

# RAPPORT PROJET

## PERCEPTIVE AUTOMOTIVE

Code source et documentation :

<https://github.com/yh175/Perceptive-Automotive>



Introduction.....	3
Contexte .....	5
Description du projet.....	6
Plateforme SaaS “builttobealive.com” .....	6
Système Embarqué Simulé via CARLA .....	7
Conception et Développement .....	8
Cas d’Utilisation (Use Cases) .....	8
Diagrammes de Classes .....	9
Diagrammes de Séquences .....	9
Architecture.....	11
Architecture de la Plateforme SaaS .....	11
Architecture du Système Embarqué des Véhicules Autonomes .....	12
Planification et gestion de projet.....	14
Stratégies de Planification.....	14
Méthode de Gestion de Projet Utilisée .....	15
Outils de Collaboration .....	15
Résultats et Évaluation.....	16
Résultats et Évaluation.....	16
Fonctionnalité de Réservation.....	16
Problèmes liés au Contrôle du Véhicule .....	16
Conclusion .....	18

# Introduction

Dans le cadre de notre formation, nous avons eu l'opportunité de travailler sur un projet ambitieux : développer une application logicielle permettant le fonctionnement d'une voiture autonome. Ce projet s'inscrit dans la volonté d'approfondir nos connaissances et de renforcer nos compétences techniques et organisationnelles au cours de nos deux années d'études. En plus de répondre aux besoins croissants de mobilité et de confort des utilisateurs, ce projet nous offre la possibilité de démontrer notre capacité à intégrer diverses disciplines telles que l'intelligence artificielle, l'infrastructure réseau, et le développement logiciel.

Les voitures autonomes représentent l'avenir de la mobilité, avec des avantages significatifs tels que la réduction des accidents, l'optimisation du trafic, et l'amélioration de l'expérience utilisateur. En tant que développeurs, nous avons entrepris de concevoir un système qui non seulement gère la réservation de véhicules autonomes mais aussi assure leur fonctionnement en toute sécurité et efficacité.

Ce projet constitue une étape clé dans notre apprentissage, car il nous permet de mettre en pratique les connaissances théoriques acquises tout en développant des compétences en gestion de projet, de la conception à la réalisation. Il démontre également notre aptitude à aborder des défis complexes, à coordonner des équipes multidisciplinaires, et à livrer des solutions innovantes répondant aux exigences du monde réel.

L'objectif de ce document est de présenter en détail les résultats de nos travaux ainsi que le processus mis en place pour réaliser ce projet. Nous y aborderons les différentes étapes de développement, de la phase de conception initiale à la mise en œuvre finale, en passant par l'analyse des besoins, l'architecture de la solution, et les tests de validation.

Ce rapport est structuré comme suit :

- **Section 1 : Contexte** - Présente le contexte du projet et les points clés du cahier des charges.
- **Section 2 : Description du Projet** - Décrit en détail le projet, ses objectifs, et la vision globale de la solution.
- **Section 3 : Conception et Développement** - Explique les différentes étapes de la conception logicielle, y compris les cas d'utilisation, les diagrammes de classes et de séquences.
- **Section 4 : Architecture de la Solution** - Détaille l'architecture technique, à la fois pour la plateforme SaaS et l'infrastructure embarquée du véhicule autonome.
- **Section 5 : Planification et Gestion de Projet** - Met en lumière les stratégies de planification, les diagrammes de Gantt, et les outils de collaboration utilisés pour gérer efficacement le projet.
- **Section 6 : Résultats et Évaluation** - Présente les résultats des tests et des validations, ainsi qu'un retour d'expérience sur le projet.
- **Section 7 : Conclusion** - Résume les principaux enseignements du projet et propose des pistes d'amélioration pour l'avenir.

Ce rapport offre une vue d'ensemble détaillée du projet "Perceptive Automotive" et illustre notre capacité à combiner des connaissances théoriques et des compétences pratiques pour développer une solution innovante dans le domaine de la mobilité autonome. À travers ce document, nous espérons non seulement démontrer la viabilité de notre approche technique et organisationnelle, mais aussi partager les défis rencontrés, les solutions mises en œuvre, et les apprentissages tirés de cette expérience.

En s'appuyant sur des technologies de pointe telles que l'intelligence artificielle et des systèmes de contrôle embarqués sophistiqués comme Apollo, notre solution vise à offrir une expérience utilisateur optimale tout en garantissant sécurité, fiabilité, et efficacité. Nous avons également utilisé des outils modernes de simulation comme CARLA pour tester et valider nos algorithmes de conduite autonome dans un environnement virtuel réaliste, permettant une évaluation rigoureuse et précise de la performance du système.

Ce projet nous a permis de développer des compétences en gestion de projet, en intégration de systèmes, en conception de logiciels et en développement collaboratif, tout en répondant à une problématique complexe qui façonnera l'avenir de la mobilité. La rédaction de ce rapport marque l'aboutissement de plusieurs mois de travail intense et constitue une vitrine de notre capacité à mener un projet d'envergure de manière structurée et méthodique.

Nous espérons que ce rapport apportera une compréhension claire et complète de notre travail, et qu'il pourra servir de référence pour de futurs projets similaires dans le domaine des véhicules autonomes.

## Contexte

Dans un monde de plus en plus automatisé, les constructeurs automobiles n'envisagent pas un monde sans voiture autonome. L'émergence de l'intelligence artificielle, les améliorations en termes de performances mais également techniques n'ont fait que faciliter la concrétisation de cette idée qui ne date pas d'aujourd'hui mais qui a émergé depuis le XXe siècle. La question d'un monde avec des voitures autonomes ne se pose plus mais la question est quand verrons-nous des voitures autonomes en circulation sur nos routes. La circulation de voitures autonomes est un défi complexe et difficile à mettre en place mais ses avantages sont indéniables tels que la réduction des accidents, l'optimisation de la circulation et bien d'autres.

Dans le cadre de notre projet les voitures autonomes seront utilisées comme des voitures de services pour le transport de passager. L'objectif étant de :

« Mettre en place une flotte de taxis autonomes (niveau 4 ou 5). Le système permet à un utilisateur de réserver un véhicule pour effectuer un trajet prévu. Le système doit prendre en compte l'itinéraire pour lui envoyer le véhicule le plus apte à répondre au besoin. Le véhicule autonome doit identifier le conducteur pour vérifier que son identité est conforme à ce qui est renseigné sur son compte et s'assurer du nombre de passagers à embarquer. Le véhicule doit être capable de conduire de manière confortable et sécurisée pour l'ensemble de ses occupants. Le véhicule doit pouvoir prendre en charge les demandes en temps réel des occupants et s'assurer que ceux-ci soient en bonne santé le long du trajet. Le véhicule doit contrôler en permanence son intégrité et céder la conduite à l'utilisateur en cas de défaillance majeure et contacter l'assistance adaptée. En fin de service, le véhicule et le système doivent analyser si le véhicule est apte pour une prochaine réservation ou s'il doit charger avant. » [1].

Un scénario illustrant l'utilisation de notre application est décrit dans le paragraphe ci-dessous :

“ M Smithschmidtlaforge se connecte au site [builttobealive.com](http://builttobealive.com) et fait une réservation pour un véhicule qui doit se trouver en bas du 34 avenue Foch, à Paris 16. Dans sa demande, il ne précise pas où il ira, mais indique bien qu'il veut pouvoir prendre la route à 10h30, le 23 janvier 2023.

Le 23 janvier 2023 à 10h28, Mr Smithschmidtlaforge descend en bas de son immeuble, et n'attend vraiment pas longtemps pour voir arriver son transport, de couleur jaune et orange. Le véhicule se gare, et ne bouge plus.

L'utilisateur se présente près de la porte d'accès au véhicule, et montre sa pièce d'identité à côté de son visage. Dans la seconde, la portière s'ouvre et laisse le client s'installer.

Mr Smithschmidtlaforge repense à la journée qu'il doit remplir et finit par annoncer :

- Je dois passer à ma banque pour retirer 200 crédits et aller retirer la commande pour le cadeau que j'ai prévu de donner ce soir, lors de notre dîner à 20h au restaurant du « Roti bien cuit » à Asnières, à Dzung. Après le restaurant nous irons nous promener sur les quais de Seine et nous rentrerons ensuite à la maison. Mais avant cela, je dois bien sûr passer ma journée au travail.

Le véhicule démarre, silencieusement, puisqu'il est électrique, aussi silencieusement que son passager qui ne dit plus un mot, et le système de bord fait jouer le Dies irae du Requiem de

Mozart, que Mr Smithschmidtlaforge adore entendre pour être bien réveillé en début de matinée. “

## Description du projet

Le projet "Perceptive Automotive" vise à concevoir et développer une solution logicielle complète qui permet le fonctionnement sécurisé et autonome d'un véhicule sans conducteur. Ce projet couvre à la fois la gestion des déplacements autonomes, la prise en charge des passagers, et la réservation de véhicules par les utilisateurs via une plateforme dédiée. L'objectif est de créer une solution robuste et évolutive qui intègre divers éléments de l'intelligence artificielle, de l'infrastructure réseau, et du développement logiciel.

La solution se divise en deux composants principaux :

### Plateforme SaaS “[builttobealive.com](https://builttobealive.com)”

La première composante est une plateforme SaaS (Software as a Service), "builttobealive.com", qui sert de point d'entrée pour les utilisateurs souhaitant interagir avec le système de véhicules autonomes. Cette plateforme a plusieurs fonctionnalités clés :

- **Authentification et gestion des comptes utilisateurs** : Les utilisateurs doivent créer un compte en fournissant des informations de base telles que le nom d'utilisateur, l'email, et un mot de passe sécurisé. Une fois enregistrés, les utilisateurs peuvent se connecter pour accéder aux services de la plateforme.
- **Réservation de véhicules autonomes** : Une fois authentifié, l'utilisateur peut louer un véhicule autonome pour un trajet particulier. La plateforme propose une interface intuitive permettant de spécifier la date, l'heure, le lieu de départ, et la destination souhaitée. Le système vérifie ensuite les disponibilités et choisit le véhicule optimal en fonction de la proximité, de la disponibilité, et d'autres critères pertinents.
- **Gestion des propriétaires de véhicules** : Si l'utilisateur est propriétaire d'un véhicule autonome, celui-ci peut gérer les accès à son propre véhicule. Le propriétaire peut restreindre ou autoriser l'accès à d'autres utilisateurs, définir des règles de conduite, et surveiller l'état du véhicule à distance.
- **Optimisation de l'allocation des véhicules** : Lorsqu'une réservation est effectuée, le système utilise des algorithmes d'optimisation pour déterminer quel véhicule envoyer à l'utilisateur. Cette optimisation est basée sur des critères tels que la distance à parcourir du point de stationnement jusqu'à l'adresse de l'utilisateur, l'état de la batterie, la disponibilité des véhicules, et les préférences de l'utilisateur.
- **Plateforme multiplateforme** : La solution est conçue pour être cross-platform, c'est-à-dire compatible avec plusieurs environnements (Web, iOS, Android) afin de permettre aux utilisateurs d'accéder aux services de manière fluide et continue, que ce soit via un smartphone, une tablette, ou un ordinateur.

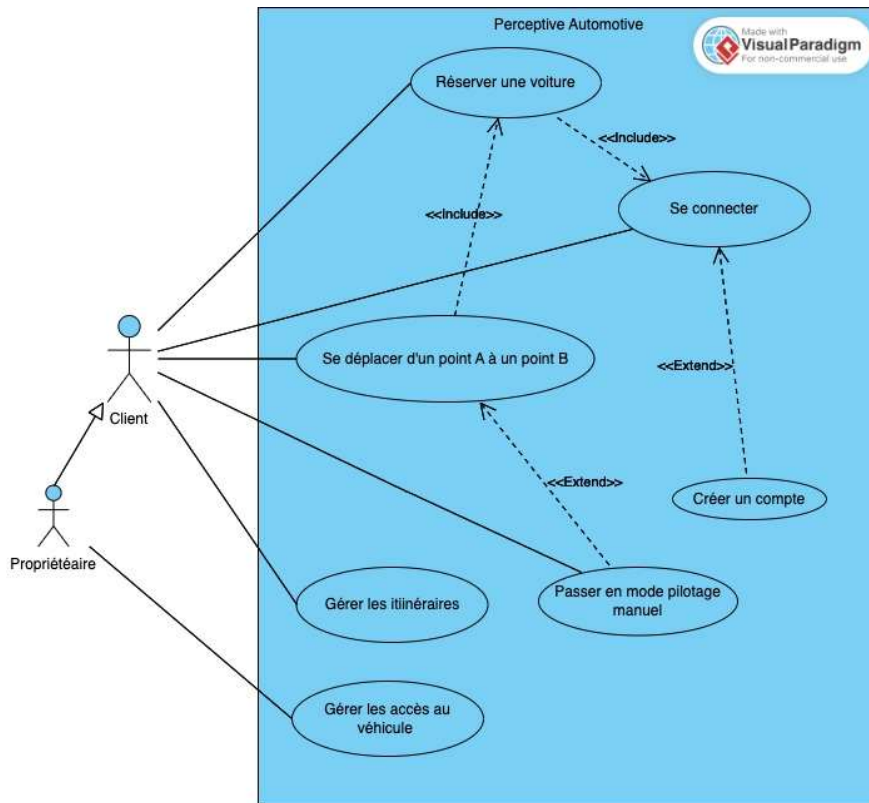
## Système Embarqué Simulé via CARLA

Le deuxième composant du projet est le système embarqué des véhicules autonomes, simulé via CARLA (Car Learning to Act). CARLA est une plateforme open-source de simulation utilisée pour la recherche et le développement de technologies de conduite autonome. Cette plateforme fournit un environnement de simulation réaliste qui reproduit des scénarios de conduite variés pour tester et valider les algorithmes de contrôle et de décision des véhicules autonomes. Voici les fonctionnalités offertes par cette composante :

- **Environnement de simulation réaliste** : CARLA permet de simuler un environnement urbain avec des conditions météorologiques variables, des trafics denses, des piétons, des infrastructures routières complexes, et des obstacles imprévus. Cela offre un cadre idéal pour tester les réactions du système de conduite autonome dans des conditions quasi-réelles.
- **Développement et tests d'algorithmes de conduite** : La plateforme permet le développement et le test de divers algorithmes de conduite autonome, y compris ceux liés à la perception de l'environnement (Lidar, Radar, Caméra), à la prise de décision (intelligence artificielle et apprentissage automatique), et au contrôle du véhicule (algorithmes PID pour la régulation de la vitesse et de la direction).
- **Gestion des scénarios de conduite** : Les développeurs peuvent créer différents scénarios pour simuler des situations d'urgence, des pannes techniques, des interventions d'urgence, et des situations de trafic complexes. Cela permet de valider la robustesse et la fiabilité du système embarqué.
- **Intégration avec d'autres outils et technologies** : CARLA est compatible avec divers outils de développement, frameworks d'apprentissage automatique, et logiciels de simulation comme Apollo et Bridge, permettant une intégration facile et un environnement de développement flexible.
- **Évaluation de la sécurité et de la fiabilité** : Grâce à la simulation, le système peut être testé pour s'assurer qu'il respecte les normes de sécurité en vigueur, telles que la gestion de la détection des objets, la prévention des collisions, et la capacité à céder le contrôle au conducteur en cas de défaillance majeure.

# Conception et Développement

## Cas d'Utilisation (Use Cases)



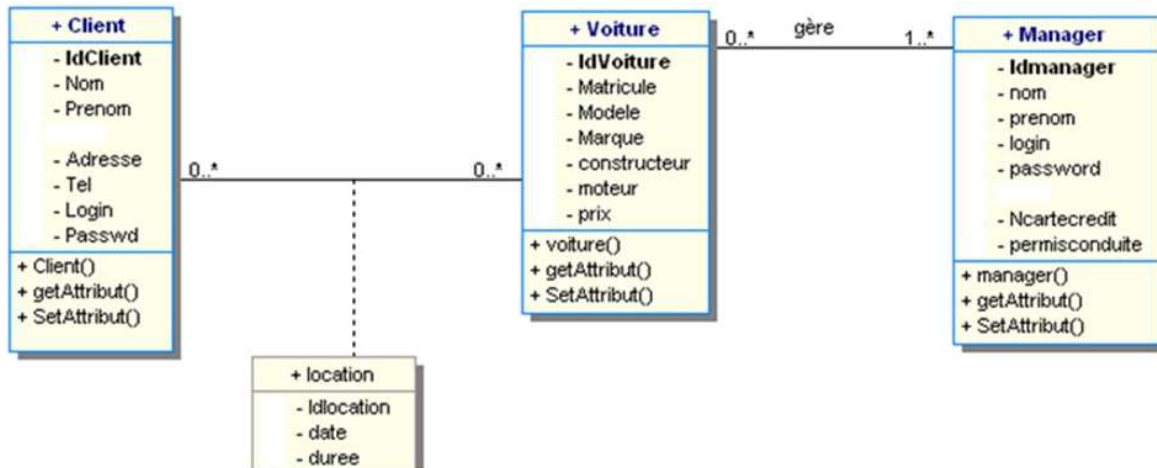
La conception des cas d'utilisation est cruciale pour identifier les interactions principales entre les utilisateurs et le système. Dans le cadre de notre projet de voiture autonome, nous avons défini plusieurs cas d'utilisation essentiels pour garantir une expérience utilisateur fluide et sécurisée. Voici les principaux cas d'utilisation identifiés :

1. **Réservation d'un véhicule autonome** : Un utilisateur se connecte sur la plateforme SaaS (builttobealive.com), s'authentifie et effectue une réservation pour un véhicule autonome. Le système vérifie les disponibilités, sélectionne le véhicule optimal en fonction de la distance, de l'état de la batterie, et d'autres critères, puis confirme la réservation à l'utilisateur.
2. **Identification et vérification des clients** : Lors de la prise en charge, le véhicule vérifie l'identité des passagers pour confirmer qu'ils correspondent aux informations fournies lors de la réservation. Cela garantit la sécurité des utilisateurs et la validité du service.
3. **Conduite autonome et prise de décisions** : Une fois les passagers à bord, le véhicule se met en route en utilisant des algorithmes d'intelligence artificielle et des systèmes de régulation (comme le contrôleur PID) pour assurer une conduite sécurisée et confortable.
4. **Gestion des scénarios d'urgence** : En cas de défaillance majeure ou de situation critique, le véhicule peut céder la conduite à l'utilisateur ou contacter l'assistance appropriée pour une intervention immédiate.



5. **Maintenance et gestion de la flotte** : Après chaque trajet, le système évalue l'état du véhicule pour déterminer s'il est prêt pour la prochaine réservation ou s'il nécessite une recharge ou une maintenance.

### Diagrammes de Classes



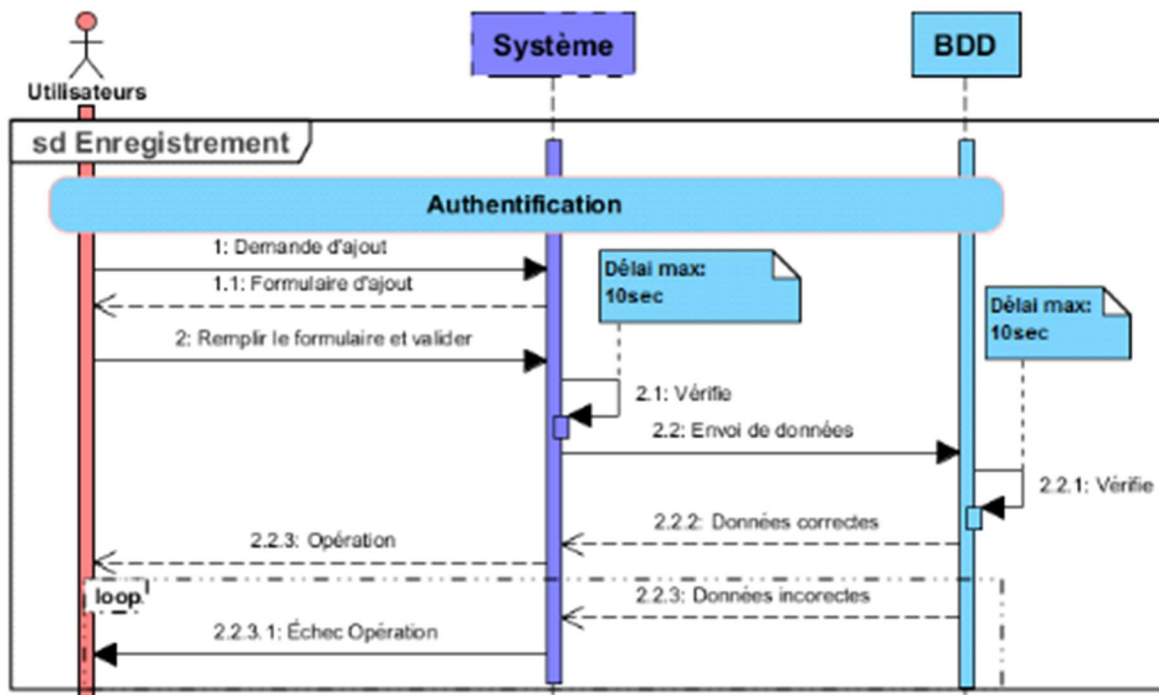
Les diagrammes de classes sont utilisés pour décrire la structure statique du système. Ils montrent les relations entre différentes classes de l'application et les attributs ou méthodes associés. Voici les principales classes définies pour le projet de voiture autonome :

1. **Classe Client** : Représente les utilisateurs de la plateforme avec des attributs tels que idUtilisateur, nom, prénom, email, et motDePasse. Les méthodes associées incluent authentifier(), reserverVehicule(), et consulterHistoriqueReservations().
2. **Classe Voiture** : Contient des informations sur le véhicule, y compris idVehicule, typeVehicule, niveauBatterie, et statut. Les méthodes associées sont verifierDisponibilite(), conduire(), et effectuerMaintenance().
3. **Classe Location** : Gère le processus de réservation avec des attributs comme idReservation, dateReservation, heureDépart, lieuDépart, et destination. Les méthodes incluent creerReservation(), annulerReservation(), et modifierReservation().
4. **Classe Système de Conduite Autonome (système embarqué)** : Englobe les composants du système de conduite autonome, avec des sous-classes pour les capteurs (Lidar, Radar, Caméra), le contrôleur PID, et les modules d'intelligence artificielle pour la prise de décision.

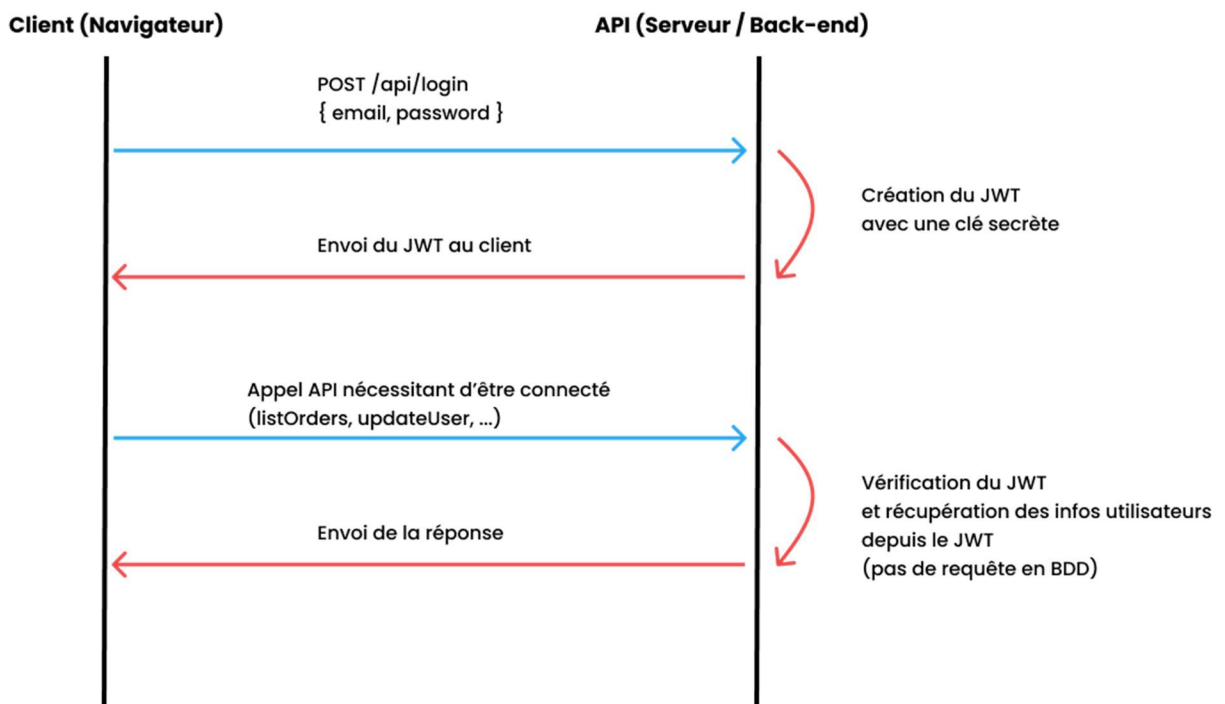
### Diagrammes de Séquences

Les diagrammes de séquences montrent comment les objets du système interagissent entre eux au fil du temps pour accomplir une tâche spécifique. Voici les deux principaux diagrammes de séquences développés :

1. **Séquence de réservation d'un véhicule** : Ce diagramme décrit les étapes du processus de réservation, depuis la connexion de l'utilisateur à la plateforme, l'envoi des détails de réservation, la vérification des disponibilités, jusqu'à la confirmation finale de la réservation.



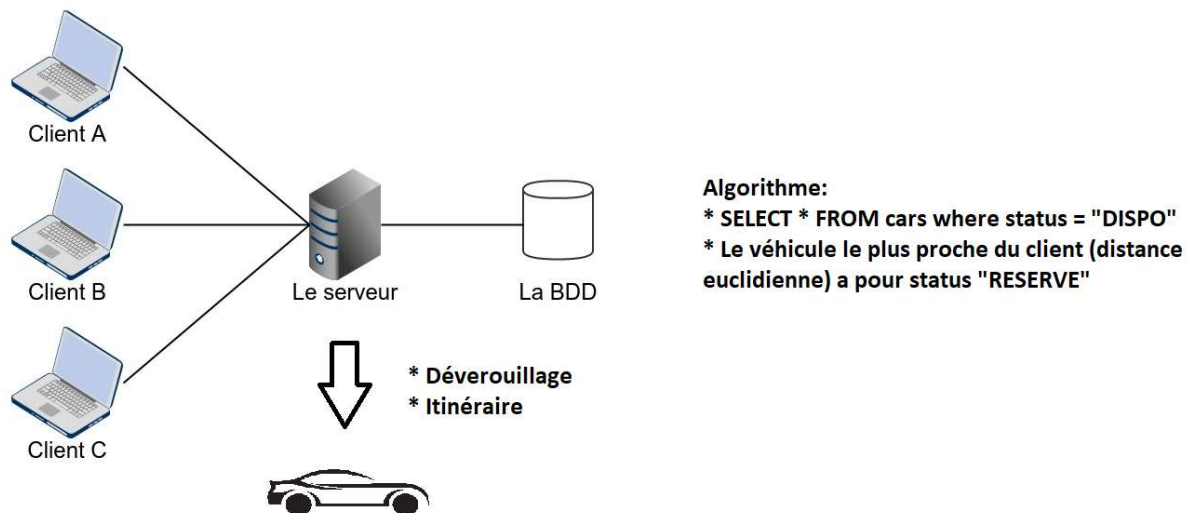
2. **Séquence d'authentification des clients**: Ce diagramme détaille le processus d'identification des passagers par le véhicule autonome. Une fois le passager à proximité du véhicule, le système de reconnaissance faciale vérifie l'identité du passager et, en cas de correspondance, déverrouille les portes pour permettre l'accès.



# Architecture

L'architecture est divisée en deux grandes parties : l'architecture de la **plateforme SaaS (Software as a Service)** et l'**architecture du système embarqué des véhicules autonomes**.

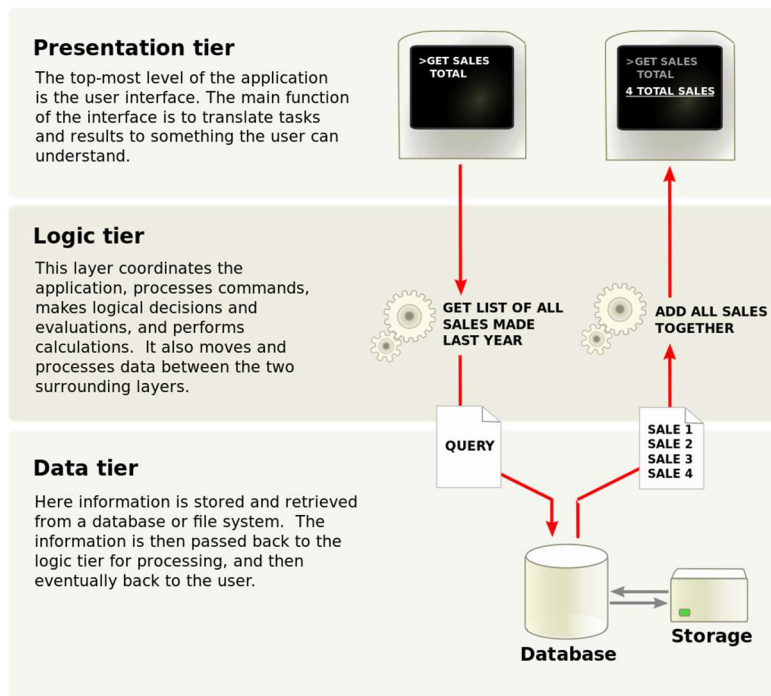
## Architecture de la Plateforme SaaS



L'architecture de la plateforme "builttoealive.com" est basée sur un modèle **client-serveur** à plusieurs niveaux, comprenant des couches de présentation, de logique métier et de données. Cette plateforme gère les interactions des utilisateurs, la gestion des comptes, la réservation des véhicules, et l'optimisation des trajets.

### Architecture à Trois Niveaux :

- **Couche de présentation** : La couche de présentation est l'interface utilisateur (UI) qui permet aux clients de se connecter, de créer des comptes, de gérer leurs informations personnelles, et de réserver des véhicules autonomes. Cette couche est accessible via des navigateurs web et des applications mobiles (iOS, Android). Elle est conçue pour être réactive et conviviale, offrant une expérience fluide sur tous les appareils.
- **Couche de logique métier** : Cette couche contient la logique applicative pour le traitement des réservations, la gestion des utilisateurs, et l'optimisation de l'allocation des véhicules. Elle comprend plusieurs services tels que le service d'authentification et le service de gestion des réservations. Chaque service est développé de manière indépendante, ce qui permet une scalabilité et une maintenance aisée.
- **Couche de données** : La couche de données gère le stockage et la récupération des informations des utilisateurs, des réservations, des véhicules, et des itinéraires. Elle est constituée de bases de données relationnelles.



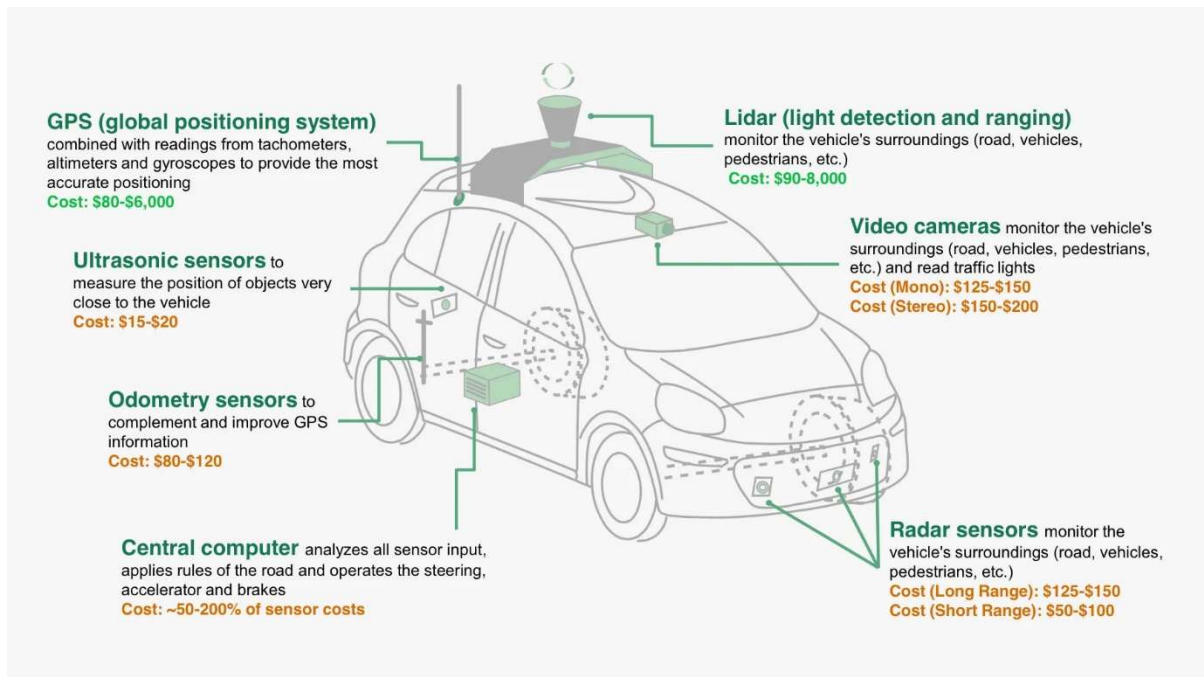
## Architecture du Système Embarqué des Véhicules Autonomes

L'architecture du système embarqué des véhicules autonomes repose sur une combinaison de matériels sophistiqués (capteurs, caméras, radars, lidars) et de logiciels avancés (algorithmes de conduite autonome, régulation PID, IA). Cette architecture est conçue pour permettre une conduite autonome sécurisée, fiable, et adaptative dans divers environnements.

- **Système de perception (capteurs) :**
  - **Lidar (Light Detection and Ranging) :** Utilisé pour détecter les objets à distance et mesurer leur position avec précision. Il permet de créer une carte 3D de l'environnement immédiat, essentielle pour la navigation.
  - **Caméras HD :** Utilisées pour la reconnaissance des objets, des panneaux de signalisation, des feux de circulation, et des piétons. Les caméras fournissent des données visuelles critiques pour la prise de décision en temps réel.
  - **Radar :** Utilisé pour détecter les objets en mouvement tels que les véhicules et les obstacles, surtout dans des conditions de faible visibilité.
  - **GPS et IMU (Inertial Measurement Unit) :** Fournissent des informations de localisation et d'orientation pour le positionnement global du véhicule.
- **Modules de Traitement et de Contrôle :**
  - **Unité Centrale de Traitement (CPU) et Unité de Traitement Graphique (GPU) :** Utilisées pour le traitement des données en temps réel provenant des capteurs et pour exécuter des algorithmes complexes d'apprentissage automatique et de vision par ordinateur.
  - **Régulation PID (Proportionnel, Intégral, Différentiel) :** Les algorithmes PID sont utilisés pour réguler la vitesse et la direction du véhicule. Ils jouent un rôle

crucial dans la stabilisation du véhicule et l'atténuation des oscillations pendant la conduite.

- **Intelligence artificielle** : Ces modules sont responsables de la prise de décision en temps réel, de la planification des itinéraires, de l'évitement des obstacles et de l'interaction avec les usagers de la route. Les modèles d'apprentissage profond sont formés sur des jeux de données simulés et réels pour améliorer la précision et la sécurité.



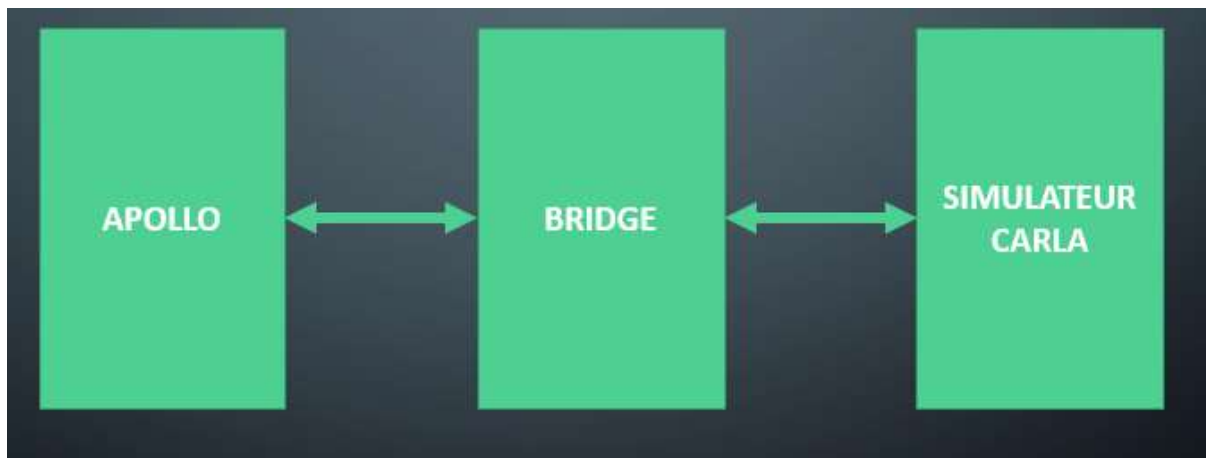
L'un des aspects les plus critiques de l'architecture embarquée est l'intégration efficace entre CARLA, et **Apollo**, un système de conduite autonome modulaire et extensible. Pour réaliser cette intégration, un **bridge** a été implémenté par une doctorante Mais Jamal, permettant une communication bidirectionnelle entre CARLA et Apollo via l'API Cyber, assurant que les deux systèmes fonctionnent en harmonie pour simuler des scénarios de conduite réalistes et valider les algorithmes de conduite autonome.



Apollo, développé par Baidu, est une plateforme de conduite autonome ouverte qui fournit une suite complète d'outils et de modules pour la perception, la planification, le contrôle et la prise de décision des véhicules autonomes.

Ce logiciel est conçu pour être modulaire, ce qui signifie que chaque composant de son système (par exemple, la perception de l'environnement, la planification des trajets, le contrôle des

véhicules) peut être développé et testé indépendamment avant d'être intégré dans une solution complète. Cette modularité permet une grande flexibilité et facilite le développement et la mise à jour des composants.

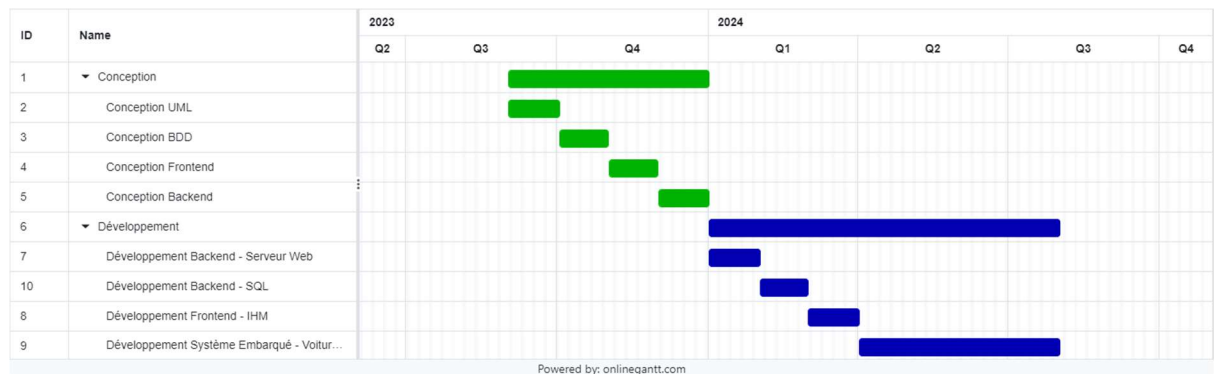


## Planification et gestion de projet

La gestion de projet pour **Perceptive Automotive** a impliqué une planification détaillée et une utilisation efficace des outils de collaboration pour garantir la réussite du projet. Cette section couvre les éléments suivants :

### Stratégies de planification

- **Définition des objectifs** : Les objectifs du projet ont été clairement définis dès le début, incluant le développement d'une solution logicielle permettant le fonctionnement autonome et sécurisé de véhicules sans conducteur. Les objectifs spécifiques incluent la réservation de véhicules, l'optimisation des trajets, la gestion des utilisateurs, et l'intégration des systèmes embarqués via la simulation CARLA.
- **Identification des tâches** : Les tâches principales ont été identifiées et décomposées en sous-tâches pour faciliter la gestion. Ces tâches comprenaient la conception logicielle, le développement, les tests, et les validations.
- **Chronologie des activités** : Un **diagramme de Gantt** a été utilisé pour organiser les tâches dans le temps. Le diagramme de Gantt a joué un rôle central dans la gestion des délais et a permis de s'assurer que toutes les étapes importantes du projet étaient accomplies selon le calendrier prévu.



## Méthode de gestion de projet utilisée

Nous avons utilisé la **méthode en cascade ou Waterfall** pour la gestion de ce projet. Voici pourquoi :

**Séparation des phases** : Le projet est clairement divisé en phases distinctes (Conception, Développement) qui se succèdent sans chevauchement significatif, typique de la méthode Waterfall où chaque phase doit être complétée avant de passer à la suivante.

**Durée définie pour chaque phase** : Les phases de conception (UML, BDD, Frontend, Backend) sont planifiées avant le début de la phase de développement, ce qui est une caractéristique clé de la méthode en cascade.

**Approche linéaire** : Le processus est linéaire, avec une progression logique des étapes du projet (de la conception à la mise en œuvre). Chaque activité semble dépendre de l'achèvement de l'étape précédente.

La méthode Waterfall est souvent utilisée pour des projets où les exigences sont bien comprises dès le début et où les changements sont peu probables. Elle permet de bien structurer le travail et de fixer des jalons clairs.

### Avantages de cette Méthode pour ce Projet

- **Clarté et structure** : Chaque membre de l'équipe sait exactement ce qu'il doit faire et quand.
- **Suivi facile** : Le progrès est facilement mesurable par rapport au plan établi.
- **Documentation complète** : Chaque étape produit des livrables clairement définis.

Cependant, cette méthode peut manquer de flexibilité si des changements importants sont nécessaires en cours de route.

### Outils de Collaboration

- **Outils utilisés** : Des outils de gestion de projet comme **Teams** et **Github** ont été mis en place pour gérer les tâches, attribuer des responsabilités et suivre les progrès en temps réel. Ces outils ont permis une meilleure coordination entre les membres de l'équipe et ont facilité la communication.



- **Réunions de suivi** : Des réunions régulières ont été organisées pour évaluer l'avancement du projet, discuter des problèmes rencontrés, et ajuster les plans si nécessaire. Ces réunions ont également servi de forum pour l'échange d'idées et la résolution de conflits.
- **Documentation** : La documentation a été maintenue à jour tout au long du projet, avec des comptes rendus de réunion, des rapports de progrès, et des mises à jour du diagramme de Gantt. Cette documentation a été cruciale pour garantir la transparence et pour permettre à tous les membres de l'équipe de rester informés des évolutions du projet.

## Résultats et évaluation

- **Tests de validation** : Les tests de validation ont été une partie intégrante de la phase d'exécution, permettant de vérifier que chaque composant du système répondait aux exigences définies. Les scénarios de test ont inclus des simulations réalistes via CARLA pour évaluer les performances des algorithmes de conduite autonome.
- **Retour d'expérience** : À la fin du projet, une évaluation complète a été réalisée pour identifier les points forts et les faiblesses de la gestion du projet. Ce retour d'expérience a été documenté pour être utilisé dans de futurs projets similaires.

## Résultats et évaluation

### Fonctionnalité de réservation

- **Fonctionnement et performance** : La fonctionnalité de réservation a été développée et fonctionne correctement. Les utilisateurs peuvent réserver un véhicule en utilisant l'interface utilisateur dédiée. Cette interface permet une gestion efficace des réservations, notamment en assurant la vérification en temps réel de la disponibilité des véhicules. Le système répond aux demandes des utilisateurs de manière fluide, permettant ainsi une expérience utilisateur satisfaisante.
- **Intégration avec le système global** : La fonctionnalité de réservation est bien intégrée avec les autres composants du système. Elle communique de manière harmonieuse avec les modules de gestion des itinéraires et les systèmes embarqués du véhicule. Cela garantit que les informations de réservation sont transmises et traitées en temps réel, permettant une préparation optimale du véhicule avant l'arrivée de l'utilisateur. Etant donné que nous avons travaillé sur une simulation, il manque la fonctionnalité de déverrouillage du véhicule.

### Problèmes liés au contrôle du véhicule

- **Difficultés lors des virages aux intersections** :



- **Constat** : Lors des tests, il a été observé que le véhicule avait des difficultés à prendre les tournants aux intersections, ce qui a conduit à des écarts par rapport à la trajectoire souhaitée. Ce problème est principalement dû au contrôleur PID latéral, qui est chargé de maintenir le véhicule sur la bonne trajectoire.
- **Analyse Technique** : L'analyse des performances du PID latéral a révélé que les coefficients utilisés pour le contrôle proportionnel, intégral, et dérivatif n'étaient pas suffisamment adaptés aux situations nécessitant des changements de direction rapides et précis, comme c'est le cas aux intersections. Le véhicule montrait une tendance à sous-virer ou à réagir de manière trop lente, ce qui a compromis sa capacité à suivre la trajectoire définie de manière optimale.
- **Tentatives de Correction** : Plusieurs ajustements ont été essayés, tels que la modification des coefficients du PID. Cependant, des ajustements supplémentaires sont nécessaires pour atteindre une performance satisfaisante aux intersections.
- **Rééquilibrage du contrôleur PID Longitudinal** :
  - **Problème identifié** : Le contrôleur PID longitudinal, responsable du contrôle de la vitesse du véhicule, a nécessité un rééquilibrage pour corriger des problèmes de fluctuations de vitesse. Ce contrôleur assure que le véhicule respecte les vitesses limites et maintient une accélération/décélération en douceur en fonction des conditions de conduite.
  - **Processus de rééquilibrage** : Le rééquilibrage a impliqué une révision des coefficients du PID pour mieux répondre aux variations de charge du véhicule et aux conditions routières. Des tests rigoureux ont été menés pour vérifier la stabilité du véhicule après ces ajustements.
  - **Résultats du rééquilibrage** : Après recalibrage, le véhicule a montré une amélioration significative dans le contrôle de la vitesse. Les fluctuations indésirables ont été réduites, permettant au véhicule de maintenir une vitesse plus constante et d'améliorer ainsi la sécurité et le confort des utilisateurs. Les tests ont confirmé que le PID longitudinal fonctionne désormais de manière plus fiable, en particulier dans les scénarios de conduite variés simulés.

## Conclusion

Le projet Perceptive Automotive représente une avancée significative dans le développement de systèmes de conduite autonome intégrés avec des fonctionnalités utilisateur avancées, telles que la réservation de véhicules. Le système de réservation, qui a été mis en œuvre avec succès, permet aux utilisateurs de planifier et de gérer leurs trajets de manière efficace et conviviale. Cette fonctionnalité, bien intégrée avec le backend, assure une synchronisation fluide et en temps réel, offrant une expérience utilisateur fluide.

L'intégration avec des technologies avancées, comme la plateforme Baidu Apollo, a également joué un rôle clé dans le développement de l'intelligence du véhicule. Baidu Apollo a fourni un cadre solide pour le développement des capacités de conduite autonome, contribuant ainsi à l'amélioration continue des algorithmes de contrôle et de navigation. Cependant, les tests ont révélé des défis techniques, notamment dans la gestion des virages aux intersections. Le contrôleur PID latéral a montré des limites dans ces situations, nécessitant des ajustements pour améliorer la réactivité et la précision du véhicule. Les efforts pour affiner ces paramètres seront cruciaux pour surmonter ces défis.

De plus, le réécalonnage du contrôleur PID longitudinal a été un succès, permettant d'améliorer le contrôle de la vitesse et la stabilité du véhicule, ce qui est essentiel pour garantir une conduite sécurisée et confortable. Ce réécalonnage a résolu les fluctuations de vitesse, assurant ainsi une réponse plus stable et prévisible du véhicule.

En conclusion, bien que le projet ait atteint plusieurs de ses objectifs clés, notamment une fonctionnalité de réservation robuste et une intégration réussie avec Baidu Apollo, il reste des domaines à perfectionner, en particulier en ce qui concerne le contrôle de la trajectoire du véhicule dans des environnements urbains complexes. Ces défis offrent des opportunités pour de futures améliorations, qui permettront de peaufiner encore davantage le système et de renforcer sa capacité à opérer de manière autonome dans des conditions réelles. Le projet Perceptive Automotive pose ainsi les bases d'un système de conduite autonome de plus en plus performant, ouvrant la voie à des applications encore plus avancées et sûres dans le domaine de la mobilité intelligente.