

République Algérienne et Démocratique Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira De Bejaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de fin cycle

En vue d'obtention du diplôme de Master Professionnel en informatique

Option : Administration Et Sécurité Des Réseaux

Thème

Conception et Réalisation d'une application mobile de Gestion des livraisons

Cas d'étude : restaurant « VIDALOCA »

Réalisé par :

- M^{lle} BOUABIDA Zohra
- M^{lle} BOUCHEBBAH Dina

Encadrer par :

- M^r OUZEGGANE Redouane

Membre de jury :

- Président : Mme GHEREBI
- Examineur : Mme MAMMERI

Promotion 2019/2020

Remerciements

Tout d'abord nous tenons à remercier le bon Dieu de nous avoir donné la force, le courage et la volonté pour accomplir ce travail.

Nous remercions nos parents ainsi nos frères et sœurs pour leurs soutiens qu'ils nous ont accordés tout au long de notre chemin.

Nous remercions nos chers ami(e)s qui sont toujours présents et fidèles.

Nous remercions notre encadreur **M. OUZEGGANE REDOUANE** pour avoir dirigé notre travail, pour tout le temps qu'il nous a consacré, pour ses précieux conseils et pour son aide et son appui tout au long de ce dernier.

Nous remercions les membres du jury qui ont accepté de juger et examiner notre modeste travail.

Dédicaces

Nous dédions notre travail à nos très chers parents pour leurs patiences, leurs amours leurs soutiens et leurs encouragements tout le long de notre cursus d'études.

À nos frères et sœurs qui n'ont cessé d'être pour nous des exemples de persévérance, pour leurs encouragements et leurs aides.

À tous les membres de nos familles qui nous ont soutenus.

À nos amis(e) et nos camarades qui ont contribué de près ou de loin.

Table Des Matières

Table des Matière.....	i
Liste des Figures.....	v
Liste des Tableaux.....	vi
Liste des Abréviations.....	vii
Introduction Générale.....	1
Chapitre I Organisme d'accueil et recueil des besoins	3
I.1. Introduction.....	3
I.2. Présentation d'organisme d'accueil.....	3
I.2.1. Définition d'un restaurant	3
I.2.2. Les différentes formules de restauration	3
a- Restauration traditionnelle	3
b- Restauration de collectivité.....	4
c- Restauration rapide	4
I.2.3. Présentation du fast-food VIDA LOCA.....	4
I.2.4. Organigramme du restaurant	5
I.2.5. Le personnel et ses activités	5
I.3. Situation informatique et le mode de travail.....	6
I.4. Problématique.....	6
I.5. Solution proposée.....	7
I.6. Recueil des besoins.....	7
I.6.1 Besoins fonctionnels.....	7
I.6.2. Besoins non fonctionnels.....	7
I.7. Conclusion.....	8
Chapitre II Analyse des besoins	9
II.1. Introduction.....	9
II.2. Présentation des acteurs du futur système.....	9
II.3. Diagramme de contexte dynamique	9

II.3.1. Identification des messages échangés.....	10
II.3.2. Diagramme de contexte dynamique	11
II.4. Capture des besoins fonctionnels.....	11
II.4.1. Identification des cas d'utilisation.....	11
II.4.2. Description textuelle des cas d'utilisations.....	12
II.4.3. Diagrammes de cas d'utilisation.....	16
II.5. Diagramme de séquence système.....	18
II.6. Conclusion.....	20
Chapitre III Conception	22
III.1. Introduction.....	22
III.2. Diagramme de séquence d'interaction.....	22
III.3. Diagramme de classe du domaine.....	24
III.3.1. Dictionnaire des données.....	25
III.4. Nosql.....	27
III.4.1. Définition	27
III.4.2. Avantages	28
III.4.3. Inconvénients	28
III.4.3. La différence entre les bases de données Sql et les bases de données Nosql.	28
III.5. Conclusion.....	29
Chapitre IV Réalisation	30
IV.1. Introduction.....	30
IV.2. Environnement de programmation et bibliothèques.....	30
IV.2.1. Flutter	30
a- Pourquoi flutter	30
b- Principe de flutter.....	30
c- Dart	31
IV.2.2. SDK.....	31
IV.2.3. JSON	32
IV.3. Implémentation de la base de données.....	32
IV.3.1. Firebase	32

a- Firebase authentication.....	32
b- RealTime Data Base.....	33
c- Firebase Storage.....	33
d- Firebase Cloud Messaging.....	33
e- Cloud Firestore.....	33
IV.3.2. Comment ajouter Firebase a notre projet ?.....	33
IV.4. Architecture de l'application	35
IV.4.1. Coté Utilisateur	35
IV.4.2. Coté administrateur.....	36
IV.5. Phase de déploiement.....	37
IV.5.1. Architecture client/serveur	37
a- Client.....	37
b- Serveur.....	37
IV.5.2. Mise en ligne.....	38
IV.5.3. Hébergement d'une application.....	38
IV.6. Interfaces homme-machines (IHM).....	39
IV.6.1. Interfaces coté clients.....	39
a- Interface d'accueil et Connexion/inscription.....	39
b- Interfaces Menu	39
c- Interfaces Panier	40
d- Interfaces Lancement de la commande.....	41
IV.6.2. Interfaces coté administrateurs.....	42
a- Interface Accueil	42
b- Interface Saisir menu	43
c- Interface Gestion des produits.....	44
d- Interface Gestion Commande.....	45
IV.7. Scénarios de tests.....	46
IV.7.1. Validation	46
IV.7.2. Guide d'utilisation.....	46
IV.8. Conclusion.....	47

Conclusion générale.....	48
Liste des références.....	49
Annexe A.....	I
Annexe B.....	III

Liste des Figures

Figure I.1 -Organigramme de l'organisme d'accueil.....	5
Figure II.1 –Diagramme de contexte dynamique.....	11
Figure II.2 –Diagramme de cas d'utilisation « administrateur ».....	17
Figure II.3 –Diagramme de cas d'utilisation « utilisateur ».....	18
Figure II.4 –Diagramme de séquence système « Se connecter ».....	19
Figure II.5 –Diagramme de séquence système « Saisir menu et prix ».....	20
Figure III.1 –Diagramme de séquence d'interaction « se connecter ».....	23
Figure III.2 –Diagramme de séquence d'interaction « Ajouter au menu »	24
Figure III.3 – Diagramme de classe de domaine.....	25
Figure IV.1 – Première étape de l'implémentation-Enregistrement de l'application	34
Figure IV.2 – Téléchargement du fichier de configuration.	34
Figure IV.3 – Ajout du SDK Firebase.	35
Figure IV.4 – la console de Firebase.	35
Figure IV.5 – Architecture de l'application « coté utilisateur ».....	36
Figure IV.6 – Architecture de l'application « Administrateur ».....	37
Figure IV.7 – Architecture client /serveur.....	38
Figure IV.8 – Interface d'accueil et interface de connexion/inscription.....	39
Figure IV.9 – Interfaces Menu globale et détaillé.....	40
Figure IV.10 – Interface panier.....	41
Figure IV.11 – Interface lancement commande.....	42
Figure IV.12 – Interface d'accueil	43
Figure IV.13 – Interface ajout des produits.....	44
Figure IV.14 – Interface gestion produits.	45
Figure IV.15 – Interface gestion commandes.	46

Liste Des Tableaux

Tableau I.1 -Liste du personnel du restaurant.	6
Tableau II.1 –Message transmis « acteur /système »« système/acteur ».	11
Tableau II.2 –Identification des cas d'utilisation.....	12
Tableau II.3 –cas d'utilisation « se connecter »	13
Tableau II.4 - cas d'utilisation « consulter menu globale»	13
Tableau II.5 - cas d'utilisation « consulter panier ».....	14
Tableau II.6 - cas d'utilisation « saisir menu et prix».....	16
Tableau III.1 -Dictionnaire des données	26
Tableau III.2 -La différence entre le SQL et le NOSQL	29

Liste Des Abréviations

ACID	Atomicité, cohérence, isolation, durabilité
FCM	Firestore Cloud Messaging
IDE	Integrated development environment
IOS	I phone operating system
IHM	Interface homme machine
JSON	JavaScript Object notation
NoSQL	Not only Sql
SDK	Software Développement Kit
SGBDR	Système de Gestion de Base de Données
Sql	Structured query language
UML	Unified modeling language
VM	Vertual machine
XML	Extensible Markup Language

Introduction générale

Le monde informatique est très vaste, son apparition correspond aux changements de culture. Jour après jour l'informatique occupe une place intéressante dans les sociétés et les organisations (grandes entreprises, petites entreprises, restaurants, boutiques) à tous les niveaux en effet l'informatique permet de traiter, mémoriser, et de diffuser l'information dans des délais plus courts et avec des méthodes plus efficaces.

Donc Le système informatique actuel est capable de gérer et de résoudre tous les problèmes rencontrés dans les entreprises et les organisations, nous nous sommes intéressés au secteur de la restauration.

Notre travail est de concevoir et de réaliser une application mobile de gestion de livraison d'un restaurant. Pour ce faire, nous avons fait visite au restaurant en question pour voir le déroulement des différentes tâches et nous permettre de mettre en place une application qui soit bénéfique dans leur gestion, pour éviter les différents problèmes liés à la gestion traditionnelle de livraison.

Pour cela, nous avons effectué un stage au niveau du restaurant « VIDA LOCA », ce qui nous a permis de bien comprendre le besoin du client et de déceler la problématique importante que confronte le gestionnaire du restaurant, à savoir : étendre l'activité du restaurant sur l'internet (à distance). Ainsi, nous avons proposé de concevoir et de réaliser une application mobile destinée au gestionnaire et les clients du restaurant.

Pour bien présenter ce travail, notre mémoire est structuré en quatre chapitres.

Dans le premier chapitre intitulé organisme d'accueil et recueil des besoins, nous parlons en bref de différentes formules d'un restaurant. Puis, nous présentons notre cas d'étude pour lequel nous effectuons notre travail. Par la suite, nous exposons la problématique et nous proposons une solution pour cette dernière.

Dans le deuxième chapitre intitulé Analyse des besoins, nous allons identifier les acteurs principaux qui joueront des rôles dans notre application et leurs fonctionnalités à travers le diagramme de cas d'utilisation, suivi par des diagrammes de séquences qui illustrent des descriptions détaillées pour les scénarios des importants cas d'utilisations.

En ce qui concerne le troisième chapitre, la conception du système sera détaillée et élaborée en utilisant les diagrammes d'interaction ainsi que le diagramme de classe.

Dans le dernier chapitre, nous présentons les différents outils techniques et les langages de programmation utilisés pour le développement de notre application. Ainsi,

nous présentons les différentes interfaces de notre application pour la gestion des livraisons du restaurant VIDA LOCA.

Enfin, nous avons finalisé ce mémoire par une conclusion générale qui récapitule notre travail, suivie de quelques perspectives qui proposent la continuité de notre projet.

CHAPITRE I

Organisme d'Accueil et Recueil des Besoins

I.1. Introduction

Dans le présent chapitre, nous entamons la présentation de notre cas d'étude. Tout d'abord, nous présentons le cadre de notre travail avec la situation informatique et le mode de travail de l'organisme, nous définissons par la suite la problématique et nous proposons une solution pour cette dernière.

Et enfin, nous présenterons les besoins fonctionnels et non fonctionnels attendus de l'application afin d'atteindre l'objectif de notre projet de fin de cycle.

I.2. Présentation d'organisme d'accueil

I.2.1. Définition d'un restaurant

Un restaurant est un établissement commercial où l'on sert des repas contre paiement. Dans un restaurant, les repas sont préparés par une équipe de cuisiniers. Il existe plusieurs types de cuisines. Les restaurants sont parfois le dispositif réservé au service des repas au sein d'une plus grande entité (hôtel, université, aéroport, etc.), on parle alors de restaurant collectif par opposition au site de cuisine. Ils peuvent aussi être associés à une activité de traiteur. 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. Un restaurant est un commerce ou un magasin [2].

I.2.2. Les différentes formules de restauration

a. 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 française.

- Les restaurants classiques
- Les restaurants d'hôtels

- La restauration à thèmes
- La restauration autour d'un produit
- La restauration autour d'un pays

b. Restauration de collectivité

La restauration de collectivité repose sur des contraintes précises : nourrir un grand nombre de gens pour un prix extrêmement bas. Comme la réussite de cette activité repose en grande partie sur le principe des économies d'échelle,

- La restauration d'entreprise
- La restauration scolaire
- La restauration Hospitalière.

c. Restauration rapide

La restauration rapide est le secteur qui a connu la plus forte croissance ces vingt dernières années. En effet, c'est ce concept qui répond en partie à nos besoins de citoyens modernes.

- Le fast-food
- Le restauroute
- La cafétéria [2].

Dans ce projet de fin de cycle notre cas d'étude repose sur cette dernière catégorie de restauration qui est la restauration rapide.

I.2.3. Présentation du fast-food VIDA LOCA

VIDA LOCA est un restaurant de type Restauration Rapide, plus exactement il représente un fast-food, situé à la cité Zirara nouvelle (cité Dallas) Bejaïa, il suggère un menu riche salé et sucré, nous trouvons des entrées, des plats, des sandwiches, des boissons chaudes et froides, etc.

VIDA LOCA est un lieu chaleureux et propre avec des employés respectueux et une bouffe délicieuse c'est pourquoi il reçoit beaucoup de monde de toute catégorie et le nombre de clients ne cesse d'augmenter.

Au plus du service sur place et à emporter, VIDA LOCA propose aussi le service de livraison à domicile.

I.2.4. Organigramme du restaurant

Il existe différents organigrammes selon la taille de l'établissement : petit restaurant, grand restaurant ou restaurant gastronomique. Dans notre projet nous nous intéressons aux petits restaurants représentés comme suit:

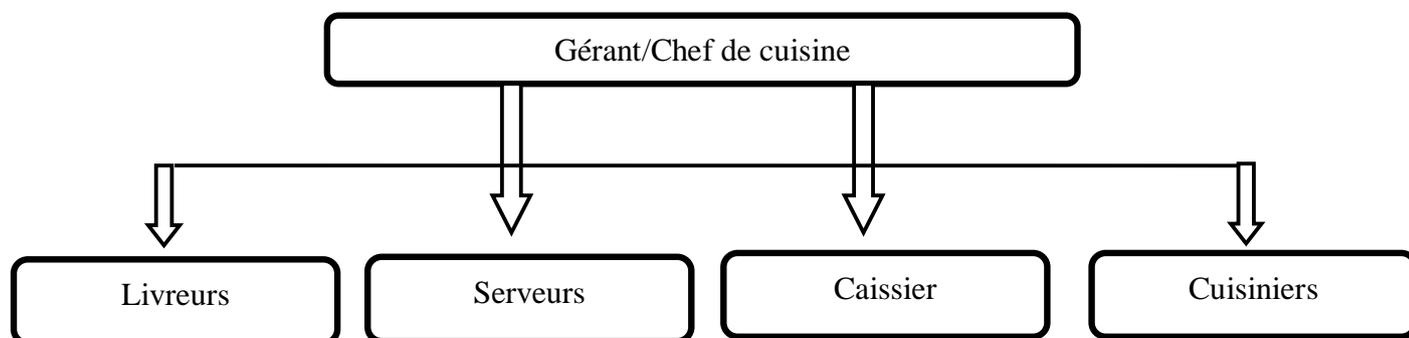


Figure I.1 - Organigramme de l'organisme d'accueil.

I.2.5. Le personnel et ses activités

Le tableau suivant (Tableau I.1) montre les différents types de personnel du restaurant ainsi que leurs rôles respectifs.

Personnels	Rôles
Gérant/chef de cuisine	<ul style="list-style-type: none"> -Il organise le travail du personnel ; -Responsable des ventes ; -Il veille à la qualité du service ; -Il gère le stock ; -Elaboration des plats inscrits sur le menu en tant que gérant et chef de cuisine ; -Il prépare les plats les plus complexes ; -Il donne les instructions aux cuisiniers ;
Cuisiniers	<ul style="list-style-type: none"> -Ils préparent les plats
Serveurs	<ul style="list-style-type: none"> - Ils dressent les tables ; - Ils prennent les commandes des clients ; - Ils servent les clients et encaissent leurs règlements ;

	<ul style="list-style-type: none"> - Ils nettoient les tables et les salles ; - Ils débarrassent les tables.
Caissier	<ul style="list-style-type: none"> - Il se charge de la caisse ; - Il prépare les factures des clients ; - Il gère les cartes de fidélité ;
Livreur	<ul style="list-style-type: none"> -Il répond aux appels téléphoniques des clients ; -Il prend les commandes des clients ; -Il livre les commandes aux clients ;

Tableau I.1 -Liste du personnel du restaurant.

I.3. Situation informatique et le mode de travail

Le service de livraison chez VIDA LOCA se fait par appel téléphonique entre le client et le livreur, la prise de commande se fait manuellement par le livreur qui à son tour reçoit le paiement à la livraison.

Le mode de travail chez VIDA LOCA se fait de la manière suivante :

- Les clients passent leurs commandes par appel téléphonique au restaurant ;
- La prise des commandes sur papier ;
- La commande est transmise en cuisine en vue de la confection des produits commandés ;
- Puis transmises en caisse pour procéder à l'addition de la facture faite par le livreur avant de livrer la commande au client et enfin procéder à l'encaissement ;

I.4. Problématique

- Prise de commandes manuelle ;
- Difficulté de gestion des livraisons ;
- Manque de traçabilité ;
- Surcharge des appels téléphoniques lors de prise des commandes ;
- Risque de se confondre entre les commandes des différents clients ;
- Menu et prix non disponible pour le client ;

Lors de la présentation du restaurant nous avons parlé sur les services proposés par ce dernier et parmi eux il existe le service de livraison.

Le service de livraison chez VIDA LOCA se fait par appel téléphonique entre le livreur et le client et la prise de commande manuelle par le livreur.

Le propriétaire souhaite avoir une application mobile pour s'approcher davantage avec leurs clients afin de faciliter la gestion des livraisons à domicile. L'application doit être facile à utiliser et accessible par tout le monde.

I.5. Solution proposée

Afin d'y remédier au problème de prise de commandes manuelle et des appels téléphoniques, nous proposons la conception et réalisation d'une application mobile de gestion des livraisons pour cela nous avons incité à notre étude les objectifs suivants :

- Automatiser les taches traitées manuellement ;
- Fiabilité du système ;
- Gestion et organisation des commandes ;
- Faciliter les taches pour le client et pour l'administrateur (livreur) ;

I.6. Recueil des besoins

Après avoir établi la problématique et les objectifs à atteindre par l'application à concevoir, nous présenterons la liste des besoins du futur système. Cette liste a été réalisée en concertation avec les futurs utilisateurs de l'application et validée par le gérant de l'établissement d'accueil.

Les besoins de notre système se divisent en deux types, à savoir :

I.6.1 Besoins fonctionnels

Ce sont les besoins spécifiant un comportement d'entrée / sortie du système, notre application "de service de livraison " doit satisfaire les besoins fonctionnels suivants :

- Saisie du menu et les prix ;
- Gestion des commandes ;
- Visualisation du menu ;
- Inscription du client ;
- Réalisation des commandes ;

I.6.2. Besoins non fonctionnels

Ce sont les besoins qui permettraient d'améliorer les performances de l'application en termes de convivialité, d'ergonomie des interfaces. Parmi ces besoins on cite :

- Besoins d'établissement de la connexion, il faut avoir une interface d'authentification qui permet à chaque utilisateur de se connecter pour commander avec son profil.
- L'application doit être facile à utiliser. Il doit présenter un enchainement logique entre les pages et un texte compréhensible, visible et lisible.
- Moyenne des temps de bon fonctionnement.

I.7. Conclusion

Au cours de ce premier chapitre, nous avons présenté l'organisme d'accueil, à savoir le restaurant VIDA LOCA, en précisant son type, ainsi que son organigramme et ses différents services. Ce stage nous a permis d'établir la problématique et de préciser les objectifs de notre travail qui consiste à la conception et réalisation d'une application Mobile multiplateforme pour la gestion des commandes. À la fin de ce chapitre, nous avons établi le recueil des besoins non-fonctionnels et fonctionnels qui feront l'objet d'une analyse dans le chapitre suivant.

CHAPITRE II

Analyse des Besoins

II.1. Introduction

Avant d'entamer la réalisation d'un projet, il est indispensable de le modéliser. En effet, chaque conception d'un système informatique repose sur un langage, définissant des formalismes de modélisation, et une démarche précisant un enchaînement d'étapes à suivre.

Après avoir fait le bilan des différents problèmes rencontrés au sein du restaurant VIDA LOCA, nous allons maintenant passer à l'analyse des besoins.

Pour cela nous présenterons d'abord les acteurs de notre système, puis le diagramme de contexte dynamique, par la suite, nous allons identifier les cas d'utilisation afin de modéliser les différentes fonctionnalités à réaliser, et nous concluons par les diagrammes de séquence système.

II.2. Présentation des acteurs du futur système

Un acteur est une entité externe qui interagit directement avec le système. Pour notre système, nous avons identifié les deux acteurs suivants :

- Administrateur : l'administrateur est chargé de la gestion et validation des commandes ainsi que la gestion des livraisons.
- Utilisateurs : les utilisateurs sont représentés par toutes personnes ayant créé un compte dans notre système.

II.3. Diagramme de contexte dynamique

Dans ce qui suit, nous allons représenter l'interaction du futur système avec son environnement extérieur. Plus précisément, cette interaction s'effectuera entre le système, qui est considéré comme une boîte noire, et les différents acteurs identifiés précédemment, en identifiant les différents messages échangés entre chaque acteur et le système.

II.3.1. Identification des messages échangés

Un message est un moyen de communication entre acteurs. Il caractérise un événement, c'est-à-dire une information envoyée à un acteur et provoquant en réponse le déclenchement d'action associée à ce dernier [1].

Le tableau ci-dessous (Tableau II.1) regroupe les messages transmis d'un acteur au système et ceux transmis du système à l'acteur :

acteur	Message acteur-système		Message système-acteur	
Administrateur	1	se connecter	2	Interface d'authentification
	19	consulter commande	20	interface commande
	21	confirmer commande	22	message de confirmation
	23	demander espace gestion menu	24	interface de mise à jour menu
Utilisateur	1	se connecter	2	Interface d'authentification
	3	espace d'accueil	4	interface d'accueil
	5	consulter menu	6	Interface menu
	7	Consulter sous-menu	8	Interface sous-menu
	9	Consulter menu détaillé	10	Interface menu détaillé
	11	consulter panier	12	Interface panier
	13	valider commande	14	Message de confirmation
	15	Consulter paramètre	16	Interface paramètre

	17	Consulter notification	18	Interface notification
--	----	------------------------	----	------------------------

Tableau II.1 -Message transmis "acteur/système" "système/acteur".

II.3.2. Diagramme de contexte dynamique

La description des différents acteurs permet de décrire le système à développer par un premier diagramme appelé « diagramme de contexte dynamique ». Ce dernier permet de présenter l'interaction des différents acteurs avec le système.

La figure ci-dessous (Figure II.1) représente le diagramme de contexte dynamique.

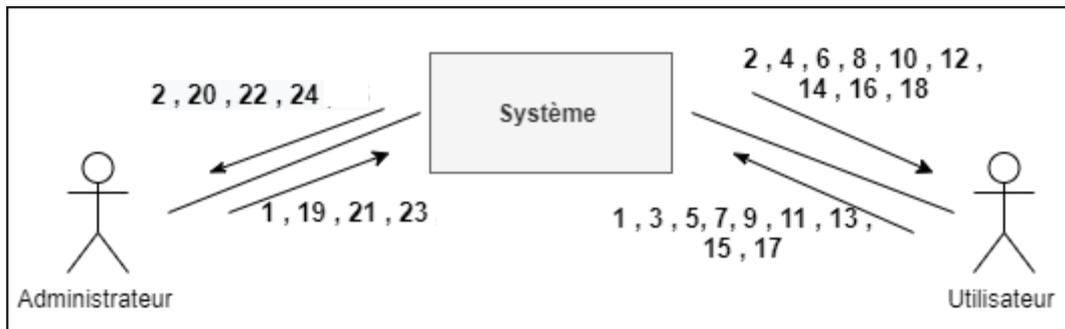


Figure II.1 –Diagramme de contexte dynamique.

II.4. Capture des besoins fonctionnels

Afin d'identifier les besoins de notre système nous allons d'abord identifier les cas d'utilisation et élaborer les diagrammes de ces derniers.

II.4.1. Identification des cas d'utilisation

Dans le système à développer, nous avons identifié les cas d'utilisation suivant :

N	Cas d'utilisation	Acteur
1	Se connecter	Utilisateurs Administrateurs
2	Consulter menu globale	Utilisateurs
3	Consulter le sous menu	Utilisateurs
4	Consulter le menu détailler	Utilisateurs
5	Consulter panier	Modifier Utilisateurs

		Supprimer	
6	Valider commande		Utilisateurs
7	Consulter notification		Utilisateurs
8	Consulter paramètre	Contacts Détaille du compte	Utilisateurs
9	Consulter commande	Valider Recaler	Administrateurs
10	Saisir menu et prix	Ajouter Modifier Supprimer	Administrateurs

Tableau II.2 -Identification des cas d'utilisation.

II.4.2. Description textuelle des cas d'utilisations

Cas d'utilisation N 1	Se connecter
Objectif	Vérification de l'identité de l'utilisateur ou l'administrateur
Acteur	Utilisateur Administrateur
Pré condition	application accessible
Scénario nominal	[Début] 1. L'utilisateur demande l'accès à la page de connexion ; 2. Le système affiche la page ; 3. L'utilisateur saisit ces informations ;

	<p>4. Le système vérifie la conformité des informations fournies ;</p> <p>[Fin]</p>
Scénario alternatif	<p>1. Les informations fournies sont incorrectes ;</p> <p>2. Le système affiche un message d'erreur ;</p> <p>3. Le système réaffiche le formulaire et attend que l'utilisateur ressaisisse ses informations ;</p>

Tableau II.3 -Cas d'utilisation "se connecter".

Cas d'utilisation N 2		Consulter menu global
Objectif	Afficher les informations du menu global	
Acteur	Utilisateur	
Pré condition	application disponible	
Scénario nominal	<p>[Début]</p> <p>1. L'utilisateur demande l'affichage de la liste du menu globale ;</p> <p>2. Le système affiche la liste demandée ;</p> <p>[Fin]</p>	
Scénario alternatif	Aucun	

Tableau II.4 -Cas d'utilisation "Consulter menu globale".

Cas d'utilisation N5		Consulter panier
Modifier	Objectif	Modifier les informations du panier afin de valider la commande
	Acteur	Utilisateur

	Pré condition	Application accessible
	Scénario nominale	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'utilisateur demande l'interface du panier ; 2. Le système affiche la page ; 3. L'utilisateur modifie les informations souhaiter ; 4. Le système confirme la modification ; <p>[Fin]</p>
	Scénario alternatif	En cas de saisit de données non compatible aux champs le système affiche un message d'erreur
Supprimer	Objectif	Supprimer des informations dans le panier
	Acteur	Utilisateur
	Pré condition	Application accessible
	Scénario nominal	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'utilisateur demande l'interface du panier ; 2. Le système affiche la page ; 3. L'utilisateur sélectionne les informations à supprimer ; 4. Le système confirme la suppression ; <p>[Fin]</p>
	Scénario alternatif	Aucun

Tableau II.5 -Cas d'utilisation "Consulter panier".

Cas d'utilisation N 9		Saisir menu et prix
Ajouter	Objectif	Ajouter au menu des informations
	Acteur	Administrateur
	Pré condition	Application accessible S'authentié
	Scénario nominal	[Début] 1. L'administrateur demande l'interface de mise à jour menu ; 2. Le système affiche la page ; 3. L'administrateur choisit l'ajout ; 4. Le système affiche le formulaire d'ajout ; 5. L'administrateur saisit les données à ajouter ; 6. Le système confirme l'ajout ; [Fin]
Scénario alternatif	En cas de champ vide le système affiche un message d'erreur ;	
Modifier	Objectif	Modifier les informations du menu ;
	Acteur	Administrateur
	Pré condition	Application accessible S'authentié
	Scénario nominal	[Début] 7. L'administrateur demande l'interface du menu ; 8. Le système affiche la page ; 9. L'utilisateur sélectionne et modifie les informations souhaiter ; 10. Le système confirme la modification ; [Fin]

	Scénario alternatif	En cas de saisit de données non compatible aux champs le système affiche un message d'erreur
Supprimer	Objectif	Supprimer des informations dans le menu ;
	Acteur	Administrateurs
	Pré condition	Application accessible S'authentifé
	Scénario nominal	[Début] 5. L'utilisateur demande l'interface du menu ; 6. Le système affiche la page ; 7. L'utilisateur sélectionne les informations à supprimer ; 8. Le système confirme la suppression ; [Fin]
	Scénario alternatif	Aucun

Tableau II.6 -Cas d'utilisation "Saisir menu et prix".

II.4.3. Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation est un formalisme permettant de modéliser le fonctionnement d'un système par un découpage en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités qui sont offertes à titre de services à des acteurs externes au système. Chaque fonctionnalité est appelée un cas d'utilisation [3].

II.4.3. Diagrammes de cas d'utilisation par acteur

La figure ci-dessous (Figure II.2) représente le diagramme de cas d'utilisation « administrateur ».

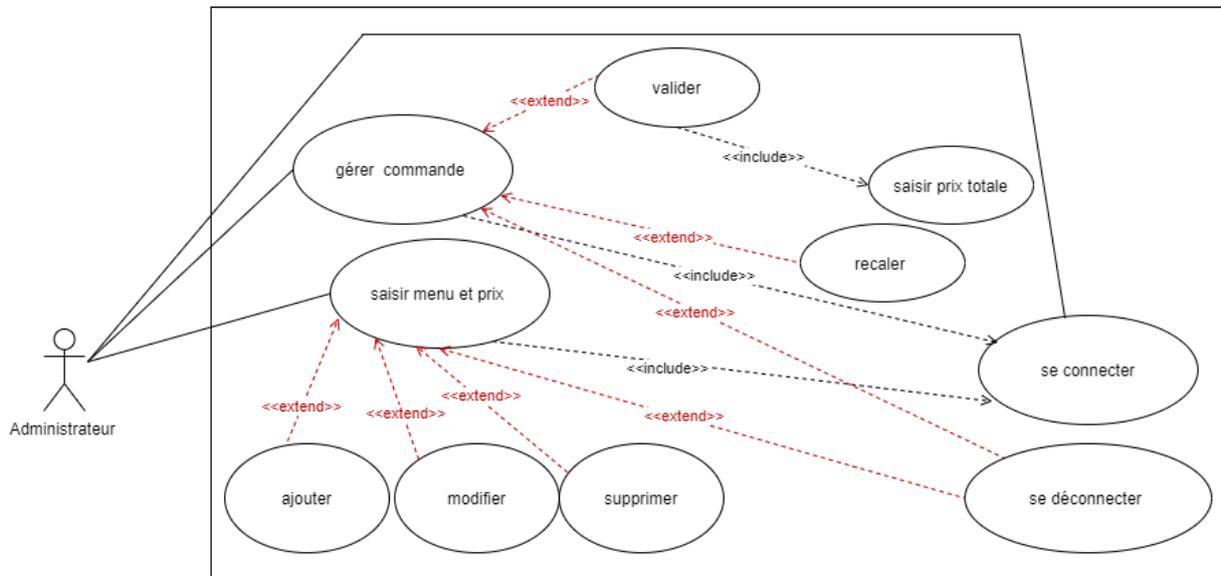


Figure II. 2 -Diagramme de cas d'utilisation "administrateur".

La figure ci-dessous (Figure II.3) représente le diagramme de cas d'utilisation « utilisateur ».

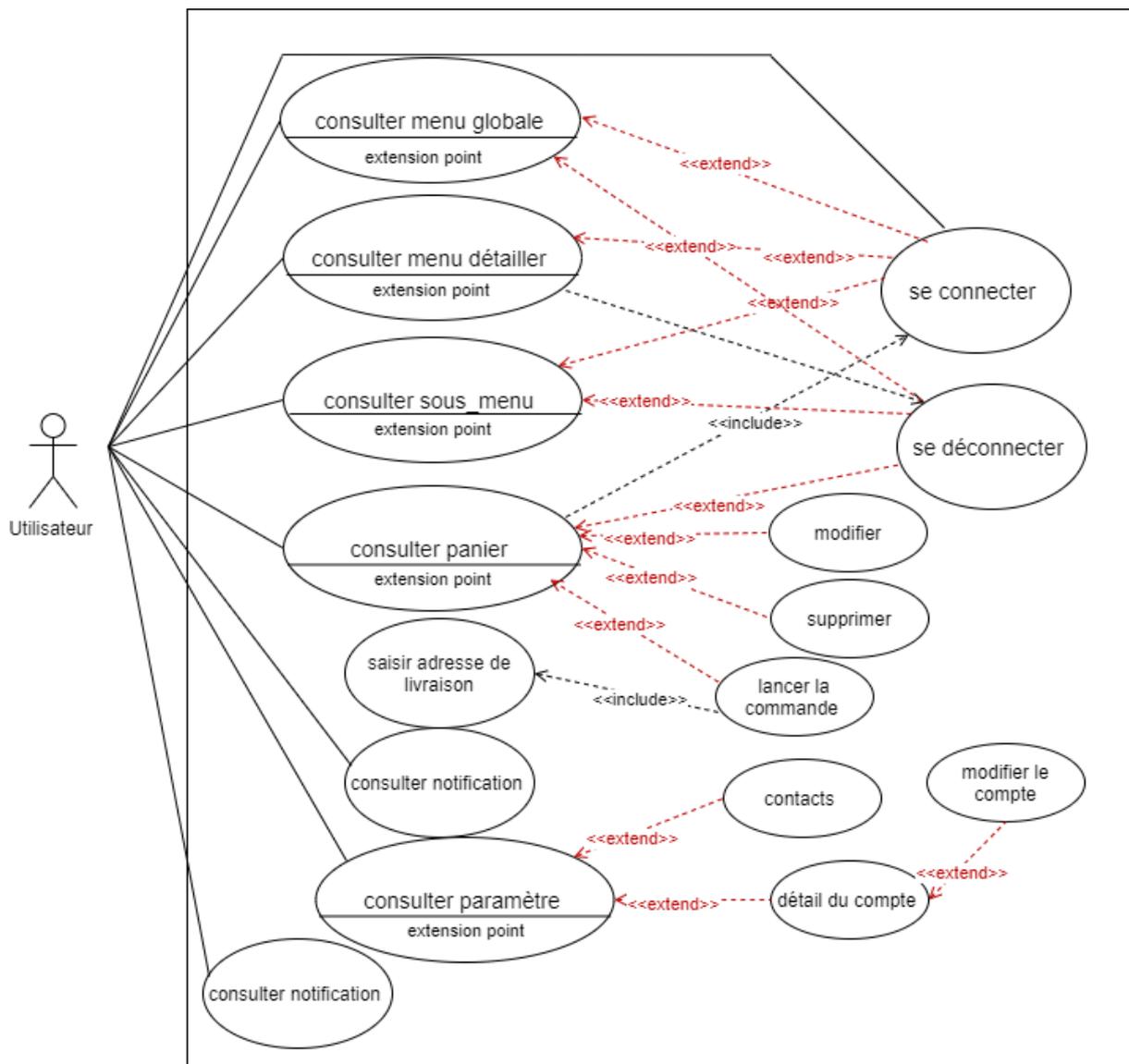


Figure II.3 -Diagramme de cas d'utilisation "utilisateur".

II.5. Diagramme de séquence système

L'objectif du diagramme de séquence est de présenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés [4].

Dans cette phase d'analyse des besoins, on considère le système comme étant une boîte noire, et nous nous intéressons sur les messages, ainsi que leur ordre chronologique, entre un ou plusieurs acteurs et le système afin de réaliser un cas

d'utilisation. Ainsi, le diagramme de séquence peut être vu comme une description d'un cas d'utilisation.

En guise de clarté, et pour éviter d'encombrer notre mémoire, nous allons illustrer deux diagrammes de séquence pour les cas d'utilisation « Se connecter » et « Saisir menu et prix ». Pour les autres cas d'utilisation, nous vous invitons à consulter l'Annexe A dédié aux diagrammes de séquence correspondant.

La figure suivante (Figure III.4) montre l'interaction entre l'acteur Utilisateur et le système afin de réaliser l'opération « Se connecter ».

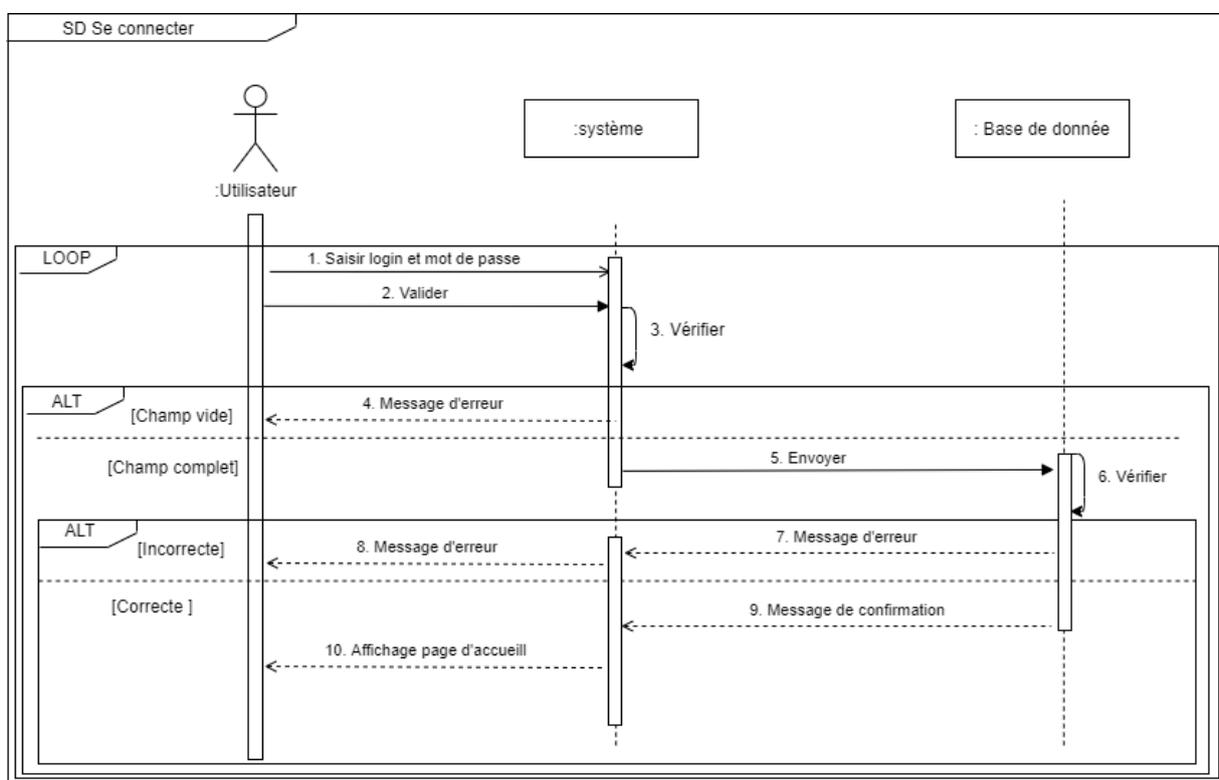


Figure II.4 -Diagramme de séquence système "Se connecter".

La figure suivante (Figure III.5) montre l'interaction entre l'acteur Administrateur et le système afin de réaliser l'opération « Saisir Menu et Prix ».

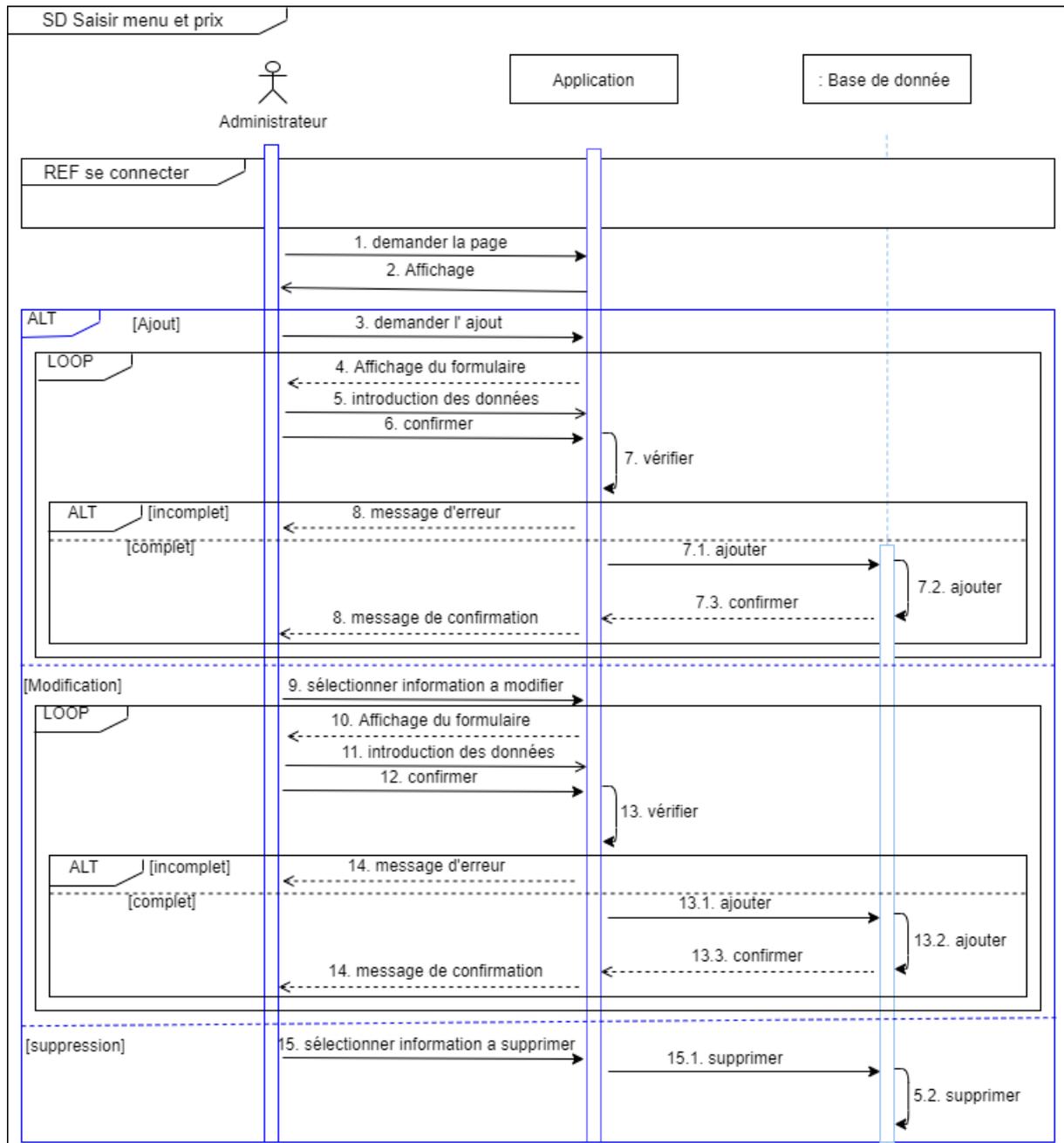


Figure II.5 -Diagramme de séquence système "saisir menu".

II.6. Conclusion

Dans ce chapitre nous avons présenté la partie analyse des besoins en se basant sur les diagrammes du langage UML, à savoir le diagramme de contexte dynamique, diagramme de cas d'utilisation ainsi que le diagramme de séquence système. Ceci nous a permis de comprendre et cerner, d'une façon globale, les fonctionnalités que le futur

ystème doit réaliser. Par la suite, en exploitant cette analyse, nous entamons la phase de conception dans le chapitre suivant.

CHAPITRE III

Conception

III.1. Introduction

Après avoir tracés les grandes lignes de phase de spécification des besoins, dans laquelle nous avons répondu à la question « QUOI ? », nous mettons l'accent, dans ce chapitre, sur la phase de conception. Cette phase a pour objectif de répondre à la question « COMMENT ? ».

En premier lieu, nous établirons, à base de diagrammes de séquence système vus dans le chapitre précédent, les diagrammes de séquence d'interaction, dans lesquels le système est remplacé par les objets qui interviennent pour réaliser les différents cas d'utilisation. Par la suite, et à base des objets entités, nous établirons le diagramme de classe du domaine. Enfin, nous présenterons les bases de données NoSQL, leurs avantages et leurs inconvénients, ainsi qu'une comparaison entre ce type de base de données et les bases de données SQL.

III.2. Diagramme de séquence d'interaction

Pour les diagrammes de séquence d'interaction, nous avons présenté deux cas d'utilisation « se connecter » et « Ajouter au menu »

La figure suivante (Figure III.1) montre l'interaction entre l'acteur Administrateur et le système afin de réaliser l'opération « Se connecter ».

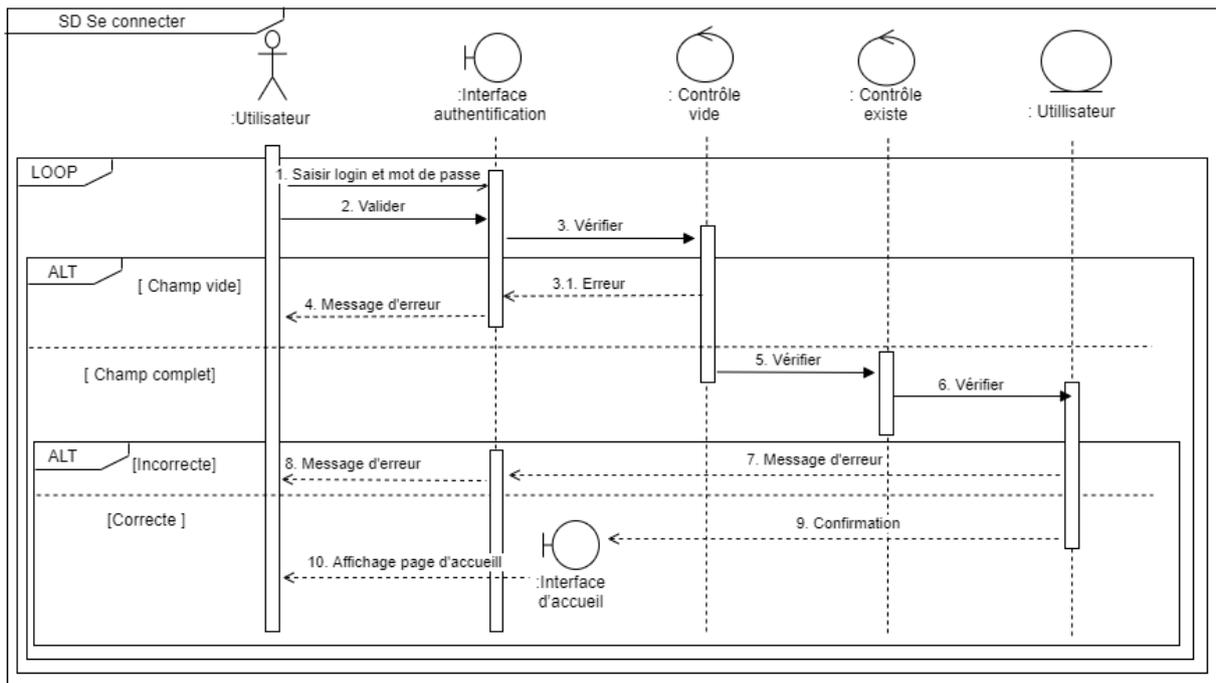


Figure III.1 – Diagramme se séquence d’interaction « Se connecter ».

La figure suivante (Figure III.2) montre l’interaction entre l’acteur Administrateur et le système afin de réaliser l’opération « Ajouter au menu ».

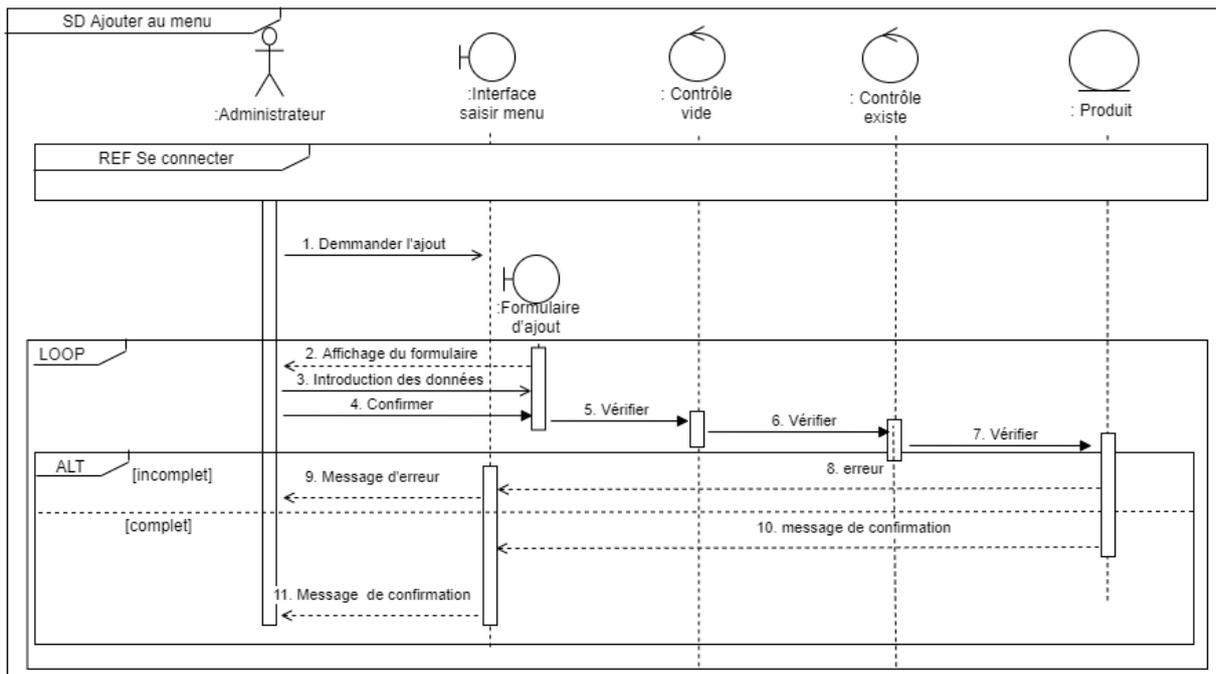


Figure III.2 – Diagramme se séquence d’interaction « Ajouter au menu ».

III.3. Diagramme de classe du domaine

Le diagramme de classe est un diagramme entités-associations décrivant les différentes classes, leurs structures et les associations statiques les unissant, il permet de décrire la structure interne des classes en termes d’attributs et d’opérations, et de représenter les associations statiques entre les classes [5].

En respectant les différentes règles de gestion de l’organisme d’accueil, VIDA LOCA, et en exploitant les différents objets entités établis dans les diagrammes de séquence d’interaction (voir la section précédente), nous avons établi le diagramme de classe suivant (Figure III.3) :

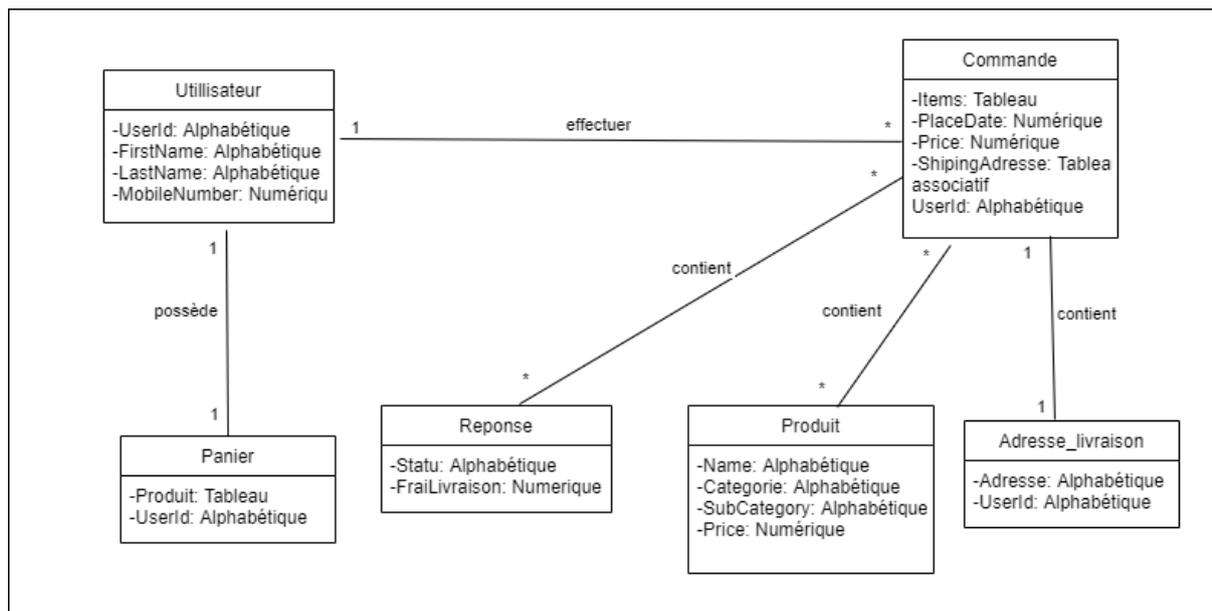


Figure IV.3 – Diagramme de classe de domaine.

III.3.1. Dictionnaire des données

Le tableau suivant (Tableau III.1) représente le dictionnaire des données représenté dans le diagramme de classe précédent

Nom de la table	Nom de l'attribut	Signification	Type
Utilisateur	UserId	Identifiant de l'utilisateur	Alphanumérique
	FirstName	Nom de l'utilisateur	Alphanumérique
	LastName	Prénom de l'utilisateur	Alphanumérique
	MobileNumber	Numéro de téléphone de l'utilisateur	Numérique
Commandes	Items	Tableau regroupant les données suivantes [id, name, quantité]	Tableau
	PlaceDate	Date et heure de prise de commande	Date

	Price	Prix de la commande	Numérique
	ShippingAdresse	Tableau associatif contenant les données suivantes : [adresse, mobileNumber, name]	Tableau Associatif
	UserId	Identifiant de l'utilisateur	Alphabétique
Panier	Produit	Tableau qui regroupe tous les produit du panier de l'utilisateur [id , Name , Quantité]	Tableau
	UserId	Identifiant de l'utilisateur	Alphabétique
Produit	Name	Nom du produit	Alphabétique
	Categorie	Catégorie du produit	Alphabétique
	SubCategory	Sous-catégorie du produit	Alphabétique
	Price	Prix du produit	Numérique
Adresse_livraison	Adresse	Tableau qui regroupe les données suivantes [adresse , MobileNumber , Number]	Tableau
	UserId	Identifiant de l'utilisateur	Alphabétique

Tableau III.1 -Dictionnaire des données

III.4. Nosql

III.4.1. Définition

Les bases de données NoSQL (également appelées Not Only SQL Databases) sont des systèmes de base de données non relationnels utilisés pour stocker et récupérer des données. Ces dernières se caractérisent par une logique de représentation non relationnelle et qui n'offrent donc pas une interface de requêtes en SQL.

Les bases de données NoSQL ont été spécifiquement développées par les principales sociétés Internet telles que Google, Yahoo, Amazon, etc, car les bases de données relationnelles existants n'étaient pas en mesure de faire face aux exigences croissantes en matière de traitement des données [8].

Il existe 4 familles de base de données Nosql qui se représente comme suit

- Famille clé-valeur

Cette catégorie de bases fonctionne comme une table associative clé/valeur ce qui en fait une base simple à mettre en place et permet un accès rapide aux informations (système de cache) [8].

- Famille orientée colonne

La représentation se fait par colonnes et non par lignes comme pour un système classique, il s'agit peut-être du modèle le plus difficile à se représenter [8].

- Famille orientée document

Cette famille est une extension de la famille clé/valeur en associant une clé à un document hiérarchique comme le XML, le JSON.

Ici on va pouvoir stocker toute forme de structure non plane simplement, pouvoir profiter de la puissance de l'indexation en ciblant les balises du document et ainsi bénéficier d'une interrogation simplifiée (absence de jointure) [8].

- Famille orientée graphe

Le principe de cette famille repose sur le fait d'utiliser des objets en s'appuyant sur la théorie des graphes (représentation de nœuds et d'arcs) : chaque élément connaît son (ses) voisin(s).

Un tel principe permet de mettre facilement des algorithmes de parcours de graphe (recherche du plus court chemin, nœud(s) ayant une position centrale, etc) [8].

III.4.2. Avantages

L'évolutivité se fait de manière horizontale (pour augmenter les performances on ajoute de nouvelles machines).

Les données sont distribuées sur plusieurs machines (sharding) de ce fait on évite les goulets d'étranglements lors de la récupération des données (fortes performances de lecture).

La représentation des données est notable par l'absence de schéma (schemaless).

La majorité des solutions est Open Source, néanmoins il existe des Support Pro pour répondre aux besoins des entreprises [8].

III.4.3. Inconvénients

- Il n'existe pas de langage d'interrogation standardisé : chaque éditeur a mis en place le sien
- La mise en œuvre d'un environnement fortement transactionnel (fort besoin d'écriture) où le séquençement des écritures est primordial, reste complexe puisque l'architecture est distribuée compliquant l'atomicité et la cohérence des transactions
- L'écriture de requêtes complexes est difficile à mettre en œuvre
- L'offre NoSQL est segmentée en plusieurs familles où chacune répond à un besoin précis [8].

III.4.3. La différence entre les bases de données Sql et les bases de données Nosql

	Sql	Nosql
Type	bases de données relationnelles (SGBDR)	base de données non relationnelle ou distribuée
Langue	langage de requête structuré (SQL)	syntaxe varie d'une base de données à une autre
L'évolutivité	Verticale	Horizontale
La structure	Tables	Paires clé-valeur Données de graphiques Documents Magasins à colonnes larges

Propriété suivie	ACID (atomicité, cohérence, isolation et durabilité)	théorème de Brewers CAP (cohérence, disponibilité et tolérance de partition)
-------------------------	--	--

Tableau III.2 – Différence entre le Sql et le Nosql.

III.5. Conclusion

Ce chapitre a été consacré à la partie conception, dans laquelle nous avons établi le diagramme de classe après avoir détaillé la communication entre les différents acteurs et les objets constituant le système. La prochaine étape de notre projet est la phase de réalisation et mise en œuvre, qui sera l'objet du chapitre suivant.

CHAPITRE IV

Réalisation

IV.1. Introduction

À ce stade du processus, les cas d'utilisation sont bien cernés, le problème a été analysé, et nous avons défini une conception appropriée aux besoins de l'application. Nous pouvons alors entreprendre l'implémentation.

Premièrement dans ce chapitre, nous illustrerons comment nous allons déployer, valider notre application, ensuite, nous survolerons quelques langages utilisés pour la phase de réalisation. Enfin, nous présenterons l'IHM de notre application.

IV.2. Environnement de programmation et bibliothèques

IV.2.1. Flutter

Flutter est un SDK ou Framework qui a été créé et développé par Google. Il nous permet de créer des applications **Android et/ou IOS**. Flutter est donc un Framework très récent jusqu'au moment de la rédaction d'articles (Octobre 2018), C'est l'une des briques essentielles de Fuchsia/Andromeda. En effet ce framework est utilisé pour tout ce qui est interface utilisateur. Mais aujourd'hui Flutter se fait surtout connaître pour sa capacité à concevoir des applications natives multiplateforme pour Android et iOS [9].

a.Pourquoi flutter

Flutter se positionne différemment par rapport aux frameworks multiplateformes existants, notamment avec l'obtention d'une application native possédant une interface utilisateur n'utilisant pas de composants natifs, ni de webview. En effet tous les composants de notre interface utilisateur sont construits par le moteur graphique de Flutter écrit en C++ et dessiné par les équipes de Google, en étroite relation avec les équipes de Material Design [10].

b.Principe de flutter

- Développement rapide : Utilisation d'un riche ensemble de widgets entièrement personnalisables pour créer des interfaces natives en quelques minutes.

- Interface utilisateur expressive et flexible : L'architecture en couches permet une personnalisation complète, ce qui se traduit par un rendu incroyablement rapide et des conceptions expressives et flexibles.

- Performance native : Les widgets de Flutter intègrent toutes les différences critiques de plateforme telles que le défilement, la navigation, les icônes et les polices, et votre code Flutter est compilé en code machine ARM natif à l'aide des compilateurs natifs de Dart.

- Hot Reload : cette fonction de Flutter nous aide à expérimenter rapidement et facilement, à créer des interfaces utilisateur, à ajouter des fonctionnalités et à corriger les bugs. Hot Reload fonctionne en injectant des fichiers de code source mis à jour dans la machine virtuelle Dart (VM) en cours d'exécution. Une fois que la VM a mis à jour les classes avec les nouvelles versions de champs et de fonctions, le framework Flutter reconstruit automatiquement l'arborescence des widgets, permettant de visualiser rapidement les effets de nos modifications [9].

c.Dart

Dart est un langage de programmation développé par la communauté Google. La première version date de 2011. Le but du développement de ce langage est de remplacer Javascript afin d'éviter les limites de performance de ce dernier. De plus, Dart peut aussi être utilisé pour la programmation de serveurs ainsi que le développement mobile avec le framework Flutter [9].

Ce langage a été créé principalement pour faire de grosses webapps (**easy to scale**). Voici le point essentiel de cette technique :

- Résoudre des problématiques de performance des Web Apps.
- Taux de productivité très élevé par rapport aux autres technologies.
- La facilité d'apprendre et surtout, pas de mauvaises surprises.
- Une plateforme et un système assez cohérents.
- Fournir les bonnes pratiques et des explications bien détaillées.
- Une documentation complète.
- Un langage Orienté objet basé sur les classes, héritage unique, typé optionnellement [9].

IV.2.2. SDK

Un **SDK**, pour Software Développement Kit, désigne un ensemble d'outils utilisés par les développeurs pour le développement d'un logiciel destiné à une plateforme déterminée (Linux, Windows, Android, etc.). On le traduit en français par kit de développement.

Généralement, un **SDK** est imaginé et mis au point pour un seul ou plusieurs langages de programmation. Il peut avoir une seule ou plusieurs cibles comme un système d'exploitation, une application web, un serveur web, un jeu vidéo, etc. Ces dernières années, on le retrouve fréquemment dans le développement des applications mobiles [11].

IV.2.3. JSON

JSON (JavaScript Object notation) : un format léger qui permet de décrire des objets en JavaScript, il est principalement utilisé pour faire des échanges entre JS et un serveur web, JSON se base sur deux structures :

- Une collection de couple nom/valeur : {nom : valeur ;}
- Un tableau qui est une collection de valeurs ordonnées [valeur, valeur]

JSON est un format de données. Autrement dit, c'est une façon de stocker des informations, un peu comme une base de données, il est maintenant lié à JavaScript qui inclut un objet_JSON, et de nombreux développeurs l'incorporent quasiment comme un sous-ensemble du langage [5].

IV.3. Implémentation de la base de données

Dans notre projet, nous avons utilisé la Platform Firebase, afin de communiquer les données entre la partie cliente mobile et le serveur web.

IV.3.1. Firebase

Firebase est une plateforme d'application mobile et web qui fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à élargir leur base d'utilisateurs et à générer davantage de profits [7].

Firebase contient plusieurs types de services de gestion de bases de données.

a- Firebase authentication

L'authentification Firebase fournit des services d'arrière-plan, des kits de développement logiciel (SDK) faciles à utiliser et des bibliothèques d'interface utilisateur prêtes à l'emploi pour authentifier les utilisateurs auprès de votre application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros

de téléphone, de fournisseurs d'identité fédérés populaires tels que Google, Facebook et Twitter, etc.

b-RealTime Data Base

Realtime Database Firebase est une base de données hébergée dans le cloud. Les données sont stockées sous forme de JSON et synchronisées en temps réel sur chaque client connecté. Lorsque nous créons des applications multi-plateformes avec kits de développement logiciel (SDK) iOS, Android et JavaScript, tous vos clients partagent une instance de base de données en temps réel et reçoivent automatiquement les mises à jour avec les données les plus récentes.

c- Firebase Storage

Cloud Storage for Firebase (stockage en nuage), est un service de stockage d'objets puissant, simple et économique, conçu pour Google. Cloud Storage est conçu pour les développeurs d'applications qui ont besoin de stocker et de diffuser du contenu généré par l'utilisateur, telles que des photos ou des vidéos.

d-Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) est une solution de messagerie multi-plateforme qui permet de livrer des messages de manière fiable, sans frais. Il permet d'envoyer des messages de notification aux utilisateurs de l'application.

e-Cloud Firestore

Cloud Firestore est une base de données flexible et évolutive pour le développement mobile, Web et serveur de Firebase et de Google Cloud Platform. Comme Firebase Realtime Database, il maintient les données synchronisées entre les applications clientes via des écouteurs en temps réel et offre une prise en charge hors ligne pour le mobile et le Web afin que vous puissiez créer des applications réactives qui fonctionnent indépendamment de la latence du réseau ou de la connectivité Internet [7].

Dans ce projet nous avons utilisé firebase authentication ainsi le cloud firestore pour notre base de données et le cloud storage pour stocker les images ajoutées par le client.

IV.3.2. Comment ajouter Firebase à notre projet ?

Pour faire fonctionner notre application avec Firebase, nous avons suivi les étapes suivantes :

- Créer un compte Gmail dans Google.
- Enregistrer l'application dans la plateforme de Firebase en remplissant les champs nécessaires.

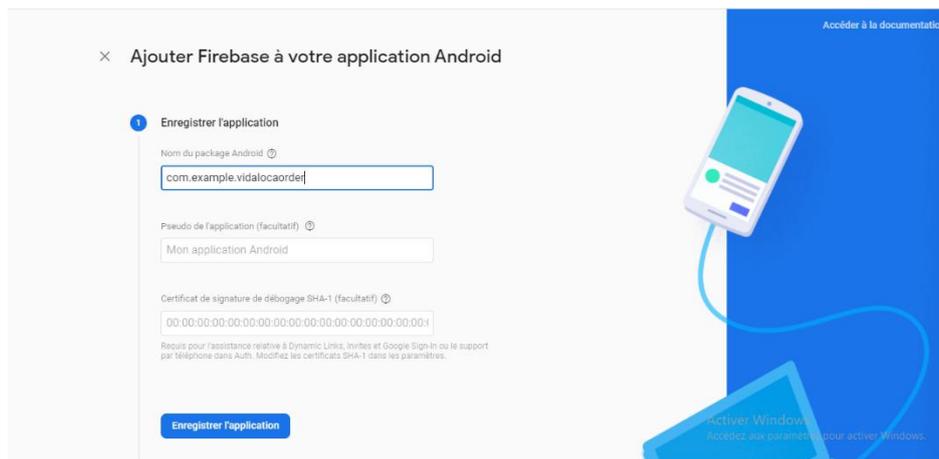


Figure IV.1 – Première étape de l'implémentation-Enregistrement de l'application.

- Télécharger le fichier contenant les services de Google, et le placer dans le dossier sous le nom « app » située dans le package de projet

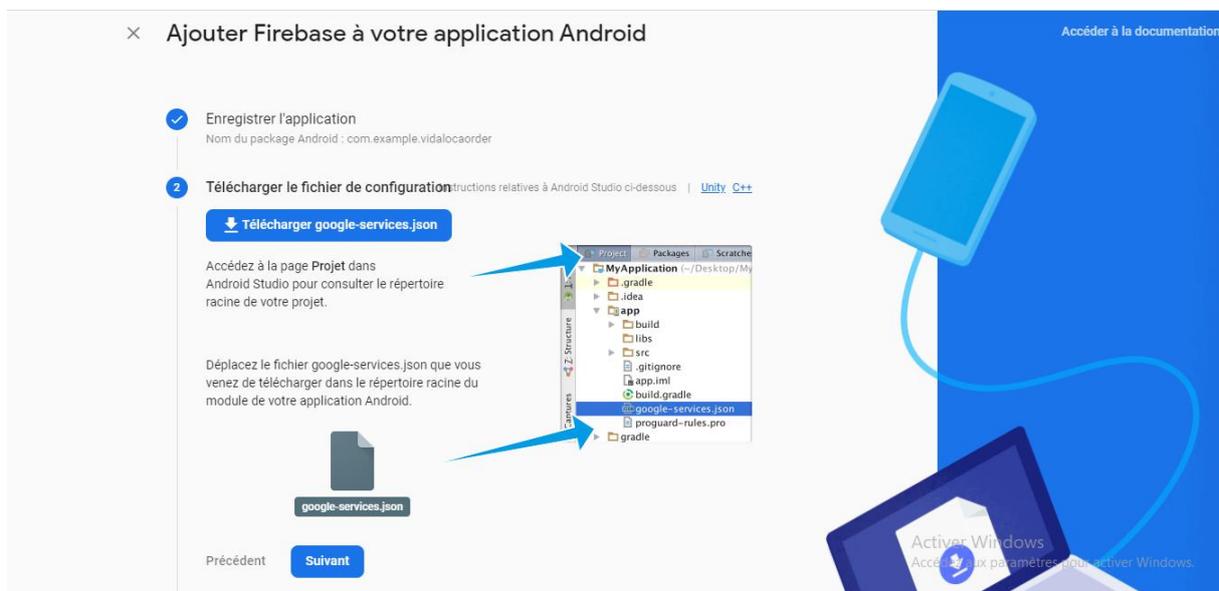


Figure IV.2 - Téléchargement du fichier de configuration.

- Ajouter le SDK de Firebase au niveau de projet et de m'application, puis appuyer sur « synchroniser » dans la barre qui apparait sans l'IDE.

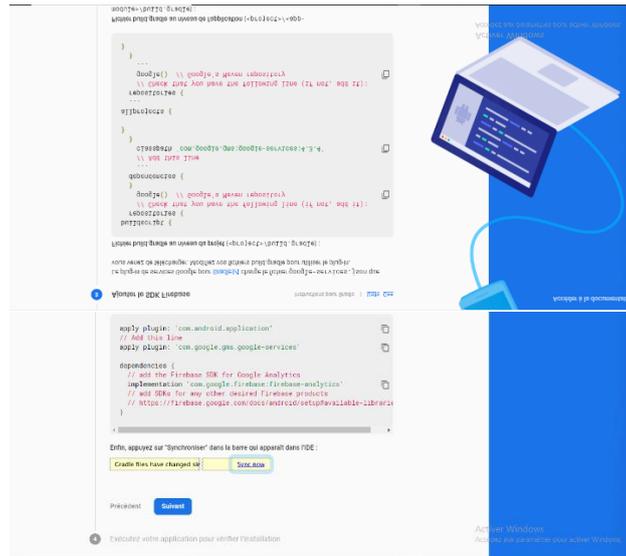


Figure IV.3 - Ajout du SDK Firebase.

- Exécuter l'application pour vérifier l'installation de SDK
- Enfin, accéder à la console de Firebase.

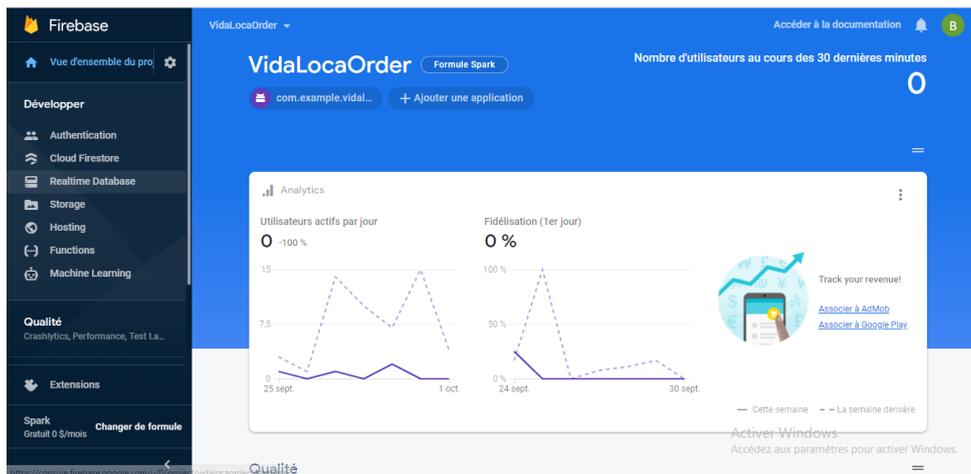


Figure IV.4 - la console de Firebase.

IV.4. Architecture de l'application

IV.4.1. Coté Utilisateur

La Figure suivante (Figure IV.5) représente l'architecture de l'application « coté utilisateur ».

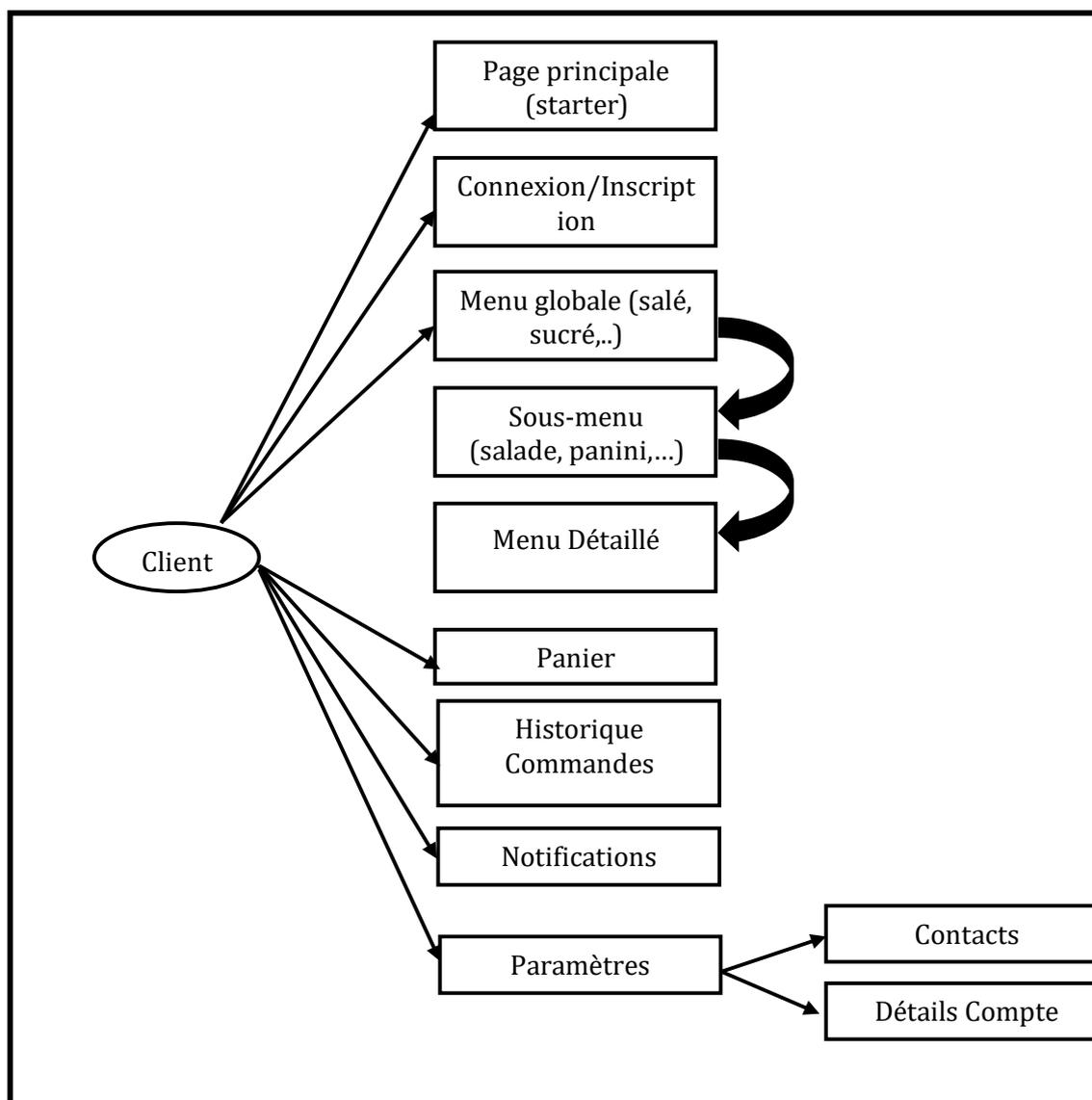


Figure IV.5 – Architecture de l'application « côté utilisateur ».

IV.4.2. Coté administrateur

La Figure suivante (Figure IV.6) représente l'architecture de l'application « coté administrateur ».

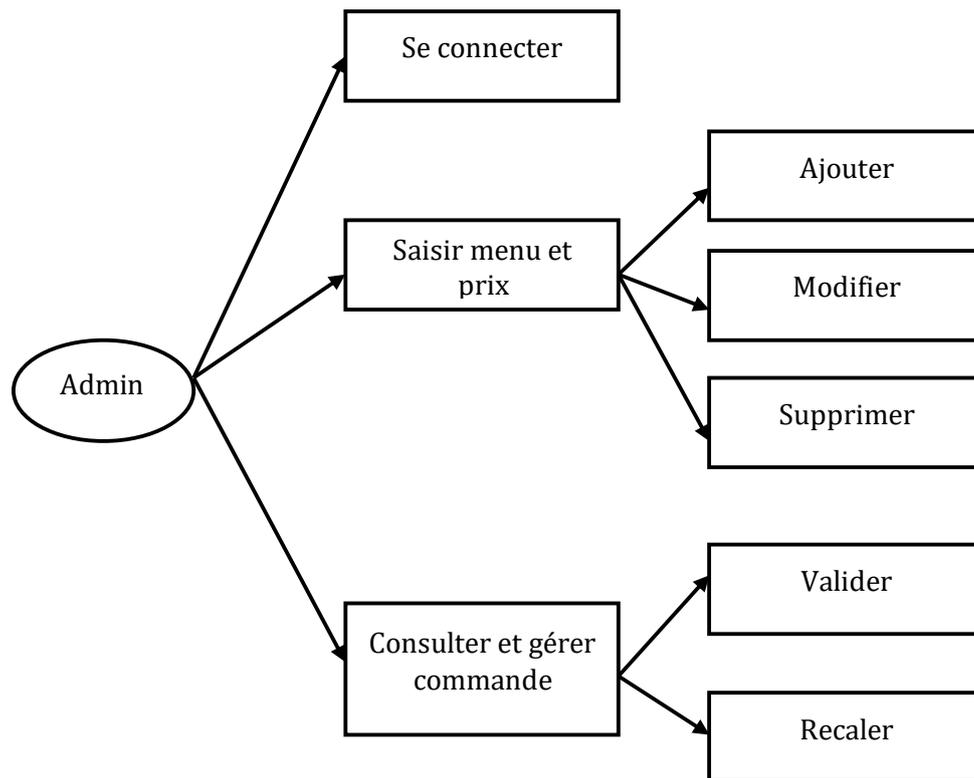


Figure IV.6 – Architecture de l’application « coté administrateur ».

IV.5. Phase de déploiement

Dans cette phase de déploiement, nous mettrons en œuvre le processus de mise en service, le déploiement de notre application met en jeu le processus de mise en ligne de l’application web. En plus de cela, nous verrons comment nous allons tester et valider notre travail avec les dirigeants de « VIDA LOCA »

IV.5.1. Architecture client/serveur

Notre application est conçue sous la forme d’une architecture client/serveur.

a- Client

Dans notre projet, le client s’agit de l’application cliente. Il a les droits pour ajouter et modifier ou charger ses données personnelles, et des taches supplémentaires.

b- Serveur

Toutes les données dans notre application sont dans le serveur de Google (Firebase).

IV.5.2. Mise en ligne

Dans le but d'offrir un service de qualité aux visiteurs, il faut donc avoir recours au service d'un hébergeur de qualité, du côté de « VIDA LOCA », ils nous ont simplement exigé une rapidité de service et une sécurité optimale.

Vu la complexité du système binaire, notre application devra être capable de surmonter les connexions qui vont être multipliées du fur et à mesure.

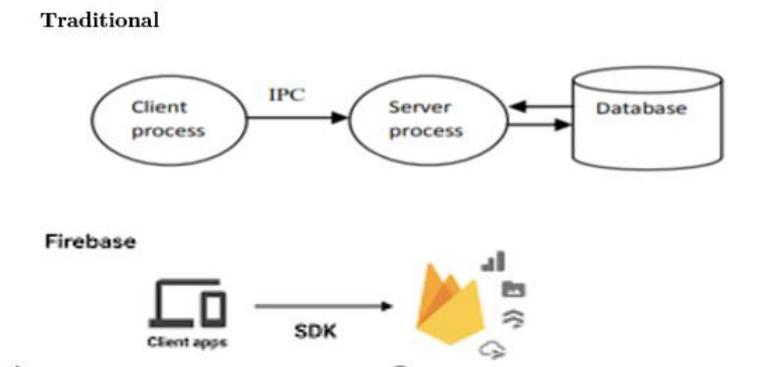


Figure IV.7 Architecture client/serveur [13].

IV.5.3. Hébergement d'une application

L'hébergement est un service visant à rendre un site ou une application web accessible sur internet.

Afin que les pages soient visibles par tout le monde, il faut qu'elles soient stockées sur un ordinateur connecté en permanence à l'internet (serveur).

Un hébergeur est comme une société spécialisée dans l'hébergement de sites web sur des serveurs, sa principale activité sur internet consiste à installer des serveurs, à les sécuriser, à les tenir à jour en installant les mises à jour de sécurité pour éviter les attaques malveillantes et à les réparer en cas de panne [12].

Il existe deux types d'hébergements :

- Hébergement gratuit

Ce type d'hébergement offre un service basique, avec un espace de stockage assez limité, avec une fiabilité qui n'est pas toujours optimale.

- Hébergement payant

Ce type d'hébergement propose un service de qualité, avec de bonnes capacités de stockage ainsi qu'une grande fiabilité.

Le support sera également plus disponible et plus réactif en cas de problème et aidera avec des réponses personnalisées [12].

IV.6. Interfaces homme-machines (IHM)

IV.6.1. Interfaces coté clients

a- Interface d'accueil et Connexion/inscription

Les figures ci-dessous montrent la première interface qui sera affichée pour l'utilisateur, qui permet à son tour de rediriger ce dernier vers l'interface Connexion ou bien Inscription.

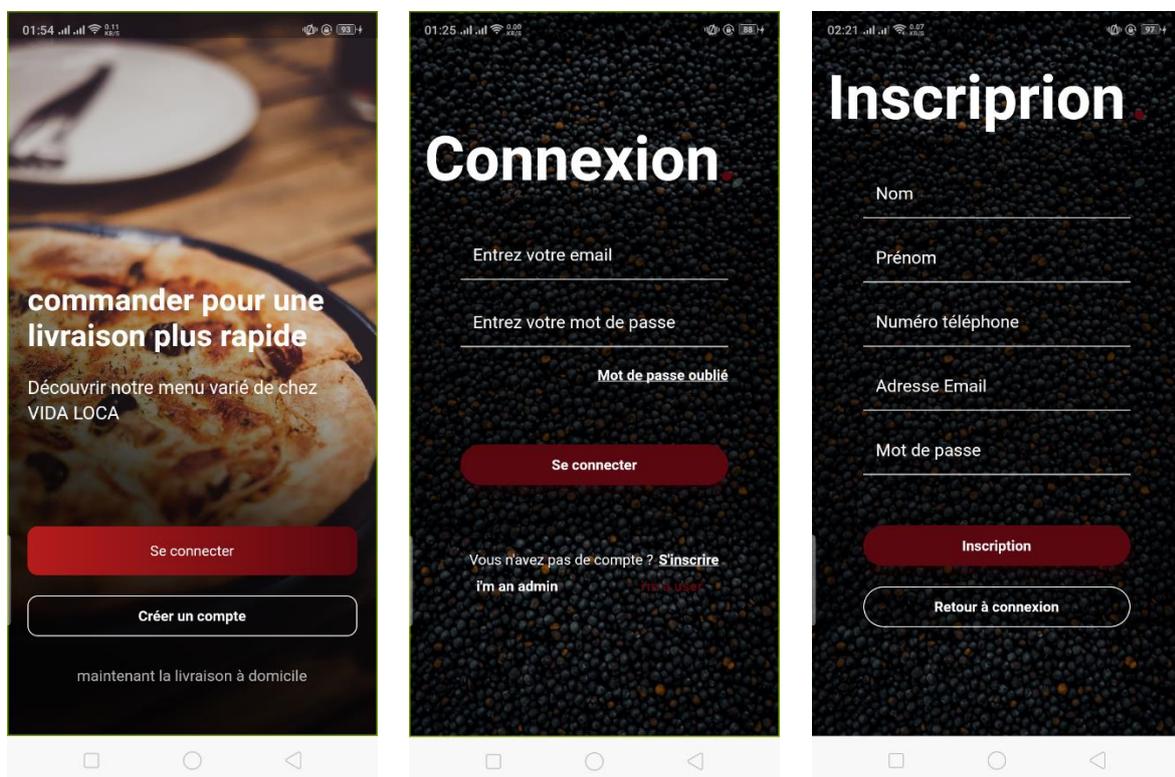


Figure IV.8 -Interface d'accueil et interface de connexion/inscription

b- Interfaces Menu

Ces figures montrent le menu global du Restaurant VIDA LOCA. Un menu détaillé sera affiché en cliquant dans une catégorie précise l'interface du menu détaillé sera affichée.

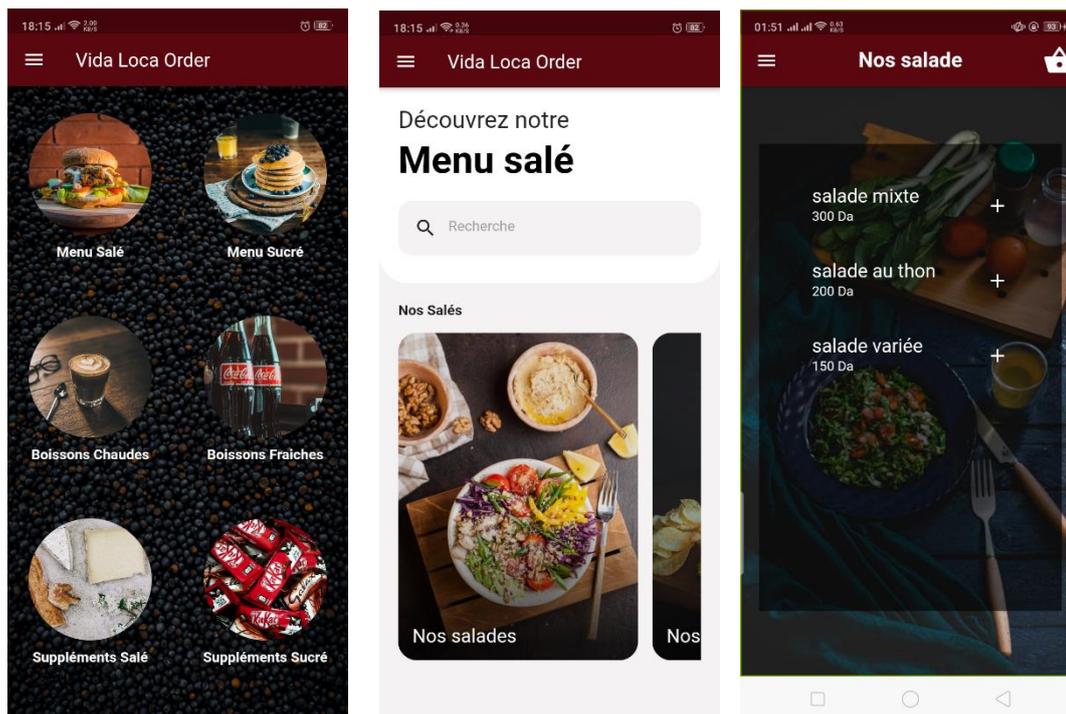


Figure IV.9 - Interfaces Menu globale et détaillé.

c- Interfaces Panier

En cliquant sur un ajouter au panier, l'interface ci-dessus apparaisse, dans cette interface nous pouvons modifier, supprimer et lancer la commande.

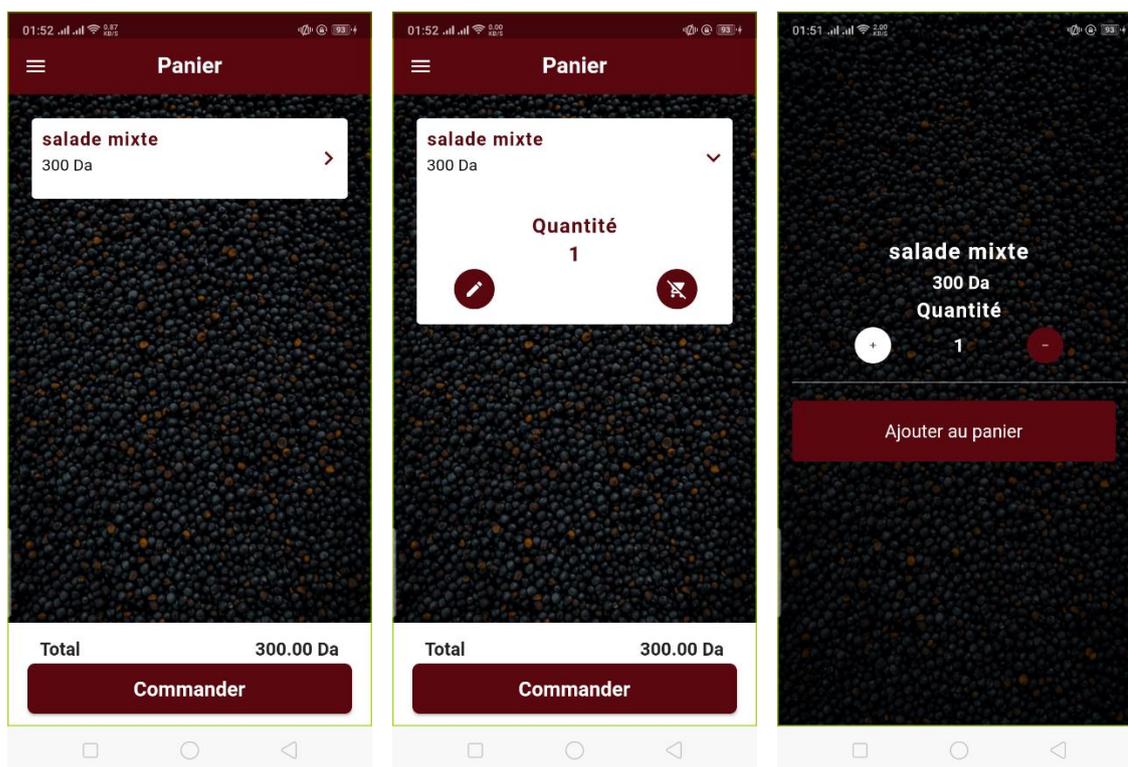


Figure IV.10 –Interface Panier.

d- Interfaces Lancement de la commande

En cliquant sur commander dans l'interface panier, le client sera redirigé vers un formulaire afin de remplir ses informations et son adresse, puis sélectionner l'adresse voulu par la suite confirmer la commande en cliquant sur le bouton passer commande.

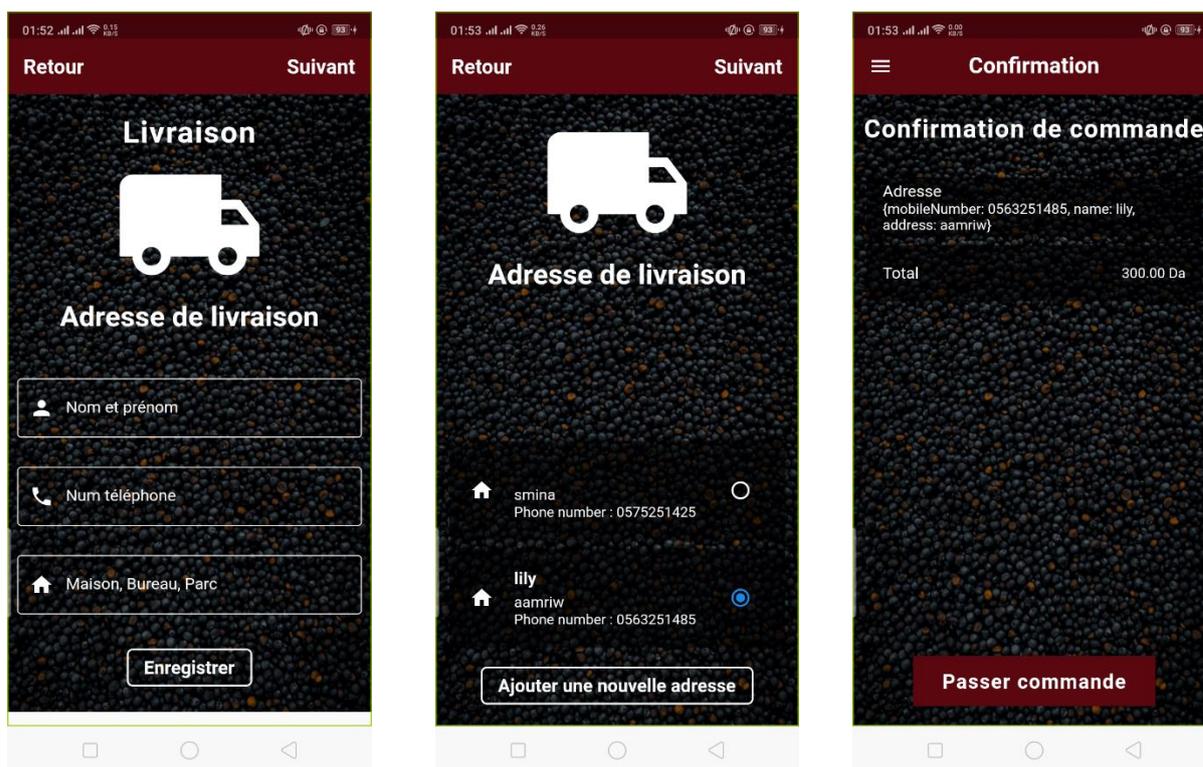


Figure IV.11 –Interface lancement commande.

IV.6.2. Interfaces coté administrateurs

a- Interface Accueil

Cette interface est affichée une fois l'administrateur est connecté, il a le choix d'ajouter des produits au menu, modifier les produits et consulter les commandes.



Figure IV.12 –Interface d'accueil.

b- Interface Saisir menu

Cette interface permet d'ajouter des produits en précisant le nom, la catégorie, sous-catégorie et prix du produit pour obtenir le menu.

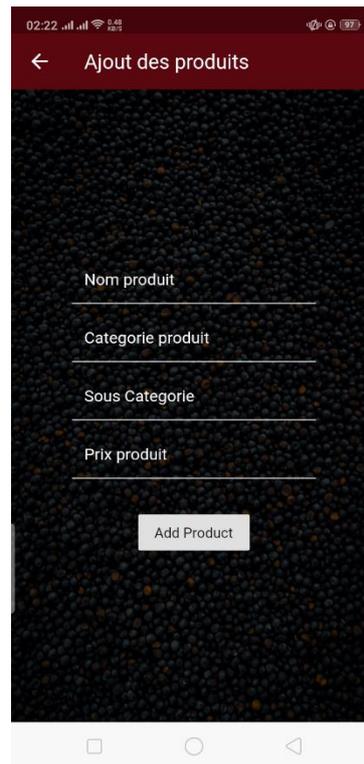


Figure IV.13 – Interface ajout des produits.

c- Interface Gestion des produits

Cette interface permet à l'administrateur de modifier et supprimer les produits du menu.

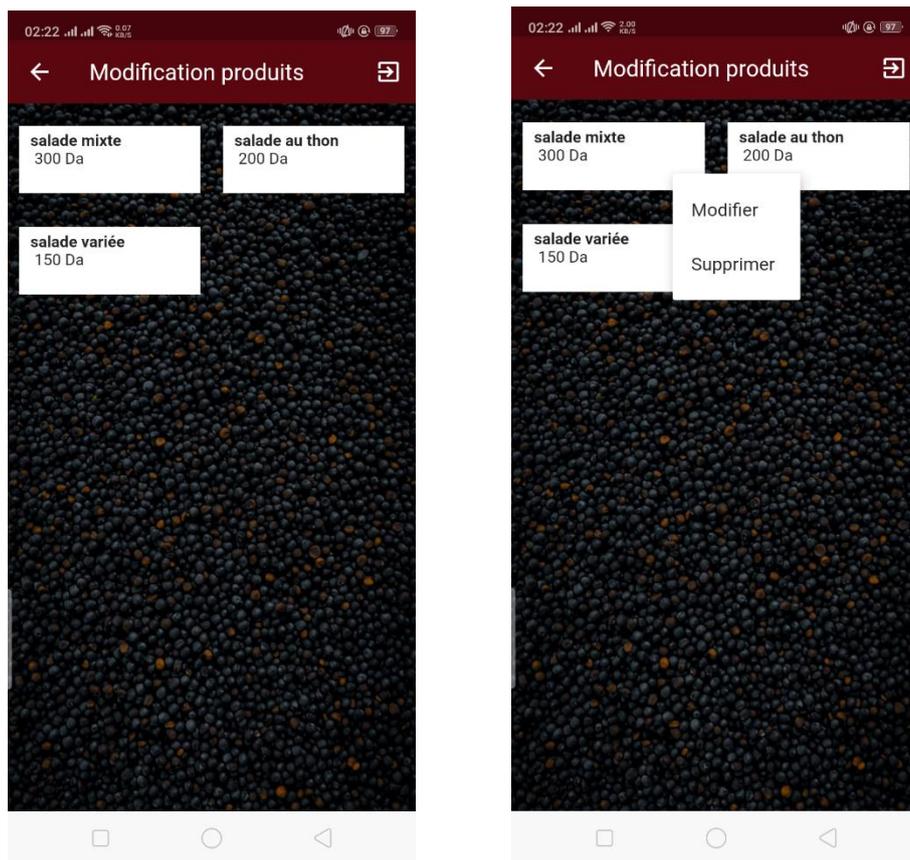


Figure IV.14 –Interface gestion produits.

d- Interface Gestion Commande

Cette interface permet à l'administrateur de consulter les commandes, il peut accepter ou refuser la commande, s'il accepte la commande il doit saisir les frais de livraison et en confirmant la réponse un message de confirmation est envoyé.

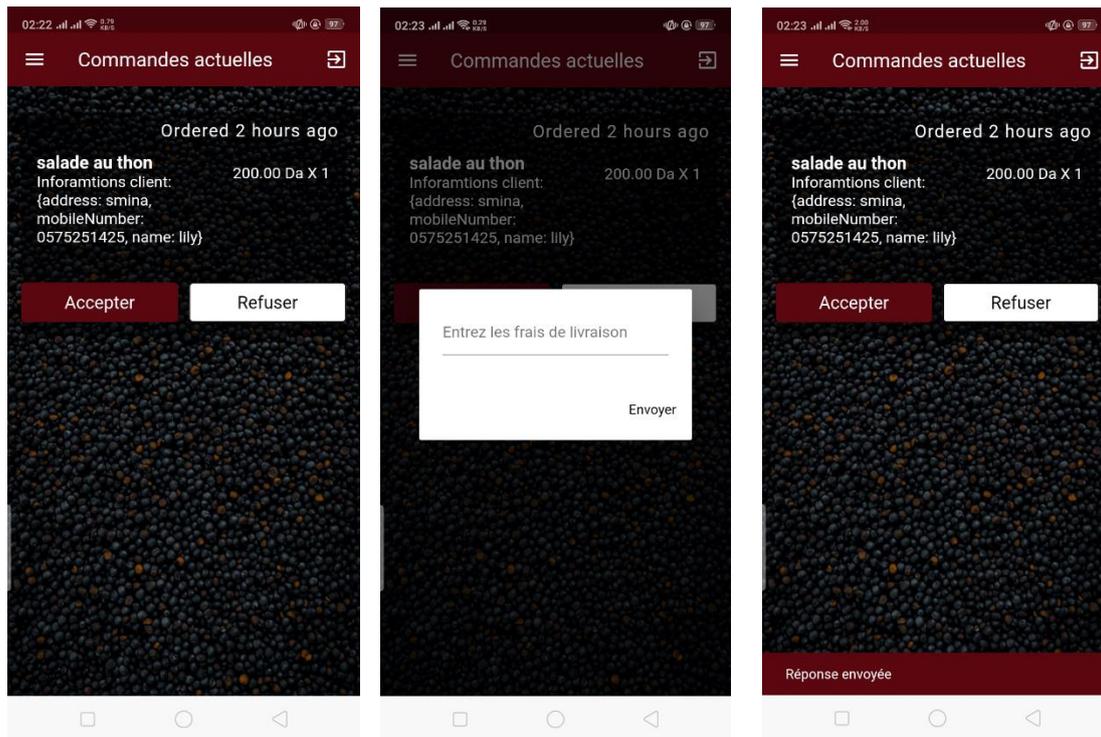


Figure IV.15 –Interface gestion commandes.

IV.7. Scénarios de tests

IV.7.1. Validation

La validation de notre système ne peut se faire qu'à travers une phase de tests sur un sous ensemble représentatif des acteurs et des données impliquées.

A cette fin, nous avons entrepris une expérimentation avec quelques clients, et les gérants de « VIDA LOCA ». Toutes les fonctionnalités prévues lors de la conception ont été testées. A part des détails techniques d'interfaces et de contrôle de saisie. Ces fonctionnalités répondent correctement aux exigences de « VIDA LOCA ». Ceci dit, il faut préciser qu'une autre phase de validation va être opérée sur le terrain d'exploitation de l'application « VIDA LOCA ». En effet, c'est cette phase qui va nous permettre de voir comment les utilisateurs effectifs de ce produit vont réagir et comment ils vont l'apprécier.

IV.7.2. Guide d'utilisation

Pour aider les clients de « VIDA LOCA » à se familiariser avec l'application web, nous avons créé quelques vidéos qui expliquent les services fondamentaux tels que : la préinscription, le parrainage, ainsi que la modification des informations et la consultation de l'arbre, etc.

IV.8. Conclusion

Dans ce dernier chapitre nous avons présenté la partie réalisation en décrivant l'ensemble des langages de programmation utilisé, l'environnement de développement ainsi que les fonctionnalités de base de l'application que nous avons mis en valeur avec quelques interfaces.

Conclusion générale

Dans ce travail de projet de fin d'études, nous avons présenté les différentes étapes de réalisation d'une application mobile pour la gestion des livraisons du restaurant VIDA LOCA.

Nous avons commencé d'abord, par la présentation de l'organisme d'accueil. Nous avons également abordé la problématique de la livraison traditionnelle. Ensuite, nous avons présenté des diagrammes UML permettant de bien comprendre la structure statique et dynamique du système.

Enfin, nous avons présenté les principaux outils qui nous ont permis de bien développer notre application tel que le framework Flutter. Ainsi, Firebase pour le développement de la base de données et la gestion des événements. Ces choix nous ont permis de réaliser une application répondant aux problèmes de gestion de livraison en ligne.

Ce projet a été pour nous l'occasion d'améliorer et renforcer nos connaissances théoriques et pratiques en conception et en développement mobile. L'aspect concret de ce projet nous a permis de faire le lien entre les notions théoriques acquises pendant notre cursus d'étude et le mode de développement de logiciel d'une façon générale, et le développement mobile d'une façon particulière.

À l'issue de ce projet, notre application est réalisée dans sa première version. Elle restera toujours ouverte à des perspectives d'amélioration. Nous envisageons d'ajouter la fonctionnalité « *trackOrder* » qui permet de suivre la commande en temps réel, et d'autres nouvelles fonctionnalités selon les recommandations des utilisateurs. Ainsi, d'héberger notre application.

Liste des références

- [1] P. Roques, F. Vallée. '*Processus de développement UML par action de l'analyse des besoins à la conception*'. EYROLLES, 2004, 4ème édition.
- [2] Toussaint Frédéric. '*Les différents concepts de restauration*'. Page web. URL : <http://technorestor.org/tr/concepts/index.html>. publié en 2011 . Date de consultation : 16/06/2020.
- [3] The Apache Software Foundation. '*Apache Tomcat 7 User Guide*'. Corporation, 2011.
- [4] Joseph Gabay, David Gabay. '*UML2 Analyse et conception*'. DUNOD, 2008. 1ere édition.
- [5] AIT HATRIT Fatima et al. '*Conception et réalisation d'un site web dynamique*'. Cas d'étude l'aéroport « ABANE RAMDANE » De Bejaia. Mémoire de Master. Université ABDERAHMANE MIRA, 2014/2015.
- [6] Amokrane chanez et al. '*Conception et réalisation d'une application web de gestion de stock cas d'étude « SONELGAZ » De Bejaia*'. Mémoire de Master. Université ABDERAHMANE MIRA. 2017/2018.
- [7] 'Firebase Guide | Firebase'. Page web. URL: <https://firebase.google.com/docs/guides>. Date de consultation :23/09/2020.
- [8] S. Crozat. '*Conception de bases de données. Bases de données non-relationnelle*'. utc Formation. 4 septembre 2017.
- [9] 'Flutter docs | Flutter '. Page web. URL : <https://flutter.dev/docs>. Date de consultation 23/09/2020.
- [10] Hugo Extrat. '*Flutter : tout le monde en parle mais pourquoi ?*'. Page web. URL : <https://www.technologies-ebusiness.com/solutions/pourquoi-flutter>. Date de consultation 25/08/2020.
- [11] Domonique REVUZ, Etienne DURIS. '*Le système d'exploitation android*'. Page web. URL : <http://igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/>. Date de consultation 13/08/2020.
- [12] Gilberto Antonio. '*L'hébergement d'application et de sites internet*' cas d'étude « *L'implémentation de la plateforme web Vault d'EDS* ». Mémoire de Master, ESSEC MANAGEMENT EDUCATION, 2001/2002.

- [13] H.S. Oluwatosin. *Client-server model*. IOSR Journal of Computer Engineering (IOSR-JCE), VOL. 16, Issue 1, pp. 2278-8727. Feb 2014.

Annexe A

1. Diagramme de séquence cas « Gérer commande ».

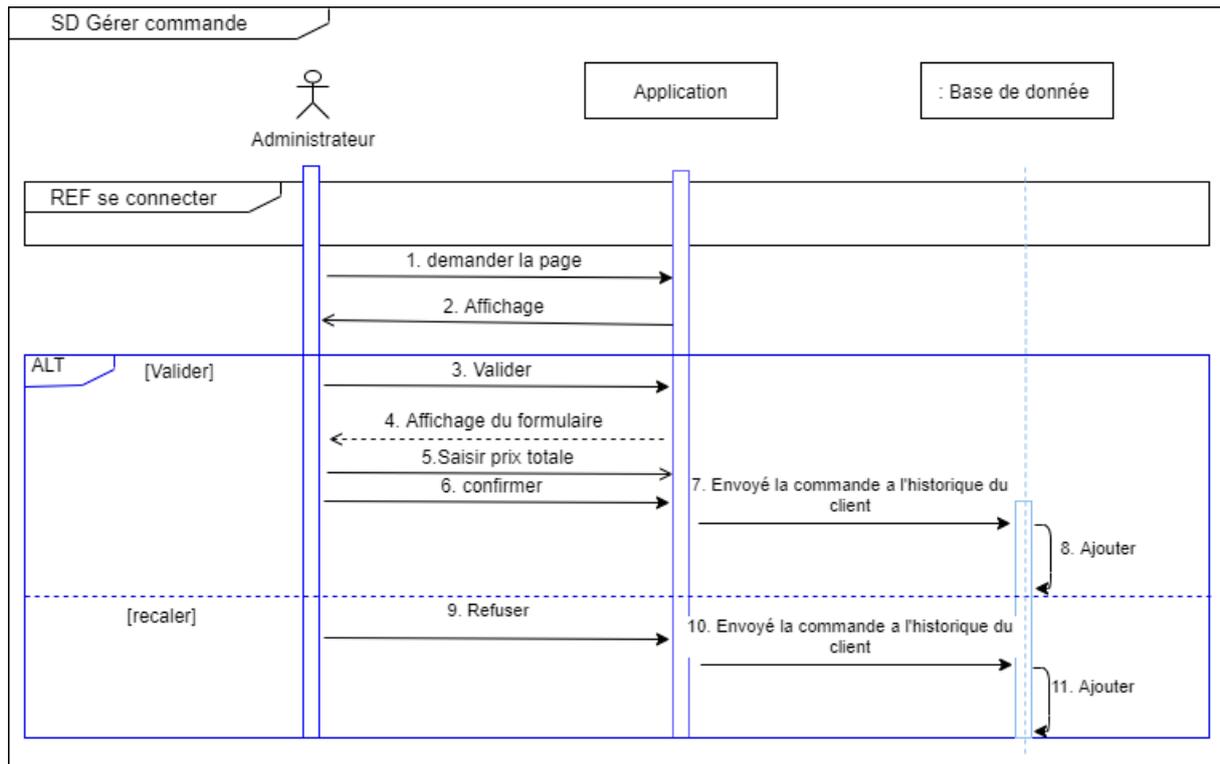


Figure 5 – Diagramme de séquence système cas « Gérer commande »

2. Diagramme de séquence cas « Consulter panier ».

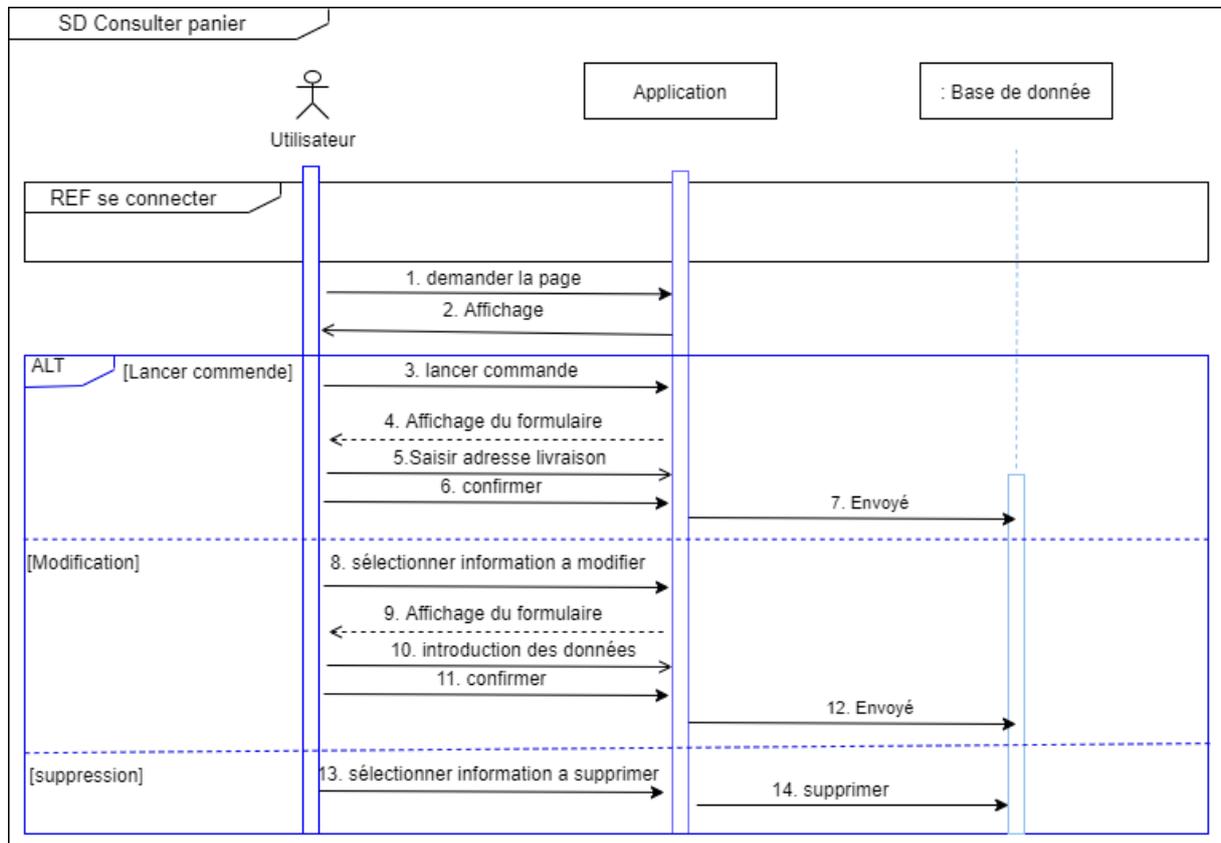


Figure 6 – Diagramme de séquence système cas « Consulter panier »

Annexe B

1. Interface utilisateur

a- Interface historique des commandes

Cette interface affiche toutes les commandes faites par le client.



Figure 1 –Interface mes commandes

b- Interface Notification

Cette interface illustre les accusés de réception des commandes passées par le client avec les frais de livraison qui dépend de l'adresse envoyée (l'accusé peut être positif ou bien négatif).

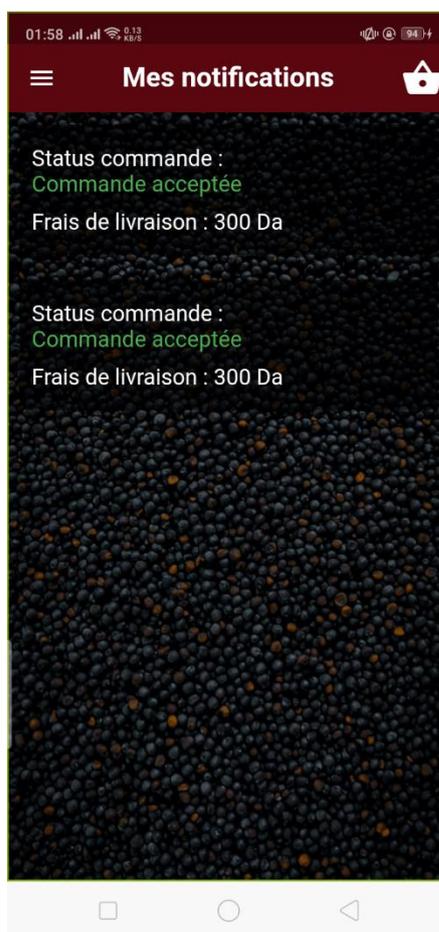


Figure 2 –Interface mes notifications.

c- Interface Paramètre

Les interfaces ci-dessous représentent les détails du compte client et les contacts du restaurant VIDALOCA, dans la première interface nous avons accès à la galerie du téléphone (après demande d'autorisation par le système), dans l'interface contact en appuyant sur le bouton appelez nous, le client sera redirigé directement à l'interface de numérotation du téléphone, et pour la dernière interface, le client peut modifier ses informations.

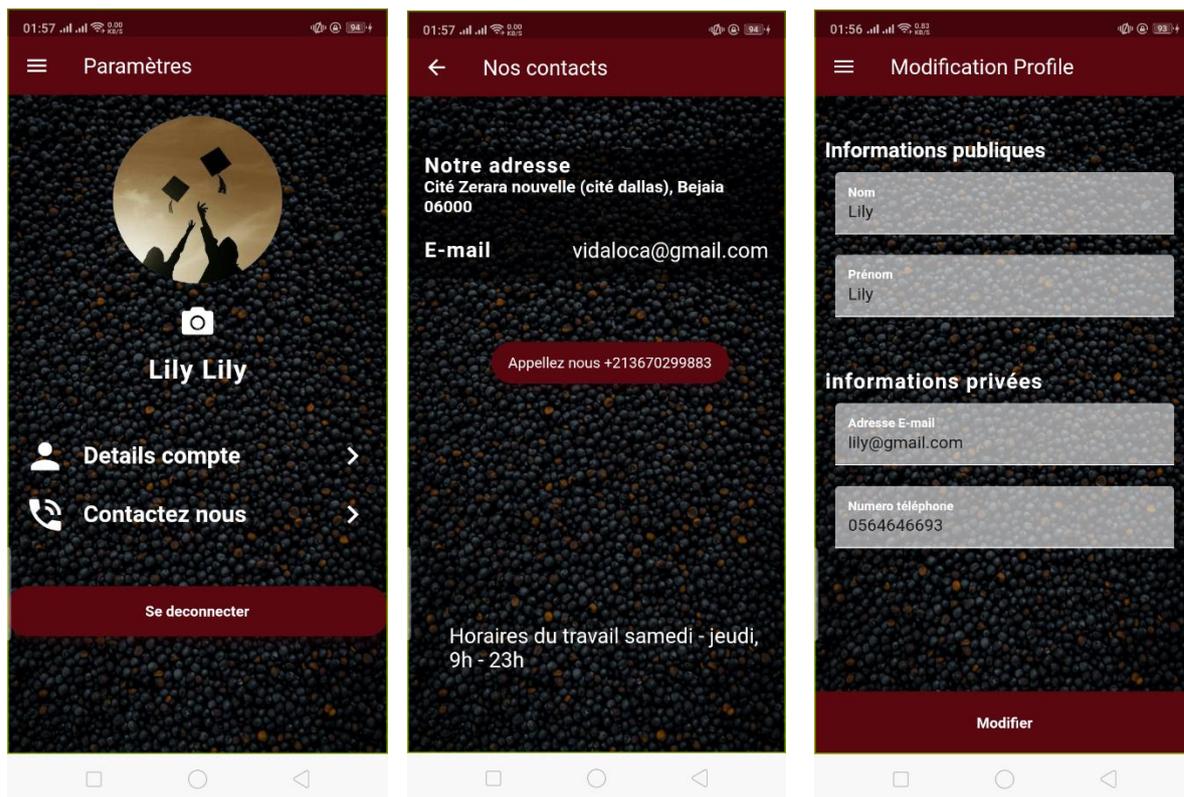


Figure 3 –Interface paramètre.