

MA-004-323-1

République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et la Recherche Scientifique

Université Saad DAHLAB Blida



Faculté des Sciences
Département d'informatique

Conception et développement d'une application de guide touristique

En vue d'obtenir le diplôme de Master

Mémoire présenté par
HAMZA Abdelmalek
MANSOURI Abdessamed

Filière : Informatique
Spécialité : Ingénierie des logiciels

Promotrice : Mme. MILOUD AOUIDATE Amal
Encadreur : M. BOUZARARI Khaled

Président de jury : Tabahime Messine
Promotion : 2015/2016

MA-004-323-1

المُلخَص

الهدف الرئيسي من هذا العمل هو القيام بإنشاء تطبيق دليل سياحي على منصة أندرويد. هذا التطبيق يسمح للمسافرين بالبحث بسهولة ويتم إبلاغهم بجميع المعلومات الضرورية لرحلاتهم (رحلات الجوية، فنادق، مطاعم، مواقع السياحية، توقعات الطقس، الساعة العالمية، تحويل العملات، اقتراح أرخص (الرحلات الجوية، الفنادق، المطاعم)، المذكرة الذكية و الماسح الضوئي لتذاكر الجو. كلمات البحث : الاندرويد، السياحة، السفر، رمز الاستجابة السريعة.

Abstract

The main objective of this work is to implement a mobile solution based on Android platform of a touristic guide. This application will allow travelers to find easily and to be informed of all information that is necessary for their trip (flights, hotels, restaurants, touristic spots, weather forecast, world clock, currency converter, proposal of the cheapest (flights, hotels and restaurants), an intelligent reminder and air tickets scanner...).

Keywords : android, tourism, travel, QR code.

Résumé

L'objectif principal de notre PFE consiste à implémenter une solution mobile basée sur la plateforme Android d'un guide touristique.

Cette Application permette aux voyageurs de chercher d'une manière facile et d'être informer de toutes les informations qui sont nécessaire pour leur voyage (vols, hôtels, restaurants, sites touristiques, prévision météorologie, horloge mondiale, convertisseur de devises, la proposition des meilleur prix des (vols, hôtels et restaurants), un rappel intelligent et un scanner des billets d'avion...).

Mots clés : android, tourisme, voyage, Code QR.

Dédicace d'Abdelmalek

A

Ma chère mère Farida, pour ses sacrifices depuis qu'elle m'a mis au monde, et qui m'a aidé surtout dans les moments difficiles, Mon père Abdellah, qui m'a toujours soutenu et guidé pour affronter les difficultés de la vie, Mes frères Issam, Ikram et Ahmed qui ont toujours été présents et à mes côtés, Mes deux grands-mères et Mes deux grands-pères

Mlle. AMROUCHE Amina pour l'aide morale et les conseils, Mon oncle Samir et sa femme et ses petites filles Loubna, Bouchra et Rama et Toute ma famille.

Mon binôme Abdessamed MANSOURI, Mes collègues, Toute la promotion de l'année scolaire 2015-2016 A tous ceux qui sont proches de mon cœur et dont je n'ai pas cité les noms et surtout à tous les martyrs d'hier et d'aujourd'hui de notre chère Monde Arabe et Islamique.

Que je dédie ce modeste travail.

Dédicace d'Abdessamed

Au début, je tiens à remercier notre Dieu, le tout-puissant, pour sa clémence, sa miséricorde et notre guidage vers la droiture, aussi pour sa bénédiction à mon âge et à ma vie.

A toute ma famille en commençant par mes parents, Salah et Rekia, mes deux frères Mohammed et Touhami, mes deux sœurs Hiba et Bouchera, mes deux belles-sœurs Nadia et Fatimazohra, mes deux nièces Meriem et Rawnek Imane, ma deuxième mère, ma tante Yassina et tous mes cousins, sans oublier la perle Merrill à la mémoire.

Bien aussi à tous mes chères amis que je les ai rencontré dans ma vie ou dans mon parcours scolaire sans aucune exclusivité et spécialement à mon binôme Abdelmalek HAMZA que je lui souhaite du bonheur et plus de succès dans sa vie.

Remerciement

Au terme de ce travail, nous tenons à remercier tout d'abord notre encadreur Mme. MILOUD-AOUIDATE Amal de nous avoir guidés tout au long de ce travail, pour ses conseils, patience et aide.

Nous tenons à remercier M. BOUZARARI Khaled et M. SIKADDOUR Boulem pour ses suivis tout le long de cette période et qui a veillé au bon déroulement du projet.

Nous remercions aussi Mme. BOUTALEB Faiza directrice de l'administration générale par l'intérim de la société ANPT pour son accueil et son aide.

Nos vifs remerciements vont à tous les Profs au niveau de département de l'informatique pour leurs soutiens, leurs disponibilités tout au long de nos cinq années de spécialité ainsi que tout le staff de département.

Nous tenons à remercier M. MENGOUUD Nadjib qui nous a beaucoup aidés dans le bon déroulement de notre projet.

Nous remercions tous ceux qui nous ont aidés de près ou de loin à faire ce travail. Nos profonds remerciements et nos gratitude vont aussi aux membres du jury, pour leur accord de juger notre travail.

Sommaire

Introduction Générale	14
Chapitre I : Présentation Générale	
1. Introduction	17
2. Cadre du travail.....	17
3. Présentation de l'organisme d'accueil.....	17
3.1. Historique	17
3.1.1. L'agence et la ville nouvelle Sidi Abdallah	18
3.2. Les missions et les activités de l'ANPT	18
3.3. Les objectifs de l'ANPT	19
3.3.1. Les objectifs spécifiques du Cyber parc.....	19
3.3.2. Objectifs relatifs à la grappe des TIC.....	19
3.3.3. Objectifs de développement économique général	20
4. Présentation du projet	20
4.1. Objectifs de l'application	21
4.2. Approches du travail	21
5. Conclusion	22
Chapitre II : Etude de l'existant	
1. Introduction.....	24
2. Les solutions mobiles disponibles sur le marché.....	24
2.1. KAYAK	24
2.2. Hotel tonight	24
2.3. Expedia	25
2.4. Uber	25
2.5. Tripadvisor	26
3. Critiques de l'existant	26
4. Conclusion	27
Chapitre III : Etude théorique et choix techniques	
1. Introduction	29
2. Etude Préliminaire	29

2.1. Smartphones et systèmes d'exploitation pour mobiles	29
2.2. Le choix de la plateforme	31
3. Android	31
3.1. Définition	31
3.2. Le SDK Android	32
3.3. L'émulateur	33
3.4. Une architecture autour du noyau Linux	33
3.5. Les versions d'Android	35
3.6. Caractéristiques d'Android	36
3.7. Les avantages d'Android	37
4. Structure du Projet Android	37
4.1. Architecture basique d'une application Android	37
4.2. Présentation d'un projet Android	38
5. Conclusion	39

Chapitre IV : Analyse, spécification et Conception

1. Introduction	41
2. Présentation de la démarche utilisée	41
2.1. Le cycle de vie	41
3. Modèle en cascade	41
3.1. Expression des besoins	42
3.2. Analyse	42
3.3. Conception	43
3.3.1 Méthode de conception	43
3.4. Implémentation	44
3.5. Tests	44
4. Présentation de l'UML	44
4.1. Pourquoi Modéliser avec UML ?	44
4.2. Les diagrammes	45
4.2.1. Diagrammes structurels ou statiques	45
4.2.2. Diagrammes comportementaux	45
4.2.3. Diagrammes d'interaction ou dynamiques	45

5. Expression des besoins	46
5.1. Les besoins fonctionnels	46
5.2. Les besoins non fonctionnels	46
6. Diagramme de cas d'utilisation	47
6.1. Description	47
6.1.1. Identification des acteurs et de cas d'utilisation	47
6.1.2. Description des acteurs	48
6.3. Diagramme de cas d'utilisation général	48
6.4. Cas d'utilisations détaillés	48
6.4.1. Cas d'utilisation de « Lancer une recherche »	48
6.4.2. Cas d'utilisation de « Utiliser un outil »	54
7. La conception de l'application	57
7.1. Les styles architecturaux utilisés	57
7.1.1. Architecture multicouche	57
7.1.2. Architecture client-serveur	58
7.2. Diagrammes de séquence	60
7.2.1. Le diagramme de séquence (connectés).....	61
7.2.2. Diagramme de séquence de « Lister les rappels »	63
7.2.3. Diagramme de séquence de « Ajouter un rappel »	64
7.2.4. Diagramme de séquence de « Modifier un rappel »	65
7.2.5. Diagramme de séquence de « Supprimer un rappel »	66
7.2.6. Diagramme de séquence de « Contacter la société ».....	67
7.2.7. Diagramme de séquence de « Convertir l'argent »	67
7.3. Diagrammes de classe	68
7.3.1. Diagramme de class de l'application	68
7.3.2. Diagramme de classe de la base locale de données	69
7.3.2.1 Description des champs	69
7.4. Isolation des couches	70
8. Conclusion	71
Chapitre V : Réalisation	
1. Introduction	73

2. Environnement de travail	73
2.1. Environnement matériel.....	73
2.2. Environnement logiciel	73
2.2.1. La plateforme « Android»	73
2.2.1.1. Présentation	73
2.2.2. Jersey	74
2.2.2.1. Présentation	74
2.2.2.2. Fonctionnalité	74
2.2.3. La bibliothèque Google Guava	75
2.2.4. La bibliothèque Joda Time	75
2.2.5. La bibliothèque Material Design Support Library	76
2.2.6. FlightStats	76
2.2.7. Hotwire	76
2.2.8. Foursquare	77
2.2.9. OpenWeatherMap	77
2.2.10. OpenExchangeRates	77
2.2.11. Amadeus	78
3. Interfaces de l'application	78
4. Conclusion	98
Conclusion Générale	99

La liste des figures

Figure 1. Le logo de l'ANPT	17
Figure 2. Le logo de KAYAK.....	24
Figure 3. Le logo de Hotel tonight	25
Figure 4. Le logo d'Expedia	25
Figure 5. Le logo d'Uber	26
Figure 6. Le logo de Tripadvisor	26
Figure 7. Le logo de l'Android	32
Figure 8. Les versions d'Android	36
Figure 9. Cycle de vie selon le modèle en cascade	42
Figure 10. Diagramme de cas d'utilisation de l'application.....	48
Figure 11. Architecture multicouche du client	58
Figure 12. Architecture globale de l'application	60
Figure 13. Diagramme de séquences des cas d'utilisation connectés	62
Figure 14. Diagramme de séquence de cas d'utilisation de « Lister les rappels »	63
Figure 15. Diagramme de séquence de cas d'utilisation « Ajouter un rappel »	64
Figure 16. Diagramme de séquence de cas d'utilisation de « Modifier un rappel »...	65
Figure 17. Diagramme de séquence de cas d'utilisation « Supprimer un rappel » ...	66
Figure 18. Diagramme de séquence de cas d'utilisation «Contacter la société»	67
Figure 19. Diagramme de séquence de cas d'utilisation «Convertir l'argent».....	67
Figure 20. Diagramme de classe de l'application sous Android	68
Figure 21. Diagramme de classe de la base locale de données	69
Figure 22. Le logo de l'Android Studio	73
Figure 23. Le logo de FlightStats	76
Figure 24. Le logo de Hotwire	76
Figure 25. Le logo de Foursquare	77
Figure 26. Le logo d'OpenWeatherMap	77
Figure 27. Le logo d'OpenExchangeRates	78
Figure 28. Le logo d'Amadeus	78
Figure 29. Le splashscreen	79

Figure 30. L'accueil	79
Figure 31. La recherche des vols par vol, par aéroport ou par route	80
Figure 32. Le résultat de la recherche par vol	81
Figure 33. L'interface scanner des billets d'avion	82
Figure 34. La recherche des hôtels simple ou par offres	83
Figure 35. Le résultat de la recherche simple des hôtels	84
Figure 36. La recherche des places par la plus proche ou par la plus populaire	85
Figure 37. Le résultat de la recherche des places les plus proches	86
Figure 38. Le résultat de la recherche des places les plus populaires	87
Figure 39. La recherche des moyens de transport : voiture à louer ou un train	88
Figure 40. Le résultat de la recherche des places les plus populaires	89
Figure 41. Le résultat de la recherche des trains	90
Figure 42. La recherche des infos supplémentaires : météo, temps, devises.....	91
Figure 43. Le résultat de la recherche de la prévision de météo	92
Figure 44. Le résultat de la recherche de convertisseur de temps	93
Figure 45. Le résultat de la recherche de convertisseur de devises	94
Figure 46. L'interface paramètre de l'application	95
Figure 47. L'interface rappel	96
Figure 48. L'interface ajouter ou modifier un rappel	97
Figure 49. L'interface contacté nous	98

La liste des tableaux

Tableau 1. Raffinement de cas d'utilisation « Rechercher les vols »	49
Tableau 2. Raffinement du cas d'utilisation « Rechercher les hôtels »	50
Tableau 3. Raffinement du cas d'utilisation « Rechercher des spots touristiques »	51
Tableau 4. Raffinement du cas d'utilisation « Recherche d'un service de location de voitures »	52
Tableau 5. Raffinement du cas d'utilisation « Recherche d'un service de Taxi »	52
Tableau 6. Raffinement du cas d'utilisation « Rechercher des trains »	53
Tableau 7. Raffinement du cas d'utilisation « S'informer de la météo »	55
Tableau 8. Raffinement du cas d'utilisation « Convertir un temps »	55
Tableau 9. Raffinement du cas d'utilisation « Consulter le taux d'échange »	56
Tableau 10. Raffinement du cas d'utilisation « Convertir de l'argent »	56
Tableau 11. Raffinement du cas d'utilisation « Contacteur la société »	57
Tableau 12. Description des champs de la base locale des données.....	57

La liste des abréviations

- R&D : Recherche et Développement.
- TIC : Technologie de l'Information et de Communication.
- UML : Unified Modeling Language.
- OS : Operating System.
- OHA : Open Handset Alliance.
- MID : Mobile Internet Device.
- SDK : Software Development Kit.
- VM : Virtual Machine.
- OOAD : Object-Oriented Analysis and Design.
- DDD : Domain-Driven Design.

Introduction Générale

Dans le cadre de notre préparation du diplôme de Master M2 parcours Informatique, nous avons été amenés à effectuer notre projet qui consiste à étudier et de développer une application mobile sous Android.

Avec l'avancée et l'émergence des technologies mobiles, les développements embarqués sont de plus en plus demandés sur le marché. Avoir un Smartphone est devenu incontournable pour l'homme contemporain qui est obligé de suivre et de s'adapter avec les nouvelles technologies et les voyages de nos jours n'ont plus la même signification qu'autrefois. Les gens sont en quête d'un enrichissement personnel et culturel. Les destinations étrangères sont prisées et le touriste n'est pas cet individu un peu rustre et sans curiosité que l'on dépeint si souvent. C'est un voyageur complexe qui transporte avec lui ses désirs et ses rêves. Ses façons de voyager en disent long sur notre société et notre époque.

D'où est née l'idée de développer une application utile, à installer sur les Smartphones permettant de suivre les vols tout en gardant son téléphone dans sa poche, puis d'être informé de tout ce qui passe dans l'aéroport et aussi chercher un hôtel pour passer la nuit et des sites touristiques pour se promener et connaître le temps et le taux de change de n'importe quel endroit dans le monde, ainsi que toutes les informations utiles aux voyageurs.

Mis à part le développement proprement dit de l'application, la première étape consistait à se familiariser avec l'environnement Android, puis de choisir les outils conviviaux et envisageables à l'aboutissement du projet. Par la suite, nous entamons la modélisation et le développement de l'application.

Ensuite, chercher les serveurs web pour pouvoir échanger les données avec notre application grâce à une requête HTTP que nous avons obtenu à partir des fournisseurs spécialisés dans le domaine aéronautique, hôtellerie, météo, taux de change, etc.

Ce projet peut ainsi être subdivisé en quatre parties. La première consistera à la présentation de l'entreprise d'accueil, le cadre de notre projet et ces objectifs principaux.

La seconde partie sera consacrée à l'étude de l'existant dans le domaine de tourisme et voyage et la critique des solutions existant et dans la troisième partie on va donner une vue globale sur les plateformes mobile, et on va préciser notre choix sur la plateforme qu'on va travailler sur elle pour réaliser notre projet.

La troisième partie sera réservée à l'analyse et la spécification des besoins et la conception de notre projet pour bien détailler les différents éléments de l'application et mieux comprendre le travail demandé en dégageant les besoins des différents utilisateurs et la quatrième et la dernière partie qu'on appelle réalisation on va présenter l'environnement matériel et logiciel de notre projet ainsi que les interfaces graphiques de l'application.

En conclusion, nous mentionnons les différents atouts de ce projet et les perspectives d'améliorations possibles.

Chapitre I :
Présentation
Générale

1. Introduction

Ce chapitre sera consacré à la présentation de la structure d'accueil, et aussi la présentation du cadre du projet et la problématique puis l'énumération des étapes de travail à réaliser pour achever ce projet.

2. Cadre du travail

Ce stage s'inscrit dans le cadre d'un projet de fin d'études pour l'obtention d'un diplôme de Master en informatique de l'Université Saad Dahlab de Blida. Notre stage a été effectué au sein d'une Société étatique "ANPT". Le sujet est intitulé "Conception et développement d'une application mobile de Tourisme et de voyage sous la plateforme Android.

3. Présentation de l'organisme d'accueil

3.1. Historique

L'agence Nationale de Promotion et de développement des Parcs Technologiques (A.N.P.T) a été créée par décret exécutif n°04-91 du Safar 1425 correspondant au 24 mars 2004 (JORA : 019 du 28-03-2004).

- Mise en place en janvier 2007.
- C'est un établissement à caractère industriel et commercial (E.P.I.C) sous tutelle du ministère de la poste et de la Technologies de l'information et de la Communication (M.P.T.I.C), Son siège se trouve au Cyber parc de Sidi Abdallah. [URL 1]



Figure 1. Le logo de l'ANPT.

3.1.1. L'agence et la ville nouvelle Sidi Abdallah

La nouvelle ville de Sidi Abdallah a été choisie pour abriter ce parc technologique créé par décret n°04-275 du 05 septembre 2004. La ville nouvelle de Sidi Abdallah s'étend sur un périmètre urbanisable de 3000HA, dont une surface de 100HA est affectée au Cyber parc.

Il s'agit d'un pôle urbain complet capable, de participer au fonctionnement économique du potentiel métropolitain et de prendre en charge toutes les dimensions (économiques, sociales et résidentielles) complémentaire à celle d'Alger.

Pour mener le développement de la nouvelle ville. Les pouvoirs publics ont créé en date du 1er septembre 1997. L'Etablissement Public d'Aménagement de l'Agglomération Nouvelle de Sidi Abdallah (E.P.A-ANSA), chargé par délégation d'aménager le site (3000 ha) et d'encadrer la réalisation des programmes publics et privés (30000 logements, zones d'activités et équipements d'accompagnement, un parc urbain de 150 ha). [URL 1]

3.2. Les missions et les activités de l'ANPT

La création à Alger du permis parc technologie (le Cyber parc de Sidi Abdallah) s'inscrit dans un cadre de la stratégie nationale visant à édifier une Société de l'information et accélérer la transition de l'Algérie vers une économie du savoir.

Il s'agit d'un projet et un stratégique d'envergure nationale, il est à la fois un catalyseur de secteur des TIC et un propulseur de l'innovation.

Le Cyber parc d'Alger est un espace d'activités et de recherche centré sur les technologies de l'information et de la communication qui rassemble les différents du marché à savoir, Les entreprises privées et publiques, les établissements d'enseignement et de formation, les instituts de la R&D (Recherche et Développement), les groupes de réflexion et sociétés d'études de marché, les services aux entreprises, incubateurs d'entreprises, les centres de conférences, hôtel de classe internationale, dans le but de créer des synergies grâce à la proximité et aux interactions entre les membres de la grappe des TIC.

C'est un parc technologique concurrentiel de classe mondiale offrant une infrastructure de pointe et des services aux entreprises et institutions TIC.

Aussi, l'ANPT et l'instrument de l'état en matière de conception et de mise en œuvre d'une politique nationale de développement des parcs technologiques de ce fait,

l'ANPT est chargée :

- Élaborer et proposer les éléments d'une stratégie nationale en matière de promotion et de développement des parcs technologiques.
- Concevoir et mettre en place les parcs technologiques destinés au renforcement des capacités nationales en vue d'assurer le développement des technologies de l'information et de la communication et contribuer au développement économique et social.
- Réaliser ou faire réaliser les infrastructures des parcs technologiques.
- Mettre en synergie les institutions nationales de formation supérieure et de recherche de développement industriel ainsi que les institutions utilisatrices des technologies de l'information et de la communication pour les programmes de développement des parcs technologiques. [URL 1]

3.3. Les objectifs de l'ANPT:

3.3.1. Les objectifs spécifiques du Cyber parc:

- Créer un espace d'accueil dynamique pour les entreprises TIC, disposant d'une infrastructure high-tech, de services aux entreprises, de locaux de formation et de bureau (intelligents) à des prix abordables.
- Créer un (laboratoire) facilitant l'innovation en réunissant des partenaires stratégiques tels que sociétés privées et publique, installation de R&D, agence de commercialisation et incubateur.
- Créer une zone bénéficiant d'incitation à même d'attirer des investisseurs étrangers et de faciliter les transferts technologiques. [URL 1]

3.3.2. Objectifs relatifs à la grappe des TIC:

- Créer un Cyber parc qui servira de plaque tournante active pour les activités TIC de la région d'Alger et du pays dans son ensemble objectifs de développement économique générale.
-

- Offrir un soutien et des services de haute qualité aux entreprises TIC publique et privées afin de développer leur capacité à innover et leur compétitivité régionale et mondiale.
- Accroître le pourcentage des nouvelles entreprises et d'essaimages orientés TIC en offrant une aide initiale et soutenue aux entreprises.
- Diversifier la chaîne de valeur Algérienne des TIC pour qu'elle comporte toute la gamme des activités depuis la R&D et la commercialisation jusqu'à la conception l'ingénierie et l'exécutions et comprenne davantage d'éléments à valeur ajoutée.
- Développer un fort secteur TIC exportateur. [URL 1]

3.3.3. Objectifs de développement économique général:

- Expansion de l'emploi dans le secteur des TIC.
- Accroître l'investissement du secteur privé.
- Stopper la fuite des cerveaux par le développement d'offres d'emploi pour les jeunes diplômés et par l'offre d'une aide et de possibilités pour la création d'entreprises.
- Diversifier les sources de revenu national en suscitant un fort secteur d'activité basé sur le savoir.
- Accélérer et renforcer le développement du secteur privé.
- Attirer les investissements étrangers et faciliter les transferts de technologie.
[URL 1]

4. Présentation du projet

Les avantages des Smartphones sont nombreux aussi bien pour les voyageurs à travers le monde, et aussi pour les sociétés qui donnent des services qui facilitent les voyages et qui préservent à long terme leurs parts dans le marché et leurs rentabilités.

Mais ces sociétés n'offrent pas ces services partout dans le monde mais seulement dans les pays développés et les pays africains et du sud Amérique et quelques pays de l'Asie sont totalement ignorés.

Nous proposons comme solution une application mobile sous Android qui a comme objectif de répondre au besoin des voyageurs dans le monde entier sans séparation.

4.1. Objectifs de l'application

- Permettre l'utilisateur d'avoir toutes les informations et les détails concernant les vols en temps réel de n'importe quel aéroport dans le monde (recherche intelligente par vol, par aéroport et par itinéraire).
- La possibilité pour l'utilisateur de savoir tous les hôtels de n'importe quelle ville dans le monde avec la proposition des meilleures offres et les moins chers prix.
- La recherche et l'exploration des lieux touristiques et populaires avec les itinéraires vers chaque lieu à partir de la position de l'utilisateur.
- L'état de climat en temps réel et la prévision de la météo de 5 jours pour n'importe quelle ville dans le monde.
- Horloge mondiale et convertisseur de temps.
- Taux de change et convertisseur de devises.
- Rappel intelligent des vols.
- Scanner des billets d'avion (QR Code Scanner).
- Service de Taxi et location de voitures.

4.2. Approches du travail

Le projet comprend deux phases : la première est la recherche d'une solution convenable pour réaliser l'application, et la deuxième, la phase de conception et de développement.

- **Phase de recherche**

C'est l'étape qui inclut l'étude bibliographique, dans laquelle nous devons saisir les différentes notions et technologies à utiliser dans le projet, les architectures, etc.

Aussi, elle renferme les tests des différentes solutions mises en hypothèse pour réaliser l'application, et nous fixons les outils nécessaires pour la réalisation du projet.

- **Phase de conception et développement**

C'est une étape, dans laquelle, nous spécifions les besoins fonctionnels et nous modélisons le système à réaliser pour clarifier les tâches à accomplir dans la partie développement.

Cette phase se termine par une partie qui comprend la programmation et les tests de validation.

5. Conclusion

Dans ce chapitre introductif, nous avons présenté l'organisme d'accueil ainsi que le projet à réaliser. Nous allons entamer maintenant la phase de préparation de ce projet qui est l'étude de l'existant et la présentation des différentes solutions disponibles sur le marché.

Chapitre II :
Etude de l'existant

1. Introduction

Ce chapitre a pour objet de présenter quelques applications mobiles de tourisme et voyages d'une manière détaillée, en exposant leurs fonctionnalités.

2. Les solutions mobiles disponibles sur le marché

La technologie mobile offre de multiples avantages en termes de facilitation de voyage et proposition des meilleures offres de voyage :

- Augmenter et faciliter la recherche des meilleurs des endroits et destinations.
- Dématérialiser le billet et les tickets de voyage.
- Proposer des offres et des réductions.
- Réduire les coûts et délais d'une campagne ciblée de communication (moins de panneaux publicitaires, moins de papier, gain de temps, gratuité des notifications).

Le marché présente une panoplie des solutions informatiques dédiées à l'industrie du voyage et de tourisme, que nous présentons ci-après.

2.1. KAYAK

Application mobile Innovante et simple à utiliser, KAYAK permet de rechercher sur les vols, les hôtels et la location de voiture en quelques secondes. [URL2]

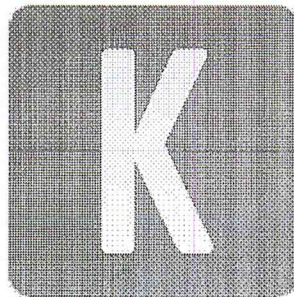


Figure 2. Le logo de KAYAK.

2.2. Hotel tonight

L'Application Hotel tonight est spécialisée dans l'hôtellerie et l'hébergement elle est facile à utiliser avec une interface simple mais elle n'offre pas d'autre service. [URL 3]

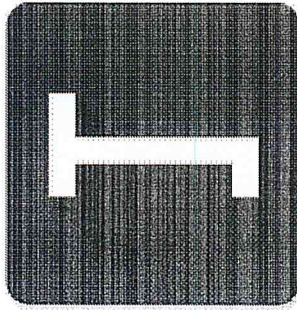


Figure 3. Le logo de Hotel tonight.

2.3. Expedia

Expedia permet au voyageur toutes les informations nécessaires mais elle a une interface difficile à utiliser ce qui perturbe l'utilisateur [URL 4]

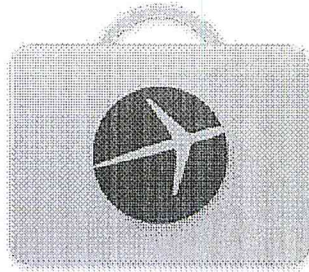


Figure 4. Le logo d'Expedia.

2.4. Uber

Uber propose des services unique de transportation car n'importe personne a une voiture peut devenir un chauffeur juste en créant un compte a Uber et l'utilisateur peut demander a n'importe quelle moment un chauffeur mais les services de Uber support seulement 400 ville dans le monde. [URL 5]

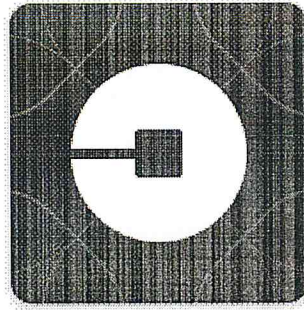


Figure 5. Le logo d'Uber.

2.5. Tripadvisor

Tripadvisor est la meilleur application de transport dans le Play Store elle offre plusieurs service sauf les services de location de voiture et taxi. [URL 6]

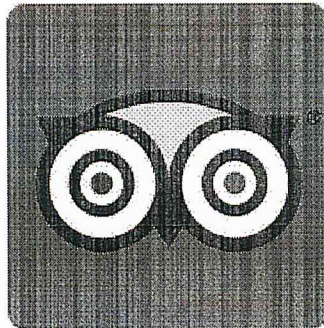


Figure 6. Le logo de Tripadvisor.

3. Critiques de l'existant

Quelle que soit la technologie utilisée ou les services supplémentaires apportés aux utilisateurs par ces solutions, un enjeu majeur reste la facilité de l'utilisation de la solution. La disponibilité des services dans tous les pays du monde car la majorité des applications de tourisme et de voyage ne supporte pas les pays en voie de développement comme les pays africains par exemple, notamment notre pays l'Algérie. De plus, nous constatons que la plupart des firmes offrant ses applications, présentent une version ciblée pour chaque pays avec des offres spéciales aux pays de l'Amérique du nord et l'Europe seulement.

Nous cherchons à développer une solution générique qui serait utilisable dans tous les pays du monde tout en gardant la facilité de l'utilisation de l'application et gratuitement.

4. Conclusion

Dans ce chapitre nous avons essayé d'étudier quelques applications mobiles de tourisme, voyage et transport dans le but d'avoir une idée sur les fonctionnalités de ces dernières et de ressortir leurs points forts. Sur la base de cette étude, nous allons élaborer la spécification de notre application. Une étude théorique sur les plates formes open source dédiées pour des applications Smartphones fonctionnant sur plusieurs Os mobiles fera l'objet du chapitre suivant.

Chapitre III :
Etude théorique et
choix techniques

1. Introduction

Dans ce chapitre, nous avons commencé par présenter les Smartphones et détaillé les différentes plates formes mobiles et nous allons terminer ce chapitre par présenter la plateforme Android celle qui nous s'intéresse beaucoup plus, avec une vue globale sur cette plateforme.

2. Etude Préliminaire

2.1. Smartphones et systèmes d'exploitation pour mobiles

Un Smartphone, un téléphone intelligent est un téléphone mobile qui offre des fonctions d'un "Personal digital Assistant ". Il fournit les fonctionnalités d'agenda, de navigation web, de calendrier, de consultation de courrier électronique, de messagerie instantanée et du GPS.

Un Smartphone peut aussi permettre d'installer des applications additionnelles sur l'appareil. Ces applications peuvent être développées par l'opérateur, par le fabricant ou par n'importe quel autre éditeur de logiciel.

Vu la grande importance des téléphones intelligents, une grande variété d'entreprises se place dans le marché des Smartphones en développant des systèmes d'exploitation pour mobiles.

Vers la fin de l'année 2014, les systèmes d'exploitation les plus répandus dans le monde sont : [URL 7]

- **BlackBerry OS**

A été créé par Research In Motion (RIM), une société Canadienne. Il est une plateforme très populaire particulièrement dans Amérique du Nord. Selon un rapport de la société Gartner, dans le marché des États-Unis, BlackBerry a maintenu le rang No1 avec 42 pour cent de marche. Mais cette plateforme seulement marche sur les appareils du BlackBerry. [URL 8]

- **PalmOS**

Est développé par la société Palm. En général, Palm OS est logique, intuitif et simple. En détaille, Palm OS est facile d'utiliser et simple d'apprendre. Il optimise les étapes pour naviguer entre les écrans et choisir les applications. Par exemple, pour lancer un

programme, appuyez sur son icône. Quand on passe à une autre application, cette application se termine. [URL 9]

- **Symbian**

Symbian est développé par la société Symbian. Selon la société Gartner, Symbian a commandé 57 pour cent des ventes globales à des utilisateurs finaux au cours du deuxième trimestre, en 2008 dans le marché des systèmes d'exploitation pour les Smartphones. Malgré que Symbian soit le leader mondial du système d'exploitation pour les Smartphones. [URL 10]

- **Ubuntu Touch**

Est développé par la communauté Ubuntu. Il est parrainé par les sociétés Canonical et Intel. Développé pour les MIDs (Mobile Internet Device) qui sont des appareils qu'on peut mettre entre téléphone portable et l'ordinateur. [URL 11]

- **Windows Phone**

Est développé par la corporation Microsoft. Alors, Windows Phone a la capacité de fonctionnement des logiciels sur « Windows » (seulement Windows), compatible avec tous les logiciels de « Windows ». Windows Phone soutient beaucoup de types audio, vidéo. Il soutient aussi la capacité de télécharger, de jouer les chansons et de regarder la TV en ligne. Grâce aux caractères, dans le domaine loisir, il est le plus bon. [URL 8]

- **Android**

Est développé par OHA (Open Handset Alliance). Android a été annoncé en 2007. De plus, en 2008, il est devenu une plateforme en code source ouverte. Selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sécurisé et ouverte. [URL 8]

- **Tizen**

Samsung est le premier à présenter un smartphone sous Tizen, sous le numéro de modèle SM-Z9005, en août 2013. Celui-ci reprend des atouts de Bada, l'OS maison et de Meegoo.

[URL 12]

- **IOS**

Est développé par Apple Inc. L'iPhone, l'iPod Touch et la tablette iPad utilisent tous le système d'exploitation iOS, qui est dérivé de Mac OS X. Les applications tierces sont

supportées depuis juillet 2008. La boutique Apple propose 300 000 applications. [URL 8]

- **Firefox OS**

(Précédemment connu sous le nom de Boot to Gecko ou B2G) est un système d'exploitation mobile libre proposé et développé par la Mozilla Corporation. [URL 13]

Durant ces deux dernières années et que Blackberry, Windows Mobile, iOS et Android sont devenus les systèmes d'exploitation les plus utilisés : La Figure présente le taux d'occupation du marché par ses systèmes d'exploitation mobile en deuxième trimestre de 2015 :

- iOS tout en occupant près de 13.9% du marché
- Windows Phone avec environ 2.6 % du marché
- Android avec environ 82.8% du marché
- Blackberry OS, Symbian, PalmOS, et autres avec des parts négligeables du marché. [URL 14]

2.2. Le choix de la plateforme

Cette diversité au niveau des plateformes mobiles, met les sociétés éditrice des logiciels dans l'embarras du choix de l'environnement du développement. On a pensé à concevoir une application mobile sous la plateforme Android, car cette plateforme est la plus utilisée et la majorité des smartphones marche sur cette plateforme.

3. Android

3.1. Définition

Android est un système d'exploitation Open Source pour smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 15 novembre 2007. Afin de promouvoir ce système d'exploitation ouvert. Google a su fédérer autour de lui une trentaine de partenaires réunis au sein de l'Open-Handset-Alliance, Open-Moko, le projet dont les spécifications logicielles et matérielles sont ouvertes.

Android doit son nom à la startup éponyme spécialisée dans le développement d'applications mobiles que Google a rachetée en août 2005. Le logiciel, qui avait été surnommé gPhone par les rumeurs de marchés, sera proposé gratuitement aux fabricants de téléphones mobiles, ce qui devrait faciliter son adoption.

L'opportunité d'intégrer un système d'exploitation puissant, gratuit et pouvant s'enrichir d'applications tierces à son matériel électronique a ouvert la route à plusieurs projets.

Android équipe maintenant des cadres photos, des tablettes tactiles, des netbooks, des voitures, des smartbooks et prochainement des Walkman Sony. [1]

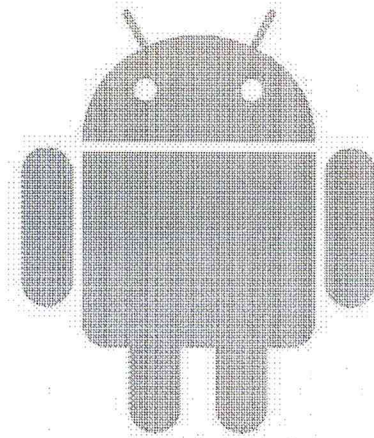


Figure 7. Le logo de l'Android.

3.2. Le SDK Android

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.)

L'android SDK (Software Development Kit) amène des outils :

- un environnement de développement.
- une machine virtuelle Java adaptée : Android Run Time machine.

- un environnement débogueur DDMS (Dalvik Debug Monitor Service) utilisant ADB (Android Debug Bridge).
- un environnement de construction d'application Android AAPT (Android Asset Packaging Tool).
- des émulateurs de téléphones ou de tablettes AVD (Android Virtual Device)
- une énorme API (pour l'intégration des fonctionnalités liées aux services Google. [1])

3.3. L'émulateur

Evoqué plus haut, le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel.

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth. [2] [4]

3.4. Une architecture autour du noyau Linux

Android est une pile applicative pour les appareils mobiles qui comprend un système d'exploitation, middleware et des applications clés. Le SDK Android fournit les outils et les API nécessaires pour commencer à développer des applications sur la plate-forme Android en utilisant le langage de programmation Java. [1]

- **Applications**

Android est fourni avec un ensemble de programmes de base (également nommés applications natives) permettant d'accéder à des fonctionnalités comme les courriels, les SMS, le téléphone, le calendrier, les photos, les cartes géographiques, le Web, pour ne citer que quelques exemples. Ces applications sont développées à l'aide du langage de programmation Java. Pour l'utilisateur final, c'est la seule couche accessible et visible. [1]

- **Le framework (Application Framework)**

En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovants. Les développeurs sont libres de profiter du matériel périphérique, les informations de localisation d'accès, exécuter les services d'arrière-plan, définir des alarmes, ajouter des notifications de la barre d'état, et beaucoup, beaucoup plus.

- **Les bibliothèques (Libraries)**

En interne, Android inclut un ensemble de bibliothèques C et C++ utilisées par de nombreux composants de la plateforme Android. Ces bibliothèques sont en réalité accessibles au développeur par l'intermédiaire du framework Android. En effet, le framework Android effectue, de façon interne, des appels à des fonctions C/C++ beaucoup plus rapides à exécuter que des méthodes Java standard. La technologie Java Native Interface (JNI) permet d'effectuer des échanges entre le code Java et le code C et C++. La liste ci-dessous énumère quelques-unes des bibliothèques disponibles dans Android :

- Bibliothèque système C, optimisée pour les systèmes Linux embarqués.
- Bibliothèques multimédias, pour le support des formats audio et vidéo.
- SurfaceFlinger : Permet l'accès au sous-système d'affichage.
- LibWebCore : Moteur de rendu de pages Internet basé sur Webkit.
- Skia : Moteur graphique 2D.
- Bibliothèques 3D : Implémentation basée sur OpenGL ES 2.0.
- FreeType: Rendu des polices de caractères.
- SQLite: Base de données légère et puissante. [1]

- **Moteur d'exécution Android (Android Runtime)**

Android inclut un ensemble de bibliothèques qui fournit la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec sa propre instance de machine virtuelle Dalvik. Dalvik VM est une implémentation de machine virtuelle ayant été conçue pour optimiser l'exécution multiple de machines virtuelles. Elle exécute du bytecode qui lui est dédié : le bytecodex. (Format qui est optimisé pour une empreinte mémoire minimale).

Cette particularité d'Android en fait un système unique, loin des systèmes Linux traditionnels que beaucoup avaient pu rencontrer auparavant. [1]

- **Noyau Linux (Linux Kernel)**

Android repose sur un noyau Linux (version 2.6) qui gère les services du système, comme la sécurité, la gestion de la mémoire et des processus, la pile réseau et les pilotes. Il agit également comme une couche d'abstraction entre le matériel et la pile logicielle. [1]

3.5. Les versions d'Android

Annoncé officiellement en novembre 2007, le système Android a fait du chemin depuis ses débuts. Depuis la version alpha, les versions possèdent des noms de desserts (en anglais) :

- Android Alpha
- Android Bêta
- cupcake
- Donut
- Eclair
- Froyo
- Gingerbread
- Honeycomb
- Ice Cream Sandwich
- Jelly Bean
- KitKat
- Lollipop
- Marshmallow [URL 15]

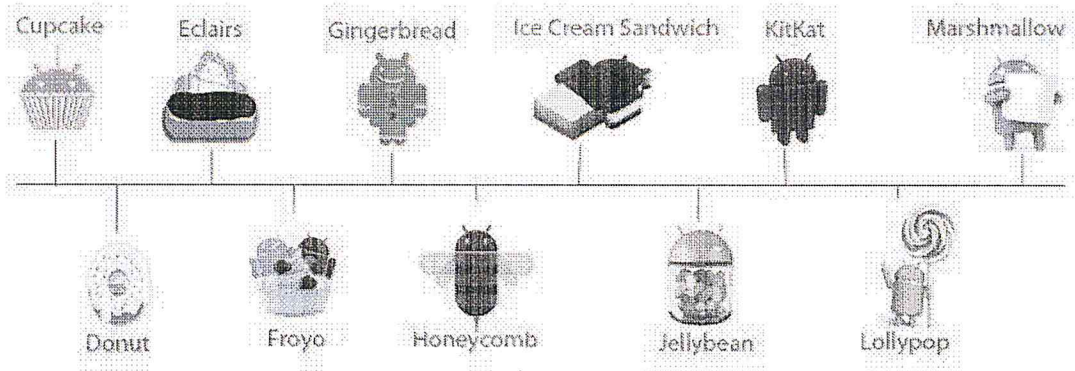


Figure 8. Les versions d'Android.

3.6. Caractéristiques d'Android

- **Open source**

Le contrat de licence pour Android respecte les principes de l'open source, c'est-à-dire les sources sont téléchargeables et modifiables. [URL 16]

- **Facile à développer**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire que l'envoi d'un SMS est en seulement deux lignes de code (concrètement, il y a un peu d'enrobage autour de ce code, mais pas tellement). [URL 16]

- **Flexible**

Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les smartphones, les tablettes, la présence ou l'absence de clavier ou de trackball, différents processeurs. . . On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android ! Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal (si votre application nécessite d'utiliser le Bluetooth, seuls les terminaux équipés de Bluetooth pourront la voir sur le Play Store). [URL 16]

- **Précis**

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est qu'on peut combiner plusieurs composants totalement différents pour

obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises. [URL 16]

3.7. Les avantages d'Android

- Android est une plateforme en code source ouverte. De plus, selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sûre et ouvert. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de replacer les composants existants. Les utilisateurs peuvent adopter les applications à leur besoin.
- Android est basé sur le noyau Linux. Alors, il y a plusieurs avantages comme une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partagé, etc.
- Le SDK de l'Android offre complètement les APIs pour développer l'application sur Android.
- Grâce au « Android Developer Challenge » que Google fournira 10 millions de dollars en prix - aucune condition pour les applications sur la plateforme Android, les développeurs ont eu l'opportunité de gagner beaucoup d'argent.
- centaines de millions d'appareils mobiles dans plus de 190 pays à travers le monde. C'est la plus grande base installée de toute plate-forme mobile et en croissance rapide chaque jour un million utilisateurs alimenter leurs appareils Android pour la première fois et commencer à chercher des applications, des jeux et d'autres contenus numériques.
- Android donne une plate-forme de classe mondiale pour créer des applications et des jeux pour les utilisateurs d'Android partout, ainsi que d'un marché ouvert pour la distribution à l'autre instantanément. [URL 16]

4. Structure du Projet Android

4.1. Architecture basique d'une application Android

Un Site Web est composé de plusieurs pages. Une activité comme une page d'un site web. Lorsqu'on change de page, on change d'activité.

Un Site Web a une page d'accueil, une application Android à une activité principale.

Une activité, est séparée en 2 morceaux :

- **un fichier XML**

Permet la déclaration des éléments de l'interface (par exemple un bouton), ainsi que ses attributs (le bouton aura pour id "bouton1", pour texte "ceci est un bouton", etc.).

- **un fichier JAVA**

Permet le contrôle des éléments déclarés dans le fichier XML. Il est directement lié au XML par une ligne de code. [URL 17]

4.2. Présentation d'un projet Android

- **MainActivity.java**

Tout d'abord j'ai la classe MainActivity.java qui correspond à l'activité principale créée avec le projet (car liée au fichier activity_main.xml). [URL 17]

- **R.java**

Fichier contenant les "paramètres" de l'application. Il se génère automatiquement, personne ne doit y toucher, au risque de bugger l'application. [URL 17]

- **Bin**

Le dossier "Bin" contient les binaires de l'application. Comme pour le fichier R.java, ils ne doivent pas être touchés - ne pas les commit sur un SVN non plus, chaque membre d'un groupe de projet a des binaires différents (risque de bugs). [URL 17]

- **Res**

Ce dossier contient les images utilisées dans le projet, les XML du projet (les fichiers XML servent à définir l'interface de l'application) & le dossier "values" (voir plus bas). [URL 17]

- **Drawable-XXX**

Ces dossiers contiennent les images de l'application. Tous ces dossiers devraient, dans l'idéal, contenir les mêmes images, mais avec des résolutions différentes. En effet, selon la résolution de l'écran utilisé, les images seront tirées du dossier correspondant. [URL 17]

- **Layout**

Ce dossier contient les XML de l'application pour les interfaces. On peut déjà voir l'activity_main.xml, qui correspond à l'interface de l'activité principale de l'application.

[URL 17]

- **Values**

Ce dossier contient aussi des XML, mais qui servent à stocker des chaînes de caractères.

[URL 17]

- **AndroidManifest.xml**

Fichier XML dans lequel nous déclarons les différentes activités de l'application, et certains paramètres de l'application (par exemple si j'autorise la rotation de l'application, le sens de l'application (vertical, horizontal), l'icône de l'application, etc.). [URL 17]

Les fichiers restants sont générés automatiquement, ils n'ont pas d'intérêts particuliers.

[URL 17]

5. Conclusion

Nous avons présenté dans ce chapitre une idée sur l'architecture à adoptée dans notre projet. De plus, nous avons présenté la solution retenue et les technologies à utiliser.

Une étude plus approfondie sera présentée dans les prochains chapitres. L'analyse des besoins et la spécification feront l'objet du chapitre suivant.

Chapitre IV :
Analyse, spécification
et Conception

1. Introduction

L'étude conceptuelle consiste à décrire le système futur. Ce nouveau système s'inspirera d'une part du système existant et d'autre part tiendra compte des nouveaux besoins des utilisateurs.

Pour la conception d'un tel système, nous avons utilisé le cycle de vie en cascade en se basant sur le langage de modélisation UML. D'abord, nous avons traduit la liste des exigences en diagrammes de cas d'utilisation, qui permettent d'analyser et d'organiser les besoins, suivis par les diagrammes de séquence. Ensuite, nous avons étudié l'aspect statique avec les diagrammes de classes et diagrammes d'activités.

2. Présentation de la démarche utilisée

Nous expliquerons dans cette étape du chapitre, les besoins de notre application afin de pouvoir passer à l'étape de conception et d'architecture en présentant le cycle de vie que nous avons suivi pour la réalisation de ce projet. Nous illustrerons les solutions apportées par notre outil face aux problèmes posés, en se basant sur le langage UML (Unified Modeling Language) en utilisant le processus UP (Unified Process).

UP est une méthode de prise en charge du cycle de vie d'un logiciel développé en orienté objet. Elle représente les étapes du cycle de vie sous formes de diagrammes UML.

2.1. Le cycle de vie

Le cycle de vie d'un logiciel est un ensemble séquentiel de phases, dont le nom et le nombre sont déterminés en fonction des besoins du projet, permettant généralement le développement d'un service ou d'un produit. Quant à notre projet il suit le modèle en cascade. [3]

3. Modèle en cascade

Le processus de développement que nous avons utilisé est le cycle de vie en cascade, ce modèle décrit par Winston W. Royce en 1970 a été largement utilisé, pour la description générale des activités liées aux logiciels .il décrit le cycle de vie d'un logiciel par une

suite de phases enchainées dans un déroulement linéaire, depuis l'analyse des besoins jusqu'à la maintenance.

Le résultat de chaque étape est testé et on ne passe à l'étape suivante que lorsque l'étape actuelle est satisfaisante. [8]

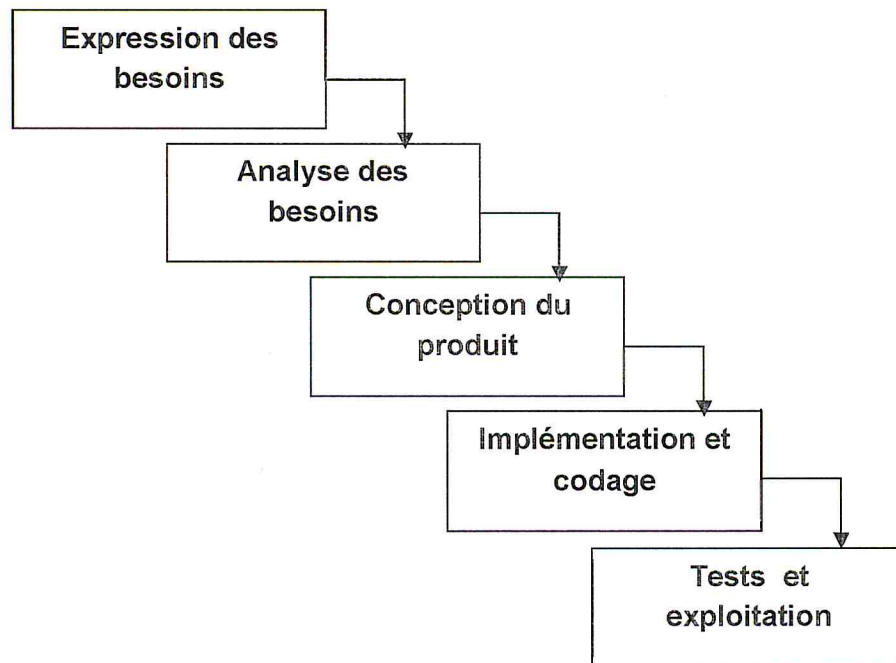


Figure 9. Cycle de vie selon le modèle en cascade.

3.1. Expression des besoins

Au début du processus de développement, la spécification des besoins est une étape essentielle et consiste généralement à déterminer précisément les besoins des utilisateurs du système afin d'éviter de développer des logiciels non adéquats.

Cette étape ne s'intéresse pas aux solutions mais aux questions uniquement: elle identifie le « quoi faire ? », et identifie les entités de l'environnement du système. Pour modéliser ces besoins on utilise le diagramme des cas d'utilisation en UML.

3.2. Analyse

L'objectif de l'analyse est d'arriver à comprendre les besoins et les exigences du client, afin de livrer des spécifications permettant de réaliser la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (Scenario) en utilisant le diagramme de séquence d'UML pour représenter les interactions entre les objets.

3.3. Conception

C'est la phase la plus importante du processus de développement d'un logiciel. Elle s'intéresse d'abord au «comment ? », à savoir la solution du problème énoncé.

La conception a pour but de décomposer le logiciel en module, de préciser les interfaces et les fonctions de chaque module. A l'issue de cette étape, on obtient une description de l'architecture du logiciel et un ensemble de spécifications de ces divers composants en utilisant les deux diagrammes de classe et d'activité d'UML.

3.3.1 Méthode de conception

- **OOAD** (Object Oriented Analyses and Design ou la Méthode d'Analyse et de Conception d'Applications Orientées Objet) , est fondée sur une démarche participative par prototypage incrémental (processus itératif) permettant aux utilisateurs d'intervenir très tôt dans le processus de développement du logiciel, OOAD utilise la notation UML afin de définir la structure du logiciel en termes de classes et de composants ainsi que pour modéliser sa dynamique à l'aide de diagrammes d'interactions ou d'états/transitions. À partir des cas d'utilisation obtenus par interviews des utilisateurs, deux types de modèles originaux sont utilisés pour représenter l'Interface Homme-machine du logiciel : un modèle conceptuel construit à partir du diagramme des classes et de patrons de conception, et un modèle de réalisation permettant une mise en œuvre optimum dans un langage orienté objet , tel que Java ou C++, ou en langage HTML pour les applications liées à l'Internet ou à intranet. Afin de limiter les tests de non-régression toujours très lourds et coûteux, OOAD applique à chaque prototype réalisé le principe de non-régression basé sur l'encapsulation et l'héritage qui font tout l'intérêt et la richesse de la programmation orientée objet.[3]
 - **DDD** (Domain-Driven Design, ou La conception pilotée par le domaine) est une approche, de la conception de logiciels, basée sur deux principes :
-

- les conceptions complexes qui doivent être basées sur un modèle.
- l'accent doit être sur le domaine et la logique associée.[3]

3.4. Implémentation

L'implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts exécutables et d'autres éléments du même types.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous forme de code source.

3.5. Tests

Les tests permettent de vérifier les résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

4. Présentation de l'UML

UML (Unified Modeling Language - langage unifié pour la modélisation objet) est un moyen d'exprimer des modèles objets, c'est-à-dire que le modèle fourni par UML est valable pour n'importe quel langage de programmation. UML couvre toutes les phases du cycle de vie d'un logiciel, et a pour but de documenter les modèles objets.

Un modèle est une simplification de la réalité, donc la modélisation consiste à créer une représentation simplifiée d'un problème, elle comporte trois composantes :

- **L'expression des besoins** : définir les besoins des Utilisateurs de système.
- **L'Analyse** : livrer une spécification complète des besoins.
- **La conception** : la mise au point d'une solution.[3]

4.1. Pourquoi Modéliser avec UML ?

UML 2.0 a introduit de nouveaux concepts tels que l'encapsulation, l'héritage, le polymorphisme, l'agrégation et la composition. Des concepts orienté objet tel est l'approche qui caractérise UML par rapport à d'autres langages, mais surtout un

meilleur mariage entre les données et Le traitement, et donc un passage plus direct vers les langages de programmation.

4.2. Les diagrammes

Pour modéliser le traitement, UML 2.0 propose quatorze (14) diagrammes dont l'objectif est de pouvoir générer une grande partie du code à partir d'un modèle visuel.

4.2.1. Diagrammes structurels ou statiques

- Diagramme de classes.
- Diagramme d'objets.
- Diagramme de composants.
- Diagramme de déploiement.
- Diagramme des paquetages.
- Diagramme de structure composite.
- Diagramme de profils.

4.2.2. Diagrammes comportementaux

- Diagramme des cas d'utilisation.
- Diagramme états-transitions.
- Diagramme d'activité.

4.2.3. Diagrammes d'interaction ou dynamiques

- Diagramme de séquence.
- Diagramme de communication.
- Diagramme global d'interaction.
- Diagramme de temps.

Dans notre projet, nous avons utilisé les diagrammes suivants :

- **Expression des besoins** : Diagramme de cas d'utilisation.
- **Analyse** : Diagramme de séquence.
- **Conception** : Diagramme de classe.

Avec sa richesse et sa grande diversité, UML n'est pas une méthode mais un langage graphique universel pour représenter les systèmes d'informations et un support dans la méthode d'analyse.

5. Expression des besoins

Cette phase consiste à définir les besoins fonctionnels de notre futur système, nous allons parler des fonctionnalités que peut offrir ce dernier afin de bien connaître les acteurs qui interagissent avec lui par la suite nous allons les modéliser en utilisant le diagramme des cas d'utilisation d'UML. [3]

5.1. Les besoins fonctionnels

- Rechercher les vols.
- Rechercher les hôtels.
- Rechercher des spots touristiques.
- Rechercher des services de location de voiture.
- Rechercher des trains.
- Scanner un code QR.
- Suivi des événements sur la carte.
- S'informer à la météo.
- Convertir le temps.
- Convertir l'argent.
- Gérer les rappels.
- Contacter la société.

5.2. Les besoins non fonctionnels

Après avoir déterminé les besoins fonctionnels nous présentons ci-dessous l'ensemble des contraintes à respecter pour garantir la performance du système, donc de fournir un produit performant qui respecte les exigences de l'utilisateur et qui peut faire face à des risques de panne ou de non fonctionnement.

- **Performance :**

Afin d'être acceptée par l'utilisateur, notre application doit respecter ce critère tout en assurant un temps de réponse minimum et des fonctionnalités répondant aux besoins de l'utilisateur.

- **Sécurité :**

L'application ne doit pas provoquer des vulnérabilités sérieuses qui menacent la stabilité du système et la confidentialité de ce dernier.

- **La simplicité:**

Un visiteur novice pourra utiliser un tel service de façon intuitive.

- **L'ergonomie de l'interface :**

Les interfaces doivent être simples et conviviales.

- **La modularité de l'application :**

Avoir un code source simple, facile à maintenir et à comprendre en cas de besoin. Le but est de diminuer la complexité du domaine en découpant le système en plusieurs modules indépendants.

6. Diagramme de cas d'utilisation

6.1. Description

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système, des associations de communication entre les acteurs et les cas d'utilisation, et des généralisations entre cas d'utilisation. Il est destiné à représenter les besoins des utilisateurs par rapport au système. [3]

6.1.1. Identification des acteurs et de cas d'utilisation

- **Acteurs :**

Les acteurs d'un système sont les entités externes à ce système qui interagissent avec lui. L'acteur a un nom, qui le définit, ou qui précise son rôle dans la transaction décrite. Dans notre application, le seul acteur qui interagit avec le système est l'utilisateur.

- **Cas d'utilisations**

Un cas d'utilisation est utilisé pour définir le comportement d'un système ou la sémantique de toute autre entité sans révéler sa structure interne. Chaque cas d'utilisation spécifie une séquence d'action, y compris des variantes, que l'entité réalise, en interagissant avec les acteurs de l'entité.

6.1.2. Description des acteurs

Cette section a pour but de présenter les acteurs et leurs fonctionnalités à lesquelles doit répondre notre application. Nous commençons notre analyse par identifier l'acteur qui agit sur notre système à savoir l'utilisateur qui est le seul acteur qui interagit avec notre application. Cet utilisateur bénéficie de toutes les fonctionnalités de l'application sans faire une connexion.

6.3. Diagramme de cas d'utilisation

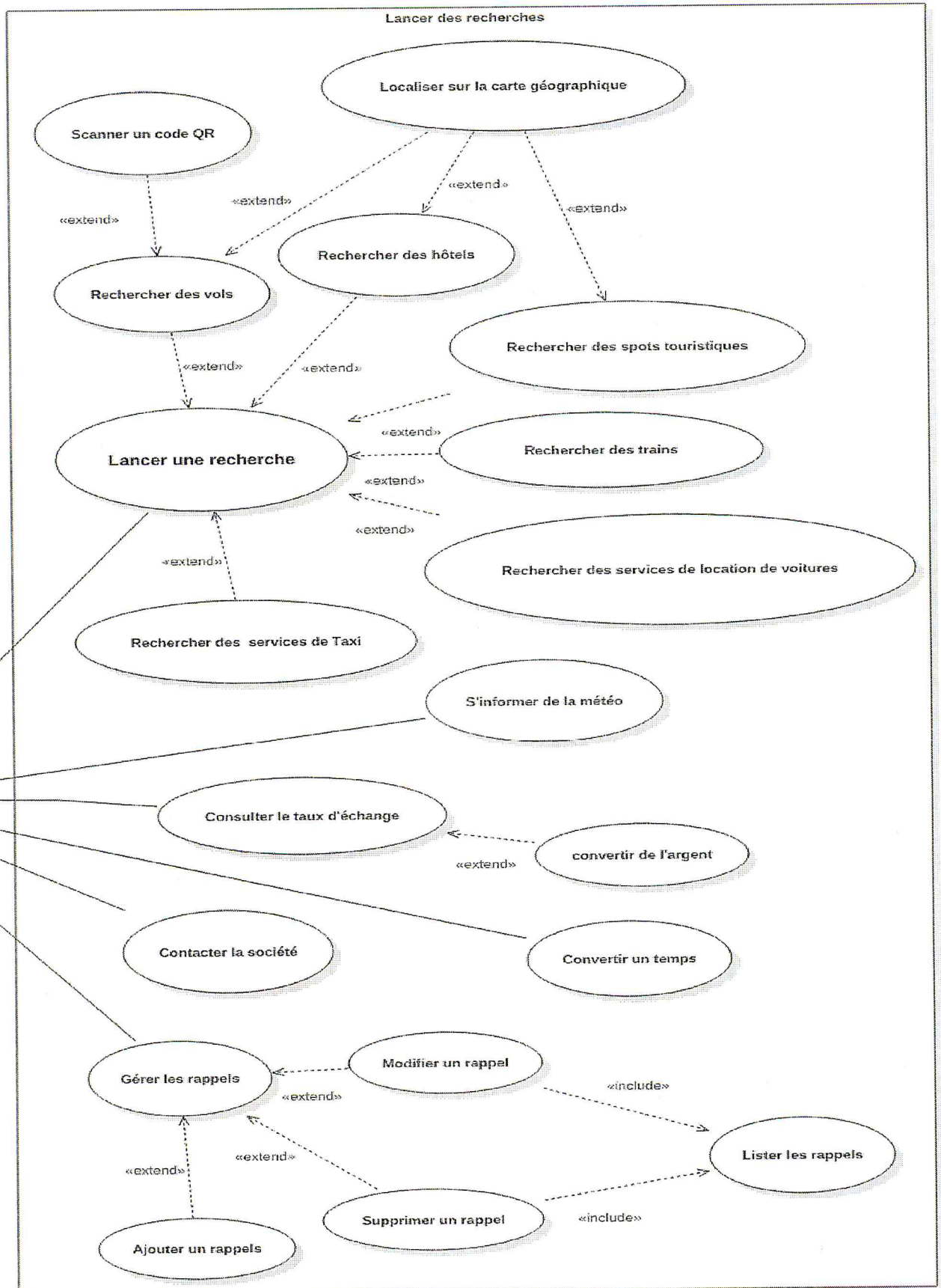


Figure 10. Diagramme de cas d'utilisation de l'application.

● **Raffinement des cas d'utilisation :**

→ Le raffinement du cas d'utilisation « **Rechercher les vols** » : l'utilisateur veut lancer une recherche sur les vols disponibles.

Scénario	Rechercher les vols
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur les vols.
Post condition	Des informations utiles sur le vol recherché.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données (la date, l'aéroport de départ ...). <ol style="list-style-type: none"> 1.1. L'utilisateur peut scanner un code QR existant (imprimé sur le billet de vol). 2. L'application vérifie les données. <ol style="list-style-type: none"> 2.1. Si les données sont fausses, l'application notifie l'utilisateur. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique. <ol style="list-style-type: none"> 4.1. Si l'avion est en cours vol, l'utilisateur peut le localiser sur la carte géographique. 4.2. Il peut aussi marquer une notification (quand le vol atteint la destination il reçoit une notification).
Exceptions	<p>Les données (la date, l'aéroport) sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>Le code QR scanné erroné ou bien ne concerne pas un code QR d'un vol.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 1. Raffinement de cas d'utilisation « **Rechercher les vols** »

→ Le raffinement du cas d'utilisation « **Rechercher les hôtels** » :
 l'utilisateur veut lancer des recherches sur les hôtels.

Scénario	Rechercher les hôtels
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur les hôtels.
Post condition	Des informations utiles sur les hôtels recherchés.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données (la date, le prix, la réputation ...). 2. L'application vérifie les données. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique. 4.1. Localiser l'emplacement de l'hôtel sur la carte géographique.
Exceptions	<p>Les données (la date, le prix, la réputation ...) sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 2. Raffinement du cas d'utilisation « **Rechercher les hôtels** »

→ Le raffinement du cas d'utilisation « **Rechercher des spots touristiques** » : l'utilisateur veut lancer des recherches sur des places célèbres.

Scénario	Rechercher des places célèbres
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur des places célèbres.
Post condition	Des informations sur les places célèbres.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données (la distance maximale ...). 2. L'application vérifie les données. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique. 4.1. Localiser l'emplacement de toutes les places sur la carte

	géographique.
Exceptions	Les données (la distance maximale...) sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées. En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.

Tableau 3. Raffinement du cas d'utilisation « **Rechercher des spots touristiques** »

→ Le raffinement du cas d'utilisation « **Recherche d'un service de location de voitures** » : l'utilisateur veut lancer des recherches sur les services de location de voitures disponibles.

Scénario	Rechercher un service de location de voitures
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur des services de location de voiture.
Post condition	Des informations sur les services de locations des voitures.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données. 2. L'application vérifie les données. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique.
Exceptions	Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées. En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.

Tableau 4. Raffinement du cas d'utilisation « **Recherche d'un service de location de voitures** »

→ Le raffinement du cas d'utilisation « **Recherche d'un service de Taxi** » : l'utilisateur veut lancer des recherches sur les services de Taxi disponibles.

Scénario	Rechercher un service de Taxi
Acteur	Utilisateur

Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur un service de Taxi.
Post condition	Des informations sur les services de Taxi.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données. 2. L'application vérifie les données. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique.
Exceptions	<p>Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 5. Raffinement du cas d'utilisation « **Recherche d'un service de Taxi** »

→ Le raffinement du cas d'utilisation « **Rechercher des trains** » :

l'utilisateur veut lancer des recherches sur les trains.

Scénario	Rechercher des trains
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut lancer des recherches sur des trains.
Post condition	Des informations sur les services de trains.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données (numéro de trains). 2. L'application vérifie les données. 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique.
Exceptions	<p>Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 6. Raffinement du cas d'utilisation « **Rechercher des trains** »

Le raffinement des cas d'utilisation :

→ Le raffinement du cas d'utilisation « **S'informer de la météo** » :
l'utilisateur veut connaître les météos sur une ville quelconque.

Scénario	S'informer de la météo
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut savoir la météo.
Post condition	Des informations météorologiques.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre à l'application (la ville). 2. L'application contact la source de données. 3. L'application faire une mise à jour de l'interface graphique.
Exceptions	<p>Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corriger.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 7. Raffinement du cas d'utilisation « **S'informer de la météo** »

→ Le raffinement du cas d'utilisation « **Convertir un temps** » : l'utilisateur veut convertir le temps entre deux fuseaux horaires.

Scénario	Convertir un temps
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut convertir le temps.
Post condition	Le temps converti.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur choisir le fuseau horaire. 2. L'application vérifie les données (si le fuseau horaire est bien écrit). 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique.
Exceptions	<p>Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 8. Raffinement du cas d'utilisation « **Convertir un temps** »→ Le raffinement du cas d'utilisation « **Consulter le taux d'échange** »

Scénario	Convertir un temps
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut convertir le temps.
Post condition	Le temps converti.
Scénario principal	<ol style="list-style-type: none"> 5. L'utilisateur choisir le fuseau horaire. 6. L'application vérifie les données (si le fuseau horaire est bien écrit). 7. L'application contact la source de données. 8. L'application faire une mise à jour de l'interface graphique.
Exceptions	<p>Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.</p> <p>En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.</p>

Tableau 9. Raffinement du cas d'utilisation « **Consulter le taux d'échange** »→ Le raffinement du cas d'utilisation « **Convertir de l'argent** » :

l'utilisateur veut l'argent d'une devise vers une autre.

Scénario	Convertir de l'argent
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut convertir de l'argent.
Post condition	L'argent converti.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur entre les données (la somme, les devises ...). 2. L'application vérifie les données (si la somme n'est inférieure ou égale à zéro ...). 3. L'application contact la source de données. 4. L'application faire une mise à jour de l'interface graphique.
Exceptions	Les données sont erronées, dans ce cas, le système notifie l'utilisateur pour les corrigées.

	En cas d'un problème de connexion avec la source de données, le système notifie l'utilisateur.
--	--

Tableau 10. Raffinement du cas d'utilisation « **Convertir de l'argent** »

→ Le raffinement du cas d'utilisation « **Contacteur la société** » : l'utilisateur veut contacter la société responsable de cette application afin de donner son avis, reporter un bug ...

Scénario	Contacteur la société
Acteur	Utilisateur
Pré condition	Le système fonctionne et l'utilisateur veut contacter la société.
Post condition	L'utilisateur mail par défaut de l'utilisateur est lancé avec l'adresse de la société.
Scénario principal	<ol style="list-style-type: none"> 1. L'utilisateur accède l'application. 2. L'application ouvre son client de mail par défaut.

Tableau 11. Raffinement du cas d'utilisation « **Contacteur la société** »

7. La conception de l'application

Nous allons présenter l'architecture globale de l'application, la manière d'interaction entre les différentes parties du système afin de produire les fonctionnalités offertes et en fin l'organisation conceptuelle de ces parties dans l'application.

7.1. Les styles architecturaux utilisés

7.1.1. Architecture multicouche

Dans l'architecture multicouche le système est découpé en plusieurs couches, où chaque couche est indépendante aux autres (couplage faible les objets de chaque couche) et communique seulement avec les couches voisines. Cette architecture est basée sur le principe de séparation des concerns.

Cette architecture bénéficie de plusieurs avantages :

- Clarté, chaque couche a des concerns spécifiques.
- Flexibilité, chaque couche communique seulement avec ses voisines sans connaître le fonctionnement intérieur des autres (Encapsulation), donc une

possibilité de remplacer une couche par une autre qui a le même comportement mais un fonctionnement intérieur qui est différent.

- Testabilité, la possibilité de tester chaque couche d'une manière indépendante aux autres.
- Etc. [5, 6]

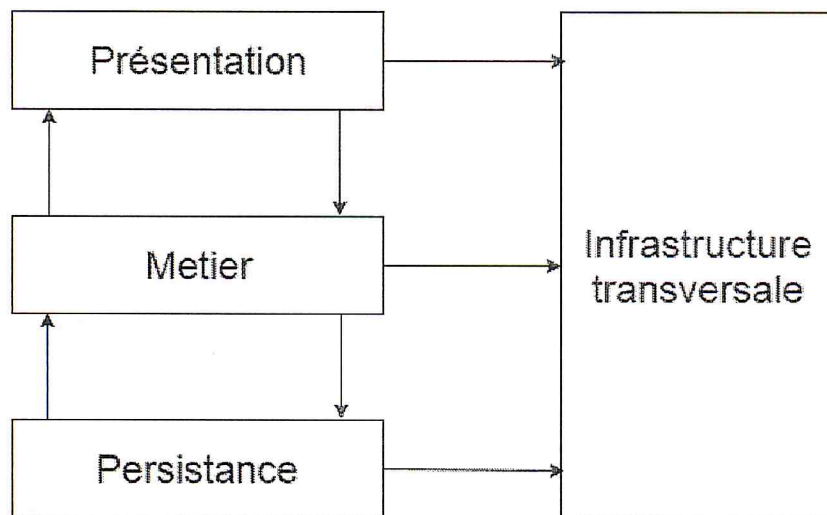


Figure 11. Architecture multicouche du client.

→ **Couche Présentation:**

La couche Présentation est le premier niveau dans notre architecture, cette couche est responsable de l'affichage des données à l'utilisateur, l'interception et l'interprétation des actions de l'utilisateur.

→ **Couche Métier :**

La couche Métier est le deuxième niveau, cette couche est le cœur de l'application, les composants de cette couche implémentent la logique métier de l'application d'une façon indépendante.

→ **Couche Persistance :**

Elle centralise les traitements d'accès aux sources de données. Son principal objectif est de dissocier la logique métier de celle de persistance.

Elle définit les entrepôts de données (« Repositories ») responsables d'instancier et de pérenniser les objets métier de l'application, et assure les échanges avec les systèmes tiers.

→ **Infrastructure transversale :**

Cette couche technique est dénuée de toute logique métier, son rôle consiste à fournir un socle solide de toutes les fonctionnalités facilitant la gestion des journaux d'exécution (logs), améliorant la robustesse et les performances de l'application (cache, montée en charge, gestion des threads, etc.) et offrant des services divers (email, accès disque dur, politique de sécurité...).

7.1.2. Architecture client-serveur :

Les fournisseurs de service répondent aux requêtes du client, le rôle de client ici est très claire, c'est une vitrine des services web sur le protocole HTTP/HTTPS.

Ces services web sont sécurisés à l'aide des standards comme OAuth2/SAML, et avec la nature de *sans état* de protocole HTTP, c'est la responsabilité du client de s'authentifier à chaque requête émise au service web.

Les services Webs (fournisseurs de services) acceptent des requêtes HTTP/HTTPS et répondent avec des réponses sous format de JSON/XML, puis le client les analyse et les affiche aux utilisateurs sous format des widgets.

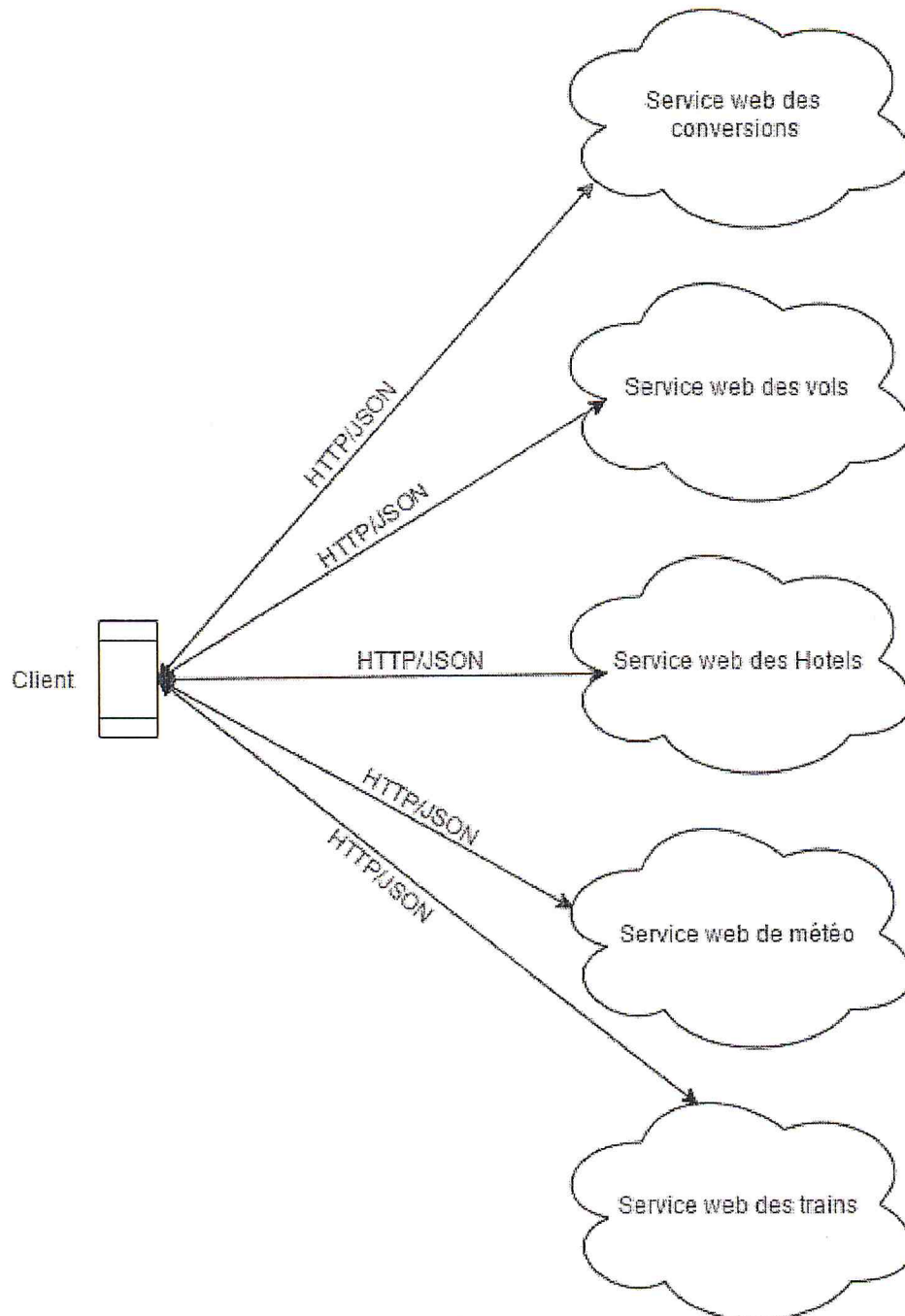


Figure 12. Architecture globale de l'application.

7.2. Diagrammes de séquence

Tous les cas d'utilisation se généralisent en deux catégories :

- Connectés (en ligne), ceux qui nécessitent une connexion Internet.

- Non connectés (hors ligne), ceux qui peuvent être utilisés sans une connexion Internet.

Classification des cas d'utilisation selon leur catégorie :

- **Les cas d'utilisation connectés :**

- Rechercher des vols.
- Rechercher des hôtels.
- Rechercher des places touristiques.
- Recherche d'un service de location de voiture.
- Recherche d'un service de Taxi.
- Recherche d'une station de train.
- Connaitre la météo.
- Convertir le temps.
- Suivi sur la carte.
- Consulter le taux d'échange.

- **Les cas d'utilisation non connectés :**

- Gérer les rappels.
- Contacter la société.
- Convertir l'argent.

7.2.1. Le diagramme de séquence (connectés)

Ce diagramme de séquence est un peu long, car la fonctionnalité traverse toutes ces couches pour aller au service web/base locale de données. Ce diagramme de séquence généralise tous les cas d'utilisation qui nécessitent une communication avec les services web, ci-dessus la liste des cas d'utilisation concernés par ce diagramme de séquence (le même chemin pour ces fonctionnalités, juste l'interaction de l'utilisateur avec l'application, les services et le repository ne sont pas les mêmes).

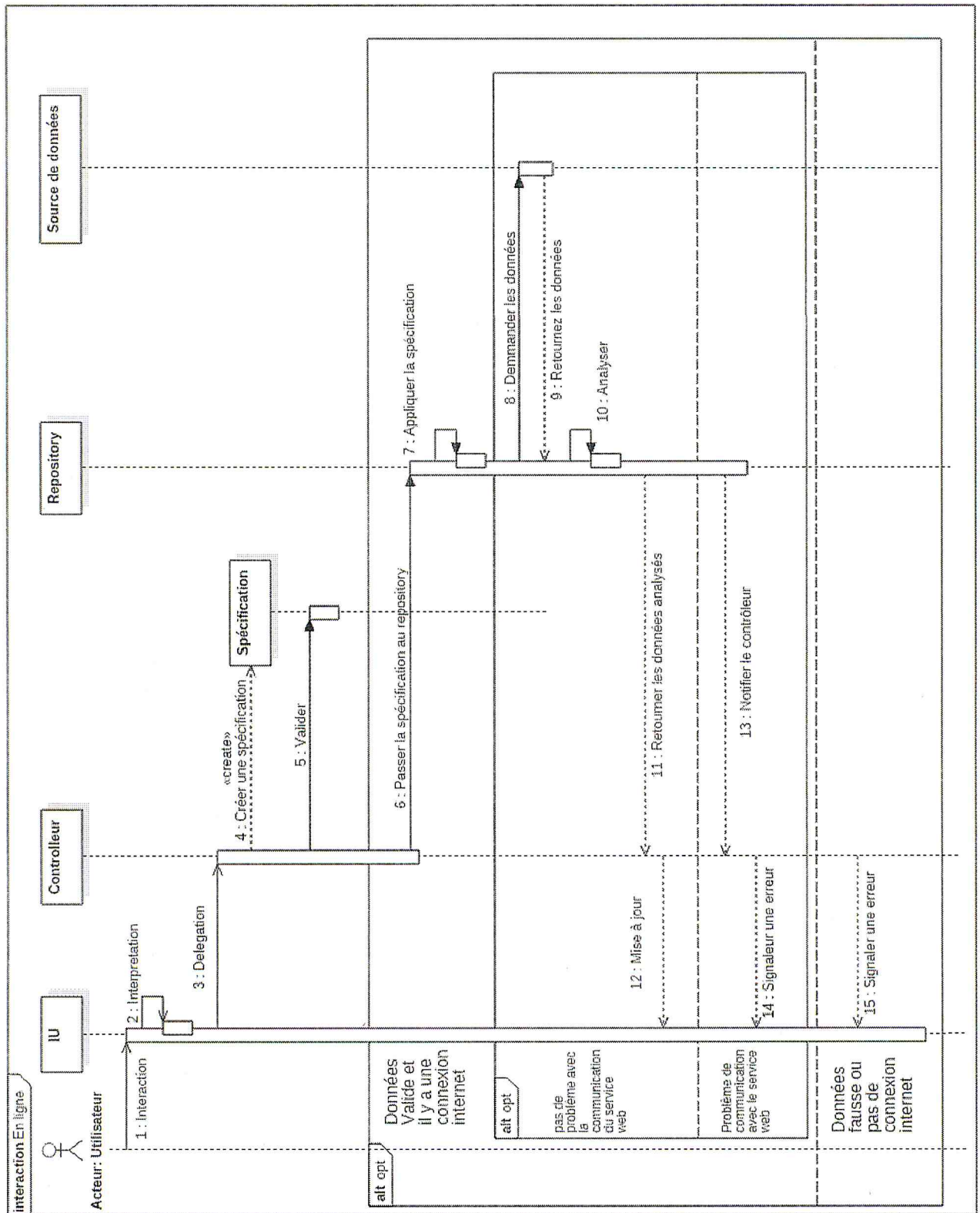


Figure 13. Diagramme de séquences des cas d'utilisation connectés.

7.2.2. Diagramme de séquence de « Lister les rappels »

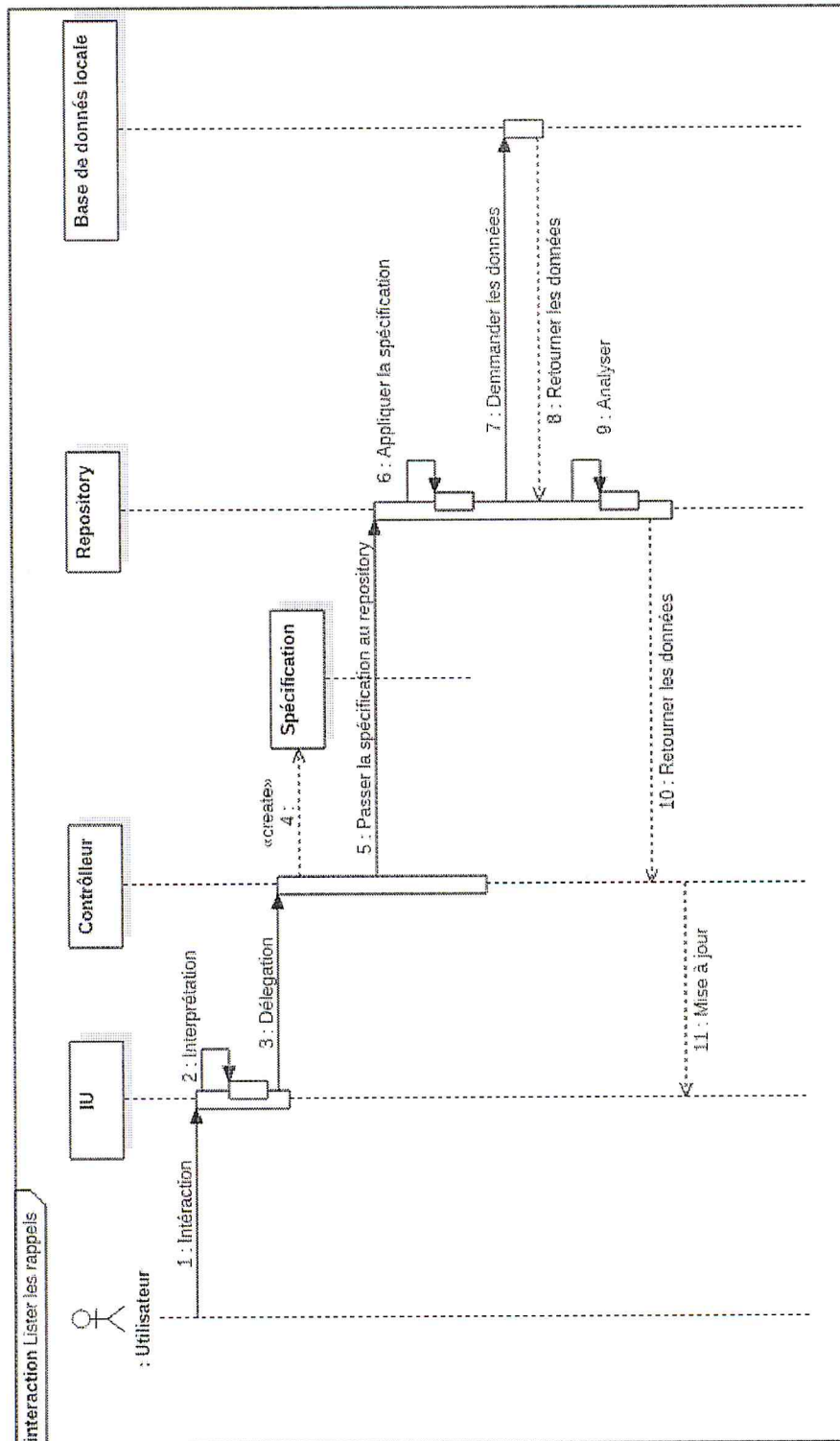


Figure 14. Diagramme de séquence de cas d'utilisation de « Lister les rappels ».

7.2.3. Diagramme de séquence de « Ajouter un rappel »

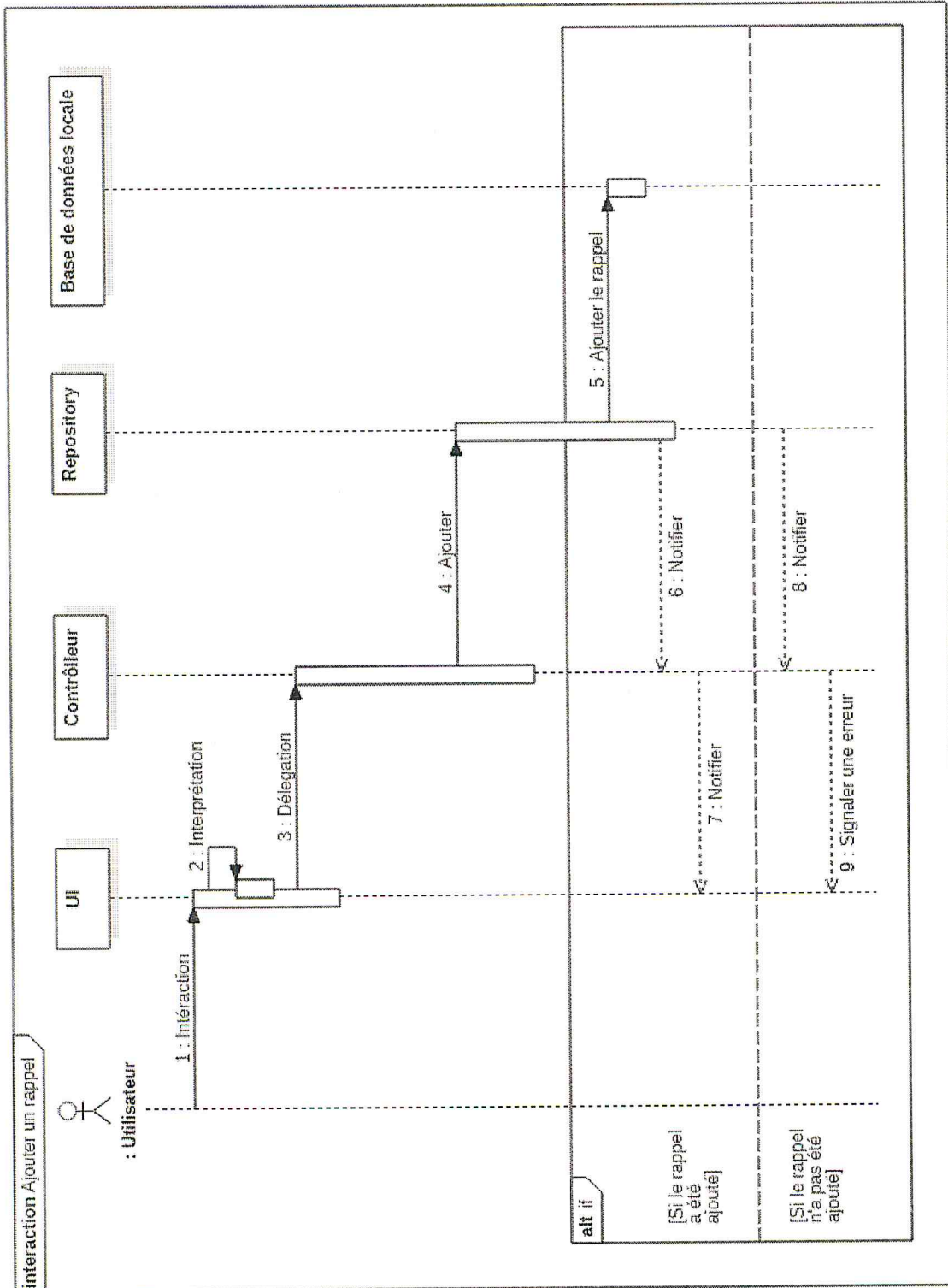


Figure 15. Diagramme de séquence de cas d'utilisation « Ajouter un rappel »

7.2.4. Diagramme de séquence de « Modifier un rappels »

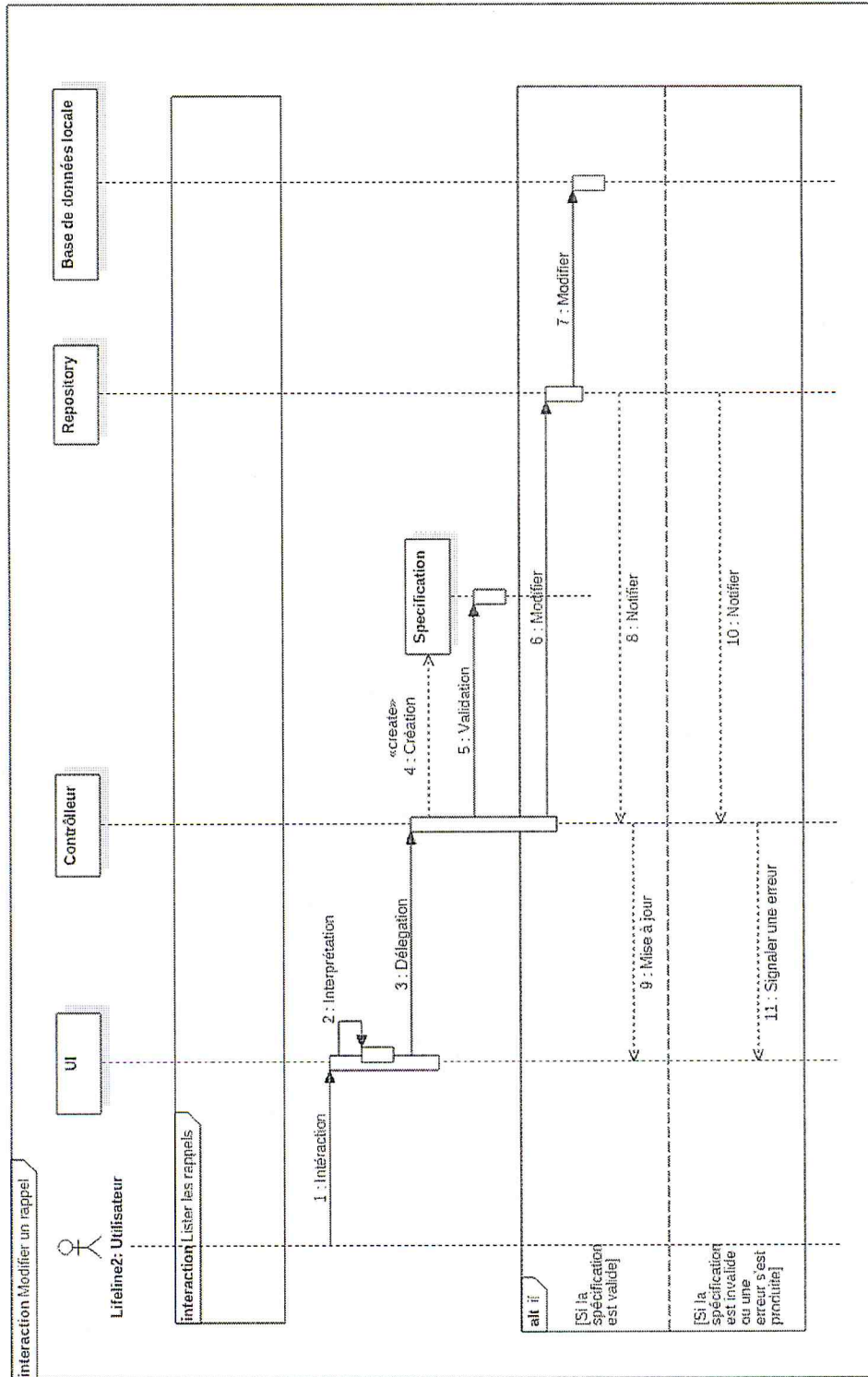


Figure 16. Diagramme de séquence de cas d'utilisation de « Modifier un rappels »

7.2.5. Diagramme de séquence de « Supprimer un rappel » :

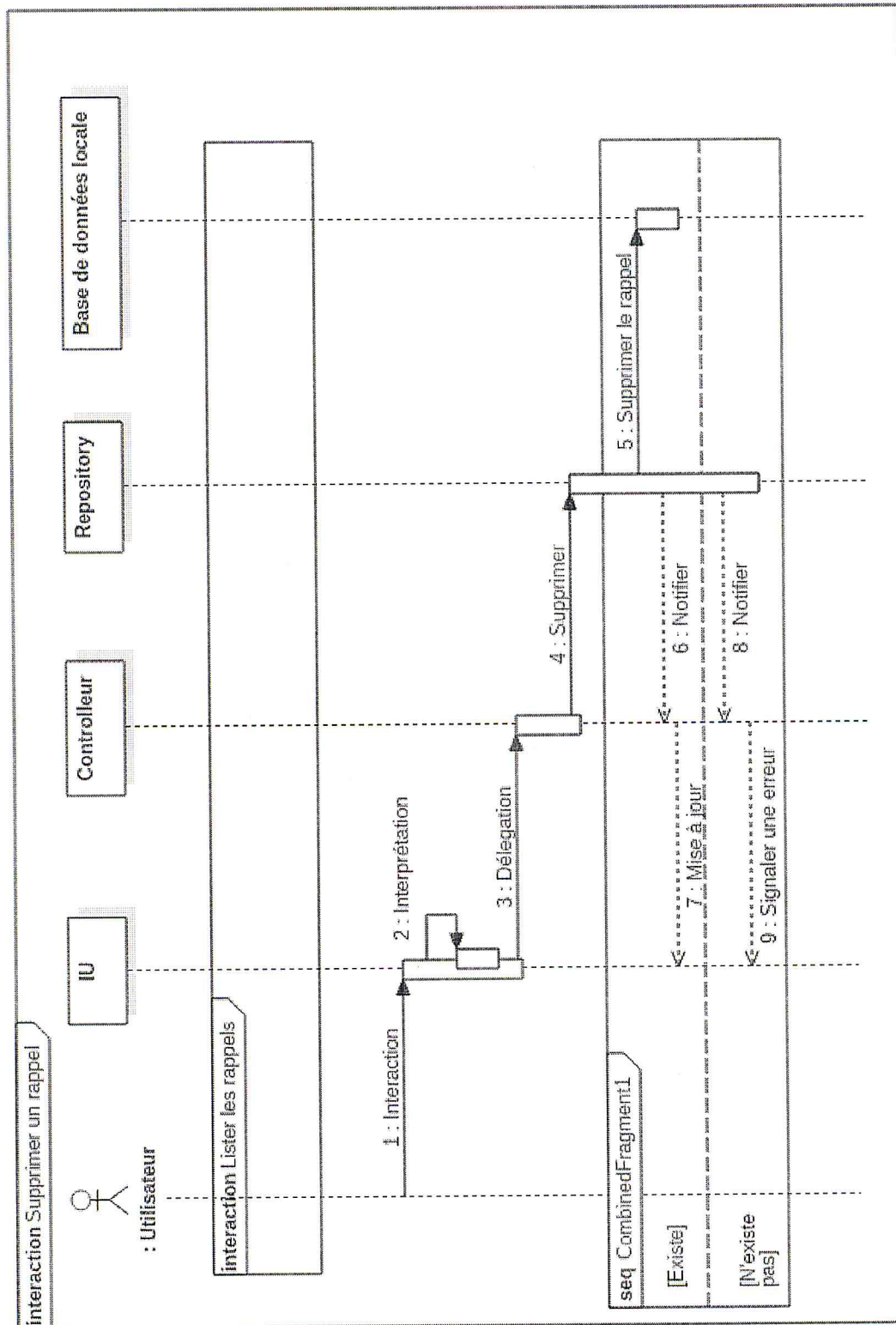


Figure 17. Diagramme de séquence de cas d'utilisation « Supprimer un rappel »

7.2.6. Diagramme de séquence de «Contacter la société»

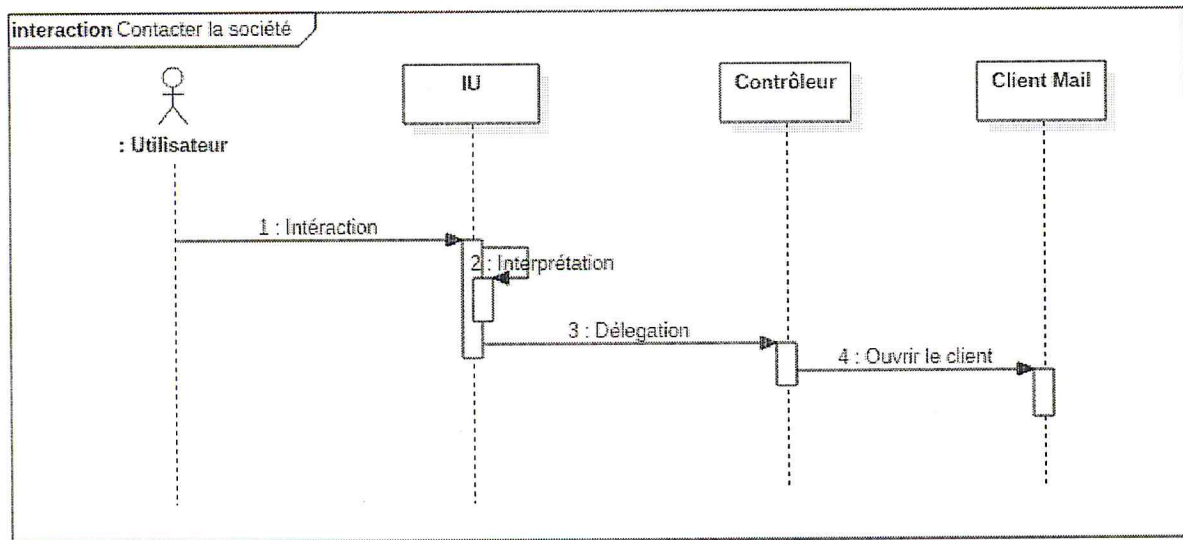


Figure 18. Diagramme de séquence de cas d'utilisation «Contacter la société»

7.2.7. Diagramme de séquence de « Convertir l'argent »

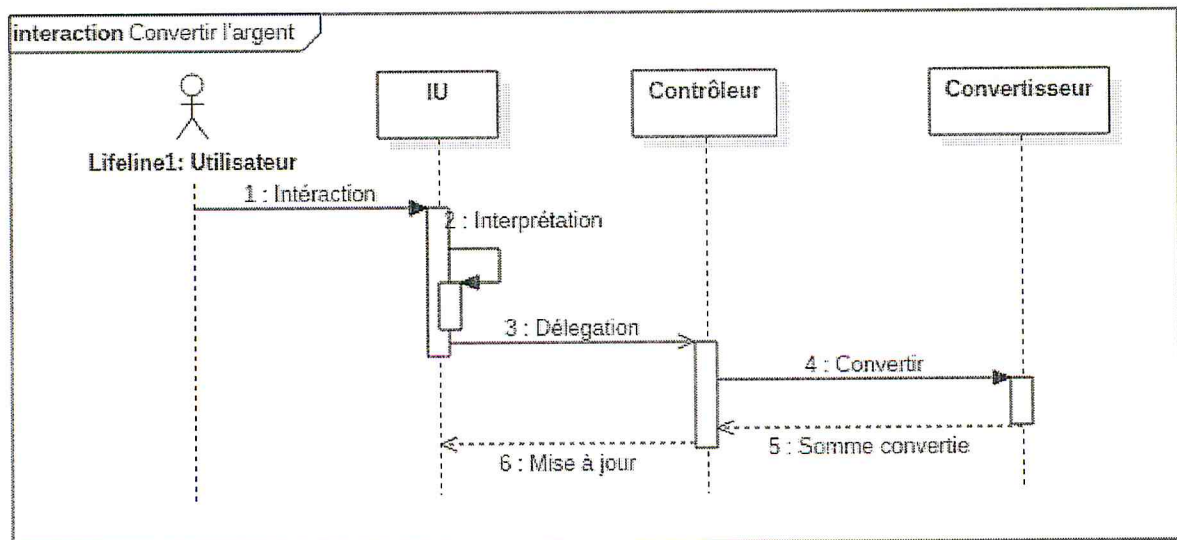


Figure 19. Diagramme de séquence de cas d'utilisation «Convertir l'argent»

7.3. Diagrammes de classes

7.3.1. Diagramme de classe de l'application

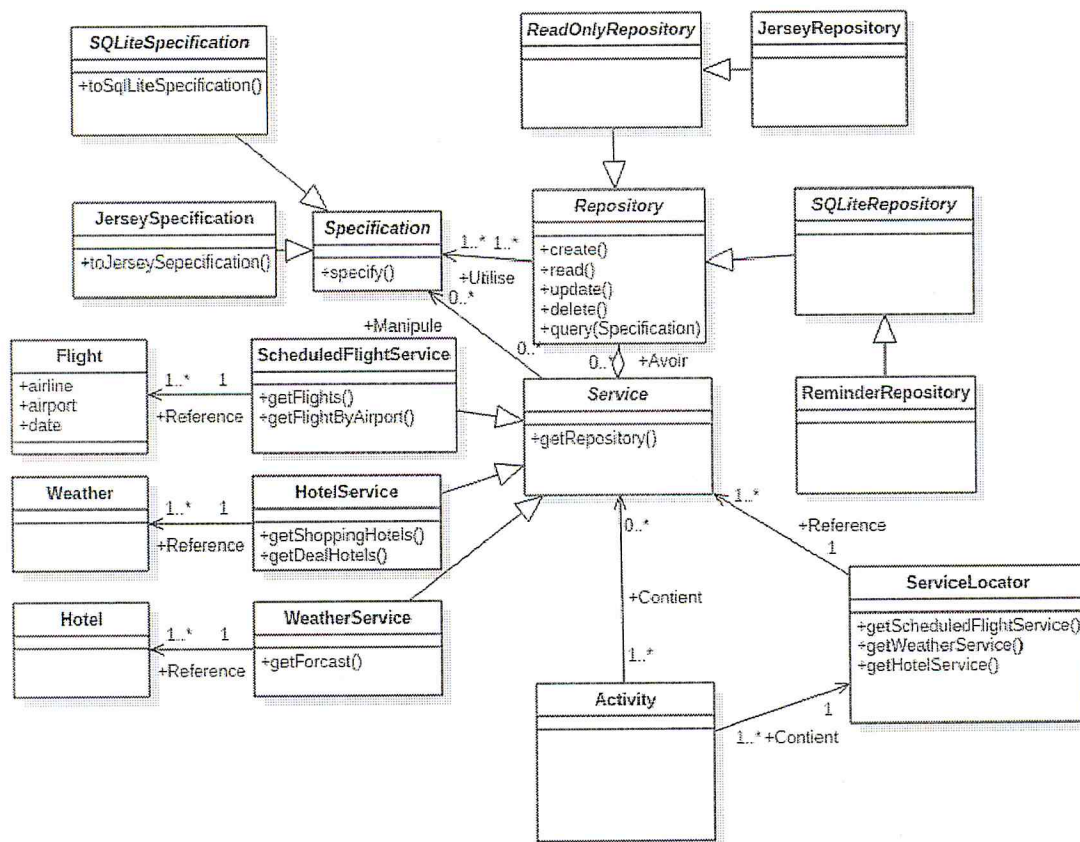


Figure 20. Diagramme de classe de l'application sous Android.

Le diagramme de classe, ci-dessus, de la conception de l'application sous Android. Cette conception est basée sur trois design patterns (Patrons de conception) **Repository**, **Spécification** et **Service Locator**, les deux premiers sont issus de DDD (Domain Driven Design) ou (Conception pilotée par le domaine), le dernier est un design pattern apparu la première fois dans le livre «GoF» [5, 7].

- **Repository** : C'est l'interface de design pattern Repository, la classe JerseyRepository implémente cette interface, cette interface avec toutes ses dérivées et ses implémentations représentent la couche persistance.

- **Spécification** : C'est l'interface de design pattern Spécification, l'interface JerseySpecification hérite cette interface et ajoute la capacité de faire des critères à l'aide de Jersey, cette interface avec toutes ses dérivés et ses implémentations représentent la couche persistance et la couche métier.
- **Service** : C'est une classe abstraite qui réunit tous les autres services. Chaque service encapsule un ensemble de fonctionnalités et les traite d'une façon indépendante aux couches supérieures. Cette classe avec toutes ces dérivées et ses implémentations représente la couche métier.
- **ServiceLocator** : Cette classe implémente le design pattern ServiceLocator afin de faciliter le référencement des services depuis les Activity à cause des limites techniques (performances) sur la plateforme Android, cette classe est dans l'infrastructure transversale.
- **Activity** : C'est une classe de la plateforme Android.

7.3.2. Diagramme de classe de la base locale des données :

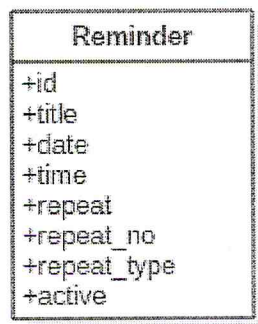


Figure 21. Diagramme de classe de la base locale de données.

7.3.2.1 Description des champs :

id	L'ID de l'alerte.
title	Le titre de l'alerte.
date	La date de déclenchement de l'alerte.
time	Le temps de déclenchement de l'alerte.
active	Un drapeau qui pour connaitre si l'alerte est active ou non.

Tableau 12. Description des champs de la base locale de données.

7.4. Isolation des couches :

Les couches de l'application (Présentation, Métier et Persistance) semblent très isolées depuis une vue architecturale, en réalité les couches sont un peu entrelacées entre eux à cause de plusieurs raisons (Plateforme, langage, typage, programmation par contrat ...), mais la présence d'un couplage faible entre les couches réduit l'effet d'entrelacement des couches et les rend facile à réutilisées dans un autre contexte qui est partiellement similaire à ce contexte (une application identique mais en web au lieu en mobile), par exemple : la couche persistance est très flexible, elle peut être utilisée dans plusieurs autres contextes qui nécessitent un accès aux objets persistés peu importe les exigences fonctionnelles du contexte, par contre la couche métier est liée au métier, nous ne pouvons pas l'utiliser dans un autre contexte qui n'exige pas les mêmes besoins, mais elle peut être facilement utilisée dans une autre application qui résout les même problèmes.

Les avantages de cette conception :

Cette conception a été créée en respectant les principes en or de l'Orienté Object comme OOAD et les principes SOLID qui donnent une flexibilité, maintenabilité et une clarté de code source.

- L'utilisation de design pattern Repository offre une très grande flexibilité et une abstraction sur les objets persistés, par exemple : si nous voulons implémenter un système de cache dans la mémoire de téléphone pour les objets déjà lu depuis la source de données, nous devons juste créer une nouvelle classe appelée **InMemoryRepository**, autre interface **InMemorySpecification** et nous devons adapter toutes les spécifications concrètes créées pour supporter la nouvelle spécification.
 - La possibilité d'un service d'utiliser plusieurs Repository sans en connaître le moindre détail, par exemple : un Service peut utiliser un Repository de service web basé sur Jersey et un autre Repository pour le cache basé sur une implémentation sur la mémoire.
 - Séparation claire des concerns dont chaque classe à une seule responsabilité.
-

8. Conclusion

Nous avons commencé le chapitre par l'analyse de besoin de notre système, puis nous avons montré l'architecture globale adapté aux besoins du système ensuite nous avons terminé le chapitre par la conception détaillée du système.

Parmi les difficultés que nous avons rencontrées dans ce chapitre : le choix de l'architecture globale de l'application, et, comment choisir et surtout comment combiner les techniques disponibles en Orienté Objet afin de faire une conception qui répondre aux exigences de nos jours et surtout qu'elle soit stable aux exigences de futur, notamment la facilité d'ajouter des nouvelles fonctionnalités sans modifier ou toucher le code qui existe et fonctionne déjà.

Après la lecture de plusieurs styles architecturaux nous avons choisi l'architecture multicouche qui s'adapte parfaitement à notre système. Afin d'atteindre ce but, nous avons respecté à la lettre les principes de l'Orienté-Objet (SOLID, KISS, DRY, STUPID etc.) et nous avons aussi utilisé certaines techniques avancées comme les design patterns de DDD (Spécification et Repository).

Chapitre V :
Réalisation

1. Introduction

Dans ce chapitre, nous présentons l'environnement matériel et logiciel du projet. Ensuite, nous nous intéressons à la description de quelques interfaces du système implémenté dans le cadre de quelques scénarios d'utilisation.

2. Environnement de travail

2.1. Environnement matériel

Machine de développement : ordinateur portable HP, Intel® Core® i7, 6 Go de RAM.

2.2. Environnement logiciel

Système d'exploitation : Ubuntu 16.04 LTS.

IDE de développement : Android Studio 2.1.

Outil pour la conception : Star UML.

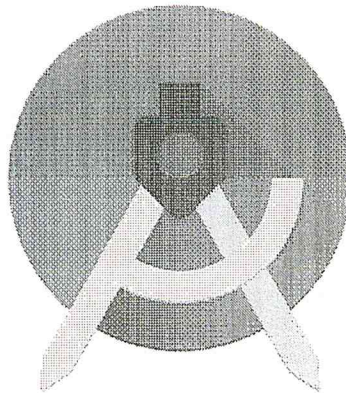


Figure 22. Le logo de l'Android Studio.

Nous avons intérêt à mettre en place une architecture rigoureuse, de manière à garantir la maintenabilité, l'évolutivité et l'exploitabilité de notre application.

2.2.1. La plateforme «Android»

2.2.1.1. Présentation

Est un système d'exploitation mobile basé sur le noyau Linux et développé actuellement par Google. 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

(Android TV), les voitures (Android Auto), les ordinateurs (Android-x86) et les smartwatch (Android Wear). Le système a été lancé en juin 2007 à la suite du rachat par Google en 2005 de la startup du même nom. En 2015, Android est le système d'exploitation le plus utilisé dans le monde avec plus de 80 % de parts de marché dans les smartphones. [4]

2.2.2. Jersey

2.2.2.1. Présentation

Jersey est un framework open-source écrit en langage Java permettant de développer des services web selon l'architecture REST suivant les spécifications de JAX-RS (JSR 311 et JSR 339).

[URL 18]

2.2.2.2. Fonctionnalité

- **API client**

L'API client de Jersey permet l'interopérabilité avec des services web REST. L'objectif de cette API est d'encapsuler une contrainte de clé de l'architecture REST coté client, faciliter l'interopérabilité avec les services web et être le pendant coté client de l'API (côté serveur) JAX-RS. [URL 18]

- **WADL (Web Application Description Language)**

WADL est un document XML qui définit un vocabulaire permettant de décrire les applications web. WADL a pour but d'être pour REST le pendant des fichiers WSDL de WS-*. Jersey génère automatiquement un document WADL qui inclut les ressources de la racine ainsi que le maximum de métadonnées, qui sont extraites des classes (Les annotations de JAX-RS).

[URL 18]

- **Formats de données**

Jersey supporte plusieurs formats parmi lesquels Atom, JSON et MIME Multipart data.

[URL 18]

- **Autres**

Jersey supporte le modèle Modèle-Vue-Contrôleur (MVC) et fournit également un framework permettant aux développeurs d'injecter des dépendances (supporte EJB, Spring et Guice).

[URL 18]

2.2.3. La bibliothèque Google Guava

La bibliothèque **Guava** a été créée chez Google pour répondre aux besoins fréquemment rencontrés par les équipes Java du géant, et ainsi avoir des réponses standardisées et performantes sur l'ensemble des projets. [URL 19]

2.2.4. La bibliothèque Joda Time

La représentation et la manipulation des dates et heures est un point notoirement faible de Java depuis sa version 1.0.

Il existe heureusement une petite librairie forte pratique pour combler ce vide sidérant : Joda Time. Développée par Stephen Colebourne, elle est en passe d'être standardisée au sein du JDK sous la JSR-310 (Date & Time API).

Joda Time fournit une API bien plus riche, répondant aux besoins concrets, toute la librairie est architecturée autour de 5 concepts :

- **Instant** : Date et heure complètes : 25/12/2012 00:00:00 (avec fuseau horaire)
- **Partial** : Date et/ou heure incomplètes, "flottantes" : 25/12 (Noël), 12:00 (midi)
- **Interval** : Intervalle entre des **Instants** de début et de fin
- **Duration** : Durée précise, en millisecondes
- **Period** : Représentation "humaine" d'une durée : 1 mois (qu'il ait 28 ou 31 jours)

[URL 20]

2.2.5. La bibliothèque Material Design Support Library

Introduit lors de la Google IO 2014, le Material Design est la nouvelle norme graphique des applications Android. Malheureusement, depuis cette dernière conférence, très peu de modules officiels permettent de l'intégrer facilement, nous avons l'habitude d'utiliser des modules open-source afin de reproduire chaque composant.

Surprise totale, lors de la Google IO 2015, est annoncé une bibliothèque de support permettant de réaliser facilement une belle application, en embarquant une bonne partie des nouveaux composants et layouts qui font l'identité du Material Design, tel que le Floating Action Button ou les Collapsing toolbar. [URL 21]

2.2.6. FlightStats

FlightStats est une entreprise de services de données de l'aviation commerciale et le principal fournisseur de données en temps réel vol global aux entreprises à travers l'écosystème Voyage.

[URL 22]

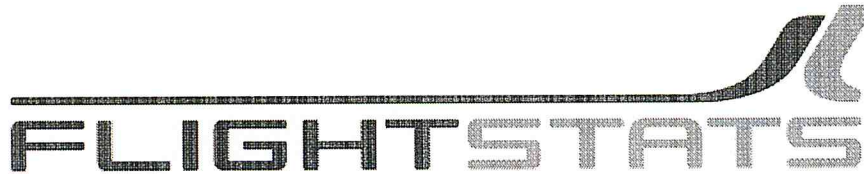


Figure 23. Le logo de FlightStats.

2.2.7. Hotwire

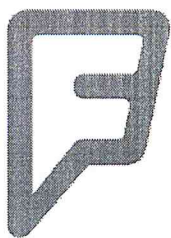
Hotwire est une petite marque qui a été alimenté par une communauté de voyageurs avertis depuis 2000, spécialisé dans les services d'hôtellerie et d'hébergement. [URL 23]



Figure 24. Le logo de Hotwire.

2.2.8. Foursquare

Foursquare est une société de technologie qui tire parti des données de localisation pour créer des expériences valorisantes pour le consommateur et des solutions pour les entreprises. [URL 24]



FOURSQUARE

Figure 25. Le logo de Foursquare.

2.2.9. OpenWeatherMap

OpenWeatherMap est un service en ligne qui fournit une API libre pour les données météorologiques, y compris les données actuelles météorologiques, des prévisions et des données historiques aux développeurs de services Web et des applications mobiles.

[URL 25]



Figure 26. Le logo d'OpenWeatherMap.

2.2.10. OpenExchangeRates

Une API simple, léger et portable qui fournit un échange en direct et l'historique de taux de change pour 170 devises mondiales en format JSON. Les données sont suivies et mélangés algorithmiquement à partir de plusieurs sources fiables, en assurant la cohérence équitable et impartiale. [URL 26]



Figure 27. Le logo d'OpenExchangeRates.

2.2.11. Amadeus

Amadeus est un environnement pour aider à transformer les idées de voyage en grands prototypes de travail, elle offre un accès à bien documentées API REST / JSON soutenues par des services d'Amadeus.

[URL 27]

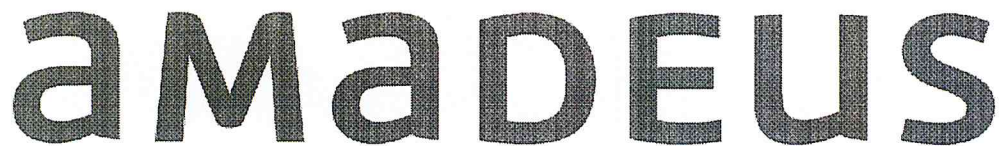


Figure 28. Le logo d'Amadeus.

3. Interfaces de l'application

Nous exposerons quelques interfaces de notre application, en essayant à chaque fois de décrire les différents objets interactifs mis à la disposition de l'utilisateur.

- Lors du lancement de l'application, une interface SplashScreen (Figure 31) apparaît

Mentionnant le nom de l'application.

- Après 4 secondes l'interface accueil s'affichera (Figure 32).



Figure 29. Le splashscreen.

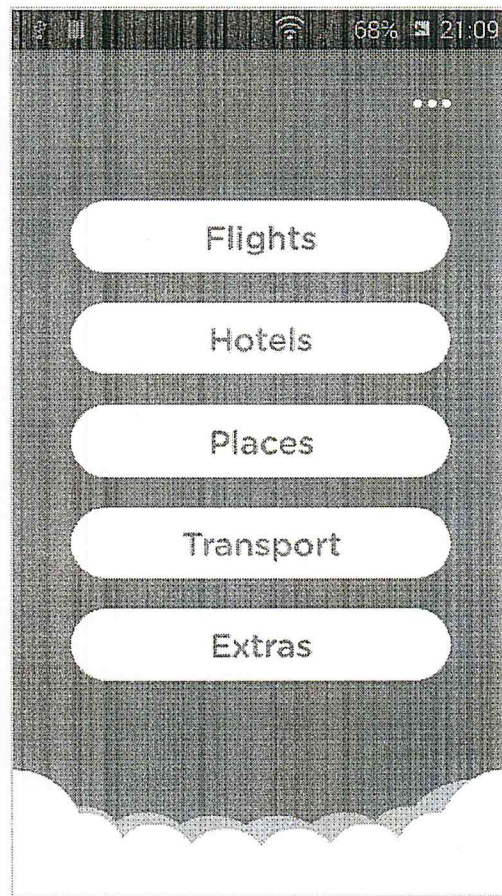


Figure 30. L'accueil.

- Lorsque l'utilisateur se click sur le bouton "Flights" pour chercher sur les vols 3 type de recherche seront afficher : recherche par vol, par aéroport ou par route, il va choisir un type selon ses besoin, ce cas est illustre dans la Figure 33.

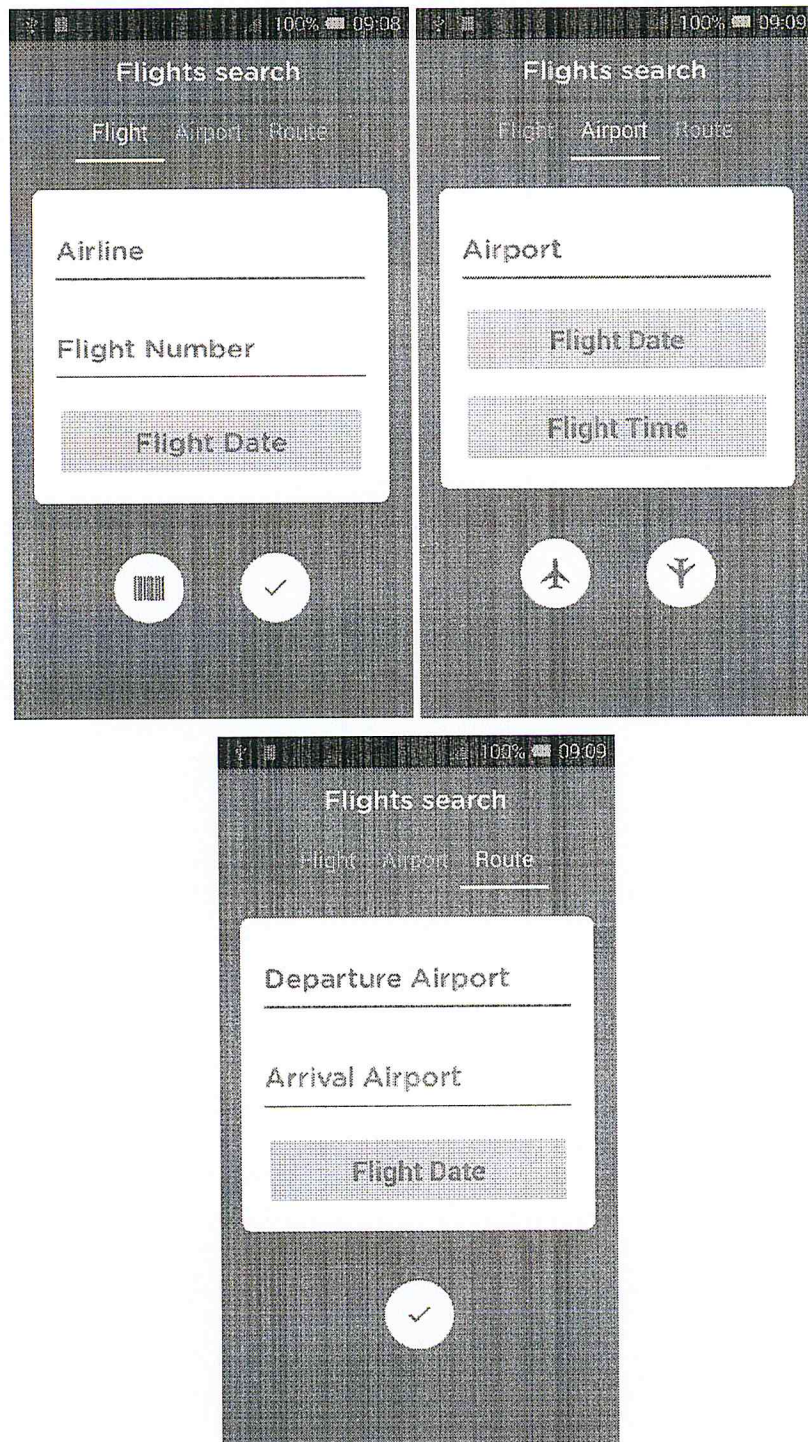


Figure 31. La recherche des vols par vol, par aéroport ou par route.

- Les résultats de la recherche par vol pour le vol AIR ALGERIE, 1000, le 4/6/2016 est illustre dans la Figure 34

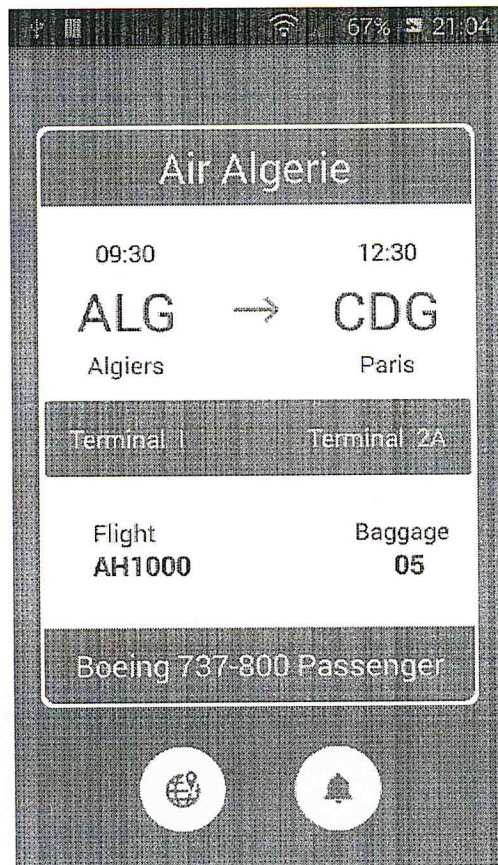


Figure 32. Le résultat de la recherche par vol.

- Lorsque l'utilisateur se click sur le bouton "scan" dans l'interface rechercher par vol, une nouvelle interface va apparaitre donc l'utilisateur doit que cliquer sur le bouton "scan the code" pour lancer la camera et scanner le code de billet d'avion, ce cas est illustre dans la Figure 35.

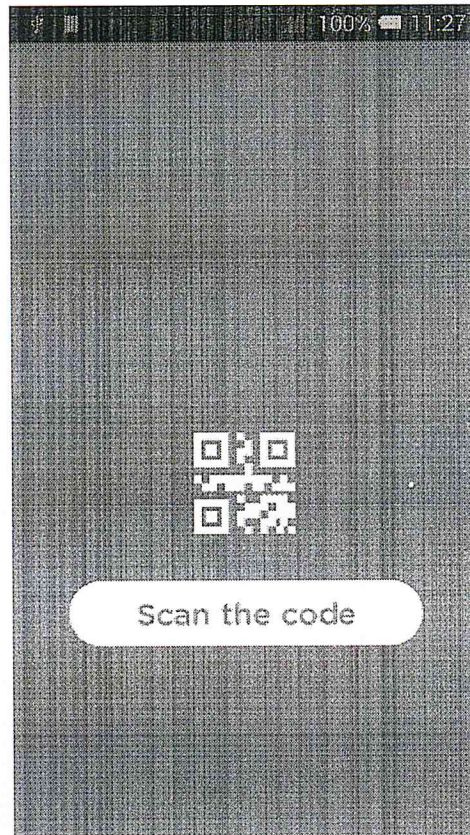


Figure 33. L'interface scanner des billets d'avion.

- Lorsque l'utilisateur se click sur le bouton "Hôtels" pour chercher sur les hôtels 2 type de recherche seront afficher : recherche simple, recherche des offres, il va choisir un type selon ses besoin, ce cas est illustre dans la Figure 36.

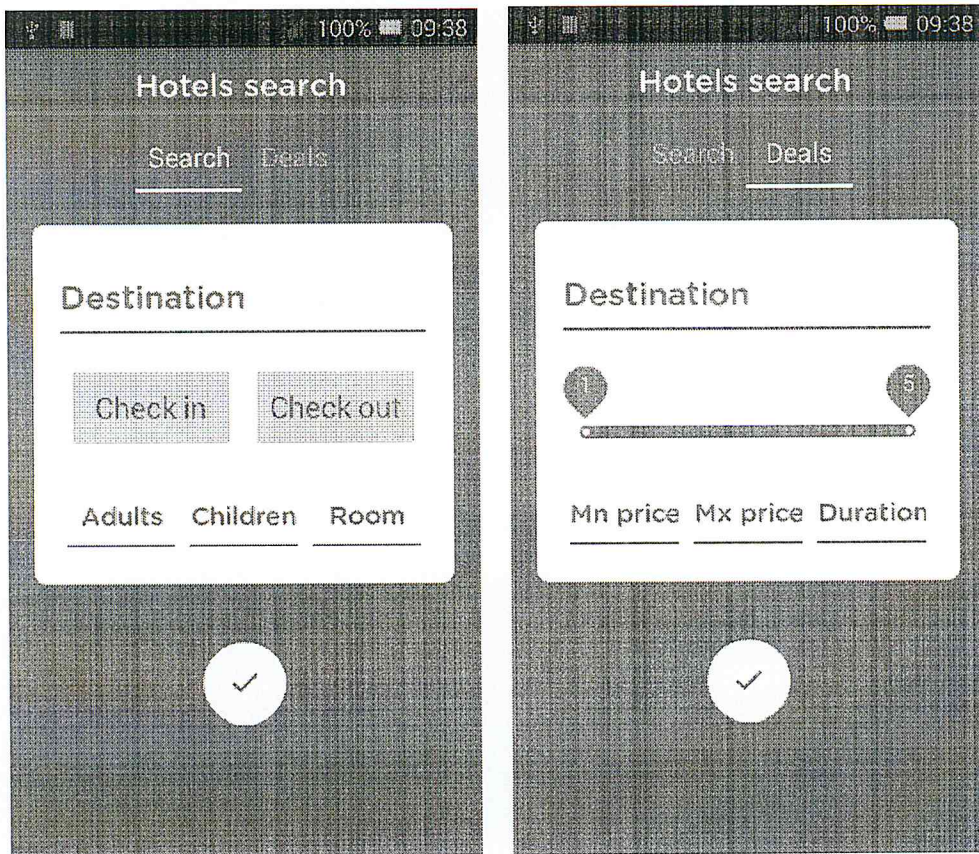


Figure 34. La recherche des hôtels simple ou par offres.

- Les résultats de la recherche simple des hôtels pour la destination BAGHDAD a partir de 5/6/2016 jusqu'a le 7/6/2016 pour 2 Adultes, 4 Enfants et 3 Chambres est illustre dans la Figure 37.

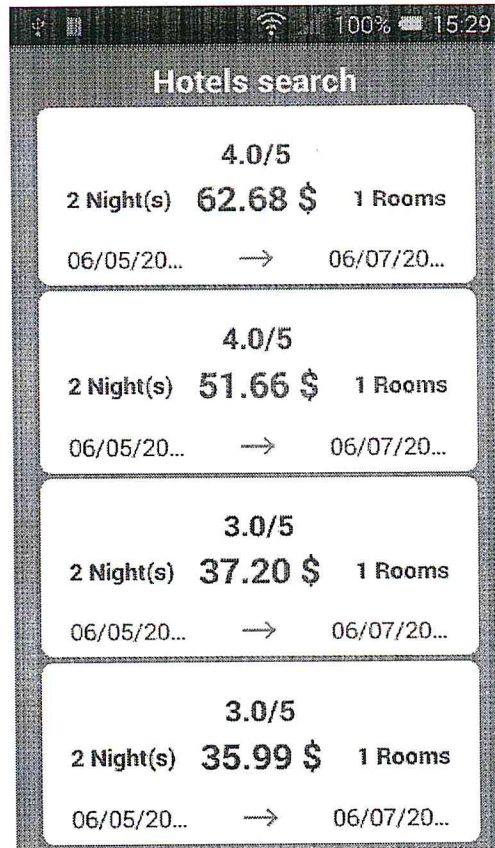


Figure 35. Le résultat de la recherche simple des hôtels.

- Lorsque l'utilisateur se click sur le bouton "Places" pour chercher sur les spots ou les endroits touristiques, les restaurant et même les jardins public, 2 type de recherche seront afficher : recherche des places les plus proches, recherche des places les plus populaire, il va choisir un type selon ses besoin, ce cas est illustre dans la Figure 38.

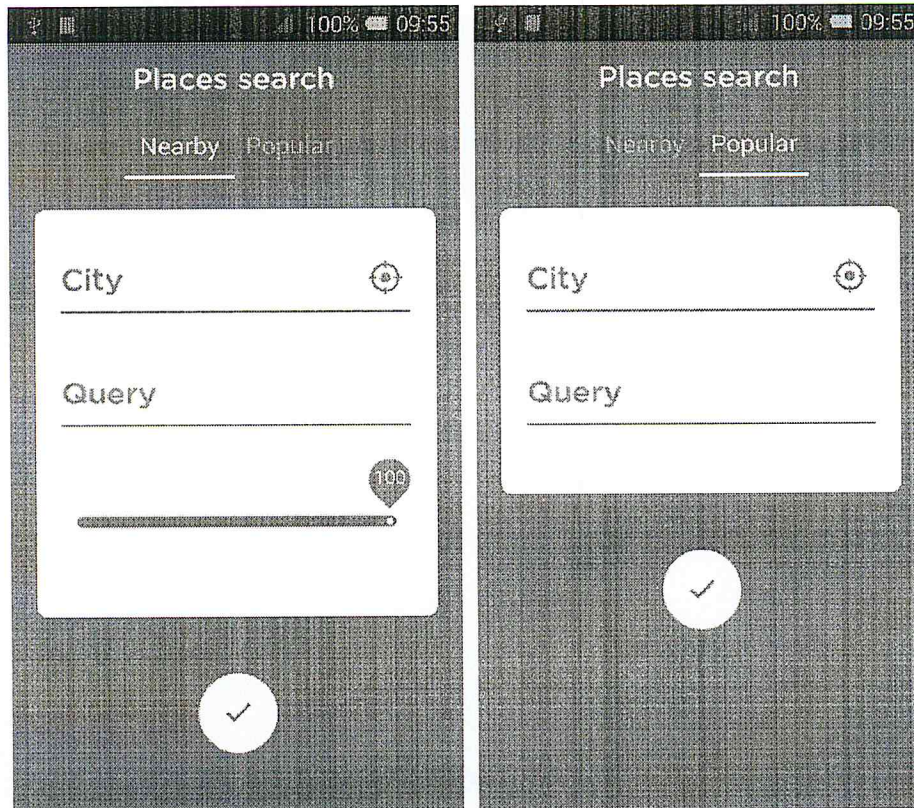


Figure 36. La recherche des places par la plus proche ou par la plus populaire.

- Les résultats de la recherche des places par la plus proche pour la ville d'ALGER pour le terme PIZZA est illustre dans la Figure 39.



Figure 37. Le résultat de la recherche des places les plus proche.

- Les résultats de la recherche des places par la plus populaire pour la ville d'ALGER pour le terme PIZZA est illustre dans la Figure 40 (les mêmes critères mais des résultats différents).

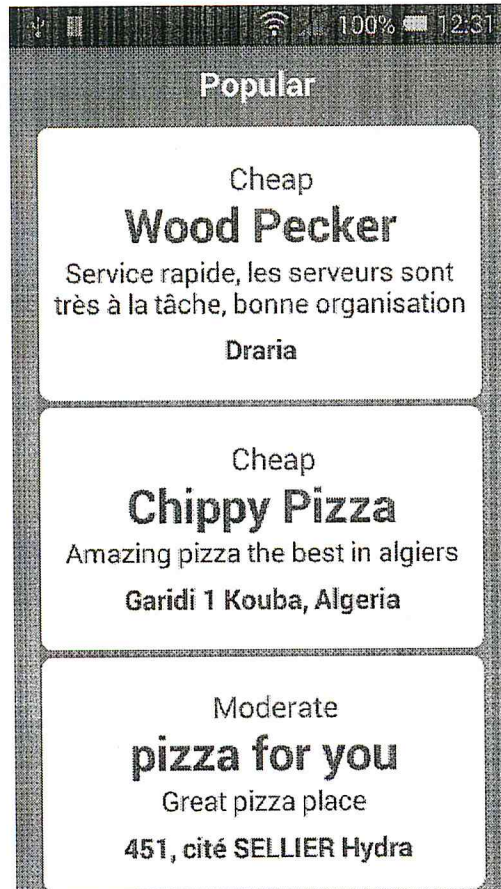


Figure 38. Le résultat de la recherche des places les plus populaires.

- Lorsque l'utilisateur se click sur le bouton "Transport" pour chercher sur les moyen de transport terrestre, 2 type de recherche seront afficher : recherche des a louer ou chercher les trains, il va choisir un type selon ses besoin, ce cas est illustre dans la Figure 41.

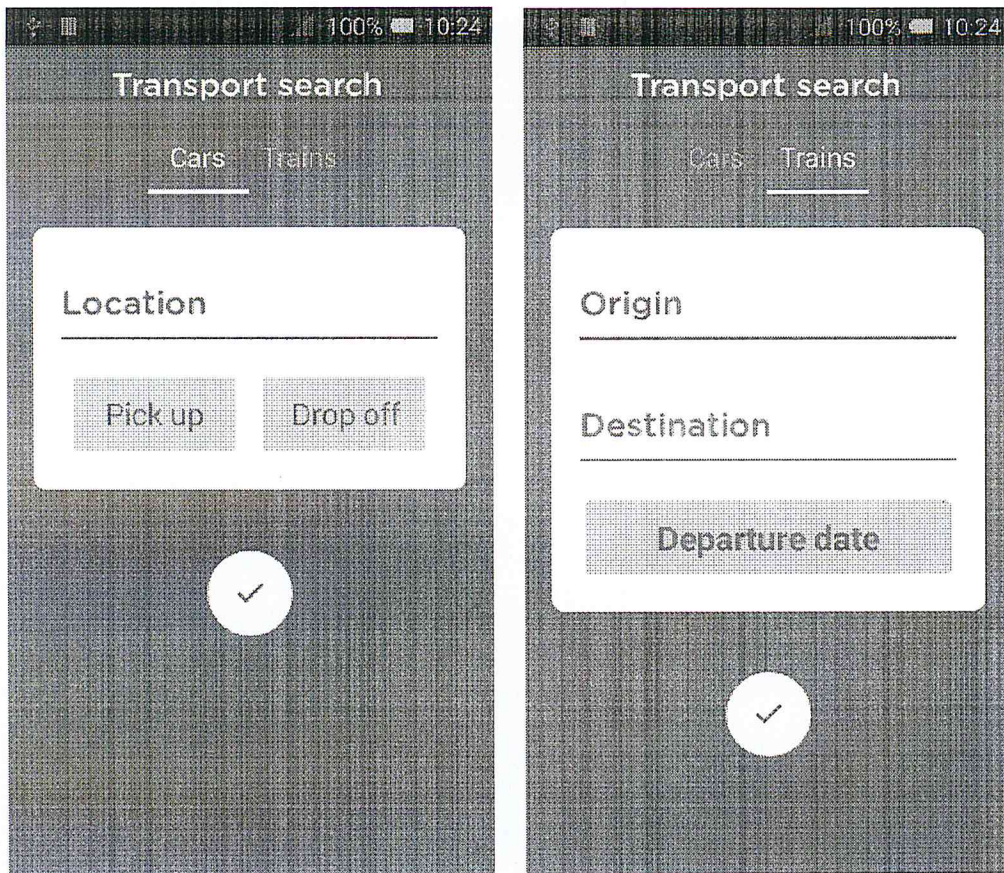


Figure 39. La recherche des moyens de transport : voiture à louer ou un train.

- Les résultats de la recherche des voitures à louer dans les environs de l'aéroport international de Amman à partir de 5/6/2016 jusqu'a le 8/6/2016 est illustre dans la Figure 42.

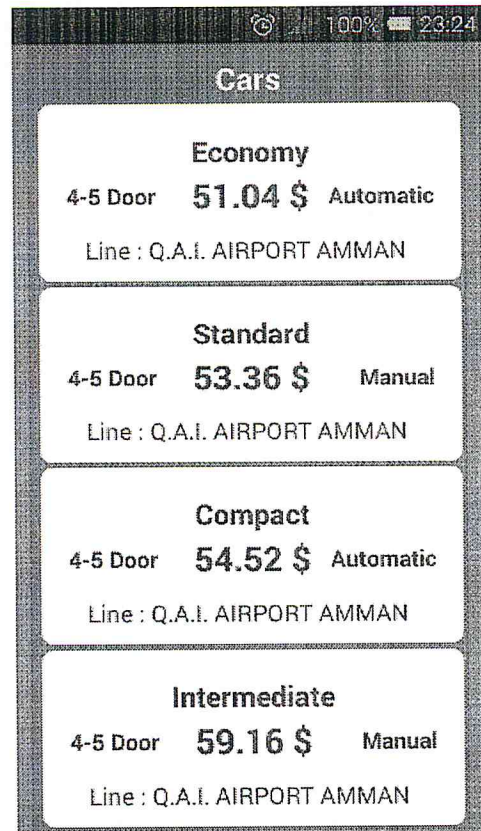


Figure 40. Le résultat de la recherche des places les plus populaires.

- Les résultats de la recherche des trains qui départ de la station BARCELONE a LYON pour le 5/6/2016 est illustre dans la Figure 43.



Figure 41. Le résultat de la recherche des trains.

- Lorsque l'utilisateur se click sur le bouton "Extras" pour chercher des informations supplémentaires, 3 type de recherche seront afficher : les prévisions de la météo, le convertisseur de temps et le convertisseur de devises, ce cas est illustre dans la Figure 44.

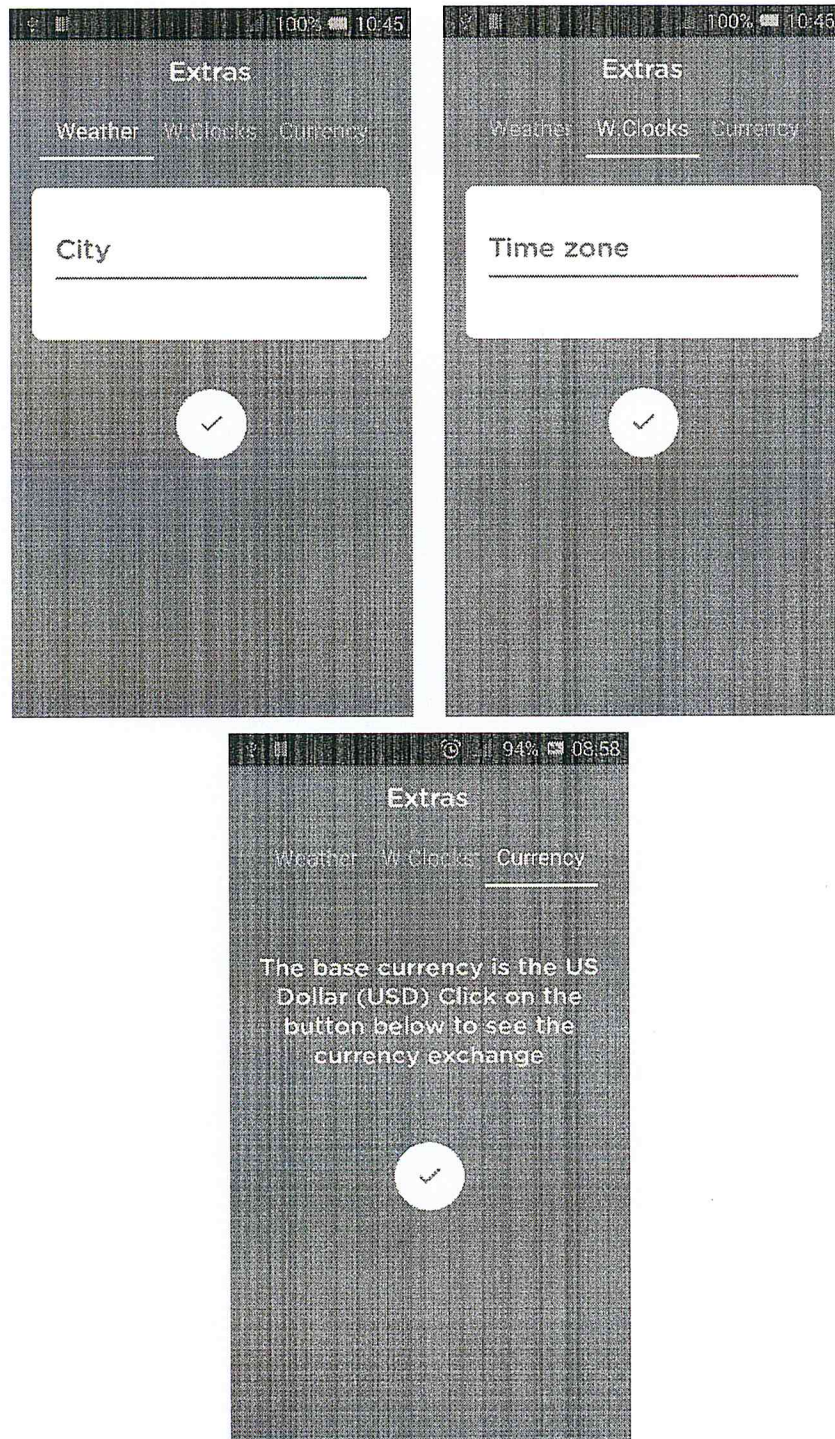


Figure 42. La recherche des infos supplémentaires : météo, temps, devises.

- Les résultats de la recherche de la prévision de météo de la ville de TRIPOLI, est illustre dans la Figure 45.

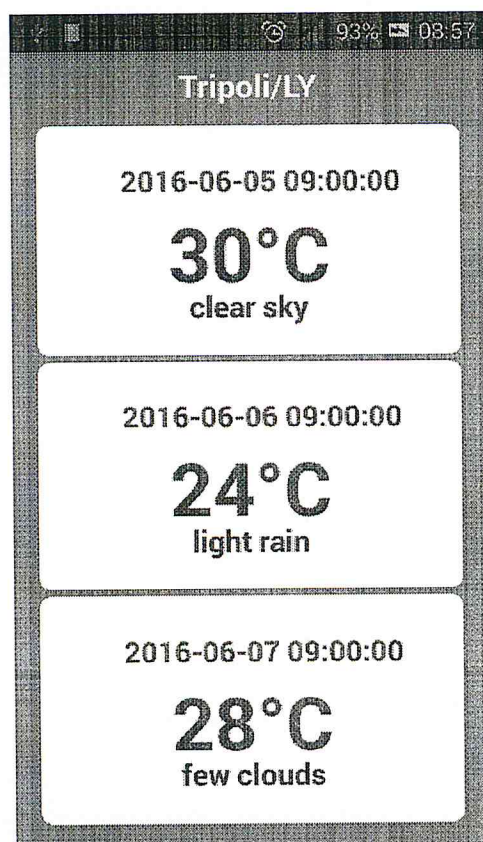


Figure 43. Le résultat de la recherche de la prévision de météo.

- Les résultats de la recherche de convertisseur de temps pour la ville de DAMAS, est illustre dans la Figure 46.

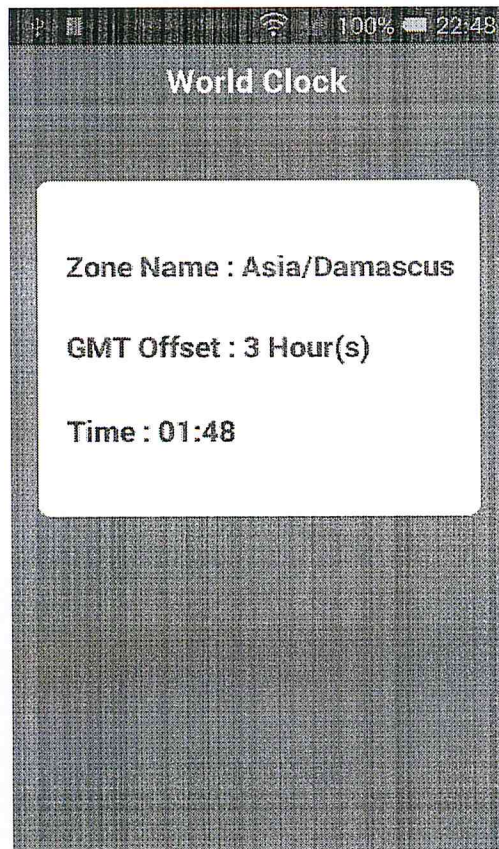


Figure 44. Le résultat de la recherche de convertisseur de temps.

- Les résultats de la recherche de convertisseur de devises pour (1.00) DOLLARS AMÉRICAIN aux autres devises du monde, est illustre dans la Figure 47.



Figure 45. Le résultat de la recherche de convertisseur de devises.

- Lorsque l'utilisateur se click sur les trois points dans l'interface d'accueil pour accéder à l'interface paramètre de l'application, ce cas est illustre dans la Figure 48.

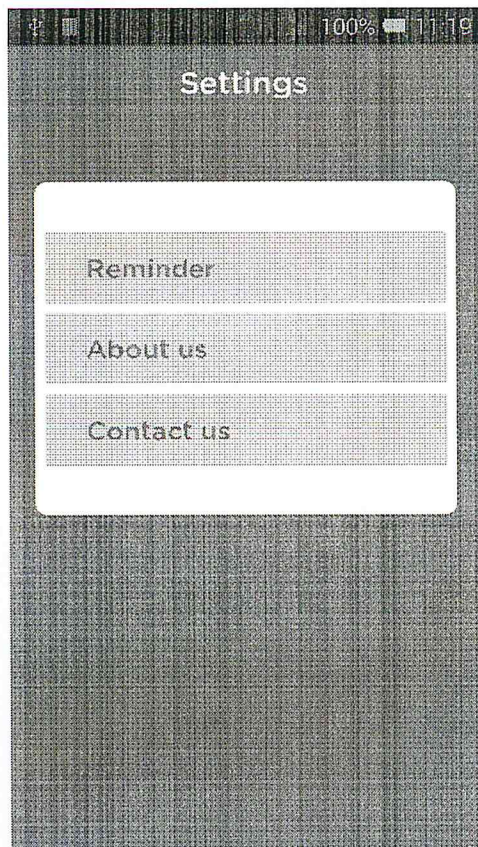


Figure 46. L'interface paramètre de l'application.

- Lorsque l'utilisateur se click sur le bouton "Reminder" dans l'interface paramètre de l'application pour ajouter ou vérifier un rappel supplémentaires, une nouvelle interface sera afficher avec les rappels qui ont déjà enregistré, ce cas est illustre dans la Figure 49.

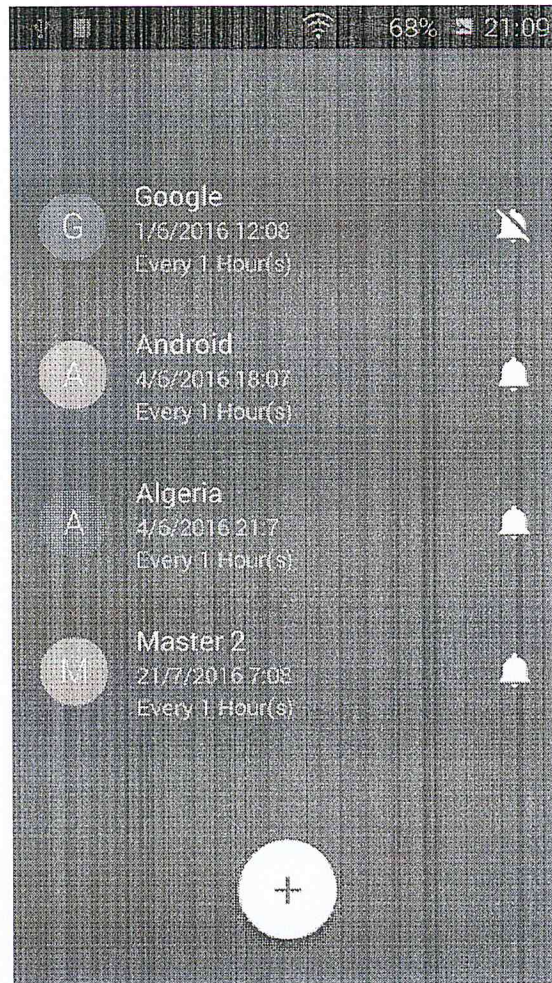


Figure 47. L'interface reminder.

- Lorsque l'utilisateur se click sur le bouton "+" une nouvelle interface sera afficher pour lui permettre d'ajouter un nouveau rappel et si il a clicker sur un rappel déjà existe cela lui permettre de faire des modifications sur ce rappel, ce cas est illustre dans la Figure 50.

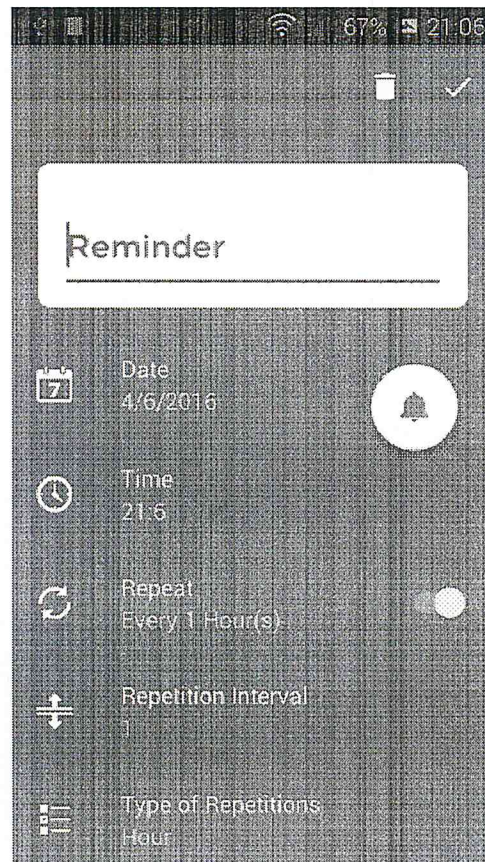


Figure 48. L'interface ajouter ou modifier un rappel.

- Lorsque l'utilisateur se click sur le bouton "About us" dans l'interface paramètre de l'application, une nouvelle interface qui montre ceux qui ont réalisé l'application sera afficher, ce cas est illustre dans la Figure 51.

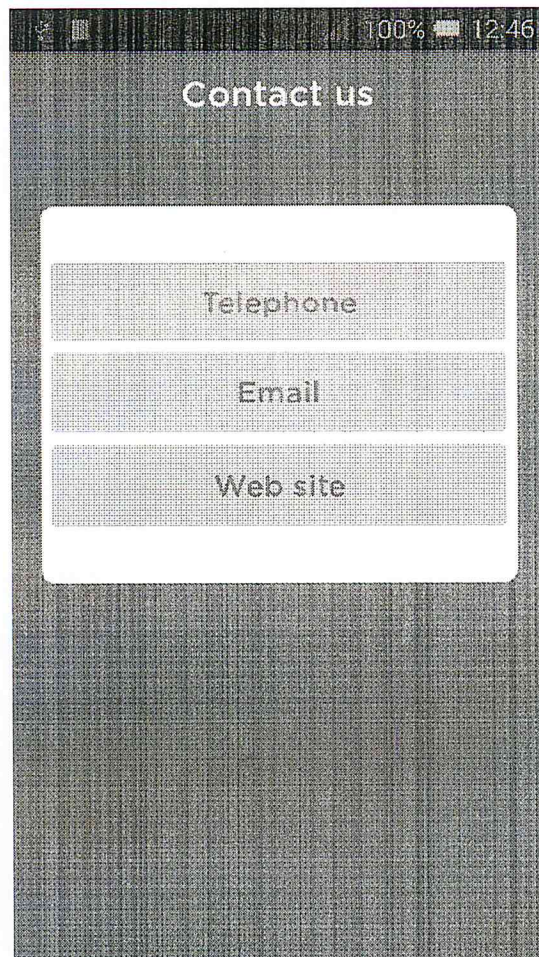


Figure 49. L'interface contacté nous.

4. Conclusion

Dans ce chapitre nous avons détaillé les technologies utilisées pour la réalisation de notre projet ainsi que les fonctionnalités de base de l'application à travers un ensemble de captures d'écran.

Conclusion Générale

A l'issue de la réalisation de ce travail, nous avons pu affirmer que notre projet a été d'une grande utilité dans la mesure où il nous a permis de familiariser bien avec le travail sur la plate-forme Android. Non seulement, les bénéfices ont été réalisés sur le plan technique mais aussi sur le plan social. Nous avons pu avoir un aperçu autour du travail au sein d'une entreprise nationale. L'intégration d'une équipe de travail a été une expérience qui marquera la période de réalisation du projet au sein de l'organisme d'accueil.

Au cours de la phase de réalisation de notre application, nous avons élaboré une étude préalable sur les smartphones et son importance sur le plan social afin de préciser le but principal pour la future application. Cette phase a constitué le point de départ pour l'étape d'analyse et de spécification des besoins. Une fois nos objectifs sont fixés nous avons enchaîné avec la conception afin de mener à bien notre projet. Nous avons procédé à la phase de réalisation au cours de laquelle nous avons familiarisé avec le langage de programmation Java.

Pour conclure, notre travail peut être un sujet à des extensions. Car on peut l'améliorer et à cause des difficultés techniques de la part des sociétés de transport et la mal organisation de transport terrestre (TAXI, Bus, etc.) et à cause du temps limite nous n'avons pas pu faire tout le travail que nous avons envisagé à faire, et nous voudrions bien mettre notre application sur Android Market (Google Play Store), Condor Store et Ooredoo Store, et nous avons des plans pour la rendre disponible sous d'autres plateformes comme IOS ou Windows Phone, et nous comptons persévérer pour la rendre meilleur et pratique et plus utile pour les voyageurs dans le monde entier.

Webographie

[URL 1] « ANPT », sur le site *ANPT*. Consulté le 24 Avril 2016.

<http://www.anpt.dz/>

[URL 2] « KAYAK - Cheap Flights, Hotels, Airline Tickets, Cheap Tickets, Cheap Travel Deals - Compare Hundreds of Travel Sites At Once », sur le site *Kayak*. Consulté le 24 Avril 2016.

<https://www.kayak.com/mobile>

[URL 3] « Home - Hotel Tonight », sur le site *Hotel Tonight*. Consulté le 24 Avril 2016.

<https://www.hoteltonight.com/fr/>

[URL 4] «Travel Apps: Best Hotel & Flight Booking Apps for your iPhone, Android & iPad | Expedia », sur le site *Expedia*. Consulté le 24 Avril 2016.

<https://www.expedia.com/app>

[URL 5] « L'histoire de Uber », sur le site *Uber*. Consulté le 24 Avril 2016.

<https://www.uber.com/our-story/>

[URL 6] « About TripAdvisor », sur le site *TripAdvisor*. Consulté le 24 Avril 2016.

https://www.tripadvisor.com/PressCenter-c6-About_Us.html

[URL 7] « Introduction aux systèmes d'exploitation mobile », sur le site *Slideshare*. Consulté le 24 Avril 2016.

<http://fr.slideshare.net/Rouinihouss/introduction-aux-systmes-dexploitation-mobile>

[URL 8] « Quel système d'exploitation mobile ? », sur le site *Toms Guide*. Consulté le 24 Avril 2016.

<http://www.tomsguide.fr/article/guide-smartphone-acheter,2-1219-3.html>

[URL 9] «What is Palm OS (Palm Operating System)? - Definition from WhatIs.com », sur le site *Search Mobile Computing*, consulté le 24 Avril 2016.

<http://searchmobilecomputing.techtarget.com/definition/Palm-OS>

[URL 10] « What is Symbian OS? – Definition », sur le site *Mobile Burn*, consulté le 24 Avril 2016.

<http://www.mobileburn.com/definition.jsp?term=Symbian+OS>

[URL 11] « Ubuntu for phones | Ubuntu », sur le site officiel Ubuntu, consulté le 24 Avril 2016.

<http://www.ubuntu.com/phone>

[URL 12] « About | Tizen », sur le site Tizen, consulté le 24 Avril 2016.

<https://www.tizen.org/about>

[URL 13] « Firefox OS — Just what you need — Great smartphone features, apps and more — Mozilla », sur le site officiel de Mozilla, consulté le 24 Avril 2016.

<https://www.mozilla.org/en-US/firefox/os/>

[URL 14] « IDC: Smartphone OS Market Share 2015, 2014, 2013, and 2012 », sur le site, consulté le 24 Avril.

<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>

[URL 15] « Android - Historique », sur le site officiel d'Android, consulté le 24 Avril 2016.

https://www.android.com/intl/fr_fr/history/#!/marshmallow

[URL 16] « Android definition final », sur le site slideshare, consulté le 24 Avril 2016.

<http://fr.slideshare.net/MohamedMohamed81/android-definition-final>

[URL 17] « Structure de projet », sur le site Tutorial Android, consulté le 24 Avril 2016.

<http://tutorielandroid.francoiscolin.fr/structproj.php>

[URL 18] « Jersey », sur le site officiel de Java, le projet Jersey, consulté le 24 Avril 2016, <https://jersey.java.net/>

[URL 19] « Google Guava », sur le site Developpez, consulté le 24 Avril 2016, <http://thierry-leriche-dessirier.developpez.com/tutoriels/java/guava/>

[URL 20] « Joda-Time », sur le site The Coders Break Fast, consulté le 26 Avril 2016.

<http://thecodersbreakfast.net/index.php?post/2013/01/29/Joda-Time>

[URL 21] « Material Design Support Library | Bienvenue sur tutos-android.com », sur le site Tutos Android, consulté le 26 Avril 2016.

<http://www.tutos-android.com/material-design-support-library>

[URL 22] « Android definition final », sur le site Flight Stats, consulté le 26 Avril 2016.

<http://www.flightstats.com/company/corporate/about-us/>

[URL 23] « Mission | Hotwire.com », sur le site Hotwire, consulté le 26 Avril 2016.

<https://www.hotwire.com/en/content/mission?cc=us>

[URL 24] « A propos », sur le site Foursquare, consulté le 26 Avril 2016.

<https://fr.foursquare.com/about>

[URL 25] « About », sur le site Open Weather Map, consulté le 26 Avril 2016.

<http://openweathermap.org/about>

[URL 26] « About – Open Exchange Rates », sur le site Open Weather Map, consulté le 26 Avril 2016.

<https://openexchangerates.org/about>

[URL 27] « Amadeus Travel Innovation », sur le site Amadeus, consulté le 26 Avril 2016.

<https://sandbox.amadeus.com/about>

[URL 28] « ArticleS.UncleBob.PrinciplesOfOod », sur le site de Robert C. Martin, consulté le 26 Avril 2016.

<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>

Bibliographie

1. D.Guignard. J.Chable. E. Robles. Programmation Android - De la conception au déploiement avec le SDK Google Android 2. Eyrolles. 2010. 486 pages.
2. Mark Murphy. l'Art du développement android. Pearson. 2011. 600 pages.
3. Xavier Blanc. Isabelle Mounier. UML 2 pour les développeurs. Eyrolles. 2006. 202 pages.
4. F. Garin. Développer des applications mobiles pour les Google Phones. Dunod. 2011. 217 pages.
5. E. Evans. Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison Wesley. 2003. 560 pages.
6. M. Fowler. Patterns of Enterprise Application Architecture. Addison Wesley 2002. 560 pages.
7. B. D. McLaughlin. G. Pollice. D. West. Head First Object-Oriented Analysis and Design. O'Reilly Media. 2006. 634 pages.
8. Gamma al. Design Patterns : Elements of Reusable Object-Oriented Software. Addison Wesley. 1994. 395 pages.
9. Hassan Gommaa. Software Modeling and Design : UML, Use Case, Petterns, and Software Achitectures. Cambridge University Press. 2011. 578 pages.