الجمـــهوريـــة الجـــزائريـــة الديـمقـراطيـــــــة الشـــعبيـــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـلـيـم الـعـالـــي و الـبـحـث الـعـلـمـي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES
قسم الإعلام الآلي
DEPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ÉTUDE
(Licence)

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : Systèmes Informatiques

THEME

Conception Et Réalisation D'une Application Android Pour La Commande Et La Livraison De Nourriture

---

Par:
Touaitia Ali
Babaghayou Abdelkader Mihielddine
Bederina Hadj Mahmoud

Proposé par : Mr. Allaoui Tahar

Année Universitaire 2020/2021

Remerciement

Tout d'abord, nous remercions Allah, notre créateur, pour nous avoir donné la force, la volonté et le courage d'accomplir ce modeste travail.
Nous remercions notre encadreur Monsieur Allaoui ansi que Madame Kerrouche pour leurs conseils et leur encouragement du début jusqu'à la fin de ce travail.
Nos remerciements vont aussi à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.
Enfin, nous tenons à exprimer notre profonde gratitude à nos familles qui nous ont toujours soutenus et à tous ceux qui ont participé à la réalisation de ce mémoire.
Ainsi qu'à tous les enseignants qui ont contribué à notre formation.

Dédicaces

Nous dédions ce modeste travail à nos parents bien-aimés qui nous ont soutenus tout au long de notre cursus universitaire et le sont toujours.
À nos amis qui n'ont jamais hésité à nous aider de quelque manière que ce soit.
Et à tous ceux qui nous ont enseigné tout au long de notre vie scolaire.
A nos camarades de la 3ème année LMD.
Merci à tous.

Résumé

في العقد الماضي ، شهد العالم ثورة في خدمة توصيل الطعام، ومع ظهور تطبيقات الهاتف المحمول ، أصبح من الضروري أن يكون لديك تطبيق جوال كمطعم أو كمؤسسة توصيل. للأسف ، لا يمكن أن يكون لكل مؤسسة تطبيق جوال خاص بها بسبب التكاليف. لذلك قمنا بتصميم تطبيق جوال كمنصة تجمع العملاء والمطاعم ومؤسسات التوصيل في مكان واحد. التطبيق مخصص لنظام Android ، مصنوع من Flutter و Firebase.
التطبيق سهل الاستخدام وعملي لكل نوع من المستخدمين.

In the last decade, the world has experienced a revolution in the food delivery industry, and with the arrival of mobile applications, it has become a necessity to have a mobile app as a restaurant or as a delivery establishment. Sadly not every establishment can have its own mobile application due to the costs. So we have made a mobile application as a platform that groups the clients, the restaurants and the delivery establishments in one place. The application is for Android, made with Flutter and Firebase.
The application is easy to use and practical for each type of user.

Au cours de la dernière décennie, le monde a connu une révolution dans le secteur de la livraison de nourriture, et avec l'arrivée des applications mobiles, il est devenu nécessaire d'avoir une application mobile en tant que restaurant ou établissement de livraison. Malheureusement, tous les établissements ne peuvent pas avoir leur propre application mobile en raison des coûts. Nous avons donc créé une application mobile comme une plateforme qui regroupe les clients, les restaurants et les établissements de livraison en un seul endroit. L'application est pour Android, réalisée avec Flutter et Firebase.
L'application est facile à utiliser et pratique pour chaque type d'utilisateur.

Table des matières

Table des figures

Liste des tableaux

Liste des Abréviations

— **MVC** : Model View Controller
— **IHM** : Interface Homme Machine
— **SDK** : Software Development Kit
— **UML** : Unified Modeling Language
— **NoSQL** : Not only Structured Query Language
— **JSON** : Java Script Object Notation
— **API** : Application Programming Interface

Introduction

Au cours de la dernière décennie, nous avons assisté à une révolution dans les commerces en ligne. Les gens se sont habitués aux achats en ligne et aux réservations à distance comme la réservation de billets, l'achat d'articles et même la commande de nourriture, principalement parce que c'est plus rapide, sans effort et moins compliqué.

Lorsque quelqu'un veut acheter des repas en ligne, il doit chercher un restaurant qui prépare les repas désirés. La commande doit lui être livrée soit par le livreur du restaurant, soit par un établissement de livraison séparé.

Dans les pays étrangers, en Europe ou aux États-Unis, beaucoup de restaurants et de sociétés de livraison ont leur propre application mobile. Malheureusement, ce n'est pas le cas pour tous les établissements qui peuvent se permettre de créer leur propre application en raison des coûts élevés de développement d'une application mobile.

Delà, nous avons pensé à contribuer à une solution. C'est une plateforme conçue comme une application mobile qui regroupe les clients, les restaurants et les établissements de livraison en un seul endroit. Nous l'avons appelé *Foudi*.

Elle répond à tous les besoins des personnes qui veulent commander leurs plats à distance, elle permettra également aux propriétaires de restaurants et aux établissements de livraison d'augmenter leurs ventes et d'accélérer leur flux de travail avec les clients à distance.

Ce travail est organisé comme suit :

— Le premier chapitre introduit les différents outils de travail nécessaires pour concevoir cette application.
— Dans le deuxième chapitre, nous citons les différentes étapes suivies pour accomplir la conception de cette application on utilisant les diagrammes UML ainsi l'identification des acteurs principaux qui jouent des rôles dans notre application.
— Le troisième chapitre nous présentons les différentes IHMs de flux de l'application conçue.
— En dernier, nous terminons ce travail par une conclusion générale suivie de quelques perspectives pour la continuité de ce projet.

Chapitre 1

Receuil des besoins

Dans ce chapitre, nous proposons de regrouper les besoins techniques necessaires pour accomplir ce travail.

1.1 Développement Android

Nous avons choisi de réaliser une application mobile dédiée aux utilisateurs d'Android. Il existe beaucoup de choix quant aux technologies que nous allons utiliser pour ce développement.

1.1.1 Les technologies utilisées

Dans ce travail nous avons utilisé le framework Flutter pour les interfaces et la logique de l'application et Cloud Firestore de Firebase pour le stockage et la gestion de la base de données.

Flutter

C'est un SDK d'interface utilisateur open-source créé par Google [1]. Il est utilisé pour développer des applications multiplateformes pour Android, iOS, Linux, Mac, Windows et le Web à partir d'une base de code unique.
Nous avons utilisé Flutter en raison de sa structure qui favorise la vitesse de développement. Sa structure est simplement une cascade de widgets faciles à personnaliser.

Flutter utilise le langage de programmation Dart, également développé par Google. Il s'agit d'un langage orienté objet, basé sur des classes avec une syntaxe de type de la language C.

Firebase

C'est une plateforme accessible de n'importe quel endroit, qui aide à développer rapidement des applications de qualité [2]. Elle dispose de nombreuses fonctionnalités, mais nous l'utilisons principalement pour son service de stockage au Cloud (Cloud Firestore).

Cloud Firestore est une base de données documentaire NoSQL (un ensemble de continuation en cascade collection-document) qui nous permet de stocker, de synchroniser et d'interroger facilement des données pour des applications mobiles et web à l'échelle mondiale [3].

1.2 Méthode de travail collaborative

L'application conçue dans ce travail, a été développée à partir de zéro par trois développeurs. Nous avons travailler en parallèle sur le même projet, de ce fait nous, les trois développeurs, allons donc utiliser Git et Github.

Git

C'est un logiciel permettant de suivre les modifications apportées à un ensemble de fichiers [4]. Il est généralement utilisé pour coordonner le travail des programmeurs qui collaborent à l'élaboration du code source lors du développement de logiciels. Ses objectifs sont la rapidité, l'intégrité des données et la prise en charge des flux de travail distribués et non linéaires.

Github

C'est un service d'hébergement de référentiel Git, mais il ajoute de nombreuses fonctionnalités qui lui sont propres [5]. Il fournit une interface graphique basée sur le Web. Il fournit également un contrôle d'accès et plusieurs fonctions de collaboration.

1.3 Architecture du code

Nous avons utilisé l'architecture de projet MVC [6]. Il s'agit d'un modèle de conception de logiciel couramment utilisé pour développer des interfaces utilisateur qui divisent la logique du programme en trois éléments interconnectés.

Cela permet de séparer les représentations internes de l'information de la manière dont l'information est présentée à l'utilisateur et acceptée par celui-ci.

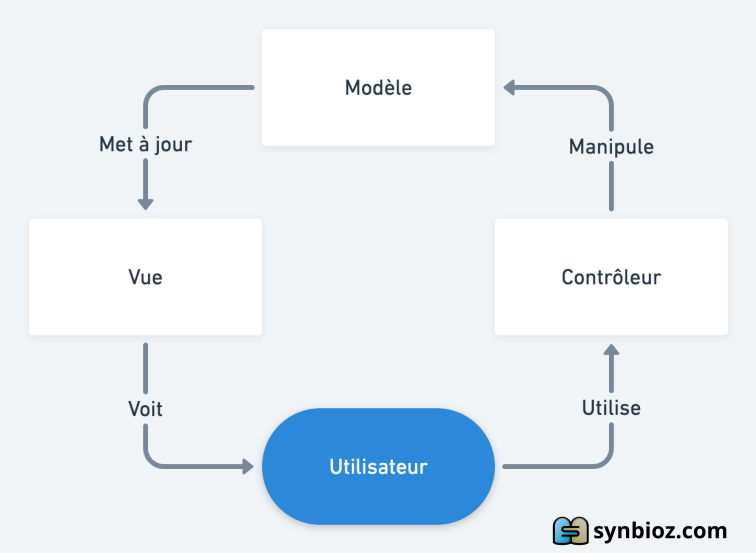


FIGURE 1.1 – Le flux MVC

Conclusion

En conclusion, dans ce chapitre nous avons résumé les besoins techniques que nous avons choisis et utilisés pour lancer ce travail. Suivant les choix faits, on va structuré la conception de l'application ceci dans le chapitre suivant.

Chapitre 2

Conception de l'application

Dans ce chapitre nous présentons les differentes étapes de la conception de l'application de gestion de commande et livraison de la restauration.

2.1 Définitions

Nous allons citer les définitions nécessaires pour comprendre les tremes de l'application.

2.1.1 Termes principaux

Les deux établissements nécessaires dans la réalisation d'application sont :

— Restaurant : Toute entité qui prépare des repas.

— Etablissements de livraison (Delivery company) : Toute entité qui livre des repas.

Il y a trois acteurs (utilisateurs) principaux qui font tous les actions humaines dans l'application :

— Client : Toute personne qui veut commander de la nourriture à distance.

— Serveur ou employé de restaurant (Server) : Tout travailleur qui représente le restaurant.

— Livreur (Deliverer) : Tout travailleur dans un établissements de livraison qui livre les repas.

2.1.2 Autre termes

— Commande (Order) : Une liste des plats désirés choisis par le client à partir du menu du restaurant.

— Repas (Dish) : Toute nourriture qui peut être préparé par un restaurant et livré par un établissement de livraison.
— Livraison (Delivery) : L'action de livrer les repas du restaurant au client.

2.2 Relations entre les acteurs prencipaux de l'application

Le flux abstrait générale de l'application est comme suit :

1. Le client passe une commande et la paye via l'application (ou cash à la livraison).
2. Le serveur est informé de la commande et commence à la préparer.
3. Une fois la commande prête, le client et le livreur en sont informés.
4. Le livreur va récupérer la commande au restaurant et en informe le client.
5. Le client reçoit sa commande et la marque comme complète dans l'application.

2.3 Modélisation

On a choisi trois diagrammes d'UML pour modéliser l'application :
— Diagramme de classes
— Diagrammes de cas d'utilisation
— Diagrammes de séquence

2.3.0.1 Diagramme de classes

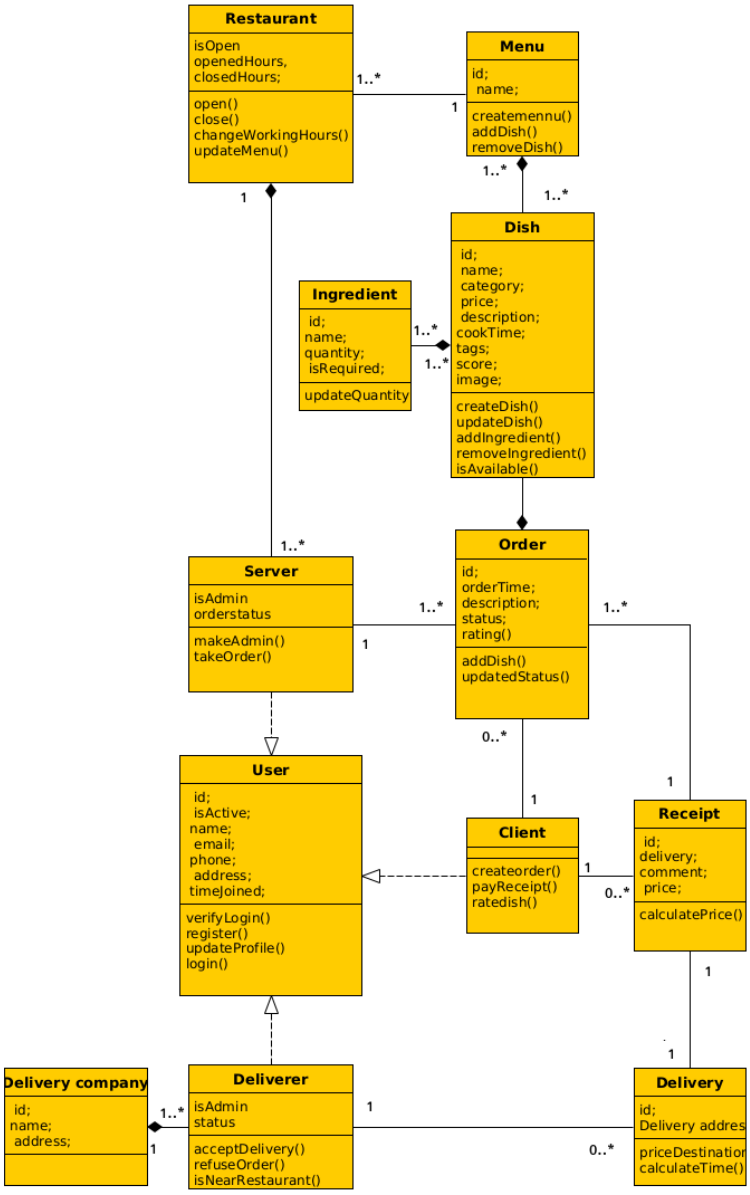


FIGURE 2.1 – Diagramme de classes

La figure (fig 2.1) décrit les classes principales d'objets sur lesquelles repose la logique de l'application, ainsi que les relations qui existent entre elles.

Le tableau (tab 2.1) explique les relations entre les classes dans le schéma de figure 2.1.

Classes	Relations
Delivery company	Elle a un ou plusieurs livreurs
Restaurant	Il a un ou plusieurs serveurs et un menu
User	Le Client et le Serveur et le Livreur héritent de ses attributs et de ses méthodes
Client	Il peut avoir des commandes et des recettes multiples
Server	Il appartient à un restaurant et peut avoir plusieurs commandes
Deliverer	Il appartient à une société de livraison et peut avoir plusieurs livraisons
Menu	Il peut appartenir à plusieurs restaurants et avoir plus d'un plat
Dish	Il appartient à un ou plusieurs menus et comporte plusieurs ingrédients
Ingredient	Il appartient à un ou plusieurs plats
Order	Il appartient à un serveur et à un client, et à un reçu
Delivery	Elle appartient à un livreur et à un reçu
Receipt	Il appartient à un client

TABLE 2.1 – Les relations des classes

2.3.0.2 Diagrammes de cas d'utilisation

Dans cette section, nous présentons les diagrammes de cas d'utilisation. Nous avons réalisé un diagramme pour les principales utilisations qui se déroulent dans l'application.

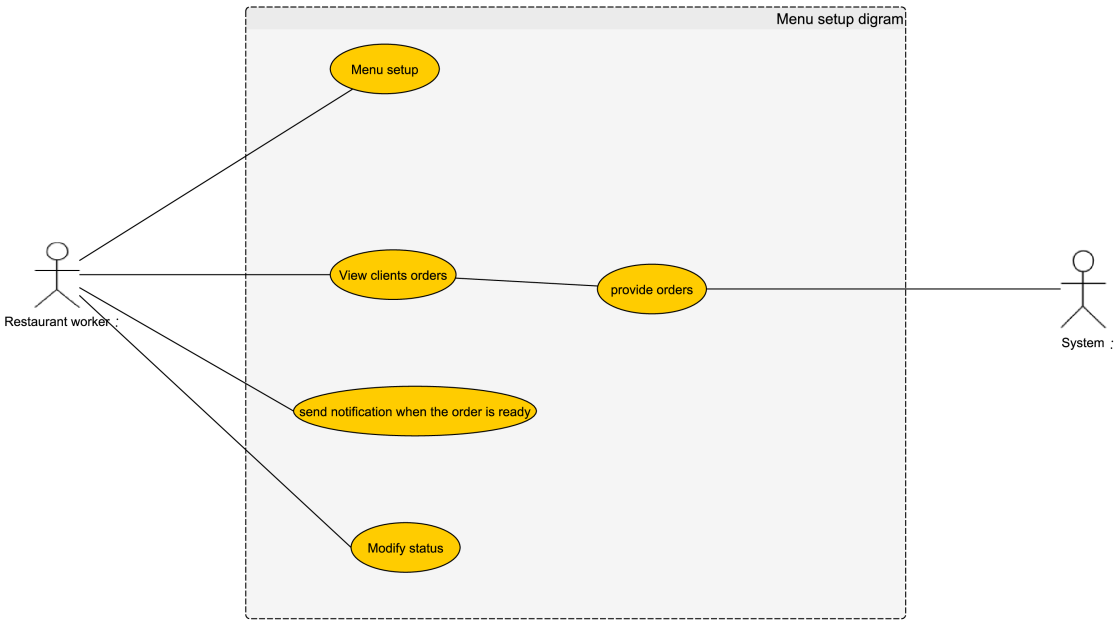


FIGURE 2.2 – Diagramme de cas d'ustilisation du configuration du menu

La figure (fig 2.2) illustre le cas de la configuration du menu du restaurant par l'employé du restaurant. L'employé du restaurant (serveur) a la possibilité de contrôler le menu (ajouter, modifier et supprimer des plats), et recevoir les commandes du client que le système lui fournit, il a aussi le contrôle total de l'état de chaque commande (en préparation, ou prête à être livrée).

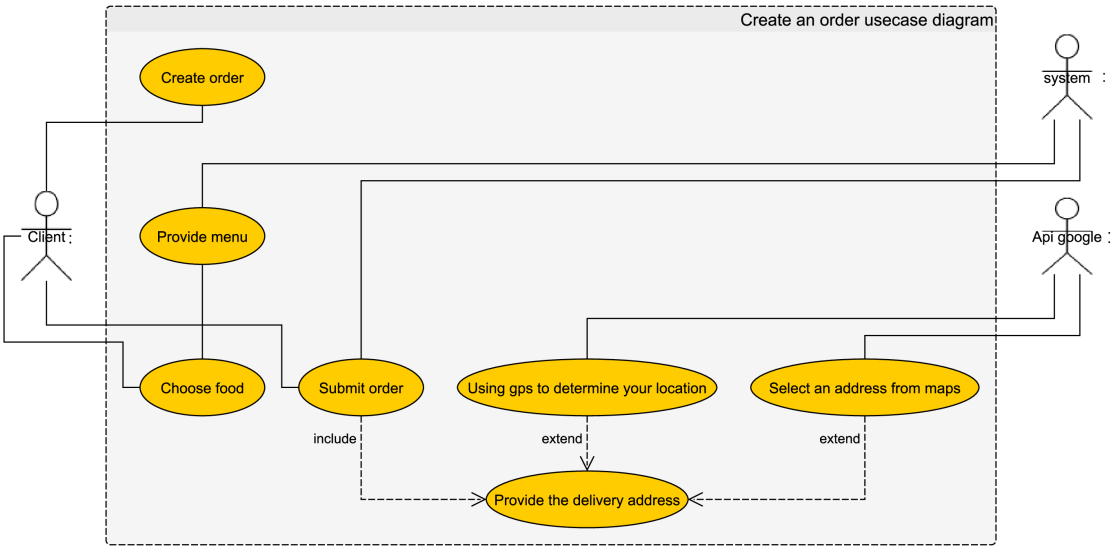


FIGURE 2.3 – Diagramme de cas d'ustilisation de passation d'une commande

La figure (fig 2.3) illustre le cas de la configuration de passation d'une commande par le client de restaurant. Le système met à disposition aux clients des restaurants et leurs menus afin que le client puisse choisir son restaurant, puis choisir ses plats. Il confirme ensuite la commande en indiquant l'adresse de livraison et les informations nécessaires. La commande est soumise au système pour être traitée.

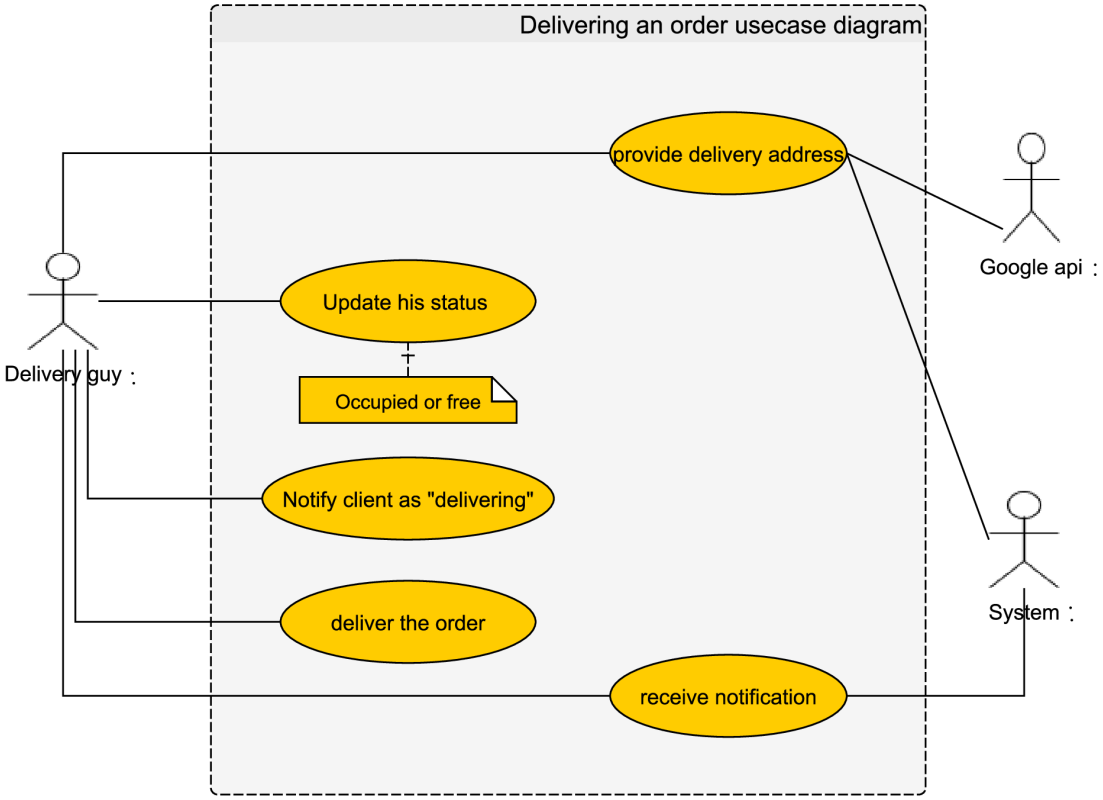


FIGURE 2.4 – Diagramme de cas d'ustilisation du livraison d'une commande

La figure (fig 2.4) illustre le cas de la livraison d'une commande par le livreur. Les livreurs ont la possibilité de définir leur statut "occupé" ou "libre". Lorsqu'un livreur proche accepte une livraison de sa liste de livraisons proches (commandes prêtes à être livrées), il a alors accès à toutes les informations nécessaires de l'adresse du restaurant et de l'adresse du client. Chaque livreur a la possibilité d'informer le client de l'état de livraison de la commande.

2.3.0.3 Diagrammes de séquences

Dans cette section, nous allons illustrer les séquences d'interactions entre les utilisateurs et le système à l'aide des diagrammes de séquence d'UML.

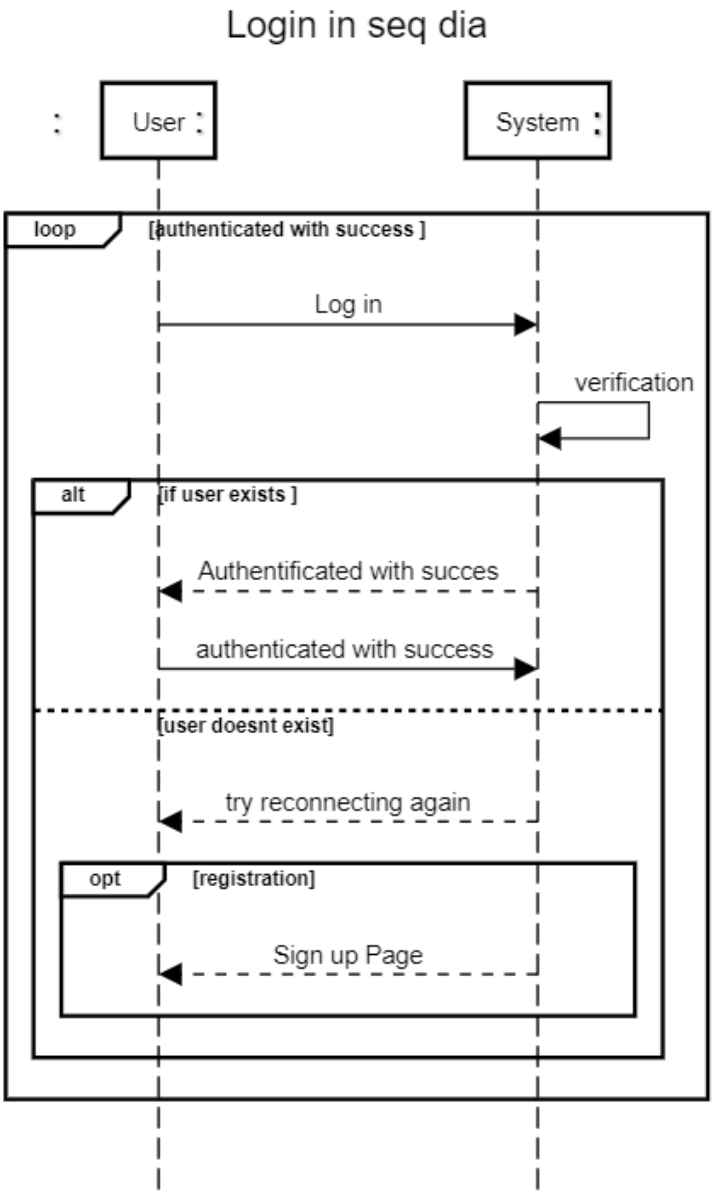


FIGURE 2.5 – Diagramme de séquences du authentification

La figure (fig 2.5) décrit la séquence de connexion au système par tout utilisateur. L'utilisateur saisit ses informations d'identification, le système vérifie si l'utilisateur existe. S'il existe, le système authentifie sa connexion avec

succès, sinon il donne à l'utilisateur la possibilité de s'inscrire ou de saisir à nouveau ses informations d'identification. L'ensemble du processus se répète jusqu'à ce que l'utilisateur se connecte avec succès.

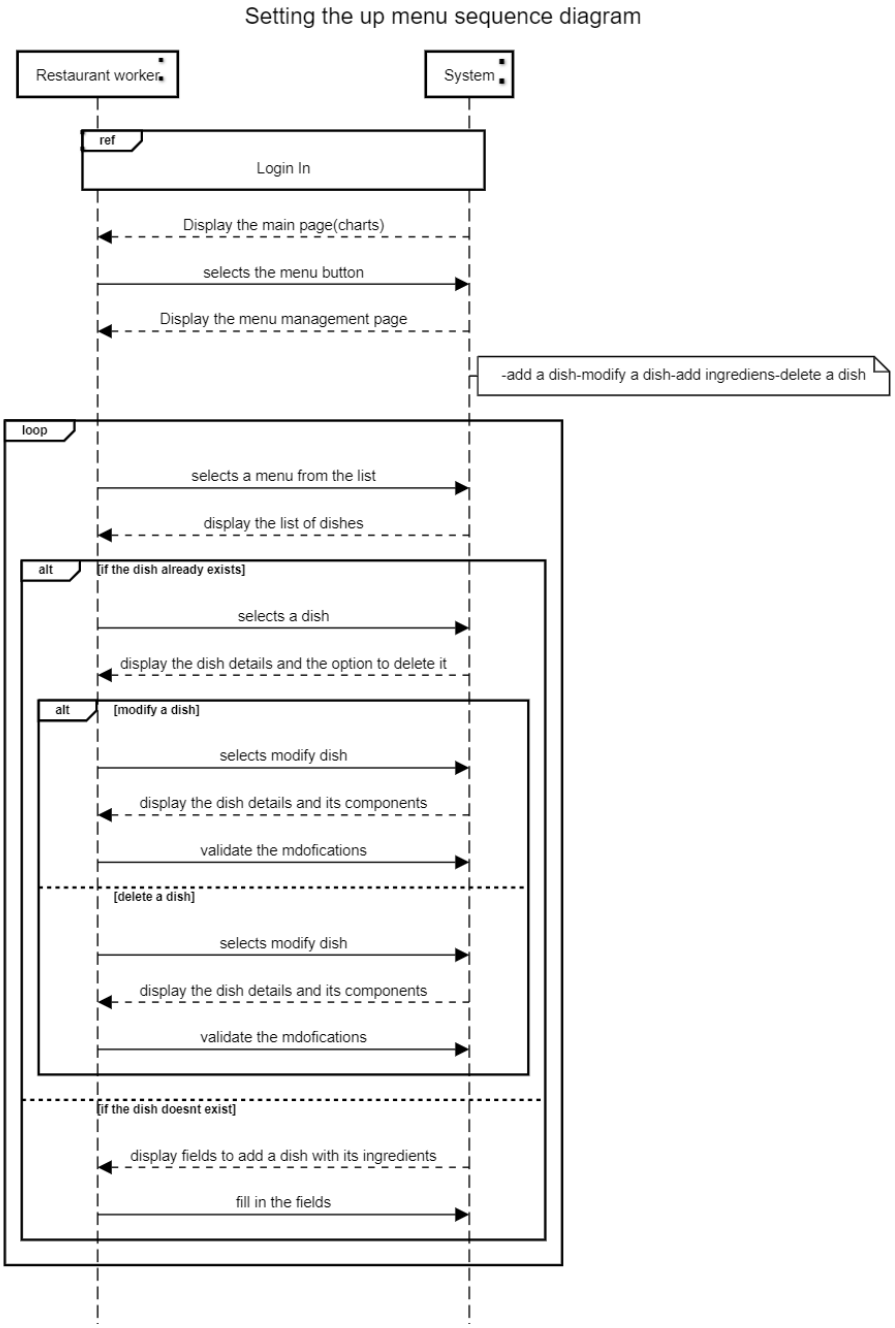


FIGURE 2.6 – Diagramme de séquences du configuration du menu

La figure (fig 2.6) décrit la séquence de configuration de menu du restaurant. Après que l'employé du restaurant se soit connecté avec succès, il peut choisir d'accéder à la page de gestion du menu dans laquelle il peut sélectionner un plat dans une liste de plats s'ils existent déjà dans le menu pour les

supprimer ou les modifier. S'il choisit de les modifier, le système lui montrera tous les détails du plat à modifier, puis après la modification, il pourra valider ses changements. Si le menu est vide, le système fournira un formulaire pour remplir toutes les informations sur le plat avec ses ingrédients.

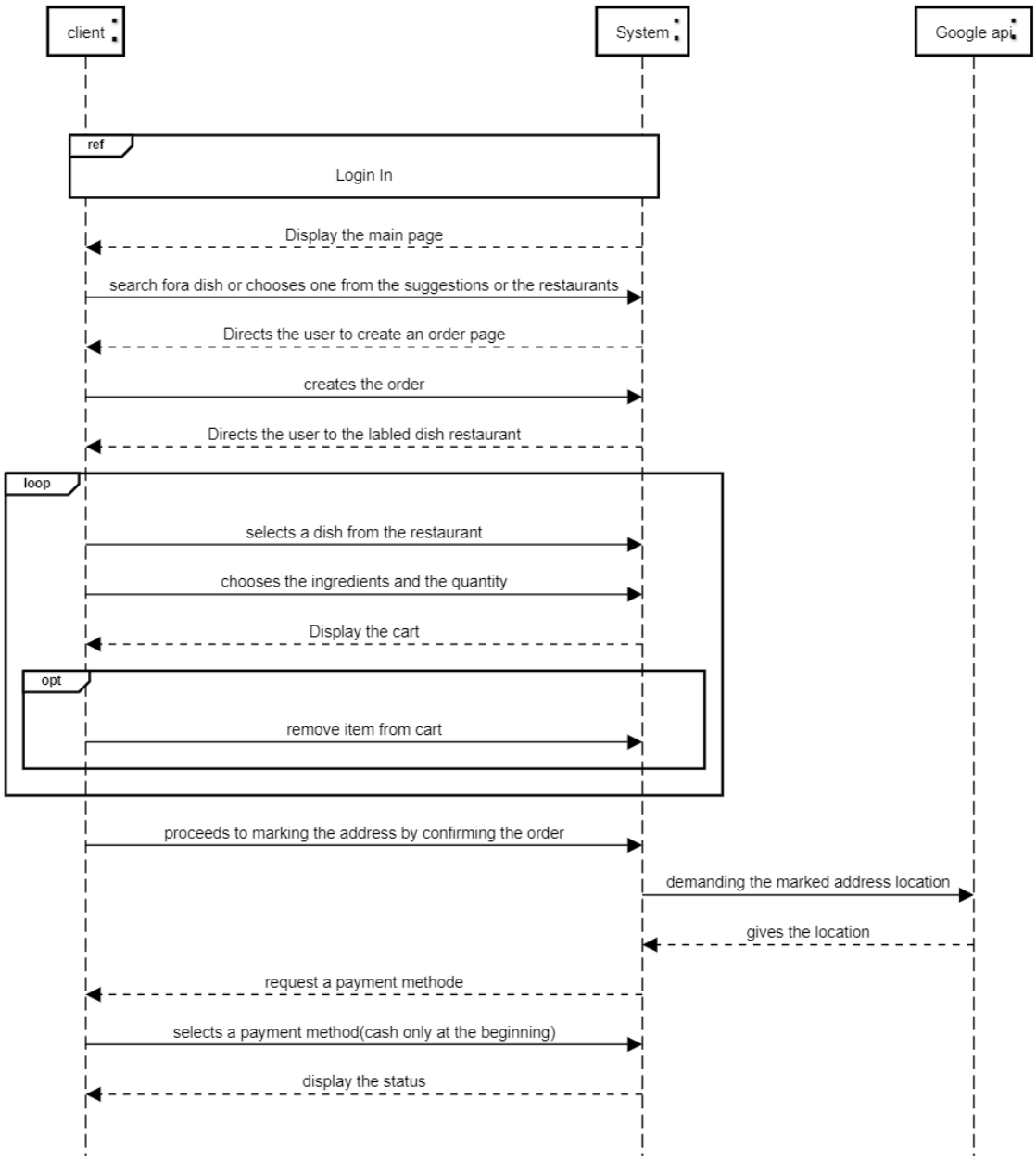


FIGURE 2.7 – Diagramme de séquences de passation d'une commande

La figure (fig 2.7) décrit la séquence de passation d'une commande par le client. Une fois que le client a réussi à se connecter, le système lui montre la page principale.
Le client peut rechercher un plat spécifique ou sélectionner un restaurant,

puis sélectionner le plat, le système affichera le plat dans lequel le client peut sélectionner les ingrédients souhaités, puis ajouter le plat au panier.
Il peut répéter ce processus jusqu'à ce que tous les plats qu'il souhaite soient ajoutés au panier (il a également la possibilité de retirer un plat du panier).

Le client peut ensuite procéder au marquage de l'adresse de livraison sur la carte, le système obtiendra les détails de l'emplacement à partir de l'API Google Maps.
Le système demandera une option de paiement, et après validation, le système mettra à jour l'état de la commande en la mettant en "attente" jusqu'à ce que l'employé du restaurant commence à la préparer.

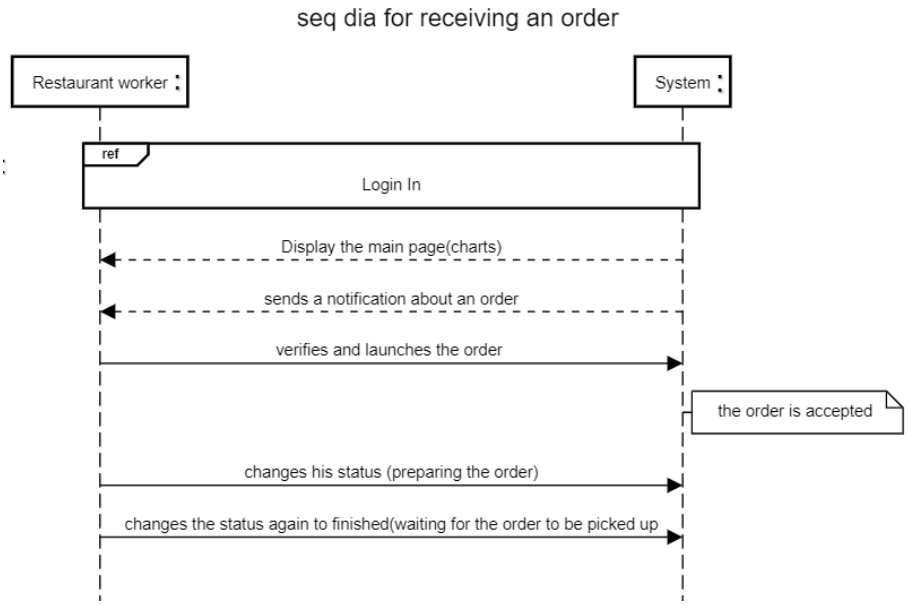


FIGURE 2.8 – Diagramme de séquences du recepetion d'une commande

La figure (fig 2.8) décrit la séquence du recepetion d'une commande par l'employé du restaurant (serveur). Lorsqu'une commande est passée, l'employé du restaurant est notifié par le système. Le serveur valide la commande et commence à la préparer. Le système notifie alors au client le changement de statut de la commande, qui passe de "en attente" à "en préparation". Une fois la commande prête, le serveur met à jour le statut de la commande en "prêt à livrer".

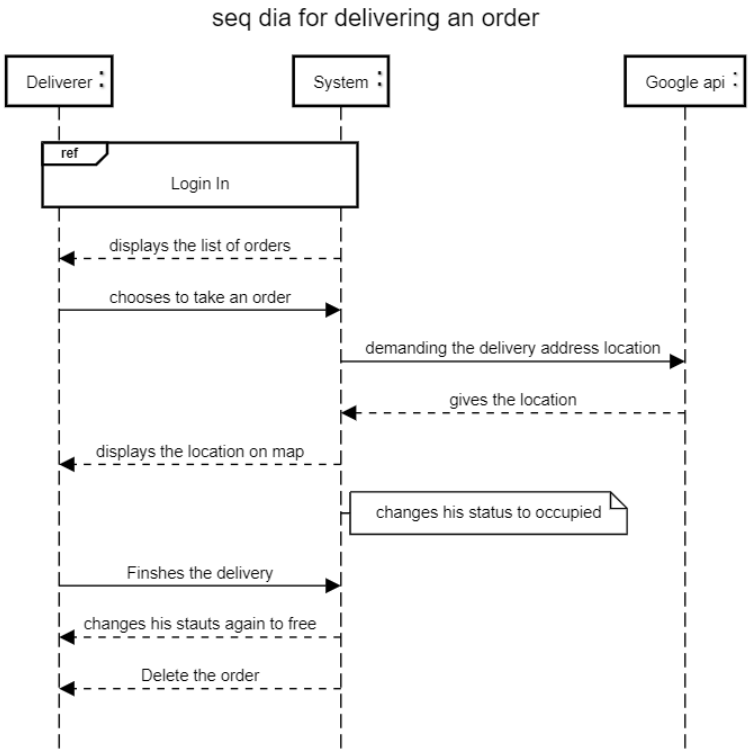


FIGURE 2.9 – Diagramme de séquences du livraison d'une commande

La figure (fig 2.9) décrit la séquence du livraison d'une commande. Si le statut d'une commande est défini comme "prêt à être livré", le livreur en est informé par le système. Lorsqu'il décide d'accepter la livraison, le système lui envoie toutes les informations nécessaires sur les lieux de départ et d'arrivée. Le livreur peut changer le statut de la commande en "livraison" une fois qu'il l'a récupérée au restaurant. Après la livraison, il peut changer son statut de "occupé" à "libre".

2.4 La structure du code

2.4.1 Modéles

Nous avons représenté chaque classe du diagramme de classes (Fig. 2.1) comme des classes Dart pour les traiter comme des objets. Chacune de ces classes possède deux méthodes permettant de convertir de et vers des objets JSON pour faciliter la communication avec la base de données.

2.4.2 Contrôleurs

Ils représentent la fonctionnalité principale de chaque action dans l'application, de la connexion à la mise à jour des commandes et tout ce qui est similaire.

2.4.3 Vues

Chaque écran qui apparaît à l'utilisateur est considéré comme une vue. Les vues contiennent plusieurs widgets en cascade.

2.4.4 Widgets

Nous avons dû créer des widgets personnalisés pour répondre à nos besoins, comme des boutons, des vues de liste, des vues de texte, etc.

Conclusion

Dans ce chapitre nous avons présenté la structure du code en général et les diagrammes du langage UML, ainsi que les définitions des termes essentiels de l'application. Le deuxième chapitre nous a permis d'organiser le corps de l'application pour faciliter le processus de mise on oeuvre décrit dans le chapitre suivant.

Chapitre 3

Réalisation de l'application

Dans ce chapitre nous allons présenter les differentes IHMs de flux de cette application.

3.1 Lancement d'application

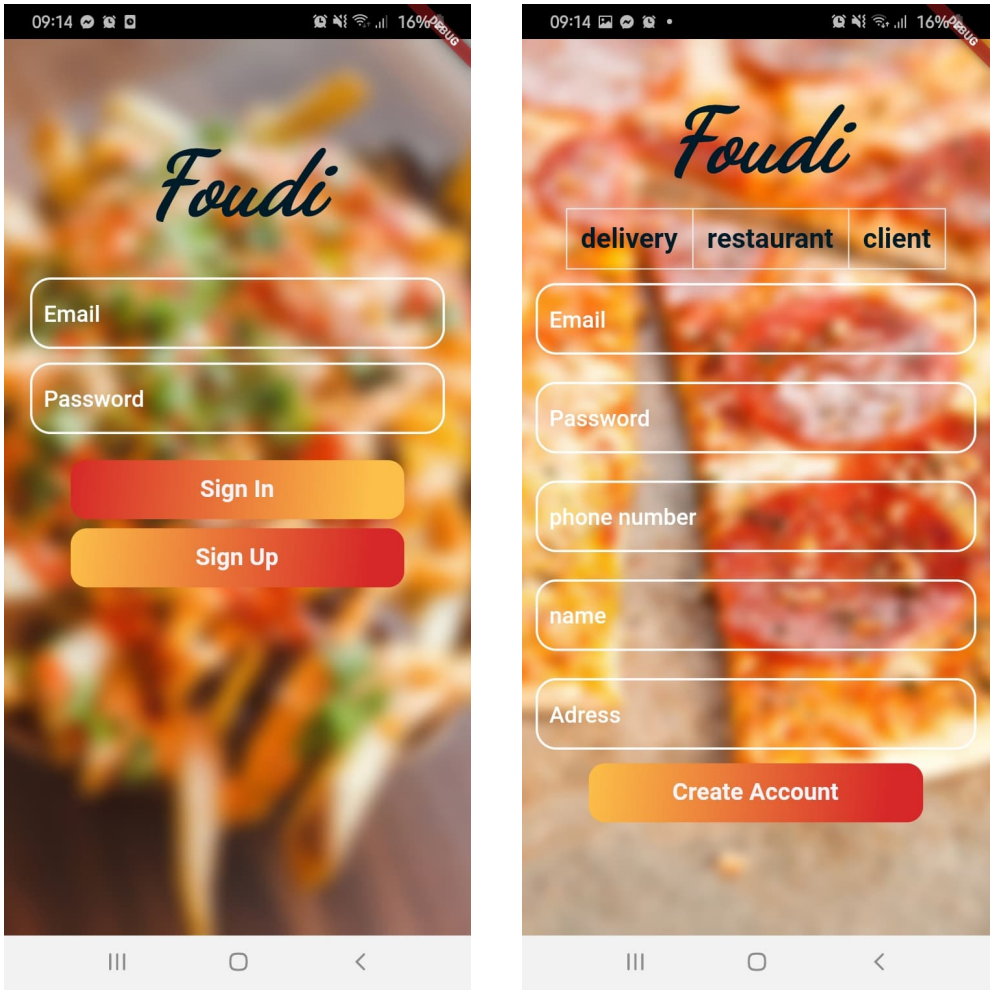
Lorsque l'application est lancée pour la première fois, elle affiche la page de connexion (fig. 3.1a). Lorsqu'un nouvel utilisateur souhaite accéder à l'application, il doit d'abord s'inscrire en cliquant sur le bouton "Sign up".

3.1.0.1 L'inscription

Comme la figure (fig 3.1b) le montre, le nouvel utilisateur doit choisir le type de compte (Client, Restaurant, ou Delivery). Il remplit ensuite les informations nécessaires et appuie sur "Create Account". Le système vérifiera si l'utilisateur n'existe pas déjà et validera ensuite son compte.

3.1.0.2 La connexion (Login)

Si l'utilisateur possède déjà un compte, il peut saisir son e-mail et son mot de passe (comme en fig. 3.1a). Le système détectera automatiquement le type d'utilisateur et lui montrera sa page d'accueil qui lui correspond.



(a) L'IHM de la page de connexion (b) L'IHM de la page d'inscription

FIGURE 3.1 – Les IHM des premières pages au lancement

3.2 Cas d'utilisateur client

Le client dispose de trois pages dans le menu de l'application :

— **Page d'accueil** : Il consiste de deux listes du suggestions : des restaurant et des plats.
— **Page de chariot** : Pour voir les plats commandés.
— **Page de cart** : Pour parcourir les restaurants à partir de la carte. Ou voir l'emplacement actuel de lui-même.
— **Page de profile** : Pour voir ses données personnelles.

3.2.1 Lancement d'un commande

Lorsque le client souhaite passer une commande, il peut se rendre sur la page d'accueil (fig. 3.2) où il peut voir les deux listes de restaurants et de plats disponibles suggérés par l'application (les suggestions sont basées sur la proximité et les commandes précédentes, le cas échéant).

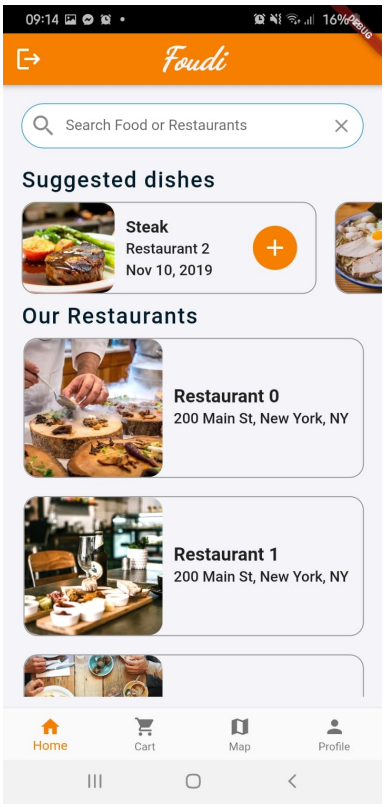


FIGURE 3.2 – L'IHM de la page d'accueil du client

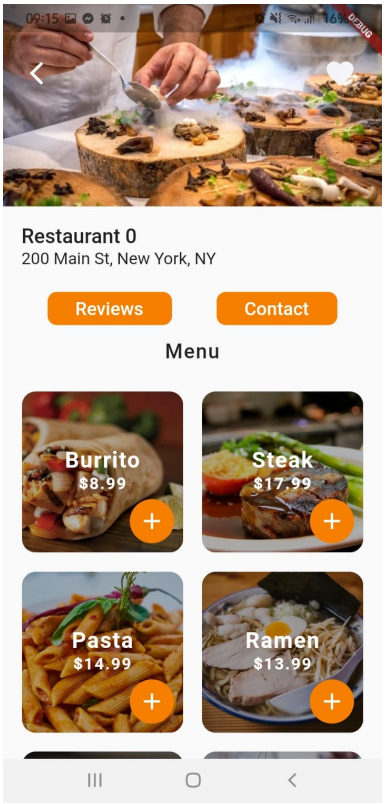


FIGURE 3.3 – L'IHM de la page du restaurant

S'il appuie sur la page d'un restaurant, celle-ci affichera les informations nécessaires (nom, adresse, avis et coordonnées) ainsi que le menu du restaurant (fig. 3.3).

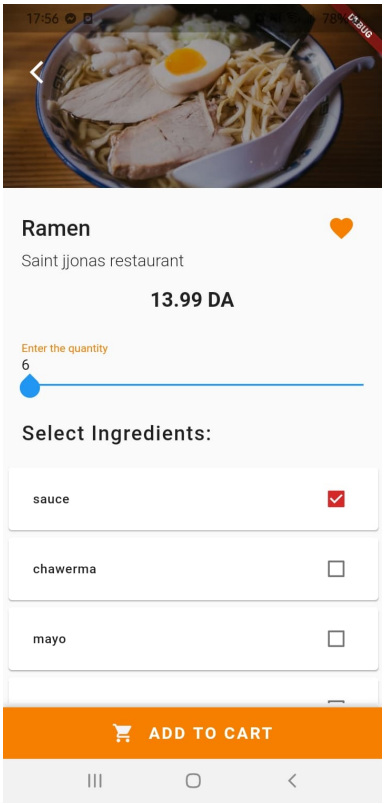


FIGURE 3.4 – L'IHM de la page du plat

En sélectionnant le plat souhaité, une page apparaît avec tous les détails du plat (fig. 3.4), où le client peut sélectionner les ingrédients. Il a également la possibilité de sélectionner la quantité du plat souhaité. Et quand il a terminé de specifier ses preferences, il va l'ajouter au panier.

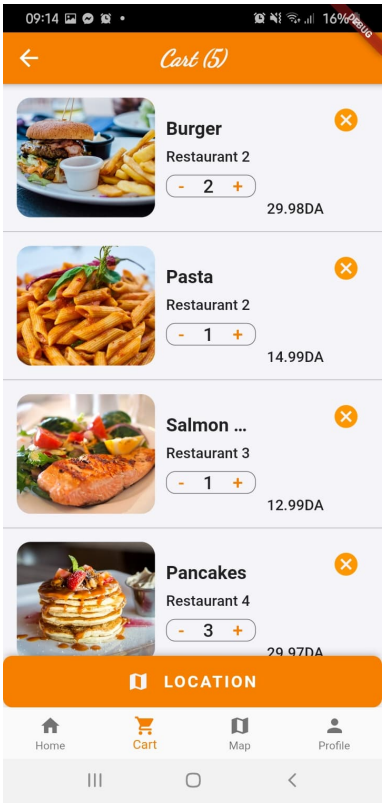


FIGURE 3.5 – L'IHM de la page du panier

Il peut passer à la page du panier lorsqu'il a terminé le processus de commande pour vérifier sa commande et son prix (fig. 3.5).

FIGURE 3.6 – L'IHM de la page du carte

En appuyant sur "Commander" dans la page du panier (fig. 3.5), l'application le redirige vers la carte (fig. 3.6) où il peut sélectionner l'adresse de livraison en faisant glisser le marqueur sur la carte. Ou sélectionner sa position actuelle en appuyant sur l'icône du GPS dans le coin supérieur droit (l'API Google Maps donne au système les coordonnées GPS actuelles de l'appareil mobile).

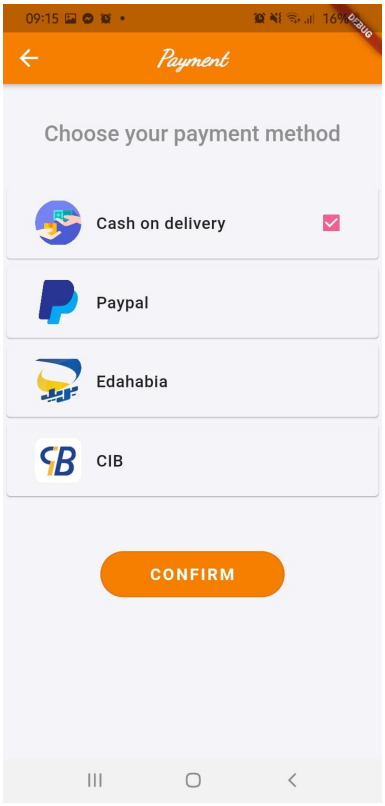


FIGURE 3.7 – L'IHM de la page du paiement

Une fois que le client a confirmé l'adresse de livraison, il est redirigé vers la page de paiement où il peut choisir une option de paiement (fig. 3.7).

Si le paiement est validé par le système, une page de statut apparaîtra (fig), pour indiquer le statut actuel de la commande. Les statuts sont les suivants :

— Acceptée (fig 3.8) : le restaurateur accepte la commande.
— En cours de préparation (fig 3.9) : la nourriture est en cours de préparation
— Prêt (fig 3.10) : la commande est prête à être livrée.

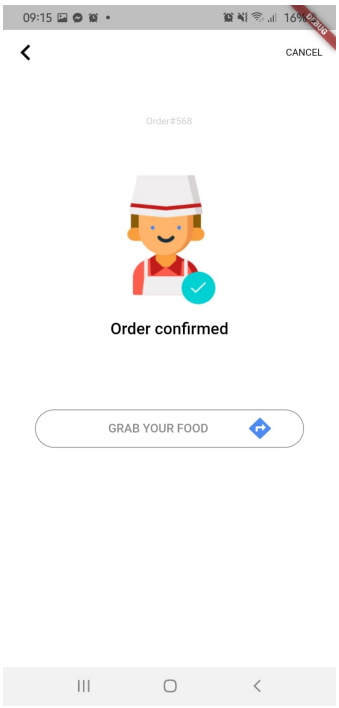


FIGURE 3.8 – L'IHM de la page du status "accepté"

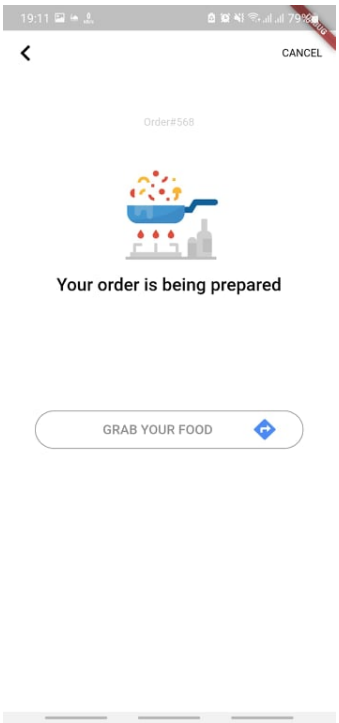


FIGURE 3.9 – L'IHM de la page du status "en cours de préparation"

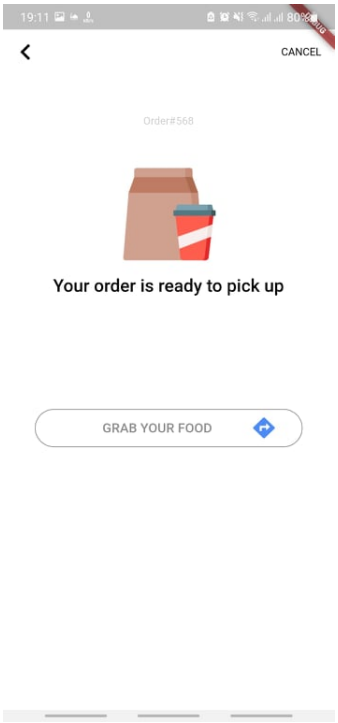


FIGURE 3.10 – L'IHM de la page du status "prêt"

3.3 Cas d'utilisateur server

Le server (employé du restaurant) dispose de trois pages dans le menu de l'application :

- **Page des commandes (page d'accueil)** : Une liste de toutes les commandes passées
- **Page de configuration du menu** : Pour modifier le menu du restaurant
- **Page de profile** : Pour voir les avis des client vis à vis de son restaurant.

3.3.1 Réception des commandes

FIGURE 3.11 – L'IHM de la page des commandes

Le système fournit au travailleur du restaurant les commandes des clients (fig 3.11), chaque commande contient (lorsqu'il est pressé) les informations nécessaires (un identifiant et une heure de commande, des plats avec les ingrédients souhaités, des notes si disponibles...etc) comme c'est indiqué dans la figure (fig 3.12).

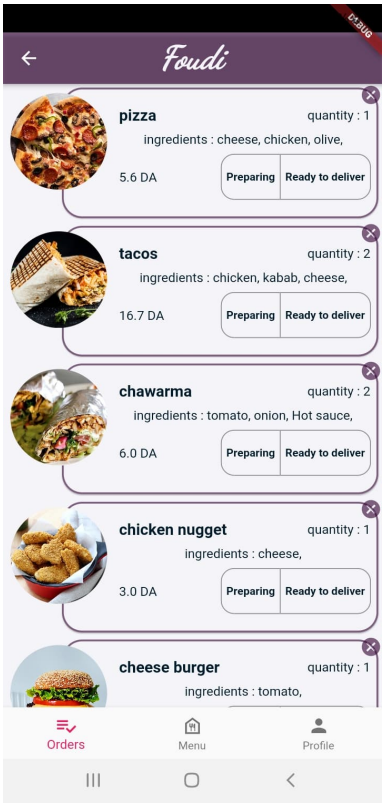
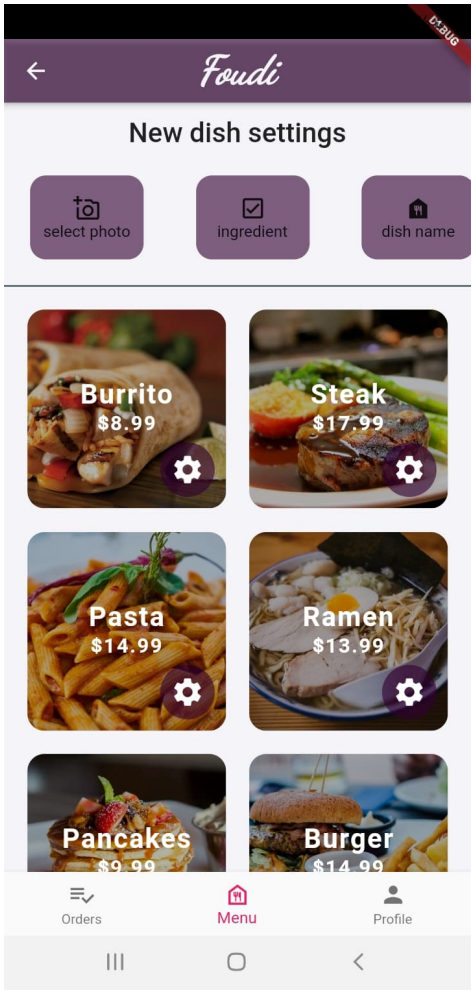


FIGURE 3.12 – L'IHM de la page plats dans un commande

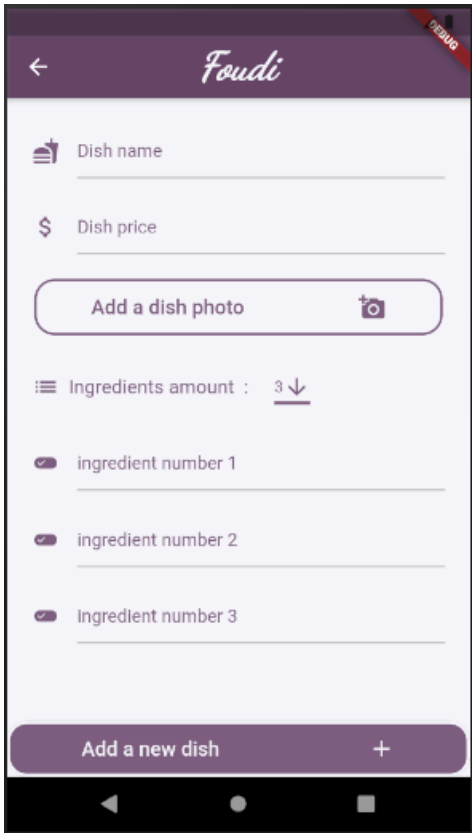
Lorsque la commande commence à être préparée, le serveur met l'état de la commande en préparation pour en informer le client. Après avoir terminé, il met l'état de la commande sur prêt à livrer pour que le livreur soit notifié.

3.3.2 La configuration du menu

L'employé du restaurant a la possibilité d'ajouter et de supprimer des plats du menu du restaurant (fig 3.13a), ainsi que la possibilité de modifier les ingrédients en les activant ou en les désactivant simplement en cliquant sur un simple bouton à bascule (fig 3.13b).



(a) L'IHM du menu



(b) L'IHM de la page de modification d'un plat

FIGURE 3.13 – Le IHMs de configuration de menu

3.3.3 Page de profile

Lorsque le restaurant fait l'objet d'une critique de la part de ses clients, lorsque l'employé du restaurant consulte son profil, il peut voir toutes les critiques et tous les commentaires (fig 3.14).

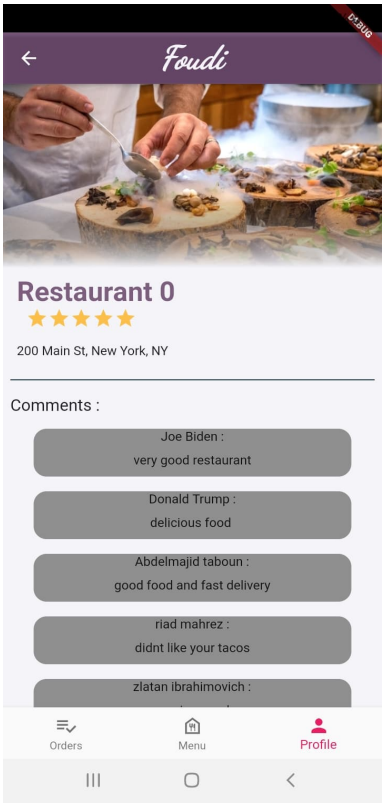


FIGURE 3.14 – L'IHM de la page de profile de restaurant

3.4 Cas d'utilisateur livreur

Le livreur dispose de trois pages dans le menu de l'application :
— **Page d'accueil** : Affiche les livraisons à proximité
— **Page des statistiques** : Montre quelques chiffres sur toutes ses livraisons et la progression des livraisons dans le temps,
— **Page de profile** : Pour voir son rating, et changer les détails de son compte.

3.4.1 Les livraisons

La page d'accueil du livreur (fig 3.15) est constituée des commandes proches prêtes à être livrées ; la liste des commandes contient les informations essentielles de chaque commande qui permettent à l'utilisateur de la choisir ou de l'ignorer.

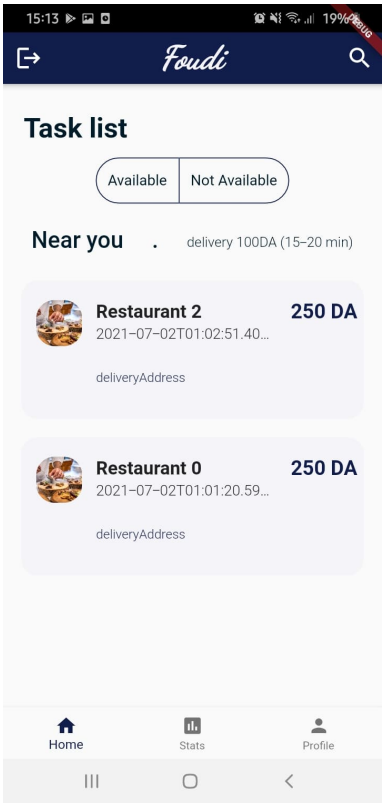
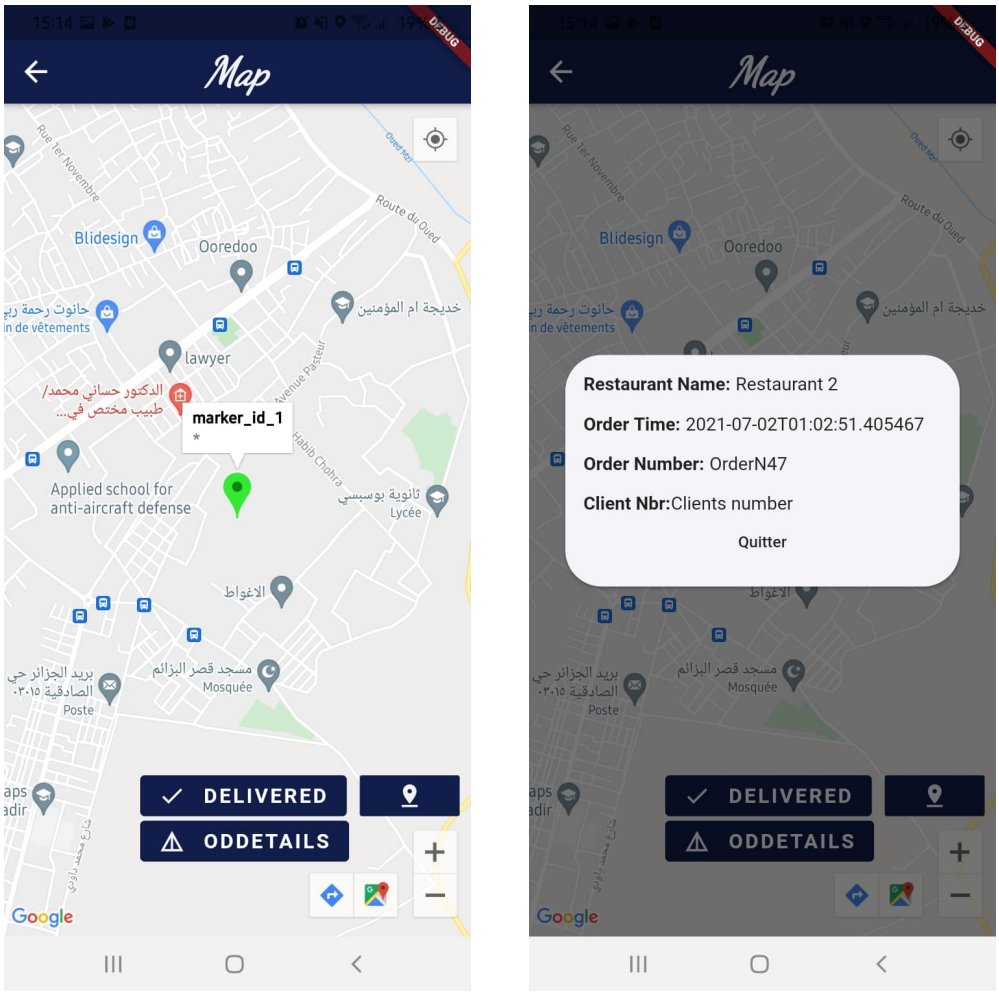


FIGURE 3.15 – L'IHM de la page d'accueil du livreur

Lorsqu'il sélectionne une commande à livrer, l'application affiche la carte avec les emplacements du restaurant et du client (fig 3.16a). Le livreur peut voir des informations supplémentaires sur la livraison en appuyant sur le bouton "Informations sur la commande" (fig 3.16b). Une fois qu'il a livré la commande, il peut appuyer sur le bouton "livré" indiqué sur la carte.



(a) La position du client (b) Les informations du livraison

FIGURE 3.16 – L'IHM de la carte de livreur

Conclusion

Notre application n'est pas destinée à fonctionner avec un seul restaurant ou établissement de livraison, nous avons créé une plateforme qui regroupe tous les différents établissements pour travailler en un seul endroit.

Notre principal objectif en créant l'application n'était pas seulement d'apprendre à la concevoir ou d'acquérir l'expérience du développement d'une grande application mobile, mais de l'utiliser comme notre entrée dans le monde des affaires et de conquérire avec cette application le marché algérien en ligne.

Pour l'instant, l'application fonctionne à petite échelle, nous nous préparions à ajouter de nombreuses caractéristiques et fonctionnalités pratiques, mais en raison du peu de temps qui nous a été accordé, nous n'avons pas atteint le produit souhaité.

Nous allons continuer à développer cette idée et la pousser à l'extrême car nous avons remarqué que nous offrons en général une solution moins chère et plus efficace.

Bibliographie

[1] Flutter
https://flutter.dev/

[2] Firebase
https://firebase.google.com/

[3] Cloud Firestore
https://firebase.google.com/docs/firestore

[4] Git
https://en.wikipedia.org/wiki/Git

[5] Github
https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/

[6] Architecture MVC
https://fr.wikipedia.org/wiki/Modèle-vue-contrôleur