

ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR

Réalisation d'une application mobile de services à la demande OnDemandAPP

*Réalisé par : ALAOUI DRAI Zineb
Encadré par : Mme NAJI Zineb*

Année scolaire : 2025/2026

Dédicaces

Chers parents bien-aimés,

Les mots sont trop rares pour exprimer votre valeur inestimable à nos yeux. Vous nous avez donné d'innombrables choses en plus des

sacrifices, vous nous avez donné une boussole pour la vie.

Votre

amour silencieux a été une grâce pour nous, votre patience a été notre bouclier. Quoi que nous soyons, c'est vous qui l'avez créé.

Que ces pages soient le miroir de toute votre valeur.

*À notre sang, notre clan, nos partenaires de vie,
Merci d'être nos premiers soutiens, nos critiques affectueux, et
ceux qui peuvent rire de nos échecs avant de nous relever.
Puissiez-vous trouver les chemins éclairés par la lumière que
vous avez allumée sur les nôtres.*

*Pour ceux qui ont choisi de nous aimer sans lien de sang,
L'amitié est une magie rare qui transforme les rires partagés en
liens indéfectibles. Vous êtes les magiciens. Ceci est votre
offrande, avec l'éclat d'un millier de « merci » dans chaque
souvenir partagé.*

*À l'absent qui nous manque,
Quelques lignes volées au silence pour dire que certaines
absences peuplent plus que de nombreuses présences. Où que
vous soyez, sachez que cette histoire porte aussi votre nom.*

Remerciements

Je tiens à adresser mes sincères remerciements à toutes les personnes qui ont contribué à la réalisation de ce projet, notamment au Professeur. NAJI Zineb, pour son encadrement expert, ses précieuses orientations et sa patience tout au long de ce travail. Son expertise technique, sa rigueur académique et son soutien ont été déterminants pour la réussite de ce projet.

Je remercie également l'EMSI pour les ressources pédagogiques et technologiques mises à disposition, ainsi que nos collègues pour leurs échanges enrichissants et leur collaboration.

Résumé

Ce projet final consiste à concevoir et développer une **application mobile de services à la demande**, intitulée *OnDemandApp*, destinée à mettre en relation des clients et des prestataires de services via une plateforme centralisée, intuitive et sécurisée. Cette application représente la synthèse des compétences acquises tout au long de notre formation et illustre notre capacité à concevoir une solution complète répondant à des besoins réels du marché.

Le développement du backend a été réalisé à l'aide du framework **ASP.NET Core**, reconnu pour sa performance, sa sécurité et sa capacité à gérer des API REST scalables, tout en s'appuyant sur une base de données **SQL Server** pour le stockage structuré des données. Le frontend mobile a été développé en **React Native avec Expo**, permettant une application multiplateforme.

Un accent particulier a été mis sur la **sécurité**, avec l'implémentation de l'authentification par **JWT**, la gestion des rôles, le chiffrement des données sensibles, ainsi que l'intégration de **Firebase Cloud Messaging** pour les notifications.

Mots clés : React Native, ASP.NET Core, SQL Server, services à la demande, JWT, 2FA .

Tableau des abréviations utilisées

<i>Abréviation</i>	<i>Signification</i>
2FA	Two-Factor Authentication (Authentification à deux facteurs)
API	Application Programming Interface
FCM	Firebase Cloud Messaging
GPS	Global Positioning System
JWT	JSON Web Token
RN	React Native
SQL	Structured Query Language
UI	User Interface
UX	User Experience
UML	Unified Modeling Language

Tableau 1 : Liste des abréviations techniques utilisées dans le projet, avec leur signification complète.

Introduction générale

Ce rapport a pour but de présenter notre projet de fin d'études qui constitue l'aboutissement des compétences techniques et méthodologiques développées tout au long de notre cursus ainsi que le fruit de notre capacité d'auto-formation, notre capacité à maîtriser des technologies modernes et notre volonté de contribuer au développement de solutions numériques répondant à des besoins concrets.

Nous avons développé une application mobile moderne de services à la demande, nommé OnDemandApp, qui permet de connecter des clients avec des prestataires de services sur une plateforme accessible via une application mobile unique, simple et rapide. En plus de cela, l'utilisateur bénéficie de :

- Une application mobile simple et facile à utiliser pour naviguer.
- Un moyen facile de chercher, commander et suivre.
- Une façon de voir les différents types d'utilisateurs (client, prestataire et administrateur).
- Les notifications de fois en fois pour une meilleure expérience.

La sécurité a également été un point important du projet puisqu'il fallait :

- Protéger les informations personnelles des utilisateurs.
- Garantir la sécurité des échanges entre l'application mobile et le serveur
- .➤ Maintenir la confiance dans le système par des authentifications strictes.

Pour ce projet, nous nous sommes appuyés sur les technologies suivantes :

- Front-end : React Native avec Expo pour le développement d'une application mobile multiplateforme.
- Back-end & Base de données : ASP.NET Core pour le développement de l'API REST et SQL Server pour la gestion des données.
- Sécurité : Authentification par JWT, gestion des rôles et authentification via firebase

Ce rapport est organisé en trois parties :

- Partie 1 : Présentation générale du projet.
- Partie 2 : Étude et conception de l'application.
- Partie 3 : Mise en œuvre et développement de la solution.

Table de figures

Figure 1 : Modélisation Y	21
Figure 2 : Diagramme de Gantt illustrant la planification des tâches	22
Figure 3 : Diagramme d'activité représentant les interactions	27
Figure 4 : Cas d'utilisation montrant les fonctionnalités principales	29
Figure 5 : Diagramme de classes décrivant les entités principales du système	31
Figure 6 : Séquence d'interactions entre l'administrateur et le système	33
Figure 7 : Séquence d'interactions entre le prestataire et le système	35
Figure 8 : Séquence d'authentification client avec, navigation dans les services	37
Figure 9 : Interface mobile – Page d'accueil client avec recherche de services	45
Figure 10 : Interface de sélection des catégories et services.	46
Figure 11 : Interface de confirmation de commande avant paiement.....	47
Figure 12 : Interface de gestion et de suivi des commandes.	48
Figure 13 : Tableau de bord prestataire avec indicateurs de performance.....	49
Figure 14 : Interface de gestion des services et compétences.	50
Figure 15 : Interface de suivi des revenus et du portefeuille.	51
Figure 16 : historique des commandes du prestataire	52
Figure 17 : Tableau de bord de l'administrateur	53
Figure 18 : Interface mobile des paramètres administrateur.....	54
Figure 19 : Interface mobile de suivi des commissions et revenus de la plateforme.	55
Figure 20 : Interface mobile de confirmation et suivi des paiements.	56
Figure 21 : interface des différentes commandes passées	57
Figure 22 : Interfaces des différents utilisateurs	58
Figure 23 : page de validation des demandes d'emploi des prestataires	59

Liste des tableaux

Tableau 1 :Liste des abréviations techniques utilisées dans le projet, avec leur signification ...	5
Tableau 2 : Comparaison des performances et fonctionnalités des plateformes	16
Tableau 3 : Liste des livrables du projet avec leur description et leurs dates de livraison.....	18

Sommaire

<i>Chapitre 1 : Contexte général du projet</i>	11
3.1 Objectif technique	13
3.2 Objectif fonctionnel.....	13
<i>4.1 Besoins Fonctionnels</i>	14
<i>4.2 Besoins Non Fonctionnels</i>	14
<i>5.1 Diagnostic du projet.....</i>	15
<i>5.2 Points d'analyse.....</i>	15
<i>5.3 Applications concurrentes</i>	15
<i>5.4 Critères de sélection.....</i>	15
<i>Conclusions du benchmark.....</i>	16
<i>Points forts</i>	16
<i>Plan d'action.....</i>	16
<i>Priorités.....</i>	16
<i>6.1 Rédaction de l'étude de faisabilité</i>	17
<i>6.2 Rédaction de l'étude fonctionnelle.....</i>	17
<i>6.3 Réalisation des tests et validations</i>	17
<i>6.4 Rédaction du guide utilisateur</i>	17
<i>6.5 Sensibilisation des utilisateurs</i>	17
<i>8.1 Présentation du modèle en Y.....</i>	19
<i>8.2 Étude de faisabilité et analyse des besoins</i>	19
<i>8.3 Spécification fonctionnelle.....</i>	20
<i>8.4 Conception technique</i>	20
<i>8.5 Mise en œuvre</i>	20
<i>8.6 Tests et validation</i>	20
<i>8.7 Progression du projet selon le modèle en Y.....</i>	21
<i>2.1 Introduction</i>	24
<i>2.2 Définition et objectifs du langage UML</i>	25
<i>2.3 Un langage visuel universel</i>	25
<i>6.1 Diagramme de séquence :Administrateur</i>	32
<i>6.2 Diagramme de séquence : Prestataire</i>	34
<i>6.3 Diagramme de séquence :Client</i>	36

<i>Chapitre 3 :</i>	39
<i>Réalisation</i>	39
<i>2.1 Architecture générale</i>	40
<i>Couche de présentation (Front-End)</i>	40
<i>Couche métier (Back-End)</i>	40
<i>Couche de données (Base de données)</i>	41
<i>2.2 Esquisse de l'architecture</i>	41
<i>3.1 Architecture générale</i>	41
<i>React Native : Front-End mobile</i>	42
<i>.NET – Back-End (API REST)</i>	42
<i>JavaScript : Logique et interactivité</i>	43
<i>Base de données relationnelle (SQL Server)</i>	43
<i>JWT (JSON Web Token) : Authentification</i>	43
<i>2FA (Authentification à deux facteurs)</i>	44
<i>Git & GitHub : Gestion de versions</i>	44
<i>Visual Studio Code : Environnement de développement</i>	44
<i>4.1 Interface de l'utilisateur</i>	45
<i>La page « HOME PAGE »</i>	45
<i>La page « Catégories & Services »</i>	46
<i>La page « Confirmation de commande »</i>	47
<i>La page « Mes commandes »</i>	47
<i>4.2 Interfaces mobiles : Prestataire</i>	48
<i>La page « Tableau de bord Prestataire »</i>	48
<i>La page « Services & Compétences »</i>	49
<i>La page « Revenus & Portefeuille »</i>	50
<i>La page « Historique des missions »</i>	51
<i>4.3 Interfaces mobiles : Administration</i>	52
<i>La page « Dashboard Administration »</i>	52
<i>La page « Paramètres »</i>	53
<i>La page « Commissions »</i>	54
<i>La page « Paiements »</i>	55
<i>La page « Commandes »</i>	56
<i>La page « Utilisateurs »</i>	57
<i>La page « Validations Prestataires »</i>	58

Chapitre 1 : Contexte général du projet

1. Introduction

Le succès grandissant des applications mobiles et des plateformes de services à la demande a révolutionnée la manière dont les utilisateurs consomment les services du quotidien. De ce fait, la performance, la sécurité ainsi que la qualité de l'expérience utilisateur sont autant d'enjeux dont il faut prendre compte. À la hauteur de ces enjeux, ce projet s'est fixé pour ambition de concevoir et développer une application mobile de services à la demande moderne, tant sur le plan technique que fonctionnel.

L'objectif de ce projet est de fournir un application mobile rapide et fiable afin de répondre au besoin des clients et des prestataires de services au sein d'une plateforme unique. Elle tire partie des technologies modernes pour offrir une interface utilisateur simple, rapide, intuitive et ergonomique, ainsi qu'un panel de fonctionnalités à destination des administrateurs pour la gestion des utilisateurs, des services et des commandes.

2. Contexte du projet

Dans un monde où tout va de plus en plus vite grâce au numérique et à la mobilité, les usagers sont toujours à la recherche de solutions leur offrant l'accès à des services rapides, mobiles et sécurisés pour satisfaire leurs besoins du quotidien. Les applications de services à la demande sont donc en forte croissance notamment du fait du développement des smartphones, et des améliorations des réseaux et accès en ligne. Mais ces applications deviennent le terrain de nombreux problèmes principalement liés à la sécurité, la fiabilité, ou encore la gestion des interactions entre utilisateurs.

C'est pourquoi, OnDemandApp se veut une réponse aux exigences situées au cœur de ces défis. Cette application mobile propose une expérience utilisateur simple, rapide, efficace mais surtout sécurisée . Des technologies de pointe telles que JWT ou encore l'authentification via firebase et multifacteur permettent de garantir l'accès au seul personnel autorisé, tout en assurant la sécurité des données. De plus, l'intégration de la géolocalisation et des notifications en temps réel améliore considérablement l'expérience utilisateur et la réactivité de la plateforme.

Afin de cibler à chaque utilisateur une expérience adaptée à ses besoins, OnDemandApp se compose de plusieurs interfaces

L'interface administrateur permettant de gérer les utilisateurs, les services, et d'assurer une bonne réactivité en suivant l'ensemble des demandes faites sur la plateforme. L'interface des prestataires qui correspond à l'espace où ils vont pouvoir proposer leurs services, gérer leurs commandes et interagir avec les clients. Et enfin la dernière interface mobile destinée aux clients leur permettant de rechercher un service, commander et suivre l'évolution de leur commande en toute sécurité et simplicité

Ainsi, le projet s'inscrit totalement dans les tendances des applications de services sur mobile.

De plus, ce projet est une bonne occasion pour nous de démontrer que nous sommes capables de concevoir et développer une application moderne, sécurisée et en adéquation avec les attentes du marché.

3. Objectifs du projet

Le projet OnDemandApp poursuit plusieurs objectifs importants, tant d'un point de vue technique que fonctionnel, afin de proposer une application mobile moderne, performante et sécurisée répondant aux besoins actuels des consommateurs.

3.1 Objectif technique

Le premier objectif d'un point de vue technique est de concevoir et développer une application mobile multiplateforme performante avec React Native via Expo, permettant d'avoir une excellente compatibilité sur différents systèmes d'exploitation mobile. L'enjeu est aussi de porter une attention particulière à l'ergonomie de l'interface utilisateur pour assurer une navigation sans accrocs et une expérience attractive.

De plus, le projet s'appuie sur la création d'une API et d'une architecture back-end puissante et scalable avec ASP.NET Core, permettant de gérer les différentes fonctionnalités métier comme l'authentification, les profils utilisateurs, les services ou encore les commandes. L'utilisation d'une base de données SQL Server est le gage d'une donnée fiable et cohérente. Enfin, une couche de sécurité est ajoutée sur les procédures d'authentification avec notamment l'utilisation de JWT, firebase et 2FA afin de sécuriser au maximum les échanges et de protéger les données des clients/usagers.

3.2 Objectif fonctionnel

En termes de fonctionnalités, les objectifs sont de proposer une expérience utilisateur simple et agréable, notamment pour la recherche, la commande, et le suivi de services en ligne. L'application mobile doit permettre à un client de retrouver différents services proposés par des fournisseurs, de visualiser les informations du service, de pouvoir commander ce service et suivre la commande directement dans l'application.

L'application doit intégrer un management des rôles clair où le fournisseur peut gérer ses services et les commandes qu'il reçoit, et où l'administrateur peut consulter l'ensemble de la plateforme. L'application doit aussi permettre une géolocalisation des utilisateurs et des fournisseurs permettant ainsi de proposer des services à proximité des clients. Des notifications en temps réel sont un plus pour accompagner au mieux les clients dans leurs commandes et améliorer leur satisfaction. Pour finir, une inscription et une connexion à l'application simple, sécurisée et rapide est indispensable.

4. Cahier des charges

4.1 Besoins Fonctionnels

L'application OnDemandApp se doit de répondre aux besoins fondamentaux des utilisateurs tout en leur proposant une expérience agréable, simple et sécurisée. Elle se doit également d'aider l'administrateur à gérer parfaitement toutes les fonctionnalités de la plateforme.

Les exigences fonctionnelles ci-dessous exposent les principales fonctionnalités que le système doit être en mesure de proposer, classées par les rôles principaux : client, prestataire, administrateur et système.

Client

- Inscription et connexion sécurisées (confirmation par email, mot de passe sécurisé).
- Recherche des services disponibles par catégorie et par emplacement géographique.
- Consultation des offres de service.
- Commander un service.
- Consulter le statut des commandes (en attente, accepté, rejeté, terminé).
- Recevoir des notifications immédiates.
- Modifier son profil (informations personnelles, mot de passe).

Prestataire

- Se connecter de manière sécurisée.
- Gérer ses offres de services (ajouter, modifier, supprimer).
- Consulter les commandes qui lui sont adressées.
- Refuser ou accepter une commande.
- Consulter l'historique de ses prestations.
- Modifiez votre profil prestataire.

Administrateur

- Se connecter de manière sécurisée.
- Gérer les utilisateurs (clients et prestataires).
- Gérer les catégories de services.
- Gestionnaires de la cantine
- Suivi des commandes et des activités globales.
- Accès à un tableau de bord statistique (nombre d'utilisateurs, commandes, activité globale).

4.2 Besoins Non Fonctionnels

Sécurité

- Impossez la création de mots de passe solides.
- Repoussez les attaques courantes et sécurisez l'accès aux données.
- Sécurisez l'accès aux ressources via des jetons JWT.
- Activez et utilisez systématiquement un second facteur d'authentification (2FA).

Performance et accessibilité

- Assurez des temps de réponse courts pour des opérations courantes.
- Sur mobiles, proposez une interface claire et intuitive.
- Profitez du mobile en ayant des interfaces adaptées à différentes tailles d'écrans.
- Garantissez la disponibilité, la résilience et la fiabilité de votre service.

5. Benchmark

5.1 Diagnostic du projet

Avant de comparer OnDemandApp à d'autres applications, nous avons d'abord pris le temps de faire notre propre bilan. L'idée, c'était de clarifier ce qu'on voulait vraiment accomplir, puis de repérer ce qu'on pouvait améliorer. Notre but principal : offrir une application mobile fiable, sécurisée et performante pour les services à la demande. On veut connecter les clients et les prestataires sans friction.

5.2 Points d'analyse

On s'est concentrés sur plusieurs aspects clés :

- La fluidité et la réactivité de l'application mobile.
- La sécurité, autant pour l'authentification que pour les données utilisateurs.
- L'étendue des fonctionnalités (gestion des services, commandes, notifications).
- La simplicité d'utilisation, que ce soit pour un client, un prestataire ou un administrateur.

Pour mesurer tout ça, on a choisi quelques indicateurs :

- Le temps de chargement des écrans.
- Le taux d'utilisation active.
- Le nombre d'étapes pour finaliser une commande.
- Le nombre d'incidents de sécurité détectés.

5.3 Applications concurrentes

Pour se situer sur le marché, on a étudié plusieurs plateformes majeures dans le secteur des services à la demande :

- Uber (transport et mise en relation)
- Glovo (livraison et services à la demande)
- Fiverr (connexion entre clients et prestataires)
- Upwork (services professionnels en ligne)

5.4 Critères de sélection

Quand on a analysé ces applications, on a porté attention à cinq critères :

- Leur mobilité et accessibilité.
- Leur niveau de sécurité, autant pour l'authentification que pour la protection des données.
- Leur performance (rapidité, fluidité).
- La qualité de l'expérience utilisateur, notamment la simplicité du parcours de commande.
- Les fonctionnalités proposées : gestion des services, suivi des commandes, notifications.

Conclusions du benchmark

Points forts

Uber se démarque par une utilisation fluide et intuitive.

Upwork et Fiverr inspirent confiance avec leur sécurité et leur professionnalisme.

Glovo propose un parcours utilisateur vraiment simple, ce qui rend l'expérience agréable.

Points faibles

Le processus d'inscription prend parfois trop de temps. Certains utilisateurs débutants se perdent dans des interfaces trop complexes.

Plan d'action

Sur la base de ce benchmark, voici les objectifs à atteindre : Réduire les étapes pour finaliser une commande. Renforcer la sécurité avec l'authentification JWT et la double authentification

Priorités

Optimiser les performances du backend ASP.NET Core.

Rendre l'expérience mobile plus fluide et agréable.

Améliorer le système de notifications et le suivi des commandes.

<i>Critères</i>	<i>OnDemandApp</i>	<i>Uber</i>	<i>Glovo</i>	<i>Fiverr</i>	<i>Upwork</i>
<i>Temps de chargement</i>	~1.3 s	1.2 s	1.5 s	1.8 s	1.7 s
<i>Processus de commande</i>	3 étapes	2 étapes	3 étapes	4 étapes	4 étapes
<i>Présence du 2FA</i>	Oui	Oui	Non	Oui	Oui
<i>Authentification JWT</i>	Oui	Oui	Oui	Oui	Oui
<i>Expérience utilisateur (/5)</i>	4.5	4.7	4.4	4.2	4.3
<i>Navigation intuitive</i>	Oui	Oui	Oui	Moyen	Moyen
<i>Sécurité globale</i>	Très forte	Très forte	Moyenne	Forte	Forte

Tableau 2 : Comparaison des performances et fonctionnalités des plateformes concurrentes selon plusieurs critères de benchmark.

6. Missions du projet

Pour développer l'application mobile OnDemandApp, qui cible les services à la demande, on a mené plusieurs missions clés du début à la fin du projet.

6.1 Rédaction de l'étude de faisabilité

On a commencé par analyser les besoins des utilisateurs et les contraintes techniques du projet. L'idée, c'était de trouver une solution qui tienne la route et qui colle aux objectifs. On a surtout creusé les questions de sécurité : comment authentifier les utilisateurs, protéger les données, gérer les accès. On a aussi regardé ce que proposent les autres plateformes de services à la demande, histoire de voir où on se situe.

6.2 Rédaction de l'étude fonctionnelle

À cette étape, on a défini les grandes fonctionnalités de l'application : gestion des utilisateurs, publication et consultation des services, gestion des commandes, suivi des interactions entre clients et prestataires. On a aussi tracé les parcours des utilisateurs et les flux fonctionnels pour chaque profil : client, prestataire, administrateur. Ça permet de s'assurer que chacun trouve ce dont il a besoin, sans se perdre.

6.3 Réalisation des tests et validations

On a testé chaque module, que ce soit l'authentification, la gestion des services ou la gestion des commandes. On a aussi passé l'application à la loupe avec des tests de performance, de sécurité—les bases, mais indispensables—and d'expérience utilisateur. Quand on a repéré des bugs ou des failles, on les a corrigés pour rendre le système plus stable et plus performant.

6.4 Rédaction du guide utilisateur

On a rédigé un guide utilisateur simple et bien structuré, pour aider chacun à prendre en main l'application. Ce guide explique les principales fonctionnalités, la création de compte, l'authentification (y compris le 2FA), et détaille comment naviguer dans l'application sans se compliquer la vie.

6.5 Sensibilisation des utilisateurs

On a organisé des sessions de démonstration pour montrer comment fonctionne l'application. C'était aussi l'occasion d'insister sur les bonnes pratiques de sécurité : utiliser le 2FA, bien gérer son compte, protéger ses infos personnelles. Les utilisateurs repartent avec les clés pour utiliser l'application en toute sécurité.

7. Livrables

Le tableau ci-dessous présente les différents livrables réalisés dans le cadre du projet *OnDemandApp*, accompagnés d'une description détaillée de chacun, afin de donner une vision claire de la méthodologie adoptée et des étapes suivies tout au long du projet.

Livrable	Description	Date
Plan d'assurance qualité	Document décrivant les méthodes, outils et processus mis en œuvre pour garantir la qualité du développement de l'application mobile et du backend sécurisé.	12/12/2025
Cahier des charges	Document définissant les besoins, les fonctionnalités attendues (gestion des services, commandes, utilisateurs) ainsi que les contraintes techniques et organisationnelles du projet.	16/12/2025
Dossier de spécifications fonctionnelles	Document détaillant l'ensemble des fonctionnalités de l'application (authentification sécurisée, gestion des services, commandes, notifications).	18/12/2025
Dossier de spécifications techniques	Document décrivant l'architecture technique du projet, le choix des technologies (React Native, Expo, ASP.NET Core, SQL Server)	20/12/2025
Application mobile (prototype)	Première version fonctionnelle de l'application mobile permettant l'inscription, la connexion sécurisée et la gestion de base des services et commandes.	24/12/2025
Rapport de tests et validation	Document présentant les résultats des tests unitaires, fonctionnels, de performance et de sécurité réalisés sur l'application.	30/12/2025
Guide utilisateur	Manuel expliquant l'utilisation de l'application mobile : création de compte, authentification avec 2FA, navigation et gestion des commandes.	02/01/2026
Rapport final de projet	Rapport détaillé retracant toutes les étapes du projet, depuis l'analyse initiale jusqu'à la réalisation, incluant les choix techniques, les résultats obtenus et les perspectives d'amélioration.	04/01/2026

Tableau 3 : Liste des livrables du projet avec leur description et leurs dates de livraison.

8. Méthodologie de gestion de projet

Pour assurer un développement efficace de l'application mobile OnDemandApp, nous avons choisi le modèle en Y. Ce modèle colle bien aux projets informatiques d'aujourd'hui, et surtout aux applications mobiles. Il permet de mener de front la conception et la validation, tout en gardant un œil constant sur la qualité et la sécurité du système.

Dans le monde des applications mobiles, le modèle en Y s'impose. Les projets demandent des vérifications fréquentes, une validation progressive des aspects techniques, fonctionnels et sécuritaires. Cette méthode facilite l'intégration continue des fonctionnalités, tout en s'assurant qu'elles restent fidèles aux besoins définis au départ.

8.1 Présentation du modèle en Y

Le modèle en Y s'articule autour de deux branches principales :

- La branche gauche couvre l'analyse, la spécification fonctionnelle, puis la conception technique de l'application.
- La branche droite regroupe les tests unitaires, les tests d'intégration, la validation fonctionnelle et les tests de sécurité.
- Au croisement des deux branches, on retrouve la phase clé : codage et mise en œuvre du système.

Pour chaque étape de la conception, on associe une phase de test. Cela garantit que chaque fonctionnalité développée répond bien aux besoins exprimés dès le lancement du projet.

8.2 Étude de faisabilité et analyse des besoins

Cette étape nous a permis de cerner les attentes des utilisateurs et les exigences du marché des services à la demande. On a aussi analysé les contraintes du projet — qu'elles soient fonctionnelles, techniques ou sécuritaires — avec une attention particulière sur la gestion des utilisateurs, des commandes, des services, et la protection des données personnelles.

8.3 Spécification fonctionnelle

Ici, on entre dans le détail. La spécification fonctionnelle fixe précisément quelles fonctionnalités intégrer dans l'application :

- Inscription et connexion sécurisées pour les utilisateurs.
- Gestion des rôles : client, prestataire, administrateur.
- Recherche et consultation des services.
- Passage et suivi des commandes.
- Notifications en temps réel.

On a structuré ces fonctionnalités sous forme de parcours utilisateurs clairs, pensés pour optimiser l'expérience dès la première utilisation.

8.4 Conception technique

La conception technique a défini l'architecture logicielle de l'application, les interactions entre le frontend mobile, le backend et la base de données. On a utilisé des diagrammes UML pour modéliser le système et montrer comment ses composants dialoguent. Dans cette phase, on a aussi précisé les mécanismes de sécurité : flux d'authentification, gestion des sessions via JWT, authentification à deux facteurs (2FA).

8.5 Mise en œuvre

Durant la mise en œuvre, nous avons développé l'application mobile avec React Native et Expo, puis construit le backend sécurisé sur ASP.NET Core.

On a intégré l'authentification JWT pour gérer les sessions utilisateurs, mis en place le 2FA, géré les services et les commandes, et assuré la communication sécurisée entre le frontend mobile et l'API.

8.6 Tests et validation

On n'a pas attendu la fin pour tester. Les tests ont eu lieu tout au long du développement, pour garantir la qualité et la fiabilité de l'application :

- Tests unitaires sur chaque composant ou fonctionnalité.

- Tests d'intégration pour s'assurer que le mobile et le backend communiquent correctement.
- Tests de validation pour vérifier que l'application répond bien aux besoins fonctionnels.
- Tests de sécurité afin de garantir l'efficacité des protections de données et des mécanismes d'authentification.

8.7 Progression du projet selon le modèle en Y

Tout au long du projet OnDemandApp, nous avons suivi les étapes du modèle en Y, en maintenant le lien entre conception et test. Cette façon de faire a permis de détecter rapidement les anomalies, d'améliorer la qualité du système, et de livrer une application mobile fiable, sécurisée, parfaitement en phase avec nos objectifs.

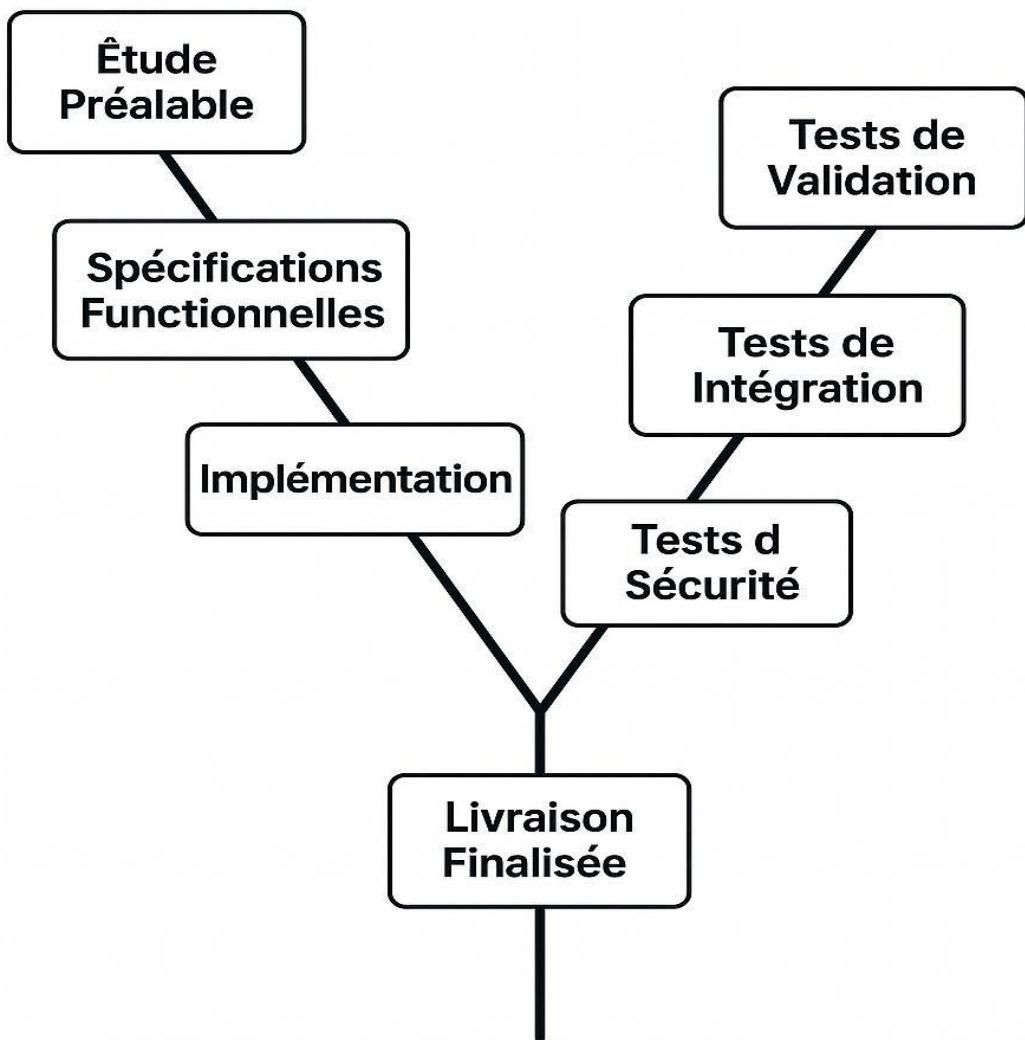


Figure 1 : Modélisation Y

9. Diagramme de Gantt

Le diagramme de Gantt ci-dessus illustre la planification temporelle des différentes phases du projet de développement de l'application, réparties entre le 12 décembre et le 04 janvier. Il permet de visualiser clairement l'enchaînement et le chevauchement des tâches clés, tout en facilitant le suivi de l'avancement.

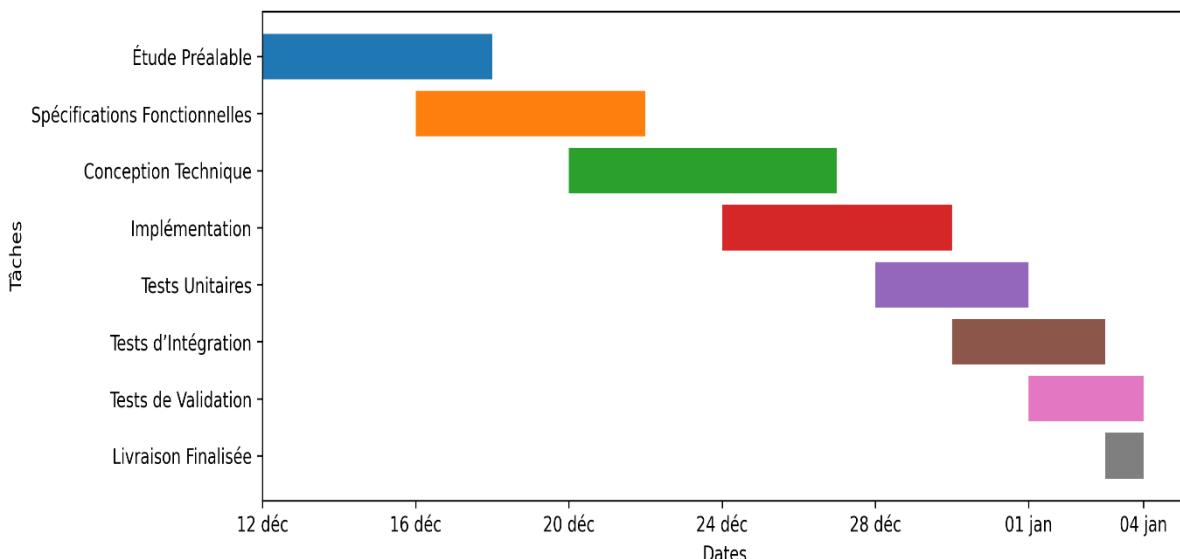


Figure 2 : Diagramme de Gantt illustrant la planification des tâches du, de l'étude préalable à la livraison finale

7. Conclusion

En résumé, ce premier chapitre de notre projet de fin d'études a permis de présenter de manière claire le **contexte général des applications mobiles de services à la demande**, ainsi que les enjeux majeurs liés à la **sécurité**, à la **performance** et à la **qualité de l'expérience utilisateur**. Il a également mis en évidence les motivations ayant conduit au développement de *OnDemandApp*, une application mobile moderne visant à faciliter la mise en relation entre clients et prestataires de services.

Les objectifs techniques et fonctionnels définis dans ce chapitre traduisent notre volonté de concevoir une solution **robuste, sécurisée et conviviale**, capable de répondre aux exigences actuelles du marché. Ce projet constitue ainsi une opportunité concrète de mettre en pratique les compétences acquises au cours de notre formation, tout en relevant les défis technologiques et organisationnels inhérents au développement d'une application mobile professionnelle.

Chapitre 2 : Analyse et conception

1. Introduction

Au Maroc, la transformation digitale s'accélère nettement. Les smartphones sont partout, les applications mobiles s'invitent dans la vie de tous les jours, et les plateformes de services à la demande sont devenues incontournables. Elles simplifient l'accès à une multitude de services et rendent l'expérience utilisateur plus fluide. Cette évolution ne tombe pas du ciel. Elle s'appuie sur une meilleure connectivité Internet, l'accès généralisé aux technologies mobiles, et un vrai engouement pour les services numériques, surtout chez les jeunes.

Mais tout n'est pas simple. Le développement de ces plateformes au Maroc pose des défis très concrets : la sécurité des données, la confiance des utilisateurs, la gestion des relations entre clients et prestataires, et l'optimisation des performances des applications. C'est dans ce contexte que nous avons conçu OnDemandApp, une application mobile pensée pour le marché marocain, mais alignée sur les standards internationaux de conception, de sécurité et de qualité logicielle.

Ce chapitre va droit au but : on analyse et on conçoit le système. On y présente les outils et méthodes de modélisation, avec un focus particulier sur UML, un langage central pour comprendre, structurer et documenter le projet.

2. Le langage UML : un outil clé pour la modélisation des systèmes

2.1 Introduction

Les systèmes informatiques et les applications mobiles deviennent chaque année plus complexes. Il faut des outils solides pour représenter, analyser et documenter tout ça de façon claire. C'est là qu'UML (Unified Modeling Language) entre en scène. UML propose une approche visuelle, standardisée, pour modéliser les systèmes logiciels.

Dans le projet OnDemandApp, on a utilisé UML pour représenter les différentes parties du système, les interactions entre les acteurs, et les flux aussi bien fonctionnels que techniques. Cette démarche améliore la compréhension globale et réduit les zones d'ombre lors de la conception et du développement.

2.2 Définition et objectifs du langage UML

UML, c'est un langage de modélisation visuelle standardisé. Son but : offrir un moyen de communication commun à tous les intervenants d'un projet logiciel — développeurs, concepteurs, chefs de projet. Né dans les années 1990 d'une fusion de plusieurs méthodes orientées objet, UML cherche à unifier et standardiser la modélisation.

UML, riche au niveau syntaxe et sémantique, permet de décrire à la fois la structure d'un système (composants, relations) et son comportement (interactions, enchaînement des actions). Il ne s'arrête pas à la conception logicielle : on peut aussi s'en servir pour l'analyse fonctionnelle, la conception technique, et la documentation des processus métier.

Dans ce projet, UML joue un rôle clé pour modéliser les fonctionnalités de OnDemandApp, clarifier les rôles des acteurs (client, prestataire, administrateur), et définir les interactions entre l'application mobile et le backend.

2.3 Un langage visuel universel

Comme un plan d'architecte ou un schéma industriel, les diagrammes UML donnent une vue d'ensemble, visuelle, des éléments d'un système et des liens qui les unissent. Le but est simple : permettre à toute l'équipe de partager une même compréhension du système, repérer les incohérences, et améliorer la qualité de la planification et de la conception.

UML propose plusieurs types de diagrammes, regroupés en deux grandes familles :

Diagrammes de structure : ils montrent l'organisation statique du système, classes, composants, architecture globale (diagrammes de classes, de composants, de déploiement).

Diagrammes comportementaux : ils modélisent le comportement du système et les interactions entre acteurs et composants (diagrammes de cas d'utilisation, de séquence, d'activité).

On utilisera ces diagrammes dans les sections suivantes pour modéliser l'architecture et le fonctionnement précis de OnDemandApp, et livrer une vision claire et structurée du système.

3. Diagramme des activités

Ce diagramme montre comment les différents acteurs : client, prestataire, système qui interagissent dans OnDemandApp, depuis la création de la commande jusqu'à la fin de la mission. On saisit tout de suite le déroulement des actions, les choix à chaque étape, et les statuts qui évoluent au fil de la mission.

Le client

Le client commence par s'authentifier de façon sécurisée. Une fois connecté, il cherche un service selon ses besoins, puis crée sa commande. Dès que c'est fait, il reçoit des notifications en temps réel et suit la progression de la mission sur la carte grâce au GPS. Quand la mission se termine, il paie la prestation et peut ensuite noter le prestataire. Ce retour aide à renforcer la qualité globale des services proposés sur la plateforme.

Le système

Le système organise tout. Il enregistre la commande avec le statut "Pending", repère les prestataires les plus proches et leur envoie des notifications. Quand ceux-ci répondent, le système attribue la mission à celui qui est disponible. Il gère aussi chaque changement de statut (En route, In progress, Done), génère la facture, et enregistre le paiement.

Le prestataire

Dès qu'une mission lui est proposée, le prestataire reçoit une notification. Il accepte ou refuse. S'il accepte, il part chez le client, envoie des pings GPS pour le suivi en temps réel, puis réalise les travaux demandés. Une fois la mission terminée, il la marque comme achevée.

Points d'intérêt

La gestion claire des statuts permet un suivi précis et transparent de chaque mission. Le GPS en temps réel rend l'expérience plus fluide et rassurante pour tout le monde. En cas de refus, le système relance automatiquement la recherche d'un autre prestataire, ce qui garantit la continuité du service.

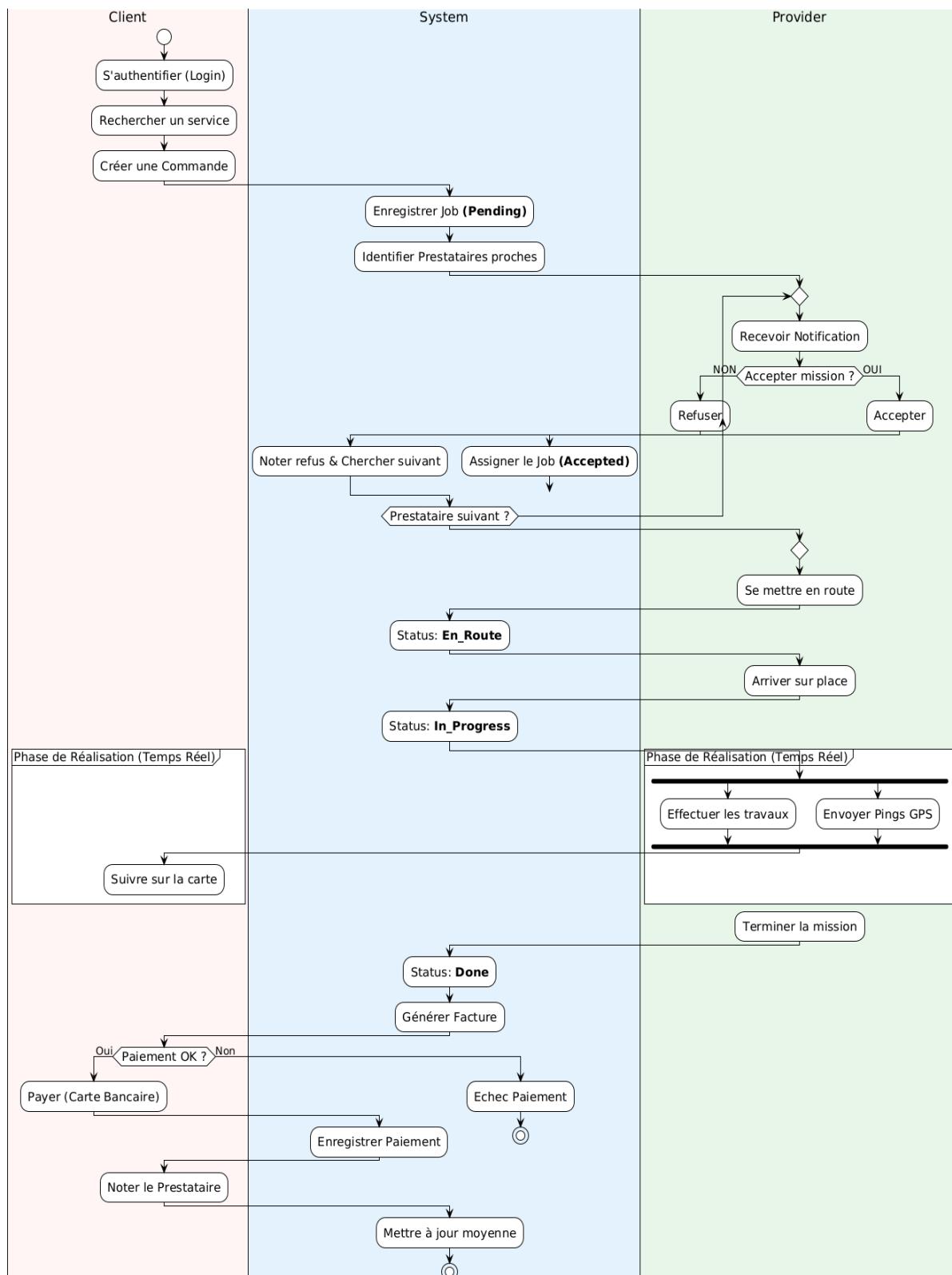


Figure 3 : Diagramme d'activité représentant les interactions entre les acteurs du système

4. Diagramme des cas d'utilisation

Description

Le diagramme des cas d'utilisation liste les fonctionnalités principales de OnDemandApp pour chaque acteur : client, prestataire, administrateur. Il met en lumière les interactions entre ces utilisateurs et la plateforme, et montre concrètement les services proposés.

Acteurs et fonctionnalités

Client

Le client peut :

- S'authentifier sur la plateforme.
- Parcourir le catalogue, filtrer les services, lancer des recherches.
- Créer une commande et régler la prestation.
- Suivre la mission en direct via le GPS.
- Discuter grâce au chat intégré.
- Noter le prestataire une fois la mission réalisée.

Prestataire

Le prestataire a la possibilité de :

- S'authentifier.
- Voir les missions qui lui sont attribuées.
- Envoyer des pings GPS pour le suivi.
- Gérer sa disponibilité, ses zones d'intervention, ses services et tarifs.

Administrateur

L'administrateur gère la plateforme depuis un espace dédié. Il peut :

- Accéder à un tableau de bord avec les KPIs.
- Gérer le catalogue de services.
- Administrer les comptes clients et prestataires.

Extensions et relations

Certaines fonctionnalités, comme l'authentification, sont indispensables avant d'accéder aux autres services. D'autres, plus avancées — paiement, suivi de mission — s'intègrent naturellement dans le parcours utilisateur.

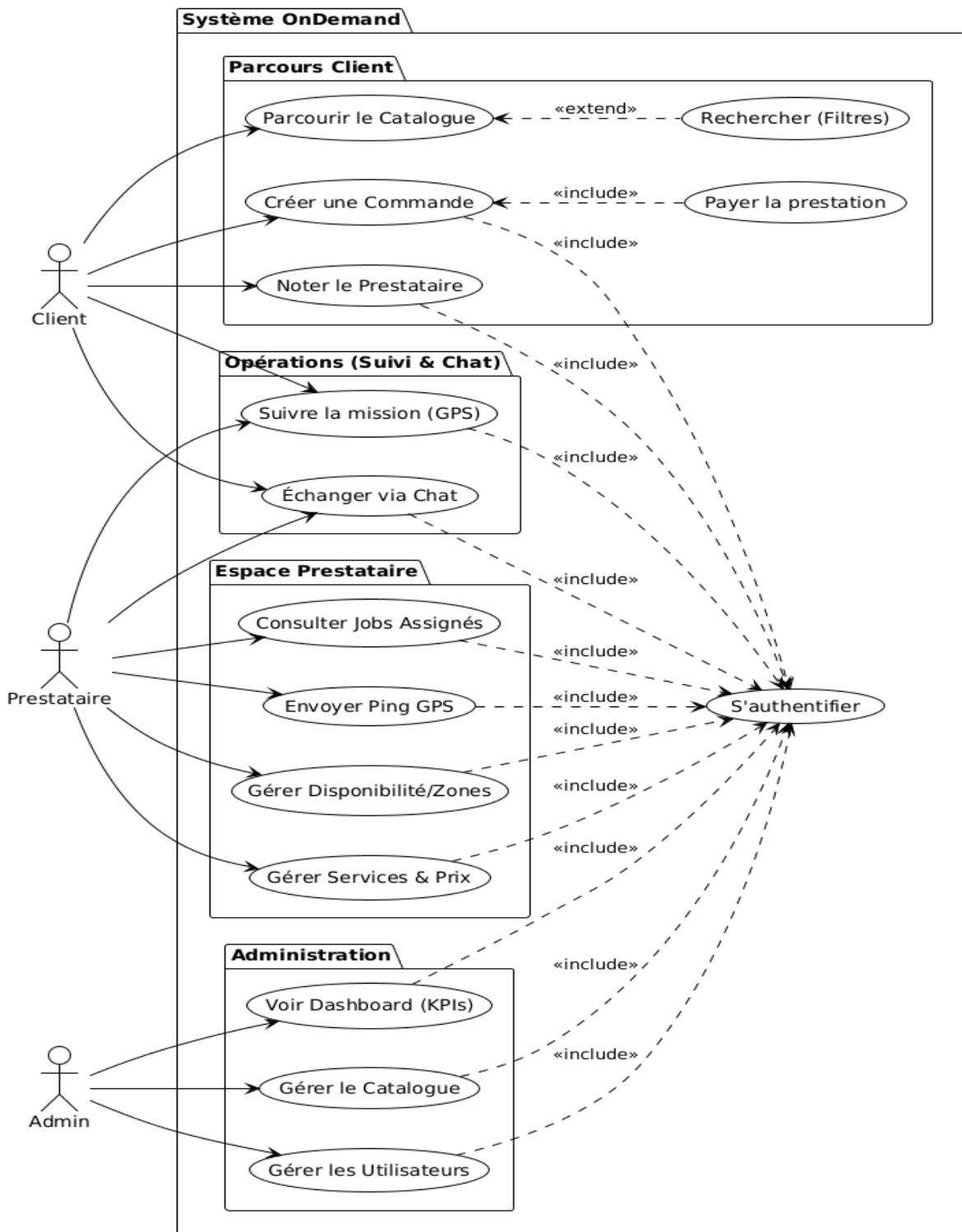


Figure 4 : Cas d'utilisation montrant les fonctionnalités principales accessibles aux utilisateurs et à l'admin.

5. Diagramme de classes

Le diagramme de classes pose les bases de la structure statique du système OnDemandApp. On y voit les grandes entités du système, leurs attributs, et comment elles se relient les unes aux autres. C'est un passage obligé pour concevoir la base de données et le backend.

Structure générale

Le système s'articule autour de plusieurs modules fonctionnels. Chacun gère une partie bien précise, ce qui clarifie les responsabilités.

Module Authentification

Ici, tout tourne autour de la gestion des utilisateurs et de la sécurité :

- Utilisateur : la classe centrale. Chaque utilisateur a un identifiant, une adresse email, un rôle, une note moyenne.
- Client, Prestataire, Admin : ces classes héritent d'Utilisateur. Elles incarnent les différents rôles possibles.
- CodeAuth : gère les codes d'authentification temporaires, essentiels pour valider et sécuriser les accès.

Des énumérations définissent les rôles et les statuts pour que les états du système restent cohérents.

Module Catalogue

Ce module organise les services proposés sur la plateforme :

- CategorieService : chaque catégorie regroupe des services similaires.
- ItemCatalogue (ServiceItem) : représente les services visibles dans le catalogue.
- OffreService : fait le lien entre un service et un prestataire, en précisant un prix de base et une durée estimée.
- ProfilPrestataire : stocke les infos propres au prestataire, comme la disponibilité, le rayon d'action, la catégorie de services.

Module Opérations

On retrouve ici tout ce qui touche à l'exécution des missions :

- Mission (Job) : une demande de prestation par un client, avec statut, prix final, date prévue.
- Service (Ops) : détaille le service choisi pour une mission.

- Paiement : gère tout ce qui concerne le paiement d'une mission.
 - Message : permet au client et au prestataire de discuter via le chat.
 - PingGPS : enregistre les coordonnées GPS envoyées en temps réel par le prestataire.
- Objectif du diagramme de classes

Ce diagramme offre une vue claire du modèle de données. Il sert à concevoir la base de données relationnelle, à mettre en place les entités dans le backend, et à assurer la cohérence entre la logique métier et la technique.

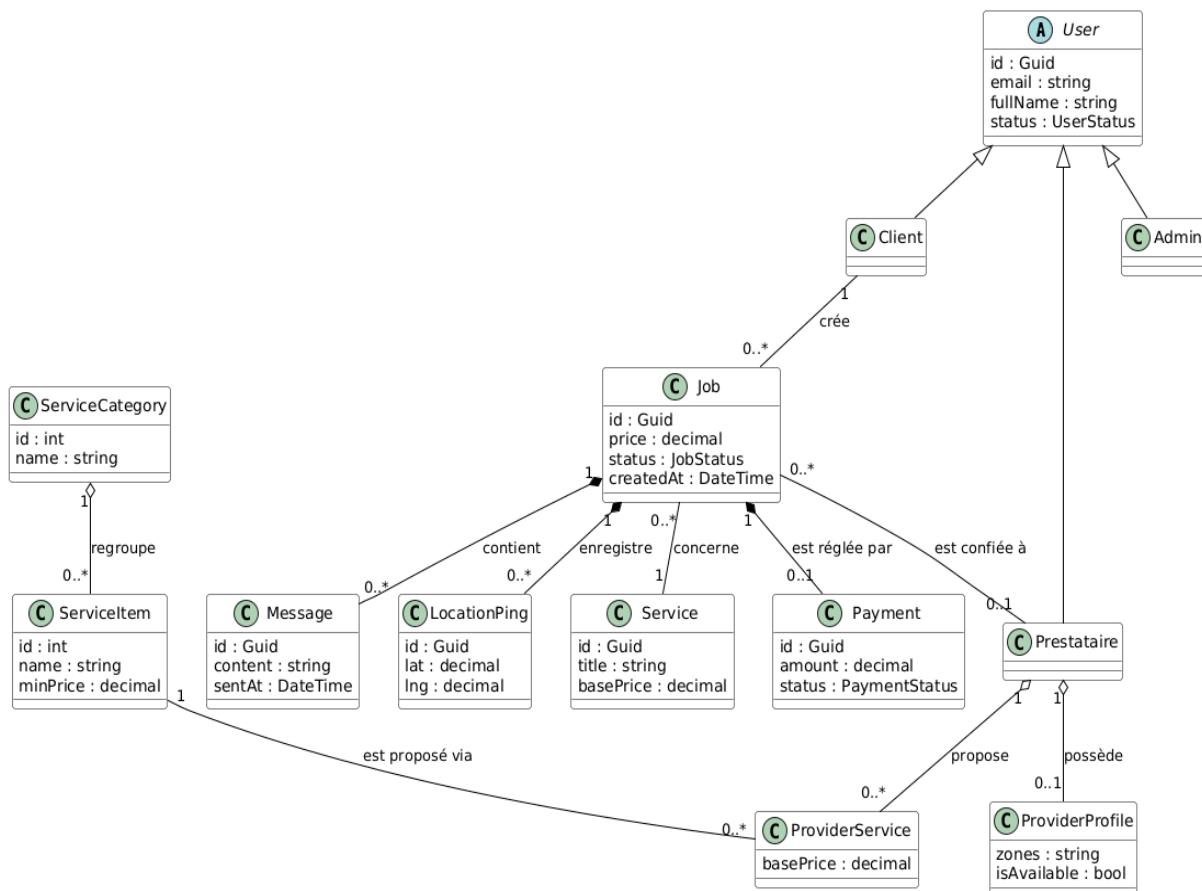


Figure 5 : Diagramme de classes décrivant les entités principales du système et leurs relations.

6. Diagrammes de séquence

Les diagrammes de séquence montrent comment le système se comporte en situation. On suit l'ordre des échanges entre les différents acteurs, interfaces et le cœur du système.

6.1 Diagramme de séquence : Administrateur

Ce scénario met l'accent sur la supervision et la gestion par un administrateur.

Description du scénario

L'administrateur se connecte au portail d'administration et passe par une authentification sécurisée. Le système vérifie son rôle, puis génère un token JWT pour lui ouvrir l'accès au tableau de bord.

Une fois connecté, l'administrateur peut :

- consulter les statistiques globales (utilisateurs, missions, paiements),
- gérer le catalogue des services,
- voir la liste des utilisateurs,
- suspendre ou modérer des comptes si besoin.

Le système central valide chaque action et enregistre tout dans la base de données.

Objectif

Ce diagramme met en avant le contrôle des accès, la centralisation des décisions, et la supervision d'ensemble de la plateforme.

Scénario Admin : Supervision et Gestion

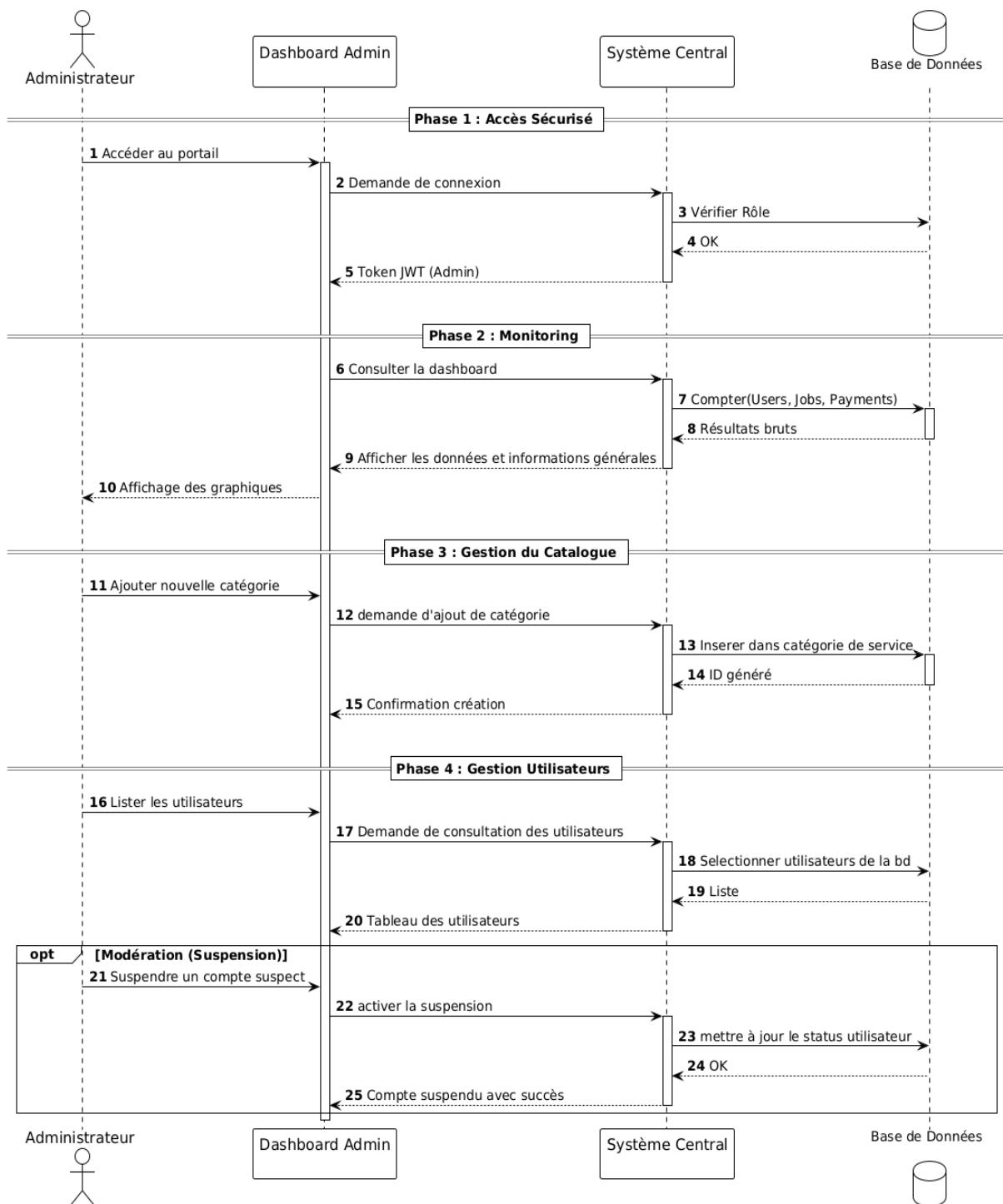


Figure 6 : Séquence d'interactions entre l'administrateur et le système à travers les différentes fonctionnalités .

6.2 Diagramme de séquence : Prestataire

Ici, le diagramme retrace tout le parcours d'une mission côté prestataire, de la connexion à la clôture.

Description du scénario

Le prestataire s'authentifie d'abord, puis active son statut de disponibilité. Dès qu'une mission adaptée se présente, il reçoit une notification et peut accepter la mission.

Après acceptation :

- le système assigne la mission et réserve le prestataire,
- il met à jour son statut (En route, puis In progress),
- le système envoie des pings GPS régulièrement pour assurer le suivi en temps réel,
- la messagerie permet d'échanger avec le client.

Quand la mission se termine, le prestataire la clôture et attend la confirmation du paiement.

Objectif

Ce diagramme illustre la gestion en temps réel des missions et la synchronisation permanente entre le prestataire et le système.

Scénario Prestataire : De la Disponibilité à la Clôture

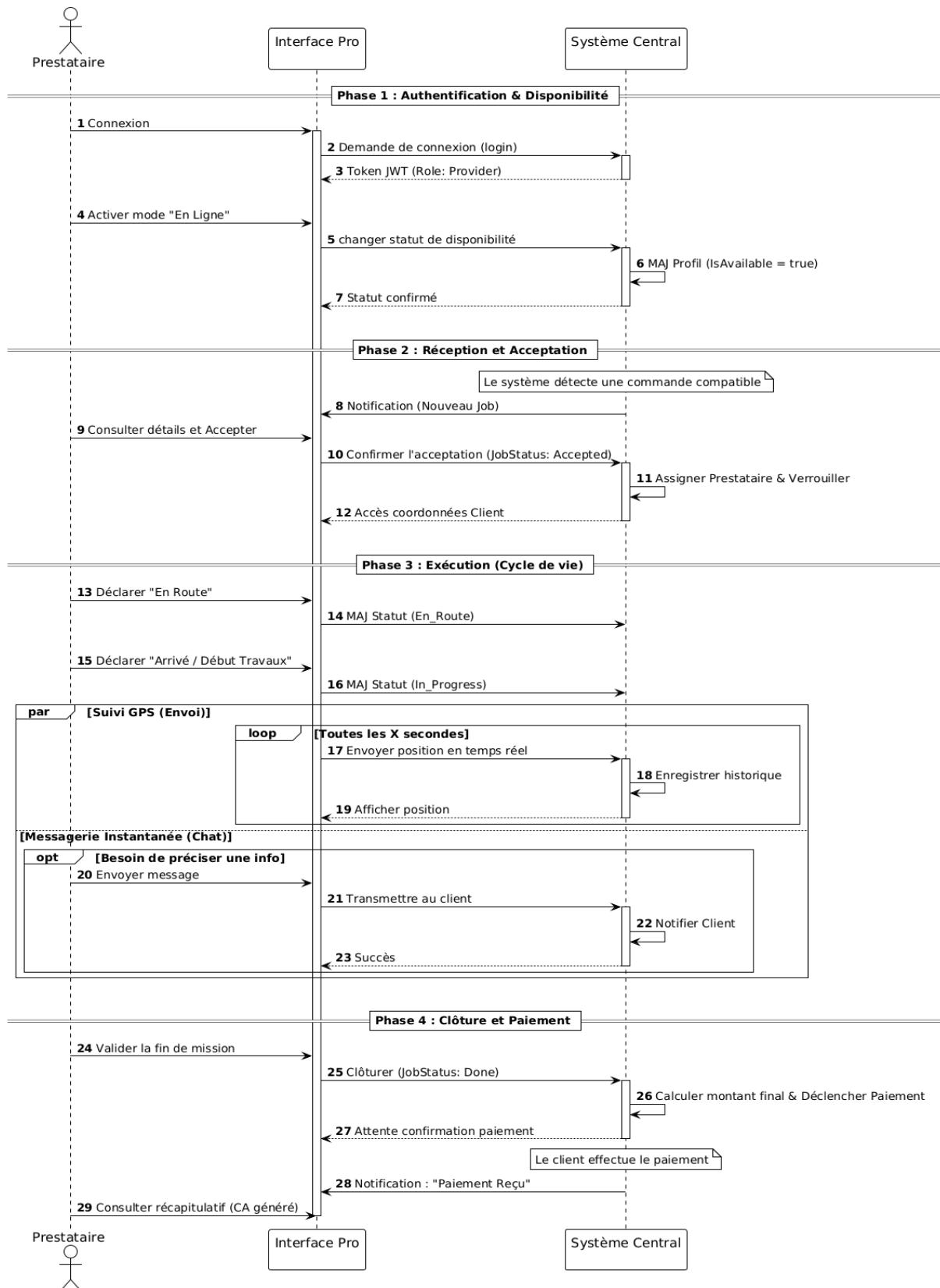


Figure 7 : Séquence d'interactions entre le prestataire et le système à travers les différentes fonctionnalités .

6.3 Diagramme de séquence :Client

Ce diagramme retrace le parcours du client du début à la fin, de l'authentification jusqu'à la notation du prestataire.

Scénario

Le client commence par s'authentifier à l'aide d'un système sécurisé : JWT, parfois 2FA. Après connexion, il cherche un service, parcourt le catalogue, puis crée une mission.

Pendant la mission, il garde un œil sur la position du prestataire en temps réel. Il échange aussi via le chat intégré si besoin.

Une fois la mission terminée, le client règle le paiement, reçoit une confirmation, puis note le prestataire. Le système met à jour la moyenne du prestataire.

Objectif

Ce diagramme montre comment l'expérience client reste fluide et sécurisée, en intégrant paiement et système d'évaluation.

Scénario Client Complet

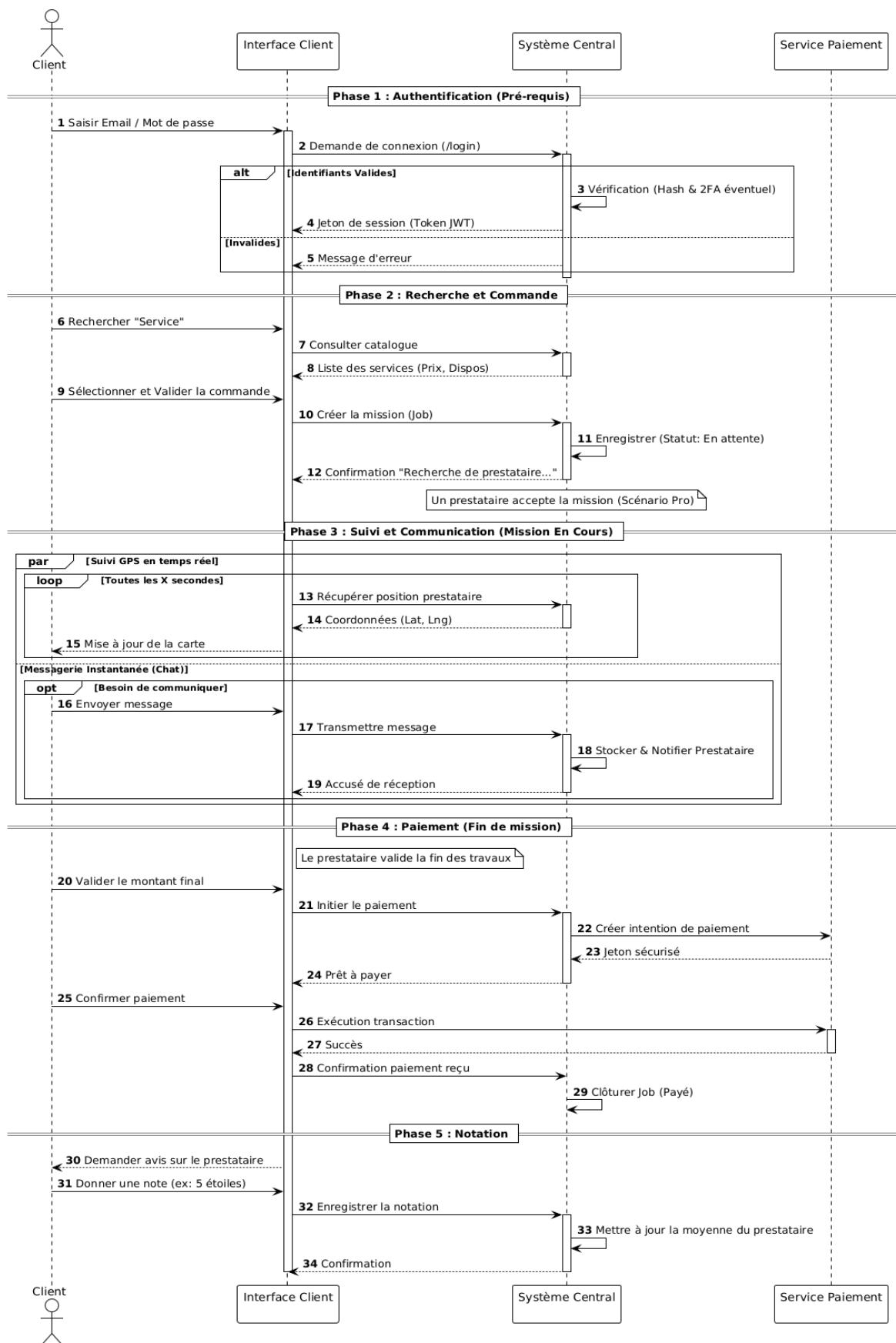


Figure 8 : Séquence d'authentification client avec, navigation dans les services

7. Conclusion

Cette phase d'analyse et de modélisation des flux et des processus fonctionnels de la plateforme **OnDemandApp** a permis de structurer de manière rigoureuse le fonctionnement global du système à travers l'utilisation de différents **diagrammes UML et schémas techniques**. Ces outils de modélisation ont joué un rôle central dans la compréhension, la formalisation et la validation des besoins fonctionnels et techniques du projet.

Les principaux points mis en évidence sont les suivants :

Modélisation intégrale du système

- Le **diagramme de classes** a permis de définir clairement l'architecture statique du système, en identifiant les principales entités métiers telles que *Utilisateur*, *Client*, *Prestataire*, *Mission (Job)*, *Paiement* et leurs relations. Cette modélisation assure une base solide pour la conception de la base de données et l'implémentation du backend.
- Les **diagrammes de cas d'utilisation** ont mis en évidence les interactions entre les différents acteurs (client, prestataire, administrateur) et les fonctionnalités offertes par la plateforme, telles que l'authentification sécurisée, la gestion des missions, le suivi en temps réel, le paiement et l'évaluation des prestataires.

Modélisation dynamique des processus

- Les **diagrammes de séquence** ont permis de décrire de manière détaillée les flux dynamiques du système, notamment les scénarios clés côté client, prestataire et administrateur.
- Ces diagrammes illustrent clairement les échanges entre les différentes composantes du système (interfaces, système central, services de paiement), en mettant en évidence les mécanismes critiques tels que l'authentification ,la gestion du cycle de vie des missions, le suivi GPS en temps réel et la clôture avec paiement.

Chapitre 3 :

Réalisation

1. Introduction

La mise en œuvre, c'est le cœur du projet. C'est là qu'on quitte les schémas, les listes de besoins et les choix techniques pour passer à la réalité : l'application mobile OnDemandApp prend forme. On transforme les idées et les exigences du cahier des charges en un système concret, qui fonctionne, qui est sécurisé, et qui répond vraiment à ce qu'on attend.

Dans cette partie, je détaille comment on a procédé, étape par étape. On commence par configurer tout l'environnement de développement. Ensuite, on attaque le codage des fonctions principales et on intègre toutes les sécurités nécessaires. À chaque étape, on s'est appuyé sur les meilleures pratiques du développement moderne, que ce soit pour la fonctionnalité ou pour la cybersécurité.

2. Architecture de la solution

On a choisi une architecture modulaire et sécurisée, pensée pour coller parfaitement aux exigences du projet. Cette structure sépare clairement les différentes couches du système, ce qui rend l'application plus facile à maintenir, à faire évoluer, et plus robuste, tout simplement.

2.1 Architecture générale

Notre solution s'appuie sur une architecture client–serveur en trois couches :

Couche de présentation (Front-End)

Ici, c'est l'application mobile développée avec React Native. On voulait une interface moderne et fluide, facile à prendre en main, que ce soit pour les clients ou les prestataires. L'expérience reste cohérente, peu importe la plateforme mobile.

Couche métier (Back-End)

Toute la logique applicative tourne sur un backend .NET, exposé en API REST. Cette couche gère les utilisateurs, les missions, les paiements, les notifications. Côté sécurité, on a mis en

place l'authentification JWT, la gestion des rôles, et la validation des données qui transitent entre client et serveur.

Couche de données (Base de données)

On s'appuie sur une base de données relationnelle (SQL Server) pour stocker toutes les infos : utilisateurs, missions, paiements, historiques. Les accès sont bien contrôlés, les données sensibles protégées contre les attaques comme l'injection SQL. On suit les bonnes pratiques de sécurisation des données.

2.2 Esquisse de l'architecture

En résumé, l'architecture se présente comme un schéma où l'app React Native discute avec l'API .NET, qui elle-même dialogue avec la base de données relationnelle.

Points clés côté sécurité :

- Authentification sécurisée avec JWT (JSON Web Tokens)
- Gestion fine des rôles (Client, Prestataire, Admin)
- Validation systématique des accès aux ressources sensibles

3. Outils et technologies utilisés

La solution repose sur une **architecture modulaire, évolutive et sécurisée**, conçue pour répondre aux exigences fonctionnelles, techniques et de cybersécurité définies dans le cahier des charges. Cette architecture permet une séparation claire des responsabilités entre les différentes couches du système, facilitant la maintenance et l'évolution de l'application.

3.1 Architecture générale

La solution adopte une architecture **client–serveur à trois niveaux** :

Couche de présentation (Front-End mobile)

Cette couche correspond à l'application mobile développée avec **React Native**, en s'appuyant sur **Expo** afin de simplifier la configuration, le développement et le déploiement. Expo permet un accès facilité aux fonctionnalités natives du smartphone telles que le GPS, les notifications

et la caméra, tout en garantissant une expérience utilisateur fluide et cohérente sur différentes plateformes mobiles.

Couche métier (Back-End)

La couche métier est implémentée sous forme d'une **API REST développée en .NET**. Elle assure la gestion de la logique applicative, des utilisateurs, des missions, des paiements et des notifications. L'API intègre des mécanismes de sécurité basés sur **JWT**, la gestion des rôles et la validation des données échangées entre le client mobile et le serveur.

Couche de données (Base de données)

La couche de données repose sur une base de données relationnelle assurant le stockage structuré et sécurisé des informations relatives aux utilisateurs, aux missions et aux transactions. Des mécanismes de contrôle d'accès et de protection contre les injections sont mis en place afin de garantir l'intégrité et la confidentialité des données.

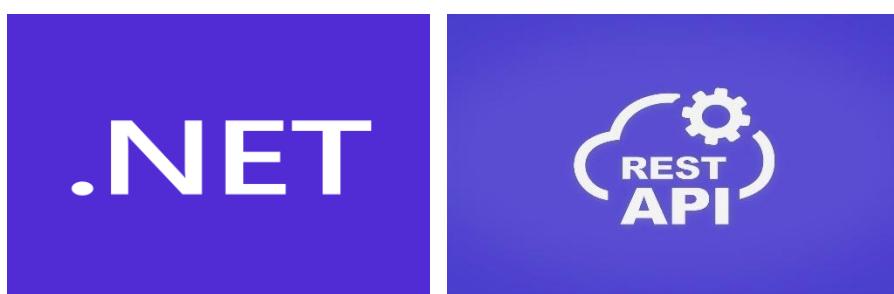
4. Outils et technologies utilisés

React Native : Front-End mobile



On a choisi React Native parce qu'il permet de développer rapidement une app mobile performante avec un seul code pour Android et iOS. L'interface reste fluide, réactive, et s'adapte bien aux besoins des deux types d'utilisateurs.

.NET – Back-End (API REST)



Le framework .NET nous donne une architecture solide et sécurisée pour l'API REST. On gère facilement la logique métier, l'authentification JWT, et tous les échanges sécurisés entre le front-end et le back-end.

JavaScript :Logique et interactivité



JavaScript gère le côté dynamique de l'app mobile. Il booste l'expérience utilisateur et gère la communication avec l'API.

Base de données relationnelle (SQL Server)



La base de données SQL Server structure clairement toutes les données et garantit leur intégrité : utilisateurs, missions, paiements, tout est bien organisé et sécurisé.

JWT (JSON Web Token) : Authentification



JWT offre une authentification solide, sans état côté serveur, parfaite pour les architectures modernes et les apps mobiles. Ça améliore aussi les performances et la montée en charge.

Authentification via firebase



L'authentification via Firebase a été intégrée afin de renforcer la sécurité et la fiabilité de l'accès à l'application. Elle permet la gestion sécurisée des comptes utilisateurs à travers la vérification par email et l'authentification centralisée. Firebase assure également une synchronisation en temps réel de l'état de connexion et une protection contre les accès non autorisés. Cette solution facilite la gestion des sessions tout en améliorant l'expérience utilisateur grâce à une authentification rapide et robuste.

Git & GitHub : Gestion de versions



Avec Git, nous pouvions suivre l'évolution du code, travailler en équipe et reproduire une version stable en cas de problème. GitHub a permis la centralisation, la collaboration et l'archivage du projet.

Visual Studio Code : Environnement de développement



Visual Studio Code, c'est notre EDI principal. Léger, extensible, performant, il supporte parfaitement React Native, .NET, et Git. Ça nous a permis de gagner en efficacité, tout au long du développement.

4. Interfaces de l'application

4.1 Interface de l'utilisateur

La page «HOME PAGE»

La page d'accueil, c'est vraiment le point de départ de l'application mobile OnDemandApp pour chaque client. Ici, l'utilisateur choisit sa ville, lance une recherche en un clin d'œil grâce à une barre vraiment intuitive, et jette un œil aux catégories de services les plus populaires. Juste sous les yeux, on trouve des chiffres clés : combien d'experts, de services disponibles, de clients actifs ? Ça rassure, ça crédibilise la plateforme. Une bannière promo attire l'attention sur les offres du moment de quoi donner envie de passer une première commande. L'interface vise la clarté, la rapidité, l'action, rien n'est là pour ralentir l'accès aux services.



Figure 9 : Interface mobile – Page d'accueil client avec recherche de services et statistiques globales.

La page « Catégories & Services »

Sur cette page, l'utilisateur navigue à travers les grandes familles de services : beauté et bien-être, ménage, plomberie, électricité, et bien plus. Chaque service se présente sous forme de carte, avec une icône claire, une courte description, et le prix de départ. La barre de recherche intégrée permet de filtrer en quelques secondes. L'idée, c'est de rendre la navigation simple et d'orienter l'utilisateur vers le service qui correspond vraiment à ce qu'il cherche.



Figure 10 : Interface de sélection des catégories et services.

La page « Confirmation de commande »

Avant de finaliser la commande et de payer, l'utilisateur accède à un récapitulatif précis : service, prestataire, note, prix proposé. Il reste à cliquer sur « Valider et payer » pour confirmer. Cette étape limite les erreurs et sécurise vraiment la prise de décision.

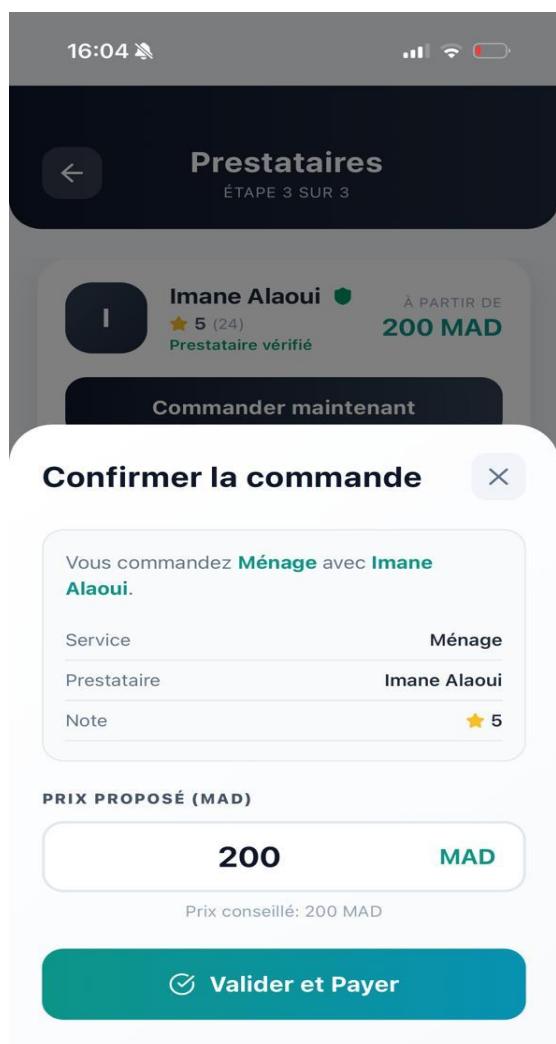


Figure 11 : Interface de confirmation de commande avant paiement.

La page « Mes commandes »

Sur cette page, l'utilisateur retrouve tout l'historique de ses commandes. Il filtre facilement selon le statut : en attente, en cours, terminées ou annulées. Pour chaque mission, on voit le service, le prestataire, la date, le montant et l'état. Ce tableau de bord rend le suivi simple et renforce la transparence sur tout le cycle de vie des commandes.



Figure 12 : Interface de gestion et de suivi des commandes.

4.2 Interfaces mobiles :Prestataire

La page « Tableau de bord Prestataire »

Le tableau de bord donne au prestataire une vue claire sur son activité. Il affiche d'un coup d'œil les revenus mensuels, le nombre de missions terminées, la note moyenne et le taux d'acceptation. Un bouton permet d'activer ou désactiver le statut « En ligne » essentiel pour recevoir de nouvelles missions. Cette interface aide vraiment à piloter l'activité jour après jour.

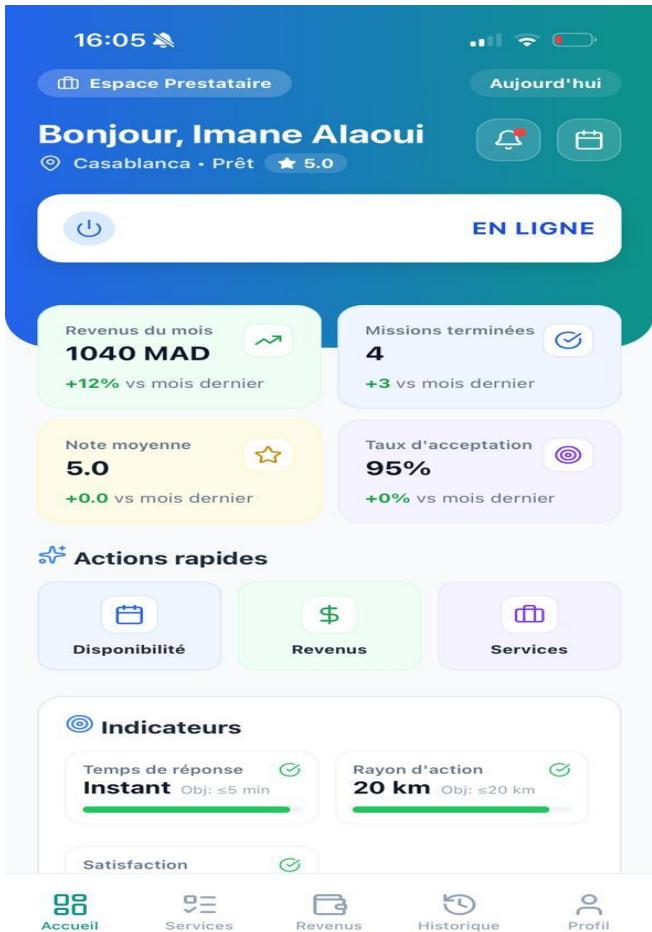


Figure 13 : Tableau de bord prestataire avec indicateurs de performance.

La page « Services & Compétences »

Ici, le prestataire gère tous les services proposés. L'interface affiche les compétences actives, les stats associées (réservations, durée moyenne, revenus générés) et le tarif pour chaque service, qu'il peut modifier à tout moment. Ajouter, changer ou supprimer une compétence se fait en quelques clics. L'objectif : donner un vrai contrôle, simple et flexible, sur l'offre du prestataire.

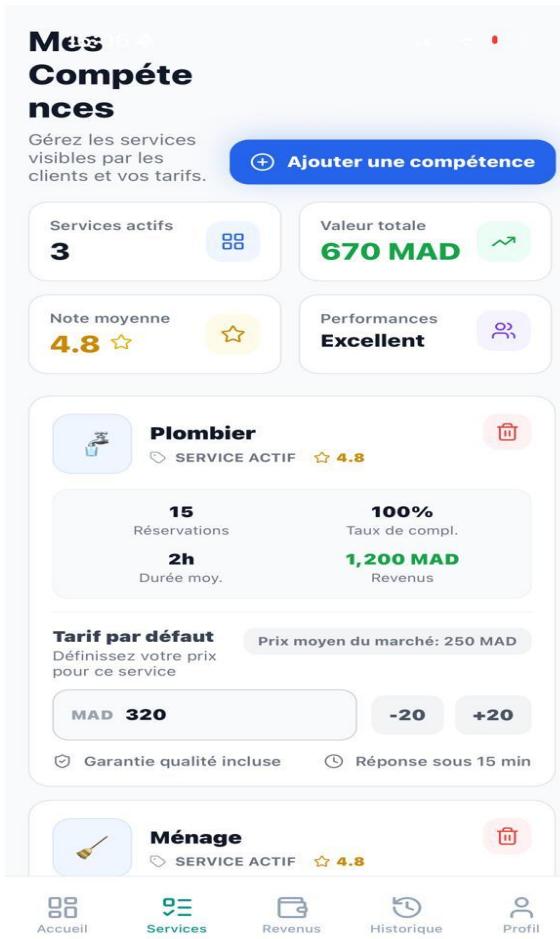


Figure 14 : Interface de gestion des services et compétences.

La page « Revenus & Portefeuille »

Sur « Mon portefeuille », le prestataire garde un œil sur ses revenus et toutes ses transactions. Le solde disponible, les montants en attente, les stats détaillées (revenus mensuels, moyenne par mission, meilleur mois) sont bien visibles. Retirer ou exporter les revenus se fait directement depuis l'interface. Ici, la gestion financière est simple, claire, transparente.

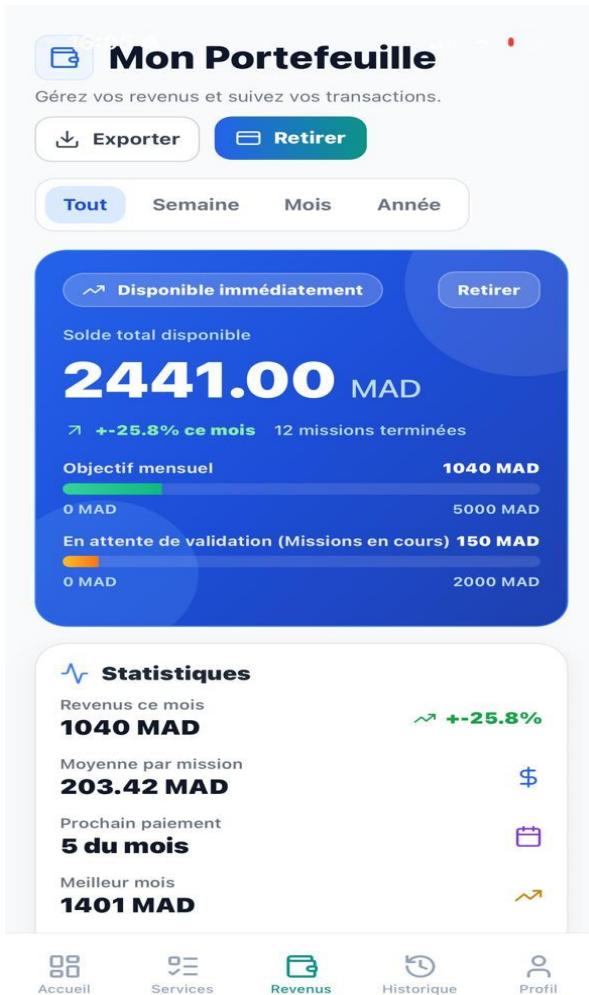


Figure 15 : Interface de suivi des revenus et du portefeuille.

La page « Historique des missions »

La page « **Historique des missions** » permet au prestataire de consulter l’ensemble des missions passées, qu’elles soient **terminées** ou **annulées**. En haut de l’interface, des indicateurs synthétiques présentent le nombre total de missions, celles finalisées avec succès et celles annulées, offrant une vue d’ensemble rapide de l’activité.



Figure 16 : historique des commandes du prestataire

4.3 Interfaces mobiles : Administration

La page « Dashboard Administration »

La page « Dashboard Administration » constitue le centre de supervision principal de la plateforme . Elle permet à l'administrateur d'avoir une vision globale et synthétique de l'activité du système, facilitant ainsi la prise de décision et le suivi des performances en temps réel.



Figure 17 : Tableau de bord de l'administrateur

La page « Paramètres »

La page « Paramètres » permet à l'administrateur de consulter et gérer les informations essentielles liées à son compte et à l'application. Elle affiche le profil administrateur avec le nom, l'adresse email et le rôle attribué, garantissant une identification claire de l'utilisateur connecté.

Une section dédiée à la sécurité informe que la session est protégée par une authentification sécurisée, tout en recommandant la déconnexion après usage sur un appareil partagé. Enfin, un encart « À propos de l'application » présente le nom de l'application, la version courante et l'environnement d'exécution .

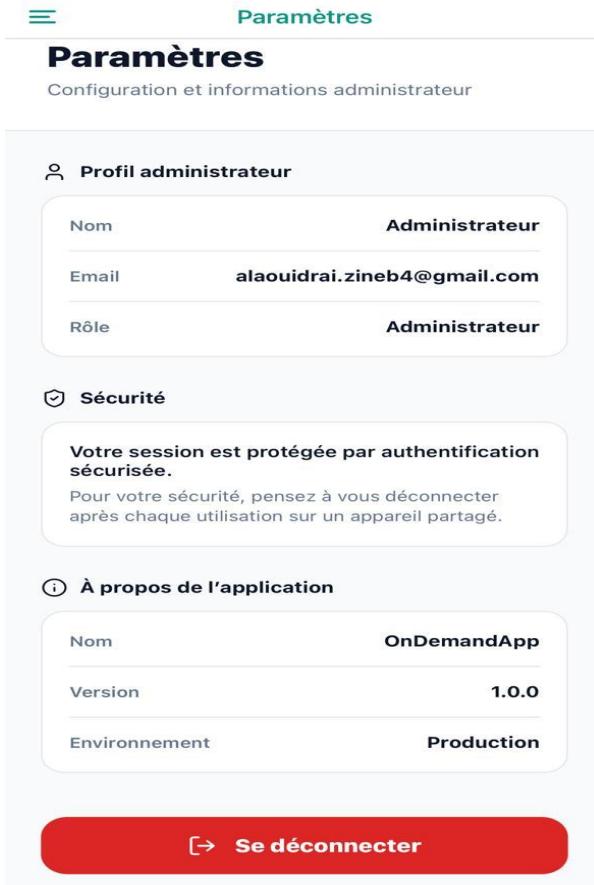


Figure 18 : Interface mobile des paramètres administrateur.

La page « Commissions »

La page « Commissions » assure le suivi des revenus nets générés par la plateforme. Elle affiche des indicateurs financiers clés tels que les gains cumulés, le nombre total de prestations réalisées.

Un tableau détaillé présente chaque mission avec le montant facturé, la commission appliquée et le revenu net correspondant, offrant une transparence totale sur les flux financiers. Cette page permet ainsi à l'administrateur de contrôler la rentabilité de la plateforme et d'analyser la performance économique des services proposés.

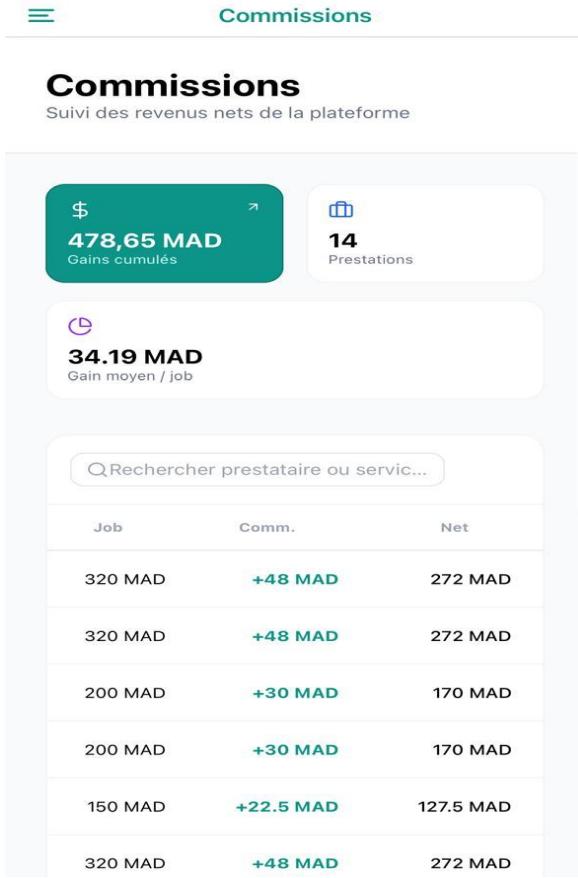


Figure 19 : Interface mobile de suivi des commissions et revenus de la plateforme.

La page « Paiements »

La page « **Paiements** » est destinée à la gestion et au suivi des transactions financières. Elle permet de visualiser les paiements effectués avec leur statut ,le montant, le client concerné, le service associé, la date et la méthode de paiement utilisée. Une fenêtre de confirmation s'affiche après chaque transaction réussie, indiquant clairement le montant payé et les détails de la transaction. Cette interface garantit la traçabilité des paiements et renforce la confiance dans le système financier de l'application.



Figure 20 : Interface mobile de confirmation et suivi des paiements.

La page « Commandes »

La page « **Commandes** » permet à l'administrateur de superviser l'ensemble des missions et interventions réalisées sur la plateforme. Elle propose un champ de recherche facilitant l'identification rapide des commandes par client, service ou identifiant. Chaque ligne de commande affiche le client, le prestataire associé, le prix, ainsi que le statut de la mission (confirmée, complétée, rejetée). Un bouton d'action permet d'accéder aux détails complets de chaque commande, assurant un suivi précis et une gestion efficace des opérations.

PRO+CL	PRIX	STATUT	ACT.
AHMED ALI & Imane Alaoui	200 MAD	Confirmed	→
AHMED ALI & Imane Alaoui	320 MAD	Completed	→
AHMED ALI & Imane Alaoui	150 MAD	Rejected	→
AHMED ALI & Imane Alaoui	320 MAD	Completed	→
AHMED ALI & Imane Alaoui	200 MAD	Completed	→

Figure 21 : interface des différentes commandes passées

La page « Utilisateurs »

La page **Utilisateurs** est dédiée à la gestion des comptes clients, prestataires et administrateurs. Elle permet à l'administrateur de superviser l'ensemble des profils inscrits sur la plateforme et d'assurer un contrôle efficace des accès.

IDENTITÉ RÔLE	STATUT	ACTIONS
S Provider	Pending	Gérer
K Client	Active	Gérer
F Client	Active	Gérer
F Client	Active	Gérer
K Provider	Pending	Gérer
I Provider	Active	Gérer

Figure 22 : Interfaces des différents utilisateurs

La page « Validations Prestataires »

La page **Validations Pro** est conçue pour examiner et valider les candidatures des nouveaux prestataires souhaitant rejoindre la plateforme. Elle joue un rôle essentiel dans le contrôle de la qualité des services proposés.

Chaque candidature est présentée sous forme de carte indiquant le nom du prestataire, son adresse email ainsi que l'état de vérification de son compte. Des indicateurs visuels signalent clairement si l'email est vérifié et si les documents requis, comme le CV, sont disponibles ou manquants.

The screenshot displays the 'Validations Pro' application interface. At the top, there is a header with three horizontal lines on the left and the text 'Validations Pro' in the center. Below the header, the word 'Validations' is prominently displayed in a large, bold, dark blue font. Underneath 'Validations', a subtitle reads 'Examiner les candidatures des nouveaux prestataires.' (Examine the applications of new providers). The main content area contains two separate validation cards.

Validation Card for Salim:

- Name:** Salim
- Email:** leon2025kd@protonmail.com
- Status:** EMAIL NON VÉRIFIÉ (Email Not Verified)
- Review Options:** Général (General) and CV Manquant (CV Missing)
- Buttons:** 'Examiner' (Examine) and 'Valider' (Validate)

Validation Card for Kawtar Bennani:

- Name:** Kawtar Bennani
- Email:** g.amez122@proton.me
- Status:** EMAIL VÉRIFIÉ (Email Verified)
- Review Options:** Général (General) and CV Disponible (CV Available)
- Buttons:** 'Examiner' (Examine) and 'Valider' (Validate)

Figure 23 : page de validation des demandes d'emploi des prestataires

5. Conclusion

En résumé, ce chapitre a décrit l'architecture du système, s'appuyant sur le modèle Multi-tiers pour garantir modularité, scalabilité et sécurité. Il a détaillé les outils technologiques essentiels au projet, introduit le langage de programmation au cœur du développement, et illustré des interfaces clés reflétant l'intégration fonctionnelle de l'application. Ces éléments, combinés, forment un socle cohérent permettant d'assurer le bon fonctionnement global du système, tout en offrant une base solide pour des évolutions futures.

Conclusion et perspectives

Ce projet a permis la conception et la réalisation d'une application mobile **On-Demand** performante, sécurisée et orientée services, visant à mettre en relation des clients et des prestataires de manière rapide et fiable. À travers le projet **OnDemandApp**, nous avons mis en œuvre une architecture moderne reposant sur une application mobile développée avec **React Native et Expo**, un backend robuste basé sur **.NET**, ainsi qu'un système de gestion des données structuré et sécurisé.

L'ensemble de la solution intègre des mécanismes de sécurité avancés, notamment l'authentification sécurisée via **JWT**, la validation des comptes par courrier électronique et l'implémentation de **l'authentification à deux facteurs** renforçant ainsi la protection des accès et des données utilisateurs. La gestion centralisée par l'administrateur (utilisateurs, prestataires, commandes, paiements, statistiques et rapports) garantit un contrôle global et une supervision efficace de la plateforme.

Les objectifs initiaux du projet ont été atteints avec succès : un parcours utilisateur fluide, une création de commande simplifiée, un suivi en temps réel des missions via le GPS, un système de paiement sécurisé avec gestion des commissions, ainsi qu'un tableau de bord analytique permettant d'évaluer les performances globales de la plateforme. L'intégration de **Firebase** a permis d'assurer des notifications en temps réel et une meilleure communication entre les différents acteurs du système.

En termes de perspectives, plusieurs axes d'amélioration peuvent être envisagés. L'intégration de l'intelligence artificielle pourrait permettre la recommandation intelligente de prestataires en fonction de la localisation, des évaluations et de l'historique des missions. Le renforcement des moyens de paiement en ligne (Stripe, PayPal) ainsi que l'automatisation avancée de la facturation constituent également des pistes d'évolution importantes. Enfin, l'ouverture vers des technologies émergentes telles que la blockchain pourrait améliorer la traçabilité des transactions et renforcer davantage la transparence du système.

Ce projet a représenté une expérience enrichissante tant sur le plan technique que méthodologique, consolidant nos compétences en développement **full-stack**, en gestion de projet et en conception de solutions numériques modernes. Il témoigne de notre capacité à concevoir une application complète, évolutive et adaptée aux exigences actuelles du marché des services à la demande.

Webographie

1. <https://reactnative.dev/docs/getting-started>
2. <https://docs.expo.dev/>
3. <https://learn.microsoft.com/en-us/aspnet/core/>
4. <https://firebase.google.com/docs>
5. <https://jwt.io/introduction>
6. <https://learn.microsoft.com/en-us/security/>
7. <https://stripe.com/docs>
8. <https://developer.paypal.com/docs/>
9. <https://openclassrooms.com/fr/courses/7150606-developpez-une-application-mobile-avec-react-native>
10. <https://www.geeksforgeeks.org/reactjs/introduction-react-native/>