Institut Supérieur des Sciences Appliquées et de Technologie de Gafsa

Département d'ASI

Présenté en vue de l'obtention du

DIPLÔME DE LICENCE APPLIQUEE EN TECHNOLOGIE DE L'INFORMATION ET LA TÈLÈCOMMUNICATION

Création d'une application mobile de localisation des véhicules

Réalisé par : Ghassen Belgacem

Sous la direction de : Mlle. Rabaa Ibrahmi

Soutenu publiquement le 08/06/2022 devant le jury

Président : Ben houssine Moez

Rapporteur : Zahra Basma

Année Universitaire 2022/2023

Remerciements

Il m'est très agréable de réserver cette page comme un témoin de reconnaissance à toutes les personnes qui m'ont soutenu et encadré pour réaliser ce travail.

Je tiens à exprimer ma gratitude et mon remerciement :

Tout d'abord à ***ALLAH*** *le tous puissant pour ces faveurs et se bonté , de nous avoir donné le courage et la patience de terminer ce modeste travail.*

Tout particuliérement , mon encadante ***Mlle Rabaa Ibrahmi*** *pour L'opportunité qu'il m'a offerte pour travailler sur ce projet intéressant, sa disponibilitée et son soutien .*

Je profite aussi de ce Projet fin d'étude pour exprimer mes remerciements envers tous les personnes , toutes l'équipe pétagogique professeurs de ma faculté ***Institut Supérieur des Siences Appliquées et Téchnologie de Gafsa*** *qui nous ont apportés du soutiens durant nos études .*

Je tiens également à exprimer l'honneur que me font les membres du jury pour avoir accepté de me prêter leur attention et évaluer mon travail.

Dédicases

Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,

A toute ma famille et spécialement mes parents.

A tout mes amis.

A toute la promotion 2022/2023

Et a tous ceux qui est ont contribué de prés ou de loin pour que ce projet soit possible, je vous dis merci.

Merci...

Sommaire

Table de figures

Introduction générale

Les applications mobiles sont devenues un outil incontournable pour faciliter la vie quotidienne des utilisateurs. En particulier, les applications de suivie de véhicules sont de plus en plus populaires car elles permettent aux conducteurs de localiser leurs véhicules en temps réel et de planifier leurs itinéraires . Ces applications peuvent également aider les entreprises à optimiser leur gestion de flotte en surveillant les performances de leurs véhicules et en offrant des alertes en cas d'incidents ou de problèmes de maintenance. En somme, les applications de suivi de véhicules apportent un gain de temps et une tranquillité d'esprit aux conducteurs et aux gestionnaires de flotte.

En Tunisie, les applications mobiles de suivi des véhicules sont de plus en plus utilisées par les entreprises. Ces applications permettent aux entreprises de surveiller en temps réel la position de leurs véhicules , de planifier les opérations de maintenance et de réduire les coûts opérationnels. Elles offrent également des avantages aux conducteurs en leur permettant de localiser leur véhicule et de gérer leur temps de travail de manière plus efficace. Ces applications contribuent ainsi à améliorer la productivité et la sécurité des entreprises tout en offrant une expérience utilisateur optimisée.

Pour cela, il nous a été confié, dans le cadre de notre projet de fin d'étude, la tâche de réaliser une application mobile qui permet aux utlisateurs la présentation et la localisation de leurs véhicules dans une carte Map.Ce rapport est constitué de quatre chapitres, le premier est consacré à une étude préalable qui contient le cadre générale de projet, le deuxième représente l'analyse et la spécification des besoins, le troisième s'intéresse à la partie la plus importante qui est la conception et le dernier chapitre traite les différentes phases de la réalisation. Enfin, une conclusion générale qui récapitule toutes les étapes de l'achèvement du programme.

Chapitre 1 : Cadre général de projet

1.1 Introduction

Dans ce premier chapitre nous exposerons l'état de l'art et la description générale du projet, ceci nous permet ainsi de mieux comprendre le sujet et de bien définir son cadre.

1.2 Etat de l'art

1.2.1 Les solutions des mobilités

Le temps a bien changé depuis l'avènement des nouvelles générations apparaissent . Des nombreux ingénieurs estiment que les applications de mobilité seront la tendance clé des dix prochaines années. Les appareils iPhone et Android gagnent en popularité auprès des consommateurs et leur croissance ne semble pas vouloir ralentir.

Une application de mobilité est un programme téléchargeable conçu pour fonctionner sur un système d'exploitation mobile spécifique. Elle est adaptée aux téléphones pour lesquels elle a été développée et peut être distribuée gratuitement ou vendue, notamment via des boutiques d'applications. Ainsi, les marques et les entreprises du monde entier se penchent désormais sur ce phénomène et souhaitent avoir leur propre application mobile, considérant ce marché comme une nouvelle source de revenus et de profits.

En effet , La localisation des véhicules est un besoin essentiel pour de nombreux conducteurs, surtout dans les zones urbaines où il est difficile de trouver une place de stationnement. Les applications de mobilité offrent cette fonctionnalité, permet aux utilisateurs de localiser facilement leurs véhicules à partir de leurs téléphones portables. Cette solution est particulièrement pratique pour les personnes qui utilisent leurs voitures pour se rendre au travail ou dans des zones très fréquentées. Elle leur évite de perdre du temps à chercher leurs voitures et leur permet de se déplacer plus rapidement et plus efficacement. De plus, cette fonctionnalité peut également aider à prévenir le vol des véhicules en permettant aux utilisateurs de surveiller leurs voitures en temps réel.

En somme, les applications du mobilité qui offrent une fonctionnalité de localisation des véhicules sont un avantage inestimable pour les conducteurs urbains,en leur offrant un moyen plus facile et plus pratique pour se déplacer.

1.2.2 Présentation du Projet :

Le sujet consiste à concevoir et développer une application mobile qui offre à l'utilisateur la possibilité de suivre et localiser sa véhicule (Voiture, Drone...) sur une carte Google Maps.

1.3 Mesure de la qualité de l'application

1.3.1 Plan qualité de projet (PQP)

Dans le domaine du service informatique, les clients sont de plus en plus exigeants. En effet, ils demandent une conception, un développement et une recette de qualité afin de faciliter et de minimiser le coût de la maintenance du système d'information réalisé par une société de service. Un projet est d'habitude réparti en plusieurs phases. Avant de commencer à les aborder, le client et le prestataire élaborent un plan qualité projet (PQP) qui a pour but d'assurer le bon déroulement et la réussite du projet. Durant notre projet de fin d'étude, nous avons eu à respecter des normes de qualité mise par notre encadreur, qui sont plus ou moins strictes, mais qui simplifient considérablement la correction des anomalies et l'intégration des évolutions.

Le *PQP* est organisé dans le cadre des réunions périodiques. En effet au début de chaque semaine nous faisions une réunion qui faisait l'objet d'un ordre du jour et d'un compte-rendu. Après chaque réunion j'étais demandé de rédiger un PV (Procès-Verbal). Ce dernier contient les points discutés durant la réunion, le travail réalisé pour chaque partie et le travail demandé pour la prochaine fois.

Les PV seront envoyés à mon encadrant *Mlle Rabaa Ibrahmi* dans 24h après la réunion, ces derniers sont accompagné parfois par les documents utilisés lors des présentations réalisées.

1.3.2 Diagramme du GANTT

Pour planifier bien notre projet et rendre plus simple le suivi du son avancement nous avons utilisé le diagramme de GANTT. Cette méthode visuelle est efficace puisqu'elle nous a permis de lister les différentes tâches à faire ainsi que la

date de début et de fin de cette tâche. La figure ci-dessous présente le diagramme de GANTT de notre projet :

Semaine / Etape	Février				Mars				Avril				Mai			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Etude préalable																
Conception																
Réalisation																
Test et validation																
Rédaction du rapport																

Figure 1- Diagramme de GANTT

Comme le montre le tableau ci-dessus , cinq principales phases peuvent être dégagées :

L'étude préalable : le résultat de cette phase est la détermination des objectifs à atteindre dans notre future application en partant de l'existant.

Conception : il s'agit de détailler les spécifications des fonctions ainsi que la structure des données, et des contrôles et les interfaces.

Réalisation : il s'agit de réaliser l'implémentation des programmes et effectuer les tests unitaires.

Test et Validation : Il s 'agit de tester notre plateforme de E-Learning.

Rédaction du rapport : description détaillée de notre travail.

1.4 Conclusion

Ce premier chapitreu présente généralement le projet tout en mettant l'accent sur le besoin du lancement, sa problématique, le cahier de charge et la mesure de qualité du projet.

Le chapitre suivant est consacré à une étude approfondie de l'existant et l'analyse de spécification des besoins.

Chapitre 2 : Analyse et spécification des besoins

2.1 Introduction

Avant de se lancer dans la réalisation d'un projet, il est primordial de comprendre les besoins de l'entreprise en étudiant l'existant et en identifiant les problématiques actuelles. Cette analyse détaillée permet de définir les besoins fonctionnels et non fonctionnels de l'application, ainsi que les acteurs impliqués dans le processus. Dans la seconde partie, il convient de présenter un cahier de charge précis et de déterminer les critères de qualité pour évaluer la réussite du projet.

En somme, l'étape cruciale de l'étude de l'existant et de l'analyse des besoins permet de comprendre le problème de manière exhaustive et de s'assurer que la réalisation du projet répondra aux attentes de l'entreprise.

2.2 Etude de l'existant :

2.2.1 Analyse de l'existant :

Dans le domaine de la gestion de flotte de véhicules, les entreprises utilisent souvent des systèmes de suivi GPS pour surveiller les mouvements de leurs véhicules en temps réel. Toutefois, la plupart de ces systèmes ne sont accessibles qu'à partir de postes de travail fixes et sont souvent limités dans leur portée, leur précision ou leur interopérabilité avec d'autres systèmes.

2.2.2 Critique de l'existant :

Cette situation actuelle présente plusieurs lacunes, notamment :

. une portée limitée et un accès restreint aux systèmes de suivi GPS .

. une précision souvent insuffisante pour des applications professionnelles .

. une interopérabilité limitée avec d'autres systèmes de gestion de la flotte de véhicules.

En outre, ces systèmes ne permettent pas de fournir des informations en temps réel aux conducteurs, ni de gérer efficacement les flottes de véhicules sur le terrain.

2.2.3 Solution proposée :

Face à ces limitations, nous proposons de développer une application mobile de suivi de flotte de véhicules qui permettra aux entreprises de surveiller les mouvements

de leurs véhicules en temps réel et de gérer efficacement leur flotte sur le terrain. Cette application offrira une portée étendue, une grande précision et une interopérabilité avec d'autres systèmes de gestion de flotte de véhicules. Les conducteurs pourront également accéder aux informations de la flotte en temps réel, ce qui améliorera la communication et la coordination entre les conducteurs et le personnel de l'entreprise.

2.3 Spécification des besoins

L'analyse du sujet nous a permis de dégager les fonctionnalités qui seront mise à la disposition de l'utilisateur. Dans cette partie, nous allons recenser les fonctionnalités que l'application doit offrir à ses différents utilisateurs.

2.3.1 Spécification des besoins fonctionnels

Nous décrivons pour chaque acteur les cas d'utilisation. On distingue les cas d'utilisation suivants :

Côté utilisateur:

- ✓ Créer un compte
- ✓ S'authentifier
- ✓ Consulter la liste des véhicule
- ✓ Ajouter son vehicule
- ✓ Suivre son véhicule sur une carte Google Maps

Côté Gestionnaire:

- ✓ S'authentifier par les login fournis par le gestionnaire du l'application
- ✓ Gérer la liste des véhicules (Ajouter ou supprimer)
- ✓ Suivre les vehicules sur la Map
- ✓ Consulter la liste des utlisateurs

Côté Admin :

- ✓ Consulter la liste des véhicules
- ✓ Consulter la liste des utilisateurs

2.3.2 Spécification des besoins non fonctionnels

Les besoins non fonctionnels présentent les exigences internes pour le système et cachées vis à vis des utilisateurs, notre application doit être facile à utiliser, avec une bonne ergonomie. Elle doit garantir un temps de réponse court.

- **L'ergonomie**

Le système doit présenter des interfaces graphiques conviviales bien structurées du point de vue contenu informationnel.

- **La sécurité des données**

Sécuriser les données revient à appliquer une stratégie d'identification, d'authentification, l'autorisation et contrôler chaque tentative d'accès à ces données. Dans notre système l'accès aux informations personnelles n'est autorisé qu'aux personnes propriétaires et selon un privilège qui détermine les droits d'accès.

2.3.3 Spécification semi-formelle des besoins

2.3.3.1 Méthode de Conception UML

2.3.3.1.1 Définition

UML n'est pas une méthode (C'est une description normative des étapes de la modélisation) : ses auteurs ont en effet estimé qu'il n'était pas une opportunité de définir une méthode en raison de la diversité des cas particuliers. Ils ont préféré de se borner à définir un langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information aux graphiques qui sont bien sûr associés des textes qui expliquent leur contenu. UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

Pierre-Alain Muller ajoute :

« UML est dans le domaine public, soutenue par le marché : Microsoft, HP, IBM, Oracle... Successeur naturel des méthodes de Booch, OMT et OOSE, UML est le fruit de l'expérience et des besoins de la communauté des utilisateurs. »

2.3.3.1.2 Les avantages de l'UML

UML est un langage formel et normalisé. Il permet ainsi : Un gain de précision, un gain de stabilité et l'utilisation d'outils.

UML est un support de communication performant : Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes. Son caractère polyvalent et sa souplesse font de lui un langage universel.

Objectif :

- ✓ Construire des modèles de systèmes

- ✓ Organiser le travail
- ✓ Gérer le cycle de vie de A à Z
- ✓ Gérer le risque

✓ S²Obtenir de manière répétitive des produits de qualité constante

En conclusion, nous avons choisi de travailler avec UML parce qu'il exprime mieux la vue statique et dynamique du système d'information et pour notre application mobile, il est nécessaire de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.

2.4 Diagrammes de cas d'utilisation

2.4.1 Diagramme des cas d'utilisation de l'utlisateur

Ce diagramme donne à l'utilisateurs la possibilité d'accéder à son compte et d'ajouter son véhicule pour le suivre sur la Map. Chaque utlisateur doit s'authentifier avant d'accéder à l'application.

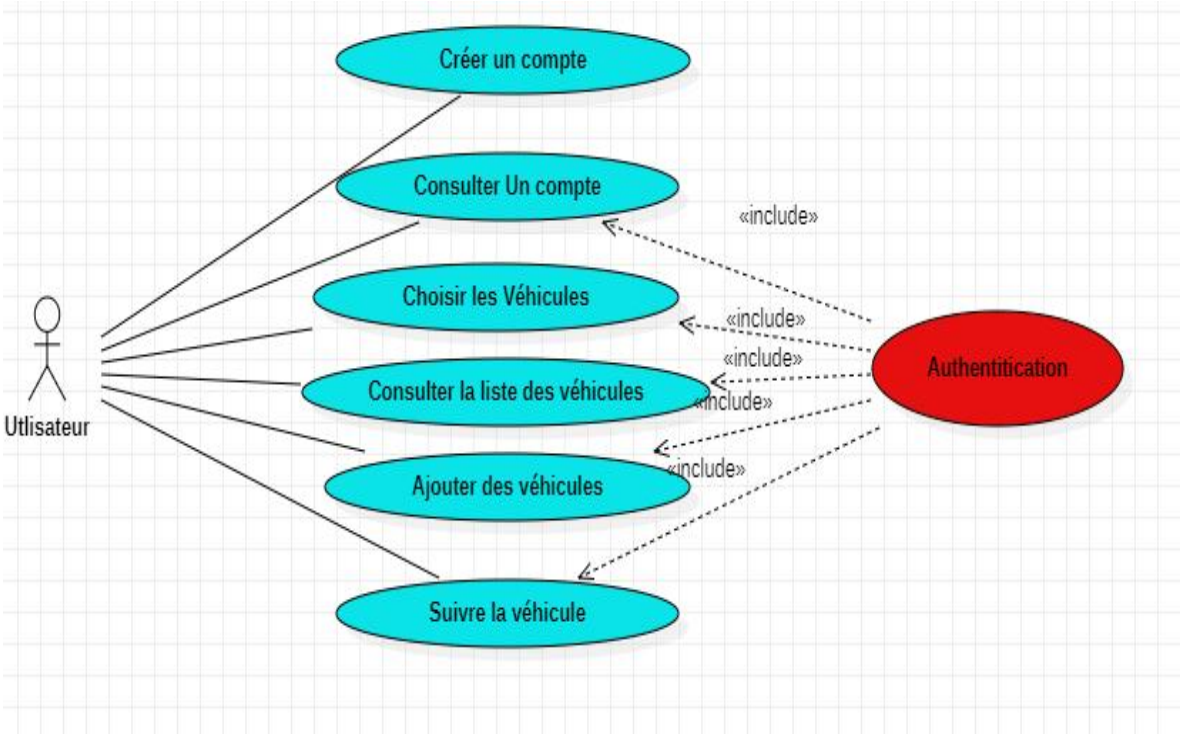


Figure2- Diagramme de cas d'utilisation pour l'utilisateur

2.4.2 Diagramme des cas d'utilisation du gestionnaire

Ce diagramme donne la possibilité au gestionnaire de gérer la liste des véhicules, consulter la liste des utilisateurs et suivre les véhicule sur la Map . Le gestionnaire doit s'authentifier avant d'accéder à l'application.

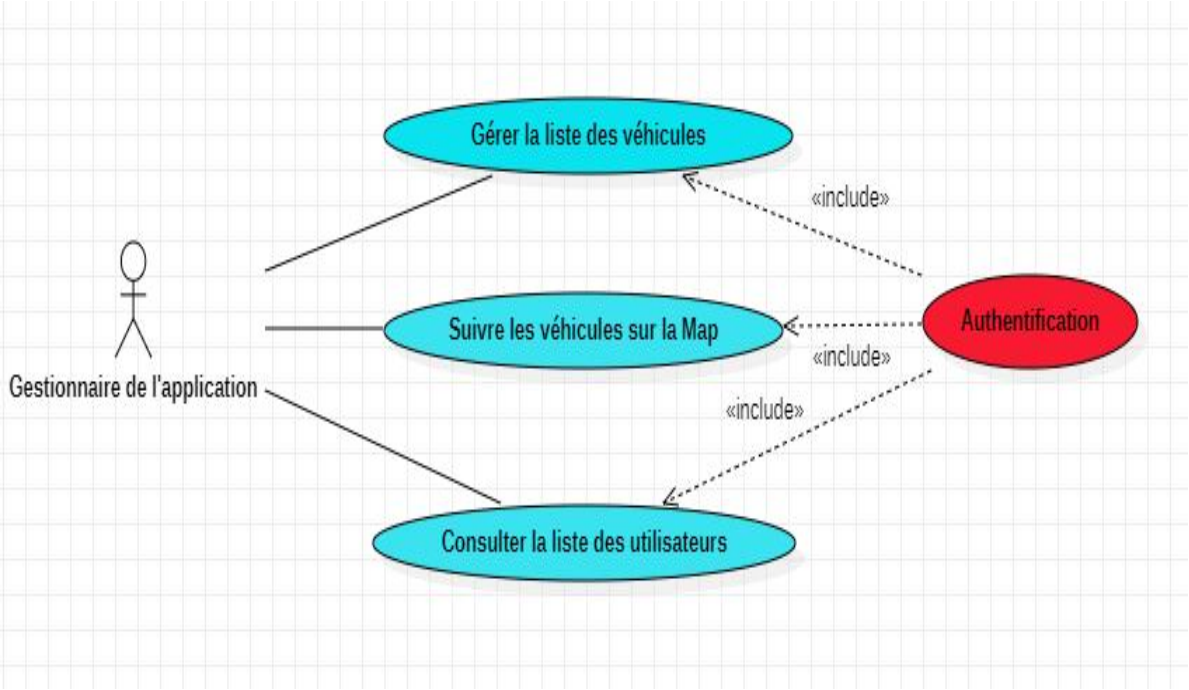
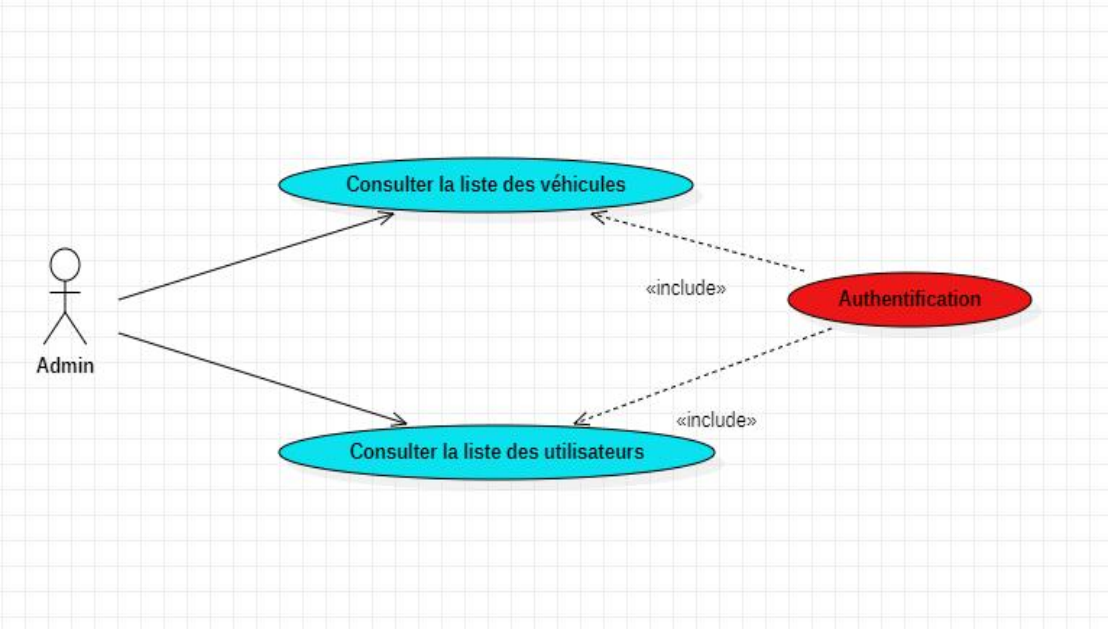


Figure 3- Diagramme de cas d'utilisation pour le gestionnaire de l'application

2.4.3 Diagramme des cas d'utilisation pour Admin

Ce diagramme donne à l'Admin la possibilité Consulter la possibilité des véhicules et la listes des utilisateurs. L'Admin doit s'authentifier pour accéder a son compte.



2.5 Conclusion

Ce chapitre précise les besoins fonctionnels que l'application développée doit offrir aux utilisateurs et les besoins non fonctionnels après l'identification des acteurs ainsi que les cas d'utilisation de chaque acteur.

Chapitre 3 : Conception

3.1 Introduction

La conception est un processus complexe qui nécessite une grande créativité et une connaissance approfondie des systèmes existants. L'expérience pratique et l'étude des systèmes similaires sont des éléments clés pour une conception réussie. Une conception efficace doit satisfaire les spécifications (objectifs et contraintes) et proposer une architecture appropriée. Pour y parvenir, plusieurs solutions doivent être envisagées, comparées et évaluées pour en choisir la meilleure. Le modèle final doit être complet, cohérent, maintenable et testable. Afin d'atteindre ces objectifs, il est nécessaire d'adopter une approche structurée et méthodique. C'est pourquoi l'approche UML est largement utilisée pour la modélisation orientée objet. UML fournit une notation graphique standardisée pour représenter les différents aspects du système, ce qui facilite la communication entre les différents acteurs impliqués dans le processus de conception. Cette approche permet de réduire les erreurs de conception et de garantir la qualité et la cohérence du système. En résumé, la conception est une étape clé de tout projet d'ingénierie et l'approche UML est un outil précieux pour réussir cette étape cruciale.

3.2 Conception générale

3.2.1 Méthodologie adapté

. La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes qui permettent de mettre en place un modèle sur lequel on va s'appuyer.

3.2.2 Orientation et modélisation:

Dans le domaine de l'informatique, la conception d'un logiciel est une étape essentielle avant de passer à sa réalisation. Cette phase de conception consiste à élaborer une représentation abstraite du logiciel à créer, qui permet de mieux comprendre ses fonctionnalités, sa structure et son architecture. Pour cela, il existe plusieurs méthodes de conception, qui proposent chacune des approches et des outils différents. Parmi les méthodes les plus connues, on peut citer l'approche Merise et l'approche UML (Unified Modeling Language). Ces deux approches ont en commun

de chercher à modéliser le système informatique à concevoir, mais elles diffèrent dans leurs concepts et leurs techniques de modélisation.

L'approche Merise propose une modélisation du système réel selon deux points de vue : un point de vue statique (les données) et un point de vue dynamique (les traitements). Cette modélisation repose sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques (MPD et PCD). L'avantage de cette approche est de bénéficier d'une impression du relief qui permet de mieux consolider et valider le système final.

L'approche UML(Unified Modeling Langage), quant à elle, est un langage de modélisation standard des systèmes informatiques. Elle utilise des diagrammes pour représenter chaque aspect d'un système (statique, dynamique, etc.) en s'appuyant sur la notion d'orienté objet, qui est un véritable atout pour ce langage. UML permet ainsi une grande souplesse dans la modélisation de différents aspects de l'application. Cette approche est particulièrement adaptée pour la modélisation des applications à base d'objets. UML n'est cependant pas une méthodologie en soi, et doit être associée à une méthode de conception pour être utilisé de manière efficace. Dans ce cadre, le processus unifié est souvent utilisé comme méthode de conception, car il est centré sur les cas d'utilisation et sur l'architecture d'utilisation, et qu'il est itératif et incrémental.

Le processus unifié est :
- piloté par les cas d'utilisation,
- centré sur l'architecture d'utilisation,
- itératif et incrémental (une itération désigne la succession des étapes de l'enchaînement d'activités, tandis qu'un incrément correspond à une avancée dans les différents stades de développement).

UML est un moyen basé sur le langage naturel, il est accessible sans formation particulière par les utilisateurs, pour qu'ils puissent exprimer leurs attentes et leurs besoins en communiquant facilement avec les experts du domaine et les informaticiens. Il définit neuf diagrammes pour représenter les différents points de vue de la modélisation. Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues: **Les vues statiques**, c'est-à-dire représentant le système physiquement

- diagrammes des cas d'utilisations
- diagrammes des classes
- diagrammes d'objets
- diagrammes des composants
- diagrammes de déploiement

Les vues dynamiques, montrant le fonctionnement du système

- diagrammes des séquences
- diagrammes des collaborations
- diagrammes d'états-transitions
- diagrammes d'activités

La figure ci-dessous décrit le positionnement des neuf diagrammes d'UML

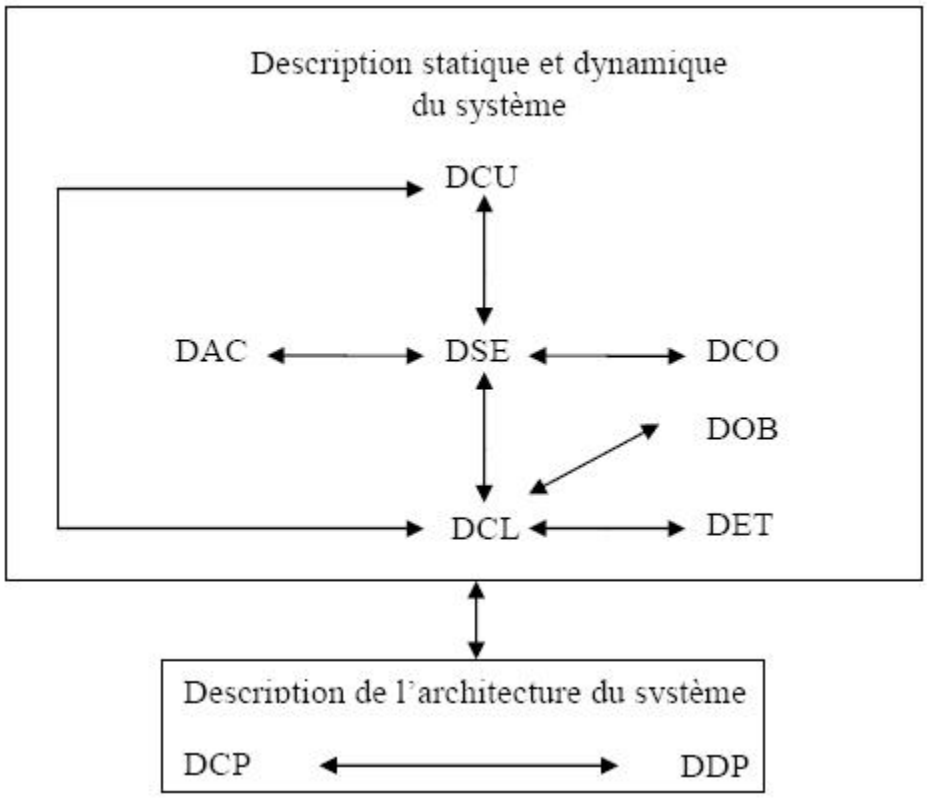


Figure 4- Positionnement des neuf diagrammes d'UML

DCU : diagramme des cas d'utilisation **DCL :** diagramme de classes. **DOB :** diagramme d'objets. **DET :** diagramme d'états-transitions. **DAC :** diagramme d'activités. **DSE :** diagramme de séquence. **DCO :** diagramme de collaboration. **DCP :** diagramme de composants. **DDP :** diagramme de déploiement. Dans notre étude de

conception nous allons utiliser cinque de ces diagrammes (diagramme des cas d'utilisation, diagramme de séquence, diagramme d'activités, diagramme d'état transition, diagramme de classes) pour modéliser les fonctionnalités de notre application.

3.2.3 Le modèle de cycle de vie

Lors de la conception et du développement d'un projet, il est primordial de choisir un modèle de cycle de vie à suivre. Ce choix aura un impact considérable sur le déroulement du projet ainsi que sur sa réussite. Il existe plusieurs modèles de cycle de vie, chacun présente des avantages et des inconvénients.

- Le modèle évolutif, par exemple, est basé sur une approche pluridisciplinaire qui vise à minimiser l'impact des changements des besoins en cours de projet. Ce modèle est souvent utilisé dans les projets où la flexibilité et l'adaptabilité sont essentielles. Le modèle de cycle de vie RAD est un exemple de modèle évolutif.

- Le modèle en cascade, quant à lui, découpe le projet en phases distinctes et linéaires, avec des livrables spécifiques à chaque étape. Ce modèle convient mieux aux projets où les exigences sont clairement définies et où le coût et le temps sont des facteurs importants. Le modèle de développement en W et le modèle de développement en V sont des exemples de modèles en cascade.

Enfin, le modèle objet se base sur la séparation de l'étude d'architecture de celle de l'étude fonctionnelle, afin de paralléliser au maximum les tâches. Ce modèle procède par itération, à l'instar du modèle évolutif. Le modèle de développement en Y est une variante du modèle objet.

Après avoir étudié les différents modèles de cycle de vie, nous avons opté pour le modèle de conception en W pour notre projet de fin d'études. Notre choix est basé sur l'importance accordée aux interfaces, ainsi que sur notre besoin de créer un prototype fonctionnel. Ce prototype nous permettra de valider les spécifications par expérimentation, conformément à la méthode de "Je saurai ce que je veux lorsque je le verrai !". Nous sommes convaincus que le modèle de conception en W nous permettra de mener à bien notre projet de manière efficace et rigoureuse.

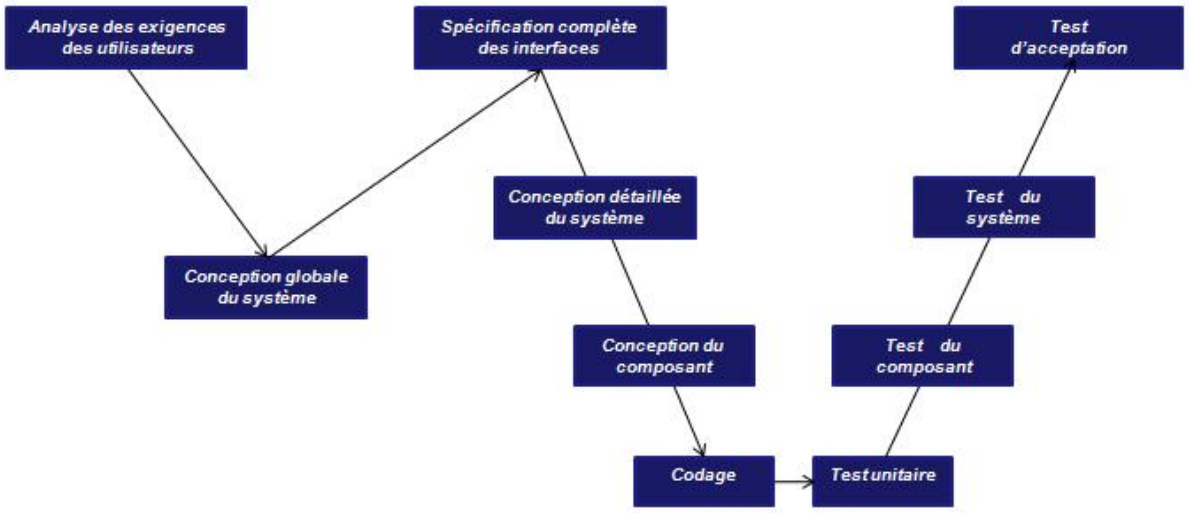


Figure 5:- Le modèle de développement en W

3.2.4 Modèle de conception

Le choix d'un modèle de conception pour un projet est une étape essentielle pour garantir sa réussite. Parmi les modèles existants, le Modèle-Vue-Contrôleur (MVC) est une architecture logicielle qui répond aux besoins de notre projet. Ce design pattern est constitué de trois éléments clés : le modèle qui représente les données et les règles métiers de l'application, la vue qui s'occupe de l'affichage des données à l'utilisateur, et le contrôleur qui agit comme un intermédiaire entre les deux en traitant les actions de l'utilisateur et en mettant à jour les données et la vue en conséquence.

Le MVC permet une séparation claire des responsabilités, facilitant ainsi la maintenance et l'évolution du code. Il offre également une grande flexibilité, permettant aux développeurs de travailler sur différentes parties de l'application en parallèle. Cette approche encourage la modularité et la réutilisation du code, contribuant ainsi à une meilleure qualité logicielle.

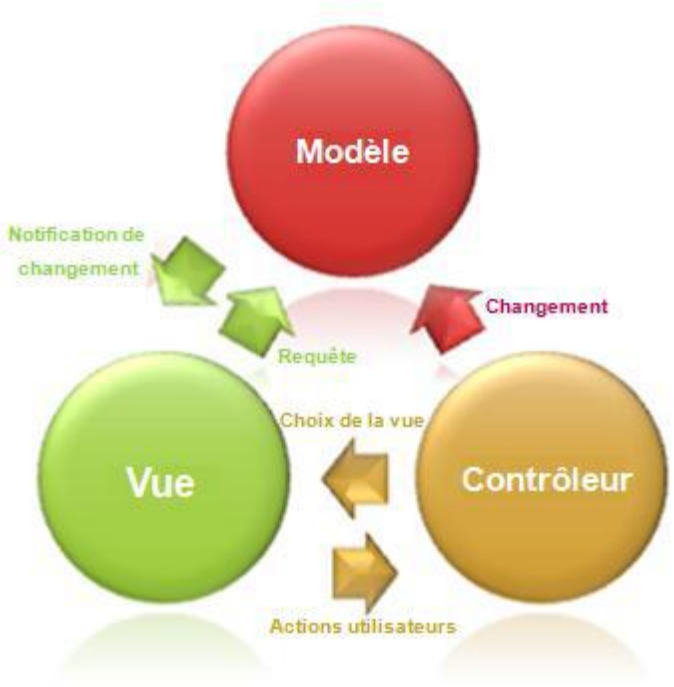


Figure6 - Le modèle de conception MVC

Modèle: représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au cœur du métier.

Vue : représente l'interface utilisateur. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.

Contrôleur : Le contrôleur se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate. Il ne doit faire aucun traitement. Il ne fait que de l'interception et de la redirection. [1]

3.2.5 Architecture générale de l'application

L'application interagit avec Firebase en utilisant l'API Firebase fournie par Google. Cette API permet à l'application de communiquer avec Firebase en utilisant des fonctions prédéfinies pour réaliser différentes opérations, telles que la lecture et l'écriture de données, l'authentification des utilisateurs, la gestion des notifications push, etc.

Lorsque l'application envoie une requête à Firebase, l'API Firebase se charge de transmettre cette requête au serveur Firebase approprié pour le traitement. Le serveur Firebase traite ensuite la requête et renvoie une réponse à l'API Firebase, qui la transmet à l'application.

Il est important de noter que la sécurité des données est une préoccupation majeure lors de l'interaction de l'application avec Firebase. Firebase permet aux développeurs de configurer des règles de sécurité pour leur base de données, afin de contrôler qui a accès à quelles données et dans quelles conditions. Les règles de sécurité sont écrites en utilisant une syntaxe spécifique et sont hébergées sur le serveur Firebase. Lorsque l'application effectue une opération de lecture ou d'écriture, les règles de sécurité sont vérifiées pour s'assurer que l'utilisateur a les autorisations nécessaires pour effectuer cette opération.

La figure suivante présente l'architecture générale de notre application.

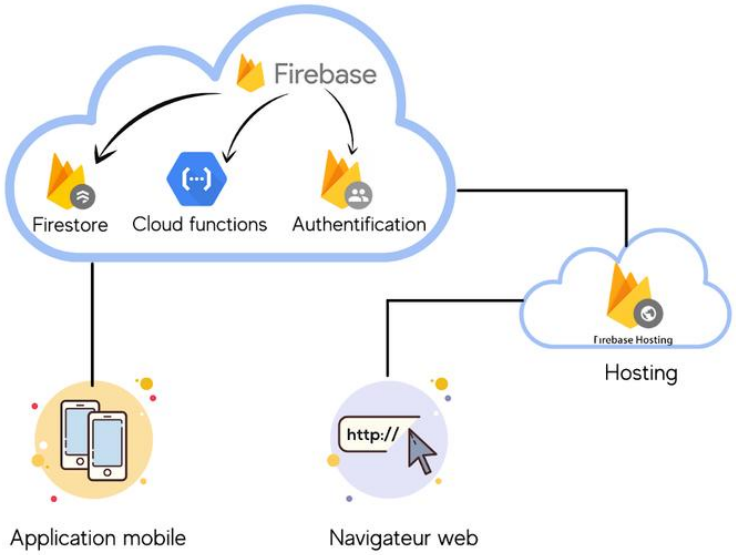


Figure 7- Architecture générale de l'application

3.3 Conception détaillée

3.3.1 Diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML Nous avons donc choisie ce diagramme puisqu'il nous permet de traiter la dynamique du système d'information, c'est-à-dire les opérations qui sont réalisées en fonction d'événements et par la suite mieux comprendre le fonctionnement du système.

3.3.1.1 Diagramme de séquences « Inscription/Authentification »

L'authentification est une étape primordiale que chaque utilisateur doit y passer pour accéder à l'application après l'inscription, cette phase assure, en effet, la sécurité de l'application. En demandant l'accès à l'application, l'utilisateur se voit dans l'obligation de s'Authentifier à travers son compte. L'application vérifie l'existence de ce compte dans sa base de données. Si l'utilisateur est identifié dans la base, il accède immédiatement à l'application. La figure ci-dessous présente le diagramme de séquence de l'authentification.

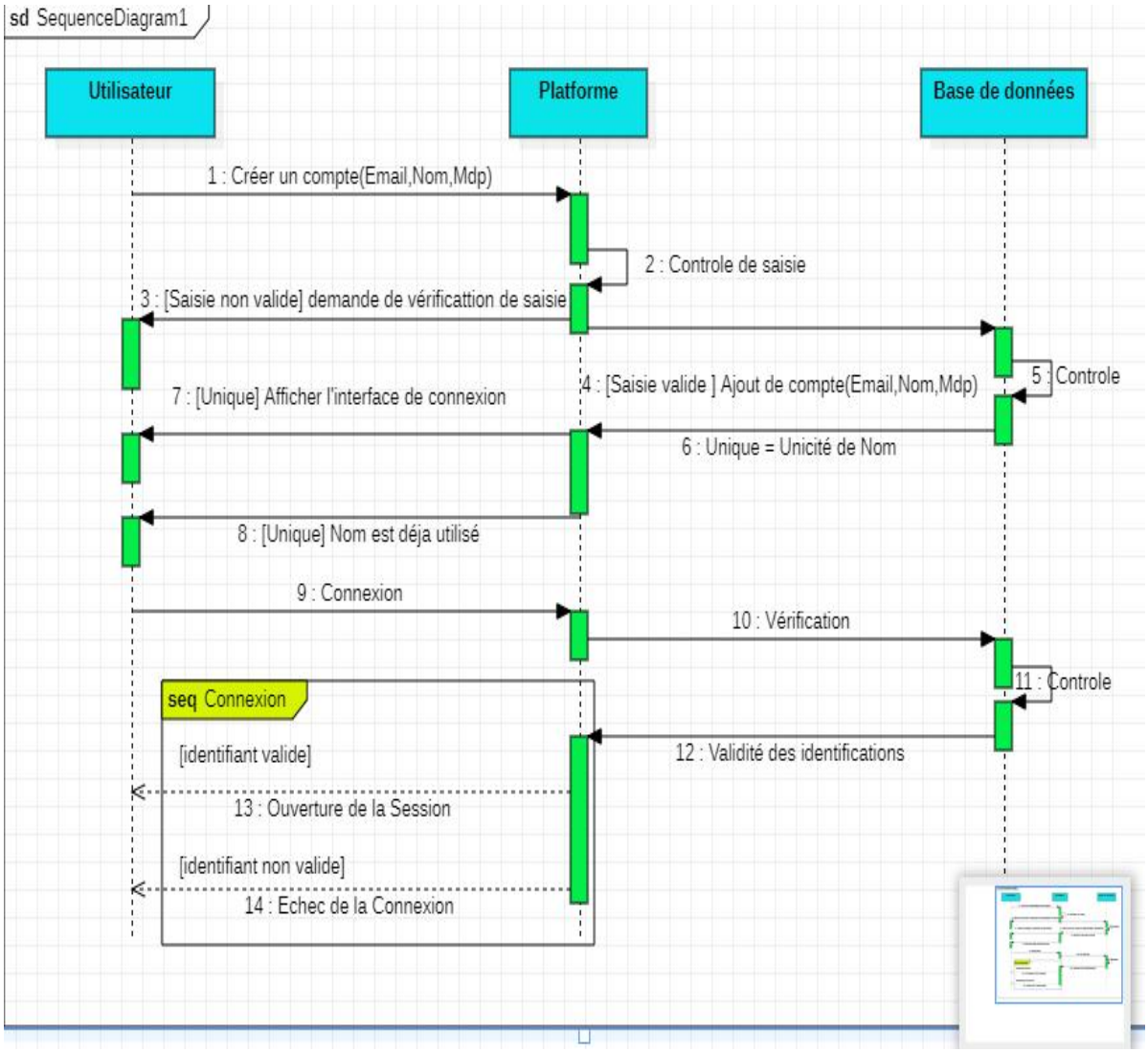


Figure 8- Diagramme de séquence pour « Inscription/Authentification »

3.3.1.2 Diagramme de séquence « Suivie des véhicule »

Ce digramme exlique bien les séquences qui se déroulent entre l'utilisateur, platforme et la base de donnée lors de localisation de sa véhicule sur la Map.

L'utilisateur entre dans la liste des véhicules choisie sa véhicule un seul click sur elle alors il s'affiche son profil il peut localiser son véhicule.

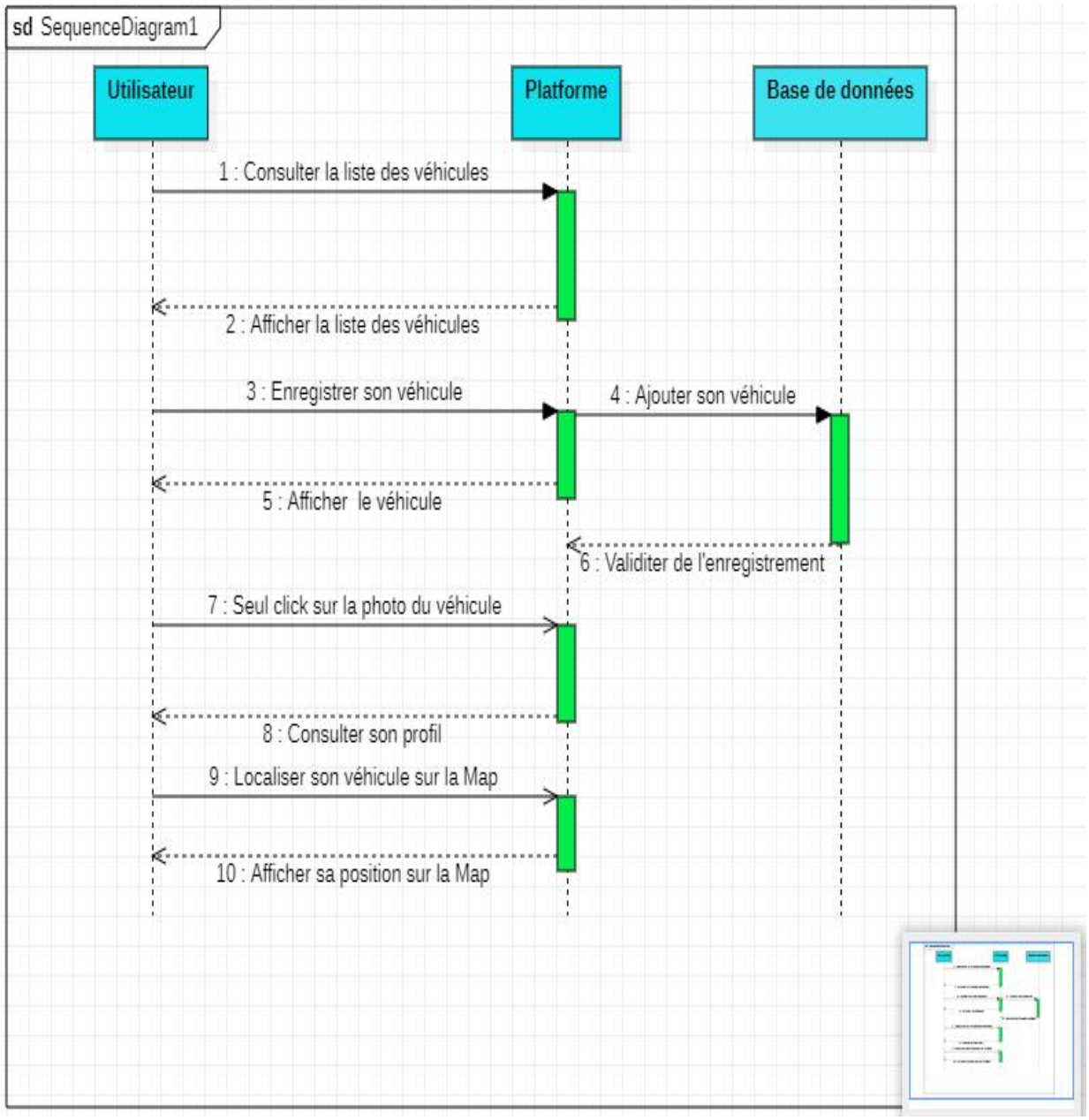


Figure 9- Diagramme de séquence pour « Suivie la véhicule »

3.3.2 Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles-ci.

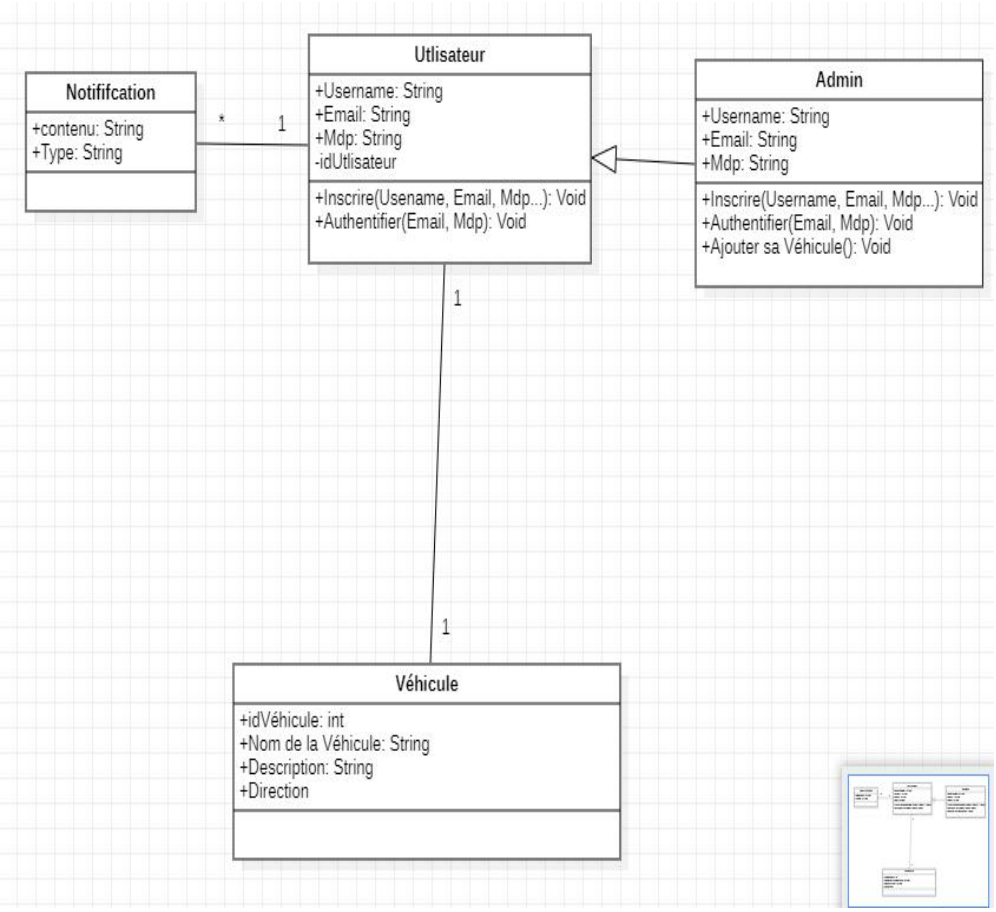


Figure 10- Diagramme de classe

Ce diagramme illustre les classes et les entités de ce projet ainsi que les cardinalités et les associations qui les relient.

3.3.3 Diagramme d'activité :

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagramme de classes ou de cas d'utilisation, par exemple:

3.3.3.1 Diagramme d'activité : « Inscription/Authentification »

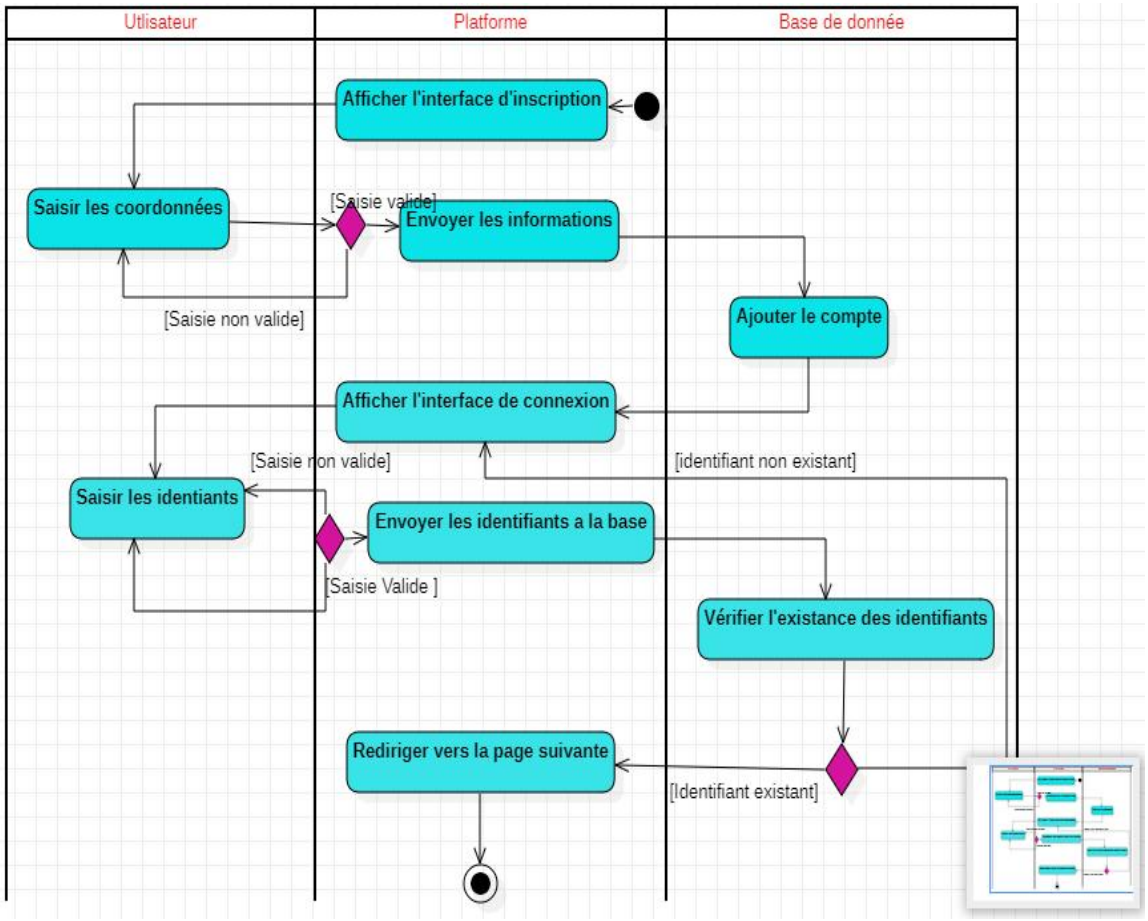


Figure 11- Diagramme d'activité pour « Inscription/Authentification »

Ce diagramme explique bien les étapes et les activités suivies pour l'authentification et les conditions qu'il faut les respecter en s'authentifiant pour éviter les erreurs et les alertes. Le système donne en premier lieu à l'utilisateur un formulaire à remplir, si les champs sont bien remplis et toutes les données entrées sont compatibles aux celles à la base l'authentification est réussie, si non, des alertes et des erreurs seront affiche par le système afin de guider l'utilisateur à éviter ces erreurs et à réussir cette phase d'authentification.

3.3.3.2 Diagramme d'activité : «Localiser la véhicule »

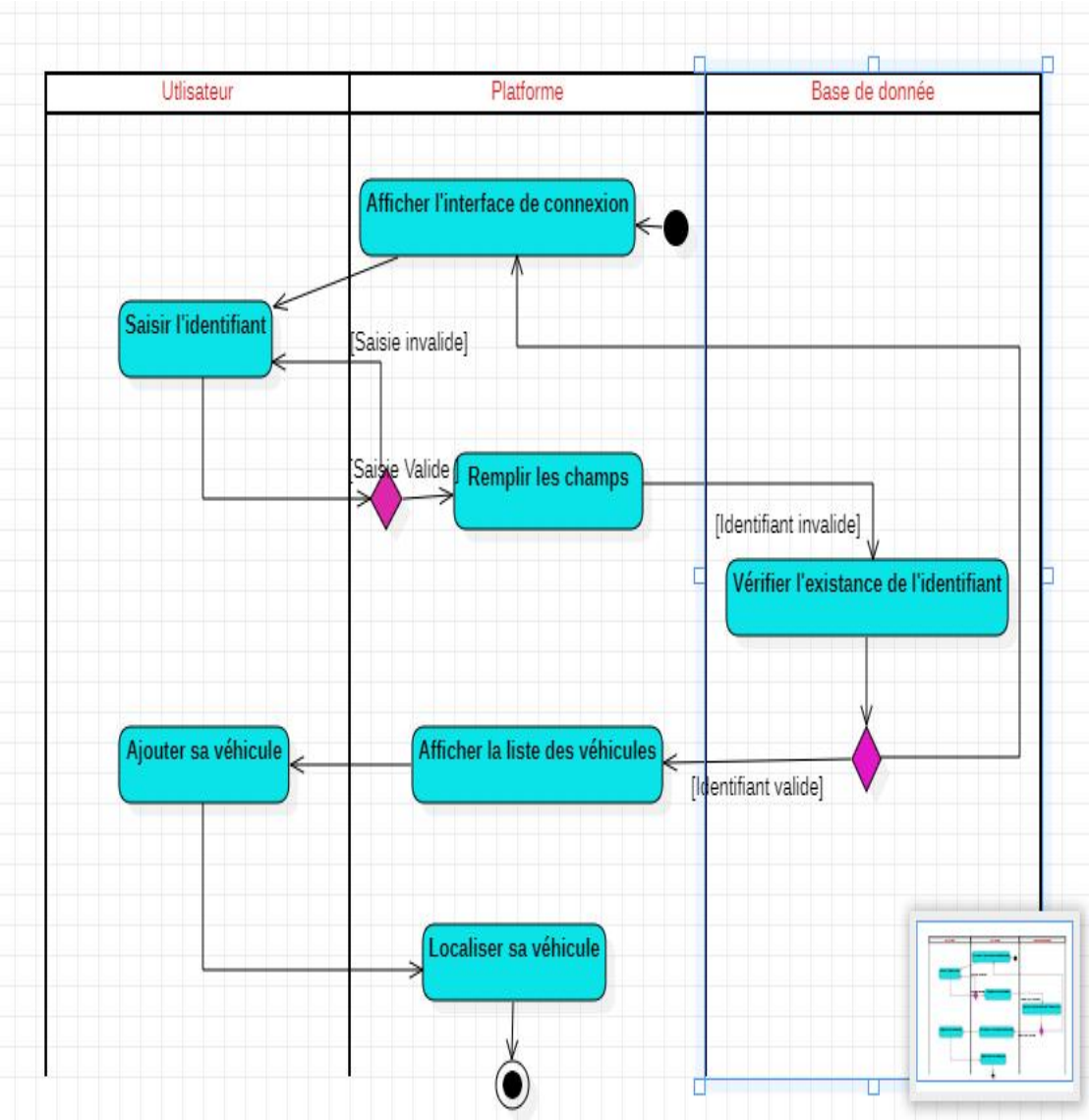


Figure 12 - Diagramme d'activité pour « Localiser la véhicule »

3.4 Conclusion

Ce chapitre a été présenté sous forme de deux grandes parties : la conception générale et la conception détaillée. Concernant la conception générale, nous avons choisi la méthodologie el le langage de modélisation convenables. Quant à la conception détaillée, nous l'avons présenté suivant l'approche UML. Le chapitre suivant présentera la réalisation de cette conception et la description des interfaces réalisées.

Chapitre 4 : Réalisation

4.1 Introduction

Ce chapitre présente les résultats obtenus dans le cadre de ce projet de recherche, ainsi que les différents outils matériels et logiciels qui ont contribué à sa réalisation. La phase de conception a permis d'obtenir une vision globale du système étudié et des interactions entre ses composants. Dans cette partie, nous procéderons à une exposition détaillée de l'environnement matériel utilisé dans le cadre de ce projet, ainsi que des outils de développement qui ont été employés. Par la suite, nous présenterons en détail les différents modules qui ont fait l'objet de notre travail. L'objectif de cette section est de fournir une vue d'ensemble approfondie du travail réalisé, en mettant en évidence les ressources utilisées et les étapes clés du processus de développement.

4.2 Aspect technique

4.2.1 Choix de la plateforme

Les architectures des applications informatiques et les technologies utilisées afin de mettre ces dernières en œuvre ne cessent d'évoluer. Chacune a ses avantages et ses inconvénients et se différencie selon le domaine d'utilisation. Le choix d'une telle architecture et d'une telle technologie dépend des besoins du concepteur et du développeur afin de satisfaire les contraintes exigées. Dans notre projet nous avons choisi la plateforme de développement Android.

4.2.1.1 La plate-forme Android

Afin de choisir la plateforme de développement adéquate il est nécessaire de faire une étude sur les différentes plateformes conçues pour les applications de mobilités. La démographie nous montre beaucoup de choses à propos des utilisateurs Android et iOS. En effet, Android détient actuellement la plus grande part du marché, au niveau mondial, des plateformes mobiles.

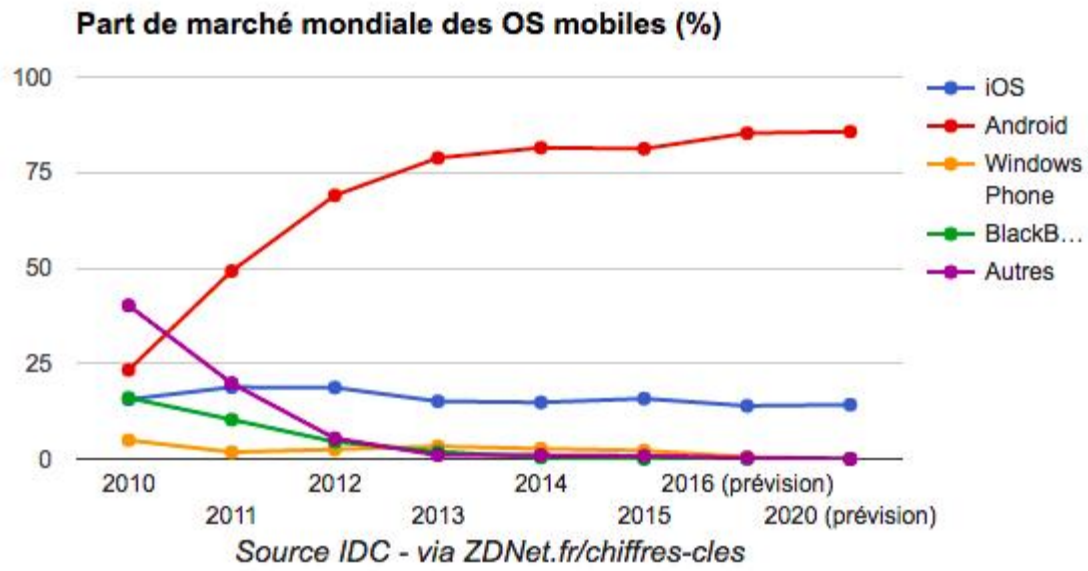


Figure 13- Comparaison entre les plateformes mobiles sur le marché mondial

Lorsque les marques choisissent une plateforme pour recueillir leurs applications mobiles, elles se tournent principalement vers l'iPhone (73%), suivi de l'iPad (35%), d'Android (28%)

et de RIM (17%). Windows Mobile et Symbian profitent de la volonté des marques de diversifier les plateformes avec respectivement 8% et 6% de parts du marché. En termes d'évolution, toutes les plateformes évoluent sauf Palm, qui s'effondre par rapport à 2009 pour n'atteindre que 2% des marques. iPad et Android connaissent les meilleurs progression avec respectivement 35 et 18 points en plus par rapport à 2009. Android et iPad disposent encore de larges viviers de croissance par rapport à iPhone qui a déjà convaincu une majorité de responsables de marques. Lorsqu'on leur demande quelle plateforme les annonceurs souhaitent intégrer en 2011, Android prend la tête du classement (55%), suivi d'iPad (45%) et de Windows Phone 7 (24%). Il est clair donc que développer sur Androïd est aujourd'hui tout à fait pertinent, il s'agit d'un vrai pari sur le futur, avec une cible qui deviendra à coups sur plus large que la concurrence. [2]

Android possède plusieurs avantages :

- Android est une plateforme puissante, moderne, sûre et ouverte. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer,

d'agrandir et de replacer les composants existants. Les utilisateurs peuvent adapter les applications à leurs besoins.

- Android est basé sur le noyau Linux. Alors, il y a plusieurs avantages comme une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partagé, etc.
- Le SDK de l'Android offre complètement les APIs pour développer l'application sur Android. Il intègre un émulateur (qui reproduit à l'identique le comportement du téléphone) gratuit, à l'inverse de celui de l'iPhone.

4.2.2 Choix de l'architecture Client/serveur

Pour la réalisation de notre application, nous avons choisi une architecture client/serveur 3 tiers basée sur une application web. L'architecture 3-tiers est un modèle logique qui vise à clairement à séparer trois couches logicielles au sein d'une même application ou système. Cette approche permet de modéliser et présenter l'application comme un empilement de trois couches distinctes, chacune ayant un rôle défini. Dans notre cas, les données sont centralisées sur un serveur de données, tandis que l'application est exécutée sur un serveur d'application, qui est un serveur web dans notre contexte. Cette séparation offre plusieurs avantages, notamment en termes de sécurité et de gestion des mises à jour des données et des systèmes de gestion de base de données.Nous avons choisi d'utiliser Firebase comme serveur de données pour notre application. Firebase est une plateforme de développement d'applications mobiles et web qui fournit des fonctionnalités de gestion de base de données en temps réel, d'authentification des utilisateurs, de stockage de fichiers et d'autres services cloud. Son utilisation simplifie les contrôles de sécurité et facilite la gestion des données, tout en offrant une bonne évolutivité.L'architecture client/serveur est largement adoptée car elle repose sur des technologies matures et offre une meilleure sécurité. Lors de la connexion, un ordinateur client ne peut accéder qu'au serveur et n'a pas accès aux autres ordinateurs clients connectés au réseau. De plus, les serveurs sont généralement bien protégés contre les attaques des pirates.Cette architecture facilite également l'évolution de l'application, car il est facile d'ajouter ou de supprimer des clients et des serveurs selon les besoins. De plus, elle permet de minimiser les pannes, car les couches sont séparées, ce qui réduit l'impact des erreurs ou des problèmes sur l'ensemble du système.

Dans ce type d'architecture, le programme client est responsable des fonctions de présentation et de contrôle de l'application. Il fait appel au serveur de données, dans notre cas Firebase, pour assurer la persistance des données. La figure suivante illustre le fonctionnement général de cette architecture.

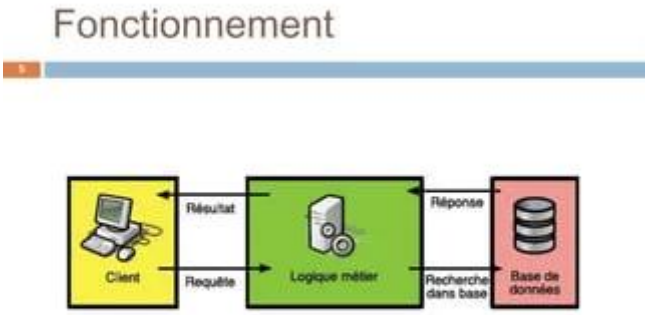


Figure 14- Architecture trois tiers

4.2.1.2 Architecture MVC

L'architecture MVC (Modèle – Vue – Contrôleur) prends place [interface utilisateur] dans le but de mieux structurer une application avec une interface graphique:

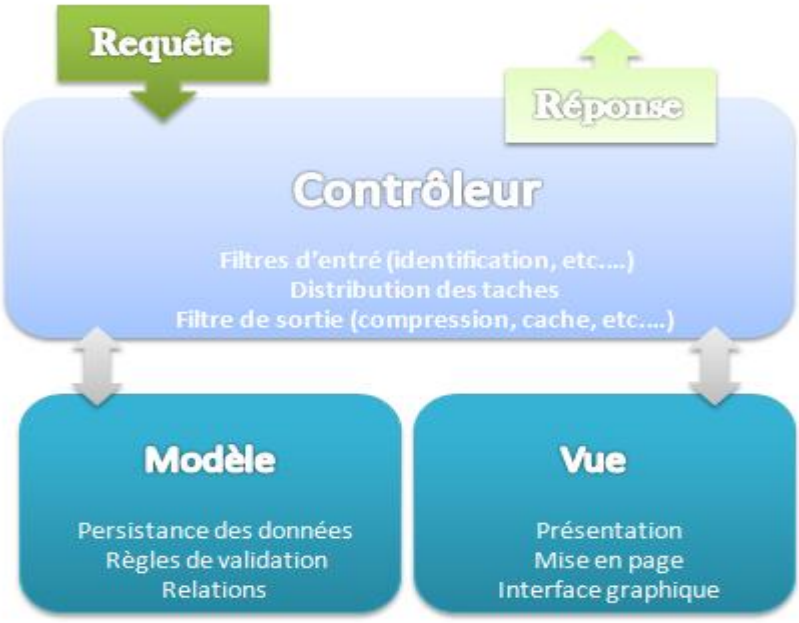


Figure 15- Architecture MVC

Ce modèle est un concept d'architecture qui propose une séparation en trois entités des données, des traitements et de l'interface :

- ✓ Le Modèle représente les données de l'application généralement stockées dans une base de données .
- ✓ La Vue correspond à l'IHM (Interface Homme Machine).
- ✓ Le Contrôleur assure les échanges entre la vue et le modèle notamment grâce à des composants métiers .

4.3 Environnement logistique

4.3.1 Environnement et outils de développement

Les principaux outils qui ont contribué à la qualité du développement sont :

StarUML

L'UML (Unified Modeling Language) est un langage de modélisation graphique utilisé dans le domaine du génie logiciel pour représenter visuellement le système logiciel à développer. Il fournit un ensemble de notations standardisées permettant de représenter les différentes perspectives d'un système, notamment sa structure, son comportement, ses interactions et ses contraintes.

Android Studio

Android Studio est l'environnement du développement intégré officiel pour le développement d'applications Android, basé sur IntelliJ IDEA. En plus de l'éditeur de code et des outils développeurs puissants d'IntelliJ, Android Studio offre encore plus de fonctionnalités qui améliorent votre productivité lors de la création des applications Android, telles que:

- Un système de construction flexible basé sur Gradle
- Un émulateur rapide et riche en fonctionnalités
- Un environnement unifié qui permet de développer pour tous les appareils Android
- Instant Run pour pousser les modifications à l'application en cours sans créer de nouveau APK

- Les modèles du code et l'intégration de GitHub pour aider à créer des fonctionnalités communes d'application et à importer un exemple de code
- Des outils et des cadres de tests étendus
- Outils de peluches pour attraper les performances, la facilité d'utilisation, la compatibilité de la version et d'autres problèmes
- Support C ++ et NDK
- Prise en charge intégrée de Google Cloud Platform , permettant d'intégrer Google Cloud Messaging et App Engine [3]

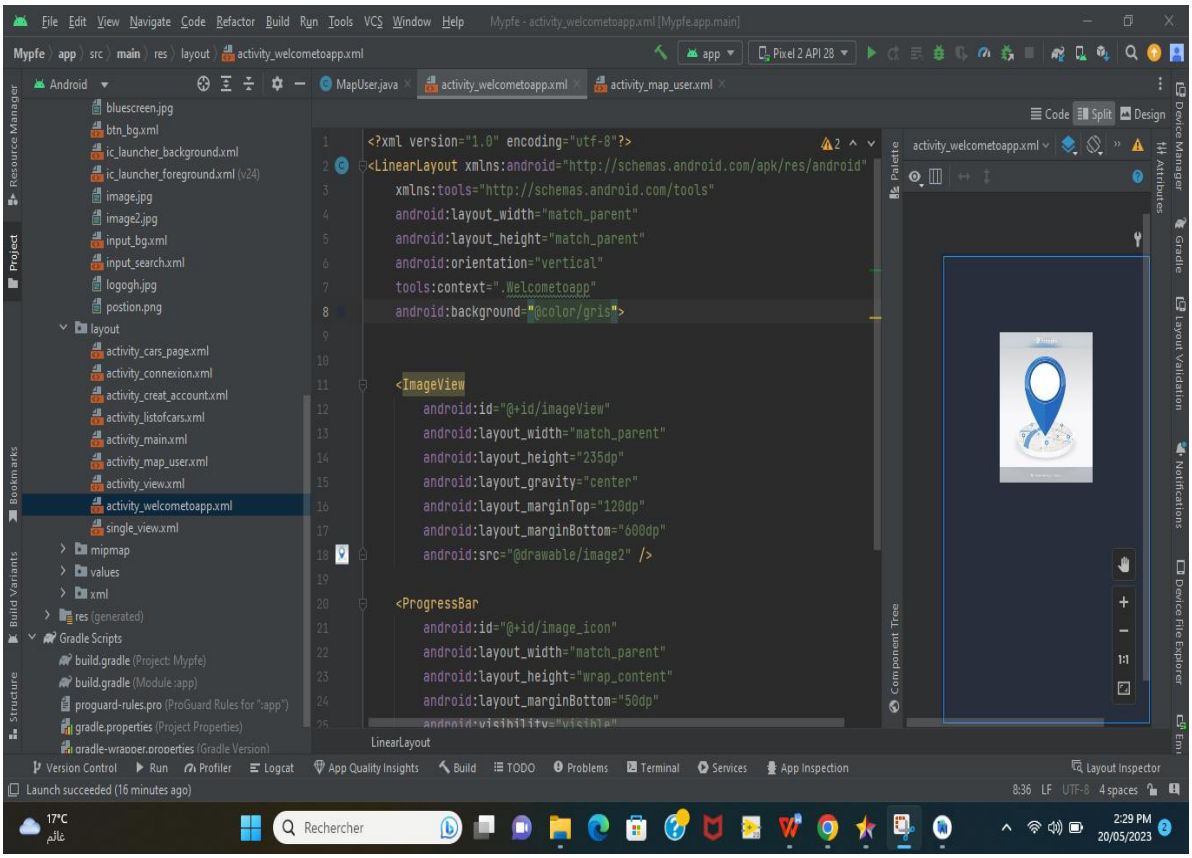


Figure16- Interface de l'EDI Android Studio

Firebase :

Firebase est une plateforme de développement d'applications mobiles et web développée par Google. Elle offre un ensemble d'outils et de services pour faciliter la création, le déploiement et la gestion d'applications en ligne, en fournissant une infrastructure évolutive et une base de données en temps réel.

Adobe Photoshop CS5

C'est un programme d'édition graphique, il permet d'effectuer un nombre impressionnant de retouches sur images. La mise à l'échelle intelligente par exemple, permet d'étendre ou de rétrécir une photo dans de multiples sens sans qu'il n'y ait pour autant de déformation de celle-ci sur le résultat final. Les formes et les distances seront déterminées et respectées automatiquement par cette fonctionnalité.

4.3.2 Langages de programmation utilisés

4.3.2.1 JAVA

Le langage Java est un langage de programmation informatique orienté, objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et Framework associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

4.3.2.2 XML

Le XML, acronyme de eXtensible Markup Language (qui signifie: langage de balisage extensible), est un langage informatique qui sert à enregistrer des données textuelles. Ce langage, grosso-modo similaire à l'HTML de par son système de balisage, permet de faciliter l'échange d'information sur l'internet.

Grâce au vocabulaire XML Android, la conception des interfaces utilisateur et les éléments de l'écran qu'ils contiennent, de la même manière de créer des pages web en HTML avec une série d'éléments imbriqués.

4.3.3 API utilisé

4.3.3.1 Osmdroid (Open Street Map) Android API

Osmdroid est une bibliothèque open-source qui permet d'ajouter des cartes utilisant des données OpenStreetMap à une application Android. Cette bibliothèque offre des fonctionnalités similaires à l'API Google Maps Android, mais en utilisant les données d'OpenStreetMap.

En utilisant Osmdroid, vous pouvez facilement intégrer des cartes OpenStreetMap dans votre application Android. La bibliothèque gère automatiquement l'accès aux serveurs OpenStreetMap, le téléchargement des données cartographiques, l'affichage de la carte et la réponse aux interactions de l'utilisateur.

Tout comme l'API Google Maps Android, Osmdroid permet d'ajouter des marqueurs, des polylignes et des polygones à la carte. Vous pouvez également ancrer des éléments graphiques bitmap à des emplacements spécifiques sur la carte en utilisant les superpositions au sol. De plus, Osmdroid prend en charge les superpositions de tuiles, qui sont des images affichées au-dessus des tuiles de la carte de base.

En résumé, Osmdroid est une alternative open-source à l'API Google Maps Android, utilisant les données d'OpenStreetMap. Il offre des fonctionnalités similaires pour ajouter des cartes, des marqueurs, des polylignes, des polygones et des superpositions à une application Android, en utilisant les données cartographiques d'OpenStreetMap.

4.4 Travail réalisé

Nous sommes proposé de réaliser une application mobile. Cette application permet aux utilisateurs de localiser leurs véhicules dans une carte Map et de suivre leurs Trajectoire en ligne .

4.4.1 Présentation les interfaces de la Base de données

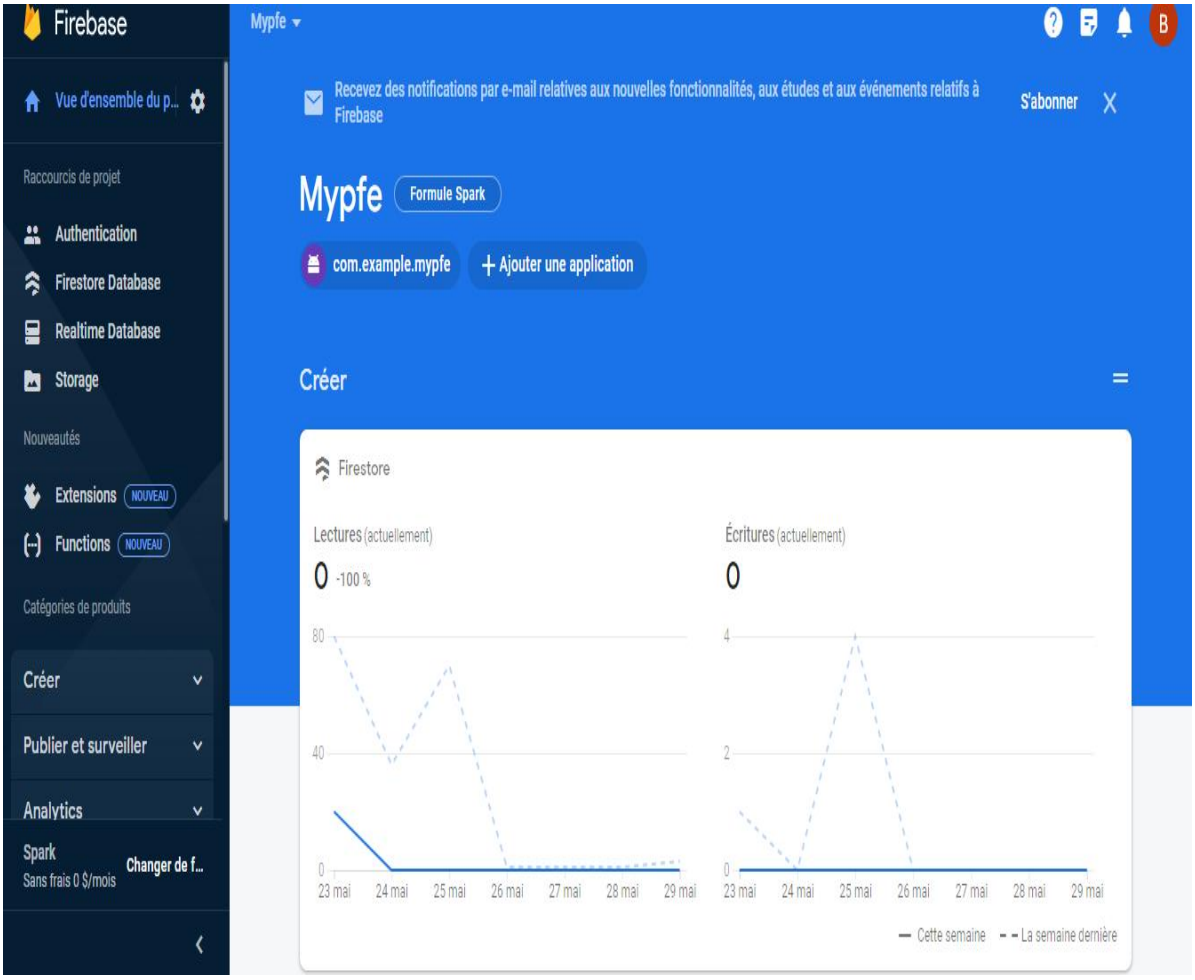


Figure17- Interface de la base de données Firebase

Pour que la base de données de l'application soit bien sécurisé en active le service de l'Authentification cette activation s'effectue dans mon projet (My pfe) associée a mon application mobile . En effet , il ya mise en place des fonctionnalités permettant aux utilisateurs de s'inscrire avec leur adresse e-mail et de créer un mot de passe sécurisé, ainsi que de se connecter avec leurs informations d'identification et pour que cette authentification soit parfaitement sécuriser il ya utilisation des protocoles de chiffrement et de sécurité fournis par Firebase pour garantir la confidentialité des informations d'identification des utilisateurs lors des processus d'inscription et de connexion. Cette Authentification sera completement réuissite si et seulement si il ya gestion des utlisateurs c'est à dire création d'un profil utilisateur unique pour chaque utilisateur, associant leur adresse e-mail et leur UID généré par Firebase .Enfin, pour mettre ma base de donnée en relation avec l'application il faut

Installer SDK Firebase pour connecter l'application mobile à l'authentification Firebase, permettant aux utilisateurs de s'inscrire, de se connecter et d'interagir avec leur compte.

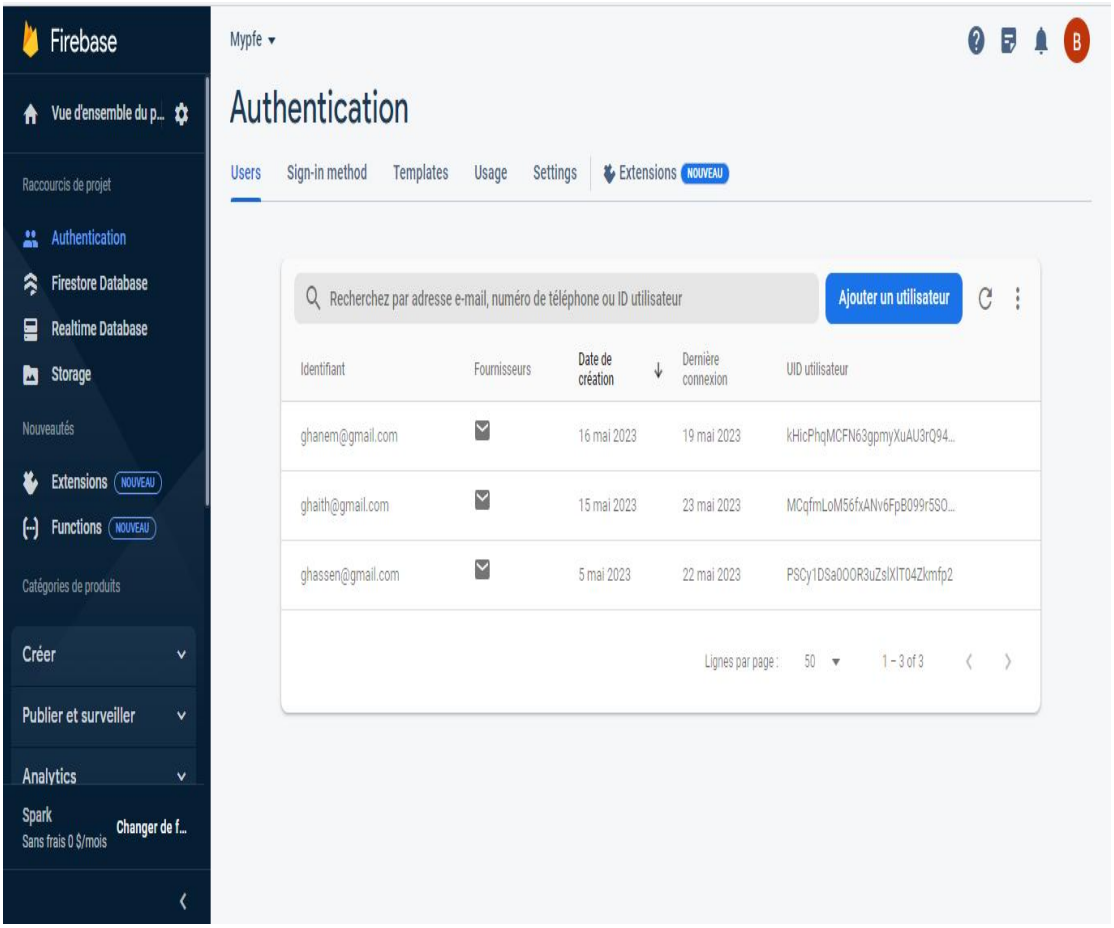


Figure18- Interface de l'Authentification dans Firebase

Une fois l'utlisateur s'authentifie , il est obligatoirement de stocker tous les utlisateurs dans la base de données Firestore . Tout d'abord , modélisation des données de l'utlisateurs tel que Utilisation d'une collection "utilisateurs" contenant des documents individuels pour chaque utilisateur, avec des champs tels que l'email, le nom d'utilisateur et le statut isAdmin ou isUser . Ensuite , utilisation des règles de sécurité Firestore pour contrôler les autorisations d'accès aux données des utilisateurs. En dernier lieu , il est obligatoirement de faire l'interaction de l application avec la

base de données au temps réel et cela d'utliser le SDK Firestore pour connecter l'application mobile à la base de données firestore.

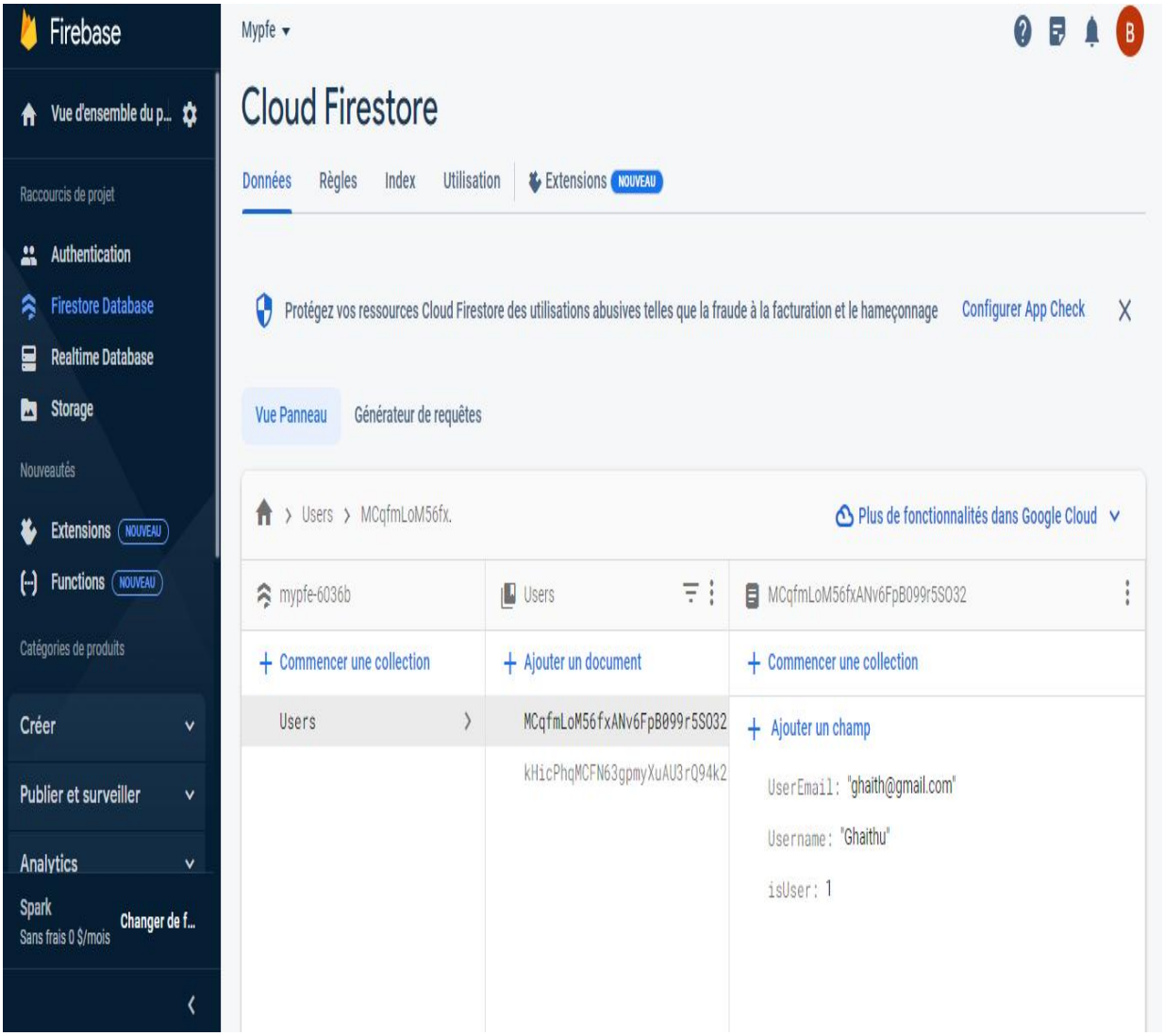


Figure19- Interface de Firestore Database

La réalisation d'une base de données en temps réel pour une application mobile, contenant le nom des voitures et leurs URL d'image, implique la configuration d'une base de données en temps réel dans Firebase. Les informations des voitures sont stockées dans des documents ou nœuds individuels, avec des champs pour le nom et l'URL de l'image. Les opérations d'écriture et de lecture en temps réel sont utilisées pour ajouter et récupérer les données des voitures. Les images peuvent être stockées dans Firebase Storage, tandis que les URL correspondantes sont enregistrées dans la base de données. L'application mobile est intégrée à la base de données en utilisant les

SDK Firebase, permettant ainsi d'afficher les informations des voitures, y compris leurs noms et leurs images, en temps réel.

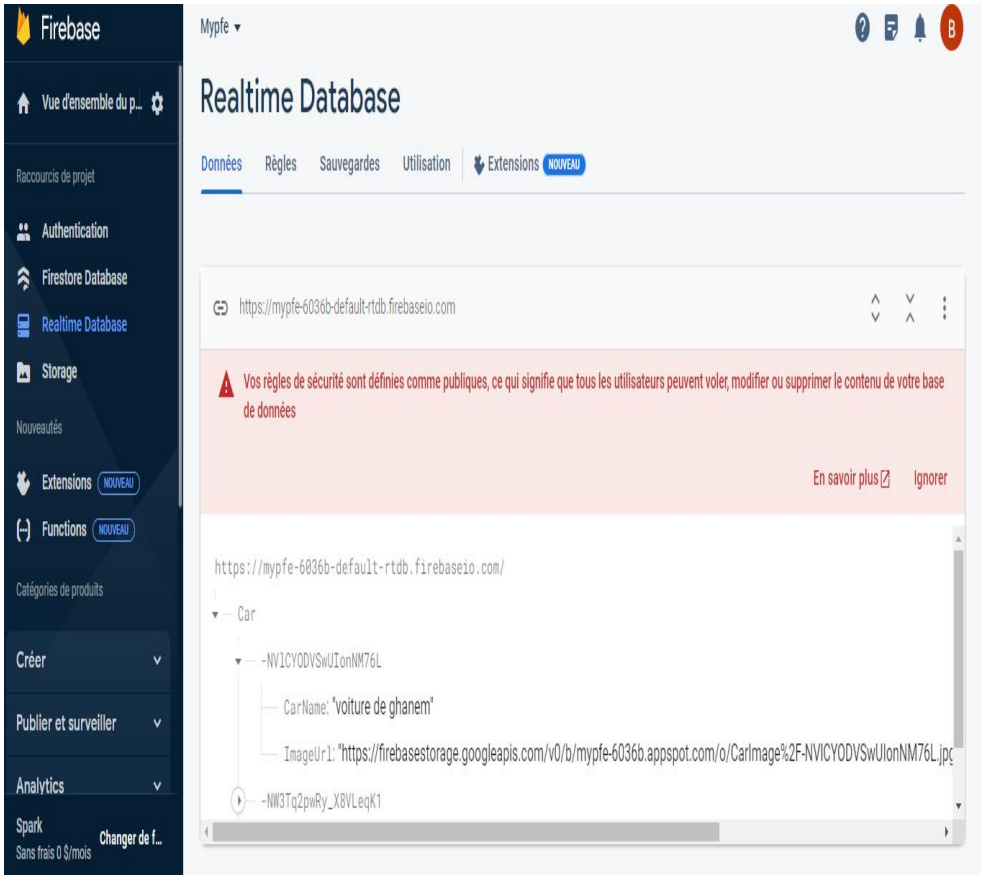


Figure20- Interface de Realtime Database

Lors de l'enregistrement des information des voiture sont ajoutées a la base de données en temps réel immédiatement les images des voitures sont enregistrées dans le stockage Firebase. Les images correspondantes sont téléchargées vers le stockage Firebase, et les URL des images sont enregistrées dans la base de données. L'application mobile peut ensuite récupérer les URL d'image à partir de la base de données et charger les images à partir du stockage Firebase pour les afficher aux utilisateurs. Enfin ,il est necéssaire d' utiliser SDK Firebase pour connecter l'application mobile au stockage Firebase, permettant ainsi de récupérer les images des voitures à partir des URL enregistrées dans la base de données en temps réel.

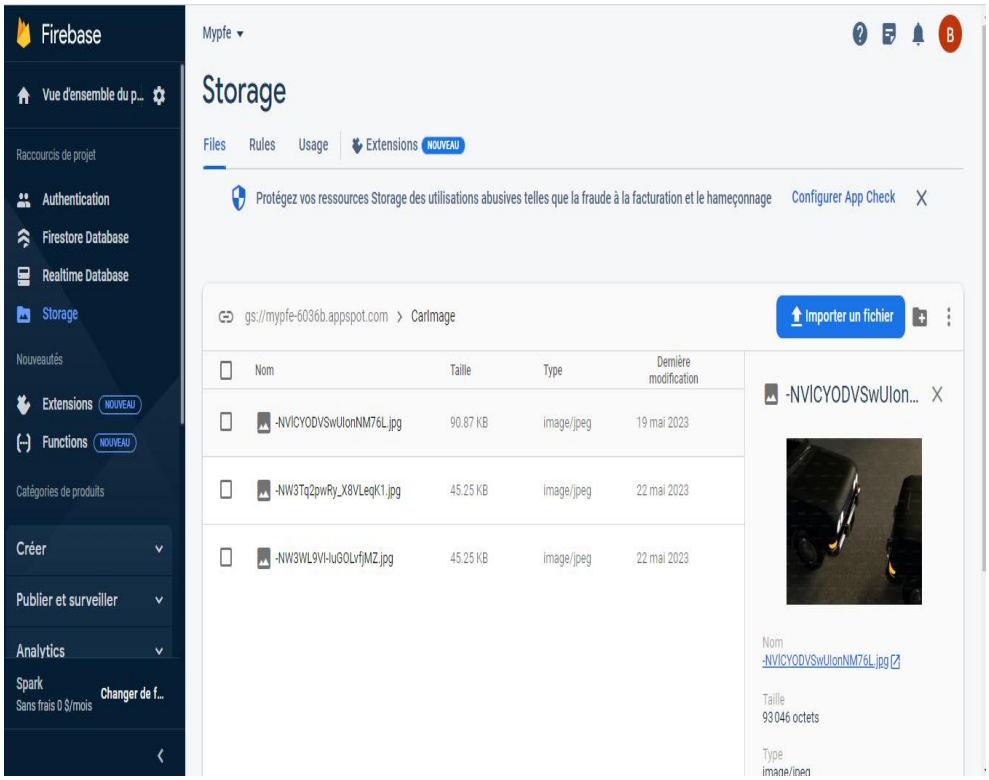


Figure21- Interface de Storage Database

4.4.2 Présentation les interfaces d'application

A l'ouverture de l'application, un écran de démarrage s'affiche pour 4 secondes contenant le logo et un Progress Bar qui tourne jusqu'a l'ouverture de la page suivante.

La figure ci-dessous présente l'écran de démarrage

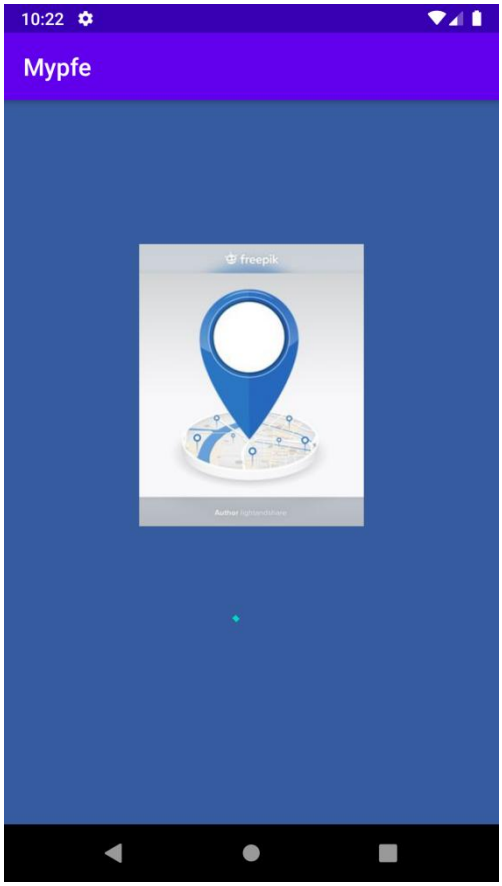


Figure 22- Ecran de démarrage de l'application

Après que la page d'accueil s'affiche, cette interface pour la Connexion et à partir de la page de Connexion on peut faire l'inscription en cliquant sur le Edit text «Sign Up».

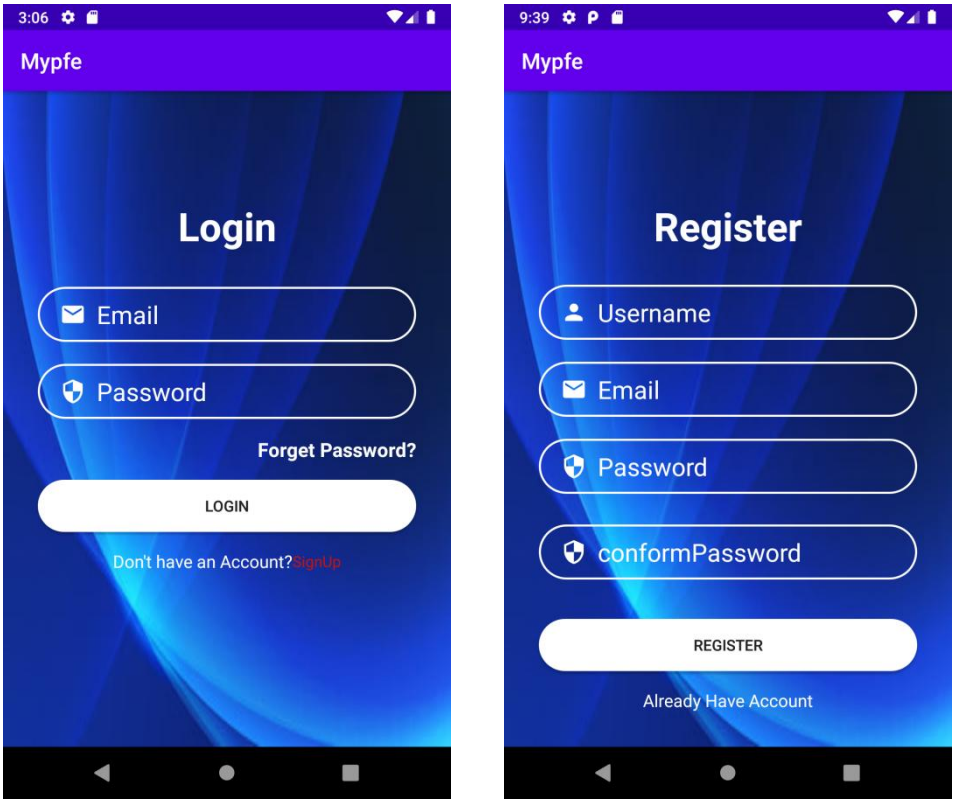


Figure 23- Interfaces, de Connexion et d'inscription.

L'authentification est une étape primordiale que chaque utilisateur de notre système doit y passer pour accéder à l'application. Cette phase assure, en effet, la sécurité de l'application. En demandant l'accès à l'application, l'utilisateur se voit dans l'obligation de S'authentifier à travers son compte. L'application vérifie l'existence de ce compte dans sa base de données de contenu. Si l'utilisateur est identifié dans la base, il accède à l'application.

Si jamais l'utilisateur a commis une erreur au niveau du type de champs de login par exemple ou au niveau de longueur de mot de passe , des messages d'erreur s'affichent à l'aide des validateurs de ces deux champs comme indiqué dans la figure suivante :

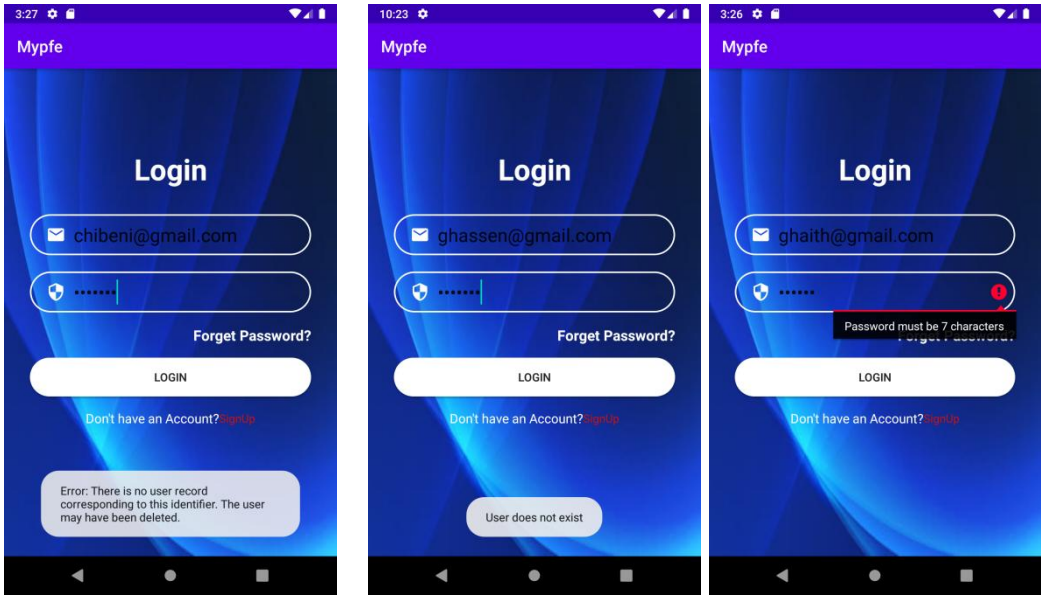


Figure 24- Message d'erreur affiché suite à une saisie invalide.

Une fois l' utlisateur s'est authentifié , il entre dans l'interface liste des véhicules et aprés doit ajouter son véhicule dans l'application en cliquant sur l'icon de l'ajout qui se trouve a l'interface de la page , il se rédirigé vers la page avec laquel en permet d'ajouter le véhicule . Lors de l'ajout du véhicule l'application se communiquer avec la base de donnéés en enregistrant le véhicule dans le Storage de l Firebase en attribuant son nom et son ID et enfin il trouve la Véhicule dans la liste des voiture avec son nom et sa photo.

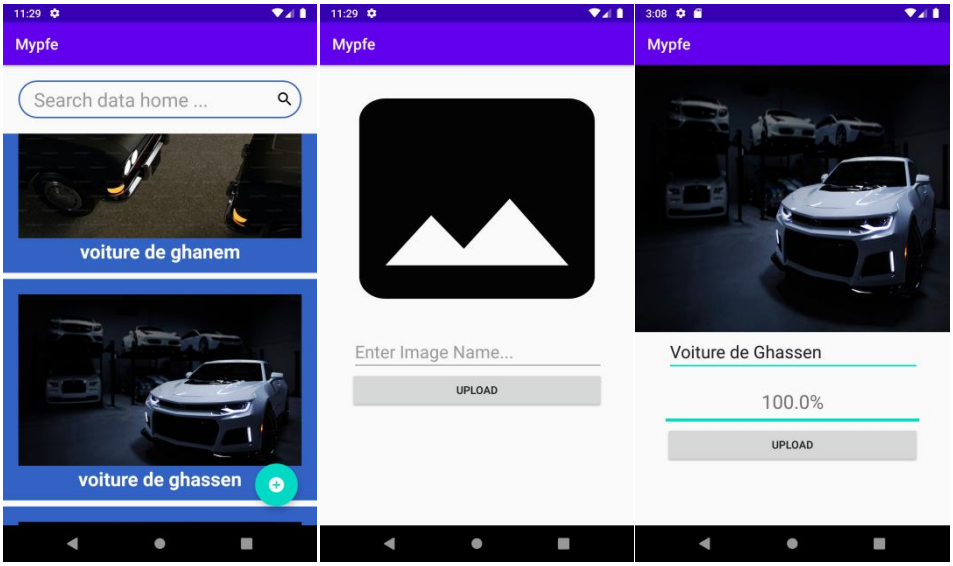


Figure 25- Interface affichant la liste des véhicules , une interface d'ajout des véhicules.

Aprés l'enregistrement du véhicule dans l'interface de liste des véhicule l'utilisateur clique sur la photo de son véhicule , il se redirige vers la page de son profil qui est sous forme d'un menu glissant. Le premier élément de ce menu est une page nommée « Home » qui contient la photo du véhicule ainsi que sa description.

La figure suivante présente l'interface de la page Home :

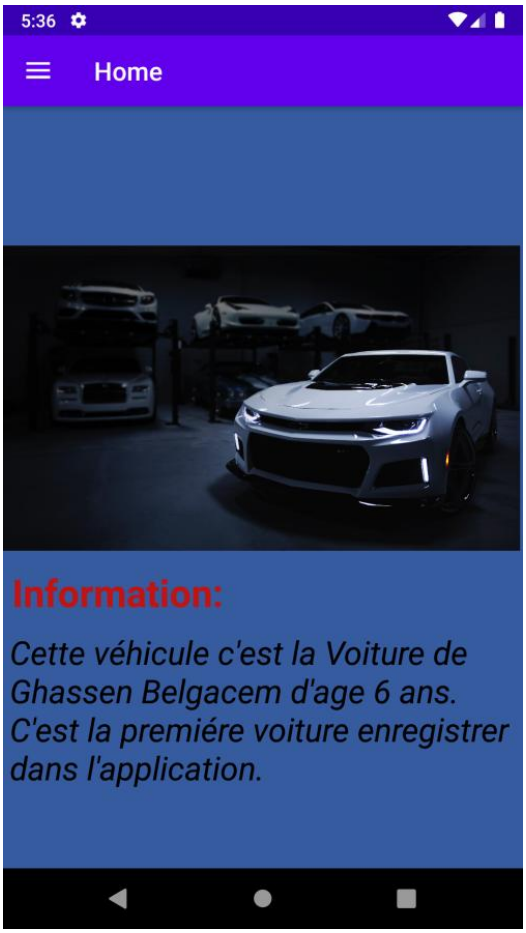


Figure 26- Premier élément du menu glissant « Home »

Aprés avoir eu les informations nécessaire de son véhicule ,l'utlisateur peut explorer les autres éléments de menu en glissant avec un doigt de gauche vers la droite ou en cliquant sur l'icône « Hamburger » à coté de « Home ».

La figure suivante présente les éléments de menu glissant :

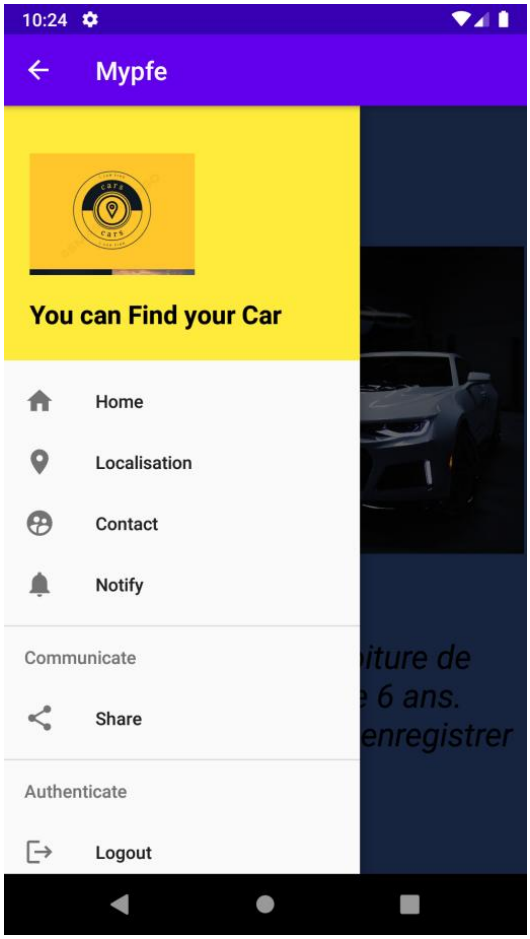


Figure 27- Les éléments du menu glissant.

Ensuite, l'utilisateur à la possibilité de localiser le centre sur la carte ainsi que de suivre la trajectoire de sa véhicule en ligne.

Les figures suivantes présentent l'interface intégrant la carte Google Maps:

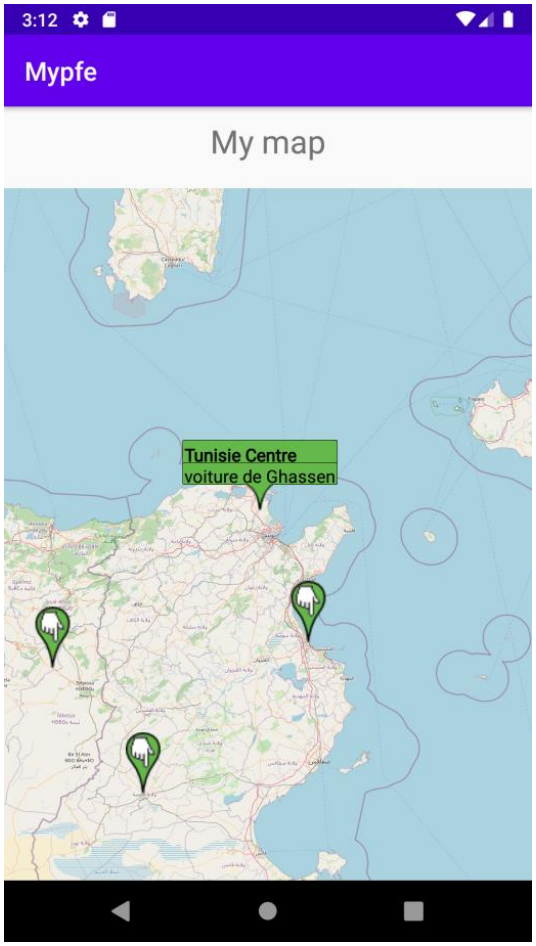


Figure 28- Interface intégrant la carte Google Maps.

L'utlisateur peut consulter l'interface de notification en cliquant sur l'élément «Notify». Dans cette page il peut envoyer une notification en cas d'une panne qui a lieu lors de la conduite. Cet élément contient deux Edits de textes et un Button avec le quel on peut envoyer le notification et concernant le premier Edit de texte c'est pour le titre de la Notification et la deuxieme pour son contenue.

La figure suivante présente cet interface :

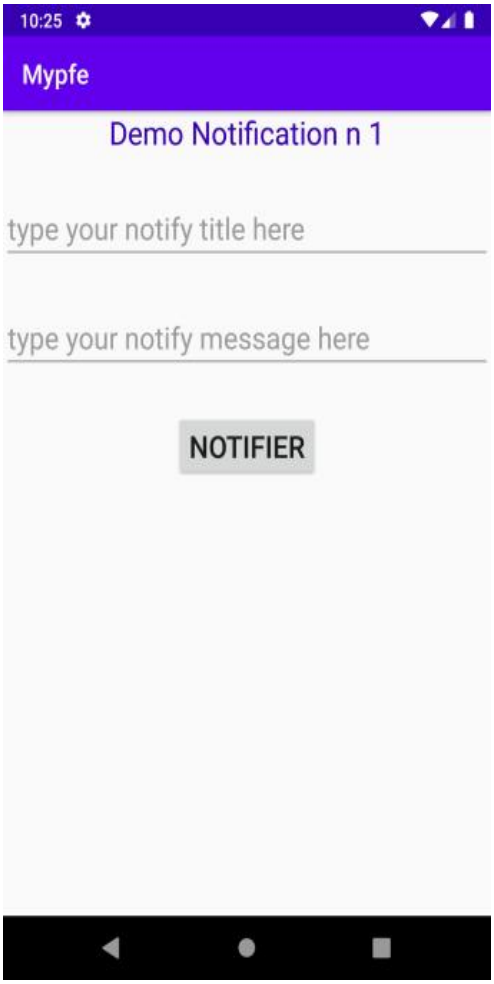


Figure 29- Interface affichant la page d'écrire une Notification.

4.5 Conclusion

Au cours de ce chapitre, on a mis en évidence l'environnement logiciel et matériel de ce projet qui représente le préparateur de la phase de réalisation et on a illustré les choix techniques pour cette opération ainsi que de récapituler les étapes de gestion de l'application. Ce chapitre nous a permis de passer au déploiement de notre application .

Conclusion générale

Notre projet intitulé « **Conception et réalisation d'une application mobile qui donne l'accées a l'utilisateur de suivre sa véhicule et de le localiser ».**

En ce qui concerne la démarche, nous avons en premier lieu effectué une phase d'étude des différents outils existants. En deuxième lieu nous avons spécifié notre application pour discerner les fonctionnalités .En troisième lieu, nous avons procédé à sa conception ainsi qu'aux choix technologiques pour sa réalisation. Enfin, nous l'avons mis en œuvre.

Ce projet nous a été bénéfique sur plusieurs plans. Sur le plan technique, nous avons acquis de nouvelles compétences en développement mobile et nous avons approfondi nos connaissances des technologies de communication actuelles. Nous avons également pu mettre en pratique nos connaissances en matière de conception logicielle et d'architecture. De plus, nous avons renforcé notre capacité à travailler en équipe, à gérer les tâches et à respecter les délais.

En conclusion, la réalisation de ce projet nous a permis d'acquérir une expérience précieuse dans le domaine du développement d'applications mobiles. Nous sommes fiers d'avoir créé une application qui répond aux besoins des utilisateurs en leurs offrant un moyen pratique et efficace de suivre et de localiser leurs véhicules. Ce projet nous a permis de développer nos compétences techniques et de nous familiariser avec les dernières technologies, tout en nous offrant une expérience enrichissante sur le plan professionnel.

Il est important à noter que la réalisation de ce projet nous a été bénéfique sur tous les plans. Sur le plan technique, ce projet nous a été une bonne occasion pour découvrir et maitriser le développement mobile, d'approfondir nos connaissances sur le plan des nouvelles technologies de communications.

Webographies

[1] https://www.google.com/search?q=Architecture

[2] https://fr.slideshare.net/HeithemAbbes1/architectures-3tiers-web

[3] https://osmdroid.github.io/osmdroid/How-to-use-the-osmdroid-library.html

[4] HTTP://developer.android.com/studio/intro?hl=fr

[5] HTTP://developer.android.com/guide/topics/permissions/overview?hl=fr

[6] HTTP://developer.android.com/studio

[7] https://softmany.com/fr/telecharger-adobe-photoshop-cs6/

[8] https://developer.android.com/studio/projects/android-library?hl=fr

[9] https://developer.android.com/tools

[10] https://firebase.google.com/docs?hl=fr

Résumé

Ce projet de fin d'études vise à créer une application mobile de suivi des véhicules. L'application fournira une localisation en temps réel, des informations sur la vitesse et la direction des véhicules, ainsi que des fonctionnalités avancées telles que la gestion des itinéraires et la génération de rapports. L'objectif est d'améliorer l'efficacité opérationnelle des entreprises de transport et de logistique en fournissant un outil de suivre pratique et complet.

Mot clé

DCU : diagramme des cas d'utilisation , DCL **:** diagramme de classes , **DOB :** diagramme d'objets , **DET :** diagramme d'états-transitions , **DAC :** diagramme d'activités , **DSE :** diagramme de séquence , **DCO :** diagramme de Collaboration , **DCP :** diagramme de composants , **DDP :** diagramme de déploiement Dans notre étude de projet , **Start uml :** Unified Modeling Languge , **Osmdroid** : Open Street Map **, MVC :** Modéle-Vue-Controleur , **PQP :** Plan qualité de Projet.

Abstract

This graduation project aims to create a mobile vehicle tracking application. The app provides real-time location, vehicle speed and direction information, as well as advanced features such as route management and reporting. The goal is to improve the operational efficiency of transport and logistics companies by providing a convenient and comprehensive tracking tool.