|  |  |  |
| --- | --- | --- |

**Réalisation d’une application mobile pour la gestion de Véhicules de transport avec chauffeurs (VTC)**

**MOOV**

Par

RAMALALAHARISON Tokiniaina

Mémoire présenté  
en vue de l’obtention du grade de Licence

en Graphique design ou Communication Digitale

option Graphic Design

octobre, 2024

Jury :

Dr Olivier Robinson, président  
 M. Tahina Razafinjoelina, examinateur  
 M. Hariniaina Rajaonarifetra, encadreur professionnel

© RAMALALAHARISON Tokiniaina, 2024

**Table des matières**

[**Liste des tableaux 2**](#_gjdgxs)

[**Liste des figures 3**](#_30j0zll)

[**Glossaire 4**](#_3znysh7)

[**Avant-propos 1**](#_2et92p0)

[L’IT University 1](#_tyjcwt)

[L’Institution/Entreprise d’accueil 2](#_3dy6vkm)

[Remerciements 3](#_1t3h5sf)

[**Introduction 4**](#_4d34og8)

[**1 Présentation du projet 5**](#_2s8eyo1)

[1.1 Objectifs du projet 5](#_17dp8vu)

[1.2 Planning de réalisation 5](#_26in1rg)

[1.3 Technologies utilisées 6](#_lnxbz9)

[1.3.1 Express (Node.js version 20.10) 6](#)

[1.3.2 React (TypeScript) 7](#)

[1.3.3 Ionic version 7 9](#_uur6ava6pw4e)

[1.3.4 Web Service REST 9](#_8okob3mugd2)

[1.3.5 Base de données PostgreSQL 11](#_rptlpndv47wh)

[1.3.6 Outils 12](#_qpokwwehndor)

[**2 Réalisation de l’application 14**](#_35nkun2)

[2.1 Analyse et conception 14](#_1ksv4uv)

[2.1.1 Analyse de l'existant 14](#_44sinio)

[2.1.2 Conception de l’application 14](#_2jxsxqh)

[2.1.2.1 Architecture 14](#_o4ors4vc20ab)

[2.1.2.2 Conception 15](#_yny2ilgaui2i)

[2.2 Développement par fonctionnalité ou module 18](#_z337ya)

[2.2.1 Création d’un Logo et identité visuelle 18](#_3j2qqm3)

[2.2.2 Création d’interface Utilisateur et Expérience utilisateur (UI/UX) 21](#_1y810tw)

[2.2.3 Développement des interfaces mobile et web 24](#_pdcvunq7hvux)

[2.2.4 Authentification et Inscription 26](#_ffc051vkrcua)

[2.2.5 Module de notification push 30](#_b885adqc9fff)

[2.2.6 Gestion des historiques de réservation de chaque client et chauffeur 32](#_bxy08lbflmdh)

[2.2.7 Gestion des historiques chez l’administrateur 34](#_two6l8w7sgiv)

[2.2.8 Gestion des véhicules 35](#_el71ghvicc41)

[2.2.9 Gestion des utilisateurs 36](#_4i7ojhp)

[2.3 État d’Analyse et Statistiques 39](#_2xcytpi)

[2.3.1 État numéro 1 39](#_1ci93xb)

[2.3.2 État numéro 2 39](#_3whwml4)

[2.3.3 Statistique numéro 1 39](#_2bn6wsx)

[2.4 Problèmes rencontrés et solutions 40](#_3as4poj)

[**3 Évaluation du projet et connaissances acquises 41**](#_1pxezwc)

[3.1 Bilan pour l'entreprise 41](#_49x2ik5)

[3.2 Bilan personnel 41](#_2p2csry)

[3.3 Extension et évolution de l’application 41](#_147n2zr)

[**Conclusion 43**](#_3o7alnk)

[**Bibliographie 1**](#_9wrvfvca96yp)

[**Annexe 1**](#_ihv636)

# 

# Liste des tableaux

[Tableau 1 :](#_qsh70q) Tableau comparatif Express - Django6

Tableau 2 : Tableau comparatif React - Angular 8

Tableau 3 : Tableau comparatif Ionic type React – React Native 9

Tableau 4 : Tableau comparatif REST - SOAP 10

Tableau 5 : Tableau comparatif PostgreSQL - MySQL 11

# Liste des figures

Figure 1 : diagramme de gantt 4

Figure 2 : Classement des bases de données en 2024 11

Figure 3 : Architecture MVC de l’application 15

Figure 4 : Modèle conceptuel de la base de données 17

Figure 5 : Logo 19

Figure 6 : Forme symbolique du logo 20

Figure 7 : Couleurs utilisées et combinaisons 21

Figure 8 : Typographie 22

Figure 9 :Écran d'accueille 24

Figure 10 :Écrans des étapes du paiement 24

Figure 11 : Écran de la liste des clients chez l’administrateur 25

# Glossaire

**Application Programming Interface (API)** : Interface de programmation qui permet de se brancher sur une application pour échanger des données.

**HTTP (Hypertext Transfer Protocol) :** Un protocole utilisé pour la transmission de données sur le web. Il permet aux navigateurs et aux serveurs de communiquer en envoyant et recevant des requêtes et des réponses pour afficher des pages web.

**IDE (Integrated Development Environment) :** Environnement de développement intégré qui combine un éditeur de code, un compilateur, un débogueur et d'autres outils dans une interface unique pour faciliter le développement de logiciels.

**MVC (Model-View-Controller)** : Un modèle d'architecture logicielle qui sépare une application en trois composants principaux : le modèle (gère les données), la vue (gère l'affichage des données) et le contrôleur (gère la logique et les interactions entre le modèle et la vue).

**RESTful (Representational State Transfer)** : Un style d'architecture pour les services web qui repose sur le protocole HTTP. Il permet aux clients et aux serveurs d'échanger des informations en utilisant des méthodes standard comme GET, POST, PUT et DELETE.

**SOAP (Simple Object Access Protocol) :** Un protocole basé sur XML utilisé pour échanger des informations structurées entre des applications via le réseau, souvent utilisé pour la communication entre systèmes distribués.

**Uniform Resource Identifier (URI)** : Un élément permettant d’identifier une ressource sur Internet.

**Uniform Resource Locator (URL) :** Adresse d’un site ou d’une page hypertexte sur Internet

**XML (Extensible Markup Language) :** Un langage de balisage utilisé pour structurer, stocker et transporter des données.

**UI (User Interface)** : Interface utilisateur, comprenant les éléments visuels avec lesquels l’utilisateur interagit (boutons, icônes, menus, etc.) pour rendre l’expérience intuitive et agréable.

**UX (User Experience)** : Expérience utilisateur, englobant les perceptions et émotions lors de l'interaction avec une application, en créant des parcours logiques et satisfaisants.

# Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d’études de Licence en Informatique de l’IT University; stage effectué au département "développement" de "WYLog" durant trois mois, du 31 juillet au 31 octobre 2024.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succinctement d’une part l’IT University et d’autre part mon entreprise d’accueil.

## L’IT University

Fondée en 2011, l’IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers, de préférence scientifiques :

* En trois ans, pour l‘obtention d’une :
  + Licence, option Développement, Réseaux et Bases de Données ou Web et Design
  + Licence, option Graphic Design ou Communication Digitale
* En cinq ans, pour l’obtention d’un :
  + Master MBDS en coopération avec l’Université Côte d’Azur à Nice Sophia Antipolis – France
  + Master BIHAR en coopération avec l’ESTIA du Pays Basque - France

Étant une formation professionnalisante, l’ITU a tissé des liens forts avec ses partenaires industriels, dont l’opérateur convergent TELMA et la plupart des entreprises et institutions du secteur des TIC[[1]](#footnote-0). Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l’envoi de conférenciers ou par l’accueil des étudiants en stage. Beaucoup de ces partenaires recrutent aussi les sortants dès leur sortie d’École.  
D’autre part, le corps enseignant de l’ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.

## L’Institution/Entreprise d’accueil

Wylog est une entreprise française dont le siège est basé à Paris, spécialisée dans le secteur informatique depuis 2003. Plus de 18 années d’expériences, la société Wylog a acquis une grande capacité à allier expertise technique et compréhension métier.

Elle dispose aujourd’hui de six bureaux, répartis dans cinq pays (France, Madagascar, Italie, Philippines, et Etats-Unis) composées d’une équipe de 120 professionnels passionnés. Les équipes sont organisées autour de plusieurs pôles technologiques (Développements Web et Mobile, Développements AR/VR1 , Blockchain, Département IoT2 /Live Streaming).

Wylog travaille dans plusieurs secteurs d’activités et développe des applications pour l’industrie, les médias & communication, la fintech et mène plusieurs projets avec des entreprises à profils variés (Start-up, compte corporate, organisme public) basées aussi bien à Madagascar qu’à l’étranger.

## Remerciements

En premier lieu, je tiens à remercier l’IT University pour les enseignements reçus et les connaissances acquises durant les années d’apprentissage.

Ensuite, je tiens à montrer ma reconnaissance envers les membres du jury, présidé par Mr Vahatriniaina RAKOTOMALALA , pour le dévouement dans l’examen et la correction de ce présent mémoire.

Je tiens également à remercier M. Hariniaina Rajaonarifetra, mon encadreur professionnel, pour m’avoir consacré du temps pour un suivi régulier, et pour m’avoir fourni de l’aide nécessaire pour l’accomplissement de ce stage. Mes remerciements aussi à l’entreprise WYLog Madagascar, pour m’avoir accueilli lors de mes trois mois de stage.

Enfin, ma gratitude envers ma famille de m’avoir soutenu pour l’aboutissement de mes études jusqu’au point où j’en suis, et aussi mes amis pour leur soutien incomparable durant ces années d’études.

# Introduction

L’essentiel du travail que j’ai réalisé a porté sur la création d’une application mobile pour la gestion de Véhicules de Transport avec Chauffeurs (VTC). À Madagascar, le transport public et privé rencontre plusieurs défis : bus surchargés, taxis peu fiables, et manque de solutions numériques pour organiser les déplacements. Une étude montre que près de 70 % des Malgaches préfèrent les bus, malgré leur manque de confort, principalement à cause du coût élevé des taxis. Le besoin d’une alternative fiable, confortable, et abordable devient de plus en plus pressant. Cette application VTC dédiée aux taxis répond à cette demande en offrant une solution de transport mieux structurée et accessible via une interface innovante.

J’ai choisi ce projet pour appliquer mes compétences à un problème concret, tout en participant à la modernisation des infrastructures numériques à Madagascar. Mon objectif est de développer une solution de mobilité pratique, fiable et accessible. Durant ce stage, ma mission principale est de participer au développement de l’application, avec un focus sur :

• la création d’un logo et identité visuelle pour l’application

• la création des interfaces pour le client, le conducteur et l'administrateur,

**Mobile :** • Authentification et inscription avec mot de passe oublié

• la mise en place de la notification push,

• le module de gestion de profils utilisateurs,

• Gestions de factures dans la partie client,

• les historiques de réservations des utilisateurs

**Web (Administrateur) :** • la gestion des véhicules,

• les statistiques et tableaux de bord,

• les historiques de toutes les courses.

Ces missions visent à répondre à la question suivante : comment offrir une solution numérique et digitale pour la gestion des VTC qui soit fiable, abordable, et adaptée aux besoins locaux ? Ce projet représente une opportunité unique de contribuer à un domaine en pleine expansion, où la mobilité durable et les nouvelles technologies s'entrecroisent. En tant qu'ingénieur logiciel, mon rôle est de trouver un équilibre entre innovation technologique et adaptation aux réalités locales.

# Présentation du projet

## Objectifs du projet

Les principaux objectifs du projet incluent la mise en place d'une plateforme numérique qui :

* Créer une identité moderne, attrayante qui évoque la confiance , le confort et non coûteuse.
* Offre une expérience utilisateur fluide et facile à utiliser, tant sur l’application mobile que sur le web, avec un design moderne, captivant et épuré.
* Permet la gestion d'une interface administrateur pour superviser les réservations, ainsi que la gestion des chauffeurs, administrateurs et clients.
* Inclut la gestion des notifications push concernant les avis des conducteurs, les demandes de courses, et l’attribution des courses aux chauffeurs.
* Facilite la gestion de courses par rapport à des statistiques et des tableaux de bord intuitifs.

## Planning de réalisation

Diagramme de GANTT .

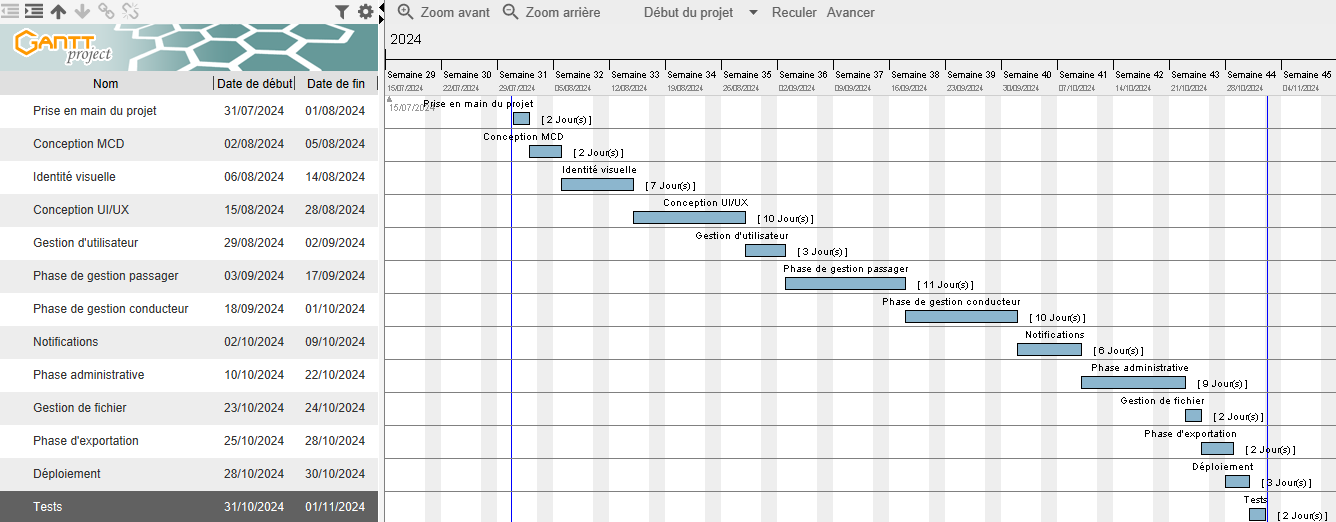


Figure 1 : diagramme de gantt

## Technologies utilisées

### Express (Node.js version 20.10)

Pour le backend, nous avons choisi Express avec Node.js, un framework minimaliste qui permet de créer des applications web et **API** légères rapidement. Sa structure **RESTful** simplifie la gestion des routes, des requêtes **HTTP** et des middlewares, tout en offrant une grande flexibilité.

Nous utilisons la version 4 d'Express, connue pour sa simplicité, ses performances, et son écosystème riche. Il facilite l'intégration avec des bases de données comme MongoDB ou PostgreSQL, et des services comme **Firebase**.

Express s'est imposé pour sa courbe d'apprentissage rapide et ses performances en temps réel, notamment pour des applications nécessitant une gestion rapide des données, comme notre solution de mobilité.

Voici un tableau comparatif entre Express et Django, deux frameworks très populaires, qui nous a conduit à choisir Express pour ce projet :

| **Caractéristiques** | **Express (Node.js)** | **Django (Python)** |
| --- | --- | --- |
| Installation | Simple et rapide, via npm | Configuration plus complexe via pip et virtualenv |
| Architecture | RESTful, minimaliste, flexible | MVC structuré, complet, batteries incluses |
| Langage | JavaScript/TypeScript | Python |
| Performance | Très rapide, adapté aux applications en temps réel | Bonnes performances, mais moins rapide que Express dans les I/O intensives |
| Modularité | Très modulaire, choix des composants | Intégration de nombreux modules par défaut |
| Gestion des bases de données | Support natif pour plusieurs bases de données (MySQL, MongoDB, PostgreSQL | ORM intégré avec migrations automatiques (Django ORM) |
| Communauté et documentation | Grande communauté, documentation exhaustive | Forte communauté, excellente documentation |
| Cas d’utilisation | Applications en temps réel, API légères | Applications web complexes, avec besoin d’une structure solide |

Tableau 1 : Tableau comparatif Express - Django

Ce tableau nous montre que bien que **Django** soit robuste et structuré, nous avons choisi **Express** pour sa légèreté, sa flexibilité et son adaptabilité aux besoins spécifiques de notre projet.

### React (TypeScript)

Pour la partie administrative de notre application, nous avons choisi React avec TypeScript. React est une bibliothèque JavaScript open-source axée sur la création d'interfaces utilisateur dynamiques et réactives, tandis que TypeScript ajoute un typage statique à JavaScript, ce qui améliore la sécurité du code et réduit les erreurs. Ce combo permet de structurer le développement de l’interface administrative de façon modulaire et maintenable, tout en offrant une expérience utilisateur fluide.

Nous avons utilisé React TypeScript car il offre une meilleure gestion des erreurs, et permet d’écrire du code plus propre et maintenable, particulièrement utile dans un contexte où la gestion des utilisateurs, véhicules, réservations et statistiques nécessite une interface claire et évolutive.

Le choix de React s’est également imposé grâce à son écosystème riche en bibliothèques et sa capacité à créer des composants réutilisables, essentiels pour la partie administrative. En comparaison avec d’autres frameworks front-end, tels que Angular, React avec TypeScript offre une plus grande flexibilité et une courbe d’apprentissage moins complexe.

| Critères | React TypeScript | Angular |
| --- | --- | --- |
| Installation et configuration | Simple à configurer avec Create React App et TypeScript | Configuration plus complexe avec Angular CLI |
| Flexibilité | Très flexible, aucune structure imposée | Structure plus rigide avec MVC intégré |
| TypeScript intégré | Support natif, mais nécessite configuration | Intégré par défaut |
| Performance | Rapide pour les applications avec des composants réutilisables | Moins performant pour de très grandes applications |
| Taille du projet | Léger, parfait pour des applications modulaires | Plus lourd en raison de l'inclusion d'outils supplémentaires |
| Courbe d’apprentissage | Relativement courte, surtout pour les développeurs JavaScript | Plus longue, nécessite l’apprentissage du framework complet |
| Écosystème | Large choix de bibliothèques tierces | Écosystème intégré, moins de dépendances tierces |

Tableau 2 : Tableau comparatif React - Angular

Ce tableau montre que **React TypeScript** a été choisi pour sa légèreté, sa flexibilité et sa facilité d’apprentissage, tout en restant performant pour des interfaces utilisateur administratives riches et réactives.

### Ionic version 7

Pour la partie mobile de notre application, nous avons choisi Ionic avec React, une solution hybride qui permet de développer des applications mobiles pour Android et iOS en utilisant un seul codebase. Ionic offre des composants prêts à l'emploi, optimisés pour le mobile, et permet d'accéder aux fonctionnalités natives du téléphone via Capacitor. React, quant à lui, permet de gérer efficacement les états et les interactions complexes avec l'interface utilisateur.

L’association d'Ionic avec React permet de créer une application mobile performante et réactive, tout en simplifiant la maintenance du projet. Cette solution est particulièrement adaptée pour une application nécessitant une interface fluide et interactive, avec un développement accéléré pour plusieurs plateformes. En comparaison avec des solutions natives comme React Native ou Flutter, Ionic React se distingue par sa simplicité d’utilisation et sa flexibilité pour des projets de taille moyenne.

| Critères | Ionic React | React Native |
| --- | --- | --- |
| Codebase | Unique pour iOS et Android | Unique pour iOS et Android |
| Accès aux fonctionnalités natives | Via Capacitor/Cordova | Accès natif direct |
| Performance | Performant, mais dépend du WebView | Très performant, natif |
| Facilité d'utilisation | Facile, composants préconstruits | Moins facile, plus de code |
| Courbe d'apprentissage | Rapide | Moyenne à longue |

Tableau 3 : Tableau comparatif Ionic type React – React Native

### Web Service REST

Un webservice est un protocole qui permet à une application d’échanger des informations/données à d’autres applications même si ces dernières sont construites dans des langages de programmation différents. Elle procède de manière à ce que le client effectue une requête du côté de l’interface utilisateur dans un des formats suivants : XML, JSON ou HTML, vers le côté serveur de l’application qui délivre les réponses sous le même format que sa demande.

REST est l’un des modes d’échanges du webservice. La plateforme présente deux couches différentes pour la présentation et le traitement, donc elle a besoin d’un mode d’échange de données. De même, le protocole permet au client de recevoir des informations d’un serveur distant sans pour autant stocker toutes les données sur sa machine. Ce serveur peut être interrogé par un grand nombre de clients de manière simultanée. REST utilise le protocole HTTP pour faire interagir le client et le serveur. Ce protocole a un style architectural bien défini lors d’un envoi de requête (ex : GET + URL ) . Les commandes HTTP les plus courantes sont :

• GET – Récupération d’une ressource spécifiée

• POST – Création d’une nouvelle ressource

• PUT – Mise à jour d’une ressource existante

• DELETE – Suppression d’une ressource existante

| REST | SOAP |
| --- | --- |
| REST est un style architectural | SOAP est un protocole |
| Autorise différents formats de données : XML, JSON, HTML | N’autorise que le format XML |
| Utilise **l’URI** pour exposer la logique métier | Utilise des interfaces de services pour exposer la logique métier |
| Architecture légère, bonne performance | Nécessite plus de bandes passantes et ressources, ralentissement du chargement des pages |
| Architecture flexible qui permet aux développeurs de mettre en œuvre les recommandations à leur manière | Possède des normes à suivre strictes |

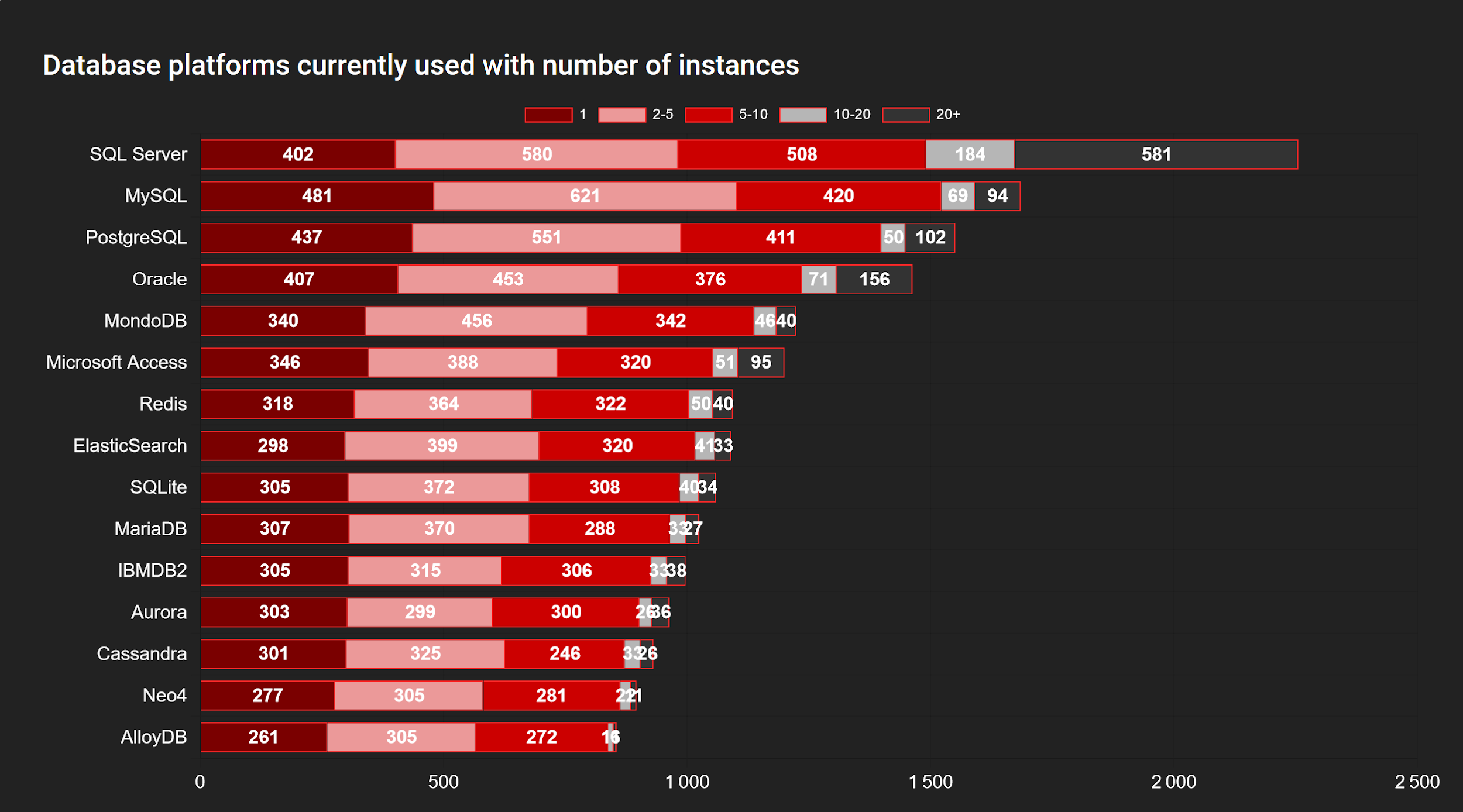
Tableau 4 : Tableau comparatif REST - SOAP

Parmi les web services existants, **SOAP** fait aussi partie des plus pertinents. Nous allons voir ci-après un tableau comparatif de ces deux web services : Certes, SOAP a beaucoup d’avantages pour un mode d’échange entre client et serveur, mais le choix s’est porté sur le WebService REST.

### Base de données PostgreSQL

Pour la gestion des données de notre projet, nous avons opté pour PostgreSQL, un système de gestion de base de données relationnel open source puissant, réputé pour sa robustesse et sa conformité aux standards ACID (Atomicité, Cohérence, Isolation, Durabilité). PostgreSQL est reconnu pour son extensibilité et sa capacité à gérer des opérations complexes et de grandes quantités de données. Il offre également un support natif pour JSON, ce qui le rend adapté pour gérer des données relationnelles, répondant ainsi aux besoins modernes des applications web.

Nous avons choisi PostgreSQL pour ses capacités avancées en matière de gestion de transactions, de performances, et pour son extensibilité avec des outils comme PostGIS pour la gestion des données géospatiales. De plus, son intégration fluide avec Node.js via des bibliothèques comme pg en fait une solution idéale pour notre projet.

Figure 2 : Classement des bases de données en 2024

| Critères | PostgreSQL | MySQL |
| --- | --- | --- |
| Licence | Open source | Open source |
| Support des types de données | Extensible (JSON, XML, etc.) | Limité par rapport à PostgreSQL |
| Conformité ACID | Complète | Partielle |
| Performance des transactions | Très performant, notamment sur les transactions complexes | Très performant sur les lectures |
| Extensibilité | Supporte des extensions (PostGIS, PL/pgSQL, etc.) | Limité par rapport à PostgreSQL |
| Gestion des données géospatiales | Support avancé avec PostGIS | Support limité |
| Prise en charge des données non relationnelles | JSON natif, HSTORE | Support JSON limité |
| Recommandation | Projets complexes, requêtes lourdes | Projets simples, performances optimisées pour les lectures |

Tableau 5 : Tableau comparatif PostgreSQL - MySQL

Bien que MySQL soit une solution très populaire, nous avons choisi PostgreSQL pour sa flexibilité, sa performance dans les environnements complexes, et son extensibilité, qui sont des atouts majeurs pour la gestion de notre base de données.

### Outils

**Git**:

Pour voir l’avancement et l’évolution de notre développement et afin d’avoir un backup de notre code source, nous avons utilisé git comme outil de gestion des codes sources.

Git est un logiciel open-source qui contrôle la version des applications que nous créons. Il permet de conserver une historique des modifications effectuées sur un projet afin de rapidement identifier les changements effectués et de revenir à une version antérieure en cas de problème.

Au lieu de gérer un projet dans plusieurs dossiers, on a un seul projet avec une base de données qui stocke toutes les versions des fichiers. Git permet également à plusieurs personnes de travailler sur un même projet en même temps en gérant les éventuels conflits et donc facilite et rend beaucoup plus efficace un travail en équipe. Son utilisation génère moins de risques de perdre le code source. L’utilisation de cet outil est indispensable pour une équipe de développeurs.

**Figma :**

Pour la conception des interfaces utilisateurs de notre application, nous avons utilisé Figma. Figma est un outil de conception d'interface collaboratif basé sur le cloud, très prisé pour le UI/UX design. Il permet de créer des maquettes interactives, des prototypes, et de travailler simultanément avec d'autres membres de l'équipe en temps réel, facilitant ainsi la communication et la coordination entre les designers et les développeurs.

L'outil offre une flexibilité et une simplicité d’utilisation, tout en permettant de suivre les évolutions des conceptions. Grâce à ses fonctionnalités comme le partage de designs en temps réel, l'exportation de composants pour le développement, et la gestion de design systems, Figma a considérablement accéléré le processus de conception et réduit les itérations.

L'intégration de Figma dans le workflow a permis de garantir la cohérence des interfaces tout au long du développement, en facilitant l'adaptation des maquettes aux contraintes techniques.

# Réalisation de l’application

## Analyse et conception

### Analyse de l'existant

Le système de transport existant à Madagascar, particulièrement pour les taxis, repose essentiellement sur des interactions directes entre les clients et les chauffeurs, sans intégration de solutions numériques pour la gestion des réservations ou du suivi en temps réel. Les utilisateurs doivent généralement appeler directement les chauffeurs ou attendre dans la rue, ce qui conduit à des pertes de temps et un manque de visibilité sur la disponibilité des véhicules.

Bien qu'il existe une application de gestion de VTC, **Misy**, celle-ci n'est pas encore largement adoptée et n'offre pas une solution complète aux défis actuels du marché. Il n'existe donc actuellement pas d'application dédiée qui centralise ces services, ce qui complique l’expérience utilisateur. Après les analyses, nous constatons un large pourcentage de citoyens urbains qui ont accès à Internet, aux téléphones et aux cartes bancaires.

### Conception de l’application

La phase de conception est une partie importante, voir même primordiale, dans le développement d’un projet informatique. Cette étape facilite l’avancement du projet car elle nous permet d’avoir une idée plus précise de ce que l’on veut concevoir. Analyser attentivement l’application à développer permet de comprendre les objectifs et attentes pour une préparation aux éventualités possibles et d’avoir une vision sur ce que le projet deviendra à l’avenir.

### Architecture

La plateforme a été conçue pour être maintenable, évolutive, et capable d'interagir avec des API externes ainsi qu'avec d'autres applications. Nous avons adopté une architecture basée sur le modèle **trois tiers** . Ce schéma représente ce dernier :

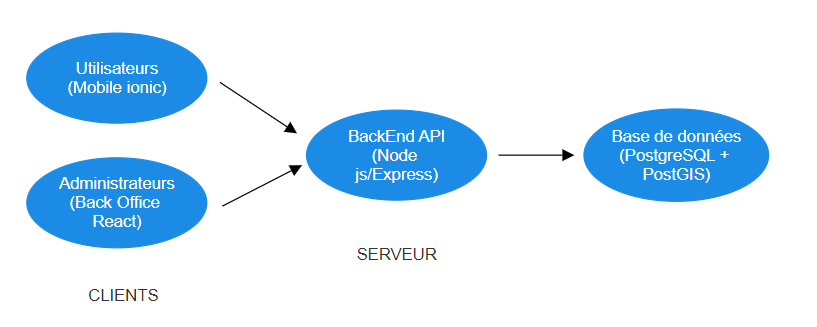


Figure 3 : Architecture trois tiers

L’architecture logicielle est une vue tournée sur l’organisation interne des logiciels du système informatique. Nous pouvons la représenter de la manière suivante

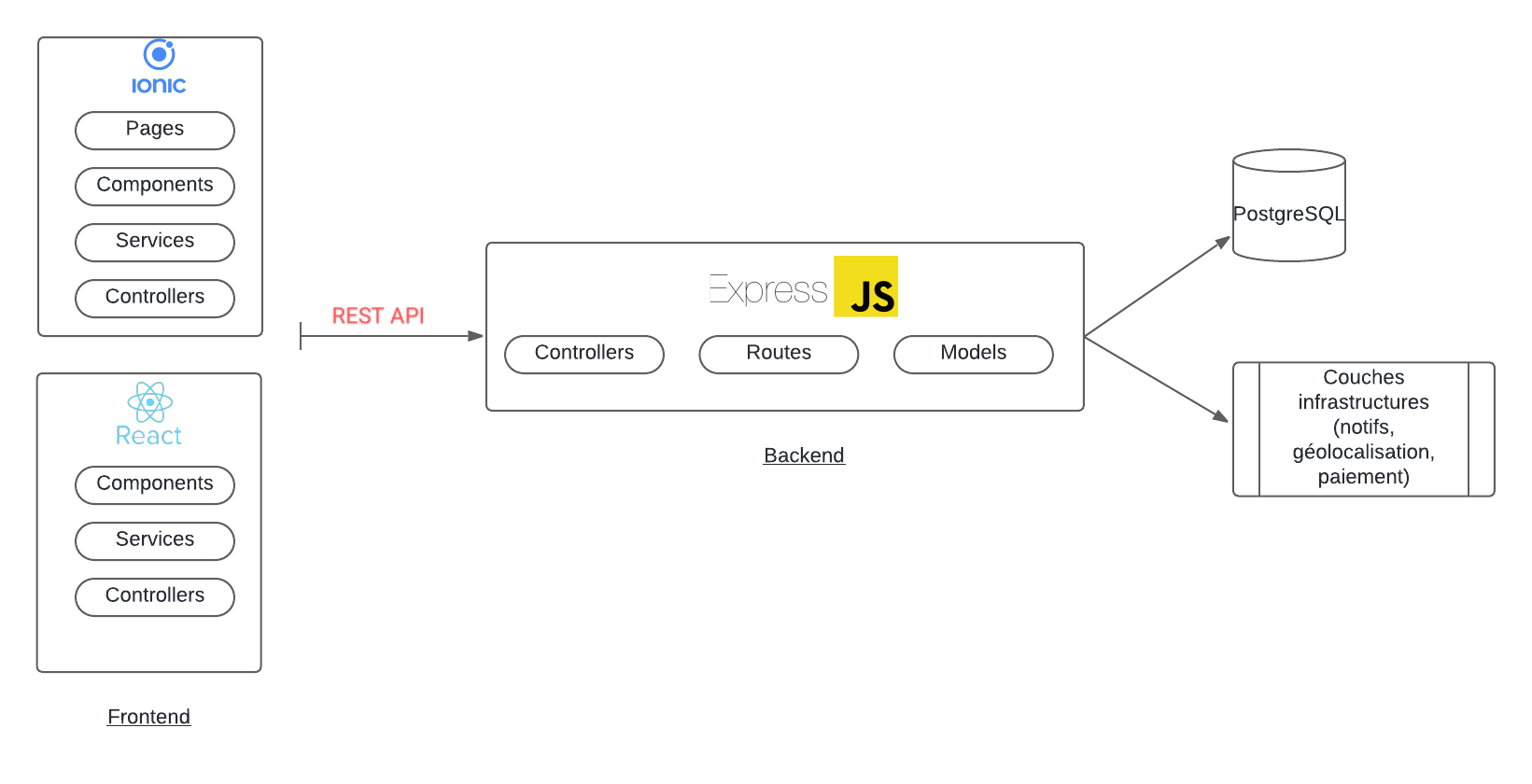


Figure 3 :Architecture du logiciel

### Conception

Concrètement, afin de réaliser le projet, on a ajouté :

• 17 tables

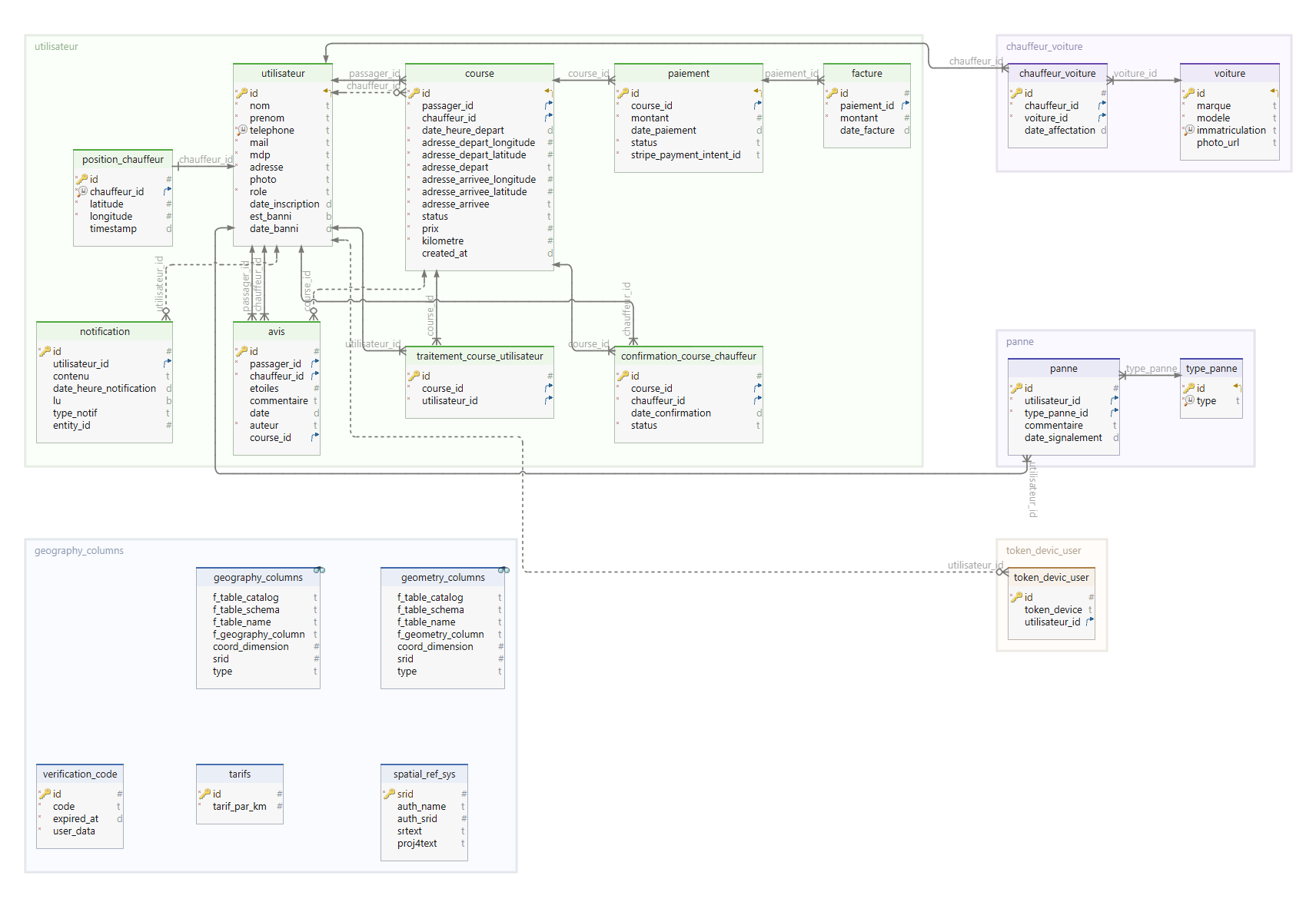
• 17 classes

• 14 contrôleurs

• 42 écrans

Nous pouvons trouver ci-dessous le modèle conceptuel de la base de données du projet :

Figure 4 : Modèle conceptuel de la base de données



## Développement par fonctionnalité ou module

### Création d’un Logo et identité visuelle

* + - 1. **Logo**

Le logo communique efficacement les aspects principaux de l'application : mobilité, rapidité et facilité d'utilisation. Visuellement attrayant et mémorable, il aide à capter et retenir l'attention des utilisateurs. La lettre 'M' y est stylisée pour rappeler les courbes d'une route, évoquant le service de réservation de véhicules avec chauffeur (VTC). Ce design incarne l'idée de déplacement, de trajet et de mobilité, des éléments au cœur de l'application.  
 En adoptant un style simple et épuré, le logo vise également à instaurer une sensation de confiance et d’accessibilité. Sa forme dynamique incite les utilisateurs à explorer le service sans appréhension, en véhiculant une image moderne et accueillante. L'intégration d'une représentation graphique minimaliste permet de renforcer l'identité visuelle de l'application et de lui conférer un aspect fluide et intuitif, en phase avec l'ère numérique. Cette approche garantit que le logo est facilement reconnaissable, qu'il inspire la fiabilité et qu'il reflète un service de qualité à un coût abordable.

* + - 1. **Identité visuelle**

**Objectifs de l'identité visuelle :** L'objectif principal de l'identité visuelle de l’application MOOV est de créer une image de marque mémorable, attractive et accessible pour les utilisateurs. Cette identité doit refléter la simplicité, la confiance et la modernité, des qualités essentielles dans le domaine des services de réservation de véhicules avec chauffeur (VTC). L'identité visuelle vise à transmettre l'idée d'un service pratique et abordable, inspirant aux utilisateurs une relation de confiance et d'accessibilité.

**Concept et symbolisme du logo :** Le logo de MOOV est centré sur la lettre "M", qui n'est pas seulement l'initiale du nom mais également un symbole de mobilité et de trajet. Conçu pour rappeler les courbes et les méandres d'une route, le "M" représente un itinéraire fluide et flexible, illustrant l'expérience simple et efficace que MOOV souhaite offrir. Ce design intègre des choix de couleurs et de formes qui inspirent la confiance, tout en reflétant une image moderne et dynamique.

**Public cible :** L'identité visuelle de MOOV a été créée pour s'adresser à un large public, comprenant aussi bien des professionnels que des jeunes et des familles recherchant un moyen de transport fiable et flexible. En veillant à ce que l'apparence du logo et des éléments visuels soient à la fois simples et modernes, nous souhaitons répondre aux attentes de cette cible en projetant un service qui soit aussi innovant que accessible, sans être intimidant ni financièrement prohibitif.

**Processus créatif :** Le processus créatif a débuté par une étude des logos d'autres entreprises de transport et d'applications mobiles. Plusieurs esquisses ont été réalisées pour intégrer des éléments de mouvement et de route. Après plusieurs essais, le concept de la lettre "M" stylisée en forme de chemin s'est imposé comme le meilleur choix, car il synthétise l'essence même du service MOOV tout en restant simple et facilement identifiable.

**Cohérence avec les valeurs du service :** Le logo et l'identité visuelle de MOOV reflètent des valeurs de confiance, de modernité et d'accessibilité. La simplicité des formes, associée à une apparence soignée et dynamique, incarne les engagements de l'application envers la transparence et la fiabilité. Ces valeurs sont au cœur de l'expérience que MOOV souhaite offrir : un service de transport sans complications, centré sur la satisfaction et la sécurité de ses utilisateurs.

**Utilisation de l'identité visuelle :** L'identité visuelle de MOOV se décline sur l'application mobile, le site web et tous les supports de communication, numériques comme imprimés. Elle crée une cohérence visuelle qui renforce la crédibilité et la reconnaissance de la marque. Sur les supports de communication, cette identité visuelle professionnelle et uniforme permet de rendre MOOV facilement identifiable, quel que soit le support utilisé par les utilisateurs.

Figure 5 : Identité visuelle



### Création d’interface Utilisateur et Expérience utilisateur (UI/UX)

Pour aborder la création de l'interface utilisateur (UI) de l'application et du site web MOOV, il est essentiel de s'appuyer sur une approche centrée sur l'utilisateur, alliant esthétique visuelle et facilité d'utilisation. Le processus de conception UI et UX a été guidé par des étapes méthodiques visant à optimiser l'expérience utilisateur. Ce parcours vise à garantir une interface intuitive et agréable qui reflète l’identité visuelle de MOOV et répond aux attentes d’une expérience de navigation fluide, cohérente et accessible sur tous les supports.

**Recherche et analyse des besoins des utilisateurs**

* Comprendre les attentes et les préférences des utilisateurs cibles.
* Étudier les parcours utilisateurs pour identifier les fonctionnalités prioritaires.

**Définition des parcours utilisateurs (User Flows)**

* Cartographier les étapes clés des parcours utilisateur pour garantir une navigation intuitive.
* Concevoir les interactions clés comme la réservation de véhicule, le suivi du chauffeur, et la gestion des notifications.

**Conception des wireframes**

* Créer des maquettes basse fidélité pour structurer l'information et visualiser l’interface générale.
* Définir les éléments interactifs, les boutons d’action, et les zones de contenu de manière claire.

**Prototypage interactif**

* Développer des prototypes haute fidélité pour tester l’apparence et le ressenti de l'application.
* Intégrer l’identité visuelle **MOOV** (couleurs, typographies, iconographie) pour une cohérence esthétique.



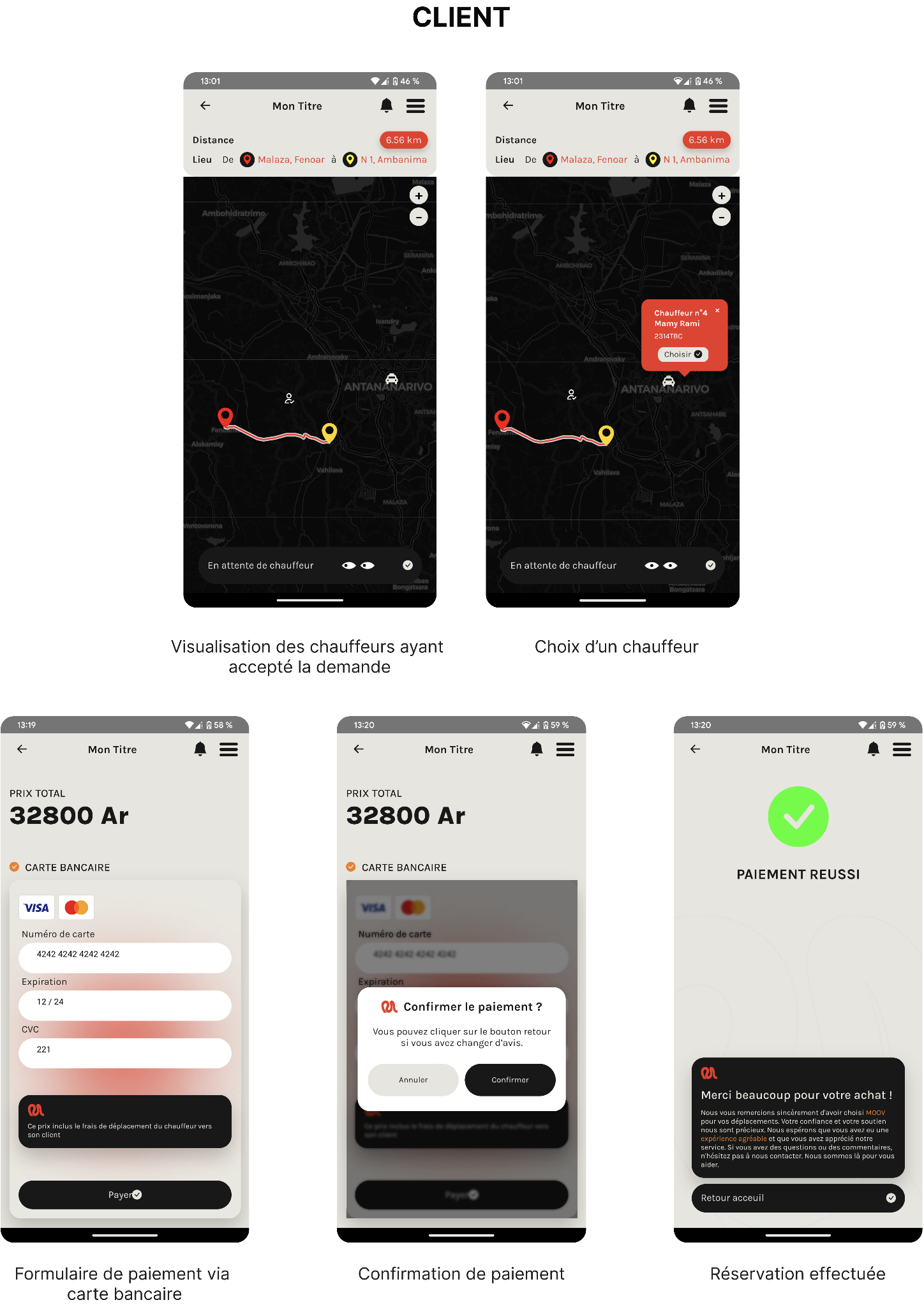
Figure 9 :Écran d'accueil

**Tests utilisateurs et ajustements UX**

* Mener des tests avec des utilisateurs réels pour identifier les points d’amélioration.
* Adapter l'ergonomie et optimiser les éléments interactifs selon les retours.

**Développement de l’interface graphique finale**

* Intégrer les éléments graphiques finaux en s’assurant de la réactivité et de l’accessibilité sur différents appareils (mobile et desktop).
* Veiller à une navigation fluide et à une disposition claire et fonctionnelle des informations.

Figure 10 :Écrans d’interface utilisateur des étapes du paiement 

### Développement des interfaces mobile et web

Pour développer les interfaces mobile et web de l'application MOOV, plusieurs éléments clés ont été intégrés pour garantir une expérience utilisateur fluide, intuitive et visuellement cohérente. Le design UI/UX, réalisé en amont, a servi de base pour structurer chaque composant en respectant les normes de taille et d'accessibilité, tant pour les éléments de navigation que pour les fonctionnalités essentielles. Chaque page, qu'elle soit destinée aux clients, aux conducteurs ou à l'administration, est structurée pour répondre aux besoins spécifiques de l’utilisateur, tout en respectant la charte graphique de l’application et en maintenant une uniformité visuelle sur toutes les plateformes.

* **Normes de taille et accessibilité des composants**

Des normes spécifiques ont été appliquées pour garantir la lisibilité et la maniabilité des composants, en respectant les tailles de polices et les proportions adaptées aux interfaces mobiles et web.

* **Respect de la charte graphique et uniformité visuelle**

Tous les composants ajoutés par les API respectent les couleurs, styles et typographies définis, pour assurer une cohérence visuelle entre les différentes sections de l’application.

* **Liste des Pages Mobile et Web**

**Client :**

* **Création de réservation (map)** : Interface pour sélectionner l’adresse de départ et destination ainsi que la date du course via la carte interactive.
* **Paiement** : Module de paiement pour finaliser la réservation avec leurs pages de confirmation.
* **Avis** : Page de notation pour laisser un avis sur le service après chaque course.
* **Profil** : Section de gestion du profil avec statistique de course et avis global avec sa modification.
* **Liste des réservations en attente** : Affiche les réservations en attente de confirmation avec leurs détails.
* **Historique de réservation** : Visualisation des réservations passées pour référence avec leurs détails.
* **Facture** : Génération et consultation des factures de chaque course avec leurs détails.
* **Notifications** : Affichage des notifications reçues, y compris la confirmation de réservation et les mises à jour.
* **Formulaire de création de panne de service** : Formulaire pour signaler une panne ou un problème technique.

**Conducteur :**

* **Liste des demandes de réservation** : Affichage des réservations disponibles pour acceptation avec leurs détails.
* **Liste d'attribution de réservation** : Réservations attribuées et prêtes à être effectuées avec leurs détails.
* **Historique de réservation** : Historique des courses terminées avec leurs détails.
* **Visualisation de la destination (map)** : Affichage de la destination sur la carte pour chaque réservation attribuée.
* **Avis** : Section pour donner des avis sur les passagers.
* **Profil** : Gestion du profil conducteur avec statistique de course et avis global avec sa modification.
* **Formulaire de création de panne de service** : Formulaire pour signaler les problèmes techniques ou pannes.

**Administration :**

* **Tableau de bord et statistiques** : Affichage des indicateurs clés de performance et des statistiques.
* **Liste des clients** : Gestion des clients avec options de bannissement .
* **Liste des conducteurs** : Gestion des chauffeurs avec options de bannissement
* **Ajout des membres (admin et conducteur)** : Formulaire d’ajout de nouveaux administrateurs et conducteurs avec option de modification et de bannissement.
* **Assignation de voiture à un chauffeur** : Interface pour attribuer un véhicule spécifique à un chauffeur.
* **Formulaire de création de voiture** : Ajout de nouveaux véhicules dans la flotte.
* **Notifications** : Module de réception de notifications sur les incidents.

### Authentification et Inscription

Ce module traite des fonctionnalités de connexion et d'inscription, qui sont essentielles pour sécuriser et personnaliser l'expérience utilisateur. Dans la partie administrateur du site web, le processus de connexion permet aux administrateurs d'accéder à l'interface de gestion. Côté application mobile, l'inscription des clients leur permet de créer un compte afin de réserver des courses et de bénéficier des services proposés. Nous aborderons également la fonctionnalité de réinitialisation du mot de passe pour les utilisateurs qui auraient oublié leurs identifiants, garantissant ainsi une accessibilité fluide et sécurisée pour tous les utilisateurs de l'application et du site.

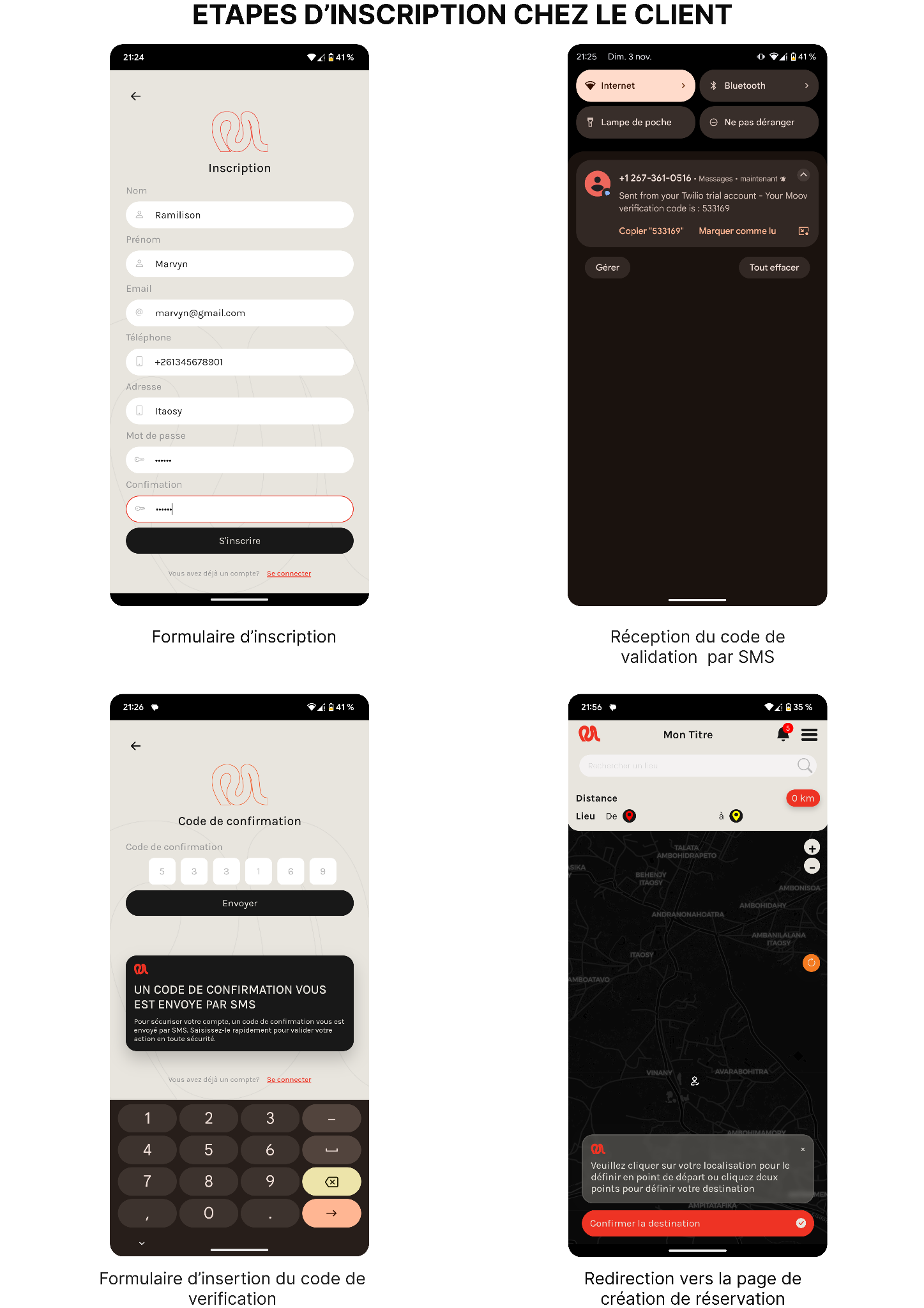
#### Inscription avec vérification de code par SMS

Le module d'inscription permet aux utilisateurs de créer un compte en saisissant leurs informations personnelles via un formulaire. Pour assurer la validité du numéro de téléphone et sécuriser l'accès, un code de vérification est envoyé par SMS à l'utilisateur grâce au service Twilio. Une fois le code validé, l'utilisateur est redirigé vers la page d'accueil de réservation.

La validation par SMS permet à l'entreprise de s'assurer que chaque utilisateur inscrit est bien une personne réelle, évitant ainsi les inscriptions frauduleuses et augmentant la fiabilité de la base de données clients. En confirmant leur numéro, les utilisateurs sécurisent leur compte, contribuant à renforcer la sécurité globale de l'application.

L'inscription inclut les étapes suivantes :

* **Étape 1** : L’utilisateur complète le formulaire d’inscription.
* **Étape 2** : L’application génère un code unique et utilise l’API Twilio pour envoyer un SMS avec ce code au numéro de l’utilisateur.
* **Étape 3** : L’utilisateur entre le code reçu dans l’application.
* **Étape 4** : Le serveur vérifie le code ; s’il est correct, le compte est activé et l’utilisateur est redirigé vers la page de réservation.

Figure 11 : Écrans des étapes de l’inscription

#### Mot de passe oublié avec authentification de code par SMS

Lorsqu'un utilisateur oublie son mot de passe, il peut se rendre sur la page "Mot de passe oublié" de l'application. Il est invité à entrer son numéro de téléphone, qui est ensuite utilisé pour envoyer un code de validation par SMS via l'API Twilio. Cette fonctionnalité assure non seulement la sécurité des informations de l'utilisateur, mais également une expérience utilisateur fluide.

Le mot de passe oublié inclut les étapes suivant :

**Étape 1** : L'utilisateur accède à la page "Mot de passe oublié".

**Étape 2** : L'utilisateur entre son numéro de téléphone et clique sur "Envoyer le code".

**Étape 3** : L'utilisateur reçoit un SMS contenant le code de validation.

**Étape 4** : L'utilisateur entre le code reçu dans le champ prévu.

**Étape 5** : Si le code est correct, l'utilisateur est redirigé vers un formulaire pour entrer un nouveau mot de passe.

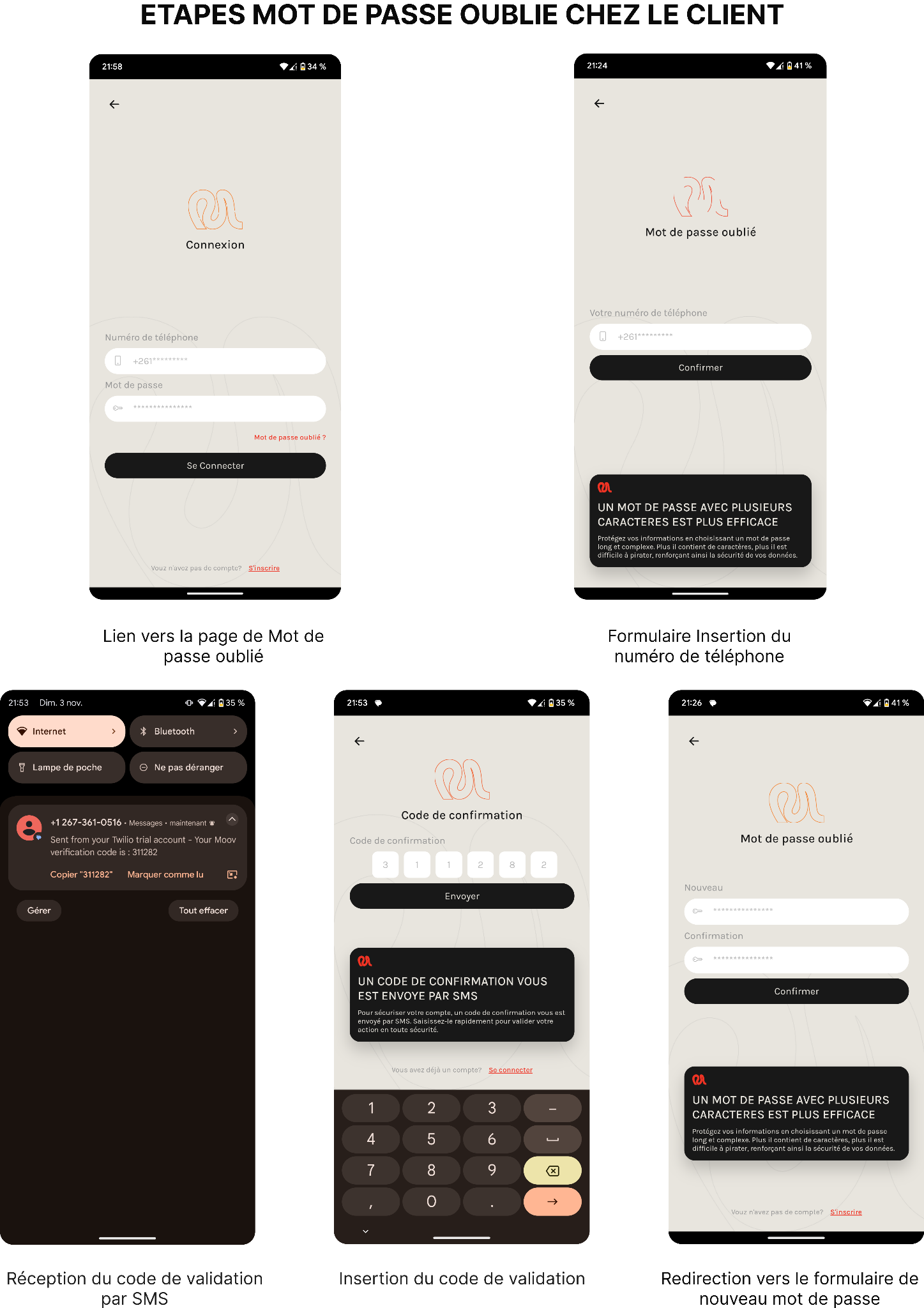


Figure 12 :Écrans des étapes du mot de passe oublié

### Module de notification push

Le module de **Notification Push** joue un rôle central dans l'expérience utilisateur de l’application MOOV, facilitant la communication en temps réel entre les chauffeurs, les clients, et l’administration. Il garantit que chaque utilisateur est informé immédiatement des mises à jour clés concernant les courses : de la demande initiale jusqu’aux retours en fin de trajet.

Pour mettre en place ce système, nous avons utilisé **Firebase Cloud Messaging (FCM)**, qui permet l’envoi et la gestion de notifications sur les appareils mobiles et le web. Grâce à FCM, l'application peut envoyer des notifications en fonction d'événements spécifiques, comme une demande de course ou l'attribution d'un chauffeur. Ce module repose sur la capacité de Firebase à gérer les jetons uniques de chaque appareil, assurant ainsi que les messages sont correctement envoyés aux utilisateurs ciblés, dans un flux sécurisé et fiable.

* **Notification push pour le chauffeur lors d’une demande de course**

Cette notification est envoyée au chauffeur dès qu'un client soumet une demande de course, l'informant de la disponibilité de cette nouvelle course. Cela lui permet de réagir rapidement et de consulter les détails pour accepter ou refuser la demande.

* **Notification push pour le client lors de l’acceptation de la course**

Une fois qu’un chauffeur a accepté la course, une notification est envoyée au client pour confirmer que la demande a bien été prise en charge. Cela rassure le client, l'informant du début du processus et facilitant la gestion de son trajet.

* **Notification push pour le chauffeur lors de l’attribution de la course**

Lorsque le client sélectionne un chauffeur, que ce dernier soit proche ou non de son point de départ, il passe au paiement pour confirmer la course. À la confirmation, une notification push est envoyée au chauffeur pour l'informer de l'attribution de la course. Le chauffeur peut alors consulter les détails du trajet et organiser son itinéraire en conséquence.

* **Notification push pour le client et le chauffeur pour les avis sur la course**

À la fin d’une course, une notification est envoyée au client et au chauffeur lorsqu'ils reçoivent un avis l'un de l'autre. Cela permet à chacun de prendre connaissance des retours sur l’expérience de course, améliorant ainsi la qualité de service et la satisfaction client.

Figure 13 :Écrans des différentes notification push

### Gestion des historiques de réservation de chaque client et chauffeur

Ce module permet aux clients et aux chauffeurs de consulter l’historique de leurs réservations passées. Il affiche une liste détaillée de toutes les réservations effectuées, incluant les informations de chaque trajet, telles que la date, l’heure, le point de départ, la destination, le prix, la distance parcourue, la voiture en service durant la course et le statut de la réservation.

L’historique de réservation est crucial pour offrir une transparence et un suivi aux utilisateurs. Il permet aux clients de revoir leurs trajets précédents et aux chauffeurs de gérer leur activité. Ce module améliore l'expérience utilisateur, renforce la confiance et répond aux besoins de traçabilité des réservations pour l’entreprise.



Figure 14 : Écrans des historiques de réservation chez le client et le conducteur

### Gestion des historiques chez l’administrateur

Ce module permet à l’administrateur de consulter l’ensemble des réservations effectuées, en accédant facilement aux détails de chaque réservation. Il offre une recherche par nom de client ou de chauffeur, ainsi qu’un filtrage des réservations par période spécifique (date de début et date de fin), améliorant la navigation et la gestion des données.

L'historique est essentiel pour la gestion et le suivi des opérations au sein de l’entreprise. Il donne aux administrateurs une vue d’ensemble sur les réservations, permettant d’analyser l’activité, d'identifier des tendances, et d’assurer la qualité du service. En centralisant l’historique des réservations, l’entreprise renforce son contrôle et sa capacité de gestion.

Le module est développé en utilisant des requêtes pour récupérer et afficher l’ensemble des réservations, avec des options de filtrage par date et de recherche par client ou chauffeur. La fonction de défilement dynamique permet d’afficher instantanément les détails sans nécessiter de clics supplémentaires. Lorsque l'administrateur sélectionne une réservation dans la liste, le détail de la course s'affiche de manière dépliée pour permettre un accès rapide et simplifié aux informations essentielles. Voici les informations disponibles dans le détail de chaque réservation :

* **Informations sur le client** : Nom ,Numéro de téléphone
* **Informations sur le chauffeur** : Nom ,Numéro de téléphone
* **Détails de la course** :
  + **Lieu de départ et de destination** : Adresse de départ et adresse de destination précises
  + **Date et heure de la réservation** : Horaires précis pour suivre l'historique de la course
  + **Prix** : Montant total de la course, permettant un suivi financier direct

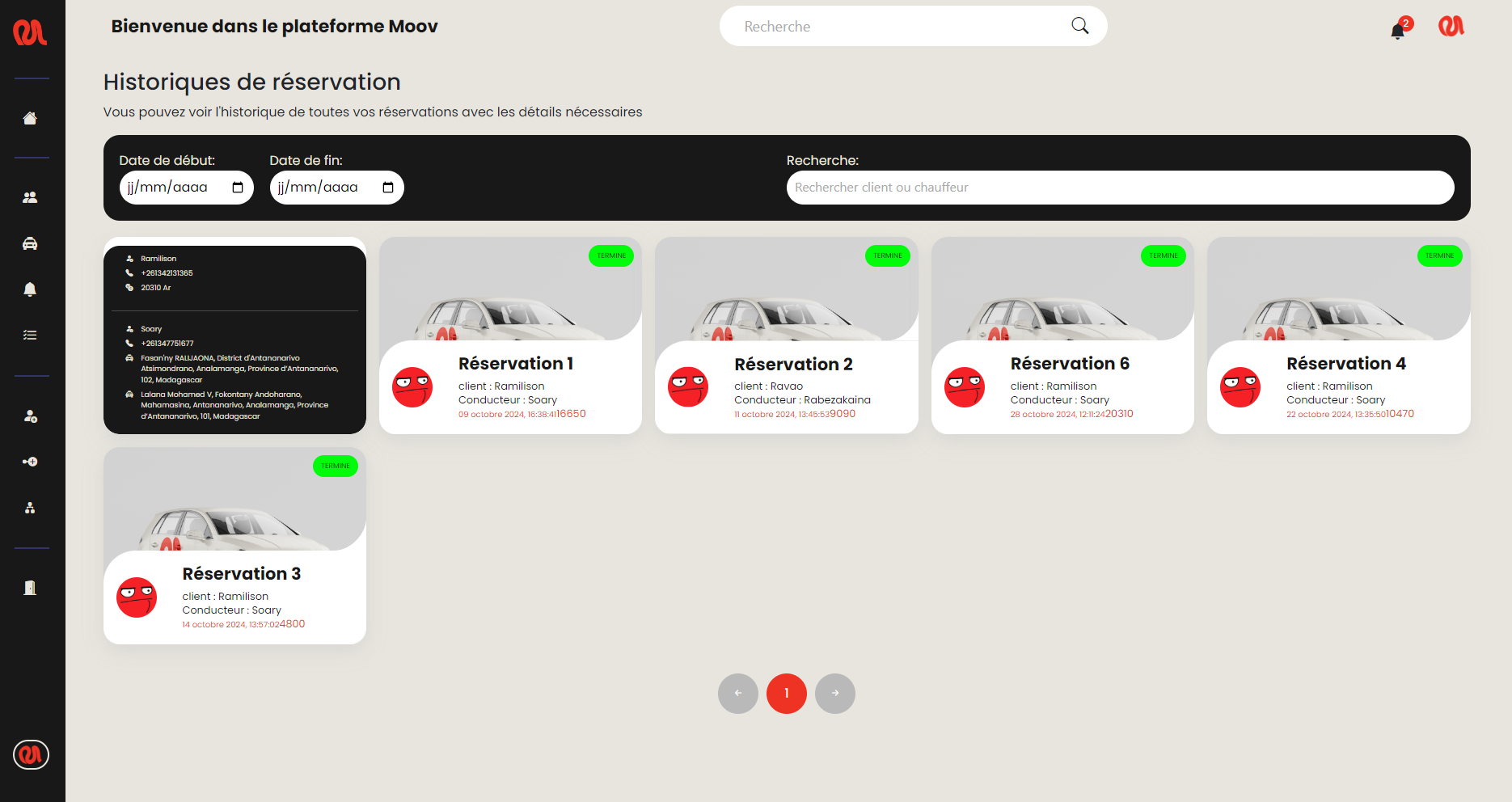


Figure 13 : Écrans des historiques de réservation chez l’administrateur

### Gestion des véhicules

Le module de gestion de voiture permet d’ajouter, de modifier, de supprimer, et de rechercher des véhicules dans le système de manière centralisée chez l’administrateur. Il comprend :

* **Formulaire de création de voiture** : pour ajouter les informations d'un véhicule (marque, modèle, immatriculation,photo.).
* **Liste des voitures** : avec des options de modification et suppression pour chaque voiture.
* **Importation CSV** : pour faciliter l'ajout en masse de voitures dans le système. Elle est accompagnée de messages d'erreur lorsqu'une voiture existe déjà avec la même immatriculation.
* **Barre de recherche** : permettant de rechercher des voitures par critères (marque, modèle, immatriculation, photo). La recherche est dotée d'une suggestion automatique et ne nécessite donc pas de bouton de validation.

Ce module est utile pour optimiser la gestion du parc de véhicules, essentiel pour une entreprise de VTC. Il simplifie l’ajout et la mise à jour des informations des véhicules et permet de visualiser facilement la liste complète, garantissant une gestion rapide et efficace du parc automobile. D’autre part, il assure que chaque véhicule est enregistré avec précision et facilite sa gestion. En centralisant la gestion des informations des véhicules, ce module aide également à maintenir des données de haute qualité et à réduire les erreurs. Voici les étapes pour la creation de voiture :

* **Étape 1** : L'administrateur accède au formulaire de création de voiture et saisit les informations.
* **Étape 2** : L'application enregistre les informations dans la base de données et met à jour la liste.
* **Étape 3** : Si un fichier CSV est importé, les véhicules sont ajoutés en lot.
* **Étape 4** : L'administrateur utilise la barre de recherche pour trouver un véhicule, puis peut le modifier ou le supprimer si nécessaire.

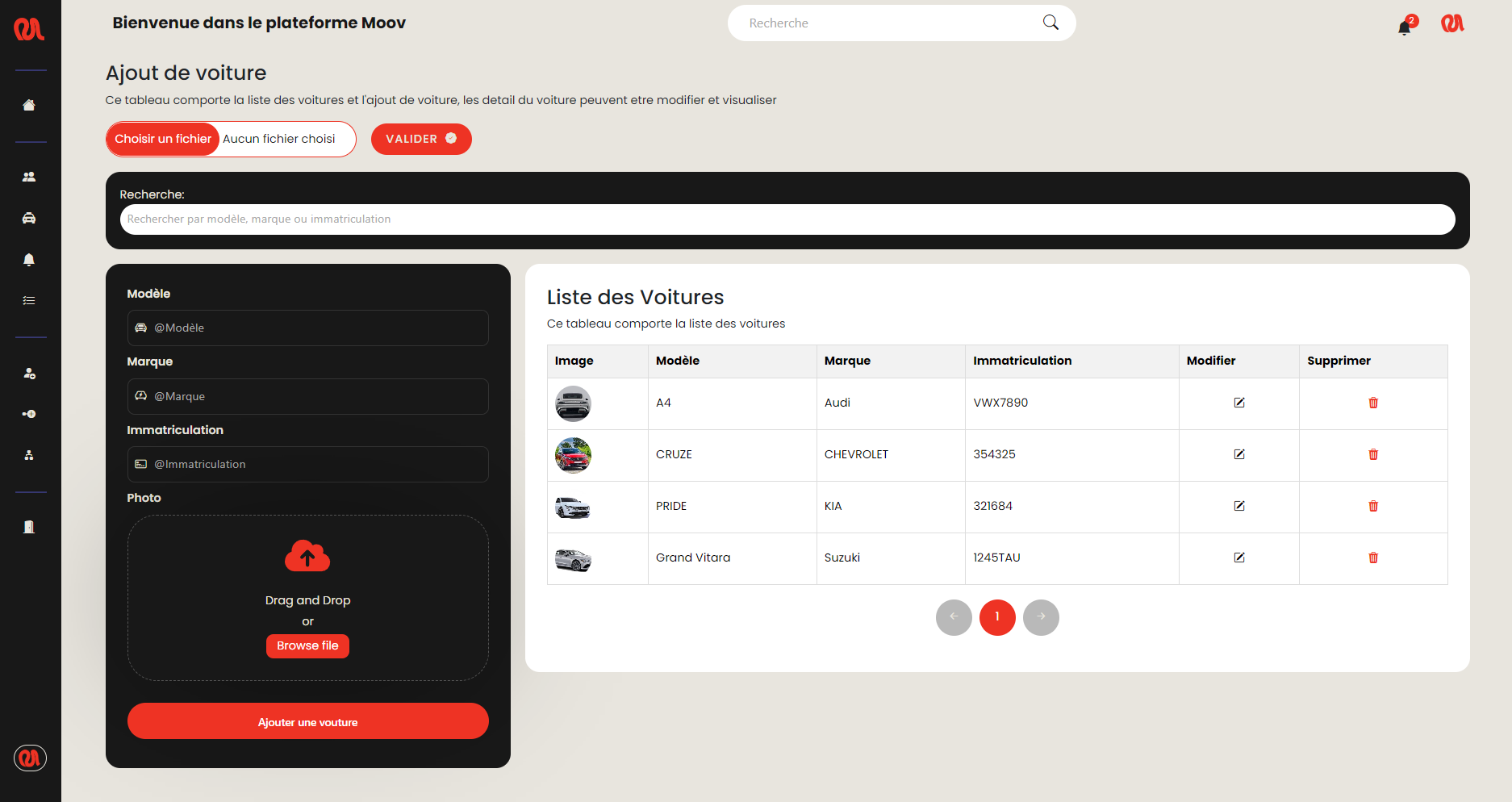


Figure 12 :Écrans de la gestion des voitures

### Gestion des utilisateurs

Ce module centralise la gestion du profil et des activités de chaque utilisateur, qu’il soit client ou chauffeur, en offrant des informations détaillées sur les réservations, la distance parcourue, les avis, et la possibilité de mise à jour des informations personnelles.

#### Fonctionnalités clés :

* **Profil Utilisateur :** Affiche les informations spécifiques de l’utilisateur connecté, comme le nom, le numéro de téléphone, et le rôle (client ou chauffeur).
* **Total de Distance Effectuée :** Indique la distance totale parcourue par l’utilisateur, permettant un suivi de ses trajets.
* **Total de réservations effectuées :** Indique le nombre total de réservations complétées. Un bouton est disponible pour rediriger vers la création d'une nouvelle réservation.
* **Avis Global et Statut :** Montre un récapitulatif des avis reçus par l’utilisateur, avec une évaluation visuelle (positif, moyen, mauvais).
* **Modification du Profil :** Un bouton est accessible sur la photo de profil pour éditer les informations de l’utilisateur ainsi que sa photo de profil.

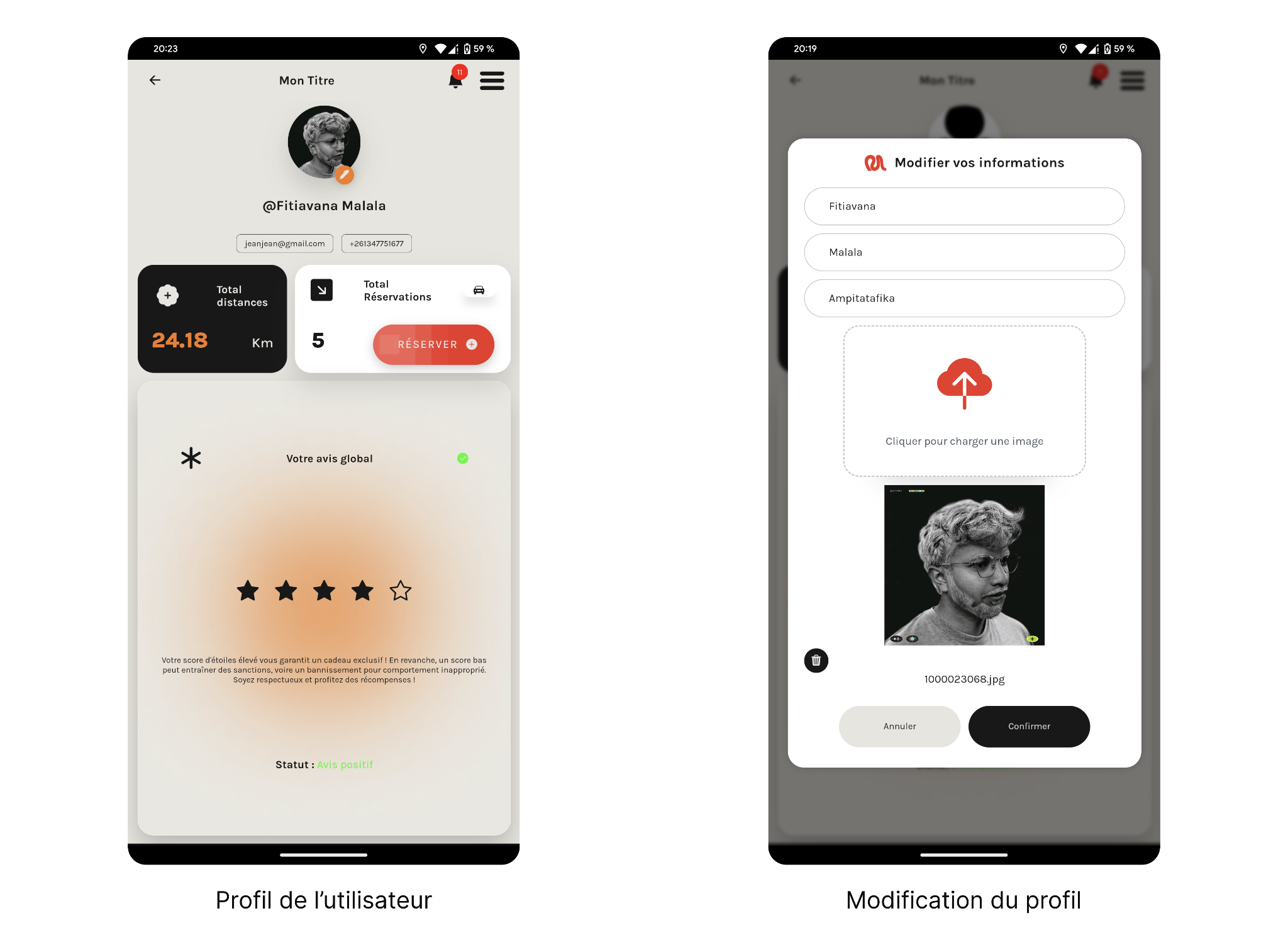
Le développement de ce module répond à un besoin essentiel : permettre aux utilisateurs d’avoir un contrôle et une vue d’ensemble de leurs activités et de leurs performances. Il améliore l’expérience utilisateur en offrant une interface intuitive et complète pour la gestion des profils et de l’historique des courses. Cela permet également à l'entreprise de renforcer la transparence et la fidélisation en offrant des informations précises et facilement accessibles.

Figure 12 :Écrans de la gestion des utilisateur dans l’application mobile

## État d’Analyse et Statistiques

### État numéro 1

### État numéro 2

### Statistique numéro 1

Tableau 1 : Classement TIOBE 2014 de quelques langages de programmation (**exemple**)

| Position Jan ‘14 | Position Jan ‘13 | Language | Share Jan ‘14 | Trend |
| --- | --- | --- | --- | --- |
| 1 | 1 | Java | 26.2% | -0.6% |
| 2 | 2 | PhP | 13.2% | -1.6% |
| 3 | 6 | Python | 10.2% | +1.3% |
| 4 | 3 | C# | 9.6% | -0.4% |
| 5 | 4 | C++ | 8.9% | 0% |
| 6 | 5 | C | 8.1% | -0.2% |
| 7 | 7 | JavaScript | 7.6% | +0.3% |

Il ne s’agit pas de reproduire ce tableau, surtout si vous ne faites pas une comparaison entre les langages de programmation : ceci est un exemple de présentation de données statistiques.  
REMARQUEZ que **des données chiffrées doivent être datées et récentes**.

## Problèmes rencontrés et solutions

Lors de la réalisation de ce projet, nous avons fait face à des problèmes majeurs :

* La création de l'identité visuelle et du logo n'était pas une tâche facile, car l'entreprise voulait un logo qui représente la confiance, la modernité, combiné avec le thème VTC. L'identité devait être dynamique, incitant les gens à utiliser le service sans leur faire penser qu'il serait trop cher. J'ai donc créé un logo simple, symbolisant une route et la lettre "M", initiale de MOOV, afin de les combiner pour obtenir un logo dynamique et fluide, rappelant un chemin en forme de M, plat et adouci. Ce logo simple permet d'intégrer le projet dans l'évolution numérique sans crainte concernant la sécurité ou le coût du service.
* Les graphiques de course et de revenus effectués par semaine, mois et année ne correspondaient pas à l'affichage prévu, car les données s'affichent toujours sur le premier jour de la semaine et le premier mois de l'année. Le problème venait du fait que les dates récupérées dans la requête SQL du backend n'étaient pas appelées correctement dans l'affichage, ce qui faisait que le graphique affichait le premier jour, mois et année par défaut.
* Un autre problème concernait la table "Avis" dans la base de données. Le souci était que dans une course, il y avait un id\_passager et un id\_chauffeur. Quand une course se terminait, les deux côtés envoyaient chacun un avis à l'autre. Il fallait donc savoir à qui l'avis était destiné et qui l'avait envoyé. Pour résoudre ce problème, l'ajout d'une colonne "auteur" dans la table "avis" était nécessaire pour savoir qui était l'envoyeur de l'avis et ainsi récupérer facilement l'avis destiné à chacun.
* La build du projet Android était une tâche très précise car il y avait plusieurs dépendances qui n'étaient pas compatibles entre elles, comme la version de Capacitor qui devait être plus récente (par exemple "@capacitor/android": "6.1.1") pour que Ionic 7 l'accepte. Un autre problème concernait la version de Gradle lors de l'exportation de l'APK dans Android Studio. La version de Gradle du projet ne devait pas être plus récente que celle supportée par Android Studio 2021. La solution consistait à définir directement la version de Gradle dans le fichier build.gradle en “gradle:8.1.4” pour permettre à Android Studio de faire la build sans erreur et de télécharger toutes les dépendances nécessaires.

# Évaluation du projet et connaissances acquises

## Bilan pour l'entreprise

L’objectif durant les trois mois de stage a été atteint pour l’entreprise. La réalisation d’une application mobile pour la gestion de Véhicules de transport avec chauffeurs (VTC) intitulée MOOV a pu être développée. Les fonctionnalités essentielles sont déjà mises en place. Il ne reste plus qu’à perfectionner certaines fonctionnalités et ajouter quelques modules pour que l’application puisse être utilisée.

En tant que stagiaire, j’ai pu appuyer l’équipe en contribuant aux projets internes de l’entreprise. Ma présence était nécessaire étant donné que les autres collaborateurs étaient occupés et ne pouvaient pas gérer le projet à temps plein

## Bilan personnel

Ce stage fut une expérience très enrichissante pour moi car il m’a permis de découvrir le milieu professionnel et le secteur privé. J’ai pu contribuer à la réalisation d’un projet qui a un impact significatif sur la vie quotidienne des gens et pour l'évolution numérique de notre pays. Ce fut vraiment gratifiant et intéressant de voir un projet aboutir.

Durant ces trois mois, j’ai appris beaucoup de choses notamment la réalisation d’un véritable projet d’entreprise dans un environnement de travail, les contraintes professionnelles, la mise en pratique de mes connaissances académiques, le développement de nouvelles méthodes de travail, l’apprentissage des bonnes pratiques d’une entreprise, le fonctionnement et l’utilisation de plusieurs technologies.

Ce stage m’a permis également de développer de nouvelles relations professionnelles et d’améliorer mes compétences sociales (savoir-être, savoir-faire, respect de la hiérarchie, prise de décision et prise d’initiative).

## Extension et évolution de l’application

La plateforme et l’application mobile sont actuellement disponibles et fonctionnelles. Les tâches demandées ont toutes été achevées, et la structure du projet est bien en place. L’extension du projet permettra d’offrir une expérience plus attrayante et moins fatigante pour les utilisateurs. Les points suivants répondront à cette demande :

- Mise en place d'animations attractives sur les composants clés, tels que les composants de réservation et le suivi de la destination, pour rendre l’application encore plus fluide et inciter les utilisateurs à l'utiliser davantage.

- Mise en place

- Intégration d’une fonctionnalité offrant des bonus ou des cadeaux aux meilleurs chauffeurs dans l’administration.

- Ajout d'une fonctionnalité permettant de visualiser l'itinéraire du chauffeur vers le point de départ du rendez-vous.

- Information en temps réel pour le client lorsque le chauffeur arrive au point de rendez-vous (point de départ).

- Utilisation d’un service de carte (map) avec plus de fonctionnalités, comme le recentrage automatique du point de départ sur la position du chauffeur et du client lorsque la course est en cours.

- Possibilité d’assigner un autre chauffeur à un client en cas de panne technique pendant une course.

# Conclusion

Ainsi, j’ai effectué mon stage de fin d’études en Licence au sein de l’entreprise WYLog Madagascar. Ces trois mois de travaux m’ont permis d’appliquer les connaissances théoriques que j’ai acquises durant mes années d’études à l’université. Ce projet m’a permis de consolider mes compétences en design, communication digitale ainsi qu’en développement mobile et web. Cette expérience a renforcé mon sens de l’organisation, m’a permis de gagner en professionnalisme, et m’a sensibilisé à l’importance de la communication au sein d’une équipe technique.

Concernant Moov, les objectifs fixés au début du stage ont été globalement atteints :

- Un identité visuelle facile à reconnaître et ne donne pas des doutes sur le service

- Une expérience utilisateur et un design conformes au thème et faciles à utiliser.

- La mise en place d'une plateforme administrative pour le suivi des courses et des utilisateurs.

- La gestion des notifications push sur les avis reçus par les conducteurs

-les notifications push d’attribution d’une réservation à un chauffeur, et les notifications push de nouvelles réservations.

- Pour finir, la mise en place des tableaux de bord dans l’administration pour mieux gérer les revenus et les utilisateurs.

Parmi les principaux problèmes rencontrés, l’envoi des notifications vers un téléphone mobile s’est avéré complexe. Une solution a été trouvée en intégrant un service externe de transmission d’informations comme Firebase pour assurer la communication.

Le projet a encore un potentiel d’amélioration, comme l’ajout d’une fonctionnalité permettant de visualiser tous les conducteurs sur une seule carte dans l’administration et de voir leur statut (s’ils sont en course ou non), afin que l’administrateur puisse leur assigner une course en conséquence. De nouveaux services pourraient également être ajoutés, comme un service de messagerie pour la communication entre chauffeurs et clients, des abonnements pour les utilisateurs réguliers, ou même l'intégration de services de réservation pour d’autres catégories de véhicules.

# Bibliographie

* *Top 10 des meilleures bases de données à choisir en 2024.* Disponible sur :

<https://www.ambient-it.net/meilleures-base-de-donnees/> (consulté le 19 - 10 - 2024)

* *Web service*. Disponible sur :

<https://www.oracle.com/fr/cloud/definition-web-service/> (consulté le 21 - 10 . 2024)

* *Comment concevoir le logo idéal* , Disponible sur

<https://www.snoweb.io/fr/web-design/creation-de-logo/> (consulté le 07 - 08 - 2024)

* *Référence d’application de réservation de taxi, Disponible sur*

<https://richestsoft.com/fr/blog/top-10-taxi-booking-apps/> (consulté le 13 - 09 - 2024)

# Annexe

S’il y en a : pour isoler des éléments techniques afin de laisser le mémoire plus lisible, surtout par des non spécialistes.

1. TIC : Technologies de l’Information et de la Communication [↑](#footnote-ref-0)