

REPUBLIQUE DU SENEGAL



Un peuple-Un but-Une foi

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Direction Générale de l'Enseignement Supérieur Privé

Institut Supérieur d'Informatique



Département : Génie Informatique

Mémoire de fin de cycle pour l'obtention de la Licence professionnelle

Option : Génie Logiciel

Sujet :

**Conception et réalisation d'une Application mobile de
covoiturage pour la ville de Dakar**

Présenté par :

Mme Houleymatou **DIALLO**
M. Cheikh Tidiane **TRAORE**

Sous la direction de : M. Abdoulaye **LY**

Spécialité : Architecte Logiciel

Soutenu à ISI Dakar le,

Membres du Jury

Président		
Superviseur de mémoire		
Examineur 1 :		
Examineur 2 :		
Directeur de mémoire		

Année académique : 2024-2025

REPUBLIQUE DU SENEGAL



Un peuple-Un but-Une foi

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

Direction Générale de l'Enseignement Supérieur Privé

Institut Supérieur d'Informatique



Département : Génie Informatique

Mémoire de fin de cycle pour l'obtention de la Licence professionnelle

Option : Génie Logiciel

Sujet :

**Conception et réalisation d'une Application mobile de
covoiturage pour la ville de Dakar**

Présenté par :

Mme Houleymatou **DIALLO**
M. Cheikh Tidiane **TRAORE**

Sous la direction de : M. Abdoulaye **LY**

Spécialité : Architecte Logiciel

Soutenu à ISI Dakar le,

Membres du Jury

Président		
Superviseur de mémoire		
Examineur 1 :		
Examineur 2 :		
Directeur de mémoire		

Année académique : 2024-2025

Dédicaces

Dédicace de Houleymatou

Je dédie ce mémoire à ma reine mère, mon pilier, pour son amour infini, sa patience sans limite et ses prières silencieuses qui m'ont portée dans l'ombre, chaque jour, sans jamais faillir.

À mon père, pour ses conseils avisés et ses douas, qui m'ont toujours accompagné et soutenu.

À toi, mon grand frère Moussa Diallo, qui as toujours été un pilier depuis mon enfance. Ta présence constante et ton soutien sans faille ont été essentiels tout au long de mon parcours.

À toi, ma grande sœur Sira Woury Diallo, ma deuxième mère, pour ton amour sans condition, ta bienveillance et ton regard protecteur qui m'ont toujours guidée.

À toi, Cheikh Tidiane Traoré, ce mémoire ne serait pas ce qu'il est sans la force tranquille de ta présence. Dans les moments d'incertitude et de doute, ta patience et ta confiance ont été un refuge. Le travail accompli ensemble porte la marque d'une complicité singulière, forgée au fil du temps, au-delà des simples échanges professionnels. Ce lien puissant est gravé dans chacune des lignes de ce document.

Dédicace de Cheikh Tidiane

À la mémoire de mon père, mon tout, mon modèle, mon ancrage. Son absence physique n'a jamais effacé sa présence dans ma vie. Il a toujours cru en moi, voulu me voir réussir, et sa foi m'accompagne chaque jour. Ce mémoire, je le lui dois autant qu'à moi.

À ma mère, qui a tenu bon depuis son départ. Elle a su combler chaque manque, chaque silence, par sa force, son amour et son dévouement sans faille. Grâce à elle, je n'ai jamais manqué de rien, même dans les moments les plus durs. À mon oncle, Pape Massar Sow, dont le soutien constant, discret mais précieux, a joué un rôle essentiel dans la poursuite de mes études.

À vous, mes frères, Silly Traoré et Idrissa Traoré, qui ont su rester solides malgré les épreuves. Leur force, leur présence silencieuse et leur amour m'ont toujours porté.

Et enfin, à Houleymatou Diallo. Partenaire de mémoire, soutien constant et présence précieuse. Son engagement dans ce travail a été sans faille, mais sa place dans ma vie dépasse largement ce cadre. Ces mots ne sauraient exprimer toute la gratitude que je lui dois, mais qu'ils restent un humble témoignage de ce qu'elle représente.

Remerciements

Par la grâce d'Allah, Le Tout-Miséricordieux, nous avons pu mener à bien ce mémoire. À Lui reviennent louange et reconnaissance. Que la paix et les bénédictions soient sur Son Prophète Muhammad (ﷺ), modèle d'excellence et de foi.

Nous remercions l'ensemble des enseignants de L'ISI, dont les efforts constants ont forgé notre parcours académique.

Nous adressons nos sincères remerciements à notre encadrant M. Abdoulaye LY, pour ses conseils, sa rigueur et sa disponibilité.

Nos remerciements s'étendent également à nos camarades de promotion, pour les échanges stimulants, les entraides spontanées et les bons moments partagés.

Enfin, nous exprimons notre reconnaissance à nos familles et nos proches pour leur soutien moral et leur patience au fil de cette aventure académique.

Houleymatou

Je souhaite remercier très sincèrement l'ONG Yaakar Africa, pour son soutien financier sans lequel ce parcours n'aurait pas été possible.

Ma gratitude va tout particulièrement à Monsieur Demba Dieng, pour ses conseils avisés, sa bienveillance et son accompagnement constant.

J'exprime également ma reconnaissance à mes colocataires de chambre, pour leur patience et leur compréhension au quotidien.

Je souhaite exprimer ma gratitude à tous mes amis et copines, dont le soutien et l'affection m'ont été d'un précieux réconfort durant ces trois années. Je ne peux pas les mentionner un par un, mais chacun(e) se reconnaîtra dans ces lignes.

Liste des Figures

Figure 1: Représentation schématique du cadre Agile et de ses méthodes associées.....	5
Figure 2: Représentation schématique de Kanban.....	6
Figure 3 : Tableau Trello du projet TYVAA utilisé pour la planification et le suivi des tâches selon la méthode Kanban	7
Figure 4: Diagramme de Contexte.....	10
Figure 5: Diagramme de Cas d'utilisation Administrateur	11
Figure 6: Diagramme de Cas d'utilisation Chauffeur	12
Figure 7: Diagramme de Cas d'utilisation Passager	13
Figure 8: Diagramme de Classe.....	14
Figure 9 : Ecran Figma de l'accueil Passager	15
Figure 10: Ecran Figma des chatbot	15
Figure 11 : Ecran Figma de la page de connexion Passager	16
Figure 12: Logo de l'outil Figma.....	18
Figure 13: Logo de l'outil Word.....	19
Figure 14: Logo de l'outil Canva.....	19
Figure 15: Logo de l'outil Webstorm.....	19
Figure 16: Logo de l'outil Android Studio	20
Figure 17: Logo de l'outil Github	20
Figure 18 : Dépôt Git – Backend	21
Figure 19: Dépôt Git – Panneau d'administration	21
Figure 20: Dépôt Git – Application mobile Flutter	22
Figure 21: Logo de l'outil Github Action	22
Figure 22: Pipeline CI/CD TYVAA	23
Figure 23: Logo de l'outil Docker	23
Figure 24: Logo de l'outil Terraform.....	23
Figure 25: Logo de l'outil Koyeb	24
Figure 26: Logo de l'outil Azure	24
Figure 27: Logo de l'outil Sonar.....	24
Figure 28: Logo de l'outil Mocha et Jest	25

Figure 29: Logo de l'outil JWT et OWASP	25
Figure 30: Logo de l'outil Dependabot et Pino	26
Figure 31: Logo de l'outil Firebase Cloud Messagin et RabbitMQ	26
Figure 32: Logo de l'outil Name.com et Zoho	27
Figure 33: Logo de l'outil Genkit	27
Figure 34: Écran de l'application – Page d'accueil et Page de Selection ChatBot	28
Figure 35: Écran de l'application – Page de selection point de départ, arrivé et créneau horaire.	28
Figure 36: Écran de l'application Panneau Admin – Page d'accueil, gestion RBAC, Demande d'application.....	30

Liste des Tableaux

Tableau 1:Glossaire des sigles, acronymes et termes techniques	VII
--	-----

Glossaire

Sigle / Acronyme	Signification	Définition
ACID	Atomicity, Consistency, Isolation, Durability	Propriétés garantissant la fiabilité des transactions dans une base de données relationnelle.
AJAX	Asynchronous JavaScript and XML	Technique permettant de charger des données en arrière-plan sans recharger la page entière.
API	Application Programming Interface	Interface de programmation Permettant à des applications de communiquer entre elles.
CI/CD	Continuous Integration / Continuous Deployment	Pratiques d'intégration et de déploiement continus pour automatiser les tests et la mise en production.
FCM	Firebase Cloud Messaging	Service de Google pour l'envoi de notifications push sur mobile.
GENKIT AI	Generative Toolkit for AI-driven tools & services.	Plateforme utilisant l'intelligence artificielle pour créer et gérer automatiquement des outils ou services numériques.
IDE	Integrated Development Environment	Environnement de développement intégré.
ISI	Institut Supérieur d'Informatique	Établissement d'enseignement supérieur privé à Dakar.
JSON	JavaScript Object Notation	Format léger d'échange de données, lisible par l'homme et facile à parser par les machines.

JWT	JSON Web Token	Standard ouvert pour la transmission sécurisée d'informations entre deux parties.
ORM	Object Relational Mapping	Technique permettant de manipuler une base de données relationnelle à travers des objets.
OWASP	Open Web Application Security Project	Projet open source qui fournit des guides et outils pour sécuriser les applications web.
RBAC	Role-Based Access Control	Modèle de contrôle d'accès basé sur les rôles des utilisateurs.
SQL	Structured Query Language	Langage de requêtes structuré utilisé pour gérer et manipuler les bases de données relationnelles.
UML	Unified Modeling Language	Langage de modélisation normalisé pour l'analyse et la conception logicielle.
UX/UI	User Experience / User Interface	Expérience utilisateur / Interface utilisateur.
WIP	Work In Progress	Limite du travail en cours, utilisée notamment dans la méthode Kanban.

Tableau 1: Glossaire des sigles, acronymes et termes techniques

Sommaire

CHAPITRE 1 :	INTRODUCTION GENERALE	1
1.1	Présentation de la structure d'accueil	1
1.2	Contexte	2
1.3	Problématique.....	2
1.4	Objectifs du mémoire.....	3
CHAPITRE 2 :	TRAVAUX REALISES AU COURS DU STAGE.....	5
2.1	Gestion Projet Agile	5
2.2	Spécialisations Fonctionnelles	7
2.3	Méthodes D'Analyse et de Conception	9
2.4	Maquettisation de la solution	14
2.5	Réalisation.....	16
2.6	Presentation de la solution	27
CHAPITRE 3 :	BILAN.....	31
3.1	Objectifs Atteints.....	31
3.2	Objectifs Non Atteints	32
3.3	Intérêts Personnels	32

Résumé

Ce projet de fin d'étude présente TYVAA, une application mobile de covoiturage urbain destinée à faciliter les déplacements à Dakar. Elle met en relation des conducteurs disposant de places libres et des passagers souhaitant effectuer un trajet, dans une logique de partage et de réduction des coûts de transport.

L'application permet de proposer ou rechercher un trajet, qu'il soit ponctuel ou régulier, avec la possibilité pour un utilisateur d'être à la fois conducteur et passager. Les fonctionnalités incluent la réservation de trajets à l'avance, la consultation des disponibilités, ainsi que des outils favorisant la confiance tels que la vérification des profils et les avis.

En offrant une solution simple, conviviale et adaptée aux besoins locaux, TYVAA contribue à améliorer la mobilité urbaine, à réduire la congestion routière et à encourager des pratiques de transport plus collaboratives.

Abstract

This final year project presents **TYVAA**, a mobile carpooling application designed to improve urban mobility in Dakar. It connects drivers with available seats to passengers wishing to travel, promoting cost sharing and more sustainable transport practices.

The application allows users to offer or search for a ride, whether occasional or regular, with the flexibility for each member to act as both driver and passenger. Key features include advance booking, availability checking, and trust-building tools such as profile verification and user reviews.

By providing a simple, user-friendly, and locally adapted solution, TYVAA aims to ease urban mobility, reduce traffic congestion, and encourage more collaborative transportation practices.

Chapitre 1 : INTRODUCTION GENERALE

1.1 Présentation de la structure d'accueil

L'Institut Supérieur d'Informatique (ISI) est un établissement privé d'enseignement supérieur fondé en 1994, reconnu par l'État sénégalais. Il propose des formations en informatique et en gestion, avec des diplômes accrédités par le ministère de l'Enseignement supérieur et contrôlés par l'Agence Nationale d'Assurance Qualité de l'Enseignement Supérieur (ANAQ-Sup).

Le groupe ISI compte plus de 2 500 étudiants répartis sur neuf campus, dont sept au Sénégal et deux en Mauritanie. L'institut offre des programmes de formation de 2, 3 et 5 ans dans les domaines de l'ingénierie informatique, des réseaux et systèmes, ainsi que de la gestion.

L'ISI se distingue par son approche pédagogique axée sur la pratique et l'innovation. Les étudiants participent à des projets de recherche, effectuent des stages en entreprise et collaborent avec des professionnels du secteur. Cette approche vise à développer des compétences techniques solides, ainsi que la créativité et l'esprit d'initiative, préparant ainsi les diplômés à relever les défis technologiques et à contribuer au développement socio-économique.

Dans le cadre de cette approche professionnalisante, et pour l'obtention de la licence en Génie Logiciel, l'ISI exige la réalisation d'un mémoire de fin de cycle. C'est dans ce contexte que nous avons élaboré ce travail, portant sur : *l'étude et la réalisation d'une plateforme de covoiturage urbain adaptée au contexte dakarois*.

Ce mémoire, fruit de notre stage académique réalisé au sein même de l'ISI, nous a permis d'appliquer nos acquis dans un environnement structuré, avec un encadrement pédagogique rigoureux. Il constitue notre première contribution académique, et nous sollicitons de ce fait l'indulgence du jury pour son évaluation.

1.2 Contexte

Dakar, capitale du Sénégal, est aujourd'hui le centre névralgique de l'activité économique, administrative et sociale du pays. Sa croissance démographique rapide, combinée à une urbanisation accélérée, a profondément transformé la dynamique de mobilité urbaine. Chaque jour, des centaines de milliers de personnes se déplacent pour se rendre au travail, à l'école, ou pour diverses activités, générant une forte pression sur les infrastructures existantes. Pourtant, malgré les efforts déployés par l'État et les collectivités pour améliorer le transport public, les difficultés persistent.

Les embouteillages quotidiens, devenus quasi structurels, entraînent une perte considérable de temps et d'énergie pour les usagers. Les transports publics, souvent surchargés, ne parviennent pas à satisfaire la demande croissante, tandis que le recours aux taxis ou aux véhicules particuliers entraîne un coût élevé pour les ménages et accentue la saturation routière. Ces contraintes pèsent lourdement sur la productivité, la santé et même sur l'environnement, avec une augmentation des émissions de gaz à effet de serre.

Dans ce contexte marqué par une demande croissante de solutions de transport plus efficaces, le covoiturage émerge comme une alternative crédible. En permettant à plusieurs personnes se rendant dans la même direction de partager un véhicule, il propose une approche plus économique, plus écologique et plus solidaire de la mobilité urbaine. Cette pratique, déjà adoptée dans plusieurs grandes métropoles à travers le monde, pourrait constituer une réponse innovante et adaptée aux réalités dakaroises.

1.3 Problématique

Avec l'émergence des solutions numériques, de nouvelles perspectives s'ouvrent pour repenser la mobilité urbaine. Le covoiturage, en particulier, apparaît comme une alternative prometteuse, car il permet de mutualiser les trajets, de réduire les dépenses individuelles et de rationaliser l'utilisation des véhicules particuliers. Cependant, la mise en place d'une telle solution soulève plusieurs défis : comment assurer la confiance entre utilisateurs ? Comment adapter le modèle du covoiturage aux spécificités sociales et économiques de Dakar ? Et surtout, comment concevoir une plateforme numérique qui soit à la fois accessible, sécurisée, conviviale et réellement efficace dans ce contexte urbain complexe ?

C'est à cette problématique que s'intéresse ce travail, en cherchant à concevoir et développer une solution innovante de covoiturage urbain adaptée au contexte dakarois, tout en répondant aux

besoins des conducteurs comme des passagers, et en contribuant à la réduction des coûts, à l'amélioration de la fluidité du trafic et à la durabilité environnementale.

1.4 Objectifs du mémoire

1.4.1 Objectif générale

L'objectif de ce mémoire est de concevoir et de développer une **application mobile de covoiturage** adaptée aux réalités urbaines de Dakar, dans le but de **réduire les coûts de transport, faciliter les trajets partagés et améliorer la mobilité urbaine** à travers une solution numérique simple, accessible et durable.

1.4.2 Objectifs spécifiques

- ❖ *Module Utilisateurs* : Permettre la création, l'authentification et la gestion des profils des utilisateurs, incluant la gestion des rôles (conducteur/passager) et la protection rigoureuse des données personnelles.
- ❖ *Module Trajets* : Offrir aux conducteurs la possibilité de publier, modifier et supprimer leurs offres de trajets, en gérant les informations clés comme lieux, date, heure, nombre de places, prix et commentaires.
- ❖ *Module Réservations* : Permettre aux passagers de rechercher, réserver et annuler des places sur des trajets disponibles, avec mise à jour en temps réel des disponibilités.
- ❖ *Module Paiement sécurisé* : Intégrer un système fiable de gestion des transactions financières entre utilisateurs, garantissant sécurité, fiabilité et traçabilité des paiements.
- ❖ *Module Chatbot intelligent* : Proposer un assistant virtuel capable d'accompagner les utilisateurs dans la navigation, la recherche de trajets et de répondre à leurs questions de manière interactive.
- ❖ *Module Suivi et monitoring administrateur* : Fournir des outils de supervision permettant à l'administrateur de contrôler l'activité de la plateforme, gérer les utilisateurs et garantir la qualité du service.
- ❖ *Sécurité et performance globale* : Assurer la protection des données et la sécurisation des échanges sur toutes les fonctionnalités, tout en optimisant la performance et la capacité d'évolution de la solution.

- ❖ *Modularité et évolutivité* : Concevoir chaque module de façon indépendante afin de faciliter les mises à jour futures et l'intégration de nouvelles fonctionnalités, telles que la géolocalisation ou d'autres services avancés.

Chapitre 2 : TRAVAUX REALISES AU COURS DU STAGE

2.1 Gestion Projet Agile

La méthode Agile est une approche de gestion de projet orientée vers l'adaptabilité, la livraison continue et la satisfaction du client. Contrairement aux méthodes traditionnelles comme le cycle en V, Agile privilégie les cycles courts et itératifs, permettant de livrer progressivement des versions fonctionnelles du produit tout en intégrant les retours réguliers des parties prenantes.

Les valeurs fondamentales de l'Agilité, issues du Manifeste Agile, sont les suivantes :

- ❖ Les individus et leurs interactions plus que les processus et les outils
- ❖ Un logiciel fonctionnel plus qu'une documentation exhaustive
- ❖ La collaboration avec le client plus que la négociation contractuelle
- ❖ L'adaptation au changement plus que le suivi d'un plan rigide

Cette approche est particulièrement bien adaptée aux projets informatiques comme TYVAA, où les besoins évoluent et où la souplesse est un avantage essentiel.

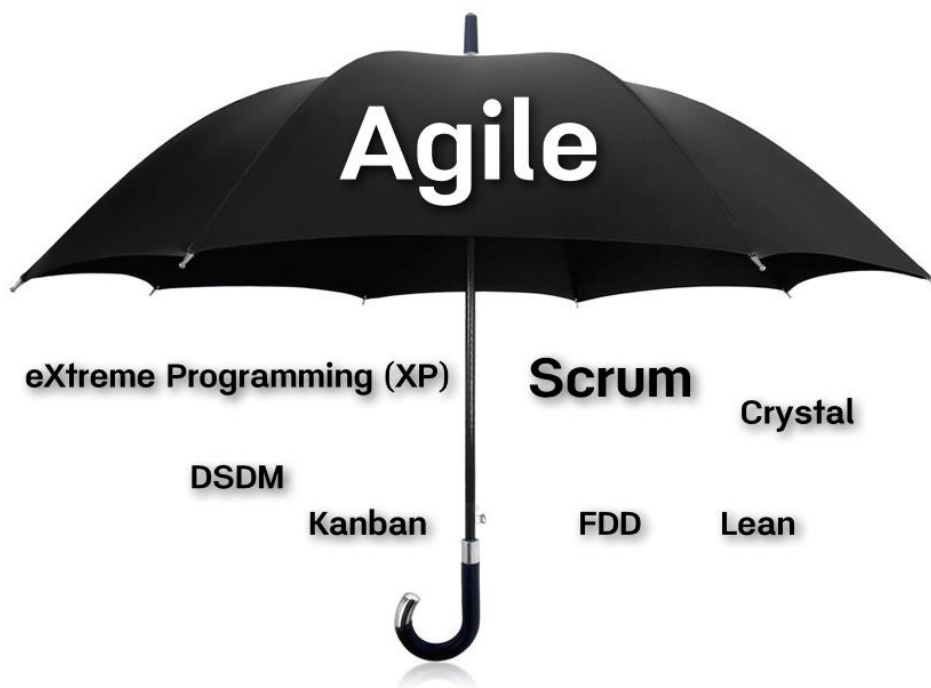


Figure 1: Représentation schématique du cadre Agile et de ses méthodes associées

2.1.1 Méthode Kanban



Figure 2: Représentation schématique de Kanban

Kanban est une méthode Agile centrée sur la visualisation du flux de travail à travers un tableau composé de colonnes représentant les différentes étapes de production (par exemple : À faire, En cours, Terminé). Chaque tâche est représentée par une carte qui circule au fur et à mesure de son avancement. Cette méthode permet une gestion fluide, limite le travail en cours (WIP), et facilite l'identification des blocages.

Dans le projet TYVAA, cette méthode a été choisie car elle est adaptée aux équipes réduites, offre une grande lisibilité sur l'état d'avancement et permet d'ajuster les priorités en temps réel. Elle favorise aussi une organisation souple, sans sprints fixes ni rôles formalisés comme dans Scrum.

2.1.2 Outil Trello

Pour mettre en œuvre Kanban, nous avons utilisé Trello, un outil de gestion de projet en ligne basé sur des tableaux visuels. Chaque fonctionnalité de TYVAA (par exemple : publication de trajet, réservation, paiement) a été modélisée sous forme de carte dans Trello, avec des colonnes telles que : À faire, En cours, En test, Terminé.

Trello a permis une collaboration efficace entre les deux membres du binôme, une meilleure organisation des tâches et une traçabilité claire des progrès réalisés. Grâce à cet outil, la planification a pu s'ajuster en fonction des imprévus techniques ou des contraintes de temps, tout en gardant une vue d'ensemble claire sur l'évolution du projet.

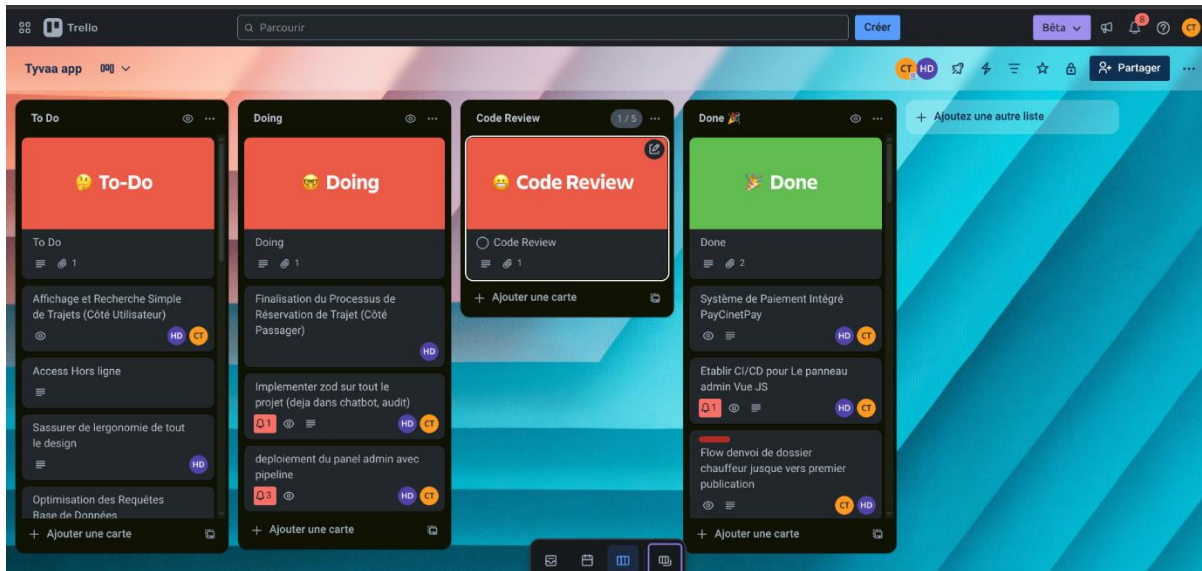


Figure 3 : Tableau Trello du projet TYVAA utilisé pour la planification et le suivi des tâches selon la méthode Kanban

2.2 Spécialisations Fonctionnelles

2.2.1 Les Besoins Fonctionnelles

Les besoins fonctionnels décrivent les fonctionnalités spécifiques que le système TYVAA doit fournir aux utilisateurs pour répondre aux objectifs du projet. Ils correspondent à **ce que le système doit faire** d'un point de vue opérationnel.

Voici les principaux besoins fonctionnels identifiés pour l'application TYVAA :

- ❖ **Inscription et authentification des utilisateurs**

Les utilisateurs doivent pouvoir créer un compte et se connecter.

- ❖ **Gestion des profils utilisateurs**

Chaque utilisateur peut consulter, modifier et supprimer son profil.

- ❖ **Publication de trajets par les conducteurs**

Le conducteur doit pouvoir publier un trajet en précisant le lieu de départ, la destination, la date, l'heure, le nombre de places disponibles, le prix, et un commentaire éventuel.

- ❖ **Recherche et réservation de trajets par les passagers**

Les passagers peuvent rechercher un trajet selon différents critères (lieu, date, prix...) et réserver une ou plusieurs places.

- ❖ **Annulation de trajet ou de réservation**

Les utilisateurs doivent pouvoir annuler un trajet publié ou une réservation effectuée, selon certaines conditions.

❖ **Paiement en ligne sécurisé**

Un module permet d'effectuer le paiement via une passerelle intégrée, avec retour du statut de la transaction.

❖ **Notification des actions**

Le système envoie des notifications pour confirmer les réservations, les paiements, ou signaler des changements.

❖ **Accès à l'historique des trajets**

Chaque utilisateur peut consulter la liste de ses trajets passés (en tant que conducteur ou passager).

2.2.2 Les Besoins Non Fonctionnelles

Les besoins non fonctionnels définissent les **contraintes techniques, de performance, de sécurité ou d'ergonomie** que l'application TYVAA doit respecter. Ils ne décrivent pas ce que le système fait, mais **comment il doit le faire** pour être efficace, fiable et adapté aux utilisateurs.

Voici les principaux besoins non fonctionnels du projet TYVAA :

Accessibilité multiplateforme

L'application doit être accessible sur différents appareils, notamment les smartphones Android et iOS.

❖ **Simplicité d'utilisation (ergonomie)**

L'interface doit être intuitive, claire et facile à prendre en main, même pour des utilisateurs peu familiers avec les applications mobiles.

❖ **Réactivité et performance**

L'application doit répondre rapidement aux actions de l'utilisateur, avec des temps de chargement courts.

❖ **Sécurité des données personnelles**

Les données des utilisateurs (nom, email, mot de passe, historique de trajets, etc.) doivent être protégées contre toute forme de fuite, de vol ou de modification non autorisée.

❖ **Fiabilité du système de paiement**

Les paiements doivent être sécurisés, traçables et intégrés dans un environnement conforme aux normes.

❖ **Tolérance aux pannes**

Le système doit rester fonctionnel même en cas de perte temporaire de connexion Internet ou de panne partielle.

❖ **Scalabilité (évolutivité)**

Le système doit pouvoir supporter un nombre croissant d'utilisateurs sans perte de performance.

❖ **Maintenabilité**

Le code de l'application doit être clair, modulaire et documenté afin de faciliter les futures mises à jour ou corrections.

❖ **Conformité aux normes locales**

L'application doit respecter les lois en vigueur au Sénégal concernant la protection des données et les transactions électroniques.

2.3 Méthodes D'Analyse et de Conception

2.3.1 Méthode UML

Pour l'analyse et la conception de notre application, nous avons utilisé la modélisation UML afin de représenter les besoins fonctionnels, la structure et les interactions du système.

Dans ce cadre, nous avons réalisé trois types de diagrammes :

- **Diagramme de contexte**, pour visualiser l'environnement global du système et ses échanges avec les acteurs externes.
- **Diagramme de classes**, pour décrire les entités principales, leurs attributs et leurs relations.
- **Diagramme de cas d'utilisation**, pour illustrer les fonctionnalités accessibles à chaque acteur.

2.3.2 Diagramme de contexte

Le diagramme de contexte de *TYVAA* représente l'application dans son ensemble, en montrant ses interactions avec les conducteurs, et les passagers. Il permet de délimiter clairement le périmètre fonctionnel du système.

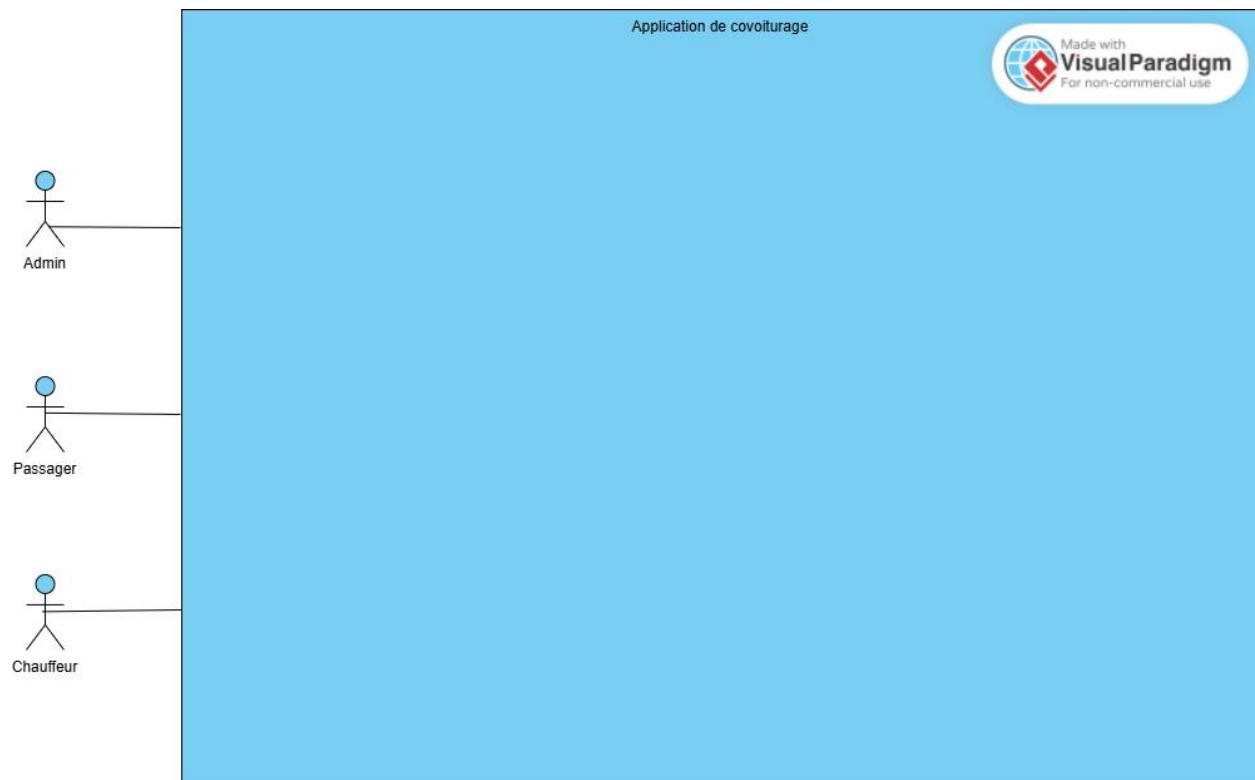


Figure 4: Diagramme de Contexte

2.3.3 Diagrammes de cas d'utilisation

Dans notre application, les cas d'utilisation sont organisés selon trois acteurs principaux : l'administrateur, le chauffeur et le passager. Cette répartition met en évidence les fonctionnalités propres à chaque type d'utilisateur et délimite clairement le périmètre fonctionnel de chacun.

❖ Cas d'utilisation – Administrateur

L'administrateur peut gérer les utilisateurs, consulter les trajets et réservations, et superviser le bon fonctionnement du système.

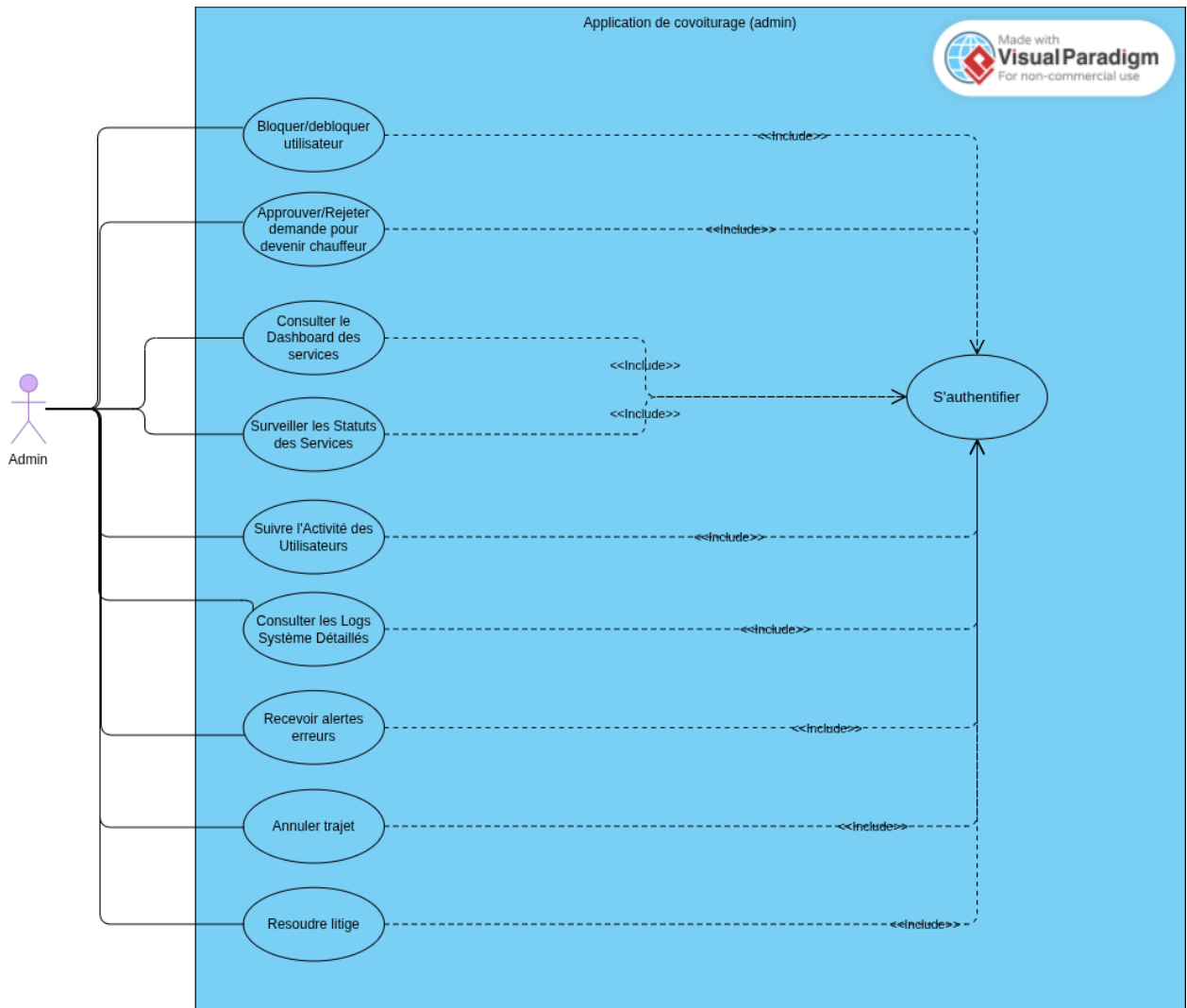


Figure 5: Diagramme de Cas d'utilisation Administrateur

❖ Cas d'utilisation – Chauffeur

Le chauffeur peut publier, modifier ou annuler ses trajets, consulter les réservations et interagir avec les passagers.

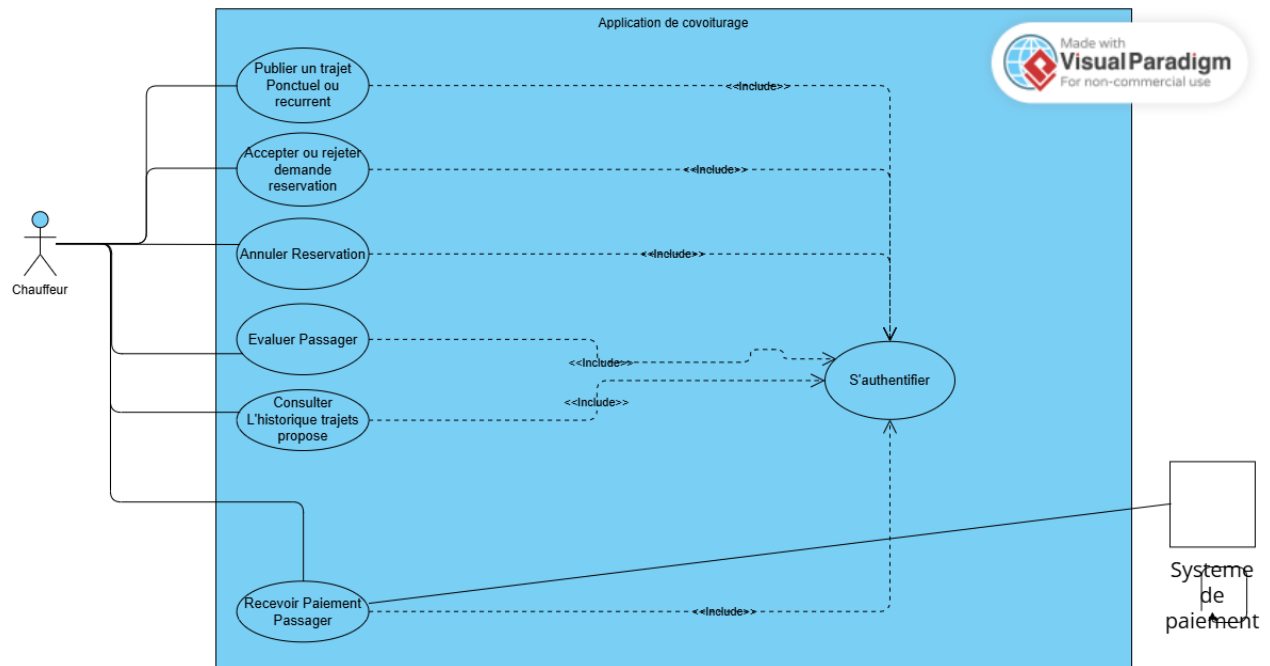


Figure 6: Diagramme de Cas d'utilisation Chauffeur

❖ Cas d'utilisation – Passager

Le passager peut rechercher des trajets, réserver une place, suivre ses réservations et consulter son historique.

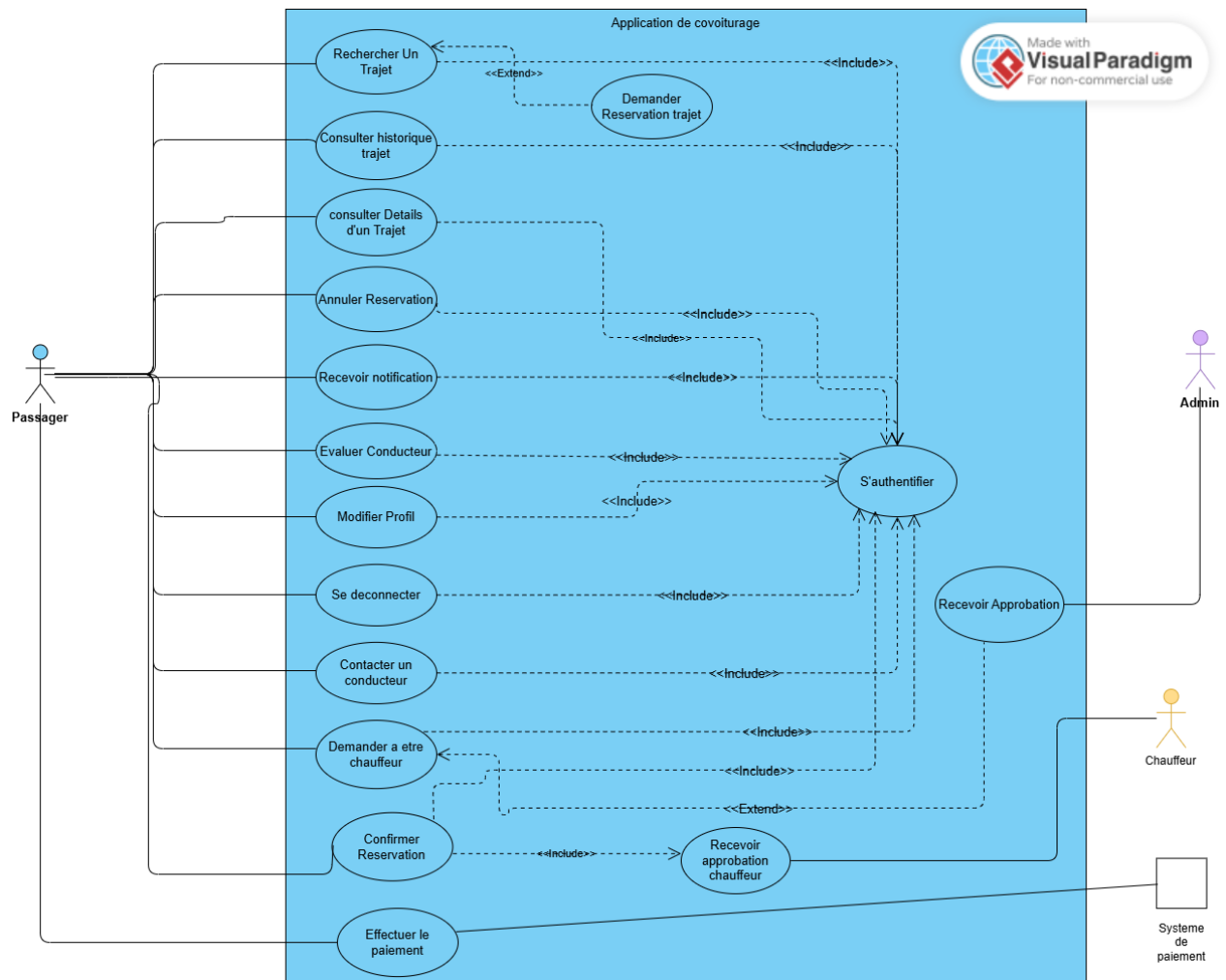


Figure 7: Diagramme de Cas d'utilisation Passager

2.3.4 Diagramme de classe

Le diagramme de classes de TYVAA illustre la structure statique de l'application. Il représente les principales entités manipulées, telles que les utilisateurs, les chauffeurs, les passagers, les trajets et les réservations, ainsi que les relations qui les lient.

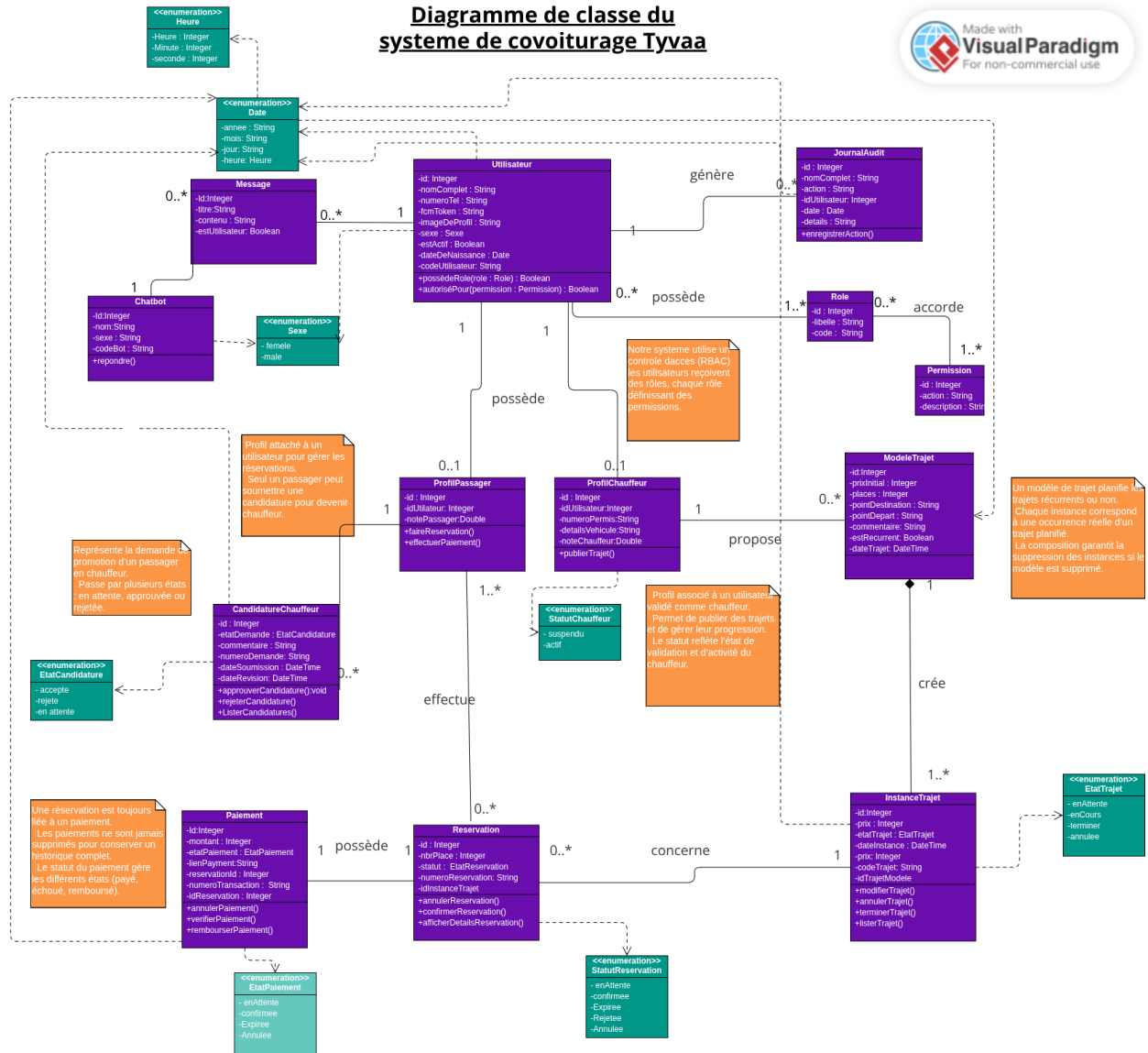


Figure 8: Diagramme de Classe

2.4 Maquettisation de la solution

Pour visualiser l'interface utilisateur de TYVAA avant le développement, nous avons réalisé la maquette de l'application à l'aide de Figma. Cette étape a permis de :

- Définir l'organisation des écrans et la navigation entre les différentes fonctionnalités ;
- Tester l'ergonomie et l'expérience utilisateur pour chaque acteur (administrateur, chauffeur, passager) ;
- Uniformiser le design en respectant la charte graphique de l'application

Les captures d'écran ci-dessous illustrent les principales pages de l'application, telles que le tableau de bord, la recherche et réservation de trajets, la publication de trajets par le chauffeur, ainsi que les écrans de gestion pour l'administrateur.

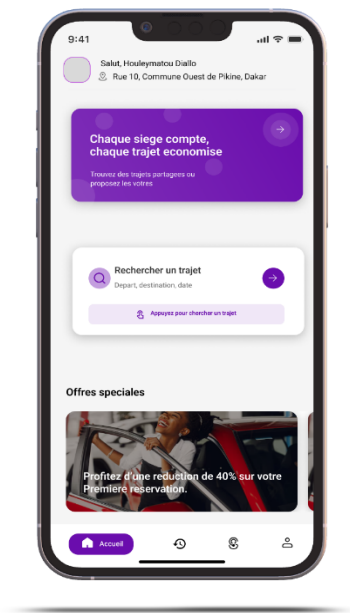


Figure 9 : Ecran Figma de l'accueil Passager

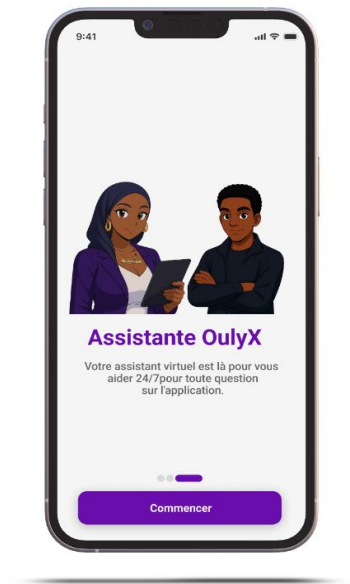


Figure 10: Ecran Figma des chatbot

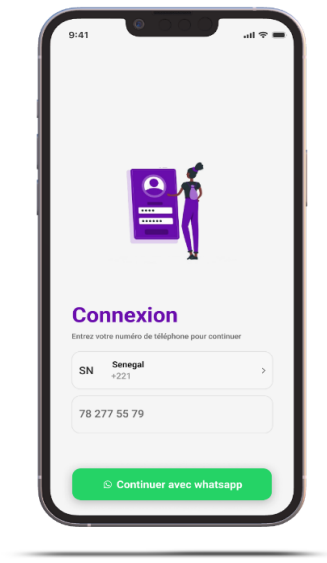


Figure 11 : Ecran Figma de la page de connexion Passager

2.5 Réalisation

2.5.1 Architecture

Le projet repose sur une architecture **client-serveur** en trois couches, séparant les responsabilités pour assurer une meilleure maintenabilité et évolutivité de l'application. Cette architecture est composée des éléments suivants :

- ❖ **La couche de présentation (frontend) :**

Elle est responsable de l'interface utilisateur. Cette couche permet aux utilisateurs d'interagir avec l'application via un navigateur web. Elle consomme les services exposés par l'API backend, en affichant dynamiquement les données selon les actions des utilisateurs.

- ❖ **La couche de logique métier (backend) :**

Cette couche contient les règles de gestion, la validation des données et le traitement principal des requêtes. Elle centralise l'ensemble des opérations métiers (création, mise à jour, suppression, etc.) et interagit directement avec la base de données via un ORM.

- ❖ **La couche de données (base de données) :**

Elle stocke toutes les informations nécessaires au fonctionnement de l'application. L'organisation des données suit un schéma relationnel cohérent, facilitant les requêtes, les relations entre entités et l'intégrité des données.

Cette structure modulaire favorise la réutilisabilité du code, la séparation des préoccupations, ainsi que la possibilité de faire évoluer indépendamment chaque composant du système. Elle s'inscrit dans une démarche moderne de développement d'applications web distribuées

2.5.2 Technologies et Langages Utilisés

Le langage principal utilisé pour la couche serveur est **JavaScript**, dans son exécution côté serveur grâce à **Node.js**. JavaScript, langage orienté objet et fonctionnel, permet une grande flexibilité, tandis que Node.js apporte un environnement d'exécution asynchrone, non-bloquant, particulièrement adapté aux applications temps réel.

Le Framework retenu est **Fastify**, choisi pour sa performance supérieure à Express dans des contextes de forte charge. Minimaliste, extensible et compatible avec les normes REST, Fastify facilite la mise en place d'API tout en garantissant un haut niveau de sécurité et une excellente gestion des erreurs.

L'accès à la base de données est assuré via **Sequelize**, un ORM (Object-Relational Mapping) permettant de manipuler les données sous forme d'objets JavaScript tout en abstrayant les requêtes SQL complexes. Ce choix favorise la lisibilité du code, la validation automatique des données, et l'indépendance vis-à-vis du moteur de base de données.

Les données sont stockées dans **PostgreSQL**, un système de gestion de base de données relationnelle open source reconnue pour sa stabilité, sa conformité ACID, sa prise en charge avancée des types de données, et sa performance dans les environnements à forte volumétrie.

❖ Interface d'administration (Frontend Web)

Le développement du panneau d'administration web repose également sur **JavaScript**, dans sa version moderne, enrichie par l'utilisation du Framework **Vue.js**. Ce dernier se distingue par son approche déclarative, sa structure basée sur des composants réutilisables, et son système de gestion réactive des données.

Vue.js permet de concevoir des interfaces riches, dynamiques et maintenables, tout en garantissant une séparation claire entre la logique métier et la présentation. L'interaction avec l'API backend s'effectue via des appels HTTP asynchrones (AJAX), en format JSON.

❖ Application mobile

La couche mobile est développée avec **Dart**, un langage moderne, typé et orienté objet, conçu pour optimiser le développement multiplateforme. Sa compilation en code natif permet d'obtenir des performances proches des applications natives, tout en conservant une base de code unifiée.

Le Framework associé est **Flutter**, maintenu par Google, qui permet de concevoir des interfaces mobiles expressives et fluides pour Android et iOS. Flutter se distingue par son moteur de rendu intégré, ses nombreux widgets personnalisables, et sa capacité à produire des interfaces graphiques très réactives avec une seule base de code.

2.5.3 Outils Utilisés

Pour accompagner le développement et garantir une productivité optimale, plusieurs outils ont été mobilisés tout au long du cycle de vie du projet. Ces outils couvrent la gestion de code source, le travail collaboratif, l'intégration continue, ainsi que la gestion de versions.

2.5.4 Outils de de modélisation

❖ Visual Paradigm

Dans ce projet nous avons utilisé Visual Paradigm pour créer les diagrammes UML (cas d'utilisation, séquences, classes, base de données). Structure l'analyse initiale et soutient la cohérence de l'implémentation.

2.5.5 Outils de prototypage

❖ Figma



Figure 12: Logo de l'outil Figma

Utilisé pour le prototypage des interfaces, aussi bien pour l'application mobile que pour le panneau admin. Il a permis de concevoir les écrans, définir les parcours utilisateurs et valider les choix UX/UI en amont.

2.5.6 Outils de productivité

Pour documenter, présenter et valoriser le projet, nous avons mobilisé :

❖ Microsoft Word



Figure 13: Logo de l'outil Word

Pour la **rédaction structurée du mémoire**, incluant la mise en page académique, les annexes et les tableaux.

❖ Canva



Figure 14: Logo de l'outil Canva

Pour la **conception graphique des visuels**, schémas d'architecture, maquettes, slides de présentation, et autres supports de communication liés à l'application.

Ces outils ont permis une communication claire et professionnelle autour du projet TYVAA, tant sur le fond que sur la forme.

2.5.7 Outils de développement

❖ WebStorm



Figure 15: Logo de l'outil Webstorm

Environnement de développement principal pour le backend (Fastify avec Node.js) et le frontend admin (Vue Js). Il offre une bonne prise en charge de TypeScript, des outils intégrés pour Git, et un refactoring puissant.

❖ **Android Studio**



Figure 16: Logo de l'outil Android Studio

Nous avons choisi Android Studio pour développer l'application mobile Flutter. Il facilite le hot reload, le debug natif et l'intégration avec les émulateurs Android.

2.5.8 Outils de Gestion du Code et CI/CD

❖ **GitHub**



Figure 17: Logo de l'outil Github

Utilisé pour la gestion du code source, le versioning collaboratif, la revue de code et la centralisation du projet.

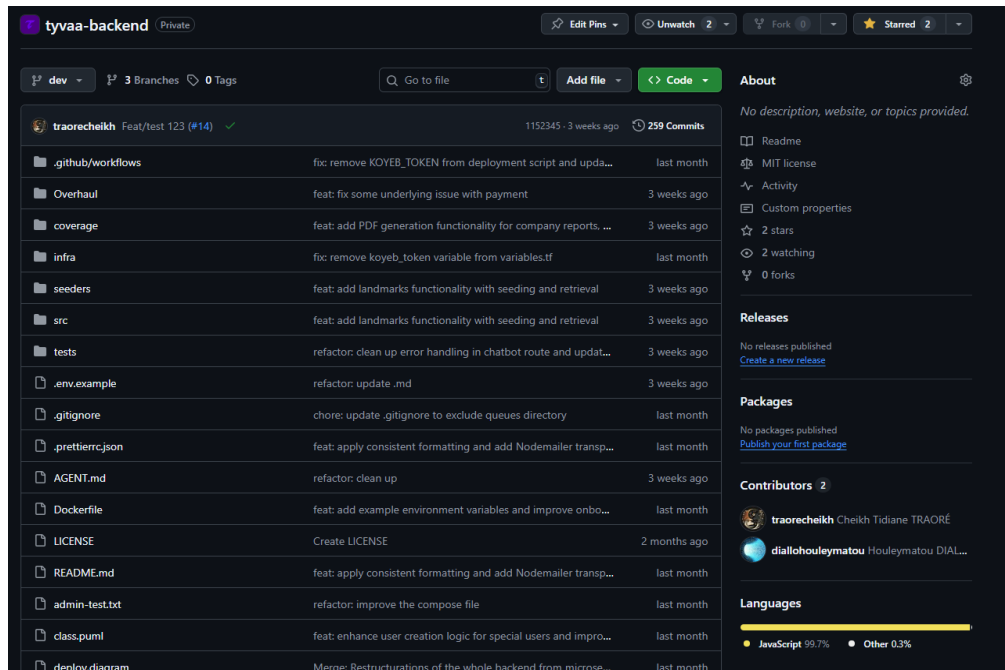


Figure 18 : Dépôt Git – Backend

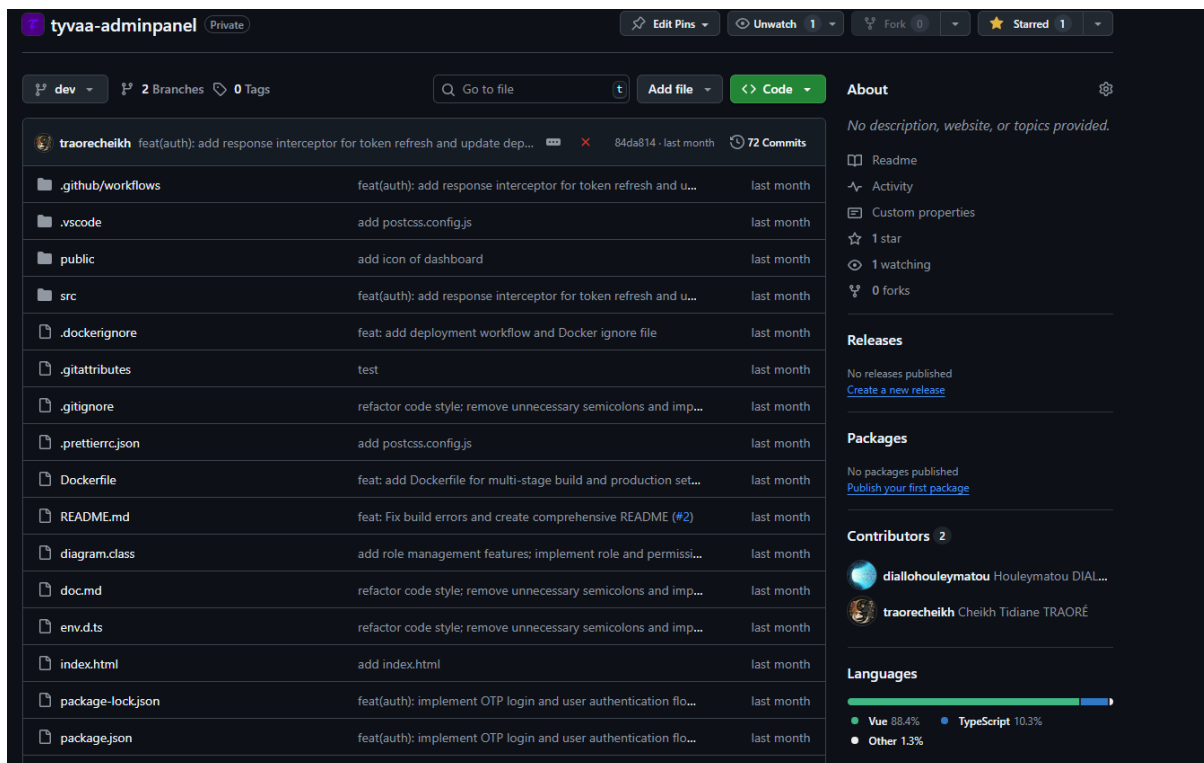


Figure 19: Dépôt Git – Panneau d'administration

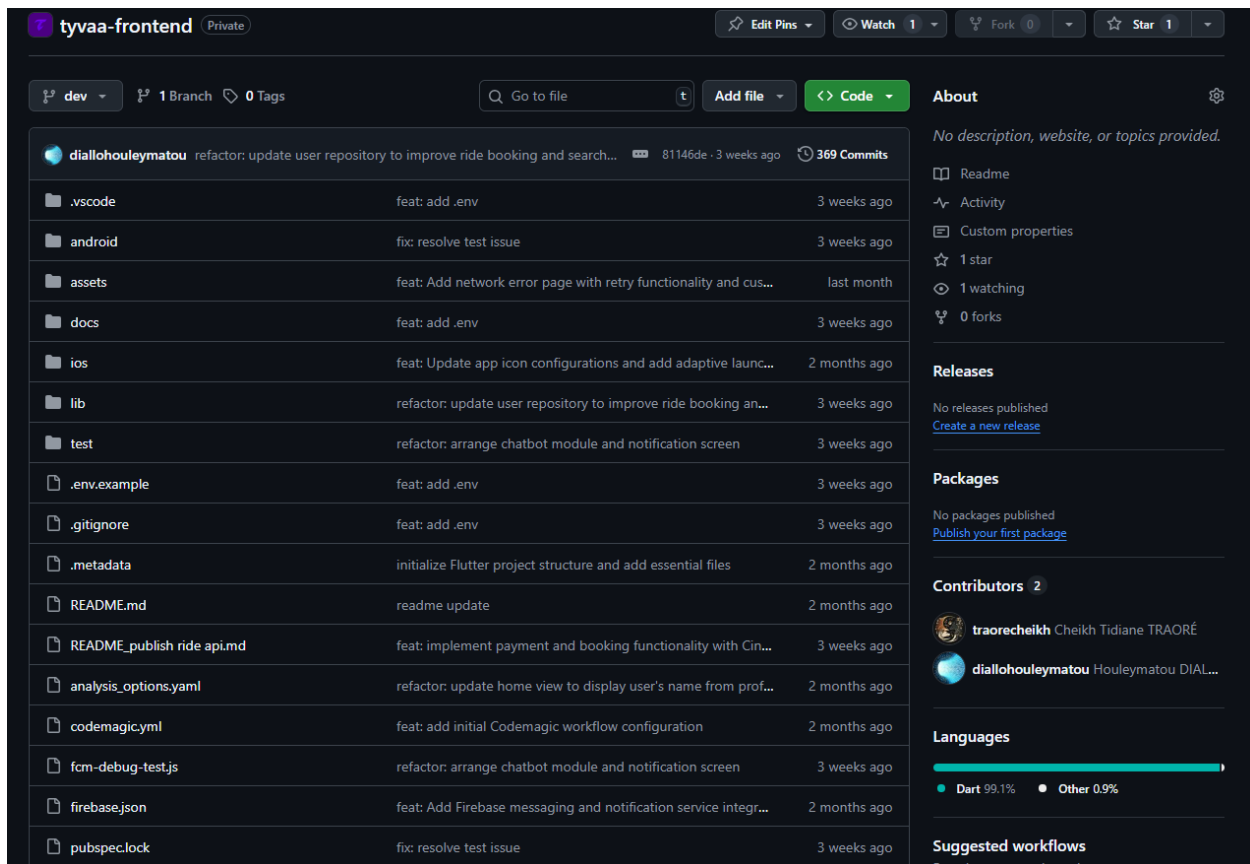


Figure 20: Dépôt Git – Application mobile Flutter

❖ GitHub Actions



Figure 21: Logo de l'outil Github Action

Mise en place de pipelines CI/CD complets : Lint, tests unitaires, analyse SonarCloud, build Docker, déploiement automatique en staging et production.

❖ One-Click Deploy

Intégration d'un pipeline de déploiement en un clic, depuis la validation Git jusqu'à la mise en production automatisée.

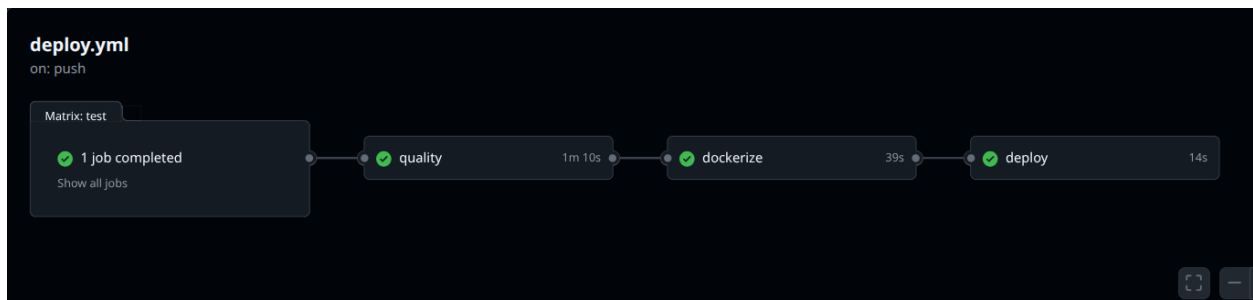


Figure 22: Pipeline CI/CD TYVAA

2.5.9 Outils de Conteneurisation et Infrastructure as Code

❖ Docker



Figure 23: Logo de l'outil Docker

L'ensemble du backend (Node.js + Fastify) et du panneau admin (Vue.js) est conteneurisé pour garantir l'uniformité entre environnements.

❖ Terraform



Figure 24: Logo de l'outil Terraform

Utilisé pour le provisioning de l'infrastructure cloud (base de données PostgreSQL, secrets, configurations, services tiers), permettant une infrastructure déclarative, traçable et versionnée.

2.5.10 Hébergement et Environnements

❖ Koyeb



Figure 25: Logo de l'outil Koyeb

Le backend Node.js est déployé sur Koyeb (cloud serverless), avec déploiement automatique, gestion SSL, montée en charge dynamique et observabilité intégrée.

❖ Azure Web Services



Figure 26: Logo de l'outil Azure

Le **panneau admin** est déployé sur Azure App Services, garantissant robustesse, scalabilité, et intégration avec les outils DevOps.

❖ Dev / Staging / Production

Trois environnements sont utilisés :

- **Dev** : pour le développement local
- **Staging** : environnement miroir de la production
- **Production** : avec monitoring actif et audit en temps réel

2.5.11 Outils de Qualité Logicielle

❖ SonarCloud



Figure 27: Logo de l'outil Sonar

Afin de garantir la qualité du code, l'ensemble du backend est analysé via **SonarCloud**, qui fournit des métriques sur la couverture des tests, la duplication, les bugs potentiels et les problèmes de sécurité.

2.5.12 Outils de Tests



Figure 28: Logo de l'outil Mocha et Jest

Le backend développé avec **Node.js (Fastify)** a été rigoureusement testé à l'aide des Framework **Jest** et **Mocha**. Ces tests couvrent à la fois les routes API, la logique métier et les cas limites (*edge cases*), avec des assertions précises et des mocks bien structurés.

Le **panneau d'administration Vue.js** a également fait l'objet de tests poussés. Des tests unitaires ont été réalisés avec **Jest**, tandis que des tests end-to-end ont été menés avec **Selenium**, afin de simuler des interactions réelles dans le navigateur et valider le comportement de l'interface.

En tout, **plus de 400 tests** ont été mis en place pour garantir la robustesse, la stabilité et la qualité du système, aussi bien sur le plan fonctionnel que sur les cas exceptionnels.

2.5.13 Outils de Sécurité, Observabilité et Résilience



Figure 29: Logo de l'outil JWT et OWASP

Pour sécuriser les échanges HTTP, les entêtes sont renforcés via **Fastify Helmet**, protégeant contre les attaques classiques comme le XSS, le clickjacking ou l'injection de scripts.

L'authentification repose sur des **JSON Web Tokens (JWT)**, permettant des sessions sécurisées et traçables, avec une gestion centralisée des rôles et permissions.

Le backend suit les **recommandations de l'OWASP**, appliquant notamment la validation stricte des entrées, la prévention des injections, et la gestion rigoureuse des erreurs.

Une logique d'**idempotence** est intégrée aux requêtes critiques (comme les POST), empêchant les traitements en double via des clés uniques générées côté client et validées côté serveur.

Les dépendances des projets (backend, frontend, mobile) sont scannées automatiquement grâce à **GitHub Dependabot**, assurant une veille continue sur les vulnérabilités connues.



Figure 30: Logo de l'outil Dependabot et Pino

L'**observabilité** est assurée à plusieurs niveaux :

- **Pino** enregistre des logs structurés pour suivre précisément les événements dans le backend.
- Un **système d'audit granulaire** journalise toutes les actions sensibles dans le panneau d'administration, garantissant la traçabilité des opérations critiques.

2.5.14 Notifications et Communication

La gestion des notifications temps réel repose sur deux technologies complémentaires :

❖ **RabbitMQ**



Figure 31: Logo de l'outil Firebase Cloud Messagin et RabbitMQ

Utilisé comme **bus de messages** pour la communication interservices. Il joue un rôle central dans l'envoi d'événements asynchrones, tels que les réservations de trajets, les validations, ou les notifications internes. Cela permet de **découpler** les services, améliorer la **scalabilité**, et garantir la **fiabilité des traitements**.

- ❖ **Firebase Cloud Messaging (FCM)** est intégré côté mobile pour l'envoi de **notifications push**. Lorsqu'un événement est déclenché (par exemple une nouvelle réservation ou un message reçu), le backend transmet le message à FCM, qui le redirige aux appareils utilisateurs, garantissant une **réception instantanée et optimisée**.

2.5.15 Nom de Domaine et Emails Professionnels



Figure 32: Logo de l'outil Name.com et Zoho

Un **achat de domaine via Name.Com** et une **configuration Zoho** a permis d'obtenir une adresse officielle @tyvaa.live, avec la mise en place d'**adresses mail professionnelles** (ex : noreply@tyvaa.live), renforçant la crédibilité du système.

Pour sécuriser et optimiser l'accès au domaine, **Cloudflare** a été intégré.

2.5.16 Intelligence Artificielle



Figure 33: Logo de l'outil Genkit

L'application TYVAA intègre **Genkit AI**, un outil d'intelligence artificielle conçu pour **assister les utilisateurs** et **répondre à leurs questions** de manière instantanée.

Ce chatbot est accessible directement dans l'application et permet aux usagers d'obtenir de l'aide sur les fonctionnalités, la réservation de trajets, ou encore les informations générales sur l'utilisation de la plateforme.

Il s'agit d'un **assistant conversationnel intelligent** qui améliore l'autonomie des utilisateurs, réduit la charge du support, et renforce l'expérience globale au sein de l'application

2.6 Présentation de la solution

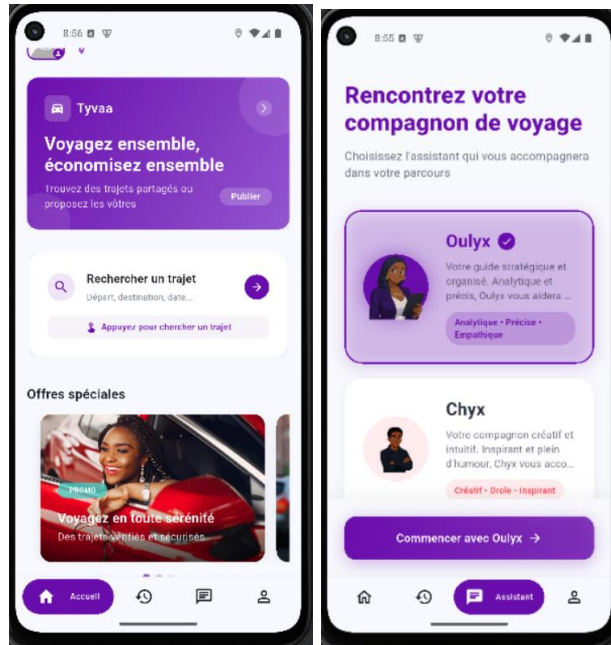


Figure 34: Écran de l'application – Page d'accueil et Page de Sélection Chatbot

- ❖ Cette interface constitue l'entrée principale pour les passagers souhaitant rechercher un trajet.
- ❖ Cette interface matérialise l'intégration du chatbot, un outil clé pour améliorer l'autonomie des utilisateurs

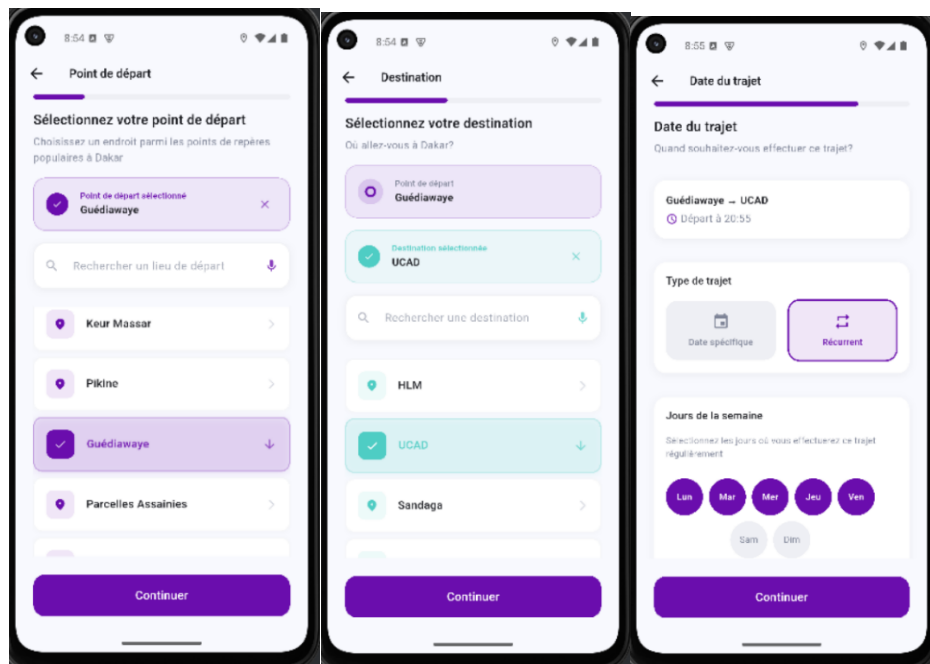
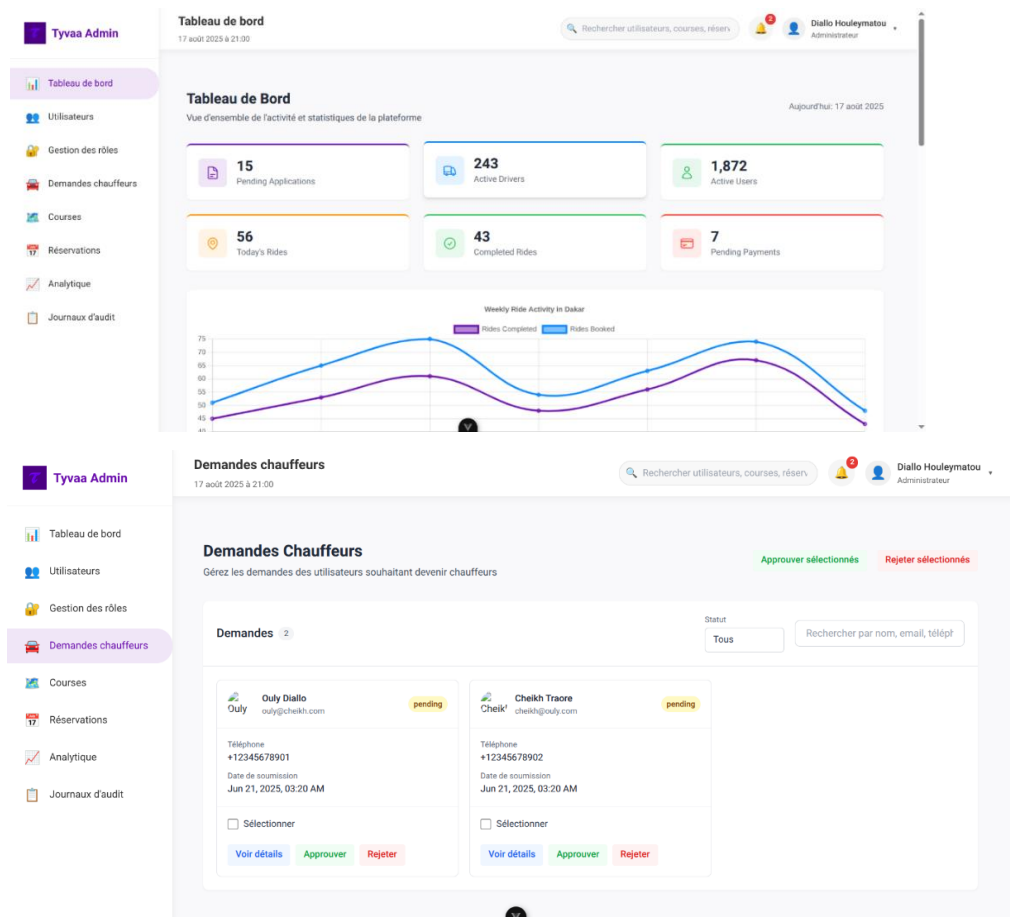


Figure 35: Écran de l'application – Page de sélection point de départ, arriv et créneau horaire

- ❖ Cette interface permet à l'utilisateur de configurer ou de corriger son point de départ. Comme pour la destination, un moteur de suggestion localisé qui facilite la sélection, renforçant ainsi la pertinence des correspondances entre conducteurs et passagers.
- ❖ Cette interface permet à l'utilisateur de définir ou de modifier sa destination finale. Le système intègre une fonction de recherche dynamique, associée à des propositions contextualisées basées sur les zones fréquentées à Dakar. Cela simplifie la saisie des adresses et optimise la compatibilité des trajets avec ceux des autres utilisateurs.
- ❖ Cette interface permet à l'utilisateur de définir les paramètres temporels de son trajet. Elle intègre deux options clés : la spécification d'une date unique ou la configuration d'un trajet répétitif (hebdomadaire), avec une sélection granulaire des jours concernés.



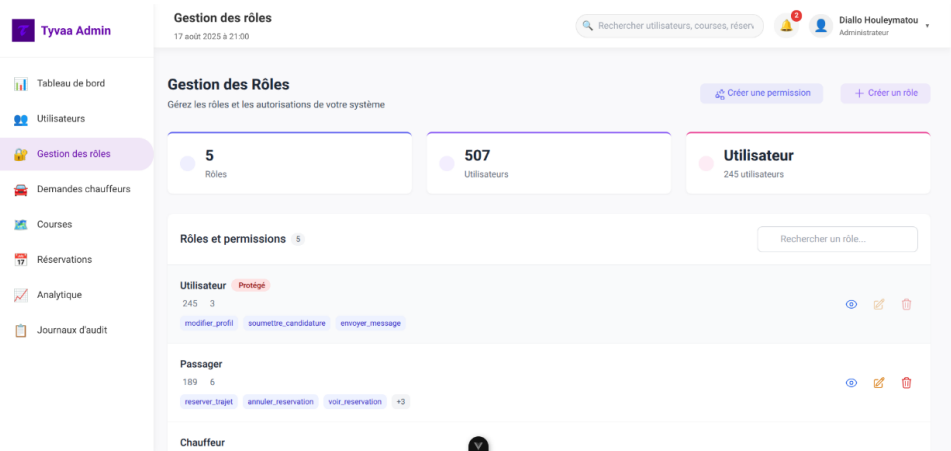


Figure 36: Écran de l'application Panneau Admin – Page d'accueil, gestion RBAC, Demande d'application

- ❖ Cette interface est un outil de supervision central pour l'administrateur. Elle utilise Vue.js pour afficher des données en temps réel, issues de la base de données
- ❖ Cette interface incarne la logique de contrôle d'accès basée sur les rôles (RBAC), essentielle pour la sécurité et la gouvernance de la plateforme
- ❖ Cette fonctionnalité illustre la gestion des rôles (RBAC) et la sécurité renforcée du système. Les administrateurs peuvent valider les profils conducteurs après vérification des documents (permis, véhicule).

Chapitre 3 : BILAN

3.1 Objectifs Atteints

❖ **Module Utilisateurs :**

- La création, l'authentification et la gestion des profils utilisateurs (conducteur/passager) ont été mises en place avec gestion rigoureuse des rôles et permissions.
- La protection des données personnelles est assurée via JWT, OWASP et dépendances scannées (GitHub Dependabot).

❖ **Module Trajets :**

- Les conducteurs peuvent publier, modifier et supprimer des trajets avec gestion complète des informations clés (lieux, date, heure, places, prix, commentaires).
- La logique d'idempotence empêche les duplications lors des publications.

❖ **Module Réservations :**

- Les passagers peuvent rechercher et réserver des trajets, avec synchronisation en temps réel des disponibilités.
- L'annulation et la mise à jour des réservations sont fonctionnelles.

❖ **Module Paiement sécurisé :**

- L'intégration de l'API **CinetPay** permet de gérer les transactions avec traçabilité.
- Le système de gestion des paiements est opérationnel pour les cas standards.

❖ **Module Chatbot intelligent :**

- Un chatbot basé sur **Genkit AI** est intégré pour assister les utilisateurs dans la recherche de trajets et répondre à leurs questions fréquentes.

❖ **Module Suivi et Monitoring Administrateur :**

- Le panneau admin intègre des outils de suivi avec **audit granulaire** et **RBAC finement défini**.
- Les statistiques en temps réel sont visibles via **Chart.js**.

❖ **Sécurité et performance globale :**

- Les échanges sont sécurisés (Fastify Helmet, JWT, validation Zod).
- Le backend est conteneurisé, optimisé pour la montée en charge, et monitoré via Koyeb/Cloudflare.

❖ **Modularité et évolutivité :**

- L'architecture est basée sur un **Domain Driven Design** modulaire pour le backend et un **Feature-Sliced Design** pour le panneau admin.
- Les modules sont conçus pour permettre l'intégration future de fonctionnalités avancées (géolocalisation, services tiers).

3.2 Objectifs Non Atteints

L'ensemble des objectifs fixés en début de projet a été atteint avec succès. Tous les modules prévus (utilisateurs, trajets, réservations, paiements, chatbot, monitoring admin, sécurité, et modularité technique) ont été conçus, développés et intégrés conformément au cahier des charges.

Chaque fonctionnalité a été implémentée en respectant les standards de qualité, de sécurité et de performance visés, notamment à travers :

- ❖ Des tests unitaires rigoureux (> 400 tests, 97% de couverture),
- ❖ Une architecture modulaire évolutive (Domain Driven Design),
- ❖ Des processus CI/CD automatisés (One-Click Deploy),
- ❖ Une infrastructure cloud sécurisée et scalable (Koyeb, Azure, Cloudflare).

Ce résultat témoigne d'une gestion rigoureuse du périmètre projet, ainsi que d'une planification réaliste des objectifs fonctionnels et techniques.

Perspectives d'Amélioration (au-delà du périmètre initial)

Bien que les objectifs définis aient été atteints, certaines fonctionnalités avancées, non incluses dans la version initiale, sont identifiées pour des évolutions futures :

- ❖ Géolocalisation en temps réel des trajets,
- ❖ Intelligence Artificielle contextuelle (chatbot prédictif),
- ❖ Automatisation de la gestion des trajets récurrents complexes.

3.3 Intérêts Personnels

Ce projet a représenté pour nous bien plus qu'un simple travail académique : il a été une véritable expérience humaine et professionnelle. Travailler ensemble sur la conception et le développement

de l'application TYVAA nous a permis de renforcer des compétences techniques, mais surtout d'acquérir des qualités essentielles telles que la rigueur, la patience et l'esprit d'équipe.

Tout au long du projet, nous avons appris à **collaborer de manière efficace et complémentaire**, en partageant les responsabilités selon nos forces respectives, tout en restant solidaires face aux difficultés. Cette dynamique nous a permis de développer une réelle capacité à dialoguer, à confronter nos idées dans un cadre constructif et à trouver des compromis qui enrichissaient la qualité finale de notre travail.

Au-delà des aspects techniques, cette expérience nous a aussi sensibilisés à l'importance de concevoir des solutions adaptées aux réalités locales, en restant à l'écoute des besoins des futurs utilisateurs. Nous avons ainsi pris conscience que la réussite d'un projet technologique ne dépend pas uniquement des compétences en programmation, mais aussi de la capacité à comprendre et anticiper les usages.

Bibliographie

- **Sakho, P., Chenal, J., & Kemajou, A. (2017).** *Villes africaines : Mobilités et transports urbains*. BOOCs EPFL Press—eBook.
- **Dziwonou, Y. (2016).** *Les difficultés de mobilité dans les villes d’Afrique subsaharienne*. Éditions universitaires européennes.
- **Alla Manga, A., Fall, M. B., Sagna, N. S., & Diaw, A. T. (2019).** *Développement des projets de mobilité urbaine et impacts socio-spatiaux au Sénégal : cas de l’autoroute à péage et du Bus Rapid Transit (BRT)*. In Actes du 22^e colloque du SIFEE, Cotonou.
- **Mike Isaac (2019) –** *Super-Pumped: The Battle for Uber*. W. W. Norton & Co.
- **Paris Marx (2022) –** *Road to Nowhere: What Silicon Valley Gets Wrong about the Future of Transportation*. Verso Books.
- **Karel Martens (2016) –** *Transport Justice: Designing Fair Transportation Systems*. Routledge.
- **Martin Kleppmann (2017) –** *Designing Data-Intensive Applications*. O’Reilly Media.
- **Alex Xu (2020) –** *System Design Interview: An Insider’s Guide* (Vol. 1 & 2). ByteByteGo.
- **Brendan Burns (2018) –** *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services*. O’Reilly Media.

Webographie

[Medium – Uber Case Study.](#) - Consulté le 27/04/2025.

[Medium – Hit the Road to Success: A Guide to Building an Uber-Like App.](#) - Consulté le 27/04/2025.

[Climate Chance – YOBBALEMA – Pitch « Écomobilité » \(PDF\).](#) - Consulté le 27/04/2025.

[Medium – Uber Engineering: An Overview.](#) - Consulté le 27/04/2025.

[Medium – L’essor de BlaBlaCar : Révolution du covoiturage.](#) - Consulté le 27/04/2025.

[Medium – The Untold Story of Ridesharing.](#) - Consulté le 27/04/2025.

[HAL Thèses – Mobilités et politiques publiques de transport à Dakar.](#) - Consulté le 27/04/2025.

[Revelations Médias – « ALLÔ DAKAR » sous une autre forme : l’application Yobbalema.](#) - Consulté le 09/05/2025.

[LinkedIn – TukkiJamm – Société de covoiturage domicile-travail.](#) - Consulté le 21/05/2025.

[YouTube – #MediUP : « Tukki Jamm » ou le covoiturage autrement \(vidéo\).](#) - Consulté le 04/06/2025.

[IIETA – Paratransit Reform and Quality of Services in West Africa Cities \(Dakar, Bamako, Conakry\).](#) - Consulté le 10/06/2025.

[arXiv – Multimodal Urban Transportation Network Equilibrium Including Intermodality and Shared Mobility Services.](#) - Consulté le 25/07/2025.

TABLE DE MATIERE

Dédicaces.....	II
Remerciements.....	II
Liste des Figures	III
Liste des Tableaux	V
Glossaire	VI
Sommaire.....	VIII
Résumé	IX
Abstract	X
CHAPITRE 1 : INTRODUCTION GENERALE	1
1.1 Présentation de la structure d'accueil	1
1.2 Contexte	2
1.3 Problématique.....	2
1.4 Objectifs du mémoire.....	3
1.4.1 Objectif générale.....	3
1.4.2 Objectifs spécifiques	3
CHAPITRE 2 : TRAVAUX REALISES AU COURS DU STAGE.....	5
2.1 Gestion Projet Agile	5
2.1.1 Méthode Kanban.....	6
2.1.2 Outil Trello	6
2.2 Spécialisations Fonctionnelles	7
2.2.1 Les Besoins Fonctionnelles	7
2.2.2 Les Besoins Non Fonctionnelles.....	8

2.3	Méthodes D'Analyse et de Conception	9
2.3.1	Méthode UML	9
2.3.2	Diagramme de contexte.....	9
2.3.3	Diagrammes de cas d'utilisation.....	10
2.3.4	Diagramme de classe	13
2.4	Maquettisation de la solution	14
2.5	Réalisation.....	16
2.5.1	Architecture	16
2.5.2	Technologies et Langages Utilisés.....	17
2.5.3	Outils Utilisés	18
2.5.4	Outils de de modélisation	18
2.5.5	Outils de prototypage	18
2.5.6	Outils de productivité.....	19
2.5.7	Outils de développement.....	19
2.5.8	Outils de Gestion du Code et CI/CD.....	20
2.5.9	Outils de Conteneurisation et Infrastructure as Code	23
2.5.10	Hébergement et Environnements.....	23
2.5.11	Outils de Qualité Logicielle.....	24
2.5.12	Outils de Tests	25
2.5.13	Outils de Sécurité, Observabilité et Résilience	25
2.5.14	Notifications et Communication.....	26
2.5.15	Nom de Domaine et Emails Professionnels	27
2.5.16	Intelligence Artificielle	27
2.6	Présentation de la solution	27
CHAPITRE 3 :	BILAN.....	31

3.1	Objectifs Atteints.....	31
3.2	Objectifs Non Atteints	32
3.3	Intérêts Personnels	32
	Bibliographie	i
	Webographie.....	ii
	TABLE DE MATIERE	iii