

```

[{"id": 3, "designation": "pizza au champignon", "prixUnitaire": 3999.0, "details": "pizza familiale \ncontenu: champignon \n- piment\n- tomate", "total": null, "seuil": null, "prixAchatMoyen": 0.0, "restant": null, "pourcentageVente": null, "pix": null, "photo": null, "image": "https://images.rapidoseat.s3.amazonaws.com/ProduitsImage/raphaelo/raphaelo29105212.jpg", "createdAt": 166445345.537342000, "updatedAt": 166445345.537342000, "createdBy": 5, "updatedBy": 5, "produit_id": 3, "categorie": "Pizzas", "stock": 100, "stockMin": 50, "stockMax": 150, "stockAlert": 75, "stockAlertMin": 50, "stockAlertMax": 100, "stockAlertTime": "2022-09-10T10:52:12.000Z", "stockAlertTimezone": "Africa/Douala", "stockAlertType": "Email", "stockAlertEmail": "raphaelo29105212@gmail.com", "stockAlertSubject": "Stock Alert for Pizza au champignon", "stockAlertMessage": "The stock level for Pizza au champignon has reached the alert threshold of 75 units. Current stock: 100 units.", "stockAlertTimezone": "Africa/Douala", "stockAlertType": "Email", "stockAlertEmail": "raphaelo29105212@gmail.com", "stockAlertSubject": "Stock Alert for Pizza au champignon", "stockAlertMessage": "The stock level for Pizza au champignon has reached the alert threshold of 75 units. Current stock: 100 units."}]
    
```

Figure 41:Test de produits

5. Test d'ajout de restaurant

Ce test consiste à ajouter un restaurant en qui sera ensuite visible chez les clients. Cette requête requiert d'une autorisation de type **Bearer token** afin d'être exécutée.

```

{
  "success": true,
  "message": "Restaurant enregistr\u00e9 avec succ\u00e8s"
}
    
```

Figure 42: Test d'ajout de restaurant

II. TESTS NON-FONCTIONNELS

Les tests non fonctionnels sont comme les tests fonctionnels ; toutefois, la principale différence est que ses fonctionnalités sont testées sous charge pour observer les performances (rapidité), la fiabilité, la facilité d'utilisation, l'évolutivité. Les types de tests non fonctionnels comprennent les tests d'installation, les tests de fiabilité et les tests de sécurité. Notre application étant basée sur des technologies de pointe comme le JAVA, Flutter, etc.... nous permet d'affirmer que notre application est à la fois sécurisée, évolutive, et facile à l'utilisation.

III. TEST DE PERFORMANCE

Les tests de performance sont un type de test non-fonctionnel, effectué pour déterminer la vitesse, la stabilité et l'évolutivité d'une application logicielle. Grâce à la technologie Flutter le système fonctionne à la fois sur Android et sur IOS côté mobile ceci avec la même vitesse d'exécution. Ceci peut aussi se confirmer à travers le temps de réponse des requêtes exécutées précédemment. Pour la plateforme web le logiciel s'adapte à toutes les formes d'écran.

CONCLUSION

En somme, cette phase mettant en exergue les **tests de l'application a été** effectuée à partir des éléments suivants : tests fonctionnels, tests non-fonctionnels et tests de performances. Cette étude a donc relevé que cette application est déployable et prête à une première utilisation effective.



GUIDE D'UTILISATION



RESUME

Dans le guide d'utilisation, nous présentons le processus d'installation de la plateforme (l'application mobile), et nous présentons par la suite quelque fonctionnalité du système final.

APERÇU

INTRODUCTION

- I. GUIDE D'INSTALLATION
- II. GUIDE D'UTILISATION

CONCLUSION



INTRODUCTION

Le guide d'utilisation est un document qui permet aux futurs utilisateurs du système de découvrir ses différentes fonctionnalités. C'est un document important pour aider à manier l'application aisément de son installation jusqu'à l'utilisation des différents modules. Il sera question tout au long de ce document de présenter d'une part le guide d'installation et d'autre part le guide d'utilisation.

I. GUIDE D'INSTALLATION

L'installation de l'application RapidoEat ne nécessite un minimum d'étapes qui sont les suivant :

- Aller sur l'application Play store



- Entrer dans la barre de recherche **RapidoEat** et cliquer dessus ;

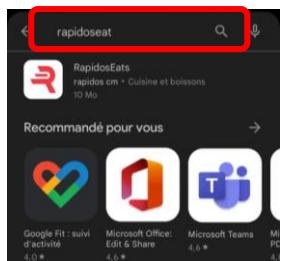


Figure 43:Guide d'installation1

- Cliquer sur installer

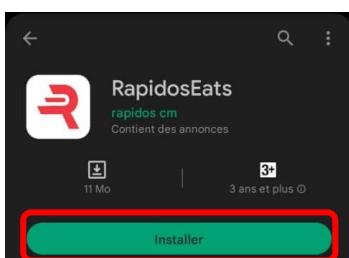


Figure 44:Guide d'installation2

II. GUIDE D'UTILISATION

Pour les utilisateurs ne possédant pas de navigateur ou souhaitant utiliser l'application mobile, ils devraient passer par les étapes suivantes pour faciliter la consultation des menus jusqu'au passage de la commande. L'utilisateur possédant la plateforme mobile pourra ainsi effectuer les actions suivantes :

- **Ouvrir l'application** : ici, l'utilisateur **doit taper sur le logo** présent sur sa page d'accueil de son téléphone ou dans son menu :



Figure 45:Icône de lancement de l'application

- Après être entré dans l'application le nouvel utilisateur aura **une page d'accueil** ou il aura la possibilité entre se **connecter** ou **s'inscrire** sur plateforme.



Figure 46:Page d'accueil Rapido Eat

- Dans le cas où l'utilisateur clique sur « **Inscription** », il aura accès au formulaire d'inscription à la plateforme RapidoEat. Où il doit remplir ses informations et cliquer sur le bouton **s'inscrire**

Figure 47:Page d'inscription Rapido Eat

- Avant d'accéder à la page de connexion il en existe trois possibilités : Soit
 - Cliquer sur « **connexion** » à la page d'accueil
 - Après s'être inscrit l'utilisateur est directement redirigé à la page connexion afin d'être connecter et éventuellement recevoir un nouveau Token
 - Soit en cliquant sur le bouton **déconnexion** qui vous sera présenté en dessous

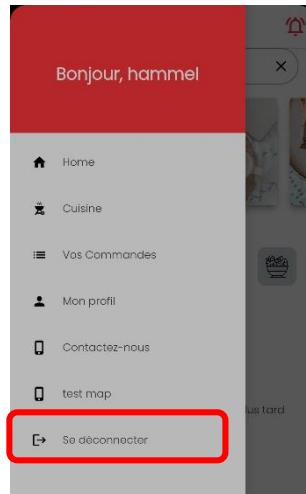


Figure 48:Drawer page RapidosEat

Et après avoir saisi son login et son mot de passe correct pourra ainsi accéder au home page



Figure 49:Page de connexion RapidosEat

- Après connexion ou possédant un Token valide, un client pourra ainsi consulter la liste des restaurants ou supermarchés partenaire à Rapidos Sarl à la page d'accueil des connectées.



Figure 50:Page d'accueil des utilisateurs authentifiés

- Un client après avoir taper sur le restaurant de son choix aura accès aux menus et au produit proposé par celui-ci.

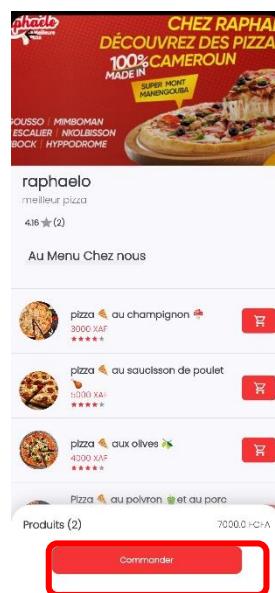


Figure 51:Page d'accueil des restaurants

- De là, le client peut directement ajouter un produit dans son panier ou **cliquer sur ce produit** pour avoir une meilleure vision de son produit. Cette page nommée **Single-Product** Donne la possibilité au client d'avoir une meilleure description du produit.



Figure 52:Single Product page

- Après avoir fait tous ses achats le client aura la possibilité de consulter son panier avant de valider ses achats en cliquant sur n'importe quel bouton **Consulter Panier**.

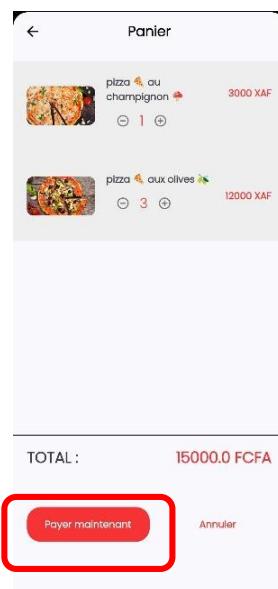


Figure 53:Panier sur Rapido Eat

- Après avoir validé son panier, le client peut taper sur « **Payer maintenant** » et ainsi il sera dirigé vers une carte Google Map qui lui permettra de localiser son lieu de livraison.

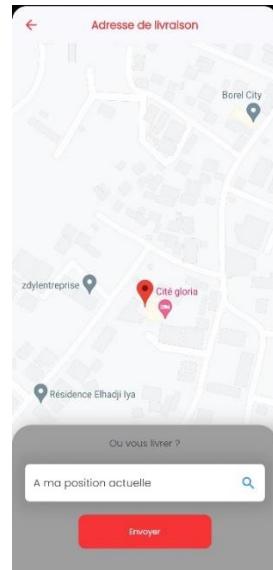


Figure 54:Page adresse de livraison

- La prochaine étape consiste au paiement : le client du système devra consulter sa facture et ensuite entrer son numéro de téléphone de paiement valide et le soumettre. Après validation il devra recevoir une notification de l'opérateur de paiement lui demandant de valider le paiement.

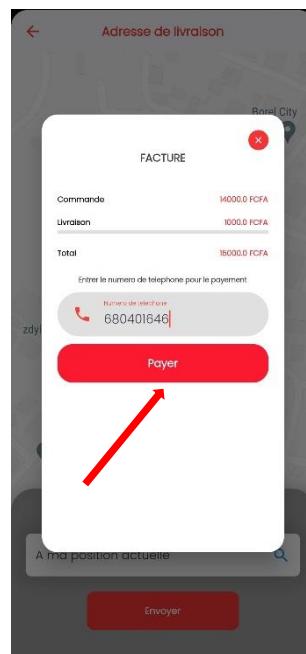
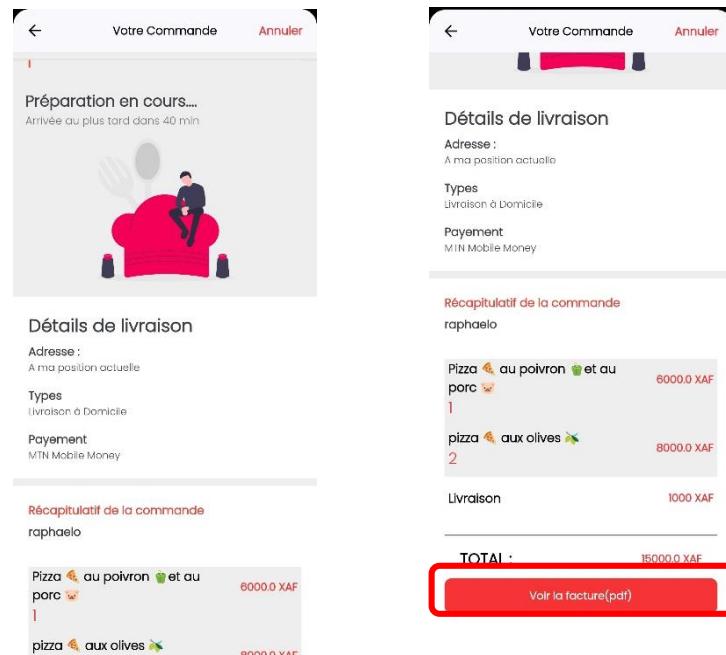


Figure 55:Page facture

- Paiement bien effectué le client sera redirigé vers une page d'attente constituée de son temps de livraison, son adresse de livraison et le total de sa commande.



- L'utilisateur pourra générer sa facture PDF en cliquant sur **voir facture PDF** et l'ouvrir directement sur la plateforme celui-ci lui permettra de récupérer sa commande lorsque sa commande sera arrivé. car il comporte un code bar que le livreur va scanner à l'arriver.

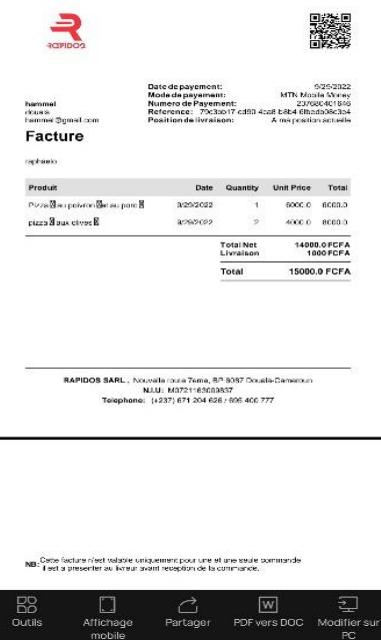


Figure 56:Facture PDF RapidosEat

CONCLUSION

En conclusion, le système n'étant pas complètement implémenté à ce stade du projet, nous nous sommes fixés comme objectif dans cette partie d'expliquer d'abord le cas d'utilisation le plus important du projet qui est « **passer une commande** » ensuite **s'inscrire** sur la plateforme et enfin **se connecter**.

CONCLUSION GENERALE

En définitive, nous avons effectué notre stage académique au sein de l'entreprise **Rapidos Sarl**. Lors de ce stage de 3 mois, il nous revenait de travailler sur le thème **analyse, conception et réalisation d'une plateforme web et mobile d'achats et de livraison de repas à domicile dans le cas de RapidosEat**. Tout au long de notre rapport le travail a été organisé en plusieurs étapes, ayant chacune un rôle bien précis, à savoir la phase d'insertion, ensuite le document d'analyse, le document de conception, puis le dossier de réalisation et enfin le guide d'utilisateur accompagné des phases de tests. La solution que nous avons développée a contribué à réaliser le service de livraison RapidosEat qui n'était encore qu'une idée sans concrétisation. Ce stage a été très enrichissant pour nous, car il nous a permis de mettre en pratique les notions reçues tout au long des 3 années de formation notamment en programmation, analyse, et conception, d'améliorer nos compétences en ce qui concerne le développement d'applications web en ajoutant à nos acquis des connaissances dans l'utilisation de nouveau langage et Framework tel que Flutter pour le langage Dart, Java natif et son Framework Spring boot ; il nous a également permis d'acquérir de l'expérience professionnelle. Notre projet est une solution importante dans la prévention et l'atténuation de la propagation du covid-19 en ceci qu'il améliore les conditions de vie des camerounais en cette période particulière, favorisant la distanciation sociale (la restriction d'un regroupement de plus de 50 personnes dans un lieu public comme restaurants ou des supermarchés). L'application n'étant achevé, nous envisageons plus tard d'intégrer les API Facebook et Google pour se connecter ou s'inscrire, de gérer le multilinguisme par rapport aux différentes cultures, et enfin mettre à point un système de discussion instantanée. A la fin de ce stage l'on se demande s'il ne serait pas important d'étendre le marché vers les pays extérieurs.



ANNEXES

BIBLIOGRAPHIE

- [1] **Mme FOFIE OLIVIA** ; 2020/2021 ; Rédaction scientifique ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [2] **Mme FOFIE OLIVIA** ; 2020/2021 ; Insertion professionnel et cahier de charges ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [3] **Mme FOFIE OLIVIA** ; 2020/2021 ; Validation de stage ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [4] **Mme FOFIE OLIVIA** ; 2020/2021 ; Cahier des charges ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [5] **Mme BELINGA Estelle** ; 2020/2021 ; Modélisation des systèmes d'information : Approche objet avec UML ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [6] **M LAMBOU LANGOUO Arnold** ; 2016/2017 ; Conception et réalisation d'une Marketplace de ventes de produits ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)
- [7] **M TCHUGWA Landry** ; 2018/2019 ; Analyse et conception d'une application mobile de ventes des produits agricoles ; IAI-CAMEROUN Centre d'Excellence Technologique Paul BIYA (Non publié)

WEBOGRAPHIE

[8] Visite de www.addictgroup.fr

- **Lien :** <https://www.addictgroup.fr/creer-service-livraison-a-domicile/>
- **Objectif :** recherche des conseils sur la création d'un service de livraison
- **Date et heure :** dimanche, 6 Juillet 2022 à 23h
- **Durée de la visite :** 5 minutes

[9] Visite de www.youtube.com

- **Lien :** <https://youtube.com/YrxBCBibVo0>
- **Objectif :** recherche sur les bases en Vue js Framework JavaScript
- **Date et heure :** Jeudi, 10 Juillet 2022 à 9h20
- **Durée de la visite :** 20 minutes

[10] Visite www.youtube.com

- **Lien :** <https://youtub.come/ixOcve5PX-Q>
- **Objectif :** recherche comment créer un formulaire avec validation et suppression de données
- **Date et heure :** vendredi, 11 Juillet 2022 à 8h20
- **Durée de la visite :** 1h 21minutes

[11] Visite www.youtube.com

- **Lien :** https://youtu.be/1ukSR1GRtMU?list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6P_BJ
- **Objectif :** recherche connaissance en Flutter Framework Dart pour débutant
- **Date et heure :** mercredi, 16 Juillet 2022 à 11h05
- **Durée de la visite :** 3h

[12] Visite www.youtube.com

- **Lien :** <https://youtu.be/vtPkZShrvXQ>
- **Objectif :** recherche comment créer une API REST utilisant PostgreSQL et JWT
- **Date et heure :** vendredi, 25 Juillet 2022 à 8h20
- **Durée de la visite :** 3h

[13] Visite www.youtube.com

- **Lien :** <https://youtu.be/5VUjP1wMqoE>
- **Objectif :** recherche connaissance de base complète en Framework Java

- **Date et heure** : jeudi, 1er Juillet 2022 à 8h20
- **Durée de la visite** : 3h 05minutes

[14] Visite www.youtube.com

- **Lien** : https://youtu.be/09Xb0A_zkzc
- **Objectif** : recherche sur le diagramme d'États-transition et d'activités en UML
- **Date et heure** : mercredi, 7 Juillet 2022 à 10h
- **Durée de la visite** : 2h

[15] Visite <https://fr.statista.com>

- **Lien** : <https://fr.statista.com/statistiques/574542/utilisateurs-de-smartphone-dans-le-monde--2019>
- **Objectif** : recherche des statistiques sur l'utilisation de smartphones au fil des années
- **Date et heure** : jeudi, 15 juillet 2022 à 15h20
- **Durée de la visite** : 10 minutes

[16] Visite <https://unctad.org>

- **Lien** : <https://unctad.org/fr/news/suite-la-covid-19-le-numerique-et-le-commerce-electronique-arrivent-un-tournant-de-leur>
- **Objectif** : recherche sur l'impact du covid sur l'expansion du monde numérique
- **Date et heure** : mardi, 20 juillet 2022 à 16h20
- **Durée de la visite** : 15 minutes

[17] Visite www.cohezio.be

- **Lien** : <https://www.cohezio.be/fr/lergonomie-informatique-ou-comment-creer-des-systemes-informatiques-faciles-utiliser#:~:text=C'est%20ce%20qu'on,laquelle%20il%20a%20%C3%A9t%C3%A9%20con%C3%A7u>
- **Objectif** : recherche la définition de l'ergonomie en informatique
- **Date et heure** : vendredi, 6 août 2022 à 23h
- **Durée de la visite** : 5 minutes

[18] Visite www.logiciels.pro

- **Lien** : <https://www.logiciels.pro/logiciel-saas/postman/#:~:text=Nous%20avons%20identifi%C3%A9%20pr%C3%A8s%20de,300%20euros%20par%20licence%20utilisateur>
- **Objectif** : recherche du prix de licence Postman

- **Date et heure** : samedi, 7 août 2022 à 00h36
- **Durée de la visite** : 5 minutes

[19] Visite <https://www.getapp.fr>

- **Lien** : <https://www.getapp.fr/software/125123/intellij-idea#:~:text=Quels%20sont%20les%20types%20de,Essai%20gratuit%3A%20Disponible>
- **Objectif** : recherche du prix de licence IntelliJ
- **Date et heure** : samedi, 7 août 2022 à 16h25

Durée de la visite : 4 minutes

[20] Visite <https://www.youtube.com>

- **Lien** : <https://youtu.be/Rx11lSvyKK8>
- **Objectif** : recherche sur la méthode Agile : Scrum
- **Date et heure** : dimanche, 15 août 2022 à 17h15
- **Durée de la visite** : 15 minutes

[21] Visite <https://fr.statista.com/>

- **Lien** : <https://fr.statista.com/statistiques/574542/utilisateurs-de-smartphone-dans-le-monde--2019>
- **Objectif** : recherche des statistiques sur l'utilisation de smartphones au fil des années
- **Date et heure** : mercredi, 18 août 2022 à 18h25
- **Durée de la visite** : 10 minutes

[22] Visite <https://unctad.org>

- **Lien** : <https://unctad.org/fr/news/suite-la-covid-19-le-numerique-et-le-commerce-electronique-arrivent-un-tournant-de-leur>
- **Objectif** : recherche sur l'impact du covid sur l'expansion du monde numérique
- **Date et heure** : Jeudi, 19 août 2022 à 16h20
- **Durée de la visite** : 15 minutes

[23] Visite <https://www.cohezio.be/fr>

- **Lien** : <https://www.cohezio.be/fr/lergonomie-informatique-ou-comment-creer-des-systemes-informatiques-faciles-utiliser#:~:text=C'est%20ce%20qu'on,laquelle%20il%20a%20%C3%A9t%C3%A9%20con%C3%A7u>
- **Objectif** : recherche la définition de l'ergonomie en informatique

- **Date et heure :** vendredi, 27 2022 à 23h
- **Durée de la visite :** 5 minutes

[24] Visite www.uv.es/

- **Lien :** <https://www.uv.es/nemiche/cursos/UML5.pdf>
- **Objectif :** recherche sur le diagramme d'États-transition et d'activités en UML
- **Date et heure :** dimanche, 29 août 2022 à 12h
- **Durée de la visite :** 2h

GLOSSAIRE

Acteur : Unité active intervenant dans le fonctionnement d'un système

Agile : méthode de gestion de projet qui met en avant la collaboration entre des équipes auto-organisées et pluridisciplinaires et leurs clients

Base de données : Ensemble structure et organise d'informations enregistrée sous forme de table et destine à être exploite.

CSS : Langage permettant de définir le style d'une page web

Flutter : est un kit de développement logiciel d'interface utilisateur open-source utilise pour développer des applications IOS MAC Windows etc.

Framework : Ensemble cohérent de fonctionnalités ou de portions de fonctionnalités conçues pour facilites le développement d'application.

HTML : Langage permettant de présenter une page web.

Maitre d'œuvre : est la personne qui réalise un projet donné par un maitre d'ouvrage.

Maitre d'ouvrage : est la personne pour qui est réalisé un projet

Micro services : ce sont des services web qui sont regroupés en API

PostgreSQL : Système de gestion de base de données relationnelle

REST : est un style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web

SGBD : Logiciel permettant la gestion des bases de données, ceci en garantissant la sécurité, la disponibilité, la pérennité, et la confidentialité.

Spring boot : C'est une déclinaison du Framework classique de **Spring** qui permet essentiellement de réaliser des micros services

Token: un dispositif matériel ou logiciel nécessaire à un utilisateur pour accéder à une application ou à un système réseau de manière plus sécurisée.

TABLE DES MATIERES

DEDICACE	I
REMERCIEMENTS	II
SOMMAIRE	III
LISTE DES FIGURES	IV
LISTE DE TABLEAUX	VI
LISTE DES ABREVIATIONS	VIII
RÉSUMÉ	IX
ABSTRACT	X
INTRODUCTION GENERALE	1
DOSSIER D'INSERTION	2
RESUME	3
INTRODUCTION	4
I. ACCEUIL ET INTÉGRATION	5
1. Accueil	5
2. Intégration	5
II. PRESENTATION DE LA STRUCTURE	5
1. Historique	5
2. Missions	6
3. Situation géographique	6
4. Activités réalisation et partenaires	7
i. Activités	7
ii. Réalisations	7
iii.	Partenaires
8	
5. Organigrammes	8

6.	Ressources de l'entreprise.....	10
i.	Ressources matérielles de Rapidos Sarl	10
ii.	Ressources logicielles de Rapidos	11
III.	PRESENTATION DU THEME	12
	CONCLUSION.....	13
	CAHIERS DES CHARGES	14
	RESUME	15
	INTRODUCTION	16
I.	CONTEXTE ET JUSTIFICATION DU PROJET	17
1.	Contexte	17
2.	Justification	17
II.	CIBLE ET BÉNÉFICIAIRES.....	17
1.	Cible.....	17
2.	Bénéficiaires	17
III.	OBJECTIFS	18
1.	Objectif général.....	18
2.	Objectifs spécifiques.....	18
IV.	EXPRESSIONS DES BESOINS	18
1.	Besoins fonctionnels	18
2.	Besoins non fonctionnels	20
V.	ESTIMATION DU COUT DU PROJET	21
1.	Ressources matérielles	21
2.	Ressources logicielles et abonnements	22
3.	Ressources humaines	25
VI.	PLANIFICATION DU PROJET	26
VII.	CONTRAINTE.....	27

1.	Coût.....	27
2.	Délai.....	27
3.	Qualité.....	27
VIII.	LIVRABLES.....	27
	CONCLUSION.....	28
	DOSSIER D'ANALYSE	29
	RESUME	30
	INTRODUCTION	31
I.	DESCRIPTION DE L'EXISTANT	32
1.	L'existant	32
2.	L'état de l'art.....	32
II.	CRITIQUES ET LIMITES DE L'EXISTANT	34
III.	PROBLEMATIQUE.....	34
IV.	PROPOSITION DE LA SOLUTION	35
V.	DEMARCHE D'ANALYSE	35
1.	Etude Comparative Entre Merise Et UML	35
2.	Choix De La Démarche D'analyse.....	36
3.	Etude Comparative Des Processus Unifiés	37
4.	Choix Du Processus	40
VI.	CAPTURE DES BESOINS FONCTIONNELS	43
A.	Diagramme de cas d'utilisation	43
	1. Formalisme	43
	2. Identifications des acteurs	44
	3. Quelques cas d'utilisation du système	45
a.	Diagramme de cas d'utilisation global du système	45
b.	Diagramme du cas d'utilisation commander repas	47

c.	Diagramme du cas d'utilisation Gérer Produit.....	48
d.	Diagramme du cas d'utilisation gérer utilisateur	49
e.	Diagramme du cas d'utilisation Gestion des véhicules.....	50
f.	Diagramme du cas d'utilisation gérer son profil	51
B.	Description textuelles de quelques cas d'utilisation	52
1.	S'inscrire.....	52
2.	Effectuer paiement.....	53
3.	Modifier information du compte	54
C.	Diagramme de séquence	55
1.	Formalisme	55
2.	Quelques diagrammes de séquence	57
i.	Authentification avec JWT token	57
ii.	Ajouter un partenaire.....	58
iii.	Inscription.....	59
D.	Diagramme d'activité.....	60
1.	Formalisme	60
2.	Quelques diagrammes d'activité	62
i.	Modifier son compte(Partenaire).	62
ii.	Passer commande	63
iii.	Accepter Commande	64
E.	Diagramme de communication	65
1.	Formalisme	65
2.	Quelques diagrammes de communication	66
i.	Ajouter Produit.....	66
ii.	Effectuer payement	67
iii.	Rechercher restaurant	68

CONCLUSION.....	69
DOSSIER DE CONCEPTION	70
RESUME	71
INTRODUCTION	72
I. DIAGRAMME DE CLASSE	73
1. Formalisme	73
2. Diagramme de classe du système	75
II. DIAGRAMME DE PACKAGE	75
1. Formalisme	76
2. Diagramme de package du système	76
III. DIAGRAMME D'ETATS-TRANSITION	77
1. Formalisme	77
2. Diagramme d'états-transition du système	78
IV. DIAGRAMME D'OBJET	79
1. Formalisme	79
Diagramme d'objet du système	79
CONCLUSION.....	81
DOSSIER DE RÉALISATION	82
RESUME	83
INTRODUCTION	84
I. ÉQUIPES PROJET ET CHOIX DES OUTILS.....	85
1. Equipe projet.....	85
2. Présentation des langages de programmation et Framework.....	85
II. PHASE D'IMPLÉMENTATION	86
1. Architecture logique de déploiement	86
2. Diagramme de composant.....	87

i.	Formalisme	87
ii.	Diagramme de composant du système	88
3.	Aspect sécurité de l'application	89
i.	Sécurité des connections.....	89
ii.	Sécurité du service web et mobile	89
iii.	Sécurité du mot de passe	
	89	
iv.	Sécurité via loggers	
	89	
III.	PHASE DE DÉPLOIEMENT.....	89
1.	Architecture Physique de déploiement	89
2.	Diagramme de déploiement	90
i.	Formalisme	90
ii.	Diagramme de déploiement du système	92
CONCLUSION.....		93
TESTS DE FONCTIONNALITÉS.....		94
RESUME		95
INTRODUCTION		96
I.	TESTS FONCTIONNELS	97
1.	Test d'inscription	98
2.	Test de connexion (Authentification)	99
3.	Test de catégorie	100
4.	Test de produits.....	101
5.	Test d'ajout de restaurant.....	101
II.	TESTS NON-FONCTIONNELS.....	102
III.	TEST DE PERFORMANCE	102
CONCLUSION.....		103

GUIDE D'UTILISATION.....	104
RESUME	105
INTRODUCTION	106
I. GUIDE D'INSTALLATION.....	107
II. GUIDE D'UTILISATION.....	107
CONCLUSION.....	114
CONCLUSION GENERALE.....	115
ANNEXES	A
BIBLIOGRAPHIE.....	B
WEBOGRAPHIE	C
GLOSSAIRE.....	G
TABLE DES MATIERES	H