



Rapport de Stage de Fin d'année

Sujet

Application Android de covoiturage

Réalisé par :

BANI Ayoub

Encadré par

M. AMESROY Kamal

Année Universitaire 2017-2018

Dédicace

À mes chers parents Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

À tous ceux et toutes celles qui m'ont accompagné et soutenu durant cette période de stage

Remerciements

Mes vifs remerciements vont au premier lieu à l'encadrant Mr AMESROY Kamal pour l'aide qu'il m'a apportée tout au long de la période de réalisation de ce projet, pour ses remarques pertinentes, ses renseignements précieux, ses conseils, ses encouragements et surtout pour sa disponibilité. Il été le meilleur support pour me mener à bien un tel projet afin qu'il soit fructueux et profitable.

J'exprime mes sincères gratitudes aux membres du jury, qui ont accepté d'évaluer ce projet de fin d'année.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce projet.

Résumé

Le projet de stage consiste à concevoir et à réaliser une application Android de covoiturage qui présente un ensemble des services pour les utilisateurs. L'utilisateur soit un conducteur ou un passager. L'application permet de gérer les trajets postés par chaque utilisateur et de faciliter la communication entre eux.

En général, le travail est réalisé en deux parties. La première a été consacrée à l'étude analytique et conceptuelle de l'application, tandis que la deuxième traite la phase de développement et à la mise en œuvre de l'application.

Mots clés : Application, Android, Covoiturage, Conducteur, Passager

Abstract

The internship project is to design and build an Android application for carpool. The application presents a set of services for users. The user is a driver or a passenger. The application makes it possible to manage the journeys posted by each user and to facilitate the communication between them.

In general, the work is done in two parts. The first was devoted to the analytical and conceptual study of the application, while the second deals with the development phase and the implementation of the application.

Keywords: Application, Android, Carpool, Driver, Passenger

Liste des abréviations

authUI	authentication User Inteface
EET	Environmental Energy Technologies
Firebase Auth	Firebase Authentication
GPS	Global Positioning System
JSON	JavaScript Object Notation
LIFO	Last In First Out
MVC	Model View Controller
NoSQL	No structured query language
OS	Operating System
SDK	Software development kit
SMS	Short Message Service

Table des figures

Figure 1 diagramme de cas d'utilisation	19
Figure 2 Diagramme de Séquence "Rechercher par proximité"	19
Figure 3 Diagramme de Séquence "Effectuer une réservation"	20
Figure 4 Diagramme de Séquence "Ajouter trajet pour conducteur"	20
Figure 5 Diagramme de Séquence "Noter conducteur"	21
Figure 6 Diagramme de classe.....	22
Figure 7 Activité Android.....	23
Figure 8 Gestion LIFO des activités.....	24
Figure 9 étapes de lancement d'application.	25
Figure 10 Les fonctionnalités de Firebase	26
Figure 11 Le nœud 'passanger'	28
Figure 12 Le nœud 'driver'	29
Figure 13 Le nœud 'cars'	29
Figure 14 Le nœud 'car_images'	30
Figure 15 Le nœud 'driver_normal_trips'	30
Figure 16 Le nœud 'driver_schudeled_trips'	31
Figure 17 Le nœud 'normal_trips'	32
Figure 18 Le nœud 'scheduled_trips'	33
Figure 19 nœud 'passanger_post_requests'	34
Figure 20 nœud 'driver_post_requests'	34
Figure 21 page de chargement.....	35
Figure 22 interface d'authentification	36
Figure 23 processus d'authentification	36
Figure 24 Interface pour effectuer le choix.....	37
Figure 25 les utilisateurs inscrit dans l'application	37
Figure 26 menu 'Navigation drawer'	38
Figure 27 menu 'Bottom Navigation'	38
Figure 28 menu de déconnexion.....	39
Figure 29 Accueil du conducteur.....	40
Figure 30 profile d'un passager	41
Figure 31 Map d'un trajet.....	42
Figure 32 Post d'un passager.....	43
Figure 33 saisie du prix	44
Figure 34 post du passager 'Nouvelle demande'	45
Figure 35 Interface ajouter nouveau post.....	46
Figure 36 ajouter un nouveau post cas d'erreur	47
Figure 37 Ajouter un nouveau trajet programmé	48
Figure 38 saisie de l'heure.....	49
Figure 39 précision de la location par autocomplété.....	50
Figure 40 précision de la location par un clic sur le map	51
Figure 41 trajet entre origine et destination.....	52
Figure 42 données de l'api sous format json	53
Figure 43 interface Mes trajets	54
Figure 44 traitement demande des passagers	55
Figure 45 interface profile conducteur	56
Figure 46 ajouter une nouvelle photo de profile	57

Figure 47 ajout de la photo de profile	58
Figure 48 fragment ajouter une nouvelle voiture	59
Figure 49 formulaire des informations de la voiture	60
Figure 50 ajout d'une nouvelle image de la voiture.....	61
Figure 51 affichage claire de l'image.....	62
Figure 52 interface Mes trajet du passager.....	63
Figure 53 choix d'un conducteur	64
Figure 54 Profile passager ajout de nouveau informations	65
Figure 55 saisie des données du passager	66
Figure 56 profile passager modifier	67

Table de matières

Liste des abréviations	7
Table des figures.....	8
Table de matières.....	10
Introduction générale.....	10
Chapitre 1 : Présentation de l'entreprise	12
Chapitre 2 : Présentation du projet.....	13
1. Problématique.....	13
2. Objectif	14
3.Contexte	14
Chapitre 3 : Analyse et Conception	15
1. Analyse des besoins.....	15
2. Conception	18
Chapitre 4 : Réalisation et Mise en œuvre	22
1. Outils de développement	22
2. La Mise en œuvre	28
2.1 Base de données :.....	28
2.2 Implémentation.....	35
Conclusion Générale.....	68
Webographie	69

Introduction générale

Bien que considéré comme une nouvelle pratique, le covoiturage existe depuis déjà un bon nombre d'années. Aujourd'hui, il est encouragé par les gouvernements, en particulier dans un souci environnemental. En EET, les changements climatiques poussent les dirigeants à mettre en place des politiques plus respectueuses de l'environnement, notamment par la promotion

de déplacements plus économes en émission de CO₂, dont le covoiturage. Les usagers n'ont pas attendu l'investissement du gouvernement pour se mettre à ce mode de déplacement. Les économies réalisées grâce au covoiturage ont déjà fait des adeptes. En encourageant les entreprises à avoir un rôle facilitateur dans la mise en place d'un covoiturage en sécurité pour les déplacements domicile-travail. Le terme « covoiturage » est employé dans différents codes notamment le code de Sécurité sociale, le code des transports, le code général des collectivités territoriales ou encore le code de l'environnement. Il est également repris dans différentes lois dont la loi Grenelle de l'environnement de 2007. Plus récemment, ce sont les plans de protection de l'atmosphère, qui ont pour objectif d'améliorer la qualité de l'air, qui promeuvent l'utilisation du covoiturage dans les entreprises. Malgré ces nombreuses incitations, le covoiturage reste une pratique marginale. Jusqu'ici, il n'existe pas de réponse claire aux questions de responsabilités en cas d'accident notamment lorsque celui-ci s'inscrit dans le cadre professionnel.

Dans le cadre d'un plan de déplacements, le covoiturage est souvent une des premières mesures que l'entreprise veut promouvoir. Toutefois, sa promotion n'est pas simple. En effet, il faut savoir que le covoiturage fonctionne surtout dans les entreprises qui ne sont pas très accessibles en transports publics et dans lesquelles les travailleurs habitent loin et/ou au sein desquelles l'organisation du travail fonctionne en horaire fixe ou en travail d'équipe (pour les entreprises de production, par exemple).

A part ces quelques « précautions », le covoiturage présente évidemment de nombreux avantages.

Avantages pour le travailleur covoitureur :

Un gain social : Création d'un réseau de liens entre les membres d'une même équipe de covoitureurs

Un gain financier via le partage des frais de transport entre les membres de l'équipe de covoitureurs tout en bénéficiant d'un avantage fiscal.

Avantages pour l'entreprise :

Un gain financier : en réalisant des économies dans la mise à disposition de places de stationnement à votre personnel. La mise en place de la mesure est également peu coûteuse.

Un gain d'image : Covoiturer donne une image dynamique et soucieuse de l'environnement de votre entreprise.

Un gain social : le lien social entre vos travailleurs est renforcé et entraîne une dynamique de cohésion entre vos travailleurs.

Avantages pour la collectivité :

Un gain pour l'environnement : La diminution de la pollution en ville sera effective !

Un gain pour la mobilité : Moins de voitures sur le réseau automobile, c'est permettre la décongestion du trafic.

Chapitre 1 : Présentation de l'entreprise

Warssalesse INVESTest précurseur dans le développement et l'intégration de 4 technologies :

- l'intégration cartographique,
- la mise en place de terminaux à bord de véhicules,
- la géolocalisation par GPS
- l'envoi et la réception de données métier en temps réel

Intégré dans différents projets, la société s'impose aujourd'hui dans le domaine des C-ITS en. En se concentrant sur son domaine d'excellence, le développement d'application pour les

Systèmes de Transport Intelligent (ITS), Warssalesse Invest établit des partenariats avec des professionnels experts, reconnus dans leurs domaines respectifs dans la recherche CO-GISTICS. Ils réinventent les processus métiers et transforme votre expérience en relevant les données provenant de vos équipements connectés, pour les intégrer dans des applications avancées qui viendront enrichir votre système d'information. Ils vous permettent ainsi de réduire considérablement le délai entre la prise de connaissance des données et vos prises de décisions.

Les technologies de communication temps réel et courte distance sont les solutions d'avenir dans la gestion de la sécurité et de l'économie d'énergie, autant en urbain qu'en interurbain.

Warssalesse Invest propose des applications de sécurité, avec la mise à disposition pour les conducteurs des événements routiers (accident, véhicule arrêté, véhicule à contre sens, bouchons...), ainsi qu'un système de gestion de la priorité des véhicules d'urgence aux feux tricolores. Les économies d'énergie ne sont pas en reste grâce à l'application GLOSA, qui permet de connaître la vitesse optimale pour passer tous les feux au vert, et délivre une information permettant de décider s'il est intéressant ou non de couper le moteur à un feu rouge.

Chapitre 2 : Présentation du projet

1. Problématique

- Problèmes des conducteurs

*le coût de l'essence

*le coût d'autoroute

- Problèmes de passagers

*le coût des moyens de transport

*le confort de ces moyens

- Problèmes des deux

*Manque d'amitié et de convivialité.

*pollution et Traffic

2. Objectif

Le projet consiste à concevoir et à développer une application mobile qui gère le covoiturage pour les habitants d'une ville spécifique ou au niveau du royaume et international. Utilisation en commun d'une voiture unique afin de réaliser un trajet.

Les objectifs de l'application sont multiples et évidents :

Pour les utilisateurs :

Diminution des frais liés à l'utilisation de la voiture.

Réduction du temps passé dans les transports.

Meilleure accessibilité vers leur point d'arrivée.

Créer des amitiés.

Pour les entreprises :

Réduction des places de parking

Diminution des émissions de CO2.

Développement d'un esprit de solidarité.

Pour tout le monde :

Moins de pollution, meilleure qualité de l'air.

Diminution des encombrements de circulations.

3.Contexte

Ecologique, économique et convivial, le covoiturage fait beaucoup parler de lui pour lutter contre les engorgements des villes, la pollution, ou bien encore pour pallier aux grèves de transports. D'où le besoin de mettre en place une application dédiée entièrement au covoiturage. Cette application propose de mettre en contact conducteurs et passagers pour

partager une voiture le temps d'un voyage déterminé, le covoiturage peut s'effectuer entre le domicile et le lieu de travail ou étude ou d'une ville à l'autre.

Chapitre 3 : Analyse et Conception

1. Analyse des besoins

Environnement :

Cette application de covoiturage est utilisée pour tous les citoyens d'une ville spécifique âgés plus que 18 ans disposant d'une carte d'identité.

Description des fonctions :

- F1 : Inscription.

Cette fonction doit permettre à un visiteur de devenir membre de l'application pour qu'il puisse utiliser le service du covoiturage. Le visiteur doit spécifier son type soit un passager ou conducteur.

- F2 : Authentification.

Pour accéder au reste des fonctionnalités, il est nécessaire de s'enregistrer et de se logger par la suite grâce à un Framework qui facilite l'authentification. Pour sortir du site il faut se déconnecter.

- F3 : Gestion des trajets.

-Ajouter trajet : l'utilisateur de l'application peut publier un trajet en saisissant les informations et les préférences d'un trajet. Pour un conducteur il faut qu'il ait une voiture.

-Consulter trajet : L'utilisateur peut rechercher et consulter les trajets de cette semaine sans qu'il voie tous les détails d'un trajet.

-Modifier/Supprimer trajet : L'utilisateur peut modifier ou supprimer ses trajets qu'il a créé.

- F5 : Gérer réservation.

-Ajout réservation : L'utilisateur effectue une recherche du trajet (ville de départ, celle d'arrivée, ou bien par proximité), il choisit le trajet qui lui convient le mieux et réserve sa place en cliquant sur un bouton "Réserver". Une notification est envoyée au conducteur pour qu'il accepte ou refuse cette demande.

-Modifier réservation : L'utilisateur peut consulter les réservations de ses trajets.

-Supprimer réservation : L'utilisateur peut supprimer la demande de réservation.

- F6 : Gestion des voitures.

L'utilisateur de type conducteur fait l'ajout de ses voitures, la modification ainsi la suppression. Il faut ajouter les informations de la voiture et ses images.

Hierarchisation des fonctions :

Rappel de la signification des niveaux hiérarchiques :

1. Fonction fondamentale
 - a-Inscription
 - b-Authentification
2. Fonction importante
 - a- Gestion des trajets
 - b-Gérer les réservation
3. Fonction utile
 - a-Gestion des voitures.

Description des contraintes :

- Contraintes organisationnelles :

-L'utilisateur a le droit de consulter les futurs trajets.

- Contraintes techniques :

-Android, Firebase (Firebase Auth, Realtime database, Cloud Storage...).

-Java, xml, javascript...

- Contraintes humaines :

L'application mobile devra être :

-Facile à gérer et à manipuler par tous les types des utilisateurs.

-Fiable a l'affichage.

-Conviviale : Les interfaces doivent être conviviales, ergonomes et faciles à utiliser avec un minimum de saisie des données et un affichage automatique des informations.

-Assure la communicabilité : fournit des interfaces qui facilite l'interaction entre les utilisateurs de l'application.

- Contraintes règlementaires :

- Le problème de la confusion entre transport commun clandestin et le covoiturage.

Critère d'appréciations générales :

Maintenir la sécurité :

- Accès à la gestion des profils, trajet et réservation sécurisé par l'authentification.
- utilisation de la notion de cache pour l'efficacité de l'application. Utilisation de smartlock pour simplifier l'authentification.

Utilisation d'une architecture MVC.

- Garantir qu'il n'y'a pas des attaques sur les droits d'accès d'utilisateur du site : hasher et crypter les droits d'accès.

Acteur et leur rôle :

- Visiteur : -Inscription et authentification

- Passager : -Gérer son profil

 - Gérer ses réservations

 - Gérer ses trajets

- Conducteur : -Gérer son profil

 - Gérer ses voitures

 - Gérer ses trajets

2. Conception

Diagramme de cas d'utilisation

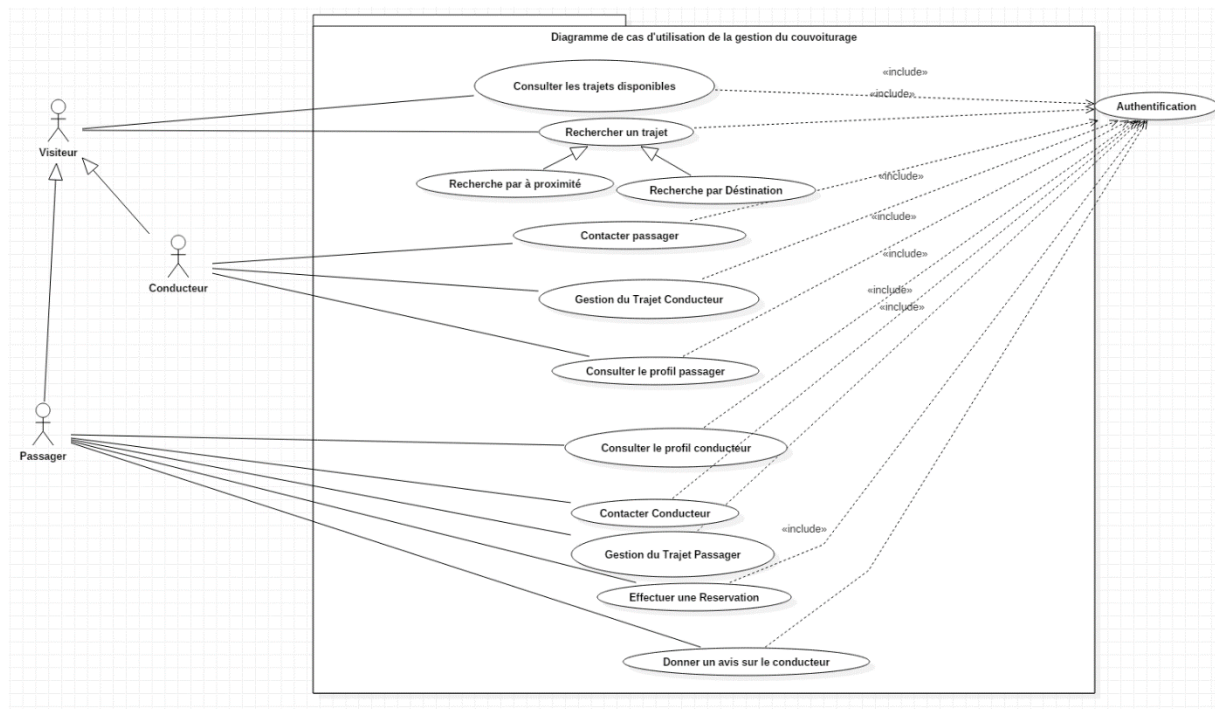


Figure 1 diagramme de cas d'utilisation

Diagramme de Séquence “Rechercher par proximité”

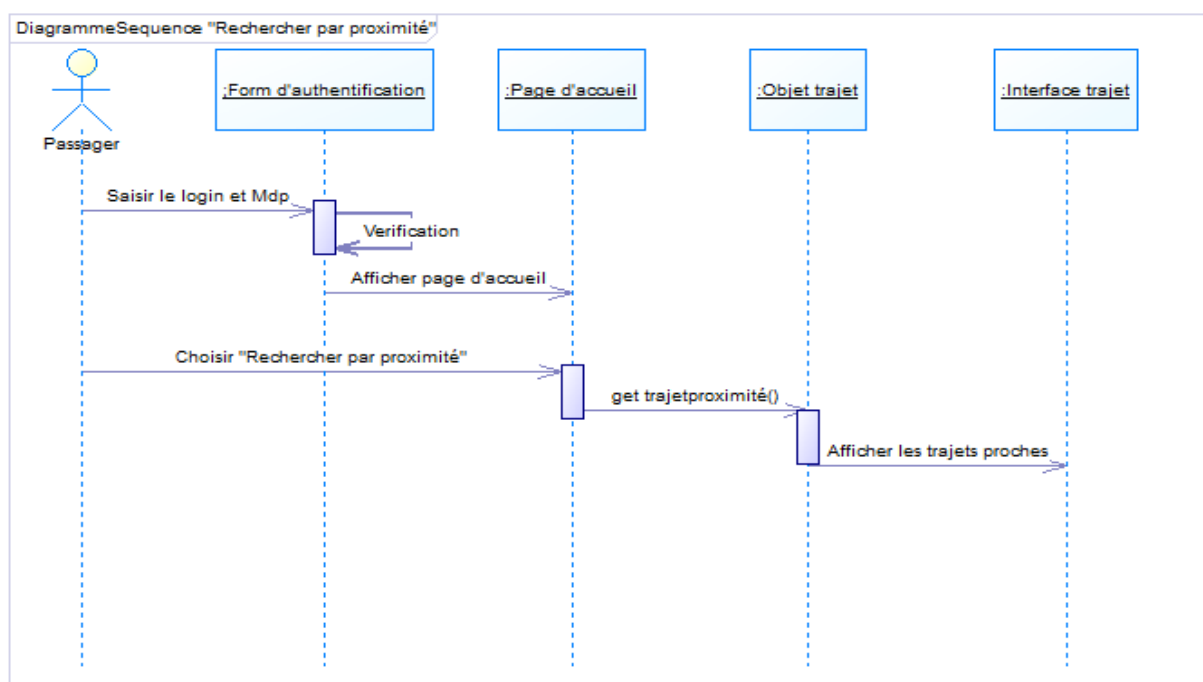


Figure 2 Diagramme de Séquence “Rechercher par proximité”

Diagramme de Séquence “Effectuer une réservation”

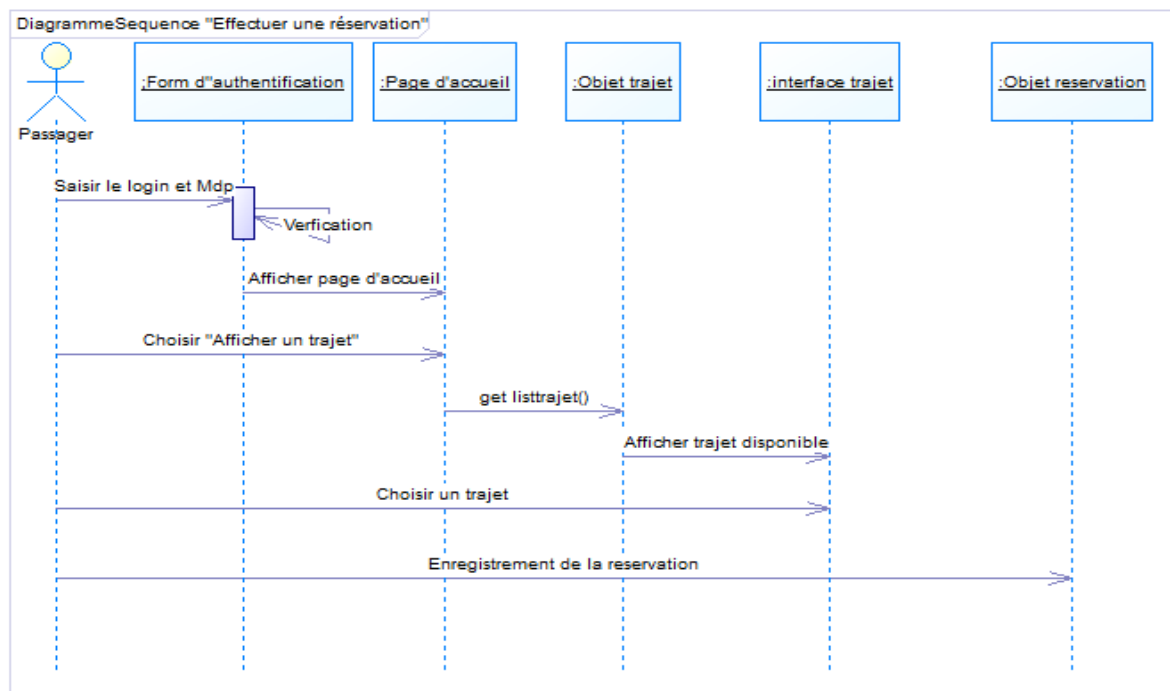


Figure 3 Diagramme de Séquence "Effectuer une réservation"

Diagramme de Séquence "Ajouter trajet pour conducteur"

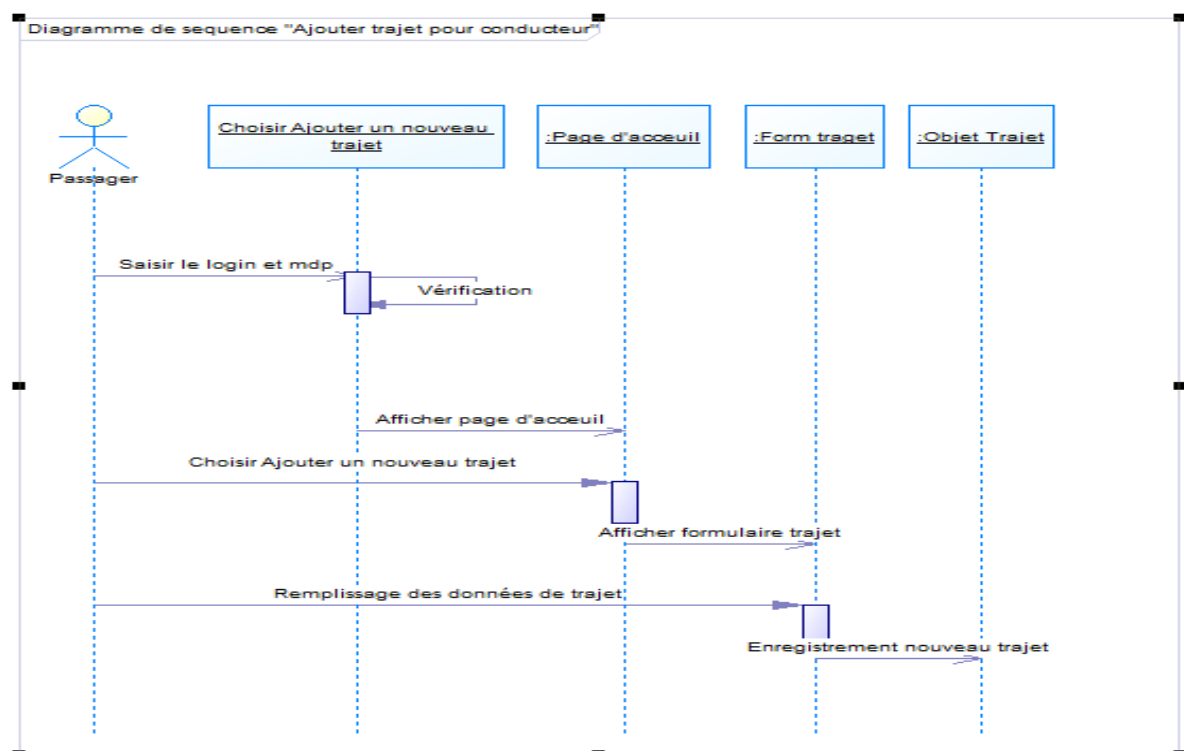


Figure 4 Diagramme de Séquence "Ajouter trajet pour conducteur"

Diagramme de Séquence “Noter conducteur”

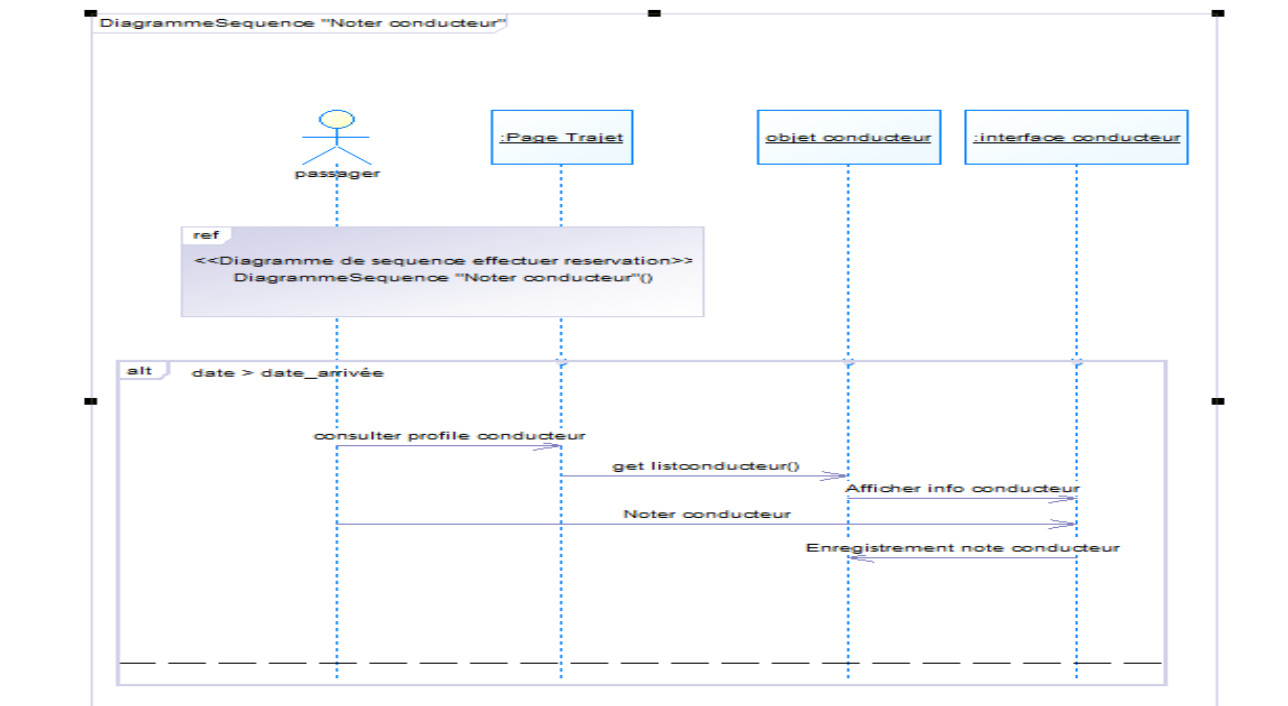


Figure 5 Diagramme de Séquence “Noter conducteur”

Diagramme de classe

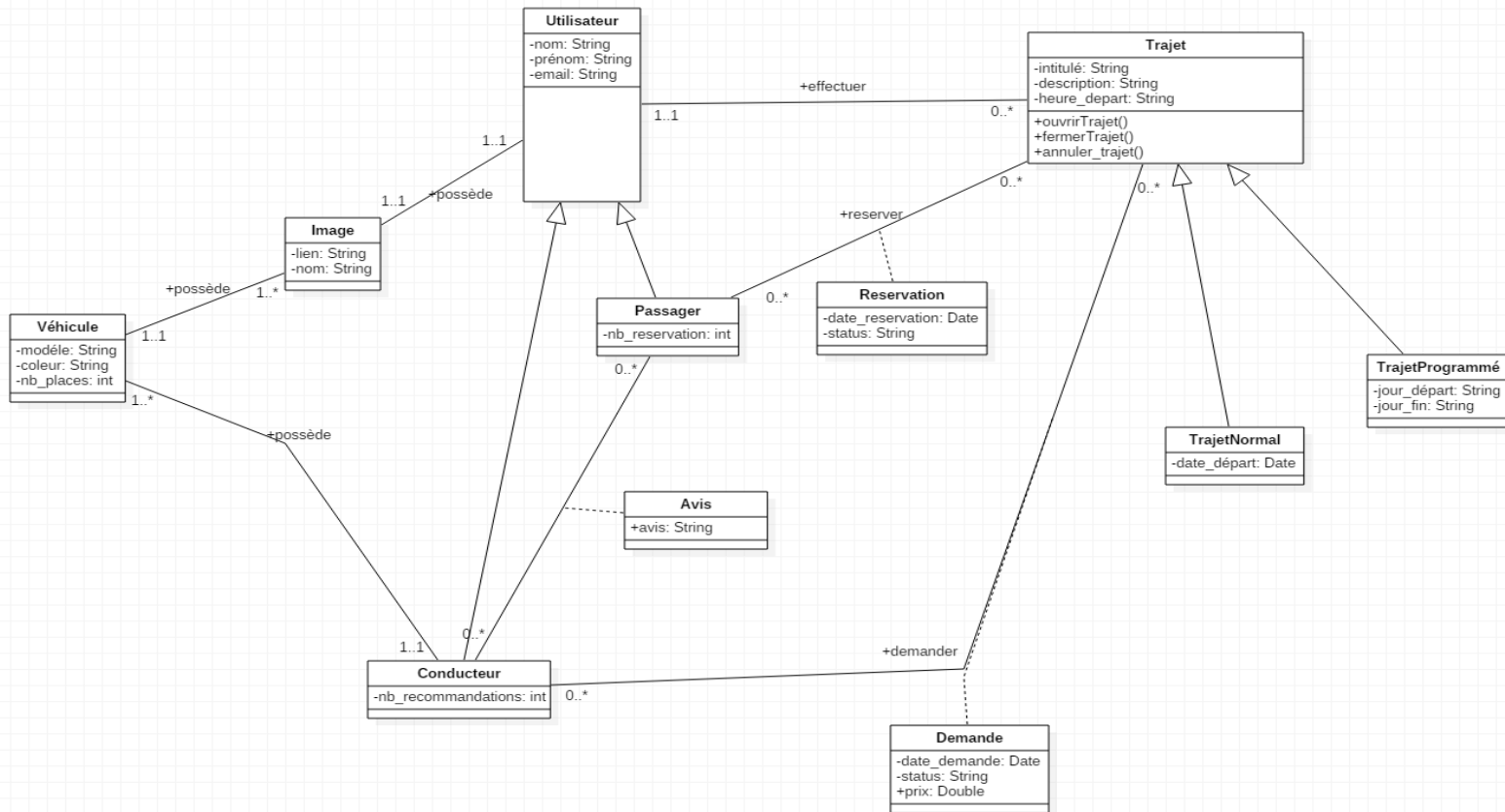


Figure 6 Diagramme de classe

Chapitre 4 : Réalisation et Mise en œuvre

1. Outils de développement

Android

Android est un système d'exploitation (OS) de Google destiné à une multitude de smartphones, tablettes et autres objets connectés. Aujourd'hui dans sa version 8 dite "Oreo", Android est l'OS mobile le plus utilisé au monde avec deux milliards d'utilisateurs mensuels actifs en mai 2017, dominant les parts de marché devant l'iOS d'Apple sur iPhone ou iPad. Créé par Andy Rubin autour d'un noyau Linux, il permet l'installation d'une infinité d'applications de messagerie et téléphonie -de Facebook à Twitter en passant par Snapchat ou Instagram- ou encore d'actualité -comme L'Express- et des jeux vidéo mobiles. Sa conception open-source et sa gratuité l'ont rendu populaire chez de nombreux fabricants de smartphones,

qui le personnalisent la plupart du temps avec leurs propres interfaces ou fonctionnalités (propriétaires). La protection de ce système et ses données personnelles, comme tous les autres OS mobiles, est devenu un enjeu crucial de cybersécurité.

Le système d'exploitation d'Android se basait sur Linux. Si on veut être plus précis, c'est le noyau (« kernel » en anglais) de Linux qui est utilisé. Le noyau est l'élément du système d'exploitation qui permet de faire le pont entre le matériel et le logiciel.

Composants principaux d'une application

Une application Android est construite à partir de quatre catégories de composants : les activités, les services, les intentions et les fournisseurs de contenu.

Une activité

L'activité est une composante principale d'une application qui représente à la fois l'implémentation et les interactions des interfaces. Elle est entièrement développée en Java. Chaque activité possède sa propre interface et remplit Tout l'écran, il n'y a donc qu'une seule activité d'affichée à la fois. L'activité gère les interfaces et les éléments visuels qui fournissent des informations ou avec lesquels les utilisateurs peuvent interagir. Elle sert aussi de support pour ces interfaces. Cependant, ce n'est pas le rôle de l'activité que de créer et de disposer les éléments graphiques, elle n'est que l'échafaudage sur lequel vont s'insérer les objets graphiques. De plus, une activité contient des informations sur l'état actuel de l'application : ces informations s'appellent le « contexte ».

Ce « contexte » constitue un lien avec le système Android ainsi que les autres activités de l'application, comme le montre la figure suivante :



Figure 7 Activité Android

Une activité possède aussi différents états qui indiquent au système comment réagir en fonction de certaines actions. Par exemple, lorsqu'on reçoit un appel, il est plus important qu'on puisse y répondre que de continuer d'utiliser une application. Pour cela, chaque application possède une priorité plus ou moins élevée : quand une application se lance, elle se met en haut de ce qu'on appelle la pile d'activités. Dans ce cas, la pile est construite avec une structure de données de type « LIFO » (Last In First Out) qui signifie que le dernier élément ajouter sera le seul visible et sera le premier à sortir. En effet, Lorsqu'on ajoute une activité, celle-ci viendra se placer au sommet. Quand cette activité sera retirée, c'est la seconde activité, anciennement première qui prendra la place du sommet comme illustrée Ci-dessous

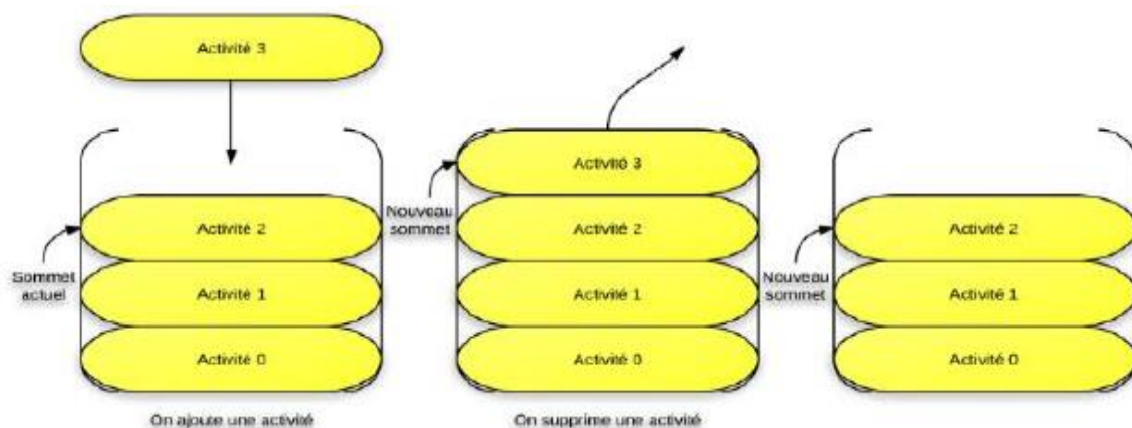


Figure 8 Gestion LIFO des activités.

Une activité qui perd sa première place ne reviendra qu'une fois que toutes les activités qui se trouvent au-dessus d'elle seront finies. On retrouve ainsi le principe expliqué précédemment. Une seule application est visible sur le terminal dans laquelle on n'aperçoit seulement l'interface graphique de l'activité en cours.

Voici une illustration du cycle de vie d'une activité :

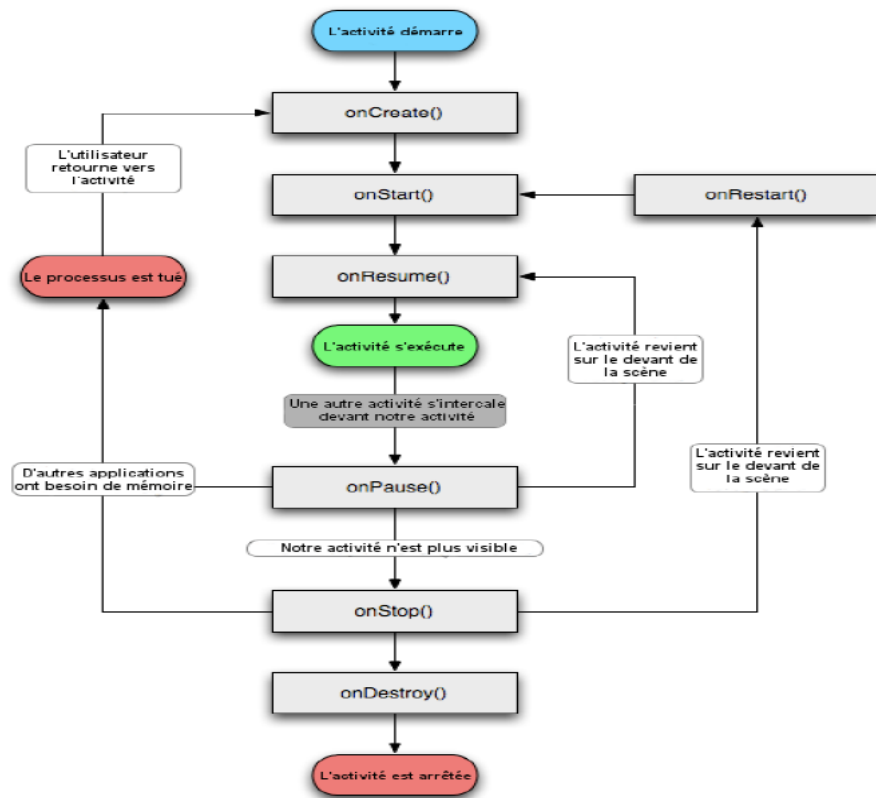


Figure 9 étapes de lancement d'application.

Un service

C'est l'analogue des services ou démons qui s'exécutent sur un OS classique. C'est un morceau de code qui s'exécute habituellement en arrière-plan entre le moment où il est lancé jusqu'à l'arrêt du mobile.

Une Intention

Une intention (intent en anglais) est un message contenant des données émis par Android (l'OS, une autre application, un autre service) pour prévenir les applications s'exécutant de la survenue d'un événement : déplacement du GPS, réception d'un SMS etc.

Un fournisseur de contenus

Une application Android répond normalement à un besoin et fournit donc certaines fonctionnalités. On appelle cela un fournisseur de contenus. Une application doit normalement se déclarer comme fournisseur de tel ou tel contenu (lorsqu'ils sont identifiés). Une autre application qui nécessiterait à un moment ce contenu émet une requête auprès de l'OS pour l'obtenir. L'OS lance alors directement l'application déclarée comme fournisseur.

Firebase

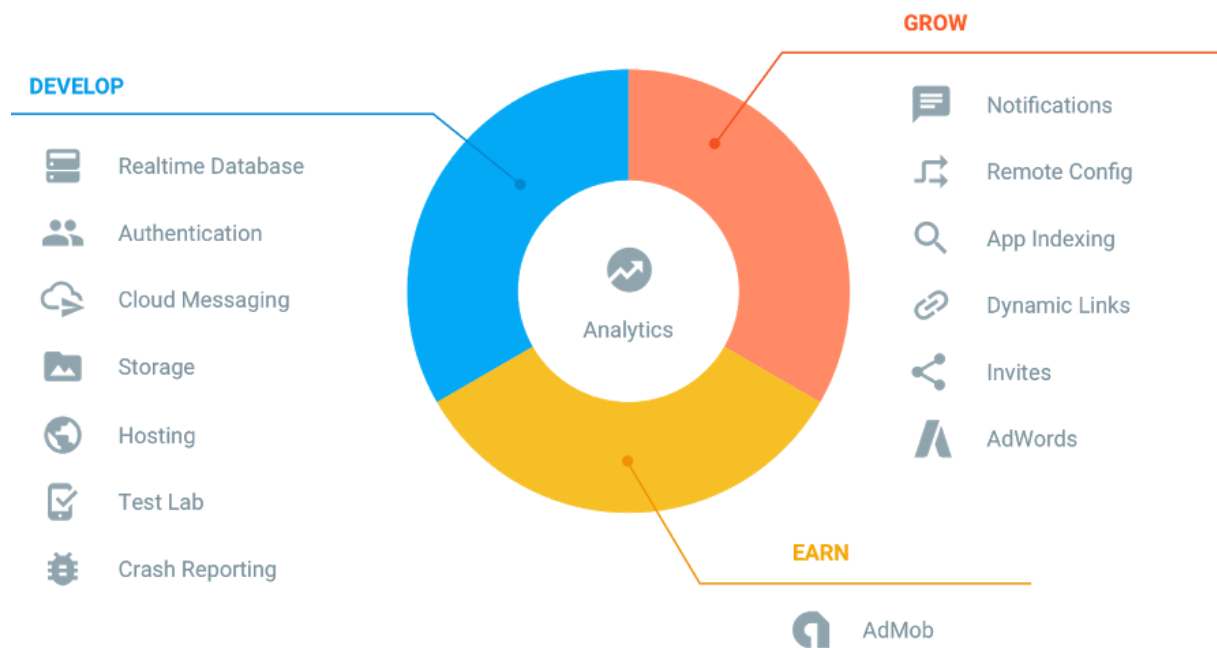


Figure 10 Les fonctionnalités de Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet.

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles.

Lors de ce projet j'utilisé les fonctionnalités suivantes :

- Realtime Database

Il permet de stockez et synchronisez les données avec une base de données cloud NoSQL. Les données sont synchronisées pour tous les clients en temps réel. On plus il y a l'utilisation de la notion de cache, les données restent disponibles lorsque l'application est hors connexion.

La base de données Firebase Realtime est une base de données hébergée dans le cloud. Les données sont stockées en format JSON et synchronisées en temps réel avec chaque client connecté.

- Authentification

La plupart des applications doivent connaître l'identité d'un utilisateur. La connaissance de l'identité d'un utilisateur permet à une application de sauvegarder en toute sécurité les données de l'utilisateur dans le cloud et de fournir la même expérience personnalisée sur tous les appareils de l'utilisateur.

Firebase Authentication SDK fournit des méthodes permettant aux utilisateurs de se connecter avec leurs comptes Google, Facebook, Twitter et GitHub.

- Cloud Storage

Cloud Storage est un service de stockage d'objets puissant, simple et économique conçu pour Google. Les SDK Firebase pour Cloud Storage ajoutent une sécurité Google aux téléchargements de fichiers, quelle que soit la qualité du réseau.

Cloud Storage peut stocker des images, du son, des vidéos ou d'autres contenus générés par les utilisateurs.

Toutes les données de base de données Firebase Realtime sont stockées en tant qu'objets JSON. On peut considérer la base de données comme une arborescence JSON hébergée dans le cloud. Contrairement à une base de données SQL, il n'y a pas de tables ou d'enregistrements. Lorsque vous ajoutez des données à l'arborescence JSON, il devient un nœud dans la structure JSON existante avec une clé associée. On peut fournir nos propres clés avec la méthode `push()`.

2. La Mise en œuvre

2.1 Base de données :

Le nœud passanger :

Ce nœud représente les infos sur le passager chaque passager est identifié par une clé. On plus on sauvegarde les informations suivantes : âge, email, points, photo de profile, sexe, le nom de l'utilisateur et si le passager est un fumeur ou non. Photo de profile est un lien qui point sur une ressource de cloud storage.



Figure 11 Le nœud 'passanger'

Le nœud driver :

Pour le conducteur on stocke presque les mêmes informations en plus un champ recommandation qui indique le nombre de recommandations de ce conducteur.

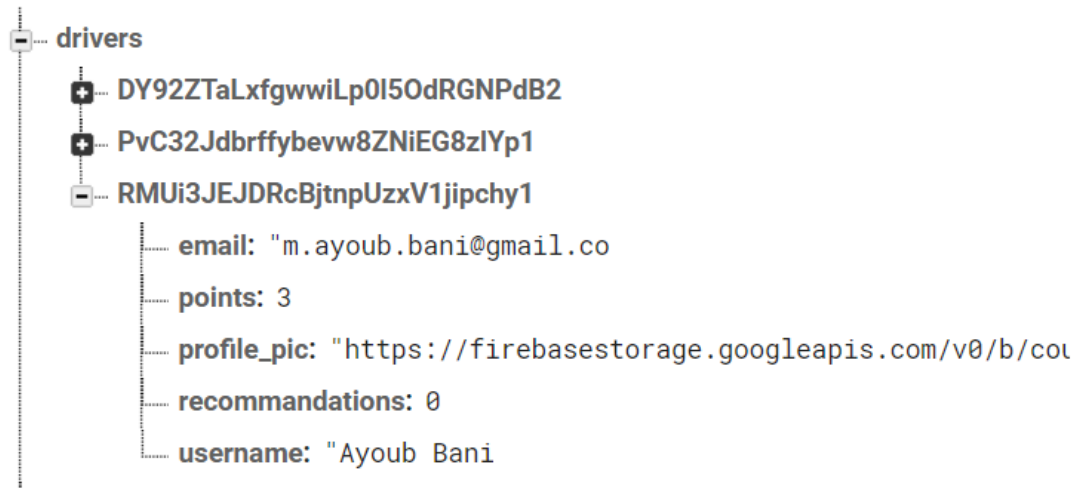


Figure 12 Le nœud 'driver'

Le nœud cars

Ce nœud contient les informations suivantes : couleur de la voiture, le model, le nombre d'image poster de la voiture et le nombre de places de la voiture. Ces informations son sauvegarder sous le nœud id qui identifier la voiture qui est lui-même sous identifiant de l'utilisateur.

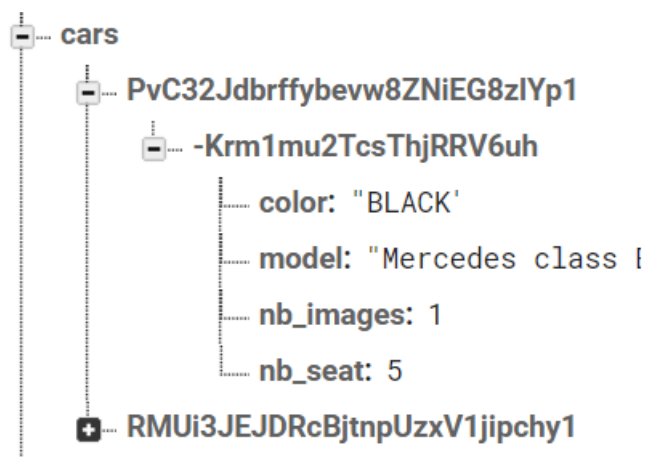


Figure 13 Le nœud 'cars'

Le nœud car_images

Pour ce nœud on sauvegarde l'url de l'image dans le cloud. Cette information est accessible par id de l'utilisateur puis l'id du nœud de la voiture.

couvoiturage-21763



Figure 14 Le nœud 'car_images'

Le nœud driver_normal_trips



Figure 15 Le nœud 'driver_normal_trips'

Ce nœud représente les informations d'un trajet d'un conducteur. Il y a deux types de trajets : trajet normal (Normal Trip) et trajet programmé (Schudeled trip).

D'abord on organise les trajets par payé et après la région. Ces données sont retenues de l'api de google map. Après on les organise par utilisateurs. Chaque utilisateur peut poster plusieurs trajets.

Chaque sous nœud contient les informations suivantes :

Date de départ, heure départ, description, destination, destination Latitude, destination longitude, nombre de places réserver, Origin (départ), origine latitude, origine longitude et un titre.

Origin et destination représente les adresses de départ et destination. On sait que chaque lieu sur la terre est identifié par des coordonnées géographique (latitude et longitude). Alors que l'origine latitude et origine longitude sont les coordonnées de l'origine.

Le nœud `driver_schudeled_trips`

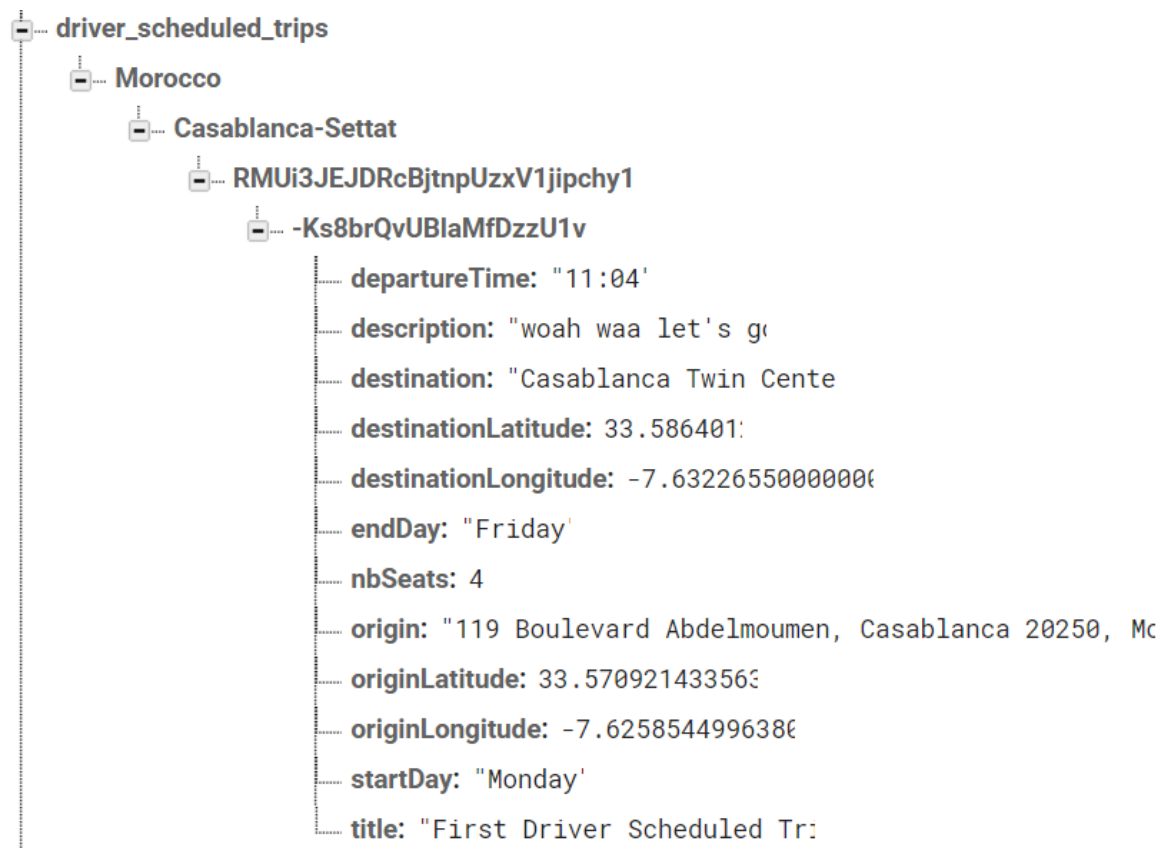


Figure 16 Le nœud 'driver_schudeled_trips'

Ce nœud contient presque les mêmes informations que le nœud précédent. La différence est qu'il n'y a pas date départ c'est par ce que ce nœud représente les informations d'un trajet

programmé, au lieu de la date on stocke les jours que ce trajet est effectué. C'est un intervalle déterminé par (startDay et endDay)

Le nœud `normal_trips`

Il est similaire au nœud `driver_normal_trips`

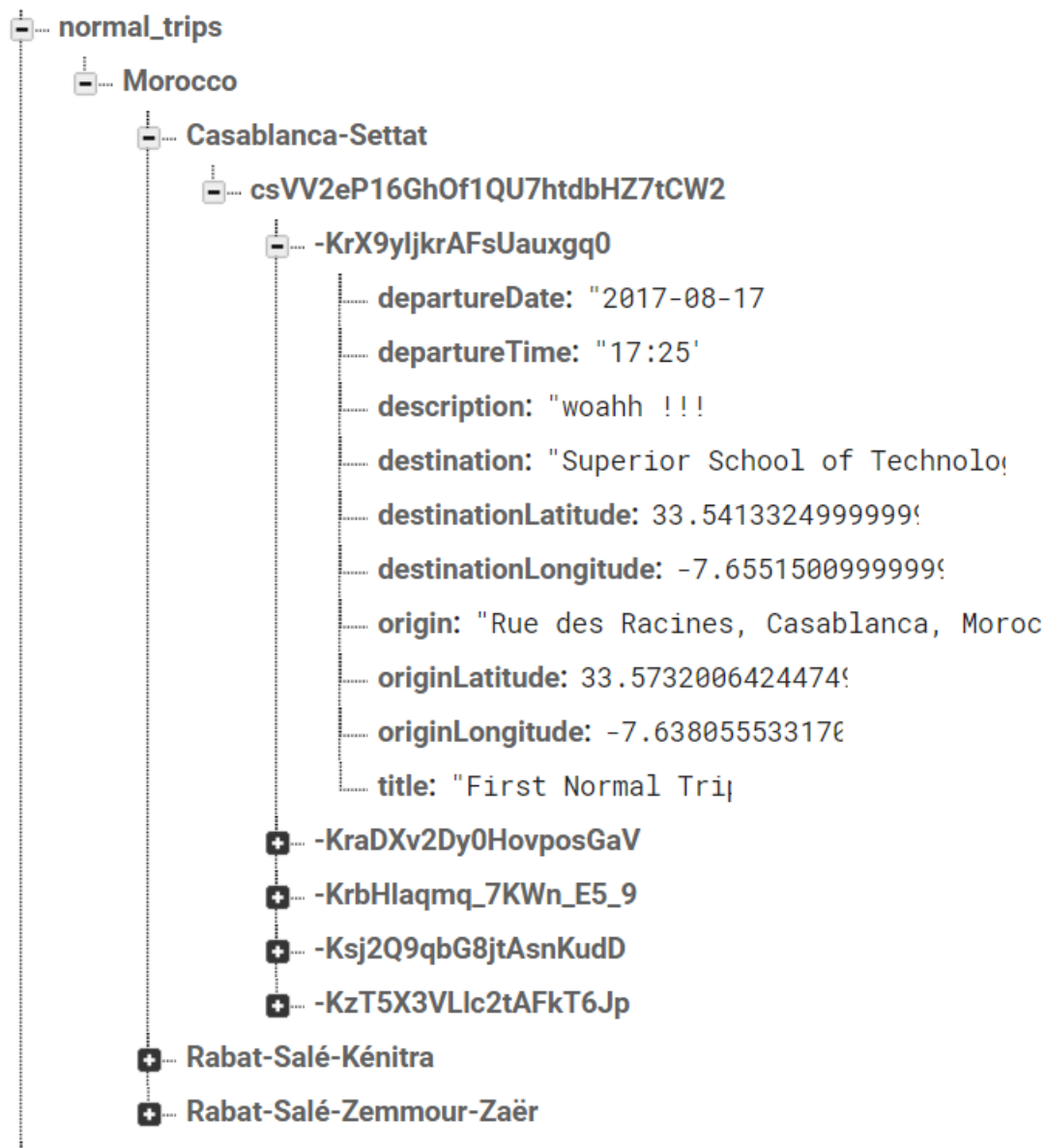


Figure 17 Le nœud 'normal_trips'

Le nœud `scheduled_trips`

Ce nœud est aussi similaire au nœud `driver_scheduled_trips`



Figure 19 nœud 'passanger_post_requests'

Nœud driver_post_requests

C'est presque les mêmes informations à part il on ne sauvegarde pas le prix. Le passager ne propose pas le prix. C'est à négocier dans la partie chat de l'application.

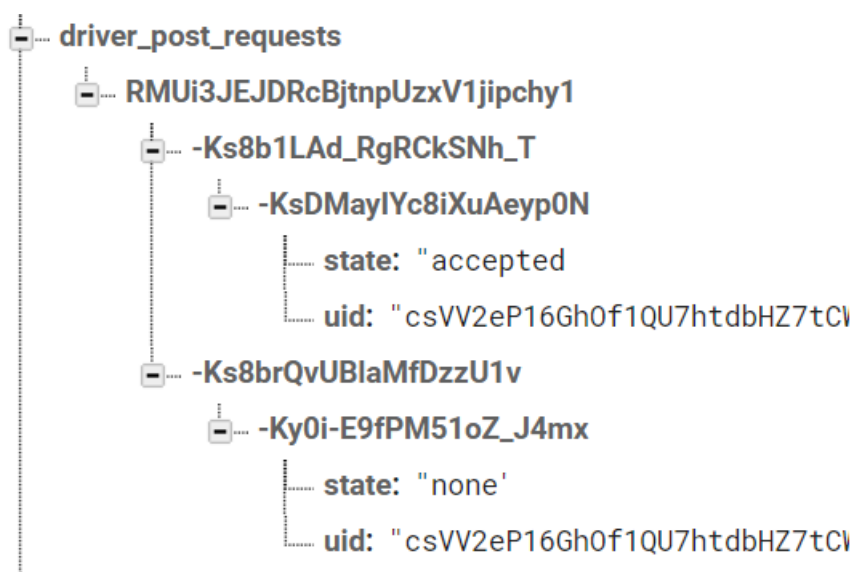


Figure 20 nœud 'driver_post_requests'

2.2 Implémentation

Page de chargement :

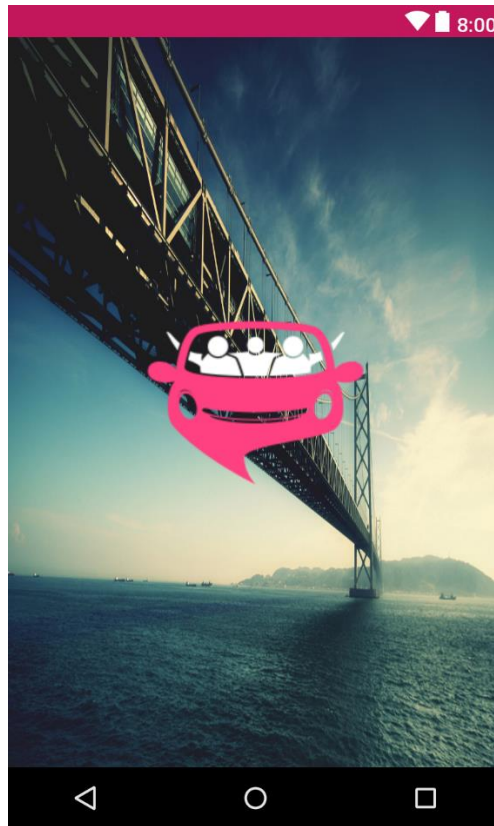


Figure 21 page de chargement

Processus de l'authentification :

Pour l'authentification j'ai utilisé le framework de firebase authUI qui permet l'authentification en utilisant plusieurs méthodes : authentification par email, google, facebook, et numéro téléphone (n'est pas en cours activé dans l'application). L'authentification fournit l'utilisation de smartlock pour l'authentification automatique.

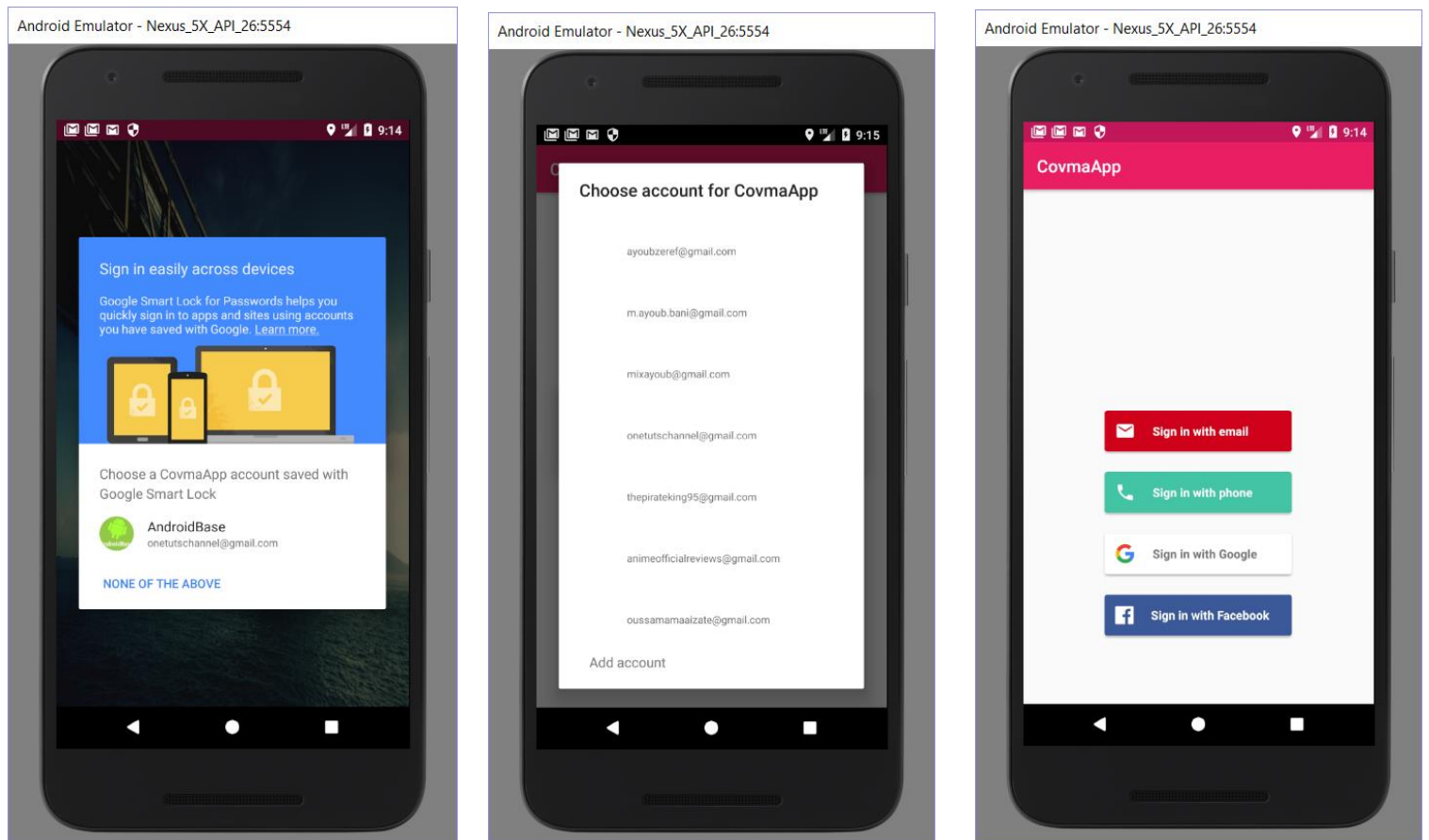


Figure 22 interface d'authentification

Par exemple en choisissant l'authentification par un compte Gmail. L'application fournit la liste des comptes récemment utilisés pour l'authentification. Pour ajouter un nouveau compte on clique sur 'Add account' puis on fournit l'email et le mot de passe du compte Gmail et après il faut respecter les conditions d'utilisation de Google.

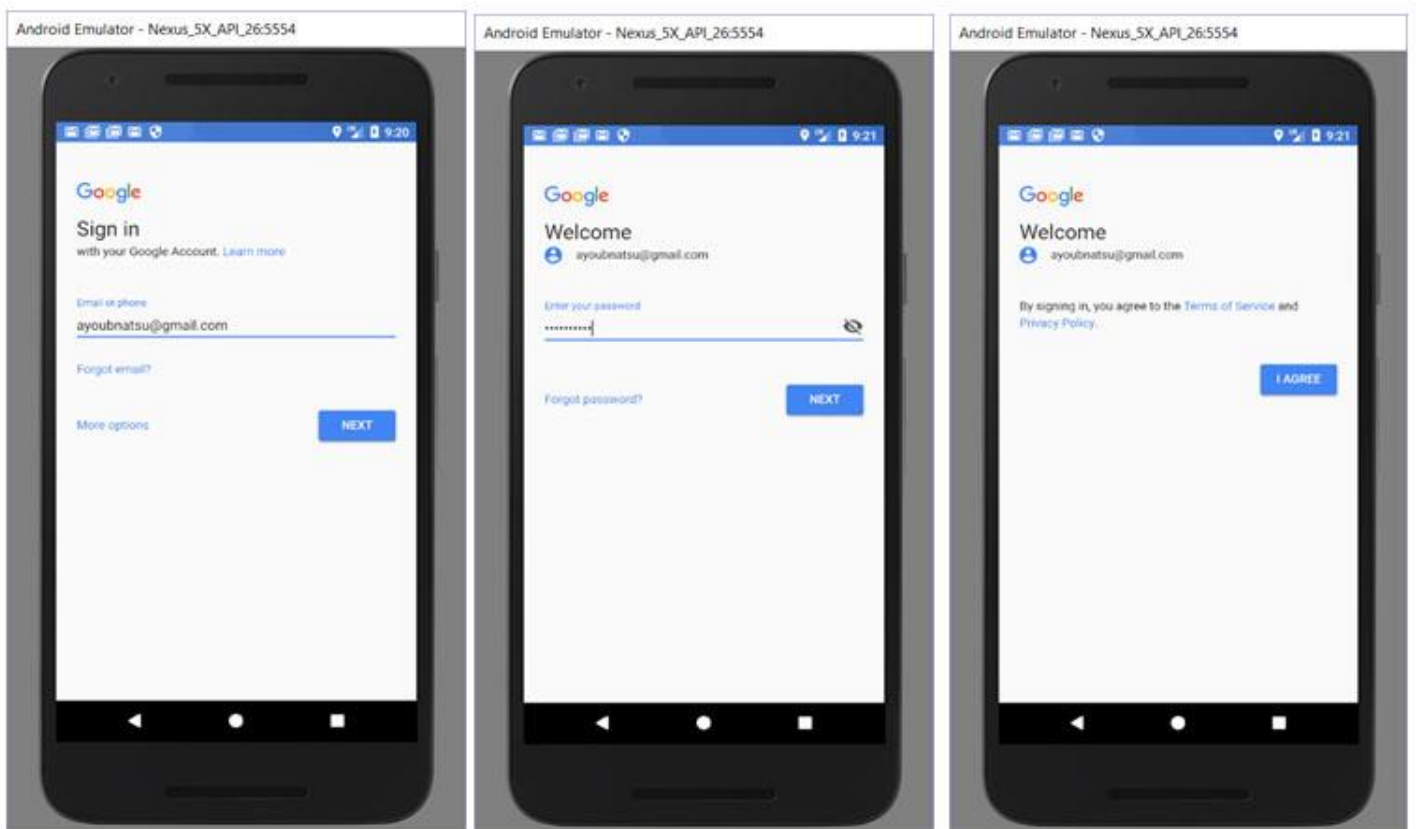


Figure 23 processus d'authentification

Si c'est une première authentification l'utilisateur doit effectuer un choix pour accéder à l'application. Soit qu'il est un conducteur ou un passager. Dans ce cas on va enregistrer l'utilisateur en tant que conducteur.

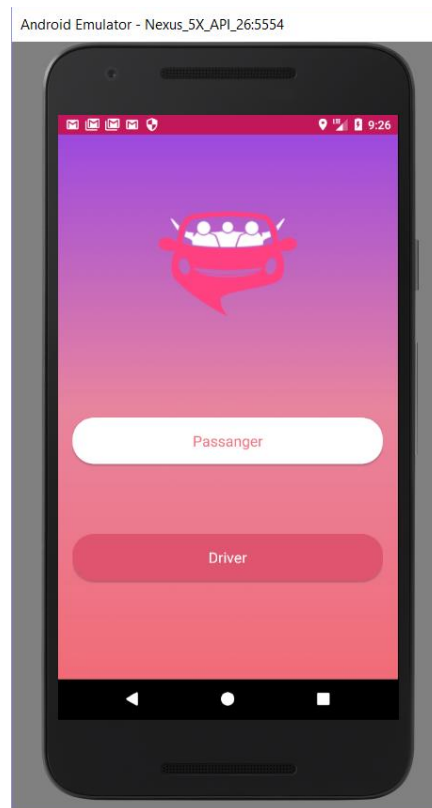


Figure 24 Interface pour effectuer le choix

On peut voir maintenant l'utilisateur dans la plateforme de firebase. Chaque utilisateur a un uid unique.

Search by email address, phone number, or user UID					ADD USER	↺	⋮
Identifiant	Providers	Created	Signed In	User UID ↑			
mixayoub@gmail.com		Jul 21, 2017	Jul 26, 2017	2wyYUxzCdHcCQyMfA0qJqSxMM...			
oussamamaaizate@gmail.co...		Nov 21, 2017	Nov 21, 2017	CsJlqcE2xdOzG7nN2oTkWYMr8gq1			
thepirateking95@gmail.com		Jul 19, 2017	Aug 22, 2017	DY92ZTaLxfgwWlP0l5OdRGNDpB2			
animeofficialreviews@gmail...		Nov 1, 2017	Nov 1, 2017	LblhvWd4FoOc0EAW0rhIC9o6SAA2			
ayoubnatsu@gmail.com		Nov 23, 2017	Nov 23, 2017	Ns467C4lI3Xqd3ipgECM4xZqRjJ2			
onetutchannel@gmail.com		Aug 17, 2017	Aug 26, 2017	PvC32JdbrffbybevW8ZNIeG8zYp1			
ayoubzeref@gmail.com		Jul 13, 2017	Nov 21, 2017	RMUi3JEJDRcBjtnpUzxV1jipchy1			
m.ayoub.bani@gmail.com		Jul 26, 2017	Nov 20, 2017	csVV2eP16GhOf1QU7htdbHZ7lCW2			
Rows per page: 50					1-8 of 8	<	>

Figure 25 les utilisateurs inscrit dans l'application

Home :

L'application a deux outils de navigation :

Navigation drawer :

Ce menu contient des fonctionnalités pas encore activées. Il permet d'accéder au profil de l'utilisateur, les paramètres et un Bouton pour se déconnecter, il permet aussi de partager l'application via les réseaux sociaux ou par message sms.

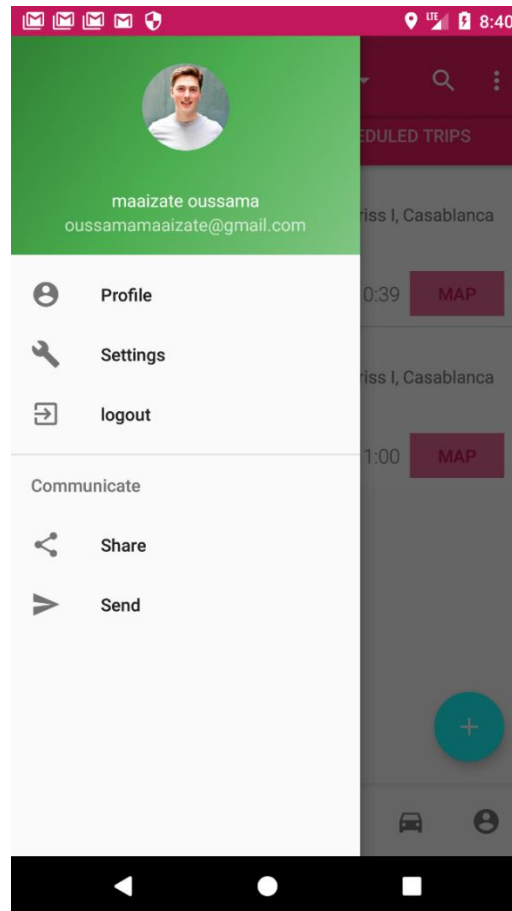


Figure 26 menu 'Navigation drawer'

Bottom navigation :



Figure 27 menu 'Bottom Navigation'

Permet d'accéder aux quatre principales interfaces :

-1^{er} icône : Home (Accueil de l'application).

-2^{ème} icône : Ajouter nouveau trajet.

-3ème icone : Consulter tous les messages reçus.

-4ème icone : Les trajets poster par l'utilisateur.

-5ème icone : profile de l'utilisateur.

Menu de déconnexion :

On haut il y a un menu noté par trois points en cliquant sur ce bouton on a l'option 'Sign out' pour se déconnecter.

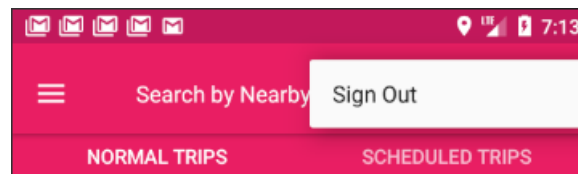


Figure 28 menu de déconnexion

Interface du conducteur

Accueil :

Cette interface est au format d'un viewPager qui permet de présenter deux fragment le premier (par défaut) Normal trips contient les postes pour des trajets ou parcours normale. Le deuxième est Scheduled trips qui concerne les postes pour des trajets programmé chaque fois. Noté que les postes sont postés par des passagers.

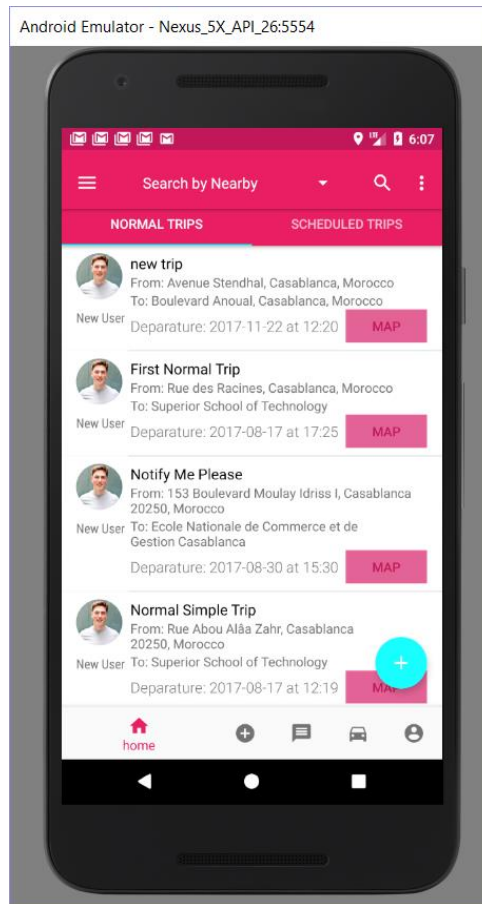


Figure 29 Accueil du conducteur

Les fragments contiennent comme contenu une listview dont chaque élément représente un post. Chaque élément est cliquable et il contient une image cliquable de l'utilisateur plus un bouton 'MAP' pour consulter le trajet sur le map. En haut il y a un menu représenté par un spinner. Ce menu permet de spécifier le critère de la recherche. Par défaut il y a le choix 'search by nearby', ce choix affiche les postes du plus proche par rapport à l'origine vers le plus loin. Puis il y a 'search by origine' pour la recherche par rapport à l'origine et 'search by destination' pour la recherche par la destination.

Si on clique sur l'image du profil on peut consulter le profil du passager :

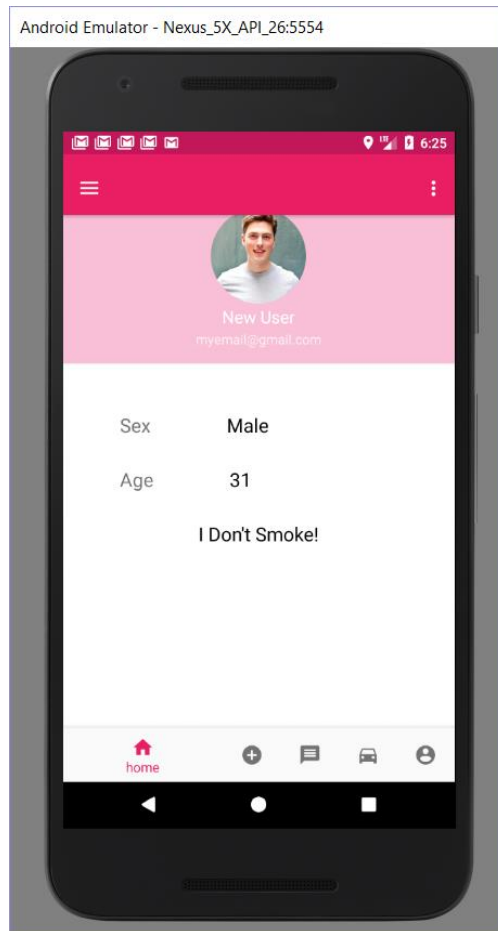


Figure 30 profile d'un passager

Le profile affiche les informations suivantes sur le passager (Nom, Prénom, email, Sexe, Age et s'il est fumeur ou non).

En cliquant sur le bouton 'MAP' on peut voir le trajet sur google Map. Le symbole 'A' indique l'origine et symbole 'B' indique la destination.

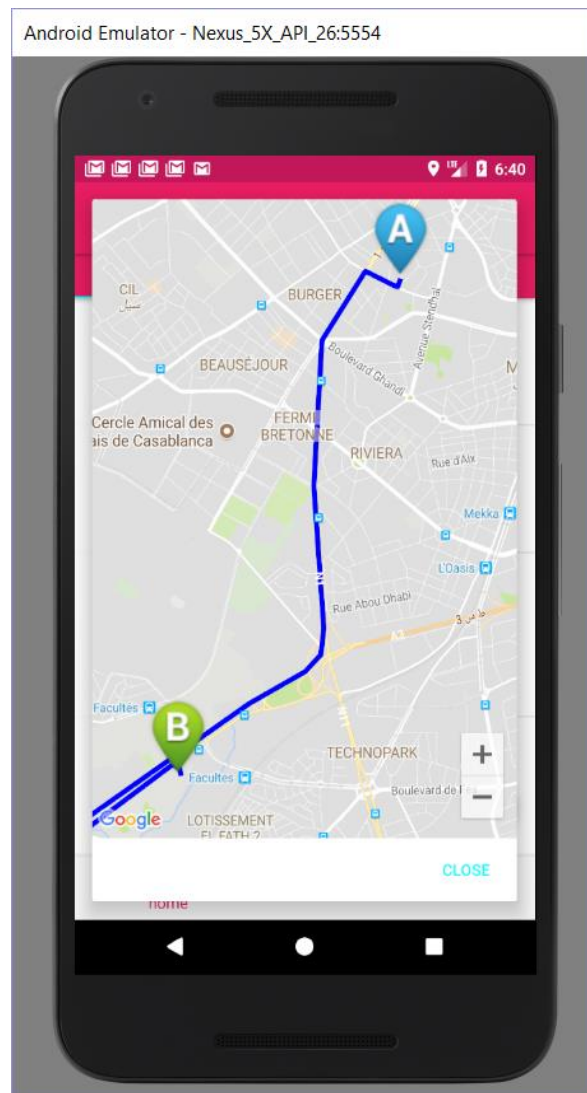


Figure 31 Map d'un trajet

Si on clique sur l'élément de la liste on peut voir le poste comme suit :

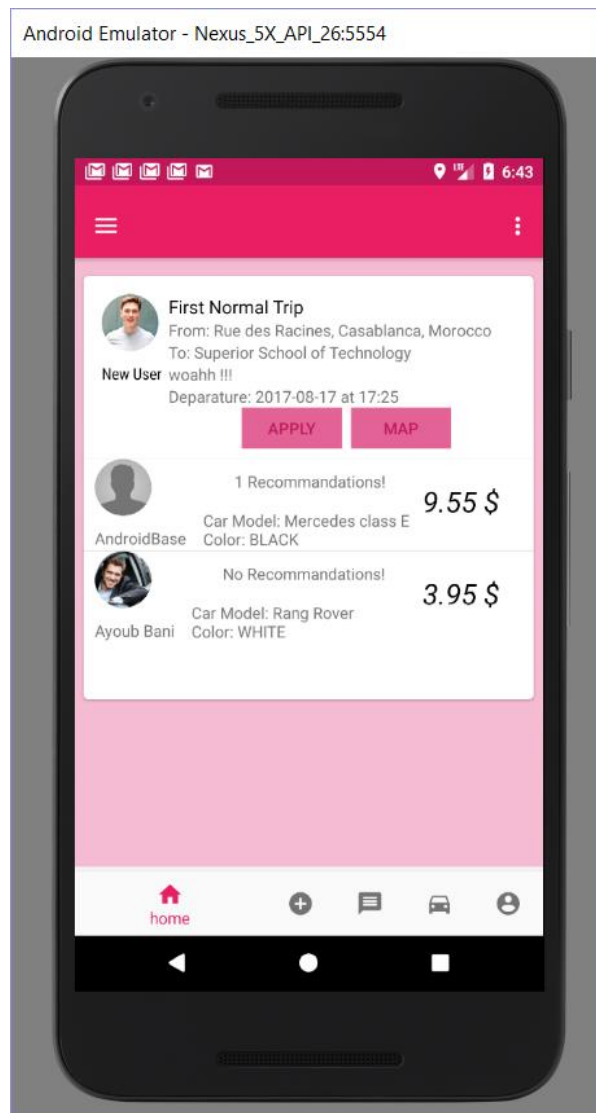


Figure 32 Post d'un passager

On peut consulter à partir de cette interface les informations sur le trajet telles que (Origine, Destination, date de départ et heure de départ). Il y a après le bouton 'map' et le bouton 'apply' pour envoyer une demande. En cliquant sur ce bouton on doit spécifier le prix que le conducteur propose au passager :

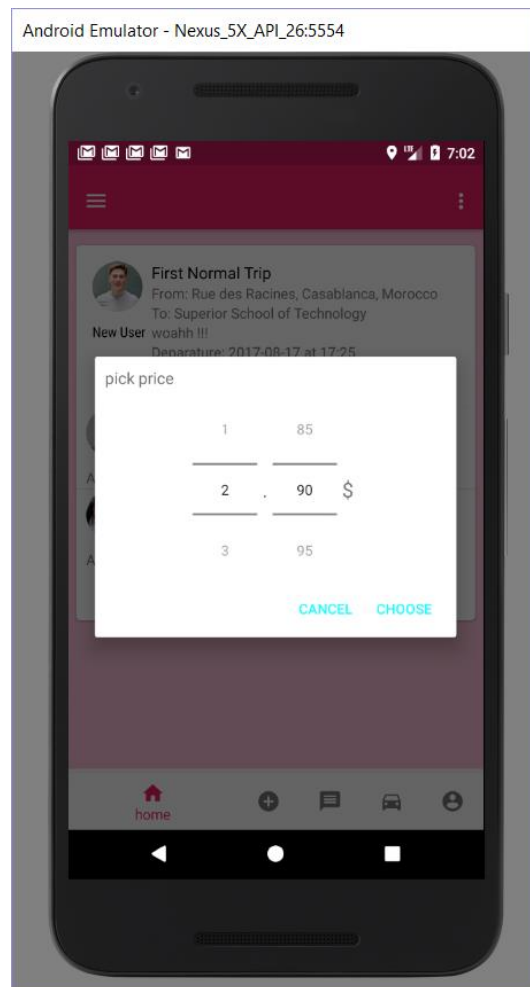


Figure 33 saisie du prix

Après on voit que la demande est ajoutée à la liste automatique sans avoir le rafraîchissement de l'interface c'est le but de l'utilisation de la base données en temps réel. Le bouton 'apply' est désactivé maintenant. Il reste de traiter la demande par le passager. Noté que les demandes sont triées par les recommandations des conducteurs. Un conducteur avec beaucoup de recommandations à la chance d'être choisi.

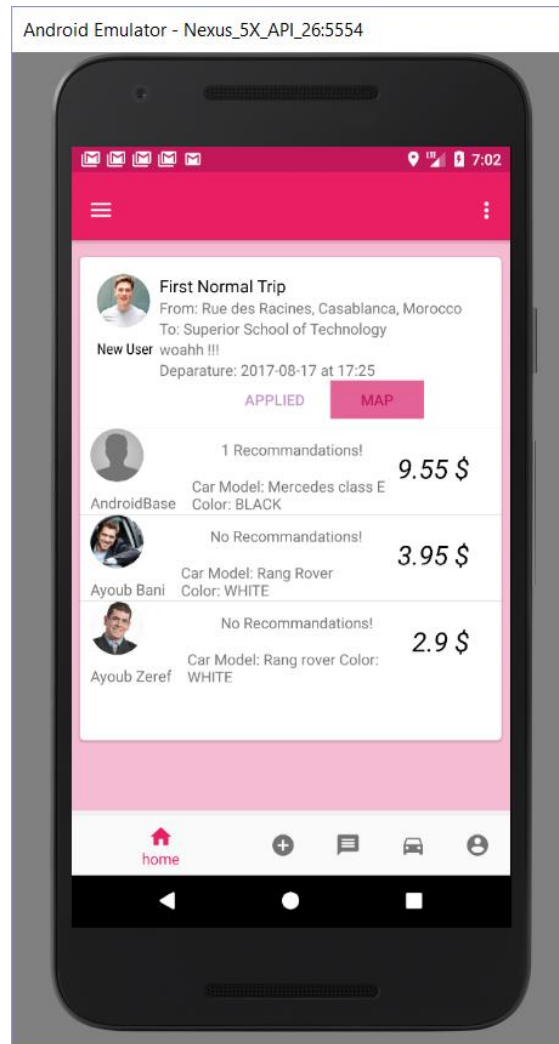


Figure 34 post du passager 'Nouvelle demande'

Sur l'interface home il y a un bouton bleu contenant un plus. Ce bouton permet de naviguer à l'interface de l'ajout d'un nouveau poste. On peut aussi y accéder par le navigateur on bas en cliquant sur le deuxième bouton plus. Cette interface est représentée comme suit :

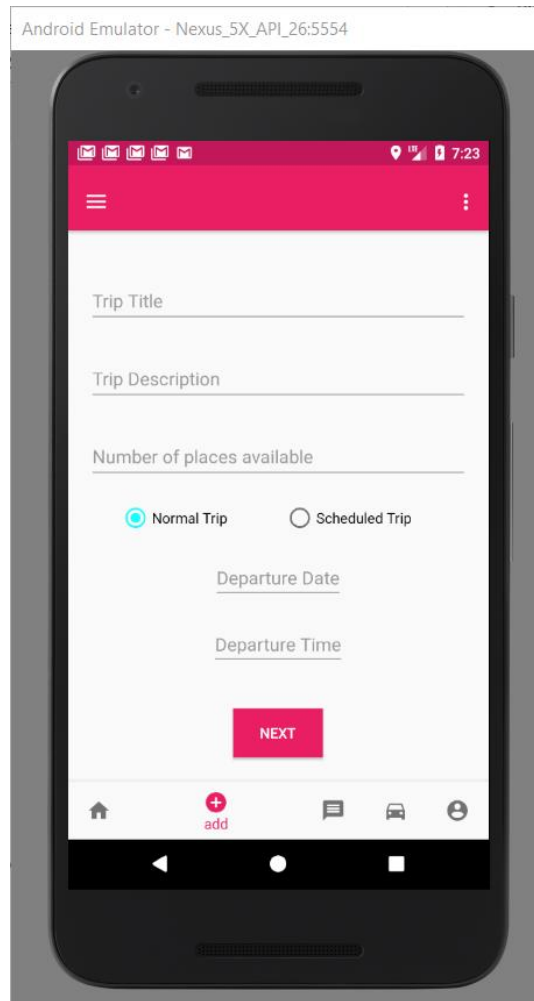


Figure 35 Interface ajouter nouveau post

Pour ajouter un nouveau post le conducteur doit donner un intitulé pour le trajet et une description. En suite le nombre de places disponible et après le conducteur doit spécifier s'il s'agit d'un trajet normal ou programmé. Par défaut c'est un trajet normal le conducteur doit spécifier date de départ et heure départ puis cliquer sur 'next'. On cas d'erreur il y a un message qui s'affiche exemple :

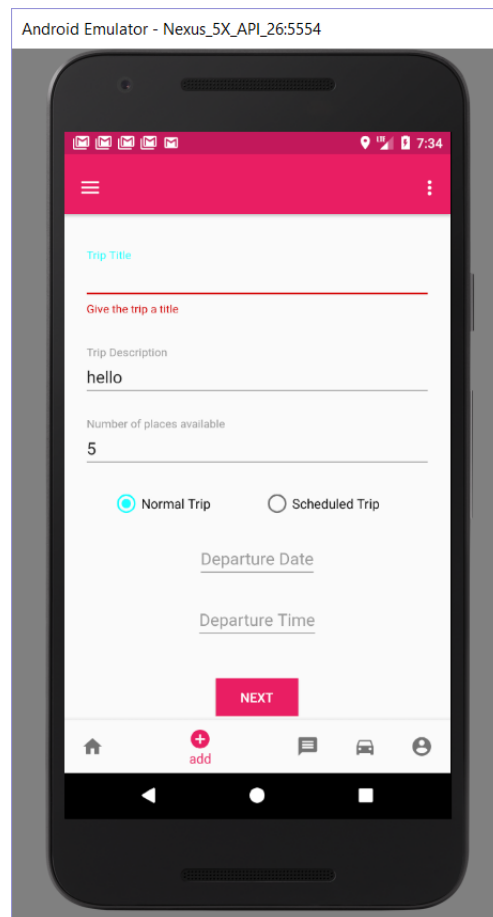


Figure 36 ajouter un nouveau post cas d'erreur

S'il s'agit d'un trajet programmé le conducteur doit spécifier la période (jour de commencement du trajet et jour de la fin du trajet) puis heure départ

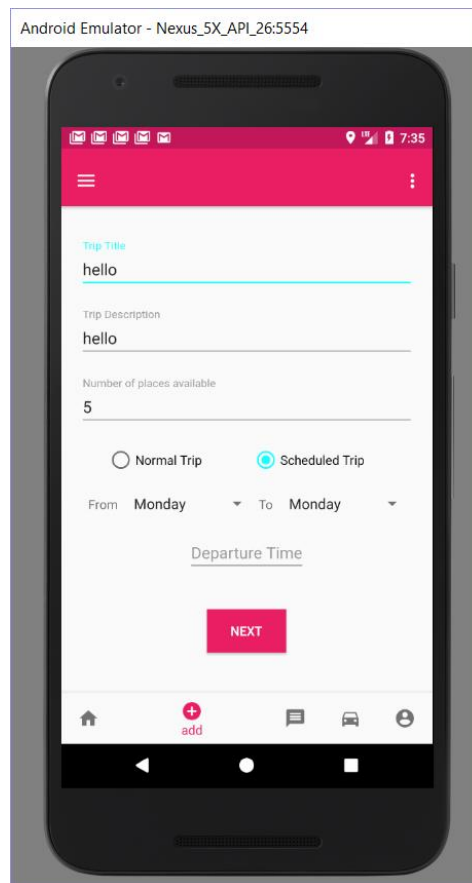


Figure 37 Ajouter un nouveau trajet programmé

Exemple choix d'heure :

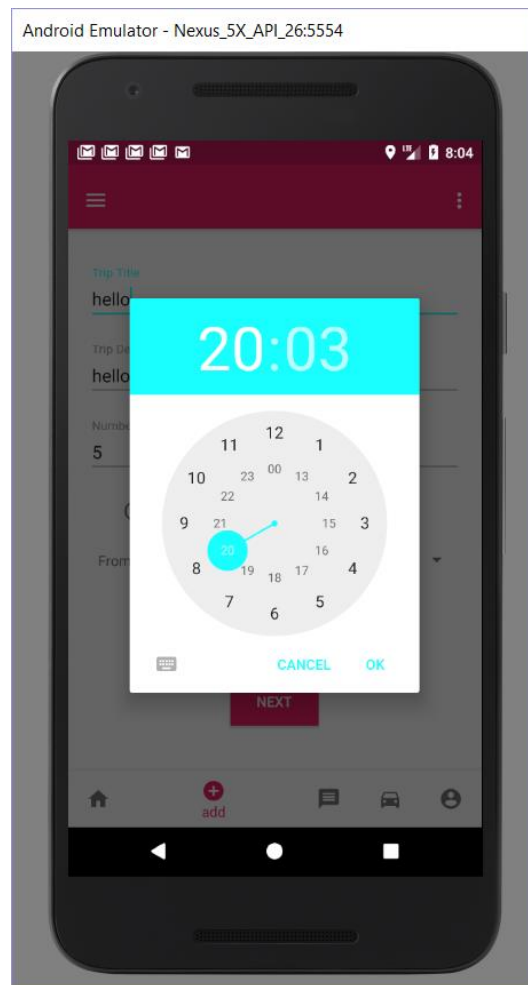


Figure 38 saisie de l'heure

Si les données sont bien saisies on peut procéder vers l'interface suivant pour spécifier le trajet. On constate le symbole qui pointe sur la localisation courante de l'utilisateur. Dans ce cas le conducteur a besoin de l'origine et la destination. Pour les préciser il y a deux méthodes :

-par l'outil autocomplète de Google qui permet de compléter une adresse d'une place précise :

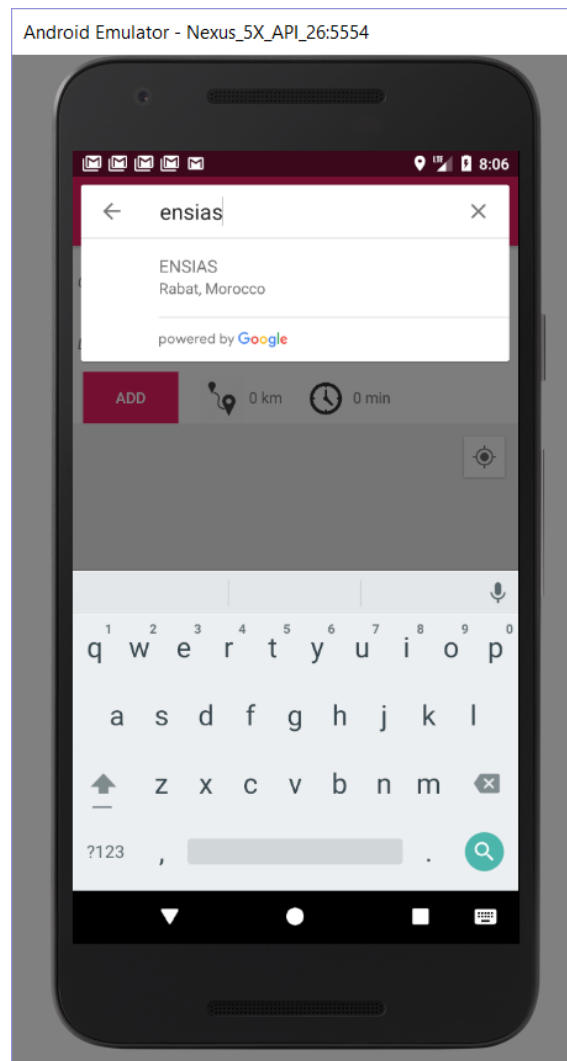


Figure 39 précision de la location par autocomplété

-par un clic sur la localisation sur le map :

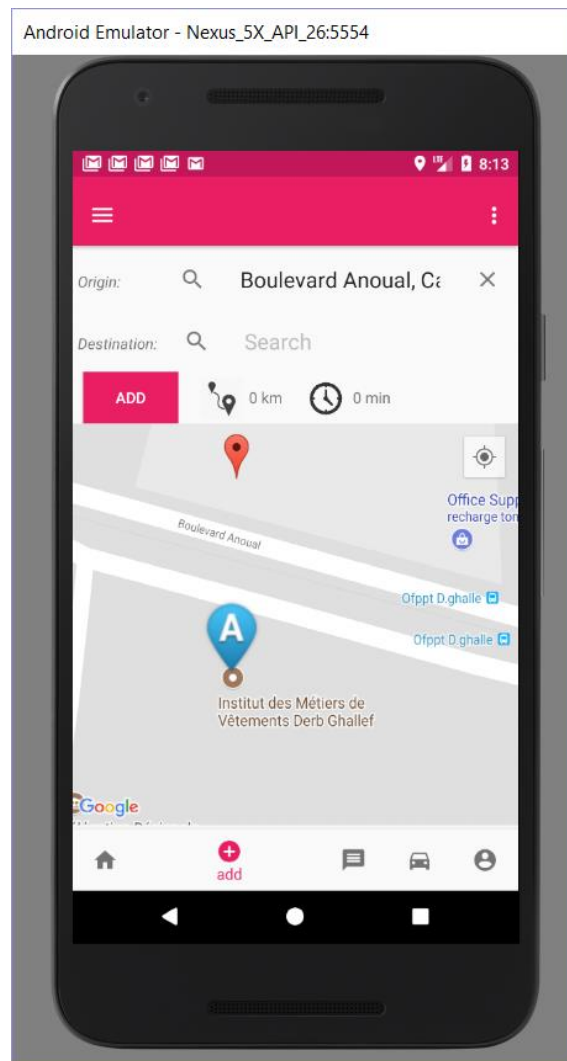


Figure 40 précision de la location par un clic sur le map

Après la spécification de l'origine et la destination l'application indique le chemin le plus court plus le nombre de kilomètres entre les deux points et le combien de minutes s'apprend pour arriver à la destination. Ces données sont reçues par l'api google de direction. On reçoit les informations sur la direction en précisant l'origine et la destination. Cette api renvoi sous format Json :

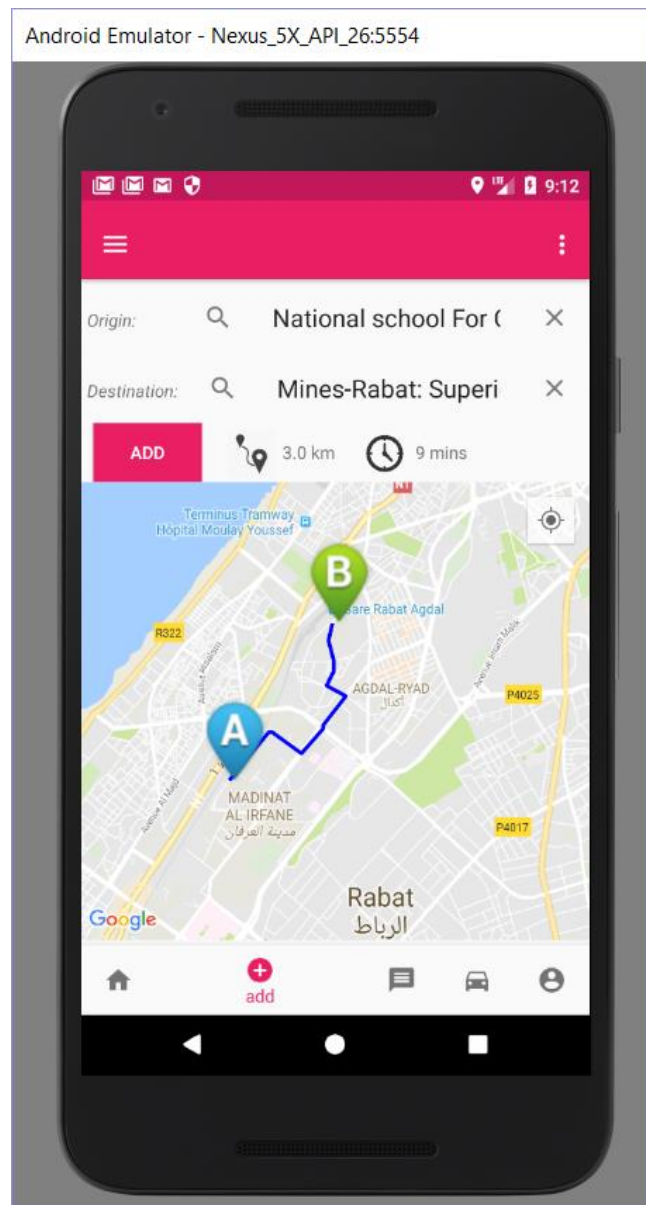


Figure 41 trajet entre origine et destination

L'api renvoi beaucoup de données exemple :

```

{
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJSyuW7edspw0ReA4weatTj3s",
      "types" : [ "route" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJUX2ygo5spw0Rsk2qn8omfFM",
      "types" : [ "route" ]
    }
  ],
  "routes" : [
    {
      "bounds" : {
        "northeast" : {
          "lat" : 33.9996831,
          "lng" : -6.851118199999999
        },
        "southwest" : {
          "lat" : 33.9847773,
          "lng" : -6.868314199999999
        }
      },
      "copyrights" : "Map data ©2017 Google",
      "legs" : [
        {
          "distance" : {
            "text" : "2.8 km",
            "value" : 2795
          },
          "duration" : {
            "text" : "9 mins",
            "value" : 551
          },
          "end_address" : "Avenue Ibn Sina, Rabat, Morocco",
          "end_location" : {
            "lat" : 33.9996831,
            "lng" : -6.851152
          },
          "start_address" : "Avenue Mohamed Ben Abdellah Erregragui, Rabat, Morocco",
          "start_location" : {
            "lat" : 33.9847773,
            "lng" : -6.868314199999999
          },
          "steps" : [
            {
              "distance" : {

```

Figure 42 données de l'api sous format json

Si le trajet est bien déterminé le post sera enregistrer sur la base de données.

Mes trajets :

Cette partie est dédié au conducteur pour consulter les trajets qu'il vient de poster. C'est similaire à l'interface de l'accueil, il y a deux fragments un pour les trajets normaux et l'autre pour les trajets programmés.

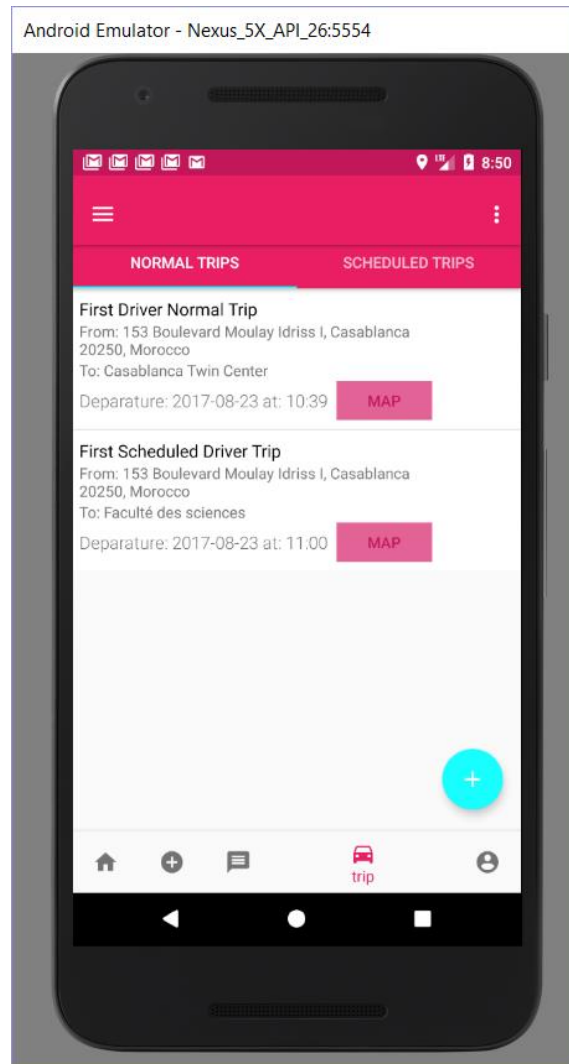


Figure 43 interface Mes trajets

Avec une clique sur l'un des trajets le conducteur peut consulter les passagers qui veut rejoindre ce déplacement. Le conducteur peut accepter, refuser ou bien contacter un passager (la partie messagerie n'est pas encore activée dans l'application).

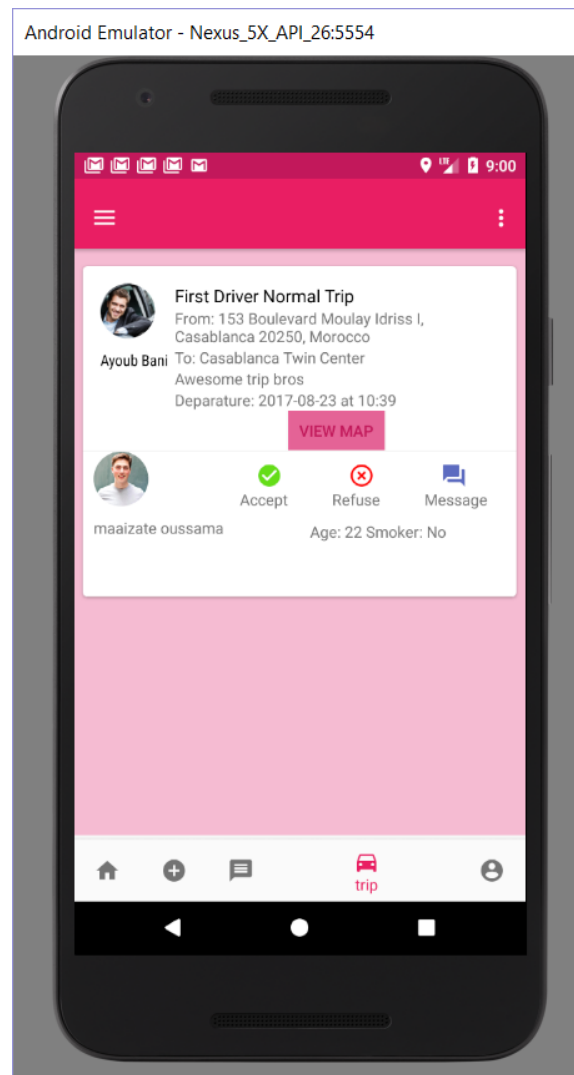


Figure 44 traitement demande des passagers

Profile :

Cette interface est aussi composée de deux fragments présenter par un viewpager. Il y a le fragment 'Reviews' qui présente les avis des passagers sur le conducteur (cette fonctionnalité n'est pas encore activée). L'autre fragment représente les informations sur la voiture du passager. Pour la première fois il faut changer la photo de profile. Le conducteur doit cliquer sur l'icône pour ajouter une nouvelle photo.

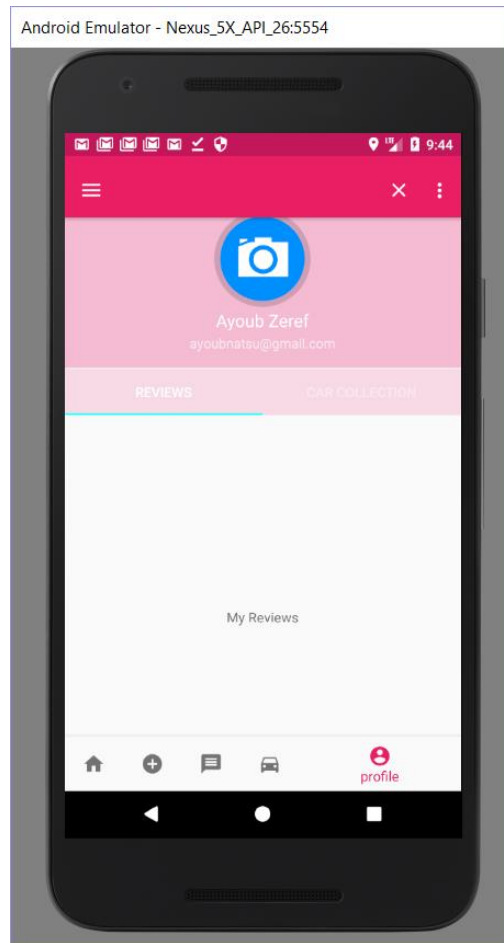


Figure 45 interface profile conducteur

Le conducteur doit choisi une photo dans sa galerie :

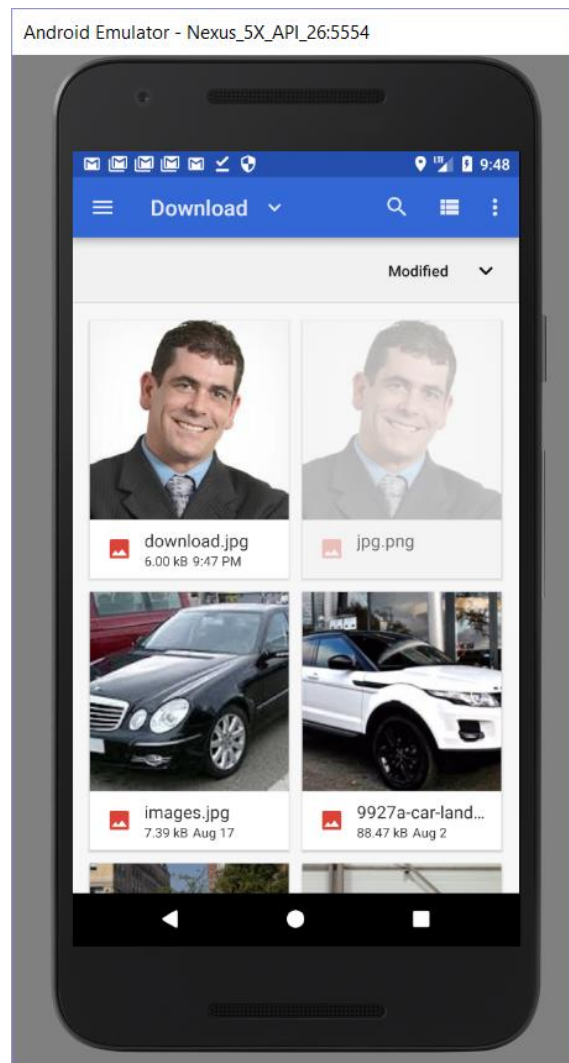


Figure 46 ajouter une nouvelle photo de profile

L'image est envoyée vers cloud storage de firebase l'image est maintenant modifiée :

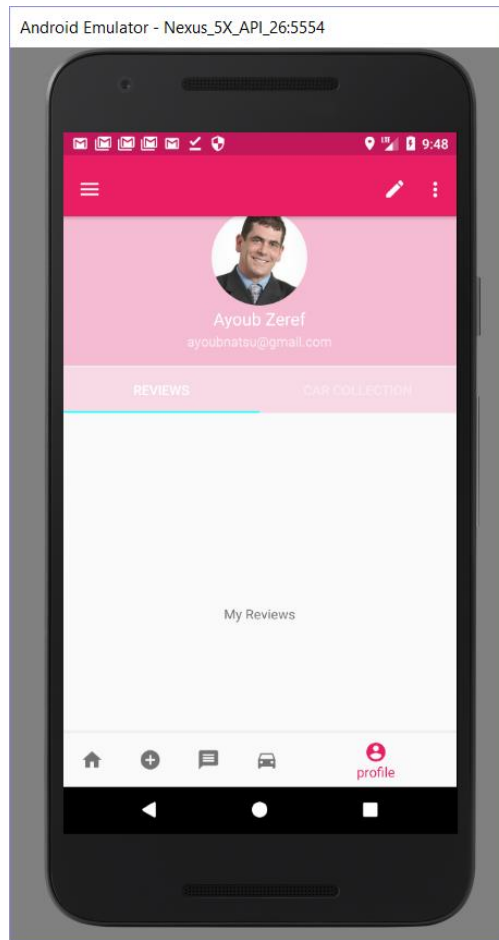


Figure 47 ajout de la photo de profile

L'autre fragment est dédié pour les informations sur la voiture.

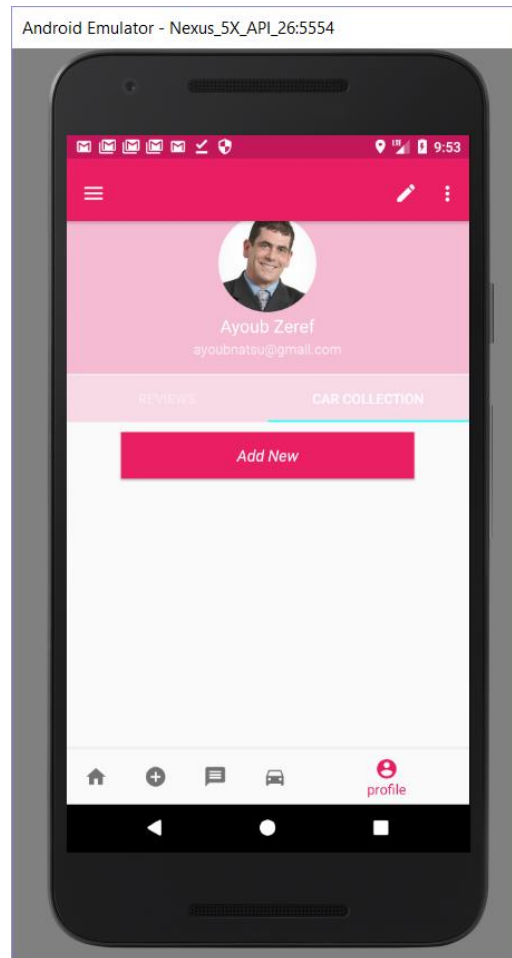


Figure 48 fragment ajouter une nouvelle voiture

Avant de poster le conducteur doit ajouter les informations de sa voiture. Pour l'ajouter il clique sur 'Add new' puis il remplit le formulaire qui contient les informations suivants (le modèle de la voiture, le nombre de places de la voiture, la couleur de la voiture et l'image de la voiture). Le conducteur peut ajouter après plusieurs images de voitures.

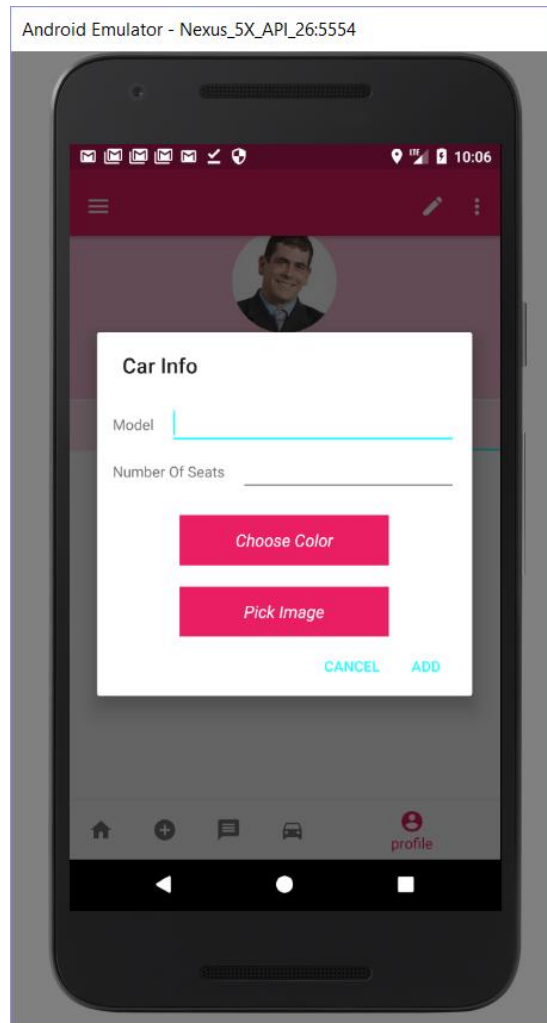


Figure 49 formulaire des informations de la voiture

Après l'ajout des informations de la voiture on peut les consulter immédiatement sur le fragment. 'Add New' maintenant permet d'ajouter seulement les photos. Le conducteur ne peut pas ajouter plus de 6 images.

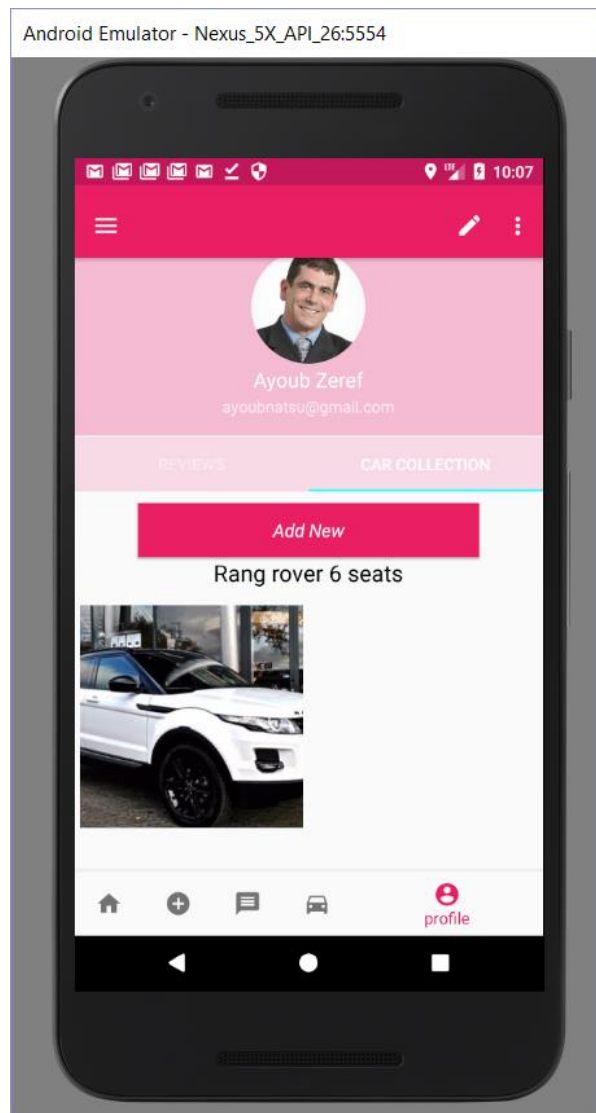


Figure 50 ajout d'une nouvelle image de la voiture

Avec un click on peut voir les images plus clairement



Figure 51 affichage claire de l'image

Interface du passager

Elle est similaire à l'interface du conducteur. Il s'agit de la même structure on va présenter seulement l'interface 'Mes Trajet' et le 'profile'.

Mes Trajet

Pour ces trajet le passager peut accepter ou refusé un conducteur. Si un conducteur est choisi les autres seront refusé.

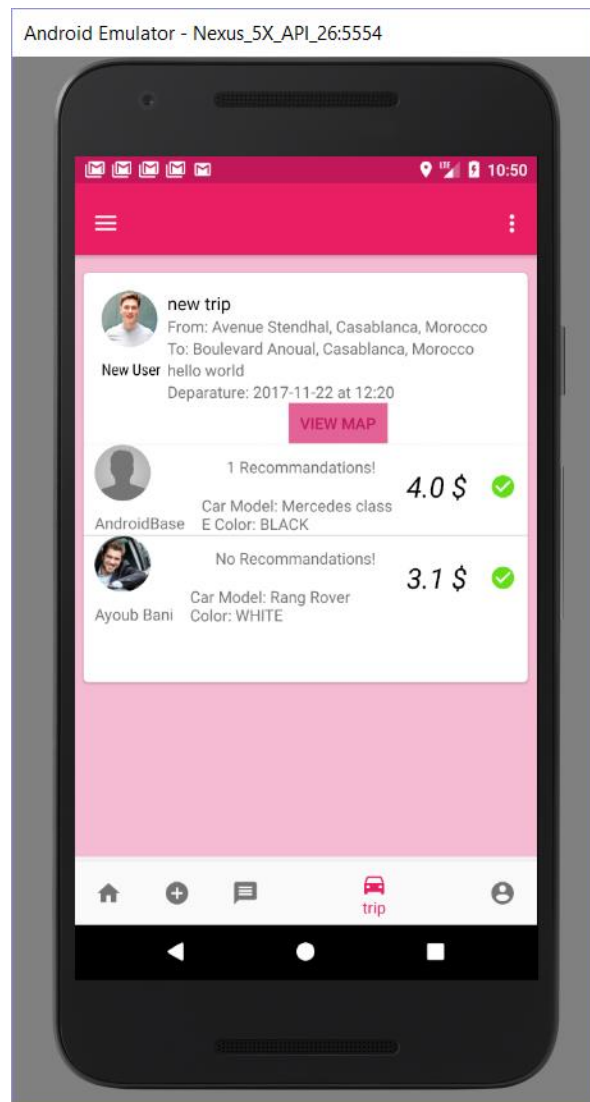


Figure 52 interface Mes trajet du passager

On voit qu'il y a un bouton en vert, si le passager clique sur ce bouton le conducteur concerné est choisi les autres sont refusés. Les boutons seront désactivés exemple :

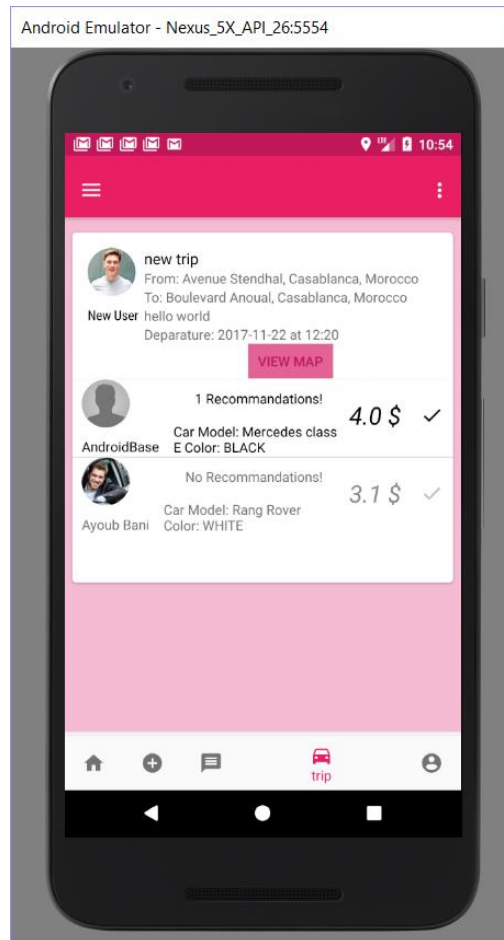


Figure 53 choix d'un conducteur

Profile

Le profile contient les informations du passager. Pour la première fois le passager doit ajouter ces informations en cliquant sur le bouton 'Add Informations'.

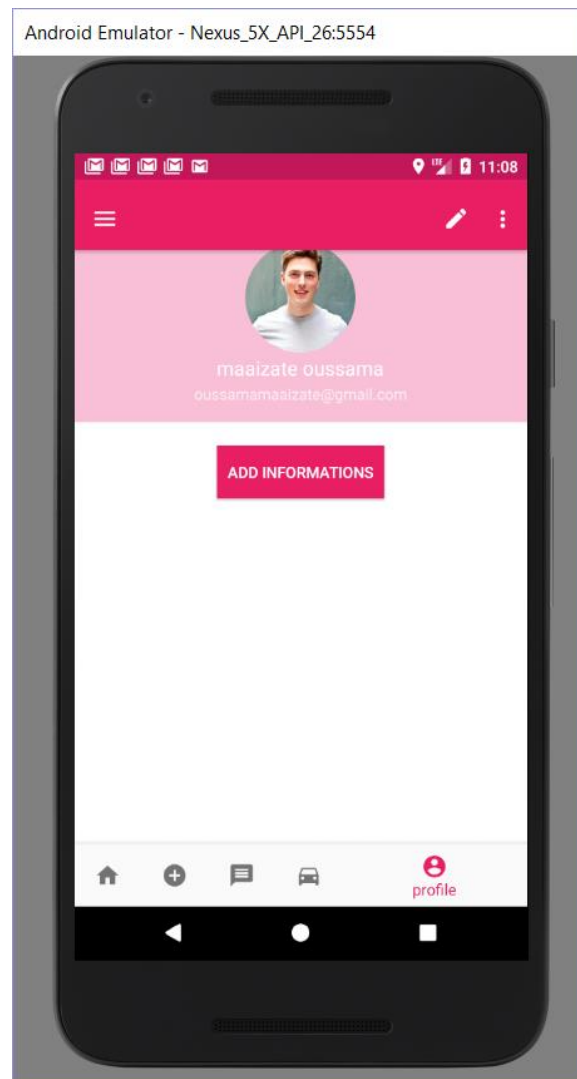


Figure 54 Profile passager ajout de nouveau informations

Un formulaire s'affiche pour la saisie des données suivants (Sexe, Date de naissance et s'il est un fumeur ou non).

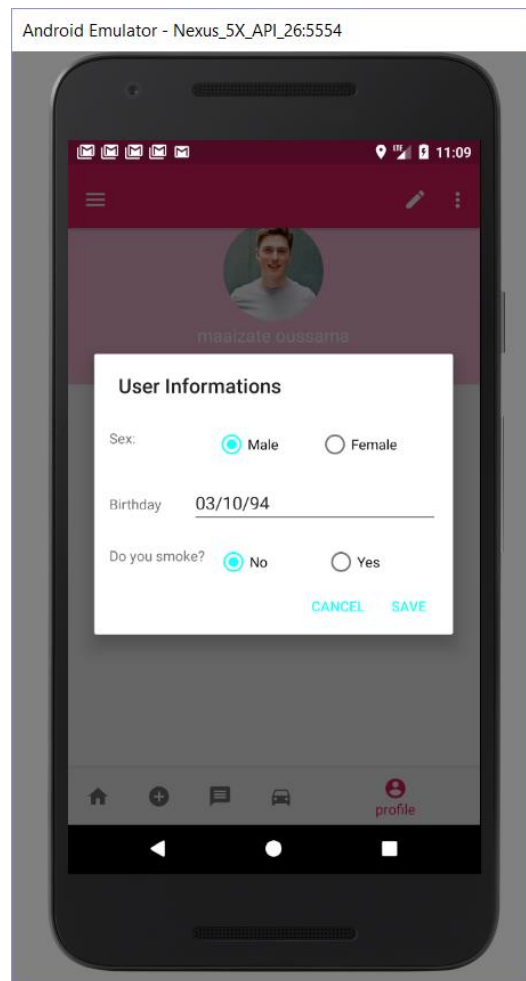


Figure 55 saisie des données du passager

Après le remplissage du formulaire les données sont affichées dans le profile et en voie que le bouton maintenant a comme fonction la modification des données personnelles du passager.

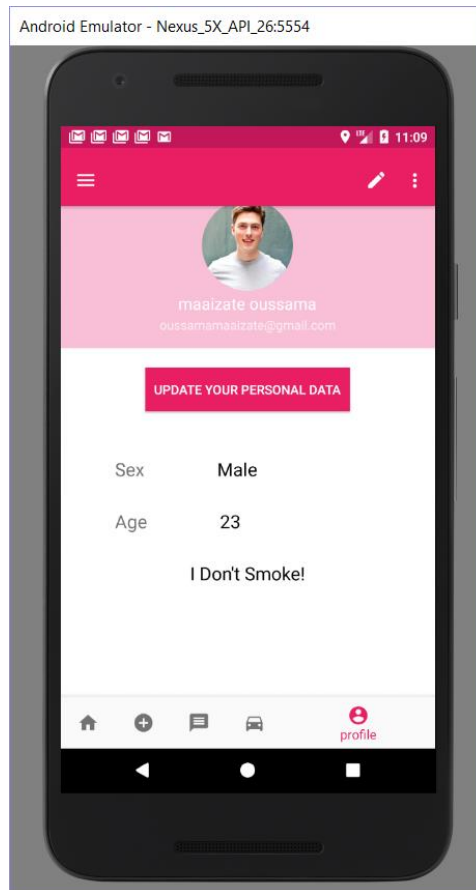


Figure 56 profile passager modifier

Conclusion Générale

En guise de conclusion, nous pouvons affirmer que le développement mobile est un domaine enrichissant au niveau expérimental et qu'il est en évolution constante.

Ce projet a été d'une grande utilité dans la mesure où il m'a permis de me familiariser avec le travail sur une nouvelle plate-forme à savoir la plate-forme Firebase. Il m'a permis d'approfondir mes connaissances théoriques et les mettre en pratique. Cette expérience a aiguisé mes capacités d'analyse et de synthèse, et m'a permis de renforcer mes connaissances concernant le développement mobile.

Enfin Ce stage a été pour moi l'occasion de découvrir et d'apprendre énormément de notions qui j'en suis sûr valoriseront mon projet professionnel. Il m'a permis d'acquérir de la méthode, de la rigueur ainsi que les technologies permettant le développement sur Android.

Webographie

<https://firebase.google.com/docs/android/setup>

<https://firebase.google.com/docs/auth/>

<https://firebase.google.com/docs/storage/>

<https://firebase.google.com/docs/database/>

<https://developer.android.com/guide/components/fragments.html>

<https://developer.android.com/training/implementing-navigation/navigation-drawer.html>