

**ROAD WARN,
Une application mobile d’alerte des nids-de-poule,
(LAVAKA) sur la route, en temps réel**

Par
CHARLES Elsy Soavita

Mémoire présenté
en vue de l’obtention du grade de Licence
en Informatique
option Développement d’application

Novembre, 2023

Jury :

Mme Baovola Rajaonarison, présidente
M. Steave Leong, examinateur
M. Rakotondralambo, encadreur professionnel

Table des matières

Liste des tableaux	iii
Liste des figures.....	iv
Glossaire	vi
Avant-propos	1
L'IT University.....	1
AZLink	2
Remerciements	2
Introduction	4
1 Présentation du projet	5
1.1 Objectifs du projet	5
1.2 Planning de réalisation.....	5
1.3 Technologies utilisées.....	8
1.4 Modèle de conception.....	11
1.5 Architecture	11
2 Réalisation du projet.....	16
2.1 Analyse et conception.....	16
2.1.1 Analyse de l'existant	16
2.1.2 Conception de l'application.....	16
2.2 Développement par fonctionnalité ou module.....	19
2.2.1 Login et Inscription	19
2.2.2 Profil et déconnexion.....	19
2.2.3 CRUD des utilisateurs	20
2.2.4 Affichage des trajets en liste.....	20
2.2.5 Création d'un trajet.....	21
2.2.6 Suppression d'un trajet	22
2.2.7 Duplication d'un trajet.....	22

2.2.8	Publication d'un trajet	23
2.2.9	Téléchargement d'un trajet.....	23
2.2.11	Détails et modification d'un trajet (avec vue de trajets sur map).....	24
2.2.12	Liste des obstacles	25
2.2.13	Enregistrement des obstacles sur un trajet.....	25
2.2.14	Alerte des obstacles sur un trajet	26
2.2.15	Représentation des obstacles sur carte et détail des obstacles avec les actions possibles sur l'obstacle (suppression, ...)	27
2.2.16	Notes d'un trajet	28
2.2.17	Commentaires d'un trajet	29
2.2.18	Suppression d'une note d'un trajet	29
2.2.19	Suppression d'un commentaire d'un trajet.....	30
2.2.20	CRUD des types d'obstacles	30
2.2.21	Modification de la configuration	30
2.3	État d'Analyse et Statistiques	31
2.3.1	Moyenne d'obstacles par kilomètre	31
2.3.2	Calendrier d'évolution des obstacles.....	32
2.3.3	Fiabilité d'un trajet par notes.....	32
2.4	Problèmes rencontrés et solutions.....	34
3	Évaluation du projet et connaissances acquises	35
3.1	Bilan pour l'entreprise.....	35
3.2	Bilan personnel	35
3.3	Extension et évolution de l'application.....	35
	Conclusion.....	37
	Bibliographie	38
	Annexe.....	i
	Etapas du déploiement du Back office	i

Extrait d'une fonction Arduino pour allumer une LED en continu.....	ii
---	----

Liste des tableaux

Tableau 1 : Tableau de synthèse des technologies	12
--	----

Liste des figures

Figure 1 : Diagramme de GANTT (1)	6
Figure 2 : Diagramme de GANTT (2)	7
Figure 3 : Diagramme de GANTT (3)	8
Figure 4 : Architecture de l'application	12
Figure 5 : MCD de PostgreSQL (tables)	17
Figure 6 : MCD de PostgreSQL (vues)	18
Figure 7 : MCD de SQLite (tables)	18
Figure 8 : Aperçu de login (front office)	19
Figure 9 : Aperçu d'inscription (front office)	19
Figure 10 : Profil et déconnexion	20
Figure 11 : liste des trajets en ligne	21
Figure 12 : Liste des trajets hors ligne	21
Figure 13 : Création d'un trajet : bouton ajouter	22
Figure 14 : Création d'un trajet : Mettre les informations	22
Figure 15 : Action sur un trajet : Télécharger	23
Figure 16 : Action sur un trajet : Supprimer, Publier, Dupliquer	23
Figure 17 : Détail d'un trajet : sélection d'un trajet (Front office)	24
Figure 18 : Détail d'un trajet : informations avec bouton modifier (Front office)	24
Figure 19 : Détail d'un trajet : informations (Back office)	25
Figure 20 : Enregistrement des obstacles : bouton enregistrer	26
Figure 21 : Enregistrement des obstacles : Enregistrement par clique	26
Figure 22 : Alerte des obstacles : bouton parcourir	27
Figure 23 : Alerte des obstacles : alertes	27
Figure 24 : Affichage sur map (Front office)	27
Figure 25 : Détail d'un obstacle (Front office)	27
Figure 26 : Confirmation de suppression d'un obstacle (Front office)	27
Figure 27 : Affichage sur map, Détail d'un obstacle, Suppression d'un obstacle (Back office)	28
Figure 28 : Note d'un trajet : Bouton " Noter "	28
Figure 29 : Note d'un trajet : Liste des notes	28
Figure 30 : Commentaire d'un trajet : bouton commenter	29
Figure 31 : Commentaire d'un trajet	29

Figure 32 : Modification de configuration de l'application	30
Figure 33 : Confirmation de configuration de l'application.....	30
Figure 34 : Moyenne d'obstacles par kilomètre.....	31
Figure 35 : Évolution des obstacles par mois	32
Figure 36 : Fiabilité des trajets par note	33
Figure 37 : Installation de PostgreSQL 14	i
Figure 38 : Modification de configuration postgresql.conf.....	i
Figure 39 : Configuration de pg_hba.conf	i
Figure 40 : configuration de tomcat-users.xml	i
Figure 41 : configuration de manager.xml	ii
Figure 42 : Extrait de fonction pour allumer une LED Arduino	ii

Glossaire

API (Application Programming Interface) : ensemble de règles et de spécifications permettant aux logiciels de communiquer entre eux.

Back-end : partie qui gère la logique, les calculs et les interactions avec la base de données.

Back Office : partie d'un système informatique qui permet à l'organisation de l'administrer et de le gérer. Il s'occupe de la partie gestion qui regroupe la partie métier et données.

CSS ou (Cascading Style Sheets) : langage de feuilles de style utilisé pour décrire la façon dont le contenu est présenté à l'utilisateur. Il est utilisé pour contrôler la présentation, le formatage et la mise en page du contenu HTML.

CRUD (Create Read Update Delete) : Il s'agit de l'opération qui permet de créer, de lire ou afficher, de modifier ou mettre à jour et de supprimer ou effacer les entrées dans la base de données.

Framework : ensemble de composants logiciels et d'outils d'une application.

Front-end : partie d'un programme informatique responsable de l'interface utilisateur, permettant à l'utilisateur d'interagir avec un Back-end.

Front Office : partie d'un système informatique qui prend en charge l'interface d'une application. Il est visible par l'utilisateur et lui permet d'interagir avec le reste du site.

GPS (Global Positioning System) : système de navigation par satellite qui fournit des informations sur la localisation et le temps, quelles que soient les conditions météorologiques.

Hot Reloading : fonctionnalité de développement permettant d'appliquer les modifications de code à une application en cours d'exécution sans avoir à redémarrer l'application à son début.

HTML (ou HyperText Markup Language) : langage de balisage standard pour la création de pages web.

IOT (Internet of Things) : Connectivité des objets physiques à internet et entre eux afin d'échanger des données créant un écosystème intelligent et interconnecté.

JavaScript ou JS : langage de programmation qui permet de modifier le contenu HTML, (pour contrôler le comportement des éléments HTML, pour créer des animations, pour envoyer des requêtes au serveur, pour manipuler les données, etc.).

JDBC (Java Database Connectivity) : moyen pour les applications Java d'interagir avec les bases de données relationnelles.

JSP (JavaServer Pages) : technologie utilisée pour développer des applications web. Elle permet aux développeurs de concevoir une structure HTML et d'y intégrer des fragments de code Java.

LED (Light Emitting Diode), ou Diode Émettrice de Lumière : composant électronique qui émet de la lumière lorsqu'une tension électrique lui est appliquée.

Login : processus par lequel un utilisateur accède à un système informatique en fournissant des identifiants, généralement au moyen d'un nom d'utilisateur et d'un mot de passe.

Map : représentation visuelle d'un espace géographique.

MCD (Modèle Conceptuel de Données) : représentation logique de l'organisation des informations et leurs relations.

Open-Source : dans le contexte de logiciel, "Open-Source" signifie que le code du logiciel est accessible au public : tous les utilisateurs peuvent voir, modifier et distribuer le code source.

POO (Programmation Orienté Objet) : paradigme de programmation qui utilise des "objets" et leurs interactions pour concevoir des applications et des programmes informatiques. Les objets sont des instances de classes, qui sont essentiellement des modèles définis par le programmeur. Chaque objet peut avoir des attributs (aussi appelés propriétés) et des méthodes (aussi appelés fonctions).

SQL (Structured Query Language) : langage normalisé utilisé pour gérer et manipuler les données.

Template : outil de programmation qui facilite et accélère le processus de création de sites web grâce aux éléments déjà préconçus et intégrés et donc faciles à modifier.

UI (User Interface) : concerne les expressions et la facilité d'action de l'internaute avec

l'interface web.

URL (Uniform Resource Locator) : référence à une ressource spécifiant son emplacement sur un réseau informatique et un mécanisme pour la récupérer.

UX (User Experience) : concerne l'interaction et le design de l'interface graphique qui permet le dialogue homme-machine.

.

Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d'études de Licence en Informatique de l'IT University, stage effectué à AZLink durant 3 mois, d'août à novembre 2023.

Afin de situer avec précision le contexte dans lequel ce mémoire a été élaboré, je vais présenter succinctement, d'une part, l'IT University, et, d'autre part, AZLink.

L'IT University

Fondée en 2011, l'IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers, de préférence scientifiques :

- En trois ans, pour l'obtention d'une :
 - Licence, option Développement, Réseaux et Bases de Données ou Web et Design.
 - Licence, option Graphic Design ou Communication Digitale.
- En cinq ans, pour l'obtention d'un :
 - Master MBDS en coopération avec l'Université Côte d'Azur située à Sophia Antipolis, Nice, France.
 - Master BIHAR en coopération avec l'ESTIA situé à Bidart, près de Bayonne (Pays Basque), France.

Étant une formation professionnalisante, l'ITU a tissé des liens forts avec ses partenaires industriels, dont l'opérateur convergent TELMA et la plupart des entreprises et institutions du secteur des TIC¹. Ces partenaires participent effectivement à la formation par la fourniture d'une connexion Internet à haut débit, par l'envoi de conférenciers et par l'accueil d'étudiants en stage. Beaucoup de ces partenaires recrutent aussi les nouveaux diplômés dès leur sortie de l'École.

Par ailleurs, le corps enseignant de l'ITU est constitué en totalité de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.

¹

TIC : Technologies de l'Information et de la Communication

AZLink

AZLink est une entreprise de services numériques (ESN), fondée en mars 2020 par Rakotondralambo. Son siège social est basé à Ankaraobato; AZLink a établi sa réputation en devenant un acteur clé dans les domaines de l'informatique et de la domotique.

La société propose une gamme diversifiée de services, notamment :

- La vente et la prestation de services en informatique.
- L'électronique et l'énergie.
- Les services de transport.
- La construction et la réhabilitation de maisons.
- L'informatique et la domotique.

AZLink se distingue par son engagement pour l'innovation et l'excellence dans ces divers domaines.

L'un des services phares d'AZLink est l'installation de maisons connectées. Grâce à son expertise en domotique, AZLink se positionne comme un leader dans la création de solutions innovantes pour les maisons intelligentes. Cette offre de services est un exemple de la manière dont AZLink s'efforce de rester à l'avant-garde de l'économie, en offrant des solutions qui répondent aux besoins de ses clients.

En plus de ses services de domotique, AZLink offre également des services de transport, de construction et de réhabilitation des maisons. Cette entreprise s'efforce de développer constamment de nouvelles technologies et de nouvelles méthodes pour améliorer ses services.

Remerciements

Je tiens à exprimer ma profonde gratitude à l'IT University, où j'ai acquis des connaissances inestimables en informatique au cours de ces trois dernières années. Un grand merci à AZLink pour m'avoir offert l'opportunité d'effectuer mon stage et d'appliquer ces connaissances dans un contexte réel.

Je suis particulièrement reconnaissant envers Monsieur Rojo Rabenanahary, mon encadreur

pédagogique, dont les directives ont été essentielles dans mon parcours académique et professionnel. Un remerciement spécial à mon encadreur professionnel, Monsieur Rakotondralambo, pour ses précieux conseils et son soutien constant pendant le stage.

Je tiens également à remercier mes condisciples de l'IT University ainsi que ma famille pour leur soutien et leurs encouragements constants. Vous avez tous joué un rôle crucial dans mon parcours et je vous en suis reconnaissant.

Merci pour votre contribution inestimable à ce travail.

Introduction

L'essentiel du travail que j'ai réalisé a porté sur le thème :

“ROADWARN, une application mobile d'alerte de nids-de-poule (LAVAKA), sur la route, en temps réel”. Cette application permet d'améliorer la capacité d'éviter les obstacles en informant immédiatement les automobilistes de la présence d'obstacles sur leur trajet. Il s'agit aussi de réduire les dommages matériels des véhicules causés par ces obstacles présents sur les routes, notamment les nids-de-poule.

L'application mobile en question n'a pas encore été développée au sein de l'entreprise. Il s'agit donc d'une nouvelle application pour AZLink.

Il est important de présenter les résultats suivants :

- L'implémentation d'une fonctionnalité d'alerte en temps réel, fonctionnant à la fois en ligne et hors ligne :
 - avec l'utilisation d'une LED.
 - avec l'appareil mobile uniquement
- La mise en œuvre d'une fonctionnalité d'enregistrement des obstacles sur les routes, en temps réel, fonctionnant à la fois en ligne et hors ligne.
- Le déploiement du côté administrateur (back-office) de l'application sur un serveur en ligne.
- Le partage des détails d'un trajet spécifique d'un utilisateur, y compris les obstacles qu'il a rencontrés, avec d'autres utilisateurs.
- Le téléchargement des trajets publiés par les utilisateurs.

Ce mémoire est organisé en trois sections principales :

- ❖ La présentation du projet.
- ❖ La réalisation du projet.
- ❖ L'évaluation du projet et des connaissances acquises.

1 Présentation du projet

1.1 Objectifs du projet

L'objectif principal de ce projet, l'application mobile "ROAD WARN", est d'offrir aux conducteurs un meilleur confort dans la conduite de leur voiture tout en réduisant les dommages matériels et les coûts de réparation de leur véhicule.

Cette application vise à lancer des alertes en temps réel pour aider les conducteurs à éviter les obstacles sur la route, à réduire les risques d'accidents et à améliorer la qualité de la conduite.

1.2 Planning de réalisation

Pour la réalisation du projet, j'ai élaboré un planning précis, mais qui, au fil du temps, a connu des changements pour tenir compte des demandes de l'entreprise. Voici le diagramme de GANTT détaillant les tâches planifiées durant ces 3 mois de stage :

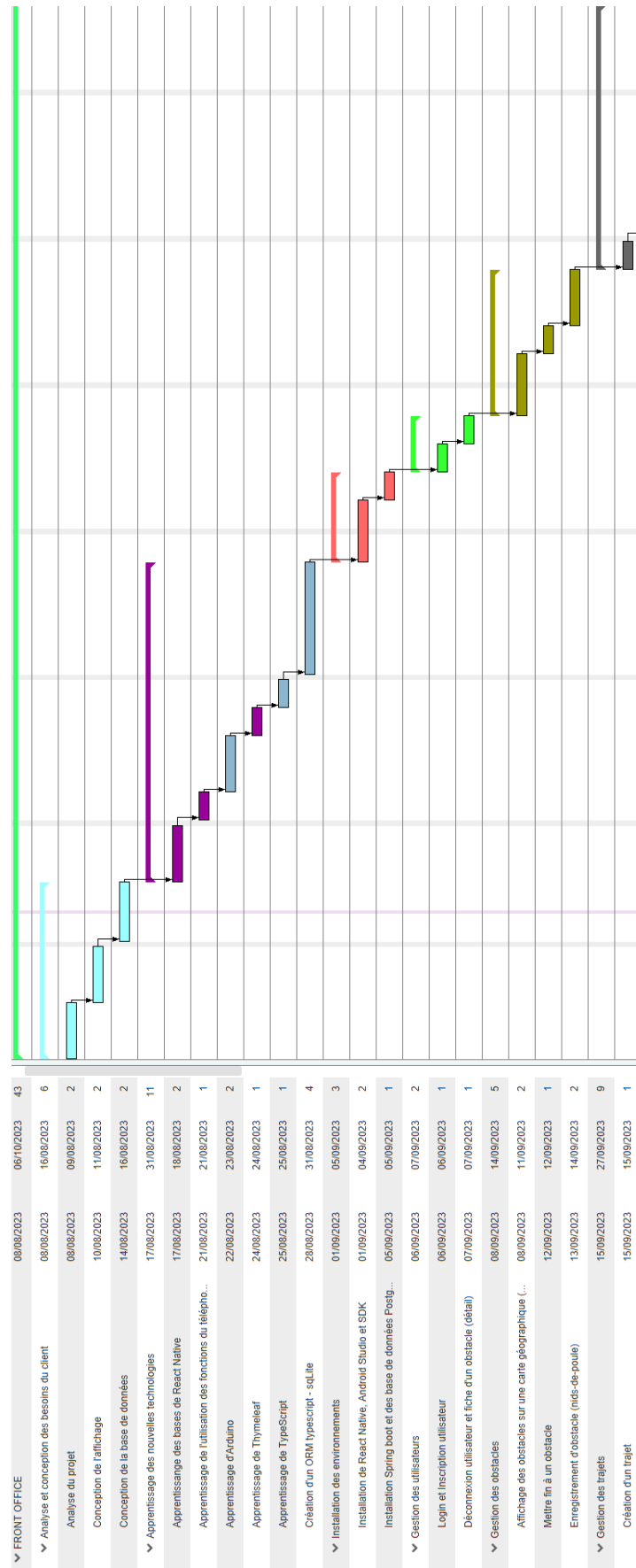


Figure 1 : Diagramme de GANTT (1)

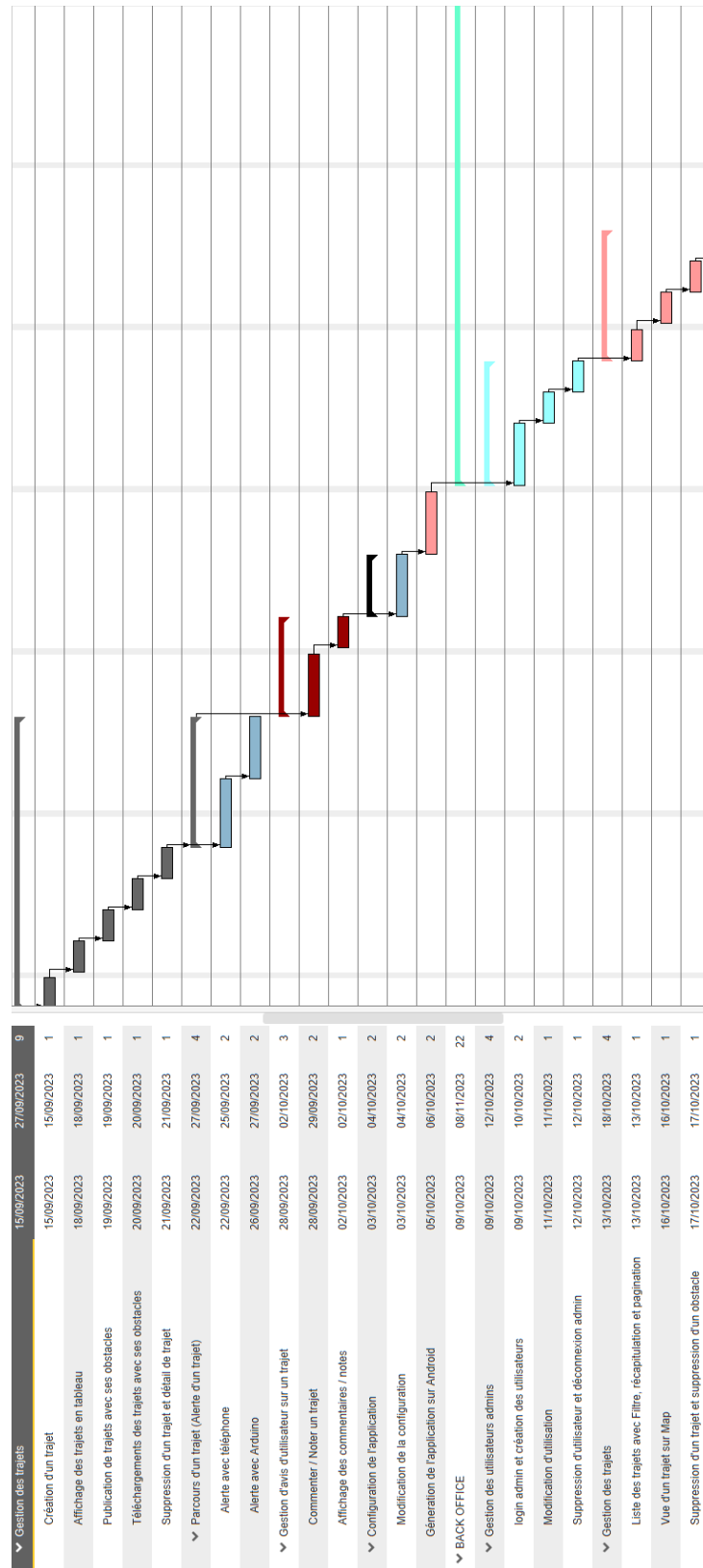


Figure 2 : Diagramme de GANTT (2)

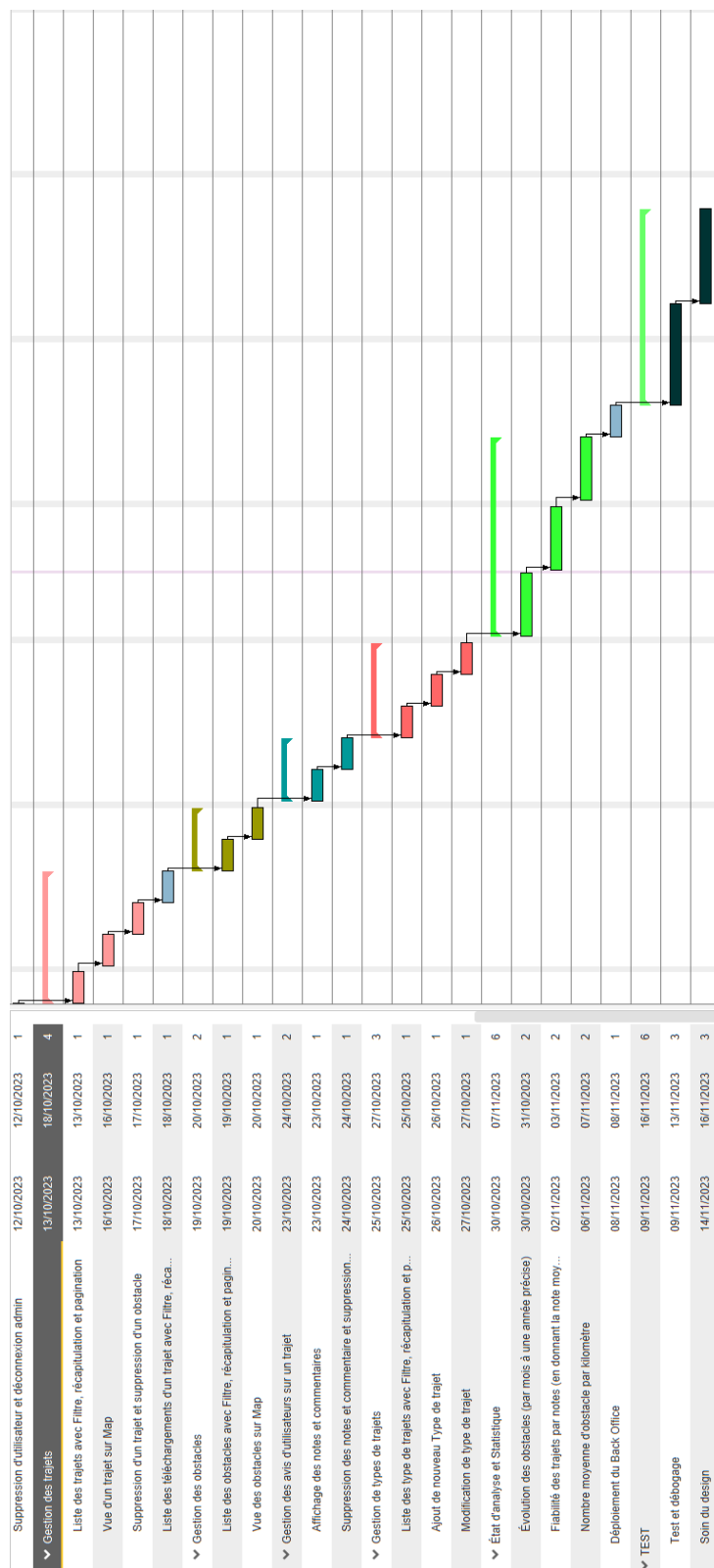


Figure 3 : Diagramme de GANTT (3)

1.3 Technologies utilisées

1.3.1 Technologies

- **Front-end**

- **Bootstrap (version 5) :**

Bootstrap est un framework CSS open-source conçu pour le développement web front-end. Il est largement utilisé pour créer des sites web responsives, car il peut s'adapter à différents appareils et tailles d'écrans (ordinateur de bureau, tablette, smartphone). Ce framework utilise trois technologies : le **HTML**, conçu pour la création de page web, le **JS** pour la modification des éléments et des contenus HTML, et le **CSS** pour le style.

- **Thymeleaf (version 2.7.14) :**

Thymeleaf est un moteur de template moderne écrit en Java dans les environnements web pour la génération des vues (modèle HTML) dans les applications web.

- **React Native (version 0.72.4) :**

React Native est un framework open-source conçu pour le développement des applications mobiles natives sur iOS et Android avec la même base de code JavaScript utilisant la librairie React.js.

- **TypeScript (version 4.8.4) :**

TypeScript est un langage de programmation open source basé sur JavaScript, mais avec des fonctionnalités de typage statique au langage. Ce langage a été choisi parce qu'il offre une meilleure organisation et une meilleure compréhension de code.

- **Back-end**

- **Java (version 11.0.20.1) :**

Java est un langage de programmation orienté objet, créé par Sun Microsystems et repris par Oracle, utilisé dans le développement des

applications mobiles, web et de bureau, et de beaucoup d'autres encore.

- **Spring Boot (version 2.7.14) :**

Spring Boot est un framework Java open-source conçu pour le développement de la partie serveur des applications. Ce framework a été utilisé dans ce projet pour sa capacité de créer des services web.

- **Le langage Arduino (version 1.8.19) :**

Le langage Arduino est basé sur le langage C++, mais il a été modifié pour être plus facile à utiliser. Il a été conçu pour être simple à comprendre et à utiliser. Le langage Arduino a été conçu pour être utilisé avec l'environnement de développement intégré (IDE) Arduino, qui est un logiciel gratuit et open source.

- **Base de données**

- **SQLite (version 6.0.1) :**

SQLite est un système de gestion de base de données relationnelle, légère, open-source et facile à intégrer dans des applications ; c'est pourquoi, dans ce projet, il est utilisé pour stocker des données des trajets et des obstacles dans l'appareil mobile.

- **PostgreSQL (version 14.9) :**

PostgreSQL est un système de gestion de base de données relationnelle open-source et conforme aux normes SQL. Ce système de base de données s'intègre facilement avec d'autres technologies, comme Spring Boot, ce qui convient bien à ce projet.

- **Tableau comparatif de PostgreSQL et MySQL**

Critères	PostgreSQL	MySQL
Compatibilités SQL	PostgreSQL répond à 160 des 179 fonctionn	MySQL n'implémente pas l a norme SQL complète. Il s'

	alités principales de SQL.	agit donc d'un SGBD partiellement conforme à SQL.
Type de données	PostgreSQL fournit un support étendu des types de données.	MySQL est limité pour le support des types de données comme les types de données pour les hachages.

- **Serveur d'application**

- **Apache Tomcat (version 9.0.82) :**

Apache Tomcat est un serveur d'applications qui est utilisé pour exécuter du code Java dans un environnement du serveur Web. Il est capable d'exécuter différentes spécifications, telles que Java Servlet, JavaServer Pages (JSP), Java EL et WebSocket.

1.4 Modèle de conception

L'application utilise le modèle MVC (Model - View - Controller), modèle de conception de logiciel qui sépare une application en trois composants principaux :

- **Le modèle :** la couche qui gère les données et la logique métier de l'application.
- **Le contrôleur :** Cette couche, qui gère la logique de l'application. Elle reçoit les requêtes de l'utilisateur et interagit avec le modèle en utilisant les fonctions spécifiques du modèle pour traiter ces requêtes.
- **La vue :** la couche qui gère l'interface utilisateur. Elle affiche les données à l'utilisateur.

1.5 Architecture

L'application est structurée en deux composantes principales :

- **Front office :** Cette partie est développée en utilisant React Native, qui est responsable de l'invocation des fonctions métiers de l'application mobile. Ces fonctions métiers, écrites en TypeScript, gèrent le traitement des données, y compris l'appel des API (hébergées sur le back-end Spring Boot). Ces

opérations peuvent impliquer des actions sur la base de données SQLite du téléphone mobile ou sur la base de données PostgreSQL en ligne. Une fois que les données ont été traitées par TypeScript, React Native gère la présentation des données sur l'écran de l'appareil mobile. Pour la gestion de la LED, React Native fait appel à une API qui active la lumière via Arduino.

- Back office :** Cette partie est développée en utilisant Spring Boot Thymeleaf. Elle utilise les fonctions hébergées sur le serveur Spring Boot pour le traitement des données. Il est important de noter que le serveur Spring Boot récupère les données de la base de données PostgreSQL en ligne.

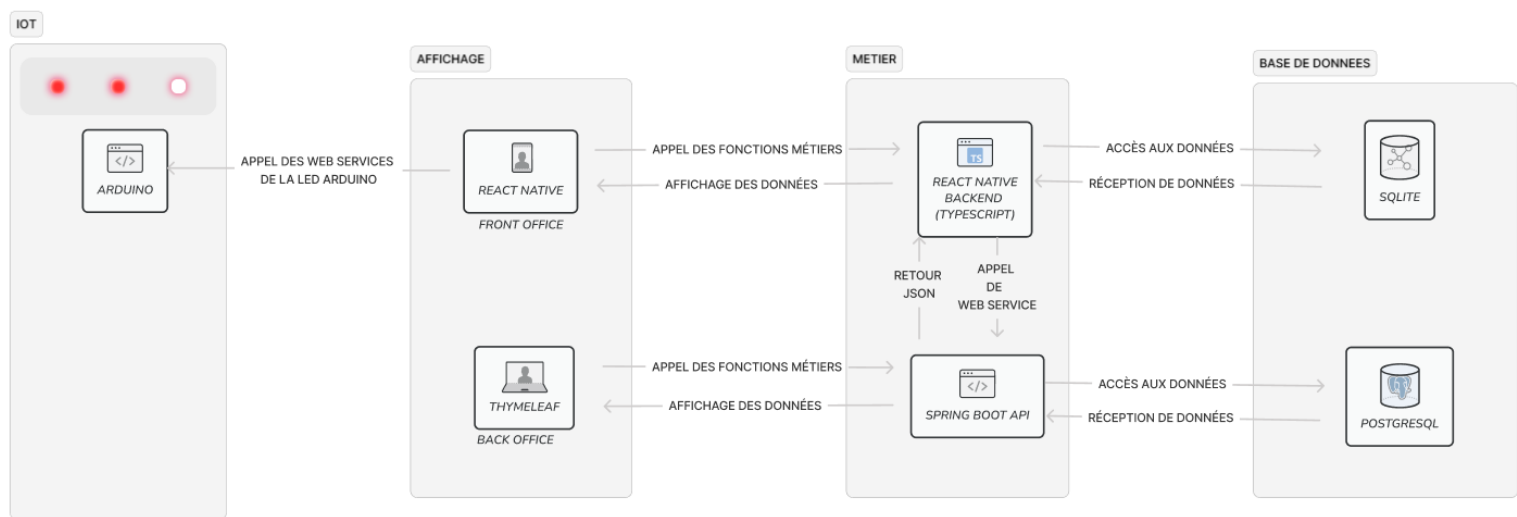


Figure 4 : Architecture de l'application

1.5.1 Tableau de Synthèse

Tableau 1 : Tableau de synthèse des technologies

Technologie	Avantages	Inconvénients
Bootstrap	<ul style="list-style-type: none"> Grille réactive : l'un des principaux avantages de Bootstrap est son système de grille réactive. Compatibilité : Bootstrap est largement compatible avec tous les principaux navigateurs et appareils. 	<ul style="list-style-type: none"> Personnalisation limitée : bien que Bootstrap offre un certain nombre de composants et de classes prédéfinis, il peut être difficile de personnaliser ces éléments pour qu'ils correspondent exactement aux besoins de l'utilisateur.

Thymeleaf	<ul style="list-style-type: none"> ● Prise en charge de divers modèles : il prend en charge divers modèles de données, y compris les objets Java, les listes, les cartes, etc. ● Syntaxe lisible : Thymeleaf utilise une syntaxe similaire à celle du HTML, qui est facile à lire et à comprendre. 	<ul style="list-style-type: none"> ● Moins de fonctionnalités avancées : par rapport à certains autres moteurs de templates, Thymeleaf ne peut pas disposer de certaines fonctionnalités avancées. Par exemple, avec Thymeleaf, il n'est pas possible d'écrire du code Java dans le HTML.
React Native	<ul style="list-style-type: none"> ● Rapidité de développement : React Native simplifie le processus de développement en permettant aux développeurs de réutiliser des composants prédéfinis. ● Accès aux périphériques mobiles : React Native offre un accès facile aux périphériques mobiles tels qu'appareil photo, accéléromètre, etc., y compris le GPS, ce qui est important pour ce projet. ● Hot Reloading 	<ul style="list-style-type: none"> ● Limitations d'Accès aux Fonctionnalités : bien que React Native offre un accès aux fonctionnalités des périphériques mobiles, certaines fonctionnalités ne peuvent pas être aussi bien prises en charge que dans le développement natif. ● Limitations d'UI/UX : React Native ne supporte pas les animations complexes.
TypeScript	<ul style="list-style-type: none"> ● Meilleure maintenance de code : grâce au typage statique, le code TypeScript est plus facile à comprendre et à maintenir, ce qui rend la collaboration avec d'autres développeurs plus efficace. 	<ul style="list-style-type: none"> ● Peut prendre du temps : TypeScript nécessite une étape de compilation avant d'être exécuté. Cela peut prendre du temps pour faire cette compilation en fonction du nombre de fichiers.
Java	<ul style="list-style-type: none"> ● Développement multiplateforme : le code Java peut être exécuté sur tous les systèmes d'exploitation qui supportent Java. 	<ul style="list-style-type: none"> ● Utilisation de mémoire : Java dispose de nombreux modules qui facilitent le développement d'applications, mais cela nécessite beaucoup de mémoire.
Spring Boot	<ul style="list-style-type: none"> ● Facilité de configuration : il simplifie la configuration des applications Java sans passer du temps à configurer manuellement chaque composant de l'application. Si, par exemple, nous avons besoin d'une base de données dans notre application, il suffit, avec Spring 	<ul style="list-style-type: none"> ● Dépendance aux versions : Spring Boot gère les versions des dépendances, mais cela peut aussi être un inconvénient. Si une nouvelle version d'une bibliothèque incluse dans Spring Boot est incompatible avec une

	<p>Boot, d'inclure les propriétés dans "application.properties" ou "application.yml".</p> <ul style="list-style-type: none"> ● Service web : simple à configurer, facile à développer, flexible et robuste. ● Gestion des dépendances : l'un des avantages majeurs de Spring Boot réside dans sa gestion des dépendances, facilitée par l'utilisation de l'outil de gestion de projet Maven ou Gradle. 	<p>application existante, la mise à jour peut poser des problèmes.</p> <ul style="list-style-type: none"> ● Taille de l'application : les applications Spring Boot peuvent avoir une certaine taille par rapport aux bibliothèques et aux dépendances incluses.
Langage Arduino	<ul style="list-style-type: none"> ● Facile à apprendre : Le langage Arduino est conçu pour être facile à comprendre et à utiliser. 	<ul style="list-style-type: none"> ● Moins de fonctionnalités : Les microcontrôleurs Arduino ont moins de fonctionnalités que de nombreux autres types de microcontrôleurs, ce qui peut limiter ce que vous pouvez faire avec eux.
SQLite	<ul style="list-style-type: none"> ● Léger : SQLite est un système de base de données légère, parfaite pour les applications mobiles. ● Facile à intégrer avec React Native : SQLite est facile à intégrer avec React Native, vu qu'il existe des bibliothèques, telles que "react-native-sqlite-storage", qui simplifient l'utilisation de SQLite dans les applications React Native. 	<ul style="list-style-type: none"> ● Taille des données limitée : en raison de limitations de la taille des fichiers, SQLite peut ne pas être adapté pour les applications nécessitant de stocker d'énormes volumes de données. SQLite ne peut stocker que 281 téraoctets au maximum.
PostgreSQL	<ul style="list-style-type: none"> ● Licence Open-Source : PostgreSQL est gratuit, contrairement à Oracle (par exemple), qui est un produit commercial avec des coûts de licence élevés. ● Facile à intégrer avec Spring Boot : PostgreSQL s'intègre facilement avec Spring Boot en raison de son support JDBC natif. ● Performant : PostgreSQL offre de bonnes performances et peut gérer de 	<ul style="list-style-type: none"> ● Diminution de la performance en écriture intensive : bien que PostgreSQL offre d'excellentes performances dans la plupart des cas d'utilisation, il peut être moins performant sur les gros volumes de données avec les opérations d'écriture intensives, en particulier si la configuration n'est pas optimisée correctement.

	grandes quantités de données.	
Apache Tomcat	<ul style="list-style-type: none"> ● Léger et performant : Apache Tomcat est très efficace dans l'utilisation de la mémoire et des autres ressources de la machine. ● Multiplateforme : Apache Tomcat peut fonctionner sur tous les systèmes d'exploitations qui supportent Java. 	<ul style="list-style-type: none"> ● Pas de déploiement à chaud prêt à l'emploi : Apache Tomcat ne supporte pas le déploiement à chaud prêt à l'emploi, ce qui signifie qu'il faut toujours redémarrer le serveur pour avoir une nouvelle version de l'application.

2 Réalisation du projet

2.1 Analyse et conception

2.1.1 Analyse de l'existant

Actuellement, l'entreprise n'a pas encore mis en œuvre une application de ce type. Bien qu'il existe des applications similaires, telles que "Intents Go" ou "City Rover", qui ne sont plus disponibles dans les stores (App Store et Play Store). Suite à une prise en compte de tous les services proposés par AZLink, il s'avère que l'entreprise offre des services de transport. Or, les véhicules de l'entreprise subissent fréquemment des dommages nécessitant des réparations, en particulier après des trajets effectués de nuit. Par conséquent, le développement d'une application d'alerte de nids-de-poule en temps réel sera une aide précieuse pour réduire le nombre de ces dommages matériels et donc les coûts de maintenance des véhicules. C'est dans cette perspective que l'application "Road Warn" a été conçue et développée.

2.1.2 Conception de l'application

La conception est une étape très importante dans tout projet, car avec un projet bien détaillé, il est plus facile de poser les bases solides sur lesquelles le reste du processus est construit. C'est l'étape où l'on transforme des idées en plans concrets et détaillés. Voici pourquoi il faut d'abord concevoir l'application avant de la développer :

- La conception donne une vision claire et détaillée du projet.
- Elle permet d'économiser du temps, car ainsi le projet est bien détaillé et il est plus facile d'estimer le temps nécessaire à l'accomplissement d'une tâche.

Après la conception de l'application, les tâches réalisées ont été réparties de la manière suivante :

- Front office :
 - Nombre de tables créées : 7
 - Nombre de classes : 7

- Nombre d'écrans :14 dont 3 sont des modales.
- Back office :
 - Nombre de tables créées : 7.
 - Nombre de vues créées : 5.
 - Nombre de classes : 32 dont 15 sont des contrôleurs et 17 des modèles.
 - Nombre d'écrans : 23 dont 11 sont des modales.

Ci-après la figure représentant le modèle conceptuel de données ou MCD :

- PostgreSQL :

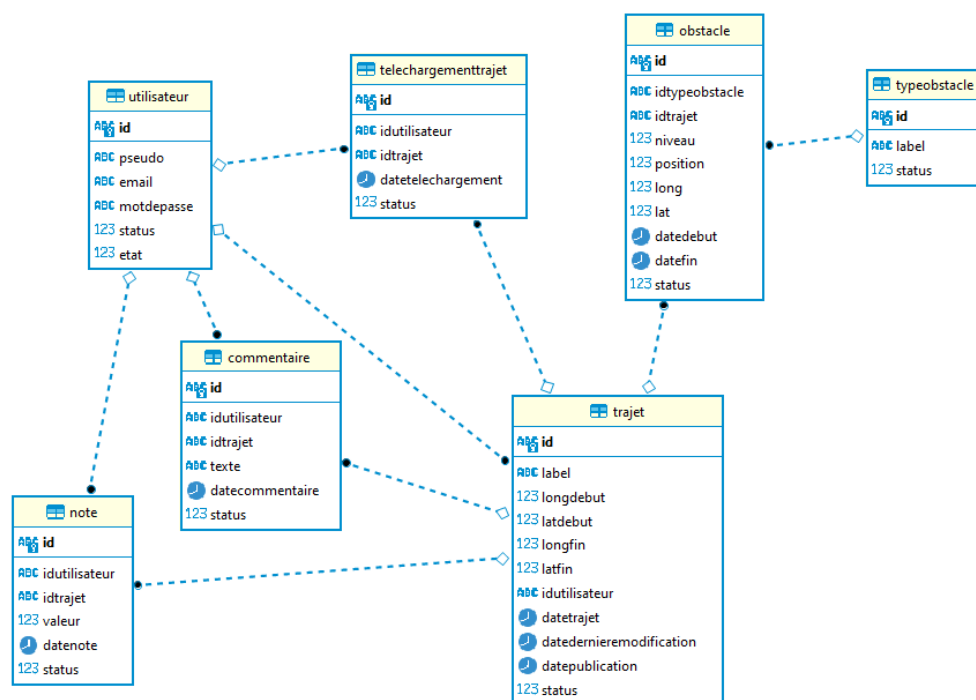


Figure 5 : MCD de PostgreSQL (tables)

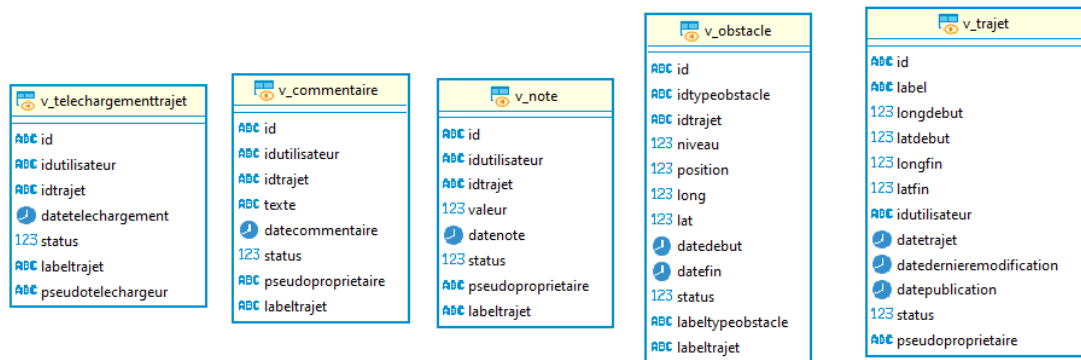


Figure 6 : MCD de PostgreSQL (vues)

- SQLite :

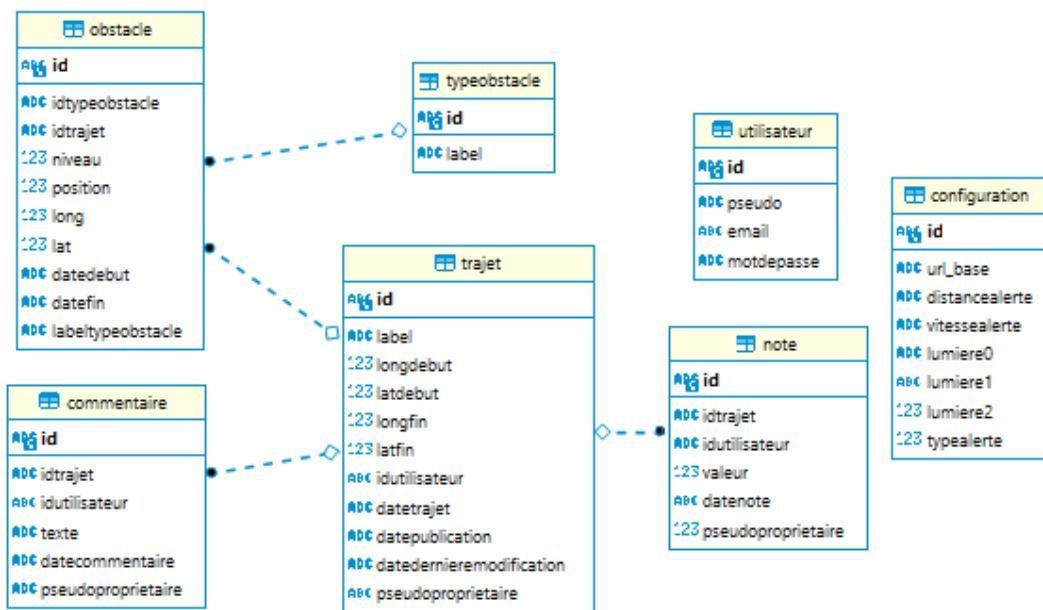


Figure 7 : MCD de SQLite (tables)

2.2 Développement par fonctionnalité ou module

2.2.1 Login et Inscription

Le login a été développé pour pouvoir sécuriser les données des trajets et des obstacles présents sur ces trajets. Si l'utilisateur n'a pas encore été inscrit à l'application, il peut créer un compte en passant par l'inscription.

Connexion



The login form features the 'road warn' logo at the top. Below it are two input fields: 'Email' and 'Mot de passe'. A blue button labeled 'Se connecter' is positioned below the password field. At the bottom, there is a link labeled 'S'inscrire'.

Inscription



The registration form features the 'road warn' logo at the top. Below it are four input fields: 'Pseudo', 'Email', 'Mot de passe', and 'Confirmer votre mot de passe'. A blue button labeled 'S'inscrire' is positioned below the confirmation field. At the bottom, there is a link labeled 'Se connecter'.

Figure 8 : Aperçu de login (front office)

Figure 9 : Aperçu d'inscription (front office)

2.2.2 Profil et déconnexion

Le profil permet d'afficher les informations du profil de l'utilisateur une fois connecté. Elle offre la possibilité de se déconnecter, pour terminer la session active de l'utilisateur. Ce module a été développé pour le côté administrateur ainsi que le côté client.

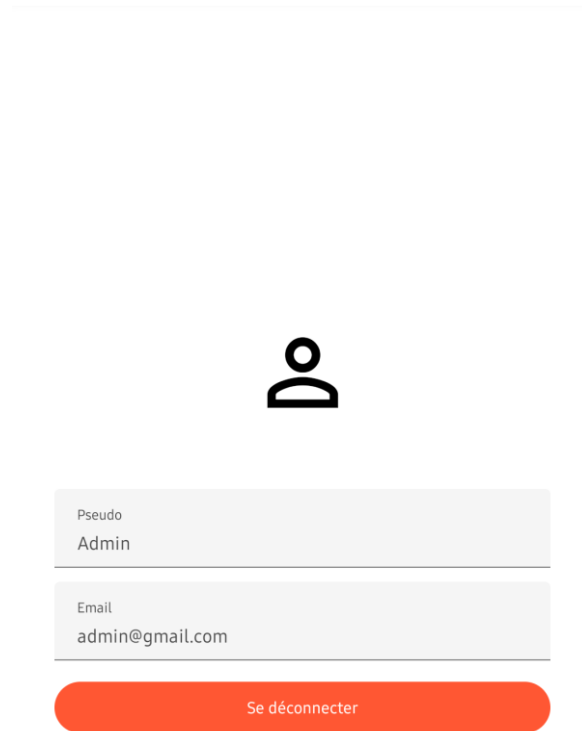


Figure 10 : Profil et déconnexion

2.2.3 CRUD des utilisateurs

Ce module, dédié au Back office, permet la création de comptes utilisateurs, offre la possibilité de convertir un utilisateur standard en administrateur et permet la suppression de comptes.

2.2.4 Affichage des trajets en liste

Cette fonctionnalité permet à l'utilisateur de visualiser la liste des trajets. Ce module a aussi été développé pour le côté administrateur, qui a ainsi un récapitulatif de tous les trajets des utilisateurs.

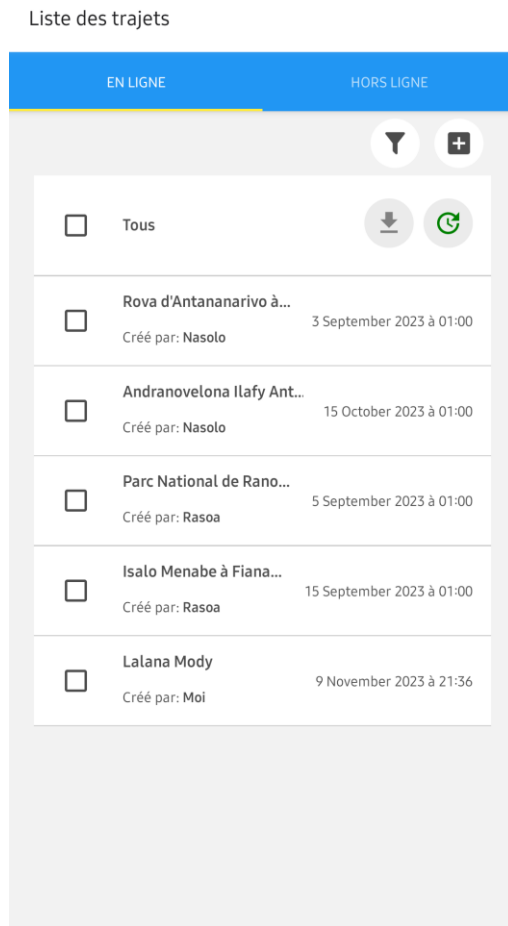


Figure 11 : liste des trajets en ligne

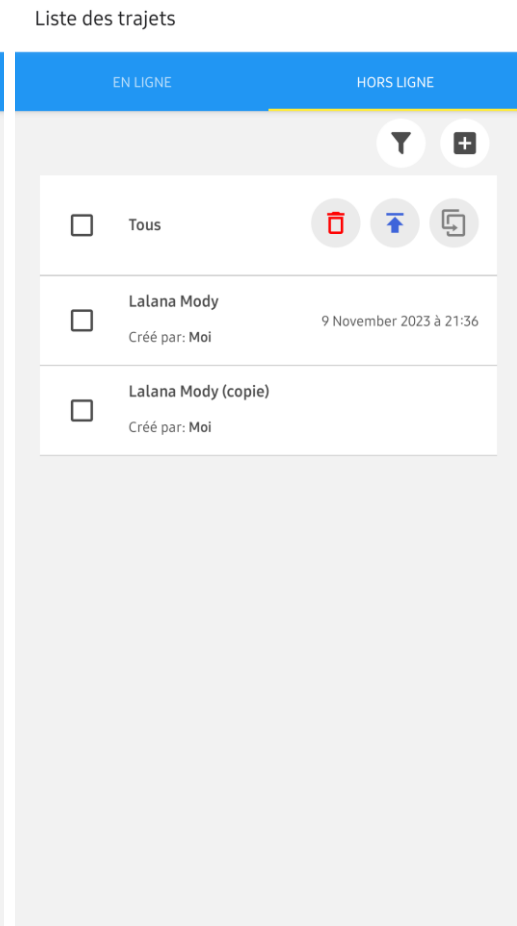


Figure 12 : Liste des trajets hors ligne

2.2.5 Création d'un trajet

Cette fonctionnalité permet à l'utilisateur de créer un trajet et de le consulter sans l'aide d'une connexion internet. Une fois le trajet créé, l'utilisateur peut le modifier, y enregistrer des obstacles, et effectuer des actions comme la publication et la suppression.

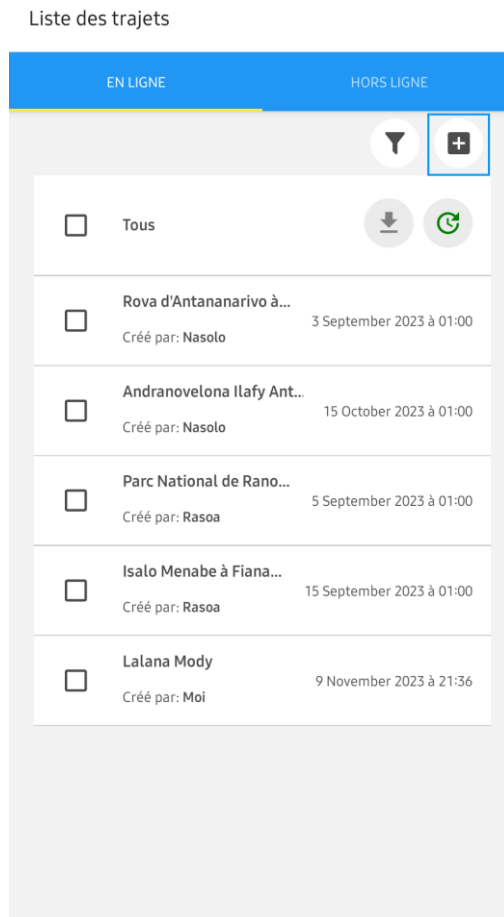


Figure 13 : Création d'un trajet : bouton ajouter

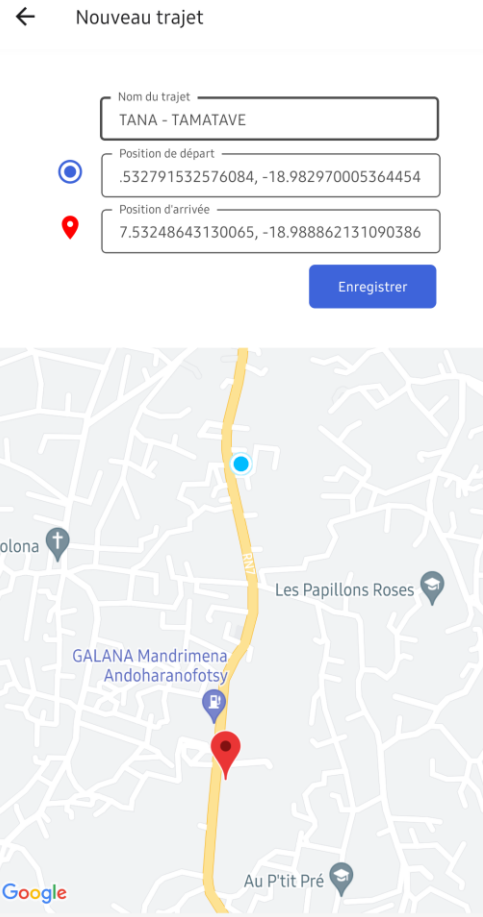


Figure 14 : Création d'un trajet : Mettre les informations

2.2.6 Suppression d'un trajet

Cette fonctionnalité permet aux utilisateurs de supprimer les trajets qu'ils ont enregistrés dans la base de données SQLite pour le côté front office. De plus, l'administrateur a la possibilité de supprimer les trajets dans la base de données PostgreSQL en ligne.

2.2.7 Duplication d'un trajet

La fonctionnalité de duplication d'un trajet (qui duplique également les obstacles présents sur ce trajet) est développée pour permettre aux autres utilisateurs de modifier le trajet afin de l'améliorer et de le publier par la suite.

2.2.8 Publication d'un trajet

La publication d'un trajet permet à d'autres utilisateurs d'envoyer leurs trajets en ligne, afin que d'autres utilisateurs puissent les voir et les télécharger.

2.2.9 Téléchargement d'un trajet

Cette fonctionnalité permet aux utilisateurs de télécharger des trajets pour que les utilisateurs puissent les utiliser sans l'aide d'une connexion internet.

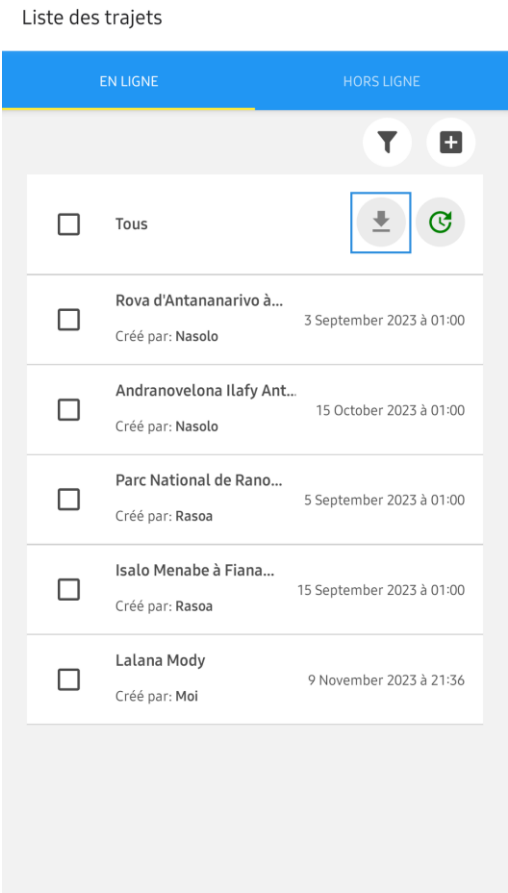


Figure 15 : Action sur un trajet : Télécharger

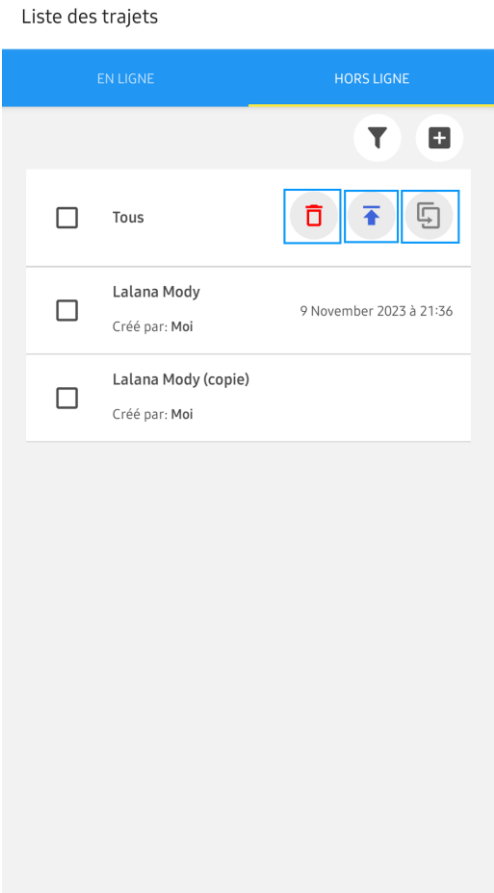


Figure 16 : Action sur un trajet : Supprimer, Publier, Dupliquer

2.2.10 Liste des téléchargements de trajets

Cette fonctionnalité est réservée pour l'administrateur afin d'avoir un tableau récapitulatif des téléchargements de trajets : le nombre de téléchargements d'un trajet, le nombre de téléchargements des trajets effectué par un utilisateur spécifique, ...

2.2.11 Détails et modification d'un trajet (avec vue de trajets sur map)

La fonctionnalité de détails et modification d'un trajet permet aux utilisateurs de voir les détails importants d'un trajet : le nom du trajet, son début et sa fin, les obstacles présents dans le trajet sont visualisés sur la carte géographique (map).

Aperçu pour le Front office

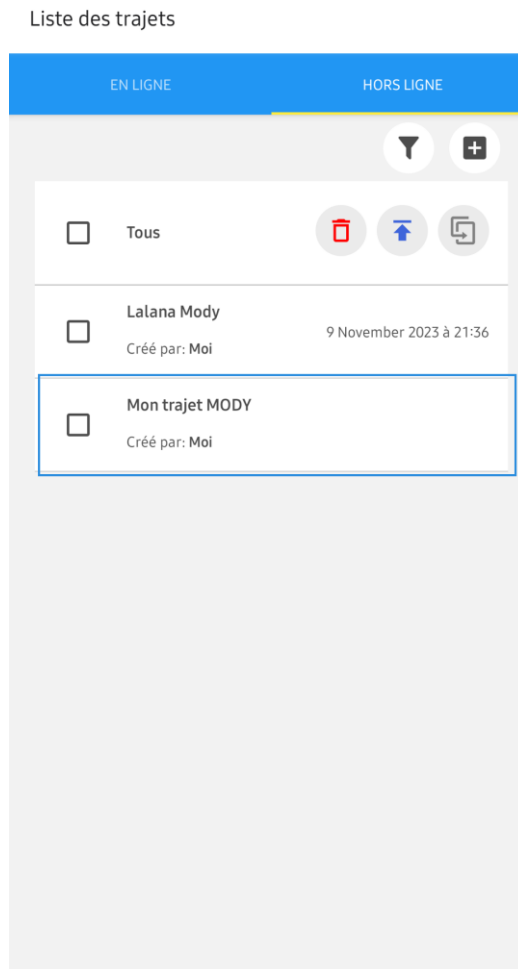


Figure 17 : Détail d'un trajet : sélection d'un trajet (Front office)

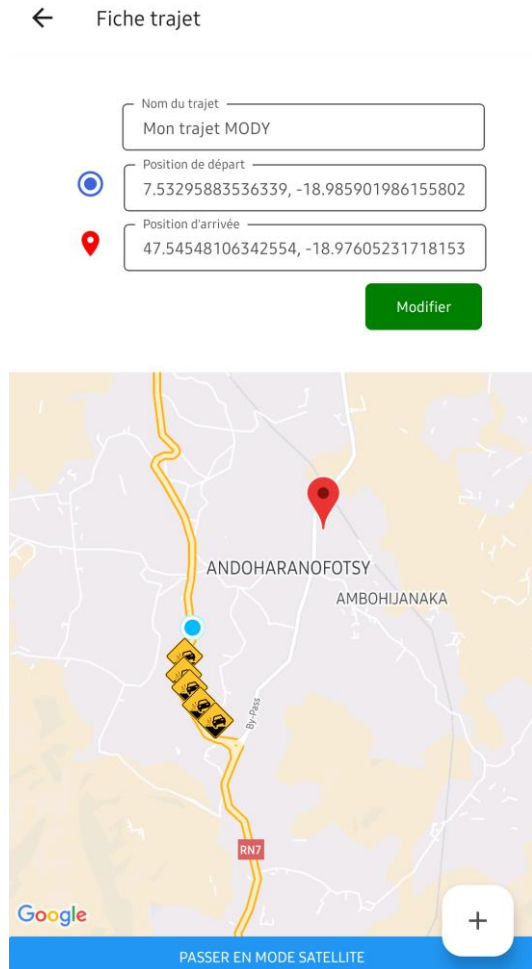


Figure 18 : Détail d'un trajet : informations avec bouton modifier (Front office)

Aperçu pour le Back office

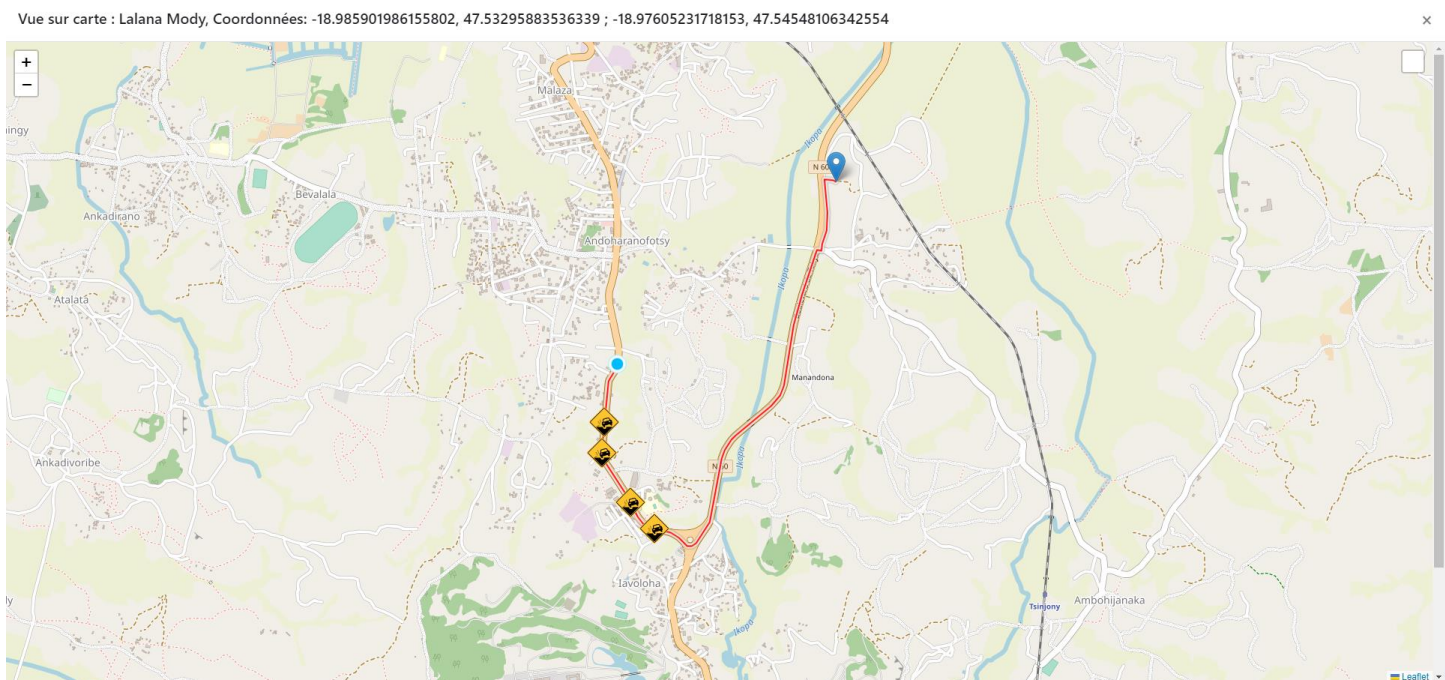


Figure 19 : Détail d'un trajet : informations (Back office)

2.2.12 Liste des obstacles

Ce module offre à l'administrateur la capacité de visualiser la liste des obstacles présents dans la base de données. Cette visualisation est facilitée par une fonction de récapitulation et de filtrage (ou recherche) intégrée.

2.2.13 Enregistrement des obstacles sur un trajet

L'objectif de cette fonctionnalité est de permettre à l'utilisateur créateur du trajet d'enregistrer en temps réel les obstacles rencontrés lors de son parcours. L'enregistrement est effectué par le biais de clics. Si un obstacle se présente à gauche, l'utilisateur est invité à cliquer sur le cercle situé à gauche de l'écran. Si l'obstacle est au centre, l'utilisateur doit cliquer sur le cercle central de l'écran. Enfin, si l'obstacle est à droite, l'utilisateur est dirigé vers le cercle droit de l'écran pour enregistrer son interaction. Cette fonctionnalité n'est pas disponible pour le côté Back office, car l'enregistrement se fait par localisation et a besoin du déplacement de l'utilisateur.

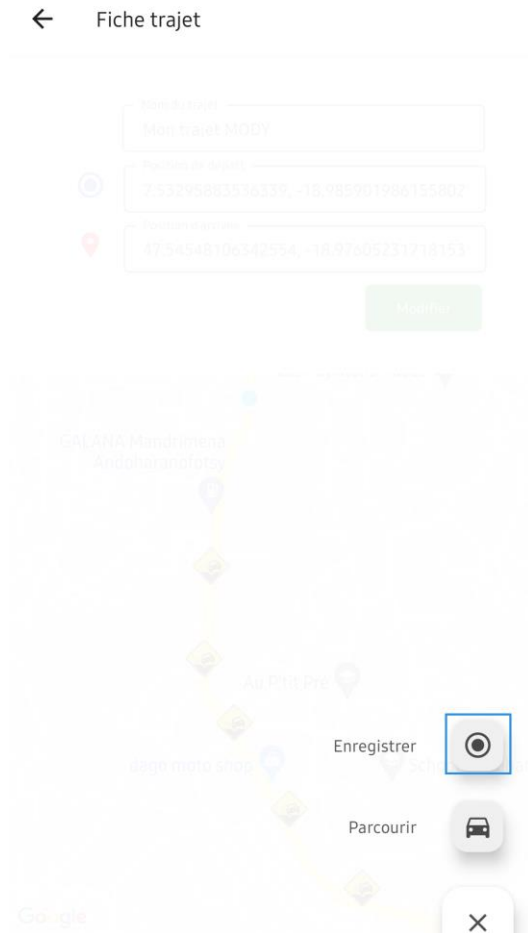


Figure 20 : Enregistrement des obstacles : bouton enregistrer

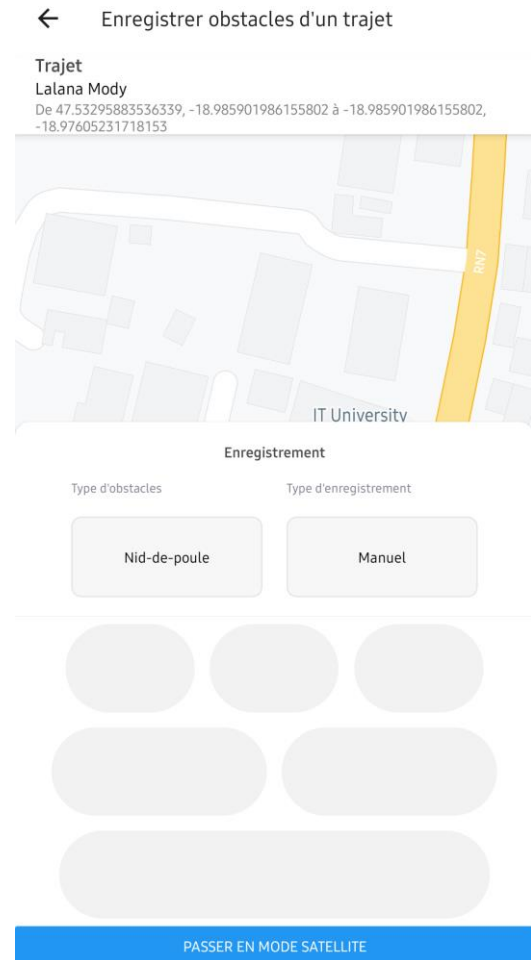


Figure 21 : Enregistrement des obstacles : Enregistrement par clique

2.2.14 Alerte des obstacles sur un trajet

Cette fonctionnalité constitue le cœur du projet, en tant que module principal dédié à la détection et à l'alerte en temps réel de la présence de nids-de-poule tout au long du trajet de l'utilisateur. Elle se distingue par sa capacité à réagir instantanément, fournissant des alertes immédiates dès qu'un obstacle est détecté.

L'alerte générée peut être transmise à l'utilisateur de deux manières principales : tout d'abord, par le biais d'une notification sur son téléphone, garantissant ainsi une communication rapide et accessible. De plus, elle peut également être signalée au moyen de LEDs Arduino, offrant une dimension visuelle à l'alerte. Cette dualité dans les modes d'alerte vise à assurer une réponse rapide et adaptée aux préférences ou aux besoins spécifiques de l'utilisateur.

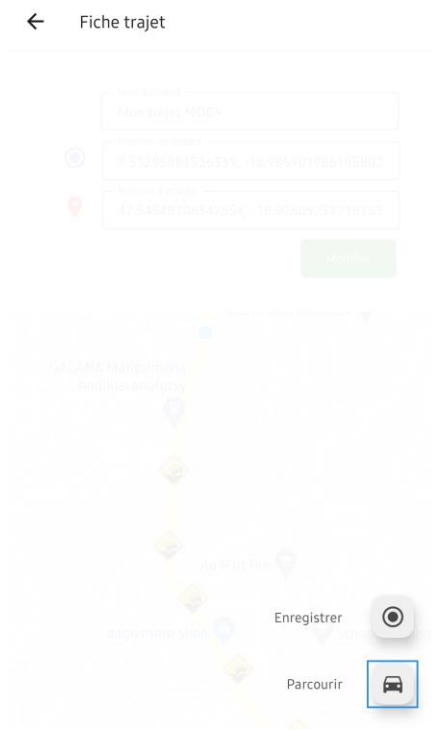


Figure 22 : Alerte des obstacles : bouton parcourir

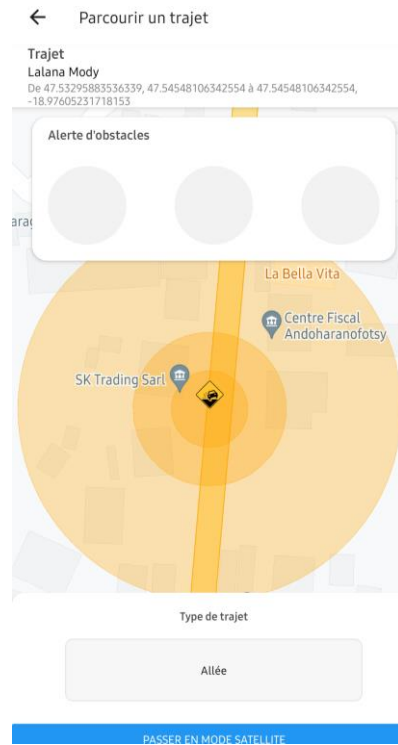


Figure 23 : Alerte des obstacles : alertes

2.2.15 Représentation des obstacles sur carte et détail des obstacles avec les actions possibles sur l'obstacle (suppression, ...)

Le détail d'un obstacle permet d'afficher ses caractéristiques (par exemple sa position sur la chaussée, etc.) et aussi de les modifier en cas de changement.

Aperçu pour le Front office



Figure 24 : Affichage sur map (Front office)

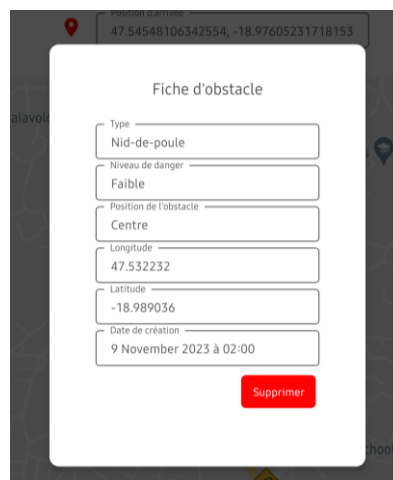


Figure 25 : Détail d'un obstacle (Front office)



Figure 26 : Confirmation de suppression d'un obstacle (Front office)

Aperçu pour le Back office

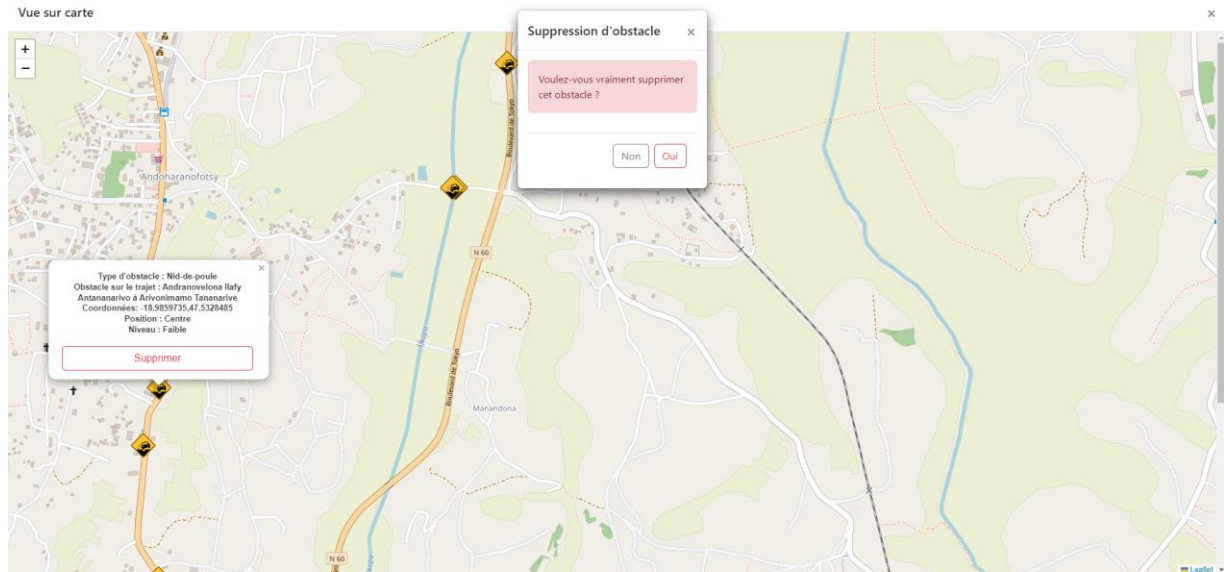


Figure 27 : Affichage sur map, Détail d'un obstacle, Suppression d'un obstacle (Back office)

2.2.16 Notes d'un trajet

Ce module permet à l'utilisateur de visualiser les notes et donner son avis sur un trajet (en donnant une note de 1 à 5). Tandis que l'administrateur visualise et analyse ces notes.

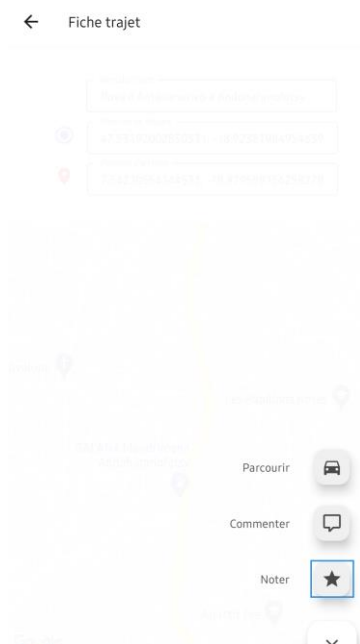


Figure 28 : Note d'un trajet : Bouton " Noter "

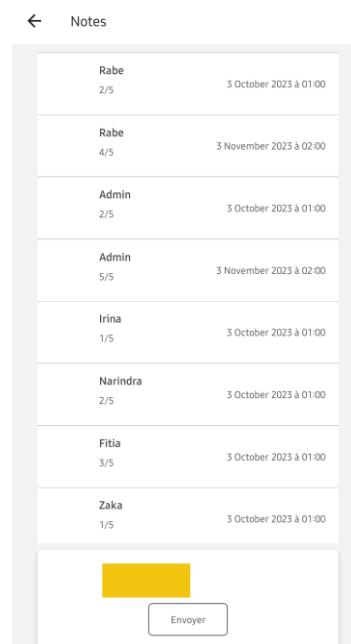


Figure 29 : Note d'un trajet : Liste des notes

2.2.17 Commentaires d'un trajet

L'objectif de cette fonctionnalité est de permettre à l'utilisateur de visualiser les commentaires et de donner son avis détaillé sur un trajet en laissant un ou plusieurs commentaires. Tandis que l'administrateur visualise et analyse ces commentaires.

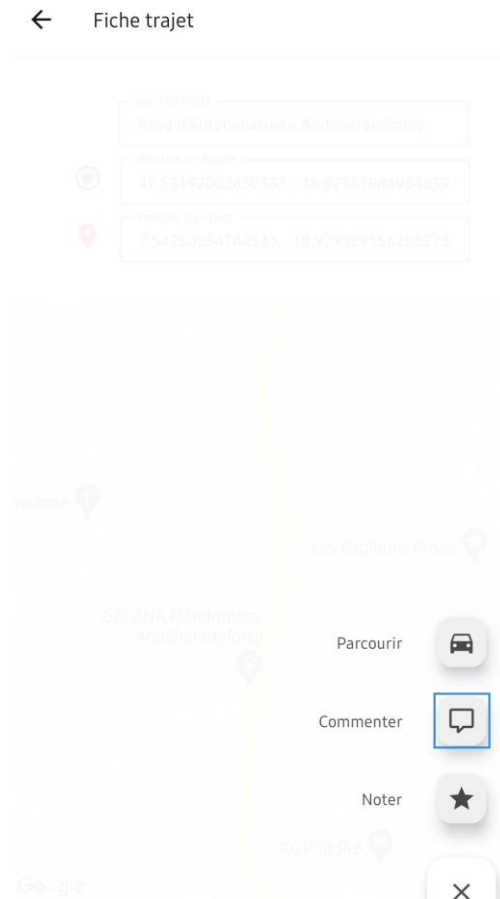


Figure 30 : Commentaire d'un trajet : bouton commenter

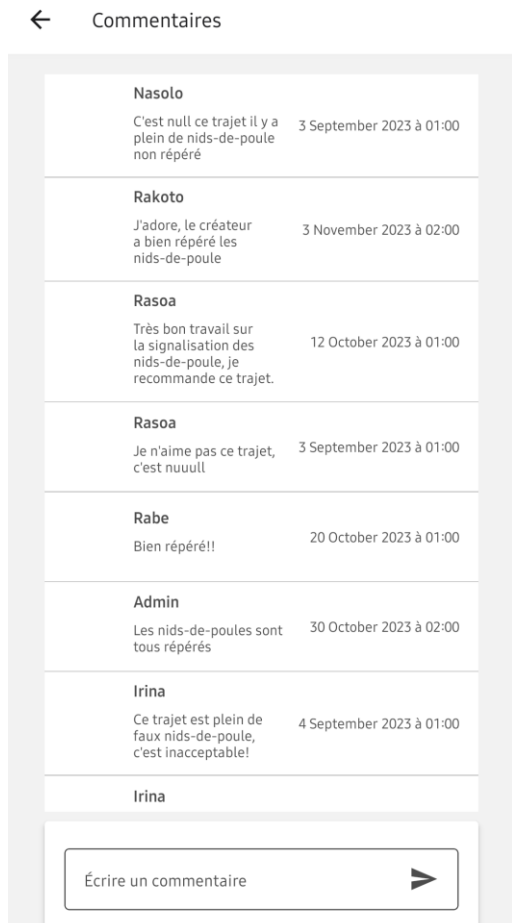


Figure 31 : Commentaire d'un trajet

2.2.18 Suppression d'une note d'un trajet

Il est possible que certains utilisateurs soumettent des évaluations négatives pour des trajets qui présentent un obstacle unique et isolé. Cette fonctionnalité permet aux administrateurs de gérer ces situations, en supprimant les évaluations injustifiées et en maintenant l'intégrité des données de l'application.

2.2.19 Suppression d'un commentaire d'un trajet

Cette fonctionnalité offre aux administrateurs la capacité de gérer les situations où les utilisateurs soumettraient des commentaires inappropriés ou vulgaires. Grâce à cette fonction, les administrateurs peuvent maintenir un environnement respectueux et professionnel pour tous les utilisateurs.

2.2.20 CRUD des types d'obstacles

Cette fonctionnalité offre la capacité de gérer efficacement les types d'obstacles. Elle permet l'ajout de nouveaux types d'obstacles autres que les nids-de-poule, afin d'avoir une vision globale des obstacles présents sur un trajet donné.

2.2.21 Modification de la configuration

Cette fonctionnalité offre la possibilité de modifier les paramètres globaux de l'application, notamment pour l'alerte avec l'utilisation de la LED Arduino. Cette fonctionnalité est réservée au front office vu que chaque utilisateur a son propre configuration.

Configuration de l'application

URL de base de la lumière http://		
Vitesse de la voiture (en millisecondes) 8		
Distances d'alerte		
Distance 1 10	Distance 2 20	Distance 3 50
Vitesses d'alerte (en millisecondes)		
Vitesse 1 0	Vitesse 2 200	Vitesse 3 500
Lumière gauche		
Chemin 1	Chemin 2	Chemin 3
Lumière centre		
Chemin 1	Chemin 2	Chemin 3
Lumière droite		
Chemin 1	Chemin 2	Chemin 3
<button>Modifier</button>		

Figure 32 : Modification de configuration de l'application

Configuration de l'application		
URL de base de la lumière http://		
Vitesse de la voiture (en millisecondes) 8		
Distances d'alerte		
Distance 1 10	Distance 2 20	Distance 3 50
Vitesses d'alerte (en millisecondes)		
Vitesse 1	Vitesse 2	Vitesse 3
0	200	500
Étes-vous sûres de modifier la configuration ?		
<button>Annuler</button>		<button>Confirmer</button>
Lumière gauche		
Chemin 1	Chemin 2	Chemin 3
Lumière centre		
Chemin 1	Chemin 2	Chemin 3
Lumière droite		
Chemin 1	Chemin 2	Chemin 3
<button>Modifier</button>		

Figure 33 : Confirmation de configuration de l'application

2.3 État d'Analyse et Statistiques

2.3.1 Moyenne d'obstacles par kilomètre

Cette statistique indique le nombre moyen d'obstacles, tels que les nids-de-poule, que l'on peut rencontrer sur une distance déterminée, de un à plusieurs kilomètres. C'est un indicateur clé de la qualité de la route. Plus ce nombre est élevé, plus la route est susceptible d'être en mauvais état. Cette information est cruciale pour les conducteurs, les transporteurs et les services d'entretien des routes.

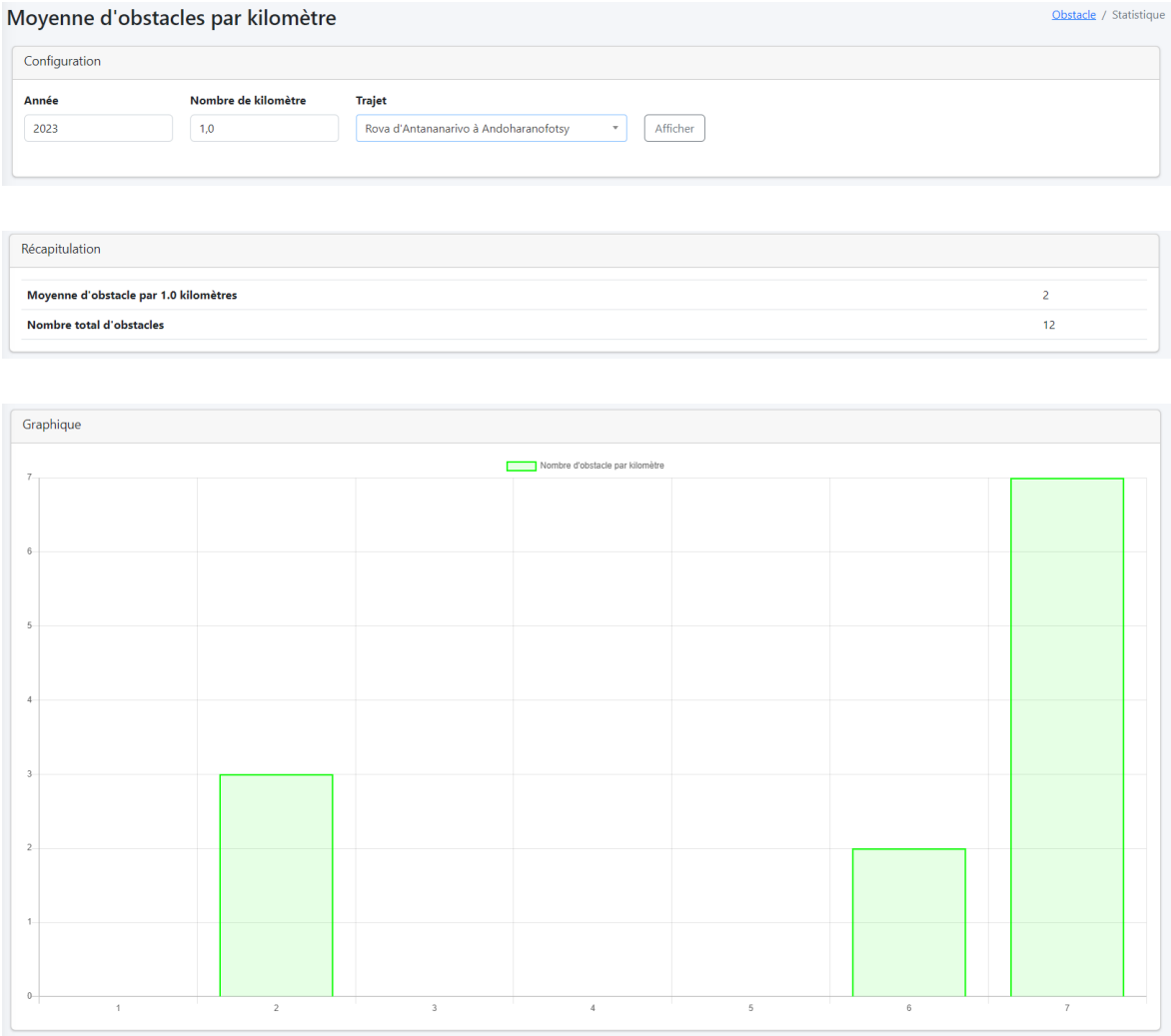


Figure 34 : Moyenne d'obstacles par kilomètre

2.3.2 Calendrier d'évolution des obstacles

Cette statistique montre l'évolution avec le temps du nombre d'obstacles sur un parcours donné, en identifiant les réparations éventuelles, la construction de nouvelles routes ou la présence de nouveaux obstacles.

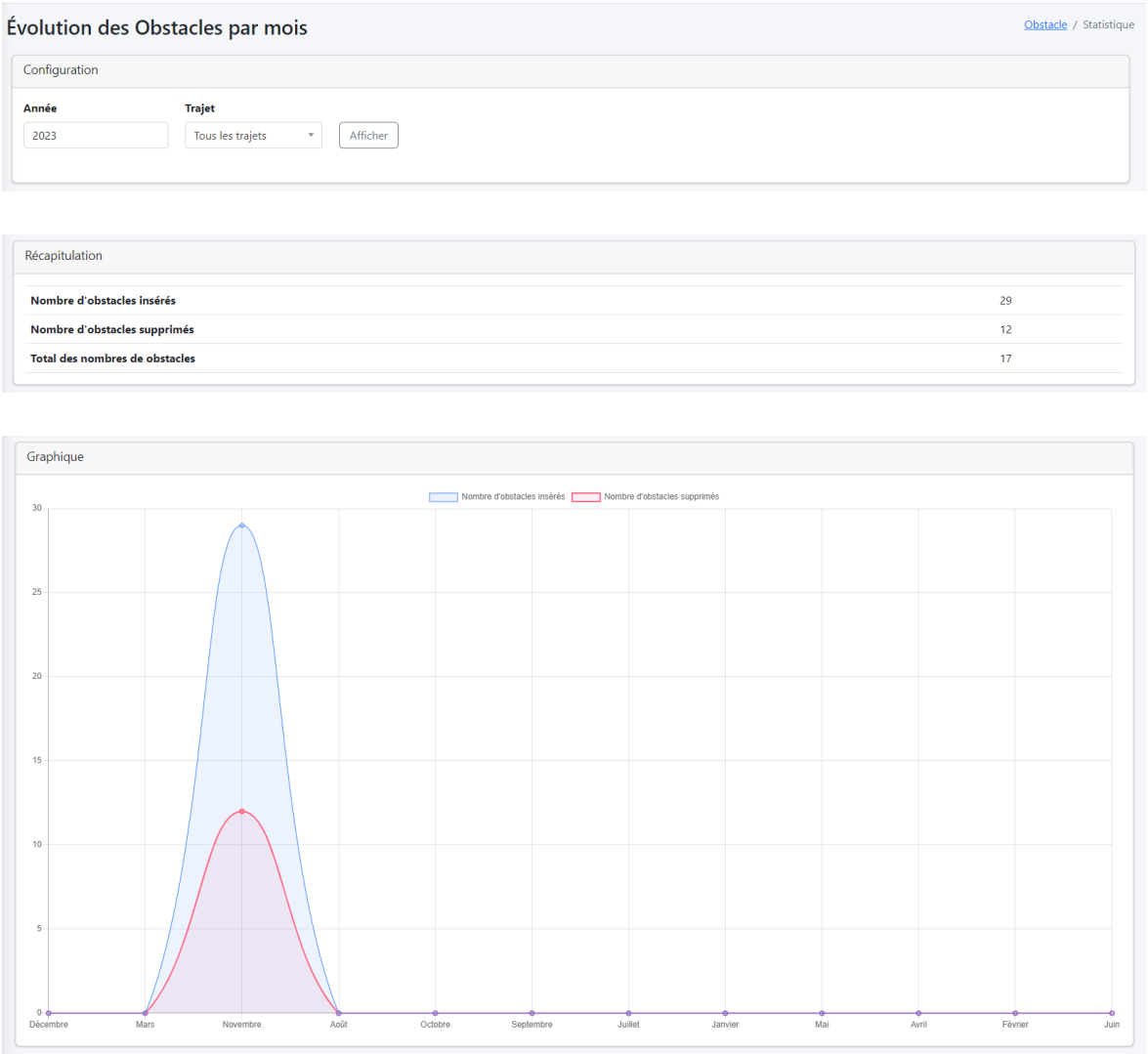
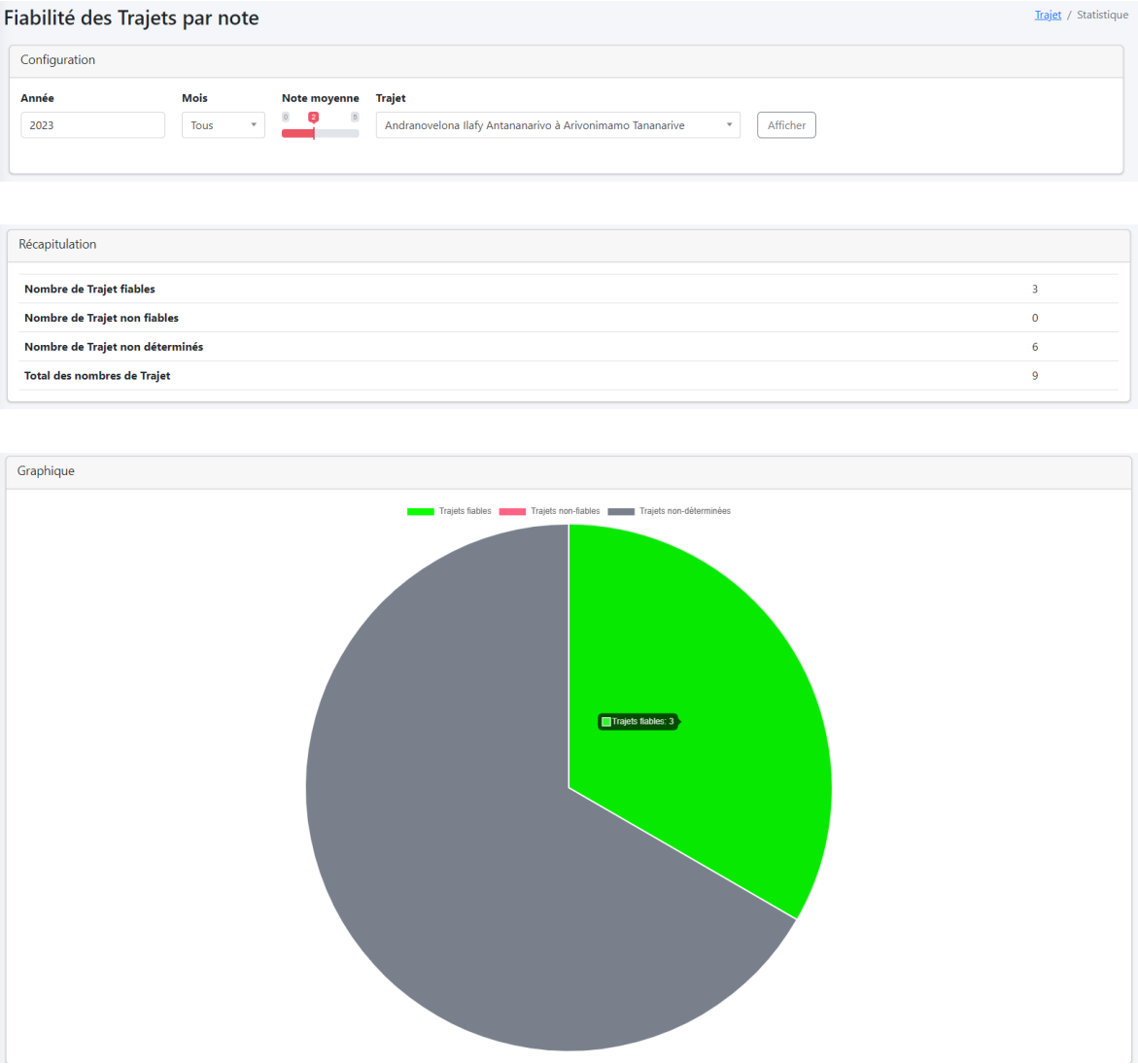


Figure 35 : Évolution des obstacles par mois

2.3.3 Fiabilité d'un trajet par notes

Cette statistique illustre la fiabilité d'un trajet en fonction d'une note moyenne attribuée. Elle permet d'identifier le nombre de trajets considérés comme fiables par rapport à ceux jugés non fiables.



Récapitulation

Nombre de Trajet fiables	3
Nombre de Trajet non fiables	0
Nombre de Trajet non déterminés	6
Total des nombres de Trajet	9

Graphique

Legend: Trajets fiables (green), Trajets non-fiables (red), Trajets non-déterminés (grey).

Catégorie	Nombre
Trajets fiables	3
Trajets non-fiables	0
Trajets non-déterminés	6

Figure 36 : Fiabilité des trajets par note

2.4 Problèmes rencontrés et solutions

- **Les services de géolocalisation, bien qu'essentiels pour de nombreux projets, peuvent souvent entraîner des coûts supplémentaires** : par exemple, l'API Google Maps, largement utilisée pour ses fonctionnalités robustes, est un service payant. De même, des solutions de remplacement autrefois gratuites, comme Mapbox GL, ont récemment modifié leur structure de tarification et ne sont plus gratuites à partir de la version 2.0.

Solution : Face à ces contraintes financières, une stratégie viable serait d'adapter le projet pour tirer le meilleur parti des fonctionnalités offertes gratuitement par ces services. Cela pourrait impliquer l'utilisation sélective de certaines fonctionnalités, l'optimisation du nombre de requêtes effectuées ou la recherche d'autres services de géolocalisation offrant des plans gratuits. Cette approche permettrait non seulement de respecter les contraintes budgétaires, mais aussi d'exploiter efficacement les ressources disponibles.

- **Problèmes lors des tests de l'application :**

J'ai également rencontré un problème lorsque je testais l'application ; en effet, pour effectuer des tests, même mineurs, de l'application, je devais me déplacer, ce qui entraînait une augmentation assez importante du temps passé à réaliser ce travail.

Solution : Pour résoudre ce problème, j'ai utilisé une application de faux GPS sur Android, appelée "Fake Location". Cette application est gratuite et disponible sur le Play Store. Elle permet de simuler une localisation, ce qui m'a permis de gagner du temps lors des tests.

3 Évaluation du projet et connaissances acquises

3.1 Bilan pour l'entreprise

L'élaboration de "ROAD WARN" constitue une grande avancée pour AZLink. En tant qu'application mobile inédite, elle offre la possibilité d'élargir le portefeuille de produits proposé par l'entreprise et d'explorer de nouvelles opportunités de marché.

Cette application peut être intégrée avec d'autres services offerts par l'entreprise, tels que les services de transport.

Les objectifs du projet, c'est-à-dire l'amélioration du confort de conduite et la réduction des dommages matériels et des coûts de réparations de véhicules, ont été réalisés avec succès, illustrant ainsi l'engagement de la société à développer des produits ayant des effets positifs sur la société.

3.2 Bilan personnel

Au cours de mon stage dans la société informatique AZLink, j'ai eu l'opportunité d'acquérir des connaissances sur des technologies nouvelles et pertinentes, notamment React Native et Thymeleaf.

Ce stage m'a également permis d'élargir mon cercle de connaissances humaines tout en développant et améliorant mes facultés de relation, grâce à l'interaction régulière avec tous les employés de l'entreprise (et notamment mon encadreur professionnel). Ainsi, ce stage a été pour moi une expérience très enrichissante et je suis convaincu que toutes les compétences techniques et sociales que j'ai acquises seront bénéfiques pour ma future carrière.

En résumé, ce stage a été une expérience enrichissante qui m'a permis d'améliorer à la fois mes compétences techniques et sociales. Je suis convaincu que ces compétences me seront bénéfiques pour ma future carrière dans le domaine informatique.

3.3 Extension et évolution de l'application

Intégration d'un gyroscope pour l'enregistrement automatique des nids-de-poule : l'intégration d'un gyroscope dans l'application pour la détection automatique

des nids-de-poule représente une avancée technologique majeure. Cette intégration vise à optimiser la précision de la détection des nids-de-poule, tout en minimisant l'intervention manuelle de l'utilisateur. En exploitant les données du gyroscope, l'application peut identifier les variations des mouvements dus aux différentes inclinaisons du véhicule lorsqu'il franchit l'obstacle ou le nid-de-poule existant sur la chaussée, permettant ainsi une détection immédiate de l'obstacle et déclenchant une alerte instantanée.

Alerte automatique d'obstacles en « Retour » : Dans cette perspective, l'utilisateur peut désormais parcourir un trajet tout en recevant des alertes précises sur les obstacles à l'aller ou au retour, sans nécessité de recourir au clic du bouton « Retour ».

Alerte d'obstacles (nids-de-poule) en mode libre : L'application recueille toutes les données d'alertes existantes dans la base de données afin que l'utilisateur puisse être alerté librement partout où il va.

Conclusion

AZLink a exprimé une grande satisfaction concernant l'achèvement du projet, soulignant la réalisation des objectifs prédéfinis. L'implémentation réussie de la fonctionnalité d'alerte en temps réel dans l'application "ROAD WARN" a été reconnue comme une avancée pour l'entreprise. De plus, la capacité à publier et à télécharger des données de trajets vers le serveur a été identifiée comme une amélioration majeure de l'infrastructure technologique de l'entreprise.

En outre, l'entreprise a apprécié cette très intéressante contribution à l'expansion de son portefeuille de produits et à la prospection de nouvelles opportunités de marché. La réalisation de l'objectif principal du projet, à savoir le confort de conduite et la réduction des dommages matériels et des coûts de réparations de véhicules, a été particulièrement saluée.

Sur le plan personnel, j'ai trouvé enrichissant d'avoir l'opportunité d'acquérir des connaissances sur des technologies innovantes et pertinentes, notamment React Native et Thymeleaf. De plus, j'ai apprécié l'occasion d'étendre mon réseau social et d'améliorer mes compétences personnelles grâce à des interactions régulières avec tous les employés de l'entreprise (et notamment mon encadreur professionnel).

Dans une perspective d'évolution, l'intégration d'un gyroscope dans l'application pour la détection automatique des nids-de-poule représente une avancée technologique majeure. Cette intégration vise à optimiser la précision de la détection des nids-de-poule, tout en minimisant l'intervention manuelle de l'utilisateur.

Bibliographie

- Les limites de SQLite.
<https://www.sqlite.org/limits.html> (Consulté le 12 octobre 2023)
- Les avantages et les inconvénients de React Native.
<https://blog.back4app.com/fr/react-native-avantages-et-inconvenients/> (Consulté le 30 octobre 2023)
- Les avantages et les inconvénients de Spring Boot.
<https://scand.com/company/blog/pros-and-cons-of-using-spring-boot/> (Consulté le 30 octobre 2023)
- La documentation de React Native Paper.
<https://callstack.github.io/react-native-paper/docs/components/ActivityIndicator>
(Consulté le 14 septembre 2023)
- L'installation de React Native.
<https://reactnative.dev/docs/environment-setup> (Consulté le 11 août 2023)
- Générateur d'icône de l'application.
<https://www.appicon.co/> (Consulté le 22 octobre 2023)
- Création de projet Spring Boot avec les dépendances.
<https://start.spring.io/> (Consulté le 14 octobre 2023)
- La documentation de Thymeleaf.
<https://www.thymeleaf.org/doc/tutorials/3.1/thymeleafspring.html> (Consulté le 16 octobre 2023)
- La connexion de TypeScript avec la base de données.
https://www.skema.cloud/en/blog/sagot-dev-2/get-started-with-sqlite-database-in-a-typescript-project-7?utm_campaign=Migration+blog+sagot.dev&utm_source=sagot.dev&utm_medium=Website (Consulté le 4 septembre 2023)
- La bibliothèque d'icônes pour React Native Paper.
<https://pictogrammers.com/library/mdi/> (Consulté le 19 septembre 2023)

- Configuration de PostgreSQL pour permettre la connexion à distance
<https://www.bigbinary.com/blog/configure-postgresql-to-allow-remote-connection>
(Consulté le 27 octobre 2023)
- Générateur de gitignore
<https://www.toptal.com/developers/gitignore> (Consulté le 25 octobre 2023)
- Modèles de base pour la navigation mobile
<https://www.nngroup.com/articles/mobile-navigation-patterns/> (Consulté le 26 octobre 2023)
- Alternatives de Google maps pour React Native
<https://stackoverflow.com/questions/48275409/react-native-google-maps-alternative> (Consulté le 27 octobre 2023)
- Exécuter fichier SQL avec SQLite
<https://database.guide/5-ways-to-run-sql-script-from-file-sqlite/> (Consulté le 27 octobre 2023)
- Génération de fichier APK
<https://medium.com/geekculture/react-native-generate-apk-debug-and-release-apk-4e9981a2ea51> (Consulté le 03 novembre 2023)
- Transmettre une valeur d'un écran à un autre en React Native
<https://aboutreact.com/react-native-pass-value-from-one-screen-to-another-using-react-navigation/> (Consulté le 30 octobre 2023)
- Site de téléchargements des paquets React Native
<https://www.npmjs.com/> (Consulté le 30 octobre 2023)
- Accès à distance avec Apache Tomcat
https://tecadmin.net/allow-tomcat-manager-access-from-remote-host/#google_vignette (Consulté le 02 novembre 2023)
- Animer vers une coordonnée avec React Native

<https://stackoverflow.com/questions/42032015/react-native-maps-how-to-animate-to-coordinate> (Consulté le 13 octobre 2023)

- Calculer la distance entre deux positions géographiques avec React Native

[https://aboutreact.com/react-native-calculate-distance-between-two-locations/#:~:text=Normal%20distance%20calculation%20can%20be,distance%20between%20two%20geo%20coordinates.&text=This%20function%20takes%20up%20to,or%20decimal\(51.5103\)%20format](https://aboutreact.com/react-native-calculate-distance-between-two-locations/#:~:text=Normal%20distance%20calculation%20can%20be,distance%20between%20two%20geo%20coordinates.&text=This%20function%20takes%20up%20to,or%20decimal(51.5103)%20format). (Consulté le 9 octobre 2023)

- Suivre statut de connexion internet avec React Native

<https://hackajob.com/talent/blog/how-track-internet-connection-status-with-react-native> (Consulté le 11 octobre 2023)

- Rafraîchissement des pages React Native

<https://enappd.com/blog/refreshcontrol-pull-to-refresh-in-react-native-apps/130/>
(Consulté le 11 octobre 2023)

- Tutoriel allumer une LED avec Arduino

<https://tesseract-it.com/blog/article/comment-allumer-une-led-en-continu-avec-arduino> (Consulté le 21 octobre 2023)

Annexe

Étapes du déploiement du Back office

Le serveur de déploiement de l'entreprise est Contabo VPS utilisant un système d'exploitation Linux.

○ Déploiement de la base de données PostgreSQL

▪ Installation de PostgreSQL 14.9

Exécution de la commande suivante en tant qu'administrateur (sudo)

```
apt-get install -y postgresql-14
```

Figure 37 : Installation de PostgreSQL 14

▪ Configuration de PostgreSQL

Modification du fichier /etc/postgresql/14/main/postgresql.conf

```
listen_addresses = '*'
```

Figure 38 : Modification de configuration postgresql.conf

Modification du fichier /etc/postgresql/14/main/pg_hba.conf

```
host      all             all             0.0.0.0/0      md5
host      all             all             :::/0          md5
```

Figure 39 : Configuration de pg_hba.conf

○ Déploiement de l'application mobile

▪ Téléchargement d'apache tomcat 9.0.82

Apache tomcat disponible sur :

<https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.83/bin/apache-tomcat-9.0.82-windows-x64.zip>

▪ Configuration d'apache tomcat

Modification du fichier apache-tomcat-9.0.82/conf/tomcat-users.xml

```
<role rolename="manager-gui"/>
<user username="your-username" password="your-password" roles="manager-gui"/>
```

Figure 40 : configuration de tomcat-users.xml

Modification du fichier apache-tomcat-9-0-82/webapps/host-manager/WEB-INF/manager.xml

```
<Context docBase="${catalina.home}/webapps/manager"
    privileged="true" antiResourceLocking="false" >
    <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookiePr
        sameSiteCookies="strict" />
    <Valve className="org.apache.catalina.valves.RemoteAddrValve"
        allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />
</Context>
```

Figure 41 : configuration de manager.xml

- **Génération d'un fichier war avec le projet Spring boot**
- **Lancement du projet sur apache tomcat**
 - Copie du fichier war dans apache-tomcat-9.0.82/webapps/
 - Lancement de l'apache tomcat en exécutant « ./apache-tomcat-9.0.82/bin/startup.sh ».
 - Ouverture le projet au navigateur.
 - Création d'un fichier script .sh qui déclenche le déploiement du projet.

Extrait d'une fonction Arduino pour allumer une LED en continu

```
void allumerLed(String emplacement, String niveau, int state, int LEDG1, int LEDG2,
if (emplacement == "0") {
    if (niveau == "1") {
        if (state != LEDG1) {
            digitalWrite(state, LOW);
        }
        digitalWrite(LEDG1, HIGH);
        delay(300);
        digitalWrite(LEDG1, LOW);
        delay(300);
        state = LEDG1;
    }
    if (niveau == "2") {
        if (state != LEDG2) {
            digitalWrite(state, LOW);
        }
        digitalWrite(LEDG2, HIGH);
        delay(50);
        digitalWrite(LEDG2, LOW);
        delay(50);
        state = LEDG2;
    }
    if (niveau == "3") {
        digitalWrite(state, LOW);
    }
}
```

Figure 42 : Extrait de fonction pour allumer une LED Arduino