

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

**FACULTE DE MATHEMATIQUE ET DE
L'INFORMATIQUE**

DEPARTEMENT D'INFORMATIQUE

N° :



**DOMAINE : MATHEMATIQUE ET
INFORMATIQUE**

FILIERE : INFORMATIQUE

OPTION : RTIC

**Mémoire présenté pour l'obtention
Du diplôme de Master Académique**

Par : ATTOUI Seyf El-islam

Intitulé

**Application Android pour les funérailles de la
ville de M'Sila**

Soutenu devant le jury composé de :

MOKHTARI Rabah

Université de M'sila

Président

BENAZI Makhlouf

Université de M'sila

Rapporteur

KAMEL Mohamed

Université de M'sila

Examineur

Année universitaire : 2018 /2019



❁❁❁❁❁ Dédicaces *❁❁❁❁❁*

*A mon grand-père Ahmed, « **بر حبيبتنا الله** ».*

A ma mère.

*A ma famille, mes chers proches et à ce qui me donnent de
l'amour et de la vivacité.*

A tous mes amis.

A tous ceux qui m'ont soutenu tout au long de ce projet.

Attoui Seyf El-Islam



Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon projet et qui m'ont aidée lors de la rédaction de ce mémoire.

Je voudrais dans un premier temps remercier, mon encadrant M.BENAZI Makhloof, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.

Je remercie également toute l'équipe pédagogique et les enseignants de la département d'Informatique.

Enfin, j' adresse mes remerciements aux membres du jury pour m'avoir honoré en acceptant d'évaluer ce travail.

Table des matières

Table des matières.....	iv
Table des figures.....	vi
Liste des tableaux.....	vii
Introduction générale.....	1
Chapitre 1 Généralités sur les applications mobiles sous Android.....	3
1. Introduction.....	3
2. Applications mobiles.....	3
2.1. Définition.....	3
2.2. Systèmes d'exploitations mobiles.....	3
2.3. Pourquoi Android.....	4
3. Présentation du système Android.....	5
3.1. Définition.....	5
3.2. Les versions d'Android.....	5
3.3. L'architecture d'Android.....	10
3.4. Présentation du SDK.....	12
3.5. Composants d'une application Android.....	13
3.6. Cycle de vie d'une application Android.....	16
4. Présentation du sujet.....	18
4.1. Problématique.....	18
4.2. Solution par une application Android.....	19
5. Conclusion.....	19
Chapitre 2 Analyse et conception.....	20
1. Introduction.....	20
2. Architecture générale de notre application.....	20
3. UML.....	21
4. Diagramme de cas d'utilisation.....	21
4.1. Définition.....	21
4.2. Acteurs.....	21
4.3. Cas d'utilisation.....	21
4.4. Spécification des exigences.....	24
5. Diagrammes des séquences.....	25
5.1. Authentification.....	25

5.2. Inscription.....	27
5.3. Ajouter un avis de décès.....	29
6. Diagramme des classes.....	30
7. Conclusion.....	31
Chapitre 3 Réalisation.....	32
1. Introduction.....	32
2. Environnement de travail.....	32
2.1. Environnement matériel.....	32
2.2. Environnement logiciel.....	32
2.3. Bibliothèques utilisés.....	33
2.4. Architecture Client/Server.....	34
3. Implémentation de la base de données.....	35
3.1. NoSQL.....	35
3.2. Firebase.....	35
3.2.1. Firebase Authentication.....	39
3.2.2. Firebase Realtime Data Base.....	39
3.2.3. Firebase Storage.....	40
3.2.4. Firebase Cloud Messaging.....	41
4. Présentation de l'application.....	41
4.1. Les interfaces de l'application.....	41
4.2. Interfaces supplémentaires pour l'admin.....	45
5. Conclusion.....	46
Conclusion générale.....	47
Bibliographie.....	48
Résumé.....	50

Table des Figures

FIGURE 1. 1 STATISTIQUES POUR LES UTILISATEURS DES MOBILES EN ALGERIE.....	4
FIGURE 1. 2 ARCHITECTURE D'ANDROID	10
FIGURE 1. 3 MACHINE VIRTUELLE DALVIK	11
FIGURE 1. 4 PAGE DE PARAMETRES DU ANDROID SDK	13
FIGURE 1. 5 FICHIER XML DE L'ACTIVITE	14
FIGURE 1. 6 CODE SOURCE DE L'ACTIVITE (EN JAVA)	14
FIGURE 1. 7 MANIFEST.XML	16
FIGURE 1. 8 CYCLE DE VIE D'UNE APPLICATION ANDROID	17
FIGURE 2. 1 FONCTIONNALITE GENERALE DU SYSTEME.....	20
FIGURE 2. 2 DIAGRAMME DES CAS D'UTILISATION.....	23
FIGURE 2. 3 DIAGRAMME DE SEQUENCE DE L'AUTHENTIFICATION	26
FIGURE 2. 4 DIAGRAMME DE SEQUENCE D'INSCRIPTION.....	27
FIGURE 2. 5 DIAGRAMME DE SEQUENCE MODE D'ABONNEMENT	28
FIGURE 2. 6 DIAGRAMME DE SEQUENCE D'AJOUT UNE PUBLICATION	29
FIGURE 2. 7 DIAGRAMME DES CLASSES.....	30
FIGURE 3. 1 LOGO OFFICIEL DE ANDROID STUDIO	33
FIGURE 3. 2 LOGO DE JDK	33
FIGURE 3. 3 LOGO DE NODE.JS	33
FIGURE 3. 4 ARCHITECTURE CLIENT/SERVER	34
FIGURE 3. 5 PREMIERE ETAPE DE L'IMPLEMENTATION.....	35
FIGURE 3. 6 DEUXIEME ETAPE DE L'IMPLEMENTATION	36
FIGURE 3. 7 TROISIEME ETAPE DE L'IMPLEMENTATION.....	36
FIGURE 3. 8 FICHIER BUILD.GRADLE AU NIVEAU DE L'APPLICATION	37
FIGURE 3. 9 FICHIER BUILD.GRADLE AU NIVEAU DU PROJET	37
FIGURE 3. 10 DERNIERE ETAPE DE L'IMPLEMENTATION	38
FIGURE 3. 11 CONSOLE DE FIREBASE.....	38
FIGURE 3. 12 AUTHENTIFICATION FIREBASE	39
FIGURE 3. 13 FIREBASE REALTIME DATABASE	40
FIGURE 3. 14 FIREBASE CLOUD STORAGE.....	40
FIGURE 3. 15 FIREBASE CLOUD MESSAGING.....	41
FIGURE 3. 16 LOGO DE L'APPLICATION	41
FIGURE 3. 17 SPLASHSCREE	42
FIGURE 3. 18 MODE DE CONNEXION.....	42
FIGURE 3. 19 FENETRE DE LOGIN	42
FIGURE 3. 20 FENETRE D'INSCRIPTION	42
FIGURE 3. 21 POP-UP ALERTE.....	43
FIGURE 3. 22 FENETRE DE CHOIX	43
FIGURE 3. 23 LA FENETRE PRINCIPALE	43
FIGURE 3. 24 FENETRE DE DETAILS.....	43
FIGURE 3. 25 NAVIGATION DRAWER	44
FIGURE 3. 26 FENETRE DE PROFIL	44
FIGURE 3. 27 AJOUTER UNE PUBLICATION.....	44
FIGURE 3. 28 PARAMETRES.....	44
FIGURE 3. 29 FENETRE SUPPLEMENTAIRE	45
FIGURE 3. 30 FENETRE DE GESTION POUR L'ADMIN	45

Liste des tableaux

TABLE 2. 1 CAS D'UTILISATION AUTHENTIFICATION	24
TABLE 2. 2 CAS D'UTILISATION INSCRIPTION	24
TABLE 2. 3 CAS D'UTILISATION AJOUTER UNE PUBLICATION	24
TABLE 2. 4 CAS D'UTILISATION AJOUTER RECOMMANDATION	25

Introduction générale

De nos jours, Le smartphone, devenu compagnon indispensable et outil multifonctions. Des millions d'individus tapotent sur leurs écrans tactiles à longueur de journée, ne pouvant plus se passer de ce petit appareil. Pour cette raison, Les éditeurs des systèmes d'exploitation pour ces appareils supportent et incitent les développeurs à développer des applications mobiles. En Algérie, tout comme dans les autres régions du monde, l'utilisation d'appareils mobiles est en croissance rapide. Les utilisations des appareils mobiles s'effectuent grâce à des applications Mobiles. Que ce soit pour communiquer et utiliser des services sur Internet, pour se divertir en regardant des vidéos, en jouant à des jeux ou en partageant des photos, pour faire des transactions telles que des achats et des opérations bancaires, le besoin de développer et de mettre à jour des applications mobiles est en croissance. L'activité de développement d'applications mobiles requiert de plus en plus une main-d'œuvre apte à réaliser des projets de développement de programmes informatiques spécifiques à la mobilité, et ce, sur différentes plates-formes.

En décembre 2018, mon grand-père est mort - Dieu le bénisse -, Les gens venaient de partout pour la consolation. Mais, il y'a des gens qui n'ont reçu l'information concernant les funérailles de mon grand-père que quelques jours plus tard. Donc, malgré tous les moyens disponibles, il y'a un manque dans les informations des funérailles. Ce manque nous met de poser des questions concernant les moyens disponibles sur le terrain, et comment petons-nous résoudre ce problème de manque.

Depuis longtemps et avant la révolution technologique, les informations des funérailles (ou des décès) arrivaient par une déclaration de l'appelant (ELBERRAH), qui lance l'appel à voix haute dans les places publiques. Et au fil de temps, les avis des décès deviennent imprimés sur papiers et accrochées dans les mosquées et les places publiques. De plus, les gens échangent les avis des décès via les téléphones, et les partagent sur des pages et groupes des réseaux sociaux. Mais, malgré tous ces moyens, il y'a toujours un manque dans les informations des décès, puisque à chaque funérailles, il y'a des gens qui ne reçoivent l'annonce de cette funérailles que quelques jours plus tard.

Pour résoudre ce problème de manque, on va développer une application mobile sous Android dans le cadre d'un projet de fin d'études dans le but de l'obtention du diplôme des

études de master académique en Réseaux et Technologie de l'Information et de la Communication à l'université de M'Sila.

Notre mémoire est divisé en trois chapitres présentant les étapes de réalisation d'une application mobile sous Android pour les funérailles de la ville de M'Sila. Dans le premier chapitre, nous parlons en bref des systèmes d'exploitations mobiles. Puis, nous présentons des généralités sur Android et les stratégies préconisées pour leur développement. Par la suite, nous exposons en détails la problématique de la funérailles et des généralités sur les funérailles en M'Sila. Dans le deuxième chapitre, nous présentons l'architecture générale de notre application. Puis, nous identifions les acteurs principaux qui joueront des rôles dans notre application et leurs fonctionnalités à travers de diagramme de cas d'utilisation, suivi par des diagrammes de séquences qui illustre des descriptions détaillées pour les scénarios des importants cas d'utilisations. Enfin on va présenter le diagramme de classes. Dans le troisième chapitre, nous présentons tous les outils matériels et logiciels utilisés pour le développement de notre application. Ainsi, nous présentons les différentes interfaces de notre application pour les funérailles de la ville de M'Sila. À l'issue de ce projet, la conclusion générale mettra en exergue les principaux résultats, et quelques perspectives intéressantes concernant ce travail.

CHAPITRE 1

GENERALITES SUR LES APPLICATIONS MOBILES

SOUS ANDROID

1. Introduction

Dans ce chapitre, nous parlons en bref sur les applications mobiles en générale. Ensuite nous présentons spécialement le système d'exploitation Android, et les outils nécessaires pour développer une application mobile sous Android. Et à la fin, on va présenter en détails la problématique de la funérailles et des généralités sur les funérailles en M'Sila.

2. Applications mobiles

2.1. Définition

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil mobile, généralement un téléphone mobile, un téléphone intelligent ou une tablette numérique. Les applications mobiles sont des programmes relativement légers, autonomes, utilisés pour des services de l'information, des médias sociaux, des jeux, etc. [1]

2.2. Systèmes d'exploitation mobiles

Tout comme un ordinateur dispose d'un système d'exploitation, les téléphones mobiles se composent également d'une plateforme qui contrôle toutes leurs fonctionnalités. Ceci est connu comme un système d'exploitation mobile. Généralement connu sous le nom d'OS (Operating System) mobile, il s'agit d'un système d'exploitation qu'exploite un appareil mobile tel qu'un Smartphone, une tablette tactile, etc. Il contrôle et coordonne toutes les opérations de base du téléphone mobile comme les options d'écran tactile, Bluetooth, Wifi, appareil photo, etc. et assure la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeux vidéo, etc.). [2]

2.2.1. Windows phone

Développé par Microsoft pour les Smartphones et Pocket PC, Windows Phone succède à Windows Mobile en proposant en plus des applications basiques comme la messagerie électronique, Internet, Chat et Multimédia, des fonctionnalités média sociaux tels que Facebook et Twitter. [2]

2.2.2. BlackBerry OS

Système d'exploitation fonctionnant sur les Smartphones BlackBerry, il permet aux développeurs de mettre en place des applications en utilisant les APIs BlackBerry. Toute

application doit être signée numériquement par le compte RIM (Research In Motion) du développeur. [2]

2.2.3.IOS

Anciennement connu sous le nom iPhone OS, iOS est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod et l'iPad. Il est dérivé de OS X dont il partage les fondations (le noyau hybride XNU basé sur le micronoyau Mach, les services Unix et Cocoa, etc.). iOS est le deuxième système d'exploitation le plus répandu sur le marché. [3]

2.2.4. Android

Android est un système d'exploitation open source développé par Android Inc., une startup rachetée par Google en 2007. Fondé sur un noyau Linux, le système a d'abord été conçu pour les Smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions, les voitures, les ordinateurs et les SmartWatch. Android a été conçu pour intégrer au mieux des applications existantes de Google telles que le service de messagerie électronique Gmail, celui de la cartographie GoogleMaps, ou encore YouTube, Google Talk et GoogleCalendar. [4]

2.3. Pourquoi Android

En Algérie, les utilisateurs des smartphones qui ont l'Android comme un système d'exploitation, deviennent 95.58 % par rapport des autres systèmes d'exploitations en janvier 2019 [5]. Et puisque notre application doit couvrir le plus grand nombre d'utilisateurs, et d'après les statistiques que nous avons vues, nous avons choisi l'Android.

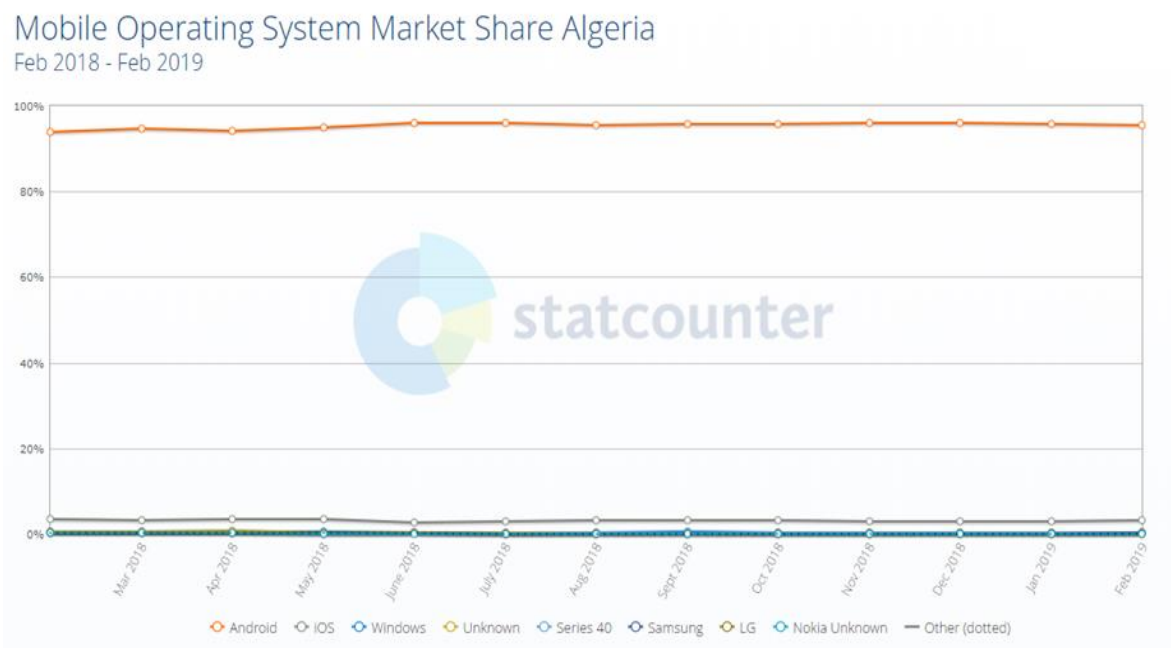


Figure 1. 1 Statistiques pour les utilisateurs des mobiles en Algérie

3. Présentation du système Android

3.2. Définition

Android est un système d'exploitation pour tablettes et smartphone, aussi appelé téléphone intelligent ou Ordiphone. Ce projet est né d'un consortium de 34 entreprises (en 2008), initié par la société Google le 5 novembre 2007, et appelé l'Open Handset Alliance ou OHA. Son but consistait à trouver une solution fiable pour concurrencer Apple avec l'iPhone OS, Microsoft avec Windows Mobile, Nokia avec Symbian et Research In Motion avec BlackBerry OS. En effet à cette époque, la quasi-totalité des smartphones fonctionnait sur ces systèmes d'exploitation. Cependant, la principale différence d'Android face aux autres solutions est qu'il soit open source. Cette particularité le rend donc gratuit, et personnalisable par les constructeurs et les opérateurs de téléphonie mobile. Malgré des personnalisations accrues en terme d'interface utilisateur et de fonctionnalité, le cœur du système reste commun ce qui permet une interopérabilité des applications. [6]

3.3. Les versions de Android

3.3.4. Android 1.0 et 1.1

À la date du 22 octobre 2008, le premier smartphone sous Android débarque chez l'opérateur T-Mobile en Amérique, le HTC Dream (ou HTC G1). Et il n'y avait pas encore de clavier ou la gestion du multitouche. La grande innovation pour cette première version d'Android fut la barre de notifications, au travers d'un menu déroulant qui regroupe plusieurs affichages pratiques et rapidement accessibles. Ce fut aussi l'apparition des widgets et de l'ancêtre du Google Play Store, l'Android Market. La mise à jour 1.1 dénommée « Banana Bread », en version bêta, est vite arrivée dès le mois de février 2009. Elle corrigeait de nombreux bugs. De plus, les mises à jour via OTA allaient enfin se démocratiser pour installer les futures versions du système en passant par une connexion Wifi ou 3G. [7]

3.3.5. Android 1.5 CupCake

La mise à jour 1.5 « CupCake » a été publiée le 30 avril 2009, améliorant l'interface graphique de Android. À partir de celle-ci, il a été décidé de nommer les différentes versions du système avec des noms de gâteaux (peut-être que les entreprises étaient gourmandes), en suivant l'ordre alphabétique. CupCake était donc la troisième itération de l'O.S, mais surtout la venue (enfin) du premier clavier tactile, et du presse-papier. [7]

3.3.6. Android 1.6 Donut

Si la version 1.5 était une mise à jour majeure, celle qui a suivi l'était moins. Android 1.6 « Donut » offrait tout de même quelques nouvelles fonctionnalités comme la prise en charge du réseau CDMA, le support d'autres définitions d'écran. Cette version apportait également la fonction de recherche directement en local sur votre appareil Android, en comptant également sur les recherches internet grâce à Google Search. [7]

3.3.7. Android 2.0 et 2.1 Eclair

Google a frappé fort à la fin du mois d'octobre 2009, en publiant la version Android 2.0 « Eclair », tandis que Motorola annonçait son téléphone Droid. Il aura surtout été possible grâce à cette mise à jour d'associer plusieurs comptes Google au même appareil. Celle-ci aura été majeure et aura apporté beaucoup de fonctionnalités intéressantes, comme les suivantes : Amélioration du clavier tactile, Fonds d'écran animés, Prise en charge du Bluetooth 2.1, Arrivée de Google Maps, Nouvelles fonctions pour l'appareil photo (flash, zoom digital, balance des blancs, effets de couleurs et focus macro) et la fonction « speech-to-text » fait également son entrée et introduit la synthèse vocale au sein du système Android. La mise à jour 2.1 quant à elle n'était que mineure, mais on se souviendra que Google a travaillé sur son premier smartphone Android, le Nexus One, un bel appareil déjà et qui était doté du premier processeur Qualcomm Snapdragon. [7]

3.3.8. Android 2.2 Froyo, et 2.3 Gingerbread

Le 20 mai 2010 arrivait la version Android 2.2 « Froyo », qui a profité au Nexus One tout d'abord (Google oblige) et ce fut à nouveau de belles améliorations pour le système open-source. De nouveaux raccourcis ont été ajoutés aux côtés du lanceur d'applications : le téléphone et le navigateur natif. La mise à jour proposait également du nouveau pour la galerie avec une meilleure interface pour regarder vos photos, des effets de transition, ce qui rendait l'O.S. encore plus convivial. C'est une version particulière qui a apporté le support de Flash Player 10.1, la prise en charge des écrans ayant une densité de pixels de 320 ppi et l'ajout du fameux écran de verrouillage à mot de passe ou code PIN. Et arriva ce qui arriva, le 6 décembre de la même année, Google publie la version 2.3 « Gingerbread ». Elle n'aura pas apporté grand-chose, mais surtout des améliorations sur la prise en charge et le support de la VoIP, le NFC, le capteur frontal sur les appareils concernés et un passage au système de fichiers ext4 que les amateurs de GNU/Linux connaissent fort bien. [7]

3.3.9. Android 3.0 Honeycomb

Dans toute l'histoire de l'OS, il existe une mise à jour, la 3.0 « Honeycomb », qui fut réservée aux tablettes Android. À cette époque, la société Motorola détenait l'exclusivité avec son smartphone Droid et la version 2.0 du système. La tablette Xoom est née avec cette mise à jour Honeycomb et nous avons pu découvrir le nouveau virement de bord de la part de Google sur l'interface. Le multitâche fut également à l'honneur dès l'arrivée de cette mise à jour grâce à Matias Duarte devenu designer chez Google pour repousser les limites de l'expérience utilisateur. Du côté des mises à jour suivantes, la 3.1 et la 3.2, on aura apprécié de pouvoir enfin redimensionner les widgets sur ses pages, mais également le support du format FLAC (pour les amoureux de la musique) et la prise en charge des processeurs Qualcomm et des tablettes tactiles de 7 pouces. [7]

3.3.10. Android 4.0 IceCreamSandwich

C'est le 19 octobre 2011 que l'histoire d'Android va atteindre son apogée (ou presque). L'arrivée de la version 4.0 « Ice Cream Sandwich », compatible avec les smartphones et les tablettes, nous laisse rêveur avec beaucoup de nouveautés tant attendues par les utilisateurs, et quelques-unes que je vous cite : Ajout de dictionnaires dans le clavier virtuel, Application photo améliorée, rapidité de capture quasiment en mode rafale, Capture d'écran native, Consultation de son trafic data, pour ne plus avoir de hors forfait à votre prochaine facture, Création plus facile de dossiers d'applications, Écran de verrouillage repensé, sous forme d'un cadenas qui permet de lancer des applications en raccourci, Mise à jour du navigateur (ouvertures de plusieurs pages, navigation privée) et enfin la Prise en charge de l'enregistrement vidéo Full HD 1080p. [7]

3.3.11. Android 4.1, 4.2, 4.3 Jelly Bean, et 4.4 KitKat

« Jelly Bean » est le nom d'Android, annoncé le 27 juin 2012 lors du Google I/O. Le premier appareil équipé de la mise à jour 4.1 fut la Nexus 7, la tablette de Google commercialisée à partir du 13 juillet 2012 aux États-Unis. C'est une mise à jour qui fut très attendue, et pour preuve, à l'heure actuelle, nos appareils sont encore équipés d'une version Jelly Bean de l'O.S. Le 13 novembre 2012 sortait la version suivante, Android 4.2, toujours une Jelly Bean. Avec cela, Google a fait l'annonce de trois nouveaux dispositifs : le Nexus 4, la Nexus 10 et une prochaine Nexus 7. Une mise à jour qui a comme d'habitude, apporté un lot de nouvelles fonctionnalités. Le 24 juillet 2013 sera la date de la dernière mise à jour Android 4.3 qui est disponible sur de très nombreux appareils grâce à la mise à jour en OTA. De prime abord, la

mise à jour 4.3 a délivré la restriction au niveau des comptes utilisateur, afin de pouvoir bloquer ou non des accès à certaines applications, si vous prévoyiez de laisser jouer vos enfants. On pourra également noter le support du Bluetooth Smart, pour un transfert de données encore plus rapide, en économisant de la batterie. C'est aussi le début des prises en charge de Android Wear, l'O.S. des smartwatches. Avec Android KitKat, la fameuse version 4.4, Google se renforce encore sur la branche 4.X. Le système est annoncé le 31 octobre 2013 d'abord sur le tout nouveau Nexus 5 puis sera déployé sur les autres terminaux. Il s'agit encore aujourd'hui de l'une des versions d'Android les plus couramment utilisées. Google a très tôt annoncé son intention de faire migrer un maximum d'appareil vers cette version d'Android. La fragmentation est en effet l'un des ennemis majeurs du système d'exploitation. [7]

3.3.12. Android 5.0, 5.0.1, 5.0.2 et 5.1 Lollipop

Android 5.0 Lollipop a été officiellement nommé ainsi le 15 octobre 2014 par Google. De nombreuses mises à jour sont depuis en cours vers cette nouvelle version majeure qui fut disponible publiquement le 3 novembre de la même année pour les appareils Nexus dans un premier temps. Les Nexus 6 et 9 en furent d'ailleurs équipés dès leur commercialisation. Premier changement qui saute aux yeux, le Material Design. Google a décidé de revoir tout l'habillage graphique de son système d'exploitation. La sécurité est aussi au centre des occupations du système et le noyau a été largement revu en ce sens. Lollipop marque l'abandon de la machine virtuelle Dalvik au profit d'ART (Android Runtime). Avec ce dernier, on nous promet de meilleures performances au niveau général du système. [7]

3.3.13. Android 6.0 Marshmallow

Le surnom de Marshmallow a été officialisé le 18 août 2015. Côté nouveautés apportées, Google a souligné 6 fonctions principales qui porteront Android 6.0, mais en réalité, rien que la version développeur en comptabilise déjà une cinquantaine, que ce soit par des ajouts de fonctionnalités inédites ou d'améliorations de certaines déjà existantes. Premièrement, une amélioration permet d'accéder à Google Now depuis l'écran de déverrouillage. Le support de Google Now a également été revu et optimisé par le Now On Tap, vous y donnant accès depuis n'importe quel écran après activation de l'option « Appuyer sur l'écran pour afficher Google Now ». [7]

3.3.14. Android 7.0 Nougat

Android Nougat, c'est la 7ème version du système d'exploitation Android. Annoncé en grande pompe par Google lors d'un événement exceptionnel à Montélimar, cette mise à jour propose plusieurs nouvelles fonctionnalités. On commence par un des plus gros changements le multi-fenêtre disponible nativement sur Android Nougat, cela permet d'avoir deux applications en même temps sur l'écran. Il y a les mises à jour dites « transparentes », il faut comprendre par là qu'elles se font sans que l'utilisateur ne s'en aperçoive, il faut juste redémarrer l'appareil. Au niveau des optimisations on remarque un multitâches toujours plus intelligent qui ferme les applications que vous n'utilisez plus depuis un moment, normalement seulement les sept dernières seront disponibles. Le mode Doze est également sûr boosté, gagnant en « intelligence » également. On note aussi l'arrivée de la fonction Data server qui permet d'économiser encore plus de data. Cette version est passée à l'OpenJDK pour éviter de nouveau procès avec Oracle. Elle supporte aussi la nouvelle API Vulkan permettant de meilleurs graphismes dans les jeux vidéo tout en consommant moins. [7]

3.3.15. Android 8.0 Oreo

Nommée à partir du célèbre biscuit Oreo, cette 8ème version a été dévoilée par Google fin du mois d'août 2017. Malgré quelques bugs de lancement plutôt ennuyeux, cette nouvelle mise à jour propose d'intéressantes nouveautés, comme 70 nouveaux émojis, le Picture-in-Picture, la gestion multitâche et le démarrage rapide, une meilleure gestion des notifications, [7]

3.3.16. Android 9.0 Pie

Android Pie (en français, « Tarte ») est la version 9.x du système d'exploitation mobile Android développé par Google. Cette version du système est disponible depuis le 6 août 2018 pour les smartphones de la gamme Google Pixel, l'Essential Phone2, Nokia et OnePlus. Elle est décrite par le numéro d'API 28. L'un des points majeurs de cette nouvelle mise à jour est son utilisation de l'intelligence artificielle Google afin d'optimiser dynamiquement votre téléphone. La navigation système par gestes est désormais totalement intégrée à Android 9.0 Pie. Elle se base sur une petite « pilule » en bas de votre téléphone que vous pouvez remonter, faire glisser sur le côté ou sur laquelle vous pouvez tout simplement appuyer pour atteindre l'accueil, ouvrir vos applications en cours ou revenir à votre action précédente. [8]

3.4. L'architecture d'Android

L'architecture de la plateforme Android se base sur quatre (04) principaux niveaux : un noyau linux, environnement d'exécution et des libraires, un module de développement d'applications et les différentes applications. La figure suivante schématise ces quatre principaux.

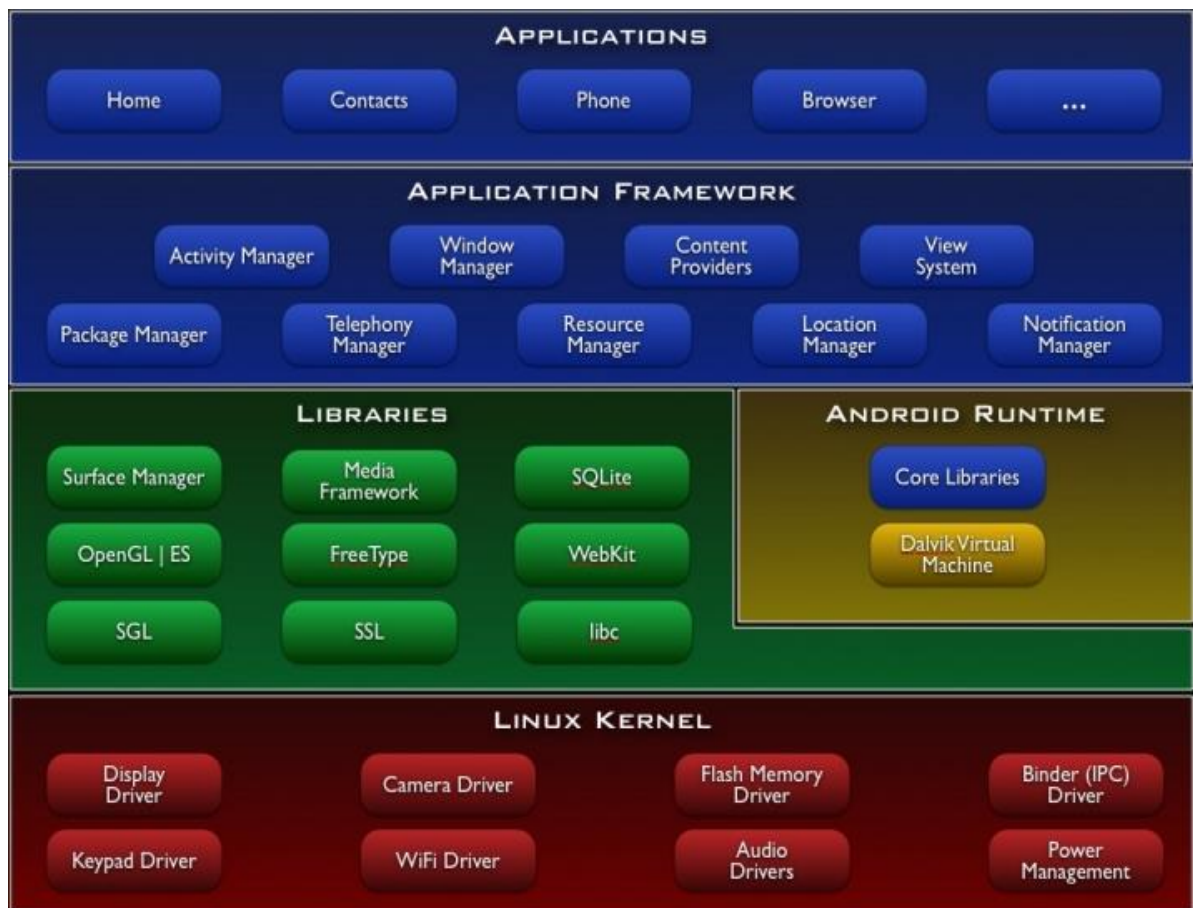


Figure 1. 2 Architecture d'Android [9]

3.4.4. Noyau linux

Android s'appuie sur un noyau (Kernel en anglais) dont la version est linux 2.6. Pour être précis, ce noyau est l'élément du système d'exploitation qui permet de faire le pont entre la partie matérielle et la partie logicielle. D'après le schéma, cette couche est la seule qui gère le matériel. Cette version de noyau est une version conçue spécialement pour l'environnement mobile.

3.4.5. Les libraires (Bibliothèques)

Ce sont des bibliothèques proviennent de plusieurs projets open-sources. La plupart de ces projets sont écrits en c/c++. [9]

3.4.6. L'environnement d'exécution

Ou Android runtime, il basé sur le concept de machine virtuelle, utilisée en Java. Etant donné les limitations des dispositifs, il n'a pas été possible d'utiliser une machine virtuelle Java standard. A cause de ça, Google a créé Dalvik - une nouvelle machine virtuelle -, afin de mieux répondre à ces limitations. On retrouve deux parties essentielles dans l'environnement : le premier est les Corêts Libraires, qui sont fournissent le langage Java disponible pour les applications. Java reprend en grande partie l'API JSE 1.5. Et la deuxième partie c'est Dalvik. Les applications Java doivent être compilées au format Dalvik exécutable (.dex) avec l'outil (dx). Cet outil compile les (.java) en (.class) puis il convertit ces (.classes) en (.dex).

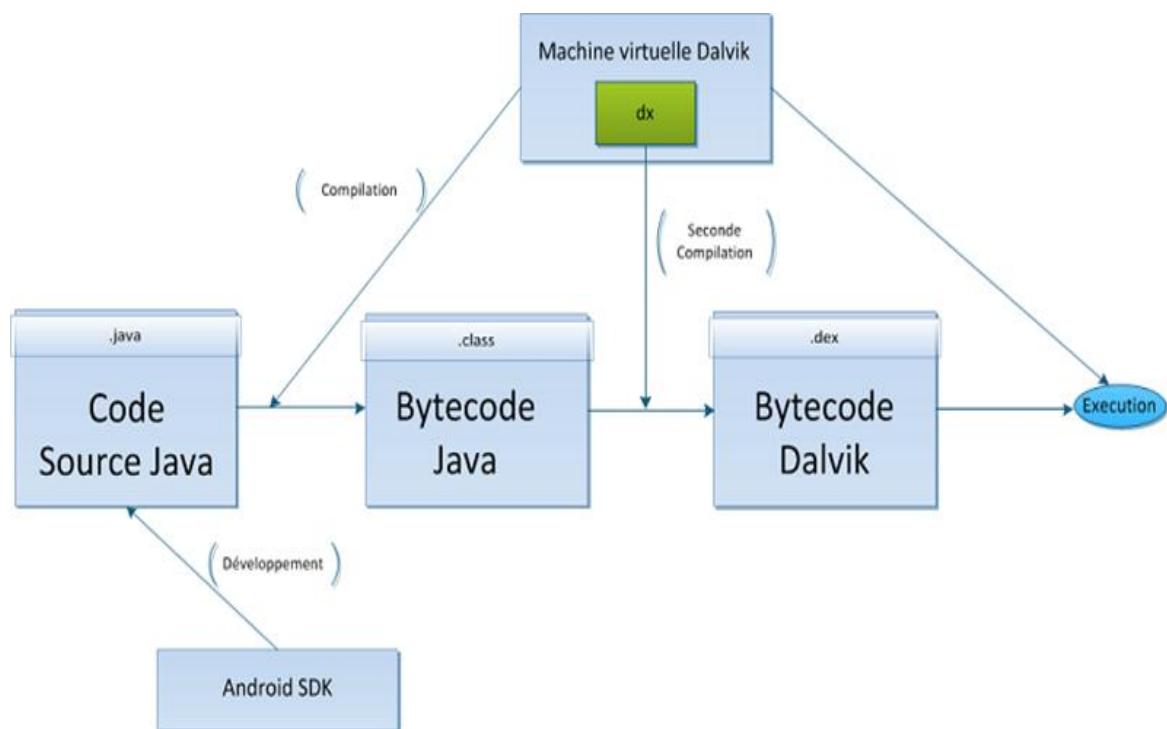


Figure 1. 3 Machine virtuelle Dalvik [9]

3.4.7. Le module de développement d'applications

Le Framework fournit les API qui permettant de créer des applications riches. Il est situé au-dessus de l'Android Runtime. Il y'a deux types de service, core PlatformService et HardwareService.

Les services cœurs de la plateforme (Corê Platform Service) fournissent des services essentiels au fonctionnement de la plateforme :

- Activity Manager : Conserve la pile de navigation permettant le transfert entre les applications et contrôle le cycle de vie de l'application.
- Package Manager : permet de charger les informations qui sont placées dans des fichiers (.apk) (Android package file).
- Window Manager : il gère les fenêtres des applications (quelle fenêtre doit être affichée devant une autre à l'écran).
- Resource Manager : gère tout ce qui n'est pas du code, toutes les ressources (images, fichier audio, etc.).
- Content Provider : gère le partage de données entre applications.
- View System : fournit tous les composants graphiques : listes, grilles, boutons, etc...

Le deuxième type, les services matériels (Hardware Service), fournissent un accès vers les API matérielles (GPS, Bluetooth, USB, Wifi, capteurs ...).

3.4.8. Les applications

Tous les applications installées et l'on peut trouver dans Android.

3.5. Présentation du SDK

Kit de développement logiciel, se présente comme un ensemble d'outils que prend à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Il permet en conséquence, de développer des applications exclusivement et uniquement pour l'Android. [10]

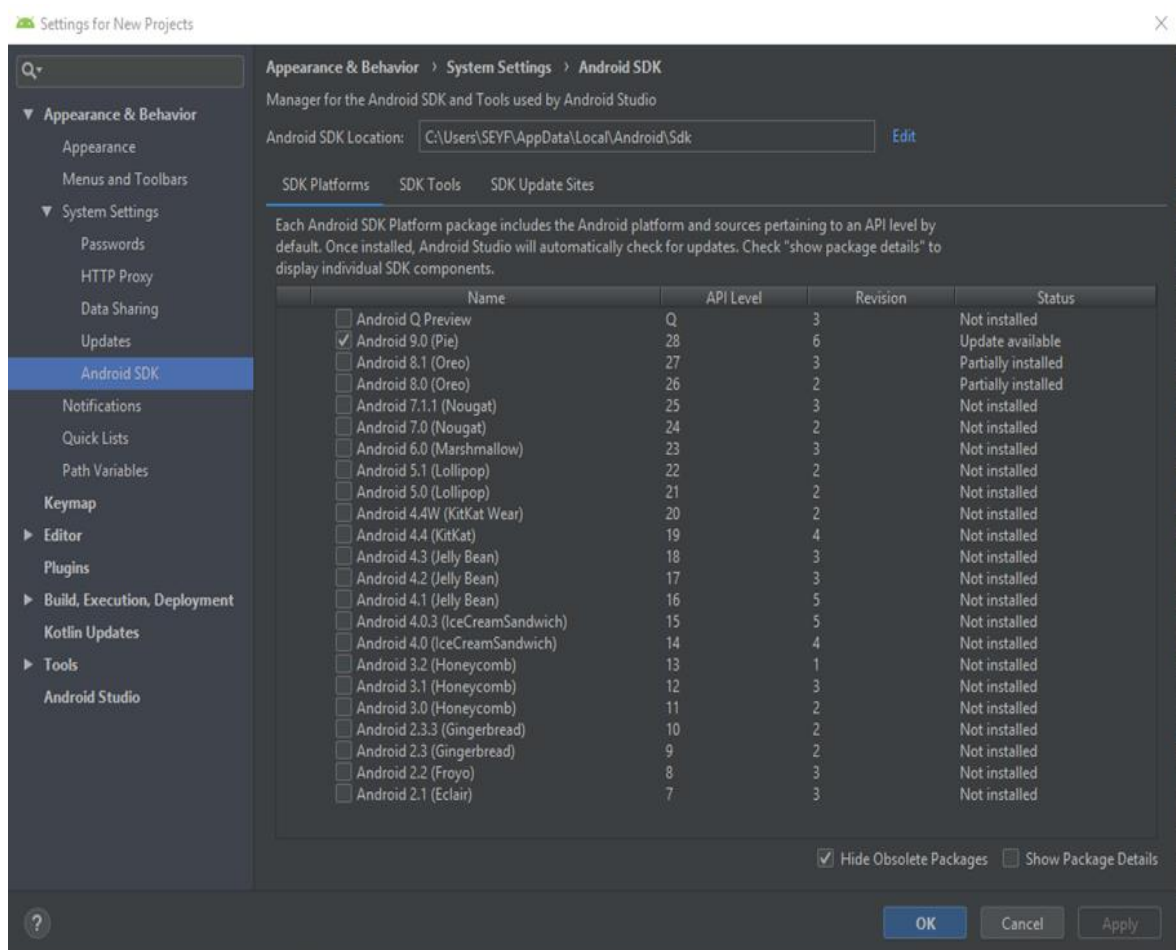


Figure 1. 4 Page de paramètres du Android SDK

D’après la figure précédente, nous remarquons que tous les noms suivent le même motif : {Android [nombre] (nom) API}. Les nombres signifie les versions de l’Android, et les noms sont les noms officiels des versions. Les nombres dans le champs API signifie la version de l’API associée. Tous ces nombres c’est d’une grande importance, puisque si on veut développer une application pour une version précise d’Android ne fonctionnera pas pour les versions qui la précède.

3.6. Composants d’une application Android

Les composants d’une application Android se présentent comme les éléments essentiels de cette application. Le fichier de manifestation d’application Android Manifest.xml contrôle et décrit chaque composant et comment il interagit.

3.6.4. Les Activités (Activity)

Une activité représente le bloc de base d'une application, elle est le composant principal d'une application. Elle signifie les différentes interactions des interfaces. On désigne et décrit l'interface par un fichier XML, liés par un code (en Java/Kotlin) pour faire les réactions et les différentes actions. Ce code implémenté sous la forme d'une sous-classe de classe mère d'activité. [11]

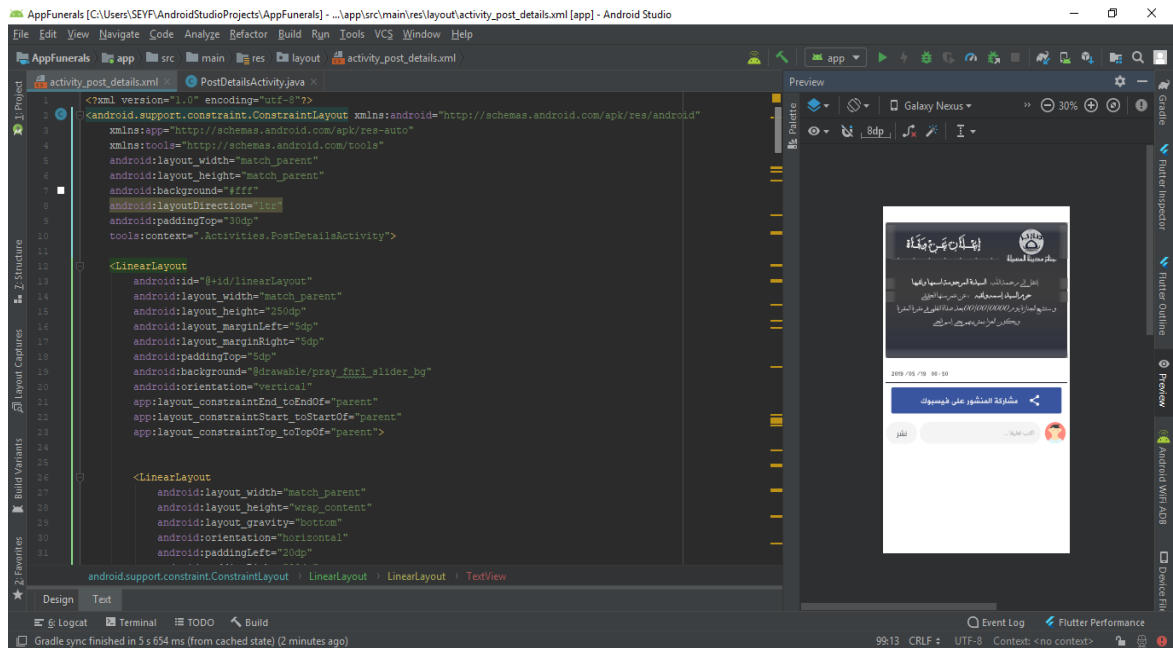


Figure 1. 5 Fichier XML de l'Activité

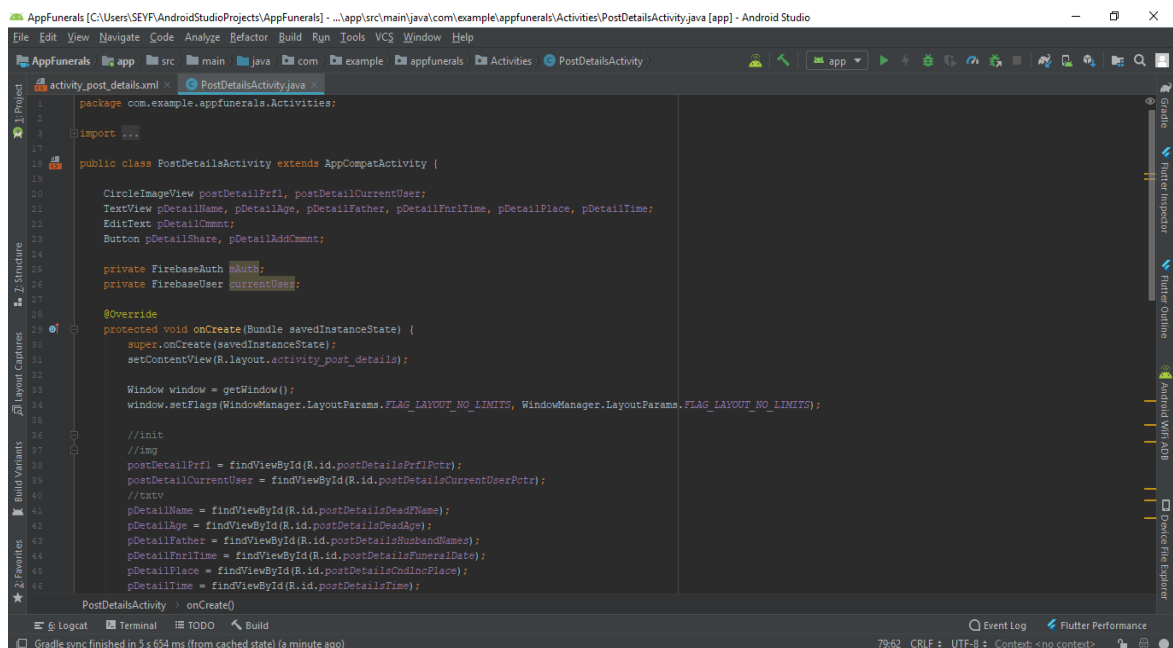


Figure 1. 6 Code source de l'Activité (en Java)

3.6.5. Les services

Un service est un composant qui fonctionne en tâche de fond, de manière invisible (en arrière-plan). Il ne nécessite pas une interface utilisateur pour l'appliquer. Le cycle de vie d'un service est différent d'un cycle de vie d'activité. Un service est mis en œuvre comme sous-classe de la classe `Service`. [11]

3.6.6. Récepteur de diffusion (Broadcast Receivers)

Un Récepteur de diffusion, permet d'écouter ce qui se passe sur le système ou sur l'application et de déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés. Les récepteurs de diffusion répondent simplement aux messages diffusés provenant d'autres applications ou du système. Un récepteur de diffusion est implémenté sous la forme d'une sous-classe de la classe `BroadcastReceiver` et chaque message est un diffuseur comme objet `Intent` : [11]

```
public Class MyReceiver extends BroadcastReceiver {  
    Public void onReceive (context, intention) {}  
}
```

3.6.7. Fournisseurs de contenu (Content Providers)

Un fournisseur de contenu permet de gérer et de partager des informations entre les applications sur demande. De telles requêtes sont traitées par les méthodes de la classe `ContentResolver`. Les données peuvent être stockées dans le système de fichiers, la base de données ou ailleurs. Un fournisseur de contenu est implémenté comme une sous-classe de la classe `ContentProvider` et doit implémenter un ensemble standard d'API permettant à d'autres applications d'effectuer des transactions. [11]

```
public Class MyContentProvider étend ContentProvider {  
    Public void onCreate () {}  
}
```

3.6.8. Composants supplémentaires [11]

Sont des composants utilisés dans la construction des entités susmentionnées.

- Fragments

Représente une partie de l'interface dans une activité.

- Views

Les éléments UI qui sont dessinés à l'écran.

- Layouts

Afficher les hiérarchies qui contrôlent le format d'écran et l'apparence des vues.

- Intents

Une intention (Intent) est un message système qu'on peut qualifier d'évènement. Les Intentions sont des objets permettant de faire passer des messages contenant de l'information entre composants principaux.

- Resources

Les éléments externes, tels que les chaînes, les constantes et les images étirables.

- Manifest

Un fichier XML permet de gérer et configurer l'application.

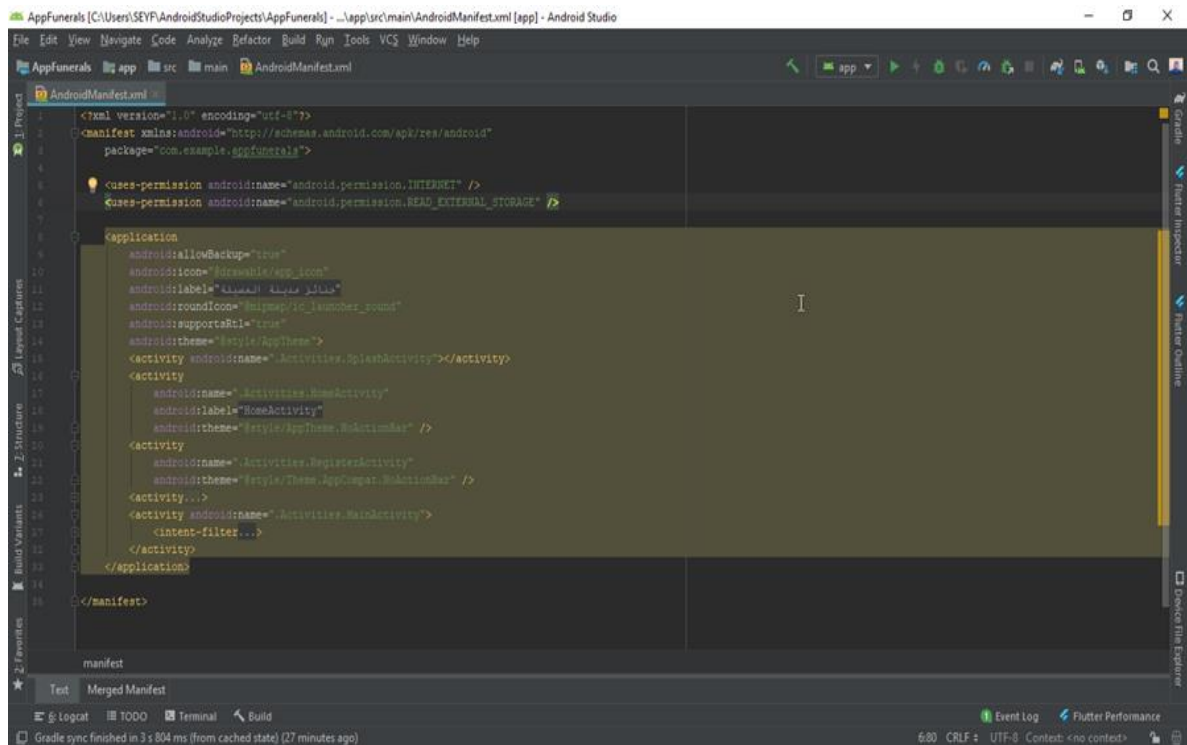


Figure 1. 7 Manifest.XML

3.7. Cycle de vie d'une application Android

Le cycle de vie d'une application est considéré les différents états d'une activité.

3.7.4. onCreate ()

Elle sert à initialiser L'activité ainsi que toutes les données nécessaires à cette dernière. Quand la méthode onCreate () est appelée, on lui passe un Bundle en argument. Ce Bundle contient l'état de sauvegarde enregistré lors de la dernière exécution de l'activité.

3.7.5. onStart ()

Signifie le début d'exécution de l'activité.

3.7.6. onResume ()

Si on avoir un état de pause, onResume () est appelée pour commencer l'activité à partir de l'état que cette activité mise en pause.

3.7.7. onPause ()

Cette méthode est appelée au passage d'une autre activité en premier plan. L'intérêt d'un tel appel est de sauvegarder l'état de l'activité et les différents traitements effectués par l'utilisateur.

3.7.8. onStop ()

Cette méthode est appelée quand l'activité n'est plus visible quel que soit la raison. Dans cette méthode vous devez arrêter tous les traitements et services exécutés par votre application.

3.7.9. onDestroy ()

Cette méthode est appelée quand votre application est totalement fermée (Processus terminé). Les données non sauvegardées sont perdues.

4. Présentation du sujet

4.2. Problématique

Comme nous savons, les funérailles sont des événements importants dans nos vies et nécessitent d'accélérer la publication des annonces de décès. Mais il y'a des problèmes affectant de ces annonces tel que :

- Uniquement les gens qui vont aux mosquées, ils voient les annonces des décès.
- La façon dont les appels téléphoniques couvrent un nombre limité de personnes.
- Les réseaux sociaux –comme Facebook-, ne sont pas toujours disponible.
- Les pages qui publient des annonces de décès ne sont pas accessibles à tout le monde.

4.2.4. Les moyens disponibles sur le marché

Depuis longtemps avant la révolution technologique, les informations des funérailles arrivaient par une déclaration de l'appelant « ELBERRAH ». ELBERRAH est un homme qui prend en charge les médias, en lançant l'appel à voix haute dans les marchés, devant les

mosquées et les places de la ville et ses routes, ainsi que dans divers villages. Il a annoncé les évènements publics (comme les funérailles). Au fil du temps les gens utilisaient les papiers imprimés accrochés dans les mosquées et les places publiques. De nos jours et avec la révolution technologique, les informations des funérailles deviennent transférées par les téléphones et les réseaux sociaux (Facebook, twitter, ...).

4.2.5. Les besoins du marché

Comme nous l'avons dit dans le problématique, malgré les moyens disponibles aujourd'hui il y'a un manque dans les informations des funérailles. Nous devons garantir que les gens recevoir l'information, en utilisant l'avantage de la technologie.

4.3. Solution par une application Android

Notre solution est de développer une application mobile permet et garantie que l'information doit être reçu par tous les utilisateurs. Et comme la plupart des gens en Algérie utilisent l'Android (et en particulier en M'sila), nous allons développer la sous Android.

4.3.4. Le but de notre application

Le but de notre application est de garantir que les informations des funérailles doivent arriver à toute utilisateur d'Android par des notifications spécialement pour les funérailles. L'utilisateur doit installer l'application puis créer un compte pour recevoir les notifications.

4.3.5. Les avantages de notre application

Une fois l'utilisateur installé l'application dans son mobile, il a les avantages suivants :

- L'application est exclusivement conçue pour les funérailles de ville de M'sila.
- L'utilisateur peut spécifier comment recevoir les notifications (par nom de famille spécifié, par cimetière et par quartier).
- L'utilisateur doit recevoir les notifications où que ce soit.
- L'utilisateur peut ajouter un avis de décès, et l'envoi aux autres utilisateurs.

5. Conclusion

Ce premier chapitre a été consacré à la présentation de la programmation sous Android, et comment peut-on utiliser ses mécanismes pour développer une application pour les funérailles de la ville de M'sila. Nous allons entamer la conception de notre projet dans ce qui suit, pour identifier les différentes fonctionnalités de l'application.

CHAPITRE 2

ANALYSE ET CONCEPTION

1. Introduction

Avant de commencer le développement de notre application, il faut préciser l'objectif de cette application en déterminant les besoins fonctionnels et non fonctionnels auxquels notre application doit répondre. Dans ce chapitre, on va présenter d'abord, une architecture générale de notre système qui contient les différentes interactions entre les acteurs et les autres composants du système. Puis, pour la conception nous utilisons le langage UML qui permet de représenter des concepts graphiques et de modéliser les applications. Cette modélisation UML montre les différents acteurs du système ainsi que les rôles qu'ils peuvent tenir. Nous avons utilisé StarUML pour créer les différents diagrammes UML.

2. Architecture générale de notre application

Le schéma ci-dessous présente les différentes réactions de l'application. Si un utilisateur fait un changement sur les données, une synchronisation démarre entre tous les autres appareils mobiles en temps réel.

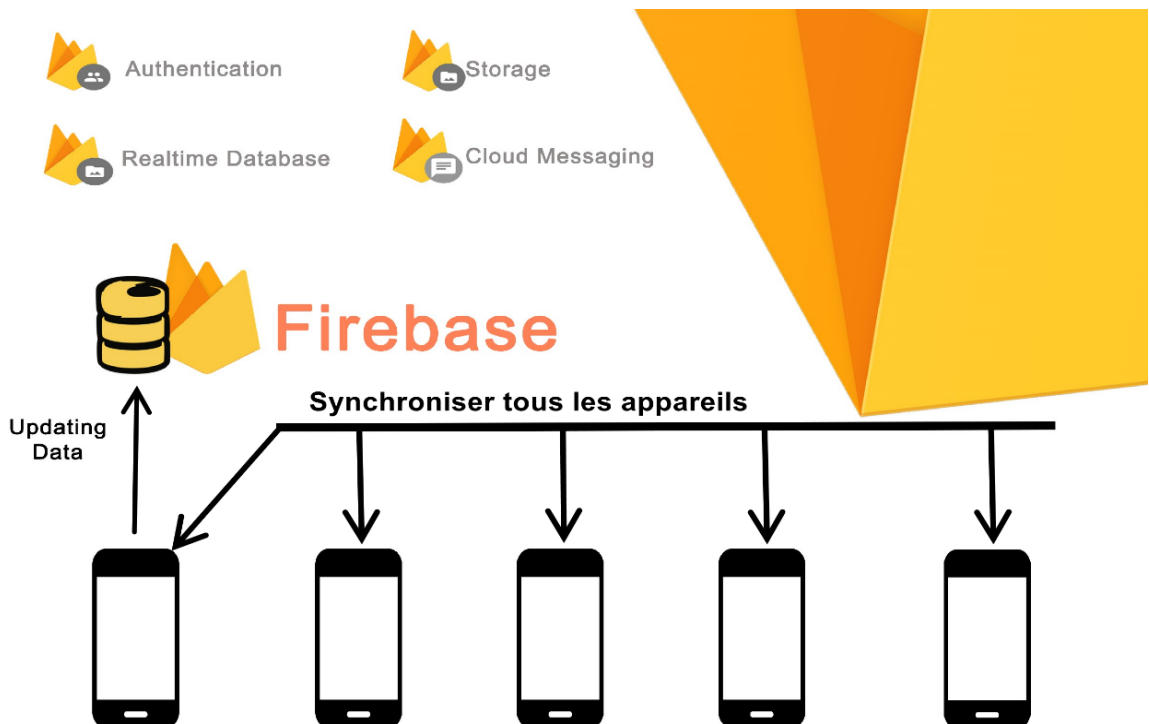


Figure 2. 1 Fonctionnalité générale du système

3. UML

UML (Unified Modeling Language) n'est pas une méthode (i.e. une description normative des étapes de la modélisation) : Ses auteurs, en raison de la diversité des cas particuliers, ont préféré de définir un langage graphique permettant de représenter et de communiquer les différents aspects d'un système d'information, au lieu de définir une méthode. Des textes sont associés aux schémas graphiques expliquant leur contenu. Il est donc un métalangage, car il fournit les éléments permettant de construire le modèle qui sera le langage du projet [12]. On va modéliser les acteurs et les différents besoins de notre système par un diagramme de cas d'utilisation, suivi de leurs diagrammes de séquences et enfin du diagramme de classes.

4. Diagramme de cas d'utilisation

4.1. Définition

Un diagramme de cas d'utilisation définit le comportement d'un système, comme vu par un utilisateur extérieur. Il décompose la fonctionnalité du système en unités cohérentes, à savoir les cas d'utilisation, ayant un sens pour les acteurs. Ceux-ci permettent d'exprimer le besoin des utilisateurs d'un système. [13]

4.2. Acteurs

Il y'a deux types d'acteurs qui peuvent utiliser notre application, utilisateur et administrateur.

- L'utilisateur est un membre qui a un compte sur notre application, il gère l'application dans la limite de son compte.
- L'admin est un utilisateur aussi, mais il a tous les droits de gestions et contrôles.

4.3. Cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'action qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera [14].

- Utilisateur

L'utilisateur peut faire les actions suivantes :

- Créer un compte
- Choisir un mode d'abonnement

- Voir tous les publications
 - Ajouter une publication
 - Partager une publication sur FACEBOOK
 - Commenter une publication
 - Supprimer son publication
 - Recevoir des notifications
 - Gérer son profil
 - Ajouter des recommandations
-
- Administrateur
 - Les mêmes actions de l'utilisateur, avec les privilèges suivants :
 - Gérer les publications (ajout, suppression et MAJ) cimetières cités
 - Gérer les utilisateurs
 - Gérer les noms des familles
 - Gérer les cimetières
 - Gérer les cités
 - Gérer les recommandations

La figure qui suite, présente le diagramme de cas d'utilisation de notre application.

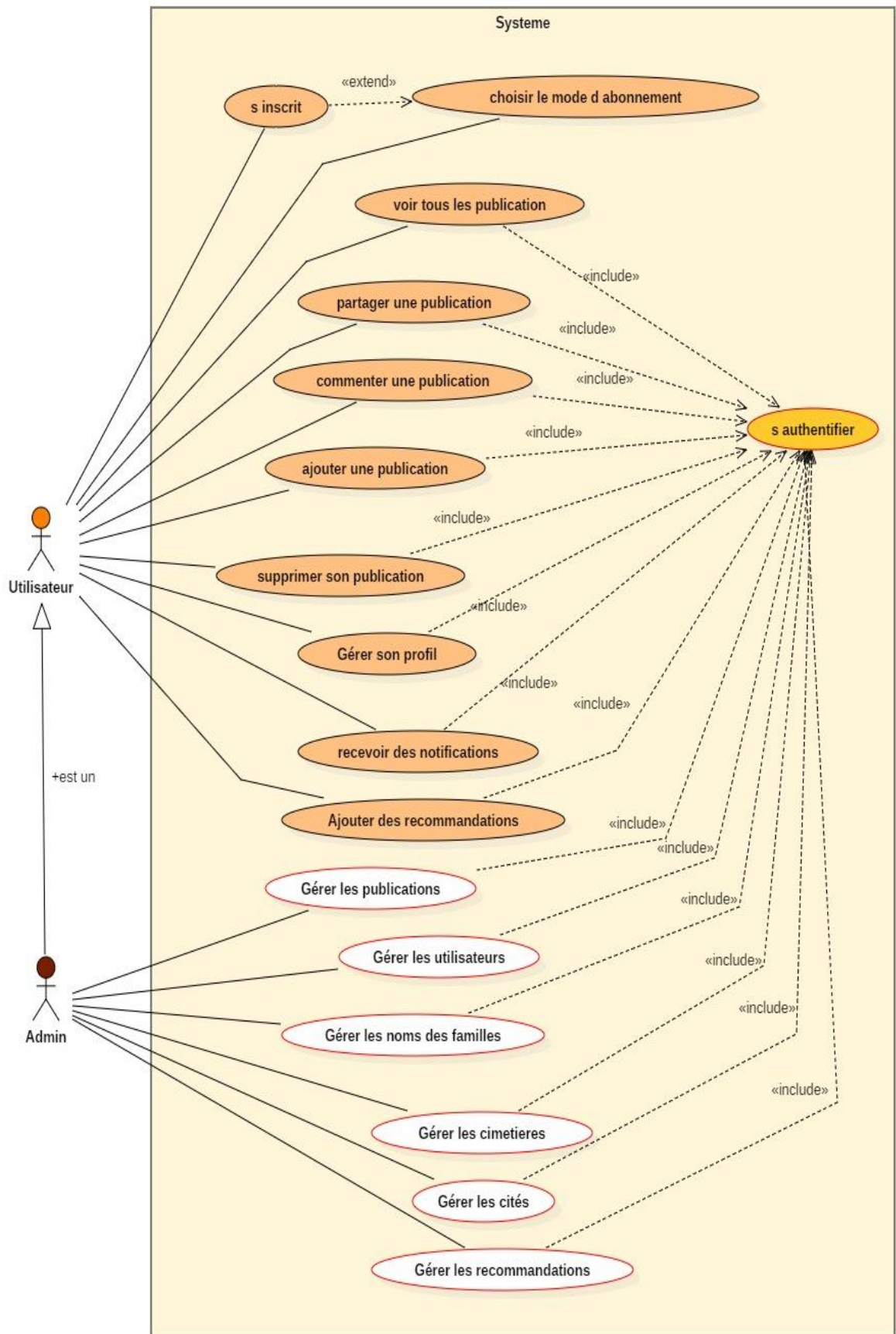


Figure 2. 2 Diagramme des cas d'utilisation

4.4. Spécification des exigences

On va décrire en détails certains cas d'utilisation identifiés dans le diagramme précédent.

- Cas d'utilisation Authentification

Cas d'utilisation	Authentification
Acteur	Utilisateur
Description	Authentifier l'utilisateur à partir de son email et mot de passe
Pré-condition	L'appareil mobile doit être connecté à Internet
Scénario nominal	Saisir l'email et mot de passe, puis valider
Post-condition	L'authentification avec succès
Scénario d'exception	Authentification échoué, message d'erreur doit s'afficher à l'utilisateur

Table 2. 1 Cas d'utilisation Authentification

- Cas d'utilisation Inscription

Cas d'utilisation	Inscription
Acteur	Utilisateur
Description	L'utilisateur crée un compte dans l'application
Pré-condition	L'appareil mobile doit être connecté à l'internet
Scénario nominal	remplir les différents champs, puis valider
Post-condition	L'inscription avec succès
Scénario d'exception	Inscription échoué, message d'erreur doit s'afficher à l'utilisateur

Table 2. 2 Cas d'utilisation Inscription

- Cas d'utilisation Ajouter une publication

Cas d'utilisation	Ajouter une publication
Acteur	Utilisateur
Description	L'utilisateur publie des avis de décès
Pré-condition	L'utilisateur doit être authentifié
Scénario nominal	Remplir les champs nécessaires, puis valider
Post-condition	L'ajout avec succès
Scénario d'exception	L'ajout échoué, message d'erreur doit s'afficher à l'utilisateur

Table 2. 3 Cas d'utilisation Ajouter une publication

- Cas d'utilisation Ajouter une recommandation

Cas d'utilisation	Ajouter des recommandations
Acteur	Utilisateur + Admin
Description	L'utilisateur peut recommander des changements. L'admin valide la recommandation ou l'annule.
Pré-condition	Les deux acteurs sont authentifiés
Scénario nominal	L'utilisateur valide son recommandation, puis l'admin traite la et valide son résultat soit accepté la sinon ignore.
Post-condition	Recommandation valider/s'ignorer
Scénario d'exception	Requête échoué, message d'erreur doit s'afficher à l'utilisateur

Table 2. 4 Cas d'utilisation Ajouter recommandation

5. Diagrammes des séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans formalisme UML. Dans ce qui suit, nous présentons le diagramme de séquence selon le diagramme de cas d'utilisation dans notre système.

5.1. Authentification

Ce diagramme, est décrire le scénario du cas d'utilisation d'authentification. Dans un premier lieu, lorsque l'utilisateur accès à l'application, une requête est démarré pour vérifie si l'utilisateur a fait le login précédemment et quitter l'application sans déconnecté ou non. Si vrai, alors, l'application retourne la page principale à l'utilisateur, sinon elle retourne la page s'authentification. L'utilisateur saisie l'email et le mot de passe puis valide ses données en clique sur le bouton de connexion. Ensuite le système vérifie s'il s'agit bien d'un membre ou pas. La figure suivante illustre une description détaillée du ce scénario.

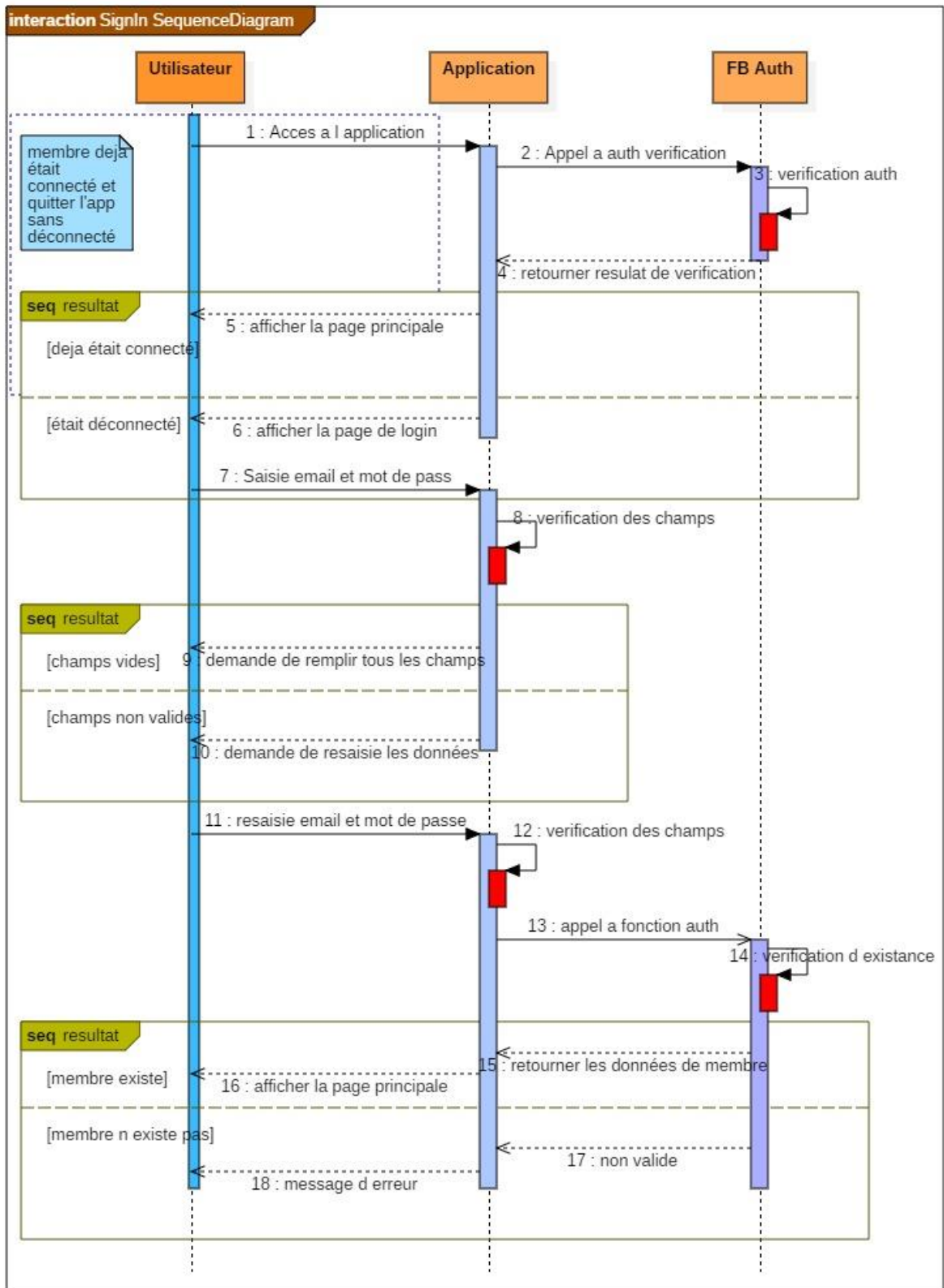


Figure 2. 3 Diagramme de séquence de l'authentification

5.2. Inscription

Ce cas est composé de deux parties. La première partie consacrée à la création d'un compte. Et la deuxième consacré à choisir le mode d'abonnement dans l'application.

- Partie d'inscription

Ce diagramme est décrit le scénario d'inscription. L'utilisateur demande au système de faire une inscription. Après d'afficher le formulaire, il remplit tous les champs puis, il enregistre les informations entrées dans la base de données. Le détail de scénario est illustré dans la figure suivante.

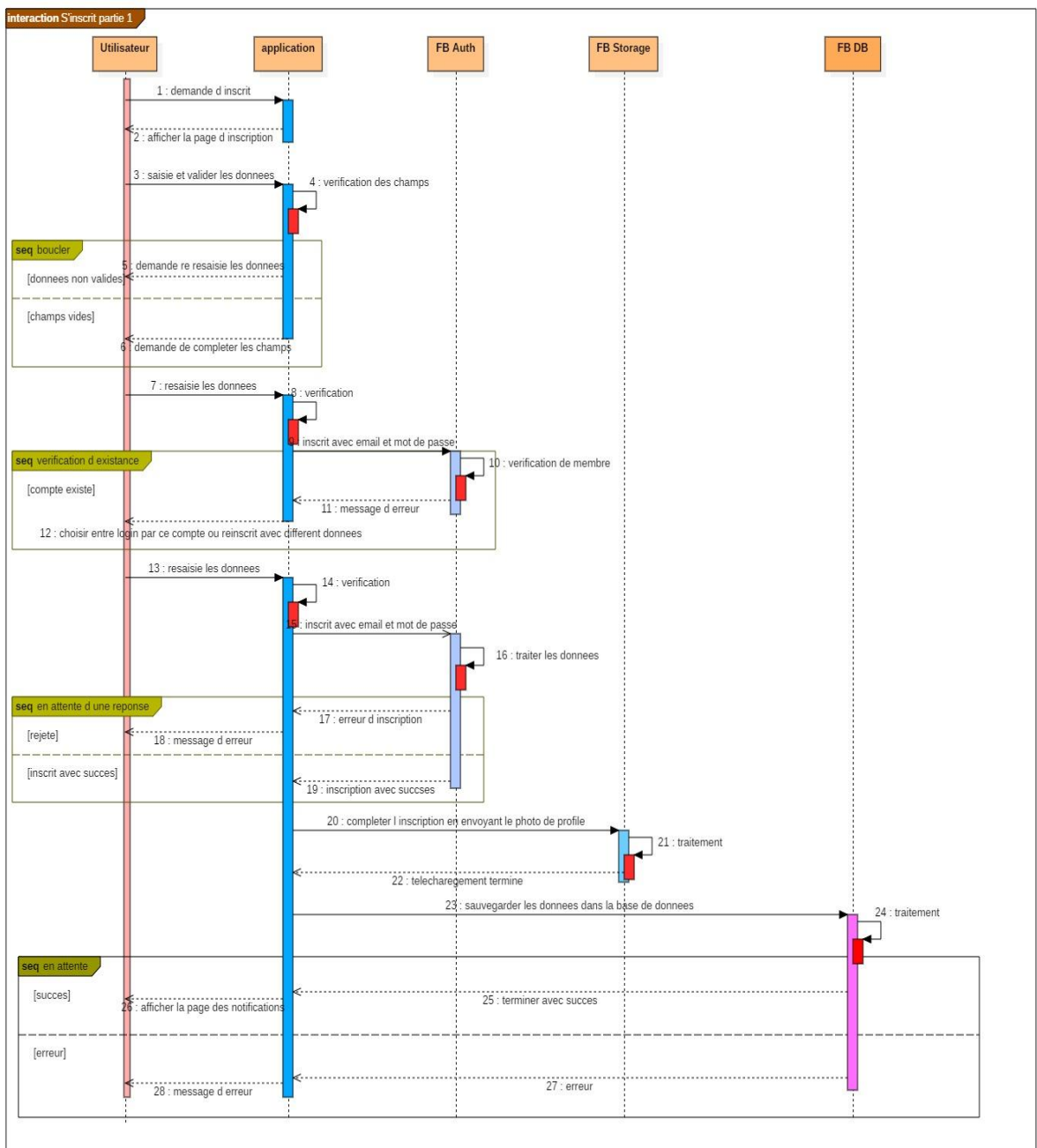


Figure 2. 4 Diagramme de séquence d'inscription

- Partie deux : choisir un mode d'abonnement

Just après compléter l'inscription, le système affiche la page d'abonnement à l'utilisateur. Dans le démarrage de cette page, il affiche POP-UP qui contient deux choix. Le premier choix est de recevoir tous les notifications, si l'utilisateur le sélectionner, le système complète l'inscription et affiche la page principale. Et le deuxième choix est de spécifier les notifications, si l'utilisateur le sélectionner, il continue l'inscription par choisir puis valider ses favoris, le système affiche la page principale après la validation. Le détail de scénario est illustré dans la figure suivante.

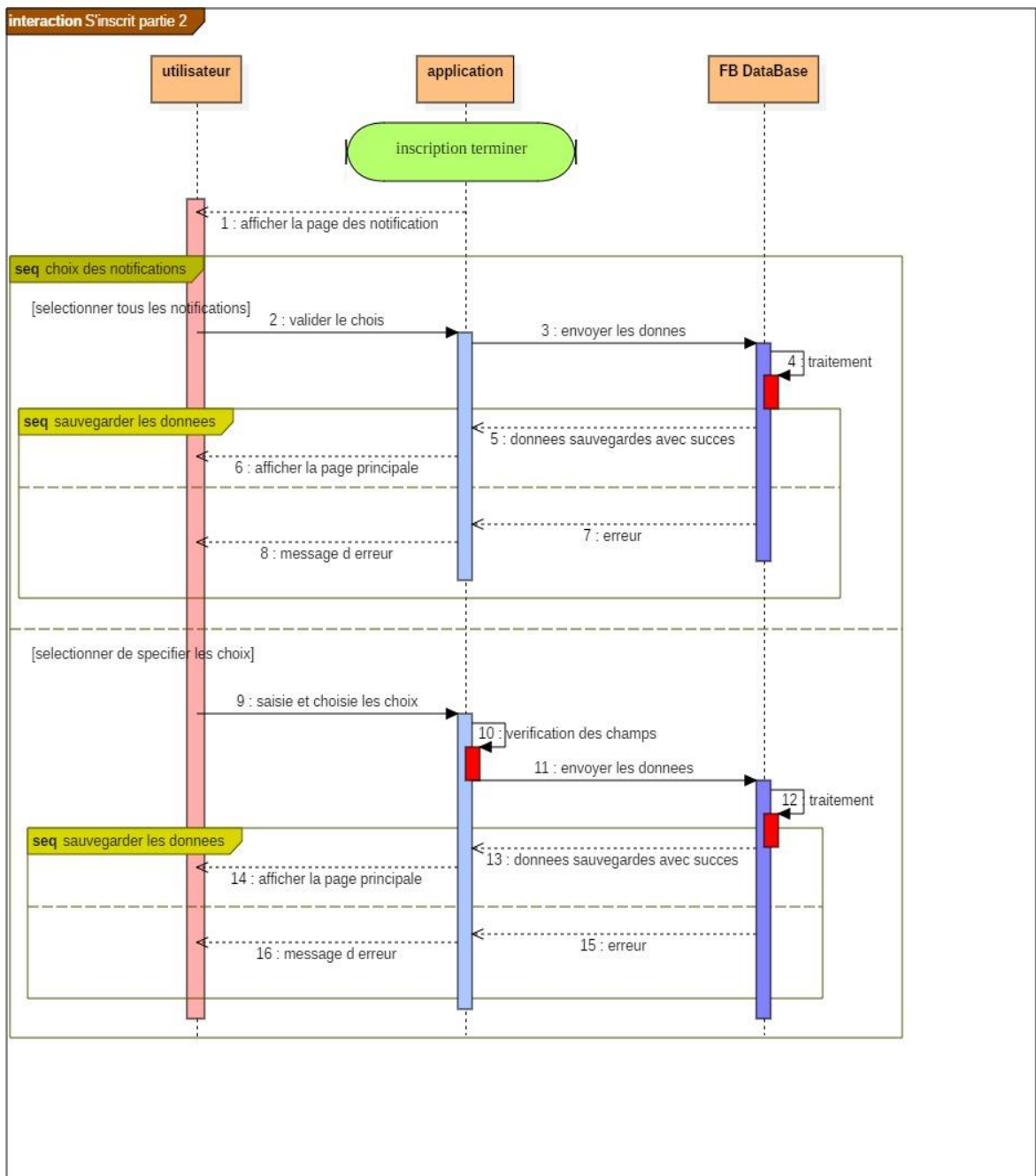


Figure 2. 5 diagramme de séquence Mode d'abonnement

5.3. Ajouter un avis de décès

Ce diagramme est consacré à l’ajout d’un avis de décès. Une fois l’utilisateur clique sur le bouton d’ajout, le système retourne POP-UP qui contient un formulaire à remplir. L’utilisateur remplit tous les champs obligatoires puis, il valide en cliquant sur le bouton d’ajout. Une fois les données ajoutées dans la base de données, la publication est affichée dans le journal de la page principale. La figure suivante illustre le détail.

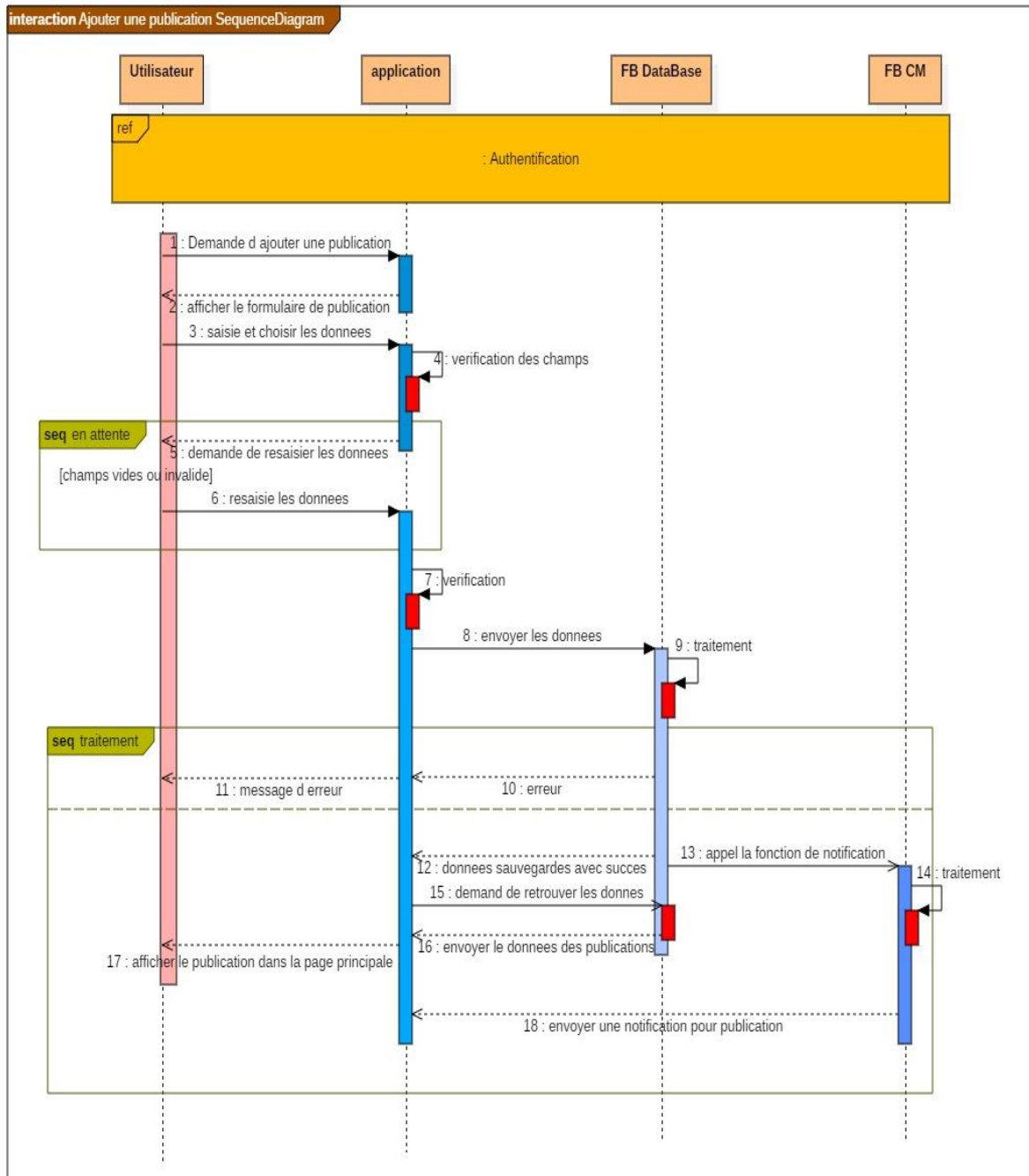


Figure 2. 6 Diagramme de séquence d’ajout une publication

6. Diagramme des classes

Un diagramme de classe se définit comme étant un ensemble de classes contenant des attributs et des opérations, reliées les unes aux autres par des relations et ceci en ayant des conditions de participation (cardinalités) ; il s'agit de la version UML de la base de données [14]. C'est le diagramme le plus important de la modélisation orienté objet. Le diagramme de classe retenu à la fin de conception est présenté dans la figure suivante :

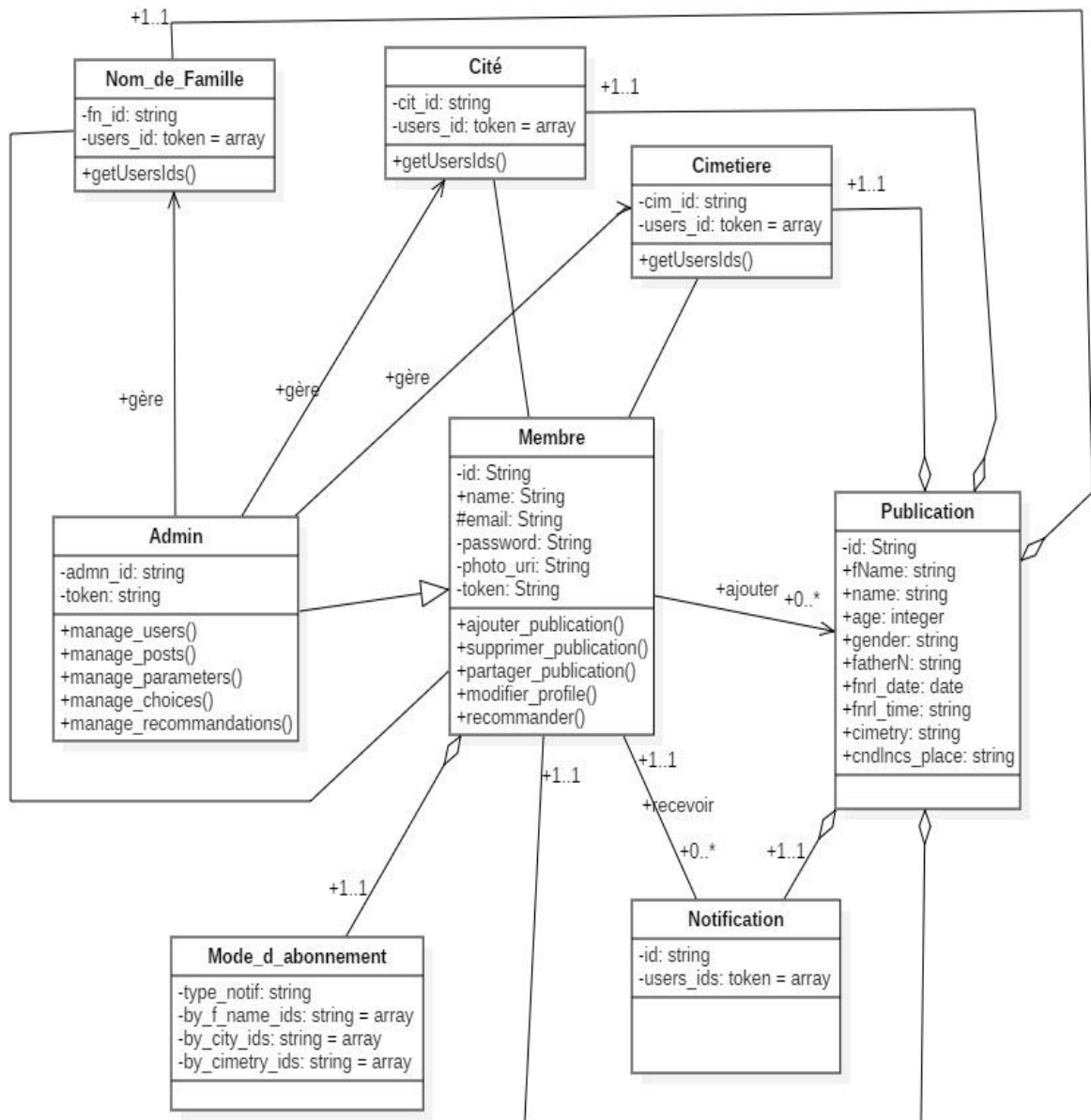


Figure 2. 7 Diagramme des classes

7. Conclusion

Dans ce chapitre, nous avons présenté les diagrammes UML nécessaires à la réalisation de notre application. Nous avons décrit d'abord les cas d'utilisation en recensant de manière textuelle toutes les interactions entre les acteurs et le système suivi par un diagramme des cas d'utilisation. Par ma suite, Nous avons complété cette description textuelle par des diagrammes des séquences. Et à la fin, définissant les relations entre les entités, nous sommes parvenus à concevoir le diagramme de classes. Ce qui nous permet de commencer le développement de l'application.

CHAPITRE 3

REALISATION

1. Introduction

Après avoir détaillé l'analyse et la conception de notre application, ce chapitre est consacré à la partie réalisation. On présentera d'abord l'environnement matériels et logiciels et quelques bibliothèques utilisées dans notre application, par la suite, on parlera à la phase d'implémentation de la base de données, et à la fin on présentera les différentes interfaces de l'application.

2. Environnement du travail

Les différents outils matériels et logiciels utilisés pour la réalisation de notre application est dans ce qui suit :

2.1. Environnement matériel

Pour la réalisation, on a utilisé un ordinateur portable DELL caractérisé par :

- Processeur : Intel core i7
- RAM :8 GO
- Disque dur : Hdd de 1 TO
- Système : Windows 10, 64 bits

2.2. Environnement logiciel

L'environnement de développement utilisé dans notre application est Android Studio, et différents APIs (Application Programming Interface) qui sont :

2.2.1. Android Studio

Android Studio est un environnement de développement intégré (IDE), créé par Google pour développer des applications sous Android. Il a été annoncé en 2013. Android Studio est basé sur logiciel IntelliJ IDEA de JetBrains, et il est disponible sous la licence Apache 2.0 librement [15]. Il permet d'éditer les fichiers de configurations (gradle et Manifest), les fichiers du code source (Java, Kotlin) et d'utiliser des bibliothèques et d'ajouter des fichiers assistants (fonts,

media...) pour faciliter à nous développons notre application. Nous avons utilisé Android Studio de version 3.3, et Java comme langage pour le code source.



Figure 3. 1 Logo officiel de Android Studio

2.2.2. JDK

Java development Kit (JDK) est un ensemble de bibliothèques logicielles de base du langage Java. Les outils avec lesquels le code Java peut être compilé et transformé en bytecode destiné à la machine virtuelle Java. [16]



Figure 3. 2 Logo de JDK

2.2.3. Node.js

Node.js est une plateforme de développement JavaScript open-source. C'est un langage JavaScript avec des bibliothèques permettant de réaliser des actions. Nous avons utilisé Node.js pour gérer l'envoi de notifications. [17]



Figure 3. 3 Logo de Node.js

2.3. Bibliothèques utilisées

Voici quelques bibliothèques que nous avons utilisé dans notre application :

2.3.1. CircleImageView

Cette bibliothèque permet d'afficher l'image sous forme d'un cercle.

2.3.2. Glide

Utilisé pour obtenir l'URI de photo de profile de la base de données, et charger le dans les interfaces.

2.3.3. RecyclerView

RecyclerView utilisé pour lister les publications tandis que l'application est en cours d'exécution dans les pages.

2.3.4. MaterialSpinner

Une beau spinner aide sur les choix.

2.3.5. Shimmer

Cette bibliothèque est utilisée tant qu'en attente de chargement les publications.

2.4. Architecture Client/Server

Notre application est conçue sous la forme d'une architecture client/server.

2.4.1. Client

Dans notre projet, le client s'agit de l'application cliente. Il a les droits pour ajouter et modifier ou charger ses données personnelles, et des taches supplémentaires.

2.4.2. Serveur

Tous les données dans notre application sont dans le serveur de Google (Firebase).

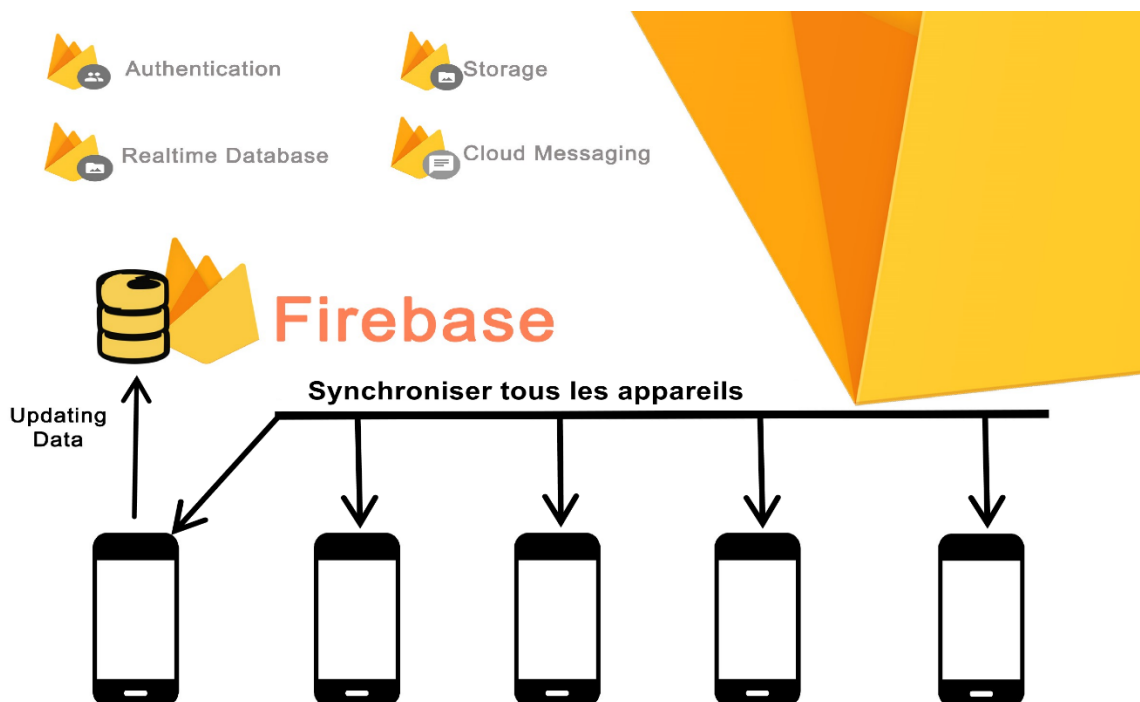


Figure 3. 4 Architecture Client / Server

3. Implémentation de la base de données

Puisque notre application est appartenue à la famille des réseaux sociaux, la meilleure solution pour la base de données est la mettre dans les Cloud, et pour cela, nous avons utilisé la base de donnée Firebase qui est de type No SQL.

3.1. No SQL

Le No SQL, pour "not only SQL", c'est-à-dire pas seulement SQL. Désigne les bases de données qui ne sont pas fondées sur l'architecture classique des bases de données relationnelles. Développé à l'origine pour gérer du grande data.

3.2. Firebase

Firebase est un BaaS (Backend-as-a-Service), qui a évolué pour devenir une plate-forme de développement d'applications de nouvelle génération sur la plate-forme Google Cloud [18]. Il propose d'héberger en NoSQL et en temps réel des bases de données, des contenus, l'authentification et encore les notifications. Lancé en 2011 sous le nom ENVOLVE, par Andrew Lee et par James Templin. Pour ajouter Firebase à notre projet dans Android Studio, il faut :

- Créer un compte Gmail dans Google.
- Enregistrer l'application dans la plateforme de Firebase en rempliant les champs nécessaires.

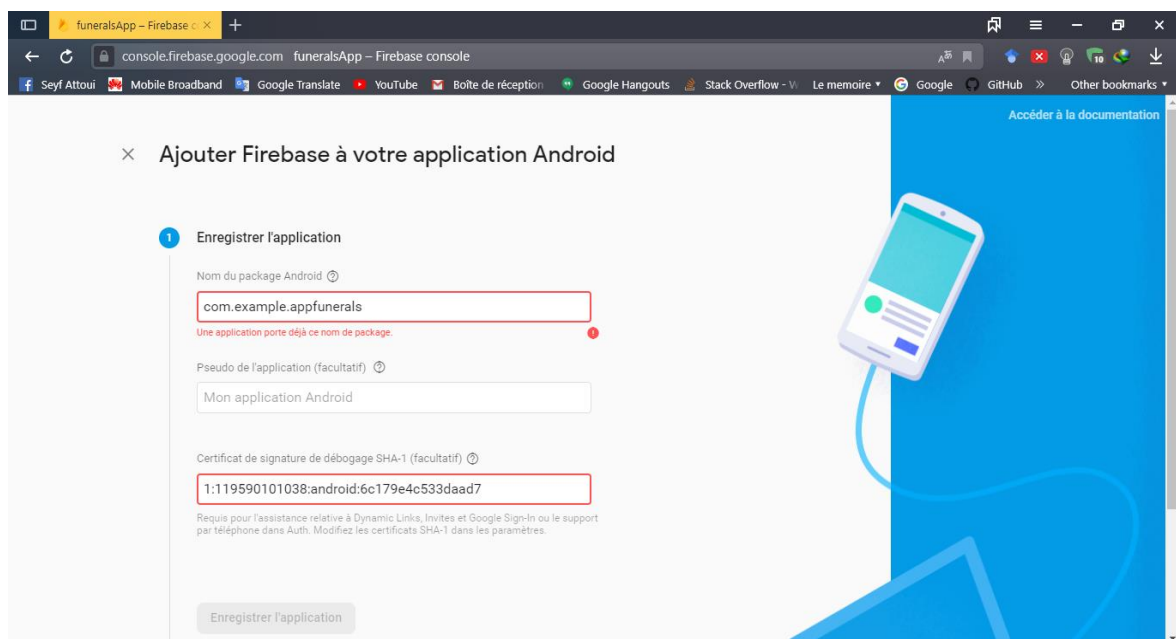


Figure 3. 5 Première étape de l'implémentation

- Télécharger le fichier contenant les services de Google, et le placer dans le dossier sous le nom « app » située dans le package de projet.

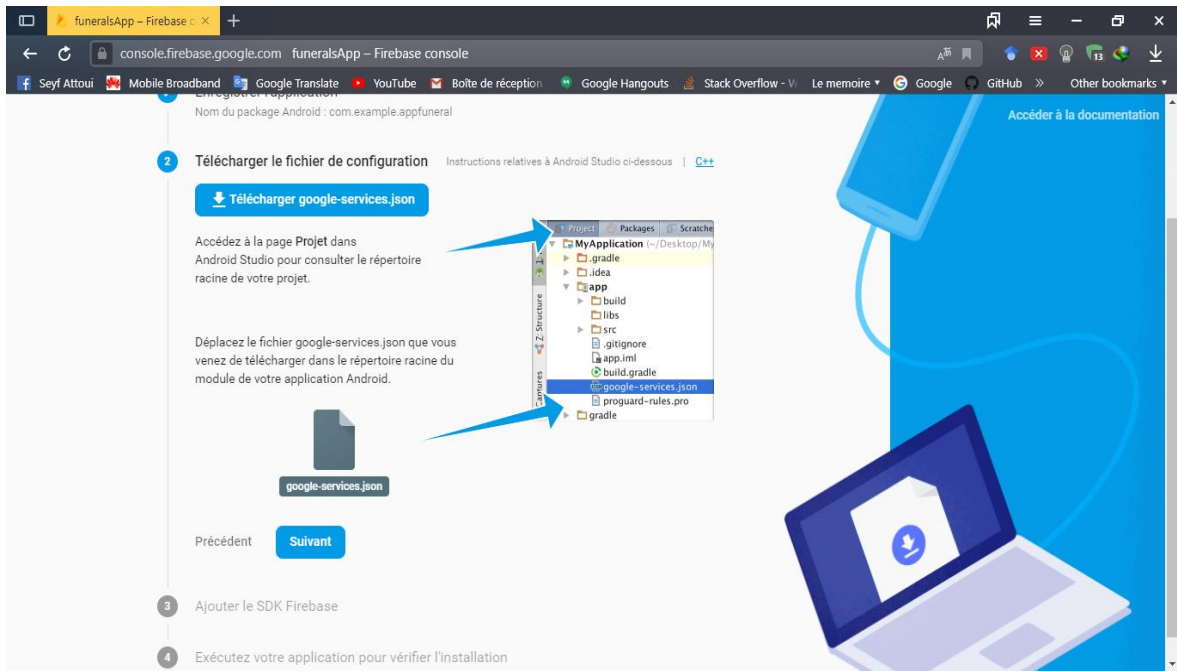


Figure 3. 6 deuxième étape de l'implémentation

- Ajouter le SDK de Firebase au niveau de projet et de m'application, puis appuyer sur « synchroniser » dans la barre qui apparaît sans l'IDE.

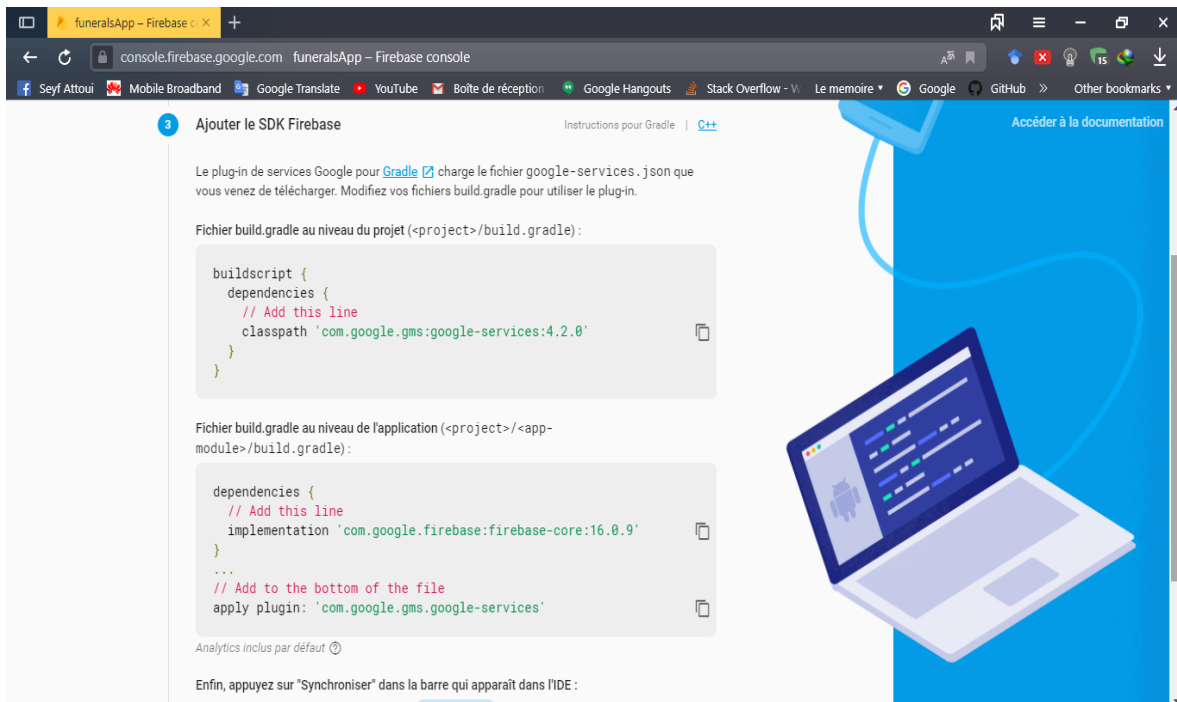


Figure 3. 7 Troisième étape de l'implémentation

- Exécuter l'application pour vérifier l'installation de SDK.

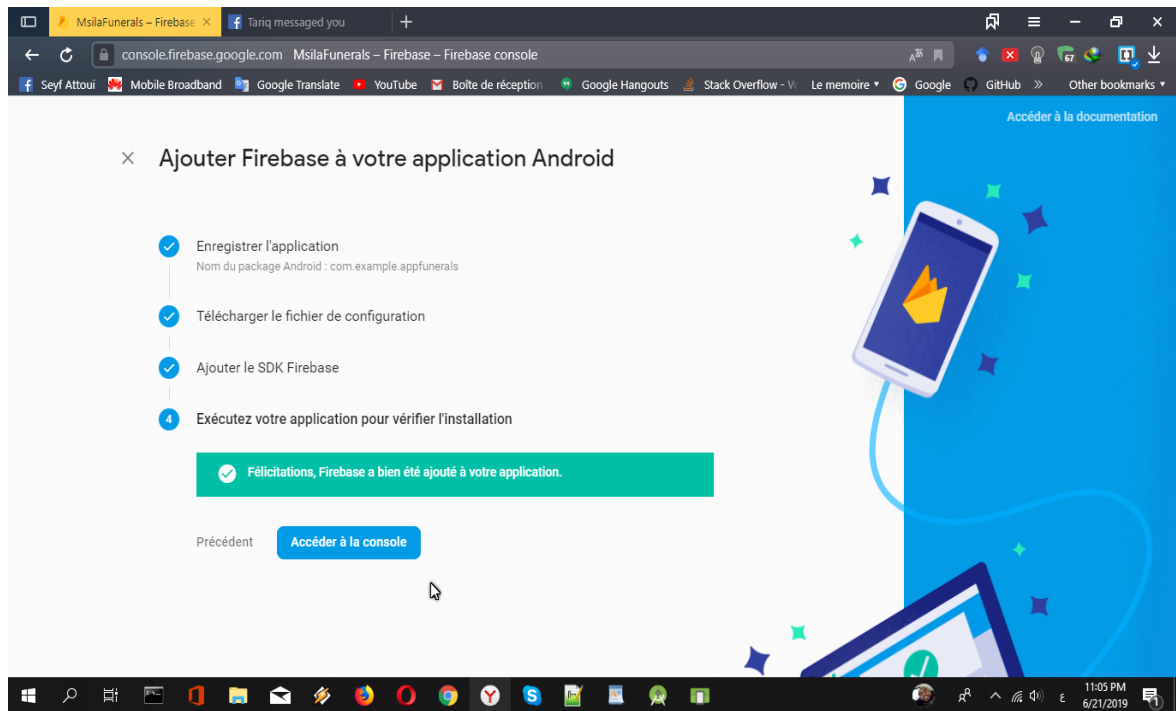


Figure 3. 10 Dernière étape de l'implémentation

- Enfin, accéder à la console de Firebase.

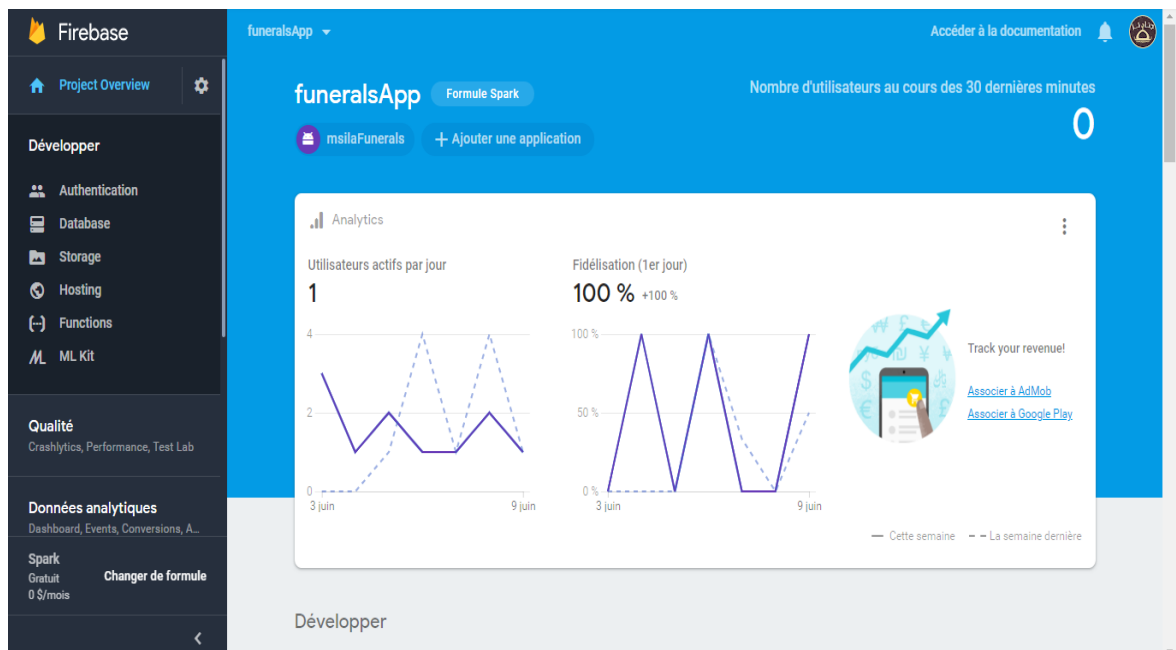


Figure 3. 11 Console de Firebase

3.2.1. 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 votre 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. [19]

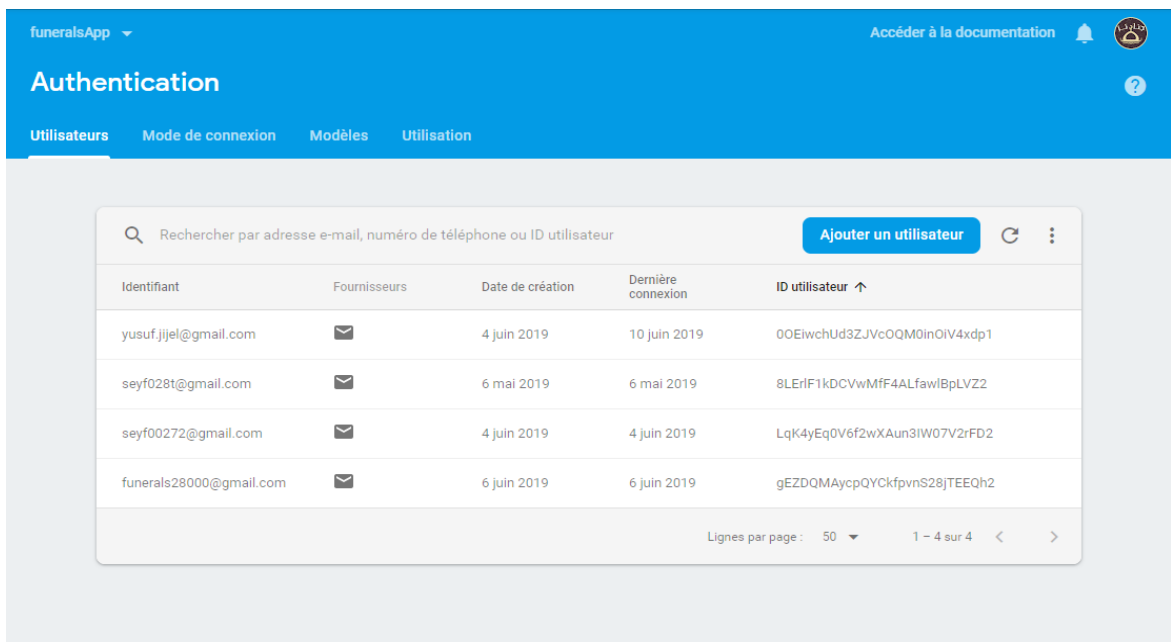


Figure 3. 12 Authentification Firebase

3.2.2. Firebase Data Base

Realtime Database 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 kits de développement logiciel (SDK) iOS, Android et JavaScript, tous vos clients partagent une instance de base de données en temps réel et reçoivent automatiquement les mises à jour avec les données les plus récentes. [19]

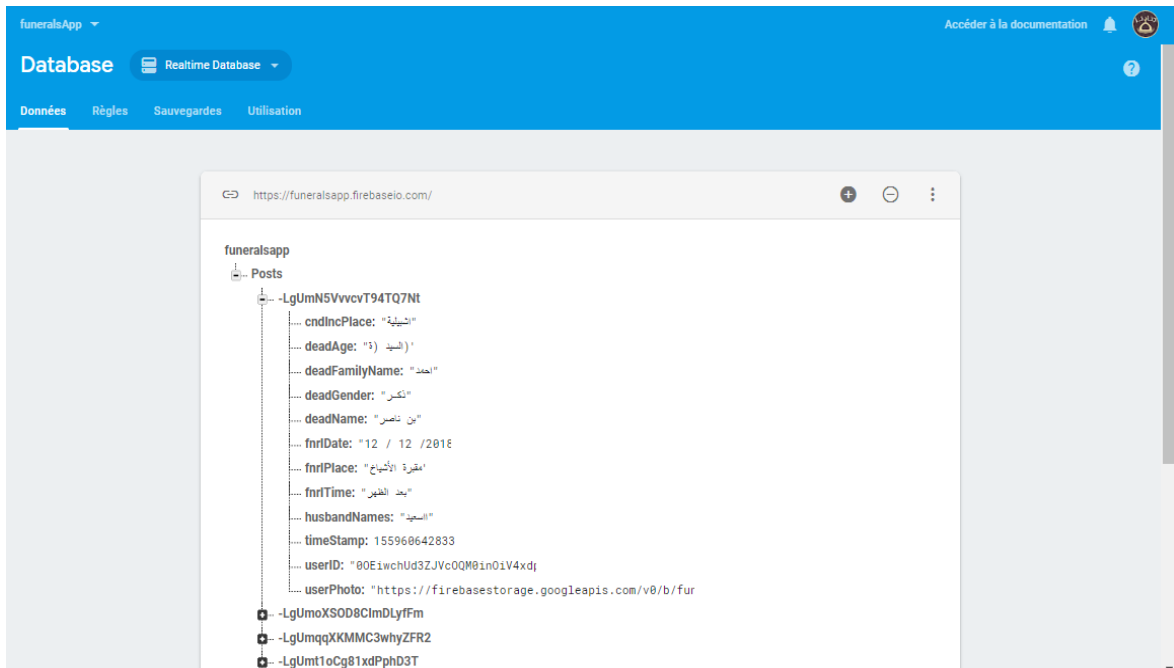


Figure 3. 13 Firebase Realtime Database

3.2.3. Firebase Storage

Cloud Storage for Firebase (stockage en nuage), est un service de stockage d'objets puissant, simple et économique, conçu pour Google. Cloud Storage est conçu pour les développeurs d'applications qui ont besoin de stocker et de diffuser du contenu généré par l'utilisateur, tel que des photos ou des vidéos. [19]

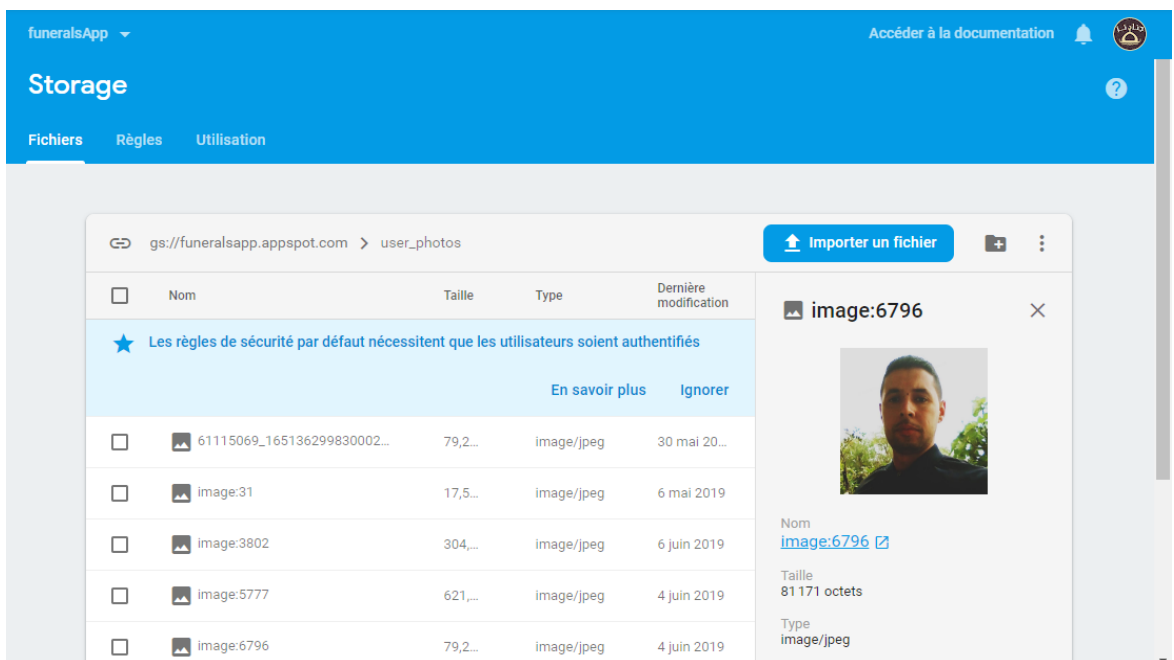


Figure 3. 14 Firebase Cloud Storage

3.2.4. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) est une solution de messagerie multiplate-forme qui permet de livrer des messages de manière fiable, sans frais. Il permet d'envoyer des messages de notification aux utilisateurs de l'application. [19]

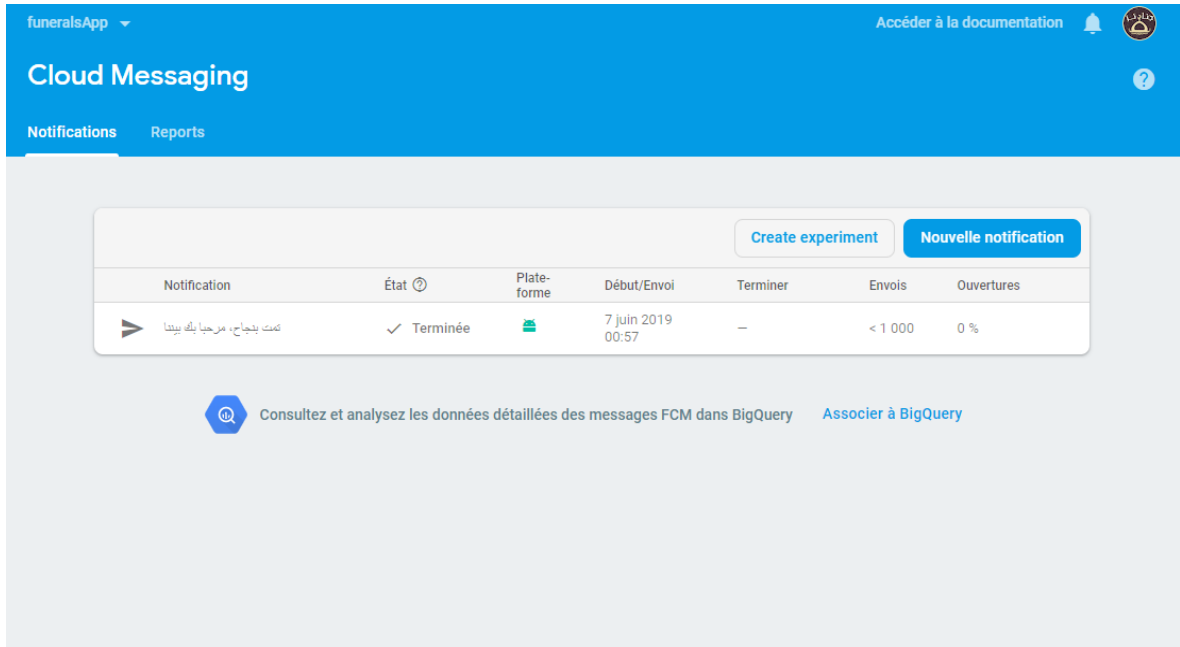


Figure 3. 15 Firebase Cloud Messaging

4. Présentation de l'application

Dans ce qui suit, nous présentons le fruit de notre travail en exposant les différentes interfaces de l'application. En commençant d'abord, par le logo de l'application.



Figure 3. 16 Logo de l'application

4.1. Les interfaces de l'application

4.1.1. SplashScreen

C'est l'interface de chargement de l'application, contenant le logo de l'application et un Progress Bar avec une animation de chargement dure 5 secondes.

4.1.2. Menu de connexion

Pour choisir le mode de connexion, il est possible de se connecter via un email, compte Facebook ou compte Google.



Figure 3. 17 SplashScreen



Figure 3. 18 Mode de connexion

4.1.3. Login

Pour les membres, qui sont déjà inscrits dans l'application.

4.1.4. S'inscrire

Pour créer un nouveau compte.

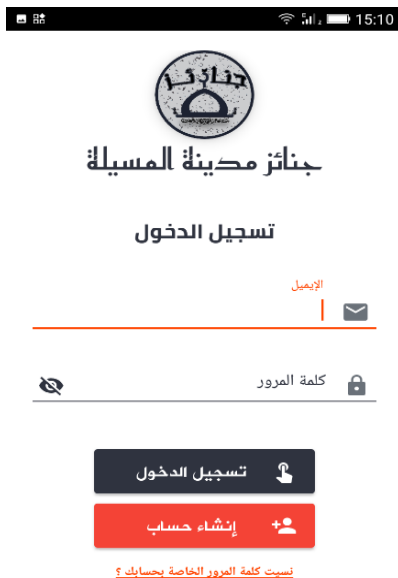


Figure 3. 19 Fenêtre de login



Figure 3. 20 Fenêtre d'inscription

4.1.5. Choisir type de notifications

Lorsqu'un utilisateur inscrit dans l'application pour la première fois, il est prêt à choisir le type de notifications des funérailles. Il peut choisir de recevoir tous les notifications, sinon spécifier quelques paramètres pour reçoit des notifications personnalisées.

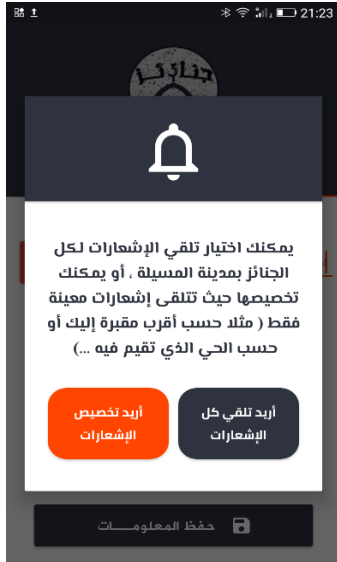


Figure 3. 21 Pop-up alerte



Figure 3. 22 Fenêtre de choix

4.1.6. La page principale

Contenant tous les publications de tous les utilisateurs.

4.1.7. Détails d'une publication

Contenant détails de publication. Il est possible de commenter la publication ou partager le sur Facebook.



Figure 3. 23 La fenêtre principale



Figure 3. 24 Fenêtre de détails

4.1.8. La page de profil

Contenant les informations de l'utilisateur et les notifications.

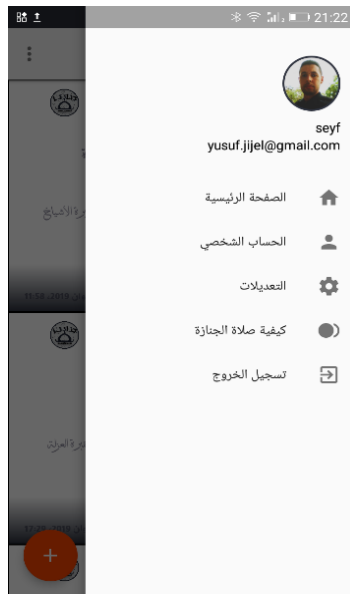


Figure 3. 25 Navigation drawer

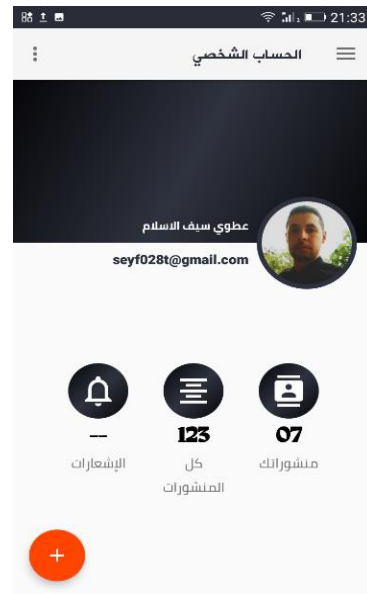


Figure 3. 26 Fenêtre de profil

4.1.9. Ajouter une publication

C'est une pop-up, contenant un formulaire à remplir avec les informations de décès pour publier la publication.

4.1.10. Paramètres

Les différents paramètres autorisés au l'utilisateur.



Figure 3. 27 Ajouter une publication

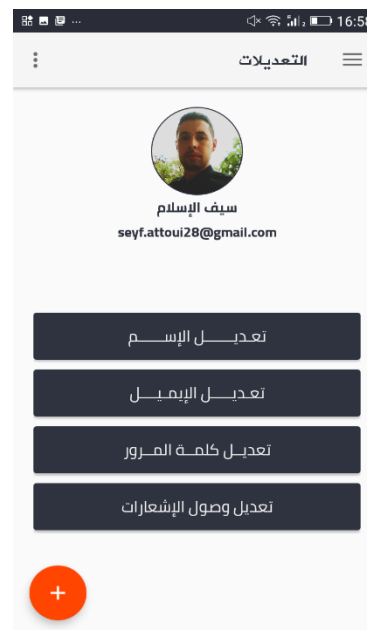


Figure 3. 28 Paramètres

4.1.11. Slide contenant des pages pour la Prière les funérailles

C'est une page supplémentaire, contenant quelques informations sur la prière des funérailles, la condoléances etc...

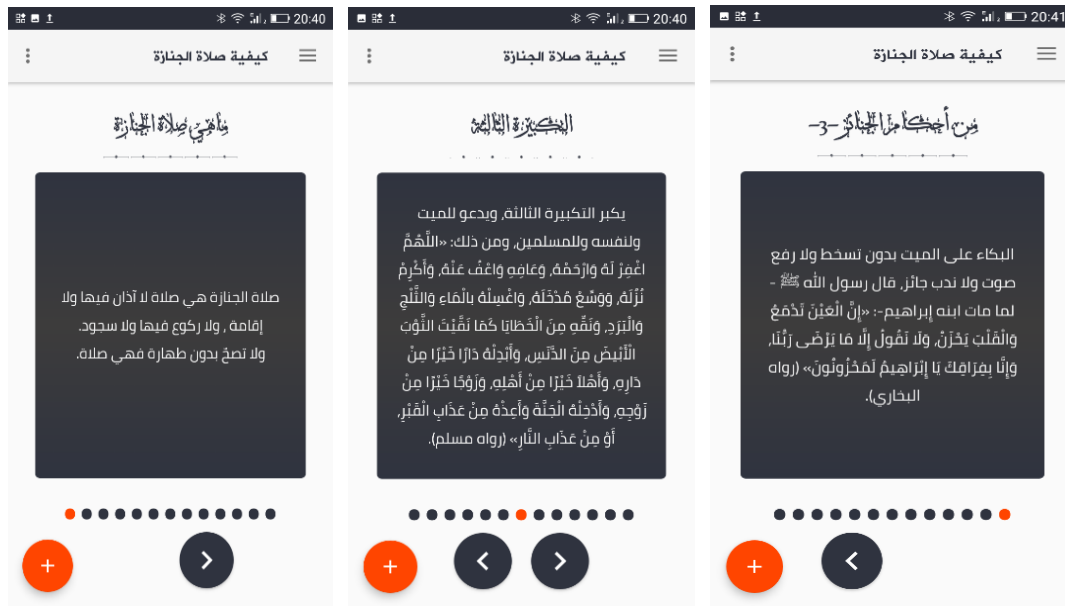


Figure 3. 29 Fenêtre supplémentaire

4.2. Interfaces supplémentaires pour l'admin

L'admin a une page supplémentaire, contenant les différents paramètres de gestion des différentes tâches.

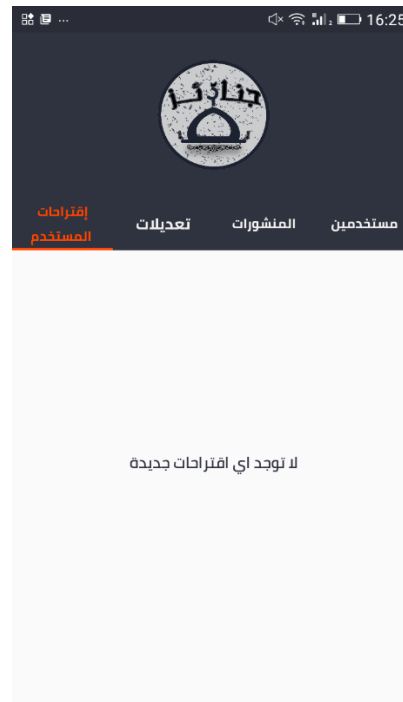


Figure 3. 30 Fenêtre de gestion pour l'admin

5. Conclusion

Dans ce dernier chapitre, nous avons présenté l'environnement de développement de notre application et les différents outils et bibliothèques utilisés. Par ma suite, nous avons présenté les différentes interfaces de notre application. Nous prenons au sérieux de respecter au maximum, les standards de l'IHM pour réaliser notre application.

Conclusion générale

Dans ce travail de projet de fin d'études, nous avons présenté les différentes étapes de réalisation d'une application mobile sous Android pour les funérailles de la ville de M'Sila. Nous avons commencé d'abord, par des généralités sur la programmation mobile sous Android. Nous avons également abordé en détails le problème de l'annonce des funérailles en générale et surtout dans la ville de M'Sila. Ensuite, nous avons présenté des diagrammes UML permettant de bien comprendre la structure statique et dynamique du système. Enfin, nous avons présenté les principaux outils qui nous ont permis de bien développer notre application tel que le langage Java sous Android Studio. Ainsi, le Firebase pour le développement de la base de données et la gestion des évènements. Nous avons aussi utilisé l'outil NodeJs pour programmation des fonctions et la spécification des notifications. Ces choix nous ont permis de réaliser une application répondant aux ces problèmes des funérailles dans la ville de M'Sila. Ce projet a été pour nous l'occasion d'améliorer et renforcer nos connaissances théoriques et pratiques en conception et en développement mobile. L'aspect concret de ce projet nous a plu car nous pouvions voir facilement le résultat de notre travail, ce qui a été très motivant.

À l'issue de ce projet, notre application est réalisée dans sa première version. Elle restera toujours ouverte à des perspectives d'amélioration. Nous envisageons d'ajouter une autre langue pour une très grande utilisation. Ainsi, nous envisageons de faire notre application couvrir tout le territoire national. Nous envisageons ainsi d'ajouter de nouvelles fonctionnalités selon les recommandations des utilisateurs.

Bibliographie

- [1] P. F, Étude sur les besoins de compétences dans le développement d'applications mobiles, Montréal: TechnoCompétences, 2013.
- [2] R. H, Introduction aux systèmes d'exploitation mobiles, 2013.
- [3] [En ligne]: [https://fr.wikipedia.org/wiki/IOS_\(Apple\)](https://fr.wikipedia.org/wiki/IOS_(Apple)). [Consulté le 30 12 2018].
- [4] [En ligne]: <http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.html>. [Consulté le 30 12 2018].
- [5] [En ligne]: <http://gs.statcounter.com/os-market-share/mobile/algeria>. [Consulté le 3 3 2019].
- [6] J. Kruger, «Opportunités d'entreprise avec Android,» Institut SupInfo, 2009.
- [7] «Phone Android,» [En ligne]: <https://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html#lhistorique-des-versions-dandroid-une-veritable-evolution>. [Consulté le 12 3 2019].
- [8] [En ligne]: https://fr.wikipedia.org/wiki/Android_Pie. [Consulté le 1 4 2019].
- [9] K. Mrabet et N. Trabelsi, Application de messagerie simple sur Android, Télécom SudParis, 2011.
- [10] [En ligne]: <http://igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/>. [Consulté le 15 3 2019].
- [11] «Developer Guides,» [En ligne]: <https://developer.android.com/guide>. [Consulté le 1 4 2019].
- [12] L. Audibert. [En ligne]: <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet#L1-4>. [Consulté le 19 4 2019].
- [13] A. Laurent. [En ligne]: <https://laurent-audibert.developpez.com/Cours-UML/?page=introduction-modelisation-objet#L1-4-3-a>. [Consulté le 19 4 2019].
- [14] R. P, UML 2 par la pratique, EYROLLES, 2009.
- [15] A. Louis, «Android Studio : une version 1.0 pour l'IDE de Google,» [En ligne]: <http://www.zdnet.fr/actualites/>. [Consulté le 22 12 2018].
- [16] «Java - Introduction. commentcamrche,» [En ligne]: <http://www..> [Consulté le 22 12 2018].
- [17] «What is Node.js? - Definition from WhatIs.com,» [En ligne]: <https://whatis.techtarget.com/definition/Nodejs>. [Consulté le 13 4 2019].
- [18] «What is firebase?» [En ligne]: <https://howtofirebase.com/what-is-firebase-fcb8614ba442>. [Consulté le 22 4 2019].

[19] «Firebase Guide | Firebase,» [En ligne]: <https://firebase.google.com/docs/guides>. [Consulté le 22 4 2019].

ملخص

يقدم هذا العمل الذي قمنا به خلال مشروع التخرج للحصول على شهادة ماجستير أكاديمي في الإعلام الآلي. حيث عملنا على تطوير تطبيق للهاتف الذكي ذو نظام التشغيل أندرويد مخصص لإعلانات جناز مدينة المسيلة. بدأنا أولاً بتحديد الأساليب المتوفرة في تلقي أخبار الجناز. ثم قمنا باقتراح حلنا المتمثل في تطوير تطبيق اندرويد لإعلانات جناز مدينة المسيلة. بالنسبة للتطوير، استعملنا منظومة التطوير المدمجة أندرويد ستوديو. كذلك استعملنا خدمات الفايبريز التي سهلت علينا تطوير التطبيق. استعملنا ايضاً، بيئة NodeJs من اجل برمجة تخصيص الاشعارات.

كلمات مفاتيح : تطبيق أندرويد، الفايبريز، جناز، إعلان وفاة.

Abstract

This dissertation is realized as a final studies project to obtain the diploma of master degree in computer science. it consists of developing an Android mobile application for M'Sila city death announces. We began with selection of available capabilities used to declare a death. Then, we propose our solution, which is to develop an Android application for M'Sila city death announces in real-time. For implementation, we used the Android Studio integrated development environment. Also, we implement Firebase services to facilitate our development. We used NodeJs to programming functions to specify notifications.

Keywords: *Android App, Firebase, funerals, death announce.*

Résumé

Ce mémoire a été réalisé dans le cadre du projet de fin d'études pour l'obtention du diplôme de master en informatique. Il consiste à développer une application mobile sous Android pour l'annonce des funérailles de la ville de M'Sila. Nous avons commencé par une étude critique des moyens disponibles pour les avis des décès dans la ville de M'Sila. Ensuite nous avons proposé notre solution qui consiste à réaliser une application Android pour l'annonce des décès en temps réel. Pour la réalisation, nous avons utilisé l'environnement de développement intègre Android Studio. Ainsi, nous avons implémenté les services de Firebase qui nous a facilité le développement. Nous avons aussi utilisé l'outil NodeJs pour programmer des fonctions pour spécifier les notifications.

Mots clés : *Android App, Firebase, funérailles, avis de décès.*