



République Algérienne Démocratique et Populaire



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique

Mémoire de Master

en Informatique

Spécialité : Ingénierie des Systèmes d'Information et des Logiciels

Thème

Réalisation d'un service de livraison de plats cuisinés

Encadré par

— DJOUABRI ABDERREZAK

Réalisé par

— HANI HOUSSAM

— DENDANI YAKOUB

2021/2022

Remerciements

Nous tenons à remercier le bon Dieu de nous avoir donné la force, la patience et la persévérance de réaliser ce modeste travail.

D'abord, nous tenons à exprimer toute notre gratitude et notre parfaite considération à notre encadreur, le docteur Djouabri Abderrezak , pour sa disponibilité, sa patience, ses conseils et son orientation. Nos vifs remerciements iront aussi aux membres de jury qui nous feront l'honneur de juger notre travail.

Nos remerciements sont destinés de même à, tous nos enseignants du département informatique de l'université de Bouira Akli Mohand Oulhadj. Nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Enfin, nous tenons à adresser nos sincères remerciements ainsi que notre gratitude la plus dévouée, pour leur dévouement et surtout pour leur aide qu'ils nous ont accordés, à nos chers parents, que le Bon Dieu nous les garde aussi longtemps.

Dédicaces

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,

A mes chers frères, pour leur appui et leur encouragement,

A toute ma famille pour leur soutien tout au long de mon parcours universitaire,

Merci d'être toujours là pour moi.

Dendani Yakoub

Dédicaces

A Mes Très chers Parents

Je dédie ce mémoire à mes parents, pour l'amour qu'ils m'ont toujours donné, leurs encouragements et toute l'aide qu'ils m'ont apportée durant mes études.

Aucun mot, aucune dédicace ne pourrait exprimer mon respect, ma considération, et mon amour pour les sacrifices qu'ils ont consentis pour mon instruction et mon bien-être.

Trouvez ici, chère mère et cher père, dans ce modeste travail, le fruit de tant de dévouements et de sacrifices ainsi que l'expression de ma gratitude et de mon profond amour.

Puisse Dieu leur accorder santé, bonheur, prospérité et longue vie afin que je puisse un jour combler de joie leurs vieux jours.

A Mes Sœurs

Je leur dédie ce travail pour tous les sacrifices qu'elles n'ont cessé de m'apporter tout au long de mes années d'études.

Que Dieu leur apporte le bonheur, les aide à réaliser tous leurs vœux et leur offre un avenir plein de succès.

Hani Houssam.

ملخص

أدى تطور أدوات الاتصال وكذلك سرعات الاتصال إلى العديد من التغييرات في حياة الناس اليومية ، ويميل الناس إلى استخدام هواتفهم الذكية المتصلة كثيرًا. الهدف من هذا المشروع هو إنشاء خدمة توصيل وجبات جاهزة باستخدام مجموعة من تطبيقات الهاتف الذكي. يتم تنفيذ هذه الخدمة باستخدام أندرويد ستوديو ، كوتلين كلغة برمجة و فاير بايز كقاعدة بيانات. **الكلمات المفتاحية** : خدمة توصيل الطعام ، توصيل ، البيع عبر الإنترنت ، الشراء عبر الإنترنت ، تطبيق.

Résumé

Le développement des outils de communication ainsi que les vitesses de communication entraînent de nombreux changements dans la vie quotidienne des gens, les gens ont tendance à beaucoup utiliser leurs smartphones connectés. Le but de ce projet est de réaliser un service de livraison de plats cuisinés sous un ensemble d'applications mobile. L'implémentation de ce service se fait en utilisant la plateforme Android Studio, Kotlin comme langage de programmation et Firebase comme base de données.

Mots clés : service de livraison de plats cuisinés, livraison, vente en ligne, achat en ligne, application.

Abstract

The development of communication tools as well as communication speeds lead to many changes in people's daily lives, people tend to use their connected smartphones a lot. The goal of this project is to create a ready meal delivery service using a set of mobile applications. The implementation of this service is done using the Android Studio platform, Kotlin as the programming language and Firebase as the database.

Key words : food delivery service, delivery, online sale, online purchase, application.

Table des matières

Table des matières	i
Table des figures	v
Liste des tableaux	viii
Liste des abréviations	x
1 Etude de l'existant	3
1.1 Introduction	3
1.2 Définition générale	3
1.2.1 Les restaurants	3
1.2.2 Les différents types de restauration	4
1.3 Présentation d'un restaurant sélectionné comme modèle	5
1.3.1 Organigramme des restaurants	5
1.3.2 Les personnels et leurs rôles	6
1.3.3 Situation informatique des restaurants	6
1.3.4 Solution proposée	7
1.4 Historique de livraison	7
1.5 Les avantages et les inconvénients de livraison des repas	8
1.5.1 Pour les clients	8
1.5.2 Pour les restaurants	8
1.5.3 Pour les livreurs	9
1.6 Les applications de livraison des repas	9

1.6.1	Uber Eats	9
1.6.2	Deliveroo	10
1.6.3	Just Eat	10
1.6.4	Fast Delivery	11
1.7	Présentation de notre projet.	12
1.7.1	Introduction	12
1.7.2	Planification du projet	12
1.7.3	Problématique	15
1.7.4	Solution envisagée	15
1.7.5	Les difficultés rencontrées	16
1.8	Les points forts des applications des livraison	16
1.9	Conclusion	17
2	Analyse et spécification des besoins	18
2.1	Introduction	18
2.2	Méthodologie de conception	18
2.2.1	UML	18
2.3	Processus unifié (UP)	19
2.3.1	Cycles de vie d'un processus unifié	19
2.3.2	Activités du processus	20
2.4	Classification des diagrammes	21
2.4.1	Les diagrammes structurels	21
2.4.2	Les diagrammes de comportement	21
2.5	Les diagrammes utilisés	22
2.5.1	Le diagramme de cas d'utilisation	22
2.6	Spécification des besoins fonctionnels	23
2.6.1	Identification des acteurs	23
2.6.2	Identification des cas d'utilisation	23
2.6.3	Diagramme de cas d'utilisation générale	26
2.7	Spécification des besoins non fonctionnels	27
2.8	Scénarios et cas d'utilisations	28
2.8.1	Scénario et cas d'utilisation raffiné relatif à S'authentifier pour client	28
2.8.2	Scénario et cas d'utilisation raffiné relatif à Consulter les magasins .	29

2.8.3	Scénario et cas d'utilisation raffiné relatif à Passer une commande	31
2.8.4	Scénario et cas d'utilisation raffiné relatif à Créer un compte pour vendeur	34
2.8.5	Scénario et cas d'utilisation raffiné relatif à Consulter les nouvelles commandes	36
2.8.6	Diagramme de cas d'utilisation « Suspendre la réception des commandes »	37
2.8.7	Scénario et cas d'utilisation raffiné relatif à Consulter les commandes en attente	39
2.8.8	Scénario et cas d'utilisation raffiné relatif à Scanner codeQR	40
2.9	Conclusion	41
3	Conception	42
3.1	Introduction	42
3.2	Les diagrammes utilisés	42
3.2.1	Diagramme de séquence	42
3.2.2	Diagramme de classe	43
3.3	Description de la vue statique	44
3.3.1	Diagramme des classes	44
3.3.2	Modélisation de la base de données « Firestore »	45
3.4	Description de la vue dynamique	53
3.4.1	Diagramme de séquence relatif à « S'authentifier »	53
3.4.2	Diagramme de séquence relatif à « Ajouter un article au panier »	54
3.4.3	Diagramme de séquence relatif à « Passer une commande »	55
3.4.4	Diagramme de séquence relatif à « Créer un compte vendeur »	56
3.4.5	Diagramme de séquence relatif à « Consulter une nouvelle commande »	57
3.4.6	Diagramme de séquence relatif à « Suspendre la réception des commandes »	58
3.4.7	Diagramme de séquence relatif à « Scanner un code QR »	59
3.5	Conclusion	60

4	Réalisation	61
4.1	Introduction	61
4.2	Environnement de travail	61
4.2.1	Environnement matériel	61
4.2.2	Environnement logiciel	62
4.2.3	Les types d'applications	64
4.3	Les outils utilisés	66
4.3.1	Android Studio	66
4.3.2	Java Development Kit (JDK)	66
4.3.3	Android Software Development Kit (SDK)	67
4.3.4	Kotlin	67
4.3.5	Firebase	67
4.3.6	Android Product Flavors	68
4.3.7	MVVM (Model View View-Model)	68
4.3.8	Material Design	69
4.3.9	API de géolocalisation de Google	70
4.3.10	Dagger Hilt	71
4.3.11	Flaticon	71
4.3.12	Zxing	71
4.3.13	Code QR	71
4.4	Interfaces des applications	72
4.4.1	Application « Garanda Food »	72
4.4.2	Application « Garanda Food Vendeurs » :	81
4.4.3	Application « Garanda Food Livreurs » :	87
4.5	Conclusion	91
	 Conclusion générale et perspectives	 92
	 Bibliographie	 93
	 Annexe	 95

Table des figures

1.1	Organigramme d'organisation du restaurant Babylon	6
1.2	Organigramme de réorganisation proposé	7
1.3	Uber Eats - Play Store	9
1.4	Deliveroo - Play Store	10
1.5	Just-Eat - Play Store	11
1.6	Fast Delivery - Play Store	11
1.7	Aperçu sur le système	12
2.1	Cycles de vie d'un processus unifié	20
2.2	Formalisme de base de représentation d'un cas d'utilisation	22
2.3	Diagramme de cas d'utilisation générale	26
2.4	Diagramme de cas d'utilisation «S'authentifier»	28
2.5	Diagramme de cas d'utilisation «Consulter les magasins»	29
2.6	Diagramme de cas d'utilisation «Passer une commande»	31
2.7	Diagramme de cas d'utilisation «Créer un compte vendeur»	34
2.8	Diagramme de cas d'utilisation « Consulter les nouvelles commandes »	36
2.9	Diagramme de cas d'utilisation « Suspendre la réception des commandes »	37
2.10	Diagramme de cas d'utilisation « Consulter les commandes en attente »	39
2.11	Diagramme de cas d'utilisation « Scanner un code QR »	40
3.1	Exemple de diagramme de séquence	43
3.2	Exemple de diagramme de classe	43
3.3	Diagramme des classes	44
3.4	Répartition des collections sur Firestore	45

3.5	Diagramme de séquence relatif à « S'authentifier »	53
3.6	Diagramme de séquence relatif à « Ajouter un article au panier »	54
3.7	Diagramme de séquence relatif à « Passer une commande »	55
3.8	Diagramme de séquence relatif à « Créer un compte vendeur »	56
3.9	Diagramme de séquence relatif à « Consulter une nouvelle commande » . .	57
3.10	Diagramme de séquence relatif à «Suspendre la réception des commandes»	58
3.11	Diagramme de séquence relatif à « Scanner un code QR »	59
4.1	Logo Android OS	63
4.2	Les composants du système Android	64
4.3	Exemple de type native et de type web sur une même application	65
4.4	Logo Android Studio	66
4.5	Logo JDK	66
4.6	Logo SDK	67
4.7	Logo Kotlin	67
4.8	Logo Firebase	68
4.9	Le modèle de conception MVVM	69
4.10	Logo Material Design	69
4.11	Logo Google Maps	70
4.12	Première étape de configuration de la carte Google Maps	70
4.13	Logo Flaticon	71
4.14	Exemple de code QR	72
4.15	Interface Connexion d'un client	72
4.16	Interface Inscription d'un client	73
4.17	Interface Accueil	74
4.18	Interface Détails magasin	75
4.19	Interface Détails article	75
4.20	Interface Paniers	76
4.21	Interface Passer une commande	76
4.22	Interface Maps	77
4.23	Interface Mes commandes	78
4.24	Interface Détails commande	78
4.25	Interface Parcourir magasins	79

4.26	Interface Compte	80
4.27	Interface Mon compte	80
4.28	Interface Créer compte Vendeur	81
4.29	Interface Principale	82
4.30	Menu déroulant	82
4.31	Interface Nouvelles commandes	83
4.32	Interface Détails commande	83
4.33	Interface Commandes prêtes	84
4.34	Interface Historique commandes	84
4.35	Interface Menu	85
4.36	Interface Ajouter un article	85
4.37	Interface Code QR	86
4.38	Interface Créer compte Livreur	87
4.39	Interface Accueil	88
4.40	Interface Commandes en attentes	89
4.41	Interface Détails commande	90
4.42	Interface Commande terminée	90
4.43	Menu déroulant	91
4.44	Configuration de la machine	95
4.45	Configuration du terminal mobile	96
4.46	Logo Garanda Food	97
4.47	Logo Garanda Food Vendeurs	97
4.48	Logo Garanda Food Livreurs	97

Liste des tableaux

1.1	Les personnels et leurs rôles	6
1.2	Les avantages et les inconvénients de livraison des repas pour les clients . .	8
1.3	Les avantages et les inconvénients de livraison des repas pour les restaurants	8
1.4	Les avantages et les inconvénients de livraison des repas pour les livreurs .	9
1.5	Les tâches de Garanda Food	13
1.6	Les tâches de Garanda Food Vendeurs	14
1.7	Les tâches de Garanda Food Livreurs	14
1.8	Les points forts des applications de livraison	17
2.1	Raffinement de cas d'utilisation « S'authentifier »	29
2.2	Raffinement de cas d'utilisation « Consulter les magasins »	30
2.3	Raffinement de cas d'utilisation « Rechercher un magasin »	31
2.4	Raffinement de cas d'utilisation « Ajouter des articles au panier. »	32
2.5	Raffinement de cas d'utilisation « Passer une commande. »	33
2.6	Raffinement de cas d'utilisation « Créer un compte » pour vendeur	35
2.7	Raffinement de cas d'utilisation « Consulter les nouvelles commandes » . .	37
2.8	Raffinement de cas d'utilisation « suspendre la réception des commandes ». 38	
2.9	Raffinement de cas d'utilisation « Consulter les commandes en attente » .	40
2.10	Raffinement de cas d'utilisation « Scanner un code QR »	41
3.1	Description relative aux attributs de la collection « Client »	46
3.2	Description relative aux attributs de la collection « Magasin »	47
3.3	Description relative aux attributs de la collection « Article »	47
3.4	Description relative aux attributs de la collection « OptionsArticle »	48

3.5	Description relative aux attributs du type « ChoixOptionArticle »	48
3.6	Description relative aux attributs de la collection « Panier »	49
3.7	Description relative aux attributs de la collection « Commande »	50
3.8	Description relative aux attributs du type « Adresse »	50
3.9	Description relative aux attributs de la collection « DemandeOuverture- Compte »	51
3.10	Description relative aux attributs de la collection « Livreur »	52
3.11	Description relative aux attributs de la collection « Vendeur »	52

Liste des abréviations

- API : Application Programming Interface.
- JDK : Interface Homme Machine
- SDK : Software Development Kit
- UML : Unified Modeling Language
- UP : Processus Unifié

Introduction générale

Tout en vivant dans un monde dynamique, les gens ont parfois du mal à gérer des tâches simples comme acheter de la nourriture ou préparer le dîner. Heureusement, les consommateurs peuvent désormais résoudre ces tâches en quelques clics sur leur téléphone portable. Les smartphones sont devenus leur outil pour obtenir tout ce qu'ils veulent à leur porte grâce aux services à la demande. En effet, la technologie numérique remodèle le marché de la livraison dont le secteur de la restauration qui ne fait pas exception.

Les applications de livraison de nourriture offrent aux consommateurs la possibilité de commander de la nourriture dans un large éventail de restaurants, leur permettant de comparer les menus, les prix et les avis d'autres utilisateurs de manière simple et rapide.

En effet, des études antérieures ont prouvé que les consommateurs utilisent plutôt les services en ligne en raison de leur rapidité, précision et facilité d'utilisation. La commodité est certainement l'une des motivations les plus fortes pour que les consommateurs intensifient leurs relations avec n'importe quelle plate-forme de service. Sans surprise, ces services de livraison de nourriture sont les plus populaires parmi les milléniaux. Dans ce cadre intervient notre projet de fin d'étude visant à utiliser les connaissances acquises au cours de notre formation à l'Université Akli Mohand Oulhadj dont l'objectif est de développer une plateforme mobile de livraison des repas cuisinés sur la Wilaya de Bouira.

Notre mémoire est subdivisé en quatre chapitres, Le premier présente les définitions générales concernant les restaurants, le service de livraison et les applications les plus populaires dans ce domaine, ensuite on explique la problématique et la solution proposé.

Le second chapitre s'attèle sur l'analyse des besoins et spécification, nous commencerons par comprendre le contexte du système, déterminer les principaux cas d'utilisation, déterminer les besoins fonctionnels et les besoins non fonctionnels.

Le troisième chapitre est consacré à la compréhension du système et à obtenir une spécification, une analyse et une conception détaillées des cas d'utilisation.

Le quatrième chapitre est destiné à exposer la partie réalisation de notre application où nous allons détailler l'architecture utilisée.

Enfin nous terminerons par une conclusion générale qui résume notre travail avec des perspectives pour les futurs travaux sur ce projet.

Chapitre 1

Etude de l'existant

1.1 Introduction

Le marché de la livraison des repas à domicile n'est pas nouveau, mais il connaît actuellement une croissance spectaculaire. En fait plusieurs restaurants et fast-foods essayent d'assurer une distribution locale de leurs repas depuis quelques années, ce qui explique l'apparition de nouvelles plateformes algériennes spécialisées dans la livraison des repas. Et c'est le cas pour notre projet qui propose un ensemble d'applications mobile pour automatiser le processus de livraison des repas, en assurant une meilleure expérience aux clients, aux restaurateurs et aux livreurs.

1.2 Définition générale

Pour que notre recherche soit plus précise on doit savoir ce que signifie un restaurant est quels sont les types des restaurants qui peut s'abonner à nos services.

1.2.1 Les restaurants

" Un restaurant est un établissement de commerce où l'on sert des plats préparés et des boissons à consommer sur place, en échange d'un paiement. La nourriture y est généralement préparée par un chef cuisinier. Le terme couvre une multiplicité de lieux et une grande diversité des types de cuisines, tant locales qu'étrangères. Les restaurants sont parfois le dispositif réservé au service des repas au sein d'une plus grande entité (hôtel, université, aéroport...), on parle alors de restaurant collectif par opposition au

site de cuisine. Ils peuvent aussi être associés à une activité de traiteur ou d'épicerie. Le restaurant offre des conditions de confort plus ou moins importantes, et la restauration est dite « rapide » quand le client peut commander et manger en quelques minutes ou dizaines de minutes, éventuellement debout ".[1]

1.2.2 Les différents types de restauration

- Restauration traditionnelle

La restauration traditionnelle Cette branche comporte des établissements très variés et se compose essentiellement de petites entreprises individuelles. C'est le type d'établissement où chaque chef d'entreprise peut laisser libre cours à sa créativité et à son imagination. Ils sont les héritiers de la tradition gastronomique.

- A- Les restaurants classiques
- B- Les restaurants d'hôtels
- C- Les restaurants de tourisme

- Restauration collective

La restauration de collectivité repose sur des contraintes précises : nourrir un grand nombre de gens pour un prix extrêmement bas.

- A- La restauration d'entreprise
- B- La restauration scolaire
- C- La restauration hospitalière
- D- La restauration pénitenciers, des armées.

- Restauration rapide

La restauration rapide La restauration rapide est le secteur qui a connu la plus forte croissance ces vingt dernières années.

- A-Le fast-food
- B-Le restauroute
- C-La cafétéria

E-La sandwicherie

F-La restauration automatique

- Restauration à thèmes

" La restauration à thème est un mode de restauration dont l'offre de prestation, les menus et la décoration de l'établissement sont influencés par un thème donné ".[2]

Dans ce projet de fin de cycle notre cas d'étude repose sur la restauration rapide et la restauration traditionnelle car ce sont les types les plus populaires dans la région, donc pour offrir une meilleure expérience aux clients, les vendeurs ont besoins d'un service de livraison à domicile pour maintenir les liens avec leurs clients et améliorer leurs rentabilités.

1.3 Présentation d'un restaurant sélectionné comme modèle

Comme nous ciblons le plus grand nombre de restaurants possible et parce que la plupart des restaurants sont similaires nous avons choisi le restaurant « Babylon » comme modèle Babylon est un restaurant de type restauration rapide, situé à Bouira (Rue Guemraoui Gharbi), il propose une variété de plats et de sandwiches, sur son menu on peut trouver des Hamburgers, Kebab, Pizza, Tacos, ou encore Shawarma, en plus des différentes boissons.

Babylon accueille ses clients tous les jours de 10h à 21h sauf le vendredi, dans un endroit très sympa et calme, il offre un service sur place ou à emporter, mais la livraison à domicile n'est pas encore disponible, et c'est le cas pour la plupart des restaurants de la région.

1.3.1 Organigramme des restaurants

Les organigrammes varient selon la taille de l'établissement : petit restaurant, grand restaurant ou restaurant gastronomique. Dans notre cas nous présentons un restaurant de petite taille : le Babylon, organisé comme suit :

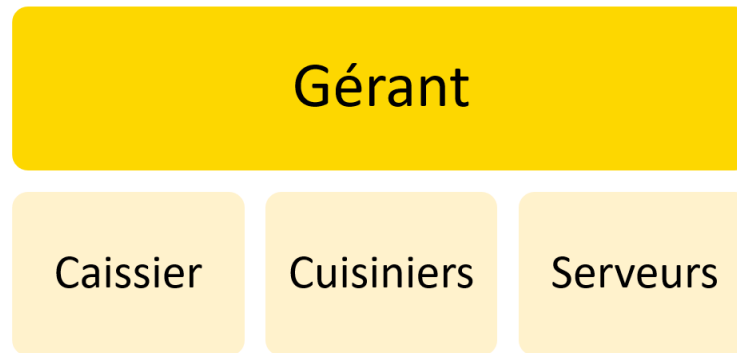


FIGURE 1.1 – Organigramme d’organisation du restaurant Babylon

1.3.2 Les personnels et leurs rôles

Le tableau 1.1 suivant présente les personnels du restaurant et leurs rôles

<i>Personnel</i>	<i>Rôle</i>
Gérant	<ul style="list-style-type: none"> - Il organise et coordonne. - Il est responsable de « l'économie » de sa cuisine. - Il est responsable de la formation de son personnel. - Il organise le travail. - Gérer les ressources financières. - S'assurer de la satisfaction de la clientèle.
Cuisinier	<ul style="list-style-type: none"> - Il prépare les plats. - Il vérifie les marchandises nécessaires
Caissier	<ul style="list-style-type: none"> - Il se charge de la caisse. - Il gère le service à emporter.
Serveur	<ul style="list-style-type: none"> - Garder les chaises et les tables propres. - Servir au client les aliments commandés.

TABLE 1.1 – Les personnels et leurs rôles

1.3.3 Situation informatique des restaurants

En remarque que la plupart des restaurants de la région ne dispose pas d'un système informatique pour gérer les commandes, ou ils ont seulement un logiciel de facturation,

les autres processus se déroulent de manière traditionnelle ce qui entraîne une mal gestion des commandes et une perte de temps.

1.3.4 Solution proposée

Nous proposons aux restaurateurs d'abonner à un service de livraison externe. Ce service va assurer la gestion des commandes et la livraison des repas, le restaurateur s'occupe seulement par la préparation

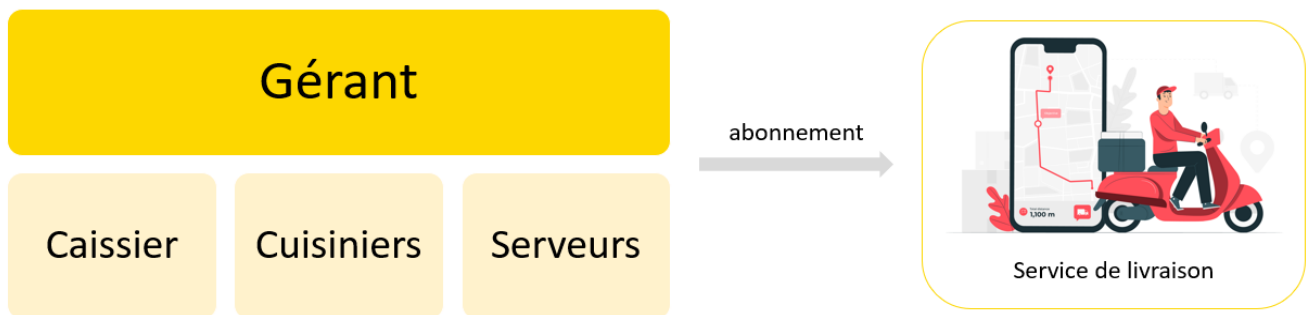


FIGURE 1.2 – Organigramme de réorganisation proposé

1.4 Historique de livraison

Il y a de cela quelques centaines d'année, la seule forme de livraison qui existait était la poste. C'est en 1763 que s'établit le premier système de poste au Canada. Au cours des années, ce système s'est répandu et a beaucoup évolué en passant par les wagons de poste en 1854 au premier vol postal en 1918. La livraison en ligne est apparue tout récemment et devient de plus en plus populaire dans le monde. au début des années 1990, de grandes compagnies déjà existantes comme Ikea, La Baie ou encore Apple on put créer leur propre site pour faciliter la distribution des produits au niveau mondial. C'est par Internet que le concept de vente en ligne est apparu. En effet, des compagnies telles qu'Amazon en juillet 1994 ou Ebay en septembre 1995 ont vu le jour par le biais de cette plateforme. Finalement, les produits de vente en ligne se sont diversifiés. Suite à la vente d'objet, la vente de vêtements s'est ensuite développée. Zaful et Romwe sont deux compagnies de vente en ligne qui propose des produits qu'on trouve rarement en magasins. Puis, ce fut le tour de la nourriture.[3]

1.5 Les avantages et les inconvénients de livraison des repas

1.5.1 Pour les clients

<i>avantages</i>	<i>inconvénients</i>
<ul style="list-style-type: none"> • Un gain de temps. • Une simplicité de fonctionnement. • Un large choix de plats au menu. • La rapidité. • La possibilité de suivre sa commande. 	<ul style="list-style-type: none"> • La livraison est peut-être coûteuse. • Temps estimé pour l'arrivée de la commande n'est pas vraiment fixe. • Si le client est loin, la nourriture peut arriver froide. • Il est difficile de juger un repas par une simple photographie de celui-ci.

TABLE 1.2 – Les avantages et les inconvénients de livraison des repas pour les clients

1.5.2 Pour les restaurants

<i>avantages</i>	<i>inconvénients</i>
<ul style="list-style-type: none"> • Augmentation de la zone de chalandise du restaurant • Gagner plus de clients. • Augmentation de la rentabilité. • L'amélioration de la visibilité auprès des clients 	<ul style="list-style-type: none"> • Risque de générer des perturbations liées au stress de l'attente. • La réorganisation des menus. • Frais du packaging et d'emballage. • Risque d'endommager les commandes par les livreurs.

TABLE 1.3 – Les avantages et les inconvénients de livraison des repas pour les restaurants

1.5.3 Pour les livreurs

<i>avantages</i>	<i>inconvénients</i>
<ul style="list-style-type: none"> • Le salaire est proportionnel aux heures de travail ou au nombre de commandes livrées. • Les horaires flexibles : le livreur se connecte quand il le souhaite sur l'application. • Un parfait moyen de concilier études et travail (pour un étudiant). • Indépendance financière. 	<ul style="list-style-type: none"> • Pas de statut de salarié. • Les risques des accidents routiers. • Contraintes de déplacement et d'isolement. • Changements climatiques.

TABLE 1.4 – Les avantages et les inconvénients de livraison des repas pour les livreurs

1.6 Les applications de livraison des repas

À l'échelle mondiale les application de livraison des repas sont largement diffusées. Nous vous proposons une liste des applications les plus populaires :

1.6.1 Uber Eats

" Uber Eats est un 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 de Uber Eats auprès des restaurants partenaires et sont livrées par des coursiers indépendants (vélo, scooter, etc.). L'entreprise opère dans le monde entier : Amérique du Nord, Amérique du Sud, Europe, Australie, Asie et dans quelques villes d'Afrique ". [4]



FIGURE 1.3 – Uber Eats - Play Store

1.6.2 Deliveroo

" Cette application appartient à la société britannique Deliveroo, fondée en 2013. La plateforme de livraison de repas est présente dans plusieurs pays Europe, d'Asie et en Australie

Les commandes de repas via Deliveroo sont possibles depuis l'application mobile et le site internet. Sur la plateforme, les restaurants partenaires à Deliveroo sont triés par délai de livraison ". [5]



FIGURE 1.4 – Deliveroo - Play Store

1.6.3 Just Eat

" Just Eat est une plateforme qui met en relation des lieux de restauration et des clients. Just Eat. Créée en 1998 en Grande-Bretagne, l'application est cotée à la bourse de Londres. Le principe de commande est simple, Just Eat est accessible depuis une application mobile et le site internet. Vous commandez un plat dans l'un des restaurants inscrits sur cette plateforme, la livraison est faite dans les 45 minutes qui suivent la commande. Les livraisons sont faites à moto ou à vélo. Cela permet une rapidité et une efficacité des livreurs. Les règlements sont possibles depuis votre compte Just Eat ou en espèce avec le livreur. Lorsque vous ouvrez votre compte sur cette plateforme, il faudra renseigner votre ville pour avoir la liste des restaurants ouverts et inscrits sur Just Eat près de chez vous Just Eat (anciennement Allo Resto), n'a pas une flotte de livreurs. Le service met simplement en relation les restaurants disposant de leurs propres livreurs avec les clients ".[6]



FIGURE 1.5 – Just-Eat - Play Store

Nous avons essayé de chercher sur Le marché de livraison des repas en Algérie, Il était difficile de trouver des études. Mais en général la livraison existe dans les grandes villes mais limité.

1.6.4 Fast Delivery

Fast Delivery est une application de livraison qui permet d'acheter et de se faire livrer n'importe quel produit à Alger en 50 minutes. Les utilisateurs peuvent commander dans tous les restaurants et toutes les boutiques d'Alger, ils peuvent également livrer les courses, tels que fruits, légumes, produits laitiers, œufs, viandes et plus.



FIGURE 1.6 – Fast Delivery - Play Store

1.7 Présentation de notre projet.

1.7.1 Introduction

L'idée du projet se résume sur la réalisation d'un service de livraison des plats cuisinés sous un ensemble d'applications mobile basé sur le cloud qui fonctionnent de manière cohérente et qui partagent la même base de données, permettant aux clients de passer des commandes de nourriture sans se déplacer et à n'importe quel moment auprès plusieurs restaurants et fast-food partenaire de la région, et la livraison se fait par des coursiers indépendants inscrits sur notre système.

1.7.2 Planification du projet

L'organigramme suivant présente un aperçu sur le système en générale :

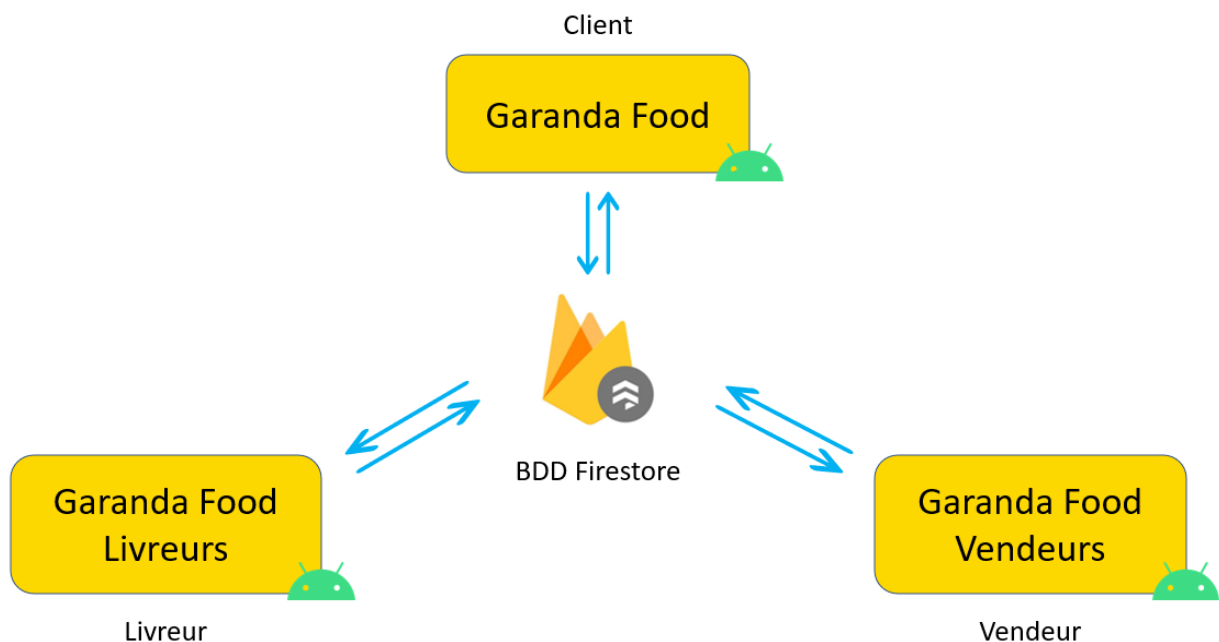


FIGURE 1.7 – Aperçu sur le système

Nous avons sélectionné les noms des applications comme suit :

- L'application destiné aux clients nommer : Garanda Food.
- L'application destiné aux restaurants nommer : Garanda Food Vendeurs.
- L'application destiné aux livreurs nommer : Garanda Food Livreurs.

Chacune des trois applications doit effectuer un ensemble de tâches pour un utilisateur particulier

Le tableau 1.5 représente les tâches de Garanda Food :

<i>Application : Garanda Food</i>
L'utilisateur
Le client : " Personne qui reçoit d'une entreprise, contre paiement, des fournitures commerciales ou des services ". [7]
Tâches
<p>A1- Créer un compte pour sauvegarder les informations du client.</p> <p>A2- Consulter les différents magasins selon la ville du client.</p> <p>A3- Rechercher les magasins par leurs types.</p> <p>A4- Parcourir les magasins selon leurs catégories de cuisine.</p> <p>A5- Consulter un magasin et voir son menu.</p> <p>A6- Ajouter des articles au panier.</p> <p>A7- Localiser le client sur le Maps.</p> <p>A8- Passer une commande et suivre son statut.</p> <p>A9- Consulter les commandes effectuées.</p> <p>A10- Ajouter des magasins au favoris.</p> <p>A11- Évaluer un magasin ou un livreur.</p> <p>A12- Contacter l'assistance pour plus d'information.</p>

TABLE 1.5 – Les tâches de Garanda Food

Le tableau 1.6 représente les tâches de Garanda Food Vendeurs :

<i>Garanda Food Vendeurs</i>
L'utilisateur
Vendeur : Propriétaire d'un restaurant.
Tâches
<p>A1- Créer une demande d'ouverture de compte Vendeur.</p> <p>A2- Personnaliser son magasin (l'image, la position géographique, coordonnées ...).</p>

Page suivante

Tâches
A3- Gérer les horaires d'ouverture du magasin.
A4- Ajouter ou modifier des articles sur le menu du magasin.
A5- Consulter les nouvelles commandes et voir leurs détails.
A6- Communiquer avec les clients et les livreurs pour avoir des informations.
A7- Préparer des commandes et changer leurs statuts.
A8- Consulter l'historique des commandes terminées.
A9- Contacter l'assistance pour avoir de l'aide.

TABLE 1.6 – Les tâches de Garanda Food Vendeurs

Le tableau 1.7 représente les tâches de Garanda Food Livreurs :

<i>Garanda Food Livreurs</i>
L'utilisateur
Le livreur : " Employé(e) qui porte chez l'acheteur la marchandise vendue ".[7]
Attentes
A2- Personnaliser son compte (photo de profile, moyen de livraison, coordonnées...).
A3- Consulter les commandes en attentes et voir leurs détails.
A4- Localiser un client ou un magasin sur le Maps.
A5- Contacter un client ou un magasin par appel ou par message.
A6- Accepter une commande et commencer la livraison.
A7- Scanner le code bar d'un magasin ou d'un client pour changer le statut d'une commande.
A8- Consulter les commandes terminées.
A9- Changer son statut « en service » ou « hors service ».

TABLE 1.7 – Les tâches de Garanda Food Livreurs

Dans notre projet, il existe aussi un autre acteur, c'est l'administrateur qui gère l'ensemble des applications et traite les demandes des vendeurs et des livreurs, il effectue d'autres tâches aussi mais on va concentrer seulement sur les acteurs précédents afin de limiter l'idée générale de notre projet.

1.7.3 Problématique

Les plateformes de livraison de plats cuisinés n'existent pas dans notre région, les restaurants et les fast-foods travaillent toujours de manière traditionnelle. Donc après analyse de l'existant nous avons pu faire sortir les critiques suivantes :

- Le client doit déplacer pour commander la nourriture.
- Le temps perdu d'un client lorsqu'il attend la préparation de sa commande.
- Parfois le client ne trouve pas de places chez le restaurant.
- Les restaurants qui se situe loin des grands boulevards n'accueillent pas beaucoup de clients.
- L'absence des prix et du menu dans certains restaurants.
- La confrontation entre les commandes lorsque le restaurant soit chargé.
- Le client ne peut pas savoir si un repas est toujours disponible ou non.

1.7.4 Solution envisagée

Après l'étude que nous avons effectuée, et selon les besoins des citoyens de la région nous opterons pour un ensemble d'applications avec lesquelles nous pourrions automatiser le processus de commande de nourriture. Afin d'y remédier à tous ses problèmes, nous avons assigné à notre étude les objectifs suivants :

- Gagner du temps pour le client et pour le personnel du restaurant.
- Améliorer la visibilité des restaurants auprès des clients.
- Gestion et organisation des commandes.
- Assurer une livraison rapide et fiable des commandes.
- Possibilité de suivre les commandes.
- Améliorer la visibilité des restaurants auprès des clients.
- Améliorer la rentabilité des restaurants.
- Créer une source de revenu pour les livreurs.

1.7.5 Les difficultés rencontrées

Lors de la conception et réalisation du projet on a rencontré quelques difficultés qu'on peut citer ci-dessous :

- Le passage d'une seule application à plusieurs applications sous un même projet Android.
- L'implémentation de la base de données pour stocker et traiter les données de différentes applications.
- La récolte des informations de plusieurs restaurants et fast-foods.
- Le suivi des commandes en temps réel.
- Le traitement des données de géolocalisation.
- Assurer des interfaces simples et faciles à utiliser.
- Le choix et l'unification du design pour toutes les applications.

1.8 Les points forts des applications des livraison

Pour assurer la réussite de notre projet, nous devons connaître les points forts des applications leader sur le marché. Pour cela, nous avons fait une petite recherche qui se résume dans le tableau suivant :

<i>L'application</i>	<i>Points forts</i>
Uber Eats	<ul style="list-style-type: none"> • Commander chez les restaurants à proximité. • Suivi des commandes en temps réel. • Toutes les principales cartes de crédit sont acceptées. • Les frais de livraison varient en fonction de votre emplacement.
Just Eat	<ul style="list-style-type: none"> • Prend en charge ApplePay, PayPal et GooglePay. • Avantage de la demande de groupe, rabais d'entreprise. • Les variétés de cuisine proposées sont diverses et multiples. • Le délai de livraison ne dépasse pas les trente minutes.

Page suivante

<i>L'application</i>	<i>Points forts</i>
Deliveroo	<ul style="list-style-type: none">• Les restaurants proposés sont affichés en fonction de la durée de livraison jusqu'à votre demeure.• Possibilité de choisir avec une grande flexibilité son délai de livraison.• Planifier une commande après certains temps.

TABLE 1.8 – Les points forts des applications de livraison

1.9 Conclusion

Dans ce chapitre nous avons fait une analyse de l'existant et présenté l'idée générale de notre projet. Dans ce qui suit nous allons entamer la première phase de la conception de notre projet « Analyse des besoins et spécifications » pour identifier les différentes fonctionnalités des trois applications.

Analyse et spécification des besoins

2.1 Introduction

Après avoir présenté le cadre général de notre projet, nous procéderons dans ce chapitre dans le but de déterminer les besoins fonctionnels et non fonctionnels auxquels nos applications doivent répondre. Ainsi nous allons définir le langage UML et ses outils : les diagrammes. Nous verrons comment ce langage peut contribuer à la phase d'analyse des besoins et du domaine d'un projet informatique.

2.2 Méthodologie de conception

2.2.1 UML

Nous parlons dans cette partie du langage de modélisation UML qui est aujourd'hui le langage standard utilisé en informatique pour la modélisation orientée objets.

Présentation d'UML

" UML (Unified Modeling Langage) n'est pas une méthode de développement mais un langage de modélisation qui définit des standards relatifs à la modélisation orientée objet. Ce langage permet de mettre en œuvre neuf diagrammes différents. Dans la mise en œuvre d'une méthode de développement, l'un ou l'autre des diagrammes est choisi en fonction des concepts que l'on veut représenter à une étape de développement donnée. Les concepts permettent de couvrir les étapes, depuis l'expression des besoins jusqu'au codage ". [8]

Les différents diagrammes d'UML

La nouvelle version du langage UML contient 13 diagrammes différents, y compris :

- Diagramme de classes.
- Diagramme d'objets.
- Diagramme de composants.
- Diagramme de déploiement.
- Diagramme de packages.
- Diagramme de structure composite.
- Diagramme de cas d'utilisation.
- Diagramme de vue d'ensemble des interactions.
- Diagramme d'état-transition.
- Diagramme d'activité.
- Diagramme de séquence.
- Diagramme de communication.
- Diagramme de temps.

2.3 Processus unifié (UP)

Comme nous l'avons expliqué dans l'introduction un modèle est souvent associé à une démarche de développement pour cela nous avons utilisés une méthode générique de développement de logiciel développée par les concepteurs d'UML. Il est défini comme suit : " Le processus unifié est un processus de développement logiciel, c'est-à-dire qu'il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel ". [9]

2.3.1 Cycles de vie d'un processus unifié

La gestion d'un tel processus est organisée suivant les quatre phases : création, élaboration, construction et transition.

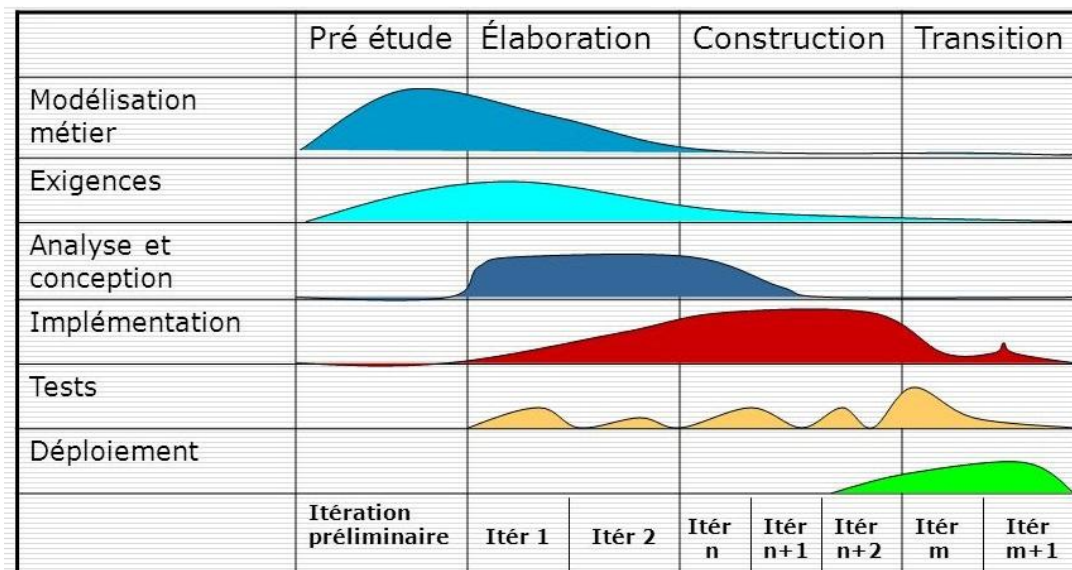


FIGURE 2.1 – Cycles de vie d'un processus unifié

2.3.2 Activités du processus

1. **Expression des besoins** : Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation
appréhender les besoins non fonctionnels (technique)
2. **Analyse** : L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences de notre projet. Le diagramme utilisé : Diagramme de cas d'utilisation.
3. **Conception** : La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et Framework et au système d'exploitation. Le diagramme utilisé :
 - Diagramme de classe pour la description de la vue statique.
 - Diagramme séquence pour la description de la vue dynamique.
4. **Implémentation** : Dans cette phase, nous devons déterminer les outils utilisés jusqu'au programmation. Le résultat de la conception pour implémenter le système sous formes de composants. Artefacts obtenus : code source, scripts, binaires.
5. **Test** : Les tests permettent de vérifier des résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque ité-

ration, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

2.4 Classification des diagrammes

Les diagrammes sont regroupés dans deux grands ensembles :

2.4.1 Les diagrammes structurels

Leur but est de représenter l'aspect statique d'un système, Ils sont sur la liste suivante :

- Diagramme de classes.
- Diagramme d'objets.
- Diagramme de composants.
- Diagramme de déploiement.
- Diagramme de packages.
- Diagramme de structure composite.

2.4.2 Les diagrammes de comportement

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Ils sont sur la liste suivante :

- Diagramme de cas d'utilisation.
- Diagramme de vue d'ensemble des interactions.
- Diagramme d'état-transition.
- Diagramme d'activité.
- Diagramme de séquence.
- Diagramme de communication.
- Diagramme de temps.

2.5 Les diagrammes utilisés

2.5.1 Le diagramme de cas d'utilisation

Il représente les fonctions du système vues sous l'angle de l'utilisateur. Les cas d'utilisation constituent un moyen de recueillir et de décrire les besoins des acteurs du système. Ils peuvent être aussi utilisés ensuite comme moyen d'organisation du développement du logiciel, notamment pour la structuration et le déroulement des tests du logiciel.

Tout système peut être décrit par un certain nombre de cas d'utilisation correspondant aux besoins exprimés par l'ensemble des utilisateurs. Pour formaliser un cas d'utilisateur on a besoin de :

Acteur : Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation.

Une interaction : permet de décrire les échanges entre un acteur et un cas d'utilisation.

Relations entre cas d'utilisation : trois relations peuvent être décrites entre cas d'utilisation :

- **une relation d'inclusion « include »** : Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

- **une relation d'extension « extend »** : Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.

- **une relation de généralisation** : Une relation de généralisation de cas d'utilisation peut être définie conformément au principe de la spécialisation-généralisation déjà présentée pour les classes.

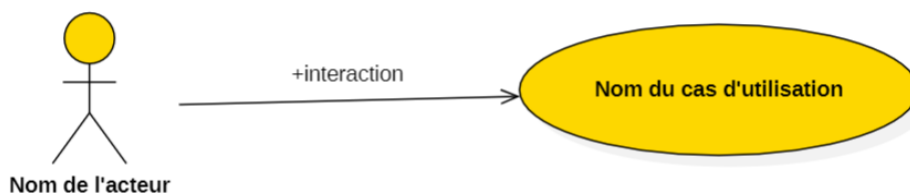


FIGURE 2.2 – Formalisme de base de représentation d'un cas d'utilisation

2.6 Spécification des besoins fonctionnels

2.6.1 Identification des acteurs

Pour répondre aux besoins des différents utilisateurs de l'application, il faut les classer par catégorie et essayer d'étudier chaque catégorie à part. Pour notre application, nous avons identifié les acteurs suivants :

- **Client** : toute personne qui utilise l'application "Garanda Food" pour commander des aliments depuis les magasins disponibles dans sa ville.
- **Vendeur** : tous les vendeurs qui utilisent l'application "GF Vendeurs" et qui possèdent un compte "Vendeur" pour gérer leurs magasins.
- **Livreur** : tous les livreurs qui utilisent l'application "GF Livreurs" et qui possèdent un compte "Livreur" pour livrer les commandes des clients depuis les magasins.
- **Administrateur** : c'est le superviseur qui possède toutes les permissions sur le système pour assurer le bon fonctionnement de système.

2.6.2 Identification des cas d'utilisation

- Coté Client

- Créer un compte : pour accéder à l'application le client doit obligatoirement créer un compte "Client" et remplir ses informations.
- Gérer son compte : le client peut à tout moment en accédant à son compte ajouté, modifier ses informations personnelles.
- Afficher les magasins par ville : le client peut consulter tous les magasins de sa ville.
- Accéder à un magasin : le client peut sélectionner un magasin pour voir ses informations et consulter son menu.
- Parcourir les catégories de cuisine : le client peut consulter la liste des catégories de cuisine disponibles
- Ajouter un article au panier : le client peut sélectionner un article dans un magasin spécifique et choisir les options souhaitées ainsi la quantité après il ajoute l'article au panier.
- Afficher les paniers : le client peut consulter tous les paniers qu'il a rempli par magasin et voir leurs détails.

- Passer une commande : après avoir rempli un panier le client peut passer une commande en sélectionnant l'adresse de livraison, et il peut aussi ajouter une remarque au vendeur.
- Afficher la liste des commandes : un client peut consulter les commandes effectuées et voir leurs statuts.

- Coté Vendeur

- Créer un compte vendeur : un vendeur doit d'abord créer une demande d'ouverture d'un magasin après un administrateur examine ses informations et décide s'il accepte la demande.
- Gérer son magasin : le vendeur peut à tout moment modifier les informations de son magasin
- Afficher le menu du magasin : le vendeur peut consulter la liste des articles de son magasin
- Ajouter ou modifier un article : le vendeur peut ajouter un nouvel article ou modifier un article existant
- Afficher les nouvelles commandes : un vendeur peut consulter la liste des nouvelles commandes classées par heure de création
- Afficher les commandes prêtes : un vendeur peut consulter la liste des commande prêtes et qui attendent un livreur pour les récupérer.
- Afficher l'historique des commandes : un vendeur peut consulter la liste de toutes les commandes terminées par le magasin que ce soit livrées ou annulées
- Voir les détails d'une commande : le vendeur peut consulter une commande qui contienne toutes les informations concernant le client, les articles commandés, les quantités et la remarque du client si elle existe
- Accepter une commande : après avoir consulter la commande et vérifier ses informations le vendeur peut accepter cette commande pour ensuite elle devienne 'En cours de préparation'
- Annuler une commande : si le vendeur constate un problème sur la commande comme des fausse informations il peut annuler cette commande
- Suspendre la réception des commandes : le vendeur peut mettre en pause la réception des nouvelles commandes

- Voir son code QR : un vendeur peut consulter le code QR du magasin pour être scanné par le livreur

- Côté Livreur

- Créer un compte livreur : un livreur doit d'abord créer une demande d'ouverture d'un compte livreur après un administrateur examine ses informations et décide s'il accepte la demande.
- Gérer son compte : le livreur peut à tout moment en accédant à son compte ajouté, modifier ses informations.
- Changer sa disponibilité : le livreur peut changer son statut à 'Hors service' pour mettre en pause la réception des nouvelles commandes, ou le contraire 'En service'.
- Afficher les commandes en attentes : un livreur peut consulter la liste des commande prêtes et qui attendent un livreur pour les récupérer.
- Accepter une commande : après avoir consulter une commande et selon l'éloignement du client et le cout de livraison, le livreur peut accepter la commande
- Annuler une commande : si une anomalie parviene lors de la livraison le livreur peut annuler une commande
- Localiser un client : le livreur peut localiser la position du client via GoogleMaps
- Appeler un client : le livreur peut communiquer avec un client par un appel téléphonique ou par SMS
- Scanner un code QR : lorsque le livreur arrive au client il doit scanner son code QR pour notifier le système que la commande a été livrée.
- Afficher l'historique des commandes : un livreur peut consulter la liste des commandes effectuées.

2.6.3 Diagramme de cas d'utilisation générale

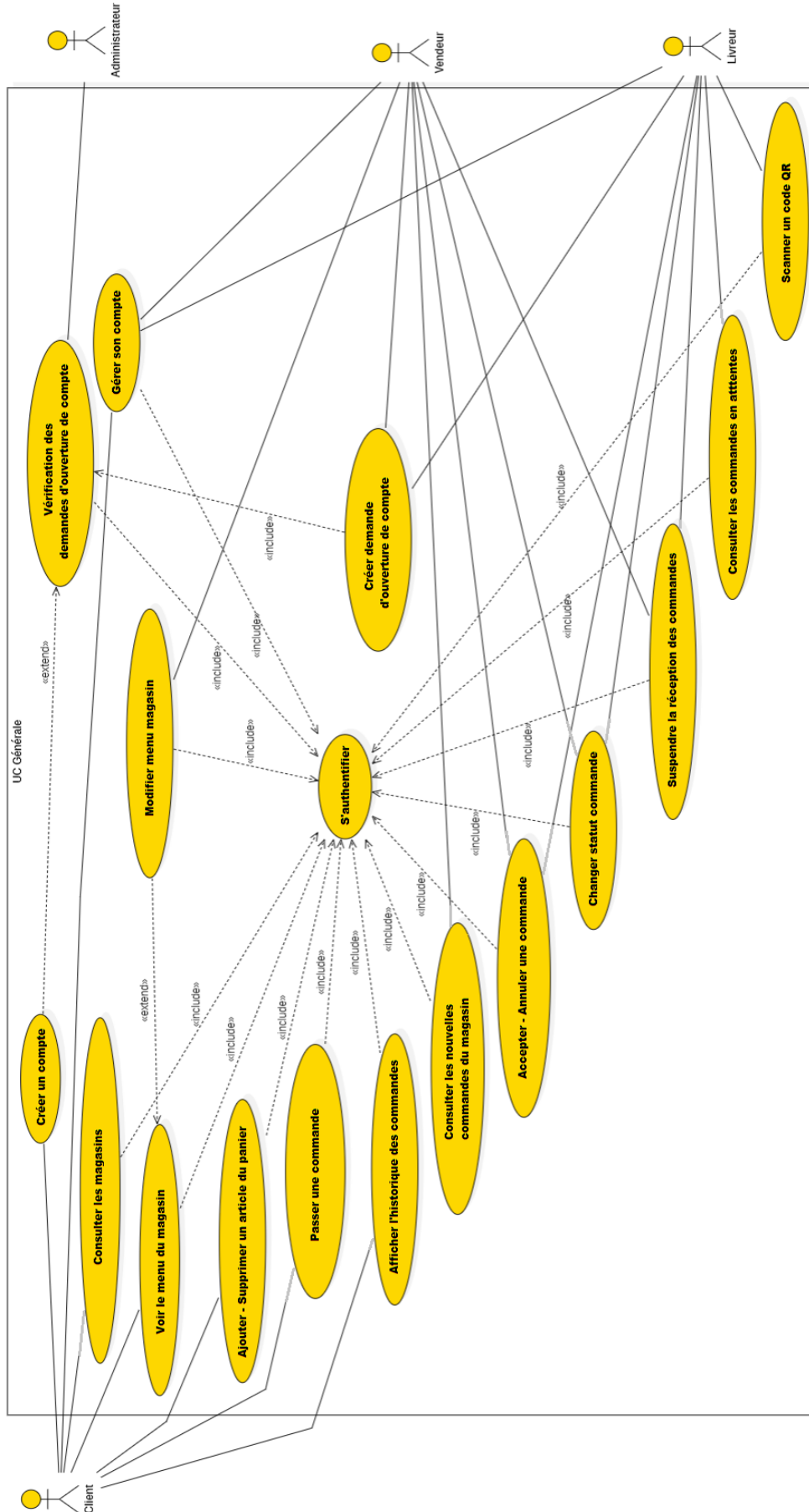


FIGURE 2.3 – Diagramme de cas d'utilisation générale

2.7 Spécification des besoins non fonctionnels

Les principaux besoins non fonctionnels de nos applications se résument dans les points suivants :

— **La sécurité :**

Les applications doivent respecter la confidentialité des données.

— **La disponibilité :**

Les applications doivent être disponibles pour être utilisées par n'importe quel utilisateur.

— **La fiabilité :**

Les données fournies par les applications doivent être fiables.

— **La convivialité de l'interface graphique :**

Les applications doivent fournir des interfaces conviviales et simples pour tout type d'utilisateur car elles présentent le premier contact de l'utilisateur avec l'une des applications et par le biais de celle-ci on découvrira leurs fonctionnalités.

— **Le code doit être clair :**

Pour permettre des futures évolutions ou améliorations.

2.8 Scénarios et cas d'utilisations

2.8.1 Scénario et cas d'utilisation raffiné relatif à S'authentifier pour client

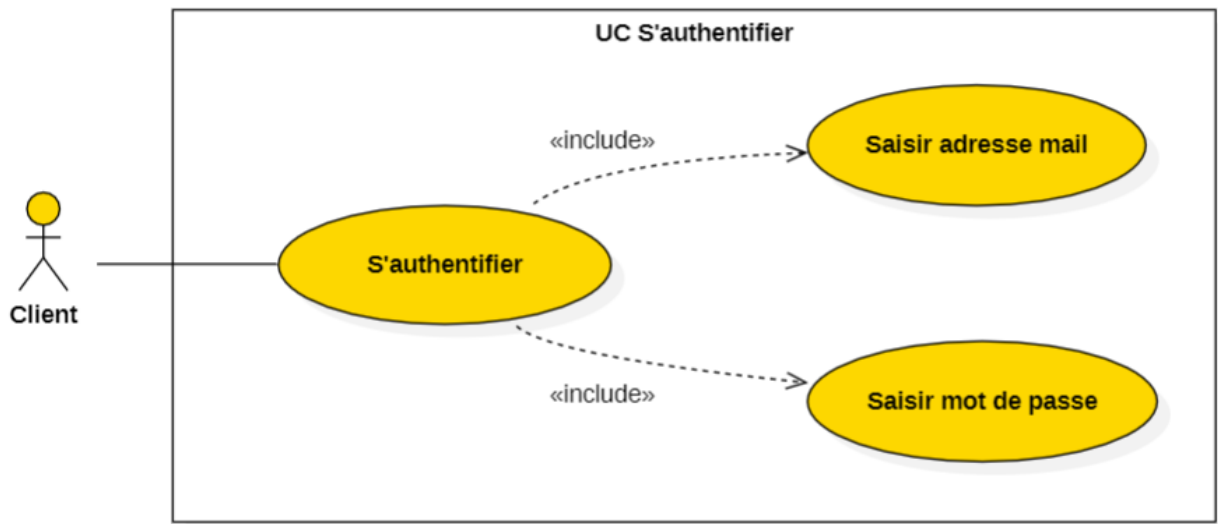


FIGURE 2.4 – Diagramme de cas d'utilisation «S'authentifier»

Le tableau 2.1 décrit textuellement le cas d'utilisation « S'authentifier » pour client

Titre	S'authentifier
But	Identification et autorisation d'accès.
Résumé	Le client introduit son adresse mail et le mot de passe pour accéder au système.
Acteurs	Client
Préconditions	
-L'application « Garanda Food » doit être installée sur le smartphone du Client. -Disponibilité d'une connexion Internet.	
Post-conditions	
-Accès aux diverses fonctionnalités de l'application.	

Page suivante

Scénario Nominal
<ol style="list-style-type: none"> 1. Le Client lance l'application. 2. L'application affiche le formulaire d'authentification (adresse mail et mot de passe). 3. Le Client saisit son adresse mail et son mot de passe. 4. Le système vérifie les champs (champs obligatoires,...). 5. Si le Client est identifié, le système affiche l'interface d'accueil.
Enchaînement alternative
<p>E1 : le Client introduit des paramètres d'authentications Incorrectes.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur et demande au Client de vérifier l'ensemble des champs.

TABLE 2.1 – Raffinement de cas d'utilisation « S'authentifier »

2.8.2 Scénario et cas d'utilisation raffiné relatif à Consulter les magasins

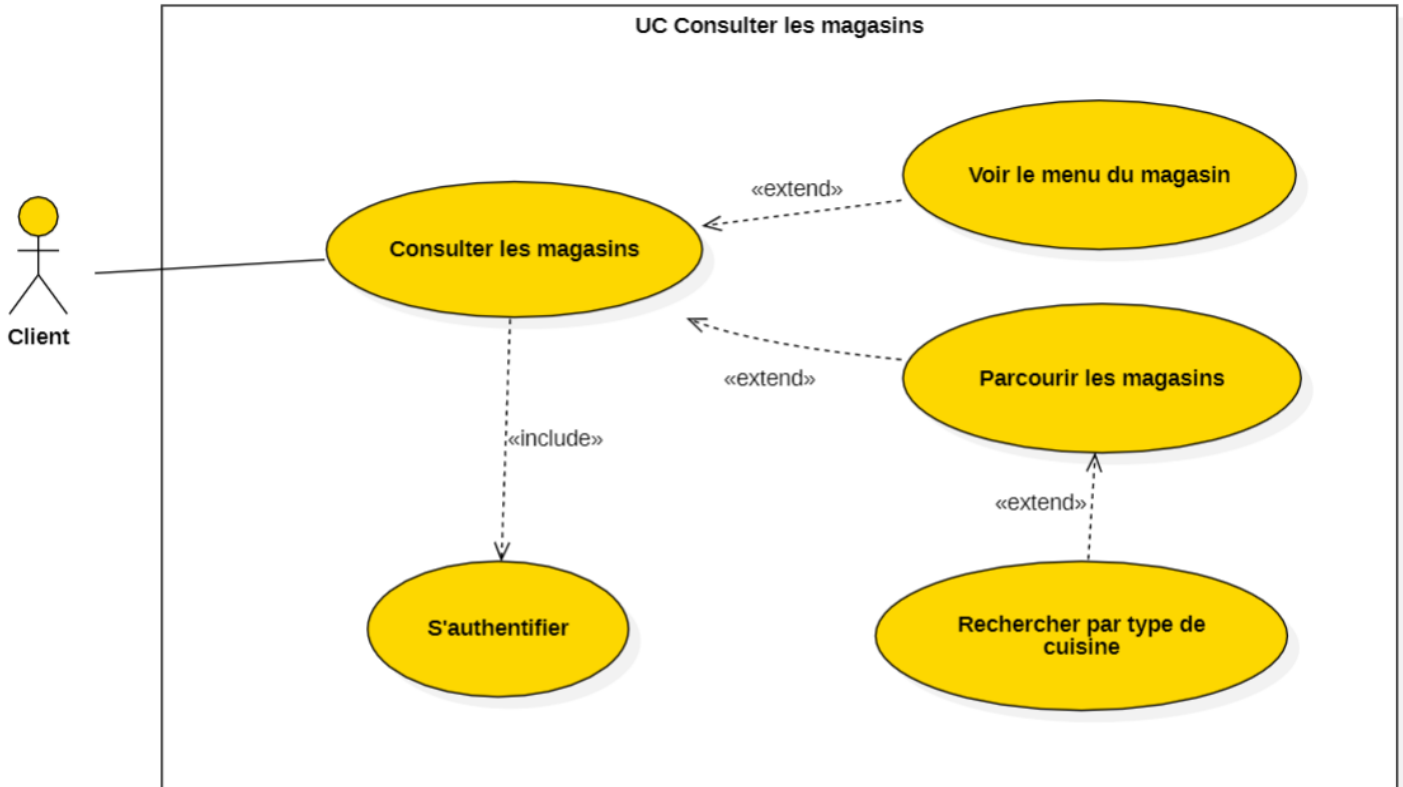


FIGURE 2.5 – Diagramme de cas d'utilisation « Consulter les magasins »

Le tableau 2.2 décrit textuellement le cas d'utilisation « Consulter les magasins ».

<i>Titre</i>	Consulter les magasins
<i>But</i>	Voir le menu d'un magasin et choisir les articles à commander.
<i>Résumé</i>	Le client peut consulter la liste des magasins et voir leurs menus.
<i>Acteurs</i>	Client
<i>Préconditions</i>	
Le Client doit être authentifié.	
<i>Post-conditions</i>	
Le Client consulte le menu d'un magasin choisi.	
<i>Scénario Nominal</i>	
<ol style="list-style-type: none"> 1. Le Client clique sur accueil. 2. Le système affiche les magasins disponibles dans la ville du Client. 3. Le Client choisit un magasin. 4. Le système affiche le menu du magasin ainsi ses informations. 	

TABLE 2.2 – Raffinement de cas d'utilisation « Consulter les magasins »

Le tableau 2.3 décrit textuellement le cas d'utilisation « Rechercher un magasin ».

<i>Titre</i>	Rechercher un magasin.
<i>But</i>	Trouver le meilleur magasin qui offre un type de cuisine spécifique.
<i>Résumé</i>	Le Client peut rechercher les magasins selon leurs types de cuisine.
<i>Acteurs</i>	Client
<i>Préconditions</i>	
-Le Client doit être authentifié.	
<i>Post-conditions</i>	
-Le Client trouve les magasins qui offrent le type de cuisine recherché.	

Page suivante

Scénario Nominal
<ol style="list-style-type: none"> 1. Le Client clique sur parcourir. 2. Le système affiche les types de cuisine disponibles. 3. Le Client choisit un type de cuisine. 4. Le système affiche tous les magasins qui correspondent à ce type.
Enchaînement alternative
<p>E1 : Aucun magasin pour ce type de cuisine dans la ville du Client.</p> <ol style="list-style-type: none"> 1. Le système affiche une interface vide avec un message d’erreur « Aucun magasin disponible pour ce type de cuisine ».

TABLE 2.3 – Raffinement de cas d’utilisation «Rechercher un magasin»

2.8.3 Scénario et cas d’utilisation raffiné relatif à Passer une commande

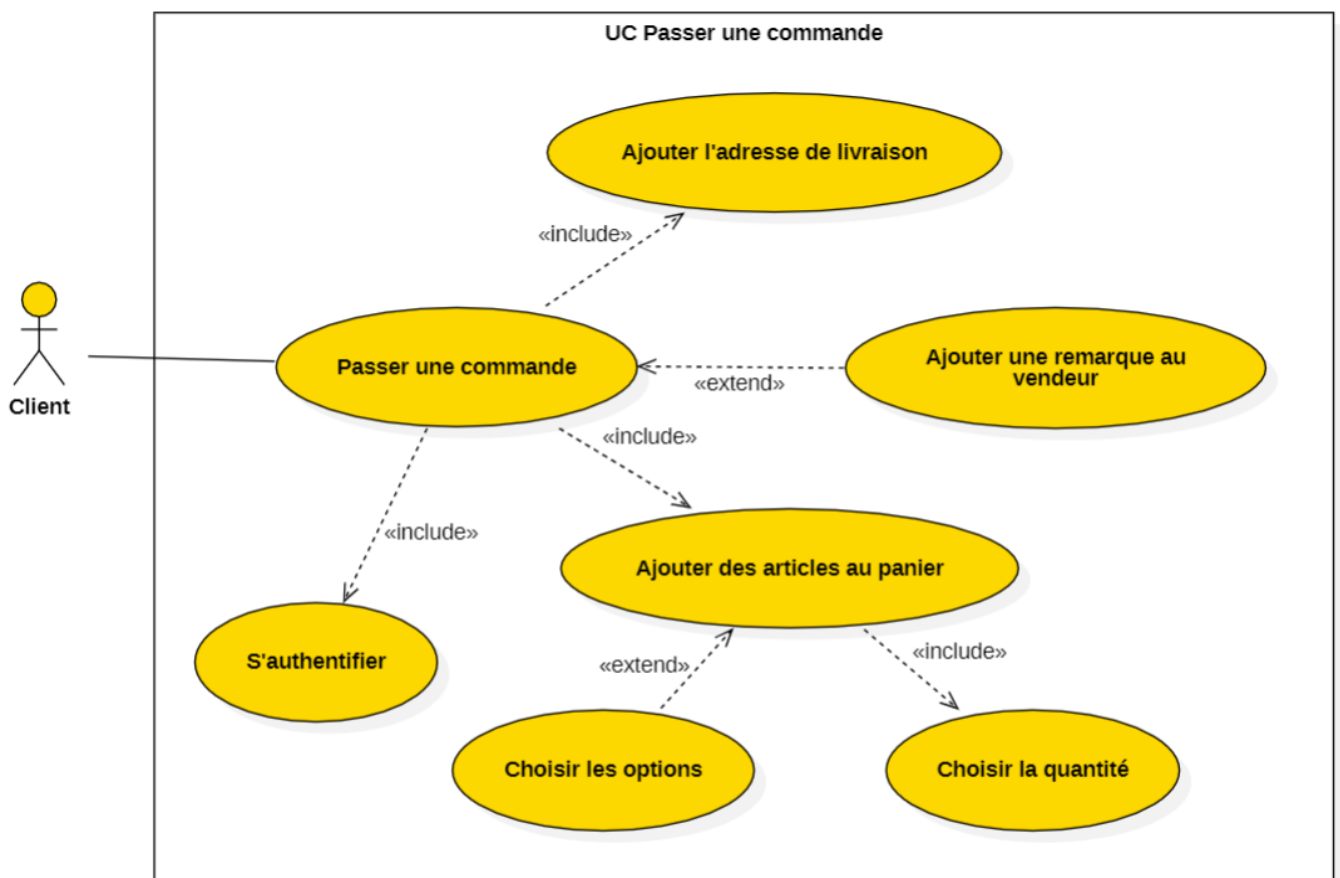


FIGURE 2.6 – Diagramme de cas d’utilisation «Passer une commande»

Le tableau 2.4 décrit textuellement le cas d'utilisation « Ajouter articles au panier ».

<i>Titre</i>	Ajouter des articles au panier.
<i>But</i>	Remplir un panier.
<i>Résumé</i>	Le Client ajoute les articles souhaités d'un magasin au panier en choisissant les options et la quantité.
<i>Acteurs</i>	Client
<i>Préconditions</i>	
-Le Client doit être authentifié.	
<i>Post-conditions</i>	
-Un nouveau panier créé avec les articles souhaités.	
<i>Scénario Nominal</i>	
<ol style="list-style-type: none"> 1. Le Client choisit un article dans le menu du magasin. 2. Le système affiche l'image, le nom et la description de l'article, ainsi les options si disponibles. 3. Le Client choisit les options souhaitées si disponibles. 4. Le Client choisit ensuite la quantité et clique sur « Ajouter au panier ». 5. Un article ajouté au panier. 	

TABLE 2.4 – Raffinement de cas d'utilisation «Ajouter des articles au panier.»

Le tableau 2.5 décrit textuellement le cas d'utilisation « Passer une commande ».

<i>Titre</i>	Passer une commande.
<i>But</i>	Commander des articles.
<i>Résumé</i>	Le Client sélectionne l'adresse de livraison et passe la commande.
<i>Acteurs</i>	Client
<i>Préconditions</i>	
-Le Client doit être authentifié.	
<i>Post-conditions</i>	
-Une nouvelle commande effectuée.	

Page suivante

<i>Scénario Nominal</i>
<ol style="list-style-type: none">1. Le Client clique sur commander dans le panier choisi.2. Le système affiche les détails de la commande : les articles ajoutés, le montant total des articles, un bouton pour sélectionner l'adresse de livraison et un autre pour ajouter une remarque au Vendeur.3. Le Client sélectionne la position de l'adresse de livraison dans le Maps, ensuite les frais de livraison seront calculés automatiquement.4. Le Client peut ajouter une remarque au Vendeur s'il veut.5. Le Client clique sur commander.6. Une nouvelle commande effectuée.
<i>Enchaînement alternative</i>
<p>E1 : La ville sélectionnée par le Client n'est disponible dans le système.</p> <ol style="list-style-type: none">1. Le système affiche un message d'erreur « Notre service n'est disponible dans la ville sélectionnée ».2. Le système reprend de l'étape 2 du scénario nominal.

TABLE 2.5 – Raffinement de cas d'utilisation «Passer une commande.»

2.8.4 Scénario et cas d'utilisation raffiné relatif à Créer un compte pour vendeur

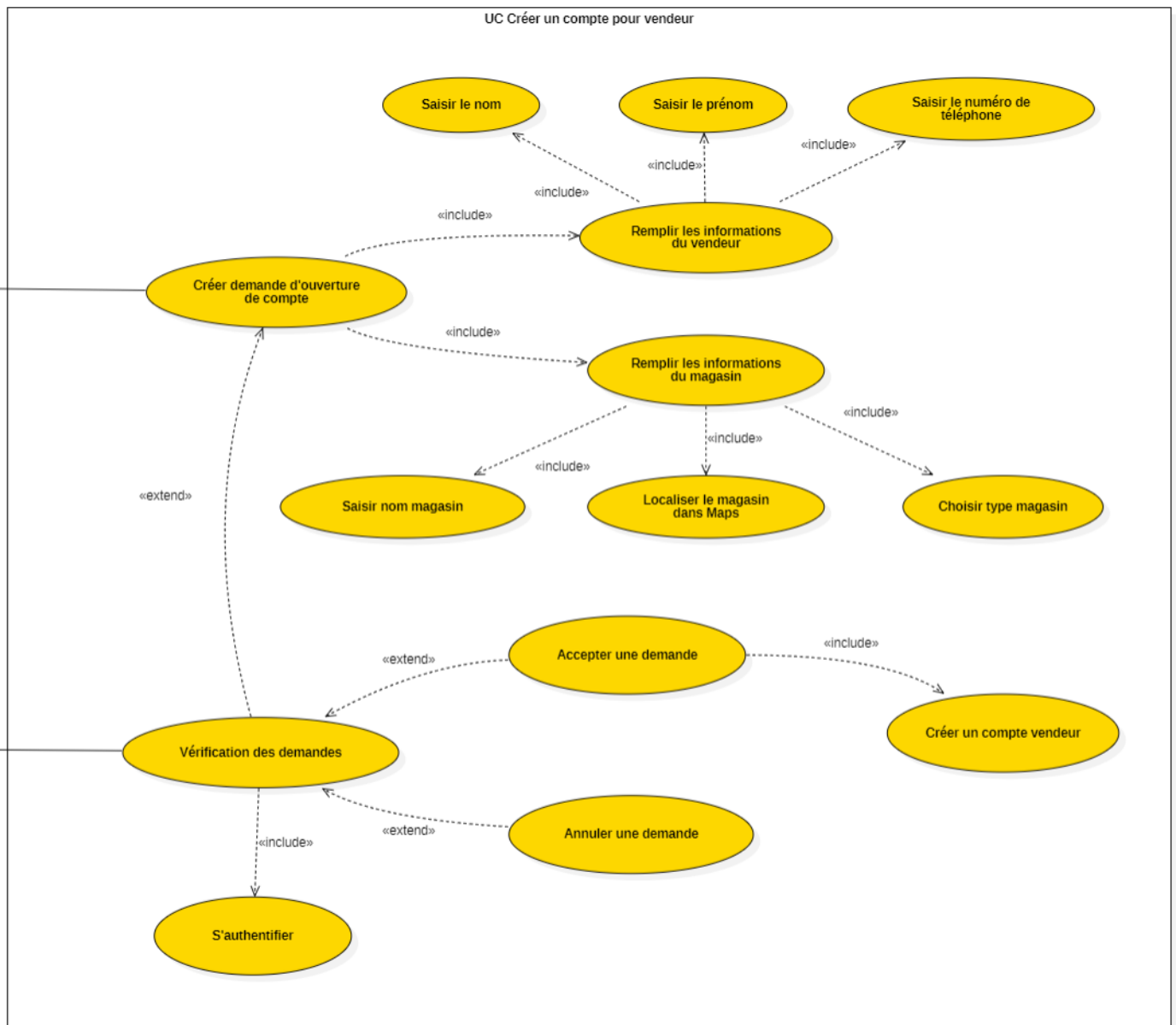


FIGURE 2.7 – Diagramme de cas d'utilisation «Créer un compte vendeur»

Le tableau 2.6 décrit textuellement le cas d'utilisation « Créer un compte » pour vendeur.

<i>Titre</i>	Créer un compte Vendeur
<i>But</i>	Créer un compte pour un vendeur.
<i>Résumé</i>	Un vendeur souhaite ouvrir un nouveau magasin dans le système.
<i>Acteurs</i>	Vendeur/ Administrateur.
<i>Préconditions</i>	
<ul style="list-style-type: none"> -L'application « Garanda Food Vendeurs » doit être installée sur le smartphone du Vendeur. -Disponibilité d'une connexion Internet. 	
<i>Post-conditions</i>	
-Un nouveau compte Vendeur et magasin ajoutés.	
<i>Scénario Nominal</i>	
<ol style="list-style-type: none"> 1. Le Vendeur lance l'application. 2. Le Vendeur choisit créer un nouveau compte. 3. Le système affiche un formulaire d'inscription. 4. Le Vendeur saisit ses informations ainsi que les informations du magasin. 5. Le système vérifie les champs. 6. Une demande d'ouverture de compte Vendeur est créée. 7. Un Administrateur vérifie la demande du Vendeur. 8. Si la demande est acceptée, un nouveau compte Vendeur est créé. 	
<i>Enchaînement alternative</i>	
<p>E1 : Champs obligatoires non valides et/ou vides</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur et demande au Vendeur de vérifier l'ensemble des champs. 2. Le système reprend de l'étape 3 du scénario nominal. <p>E2 : L'adresse mail existe dans la base de données.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur « Adresse mail existe déjà ». 2. Le système reprend de l'étape 3 du scénario nominal. <p>E3 : La position du magasin est en dehors de la zone qui couvre le système.</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur « Notre service n'est pas disponible dans la ville sélectionnée ». 2. Le système reprend de l'étape 3 du scénario nominal. 	

TABLE 2.6 – Raffinement de cas d'utilisation « Créer un compte » pour vendeur

2.8.5 Scénario et cas d'utilisation raffiné relatif à Consulter les nouvelles commandes

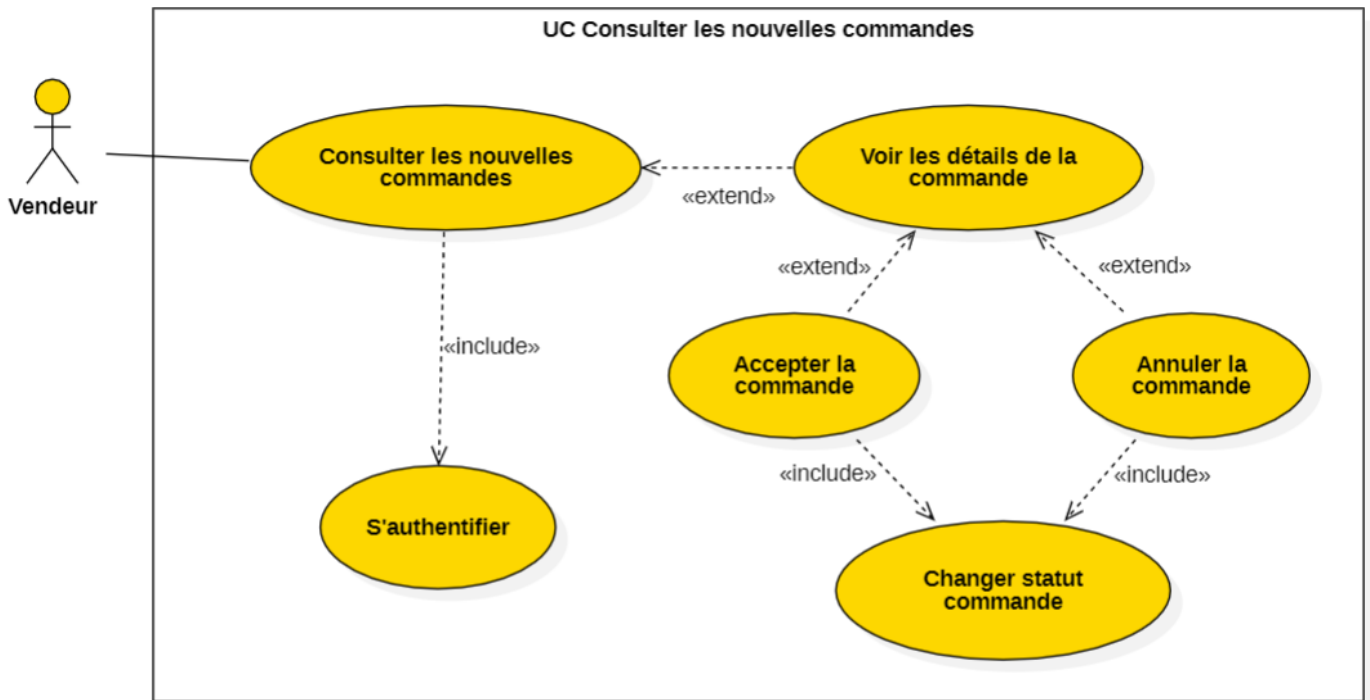


FIGURE 2.8 – Diagramme de cas d'utilisation « Consulter les nouvelles commandes »

Le tableau 2.7 décrit textuellement le cas d'utilisation « Consulter les nouvelles commandes ».

Titre	Consulter les nouvelles commandes.
But	Préparer les commandes pour être livrées par le Livreur.
Résumé	Un Client passe une commande, ensuite le Vendeur sera notifié et peut voir les détails de la commande, après le Vendeur décide s'il accepte la commande ou non.
Acteurs	Vendeur.
Préconditions	
-Le Vendeur doit être authentifié.	
Post-conditions	
-Le statut de la commande est changé « Accepté » ou « Annulé ».	

Scénario Nominal
<ol style="list-style-type: none"> 1. Le système affiche les nouvelles commandes dans d'accueil. 2. Le Vendeur choisit une commande pour voir ses détails. 3. Le système affiche les informations de la commande : nom Client, numéro de téléphone du Client, les articles commandés, une remarque si elle existe. 4. Si le Vendeur accepte la commande, il clique sur « Accepter » et son statut sera changé à « Confirmée et en cours de préparation ». 5. Si le Vendeur refuse la commande, il clique sur « Annuler » et son statut sera changé à « Annulée ».

TABLE 2.7 – Raffinement de cas d'utilisation « Consulter les nouvelles commandes »

2.8.6 Diagramme de cas d'utilisation « Suspendre la réception des commandes »

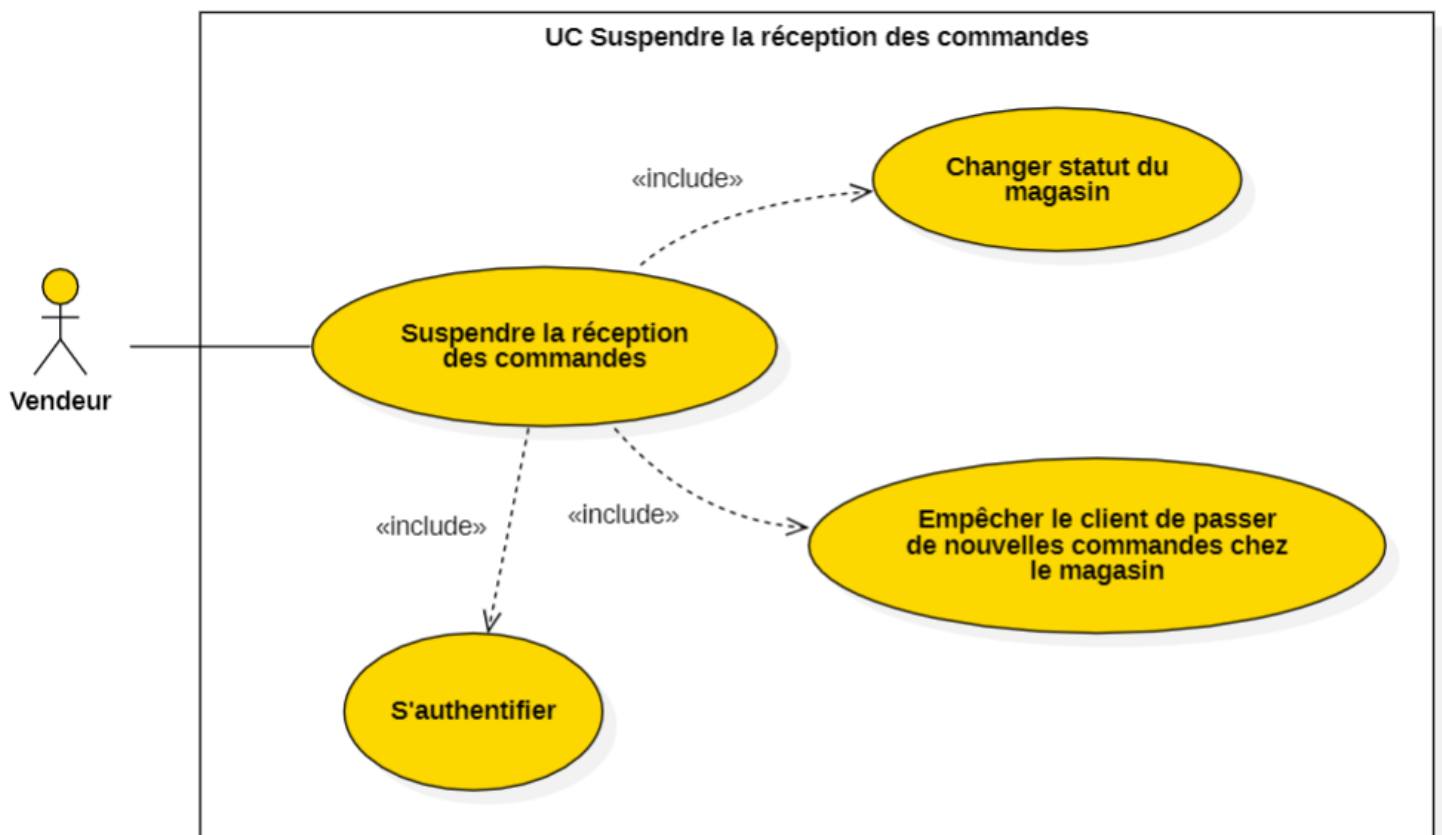


FIGURE 2.9 – Diagramme de cas d'utilisation « Suspendre la réception des commandes »

Le tableau 2.8 décrit textuellement le cas d'utilisation « Suspendre la réception des commandes »

<i>Titre</i>	Suspendre la réception des commandes.
<i>But</i>	Le Vendeur peut suspendre la réception des nouvelles commandes pour certains raisons.
<i>Résumé</i>	Si le Vendeur ferme le magasin dans les heures de travaille ou rencontre un problème, il peut suspendre la réception des nouvelles commandes.
<i>Acteurs</i>	Vendeur.
<i>Préconditions</i>	
-Le Vendeur doit être authentifié.	
<i>Post-conditions</i>	
-La réception des nouvelles commandes est suspendu.	
<i>Scénario Nominal</i>	
<ol style="list-style-type: none"> 1. Dans l'interface d'accueil le Vendeur clique sur l'icône « Pause ». 2. Le système affiche un message de confirmation. 3. Le Vendeur clique sur confirmer. 4. Le système change le statut du magasin à « non disponible » et affiche un message permanant que la réception des commandes est suspendue. 5. Le système empêche le Client de cliquer sur « Commander » pour ce magasin. 	

TABLE 2.8 – Raffinement de cas d'utilisation « Suspendre la réception des commandes ».

2.8.7 Scénario et cas d'utilisation raffiné relatif à Consulter les commandes en attente

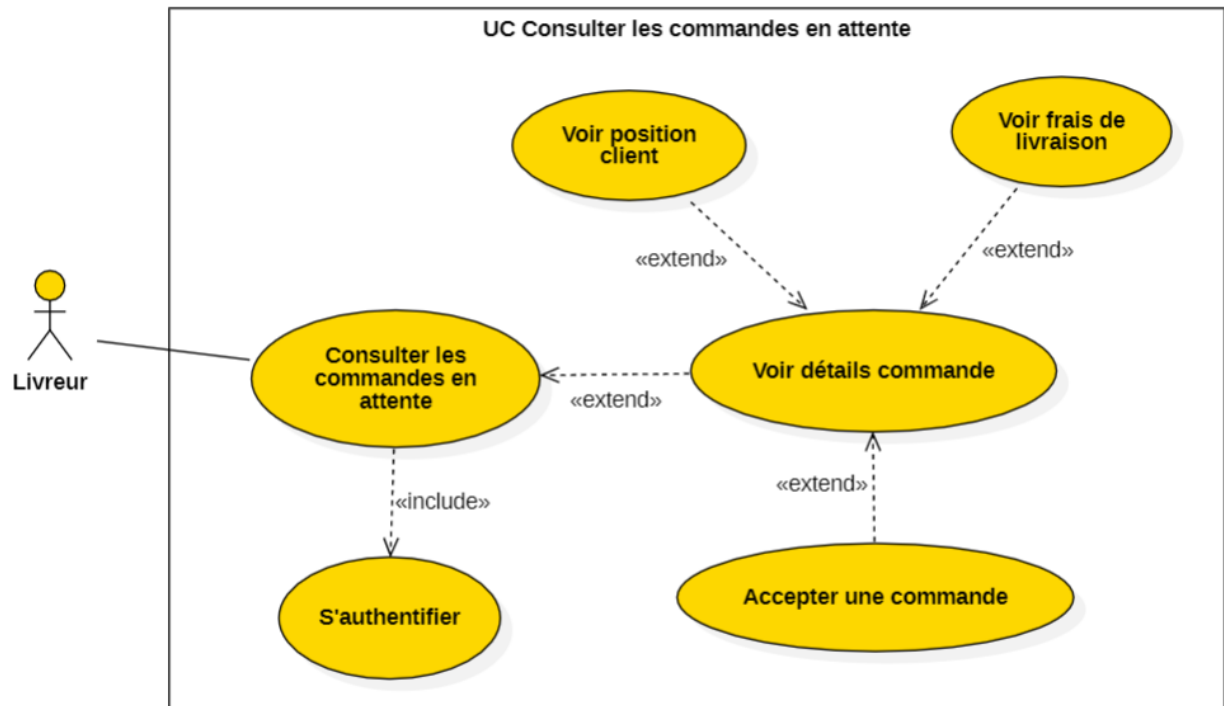


FIGURE 2.10 – Diagramme de cas d'utilisation « Consulter les commandes en attente »

Le tableau 2.9 décrit textuellement le cas d'utilisation « Consulter les commandes en attente ».

Titre	Consulter les commandes en attente.
But	Choisir une commande pour commencer la livraison.
Résumé	Le Livreur peut voir les commandes prêtes et en attente de livraison, il choisit une commande pour ensuite commencer la livraison.
Acteurs	Livreur.
Préconditions	
	-Le Livreur doit être authentifié.
Post-conditions	
	-La livraison est commencé.

Page suivante

Scénario Nominal
<ol style="list-style-type: none"> 1. Le système affiche les commandes en attente dans l'accueil. 2. Le Livreur choisit une commande pour consulter ses informations. 3. Le système affiche les détails de la commande : numéro de la commande, nom du magasin, nom du Client ainsi son numéro de téléphone, les articles commandés et les frais de livraison. 4. Si le Livreur décide de livrer la commande, il clique sur « Accepter ». 5. Le système affiche une interface qui affiche aussi les détails de la commande et montre le statut de la livraison, le Vendeur peut localiser la position du Magasin ou du Client.

TABLE 2.9 – Raffinement de cas d'utilisation « Consulter les commandes en attente »

2.8.8 Scénario et cas d'utilisation raffiné relatif à Scanner codeQR

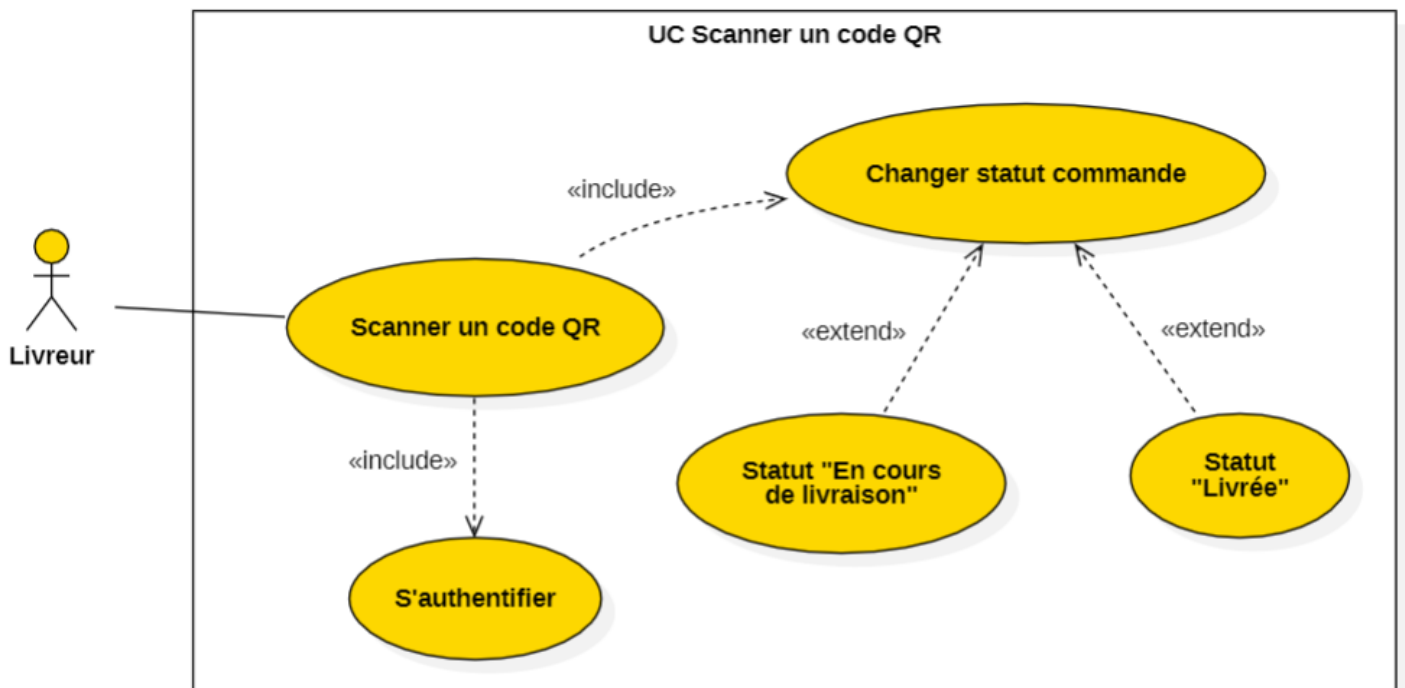


FIGURE 2.11 – Diagramme de cas d'utilisation « Scanner un code QR »

Le tableau 2.10 décrit textuellement le cas d'utilisation « Scanner un code QR ».

<i>Titre</i>	Scanner un code QR.
<i>But</i>	Changer le statut de la commande.
<i>Résumé</i>	Lorsque le Livreur arrive au magasin pour récupérer une commande, il scanne le code QR du magasin ensuite le statut de la commande devient « En cours de livraison ». Et lorsqu'il arrive au Client et lui DONNE la commande, il scanne aussi le code QR du Client. Et enfin le statut de la commande devient « Livrée ».
<i>Acteurs</i>	Livreur.
<i>Préconditions</i>	
-Le Livreur doit accepter une commande.	
<i>Post-conditions</i>	
-Le statut de la commande est changé.	
<i>Scénario Nominal</i>	
<ol style="list-style-type: none"> 1. Le Livreur arrive au magasin. 2. Le Livreur clique sur l'icône « scanner ». 3. Le système lance la caméra et commence à détecter les codes QR. 4. Le système scanne le code QR du magasin et change le statut de la commande à « En cours de livraison » 5. Et lorsque le Livreur arrive au Client. 6. Le Livreur clique encore sur l'icône « scanner ». 7. Le système scanne le code QR du Client et change le statut de la commande à « Livrée ». 	
<i>Enchaînement alternative</i>	
E1 : le code QR n'est pas identique au ID du magasin ou du Client.	
<ol style="list-style-type: none"> 1. Le système affiche un message d'erreur « Code QR non valide ». 	

TABLE 2.10 – Raffinement de cas d'utilisation «Scanner un code QR»

2.9 Conclusion

Ce deuxième chapitre a été dédié à l'analyse et la spécification des besoins. Nous avons parlé du langage de modélisation UML illustré par les diagrammes de cas d'utilisation. Dans ce qui suit nous allons entamer la phase de conception.

Conception

3.1 Introduction

La conception de système d'information est la première étape dans le cycle de vie d'une application et pour la réalisation de cette étape nous avons commencé par la conception technique en décrivant les vues statiques et dynamique du système en utilisant les diagrammes UML appropriés.

3.2 Les diagrammes utilisés [10]

3.2.1 Diagramme de séquence

C'est une représentation temporelle des objets et leurs interactions. L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Deux types de messages peuvent être distingués, dans un diagramme de séquence :

Message synchrone : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions.

Message asynchrone : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. Un diagramme de séquence se représente globalement dans un grand rectangle avec indication du nom du diagramme en haut à gauche. Une ligne de vie : représente l'ensemble des opérations exécutées par un objet.

Contrainte temporelle : Des contraintes de chronologie entre les messages peuvent être spécifiées. De plus lorsque l'émission d'un message requiert une certaine durée, il se re-

présente sous la forme d'un trait oblique.

Les opérateurs : Treize opérateurs ont été définis dans UML : alt, opt, loop, par, strict/weak, break, ignore/ consider, critical, negative, assertion et ref.

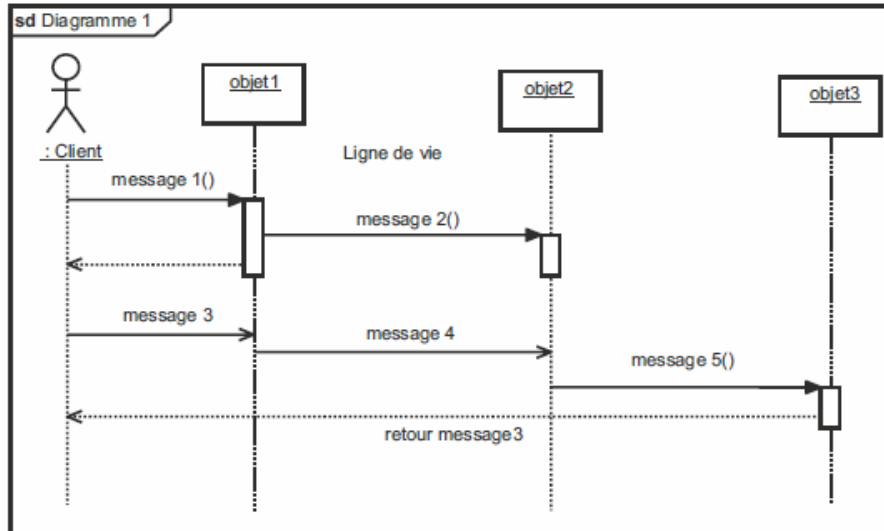


FIGURE 3.1 – Exemple de diagramme de séquence

3.2.2 Diagramme de classe

C'est un diagramme qui permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe, seul l'algorithme général et le pseudo-code correspondant peuvent être associés à la modélisation.

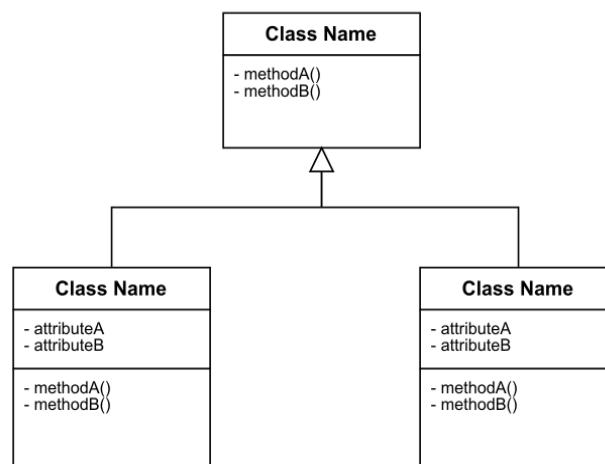


FIGURE 3.2 – Exemple de diagramme de classe

3.3 Description de la vue statique

3.3.1 Diagramme des classes

Le diagramme de classe retenu à la fin de la conception (Figure 3.3) est le suivant :



FIGURE 3.3 – Diagramme des classes

3.3.2 Modélisation de la base de données « Firestore »

En suivant le modèle de données NoSQL de Cloud Firestore, sur la figure 4.2 ci-dessous on a organisé nos classes principales dans des collection. Chaque collection contient des documents (instance de classe) qui contient des champs mappés à des valeurs (attributs). Un document peut contenir des sous-collections donc on a créé des collections imbriquées pour bien structuré notre base de données. Pour les classes secondaires qui ne sont pas cités dans la figure 4.2 on les a utilisées comme types d'attributs pour les documents.

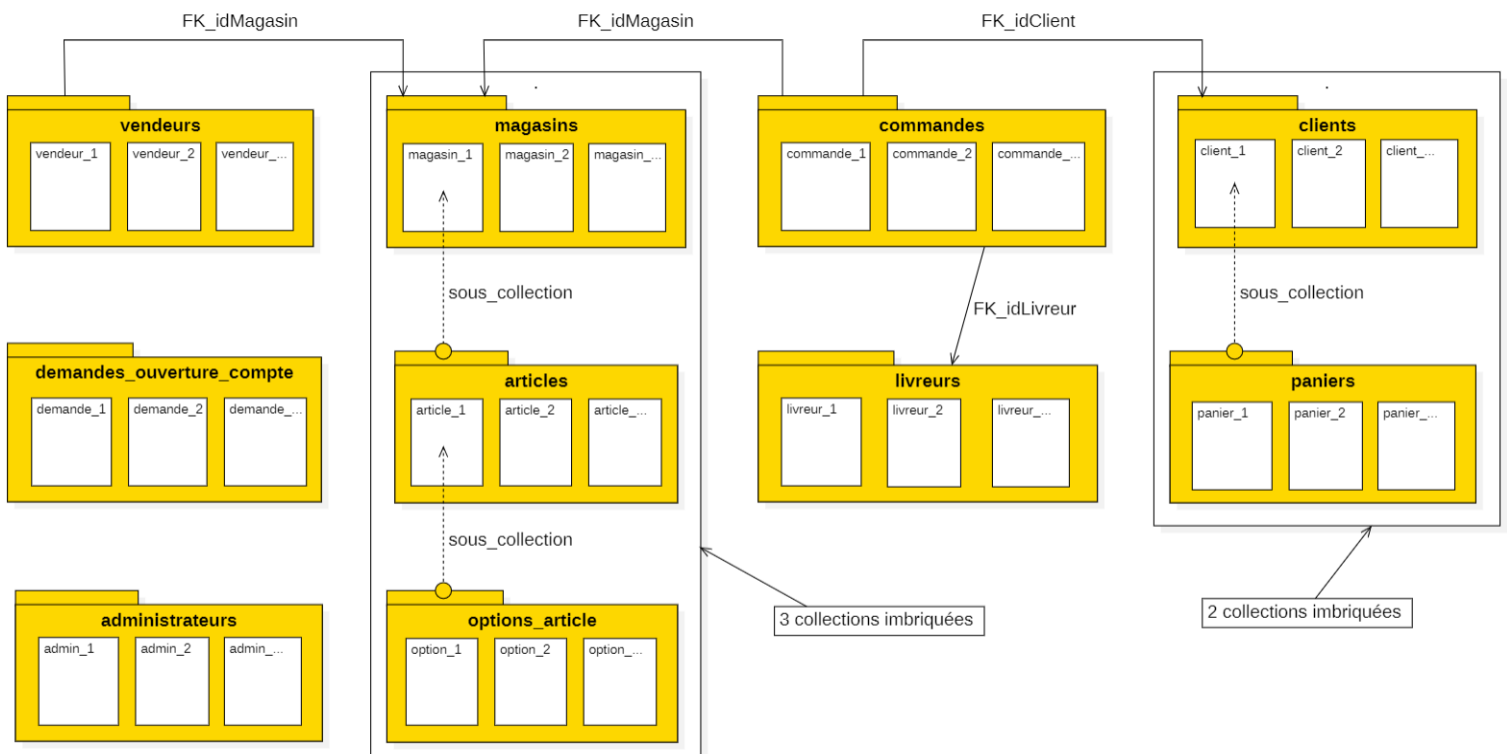


FIGURE 3.4 – Répartition des collections sur Firestore

Les Collections de la base de données

Les Collections de la base de données En s'appuyant sur la modélisation de la base de données et le diagramme de classes nous présentons les tableaux des collections ci-dessous, sachant que les types de données pris en charge par Cloud Firestore sont : string, number (contient int, float, double ...), timestamp (date), boolean, array (pour tous les types de listes) et null. Les objets sont décomposés en des types primitifs.

Le tableau 3.1 contient une description des différents attributs de la collection «Client»

Attribut	Description
<i>idClient</i>	C'est un attribut de type « string », il représente l'identifiant du client (clé primaire).
<i>nomComplet</i>	C'est un attribut de type « string », il représente le nom et le prénom du client.
<i>email</i>	C'est un attribut de type « string », il représente l'adresse mail du client.
<i>numeroTelephone</i>	C'est un attribut de type « string », il représente le numéro de téléphone du client.
<i>listeAdresses</i>	C'est un attribut de type « array », il représente la liste des adresses du client.

TABLE 3.1 – Description relative aux attributs de la collection « Client »

Le tableau 3.2 contient une description des différents attributs de la collection « Magasin »

Attribut	Description
<i>idMagasin</i>	C'est un attribut de type « string », il représente l'identifiant du magasin (clé primaire).
<i>nomMagasin</i>	C'est un attribut de type « string », il représente le nom du magasin.
<i>urlImageMagasin</i>	C'est un attribut de type « string », il représente l'URL de l'image du magasin.
<i>adresseMagasin</i>	C'est un attribut de type « Adresse », il représente l'adresse du magasin.

<i>typeMagasin</i>	C'est un attribut de type « string », il représente le type du magasin.
<i>listeCategoriesCuisine</i>	C'est un attribut de type « array », il représente la liste des catégories de cuisine qui contient le magasin.
<i>listeCategoriesMenu</i>	C'est un attribut de type « array », il représente la liste des catégories dans le menu du magasin.
<i>estSuspendu</i>	C'est un attribut de type « boolean », il représente l'état du magasin s'il reçoit des nouvelles commandes ou non.
<i>statutMagasin</i>	C'est un attribut de type « number », il représente le statut du magasin « Activé » ou « Désactivé ».

TABLE 3.2 – Description relative aux attributs de la collection « Magasin »

Le tableau 3.3 contient une description des différents attributs de la collection «Article»

Attribut	Description
<i>idArticle</i>	C'est un attribut de type « string », il représente l'identifiant de l'article (clé primaire).
<i>nomArticle</i>	C'est un attribut de type « string », il représente le nom de l'article.
<i>description</i>	C'est un attribut de type « string », il représente la description de l'article.
<i>urlImageArticle</i>	C'est un attribut de type « string », il représente l'URL de l'image de l'article.
<i>categorieMenu</i>	C'est un attribut de type « string », il représente la catégorie de l'article dans le menu du magasin.
<i>typeCuisine</i>	C'est un attribut de type « string », il représente le type de cuisine de l'article.
<i>prix</i>	C'est un attribut de type « number », il représente le prix de l'article en dinars.
<i>optionsSontDisponibles</i>	C'est un attribut de type « boolean », il représente les options de l'article s'il existe ou non.
<i>estDisponible</i>	C'est un attribut de type « boolean », il représente la disponibilité de l'article dans le magasin.

TABLE 3.3 – Description relative aux attributs de la collection «Article»

Le tableau 3.4 contient une description des différents attributs de la collection « OptionsArticle »

Attribut	Description
<i>idOption</i>	C'est un attribut de type « string », il représente l'identifiant de l'option d'article (clé primaire).
<i>nomOption</i>	C'est un attribut de type « string », il représente le nom de l'option d'article.
<i>typeOption</i>	C'est un attribut de type « number », il représente le type de l'option d'article « obligatoire » ou « optionnel ».
<i>nbrChoixMaximum</i>	C'est un attribut de type « number », il représente le nombre maximal de choix qu'on peut choisir dans cette option.
<i>listeChoix</i>	C'est un attribut de type « array », il représente la liste des choix pour cette option

TABLE 3.4 – Description relative aux attributs de la collection «OptionsArticle»

Le tableau 3.5 contient une description des différents attributs du type « ChoixOptionArticle »

Attribut	Description
<i>idChoix</i>	C'est un attribut de type « string », il représente l'identifiant du choix (clé primaire).
<i>choix</i>	C'est un attribut de type « string », il représente le nom du choix.
<i>prixChoix</i>	C'est un attribut de type « number », il représente le prix du choix en dinars.

TABLE 3.5 – Description relative aux attributs du type « ChoixOptionArticle »

Le tableau 3.6 contient une description des différents attributs de la collection « Panier »

Attribut	Description
<i>idPanier</i>	C'est un attribut de type « string », il représente l'identifiant du panier (clé primaire).
<i>idMagasin</i>	C'est un attribut de type « string », il représente l'identifiant du magasin (clé étrangère de la collection Magasin).
<i>nbrArticles</i>	C'est un attribut de type « number », il représente le nombre des articles que le panier contient.
<i>totalPanier</i>	C'est un attribut de type « number », il représente le montant total du panier en dinars.

TABLE 3.6 – Description relative aux attributs de la collection « Panier »

Le tableau 3.7 contient une description des différents attributs de la collection « Commande »

Attribut	Description
<i>idCommande</i>	C'est un attribut de type « string », il représente l'identifiant de la commande (clé primaire).
<i>idClient</i>	C'est un attribut de type « string », il représente l'identifiant du client (clé étrangère de la collection Client).
<i>idMagasin</i>	C'est un attribut de type « string », il représente l'identifiant du magasin (clé étrangère de la collection Magasin).
<i>idLivreur</i>	C'est un attribut de type « string », il représente l'identifiant du livreur (clé étrangère de la collection Livreur).

Page suivante

<i>numeroCommande</i>	C'est un attribut de type « string », il représente le numéro unique de la commande.
<i>adresseLivraison</i>	C'est un attribut de type « Adresse », il représente l'adresse de livraison pour cette commande.
<i>dateCreation</i>	C'est un attribut de type « timestamp », il représente la date de création de la commande.
<i>statut</i>	C'est un attribut de type « number », il représente le statut de la commande.
<i>listeDesArticles</i>	C'est un attribut de type « array », il représente la liste des articles de la commande.
<i>totalArticles</i>	C'est un attribut de type « number », il représente le total des articles de la commande en dinars.
<i>fraisDeLivraison</i>	C'est un attribut de type « number », il représente les frais de livraison de la commande en dinars.
<i>totalCommande</i>	C'est un attribut de type « number », il représente le montant total de la commande en dinars.
<i>remarque</i>	C'est un attribut de type « string », il représente une remarque pour le vendeur pour cette commande.

TABLE 3.7 – Description relative aux attributs de la collection « Commande »

Le tableau 3.8 contient une description des différents attributs du type « Adresse »

Attribut	Description
<i>latitude</i>	C'est un attribut de type « number », il représente la latitude d'une position géographique.
<i>longitude</i>	C'est un attribut de type « number », il représente la longitude d'une position géographique.
<i>detailsAdresse</i>	C'est un attribut de type « string », il représente une description détaillée de l'adresse.

TABLE 3.8 – Description relative aux attributs du type « Adresse »

Le tableau 3.9 contient une description des différents attributs de la collection « DemandeOuvertureCompte »

Attribut	Description
<i>idDemande</i>	C'est un attribut de type « string », il représente l'identifiant de la demande (clé primaire).
<i>nom</i>	C'est un attribut de type « string », il représente le nom du vendeur ou livreur.
<i>prenom</i>	C'est un attribut de type « string », il représente le prénom du vendeur ou livreur.
<i>email</i>	C'est un attribut de type « string », il représente l'adresse mail du vendeur ou livreur.
<i>numeroTelephone</i>	C'est un attribut de type « string », il représente le numéro de téléphone du vendeur ou livreur.
<i>dateDemande</i>	C'est un attribut de type « timestamp », il représente la date de création de la demande.
<i>estAccepter</i>	C'est un attribut de type « boolean », il représente le statut de la demande si elle est acceptée ou non.

TABLE 3.9 – Description relative aux attributs de la collection « DemandeOuverture-Compte »

Le tableau 3.10 contient une description des différents attributs de la collection « Livreur »

Attribut	Description
<i>idLivreur</i>	C'est un attribut de type « string », il représente l'identifiant du livreur (clé primaire).
<i>nom</i>	C'est un attribut de type « string », il représente le nom du livreur.
<i>prenom</i>	C'est un attribut de type « string », il représente le prénom du livreur.
<i>urlImageProfile</i>	C'est un attribut de type « string », il représente l'URL de la photo du livreur.

<i>email</i>	C'est un attribut de type « string », il représente l'adresse mail du livreur.
<i>numeroTelephone</i>	C'est un attribut de type « string », il représente le numéro de téléphone du livreur.
<i>villeDeLivraison</i>	C'est un attribut de type « string », il représente la ville ou le livreur travaille.
<i>moyenDeLivraison</i>	C'est un attribut de type « string », il représente le moyen de livraison « vélo », « moto » ou « voiture ».
<i>enService</i>	C'est un attribut de type « boolean », il représente l'état du livreur s'il est en service ou hors service.
<i>statutCompte</i>	C'est un attribut de type « number », il représente le statut du compte du livreur « activé » ou « désactivé ».
<i>idCommandeEnCours</i>	C'est un attribut de type « string », il représente l'identifiant de la commande (clé étrangère de la collection Commande) qu'elle est en cours de livraison.

TABLE 3.10 – Description relative aux attributs de la collection « Livreur »

Le tableau 3.11 contient une description des différents attributs de la Cl. «Vendeur»

Attribut	Description
<i>idVendeur</i>	C'est un attribut de type « string », il représente l'identifiant du vendeur (clé primaire).
<i>nom</i>	C'est un attribut de type « string », il représente le nom du vendeur.
<i>prenom</i>	C'est un attribut de type « string », il représente le prénom du vendeur.
<i>email</i>	C'est un attribut de type « string », il représente l'adresse mail du vendeur.
<i>numeroTelephone</i>	C'est un attribut de type « string », il représente le numéro de téléphone du vendeur.
<i>idMagasin</i>	C'est un attribut de type « string », il représente l'identifiant du magasin (clé étrangère de la collection Magasin) que le vendeur possède.

TABLE 3.11 – Description relative aux attributs de la collection « Vendeur »

3.4 Description de la vue dynamique

3.4.1 Diagramme de séquence relatif à « S’authentifier »

A travers ce diagramme, nous allons décrire le scénario du cas d’utilisation « S’authentifier ». Dans un premier lieu, le Client saisi son adresse mail et son mot de passe et clique sur le bouton « Connexion ». Ensuite le système vérifie la validité des données entrées pour ensuite authentifier le Client.

La figure 3.5 illustre une description détaillée du scénario relatif au cas d’utilisation « S’authentifier »

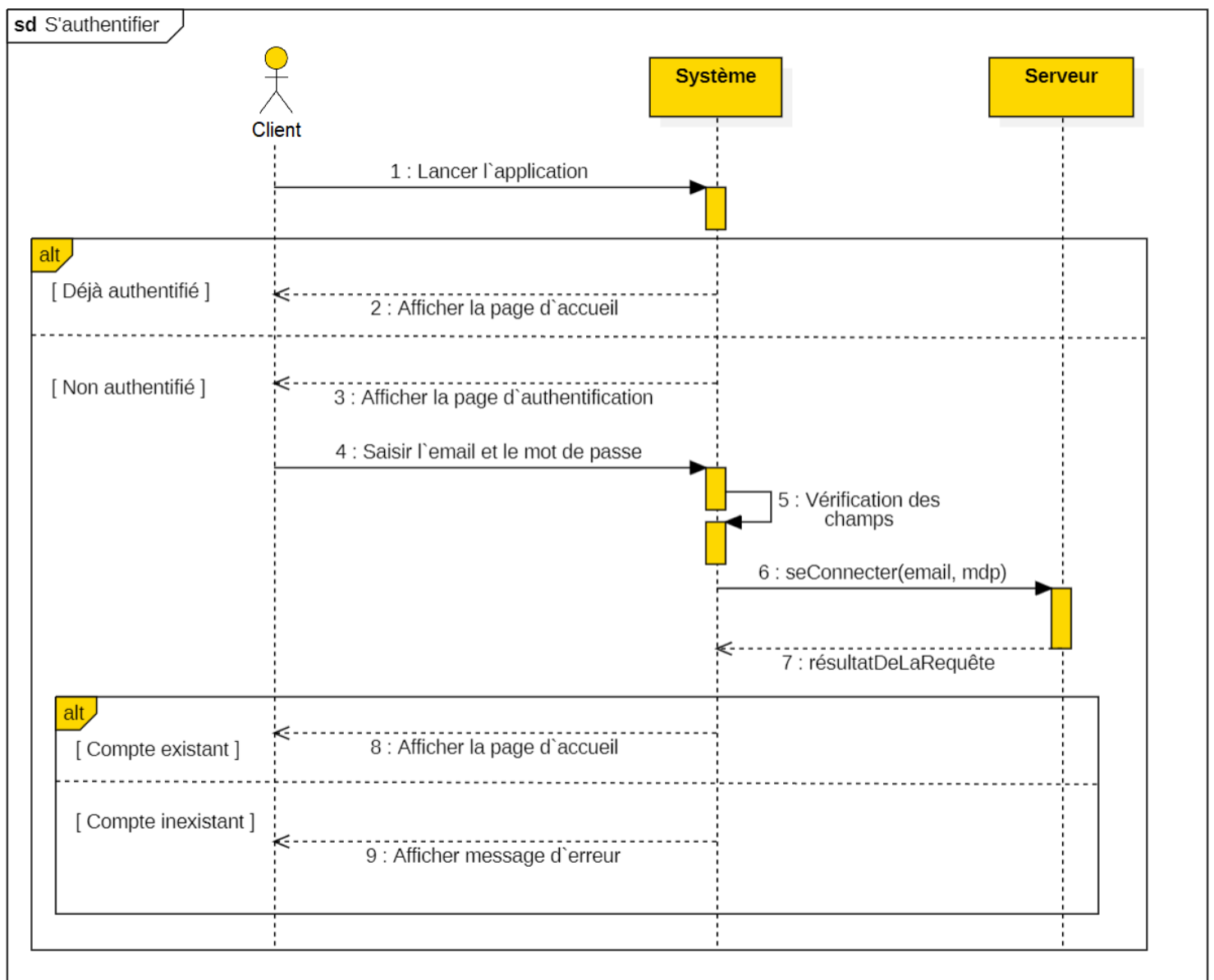


FIGURE 3.5 – Diagramme de séquence relatif à « S’authentifier »

3.4.2 Diagramme de séquence relatif à « Ajouter un article au panier »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Ajouter un article au panier ». Sur l'interface d'accueil le Client clique sur un magasin, le système affiche le menu du magasin, ensuite le Client choisit article et sélectionne la quantité et les options si disponibles et clique sur « Ajouter au panier ». Le système vérifie si un panier du même magasin existe ou pas pour remplir le panier existant ou créer un nouveau panier. La figure 3.6 illustre une description détaillée du scénario relatif au cas d'utilisation « Ajouter un article au panier »

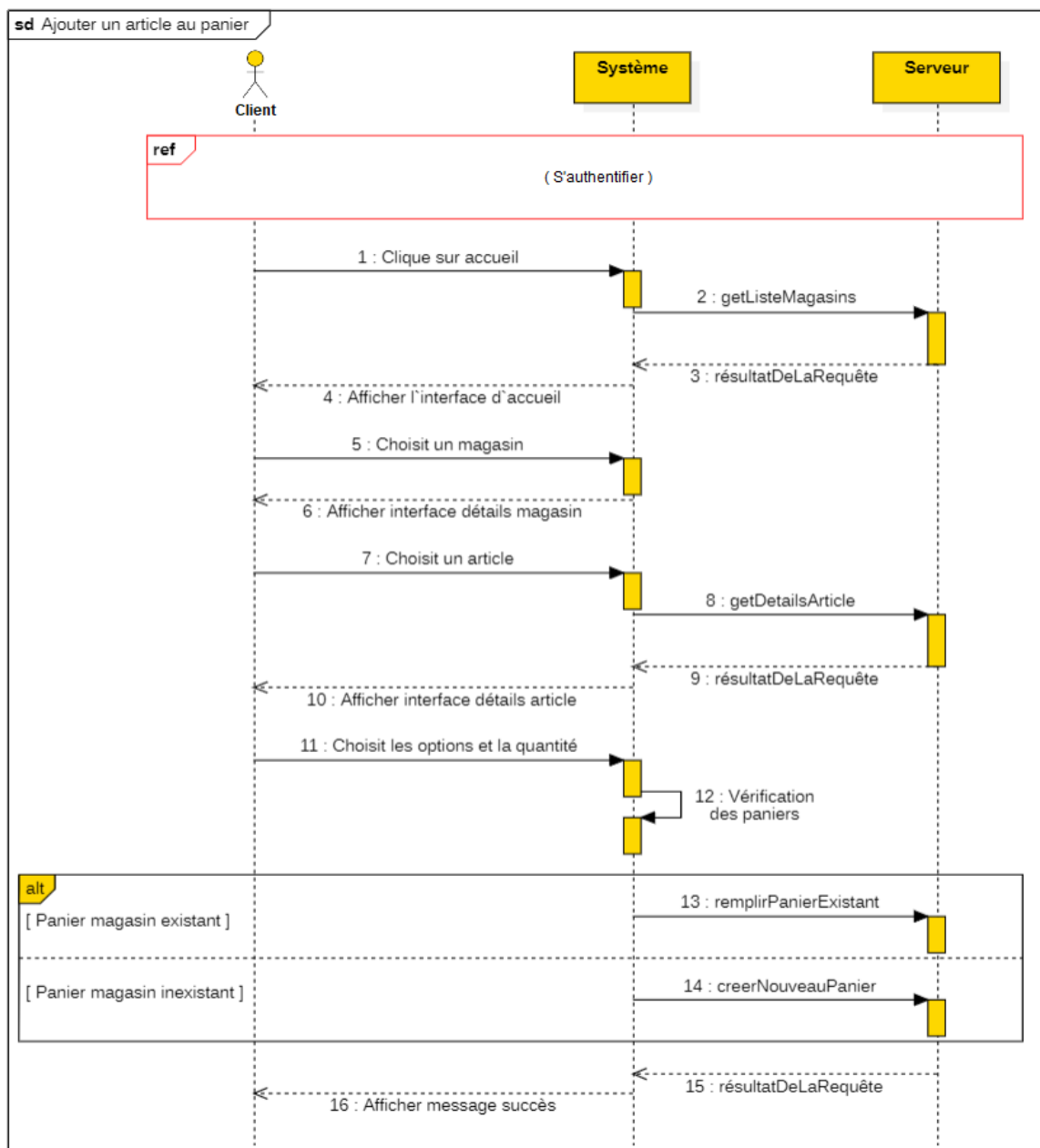


FIGURE 3.6 – Diagramme de séquence relatif à « Ajouter un article au panier »

3.4.3 Diagramme de séquence relatif à « Passer une commande »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Passer une commande ». Sur l'interface des paniers le Client clique sur un panier, le système affiche les détails du panier sur l'interface « Passer une commande », ensuite le Client sélectionne la position de l'adresse de livraison et clique sur commander. Le système vérifie si l'adresse sélectionnée se situe dans les villes disponibles sur le système, si oui le système crée une nouvelle commande, sinon il affiche un message au Client.

La figure 3.7 illustre une description détaillée du scénario relatif au cas d'utilisation « Passer une commande »

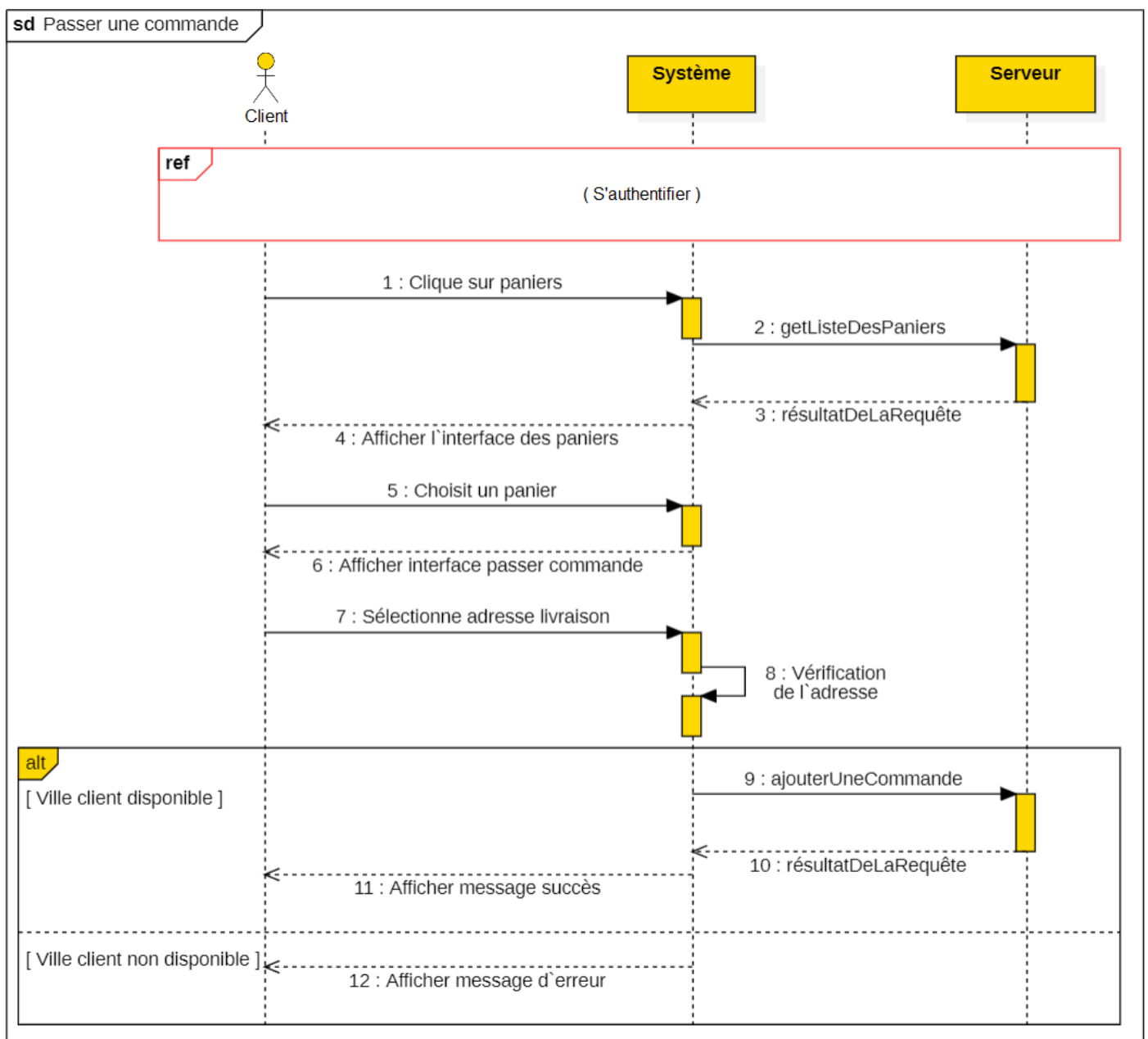


FIGURE 3.7 – Diagramme de séquence relatif à « Passer une commande »

3.4.4 Diagramme de séquence relatif à « Créer un compte vendeur »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Créer un compte Vendeur ». Dans un premier lieu, le Vendeur clique sur « créer un compte » sur l'interface d'authentification, le système affiche le formulaire de création de compte, ensuite le Vendeur remplit le formulaire et clique sur « créer un compte ». Le système vérifie la validité des données entrées et ensuite créé une nouvelle demande d'ouverture de compte Vendeur.

La demande sera traitée par un Administrateur, s'il l'accepte le système créé un nouveau compte Vendeur, sinon le système informe le Vendeur que la demande a été refusée.

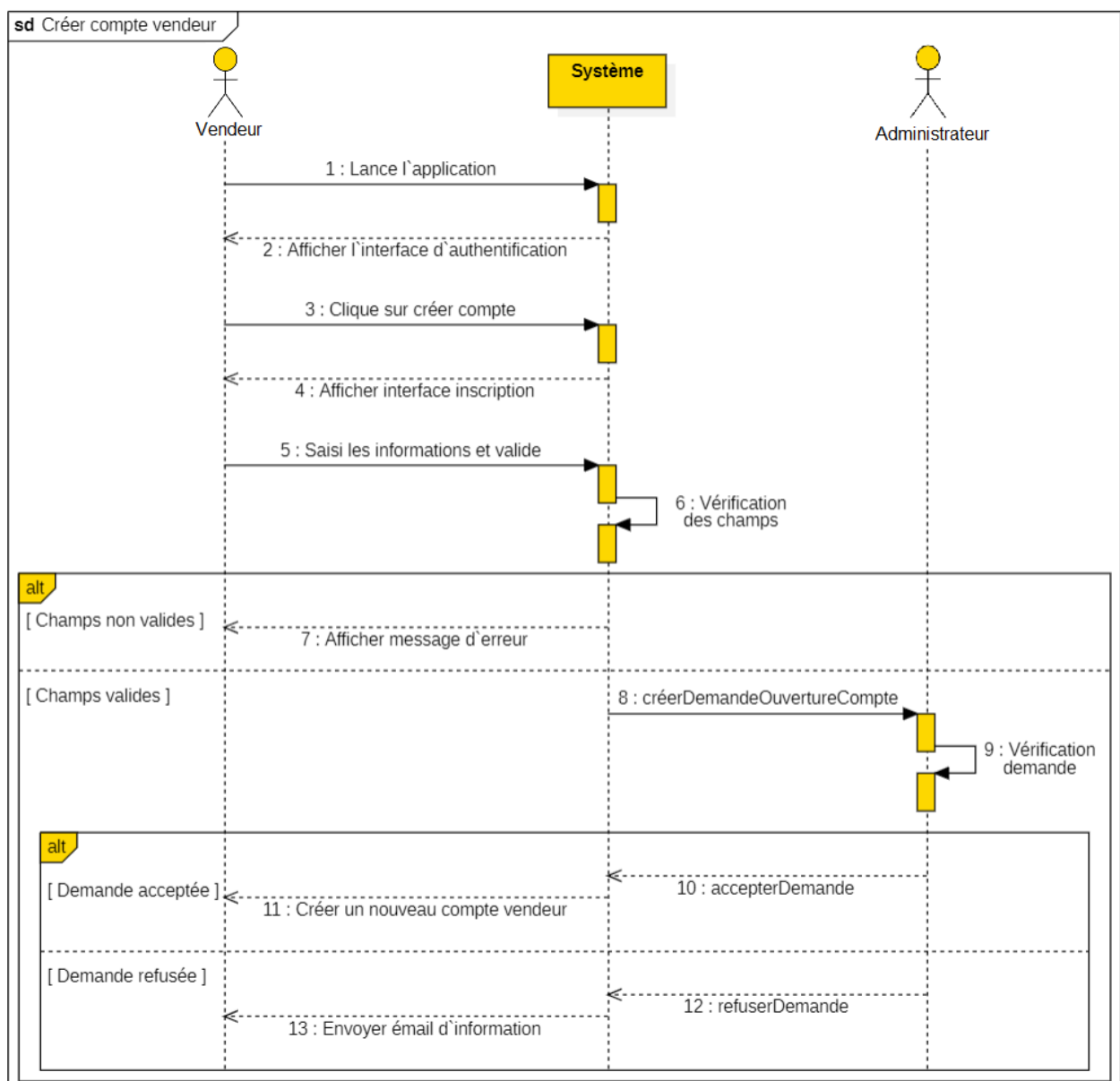


FIGURE 3.8 – Diagramme de séquence relatif à « Créer un compte vendeur »

3.4.5 Diagramme de séquence relatif à « Consulter une nouvelle commande »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Consulter une nouvelle commande ». Sur l'interface des nouvelles commandes le Vendeur choisit une commande et le système affiche les détails de la commande, après le Vendeur examine la commande et clique sur accepter ou annuler, ensuite le système change le statut de la commande selon la décision du Vendeur.

La figure 3.9 illustre une description détaillée du scénario relatif au cas d'utilisation « Consulter une nouvelle commande »

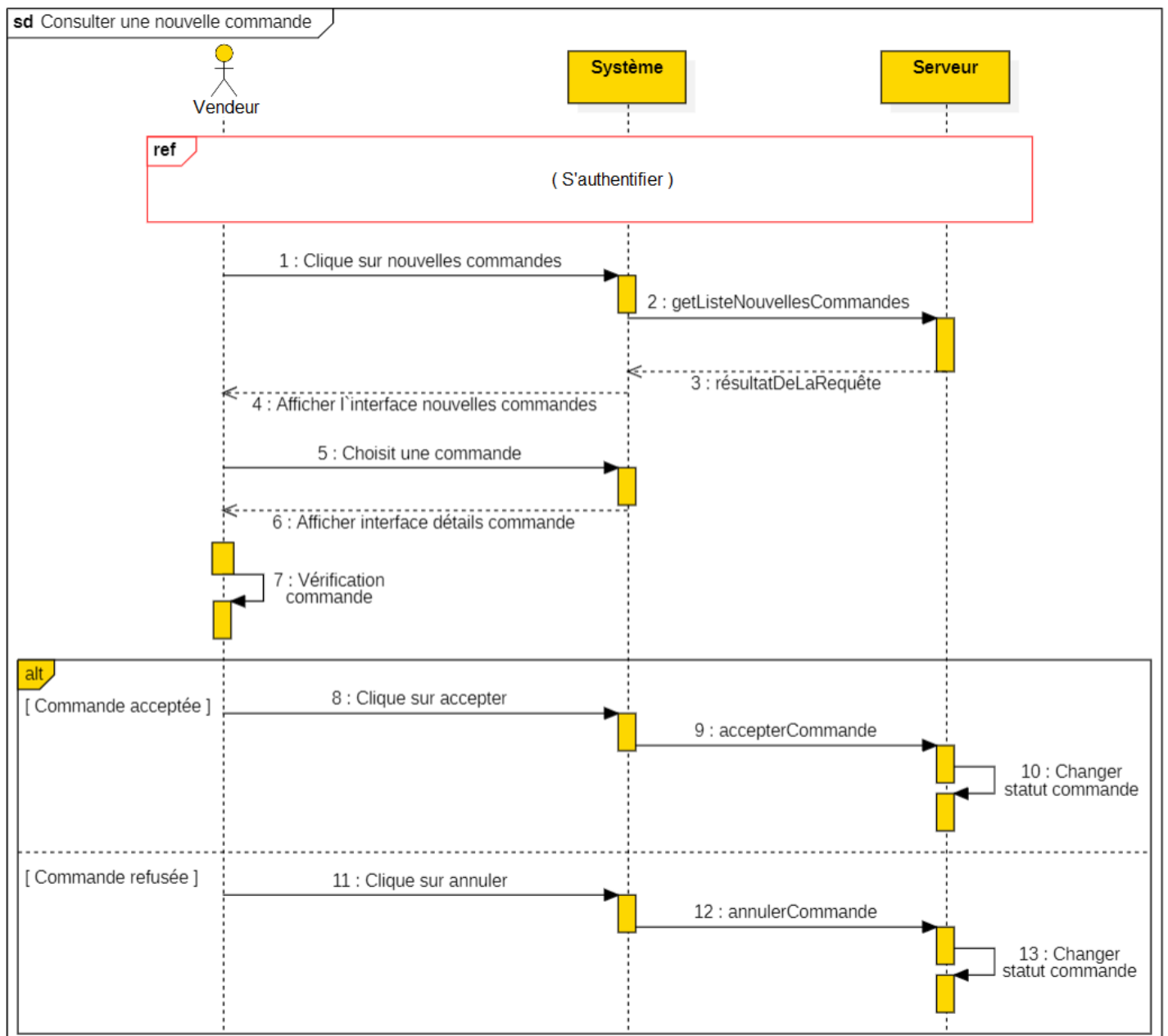


FIGURE 3.9 – Diagramme de séquence relatif à « Consulter une nouvelle commande »

3.4.6 Diagramme de séquence relatif à « Suspendre la réception des commandes »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Suspendre la réception des commandes ». Sur l'interface d'accueil le Vendeur clique sur l'icône « Pause », le système affiche un Dialog de confirmation au Vendeur s'il veut vraiment suspendre la réception des commandes, le Vendeur clique sur confirmer, ensuite le système change le statut du magasin à « non disponible » et empêche les clients de passer des commandes chez le magasin.

La figure 3.10 illustre une description détaillée du scénario relatif au cas d'utilisation « Suspendre la réception des commandes »

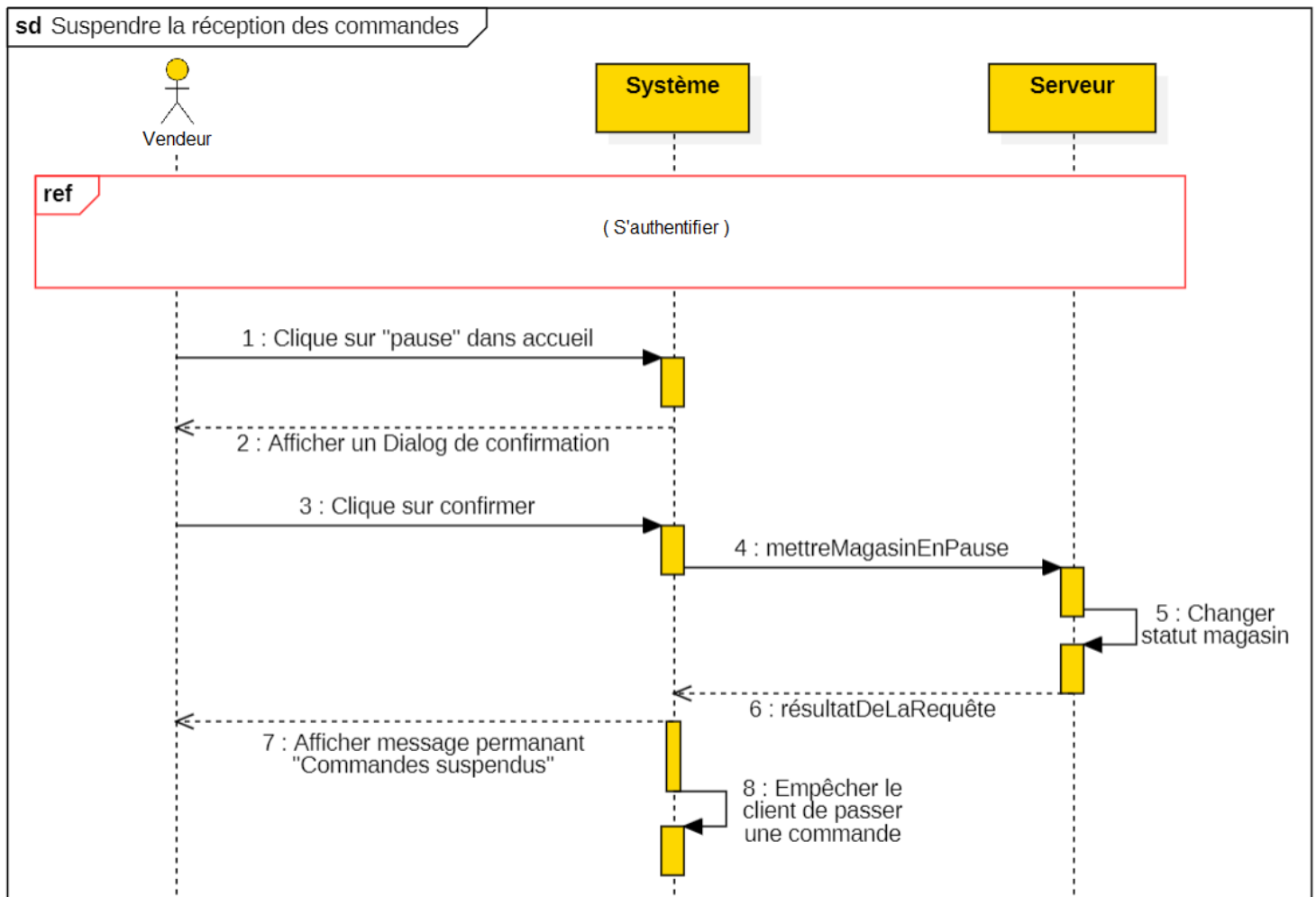


FIGURE 3.10 – Diagramme de séquence relatif à «Suspendre la réception des commandes»

3.4.7 Diagramme de séquence relatif à « Scanner un code QR »

A travers ce diagramme, nous allons décrire le scénario du cas d'utilisation « Scanner un code QR ». Sur l'interface « Commande en cours » le Livreur clique sur « Scanner », le système lance la caméra et commence la détection des codes QR, une fois un code QR est scanné, le système vérifie si ce code est identique au ID du Magasin (ou Client), si oui le système change le statut de la commande, sinon le système affiche un message que le code QR n'est pas valide.

La figure 3.11 illustre une description détaillée du scénario relatif au cas d'utilisation « Scanner un code QR »

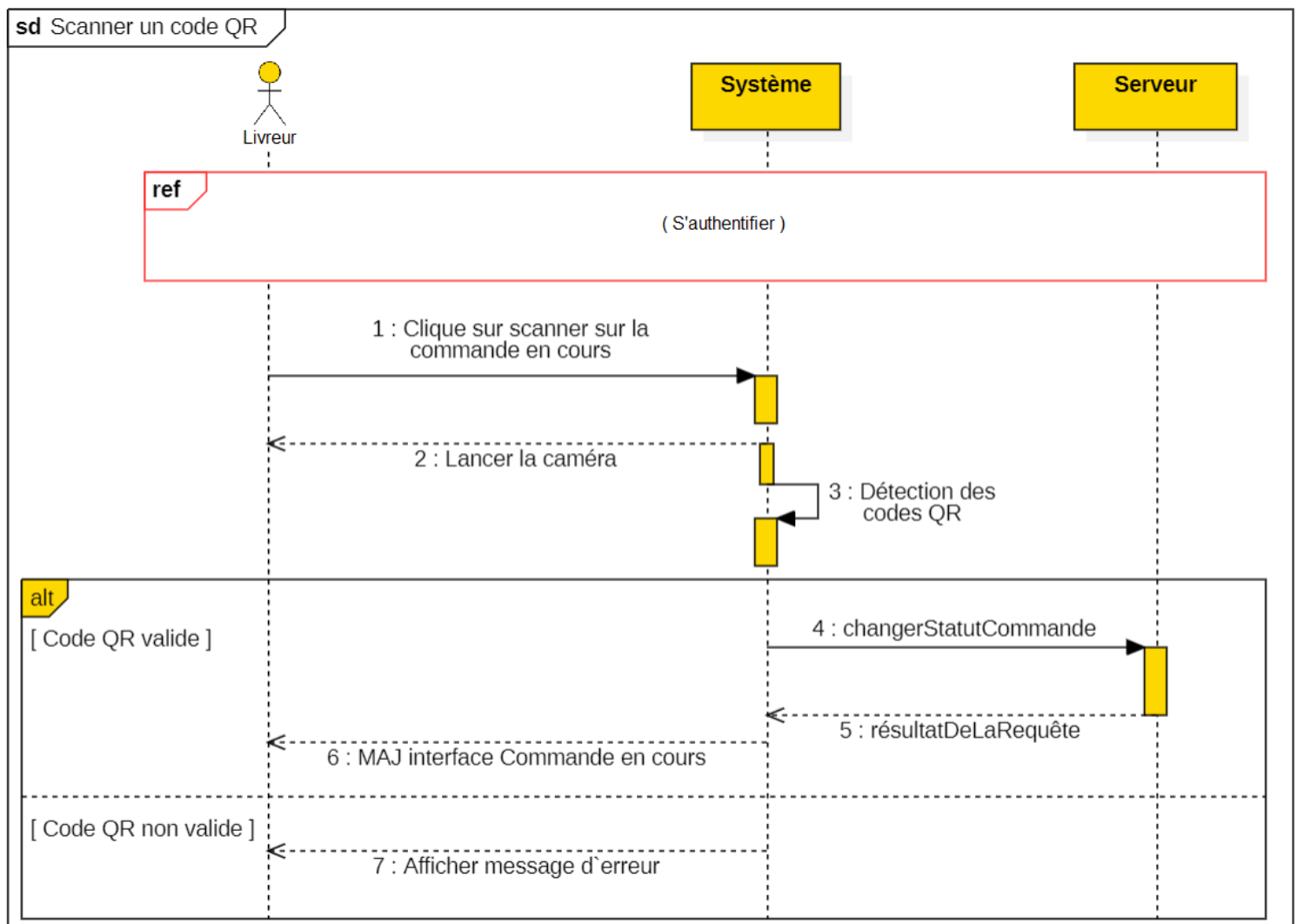


FIGURE 3.11 – Diagramme de séquence relatif à « Scanner un code QR »

3.5 Conclusion

Dans l'étape de la conception, nous avons détaillée le fonctionnement de notre application d'une manière globale afin de faciliter l'implémentation. Dans le prochain chapitre nous allons présenter la phase de réalisation de l'ensemble des applications et leurs outils de développement, ainsi que l'observation des résultats obtenus.

Chapitre **4**

Réalisation

4.1 Introduction

Une fois la partie de la conception achevée, tous les éléments nécessaires au développement des applications deviennent disponibles. Dans ce chapitre nous allons présenter dans un premier lieu l'environnement matériel et logiciel de développement, par la suite, nous décrirons la phase d'implémentation en nous basant sur quelques interfaces.

4.2 Environnement de travail

Nous montrons dans cette partie le matériel et les logiciels utilisés pour la réalisation

4.2.1 Environnement matériel

Nous avons programmé l'ensemble des applications sur un ordinateur ayant les caractéristiques suivantes :

- Marque : HP Omen.
- Système d'exploitation : Windows 11 Pro.
- Processeur : Intel ® Core™ i5 (8th Gen) up to 4.0GHz.
- Mémoire vive : 16 Go.
- Disque dur : 500 Go SSD.
- Carte graphique : NVIDIA GeForce GTX 1050 Ti.

Pour les différentes étapes de test, d'installation et le déploiement de l'ensemble des applications nous avons eu besoin d'une terminale mobile supportant le système d'exploitation Android 11. dont les caractéristiques sont les suivantes :

- Nom de l'appareil : Xiaomi Redmi Note 10 Pro
- Processeur : Snapdragon 732G Qualcomm.
- Mémoire vive : 6 Go.
- Stockage : 128 Go.
- Taille d'écran : 6.67 pouces.
- Caméra : 108 Mpx 4 capteurs.

4.2.2 Environnement logiciel

Les différents systèmes d'exploitation pour les smartphones :

Il y a une concurrence intense dans le domaine des systèmes d'exploitation pour les appareils mobiles, voici une liste des concurrents les plus célèbres et puissants dans ce domaine :

- **Android** : est le système d'exploitation mobile créé par Google. Il équipe la majorité des téléphones portables du moment (smartphones). Son principal concurrent est Apple avec l'iPhone.
- **iPhone OS** : il s'agit bien sûr du système présent sur les différentes générations d'iPhone, produits par Apple mais également sur ses tablettes iPad.
- **Windows Phone** : tout autant propriétaire, le système pour mobiles proposé par Microsoft est distribué sur de nombreux téléphones.
- **HarmonyOS** : le nouveau système d'exploitation intelligent et universel développé par Huawei, sa première version est sortie en 2019.
- **KaiOS** : est un système d'exploitation mobile destiné aux téléphones non tactiles, lancé en 2017 par l'entreprise KaiOS Technologies.
- **BlackBerry OS** : il est présent sur tous les téléphones de la marque RIM (Research In Motion).

Et bien d'autres systèmes plus marginaux, souvent développés pour des appareils spécifiques (PDA, etc.). [11]

Définition d'Android

" Android est un système d'exploitation mobile fondé sur le noyau Linux et développé par des informaticiens sponsorisés par Google.

Suite du rachat par Google en 2005 de la startup du même nom, le système avait d'abord été lancé en juin 2007 pour les smartphones et tablettes tactiles, avant de se diversifier dans les objets connectés, ordinateurs comme les télévisions (Android TV), les voitures (Android Auto), les Chromebook (Chrome OS qui utilise les applications Android) et les smartwatch (Wear OS).

En 2015, Android est le système d'exploitation mobile le plus utilisé dans le monde, devant iOS d'Apple, avec plus de 80 % de parts de marché dans les smartphones pour l'ensemble de ses versions et adaptations ". [12]



FIGURE 4.1 – Logo Android OS

Architecture du système Android

Le système d'exploitation Android est basé sur une architecture à quatre niveaux, comme le montre la figure 4.2, on peut citer quelques composants :

- Un noyau Linux qui lui confère notamment des caractéristiques multitâches.
- Des bibliothèques graphiques, multimédias.
- Une machine virtuelle Java adaptée : la Dalvik Virtual Machine.
- Un Framework applicatif proposant des fonctionnalités de gestion de fenêtres, de téléphonie, de gestion de contenu...
- Des applications dont un navigateur web, une gestion des contacts, un calendrier... [13]

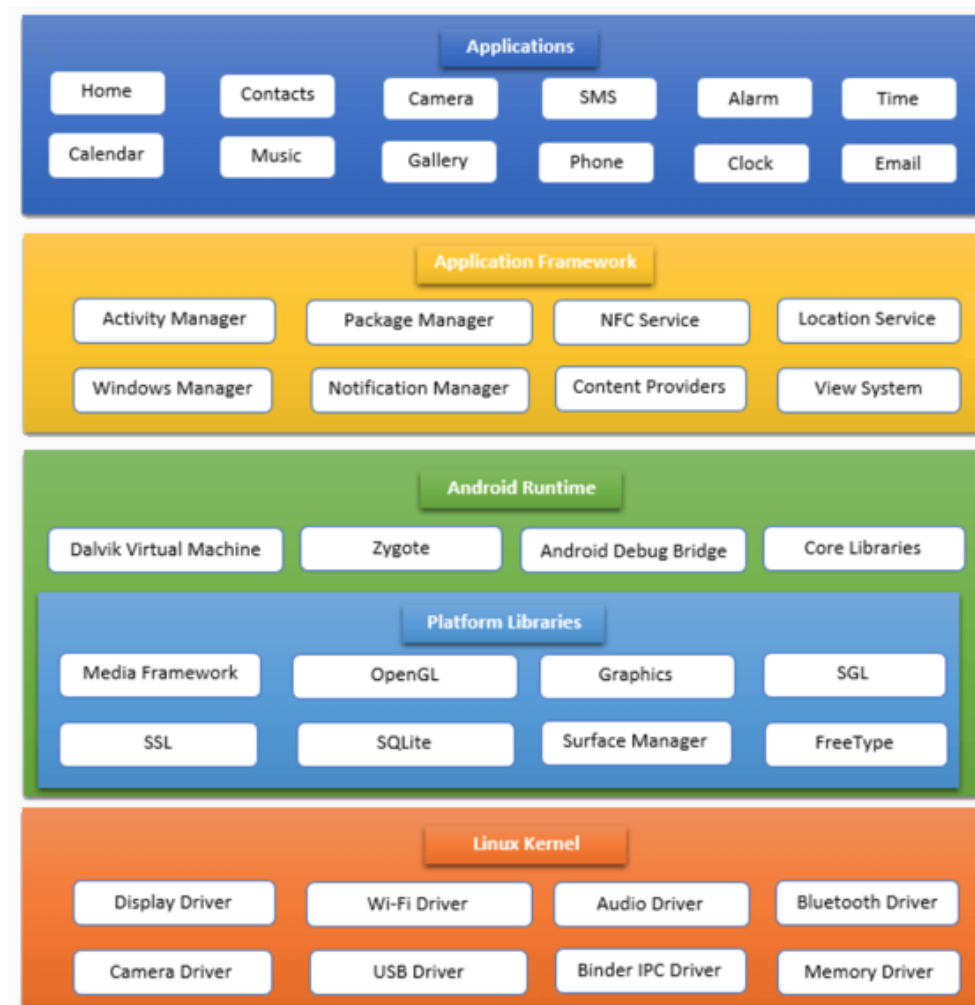


FIGURE 4.2 – Les composants du système Android

Définition d'application

" Les applications Android sont des applications fonctionnant sur la plate-forme Android. Parce que la plate-forme Android est conçue pour les appareils mobiles, une application Android typique est conçue pour un smartphone ou une tablette PC fonctionnant sur le système d'exploitation Android ". [14]

4.2.3 Les types d'applications

Applications Natives

Une application native est une application mobile qui est développée spécifiquement pour un des systèmes d'exploitation utilisé par les Smartphones et tablettes (iOS, Android, Windows Phone, etc.). Elle est développée avec un langage spécifique à son système

d'exploitation et ne peut être distribuée que par l'intermédiaire des plateformes d'applications qui contrôlent sa nature Et c'est ce que nous allons faire dans ce projet.

Le fait de développer une application native permet généralement d'utiliser toutes les fonctionnalités liées au système d'exploitation visé (GPS, accéléromètre, appareil photo, etc.) ainsi que la mémoire du Smartphone. En plus, ceci permet également de proposer des applications généralement plus riches en termes de fonctionnalité, de qualité, de performance et de résolution que les applications web ou les applications hybrides.

Applications Web

" Une application web est une application mobile développée en HTML accessible et exécutable par le biais d'un navigateur Internet pour téléphone portable. Elle utilise le navigateur du Smartphone et ne nécessite pas forcément de télécharger l'application ".
[15]

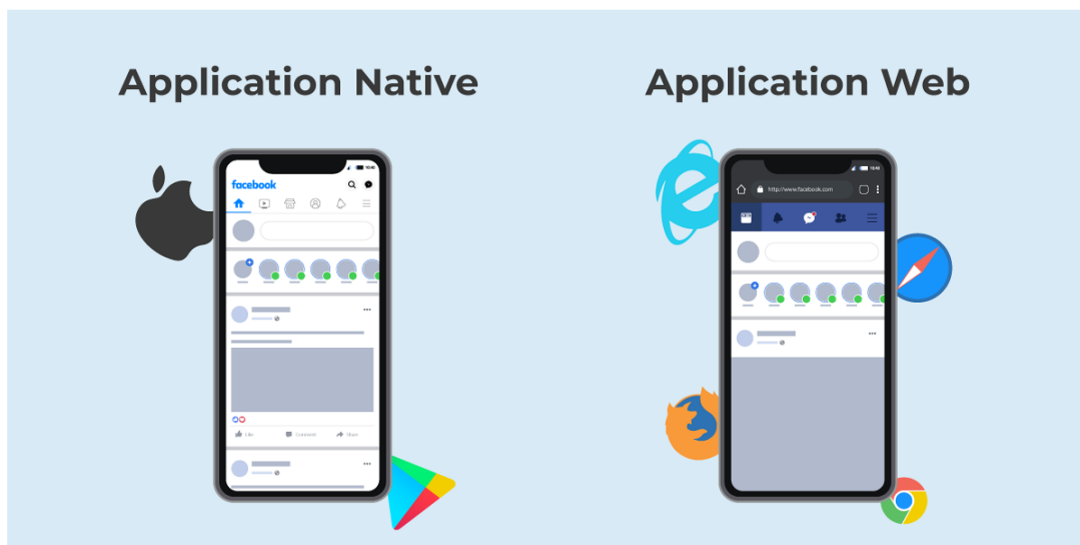


FIGURE 4.3 – Exemple de type native et de type web sur une même application

Applications Hybrides

Une application hybride est une application pour mobiles qui combine des éléments HTML5 sous forme d'application web et des éléments d'une application native, permettant d'utiliser les fonctionnalités natives des Smartphones et d'être distribuée en tant qu'application sur les plateformes d'App Store, de GooglePlay, etc.

4.3 Les outils utilisés

Lors du développement de l'ensemble d'applications nous avons utilisés des fonctionnalités, des services et des plugins, afin d'améliorer le résultat et faciliter le travail, on va les mentionner ci-dessous :

4.3.1 Android Studio

" Android Studio est un environnement de développement intégré (IDE) pour le développement sur la plateforme Android. Il a été annoncé en mai 2013. Android est disponible librement sous la licence Apache 2.0. Basé sur le logiciel IDEA de JetBrains 'IntelliJ, Android Studio est conçu spécifiquement pour le développement Android. Il est disponible en téléchargement sur les systèmes d'exploitation ; Windows, Mac OS et Linux ".[16]

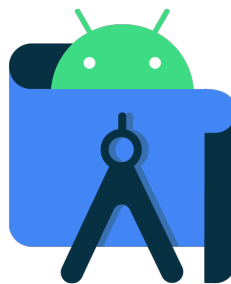


FIGURE 4.4 – Logo Android Studio

4.3.2 Java Development Kit (JDK)

Le Java Development Kit (JDK) désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.



FIGURE 4.5 – Logo JDK

4.3.3 Android Software Development Kit (SDK)

Le SDK est un ensemble d'outils que met à disposition Google afin de nous permettre de développer des applications pour Android. Il est disponible pour Windows, MacOS X et linux et inclut des outils ainsi qu'un émulateur Android pour exécuter des applications.



FIGURE 4.6 – Logo SDK

4.3.4 Kotlin

" Kotlin est un langage de programmation orienté objet et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java, JavaScript, et vers plusieurs plateformes en natif. En 8 mai 2019, Kotlin devient officiellement le langage de programmation voulu et recommandé par le géant américain Google pour le développement des applications Android ". [17]



FIGURE 4.7 – Logo Kotlin

4.3.5 Firebase

" Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolv, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet ". [18]



FIGURE 4.8 – Logo Firebase

4.3.6 Android Product Flavors

" Pour simplifier, Android Product Flavor est une variante de votre application. Il est très utile lorsque vous souhaitez créer plusieurs versions de votre application. Cela signifie que vous pouvez générer différentes versions ou variantes de votre application à l'aide d'une seule base de code.

Product Flavors est une fonctionnalité puissante du plugin Gradle d'Android Studio pour créer des versions personnalisées de produits. Ils font partie de ce que nous appelons Build Variants ". [19]

4.3.7 MVVM (Model View View-Model)

" Model View View-Model est introduit dans Google I/O de l'année 2017. Ce plan architectural devient populaire par les fonctionnalités qu'il fournit. Il implémente principalement le Data Binding Framework, il permet de "lier" des vues à des champs sur un objet arbitraire. Lorsqu'un champ est mis à jour, le framework est notifié et la vue est mise à jour automatiquement.

L'architecture introduit une communication bidirectionnelle entre ses composants. Outre des fonctionnalités telles que la liaison, la mise à jour automatique des vues, il est également facile à des fins de test. La fonctionnalité de modèle et de vue est la même que celle dont nous avons discuté dans MVP.

View-Model : - Il est responsable de l'exposition des méthodes, des commandes et d'autres propriétés qui aident à maintenir l'état de la vue, à manipuler le modèle à la suite d'actions sur la vue et à déclencher des événements dans la vue elle-même. La vue a une référence à View-Model mais View-Model n'a aucune information sur la vue. Il est complètement indépendant de Views ". [20]

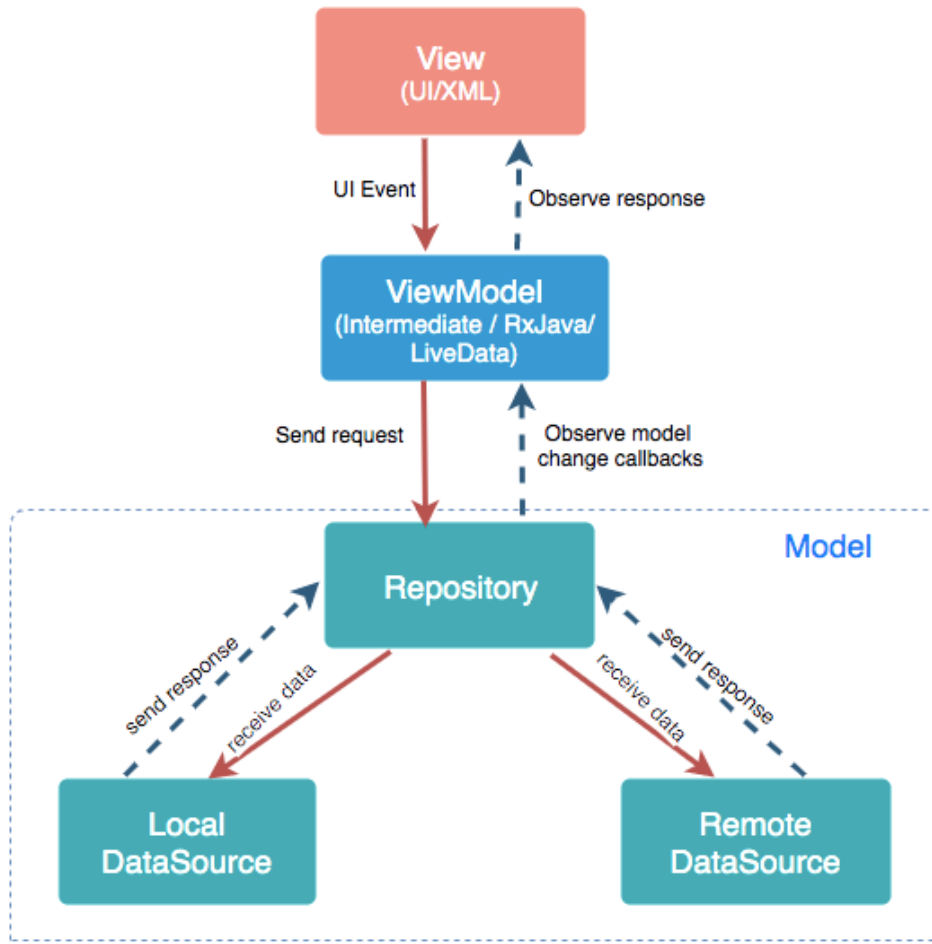


FIGURE 4.9 – Le modèle de conception MVVM

4.3.8 Material Design

Afin de faciliter la conception des interfaces et des champs de saisie ainsi que des boutons nous avons utilisés la bibliothèque Material Design. " C'est un système de conception créé par Google pour aider les développeurs à créer des expériences numériques de haute qualité pour Android, iOS, Flutter et le web ". [21]



FIGURE 4.10 – Logo Material Design

4.3.9 API de géolocalisation de Google

Le Google API est un kit de développement logiciel disponible de façon libre sur le web, qui offre la possibilité de créer de nouvelles applications qui utilisent la base de données des pages référencées et indexées par le géant Google et ceci grâce à des web services. [22]



FIGURE 4.11 – Logo Google Maps

L’affichage et l’utilisation de la carte Google sur notre application nécessitent l’utilisation de l’API de Google Maps. Pour pouvoir l’utiliser, il est indispensable de passer par les étapes suivantes :

Comme l’indique la figure ci-dessous, il faut d’abord accéder à la plateforme « Google Developers » et se placer sous la rubrique « Maps Android API » et cliquer sur le bouton « Obtenir une clé »

API ↑	Demandes	Erreurs	Latence moyenne (ms)
Maps SDK for Android	126	-	-
Places API	-	-	-

FIGURE 4.12 – Première étape de configuration de la carte Google Maps

En second lieu, il faut créer un nouveau projet ou chargé un ancien projet s’il existe afin d’enregistrer l’application sur la console d’API Google.

4.3.10 Dagger Hilt

L'injection de dépendances est une technique couramment utilisée en programmation et parfaitement adaptée au développement sur Android.

L'implémentation de cette technique vous garantit les avantages suivants :

- Code réutilisable
- Facilité de refactorisation
- Facilité de test

" Hilt est une bibliothèque d'injection de dépendances "orientée" pour Android qui permet de réduire la tâche répétitive d'injection manuelle de dépendances dans votre projet.

Hilt repose sur la bibliothèque d'injection de dépendances Dagger et bénéficie ainsi de l'exactitude du temps de compilation, des performances d'exécution, de l'évolutivité et de la compatibilité avec Android Studio qu'offre Dagger ". [23]

4.3.11 Flaticon

Les icônes et les animations sont des éléments importants dans toute application pour cela, nous avons utilisé Flaticon. Flaticon est le port d'attache de nombreux graphistes talentueux. Chacun travaille dans des styles différents et explore des concepts différents, de sorte que vous pouvez trouver les icônes qui conviennent parfaitement à votre projet. [24]



FIGURE 4.13 – Logo Flaticon

4.3.12 Zxing

Zxing est une bibliothèque de traitement d'image de code-barres 1D/2D open-source multi-format implémentée en Java, avec des ports vers d'autres langues. [25]

4.3.13 Code QR

" Un code QR, en forme longue Quick Response Code (code à réponse rapide), est un type de code-barres à deux dimensions constitué de modules-carrés noirs disposés dans

un carré à fond blanc. Ces points définissent l'information que contient le code. Ce dessin, lisible par machine, peut être visualisé sur l'écran d'un appareil mobile ou imprimé sur papier ". [26]

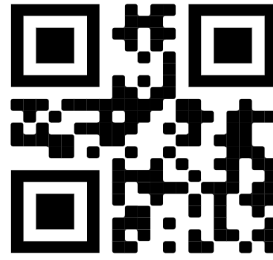


FIGURE 4.14 – Exemple de code QR

4.4 Interfaces des applications

Dans ce qui suit, nous présentons le fruit de notre travail en exposant les différentes interfaces de l'ensemble des applications que nous avons réussie à les mettre en place.

4.4.1 Application « Garanda Food »

Lorsque l'application se lance, une interface comme présenté sur la figure 4.15 ci-dessous se lance et invite le client à se connecter en introduisant son adresse mail et son mot de passe ou s'inscrire s'il ne possède pas encore un compte sur l'application.

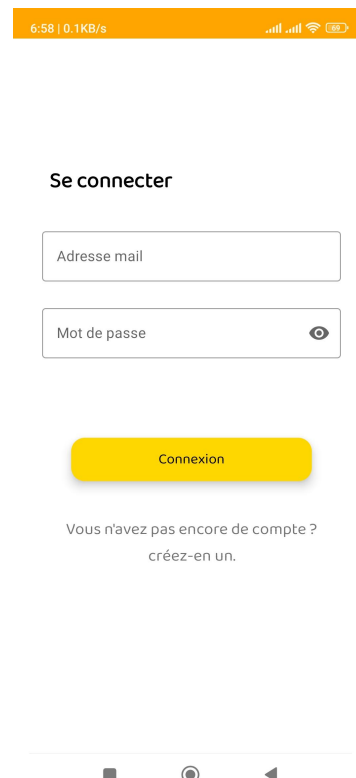


FIGURE 4.15 – Interface Connexion d'un client

Lorsque le client clique sur « Créer un compte », un formulaire d'inscription se lance pour lui donner la main pour introduire un ensemble d'informations afin d'avoir un compte sur l'application comme indiqué sur la figure 4.16 ci-dessous



The image shows a mobile application registration screen. At the top, there is a status bar with the time 7:00, data usage 0.2KB/s, and signal strength indicators. The main heading is "S'inscrire". Below it are four input fields: "Nom complet" with the value "Daci Ali", "Adresse mail" with "daci.ali.13@gmail.com", "numéro de téléphone" with "0558236963", and "Mot de passe" with masked characters and a visibility icon. A fifth field, "Confirmez le mot de passe", is highlighted with a yellow border and also contains masked characters and a visibility icon. Below the fields is a yellow button labeled "Créer un compte". At the bottom, there is a link: "Vous avez déjà un compte ? Se connectez". The Android navigation bar is visible at the very bottom.

FIGURE 4.16 – Interface Inscription d'un client

Après l'authentification du client, l'interface principale apparaît sur « Accueil » contenant la liste des magasins disponibles dans la ville du client, des publicités déroulantes, et un bouton de changement de ville comme indiqué sur la figure 4.17 ci-dessous

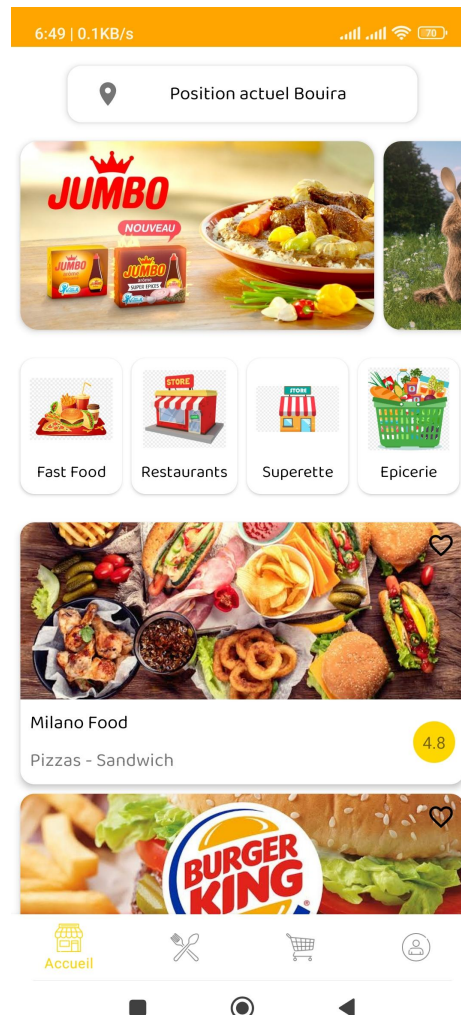


FIGURE 4.17 – Interface Accueil

Pour ajouter des articles au panier, le client clique sur un magasin, l'interface « détails magasin » apparaît contenant le menu et les informations du magasin comme indiqué sur la figure 4.18, ensuite il choisit un article ou un repas puis l'interface « détails articles » apparaît comme indiqué sur la figure 4.19 contenant une description sur l'article, une image, le prix et les options si disponibles, ensuite le client choisit les options et la quantité et enfin clique sur « Ajouter au panier ».

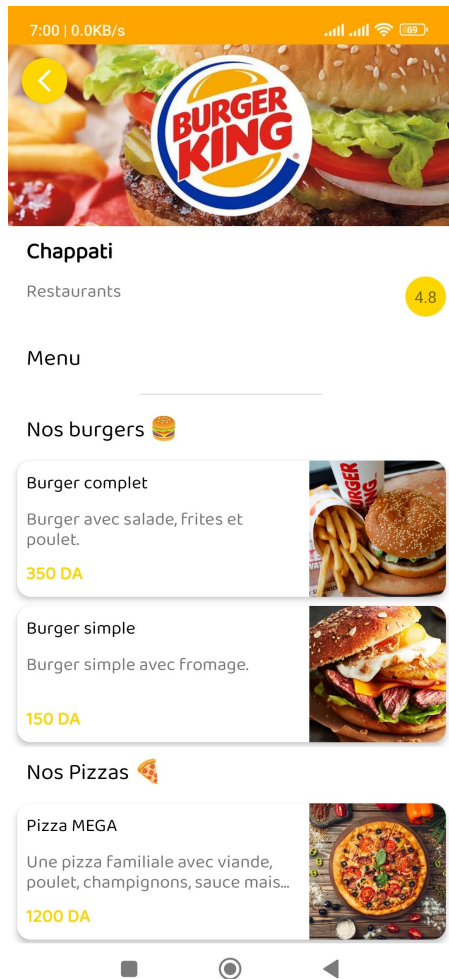


FIGURE 4.18 –
Interface Détails magasin

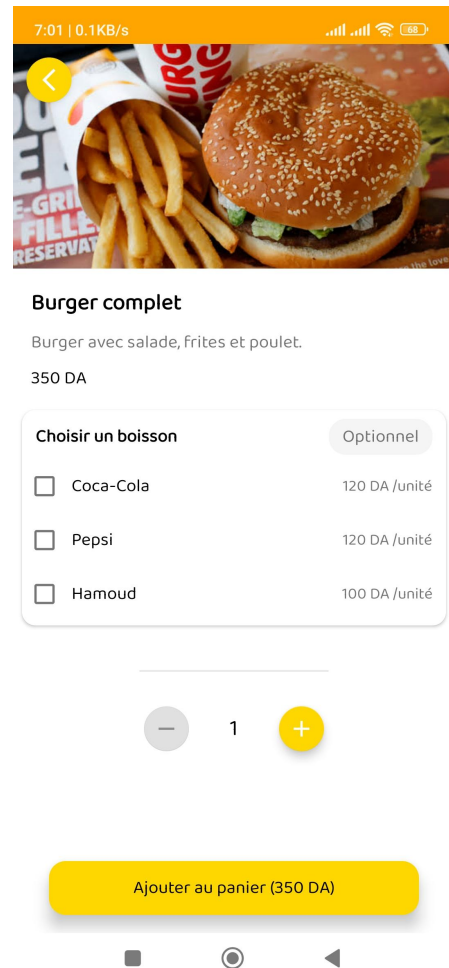


FIGURE 4.19 –
Interface Détails article

Lorsqu'un client veut passer une commande il clique sur « Paniers » dans l'interface principale comme indiqué sur la figure 4.20, ensuite l'interface « Passer commande » apparaît contenant les détails du panier ainsi les informations du client comme indiqué sur la figure 4.21 ci-dessous.

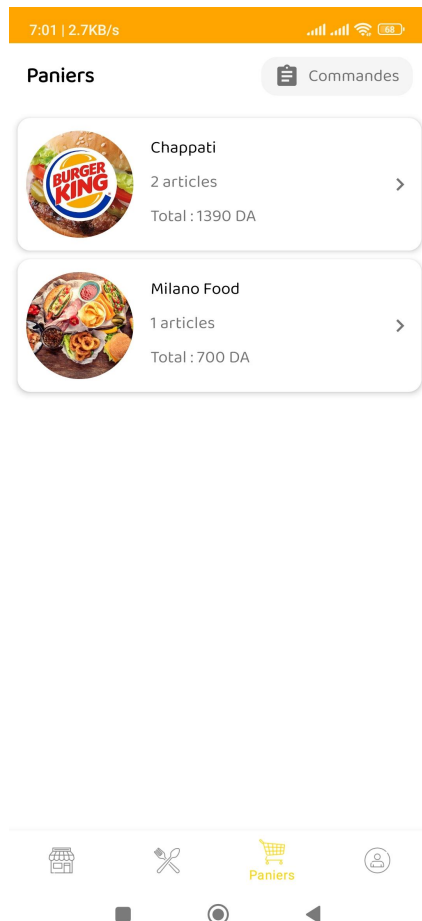


FIGURE 4.20 –
Interface Paniers

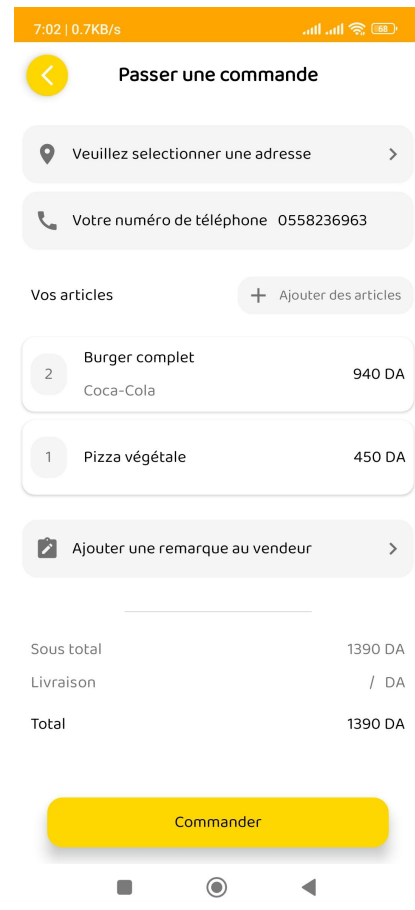


FIGURE 4.21 –
Interface Passer une commande

Le client doit sélectionner une adresse de livraison donc il clique sur « Sélectionner une adresse », l'interface « Maps » apparaît comme indiqué sur la figure 4.22, ensuite il positionne le pointeur sur son adresse et clique sur « Choisir cette position », et enfin il revient à l'interface « Passer une commande » et clique sur « Commander ».

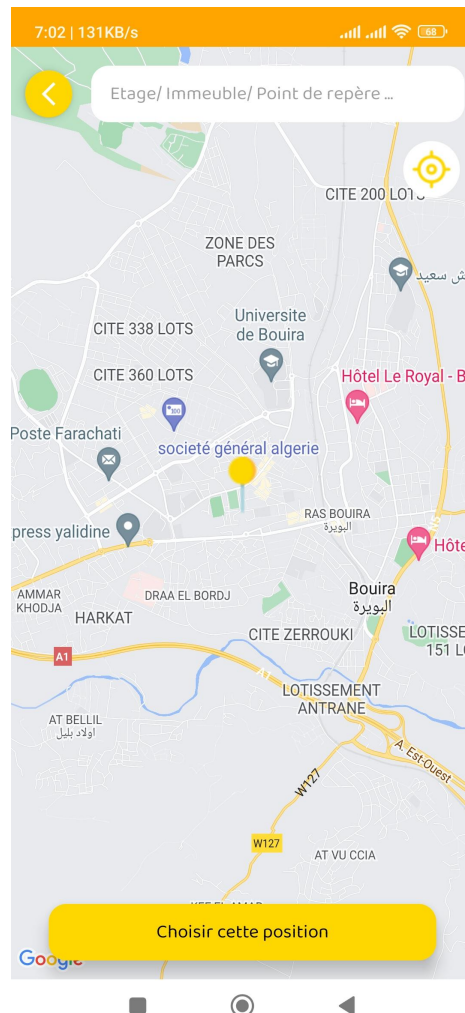


FIGURE 4.22 – Interface Maps

Un client peut consulter les commandes effectuées sur l'interface « Mes commandes », qui montre les détails d'une commande, son statut et son numéro unique ainsi d'autres informations comme indiqué sur la figure 4.23 et la figure 4.24 ci-dessous

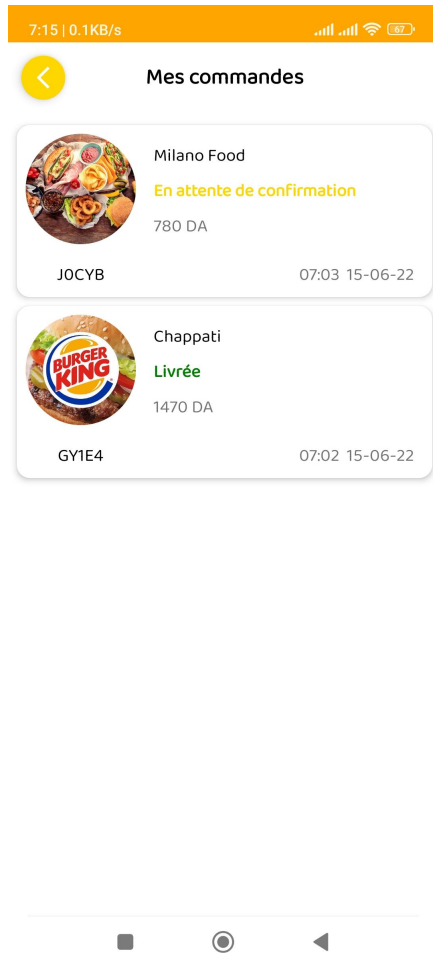


FIGURE 4.23 –
Interface Mes commandes

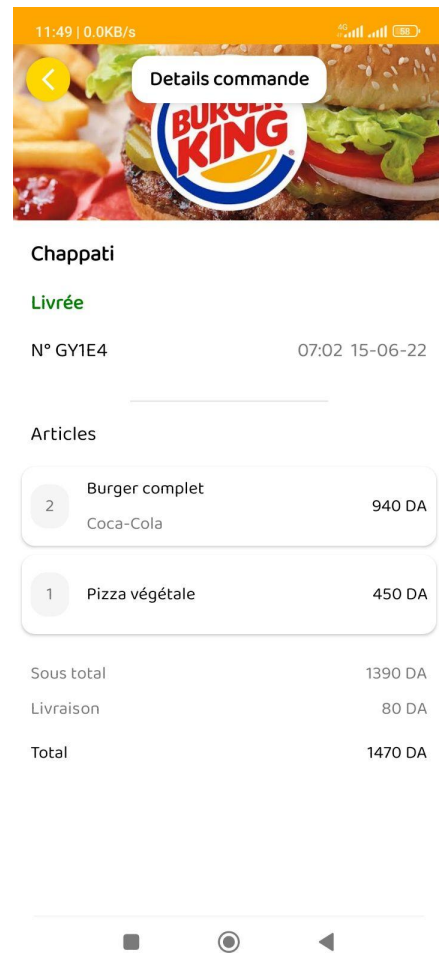


FIGURE 4.24 –
Interface Détails commande

Lorsqu'un client veut chercher des magasins par catégorie de cuisine il suffit de cliquer sur « Parcourir » dans l'interface principale et choisir une catégorie comme indiqué sur la figure 4.25 ci-dessous.

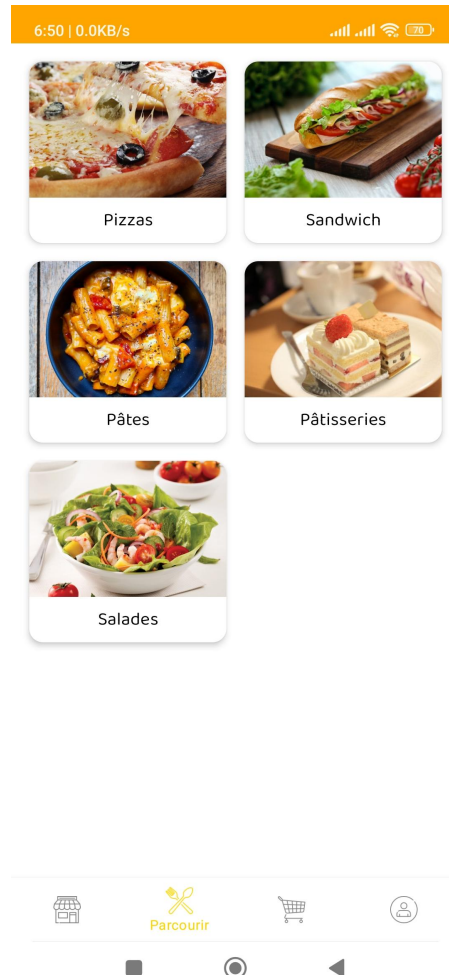


FIGURE 4.25 – Interface Parcourir magasins

Lorsqu'un client veut consulter son compte il clique sur « Compte » dans l'interface principale, ensuite il clique sur « Mon compte », l'interface « Mon compte » apparait contenant les informations du client comme indiqué sur la figure 4.26 et la figure 4.27 ci-dessous

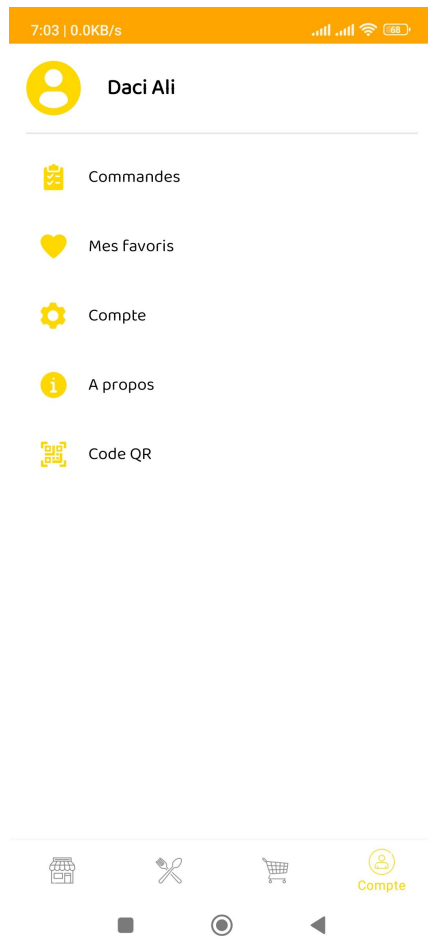


FIGURE 4.26 –
Interface Compte

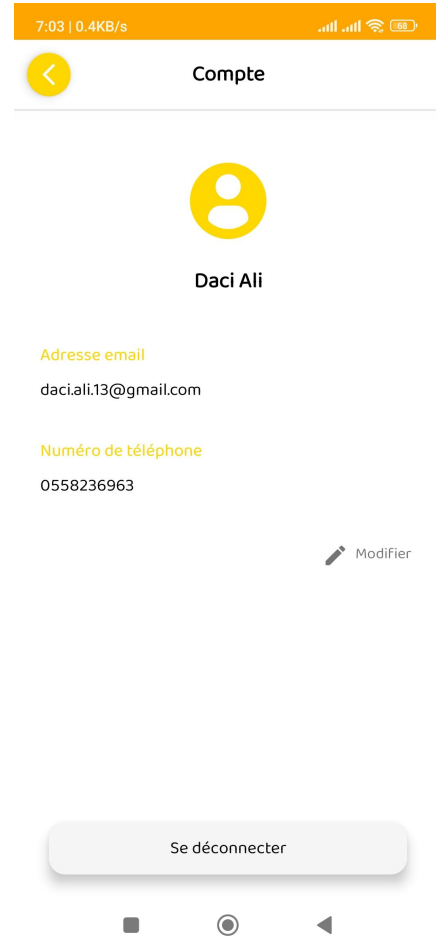


FIGURE 4.27 –
Interface Mon compte

4.4.2 Application « Garanda Food Vendeurs » :

Pour créer un nouveau compte le vendeur clique sur « Créer un compte » sur l'interface d'authentification, un formulaire de création de compte se lance comme indiqué sur la figure 4.28 ci-dessous, le vendeur remplit les champs et choisit l'adresse de son magasin, ensuite il clique sur le bouton « Créer un compte » pour envoyer une demande à un administrateur.

7:16 | 0.1KB/s

Nom vendeur

Prénom vendeur

Adresse mail

Numéro de téléphone

Nom du Magasin

Veillez sélectionner l'adresse du magasin

Type Magasin Fast Food

Créer un compte

Vous avez déjà un compte ?

FIGURE 4.28 – Interface Créer compte Vendeur

Lorsqu'un vendeur est authentifié une interface principale se lance au démarrage de l'application avec un menu déroulant qui englobe toutes les fonctionnalités offertes au vendeur comme indiqué sur la figure 4.29 et la figure 4.30 ci-dessous.

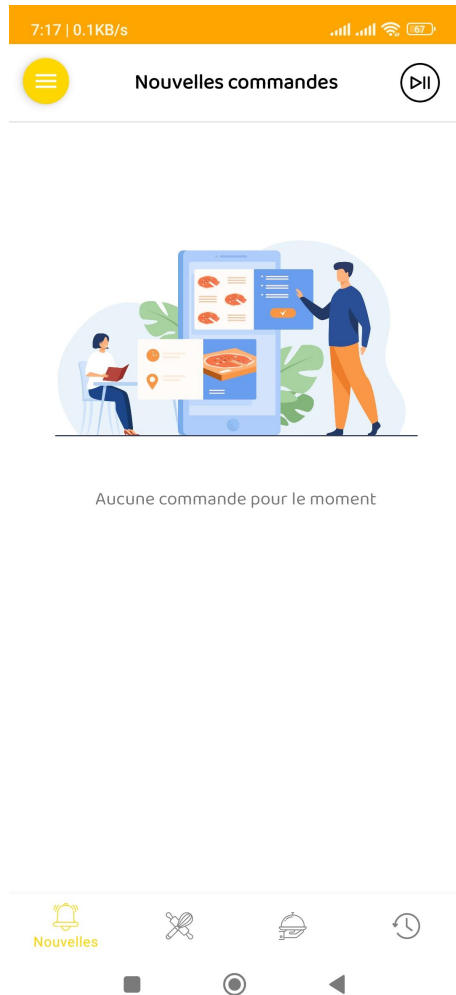


FIGURE 4.29 – Interface Principale

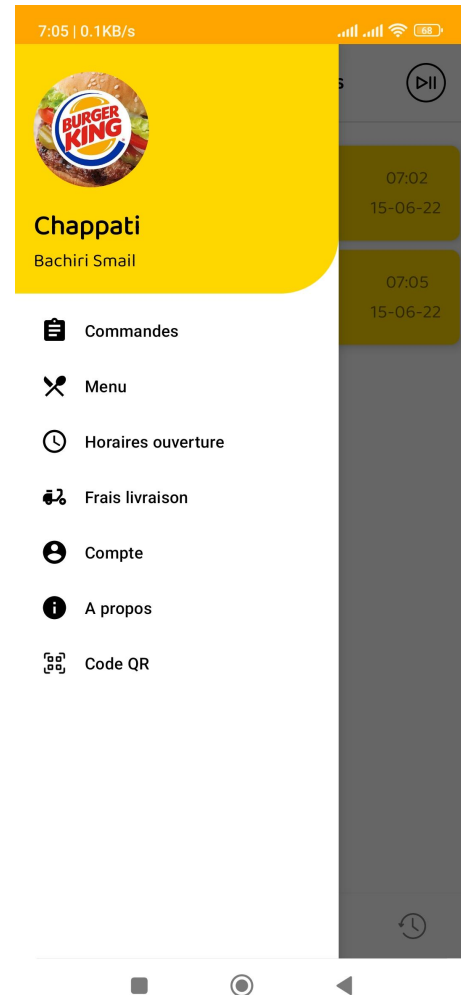


FIGURE 4.30 – Menu déroulant

Pour consulter une nouvelle commande le vendeur clique sur « Nouvelles commandes » dans l'interface principale pour voir les commandes en attente comme indiqué sur la figure 4.31, ensuite quand il clique sur une commande l'interface « Détails commande » apparaît contenant toutes les informations sur la commande comme indiqué sur la figure 4.32 ci-dessous.

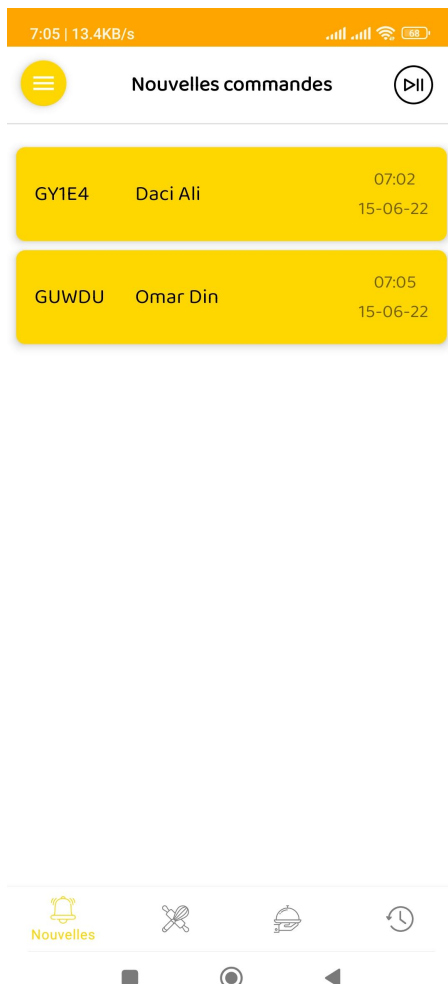


FIGURE 4.31 –
Interface Nouvelles commandes

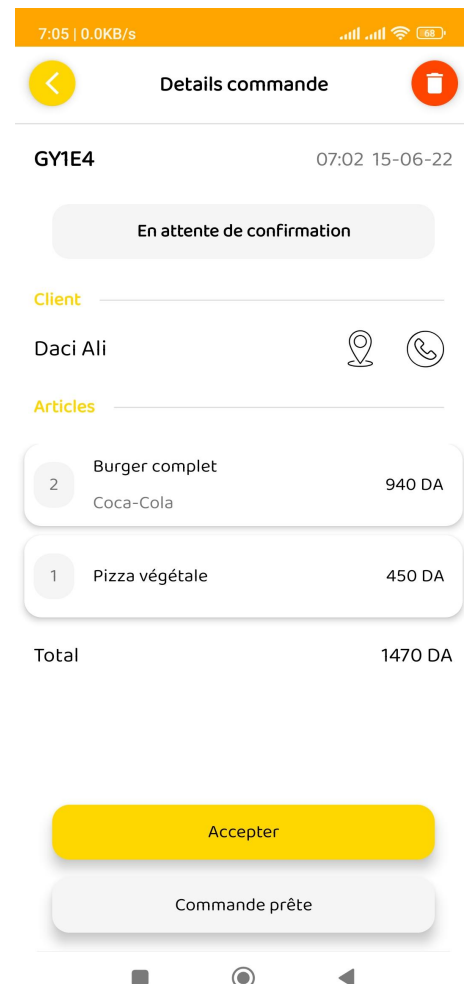


FIGURE 4.32 –
Interface Détails commande

Pour consulter les commandes prêtes il suffit juste de cliquer sur « Prêtes » dans l'interface principale comme indiqué sur la figure 4.33. Et pour les commande terminées le même chose en cliquant sur « Historique » dans l'interface principale comme indiqué sur la figure 4.34 ci-dessous



FIGURE 4.33 –
Interface Commandes prêtes

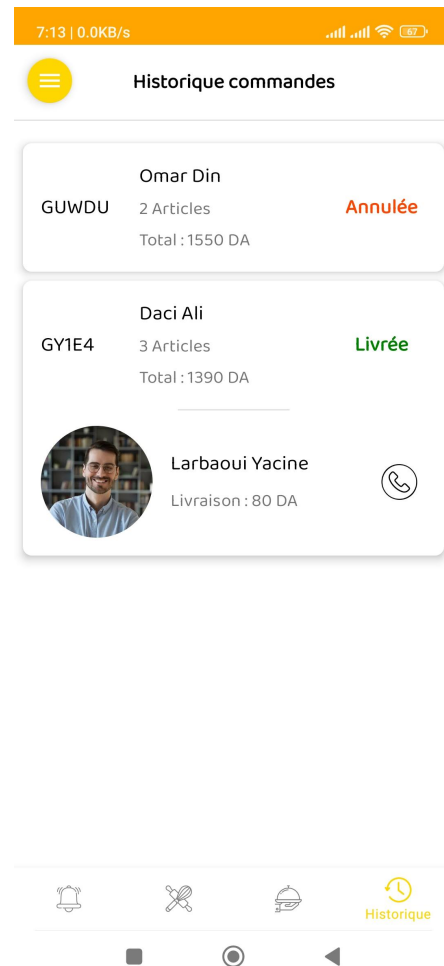


FIGURE 4.34 –
Interface Historique commandes

Lorsqu'un vendeur veut consulter ou modifier son menu il suffit de cliquer sur « Menu » sur le menu déroulant. L'interface « Menu » s'affiche contenant tout le menu du magasin comme indiqué sur la figure 4.35. Pour créer un nouvel article le vendeur clique sur le bouton « Ajouter un article », l'interface « Ajouter Article » s'affiche ensuite le vendeur remplit les champs nécessaires et clique sur « Ajouter » comme indiqué sur la figure 4.36 ci-dessous.

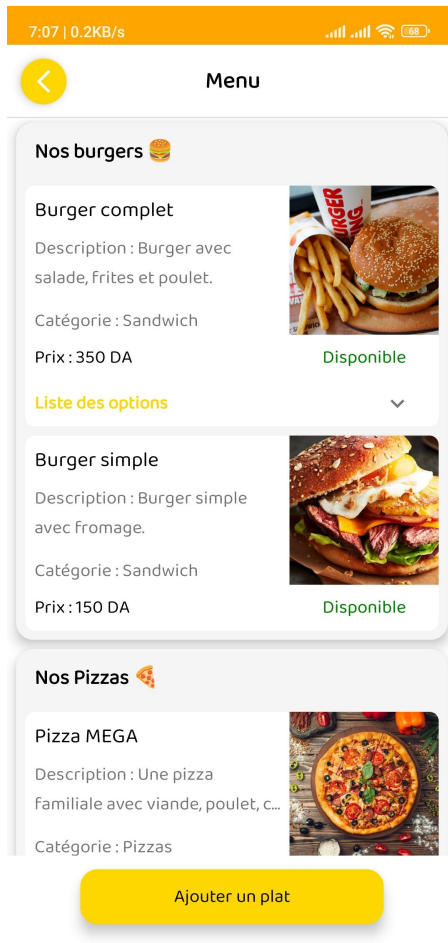


FIGURE 4.35 –
Interface Menu

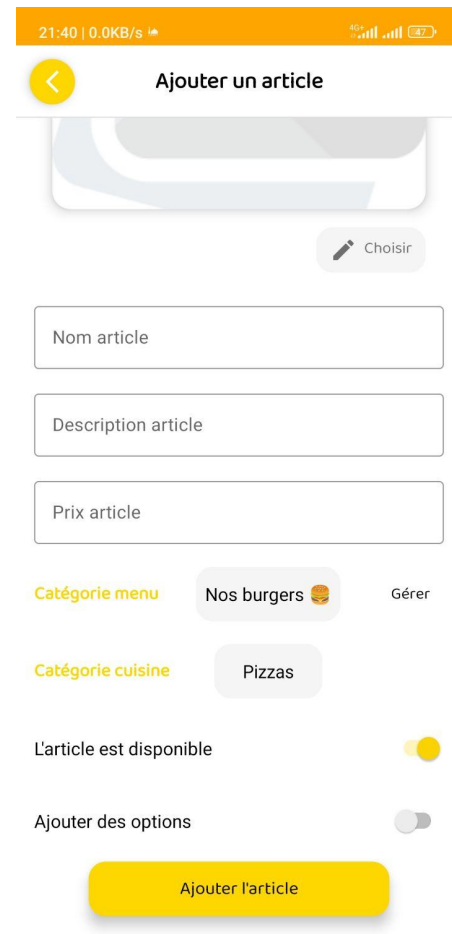


FIGURE 4.36 –
Interface Ajouter un article

Lorsqu'un vendeur veut consulter son code QR pour être scanner par un livreur il suffit de cliquer sur « Code QR » sur le menu déroulant, l'interface « Code QR » s'affiche avec un grand code QR au milieu comme indiqué sur la figure 4.37 ci-dessous.



FIGURE 4.37 – Interface Code QR

4.4.3 Application « Garanda Food Livreurs » :

Pour créer un nouveau compte, le livreur clique sur « Créer un compte » sur l'interface d'authentification, un formulaire de création de compte se lance comme indiqué sur la figure 4.38 ci-dessous, le livreur introduit ses informations et choisit une image pour son profile, ensuite il clique sur le bouton « Créer un compte » pour envoyer une demande à un administrateur

7:18 | 0.1KB/s

Choisir

Nom livreur

Prénom livreur

Adresse mail

Numéro de téléphone

Ville de livraison Bouira

Moyen de livraison Moto

Créer un compte

FIGURE 4.38 – Interface Créer compte Livreur

Lorsqu'un livreur est authentifié l'interface d'accueil s'affiche indiquant que le statut du livreur est « Hors service » comme indiqué sur la figure 4.39 ci-dessous. Pour commencer la livraison et consulter les commandes en attentes il suffit de cliquer sur le bouton « Commencer »



FIGURE 4.39 – Interface Accueil

Après le commencement de la livraison le statut du livreur devient « En service » et l'interface « Commandes en attentes » s'affiche contenant la liste des commandes en attentes dans la ville du livreur comme indiqué sur la figure 4.40 ci-dessous.

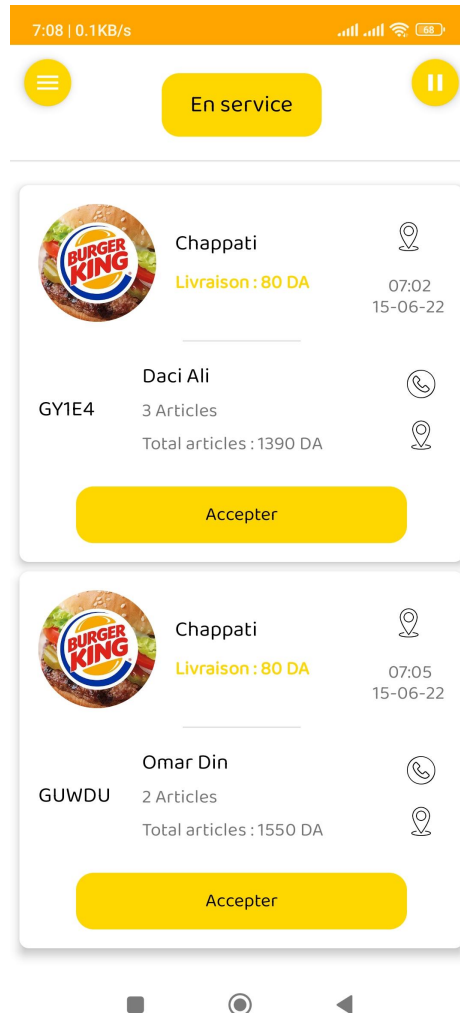


FIGURE 4.40 – Interface Commandes en attentes

Lorsqu'un livreur accepte une commande l'interface « Détails commande » s'affiche contenant le numéro unique et le statut de la commande, les coordonnées du magasin (position et contact), ainsi les coordonnées du client (nom, position, numéro de téléphone) comme indiqué sur la figure 4.41 ci-dessous. Et quand le livreur termine une commande l'interface « Commande terminée » s'affiche comme indiqué sur la figure 4.42 ci-dessous.



FIGURE 4.41 –
Interface Détails commande

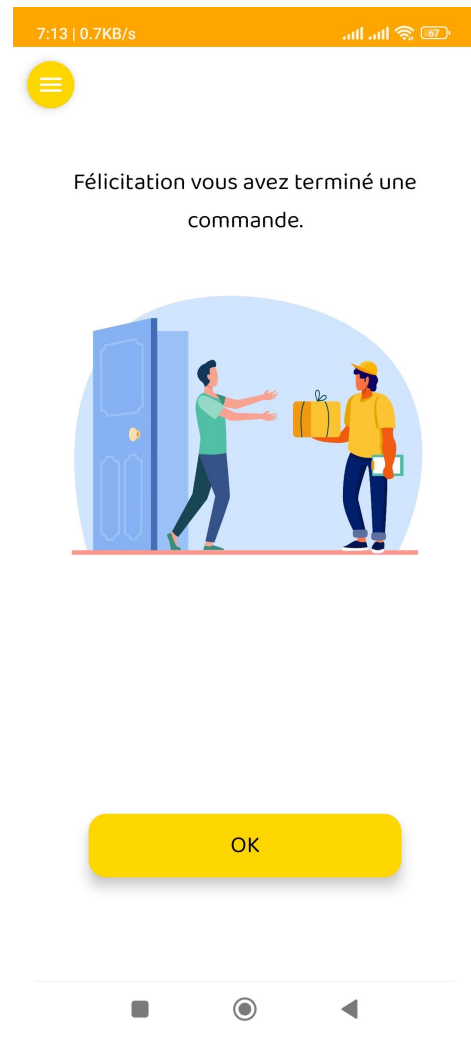


FIGURE 4.42 –
Interface Commande terminée

L'interface principale contient aussi un menu déroulant qui offres d'autres fonctionnalités sur l'application comme indiqué sur la figure 4.43 ci-dessous.

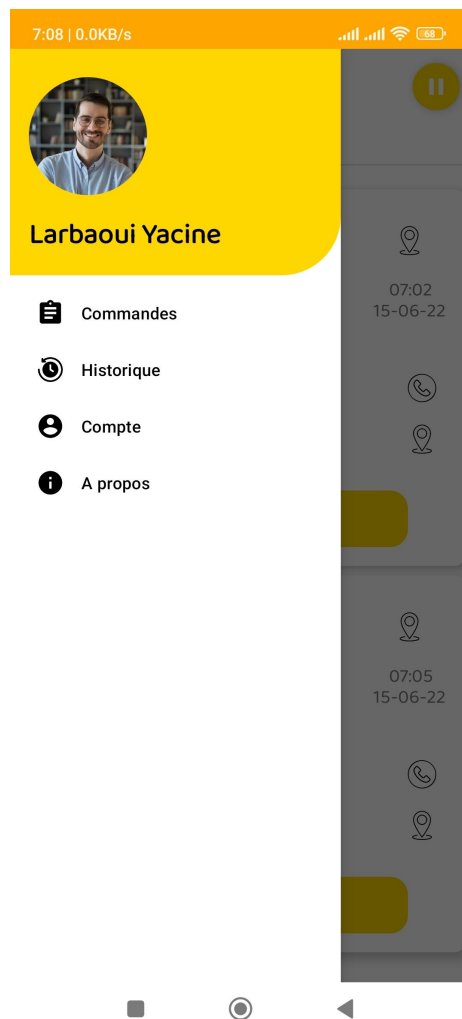


FIGURE 4.43 – Menu déroulant

4.5 Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et logiciel du développement, ainsi l'ensemble des technologies utilisées. Et enfin, nous avons illustré quelques interfaces graphiques de nos applications.

Conclusion générale et perspectives

L'objectif de notre projet de fin d'étude était la conception et la réalisation d'un ensemble d'applications mobile pour la livraison de nourriture et de plats cuisinés à domicile basé sur le Cloud, dans le but de permettre aux citoyens de notre région de passer des commandes de nourriture sans se déplacer et à n'importe quel moment auprès plusieurs restaurants et fast-foods partenaire dans la même région.

Nous avons commencé par une étude de l'existant en regroupant des informations sur les restaurants, la livraison et les applications qui existe déjà dans le domaine, ensuite nous avons étudié les problématiques liés à l'absence d'un service de livraison dans la région et proposés nos solutions. Pour la modélisation, nous avons utilisé UML comme technique, ce qui nous a permis de mener correctement la partie d'analyse des besoins grâce aux diagrammes des cas d'utilisations et aussi la partie conception.

L'application mobile est développée en KOTLIN sous Android Studio en utilisant le Cloud Firestore comme base de données NoSQL. Ces choix nous ont permis de réaliser un ensemble d'applications complètes répondant aux attentes prévues. Ce projet de fin d'étude a fait l'objet d'une expérience intéressante, qui nous a permis de découvrir le monde professionnel et de mettre en pratique ce que nous avons acquis durant notre cursus, et nous a permis aussi de perfectionner nos connaissances acquises en programmation et en conception et de renforcer nos connaissances concernant le développement mobile. Comme perspective nous envisageons une version iOS pour iPhone, le multilinguisme, suivi de la commande sur le Maps, le e-paiement (EDAHABIA ou CIB).

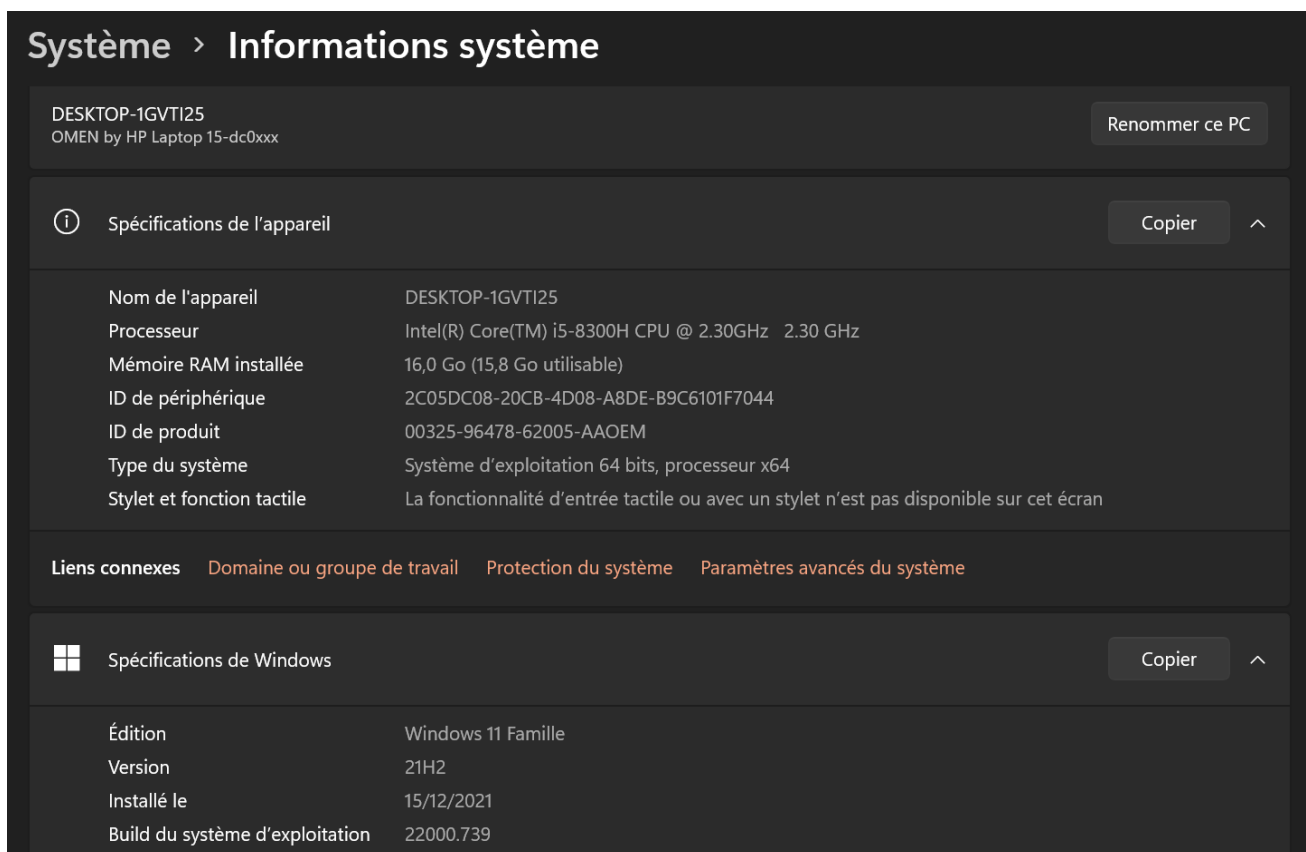
Bibliographie

- [1] Définition de restaurant. Consulté le 25/02/2022, à partir de <https://en.wikipedia.org/wiki/Restaurant>
- [2] Frédéric, toussaint. les différentes formules de restauration. Consulté le 25/02/2022, à partir de <http://technorestor.org/tr/concepts/>
- [3] Histoire de la livraison. Consulté le 27/02/2022, à partir de <https://livrolution.wordpress.com/2018/05/10/histoire-de-la-livraison/>
- [4] Site Web officiel d'UberEats. Consulté le 14/03/2022, à partir de <https://www.ubereats.com/>
- [5] Site Web officiel Deliveroo. Consulté le 14/03/2022, à partir de <https://deliveroo.ae/>
- [6] Site Web officiel Just Eat. Consulté le 14/03/2022, à partir de <https://www.just-eat.co.uk/>
- [7] Dictionnaire Larousse. Édition 2008, p.370.
- [8] Expression et analyse des besoins en UML avec PowerAMC (Manuel d'utilisation), Version 1.1, Édition 2005, p.32.
- [9] The Unified Modeling Language User Guide. Par J. Rumbaugh, I. Jacobson, G. Booch Published. Édition Octobre 1998, p.91.
- [10] UML2, pour l'analyse d'un système d'information. Par Chantal morley, Jean hugues, Bernard le blanc. Édition 2008, p.185.
- [11] Mobile OS. Consulté le 23/04/2022, à partir de <https://www.problogbooster.com/2016/02/Top-Mobile-Operating-Systems-Mobile-OS-Overview-Guide.html>
- [12] Définition Android. Consulté le 23/04/2022, à partir de <https://fr.wikipedia.org/wiki/Android>
- [13] Programmation Android de la conception au déploiement. Par Robles, Damien Guignard-Julien Chable-Emmanuel. Eyrolles, édition 2010.

- [14] Définition des applications. Consulté le 25/04/2022, à partir de <https://www.techopedia.com/definition/4224/application-software>
- [15] Qu'est-ce qu'une application web. Consulté le 25/04/2022, à partir de <https://www.synbioz.com/blog/definition-application-web>
- [16] Android Studio : une version 1.0 pour l'IDE de Google. Consulté le 04/05/2022, à partir de <http://www.zdnet.fr/actualites/>
- [17] Kotlin Documentation. Consulté le 04/05/2022, à partir de [https://fr.wikipedia.org/wiki/Kotlin_\(langage\)](https://fr.wikipedia.org/wiki/Kotlin_(langage))
- [18] Firebase. Consulté le 06/05/2022, à partir de <https://fr.wikipedia.org/wiki/Firebase>
- [19] Cleopas Banda article sur Medium. Consulté le 10/05/2022, à partir de <https://levelup.gitconnected.com/simple-guide-to-android-product-flavors-674106455038>
- [20] Why to use MVVM instead of MVC in Android. Consulté le 10/05/2022, à partir de <https://stackoverflow.com/questions/56770241/why-to-use-mvvm-instead-of-mvc-in-android/Mariam%20Aljamea,%20and%20Mohammad%20Alkandari/>. Par Dheeraj Singh Bhadoria.
- [21] Site officiel Material Design. Consulté le 10/05/2022, à partir de <https://material.io/design/introduction>
- [22] Les API de Google Développer un outil avec la Google API (SOAP). webrankinfo. Consulté le 13/05/2022, à partir de <http://www.webrankinfo.com/dossiers/apigoogle/api-soap#gref>
- [23] Utilisation de Hilt dans votre application Android. Consulté le 18/05/2022, à partir de <https://developer.android.com/codelabs/android-hilt>
- [24] Site officiel Flaticon. Consulté le 20/05/2022, à partir de <https://www.flaticon.com/fr/auteurs>
- [25] Projet sur GitHub. Consulté le 20/05/2022, à partir de <https://github.com/zxing/zxing>
- [26] Définition Code QR. Consulté le 20/05/2022, à partir de https://fr.wikipedia.org/wiki/Code_QR
- [27] Qu'est-ce que le processus unifié. Consulté le 19/03/2022, à partir de <https://www.semanticscholar.org/paper/The-unified-modeling-language-reference-manual-Rumbaugh-Jacobson/18f910f755ac1ced2d8a644a50c7fdb20083>

Annexe

Configuration de la machine utilisée durant la phase de développement



The screenshot shows the Windows 'System' > 'System Information' page. It is divided into two main sections: 'Specifications of the device' and 'Specifications of Windows'. The device section lists hardware details like the name (DESKTOP-1GVTI25), processor (Intel Core i5-8300H), RAM (16.0 Go), and system type (64-bit). The Windows section lists the edition (Windows 11 Famille), version (21H2), and installation date (15/12/2021).

Système > Informations système

DESKTOP-1GVTI25
OMEN by HP Laptop 15-dc0xxx Renommer ce PC

Spécifications de l'appareil Copier ^

Nom de l'appareil	DESKTOP-1GVTI25
Processeur	Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz 2.30 GHz
Mémoire RAM installée	16,0 Go (15,8 Go utilisable)
ID de périphérique	2C05DC08-20CB-4D08-A8DE-B9C6101F7044
ID de produit	00325-96478-62005-AAOEM
Type du système	Système d'exploitation 64 bits, processeur x64
Stylet et fonction tactile	La fonctionnalité d'entrée tactile ou avec un stylet n'est pas disponible sur cet écran

[Liens connexes](#) [Domaine ou groupe de travail](#) [Protection du système](#) [Paramètres avancés du système](#)

Spécifications de Windows Copier ^

Édition	Windows 11 Famille
Version	21H2
Installé le	15/12/2021
Build du système d'exploitation	22000.739

FIGURE 4.44 – Configuration de la machine

Configuration du terminal mobile utilisé lors de la phase de développement

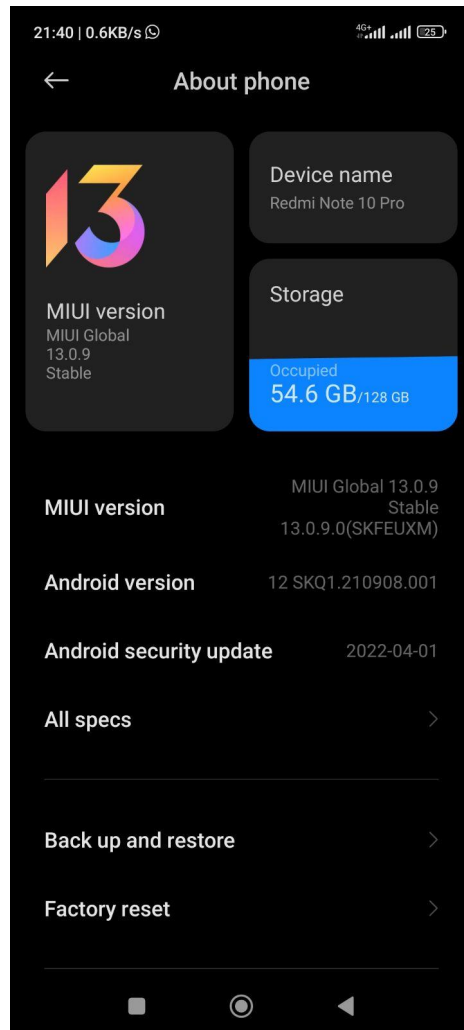


FIGURE 4.45 – Configuration du terminal mobile

Les trois images suivantes représentent le logo de chaque application :



FIGURE 4.46 – Logo Garanda Food



FIGURE 4.47 – Logo Garanda Food Vendeurs

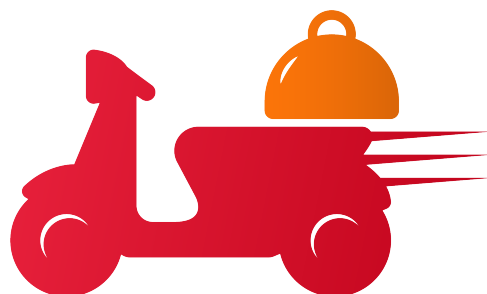


FIGURE 4.48 – Logo Garanda Food Livreurs

ملخص

أدى تطور أدوات الاتصال وكذلك سرعات الاتصال إلى العديد من التغييرات في حياة الناس اليومية ، ويميل الناس إلى استخدام هواتفهم الذكية المتصلة كثيرًا. الهدف من هذا المشروع هو إنشاء خدمة توصيل وجبات جاهزة باستخدام مجموعة من تطبيقات الهاتف الذكي. يتم تنفيذ هذه الخدمة باستخدام أندرويد ستوديو ، كوتلين كلغة برمجة و فاير بايز كقاعدة بيانات. **الكلمات المفتاحية** : خدمة توصيل الطعام ، توصيل ، البيع عبر الإنترنت ، الشراء عبر الإنترنت ، تطبيق.

Résumé

Le développement des outils de communication ainsi que les vitesses de communication entraînent de nombreux changements dans la vie quotidienne des gens, les gens ont tendance à beaucoup utiliser leurs smartphones connectés. Le but de ce projet est de réaliser un service de livraison de plats cuisinés sous un ensemble d'applications mobile. L'implémentation de ce service se fait en utilisant la plateforme Android Studio, Kotlin comme langage de programmation et FireBase comme base de données.

Mots clés : service de livraison de plats cuisinés, livraison, vente en ligne, achat en ligne, application.

Abstract

The development of communication tools as well as communication speeds lead to many changes in people's daily lives, people tend to use their connected smartphones a lot. The goal of this project is to create a ready meal delivery service using a set of mobile applications. The implementation of this service is done using the Android Studio platform, Kotlin as the programming language and Firebase as the database.

Key words : food delivery service, delivery, online sale, online purchase, application.