

METZ NUMERIC SCHOOL

LOT 1

Plateforme de covoitage

Auteurs :

Alexandre GERARD

Vincent GIANGRECO

*Description du contexte, des contraintes et de la méthodologie du
projet de plateforme de covoitage*



5 août 2022

Table des matières

1	Introduction	1
2	Besoin et objectifs du projet	3
2.1	Contexte	3
2.2	Besoins	3
2.3	Analyse de l'existant	5
2.3.1	Sites pré-existants	5
2.3.2	Stack technique	5
2.3.3	Interface utilisateur	6
2.3.4	Expérience utilisateur	6
2.4	Attentes de l'utilisateur	6
2.5	Contraintes du projet	7
2.5.1	Contrainte technique	7
2.5.2	Contrainte de temps	7
2.5.3	Contrainte monétaire	7
2.5.4	Contrainte législative	8
3	Méthodologie	9
3.1	Outils de gestion de projet et d'équipe	9
3.2	Structure de la plateforme	9
3.2.1	Une architecture monolithique	10
3.2.2	Structure de la base de données	10
3.2.3	Structure coté client	11
3.2.4	Structure coté API	12
3.2.5	CI/CD	12
3.3	Gestion de projet	13
3.3.1	Diagramme de GANTT	13
3.3.2	Jalons	13
3.3.3	Répartition des tâches	13

4 Modélisation	15
4.1 Identité du site	15
4.1.1 VroomMates	15
4.1.2 Logo	15
4.1.3 Identité du site	15
4.1.4 Palette de couleurs	16
4.2 Diagrammes de séquence	16
4.3 Maquettage	17
5 Bilan du lot 1 et perspectives	19
Annexe	21
A.1 Définition des tâches et de leur dates	21
A.2 Diagramme de GANTT	22
A.3 Stack technique complète du site	23
A.4 Modèle non-relationnel de la base de données	24
A.5 Diagramme de séquence : Authentification de l'utilisateur	25
A.6 Diagramme de séquence : Participer à un voyage	26
A.7 Diagramme de séquence : Partie conducteur	27
A.8 Diagramme de séquence : Partie administrateur	28
A.9 Plan du site	29
A.10 Maquettage : Accueil	30
A.11 Maquettage : Recherche	31
A.12 Maquettage : Détails d'un voyage	32
A.13 Maquettage : Profil d'un utilisateur	33
A.14 Maquettage : Messagerie instantanée	34
Bibliographie	35

Introduction

Dans le cadre de la validation de notre année de BAC+3 pour la formation Développeur Full Stack, *Metz Numeric School* nous a proposé le choix entre trois projets :

- Une plateforme d'E-Learning ;
- Une plateforme de covoiturage ;
- Un "serious game" ;

La première priorité du projet était la constitution d'un groupe de travail. C'est ainsi que vint à naître ce binôme composé de GERARD Alexandre et de GIAN-GRECO Vincent, ayant déjà travaillées ensemble.

Nous nous sommes donc décidés de s'intéresser de plus près au sujet de **covoiturage**.

Nous avons choisi ce sujet car c'était tout d'abord un projet nous tenant à cœur ainsi qu'un challenge pour nous, tous deux habitués de sites de covoiturage comme BlaBlaCar, de pouvoir concevoir en peu de temps une plateforme concurrente à celle-ci et ainsi pouvoir comprendre certaines problématiques qui s'imposent.

Besoin et objectifs du projet

2.1 Contexte

On remarque que de nouvelles questions environnementales ont été soulevées ces dernières années, surtout envers les automobilistes. L'impact environnemental des voitures se fait directement ressentir sur l'état de la planète et il est simplement impossible d'empêcher les déplacements en voiture pour certains (par exemple pour ceux qui habiteraient loin de leur lieu de travail).

Désormais il existe un système alternatif de partage de trajets en voiture, permettant une diminution importante des congestions routières, de la pollution et du coût d'utilisation : **le covoiturage** (*ou carpooling*).

2.2 Besoins

Afin de définir les besoins précis du projet, établissons un diagramme bête à cornes puis utilisons de la méthode empirique "QQOQCCP" afin de mieux définir nos attentes.

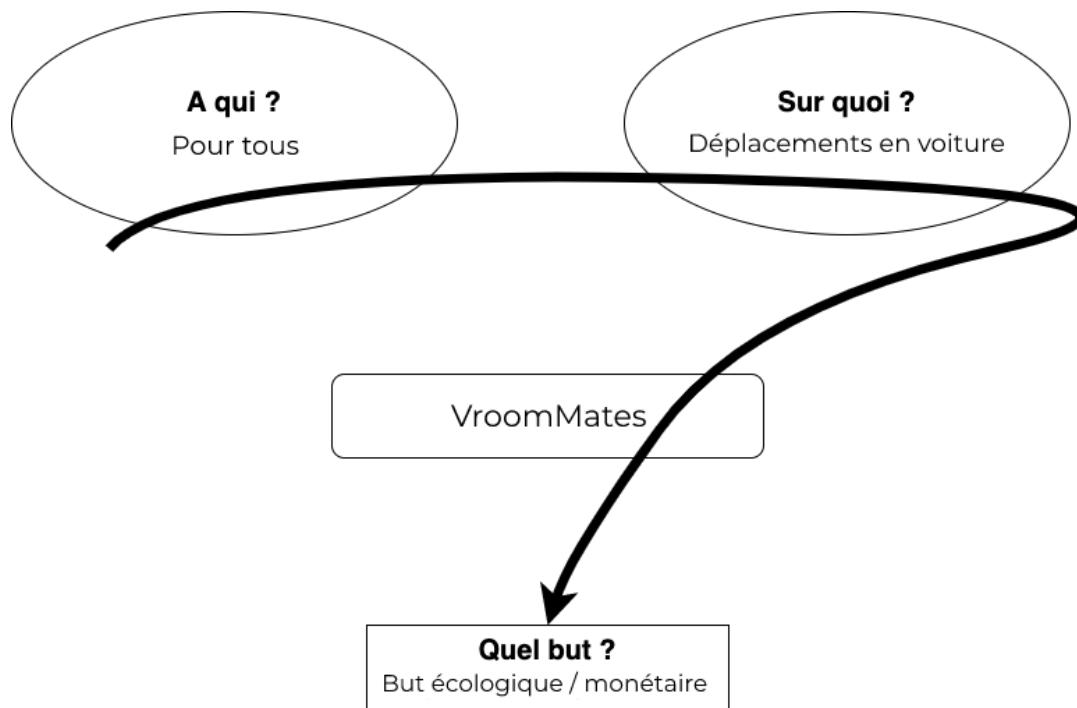


FIGURE 2.1 – Diagramme de bêtes à cornes

Pourquoi ? Les moyens de déplacement par voiture deviennent de plus en plus problématiques, surtout quand plusieurs personnes vont au même endroit avec une voiture chacun.

Quoi ? Nous voudrions offrir une solution de mobilité pour ses utilisateurs qui soit abordable, efficace et inscrite dans une démarche écologique. Nous devons donc mettre en place certains outils, notamment pour la gestion de projet et d'équipe, afin de résoudre notre problématique de la meilleure façon possible (voir [3.1](#)).

Pour qui ? Nous voudrions permettre l'utilisation du service au plus grand nombre, sans distinction d'âge, d'origine ou de situation sociale... Cela se ressentira au niveau de notre interface graphique intuitive ainsi qu'à notre politique.

Qui ? Les prestataires sont nous-mêmes, le groupe chargé de travailler sur ce projet. Nous avons tout deux les mêmes responsabilités et le même pouvoir décisionnel vis-à-vis du projet.

Où ? On remarque cette problématique non seulement en France, mais aussi dans la majorité des autres pays du monde, d'où la nécessité de ne pas se contenter de fournir une solution accessible exclusivement dans l'hexagone.

Comment ? Nous allons devoir mettre en place une plateforme Web accessible depuis n'importe quel appareil dit "smart" (téléphone, ordinateur, tablette, ...). Le site sera cependant optimisé pour mobile et ordinateur. Elle se composera d'un serveur, pour stocker les données(data) et d'une interface Web qui permettra à un utilisateur de se connecter pour commander et/ou de proposer un trajet. On notera aussi l'importance d'avoir un accès administrateur afin de superviser le site.

Combien ? Étant donné la nature du projet (projet étudiant), nous allons devoir réaliser ce travail sans faire recours à des services payants (voir [2.5.3](#)).

Quand ? Nous avons jusqu'au 05 octobre pour terminer ce projet et le présenter. Un calendrier mis en place sera explicité plus en détail dans les sections à venir (voir [2.5.2](#)).

2.3 Analyse de l'existant

2.3.1 Sites pré-existants

La recherche de l'existant est une étape qui nous est nécessaire lors de la création d'une application ou d'un site d'une telle ampleur : il permet de comparer les spécificités fonctionnelles attendues à celles déjà en place pour des entreprises concurrentes, d'analyser leurs problèmes, leurs points forts, et leurs points faibles afin de concevoir un site innovant autant techniquement que fonctionnellement parlant. Tout au long de la création de la plateforme, nous nous sommes inspirés de sites connus dans le domaine. Le marché, n'étant pas si récent, une multitude de sites et applications existent déjà sur le covoitage, notamment :

- **Gett**, leader sur le marché mondial de covoitage.
- **BlaBlaCar**, une plateforme communautaire payante avec plus de 100 millions d'utilisateurs.
- **MobiCoop**, une coopérative des trajets quotidiens, avec plus de 500 000 utilisateurs, présent dans 1200 communes environ.

2.3.2 Stack technique

Nous nous sommes aussi par ailleurs intéressés aux technologies utilisées par la plupart des sites proposant des services similaires, afin de mieux cerner celles dont nous aurions la nécessité :

- En termes d'outils collaboratifs, on note que BlaBlaCar et Mobicoop utilisent tous deux principalement **Trello**.
- Pour stocker les données, on remarque une majorité de sites de covoitage utilisant des bases de données "**NoSQL**" (en schémas **document**).
- Comme Gett, BlaBlaCar fonctionne essentiellement avec **Node.js et Express** pour la gestion des routes.
- Un nombre conséquent de sites se voient utiliser soit PHP avec Symfony, soit **TypeScript couplé avec ReactJS** afin de réaliser leur client (c'est le cas de BlaBlaCar et de Gett qui utilisent React).
- Côté DevOps, les sites utilisant les technologies en JS/TS cités plus haut utilisent **Jest** pour réaliser leurs tests ainsi que des services comme **GitHub Actions** comme système de déploiement / d'intégration continue (CI/CD).

2.3.3 Interface utilisateur

En ce qui concerne l'interface utilisateur, BlaBlaCar a un style très intuitif pour la version navigateur. Leur barre de recherche disposée au centre de l'écran directement à l'accueil, des onglets bien nommés facilitant le parcours de l'utilisateur, et la colorimétrie faisant ressortir directement tous les éléments importants de la page font de BlaBlaCar un bon modèle à suivre. Cependant, plusieurs choses n'allait pas, comme par exemple leurs zones de texte à répétition, qui ajoute un certain poids visuellement au site, leur interface bien trop blanche à notre goût. Ce sont donc des points sur lesquels nous nous sommes penchés, afin de réaliser un site économiquement meilleur. En ce qui concerne mobicoop, leur interface est très colorée. Selon nous, trop colorée. L'utilisateur se perd vite, de plus, la partie de recherche de trajet n'est pas bien mise en avant, il faut faire plusieurs allers et retours afin de trouver les informations recherchées. De plus, nous avons remarqué une absence de responsive design.

2.3.4 Expérience utilisateur

Grâce à cette analyse, nous avons pu observer quelques points faibles pour BlaBlaCar, par exemple : une recherche souvent soumise à des erreurs de saisie de l'utilisateur, fréquence assez élevée d'apparition de message de problème technique, etc. Ce qui est dérangeant lors du parcours de l'utilisateur, même pour nous, alors que nous sommes des habitués aux sites internet et ses erreurs. Quelqu'un de plus "novice" en informatique serait peut-être soumis à un stress et pourrait interrompre son parcours. De même pour les paiements : nous avons tous deux déjà effectué au moins un voyage via BlaBlaCar. Une fois un voyage réservé, peu importe si le conducteur accepte ou non, l'argent est bloqué sur le compte. De plus, beaucoup de conducteur refuse les demandes, on se retrouve facilement avec plus de 200 voire 300€ alloués à BlaBlaCar. Heureusement, l'argent ne quitte pas le compte, mais pendant une semaine, celui-ci ne pourra pas être utilisé. La confiance est vite perdue entre l'utilisateur et le site, d'autant plus quand il s'agit du paiement.

2.4 Attentes de l'utilisateur

À l'accès du site par un utilisateur, ce dernier s'attend à pouvoir effectuer une série d'actions. On note particulièrement :

La création d'un compte (via Google ou non) et la possibilité de le modifier par la suite. Mettre une photo de profil, renseigner une biographie descriptive, plusieurs renseignements en ce qui concerne les voyages (bavard ou non, présence de musiques

pendant les trajets ou non, etc.), et quelques hobbies. Pouvoir réserver un trajet (fréquent ou non), en passant par la recherche d'annonces, le paiement, les messages privés pour plus de renseignements, la notation du conducteur (et du passager dans le cas inverse). La possibilité de visualiser tous les voyages futurs ou passés, que ce soit pour un conducteur ou un passager. Une page d'administration avec la possibilité de bannir des membres, la présence de diagrammes statistiques importantes sur le site (nombre de comptes par pays, nombre total de voyages), ainsi que la possibilité d'accepter ou de refuser une demande de vérification du permis.

2.5 Contraintes du projet

2.5.1 Contrainte technique

La première contrainte est technique : La structure et les outils qui vont être mis en place peuvent ne pas avoir été utilisés préalablement par nos membres de l'équipe, il faudra donc s'adapter en conséquence grâce à la planification de ce lot 1 et sur les connaissances pré-existantes de l'un et de l'autre.

2.5.2 Contrainte de temps

Le projet a été lancé le 7 mars 2022. Pour le 19 juillet, nous devions rendre ce premier rapport constituant toutes les étapes de conception et de coordination du projet. Ce temps a l'air amplement suffisant à premier abord. Cependant, quand on se penche un peu plus sur le sujet, on s'aperçoit vite de la quantité de travail que cela demande, à réaliser dans une période de la journée très brève (après le travail ou les cours). Forcément, plus nous avancions dans le temps, plus nous avions de nouvelles idées à ajouter au site, avec d'autant plus de travail derrière. Pour réaliser et administrer tout le système, nous avons une fourchette de plus ou moins 2 mois et demi, ce qui à première vu paraît aussi bref que le premier lot l'est.

2.5.3 Contrainte monétaire

En ce qui concerne le budget des dépenses en vue de la réalisation de ce projet, nous nous sommes fixés comme objectif de ne rien dépenser. Nous avons donc dû trouver des alternatives à des services de stockage de données comme DynamoDB, qui est payant.

2.5.4 Contrainte législative

Nous avons ajouté des pages de termes et de conditions. Cette page est née d'une certaine contrainte législative liée à la GDPR[3] (RGPD), et sera explicitée plus en détails dans le second lot.

Méthodologie

Pour réaliser ce travail de méthodologie, une série d'objectifs (du choix des outils en passant par la création de divers diagrammes) a été accomplie.

3.1 Outils de gestion de projet et d'équipe

GanttProject

GanttProject est un logiciel sous GNU GPL (licence publique de distribution), existant depuis 2003. Il fait parti des pionniers dans ce type d'application.

Comme son nom l'indique, il va nous permettre de réaliser un diagramme de GANTT afin de suivre notre avancée au fil du temps.

Visual Paradigm

Visual Paradigm est un logiciel gratuit mis au point en Java le 20 juin 2002 par Visual Paradigm International Ltd.

Ce logiciel va nous permettre de facilement créer tous les diagrammes UML/SYML qui vont nous être utiles.

Git

Git est un logiciel connu de tout développeur informatique, lancé en avril 2005. Utilisant les protocoles HTTP(S) / SSH ainsi que celui spécifique à Git, sa fonction première est la gestion de versions.

Ce dernier, couplé à GitHub, va nous servir d'outil de versionning de code.

Trello

Trello est un logiciel de gestion de projet en collaboration en ligne lancé en septembre 2011.

Le kanban de Trello va nous servir à suivre notre avancée en termes de jalons et de tâches.

3.2 Structure de la plateforme

Une fois les outils de collaboration choisis, le choix des technologies de programmation ainsi que la structure du projet allaient devoir être explicités. Pour nos

besoins, nous avons tout de suite su quel modèle architectural nous allions choisir : **une API RESTful monolithique.**

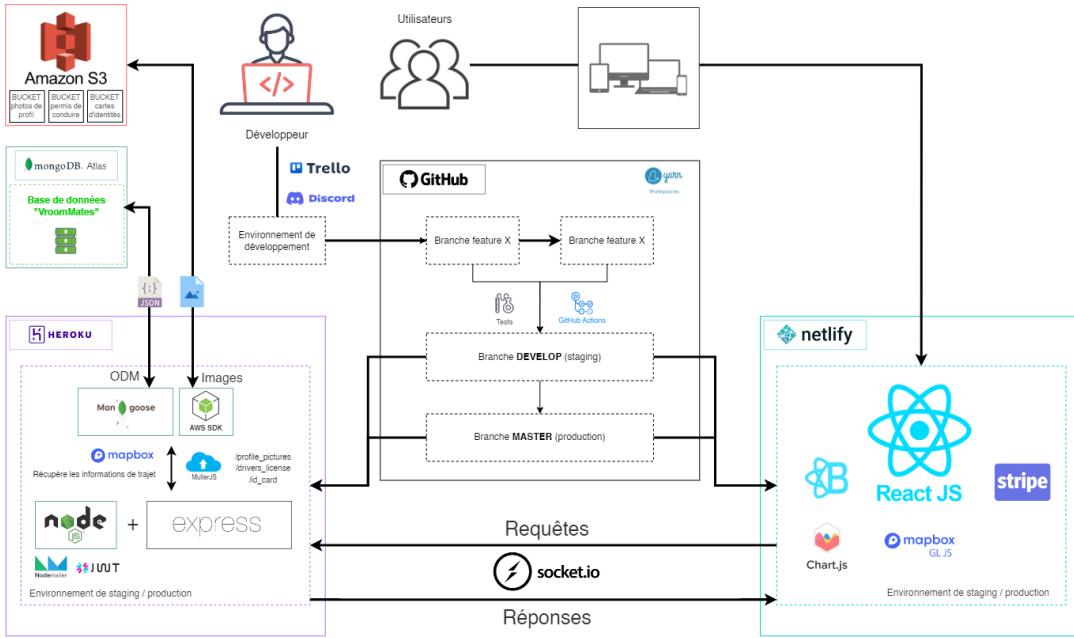


FIGURE 3.1 – Structure de la stack technique (voir en grand : A.3)

3.2.1 Une architecture monolithique

Une architecture monolithique est formée d'un seul bloc regroupant l'intégralité de nos services : avec l'interface utilisateur, la logique métier, l'interface de données, ainsi que la base de données.

Cette architecture nous a permis de centraliser notre code dans un seul et même repository (dépôt) → Il s'agit de l'architecture mono-repo[1].

Nous utiliserons **Yarn Workspace** afin de profiter des avantages de ce type d'architecture (un seul fichier lockfile, toutes les dépendances installées au même endroit, une gestion des tests plus simple).

Toute la codebase sera donc accessible publiquement via ce lien GitHub : <https://github.com/agerard57/VroomMates>.

3.2.2 Structure de la base de données

Nous avons, pour cette partie, beaucoup effectué de tests et de recherche, non seulement pour trouver quel outil utiliser (MongoDB, DynamoDB, Elasticsearch, mySQL, etc...), mais déjà pour savoir quel type de schéma la table allait avoir. Pour

cela, nous devions déjà répertorier quelles données allaient être stockées ; Hormis les éléments stockables dans une base de données relationnelle "classique" (comme par exemple les données sur un utilisateur), nous avions des données plus spécifiques à notre projet à entreposer : il fallait trouver un moyen de stocker des positions géographiques (un trajet = point A → point B). C'est pour cela que nous nous sommes tournés vers **les bases de données document**.

Étant un système de documents non relationnel, nous pouvons imbriquer des listes et des objets, ce qui s'avère pratique pour des coordonnées. Nous allons donc utiliser de façon plus précise **MongoDB** : nous pouvons ainsi utiliser leur possibilité de stockage d'objets **GeoJSON**[2].

Notre base de donnée sera déployée dans le cloud et uniquement accessible par l'api grâce à la mise en place du service **MongoDB Atlas**.

Voir le modèle de la base de données : [A.4](#)

En plus de MongoDB, un système de stockage d'objets allait devoir être mis en place afin de garder des images comme les photos de profil ainsi que les permis de conduire). Nous avons sélectionné **Amazon S3**, l'un des membres du groupe étant déjà familier avec le service.

3.2.3 Structure côté client

Côté client, après avoir fait la recherche de l'existant sur le point technique (voir [2.3.2](#)), et, avec l'expérience pré-acquise, nous allons nous diriger vers **ReactJS**, **en TypeScript**. Une grande rigueur sera attendue au niveau de la qualité du code, puisque ce dernier sera testé par des outils comme ESLint, prettier, prettier-sort-imports, etc.... Nous allons nous aider de React-Bootstrap au niveau de la réalisation de la responsiveness. Sur ce thème, nous allons d'ailleurs adopter une approche "**mobile-first**", vu la nature de la plateforme. En ce qui concerne les api externes, nous seront aidés de **Stripe** (pour les paiements) et **mapbox** (pour les cartes). Le tout sera automatiquement déployé sur Netlify grâce à notre système de CI/CD.

L'URL du site étant par défaut accessible uniquement par un sous-domaine netlify (<https://tourmaline-cascaron-c8cbd7.netlify.app>), nous avons décidé de rediriger le client vers un TLD personnel, question de simplicité et d'esthétisme. Le site sera donc accessible au sous-domaine <https://www.vroommates.agerard.dev>. Cela nous permettra d'avoir une URL plus présentable sans pour autant dépenser pour un nouveau domaine. Le déploiement du site n'étant pas demandé initialement, cela nous permettra de pratiquer certains aspects de création de site que nous n'aurions pas pu

faire si nous étions restés uniquement en local (par exemple, de la CI/CD ou de la SEO)...

3.2.4 Structure coté API

Afin de se conformer aux choix de langages du client, nous avons décidé d'utiliser **NodeJS + Express.js, en JavaScript**. Dans cette partie du projet, plusieurs outils ont été pensés à des fins bien précises :

- Pour les fonctionnalités de chat, étant donné la création d'un système de messagerie instantanée, nous avons opté pour **socket.io**
- Pour la protection des routes et pour l'authentification, **JWT** nous permettra d'utiliser leurs jetons.
- Pour relier la base de données MongoDB et l'API, nous utiliserons un ODM, précisément **Mongoose**
- L'API de **mapbox** nous servira à récupérer des informations sur le trajet (principalement convertir des coordonnées en nom de rue/ville)
- Afin de valider l'adresse Email de l'utilisateur, nous allons utiliser le module **NodeMailer** qui va envoyer des codes de confirmation par ce même biais.
- Comme précisé plus tôt, étant donné que nous avons besoin de stocker des images, nous ne voulions pas simplement les stocker sur le serveur avec **MulterJS**. Nous nous en servons alors juste pour stocker une image le temps de l'envoyer dans son bucket S3 respectif avec **AWS SDK**

Nous avons donc eu, tout comme le client, l'intention de déployer l'API sur un serveur. Nous utiliserons les service d'**Heroku**. Ce dernier est disponible à l'adresse suivante : <https://vroommates-agerard57.herokuapp.com/>

Notons que nous n'avons pas besoin de changer le nom de domaine / sous-domaine car ce dernier ne sera utilisé uniquement que par le client, et non pas directement l'utilisateur.

3.2.5 CI/CD

Notre plateforme aura aussi la particularité de proposer un système avancé de CI/CD. Pour commencer, notre repository utilisera les fonctionnalités de github actions.

Voici le fonctionnement d'un déploiement :

- Avant chaque pull request vers la branche "develop", nous mettrons en place des workflows qui vérifieront la bonne mise en forme du code (eslint, prettier, sort imports) ainsi que de la documentation (markdownlint) ;

- Ensuite, un second set de workflows feront des tests unitaires, des tests d'intégration, des tests end-to-end et/ou des tests d'acceptation. Nous utiliserons Jest couplé avec Puppeteer pour ces tests ;
- Une phase de code review entre nos membres va dès lors se faire ;
- Enfin, après notre validation, lors du merge (fusion) vers la branche "develop", un dernier set de workflows va respectivement déployer le "front-end" et le "back-end" à Netlify et Heroku. Bien évidemment, ces workflows ne vont se déclencher qu'à la réussite des précédents tests. On vérifie donc une dernière fois que toutes les features sont fonctionnelles en staging, puis on passe le tout en production.

3.3 Gestion de projet

3.3.1 Diagramme de GANTT

Afin de s'organiser au mieux pour les mois à venir, l'une des premières choses que nous avons fait est répertorier les tâches nécessaires aux deux lots (voir [A.1](#)). Une fois cette étape effectuée, nous avons transformé ce tableau en un diagramme de GANTT (voir [A.2](#)).

3.3.2 Jalons

Nous avons décidé, au vu du diagramme de GANTT, de travailler par jalons (milestones). Chaque milestone sera considérée comme une feature majeure, qui permet de segmenter le travail en blocs (afin de faciliter le travail de code review et de CI/CD de façon plus générale). Il y aura une milestone :

- à la fin du déploiement initial
- à la création de la DB
- à la création de la Back-End
- à la création des routes + des composants cores
- à la création de chaque page / groupe de pages

3.3.3 Répartition des tâches

En ce qui concerne la répartition des tâches, nous nous sommes accordés en fonction de nos compétences personnelles (voir [2.5.1](#)).

Ensemble, nous allons initialiser l'environnement de développement, la base de données (sa création ainsi que le remplissage des documents de données essentielles),

le front et le back end. Pour le front-end, beaucoup de travail est attendu, par conséquent, nous ne pourrons qu'essayer de se répartir équitablement les pages.

En ce qui concerne le back-end, **Alexandre** s'occupera déjà de la mise en place des buckets S3 avec AWS SDK. Il s'occupera aussi de NodeMailer, MulterJS et des requêtes mapbox avant de réaliser le premier déploiement. **Vincent** va s'occuper spécifiquement de la partie ODM (Object Document Manager) c'est à dire la création de Modèles, de Schémas. Il s'occupera aussi de la gestion des JWT. En ce qui concerne le front-end, la réalisation des routes ainsi que la redirection.

Modélisation

4.1 Identité du site

Cette section fait office de charte d'identité / charte graphique de notre plate-forme.

4.1.1 VroomMates

Nous avons choisi "VroomMates" afin de représenter la plateforme. Il s'agit d'un jeu de mots avec le mot "roommates", l'idée est de rappeler le principe de collocation et en quelques-sorte l'appliquer dans le monde du covoiturage.

4.1.2 Logo



FIGURE 4.1 – Différents logos de la plateforme

En ce qui concerne le logo, celui-ci est l'une des premières choses que nous avons faite au sein du groupe. Nous pensons que la réalisation d'un logo constitue en outre la base de notre charte graphique : toute l'identité visuelle y allait être forcément influencée. L'identité du site (voir 4.1.3) ainsi que le choix des couleurs (voir 4.1.4) s'est, de ce fait, décidé ici même. Nous avons aussi fait le choix de faire une version raccourcie, que nous allons utiliser en tant qu'image de profil sur la plateforme ou de favicon pour le client.

4.1.3 Identité du site

Nous voulons avoir un certain lien de proximité avec l'utilisateur et renvoyer une certaine image de jeunesse que l'on pourrait retrouver dans certaines start-ups connues ; Pour adopter ce style décontracté / jovial, nous avons décidé d'employer une police d'écriture arrondie et sans éperons, **Baloo 2** est ainsi un candidat idéal.

4.1.4 Palette de couleurs

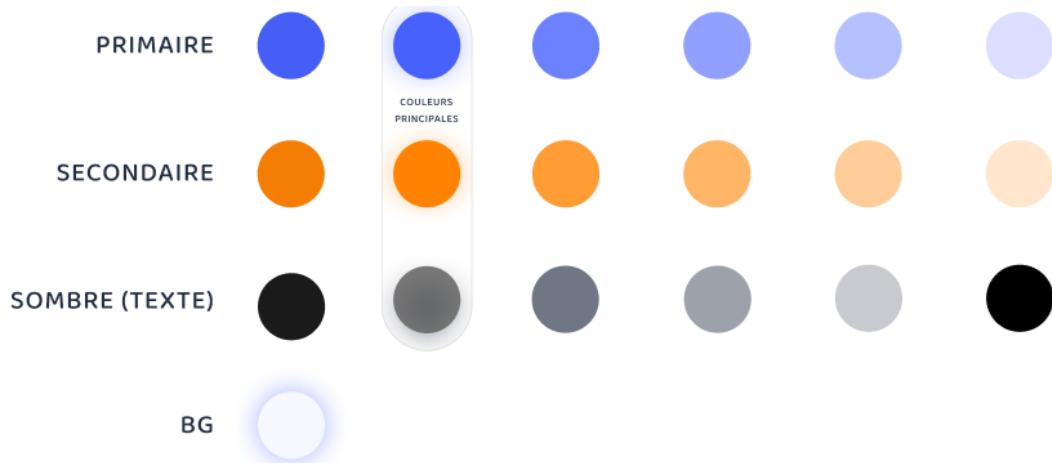


FIGURE 4.2 – Palette de couleur

En guise de palette de couleur, nous nous sommes inspirés de la dualité entre l'orange et le bleu, présents sur le site de Metz Numeric School. Ces deux couleurs opposées sont la base de la quasi-totalité de notre site, et nous avons créé plusieurs variantes de ces deux couleurs (sous formes de dégradés).

4.2 Diagrammes de séquence

A.5 - Authentification de l'utilisateur Ce diagramme prend en compte l'authentification de l'utilisateur au moment de *la création de son compte* ainsi qu'à *la connexion de ce dernier*.

A.6 - Participer à un voyage Cette partie regroupe *la recherche d'un voyage* et *la commande d'un voyage*. On se place du côté d'un utilisateur voulant utiliser la plateforme pour avoir un trajet.

A.7 - Partie conducteur Ici, nous nous intéressons à la procédure pour un utilisateur afin de *devenir conducteur* ainsi qu'à *la création et la mise en ligne d'un voyage*.

A.8 - Partie administrateur Les derniers cas principaux se situent dans la partie "back-office" (côté administrateur). Ils y figurent les cas de *consultation des statistiques du site*, de *consultation de la liste des utilisateurs* et de *consultation des demandes de conducteurs*.

4.3 Maquettage

Afin de réaliser l'étape la plus longue de ce lot, nous avons répertorié toutes les routes nécessaires et nous avons réalisé un diagramme nous permettant de correctement recenser toutes les pages du client.

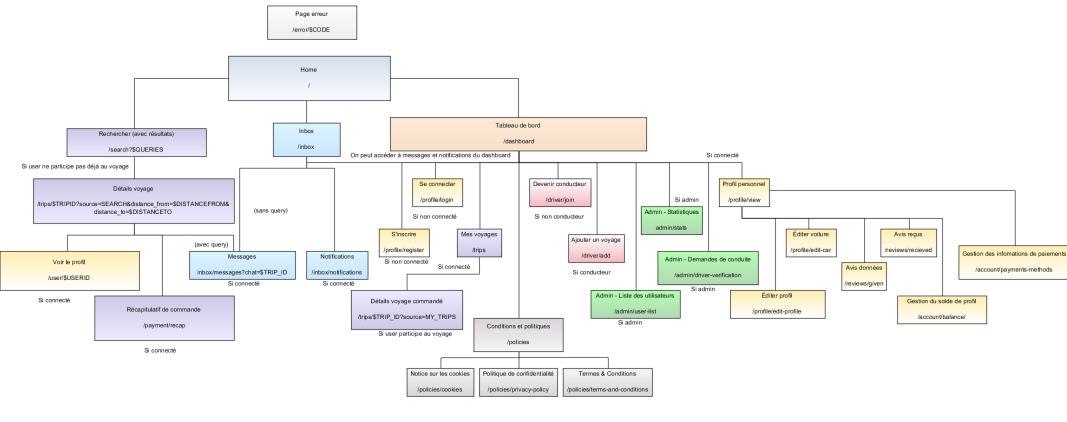


FIGURE 4.3 – Plan du site (voir en grand : [A.9](#))

Comme indiqué précédemment, le site étant "mobile first", nous nous sommes concentrés sur la partie mobile lors du maquettage. Ci-dessous, nous avons pris des extraits de ce qui représente au mieux notre site, d'un point de vue graphique. Néanmoins, la maquette dans son entièreté est disponible sur notre [Figma](#).

Nota Bene : Afin d'établir le copywriting intégral du site, les écrans mobile seront en anglais et les écrans ordinateur seront en français.

A.10 - Accueil Premièrement, nous avons établi la maquette de la page d'accueil, car c'est évidemment la première que l'utilisateur verra. La version mobile et la version ordinateur sont très similaires, afin de garder une ligne directrice et une identité similaire flagrante pour l'utilisateur dès le premier coup d'œil.

A.11 - Recherche Pour rechercher un trajet, il faut d'abord indiquer si c'est un trajet unique ou régulier. On rempli ensuite les 3 champs suivants : *adresse de départ*, *adresse d'arrivée*, et *date*. Les annonces s'affichent ensuite sous forme de prévisualisation les unes en dessous des autres, comprenant notamment les *deux adresses précédemment indiquées*, les *heures d'arrivée et de départ* et *les dates de répétition de ce trajet*. Il y a aussi évidemment *le prix* et des *indications sur le conducteur*. Graphiquement, nous avons voulu garder une interface claire pour l'utilisateur, tout en indiquant exclusivement les informations nécessaires pour chaque annonce.

A.12 - Détails En ce qui concerne les détails d'un voyage, l'interface est déjà légèrement plus complète : sur cette page sont indiqués tous les détails dont à besoin l'utilisateur. On note que la version sur ordinateur contient dans la même page à la fois la liste de résultats ainsi que les détails du voyage sélectionné.

A.13 - Profil utilisateur La réalisation sous forme de maquettes du profil est certainement celle qui nous aura causée le plus de soucis : on y trouve une grande quantité d'informations à faire apparaître à l'utilisateur, compliquant l'optimisation de l'expérience utilisateur. Pour se faire, une dizaine de maquettes ont été préalablement établies afin de satisfaire tous ces besoins ; c'est finalement sur une version colorée et intuitive que nous avons décidé de jeter notre dévolu. L'avantage de cette interface de visualisation est la possibilité de réutiliser cette dernière pour voir le profil d'un utilisateur ou bien même son propre profil (à condition d'y effectuer les changements adéquats).

A.14 - Messagerie instantanée La messagerie a été conçue pour être harmonieuse avec d'autres messageries classiques afin de faciliter la compréhension et l'utilisation pour des clients de n'importe quel niveau. De plus, c'est aussi ici que s'afficheront les notifications pour noter le conducteur.

Bilan du lot 1 et perspectives

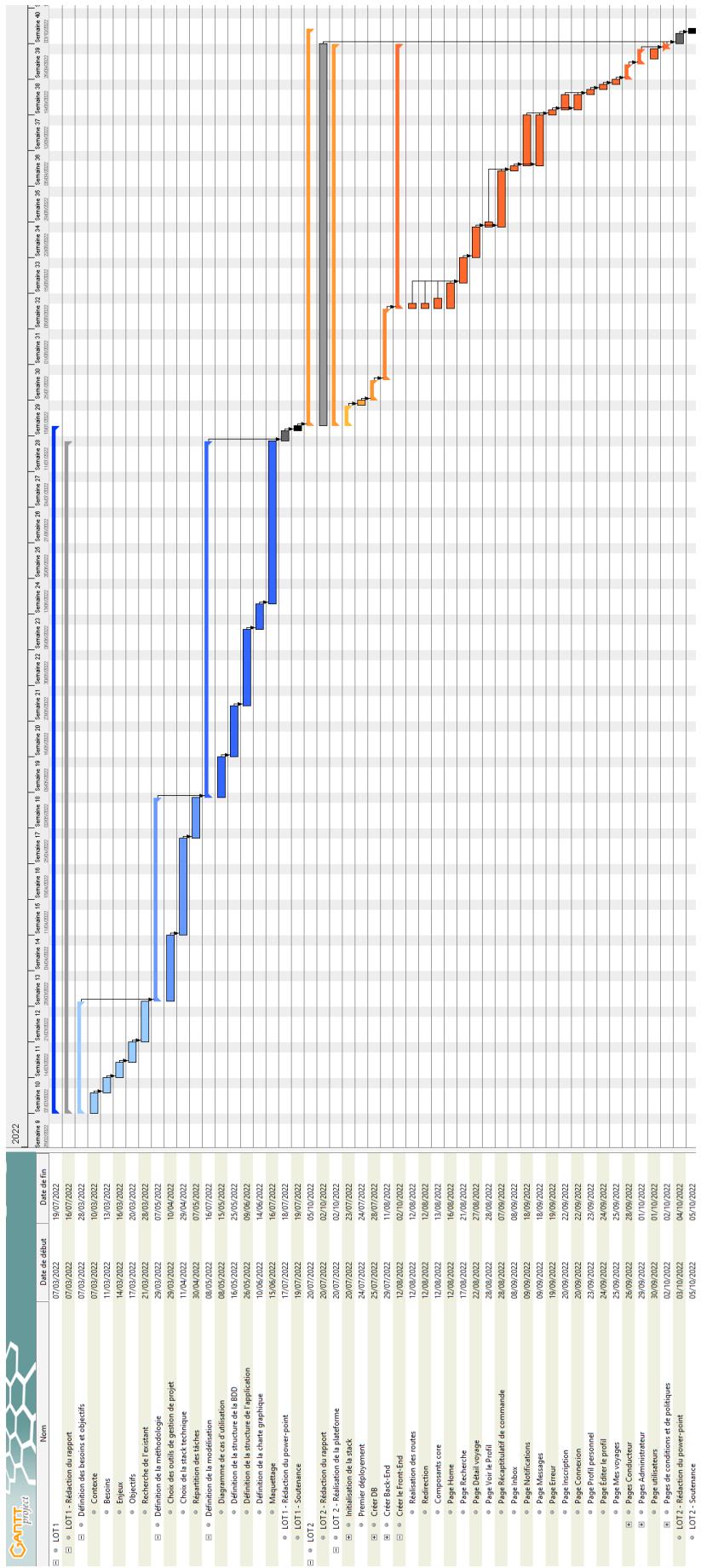
Finalement, la réalisation de ce premier lot mettant en place l'organisation minutieuse à respecter lors de la réalisation de ce projet nous aura permis de voir toutes les possibilités existantes pour créer une telle plateforme, et ainsi pouvoir les comparer afin de trouver la meilleure combinaison de technologies possible. Elle nous permet aussi de mieux organiser notre temps à l'avenir, pour ne pas s'éparpiller sur des tâches plutôt annexes, en laissant de côté certaines plus que nécessaires. C'est une véritable ligne directrice et règle de conduite à respecter à tout prix afin d'optimiser ce temps qui nous paraît déjà si bref.

Annexe

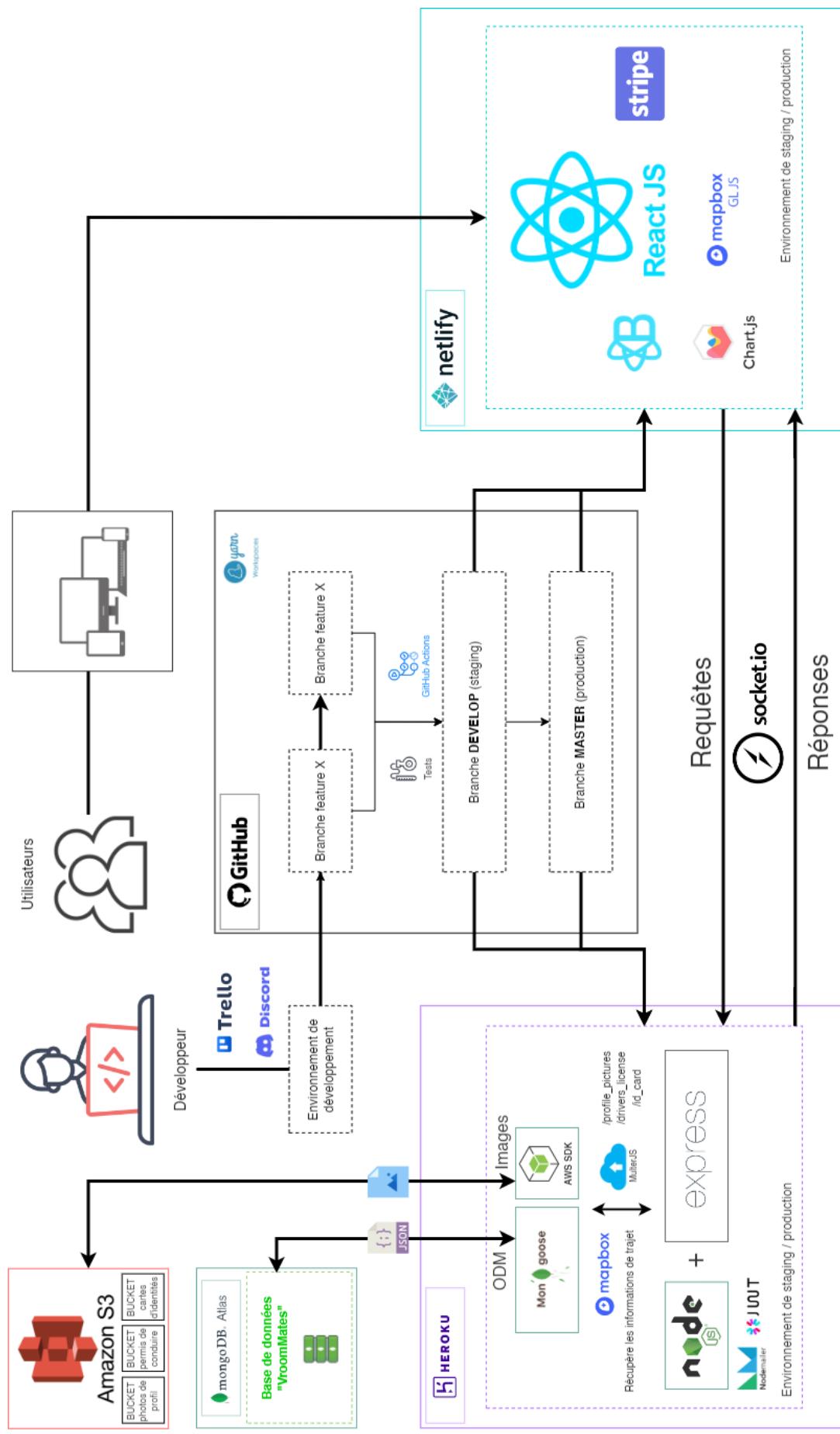
A.1 Définition des tâches et de leur dates

Nom	Date de début	Date de fin
LOT 1	07/03/2022	19/07/2022
LOT 1 - Rédaction du rapport	07/03/2022	16/07/2022
Définition des besoins et objectifs	07/03/2022	28/03/2022
Contexte	07/03/2022	10/03/2022
Besoins	11/03/2022	13/03/2022
Enjeux	14/03/2022	16/03/2022
Objectifs	17/03/2022	20/03/2022
Recherche de l'existant	21/03/2022	28/03/2022
Définition de la méthodologie	29/03/2022	07/05/2022
Choix des outils de gestion de projet	29/03/2022	10/04/2022
Choix de la stack technique	11/04/2022	29/04/2022
Répartition des tâches	30/04/2022	07/05/2022
Définition de la modélisation	08/05/2022	16/07/2022
Diagramme de cas d'utilisation	08/05/2022	15/05/2022
Définition de la structure de la BDD	16/05/2022	25/05/2022
Définition de la structure de l'application	26/05/2022	09/06/2022
Définition de la charte graphique	10/06/2022	14/06/2022
Maquettage	15/06/2022	16/07/2022
LOT 1 - Rédaction du power-point	17/07/2022	18/07/2022
LOT 1 - Soutenance	19/07/2022	19/07/2022
LOT 2	20/07/2022	05/10/2022
LOT 2 - Rédaction du rapport	20/07/2022	02/10/2022
LOT 2 - Réalisation de la plateforme	20/07/2022	02/10/2022
Initialisation de la stack	20/07/2022	23/07/2022
Initialiser env de dev	20/07/2022	23/07/2022
Initialiser DB	20/07/2022	23/07/2022
Initialiser Back-End	20/07/2022	23/07/2022
Initialiser Front-End	20/07/2022	23/07/2022
Premier déploiement	24/07/2022	24/07/2022
Créer DB	25/07/2022	28/07/2022
Créer la structure de la DB	25/07/2022	26/07/2022
Créer mock data dans la DB	27/07/2022	28/07/2022
Créer Back-End	29/07/2022	11/08/2022
Lien DB (mongoose)	29/07/2022	29/07/2022
Créer models	29/07/2022	31/07/2022
Créer routes	01/08/2022	04/08/2022
Protéger les routes avec JWT	05/08/2022	07/08/2022
Mise en place des buckets S3	08/08/2022	11/08/2022
Créer le Front-End	12/08/2022	02/10/2022
Réalisation des routes	12/08/2022	12/08/2022
Redirection	12/08/2022	12/08/2022
Composants core	12/08/2022	13/08/2022
Page Home	12/08/2022	16/08/2022
Page Recherche	17/08/2022	21/08/2022
Page Détail voyage	22/08/2022	27/08/2022
Page Voir le Profil	28/08/2022	28/08/2022
Page Récapitulatif de commande	28/08/2022	07/09/2022
Page Inbox	08/09/2022	08/09/2022
Page Notifications	09/09/2022	18/09/2022
Page Messages	09/09/2022	18/09/2022
Page Erreur	19/09/2022	19/09/2022
Page Inscription	20/09/2022	22/09/2022
Page Connexion	20/09/2022	22/09/2022
Page Profil personnel	23/09/2022	23/09/2022
Page Éditer le profil	24/09/2022	24/09/2022
Page Mes voyages	25/09/2022	25/09/2022
Pages Conducteur	26/09/2022	28/09/2022
Page Devenir Conducteur	26/09/2022	26/09/2022
Page Ajouter un voyage	27/09/2022	28/09/2022
Pages Administrateur	29/09/2022	01/10/2022
Page d'accueil	29/09/2022	29/09/2022
Page Demande de conduite	29/09/2022	01/10/2022
Page utilisateurs	30/09/2022	01/10/2022
Pages de conditions et de politiques	02/10/2022	02/10/2022
Page Contract d'utilisation	02/10/2022	02/10/2022
Page Politique de confidentialité	02/10/2022	02/10/2022
Page Politique de contenu	02/10/2022	02/10/2022
Page Notice sur les cookies	02/10/2022	02/10/2022
LOT 2 - Rédaction du power-point	03/10/2022	04/10/2022
LOT 2 - Soutenance	05/10/2022	05/10/2022

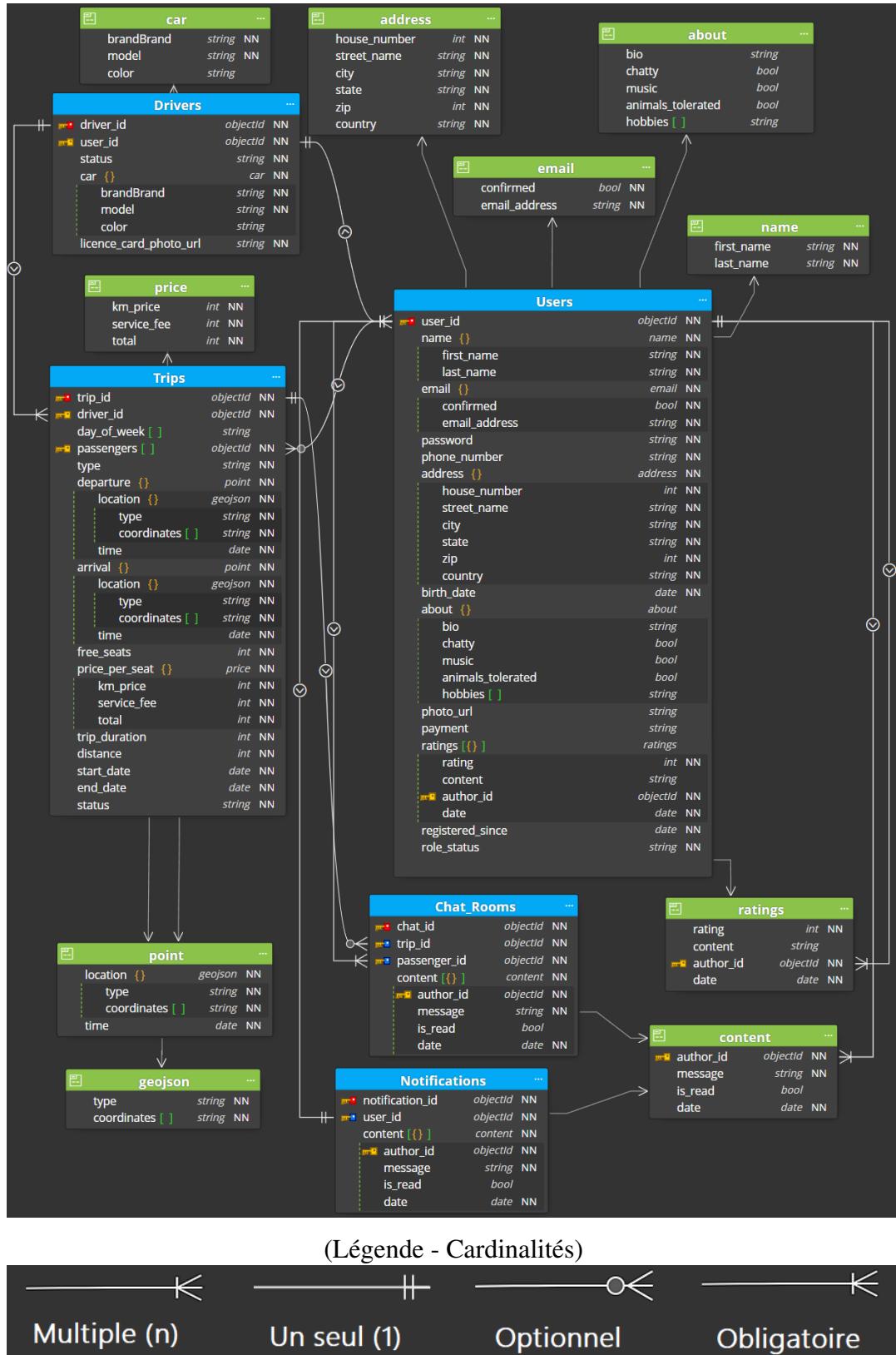
A.2 Diagramme de Gantt



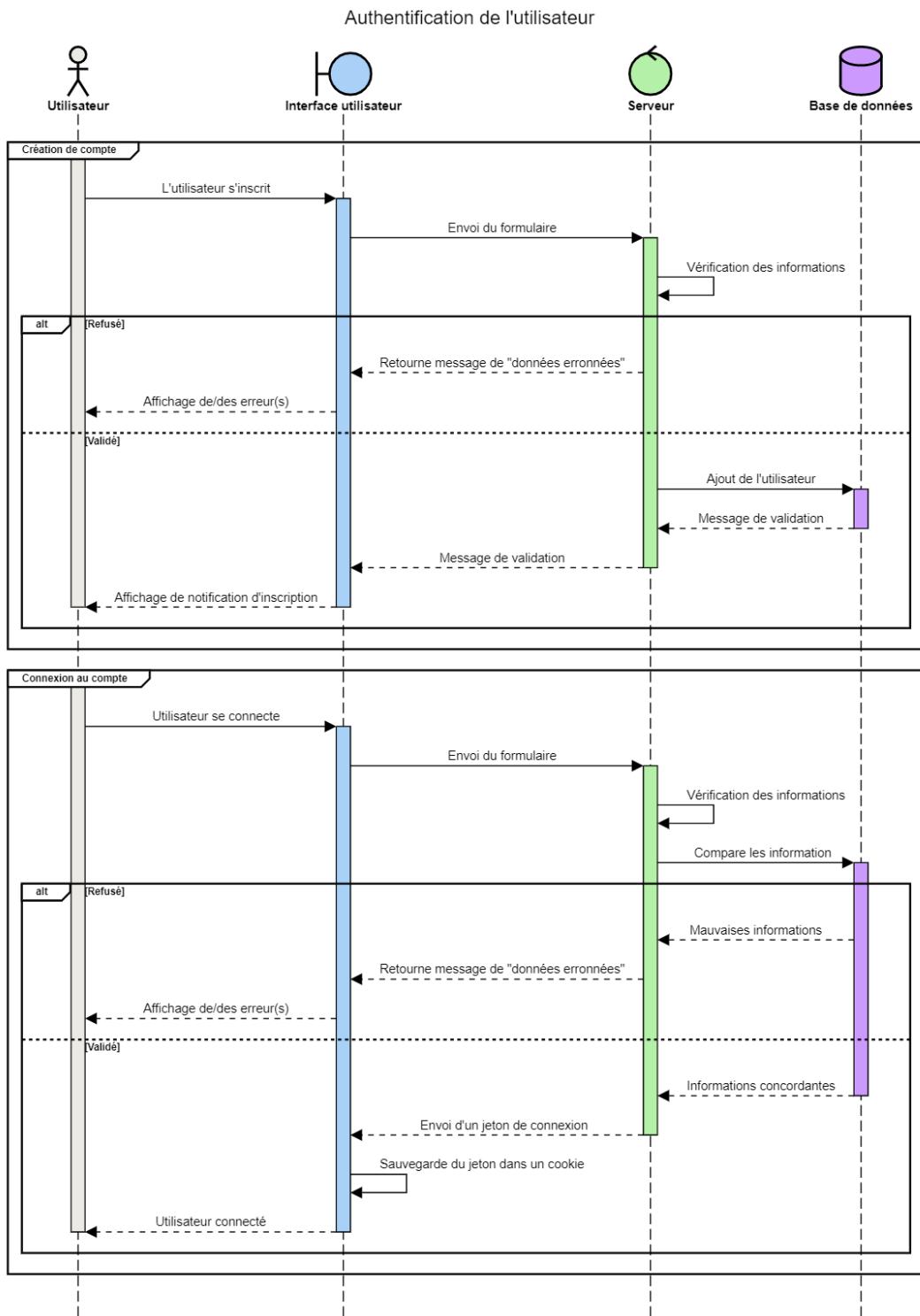
A.3 Stack technique complète du site



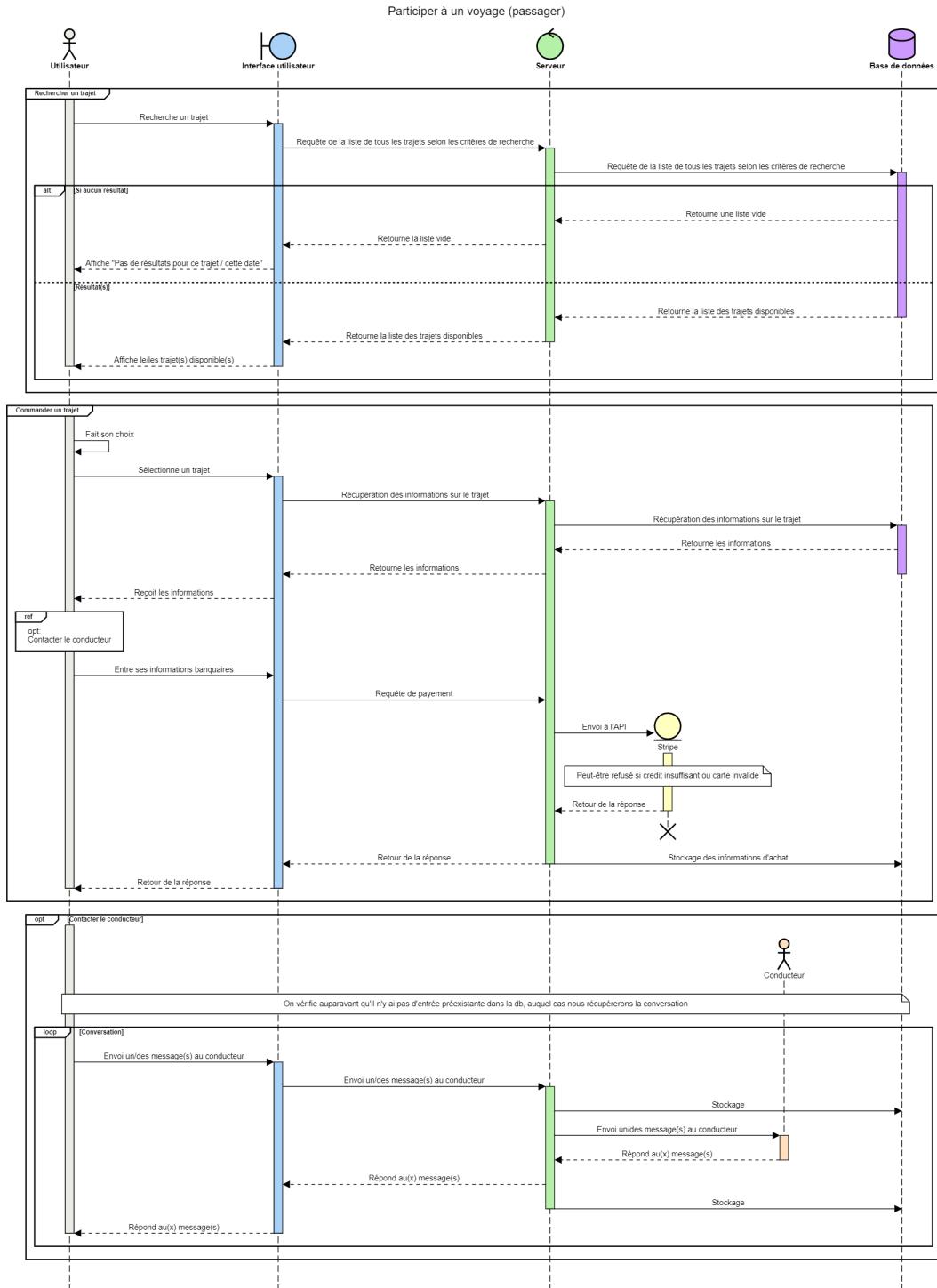
A.4 Modèle non-relationnel de la base de données



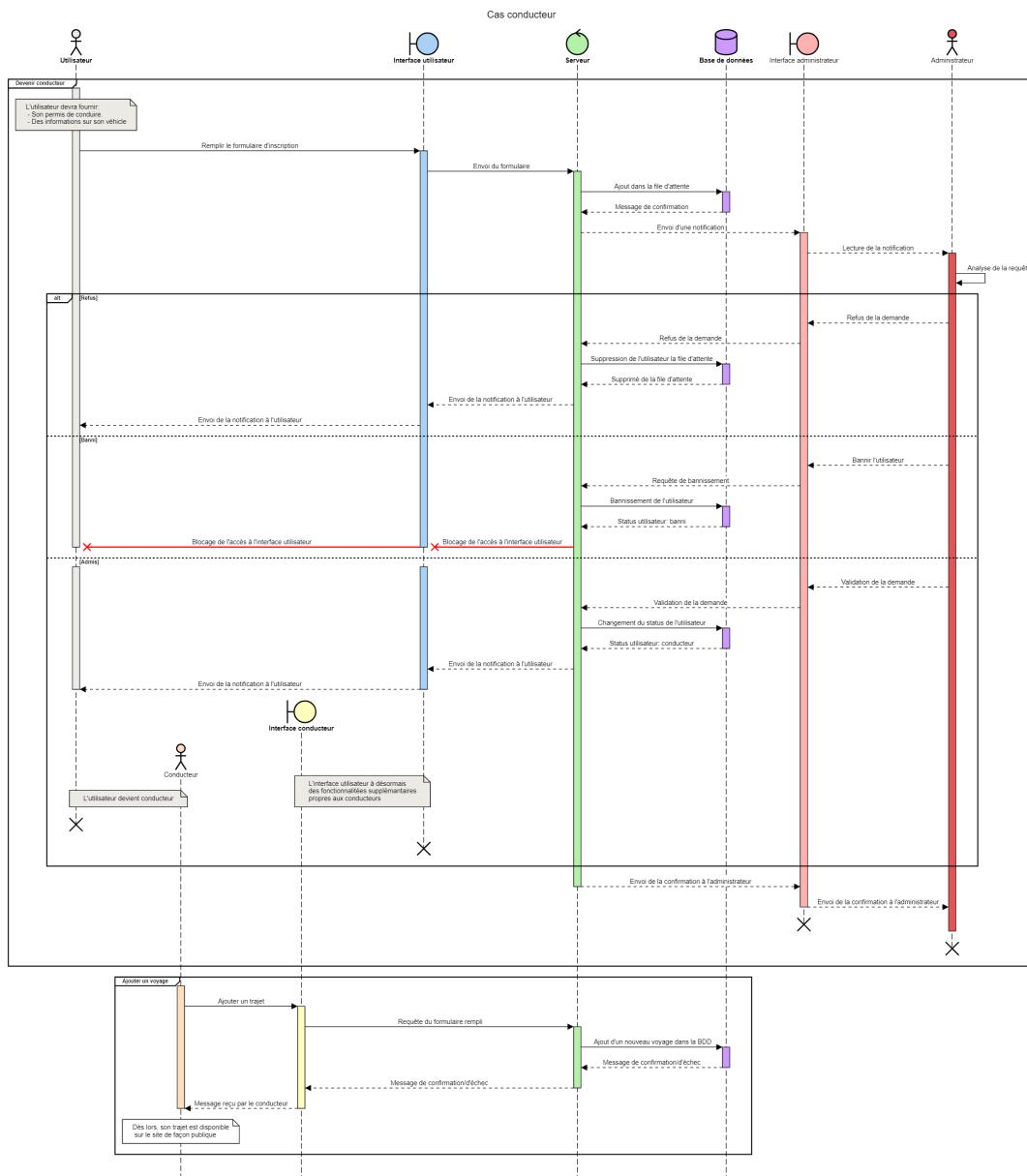
A.5 Diagramme de séquence : Authentification de l'utilisateur



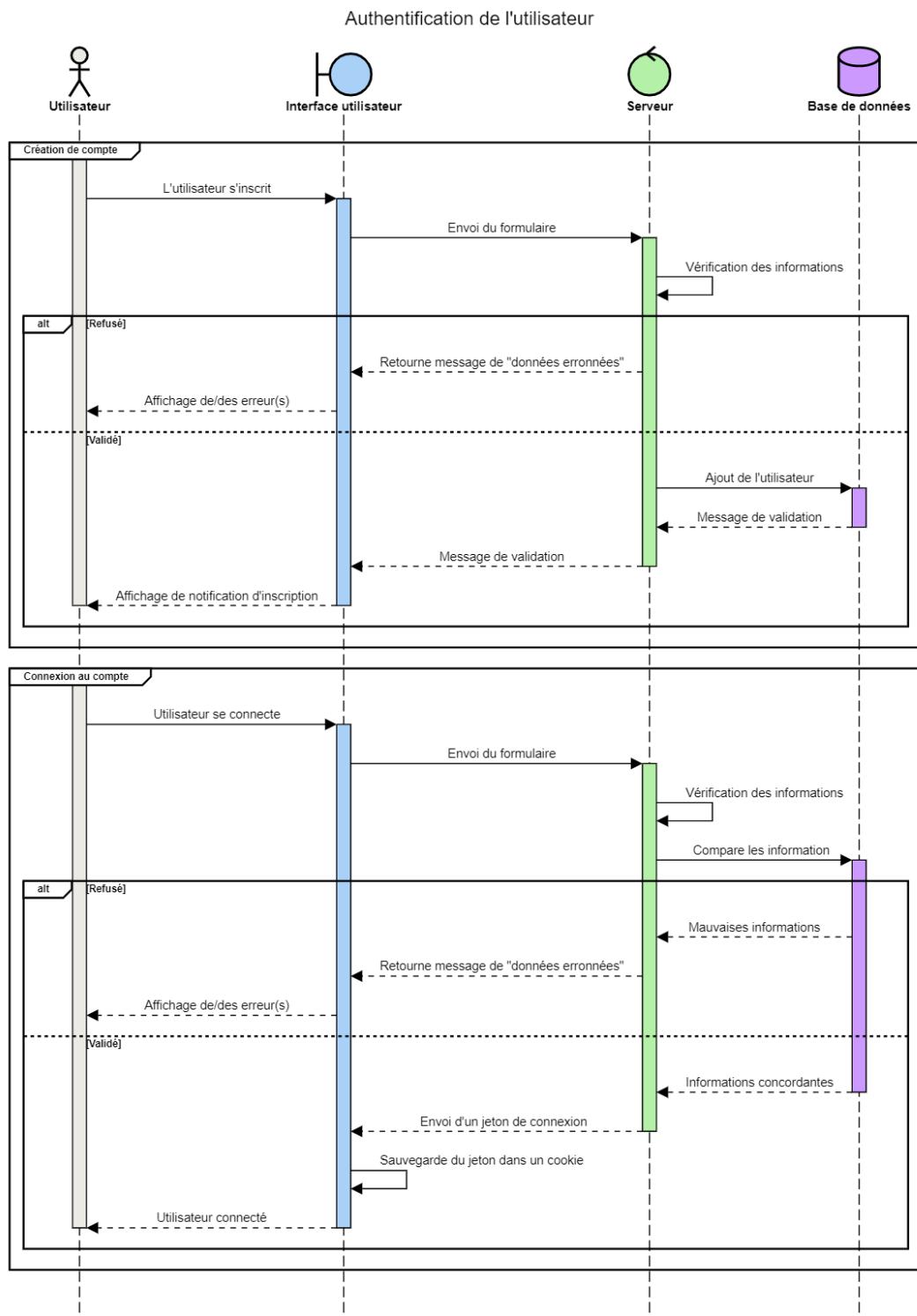
A.6 Diagramme de séquence : Participer à un voyage



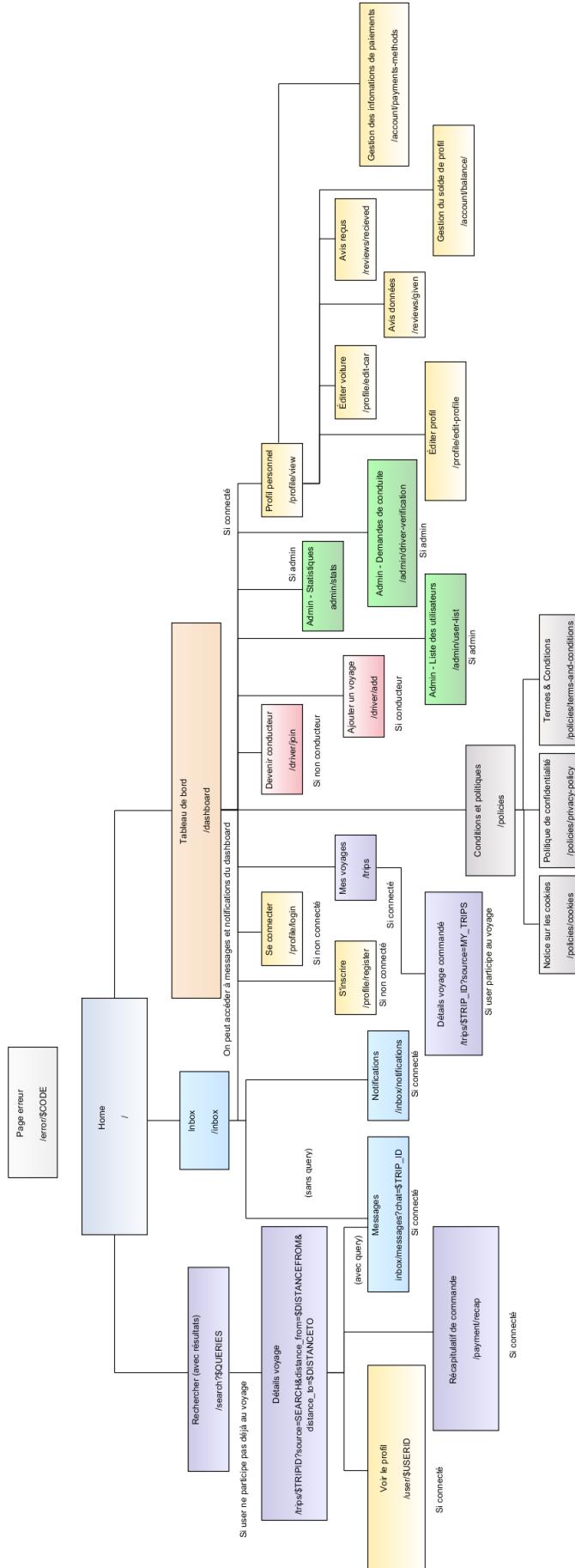
A.7 Diagramme de séquence : Partie conducteur



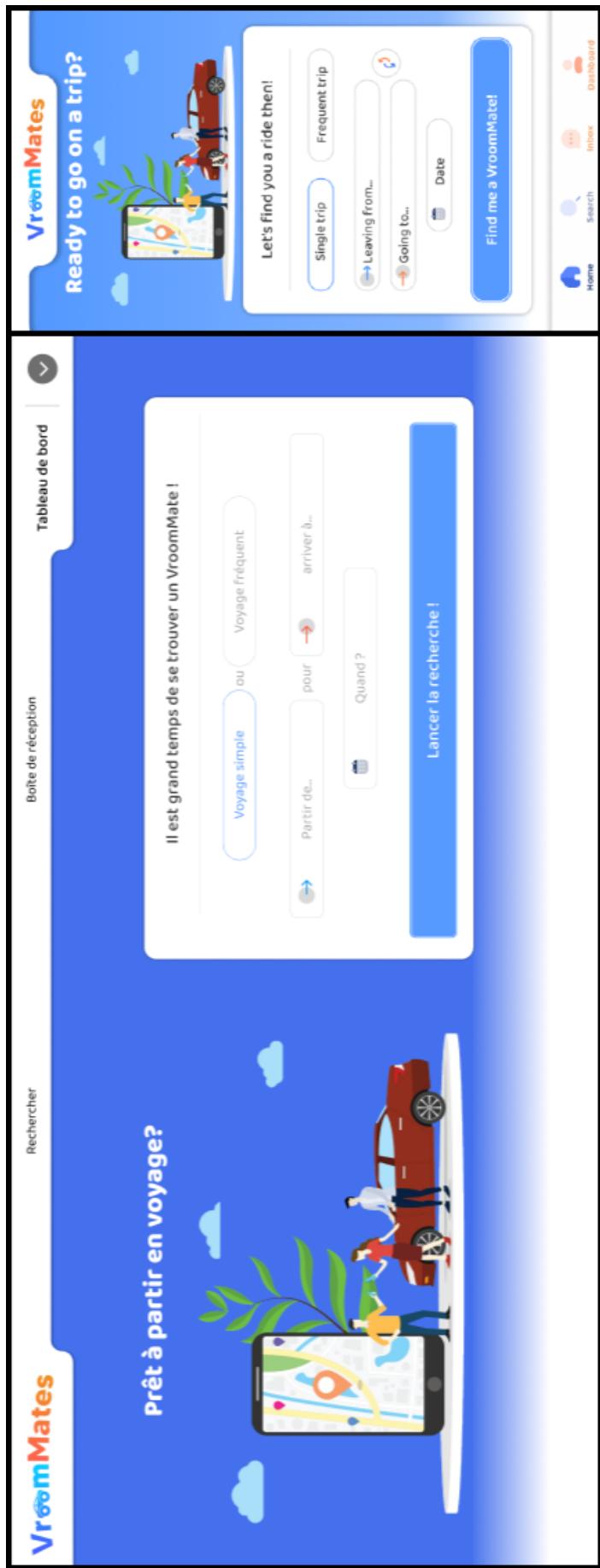
A.8 Diagramme de séquence : Partie administrateur



A.9 Plan du site



A.10 Maquettage : Accueil



A.11 Maquettage : Recherche

The image displays two screenshots of the VRoamMates app interface, one above the other, illustrating a search result for a carpool from Metz to Marseille.

Top Screenshot:

- Header:** VRoamMates
- Search Bar:** Metz, place de la Ré... → Marseille, rue de la...
14 June
1 passenger
- Profile Card (Jean):** Sat. 6 August 2022
Metz → Marseille
8:00 - 15:55 - 807 km
Jean **10\$**
- Profile Card (Michael):** Sat. 6 August 2022
Metz → Marseille
8:00 - 15:55 - 807 km
Michael **FULL**
- Bottom Section:** Tableau de bord, Boîte de réception, Rechercher
- Map View:** A detailed map of Metz showing the route from Metz to Marseille, which follows the Seille river through the city center.

Bottom Screenshot:

- Header:** VRoamMates
- Search Bar:** Metz, place de la Ré... → Marseille, rue de la...
14 June
1 passenger
- Profile Card (Alexandre):** Sat. 6 August 2022
Metz → Marseille
8H20 - 0h02 - 2.3 km
Alexandre **0.14€**
- Profile Card (Alexandre):** Sat. 6 August 2022
Metz → Marseille
8H20 - 0h02 - 2.3 km
Alexandre
- Bottom Section:** Home, Search, Inbox, Dashboard

A.12 Maquettage : Détails d'un voyage

The screenshot displays the VroomMates app interface. At the top, there's a header with the brand name "VroomMates". Below it, a navigation bar includes tabs for "Tableau de bord", "Boîte de réception", and "Rechercher". The main content area features a large map of Metz with a blue route line. A green polygon highlights a specific area, and a red arrow points from the text "vers Rue du père potot, Metz" to this area. On the right side of the map, there's a driver profile for "Alexandre" with a 5-star rating, a car icon, and a "Commander" button. The bottom section shows a table for vehicle availability and a summary of the trip details: "Début: Lundi 11 Avril 2022" and "Jusqu'à: Jeudi 15 Septembre 2022". It also includes a "Voir le profil" button and a "Contacter" button.

A.13 Maquettage : Profil d'un utilisateur

The wireframe illustrates the layout of a user profile page on the VRoomeMates platform.

Header: The top navigation bar includes the VRoomeMates logo, a search bar, and tabs for "Tableau de bord" and "Boîte de réception".

User Profile Section: This section displays the user's name, Alexandre, a profile picture, and ratings. It shows 245 friends and 260 rides, along with a 5-star rating icon.

Bio Section: Contains a bio text where Alexandre introduces himself as GÉRARD and expresses his desire to drive passengers. It also lists his age (21 years old) and gender (Male).

Hobbies Section: Shows interests such as driving, reading, cooking, and music.

Car Section: Details the user's vehicle, a Renault Mégane 3 Orange.

Reviews Section: A section titled "Avis" contains reviews from other users. One review from Alexandre G. is highlighted, stating: "Je ne comprend pas de quoi Vincent se parle, Alexandre doit être un des plus grands pilotes de la plateforme!" Another review from De Vincent G. says: "Il roule comme un dingue! Encore, je lui aurai mis au moins une étoile si il avait pas essayé déclasser une vieille dame sur le trottoir..." A third review from De Nicolas P. says: "Conduise convenable."

Footer: The footer features links for "Home", "Search", "Inbox", and "Dashboard".

A.14 Maquettage : Messagerie instantanée

34

La maquette de messagerie instantanée VroomMates est organisée en plusieurs sections :

- Barre de navigation (en bleu) :** Contient les icônes pour Home, Search, ... (trois points), Inbox et Dashboard.
- Section principale :** Titre "VroomMates" et sous-titre "Rechercher".
- Section Alexandre :** Titre "Alexandre" et sous-titre "METZ → METZ Jeu. 26 Janvier 2022, 16h30".
- Section Alexandre G. :** Titre "Alexandre G." et sous-titre "METZ → METZ Jeu. 26 Janvier 2022, 16h30".
- Section messages :** Affiche une liste de messages entre l'utilisateur et Alexandre G. et Alexandre.
- Section pied de page :** Icône d'envoi et champ "Enter a message".

Détails des messages :

- Message 1 (à Alexandre) :** "Hey, je n'ai pas eu encore le temps de vous remercier..." (le 26/01/2022 à 16h30).
- Message 2 (à Alexandre) :** "Vos messages peuvent être modérés. Plus d'informations [ici](#). (le 26/01/2022 à 16h30).
- Message 3 (de Alexandre) :** "Hello" (le 26/01/2022 à 16h30).
- Message 4 (à Alexandre) :** "Bonjour" (le 26/01/2022 à 16h30).
- Message 5 (de Alexandre) :** "Hello, where do you want to meet up?" (le 25/01/2022 à 20h15).
- Message 6 (à Alexandre) :** "Is it okay if I meet you at the bus stop?" (le 25/01/2022 à 20h17).
- Message 7 (de Alexandre) :** "Okay, I'll meet you there!" (le 25/01/2022 à 20h20).
- Message 8 (à Alexandre) :** "Le voyage est terminé, notez cette personne en cliquant ici!" (le 25/01/2022 à 20h20).
- Message 9 (de Alexandre) :** "The trip has ended, rate your trip by clicking here!" (le 26/01/2022 à 08h27).
- Message 10 (à Alexandre) :** "Hey, j'ai pas encore eu le temps de vous remercier pour le trajet. Je vous mettrai une bonne note, pour sûr!" (le 26/01/2022 à 16h30).
- Message 11 (de Alexandre) :** "Hey, I haven't got the chance to tell you thanks for the ride, I'll be sure to give you a good review!" (le 26/01/2022 à 16h30).

Bibliographie

- [1] Tomas FERNANDEZ. « What is monorepo ? (and should you use it ?) » In : *Semaphore Blog* (juin 2022). URL : <https://semaphoreci.com/blog/what-is-monorepo>.
- [2] MONGODB. « GeoJSON Objects ». In : *MongoDB Documentation* (avr. 2008). URL : <https://www.mongodb.com/docs/manual/reference/geojson/>.
- [3] European PARLIAMENT. « REGULATION (EU) 2016/679 ». In : *Official Journal of the European Union* (mai 2016). URL : <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679>.

Lot 1, par —

GERARD Alexandre : <https://github.com/agerard57>
GIANGRECO Vincent : <https://github.com/clevinjuna>