مذكرة تخرج مقدمة في

جامعة محمد بوضياف – المسيلة

جامعة محمد بوضياف - المسيلة
University of Mohamed Boudiaf-Msila

كلية الرياضيات والإعلام الآلي

قسم الإعلام الآلي

من أجل الاستيفاء الجزئي لمتطلبات شهادة

ماستر في الإعلام الآلي

من طرف

طاهري أحمد الامين

زبيري محمد الصديق

عنوان المذكرة

Smart Parking Real-Time Parking Management System

تحت إشراف الأستاذ

هماك علاوة

أعضاء لجنة المناقشة

الصديق بوغرارة	جامعة محمد بوضياف- المسيلة	**رئيسا**
علاوة هماك	جامعة محمد بوضياف- المسيلة	**مشرفا**
نورالدين عمراوي	جامعة محمد بوضياف- المسيلة	**ممتحنا**

جوان، 2026

الإهداء:

"بسم الله الذي أضحك و أبكى ،بسم الله الذي أمات وأحيا ،بسم الله الذي أغنى و أقنى،بسم الله الرحمن الرحيم وصلاة وسلام على اشرف المرسلين محمد ﷺشفيعنا وقائدنا يوم القيامة.

اما بعد فإني اهدي هذا العمل المتواضع الذي بفضل الله عزوجل تم اتمامه ، الى الوالد كرمه الله في الدنيا والاخرة المختارطاهري وكذلك امي الغالية ، اطال الله في عمرهما،فهما كانا طوال حياتي سندا ورفيقا وداعما لي في كل خطوة اخطوها ،وعنوان الفضل والعطاء في حياتي.

فلا الكلمات ولا الاوصاف ولا الألفيّات تكفي لشكرهما ولوكتبت بماء الذهب،وكذلك اهدي هذا العمل الى اخوتي، بن سنوسي طاهري، محمد طاهري ،زكرياء طاهري، ،والى اختي الكبرى حفظهم الله ورعاهم بحرصه الشديد العظيم،والى جميع افراد عائلتي، وكذلك الى اصدقائي الذين عشنا معهم ذكريات جميلة في هذا المشوارالطويل،شكرا لكم جميعا حفظكم الله لي في الدنيا والاخرة" .

طاهري أحمد الأمين.

"الحمد لله حمدًا كثيرًا طيبًا مباركًا فيه، الذي بنعمته تتمّ الصالحات، وبفضله تتحقق الأمنيات، والصلاة والسلام على سيدنا محمد ﷺ خيرمن علّم وربّى، وعلى آله وصحبه أجمعين.

أما بعد،بكل مشاعرالفخروالمحبة والامتنان، وبقلبٍ يملؤه الشكرلله تعالى على ما أنعم به عليّ من توفيق وصبروقوة، أهدي هذا العمل المتواضع إلى أعظم نعمة رزقني الله بها بعد الإيمان، إلى والديَّ الكريمين.

إلى أبي العزيز، علي زبيري، الرجل الذي كان ولا يزال مثالًا للصبروالكفاح والتضحية، الذي علّمني أن النجاح لا يأتي إلا بالسعي والإرادة، وكان سندي الأول وظهري الثابت في كل مراحل حياتي. لم تكن كلماتك ودعمك مجرد تشجيع، بل كانت نورًا ير افقني في كل خطوة، فجزاك الله عني خيرالجزاء، وأدامك تاجًا فوق رأسي.

وإلى أمي الغالية، نبع الحنان والرحمة، وصاحبة القلب الطاهرالذي احتو اني بدعو اته قبل حضوره، إلى من سهرت وتعبت وضحّت لأجل راحتي ونجاحي، إلى من كانت دعواتها الصادقة تفتح لي أبواب الأمل كلما ضاقت بي الطرق. أمي العزيزة، مهما كتبت من كلمات فلن أوفيك حقك، فأنتِ سر قوتي وطمأنينتي، حفظكِ الله وأطال عمركِ وأدامكِ نعمة لا تزول.

كما أهدي هذا الإنجاز إلى إخوتي الأعزاء، وخاصة أخي زبيري حسين، الذي كان لي أخًا وسندًا ورفيقًا، أجد في دعمه وتشجيعه القوة للاستمراروتحقيق الأفضل، فله مني كل الشكروالتقديروالمحبة.

وإلى أصدقائي الأوفياء، وخاصة صديقي العزيزعبد الرحمان، شكرًا لكل لحظة دعم ومساندة، ولكل الكلمات التي رفعت من معنوياتي وجعلت هذه الرحلة أخف وأجمل.

وإلى كل من وقف إلى جانبي، وآمن بقدرتي على النجاح، أهدي هذا العمل المتواضع بكل حب ووفاء وامتنان، سائلًا الله تعالى أن يجعله ثمرة خيروبركة، وأن يوفقنا جميعًا لما يحب ويرضى".

زبيري محمد الصديق.

الشكر:

بسم الله الرحمن الرحيم الحمد لله ربّ العالمين، والصلاة والسلام على سيدنا محمد، وعلى آله وصحبه أجمعين.

نتقدّم بجزيل الشكروعظيم الامتنان إلى الأستاذ المشرف علاوة هماك، على إشر افه الدقيق وتوجيهاته القيّمة طوال فترة إعداد هذه المذكرة، وعلى ما بذله من جهود كبيرة خلال مسيرتنا الدراسية التي امتدت لعامين في هذا التخصص، وهي سنوات ستبقى راسخة في ذاكرتنا.

كما نتوجّه بخالص الشكروالتقدير إلى أعضاء لجنة المناقشة، الأستاذ بوغرارة الصديق، والأستاذ نورالدين عمرواي، على ما تفضّلوا به من ملاحظات وتوجيهات قيّمة.

ونخصّ بالشكر الأستاذ أخروف سمير، الذي كان لنا بمثابة الأب الثاني والناصح الأمين، فكان نعم الأستاذ والمربي، كما نتقدّم بالشكر إلى الأستاذ بونيف محمد، الذي لم يبخل علينا يومًا بدعمه ومساندته طوال مشوارنا الدراسي، وكذلك إلى الأستاذ قادري السعيد، الذي كان نعم السند والموجّه.

كما نتوجّه بالشكر إلى كلية الإعلام الآلي بجامعة المسيلة، على ما وفّرته من دعم علمي وإمكانات ساهمت في إنجاز هذا البحث. ولا يفوتنا أن نعبر عن بالغ امتناننا للأستاذ بوخلط محمد، الذي كان خير ناصح وداعم معنوي، وستظل توجيهاته وكلماته محفورة في أذهاننا.

وفي الختام، نتقدّم بأسمى عبارات الشكروالعرفان إلى والدينا الكريمين و أفراد أسرتنا، على دعمهم المتواصل وتشجيعهم الدائم ووقوفهم إلى جانبنا في مختلف مراحل هذا المشوار.

.والحمد لله أولًا وآخرًا

طاهري أحمد الأمين ،زبيري محمد الصديق.

اتقدم بالشكر من هذا المقام إلى جميع معلمي وأساتذتي الذين كان لهم الفضل في مسيرتي التعليمية، بدايةً من معلمي ابتدائية الشهيد أحمد زبانة بعين خرمام، ومرورًا بأساتذة متوسطة الشهيد طاهري سليمان وطاقمهما الاداري وعمالها ، إلى أساتذة ثانوية المجاهد المتوفي بن عبد الرحمان عيسى أولاد سيدي إبراهيم وطاقمهما الاداري وعمالها ، وصولًا الى اساتذة كلية الاعلام الالي بجامعة المسيلة كل بإسمه ومقامه فهم الذين أسهموا كلٌّ بطريقته في تكويني العلمي والتربوي.

وأخصّ بالذكر أستاذي الفاضل قليليز محمد، أستاذ الأدب العربي، الذي كان سندًا لي بدعمه وتشجيعه المتواصل ، وكذلك الأستاذ كسيبي علي،أستاذ الرياضيات لما قدمه من توجيهٍ ونصحٍ وعطاءٍ علمي متميز، فلهما مني بالغ الامتنان وعظيم الشكروالتقدير فقد كان لهم الأثر الكبير في غرس حب العلم والاجتهاد في نفسي.

وفي الختام، أتقدم بأسمى عبارات الشكروالعرفان إلى والدي الكريمين و أفراد أسرتي، على دعمهم المتواصل، وتشجيعهم الدائم، ووقوفهم إلى جانبي في مختلف مراحل هذا المشوار.

طاهري أحمد الأمين .

فهرس المحتويات:

الإهداء: 2

الشكر: 3

جدول الاشكال: 8

قائمة الجداول: 9

المقدمة العامة: 11

الفصل الاول: دراسة الموجود: 13

1- مقدمة: 13

2-مفهوم نظام المو اقف الذكية (SmartParking Real-Time Parking Management System): 14

2-1 التعريف الدقيق: 14

2-2 النماذج التجارية: 14

3-أطراف المنظومة (The Actors): 15

1.3- السائقون (Drivers): 15

2.3- أصحاب المو اقف / مشغلو المو اقف (Parking Owners / Operators): 15

3.3- مديرو النظام / المطورون (System Administrators / Platform Operators): 16

4.3- الجهات الحكومية والرقابية (Government Entities – B2G): 16

5.3- الأطراف الداعمة / الشركاء التقنيين (Supporting Partners): 16

4. المكونات التقنية والبنية التحتية: 16

1.4- طبقة الاستشعاروالكشف (Sensing Layer): 17

2.4- حساسات مغناطيسية (Magnetic Sensors): 17

3.4- حساسات فوق صوتية (Ultrasonic Sensors): 17

4.4- الكاميرات الذكية و ANPR (Automatic Number Plate Recognition): 17

5.4- طبقة الاتصالات والنقل (Communication Layer): 17

6.4- طبقة المعالجة والتخزين (Processing & Cloud Layer): 18
7.4 -طبقة التطبيقات والواجهات (Application Layer): 18
8.4- طبقة الأمان والخصوصية (Security Layer): 18
5-نموذج الحجز والخدمات الديناميكية: 19
1.5-آلية الحجز الذكي: 19
2.5-التسعير الديناميكي(Dynamic Pricing) : 19
3.5-الخدمات الديناميكية المضافة: 20
4.5-دور البيانات والذكاء الاصطناعي في النموذج الديناميكي: 20
6. تشخيص النقائص في السياق الحالي: 21
1.6-غياب التنسيق الرقمي والتكامل بين الأطراف: 21
2.6- الاعتماد على الأساليب التقليدية واليدوية: 21
3.6-ضعف البنية التحتية التقنية واللوجستية: 21
4.6- نقص الوعي لدى المستخدمين: 22
5.6- غياب التشريعات والسياسات الداعمة: 22
6.6 -خلاصة التشخيص: 22
7-الحلول المقترحة عبر مشروع SmartParking Real-Time Parking Management System: 22
1.7- الرؤية العامة للمشروع: 22
2.7- الحلول التقنية والتشغيلية المقترحة: 23
الخاتمة: 25
الفصل الثاني: الإطار المنهجي للدراسة وعرض نتائج الدراسة : 26
1-مقدمة: 26
2- تعريف منهج البحث و انواعه الرئيسية: 26
1.2-تعريف منهج البحث: 26
3-المنهج البحثي المعتمد ومبررات اختياره: 27

1.3-طبيعة المشكلة المطروحة:.....27
2.3- التثليث(Triangulation): 27
4- طرق البحث وإجراءات جمع البيانات: 28
1.4-الاستبيان الالكتروني(Electronic Questionnaire): 28
2.4-المقابلات(Interviews) : 29
3.4-الملاحظة الميدانية(Observational Techniques):.....29
5-عرض وتحليل النتائج الكمية والنوعية:.....30
1.5-التحليل الكمي(Quantitative Analysis) :.....30
2.5- التحليل النوعي(Qualitative Analysis): 33
6-النتائج والتوصيات(Research Findings & Recommendations) : 34
1.6- النتائج الرئيسية للدراسة:.....34
2.6-التوصيات(Recommendations) : 35
الخاتمة: 36
الفصل الثالث: متطلبات النظام وتصميمه.....37
1-المقدمة:.....37
2-منهجية إدارة المشروع:.....37
1.2- مفهوم المنهجية الرشيقة ومبادئها: 37
2.2-المبادئ الرئيسية للمنهجية الرشيقة(Key Principles of Agile Methodology):.....38
3.2-الفكرة الأساسية للمنهجية الرشيقة(Agile Methodology):.....39
4.2- مبررات اختيار المنهجية الرشيقة في هذا المشروع: 39
2-المتطلبات الوظيفية(Functional Requirements): 40
1.3-المتطلبات الوظيفية الرئيسية لمستخدم تطبيق الهاتف(Driver): 40
2.3- المتطلبات الوظيفية لأصحاب المو اقف والمسؤولين(Admin Dashboard and Manager Dashboard) : 41
3.3-المتطلبات الوظيفية العامة (للنظام ككل): 41

4-المتطلبات غير الوظيفية(Non-Functional Requirements) :..... 41
1- الأداء(Performance): 42
2- الأمان(Security): 42
3- الموثوقية(Reliability): 42
4- قابلية التوسع(Scalability): 42
5- سهولة الاستخدام(Usability): 42
6- التو افق(Compatibility) :..... 42
7- التوفر (Availability): 43
8-قابلية الصيانة(Maintainability): 43
5-تصميم قاعدة البيانات: 43
1.5- نمذجة قاعدة البيانات(Database Modeling): 43
2.5 - مجموعات قواعد البيانات(data base collections) : 44
6-نظام النمذجة System Modeling:..... 45
1.6-مخططات حالات الاستخدام (Use Case Diagrams):..... 45
2.6-مخططات التسلسلية(Sequence Diagrams):.....49
3.6-مخطط الفئات(Class diagram):.....57
الخاتمة: 60
الفصل الر ابع: التنفيذ والاختبار:..... 61
1-المقدمة:..... 61
2-بيئة التطوير و اللغات البرمجية:..... 61
2.2- جانب موقع الإداري (Web Dashboard): 61
3.2- البنية التحتية والخلفية (Backend & Infrastructure):..... 62
3-أدوات البرمجية: 63
4-بنية المشروع(System Architecture) : 64

5- مراحل التنفيذ (Step Implementation): 67

1.5-تخطيط المهام وفق المنهجية الرشيقة(Scrum/Agile Methodology):..... 67

2.5-تحديد المهام وتتبعها:..... 67

3.5- إعداد بيئة تطوير لوحة التحكم باستخدام React و(React & Vite Development Environment Setup) Vite:.... 69

4.5-إعداد بيئة تطوير تطبيق الهاتف المحمول باستخدام(Flutter Development Environment Setup) Flutter :..... 70

5.5- إدارة المشروع وتوزيع المهام(Project Management and Task Organization) :..... 71

6.5-إعداد قاعدة البيانات وتصميم الهيكلية(Database Setup and Schema Design) : 71

7.5-تطوير لوحة التحكم باستخدام(ReactJS (Web Dashboard Development :..... 72

8.5-تطوير تطبيق الهاتف المحمول باستخدام(Flutter (Flutter Mobile Application Development 74

9.5-إعداد الواجهة الخلفية وطبقة الخدمات(Backend and Service Layer) :..... 76

10.5-طبقة API المركزية (firebaseApi.js):..... 77

11.5- خدمة Firebase في تطبيق Flutter (firebase_service.dart):..... 77

12.5-آليات الأمان:..... 78

13.5-إدارة الإصدارات باستخدام(Git (Version Control with Git: 80

6-الشكل النهائي للنظام:..... 80

حدود النظام:..... 90

الخاتمة: 91

الخاتمة العامة:..... 92

المصادر: 93

جدول الاشكال:

الشكل 1.2:مخطط يوضح مبدأ التثليث(Triangulation) في البحث العلمي 28

الشكل 2.2: دائرة نسبية توضح نسب المشاركين في الاستبيان عبر الولايات 30

الشكل 3.2:دائرة نسبية توضح متوسط الوقت المستغرق للبحث عن مكان ركن شاغر في المدين 31

الشكل 4.2:دائرة نسبية توضح الاماكن التي يواجه فيها السائقون مشكلة في ايجاد مكان للركن 32

الشكل 5.2:دائرة نسبية توضح أبرز المشاكل التي يواجهها السائقون عند البحث عن مكان ركن 33
الشكل 1.3:صورة توضح دورة حياة التطبيق في المنهجية الرشيقة 39
الشكل 2.3:نموذج قاعدة البيانات 44
الشكل 3.3:مخطط حالات الاستخدام للمسؤول 47
الشكل 4.3:مخطط حالات الاستخدام للمدير 48
الشكل 5.3:مخطط حالات الاستخدام للسائق 49
الشكل 6.3:مخطط تسلسلي لعملية تسجيل الدخول. 51
الشكل 7.3:المخطط التسلسلي لعملية تسجيل الدخول للتطبيق بالنسبة للسائق. 52
الشكل 8.3:المخطط التسلسلي لعملية تسجيل حساب في تطبيق بالنسبة للسائقين. 53
الشكل 9.3:المخطط التسلسلي لعملية البحث عن اقرب موقف بإستخدام خوارزمية Dijkstra 54
الشكل 10.3:المخطط التسلسلي لعملية الحجز بالتطبيق 55
الشكل 11.3:المخطط التسلسلي لعملية المعالجة الخلفية. 56
الشكل 12.3:المخطط التسلسلي لحياة التطبيق. 57
الشكل 13.3:المخطط الفئات لنظام 59
الشكل 1.4:مخطط يوضح كيف يتم تطبيق Role-Sliced Dashboard Architecture في صفحة الويب (Web Dashboard)..... 65
الشكل 2.4:مخطط يوضح كيف يتم تطبيق Layered Architecture في تطبيق الهاتف المحمول (Flutter). 66
الشكل 3.4:استخدام جدول EXCELلتسجيل سيرورة العمل ومدى الانجاز في كل جزء 69
الشكل 4.4:لوحة تسجيل الدخول للمسؤول او المدير 81
الشكل 5.4:صفحة الاحصائيات للمسؤول 82
الشكل 6.4:صفحة الخريطة والمو اقف للمسؤول 83
الشكل 7.4:صفحة ادارة المو اقف للمسؤول 84
الشكل 8.4:صفحة الاحصائيات للمدير 85
الشكل 9.4:صفحة المو اقف للمدير. 86
الشكل 10.4:صفحة تسجيل الدخول للسائق. 87
الشكل 11.4:صفحة المو اقف للمدير. 87
الشكل 12.4:صفحة التوجيه للموقف. 88

قائمة الجداول:

جدول 1.3:انواع المجموعات في قاعدة البيانات 45

جدول 1.4:جدول يوضح مواصفات اجهزة التطوير.....64
جدول 2.4:انواع المجموعات في قاعدة البيانات.....72

المقدمة العامة:

تشهد المدن الجزائرية في السنوات الأخيرة نموًا سريعًا ومستمرًا في عدد المركبات، مما أدى إلى تفاقم أزمة المواقف بشكل كبير ،بحيث انه يعاني السائقون يوميًا من صعوبة بالغة في إيجاد مكان مناسب لركن سياراتهم، خاصة في المناطق التجارية المزدحمة ووسط المدن، حيث يضطرون إلى قضاء وقت طويل في الدوران والتجول بحثا عن موقف شاغر. هذا الواقع يؤدي إلى إهدار كبير للوقت والوقود، وزيادة الازدحام المروري، وارتفاع مستويات التلوث البيئي، بالإضافة إلى ارتفاع التوتر والضغط النفسي اليومي الذي يعيشه السائقون.

يهدف هذا المشروع إلى معالجة التحديات والقيود التي يواجهها السائقون وأصحاب المواقف في السياق الجزائري، وتقديم حلول عملية لتحسين تجربة إدارة المواقف. بالإضافة إلى ذلك، يستعرض البحث سبل تحسين الأنظمة الحالية، وأفضل الممارسات في تصميم وتنفيذ منصات سهلة الاستخدام، وآليات التوافر في الوقت الفعلي، وأنظمة الحجز والدفع الآمنة، وعوامل تعزيز رضا المستخدمين وولائهم.

تتمثل أسئلة البحث التي يتناولها هذا العمل فيما يلي:

- ما هي أبرز المشكلات التي يواجهها السائقون وأصحاب المواقف في الجزائر، وكيف يمكن حل هذه المشكلات؟
- كيف يمكن تحسين أنظمة إدارة المواقف الحالية لتلبية احتياجات السائقين والمشغلين بشكل أفضل؟
- ما هي أفضل الممارسات في تصميم وتنفيذ منصة ذكية سهلة الاستخدام وفعالة لإدارة المواقف في الوقت الفعلي؟
- كيف يمكن لمنصات المواقف الذكية ضمان معلومات توافر دقيقة ونظام حجز موثوق؟
- ما هي أنجح الاستراتيجيات لإدارة الحجوزات والمدفوعات بين السائقين وأصحاب المواقف؟

يقدم الفصل الأول من هذه المذكرة دراسة شاملة لواقع تسيير المواقف في الجزائر، ويعرف نظام المواقف الذكية تعريفًا دقيقًا، ويبرز أهميته في منظومة المدن الذكية. كما يتناول النماذج التجارية المختلفة (B2C, B2B, B2G, C2C/C2B2C)، وأطراف المنظومة، والمكونات التقنية والبنية التحتية، مع تشخيص النقائص في السياق الجزائري (غياب التنسيق الرقمي، الاعتماد على الأساليب التقليدية، ضعف البنية التحتية، نقص الوعي، وغياب التشريعات الداعمة)، ويقترح الحلول من خلال مشروع SmartParking.

أما الفصل الثاني فيحدد الإطار المنهجي للدراسة، معتمدًا على المنهج المختلط (Mixed Approach) القائم على التثليث (Triangulation). وقد تم استخدام أدوات متعددة لجمع البيانات شملت الاستبيان الإلكتروني، والمقابلات، والملاحظة الميدانية. وتم تحليل البيانات كميًا ونوعيًا لاستخلاص نتائج حول المشكلات اليومية للسائقين، ومناطق الضغط، ومستوى الرضا، بالإضافة إلى التوصيات العملية.

بالاعتماد على نتائج الدراسة الميدانية، يركز الفصل الثالث على متطلبات النظام وتصميمه، من خلال تحديد المتطلبات الوظيفية وغير الوظيفية، وتصميم قاعدة البيانات باستخدام Firebase Firestore، ونمذجة النظام عبر مخططات UML (مخططات حالات الاستخدام، والمخططات التسلسلية للعمليات الرئيسية، ومخطط الفئات).

يناقش الفصل الرابع مرحلة التنفيذ والاختبار، حيث يوضح المنهجية الرشيقة (Agile - Scrum) المعتمدة، والتقنيات المستخدمة لتطبيق الهاتف(Dart ،Flutter) للوحة التحكم(ReactJS ، Vite، JavaScript، HTML5، CSS3)، وا لبنية الخلفية(Firebase: Firestore ، Authentication، Cloud Functions،Cloud Messaging)،والبنى المعمارية ، ومراحل التطوير، وآليات الأمان، والشكل النهائي للمنصة،ويبرز حدود النظام الحالي ويناقش الطموحات المستقبلية لهذا المشروع.

بشكل عام، تقدم هذه المذكرة استكشافًا شاملاً لأزمة المواقف في الجزائر والحلول التقنية المقترحة. ومن خلال الإجابة على أسئلة البحث عبر التحليل الميداني وتصميم النظام وتنفيذه، تهدف هذه الدراسة إلى المساهمة في تطوير منظومة مواقف ذكية متكاملة، وتمهيد الطريق لمبادرات مستقبلية في مجال المدن الذكية بالجزائر.

الفصل الاول: دراسة الموجود:

1- مقدمة:

نظام المواقف الذكية (SmartParking Real-Time Parking Management System) هو نظام يعتمد على استخدام تكنولوجيا المعلومات والاتصال من أجل إدارة وتنظيم مواقف السيارات بشكل آلي وفعال. يهدف هذا النظام إلى تمكين السائقين من معرفة أماكن الركن الشاغرة في الوقت الفعلي، وتسهيل الوصول إليها، مع تقليل الوقت المهدور في البحث العشوائي عن موقف، والحد من الازدحام المروري داخل المدن.

ويتحقق ذلك من خلال الاعتماد على تقنيات مثل إنترنت الأشياء (IoT)، الحساسات أو الكاميرات، أنظمة تحديد المواقع (GPS)، والواجهات الرقمية التي تربط بين المستخدمين ومزودي الخدمة.

يهدف هذا الفصل إلى تشخيص واقع تسيير المواقف في الجزائر حاليا، تحديد الثغرات الرئيسية التي تعيق تحولها إلى مواقف ذكية، واستعراض المكونات التقنية والتشغيلية اللازمة لتجاوز الأساليب التقليدية المعتمدة اليوم. وسيتم ذلك من خلال التركيز على نموذج عمل يعتمد على الذكاء الاصطناعي، معالجة البيانات في الوقت الفعلي، والحوكمة الرقمية.

وسنتناول في هذا الفصل النقاط التالية:

1. مفهوم نظام المواقف الذكية (SmartParking Real-Time Parking Management System)
2. أطراف المنظومة (The Actors).
3. المكونات التقنية والبنية التحتية.
4. نموذج الحجز والخدمات الديناميكية.
5. تشخيص النقائص في السياق الحالي.
6. الحلول المقترحة عبر مشروع SmartParking Real-Time Parking Management System

2-مفهوم نظام المو اقف الذكية (SmartParking Real-Time Parking Management System):

نظام المواقف الذكية (SmartParking Real-Time Parking Management System) هو نظام متكامل يعتمد على التكنولوجيا لتحسين إدارة مواقف السيارات في المناطق العامة والخاصة، من خلال توفير معلومات فورية عن توافر الأماكن، تسهيل الحجز والدفع، وتقليل وقت البحث العشوائي عن موقف شاغر**[11,10]**. يُعرف هذا النظام كحل ذكي يجمع بين إنترنت الأشياء (IoT)، الحساسات أو الكاميرات، والواجهات الرقمية لتحويل عملية الركن من عملية تقليدية فوضوية إلى عملية منظمة وفعالة**[22,13]**.

2-1 التعريف الدقيق:

نظام المواقف الذكية هو منصة رقمية ثنائية الواجهة تتكون من:

- تطبيق موبايل موجه للمستخدمين النهائيين (السائقين).
- موقع ويب إداري موجه للمسؤولين (أصحاب المواقف، المشغلين، الجهات الإدارية).

يهدف النظام إلى رقمنة العملية الكاملة لركن المركبات، بدءاً من كشف التوافر، مروراً بالحجز والتوجيه، وانتهاءً بالدفع والإدارة**[23,22]**، مع تحقيق الأهداف التالية:

- تقليل الازدحام الناتج عن البحث العشوائي عن مكان ركن.
- تحسين استغلال المساحات المتاحة (حيث يبقى جزء كبير من المواقف فارغاً في أوقات معينة).
- دعم التنقل الحضري المستدام من خلال تقليل استهلاك الوقود والوقت المهدور.

لتحقيق هذه الأهداف، يعتمد النظام على نماذج تجارية متعددة (**B2C, B2B, B2G, C2C / C2B2C**) تسمح بتغطية مختلف الأطراف المعنية (السائقون، أصحاب المواقف، الجهات الإدارية)، وتتنوع هذه النماذج حسب طبيعة الاستخدام والطرف المستهدف**[5,4,3]**.

2-2 النماذج التجارية:

1.2.2 -B2C (Business-to-Consumer):

المنصة تقدم الخدمة مباشرة للمستخدم (السائق العادي) مثل حجز موقف بدفع إلكتروني عبر التطبيق، عرض التوافر في الوقت الفعلي، توجيه GPS مباشر إلى المكان**[5,3]**.

2.2.2 B2B (Business-to-Business):

المنصة تبيع الحلول أو الخدمات لشركات أو مؤسسات (أصحاب مواقف، مراكز تجارية، مستشفيات، فنادق)**[4,3]**.

B2G (Business-to-Government) -3.2.2:

المنصة تتعاون مع الجهات الحكومية (بلديات، وزارة النقل، ولايات) لإدارة المواقف العمومية والتخطيط الحضري**[4,3]**.

C2B2C (Consumer-to-Consumer / Consumer-to-Business-to Consumer) أو C2C -4.2.2:

- C2C: سائق يؤجر موقفه الخاص (منزلي أو مكان عمل) مباشرة لسائق آخر عبر المنصة.

- C2B2C: المالك (consumer) يقدم موقفه للمنصة (business)، والمنصة تبيعه لسائق آخر (consumer).

هذا النموذج يسمح باستغلال المواقف غير المستخدمة في الأحياء السكنية، وهو أقل انتشاراً حالياً في الجزائر لكنه يُعتبر فرصة مستقبلية كبيرة لزيادة العرض دون بناء مواقف جديدة**[9,6]**.

بهذا الشكل، يصبح نظام المواقف الذكية ليس مجرد تطبيق للحجز، بل نظام بيئي اقتصادي واجتماعي يربط بين جميع الأطراف لتحقيق تنقل حضري أكثر كفاءة واستدامة**[7,6]**.

3-أطراف المنظومة (The Actors):

في نظام المواقف الذكية (**Smart Parking**)، يتكون النظام البيئي من مجموعة من الأطراف الرئيسية التي تتفاعل مع بعضها البعض لضمان نجاح العملية ككل. كل طرف له دور محدد، ويتم الربط بينهم عبر المنصة الرقمية (تطبيق الموبايل للمستخدمين و موقع الويب للمسؤولين)**[23,22]**. ويمكن تصنيف الأطراف الرئيسية كالتالي:

1.3- السائقون (Drivers):

- هم الطرف الأساسي الذي يستفيد مباشرة من النظام.

- دورهم: البحث عن مواقف شاغرة، حجز المكان مسبقاً، الدفع الإلكتروني، الوصول إلى الموقع المحجوز، وتقييم الخدمة.

- واجهتهم: تطبيق الموبايل الذي يقدم عرض التوافر في الوقت الفعلي، خرائط التوجيه، الإشعارات، وسجل الحجوزات.

- أهميتهم: يمثلون مصدر الطلب ويولدون الإيرادات من خلال رسوم الحجز أو الدفع**[22,23]**.

2.3- أصحاب المو اقف / مشغلو المو اقف (Parking Owners / Operators):

- يشملون: أصحاب المواقف الخاصة (مراكز تجارية، فنادق، مستشفيات، مكاتب، أحياء سكنية)، أو الشركات الخاصة التي تدير مواقف.

- دورهم: توفير المساحات المتاحة، تحديد الأسعار (ثابتة أو ديناميكية حسب الوقت/الطلب)، مراقبة الاستخدام، وإدارة الوصول (بوابات، ANPR إن وجد)**[22,10]**.

- واجهتهم: موقع الويب الإداري (**Dashboard**) الذي يتيح تسجيل المناطق، عرض التوافر الحالي، إدارة الحجوزات، عرض التقارير (إيرادات، نسبة الاستخدام، ساعات الذروة)، وإصدار فواتير أو استلام الدفعات.

- أهميتهم: يمثلون مصدر العرض (supply) ويحققون دخلاً إضافياً من خلال استغلال المساحات غير المستخدمة.

3.3- مديرو النظام / المطورون (System Administrators / Platform Operators):

- هم فريق المشروع أو الشركة المسؤولة عن تشغيل المنصة.

- دورهم: إدارة المنصة ككل، مراقبة الأداء، حل المشاكل التقنية، إضافة مواقف جديدة، تحديث الخوارزميات، ضمان الأمان والخصوصية، وتحليل البيانات لتحسين النظام**[35,34]**.

- واجهتهم: لوحة تحكم متقدمة داخل موقع الويب (Admin Panel) تتيح الوصول الكامل إلى البيانات، إدارة المستخدمين (سائقين وأصحاب مواقف)، ومراقبة النشاط في الوقت الفعلي.

4.3- الجهات الحكومية والرقابية (Government Entities – B2G):

- تشمل: البلديات، وزارة النقل، الولايات، الشرطة المرورية، أو هيئات التنظيم الحضري.

- دورهم: وضع السياسات والتعرفة في المواقف العمومية، مراقبة المخالفات، استخدام البيانات للتخطيط المروري والحضري (مثل تحديد مناطق الذروة أو توسيع الشبكة)**[4,3]**.

- واجهتهم: تقارير مخصصة أو لوحات إحصائية (Dashboard) يمكن مشاركتها معهم، أو واجهة API للتكامل مع أنظمتهم الحكومية.

- أهميتهم: يضمنون الشرعية والتوسع على نطاق واسع، ويمكن أن يصبحوا شركاء في تمويل أو دعم النظام.

5.3- الأطراف الداعمة / الشركاء التقنيين (Supporting Partners):

- تشمل: شركات الدفع الإلكتروني، مزودي خرائط (Google Maps، OpenStreetMap)، شركات IoT (حساسات، كاميرات)، أو شركات اتصالات (للإنترنت والإشعارات).

- دورهم: توفير الخدمات التقنية الداعمة (دفع آمن، خرائط دقيقة، نقل بيانات)**[19,17,13].**

- أهميتهم: يضمنون استمرارية النظام وتحسين تجربة المستخدم.

4. المكونات التقنية والبنية التحتية:

يستند نظام المواقف الذكية إلى بنية تحتية تقنية متكاملة تضمن جمع البيانات بدقة، نقلها بسرعة، معالجتها في الوقت الفعلي، وتقديمها للمستخدمين والمسؤولين عبر الواجهات المزدوجة (تطبيق الموبايل وموقع الويب الإداري)**[22,17].** تُقسم هذه البنية عادةً إلى طبقات رئيسية،

مع التركيز في مشاريع التخرج مثل **SmartParking** على حلول اقتصادية ومناسبة للسياق الجزائري (تكلفة منخفضة نسبياً، سهولة التركيب، واعتماد جزئي على البنية الموجودة).

1.4- طبقة الاستشعار والكشف (Sensing Layer):

هذه الطبقة مسؤولة عن جمع البيانات الأولية حول حالة المواقف (شاغر/مشغول). تشمل التقنيات الأساسية المستخدمة:

- الحساسات (Sensors):

2.4- حساسات مغناطيسية (Magnetic Sensors):

تكتشف وجود السيارة عن طريق تغير المجال المغناطيسي.

مزايا: تكلفة منخفضة، استهلاك طاقة قليل، مناسبة للمواقف المغلقة أو شبه المغلقة.

عيوب: لا تعطي معلومات عن نوع السيارة أو لوحتها**[16,15,14].**

3.4- حساسات فوق صوتية (Ultrasonic Sensors):

تقيس المسافة باستخدام موجات صوتية.

مزايا: رخيصة، سهلة التركيب.

عيوب: حساسة للعوامل الجوية (مطر، غبار) وغير دقيقة في المساحات المفتوحة الكبيرة**[15,14].**

4.4- الكاميرات الذكية و ANPR (Automatic Number Plate Recognition):

- كاميرات مراقبة مدعومة بالذكاء الاصطناعي للكشف عن السيارات وتحديد لوحاتها.

مزايا: دقة عالية، لا تحتاج إلى تركيب في كل مكان (كاميرا واحدة تغطي عدة مواقف)، تسمح بتحديد هوية السيارة (مفيد للحجز والدفع التلقائي).

عيوب: تكلفة أعلى، تحتاج إضاءة جيدة، مشاكل في الخصوصية إذا لم تُدار بشكل صحيح**[15,13].**

- في السياق الجزائري: غالباً ما تكون ANPR الخيار الأكثر شيوعاً في المشاريع الناجحة.

5.4- طبقة الاتصالات والنقل (Communication Layer):

تنقل البيانات من الحساسات/الكاميرات إلى الخادم الرئيسي. التقنيات الشائعة:

- شبكات منخفضة الطاقة: **LoRaWAN** (مفضلة في المدن الكبيرة لتغطية واسعة واستهلاك طاقة منخفض)**[19,17].**

- Wi-Fi أو 4G/5G: للمواقف ذات التغطية العالية أو الداخلية.

- البروتوكولات: **MQTT** أو **HTTP/HTTPS** لنقل البيانات، مع Kafka في الحالات المتقدمة لمعالجة التدفق اللحظي (real-time streaming) عند وجود آلاف المواقف**[19,18]**.

6.4- طبقة المعالجة والتخزين (Processing & Cloud Layer):

- الخوادم السحابية **Google Cloud :(Cloud Backend)**، **AWS**، أو خوادم محلية (VPS في الجزائر).

- قواعد البيانات: **PostgreSQL** أو **Firebase**أو **MongoDB** لتخزين بيانات المواقف، الحجوزات، المستخدمين**[22,17]**.

- معالجة البيانات في الوقت الفعلي:

- أدوات مثل **Apache Kafka** أو **RabbitMQ** لتدفق البيانات اللحظي.

- خوارزميات تحسين (**Optimization Algorithms**) لتحديد أقرب موقف شاغر أو أفضل مسار (مثل **Dijkstra** أو**A*** او **Hybrid A*)[22,21]**.

7.4 -طبقة التطبيقات والواجهات (Application Layer):

- تطبيق الموبايل (للسائقين)

وظائف: عرض الخريطة، الحجز، الدفع، الإشعارات، التقييم**[23,22]**.

- موقع الويب الإداري .

وظائف: لوحة تحكم (**Dashboard**)، إدارة المواقف، تقارير الإيرادات، إدارة المستخدمين، تحليلات**[22]**.

8.4- طبقة الأمان والخصوصية (Security Layer):

- تشفير البيانات **(HTTPS، TLS)**.

- مصادقة المستخدمين **(JWT، OAuth)**.

- حماية خصوصية لوحات السيارات (تخزين مؤقت، عدم تخزين صور دائمة إلا بموافقة).

- في السياق الجزائري: الالتزام بقانون حماية البيانات الشخصية (القانون 18-07)**[25,24]**.

5-نموذج الحجزوالخدمات الديناميكية:

يمثل نموذج الحجز والخدمات الديناميكية أحد الركائز الأساسية في أنظمة المواقف الذكية، حيث ينتقل تسيير عملية الركن من منطق الاستخدام العشوائي والثابت إلى منطق ذكي يعتمد على البيانات في الوقت الفعلي**[35,34]**، التنبؤ بالطلب، والتفاعل المستمر مع المستخدمين. ويهدف هذا النموذج إلى تحسين تجربة السائق من جهة، وتعظيم كفاءة استغلال المواقف والإيرادات من جهة أخرى.

يعتمد هذا النموذج على التكامل بين تطبيق الموبايل، الخوادم الخلفية، وخوارزميات المعالجة، بما يسمح بإدارة مرنة وديناميكية لعمليات الحجز، التسعير، والخدمات المضافة.

1.5-آلية الحجز الذكي:

يقوم نظام الحجز الذكي على تمكين السائق من حجز موقف سيارة مسبقًا أو في الوقت الفعلي عبر التطبيق**[32]**، وذلك بناءً على مجموعة من المعايير، من بينها:

- الموقع الجغرافي.
- وقت الوصول المتوقع.
- مدة الركن.
- نوع الموقف (عمومي أو خاص).

بعد اختيار الموقف المناسب، يقوم النظام بحجز المكان تلقائيًا لفترة زمنية محددة، مع تحديث حالة التوافر في قاعدة البيانات بشكل فوري**[22]**، ما يمنع الازدواجية أو التضارب في الحجوزات. كما يتم توجيه السائق إلى الموقع المحجوز عبر نظام تحديد المواقع(GPS) ، مع إمكانية إلغاء أو تعديل الحجز وفق شروط محددة**[23].**

2.5-التسعير الديناميكي(Dynamic Pricing) :

يعتمد التسعير الديناميكي على تعديل سعر الركن تلقائيًا وفق مجموعة من العوامل المتغيرة**[29,28,27,26]**، مثل:

- مستوى الطلب في فترات الذروة.
- مدة الركن.
- الموقع (مركز المدينة، مناطق تجارية، أحياء سكنية).
- نوع اليوم (عادي، عطلة، مناسبات خاصة).

يسمح هذا الأسلوب بتحقيق توازن بين العرض والطلب، تشجيع الاستخدام في الفترات الأقل ازدحامًا**[33,32,30]**، وتحسين العائد المالي لأصحاب المواقف. كما يساهم في توجيه السائقين نحو مواقف بديلة أقل ازدحامًا، مما يخفف الضغط المروري داخل المدن.

3.5-الخدمات الديناميكية المضافة:

لا يقتصر نموذج SmartParking Real-Time Parking Management System على الحجز فقط، بل يشمل مجموعة من الخدمات الديناميكية التي تتكيف مع سلوك المستخدم واحتياجاته، من بينها:

- الإشعارات الفورية حول انتهاء مدة الركن أو توفر مواقف قريبة،
- تمديد مدة الركن تلقائيا عند الحاجة.
- اقتراح مواقف بديلة في حال الامتلاء.
- سجل الحجوزات والدفع الإلكتروني.
- أنظمة التقييم والتغذية الراجعة (Feedback).

تساهم هذه الخدمات في تحسين تجربة المستخدم، رفع مستوى الثقة في النظام**[35,22]**، وزيادة معدلات الاستخدام المتكرر.

4.5-دورالبيانات والذكاء الاصطناعي في النموذج الديناميكي:

يعتمد نموذج الحجز والخدمات الديناميكية على جمع وتحليل كميات كبيرة من البيانات، مثل أنماط الطلب، أوقات الذروة، وسلوك المستخدمين. ويتم توظيف خوارزميات الذكاء الاصطناعي والتعلم الآلي من أجل:

- التنبؤ بمستويات التوافر المستقبلية.
- اقتراح أفضل المواقف للسائقين.
- تحسين سياسات التسعير.
- دعم الجهات الإدارية في التخطيط الحضري واتخاذ القرار.

وبذلك، يتحول نظام المواقف الذكية من أداة تشغيلية بسيطة إلى نظام ذكي قائم على البيانات (Data-Driven System) يساهم في تحقيق تنقل حضري أكثر كفاءة واستدامة**[34,31,12].**

6. تشخيص النقائص في السياق الحالي:

رغم الجهود المبذولة في بعض المبادرات المحلية، يظل تسيير مواقف السيارات في الجزائر يعاني من نقائص هيكلية وتشغيلية عميقة تحول دون تحقيق انتقال فعال نحو أنظمة مواقف ذكية متكاملة. وفقاً لآخر البيانات المنشورة من الديوان الوطني للإحصاء (حتى نهاية 2021)، بلغ عدد المركبات التي لوحات تسجيلها دائمة حوالي 7.8 مليون مركبة **[1]**في الديوان، مع استمرار الارتفاع البطيء رغم محدودية الاستيراد**[2].**

وهذا الرقم لا يعكس الحجم الفعلي الكلي للحظيرة الوطنية، مما يعمق الأزمة يوماً بعد يوم ويجعل التحول الجذري نحو أنظمة مواقف ذكية ضرورة ملحة.

وانطلاقاً من الرؤى الحديثة التي طرحت في الفضاءات الابتكارية (مثل مشاريع Smart DZ Hackathon المنظمة من قبل Guiddini برعاية Djezzy)**[12]**، يتضح أن الحل لا يكمن فقط في توفير أماكن إضافية للركن، بل في بناء "نظام بيئي" متكامل (Ecosystem) يربط بين التكنولوجيا (IoT)، الإدارة الرقمية (B2B/B2G)، والجهات الحكومية والخاصة، ليحقق تنسيقاً فورياً وفعالاً.

إلا أن الواقع الحالي يظهر فجوات واضحة تحول دون تحقيق هذا الهدف، ويمكن تشخيص أبرزها كالتالي:

1.6-غياب التنسيق الرقمي والتكامل بين الأطراف:

- لا توجد حتى الآن منصة وطنية أو جهوية موحدة تربط بين المواقف العمومية (البلديات) والخاصة (مراكز تجارية، مستشفيات، فنادق)، مما يؤدي إلى تشتت المعلومات وعدم قدرة السائق على معرفة التوافر الشامل**[34,11].**

- غياب تكامل حقيقي **B2B/B2G** يجعل كل جهة تعمل بشكل منفصل، ويحول دون تبادل البيانات في الوقت الفعلي.

2.6- الاعتماد على الأساليب التقليدية واليدوية:

- معظم مواقف السيارات في المدن الكبرى (الجزائر العاصمة، وهران، قسنطينة) لا تزال تعتمد على الحراس اليدويين أو التذاكر الورقية، مما يؤدي إلى فوضى مرورية، سيطرة المسيرين غير القانونيين، وإهدار وقت ووقود كبيرين أثناء البحث العشوائي.

- قلة استخدام التقنيات الحديثة (حساسات، كاميرات ANPR، IoT) خارج نطاق محدود جداً (مثل بعض المطارات أو مراكز تجارية كبرى)**[16,15,13].**

3.6-ضعف البنية التحتية التقنية واللوجستية:

- محدودية تغطية الإنترنت عالي السرعة في بعض المناطق الحضرية، مما يعيق نقل البيانات في الوقت الفعلي**[19,17].**

- ارتفاع تكلفة تركيب الحساسات أو الكاميرات في ظل الميزانيات المحدودة للبلديات والجهات الخاصة.

- عدم وجود معايير وطنية موحدة للأمان والخصوصية في معالجة بيانات السيارات (لوحات التسجيل، بيانات الموقع)**[25,24].**

4.6- نقص الوعي لدى المستخدمين:

- قلة الوعي بطريقة الدفع الإلكتروني والحجز عبر التطبيقات، خاصة في ظل انتشار الدفع النقدي التقليدي.

- قلة الوعي بفوائد أنظمة المواقف الذكية، مما يقلل من معدلات التبني**[35,11].**

- مقاومة أصحاب المواقف التقليديين للتحول الرقمي خوفاً من فقدان السيطرة أو انخفاض الإيرادات.

5.6- غياب التشريعات والسياسات الداعمة:

- عدم وجود إطار قانوني واضح ينظم استخدام البيانات في أنظمة المواقف الذكية أو يفرض التكامل بين الجهات**[25,24].**

- ضعف الدعم الحكومي للمشاريع الناشئة في مجال المدن الذكية، مما يحد من التوسع السريع للمبادرات مثل **Garini** أو مشاريع **Smart DZ Hackathon.**

6.6 -خلاصة التشخيص:

تكشف هذه النقائص عن فجوة كبيرة بين الواقع الحالي والمتطلبات الضرورية لتحقيق نظام مواقف ذكي فعال. غياب التنسيق الرقمي، الاعتماد على الأساليب التقليدية، ضعف البنية التحتية، ونقص الثقة والتشريعات، كلها عوامل تجعل الأزمة تتفاقم يوماً بعد يوم، خاصة مع استمرار ارتفاع عدد المركبات**[2,1]**. هذه الثغرات تبرز الحاجة الملحة إلى حلول مبتكرة ومتكاملة، وهو ما يقدمه مشروع -**SmartParking Real Time Parking Management System** كرؤية عملية لمعالجة هذه النقائص من خلال نموذج هجين يجمع بين الحلول الاقتصادية والتقنية المناسبة للسياق الجزائري.

7-الحلول المقترحة عبرمشروع SmartParking Real-Time Parking Management System:

انطلاقًا من التشخيص السابق لواقع تسيير مواقف السيارات في الجزائر، وما تم رصده من نقائص تنظيمية وتقنية، يأتي مشروع **SmartParking Real-Time Parking Management System**كحل عملي يهدف إلى تحسين هذا الواقع من خلال اعتماد منصة رقمية ذكية قابلة للتطبيق في السياق المحلي. يعتمد المشروع على فكرة **النظام البيئي المتكامل** الذي يربط بين مختلف الأطراف (السائقون، أصحاب المواقف، الجهات الإدارية)، مع مراعاة الجانب الاقتصادي وسهولة التنفيذ**[34,22]** .

1.7- الرؤية العامة للمشروع:

مشروع SmartParking Real-Time Parking Management System لا يقتصر على كونه تطبيقا لحجز المواقف، بل هو منصة رقمية متكاملة تتكون من:

- **تطبيق موبايل** موجه للسائقين**[23]**.
- **موقع ويب إداري** موجه لأصحاب المواقف والمشغلين والبلديات**[22].**

ويهدف المشروع إلى:

- تقليل الوقت المهدور في البحث عن موقف شاغر.
- تحسين استغلال المواقف المتاحة وتقليل نسبة الفراغ.
- زيادة إيرادات أصحاب المواقف.
- المساهمة في تخفيف الازدحام المروري وتقليل الانبعاثات**[35,11].**
- إنشاء نظام قابل للتوسع مستقبلًا على المستوى المحلي أو الوطني.

2.7- الحلول التقنية والتشغيلية المقترحة:

1.2.7- رقمنة الحجز ومعرفة التوافر في الوقت الفعلي:

يقترح المشروع رقمنة عملية الحجز من خلال:

- استخدام تقنية التعرف الآلي على لوحات السيارات (**ANPR**) أو التحديث اليدوي من قبل المشغلين كحل أولي منخفض التكلفة.
- عرض حالة التوافر بشكل فوري داخل التطبيق باستخدام تقنيات التحديث اللحظي.
- توجيه السائق إلى أقرب موقف شاغر عبر خوارزميات توجيه ذكية **[32,21,20].**

2.2.7- نموذج حجز مرن وخدمات ديناميكية:

يعتمد **SmartParking Real-Time Parking Management System** على نموذج حجز مرن يشمل:

- حجز قصير المدى، حجز مميز، أو اشتراكات طويلة المدى.
- تطبيق تسعير ديناميكي بسيط يعتمد على مستوى الطلب والتوافر.

1- توفير خدمات إضافية مثل الإشعارات، تمديد مدة الركن، اقتراح بدائل، وتقييم الخدمة**[28,27,26].**

3.2.7-واجهات المستخدم والإدارة:

- **تطبيق الموبايل** :يوفر واجهة سهلة الاستخدام، خريطة تفاعلية، توجيه، ودفع إلكتروني آمن**[23].**
- **موقع الويب الإداري** :يسمح لأصحاب المواقف بإدارة المساحات، الأسعار، الحجوزات، والاطلاع على تقارير الإيرادات ونسب الاستغلال**[22].**

4.2.7-نماذج الأعمال المعتمدة:

يعتمد المشروع على نماذج أعمال متعددة، من بينها:

- **B2C**: تقديم الخدمة مباشرة للسائقين.
- **B2B**: تقديم حلول إدارة ذكية لأصحاب المواقف.
- **B2G**: التعاون مستقبلاً مع البلديات لإدارة المواقف العمومية.
- **C2B2C**: كخيار مستقبلي لتأجير المواقف الخاصة الفردية**[3,4,6].**

5.2.7-حلول ملائمة للسياق الجزائري:

يراعي مشروع **SmartParking Real-Time Parking Management System** الخصوصيات المحلية من خلال:

- تقليل التكاليف الأولية عبر حلول تقنية بسيطة في المرحلة الأولى.
- الاعتماد على خوادم محلية أو منخفضة التكلفة.
- الالتزام بقانون حماية البيانات الجزائري(07-18) **[25,24]**.
- دعم الدفع عند الوصول إلى جانب الدفع الإلكتروني لزيادة ثقة المستخدمين.

6.2.7- الفوائد المتوقعة من مشروع SmartParking Real-Time Parking Management System:

من المنتظر أن يحقق المشروع عدة فوائد، من أهمها:

- تقليل وقت البحث عن موقف بنسبة ملحوظة.
- رفع نسبة استغلال المواقف وزيادة العائدات.
- تخفيف الازدحام المروري داخل المدن.
- توفير قاعدة بيانات تساعد في التخطيط الحضري مستقبلاً.

الخاتمة:

في ظل التحديات المتزايدة المرتبطة بتسيير مواقف السيارات في المدن الجزائرية، يطرح مشروع SmartParking Real-Time Parking Management System حلا رقميا متكاملا يعتمد على توظيف التقنيات الذكية لتحسين استغلال الفضاءات، تقليص الازدحام المروري، ورفع كفاءة الخدمات المقدمة. ومن خلال الجمع بين منصة رقمية موجهة للمستخدمين وأدوات إدارية متقدمة، إضافة إلى خدمات ديناميكية مرنة، ينسجم المشروع مع متطلبات التحول الرقمي ويعكس توجهاً عملياً نحو دعم نماذج المدن الذكية في السياق الجزائري.

وبذلك، يشكل SmartParking Real-Time Parking Management System نموذجاً قابلا للتوسع والتكييف، يساهم في تحسين جودة التنقل الحضري وتعزيز فعالية إدارة المرافق العمومية والخاصة.

الفصل الثاني: الإطار المنهجي للدراسة وعرض نتائج الدراسة :

1-مقدمة:

بعد أن تناول الفصل الأول دراسة الموجود لتسيير مواقف السيارات في الجزائر، وكشف النقائص التنظيمية والتقنية التي تعيق التحول نحو أنظمة مواقف ذكية، يأتي هذا الفصل ليحدد الإطار المنهجي الذي يوجه عملية البحث والتحقق من المشكلة المطروحة، مع عرض النتائج الميدانية التي تم الحصول عليها وتحليلها.

يهدف الفصل إلى وضع استراتيجية بحثية علمية دقيقة تمكن من جمع البيانات اللازمة وتحليلها بموضوعية، بما يتناسب مع طبيعة مشروع **SmartParking Real-Time Parking Management System**كحل تقني تطبيقي، وخصوصيات السياق الجزائري (البنية التحتية المحدودة، مستوى الوعي الرقمي، والتشريعات المتعلقة بحماية البيانات).

وسيتناول هذا الفصل النقاط التالية:

1-تعريف منهج البحث وأنواعه الرئيسية.

2-تحديد المنهج البحثي المعتمد ومبررات اختياره.

3-وصف الأدوات وإجراءات جمع البيانات المستخدمة.

4-عرض وتحليل النتائج الكمية والنوعية.

2- تعريف منهج البحث و انواعه الرئيسية:

1.2-تعريف منهج البحث:

يعرف منهج البحث بأنه الخطة أو الاستراتيجية التي يعتمدها الباحث لإجراء الدراسة، والتي توجهه في عملية جمع البيانات وتنظيمها وتحليلها للوصول إلى نتائج علمية دقيقة[36].
ويعد اختيار المنهج البحثي المناسب من أهم القرارات المنهجية في أي دراسة علمية، لأنه يحدد مسار الدراسة، وأدواتها، وطريقة تحليل البيانات، وكيفية تفسير النتائج[36].
2.2-أنواع مناهج البحث:
تنقسم مناهج البحث البحث العلمي الى ثلاثة أنواع رئيسية:

1.2.2-المنهج النوعي(Qualitative Research Approach):

يستخدم للحصول على فهم عميق للظاهرة المدروسة من خلال استكشاف التجارب والمواقف الذاتية للافراد،باستخدام أدوات كالملاحظة والمقابلات ومجموعات النقاش،يجيب هذا المنهج على اسئلة من نوع :"لماذا؟"و"كيف؟"[36].

2.2.2-المنهج الكمي(Quantitative Research Approach):

يستخدم لاختبار الفرضيات وقياس المتغيرات من خلال التحليل الإحصائي للبيانات العددية المجمّعة عبر الاستبيانات أو التجارب.

يُجيب هذا المنهج على أسئلة من نوع: "كم؟" و"بأي نسبة؟"[36].

3.2.2-المنهج المختلط(Mixed Research Approach):

هو منهج يجمع بين المنهجين الكمي والنوعي في دراسة واحدة، مما يتيح فهماً أشمل وأدق للمشكلة البحثية. يعترف هذا المنهج بحدود كل منهج على حدة، ويسعى إلى الاستفادة من مزايا كليهما لتعزيز مصداقية النتائج وصلاحيتها[36].

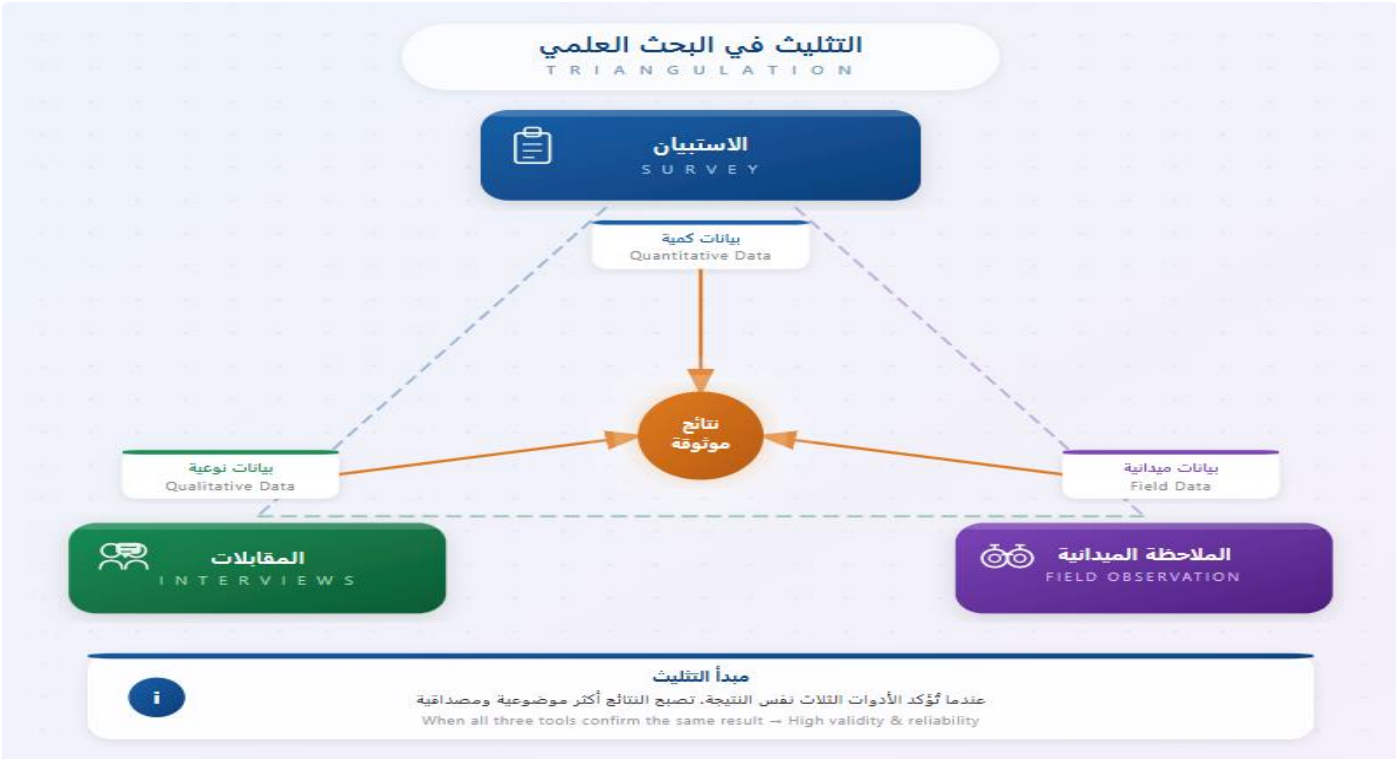
3-المنهج البحثي المعتمد ومبررات اختياره:

اعتمدت هذه الدراسة المنهج االمختلط(**Mixed Research**) الذي يدمج بين الاساليب الكمية والنوعية في مراحل جمع البيانات وتحليلها وتفسيرها .وقد جاء هذا الاختيار مدفوعا بعدة اعتبارات:

1.3-طبيعة المشكلة المطروحة:يتضمن مشروع **SmartParking Real-Time Parking Management System** جانبا تقنيا يستدعي قياسات كمية دقيقة(كمدة البحث عن الموقف، ونسب الاستعداد لاستخدام التطبيق)،وجانبا انسانيا يستلزم فهما نوعيا لسلوك المستخدمين ومقاومة التحول الرقمي.

2.3- التثليث (Triangulation): يعد التثليث من أبرز المزايا التي يوفرها المنهج المختلط، إذ يقوم على مبدأ التحقق من صحة النتائج عبر توظيف أكثر من أداة بحثية في آن واحد، بدلا من الاعتماد على مصدر واحد قد يشوبه القصور أو الانحياز[36].

وفي سياق مشروع **SmartParking Real-Time Parking Management**تحديدا، يتجلى التثليث من خلال ثلاثة مسارات متكاملة: فبينما يكشف الاستبيان عن الأرقام والنسب المتعلقة بمعاناة السائقين (كمتوسط وقت البحث عن موقف، ونسب الاستعداد لاستخدام التطبيق)، تضيف المقابلات مع أصحاب المواقف بعدا نوعيا أعمق يتعلق بواقع الإدارة اليدوية ومدى الاستعداد للتحول الرقمي، في حين توفر الملاحظة الميدانية صورة حية ومباشرة عن السلوكيات الفعلية للسائقين وآليات تسيير المواقف على أرض الواقع.



الشكل1.2:مخطط يوضح مبدأ التثليث(Triangulation) في البحث العلمي

فإذا أسفرت هذه الأدوات الثلاث عن نتائج متقاربة ومتوافقة، كان ذلك دليلا قاطعا على موثوقية النتائج ودقتها. أما إذا تباينت النتائج، فإن ذلك يدفع الباحث إلى التعمق أكثر في التحليل وإعادة النظر في الفرضيات، مما يجعل البحث في مجمله أكثر صرامة علمية وأبعد عن الاستنتاجات المتسرعة.

4- طرق البحث وإجراءات جمع البيانات:

يشير جمع البيانات في البحث العلمي إلى مجموعة التقنيات والأدوات التي يستخدمها الباحث للحصول على المعلومات اللازمة للإجابة عن أسئلة الدراسة وتحليلها.

وبما يتوافق مع المنهج المختلط المعتمد في هذه الدراسة، فقد تم الاعتماد على ثلاثة طرق او أساليب رئيسية لجمع البيانات، وهي:

1.4-الاستبيان الالكتروني(Electronic Questionnaire):

تم تصميم الاستبيان بهدف جمع بيانات كمية من السائقين في الجزائر حول واقع تجربتهم مع مواقف السيارات. وقد تضمن الاستبيان مجموعة من المحاور التي تغطي الجوانب التالية:

- المعطيات الديمغرافية للمشاركين مثل: المدينة، وتكرار استخدام السيارة.
- تجربتهم اليومية في البحث عن موقف للسيارة، بما في ذلك الوقت المستغرق في البحث عن موقف ومناطق الازدحام الأكثر شيوعا.
- مستوى رضاهم عن الحلول الحالية لتسيير المواقف، ومدى استعدادهم لاستخدام تطبيق ذكي لإدارة المواقف.
- تقييمهم للميزات المقترحة في مشروعSmartParking Real-Time Parking Management System ، مثل الحجز المسبق، والدفع الإلكتروني، والتوجيه نحو المواقف المتاحة.
- العوائق المحتملة التي قد تحول دون استخدام التطبيق، مثل ضعف الثقة في الأنظمة الرقمية أو المخاوف المتعلقة بالتكلفة.

2.4-المقابلات(Interviews) :

أُجريت مقابلات شبه منظمة مع عينة من الأطراف المرتبطة بموضوع الدراسة، وذلك بهدف الحصول على فهم أعمق لواقع تسيير مواقف السيارات والتحديات المرتبطة به. وقد شملت المقابلات الفئات التالية:

- **أصحاب ومشغلو مو اقف السيارات:**

وذلك لفهم آليات الإدارة الحالية للمواقف، ومعرفة مدى اعتمادهم على التسيير اليدوي، واستكشاف مدى استعدادهم للتحول نحو أنظمة رقمية، بالإضافة إلى تحديد احتياجاتهم المحتملة من لوحة التحكم الإدارية (Dashboard) في نظام .**SmartParking**

- **سائقون مستخدمون لمو اقف السيارات:**

تم إجراء مناقشات قصيرة مع بعض السائقين الذين يستخدمون المواقف بشكل يومي، بهدف التعرف على الصعوبات التي يواجهونها أثناء البحث عن موقف، ومدى اهتمامهم باستخدام تطبيق يساعدهم في العثور على مواقف متاحة.

3.4-الملاحظة الميدانية(Observational Techniques):

اعتمدت الملاحظة الميدانية اسلوبا تكميليا يرمي الى رصد السلوكيات والظواهر في بيئتها الطبيعية دون تدخل مباشر،وقد تمثل ذلك في :

-زيارة مواقف السيارات في المدن الكبرى (الجزائر العاصمة)لرصد أليات الإدارة الحالية وسلوك السائقين.

-مراجعة تطبيقات مشابهة ك **ParkWhiz**و **SpotHero** لإستخلاص الممارسات الجيدة**[38,37]**.

5-عرض وتحليل النتائج الكمية والنوعية:

1.5-التحليل الكمي(Quantitative Analysis) :

تم تحليل البيانات الكمية المجمعة من الاستبيان باستخدام أدوات التحليل الإحصائي المتاحة في منصة Google Forms، والتي تسمح بحساب النسب المئوية وعرض النتائج في شكل مخططات بيانية تلقائية مثل المخططات الدائرية والعمودية.

وقد ساعدت هذه الأدوات في تنظيم الإجابات الواردة من المشاركين واستخلاص المؤشرات الإحصائية المتعلقة بتجربة السائقين مع مواقف السيارات، مثل توزيع المشاركين حسب المدن، ومتوسط استخدام السيارات، والصعوبات المرتبطة بالبحث عن موقف.

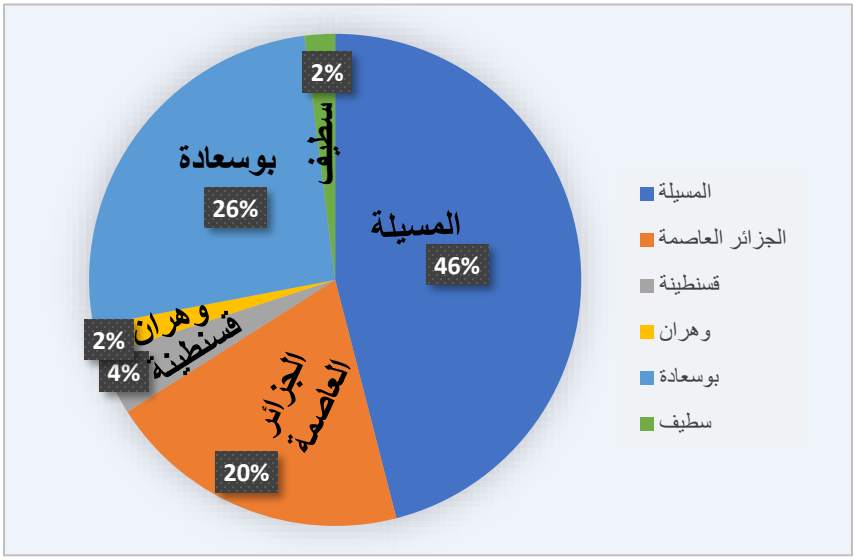
وفيما يلي عرض لأبرز نتائج التحليل الكمي المستخلصة من بيانات الاستبيان.

1.1.5- توزيع المشاركين حسب الولايات:

تظهر نتائج الاستبيان، الذي شارك فيه 50مشاركا، أن غالبية المشاركين ينتمون إلى ولاية المسيلة بنسبة 46%،(ما يعادل 23مشاركا من إجمالي العينة)، تليها بوسعادة بنسبة 26% (حوالي 13مشاركا)، ثم الجزائر العاصمة بنسبة 20% حوالي 10مشاركين

أما بقية المشاركين فقد توزعوا على ولايات أخرى بنسب أقل، حيث بلغت نسبة المشاركين من قسنطينة حوالي 4% (مشاركان)، في حين سجّلت كل من وهران وسطيف نسبة 2% لكل منهما (مشارك واحد لكل ولاية).

ويعكس هذا التوزيع الجغرافي تنوع عينة الدراسة، مع تركز أكبر للمشاركين في ولاية المسيلة والمناطق المجاورة، وهو ما يوفر مؤشرات أولية حول واقع مشكلة مواقف السيارات ومدى الحاجة إلى حلول رقمية مثل نظام SmartParking.



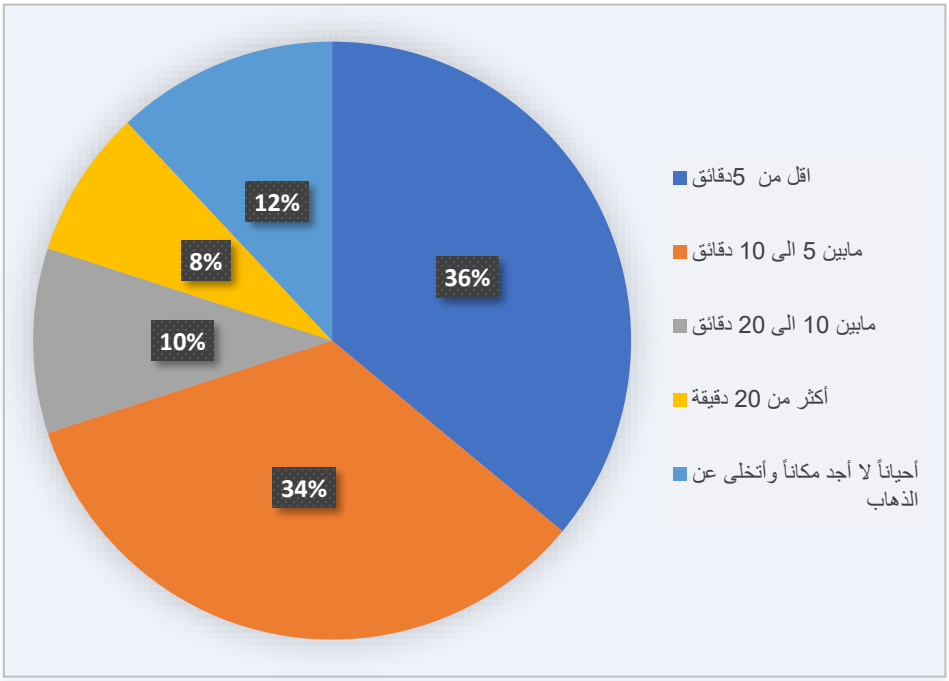
الشكل 6.1: دائرة نسبية توضح نسب المشاركين في الاستبيان عبر الولايات

2.1.5-متوسط الوقت المستغرق للبحث عن موقف:

كشفت البيانات أن 36%من السائقين يجدون موقفًا في أقل من 5 دقائق، بينما يستغرق 34% منهم ما بين 5 إلى 10 دقائق. أما 12% فيقضون 10الى20 دقيقة في البحث، و8% فقط يتجاوز وقتهم 20 دقيقة. في المقابل، أشار 10% من المشاركين إلى أنهم أحيانا لا يجدون مكانا للوقوف أبدا (أو يضطرون للبحث طويلاً جدًا).

هذا يعني أن أكثر من 70% من السائقين (36% زائد 34%) يجدون موقفا في مدة زمينة على الأكثر 10 دقائق في أغلب الأحيان، لكن النسبة الكبيرة التي تقضي 5–10 دقائق (34%) زائد الحالات التي تتجاوز 10 دقائق أو لا تجد موقفًا أحيانًا، تُسبب هدرًا زمنيًا ملحوظًا بشكل يومي.

ترجم هذه الأرقام إلى خسائر اقتصادية وبيئية واجتماعية كبيرة (استهلاك وقود، تلوث، توتر، إضاعة وقت)، وهو ما يبرر الحاجة الملحة إلى حل رقمي ذكي لإدارة المواقف وتقليل وقت البحث.

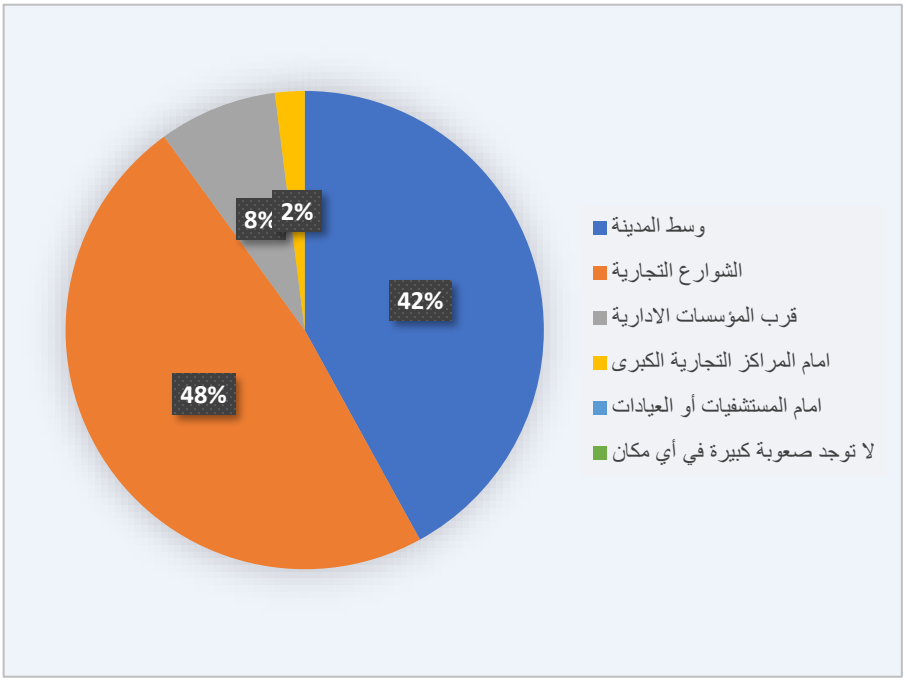


الشكل3.2:دائرة نسبية توضح متوسط الوقت المستغرق للبحث عن مكان ركن شاغر في المدين

3.1.5-أكثر الأماكن التي يواجه فيها السائقون صعوبة متكررة في إيجاد موقف:

كشفت البيانات أن غالبية السائقين يواجهون صعوبة متكررة في إيجاد موقف سيارة في المناطق ذات الكثافة العالية في النشاط، إذ إختار 48% من المشاركين الشوارع التجارية (المحلات، الأسواق، المقاهي والمطاعم) كالمكان الأكثر إرهاقا وتكرارا لهذه المشكلة، في حين يشير 42% من المشاركين إلى وسط المدينة بشكل عام كمصدر رئيسي للصعوبة. أما النسبة المتبقية فتوزعت بشكل ضئيل، حيث لا تتجاوز 8% قرب المؤسسات الإدارية أو الحكومية، ونسب أقل بكثير في الأماكن الأخرى أو عدم وجود صعوبة كبيرة في أي مكان .تترجم هذه الأرقام إلى تركز واضح للمشكلة في

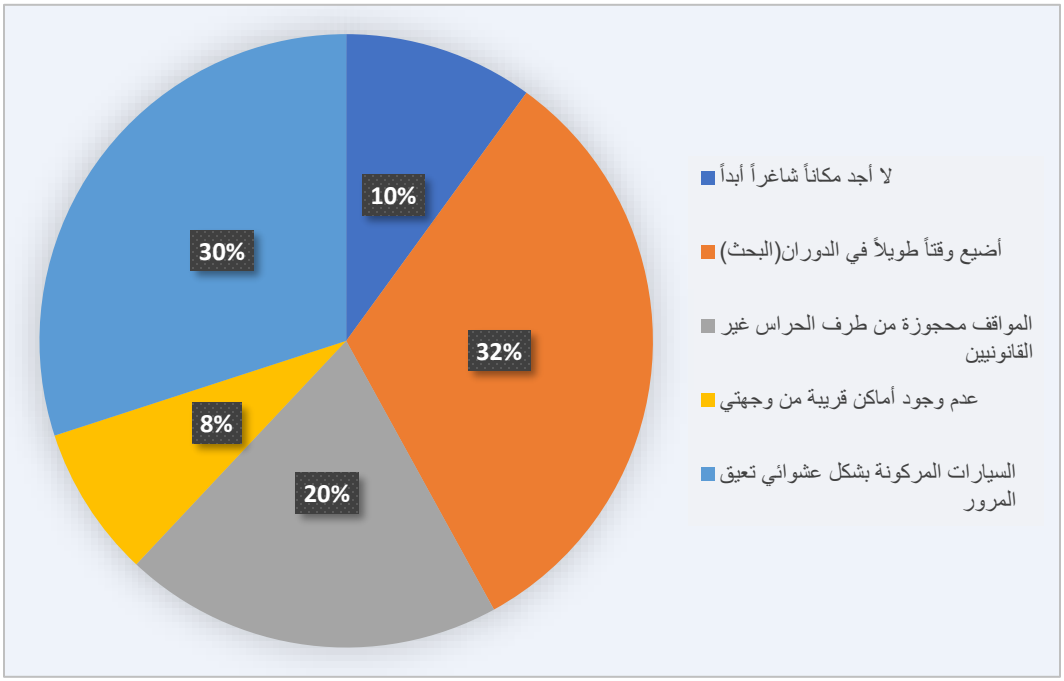
المناطق ذات النشاط التجاري والاقتصادي المكثف، مما يسبب هدرا زمنيا يوميا، ازدحاما مروريا متفاقما، وتأثيرات سلبية اقتصادية وبيئية واجتماعية كبيرة، وهو ما يبرر الحاجة الملحة إلى حلول ذكية تستهدف بشكل أولوي المناطق التجارية ووسط المدن.



الشكل4.2:دائرة نسبية توضح الاماكن التي يواجه فيها السائقون مشكلة في ايجاد مكان للركن

4.1.5- المشاكل الأكثر شيوعا لدى السائقين:

كشفت البيانات أن السائقين يواجهون مجموعة من المشكلات المتكررة أثناء البحث عن مكان لركن السيارة، إذ يُعدّ إضاعة وقت طويل في الدوران المشكلةَ الأبرز بنسبة 32 % من المشاركين، في حين يشير 30 % إلى أن السيارات المركونة بشكل عشوائي تعيق المرور كأحد أكثر العوائق شيوعًا، أما 20 % فيواجهون مشكلة المواقف المحجوزة من طرف حراس غير قانونيين، و8%يرون أن الأماكن القريبة من وجهتهم غير متوفرة، بينما يعتبر 10 % فقط أنهم لا يجدون موقفًا مناسبا أبدا .تترجم هذه النسب إلى واقع يومي يعكس اختلالا واضحا في إدارة المواقف العمومية، حيث يفاقم الركن العشوائي والحجز غير القانوني والنقص في عدد المواقف من صعوبة الوصول إلى الأماكن المرغوبة، مما يولد هدرا زمنيا كبيرا، ازدحاما مروريا متفاقما، وتأثيرات سلبية اقتصادية وبيئية واجتماعية ملحوظة، وهو ما يبرر الحاجة الملحة إلى تدخلات تنظيمية وحلول رقمية ذكية لإعادة ترتيب وتوزيع المواقف بفعالية أكبر.



الشكل5.2:دائرة نسبية توضح أبرز المشاكل التي يواجهها السائقون عند البحث عن مكان ركن

2.5- التحليل النوعي(Qualitative Analysis):

اعتمد التحليل النوعي على عدة أدوات لجمع البايانات تشمل الاستبيانات ذات الأسئلة المفتوحة ، والمقابلات المجراة مع أصحاب المواقف والملاحظات الميدانية وفيما يلي ابرز المحاور التي كشف عنها هذا التحليل:

1.2.5-ضعف الثقة في معاملات الالكترونية:

تشير البيانات النوعية إلى أن ثقافة الدفع الإلكتروني لا تزال محدودة في الجزائر، لا سيما في قطاع الخدمات الحضرية. يفضل كثير من السائقين الدفع النقدي لأسباب عدة: غياب الوعي بأنظمة الدفع الرقمي، خشية الاحتيال الإلكتروني، أو انعدام الثقة في أمان بيانات البطاقات البنكية. ويقترح المشروع معالجة هذا التحدي من خلال توفير خيار الدفع عند الوصول إلى جانب الدفع الإلكتروني في المرحلة الأولى.

2.2.5- غياب التنسيق بين الجهات الحكومية والخاصة:

تكشف الملاحظات الميدانية أن البلديات (التي تدير المواقف العمومية عبر نظام المزايدة العلنية والكراء لمستغلين خاصين) وأصحاب المواقف الخاصة (المراكز التجارية، الفنادق، المستشفيات، إلخ) يعملون بصورة منفصلة ، دون قاعدة بيانات مشتركة أو آلية تنسيق رقمية فعلية على المستوى الوطني أو البلدي.

- نظام الكراء عبر المزاد يمثل شراكة محدودة (B2Gجزئية) في المواقف العمومية فقط، حيث تكتفي البلدية بتحصيل إيجار ثابت، بينما يدير المستغل الخاص الجانب اليومي (بطاقات الاشتراك، التذاكر، الإحصاءات) دون مشاركة بيانات التوافر الحي أو الدخل مع البلدية أو مع مواقف أخرى.

- لا يوجد تكامل مع المواقف الخاصة غير المكراة، والتي تعمل بشكل مستقل تماما (مثل مواقع ParkingDZ ، Garini)التي تركز حصرياً على شبكتها الخاصة**[40,39].**
- هذا الفصل النسبي يعقد عملية توفير معلومات التوافر الشاملة والفورية للسائق (المواقف العمومية والخاصة معاً)، مما يجعل التكامل الرقمي المتقدم) B2G مثل إنشاء منصة مشتركة للبيانات الحية مدعومة بتقنيات إنترنت الأشياء IoT والذكاء الاصطناعي (AIضرورة استراتيجية لنجاح أي مشروع Smart Parking على المدى الطويل.

3.2.5- التدرج في تكلفة الحلول التقنية لأنظمة الانتظار الذكي في السياق الجزائري:

من خلال رصد المبادرات الرقمية الجزائرية، وخاصة مشاريع الهاكاثون مثل Smart DZ Hackathon، يلاحظ أن تطوير أنظمة الانتظار الذكي (Smart Parking)يتطلب مراعاة الاعتبارات الاقتصادية والتمويلية في السياق المحلي .في مشروع SmartParking Ecosystem for Algeriaالمقدم ضمن الهاكاثون، تم اعتماد حلول منخفضة التكلفة في المرحلة الأولى، مثل استخدام حساسات الحركة (motion sensors) للمواقف الداخلية، إلى جانب الكاميرات الذكية والذكاء الاصطناعي للمواقف الخارجية .هذا النهج يعكس وعيا بأهمية البدء بحلول اقتصادية قابلة للتنفيذ، مع إمكانية التطور التدريجي نحو الأتمتة الكاملة المعتمدة على تقنيات IoT و AIالمتقدمة.

وفي هذا الإطار، يمكن النظر في استراتيجيات تدريجية أوسع تشمل، في مراحلها الأولى، آليات تحديث يدوي أو تقنيات مثل ANPR قبل الانتقال إلى الاعتماد الكامل على الحساسات، وهو أمر شائع في العديد من مشاريع المدن الذكية في الدول النامية.

6-النتائج والتوصيات(Research Findings & Recommendations) :

1.6- النتائج الرئيسية للدراسة:

أسفر التحليل المزدوج الكمي والنوعي المجرى على عينة من 50 مشاركاً عن جملة من النتائج الجوهرية التي تعكس واقع إشكالية مواقف السيارات في السياق الجزائري:

1- على صعيد الهدر الزمني :كشفت البيانات أن أكثر من 30% من السائقين يستغرقون ما يزيد عن 10 دقائق في البحث عن موقف يوميا، فيما أفاد 10% بأنهم لا يجدون مكانا في بعض الأحيان. وتترجم هذه الأرقام إلى خسائر اقتصادية وبيئية متراكمة تتمثل في استهلاك الوقود الزائد والتلوث والتوتر اليومي، مما يؤكد الحاجة الملحة إلى حل رقمي يقلص وقت البحث بشكل فعلي.

2- على صعيد مناطق الضغط :تتركز المشكلة بشكل واضح في المناطق التجارية التي أشار إليها 48% من المشاركين، يليها وسط المدينة بنسبة 42%، وهو ما يحدد أولويات التوسع الجغرافي للنظام في مراحله الأولى.

3- على صعيد الأسباب :يفيد 32% من السائقين بأن الدوران المطوّل هو المشكلة الأبرز، فيما يرجع 30% الصعوبة إلى الركن العشوائي الذي يعيق حركة المرور، و20% إلى الحجز غير القانوني من طرف حراس غير رسميين. وتكشف هذه النتائج أن المشكلة مركبة تتشابك فيها أبعاد تنظيمية وقانونية وتقنية في آن واحد.

4- على صعيد الموانع الثقافية والمؤسسية :أظهر التحليل النوعي أن محدودية ثقافة الدفع الإلكتروني وغياب التنسيق بين الجهات البلدية والمواقف الخاصة يمثلان عائقَين بنيويَين أمام أي حل رقمي، ينبغي أخذهما بعين الاعتبار في تصميم النظام واستراتيجية انتشاره.

2.6-التوصيات(Recommendations) :

استنادا إلى المقارنة مع أنظمة مشابهة **(مثل ParkWhiz وSpotHero)** وإلى نتائج الدراسة، يمكن صياغة التوصيات المتعلقة بتصميم وتنفيذ مشروع **SmartParking Real-Time Parking Management System** على النحو التالي:

1- التوصيات التقنية: يوصى باعتماد نهج تدريجي في بناء البنية التحتية، يبدأ بالتحديث اليدوي لحالة المواقف من قبل المشغلين كحل فوري منخفض التكلفة، ثم الانتقال تدريجيا نحو تقنية ANPR في المرحلة المتوسطة، مع التخطيط لاستيعاب حساسات IoT على المدى البعيد. كما يوصى بالبدء بنموذج حد أدنى قابل للتطبيق (MVP) يركز على الوظائف الجوهرية المتمثلة في التوافر والحجز والدفع، قبل الانتقال إلى الميزات المتقدمة، واعتماد منهجية الرشيقة(Agile)في التطوير لضمان المرونة والاستجابة للتغييرات المستمرة في متطلبات المشروع. فضلا عن ذلك، يوصى بتصميم قاعدة بيانات محكمة تضمن سرعة الاستجابة في الوقت الفعلي مع الالتزام بمعايير حماية البيانات الشخصية وفق القانون 18-07، وتوفير نظام إشعارات ذكي لتوجيه السائقين وتذكيرهم بحجوزاتهم في الوقت المناسب.

2- التوصيات المتعلقة بتجربة المستخدم :يوصى بتصميم واجهة بسيطة وبديهية تدعم اللغتين العربية والانجليزية، مع استهداف إتمام عملية البحث والحجز في أقل من دقيقتين. كما يوصى بدعم خيارات دفع متعددة تشمل النقد عند الوصول وبطاقات الذهبية(Edahabia) ، للتكيف مع واقع محدودية الثقة في المعاملات الرقمية الملاحظة من البيانات النوعية.

3- التوصيات المتعلقة بالتوزيع الجغرافي :يوصى بالتركيز في المرحلة الأولى على المناطق التجارية ووسط المدن كونهما الأكثر ضغطا وفق البيانات الميدانية، مع التركيز على المناطق ذات الكثافة الحركية العالية والنشاط الاقتصادي المكثف، باعتبارها البيئة الأنسب لإثبات جدوى النظام وقياس أثره الفعلي قبل التوسع التدريجي نحو مناطق أخرى.

4- التوصيات المتعلقة بالشراكات والحوكمة : يوصى بالسعي نحو إنشاء أليات تواصل مع الجهات البلدية وأصحاب المواقف الخاصة، بهدف بناء قاعدة بيانات موحدة للتوافر تتيح للسائق معلومات شاملة تجمع المواقف العمومية والخاصة في آن واحد، وهو ما يمثل الهدف الاستراتيجي لأي منظومة Smart Parking ناجحة على المدى البعيد.

الخاتمة:

تناول هذا الفصل الإطار المنهجي الذي اعتمدته الدراسة في جمع البيانات وتحليلها، حيث جمع بين المنهج الكمي من خلال الاستبيان الإلكتروني، والمنهج النوعي عبر المقابلات شبه المنظمة والملاحظة الميدانية، ضمن منهجية مختلطة تكاملية. وقد سمح هذا النهج بفهم أعمق وأشمل لإشكالية تسيير مواقف السيارات في السياق الجزائري.

كشف التحليل المزدوج (الكمي والنوعي) عن واقع ميداني معقد يتسم بتشابك أبعاد تقنية وتنظيمية وثقافية واقتصادية، مما يجعل الحل الرقمي المتكامل ضرورة ملحة وليس خيارًا. كما أسفر هذا التحليل عن مجموعة من التوصيات العملية الموجهة لتصميم وتنفيذ نظام **SmartParking Real-Time Parking Management System**، تراعي خصوصيات السياق المحلي من حيث محدودية البنية التحتية، مستوى الثقة في المعاملات الرقمية، والقدرة التمويلي

الفصل الثالث: متطلبات النظام وتصميمه

1-المقدمة:

بعد أن كشف الفصل الأول عن واقع تسيير مواقف السيارات في الجزائر وما يعتريه من نقائص تنظيمية وتقنية كبيرة، وبعد أن قدم الفصل الثاني مجموعة من التوصيات العملية المستخلصة من تحليل ميداني دقيق، يأتي هذا الفصل ليترجم تلك النتائج إلى هيكل تقني واضح وقابل للتنفيذ.

يستعرض هذا الفصل في مرحلته الأولى المتطلبات الوظيفية وغير الوظيفية لنظام **SmartParking - Real-Time Parking Management System**. تشمل المتطلبات الوظيفية الرئيسية وظائف أساسية مثل: الحجز الفوري للمواقف، تتبع توافر الأماكن في الوقت الفعلي، الدفع الإلكتروني، وإدارة المستخدمين. أما المتطلبات غير الوظيفية فتركز على جوانب الأداء، الأمان، الموثوقية، قابلية التوسع، وسهولة الاستخدام.

وقد اعتمدتُ المنهجية الرشيقة (**Agile**) إطاراً رئيسياً لإدارة وتطوير المشروع، لما توفره من مرونة عالية في التكيف مع المتطلبات المتغيرة، والتسليم التدريجي للوظائف حسب الأولويات، مما يضمن تلبية احتياجات المستخدمين بشكل أفضل.

ثم ينتقل الفصل إلى تقديم التصميم المعماري الشامل للنظام من خلال مجموعة من مخططات **UML**، بالإضافة إلى تصميم قاعدة البيانات. كما يتم تفصيل الخيارات التقنية المعتمدة مع مراعاة الخصوصيات الجزائرية، خاصة فيما يتعلق بالتكلفة المنخفضة وقابلية التوسع المستقبلي.

يشكل هذا الفصل الجسر الرابط بين التحليل الميداني والدراسة النظرية من جهة، والتطوير والتنفيذ الفعلي للنظام من جهة أخرى.

2-منهجية إدارة المشروع:

1.2- مفهوم المنهجية الرشيقة ومبادئها:

المنهجية الرشيقة (**Agile Methodology**) هي طريقة لإدارة المشاريع من خلال تقسيمها إلى أجزاء أصغر. تركز على التعاون والتحسين المستمر، حيث تقوم الفرق بالتخطيط والعمل على المشروع ثم مراجعته في دورات متكررة**[41]**.

وبخلاف المنهجيات التقليدية التي تعتمد على التخطيط الشامل والتسليم النهائي الكبير، تعتمد المنهجية الرشيقة على:

- تقسيم المشروع إلى دورات قصيرة (**Sprints**).

- تسليم أجزاء صغيرة وقابلة للاستخدام بشكل متكرر ومستمر (بدلاً من إطلاق كبير واحد).

- التركيز على التعاون بين الفرق والعميل.

- جمع ملاحظات العميل (أو المستخدم النهائي) بشكل مستمر.

- التكيف السريع مع التغييرات والمتطلبات الجديدة حتى في المراحل المتأخرة.

- قياس التقدم من خلال البرمجيات العاملة وليس الوثائق فقط.

2.2-المبادئ الرئيسية للمنهجية الرشيقة(Key Principles of Agile Methodology):

تسترشد المنهجية الرشيقة (Agile Methodology) ب 12 مبدأ أساسيا بسيطا، تساعد الفرق على العمل بكفاءة أعلى والتكيف السريع مع التغييرات وتقديم قيمة مستمرة للعملاء. تركز هذه المبادئ على التعاون والمرونة والتحسين المستمر، وهي**[41]** :

1. رضا العميل (Customer Satisfaction):

تسليم برمجيات قيمة مبكرا وبشكل مستمر.

2. الترحيب بالتغيير (Welcome Change):

قبول التغييرات في المتطلبات حتى في المراحل المتأخرة.

3. التسليم المتكرر (Frequent Delivery):

تسليم برمجيات شغالة في فترات قصيرة ومنتظمة.

4. التعاون (Collaboration):

التعاون اليومي بين فرق الأعمال والمطورين.

5. الفرق المتحمسة او ذات الشغف (Motivated Teams):

بناء المشاريع حول أفراد ماهرين وموثوق بهم ومتحمسين.

6. التواصل وجها لوجه (Face-to-Face Communication):

تفضيل التواصل المباشر لتحقيق وضوح أفضل.

7. البرمجيات العاملة (Working Software):

قياس التقدم بناءً على البرمجيات الوظيفية وليس الوثائق.

8. الوتيرة المستدامة (Sustainable Pace):

الحفاظ على سرعة تطوير ثابتة وقابلة للإدارة.

9. التميز التقني (Technical Excellence):

التركيز على التصميم الجيد وممارسات البرمجة عالية الجودة.

10. البساطة (Simplicity):

تقليل العمل غير الضروري والحفاظ على الحلول بسيطة.

11. الفرق ذاتية التنظيم (Self-Organizing Teams):

تشجيع الفرق على تحمل المسؤولية واتخاذ القرارات بنفسها.

12. التحسين المستمر (Continuous Improvement):

التأمل بانتظام وتحسين العمليات للحصول على نتائج أفضل.

3.2-الفكرة الأساسية للمنهجية الرشيقة(Agile Methodology):

يتم بناء المنتج بشكل تدريجي وتكراري (Iterative & Incremental)، مع اختباره ومراجعته المبكرة مع العميل، وتعديله باستمرار حسب الملاحظات. هذا النهج يقلل المخاطر بشكل كبير، يسرع تقديم القيمة للعميل، ويزيد من فرص نجاح المشروع مع ضمان تلبية الاحتياجات الفعلية**[41]**.



الشكل1.3:صورة توضح دورة حياة التطبيق في المنهجية الرشيقة

4.2- مبررات اختيار المنهجية الرشيقة في هذا المشروع:

تم اختيار المنهجية الرشيقة لهذا المشروع للأسباب التالية:

1- مرونة عالية في مواجهة التغييرات المحتملة في متطلبات نظام إدارة مواقف السيارات.

2- إمكانية تسليم نسخ قابلة للاستخدام في مراحل مبكرة، مما يسمح باختبار النظام ميدانيا بشكل تدريجي.

3- تقليل مخاطر الفشل من خلال الاكتشاف المبكر للمشكلات.

4- مشاركة فعالة لأصحاب المصلحة (مديري المواقف، السائقين، الإدارات البلدية) خلال عملية التطوير.

5- ملاءمة المنهجية للمشاريع ذات الطبيعة التقنية والتي تتطلب حلولاً مبتكرة في السياق الجزائري.

2-المتطلبات الوظيفية(Functional Requirements):

المتطلبات الوظيفية (**Functional Requirements**) هي مجموعة المواصفات التي تحدد الوظائف والخدمات التي يجب أن ينفذها النظام من أجل تلبية احتياجات المستخدمين وتحقيق أهداف المشروع. وهي تشرح بالتفصيل ما الذي يجب على النظام القيام به، أي السلوك المتوقع منه عند التفاعل مع المستخدم أو مع الأنظمة الأخرى[42].

تشمل متطلبات النظام الوظيفية للمنصة الحالية المتطلبات التالية:

1.3-المتطلبات الوظيفية الرئيسية لمستخدم تطبيق الهاتف (Driver):

1. **عرض تو افر المو اقف في الوقت الفعلي(Real-time Availability) :**
   - عرض حالة المواقف (متاح/مشغول) على الخريطة بشكل فوري.
2. **البحث والتصفية عن مو اقف:**
   - البحث حسب الموقع، السعر، المسافة، التوافر.
3. **حجز موقف مسبق أو فوري(Smart Booking) :**
   - حجز موقف لمدة محددة (قصيرة أو طويلة).
   - تعديل أو إلغاء الحجز وفق شروط واضحة.
4. **التوجيه والملاحة(GPS Navigation) :**
   - توجيه السائق إلى الموقف المحجوز عبر خرائط تفاعلية (Google Maps) أو.(OpenStreetMap)
5. **نظام الإشعارات(Notifications) :**
   - إشعارات فورية عند الوصول إلى وقت انتهاء الحجز، توفر مواقف قريبة، أو تغييرات في الحالة.
6. **سجل الحجوزات والمعاملات:**
   - عرض تاريخ الحجوزات السابقة والدفعات.

7. نظام التقييم والتغذية الراجعة(Feedback) :

- تقييم الموقف والخدمة بعد الانتهاء.

2.3- المتطلبات الوظيفية لأصحاب المو اقف والمسؤولين(Admin Dashboard and Manager Dashboard) :

8.إدارة المو اقف:

- إضافة/تعديل مواقف ومناطق.

9.إدارة التو افروالحجوزات:

- تحديث حالة المواقف.
- متابعة الحجوزات الحالية والسابقة.

10.التسعير الديناميكي(Dynamic Pricing) :

- تحديد أسعار ثابتة أو ديناميكية حسب الوقت والطلب.

11.لوحة التحكم والتقارير(Analytics & Reports) :

- تقارير الإيرادات، نسبة الاستغلال، ساعات الذروة، والإحصائيات.

12.إدارة المستخدمين:

- إدارة حسابات أصحاب المواقف.

3.3-المتطلبات الوظيفية العامة (للنظام ككل):

13.نظام الأمان والمصادقة:

- تسجيل الدخول، استعادة كلمة المرور، وإدارة الصلاحيات.

14.دعم متعدد اللغات:

- اللغة العربية والفرنسيةوالإنجليزية اختيارياً.

4-المتطلبات غير الوظيفية(Non-Functional Requirements) :

المتطلبات غير الوظيفية (Non-Functional Requirements) هي مجموعة الخصائص والمعايير التي تحدد كيفية عمل النظام وجودة أدائه، دون أن تصف الوظائف المباشرة التي يقدمها للمستخدم. وتهدف هذه المتطلبات إلى ضمان الكفاءة، الأمان، الاستقرار، وسهولة الاستخدام داخل نظام[42].

تشمل المتطلبات غير الوظيفية للنظام ما يلي:

1- الأداء(Performance):

- يجب أن يستجيب النظام لطلبات المستخدم خلال أقل من 3 ثوانٍ .
- يجب تحديث حالة المواقف في الوقت الفعلي .(**Real-Time**)
- يجب أن يدعم النظام عدداً كبيراً من المستخدمين المتصلين في نفس الوقت دون انخفاض ملحوظ في الأداء .

2- الأمان(Security):

- يجب تشفير جميع البيانات المتبادلة باستخدام بروتوكول HTTPS و TLS.
- يجب حماية حسابات المستخدمين عبر نظام مصادقة آمن .
- يجب عدم تخزين كلمات المرور بصيغتها الأصلية داخل قاعدة البيانات .
- يجب حماية بيانات المستخدمين ولوحات السيارات وفق قانون حماية البيانات الجزائري 18-07 .

3- الموثوقية(Reliability):

- يجب أن يعمل النظام بشكل مستقر دون انقطاعات متكررة .
- يجب حفظ البيانات بشكل تلقائي لتجنب فقدان المعلومات .

4- قابلية التوسع(Scalability):

- يجب أن يكون النظام قابلاً لإضافة تقنيات جديدة ومواقف جديدة ومدن أخرى مستقبلاً .

5- سهولة الاستخدام(Usability):

- يجب أن تكون واجهة التطبيق بسيطة وسهلة الفهم .
- يجب أن يدعم النظام اللغات العربية والفرنسية والإنجليزية .
- يجب أن يتمكن المستخدم من إجراء الحجز خلال خطوات قليلة وبسيطة .

6- التو افق(Compatibility) :

- يجب أن يعمل التطبيق على مختلف أحجام الهواتف الذكية .
- يجب أن يكون موقع الويب متوافقاً مع المتصفحات الحديثة مثل Chrome و Edgeو .Firefox

7- التوفر (Availability):

- يجب أن يكون النظام متاحاً للمستخدمين على مدار 7/24 .
- يجب تقليل فترات التوقف والصيانة إلى الحد الأدنى الممكن .

8-قابلية الصيانة(Maintainability):

- يجب تصميم النظام بطريقة تسهّل تحديثه وإضافة ميزات جديدة .
- يجب تنظيم الكود البرمجي وتوثيقه لتسهيل الصيانة المستقبلية.

5-تصميم قاعدة البيانات:

يُعد تصميم قاعدة البيانات أحد المكونات الرئيسية في بناء أي نظام معلومات، إذ يشكل العمود الفقري الذي يعتمد عليه النظام بأكمله. ويشمل ذلك عملية تصميم وبناء وصيانة قاعدة البيانات، مثل تنظيم المجموعات (**Collections**) والوثائق(**Documents**) ، بالإضافة إلى إدارة العلاقات بينها. ومن الضروري أن تصمم قاعدة بيانات تضمن سلامة البيانات وكفاءة أداء النظام، مع التركيز على الفعالية والأمان والموثوقية العالية.

ويختلف نظام **Firebase Firestore** اختلافًا تاما عن قواعد البيانات العلائقية التقليدية، حيث لا يعتمد على تخزين البيانات في جداول مكونة من صفوف وأعمدة، بل يستخدم الوثائق (**Documents**)كوحدة تخزين أساسية ضمن مجموعات(**Collections**) **[43]** .

تتضمن عملية تصميم قاعدة البيانات عدة مراحل، تشمل تحديد متطلبات البيانات، وتنظيم المجموعات والوثائق، وتحديد بنية البيانات المناسبة، ثم تنفيذها داخل قاعدة البيانات الفعلية. ويعتمد تصميم قاعدة البيانات في هذا المشروع على الكيانات الرئيسية للنظام، وخصائصها، والعلاقات المرجعية بينها بما يتوافق مع متطلبات التطبيق وعمليات التحديث في الوقت الفعلي**[44].**

ولتحسين الأداء وقابلية التوسع، تم اعتماد أسلوب شبه إزالة التكرار **(Semi-Denormalization)** المستخدم عادة في قواعد بيانات NoSQL، حيث يتم تنظيم البيانات بطريقة تقلل عدد عمليات القراءة المعقدة وتدعم المزامنة الفورية للبيانات. كما يتم استخدام قواعد الحماية **(Firebase Security Rules)** وآليات المصادقة لضمان سلامة البيانات واتساقها والتحكم الآمن في الوصول إليها.

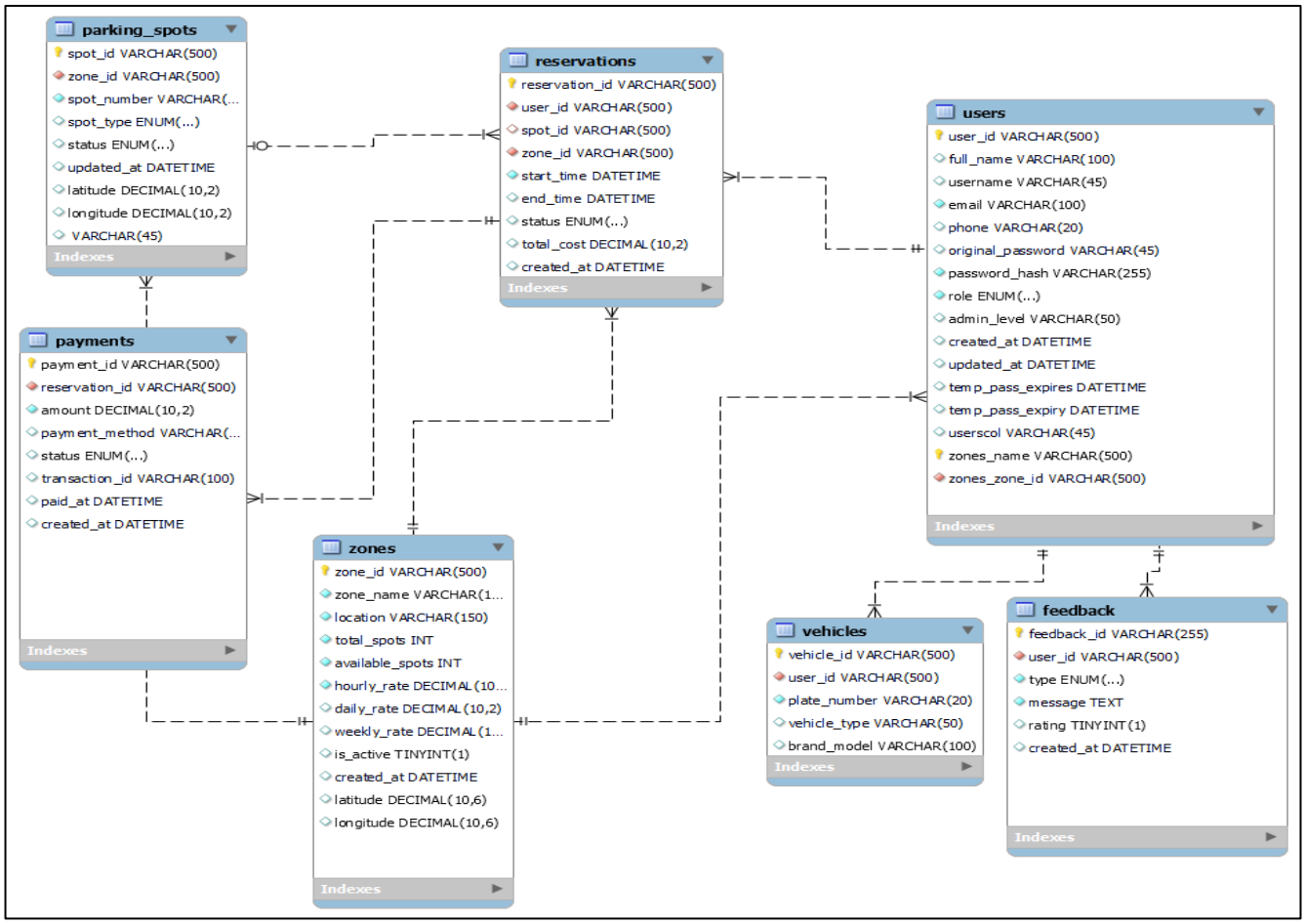
1.5- نمذجة قاعدة البيانات(Database Modeling):

تم تصميم نموذج قاعدة البيانات باستخدام أداة**MySQL Workbench** ، كما هو موضح في الشكل التالي. تُعد **MySQL Workbench** أداةً مرئية تُستخدم لنمذجة وتصميم قواعد البيانات، حيث توفر تمثيلاً بيانياً للكيانات والخصائص والعلاقات المنطقية بينها. وقد استُخدمت الأداة في هذا المشروع لغرض تصميم البنية المنطقية للبيانات وتوضيح كيفية تنظيم المعلومات داخل النظام.

وعلى الرغم من أن النظام يعتمد على**Firebase Firestore** ، وهو نظام قواعد بيانات**NoSQL** ، فقد تم استخدام مخطط الـ **ERD** من أجل تسهيل تصور بنية البيانات والعلاقات المرجعية بين مختلف مكونات النظام**[48]**. وفي **Firestore** يتم تخزين البيانات باستخدام المجموعات

(Collections)والوثائق (Documents) بدلاً من الجداول التقليدية، مع الاعتماد على المراجع (References) ومعرفات الوثائق لربط البيانات فيما بينها) [45].

ويهدف نموذج قاعدة البيانات المصمم إلى تنظيم البيانات الخاصة بالمستخدمين، الحجوزات، المركبات، المواقف، وعمليات الدفع بطريقة تضمن الأداء الجيد، سهولة التوسع، وسلامة البيانات داخل النظام[43].



الشكل 72.3:نموذج قاعدة البيانات

2.5 - مجموعات قواعد البيانات(data base collections) :

اسم المجموعة	الوظيفة
users	هذه المجموعة يتم تخزين فيها معلومات المستخدمين(مدراء-مسؤولين-سائقين) بما في ذلك اسم المستخدم وكلمة المروروعنوان البريد الإلكتروني.
zones	هذه المجموعة يتم تخرين فيها اسم المنطقة ومعلوماتها مثل احداثياتها وكم تحتوي على مو اقف

parking_spots	هذه المجموعة يتم تخزين معلومات مو اقف السيارات (Parking Spots) مثل الموقع، النوع، الحالة، والإحداثيات الجغر افية.
reservations	هذه المجموعة يتم تخزين فيها معلومات الحجوزات على مو اقف السيارات، بما في ذلك وقت البداية
payments	هذه المجموعة يتم تخزين فيها عمليات الدفع المتعلقة بالحجوزات .
vehicles	هذه المجموعة يتم تخزين فيها معلومات المركبات الخاصة بالمستخدمين
feedback	هذه المجموعة فيها تقييمات وآراء المستخدمين بعد استخدام الخدمة (التقييم، التعليق).

جدول 3.71:انواع المجموعات في قاعدة البيانات

6-نظام النمذجة System Modeling:

يعد تصميم معمارية النظام لمشروعنا من الأمور الأساسية، وقد اعتمدنا في ذلك على مخططات لغة النمذجة الموحدة(**UML**) ، والتي هي عبارة عن رسوم بيانية تستخدم لنمذجة أنظمة البرمجيات. حيث تساعد مخططات **UML** في تصور وتحديد وبناء وتوثيق الأنظمة البرمجية المعقدة من خلال تمثيل او نمذجة معماريتها وسلوكها وتفاعلاتها**[46]**.

وبشكل خاص، استخدمنا ثلاثة أنواع من مخططات **UML** في تصميم معمارية النظام وهي :مخططات حالات الاستخدام

(**Use Case Diagrams**) ، ومخططات التسلسل(**Sequence Diagrams**) ، ومخططات الفئات (**Class Diagrams**) ،تمثل مخططات حالات الاستخدام وظائف النظام من وجهة نظر المستخدم، بينما توضح مخططات التسلسل التفاعلات بين مكونات النظام، أما مخططات الفئات فتوضح الهيكل الثابت للنظام والعلاقات بين فئاته**[46]**.

ولإنشاء هذه المخططات، استخدمنا أداة **StarUML**، وهي برنامج متخصص في النمذجة القائمة على**UML** ، يوفر **StarUML** واجهة مستخدم سهلة و بسيطة لإنشاء وتحرير وعرض مخططات**UML** ، كما يدعم مجموعة واسعة من الميزات مثل أنواع مختلفة من المخططات، والتحقق من صحة النموذج، وتوليد الكود، ودعم الإضافات.

1.6-مخططات حالات الاستخدام (Use Case Diagrams):

يعد مخطط حالات الاستخدام أحد مخططات لغة النمذجة الموحدة(**UML**) ، ويستخدم لتوضيح الوظائف والخدمات التي يقدمها النظام للمستخدمين، بالإضافة إلى بيان التفاعل بين الأطراف المختلفة (**Actors**) والنظام. ويساعد هذا المخطط في تحديد متطلبات النظام بشكل واضح من خلال عرض حالات الاستخدام الرئيسية والعلاقات بينها وبين المستخدمين**[46]**.

توضخ مخططات حالات الاستخدام انه يوجد اربعة اطراف رئيسية في النظام وهم :

1-المسؤول:يمثل اعلى سلم الهرم ويمثل والدرجة العليا في نظام ومن ابرز حالات استخدامه:

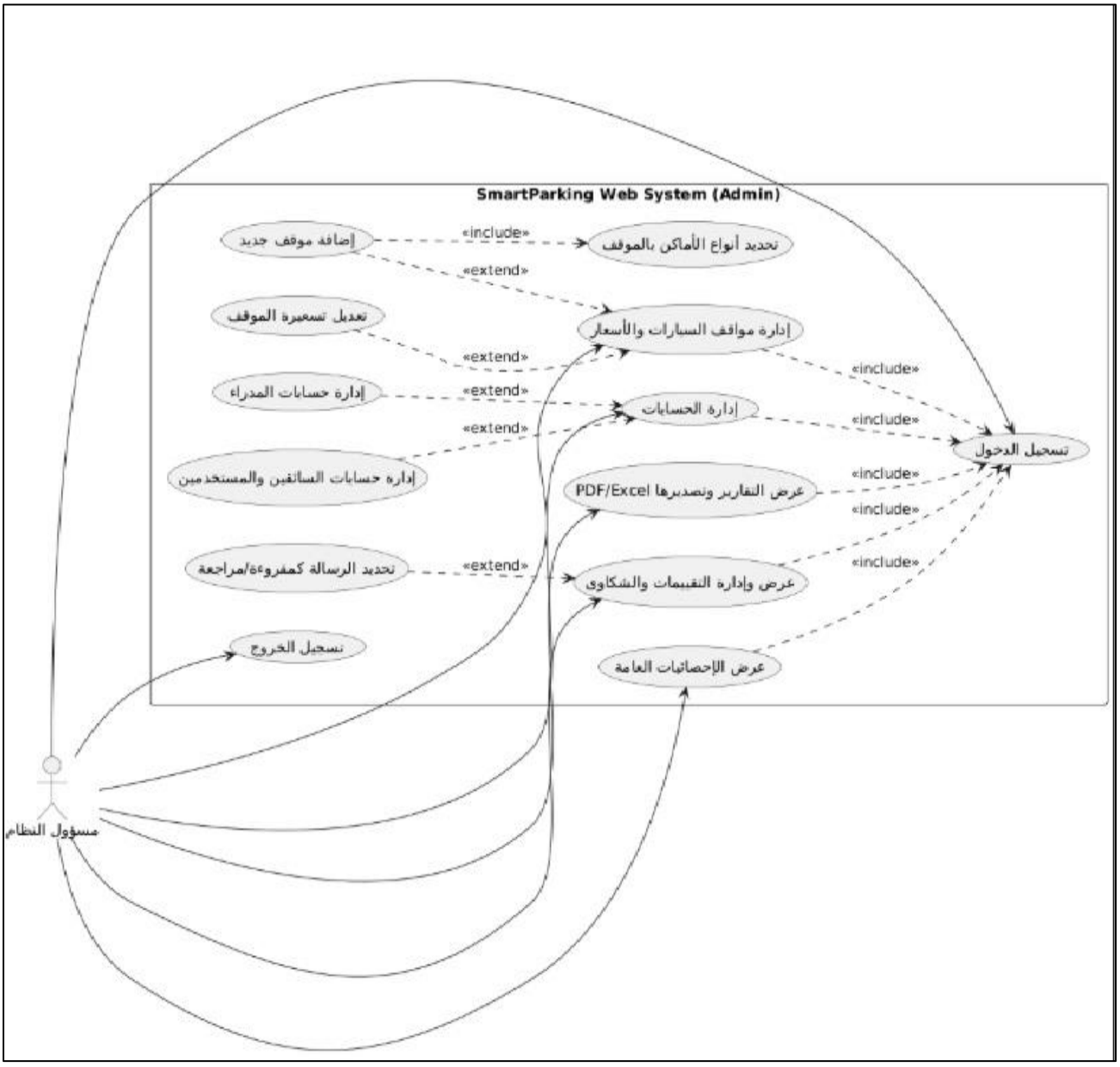
1- إدارة المواقف (اضافة منطقة ،تحديد نوع المواقف ،تعطيل منطقة،اسناد مدير للمنطقة).
2- عرض الاحصائيات(نسبة الإشغال،الايرادات،حركة المرورية).
3- الوصول الى الخريطة والمواقف.
4- ادارة التغذية الراجعة.
5- التقارير واصدارها حسب النوع.
6- إضافة مديرين ومسؤولين.

2-المدير:يمثل المستوى الايداري التشغيلي للمواقف ويشارك في عدة حالات للاستخدام من ابرزها:

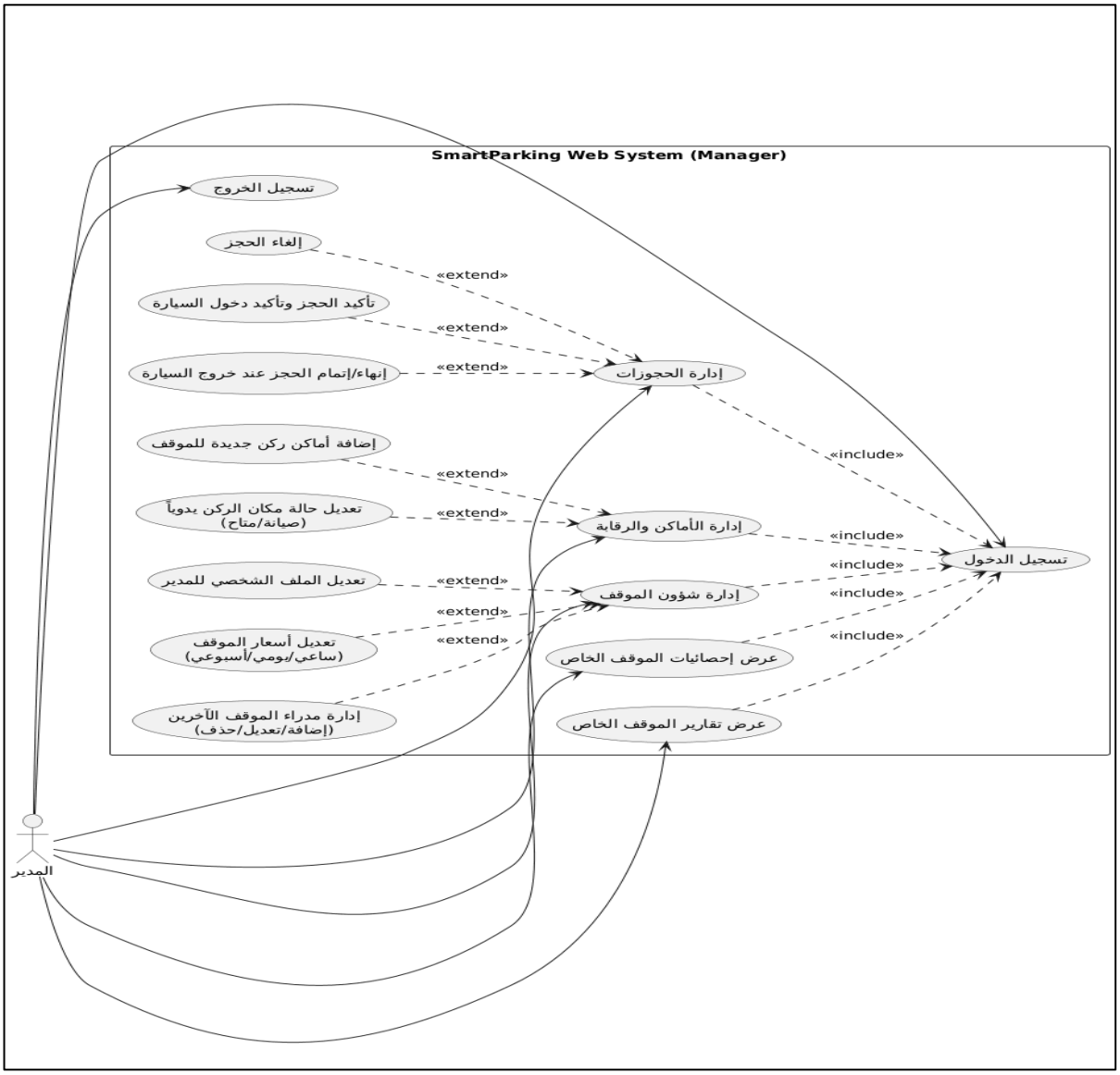
1- إدارة الحجوزات(إلغاء ،تأكيد).
2- عرض الاحصائيات..
3- التقارير وإصدارها حسب النوع.
4- تغيير التسعير المخصص للمواقف.
5- تعديل عدد المواقف في المنطقة.
6- إضافة مدراء الى المنطقة.

3-السائق/ الزائر :يمثل مستخدم التطبيق ويشارك في عدة حالات استخدام من ابرزها:

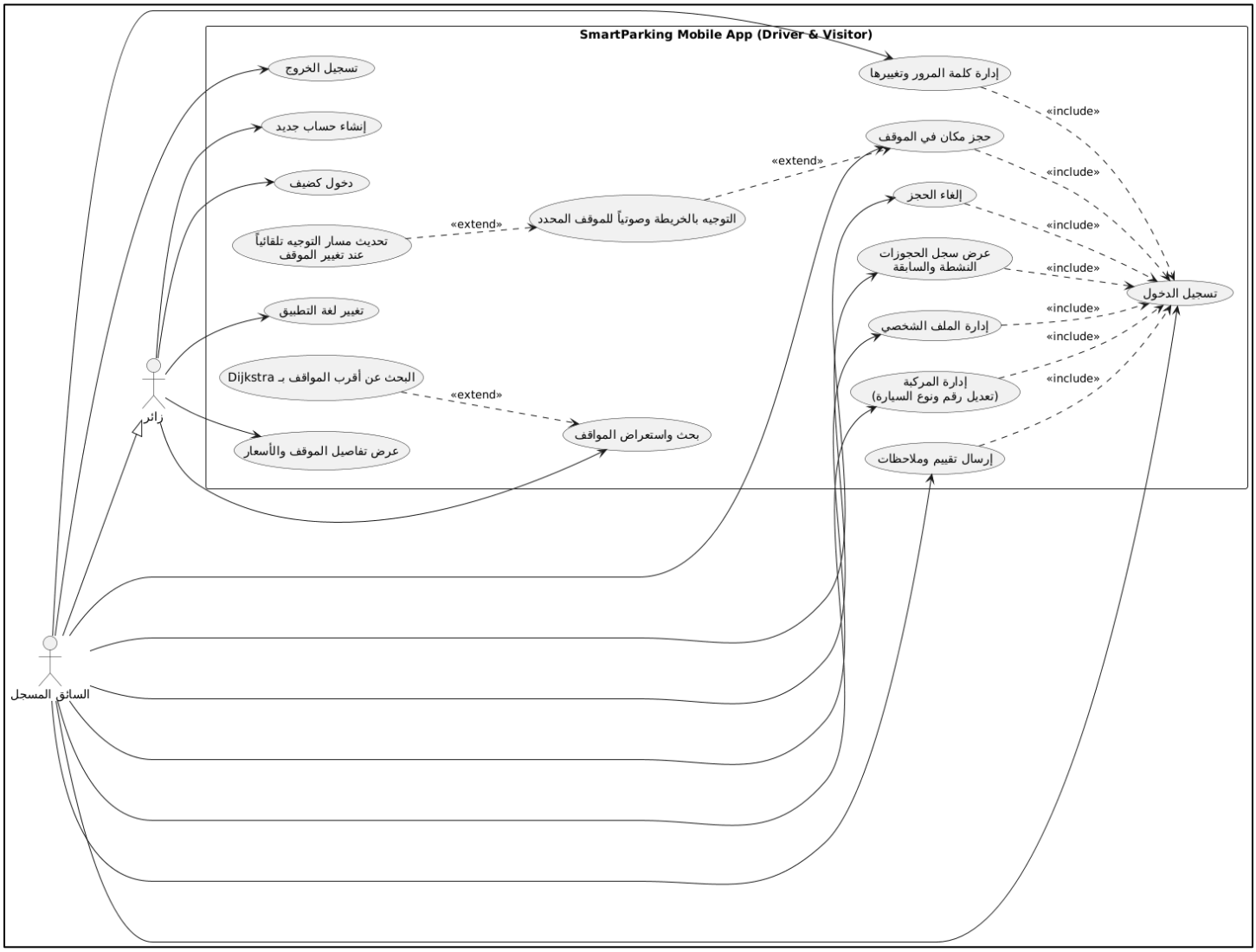
1- الدخول للتطبيق كضيف لمعرفة مزايا التطبيق.
2- إنشاء حساب او تسجيل الدخول للتطبيق.
3- البحث عن مواقف واستعراضها.
4- حجز موقف حسب النوع.
5- إلغاء الحجز.
6- التغذية الراجعة(تقديم اطراء،شكوى،تقييم).



الشكل 7.1:مخطط حالات الاستخدام للمسؤول



الشكل4.3:مخطط حالات الاستخدام للمدير



الشكل 5.3:مخطط حالات الاستخدام للسائق

2.6-مخططات التسلسلية(Sequence Diagrams):

Sequence Diagramهو أحد مخططات السلوك (**Behavioral Diagrams**) في لغة**UML** ، يُستخدم لتوضيح التفاعلات بين الأطراف المختلفة (**Actors**)والكائنات أو المكونات داخل النظام مع مراعاة الترتيب الزمني للرسائل (**Messages**) التي تتبادلها**[46].**

الغرض الرئيسي من المخطط التسلسلي(**Sequence Diagram**) :

1- يبين كيف يتم تنفيذ حالة استخدام (**Use Case**) بشكل تسلسلي.

2- يوضح ترتيب الرسائل بين الممثلين (**Actors**) والنظام أو بين أجزاء النظام الداخلية.

3- يساعد في فهم سيناريوهات النجاح وسيناريوهات الاستثناء.

الأطراف المشاركة في المخطط التسلسلي(**Sequence Diagram**) :

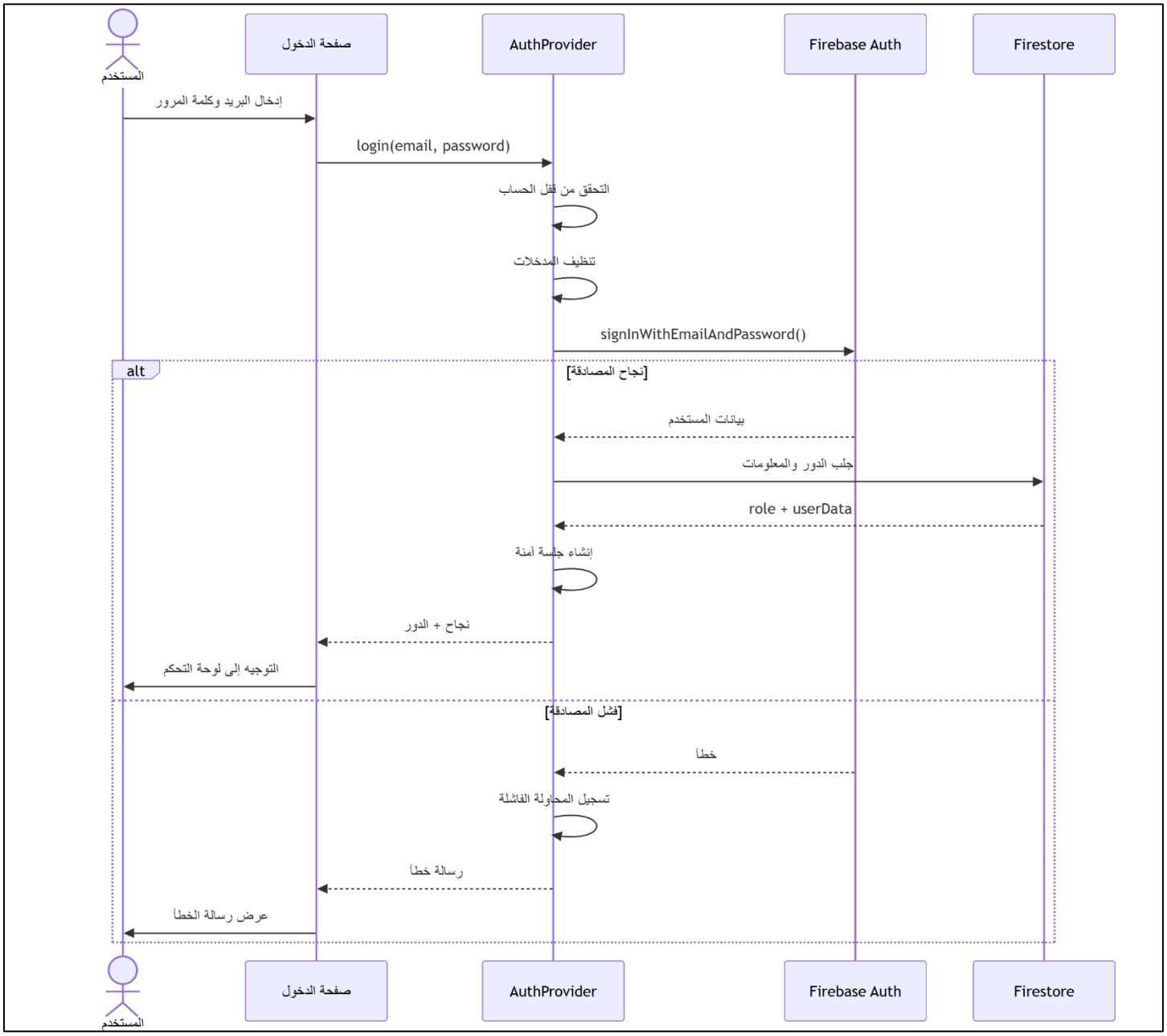
1- المسؤول (**Admin**):يمثل أعلى مستوى إداري في النظام، حيث يمتلك الصلاحيات الكاملة لإدارة المواقف والمستخدمين ومراقبة النظام.

2- المدير (**Manager**):يمثل المستوى الإداري التشغيلي المسؤول عن إدارة مناطق المواقف ومتابعة الحجوزات.

3- السائق / الزائر - (**Driver / Visitor**):يمثل المستخدم النهائي للتطبيق، حيث يمكنه الاستفادة من خدمات البحث والحجز والتفاعل مع النظام.

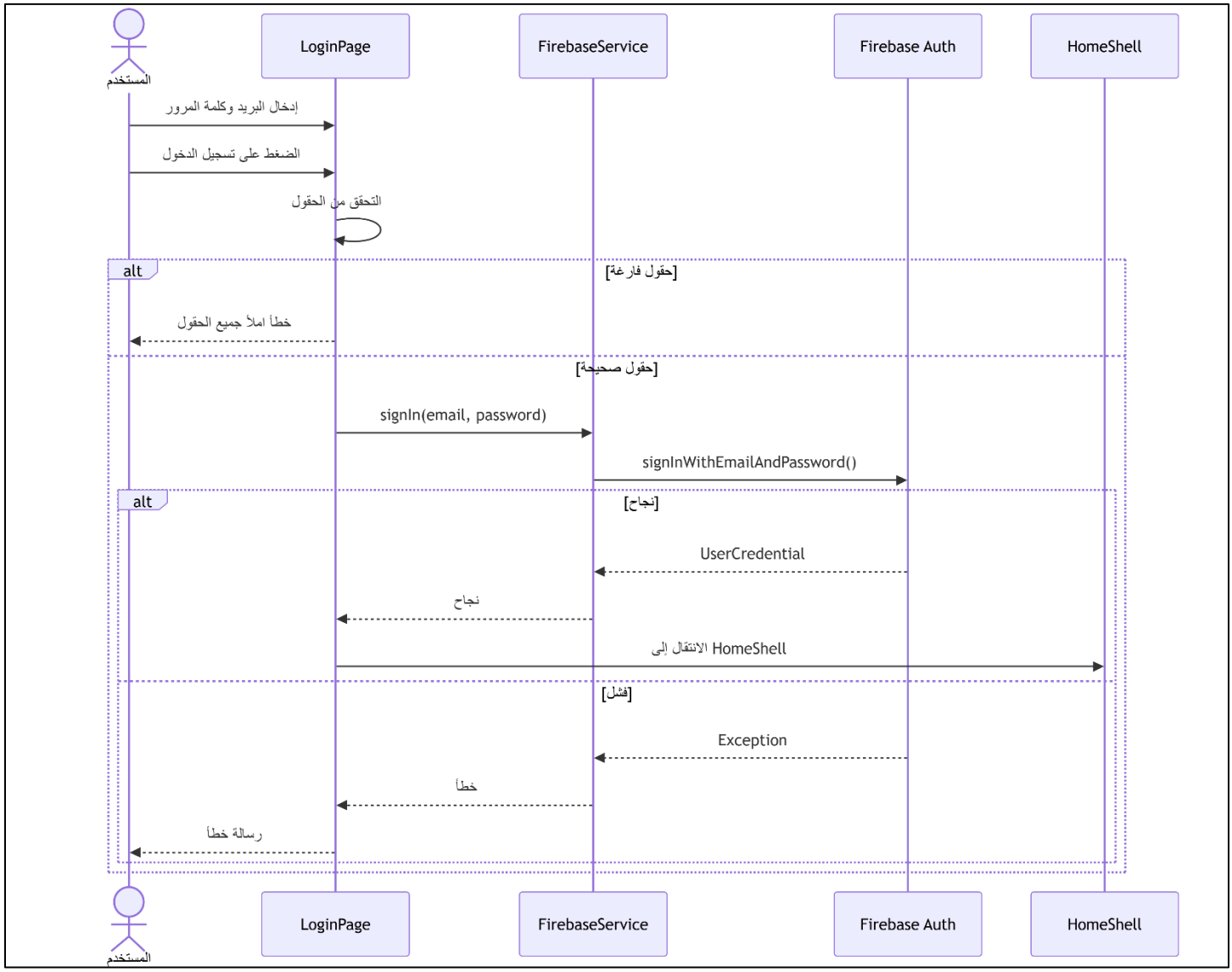
4- النظام (**System**):يمثل الجهة المسؤولة عن معالجة الطلبات وتنفيذ العمليات الداخلية، مثل التحقق من البيانات، إدارة قواعد البيانات، معالجة الحجوزات، وتحديث حالة المواقف بشكل آني.

عملية تسجيل الدخول:



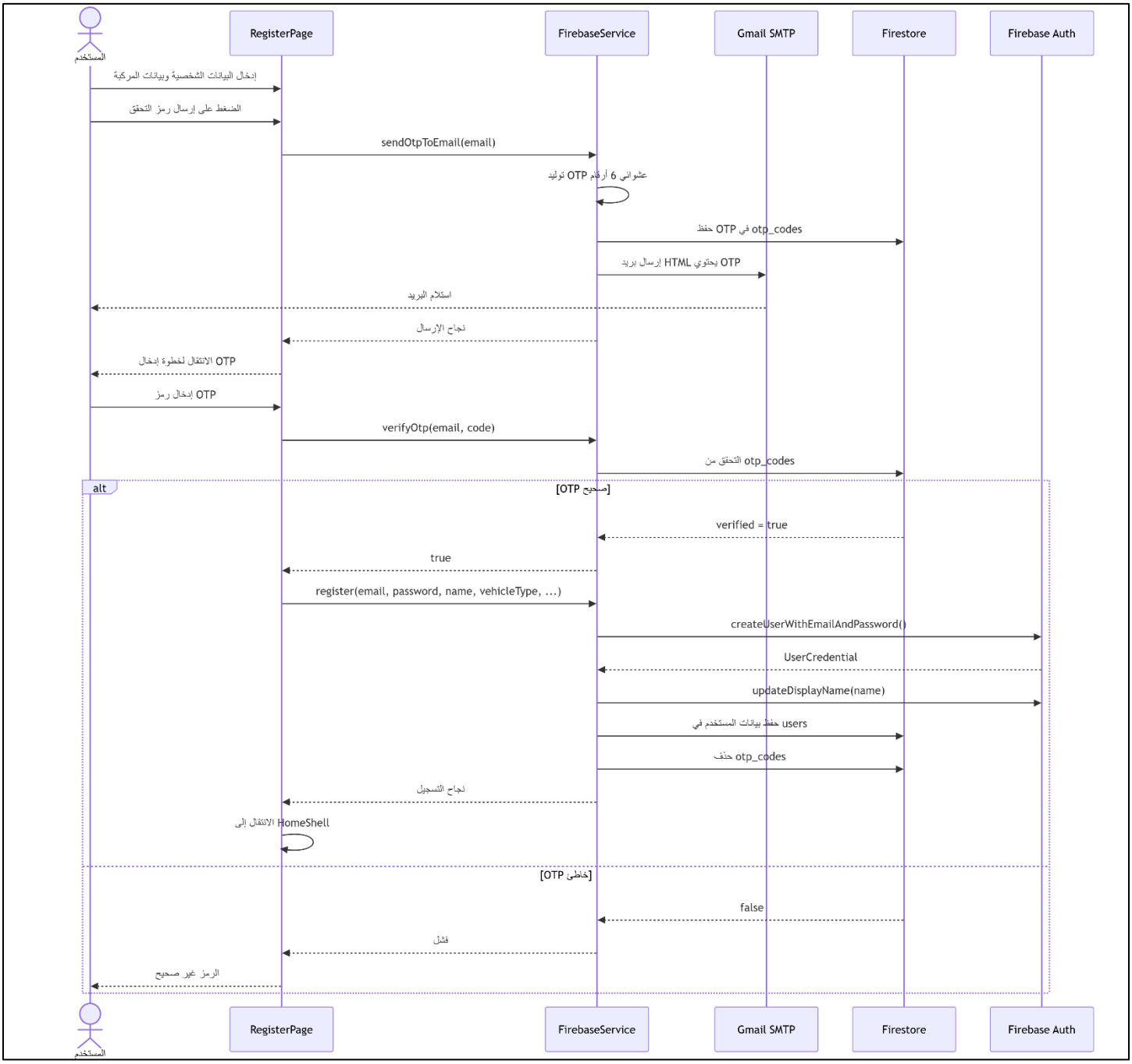
الشكل 6.3:مخطط تسلسلي لعملية تسجيل الدخول.

عملية تسجيل الدخول بالنسبة للسائق:



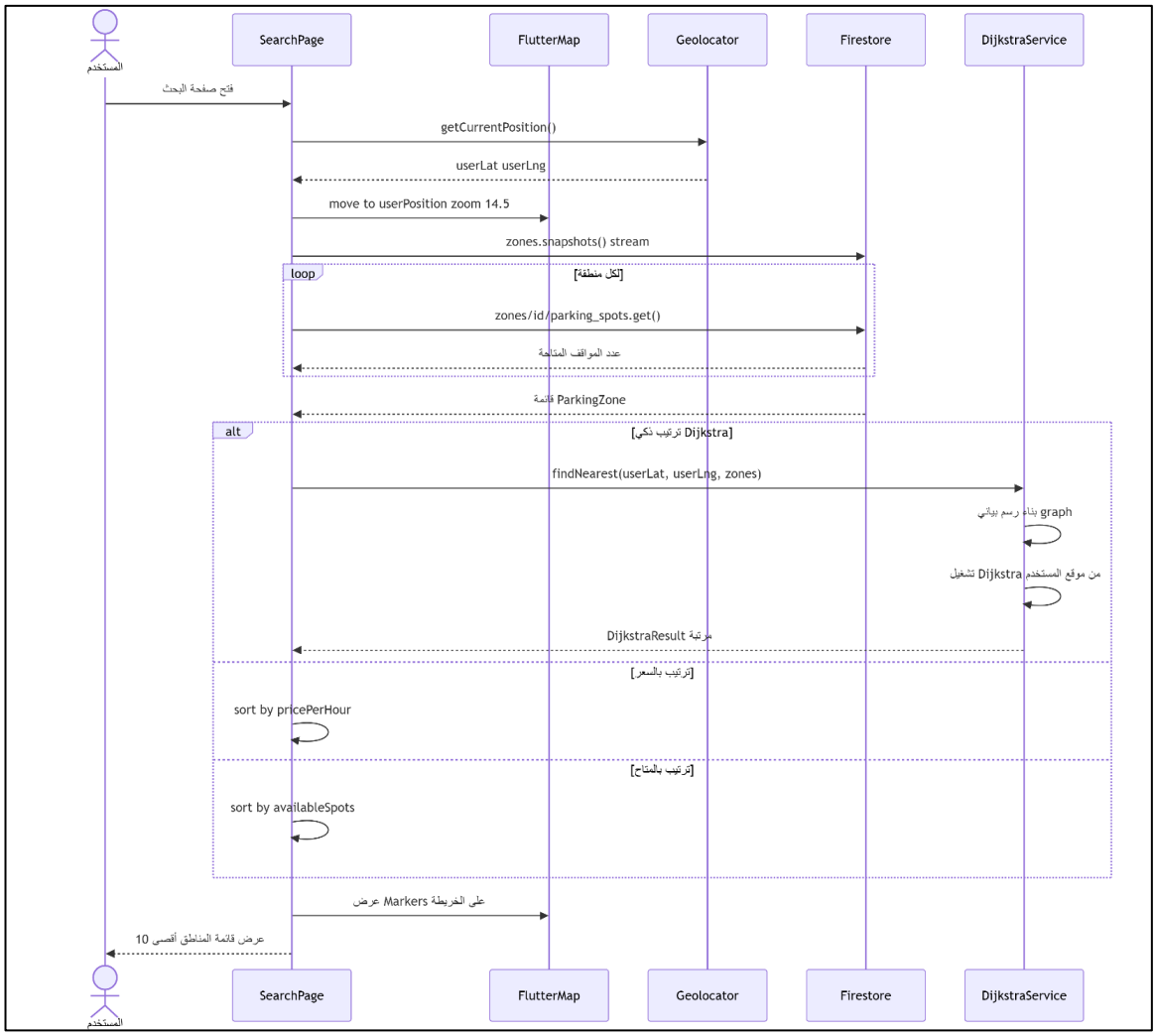
الشكل 7.3:المخطط التسلسلي لعملية تسجيل الدخول للتطبيق بالنسبة للسائق.

عملية فتح حساب بالنسبة للسائق:



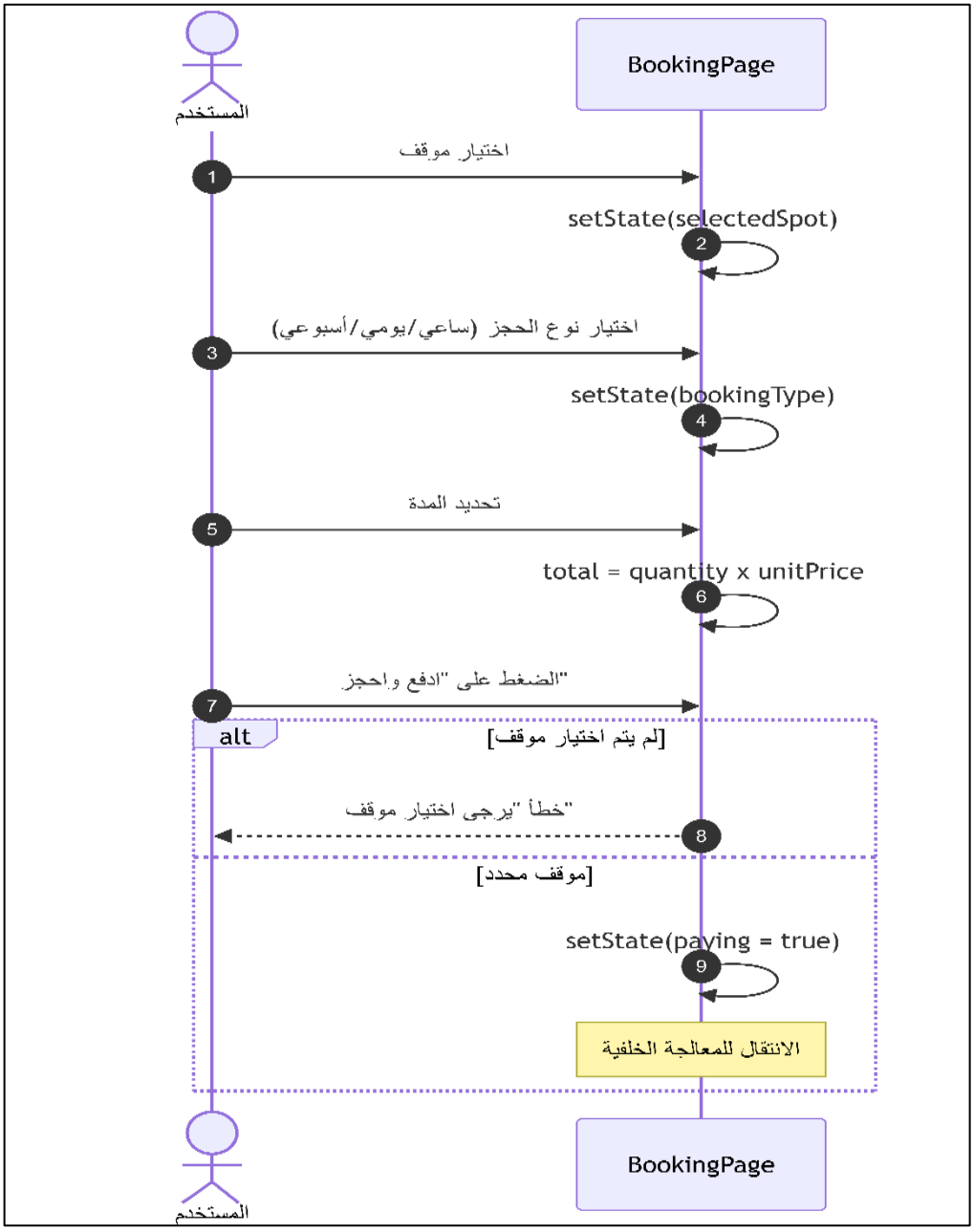
الشكل 8.3:المخطط التسلسلي لعملية تسجيل حساب في تطبيق بالنسبة للسائقين.

عملية البحث عن اقرب موقف بإستخدام خوارزمية Dijkstra:



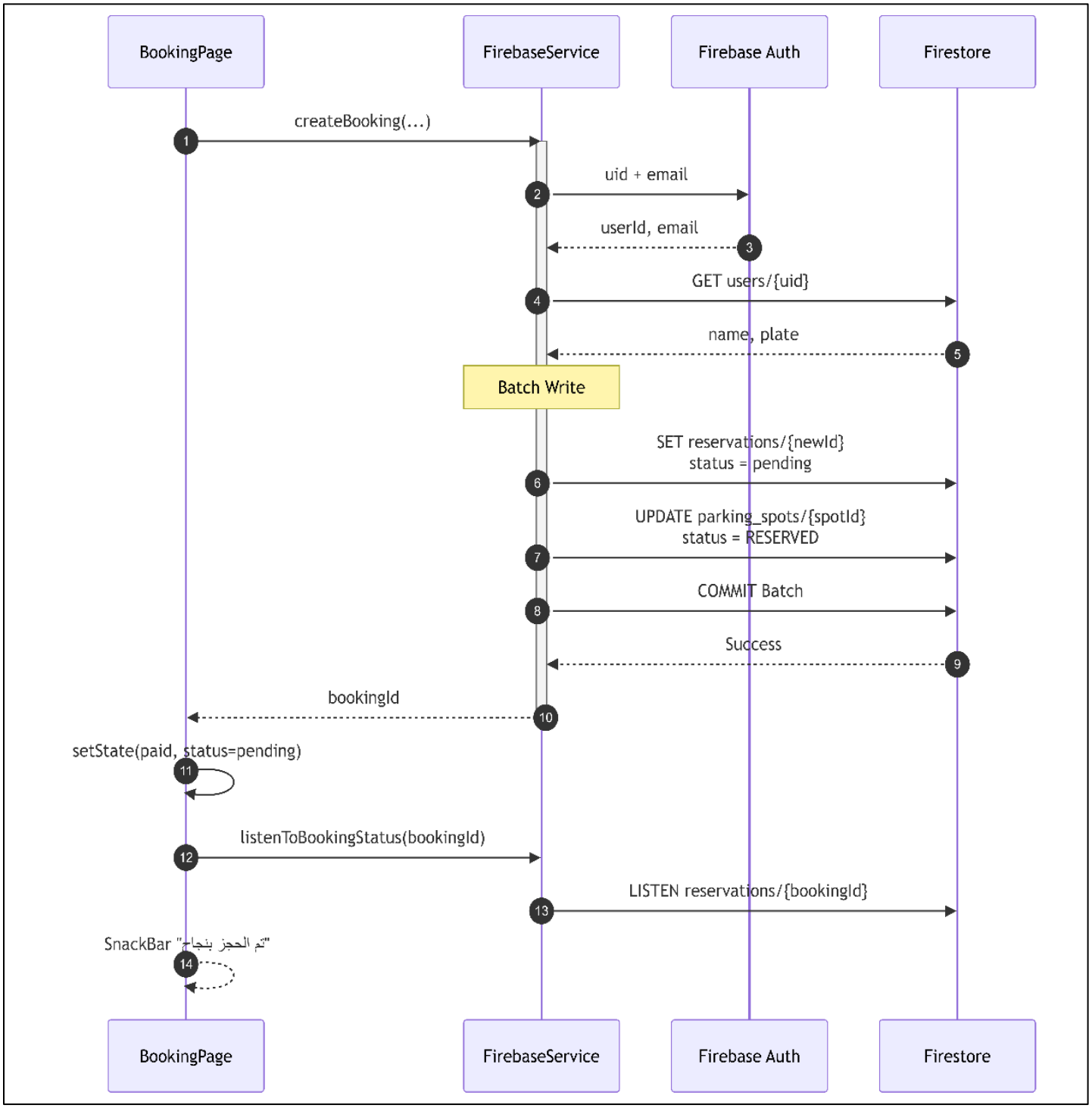
الشكل 9.3:المخطط التسلسلي لعملية البحث عن اقرب موقف بإستخدام خوارزمية Dijkstra

عملية الحجز بالتسخدام التطبيق:



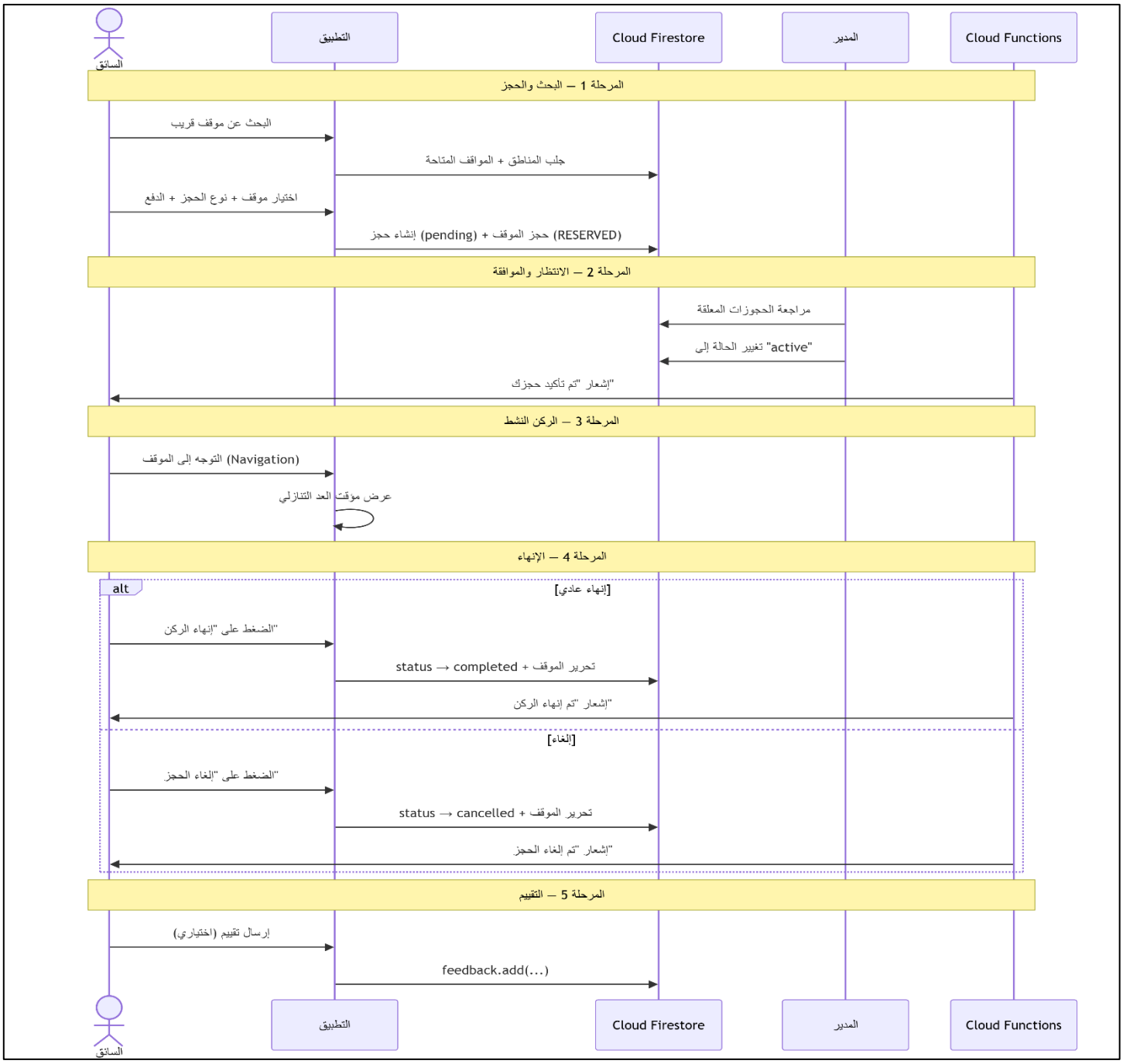
الشكل 10.3: المخطط التسلسلي لعملية الحجز بالتطبيق

عملية المعالجة الخلفية:



الشكل 11.3:المخطط التسلسلي لعملية المعالجة الخلفية.

حياة التطبيق:



الشكل 12.3: المخطط التسلسلي لحياة التطبيق.

3.6-مخطط الفئات(Class diagram):

يعد مخطط الفئات أحد أهم مخططات لغة النمذجة الموحدة(UML) ، حيث يُستخدم لتمثيل البنية الثابتة للنظام من خلال توضيح الفئات (Classes) وخصائصها (Attributes) والعمليات (Methods) بالإضافة إلى العلاقات فيما بينها مثل الوراثة والارتباط والتجميع. ويساعد هذا المخطط في تقديم تصور شامل للبنية الداخلية للنظام وكيفية تفاعل مكوناته المختلفة**[46]**.

يوضح مخطط الفئات الخاص بنظام إدارة المواقف الذكية مجموعة من الفئات الأساسية التي تمثل الكيانات الرئيسية للنظام، ومن أبرزها:

1- فئة المستخدم (User) :

تمثل المستخدم الأساسي للنظام، وتحتوي على بيانات تسجيل الدخول والمعلومات الشخصية مثل الاسم والبريد الإلكتروني ورقم الهاتف وكلمة المرور .

2- فئة السائق (Driver) :

ترث من فئة المستخدم وتمثل مستخدم التطبيق النهائي، حيث تُمكنه من البحث عن المواقف وإجراء الحجوزات وإدارة المركبات والمدفوعات .

3- فئة المسؤول(System Admin) :

تمثل الجهة الإدارية العليا للنظام، وتمتلك صلاحيات إدارة مناطق الوقوف والمستخدمين والتقارير والإحصائيات .

4- فئة مدير المنطقة (Parking Manager) :

تمثل المسؤول التشغيلي عن مناطق الوقوف، حيث يمكنه إدارة المناطق والمواقف وتعديل الأسعار ومتابعة الحجوزات .

5- فئة منطقة الوقوف(Parking Zone) :

تمثل مناطق المواقف داخل النظام، وتتضمن معلومات مثل اسم المنطقة والموقع وعدد المواقف المتاحة .

6- فئة موقف السيارة (Parking Spot) :

تمثل الموقف الفردي داخل المنطقة، وتحتوي على حالة الموقف ونوعه وإمكانية حجزه .

7- **فئة الحجز(Reservation):**

تمثل عمليات الحجز التي يقوم بها السائق، وتتضمن معلومات وقت البداية والنهاية وحالة الحجز والتكلفة الإجمالية .

8- فئة الدفع(Payment) :

تمثل عمليات الدفع المرتبطة بالحجوزات، وتشمل مبلغ الدفع وطريقة الدفع وحالة العملية .

9- **فئة الإشعارات (Notification) :**

تستخدم لإرسال التنبيهات والإشعارات للمستخدمين مثل تأكيد الحجز أو نجاح الدفع أو التذكيرات .

10- فئة التقارير (Report):

تمثل التقارير والإحصائيات التي يتم إنشاؤها من قبل المسؤول أو المدير لتحليل البيانات وحركة المواقف .

11- فئة التغذية الراجعة(Feedback) :

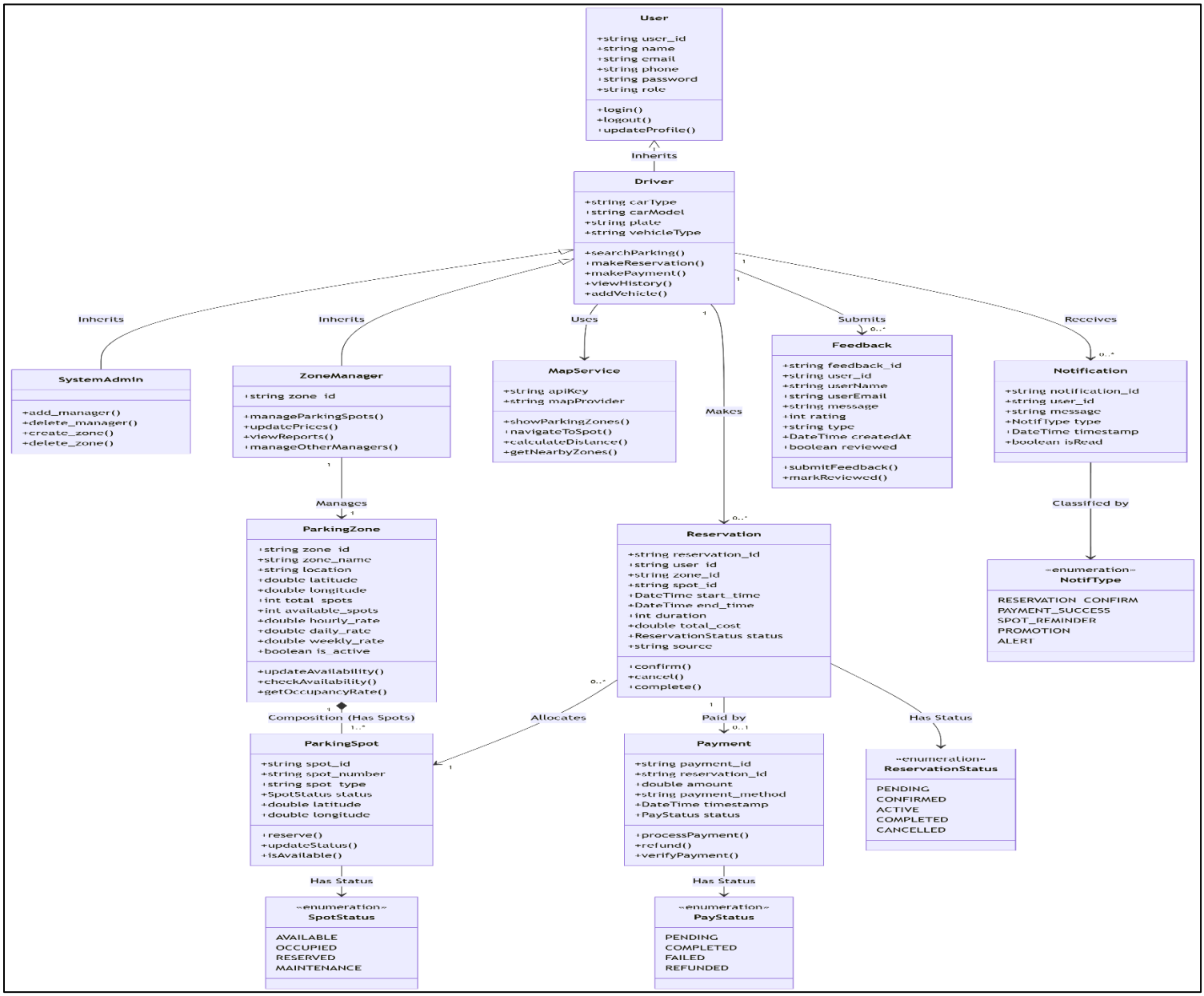
تُستخدم لتخزين تقييمات المستخدمين والشكاوى والاقتراحات المتعلقة بالنظام .

12- فئة التسعير(Pricing) :

تمثل سياسات التسعير الخاصة بالمواقف مثل السعر بالساعة أو اليوم أو الأسبوع

13- فئة خدمة الخرائط(MapService):

هي فئة مسؤولة عن التعامل مع الواجهات الجغرافية لحساب المسافات وتحديد أقرب مواقف السيارات المتاحة للمستخدم



الشكل 13.3: المخطط الفئات لنظام

الخاتمة:

يأتي هذا الفصل الثالث لتقديم تصميم متكامل ومنهجي لنظام**SmartParking Real-Time Parking Management System** ، فقد تم فيه تحديد المتطلبات الوظيفية وغير الوظيفية، وتصميم البنية المعمارية متعددة الطبقات، ونمذجة المكونات الرئيسية للنظام مع مراعاة الخصوصيات الجزائرية من حيث التكلفة و سهولة التنفيذ.

يعد التصميم المقترح أساسا متينا يربط بين النتائج النظرية والميدانية السابقة وبين التنفيذ العملي، حيث يوفر حلولا ذكية لعرض التوافر في الوقت الفعلي، الحجز، التوجيه، والتسعير الديناميكي، مع ضمان تجربة مستخدم سلسة وآمنة.

الفصل الرابع: التنفيذ والاختبار:

1-المقدمة:

بعد أن تناول الفصل الأول واقع تسيير مواقف السيارات في الجزائر وحدد الثغرات الرئيسية، وبعد أن قدم الفصل الثاني الإطار المنهجي والنتائج الميدانية التي أكدت الحاجة الملحة لحل رقمي متكامل، وبعد ان قدم الفصل الثالث المتطلبات الوظيفية وغير الوظيفية وتصميم البنية المعمارية والنماذج (UML) وقاعدة البيانات يأتي هذا الفصل ليترجم التصميم النظري إلى واقع عملي ملموس.

يهدف الفصل الرابع إلى عرض مراحل التنفيذ الفعلي للنظام، بدءاً من اختيار بيئة التطوير والتقنيات المستخدمة، مروراً بتفصيل تنفيذ المكونات الرئيسية (تطبيق الموبايل، الموقع الإداري، الباك إند، قاعدة البيانات، والتكامل بينها)، وانتهاءً بمرحلة الاختبار الشاملة التي تثبت مدى تطابق النظام مع المتطلبات المحددة سابقا.

وسيتناول هذا الفصل النقاط التالية:

1- بيئة التطوير واللغات البرمجية.
2- أدوات البرمجية.
3- بنية المشروع .
4- خطوات التنفيذ.
5- المظهر النهائي للمنصة.
6- حدود النظام.

2-بيئة التطوير و اللغات البرمجية:

عندما نتحدث عن بئية التطوير واللغات البرمجية فإنه يوجد لدينا جانبين هما:

1.2- جانب تطبيق الهاتف (Mobile Application):

وهو التطبيق المخصص للسائقين لحجز وتصفح المواقف، وقد تم بناؤه باستخدام إطار العمل Flutter المبني على لغة **Dart [49]**، وهو إطار عمل تم تطويره من طرف شركة **Google [52,49]** من أجل بناء واجهات المستخدم (**frontend**) بسهولة وكفاءة عالية على منصتي **Android** و **iOS.**

2.2- جانب موقع الإداري (Web Dashboard):

وهو الجزء المخصص للمسؤولين والمدراء، حيث يمكن للمسؤولين إدارة المناطق والإحصائيات العامة للنظام، بينما يستطيع المدراء إدارة منطقتهم الخاصة ومراقبة الحجوزات والمواقف في الوقت الفعلي.

وقد تم بناء هذا الموقع باستخدام **React (v19.2)**، وهي مكتبة **JavaScript** مفتوحة المصدر تم تطويرها من قبل **Facebook (Meta)** لبناء واجهات مستخدم تفاعلية وقابلة لإعادة الاستخدام باستخدام معمارية المكونات**(Component-Based Architecture) [53]** . كما تم استخدام **Vite (v7.3)**، وهي أداة بناء حديثة وسريعة جداً **(Build Tool)** توفر تجربة تطوير سريعة مع **Hot Module Replacement** وتحسين أداء البناء النهائي**[54]** .

واعتمد الموقع على لغة **JavaScript (ES6+)**، وهي النسخة الحديثة من لغة البرمجة **JavaScript** التي تدعم مميزات متقدمة مثل **Arrow Functions**، **Promises**، **Async/Await**، **Classes**،**Modules [55]**، وأما هيكلة الصفحات ومحتواها فقد تم بناؤها باستخدام **HTML5**، الإصدار الخامس من لغة ترميز صفحات الويب الذي يدعم العناصر الدلالية والوسائط المتعددة**[56]**، وتم تنسيق الواجهات وتصميمها باستخدام **CSS3**، الإصدار الثالث من لغة تنسيق صفحات الويب الذي يتيح التصميم المتجاوب (Responsive Design) والتأثيرات البصرية المتقدمة**[57]**.

3.2- البنية التحتية والخلفية (Backend & Infrastructure):

اعتمد النظام على بنية تحتية سحابية بدون خوادم **(Serverless Architecture)** تعرف ب**(Backend-as-a-Service (BaaS** ، وذلك لضمان المرونة والتوسع والأمان العالي مع تقليل تكاليف الصيانة والإدارة.

تتمثل هذه البنية في:

- **منصة Firebase من Google:** المنصة السحابية الرئيسية التي تربط جميع أجزاء النظام (تطبيق الموبايل + لوحة التحكم).

- **Cloud Firestore:** قاعدة بيانات غير علائقية (**NoSQL Document Database**) تدعم التزامن وتحديث البيانات في الوقت الفعلي **(Real-time) [45]**، مما يسمح بمزامنة حالة المواقف والحجوزات فورياً بين جميع المستخدمين.

- **Firebase Cloud Functions**: لتنفيذ المنطق الخلفي **(Serverless Logic)** في بيئة خادمية آمنة. تم استخدامها لعمليات حساسة مثل إعادة تعيين كلمات المرور، إرسال رموز التحقق **(OTP)**، ووظائف تلقائية **(مثل إنهاء الحجز تلقائياً)**، وذلك باستخدام لغة **JavaScript** ضمن بيئة **Node.js [50].**

- **Firebase Cloud Messaging (FCM):** لإرسال الإشعارات الفورية **(Push Notifications)** للهواتف عند تحديث حالة الحجز أو الموافقة عليه أو حدوث أي تغيير مهم **[51].**

- **Firebase Authentication:** لإدارة تسجيل الدخول والمصادقة بأمان لجميع أنواع المستخدمين (سائقين، مدراء، مسؤولين).

3-أدوات البرمجية:

تم تنفيذ كامل المشروع باستخدام بيئة تطوير رئيسية واحدة مع أدوات مساعدة، وهي كالتالي:

1- **IntelliJ IDEA** (Ultimate Edition):

هو بيئة تطوير متكاملة (**Integrated Development Environment**) قوية ومتعددة اللغات تم تطويرها من قبل شركة **JetBrains**.يُعتبر من أفضل بيئات التطوير المهنية، حيث يدعم بشكل ممتاز لغات **Dart**و **Flutter**و **JavaScript**و **React**. تم استخدامه كبيئة التطوير الرئيسية الوحيدة لتطوير تطبيق الهاتف ولوحة التحكم الإدارية، ويتميز بأدوات ال**Intelligent Code Completion** ، **Refactoring**، **Debugging**، و **Git Integration** المتقدمة**[58]** .

2- **Android Studio**:

هو بيئة التطوير الرسمية (**Official IDE**) من شركة **Google**لتطوير تطبيقات **Android** يستند في تشغيله على **IntelliJ IDEA** او**Visual studio code** تم استخدامه كأداة مساعدة رئيسية لتطوير واختبار تطبيق **Flutter** على نظام **Android** بشكل مباشر وفعال **[59].**

3- **Firebase Console**:

هي واجهة التحكم السحابية الرسمية لمنصة **Firebase**من **Google**. تستخدم لإعداد وإدارة قاعدة البيانات(**Firestore**) ، المصادقة(**Authentication**) ، الوظائف السحابية(**Cloud Functions**) ، والإشعارات(**FCM**) ، بالإضافة إلى مراقبة أداء النظام وتحليل البيانات في الوقت الفعلي**[60]**.

Git & GitHub:

هو نظام التحكم في الإصدارات (**Version Control System**) الأكثر استخداماً في العالم، بينما **GitHub**هو منصة استضافة الكود السحابية. تم استخدامهما لإدارة الشيفرة المصدرية، تتبع التغييرات، والعمل الجماعي **[62,61].**

4- **أدوات الاختبار والتصحيح :**

- Microsoft Edgeو **Chrome:**

DevToolsتم استخدامهما لاختبار وتصحيح لوحة التحكم الإدارية **(Web Dashboard)** المبنية بـ **React**. حيث ساهم **Microsoft Edge**مع **Chrome DevTools** في فحص وتحسين أداء التطبيق، اكتشاف الأخطاء، تحليل استهلاك الموارد، واختبار تجربة المستخدم **(UX)** عبر المتصفحات المختلفة، مع التركيز على السرعة والاستجابة والتوافق(**Compatibility**) .

5-مواصفات الاجهزة المستخدمة في التطوير:

مواصفات الجهاز	اسم الجهاز
Intel Core i5-6300U	HP ProBook G2
16 GB RAM	
237GB SSD nvme	
octa-core Samsung Exynos 850 processor	SAMSUNG M13
6GB RAM	
128 GB	

جدول1.4:جدول يوضح مواصفات اجهزة التطوير

4-بنية المشروع(System Architecture) :

في مشروع **SmartParking Real-Time Parking Management System** تم استعمال بنيتين معماريتين مختلفتين هما:

1- Role-Sliced Dashboard Architecture في صفحة الويب (Web Dashboard):

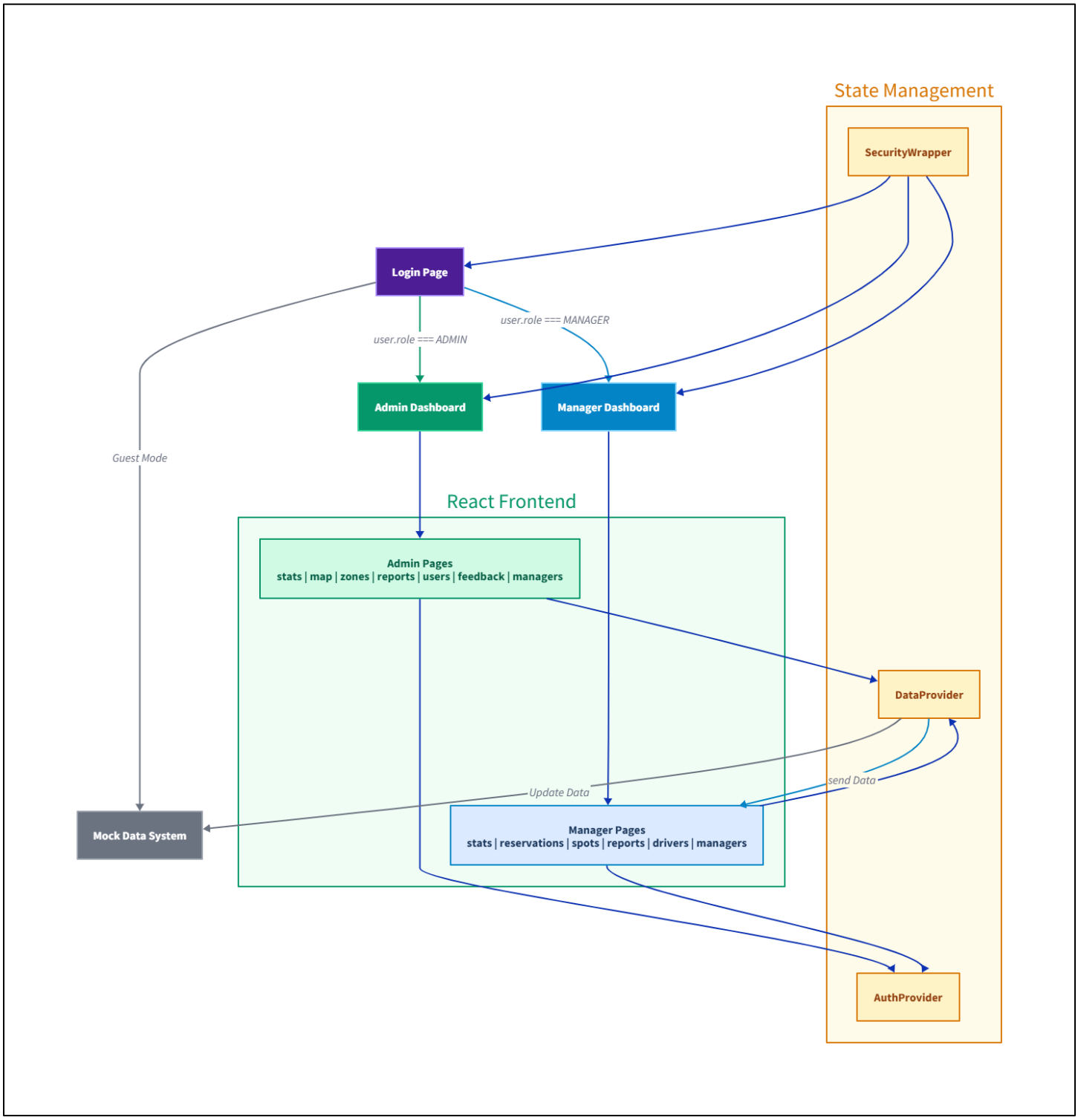
هي بنية تصميم واجهة الويب تعتمد على تقسيم كامل للواجهة حسب دور المستخدم (Role)، بحيث يحصل كل مستخدم على **Dashboard** منفصل تمامًا مخصص له**[63].**

تم تطبيق هذه البنية في صفحة الويب لفصل تجربة المستخدمين حسب صلاحياتهم. بدلاً من **Dashboard** واحد كبير، تم تقطيع

الـ **Dashboard** إلى واجهات مستقلة:

1- Admin Dashboard: يحتوي على كل الصلاحيات مثل الإحصائيات العامة، إدارة المناطق، إدارة المديرين والمستخدمين، الخريطة التفاعلية الكاملة، والتقارير الشاملة.

2-Manager Dashboard: مخصص لمدير المنطقة فقط، ويعرض إحصائيات منطقته، الحجوزات، الأماكن، والتقارير الخاصة به.



الشكل1.4:مخطط يوضح كيف يتم تطبيق Role-Sliced Dashboard Architecture في صفحة الويب (Web Dashboard)

المميزات الرئيسية:

- أمان مرتفع جدًا

- واجهة نظيفة ومرتبة لكل مستخدم

- سهولة الصيانة والتطوير

- أداء أفضل بسبب تحميل البيانات حسب الحاجة فقط

2-Layered Architecture في تطبيق الهاتف المحمول (Flutter):

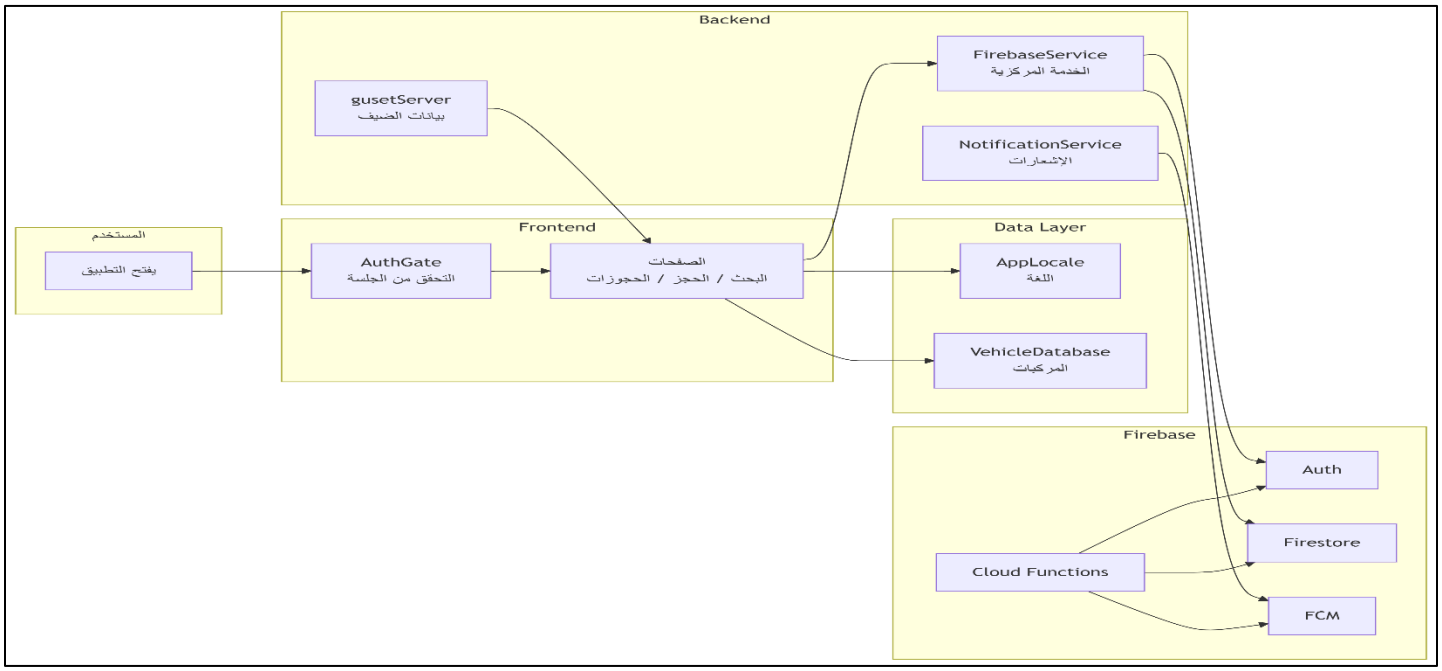
هي بنية لتنظيم التطبيق على شكل طبقات منفصلة، كل طبقة لها مسؤولية واضحة ومستقلة**[65]**.

تم تقسيم تطبيق الهاتف إلى ثلاثة طبقات رئيسية:

1- طبقة البيانات (**Data Layer**).

2- طبقة الخدمات(**Backend**)

3- طبقة العرض والواجهات (**Frontend**).



الشكل2.4:مخطط يوضح كيف يتم تطبيق Layered Architecture في تطبيق الهاتف المحمول (Flutter).

المميزات الرئيسية:

- تنظيم ممتاز وسهل الفهم .

- فصل واضح بين المنطق والواجهة .

- سهولة الصيانة والتوسع.

- أداء خفيف مناسب لتطبيق الموبايل.

5- مراحل التنفيذ (Step Implementation):

1.5-تخطيط المهام وفق المنهجية الرشيقة(Scrum/Agile Methodology):

تم اعتماد المنهجية الرشيقة **Agile Methodology(Scrum)**[64] في تطوير تطبيق **SmartParking Real-Time Parking Management System** بهدف إدارة متطلبات المشروع والمهام بشكل مرن وفعال. في البداية، تم تحديد متطلبات النظام الأساسية مثل تسجيل الدخول، عرض المواقف على الخريطة، نظام الحجز، والإشعارات، ثم ترتيبها حسب الأولوية داخل قائمة Product Backlog.

بعد ذلك، تم تقسيم عملية التطوير إلى دورات قصيرة(**Sprints**) ، حيث تم التركيز في كل **Sprint** على تنفيذ مجموعة محددة من الوظائف لضمان تطوير تدريجي ومنظم للتطبيق. ساعد هذا الأسلوب على تحسين التحكم في تقدم المشروع، والبدء بالوظائف الأساسية أولاً مثل المصادقة وعرض الخرائط، ثم الانتقال إلى الميزات الإضافية مثل الإشعارات والتقارير.

كما ساهمت هذه المنهجية في تسهيل التعديل على المتطلبات أثناء التطوير، وتحسين جودة التطبيق النهائي من خلال الاختبار والتحسين المستمر بعد كل مرحلة تطوير

2.5-تحديد المهام وتتبعها:

تم استخدام جدول Excel لتسجيل جميع المهام المنجزة ضمن كل Sprint وتتبع حالتها، مما أتاح مراقبة تقدم المشروع بشكل دوري ومستمر. وفيما يلي تفصيل مراحل التطوير:

- **Sprint 1 (05/02/2026 – 18/02/2026):**

شمل تحليل المتطلبات الوظيفية وغير الوظيفية، وتصميم المخططات الأولية للنظام (Use Case, Class Diagram, Sequence Diagram)، وتصميم هيكل قاعدة البيانات السحابية وتحديد مجموعاتFirestore ، إضافةً إلى إعداد بيئة التطوير لكلٍّ من منصة الويب (React + Vite)وتطبيق الهاتف(Flutter) ، وإنشاء مشروع Firebase وربطه بكلا التطبيقين.

- **Sprint 2 (19/02/2026 – 04/03/2026):**

تضمّن تصميم وتطوير نظام المصادقة والتحقق من الهوية باستخدام Firebase Auth في كلٍّ من منصة الويب وتطبيق الهاتف، وبناء نظام التحقق الثنائي عبر البريد الإلكتروني(OTP) ، وكتابة قواعد أمان Firestore لحماية البيانات السحابية.

- **Sprint 3 (05/03/2026 – 20/03/2026):**

شمل تطوير لوحة التحكم الرئيسية (Dashboard) مع الإحصائيات اللحظية، وبناء واجهات إدارة مواقف السيارات (Parkings CRUD) وإدارة المستخدمين والسائقين، وتطوير خدمة الاتصال بـ Firestore وطبقة إدارة الحالة باستخدام React Context ، فضلاً عن تهيئة نظام Git للتحكم في الإصدارات.

- **Sprint 4 (21/03/2026 – 05/04/2026) :**

ركّز على تطوير واجهات تطبيق الهاتف الرئيسية بما فيها عرض المواقف المتاحة، وتفاصيل الموقف ونظام الحجز اللحظي، وشاشة الملف الشخصي للسائق، إضافةً إلى دمج خريطة OpenStreetMap التفاعلية باستخدام flutter_map وربط خدمة تحديد الموقع الجغرافي.

- **Sprint 5 (06/04/2026 – 22/04/2026) :**

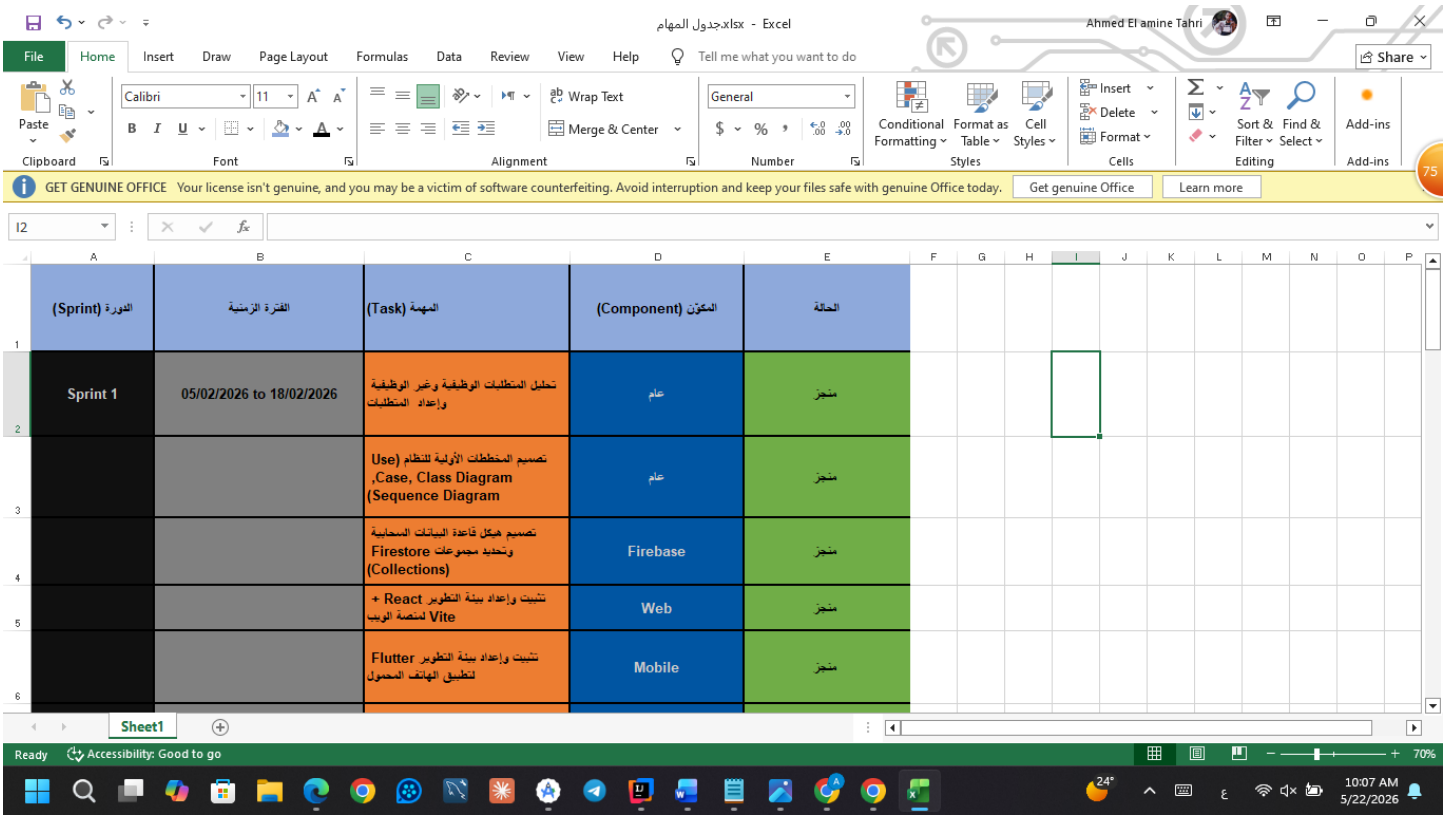
تضمّن تطوير نظام الملاحة وجلب مسارات الطرق من خوادم OSRM ، ورسم مسار التنقل البصري (Polyline) على الخريطة، ودمج التوجيه الصوتي باستخدام flutter_tts ، وبناء نظام الإشعارات السحابية الفورية عبر Firebase Cloud Messaging (FCM) ، وتخزين تفضيلات المستخدم محلياً.

- **Sprint 6 (23/04/2026 – 04/05/2026) :**

شمل تطوير نظام إدارة الحجوزات في منصة الويب، وبناء واجهة الإحصائيات والتقارير المتقدمة، ونشر الدوال السحابية (Cloud Functions)، إضافةً إلى دعم تعدد اللغات (العربية والفرنسية) ودمج وظائف الروابط الخارجية والاتصال الهاتفي.

- **Sprint 7 (05/05/2026 – 10/05/2026):**

خُصِّص للمرحلة النهائية التي شملت اختبارات التكامل الشاملة بين منصة الويب والتطبيق وقاعدة البيانات، واختبارات الأداء وتجربة المستخدم (UI/UX) ، وإصلاح الأخطاء البرمجية، والمراجعة النهائية للشيفرة المصدرية، وإعداد الوثائق التقنية النهائية.



الشكل3.4:استخدام جدول EXCELلتسجيل سيرورة العمل ومدى الانجاز في كل جزء

وقد أُنجزت جميع المهام المخطط لها عبر الدورات السبع بنجاح، مما يعكس فاعلية المنهجية الرشيقة في ضبط سير التطوير وضمان تسليم المشروع في الوقت المحدد

3.5- إعداد بيئة تطوير لوحة التحكم باستخدام React و(React & Vite Development Environment Setup) Vite:

تم إعداد بيئة تطوير لوحة التحكم الإدارية الخاصة بمنصة SmartParking بالاعتماد على مكتبة **ReactJS** وأداة البناء الحديثة **Vite**، وذلك بهدف توفير بيئة تطوير سريعة، مرنة، وقادرة على دعم التحديث الفوري للواجهات الرسومية أثناء عملية البرمجة. وقد تم تنفيذ عملية التهيئة وفق المراحل التالية:

1- **تثبيت بيئة التشغيلNode.js :**

تم تثبيت الإصدار المستقر طويل الدعم (LTS) من بيئة **Node.js**، والتي توفر مدير الحزم **npm**المستخدم في إدارة وتنزيل الاعتماديات البرمجية اللازمة للمشروع.

2- **إنشاء المشروع باستخدامVite :**

تم استخدام أداة **Vite[67,66]**،لإنشاء مشروع React حديث يتميز بسرعة البناء والتشغيل مقارنةً بالأدوات التقليدية مثل

Create-React-App

وقد تم تنفيذ الأوامر التالية:

```
npm create vite@latest smartparking-dashboard -- --template react[65]
cd smartparking-dashboard
npm install[69]
```

3- **تشغيل خادم التطوير المحلي:**

بعد الانتهاء من التهيئة، تم تشغيل خادم التطوير المحلي الخاص بـ Vite عبر الأمر التالي:

```
npm run dev[70]
```

يوفر Vite تقنية **Hot Module Replacement (HMR)** التي تسمح بتحديث الواجهة مباشرة بعد تعديل الملفات دون إعادة تحميل الصفحة بالكامل، مما يساهم في تسريع عملية التطوير وتحسين تجربة البرمجة.

4.5-إعداد بيئة تطوير تطبيق الهاتف المحمول باستخدام(Flutter Development Environment Setup) Flutter :

تم تطوير تطبيق الهاتف المحمول الخاص بمنصة SmartParking باستخدام إطار العمل **Flutter** ولغة البرمجة **Dart**، لما يوفره من إمكانية بناء تطبيقات هجينة تعمل على أنظمة Android و iOS من خلال قاعدة كود موحدة. وقد تمت تهيئة بيئة التطوير كما يلي:

1- **تثبيت:Flutter SDK**

تم تنزيل النسخة المستقرة من حزمة تطوير Flutter من الموقع الرسمي، ثم إضافة المسار الخاص بالمجلد bin إلى متغيرات البيئة (Environment Variables) لتمكين استخدام أوامر Flutter من الطرفية.

2- **إعداد Android Studio و:Android SDK**

تم تثبيت بيئة التطوير **Android Studio** مع تنزيل أدوات Android SDK وحزم البناء اللازمة لتجميع التطبيق وتشغيله على أجهزة أندرويد.

كما تم تنفيذ الأمر التالي للموافقة على تراخيص التطوير:

```
flutter doctor --android-licenses
```

3. **فحص سلامة بيئة التطوير:**

تم استخدام الأداة التشخيصية flutter doctor للتحقق من تكامل بيئة التطوير والتأكد من توفر جميع المتطلبات البرمجية اللازمة.

```
flutter doctor
```

4. **ضبط إعدادات البناء:**

تم تعديل إعدادات ملف build.gradleلتحديد نسخ الـSDK المستخدمة:

- compileSdkVersion = 34
- targetSdkVersion = 34
- minSdkVersion = 23

وذلك لضمان توافق التطبيق مع أحدث إصدارات نظام Android ودعم مكتبات الخرائط والملاحة.

5. **إعداد بيئات الاختبار:**

تم الاعتماد على الأجهزة الحقيقية عبر تفعيل وضع المطور وميزة USB Debugging.

ولتثبيت التطبيق وتشغيله على الجهاز تم تنفيذ:

```
flutter run
```

5.5- إدارة المشروع وتوزيع المهام(Project Management and Task Organization) :

تم اعتماد منهجية التطوير الرشيقة **Agile**وبالتحديد إطار العمل **Scrum**لإدارة عملية تطوير المنصة، حيث جرى تقسيم العمل إلى دورات تطويرية قصيرة (Sprints) تسمح بمتابعة التقدم وإدخال التعديلات بشكل مستمر.

كما تم استخدام أدوات رقمية لتنظيم المهام وتتبع مراحل الإنجاز، الأمر الذي ساهم في:

1. مراقبة تقدم تنفيذ المهام البرمجية.
2. الكشف المبكر عن المشكلات التقنية ومعالجتها.
3. توزيع العمل بين مكونات النظام المختلفة، بما في ذلك لوحة التحكم وتطبيق الهاتف المحمول.

وقد ساعد هذا الأسلوب في تحسين التنسيق بين مكونات النظام وضمان تنفيذ المتطلبات الوظيفية وغير الوظيفية بكفاءة.

6.5-إعداد قاعدة البيانات وتصميم الهيكلية(Database Setup and Schema Design) :

تم الاعتماد على منصة Google Firebase كخدمة خلفية سحابية (**Backend-as-a-Service - BaaS**) لتوفير خدمات المصادقة، التخزين، وقاعدة البيانات اللحظية.

تم استخدام أداة Firebase CLI لتهيئة المشروع وربط الخدمات السحابية عبر الأوامر التالية:

```
npm install -g firebase-tools
firebase login
firebase init[71,72]
```

وقد تم اختيار **Cloud Firestore**كقاعدة بيانات رئيسية نظراً لاعتمادها على نموذج NoSQL القائم على المستندات (Documents) والمجموعات(Collections) ، بالإضافة إلى دعمها للمزامنة اللحظية للبيانات.

تضمنت قاعدة البيانات المجموعات الرئيسية التالية:

اسم المجموعة	الوظيفة
users	**هذه المجموعة يتم تخزين فيها معلومات المستخدمين(مدراء-مسؤولين-سائقين) بما في ذلك اسم المستخدم وكلمة المروروعنوان البريد الإلكتروني.**
zones	**هذه المجموعة يتم تخرين فيها اسم المنطقة ومعلوماتها مثل احداثياتها وكم تحتوي على مو اقف**
parking_spots	**هذه المجموعة يتم تخزين معلومات مو اقف السيارات (Parking Spots) مثل الموقع، النوع، الحالة، والإحداثيات الجغر افية.**
reservations	**هذه المجموعة يتم تخزين فيها معلومات الحجوزات على مو اقف السيارات، بما في ذلك وقت البداية**
payments	**هذه المجموعة يتم تخزين فيها عمليات الدفع المتعلقة بالحجوزات .**
vehicles	**هذه المجموعة يتم تخزين فيها معلومات المركبات الخاصة بالمستخدمين**
feedback	**هذه المجموعة فيها تقييمات وآراء المستخدمين بعد استخدام الخدمة (التقييم، التعليق).**

جدول2.4 8:انواع المجموعات في قاعدة البيانات

وقد تم تصميم الهيكلية بطريقة تدعم المرونة وسهولة التوسع مع الحفاظ على سرعة الاستجابة أثناء عمليات القراءة والكتابة.

7.5-تطوير لوحة التحكم باستخدام(ReactJS (Web Dashboard Development :

تم تطوير لوحة التحكم الإدارية باستخدام مكتبة ReactJS ضمن بنية تطبيقات الصفحة الواحدة(Single Page Application - SPA) ، بهدف توفير واجهة مستخدم تفاعلية وسريعة الاستجابة. ويتكون هذا القسم من عدة عناصر رئيسية:

1- تنظيم الملفات والمكونات:

تم استخدام بنية Role-Sliced Dashboard Architecture وبالاضافة الى ذلك تم تقسيم المشروع إلى عدة مجلدات رئيسية:

1- api: يحتوي على جميع الدوال والملفات الخاصة بالاتصال مع الخادم.
2- components: للمكونات القابلة لإعادة الاستخدام.
3- layouts: لتخطيطات الواجهات.
4- context: لإدارة الحالة العامة والبيانات.
5- pages_admin_dashboard : يحتوي على جميع صفحات لوحة تحكم المدير العام(Admin).
6- pages_manager_dashboard: يحتوي على جميع صفحات لوحة تحكم المدير(Manager).
7- mock: يحتوي على بيانات وهمية(Mock Data)
8- login_page: مجلد خاص بصفحة تسجيل الدخول فقط.

2 - المكتبات البرمجية:

لدعم الوظائف المختلفة للوحة التحكم، تم تثبيت مجموعة من المكتبات البرمجية المتخصصة، من بينها:

1- react-router-domلإدارة التنقل بين الصفحات[74,73] .
2- leafletو react-leafletلعرض الخرائط الجغرافية[78,77,76,75] .
3- chart.jsو rechartsلإنشاء الرسوم البيانية[84,83,82,81,80,79] .
4- jspdfو xlsxلتوليد ملفات التقارير[88,87,86,85] .
5- @emailjs/browserلإرسال رسائل البريد الإلكتروني[90,89] .

```
npm install react-router-dom
npm install leaflet react-leaflet
npm install chart.js react-chartjs-2 recharts
npm install jspdf jspdf-autotable xlsx
npm install @emailjs/browser
```

3-وضع الضيف(Guest Mode)

تم تطوير وضع تجريبي يسمح للمستخدم باستكشاف النظام دون الحاجة للاتصال المباشر بقاعدة البيانات، وذلك بالاعتماد على بيانات محاكاة محلية.

كما تم تضمين آلية حماية (guestGuard) تمنع تنفيذ أي عمليات تعديل فعلية أثناء استخدام هذا الوضع.

4-نظام الإنهاء التلقائي للحجوزات

تم تطوير آلية تعمل بشكل دوري للتحقق من الحجوزات التي تجاوزت وقتها المحدد، ثم تحديث حالتها تلقائيا وتحرير الموقف المرتبط بها لضمان دقة بيانات الإشغال.

8.5-تطوير تطبيق الهاتف المحمول باستخدام(Flutter Mobile Application Development) Flutter

تم تطوير تطبيق SmartParking باستخدام Flutter بهدف توفير تجربة استخدام موحدة على أنظمة Android و.iOS ويتكون هذا القسم من عدة عناصر رئيسية:

1- هيكلية المشروع

تم تقسيم المشروع إلى عدة طبقات:

- backendللتعامل مع الخدمات السحابية.
- data_layerلإدارة البيانات المحلية والترجمة.
- frontendلواجهات الاستخدام والشاشات.

2- المزايا البرمجية الرئيسية

- **خوارزمية :Dijkstra** لترتيب المواقف الأقرب للمستخدم.
- **الملاحة الجغرافية :**عبر خرائط OpenStreetMap وخدمة.OSRM
- **الإرشاد الصوتي :**باستخدام تقنيات.Text-to-Speech
- **الإشعارات الفورية :**عبر.Firebase Cloud Messaging

3- المكتبات البرمجية:

1- مكتبة تهيئة firebaseداخل المشروع.

2- مكتبة ادارة وتوثيق الحسابات (Authentication).

3- مكتبة الخاصة بجلب البيانات من قاعدة البيانات.

4- مكتبة وظائف الـ backend .

5- مكتبة نظام إشعارات الدفع السحابية.

6- مكتبة نظام الإشعارات المحلية التفاعلية.

7- مكتبة تحديد الموقع الجغرافي .

8- مكتبة نظام الخرائط التفاعلية OpenStreetMap.

9- مكتبة الحسابات الرياضية الجغرافية للخرائط.

10- مكتبة بروتوكول طلبات الويب HTTP.

11- مكتبة التوجيه الصوتي التفاعلي TTS.

12- مكتبة ارسال الرسائل الالكتيرونيةSMTP Mailer .

13- مكتبة التخزين المحلي الدائم البسيط على الهاتف.

14- مكتبة متغيرات البيئة.

15- مكتبة الايقوناتFontAwesome .

16- مكتبة الايقونات لios.

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add firebase core [91]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add firebase_auth [92]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add cloud_firestore [93]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add cloud_functions [94]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add firebase_messaging [95]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add flutter_local_notifications [96]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add geolocator [97]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add flutter_map [98]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add latlong2 [99]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add http [100]

```
C:\devlopment\flutter-apps\smart_parking_project> flutter pub add flutter_tts [101]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add mailer [102]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add shared_preferences [103]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add flutter_dotenv [104]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add font_awesome_flutter [105]

C:\devlopment\flutter-apps\smart_parking_project> flutter pub add cupertino_icons [106]
```

9.5-إعداد الواجهة الخلفية وطبقة الخدمات(Backend and Service Layer) :

اعتمد النظام على هندسة Serverlessمع فصل كامل بين منطق تطبيق Flutter ولوحة التحكم ReactJS.

1.9.5- ربط Firebase بمشروع(firebase.js):

تم تأسيس الاتصال وتضمين إعدادات البنية الأساسية في ملف firebase.jsكالتالي:

```
import { initializeApp } from "firebase/app"; import { getFirestore } from "firebase/firestore";
import { getAuth } from "firebase/auth";
 const firebaseConfig = {
apiKey: "---------------------------------------------- ",
authDomain: "smartparking-realtime.firebaseapp.com",
 projectId: "smartparking-realtime",
storageBucket: "smartparking-realtime.firebasestorage.app"
, messagingSenderId: "--------------------------------------",
appId: "-------------------------------------------------"
, measurementId: "-------------------" }
; const app = initializeApp(firebaseConfig); export const db = getFirestore(app);
Firestore export const auth = getAuth(app);
```

10.5-طبقة API المركزية (firebaseApi.js):

تتكون الطبقة من وحدات متخصصة (.authAPI,zonesAPI,reservationsAP.....etc). ومن أبرز مميزاتها:

- استخدام Secondary App Instance لإجراء العمليات الإدارية (إنشاء حسابات، تغيير كلمات المرور) دون التأثير على جلسة المسؤول.

- دالة migrateZoneIds لضمان توافق البيانات التاريخية.

- اشتراكات onSnapshot للتحديثات في الوقت الفعلي.

- تكامل مع EmailJSلإرسال كلمات المرور المؤقتة.

11.5- خدمة Firebase في تطبيق Flutter (firebase_service.dart):

يمثل ملف **firebase_service.dart**العصب المنطقي والتشغيلي لتطبيق الهاتف المحمول الخاص بالسائقين، حيث تم تصميمه بأسلوب منظم وآمن لتولي المهام الأساسية للنظام، ويشمل ذلك ما يلي:

1- **النماذج و أنواع البيانات والتحويل البرمجي:(Models & Enums)**

يتضمن الملف مجموعة من الأدوات المسؤولة عن إدارة وتحويل البيانات، من بينها:

- دالة التحويل الآمن (_toDouble) التي تقوم بتحويل القيم الرقمية أو النصية الواردة من قاعدة البيانات إلى أرقام عشرية، بما يضمن استقرار التطبيق وتفادي أخطاء اختلاف أنواع البيانات.

- قائمة نوع الحجز (BookingType) التي توفر تمثيلاً منظماً لأنواع الحجوزات (ساعي، يومي، أسبوعي) مع دعم الترجمة النصية والتحكم الديناميكي في قواعد الحجز.

- بنية نوع المركبة (VehicleType) التي تضمن التوافق بين المسميات المحلية وقيمها المخزنة في قاعدة البيانات السحابية(STANDARD ، VIP، MOTORCYCLE، ELECTRIC،DISABLED).

- نموذج الموقف الفعلي (RealSpot) المرتبط بحالة الموقف (SpotStatus) والذي يسمح بإدارة توفر المواقف وإتاحة الحجز المباشر.

- نموذج سجل الحجز (BookingRecord) الذي يمثل السجل الكامل لكل عملية حجز بما يدعم تتبع الحالة وإدارة العمليات اللاحقة.

2- **محرك مصادقة OTP والتحقق عبر:SMTP**

نظراً لأهمية تقليل التكاليف التشغيلية المرتبطة بالخدمات السحابية، تم تطوير محرك تحقق مستقل داخل الفئة FirebaseService يتولى عمليات إرسال رموز التحقق واستعادة الحسابات. يعتمد هذا المحرك على بروتوكول SMTP للاتصال المباشر بخوادم Gmail عبر قناة مشفرة وآمنة، باستخدام بريد المنصة وكلمة مرور تطبيق مخصصة(App Password) ، مما يتيح تنفيذ عمليات التحقق بكفاءة مع تقليل الاعتماد على الوظائف السحابية الخارجية.

3- **دالة إرسال الرمز:(sendOtpToEmail)**

تتولى هذه الدالة توليد رمز تحقق عشوائي مكوّن من ستة أرقام، ثم تخزينه في قاعدة البيانات السحابية ضمن المجموعات المخصصة لعمليات تسجيل الدخول أو استعادة الحسابات، مع تحديد مدة صلاحية قدرها عشر دقائق. بعد ذلك يتم إنشاء رسالة بريد إلكتروني بصيغة HTML تتضمن رمز التحقق وتعليمات الاستخدام، ثم إرسالها تلقائياً إلى المستخدم.

4- **دالة استعادة الحساب الذاتي:(resetPasswordWithOtp)**

تقوم هذه الدالة بالتحقق من صحة رمز OTP المدخل ومطابقته مع البيانات المخزنة، مع مراعاة مدة الصلاحية وعدد محاولات الإدخال. وعند نجاح عملية التحقق، يتم تنفيذ عملية إعادة تعيين كلمة المرور بشكل آمن، وتحديث بيانات المستخدم في النظام، ثم حذف رمز التحقق مباشرة بعد استخدامه لضمان حماية البيانات ومنع إعادة الاستخدام.

5- **حماية تكامل العمليات والتحكم المتزامن بالحجوزات :(Firestore Transactions & Batch Writes)** لتجنب التعارض الأكثر خطورة في تطبيقات حجز المواقف، وهو محاولة سائقين اثنين حجز نفس الموقف البرمجي المهيأ في نفس الميلي ثانية من هواتف مختلفة؛ يعتمد التطبيق على آليات العمليات الموحدة (**Batch Writes**) والمعاملات السحابية المتزامنة .(**Transactions**) تقوم الدالة createBookingبإجراء عمليات متعددة غير قابلة للتجزئة برمجياً(Atomicity) ؛ حيث يتم فحص حالة توفر الموقف في Firestore، ثم حجز الموقف، وتحديث حالة وثيقة الموقف الفرعي إلى RESERVEDوتخصيصه لمعرف المستخدم السائق، وإنشاء وثيقة سجل الحجز الأساسية في مجموعة reservations كاملة وتمرير الوقت الزمني للحجز للخادم serverTimestampفي خطوة برمجية واحدة متكاملة. وفي حال تعارض الحجز مع مستخدم آخر، يتم إلغاء المعاملة بالكامل وإرجاع الخطأ بأمان دون إتلاف سلامة قاعدة البيانات.

12.5-آليات الأمان:

تم تطبيق استراتيجية أمنية صارمة متعددة الطبقات (Defense in Depth) لتأمين منصة SmartParking بالكامل على مستوى قواعد البيانات، الخدمات السحابية، والواجهات الأمامية للويب والهواتف المحمولة، وتتكون هذه الاستراتيجية من طبقتين رئيسيتين.:

أولاً: طبقة قواعد الأمان السحابية للصلاحيات(Firestore Security Rules) :

تمت كتابة قواعد أمان منظمة ومفصلة للتحكم الصارم في عمليات القراءة والكتابة السحابية مباشرة في ملف .firestore.rulesتعتمد هذه القواعد على التحقق من هوية المستخدم ودوره الموثق قبل السماح بالوصول الفعلي للبيانات، وتشتمل على التدابير التالية:

1. **التحقق من الهوية والأدوار :(Authentication and Roles)** صياغة دوال مساعدة معزولة للتحقق من هوية المستخدم (isAuthenticated)وجلب وثيقة الصلاحيات الخاصة به (getUserData) وتصنيف الأدوار برمجياً.(isAdmin, isManager, isDriver)
2. **التحكم المستند إلى الأدواروالمهام الجغرافية:**

- **المناطق والمواقف (zones & parking_spots):** يُسمح للعامة (السائقين) فقط بقراءة مناطق الموقف وجلب بيانات ، ولكن يمنع عليهم منعاً باتاً إنشاء أو تعديل أو حذف أي منطقة أو موقف. يُسمح لمديري المواقف فقط بتعديل أسعار وأوضاع المواقف الواقعة جغرافياً تحت نطاق إشرافهم الموثق برمجياً عبر مطابقة الشرط المتقاطع:

```
function isManagerOfZone(zoneId) {
  return isManager() && getUserData().zone_id == zoneId;}
```

- **الحجوزات(reservations) :**

يمنع لأي سائق قراءة أو كتابة أي حجز لا ينتمي لمعرف حسابه الفريد(request.auth.uid == resource.data.user_id) كما يتم تقييد عمليات تحديث سجلات الحجز قيد التشغيل؛ فبمجرد تنشيط الحجز وتحويل حالته إلى activeبعد دخول السيارة للموقف، يمنع السائق تماماً من تعديل أي تفاصيل مالية أو فترات زمنية، ويُسمح له فقط بتعديل حقل الحالة عند إنهاء الركن.

3. **حماية الحقول الحساسة وسجلات النظام :**تضمين شروط صارمة تمنع تحديث حقول النظام مثل دور المستخدم (role) أو تاريخ الإنشاء (created_at) حتى لو كان المستخدم صاحب الوثيقة نفسه لمنع هجمات تصعيد الصلاحيات السحابية.

ثانياً: طبقة الأمان والحماية البرمجية(Front-End Dashboard Security Services) :

تضم لوحة تحكم الويب نظام أمان مدمج في ملف security.jsيوفر حماية متكاملة ضد الهجمات الخبيثة من خلال الأنظمة الفرعية التالية:

1. **نظام منع محاولات التخمين والقوة الغاشمة (Rate Limiting Engine) :**يتعقب محرك rateLimiterمحاولات تسجيل الدخول الفاشلة للمسؤول أو المدير محلياً ويحفظها في localStorage.عند حدوث 5 محاولات خاطئة متتالية، يتم قفل الحساب برمجياً ورفض استقبال أي طلبات اتصال جديدة من المتصفح لمدة 15 دقيقة كاملة (LOCKOUT_DURATION) لحماية كلمات المرور من الاختراق.

2. **إدارة دورة حياة الجلسات وتأمين الخروج التلقائي (Session Lifecycle) :**يقوم كائن sessionManagerبإنشاء وتوثيق جلسة مشفرة قصيرة المدى تحفظ البيانات التعريفية للجهاز المتصل ووقت النشاط. يراقب التطبيق فترات الخمول والنشاط للوحة التحكم؛ وإذا لم يتم رصد أي تفاعل من المستخدم لمدة 30 دقيقة متواصلة(SESSION_TIMEOUT) ، يتم تدمير الجلسة ومسح التخزين الفرعي وتسجيل خروج المستخدم فورياً وتلقائياً لحماية شاشات الإدارة المتروكة دون مراقبة.

3. **حماية من هجمات حقن التعليمات البرمجية وتوليد المخرجات :(XSS & Sanitization)** :تطهير وتصفية كافة المدخلات النصية ومربعات البحث عبر كائن sanitizerوتمريرها على فلاتر معالجة خاصة لإزالة وتطهير رموز الوسوم البرمجية ومنع تنفيذ أي تعليمات خبيثة مثل <script>قبل تمريرها لقاعدة البيانات أو استعراضها في واجهة العرض لمنع هجمات الاختراق عبر المتصفح.

4. **تحديد بصمة الأجهزة والإنذار بالاتصالات المجهولة :(Device Fingerprinting)** :توليد شفرة تمثيلية فريدة (Hash) تمثل بصمة الجهاز المتصل بالاعتماد على خصائص محددة للنظام تشمل: نوع المتصفح، محركات المعالجة، دقة الشاشة، المنطقة الزمنية

المحددة، وعدد المعالجات البرمجية المتوفرة للنظام. يقوم النظام بالتحقق ومطابقة البصمة عند تسجيل الدخول ورصد أي أجهزة جديدة غير معروفة مسبقاً وتوثيقها كحدث أمني يتطلب الانتباه وتأكيد الهوية.

5. **تأمين البيانات والتشفير المحلي المتوافق (Secure Obfuscated Storage) :**استخدام دوال تشفير تعتمد على خوارزميات Base64 لحماية وتعمية البيانات الحساسة المحفوظة مؤقتاً في جلسات المتصفح لضمان عدم إمكانية قراءة الصلاحيات الحساسة أو التلاعب بمستويات التحكم من خلال تعديل بيانات التخزين المفتوحة للمتصفح.

13.5-إدارة الإصدارات باستخدام(Git (Version Control with Git:

تم استخدام نظام التحكم بالإصدارات **Git**لإدارة الأكواد البرمجية الخاصة بالمشروع، حيث ساهم في:

1. تتبع تاريخ التعديلات البرمجية.
2. إدارة الفروع البرمجية.(Branches)
3. دعم التطوير المتوازي.
4. حفظ نسخ احتياطية من المشروع على مستودعات سحابية.

كما تم استخدام أدوات Git المدمجة داخل بيئات التطوير مثل Visual Studio Code و IDEA IntelliJالتسهيل عمليات المزامنة وإدارة الإصدارات.

6-الشكل النهائي للنظام:

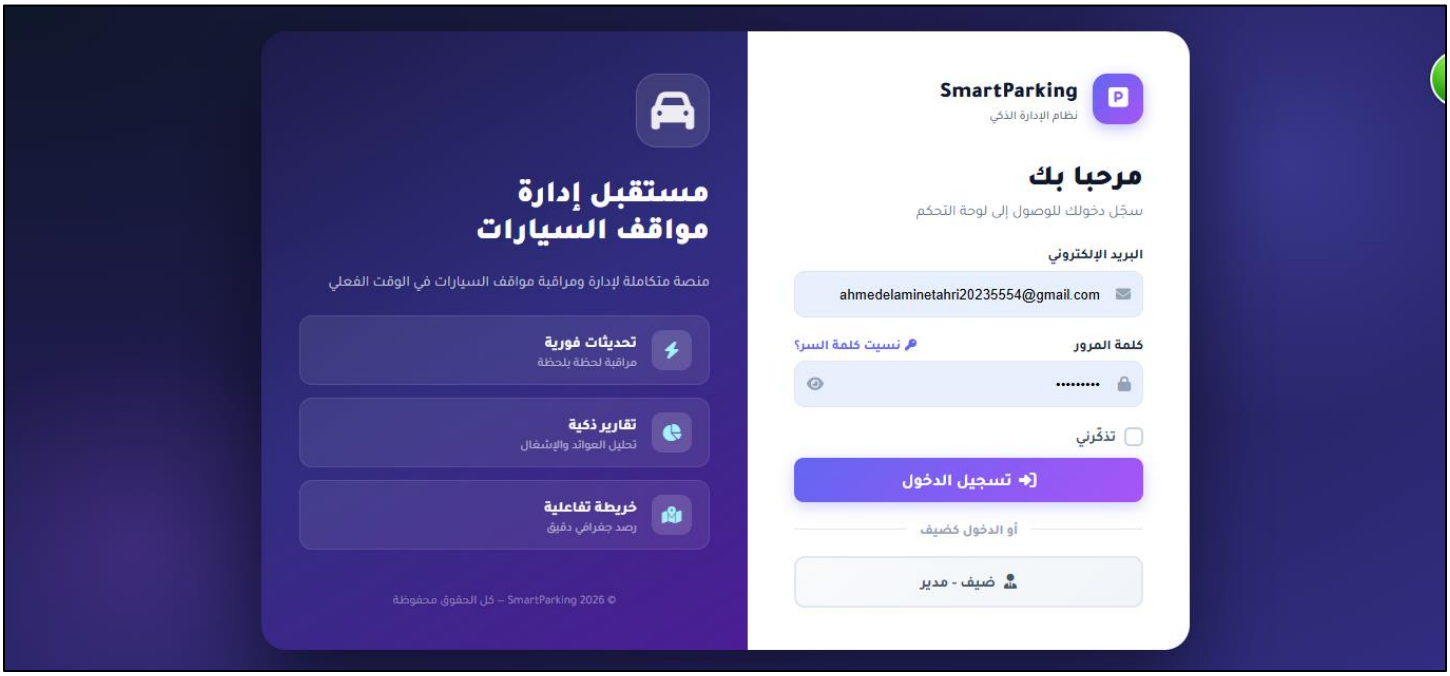
بعد اكتمال مراحل التطوير وإجراء الاختبارات الشاملة، بلغ نظام SmartParkingشكله النهائي ووظيفته الكاملة. يتجسّد النظام في واجهتين متكاملتين، تطبيق هاتف محمول ولوحة تحكم إدارية عبر الويب، يجمع بينهما خلفية سحابية موحّدة مبنية على**Firebase** ، وتتوزع الأدوار داخل النظام على ثلاثة أطراف رئيسية.

الطرف الأول هو السائق، الذي يتفاعل مع النظام عبر تطبيق الهاتف المحمول، إذ يتمكن من تصفح المواقف المتاحة على خريطة تفاعلية في الوقت الفعلي، وإجراء الحجز باختيار نوع الوقت والمركبة، ثم التوجه مباشرةً إلى الموقف عبر**GPS** ، مع إمكانية متابعة سجل حجوزاته وإدارتها في أي وقت.

الطرف الثاني هو مدير الموقف، وهو صاحب الموقف أو المشغّل المسؤول عنه، ويملك صلاحيات محددة من خلال لوحة التحكم تتيح له إدارة مناطقته ومواقفه، وتحديد الأسعار، ومتابعة الحجوزات والإيرادات المتعلقة بموقفه.

أما الطرف الثالث فهو مدير النظام(Admin) ، وهو المسؤول الأعلى صلاحيةً على المنصة بأكملها، إذ يملك رؤية شاملة وفورية على جميع المواقف والمستخدمين والحجوزات، مع أدوات متكاملة لإدارة الحسابات، ومراقبة الأداء العام، وتحليل بيانات الاستخدام في الوقت الفعلي.

وبهذا الشكل النهائي، يُجسّد النظام الهدف الذي انطلق منه؛ منصة رقمية متكاملة تربط السائق بالموقف الشاغر بأسرع وقت وأقل جهد، مع منح كل طرف الأدوات المناسبة لدوره ومسؤوليته.



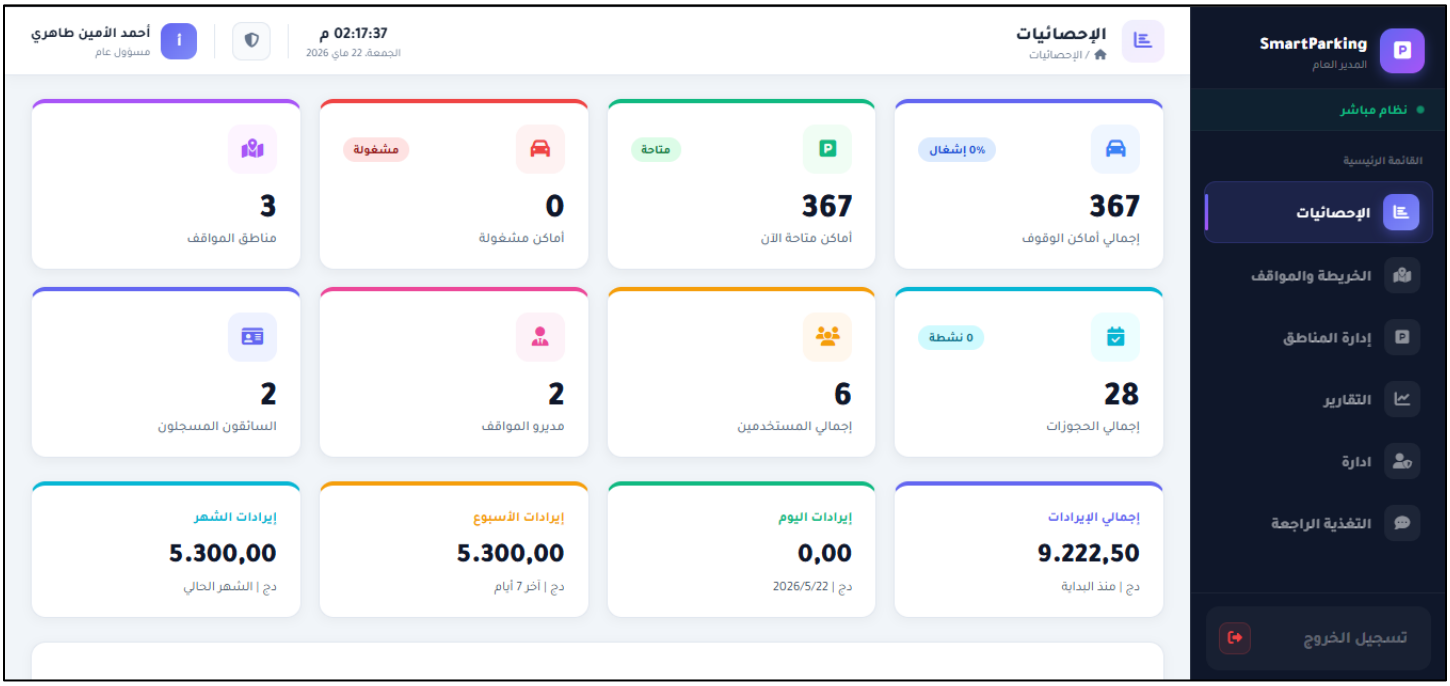
الشكل4.4:لوحة تسجيل الدخول للمسؤول او المدير

في هذه الصفحة يتم عرض واجهة تسجيل الدخول إلى نظام SmartParking ،يمكن للمستخدم إدخال البريد الإلكتروني وكلمة المرور مع إمكانية إظهار/إخفاء كلمة المرور عبر زر العين . كما يتوفر خيار "تذكّرني" لحفظ البريد الإلكتروني تلقائيًا عند الدخول مرة أخرى .

إذا نسي المستخدم كلمة السر يضغط على زر "نسيت كلمة السر؟" فينتقل إلى واجهة خاصة لإدخال بريده وإرسال رابط إعادة التعيين عبر Gmail.

بعد ملء المعلومات يضغط المستخدم على زر "تسجيل الدخول" فيتم التحقق من الهوية ويُوجَّه إلى لوحة التحكم حسب دوره(مسؤول ADMINأو مدير MANAGER) كما يوجد خيار "الدخول كضيف — مدير" للاطلاع على النظام بدون حساب حقيقي.

ميزات الحماية :عند تكرار المحاولات الخاطئة يظهر تنبيه "قفل الحساب مؤقتًا" مع عداد الوقت المتبقي، ويصبح زر تسجيل الدخول معطلاً حتى انتهاء فترة القفل.

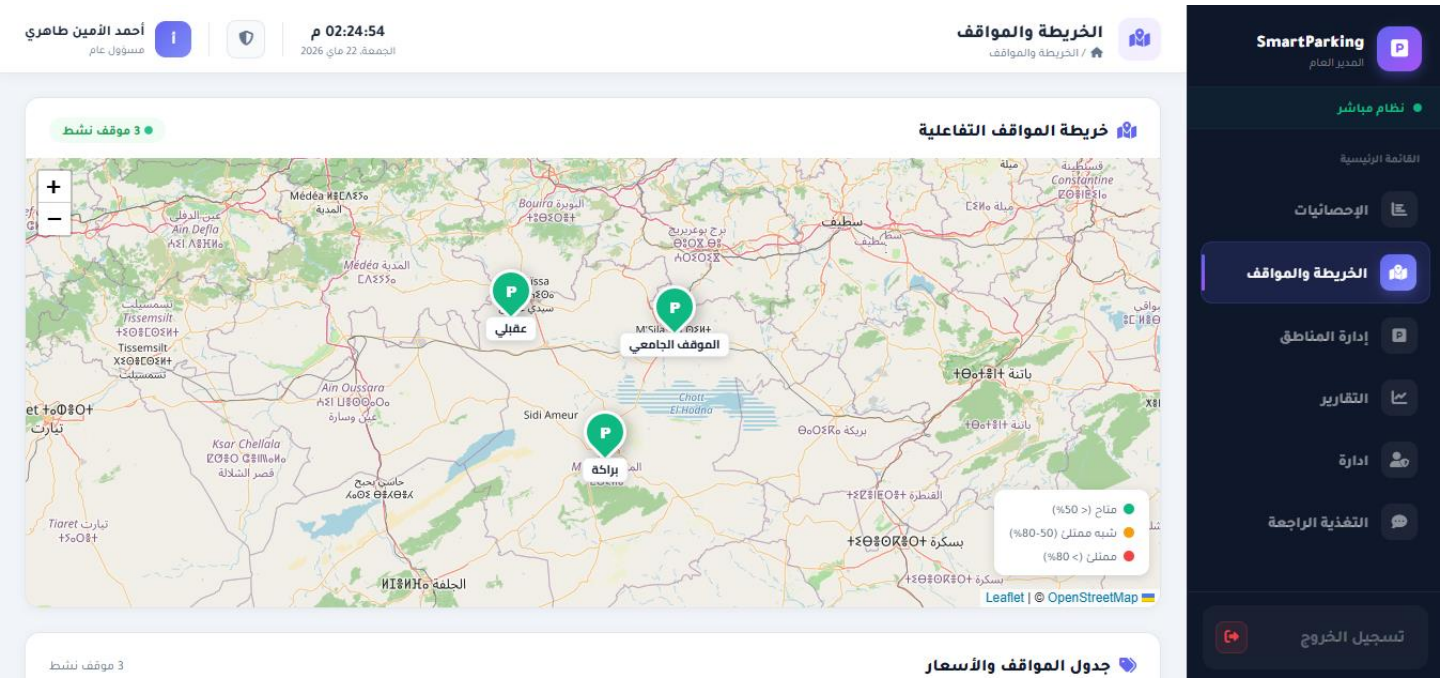


الشكل5.4:صفحة الاحصائيات للمسؤول .

في هذه الصفحة يتم عرض لوحة الإحصائيات الشاملة لنظام SmartParkingالخاصة بالمسؤول (ADMIN)، حيث تتيح مراقبة ومتابعة حالة النظام بشكل فوري ودقيق من خلال عدة أقسام تفاعلية.

تتضمن الصفحة بطاقات لمؤشرات الأداء الرئيسية (KPI Cards) تعرض إجمالي مواقف السيارات ونسبة إشغالها، المواقف الشاغرة والممتلئة حاليًا، وعدد المشتركين بالنظام مقسمين حسب أدوارهم (مديرون وسائقون). كما يتوفر قسم خاص بمراقبة الإيرادات المالية بالدينار الجزائري (دج) مقسمة حسب الفترات الزمنية (اليوم، الأسبوع، والشهر) مع حساب متوسط الإيراد لكل عملية حجز.

ولتحليل البيانات بسهولة، تعرض الصفحة رسومات بيانية تفاعلية تشمل منحنى الحركة المرورية خلال اليوم، ومخططًا شريطيًا للحجوزات اليومية لآخر 7 أيام، بالإضافة إلى مخططات دائرية توضح نسب الإشغال وتوزيع حالات الحجوزات (نشطة، معلقة، مكتملة، ملغاة).



الشكل6.4:صفحة الخريطة والمو اقف للمسؤول .

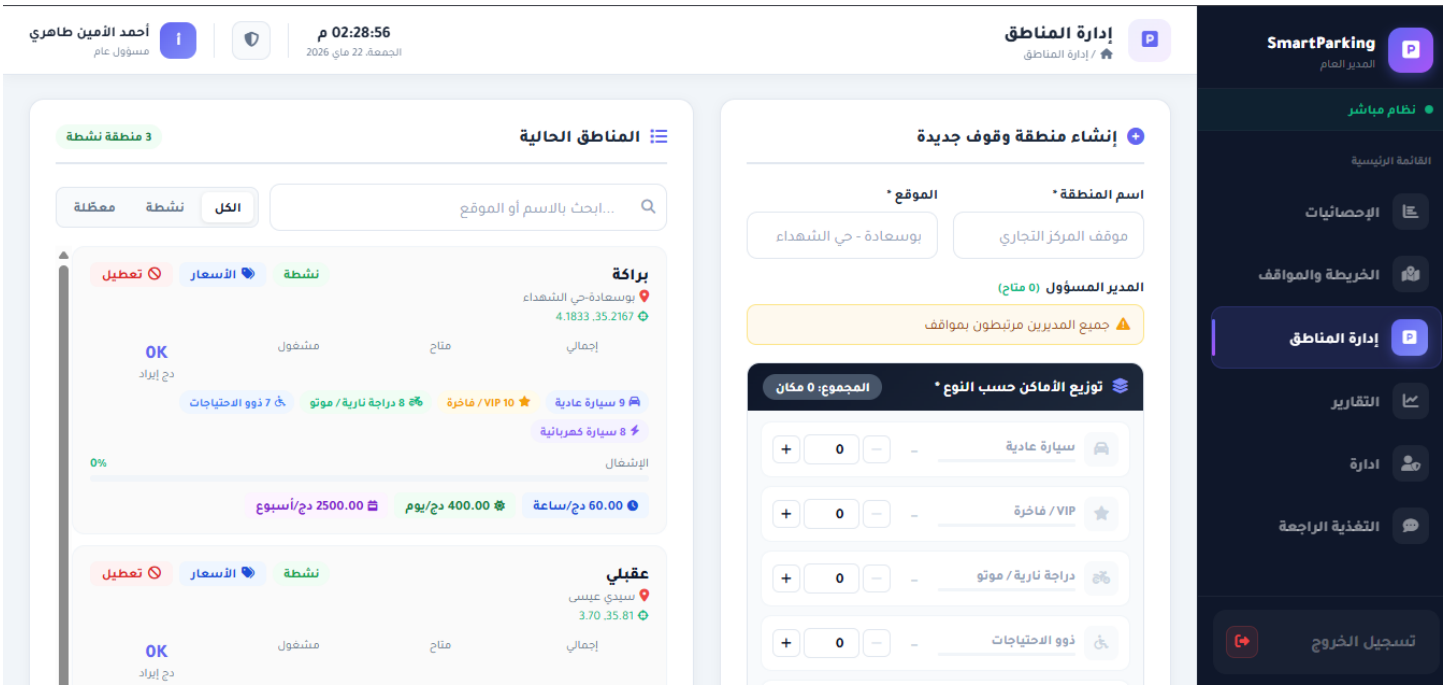
وأخيرًا، تحتوي الصفحة على جدول يعرض تفاصيل آخر 8 حجوزات مسجلة في النظام (اسم العميل، الموقف، المكان، وقت البدء، التكلفة، والحالة) مما يسهل على المسؤول الإشراف المباشر واللحظي على سير العمليات بالكامل.

في هذه الصفحة يتم عرض واجهة الخريطة والمواقف لنظام **SmartParking**الخاصة بالمسؤول(**ADMIN**) ، وتتيح متابعة الحجوزات ونسب الإشغال في المواقف بصورة جغرافية ومرئية فوريّة من خلال ثلاثة أقسام رئيسية:

1. **الخريطة التفاعلية :**تعتمد على نظام الخرائط (Leaflet) لتحديد مواقع مواقف السيارات النشطة جغرافيًا. تظهر المواقف على الخريطة كدبابيس ملونة تتغير ألوانها ديناميكيًا حسب نسبة إشغالها (الأخضر للمواقف المتاحة، البرتقالي لشبه الممتلئة، والأحمر للممتلئة تمامًا). عند النقر على أي دبوس موقف، تنبثق نافذة غنية بالتفاصيل تحتوي على العنوان الدقيق، وتوزيع الأماكن (المتاحة والمشغولة)، مع شريط تقدم بياني يوضح نسبة الإشغال بدقة.

2. **جدول المو اقف والأسعار :**يعرض جدولاً تفصيليًا لكافة المواقف النشطة، ويحتوي على أسمائها وإحداثياتها الجغرافية، وتفصيل أعداد الأماكن (إجمالي/شاغر/مشغول)، ونسب إشغالها الحالية كأشرطة مرئية ملونة، بالإضافة إلى أسعار الحجز والاشتراك المتاحة (بالساعة، باليوم، وبالأسبوع) وحالة الموقف العامة.

3. **الرسوم البيانية والتحليلات :**تتضمن مخططاً دائرياً تفصيلياً لتوزيع أنواع الأماكن تقديرياً) عادي، VIP، ذوي احتياجات خاصة، شحن كهربائي، ودراجات نارية(، إلى جانب بطاقات ملونة تلخص الأسعار (الحد الأدنى، الحد الأقصى، والمتوسط لسعر الحجز بالساعة) عبر جميع المواقف النشطة لمساعدة المسؤول في التقييم المالي المباشر.



الشكل7.4:صفحة ادارة المواقف للمسؤول .

في هذه الصفحة يتم عرض واجهة إدارة مناطق ومواقف السيارات لنظام SmartParking الخاصة بالمسؤول (ADMIN)، وتتيح التحكم الشامل والمباشر في مواقف السيارات والتعريفات الخاصة بها عبر الأقسام والوظائف التالية:

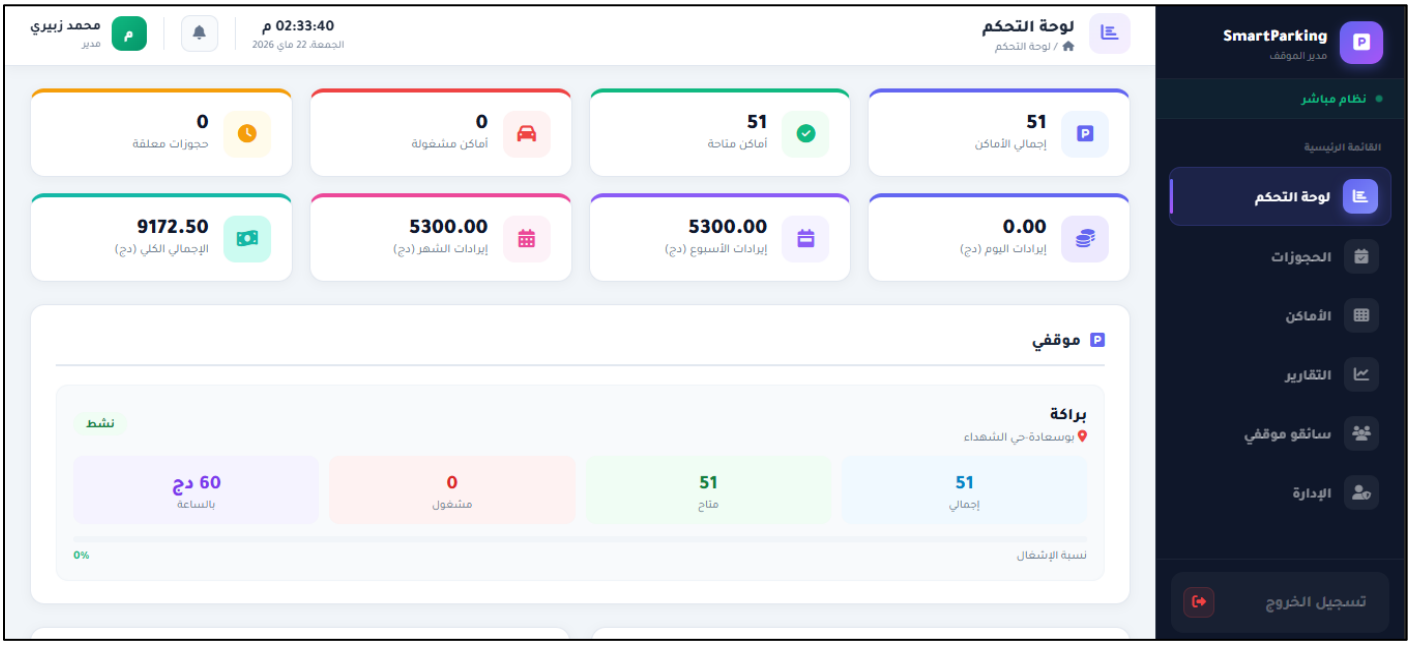
1. **إنشاء منطقة وقوف جديدة:** نموذج متكامل لتسجيل موقف جديد بالكامل، يتطلب إدخال اسم المنطقة وعنوانها الجغرافي، مع إمكانية تعيين "المدير المسؤول" عن الموقف من خلال محرك بحث منسدل ذكي (Combobox) يعرض فقط المديرين المتاحين غير المرتبطين بمواقف أخرى.

2. توزيع أنواع وأعداد المواقف: واجهة تفاعلية فريدة تسمح للمسؤول بتحديد سعة الموقف الإجمالية وتوزيعها على فئات مختلفة باستخدام أزرار الزيادة والنقصان، وهي: (مواقف سيارات عادية، مواقف VIP، مواقف ذوي الاحتياجات الخاصة، مواقف شحن السيارات الكهربائية، ومواقف الدراجات النارية)، مع تمثيل بياني لحظي لنسبة كل فئة.

3. تحديد التعريفات والإحداثيات: قسم مخصص لتسعير الموقف بالدينار الجزائري (دج) مقسمًا إلى سعر الساعة (إلزامي)، وسعر اليوم والأسبوع (اختياريين)، مع حقول مخصصة لإدخال إحداثيات الموقع الجغرافي (خطوط الطول والعرض) لضمان ظهور الموقف على الخريطة التفاعلية.

4. قائمة المناطق الحالية وتصفيتها: قسم جانبي لعرض قائمة المواقف الحالية مع محرك بحث بالاسم أو الموقع وفلاتر لعرض المواقف (الكل، النشطة، أو المعطلة). تعرض بطاقة كل موقف مؤشرًا بيانيًا ملونًا لنسبة الإشغال الحالية (أخضر، برتقالي، أحمر)، وإحصائيات مفصلة تشمل إجمالي الإيرادات المحققة، بالإضافة إلى أزرار سريعة لتفعيل/تعطيل الموقف، وزر لتحديث الأسعار مباشرة عبر نافذة منبثقة (Modal).

.



الشكل8.4:صفحة الاحصائيات للمدير

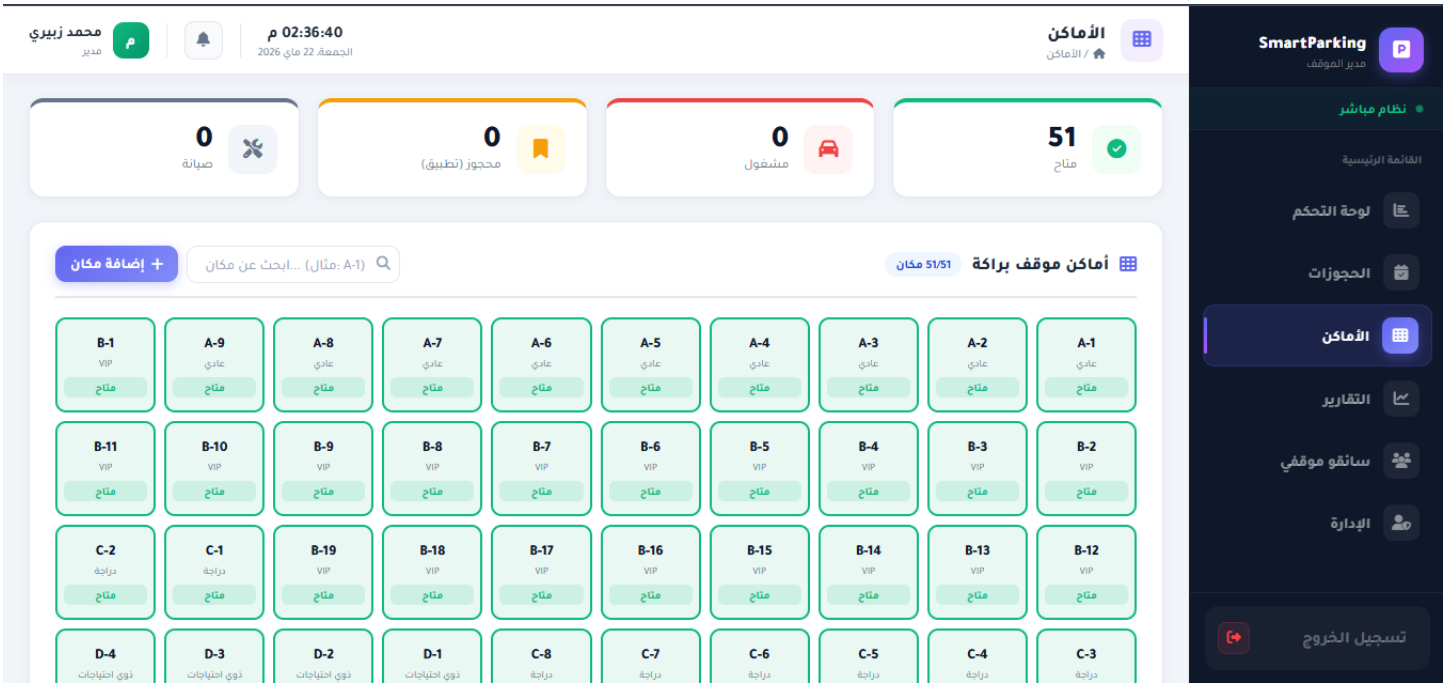
في هذه الصفحة يتم عرض لوحة الإحصائيات الخاصة بمدير الموقف (MANAGER)، حيث تتيح له متابعة حالة الموقف الخاص به وإيراداته المالية بشكل فوري ودقيق عبر الأقسام والتحليلات التالية:

1. مؤشرات الأداء والتقارير المالية: تعرض 8 بطاقات تفاعلية تلخص الأداء التشغيلي والمالي للموقف، تشمل: إجمالي الأماكن، الأماكن المتاحة والشاغرة حاليًا، الحجوزات المعلقة، بالإضافة إلى تفصيل الإيرادات بالدينار الجزائري (دج) مقسمة زمنيًا (اليوم، الأسبوع، الشهر، والإجمالي الكلي منذ البداية).

2. قسم معلومات الموقف (موقفي): يعرض كرتًا خاصًا بالموقف المُسند للمدير يوضح اسمه وعنوانه وحالته (نشط/معطل)، مع تلخيص سريع للأرقام والتعريفة بالساعة، بالإضافة إلى شريط تقدم ملون يوضح النسبة المئوية لإشغال الموقف الحالي لتسهيل المتابعة اللحظية.

3. الرسوم البيانية والتحليلات: تشمل مخططًا شريطيًا (Bar Chart) يعرض حركة وتدفق الحجوزات وتوزيعها التفصيلي على مدار 24 ساعة، ومخططًا دائريًا (Pie Chart) يوضح توزيع أعداد مواقف السيارات بين "متاح" و "مشغول" لتسهيل القراءة المرئية السريعة.

4. جدول آخر الحجوزات: يعرض قائمة محدثة لحظيًا لآخر 8 حجوزات مسجلة للموقف الخاص بالمدير، وتتضمن: اسم العميل، رقم مكان الوقوف، وقت وتاريخ بدء الحجز، التكلفة الإجمالية للحجز، وشارة ملونة توضح حالة الحجز (مكتمل، نشط، معلق، ملغي) لتأكيد العمليات عند دخول أو خروج السيارات.



الشكل9.4:صفحة المواقف للمدير.

في هذه الصفحة يتم عرض واجهة إدارة الأماكن ومساحات الوقوف الخاصة بمدير الموقف(MANAGER) ، وتتيح له التحكم الكامل والمتابعة التفصيلية والتفاعلية لكل مساحة وقوف داخل موقفه المسؤول عنه عبر الميزات التالية:

1- **شبكة الأماكن التفاعلية** :تعرض جميع مساحات الوقوف كشبكة من البطاقات الملونة التي تتغير ألوانها لحظيًا حسب حالتها الفعلية: الأخضر (متاح)، الأحمر (مشغول)، البرتقالي (محجوز مسبقاً عبر تطبيق السائق)، والرمادي (تحت الصيانة). تدعم البطاقات تأثيرات بصرية تفاعلية عند تمرير المؤشر، ويمكن النقر على أي مكان لفتح خيارات التحكم به.

2- **تسجيل الدخول والخروج اليدوي :(Walk-in)** تمكن المدير من إدارة السيارات التي تقف بدون حجز مسبق؛ فعند دخول سيارة زائر، يضغط المدير على الموقف الشاغر ويسجل بيانات الزائر (الاسم، رقم اللوحة، نوع التعريفة، والمدة المتوقعة) ليتم تغيير الحالة إلى "مشغول" وتوليد حجز يدوي. وعند خروج السيارة، يقوم النظام تلقائيًا بحساب التكلفة الفعلية بناءً على زمن الوقوف والتعريفة المعتمدة، لتوثيق الإيراد وتحرير الموقف ليصبح "متاحًا" مجددًا.

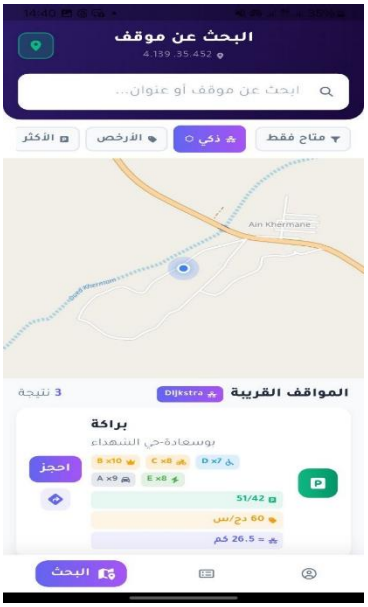
3- **التزامن التلقائي مع تطبيق الهاتف** :يقوم النظام بفحص قاعدة البيانات بشكل فوري، وتعديل حالة الموقف تلقائيًا إلى "محجوز" إذا قام سائق بحجزه عبر تطبيق الهاتف الذكي، مع توفير خيار للمدير لإلغاء هذا الحجز وتحرير الموقف يدوياً في الحالات الطارئة.

4- **إضافة مواقف جديدة دفعة واحدة (Bulk Add):** توفر نموذجًا سريعًا للمدير لإضافة مساحات وقوف إضافية إلى الموقف (حتى 50 مكانًا في المرة الواحدة)، مع اختيار الفئة المناسبة (عادي، VIP، دراجات نارية، ذوو احتياجات خاصة، سيارات كهربائية) .يقوم النظام تلقائيًا بتوليد تسلسل الأرقام بناءً على الأحرف المخصصة لكل نوع(A ، B، C، D، E)مع عرض معاينة مسبقة للمواقف قبل إنشائها.

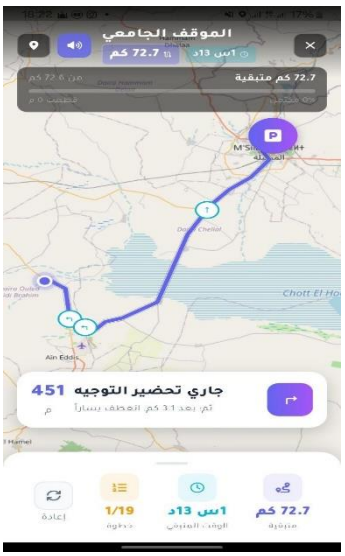
5- **محرك البحث والتصفية المتقدمة :**تحتوي الصفحة على محرك بحث سريع عن مواقف معينة بالرقم مثل(A-1)، إلى جانب فلاتر علوية تفاعلية لعزل وعرض الأماكن حسب حالتها الحالية (متاحة، مشغولة، محجوزة، صيانة) مع كاونتر يوضح إجمالي الأماكن في كل فئة



الشكل 10.4:صفحة تسجيل الدخول للسائق



الشكل 11.4:صفحة المواقف للمدير.



الشكل 12.4:صفحة التوجيه للموقف

1- صفحة تسجيل الدخول(Login Page) :

في هذه الصفحة يتم عرض واجهة تسجيل الدخول والتحقق من الهوية الخاصة بالمستخدمين للولوج إلى تطبيق **SmartParking**، وتتيح لهم الدخول الآمن والسلس إلى حساباتهم وتخصيص تجربتهم عبر الميزات التالية:

- **بوابة المصادقة الآمنة وسريعة الاستجابة :(Firebase Auth)** تمكن المستخدمين من تسجيل الدخول باستخدام البريد الإلكتروني وكلمة المرور مع توفير خيار إظهار أو إخفاء كلمة المرور بلمسة واحدة لمزيد من السهولة، إلى جانب وجود محرك ذكي لتصفية وتبسيط رسائل الخطأ الناتجة عن خوادم المصادقة (مثل: كلمة مرور خاطئة، حساب غير موجود، أو انقطاع الاتصال) وعرضها في لافتة تنبيه واضحة ومترجمة.

- **قائمة اختيار اللغة الفورية :**يتوفر في أعلى البطاقة شريط منسدل تفاعلي يسمح للمستخدم بتغيير لغة واجهة التطبيق بالكامل وبشكل لحظي بين ثلاث لغات رئيسية) **العربية**، **الفرنسية**، و**الإنجليزية**(، مع عرض أيقونات مميزة لكل لغة لضمان تجربة مستخدم مخصصة ومتكاملة.

- **آلية حفظ الجلسة واسترجاع الحساب :**توفر خيار "تذكرني" لحفظ بيانات الدخول وتسهيل الوصول لاحقاً دون الحاجة لإعادة الكتابة، مع زر مباشر "نسيت كلمة المرور؟" يفتح نافذة حوار تفاعلية لإرسال رابط إعادة تعيين كلمة المرور إلى البريد الإلكتروني المسجل.

- **الدخول السريع بصفة ضيف :(Guest Mode)** تتيح للزوار الجدد أو غير المسجلين تجاوز شاشة الدخول بالكامل والولوج مباشرة إلى التطبيق بصفة "ضيف" لتصفح المناطق والمواقف الشاغرة دون الحاجة لإتمام عملية التسجيل.

2- صفحة حجز المواقف(Booking Page) :

في هذه الصفحة يتم عرض واجهة تفصيلية وتفاعلية لحجز مساحة الوقوف داخل منطقة معينة، وتتيح للسائقين اختيار مكانهم المناسب ودفع المستحقات ومتابعة حجزهم النشط لحظيًا عبر الميزات التالية:

- **الخريطة المصغرة والتوجيه السريع** :تحتوي الصفحة في جزئها العلوي على خريطة تفاعلية مصغرة تعرض الموقع الدقيق لمنطقة المواقف المحددة مع إمكانية التحريك والتكبير، وتوفر زر توجيه فوري (Direction) لنقل المستخدم مباشرة إلى صفحة الملاحة التفصيلية.

- **محدد الأماكن التفاعلي والذكي** :يعرض شبكة المواقف مقسمة ومصنفة تلقائيًا حسب نوع المركبة (عادي، VIP، دراجات نارية، ذوو احتياجات خاصة، سيارات كهربائية) .يبرز النظام الموقف المتوافق مع سيارة المستخدم بعلامة "مناسب"، ويعرض الحالات بصريًا عبر ألوان تفاعلية تفصل بين الموقف المتاح، والموقف المحجوز، والموقف المختار للحجز الحالي.

- **خطط الحجز المرنة والمدد المتعددة** :توفر خيارات حجز متعددة ومخصصة (ساعي، يومي، أسبوعي) مع أزرار تحكم تفاعلية لزيادة أو تقليص المدة المطلوبة، ليقوم النظام بتحديث وحساب الفاتورة الإجمالية للمستخدم بشكل فوري استنادًا إلى أسعار المنطقة المحددة.

- **متابعة الحجز النشط والتحكم الفوري** :عند إتمام الحجز، تظهر لافتة ديناميكية ملونة توضح حالة الحجز الحالية) معلق Pendingأو نشط (Active مباشرة من قاعدة البيانات، مع أزرار تحكم سريعة تمكن المستخدم من إلغاء الحجز أو إنهاء الوقوف بضغطة زر.

- **دعم الدفع الفوري والتقييم السريع** :توفر زر دفع وتأكيد تفاعلي يعالج العملية بأمان، وتستدعي تلقائيًا نموذج تقييم سريع (Feedback Form)بعد انتهاء الحجز لتمكين المستخدم من تقييم تجربته وتدوين ملاحظاته لدعم جودة الخدمة.

3- صفحة التوجيه والملاحة للخريطة(Navigation Page)

في هذه الصفحة يتم عرض واجهة الملاحة والارشاد الداخلي والتوجيه خطوة بخطوة (Turn-by-Turn) لمساعدة السائقين في الوصول إلى موقفهم بدقة، وتتيح لهم التوجيه الصوتي والبصري اللحظي عبر الميزات التالية:

- **محرك الملاحة خطوة بخطوة (OSRM):** يتصل النظام بخادم الملاحة المفتوح (OSRM) لرسم المسار الأسرع على الخريطة لحظيًا انطلاقًا من موقع المستخدم الحالي وحتى إحداثيات الموقف المحجوز، مع عرض أسهم توجيهية مرسومة ومتحركة بدقة على الخريطة.

- **التوجيه الصوتي باللغة العربية (TTS):** تدعم الصفحة محرك تحويل النص إلى كلام (Text-to-Speech) لإصدار تعليمات ملاحة صوتية واضحة للمستخدم باللغة العربية (مثل: "بعد 100 متر انعطف يمينًا")، مما يساعد السائق في التركيز على الطريق دون الحاجة للنظر المستمر إلى الشاشة.

- **تتبع الموقع والتموضع التلقائي** :يتم استخدام خدمات تحديد الموقع (GPS) لتحديث مكان السيارة على الخريطة بشكل فوري كل 10 أمتار، مع ميزة التمركز التلقائي للمركبة في وسط الشاشة لضمان عدم خروجها من إطار الرؤية أثناء القيادة.
- **حساب المسافة والزمن الفعلي المتبقي** :يعرض الشريط العلوي والسفلي معلومات حية ومحدثة باستمرار عن المسافة المتبقية بالأمتار أو الكيلومترات، والزمن المتوقع للوصول بالدقائق بناءً على معطيات الطريق الفعلية وحركة المستخدم.
- **مؤشرات الانعطاف وبطاقة الوصول الذكية** :تعرض الشاشة بطاقة سفلية عائمة تحتوي على التعليمة القادمة مع أيقونة الانعطاف المناسبة (يسار، يمين، دوران، إلخ)، وعند الاقتراب من الموقف بأقل من 50 مترًا، تظهر بطاقة تهنئة تفاعلية تعلن للمستخدم وصوله بنجاح إلى وجهته المحددة.

حدود النظام:

على الرغم من تحقيق الأهداف الأساسية للمشروع وبناء منصة متكاملة ومستقرة، إلا أن عملية التطوير والتنفيذ واجهت مجموعة من التحديات والمحددات التقنية والاقتصادية والتشريعية والزمنية التي رسمت الحدود الحالية للنظام.

تمثلت العقبة الكبرى في الارتفاع الحاد لتكاليف العتاد الصلب، وخاصة كاميرات المراقبة الذكية ذات الدقة العالية وحساسات الإنترنت الأشياء مثل الحساسات المغناطيسية وحساسات الموجات فوق الصوتية. ونظراً لعدم توفر هذه المكونات بأسعار مناسبة في السوق المحلية للمشاريع الأكاديمية، بالإضافة إلى ضيق الوقت المتاح، لم يتم دمج الطبقة المادية الفعلية في هذه المرحلة من المشروع. وبالتالي، ركز النظام حالياً على الجانب البرمجي وقاعدة البيانات السحابية.

كما واجه المشروع صعوبة بالغة في دمج نظام دفع إلكتروني حقيقي مباشر يعتمد على البطاقات الائتمانية والبنكية، وذلك بسبب القيود التنظيمية والتشريعية الصارمة المفروضة على بوابات الدفع، بالإضافة إلى صعوبة الحصول على بيئات اختبارية (Sandbox) من المؤسسات المصرفية، وعدم اكتمال البنية التحتية البرمجية لبعض الخدمات المالية المحلية التي تسمح بالربط السلس مع التطبيقات الناشئة. وقد رسمت هذه الصعوبات، إلى جانب ضيق الإطار الزمني، حدود النظام الحالي كنموذج برمجي متقدم (Software Prototype) يركز على كفاءة البرمجيات وإدارة البيانات سحابياً.

الخاتمة:

في ختام هذا الفصل، نجح مشروع **SmartParking Real-Time Parking Management System** في الانتقال بفعالية من مرحلة التصميم إلى مرحلة التنفيذ العملي، محققاً بذلك معظم المتطلبات الوظيفية وغير الوظيفية التي تم تحديدها سابقاً. اعتمد التطوير على منهجية أجايل سكرم مما سمح بمرونة عالية وتسليم تدريجي للمكونات مع الاستجابة السريعة للتعديلات. وقد تم بناء النظام باستخدام تقنيات حديثة ومناسبة للسياق الجزائري، حيث شمل تطبيق الهاتف المحمول بإطار Flutter ، ولوحة التحكم الإدارية باستخدام ReactJS مع Vite ، بالإضافة إلى بنية خلفية متكاملة تعتمد على Firebase.

وقد أسفرت هذه المرحلة عن نظام متكامل يضم واجهات نهائية سلسة وبسيطة تدعم اللغة العربية، مع توفير وظائف الحجز الذكي، التوجيه، الإشعارات الفورية، وإدارة المواقف بكفاءة عالية. ورغم بعض القيود التقنية والاقتصادية، خاصة عدم دمج الطبقة المادية (IoT و ANPR) وعدم الربط الكامل مع بوابات الدفع الرسمية في هذه المرحلة، إلا أن النظام المطور يمثل نموذجاً برمجياً متيناً وقابلاً للتوسع يلبي الاحتياجات الأساسية لجميع الأطراف

الخاتمة العامة:

في ختام هذه المذكرة، يمكن القول إن مشروع **SmartParking Real-Time Parking Management System** قد سعى إلى معالجة إحدى أبرز الإشكاليات الحضرية في الجزائر، ألا وهي أزمة تسيير مواقف السيارات وما يترتب عنها من هدر زمني، ازدحام مروري، وتأثيرات اقتصادية وبيئية سلبية.

من خلال دراسة تحليلية للواقع الحالي واستقصاء ميداني شمل استبيانات ومقابلات وملاحظات ميدانية، تم تشخيص الثغرات الرئيسية في منظومة المواقف التقليدية، خاصة غياب التنسيق الرقمي والاعتماد على الأساليب اليدوية وضعف البنية التحتية التقنية. اعتمدت الدراسة على منهجية بحثية مختلطة (كمية ونوعية)، ثم انتقلت إلى مرحلة التصميم المنهجي باستخدام مخططات UML ونمذجة قاعدة بيانات Firestore. وقد تم تنفيذ المشروع باستخدام منهجية Agile (Scrum) مع تقنيات حديثة شملت Flutter لتطبيق الهاتف، و ReactJS للوحة التحكم الإدارية، و Firebase كحل سحابي.

لقد نجح المشروع في تحقيق أهدافه الأساسية في بناء منصة رقمية متكاملة ومستقرة تربط بين السائقين وأصحاب المواقف والجهات المعنية في الوقت الفعلي. ومع ذلك، واجه المشروع بعض التحديات التقنية والاقتصادية والتشريعية، أبرزها صعوبة دمج الطبقة المادية (أجهزة IoT وكاميرات ANPR) بسبب التكاليف المرتفعة، وعدم إمكانية الربط المباشر مع بوابات الدفع الإلكتروني الرسمية بسبب القيود القانونية . وعلى الرغم من هذه التحديات، فقد تم تطوير نظام وظيفي متكامل يوفر جميع الخدمات الرقمية الأساسية المستهدفة، مع تصميم مرن يدعم التطوير المستقبلي.

ويطمح المشروع في المراحل القادمة إلى تطوير محفظة رقمية افتراضية **(Virtual Wallet)** داخل التطبيق كنظام محاكاة مالي داخلي، حيث يتم شحن رصيد افتراضي للسائقين لإجراء عمليات الحجز واقتطاع المستحقات برمجيا. كما يهدف إلى الارتباط المباشر ببوابات الدفع الرسمية (البطاقات الائتمانية والبنكية) فور توفر الصلاحيات القانونية والتقنية اللازمة، ليتحول النظام إلى منصة تجارية فعلية بالكامل.

وعلى المستوى التقني، يسعى المشروع إلى التحول نحو إنترنت الأشياء والذكاء الاصطناعي من خلال ربط النظام بحساسات الموجات فوق الصوتية وكاميرات المراقبة الذكية القادرة على تحديد حالة المواقف بشكل آلي تماماً عبر خوارزميات الرؤية الحاسوبية القادرة على تحليل الصور والفيديوهات واكتشاف المركبات وتحديد حالة المواقف بشكل آلي ، مما يقضي على الأخطاء البشرية. كما سيتم دمج تقنية الواقع المعزز **(Augmented Reality)** داخل المواقف الكبرى والمغطاة لإرشاد السائق بصرياً إلى موقفه المحدد بدقة عالية.

إن **SmartParking** يمثل خطوة عملية ومهمة نحو المدن الذكية في الجزائر، حيث يمكن ان يساهم في تخفيف الازدحام المروري، وتحسين استغلال المساحات، وتعزيز التنقل الحضري المستدام. ونأمل أن يشكل هذا المشروع أساساً متيناً لمبادرات أكبر، وأن يفتح آفاقاً جديدة للبحث والتطوير في مجال التطبيقات الذكية المحلية.

.

المصادر:

[1] Office National des Statistiques (ONS), Parc automobile en Algérie, 2023. [Online]. Available: https://www.ons.dz/spip.php?article3030. [Accessed: Feb. 4, 2026].

[2] L'Echo d'Algérie, Parc automobile de l'Algérie : la structure d'âge élevée rend urgent son renouvellement, 2023. [Online]. Available: https://lechodalgerie.dz/parc-automobile-de-lalgerie-la-structure-dage-eleve-rend-urgent-son-renouvellement-dun-cout-total-estime-a-125-milliards-de-dollars-sur-cinq-ans/. [Accessed: Feb. 4, 2026].

[3] Kibo Commerce, Commerce Landscape: B2G, B2B, B2B2C, D2C, B2C, 2022. [Online]. Available: https://kibocommerce.com/blog/commerce-landscape-b2g-b2b-b2b2c-d2c-b2c. [Accessed: Feb. 7, 2026].

[4] Ecommand, Exploring Different E-Commerce Business Models, 2022. [Online]. Available: https://ecommand.co.uk/exploring-different-e-commerce-business-models-from-b2c-b2b-c2c-c2b-b2g-to-b2b2c. [Accessed: Feb. 9, 2026].

[5] Wikipedia, Types of E-commerce, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Types_of_e-commerce. [Accessed: Feb. 9, 2026].

[6] UNSW BusinessThink, The C2B2C Business Model: Parking Space Sharing (Airparking), 2021. [Online]. Available: https://www.businessthink.unsw.edu.au/articles/C2B2C-business-model-parking-space-sharing-Airparking. [Accessed: Feb. 12, 2026].

[7] StreetSecu, Smart Parking Locks and the Sharing Economy, 2022. [Online]. Available: https://www.streetsecu.com/smart-parking-locks-and-the-sharing-economy-enabling-peer-to-peer-parking-rentals. [Accessed: Feb. 14, 2026].

[8] DropDesk, How to Create a Parking Space Marketplace, 2022. [Online]. Available: https://drop-desk.com/blog/guides/how-to-create-a-parking-space-marketplace. [Accessed: Feb. 14, 2026].

[9] ParqEx, The Airbnb of Parking, 2021. [Online]. Available: https://parqex.com/the-airbnb-of-parking. [Accessed: Feb. 17, 2026].

[10] City Parking Egypt, From Concept to Reality: The Benefits of Smart Parking for Commercial Spaces, 2021. [Online]. Available: http://cityparking-eg.com/from-concept-to-reality-the-benefits-of-smart-parking-for-commercial-spaces. [Accessed: Feb. 19, 2026].

[11] Market.us, Smart Parking Systems Market Report, 2023. [Online]. Available: https://market.us/report/smart-parking-systems-market. [Accessed: Feb. 21, 2026].

[12] N. Bellil, Smart Parking, IoT and AI, LinkedIn post, 2025. [Online]. Available: https://www.linkedin.com/posts/nawel-bellil-5728902a0_smartparking-iot-ai-activity-7387927571188953088-RM10/. [Accessed: Feb. 24, 2026].

[13] M. S. Rahman et al., "Smart Parking Systems: A Review of Sensor-Based and Image Processing Approaches," International Journal of Innovative Research in Technology (IJIRT), 2021. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT172397_PAPER.pdf. [Accessed: Feb. 26, 2026].

[14] Nwave, "Advantages and Disadvantages of Smart Parking Sensors," 2019. [Online]. Available: https://www.nwave.io/pros-and-cons-of-smart-parking-systems. [Accessed: Mar. 1, 2026].

[15] Parklio, "Parking Space Detection: Sensors vs Camera – Comparison Review," 2021. [Online]. Available: https://parklio.com/en/blog/parking-space-detection-sensors-vs-camera-comparison-review. [Accessed: Mar. 1, 2026].

[16] ResearchGate, "Challenges and Opportunities in Smart Parking Sensor Technologies," 2023. [Online]. Available: https://www.researchgate.net/publication/372412628_Challenges_and_Opportunities_in_Smart_Parking_Sensor_Technologies. [Accessed: Mar. 4, 2026].

[17] A. K. Sangaiah et al., "A Smart Parking Solution Architecture Based on LoRaWAN and Kubernetes," Applied Sciences (MDPI), vol. 10, no. 13, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/13/4674. [Accessed: Mar. 7, 2026].

[18] V. Saraswat, "Designing a Real-Time Smart Parking System with Kafka & MQTT," Medium, 2025. [Online]. Available: https://medium.com/@vishal29saraswat/designing-a-real-time-smart-parking-system-with-kafka-mqtt-4e8c63c365ca. [Accessed: Mar. 10, 2026].

[19] EMQX, "Efficient IoT Communication with LoRaWAN & MQTT," 2022. [Online]. Available: https://www.emqx.com/en/blog/lorawan-and-mqtt. [Accessed: Mar. 10, 2026].

[20] M. A. Hossain et al., "Smart Indoor Parking System Based on Dijkstra's Algorithm," 2020. [Online]. Available: https://www.researchgate.net/publication/339209665_Smart_Indoor_Parking_System_Based_on_Dijkstra's_Algorithm. [Accessed: Mar. 13, 2026].

[21] F. Russo et al., "Path Planning Algorithms for Smart Parking: Review and Prospects," Journal of Traffic and Logistics Engineering (MDPI), vol. 15, no. 7, 2024. [Online]. Available: https://www.mdpi.com/2032-6653/15/7/322. [Accessed: Mar. 16, 2026].

[22] Sirin Software, "Smart Parking Solution with Cloud, Mobile App and Web Dashboard," 2021. [Online]. Available: https://sirinsoftware.com/success/smartparking-solution-with-cloud-mobile-app-and-web-dashboard. [Accessed: Mar. 19, 2026].

[23] R. Gour, "Building a Parking System App with Flutter: A Step-by-Step Guide," Medium, 2022. [Online]. Available: https://ranveergour781.medium.com/building-a-parking-system-app-with-flutter-a-step-by-step-guide-a563c5a163e1. [Accessed: Mar. 22, 2026].

[24] DLA Piper, "Data Protection Laws of the World – Algeria (Law No. 18-07)," 2023. [Online]. Available: https://www.dlapiperdataprotection.com/index.html?c=DZ&t=law. [Accessed: Mar. 25, 2026].

[25] CookieYes, "Guide on Algeria Data Protection Law: Law No. 18-07," 2023. [Online]. Available: https://www.cookieyes.com/blog/algeria-data-protection-law. [Accessed: Mar. 25, 2026].

[26] M. Pérez-Chacón et al., "Smart Parking System with Dynamic Pricing, Edge-Cloud Computing and LoRa," Sensors, vol. 20, no. 17, p. 4669, Aug. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/17/4669. [Accessed: Mar. 28, 2026].

[27] A. M. Ahmed, M. A. Youssef, and A. A. El-Khattam, "Smart parking management system with dynamic pricing," ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/356073163_Smart_parking_management_system_with_dynamic_pricing. [Accessed: Mar. 31, 2026].

[28] W. Qian and E. Rajagopal, "Dynamic pricing for reservation-based parking system: A revenue management method," Transport Policy, vol. 71, pp. 36-44, Nov. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0967070X17306704. [Accessed: Apr. 3, 2026].

[29] Z. Qian and R. Rajagopal, "Dynamic pricing and reservation for intelligent urban parking management," Transportation Research Part C: Emerging Technologies, vol. 77, pp. 226-244, Apr. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0968090X17300232. [Accessed: Apr. 3, 2026].

[30] R. Coric and M. Staron, "Smart Parking Pricing: A Machine Learning Approach," Boston University, Boston, MA, USA, Tech. Rep., 2017. [Online]. Available: https://people.bu.edu/staro/SDP2017.pdf. [Accessed: Apr. 6, 2026].

[31] A. Al-Turjman, H. Zahmatkesh, and R. Shahroze, "Deep Reinforcement Learning-Based Dynamic Pricing for Parking Solutions," Algorithms, vol. 16, no. 1, p. 32, Jan. 2023. [Online]. Available: https://www.mdpi.com/1999-4893/16/1/32. [Accessed: Apr. 9, 2026].

[32] S. Ahuja, "A Reservation-based Smart Parking System," Ph.D. dissertation, Dept. Computer Science, University of Nebraska, Lincoln, NE, USA, 2017. [Online]. Available: https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1045&context=computerscidiss. [Accessed: Apr. 12, 2026].

[33] A. Sharma and P. Kumar, "Dynamic Pricing for Parking System Using Reinforcement Learning," Academia.edu, 2022. [Online]. Available: https://www.academia.edu/79875311/Dynamic_Pricing_for_Parking_System_Using_Reinforcement_Learning. [Accessed: Apr. 12, 2026].

[34] P. Lin, X. Wang, and Y. Zhang, "Smart parking systems: A survey," IEEE Access, vol. 4, pp. 4629-4642, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7772297. [Accessed: Apr. 15, 2026].

[35] R. Singh and M. Gupta, "A Comprehensive Review of Smart Parking Systems: Technologies, Challenges, and Future Directions," International Journal of Modern Manufacturing Practices, vol. 7, no. 1, pp. 1-13, Apr. 2025. [Online]. Available: https://journalspub.com/wp-content/uploads/2025/04/1-13-IJMMP_-A-Comprehensive-Review-of-Smart-Parking-Systems-Technologies-Challenges-and-Future-Directions-1.pdf. [Accessed: Apr. 18, 2026].

[36] J. W. Creswell and J. D. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, 6th ed. Thousand Oaks, CA, USA: SAGE Publications, 2023. [Online]. Available: https://www.ucg.ac.me/skladiste/blog_609332/objava_105202/fajlovi/Creswell.pdf. [Accessed: Apr. 20, 2026].

[37] ParkWhiz Inc., ParkWhiz -- Parking App, Google Play. [Online]. Available: https://play.google.com/store/apps/details?id=com.parkwhiz.driverApp. [Accessed: Apr. 22, 2026].

[38] SpotHero, SpotHero - Find Parking. [Mobile application software]. Google Play. [Online]. Available: https://play.google.com/store/apps/details?id=com.spothero.spothero. [Accessed: Apr. 22, 2026].

[39] ParkingDZ, ParkingDZ | Logiciel N°1 de Gestion de Parking en Algérie. [Online]. Available: https://parkingdz.com/. [Accessed: Apr. 25, 2026].

[40] Garini Startup, Garini - Solution de Gestion et Réservation de Parking. Oran, Algérie. [Online]. Available: https://garini.dz/. [Accessed: Apr. 25, 2026].

[41] "Agile Methodology," GeeksforGeeks, Apr. 27, 2026. [Online]. Available: https://www.geeksforgeeks.org/software-testing/what-is-agile-methodology/. [Accessed: Apr. 28, 2026].

[42] AltexSoft, "Functional and Non-Functional Requirements: Specification and Types," AltexSoft Blog. [Online]. Available: https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/. [Accessed: Apr. 30, 2026].

[43] Firebase, "Structure Your Database," Google Firebase Documentation. [Online]. Available: https://firebase.google.com/docs/database/web/structure-data. [Accessed: May 2, 2026].

[44] OpenIllumi, "Firebase NoSQL Normalization and Denormalization," OpenIllumi. [Online]. Available: https://openillumi.com/firebase-nosql-normalization-and-denormalization/. [Accessed: May 2, 2026].

[45] Google Firebase, "Cloud Firestore data model," [Online]. Available: https://firebase.google.com/docs/firestore/data-model. [Accessed: May 4, 2026].

[46] "Unified Modeling Language (UML) Diagrams," GeeksforGeeks, Apr. 16, 2026. [Online]. Available: https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-introduction/. [Accessed: May 5, 2026].

[47] StarUML, "StarUML Official Website," 2025. [Online]. Available: https://staruml.io/. [Accessed: May 5, 2026].

[48] Visual Paradigm, "What is Entity Relationship Diagram (ERD)?," Visual Paradigm Online. [Online]. Available: https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/. [Accessed: May 7, 2026].

[49] Flutter Dev, "Flutter: Build apps for any screen," [Online]. Available: https://flutter.dev/. [Accessed: May 7, 2026].

[50] Firebase, "Cloud Functions for Firebase," [Online]. Available: https://firebase.google.com/docs/functions. [Accessed: May 9, 2026].

[51] Firebase, "Firebase Cloud Messaging (FCM)," [Online]. Available: https://firebase.google.com/docs/cloud-messaging. [Accessed: May 9, 2026].

[52] Flutter, "About Flutter," Flutter.dev. [Online]. Available: https://blog.flutter.dev/about. [Accessed: May 10, 2026].

[53] React, "React – A JavaScript library for building user interfaces," React.dev. [Online]. Available: https://react.dev/. [Accessed: May 10, 2026].

[54] Vite, "Vite | Next Generation Frontend Tooling," Vite.dev. [Online]. Available: https://vite.dev/. [Accessed: May 10, 2026].

[55] Ecma International, "ECMAScript 2015 Language Specification (ECMA-262 6th Edition)," Geneva, Switzerland, Standard ECMA-262, June 2015. [Online]. Available: https://262.ecma-international.org/6.0/. [Accessed: May 11, 2026].

[56] World Wide Web Consortium (W3C), "HTML5: A vocabulary and associated APIs for HTML and XHTML," W3C Recommendation, Oct. 28, 2014. [Online]. Available: https://www.w3.org/TR/html5/. [Accessed: May 11, 2026].

[57] World Wide Web Consortium (W3C), "Cascading Style Sheets (CSS) Snapshot 2024," W3C. [Online]. Available: https://www.w3.org/Style/CSS/. [Accessed: May 11, 2026].

[58] JetBrains, "IntelliJ IDEA - The Leading Java and Kotlin IDE," [Online]. Available: https://www.jetbrains.com/idea/. [Accessed: May 12, 2026].

[59] Google, "Android Studio - Official IDE for Android," [Online]. Available: https://developer.android.com/studio. [Accessed: May 12, 2026].

[60] Google, "Firebase Console," [Online]. Available: https://console.firebase.google.com/. [Accessed: May 13, 2026].

[61] Git, "Git - Distributed Version Control System," [Online]. Available: https://git-scm.com/. [Accessed: May 13, 2026].

[62] GitHub, "GitHub: The Complete Developer Platform," [Online]. Available: https://github.com/. [Accessed: May 13, 2026].

[63] "How to Build Role-Based Dashboards in Modern Web Applications," Canadian Software Agency. [Online]. Available: https://canadiansoftwareagency.com/role-based-dashboards/. [Accessed: May 14, 2026].

[64] Scrum Alliance, "What is Scrum?," Scrum Alliance. [Online]. Available: https://www.scrumalliance.org/about-scrum. [Accessed: May 14, 2026].

[65] Flutter Documentation, "Common architecture concepts," docs.flutter.dev. [Online]. Available: https://docs.flutter.dev/app-architecture/concepts. [Accessed: May 15, 2026].

[66] Vite, "Vite - Next Generation Frontend Tooling," [Online]. Available: https://vitejs.dev/. [Accessed: May 15, 2026].

[67] Vite, "create-vite package," npm. [Online]. Available: https://www.npmjs.com/package/create-vite. [Accessed: May 15, 2026].

[68] React Team, "React Documentation," [Online]. Available: https://react.dev/. [Accessed: May 16, 2026].

[69] npm, "npm CLI Documentation," [Online]. Available: https://docs.npmjs.com/cli/v9/commands/npm-install. [Accessed: May 16, 2026].

[70] npm, "npm run-script (npm run)," [Online]. Available: https://docs.npmjs.com/cli/v9/commands/npm-run-script. [Accessed: May 16, 2026].

[71] Firebase, "Firebase CLI (firebase-tools)," npm. [Online]. Available: https://www.npmjs.com/package/firebase-tools. [Accessed: May 17, 2026].

[72] Firebase, "Firebase CLI Documentation," [Online]. Available: https://firebase.google.com/docs/cli. [Accessed: May 17, 2026].

[73] React Router Team, "React Router Documentation," [Online]. Available: https://reactrouter.com/en/main. [Accessed: May 17, 2026].

[74] npm, "react-router-dom package," [Online]. Available: https://www.npmjs.com/package/react-router-dom. [Accessed: May 17, 2026].

[75] Leaflet, "Leaflet - an open-source JavaScript library for interactive maps," [Online]. Available: https://leafletjs.com/. [Accessed: May 18, 2026].

[76] React Leaflet, "React Leaflet Documentation," [Online]. Available: https://react-leaflet.js.org/. [Accessed: May 18, 2026].

[77] npm, "leaflet package," [Online]. Available: https://www.npmjs.com/package/leaflet. [Accessed: May 18, 2026].

[78] npm, "react-leaflet package," [Online]. Available: https://www.npmjs.com/package/react-leaflet. [Accessed: May 18, 2026].

[79] Chart.js, "Chart.js Documentation," [Online]. Available: https://www.chartjs.org/. [Accessed: May 19, 2026].

[80] React Chart.js 2, "react-chartjs-2 Documentation," [Online]. Available: https://react-chartjs-2.js.org/. [Accessed: May 19, 2026].

[81] Recharts, "Recharts Documentation," [Online]. Available: https://recharts.org/en-US. [Accessed: May 19, 2026].

[82] npm, "chart.js package," [Online]. Available: https://www.npmjs.com/package/chart.js. [Accessed: May 19, 2026].

[83] npm, "react-chartjs-2 package," [Online]. Available: https://www.npmjs.com/package/react-chartjs-2. [Accessed: May 19, 2026].

[84] npm, "recharts package," [Online]. Available: https://www.npmjs.com/package/recharts. [Accessed: May 19, 2026].

[85] GitHub, "jsPDF - PDF generation library," [Online]. Available: https://github.com/parallax/jsPDF. [Accessed: May 20, 2026].

[86] GitHub, "jsPDF-AutoTable plugin," [Online]. Available: https://github.com/simonbengtsson/jsPDF-AutoTable. [Accessed: May 20, 2026].

[87] SheetJS, "xlsx (SheetJS) library," [Online]. Available: https://github.com/SheetJS/sheetjs. [Accessed: May 20, 2026].

[88] npm, "xlsx package," [Online]. Available: https://www.npmjs.com/package/xlsx. [Accessed: May 20, 2026].

[89] EmailJS, "EmailJS JavaScript SDK Documentation," [Online]. Available: https://www.emailjs.com/docs/sdk/installation/. [Accessed: May 21, 2026].

[90] npm, "@emailjs/browser package," [Online]. Available: https://www.npmjs.com/package/@emailjs/browser. [Accessed: May 21, 2026].

[91] Firebase Core, firebase_core package, pub.dev. [Online]. Available: https://pub.dev/packages/firebase_core. [Accessed: May 21, 2026].

[92] Firebase Authentication, firebase_auth package, pub.dev. [Online]. Available: https://pub.dev/packages/firebase_auth. [Accessed: May 21, 2026].

[93] Cloud Firestore, cloud_firestore package, pub.dev. [Online]. Available: https://pub.dev/packages/cloud_firestore. [Accessed: May 21, 2026].

[94] Cloud Functions, cloud_functions package, pub.dev. [Online]. Available: https://pub.dev/packages/cloud_functions. [Accessed: May 21, 2026].

[95] Firebase Messaging, firebase_messaging package, pub.dev. [Online]. Available: https://pub.dev/packages/firebase_messaging. [Accessed: May 22, 2026].

[96] Flutter Local Notifications, flutter_local_notifications package, pub.dev. [Online]. Available: https://pub.dev/packages/flutter_local_notifications. [Accessed: May 22, 2026].

[97] Geolocator, geolocator package, pub.dev. [Online]. Available: https://pub.dev/packages/geolocator. [Accessed: May 22, 2026].

[98] Flutter Map, flutter_map package, pub.dev. [Online]. Available: https://pub.dev/packages/flutter_map. [Accessed: May 22, 2026].

[99] LatLong2, latlong2 package, pub.dev. [Online]. Available: https://pub.dev/packages/latlong2. [Accessed: May 22, 2026].

[100] HTTP, http package, pub.dev. [Online]. Available: https://pub.dev/packages/http. [Accessed: May 22, 2026].

[101] Flutter TTS, flutter_tts package, pub.dev. [Online]. Available: https://pub.dev/packages/flutter_tts. [Accessed: May 22, 2026].

[102] Mailer, mailer package, pub.dev. [Online]. Available: https://pub.dev/packages/mailer. [Accessed: May 22, 2026].

[103] Shared Preferences, shared_preferences package, pub.dev. [Online]. Available: https://pub.dev/packages/shared_preferences. [Accessed: May 22, 2026].

[104] Flutter Dotenv, flutter_dotenv package, pub.dev. [Online]. Available: https://pub.dev/packages/flutter_dotenv. [Accessed: May 22, 2026].

[105] Font Awesome Flutter, font_awesome_flutter package, pub.dev. [Online]. Available: https://pub.dev/packages/font_awesome_flutter. [Accessed: May 22, 2026].

[106] Cupertino Icons, cupertino_icons package, pub.dev. [Online]. Available: https://pub.dev/packages/cupertino_icons. [Accessed: May 22, 2026].

Abstract:

This thesis examines the current state of parking management systems in Algeria, aiming to develop a comprehensive digital solution for real-time parking management under the name SmartParking Real-Time Parking Management System. The research provides a thorough study of existing smart parking systems, identifies the shortcomings of the current situation, including the lack of digital coordination between stakeholders, excessive reliance on traditional and manual methods, and weak technical infrastructure, and proposes integrated technical and operational solutions to address these challenges. The methodology encompasses quantitative and qualitative data collection and analysis through electronic questionnaires, interviews, and field observation. Based on the findings, the functional and non-functional system requirements were defined, and the system architecture was designed using UML diagrams. This was followed by the implementation and testing of the system, which relies on a mobile application built with Flutter, an administrative dashboard using React and Vite, and an integrated cloud infrastructure via Firebase. The study contributes to building a smart and efficient parking ecosystem in the Algerian market, benefiting drivers, parking operators, and administrative entities alike.

Keywords: Smart Parking, Real-Time System, Parking Management System, SmartParking, Mobile Application, Flutter, React, Vite, Firebase, Urban Mobility, Parking Optimization, IoT-based Parking, Digital Transformation, System Design, UML Modeling, Algeria.

الملخص:

هذه المذكرة تستعرض واقع منظومة مواقف السيارات في الجزائر، وتهدف إلى تطوير نظام رقمي متكامل لإدارتها في الزمن الفعلي تحت مسمى SmartParking Real-Time Parking Management System. يقدم البحث دراسة شاملة للأنظمة الذكية القائمة، ويرصد نقائص الواقع الحالي المتمثلة في غياب التنسيق الرقمي بين الأطراف، والاعتماد المفرط على الأساليب التقليدية واليدوية، وضعف البنية التحتية التقنية، ويقترح حلولاً تقنية وتشغيلية متكاملة لمعالجة هذه التحديات. تشمل المنهجية جمع البيانات وتحليلها كمياً ونوعياً عبر استبيانات إلكترونية ومقابلات وملاحظة ميدانية، وبناءً على النتائج تم تحديد متطلبات النظام الوظيفية وغير الوظيفية وتصميم هيكله المعماري بمخططات UML .تلا ذلك تنفيذ النظام واختباره، إذ اعتمد على تطبيق هاتفي مبني بـFlutter، ولوحة تحكم إدارية بـ Reactو Vite، وبنية سحابية متكاملة عبر Firebase . تُسهم هذه الدراسة في بناء منظومة مواقف ذكية وفعّالة في السوق الجزائرية، تعود بالنفع على السائقين وأصحاب المواقف والجهات الإدارية على حدٍّ سواء.

الكلمات المفتاحية: مواقف سيارات ذكية، نظام إدارة مواقف السيارات في الزمن الحقيقي، SmartParking، تطبيق موبايل، Flutter، لوحة تحكم ويب، React، Vite، Firebase، التنقل الحضري، تحسين استغلال المواقف، التحول الرقمي، تصميم النظم، مخططاتUML ، الجزائر.