

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Bordj Bou Arréridj
Faculté des mathématiques et d'informatiques
Département Informatique



MÉMOIRE

Présenté en vue de l'obtention du diplôme
Master en informatique

Spécialité : Technologies de l'Information et de la Communication.



Thème :

Conception et réalisation d'une application mobile pour un parking intelligent.

Présenté par :

- **Boumergued Ahlem**
- **Nasri Sarah**

Soutenu publiquement le :12 /10/2020 devant le jury composé de :

Président :	M. Saifi Abdelhamid	MAA à L'U.El Bachir El Ibrahimi-BBA.
Examinatrice :	M. Belhadj Foudil	MCA à L'U.El Bachir El Ibrahimi-BBA.
Encadrant :	M.Nouioua Farid	MCA à L'U.El Bachir El Ibrahimi-BBA.

Promotion : 2019/ 2020.

Dédicace

Je dédie ce mémoire à.....

Mes chers parents

Vous êtes les parents les plus merveilleux, Pour tous vos grands sacrifices et votre patience illimitée, votre amour, votre aide, vos encouragements et vos prières tout au long de ma vie.

Mon Père : Papa, personne ne viendra comme lui, et personne ne prendra sa place.

ابي الحبيب

Ma Mère : Le cœur le plus merveilleux est ton cœur, les plus beaux mots sont tes chuchotements, et la chose la plus douce de ma vie est que tu es ma mère.

امي ثم امي ثم امي

Mes chers frères : Salim et Lamri

Pour leur grand amour et leur soutien qu'ils trouvent ici l'expression de ma haute gratitude.

اخواني الأعزاء

Ma chère sœur : Nessrin

اختي توأم الروح

Pour son encouragement permanent, et son soutien moral

Ma sœur ma meilleure amie

Ma tante : Wahiba

خالتي الحبيبة

Pour tout ce que tu m'as donné et pour ton amour

Mes chères amies :

Qui sans leur encouragement ce travail n'aura jamais vu le jour surtout Sarah Nasri.

Dédicace

Je rends un grand hommage à travers ce modeste travail, en
signe de respect et de reconnaissance envers :

A mon père pour son amour, son soutien et pour ses
encouragements et ses précieux conseils durant toute ma vie.

A ma mère pour son amour inconditionnel et sa présence à
mes côtés dans les moments difficiles.

Je te serai reconnaissante toute ma vie, qu'Allah t'accorde
longue vie dans la santé !

A mes chers frères : AYMENE, KHLEDE.

Pour leur appui, leur grand amour et leur encouragement.

A mes chères tantes : ASSIA, NINA.

Pour leur encouragement permanent, et leur soutien moral

A mes chères amies : Oussama, BADRO, Imane, MESSOUDA,
Hayat, HOUDA, MANEL et AHLEM et tous mes camarades de
la promotion 2020.

A toutes les personnes que j'aime et a tous ceux qui par leurs
conseils, leurs attentions, leurs encouragements et leurs
soutient m'ont aidé à réaliser ce travail.

NASRI SARA ✍

Remerciement

Tout d'abord, nous voulons remercier le Dieu " ﷻ "Tout-Puissant qui nous a donné la force et la volonté de réaliser au maximum de notre travail.

Nous tenons à remercier sincèrement notre encadreur, l'enseignant : « **Dr. Nouioua Farid** » d'avoir accepté de nous diriger, et pour tous les conseils et informations qu'il nous a fournis. Et pour tous les enseignants du département informatique.

Nous remercions les membres du jury pour cet honneur qu'ils nous font en jugeant cette humble œuvre.

Enfin, nous n'oublions pas nos enseignants qui ont supervisé notre enseignement tout au long du cycle d'études, et un grand merci également à tous ceux qui ont contribué directement ou indirectement à l'aboutissement de ce travail.

Table des matières

Table des figures	8
1 Introduction générale :.....	1
I Chapitre 1 : Généralités sur la ville, la mobilité et le stationnement intelligents :	3
1 Introduction :	3
2 La ville intelligente :.....	3
2.1 Définition de la ville intelligente :.....	3
2.2 Les composants de la ville intelligente :	4
2.3 Les promesses de la ville intelligente :	7
3 La mobilité intelligente :	8
3.1 Définition de la mobilité intelligente :	8
3.2 Les objectifs de la mobilité intelligente :.....	9
4 Le stationnement intelligent (smart parking) :.....	9
4.1 Définition du parking intelligent :.....	9
4.2 Le système de stationnement intelligent :.....	10
4.2.1 Les catégories de système de stationnement intelligent :	10
4.3 Les défis de stationnement intelligent :	11
4.4 Les fonctionnalités de smart parking :.....	12
4.5 Architecture de système de parking intelligent :	13
4.6 Les avantages de stationnement intelligent :	14
4.7 Les inconvénients du stationnement intelligent :.....	15
5 Conclusion :.....	15
II Chapitre 2 : les applications mobiles et les systèmes d'exploitation	17
1 Introduction	17
2 Les applications mobiles :	17
2.1 Définition des applications mobiles	17
2.2 Les différents types d'applications mobiles, leurs avantages et inconvénients :.....	18
2.2.1 Les Applications Natives :.....	18
2.2.2 Applications Web :.....	19
2.2.3 Une application hybride :	21
2.3 Les domaines d'application mobile :	22
3 Les systèmes d'exploitation mobiles OS :	23
3.1 Qu'est-ce qu'un système d'exploitation mobile (OS) ?	23
3.2 Systèmes d'exploitation mobiles les plus populaires :	24
3.3 Rôles du système d'exploitation mobiles :	26

4	Conclusion :	27
III	Chapitre 3 : Analyse des besoins et conception	29
1	Introduction :	29
2	Analyse des besoins :	29
4.1	Spécification des besoins :.....	29
5	Choix de La démarche :	30
6	La conception :	30
6.1	Les diagrammes d'UML :.....	31
6.1.1	Les diagrammes UML structurels :.....	31
6.1.2	Les diagrammes UML comportementaux :	31
7	Modélisation de notre système :	32
7.1	Diagramme de cas d'utilisation :	32
7.1.1	Diagramme de cas d'utilisation globale :	33
7.2	Description des cas d'utilisation :.....	34
7.2.1	Cas d'utilisation « Création d' un compte ».....	35
7.2.2	Cas d'utilisation « Authentification » :.....	36
7.2.3	Cas d'utilisation « Consulter le profil » :	37
7.2.4	Cas d'utilisation « Rechercher un parking » :	38
7.2.5	Cas d'utilisation « Choisir un parking » :	39
7.2.6	Cas d'utilisation « Garer » :	40
7.2.7	Cas d'utilisation « Demande de réservation » :.....	41
7.2.8	Cas d'utilisation « Annuler la réservation » :	42
7.2.9	Cas d'utilisation « Demande de déconnexion » :.....	43
7.3	Diagrammes de séquence :.....	44
7.3.1	Diagramme de séquence « Authentification » :.....	45
7.3.2	Diagramme de séquence « Création d'un compte : l'inscription ».....	46
7.3.3	Diagramme de séquence « Consultation de profil » :.....	47
7.3.4	Diagramme de séquence « Réservation de place » :.....	48
7.3.5	Diagramme de séquence « Annuler la réservation » :.....	49
7.3.6	Diagramme de séquence « Garer » :.....	50
7.4	Diagramme d'activité :.....	51
7.4.1	Les composants de base du diagramme d'activités :.....	51
7.4.2	Les avantage de diagramme d'activité :.....	51
7.4.3	Diagramme d'activité « authentification » :.....	52
7.5	Diagramme de classe :.....	53
7.5.1	Diagramme de classe global :.....	54
7.6	Le dictionnaire de données :.....	55

7.7	Modèle relationnel :	56
7.8	Présentation de la base de données :	57
8	Conclusion :	57
IV	Chapitre 4 : Implémentation.....	59
1	Introduction :	59
2	Choix de l'architecture de l'application :	59
2.1	Avantages de l'architecture client/serveur :	60
3	Langage de développement et le Framework	61
3.1	Langage Dart :	61
3.2	Flutter Framework :	62
3.3	Framework architecture :	62
4	Les outils de développement :	64
4.1	Firebase.....	64
4.1.1	Les services de Firebase :	64
4.2	Visual Studio Code :	65
5	Présentation de l'application :	66
5.1	La page de l'inscription :	67
5.2	La page de l'authentification :	68
5.3	La page de profil :	69
5.4	La page de la recherche :	70
5.5	La page information de parking :	71
6	Les services que l'application doit fournir :	71
7	Quelques scénarios d'utilisation de l'application :	74
7.1	1 ^{er} scénario :	74
7.2	2 ^{ème} scénario :	75
7.3	3 ^{ème} scénario :	76
8	Conclusion :	77
9	Bibliographe	80

Table des figures

Figure 1: la ville intelligente.	4
Figure 2: les composants de la ville intelligente.	4
Figure 3: la mobilité intelligente dans la ville.	8
Figure 4: un parking intelligent.	9
Figure 5: un stationnement intelligent avec ces capteurs.	10
Figure 6: La circulation dans une ville avec et sans le stationnement intelligent.	11
Figure 7: Architecture de système de parking intelligent.	13
Figure 8: les différents types d'applications mobiles.	18
Figure 9: les applications natives.	19
Figure 10: les interfaces de l'application Zenpark.	22
Figure 11: les interfaces de l'applications de Passport Parking canada.	23
Figure 12: Pourcentages d'usage des différentes plateformes (Février 2013).	23
Figure 13: IOS (operating System).	24
Figure 14: Symbian (Operating System).	24
Figure 15: Windows phone (Operating System).	25
Figure 16: BlackBerry (Operating System).	25
Figure 17: Android OS.	25
Figure 18: les Rôles du système d'exploitation mobiles.	26
Figure 19: diagramme de cas d'utilisation global.	33
Figure 20: Cas d'utilisation (création d'un compte).	35
Figure 21: Cas d'utilisation " Consulter le profil ".	37
Figure 22 : Cas d'utilisation " Recherche un parking ".	38
Figure 23: Cas d'utilisation " choisir un parking ".	39
Figure 24: Cas d'utilisation " garer ".	40
Figure 25 : Cas d'utilisation " Réservation de place ".	41
Figure 26: Cas d'utilisation " Annuler une réservation ".	42
Figure 27: Cas d'utilisation " Déconnexion ".	43
Figure 28 : Diagramme de séquence « Authentification ».	45
Figure 29: Diagramme de séquence « l'inscription ».	46
Figure 30 : Diagramme de séquence « Consultation de profil ».	47
Figure 31: Diagramme de séquence « Réservation de place ».	48
Figure 32 : Diagramme de séquence « Annuler la réservation ».	49
Figure 33 : Diagramme de séquence « Garer ».	50
Figure 34 : Diagramme d'activité « authentification ».	52
Figure 35: Diagramme de classe global.	54
Figure 36 : la base de données.	57
Figure 37: l'architecture de l'application.	59
Figure 38: Logo de langage DART.	61
Figure 39: Logo de Flutter.	63
Figure 40: Logo de Firebase.	64
Figure 41 : les services de Firebase.	64
Figure 42: Logo de Visual Studio Code.	65
Figure 43 : Interface de l'installation de l'application.	66
Figure 44 : la première interface de l'ouverture de l'application.	66
Figure 45: L'interface de la page de l'inscription.	67

Figure 46: l'interface de la page de l'authentification.....	68
Figure 47: l'interface de l'authentification par Google.....	69
Figure 48: l'interface de la page profil.....	69
Figure 49 : l'interface de la page de la recherche.....	70
Figure 50: l'interface de la page d'information de parking.....	71
Figure 51: les services de l'application.....	71
Figure 52: l'interface de réservation d'une place.....	72
Figure 53: l'interface de l'opération Garer.....	73
Figure 54 : l'interface de l'annulation d'une réservation.....	73
Figure 55: 1 ^{er} scénario d'utilisation de l'application.....	74
Figure 56: 2 ^{ème} scénario de l'application "utilisateur1".....	75
Figure 57: l'alerte de "Réservation terminer".....	75
Figure 58: 2 ^{ème} scénario "L'utilisateur 2".....	76
Figure 59 : 3 ^{ème} scénario "L'utilisateur 1".....	76
Figure 60 : 3 ^{ème} scénario "L'utilisateur 2".....	77

Liste des abréviations :

TIC : technologies d'information et de la communication.

STI : système de transport intelligent.

NTIC : nouvelles technologies d'information et de communication.

IOT : interconnexion entre l'Internet et les objets (Internet of Things)

WSN : réseau de capteurs sans fil (Wireless Sensor Network)

U.M.L : langage de modélisation unifié.

OMT : Technique de modélisation d'objets (Object-modeling technique).

OMG : Object management group.

OS : system exploitation mobile.

URL : localisateur uniforme de ressource (Uniform Resource Locator).

RIM : Recherche en mouvement (Research In Motion).

APIs : Interface de programmation d'applications.

RAM : mémoire physique vivre.

IOS : system exploitation propriétaire d'Apple.

Introduction générale

1 Introduction générale :

Contexte :

Accompagnant la forte croissance démographique mondiale, le nombre de personnes habitant dans les villes augmente également. Cette croissance crée de nombreux problèmes pour les villes, donc elles doivent s'adapter à ce phénomène en augmentant leur capacité d'accueil et leurs services.

Comme l'affirment Attour et Rallet : les villes doivent « continuer à se développer tout en limitant les effets négatifs de leur croissance sur les coûts d'agglomération et le bien être des habitants » [1].

Il existe également une tendance croissante dans les applications de la ville intelligente qui peuvent aider à améliorer pour réduire les problèmes des villes. Dans les villes intelligentes nous trouvons beaucoup de difficultés et des obstacles à surmonter comme la congestion causée par le grand nombre de véhicules qui augmente chaque jour dans tout le monde. L'un des plus importants facteurs pour régler cette situation c'est le parking.

Le parking devrait offrir aux conducteurs suffisamment d'espace pour garer leur voiture, car la voiture joue un rôle énorme dans le transport et la mobilité de la plupart des personnes pour se rendre à leurs lieux de destination.

Problématique :

Nous observons ces dernières années une accélération du développement des nouvelles technologies. Ceci a donné naissance à une nouvelle notion appelé « stationnement intelligent », « smart parking » ou « parking intelligent ».

La question que l'on peut se poser est : **pourquoi le smart parking ?**

Le stationnement intelligent aide à résoudre l'un des plus gros problèmes de conduite dans les zones urbaines :

- Le problème principal est de trouver une place vide de parking facilement : il y a un peu de place.
- Un problème de pollution se pose également à cause des conducteurs qui tournent dans la ville pour chercher une place.
- Le temps de recherche : perdre beaucoup de temps pour trouver une place.
- Un problème de sécurité : manque de sécurité.

D'autres problèmes comme retrouver une place dans un parking souterrain, un parking sombre, des voitures identiques, etc.

Objectifs :

Dans la ville intelligente nous avons besoin d'un nouveau système, un système qui peut aider à gérer et réduire la circulation routière.

L'objectif est de proposer un système qui facilite de trouver une place vide de parking et qui aide les clients à gagner du temps. Pour cela on veut développer une application mobile d'un parking intelligent pour le but d'améliorer et de faciliter le stationnement, et

pour résoudre ou diminuer les problèmes rencontrés au stationnement des voitures dans les parkings.

Cette application permet aux utilisateurs de trouver rapidement une place libre et de réserver, gérer leur stationnement à tout moment.

Contribution :

Dans le but de répondre à l'objectif de notre projet, notre contribution consiste à développer une application mobile d'un parking intelligent.

Pour la réalisation de cette application, nous avons suivi une démarche qui consiste à étudier l'existant, concevoir (l'analyse et la conception) et modéliser l'application et ensuite l'implémenter.

Plan de travail :

Dans notre travail nous avons commencé par une introduction générale, suivie de quatre chapitres :

- **Le premier chapitre :** fait l'objet d'une généralité sur la ville intelligente et la mobilité intelligente. Nous nous sommes concentrées en particulier sur le cas du parking intelligent.
- **Le deuxième chapitre :** concerne les applications mobiles et leurs différents types, les systèmes d'exploitation mobiles.
- **Le troisième chapitre :** concerne l'analyse et la conception générale : dans ce chapitre nous avons modélisé l'application à l'aide du langage UML. Nous avons présenté différents diagrammes qui modélisent notre application.
- **Le dernier chapitre :** comprend la partie pratique de notre travail. Nous avons présenté dans ce chapitre les outils et les langages utilisés dans le développement de l'application et nous avons ensuite décrits le déroulement de notre application en pratique.

Enfin nous terminons notre mémoire avec une conclusion qui résume notre travail et montre ses limites puis donne des perspectives pour des travaux futurs.

Chapitre1 :
**Généralités sur la ville, la
mobilité et le stationnement
intelligents**

I Chapitre 1 : Généralités sur la ville, la mobilité et le stationnement intelligents :

1 Introduction :

Dans le contexte des conditions technologiques avancées, les parties concernées cherchent à améliorer les conditions de vie des citoyens. La recherche vise à apporter confort, facilité de mouvement et rapidité dans les besoins de consommation comme aller au travail et faire les courses, etc. Donc, différents moyens de transport sont nécessaires à l'heure actuelle, mais avec l'augmentation de ces moyens, les goulets d'étranglement augmentent les problèmes de circulation qui sont à l'origine de conséquences économiques, sociales et environnementales malheureuses telles que la pollution atmosphérique, les émissions de gaz à effet de serre, le bruit et les accidents de la route. La congestion du trafic causée par la voiture est un problème gênant à l'échelle mondiale et a considérablement augmenté. Le problème du stationnement est un contributeur majeur, et c'est toujours un gros problème avec l'augmentation de la taille de la voiture dans le secteur de la voiture luxe et des espaces de stationnement confinés dans les villes. Trouver une place de stationnement est une activité courante (et souvent frustrante) pour de nombreuses personnes dans les villes du monde entier. Cette recherche brûle environ un million de barils de pétrole mondial chaque jour [2]. Alors que la population mondiale continue de s'urbaniser, sans une retraite bien planifiée et confortable de la voiture, ces problèmes vont empirer.

Le parking intelligent est donc une passerelle vers une ville intelligente, il aide à résoudre de nombreux problèmes et besoins tels que la facilité de circulation, l'accès au travail, la réduction de la pollution et aide le conducteur à trouver un espace libre avec la possibilité de réserver via un téléphone mobile

2 La ville intelligente :

2.1 Définition de la ville intelligente :

Une ville intelligente est une zone urbaine qui utilise différents capteurs de collecte de données électroniques pour fournir des informations permettant de gérer efficacement les ressources et les actifs. Il s'agit d'une ville qui investit dans son capital humain et social, et privilégiant les technologies de l'information et de la communication (T.I.C.) afin de répondre aux problèmes publics, de réaliser un développement durable et favoriser une meilleure interaction avec ses citoyens et garantir à ses habitants d'amélioration de la qualité de vie. Chaque ville numérique n'est pas nécessairement intelligente, mais chaque ville intelligente a des composants numériques. [2]

Les termes pour désigner la ville intelligente sont nombreux : **smart city, ville numérique, green city, connected city, éco-cité, ville durable.**



Figure 1 : la ville intelligente.

2.2 Les composants de la ville intelligente :

La ville intelligente ou la ville numérique cherche toujours à répondre aux enjeux socio-économiques et environnementaux en utilisant les technologies de l'information et de la communication. Elle a pour but de diminuer la consommation énergétique et de faciliter la vie des habitants. [3]

La ville intelligente selon Rudolf Giffinger [4] :

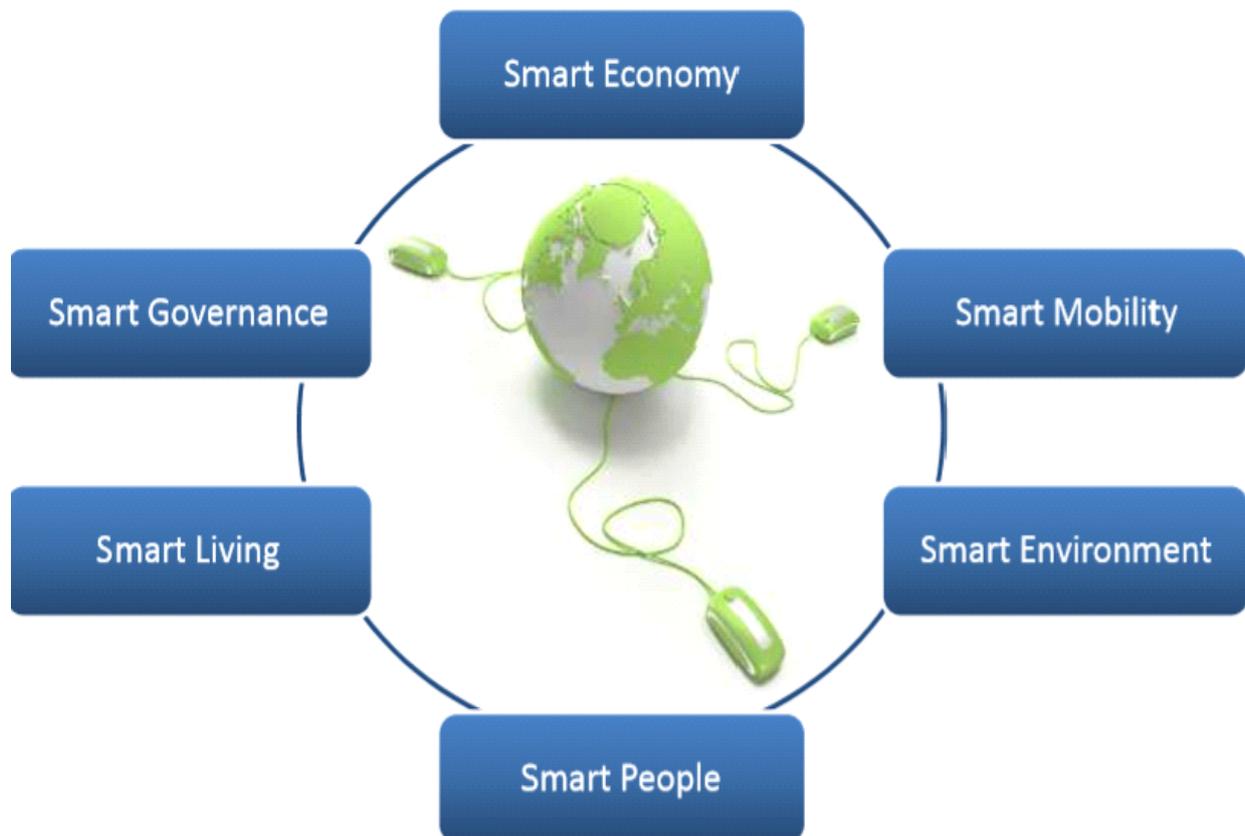


Figure 2 : les composants de la ville intelligente.

Chapitre 1 : Généralités sur la ville, la mobilité et le stationnement intelligents

- **L'économie intelligente :**

C'est un ensemble d'actions, de concepts, d'outils et de de moyens organisés en système de gestion de la connaissance qui produit de l'information pour l'aide à la prise de décisions stratégiques.

Définition de Philippe Baumard : "L'intelligence économique n'est plus seulement un art d'observation mais une pratique offensive et défensive de l'information. Son objet est de relier plusieurs domaines entre eux pour servir à des objectifs tactiques et stratégiques de l'entreprise. Elle est un outil de connexion entre l'action et le savoir de l'entreprise." [5]

- **La mobilité intelligente :**

La mobilité intelligente est une expression désignant généralement des nouvelles technologies au domaine du transport. On parle également de système de transport intelligent (S.T.I) pour un but à atteindre au sien d'une ville intelligente averse de solution permettant de faciliter les déplacements des hommes et marchandises et se heurtent à de nouvelles contraintes, en particulier budgétaires et environnementales.

- **L'environnement intelligent :**

La ville intelligente se concentre sur des domaines pertinents : **gestion d'eau, gestion de déchets et l'énergie** pour une vie saine.

Concernant la gestion de déchets, les villes auront pour une mission de mettre en place des systèmes efficace de récupération, de recyclage et de valorisation des déchets.

Concernant l'énergie, les villes devront renforcer leur action en matière d'efficacité énergétique et devront mettre en place des systèmes de production locale d'énergie.

La gestion intelligente de l'eau consiste à s'appuyer sur le savoir et la participation de toutes les parties prenantes de la gestion de l'eau, ainsi que sur diverses ressources et infrastructures des TIC, afin de mettre en place un système de gestion de l'eau qui exploite les écosystèmes aquatiques sans compromettre la viabilité sociale, économique ou environnementale.

- **Des habitants intelligents :**

Le citoyen est le cœur battant de la ville, c'est l'acteur clé de la Smart City. Les habitants doivent être impliqués dans les transformations de la ville. Ils peuvent alors donner leurs avis et soumettre leurs idées pour devenir de réelles parties prenantes. Ils savent utiliser les nouvelles technologies pour gérer leur temps, leurs modes de consommation, optimiser leurs déplacements, réduire leur impact sur l'environnement et communiquer entre eux ainsi qu'avec l'extérieur. Un effort d'éducation doit être fait pour accompagner les citoyens à l'utilisation des nouvelles technologies dans la vie quotidienne, y compris pour leur relation à l'administration ou encore la gestion des ressources.

Chapitre 1 : Généralités sur la ville, la mobilité et le stationnement intelligents

- **Un mode de vie intelligent :**

La ville intelligente ouvre la voie à un mode de vie productif et écologique. Elle accorde une attention toute particulière à la création d'un espace public de qualité propice à la cohésion social des habitants et d'améliorer les conditions de vie avec l'accès à la culture et l'enseignement et aussi favoriser la santé, la sécurité et la qualité de logement pour vivre de façon saine dans un environnement à la fois dynamique harmonieux et riche.

- **L'administration intelligente :**

Une ville intelligente est d'abord une administration intelligente où la gestion et la gouvernance sont facilitées grâce aux nouvelles technologies de l'information et de la communication (N.T.I.C).

Ces technologies doivent être utilisées dans le cadre d'une stratégie globale dont l'objectif est de répondre aux besoins des citoyens pour améliorer leur qualité de vie. Les TIC sont des outils qui facilitent la gestion et la prise de décisions publiques.

[6]

Smart Economie (Compétition)	Smart People (Capital humain et social)	Smart Gouvernance (Participation)
<ul style="list-style-type: none"> • Esprit novateur • Entrepreneuriat • Image économique et marque • Productivité • Flexibilité du marché du travail • Ancrage international • Capacité de transformer. 	<ul style="list-style-type: none"> • Niveau de qualification • Apprentissage tout au long de la vie • Pluralité sociale et ethnique • Flexibilité • Créativité • Cosmopolitisme • Participation à la vie publique. 	<ul style="list-style-type: none"> • Participation dans la prise de décision • Services publics et sociaux • Transparence Gouvernementale • Stratégie et perspectives politiques
Smart Mobility (Transports et TIC)	Smart Environment (Ressources naturelles)	Smart Living (Qualité de vie)
<ul style="list-style-type: none"> • Accessibilité locale • Accessibilité (internationale) • Disponibilité de l'infrastructure--TIC • Systèmes de transport durables, innovateurs et sécurisés. 	<ul style="list-style-type: none"> • Attractivité des conditions naturelles • Pollution • Protection environnementale • Gestion des ressources durables 	<ul style="list-style-type: none"> • Équipements culturels • Conditions sanitaires • Qualité et sécurité des logements • Équipements pour l'éducation • Attractivité touristique • Cohésion sociale

Tableau 1: les six dimensions pour devenir une ville intelligente.

2.3 Les promesses de la ville intelligente :

La ville intelligente constituerait le rempart au chaos. La ville deviendrait plus efficace car rationalisée par l'alimentation en données qui permettent de contrôler la ville, la rendre plus contrôlable [7]

- **Ville efficace, rationalisé, et contrôlable :**

Une ville axée sur les données est une ville plus efficace car rationalisée par l'alimentation et l'utilisation intelligente des données pour mieux fournir des services essentiels. Transparence, open data, innovation, visibilité et créativité sont tous des éléments importants pour l'identité civique moderne qui permettent de contrôler la ville, et la rendre plus contrôlable.

- **Ville durable** (sustainable city en anglais) :

La smart city est avant tout une ville numérique qui repose sur les nouvelles TIC. Ceci permet la fusion entre numérique et environnement avec l'élimination progressive des erreurs de gouvernance et des mauvais comportements humains, dus à l'infrastructure de données afin d'atteindre l'objectif d'implémenter des énergies renouvelables au cœur des villes et développer des réseaux intelligents d'énergie en proposant de nouvelles formes d'architecture pour les réseaux de distribution d'électricité notamment (construire des bâtiments à très faible consommation énergétique qui s'intègrent au mieux dans les environnement naturel).

La ville intelligent permettrait indirectement aux habitants d'adopter des comportements plus favorables à l'environnement (rendre la ville neutre sans impact négatif sur l'environnement) grâce aux données et à faire passer les sociétés urbaines à une transition énergétique.

- **Une ville plus stimulante où il fait bon vivre**

La ville intelligente vise à l'amélioration de la qualité de vie pour les citoyens et environnements (optimiser les services publics comme les services culturels, les services de santé, de sécurité publique, contrôler la pollution et la qualité de l'air, réduire les émissions des gaz polluants, assurer une meilleure gestion des ressources de l'environnement fondée sur les principes du développement durable). La ville intelligente vise aussi à améliorer la gouvernance urbaine grâce à l'interopérabilité entre les différents services de la ville, la protection des informations individuelles des citoyens et le développement de la politique urbaine.

3 La mobilité intelligente :

3.1 Définition de la mobilité intelligente :

La mobilité intelligente est une nouvelle façon de se déplacer qui consiste à déplacer des personnes et des biens de manière plus simple, plus efficace et plus respectueuse de l'environnement. Elle utilise de nouvelles idées et de nouvelles technologies pour aller au-delà des approches traditionnelles des transports et des infrastructures, et pour trouver des moyens innovants d'améliorer la mobilité et de rendre les voyages meilleurs et accessibles à tous. Le concept de mobilité intelligente comprend un large éventail de modes de transport : trottinettes, bus, trains légers sur rail, métros, tramways, taxis, véhicules autonomes. De plus, les utilisateurs ont la possibilité de posséder ou de partager ces moyens. La mobilité intelligente créera des solutions qui diffèrent considérablement des méthodes de transport que nous utilisons aujourd'hui. Mais, pour ce faire, elle doit s'appuyer sur l'expertise et les connaissances d'un éventail beaucoup plus large de domaines techniques, sociaux et réglementaires et cela n'a pas été le cas avec les transports traditionnels. [8]



Figure 3 : la mobilité intelligente dans la ville.

3.2 Les objectifs de la mobilité intelligente :

La mobilité intelligente (ou "Smart-Mobility") vise à optimiser nos déplacements urbains, grâce aux nouveaux moyens informatiques et techniques. On peut résumer les objectifs de la mobilité intelligente en 6 catégories :

- Réduire la pollution environnementale.
- Réduire la congestion.
- Augmenter la sécurité des transports.
- Réduire la pollution sonore.
- Augmenter les vitesses de déplacement.
- Réduire les coûts de déplacement.

4 Le stationnement intelligent (smart parking) :

4.1 Définition du parking intelligent :

Le parking intelligent est l'une des initiatives de ville intelligente les plus populaires à l'heure actuelle. Il s'appuie sur les données recueillies à partir d'une variété de capteurs pour mieux gérer les services, les ressources et les actifs de la ville afin d'améliorer la qualité de vie des citoyens. La technologie de stationnement intelligent répond à l'une des plus grandes frustrations des citoyens car elle aide les conducteurs à trouver des places de stationnement facilement et rapidement grâce à l'application. Elle permet aussi le transfert des informations aux responsables du stationnement pour identifier les infractions et d'inciter les gens à utiliser d'autres options de transport en cas de stationnement encombré. Les initiatives de stationnement intelligent sont souvent combinées à la technologie des rues intelligentes telles que les feux de circulation intelligents et les lampadaires intelligents qui améliorent la sécurité, réduisent les temps de transport et réduisent la congestion du trafic.



Figure 4 : un parking intelligent.

4.2 Le système de stationnement intelligent :

Le stationnement intelligent est un système qui est développé avec l'intégration de nouvelles technologies pour résoudre tous les problèmes rencontrés par les clients dans les parkings. Guider l'utilisateur vers une place vide rapidement et facilement est un objectif pour le système qui ne vise pas seulement à trouver une solution mais représente également un outil pour gagner du temps et ainsi limiter la pollution. Fluidifier la recherche d'une place implique une réduction du trafic.

Les capteurs sont installés pour identifier les places disponibles et occupées. Ils s'installent directement dans la chaussée, au centre de chaque place de stationnement.



Figure 5 : un stationnement intelligent avec ces capteurs.

4.2.1 Les catégories de système de stationnement intelligent :

4.2.1.1 Système de guidage et d'information sur le stationnement :

Le système de guidage est intégré au système de stationnement intelligent et il consiste à aider, orienter et informer les conducteurs des véhicules pour trouver des places et des services disponibles, afin d'accéder aux systèmes d'information clés pour le stationnement dans le parking connecté.

4.2.1.2 Système de paiement intelligent :

Le système de paiement a été mis en place pour le but de faciliter et de limiter les problèmes de paiement conventionnel en réorganisant le mode de paiement pour le stationnement en introduisant des nouvelles technologies et techniques.

Cela évite les nombreux problèmes auxquels le client est confronté lorsqu'il utilise le paiement avec les méthodes conventionnelles qui entraînent des perturbations ou des retards, en plus de la congestion à l'entrée de la station (parking) s'il y a plusieurs véhicules en même temps.

4.2.1.3 Système d'information basé sur le transport en commun :

C'est un système qui aide la ville intelligente à mettre en œuvre une politique intégrée de transport durable propre et efficace. Ce système d'information qui permet de faciliter l'accès au réseau de transport en commun pour chaque citoyen qui utilise ce mode de transport. Il fournit des informations corrects et fiables en temps réel sur l'état de chaque stationnement tels que les horaires et les conditions de circulation pour le public.

4.3 Les défis de stationnement intelligent :

La ville intelligente utilise les nouvelles technologies pour offrir des solutions pratiques aux utilisateurs, telles que le stationnement connecté qu'il devrait relever plusieurs défis :

- **Diminuer la pollution.**
- **Améliorer le confort des usagers** : pour obtenir des clients permanents et fidèles.
- **Fluidité de trafic** : minimiser la congestion de trafic.
- **Sécurité** : la surveillance des véhicules et des conducteurs contre le vol et les squatters.
- **Optimisation de l'occupation** : mieux gérer le taux d'occupation des places de stationnement.
- **Fournir un service de guidage pour les touristes** : Le stationnement intelligent aide tout étranger à la ville à obtenir une place pour sa voiture facilement.
- Le système d'information sur la disponibilité du stationnement et le système de réservation de stationnement devraient fournir des services avancés de navigation.
- Fournir des fonctionnalités intéressantes pour faciliter la tâche des administrateurs et des gestionnaires dans la gestion du stationnement.

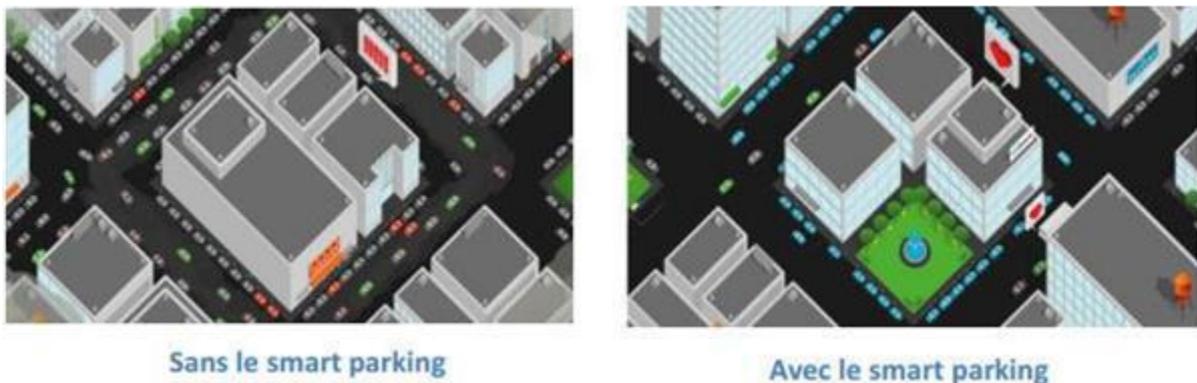


Figure 6 : La circulation dans une ville avec et sans le stationnement intelligent.

4.4 Les fonctionnalités de smart parking :

Les différentes parties du système d'un smart parking :

- **Points de surveillance et de contrôle** : ils sont installés aux portes d'entrée et de sortie d'un site et à chaque endroit désigné pour attendre de révéler son état d'occupation, puis tous les points envoient des informations aux points de transmission, et des ordres sont reçus des centres de contrôle pour mettre en œuvre certaines procédures.
- **Points de transmission** : Ils sont également installés dans la zone d'attente, reçoivent les données des points de contrôle et les transfèrent au point de collecte. Les commandes vont également du centre d'information à tous les points de surveillance.
- **Point de montage** : à installer dans la salle d'observation. Il collecte les informations des points de transmission dans les zones d'attente mixtes et les connecte au centre d'information et de gestion, c'est la passerelle vers le centre de contrôle
- **Le centre de contrôle ou centre d'information**, responsable de la gestion et de la maintenance, ce centre est installé dans la salle de contrôle. Le centre reçoit des informations du réseau de capteurs, et s'occupe du calcul du nombre de places de stationnement occupées/vides nécessaires, ainsi que du contrôle de la transmission et l'affichage des informations sur les écrans. Ce processus est réeffectué lorsqu'une voiture se gare ou laisse sa place de stationnement.
- **Centre de diffusion de l'information** :
 - Il affiche toutes les informations pour informer les automobilistes sur le nombre de places disponibles sur un site A et les différents rôles sur le site, pour les aider à prendre des décisions et déterminer leur direction à tout moment. Parmi les techniques de présentation des informations les plus importantes, on a :
 - **Signes de message variables** :
Ils sont placés dans divers emplacements routiers ou latéraux et à l'entrée du parking.
 - **Tableaux d'affichage numérique** :
Il est placé à l'intérieur du site dans les zones d'attente. Chaque rôle du site étant de maintenir le trafic à l'intérieur du site.
 - **Téléphones portables** :
Il est également utilisé pour identifier l'état des emplacements des véhicules pour bien aider les conducteurs. Les informations affichées par le système permettent de bien guider les automobilistes à trouver des places de parking appropriées.

4.5 Architecture de système de parking intelligent :

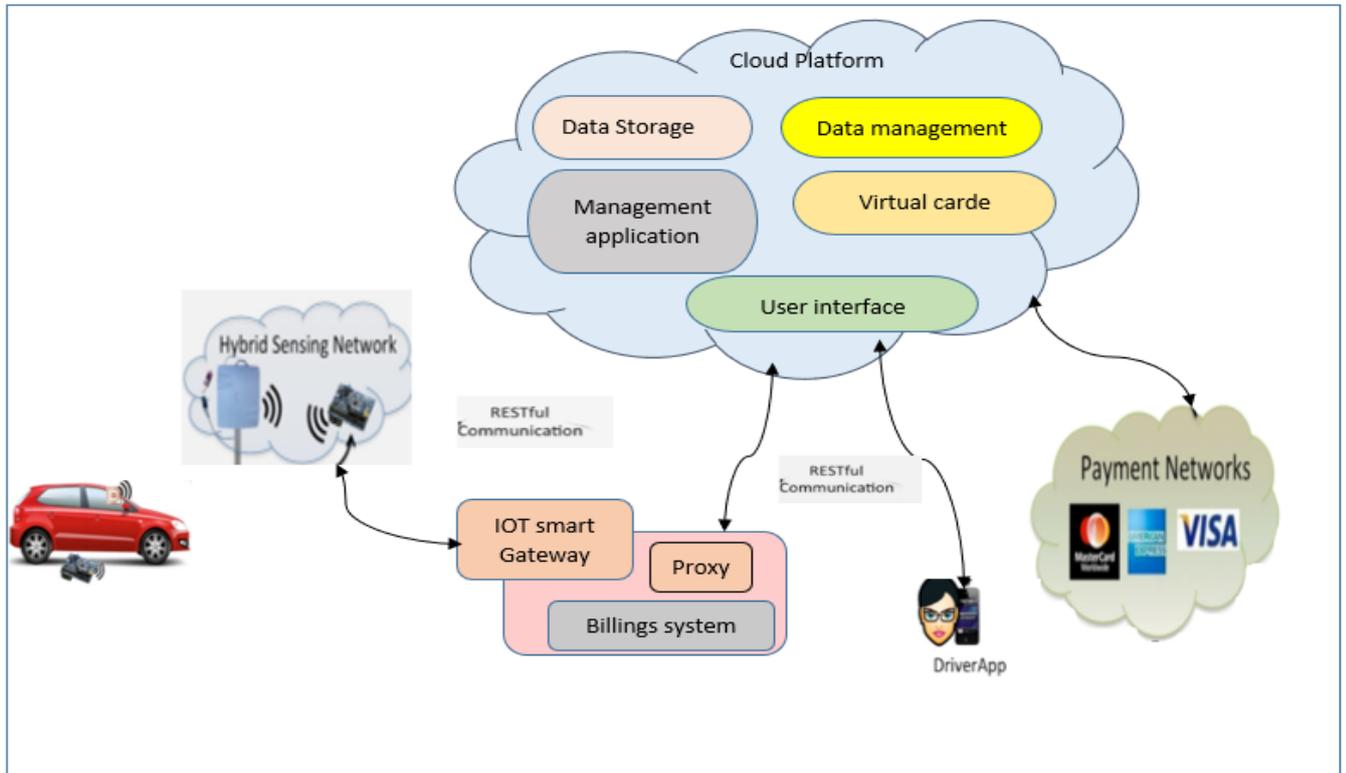


Figure 7 : Architecture de système de parking intelligent.

Au départ, l'utilisateur doit enregistrer ses détails dans l'application mobile qui les stocke sur le serveur. Une fois les identifiants enregistrés et qu'il est autorisé à accéder à son compte et à réserver une place pour son parking, le véhicule est autorisé à entrer dans le parking et le numéro de trappe est donné à partir de l'emplacement disponible le plus proche. Une fois que la voiture est arrêtée avec succès dans la trappe, son temps de stationnement commence. Ainsi, une fois que le chauffeur est revenu à la voiture et qu'il a récupéré sa voiture, le temps de stationnement peut être interrompu et le reçu est envoyé sur son téléphone portable. Il peut ensuite payer en ligne avec son portefeuille électronique et quitter Parking. [9]

4.6 Les avantages de stationnement intelligent :

Voici quelques-uns des principaux avantages que le système de stationnement intelligent offre aux conducteurs.

➤ **Moins de pollution, moins de gaspillage de carburant et moins d'empreinte environnementale individuelle**

Les stationnements intelligents contribuent à un environnement plus propre. Les conducteurs sont dirigés directement vers les parkings disponibles. En conséquence, ils perdent moins de kilomètres à tourner en rond à la recherche de places de stationnement vacantes. Cela comprend la réduction du temps nécessaire pour trouver une place de stationnement, ce qui réduira la quantité de carburant utilisée. Cela signifie une réduction de la quantité de dioxyde de carbone (Co2) émise, qui à son tour permet de ne pas nuire directement à la vie humaine.

➤ **Gagner du temps**

Parcourir le minimum possible de kilomètres à la recherche de places de stationnement, permettra de gagner un temps précieux qui peut être consacré au travail, aux loisirs ou au divertissement.

En effet, les capteurs du Smart Parking transmettant des données en direct, les conducteurs auront un aperçu en temps réel de l'occupation de l'espace de stationnement. Cela signifie que les conducteurs peuvent éviter les parkings entièrement occupés et localiser facilement les places de stationnement vacantes ailleurs. Par conséquent, les places libres sont occupées plus rapidement, ce qui réduira le temps d'attente lorsque des places de stationnement sont vivantes.

➤ **L'espace d'une municipalité sera utilisé plus efficacement**

Le stationnement intelligent permet d'avoir moins de véhicules dans les rues à la recherche d'espace de stationnement. En plus, il permet de réduire le nombre de véhicules stationnés illégalement sur le bord de la route, ce qui entraîne des embouteillages. Cela contribuera à réduire la circulation dans les zones où il n'y a pas assez de stationnement. En conséquence, il y aura moins de congestion et les conducteurs bénéficieront d'une circulation plus fluide dans les rues.

➤ **Augmentation de la sécurité.**

Les solutions des parkings intelligentes améliorent la sécurité en diminuant l'incertitude et la pression pour trouver un parking près de la destination visée qui peut conduire au stationnement dans des zones à risque ou mal adaptées au stationnement. En utilisant des parkings intelligents, les capteurs mis au sol dans les parkings informent les conducteurs en temps réels des places de stationnement disponibles, ce qui leur permet de se rendre directement dans un parking ouvert et sécurisé sans aucune pression ou stress.

4.7 Les inconvénients du stationnement intelligent :

Le parking intelligent peut avoir tout de même certains inconvénients comme les suivants :

- Les concepteurs du « stationnement intelligent » avaient aussi oublié un détail : quand on est au volant et qu'on cherche une place, on n'a pas nécessairement le loisir de consulter son smartphone.
- Les difficultés de stationnement engendrent ainsi une densité de trafic plus importante, de nombreux frais pour les usagers(carburant), mais également de la pollution et une nuisance sonore dans les grandes agglomérations.
- Ainsi, le stationnement est un véritable défi à relever pour tout espace urbain.
- Le stationnement intelligent est certainement une invention très coûteuse.

5 Conclusion :

Le stationnement intelligent constitue un pilier de la mobilité intelligente puisqu'il joue un rôle très important pour assurer cette mobilité. En effet, il participe à minimiser la congestion et à éviter de nombreux problèmes dans le but de faciliter la mobilité dans la ville intelligente.

Dans ce chapitre nous avons donné une idée sur la ville et la mobilité intelligentes et nous nous sommes intéressés en particulier au stationnement intelligent qu'il a un rôle important dans l'amélioration des niveaux de services dans la ville de demain.

Chapitre 2 :
**Les applications mobiles et
les systèmes d'exploitation**

II Chapitre 2 : les applications mobiles et les systèmes d'exploitation

1 Introduction

La technologie mobile n'est pas juste un terme à la mode mais un sujet d'actualité au sein de nombreux secteurs d'activité. Il est extrêmement pertinent pour le domaine de développement d'applications.

De nos jours, les technologies mobiles prennent de plus en plus de place sur le marché et dans le paysage numérique où les projets des applications mobiles sont devenus un moyen essentiel de création de nouveaux services à destination des mobinautes. Les Smartphones et les tablettes sont considérés comme des petits ordinateurs. Ils deviennent de plus en plus fréquents et cela grâce aux différents systèmes d'exploitation mobiles tel que : « Android », « iOS » ou bien « Windows Phone » qui ne cessent de se développer.

Dans ce chapitre on s'intéresse à la notion d'application mobile.

2 Les applications mobiles :

2.1 Définition des applications mobiles

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil électronique mobile, comme un Smartphone, une tablette ou un baladeur numérique.

Une application mobile peut être soit installée directement sur l'appareil dès sa fabrication en usine, soit téléchargée depuis un magasin d'applications dit « application store » telle que Google Play, l'App Store ou encore le Windows Phone Store. Une partie des applications disponibles sont gratuites tandis que d'autres sont payantes.

Le téléchargement de l'application mobile se fait suivant deux options :

- Sur téléphone par le biais de connexion internet.
- Sur pc en le branchant avec le téléphone mobile.

Chez les mobinautes, l'application mobile est similaire à un site internet pointu en raison de sa connexion à internet. De plus l'interface de site et de l'application mobile s'avèrent identiques sauf que l'application demeure fondamentalement définie comme un logiciel. En ce sens, les applications mobiles se regroupent en plusieurs séries suivant des critères basiques [10] :

- Applications fonctionnant sans internet : Appelées applications indépendantes, ce sont des applications qui fonctionnent sans avoir besoin de connexion internet ou téléphonique. On peut citer les applications : liste de contacts, calculatrice entre autres.

Chapitre 2 : les applications mobiles et les systèmes d'exploitation.

- Applications exigeant une connexion : Contrairement aux applications indépendantes, ces applications doivent avoir accès à l'internet pour fonctionner.

2.2 Les différents types d'applications mobiles, leurs avantages et inconvénients :

Actuellement, il y a principalement trois types d'applications mobiles [11] :



Figure 8 : les différents types d'applications mobiles.

2.2.1 Les Applications Natives :

Une application native est une application créée individuellement pour un système d'exploitation mobile spécifique (par exemple : Android, Windows phone ou encore iOS) et ne fonctionne que sur ce système. Elle est développée par le langage de programmation native propre au système tel que Java pour Android ou le langage Objective-C. Les applications natives sont téléchargées à partir d'une plateforme de téléchargement qui est souvent un Store applicatif. C'est par exemple le cas pour l'Apple store ou encore Google Play.

Le développement de l'application native nécessite le recours à la mémoire du smartphone sans omettre les options liées au système d'exploitation en question. De cette façon, le résultat se résume dans l'aboutissement à des applications mobiles avec des fonctions plus professionnelles, développées et performantes au même niveau que les applications en HTML5/CSS3 et les applications hybrides.

2.2.1.1 Les avantages de ce type d'applications :

- Une expérience utilisateur optimale
- Un accès aux fonctionnalités natives du téléphone
- Moins de bugs pendant le développement et plus de sécurité
- Ces applications sont capables d'utiliser l'ensemble des fonctionnalités du mobile et peuvent être utilisées sans avoir accès à Internet. L'installation est gratuite, avec en plus, la possibilité d'avoir accès à des options supplémentaires, mais qui sont payantes.

2.2.1.2 Les inconvénients :

- Les applications natives prennent du temps à être développées et sont relativement coûteuses.
- Problèmes de rétrocompatibilité, en raison notamment de la création de nouvelles versions du système d'exploitation.
- Celles-ci peuvent aussi se voir refuser l'accès au Store sur lesquelles elles souhaitent être mises à disposition. En effet, elles doivent passer par une batterie de tests de validation de ces Stores, qui peut les accepter ou les refuser.

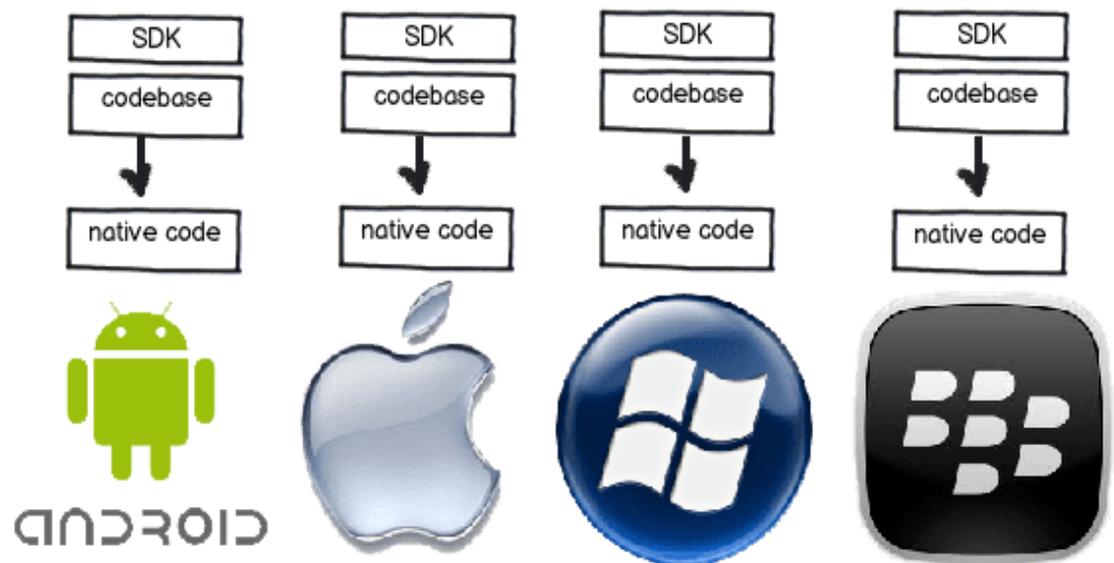


Figure 9 : les applications natives.

2.2.2 Applications Web :

Une application Web est une application conçue avec HTML, CSS ou JavaScript qui s'exécute sur un navigateur Web de smartphone.

Une connexion Internet solide est essentielle au comportement et à l'expérience utilisateur de ce groupe d'applications. Il est donc possible d'accéder à autant de fonctions que possible sur un téléphone mobile. Par exemple, cela permet d'accéder à diverses fonctions, telles que la géolocalisation, l'accéléromètre, la gestion des fonctions multitouche ou encore la synchronisation hors ligne lorsque le téléphone mobile perd puis retrouve sa connexion.

Ces applications Web peuvent être de deux types :

- Les applications génériques compatibles avec toutes les plateformes mobiles et donc utilisables sur n'importe laquelle de ces plateformes,
- Celles spécifiquement conçues pour un type de média spécifique.

Chapitre 2 : les applications mobiles et les systèmes d'exploitation.

2.2.2.1 Les avantages.

- L'application Web utilise des langages bien connus des programmeurs tels que : HTML5 et CSS.
- Les applications web fonctionnent à l'intérieur du navigateur web de l'appareil via un simple URL.
- Elles n'ont pas besoin d'être installées à partir des App Store, comme Google Play Store ou Apple App Store. Cela signifie des économies, puisqu'il est gratuit d'avoir le lien direct via une application Web.
- Contrairement à l'application native, une seule application Web atteindra tous les périphériques. Elle est programmée pour être reproduite dans n'importe quel système d'exploitation. Elle s'adapte facilement à IOS, Android ou Windows Phone entre autres.
- De plus, les Web App sont simples à développer sur ces plateformes, puisque celles-ci ne les soumettent pas au test de validation.
- Leur code unifié permet la comptabilité avec tous les navigateurs, cela permet aux applications Web d'être développées plus rapidement et donc d'obtenir une réduction conséquente des coûts du projet.

2.2.2.2 Les inconvénients :

- Une seule application Web atteindra tous les périphériques. Cependant, le site Web doit être programmé pour être lu sur n'importe quel système d'exploitation. S'il n'est pas adapté, vous pouvez avoir des problèmes pour l'ouvrir sous IOS, Android ou Windows Phone.
- Elles n'ont pas accès à toutes les fonctions présentes sur le mobile. Par exemple, il leur est impossible d'accéder au répertoire du mobile
- On a nécessairement besoin d'une connexion internet pour sa reproduction. Sinon, on ne peut pas naviguer sur le Web et les applications Web ne seront d'aucune utilité.

Les types d'applications Web les plus pertinents :

Cette classification a été faite en fonction de la façon dont les applications Web sont représentées avec le contenu qu'elles doivent afficher. Nous distinguons six types d'applications Web :

- Application Web statique
- Application Web dynamique
- Boutique virtuelle ou e-commerce
- Portail d'applications Web
- Application web animée
- Application Web avec 'gestionnaire de contenu'

2.2.3 Une application hybride :

Est un mix, entre les applications Web et les applications natives. Ce type d'application utilisent le navigateur web intégré du support (Smartphone ou tablette) et les technologies Web (HTML, CSS et JavaScript) pour fonctionner sur différents OS (iOS, Android, Windows Phone, etc.). Une telle application utilise les fonctionnalités natives des Smartphones et peut être distribuée sur les plateformes d'applications telles que l'AppStore, le Google Play, etc. Ainsi, une application dite hybride, contrairement à une application native, n'est pas dépendante d'une plateforme mobile en particulier. De la même manière, et contrairement aux applications Web, les applications hybrides peuvent accéder à toutes les fonctions présentes sur le mobile. Cela est rendu possible par des liens qui sont faits entre le langage natif et la technologie Web présente dans l'application hybride.

2.2.3.1 Les avantages

Les applications hybrides présentent un certain nombre d'avantages :

- Allier l'expérience utilisateur avec l'agilité du cycle de développement et des coûts maîtrisés.
- Réutiliser le code de la partie application web : le code est écrit une fois et déployé ensuite sur toutes les plateformes mobiles.
- Réduire les temps de développement et les coûts : le code étant écrit une fois, cela réduit considérablement les temps de développement et les coûts par rapport aux applications natives nécessitant d'une part un développement pour iOS et d'autre part un développement pour Android.

2.2.3.2 Les limitations des applications hybrides

- La difficulté ou l'impossibilité d'exploiter toutes les capacités des plateformes : chaque plateforme, iOS ou Android, a des capacités uniques qu'un développeur peut souhaiter exploiter. Si tel est le cas, c'est en utilisant une combinaison de plugins et de code spécifique à la plateforme que cela pourra être réalisé pour une application hybride. Cela complexifie d'autant le projet de développement de l'application.
- Des performances et des transitions entre les pages plus lentes.
- Une dépendance vis à vis de la vitesse du navigateur.

2.3 Les domaines d'application mobile :

Plusieurs domaines d'application sont touchés par les applications mobiles. Parmi ces principaux domaines, nous citons : le domaine de la sécurité, le domaine du transport, l'environnement et l'infrastructure et les services publics. etc.

Nous présentons **quelques exemples d'applications mobiles** dans le domaine de transports intelligent (parking intelligents) :

I. Zenpark :

Créé en 2011, « Zenpark » est le premier opérateur de parkings partagés automatisés d'Europe. « Zenpark » propose aux automobilistes qui sont à la recherche d'un stationnement, des places de parking usuellement non disponibles. « Zenpark » a pour objectif de déployer un réseau de parking le plus dense possible, pour qu'automobilistes, riverains et collectivités locales puissent mieux vivre ensemble. [12]

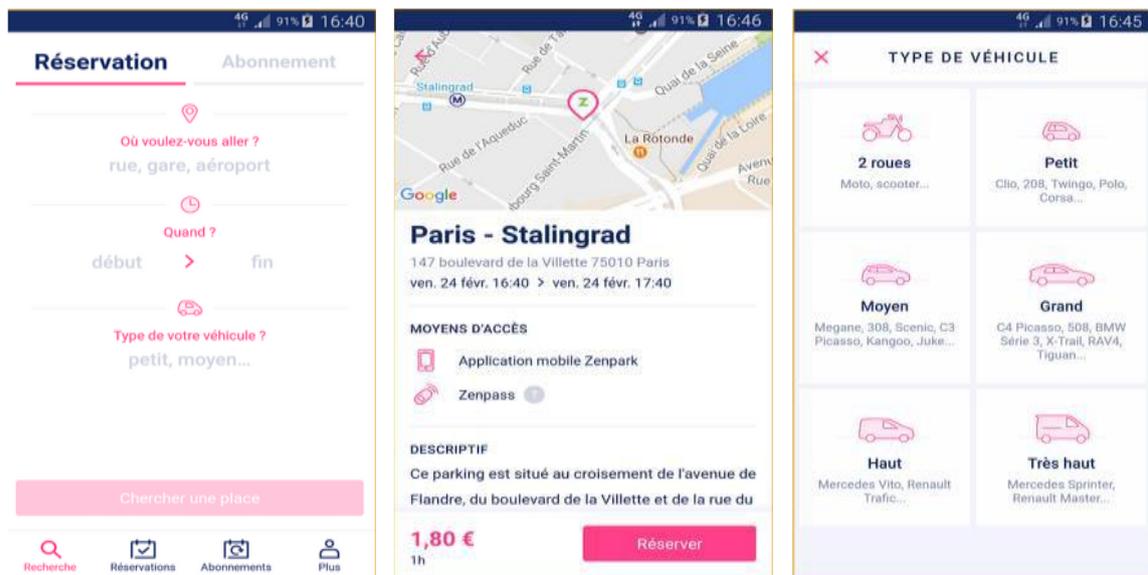


Figure 101: les interfaces de l'application Zenpark.

II. Passport parking canada :

Passport Parking, LLC (Limited liability company) a été créée en 2010 et est le premier fournisseur de solutions de stationnement en nuage. Il offre le moyen le plus simple de payer le stationnement à l'aide de votre téléphone mobile. C'est le premier fournisseur de services de paiement mobile à proposer une gamme complète de solutions de paiement mobile. L'enregistrement prend 2 minutes et la pause se termine en quelques secondes. En plus, cette application offre des options de paiement vocales et textuelles. Une fois inscrit, vous pouvez utiliser l'un de ces systèmes de manière interchangeable. On trouve également la possibilité de se connecter à leur site sécurisé pour accéder aux frais de stationnement, à l'historique, aux informations de profil et aux rapports d'impression. [13]

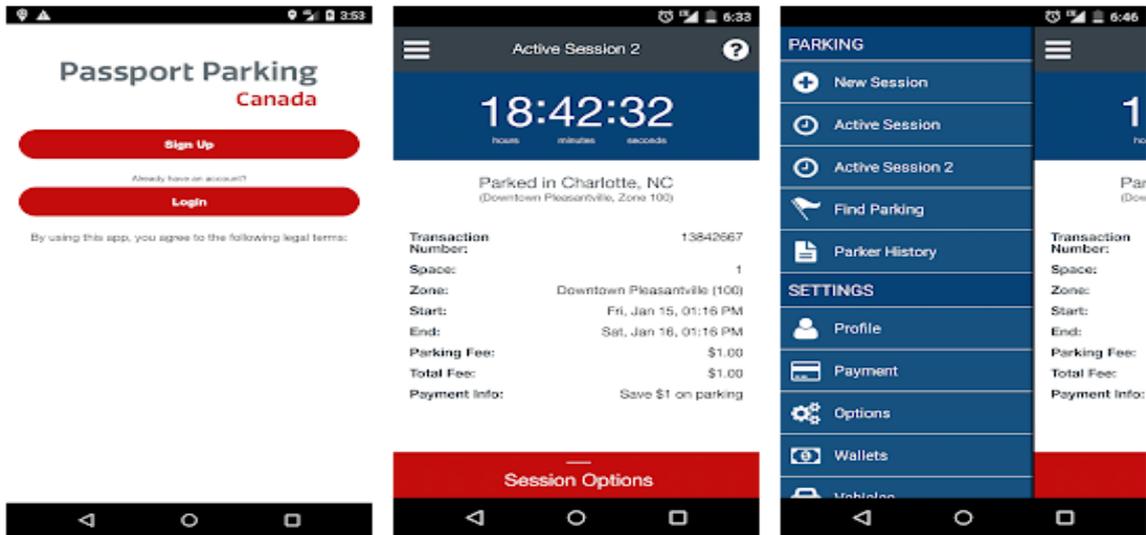


Figure 2 : les interfaces de l'application de Passport Parking Canada.

3 Les systèmes d'exploitation mobiles OS :

Au jour d'aujourd'hui, les applications mobiles continuent d'évoluer et la demande de ces applications se multiplie. Dans le cadre de développement des applications, plusieurs options de plateformes mobiles existent.

Les systèmes d'exploitation mobiles (OS) les plus utilisées sont : **IOS** (qui appartient à Apple : iPhone et iPad), **Android** (Google) et **Windows de Microsoft** [14]

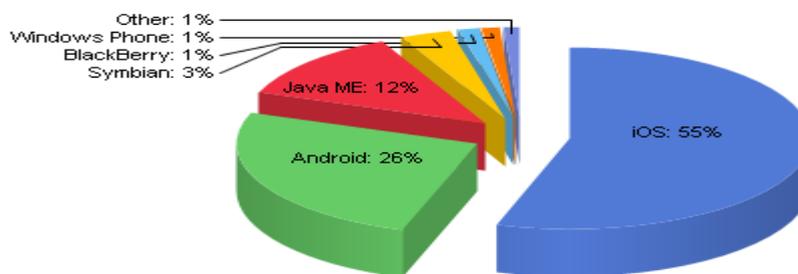


Figure 3: Pourcentages d'usage des différentes plateformes (Février 2013).

3.1 Qu'est-ce qu'un système d'exploitation mobile (OS) ?

Tous les téléphones mobiles contiennent une plateforme qui contrôle toutes ses fonctionnalités. Le système d'exploitation mobile OS est un système qui exploite l'appareil mobile (smartphone, tablette, etc.). Il contrôle toutes les opérations de base de téléphone mobile : Bluetooth, Wifi, appareil photo, etc.

Il permet de ce fait aux utilisateurs de pouvoir passer un appel téléphonique, utiliser le lecteur de musique et d'autres fonctionnalités.

3.2 Systèmes d'exploitation mobiles les plus populaires :

Nous allons présenter brièvement des systèmes d'exploitation qui dominent aujourd'hui le monde :

- a. **IOS** : c'est le système d'exploitation propriétaire d'Apple pour ses appareils portables tels que l'iPhone, l'IPad et l'IPod. iOS comporte quatre couches d'abstraction, similaires à celles de MacOS : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ».

iOS propose la compatibilité avec iCloud, le service Cloud d'Apple et un accès direct au catalogue Apple Store : applications, musique, podcasts, émissions de télévision et films.



Figure 4: IOS (operating System).

- b. **Symbian OS** : Symbian est un système d'exploitation visant spécifiquement les téléphones portables qui proposent un niveau maximal d'intégration avec les fonctionnalités de communication et de gestion des informations personnelles (PIM). Il est récemment passé en open source. C'est un système libre, open source qui se base sur un noyau Symbian.



Figure 5: Symbian (Operating System).

Chapitre 2 : les applications mobiles et les systèmes d'exploitation.

- c. **Windows Mobile :** Windows est le système d'exploitation développé par Microsoft. Il fonctionne uniquement sous « Windows », Ce système d'exploitation mobile se base sur le noyau Windows CE 5.2.

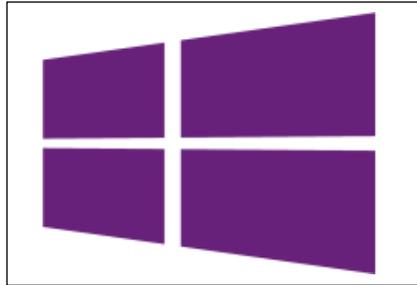


Figure 6: Windows phone (Operating System).

- d. **BlackBerry OS :** est un système d'exploitation pour les téléphones mobiles de la gamme BlackBerry qui a été créé par Research In Motion (RIM) et a été publié en 1999. RIM utilise ce système d'exploitation pour soutenir les fonctions spécialisées. Il comporte la technologie de la pièce jointe telles que les fichiers d'extensions .zip, .html, .doc, .dot, .ppt, .pdf, etc. C'est pourquoi son service de messagerie électronique est meilleur que les autres plateformes.



Figure 7: BlackBerry (Operating System).

- e. **Android OS :** Android est un système d'exploitation mobile basé sur Linux, développé par Google, et distribuée sous licence open source Apache. C'est-à-dire que le code source et les APIs sont ouvertes. Android connaît la plus forte croissance actuelle sur le marché des OS mobiles et est en train de dépasser Apple en parts de marchés mondiales. [15]



Figure 8: Android OS.

3.3 Rôles du système d'exploitation mobiles :

Les rôles des systèmes d'exploitation mobiles sont divers et concernent notamment [16] :

- **La gestion du processeur** : c'est-à-dire la gestion de l'allocation du processeur entre les différentes applications grâce à un algorithme d'ordonnancement.
- **Gestion de la mémoire vive** : c'est-à-dire la gestion de l'espace mémoire alloué à chaque application. En cas d'insuffisance de mémoire physique (RAM), le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée **mémoire virtuelle**.
- **Gestion des entrées/sorties** : c'est-à-dire la gestion de l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes.
- **Gestion de l'exécution des applications** : c'est-à-dire s'assurer de la bonne exécution des applications en leur affectant les ressources nécessaires à leur bon fonctionnement, mais aussi permettre de *tuer* une application ne répondant plus correctement, par exemple.
- **Gestion des fichiers** : c'est-à-dire la gestion de la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.
- **Gestion des droits** : c'est-à-dire la gestion de la sécurité liée à l'exécution des programmes en garantissant que les ressources ne sont pas utilisées que par des programmes et utilisateur possédant les droits adéquats.
- **Gestion des informations** : le système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine.

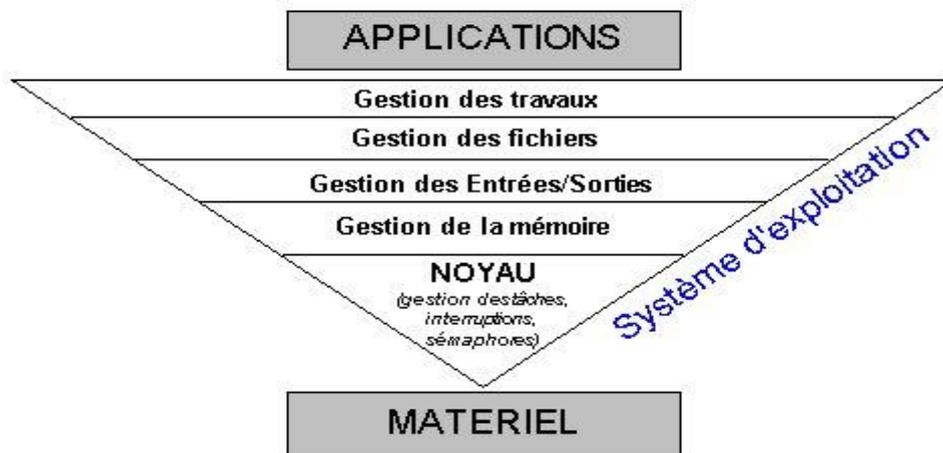


Figure 9: les Rôles du système d'exploitation mobiles.

4 Conclusion :

Dans ce chapitre nous avons tout d'abord présenté et expliqué brièvement le concept d'une application mobile et ses différents types en discutant les avantages et les inconvénients de chaque type. Nous avons aussi donné quelques exemples d'applications mobiles de parking intelligent.

Ensuite, nous avons expliqué ce que c'est un système d'exploitation mobile, présenté les différents systèmes mobiles dans le marché et montré les tâches assurées par ces systèmes.

Chapitre 3 :
Analyse des besoins et
conception

III Chapitre 3 : Analyse des besoins et conception

1 Introduction :

Le but principal de l'application est d'être utile à ses utilisateurs. Par conséquent il est important de comprendre leurs désirs et leurs besoins. Donc, une étape essentielle pour tout programme ou cycle de développement conceptuel est de mener une étude préliminaire. L'objectif principal de cette étape est de comprendre le contexte du système, de mieux illustrer les besoins fonctionnels et non fonctionnels, de montrer les acteurs et de définir des cas d'utilisation. Dans ce chapitre, nous choisissons l'approche UML pour la modélisation des différents diagrammes d'une façon claire et plus compréhensible.

2 Analyse des besoins :

4.1 Spécification des besoins :

C'est une étape nécessaire qui consiste à expliquer, identifier et formaliser tous les besoins de notre application et de définir les contraintes pour la description du modèle fonctionnel (comment fonctionnera l'application ?).

A. Spécification des besoins fonctionnels : cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Le système doit permettre à l'utilisateur :

- L'authentification de client.
- Réservation de place.
- Annulation de réservation.
- Consulter le profil de l'application.

B. Spécification des besoins non fonctionnels : ce sont les besoins qui caractérisent le système, les spécifications supplémentaires c'est-à-dire tout ce qui concerne la qualité, la fiabilité, les performances, les aspects juridiques,

- **L'utilisabilité :** fait référence aux aspects de l'interface utilisateur.
- **La Performance :** les performances d'exécution du système en termes de temps de réponse, de facilité de maintenance à long terme.
- **La disponibilité :** possibilité d'utiliser l'application par n'importe quel utilisateur.
- **La fiabilité :** les données qui sont fournies par l'application doivent être fiables.
En cas d'indisponibilité du système, au moins une solution locale provisoire doit être mise en place.
- **La sécurité :** le système doit être sécurisé (une sécurité totale) surtout au niveau de gestion des accès aux données.

5 Choix de La démarche :

Pour représenter graphiquement les modèles, il existe plusieurs langages de modélisation. Parmi lesquels, on va utiliser le langage UML.

En effet, le choix d'UML n'est pas aléatoire ; il s'agit d'un standard très répandu et qui a fait ses preuves dans le domaine de la modélisation. UML n'est, toutefois, qu'un langage permettant de modéliser un problème d'une façon standard (contrairement à MERISE qui est un langage assorti d'une méthode). Donc, Il doit être accompagné d'un processus ou d'une méthode qui devra guider la modélisation, étape par étape jusqu'à la réalisation.

6 La conception :

Présentation du langage UML :

UML est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

UML est né de la fusion des trois méthodes qui ont influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE. Il s'agit d'un compromis qui a été trouvé par une équipe d'experts : Grady Booch, James Rumbaugh et Ivar Jacobson. UML est à présent un standard défini par l'Object Management Group (OMG). De très nombreuses entreprises de renom ont adopté UML et participent encore aujourd'hui à son développement.

6.1 Les diagrammes d'UML :

L'UML utilise des éléments et les associe de différentes manières pour former des diagrammes qui représentent les aspects statiques ou structurels d'un système, ainsi que des diagrammes comportementaux qui capturent les aspects dynamiques d'un système.

6.1.1 Les diagrammes UML structurels :

- **Diagramme de classe** : Il représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.
- **Diagramme d'objet** : La représentation des instances des classes constitue l'objet du diagramme d'objets.
- **Diagramme de composants** : Il représente les différents constituants logiciels d'un système.
- **Diagramme de déploiement** : Il décrit l'architecture technique d'un système.
- **Diagramme des packages** : Il existe deux types de dépendances entre les packages : l'importation de packages et la combinaison de packages. Les packages peuvent représenter différents niveaux d'un système pour révéler sa structure. Les dépendances du package peuvent être distinguées pour mettre en évidence le mécanisme de communication entre les différents niveaux.
- **Diagramme de structure composite** : Les diagrammes de structure composite sont utilisés pour présenter la structure interne d'une classe.

6.1.2 Les diagrammes UML comportementaux :

- **Diagramme de cas d'utilisation** : Il est destiné à représenter les besoins des utilisateurs par rapport au système.
- **Diagramme état-transition** : Il montre les différents états des objets en réaction aux événements.
- **Diagramme d'activité** : Il donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation.
- **Diagramme de collaboration** : C'est une autre représentation des scénarios des cas d'utilisations qui met plus l'accent sur les objets et les messages échangés.
- **Diagramme de séquence** : Montre comment les objets interagissent les uns avec les autres et dans quel ordre pour réaliser une fonctionnalité donnée : un scénario d'interaction.
- **Diagramme de temps** : À l'instar d'un diagramme de séquence, le comportement des objets est représenté sur un laps de temps donné.

7 Modélisation de notre système :

Pour la modélisation de notre travail, nous avons utilisé trois diagrammes : le diagramme de classe pour modéliser l'aspect statique et les diagrammes de cas d'utilisation et de séquence pour modéliser l'aspect dynamique.

7.1 Diagramme de cas d'utilisation :

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.

La représentation d'un diagramme de cas d'utilisation met en jeu les concepts suivants :

Acteur : Représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Cas d'utilisation (use case) : Représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.

Les relations entre les acteurs : La seule relation entre les acteurs est la relation de généralisation. Quand un acteur fils hérite d'un acteur père, il hérite en réalité de toutes les associations du père.

Les relations entre cas d'utilisation : on peut distinguer les trois types de relations entre cas d'utilisation.

Relation d'inclusion : Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.

Relation d'extension : Une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B

7.2 Description des cas d'utilisation :

Nous proposons une description textuelle des cas d'utilisation comme suit :

- Un titre du cas d'utilisation.
- L'acteur principale du cas d'utilisation.
- L'objectif visé par ce cas.
- Les conditions liés à ce cas : pré-condition, post-condition.
- Les scénarios : les différents scénarios de ce cas.

Chapitre 3 : Analyse des besoins et conception.

7.2.1 Cas d'utilisation « Création d'un compte »

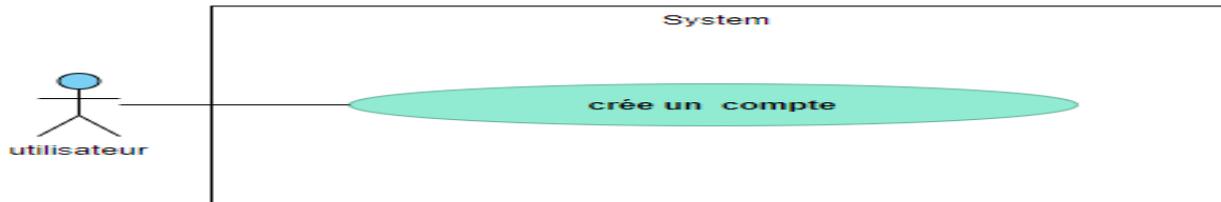


Figure 11: Cas d'utilisation (création d'un compte).

SOMMAIRE	
Titre :	Inscription (la création d'un compte).
Objectif :	La création d'un compte pour un utilisateur.
Acteur :	L'utilisateur : n'importe quel internaute.
Description des enchaînements	
Pré conditions	Post conditions
- Téléchargement de l'application (l'application est déjà installée sur le mobile).	- L'utilisateur peut accéder à l'application.
Scénario nominal	
<ol style="list-style-type: none">1. Un internaute veut être un utilisateur de l'application.2. Il accède au formulaire de l'inscription.3. Il remplit tous les champs.4. Il valide son inscription.5. Le système vérifie les informations et le compte est créé.	
Enchaînement alternatif	
le système trouve que les informations sont incomplètes ou erronés.	

Tableau 2 : Description de scénario "création de compte".

Chapitre 3 : Analyse des besoins et conception.

7.2.2 Cas d'utilisation « Authentification » :

SOMMAIRE	
Titre :	Authentification
Objectif :	La connexion d'un utilisateur au système.
Acteur :	L'utilisateur.
Description des enchainements	
Pré conditions	Post conditions
-L'utilisateur possède un compte.	- L'accès au système est effectué avec succès.
Scénario nominal	
<ol style="list-style-type: none">1. L'utilisateur se connecte au système de parking à travers son application mobile.2. Le système demande un login et un mot de passe.3. L'utilisateur entre le login et le mot de passe.4. Le système vérifie les informations et donne le droit d'accès.	
Enchaînement alternatif	
le système trouve que les informations sont incomplètes ou erronés.	

Tableau 3: Description de scénario "Authentification".

Chapitre 3 : Analyse des besoins et conception.

7.2.3 Cas d'utilisation « Consulter le profil » :

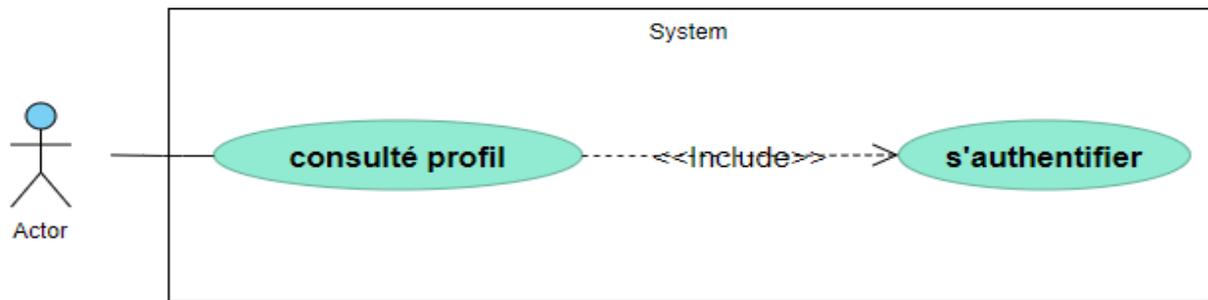


Figure 12: Cas d'utilisation "Consulter le profil".

SOMMAIRE	
Titre :	Consulter la page profil.
But :	La connexion d'un utilisateur au système puis consulter leur page profil.
Acteur :	L'utilisateur
Description des enchainements	
Pré conditions	Post conditions
-l'utilisateur est authentifié	-L'accès à la page profil est effectué avec succès.
Scénario nominal	
1.L'utilisateur communique son désir d'accéder au système à travers son application mobile. 2.Le système demande un login et un mot de passe. 3.L'utilisateur Consulte sa page profil 5.l'utilisateur a la possibilité d'avoir les informations sur sa page profil	
Enchaînement alternatif	

Tableau 4: Description de scénario "Consulter profil".

7.2.4 Cas d'utilisation « Rechercher un parking » :

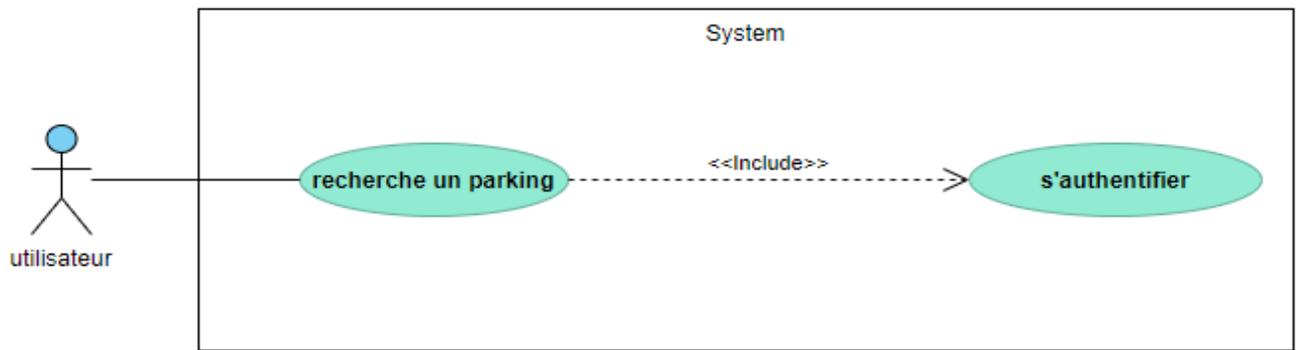


Figure 13 : Cas d'utilisation " Recherche un parking".

SOMMAIRE	
Titre :	Recherche un parking.
Objectif :	L'utilisateur peut rechercher un parking.
Acteur :	L'utilisateur.
Description des enchainements	
Pré conditions	Post conditions
- L'utilisateur est authentifié.	- le système affiche la liste des parkings disponibles.
Scénario nominal	
1. L'utilisateur se connecte à l'application par un login et un mot de passe. 2. Le client clique sur l'icône « rechercher un parking ». 3. Le système présente une page permettant de rechercher un parking à partir d'une liste.	
Enchaînement alternatif	

Tableau 5: Description de scénario "Recherche un parking".

7.2.5 Cas d'utilisation « Choisir un parking » :

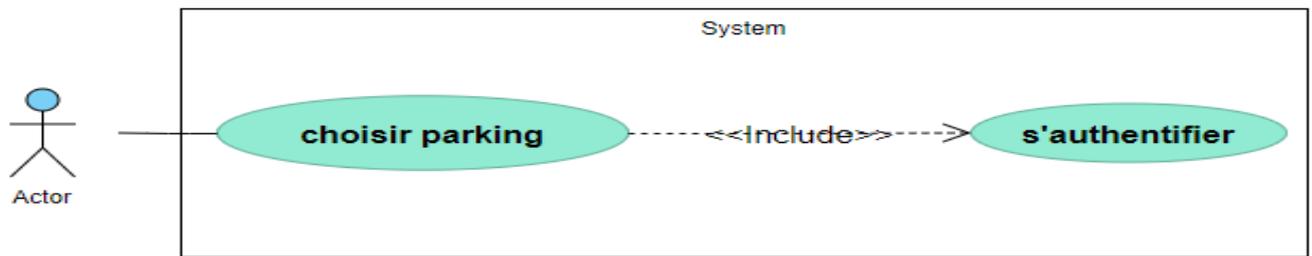


Figure 14: Cas d'utilisation "choisir un parking".

SOMMAIRE	
Titre :	Le choix d'un parking.
Objectif :	Choisir un parking parmi ceux présents dans la liste des parkings disponibles.
Acteur :	L'utilisateur.
Description des enchaînements	
Pré conditions	Post conditions
-L'utilisateur doit être authentifié. -l'utilisateur a déjà fait une recherche (la recherche d'un parking)	-L'utilisateur accède à l'information de ce parking (le parking qu'il a choisi)
Scénario nominal	
1. L'utilisateur est déjà connecté (authentifié) et fait une recherche sur les parkings. 2. Le système affiche une liste de parkings. 3. L'utilisateur choisit un parking. 4. Le système affiche toutes les informations concernant ce parking.	
Enchaînement alternatif	

Tableau 6 : Description de scénario "Choisir un parking".

7.2.6 Cas d'utilisation « Garer » :

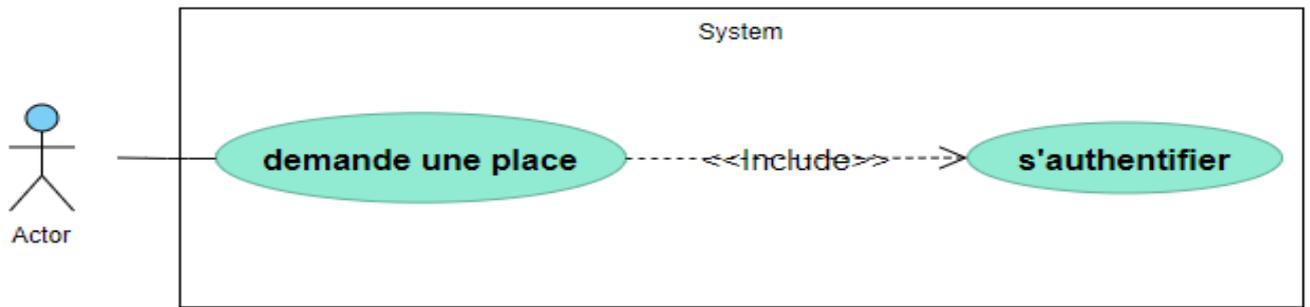


Figure 15: Cas d'utilisation " garer".

SOMMAIRE	
Titre :	Demander une place pour stationner.
Objectif :	Trouver une place pour se garer.
Acteur :	L'utilisateur.
Description des enchainements	
Pré conditions	Post conditions
-L'utilisateur doit être connecté (authentifié).	-L'utilisateur trouve une place.
Scénario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur est déjà connecté. 2. Le système affiche le menu de l'application. 3. L'utilisateur accède à l'interface et clique sur l'icône « garer ». 4. Le système présente une page qui présente les places disponibles. 	
Enchaînement alternatif	

Tableau 7: Description de scénario « Garer ».

7.2.8 Cas d'utilisation « Annuler la réservation » :

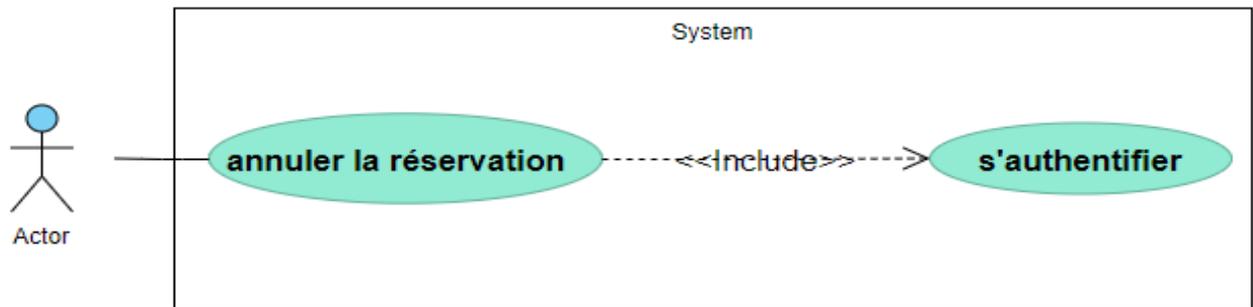


Figure 17: Cas d'utilisation " Annuler une réservation".

SOMMAIRE	
Titre :	Annuler la réservation d'une place.
But :	Annuler une réservation déjà existante.
Acteur :	L'utilisateur.
Description des enchaînements	
Pré conditions	Post conditions
-l'utilisateur est authentifié - l'utilisateur a déjà fait des demandes de réservation.	-l'annulation de la réservation est effectué avec succès.
Scénario nominal	
1. L'utilisateur se connecte au système de parking à travers son application mobile par un login et un mot de passe. 2. Le système présente une page permettant de se connecter au menu de l'application. 3. L'utilisateur fait une annulation de réservation via le clic sur l'icône d'annuler la réservation 4. Le système traite la demande d'annulation et affiche un message de confirmation.	
Enchaînement alternatif	

Tableau 9: Description de scénario « Annuler la réservation ».

7.2.9 Cas d'utilisation « Demande de déconnexion » :

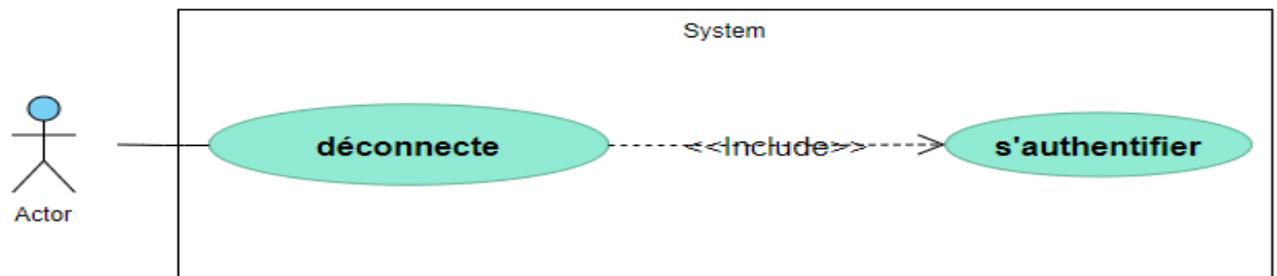


Figure 29 : Cas d'utilisation " Déconnexion".

SOMMAIRE	
Titre :	La déconnexion (log Out).
Objectif :	La déconnexion de compte.
Acteur :	L'utilisateur.
Description des enchaînements	
Pré conditions	Post conditions
-L'utilisateur doit être connecté.	-L'utilisateur est déconnecté.
Scénario nominal	
<ol style="list-style-type: none"> 1. L'utilisateur est déjà connecté et demande de se déconnecter en cliquant sur l'icône <<déconnecter ou <log out> >>. 2. Le système accepte la demande. 3. L'utilisateur est déconnecté 	
Enchaînement alternatif	

Tableau 10: Description de scénario "déconnexion".

7.3 Diagrammes de séquence :

Les diagrammes de séquence décrivent le déroulement de chaque cas d'utilisation, en montrant la façon dont les diverses entités mises en œuvre dans les cas d'utilisation interagissent et collaborent dans le temps afin de réaliser les fonctionnalités attendues.

Le but est de déterminer :

- Les divers entités, appelées objets, mises en jeu dans la réalisation d'une fonctionnalité.
- Les interactions entre ces divers objets.
- Le déroulement dans le temps de ces interactions.

Le diagramme de séquence est en réalité un cas particulier de diagramme d'interaction ayant pour but de mettre en avant l'aspect chronologique des interactions décrites.

Typiquement, chaque cas d'utilisation déterminé dans le diagramme des cas d'utilisation fait l'objet d'une étude temporelle des interactions en utilisant un diagramme de séquence.

Par conséquent, on doit avoir autant de diagrammes de séquence que de cas d'utilisation, occasionnellement, un cas d'utilisation peut être décrit par plusieurs diagrammes de séquence afin de clarifier l'ensemble.

Les diagrammes de séquence sont donc la suite logique du diagramme des cas d'utilisation.

En fonction du niveau de détail désiré, on peut les utiliser de deux manières différentes :

- Forme simplifiée : description sommaire du cas d'utilisation en se basant sur la description textuelle – fin de la phase de spécification du système.
- Forme détaillée : présentation d'une ébauche de la structure du code objet – phase de conception préliminaire.

7.3.1 Diagramme de séquence « Authentification » :

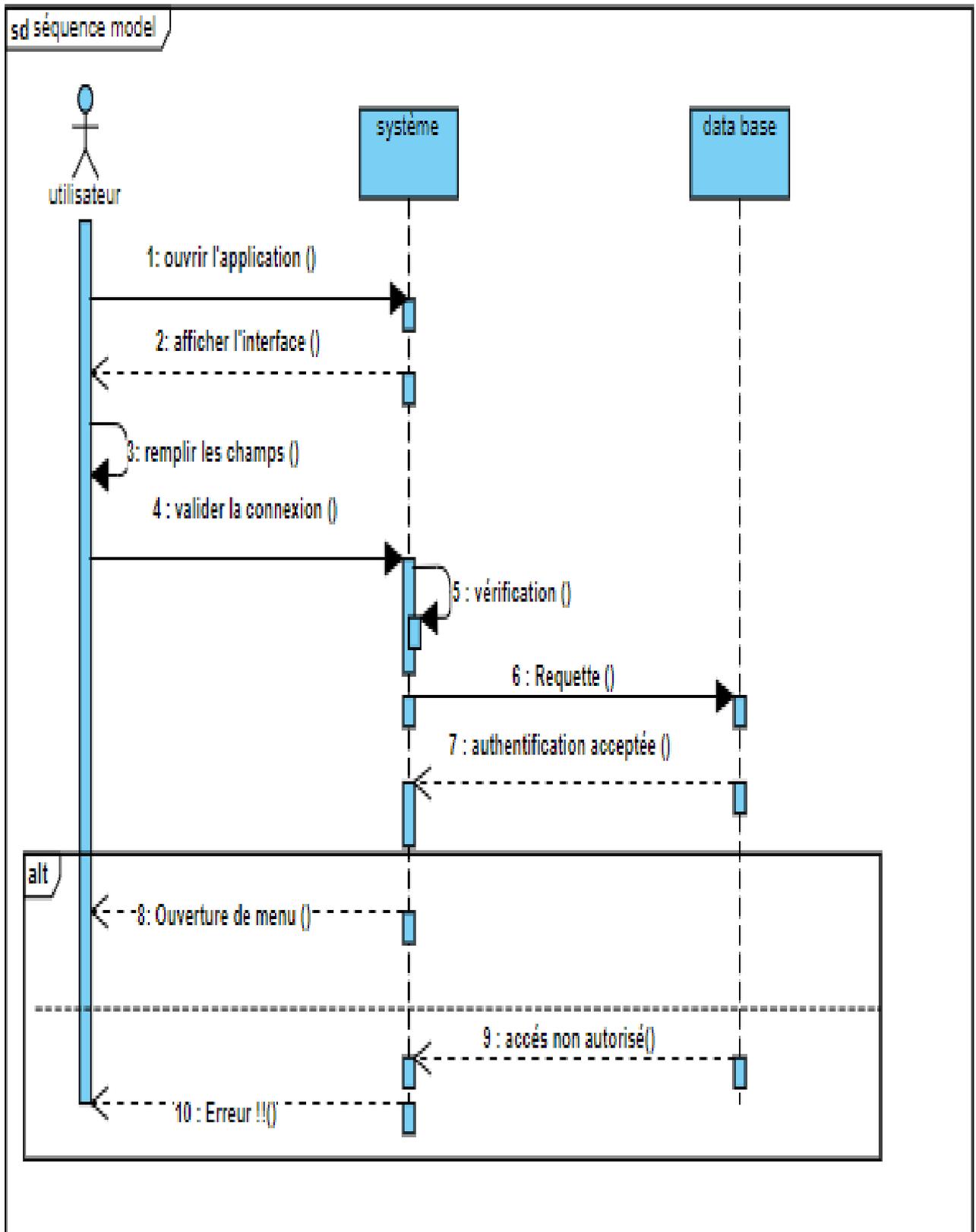


Figure 18 : Diagramme de séquence « Authentification ».

7.3.2 Diagramme de séquence « Création d'un compte : l'inscription »

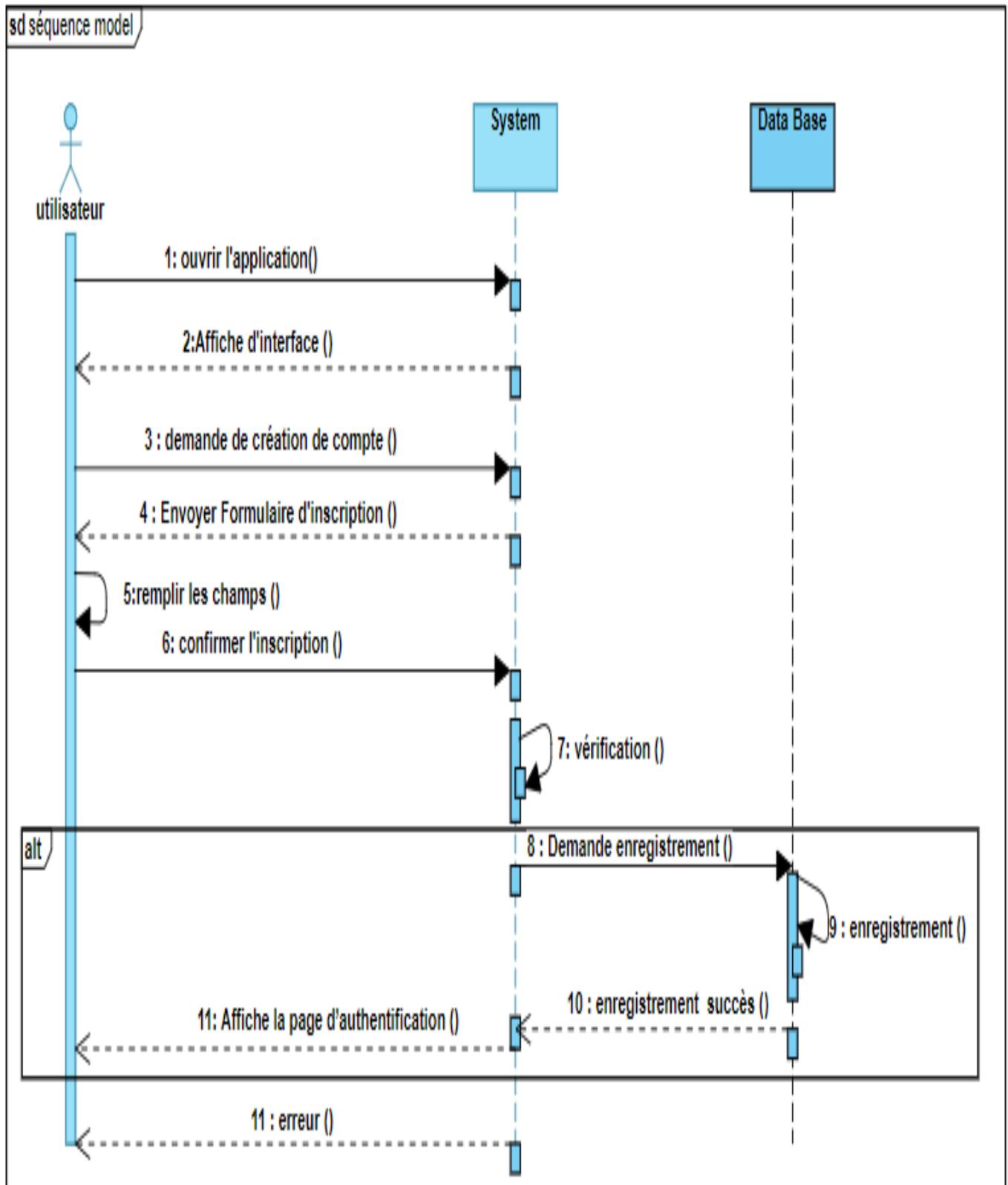


Figure 19: Diagramme de séquence « l'inscription ».

7.3.3 Diagramme de séquence « Consultation de profil » :

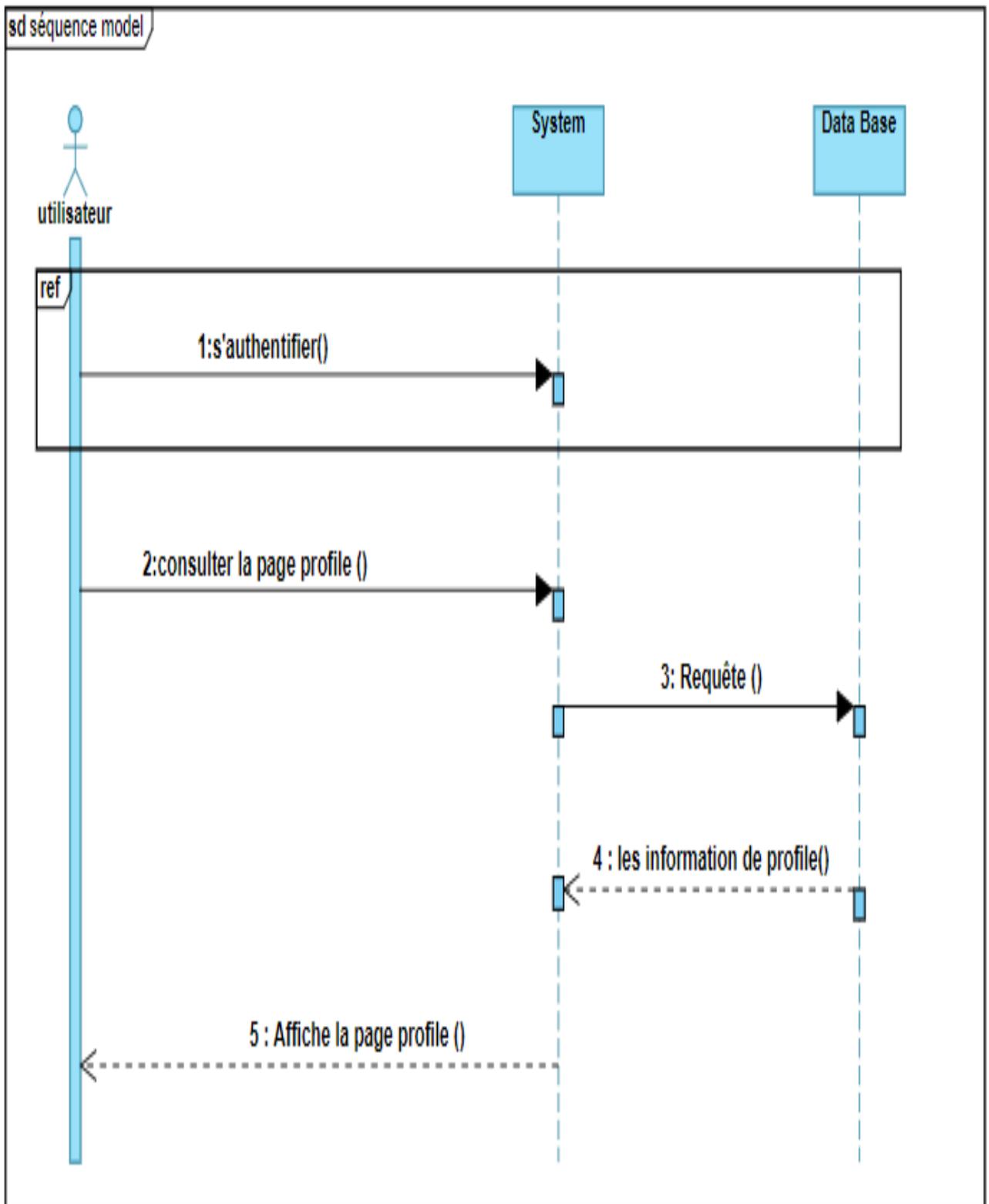


Figure 20 : Diagramme de séquence « Consultation de profil ».

7.3.4 Diagramme de séquence « Réservation de place » :

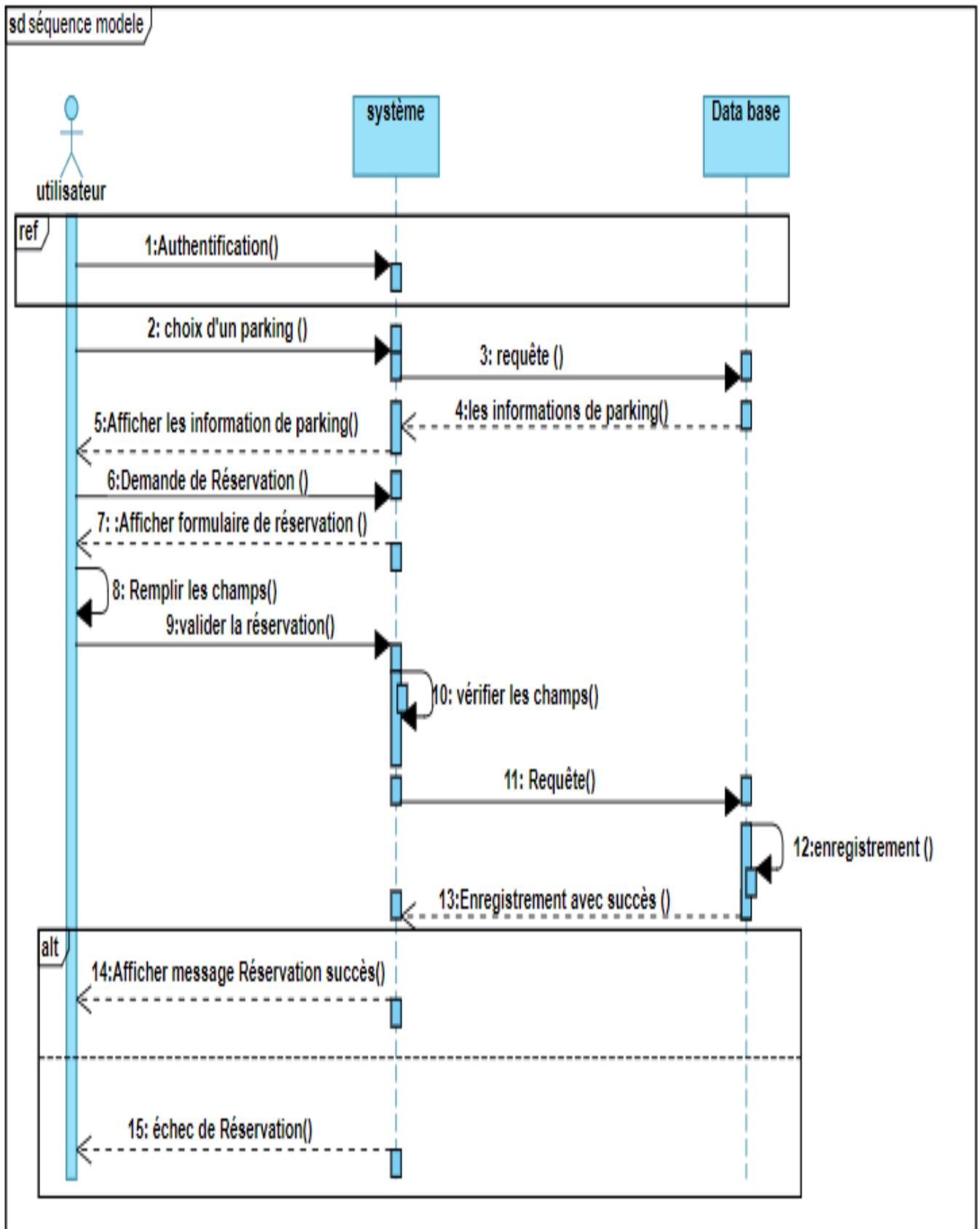


Figure 21: Diagramme de séquence « Réservation de place ».

7.3.5 Diagramme de séquence « Annuler la réservation » :

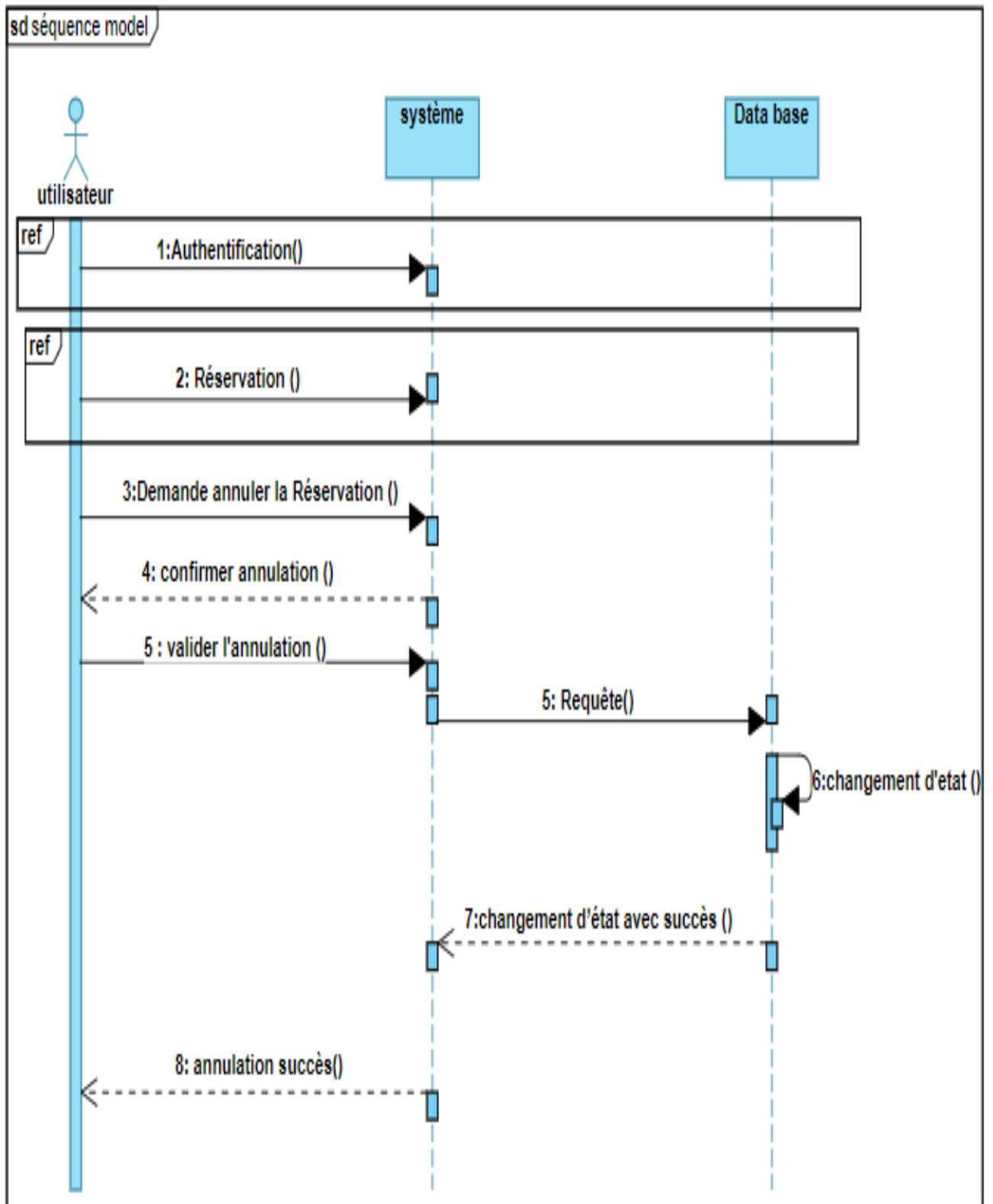


Figure 22 : Diagramme de séquence « Annuler la réservation ».

7.3.6 Diagramme de séquence « Garer » :

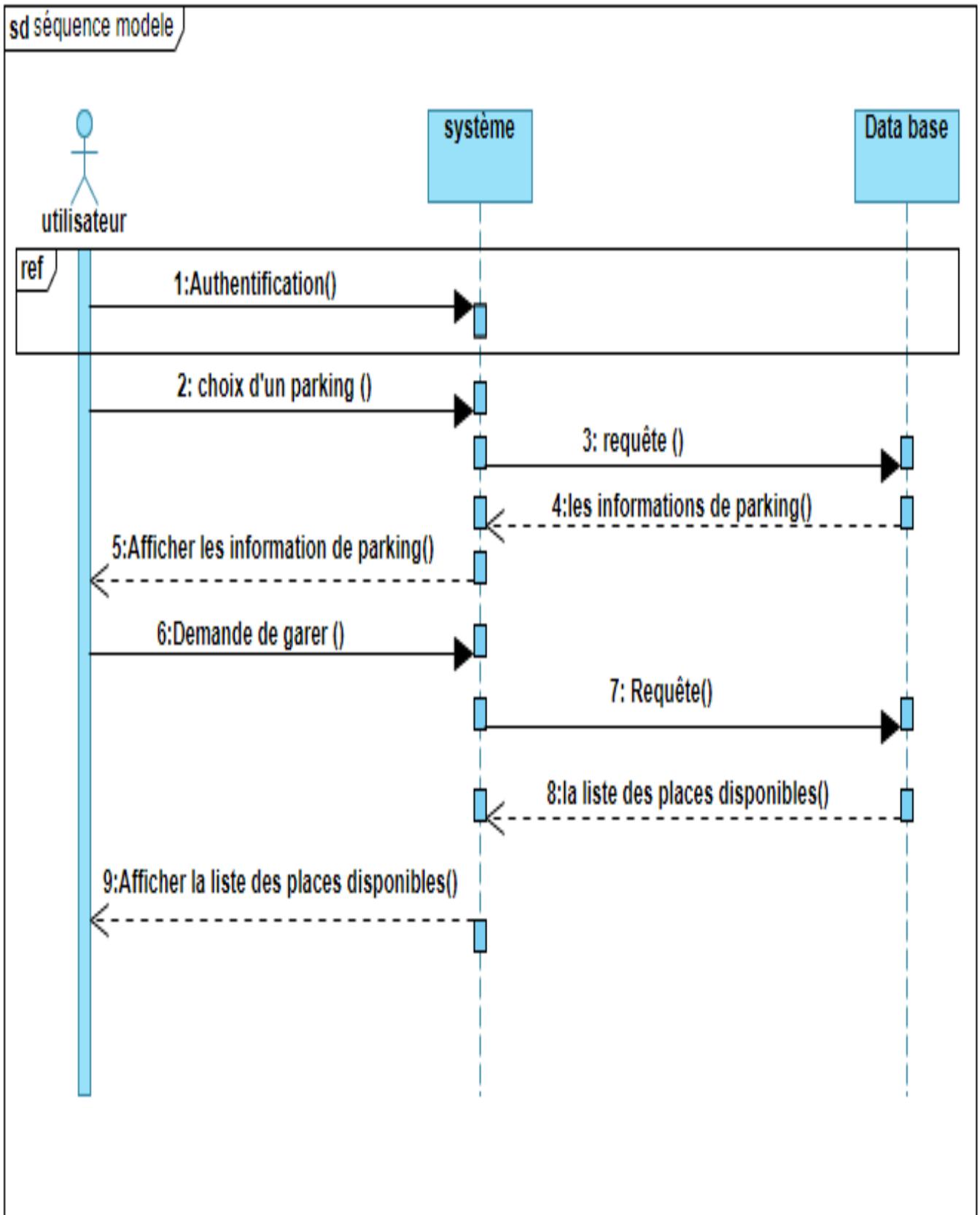


Figure 23 : Diagramme de séquence « Garer ».

7.4 Diagramme d'activité :

Un diagramme d'activités est un diagramme comportemental qui est sémantiquement proche des diagrammes de communication ou état-transitions, ces derniers offrant une vision microscopique des objets du système.

Les diagrammes d'activités sont des représentations graphiques d'activités incrémentielles et de tâches de workflow d'action avec prise en charge de la sélection, de l'itération et de la concurrence. Ces interactions apparaissent dans un scénario décrivant comment les actions se produisent entre les acteurs ou les objets. Bien que les diagrammes d'activités montrent principalement le flux global de contrôle, ils peuvent également inclure des éléments qui illustrent le flux de données entre les activités via un ou plusieurs entrepôts de données.

7.4.1 Les composants de base du diagramme d'activités :

Les diagrammes d'activités sont créés à partir d'un nombre limité de formes, reliées par des flèches. Les types de formes les plus importants sont :

- **Action** : Une étape d'activité au cours de laquelle un utilisateur ou un programme effectue une tâche spécifique. Les actions sont désignées par des rectangles aux bords arrondis.
- **Nœud de décision** : la branche conditionnelle du flux est représentée par un losange. Il a une entrée et deux ports ou plus.
- **Flux de contrôle** : autre nom pour les connecteurs qui affichent le flux entre les étapes d'un diagramme.
- **Nœud de départ** : indique le début de l'activité. Le nœud de départ est représenté par un cercle noir.
- **Nœud de fin** : il s'agit de la dernière étape de l'activité. Le nœud final est représenté par un cercle noir.
- **Les barres** représentent le début (fractionnement) ou la fin (jointure) des activités simultanées.
- **Les flèches** vont du début à la fin et représentent l'ordre dans lequel les activités se déroulent.
- **Les ellipses** représentent des verbes

7.4.2 Les avantages de diagramme d'activité :

Les diagrammes d'activités présentent un certain nombre d'avantages pour les utilisateurs. On crée un diagramme d'activités pour :

- Démontrer la logique d'un algorithme.
- Décrire les étapes effectuées dans un cas d'utilisation UML.
- Illustrer un processus métier ou un flux de travail entre les utilisateurs et le système.
- Simplifier et améliorer n'importe quel processus en clarifiant les cas d'utilisation complexes.
- Modéliser les éléments de l'architecture logicielle, tels que la méthode, la fonction et le fonctionnement.

7.4.3 Diagramme d'activité « authentification » :

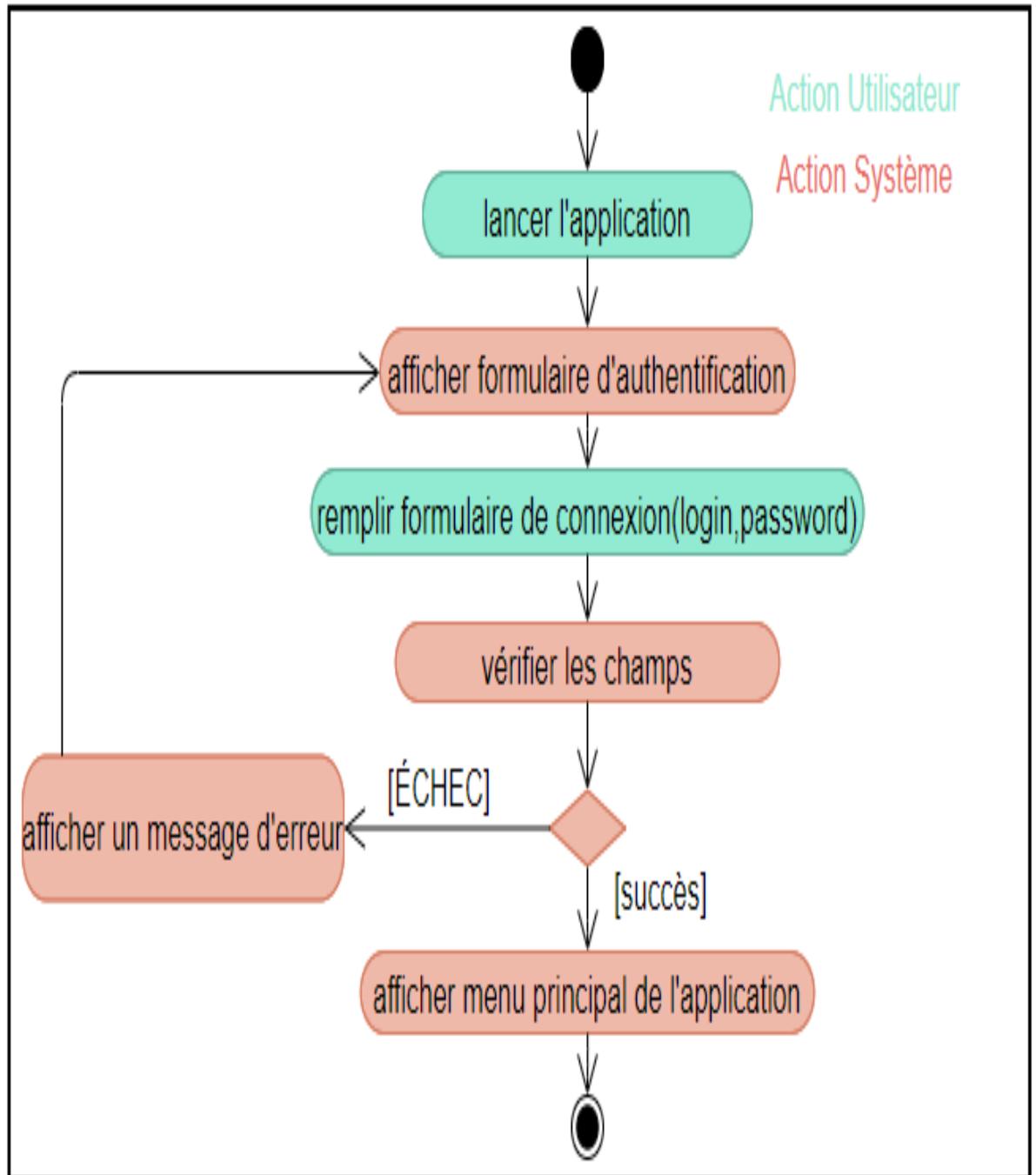


Figure 24 : Diagramme d'activité « authentification ».

7.5 Diagramme de classe :

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation.

Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application. Chaque langage de Programmation Orienté Objets donne un moyen spécifique d'implémenter le paradigme objet (pointeurs ou pas, héritage multiple ou pas, etc.), mais le diagramme de classes permet de modéliser les classes du système et leurs relations indépendamment d'un langage de programmation particulier.

Les principaux éléments de cette vue statique sont les classes et leurs relations :

- Association
- Généralisation
- Plusieurs types de dépendances, telles que la réalisation et l'utilisation.

7.5.1 Diagramme de classe global :

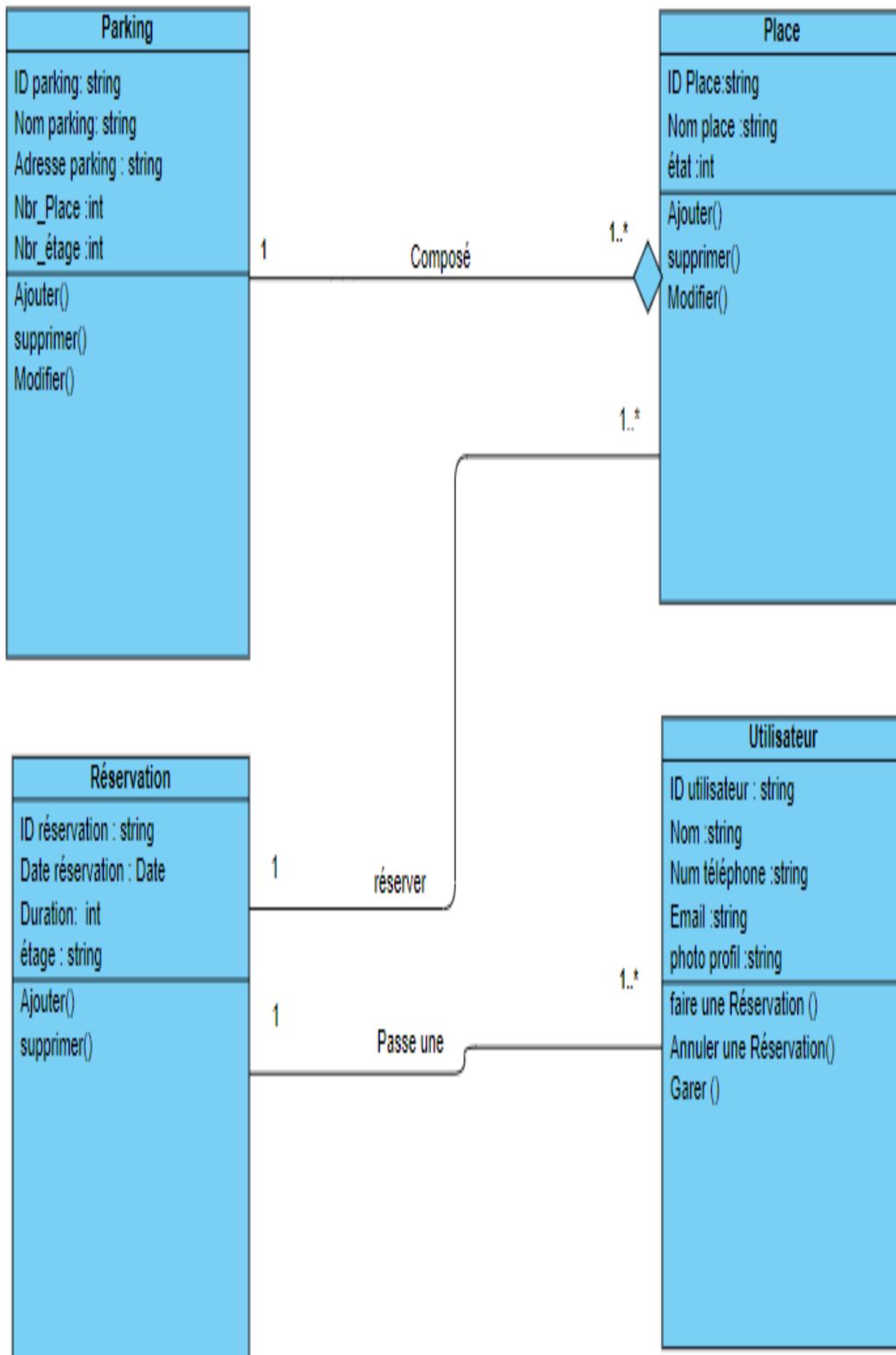


Figure 25: Diagramme de classe global.

Chapitre 3 : Analyse des besoins et conception.

7.6 Le dictionnaire de données :

Pendant la phase de conception, les données recueillies et spécifiées sont inscrites dans un dictionnaire. Ce dictionnaire est un outil important car il constitue la référence de toutes les études effectuées ensuite.

Tableau 11: Dictionnaire de données.

N°	Code	Désignation	Type
1	IdParking	Identificateur de parking	Varchar
2	NomParking	Nom de parking	Integer
3	AdressParking	Adresse de parking	Text
4	NbrPlace	Nombre de place dans le parking	Integer
5	Etat _parking	Etat de parking	Text
6	Login	Login d'utilisateur	Varchar
7	Mot-passe	Mot de passe de l'utilisateur	Varchar
8	Num_téléphone	Numéro téléphone de l'utilisateur	Text
9	Email	Email de l'utilisateur	Text
10	Nom _utilisateur	Nom de l'utilisateur	Text
11	Prénom	Prénom de l'utilisateur	Text
12	IdPlace	Identificateur de la place	Varchar
13	Etat _place	Etat de la place	Varchar
14	Num _place	Numéro de la place	Integer
15	IdRéservation	Identificateur de la réservation	Varchar
16	Date _réservation	La date de la réservation	Time
17	Duration	La durée de la réservation	Time
18	L'heur _début _R	L'heur de début de la réservation	Time

Chapitre 3 : Analyse des besoins et conception.

19	Durée de stationnement	La durée	TIME
20	nbrEtage	Nombre de l'étage	Integer

7.7 Modèle relationnel :

A partir de la description conceptuelle que nous avons effectuée, on présente ici le modèle relationnel que l'on obtient à partir du diagramme de classes après l'utilisation des règles de passages de l'UML vers le relationnel.

Remarque : pour la notation, nous avons choisi de mettre une ligne sous les clés primaires et de mettre (*) à la fin de chaque clé étrangère. Voici les tables du modèle relationnel que nous avons obtenu :

Utilisateur (id utilisateur, nom, prénom, email, Num_téléphone, mot de passe)

Réservation (id réservation, date_réservation, duration, l'heur_début_R, id_place*, id_utilisateur*)

Place (id place, Num_place, état _ place, id parking*)

Parking (id parking, nom, Numéro Rue, Ville, code postale, Nbr_place, état_parking)

7.8 Présentation de la base de données :

Notre base de données « **SmartParking** » a été créée avec Firebase. Les données sont stockées au format JSON et synchronisées en temps réel avec chaque client connecté.

- La base de données contient des collections : **SmartParking** qui contient 4 collections : Parkings, Réservation, Users et Places.
- Chaque collection contient des documents, comme la collection Parkings qui contient 3 documents.
- Chaque document contient des propriétés

Voilà notre base de données :

smartparking	Parkings	58vAbalXQpAoiWNqcDIN
+ Commencer une collection	+ Ajouter un document	+ Commencer une collection
Parkings >	58vAbalXQpAoiWNqcDIN >	+ Ajouter un champ
Places	UYGw5qbsGxUoq901eMSN	addressParking: [36.0736438° N, 4.7685346° E]
Reservation	eTIkrrDLhhQArnRaSFEA	idParking: "58vAbalXQpAoiWNqcDIN"
Users		idPlace
		nbrEtages: "1"
		nbrPlaces: "5"
		nomParking: "Big Parking"

Figure 26 : la base de données.

8 Conclusion :

Dans ce chapitre nous avons présenté l'analyse et les différentes étapes de conception de notre application en élaborant différents diagrammes d'UML. Ces diagrammes nous ont facilité l'implémentation que nous présentons dans le chapitre suivant.

Chapitre 4 :

Implémentation

IV Chapitre 4 : Implémentation

1 Introduction :

La phase de l'implémentation représente le dernier volet de ce travail, après avoir terminé la phase de l'analyse et spécification des besoins et de la conception.

Dans ce chapitre nous présentons les principales étapes de notre réalisation, qui traduit le passage du modèle conceptuel vers une application mobile d'un parking intelligent et nous exposons les choix techniques utilisés et le langage adopté. La dernière partie de ce chapitre est consacrée à présenter l'application.

L'idée derrière notre application est d'aider le client (utilisateur) à trouver des places libres pour stationner et de faciliter la réservation en utilisant son smartphone.

2 Choix de l'architecture de l'application :

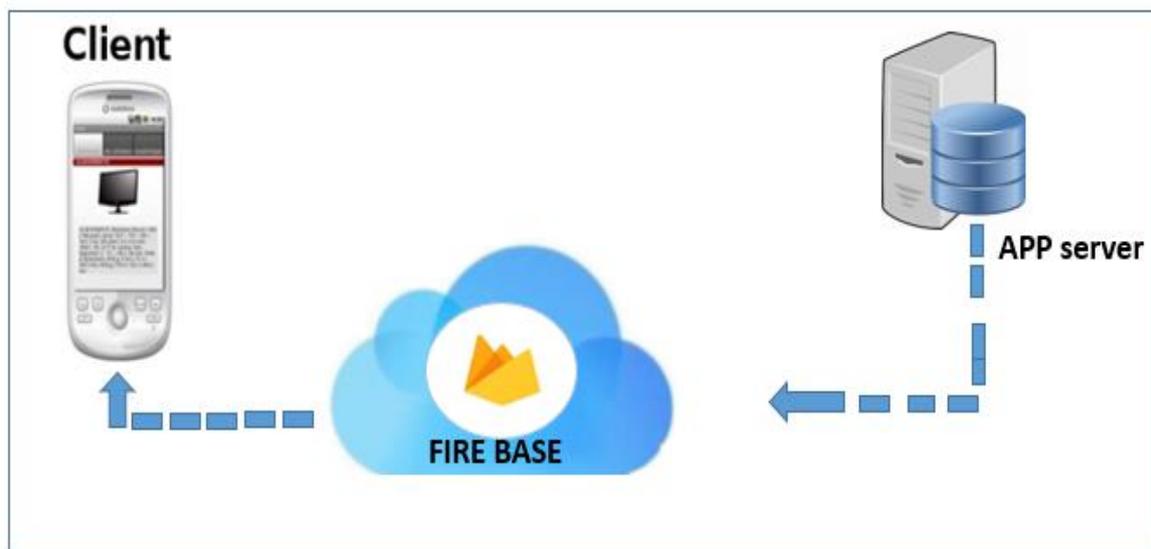


Figure 39: L'architecture de l'application.

L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches [18], est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- **Niveau de présentation (premier couche)**

Il s'agit du niveau le plus élevé de l'application associé au client qui de fait est dit « léger ». Elle correspond à la partie visible et interactive de l'application pour les utilisateurs. On parle d'interface homme-machine. Elle a donc une importance primordiale pour rendre l'information lisible, compréhensible et accessible.

La couche de présentation relaie les requêtes de l'utilisateur à destination de la couche de traitement, et en retour lui présente les informations renvoyées par les traitements de cette

Chapitre 4 : Implémentation.

couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.

- **Niveau d'application (logique métier ou niveau intermédiaire)**

Elle correspond à la partie fonctionnelle de l'application, Il contient un ensemble de règles pour le traitement des informations : la logique métier et peut accueillir de nombreux utilisateurs. Ce niveau est parfois également appelé middleware.

Il traite les entrées reçues des clients et interagit avec la base de données.

La couche de traitement offre des services applicatifs et métier à la couche de présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche de présentation les résultats qu'elle a calculés.

- **La couche de données liée au serveur de base de données (SGBD) :**

Elle comprend à la fois des ensembles de données et le système de gestion de base de données ou le logiciel SGBDR qui gère et donne accès aux données d'application. Le système de stockage des données a pour but de conserver une quantité de données de façon structurée. Il fournit la sécurité, l'intégrité des données et l'application de support.

Le niveau de données serait une sorte de base de données, telle qu'une base de données MySQL, SQLite ou PostgreSQL. Tous ces éléments sont exécutés sur un serveur de base de données distinct.

2.1 Avantages de l'architecture client/serveur :

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont [19]:

- **Maintenabilité** : chaque niveau étant indépendant des autres niveaux, des mises à jour ou des modifications peuvent être effectuées sans affecter l'application dans son ensemble.
- **Évolutivité** : Les niveaux étant basés sur le déploiement de couches, la mise à l'échelle d'une application est raisonnablement simple.
- **Flexibilité** : Étant donné que chaque niveau peut être géré ou mis à l'échelle indépendamment, la flexibilité est accrue.
- **Disponibilité** : Les applications peuvent exploiter l'architecture modulaire des systèmes habilitants à l'aide de composants facilement évolutifs, ce qui augmente la disponibilité.
- **Réutilisabilité** : Les composants sont réutilisables
- **Une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important.
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

3 Langage de développement et le Framework

3.1 Langage Dart :

Dart (initialement appelé Dash4) est un langage de programmation open-source polyvalent destiné au serveur ainsi qu'au navigateur. Il est à l'origine développé par Google en 2007 et approuvé plus tard en tant que norme par l'ECMA (European computer manufactures Association). Il est utilisé pour la programmation côté serveur, ainsi que le développement d'applications mobiles (via l'API Flutter). Son but initial est de remplacer JavaScript.

Dart est un langage simple, productif, rapide et portable qui utilise une approche entièrement orientée objet, basé sur les classes, avec une syntaxe de style C, pour le rendre simple et accessible. Le code Dart peut compiler en code natif ou en JavaScript compatible avec tous les navigateurs modernes, ainsi que sur le développement d'application multiplateforme. Il prend en charge les interfaces, les classes abstraites, les génériques réifiés et l'inférence de type.

Dart est un langage conçu pour être familier et donc accessible à de nombreux développeurs existants, grâce à ses aspects orientés objet et sa syntaxe qui (selon ses utilisateurs) permet à tout développeur C++, C#, Objective-C ou Java d'être productif en quelques jours. [20]



Figure 27: Logo de langage DART.

3.2 Flutter Framework :

Flutter est un Framework de développement de logiciel d'interface utilisateur open source créé par Google. Il permet de développer des applications de haute qualité et hautes performances pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le Web à partir d'une seule base de code.

La première version de Flutter était connue sous le nom de code "Sky" et fonctionnait sur le système d'exploitation Android. Il a été dévoilé lors du sommet des développeurs Dart 2015.

Le cadre a été officiellement publié en décembre 2018 lors d'une conférence organisée par Google.

Le Framework a acquis une immense popularité dans différents espaces à partir de sites de développement de logiciels prestigieux tels que GitHub et Stack Overflow en se démarquant de ses concurrents.

Flutter intègre un kit de développement logiciel (SDK) qui comprend des outils qui aident le Framework à engager un développement interactif et plus rapide de code machine natif pour différentes plates-formes. Les applications mobiles développées à l'aide de Flutter sont codées dans le langage de programmation Dart qui est compilé en JavaScript. Cette approche permet à Flutter d'utiliser une bibliothèque de base avancée et des widgets orientés conception pour développer des applications. [21]

3.3 Framework architecture :

Les principaux composants de Flutter incluent [22] :

- **Dart Platform** : Les applications Flutter sont écrites dans le langage Dart et utilisent bon nombre des fonctionnalités plus avancées du langage.
- **Le moteur de Flutter** : écrit principalement en C ++, fournit une prise en charge du rendu de bas niveau à l'aide de la bibliothèque graphique Skia de Google.
- **La bibliothèque Fondation** : écrite en Dart, fournit des classes et des fonctions de base permettant de créer des applications utilisant Flutter, telles que des API pour communiquer avec le moteur.
- **Design-spécifique widgets** : Le Framework Flutter contient deux ensembles de widgets conformes à des langages de conception spécifiques. Les widgets Materials Design implémentent le langage de design Google du même nom, et les widgets Cupertino imitent le design iOS d'Apple.

Chapitre 4 : Implémentation.

◆ Flutter : points positifs.

- Une très bonne performance proche du natif.
- Utilisation d'Android studio, Visual studio code ou de n'importe quel autre IDE.
- Moins de code pour le même résultat en natif.
- Compilation extrêmement rapide : l'application se charge automatique quand le code est modifié de manière presque instantanée.
- Maintenance moins coûteuse.
- Montée en compétence très rapide en venant d'Android.
- Les corrections de bugs sont rapides et fréquentes



Figure 28: Logo de Flutter.

4 Les outils de développement :

4.1 Firebase

Firebase est une plateforme de google qui permet de développer rapidement des applications pour mobile et pour le web. Elle fournit aux développeurs une pléthore d'outils et de services pour les aider à développer des applications de haute qualité, à élargir leur base d'utilisateurs et à générer davantage de profits. [23]

L'objectif de la création de Firebase.google.com en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.



Figure 29: Logo de Firebase.

4.1.1 Les services de Firebase :

Les services de Firebase sont [24]:

- Les outils de développement et de test de son application ;
- Les outils permettant d'augmenter et d'engager ses cibles.



Figure 30 : les services de Firebase.

4.2 Visual Studio Code :

Visual Studio code est un éditeur de code multiplateforme édité par Microsoft, open-source et gratuit. [25]

Cet outil destiné aux développeurs supporte plusieurs dizaines de langages de programmation comme le HTML, C++, PHP, Javascript, Markdown, CSS, etc.

Visual Studio Code intègre plusieurs outils facilitant la saisie de code par les développeurs comme la coloration syntaxique. En outre, l'outil permet aux développeurs de corriger leur code et de gérer les différentes versions de leurs fichiers de travail puisqu'un module de débogage est aussi de la partie



Figure 31: Logo de Visual Studio Code.

5 Présentation de l'application :

Dans cette partie nous allons présenter notre application, sous forme d'un guide pour l'utilisateur.

Pour accéder à notre application **Smart Parking**, l'utilisateur doit télécharger et installer l'application sur son smartphone.

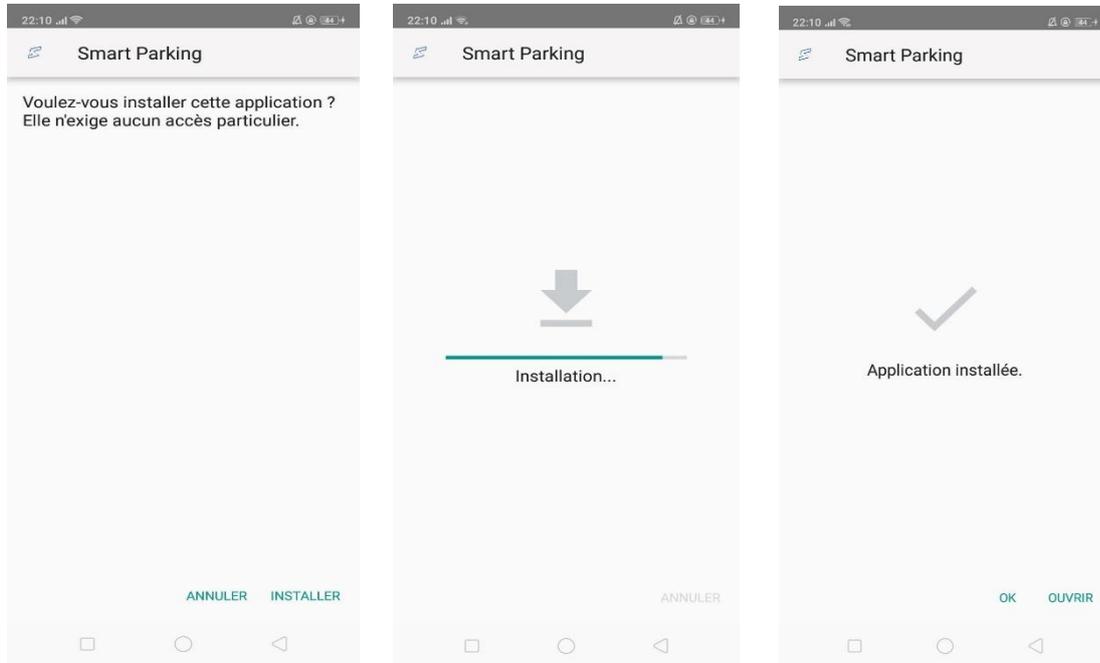


Figure 32 : Interface de l'installation de l'application.

La figure ci-après donne la première interface de l'application.



Figure 33 : la première interface de l'ouverture de l'application.

5.1 La page de l'inscription :

Après l'installation de l'application sur le téléphone portable, l'utilisateur doit s'inscrire à travers la saisie de ses informations dans tous les champs.

En cas où l'utilisateur a oublié de remplir un des champs, un message sera affiché pour informer que le remplissage est obligatoire.

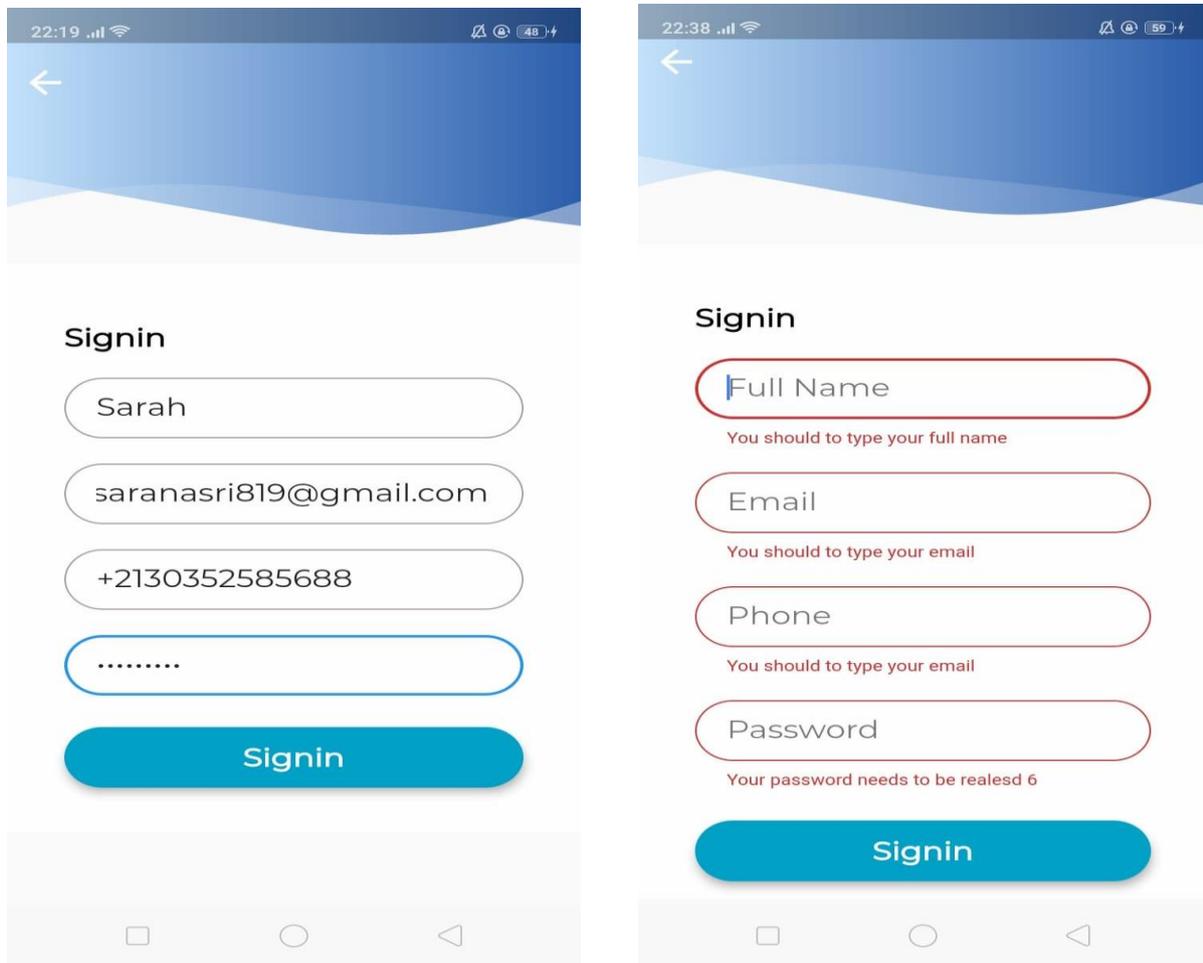


Figure 34: L'interface de la page de l'inscription.

5.2 La page de l'authentification :

Comme toute application, pour accéder à notre application Smart Parking, l'utilisateur doit s'authentifier.

Nous avons adopté deux méthodes pour l'authentification :

- **Authentification par email et mot de passe.**

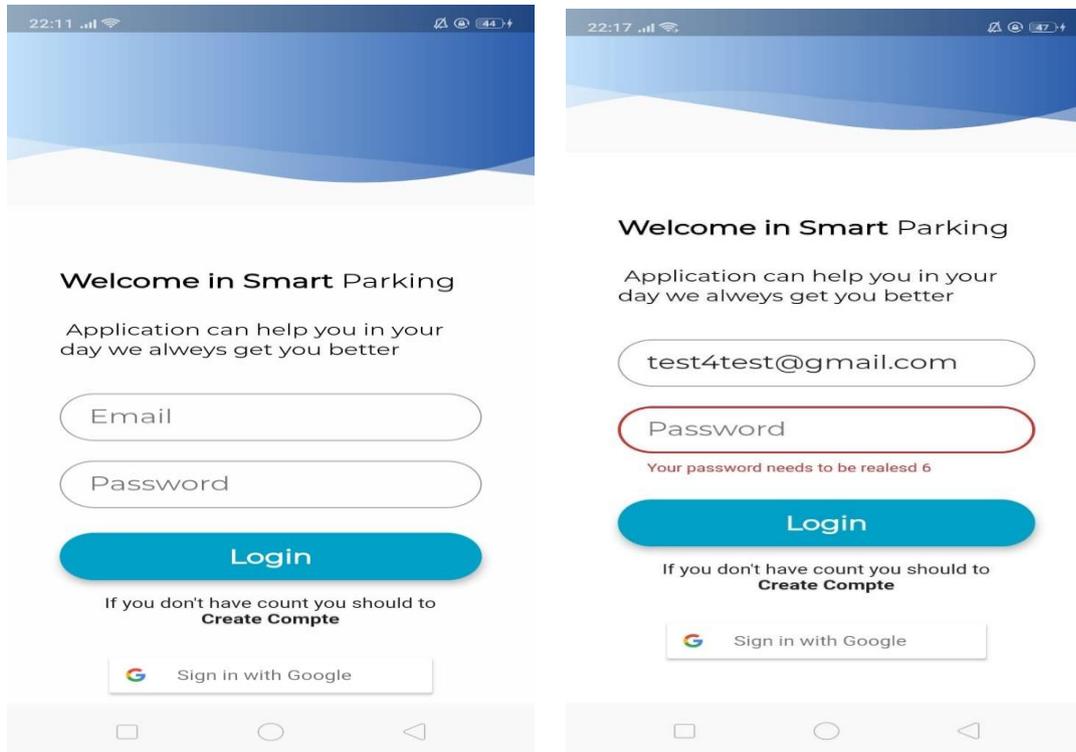


Figure 35: l'interface de la page de l'authentification.

Chapitre 4 : Implémentation.

- **Authentification par Google :**

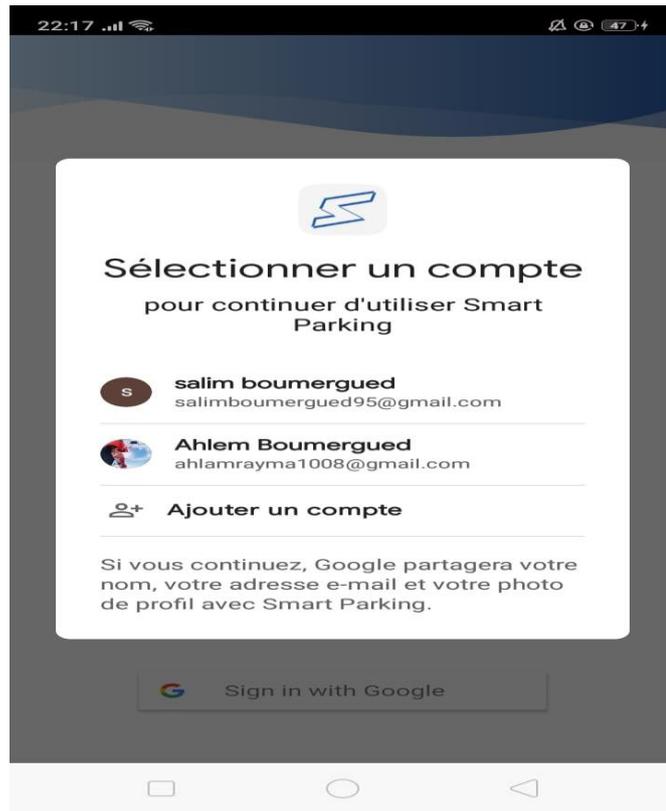


Figure 49: l'interface de l'authentification par Google.

5.3 La page de profil :

Une fois l'utilisateur s'est authentifié, il peut consulter son profil pour voir ses informations.

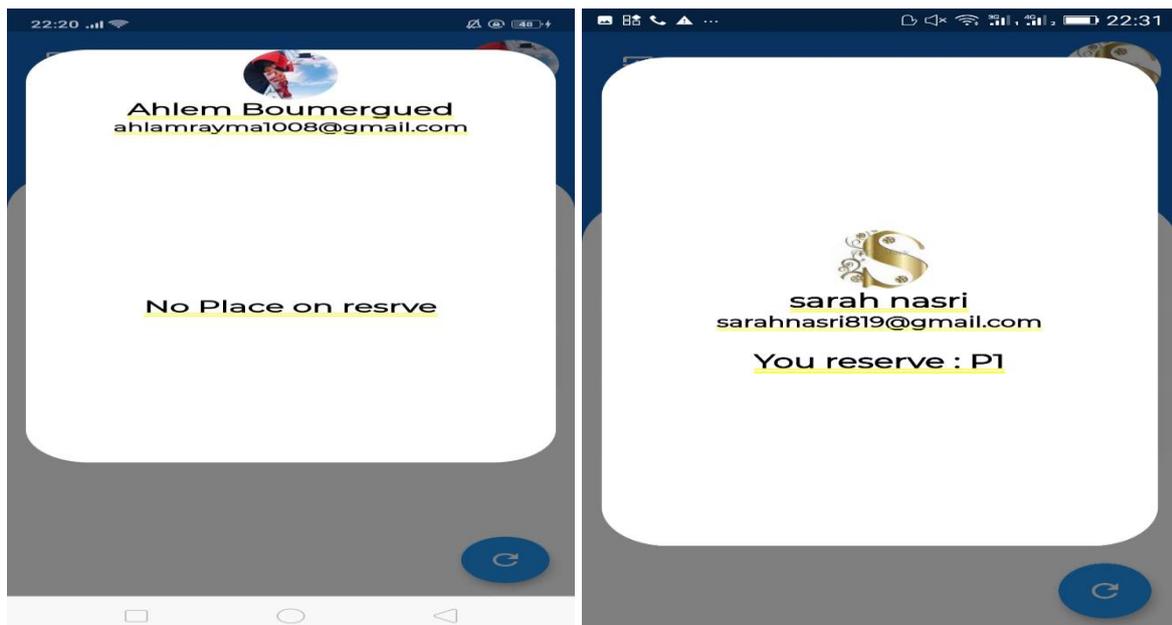


Figure 36: l'interface de la page profil.

5.4 La page de la recherche :

Une fois que l'utilisateur accède à l'application, il peut faire une recherche pour voir la liste des parkings disponibles.

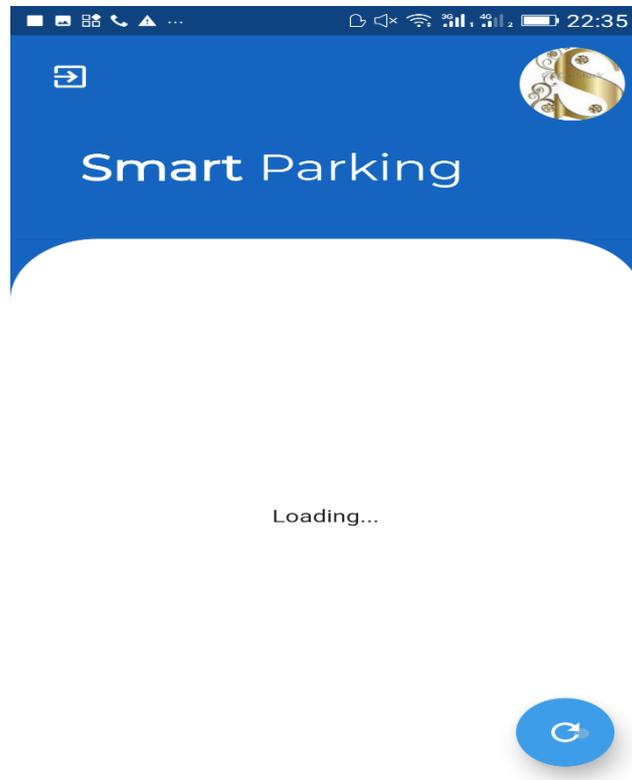


Figure 37 : l'interface de la page de la recherche.

5.5 La page information de parking :

Ici l'utilisateur peut voir toutes les informations concernant les parkings qu'il peut choisir.

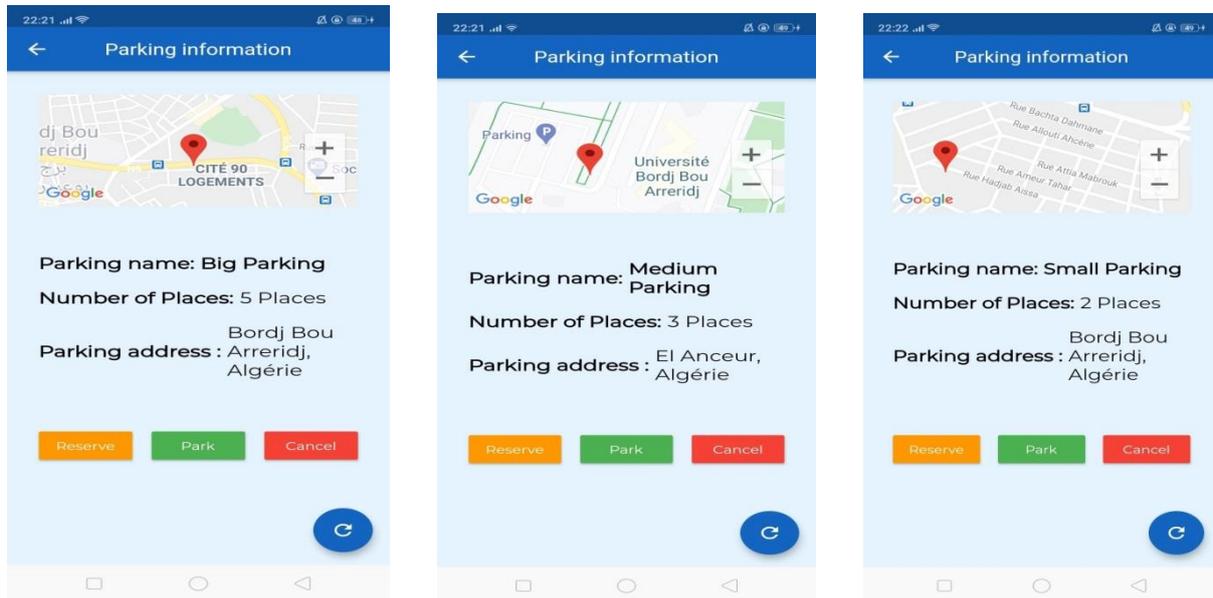


Figure 38: l'interface de la page d'information de parking.

6 Les services que l'application doit fournir :

L'utilisateur a le droit de faire trois opérations :



Figure 39: les services de l'application.

A. La première opération : réservation d'une place :

L'utilisateur ne peut réserver qu'une place à la fois. Pour cela il doit choisir une place et remplir les champs.

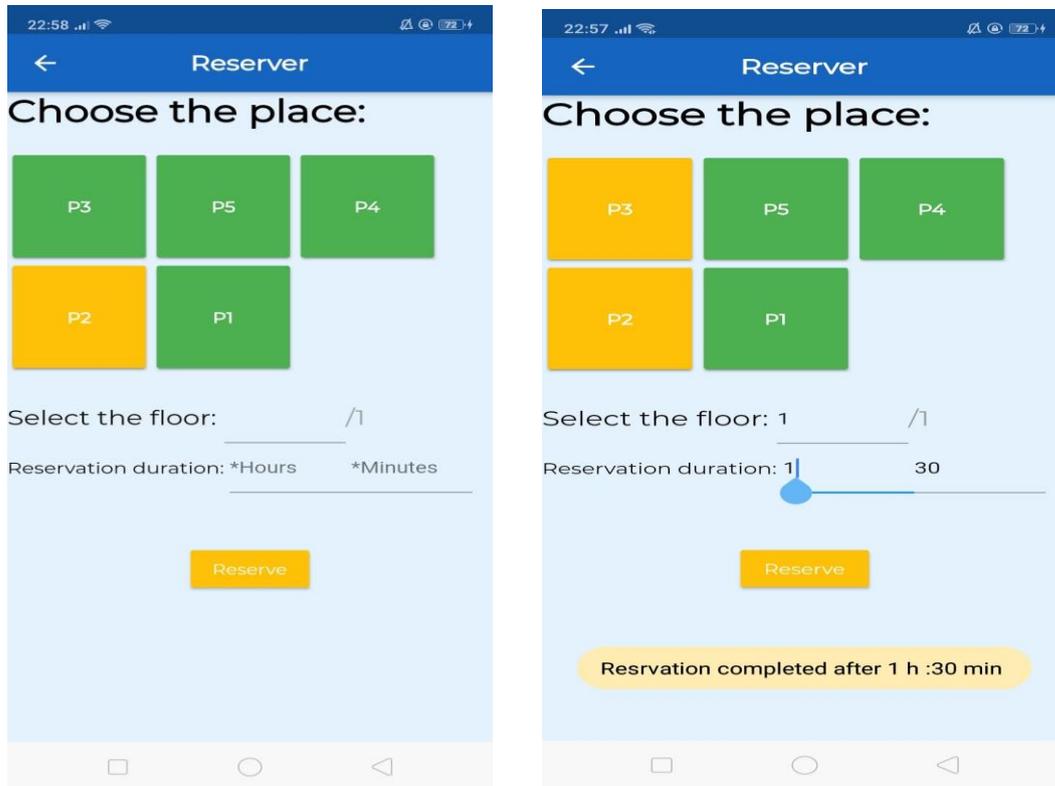


Figure 40: l'interface de réservation d'une place.

B. La deuxième opération : Garer ou bien Stationner :

L'utilisateur peut garer d'après la liste de places disponibles que le système affichera.

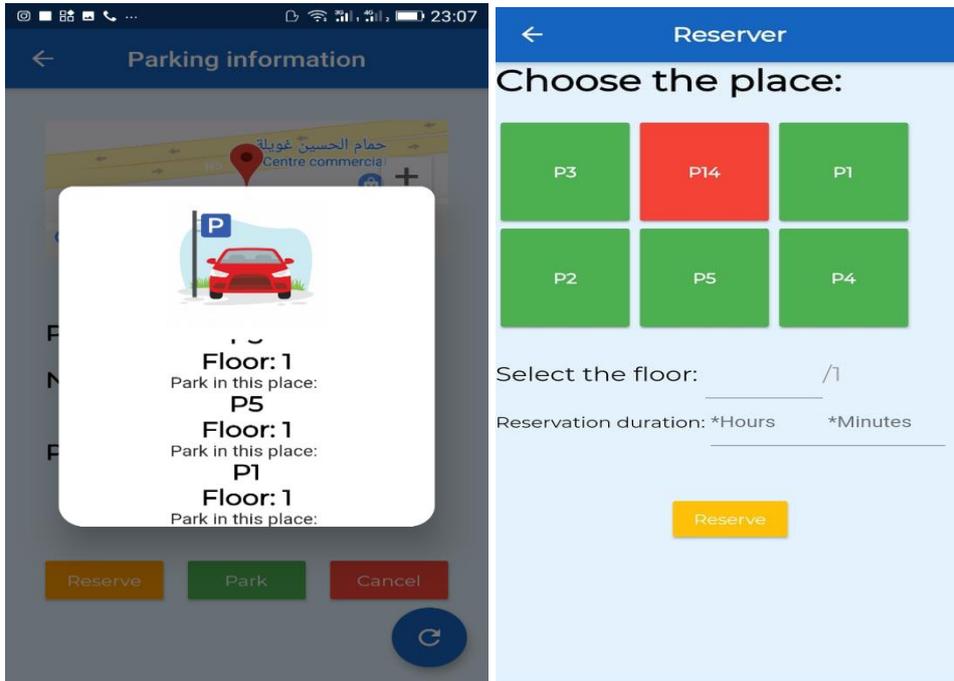


Figure 41: l'interface de l'opération Garer.

C. La troisième opération : Annuler une réservation :

L'utilisateur peut annuler une réservation déjà existante.

L'utilisateur doit confirmer l'annulation.

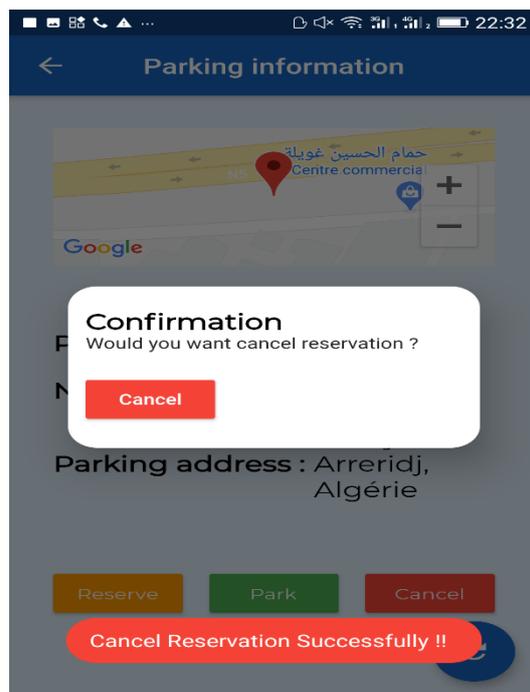


Figure 42 : l'interface de l'annulation d'une réservation.

7 Quelques scénarios d'utilisation de l'application :

7.1 1^{er} scénario :

Lorsque tous les places sont disponibles ou bien libres (état = vert), ici l'utilisateur peut réserver une place pour cela il doit choisi n'importe qu'elle place et remplir les champs et confirmer la réservation (état de la place réserver = jaune).

Dans cette cas l'utilisateur peut garer et le système sera affiché que la place qui était déjà réserver.

Finalement l'utilisateur à la possibilité de l'annuler la réservation qui déjà effectuée d'après la confirmation donc (changement d'état de la place jaune sera vert). Et l'utilisateur reçoit une alerte.

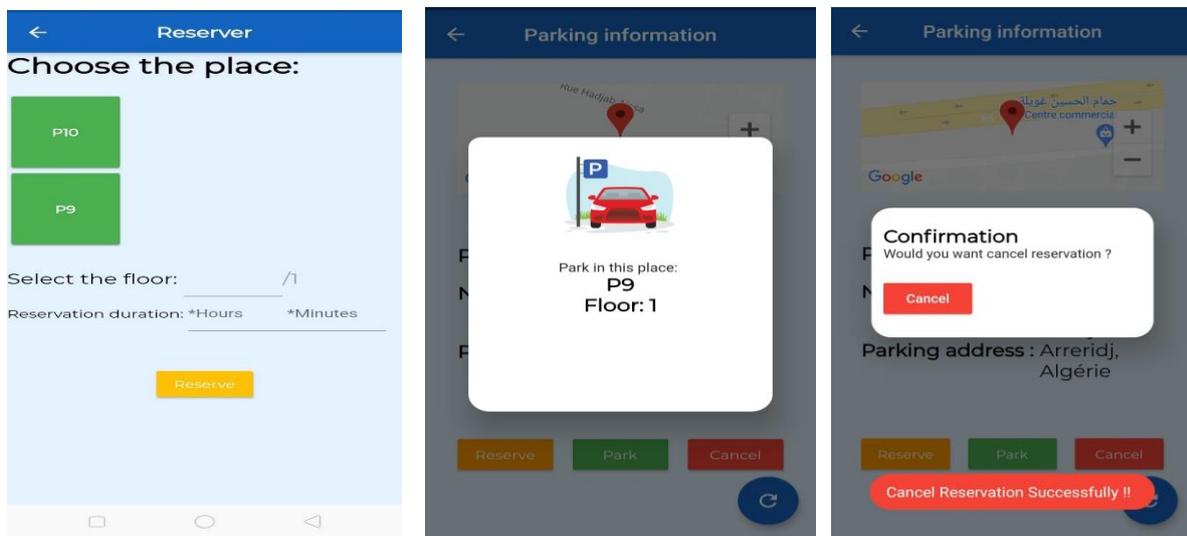


Figure 43: 1^{er} scénario d'utilisation de l'application.

Chapitre 4 : Implémentation.

7.2 2ème scénario :

Dans ce scénario il existe des places libre (état = vert) et autre sont occupées (état = jaune).

Il existe deux utilisateurs qui sont accéder à l'application au même temps.

L'utilisateur 1 : faire une réservation déjà et il vaut garer.

Dans ce cas le système sera affiché que la place qui était déjà réserver. Et cette information (la place réserver) sera affiché aussi sur leur page profil.

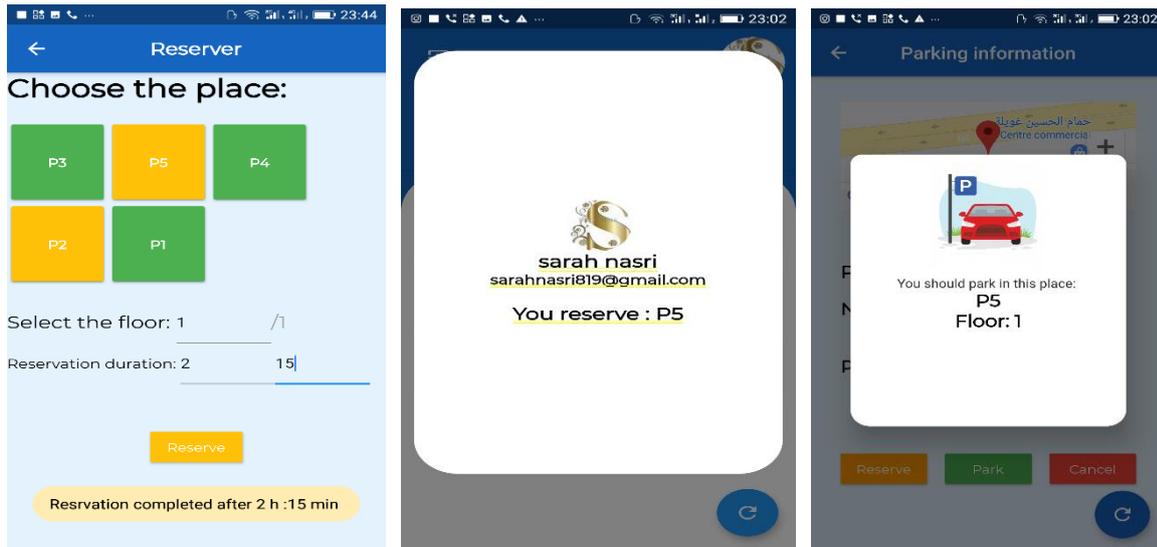


Figure 44: 2ème scénario de l'application "utilisateur1".

Quand la réservation est complet l'utilisateur reçoit une alerte.

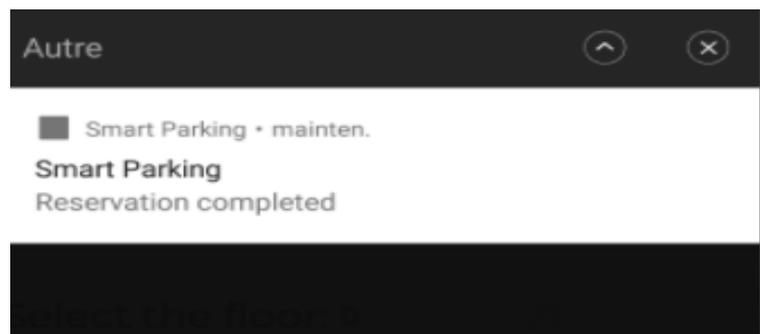


Figure 45: l'alerte de "Réservation terminer".

Chapitre 4 : Implémentation.

L'utilisateur 2 : il vaut garer directement sans réservation alors le système va afficher dans une liste tout les places qui sont disponible (juste les places que leur état = vert).

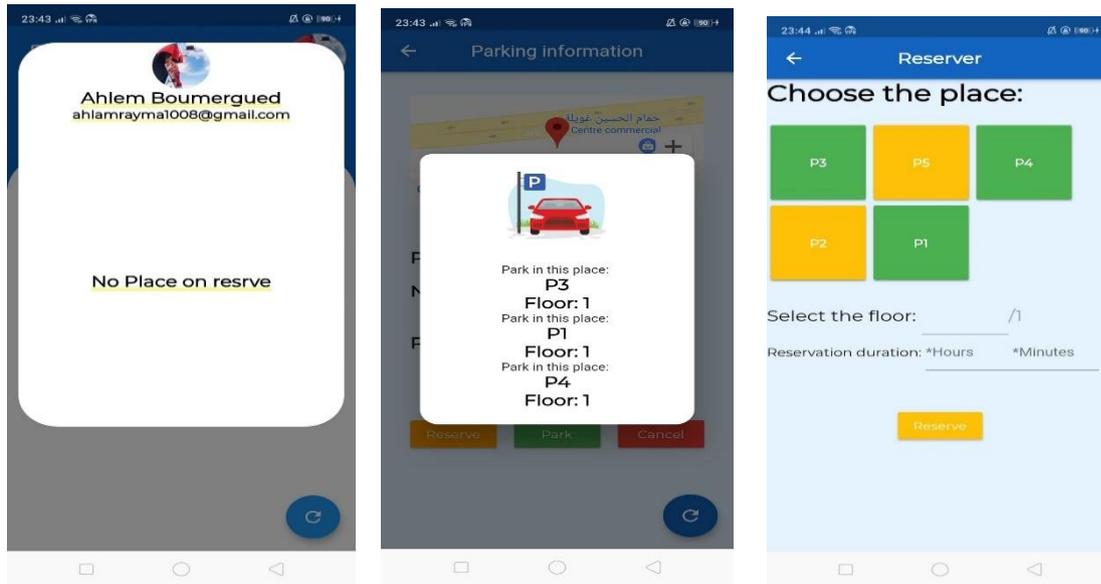


Figure 46: 2ème scénario "L'utilisateur 2".

7.3 3ème scénario :

Dans ce scénario tous les places sont occupées (état =jaune et les places libres = 0)

L'utilisateur 1 : qu'il vaut faire une réservation mais il ne peut pas faire.

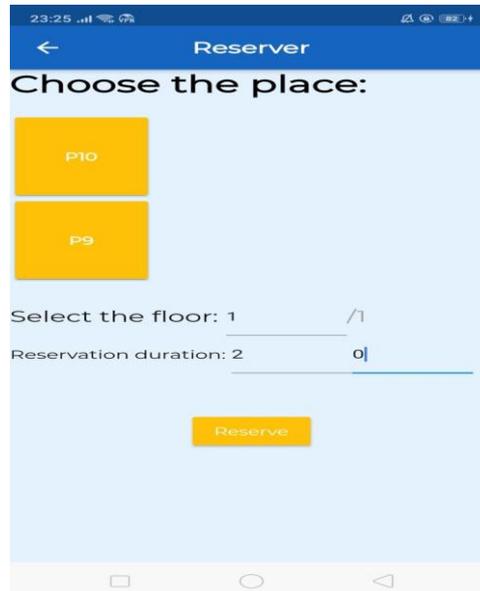


Figure 47 : 3ème scénario "L'utilisateur 1".

Chapitre 4 : Implémentation.

L'utilisateur 2 : qu'il vaut garer directement dans ce cas le système afficher une liste vide car il n'existe aucune place libre dans ce moment.

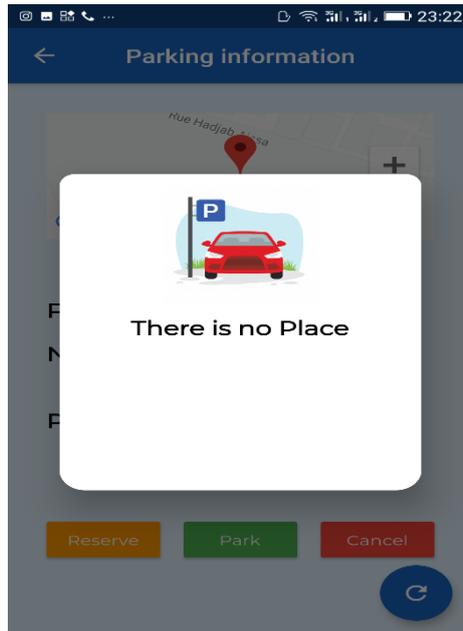


Figure 48 : 3ème scénario "L'utilisateur 2".

8 Conclusion :

Dans ce chapitre, nous avons présenté l'implémentation de notre application. Nous avons tout d'abord expliqué l'architecture de l'application, les différents outils et langage de développement ainsi que les technologies utilisées lors du développement de notre application et qui nous ont permis de réaliser certaines tâches spécifiques. Nous avons aussi présenté les principales interfaces et fenêtres de notre application.

Conclusion générale

Conclusion générale :

Depuis l'aube de l'humanité, l'homme a cherché à trouver et à utiliser des moyens qui lui facilitent l'accomplissement de ses tâches et lui permettent d'accomplir ses fonctions rapidement, facilement et en toute sécurité pour suivre le rythme du développement du monde, en particulier le développement continu de la technologie. Cela est dû à la nature de l'être humain ambitieux qui aime le changement positif dans tous les aspects de la vie.

La technologie dans le monde s'est développée rapidement au cours des dernières années, en particulier dans le domaine des téléphones intelligents, qui est un domaine dynamique de profits élevés et où la concurrence est intensifiée par l'innovation et la créativité de design. Les téléphones et les différentes applications font maintenant partie de notre vie quotidienne.

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser une application permettant d'exploiter dans les villes intelligentes (Smart Cities).

Notre application mobile **SmartParking** est destinée à tous les conducteurs. Elle permet de guider ces conducteurs d'une manière intelligente à trouver un parking possédant une place libre pour stationner et aussi pour réserver une telle place. L'objectif global de l'application est d'aider l'utilisateur à gagner du temps et d'éviter au maximum la circulation dense qui peut provoquer des accidents.

Durant la préparation de notre travail nous avons beaucoup appris. Ce projet nous a permis de valider les connaissances acquises aussi bien sur le plan théorique que sur le plan pratique.

Pour la conception de notre application, nous avons utilisé le langage de modélisation UML, et pour la mise en œuvre, nous avons programmé avec Dart grâce à Flutter et Visual studio code.

Perspectives :

Le travail que nous avons réalisé peut-être amélioré et enrichi pour en faire un système plus performant.

Parmi les perspectives à prendre en compte pour améliorer le fonctionnement de notre application nous citons notamment :

- ✓ Création d'un site web pour cette application.
- ✓ Ajouter d'autres fonctionnalités utiles.
- ✓ Ensuite, envisager d'ajouter cette application dans le Google Play.

9 Bibliographe

- [1] A. & R. Attour, Le rôle des territoires dans le développement des systèmes trans-sectoriels d'innovation locaux : le cas des smart cities. *Innovations*, 1(43), 253-279., 2014.
- [2] «Policy Department - Economic et scientific policy. Mapping smart cities in the eu,,» 2014. [En ligne]. [Accès le 25 mars 2020].
- [3] «Outil d'autodiagnostic de la ville intelligente,» [En ligne]. Available: https://diagnostic.umq.qc.ca/dimensions?fbclid=IwAR3CSvyCbCE_nNAnYOnzrekqc5DnQjUH6T6dJVStyzaE0MC3cmKlt9dsQe0. [Accès le 25 mars 2020].
- [4] «smartcities-caracteristiques,» smartgrids-cre, [En ligne]. Available: <http://www.smartgrids-cre.fr/index.php?p=smartcities-caracteristiques..> [Accès le 26 mars 2020].
- [5] J. FREZAL, «Réflexion sur l'Éthique en intelligence économique.,» [En ligne]. Available: www.lyon-ethique.org/IMG/pdf/intelligenceecofrezal.pdf. [Accès le mars 2020].
- [6] «"Les villes intelligentes" : L'open data contribue-t-il à leur développement ?,» pp. Louvain School of Management. Louvain School of Management, Université catholique de Louvain, Prom. : Belleflamme, Paul., 2015.
- [7] C. Toulon, «LA VILLE INTELLIGENTE Origine, définitions, forces et limites d'une expression polysémique Sandra Breux et Jérémy Diaz».
- [8] «Mobilité intelligente,» [En ligne]. Available: <https://www.mobilite-intelligente.com/politiques-publiques/mobilite3>. [Accès le 25 mars 2020].
- [9] L. P. M. S. R. V. L. Mainetti, «A Smart Parking System based on IoT protocols and emerging enabling technologies,» [En ligne]. Available: <https://www.semanticscholar.org/paper/A-Smart-Parking-System-based-on-IoT-protocols-and-Mainetti-Patrono/507c70d71b985548ec5787910283ddaad3d9b608/figure/0?fbclid=IwAR01Di4pNk7-ksRdl-400-1ZUQQ5ckrMcHUwu3cpmFKFT-sU1sMm1niE400..> [Accès le avril 2020].
- [1] «application mobile,» juillet 2015. [En ligne]. Available: <https://www.taktillcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html>. [Accès le 3 aout 2020].
- [1] D. Rosul, «thinkmobiles,» [En ligne]. Available: <https://thinkmobiles.com/blog/popular-types-1-of-apps/>. [Accès le aout 2020].
- [1] «Zenpark blog,» Zenpark, [En ligne]. Available: <https://blog.zenpark.com/zenpark-lance-son-service-de-parking-partage/?fbclid=IwAR2GOzmVK4tDgvYirAKSL4wz1Ep4A6ItJy5eEa6LWdD-uGtGHnOWdCqrBXM>. [Accès le aout 2020].

- [1 «Passport Parking Canada,» [En ligne]. Available:
3] https://play.google.com/store/apps/details?id=ca.passportparking.mobile.passportcanada&hl=en_CA. [Accès le aout 2020].
- [1 «Les systèmes d'exploitation des smartphones,» [En ligne]. Available:
4] <https://www.monpetitmobile.com/choisir-mobile/systemes-exploitation-smartphones>. [Accès le aout 2020].
- [1 G. Picard, *Un système d'exploitation open source*, Mines, 2012.
5]
- [1 «CCM/Système d'exploitation,» [En ligne]. Available:
6] <https://www.commentcamarche.net/contents/1092-systeme-d-exploitation>. [Accès le aout 2020].
- [1 «LE CYCLE DE DEVELOPPEMENT EN V,» [En ligne]. Available: http://www-igm.univ-mlv.fr/~dr/XPOSE2002/Site_Vaubourg_Stephane_IR3/html/cycleV.htm. [Accès le 2020].
7]
- [1 M. Rouse, «Architecture client/serveur (ou modèle client/serveur),» [En ligne]. Available:
8] <https://whatis.techtarget.com/fr/definition/Architecture-client-serveur-ou-modele-client-serveur>. [Accès le aout 2020].
- [1 J.-F. Pillou.. [En ligne]. Available: <https://www.commentcamarche.net/contents/222-environnement-client-serveur>. [Accès le 22 aout 2020].
9]
- [2 [En ligne]. Available: <https://dart.dev/>. [Accès le 18 aout 2020].
0]
- [2 L. Tung. [En ligne]. Available: <https://www.zdnet.com/article/googles-flutter-1-20-framework-is-out-vs-code-extension-and-mobile-autofill-support/>. [Accès le 20 aout 2020].
1]
- [2 «Flutter Tutorial,» [En ligne]. Available: <https://www.tutorialspoint.com/flutter/index.htm>.
2] [Accès le aout 2020].
- [2 «junto,» [En ligne]. Available: <https://junto.fr/blog/firebase/>. [Accès le 22 aout 2020].
3]
- [2 «Introduction à Firebase,» [En ligne]. Available: <https://lesveilleursdenuit.fr/introduction-a-firebase/>. [Accès le 23 aout 2020].
4]
- [2 [En ligne]. Available:
5] <https://www.01net.com/telecharger/windows/Programmation/creation/fiches/130819.html>. [Accès le aout 2020].

Résumé

Ce travail présenté dans le cadre du mémoire de deuxième année Master, vise à développer une application mobile de stationnement intelligent sur Android, en tenant compte du potentiel de déploiement au niveau de la ville intelligente.

Cette application permet aux conducteurs d'accéder aux informations de stationnement sur leurs appareils mobiles, et de réserver en ligne une place pour stationner afin de réduire les embouteillages dans les villes intelligentes.

Les mots clés : Application mobile, parking intelligent, Android, Villes intelligentes.

ملخص

يهدف هذا العمل، الذي تم تقديمه كجزء من رسالة السنة الثانية الماجستير، إلى تطوير تطبيق للأجهزة المحمولة لموقف السيارات الذكي على نظام الأندرويد.

يسمح هذا التطبيق للسائقين بالوصول إلى معلومات الركن على أجهزتهم المحمولة، وحجز موقف عبر الإنترنت بهدف الحد من حالات انحصار المرور في المدن الذكية.

الكلمات المفتاحية

موقف السيارات الذكي تطبيق الهواتف الذكية نظام الأندرويد المدن الذكية.

Summary

This work, which was presented as part of the second-year master's thesis, aims to develop a mobile application for intelligent parking on Android, taking into account the potential of deployment at the smart city level.

This app allows drivers to access parking information on their mobile devices, and book online a parking spot for the purpose of reducing traffic jams in smart cities.

Keywords: Mobile app, smart parking, Android, Smart cities.