

Université Internationale de Casablanca

# Rapport du Mini-Projet

Ayman Radi - Hamza ARROUBAT – Moad Kacemi

GENIE INFORMATIQUE  
2018-2019

## Table des matières

INTRODUCTION AU MINI-PROJET .....	- 3 -
<b>1. PRESENTATION DU PROJET.....</b>	<b>- 4 -</b>
<b>1.1. Problématique .....</b>	<b>- 4 -</b>
<b>2. PHASE DE PROTOTYPAGE.....</b>	<b>- 4 -</b>
<b>2.1. BESOINS FONCTIONNELS .....</b>	<b>- 4 -</b>
<b>3. CHOIX DES TECHNOLOGIES UTILISES .....</b>	<b>- 5 -</b>
<b>3.1.1. Android .....</b>	<b>- 5 -</b>
<b>3.2. FIREBASE .....</b>	<b>- 6 -</b>
<b>3.2.1. Firebase Authentication .....</b>	<b>- 6 -</b>
<b>3.2.2. Realtime Database .....</b>	<b>- 6 -</b>
<b>3.2.3. Firestore .....</b>	<b>- 7 -</b>
<b>3.2.4. Crashlytics .....</b>	<b>- 7 -</b>
<b>3.3. GOOGLE CLOUD PLATFORM .....</b>	<b>- 7 -</b>
<b>3.3.1. Google Maps .....</b>	<b>- 8 -</b>
<b>3.4. MAPKIT .....</b>	<b>- 8 -</b>
<b>4. PHASE DE CONCEPTION .....</b>	<b>- 9 -</b>
-	
<b>4.1. PARTIE LOGIN .....</b>	<b>- 9 -</b>
<b>4.1.1. Android .....</b>	<b>- 9 -</b>
<b>4.2. PAGE D'ACCUEIL .....</b>	<b>- 9 -</b>
<b>4.2.1. Android .....</b>	<b>- 9 -</b>
<b>4.3. PAGE NAVIGATION .....</b>	<b>- 10 -</b>
<b>4.3.1. Android .....</b>	<b>- 10 -</b>
<b>4.4. PAGE PROFILE .....</b>	<b>- 11 -</b>
<b>4.4.1. Android .....</b>	<b>- 11 -</b>
<b>CONCLUSION .....</b>	<b>- 12 -</b>

## **Introduction au mini-projet**

Tout au long du semestre, nous avons eu la chance de mener un mini-projet sur le domaine du « développement mobile ». Ce projet nous a permis d'acquérir de nombreuses compétences essentielles telles que l'organisation, l'auto-éducation et, surtout, on a dû développer un esprit d'analyse en filtrant l'énorme quantité d'informations afin de mieux structurer les idées.

L'une des particularités de ce module est de travailler sur un projet d'envergure avec une quantité de données importante, une approche de l'utilisateur comme entité inconnue et complexe à cerner. Il y a aussi la partie qualité qui est essentielle à tout produit destiné au grand public.

C'est à la fois motivant et frustrant, puisque nous n'avons que rarement l'occasion de pouvoir travailler sur des sujets réels et pertinent dans le monde des technologies présent dans le marché et surtout de pouvoir faire le tour de tous les aspects.

Bien entendu, nous n'avons jamais fait de développement mobile dans le cadre de notre formation académique.

Le présent rapport a pour objectif d'expliquer toutes les phases du projet qui a duré plus de 5 mois.

Certains éléments complexes sont ajoutés progressivement pour satisfaire la curiosité des lecteurs les plus avancés.

À travers ces pages, nous essayerons de donner un aperçu général du projet, nous parlerons aussi de la partie réflexion et réalisation et enfin nous exposerons les résultats finaux.

## 1. Présentation du projet

### 1.1. Problématique

- \_ Retard bus
- \_ Incertitude des bus

## 2. Phase de prototypage

### 2.1. Besoins fonctionnels

L'étape de Benchmarking et celle d'étude des besoins réalisées, nous pouvons maintenant passer à l'énumération des différentes fonctionnalités de l'application qui seront implémenter afin de pouvoir offrir aux utilisateurs tous les éléments nécessaires pour pouvoir venir en aide aux personnes prenant le bus.

Parmi les fonctionnalités, nous pouvons retrouver :

- Le signalement :

Le signalement est considéré comme étant la fonctionnalité mère de l'application, ce dernier donnera la possibilité aux Chauffeurs en se connectant de signaler leurs présences dans une localisation spécifique.

Une fois le signalement établi, il sera mis au niveau de la map de l'application qui rassemblera l'ensemble des signalements fait par les autres chauffeurs.

- La navigation :

La navigation comportera une Map (carte) mettant en évidence la localisation précise des différents chauffeurs qui ont signalés leurs présence grâce à l'application, ainsi que la distance en temps réel séparant ces derniers à l'utilisateur en question.

Comme au niveau des autres applications de géolocalisation, les endroits où les signalements ont été réalisé seront modélisé sous forme de curseurs « pins » au niveau de la page navigation.

- Authentification des utilisateurs :

La gestion des utilisateurs dans un réseau social est cruciale pour son bon fonctionnement, d'ailleurs c'est sur quoi se basent ce genre d'application, de plus, nous avons intégré les systèmes d'authentification les plus évolués, sécurisés et appréciés par les différents utilisateurs d'application mobile connus à ce jour et qui sont l'authentification avec Facebook ou Google.

- Profile utilisateur :

Chaque utilisateur bénéficiera d'une page profile contenant les informations suivantes :

- Le nom. ○ Le prénom. ○ localisation qui change en temps réel. ○ Une photo de profil.
- Un volet réservé au niveau du profil pour ses différents signalements actifs.

- Un système de notification :

Comme toute application connue à ce jour, l'application Mdina-Pocket comportera un système de gestion de notification. Ce système sera implémenté dans le but d'alerter un utilisateur spécifique de la création d'un signalement réalisé par un chauffeur lambda.

### 3. Choix des technologies utilisés

#### 3.1. Android

##### 3.1.1. Java

Pour le développement Android, nous avons décidé d'utiliser le logiciel Android Studio qui est un logiciel optimisé pour la réalisation d'applications Android en utilisant le langage Java.

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, MacOS ou GNU/Linux, avec peu ou pas de modifications, mais qui ont l'inconvénient d'être plus lourd à l'exécution (en mémoire et en temps processeur) à cause de sa machine virtuelle. Pour cela, divers plateformes et Framework<sup>1</sup> associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

##### 3.1.2. Android Studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle.

#### 3.2. Firebase

##### 3.2.1. Firebase Authentication

La plupart des applications doivent connaître l'identité d'un utilisateur. Connaître 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.

---

<sup>1</sup> **Framework** : plateforme logicielle

Identifier	Providers	Created	Signed In	User UID ↑
omarmhaimdat@gmail.com	Google	Mar 9, 2019	Jul 2, 2019	HsfRyHaKiNeE3slvzYPeQPYB6jO2
omarmhaimdat@hotmail.fr	Facebook	Mar 11, 2019	Jul 1, 2019	ZqALnX01IZRYALvHK1Ef0eeXQYg2
mohamedmhaimdat@gmail...	Google	Apr 16, 2019	May 25, 2019	dipL0gwh4abcaHTX1UTtjBAyEYE2
aymanehachcham1310199...	Google	May 22, 2019	May 22, 2019	fbqQ2HRt1iTuVb6vqyMyMQ2jQd32
anasokok@gmail.com	Google	Apr 19, 2019	Jun 3, 2019	zjvwIV5GyBYtPjJ96Yu2p4DUZto1

Figure 11: Gestion des utilisateurs avec Firebase Authentication

L'authentification Firebase fournit des services d'arrière-plan, des kits de développement logiciel (SDK) faciles à utiliser et des bibliothèques d'interface utilisateur prêtes à l'emploi pour authentifier les utilisateurs auprès de notre application. Il prend en charge l'authentification à l'aide de mots de passe, de numéros de téléphone, de fournisseurs d'identité fédérés populaires tels que Google, Facebook et Twitter, etc.

L'authentification Firebase s'intègre étroitement à d'autres services Firebase et s'appuie sur les normes de l'industrie, telles que OAuth 2.0 et OpenID Connect, pour s'intégrer facilement à notre système personnalisé.

### 3.2.2. Realtime Database

La base de données en temps réel Firebase est une base de données hébergée dans le cloud. Les données sont stockées sous forme de JSON et synchronisées en temps réel sur chaque client connecté. Lorsque nous créons des applications multiplateformes avec les kits de développement logiciel (SDK) iOS, Android, tous les utilisateurs partagent une instance de base de données en temps réel et reçoivent automatiquement les mises à jour avec les données les plus récentes.

Présentée comme étant un arbre, Realtime database offre une interface simple à utiliser et à lire tel que les tables connues au niveau des bases de données relationnelles seront implémentées comme des collections englobant une multitude d'objets auxquels nous pouvons accéder par le biais d'une clé générique créée par Realtime database elle-même lors de l'insertion d'un nouvel objet au niveau de la collection en question. Vu que c'est du NoSQL, nous ne pouvons pas créer de relations entre les différentes collections comme à la façon des bases de données relationnelles néanmoins chaque objet de collection dépendant d'un objet d'une collection différente comportera les données jugées nécessaires pour lui de l'objet duquel il dépend.

### 3.2.3. Firestore

Cloud Firestore est une base de données flexible et évolutive pour le développement mobile, Web et serveur de Firebase et de Google Cloud Platform. Comme la base de données en temps réel Firebase (Realtime Database), elle synchronise nos données entre les applications clientes et notre back office en temps réel. Elle offre également une prise en charge hors ligne pour mobile et Web, de sorte que nous puissions créer des applications réactives fonctionnant indépendamment de la latence du réseau ou de la connectivité Internet. Cloud Firestore offre également une intégration transparente avec d'autres produits Firebase et Google Cloud Platform, y compris les fonctions cloud (Pour la maintenance de la base de donnée).

### 3.2.4. Crashlytics

Firebase Crashlytics est un reporter de crash léger en temps réel qui nous permet de suivre, de hiérarchiser et de résoudre les problèmes de stabilité qui érodent la qualité de nos applications (iOS et Android). Crashlytics nous fait gagner du temps en résolvant les problèmes de crash en mettant en évidence les circonstances qui les ont provoqués.

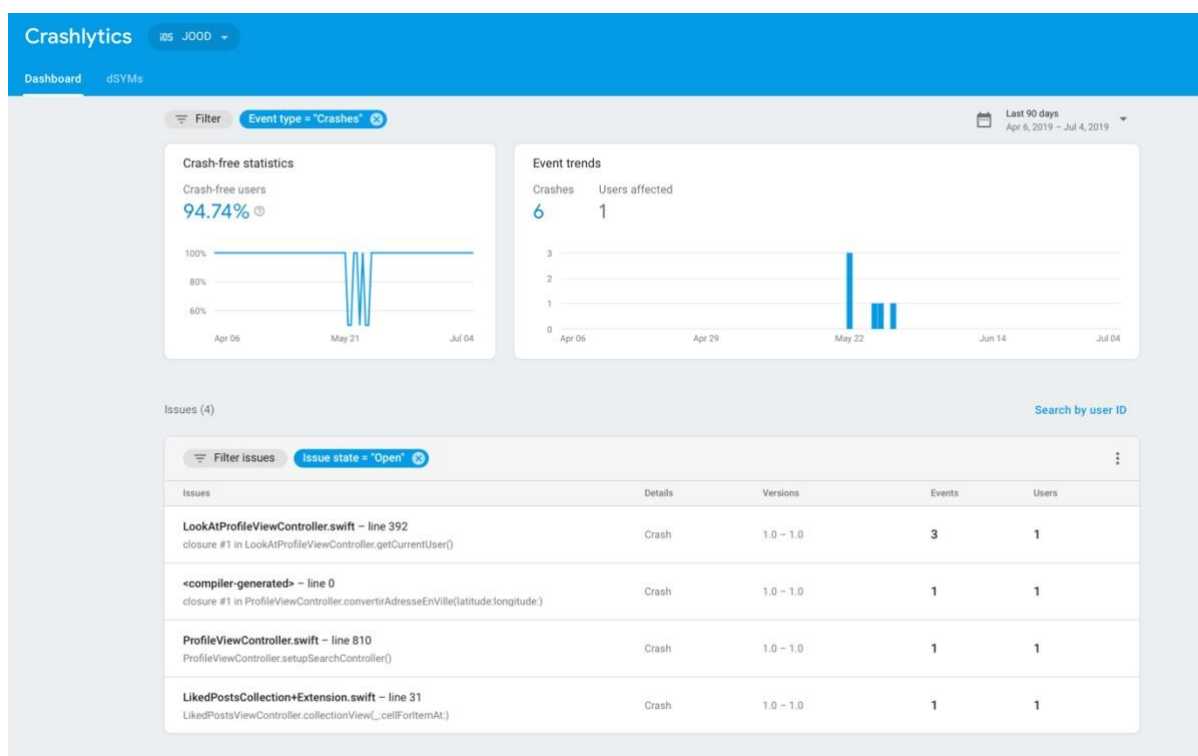


Figure 12: Exemple de notre implémentation de Crashlytics

On peut identifier si un accident particulier affecte de nombreux utilisateurs. On reçoit des alertes lorsqu'un problème augmente soudainement en gravité. On peut aussi déterminer quelles lignes de code provoquent des plantages.

### **3.3. Google Cloud Platform**

#### **3.3.1. Google Maps**

Google Maps est un service de géolocalisation en temps réel développé par Google. Nous avons pensé à l'intégrer dans le but de donner aux utilisateurs la possibilité de localiser au niveau d'une Map les différents signalements d'SDF réalisé. Google met à notre disposition une API<sup>2</sup> riche en méthode réalisant des fonctionnalités qui vont de l'affichage de la carte de la ville (Map) au niveau de la page navigation jusqu'à leurs fournir l'itinéraire menant à la personne sans domicile fixe en question. Cet API a surtout été utilisée pour la version Android de l'application.

#### **3.4. MapKit**

MapKit est une API disponible pour les applications iOS. Similaires à Google Maps, elle nous permet d'afficher une carte, à mettre des points sur des localisations spécifiques et aussi à pouvoir donner des itinéraires.

Il est à noter que des deeplinks vers des applications de géolocalisation comme Waze Maps (iOS), Google Maps et Apple Maps ont été rajouté dans le but de satisfaire les utilisateurs familiers avec l'une de ces applications sans oublier que cela nous permet d'avoir une précision de la localisation générée par notre signalement.

---

<sup>2</sup> API : interface de programmation applicative



## 4. Phase de conception

### 4.1. Partie Login

#### 4.1.1. Android

La page Login fait office de première interface accueillant l'utilisateur et de la même façon elle esquisse l'aspect visuel général de l'application.

En ce qui concerne les fonctionnalités de la page, on a implémenté trois possibilités différentes d'inscription : Automatiquement par l'intermédiaire de Facebook et de Google ou manuellement en entrant son propre e-mail et mot de passe.

Si l'utilisateur possède déjà un compte l'enregistrement se fait en comparant les informations d'inscription avec celles présentes dans la base de données pour lui permettre l'accès. Dans le cas où il s'agit d'un nouvel utilisateur on a mis en place toute une procédure de création de compte pour l'enregistrer auprès du service d'authentification de Firebase.

### 4.2. Page d'accueil

#### 4.2.1. Android

La page d'accueil est considérée comme étant la partie la plus importante de notre application. Il existe deux différentes pages d'accueil dans l'application :

La première celle d'un utilisateur (chauffeur), contient essentiellement trois boutons dont pour chaque une fonctionnalité parmi les suivantes [signaler une panne, signaler le fonctionnement ou arrêt de fonction (pendant les pauses), donner le numéro de la plaque du bus utilisé].

La deuxième celle d'un utilisateur (client), contient essentiellement trois boutons dont pour chaque une fonctionnalité parmi les suivantes [acheter un billet, consulter les infos des bus en questions, faire une réclamation]

En termes de design la page d'accueil comprend un Toolbar contenant un menu hamburger qui permet l'accès au « drawer layout<sup>3</sup> » affichant quelques informations de l'utilisateur issue de la collection « Users » se trouvant au niveau de Firebase Realtime database tel que son nom, prénom et email. Ce dernier permettant aussi la navigation au niveau des

---

<sup>3</sup> **Drawer layout** : disposition en tiroir <sup>6</sup>

**Toolbar** : barre outils

différentes pages de l'application comme sa page profile, modification profile et une page dédiée au paramètres de l'application.

En bas de la page accueil, nous avons implémente un deuxième moyen de navigation sous forme de navigation bar contenant 4 liens vers les pages principales de l'application qui se trouvent être :

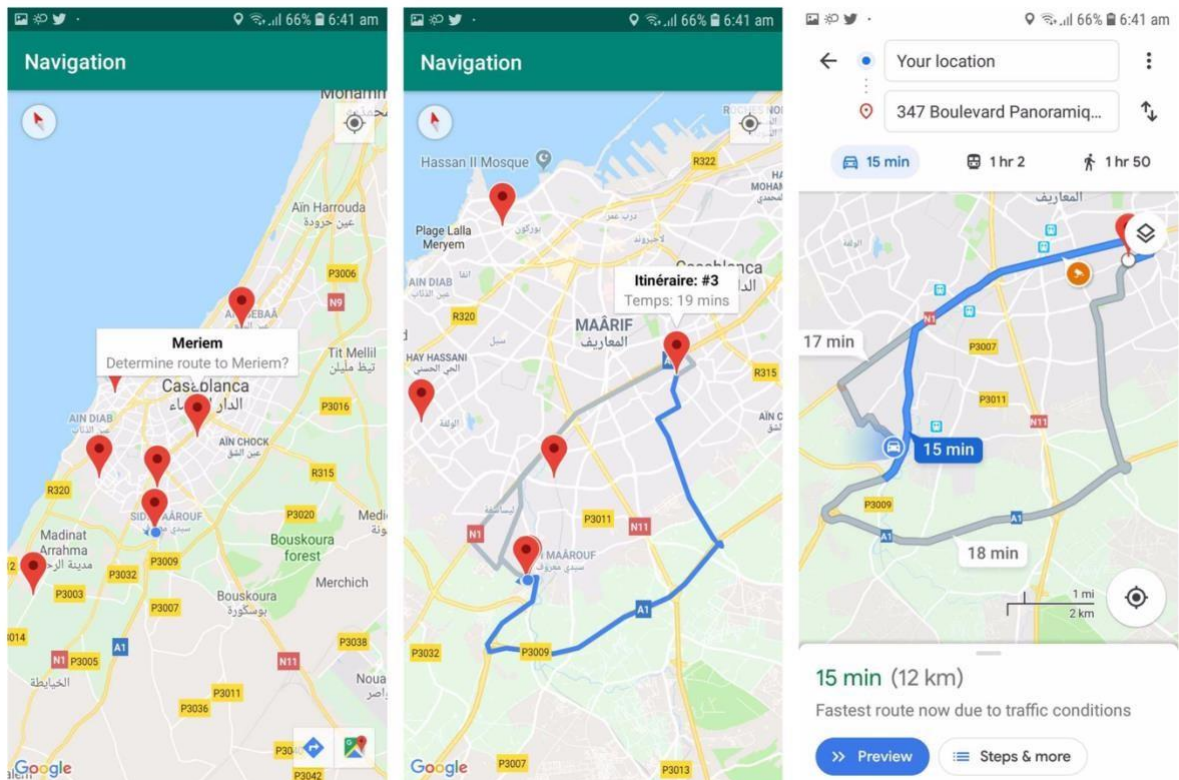
- La page d'accueil
- La page profile
- La page navigation
- La page notification

Ce type de navigation est le plus optimal c'est pour cela que nous avons jugé bon de l'intégrer afin d'offrir la meilleure expérience utilisateur possible.

### **4.3. Page navigation**

#### **4.3.1. Android**

La page navigation sera une page qui mettra en évidence les différents signalements des chauffeurs.



*La page navigation | Android*

En termes de design, la page navigation comportera une MapView dynamique que l'on aura mis en place grâce à l'API Google Maps. Cette dernière affichera l'ensemble des signalements en fonctions des données de localisations de chacun que l'on aura ramené de Realtime database, sous forme de marqueurs « pins ». Les marqueurs seront interactifs activant un pop-up contenant le nom et prénom de l'utilisateur ayant réalisé le signalement en question.

Il y'a possibilité de dessiner un itinéraire de la localisation de l'utilisateur vers un signalement donné grâce à l'implémentation de Google Maps API.

#### 4.4. Page profile

##### 4.4.1. Android

Notre application aura un aspect de réseau sociale, c'est pour cela qu'un espace pour chaque utilisateur sera créé.

Les informations principales de l'utilisateur comme son nom, prénom, localisation.

Un bouton « Modifier profile » est présent. Ce dernier permet d'envoyer les utilisateurs vers une page où il pourra modifier ses informations tel que son nom, prénom ainsi que sa photo de profile et de confirmer ses changements en les sauvegardant au niveau de la base de données.

La barre de navigation sera présente pour que l'utilisateur puisse quitter son profile et aller vers d'autres pages

## Conclusion

Ce projet s'est révélé très enrichissant dans la mesure où il nous a permis d'explorer des domaines d'ingénierie encore inconnus pour nous jusqu'à lors, tel que le contact avec le client, la rédaction d'un cahier de charges, le développement mobile dans tous ses aspects et la mise en place d'un Back-end évolutif et scalable<sup>4</sup>.

Par ailleurs, nous avons essayés dans la mesure du possible et avec toute notre persévérance de répondre aux besoins stipulés dans le cahier de charges afin d'atteindre les objectifs qu'on s'est donnés dans le respect des conventions et des délais établis.

Nous aimerions aussi faire constater que la charge de travail que nous nous sommes imposé pour réussir à relever ce défi était considérable.

Pour faire aboutir les très hauts standards que nous avons en tête il nous a fallu beaucoup de courage et d'abnégation. Nous avons sous-estimé le projet tel qui nous a été confié en première instance, mais au fur et à mesure que nous avançons de nouvelles perspectives se dévoilait à nous, sans compter le souci du détail qui nous animé tout au long des 5 mois de travail. Nous avons dû changer à plusieurs reprises notre planification de départ, pour améliorer constamment notre modèle, chaque phase de test révélé de nouveaux défauts auxquels on n'y attendait pas, ce fût tout une dynamique à peaufiner et à améliorer pour espérer être à la hauteur des attentes de notre cliente. Une tâche pas toujours aisé contenu du fait de notre situation d'étudiant universitaire.

Une épreuve sans doute à relever, mais aussi un apprentissage très bénéfique pour notre futur carrière d'ingénieur.

---

<sup>4</sup> **Scalable** : évolutif