

# **Rapport de projet en React Native**

**Filière : Informatique & Réseaux**

---

**Conception et développement d'une application mobile de  
gestion et de suivi des transports de biens – AlloHonda -**

---



**Réalisé par :**

EL HATTAB Maria

DAFRAOUI El Mehdi

AIT OUIAZZANE Ali

**Encadré par :** Mme. Naji Zineb

**Année Universitaire : 2025-2026**

# Dédicace

Nous dédions ce travail en premier lieu à nos familles, pour leur soutien constant, leur patience et leurs encouragements tout au long de notre parcours universitaire. Leur présence et leur confiance ont été une source de motivation essentielle dans la réalisation de ce projet.

Nous adressons également nos sincères remerciements à nos enseignants et à notre encadrant, pour leur accompagnement, leurs conseils et leur disponibilité, qui ont grandement contribué à l'orientation et à l'aboutissement de ce travail.

Enfin, nous dédions ce projet à toutes les personnes qui, de près ou de loin, ont participé à sa réalisation, que ce soit par leurs conseils, leur aide technique ou leur soutien moral. À tous, nous exprimons notre profonde gratitude.

# Remerciements

Nous exprimons nos sincères remerciements à notre professeure et encadrante, pour son accompagnement, sa disponibilité et ses précieux conseils tout au long de la réalisation de ce projet. Ses orientations et ses remarques constructives nous ont permis d'améliorer la qualité de notre travail et d'approfondir nos connaissances.

Nous adressons également nos remerciements à l'ensemble de nos enseignants, qui ont contribué à notre formation académique et au développement de nos compétences tout au long de notre parcours universitaire.

Nous remercions aussi nos camarades et collègues pour leur aide, leurs échanges et leur esprit de collaboration, qui ont été essentiels à l'avancement de ce projet. Enfin, nous exprimons notre gratitude à toutes les personnes qui ont participé, de près ou de loin, à la réalisation de ce travail, par leur soutien, leurs conseils ou leur disponibilité.

# Résumé

AlloHonda est une application mobile de mise en relation entre des clients ayant des besoins de transport de biens et des chauffeurs disposant de véhicules utilitaires (Honda, camionnettes, camions). Elle vise à faciliter la gestion des demandes de transport, l'affectation des missions, ainsi que le suivi en temps réel des trajets.

L'application repose sur une architecture client–serveur avec un frontend développé en React Native et un backend en .NET (MVC), connecté à une base de données SQL Server. Elle intègre des services externes tels que Google Maps API pour la géolocalisation et les itinéraires, ainsi que Firebase.

Le système prend en charge trois interfaces principales :

- **Interface Client** : création de demandes de transport, suivi en temps réel des trajets, consultation de l'historique et gestion du profil.
- **Interface Chauffeur** : gestion des demandes reçues, acceptation ou refus des missions, suivi des trajets, consultation des statistiques et des gains.
- **Interface Administrateur** : supervision globale de la plateforme, gestion des utilisateurs (clients et chauffeurs), suivi des demandes et génération de statistiques.

AlloHonda se distingue par son suivi en temps réel, sa gestion multi-rôles, sa sécurité des données et sa scalabilité, offrant ainsi une solution moderne, fiable et performante pour le transport de biens.

**Mots-clés** : Application mobile, Transport de biens, React Native, .NET, SQL Server, Firebase, Google Maps API, Géolocalisation, Notifications push, Architecture MVC, Suivi en temps réel, Multi-rôles.

# Abstract

AlloHonda is a mobile application that connects clients with transportation needs for goods and drivers who own utility vehicles (Honda vehicles, vans, trucks). It aims to facilitate the management of transport requests, the assignment of missions, as well as the real-time tracking of trips.

The application is based on a client–server architecture with a frontend developed in React Native and a backend built with .NET (MVC), connected to a SQL Server database. It integrates external services such as Google Maps API for geolocation and routing, as well as Firebase.

The system supports three main interfaces:

- **Client Interface:** creation of transport requests, real-time trip tracking, consultation of transport history, and profile management.
- **Driver Interface:** management of received requests, acceptance or rejection of missions, trip tracking, and consultation of statistics and earnings.
- **Administrator Interface:** global supervision of the platform, management of users (clients and drivers), monitoring of requests, and generation of statistics.

AlloHonda stands out for its real-time tracking, multi-role management, data security, and scalability, offering a modern, reliable, and high-performance solution for goods transportation.

**Keywords:** Mobile application, Goods transportation, React Native, .NET, SQL Server, Firebase, Google Maps API, Geolocation, Push notifications, MVC architecture, Real-time tracking, multi-role.

# Acronymes

- **API** : Application Programming Interface
- **MVC** : Model View Controller
- **UML** : Unified Modeling Language
- **GPS** : Global Positioning System
- **UI** : User Interface
- **UX** : User Experience
- **BD** : Base de donnees
- **SQL** : Structured Query Language
- **JWT** : JSON Web Token
- **REST** : Representational State Transfer
- **CRUD** : Create, Read, Update, Delete
- **SDK** : Software Development Kit
- **HTTP** : HyperText Transfer Protocol
- **HTTPS** : HyperText Transfer Protocol Secure
- **JSON** : JavaScript Object Notation
- **IDE** : Integrated Development Environment
- **RN** : React Native
- **VS Code** : Visual Studio Code
- **.NET** : Framework de developpement Microsoft

# La table des matières

Dédicace .....	i
Remerciements.....	ii
Résumé .....	iii
Abstract.....	iv
Acronymes .....	v
La table des matières .....	vi
Liste des figures .....	viii
Introduction Générale.....	1
Chapitre 1. Contexte général du projet .....	3
1. Introduction .....	3
2. Contexte du projet .....	4
2.1. Problématique du transport de biens.....	4
2.2. Évolution des technologies mobiles et des services connectés .....	4
2.3. Cadre académique et technique du projet .....	4
3. Objectifs du projet .....	5
3.2. Objectifs spécifiques .....	5
3.3. Résultats attendus .....	5
4. Cahier des charges .....	6
4.1. Besoins fonctionnels.....	6
4.2. Besoins non fonctionnels .....	7
5. Benchmark.....	7
5.1 Solutions internationales.....	7
5.2 Solutions locales .....	8
5.3 Enseignements clés .....	8
5.4 Positionnement de AlloHonda.....	8
6. Mission du projet .....	9
7. Livrables.....	9
7.1 Code source .....	9

7.2 Rapport technique .....	10
7.3 Lien GitHub / GitLab .....	10
8. Méthodologie de gestion de projet.....	11
9. Conclusion.....	12
Chapitre3. Analyse et conception .....	14
1. Introduction .....	14
2. Identification des acteurs .....	14
3. Langage de modélisation UML .....	14
4. Diagrammes de cas d'utilisation .....	15
5. Diagrammes de séquence .....	16
6. Diagramme de classe .....	17
7. Conclusion.....	19
Chapitre 4. Réalisation .....	21
1. Introduction .....	21
2. Architecture .....	21
3. Outils et technologies .....	22
3.1 Environnement logiciel .....	22
3.2 Technologies utilisées .....	24
3.3 Le modèle MVC : .....	25
4. Interfaces de l'application .....	26
4.1 Interface Authentification : .....	26
4.2 Interface Client : .....	27
4.3 Interface Chauffeur : .....	29
4. Interface Admin : .....	30
5. Conclusion .....	31
Conclusion générale et perspectives .....	32
Webographie .....	33



# Liste des figures

FIGURE 1 : GESTION DE PROJET SUR TAIGA AVEC LA METHODE KANBAN.....	11
FIGURE 2 : DIAGRAMME DE CAS D'UTILISATION.....	16
FIGURE 3: DIAGRAMME DE SEQUENCE .....	17
FIGURE 4: DIAGRAMME DE CLASSES.....	18
FIGURE 5: ARCHITECTURE DU PROJET .....	22
FIGURE 6: LOGO DE VISUAL STUDIO CODE .....	22
FIGURE 7: LOGO DE VISUAL STUDIO .....	23
FIGURE 8 : LOGO DE TAIGA.....	23
FIGURE 9 : LOGO DE MODELIO .....	23
FIGURE 10 : LOGO DE REACT NATIVE .....	24
FIGURE 11 : LOGO DE .NET.....	24
FIGURE 12 : LOGO DE SQL SERVER.....	24
FIGURE 13 : LOGO DE FIREBASE .....	24
FIGURE 14 : LOGO DE GOOGLE MAPS .....	25
FIGURE 15 : LOGO DE GITHUB.....	25
FIGURE 16 : LOGO DE POSTMAN .....	25
FIGURE 17 : ARCHITECTURE DU MODELE MVC .....	26
FIGURE 18 : PAGES D'AUTHENTIFICATION .....	27
FIGURE 19 : INTERFACES CLIENTS.....	28
FIGURE 20 : INTERFACES CHAUFFEURS .....	29
FIGURE 21 : INTERFACES ADMINISTRATEUR .....	30

# Introduction Générale

Avec l'évolution rapide des technologies mobiles et la généralisation des smartphones, les applications mobiles jouent aujourd'hui un rôle central dans l'optimisation des services de transport et de logistique. La digitalisation de ces services permet non seulement de simplifier les échanges entre les différents acteurs, mais aussi d'améliorer la rapidité, la traçabilité et la fiabilité des opérations de transport de biens.

Dans ce contexte, le transport de marchandises à petite et moyenne échelle représente un enjeu majeur pour les particuliers comme pour les professionnels. La recherche d'un chauffeur disponible, le choix d'un véhicule adapté, la coordination des trajets et le suivi en temps réel des livraisons constituent souvent des tâches complexes, chronophages et peu optimisées lorsqu'elles sont réalisées de manière traditionnelle. Ces difficultés soulignent la nécessité de solutions numériques intelligentes, accessibles et performantes.

C'est dans cette optique que s'inscrit le projet AlloHonda, une application mobile de mise en relation entre des clients ayant des besoins de transport de biens et des chauffeurs disposant de véhicules utilitaires. L'objectif principal de cette application est de faciliter la gestion des demandes de transport, d'optimiser l'affectation des missions et d'assurer un suivi en temps réel des trajets, tout en garantissant une expérience utilisateur fluide et sécurisée.

Le développement de l'application repose sur une architecture client-serveur moderne, intégrant un frontend mobile conçu avec React Native et un backend développé en .NET selon le modèle MVC. Cette architecture permet une séparation claire des responsabilités, une meilleure maintenabilité du code et une évolutivité du système. La gestion des données est assurée par une base de données relationnelle SQL Server, garantissant la cohérence, la sécurité et la persistance des informations.

L'application AlloHonda prend en charge trois types d'utilisateurs distincts : le client, le chauffeur et l'administrateur. Chaque acteur dispose d'une interface spécifique adaptée à ses besoins et à ses responsabilités.

# Chapitre 1 :

# Contexte général du projet

# Chapitre 1. Contexte général du projet

## 1. Introduction

Le développement des technologies mobiles et des systèmes d'information a profondément transformé les services de transport et de logistique. Les applications mobiles constituent aujourd'hui des outils essentiels pour faciliter la communication entre les utilisateurs, optimiser la gestion des ressources et améliorer la qualité des services proposés. Dans le domaine du transport de biens, ces technologies permettent d'automatiser les processus, de réduire les délais et d'assurer un meilleur suivi des opérations.

Dans ce cadre, le projet **AlloHonda** a été conçu comme une solution numérique visant à répondre aux besoins croissants de mise en relation entre des clients ayant des demandes de transport de marchandises et des chauffeurs disposant de véhicules utilitaires adaptés. Cette application mobile s'inscrit dans une démarche de modernisation des services de transport, en proposant une plateforme centralisée, intuitive et performante.

À travers ce projet, l'objectif est non seulement de proposer une solution technique fonctionnelle, mais également de mettre en pratique les concepts fondamentaux du développement d'applications mobiles, de la conception logicielle, de la modélisation des données et de l'intégration de services tiers. Ce travail s'inscrit ainsi dans une démarche académique visant à démontrer la capacité à concevoir, développer et déployer une application mobile complète, répondant à des exigences fonctionnelles et non fonctionnelles bien définies.

## 2. Contexte du projet

Le projet de Allo Honda s'inscrit dans un environnement où le domaine de logistique connaît une transformation rapide, portée par la digitalisation sur tous les niveaux.

### 2.1. Problématique du transport de biens

Le transport de biens à petite et moyenne échelle est souvent confronté à plusieurs difficultés, notamment la recherche d'un chauffeur disponible, l'adéquation entre le type de marchandise et le véhicule, ainsi que le manque de visibilité sur l'avancement des livraisons. Les méthodes traditionnelles, basées sur des appels téléphoniques ou des intermédiaires informels, sont généralement peu efficaces, coûteuses en temps et sujettes à des erreurs de coordination.

Ces contraintes mettent en évidence la nécessité d'une solution numérique capable de simplifier et d'automatiser la gestion des demandes de transport tout en assurant une meilleure communication entre les différents acteurs.

### 2.2. Évolution des technologies mobiles et des services connectés

L'essor des smartphones et des technologies mobiles a favorisé l'émergence de nouvelles plateformes de services basées sur la géolocalisation, le suivi en temps réel et les notifications instantanées. Les applications de transport exploitent aujourd'hui des services tels que la cartographie numérique, le GPS et les systèmes de notification push pour améliorer l'expérience utilisateur et optimiser les opérations.

Dans ce contexte, l'utilisation de technologies comme **React Native**, **.NET**, **SQL Server**, **Google Maps API** et **Firebase** permet de concevoir des applications performantes, évolutives et sécurisées, répondant aux exigences actuelles du marché.

### 2.3. Cadre académique et technique du projet

Le projet AlloHonda s'inscrit dans un cadre académique visant à mettre en pratique les connaissances acquises en développement mobile, en génie logiciel et en conception de systèmes d'information. Il constitue une application concrète des notions d'architecture

logicielle, de modélisation des données, de gestion des rôles utilisateurs et d'intégration de services externes.

Ce projet a été réalisé en collaboration au sein d'une équipe, favorisant le travail collectif, la répartition des tâches et l'utilisation de bonnes pratiques de développement logiciel.

### 3. Objectifs du projet

Mettre en place une solution ergonomique, sécurisée et performante qui permette aux clients d'allouer des camions selon leurs besoin tout en assurant une cohérence et communication directe entre eux et chauffeur avec un suivi du transit en temps réel.

#### 3.2. Objectifs spécifiques

Les principaux objectifs du projet AlloHonda sont les suivants :

- Concevoir une application mobile permettant la création et la gestion des demandes de transport de biens.
- Mettre en place un système de mise en relation entre clients et chauffeurs basé sur la disponibilité et le type de véhicule.
- Assurer un suivi en temps réel des trajets grâce à la géolocalisation.
- Développer une gestion multi-rôles avec des interfaces distinctes pour le client, le chauffeur et l'administrateur.
- Garantir la sécurité des données et la fiabilité du système.
- Intégrer des notifications push pour informer les utilisateurs des changements de statut des demandes.

#### 3.3. Résultats attendus

À l'issue de ce projet, les résultats attendus sont :

- Une application mobile fonctionnelle et ergonomique répondant aux besoins des utilisateurs.
- Une plateforme centralisée facilitant la gestion des demandes de transport et des missions.
- Une amélioration de la communication et de la transparence entre clients et chauffeurs.

- Un système évolutif pouvant intégrer de futures fonctionnalités telles que le paiement en ligne ou le chat en temps réel.
- Une démonstration concrète des compétences acquises en développement mobile, backend et intégration de services externes.

## 4. Cahier des charges

### 4.1. Besoins fonctionnels

Les besoins fonctionnels décrivent l'ensemble des fonctionnalités que l'application AlloHonda doit offrir afin de répondre aux attentes des utilisateurs et aux objectifs du projet. Ces besoins sont définis selon les différents types d'utilisateurs de la plateforme.

#### **Pour le client, l'application doit permettre :**

- L'inscription et l'authentification sécurisée.
- La consultation de la liste des chauffeurs disponibles avec leurs informations (type de véhicule, disponibilité).
- La création d'une demande de transport en précisant le lieu de départ, la destination, le type de marchandise, le poids et le volume.
- Le suivi en temps réel de l'état de la demande et du trajet du chauffeur.
- La consultation de l'historique des transports effectués.
- La gestion du profil personnel et des paramètres de l'application.

#### **Pour le chauffeur, l'application doit permettre :**

- L'inscription avec validation des informations et des documents.
- La réception des demandes de transport.
- L'acceptation ou le refus des missions.
- La visualisation de l'itinéraire et le suivi du trajet via une carte intégrée.
- La mise à jour du statut des missions (en cours, terminé).
- La consultation des statistiques et des gains réalisés.
- La gestion du profil et des informations du véhicule.

#### **Pour l'administrateur, l'application doit permettre :**

- La gestion des utilisateurs (clients et chauffeurs).
- La validation des comptes chauffeurs.
- La visualisation et le suivi global des demandes de transport.

- L'accès à des statistiques générales sur l'activité de la plateforme.

## 4.2. Besoins non fonctionnels

Les besoins non fonctionnels définissent les contraintes de qualité que doit respecter l'application AlloHonda afin d'assurer son bon fonctionnement.

Parmi les principaux besoins non fonctionnels, on distingue :

- **Performance** : l'application doit offrir des temps de réponse rapides, notamment lors de l'affichage des cartes et du suivi en temps réel.
- **Sécurité** : les données personnelles des utilisateurs doivent être protégées par des mécanismes d'authentification et de contrôle d'accès.
- **Disponibilité** : le système doit être accessible de manière continue, avec une tolérance minimale aux pannes.
- **Scalabilité** : l'architecture doit permettre une montée en charge progressive en fonction du nombre d'utilisateurs.
- **Ergonomie** : l'interface utilisateur doit être intuitive, simple à utiliser et adaptée aux appareils mobiles.
- **Maintenabilité** : le code doit être structuré de manière claire afin de faciliter les évolutions futures.

## 5. Benchmark

Afin de concevoir et développer efficacement cette application, il est nécessaire d'analyser les solutions déjà présentes sur le marché, qu'elles soient internationales ou locales. Ce benchmark permettra d'identifier les meilleures pratiques, d'évaluer les fonctionnalités clés proposées par la concurrence et de mettre en évidence les axes de différenciation pour AlloHonda.

### 5.1 Solutions internationales

À l'échelle internationale, plusieurs applications proposent des services de transport et de livraison basés sur la mise en relation entre utilisateurs et conducteurs. Des plateformes comme Uber Freight ou Lalamove offrent des fonctionnalités avancées telles que le suivi en temps réel, la gestion des itinéraires et le paiement en ligne. Ces solutions se distinguent par leur maturité technologique, leur grande base d'utilisateurs et leur couverture géographique étendue.



Cependant, ces applications sont souvent orientées vers des marchés spécifiques et peuvent ne pas répondre pleinement aux besoins locaux ou aux contraintes régionales.

## 5.2 Solutions locales

Au niveau local, les solutions de transport de biens reposent majoritairement sur des moyens traditionnels, tels que les appels téléphoniques ou les intermédiaires informels. Certaines applications locales commencent à émerger, mais elles restent limitées en termes de fonctionnalités, de suivi en temps réel et de gestion centralisée des utilisateurs.

Ces solutions manquent généralement d'une interface administrative complète et d'un système structuré de gestion des demandes et des chauffeurs.

## 5.3 Enseignements clés

L'analyse des solutions existantes met en évidence plusieurs éléments importants :

- ✓ L'importance du suivi en temps réel pour améliorer la transparence et la confiance des utilisateurs.
- ✓ La nécessité d'une gestion multi-rôles bien définie.
- ✓ L'impact positif des notifications instantanées sur la réactivité du système.
- ✓ Le besoin d'une solution adaptée au contexte local, tant sur le plan fonctionnel que technique.

Ces enseignements ont guidé la conception et le développement du projet AlloHonda.

## 5.4 Positionnement de AlloHonda

AlloHonda se positionne comme une solution locale, moderne et évolutive, dédiée au transport de biens à petite et moyenne échelle. Contrairement aux solutions internationales, elle est conçue pour répondre spécifiquement aux besoins du contexte local tout en intégrant des technologies récentes.

Grâce à son architecture basée sur React Native et .NET, son intégration de la géolocalisation et des notifications push, ainsi que sa gestion multi-rôles, AlloHonda offre une alternative fiable et performante aux solutions existantes, avec un fort potentiel d'évolution vers des fonctionnalités avancées telles que le paiement en ligne ou le chat en temps réel.

## 6. Mission du projet

La mission principale du projet AlloHonda consiste à concevoir et développer une application mobile complète dédiée au transport de biens, permettant la mise en relation efficace entre les clients, les chauffeurs et les administrateurs au sein d'une plateforme unique.

Ce projet vise à mettre en œuvre une solution numérique moderne capable de :

- ✓ Digitaliser le processus de demande et de gestion des transports de marchandises.
- ✓ Améliorer la communication et la coordination entre les différents acteurs du système.
- ✓ Exploiter les technologies mobiles et les services connectés pour assurer un suivi en temps réel des trajets.
- ✓ Appliquer les bonnes pratiques du développement logiciel, notamment en matière d'architecture, de sécurité et de maintenabilité.

Sur le plan académique, la mission du projet est également de permettre aux membres de l'équipe de mettre en pratique les connaissances théoriques acquises dans les domaines du développement mobile, du backend, de la gestion de bases de données et de l'intégration de services externes. Le projet AlloHonda constitue ainsi une expérience concrète de travail collaboratif, de gestion de projet et de développement d'une application professionnelle répondant à des exigences fonctionnelles et non fonctionnelles précises.

## 7. Livrables

Les livrables du projet AlloHonda représentent l'ensemble des éléments produits au cours du développement et remis dans le cadre de l'évaluation académique. Ils attestent de la réalisation effective du projet et de l'implication de tous les membres de l'équipe.

### 7.1 Code source

Le code source complet de l'application constitue le principal livrable technique du projet. Il est :

- Hébergé sur une plateforme de gestion de versions telle que GitHub ou GitLab.
- Organisé de manière claire et structurée (dossiers *components*, *screens*, *services*, *assets*, etc.).

- Versionné avec un historique de commits réguliers, clairs et explicites, reflétant l'évolution progressive du projet.
- Développé de façon collaborative, démontrant la participation active de tous les membres de l'équipe.

Un lien vers le dépôt du projet est fourni afin de permettre la consultation et l'évaluation du code.

## 7.2 Rapport technique

Un rapport technique au format PDF est fourni afin de documenter le projet. Ce rapport comprend :

- Une page de garde contenant le nom du projet, les noms des membres de l'équipe, la filière, l'encadrant et l'année universitaire.
- Une introduction générale et une présentation du projet.
- La description des fonctionnalités principales et de l'architecture globale du système.
- Les technologies utilisées, notamment React Native, .NET, SQL Server, Google Maps API et Firebase.
- Des captures d'écran illustrant les principaux écrans de l'application.
- Une section consacrée aux difficultés rencontrées et aux solutions mises en œuvre.
- Une conclusion accompagnée de perspectives d'évolution du projet.

Ce rapport permet de présenter le projet de manière structurée et pédagogique

## 7.3 Lien GitHub / GitLab

Un lien direct vers le dépôt GitHub ou GitLab est fourni comme livrable complémentaire. Ce lien permet :

- L'accès au code source du projet.
- La consultation de l'historique des commits.
- La vérification de la contribution de chaque membre de l'équipe au développement de l'application.

Ce livrable garantit la transparence du travail réalisé et facilite l'évaluation du projet sur le plan technique et collaboratif.

## 8. Méthodologie de gestion de projet

La réalisation du projet AlloHonda a été organisée selon une méthodologie de gestion de projet Agile, en s'appuyant sur l'outil Taiga et sur la méthode Kanban. Cette approche a permis d'assurer une meilleure planification des tâches, un suivi continu de l'avancement et une collaboration efficace entre les membres de l'équipe.

Un espace projet dédié a été configuré sur la plateforme Taiga. Cet espace comprend l'ensemble des éléments nécessaires à la gestion du projet, notamment le backlog des fonctionnalités, les tickets de travail et les différentes colonnes du tableau Kanban (à faire, en cours, terminé). Chaque fonctionnalité ou tâche technique a été formalisée sous forme de ticket, avec une description claire, des priorités et des responsables désignés.

La méthode Kanban a été adoptée afin de visualiser en temps réel l'état d'avancement du projet. Elle a permis de suivre précisément la progression des tâches, d'identifier rapidement les blocages et d'assurer une répartition équilibrée du travail entre les membres de l'équipe. Le déplacement des tickets entre les différentes colonnes du tableau a servi d'indicateur visuel de l'avancement global du projet.

Le suivi d'avancement a été démontré à travers la validation progressive des tâches, avec une visibilité claire sur les fonctionnalités terminées, celles en cours de développement et celles restant à implémenter. Cette approche a favorisé une amélioration continue du produit et une adaptation rapide aux éventuels changements ou contraintes rencontrées.

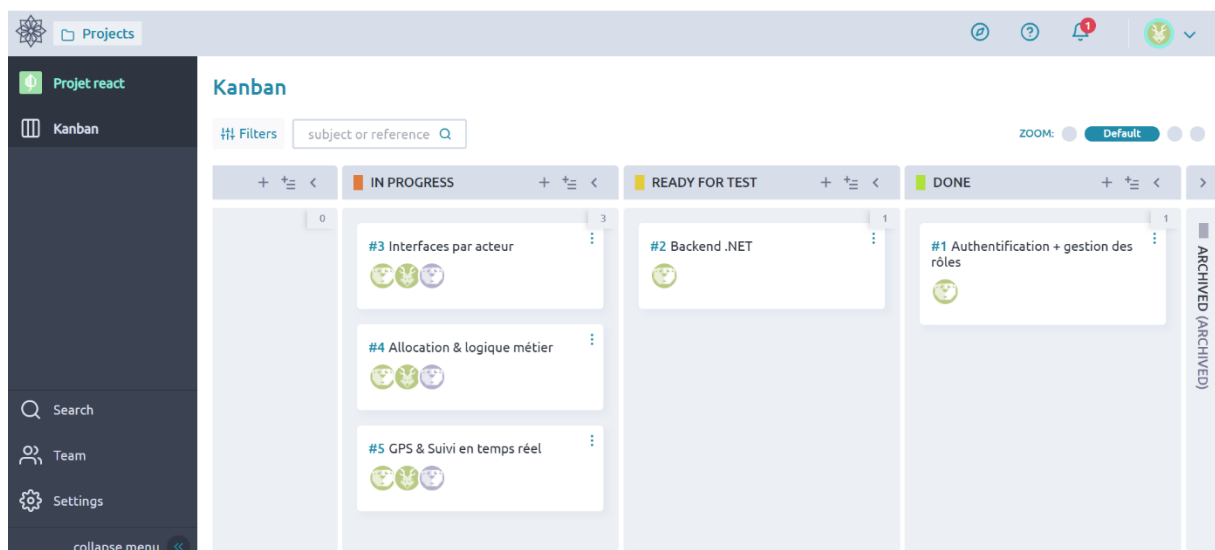


Figure 1 : Gestion de projet sur Taiga avec la méthode Kanban

## 9. Conclusion

Ce chapitre a permis de présenter le contexte général du projet AlloHonda, en mettant en évidence les motivations, les objectifs et les enjeux liés à la conception d'une application mobile dédiée au transport de biens. L'analyse du contexte a souligné les limites des solutions traditionnelles et la nécessité de recourir à des outils numériques modernes pour améliorer la gestion des demandes de transport, la communication entre les acteurs et le suivi des trajets.

Le cahier des charges a permis de définir clairement les besoins fonctionnels et non fonctionnels, constituant ainsi une base solide pour la conception et le développement de l'application. Le benchmark réalisé a mis en lumière les solutions existantes, tant au niveau international que local, et a permis de positionner AlloHonda comme une solution adaptée au contexte local, intégrant des fonctionnalités modernes et une architecture évolutive.

La mission du projet, les livrables attendus ainsi que la méthodologie de gestion adoptée ont également été détaillés, mettant en évidence une organisation structurée basée sur une approche Agile et l'utilisation de l'outil Kanban. Ces éléments ont contribué à assurer un suivi efficace de l'avancement du projet et une collaboration active entre les membres de l'équipe.

Ainsi, ce chapitre pose les fondements nécessaires à la compréhension globale du projet AlloHonda et prépare la transition vers les chapitres suivants, qui porteront sur la conception, l'architecture et la réalisation technique de l'application.

# **Chapitre 3 :**

## **Analyse et conception**

# Chapitre3. Analyse et conception

## 1. Introduction

La phase d'analyse et de conception constitue une étape essentielle dans le développement du projet **AlloHonda**. Elle permet de traduire les besoins fonctionnels exprimés dans le cahier des charges en modèles formels et compréhensibles, facilitant ainsi la conception technique et l'implémentation du système.

Dans ce chapitre, nous présentons l'identification des acteurs du système, le langage de modélisation utilisé ainsi que les principaux diagrammes UML (cas d'utilisation, séquence et classes) permettant de décrire le fonctionnement global et la structure interne de l'application.

## 2. Identification des acteurs

Le système AlloHonda repose sur trois acteurs principaux, chacun disposant de rôles et de responsabilités spécifiques :

**Client** : utilisateur qui crée des demandes de transport, suit les trajets et consulte l'historique.

**Chauffeur** : utilisateur chargé d'accepter ou de refuser les demandes et d'assurer le transport des biens.

**Administrateur** : responsable de la supervision globale du système, de la gestion des utilisateurs et du suivi des activités.

## 3. Langage de modélisation UML

Le langage UML (Unified Modeling Language) est un langage de modélisation standardisé utilisé dans le domaine de l'ingénierie logicielle. Il permet de représenter graphiquement la structure et le comportement d'un système afin de faciliter sa compréhension, sa conception et sa communication entre les différents acteurs du projet.

## Objectifs de l'UML :

Fournir une vision claire et partagée du système entre les développeurs, les chefs de projet et les utilisateurs.

Modéliser les besoins fonctionnels et techniques avant la phase d'implémentation.

Réduire les risques d'ambiguïtés en offrant une description normalisée.

Servir de base documentaire pour le suivi et l'évolution du projet.

## Types de diagrammes UML utilisés dans ce projet :

**Diagramme de cas d'utilisation (Use Case Diagram)** : décrit les interactions entre les acteurs (clients, chauffeurs, administrateurs) et le système.

**Diagramme de séquence** : illustre le déroulement chronologique des échanges entre les acteurs et le système.

**Diagramme de classes** : présente la structure des données et des objets du système.

Ainsi, UML constitue un outil central pour l'analyse et la conception du module d'entretien différé, garantissant que les besoins identifiés seront correctement traduits en solutions techniques.

## 4. Diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter les fonctionnalités offertes par le système AlloHonda du point de vue des utilisateurs. Il met en évidence les interactions entre les différents acteurs et le système, sans entrer dans les détails techniques de l'implémentation.

Dans le cadre du projet AlloHonda, trois acteurs principaux ont été identifiés : le Client, le Chauffeur et l'Administrateur. Chaque acteur interagit avec le système à travers un ensemble de cas d'utilisation correspondant à ses responsabilités. Le client peut s'inscrire, se connecter, créer une demande de transport, suivre le trajet et consulter l'historique de ses demandes. Le chauffeur, quant à lui, peut consulter les demandes reçues, accepter ou refuser une mission, suivre le trajet et consulter ses gains. L'administrateur dispose de fonctionnalités de gestion globale, telles que la gestion des utilisateurs, la validation des chauffeurs, le suivi des demandes et la consultation des statistiques.



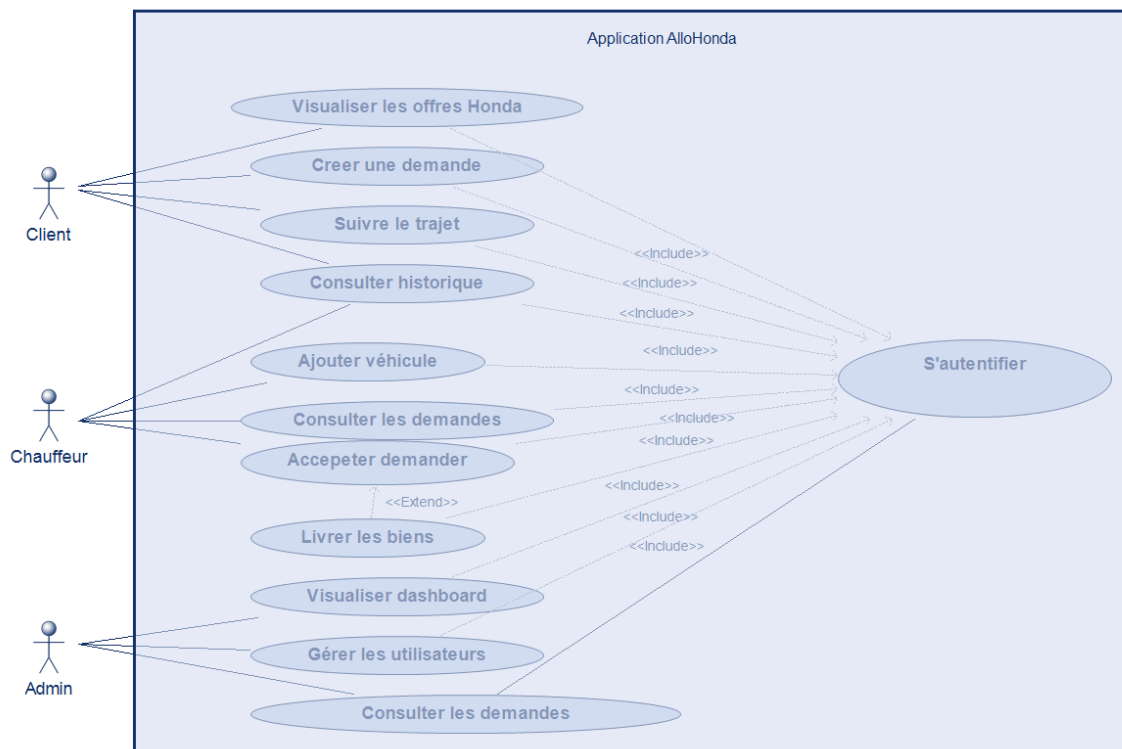


Figure 2 : Diagramme de cas d'utilisation

## 5. Diagrammes de séquence

Le diagramme de séquence permet de décrire le déroulement temporel des interactions entre les différents composants du système lors de l'exécution d'un scénario précis. Il met en évidence l'ordre des messages échangés entre les acteurs, l'application mobile, et la base de données.

Dans le projet AlloHonda, les diagrammes de séquence illustrent notamment la création d'une demande de transport par un client ainsi que l'acceptation d'une mission par un chauffeur.

Ce diagramme permet de mieux comprendre la logique fonctionnelle, les responsabilités de chaque composant et le fonctionnement de l'architecture client-serveur adoptée. Ils constituent un outil essentiel pour la phase d'implémentation, en clarifiant le comportement dynamique du système.

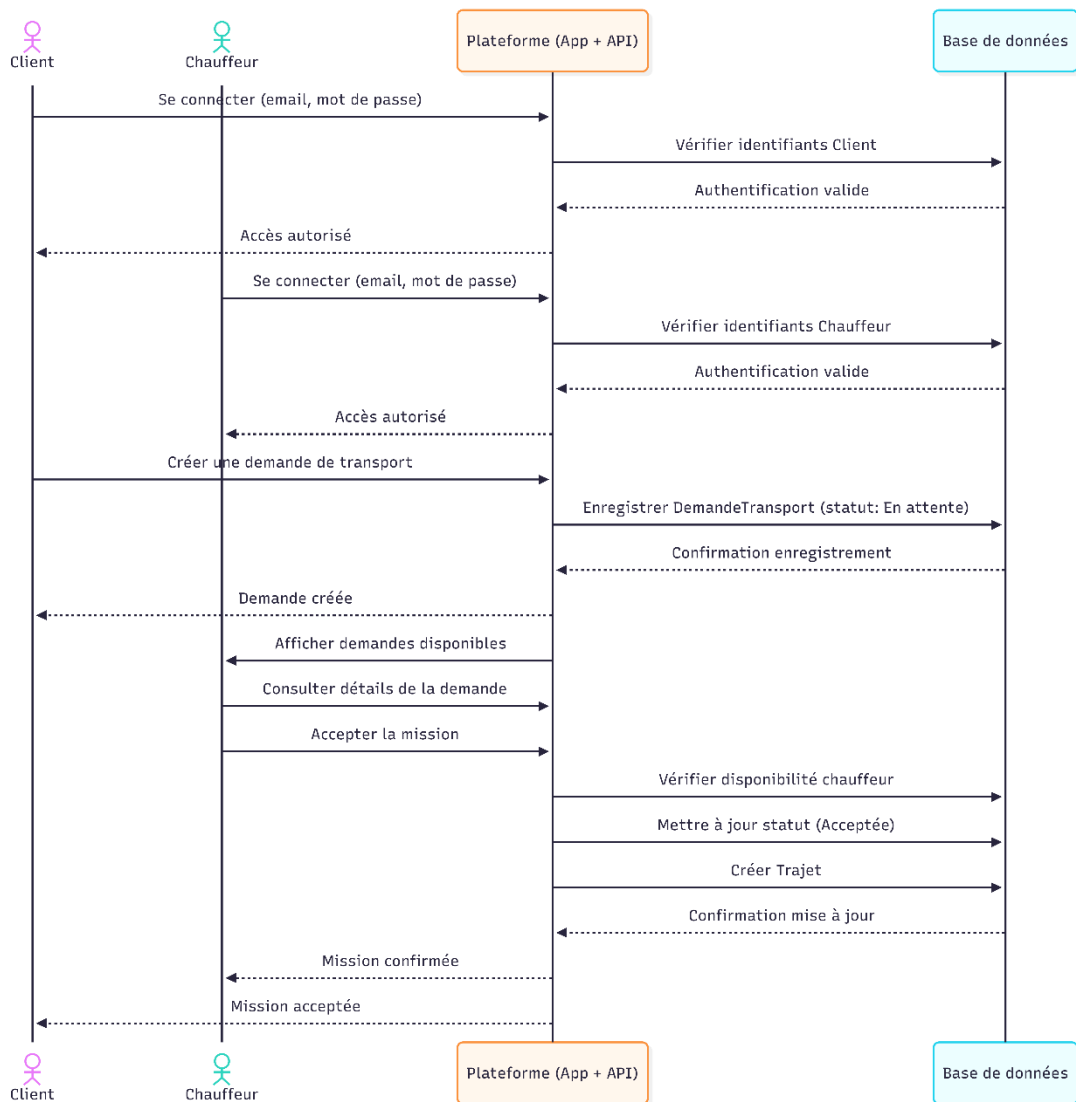


Figure 3: Diagramme de séquence

## 6. Diagramme de classe

Le diagramme de classes représente la structure statique du système AlloHonda. Il décrit les principales classes du système, leurs attributs, leurs relations ainsi que les mécanismes d'héritage entre les entités.

Dans ce projet, la classe Utilisateur constitue la classe de base, regroupant les informations communes à tous les types d'utilisateurs. Elle est spécialisée en trois sous-classes : Client, Chauffeur et Administrateur, chacune ajoutant des attributs spécifiques à son rôle.

Le diagramme inclut également les classes Véhicule et DemandeTransport, qui jouent un rôle central dans la gestion des missions de transport. Les relations entre les classes traduisent les liens existants entre les clients, les chauffeurs, les véhicules et les demandes.

Ce diagramme permet de structurer les données du système et sert de référence pour la conception de la base de données ainsi que pour l'implémentation des modèles côté backend.

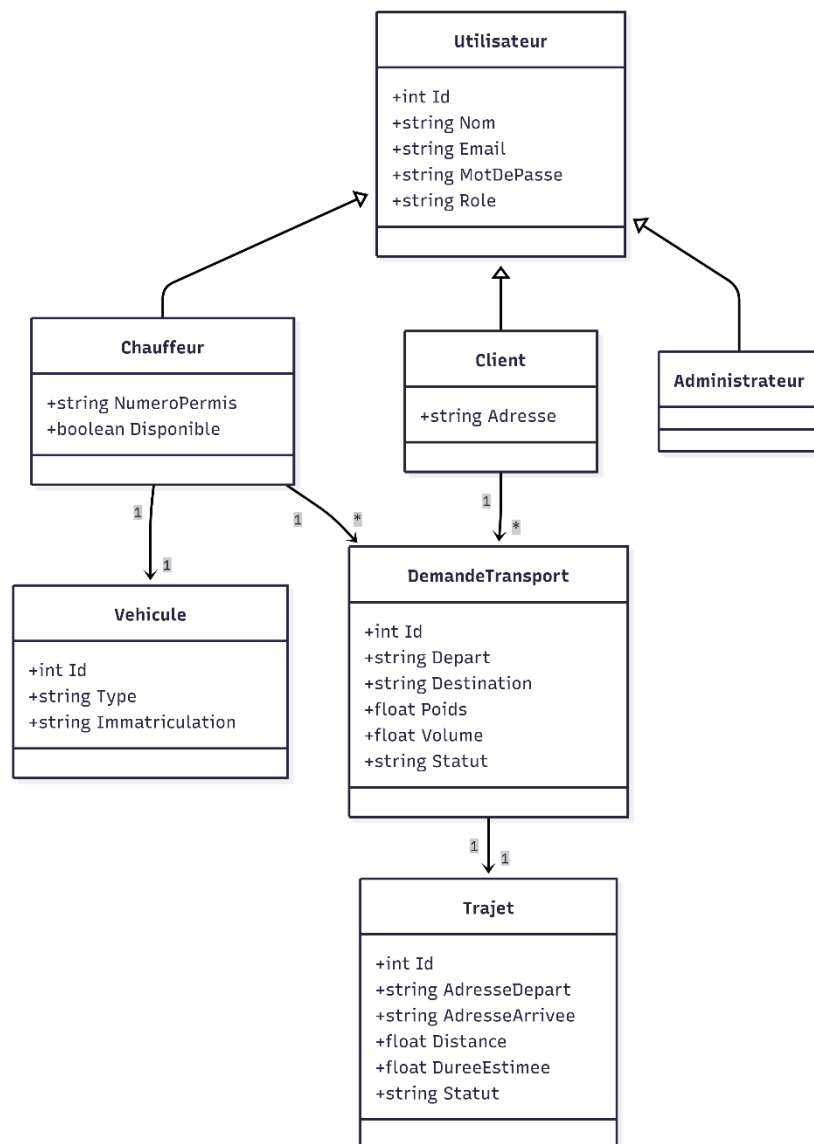


Figure 4: Diagramme de classes

## 7. Conclusion

Ce chapitre a été consacré à l'analyse et à la conception du projet AlloHonda, étapes fondamentales dans le processus de développement de l'application. À travers l'identification des acteurs et l'utilisation du langage de modélisation UML, il a été possible de formaliser les besoins fonctionnels et de structurer le système de manière claire et cohérente.

Les différents diagrammes UML présentés, notamment les diagrammes de cas d'utilisation, de séquence et de classes, ont permis de modéliser aussi bien les interactions entre les utilisateurs et le système que le comportement dynamique et la structure statique de l'application. Ces modèles offrent une vision globale du fonctionnement de la plateforme et facilitent la compréhension des rôles, des responsabilités et des échanges entre les différents composants.

Cette phase de conception constitue une base solide pour la mise en œuvre technique du projet. Elle permet de réduire les risques d'erreurs lors du développement, d'assurer la cohérence entre les exigences exprimées et la solution implémentée, et de garantir une meilleure maintenabilité du système. Le chapitre suivant sera consacré à la présentation de l'architecture générale et des choix technologiques retenus pour la réalisation de l'application AlloHonda.

# Chapitre 4 :

# Réalisation

# Chapitre 4. Réalisation

## 1. Introduction

Ce chapitre est consacré à la phase de réalisation du projet AlloHonda, qui correspond à la mise en œuvre concrète des éléments définis lors des phases d'analyse et de conception. Il présente l'architecture adoptée, les outils et technologies utilisés ainsi que le modèle de développement retenu pour assurer une application robuste, évolutive et maintenable.

L'objectif de cette phase est de traduire les modèles UML et les exigences fonctionnelles en une application mobile fonctionnelle, intégrant un backend performant et des services externes adaptés aux besoins du système.

## 2. Architecture

L'application AlloHonda repose sur une architecture client-serveur, permettant une séparation claire entre la couche présentation et la couche métier. Cette architecture favorise la modularité, la sécurité et la scalabilité du système.

- Frontend (Client mobile) : développé en React Native, il assure l'interface utilisateur et la gestion des interactions avec les utilisateurs (clients, chauffeurs).
- Backend (Serveur applicatif) : développé en .NET selon le modèle MVC, il gère la logique métier, les règles de gestion, la sécurité et l'accès aux données.
- Base de données : une base SQL Server est utilisée pour stocker les informations relatives aux utilisateurs, aux véhicules, aux demandes de transport et aux missions.
- Services externes :
  - Google Maps API pour la géolocalisation, l'affichage des cartes et le calcul des itinéraires.
  - Firebase pour la gestion de l'authentification admin.

Cette architecture permet une communication fluide entre les différentes composantes du système via des API REST sécurisées.

## Architecture AlloHonda

Modularité, Sécurité, Scalabilité

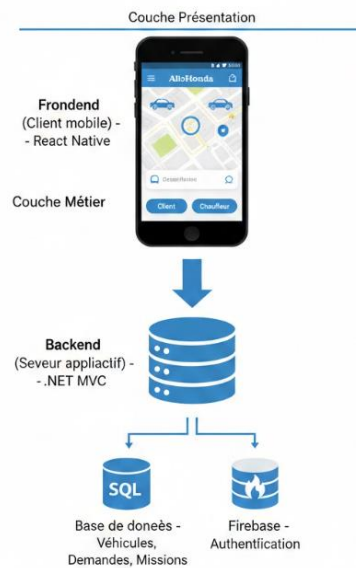


Figure 5: Architecture du projet

## 3. Outils et technologies

### 3.1 Environnement logiciel

#### ➤ Visual Studio Code (IDE) :



Figure 6: Logo de Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source gratuit et open source, développé par Microsoft. Il est conçu pour être léger, polyvalent et hautement personnalisable, tout en offrant une expérience de développement productive. VS Code prend en charge de nombreux langages de programmation populaires tels que JavaScript, Python, C++, Java, et bien d'autres, grâce à une large gamme d'extensions développées par la communauté.

## ➤ Visual Studio 2022 :



Figure 7: Logo de Visual Studio

Visual Studio 2022 est un environnement de développement intégré (IDE) complet et puissant proposé par Microsoft. Il permet aux développeurs de concevoir, développer, tester et déployer des applications de manière efficace. Visual Studio 2022 prend en charge de nombreux langages et technologies tels que C#, .NET, C++... et offre des outils avancés pour le débogage, la gestion de versions, le développement d'interfaces graphiques et l'intégration avec des services cloud.

## ➤ Visual Studio 2022 :



Figure 8 : Logo de Taiga

Taiga est un outil de gestion de projet Agile en ligne, utilisant les méthodes Kanban et Scrum. Le Kanban Taiga permet de visualiser le flux de travail sous forme de colonnes (Backlog, À faire, En cours, En revue, Terminé), facilitant le suivi des tâches, l'identification des blocages et l'amélioration continue du processus de développement. Il favorise la collaboration entre les membres de l'équipe grâce à une mise à jour en temps réel, une gestion claire des priorités et une traçabilité complète de l'avancement du projet.

## ➤ Modelio (Open Source) :



Figure 9 : Logo de Modelio

Modelio est un outil de modélisation open source conçu pour supporter les standards UML (Unified Modeling Language) et BPMN (Business Process Model and Notation). Développé par Modeliosoft, il permet de modéliser, documenter et générer du code à partir de diagrammes représentant des systèmes logiciels, des processus métiers ou encore des bases de données.



## 3.2 Technologies utilisées

### ➤ React Native:



Figure 10 : Logo de React Native

Framework JavaScript permettant de développer une application mobile multiplateforme à partir d'un seul code source, avec des performances proches du natif, notamment pour Android.

### ➤ .Net :



Figure 11 : Logo de .Net

Plateforme de développement de Microsoft utilisée pour la création d'API backend robustes, la gestion de la logique métier, la sécurité et la communication avec la base de données.

### ➤ SQLServer :



Figure 12 : Logo de SQL Server

Système de gestion de base de données relationnelle assurant le stockage structuré, la cohérence et la sécurité des données de l'application.

### ➤ Firebase :



Figure 13 : Logo de Firebase

Plateforme cloud fournissant des services tels que les notifications push, la communication en temps réel et l'authentification, facilitant l'interaction instantanée entre clients et chauffeurs.

### ➤ Google Maps API :



Figure 14 : Logo de Google Maps

Service de cartographie permettant la géolocalisation, l'affichage des cartes, le calcul d'itinéraires et le suivi des trajets en temps réel.

### ➤ Github :



Figure 15 : Logo de Github

Système de contrôle de version distribué permettant le suivi des modifications du code, la collaboration entre développeurs et la gestion des versions du projet.

### ➤ Postman :



Figure 16 : Logo de Postman

Postman est un outil de test d'API qui permet d'envoyer des requêtes HTTP (GET, POST, PUT, DELETE, etc.) vers un serveur et d'en visualiser les réponses. Il est très utilisé pour tester, déboguer et documenter des APIs, notamment dans les projets web ou mobiles.

## 3.3 Le modèle MVC :

Le backend de l'application AlloHonda est basé sur le modèle MVC (Model–View–Controller), qui permet une organisation claire et structurée du code.

- **Model** : représente les entités métier et les données du système (Utilisateur, Client, Chauffeur, DemandeTransport, Véhicule). Il assure l'interaction avec la base de données SQL Server.
- **View** : dans le cadre d'une API, la vue correspond principalement aux données retournées sous forme de réponses JSON destinées à l'application mobile.
- **Controller** : gère les requêtes provenant du frontend, traite la logique métier et appelle les modèles appropriés avant de retourner une réponse.

L'adoption du modèle MVC permet une meilleure séparation des responsabilités, facilite la maintenance du code et rend le système plus évolutif.

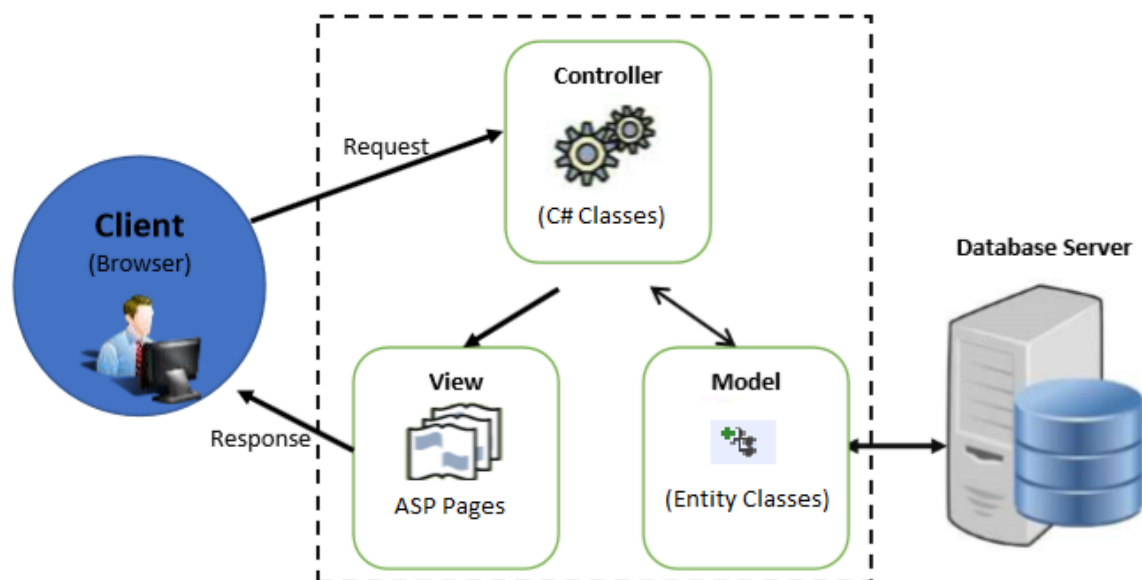


Figure 17 : Architecture du modèle MVC

## 4. Interfaces de l'application

### 4.1 Interface Authentification :

Le système d'authentification de l'application AlloHonda repose sur une approche hybride afin de répondre aux exigences de sécurité et de gestion des rôles. L'authentification des clients et des chauffeurs est assurée par ASP.NET Identity, qui

permet une gestion sécurisée des comptes utilisateurs, incluant l'inscription, la connexion, le stockage des mots de passe sous forme chiffrée (hash), ainsi que la gestion des rôles et des autorisations. Ce mécanisme garantit un contrôle d'accès fiable aux fonctionnalités de l'application en fonction du profil de l'utilisateur. En parallèle, une interface d'authentification dédiée à l'administrateur est mise en place en utilisant Firebase Authentication. Cette solution permet une authentification rapide et sécurisée de l'administrateur, tout en facilitant l'intégration avec les services Firebase utilisés par l'application, notamment pour la gestion des notifications push. Cette séparation des mécanismes d'authentification permet de renforcer la sécurité globale du système, d'isoler les responsabilités administratives et d'assurer une gestion efficace des accès selon les rôles.

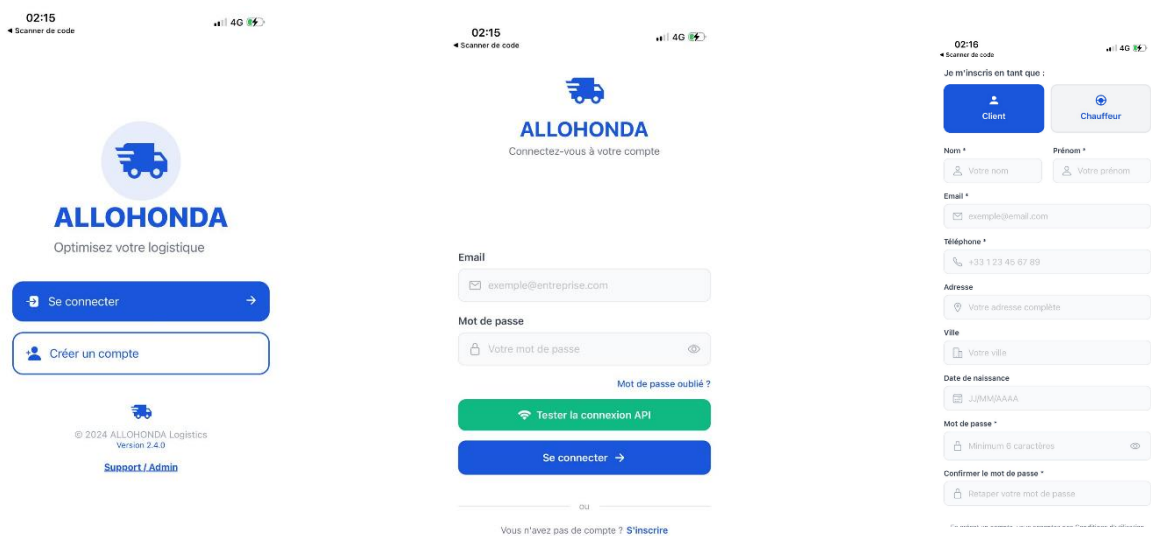


Figure 18 : Pages d'authentification

## 4.2 Interface Client :

L'interface Client de l'application AlloHonda a été conçue pour offrir une expérience utilisateur simple, intuitive et complète. Elle permet au client de gérer son profil personnel, comprenant la consultation et la modification des informations personnelles ainsi que les paramètres de l'application. Le client dispose également d'un historique des transports, qui regroupe l'ensemble des demandes effectuées, avec leurs statuts (en attente, acceptée, annulée ou livrée) et les détails associés tels que les lieux, les dates et le chauffeur assigné. Un tableau de bord (dashboard) est mis à disposition afin de présenter des statistiques globales, notamment le nombre total de transports réalisés, le taux de réussite et les livraisons effectuées, offrant ainsi une vue synthétique de l'activité du client. Par ailleurs, l'interface client intègre une fonctionnalité d'allocation de

transport, permettant la création d'une nouvelle demande à travers un formulaire détaillé incluant les informations de départ, de destination, le type de marchandise, le poids, le volume et des commentaires supplémentaires. L'ensemble de ces fonctionnalités vise à faciliter la gestion des demandes de transport et à assurer un suivi efficace et transparent des opérations pour le client.

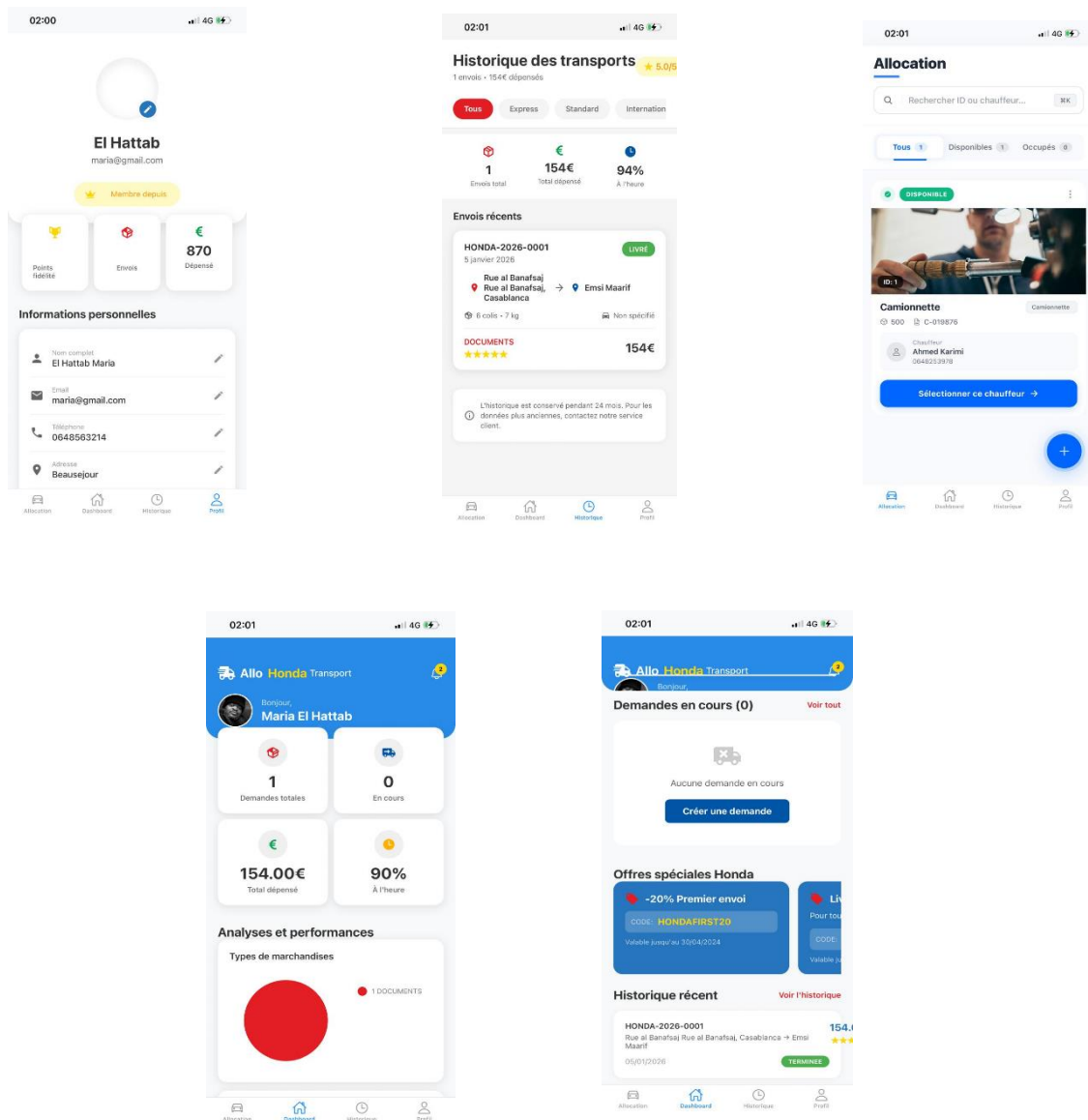


Figure 19 : Interfaces clients

### 4.3 Interface Chauffeur :

L'interface Chauffeur de l'application AlloHonda a été conçue pour permettre une gestion efficace et autonome des missions de transport. Après une authentification sécurisée, le chauffeur accède à un tableau de bord regroupant des statistiques clés, telles que le nombre total de missions effectuées, les missions du jour, la distance parcourue et les gains cumulés. L'interface permet également la gestion des demandes de transport reçues, avec la possibilité de consulter les détails de chaque demande (client, lieux de départ et d'arrivée, type et volume de marchandise) et d'accepter ou refuser les missions en fonction de la disponibilité du chauffeur. Une fois la mission acceptée, le chauffeur peut suivre l'itinéraire en temps réel grâce à l'intégration de la cartographie, mettre à jour le statut du trajet (en cours, terminé, livré) et recevoir des notifications instantanées liées à l'évolution des missions. L'interface inclut également un historique des missions, permettant au chauffeur de consulter les transports réalisés ainsi que les gains associés, en plus d'un espace profil dédié à la gestion des informations personnelles, du véhicule et des documents. L'ensemble de ces fonctionnalités vise à optimiser l'organisation du travail du chauffeur et à assurer un suivi précis et transparent des activités.

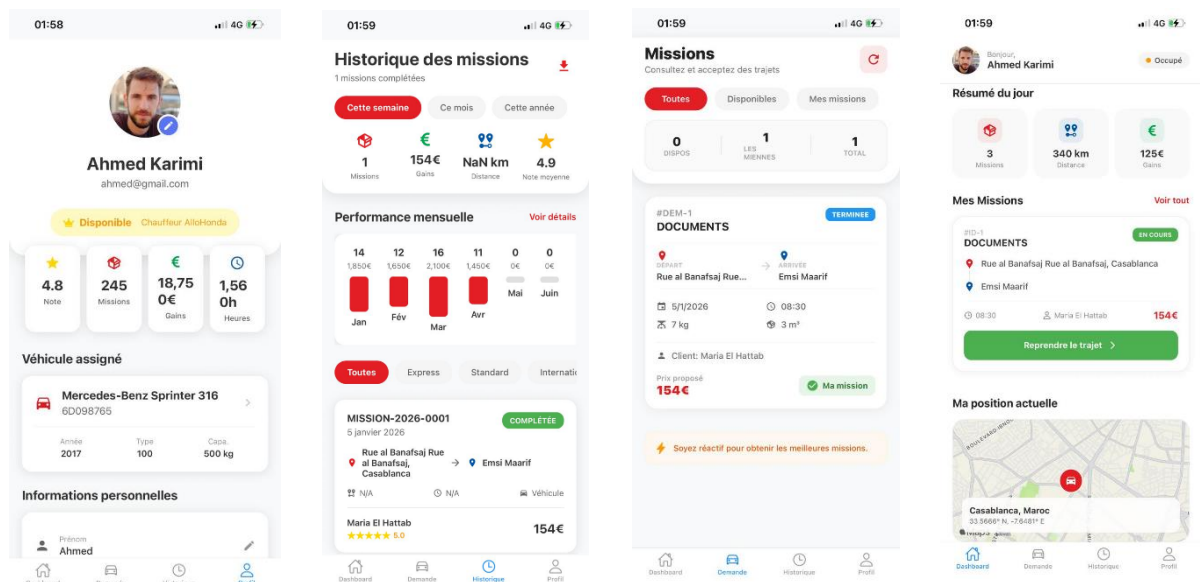


Figure 20 : Interfaces chauffeurs

## 4. Interface Admin :

L'interface Administrateur de l'application AlloHonda est conçue pour assurer la supervision globale et la gestion centralisée de la plateforme. Après une authentification sécurisée, l'administrateur accède à un tableau de bord offrant une vue d'ensemble sur l'activité du système, incluant des statistiques générales telles que le nombre total de demandes de transport, les livraisons effectuées, le nombre de clients actifs et de chauffeurs actifs. L'interface permet également la gestion des utilisateurs, avec des fonctionnalités de création, de modification et de suppression des comptes clients et chauffeurs, ainsi que la validation des comptes chauffeurs afin de garantir la conformité des informations et des documents fournis. L'administrateur dispose aussi d'outils de suivi et de contrôle des demandes de transport, lui permettant de consulter l'ensemble des missions, leurs statuts et leur évolution en temps réel. Enfin, l'interface intègre des fonctionnalités de statistiques et de reporting, facilitant l'analyse de l'activité de la plateforme et l'aide à la prise de décision. Cette interface joue un rôle clé dans le bon fonctionnement du système en assurant la sécurité, la fiabilité et l'efficacité de l'application AlloHonda.

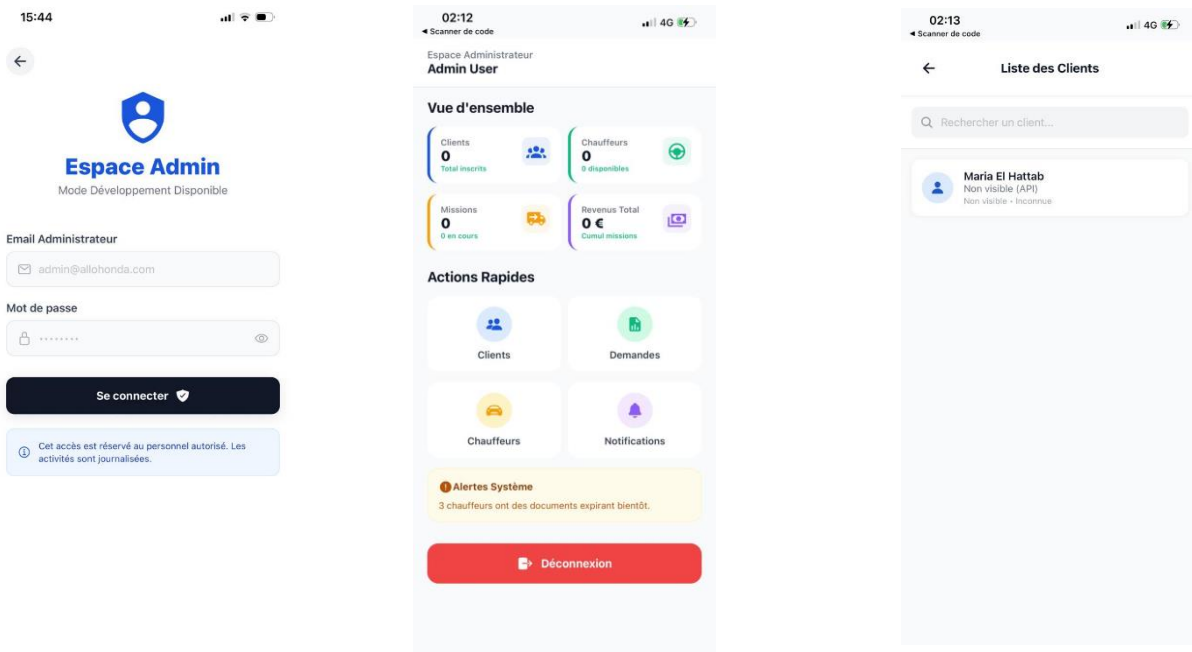


Figure 21 : Interfaces administrateur

## 5. Conclusion

Ce chapitre a permis de présenter la phase de réalisation du projet AlloHonda, en mettant en évidence l'architecture adoptée, les outils et technologies utilisés ainsi que les choix techniques effectués. L'architecture client-serveur mise en place assure une séparation claire entre la couche présentation, la couche métier et la couche données, favorisant ainsi la modularité, la sécurité et la scalabilité du système.

L'utilisation de technologies modernes telles que React Native pour le développement mobile, .NET basé sur le modèle MVC pour le backend, SQL Server pour la gestion des données, ainsi que l'intégration de Firebase et de Google Maps API, a permis de développer une application performante et adaptée aux exigences fonctionnelles définies. Par ailleurs, les environnements de développement Visual Studio 2022 et Visual Studio Code ont facilité l'implémentation, le débogage et le travail collaboratif.

Les différentes interfaces (Client, Chauffeur et Administrateur) ont été réalisées de manière à répondre aux besoins spécifiques de chaque acteur, tout en garantissant une expérience utilisateur fluide et sécurisée. Ce chapitre constitue ainsi une étape clé reliant la conception théorique à l'implémentation concrète, et prépare la transition vers le chapitre suivant consacré aux tests, à la validation et à l'évaluation des résultats de l'application AlloHonda.



## Conclusion générale et perspectives

Au terme de ce travail, le projet AlloHonda a permis de concevoir et de réaliser une application mobile complète dédiée au transport de biens, répondant aux besoins des clients, des chauffeurs et des administrateurs à travers une plateforme unique et centralisée. L'objectif principal de ce projet était de faciliter la gestion des demandes de transport, d'optimiser l'affectation des missions et d'assurer un suivi en temps réel des trajets, tout en garantissant une expérience utilisateur fluide et sécurisée.

L'analyse des besoins, la modélisation à l'aide des diagrammes UML et la mise en œuvre d'une architecture client-serveur basée sur React Native et .NET (MVC) ont permis de développer une solution robuste, évolutive et conforme aux exigences fonctionnelles et non fonctionnelles définies. L'intégration de SQL Server pour la gestion des données, de Google Maps API pour la géolocalisation et de Firebase pour l'authentification et les notifications push a renforcé la performance et l'interactivité de l'application.

Sur le plan académique, ce projet a constitué une expérience enrichissante, permettant de mettre en pratique les connaissances acquises en développement mobile, en conception logicielle, en gestion de bases de données et en travail collaboratif. L'utilisation d'outils de gestion de projet Agile et de contrôle de version a favorisé une organisation efficace et une implication continue de l'ensemble des membres de l'équipe.

En perspective, plusieurs axes d'amélioration peuvent être envisagés afin d'enrichir davantage l'application AlloHonda. Parmi ces évolutions figurent l'intégration d'un système de paiement en ligne, la mise en place d'un chat en temps réel entre le client et le chauffeur, le support du multilinguisme, ainsi que l'optimisation des algorithmes d'affectation des missions. Ces perspectives ouvrent la voie à une extension fonctionnelle et technique du projet, renforçant ainsi son potentiel d'adoption et d'évolution vers une solution à plus grande échelle.

# Webographie

## 1. React Native – Documentation officielle

<https://reactnative.dev/>

Documentation officielle du framework React Native, utilisée pour le développement de l'application mobile multiplateforme.

## 2. Microsoft .NET – Documentation officielle

<https://learn.microsoft.com/en-us/dotnet/>

Ressource officielle pour le développement backend en .NET, incluant les API, l'architecture MVC et les bonnes pratiques.

## 3. ASP.NET Core MVC – Guide

<https://learn.microsoft.com/en-us/aspnet/core/mvc/>

Documentation détaillée sur le modèle MVC utilisé pour structurer la logique métier du backend.

## 4. Firebase – Documentation officielle

<https://firebase.google.com/docs>

Documentation utilisée pour l'authentification et l'envoi des notifications push en temps réel.

## 5. Firebase Authentication

<https://firebase.google.com/docs/auth>

Guide détaillé sur la gestion sécurisée de l'authentification des utilisateurs via Firebase.

## 6. Google Maps Platform – Documentation

<https://developers.google.com/maps/documentation>

Ressource officielle pour l'intégration de la géolocalisation, des cartes et du calcul d'itinéraires.

## 7. Taiga – Agile Project Management Tool

<https://www.taiga.io/>

Plateforme de gestion de projet Agile utilisée pour le suivi des tâches via la méthode Kanban.

## 8. GitHub – Documentation

<https://docs.github.com/>

Documentation officielle de GitHub pour le versionnement du code et le travail collaboratif.