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



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

**MOOV**

par

Solofonaina Mendrika Fitahiana RAMILISON

Mémoire présenté  
MASTER de SCIENCES, TECHNOLOGIES, SANTE, mention INFORMATIQUE  
Spécialité MOBIQUITE, BASES DE DONNEES ET INTEGRATION DE SYSTÈMES (MBDS)

Jury :

© Solofonaina Mendrika Fitahiana RAMILISON

Résumé

Dans le cadre de ses activités, notre entreprise envisage la mise en place d'une application mobile de gestion de VTC, considérée comme une solution essentielle pour optimiser les services de transport à la demande et améliorer l'expérience utilisateur. L'objectif ultime est de créer une plateforme complète permettant la réservation, le suivi en temps réel et la gestion efficace des courses.

L'application envisagée permettra de centraliser et de coordonner toutes les interactions entre passagers, chauffeurs et administrateurs dans une interface unique. Cette centralisation vise principalement à fournir un service fluide et à faciliter la prise de décision pour tous les acteurs impliqués.

Le projet vise à intégrer divers types de fonctionnalités, notamment l'authentification sécurisée, la géolocalisation, la réservation de courses, le paiement en ligne, et un système de notation bidirectionnel, dans une application mobile unifiée.

L'un des objectifs clés de l'application est de générer automatiquement des mises en relation entre passagers et chauffeurs, tout en proposant un outil de suivi en temps réel simple et flexible pour les courses en cours.

Enfin, le projet s'attachera également à garantir la sécurité des transactions, la protection des données personnelles et la traçabilité des courses, afin de renforcer la fiabilité et la confiance dans le service proposé.

**Mots-clés** : Application mobile, VTC, Géolocalisation, Paiement en ligne, Système de notation bidirectionnel, Authentification à deux facteurs

Abstract

As part of its activities, our company is planning to implement a mobile application for managing VTC services, seen as an essential solution to optimize on-demand transport services and improve the user experience. The ultimate goal is to create a comprehensive platform that allows for booking, real-time tracking, and efficient management of rides.

The envisioned application will centralize and coordinate all interactions between passengers, drivers, and administrators in a single interface. This centralization primarily aims to provide a smooth service and facilitate decision-making for all stakeholders involved.

The project aims to integrate various features, including secure authentication, geolocation, ride booking, online payment, and a bidirectional rating system, into a unified mobile application.

One of the key objectives of the application is to automatically match passengers with drivers while offering a simple and flexible real-time tracking tool for ongoing rides.

Finally, the project will focus on ensuring the security of transactions, the protection of personal data, and the traceability of rides, in order to enhance the reliability and trust in the service provided.

Keywords: Mobile application, VTC, Geolocation, Online payment, Bidirectional rating system, Two-factor authentication

**Table des matières**

[**Liste des tableaux 6**](#_heading=h.m8hw83klfnfr)

[**Liste des figures 7**](#_heading=h.1fob9te)

[**Liste des acronymes 8**](#_heading=h.yi50v5b0x8wu)

[**Glossaire 9**](#_heading=h.fgelw2stxwyu)

[**Introduction 10**](#_heading=h.tyjcwt)

[**1 Présentation du stage 11**](#_heading=h.3dy6vkm)

[1.1 Présentation de l'entreprise 11](#_heading=h.1t3h5sf)

[1.2 Présentation du sujet et objectifs du projet (problème traité) 11](#_heading=h.4d34og8)

[**2 État de l’art sur le sujet traité 13**](#_heading=h.17dp8vu)

[2.1 Critères de comparaison 14](#_heading=h.3rdcrjn)

[2.2 Etude de chaque solution au vu des critères 14](#_heading=h.26in1rg)

[2.2.1 Uber 14](#_heading=h.wyta4oy4guvo)

[2.2.2 Yango 14](#_heading=h.ip8f28qxwoa7)

[2.2.3 Bolt (Taxify) 15](#_heading=h.u0nmbqy9x7xj)

[2.2.4 Misy 15](#_heading=h.owqm5bonxutk)

[2.3 Tableau comparatif des solutions au vu des critères 16](#_heading=h.lnxbz9)

[**3 Etude de l’existant et solution envisagée 17**](#_heading=h.44sinio)

[3.1 Étude de l’existant 17](#_heading=h.2jxsxqh)

[3.1.1 Description externe du système logiciel existant (vision utilisateur) 17](#_heading=h.z337ya)

[3.1.2 Description interne du système logiciel existant (vision développeur/conception) 17](#_heading=h.3j2qqm3)

[3.2 Critique de l’existant 17](#_heading=h.1y810tw)

[3.3 Solutions envisagées 18](#_heading=h.4i7ojhp)

[3.4 Objectifs principaux et livrables 18](#_heading=h.2xcytpi)

[**4 Démarche projet 20**](#_heading=h.3whwml4)

[4.1 Principes de la démarche projet 20](#_heading=h.2bn6wsx)

[4.1.1 Activités d’ingénierie logicielle 20](#_heading=h.qsh70q)

[4.1.2 Méthode de gestion de projet utilisée 20](#_heading=h.3as4poj)

[4.1.3 Rôles et responsabilités 20](#_heading=h.1pxezwc)

[4.1.4 Outils 21](#_heading=h.49x2ik5)

[4.1.5 Gestion de la configuration 22](#_heading=h.2p2csry)

[4.2 Contraintes et risques sur le projet 22](#_heading=h.147n2zr)

[4.3 Démarche projet mise en œuvre 22](#_heading=h.3o7alnk)

[4.4 Budget du projet 23](#_heading=h.ihv636)

[**5 Exigences réalisées dans le projet (vision externe/utilisateur) 24**](#_heading=h.1hmsyys)

[5.1 Exigences fonctionnelles – Cas d’utilisation 24](#_heading=h.41mghml)

[5.1.1 Cas d’utilisation 1 - Réservation d’une course 24](#_heading=h.2grqrue)

[5.1.2 Cas d’utilisation 2 - Suivi en temps réel de la course 26](#_heading=h.vx1227)

[5.1.3 Cas d’utilisation 3 - Paiement en ligne 27](#_heading=h.3fwokq0)

[5.2 Exigences non fonctionnelles transverses 29](#_heading=h.1v1yuxt)

[5.3 Interfaces détaillées 29](#_heading=h.4f1mdlm)

[5.3.1 IHM 29](#_heading=h.2u6wntf)

[5.3.2 Interfaces avec d’autres systèmes 29](#_heading=h.19c6y18)

[**6 Architecture(s) système 30**](#_heading=h.28h4qwu)

[6.1 Architecture logicielle 30](#_heading=h.hd4gelvdi1mc)

[6.2 Architecture technique 30](#_heading=h.e787hdo4c5qo)

[**7 Conception du système logiciel réalisée dans le projet (vision interne/développeur) 31**](#_heading=h.1mrcu09)

[7.1 Plate-forme technique 31](#_heading=h.46r0co2)

[7.1.1 Architecture matérielle 31](#_heading=h.52rt3gziijj6)

[7.1.2 Framework Logiciel 31](#_heading=h.7i4napcwtmmb)

[7.2 Conception du logiciel développé 32](#_heading=h.2lwamvv)

[7.2.1 Conception du code source 32](#_heading=h.111kx3o)

[7.2.2 Le code source – vue statique 33](#_heading=h.3l18frh)

[7.2.3 Modélisation de données 34](#_heading=h.206ipza)

[7.2.4 Les composants et leur déploiement 34](#_heading=h.2zbgiuw)

[**8 Tests du système logiciel 35**](#_heading=h.3ygebqi)

[**9 Conclusion générale 36**](#_heading=h.2dlolyb)

[**10 Références et Bibliographie 37**](#_heading=h.3cqmetx)

[**11 Annexes 38**](#_heading=h.1rvwp1q)

# Liste des tableaux

Tableau 1 : Tableau comparatif des solutions au vu des critères ............................................ 15

Tableau 2 : Choix des outils utilisés …………………………................................................ 21

Tableau 3 : Contraintes et risques …………………………................................................... 22

Tableau 4 : Liste des cas d’utilisation …………………………............................................. 24

# Liste des figures

Figure 1 : Diagramme de Gantt ………………………………............................................... 23

Figure 2 : Cas d’utilisation 1 - Réservation d’une course ………........................................... 25

Figure 3 : Cas d’utilisation 2 - Suivi en temps réel de la course ............................................. 27

Figure 4 : Cas d’utilisation 3 - Paiement en ligne ……………............................................... 28

Figure 5 : Architecture logicielle ……………………………............................................... 30

Figure 6 : Architecture technique ……………………………............................................... 30

Figure 7 : Architecture matérielle …………………………….............................................. 31

Figure 8 : Vue statique frontend (Ionic/React) ……………………………........................... 33

Figure 9 : Vue statique backend (Node.js/Express) ……………………………................... 33

# Liste des acronymes

**API :** Application Programming Interface

**IDE :** Integrated Development Environment

**VTC:** Véhicules de transport avec chauffeurs

# Glossaire

**API :** Une API est un ensemble de protocoles qui permet à un service de communiquer avec d’autres services et qui facilite l’intégration de nouveaux composants dans une architecture existante.

**IDE:** Un environnement de développement intégré (IDE) est une application logicielle qui aide les programmeurs à développer efficacement le code logiciel.

# Introduction

À Madagascar, le transport public et privé souffre de plusieurs défis : bus surchargés, taxis peu fiables, et absence de solutions numériques pour organiser les déplacements. Une étude montre que près de 70 % des Malgaches préfèrent les bus, malgré leur manque de confort, principalement à cause du coût élevé des taxis. Le besoin d’une alternative fiable, confortable, et abordable devient de plus en plus pressant.

Cette application mobile VTC dédiée aux taxis répond à cette demande en offrant une solution de transport mieux structurée et accessible via une interface simple. Elle permet aux utilisateurs de réserver rapidement un véhicule, suivre leur trajet en temps réel et effectuer des paiements sécurisés.

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

Ces objectifs et missions visent à poser la question suivante : Comment offrir une solution numérique pour la gestion de véhicules de transport avec chauffeur (VTC) qui soit à la fois fiable, abordable, et adaptée aux besoins spécifiques des utilisateurs à Madagascar ?

Afin de répondre à cette question, ce document détaillera dans les prochains chapitres, en premier lieu l’état de l’art du projet, puis l’analyse de l’existant, suivi des démarches et approches dans la réalisation du projet. Ensuite, détailler les exigences réalisées dans le projet et enfin finir par la conception du projet.

# Présentation du stage

## Présentation de l'entreprise

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

Elle dispose aujourd’hui de six bureaux, répartis dans cinq pays (France, Madagascar, Italie, Philippines, et Etats-Unis) composées d’une équipe de 120 professionnels passionnés. Les équipes sont organisées autour de plusieurs pôles technologiques (Développements Web et Mobile, Développements AR/VR[[1]](#footnote-0), Blockchain, Département IoT[[2]](#footnote-1)/Live Streaming).

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

## Présentation du sujet et objectifs du projet (problème traité)

Le projet de développement d'une application mobile de gestion de VTC répond à un besoin urgent de modernisation et d'amélioration du transport urbain dans le pays. Face aux défis actuels tels que le manque de confort, la désorganisation des transports en commun et les coûts élevés des taxis traditionnels, notre solution vise à révolutionner la mobilité urbaine.

L'aspect innovant de ce projet réside dans l'introduction d'une plateforme numérique intégrée dans un marché où les solutions technologiques pour le transport sont rares. Cette application apporte une approche novatrice en combinant la commodité des VTC avec des tarifs abordables, adaptés au contexte économique local.

Dans le contexte malgache, où le transport est souvent synonyme de difficultés quotidiennes, cette application représente un enjeu majeur pour améliorer la qualité de vie des citoyens. Elle vise à offrir une alternative fiable et confortable aux bus surpeuplés et aux taxis peu entretenus, tout en restant accessibles financièrement.

La criticité du projet est élevée car il touche à un besoin fondamental de la population : se déplacer efficacement et en toute sécurité. Les principaux enjeux incluent l'amélioration de l'expérience utilisateur dans les transports, la création d'opportunités d'emploi pour les chauffeurs locaux, et la contribution à la modernisation du secteur des transports à Madagascar.

Les risques majeurs comprennent la résistance potentielle des acteurs traditionnels du transport, les défis liés à l'adoption technologique dans un marché peu numérisé, et la nécessité d'assurer une sécurité optimale pour les utilisateurs et les chauffeurs. De plus, l'adaptation du modèle économique au pouvoir d'achat local tout en maintenant un service de qualité représente un défi crucial.

En somme, ce projet d'application VTC n'est pas seulement une innovation technologique, mais une transformation profonde de la mobilité urbaine à Madagascar, posant les bases d'un système de transport plus efficace, sûr et adapté aux besoins locaux.

# État de l’art sur le sujet traité

Le marché du transport à Madagascar n’est pas encore pleinement digitalisé, en particulier pour ce qui concerne les systèmes de gestion des véhicules de transport avec chauffeurs (VTC). Actuellement, les usagers se tournent principalement vers des solutions de transport public telles que les bus ou les taxi-brousses, mais celles-ci sont souvent inefficaces en termes de confort et d’organisation. Les alternatives comme les taxis et moto-taxis, bien que existantes, sont généralement perçues comme coûteuses et peu fiables. Il existe donc un besoin non satisfait pour une solution de transport moderne, efficace, abordable et confortable à Madagascar. À ce titre, les applications VTC pourraient représenter une solution de choix.

## Critères de comparaison

Pour mener à bien cette étude comparative des solutions VTC existantes, nous retiendrons les critères suivants :

* **Accessibilité et facilité d’utilisation** : capacité de l’application à être simple à utiliser, surtout dans des contextes où l’adoption numérique n'est pas encore généralisée.
* **Confort et fiabilité du service** : niveau de confort et de fiabilité perçu par les utilisateurs pour les véhicules, ainsi que l’état de ceux-ci.
* **Tarification et rapport qualité/prix** : accessibilité des prix en comparaison avec d’autres moyens de transport locaux (bus, taxis traditionnels).
* **Technologie et innovation** : utilisation des technologies modernes (géolocalisation, paiements en ligne, suivi en temps réel) pour améliorer l'expérience utilisateur.
* **Impact sur la société et l'économie locale** : contribution de la solution à l’amélioration de la qualité de vie et au développement numérique.

En considérant ces critères de comparaison adaptés à ce projet, nous sommes en mesure de mieux évaluer les solutions disponibles sur le marché et de choisir celle qui répond le mieux aux besoins spécifiques de la clientèle.

## Etude de chaque solution au vu des critères

### 2.2.1 Uber

Uber, une référence mondiale, propose une plateforme très accessible, mais elle n'est pas encore disponible à Madagascar. Elle offre un service fiable avec des technologies modernes comme la géolocalisation et les paiements en ligne. Cependant, le coût d’utilisation d’Uber serait élevé pour le marché malgache.

### 2.2.2 Yango

Yango, présent dans plusieurs pays africains, n’est pas encore implanté à Madagascar. L’application se concentre sur l'accessibilité et des prix plus abordables que Uber. Comme Uber, Yango offre un service technologique avancé avec une bonne gestion de la tarification, mais son absence à Madagascar limite son influence dans le contexte local.

### 2.2.3 Bolt (Taxify)

Bolt est une autre application VTC opérant principalement dans plusieurs pays africains. Elle est perçue comme plus abordable que Uber et met l'accent sur la simplicité d’utilisation. Cependant, à Madagascar, Bolt n’est pas encore implanté, ce qui limite sa pertinence locale. Sur le plan technologique, Bolt propose une solution similaire à Uber avec géolocalisation et suivi des chauffeurs en temps réel, tout en offrant des prix plus bas, ce qui pourrait convenir au marché malgache si elle y était déployée.

### 2.2.4 Misy

**Misy** est une application locale conçue pour Madagascar, offrant une solution numérique pour la réservation de taxis et de VTC. Misy se distingue par son adaptation aux besoins du marché local, notamment en termes de tarification et d’accessibilité. En intégrant des fonctionnalités comme la géolocalisation et la réservation via mobile, Misy répond aux contraintes du transport à Madagascar. De plus, l’application met en avant une tarification adaptée au pouvoir d’achat des utilisateurs locaux, ce qui en fait un choix pertinent pour un marché malgache encore en phase de digitalisation.

## Tableau comparatif des solutions au vu des critères

| **Critères de comparaison** | **Uber** | **Yango** | **Bolt** | **Misy** |
| --- | --- | --- | --- | --- |
| **Accessibilité et facilité d’utilisation** | Très simple à utiliser | Facile à utiliser | Simple à utiliser | Adapté au marché malgache |
| **Confort et fiabilité du service** | Haute fiabilité et confort | Fiabilité moyenne | Fiabilité et confort modérés | Fiabilité adaptée au marché local |
| **Tarification et rapport qualité/prix** | Coût plus élevé, mais de qualité | Abordable | Moins cher que Uber | Tarification adaptée au contexte malgache |
| **Technologie et innovation** | Géolocalisation, paiements en ligne | Géolocalisation, paiement en ligne | Géolocalisation, suivi en temps réel | Géolocalisation, suivi en temps réel |
| **Impact sur la société et l'économie locale** | Transformation numérique globale | Appropriée pour les pays émergents | Adaptée aux marchés africains | Contribution directe à la modernisation du transport à Madagascar |

Tableau 1 : Tableau comparatif des solutions au vu des critères

# Etude de l’existant et solution envisagée

## Étude de l’existant

### Description externe du système logiciel existant (vision utilisateur)

Le système de transport existant à Madagascar, particulièrement pour les taxis, repose essentiellement sur des interactions directes entre les clients et les chauffeurs, sans intégration de solutions numériques pour la gestion des réservations ou du suivi en temps réel. Les utilisateurs doivent généralement appeler directement les chauffeurs ou attendre dans la rue, ce qui conduit à des pertes de temps et un manque de visibilité sur la disponibilité des véhicules. Il n'existe actuellement pas d'application dédiée qui centralise ces services, ce qui complique l’expérience utilisateur.

### Description interne du système logiciel existant (vision développeur/conception)

D’un point de vue technique, la gestion des réservations et des courses est principalement manuelle, ce qui limite l'efficacité des opérations. Aucune infrastructure logicielle ou base de données centralisée n'est en place pour automatiser la gestion des véhicules, des conducteurs, ou des clients. Les outils de communication utilisés sont généralement basiques, comme les appels téléphoniques ou les SMS, sans intégration de technologies modernes telles que les notifications push ou la géolocalisation en temps réel. De plus, aucune plateforme technique ne permet de gérer les paiements en ligne, créant une friction supplémentaire pour les utilisateurs.

## Critique de l’existant

Le système actuel présente certains avantages, comme la simplicité d'interaction directe entre les clients et les chauffeurs, ce qui peut convenir à ceux qui préfèrent un mode de communication traditionnel. Cependant, cette méthode est inefficace à grande échelle et ne répond pas aux attentes des utilisateurs modernes qui souhaitent un service plus rapide et structuré.

Les principales faiblesses du système existant incluent :

* L'absence d'un système de réservation centralisé, ce qui rend la gestion des courses chaotique.
* L'absence de suivi en temps réel prive les utilisateurs d'une visibilité sur l'arrivée des taxis.
* Le manque d'options de paiement numérique, limitant les modes de transaction disponibles.
* L'absence de retour utilisateur ou de système de notation pour améliorer la qualité du service.

## Solutions envisagées

La solution proposée consiste à développer une application mobile dédiée qui centralisera tous les services liés aux taxis VTC. Cette application offrira :

* Un système de réservation en ligne permettant aux utilisateurs de planifier et de suivre leurs courses en temps réel.
* L'intégration de la géolocalisation, qui permettra aux clients de visualiser les véhicules à proximité et aux chauffeurs d'optimiser leurs trajets.
* Un module de paiement numérique sécurisé, facilitant les transactions entre les clients et les chauffeurs.
* Un système de notation et de retour d’expérience, permettant aux utilisateurs d’évaluer la qualité du service.

Ces solutions permettent d'améliorer la fluidité du service, de renforcer la transparence et d'accroître la satisfaction des utilisateurs. La solution choisie a été justifiée par son potentiel à répondre aux attentes des utilisateurs en matière de simplicité d'utilisation, de rapidité et de sécurité.

## Objectifs principaux et livrables

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

* Automatise la gestion des réservations et des courses.
* Permet aux utilisateurs de suivre les véhicules en temps réel.
* Intègre un système de paiement en ligne sécurisé.
* Fournit un service fiable et structuré, tout en offrant une interface utilisateur intuitive.

Les livrables prévus sont :

* Une application mobile fonctionnelle pour Android et iOS.
* Un site web administrative pour la gestion des données et des services de l’application
* Un tableau de bord pour la gestion des conducteurs et des courses.
* Un système de gestion des paiements et des retours utilisateurs.

# Démarche projet

## Principes de la démarche projet

### Activités d’ingénierie logicielle

Le projet suit une méthodologie de gestion logicielle structurée en utilisant principalement la **méthode MERISE** pour la modélisation des données et **UML** pour les diagrammes fonctionnels et d'interaction. Les activités d’ingénierie logicielle incluent : la spécification des **exigences**, la **conception du modèle de données**, le **développement du code** et la réalisation de tests. La stratégie de tests repose sur des **tests de bout en bout (End-to-End)**, visant à valider l’intégration complète du système, et à garantir que toutes les fonctionnalités sont testées dans leur ensemble, depuis l’interface utilisateur jusqu’à la base de données.

### Méthode de gestion de projet utilisée

La méthode itérative MERISE est utilisée pour la modélisation des données et UML pour la conception fonctionnelle. La communication se fait par des comptes rendus hebdomadaires, assurant une bonne coordination des tâches. Le cahier des charges a été défini et détaillé avant le développement, pour que l’on puisse nous concentrer sur un objectif sans vouloir changer ce qui a été prévu d’être implémenté.

### Rôles et responsabilités

Les acteurs concernés par ce projet sont les suivants :

* Le chef de projet, Hariniaina RAJAONARIFETRA : acteur dans la présentation du projet aux prospects, la spécification des besoins, analyse de l’existant et les tests
* L’équipe de développement, y inclus moi, Solofoniaina Mendrika Fitahiana RAMILISON : chargé de l’analyse, de la conception, le développement, et le déploiement du projet.

### Outils

Durant l’étude, le développement du projet, nous avons utilisé les outils ci-après :

| Domaines | Outils | Description |
| --- | --- | --- |
| Modélisation UML | LucidChart | un outil en ligne de création de diagrammes et de visualisations, permettant de concevoir des schémas comme des organigrammes, des diagrammes UML, ou des cartes conceptuelles. |
| Conception MCD | DBSchema | un outil permettant de structurer les bases de données et de générer les schémas relationnels. |
| IDE | Visual Studio Code | IDE très flexible qui a servi à développer le back-end Express et le front-end web React ainsi que l’application mobile Ionic React. Il prend en charge le Javascript en tant que langage de programmation principal, et fournit des outils pour la mise en évidence des syntaxes et de débogages pour faciliter le développement. |
| Environnement de travail pour le test des APIs | Postman | Plateforme de construction et d’utilisation d’API |
| Gestion des versions | GitHub | plateforme de gestion de version collaborative qui permet de suivre l'historique des codes sources du projet et de les sauvegarder. Il facilite le travail d'équipe en permettant de gérer les modifications et les branches de développement de manière efficace. |

Tableau 2 : Choix des outils utilisés

### Gestion de la configuration

Notre équipe utilise l'outil Git pour gérer les différentes versions du projet au fur et à mesure de son avancement. Le dépôt est hébergé dans GitHub, où les versions du projet sont stockées dans le Cloud. Le processus de gestion des versions implique la création de branches spécifiques pour les environnements tels que la branche develop et la branche main. Pendant la phase de développement, chaque branche est basée sur des tâches spécifiques. Une fois le développement terminé, le responsable fusionne la branche du tâche dans la branche de développement avant de la livrer en main. Lorsque les branches sont livrées dans des environnements spécifiques, nous procédons au déploiement pour effectuer des tests.

## Contraintes et risques sur le projet

| **N°** | **Libellé du risque** | **Importance** | **Facteur contribuant** | **Solutions proposées** | **Status** |
| --- | --- | --- | --- | --- | --- |
| 1 | Retard dans la livraison du projet | Haute | - Complexité de la fonctionnalité de géolocalisation en temps réel  - Complexité de la fonctionnalité de la notification push | - Effectuer une étude approfondie des sources de données avant de commencer l'intégration.  - Prioriser les fonctionnalités critiques dans les premières itérations | En cours |
| 2 | Délai non respecté pour le développement complet | Haute | Mauvaise estimation de la charge de travail | Réaliser des points réguliers pour ajuster le planning | En cours |

Tableau 3 : Contraintes et risques

## Démarche projet mise en œuvre

Pour la mise en oeuvre du projet, nous avons défini ces grandes étapes :

* Phase de conception :

Cette partie consiste à étudier les demandes aux niveaux des utilisateurs finaux. Après la synthétisation des besoins des utilisateurs, l’équipe passe à la conception des scénarios, les spécifications techniques, les tables à utiliser, les dessins d’écran.

* Phase de développement :

Cette étape a pour but de développer l’application. L’équipe procède donc à la réalisation des analyses demandées, la création des interfaces graphiques, l’implémentation des spécifications mentionnées dans la conception.

* Phase de test et de validation :

A chaque fonctionnalité terminée, un test est effectué et une correction est ajoutée si besoin.

Après la phase de développement, un test final des cas d’utilisation possibles est effectué avant leur validation.

Après les corrections nécessaires, les métiers procèdent aux tests finaux. Une fois les corrections validées on peut procéder au déploiement.

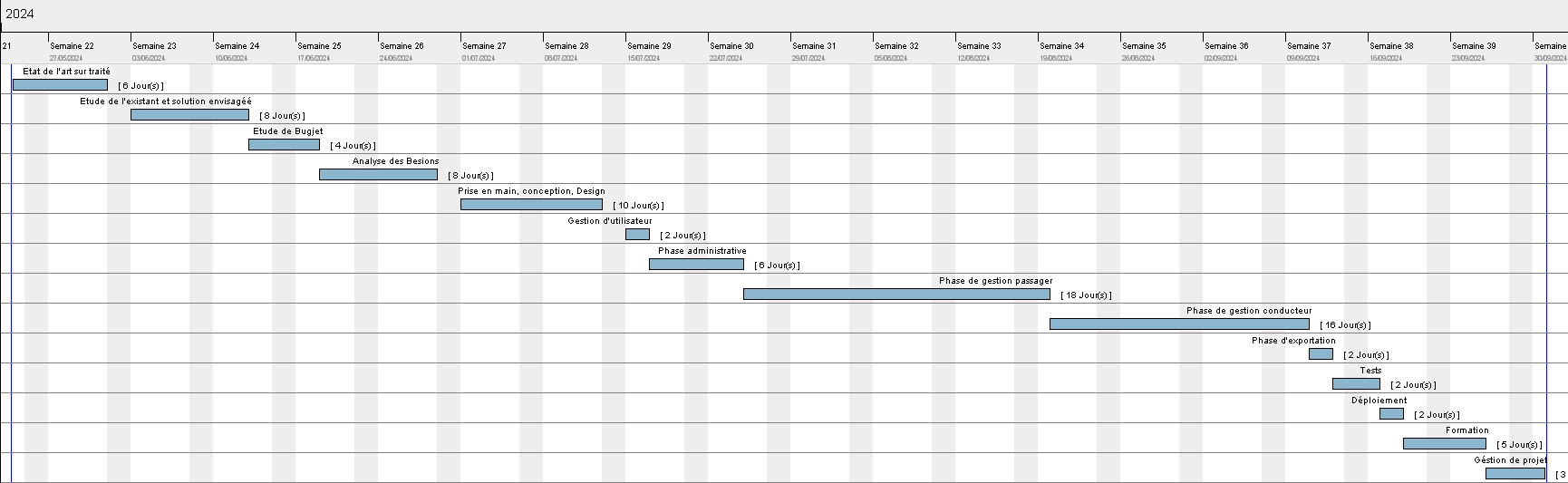
Le diagramme de Gantt ci-dessous résume la planification du projet.

Figure 1 : Diagramme de Gantt

## Budget du projet

# Exigences réalisées dans le projet (vision externe/utilisateur)

## Exigences fonctionnelles – Cas d’utilisation

Ces exigences décrivent les fonctionnalités principales que l’utilisateur final peut accomplir à travers l’application. Elles sont modélisées à travers des cas d’utilisation.

Avant d’entrer dans le détail de certains cas, nous pouvons voir ci-dessous un tableau récapitulatif des cas d’utilisations dans le projet.

| N° | Cas d’utilisation | Description |
| --- | --- | --- |
| CU1 | Inscription et authentification | Permet aux utilisateurs de s’inscrire et se connecter. |
| CU2 | Réservation d’une course | Permet de réserver un véhicule pour une course. |
| CU3 | Suivi en temps réel de la course | Permet de suivre la position du véhicule. |
| CU4 | Paiement en ligne | Permet de payer la course via une plateforme sécurisée. |
| CU5 | Système de notation bidirectionnel | Permet aux passagers et chauffeurs de se noter mutuellement. |

Tableau 4 : Liste des cas d’utilisation

### Cas d’utilisation 1 - Réservation d’une course

**Nom :** Réservation d’une course

**Préconditions :** L’utilisateur doit être connecté à l’application et l’application doit être en état de fonctionnement

**Postconditions :** la course est confirmée, la course est assignée à un chauffeur.

**Description :**

* L’utilisateur choisit son lieu de départ et d’arrivée sur la carte
* L’application affiche une estimation du prix de la course
* L’utilisateur fixe la date de réservation, valide la réservation et attend sa confirmation
* Tous les chauffeurs disponibles sont notifiés de la nouvelle course
* Un message de confirmation de la réservation est envoyé à l’utilisateur

**Erreurs possibles :**

* Aucun chauffeur disponible dans le créneau
* Erreur de connexion réseau

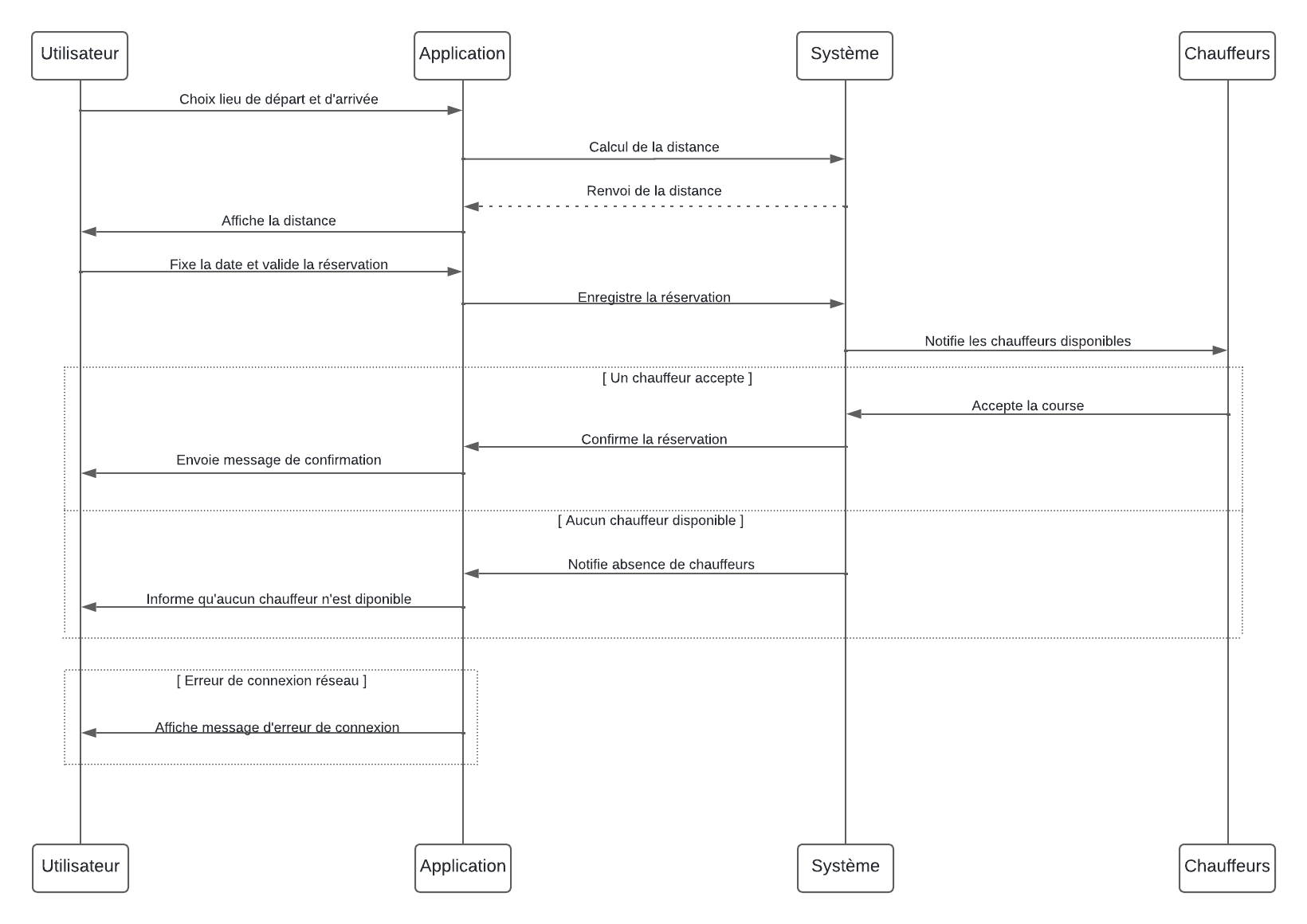


Figure 2 : Cas d’utilisation 1 - Réservation d’une course

### Cas d’utilisation 2 - Suivi en temps réel de la course

**Nom :** Suivi en temps réel de la course  
**Préconditions :** La course est en cours, l'utilisateur et le chauffeur sont connectés.  
**Postconditions :** La position du véhicule est mise à jour en temps réel.  
**Description :**

* L'utilisateur peut voir la position en temps réel du véhicule sur la carte.
* Le chauffeur suit l'itinéraire indiqué par l'application.
* L'application met à jour la distance restante et le temps estimé d'arrivée.

**Erreurs possibles :**

* GPS indisponible.
* Mauvaise connexion réseau.

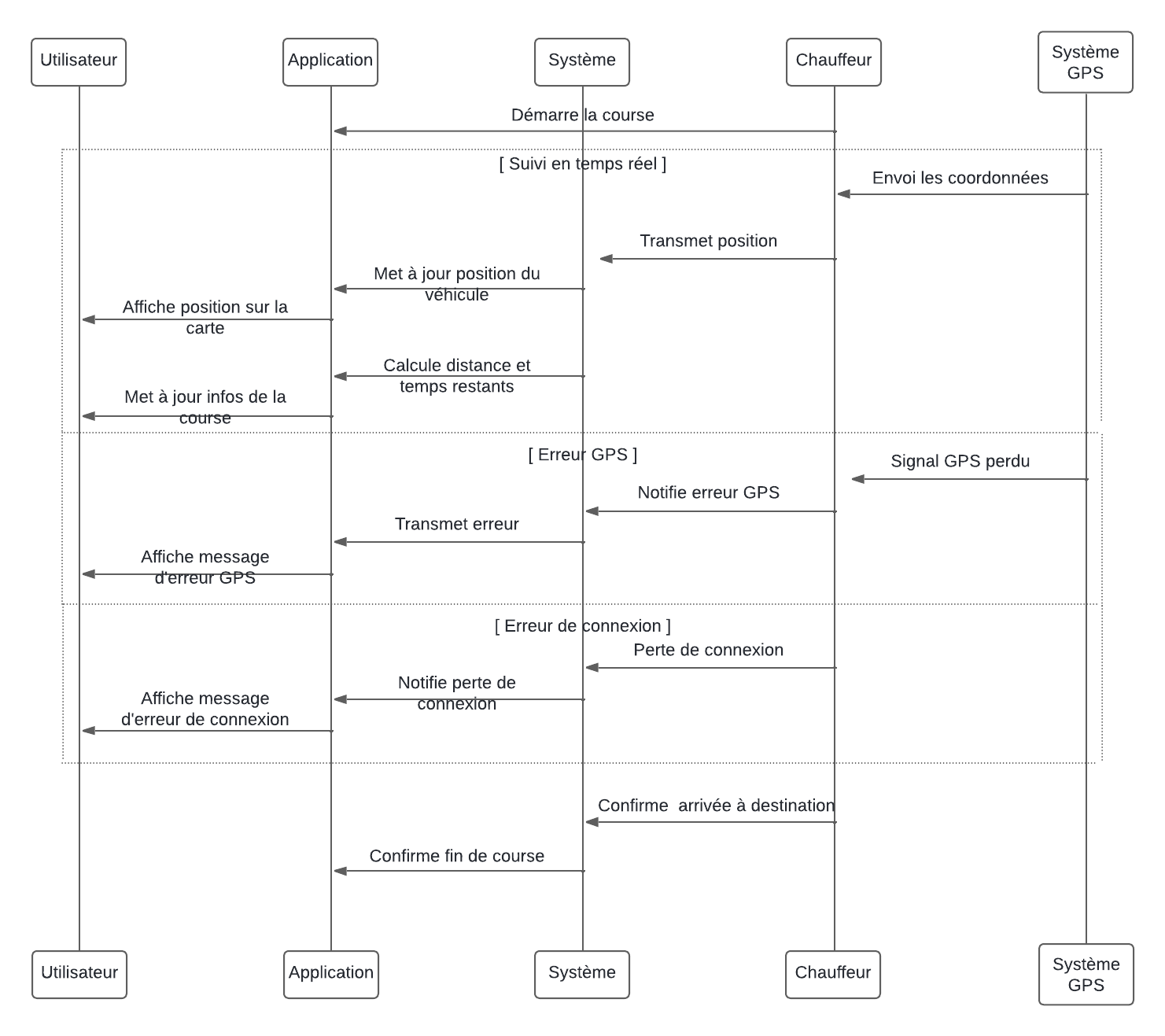


Figure 3 : Cas d’utilisation 2 - Suivi en temps réel de la course

### Cas d’utilisation 3 - Paiement en ligne

**Nom :** Paiement en ligne  
**Préconditions :** L’utilisateur a fait son choix parmi les chauffeurs ayant accepté sa demande de réservation.  
**Postconditions :** Le paiement est confirmé et une facture est générée.  
**Description :**

* L'utilisateur choisit son chauffeur parmi la liste ayant accepté sa demande.
* L'utilisateur entre ses informations de paiement (paiement via carte bancaire).
* L'application envoie la demande de paiement à la plateforme de paiement sécurisée.
* Une fois le paiement validé, une facture est envoyée à l'utilisateur.

**Erreurs possibles :**

* Paiement refusé.
* Problème de connexion avec la plateforme de paiement.

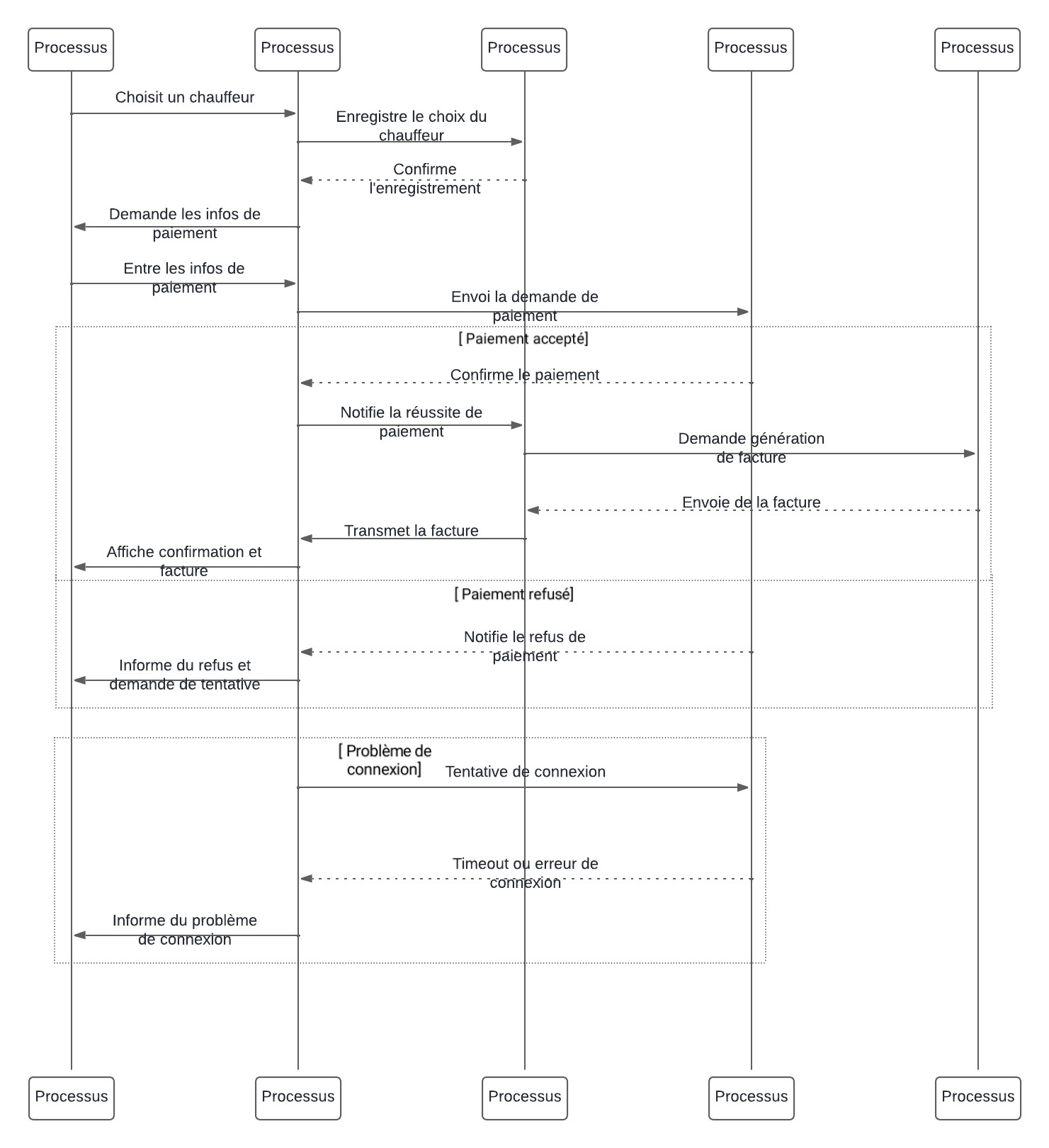


Figure 4 : Cas d’utilisation 3 - Paiement en ligne

## Exigences non fonctionnelles transverses

Cette section décrit les contraintes générales que l’application doit respecter pour assurer une bonne expérience utilisateur et une performance adéquate.

* **Utilisabilité** : L’interface doit être simple à utiliser, intuitive, avec un design épuré respectant une charte graphique moderne. L’utilisateur doit pouvoir réserver une course en moins de 5 étapes.
* **Performances** : Le temps de réponse pour toutes les requêtes, y compris la réservation et le suivi des courses, devrait être court pour l’expérience utilisateur.
* **Robustesse** : L’application doit tolérer les pannes réseau (reprise de session).
* **Sécurité** : L'authentification à deux facteurs doit être disponible pour les utilisateurs. Les mots de passe doivent être cryptés et les APIs doivent être protégés par des jetons d'authentification.
* **Supportabilité** : Le système doit être évolutif, avec une capacité à ajouter des fonctionnalités sans affecter les services existants. De plus, il doit être maintenable sur le long terme.

## Interfaces détaillées

### IHM

#### Cas d’utilisation 1 - Réservation d’une course

#### Cas d’utilisation 2 - Suivi en temps réel de la course

#### Cas d’utilisation 3 - Paiement en ligne

### Interfaces avec d’autres systèmes

# Architecture(s) système

## Architecture logicielle

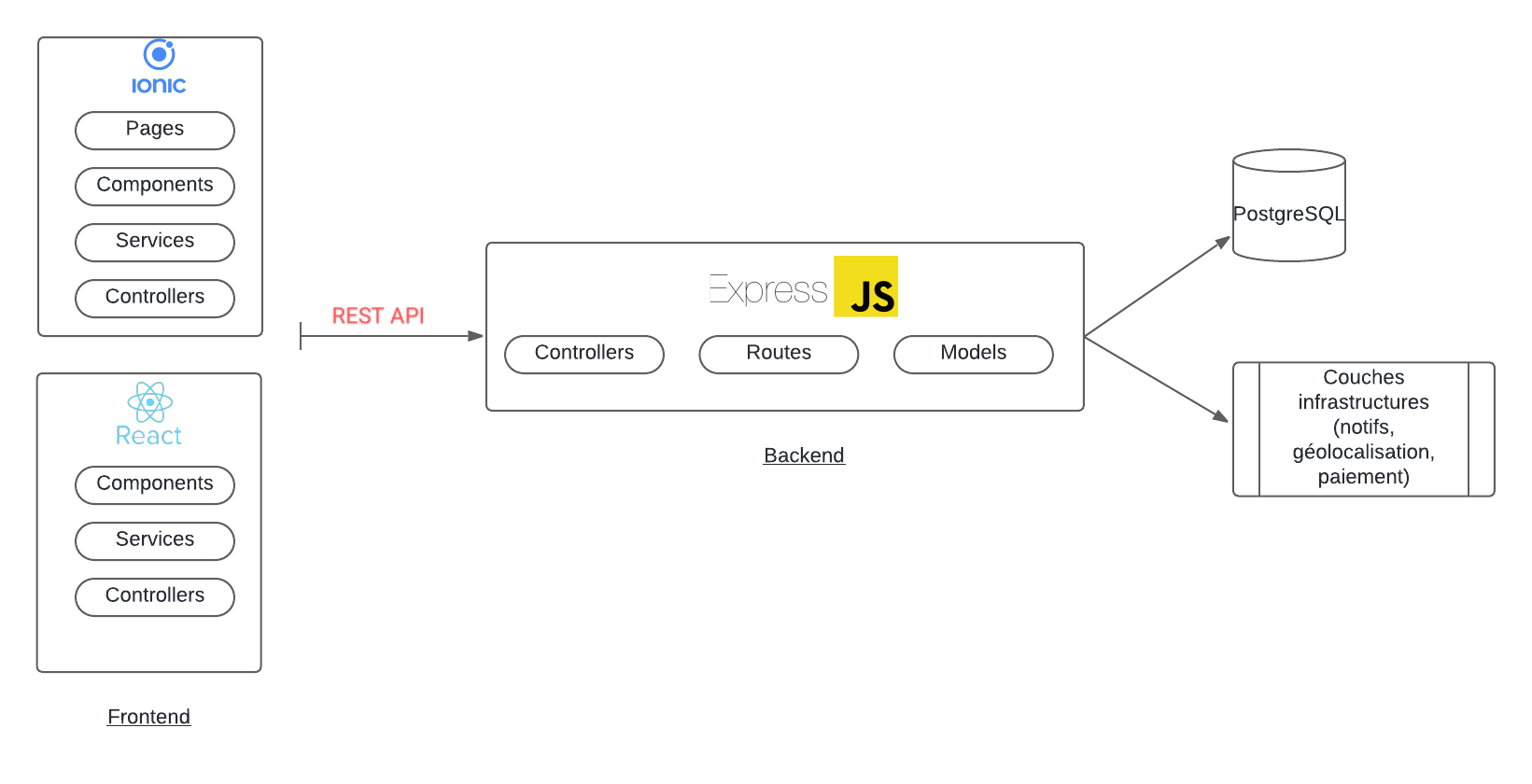
L’architecture logicielle est une vue tournée sur l’organisation interne des logiciels du système informatique. Nous pouvons la représenter de la manière suivante : 

Figure 5 : Architecture logicielle

## **Architecture** technique

L’architecture technique s’appuie sur la vision des flux d’informations entre composants d’une part, et le diagramme de déploiement applicatif d’autre part. Elle peut être schématisée de la manière suivante :

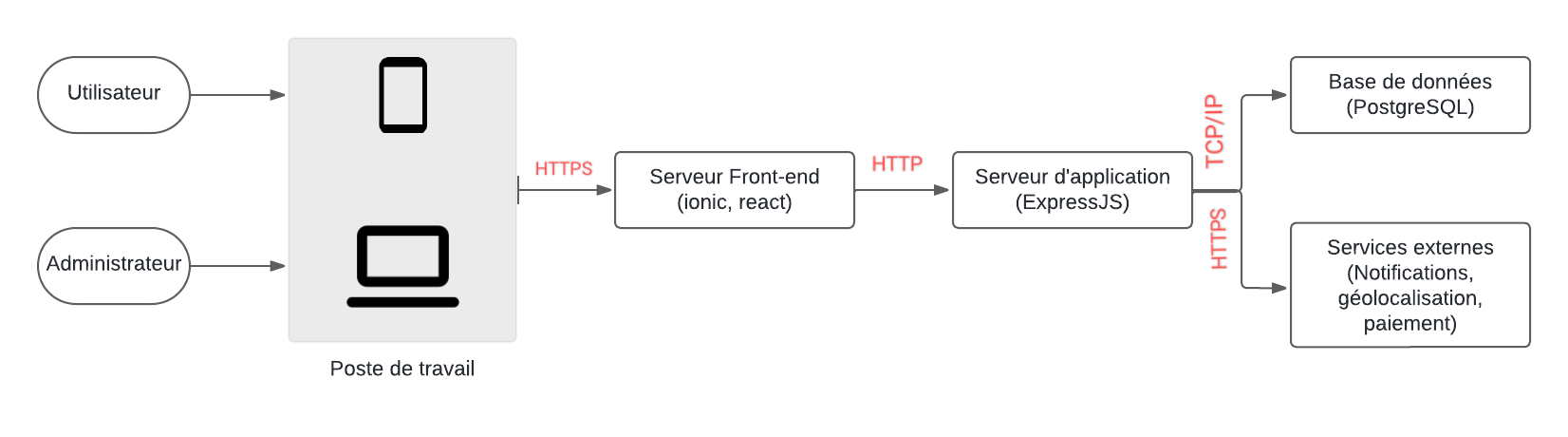
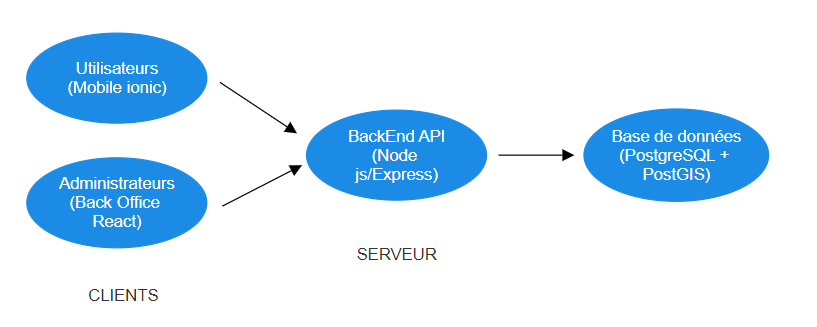


Figure 6 : Architecture technique

# Conception du système logiciel réalisée dans le projet (vision interne/développeur)

## Plate-forme technique

### 7.1.1 Architecture matérielle

Figure 7 : Architecture matérielle

### 7.1.2 Framework Logiciel

* **Ionic** pour l'interface utilisateur mobile, permettant le développement multiplateforme (Android/iOS). **Ionic** a été choisi pour sa capacité à produire des applications multiplateformes avec un code unique, réduisant ainsi le temps de développement et de maintenance.
* **Node.js/Express** pour le backend, qui assure la gestion des utilisateurs, des courses, des paiements, et des notifications. **Node.js** a été sélectionné pour sa rapidité dans la gestion des requêtes et sa compatibilité avec JavaScript, ce qui facilite la communication avec le frontend.
* **PostgreSQL** avec l'extension **PostGIS** pour stocker et traiter les données géospatiales. **PostGIS** permet une gestion efficace des données géographiques, essentielle pour la fonctionnalité de suivi en temps réel.
* **React** pour le back-office administrateur, permettant la gestion des utilisateurs, des chauffeurs, et des courses.

### 

## Conception du logiciel développé

### Conception du code source

* Pour l’application mobile (Ionic/React) :
* **Pages** : Chaque page représente une vue distincte de l’application (ex. : page de connexion, page de réservation). Le dossier *“pages”* regroupe toutes les pages, favorisant une navigation claire et organisée.
* **Components** : Les composants réutilisables (boutons, formulaires, etc.) sont placés dans le dossier “c*omponents”.* Cette séparation permet une meilleure réutilisation du code et facilite les tests unitaires.
* **Services** : Les appels à l'API sont centralisés dans le dossier *“services”*. Cela permet de gérer les interactions avec le backend de manière cohérente et modulaire, et de respecter le principe de séparation des préoccupations.
* Pour le backend (Node.js/Express) :
* **Models** : Le dossier *“models”* contient les schémas de données pour PostgreSQL. Chaque modèle représente une entité (ex. : Utilisateur, Course) et applique des bonnes pratiques de modélisation relationnelle. Ils sont responsables de traiter les requêtes et les opérations.
* **Controllers** : Le dossier *“controllers”* contient les contrôleurs pour chaque route. Les contrôleurs sont responsables de valider les données, et d'interagir avec les modèles.
* **Routes** : Le dossier *“routes”* gère les points d'entrée de l'API (ex. : /auth, /courses). Chaque route est associée à un contrôleur, et un middleware est utilisé pour la gestion de l’authentification et de la sécurité.

### Le code source – vue statique

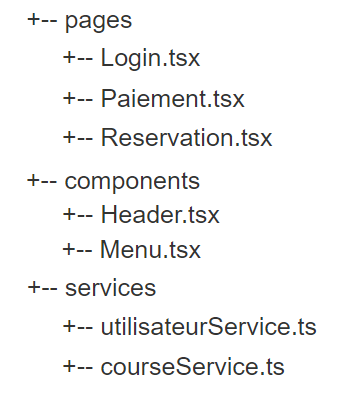


Figure 8 : Vue statique frontend (Ionic/React)



Figure 9 : Vue statique backend (Node.js/Express)

### Modélisation de données

### Les composants et leur déploiement

L'API backend est développée avec **Node.js** et **Express** et organisée en **modèles**, **contrôleurs**, et **routes**. Lors du déploiement, l'application Node.js est packagée et déployée sur un serveur d'applications, tel que **Render**.

L'application mobile est développée avec **Ionic**, et le code est structuré en composants, pages, et services. Lors du déploiement, le code source de l’application est compilé en un ensemble de fichiers statiques (HTML, CSS, JavaScript) qui sont ensuite empaquetés pour être distribués sous forme d’**APK** (pour Android) et d’**IPA** (pour iOS).

# Tests du système logiciel

À la fin de chaque fonctionnalité, l'application est d'abord testée par mes soins, puis par le chef de projet à qui j'explique son fonctionnement. Après les tests effectués par les équipes métiers, nous apportons les ajustements nécessaires en cas de retour.

# Conclusion générale

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

Concernant Moov, les objectifs posés au début du stage ont été généralement atteints :

* La réservation d’une course et sa suivie
* Système de gestion de paiement
* La notation bidirectionnelle pour l’amélioration des qualités de service

Parmi les principaux problèmes rencontrés, la gestion de la géolocalisation en temps réel s’est avérée complexe. Une solution a été trouvée en intégrant un service externe de cartographie pour améliorer la précision.

Le projet a encore de la place pour des améliorations, telles que l’ajout d’une fonctionnalité de suggestion automatique de chauffeurs en fonction de la proximité et de la disponibilité. De nouveaux services pourront encore être ajoutés, comme un service de messagerie pour la communication entre chauffeur et client par exemple, ou bien des abonnements pour les utilisateurs réguliers.

# Références et Bibliographie

# Annexes

1. AR/VR : Augmented Reality/ Virtual Reality [↑](#footnote-ref-0)
2. IoT : Internet of Things [↑](#footnote-ref-1)