République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de La Recherche Scientifique
Université Mouloud MAMMERI de Tizi-Ouzou
Faculté du Génie électrique et Informatique
Département Informatique

En vue de l'obtention du diplôme de Master en informatique
Option : Systèmes Informatiques

Thème :
Conception et réalistaion d'une application mobile de réservation de place dans un parking

Proposé et dirigé par :
Mme Aoudjit

Réalise par :
Melle Makri djoura
Melle Lammari katia

2015/2016

To ...

Table des matières

Liste des figures

Liste des Tables

Introduction générale

Avec la diversité des moyens de communication humaine, les technologies de l'information et des communications sont devnues une condition suffisante pour assurer une communication illimitée avec tous les habitants de la planète.

La communication homme_machine ou machine_machine peut être considérée comme étant un nouveau type de dialogue possible, se qui a permis aujourd'hui de disposer de nouveaux systèmes ou services de transports, les « Systèmes de Transport Intelligents » ou ITS (Intelligent Transport Systems) selon une appellation mondiale que tous les pays ou presque ont adoptée ou vont adopter.

La mauvaise gestion des espaces de stationnement et les renseignements insuffisants sur la disponibilité des espaces occasionnent des bouchons de circulation, font perdre du temps précieux aux automobilistes, entraînent un gaspillage de carburant. Les systèmes de stationnement intelligents, qui offrent des renseignements en temps réel sur la disponibilité des espaces de stationnement, facilitent grandement la recherche d'espaces de stationnement. C'est dans ce cadre que s'inscrit notre projet de fin d'étude. En effet, on cherche à mettre sur pied un environnement apte pour développer et déployer une application destinée à des téléphones portables.

Ce rapport explicite les différents stades théoriques et pratiques de l'accomplissement de notre projet et il se compose de quatre chapitres articulés comme suit :
Le premier chapitre : comporte essentiellement sur le système d'exploitation des appareils mobiles et embarquées Android.
Le deuxième chapitre : une présentation sur les systèmes de transport intelligents.
Le toisième chapitre : est consacré à l'analyse et la conception du système, et ce, par le biais de la présentation des diagrammes de cas d'utilisation, des diagrammes de classes et des diagrammes de séquences ainsi que des diagrammes d'activités.
Le quatrième et dernier chapitre expose la réalisation et la mise en œuvre de notre projet.

Nous finirons ce rapport par une conclusion générale récapitulative des différentes phases de notre travail, la problématique et signalant les cotés bénéfiques du projet, ensuite énonçant les perspectives du travail élaboré.

Chapitre I

Généralités sur android

I.1 Introduction

Ce présent chapitre sera consacré à la présentation générale de la plateforme Android, à ses différentes caractéristiques et à la présentation du kit de développement logiciel Android «SDK».

I.2 Généralités sur Android

I.2.1 Introduction

Le marché de la téléphonie mobile connaît actuellement une véritable révolution. Google, ayant réalisé le potentiel de ce marché, a décidé de s'y introduire en rachetant une startup travaillant sur un système d'exploitation ouvert pour terminal mobile : Android.

Ces dernières années, l'utilisation des applications « intelligentes » sur les Smartphones et tablettes devient de plus en plus fréquente et cela grâce aux différents systèmes d'exploitation mobiles tel que « Android », « IOS» ou bien « Windows Phone » qui ne cessent de se développer.

I.2.2 Présentation générale d'Android

Afin de s'immerger pleinement dans l'univers d'Android, une présentation de la plateforme s'impose. Nous allons donc découvrir les origines et l'historique de la plateforme Android, ses différentes versions et enfin l'architecture de cette plateforme sur laquelle s'exécutent les différentes applications.

Définition

Android est un système d'exploitation édité par Google pour appareils embarqués et/ou mobiles, comme les smartphones ou les tablettes. On le retrouve aussi dans certains GPS, ordinateurs de bord de voitures, dans des télévisions, autoradios, et même des montres. Le système Android est basé sur un fork du noyau de Linux. Ce dernier a été modifié pour être plus adapté aux terminaux mobiles ayant peu de puissance de calcul, de mémoire et de batterie. De fait, certaines librairies standards ne sont pas supportées par le système, et des améliorations ont été apportées sur la gestion de l'énergie. Les applications sont écrites en Java, et fonctionnent au sein d'une machine virtuelle Dalvik. Cette machine virtuelle a elle aussi été modifiée pour être le plus adapté possible aux appareils de faible puissance. Ainsi, beaucoup d'efforts ont été fait sur la consommation de mémoire, qui a été largement diminuée par rapport à la machine virtuelle java classique.[**1**]

Plateforme Android

La plateforme Android propose notamment : [**2**]

— Un framework permettant la réutilisation et le remplacement de composants,

— Une machine virtuelle optimisée pour les appareils mobiles,

— Un navigateur intégré basé sur le moteur open source WebKit,

— Le système de gestion de base de données SQLite pour le stockage de données,

— Un support média pour les principaux formats audio, vidéo et images,

— Un accès à la caméra, au GPS, à la boussole et aux accéléromètres,

— La téléphonie GSM, les communications Bluetooth, 3G, et WiFi,

— Un environnement de développement riche : émulateur, outils de débogage, ...

Historique d'Android

L'histoire d'Android commence en octobre 2003, où la société Android Inc est créée. Officiellement, elle développe des logiciels pour mobiles. Mais en réalité, elle se préparait à sortir un tout nouveau système d'exploitation pour smartphones. En 2005, Google rachète cette entreprise, et sort une première béta en novembre 2007, avant de lancer la version 1.0 en septembre 2008 avec le HTC Dream. À partir de ce moment là, le rythme des nouvelles sorties est très élevé : pas moins de 11 versions différentes sont sorties en 3 ans ! On remarquera au passage que chaque version d'Android porte le nom d'un dessert. [**1**]

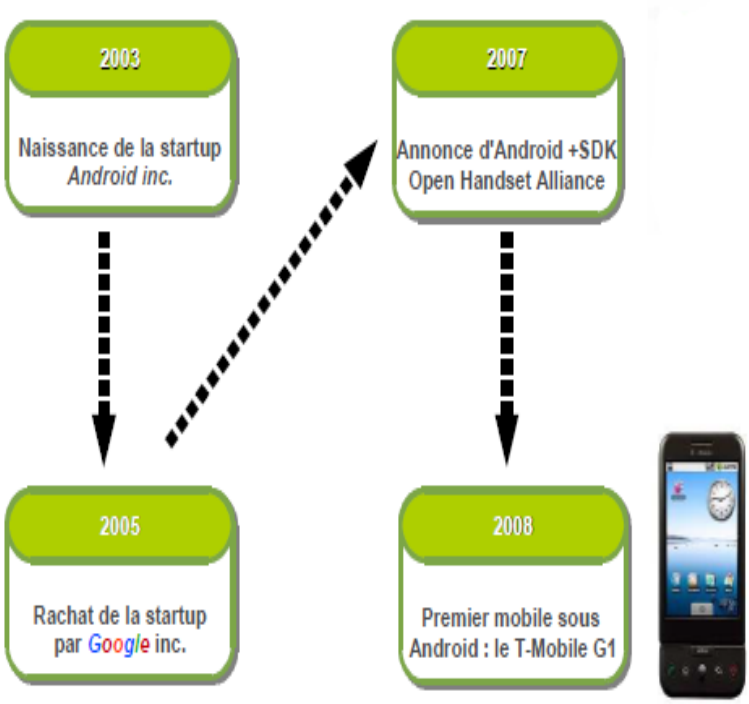
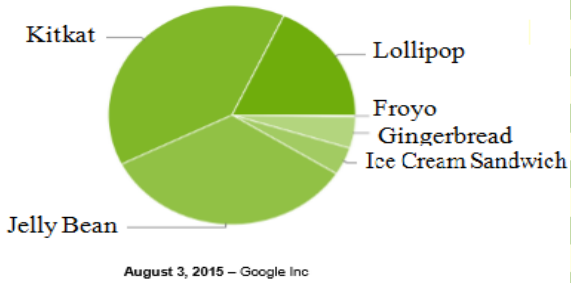


FIGURE I.1 – L'historique d'Android

Chronologie des versions

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces noms de codes suivent une logique alphabétique. **[3]**



CodeName	Platform	API Level
Cupcake	Android 1.5	3
Donut	Android 1.6	4
Eclair	Android 2.1	7
Froyo	Android 2.2	8
Gingerbread	Android 2.3	9
Honeycomb	Android 3.0	11
Ice Cream Sandwich	Android 4.0	14
Jelly Bean	Android 4.1	16
KitKat	Android 4.4	19
Lollipop	Android 5.0	20
Marshmallow	Android 6.0	23

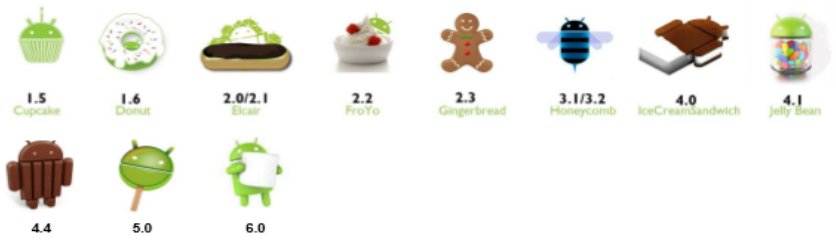


FIGURE I.2 – Evolution des versions d'Android

- La version 1.5 « Cupcake » :a été publiée en avril 2009, améliorant l'interface graphique d'Android et rendant le système plus utilisable. En effet, c'est à partir de cette version 1.5 qu'est apparu le premier clavier tactile et le presse-papier. Avant cela, il était impossible de faire un copier/coller d'un texte ;
- La mise à jour 1.6 « Donut » : sortie en septembre 2009 a permis d'apporter la fonction de recherche directement en local sur l'appareil Android ou encore les recherches Internet grâce à Google Search. Les deux versions 2.0 « Éclair » et 2.1 quant à elles ont apporté beaucoup de fonctionnalités intéressantes, comme l'amélioration du clavier tactile, l'utilisation des fonds d'écran animés, la prise en charge du Bluetooth 2.1, etc ;

- La version 2.2 « Froyo » : sortie en mai 2010 a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. De plus, Google a annoncé que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8 (Version 8) ;
- La version 2.3 « Gingerbread » :voit le jour en 2010 apportant des mises à jour pour l'interface utilisateur et des améliorations du système, des performances de réseau sur la version 4G du Nexus S, de Gmail, de l'autonomie et enfin de la caméra. Ajouté à cela la correction du bug du Bluetooth sur le Samsung Galaxy S, ainsi que d'autres fonctionnalités ;
- Android 3.0 « Honeycomb » : est apparue en février 2011 et est spécialement étudiée pour les tablettes tactiles. Cette mise à jour comprend de nombreux changements dans l'interface utilisateur ;
- En octobre 2011 apparait la version 4.0 « Ice Cream Sandwich » : Cette nouvelle version unifiée pour Smartphones et tablettes tactiles apporte de nombreux changements notamment l'ajout des Widgets depuis un nouveau menu similaire à celui des applications, l'amélioration de la messagerie visuelle, l'ajout d'un éditeur de photos et l'amélioration de l'application photo, l'ajout de dictionnaires dans le clavier virtuel, etc ;
- Android 4.1 « Jelly Bean » :en Juin 2012,apparait ayant comme principale nouveauté l'amélioration des fonctionnalités et des performances de l'interface utilisateur rendant celle-ci plus fluide ainsi que l'amélioration de l'accessibilité et la possibilité d'ajout des Widgets depuis le tiroir d'applications depuis un écran d'accueil alternatif sans avoir à rooter l'appareil, etc ;
- Android 4.4 « KitKat » :sortie en septembre 2013 apporte des mises à jour à l'exemple du clavier Google qui ajoute des émoticônes, en plus d'un changement de design ainsi que des mises à jour de l'application Téléchargements permettant une meilleure gestion avec de nouvelles options de triage et une nouvelle interface ;
- Android 5.0 « Lollipop » : sortie en novembre 2014 est une évolution majeure d'Android qui propose de nombreuses modifications et nouveautés, et qui étend sa disponibilité sur de nouveaux supports tels que la télévision, la voiture ou les montres connectées.
- Android 6 « Marshmallow » : sortie en fin mai 2015, lors de la Google I/O, que la firme de Mountain View a dévoilé les détails sur Android 6.0. La version grand public a été quant à elle déployée au courant du troisième trimestre 2015, en même temps que la sortie du nexus 5x et 6p.

Utilisation

Android est un système d'exploitation open source pour smartphones, tablettes tactiles, PDA et terminaux mobiles ; mais d'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radio-réveils, des autoradios et même des voitures. Les appareils équipés d'Android affichent un écran d'accueil qui sert de point de départ à partir duquel l'utilisateur peut accéder aux applications. Le contenu de cet écran peut être librement personnalisé par l'utilisateur (et donc aussi par les fabricants) ; il se comporte comme une pile de feuilles qui peuvent être glissées l'une sur l'autre avec le doigt.[1]

I.2.3 Caractéristiques d'Android

Architecture d'Android

Dans la figure I.3 on observe toute une pile de composants qui constituent le système d'exploitation. Le sens de lecture se fait de bas en haut, puisque le composant de plus bas niveau (le plus éloigné des utilisateurs) est le noyau Linux et celui de plus haut niveau (le plus proche des utilisateurs) est constitué par les applications.[4]

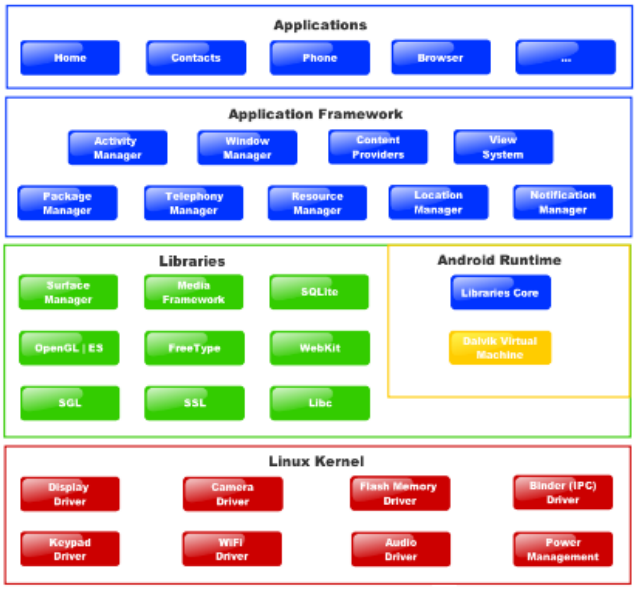


FIGURE I.3 – Architecture d'Android

- Les applications :
  Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur, ... toutes écrites en JAVA. Toutes les applications, qu'elles soient natives ou non, sont construites sur la couche applicative avec les mêmes bibliothèques d'API(Application Programming Interface). Les applications sont exécutées par le moteur.
- Le framework applicatif :
  En fournissant une plateforme de développement ouverte, Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs ont un accès complet à l'API utilisé par les applications de base. L'architecture d'application est conçue pour simplifier la réutilisation des composants ; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités. Ce même mécanisme permet aux composants de base d'être remplacés par l'utilisateur. Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris :
  - Un jeu extensible de vues qui peuvent être utilisées pour construire une application,y compris des listes, des grilles,des zones de texte,boutons,et même un navigateur web,
  - Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données,
  - Un gestionnaire de ressources,
  - Un gestionnaire de notification qui permet d'afficher des alertes personnalisées dans la barre d'état,
  - Un gestionnaire d'activités qui gère le cycle de vie des applications.
- Les bibliothèques :
  Android dispose d'un ensemble de librairies C/C++ utilisées par les différents composants du système. Elles sont offertes aux développeurs à travers le framework. Exemples :
  - Système de bibliothèque C, dérivé de la bibliothèque C standard du système (libc),
  - Médiathèques, basées sur PacketVideo de OpenCore ; les librairies permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fchiers image, y compris MP3, JPG et PNG,
  - Surface Manager qui gère l'accès au sous-système d'affchage de façon transparente,
  - LibWebCore,un moteur de navigateur web moderne qui fait tourner, à la fois le navigateur Android et une vue web intégrable,
  - SGL, le moteur graphique 2D ; et les bibliothèques 3D, implémentation basée sur OpenGL ES 1.0,
  - FreeType, bitmaps et vectoriels de rendu de polices,

— SQLite, un moteur de base de données relationnelle puissant et léger,...

- Android Runtime :
  Android inclut un ensemble de librairies de base offrant 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 la machine virtuelle Dalvik. Dalvik a été écrite pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik « .DEX », un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX. La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

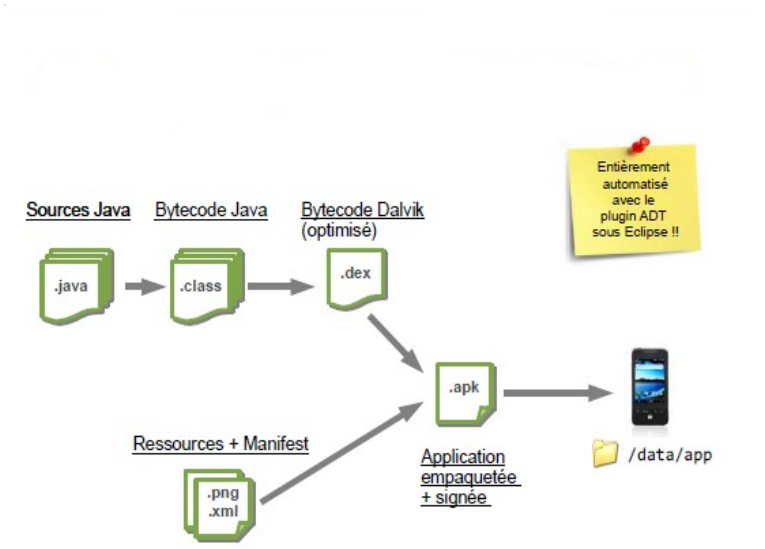


FIGURE I.4 – Compilation et déploiement d'une application Android

- Le noyau Linux :
  Android repose sur la version Linux 2.6 pour les services système de base tels que la sécurité, la gestion mémoire, la gestion de processus, la pile réseau, la gestion de l'alimentation, et les pilotes de périphériques. Le noyau agit comme une couche d'abstraction entre le matériel et le reste de la pile logicielle.

Le SDK d'Android

Un SDK, pour Software Development Kit est un ensemble d'outils logiciels permettant de faciliter le développement d'un logiciel sur une plateforme donnée (par exemple, il existe un SDK pour Android, un pour iOS, etc...). Le SDK d'Android est multi-plateforme, et se compose d'un plugin s'intégrant à Eclipse et d'un émulateur, permettant de tester son application sur plusieurs versions différentes d'Android, plusieurs tailles d'écran, etc...,et ce même si on n'a pas d'appareil physique.[4]

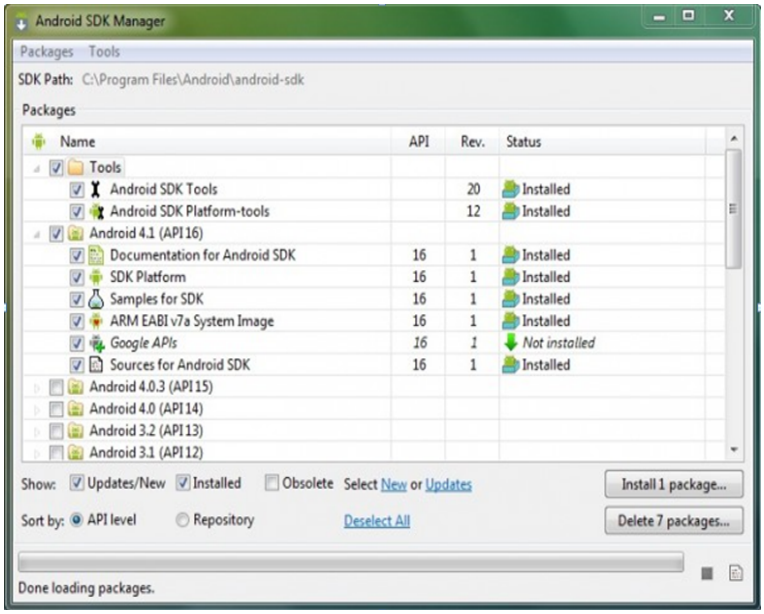


FIGURE I.5 – Android SDK

Environnement de developpement

Un projet Android doit respecter une arborescence bien précise, comme le montre la figure I.6.[**2**]

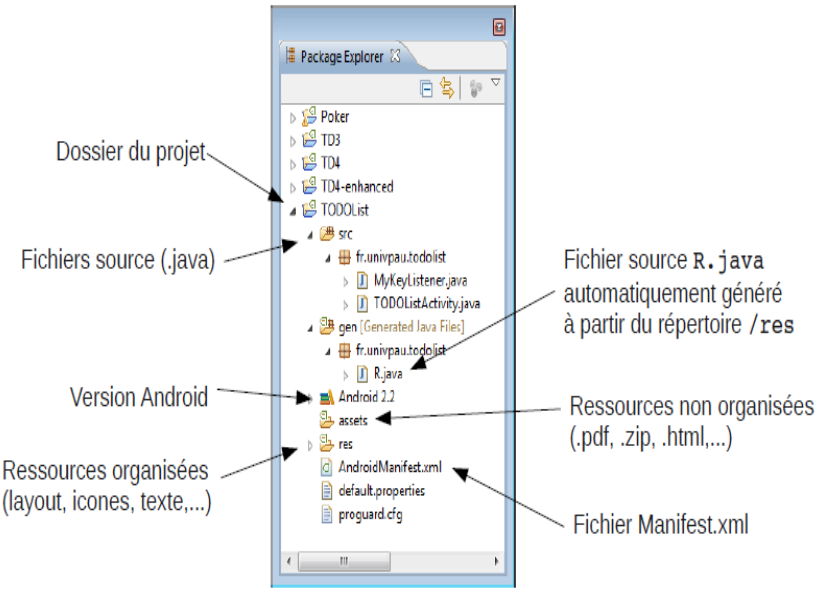


FIGURE I.6 – Arborescence d'un projet Android

— Le dossier "***src***" est classique : il contient, comme pour tout projet Java, l'ensemble des sources du projet.

— Le répertoire "***gen***" est, comme son nom l'indique, généré automatiquement par Eclipse. On y trouve un fichier "**R.java**", qui reprend l'ensemble des ressources du projet (les images, les éléments des interfaces graphiques, les valeurs contenues dans le dossier "**values**", etc.). Cela permet d'accéder, depuis le code, aux différents éléments disséminés dans les fichiers XML.

— Le dossier "***bin***" est généré lors de la compilation et contient, entre autres, le fichier à installer sur le terminal. Ce fichier porte l'extension "**apk**", et correspond, pour faire l'analogie avec le monde Java, à un "**jar**" exécutable.

— Le dossier "***res***" contient les ressources nécessaires à l'exécution de l'application. Ainsi, les différentes images sont placées dans les dossiers "**drawable**" (chaque dossier contient les mêmes images de taille différentes).

— Le répertoire "***layout***" contient les fichiers XML définissant les interfaces graphiques.

— Le dossier "***values***" peut contenir plusieurs fichiers XML déclarant certaines valeurs particulières. On peut y stocker des chaines de caractères, des dimensions, des couleurs, des tableaux, et y accéder simplement depuis le code ou l'interface graphique. Cela permet notamment d'internationaliser les applications de manière transparente dans le code source.

— Enfin, le fichier "***AndroidManifest.xml***" situé à la racine du projet, permet de configurer l'installation de l'application sur les terminaux mobiles.

Les outils de développement du SDK

Le SDK est livré avec un certain nombre d'outils couvrant différents aspects du cycle de développement d'une application Android. En effet, le kit de développement propose une boîte à outils complète pour les tâches de compilation, débogage, génération de code AIDL, signature des applications, etc. Le répertoire Tools du kit de développement contient ainsi un ensemble de programmes dont voici quelques exemples : [**3**]

Emulateur

L'émulateur est le parfait outil de test et de débogage des applications. C'est une implémentation de la machine virtuelle Dalvik, faisant de celle-ci une plateforme exécutant les applications Android comme le ferait n'importe quel téléphone Android. Étant découplé de tout matériel, c'est une excellente base de test des applications. Il fournit une connectivité réseau complète ainsi que la possibilité d'ajuster finement la vitesse de connexion et la latence au cours du débogage des applications. Il est également possible de simuler l'envoi et la réception d'appels et de SMS. Le plugin ADT(Android Development Tools) intègre l'émulateur à Eclipse, qui le lance automatiquement avec l'AVD(Android Virtuel Device) sélectionné lorsqu'un projet est exécuté ou débogué. Si le plugin n'est pas utilisé ou l'émulateur est utilisé hors d'Eclipse, il devient possible de s'y connecter via Telnet et de le contrôler depuis une console. Avant d'exécuter l'émulateur, un appareil virtuel doit être créé, celui-ci sera lancé par l'émulateur, qui y exécutera une instance Dalvik.[5]

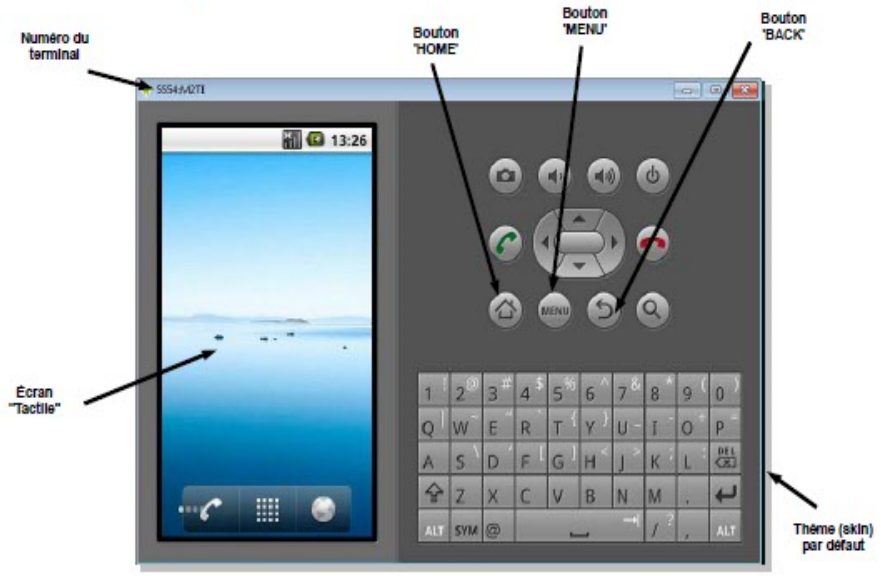


FIGURE I.7 – Emulateur Android

ADB(Android Debug Bridge)

L'ADB est à la fois un client et un service jouant le rôle d'un conduit de communications entre la machine de développement et l'émulateur ou le dispositif Android. ADB permet d'installer des applications, de copier ou lire des fichiers et d'exécuter des commandes Shell sur le dispositif cible. [**5**] Il est composé de trois parties :

- Un démon exécuté par l'émulateur ;
- Un service exécuté par la machine de développement ;
- Des applications clientes (comme le DDMS) qui communiquent avec le démon via le service.

DDMS(Dalvik Debug Monitor Service)

Contrairement à l'émulateur qui permet de voir à quoi ressemble l'application et la façon dont elle se comporte et interagit, le DDMS permet de voir ce qui se passe en profondeur. En effet, c'est un puissant outil de débogage avec lequel il est possible d'interroger les processus actifs, d'examiner la pile et le tas, de surveiller et de mettre en pause les threads actifs et enfin d'explorer le système de fichier de n'importe quel matériel Android connecté. La perspective DDMS sous Eclipse fournit également un accès simplifié aux captures d'écran de l'émulateur et aux journaux générés par LogCat. Le DDMS est complètement intégré à Eclipse via le plugin ADT et disponible dans une perspective. [**5**]
Les composants du DDMS sont :

- Debug Process Icon : il montre l'état de connexion du débogueur.
- Update Heap : en cliquant sur ce tas d'informations, il permet de vider le processus de sorte que la mémoire ne soit pas libérée manuellement.
- Dump HPROF : génère un vidage de tas, utile pour le suivi des fuites de mémoire dans l'application.
- Cause : appelle le garbage collector de collecter des données heap.
- Update thread : affiche les informations sur le fil conducteur pour le processus sélectionné.

- Start Method Profiling : assure le suivi des mesures liées à une méthode. Il collecte des informations comme la quantité de temps nécessaire pour exécuter une méthode, le nombre d'appels.
- Stop Process : arrête le processus en cours de sélection.
- Screen Capture : prend une capture d'écran de ce qui est affiché sur l'écran.
- Reset ADB : comme son nom l'indique, il réinitialise l'ADB.

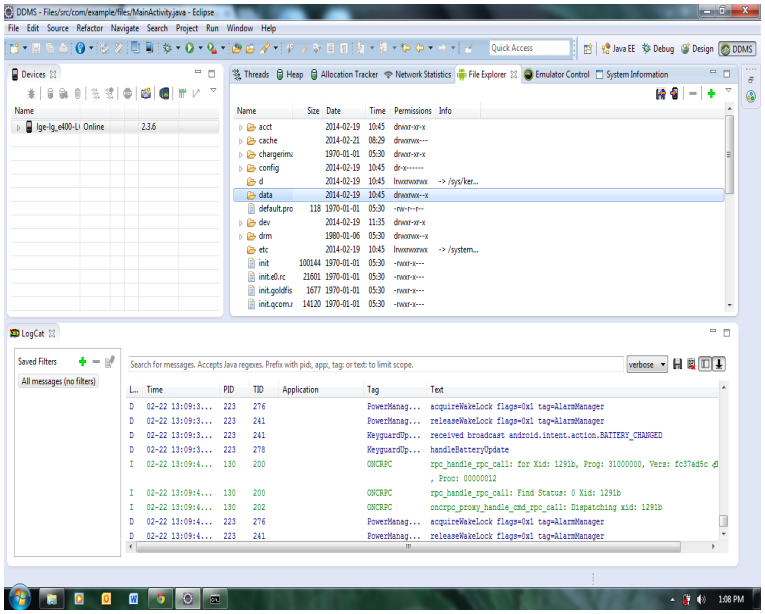


FIGURE I.8 – Le DDMS(Dalvik Debug Monitor Service)

Android Market

L'Android Market renommé "Google Play Store" en mars 2012 est le système de publication d'applications officiel d'Android. En effet, c'est une application préinstallée sur chaque téléphone fonctionnant sous Android, permettant de télécharger des « applications » développés par des sociétés ou des développeurs indépendants. Néanmoins, contrairement à l'App Store d'Apple, la distribution d'applications au public n'est pas un monopole de Google et techniquement, rien n'empêche d'installer des applications depuis une autre source. D'ailleurs, des alternatives à Android Market, comme par exemple SlideMe émergent peu à peu. Pour proposer son application sur Android Market, il faudra détenir un compte Google et s'acquitter des frais d'enregistrement. Bien sûr, l'application devra être parfaitement packagée et signée, par ailleurs Android Market requiert que la date de validité du certificat soit postérieure au 22 octobre 2033, qu'une icône et un intitulé soient précisés dans le manifeste et que les champs de version soient renseignés. [**6**]

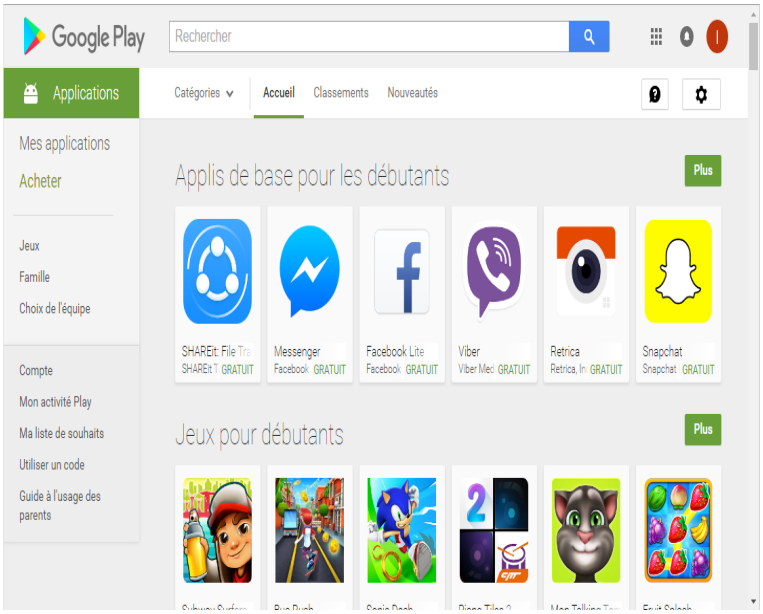


FIGURE I.9 – La page principale de Google Play Store sur Internet

I.3 Conclusion

Au cours de ce chapitre, nous avons présenté la plateforme Android, ses différentes caractéristiques ainsi que le kit de développement Android SDK.

Les différents concepts traités dans ce chapitre nous aiderons à comprendre au mieux notre environnement de développement et les notions fondamentales pour mener à bien notre application.

Une présentation des systèmes de transport intelligent fera l'objet du chapitre suivant.

Chapitre II

Système de Transport Intelligent

II.1 Introduction

Le comportement des usagers des transports est en pleine évolution, l'objet du déplacement, les circonstances de celuici(intempéries, horaires,. . .) et l'offre de transport sont autant de paramètres qui interviennent aujourd'hui dans le choix du mode de déplacement, et le recours successif à différents modes de transport est de plus en plus fréquent.

Pour choisir son mode de transport, l'usager ne s'intéresse plus seulement aux moyens mis en oeuvre mais cherche le meilleur résultat en termes de temps de parcours, de simplicité d'utilisation, de sécurité, de confort et de coût.

Les Nouvelles Technologies d'Information et de Communication, associées à l'ingénierie des transports, nous permettent aujourd'hui de disposer de nouveaux systèmes ou services de transports, les « Systèmes de Transport Intelligents » ou ITS (Intelligent Transport Systems) selon une appellation mondiale que tous les pays ou presque ont adoptée ou vont adopter.

II.2 Présenation des Systèmes de Transport Intelligent(STI)

Les Systèmes de Transport Intelligents "STI" (Intelligent Transportation Systems "ITS") constituent une initiative mondiale visant à intégrer les technologies de l'information et de la communication aux infrastructures de transport ainsi qu'aux véhicules.
Les systèmes de transport intelligents doivent apporter leur concours au déploiement des politiques publiques de transport : [**7**] [**8**]

- **Optimisation de l'utilisation des réseaux** par la multimodalité pour les transports de personnes et de marchandises.
- **Promotion des transports publics** par l'information des usagers.

FIGURE II.1 – Valideur de carte de transport

- **Sécurité routière** par l'automatisation des contrôles et la télématique routière.
- **Réduction des consommations d'énergie et de la pollution** par les systèmes de gestion de trafic et les véhicules communicants.

II.3 Les grands objectifs des politiques STI

1. Donner aux exploitants des réseaux d'infrastructures et de services de transport des outils leur permettant de proposer à leurs clients des solutions de bout en bout optimisant l'usage des capacités existantes des différents modes tout en respectant les règles de la concurrence.
2. Améliorer la sécurité, en particulier la sécurité routière, par la connaissance et la compréhension des règles à appliquer et faire évoluer les comportements en automatisant les contrôles.
3. Améliorer la qualité de la vie dans les villes et les services de mobilité qu'elles offrent à leurs habitants et à leurs visiteurs, en facilitant l'usage des transports publics et des modes doux, notamment par l'information multimodale, la billettique et des organisations de logistique urbaine performantes.
4. Réduire les inégalités en agissant sur l'accessibilité, à la fois pour les personnes à mobilité réduite et pour les territoires les moins bien desservis.
5. Maîtriser les consommations d'énergie et les émissions de polluants en réduisant la congestion et en produisant des indicateurs adaptés aux différentes catégories de décideurs (y compris les usagers finaux) sur les effets de leurs actions par des exploitations assez réactives des données disponibles pour faire évoluer les comportements. [**7**]

FIGURE II.2 – Surveillance radar

II.4 Les domaines d'application des STI

1. **L'information routière :**
   Elle existe en France de longue date et se développe continûment (médias grand public, radios autoroutières, panneaux à message variables, information temps de parcours, services divers créés par des opérateurs publics ou privés . Cependant des services multimodaux commencent à se développer, notamment sous forme de sites internet.[**7**]

FIGURE II.3 – Site Sytadin : Etat du trafic en Ile de France

2. **Transports de marchandises :**
   Les STI se sont rapidement développés dans le domaine du transport de marchandises, d'autant plus aisément que les flottes de véhicules, trains ou autres engins sont complètement identifiées et peuvent être facilement équipées. Par exemple aujourd'hui, toutes les flottes sont équipées de systèmes de navigation embarqués. Outre les véhicules, le fret (conteneurs, palettes, colis...) peut aussi voyager muni de systèmes de localisation pour suivre l'avancement et éviter les pertes. L'apport des STI intervient dans plusieurs champs : [**7**] [**8**]

   a) L'intégration de la chaîne de transport et de la logistique (suivi des véhicules, des marchandises).

   b) La prise en compte de la multimodalité et de l'intermodalité.

   c) L'application de la réglementation et la protection de l'environnement (notamment dans le cas du transport de matières dangereuses pour lequel un suivi particulier est obligatoire).

3. **Gestion de données partagées :**
   Un "historique" de données est indispensable. Pour connaître les caractéristiques des réseaux, les caractéristiques de la demande de trafic et des problèmes récurrents, il est indispensable de disposer de bases de données. Les nouvelles technologies permettent de stocker des volumes importants d'information. L'enjeu réside aujourd'hui dans la gestion de ces données. Par exemple, la création de base de données communes regroupant les données de différents exploitants est intéressante pour analyser et comprendre les situations de trafic.[**8**]
4. **Utilisation des STI pour la protection de l'environnement :**
   L'obligation d'affichage des émissions de CO2 liées aux prestations de transport fait partie des décisions prises. Les textes d'applications sont en cours d'adoption et la normalisation des méthodologies avance au niveau européen. On peut penser que ces informations seront issus d'un traitement a posteriori des données issues des systèmes de comptabilité et de données complémentaires produites par les systèmes d'aide à l'exploitation des entreprises et de gestion des infrastructures.[**7**]

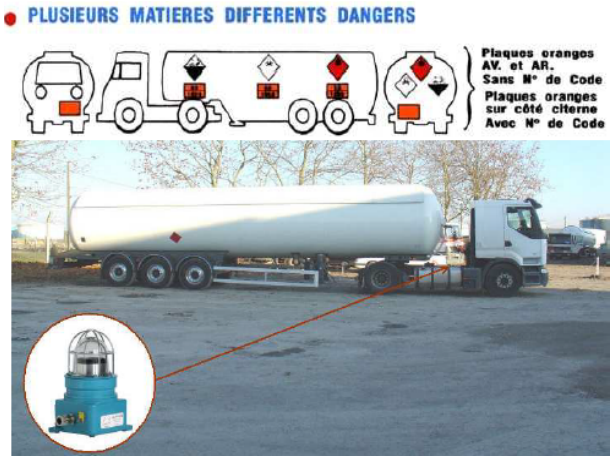


FIGURE II.4 – Transport des matières dangereuses

5. **STI et aides à la conduite :**
   Les STI apportent au conducteur plus de confort et de sécurité.
   Les systèmes de navigation permettent d'optimiser le déplacement et peuvent être complétés par des indications sur la géométrie de la route et les limitations de vitesse.
   Les systèmes d'écoconduite peuvent réduire les consommations de carburant de façon significative. Les limiteurs de vitesse pourraient s'adapter à la réglementation applicable au lieu où se trouve le véhicule, avec des effets positifs sur la sécurité et la fluidité du trafic. La recherche est active dans ce domaine, notamment sur les systèmes coopératifs qui peuvent impliquer l'installation d'équipements sur les infrastructures.[**8**]

FIGURE II.5 – Chronotachygraphe :appareil numérique de contrôle des temps de conduite et de repos

6. **Gestion des urgences :**
   La gestion d'urgence, en particulier en cas d'accident de la route, utilise au maximum des systèmes automatisés de recueil de l'information et des transmissions performantes. Les principaux enjeux sont la rapidité d'intervention, l'évitement d'accidents en chaîne et le rétablissement de la circulation.
   Quelques exemples de STI aidant en cas de situation d'urgence sont :
   - La Détection Automatique d'Incidents (DAI) par les capteurs routiers installés sur l'infrastructure et qui préviennent l'exploitant du réseau.
   - Les services d'assistance à l'automobiliste (ex : appel automatique des secours en cas de collision, envoi automatique de la localisation précise du lieu d'accident) qui ne sont pas encore bien démocratisés.[**7**]

7. **Contrôle du respect de la réglementation :**
   Ces systèmes font appel à des technologies telles les flash infrarouges utilisés de nuit pour lire les plaques minéralogiques sans éblouir les conducteurs, ou des capteurs précis pour déterminer la vitesse des véhicules.
   Le contrôle du respect de la réglementation dans les transports en commun (par caméra ou autres équipements) répond aussi à des objectifs économiques de lutte contre la fraude et le vandalisme.
   Toutefois, ces systèmes sont aussi utilisés à des fins de prévention et de répression des crimes, et photographient notamment les passagers des véhicules.[**8**]

II.5 Les technologies des STI

Les technologies utilisées dans les systèmes de transport intelligents varient, allant de systèmes de gestion basiques comme les systèmes de gestion des carrefours à feux, les systèmes de gestion des conteneurs, les panneaux à messages variables, les radars automatiques ou la vidéosurveillance aux applications plus avancées qui intègrent des données en temps-réel avec retours d'informations de nombreuses sources, comme les informations météorologiques, les systèmes de dégivrage des ponts, les systèmes de navigation embarqués informant des temps de parcours en temps réel etc. De plus, les techniques prédictives sont développées pour permettre une modélisation avancée et une comparaison avec une base regroupant des données historiques de référence.[**7**]

II.5.1 Communication sans fil

Diverses technologies de communication sans fil sont proposées pour les systèmes de transport intelligent :

— Des communications à courte portée (moins de 350 mètres) comme le Wi-Fi ;

— Des communications à plus longue portée comme le WiMAX, le GSM ou la technologie 3G.

II.5.2 Technologies de localisation

— GPS :

Le principe du positionnement par GPS est très proche du principe de triangulation(technique permettant de déterminer la position d'un point en mesurant les angles entre ce point et d'autres points de référence dont la position est connue). La distance entre l'utilisateur du GPS et un certain nombre de satellites de positions connues est mesurée pour permettre une localisation de l'utilisateur à une dizaine de mètres près. La vitesse de déplacement est aussi disponible. Certains STI s'appuient donc sur cette technologie qui permet la navigation en temps réel. Ce système a été installé par les États-Unis.

— Téléphonie mobile :

En admettant que les voitures contiennent au moins un ou plusieurs téléphones mobiles ou cellulaires, les téléphones transmettent leur position de façon régulière au réseau, même s'il n'y a pas de communication vocale établie. Ils peuvent alors être utilisés dans les voitures comme des sondes anonymes du trafic. Quand la voiture bouge, le signal du téléphone mobile bouge également. Il est alors possible de mesurer et d'analyser par triangulation les données fournies par le réseau cellulaire, de manière anonyme, puis de convertir ces données en une information précise sur la circulation automobile. Plus il y a de congestion, plus il y a des voitures, des téléphones et donc de sondes.[**7**]

II.6 Exemples d'applications des Systèmes de Transport Intelligent

1. **Surveillance et gestion des feux de circulation :**
   Les capteurs D-tec de Leddar sont utilisés dans les SAGC (Systèmes Avancés de Gestion de la Circulation) pour détecter avec précision les lignes d'arrêt et la présence de circulation. Cette solution optique novatrice est la seule du secteur des transports à compiler des données des milliers de fois par secondes pour offrir une détection fiable et précise des véhicules de toute taille, y compris les bicyclettes et les motos, à proximité des feux de circulation et des intersections.[**8**]

FIGURE II.6 – Le capteurs D-tec de Leddar

Les organismes de contrôle de la circulation routière européens font appel à la technologie Leddar pour leur système non intrusif et indétectable de détection de la vitesse moyenne des véhicules. L'installation routière, qui peut mesurer des vitesses pouvant aller jusqu'à 250 km/h, offre des mesures ultra précises pour tous les types de véhicules et dans toutes les conditions météorologiques. Les modules Leddar s'intègrent facilement aux plateformes existantes à des fins d'application de la loi.

2. **Détection d'occupation des espaces de stationnement :**
   À la base de ces applications intelligentes émergentes, on retrouve des réseaux de capteurs qui détectent l'occupation des espaces de stationnement. Les capteurs Leddar ont la capacité de surveiller un ou plusieurs espaces de stationnement intérieurs ou extérieurs. La puissante capacité de détection de Leddar tient la route dans les conditions météorologiques les plus difficiles, et ses algorithmes évolués éliminent les interférences de l'environnement et les détections faussement positives. La petite taille des capteurs leur permet d'être intégrés aux infrastructures existantes, y compris aux parcomètres et aux lampadaires.. [**7**]

FIGURE II.7 – Détecteur d'occupation d'espace de stationnement sur voirie

3. **Radar anticollision pour véhicules routiers :**
   Capable de détecter la nature de l'obstacle (moto, vélo, piéton, bordure d'autoroute, ...) pour une meilleure prévention des accidents. Dans le cadre du projet ANR « PRIMACARE » labellisé ITrans, le système radar conçu signale au conducteur les obstacles perçus sur 360°, au moyen d'alarmes spatialisées (son 3D). Le système englobant les réalisations de tous les partenaires du projet a fait l'objet d'un démonstrateur en Mars 2012 montrant les performances de ce concept. La figure suivante présente le véhicule qui a été équipé pour tester les algorithmes proposés.[**8**]

FIGURE II.8 – Véhicule équipé dans le cadre du projet PRIMACARE

II.7 Conclusion

Au cours de ce chapitre, nous avons présenté la notion des systèmes de transport intelligent.

Pour que ces systèmes apportent des réponses efficaces aux multiples problématiques de transport, il conviendra de veiller à ce que les intérêts collectifs ne soient pas oubliés dans le développement des dynamiques commerciales. Il peut s'agir de questions de service public, qui apportent un bénéfice à la collectivité, ou bien de services purement commerciaux. Il sera donc primordial d'attacher une attention particulière aux outils à mettre en place pour garantir une bonne régulation de ces services.

Les étapes et détails de la conception de notre application feront l'objet du prochain chapitre.

Chapitre III

Analyse et Conception

III.1 Introduction

La réalisation d'une application mobile doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'une application afin de rendre ce développement plus fidèle aux besoins du client.

III.2 Objectif de l'application

Les applications mobiles sont actuellement en pleine croissance grâce aux succès des smartphones et plus particulièrement ceux qui utilisent le système d'exploitation Android (Android OS). Cette tendance offre aux entreprises plusieurs opportunités pour créer des nouveaux services et élargir leurs supports de communication.

Le stationnement est considéré comme un levier essentiel pour une politique des déplacements, et la mauvaise gestion des espaces de stationnement et les renseignements insuffisants sur la disponibilité des espaces occasionnent des bouchons de circulation, font perdre du temps précieux aux automobilistes.

Pour remédier à ce genre de problème, les automobilistes peuvent faire appel à cette nouvelle technologie à savoir : les applications mobiles qui permettent de gagner du temps et accéder à un parking choisi par le client en quelques clics seulement.

Ainsi, l'objectif de ce travail est de réaliser ce besoin en développant une application mobile sous Android permettant de localiser le parking le plus proche à la position actuelle de l'utilisateur, de consulter la disponibilité de places en temps réel, d'effectuer des réservations de places pour un accès immédiat ou futur et de communiquer avec l'agent.

III.3 Processus de developpement

III.3.1 Analyse

La phase d'analyse débute par la mise en évidence des différents acteurs qui interviennent dans le système, et leurs besoins. Ensuite, modéliser les objectifs à atteindre dans la phase de conception et ce en s'appuyant sur la phase d'analyse.

UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue, il permet d'unifier les notations et les concepts orienté objet, il unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la spécification jusqu'au codage.

FIGURE III.1 – Le logo d'UML

La présence de la modélisation dans le langage UML met en œuvre le terme méta modèle qui permet de représenter l'ensemble des éléments et leurs liens respectifs. Ces éléments sont représentés par un ensemble des diagrammes classés suivant deux vues : [**9**]
Les vues statiques, c'est à dire représentant le système physiquement :

- Diagrammes d'objets.
- Diagramme de classes.
- Diagrammes de cas d'utilisation.
- Diagrammes de composants.
- Diagrammes de déploiement.

Les vues dynamiques, montrant le fonctionnement du système :

- Diagrammes de séquence.
- Diagrammes de collaboration.
- Diagrammes des états transitions.
- Diagrammes d'activités.

Quelques définitions de base

- **Un acteur :** représente un rôle joué par une personne externe (utilisateur humain, matériel ou autre système), qui interagissent directement avec le système étudié.
- **Cas d'utilisation :** un cas d'utilisation « use case » modélise une interaction entre le système informatique à développer et un utilisateur ou acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur. [**10**]

Relations entre cas d'utilisations

- **Relation « Include » :** une relation d'inclusion d'un cas d'utilisation A à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B. Le cas d'utilisation A ne peut être utilisé seul.
- **Relation « Extend » :** une relation d'extension d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.
- **Relation « Use » :** lorsqu'une ou plusieurs tâches sont utilisées régulièrement on peut les factoriser dans un même use case et faire de telle sorte que d'autres uses cases l'utilisent en pointant par une flèche. [**9**]

Identification des acteurs

Les acteurs de notre application sont :

- **Administrateur (agent du parking) :** la personne chargée du maintien et de la mise à jour de la BDD (gestion du parking) ;
- **Visiteur :** toute personne visitant l'application ;
- **Client (l'utilisateur de l'application) :** toute personne pouvant accéder à son compte personnel et effectuer une recherche rapide du parking le plus proche, des places disponibles. Un client peut aussi faire une réservation ou encore accéder aux réservations personnelles pour effectuer d'éventuels changements (modifier / annuler une réservation).

Identification et représentation des cas d'utilisations

1. **Spécification des tâches**

Acteur	**Tâche**
Visiteur	**T1** : Se connecter a l'application **T2** : Consulter les parkings **T3** : S'inscrire à l'application client **T4** : Contacter l'agent
Client	**T5** : Idem que le visiteur sauf l'inscription **T6** : Accéder à l'espace client **T7** : Consulter les parkings **T8** : Consulter ses réservations **T9** : Réservation d'une place dans un parking **T10** : Annuler une réservation **T11** : Modifier une réservation **T12** : Envoyer un email à l'agent **T13** : Contacter l'agent par téléphone **T14** : Changer son mot de passe **T15** : Récupérer son mot de passe oublié **T16** : Recherche parking
Agent	**T17** : S'authentifier **T18** : Accéder à l'espace agent **T19** : Consulter les réservations courantes et l'historique des réservations **T20** : Libérer place **T21** : Consulter la liste des clients **T22** : Contacter client par téléphone **T23** : Envoyer un email au client **T24** : Modifier le nombre de place d'un parking (extension) **T25** : Changer son mot de passe **T26** : Récupérer son mot de passe oublié

TABLE III.1 – Tableau de spécification des tâches

2. **Spécification de cas d'utilisation**

Dans ce qui suit, nous allons présenter quelques diagrammes correspondant à quelques cas d'utilisation de notre application :

a) **Cas utilisation « Effectuer réservation »**

Résumé : cette étape permet au client de chercher les parkings qui se trouvent dans la wilaya recherchée et choisir le plus proche à sa position actuelle, ensuite il pourra réserver une place à distance via son téléphone mobile.

Acteur : client.

Pré condition : l'application est lancée.

Description :

1. Demande de saisie date&heure d'entrée
2. Vérification de disponibilité de places selon la date&heure d'entrée
3. Remplissage du champ matricule
4. Validation de la réservation

Enchainement d'erreurs : pas de parking proche à la position actuelle du client.

Pas de place libre pour les horaires données.

FIGURE III.2 – Cas utilisation « Effectuer réservation »

b) **Cas utilisation « Annuler réservation »**

Résumé **:** cette étape permet au client ou à l'agent d'effectuer l'annulation de réservations.

Acteur : client.

Pré condition : authentification.

Description :

1. Le client accède à son espace après son authentification
2. Le client clique sur le bouton « Mes réservations »
3. Le client sélectionne la réservation à annuler et clique sur « annuler »
4. Le système affiche une boite de dialogue pour confirmer l'annulation
5. Le client clique sur le bouton « Oui » pour confirmer l'annulation ou le bouton « Non » pour annuler son choix
6. Le système supprime la réservation de la BDD si le client appuie sur le bouton « Oui » et affiche la nouvelle liste.

FIGURE III.3 – Cas utilisation « Annuler réservation »

c) **Cas utilisation « Modifier réservation »**

Résumé : cette étape permet au client de modifier une (des) réservation(s) courante(s).

Acteur : client.

Pré condition : authentification.

Description :

1. Le client accède à son espace après son authentification
2. Le client clique sur le bouton « Mes réservations »
3. Le client sélectionne la réservation à modifier
4. Le client choisit une autre date et heure d'entrée et clique sur le bouton « modifier »
5. Le système affiche une boite de dialogue pour confirmer la modification
6. Le client clique sur le bouton « Oui » pour confirmer la modification ou le bouton « Non » pour annuler son choix
7. Le système supprime la réservation de la BDD si le client appuie sur le bouton « Oui » et affiche la nouvelle liste.

FIGURE III.4 – Cas utilisation « Modifier réservation »

d) **Cas utilisation « Ajouter place »**

Résumé : cette étape permet à l'agent d'ajouter une (des) place(s).

Acteur : agent.

Pré condition : authentification.

Description :

1. L'agent accède à son espace après son authentification
2. L'agent clique sur le bouton « Parking »
3. L'agent clique sur le parking en question puis édite le nombre de places
4. L'agent clique sur enregistrer pour confirmer l'ajout
5. Le système met à jour le nombre de places de la BDD et affiche le nouveau nombre.

FIGURE III.5 – Cas utilisation « Ajouter place »

e) **Cas d'utilisatisation pour l'administrateur (Agent du parking)**

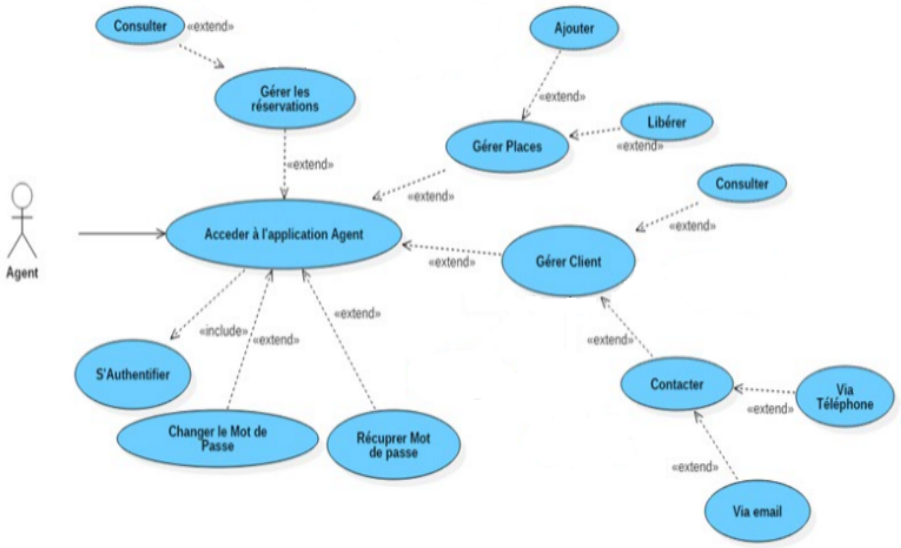


FIGURE III.6 – Cas d'utilisatisation pour l'administrateur (Agent du parking)

f) **Cas d'utilisatisation pour le client**

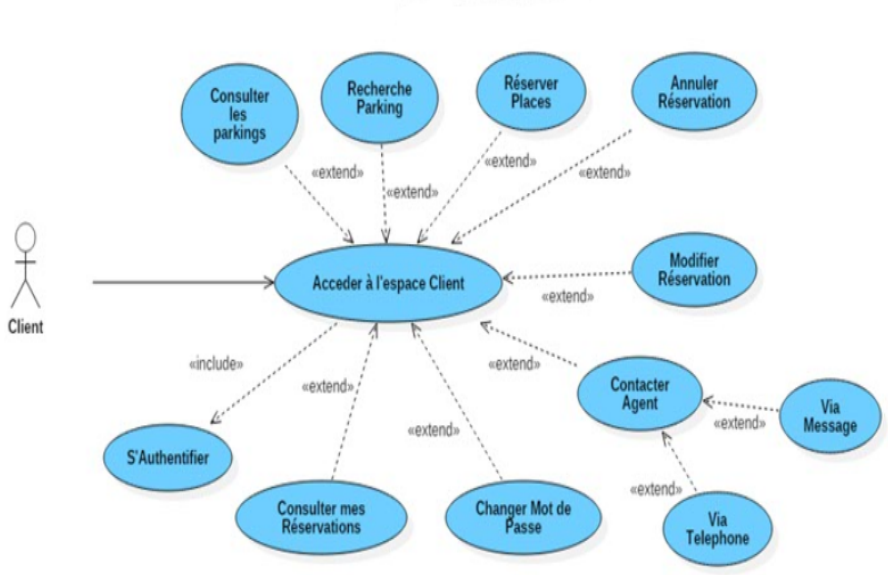


FIGURE III.7 – Cas d'utilisatisation pour le client

g) **Cas d'utilisatisation pour le visiteur**

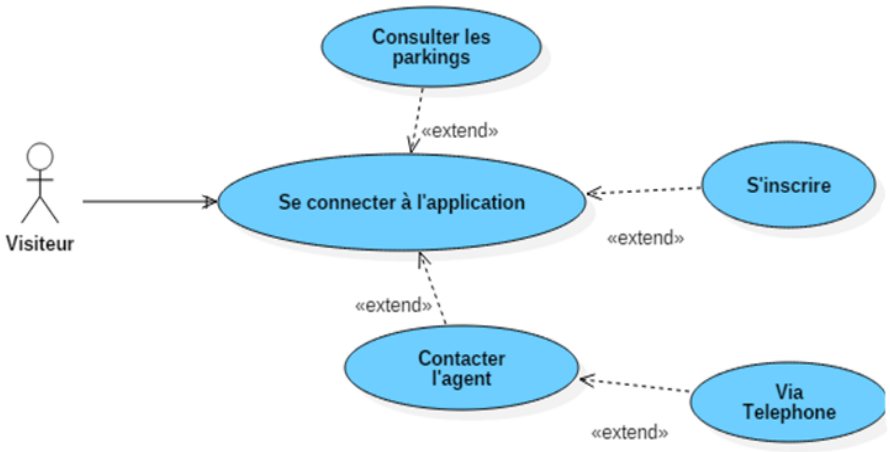


FIGURE III.8 – Cas d'utilisatisation pour le visiteur

III.3.2 Conception

La conception est en grande partie un processus d'affinement du modèle d'analyse, cependant elle consiste à apporter des solutions techniques aux descriptions définies lors de l'analyse.

Pour développer notre projet nous sommes appelés à concevoir l'application et la base de données avec laquelle interagie le système. Dans la conception de notre application, nous allons construire les diagrammes de séquence suivis des diagrammes de classes et enfin les diagrammes d'activités.

1. **Diagramme de séquence :**

   Le diagramme de séquence est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. On montre ces interactions dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Le but étant de décrire comment se déroule les actions entre les acteurs ou objets. [**10**]

   — **Quelques diagrammes de séquences**

a) Diagramme de séquence « Effectuer réservation »

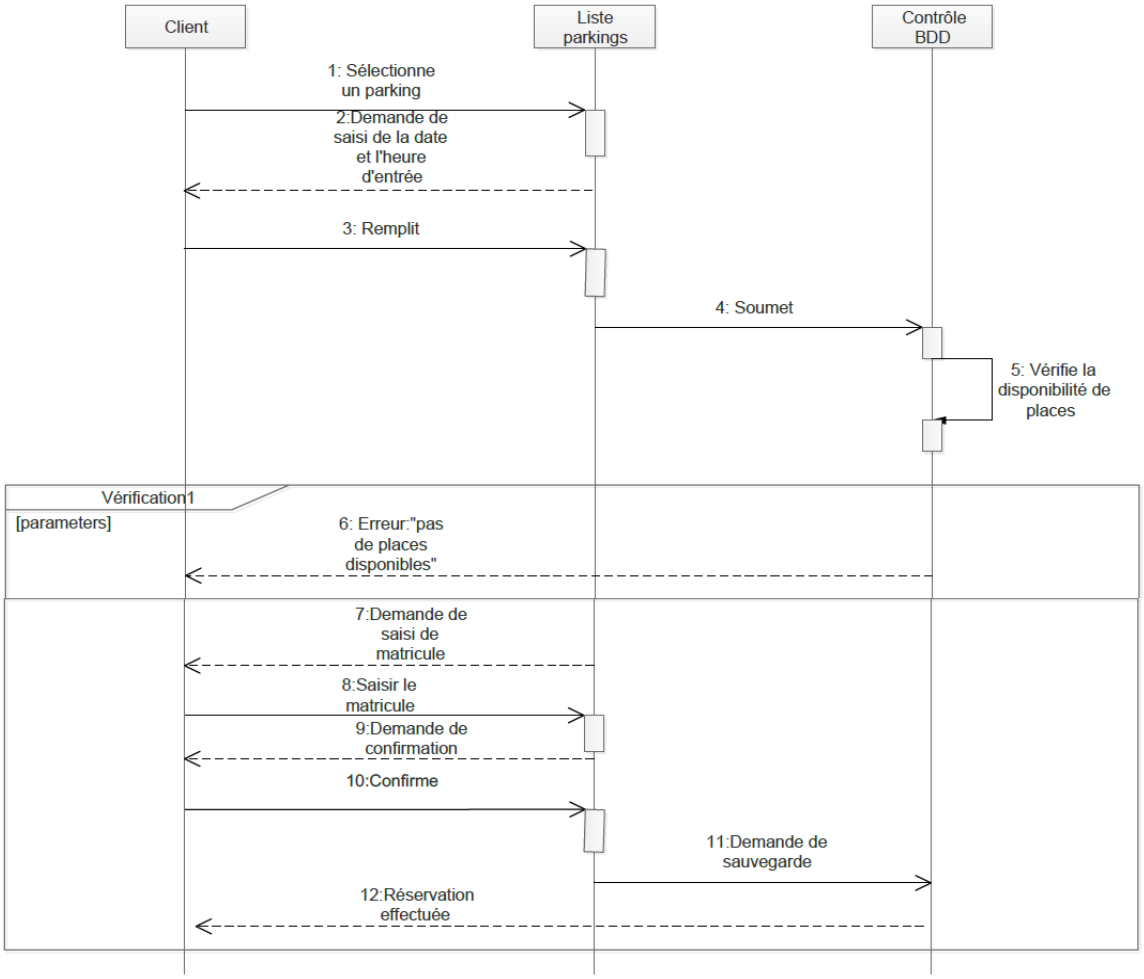


FIGURE III.9 – Diagramme de séquence « Effectuer réservation »

b) Diagramme de séquence « Annuler réservation »

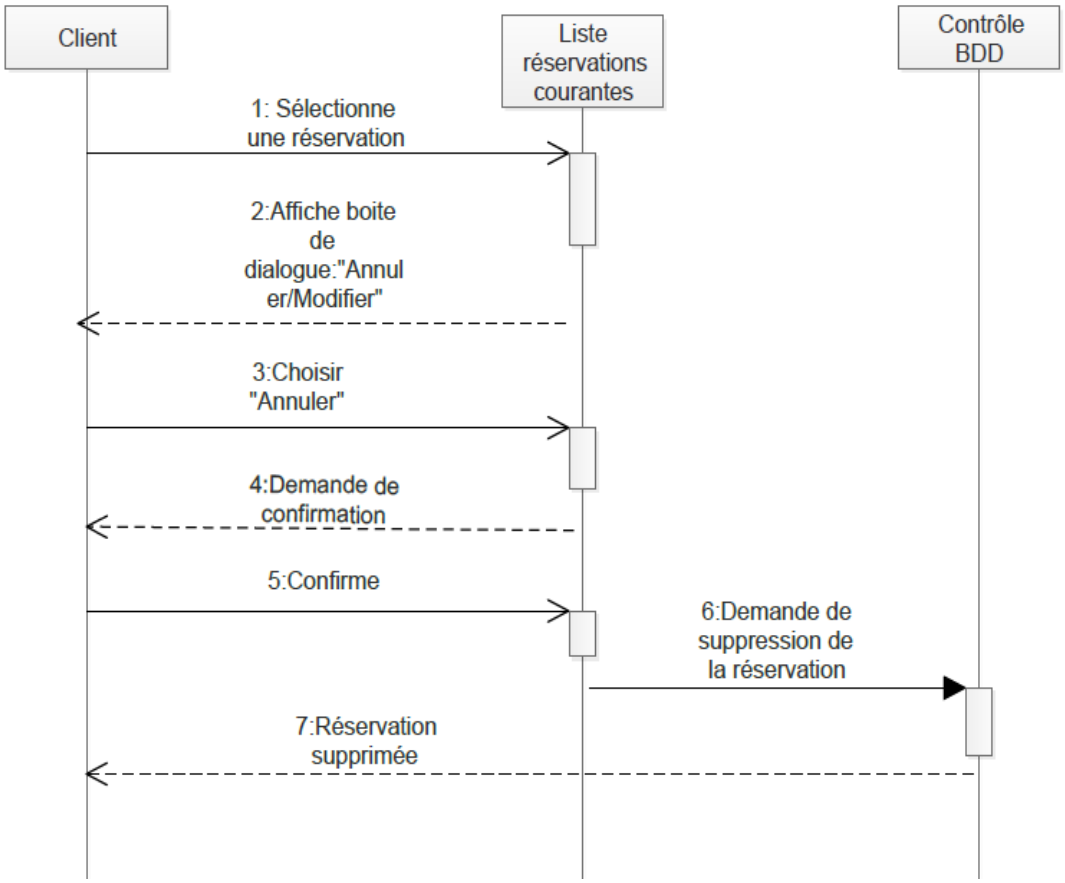


FIGURE III.10 – Diagramme de séquence « Annuler réservation »

c) Diagramme de séquence « Modifier réservation »

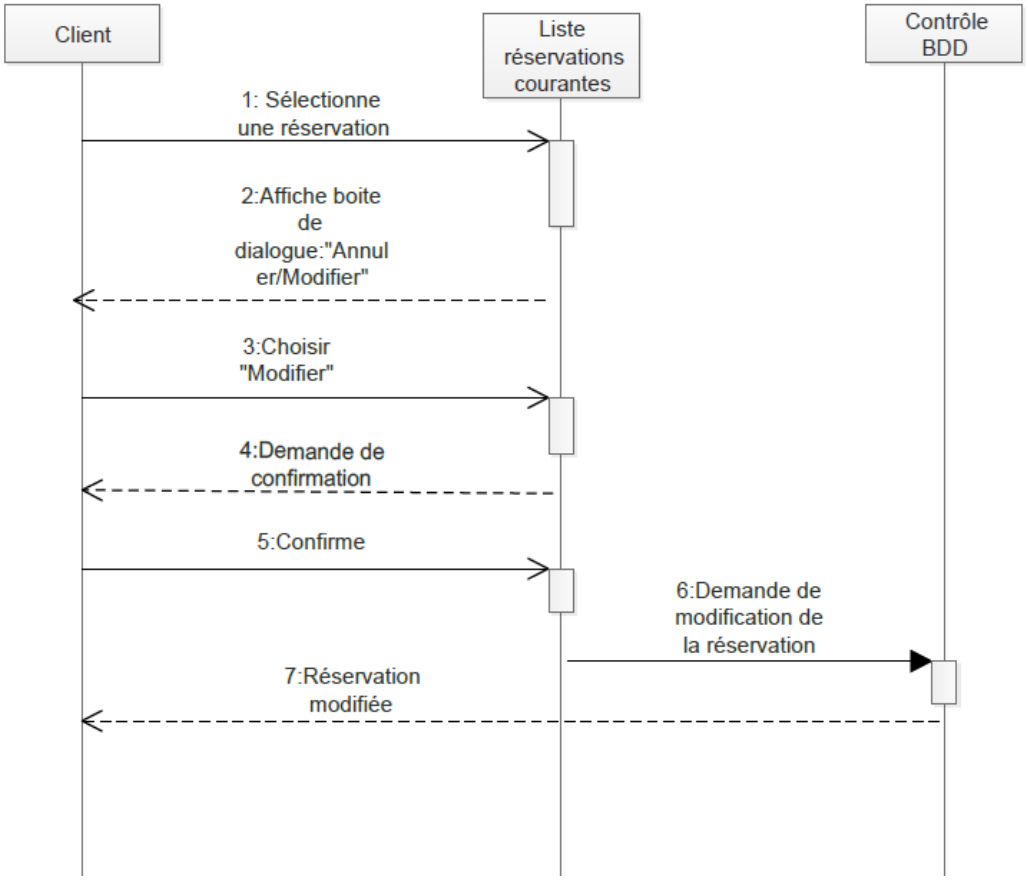


FIGURE III.11 – Diagramme de séquence « Modifier réservation »

d) Diagramme de séquence « Ajouter place »

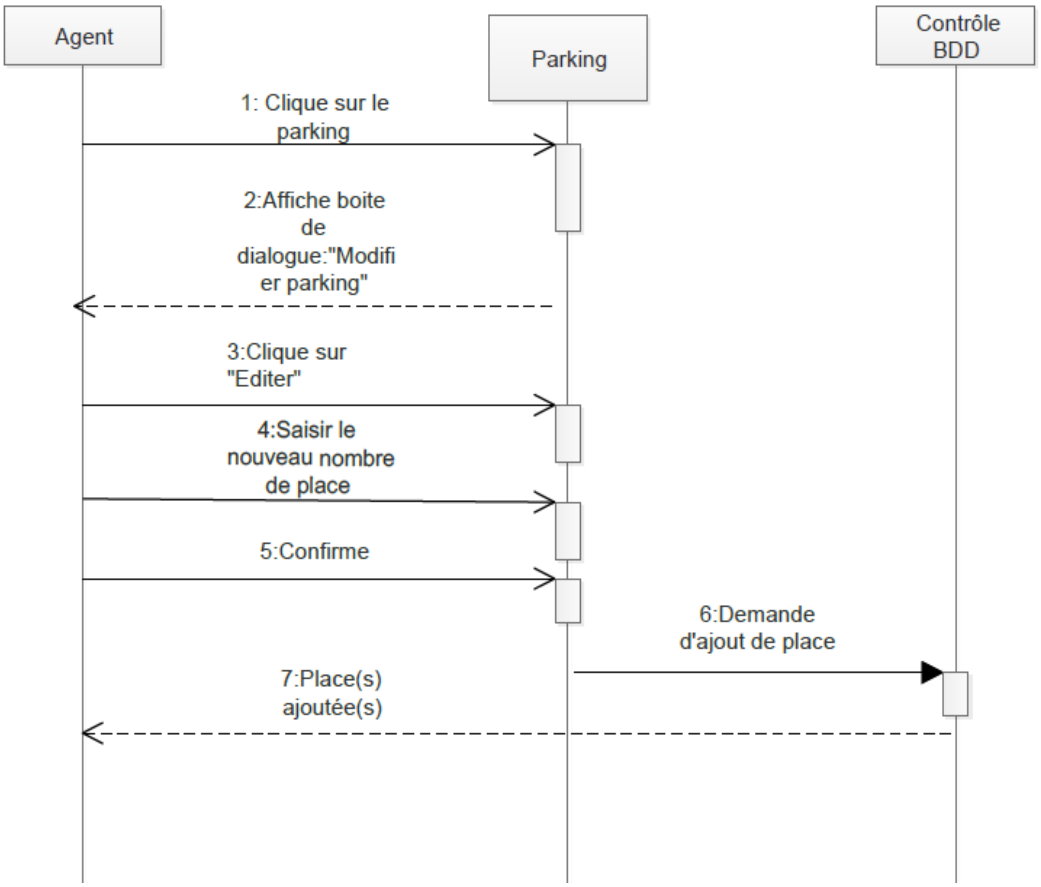


FIGURE III.12 – Diagramme de séquence « Ajouter place »

2. **Diagramme de classe**

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles ci. Ce diagramme fait partie des vues statiques d'UML car il fait une abstraction des aspects temporels et dynamiques.[**10**]
La réalisation des diagrammes de classes se fait à l'aide des associations, nous distinguons quatre types d'association :

- **Lien « Link » :** c'est une association entre une page client et une autre page client ou serveur dans le diagramme de classe. Elle représente un pointeur entre ces pages. A chaque lien correspond une balise ancre HTML.
- **Soumet « Submit » :** une association de soumission se trouve toujours entre un formulaire et une page serveur. Les valeurs des champs du formulaire sont soumises au serveur qui les traite, par l'intermédiaire de pages serveur. Le serveur traite la page serveur qui accepte et utilise les informations de formulaire.
- **Construit « Build » :** c'est une association particulière orientée entre les pages clients et les pages serveurs. Elle indique quelle page serveur est responsable de la création de la page client. Une page serveur peut construire plusieurs pages clients, mais une page client n'est construite que par une et une seule page serveur.
- **Rediriger « Redirect » :** c'est une association unidirectionnelle qui relie deux pages client ou serveur.

— **Quelques diagrammes de classe :**

a) Diagramme de classe « Effectuer réservation »

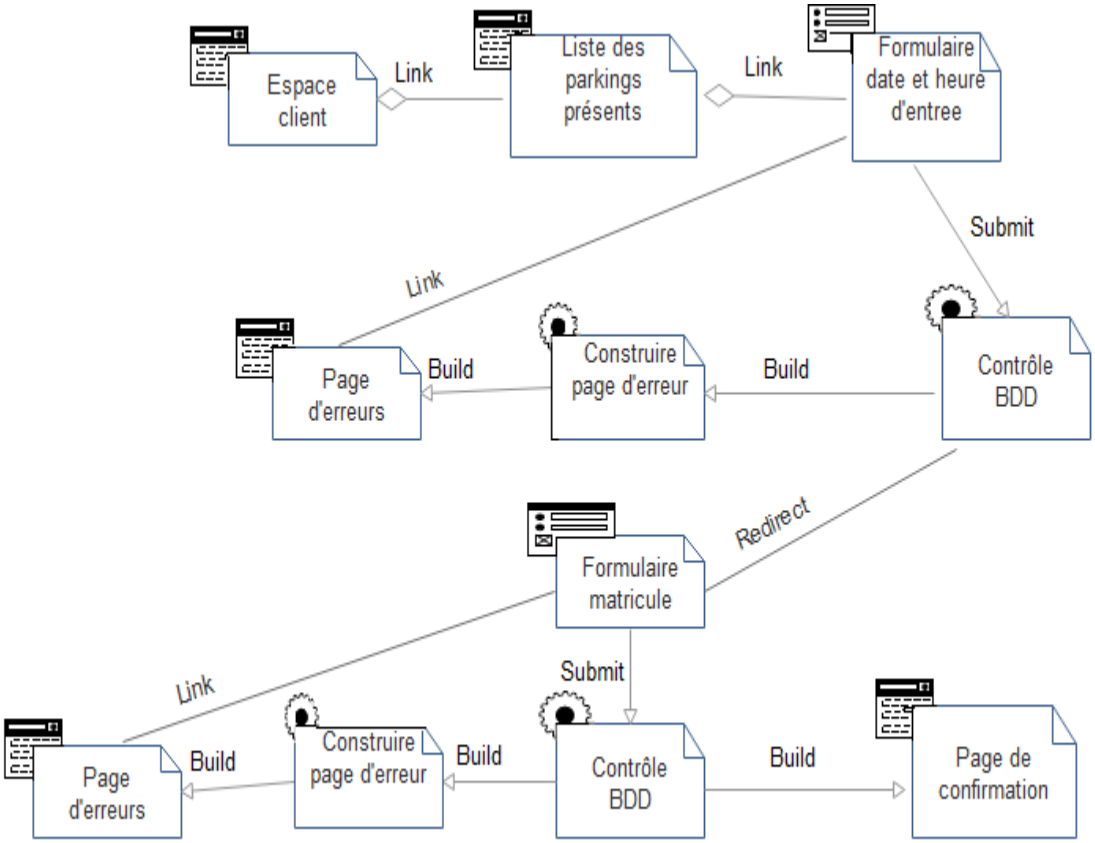


FIGURE III.13 – Diagramme de classe « Effectuer réservation »

b) Diagramme de classe « Annuler réservation »

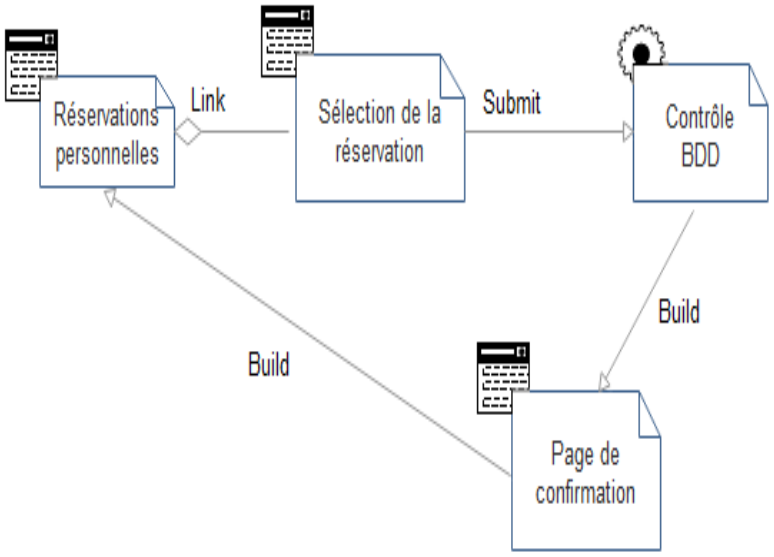


FIGURE III.14 – Diagramme de classe « Annuler réservation »

c) Diagramme de classe « Ajouter place »

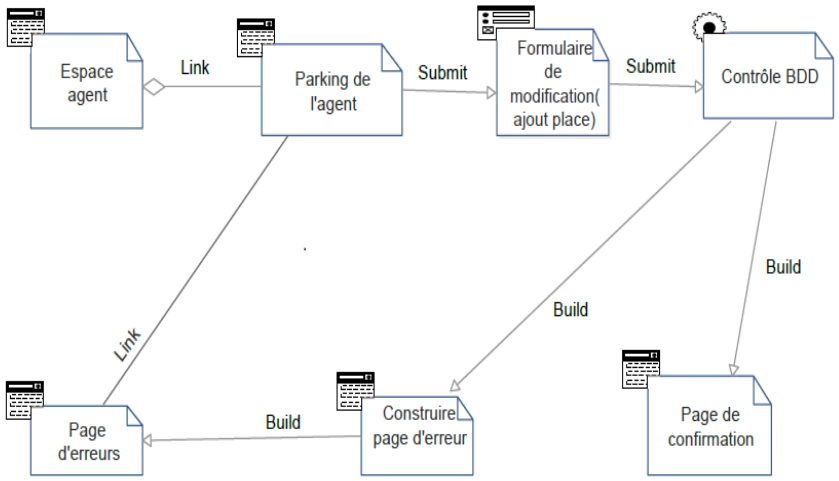


FIGURE III.15 – Diagramme de classe « Ajouter place »

— **Diagramme de classe global**

Le diagramme de classes global est représenté par le schéma suivant :

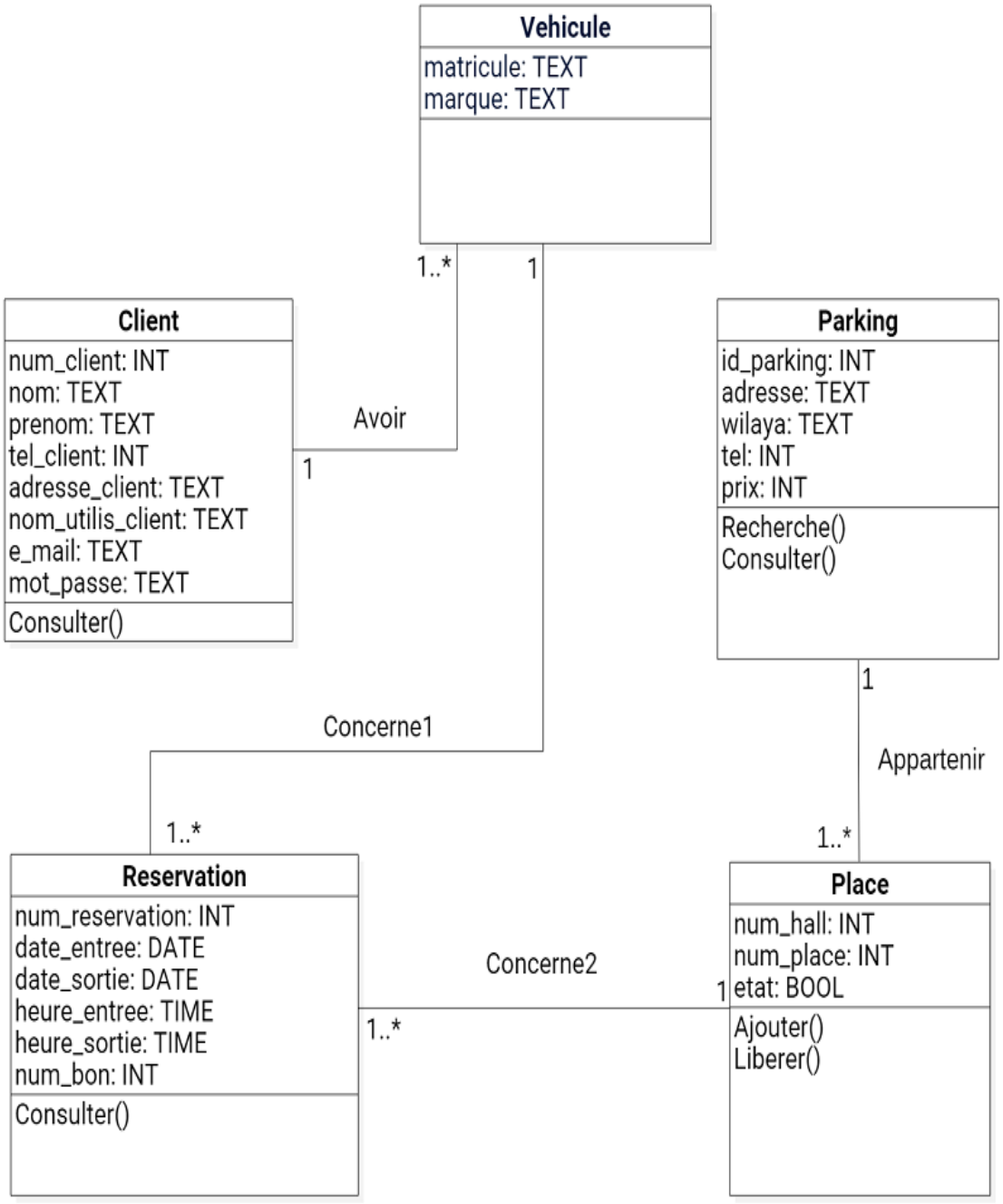


FIGURE III.16 – Diagramme de classe global

3. **Diagramme d'activité**

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flux de contrôle et de flux de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et accessibles. [**10**]

— **Quelques diagrammes d'activité**

a) Diagramme d'activité « Effectuer réservation »

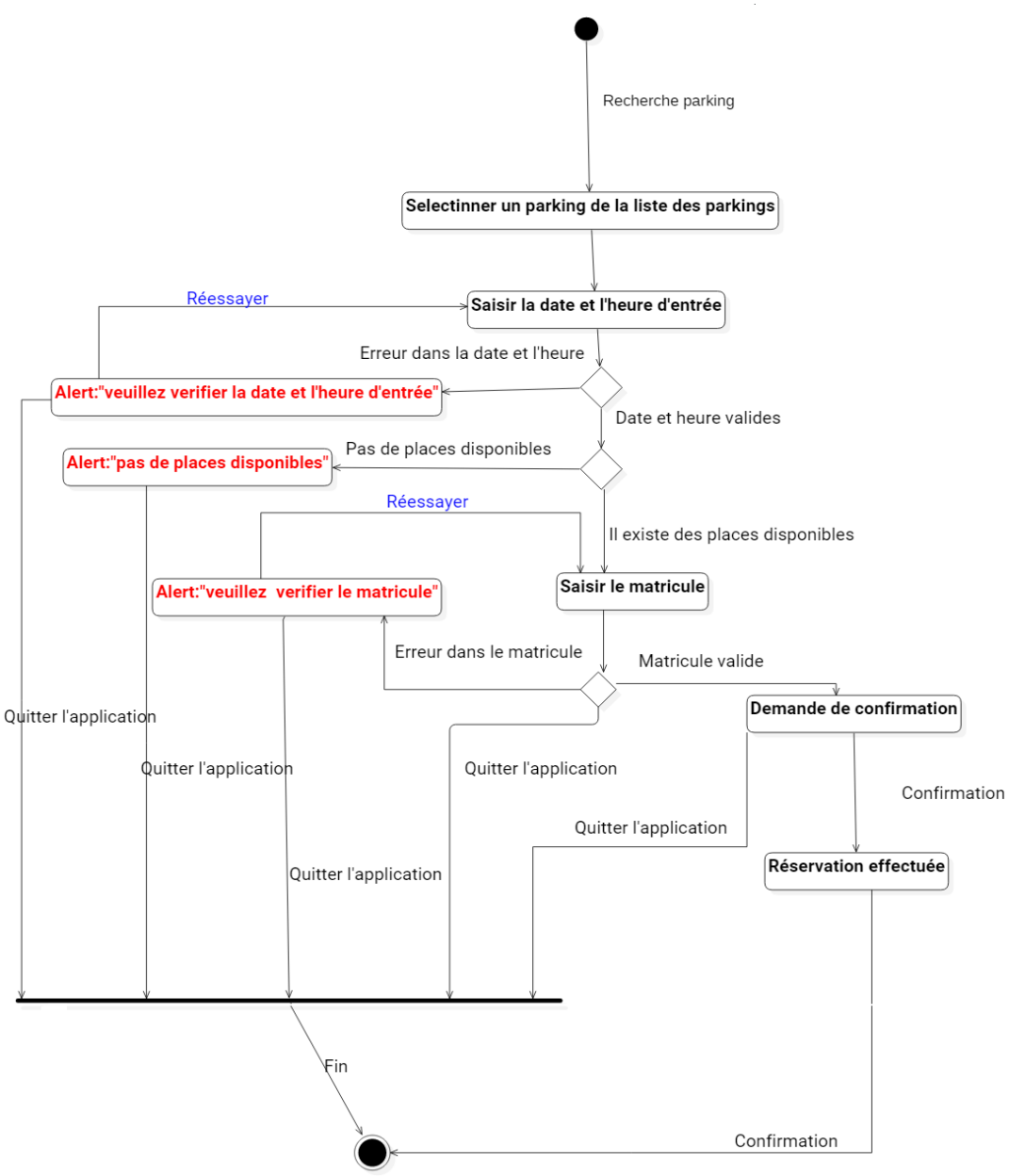


FIGURE III.17 – Diagramme d'activité « Effectuer réservation »

b) Diagramme d'activité « Modifier réservation »

FIGURE III.18 – Diagramme d'activité « Modifier réservation »

c) Diagramme d'activité « Annuler réservation »

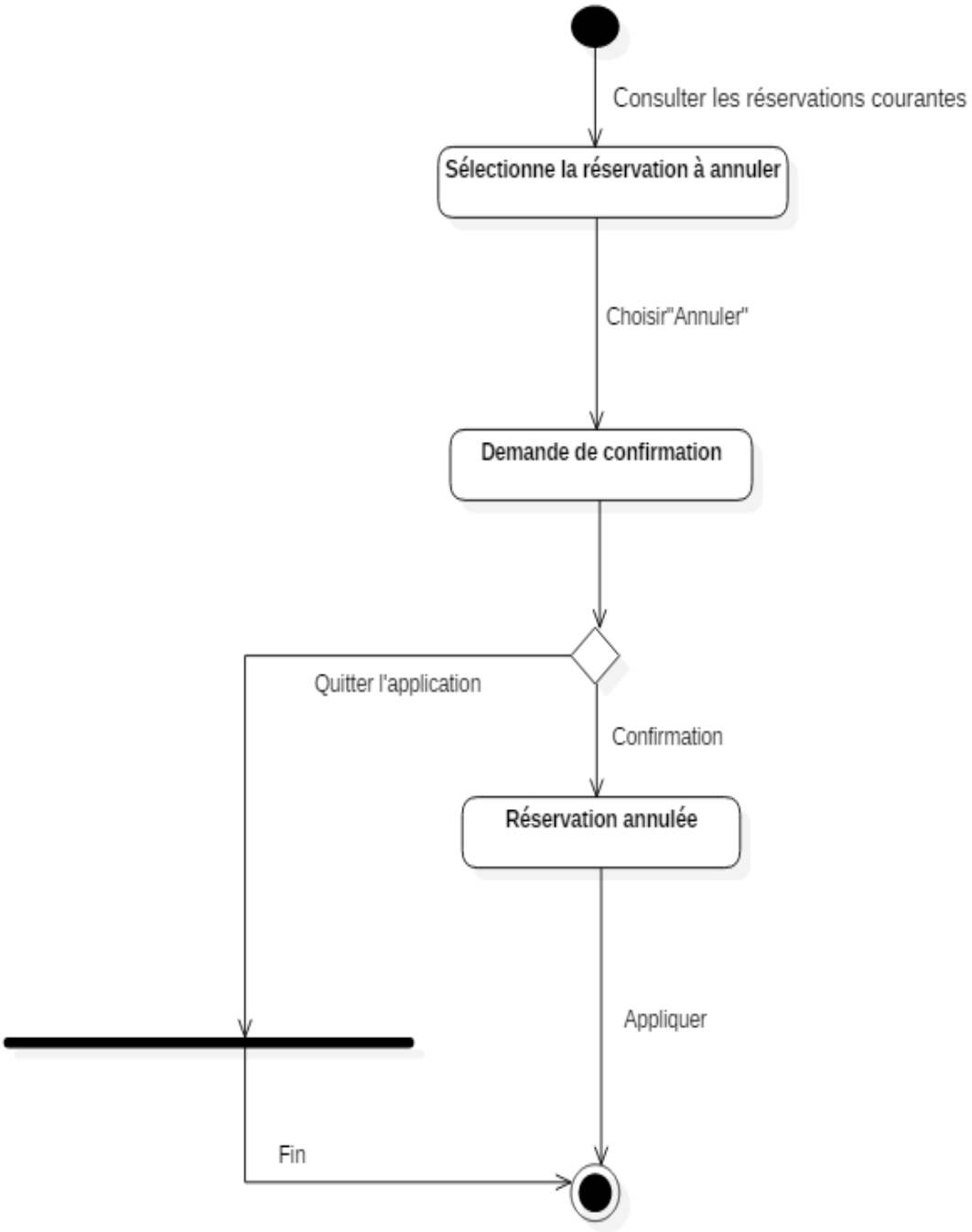


FIGURE III.19 – Diagramme d'activité « Annuler réservation »

d) Diagramme d'activité « Ajouter place »

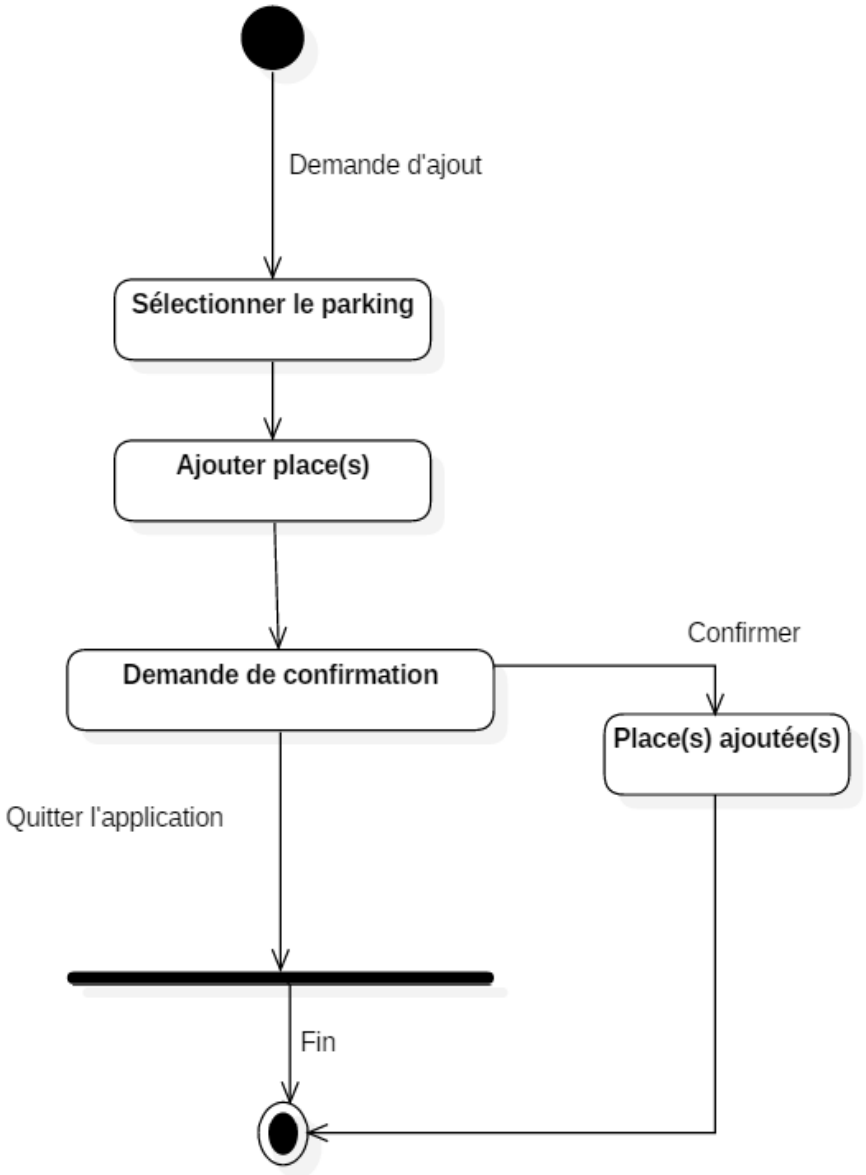


FIGURE III.20 – Diagramme d'activité « Ajouter place »

Conception de la base de données

Une base de données « BDD » est un ensemble de données mémorisées sur des supports accessibles par un ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en temps très court. Elles constituent le cœur du système d'information.
Notre base de données est constituée de sept (05) tables à savoir : la table « client », la table «véhicule », la table « réservation », la table « parking » et la table « place».
Ces six tables sont liées entre elles avec les relations suivantes :

- Un client peut effectuer plusieurs réservations ou ne pas effectuer ;
- Une réservation concerne un et un seul client ;
- Un parking est retrouvé par aucun ou plusieurs clients ;
- Un client peut localiser aucun ou plusieurs parkings ;
- Un client peut posséder un ou plusieurs véhicules ;
- Un véhicule appartient à un et un seul client ;
- Une place appartient à un et un seul parking ;
- Une réservation s'effectue dans un et un seul parking et concerne une et une seule place, un et un seul véhicule ;
- Un parking contient aucune ou plusieurs réservations ;
- Un parking contient une ou plusieurs places.

— **Régles de passage du diagramme de classe au MLD(Modèle Logique de Données) :**
Dans ce qui suit, nous allons présenter les différentes régles de passage, qui nous ont servis lors de l'élaboration du modèle logique de données(MLD) : [**11**]

- Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé.
- Transformation des associations : Nous distinguons trois familles d'associations :

  1. Association 1..*(un à plusieurs) : il faut ajouter un attribut de type clé étrangère dans la relation fille de l'association. L'attribut porte le nom de la clé primaire de la relation mére de l'association.
  2. Association *..* (plusieurs à plusieurs) : la classe-association devient une relation. La clé primaire de cette relation est la concatenation des identifiants des classes connectées à l'association.

3. Association 1..1 (un à un) : il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule.

— **Modèle relationnel de données :**

Le modèle relationnel représente l'information dans une collection de relations, En appliquant les règles de transformation d'un diagramme de classe vers un modèle relationnel(citées dans le paragraphe précédent), nous avons aboutit au schéma relationnel suivant :

- Client(**num_client**,nom_client,prenom_client,tel_client,adresse_client).
- Reservation(**num_reservation**,date_entree,date_sortie,heure_entree,heure_sortie, num_place*,matricule*,num_bon).
- Parking(**id_parking**,adresse,wilaya,tel,prix).
- Place(**num_place**,num_hall,etat,id_parking*).
- Vehicule(**matricule**,marque,num_client*).

Remarque :

Les attributs désignés par * sont des clés étrangères(des clés primaires pour d'autres tables).

— **Les différentes tables avec la codification utilisée :**

Chaque table stockée dans notre base de données contient un ensemble d'attributs qui seront détaillés dans ce qui :

1. **Table client :**

Champ	Signification	Type	Clé
num_client	Numéro unique pour chaque client (auto incrémenté)	INT	Primaire(Primary Key)
nom	Le nom du client	TEXT	
prenom	Le prénom du client	TEXT	
adresse_cient	L'adresse du client	TEXT	
tel_client	Le numéro de téléphone du client	INT	
nom_utilis_client	Le nom utilisateur du client pour accéder aux services fournis par l'application(unique pour chaque client)	TEXT	
e_mail	L'adresse électronique du client	TEXT	
mot_passe	Le mot de passe attribué au nom d'utilisateur	TEXT	

TABLE III.2 – Table client

2. **Table vehicule :**

Champ	Signification	Type	Clé
matricule	La matricule du véhicule	TEXT	Primaire(Primary Key)
marque	La marque du véhicule	TEXT	
num_client	Le numéro du client(propriétaire du véhicule)	TEXT	Etrangère(Foreign Key)

TABLE III.3 – Table vehicule

3. **Table reservation :**

Champ	Signification	Type	Clé
num_reservation	Numéro unique attribué à chaque réservation (auto incrémenté)	INT	Primaire(Primary Key)
date_entree	La date d'entrée au parking	DATE	
date_sortie	La date de sortie du parking	DATE	
heure_entree	L'heure d'entrée au parking	HEURE	
heure_sortie	L'heure de sortie du parking	HEURE	
matricule	Le matricule	INT	Etrangère(Foreign Key)
num_bon	Numéro du bon	INT	
num_place	Numéro de la place	INT	Etrangère(Foreign Key)

Table III.4 – Table reservation

4. **Table place :**

Champ	Signification	Type	Clé
num_place	Le numéro de la place	INT	Primaire(Primary Key)
id_parking	Le numéro du parking	INT	Etrangère(Foreign Key)
num_hall	Le numéro du hall	INT	
etat	L'état de la place : occupée/-libre	Booleen	

Table III.5 – Table place

5. **Table parking** :

Champ	**Signification**	**Type**	**Clé**
id_parking	Numéro du parking	INT	Primaire(Primary Key)
adresse	L'adresse du parking	TEXT	
wilaya	La wilaya ou se trouve le parking	TEXT	
tel	Le numéro de téléphone du parking(agent du parking)	INT	
prix	Le prix unitaire de réservation pour chaque parking (Da/H)	INT	

TABLE III.6 – Table parking

III.4 Conclusion

A l'issue de ce chapitre, nous avons cerné les objectifs de notre application et pour les atteindre nous avons suivi une démarche de modélisation basée sur la méthode UML.

Pour la phase d'analyse, nous avons présenté quelques diagrammes de cas d'utilisations, en revanche, pour la partie conception, nous avons construit les diagrammes de séquence, de classes et enfin nous avons conclus avec qulques diagrammes d'activité.

Le prochain chapitre sera consacré à la présentation de l'environnement de travail et des différents outils utilisés ainsi qu'à la présentation de quelques interfaces de notre application.

Chapitre IV

Fonctionnement et Réalisation

IV.1 Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, nous allons à présent décrire l'environnement de développement et les outils qui ont servi à la mise en oeuvre de notre application. Enfin, nous terminerons par la présentation des différentes fonctionnalités qu'offre notre application Android à travers diverses interfaces.

IV.2 Environnement de développement :

Notre application mobile sous Android a été développée sous le système d'exploitation Windows 10 Professionnel, tout en exploitant le serveur Web Apache et le serveur de base de données MySQL, ce qui nous impose de travailler sous Eclipse avec le langage de programmation JAVA et le SDK Android pour son développement afin d'obtenir un fichier .APK qui sera par la suite installé sur les terminaux mobiles de type Smartphone fonctionnant sous système Android.

IV.2.1 Environnement matériel

- **PC n°1 :** Un **Dell** avec les caractéristiques suivantes :
  - Un processeur Intel(R) Core (TM) i3 CPU M 380 @2.53 GHZ 2.53GHZ
  - Une RAM de 3Gigas ;
  - Un disque dur de 500 Gigas ;
  - Système d'exploitation Microsoft Windows 10.
- **PC n°2 :** Un **MSI** qui a les caractéristiques suivantes :
  - Un microprocesseur Intel B970 ;
  - Une RAM de 4Gigas ;
  - Un disque dur de 500 Gigas ;

— Système d'exploitation Microsoft Windows 10.

IV.2.2 Environnement logiciel

Outils de développement

1. **Pourquoi WAMP (Windows Apache MySQL PHP) ?**
   - Utiliser Windows 10 professionnel en raison de sa simplicité d'utilisation, sa popularité et convivialité et surtout pour sa puissance ainsi pour toutes les ressources et options qu'il propose.
   - **Apache** est l'un des serveurs web les plus utilisé sur Internet (près de 57web sont administrés par le serveur Apache). Apache est multi plate-forme et gratuit, son installation est facile et rapide et son utilisation pas trop compliqué. Grâce à une association avec PHP, Apache devient un serveur Web dynamique.
   - **MySQL** est un gestionnaire de base de données SQL robuste. Il se compose d'un langage de définition de données et de droit ainsi qu'un langage de manipulation des données. Il est rapide, multithread et multi utilisateurs.
   - **PHP** est un langage généralisé pour la production de contenu Web dynamique, associé à un serveur web Apache et une base de données MySQL, c'est aujourd'hui une solution gratuite, fiable et puissante pouvant être déployé sur de nombreuses plateforme telles que Windows.
2. **Le serveur Web Apache :**
   Le serveur Web Apache (www.apache.org) est le serveur le plus répandu sur Internet. Il s'agit d'une application fonctionnant à la base sur les systèmes d'exploitation de type Unix, mais il a désormais été porté sur de nombreux systèmes, dont Microsoft Windows.Apache (prononcez à la française ou bien pour les puristes à l'anglophone « Apatchy »)tiré son nom de la façon dont il a été mis au point (« A patchy server » traduisez « un serveur rafistolé») car il est le fruit d'une multitude de correctifs logiciels afin d'en faire une solution très sûre. En effet Apache est considéré comme sûr dans la mesure où peu de vulnérabilités le concernant sont publiées.
   Tout développement de site web requiert un serveur Web qui s'occupe de traitement des requêtes des clients et l'exécution des programmes sur la machine serveur.
   Notre méthode s'est basée sur Apache, du fait qu'il est le serveur Web le plus utilisé présentant un niveau de performances élevé pour des exigences matérielles modestes, en plus de sa garantie et sa robustesse.[**12**]

3. **Le serveur de base de données MySQL :**

   MySQL est un système de gestion de bases de données relationnelles (SGBDR) robuste et rapide. Une base de données permet de manipuler les informations de manière efficace, de les enregistrer, de les trier, de les lire et d'y effectuer des recherches. Le serveur MySQL contrôle l'accès aux données pour s'assurer que plusieurs utilisateurs peuvent se servir simultanément d'une même base de données pour y accéder rapidement et pour garantir que seuls les utilisateurs autorisés peuvent accéder aux données. MySQL est donc un serveur multiutilisateur et multithread. Il utilise SQL (Structured Query Language), le langage standard des requêtes de bases de données. MySQL est disponible depuis 1996, mais son développement remonte à 1979. Il s'agit de la base de données open-source la plus employée au monde.

   Il est implémenté sur mode client-serveur avec du côté serveur : le serveur MySQL, et du côté client : les différents programmes et librairies.

   MySQL se caractérise par sa rapidité et sa facilité d'utilisation, il est offert avec l'outil D'administration de base de données « PhpMyadmin » par WampSever 3.0.4.

4. **Eclipse (logiciel) :**
   Eclipse IDE « Integrated Development Environment » est un environnement de développement intégré principalement écrit en Java, libre, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en oeuvre n'importe quel langage de programmation.
   La spécificité d'Eclipse IDE vient du fait que son architecture soit totalement développée autour de la notion de plugin : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plugin notamment le plugin « Android ».
   Dans le cadre de notre projet, nous avons utilisé la version Eclipse Mars, avec le plugin ADT de Google.

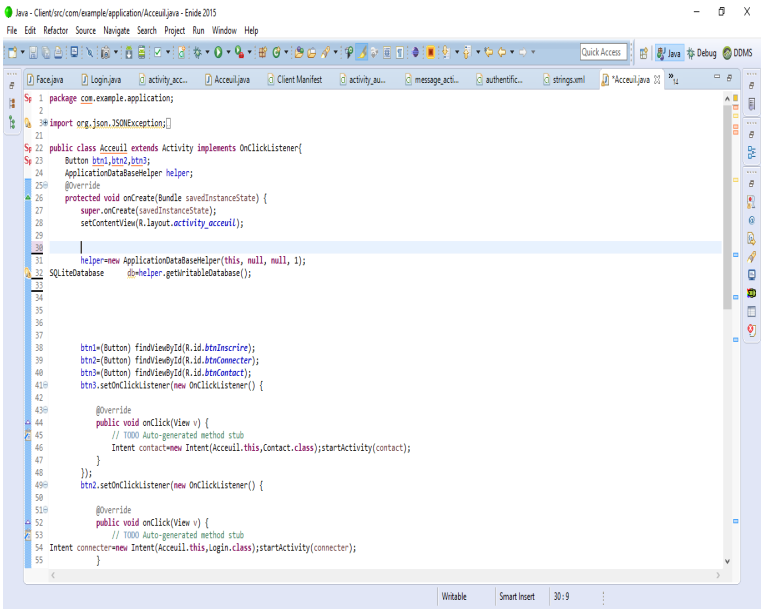


FIGURE IV.1 – Capture d'écran d'une fenêtre Eclipse.

5. **WampServer :** (anciennement WAMP5)
   WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

   Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icon (icône près de l'horloge de Windows).

   La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

Les langages de programmation

- **Le langage XML :** Le langage XML « eXtended Markup Language » est un format général de documents orienté texte. Il s'est imposé comme un standard incontournable de l'informatique. Il est aussi bien utilisé pour le stockage de documents que pour la transmission de données entre applications. Sa simplicité, sa flexibilité et ses possibilités d'extension ont permis de l'adapter à de multiples domaines.
  De nombreuses technologies se sont développées autour de XML et enrichissent ainsi son environnement. Le langage XML dérive de SGML « Standard Generalized Markup Language » et de HTML « HyperText Markup Language ».
  Grâce à ce langage, il devient possible de décrire les interfaces sous la plateforme Android dans un format spécial. Celles-ci seront converties automatiquement en objets Java qui seront par la suite disponibles comme tout autre objet dans le code. XML offre ainsi plus de souplesse de développement, facilite les modifications du code et assure la séparation entre la présentation et le comportement des objets.[**13**]
- **Le langage PHP :**PHP est un langage de script côté serveur qui a été conçu spécifiquement pour le Web.
  Le code PHP est interprété au niveau du serveur web et génère du code HTML ou toute autre donnée affichable dans le navigateur de l'utilisateur.PHP a été conçu en 1994 par RasmusLerdorf. Il a ensuite été adopté par d'autres personnes talentueuses et réécrit quatre fois avant de devenir le produit abouti que nous connaissons aujourd'hui.
  PHP est un projet open-source, ce qui signifie qu'on peut se procurer son code, l'utiliser, le modifier et le redistribuer gratuitement.
  PHP signifiait à l'origine Personal Home Page, mais ce nom a été changé en un acronyme récursif comme GNU (Gnu's Not Unix) : il signifie maintenant PHP Hyper text Preprocessor. La dernière version principale de PHP est la version 5.[**14**]
- **Le langage java :** est un langage généraliste de programmation synthétisant les principaux langages existants lors de sa création en 1995 par Gosling et Patrick Naughton, employés de Sun Microsystems. Il permet une programmation orientée objet, modulaire et reprend une syntaxe très proche de celle du langage C. Outre son orientation objet, le langage Java a l'avantage d'être modulaire (on peut écrire des portions de code génériques, c'est-à-dire utilisables par plusieurs applications), rigoureux (la plupart des erreurs se produisent à la compilation et non à l'exécution) et portable (un même programme compilé peut s'exécuter sur différents environnements à savoir sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou Linux). En contrepartie, les applications Java ont le défaut d'être plus lentes à l'exécution que des applications programmées en C par exemple.

- **Langage SQL :**Etant données que les requêtes SQLITE sont basées sur le standard SQL, une petite présentation de ce dernier s'impose.
  En effet, SQL « Structured Query Language » est un langage de manipulation de bases de données mis au point dans les années 70 par IBM. Il permet notamment :
  - **La manipulation des tables :** création, suppression, modification de la structure des tables ;
  - **La manipulation des bases de données :** sélection, modification et suppression d'enregistrements ;
  - **La gestion des droits d'accès aux tables :** contrôles des données et validation des modifications.

IV.3 Présentation des interfaces :

Nous allons présenter dans ce qui suit quelques interfaces de notre application mobile, qui garantit une meilleure convergence entre notre système et les besoin de ces futurs utilisateurs.

1. **Interfaces communes entre :Espace visiteur,Espace client et L'Espace agent :**

   — **Page d'acceuil :**
   C'est la premiere interface visualisée par l'utilisateur lors d'accés à l'application :
   — Le premier bouton « **S'inscrir** » permet à l'utilisateur de créer un compte ;
   — Le Le deuxième bouton « **S'authentifier** » permet à l'utilisateur d'accéder à son espace ;
   — Le troisième bouton « **Contact** » permet a l'utilisateur de contacter par téléphone l'agent d'un parking.

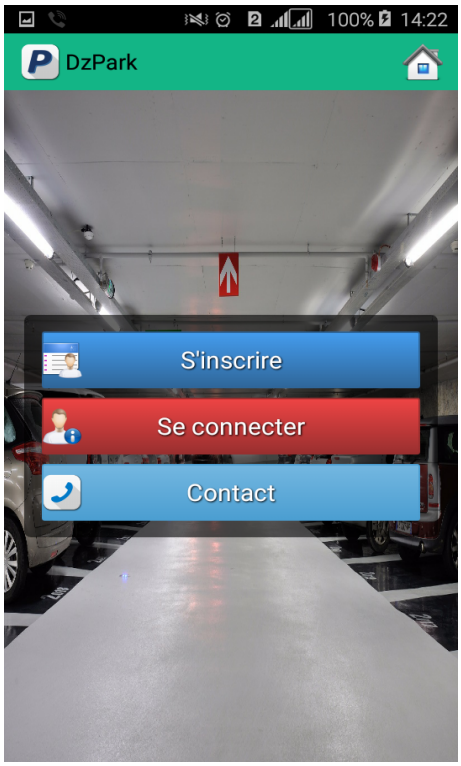


FIGURE IV.2 – Interface d'accueil de l'application.

— **Page Authentification :**
L'accès aux services de l'application nécessite une authentification avec nom d'utilisateur et un mot de passe via un formulaire de connexion.

FIGURE IV.3 – Page Authentification

— **Page Inscription :** La page inscription permet de s'inscrire et d'avoir un compte pour accèder aux differents services offerts par l'application.

FIGURE IV.4 – Page Inscription.

2. **Espace visiteur :**
   C'est la page téléchargée et visualisée par le visiteur lorsqu'il clique sur le bouton Home, elle lui permet d'avoir une idée générale sur le contenu de l'application, à partir de laquelle il peut naviguer ouvertement sur l'application grâce au menu principal, il peut ainsi découvrir les différentes fonctionnalités offertes par cette derniere tel que la réservation et la recherche d'un parking, et contacte qui lui permet d'appeler directement l'agent d'un parking choisit.La page d'accueil offre aussi des liens qui permettent d'accéder à d'autres pages de l'application pour voir les différentes prestations de notre parking.

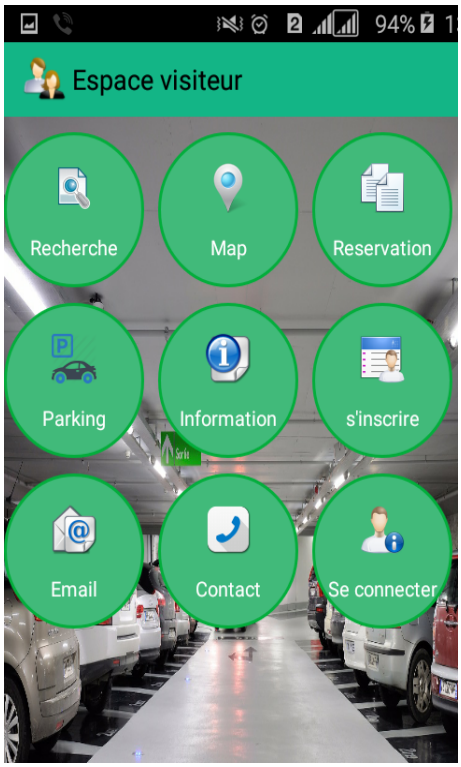


FIGURE IV.5 – Page Espace Visiteur

3. **Espace client :**

— **Page Recherche :**

En appuyant sur la bulle « **Recherche** » figurant sur l'interface visiteurs de l'application et sur l'espace Client, l'utilisateur est encore une fois invité à effectuer une « Recherche » qui lui permettent respectivement d'effectuer une recherche de places, et de parkings disponibles.

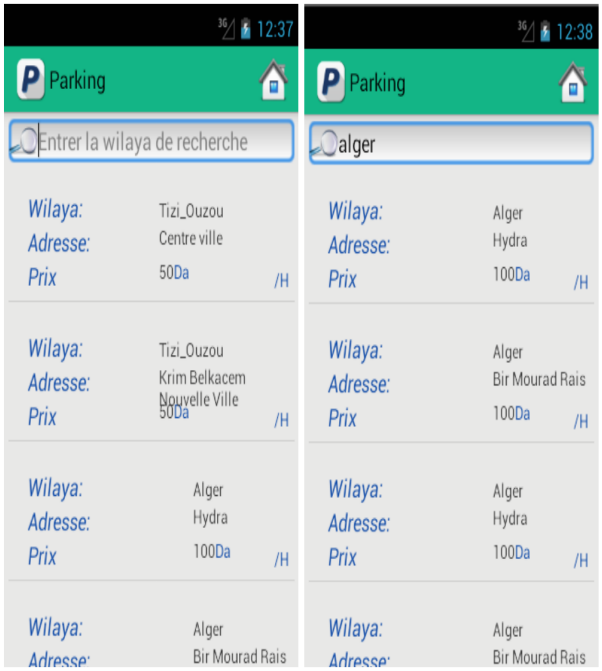


FIGURE IV.6 – Page Recherche Parking

— **Page éffectuer réservation :**

Après avoir trouver un parking, le client doit selectionner une date et une heure d'entrée, remplit le champ ***Matricule*** en cas de disponibilité de place,puis il confirme sa réservation.

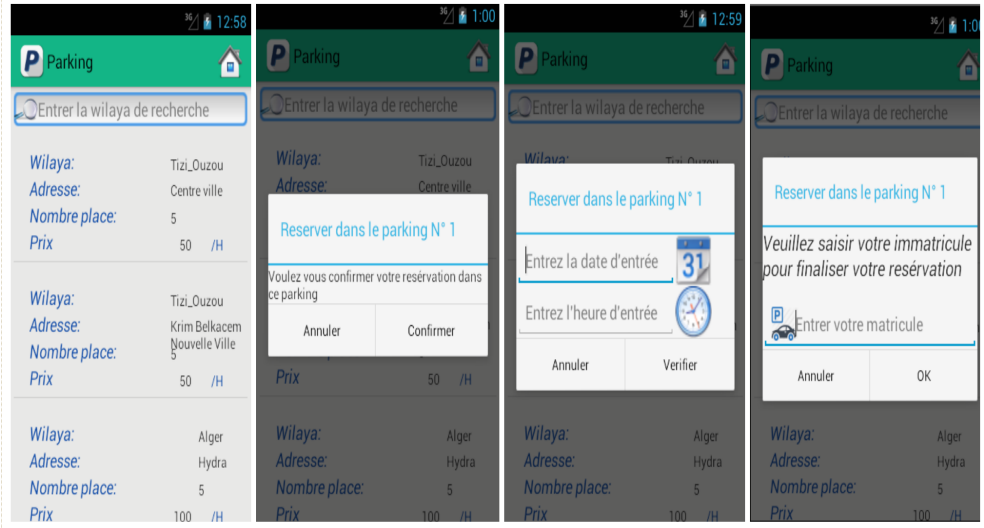


FIGURE IV.7 – Les étapes de réservation

— **Page Compte :**

Le Client à la possibilité de changer son mot de passe, il lui suffit d'accéder à **Mon compte** et de cliquer sur le lien changer mot de passe et de remplir le formulaire qui lui est présenté et puis cliquer sur le bouton valider, et aussi consulter la liste des bons de toutes ces réservations.

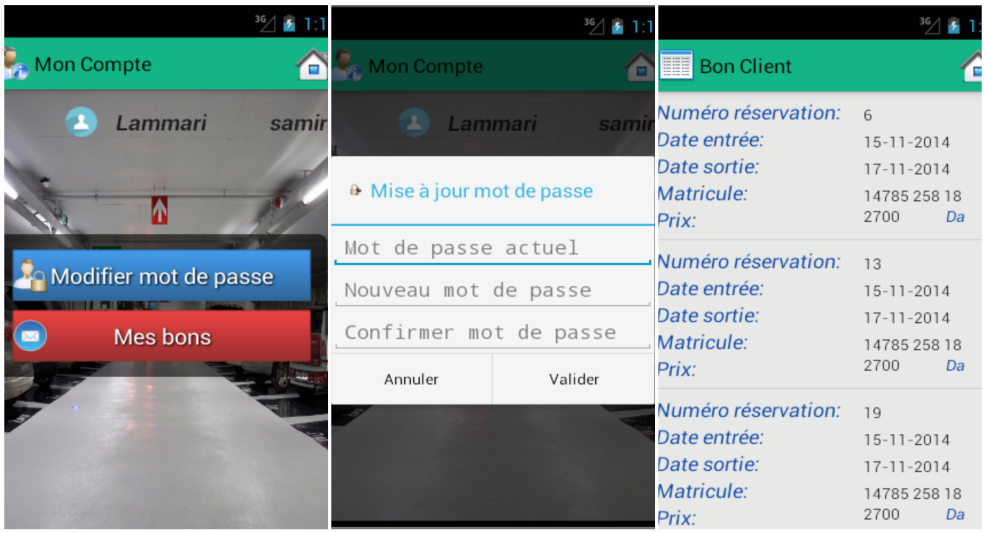


FIGURE IV.8 – Page Compte

— **Page envoyer email :**
Le client a la possibilité via la section contact d'envoyer un email pour l'agent. Pour le faire il doit choisir le lien email et remplir les champs.

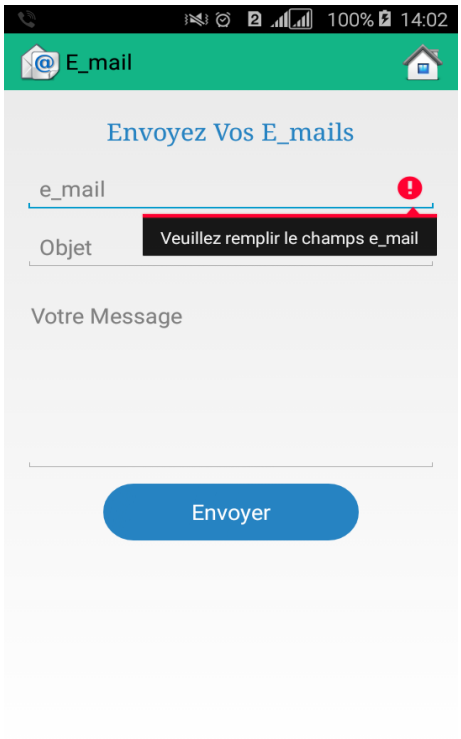


FIGURE IV.9 – Page d'envoie d'un email.

— **Page contacter agent :**
Le client a la possibilité de contacter l'agent par téléphone en choisissant un parking et cliquer dessus pour appeler.

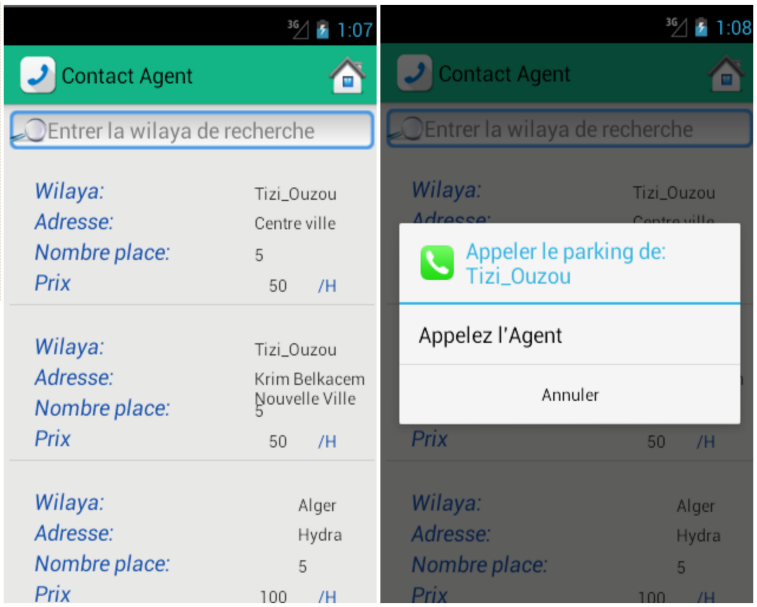


FIGURE IV.10 – Page contacter Agent

4. **Espace agent :**
   L'interface Agent est réservée aux agents du parking qui auront pour tâche de gèrer les activités(places, réservations,...) du parking.

FIGURE IV.11 – Page interface Agent

— **Contacter Client :**
   L'agent quand il cliquera sur un client une boite de dialogue s'affichera, il pourra choisir d'envoyer un e_mail ou d'appeler, l'adresse e_mail et le numéro de téléphone du client seront générés automatiquement, il pourra aussi annuler en cliquant sur **Annuler**.

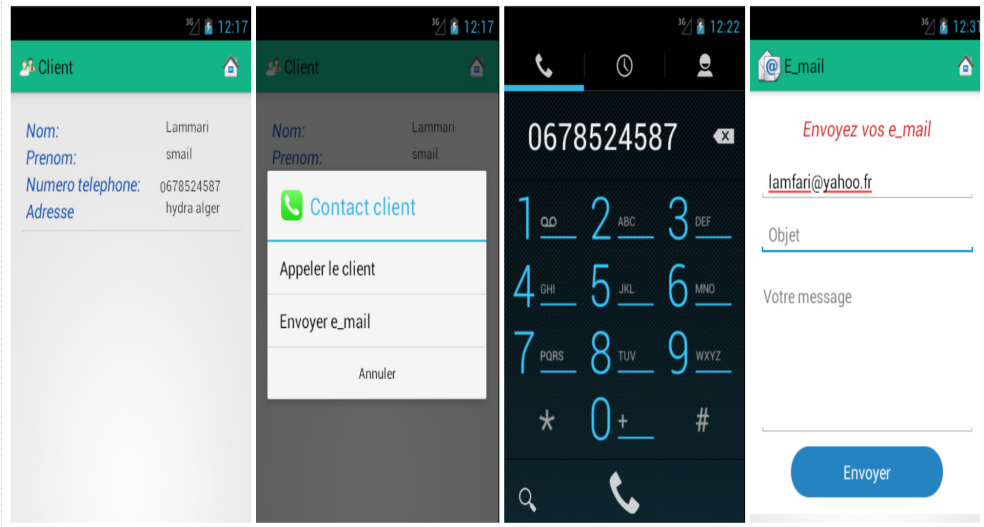


FIGURE IV.12 – Page Contacter Client

— **Ajouter places :** L'agent peut mettre à jour le nombre de place du parking.

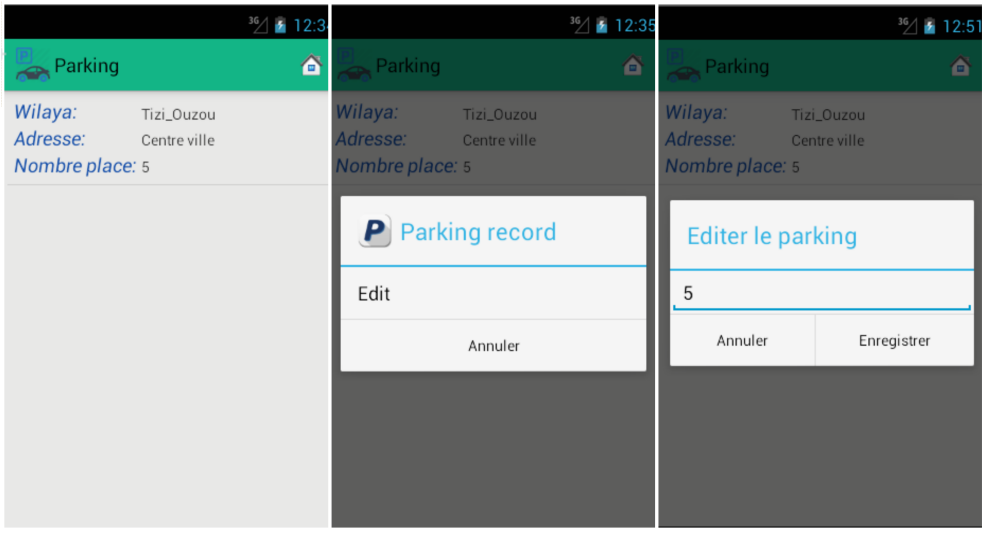


FIGURE IV.13 – Page Edite Place

— **Liberer places :** L'agent peut liberer la place aprés la sortie du client.

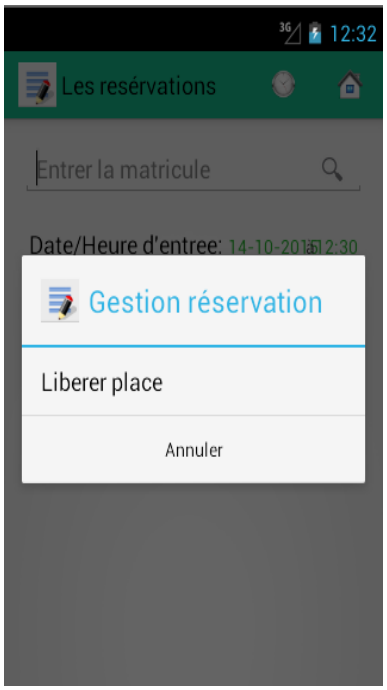


FIGURE IV.14 – Page liberer place

IV.4 Conclusion

Ce chapitre offre une description plus ou moins détaillée des interfaces que nous avons développées. Il étale un peu l'aspect réalisation de notre application. Toutefois, notre travail reste ouvert pour des extensions et des améliorations éventuelles.

Conclusion générale

Pour garer un véhicule, où foisonnent les plaques interdisant le stationnement, l'automobiliste doit à chaque fois effectuer un véritable parcours du combattant, qui occasionnent des bouchons de circulation, font perdre du temps précieux aux automobilistes, entraînent un gaspillage de carburant et font tourner les activités économiques au ralenti. S'il s'aventure à placer son véhicule à l'endroit «placardé», le sabot immobilise immédiatement le véhicule. En plus du désagrément moral, le « fautif » est obligé de payer une forte amende. Cette situation n'incommode ni les élus de la commune ni les autorités de la wilaya qui ont le «droit» de griller le code de la route, et aussi la mauvaise gestion des espaces de stationnement et les renseignements insuffisants sur la disponibilité des espaces .
Les systèmes de stationnement intelligents, qui offrent des renseignements en temps réel sur la disponibilité des espaces de stationnement, facilitent grandement la recherche d'espaces de stationnement.

Notre projet a pour but de louer sa prope place de parking le plus proche à sa position actuelle en temps réel et au meilleurs prix.

Il vise, entre autres, à exploiter les technologies avancées des nouveaux mobiles présents sur le marché.

C'est dans ce cadre que s'inscrit notre application de réservation sur mobile, les automobilistes peuvent faire appel à cette nouvelle technologie ; notre application mobile permet de gagner du temps et accéder à un parking choisi par le client en quelques clics seulement. Cette expérience a été une occasion pour bien cerner le concept de simulation de contexte à travers le développement (en java) d'une simulation . De plus, nous avons approfondi nos connaissances dans le développement orienté objet et nous avons appris l'utilisation et la manipulation du langage java sur mobile.

Au bout de ce projet, nous avons pu développer une application qui aide les automobilistes à trouver une place dans un parking d'une manière assez facile sans perdre son temps . Nous avons, en outre, programmer notre application et communiquant cette dernière à un serveur (que nous avons aussi mis en place). Touefois notre application reste ouverte pour tout eventuelle amélioration, la faire connecter à des réseaux de capteurs qui détectent l'occupation des espaces de stationnement etc..

Bibliographie

[1] Sébastien PEROCHON. *Android Guide de développement d'applications pour Smartphones et Tablettes*. Edition ENI, juillet 2011.

[2] Mark MURFY. *L'art du developpement Android*. PEARSON, 2009.

[3] Emmanuel ROBLES Damien GUIGNARD, Julien GHABLE. *Programmation Android,de la conception au deploiement avec le SDK Google Android 2*. Eyrolles, 2009.

[4] Alain MENU. *Android Howto : Présentation générale*. Edition 2.1, septembre 2013.

[5] Reto MEIER. *Android 4 Développement d'applications avancées*. PEARSON, 2012.

[6] Florent GARIN. *Développer des applications mobiles pour les Google Phones*. DUNOD, 2009.

[7] Les systèmes de transport intelligents en france(sti). http ://www.google.com/Brochure_STI_FR_Finale.

[8] Système de transport intelligent, juin 2016. https ://fr.wikipedia.org/wiki/Systéme_de_transport_intelligent.

[9] Pascal ROQUES. *UML 2 par la pratique,5° édition*. EYROLLES, 2006.

[10] Christine SOLNON. *Modèlisation UML*. PEARSON, 2013.

[11] Eric GODOC. *SQL, les fondamentaux du langage*. 2ème édition ENI, 2014.

[12] Appache introduction. http ://www.commentcamarche.net/contents/8-apache-introduction.

[13] Hypertext markup language. http ://fr.wikipedia.org/wiki/Hypertext_Markup_Language.

[14] Programmation php. http ://www.google.com/coursPHP Principes.