



25 Cr Mgr Roméro,
91000 Évry-Courcouronnes

Alternant
LETTE VOUETO
JIMMY

RESPONSABLE
Pédagogique
Hicham EL
KADIRI

Tuteur de
l'entreprise
NTALA Toutel

RAPPORT DE SOUTENANCE

Développeur React Native

2022/2023

Remerciement

Je tiens à prendre un moment pour exprimer ma profonde gratitude envers mon tuteur à l'occasion de ma soutenance en tant que développeur mobile. Votre rôle en tant que tuteur a été inestimable et a joué un rôle crucial dans ma réussite au cours de cette période de formation.

Votre expertise, vos conseils avisés et votre engagement ont été des facteurs déterminants dans mon apprentissage et mon développement en tant que professionnel du développement mobile. Vos orientations m'ont permis de relever des défis techniques complexes, d'acquérir une compréhension approfondie des meilleures pratiques et de développer une perspective globale sur le monde de la technologie.

Votre disponibilité pour répondre à mes questions, discuter des problèmes et fournir des retours constructifs a été une source de confiance et d'inspiration pour moi. Votre mentorat m'a encouragé à repousser mes limites, à rechercher l'excellence et à viser constamment l'amélioration.

Je tiens également à remercier mon tuteur d'école. Votre approche pédagogique patiente et bienveillante a favorisé mon épanouissement tant sur le plan professionnel que personnel. Vos conseils lors de votre visite m'ont été d'une aide efficace pour la réalisation du PowerPoint.

En ce jour de soutenance, je suis fier de vous présenter mon travail, sachant que c'est grâce à votre soutien que j'ai pu en arriver là. Je reconnais la chance que j'ai eue de vous avoir comme tuteur et je suis reconnaissant pour toutes les compétences et les connaissances que vous avez partagées avec moi.

Je tiens à exprimer ma profonde gratitude envers toute l'équipe pédagogique de l'Université d'Evry, plus particulièrement dans le cadre de ma Licence Professionnelle Métier de l'Informatique et de l'Application Web

Vos encouragements et vos conseils m'ont été d'une valeur inestimable et continueront d'influencer ma carrière pour les années à venir en tant que développeur. Je vous remercie sincèrement pour tout ce que vous avez fait pour moi et pour m'avoir aidé à devenir le développeur que je suis aujourd'hui.

Avec toute ma reconnaissance,

Table des matières

2	INTRODUCTION.....	1
2.1	Contexte.....	1
2.2	Problématique.....	1
2.3	Constatation.....	1
2.4	Critiques	2
3	ABSTRACT	3
3.1	Contextual.....	3
3.2	Core Issue.....	3
3.3	Notable Observation.....	3
3.4	Key Challenges	4
4	INTEGRATION DU DOMAINE D'ETUDE DANS L'ACTIVITE DE L'ENTREPRISE	5
4.1	Identité de l'entreprise :	5
4.2	Informations commerciales :	5
5	ENVIRONNEMENT DE TRAVAIL	6
5.1	Forme d'organisation :	6
5.1.1	Télétravail :	6
5.1.2	Daily meeting/réunion :	6
5.1.3	Code review :	7
5.2	Moyen matériel :	7
5.3	Système d'exploitation :	7
5.4	Hébergement :	7
5.4.1	OVH Cloud :	7
5.5	Environnement de développement :	8
5.5.1	Visual studio code :	8
5.5.2	Expo go :	8
5.5.3	Android studio :	8
5.5.4	Git :	9
5.5.5	Tig :	10
5.6	Framework :	11
5.6.1	React native :	11
5.6.2	Expo :	11
6	DÉVELOPPEMENT DES DIFFÉRENTES TÂCHES RÉALISÉES	12
6.1	Introduction	12
6.2	Application source :	13
6.2.1	Montée en compétence	13
6.2.2	Avancement :	13

6.2.3	Diagrammes simplifiés de flux :	14
6.3	Application client :	16
6.3.1	Montée en compétence :	16
6.3.2	Avancement :	16
6.3.3	Diagrammes simplifiés de flux :	17
6.4	Application de livraison :	18
6.4.1	Présentation :	18
6.4.2	Montée en compétence :	18
6.4.3	Responsabilité niveau des tâches	19
6.4.4	Présentation des maquettes :	19
6.4.5	Diagramme de flux :	29
6.4.6	Développements réalisés :	30
6.4.7	Backlog :	32
7	CONCLUSION	33
8	LISTE DES REFERENCES BIBLIOGRAPHIQUES	34
9	ANNEXES.....	35
9.1	Tiroir latéral de navigation.....	35
9.2	Barre de menu horizontal.....	36
9.3	Accueil.....	37
9.4	Liste de commandes prêtes pour la livraison.....	38
9.5	Prendre en charge une livraison en attente	39

1 INTRODUCTION

1.1 Contexte

Les applications de livraison de repas comme Uber Eats, Deliveroo, etc ont changé nos habitudes de consommation, rendant possible le fait de se faire livrer quasiment tout type de plat en quelques minutes.

Ces applications ont pour modèle de proposer des repas dans un certain rayon autour du client final. Cela pose de facto un problème logique de localisation : ceux qui vivent ou travaillent dans des environnements très urbains ont accès à un plus large éventail de choix que ceux qui se trouvent dans des secteurs un peu plus éloignés.

1.2 Problématique

Pour prendre l'exemple de l'Île-de-France, beaucoup de gens travaillent à Paris et proche banlieue mais vivent un peu loin de leur travail. Lorsque ces personnes sont en télétravail par exemple ou tout simplement en week-end, les applications « main stream » de livraison leur proposent dans la grande majorité des fast-foods, ce qui pose donc un problème de la qualité et de diversité des repas.

1.3 Constatation

Paradoxalement, pour continuer sur l'exemple de l'IDF, en banlieue nous trouvons très souvent une grande mixité culturelle qui se traduit par de nombreux aspects et en particulier – ce qui va nous intéresser – l'aspect culinaire. Une réflexion peut alors émerger : « Ne serait-il pas possible d'enrichir l'offre de livraison de repas dans ces secteurs en mettant à profit la mixité culinaire déjà présente ? »

1.4 Critiques

Une solution inspirée des applications « main stream » qui souhaiterait répondre à la réflexion soulevée en 2.3 se trouverait confrontée rapidement à 2 obstacles majeurs :

- Un client qui commande via les applis susmentionnées souhaite être livré le plus rapidement possible. Mais cela est permis par le fait que ces applications travaillent avec des restaurants dont l'activité première est de préparer des plats. Or dans un concept tel que celui que nous décrivons, les plats seraient préparés par des personnes qui ne sont pas forcément des restaurateurs, provoquant de facto un délai de livraison plus important.
- Induit par le point ci-dessus, il faudra que ce concept trouve son audience. En effet nous sommes dans une ère de livraison rapide donc une application de livraison proposant des repas livrés dans 2-3 jours (par exemple) peut être vu comme une solution à « contre-courant »

À contrario ri, un tel concept en plus de proposer une alternative à la livraison de fast-food pourrait permettre de :

- Donner un moyen de s'essayer à la restauration pour les personnes n'ayant pas forcément le temps ou les moyens d'ouvrir un restaurant. Ils auront via notre solution un moyen de tester leurs plats.
- Donner plus visibilité aux cuisiniers indépendants qui passent parfois par du bouche à oreille ou par les réseaux sociaux pour les plus modernes.

Cette solution resterait finalement une application géolocalisée, mais pourrait permettre d'apprécier les talents culinaires cachés dans son secteur.

2 ABSTRACT

2.1 Contextual

The emergence of food delivery platforms like Uber Eats and Deliveroo has transformed our consumption patterns, enabling the rapid delivery of a wide array of dishes. These apps operate by presenting meal options within a specific radius of the end user, which inherently introduces a logistical challenge related to location. Individuals in urban settings enjoy a more extensive selection compared to those situated slightly farther away.

2.2 Core Issue

Consider the scenario of Île-de-France: numerous individuals work in central Paris and its immediate environs but reside at a distance from their workplaces. During periods of remote work or leisure, conventional delivery apps predominantly offer fast food choices. This predicament raises concerns about meal variety and quality.

2.3 Notable Observation

Ironically, suburban areas like Île-de-France exhibit remarkable cultural diversity, particularly evident in culinary preferences—a point of interest here. This leads to contemplation: "Could we not enhance meal delivery options in these localities by tapping into the existing culinary diversity?"

2.4 Key Challenges

A solution inspired by mainstream apps that aims to address the consideration presented in section 2.3 would swiftly encounter two major obstacles:

- App users prioritize swift deliveries, a feat achieved by collaborating with eateries specializing in meal preparation. Conversely, the concept we're discussing entails meals crafted by individuals who may not be traditional restaurateurs, potentially leading to longer delivery times.
- Additionally, this novel approach needs to find its audience in an era favoring quick deliveries. Introducing a delivery app offering meals in 2-3 days (for instance) might seem counterintuitive.
- Conversely, such a concept could serve as not only an alternative to fast food delivery but also a means for:
 - Aspiring culinary enthusiasts to explore gastronomy without the need or means to open a restaurant, providing them a platform to test their culinary creations.
 - Offering greater visibility to independent chefs who often rely on word-of-mouth or social media for promotion, particularly the more contemporary ones.
 - Ultimately, this solution would remain a location-specific application but could unveil hidden culinary talents within the vicinity.

3 INTEGRATION DU DOMAINE D'ETUDE DANS L'ACTIVITE DE L'ENTREPRISE

3.1 Identité de l'entreprise :

Raison sociale : TOUT'IT

Forme juridique et capital : SARL au capital de 5000€

Taille : 0 salarié, un alternant et un chiffre d'affaires de 186000€ en 2022

Historique : Créée en 2021, en plein développement

3.2 Informations commerciales :

Offre : Sous-traitance informatique

Marché et clientèle : Intervient surtout pour les grandes entreprises (TOTAL, ENGIE, NATIXIS, etc)

Stratégie de développement :

1. Réaliser des prestations pour les différents clients
2. Utiliser la trésorerie générée par ces prestations pour financer d'autres projets et tester d'autres idées comme l'application mentionnée dans ce document.
3. En cas de concept assez rentable mis sur pied, en faire une entité à part détenue par TOUT'IT.

4 ENVIRONNEMENT DE TRAVAIL

4.1 Forme d'organisation :

4.1.1 Télétravail :

Mon tuteur pratique le télétravail en raison de sa position individuelle dans l'entreprise et de la nature distante de ses prestations. Actuellement, l'absence de locaux physiques ne pose pas de problème.

Pour faciliter notre collaboration, mon tuteur a opté pour l'utilisation de Slack en tant qu'outil de collaboration pour les raisons suivantes :

- Le partage d'écran permet de suivre mes développements en temps réel
- Messagerie instantanée
- Possibilité de brancher slack avec des outils externes (Bitbucket, Gitlab, etc)

L'utilisation de GitLab qui joue un rôle essentiel dans :

- Le versioning de l'application
- La gestion des tâches et des branches
- Résolution de conflits.
- Code Review.

4.1.2 Daily meeting/réunion :

Mon tuteur s'est inspiré des méthodes AGILE (Scrum) pour la gestion du projet. Cependant ces méthodes devant s'adapter à la taille de l'équipe, il a choisi de faire un compromis entre daily meeting et réunions classiques. Cela s'implémente via des réunions entre 10 min et 45 min, celles-ci peuvent avoir lieu plusieurs fois par jour en cas de sujet(s) complexe(s).

De plus, ces réunions permettent d'ouvrir à la discussion sur des sujets plus large notamment :

- Des technologies émergentes
- Modalité de travail en entreprise
- Salaire à l'embauche

4.1.3 Code review :

La pratique régulière de code review avec mon tuteur a grandement renforcé mon processus de développement d'application. La soumission du code et les discussions architecturales qui s'ensuivent créent un espace d'apprentissage continu, où les choix de conception et de mise en œuvre sont discutés.

Cette interaction régulière permet d'explorer diverses méthodes de résolution de problèmes et de s'assurer que le scope de la tâche qui m'a été attribuée est correct.

Cela nous permet également d'avoir des échanges sur :

- Comment faire du clean code
- Comment faire du code maintenable
- Comment faire du code modulaire

4.2 Moyen matériel :

Mon pc portable personnel suffit amplement pour pouvoir réaliser les tâches qui me sont dédiées lors de l'alternance.

4.3 Système d'exploitation :

Mon tuteur étant un habitué de linux a opté pour Ubuntu comme système d'exploitation. La familiarité de mon tuteur avec Ubuntu permet une assistance sur le développement et le suivi des objectifs de mon alternance.

4.4 Hébergement :

4.4.1 OVH Cloud :

Mon tuteur a fait le choix d'utiliser OVH Cloud comme hébergeur pour les raisons suivantes :

- Mettre en place un environnement partagé aisément accessible en ligne.
- Bénéficier d'un nom de domaine valide et maintenu par OVH directement
- Les services offerts sont variés et les coûts sont raisonnables.
- Pas besoin de « gros » cloud pour l'instant comme AWS ou Azure, un simple VPS (Virtual Private Server) suffit pour l'instant.

4.5 Environnement de développement :

4.5.1 Visual studio code :

Avec mon tuteur nous avons fait le choix d'utilisé Visual studio. En effet l'utilisation de Visual Studio Code en tant qu'IDE pour développer une application présente de nombreux avantages pour nous, comme pouvoir intégrer un gestionnaire de version tel que git, l'auto complétion de différentes fonctions ou encore ses outils de débogage avancés.

4.5.2 Expo go :

Lorsque je suis arrivé sur le projet, mon tuteur utilisait déjà l'outil (application mobile) Expo go.

Cette application démarre un serveur local qui exécute en temps réel le code React pour produire les interfaces (les vues) sur le périphérique (téléphone, tablette). Cela permet de voir en direct les résultats du développement, ces interfaces sont aussi mises à jour au moindre changement dans le code (Hot reload).

En effet, travaillant principalement à distance avec mon tuteur, il a fallu trouver un moyen de pouvoir partager mon travail avec lui dès que nécessaire, et ce moyen est Android studio.

4.5.3 Android studio :

Android Studio propose un émulateur Android intégré, qui nous permet de tester et de déboguer l'application sur différentes configurations de périphériques Android.

De plus, Android Studio facilite grandement le code review car l'émulateur est accessible directement sur notre écran contrairement à expo go qui nous affiche les erreurs visuelles sur le téléphone. Cela nous permet de partager aisément le code et en même temps les écrans de l'application ce qui nous offre un gain de temps au niveau du code review.

4.5.4 Git :

L'intégration de Git lors du développement nous est indispensable pour pouvoir interagir avec des plateformes de travail comme Gitlab.

Git nous offre :

- **Viersoning** du code
- Gestion des branches
- Réversibilité
- Sécurité et sauvegardes

Cela signifie que chaque changement effectué dans le code source est enregistré et archivé, ce qui permet aux développeurs de suivre l'évolution du projet dans le temps et surtout de pouvoir revenir à une version précédente du code en cas de problème (bug, régression, etc.)

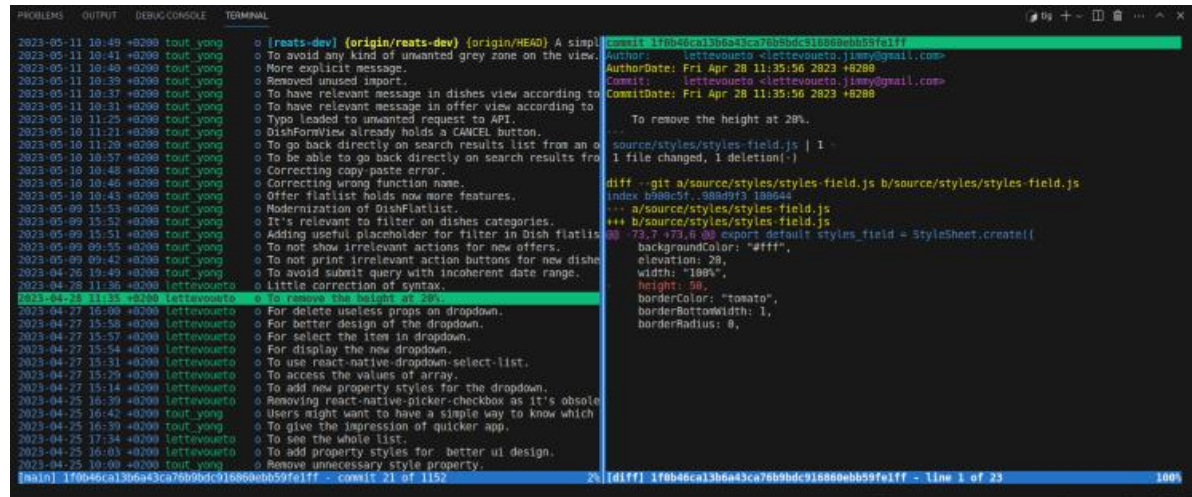
Étant donné que nous travaillons avec Ubuntu, il semblait naturel pour mon tuteur de me familiariser le plus possible avec les lignes de commande de git. Git étant un outil très puissant, mon tuteur a dans un premier temps souhaité que je connaisse et assimile les commandes de base pour un développeur (commit, push, checkout, reset, pull, merge).

De plus selon les situations, le tuteur me guide avec des lignes de commande permettant des actions plus avancées si besoin.

4.5.5 Tig :

Non intégré dans Ubuntu, tig est un outil qu'il faut installer à part. Son intérêt est double, à savoir :

- Visualisation interactive des commits
- Une interface plus « user-friendly » que celle rendue par git

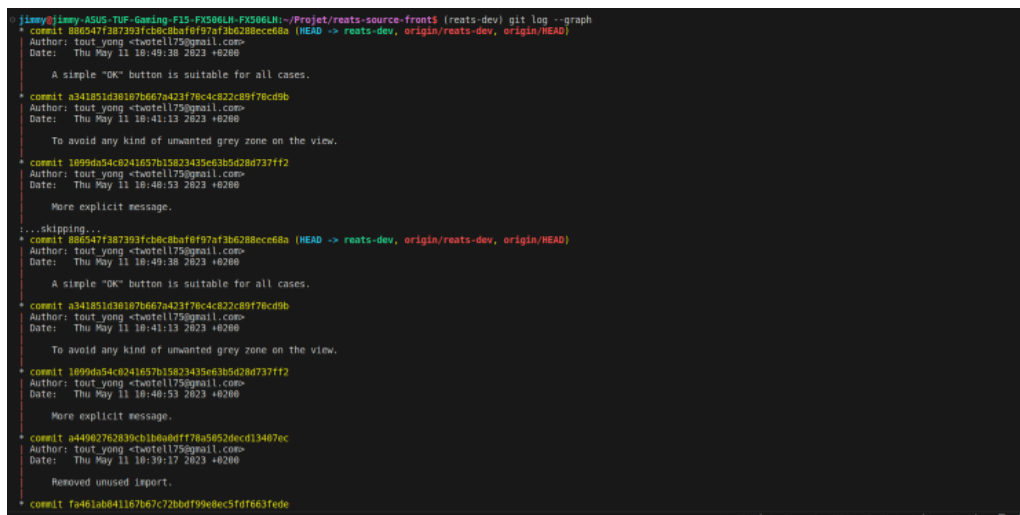


The image shows the Tig interface, a text-based tool for viewing Git commit history. The left pane displays a list of commits with their hashes, dates, and authors. The right pane shows the diff for the selected commit, highlighting changes in the file `source/styles/styles-field.js`. The interface is dark-themed and includes a status bar at the bottom.

Image 1 : Exemple d'affichage avec la commande «tig»

La commande **tig** permet d'explorer en détail chaque commit, en affichant de manière organisée les modifications apportées aux fichiers ainsi que les commentaires associés.

Sans **tig** il faut utiliser **git log --graph** qui affiche également les détails de commits, mais le rendu est moins « user friendly ».



The image shows the output of the `git log --graph` command in a terminal. It displays a graph of commit history with hashes, dates, and authors. The output is text-based and lacks the interactive features of Tig.

Image2 : Exemple d'affichage avec la commande **git log --graph**

4.6 Framework :

4.6.1 React native :

Mon tuteur étant un développeur back-end, il a effectué des recherches approfondies pour choisir la technologie front-end la plus adaptée pour son projet.

Son choix s'est porté vers React Native aux vues de sa courbe d'apprentissage plus simple, de son code cross-platform qui permet de créer des application Apple/Android et dans sa polyvalence en tant que langage web et mobile.

4.6.2 Expo :

Le choix technologique du tuteur d'utilisé expo s'appuie ses raisons suivantes :

- Un SDK bien fourni
- Une Bibliothèque de composants prêts à l'emploi.
- Un Déploiement rapide sur Android ou Ios.
- Des outils de débogage simplifié.
- Des générations de template de projet structuré.
- Mises à jour d'application facile.
- Expo est une couche au-dessus de React Native qui simplifie la prise en main et le développement via l'utilisation des commandes fournies par l'outil.

5 DÉVELOPPEMENT DES DIFFÉRENTES TÂCHES RÉALISÉES

5.1 Introduction

Sur le plan technique, ce projet repose sur trois applications interconnectées.

- L'application destinée aux cuisiniers : elle leur permet d'exposer leurs créations et de recevoir des commandes,
- L'application destinée aux clients finaux : elle leur permet de passer des commandes.
- L'application livreur : elle permet d'assurer l'acheminement des denrées commandées par le client.

Lorsque je suis arrivé sur le projet, l'état des lieux était le suivant sur la partie front-end :

- Application des cuisiniers développée à 90 % (les fonctionnalités majeures étaient déjà implémentées)
- Application cliente à peu près à la moitié selon mon tuteur.
- Application de livraison à faire (aucun développement entamé)

Les objectifs visés à la fin de cette alternance sont ambitieux. Après discussions avec le tuteur, il a été établi la liste des objectifs suivants au tout début de l'alternance :

- Finaliser l'application cliente : intégrant les fonctionnalités de commande et de paiement,
- Compléter le développement de l'application pour les cuisiniers.
- Démarrer l'application de livraison
- Premiers contacts avec le backend.

Le tuteur était cependant conscient que cela représentait beaucoup de travail et que ces objectifs seraient revus à la hausse ou à la baisse selon mes compétences et ma vitesse d'avancement.

Après quelques semaines, le tuteur a pu se faire une meilleure idée de ce qui pourrait être fait ou non et a réduit mes objectifs aux points ci-dessous, dans l'ordre chronologique :

- Montée en compétence sur React Native via :
 - Des tutoriels
 - De la documentation
 - De la lecture de code
 - Des échanges avec le tuteur
- Prise en main de l'application source et implémenter des fonctionnalités simples pour gagner en confiance.
- Se familiariser aux process de développement (gitflow)
- Utiliser tous les outils possibles pour produire du code propre.
- Avancer sur certaines fonctionnalités simples de l'application cliente
- Entamer le développement de l'application livreur pour aborder et mettre en pratique des notions plus avancées sur React.

5.2 Application source :

5.2.1 Montée en compétence

J'ai pu me familiariser avec les fondamentaux de :

- React Native
- Git
- Expo

Difficultés ressenties lors du développement d'une partie de l'application livreur :

- Comprendre la logique et la structure du code existant
- Développer en respectant les normes et la structure du code dans l'application

Responsabilité au niveau de l'application :

- Travailler dans les délais impartis et communiquer avec le tuteur en cas de problèmes ou de retards potentiels.
- Travailler sur des tâches spécifiques :
 - La résolution de bugs
 - L'ajout de petites fonctionnalités
 - L'amélioration de l'expérience utilisateur (« UX »).
- Se documenter sur :
 - React native
 - Git
 - Expo
 - Expo go

5.2.2 Avancement :

Actuellement la 1^{ère} version de l'application est presque terminée, le branchement avec les premières briques du back-end est en cours.

5.2.3 Diagrammes simplifiés de flux :

5.2.3.1 Exposer un plat

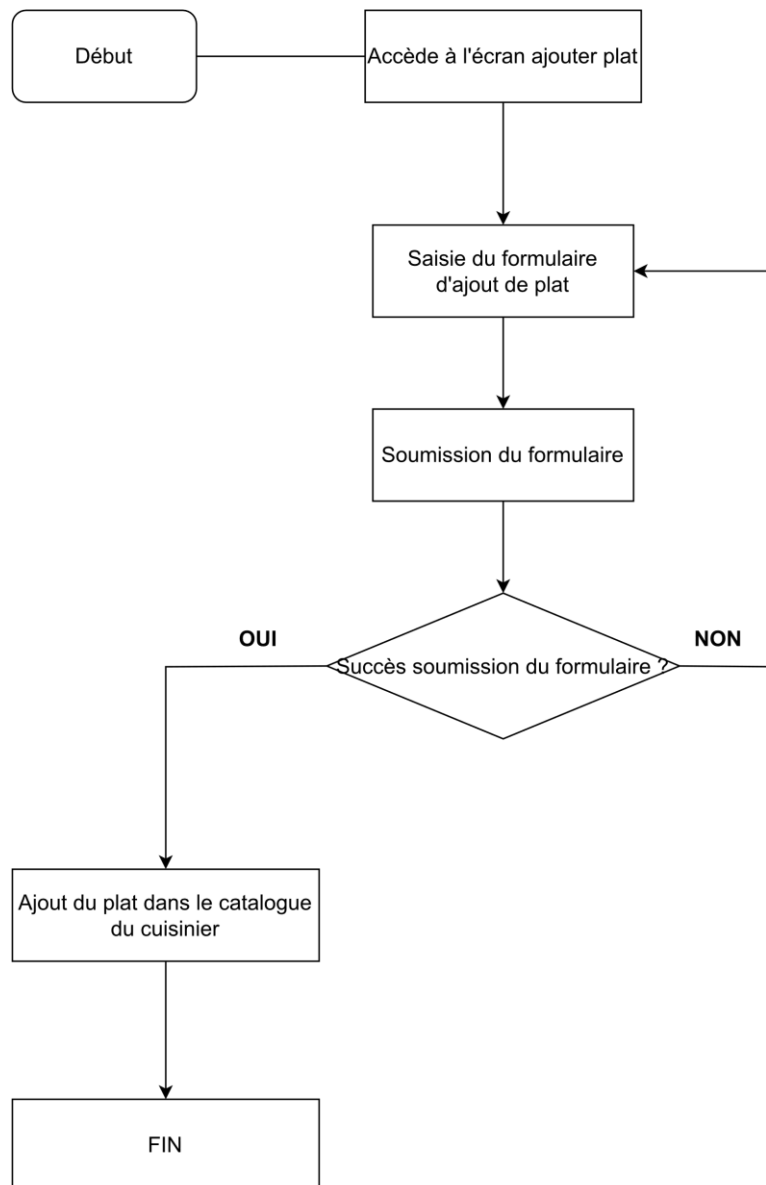


Image 3 : diagramme de flux d'exposition d'un plat

5.2.3.2 Réception d'une commande

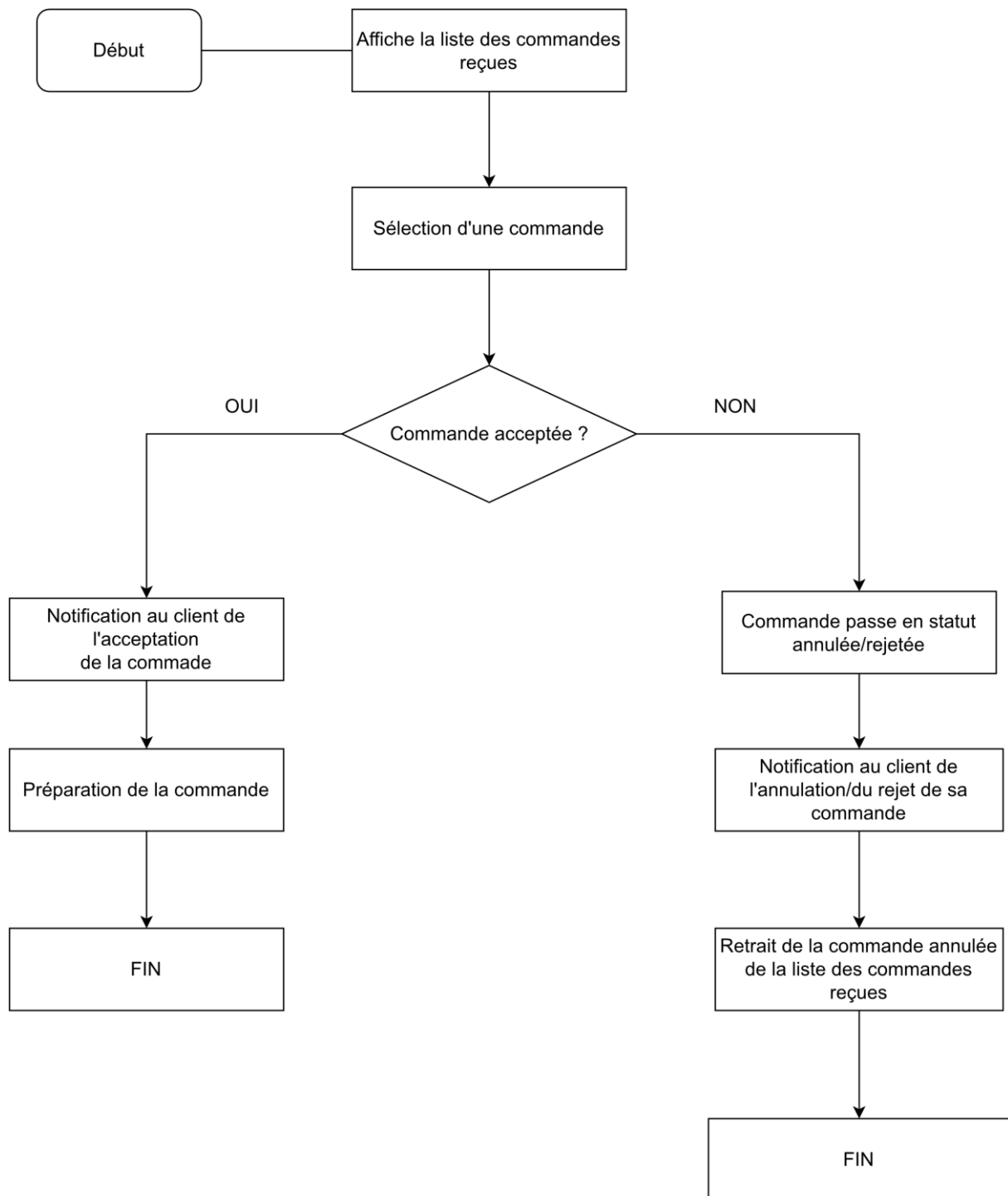


Image 4 : diagramme de flux réception d'une commande

5.3 Application client :

5.3.1 Montée en compétence :

Difficultés ressenties lors du développement d'une partie de l'application client :

- Manipuler la navigation entre plusieurs écrans dans React Native
- Créer une loading animation lors du chargement des données
- Passé d'expo go à Android studio

Responsabilité au niveau de l'application :

- Faire moi-même mes issues sur Gitlab
- Simuler des appels API vers le backend
- Créer un DrawerNavigateur

5.3.2 Avancement :

De nouvelles features ont été développées :

- L'ajouter d'un indicateur d'activité sur les commandes en attente
- Implémentation d'un formulaire de recherche dans pour rechercher des commandes

5.3.3 Diagrammes simplifiés de flux :

5.3.3.1 Créer une commande

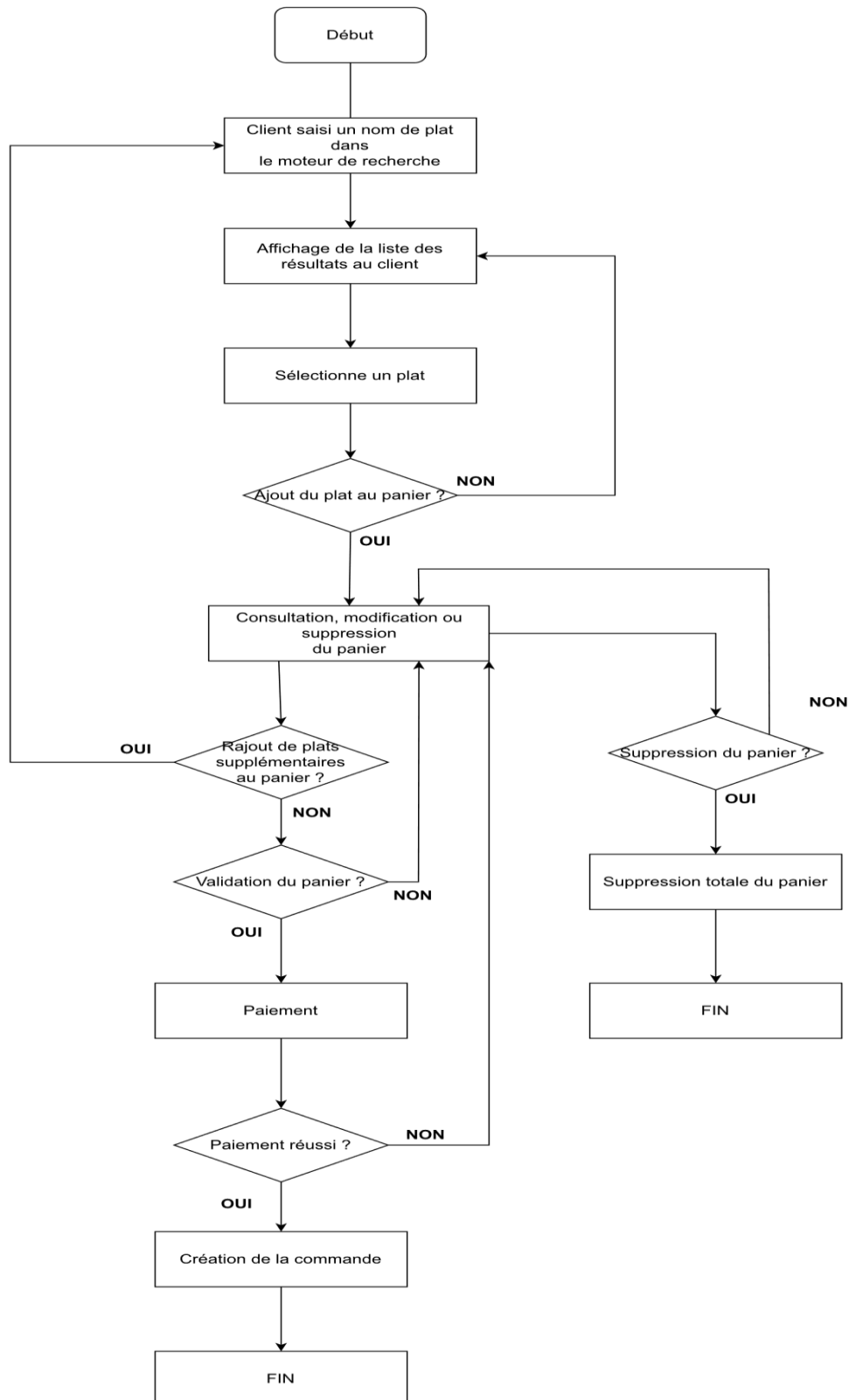


Image 5 : diagramme de flux crée une commande

5.4 Application de livraison :

5.4.1 Présentation :

Cette application permet aux utilisateurs de livrer des commandes au client.

5.4.2 Montée en compétence :

Les notions acquises au cours du développement de l'application livreur :

Pour cette application, le tuteur a souhaité me responsabiliser davantage et m'a donc demandé de m'occuper de réaliser un design simple pour les différentes interfaces.

Des outils comme « Figma » existent et sont dédiés à la réalisation de maquettes avancées dans le monde professionnel, mais nécessitent également une montée en compétence. Cependant au vu du temps et des tâches qui m'attendaient il a opté pour une option plus simple que « Figma » qui est « Excalidraw »

- Maquette :
 - Utiliser Excalidraw pour concevoir des maquettes.
 - Concevoir des interfaces simples et intuitives.
 - Intégrer des notions d'UX et d'UI.
 - Créer une séquence d'actions et d'écrans à travers un storyboard.
- React Native :
 - Créer des composants réutilisables.
 - Écrire des fonctions asynchrones.
 - Gestion du cycle de vie d'un composant.
 - Gestion des données.
 - Gestion des erreurs.
 - Styliser les composants.

Difficultés ressenties lors du développement de l'application livreur :

- Concevoir une navigation fluide entre les écrans.
- Résoudre les problèmes de dépendance des paquets.
- Utilisation des **Hooks** de React
- Utilisation d'un **DrawerNavigator**
- Utilisation des modules **React-Navigation/native** et **React-Navigation/native**

5.4.3 Responsabilité niveau des tâches

- Liberté totale sur la conceptualisation des maquettes.
- Implémenter les interfaces en suivant les maquettes.
- Création et organisation de mes tickets dans Gitlab.
- Gestion quasi-autonome des priorités.

5.4.4 Présentation des maquettes :

5.4.4.1 Tiroir navigateur et barre de navigation



Image 6 : Maquette du tiroir de navigateur et barre de navigation

La maquette présente une barre latérale et une barre horizontale qui offre une navigation agréable entre plusieurs écrans.

5.4.4.2 Accueil

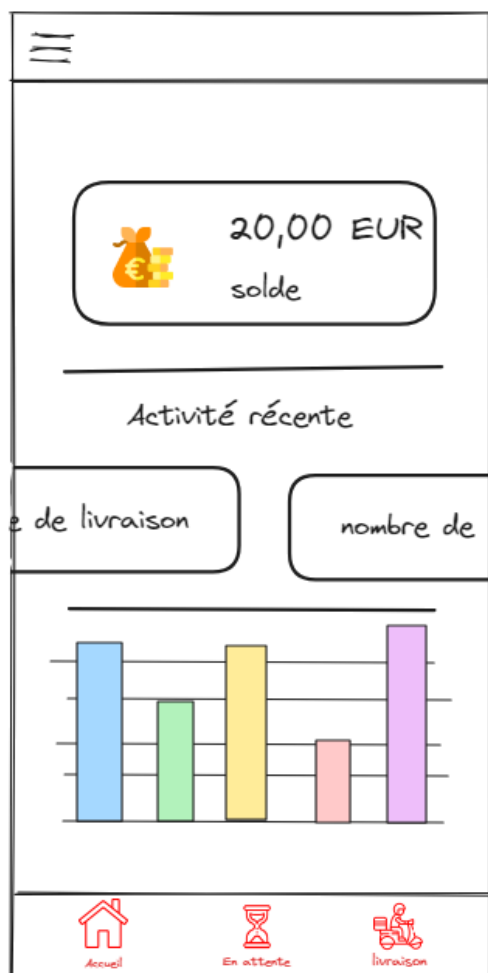


Image 4 : capture d'écran de la maquette d'accueil

La maquette de l'écran d'accueil de l'application mobile présente une disposition optimisée pour les livreurs. Les soldes du livreur sont affichés en haut, tandis que les statistiques hebdomadaires utilisent diagramme en bâton pour favoriser un sentiment de satisfaction. Les informations sur les trois dernières livraisons sont également incluses au milieu de l'écran pour une vue d'ensemble rapide de l'activité récente.

5.4.4.3 Sélectionner une livraison dans la liste des livraisons disponible



Image 7 : Maquette de la liste des livraisons disponibles

Cette maquette présente une liste de livraisons disponibles, triée par ordre croissant en fonction de leur proximité par rapport au domicile du livreur.

Chaque élément affiche :

- Une photo du plat
- Le numéro de commande
- La quantité de plats
- La date de récupération prévue

La photo du plat principal est purement esthétique car nous ne pouvons pas afficher les photos de tous les plats dans ces éléments nous avons donc fait le choix de mettre la photo du 1^{er} plat principale pour représenter la commande de livraison.

Le numéro de commande fournit une identification unique pour chaque livraison, simplifiant ainsi la communication et la gestion.

Le nombre de plats offre au livreur une estimation précise de la quantité de plats à transporter, ce qui lui permet de savoir s'il est équipé ou non pour effectuer cette livraison.

Enfin, la date de récupération anticipée permet au livreur de planifier son itinéraire en conséquence, garantissant ainsi le respect des délais.

L'affichage de ses informations est justifié par le désir d'offrir aux livreurs une vue instantanée des informations cruciales.

5.4.4.4 Prendre en charge une livraison en attente

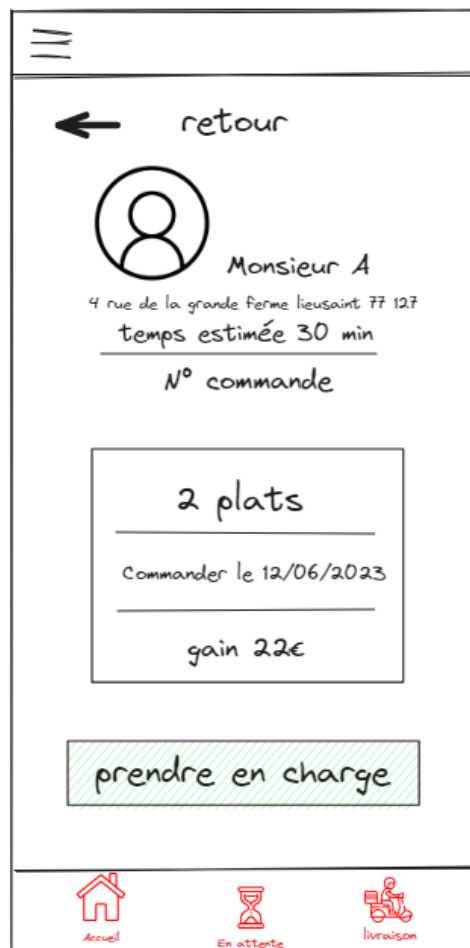


Image 8 : Maquette de prise en charge de livraison

En cliquant sur une livraison disponible dans la maquette de sélection de livraison s'affiche ceci, fournissant des informations en plus sur la livraison telles que :

- Le nom du cuisinier
- La photo du cuisinier
- Son adresse
- Le temps estimé pour effectuer la livraison
- Et un bouton pour valider la prise en charge.

L'affichage de ces informations dans cette maquette vise à fournir au livreur des informations approfondies lorsqu'il sélectionne une livraison spécifique.

La photo du cuisinier est optionnelle mais elle permet une identification visuelle du cuisinier.

L'adresse du cuisinier est cruciale pour la navigation du livreur vers le point de prise en charge

En incluant le temps estimé pour atteindre la destination, le livreur peut mieux s'organiser.

Globalement, ses deux maquettes ont pour but d'offrir au livreur toutes les informations nécessaires pour accomplir efficacement la prise en charge de la livraison.

5.4.4.5 Commencer et valider la livraison

Concernant le commencement et la validation de livraison deux maquettes ont été faites.

Cela a été justifié par des remises en question sur certain point dû à d'éventuelle limitation technique :

- Comment intégrer une maps dans l'application en React Native
- Comment rediriger le livreur vers une tiers application sur son téléphone

Dans le cas où nous intégrant une maps dans l'application, la version 1 se trouve être la plus judicieuse en effet notre application sera autonome et donc aptes à générer un itinéraire pour chaque commande lancer

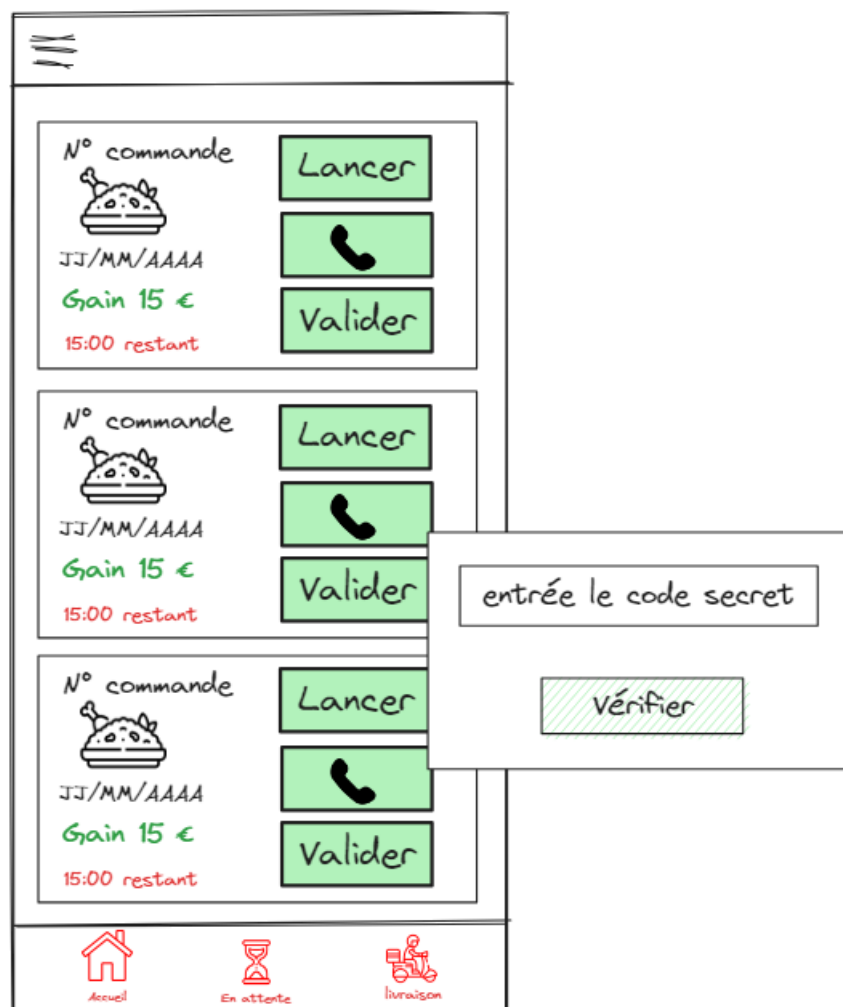


Image 8 : capture d'écran de la livraison version1

Dans le cas contraire la version 2 répond à la problématique du deuxième point lancer la livraison qui va nous rediriger vers une application tierce possiblement par **AppLink** dans laquelle un itinéraire va être fait avec la livraison qu'il a lancé il viendra ensuite dans l'appli pour confirmer valider étape par étape la livraison de chaque comment

L'utilisateur sera alors redirigé vers cette application avec les itinéraires dès qu'il l'a validé

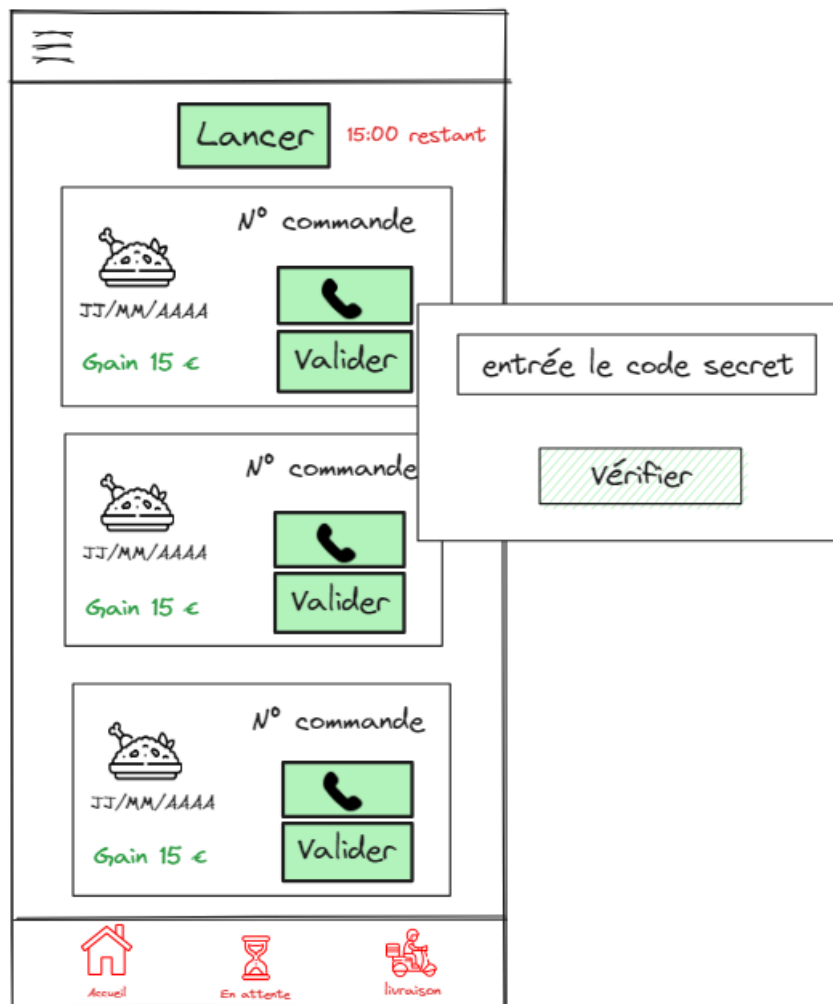


Image 9 : capture d'écran de la livraison version 2

Les deux maquettes illustrent un écran de livraison, chacun affichant une liste de livraisons comprenant les mêmes informations essentielles que la maquette de livraison disponible.

Chacun comporte également trois boutons :

- Un pour initier la livraison,
- Un autre pour entrer en contact avec le client
- Un troisième pour confirmer.

Après avoir actionné ce dernier bouton, un élément apparaît, exigeant du livreur qu'il saisisse un code secret pour confirmer l'achèvement de la livraison

Nous avons ajouté un compteur lorsque que le compteur arrive à 0 si le livreur n'a toujours pas démarré la livraison le jour de l'échéance alors la livraison est de nouveau disponible pour d'autre livreur.

5.4.4.6 Liste d'historique de livraison

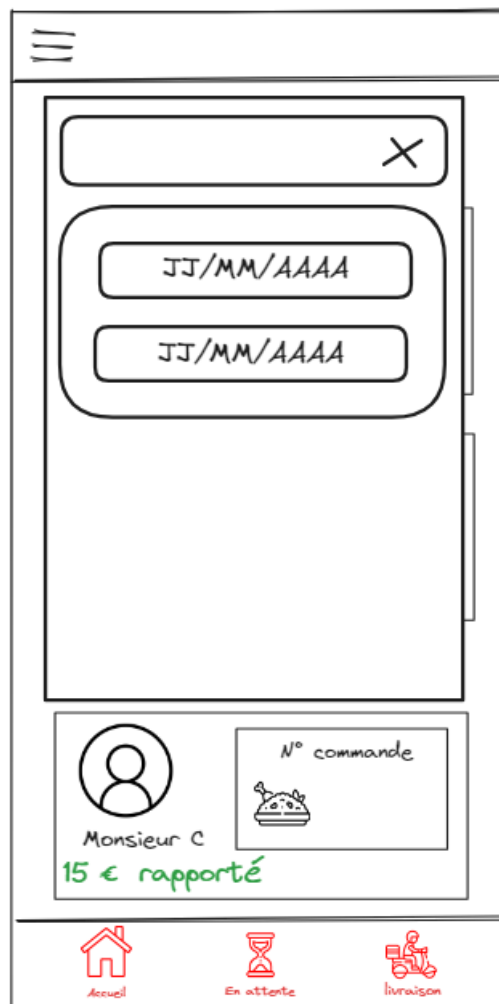


Image 10 : capture d'écran de la maquette liste d'historique de livraison

La maquette de liste d'historique de livraison présente simplement les anciennes livraisons effectuées.

Un filtre de recherche calendaire est intégré pour permettre aux utilisateurs de trier et d'afficher les commandes livrées en fonction de la période désirée.

5.4.4.7 Détail d'une livraison effectué

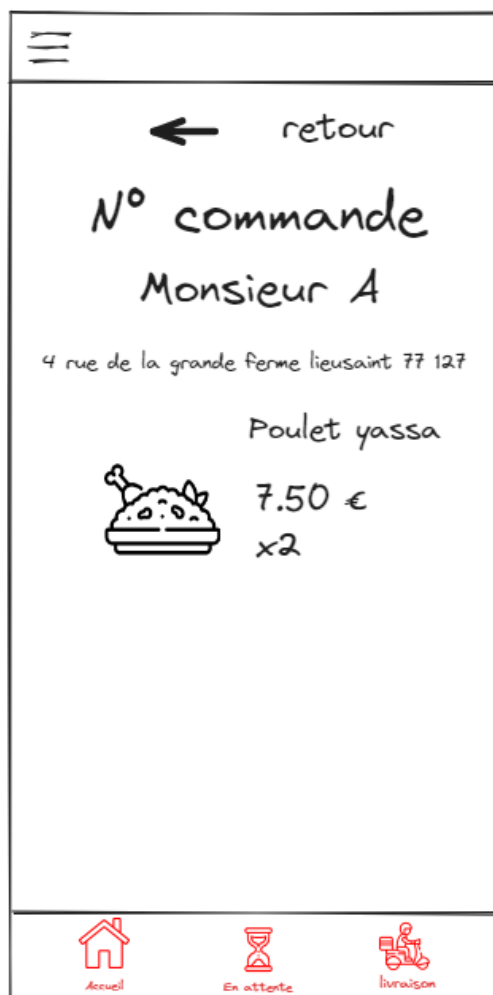


Image 11 : Maquette détails d'une livraison effectué

Cette maquette représente la vue qui s'affiche lorsque que l'on sélectionne une livraison dans l'image 10.

Elle permet d'accéder à des informations plus détaillées sur la livraison.

5.4.4.8 Statistiques

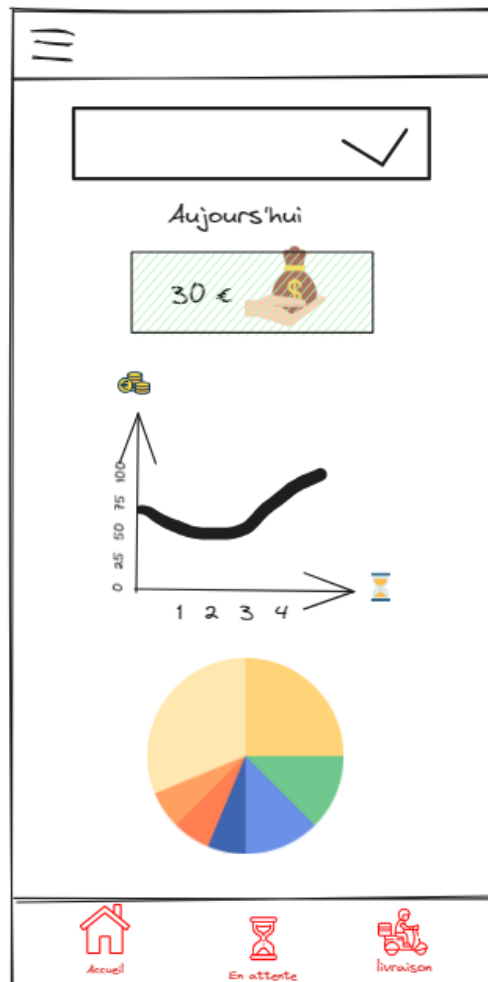


Image 12 : Maquette des statistiques de livraison

Dans la section statistique de la maquette, j'ai inclus un graphique circulaire et un diagramme en courbe représentant les statistiques de ses gains sur une période donnée, lui permettant de visualiser ses performances s'il le désire.

La fonction de filtrage par semaine permet aux utilisateurs de cibler des périodes spécifiques pour une évaluation plus précise de leur gain.

5.4.5 Diagramme de flux :

5.4.5.1 Prise en charge de la livraison

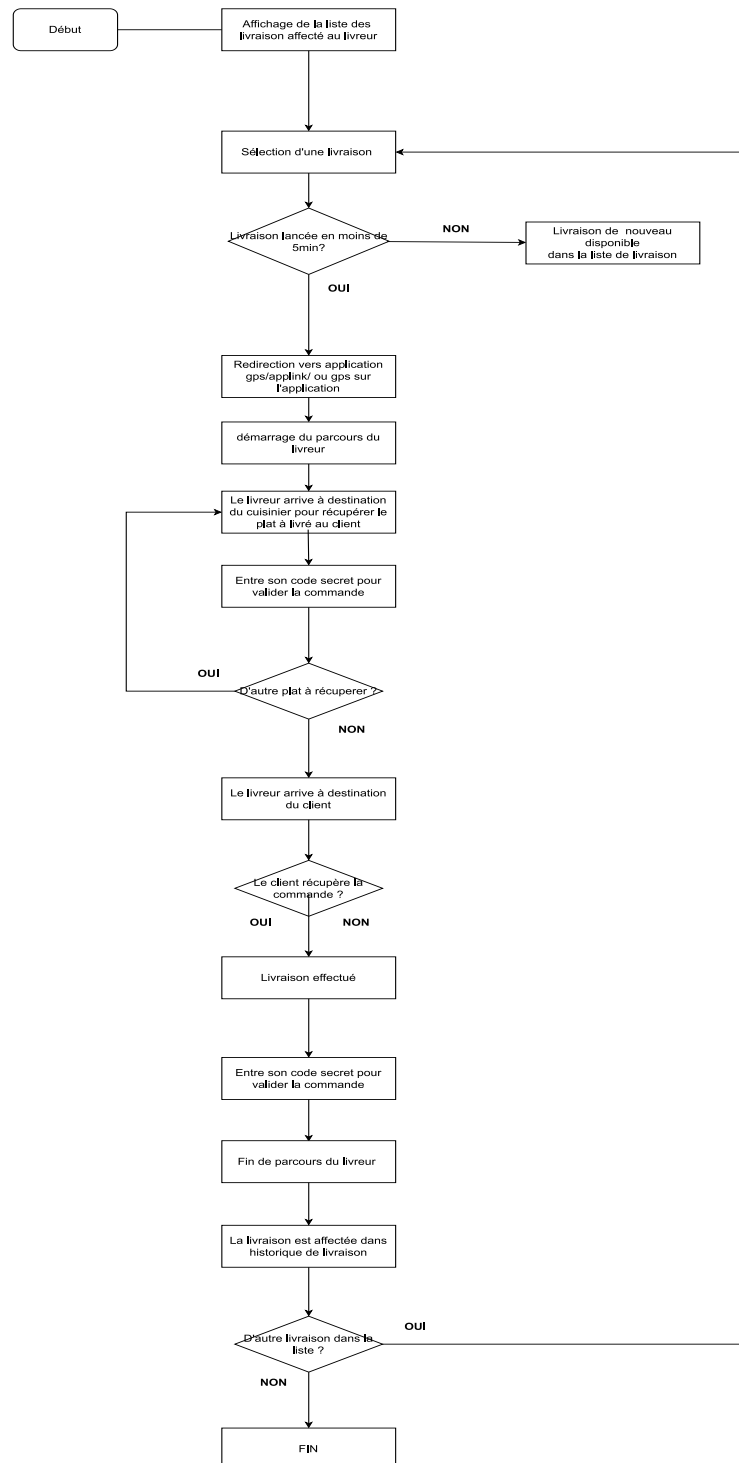


Image 13 : diagramme de flux prise en charge de livraison

Ce diagramme de flux représente la procédure pour pouvoir effectuer une livraison, c'est à partir de cette base que nous sommes partis pour la conception des maquettes et l'implémentation de celle-ci dans l'application.

5.4.6 Développements réalisés :

5.4.6.1 Tiroir de navigation latérale

L'intégration du tiroir latérale a été effectuée grâce au composant **CreateDrawerNavigator** pour créer la navigation par tiroir.

Nous avons personnalisé le contenu du tiroir en utilisant le composant **MainDrawerContent** on y ajoutant des infos tel que :

- La photo de profil de l'utilisateur
- Son nom
- Des sections

Ses infos sont uniques pour chaque utilisateur alors nous avons utilisé le composant et le **Hook UseEffect** pour simuler un appel API pour récupérer les informations spécifiques à chaque livreur et le composant **ActivityIndicator** pour créer une animation qui indique qu'une tâche est en cours en arrière-plan notamment la récupération des informations du livreur.

Résultat voire l'annexe 9.1

5.4.6.2 Barre menu horizontal

La barre de menu horizontale a été implémentée à l'aide du composant **BottomTabNavigator**.

J'ai également ajouté des icônes personnalisées. En effets les icônes des onglets sont personnalisées en fonction de leur état (sélectionné ou non) à l'aide des icônes fournies par les composants **Ionicons** et **MaterialIcons** d'expo **VectorIcons**. Lorsque l'utilisateur sélectionne un onglet, l'icône associée est affichée en couleur, tandis que les autres icônes restent en nuances de gris pour indiquer leur état inactif.

Résultat voire l'annexe 9.2

5.4.6.3 Accueil

Pour implémenter la page d'accueil j'ai utilisé des composants natifs React tels que :

- ***View***
- ***Text***
- ***Image***
- ***FlatList***.

Ces composants ont été utilisés pour afficher les textes, les images et la liste des 3 dernières livraisons effectuées.

Pour enrichir l'expérience visuelle de l'utilisateur, nous avons intégré des icônes avec le module ***Expo VectorIcons***

L'actualisation en temps réel des données a été gérée grâce au concept d'état (state) de React. Nous avons fait des appels asynchrones pour récupérer périodiquement les données à afficher, maintenant ainsi les informations à jour pour les utilisateurs.

Ces fonctions ont été utilisées pour faire cela :

- ***ComponentDidMount***
- ***ComponentWillUnmount***

Pour afficher les statistiques sous forme graphique, j'ai dû utiliser le composant ***BarChart***.

Résultat voir l'annexe 9.3

5.4.6.4 Liste de commandes prêtes pour la livraison

Concernant l'implémentation de la liste de commandes prêtes pour la livraison.

Lorsqu'un élément est sélectionné, l'application navigue vers une vue détaillée de cette commande. Les données pertinentes sont transmises via les paramètres de navigation.

Cela a été possible grâce au composant ***TouchableHighlight***.

Nous avons mis en place un affichage conditionnel basé sur l'état de chargement. Si les données sont en cours de récupération, un indicateur de chargement est affiché pour une expérience utilisateur plus agréable. Sinon, une liste déroulante est affichée pour présenter les éléments de commande.

Cela a été réalisé à l'aide des composants suivants :

- ***ActivityIndicator***
- ***Flatlist***

Et réaliser grâce au ***Hooks*** :

- ***UseState***
- ***UseEffect***

Résultat voire l'annexe 9.4

5.4.6.5 Prendre en charge une livraison en attente

Pour implémenter l'écran de prise en charge d'une livraison en attente j'ai utilisé des composants tels que :

- ***View***
- ***Text***
- ***Image***
- ***FlatList***.

Résultat voire l'annexe 9.5

5.4.7 Backlog :

Ce qu'il reste à faire :

- Implémenter les écrans suivants :
 - Livraison
 - Stats
 - Historique de livraison

6 CONCLUSION

En résumé, ce rapport de soutenance présente mes réalisations pendant l'alternance en développement d'applications mobiles, en se concentrant sur React Native et Expo. J'ai appris les technologies et les outils nécessaires pour créer des applications conviviales, en travaillant étroitement avec mon superviseur.

À travers l'organisation et les méthodes de travail détaillées, il est clair que le télétravail a été adopté de manière efficace, permettant une collaboration fluide et constante grâce à des outils tels que Slack et GitLab. Les réunions régulières ont créé un environnement propice à la discussion et à l'apprentissage, tout en favorisant la communication sur des sujets plus larges liés au développement et au fonctionnement de l'entreprise.

La section portant sur les ressources matérielles et techniques a souligné l'usage de ressources telles qu'OVH Cloud, Visual Studio Code, et Android Studio, qui ont été fondamentales pour le développement et le déploiement des applications. Le mélange de React Native et Expo a permis la création d'interfaces ergonomiques et a accéléré le processus de développement.

En somme, cette période d'alternance a été marquée par des accomplissements significatifs, incluant la finalisation de la première version de l'application source, la création d'interfaces pour l'application de livraison, et l'acquisition des procédés de développement et de collaboration. Les objectifs initiaux ont été atteints et adaptés en fonction des compétences développées et des contraintes rencontrées.

Ce rapport de soutenance reflète une expérience enrichissante dans le domaine du développement d'applications mobiles, en mettant en évidence l'apprentissage continu, la collaboration fructueuse avec le superviseur, et les accomplissements réalisés au sein de l'entreprise. Les compétences obtenues et les réussites accomplies pendant cette période d'alternance constitueront une base solide pour progresser davantage dans le domaine du développement d'applications mobiles.

7 LISTE DES REFERENCES BIBLIOGRAPHIQUES

React native :

- <https://reactnative.dev/docs/intro-react-native-components>

Expo :

- <https://expo.dev/>
- <https://docs.expo.dev/>

Git :

- <https://git-scm.com/>
- <https://git-scm.com/doc>

Tig :

- <https://jonas.github.io/tig/>

8 ANNEXES

8.1 Tiroir latéral de navigation

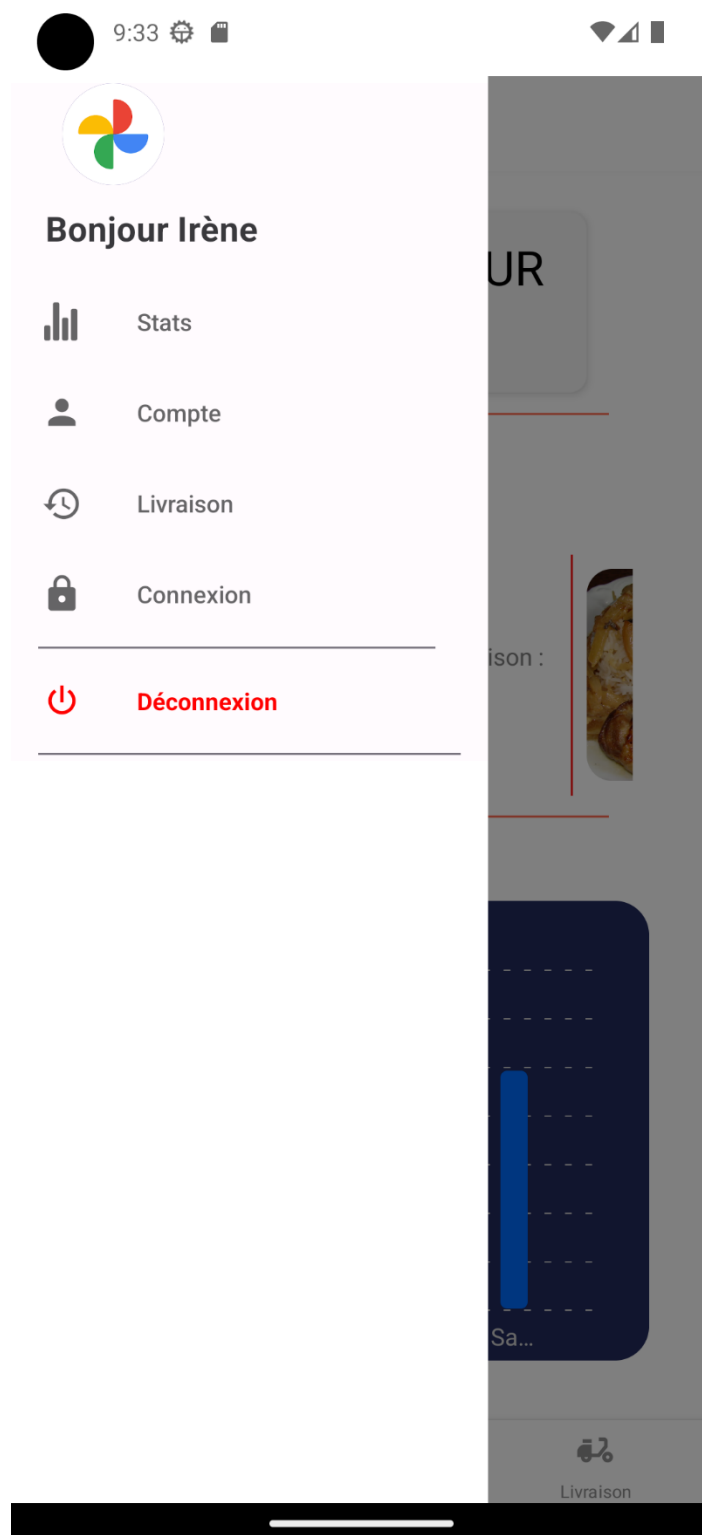


Image 12 : Tiroir latéral de navigation

8.2 Barre de menu horizontal

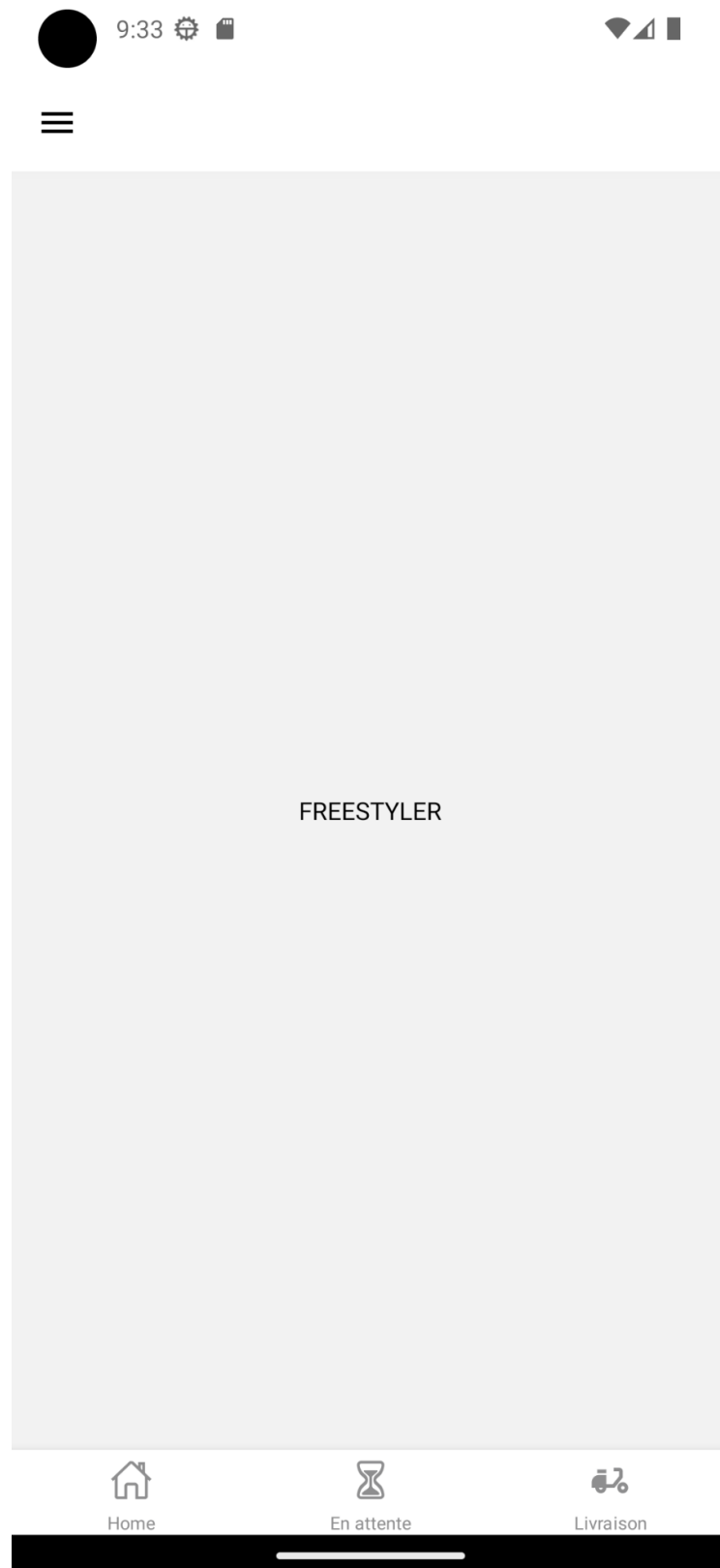


Image 13 : Barre de menu horizontal

8.3 Accueil



Image 14 : Accueil

8.4 Liste de commandes prêtes pour la livraison

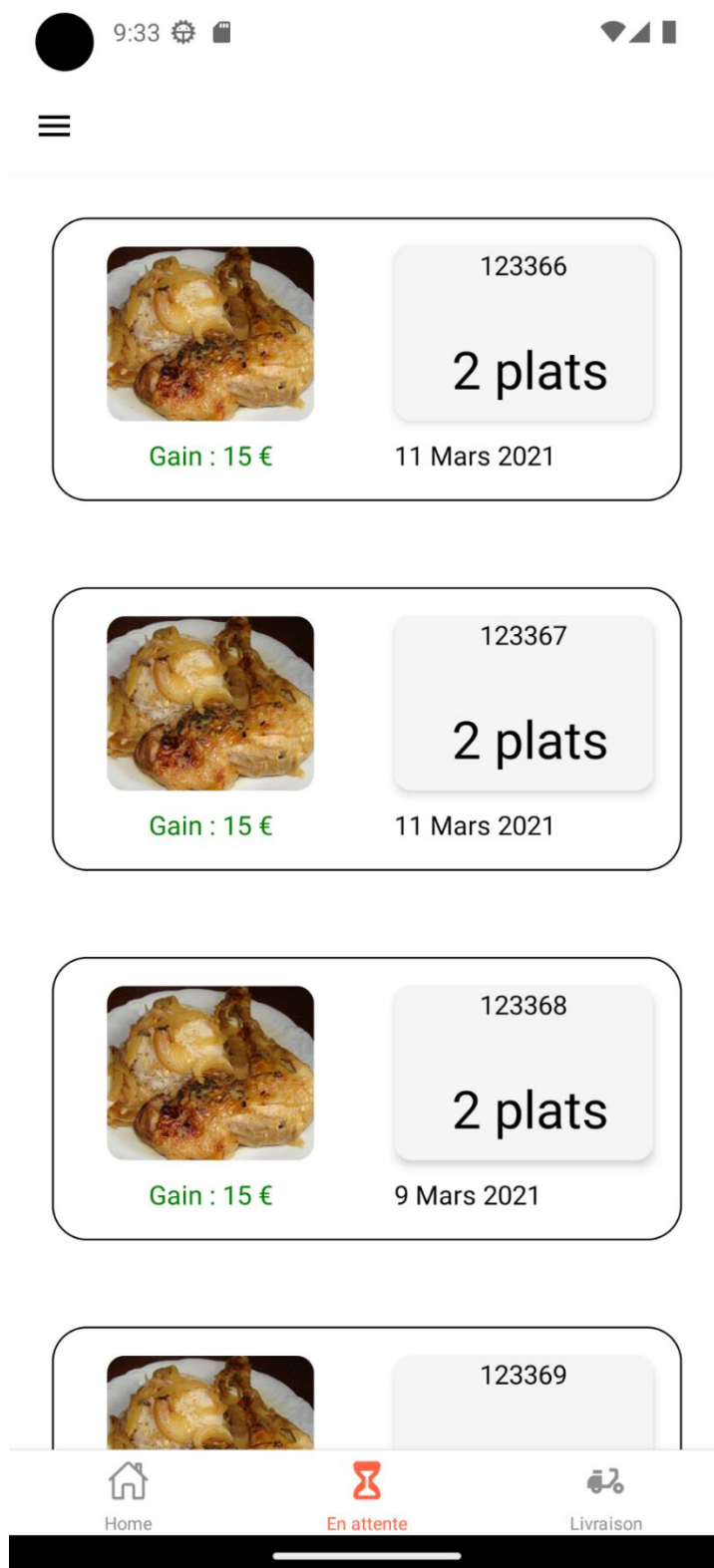


Image 15 : Liste de commandes prêtes pour la livraison

8.5 Prendre en charge une livraison en attente



Image 16 : Prendre en charge une livraison en attente