



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



Université d'El Oued

FACULTÉ DES SCIENCES EXACTES

Mémoire de fin d'études

DIPLÔME DE LICENCE ACADÉMIQUE

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Informatique Fondamentale

Présenté par : Mohammed Ali Atoussi

Mohammed Yacine Belloum

Theme

Conception et réalisation une application
ANDROID pour la gestion des parkings

Encadrante : Dr. Maamra Oum Elhana

Année universitaire 2024 - 2025

ملخص

مع تزايد أعداد السيارات وصعوبة إيجاد مواقف شاغرة في المدن، ظهرت الحاجة إلى حلول ذكية تسهّل عملية البحث والحجز وتوفّر الوقت والجهد. يهدف هذا المشروع إلى تطوير تطبيق يتيح للمستخدمين استعراض المواقف المتاحة في الوقت الفعلي، حجزها إلكترونياً، والدفع بسهولة عبر وسائل إلكترونية آمنة. يعتمد التطبيق على تقنيات حديثة مثل الخرائط الذكية وأجهزة الاستشعار، ما يساهم في تقليل الازدحام وتحسين تجربة القيادة وتنظيم مواقف السيارات

Résumé

Avec l'augmentation du nombre de voitures et la difficulté de trouver des places libres en ville, il devient essentiel de proposer des solutions intelligentes pour faciliter la recherche et la réservation. Ce projet vise à développer une application permettant aux utilisateurs de consulter en temps réel les places disponibles, de les réserver et de payer facilement en ligne. L'application utilise des technologies modernes comme les cartes intelligentes et les capteurs, contribuant à réduire la congestion et à améliorer la gestion du stationnement.

Abstract

With the increasing number of cars and the difficulty of finding available parking in cities, there is a growing need for smart solutions to simplify the search and reservation process. This project aims to develop a mobile app that allows users to view available parking spots in real time, book them electronically, and pay easily through secure online methods. The app leverages modern technologies such as smart maps and sensors, helping reduce congestion and improve parking management and the overall driving experience.

Liste des figures

1.1 Parking de surfacé	10
1.2 Parking souterrain	10
1.3 Parking a étages	11
1.4 Parking en tour automatisée	11
2.1 Méthodologie 2TUP	17
3.1 Diagramme global	20
3.2 Diagramme de cas d'utilisation Authentification	21
3.3 Diagramme de séquence Authentification	21
3.4 Diagramme de cas d'utilisation Réservation	22
3.5 Diagramme de séquence Réservation	23
3.6 Diagramme de cas d'utilisation Paiement	24
3.7 Diagramme de séquence Paiement	25
3.8 Diagramme de cas d'utilisation Consultation.	26
3.9 Diagramme de séquence Consultation	28
3.10 Diagramme de cas d'utilisation Gestion Parking.	29
3.11 Diagramme de séquence Gestion Parking.	29
3.12 Diagramme de cas d'utilisation Gestion des Tarifs	30
3.13 Diagramme de séquence Gestion des Tarifs	30
3.14 Diagramme de cas d'utilisation Consultation des Commentaires	31
3.15 Diagramme de séquence Consultation des Commentaires	32
3.16 Diagramme de Classe	33
4.1 Interface principal	37
4.2 Présentation des détails du parking	38
4.3 réservation	39
4.4 d'affichage du paiement	40
4.5 d'affichage réservation	41
4.6 d'affichage du code QR	41
4.7 d'affichage portefeuille	42

4.8 Interface d'administration pour l'ajout des parkings	43
--	----

Table des matières

Theme.....	0
Table des matières.....	6
1 .Introduction.....	10
2 .Définition du parking.....	10
3 .Types de parkings.....	10
3.1 .Parking en surface.....	11
3.2 .Parking souterrain.....	11
3.3 .Parking à étages (silo).....	12
3.4 .Parking en tour automatisée.....	12
4 .La gestion de parking.....	13
4.1 .Défis.....	13
4.2 Solutions traditionnelles :.....	13
4.3 Solutions numériques modernes.....	14
5 .Les Technologies Mobiles Pour les Applications.....	15
5.1 .Définition :.....	15
5.2 .Importance des applications Android.....	15
5.2.1 .Diversité des appareils.....	15
5.2.2 .Coût de développement abordable.....	15
5.2.3 .Flexibilité et personnalisation.....	16
5.2.4 .Intégration avec l'écosystème Google.....	16
5.2.5 .Distribution facile.....	16
5.2.6 .Innovation continue.....	16
6 .Analyse des Application de Stationnement existantes.....	16
6.1 . ParkMobile.....	16
6.1.1 .Explication courte de ParkMobile.....	16
6.1.2 .Les défauts de l'application.....	17
6.2 . JustPark.....	17
6.2.1 .Explication courte de JustPark.....	17
6.2.2 .Les défauts de JustPark :.....	17
7 .Conclusion.....	18
1 .Introduction.....	19
2 .Processus 2TUP.....	19
2.1 Définition.....	19
2.2 Spécification des exigences.....	20
• Les exigences fonctionnelles.....	20
• Les exigences non fonctionnelles (techniques).....	21
3 .UML.....	21
3.1 Définition.....	21
3.2 Identifier les cas d'utilisation.....	23

3.2.1 Diagramme global.....	23
3.3 Détail des cas d'utilisation.....	24
3.3.1 Diagramme de Authentification.....	24
3.3.2 Diagramme de Réservation.....	26
3.3.3 Diagramme de Paiement.....	28
3.3.4 Diagramme de Consultation.....	30
3.3.5 Diagramme de Gestion des Parkings.....	32
3.3.6 Diagramme de Gestion des Tarifs.....	33
3.3.7 Diagramme de Consultation des Commentaires.....	34
3.4 Diagramme de classe.....	36
1.Introduction :.....	38
2.Environnement de développement :.....	38
3. Langages de programmation utilisés.....	39
3.1. React Native et le langage JavaScript :.....	39
3.2. Supabase pour le backend :.....	39
4.Présentation de l'application.....	40
4.1. Interface principale.....	41
4.2. Présentation des détails du parking.....	42
4.3. Écran de réservation.....	43
4.5. Écran d'affichage du paiement.....	44
4.6 Écran d'affichage du code QR.....	45
4.7 Écran du portefeuille.....	46
4.8 Interface d'administration pour l'ajout des parkings.....	47

I. Introduction générale

Le stationnement est devenu un vrai problème dans les villes modernes à cause du nombre croissant de voitures et du manque d'espaces disponibles. Les solutions classiques, comme les parcmètres et les parkings traditionnels, ne suffisent plus à répondre aux besoins des conducteurs. De plus, les applications existantes présentent souvent des défauts : elles sont compliquées à utiliser, manquent de fiabilité, ou ne protègent pas assez les données des utilisateurs.

Pour répondre à ces défis, notre projet propose de créer une application mobile simple et efficace pour réserver une place de parking. L'objectif principal est de faciliter la vie des utilisateurs : ils pourront trouver rapidement une place libre, réserver en quelques clics, payer en toute sécurité, et accéder facilement au parking grâce à un code QR.

Notre travail s'organise en trois parties principales :

Dans le premier chapitre, nous présentons le contexte du stationnement en ville, les différents types de parkings, et les problèmes rencontrés aujourd'hui. Nous analysons aussi les applications existantes pour comprendre leurs limites.

Le deuxième chapitre explique la méthode de travail utilisée pour concevoir l'application. Nous montrons, à l'aide de schémas simples, comment l'application fonctionne et comment les données circulent entre les différents utilisateurs et le système.

Le dernier chapitre décrit la réalisation concrète de l'application : le choix des technologies, la création de l'interface utilisateur, la gestion de la base de données, et la sécurité des informations. Nous présentons aussi les écrans principaux de l'application pour montrer son côté pratique et intuitif.

En résumant, ce projet vise à offrir une solution moderne et accessible pour le stationnement, en simplifiant la réservation et en améliorant l'expérience des utilisateurs. Nous terminons par une conclusion générale qui résume les résultats obtenus et propose des pistes pour de futurs développements.

Chapitre 1 : État de l'art

1 .Introduction

Le stationnement est un défi majeur dans les villes modernes, où la demande croissante de places entre en conflit avec des objectifs tels que la lutte contre l'étalement urbain et la promotion des transports en commun. Les avancées technologiques ont permis l'émergence de systèmes intelligents pour optimiser la gestion des parkings et répondre aux besoins des usagers. [1] Cependant, certains pays, comme l'Algérie, restent en retard par rapport aux normes internationales.

Ce projet s'inscrit dans cette optique en proposant un système intelligent de gestion des parkings, reposant sur des solutions diversifiées et adaptées. L'objectif est d'étudier et de concevoir une approche globale qui optimise l'utilisation des espaces, améliore l'expérience des usagers et s'adapte aux spécificités locales et internationales.

2 .Définition du parking

Le parking désigne un espace dédié à l'arrêt temporaire des véhicules, principalement des automobiles, pendant une durée déterminée. Ces espaces, souvent clôturés ou délimités, jouent un rôle essentiel dans l'infrastructure urbaine moderne [2].

On les retrouve dans divers lieux tels que les aéroports, les hôpitaux, les grandes surfaces et les marchés.

Le stationnement joue également un rôle vital dans la mobilité urbaine, l'accessibilité des services et la gestion de l'espace public. Il contribue significativement à la qualité de vie en milieu urbain en permettant aux conducteurs de laisser leur véhicule en sécurité tout en facilitant leurs déplacements et leurs activités quotidiennes.

3 .Types de parkings

3.1 .Parking en surface

Situé à l'extérieur, de plain-pied, sur un espace public ou privé, il comprend des places le long des rues ou sur des terrains dégagés. Simple à aménager, il nécessite toutefois une grande surface.



figure (1.1) : parking de surfacé

3.2 .Parking souterrain

Construit sous les bâtiments ou infrastructures, il économise l'espace en ville et s'étend sur plusieurs niveaux reliés par des rampes. Il nécessite des mesures spécifiques comme la ventilation et la sécurité [3].

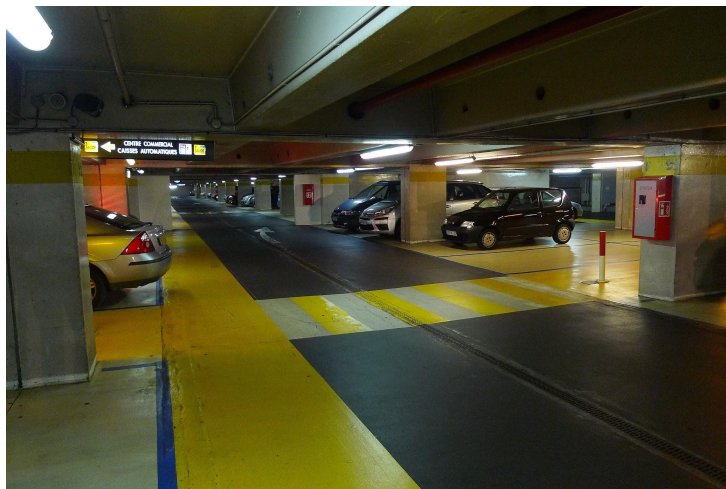


figure (1.2) : parking souterrain

3.3 .Parking à étages (silo)

Bâtiment extérieur à plusieurs niveaux reliés par des rampes, iconstitue une alternative aux parkings souterrains sans nécessiter de creusement.



figure (1.3) : parking a étages

3.4 .Parking en tour automatisée

Parking vertical compact où un système automatisé empile les voitures sur plusieurs niveaux, idéal pour les zones à espace limité.



figure (1.4) : parking en tour automatisée

4 .La gestion de parking

La gestion de parking est un aspect crucial de l'aménagement urbain et de la mobilité, qui soulève de nombreux enjeux et problématiques. Voici une analyse de la définition, des enjeux et des solutions traditionnelles et numériques

4.1 .Défis

L'augmentation rapide du nombre de véhicules et la densification urbaine rendent la gestion du stationnement plus complexe et soulèvent de nouveaux défis pour les villes modernes

- **Optimisation de l'espace urbain limité** : La croissance démographique et l'augmentation du nombre de véhicules exercent une pression importante sur les espaces disponibles.
- **Réduction de la congestion et des émissions** : La recherche prolongée d'un stationnement contribue aux embouteillages et à la pollution.
- **Accessibilité et qualité de vie** : Les parkings doivent être accessibles, sûrs et bien organisés pour améliorer l'expérience des usagers.
- **Manque d'intégration technologique** : L'absence de solutions modernes complique la gestion efficace des espaces.
- **Stationnement illégal** : Les pratiques non réglementées perturbent l'ordre public et réduisent l'efficacité des infrastructures existantes.

4.2 Solutions traditionnelles :

- **Zones réglementées** : Mise en place de parcmètres, horodateurs ou permis de stationnement pour contrôler l'accès.

- **Augmentation de l'offre :** Construction de nouveaux parkings pour répondre à la demande croissante.

Ces approches, bien que efficaces dans certains contextes, manquent souvent de flexibilité et d'adaptabilité face aux défis modernes.

4.3 Solutions numériques modernes

- **Capteurs IoT et données en temps réel :** Permettent de surveiller l'occupation des places et d'informer les conducteurs via des applications mobiles ou des panneaux numériques.
- **Paielement sans contact :** Simplifie les transactions grâce à des applications mobiles, QR codes ou cartes NFC.
- **Systèmes automatisés :** Utilisation de caméras ANPR (reconnaissance automatique des plaques) pour gérer les accès et appliquer les règles en temps réel.
- **Analyse prédictive et IA :** Prévoit les besoins futurs en stationnement pour optimiser la planification et réduire les inefficacités.
- **Parkings intelligents intégrés aux Smart Cities :** Connectés aux réseaux urbains pour une gestion globale du trafic et une réduction des impacts environnementaux.

5 .Les Technologies Mobiles Pour les Applications

5.1 .Définition :

Avec l'essor des smartphones et des terminaux mobiles, ces dispositifs sont devenus incontournables pour simplifier notre vie quotidienne. Cette popularité croissante a entraîné une évolution constante des systèmes d'exploitation et des applications mobiles[4].

Une application mobile est un logiciel conçu pour être installé sur un appareil mobile (smartphone, tablette, etc.), qu'elle soit gratuite ou payante. Elle permet d'accéder à divers services pratiques, informations multimédias ou contenus de divertissement.

Les usages des applications mobiles sont variés et touchent presque tous les domaines : réseaux sociaux, santé, actualités, jeux, banque en ligne, GPS, shopping, météo, musique, et bien plus encore. Ces applications sont généralement accessibles via des plateformes de téléchargement comme l'App Store pour iOS ou Google Play pour Android.

5.2 .Importance des applications Android

Les applications Android jouent un rôle crucial dans l'écosystème mobile actuel pour plusieurs raisons :

5.2.1 .Diversité des appareils

Android est compatible avec une large gamme de smartphones et de tablettes provenant de diverses marques. Cette diversité permet de toucher des utilisateurs ayant des budgets et des préférences variés.

5.2.2 .Coût de développement abordable

Les outils de développement pour Android sont souvent gratuits et open-source, ce qui facilite l'accès au développement d'applications pour les petites entreprises et les développeurs indépendants.

5.2.3 .Flexibilité et personnalisation

Android permet une plus grande liberté dans la personnalisation des applications. Les développeurs ont la possibilité de modifier en profondeur l'interface utilisateur et d'exploiter de nombreuses fonctionnalités du système.[5]

5.2.4 .Intégration avec l'écosystème Google

Les applications Android s'intègrent aisément aux services populaires de Google tels que Gmail, Google Maps ou Google Drive, offrant ainsi des expériences utilisateur plus riches et mieux interconnectées.

5.2.5 .Distribution facile

Le Google Play Store propose une plateforme de distribution vaste et facilement accessible. Les développeurs ont également la possibilité de diffuser leurs applications par d'autres moyens ou directement auprès des utilisateurs.

5.2.6 .Innovation continue

Le caractère open-source d'Android favorise l'innovation et l'expérimentation. De plus, les nouvelles fonctionnalités sont fréquemment disponibles sur Android avant les autres plateformes.

Cette importance fait des applications Android un élément essentiel de la stratégie mobile pour de nombreuses entreprises et développeurs.

6 .Analyse des Application de Stationnement existantes

6.1 . ParkMobile

6.1.1 .Explication courte de ParkMobile

ParkMobile est une application pour payer le stationnement avec le téléphone. Elle permet de payer dans la rue, réserver une place, prolonger le temps à distance et recevoir des alertes. Elle fonctionne dans plus de 400 villes aux États-Unis et accepte les cartes bancaires ou PayPal. On entre le numéro de la zone pour l'utiliser.

6.1.2 .Les défauts de l'application

- **Problèmes de connexion** : L'application a besoin d'une bonne connexion internet. S'il n'y en a pas, elle ne fonctionne pas bien, et on peut recevoir une amende.
- **Erreurs dans les données** : Si on se trompe en entrant le numéro de la voiture ou de la zone, on risque une amende, et c'est difficile à corriger.
- **Politique de remboursement** : On ne peut pas facilement récupérer son argent si on a payé pour une place qu'on n'a pas utilisée.
- **Problème de sécurité** : En 2021, des données des utilisateurs ont été piratées, ce qui a fait baisser la confiance .
- **Interface compliquée** : Certains trouvent l'application difficile à utiliser, surtout pour changer de voiture ou de moyen de paiement.
- **Problèmes avec les autorités** : Parfois, il y a des erreurs avec les autorités locales, comme des amendes injustes.
- **Support faible** : Le service client est lent et ne résout pas les problèmes rapidement.

6.2 . JustPark

6.2.1 .Explication courte de JustPark

JustPark est une application qui aide les conducteurs à trouver et réserver des places de stationnement via leur téléphone. Elle propose des millions d'espaces, publics ou privés (comme des allées), dans plus de 45 000 lieux au Royaume-Uni. On peut réserver à l'avance, payer via l'application (carte bancaire, PayPal, etc.), et même prolonger le temps facilement. Elle donne des infos en temps réel sur les prix et la disponibilité.

6.2.2 .Les défauts de JustPark :

- **Connexion internet nécessaire** : Sans réseau, l'application ne fonctionne pas bien, ce qui peut poser problème dans les zones mal couvertes.
- **Erreurs de réservation** : Si on entre de mauvaises infos (numéro de voiture ou lieu), c'est difficile à corriger, et on risque une amende.

- **Remboursement compliqué** : Récupérer son argent pour une place non utilisée est souvent difficile.
- **Sécurité des données** : En 2015, des données ont été piratées, ce qui inquiète certains utilisateurs.
- **Interface pas toujours simple** : Certains trouvent l'application compliquée, surtout pour gérer plusieurs véhicules ou paiements.
- **Dépendance aux propriétaires** : La disponibilité dépend des espaces proposés par les particuliers ou entreprises, pas toujours fiables.
- **Service client lent** : Les réponses aux problèmes techniques ou aux plaintes peuvent prendre du temps.

7 .Conclusion

Ce chapitre a exploré l'état de l'art du stationnement, un défi clé dans les villes modernes. Il a défini le parking, ses types (surface, souterrain, étages, automatisé), et analysé sa gestion face à des enjeux comme l'espace limité et la pollution, avec des solutions traditionnelles et numériques (IoT, IA, applications). Les technologies mobiles, notamment Android, jouent un rôle majeur grâce à leur flexibilité. L'analyse de ParkMobile et JustPark a révélé leurs avantages mais aussi leurs défauts (connexion, sécurité, interface). Ce travail pose les bases pour un système intelligent adapté aux besoins locaux et internationaux.

Chapitre 2 : CONCEPTION

1 .Introduction

Dans un environnement informatique de plus en plus complexe, l'adoption de méthodologies structurées et d'outils de modélisation efficaces est devenue essentielle pour le développement logiciel. Ce chapitre aborde deux éléments fondamentaux et complémentaires: le processus 2TUP (2 Track Unified Process) et le langage UML (Unified Modeling Language). Le 2TUP propose une approche en Y qui sépare les aspects techniques des considérations fonctionnelles, permettant une gestion optimisée du développement. UML, quant à lui, offre un ensemble standardisé de notations graphiques pour modéliser les différentes dimensions d'un système informatique. L'intégration de ces deux composantes permet aux organisations de structurer efficacement leurs projets tout en assurant une documentation précise et une communication claire entre tous les intervenants. Cette synergie entre méthodologie de développement et langage de modélisation constitue un atout majeur pour répondre aux exigences de qualité, optimiser les coûts et respecter les délais dans la réalisation de solutions informatiques.

2 .Processus 2TUP

2.1 Définition

Le 2TUP (2 Track Unified Process) est un processus de développement en forme de Y qui sépare les aspects techniques des aspects fonctionnels. Cette approche permet de traiter parallèlement deux dimensions du développement: les besoins métier et les contraintes techniques. Le processus comprend trois branches principales: la branche fonctionnelle (gauche) qui capture les besoins métier, la branche technique (droite) qui gère les aspects technologiques, et la branche centrale qui représente la fusion des deux premières lors de la phase de réalisation [6].

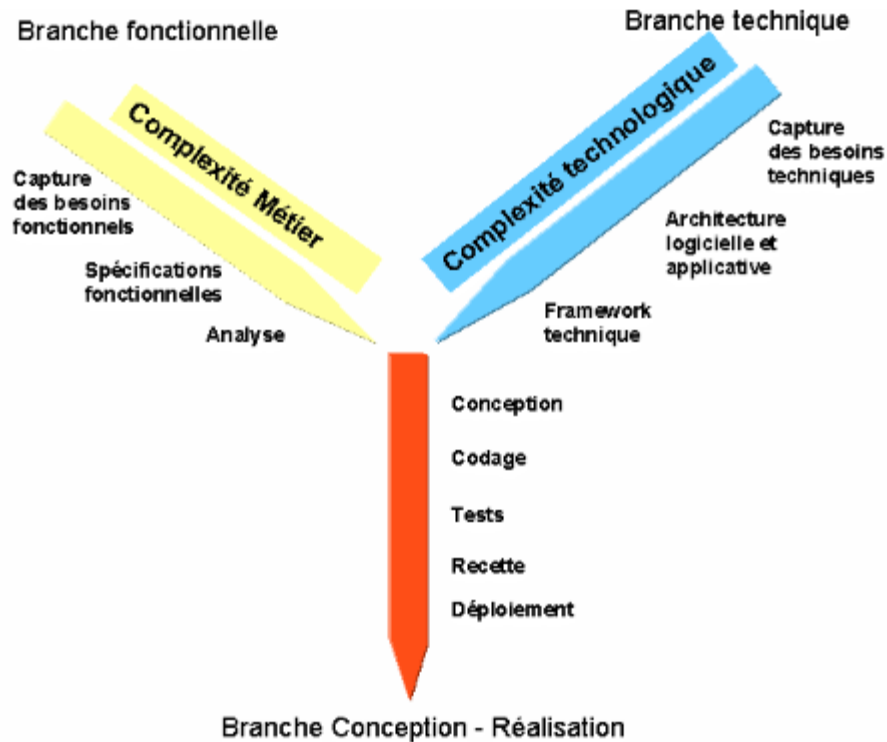


figure (2.1) : Méthodologie 2TUP

2.2 Spécification des exigences

Dans le cadre méthodologique du processus 2TUP (2 Track Unified Process), la spécification des exigences se structure autour de deux catégories fondamentales qui sont analysées parallèlement lors du développement d'un système informatique

- Les exigences fonctionnelles

Ces exigences constituent l'ensemble des spécifications que notre système doit satisfaire pour répondre efficacement aux besoins des utilisateurs. Elles représentent concrètement la collection des services et fonctionnalités que notre application doit proposer. En d'autres termes, elles décrivent précisément ce que le système doit faire et les comportements qu'il doit adopter face aux différentes interactions avec ses utilisateurs.

- Les exigences non fonctionnelles (techniques)

Ces exigences regroupent l'ensemble des spécifications qui décrivent les capacités techniques et les contraintes opérationnelles auxquelles le système est soumis. Contrairement aux exigences fonctionnelles qui définissent les services fournis, les exigences non fonctionnelles déterminent les conditions dans lesquelles ces services doivent être délivrés.

Parmi ces exigences techniques essentielles figurent notamment :

- La sécurité (protection des données et authentification)
- L'environnement d'exploitation (compatibilité avec différentes plateformes)
- La maintenance (facilité de mise à jour et d'évolution)
- La fiabilité (fonctionnement sans interruption ni défaillance)

Ces deux types d'exigences, analysés dans les branches distinctes du Y caractéristique du 2TUP, convergent ultérieurement dans la phase de conception et de réalisation pour produire un système cohérent et performant .

3 .UML

3.1 Définition

L'UML (Unified Modeling Language) est un langage de modélisation standardisé qui permet de représenter graphiquement la structure et le comportement d'un système logiciel. Selon les standards actuels, il existe treize types de diagrammes UML : classe, activité, objet, cas d'utilisation, séquence, paquetage, état, composant, communication, structure composite, vue d'interaction, timing et déploiement [7].

Pour la conception de notre système, seuls trois types de diagrammes sont nécessaires

- Diagramme de cas d'utilisation (Use Case Diagram)
- Diagramme de séquence (Sequence Diagram)
- Diagramme de classes (Class Diagram)

Définitions simples des diagrammes utilisés :

- **Diagramme de cas d'utilisation (Use Case Diagram)**

Ce diagramme permet de représenter les différentes interactions possibles entre les utilisateurs (appelés acteurs) et le système. Il décrit les fonctionnalités principales du système du point de vue de l'utilisateur, sans entrer dans les détails techniques.

- **Diagramme de séquence (Sequence Diagram)**

Ce diagramme illustre comment les objets du système interagissent entre eux au cours du temps pour réaliser une fonctionnalité précise. Il met en évidence l'ordre des échanges de messages entre les différents composants du système.

- **Diagramme de classes (Class Diagram)**

Ce diagramme présente la structure statique du système en modélisant les classes, leurs attributs, leurs méthodes et les relations entre elles. Il permet de visualiser les principaux objets du système et la façon dont ils sont liés

3.2 Identifier les cas d'utilisation

3.2.1 Diagramme global

Dans la figure suivante, nous présentons notre diagramme de cas d'utilisation global :

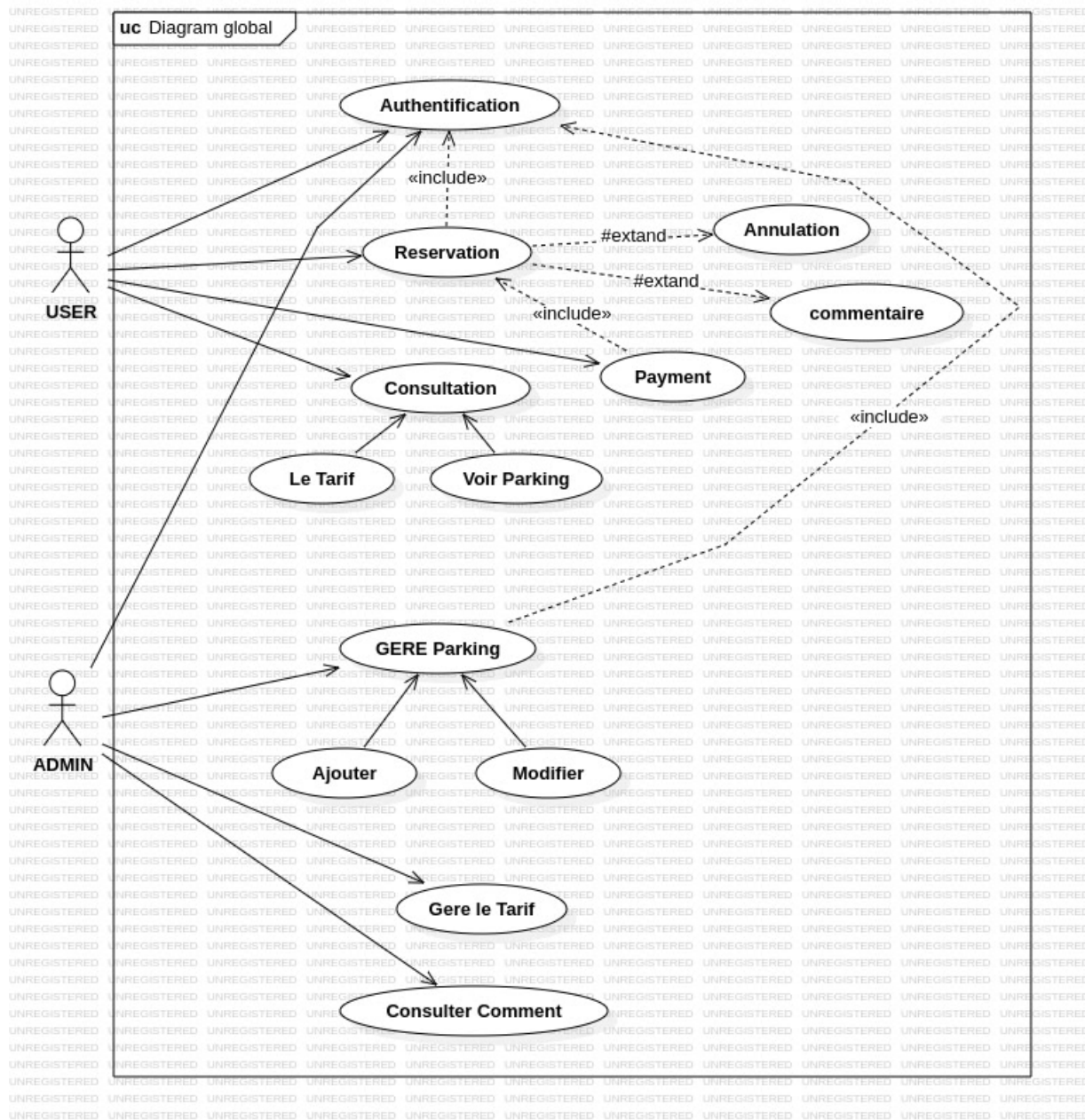


figure (3.1) : diagramme global

3.3 Détail des cas d'utilisation

3.3.1 Diagramme de Authentification

- L'utilisateur doit se connecter ou créer un compte pour utiliser l'application de réservation de stationnement.
- Quand il saisit ses informations de connexion, si elles sont correctes il accède à la page principale, sinon un message d'erreur s'affiche.

Diagramme de cas d'utilisation :

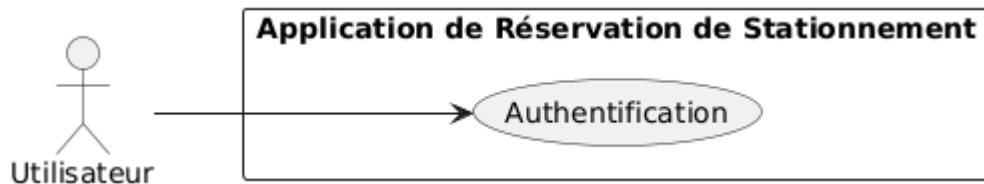


figure (3.2) : diagramme de cas d'utilisation Authentification

Diagramme de séquence :

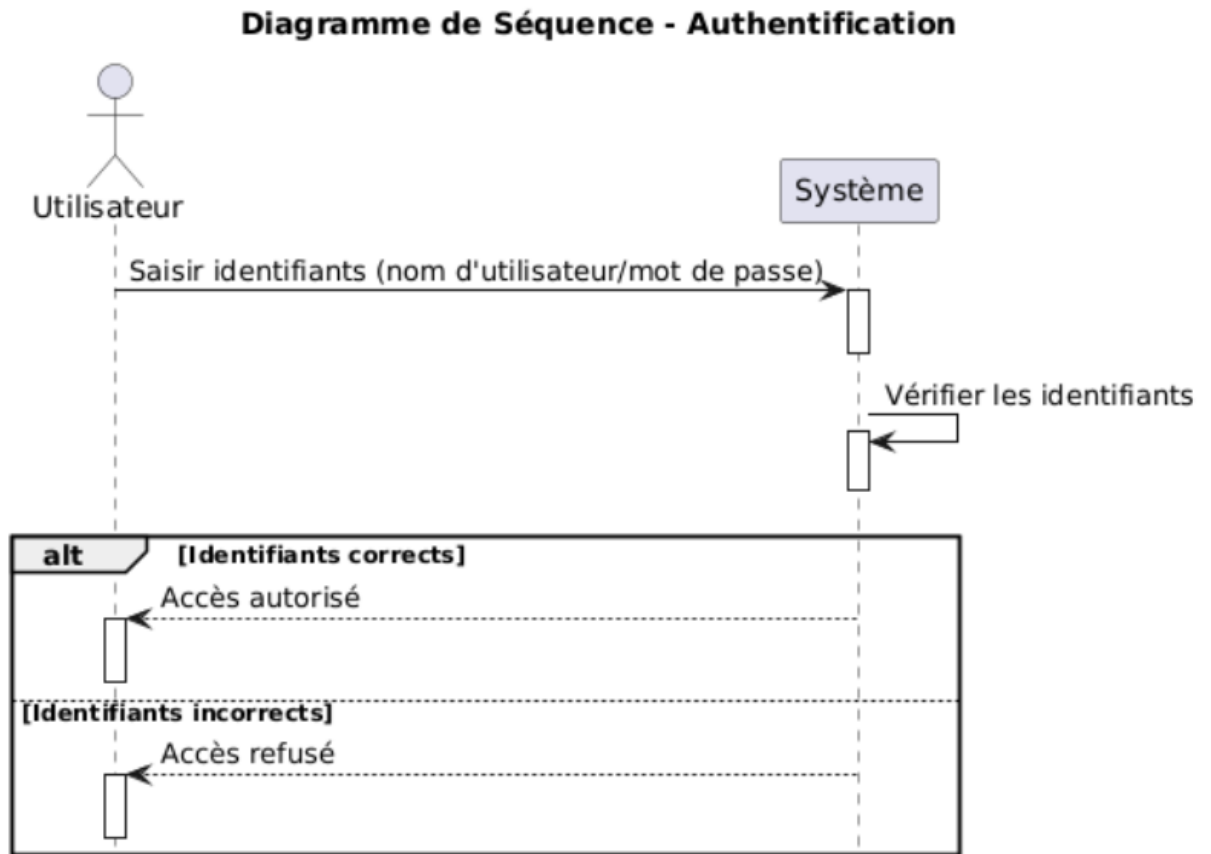


figure (3.3) : diagramme de séquence Authentification

3.3.2 Diagramme de Réservation

- L'utilisateur doit demander une réservation via l'application.
- Lorsque la demande est envoyée, si des places sont disponibles, la réservation est confirmée pour l'utilisateur ; sinon, un message s'affiche indiquant que toutes les places sont complètes.

Diagramme de cas d'utilisation :



figure (3.4) : diagramme de cas d'utilisation Réservation

Diagramme de séquence

Diagramme de Séquence - Réservation de Stationnement

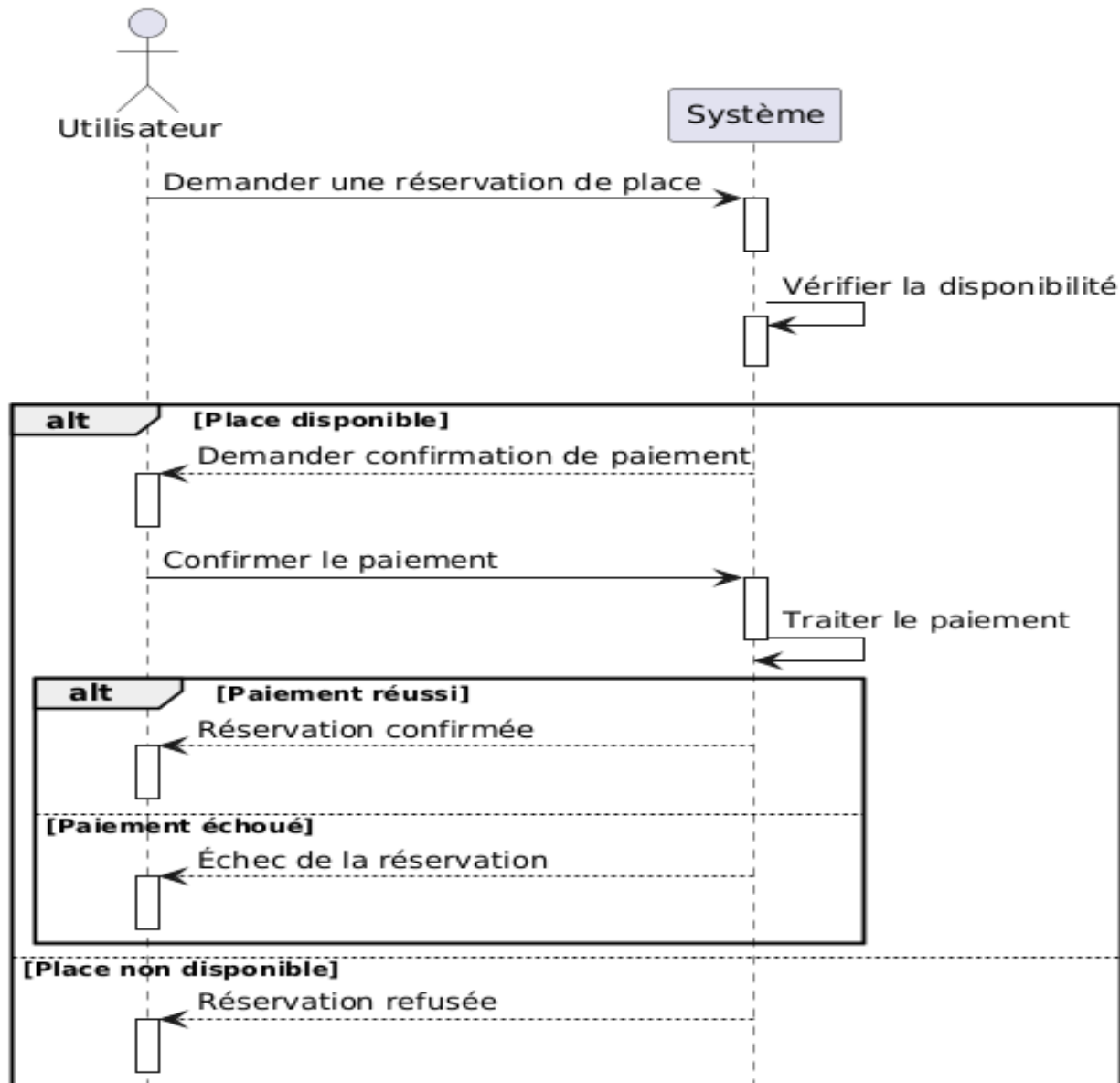


figure (3.5) : diagramme de séquence Réservation

3.3.3 Diagramme de Paiement

- L'utilisateur doit payer le montant de la réservation via l'application après la confirmation de la disponibilité et le calcul du prix.
- Lors du paiement, si la vérification du portefeuille est réussie, la réservation est confirmée et l'utilisateur reçoit une notification ; sinon, un message d'erreur s'affiche.

Diagramme de cas d'utilisation :

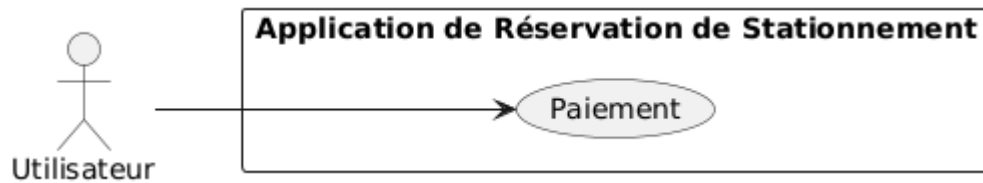


figure (3.6) : diagramme de cas d'utilisation Paiement

Diagramme de séquence :

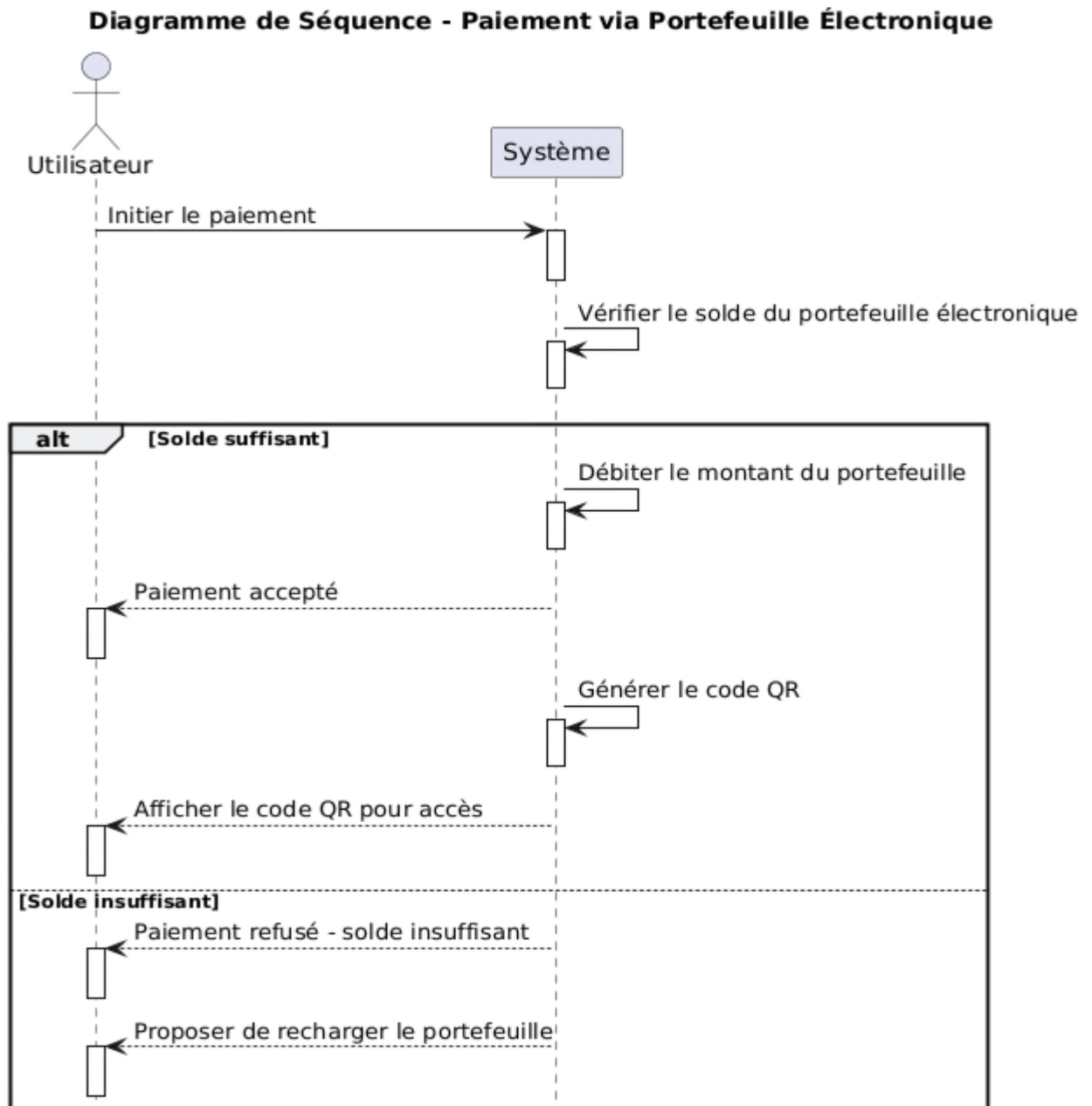


figure (3.7) : diagramme de séquence Paiement

3.3.4 Diagramme de Consultation

- L'utilisateur doit consulter les informations via l'application.
- Lorsqu'il accède à la section de consultation, si le système est disponible, les places disponibles et les tarifs sont affichés à l'utilisateur .

Diagramme de cas d'utilisation :

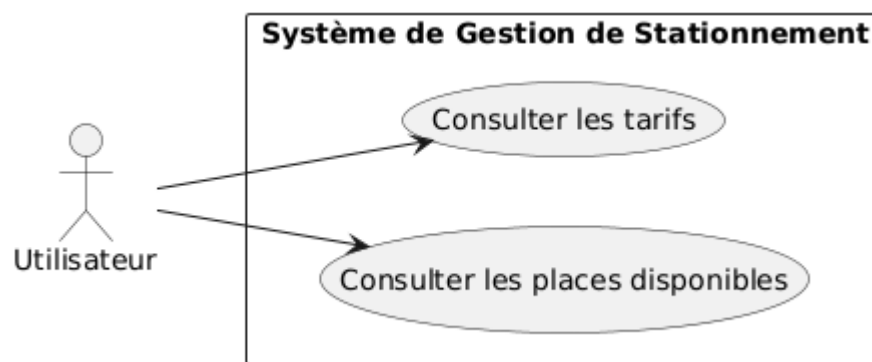


figure (3.8) : diagramme de cas d'utilisation Consultation

Diagramme de séquence :

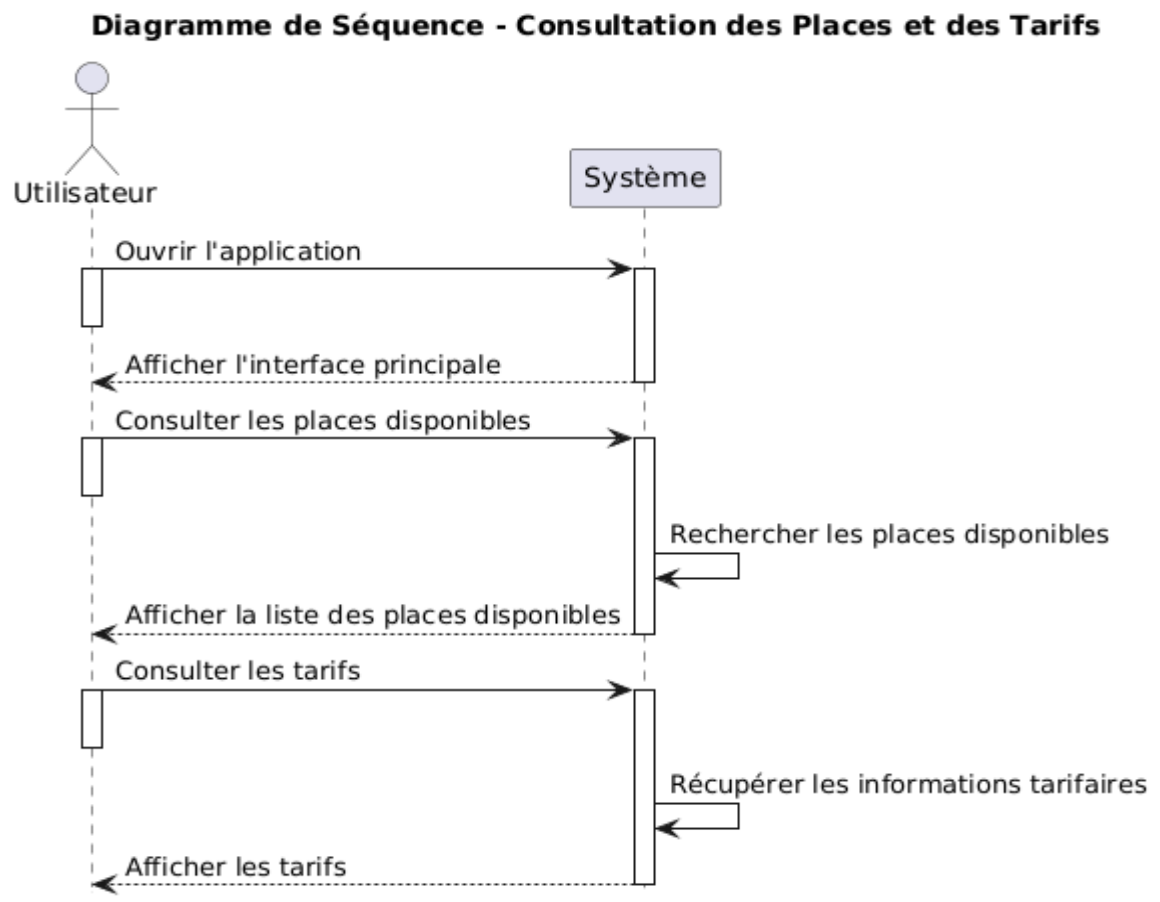


figure (3.9) : diagramme de séquence Consultation

3.3.5 Diagramme de Gestion des Parkings

- Illustre le processus d'interaction entre l'administrateur et le système pour gérer les parkings. L'administrateur accède à l'interface de gestion, le système affiche la liste des parkings, puis l'administrateur effectue des modifications et le système confirme l'opération.

Diagramme de cas d'utilisation :

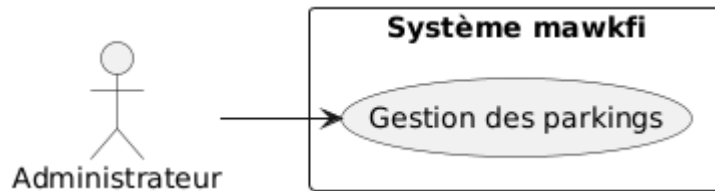


figure (3.10) : diagramme de cas d'utilisation Gestion Parking

Diagramme de séquence :

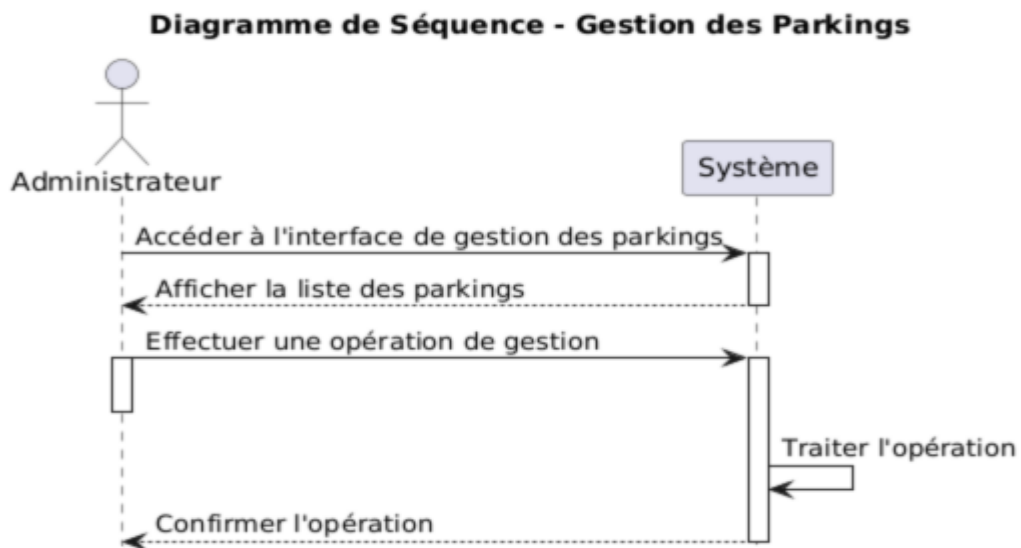


figure (3.11) : diagramme de séquence Gestion Parking

3.3.6 Diagramme de Gestion des Tarifs

- Présente comment les tarifs des parkings sont ajustés. L'administrateur consulte les tarifs actuels, effectue des modifications, le système vérifie la validité des changements, met à jour la base de données et affiche une confirmation à l'administrateur.

Diagramme de cas d'utilisation :

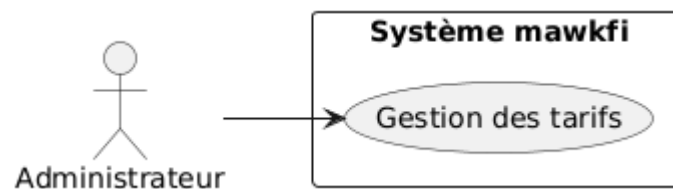


figure (3.12) : diagramme de cas d'utilisation Gestion des Tarifs

Diagramme de séquence :

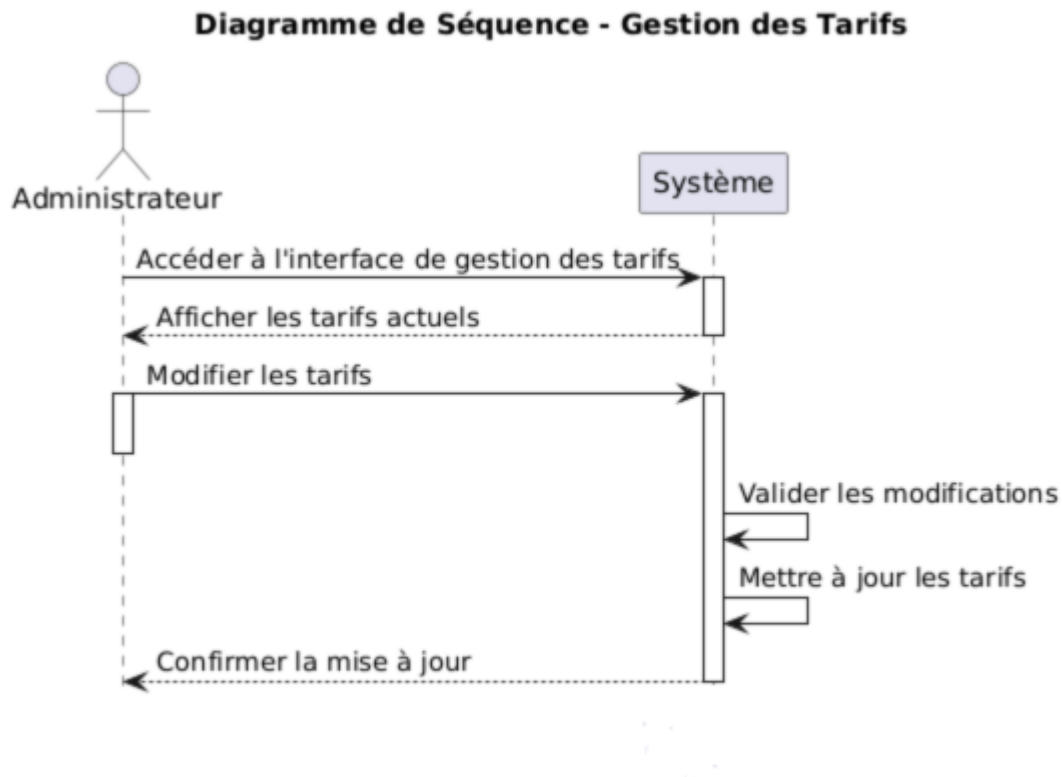


figure (3.13) : diagramme de séquence Gestion des Tarifs

3.3.7 Diagramme de Consultation des Commentaires

- Explique comment les suggestions des utilisateurs sont examinées. Le système présente une liste de commentaires, l'administrateur sélectionne un commentaire spécifique pour le consulter, puis décide comment le traiter (répondre ou archiver) et le système confirme l'action.

Diagramme de cas d'utilisation :

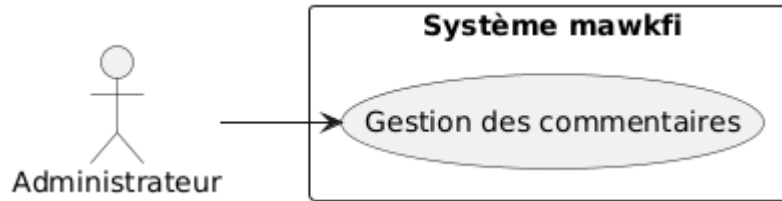


figure (3.14) : diagramme de cas d'utilisation Consultation des Commentaires

Diagramme de séquence :

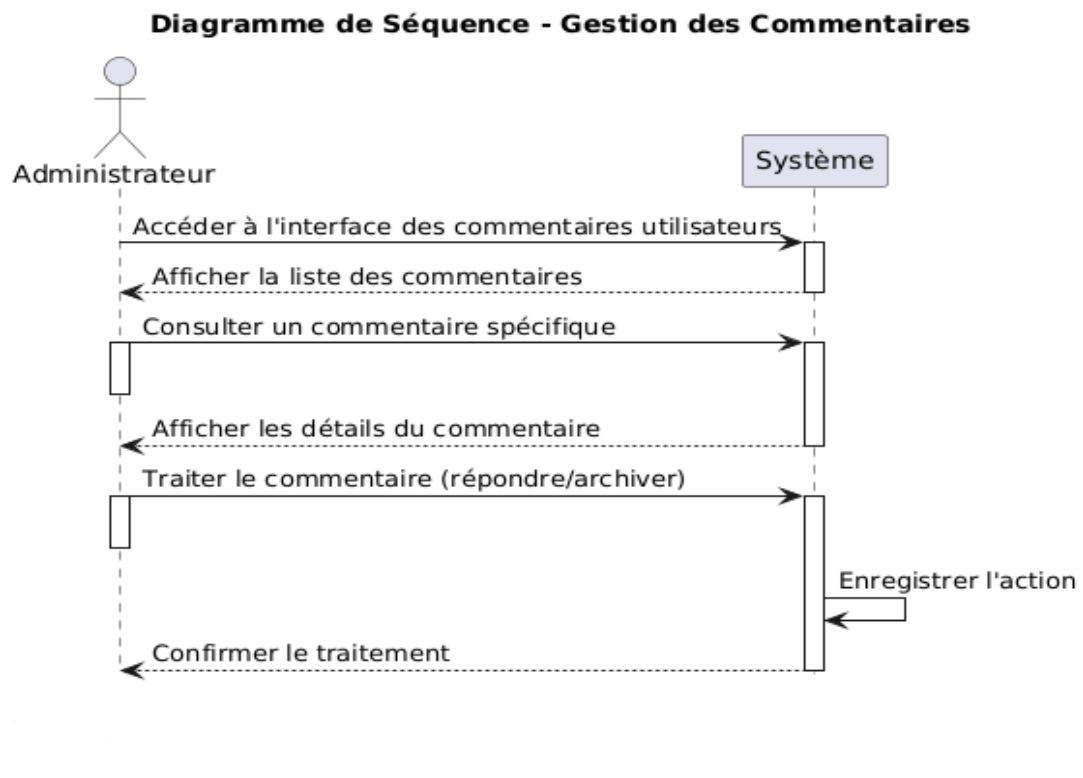


figure (3.15) : diagramme de séquence Consultation des Commentaires

3.4 Diagramme de classe

- Le diagramme de classes montre que l'utilisateur peut enregistrer ses véhicules et réserver une place dans un parking. Chaque réservation est liée à un paiement et à un ticket, et chaque parking contient plusieurs places réservables.

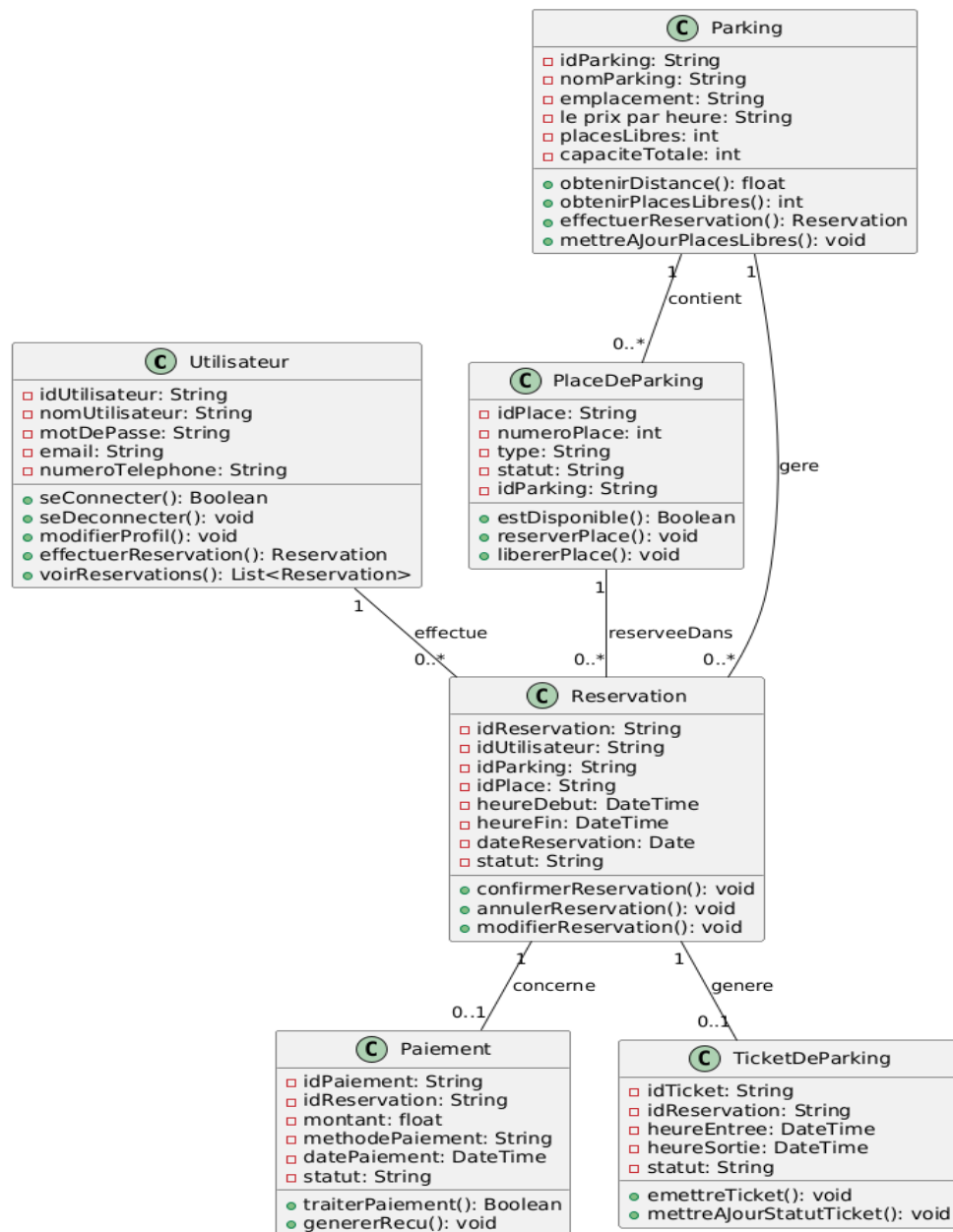


Figure (3.16) : diagramme de Classe

4. Conclusion

Au début de ce chapitre, la méthodologie 2TUP a été utilisée pour distinguer l'analyse fonctionnelle de l'analyse technique, facilitant ainsi la structuration du projet. Les diagrammes UML présentés ont permis de modéliser clairement les fonctionnalités principales de l'application "mawkfi" et les interactions entre les différents acteurs du système. Cette approche a assuré une compréhension globale et cohérente du système, posant ainsi les bases pour la phase suivante, dédiée à l'implémentation et à l'intégration de l'application dans un contexte réel de gestion de stationnement.

CHAPITRE 3

1.Introduction :

La phase d'implémentation constitue la dernière étape de notre projet et représente l'une des plus importantes, car elle concrétise l'ensemble du travail réalisé en amont. Nous commençons par une brève explication de l'ergonomie ainsi que des langages de programmation utilisés pour la création de notre application de réservation de places de parking et la mise en place de la base de données. Ensuite, nous présentons un aperçu des interfaces principales et les plus significatives de notre application.

2.Environnement de développement :

2.1Android Studio :

Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android, proposé par Google. Basé sur IntelliJ IDEA, il offre une suite complète d'outils pour concevoir, développer, tester et déployer des applications Android. Parmi ses fonctionnalités principales, on retrouve l'édition et le débogage du code, le profilage des performances, l'émulation d'appareils, la gestion de projets via Gradle, ainsi qu'une intégration facilitée avec GitHub et Google Cloud. Son interface conviviale et sa documentation riche en font un choix privilégié pour les développeurs Android de tous niveaux [8].

2.2Visual Studio Code (VS Code) :

Visual Studio Code est un éditeur de code source moderne, léger et multiplateforme, très apprécié pour sa flexibilité et la richesse de ses extensions. Grâce à des plugins spécifiques, il est possible de

développer des applications Android avec VS Code, notamment en configurant le SDK Android et les outils de ligne de commande nécessaires. VS Code propose la coloration syntaxique, l'auto-complétion intelligente (IntelliSense), le débogage, la gestion intégrée de Git, et un terminal intégré. Bien qu'il ne soit pas un IDE complet comme Android Studio, il s'avère très utile pour l'édition rapide, la gestion de projets multi-technologies, et le développement collaboratif [9].

3. Langages de programmation utilisés

Dans le développement de l'application "mawkfi", deux technologies principales ont été adoptées

3.1. React Native et le langage JavaScript :

L'interface utilisateur et la logique de l'application ont été développées avec le framework React Native, qui repose sur le langage de programmation JavaScript. JavaScript est un langage orienté objet, performant et largement utilisé, conçu pour faciliter le développement d'applications multiplateformes (Android et iOS) à partir d'une seule base de code. React Native permet de créer des interfaces interactives et fluides, avec une expérience utilisateur homogène sur toutes les plateformes [10].

3.2. Supabase pour le backend :

La plateforme Supabase a été utilisée comme backend (Backend as a Service) pour la gestion de la base de données, l'authentification, les notifications et le stockage des données. Supabase prend en charge plusieurs langages de programmation pour ses services, mais dans "mawkfi", l'intégration s'est faite directement via les bibliothèques React Native (JavaScript), sans avoir besoin d'un serveur intermédiaire. Cela permet une connexion sécurisée et efficace entre l'application et les services cloud de Supabase [11].

Résumé :

Le développement de l'application "mawkfi" s'appuie sur Visual Studio Code comme éditeur de code principal, avec recours à Android Studio pour la simulation d'appareils et la gestion de projets complexes. L'interface et la logique de l'application ont été réalisées avec React Native et JavaScript, facilitant un développement multiplateforme cohérent. La gestion des données, de l'authentification et du stockage s'effectue via Supabase, intégré directement avec React Native sans serveur intermédiaire, assurant sécurité et efficacité. Enfin, Android Studio a permis de générer un fichier APK signé, prêt pour l'installation sur les appareils Android.

4.Présentation de l'application

4.1. Interface principale

L'interface principale affiche la liste des parkings avec des informations clés comme l'emplacement, la note et le nombre de places disponibles. Elle offre aussi un accès rapide au portefeuille électronique, permettant à l'utilisateur de consulter son solde et de payer facilement, pour une expérience fluide et pratique

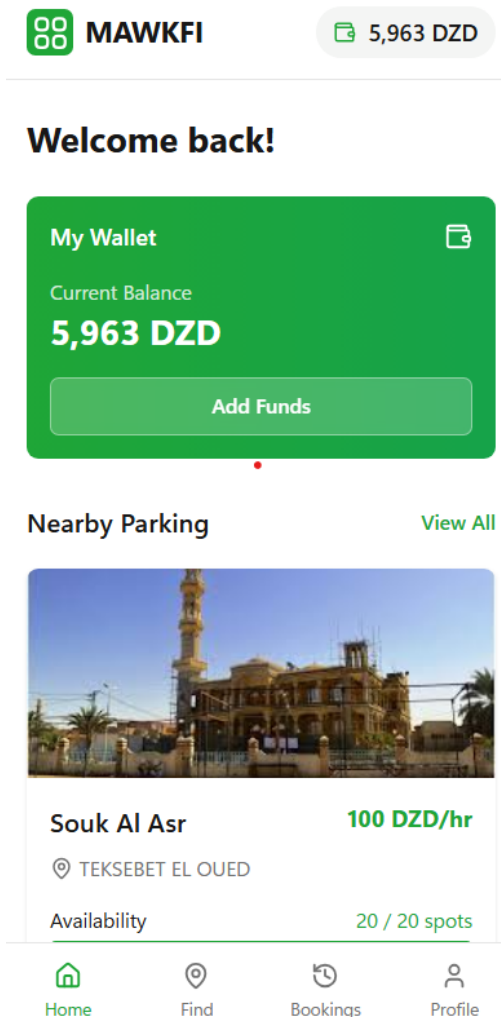


Figure (4.1) : Interface principal

4.2. Présentation des détails du parking

Cet écran présente toutes les informations détaillées sur le parking sélectionné par l'utilisateur, telles que l'adresse complète, le nombre de places disponibles, le tarif de stationnement.

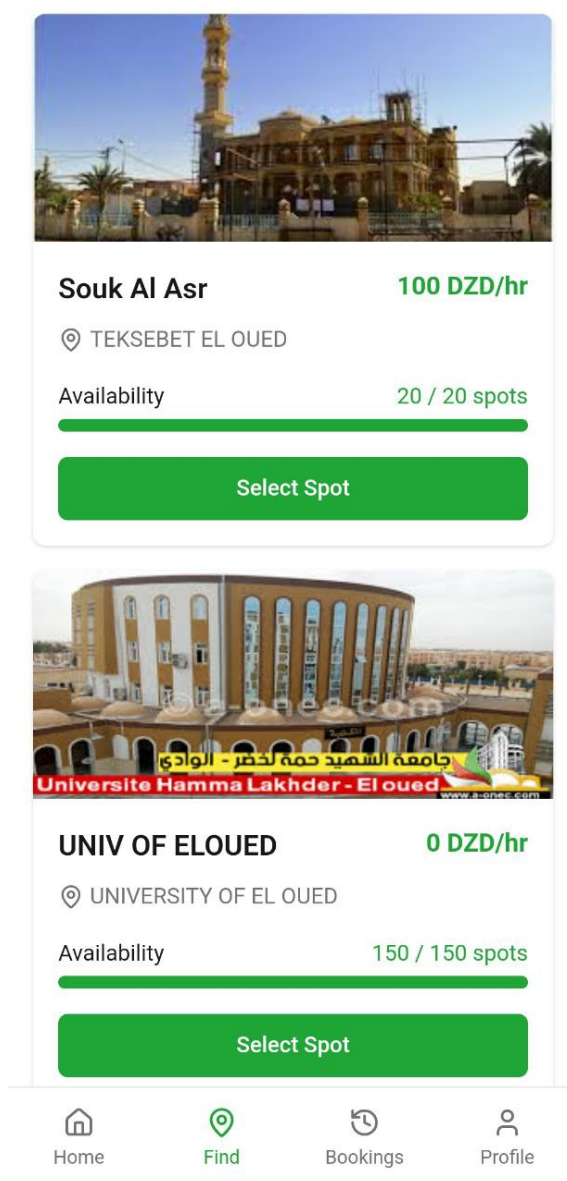


Figure (4.2) : Présentation des détails du parking

4.3. Écran de réservation

Cet écran permet à l'utilisateur de consulter et de choisir facilement une place de parking libre parmi celles disponibles.

Select a Parking Spot

Entrance

1	2	3	4
5	6	7	8
9	10		

☐ Available ☒ Selected ☐ Occupied

Select Duration

1h	2h	3h	4h
----	----	----	----

Spot	1
Duration	1 hour
Price	150 DZD

Proceed to Payment

Home Find Bookings Profile

Figure (4.3) : réservation

4.5. Écran d’affichage du paiement

affichant les détails de réservation d'une place de parking (emplacement, durée, coût total) avec l'option de paiement via le solde du portefeuille numérique .

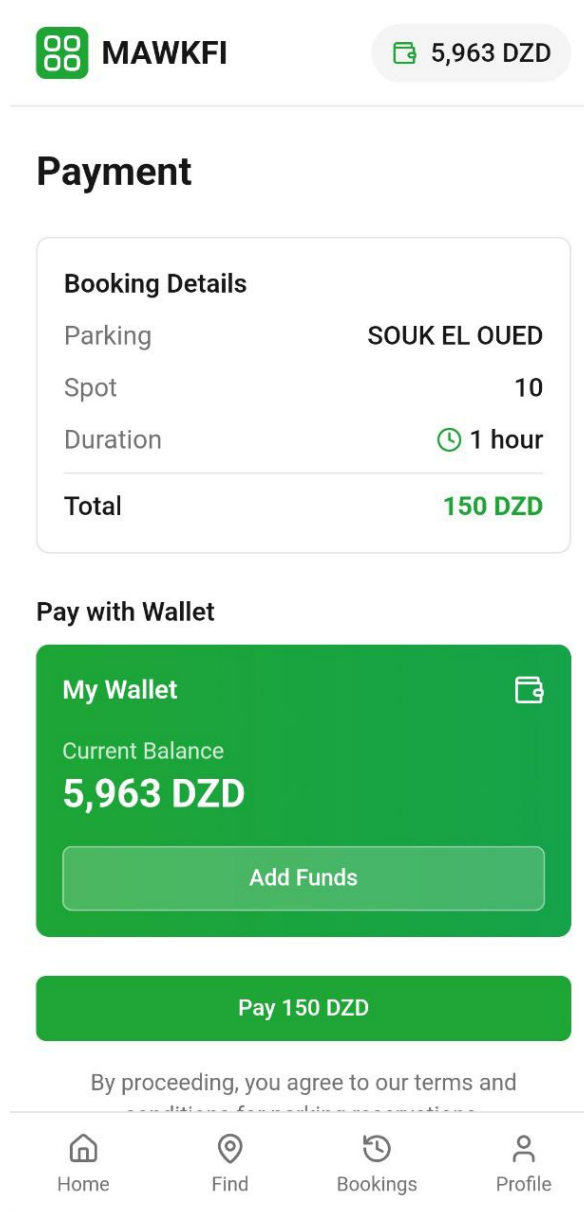


Figure (4.4) : d’affichage du paiement

4.6 Écran d'affichage du code QR

Après le paiement, un écran affiche le code QR de la réservation à utiliser à l'entrée du parking.

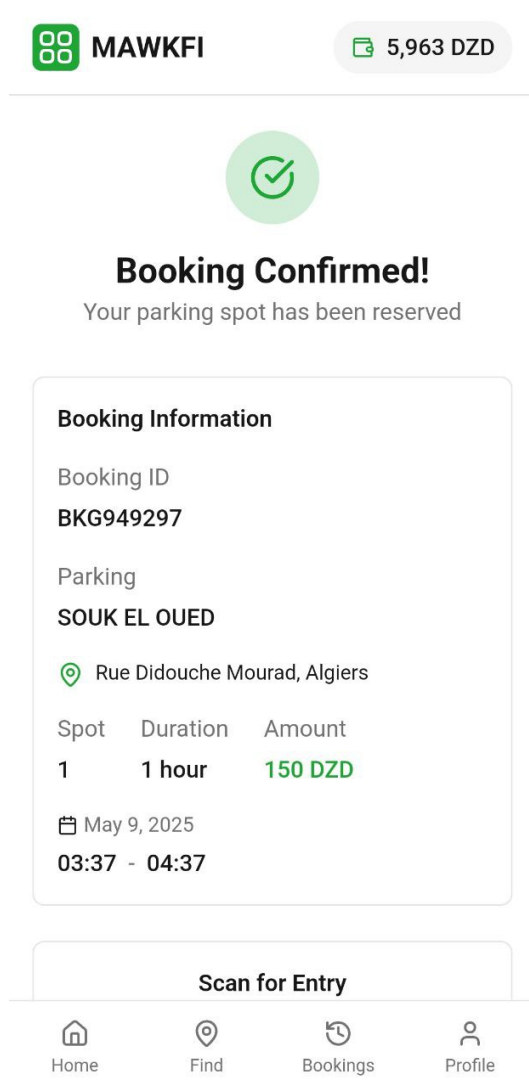


Figure (4.5) : d'affichage booking

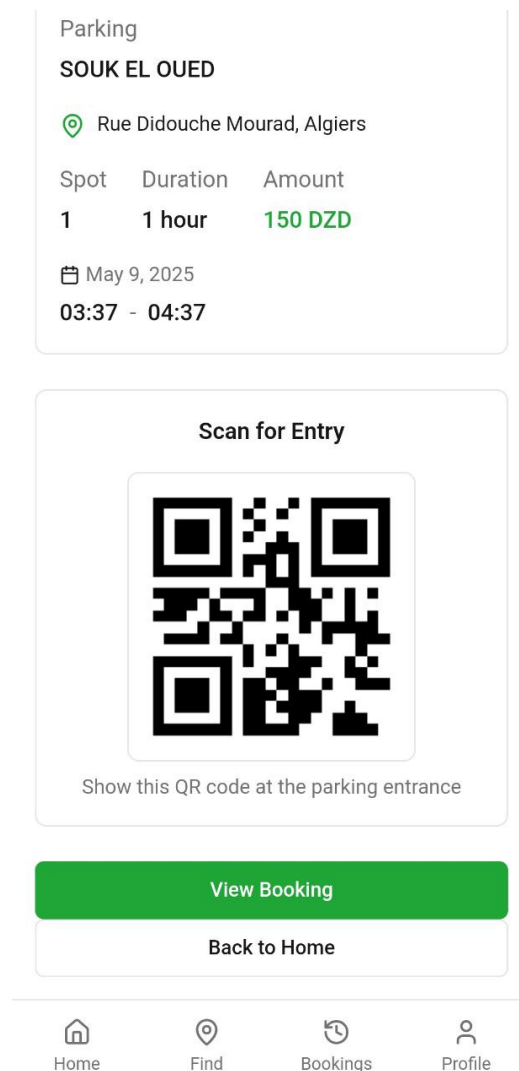


Figure (4.6) : d'affichage du code QR

4.7 Écran du portefeuille

Cet écran affiche le solde de l'utilisateur et l'historique des transactions, avec la possibilité de recharger le portefeuille pour les réservations.

The screenshot displays the MAWKFI app's wallet interface. At the top, the MAWKFI logo is on the left, and the current balance of 5,963 DZD is on the right. Below this, the word "Wallet" is centered. The main content area features three sections: a "Current Balance" section showing "5963 DZD" in large green text; an "Add Funds" section with a text input field labeled "Enter amount" and a green "Add Funds" button; and a "Withdraw Funds" section with a text input field labeled "Enter amount" and a light gray "Withdraw Funds" button. At the bottom, a navigation bar contains four icons with labels: "Home" (house icon), "Find" (location pin icon), "Bookings" (clock icon), and "Profile" (person icon).

MAWKFI 5,963 DZD

Wallet

Current Balance

5963 DZD

Add Funds

Amount (DZD)

Add Funds

Withdraw Funds

Amount (DZD)

Withdraw Funds

Home Find Bookings Profile

Figure (4.7) : d'affichage portefeuill

4.8 Interface d'administration pour l'ajout des parkings

Interface administrative permettant à l'administrateur d'ajouter de nouveaux parkings à l'application "mawkfi". Elle comprend un formulaire pour saisir les informations essentielles comme : le nom du parking, l'adresse, la capacité et le tarif, avec la possibilité d'ajouter des photos du parking.

Table Editor

schema public

+ New table

Search tables...

bookings

parking_locations

parking_slots

profiles

transactions

wallets

Filter

Sort

Insert

Auth policy

Role postgres

Realtime off

API Docs

☐	id	name	address	hourly_price	total_spo
☐	46544afd-2858-4009-9a7e-e14f05740c5e	UNIV OF ELOUED	UNIVERSITY OF EL OUED	0	150
☐	903f8346-1e1c-4a89-801f-5fbd13915f94	SABLETTE D'ALGER	SABLETTE D'ALGER	150	100
☐	9bebfd5-5303-4332-9ae6-711d3dc2ba2f	Grand Mosque Algiers	Grand Mosque Algiers	50	85
☐	b6feb384-303b-50de-b4ee-ce9e583844c	Souk Al Asr	TEKSEBET EL OUED	100	20
☐	f830632c-a16a-4303-b8fd-ceeda28e89e93	SOUK EL OUED	DAR AL WALI	150	50

Page 1 of 1100 rows5 records

RefreshDataDefinition

Figure (4.8) : Interface d'administration pour l'ajout des parkings

5. Conclusion

Ce chapitre présente l'implémentation de l'application *mawkfi*, en décrivant l'environnement de développement et les technologies utilisées, telles que React Native, JavaScript et Supabase. Nous avons mis en avant les principales interfaces de l'application, offrant une expérience utilisateur fluide. Bien que les objectifs initiaux aient été atteints, des améliorations futures sont possibles, notamment l'ajout de nouvelles fonctionnalités et l'extension de la couverture des parkings.

II. Conclusion General

Ce projet a permis de développer une application Android efficace pour la gestion intelligente des parkings, offrant aux utilisateurs une solution simple, sécurisée et moderne pour trouver, réserver et payer des places en temps réel. Cette application améliore la gestion urbaine du stationnement et ouvre la voie à des innovations futures pour une mobilité plus fluide et connectée.

Liste des références

1. <https://carrefour.vivreenville.org/> 19-3-2024 05:00PM
2. <https://fr.wikipedia.org/wiki/Parking> 21-3-2024 04:00PM
3. <https://codedelaroute.io/blog/parking-souterrain/#:~:text=Les%20parkings%20souterrains%20sont%20des,permettent%20aux%20usagers%20de%20remonter.> 22-3-2024 06:00PM
4. <https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/> 24-3-2024 12:00PM
5. <https://developer.android.com/> 25-3-2024 08:00AM
6. https://www.researchgate.net/publication/387232139_A_Y-Design_formal_modeling_framework_for_RESTful_APIs-Based_E-Learning_system_development 29-3-2024 08:00AM
7. <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/> 1-4-2024 17:00PM
8. <https://developer.android.com/studio> 3-4-2024 10:00PM
9. <https://code.visualstudio.com/docs> 3-4-2024 01:00PM
10. <https://reactnative.dev/docs/getting-started> 4-4-2024 01:00PM
11. <https://supabase.com/docs> 5-4-2025 05:00PM