**UNIVERSITE CADI AYYAD**

**FACULTE DES SCIENCES ET TECHNIQUES MARRAKECH**



**Conception et réalisation d’une application mobile** **de gestion du trafic routier en temps réel de Marrakech lors du Marathon annuel**



**Présenté par :**

**Omar Elaibi**

**Encadré par :**

**Mme. Meriem Benhaddi**

**Filière : LST-MIASI**

**Année universitaire : 2015/2016**

**Soutenu en 2017**

Remerciements

Avant de présenter ce travail, Nous tenons, avec grand plaisir, à remercier tout d’abord Dieu le tout puissant de nous avoir permis d’arriver à ce niveau d’étude et encore pour ne donner la force le courage et la patience pour accomplir ce travail. « Dieu merci ».

Nous tenons à remercier encore nos chers parents pour leur soutien moral et matériel dans le but d’assurer notre réussite.

C’est avec grand plaisir que nous tenons à exprimer notre profonde gratitude et notre immense respect à notre encadrante Mme Meriem Benhaddi pour son accueil et ses précieux conseils, nous la remercions de nous avoir incités sans cesse pour améliorer notre travail et l’accomplir en un bon état.

Nous tenons à remercier aussi les membres du jury pour l’honneur qu’ils nous en accordé en acceptant de juger et évaluer notre travail.

Avec grand égard, nous ne manquerons pas d’exprimer notre grande reconnaissance à tous les enseignants et administrateurs de la Faculté des Sciences et Techniques de Marrakech.

Enfin, merci à tous ceux qui ont contribués dans l’élaboration de ce travail que ça soit du près ou de loin.

Table des matières

[1 Introduction 3](#_Toc453884359)

[2 Chapitre 1 : Représentation générale. 4](#_Toc453884360)

[2.1 Contexte général 4](#_Toc453884361)

[2.2 Objectif 4](#_Toc453884362)

[2.3 Analyse fonctionnelle 5](#_Toc453884363)

[2.3.1 Description des fonctionnalités 5](#_Toc453884364)

[2.3.2 L’aspect dynamique et temps réel 7](#_Toc453884365)

[3 Chapitre 2 : Etude conceptuelle. 8](#_Toc453884366)

[3.1 Introduction 8](#_Toc453884367)

[3.2 Démarche utilisée 8](#_Toc453884368)

[3.3 Etude conceptuelle 9](#_Toc453884369)

[3.3.1 Diagrammes UML 9](#_Toc453884370)

[3.3.2 Exigence et défis de l’interface homme-machine 15](#_Toc453884371)

[3.3.3 Utilisateur cible 15](#_Toc453884372)

[4 Chapitre 3 : Mise en œuvre. 17](#_Toc453884373)

[4.1 Introduction 17](#_Toc453884374)

[4.2 Environnement technique 17](#_Toc453884375)

[4.2.1 Environnement de développement 17](#_Toc453884376)

[4.2.2 Architecture cible 17](#_Toc453884377)

[4.2.3 Contraintes pour l’utilisateur 18](#_Toc453884378)

[4.3 Les outils techniques utilisés 19](#_Toc453884379)

[4.3.1 Android Studio 19](#_Toc453884380)

[4.3.2 Le kit de développement SDK Android 19](#_Toc453884381)

[4.3.3 Gradle Android 19](#_Toc453884382)

[4.3.4 PHP (Hypertext Preprocessor) 20](#_Toc453884383)

[4.3.5 MYSQL 20](#_Toc453884384)

[4.3.6 phpMyAdmin 20](#_Toc453884385)

[4.3.7 L’API Google Maps 21](#_Toc453884386)

[4.4 Présentation de l’application 22](#_Toc453884387)

[4.4.1 L’interface d’authentification 22](#_Toc453884388)

[4.4.2 L’interface d’administrateur 25](#_Toc453884389)

[4.4.3 L’interface d’usager routier 31](#_Toc453884390)

[4.4.4 Description du service Directions de Google maps API 35](#_Toc453884391)

[5 Conclusion 38](#_Toc453884392)

**Liste des Figures**

Figure 1: Illustration du problème et sa solution 7

Figure 2: dix principaux diagrammes UML 11

Figure 3: Diagramme de cas d'utilisation 12

Figure 4: Diagramme des classes 13

Figure 5: diagramme de séquence de cas « ajout à la base de données » 14

Figure 6: Diagramme de séquence de cas « suppression de la base de données » 15

Figure 7: Diagramme de séquence de cas « modification des entités dans la base de données » 15

Figure 8: Diagramme de séquence de cas « détection de la position » 16

Figure 9:Diagramme de séquence de cas « visualisation de tous les boulevards de Marrakech » 16

Figure 10: Diagramme de séquence de cas « affichage des routes circulantes ouvertes à proximité » 17

Figure 11: Diagramme de séquence de cas « obtention d'un itinéraire » 18

Figure 12: Une définition de l'interface homme-machine 19

Figure 13: tableau des défis et exigences IHM pour l'application 19

Figure 14: tableau des critères de satisfaction de l’utilisateur de l’application 20

Figure 15: tableau des critères du profil d'utilisateur cible 20

Figure 16: Légende des couleurs et signes utilisées dans l'application 21

Figure 17: Architecture cible de la solution 23

Figure 18: interface d’authentification 27

Figure 19: activité d’inscription 28

Figure 20: table « utilisateur » contenant deux utilisateurs 29

Figure 21: interface d’administration 30

Figure 22: activité « Ajout d’un boulevard/rue » 31

Figure 23: table boulevard\_rue contenant un échantillon de boulevards/rues. 31

Figure 24: activité «Suppression d’un boulevard/d’une rue» 32

Figure 25: activité «modification d’un boulevard/d’une rue» 33

Figure 26: activité « ouverture/fermeture d’un boulevard/d’une rue» 34

Figure 27: interface « tous les boulevards de la ville » 37

Figure 28: interface « les boulevards à proximité » 38

Figure 29: interface « obtenir un itinéraire » 39

# Introduction

De nos jours, l’usage des téléphones portables a connu une croissance phénoménale, surtout après l’invention des Smartphones. Ces derniers, et grâce à leur développement rapide et continu, offrent plusieurs opportunités de communication poussée, d’apprentissage et de divertissement. On peut dire qu’ils sont devenus indispensable pour la majorité des gens.

Comme cité avant, le développement des Smartphones est très rapide, c’est grâce à l’apparition quotidienne des nouvelles applications avec des nouvelles fonctionnalités, grâce à la mise à jour de celles déjà existantes et encore grâce à la mise à jour de leurs systèmes d’exploitation.

Ainsi, dans le cadre de ce projet de fin d’études, nous avons comme objectif de concevoir et réaliser une nouvelle application mobile qui a pour but la gestion du trafic routier en temps réel de Marrakech lors du Marathon annuel. Elle permet pour un utilisateur en véhicule de se localiser dans la ville, puis d’obtenir une liste de tous les boulevards et rues circulants à proximité de sa position.

Ce rapport présente l’ensemble des étapes suivies pour développer et réaliser l’application, il contient trois chapitres organisés comme suit :

Le premier chapitre intitulé « Représentation générale » est consacré à la présentation du contexte général du travail ainsi que l’objectif principal de l’application.

Le chapitre « Etude conceptuelle » détaille les différents aspects conceptuels de l’application, ainsi que les exigences et défis liés aux interfaces homme-machine.

Le chapitre suivant et dernier intitulé « Mise en œuvre » présente l’environnement du travail ainsi que les outils logiciels utilisés pour la réalisation du projet, il illustre aussi le travail réalisé par un ensemble des interfaces graphiques conçues pour l’application.

En conclusion, nous mentionnons les bénéfices du travail réalisé ainsi que les difficultés confrontées lors de la réalisation de l’application.

# Chapitre 1 : Représentation générale.

## Contexte général

Chaque année, la ville de Marrakech organise un Marathon international pendant deux jours du mois janvier. Durant cet évènement sportif et aussi touristique, il sera interdit de circuler et de stationner sur plusieurs boulevards et rues à cause du passage des coureurs par ces derniers.

Ainsi, les utilisateurs des véhicules ont besoin de savoir les chemins circulants les plus courts qui sont disponibles et qui leur permettent d’arriver à leurs destinations.

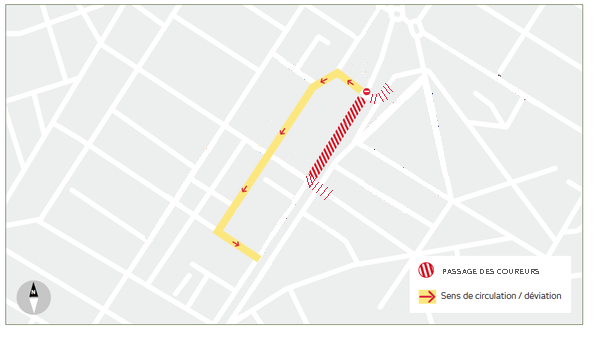


Figure : Illustration du problème et sa solution

## Objectif

L’objectif principal de cette application consiste à implémenter une solution mobile pour gérer le trafic routier pendant le marathon annuel de Marrakech. Cette solution sera basée sur une base de données contenant une liste de tous les chemins disponibles pour la circulation pendant le marathon.

Ainsi, l’application permet dans un premier temps de détecter la position d’utilisateur en temps réel, puis de montrer tous les chemins circulants disponibles en se basant sur la liste de la base de données. Elle pourra aussi proposer un itinéraire pour une destination choisie par l’utilisateur.

## Analyse fonctionnelle

### Description des fonctionnalités

#### Authentification

Chaque utilisateur que ce soit administrateur ou bien simple usager doit s’authentifier pour accéder aux fonctionnalités de l’application.

Ainsi, l’accès à différentes fonctionnalités de l’application sera déterminé selon le type de l’utilisateur (Administrateur/usager routier).

* **Cas où l’utilisateur est un usager routier**

Un usager routier pourra accéder à certaines fonctionnalités que nous citons ci-après :

#### Détection de la position

L’utilisateur va utiliser cette fonctionnalité pour détecter sa position dans la carte de la ville de Marrakech. Pour ceci il aura besoin d’activer les services de localisation sur son appareil.

#### Visualisation de tous les boulevards circulants de la ville

Avec cette fonctionnalité, l’utilisateur pourra visualiser tous les boulevards circulants de la ville de Marrakech qu’ils soient ouverts ou fermés (toute la carte des boulevards de Marrakech).

#### Affichage des chemins circulants disponibles à proximité

L’utilisateur pourra voir tous les boulevards et rues circulants ouverts à proximité de sa position. Ceci sera basé sur sa position détectée et sur une liste de tous les chemins circulants disponibles.

Ainsi les routes fermées à cause du marathon seront distinguées par une couleur différente.

#### Obtenir un itinéraire

L’utilisateur pourra choisir une destination, et l’application va proposer un itinéraire avec sa position comme point de départ.

* **Cas où l’utilisateur est un administrateur**

Un administrateur est responsable de la gestion de tous ce qui concerne le fonctionnement de l’application et les mises à jour. Ainsi, il aura accès à des fonctionnalités telles que celles qui suivent :

#### Ajout des boulevards et rues à la base de données

Un administrateur sera garant d’ajouter des nouveaux boulevards et rues à la base de données.

#### Suppression des boulevards et rues de la base de données

Un administrateur pourra aussi supprimer des rues et des boulevards de la base de données.

#### Modification des boulevards et rues de la base de données

Un administrateur pourra modifier les attributs des entités de la base de données (les boulevards et les rues). Ainsi, il pourra changer leurs noms, leurs coordonnées …

#### Ouverture/Fermeture des boulevards et rues de la base de données

Un administrateur pourra visualiser une liste de tous les boulevards et rues de la base de données et cocher ceux qu’il veut fermer.

#### Visualisation de la carte (Test)

Un administrateur pourra tester si ses mises à jour sont bien passées en accédant à l’activité de la carte qui offre toutes les fonctionnalités prévues pur un simple usager.

### L’aspect dynamique et temps réel

### 

Vu qu’un utilisateur de l’application bouge continument, sa position change rapidement. D’autre part, les chemins circulants changent durant les deux jours du marathon. Ainsi l’ensemble des fonctionnalités de l’application exige un traitement en temps réel. Ceci signifie qu’elles doivent s’exécuter suffisamment vite par rapport à la dynamique de l’utilisateur (contrainte temporelle), aussi elles doivent s’actualiser à chaque fois que les boulevards et rues circulants changent, et accorder une grande importance à l’exactitude des réponses.

# Chapitre 2 : Etude conceptuelle.

## Introduction

Ce chapitre est dédié à une étape très importante dans la réalisation de l’application mobile, il s’agit de l’étude conceptuelle qui permet de mettre en place un système d’information, capable à répondre aux objectifs souhaités, et de bien spécifier les fonctionnalités de ce système.

Ainsi, nous abordons dans cette partie la modélisation conceptuelle du système de l’application en utilisant le langage de modélisation UML.

## Démarche utilisée

Pour concevoir notre application, nous avons opté pour le choix du formalisme UML (Unified Modeling Language) qui permet d’élaborer des modèles à l’aide des diagrammes qui offrent une vue complète des aspects statiques et dynamiques du système de l’application et qui montrent les différentes interactions entre ce dernier et l’environnement extérieur.

UML comporte dix principaux types de diagrammes qui seront présentés dans la diapositive ci-dessous. Cependant dans ce chapitre nous considérons seulement trois types de diagrammes : diagramme de cas d’utilisation, diagramme de classes et diagramme de séquence qui sont assez suffisants pour concevoir notre application.

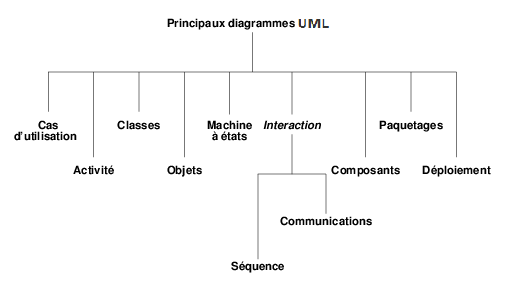


Figure : dix principaux diagrammes UML

## Etude conceptuelle

### Diagrammes UML

#### Modèle statique

##### Diagramme de cas d’utilisation

Le diagramme de cas d’utilisation nous permet de savoir précisément à quoi l’application devra servir, c’est-à-dire à quels besoins elle devra répondre.

Les deux concepts de ce diagramme sont : l’acteur et le cas d’utilisation qui modélise un service rendu par l’utilisateur. Ainsi, la détermination des besoins est basée sur la représentation des interactions entre l’acteur et le système de l’application.

La figure suivante représente le diagramme de cas d’utilisation de notre application :

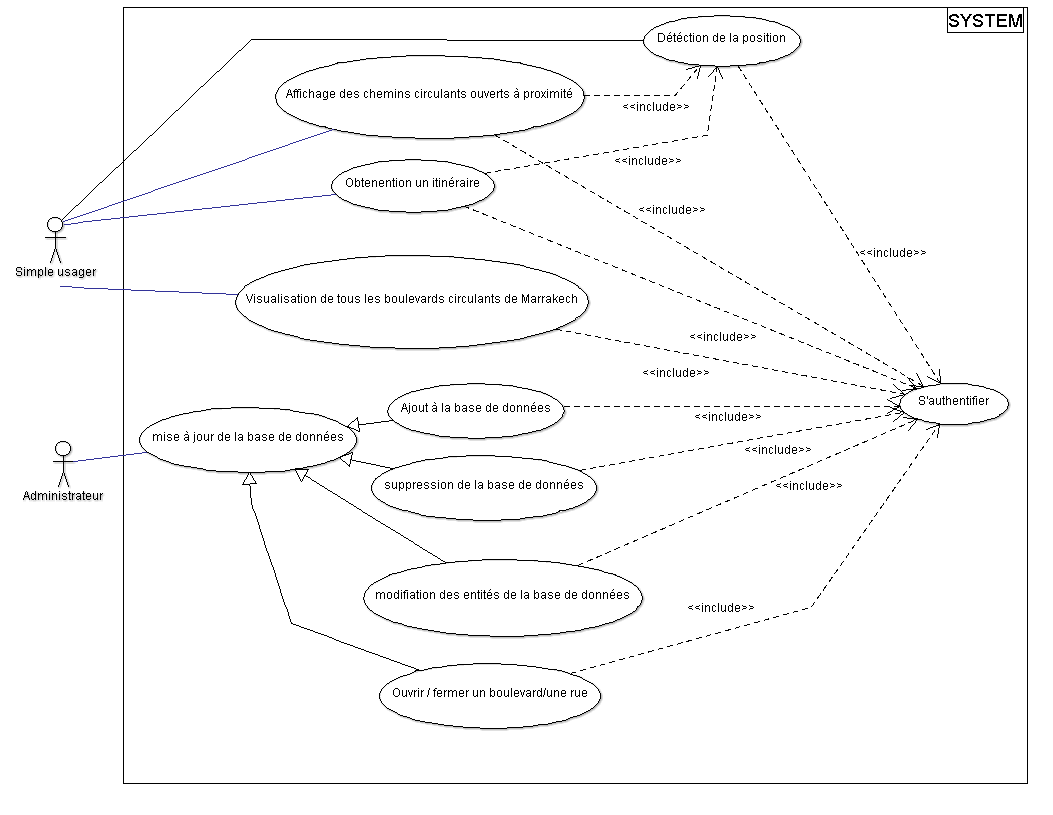


Figure : Diagramme de cas d'utilisation

##### Diagramme des classes

Le diagramme des classes nous permet de spécifier la structure et les liens entre les objets dont le système est composé.

Les différentes classes de notre système sont énumérées dans le tableau suivant :

|  |  |
| --- | --- |
| **Classe** | **Description** |
| Utilisateur | Désigner un personne utilisateur du système, il est caractérisé par son id (identificateur), son nom, son prénom, son pseudo, et son mot\_de\_passe. |
| Administrateur | Désigner un administrateur de l’application qui est un cas particulier de la classe Utilisateur. |
| Usager\_routier | Désigner un usager routier qui est un cas particulier de la classe Utilisateur. |
| Boulevard\_Rue | Désigner un boulevard ou une rue dans la ville de Marrakech, il est caractérisé par son numéro et son nom. |
| Route\_fermee | Désigner une route fermée, il est un cas particulier de la classe Boulevard\_Rue. |
| Itineraire | Désigner un itinéraire sollicité par un usager routier. |
| Map | Désigner la carte où les boulevards, les itinéraires et les positions vont être dessinés. |

**Tableau 1 : Tableau des classes du système**

Ci-dessous le diagramme des classes :

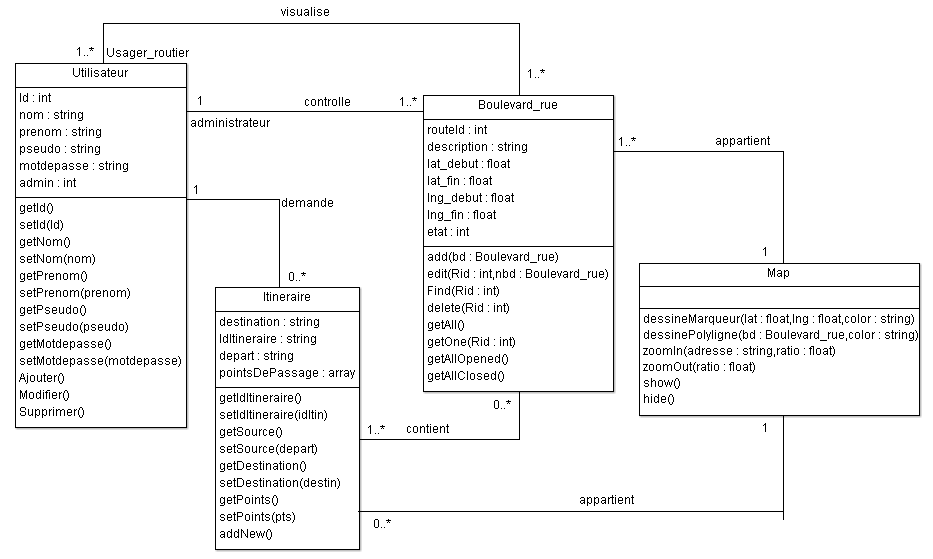


Figure : Diagramme des classes

#### Modèle dynamique

##### Diagramme de séquence

Le diagramme de séquence nous permet de décrire comment les éléments du système interagissent entre eux et avec les acteurs selon un ordre chronologique.

Plus précisément, il nous décrit, pour chaque cas d’utilisation, comment se déroule les actions entre acteurs et objets.

Ci-dessous le diagramme de séquence correspondant au diagramme de cas d’utilisation de notre système :

###### Cas 1 : Ajout à la base de données

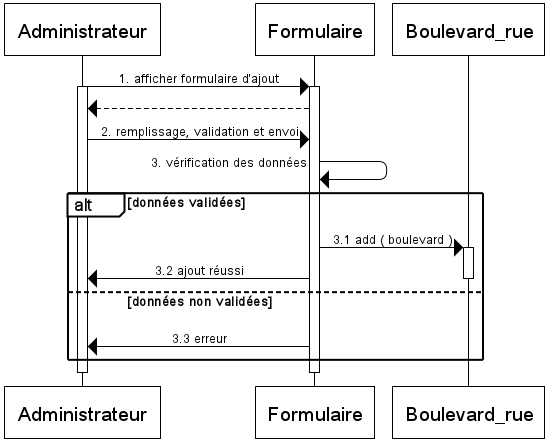


Figure : diagramme de séquence de cas « ajout à la base de données »

###### Cas 2 : Suppression de la base de données

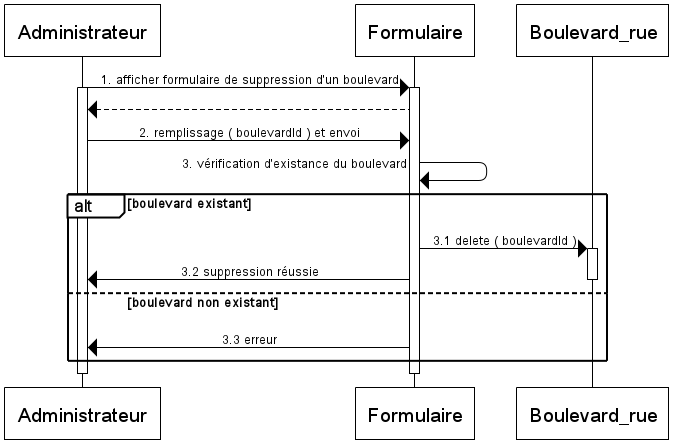


Figure : Diagramme de séquence de cas « suppression de la base de données »

###### Cas 3 : Modification des entités de la base de données

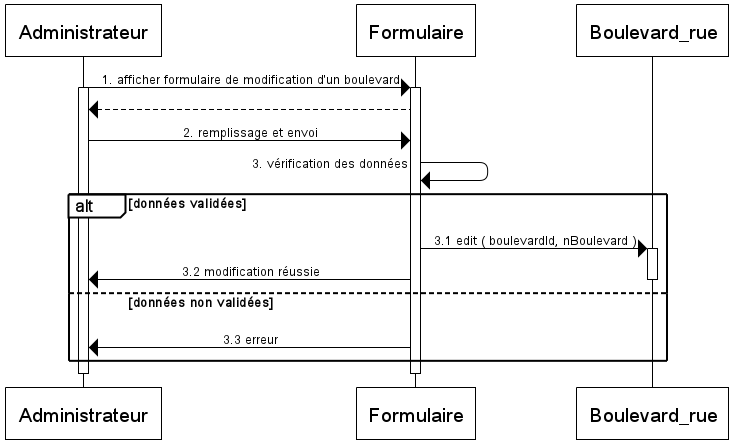


Figure : Diagramme de séquence de cas « modification des entités dans la base de données »

###### Cas 4 : détection de la position

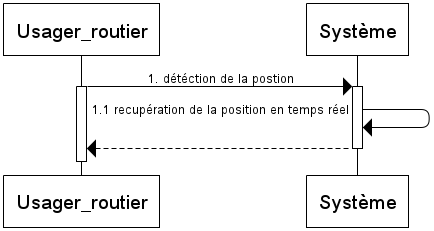


Figure : Diagramme de séquence de cas « détection de la position »

###### Cas 5 : visualisation de tous les boulevards de Marrakech

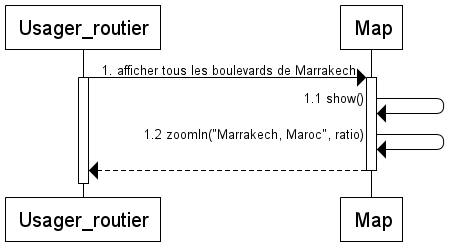


Figure :Diagramme de séquence de cas « visualisation de tous les boulevards de Marrakech »

###### Cas 6: Affichage des routes circulants à proximité

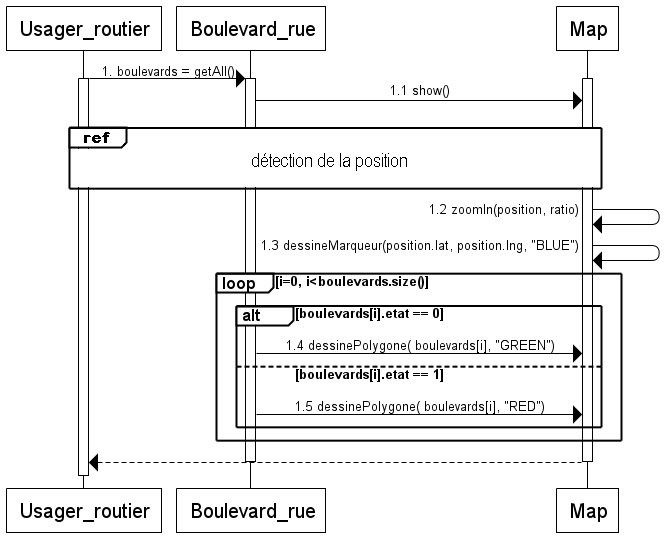


Figure : Diagramme de séquence de cas « affichage des routes circulantes ouvertes à proximité »

La valeur de ‘’position’’ est obtenue dans le cas d’utilisation « détection de la position ».

###### Cas 7 : obtention d’un itinéraire

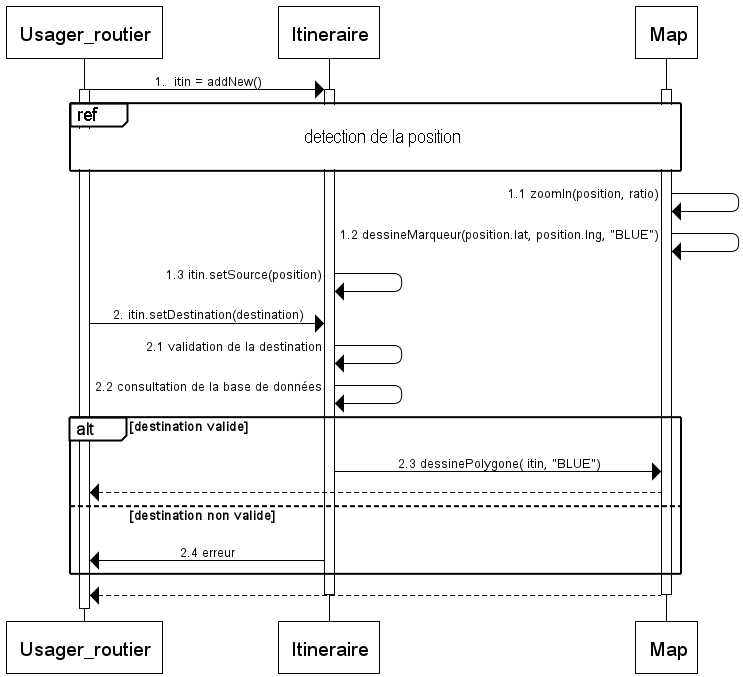


Figure : Diagramme de séquence de cas « obtention d'un itinéraire »

La valeur de ‘’position’’ est obtenue dans le cas d’utilisation « détection de la position ».

### Exigence et défis de l’interface homme-machine

L’interface Homme-Machine (IHM) est l’ensemble des dispositifs matériels et logiciels permettant à un utilisateur d’interagir avec un système interactif.

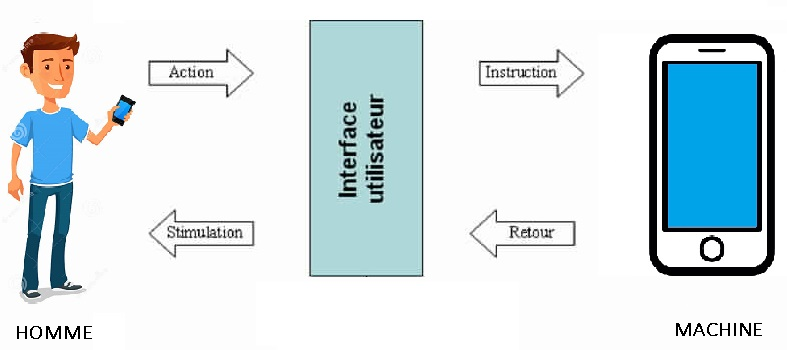


Figure : Une définition de l'interface homme-machine

La conception des IHM est une partie très importante dans la conception des applications, en effet elle peut représenter jusqu’à 80% dans l’étude conceptuelle. Elle est indépendante du cœur du système, cependant les deux doivent être pensés parallèlement.

Dans le cadre de ce projet et du point de vue IHM, les défis sont liés d’une part à l’adaptation au contexte d’usage, aux différents critères des utilisateurs (niveau de formation, les compétences et les expériences sur le domaine…) et d’une autre part à la satisfaction des exigences fonctionnelles et opérationnelles des utilisateurs. Ainsi, nous détaillerons, dans ce qui suit, quelques critères des deux parts.

|  |  |  |  |
| --- | --- | --- | --- |
| Critère/ défi | | Exigences | |
| **exigence** | **description** |
| Contexte d’usage  *<utilisateur, plate-forme>* | Utilisateur | Différences physique | * Age * Handicap |
| Connaissances et expériences | * dans le domaine de la tache ( novice, expert… ) * en informatique, sur système manipulation occasionnelle/habituelle) |
| Caractéristiques  culturelles | * langue * signification des couleurs |
| Plate-forme | Structure matérielle | * machine utilisée (PC, smartphone…) * taille d’écran (application responsive ou non) * dispositifs d’interaction (tactile, souris et clavier…) |

Figure : tableau des défis et exigences IHM pour l'application

|  |  |
| --- | --- |
| Critère de satisfaction | Description |
| Fiabilité | L’aptitude de l’application à rendre des résultats juste pour chaque requête ou traitement dans un temps limité et rapide peu importe les conditions d’exploitation. |
| Efficacité | la capacité de réaliser les objectifs fixés et d’atteindre le résultat prévu. |
| Exactitude/Précision | La capacité des fonctionnalités à répondre aux besoins des utilisateurs avec précision. |
| Convivialité | La qualité de minimiser l’effort d’utilisateur pour apprendre et manipuler l’application en lui offrant un usage intuitif ou en lui dirigeant suffisamment. |
| Flexibilité | La capacité de minimisation du coût de modification et d’actualisation de l’application |
| Testabilité | La facilité et la minimisation du coût du test ses méthodes et du traitement des requêtes dans l’application |

Figure : tableau des critères de satisfaction de l’utilisateur de l’application

Ainsi, Nous allons discuter dans ce qui suit le type des utilisateurs ciblés afin de réaliser des IHM qui facilitent l’utilisation de l’application.

### Utilisateur cible

Un utilisateur de l’application est imaginé comme mobile, évoluant dans un monde dynamique, souvent imprévisible.

L’application sera destinée à des utilisateurs motorisés qui circulent dans la ville de Marrakech et qui disposent d’un système Android (smartphone, tablette, …). Citons ci-dessous dans un tableau d’autres critères du profil et de la catégorie d’utilisateur cible.

|  |  |  |
| --- | --- | --- |
| Critère | | Description |
| Différences physiques | Age | Il est préférable que l’utilisateur soit âgé de plus de 7 ans |
| Handicap | Handicap moteur, [Cécité](https://fr.wiktionary.org/wiki/c%C3%A9cit%C3%A9). |
| Compétences techniques | Dans le domaine de la tache | Même un novice peut utiliser l’application. |
| En informatique (sur le système) | L’utilisateur doit savoir manipuler un smartphone Android |
| Caractéristiques socio-culturelle | langue | L’application utilise la langue française |

Figure : tableau des critères du profil d'utilisateur cible

Pour faciliter la compréhension et l’utilisation aux utilisateurs cibles, les interfaces de l’application vont utiliser :

* Des icônes habituelles (souvent utilisées dans Google Maps)
* Des couleurs formelles (rouge pour indiquer la fermeture, vert pour indiquer l’ouverture et bleu pour indiquer un itinéraire)

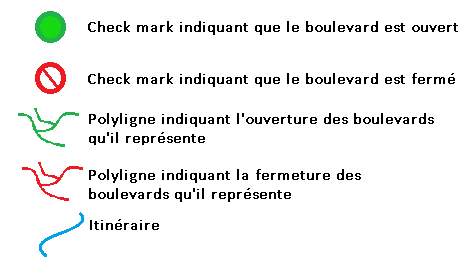


Figure : Légende des couleurs et signes utilisées dans l'application

* Des textes et des indications d’explication
* …

# Chapitre 3 : Mise en œuvre.

## Introduction

Après avoir terminé la phase d’étude conceptuelle nous arrivons dans ce chapitre à réaliser la mise en œuvre du système, l’implémentation de l’application mobile nécessite plusieurs outils techniques et langages qui servent à répondre à nos requêtes notamment l’analyse en temps réel des grandes données.

## Environnement technique

### Environnement de développement

- La réalisation d’une application mobile nécessite un ordinateur ayant des caractéristiques spécifiques avec des softwares particuliers (Eclipse, JDK 7/8, …).

- Le délai de la réalisation du projet ne pourra pas dépasser 2 mois.

### Architecture cible

Le schéma suivant illustre l’architecture cible de la solution, elle se compose de trois parties :

* L’application mobile sur Android.
* Le serveur web de l’application.
* Le serveur de la base de données.

****

Figure : Architecture cible de la solution

Pour avoir accès aux fonctionnalités de l’application mobile sur Android, l’utilisateur aura besoin de communiquer, via une connexion à Internet, avec le serveur d’applications qui offre un contexte d’exécution pour ces fonctionnalités et qui a un accès direct à la base de données de l’application.

### Contraintes pour l’utilisateur :

L’utilisation de l’application exige :

- Une connexion à Internet.

- Avoir un compte d’utilisateur.

## Les outils techniques utilisés

### Android Studio



C’est un environnement pour développement et programmation qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l’IDE (Integrated developement environnement) le plus utilisé.

### Le kit de développement SDK Android



C’est un ensemble d'outils que met à disposition Google afin de vous permettre de développer des applications pour Android.

### Gradle Android



Gradle est le digne successeur de Maven et de Ant, alliant ces deux outils afin de créer une plateforme de production Java simple à utiliser, et bien adaptée pour les projets Android.

Gradle est intégré à Android Studio et est utilisé afin de gérer et construire les projets Android (en utilisant le langage Groovy).

Il permet entre-autre de gérer la construction d’un projet utilisant plusieurs modules et dépendances de librairies Maven, et ce, de façon très simple.

### PHP (Hypertext Preprocessor)



C’est un langage de scripts généraliste et Open Source (libre), spécialement conçu pour le développement des applications web.

### MYSQL



C’est système de gestion de bases de données relationnelles (SGBDR) Open Source (libre), gratuit et multi-utilisateurs. C’est l’outil le plus utilisé par les développeurs dans le monde.

### phpMyAdmin



C’est une application Web développée en PHP sous licence libre, offrant une interface pour la gestion des bases de données MySQL. Il est ainsi possible, grâce à un simple navigateur Internet de procéder à des opérations permettant de modifier / supprimer des enregistrements, sauvegarder sa base…

### L’API Google Maps



C’est une bibliothèque permettant l'intégration sur une page Web ou une application web du service Google Maps. C'est un service de cartographie en perpétuelle évolution. Il permet notamment de zoomer sur une carte jusqu'à obtenir un rendu visuel des habitations, des rues, etc.

Avec l'API, vous pouvez alors intégrer à votre page Web une carte et utiliser ses fonctionnalités de zoom ou de marker (position d'un endroit précis).

## Présentation de l’application

Nous présentons dans cette section quelques interfaces principales de notre application.

### L’interface d’authentification

Lors de lancement de l’application, une interface d’authentification apparait invitant l’utilisateur à s’authentifier en saisissant ses paramètres d’accès (Identifiant et mot de passe).

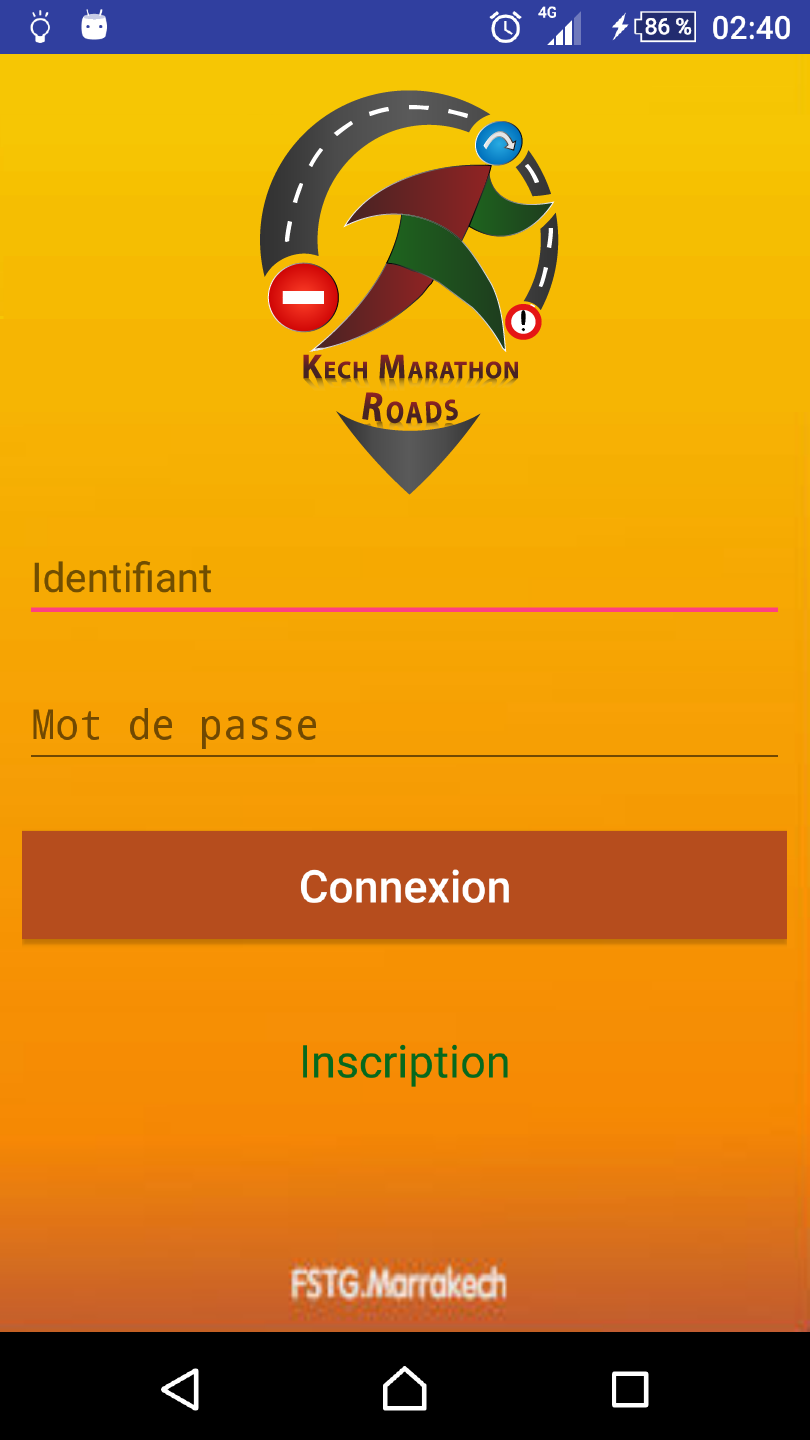


Figure : interface d’authentification

Lorsque le client ne possède pas un compte, il pourrait créer un en accédant à la fonctionnalité Inscription par appui sur le Link « Inscription ».

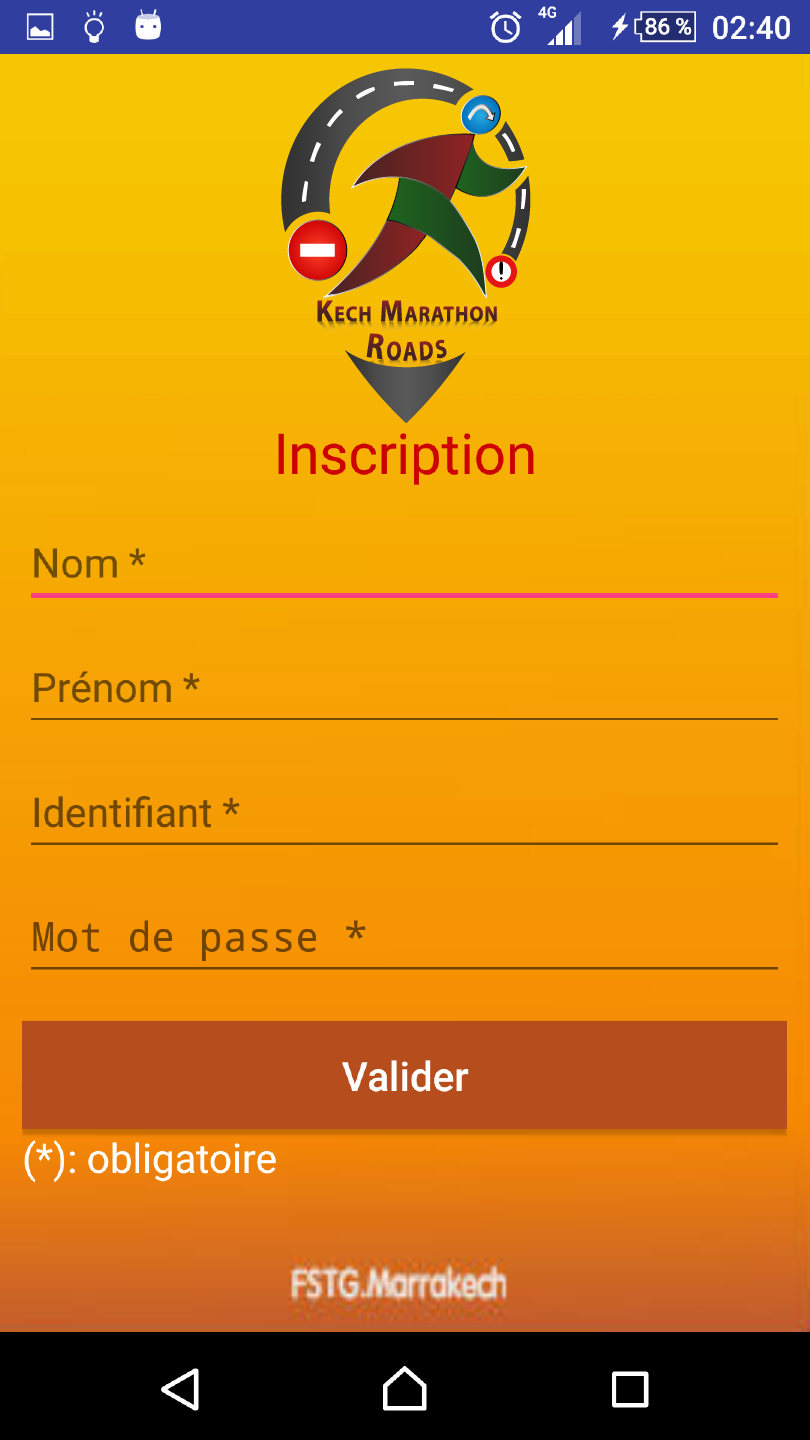


Figure : activité d’inscription

Les informations des utilisateurs sont enregistrées dans la base de données de l’application dans une table nommée « utilisateur ».

L’illustration suivante montre cette table contenant deux utilisateurs dont un est un administrateur et l’autre est un simple usager

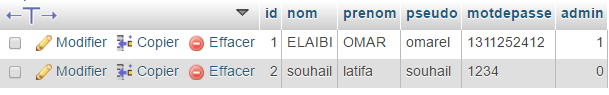


Figure : table « utilisateur » contenant deux utilisateurs

Dans le cas où l’authentification est validée, l’interface qui s’ouvre dépend du type de l’utilisateur (Administrateur / usager routier).

* Si l’utilisateur est un administrateur :

### L’interface d’administrateur

L’interface de l’administrateur est constituée d’un menu de boutons, chacun transmit à une autre activité qui réalise une fonctionnalité parmi celles que seul l’administrateur pourra y accéder.



Figure : interface d’administration

#### Ajouter un boulevard/une rue

Une fois l’administrateur appuie sur le bouton « Ajouter un boulevard/une rue », un formulaire s’affiche pour saisir les coordonnées et la description du boulevard/rue qu’il veut ajouter, puis il doit appuyer sur valider pour l’ajouter.

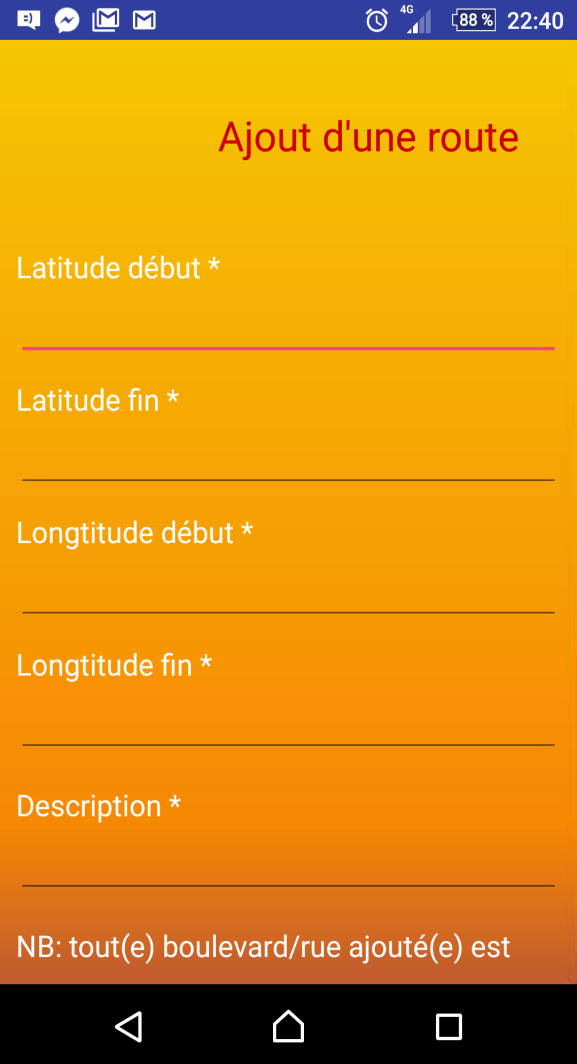
 

Figure : activité « Ajout d’un boulevard/rue »

Chaque boulevard/rue est stocké dans la base de données de l’application dans la table boulevard\_rue par un point de départ et un point d’arrivée, une description et un état, chaque point est défini par une latitude et longitude.

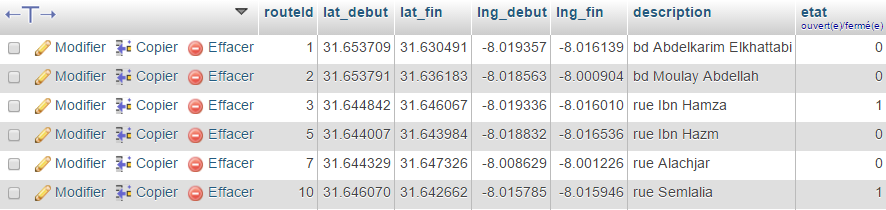


Figure : table boulevard\_rue contenant un échantillon de boulevards/rues.

#### Supprimer un boulevard/une rue

En appuyant sur le bouton « Supprimer un boulevard/une rue » l’administrateur sera invité à saisir l’Id de la route qu’il veut supprimer, puis d’appuyer sur le bouton « supprimer »

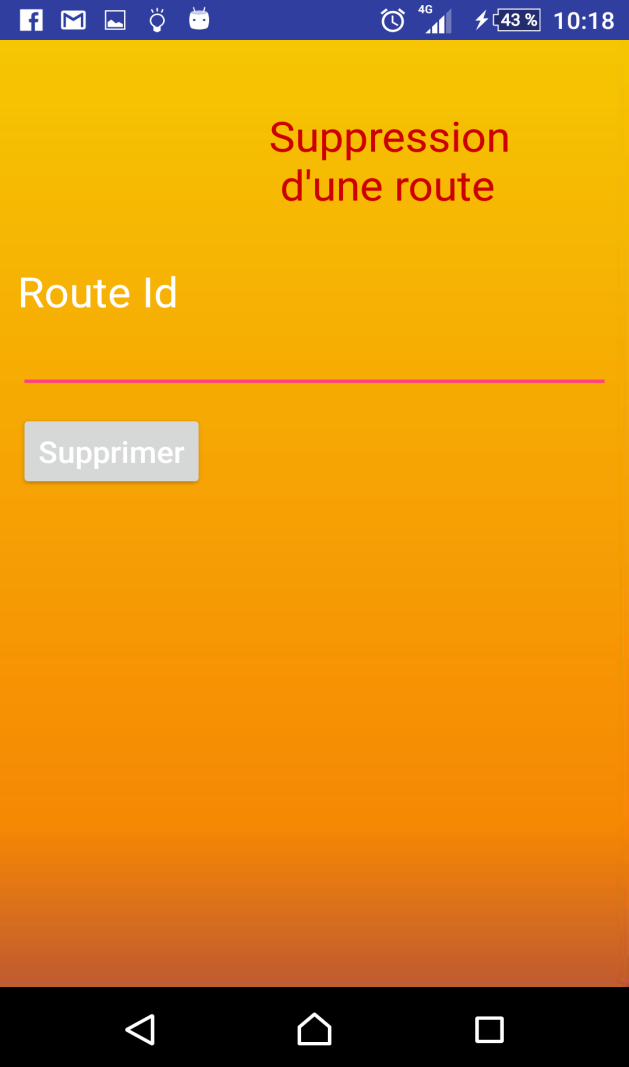


Figure : activité «Suppression d’un boulevard/d’une rue»

#### Modifier un boulevard/une rue

Pour modifier les coordonnées et la description d’un Boulevard /d’une rue l’administrateur doit appuyer sur le bouton « modifier un boulevard/une rue ».

Un formulaire s’affiche invitant l’administrateur de saisir l’Id de la route à modifier, les nouvelles coordonnées et la nouvelle description. Juste après, il doit appuyer sur le bouton « valider » pour l’application des modifications.

Figure : activité «modification d’un boulevard/d’une rue»

#### Ouvrir/fermer un boulevard/une rue

Quand l’administrateur tape le bouton « Ouvrir/ Fermer un boulevard/ une rue », une liste s’affiche contenant tous les boulevards/rues stockés dans la base de données dans la table boulevard\_rue.

Chaque boulevard/rue cité dans cette liste à un « Check Mark », l’administrateur doit cocher/décocher ceux qu’il veut fermer/ouvrir.

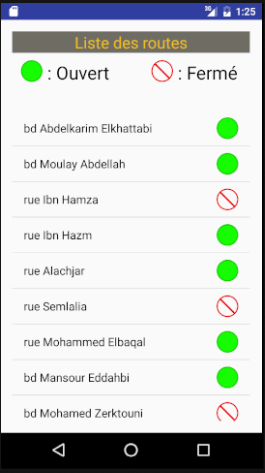


Figure : activité « ouverture/fermeture d’un boulevard/d’une rue»

#### Visualisation de la carte

Cette fonctionnalité est faite pour que l’administrateur puisse tester l’affichage des données qu’il gère dans la carte.

Quand il appuie sur le bouton « visualisation du map », l’activité du map s’ouvre pour qu’il puisse tester les fonctionnalités d’usager routier.

* Si l’utilisateur est un usager routier :

### L’interface d’usager routier

L’interface d’usager routier affiche la carte de la ville de Marrakech incluant un bouton pour aller à la localisation courante de l’utilisateur, elle se constitue aussi d’un menu des boutons (options/fonctionnalités).

Tous les boulevards et rues de la ville de Marrakech sont enregistrées dans une base de données, chaque boulevard/rue est défini par un point de début, un point d’arrivé, une description et un état (ouvert/fermé).

Dans notre application, on prend juste un échantillon de n boulevards stockés dans la base de données dont m sont fermés.

#### Visualisation de tous les boulevards circulants de la ville

Si l’utilisateur appuie sur le bouton « tous les boulevards », chaque boulevard stocké dans la base de données sera affiché dans la carte par une couleur verte et ceux qui sont fermés seront distingués par une couleur rouge.



Figure : interface « tous les boulevards de la ville »

#### Visualisation des boulevards circulants ouverts à proximité

Si l’utilisateur appuie sur le bouton « les boulevards à proximité », seuls les boulevards stockés dans la base de données qui se trouve à proximité de sa position courante seront affichés par les mêmes couleurs de la partie précédente (vert pour les ouverts et rouge pour les fermés).

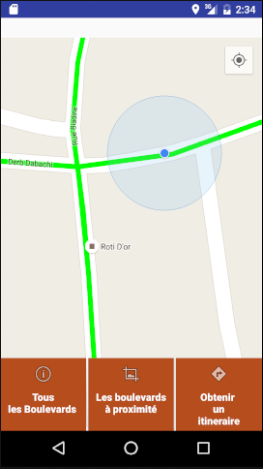


Figure : interface « les boulevards à proximité »

#### Obtention d’un itinéraire.

Si l’utilisateur appuie sur le bouton «obtenir un itinéraire », il sera invité à choisir une destination en appuyant sur la carte, et un chemin sera affiché en vert montrant un itinéraire qui relie la position courante de l’utilisateur comme position de départ et la destination choisie comme point d’arrivé.

Ainsi, un texte sera affiché en haut de l’écran indiquant la distance estimée de l’itinéraire, le temps estimé pour la parcourir en voiture et le temps estimé pour la parcourir à pieds.

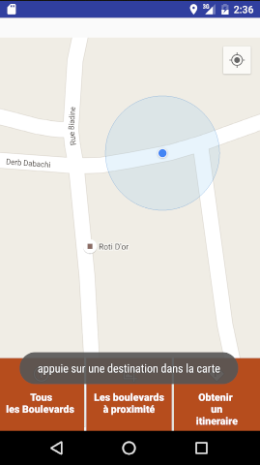
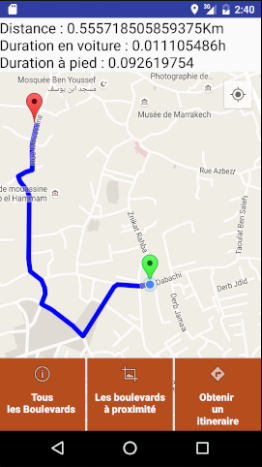
 

Figure : interface « obtenir un itinéraire »

### Description du service Directions de Google maps API

La réalisation des trois dernières fonctionnalités s’est basé sur un service très utile de Google Maps API, c’est le service de Directions.

C’est un service qui sert à calculer des itinéraires  (en utilisant divers moyens de transport), il communique avec le service Directions de Google Maps API, qui reçoit les demandes d'itinéraire et renvoie le résultat des calculs. Vous pouvez utiliser ces résultats d'itinéraire vous-même ou utiliser l'objet DirectionsRenderer pour en effectuer le rendu.

Lorsque vous indiquez le point de départ ou la destination dans une requête d'itinéraire, vous pouvez spécifier une chaîne de requête, une valeur LatLng ou un objet google.maps.Place.

Le service Directions peut renvoyer des itinéraires multi-segments passant par une série de points de cheminement. L'itinéraire est affiché sous forme de polyligne qui représente le trajet sur une carte.

Voici le code affichant l’utilisation de ce service :

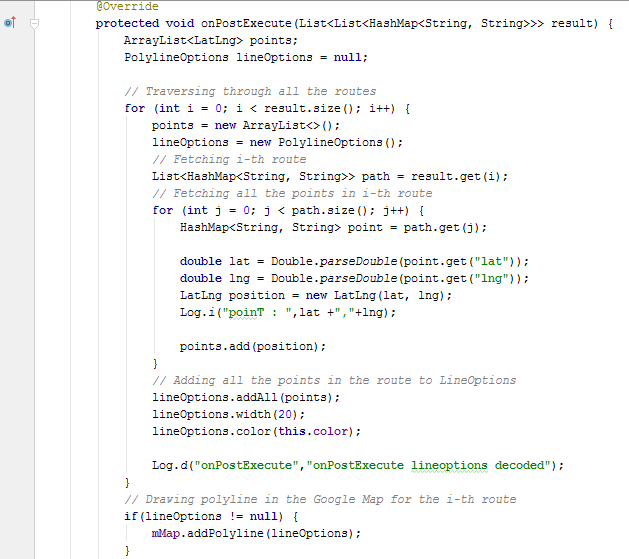
Ces deux méthodes sont pour avoir l’url du web service Directions.



Après l’obtention du l’url du service, on l’utilise dans la classe ci-dessus pour apporter les données qui nous servent à dessiner les polylignes voulues.



Juste après, on utilise une autre classe pour analyser les données rendues par la classe précédentes et dessiner les polylignes à partir d’elles. Voici la partie de la classe qui ait le dessin des polylignes sur la carte :



# Conclusion

Dans ce projet de fin d’étude, Notre mission a été de mettre en place une application de gestion du trafic routier en temps réel de Marrakech lors du marathon international annuel, le défi était alors de faire une recherche générale sur la conception et la réalisation de ce type d’applications, d’approfondir nos connaissances à propos des techniques et langages dont on a besoin pour la réalisation et d’utiliser tous ce qui précède pour accomplir notre mission.

Au fil de ce rapport, nous avons vu plusieurs aspects, nous avons commencé par une représentation générale de la problématique et de l’objectif ciblé. Puis nous avons décrit les fonctionnalités de notre application en les divisant en deux catégories : celles destinées aux administrateurs et celle destinées aux usagers routiers. Ensuite nous nous sommes concentrés sur l’aspect conceptuel et implémentation.

Ce travail nous a, non seulement, donné l’opportunité de manipuler de nouvelles technologies et langages de programmation, mais il nous a aussi permis d’approfondir nos connaissances en analyse et conception des systèmes et d’exploiter le monde de la programmation Android.

Lors de la réalisation de notre projet, nous avons été confrontés à plusieurs difficultés, parmi eux, l’utilisation d’un émulateur qui ne supporte pas la détection automatique de la position et qui demande de longue durée pour qu’il puisse exécuter l’application, ainsi la création d’un serveur web pour contenir la base de données de l’application et les PHP scripts qui lient l’application avec la base de données.

La difficulté majeure rencontrée dans la réalisation de cette application était la réalisation de la fonctionnalité « les boulevards à proximité », l’API Google maps n’offre pas un service qui peut détecter des lignes ou des poly lignes à proximité d’une position, cependant il offre un service de détection des points à proximité, ceci m’a obligé de faire plusieurs tests sur des points d’un boulevard afin de confirmer qu’il se trouve à proximité.

Webographie

1. [openclassrooms.com](http://www.openclassrooms.com)
2. [maps.google.com](http://www.maps.google.com)
3. [stackoverflow.com](http://www.stackoverflow.com)
4. [developers.google.com](http://www.developers.google.com)
5. [joshmorony.com](http://www.joshmorony.com)
6. [tutorielandroid.francoiscolin.fr](http://www.tutorielandroid.francoiscolin.fr)
7. [phpdebutant.org](http://www.phpdebutant.org)
8. [androidtutorialpoint.com](http://www.androidtutorialpoint.com)
9. [developpez.net](http://www.developpez.net)
10. [sitepoint.com](http://www.sitepoint.com)
11. [androidhive.info](http://www.androidhive.info)
12. La chaine YouTube de Prabeesh K [youtube.com/channel/UCfQkNueQenRQQ1NnCBe6eQQ](https://www.youtube.com/channel/UCfQkNueQenRQQ1NnCBe6eQQ)