



الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

الجمهورية العربية السورية

جامعة الافتراضية

ماجستير علوم الحاسوب

منصة كراج سيارات لإدارة الجوزات

مشروع مقدم لنيل درجة الماجستير في علوم الحاسوب

جامعة الافتراضية السورية

إعداد الطالبة:

منار ياسر عسراوي

إشراف:

أ.د. عمار علي زقزوق

العام الدراسي: ٢٠٢٥

Syrian Virtual University
Virtual University
Master of Computer Science



**PARKING GARAGE PLATFORM FOR RESERVATION
MANAGEMENT**

**Submitted in Partial Fulfillment of the Requirements
for the Degree of
Master of Computer Science (MCS)**

Prepared by:

Manar Yaser Esrawi

Supervisor:

Dr. Ammar Ali Zakzouk

Year/2025

إهادء

إلى من وضعوني على طريق الحياة إلى من أفضلها على نفسي. إلى من سهرت على راحتني ليالٍ طوال، صاحبة القلب الصافي، التي أنارت دربي ودعواتها سر نجاحي. إلى من تحت قدميها الجنة، أطال الله عمرها وبارك صحتها.

(أمِي الغالية)

إلى صاحب السيرة العطرة، إلى من إذا طلبت منه نجمة أتى لي بالقمر، إلى قدوتي ومعلمي، إلى داعمي ومثلي الأعلى حفظه الله وبارك عمره.

(أبي الغالي)

إلى من لا تحلو حياتي إلى بوجودهم، إلى سendi وقوتي وأمانني، عشنا كل الأيام بحلوها ومرّها، إليهم نجوم حياتي التي لا تنطفئ، أدامهم لي الله وحفظهم

(إخوتي الأعزاء)

إلى إخوتي الذين لم تلدهم أمي.. إلى أصدقاء دربي ورفاق الحلم، الذين تميزوا بالعطاء والوفاء، الذين جملوا الطريق وذللوا الصعاب بلطفهم ودعمهم، وفقهم الله وبارك بنجاحاتهم.

(أصدقائي الأوفياء)

إلى كل من شجعني ودعمني إلى أقاربي وزملائي... إلى كل من آمن بنجاحي ...

شكر وتقدير

الحمد والشكر لله عز وجل على عظيم فضله وعطاء كرمه حمدًا طيباً مباركاً فيه

قال سيدنا محمد ﷺ: من لا يشكر الناس لا يشكر الله

قال الشاعر:

ومعلم وكفى بكل معلم شرفا ... وقد بعث النبي معلما

إلى حضرة الأستاذ الدكتور الفاضل عمار علي زقزوق الذي كان نوراً أهدي به،
ومشعلاً أستضيء برشهده وعلمه.

لقد بذل جهده ووقته الثمين ليكون مشرفاً حريصاً، ومعلماً مباركاً، دلني على طريق
العلم والعمل، وسهر على أن يرفع من كفاءتي لأحقن أعلى مراتب التميز والنجاح.
فله مني أصدق الدعاء، وأعزب الأماني بدوام التوفيق والنجاح.

لا يكتمل الشكر دون التوجه بالثناء لإدارة برنامج ماجستير علوم الحاسوب، وعلى
رأيها مدير البرنامج الدكتورة سيراً أستور، لما قدّمته من دعم لا محدود ويسرت لنا
سبل العلم والمعرفة.

كما أعبر عن امتناني غير المحدود لجميع أساتذتي الكرام، الذين لم يبخلو علينا
بعلمهم ونصحهم خلال مسيرتي الدراسية.

وختالص الشكر والتقدير لأعضاء لجنة المناقشة الأفاضل، على تكرّمهم وقبولهم
مناقشة هذا البحث.

الملخص

تمثل إدارة مواقف السيارات تحدياً مرورياً واقتصادياً في المناطق الحضرية المزدحمة. تواجه الأنظمة التقليدية قيوداً تشغيلية كبيرة، تشمل صعوبة تحديد المواقف المتاحة، وإهار الوقت في البحث، وعدم كفاءة عمليات الدفع اليدوي، بالإضافة إلى عجز الإدارات عن المراقبة الآلية للإشغال وتحليل البيانات لتعزيز الإيرادات.

انطلاقاً من هذه المشكلة، تقدم هذه الأطروحة تصميم وتنفيذ منصة ويب متكاملة لإدارة كراجات السيارات. يهدف النظام إلى معالجة فجوتين رئيسيتين: تحسين تجربة المستخدم (سواء كان زائراً عابراً أو مشتركاً دائماً) وتمكين الإدارة التشغيلية من خلال أدوات قوية للرقابة واتخاذ القرار. يعتمد الحل على فلسفة الإدارة الإلكترونية لتحويل العمليات اليدوية إلى نظام آلي مركزي.

تم تطوير النظام باستخدام مجموعة من التقنيات الحديثة، شملت React.js و Node.js و Express.js و QR لبناء واجهات المستخدم والخوادم، بالإضافة إلى دمج تقنيات مثل رموز QR لتبسيط عمليات الدخول والخروج، وخدمة Twilio لإرسال الإشعارات والرسائل الإلكترونية. واعتمد المشروع منهجهية الشلال التكراري لإدارة دورة الحياة، مما أتاح مرونة في مراجعة المتطلبات وتصحيح الأخطاء في مراحل مبكرة.

يكمن الإسهام الرئيسي للنظام في مجموعة من الميزات المتكاملة، أهمها: (١) البحث والجز المسبق للمواقف مع دفع إلكتروني، (٢) نظام متكامل لإدارة الاشتراكات الشهرية والسنوية، (٣) لوحة تحكم شاملة تمكن المديرين والموظفين من مراقبة العمليات الآلية وإعداد التقارير التحليلية، (٤) وأدوات مرنة للتعامل مع الحالات الطارئة مثل فقدان رمز الاستجابة السريعة.

النتيجة النهائية هي منتج كامل يعمل كحل مركزي، يستفيد منه جميع أصحاب المصلحة: حيث يمنح السائقين تجربة سلسة وموثقة، ويمكن موظفي الخدمة من إدارة العمليات بكفاءة، ويزود مديري الكراجات بأدوات قوية لتحسين الأداء التشغيلي والمالي. يُوصى بالنظام كحل عملي يساهم في تخفيف الازدحام المروري ويدعم تحول المدن نحو مفهوم الذكية والاستدامة.

Abstract

Urban parking management is a serious traffic and monetary problem in highly populated cities. Conventional systems have huge operational drawbacks, such as inability to detect available spaces, time loss searching, inefficiency in the manual mode of payment, and failure by administrations to dynamically monitor occupancy or exploit the data for increased revenue.

Addressing this problem, the present thesis proposes the design and implementation of an integrated web-based platform for parking garage management. The system aims to bridge two primary gaps: enhancing the user experience, whether for occasional visitors or regular subscribers—and empowering operational management through robust monitoring and decision-making tools. The solution adopts the philosophy of e-management to transform manual processes into a centralized automated system.

The platform was built through the utilization of a suite of contemporary technologies, such as React.js and Node.js and Express.js for developing user and server-side components, and the usage of QR codes for facilitating entry and exit processes, and Twilio services for notifying and e-mailing. An iterative waterfall approach was taken by the project for managing lifecycles, allowing for flexibility in going back and forth concerning requirements and error rectification at early phases.

The system's main contributions include: (1) search and advance reservation of parking spaces with electronic payment, (2) a comprehensive subscription management module for monthly and annual plans, (3) an all-inclusive dashboard enabling managers and staff to monitor real-time operations and generate analytical reports, and (4) flexible mechanisms for handling emergencies, such as lost QR codes.

The outcome is a fully operational, centralized system that benefits everyone involved: providing drivers with smooth and reliable experience, enabling service personnel to work effectively, and providing garage managers with effective tools for improved operational and financial success. The system is a solution that can truly exist in the real world that has the potential to reduce traffic congestion and help cities become smart and sustainable.

فهرس المحتويات

الفصل الأول: الإطار النظري العام.....	١٣
١.١- مقدمة:.....	١٣
٢.١- المشكلة البحثية:.....	١٤
٣.١- أهمية البحث:.....	١٤
٤.١- أهداف البحث:.....	١٥
٥.١- النتائج التطبيقية والجهات المستفيدة منها:.....	١٦
٥.١.١- النتائج التطبيقية:.....	١٦
٥.١.٢- الجهات المستفيدة:.....	١٧
الفصل الثاني: التقنيات والبرمجيات المستخدمة ومنهجياتها.....	١٨
١.٢- مقدمة:.....	١٨
٢.٢- لغة توصيف صفحات الويب HTML.....	١٨
٣.٢- لغة تنسيق وتصميم الواجهات CSS.....	٢٢
٤.٢- مكتبة تصميم واجهات جاهزة Tailwind CSS.....	٢٣
٥.٢- لغة برمجة تفاعلية للويب JavaScript.....	٢٥
٦.٢- مكتبة لبناء واجهات المستخدم React.js.....	٢٧
٧.٢- بيئة تشغيل الخادم Node.js.....	٣٠
٩.٢- نظام إدارة قواعد البيانات MySQL.....	٣٤
١٠.٢- تقنية ترميز بصري سريع QR Code.....	٣٦
١١.٢- مكتبة إرسال البريد في Node.js (Nodemailer).....	٣٨
١٢.٢- مكتبات إشعارات React (react-toastify).....	٤٠
الفصل الثالث: الدراسة المرجعية.....	٤٢
١.٣- مقدمة:.....	٤٢
٢.٣- أبرز الدراسات:.....	٤٢
الفصل الرابع: الدراسة التحليلية.....	٤٨
٤.٤- منهجية النظام:.....	٤٨
٥.٤- مراحل تشكيل النظام:.....	٤٩
٥.٤.٤- مرحلة تحليل المتطلبات (Requirements Analysis).....	٤٩
٥.٤.٥- مرحلة التصميم (System Design).....	٤٩

٥٠	: مرحلة التطوير (Implementation)	٣.٢.٤
٥٠	: مرحلة الاختبار (Testing)	٤.٢.٤
٥١	٣.٣- تحليل النظام:.....	
٥١	١.٣.٤- جمع المتطلبات من المستخدمين المحتملين.....	
٥٢	٢.٣.٤- تحليل النظم المشابهة:.....	
٥٣	٤.٤- متطلبات النظام:.....	
٥٣	١.٤.٤- المتطلبات الوظيفية:.....	
٥٤	٢.٤.٤- المتطلبات غير الوظيفية:.....	
٥٥	٣.٤.٤- المستخدمون والأدوار:	
٥٥	٤.٥- الوظائف الأساسية للنظام:.....	
٥٩	٦.٤- حالات الاستخدام:.....	
١٠٧.....	٧.٤- مخططات التتفق:.....	
١٠٨.....	١.٧.٤- مخطط تفصيلي للعلاقات بين مكونات النظام:.....	
١١٠.....	٢.٧.٤- مخطط تدفقى لعملية تسجيل الدخول:.....	
١١٣	الفصل الخامس: تصميم النظام.....	
١١٣.....	١.٥- مقدمة:.....	
١١٤.....	٢.٥- مخطط الفئات (Class Diagram)	
١١٧.....	٣.٥- مخططات التسلسل:.....	
١١٧.....	١.٣.٥- مخطط sign in	
١١٩.....	٢.٣.٥- مخطط booking	
١٢١.....	٣.٣.٥- مخطط Sequence For All Actors	
١٢٣.....	٤.٥- تصميم قاعدة البيانات:.....	
١٤٨.....	٥.٥- تصميم الواجهات:.....	
١٤٨.....	١.٥.٥- الواجهة الرئيسية للمنصة:.....	
١٥٠.....	٢.٥.٥- واجهة العميل الأساسية:.....	
١٥٢.....	٣.٥.٥- واجهة الموظف الأساسية:.....	
١٥٦.....	٤.٥.٥- الواجهة الأساسية لمدير الكراج:.....	
١٦٠.....	٥.٥.٥- الواجهة الأساسية لمدير النظام:.....	
١٦٢.....	٦.٥.٥- واجهة تسجيل الدخول:.....	

١٦٣.....	- واجهة إنشاء حساب:.....	٧.٥.٥
١٦٣.....	- دورة حياة طلب الحجز من قبل العميل:.....	٨.٥.٥
١٧٣.....	- دورة حياة طلب الحجز عند الموظف:.....	٩.٥.٥
١٧٩.....	- حالات ثانوية للحجز:.....	١٠.٥.٥
١٨٣.....	- البريد الإلكتروني الذي يصل للعميل من النظام:.....	١١.٥.٥
١٨٧.....	الخاتمة.....	
١٨٨.....	الآفاق المستقبلية.....	
١٨٩.....	الملحقات.....	
١٩٧	المراجع.....	

فهرس الجداول

جدول (١٠.٣) : مقارنة بين الدراسات المرجعية للموقع المشابهة للمنصة.....	٤٥
جدول (١٠.٤) : الوظائف الأساسية للنظام	٥٧
جدول (٢٠.٤) : حالة الاستخدام ١ تسجيل الدخول للنظام.....	٦٢
جدول (٤٠.٤) : حالة الاستخدام ٣ إنشاء حساب جديد.....	٦٥
جدول (٥٠.٤) : حالة الاستخدام ٤ عرض قائمة الكراجات.....	٦٦
جدول (٦٠.٤) : حالة الاستخدام ٥ عرض أماكن الكراجات.....	٦٧
جدول (٨٠.٤) : حالة الاستخدام ٦ عرض تفاصيل الكراج.....	٦٨
جدول (٩٠.٤) : حالة الاستخدام ٧ حجز موقف	٧٠
جدول (١١٠.٤) : حالة الاستخدام ٩ عرض تفاصيل الحجز.....	٧٢
جدول (١٢٠.٤) : حالة الاستخدام ١٠ إدارة الحجوزات.....	٧٤
جدول (١٢٠.٤) : حالة الاستخدام ١١ إلغاء الحجز.....	٧٥
جدول (١٣٠.٤) : حالة الاستخدام ١٢ شراء اشتراك جديد	٧٦
جدول (١٤٠.٤) : حالة الاستخدام ١٣ إدارة الاشتراكات.....	٧٨
جدول (١٥٠.٤) : حالة الاستخدام ١٤ إلغاء اشتراك	٧٩
جدول (١٦٠.٤) : حالة الاستخدام ١٥ دفع إلكتروني.....	٨١
جدول (١٧٠.٤) : حالة الاستخدام ١٦ تسجيل الشكاوى.....	٨٢
جدول (١٨٠.٤) : حالة الاستخدام ١٧ تقييم الخدمة.....	٨٣
جدول (١٩٠.٤) : حالة الاستخدام ١٨ إنشاء حجز يدوى للعملاء.....	٨٤
جدول (٢٠٠.٤) : حالة الاستخدام ١٩ مسح QR.....	٨٦
جدول (٢١٠.٤) : حالة الاستخدام ٢٠ إعادة إصدار QR	٨٧
جدول (٢٢٠.٤) : حالة الاستخدام ٢١ تسجيل الحوادث.....	٨٨
جدول (٢٣٠.٤) : حالة الاستخدام ٢٢ تفعيل الدخول اليدوي.....	٨٩
جدول (٢٤٠.٤) : حالة الاستخدام ٢٣ إدارة المواقف.....	٩٠
جدول (٢٥٠.٤) : حالة الاستخدام ٢٤ ضبط ساعات العمل.....	٩١
جدول (٢٦٠.٤) : حالة الاستخدام ٢٥ إدارة المحافظ.....	٩٢
جدول (٢٧٠.٤) : حالة الاستخدام ٢٦ تحديد الأسعار (الاشتراكات/المؤقتة).....	٩٣
جدول (٢٨٠.٤) : حالة الاستخدام ٢٧ عرض الموظفين.....	٩٥
جدول (٢٩٠.٤) : حالة الاستخدام ٢٨ إضافة موظفين جدد	٩٦

جدول (٤٠.٤) : حالة الاستخدام _٢٩ إدارة حسابات المستخدمين	٩٧
جدول (٣١.٤) : حالة الاستخدام _٣٠ إضافة كراج.....	٩٩
جدول (٣٢.٤) : حالة الاستخدام _٣١ إضافة صورة للكراج.....	١٠٠
جدول (٣٣.٤) : حالة الاستخدام _٣٢ الاطلاع على التقارير المقدمة.....	١٠١
جدول (٣٤.٤) : حالة الاستخدام _٣٣ مخططات الحجوزات والإيرادات.....	١٠٢
جدول (٣٥.٤) : حالة الاستخدام _٣٤ عرض التقييم للخدمات.....	١٠٣
جدول (٣٦.٤) : حالة الاستخدام _٣٥ إضافة حساب مدير كراج.....	١٠٥
جدول (٣٧.٤) : حالة الاستخدام _٣٦ تسجيل الخروج من النظام.....	١٠٦
جدول (١٠.٥) : جدول قاعدة البيانات <u>users</u> _١	١٢٨
جدول (٢٠.٥) : جدول قاعدة البيانات <u>garages</u> _٢	١٢٩
جدول (٣٠.٥) : جدول قاعدة البيانات <u>notifications</u> _٣	١٣١
جدول (٤٠.٥) : جدول قاعدة البيانات <u>garage_images</u> _٤	١٣٢
جدول (٥٠.٥) : جدول قاعدة البيانات <u>parking_spots</u> _٥	١٣٣
جدول (٦٠.٥) : جدول قاعدة البيانات <u>garage_employees</u> _٦	١٣٥
جدول (٧٠.٥) : جدول قاعدة البيانات <u>wallets</u> _٧	١٣٦
جدول (٨٠.٥) : جدول قاعدة البيانات <u>bookings</u> _٨	١٣٧
جدول (٩٠.٥) : جدول قاعدة البيانات <u>transactions</u> _٩	١٣٩
جدول (١٠٠.٥) : جدول قاعدة البيانات <u>ratings</u> _١٠	١٤١
جدول (١١٠.٥) : جدول قاعدة البيانات <u>audit_logs</u> _١١	١٤١
جدول (١٢٠.٥) : جدول قاعدة البيانات <u>subscriptions</u> _١٢	١٤٢
جدول (١٣٠.٥) : جدول قاعدة البيانات <u>loyalty_history</u> _١٣	١٤٣
جدول (١٤٠.٥) : جدول قاعدة البيانات <u>external_customers</u> _١٤	١٤٤
جدول (١٥٠.٥) : جدول قاعدة البيانات <u>garage_access_logs</u> _١٥	١٤٥
جدول (١٦٠.٥) : جدول قاعدة البيانات <u>complaints</u> _١٦	١٤٥
جدول (١٧٠.٥) : جدول قاعدة البيانات <u>incidents</u> _١٧	١٤٧

فهرس الأشكال

الشكل (١٠.٤) - نموذج الشلال التكراري (Iterative Waterfall Model)	٤٨
الشكل (٢٠.٤) - مخطط حالات الاستخدام (الممثلين الرئيسيين)	٦٠
الشكل (٣٠.٤) - مخطط حالات الاستخدام (الممثلين الثانويين)	٦٠
الشكل (٤٠.٤) - العلاقات بين المكونات المختلفة للنظام حسب دور كل مستخدم	١٠٩
الشكل (٥٠.٤) - مخطط تدفقي لتسجيل دخول المستخدم	١١١
الشكل (١٠.٥) - مخطط الفئات (Class Diagram)	١١٤
الشكل (٢٠.٥) - مخطط تسلسل sign in - (Sequence Diagram)	١١٧
الشكل (٣٠.٥) - مخطط تسلسل booking - (Sequence Diagram)	١١٩
الشكل (٤٠.٥) - مخطط تسلسل Sequence For All Actors - (Sequence Diagram)	١٢٢
الشكل (٥٠.٥) - مخطط (Entity Relationship Diagram) ERD	١٢٥
الشكل (٦٠.٥) - الواجهة الرئيسية للمنصة	١٤٩
الشكل (٧٠.٥) - واجهة العميل الأساسية	١٥٠
الشكل (٨٠.٥) - واجهة الكراجات لدى العميل	١٥٠
الشكل (٩٠.٥) - واجهة الحجوزات لدى العميل	١٥١
الشكل (١٠٠.٥) - واجهة الاشعارات لدى العميل	١٥١
الشكل (١١٠.٥) - واجهة الموظف الأساسية	١٥٢
الشكل (١٢٠.٥) - واجهة الحجوزات عند الموظف	١٥٢
الشكل (١٣٠.٥) - واجهة مسح رمز QR عند الموظف	١٥٣
الشكل (١٤٠.٥) - واجهة مراقبة التأخير عند الموظف	١٥٣
الشكل (١٥٠.٥) - واجهة الملف الشخصي عند للموظف	١٥٤
الشكل (١٦٠.٥) - واجهة الشكاوى عند الموظف	١٥٤
الشكل (١٧٠.٥) - واجهة عرض تفاصيل الشكاوى عند الموظف	١٥٥
الشكل (١٨٠.٥) - الواجهة الأساسية لمدير الكراج	١٥٦
الشكل (١٩٠.٥) - واجهة ضبط ساعات العمل عند مدير الكراج	١٥٦
الشكل (٢٠٠.٥) - واجهة إدارة الموظفين عند مدير الكراج	١٥٧
الشكل (٢١٠.٥) - واجهة التقارير المقدمة لمدير الكراج	١٥٧
الشكل (٢٢٠.٥) - واجهة إعدادات الكراج عند مدير الكراج	١٥٨

الشكل (٢٣.٥) - واجهة إدارة الصور مدير الكراج.....	١٥٨
الشكل (٢٤.٥) - واجهة المخطوطات عند مدير الكراج.....	١٥٩
الشكل (٢٥.٥) - واجهة إدارة المحافظ مدير الكراج.....	١٥٩
الشكل (٢٦.٥) - الواجهة الأساسية لمدير النظام.....	١٦٠
الشكل (٢٧.٥) - واجهة إدارة المستخدمين مدير النظام.....	١٦١
الشكل (٢٨.٥) - واجهة إدارة الكراجات عند مدير النظام.....	١٦١
الشكل (٢٩.٥) - واجهة الملف الشخصي لمدير النظام.....	١٦٢
الشكل (٣٠.٥) - واجهة تسجيل الدخول.....	١٦٢
الشكل (٣١.٥) - واجهة إنشاء حساب.....	١٦٣
الشكل (٣٢.٥) - واجهة فتح صفحة الكراجات.....	١٦٤
الشكل (٣٣.٥) - واجهة اختيار كراج للحجز.....	١٦٤
الشكل (٣٣.٥) - واجهة نموذج بيانات الحجز	١٦٥
الشكل (٣٤.٥) - واجهة اختيار نوع الموقف.....	١٦٦
الشكل (٣٥.٥) - واجهة بيانات الحجز مع حساب التكلفة الإجمالية.....	١٦٧
الشكل (٣٦.٥) - واجهة سؤال تأكيد الحجز.....	١٦٨
الشكل (٣٧.٥) - واجهة تأكيد الحجز.....	١٦٨
الشكل (٣٨.٥) - واجهة وجود حجز نشط.....	١٦٩
الشكل (٣٩.٥) - واجهة حالة الحجز قبل الدفع.....	١٧٠
الشكل (٤٠.٥) - واجهة تأكيد الدفع.....	١٧٠
الشكل (٤١.٥) - واجهة حالة الحجز بعد الدفع	١٧١
الشكل (٤٢.٥) - واجهة معلومات الحجز في قائمة الحجوزات.....	١٧١
الشكل (٤٣.٥) - واجهة رمز QR.....	١٧٢
الشكل (٤٤.٥) - واجهة تفاصيل الحجز بعد مسح الدخول عند العميل.....	١٧٢
الشكل (٤٥.٥) - واجهة مسح QR عند الموظف.....	١٧٣
الشكل (٤٥.٥) - واجهة مسح QR دخول.....	١٧٤
الشكل (٤٦.٥) - واجهة تأكيد الدخول للكراج.....	١٧٥
الشكل (٤٧.٥) - واجهة تفاصيل الحجز بعد مسح الدخول عند الموظف.....	١٧٦
الشكل (٤٨.٥) - واجهة مسح QR خروج.....	١٧٧

الشكل (٤٩.٥) - واجهة تأكيد الخروج من الكراج.....	١٧٨
الشكل (٥٠.٥) - واجهة تفاصيل الحجز بعد مسح الخروج عند الموظف.....	١٧٩
الشكل (٥١.٥) - واجهة دخول مبكر أكثر من ٥ دقائق.....	١٧٩
الشكل (٥٢.٥) - واجهة دخول مبكر أقل من ٥ دقائق.....	١٨٠
الشكل (٥٣.٥) - واجهة مسح دخول لحجز نشط.....	١٨١
الشكل (٥٤.٥) - واجهة حجوزات قادمة.....	١٨٢
الشكل (٥٥.٥) - واجهة حجوزات متاخرة.....	١٨٢
الشكل (٥٦.٥) - واجهة بريد ترحبيي.....	١٨٣
الشكل (٥٧.٥) - واجهة بريد تأكيد الحجز.....	١٨٤
الشكل (٥٨.٥) - واجهة بريد إلغاء حجز.....	١٨٤
الشكل (٥٩.٥) - واجهة بريد تتبية حجز.....	١٨٥
الشكل (٦٠.٥) - واجهة بريد تجاوز الوقت.....	١٨٥
الشكل (٦١.٥) - واجهة بريد إعادة تعين كلمة المرور	١٨٦

الفصل الأول: الإطار النظري العام

1.1 - مقدمة:

أصبحت إدارة مواقف السيارات (الكراجات) تحدياً متزايد التعقيد في المدن المزدحمة. يواجه السائقون يومياً صعوبات جمة في تحديد موقع المواقف الشاغرة، مما يؤدي إلى إهار الوقت وزيادة الازدحام المروري. إلى جانب ذلك، تستهلك عمليات الدفع اليدوي التقليدية وقتاً طويلاً وتنقل كاهل المستخدمين والموظفين على حد سواء. من جهة أخرى، تعاني إدارات الكراجات من نقص الأدوات الفعالة لمراقبة حالة الإشغال بشكل آني، وإدارة اشتراكات المستخدمين بكفاءة، وتحسين العمليات التشغيلية لتعزيز الإيرادات وخفض التكاليف. انطلاقاً من هذه التحديات الملحة، اعتمد هذا المشروع على تطوير موقع ويب شامل ومتكملاً، يعمل كمنصة مركزية لإدارة كراجات السيارات، يركز على خدمة جميع الأطراف المعنية: السائقين (سواء كانوا زائرين عابرين أو مشتركين دائمين)، وموظفي التشغيل الميدانيين، ومديري الكراجات المسؤولين عن التخطيط والرقابة.

تعتمد فلسفة التطبيق على مفهوم "الإدارة الإلكترونية" والتي تُعرف بأنها: منظومة إلكترونية متكاملة تهدف إلى تحويل العمل الإداري العادي من إدارة يدوية إلى إدارة باستخدام الحاسب وذلك بالاعتماد على نظم معلوماتية قوية تساعد في اتخاذ القرار الإداري بأسرع وقت وبأقل التكاليف [١].

وتمثل أحد المحاور الرئيسية للتطبيق في تمكين "الحجز الإلكتروني" المسبق للمواقف، حيث يُعرف هذا المفهوم (الحجز الإلكتروني) بأنه: عملية حجز تذاكر الخدمات الخاصة بمحال السياحة والسفر والفنادق والمواصلات عن طريق موقع إلكترونية. وفي ضوء هذا التعريف يربط الحجز الإلكتروني، بمفهوم التذكرة الإلكترونية التي يتم فيها تخزين تفاصيل بيانات الحجز والتذكرة بشكل آمن في نظام شركات الحجز الآلي، ونعرف التذكرة الإلكترونية على أنها طريقة لتوثيق بيع التذاكر وتخزينها في قواعد البيانات الخاصة بالشركة وتعقب آلية استعمالها دون الحاجة إلى إصدار بطاقة ورقية (تسليم أو استلام تذاكر ورقية) من خلال الحجز عبر الانترنت [٢]. تعمل المنصة على تطبيق هذا المفهوم بشكل مبتكر في سياق إدارة مواقف السيارات، لتحويل تجربة المستخدم وتبسيط العمليات الإدارية بشكل جذري.

٢.١ - المشكلة البحثية:

تواجه أنظمة إدارة مواقف السيارات التقليدية تحديات متربطة تعيق تجربة المستخدمين وتعقد العمليات التشغيلية. فمن ناحية السائقين - سواء كانوا زائرين عابرين أو مشتركين دائمين - يعانون من غياب الرؤية الواضحة حول توافر المواقف الفارغة، خاصة خلال فترات الذروة، مما يضطرهم لدوامات بحث مرهقة تستهلك الوقود والوقت وتزيد الازدحام. كما تفرض عمليات الدفع التقليدية (نقداً أو بطاقات) انتظاراً طويلاً عند المخارج، مع افتقار المشتركين تحديداً لمنصة سهلة لتحديث بياناتهم أو تجديد اشتراكاتهم أو مراجعة سجلات الاستخدام.

ومن ناحية أخرى، تكافح إدارات الكراجات مع محدودية الأدوات المتاحة، حيث تعجز عن مراقبة حالة الإشغال لحظياً، وتواجه صعوبات في إدارة الحجوزات والاشتراكات بشكل مرن، مع ضعف القدرة على تحليل البيانات التشغيلية (مثل أنماط الإشغال وتدفق المركبات) لاتخاذ قرارات مدروسة. هذا بالإضافة إلى التعامل البطيء مع الأعطال المفاجئة (كأعطال بوابات الدخول/الخروج) مما يزيد من التكاليف التشغيلية ويقلل رضا المستخدمين.

تتجلى المشكلة الجوهرية في غياب حل رقمي متكامل يربط بين احتياجات السائقين التشغيلية ومتطلبات الإدارة التحليلية، وهو ما يستدعي تطوير منصة ذكية تعالج هذه الفجوات عبر دمج تقنيات المراقبة الفورية، وإدارة الحجوزات الشاملة، وآليات الدفع الإلكتروني.

٣.١ - أهمية البحث:

يتمثل تطوير منصة رقمية متكاملة لإدارة حجوزات مواقف السيارات حلّاً استراتيجياً للتحديات التشغيلية والخدمية المزمنة في المنظومة الحضرية. فمن الناحية الوظيفية، تُعدُّ هذه المنصة جسراً رقمياً يربط بين احتياجات المستخدمين ومتطلبات إدارة الكراجات، عبر آلية موحدة تحقق ثلاثة أهداف جوهرية وهي:

أولاً: تحويل تجربة المستخدم من نموذج البحث العشوائي المكلف إلى نظام قائم على اليقين والشفافية، حيث يمكن الحجز المسبق والدفع الإلكتروني من توفير الوقت والجهد، ويقلل الازدحام المرتبط بالبحث عن المواقف أو انتظار الدفع.

ثانياً: تمكين الإدارات التشغيلية من خلال توفير أدوات مراقبة آنية لمعدلات الإشغال، وإدارة الحجوزات والاشتراكات آلياً، مما يعزّز الكفاءة التشغيلية ويقلل الهدر المادي.

ثالثاً: الإسهام في بناء المدن الذكية عبر دمج التقنيات، حيث تُسهم المنصة في تخفيف الازدحام المروري، وخفض الانبعاثات الكربونية الناتجة عن دوران المركبات بحثاً عن مواقف، ورفع جودة الحياة الحضرية وفق معايير الاستدامة.

هذا التحول الرقمي ليس رفاهية تقنية، بل ضرورة عصرية تُواكب التوسع الحضري المتسارع، وتحوّل الكراجات من مجرد مساحات خرسانية إلى أصول ذكية قادرة على تحقيق عوائد اقتصادية أعلى، مع ضمان تجربة مستخدمٍ سلسةً ثُلبيّ توقعات العصر الرقمي.

٤.١ - أهداف البحث:

إن إنشاء منصة رقمية لإدارة مواقف السيارات يتطلب تصميماً متكاملاً يجمع بين سهولة الاستخدام والكفاءة التشغيلية. تهدف هذه المنصة إلى تيسير عملية البحث عن المواقف الشاغرة، حجزها، ودفع الرسوم إلكترونياً، مما يسهم في تحسين تجربة السائقين. بالإضافة إلى ذلك، يجب أن يتضمن النظام آليات فعالة لإدارة الاشتراكات الشهرية والسنوية، مما يعزز من استدامة العمليات المالية.

تتطلب المنصة أيضاً توفير لوحة تحكم شاملة لموظفي ومديري الكراج، تتيح لهم مراقبة الإشغال وإدارة العمليات اليومية، بما في ذلك تحديد الأسعار وإدارة الاشتراكات. من الضروري تمكين المديرين من استعراض تقارير مفصلة حول الإشغال والاشتراكات وأنماط الاستخدام، مما يساعد في اتخاذ قرارات مستنيرة.

تعتبر تقنيات مثل مسح رمز QR ضرورية لتسهيل عملية دخول وخروج المركبات، حيث تتيح هذه التقنية معالجة البيانات بسرعة وكفاءة. كما يجب تضمين آليات لمعالجة المشاكل الشائعة مثل فقدان رمز QR أو مشاكل الدفع، لضمان تجربة سلسة للمستخدمين.

ومن المهم إرسال تنبيهات تلقائية للسائقين قبل انتهاء وقت الحجز أو الاشتراك، وللموظفين حول المشكلات التشغيلية، مما يعزز من فعالية النظام ويقلل من الفوضى المحتملة.

وبذلك تتمثل الأهداف الرئيسية للمنصة:

- إنشاء منصة رقمية سهلة الاستخدام للسائقين للبحث عن المواقف الشاغرة وحجزها ودفع الرسوم إلكترونياً (بالساعة).
- تطوير نظام فعال لإدارة الاشتراكات الشهرية والسنوية.

- توفير لوحة تحكم شاملة لموظفي ومديري الكراج لمراقبة الإشغال، وإدارة العمليات اليومية، وتحديد الأسعار، وإدارة الاشتراكات.
- تمكين المديرين من استعراض تقارير مفصلة حول الإشغال والاشتراكات وأنماط الاستخدام.
- تسهيل عملية دخول وخروج المركبات باستخدام تقنيات مثل مسح رمز QR .
- تضمين آليات لمعالجة المشاكل الشائعة مثل فقدان رمز QR أو مشاكل الدفع.
- إرسال تنبيهات تلقائية للسائقين (قبل انتهاء وقت الحجز/الاشراك) وللموظفين (حول المشكلات التشغيلية).

١.٥ - النتائج التطبيقية والجهات المستفيدة منها:

١.٥.١ - النتائج التطبيقية:

تمثل النتائج التطبيقية في تطوير نظام شامل ومتوازن لإدارة الكراجات، يهدف إلى تحسين الكفاءة التشغيلية وتقديم تجربة متميزة للعملاء. هذه النتائج تشمل:

- منصة ويب للسائقين: تتيح هذا المنصة للسائقين الوصول إلى معلومات لحظية حول توفر المواقف، الحجز، والدفع. من خلال هذه المنصة، يمكن للسائقين اتخاذ قرارات سريعة ومستقرة بشأن استخدام خدمات الكراج، مما يعزز من سهولة الاستخدام ويقلل من الوقت المستغرق في البحث عن مواقف مناسبة.
- نظام إدارة اشتراكات آلي وسهل: يُعد هذا النظام أداة أساسية لتبسيط عملية الاشتراك وتحديدها، حيث يوفرواجهة سهلة الاستخدام تتيح للعملاء إدارة اشتراكاتهم بكفاءة. كما يساهم في تقليل الأخطاء اليدوية وزيادة دقة العمليات، مما يعزز من رضا العملاء ويساهم في استقرارية الإيرادات.
- لوحة تحكم ويب قوية للموظفين والمديرين: توفر لوحة التحكم واجهة متقدمة لإدارة العمليات اليومية، بما في ذلك مراقبة الأداء، إدارة الحجوزات، وتصنيص الموارد. هذه اللوحة تُمكّن الموظفين والمديرين من اتخاذ قرارات استراتيجية بناءً على بيانات دقيقة وتحليلات شاملة.
- نظام تقارير وتحليلات لدعم اتخاذ القرار: يُعد نظام التقارير والتحليلات أداة حيوية لفهم أداء الكراجات وتحديد المجالات التي تحتاج إلى تحسين. من خلال تحليل البيانات، يمكن

للمديرين تحسين استراتيجيات التشغيل وزيادة الكفاءة، مما يساهم في تحقيق أهداف المؤسسة على المدى الطويل.

- تحسين تجربة العملاء وزيادة كفاءة تشغيل الكراج: يهدف النظام إلى تقديم تجربة سلسة ومرحية للعملاء، بدءاً من الحجز وحتى الدفع. كما يساهم في تحسين العمليات التشغيلية وتقليل التكاليف، مما يعزز من تنافسية الكراجات في السوق.

٢.٥.١ - الجهات المستفيدة:

تستفيد عدة جهات من تطبيق هذا النظام، حيث يساهم في تحسين العمليات وتقديم خدمات متميزة لكل من:

- العملاء (السائقون): يشمل ذلك الزوار المؤقتين الذين يبحثون عن موقف بالساعة، بالإضافة إلى المشتركين الذين يعتمدون على خدمات الكراج بشكل منتظم. يوفر النظام لهؤلاء العملاء تجربة مرحة وسهلة الاستخدام، مما يعزز من رضاهم ويشجعهم على الاستمرار في استخدام الخدمة.
- موظفو خدمة العملاء: يساهم النظام في تسهيل إدارة المواعيد والحجوزات، مما يقلل من الضغط على الموظفين ويتيح لهم التركيز على تقديم خدمة عالية الجودة. كما يوفر أدوات متقدمة للتعامل مع استفسارات العملاء وحل المشكلات بسرعة وكفاءة.
- مدورو الكراجات: يمكن النظام للمديرين من الإشراف على العمليات اليومية، تحليل الأداء، واتخاذ قرارات استراتيجية بناءً على بيانات دقيقة. كما يساعدهم في تحسين استراتيجيات التسويق، إدارة الموارد، وزيادة الإيرادات، مما يعزز من نجاح الكراجات على المدى الطويل.

الفصل الثاني: التقنيات والبرمجيات المستخدمة ومنهجياتها

١.٢ - مقدمة:

التقنيات المستخدمة لإنجاز المشروع:

١. [12] QR Code : مكتبة لتوليد وقراءة رمز QR .

٢. [13] Gmail : لإرسال email .

٣. [14] HTML, CSS, JavaScript, React js : Front-End .

٤. [14] Node.js, Express.js : Back-End .

وهنا سنذكر شرح موسع لكل تقنية ماهي وكيف تعمل ولماذا تستخدم وما هي ميزاتها:

٢.١ - لغة توصيف صفحات الويب HTML

٢.١.٢ - ما هي لغة HTML؟

تعرف لغة HTML (HyperText Markup Language) بأنها لغة ترميز قياسية ومفتوحة المصدر، تُستخدم لإنشاء وتنظيم محتوى صفحات الويب. على عكس لغات البرمجة التقليدية، فإن HTML ليست لغة برمجة، بل هي لغة وصفية (Markup Language) تحدد بنية المستندات المعروضة على الويب. تُعتبر بمثابة الهيكل العظمي لأي موقع إلكتروني، حيث تحدد بنية المحتوى وتفصل بين عناصره المختلفة مثل النصوص، الصور، الروابط، والجداول، مما يسهل على المتصفحات فهم المحتوى وتحليله.

تأسست HTML في أوائل التسعينيات من القرن الماضي كجزء من تطور الشبكة العالمية، وقد تطورت عبر العديد من الإصدارات لتلبية المتطلبات المتزايدة لтехнологيا الويب. النسخة الأحدث، HTML5، أصبحت داعمة لميزات متقدمة تشمل التفاعلية، الوسائل المتعددة، وتطبيقات الويب الغنية، مما يعكس تطورها المستمر لتلائم احتياجات العصر الرقمي.

٢.٢.٢ - كيف تعمل لغة HTML؟

تعمل لغة HTML بشكل علمي يعتمد على مبدأ "الترميز" أو "التوصيف"، حيث تُستخدم لإنشاء هيكل منطقي للمحتوى الرقمي. عندما يقوم المطور بإنشاء صفحة ويب، يستخدم عناصر HTML التي تُعرف بـ "الوسوم" (Tags). هذه الوسوم هي عبارة عن علامات تُحاط بها الكلمات أو العبارات لتوجيه المتصفح حول كيفية عرض المحتوى. على سبيل المثال، يشير وسم <p> إلى بداية فقرة نصية، بينما يشير وسم <h1> إلى عنوان رئيسي.

عندما يطلب المستخدم صفحة ويب، يقوم الخادم بإرسال ملف HTML إلى المتصفح. يقوم المتصفح بعد ذلك بقراءة هذه الأكواد وتحليلها، ثم يفسر الوسوم والعناصر لتحديد كيفية تنظيم وعرض المحتوى. هذه العملية تحول التعليمات المكتوبة في كود HTML إلى واجهة مరئية يمكن للمستخدم رؤيتها والتفاعل معها. من الناحية العلمية، تعتمد HTML على معايير وضعتها منظمات مثل (W3C) World Wide Web Consortium، والتي تضمن أن العناصر تكون متوافقة عبر الأجهزة والمتصفحات، مما يقلل من الأخطاء ويسهل كفاءة الوصول إلى المحتوى.

لا تقوم HTML بإنشاء وظائف ديناميكية بحد ذاتها، بل تُستخدم بالتكامل مع لغات أخرى. تُكمل غالباً بلغات مثل (CSS) Cascading Style Sheets لتنسيق المظهر الجمالي للصفحة وإضافة الألوان والخطوط والتصميمات، وجافا سكريبت (JavaScript) لإضافة التفاعلية والحركة والوظائف الديناميكية. تعمل HTML كخطوة هيكلية، تحدد أماكن العناصر وترتيبها، وتتوفر إطاراً منظماً للموقع الذي يتم عرضه للمستخدم.

دورة عمل صفحة الويب مع HTML: لفهم أعمق لكيفية عمل HTML، يمكن تصور العملية على النحو التالي:

- إنشاء المستند: يقوم المطور بكتابة محتوى الصفحة وتحديده باستخدام وسوم HTML في ملف نصي (عادةً بملحق .html).
- طلب الصفحة: عندما يكتب المستخدم عنوان URL لصفحة ويب في متصفحه، يُرسل المتصفح طلباً إلى الخادم الذي يستضيف الموقع.
- الاستجابة من الخادم: يستجيب الخادم بإرسال ملف HTML إلى المتصفح.
- تحليل المتصفح: يقرأ المتصفح ملف HTML سطراً بسطر، ويقوم ببناء "نموذج كائن المستند" (DOM – Document Object Model) الذي يمثل البنية الهرمية للصفحة.
- العرض والتكامل: يقوم المتصفح بعرض المحتوى المرئي بناءً على تعليمات HTML، ويطبق التنسيقات من CSS، ويقوم بتنفيذ السكريبت من JavaScript لإنشاء تجربة تفاعلية وغنية للمستخدم.

٣.٢.٢ - لماذا تستخدم لغة HTML؟

تُستخدم HTML بشكل أساسي لأغراض متعددة في تطوير الويب، مما يجعلها ضرورية للوجود الرقمي. أهم هذه الأسباب تشمل:

- هيكلة المحتوى وتنظيمه: تُمكن HTML من هيكلة المحتوى وتحديد عناصره الأساسية بطريقة منطقية، مما يسهل على المتصفحات ومحركات البحث فهم محتوى الصفحة وفهرسته بشكل فعال. هذا التنظيم ضروري لتحسين محركات البحث (SEO)، حيث يساعد في ترتيب الموقع في نتائج البحث وزيادة وضوحتها.
- تيسير التصفح والوصول: تساهم HTML في تسهيل عملية تصفح الإنترنت عبر إنشاء الروابط التشعبية (Hyperlinks) التي تسمح بالانتقال السلس بين الصفحات والمواقع المختلفة. هذه الروابط هي جوهر الشبكة العنكبوتية وتُمكن المستخدمين من استكشاف كم هائل من المعلومات.
- تضمين الوسائط المتعددة: تُستخدم لغة HTML لتضمين مختلف أنواع الوسائط المتعددة مثل الصور، مقاطع الفيديو، الملفات الصوتية، والنماذج التفاعلية. هذا يسمح بإنشاء صفحات ويب غنية وجذابة بصرياً ومسموعاً، مما يعزز تجربة المستخدم.
- دعم التفاعلية والتطبيقات: بالتكامل مع CSS وJavaScript، توفر HTML أساساً قوياً لبناء واجهات تفاعلية وتطبيقات ويب متقدمة. هذا يسمح للمطوريين بإنشاء تجارب مستخدم ديناميكية تتكيف مع تفاعلاتهم واحتياجاتهم.

٤.٢.٢ - ماهي المزايا الأساسية لغة HTML؟

تمتاز لغة HTML بعدد من المزايا الأساسية التي تجعلها خياراً مثالياً في تطوير الويب، وهي عوامل رئيسية في انتشارها وتطورها:

- سهولة التعلم والاستخدام: تُعتبر HTML لغة بسيطة وسهلة التعلم، خاصة للمبتدئين في مجال تطوير الويب. لا تتطلب معرفة برمجية متقدمة، ويمكن تحريرها باستخدام محررات نصية بسيطة، مما يجعلها متوافقة للمبتدئين والمحترفين على حد سواء. هذه البساطة هي نقطة انطلاق مثالية لأي شخص يرغب في الدخول إلى عالم تطوير الويب.
- التوافق الواسع مع المتصفحات: تدعم HTML جميع المتصفحات الشائعة (مثل Chrome, Firefox, Safari, Edge) وما يضمن عرض المحتوى بشكل موحد عبر الأجهزة والمنصات المختلفة. هذا التوافق المعياري يقلل من مشكلات العرض ويضمن تجربة متسقة للمستخدمين بغض النظر عن المتصفح الذي يستخدمونه.

- دعم تحسين محركات البحث (SEO): تساعد HTML في تنظيم المحتوى بشكل يعزز التهيئة لمحركات البحث. من خلال استخدام الوسوم الدلالية الصحيحة (Semantic HTML)، يمكن لمحركات البحث فهم محتوى الصفحة بشكل أفضل وتصنيفه بدقة، مما يحسن من ظهور الموقع ويزيد من تفاعل المستخدمين. هذا الجانب الاستراتيجي يجعل HTML أداة حاسمة في التسويق الرقمي.

- المرونة والقدرة على التوسيع: تتيح HTML دمج تقنيات أخرى لإنشاء موقع ديناميكية وتطبيقات ويب متقدمة. يمكنها التوسيع عبر التكامل مع CSS للتصميم الجمالي، وJavaScript للوظائف التفاعلية. هذا التكامل المرن يسمح للمطوريين ببناء جميع أنواع الموقع، من البسيطة إلى المعقدة، ويدعم الابتكار المستمر في تطوير الويب.

• CSS و HTML: البنية والمظهر

بينما تحدد HTML بنية المحتوى (مثل الفقرات، العناوين، والصور)، فإن CSS (Cascading Style Sheets) هي المسئولة عن المظهر الجمالي والتصميم البصري للصفحة. تسمح CSS للمطوريين بالتحكم في الألوان، الخطوط، التباعد، تخطيط العناصر، وحتى الاستجابة لأحجام الشاشات المختلفة. بدون CSS، ستظهر صفحات الويب كوثائق نصية بسيطة بدون أي جاذبية بصرية.

• JavaScript و HTML: البنية والتفاعل

JavaScript هي لغة البرمجة التي تُضفي الحياة على صفحات الويب، حيث تمكّنها من الاستجابة لتفاعلات المستخدمين، تحديث المحتوى ديناميكياً، وإضافة وظائف معقّدة مثل النماذج التفاعلية، الرسوم المتحركة، والألعاب. HTML توفر العناصر التي يمكن لـ JavaScript التفاعل معها وتعديلها. هذا التأثير يسمح بإنشاء تطبيقات ويب كاملة الوظائف وتجارب مستخدم غنية.

• SEO و HTML: البنية والرؤية

تلعب HTML دوراً حاسماً في تحسين محركات البحث (SEO). من خلال استخدام الوسوم الدلالية (Semantic Tags) مثل <header>, <nav>, <article>, و <footer>, تساعد HTML محركات البحث على فهم سياق وهيكل المحتوى بشكل أفضل. هذا الفهم يُترجم إلى تصنيف أعلى في نتائج البحث، مما يزيد من وصول الجمهور للموقع.

٣-٣- لغة تنسيق وتصميم الواجهات CSS

١.٣.٢ - ما هي CSS؟

المعروفة اختصاراً بـ CSS (Cascading Style Sheets)، إنها لغة توصيفية متخصصة تُستخدم لتحديد المظهر البصري للمستندات المكتوبة بلغات ترميزية مثل HTML أو XML. الهدف الأساسي لـ CSS هو فصل المحتوى عن طريقة عرضه، مما يعني أنها تُعنى بتحديد أسلوب تصميم الصفحات من حيث الألوان، الخطوط، التخطيطات، المسافات، الحدود، وتأثيرات الحركة، دون أن تؤثر على محتوى الصفحة نفسه أو هيكلها المنطقي.

٢.٣.٢ - كيف تعمل CSS؟

تعمل CSS من خلال تطبيق "قواعد" على عناصر HTML. تتكون كل قاعدة من مُحدّد selector (يحدد العنصر أو العناصر التي سيتم تطبيق التنسيق عليها، ثم مجموعة من الخصائص properties) والقيم values (التي تحدد شكل هذا العنصر. على سبيل المثال، يمكن لمُحدّد مثل p (الذي يشير إلى فقرة نصية) أن يكون لديه خاصية color بقيمة blue، مما يجعل جميع الفقرات النصية تظهر باللون الأزرق. يمكن أيضاً ربط ملفات CSS بصفحات HTML لتوحيد التصميم عبر موقع ويب بأكمله، مما يسهل إدارة التحديثات).

٣.٣.٢ - لماذا تُستخدم CSS؟

تستخدم CSS بشكل رئيسي لإضفاء الطابع الجمالي والوظيفي على صفحات الويب، إلى جانب تحسين تجربة المستخدم ورفع مستوى التفاعلية والوضوح. فهي تُسهل على المطورين فصل المحتوى عن العرض، مما يُسهم في تقليل مجهود التحديث والتعديل على التصميم، إذ يمكن تعديل الشكل العام للموقع عبر تعديل ملف CSS واحد، بدلاً من تعديل كل صفحة برمجية بشكل فردي. كما تساهم CSS في جعل الموقع أكثر استجابة وتوافقاً مع مختلف أنواع الأجهزة، عبر تصميم يُعرف بالتصميم المرن أو التكيفي (Responsive Design)، بحيث يظهر بشكل متناسق سواء كان المشاهد يستخدم حاسوباً أو هاتف ذكيًّا أو جهاز لوحي. بالإضافة إلى ذلك، فإن CSS تتيح إمكانية إدراج أنماط وتنسيقات متقدمة، مثل تأثيرات الحركة، والانتقالات، والتشويش، والتحكم في أنماط الطباعة، مما يعزز من فعالية الواجهات البصرية للموقع.

٤.٣.٢ - ماهي المزايا الأساسية لـ CSS؟

تتمتع CSS بعدة مزايا أساسية تجعلها أداة حيوية في تصميم وتطوير الويب:

- فصل المحتوى عن التصميم: هذه الميزة هي الأكثر أهمية، حيث تُمكن المطوريين من إدارة المحتوى والتصميم بشكل مستقل تماماً، مما يُسرّع من عمليات التحديث والصيانة.
 - تحسين سرعة تحميل الصفحات: بما أن ملفات CSS يمكن ربطها خارجياً وتتخزينها مؤقتاً، فإنها تُحمل مرة واحدة وتُطبق على صفحات متعددة، مما يقلل من حجم البيانات المنقولة ويُسرّع من عرض الصفحات.
 - تقليل ازدواجية العمل وتوفير الوقت: بدلاً من تكرار قواعد التنسيق في كل صفحة HTML، يمكن تعديل تصميم الموقع بالكامل عبر ملف CSS واحد، مما يوفر جهداً ووقتاً كبيرين.
 - دعم التوافق مع الأجهزة والشاشات المختلفة: بفضل قدرتها على إنشاء تصميمات متباينة، تضمن CSS عرضاً مثالياً للمحتوى على أي جهاز، مما يعزز تجربة المستخدم.
 - توفير بنية نظيفة وواضحة لصفحات الويب: يجعل فصل المحتوى عن التصميم كود HTML أبسط وأكثر تنظيماً، مما يسهل على المطوريين والبرامج المساعدة (مثل قارئات الشاشة) العمل معه.
 - مرونة التصميم: تتيح CSS للمطوريين نطاقاً واسعاً من خيارات التنسيق، بما في ذلك التحكم الدقيق في الألوان، الخطوط، التباعد، الظل، والانتقالات، مما يُمكن من إنشاء تصميمات معقدة وجذابة.
 - إمكانية تطبيق تأثيرات جمالية متقدمة: تُمكن CSS من إضافة تأثيرات مثل الرسوم المتحركة، والانتقالات السلسة، والظل، والتدرجات اللونية، مما يعزز جاذبية الموقع.
- ## ٤- مكتبة تصميم واجهات جاهزة Tailwind CSS
- ### ١٤٢- ما هو Tailwind CSS؟
- يعد Tailwind CSS إطار عمل CSS مفتوح المصدر يتبني منهجية "utility-first" ، وهي مقاربة حديثة في تصميم الويب تهدف إلى تسريع عملية بناء واجهات المستخدم بشكل كبير. على عكس أطر العمل التقليدية التي قد توفر مكونات جاهزة مع أنماط مسبقة (مثل الأزرار أو البطاقات)، يقدم Tailwind CSS مجموعة واسعة من "فئات الأدوات" (utility classes). كل فئة من هذه الفئات تُطبق خاصية CSS محددة ومفردة، مما يمنح المطوريين تحكمًا دقيقًا في مظهر كل عنصر في صفحة الويب.

تتمثل الفكرة الجوهرية لـ Tailwind CSS في السماح للمطوريين ببناء تصاميم معقدة وفريدة من خلال تركيب هذه الفئات الصغيرة والموجهة مباشرة داخل ترميز HTML. هذا يعني أن المطوريين يمكنهم تعديل الهوامش، الألوان، أحجام الخطوط، الظلل، والتجاوب، وغيرها، دون الحاجة إلى كتابة قواعد CSS مخصصة ومنفصلة لكل تغيير. هذا النهج يقلل بشكل كبير من الحاجة للتغلب بين ملفات HTML وCSS، ويزيد من سرعة التطوير ومرنة التصميم.

٢٠٤.٢ - لماذا يستخدم Tailwind CSS؟

يُستخدم Tailwind CSS على نطاق واسع في تطوير الويب الحديث لعدة أسباب جوهرية تساهم في تحسين عملية الإنتاجية والجودة حيث:

- يسرّع بشكل كبير من عملية تطوير واجهات المستخدم. فبدلاً من قضاء الوقت في ابتكار أسماء فئات CSS فريدة لكل مكون أو خاصية، يمكن للمطوريين ببساطة تطبيق الفئات الجاهزة التي توفرها Tailwind.
- يقلل من "مشكلة تسمية الفئات" (naming conventions) التي غالباً ما تعيق سير عمل المطوريين في مشاريع CSS التقليدية الكبيرة.
- يوفر مرنة لا مثيل لها. نظراً لأن التصميم يتم بناؤه من خلال تركيب فئات صغيرة، يمكن للمطوريين إنشاء أي تصميم تقريباً دون الحاجة إلى تجاوز أو تعديل الكثير من CSS، مما يمنحهم سيطرة كاملة على الجمالية النهائية.
- يعزز الاتساق في التصميم عبر المشروع. على الرغم من أن Tailwind يمنح المرنة، إلا أنه يشجع على استخدام مجموعة محددة من قيم التصميم (design tokens) مثل مقاييس التباعد أو لوحة الألوان، مما يساعد في الحفاظ على هوية بصرية موحدة.
- وأخيراً، يؤدي التحسين التلقائي لملفات CSS في بيئه الإنتاج إلى أداء أفضل للموقع، حيث يتم تحميل ملفات CSS أصغر حجماً..

٣٠٤.٢ - ما هي المميزات الأساسية لـ Tailwind CSS؟

يمكن تلخيص المزايا الأساسية لـ Tailwind CSS في النقاط التالية:

- سرعة التطوير الفائقة: بناء واجهات مستخدم معقدة في وقت قياسي بفضل فئات الأدوات الجاهزة.
- مرنة وتصصيص لا حدود لهما: القدرة على إنشاء تصاميم فريدة ومعقدة من خلال تركيب الفئات، مع إمكانية تخصيص الإعدادات الأساسية بالكامل.

- دعم متقدم للتصميم المتجاوب: يسهل إنشاء تصميمات تتكيف بسلاسة مع مختلف أحجام الشاشات والأجهزة بفضل فئات التحاوب المدمجة.
- تحسين الأداء: ملفات CSS صغيرة ومحسنة للإنتاج بفضل عملية إزالة CSS غير المستخدم تلقائياً.
- صيانة أسهل للكود: تقليل التعقيد في إدارة الكود CSS، حيث تكون الأنماط مرتبطة مباشرة بالعناصر في HTML.
- تجنب اصطدامات التسمية: لا حاجة لابتکار أسماء فئات CSS معقدة أو القلق بشأن تداخل الأنماط.

٤.٤.٢ - كيف يعمل Tailwind CSS؟

يعمل Tailwind CSS عن طريق تطبيق هذه الفئات مباشرة على عناصر HTML. على سبيل المثال، لتطبيق هامش أيسير بعرض معين، يمكن استخدام فئة مثل `ml-4`. لتحديد لون خلفية، تُستخدم فئة مثل `bg-blue-500`. هذه المنهجية تسمح للمطورين بالبقاء داخل ملف HTML الخاص بهم أثناء عملية التصميم، مما يقلل من الحاجة إلى التنقل بين ملفات HTML وملفات CSS منفصلة. يدعم Tailwind CSS أيضًا التصميم المتجاوب (*responsive design*) من خلال فئات معدلة يمكن تطبيقها بناءً على أحجام الشاشات المختلفة، مثل `md:ml-8` لتطبيق هامش أيسير معين على الشاشات المتوسطة وما فوق. كما أن الإطار قابل للتخصيص بدرجة عالية من خلال ملف التكوين الخاص به، مما يسمح للمطورين بتعديل أو توسيع إعدادات التصميم الافتراضية لتناسب احتياجات مشاريعهم الفريدة.

٤.٥ - لغة برمجة تفاعلية للويب JavaScript

٤.٥.١ - ما هي JavaScript؟

JavaScript هي لغة برمجة ديناميكية تُستخدم بشكل رئيسي لتطوير تطبيقات الويب التفاعلية. تم تطويرها لأول مرة في عام ١٩٩٥ بواسطة Brendan Eich أثناء عمله في شركة Netscape، ومنذ ذلك الحين أصبحت واحدة من أكثر لغات البرمجة شيوعاً في العالم. تُعتبر JavaScript لغة نصية (*Scripting Language*) تُنفذ على جانب العميل (*Client-Side*)، مما يعني أنها تعمل مباشرة داخل متصفح المستخدم. ومع ذلك، يمكن استخدامها أيضًا على جانب الخادم (*Server-Side*) باستخدام منصات مثل `Node.js`.

JavaScript ليست مرتبطة فقط بتطوير الويب، بل تُستخدم أيضًا في تطوير تطبيقات الهواتف الذكية، الألعاب، وحتى تطبيقات سطح المكتب. تُعتبر JavaScript جزءاً أساسياً من ثلاثة تقنيات تطوير الويب: HTML، CSS، JavaScript، حيث تُستخدم HTML لتحديد هيكل الصفحة، وCSS لتنسيقها، وJavaScript لإضافة التفاعلية.

٢.٥.٢ - كيف تعمل JavaScript؟

JavaScript تعمل من خلال تنفيذ التعليمات البرمجية داخل متصفح المستخدم أو على الخادم. عند زيارة المستخدم لموقع ويب يحتوي على JavaScript، يقوم المتصفح بتحميل صفحة HTML وCSS، ثم يبدأ في تفسير وتشغيل كود JavaScript. يتم تنفيذ الكود بواسطة محرك SpiderMonkey، Google Chrome V8 في JavaScript الموجود داخل المتصفح، مثل Microsoft Edge في Chakra، أو Firefox في

تعمل JavaScript بناءً على نموذج الحدث (Event-Driven Model)، حيث تستجيب للأحداث مثل النقرات، التمرير، أو إدخال البيانات. يمكنها التفاعل مع عناصر HTML باستخدام واجهة برمجة التطبيقات (DOM) (Document Object Model)، مما يسمح لها بتغيير محتوى الصفحة أو تنسيقها دون الحاجة إلى إعادة تحميل الصفحة بالكامل.

على جانب الخادم، تعمل JavaScript باستخدام منصات مثل Node.js، حيث يتم تشغيل الكود على الخادم لمعالجة الطلبات، إدارة قواعد البيانات، وتقديم المحتوى الديناميكي للمستخدمين.

٣.٥.٢ - لماذا تُستخدم JavaScript؟

JavaScript تُستخدم لأنها تُعتبر أداة قوية ومرنة لتطوير تطبيقات الويب الحديثة. توفر اللغة إمكانيات واسعة تجعلها الخيار الأول للمطوريين في العديد من المجالات. فيما يلي بعض الأسباب الرئيسية لاستخدام JavaScript:

- التفاعلية: تُتيح JavaScript إنشاء صفحات ويب ديناميكية وتفاعلية، مما يحسن تجربة المستخدم بشكل كبير.
- التوافق: تعمل JavaScript على جميع المتصفحات الحديثة دون الحاجة إلى تثبيت إضافات أو برامج.
- الأداء: تُنفذ JavaScript مباشرة داخل المتصفح، مما يجعلها سريعة وفعالة.

- التكامل: يمكن دمج JavaScript بسهولة مع HTML و CSS، مما يجعلها أداة مثالية لتطوير الواجهات الأمامية.
- التوسيع: باستخدام منصات مثل Node.js، يمكن استخدام JavaScript لتطوير تطبيقات الخادم، مما يجعلها لغة شاملة لتطوير التطبيقات الكاملة (Full-Stack Development).

٤.٥.٢ - ما هي المزايا الأساسية لـ JavaScript؟

JavaScript تتمتع بالعديد من المزايا التي جعلتها واحدة من أكثر لغات البرمجة استخداماً في العالم. من بين هذه المزايا:

- سهولة التعلم: JavaScript تعتبر لغة سهلة التعلم للمبتدئين، حيث يمكن البدء في استخدامها بسرعة دون الحاجة إلى إعدادات معقدة.
- التنفيذ الفوري: يتم تنفيذ كود JavaScript مباشرة داخل المتصفح، مما يقلل من وقت الاستجابة ويعزز الأداء.
- التوافق عبر المنصات: JavaScript تعمل على جميع أنظمة التشغيل والمتصفحات الحديثة، مما يجعلها لغة متعددة الاستخدامات.
- مجتمع واسع: تمتلك JavaScript مجتمعاً كبيراً من المطورين، مما يعني توفر العديد من الموارد، المكتبات، والأدوات التي تسهل عملية التطوير.
- التوسيع والإمكانات: يمكن استخدام JavaScript لتطوير تطبيقات الويب، تطبيقات الهواتف الذكية، الألعاب، وحتى تطبيقات سطح المكتب.
- التكامل مع المكتبات والإطارات: توفر JavaScript العديد من المكتبات والإطارات مثل React، Angular، Vue.js، والتي تسهل تطوير التطبيقات المعقدة.

٦.٢ - مكتبة لبناء واجهات المستخدم React.js

١.٦.٢ - ما هو React.js؟

React.js، التي تُعرف اختصاراً بـ React أو ReactJS، هي مكتبة جافا سكريبت مفتوحة المصدر طورتها شركة ميتا (فيسبوك سابقاً). انطلقت هذه المكتبة في عام ٢٠١٣ لتحل ثورة في طريقة بناء واجهات المستخدم (UIs) وتطبيقات الويب التفاعلية. تميز React.js بتركيزها على تسهيل عملية إنشاء مكونات واجهة المستخدم القابلة لإعادة الاستخدام، مما يجعلها أداة

محورية لتطوير تطبيقات الويب أحادية الصفحة (Single-Page Applications) وتطبيقات الويب المعقدة التي تتطلب تفاعلية عالية.

٢.٦.٢ - لماذا يُستخدم React.js؟

تُستخدم React.js على نطاق واسع في تطوير الويب نظراً لكتفاتها وموارنتها في التعامل مع المحتوى الديناميكي والتفاعلات المعقدة. في البداية، تم تصميمها لتلبية الاحتياجات

المتزايدة للتطبيقات الحديثة التي تعتمد على تبادل البيانات المستمر مع خوادم الواجهة الخلفية (Back-end APIs)، مثل منصات التواصل الاجتماعي أو تطبيقات التجارة الإلكترونية. بدلًا من إعادة كتابة الكود لكل تفاعل أو تحديث، تقدم React.js إطاراً يسمح بإدارة هذه التفاعلات بطريقة منظمة وفعالة، مما يقلل من وقت التطوير ويسهل من جودة المنتج النهائي.

من منظور أكاديمي وعملي، تُستخدم React.js لأنها تلبي احتياجات البرمجة الحديثة في بناء تطبيقات قابلة للتوسع والصيانة. تتيح مرونتها سهولة دمجها مع تقنيات أخرى مثل Redux لإدارة الحالة المعقدة أو Next.js لتقديم التطبيقات من جانب الخادم (Server-Side Rendering)، مما يعزز من إمكانيات تحسين محركات البحث (SEO). هذا الاستخدام يعود أيضًا إلى دعمها القوي من مجتمع المطوريين، الذي يوفر موارد وأدوات مفتوحة المصدر، مما يجعلها خيارًا مثالياً للمشاريع التجارية والأكاديمية على حد سواء.

٣.٦.٢ - ماهي المميزات الأساسية لـ React.js؟

تمتاز React.js بعدة مزايا أساسية تجعلها خيارًا مفضلاً في تطوير الويب الحديث، وهي:

- كفاءة الأداء والسرعة: بفضل تقنية Virtual DOM، تضمن React.js تحديثات سريعة وفعالة لواجهة المستخدم. يتم تحديد التغييرات الضرورية فقط وتطبيقها على الـ DOM الفعلي، مما يقلل من عمليات إعادة الرسم المكلفة للمتصفح ويسهل من سرعة الأداء العام للتطبيق، خاصة مع المحتوى الديناميكي والتفاعلات المتكررة.

- بنية المكونات وإعادة الاستخدام: تشجع React.js على استخدام بنية المكونات (Component-Based Architecture)، حيث يمكن استخدام المكونات في أماكن مختلفة من التطبيق أو حتى في مشاريع متعددة دون الحاجة لإعادة كتابة الشيفرة. هذا يعزز الإنتاجية، يقلل من الأخطاء، ويجعل الكود أكثر تنظيماً وقابلية للإدارة.

- تصميم تصريحي (Declarative Design): تبني React.js مبدأ البرمجة التصريحية، حيث يصف المطور ما يجب أن تظهره الواجهة بدلاً من تحديد خطوات كيفية التغيير أو

التحديث. هذا يجعل الكود أكثر وضوحاً وسهولة في الفهم والتطوير والصيانة، ويسهل تتبع تدفق البيانات.

- مرونة وقابلية للتوسيع: React.js هي مكتبة وليس إطار عمل شامل (framework)، مما يمنح المطورين حرية أكبر في اختيار الأدوات والمكتبات الأخرى التي تتكامل معها (مثل مكتبات التوجيه أو إدارة الحالة). هذه المرونة تجعلها مناسبة لبناء تطبيقات قابلة للتوسيع بسهولة لتلبية الاحتياجات المستقبلية.

- مجتمع كبير ودعم مستمر: تحظى React.js بدعم قوي من شركة ميتا ومجتمع عالمي واسع من المطورين. هذا المجتمع يوفر موارد تعليمية غنية، أدوات مساعدة، ومكتبات جاهزة، مما يسهل عملية التعلم والتعامل مع التحديات التي قد تواجه المطورين.

٤.٦.٢ - كيف يعمل React.js؟

تعمل React.js على مبدأ محوري يُعرف باسم المكونات القابلة لإعادة الاستخدام (Reusable Components). تُقسم واجهة المستخدم إلى أجزاء أصغر ومنظمة تُسمى "المكونات"، حيث يمثل كل مكون جزءاً مستقلاً من الواجهة، مثل زر أو نموذج أو قائمة. يمكن دمج هذه المكونات معاً لبناء الواجهة الكاملة، وكل مكون قادر على إدارة حاليه وبياناته بشكل مستقل.

الابتكار الأساسي في آلية عمل React.js يكمن في استخدامها لتقنية الا DOM (نماذج كائن المستند الافتراضي). بدلاً من التعامل مباشرة مع الا DOM الفعلي للمتصفح، تحفظ React بنسخة افتراضية خفيفة من الا DOM في الذاكرة. عندما تحدث تغييرات في حالة التطبيق أو تفاعلات المستخدم، تقوم React بإنشاء Virtual DOM جديد ثم تُجري مقارنة دقيقة بينه وبين النسخة السابقة لتحديد الاختلافات الضرورية فقط. بعد تحديد هذه التغييرات، تقوم React بتحديث الجزء المتأثر فقط في الا DOM الفعلي للمتصفح، دون الحاجة لإعادة تحميل الصفحة بأكملها. هذا النهج يضمن كفاءة عالية في الأداء، حيث يقلل بشكل كبير من العبء على المتصفح ويسرع من عملية التحديثات، مما يؤدي إلى تجربة مستخدم سلسة وسريعة الاستجابة.

بالإضافة إلى ذلك، تستخدم React ما يُعرف بـ JSX (JavaScript XML)، وهو امتداد لبناء الجملة في JavaScript يسمح للمطورين بكتابة كود يشبه HTML داخل ملفات JavaScript. يسهل JSX قراءة الكود وصيانته، حيث يتم تحويله في النهاية إلى استدعاءات JavaScript قياسية بواسطة أدوات مثل Babel. هذا التوليف بين المكونات، Virtual DOM، وJSX يجعل عملية تطوير واجهات المستخدم المعقدة أكثر تبسيطًا وتنظيمًا.

٢.٧.١ - بيئة تشغيل الخادم Node.js

٢.٧.٢ - ما هي لغة Node.js؟

تعد Node.js بيئة تشغيل مفتوحة المصدر ومتحدة المنصات، تمكّن مطوري الويب من استخدام لغة JavaScript ليس فقط في واجهة المستخدم الأمامية (frontend)، بل أيضًا في بناء تطبيقات (backend)، وأدوات سطر الأوامر، وتطبيقات الويب واسعة النطاق. تم بناء Node.js على محرك V8 الخاص بجوجل كروم، وهو محرك JavaScript عالي الأداء يقوم بتحويل كود JavaScript مباشرة إلى كود آلي فعال، مما يساهم في كفاءتها وسرعتها الفائقة في تنفيذ الأكواد.

٢.٧.٣ - كيف تعمل لغة Node.js؟

تعتمد Node.js على نموذج برمجي فريد من نوعه، وهو النموذج غير المتزامن المدفوع بالأحداث (I/O Event-driven, Non-blocking). هذا النموذج يسمح لـ Node.js بمعالجة العديد من المهام في وقت واحد دون الحاجة إلى انتظار اكتمال كل مهمة على حدة، على عكس النماذج التقليدية التي قد تتوقف فيها العمليات في انتظار استجابات الإدخال/الإخراج (I/O) مثل قراءة الملفات أو طلبات الشبكة. بدلاً من ذلك، تستخدم Node.js حلقة أحداث (Event Loop) لإدارة هذه العمليات. عندما يتم استقبال طلب جديد، تتم معالجته بشكل غير متزامن، مما يتيح للنظاممواصلة العمل واستقبال طلبات أخرى دون توقف. هذه الآلية تجعل Node.js قادرة على التعامل مع عدد ضخم من الاتصالات المتزامنة بكفاءة عالية، مما يجعلها مثالية لتطبيقات الوقت الفعلي والتطبيقات كثيفة الاستخدام للمدخلات والمخرجات.

٣.٧.٢ - لماذا تستخدم لغة Node.js؟

يُستخدم Node.js على نطاق واسع في تطوير الويب الحديث لعدة أسباب جوهرية تساهم في كفاءة وفعالية التطبيقات. من أبرز هذه الأسباب:

- توحيد لغة البرمجة: تُتيح Node.js للمطوريين استخدام لغة JavaScript لكل من الواجهة الأمامية (frontend) والواجهة الخلفية (backend) للتطبيقات. هذا التوحيد يقلل من التعقيد المعرفي، حيث لا يحتاج المطوروون إلى تعلم لغات متعددة للتعامل مع جوانب مختلفة من التطبيق، مما يسّر من دورة التطوير ويسهل على الفرق التعاون.

- الأداء العالي وقابلية التوسيع: بفضل نموذجها غير المتزامن ومحرك V8، تتميز Node.js بالسرعة والكفاءة في معالجة البيانات، خاصة في التطبيقات كثيفة الإدخال والإخراج.

تصميماً يسمح بالتوسيع الأفقي والرأسي بسهولة، مما يجعلها قادرة على دعم النمو في عدد المستخدمين والبيانات دون تأثير سلبي على الأداء.

- تطبيقات الوقت الفعلي: تُعد Node.js خياراً مثالياً لبناء تطبيقات تتطلب تحديثات فورية واستجابة سريعة، مثل غرف الدردشة الفورية، والألعاب عبر الإنترنت، وأنظمة بث المحتوى المباشر، وخدمات التعاون الجماعي. قدرتها على معالجة عدد كبير من الاتصالات المتزامنة يجعلها مثالية لهذه الأنواع من التطبيقات.
- نظام بيئي غني ومجتمع نشط: تمتلك Node.js مجتمع مطوريين نشطاً للغاية ونظاماً بيئياً واسعاً من الحزم والأدوات المترابطة عبر مدير الحزم Node Package Manager (npm). يوفر npm آلاف المكتبات والوحدات الجاهزة التي تسهل وتسرع عملية التطوير بشكل كبير.

٤.٧.٢ - ماهي المزايا الأساسية لغة Node.js؟

يمكن تلخيص المزايا الرئيسية لـ Node.js في النقاط التالية:

- السرعة والكفاءة: بفضل محرك V8 غير المتوقف، تقدم Node.js أداءً سريعاً للغاية في تنفيذ كود JavaScript ومعالجة الطلبات.
 - توحيد لغة البرمجة: استخدام JavaScript للواجهتين الأمامية والخلفية يبسط عملية التطوير ويقلل من تعقيد المكدس التكنولوجي (tech stack).
 - قابلية التوسيع العالية: قدرتها على التعامل مع عدد كبير من الاتصالات المتزامنة يجعلها مناسبة للتطبيقات التي تتطلب نمواً كبيراً.
 - نظام حزم ضخم (npm): يوفر وصولاً سهلاً إلى آلاف الوحدات والمكتبات الجاهزة، مما يسرع من عملية البناء والتطوير.
 - دعم متعدد المنصات: يمكن تشغيل تطبيقات Node.js على أنظمة تشغيل مختلفة مثل macOS و Linux و Windows.
- ٤.٨.٢ - إطار عمل لتطبيقات الويب Express.js

٤.٨.٢ - ما هو Express.js؟

Express هو إطار عمل ويب مصمم خصيصاً لبيئة تشغيل Node.js. يُعد هذا الإطار من أكثر الأدوات شيوعاً واستخداماً في بناء تطبيقات الويب والواجهات البرمجية (APIs) على

جانب الخادم. يهدف Express.js إلى تبسيط عملية تطوير الويب من خلال توفير مجموعة من الميزات الأساسية التي لا تُعَد وظائف Node.js الأصلية، مما يسمح للمطورين بالتركيز على منطق الأعمال بدلاً من التعامل مع تعقيدات بروتوكول HTTP بشكل مباشر.

نشأ Express.js كاستجابة للحاجة إلى حل منظم وفعال لبناء تطبيقات الويب في بيئة Node.js، التي كانت تتطلب في البداية كتابة الكثير من التعليمات البرمجية الأساسية يدوياً. بفضل تصميمه الخفيف والمرن، يوفر Express.js للمطورين الحرية في تخصيص هيكلة تطبيقاتهم واختيار الأدوات والمكتبات التي تناسب احتياجاتهم، مما يجعله مناسباً لمجموعة واسعة من المشاريع، بدءاً من التطبيقات الصغيرة وحتى الأنظمة المعقدة والكبيرة.

٢.٨.٢ - لماذا يستخدم Express.js؟

يستخدم Express.js بشكل أساسي لتوفير هيكل واضح ومنظم لتطبيقات الويب. بدلاً من التعامل مباشرة مع تعقيدات بروتوكول HTTP والطلبات والاستجابات على مستوى منخفض باستخدام Node.js الأساسي، تقدم Express.js واجهة أبسط وأكثر سهولة للكتابة والتنظيم. هذا يسمح للمطورين بالتركيز بشكل أكبر على منطق العمل الخاص بالتطبيق بدلاً من التفاصيل التنفيذية. الأسباب الرئيسية لاستخدامه تتضمن:

- التطوير السريع: بفضل واجهته البسيطة والموثقة جيداً، يتيح Express.js للمطورين بناء تطبيقات الويب بسرعة وكفاءة، مما يقلل من وقت الوصول إلى السوق.
- المرونة وقابلية التوسيع: كونه إطار عمل "غير رأي" (unopinionated)، يمنح Express.js المطورين حرية كبيرة في تخصيص هيكل التطبيق واختيار الأدوات والمكتبات التي يرغبون في استخدامها. يمكنه التعامل مع مجموعة واسعة من التطبيقات، من المشاريع الصغيرة إلى التطبيقات الكبيرة والمعقدة، ويمكن توسيع نطاقها بسهولة.
- نظام توجيه قوي: يوفر Express.js نظام توجيه فعال لربط مسارات URL بوظائف معالجة محددة، مما يسهل تنظيم التطبيق ويجعل إدارة الطلبات أكثر فعالية.
- دعم مجتمعي واسع: نظراً لشعبنته، يتمتع Express.js بمجتمع كبير ونشط يوفر دعماً كبيراً ومجموعة واسعة من المكتبات الإضافية (Middleware) التي يمكن استخدامها لتوسيع الوظائف.
- تقليل التكاليف: كونه مفتوح المصدر وتحت ترخيص MIT، يمكن استخدامه وتعديلاته مجاناً، مما يقلل من تكاليف الأدوات ويشجع الابتكار..

٣.٨.٢ - ما هي المميزات الأساسية لـ Express.js ؟

تمثل المزايا الأساسية لـ Express.js في عدة جوانب تجعله خياراً مفضلاً للمطوريين:

- البساطة والحد الأدنى: هو إطار عمل خفيف الوزن يوفر الأساسية فقط، مما يجعله سهل التعلم والاستخدام، ولا يفرض تعقيدات غير ضرورية.
- قابلية التخصيص العالية: يسمح للمطوريين بتخصيص كل جانب من جوانب التطبيق تقريباً، من هيكلة الملفات إلى اختيار قواعد البيانات ومحركات القوالب.
- نظام التوجيه المتسق والفعال: يوفر طريقة واضحة ومنظمة لتحديد كيفية استجابة التطبيق لطلبات HTTP المختلفة، مع دعم قوي للمعالجة غير المتزامنة التي تحسن الأداء.
- دعم البرمجيات الوسيطة الشامل: تتيح البرمجيات الوسيطة إمكانية إضافة وظائف متعددة ومعالجة الطلبات بطرق مرنة، مثل المصادقة، التحقق من الصحة، والتسجيل.
- مجاني ومفتوح المصدر: يوفر Express.js حلًا اقتصاديًا للمطوريين والشركات، حيث لا يتطلب تراخيص أو تكاليف استخدام.
- الاستخدام الفعال لـ Node.js: يبني على قدرات Node.js ويقدم واجهات برمجة تطبيقات أبسط للمهام الشائعة في تطوير الويب، مما يعزز من كفاءة التطبيق.
- تكامل قوي مع أدوات ومكتبات الطرف الثالث: يتيح الاتصال السلس مع العديد من الأدوات والمكتبات الإضافية التي توسيع قدراته وتسهل عمليات النكامل مع أنظمة أخرى.

٤.٨.٢ - كيف يعمل Express.js ؟

يعمل Express.js كطبقة تجريد فوق Node.js، مما يسهل معالجة طلبات HTTP وإرسال الاستجابات. عندما يتلقى الخادم طلباً من العميل، يقوم Express.js بتوجيه هذا الطلب إلى معالج مناسب بناءً على المسار (Route) وطريقة HTTP المستخدمة (مثل GET، POST، PUT، DELETE). يمكن للمطوريين تحديد مسارات محددة ومعالجاتها الخاصة لإنشاء منطق التطبيق.

إحدى الركائز الأساسية لعمل Express.js هي مفهوم "البرمجيات الوسيطة" (Middleware). البرمجيات الوسيطة هي وظائف تسلسل معالجة الطلب والاستجابة. يمكن استخدامها لتنفيذ مهام متعددة مثل تحليل بيانات الطلب الواردة (مثلاً JSON أو بيانات النماذج)، إدارة الجلسات، المصادقة، التقويض، وتسجيل الأخطاء. عند وصول الطلب، يمر عبر سلسلة من هذه البرمجيات الوسيطة قبل الوصول إلى المعالج النهائي، أو يمكن لأي منها إنهاء دورة الطلب والاستجابة مباشرة. هذا النهج يضمن معالجة منظمة وفعالة للطلبات ويعقل من التعقيد.

٩.٦ - نظام إدارة قواعد البيانات MySQL

١٩.٢ - ما هي لغة MySQL؟

تعرف MySQL بأنها نظام إدارة قواعد بيانات علانقية (RDBMS) مفتوح المصدر، وتعتبر واحدة من أكثر أنظمة إدارة قواعد البيانات انتشاراً واستخداماً على مستوى العالم، خاصةً في تطوير تطبيقات الويب. تم تطويرها في منتصف التسعينيات، وتتولى شركة Oracle حالياً مسؤولية تطويرها وصيانتها. تُستخدم MySQL لتخزين البيانات وتنظيمها بطريقة منهجية داخل جداول، حيث يتم تمثيل البيانات في صنوف وأعمدة ضمن مخططات تحدد العلاقات بين هذه الجداول. تعتمد MySQL على لغة SQL (لغة الاستعلام المهيكلة) لإجراء عمليات استرجاع البيانات، تحديثها، حذفها، وإدارتها بكفاءة عالية.

٢٩.٢ - كيف تعمل لغة MySQL؟

تعمل MySQL بنموذج العميل-الخادم (Client-Server Model)، وهو ما يُمثل هيكلها التشغيلي الأساسي. يتكون هذا النموذج من مكونين رئيسيين:

- خادم MySQL (MySQL Server)

هو المحرك الأساسي الذي يدير قواعد البيانات فعلياً. يستقبل الخادم الأوامر والطلبات من العملاء ويقوم بمعالجتها، مثل تخزين البيانات، استرجاعها، تحديثها، أو حذفها. يعمل الخادم بشكل مستقل وينتظر الأوامر، وهو المسؤول عن تنفيذ استعلامات SQL ومعالجة البيانات وفقاً للمخطط المحدد. يدعم الخادم آليات التحميل السريع للبيانات، حيث يمكن استرجاع ملايين السجلات في وقت قياسي بفضل تحسيناته في أداء الاستعلامات. كما يضمن الاعتمادية من خلال آليات التكرار (Replication)، التي تسمح بنسخ البيانات عبر خوادم متعددة لتجنب فقدانها.

- عميل MySQL (MySQL Client)

هو الواجهة التي يتفاعل معها المستخدمون أو التطبيقات للتواصل مع خادم MySQL. يمكن أن يكون العميل برنامجاً مستقلاً، أو تطبيق ويب، أو أي برنامج آخر يحتاج إلى الوصول إلى قاعدة البيانات. يتم التواصل بين العميل والخادم باستخدام لغة SQL (Structured Query Language)، وهي لغة قياسية تُستخدم لتعريف البيانات، معالجتها، استرجاعها، والتحكم في الوصول إليها. يُخزن MySQL البيانات في جداول منتظمة، حيث تكون كل جدول من صنوف وأعمدة، وتمكن العلاقات بين هذه الجداول من هيكلة البيانات بشكل فعال ومنطقي.

٣.٩.٢ - لماذا تستخدم لغة MySQL؟

تُستخدم MySQL على نطاق واسع نظرًا لمجموعة من العوامل التي يجعلها خيارًا مفضلاً للعديد من المطوريين والمؤسسات. تشمل هذه العوامل:

- الأداء العالي: تم تحسين MySQL للتعامل مع كميات كبيرة من البيانات واسترجاعها بسرعة، مما يجعلها مناسبة للتطبيقات التي تتطلب معالجة سريعة للبيانات، مثل منصات التواصل الاجتماعي وخدمات البث.
- الموثوقية: تُعرف MySQL بموثوقيتها في معالجة المعاملات وضمان سلامة البيانات، حيث تدعم معاملات ACID (الذرية، الاتساق، العزل، والمتنانة)، مما يضمن معالجة موثوقة ودقيقة للمعاملات في قاعدة البيانات.
- سهولة الاستخدام: يمكن للمطوريين تثبيت وإدارة MySQL بسهولة نسبية، وغالبًا ما يتم ذلك في دقائق. كما أنها توفر أدوات معايدة موثوقة مثل MySQL Workbench لتسهيل الإدارة.
- قابلية التوسيع: يمكن لـ MySQL التعامل مع كميات متزايدة من البيانات والمستخدمين، مما يجعلها مناسبة للمشاريع التي تبدأ صغيرة وتتم توسيعها تدريجياً لتلبية متطلبات أحجام البيانات الضخمة وعدد كبير من الاتصالات المتزامنة.
- مفتوح المصدر وفعال من حيث التكلفة: كونها مفتوحة المصدر يعني أنها متاحة للاستخدام والتعديل والتوزيع مجانيًا بموجب ترخيص GNU General Public License، مما يقلل من تكاليف التطوير.
- مجتمع واسع الدعم: تتمتع MySQL بمجتمع كبير ونشط من المطوريين والمستخدمين الذين يساهمون في تطويرها وتقديم الدعم عبر المنتديات والقنوات المختلفة.
- التوافق الواسع: يتواافق MySQL مع العديد من أنظمة التشغيل واللغات البرمجية والأطر التطويرية، مما يسهل دمجه في بيئات تطوير متعددة، ويدعم مجموعة واسعة من أنواع البيانات، بما في ذلك النصوص، الأرقام، التوارikh، وكذلك أنواع البيانات الحديثة مثل JSON.
- الأمان: يوفر ميزات أمنية لحماية البيانات من خلال آليات التشفير والتحكم في الوصول، مما يُعد أساسياً في التطبيقات الحساسة.

٤.٩.٢ - ماهي المزايا الأساسية لغة MySQL؟

تلخص المزايا الأساسية لـ MySQL في النقاط التالية، والتي تُبرز سبب تفضيلها كحل لإدارة قواعد البيانات في مختلف الصناعات:

- الأداء العالي: تصميم يركز على سرعة عمليات تخزين واسترجاع البيانات.
- الموثوقية وسلامة البيانات: دعم معاملات ACID يضمن دقة وسلامة البيانات حتى في حالات الأعطال.
- قابلية التوسيع: القدرة على التعامل مع قواعد بيانات ضخمة وعدد كبير من المستخدمين المتزامنين.
- سهولة الاستخدام والدعم: تثبيت سريع وبسيط، مع أدوات إدارة سهلة ومجتمع دعم متفاعل.
- مفتوح المصدر وفعال التكاليف: إمكانية التعديل والتخصيص مجاناً، مع استفادة من تحديات مستمرة.
- التوافق الواسع: يعمل على أنظمة تشغيل ولغات برمجة مختلفة، مما يسهل دمجه في أي بيئة.
- الأمان: يوفر ميزات أمنية لحماية البيانات، بما في ذلك التشفير والتحكم في الوصول.

١٠.٢- تقنية ترميز بصري سريع QR Code

١٠.٢.١- ما هو رمز الاستجابة السريعة (QR Code)؟

رمز الاستجابة السريعة، المعروف اختصاراً بـ "QR Code" (من Quick Response)، هو نظام رمزي ثنائي الأبعاد يتكون من مجموعة مربعات سوداء وبضاء مرتبة في شكل مربع. يُشبه هذا الرمز الباركود التقليدي، ولكنه يتفوق عليه بقدراته على تخزين كميات أكبر بكثير من البيانات. تم تطويره في اليابان عام ١٩٩٤ بواسطة شركة Denso Wave، وهي إحدى الشركات التابعة لشركة تويوتا، وكان الهدف الأولي منه هو تتبع قطع غيار السيارات في أثناء عملية التصنيع. منذ ذلك الحين، شهدت هذه التقنية انتشاراً واسعاً في مجالات متعددة بفضل فعاليتها وسهولة استخدامها.

١٠.٢.٢- لماذا يستخدم QR Code؟

يستخدم رمز QR Code لأسباب متعددة تتعلق بتبسيط الوصول إلى المعلومات وتوفير حلول فعالة في العديد من القطاعات. تتمثل أهمية استخدامه في قدرته على توفير رابط فوري و مباشر بين المحتوى المادي والمحتوى الرقمي، مما يلغي الحاجة إلى إدخال البيانات يدوياً. تشمل أبرز استخداماته ما يلي:

- التسويق والإعلان: يُوضع الرمز على المنتجات، الملصقات، والإعلانات لتوجيه العملاء مباشرة إلى موقع الويب، الصفحات الترويجية، أو تزيل التطبيقات، مما يزيد من تفاعل المستخدمين وفعالية الحملات التسويقية.
- الدفع الإلكتروني: يستخدم لتسهيل عمليات الدفع، حيث يمكن للمستخدم مسح الرمز لإتمام عملية شراء أو تحويل مالي بأمان وسرعة.
- التعليم: يمكن استخدامه لتوفير وصول سريع إلى مواد تعليمية إضافية، مقاطع فيديو، أو مواقع ويب ذات صلة بالدروس، مما يعزز التعلم التفاعلي.
- إدارة المخزون وتتبع المنتجات: في سلسل التوريد والخدمات اللوجستية، يُسهل رمز QR تتبع المنتجات وتوثيقها بدقة عالية.
- المعلومات والتواصل: يمكن استخدامه لمشاركة معلومات الاتصال (مثل بطاقات العمل الرقمية)، أو تقديم تفاصيل عن الفعاليات، أو توجيه المستخدمين إلى قوائم الطعام في المطعم.
- الخدمات الحكومية والخاصة: يستخدم في توثيق الهوية، وتسجيل الدخول، وتبسيط الإجراءات الإدارية.

٣٠٢- ما هي المميزات الأساسية ل QR Code؟

تتمتع رموز QR Code بالعديد من المزايا التي تُسهم في انتشارها الواسع وتجعلها أداة قوية في العصر الرقمي. من أبرز هذه المزايا:

- سعة تخزين عالية: يمكنها تخزين كميات كبيرة من البيانات مقارنة بالباركود التقليدي، بما في ذلك النصوص، عناوين URL، أرقام الهواتف.
- سرعة القراءة والاستجابة: تُمكن الأجهزة الذكية من مسح وقراءة الرمز بسرعة فائقة، مما يوفر وصولاً فورياً إلى المعلومات.
- سهولة الاستخدام: لا تتطلب سوى جهاز مزود بكاميرا وتطبيق قارئ QR Code (وكثيراً ما يكون مدمجاً في كاميرا الهاتف)، مما يجعلها في متناول الجميع.

- تحمل الأخطاء (Error Correction): بفضل تقنيات تصحيح الأخطاء المدمجة، يمكن قراءة الرمز، حتى لو تعرض جزء منه للتلف أو التشوّه، مما يزيد من موثوقيته.
- المرونة وتعدد التطبيقات: تُستخدم في مجموعة واسعة من المجالات مثل التسويق، التعليم، الخدمات اللوجستية، والدفع الإلكتروني.
- فعالية التكلفة: يُعد إنشاء وطباعة رموز QR Code عملية غير مكلفة وسريعة، مما يجعلها خياراً اقتصادياً للشركات والأفراد.
- إمكانية التتبع والتحليل: يمكن تتبع عدد مرات مسح الرمز وجمع بيانات حول التفاعلات، مما يُقدم رؤى قيمة للحملات التسويقية.

٤.١٠.٢ - كيف يعمل QR Code؟

يُعمل رمز QR Code بناءً على مبدأ تحويل البيانات الرقمية إلى نمط مرئي يمكن للأجهزة الإلكترونية المزودة بكاميرا مسحه وتقسيره. يتكون الرمز من وحدات بكسل فردية (مربعات صغيرة) تُشكل نمطاً فريداً يمثل البيانات المشفرة. يتضمن الرمز عناصر تنظيمية أساسية، مثل المربعات الثلاثة الكبيرة الموجودة في زواياه، والتي تُعرف بأنماط تحديد الموقع (Finder Patterns). تساعد هذه الأنماط الماسح الضوئي (مثل كاميرا الهاتف الذكي) على تحديد الرمز وتوجيهه بشكل صحيح، حتى لو كان الرمز مائلاً أو غير واضح تماماً. عند مسح الرمز، تلتقط الكاميرا صورته، ثم يقوم البرنامج المتخصص بفك تشفير هذا النمط المعقد من المربعات البيضاء والسوداء. تُترجم هذه الأنماط إلى بيانات رقمية قابلة للاستخدام، سواء كانت عنوان موقع إلكتروني، نصاً، معلومات اتصال، أو بيانات دفع. تتميز رموز QR Code بخاصية تصحيح الأخطاء (Error Correction)، التي تعتمد على خوارزميات مثل Reed-Solomon، مما يعني أن الرمز يمكن أن يقرأ بنجاح، حتى لو تعرض جزء منه للتلف أو التشويش (قد يصل إلى ٣٠٪ من الرمز).

١١.٢ - مكتبة إرسال البريد في Node.js (Nodemailer)

١.١١.٢ - ما هي Nodemailer؟

Nodemailer هي مكتبة برمجية مُصممة خصيصاً لتطبيقات Node.js لتسهيل عملية إرسال البريد الإلكتروني. تعتبر أداة قوية وموثوقة تتيح للمطوريين إرسال رسائل البريد الإلكتروني مباشرة

من تطبيقاتهم دون الحاجة إلى تعقيدات إعدادات الخوادم يدوياً. تعمل كطبقة تجريد بين التطبيق وخوادم البريد، مما يجعل عملية الإرسال سلسة وفعالة.

٢.١١.٢ - كيف تعمل؟

تعتمد Nodemailer على إنشاء "ناقل" (Transporter) يتصل بخادم البريد الإلكتروني (مثل Gmail أو خادم Outlook أو خادم SMTP آخر). يتم تكوين هذا الناقل بمعلومات المصادقة الخاصة بحساب البريد الإلكتروني وإعدادات الخادم. بمجرد إنشاء الناقل، يمكن تحديد محتوى الرسالة (مثل المرسل والمسلتم والموضوع والنص) ثم إرسالها باستخدام طريقة بسيطة. المكتبة تدعم بروتوكولات الأمان مثل SSL/TLS لضمان تشفير الاتصال، كما تعالج الأخطاء تلقائياً وتتوفر ردود فعل واضحة حول حالة الإرسال.

٣.١١.٢ - لماذا تستخدم Nodemailer؟

السهولة والكفاءة: تتيح للمطوريين إرسال البريد الإلكتروني ببضعة أسطر من التعليمات البرمجية دون الحاجة إلى معرفة عميقة ببروتوكولات البريد.

التكامل السلس مع Node.js: مصممة خصيصاً لبيئة Node.js، مما يجعلها متواقة تماماً مع التطبيقات الحديثة القائمة على هذه المنصة.

المرونة: تدعم مجموعة واسعة من خوادم البريد (مثل Gmail, Yahoo, SendGrid، وغيرها) وتتيح تخصيص الإعدادات حسب الحاجة.

الميزات المتقدمة: تدعم إرفاق الملفات، وتنسيق HTML، والقوالب الديناميكية، مما يجعلها مناسبة للتطبيقات المعقدة مثل أنظمة التسويق أو الإشعارات.

الأمان: تقدم خيارات أمان قوية مثل المصادقة ثنائية العوامل والتشفير، مما يحمي البيانات أثناء النقل.

٤.١١.٢ - ما هي ميزاتها الرئيسية؟

إرسال عبر بروتوكولات متعددة: تدعم SMTP (البروتوكول القياسي)، بالإضافة إلى خدمات Amazon SES وSend mail.

المرفقات والقوالب: تسمح بإرفاق ملفات متعددة (مستندات، صور، إلخ) واستخدام قوالب مخصصة لتصميم الرسائل.

معالجة الأخطاء: تُقدم رسائل خطأ واضحة تساعد في تصحيح المشكلات بسرعة، مثل إعدادات المصادقة الخاطئة أو اتصال الشبكة.

الدعم غير المتزامن: تعمل بشكل غير متزامن (Asynchronous) لضمان عدم حظر التطبيق الرئيسي أثناء إرسال البريد.

التخصيص المتقدم: تدعم إعدادات متقدمة مثل تحديد أولوية الرسائل، وإضافة رؤوس مخصصة، وإدارة طابور الإرسال.

المجتمع والوثائق: تحظى بدعم مجتمعي قوي ووثائق شاملة تجعل عملية التعلم والتطوير أسهل.

١٢.٢ - مكتبات إشعارات React (react-toastify)

١.١٢.٢ - ما هي react-toastify ؟

مكتبة شاملة ومتطورة لعرض الإشعارات، توفر تحكماً كاملاً في مظهر وسلوك الإشعارات مع دعم لمجموعة واسعة من الميزات المتقدمة.

٢.١٢.٢ - كيف تعمل react-toastify ؟

- تقدم نظاماً متكاملاً لإدارة الإشعارات

- تدعم إنشاء قوالب مخصصة باستخدام مكونات React

- توفر تحكماً دقيقاً في حركات وتقويم الإشعارات

٣.١٢.٢ - لماذا تستخدم react-toastify ؟

- لشموليتها وتنوع خيارات التخصيص

- لدعمها المتقدم للرسوم المتحركة

- لتوفيقها الممتاز ومجتمعها النشط

- لتوافقها مع مختلف أنماط التصميم والمكتبات

٤.١٢.٢ - ما هي الميزات الرئيسية react-toastify ؟

- الأنواع المتعددة: ٦ أنواع مختلفة من الإشعارات

- الرسوم المتحركة: تأثيرات بصرية جميلة ومتعددة

- القوالب المخصصة: دعم لإنشاء إشعارات باستخدام مكونات React خاصة

- شريط التقدم: عرض شريط تقدم للتنبؤ بزمن اختفاء الإشعار
- التحكم الكامل: إمكانية التحكم في كل جانب من جوانب الإشعار
- الإضافات: دعم لإضافة أيقونات مخصصة وإجراءات تفاعلية

الفصل الثالث: الدراسة المرجعية

١.٣ - مقدمة:

بناءً على البحث الموسّع في الحلول الرقمية لإدارة مواقف السيارات، تبرز دراسات مرجعية رئيسية ساهمت في استخلاص رؤى قيمة لمشروع المنصة.

٢.٣ - أبرز الدراسات:

"Auckland Airport" [3] - من أهم المنصات المفيدة في مجال حجز مواقف السيارات كانت منصة "Auckland Airport" تقوم خدمات حجز مواقف السيارات في مطار أوكلاند، نيوزيلندا. تشمل خيارات متعددة مثل مواقف السيارات المغطاة وغير المغطاة، مع إمكانية الحجز المسبق.

١.١.٢.٣ - الإيجابيات:

توفر خيارات متعددة للمواقف، مع إمكانية الحجز المسبق، وسهولة الوصول إلى المطار، وخدمات عملاء متاحة.

٢.١.٢.٣ - السلبيات:

بعض المواقف تتطلب الدفع بوساطة بطاقة ائتمان فقط، وقد تكون الأسعار غير واضحة في بعض الأحيان.

"ParkWhiz" [4] - أيضاً منصة تسمى "ParkWhiz" هي منصة أمريكية تتيح للمستخدمين حجز مواقف السيارات مسبقاً في المدن الكبرى بالولايات المتحدة.

١.٢.٢.٣ - الإيجابيات:

توفر خيارات متنوعة للمواقف، وأسعار تنافسية، مع إمكانية الحجز عبر التطبيق.

٢.٢.٢.٣ - السلبيات:

قد تكون بعض المواقع محدودة في بعض المدن، وقد تتطلب رسوماً إضافية للحجز.

"SpotHero" [5] - كما أن هناك منصة "SpotHero"، وهي أيضاً منصة أمريكية مشابهة لـ "ParkWhiz"، تتيح للمستخدمين حجز مواقف السيارات مسبقاً في العديد من المدن الأمريكية.

[5]

١.٣.٢.٣ - الإيجابيات:

واجهة مستخدم سهلة، وخيارات متعددة للمواقف، وخصومات للحجز المبكر.

٢.٣.٢.٣ - السلبيات:

قد تكون الأسعار مرتفعة في أوقات الذروة، وبعض المواقف قد تتطلب الدفع نقداً .

٤.٢.٣ - لوحظ وجود منصة "JustPark" هي منصة بريطانية تتيح للمستخدمين حجز مواقف السيارات في موقع مختلف، بما في ذلك المواقف الخاصة[6] .

١.٤.٢.٣ - الإيجابيات:

توفر خيارات متعددة، مع إمكانية حجز المواقف الخاصة، ودعم لتطبيقات الهواتف الذكية.

٢.٤.٢.٣ - السلبيات:

قد تكون بعض المواقف بعيدة عن الوجهة النهائية، وقد تختلف الأسعار حسب الموقع.

٥.٢.٣ - أيضاً منصة "ParkMe" تقدم معلومات عن مواقف السيارات المتاحة في الوقت الحقيقي، مع إمكانية الحجز في بعض المواقع[7] .

١.٥.٢.٣ - الإيجابيات:

معلومات دقيقة عن توفر المواقف، واجهة مستخدم سهلة، دعم للخرائط.

٢.٥.٢.٣ - السلبيات:

قد لا تتوفر إمكانية الحجز في جميع المواقع، وبعض المعلومات قد تكون غير دقيقة .

٦.٢.٣ - لا حظنا انتشار الإقبال على منصة تسمى "Parkopedia"، على الرغم من أنها منصة عالمية، إلا أنها تحتوي على معلومات عن مواقف السيارات في بعض المدن العربية[8] .

١.٦.٢.٣ - الإيجابيات:

تغطية واسعة لمواقف السيارات، معلومات دقيقة حول الأسعار وتواجد المواقف، مع واجهة مستخدم سهلة الاستخدام، وتحديثات في الوقت الحقيقي، ودعم ل التطبيقات الأخرى.

٢.٦.٢.٣ - السلبيات:

عدم توفر الحجز المسبق في جميع المواقع، وتناقض المعلومات في بعض المناطق، تعتمد على البيانات المقدمة من المستخدمين، وعدم وجود خيارات دفع مباشرة، وقد تكون الخدمات محدودة في بعض الدول العربية.

٧.٢.٣ - كما أن هناك دراسة لتحسين إدارة مواقف السيارات الحضرية عبر نظام حجز عبر الإنترنط: خطوة نحو مدن أكثر ذكاءً "Enhancing Urban Parking Management Through an Online Reservation System: A Step Towards Smarter Cities" (مقالة بحثية)، النظام المذكور في الدراسة يُعد حلًا متكاملًا لإدارة مواقف السيارات، لكن نجاحه يعتمد على التكامل بين التقنية وقبول المستخدمين [9].

١٧.٢.٣ - الإيجابيات:

- للمستخدمين: توفير الوقت وتقليل التوتر عبر الحجز المسبق، وشفافية الأسعار (تسعير ديناميكي).
- لمشغلي المواقف: زيادة الإيرادات عبر التسعير الديناميكي. وتحسين استخدام المواقف وتقليل الفراغات.
- للمدن الذكية: تقليل الازدحام المروري والتلوث، ودمج التقنيات الحديثة (IoT) والذكاء الاصطناعي.

٢٠.٧.٢.٣ - السلبيات:

- اعتماد النظام على البنية التحتية: يحتاج إلى أجهزة استشعار (IoT) واتصال إنترنت قوي.
- التكلفة الأولية: تطوير النظام وتنشيط الأجهزة قد يكون مكلفاً.
- قبول المستخدمين: قد يتعدد بعض السائقون في استخدام النظام بسبب التعقيد أو عدم الثقة في التسعير الديناميكي.
- الأمان السيبراني: حماية بيانات المستخدمين والمدفوعات الإلكترونية تتطلب إجراءات أمان قوية.

٨.٢.٣ - أيضاً دراسة تطوير نظام إدارة تأجير سيارات عبر الويب "Web-Based Car Rental Management System" (مقالة بحثية)، تهدف الدراسة إلى تطوير نظام ويب متكامل لإدارة تأجير السيارات لشركة EDSU Cabs في نيجيريا، لتحويل العمليات اليدوية إلى نظام آلي يحسن الكفاءة ويسهل تجربة المستخدم [10].

١٨.٢.٣ - الإيجابيات:

- للمستخدمين: حجز السيارات عبر الإنترنط في أي وقت، وعرض تفاصيل السيارات (الصور، المواصفات)، وإدارة الحجوزات (تعديل/حذف) وتلقي إشعارات.
- لالمشغلين: إدارة المخزون (السيارات المتاحة/المؤجرة)، وتوليد تقارير عن الإيرادات والحجوزات. وتحسين الأمان عبر (جيوفسينج) تتبع المركبات.

- تقنية: واجهة مستخدم بسيطة (HTML, CSS, Bootstrap) وقاعدة بيانات منظمة Microsoft SQL Server) واختبارات شاملة (١٠٠٪ نجاح في الاختبارات).

٢.٨.٢.٣- السلبيات:

- قيود تقنية: النظام مصمم لبيئة Windows (IIS)، وقد لا يكون متوفراً مع جميع الخوادم، ولا يدعم التطبيقات المحمولة (يقتصر على الويب).
- الأمان: لم يذكر تفصيلاً حول تشفير البيانات أو حماية الدفع الإلكتروني.
- التكامل: لا يدعم الذكاء الاصطناعي للتباين بالطلب أو الصيانة الوقائي.

٩.٢.٣- في مقالة إخبارية "Parkin" تطلق تطبيقاً جواً متطوراً لإحداث ثورة في تجربة موافقات السيارات في دبي : تبرز هذه المقالة تطبيق باركين كنموذج رائد لمنصات موافقات السيارات الذكية، مقدمةً حلولاً مبتكرة لإدارة الموافق والدفع التلقائي، ما يوفر تجربة سلسة للمستخدمين ويحسن التنقل الحضري[11] .

١.٩.٢.٣- الإيجابيات:

- وظائف يمكن تطبيقها في الكراج: الحجز المسبق عبر التطبيق، والدفع التلقائي عبر ربط لوحة السيارة (تكنولوجيا ANPR)، وإدارة المخالفات أو رسوم التأخير.
- خدمات إضافية: غسيل السيارات أو الصيانة.
- تحسين تجربة المستخدم: تسجيل سهل عبر بطاقات الهوية أو الحسابات الرقمية، وتذكيرات تلقائية (مثل تجديد الحجز أو مواعيد الصيانة) .
- التقنيات الملهمة: التكامل مع أنظمة الدفع وتتبع المركبات داخل الكراج.

٢.٩.٢.٣- السلبيات:

- التكلفة العالية: تقنية التعرف إلى لوحات السيارات (ANPR) قد تكون مكلفة للكراجات الصغيرة.
- الاعتماد على بنية تحتية: يحتاج إلى اتصال إنترنت قوي وتكامل مع أنظمة حكومية (مثل تسجيل المركبات).
- منافسة الخدمات القائمة: في دبي، النظام مدعوم حكومياً، بينما مشروع ما قد يحتاج إلى شركات مماثلة.

فيما يلي جدول مقارنة بين الدراسات المرجعية للموقع المشابهة:

جدول (١,٣): مقارنة بين الدراسات المرجعية للموقع المشابهة للمنصة

المنصة / الدراسة	الإيجابيات	السلبيات
Auckland Airport [3] (منصة - نيوزيلندا)	- حجز مسبق لمواقف مغطاة/غير مغطاة - سهولة الوصول للمطار - دعم خدمة العملاء	- دفع حصري ببطاقات ائتمان - غموض الأسعار أحياناً
ParkWhiz [4] (منصة - أمريكية)	- خيارات متنوعة للمواقف - أسعار تنافسية - حجز عبر تطبيق مرن	- تعطية محدودة لبعض المدن - رسوم إضافية للحجز
SpotHero [5] (منصة - أمريكية)	- واجهة مستخدم سهلة - خيارات متعددة - خصومات للحجز المبكر	- أسعار مرتفعة في الذروة - دفع نقدi مطلوب ببعض المواقف
JustPark [6] (منصة - بريطانية)	- حجز مواقف خاصة - دعم تطبيقات الجوال - تنوع الخيارات	- بعد بعض المواقف عن الوجهة - تذبذب الأسعار حسب الموقع
ParkMe [7] (منصة - عالمية)	- معلومات آنية عن التوافر - دعم الخرائط - واجهة سهلة	- عدم شمولية الحجز - دقة معلومات متغيرة
Parkopedia [8] (منصة - عالمية)	- تغطية واسعة - تحديثات آنية - دقة - معلومات - (أسعار/توافر)	- لا تدعم الحجز المسبق - بيانات غير موثوقة ببعض المناطق - محدودية الدفع
دراسة: تعزيز إدارة المواقف الحضرية [9]	- تقليل الازدحام والتلوث - تسعير ديناميكي	- تكلفة بنية تحتية عالية - تحديات أمن سيبراني

	- دمج IoT و AI -	- مقاومة المستخدمين -
دراسة: نظام تأجير سيارات ويب [10]	<ul style="list-style-type: none"> - واجهة مستخدم بسيطة - إدارة مخزون وتقارير - تتبع مركبات(جيوجن-فنسينج) 	<ul style="list-style-type: none"> - محدودية النظام Windows (فقط) - غياب تفاصيل الأمان - لا يدعم الذكاء الاصطناعي
Parkin [11] تطبيق (دبي)	<ul style="list-style-type: none"> - دفع تلقائي (تعرف لوحات) - خدمات إضافية (غسيل/صيانة) - تذكير ذكي 	<ul style="list-style-type: none"> - تكلفة تقنية ANPR مرتفعة - اعتماد على بنية تحتية قوية - حاجة لشراكات حكومي

الفصل الرابع: الدراسة التحليلية

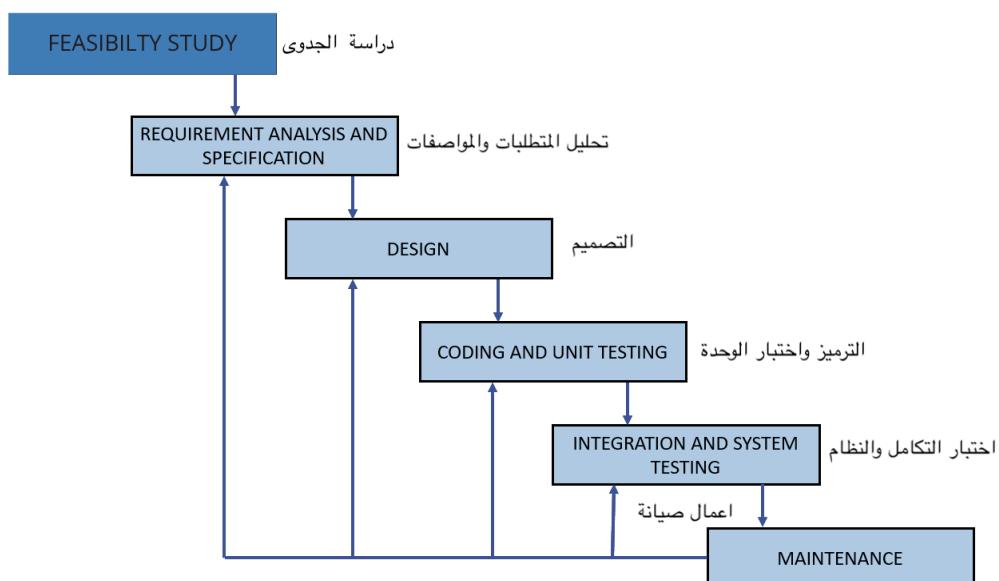
٤.١ - منهجية النظام:

تم الاعتماد على نموذج الشلال التكراري (Iterative Waterfall Model) ، الذي يجمع بين منهجية الشلال التقليدية وإمكانية التكرار لتصحيح الأخطاء في مراحل مبكرة. حيث يُعد هذا النموذج تطوراً للنموذج الكلاسيكي، تم تصميمه للتغلب على أبرز قيود النموذج الأصلي، لا سيما صعوبة الرجوع إلى المراحل السابقة لتصحيح الأخطاء بعد اكتشافها. إذ يتميز هذا النموذج بقدرتة على توفير نموذج عمل أولي للنظام في مراحل التطوير المبكرة، مما يُسهل الكشف عن العيوب الوظيفية أو التصميمية بشكل استباقي. وبالتالي، يمكن معالجة المشكلات في وقت مبكر، مما يقلل من التكاليف والجهود المطلوبة للإصلاح مقارنةً باكتشافها في مراحل متاخرة.

وقد تم اختيار هذا النموذج لملاءمته لطبيعة المشروع، إذ تتسق المتطلبات بالوضوح والثبات، ولا توجد حاجة لتعديلات جذرية أثناء التنفيذ. والتقنيات المستخدمة مفهومة ومحددة مسبقاً.

إضافة إلى ذلك، وفر النموذج مرونة كافية للعودة إلى المراحل السابقة عند الحاجة، سواءً بسبب عدم الفهم الدقيق للمتطلبات أو وجود اختلافات بين التصميم والأهداف المرجوة.

هذا المنهج ساهم في تعزيز الجودة والكفاءة، مع الحفاظ على الموارد والوقت، مما جعله خياراً مثالياً للمشاريع قصيرة المدى ذات المتطلبات المحددة بوضوح.



الشكل (٤) - نموذج الشلال التكراري (Iterative Waterfall Model)

٤.٢- مراحل تشكيل النظام:

٤.١.٢- مرحلة تحليل المتطلبات (Requirements Analysis):

٤.١.٢.١- جمع المتطلبات من المستخدمين المحتملين:

- إجراء مقابلات مع أصحاب السيارات وموظفي الكراج لفهم احتياجاتهم ومتطلباتهم.
 - تحليل البيانات المجمعة لتحديد المتطلبات بدقة، بما في ذلك أنواع الخدمات المطلوبة وطرق الدفع المفضلة.
 - تحليل النظم المشابهة: دراسة منصات إدارة حجوزات الكراجات المشابهة لتحديد المزايا والعيوب.
 - تحديد نقاط القوة والضعف للاستفادة منها في تصميم المنصة الجديدة.
- ٤.١.٢.٣- تحديد المتطلبات الوظيفية وغير الوظيفية:
- تم تحديد المتطلبات الوظيفية مثل إمكانية حجز موعد، إدارة الحسابات، وتقديم تقييمات للخدمات.
 - تم تحديد المتطلبات غير الوظيفية مثل الأمان، الأداء، وسهولة الاستخدام.

٤.٢.٢- مرحلة التصميم (System Design):

٤.٢.٢.١- تصميم هيكلية النظام:

- إعداد مخطط تفصيلي يوضح العلاقات بين المكونات المختلفة للنظام.

٤.٢.٢.٢- تصميم قاعدة البيانات: إعداد مخطط ERD لتحديد الجداول والعلاقات بينها، مثل جدول المستخدمين، جدول الحجوزات، وجدول الخدمات.

٤.٢.٢.٣- تصميم واجهات المستخدم:

- تقديم نماذج لواجهة المنصة الرئيسية، تشمل صفحة تسجيل الدخول، صفحة الحجز، وصفحة إدارة الحساب.

٤.٢.٢.٤- تصميم المخططات:

- إعداد مخطط حالات الاستخدام (Use Case Diagram) لتوضيح العمليات المختلفة التي يمكن للمستخدمين القيام بها.
- إعداد مخطط الفئات (Class Diagram) لتحديد الكائنات وال العلاقات بينها، مثل كائن المستخدم وكائن الحجز.

٣.٢.٤ - مرحلة التطوير (Implementation):

٤.١.٣.٢.٤- تطوير قاعدة البيانات:

- إنشاء الجداول باستخدام MYSQL
- ربط الجداول باستخدام:

 - (a) استخدام المفاتيح الأساسية والأجنبية:
 - المفتاح الأساسي (Primary Key): هو عمود أو مجموعة من الأعمدة التي تميز كل صف في الجدول بشكل فريد.
 - المفتاح الأجنبي (Foreign Key): هو عمود في جدول يشير إلى المفتاح الأساسي في جدول آخر. هذا الرابط يسمح بإنشاء علاقة بين الجداول.
 - (b) استخدام JOIN: لدمج البيانات من جداول متعددة.

٤.٢.٣.٢.٤- تطوير واجهات المستخدم:

- استخدام JavaScript و React.js للتصميم الواجهة الأمامية.

٤.٣.٣.٢.٤- تطوير خدمات الخلفية:

- استخدام Express.js و Node.js

٤.٤.٣.٢.٤- ربط المكونات مع بعضها:

- تكامل الواجهة الأمامية مع خدمات الخلفية وقاعدة البيانات.

٤.٤.٢.٤ - مرحلة الاختبار (Testing):

٤.٤.٢.٤.١- اختبار الوحدات (Unit Testing):

- اختبار كل وحدة برمجية للتأكد من عملها بشكل مستقل.
- ٤.٤.٢.٤.٢- اختبار التكامل (Integration Testing):
- التأكد من أن جميع المكونات تعمل معًا بشكل صحيح.
- ٤.٤.٢.٣- اختبار النظام (System Testing):
- التحقق من أداء النظام ككل.

٤.٤.٤.٢.٤- اختبار قبول المستخدم (User Acceptance Testing):

- إجراء اختبارات مع المستخدمين للحصول على ملاحظاتهم.

٤.٣- تحليل النظام:

٤.١.٣.٤- جمع المتطلبات من المستخدمين المحتملين

جمع المتطلبات يُعد خطوة محورية في تصميم أي نظام جديد، حيث يهدف إلى فهم احتياجات المستخدمين وتحديد الخصائص التي يجب أن يتضمنها النظام لتحقيق أهدافه. يمكن تنفيذ هذه العملية عبر مرحلتين رئيسيتين:

٤.١.١.٣.٤- إجراء مقابلات مع أصحاب السيارات وموظفي الكراج:

أ. أصحاب السيارات:

ركزت المقابلات مع أصحاب السيارات على فهم التحديات التي تواجههم أثناء البحث عن مواقف أو استخدام الكراجمات. تشمل الأسئلة استفسارات حول الوقت المستغرق للعثور على موقف، مدى سهولة عملية الحجز، طرق الدفع المفضلة، والخدمات الإضافية التي قد يحتاجونها مثل الصيانة أو الشحن الكهربائي. هذه المقابلات تساعد في تحديد الأولويات من وجهة نظر المستخدم النهائي، مما يضمن أن النظام يلبي احتياجاتهم بشكل مباشر.

ب. موظفو الكراج:

أجريت مقابلات مع موظفي الكراج لفهم العمليات التشغيلية اليومية، التحديات التي تواجههم، والاحتياجات التقنية التي يمكن أن تسهم في تحسين الكفاءة. على سبيل المثال، كيفية إدارة الحجوزات، التعامل مع العملاء، أو استخدام الأدوات الحالية لتبني المواقف المتاحة. هذه المعلومات تساعد في تصميم نظام يسهل عمل الموظفين ويحسن من جودة الخدمة المقدمة.

٤.٢.١.٣.٤- تحليل البيانات المجمعة لتحديد المتطلبات بدقة:

بعد جمع البيانات من المقابلات، يتم تحليلها باستخدام أدوات تحليل البيانات النوعية والكمية. الهدف هو استخراج المتطلبات الواضحة، مثل:

أنواع الخدمات المطلوبة: حجز المواقف، الدفع الإلكتروني، خدمات الصيانة.

طرق الدفع المفضلة: بطاقات الائتمان، المحافظ الرقمية، الدفع النقدي.

الخصائص التقنية: مثل توفير واجهات سهلة الاستخدام.

يتم تصنيف هذه المتطلبات إلى فئات تشمل المتطلبات الوظيفية (features) وغير الوظيفية (مثل الأمان وسهولة الاستخدام). هذا التحليل ساهم في بناء نظام يلبي توقعات المستخدمين ويعالج نقاط الألم الحالية.

٤.٢.٣.٤ - تحليل النظم المشابهة:

تحليل النظم المشابهة يُعد خطوة استراتيجية لفهم السوق الحالي وتحديد كيفية تحسين النظام الجديد. يمكن تنفيذ ذلك عبر مرحلتين رئيسيتين:

٤.١.٢.٣.٤- دراسة منصات إدارة حجوزات الكراجات المشابهة:

يتم تحليل الأنظمة الحالية المستخدمة في إدارة حجوزات الكراجات، سواء كانت محلية أو عالمية. يشمل ذلك دراسة ميزات هذه الأنظمة مثل واجهات المستخدم، طرق الحجز، التكامل مع تطبيقات الهاتف المحمول، وإمكانيات الدفع الإلكتروني.

على سبيل المثال، تم دراسة أنظمة مثل "SpotHero" و "ParkWhiz"، التي توفر حلولاً مبتكرة لحجز المواقف مسبقاً، مقارنة الأسعار، والدفع الإلكتروني. حيث يتم التركيز على فهم كيفية تقديم هذه الأنظمة لتجربة سلسة للمستخدمين، مثل توفير معلومات في الوقت الفعلي حول توفر المواقف.

٤.٢.٢.٣.٤- تحديد نقاط القوة والضعف للاستفادة منها في تصميم المنصة الجديدة:

أ. نقاط القوة:

تشمل الميزات التي أثبتت نجاحها في الأنظمة المشابهة، مثل توفير معلومات في الوقت الفعلي، أو تقديم خيارات دفع متعددة. يمكن الاستفادة من هذه النقاط في تعزيز تصميم النظام الجديد.

ب. نقاط الضعف:

يتم تحديد العيوب أو التحديات التي تواجهها الأنظمة الحالية، مثل صعوبة الاستخدام، محدودية طرق الدفع، أو عدم توفر دعم فني كافٍ. يمكن معالجة هذه النقاط في النظام الجديد لضمان تقديم تجربة أفضل للمستخدمين.

ت. الفرص:

يتم استكشاف المجالات التي يمكن تحسينها أو تقديم ميزات جديدة فيها، مثل تقديم خدمات إضافية كالصيانة أو الشحن الكهربائي.

ث. التهديدات:

تشمل التحديات التي قد تواجه النظام الجديد، مثل المنافسة الشديدة أو التغيرات في اللوائح التنظيمية.

٤.٤ - متطلبات النظام:

من خلال جمع المتطلبات ودراسة النظم المشابهة، تم استخلاص متطلبات النظام كما يلي:

٤.٤.١ - المتطلبات الوظيفية:

المتطلبات الوظيفية تمثل الوظائف الأساسية التي يجب أن يؤديها النظام لتلبية احتياجات المستخدمين. وفيما يلي شرح موسع للمتطلبات الوظيفية:

٤.٤.١.١ - وحدة البحث والعرض:

- عرض المواقف الشاغرة: يجب أن يتمكن النظام من عرض عدد المواقف الشاغرة بشكل دقيق، سواء كان ذلك على مستوى إجمالي الكراج. هذه الميزة تتيح للسائقين اتخاذ قرارات سريعة بناءً على توفر المواقف.

٤.٤.٢ - وحدة الحجز والدفع للزوار:

- الحجز لمدة محددة: يتيح النظام للزوار حجز موقف لفترة زمنية محددة مسبقاً، مما يقلل من عدم اليقين ويضمن توفر الموقف عند الوصول.

- آلية الدفع: يوفر النظام خيارات دفع متعددة، مثل المحفظة الداخلية أو الدفع المسبق، لتلبية تفضيلات المستخدمين المختلفة.

- إنشاء وتأكيد الحجز مع توليد QR: يتم إنشاء رمز QR فريد لكل حجز، يستخدم لتأكيد الحجز عند الدخول.

- مسح QR عند الدخول: يتم مسح رمز QR عند بوابة الدخول لبدء احتساب الوقت تلقائياً.

- آلية التعامل مع المغادرة: يتعامل النظام مع حالات المغادرة المختلفة، سواء كانت في الوقت المحدد أو متأخرة. في حالة التأخير، يمكن تجديد الحجز وخصم الرسوم الإضافية من خلال الموظف.

- إصدار فاتورة إلكترونية: يتم إرسال الفاتورة إلكترونياً عبر البريد الإلكتروني أو داخل التطبيق، مما يسهل على المستخدمين تتبع معاملاتهم.

- نظام تنبئات: يتضمن النظام تنبئات لإبلاغ المستخدمين باقتراب انتهاء الوقت، مع إمكانية التجديد عند التأخير.

٤.٤.٣ - وحدة إدارة الاشتراكات:

- تسجيل الدخول للمشترك: يتيح النظام للمشتركيين تسجيل الدخول للوصول إلى حساباتهم.

- عرض وشراء باقات الاشتراك: يمكن للمشتركيين اختيار باقات اشتراك (شهري، سنوي)

- الدفع الإلكتروني للاشتراك: يوفر النظام خيارات دفع إلكترونية آمنة لتسهيل عملية الاشتراك.

- نظام QR خاص بالاشتراك: يتم استخدام رمز QR مخصص للمشتركين عند الدخول والخروج.

- عرض حالة الاشتراك وتنبيهات الانتهاء: يمكن للمشتركين عرض حالة اشتراکهم وتلقي تنبيهات قبل انتهاء الاشتراك.

٤.٤.٤ - وحدة لوحة التحكم (الموظف/المدير):

- تسجيل الدخول للموظفين والمديرين: يتيح النظام تسجيل الدخول بأدوار وصلاحيات محددة.

- عرض لوحة التحكم (Dashboard): تعرض لوحة التحكم بيانات حية مثل نسبة الإشغال، عدد السيارات الداخلة والخارجة، وتنبيهات النظام.

- إدارة الاشتراكات (مدير الكراج): يمكن للمدير مراجعة طلبات الاشتراك وإدارة قائمة المشتركين.

- تحديد وتعديل الأسعار (مدير الكراج): يتيح النظام للمدير تعديل أسعار الساعة، اليومية، الاشتراكات، والفعاليات.

- توليد واستعراض التقارير (مدير الكراج): يمكن للمدير توليد تقارير حول الإشغال، الاشتراكات، ومدة الوقوف لتحليل الأداء.

٤.٥.١ - وحدة التعامل مع المشكلات (الموظف):

- تلقي البلاغات: يمكن للموظفين من تلقي البلاغات عبر الاتصال أو النظام.

- البحث في النظام: يمكن البحث باستخدام رقم اللوحة أو وقت الدخول لتحديد الرسوم في حالة فقدان QR.

- تسجيل الدفع اليدوي: يتيح النظام تسجيل المدفوعات اليدوية في حالات الطوارئ.

- تسجيل تفاصيل الحوادث: يتم تسجيل تفاصيل الحوادث والإجراءات المتخذة لضمان التوثيق.

٤.٤.٢ - المتطلبات غير الوظيفية:

المتطلبات غير الوظيفية ترتكز على كيفية عمل النظام بدلاً من الوظائف التي يؤديها. وهي تشمل:

٤.١.٢ - الأداء:

يجب أن يتميز النظام بسرعة استجابة عالية، خاصة عند عرض التوفير وتحديث نسبة الإشغال.

يضمن ذلك تجربة مستخدم سلسة، حتى في أوقات الذروة.

٤.٢.٢ - الأمان:

يجب تأمين بيانات المستخدمين والمعاملات المالية من خلال تقنيات تشفير متقدمة.

حماية النظام من الوصول غير المصرح به لضمان سلامة البيانات.

٤.٢.٤.٣ - القابلية للتتوسيع:

تصميم النظام ليكون قادرًا على التعامل مع زيادة عدد المستخدمين، الكراجات، والعمليات دون التأثير على الأداء.

٤.٢.٤.٤ - الموثوقية والإتاحة:

ضمان عمل النظام بشكل مستمر وموثوق، مع تقليل فترات التوقف إلى الحد الأدنى.

يتيح ذلك تقديم خدمة دائمة للعملاء والموظفين.

٤.٢.٤.٥ - سهولة الاستخدام:

تصميم واجهات مستخدم بديهية وسهلة الاستخدام لكل من العملاء والموظفين.

يضمن ذلك تقليل الحاجة إلى التدريب وزيادة رضا المستخدمين.

٤.٤.٣ - المستخدمون والأدوار:

٤.١,٤,٣ - مدير النظام: يتبع التقييمات ويطلع على التقارير ويضيف مدراء للكراجات.

٤.٢,٤,٣ - مدير الكراج: يشرف على العمليات والتقارير، ويبير الاشتراكات، ويحدد الأسعار.

٤.٣,٤,٣ - موظف خدمة العملاء: يدير المواعيد، ويساعد في حل مشكلات العملاء مثل فقدان QR، ويقوم بتسجيل الدخول اليدوي.

٤.٤,٣,٤ - العميل (السائق):

a) زائر مؤقت: يبحث عن موقف، ويحجز، ويدفع بالساعة.

b) مشترك: يشترك، ويستفيد من الدخول والخروج السلس.

٤.٥ - الوظائف الأساسية للنظام:

٤.١.٥.٤ - إدارة الوصول والمصادقة وتشمل:

- تسجيل الدخول إلى النظام

- تسجيل الخروج من النظام

٤.٢.٥.٤ - استعراض المعلومات والبحث وتشمل:

- عرض الكراجات وتفاصيلها

- عرض تفاصيل الاشتراك الحالي
- عرض الحجوزات النشطة
- الاطلاع على التقارير
- الاطلاع على تقييمات الكراجات
- ٣.٥.٤ - إدارة الحجوزات وتشمل:
 - حجز موقف
 - عرض تفاصيل الحجز
 - عرض سجل الحجوزات السابقة
 - تعديل الحجوزات (تمديد/إلغاء)
 - إنشاء حجز يدوي للعملاء
- ٤.٥.٤ - إدارة الاشتراكات والمدفوعات وتشمل:
 - شراء اشتراك جديد (شهري/سنوي)
 - تحميل رصيد المحفظة
- ٤.٥.٥ - إدارة الكراجات والمواقف وتشمل:
 - تحديث حالة الموقف (صيانة / إغلاق)
 - إضافة كراج جديد
 - ضبط ساعات العمل
 - تحديد الأسعار (الاشتراكات المؤقتة)
- ٤.٦.٥ - إدارة المستخدمين والصلاحيات وتشمل:
 - إدارة حسابات المستخدمين
- ٤.٧.٥ - إدارة الدعم والخدمة وتشمل:
 - تسجيل الشكاوى
 - تقييم الخدمة
 - حل مشاكل الدخول والخروج
- ٤.٨.٥ - عمليات التشغيل الميداني وتشمل:
 - مسح QR للدخول والخروج
- ٤.٩.٥ - وظائف النظام الأساسية (الدعم التقني) وتشمل:
 - إرسال إيميلات
 - تنبيهات إشعارات

- توليد رموز فريدة
- التحقق من الصلاحية
- تسجيل أحداث المسح

الجدول التالي يوضح الوظائف الأساسية للنظام والممثل المسؤول عن كل وظيفة:

جدول (٤) : الوظائف الأساسية للنظام

Actors الممثلين	حالة الاستخدام Use Case	UC
All	تسجيل الدخول الى النظام	1
All	استعادة كلمة المرور	2
Client	إنشاء حساب جديد	3
Client, visitor	عرض قائمة الكراجات	4
Client	عرض أماكن الكراجات	5
Client	عرض تفاصيل الكراج	6
Client	حجز موقف	7
Client	عرض الحجوزات	8
Client	عرض تفاصيل الحجز	9
Client	إدارة الحجوزات	10
Client	إلغاء الحجز	11
Client	شراء اشتراك جديد(شهري أ سنوي)	12
Client	إدارة الاشتراكات	13
Client	إلغاء اشتراك	14

Client	دفع إلكتروني	15
Client	تسجيل الشكاوى	16
Client	تقييم الخدمة	17
Employee	إنشاء حجز يدوي للعملاء	18
Employee	مسح QR	19
Employee	إعادة إصدار QR	20
Employee	تسجيل الحوادث	21
Employee	تفعيل الدخول اليدوي	22
Employee	إدارة المواقف	23
Garage Admin	ضبط ساعات العمل	24
Garage Admin	إدارة المحافظ	25
Garage Admin	تحديد الأسعار (الاشتراكات المؤقتة)	26
Garage Admin	عرض الموظفين	27
Garage Admin	إضافة موظفين جدد	28
System Admin	إدارة حسابات المستخدمين	29
Garage Admin, System Admin	إضافة كراج	30
Garage Admin	إضافة صورة للكراج	31
Garage Admin	الاطلاع على التقارير المقدمة	32
Garage Admin	مخططات الحجوزات والإيرادات	33

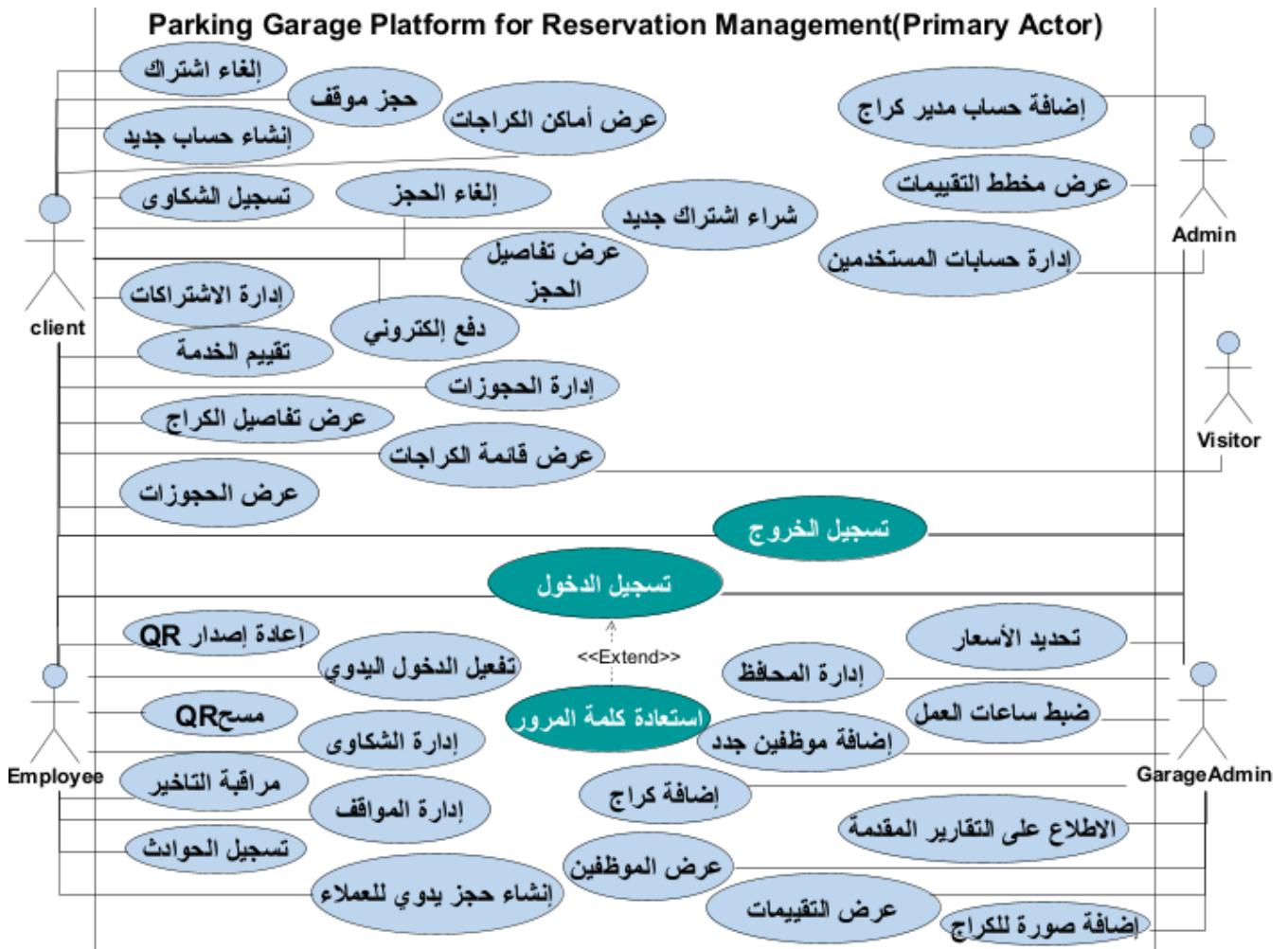
Garage Admin	عرض التقييم للخدمات	34
System Admin	إضافة حساب مدير كراج	35
All	تسجيل الخروج من النظام	36

٦.٤ - حالات الاستخدام:

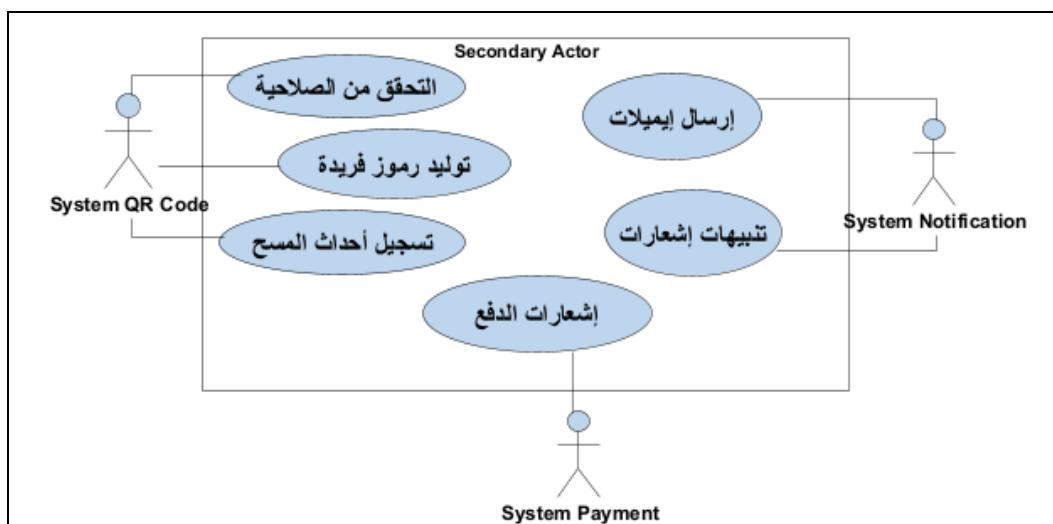
تُعدّ نماذج حالات الاستخدام أداة محورية لفهم تفاعل الأطراف المختلفة مع مكونات النظام. يعتمد هذا النظام على أربعة ممثليين رئيسيين يشكلون النواة التشغيلية: العميل الذي يستفيد من الخدمات، الموظف الذي ينفذ العمليات اليومية، مدير الكراج الذي يشرف على تنظيم الموارد وضمان جودة الخدمة، ومدير النظام الذي يتولى الصلاحيات التقنية والإدارية لضمان استمرارية الأداء بكفاءة.

إلى جانب هؤلاء، يتكامل النظام مع ممثليين ثانويين أو أنظمة مساعدة، تشمل نظام الدفع الإلكتروني لتأمين المعاملات المالية، نظام مسح رمز الاستجابة السريع (QR) لتبسيط عمليات الدخول والتوثيق، ونظام الإشعارات لضمان التواصل الفعال مع المستخدمين. إن هذا التوزيع المتكامل للأدوار قد ساهم في توفير إطاراً شاملًا يربط بين متطلبات المستخدمين وعمليات التشغيل الداخلية، مما عزّز من مرونة النظام وكفاءته في تحقيق أهدافه الخدمية.

فيما يلي مخطط لحالات الاستخدام لكل من الممثليين الرئيسيين والممثليين الثانويين:



الشكل (٢٠.٤) – مخطط حالات الاستخدام (الممثلين الرئيسيين)



الشكل (٣٠.٤) – مخطط حالات الاستخدام (الممثلين الثانويين)

تحليل حالات الاستخدام لمنصة كراج السيارات لإدارة الحجوزات:

سنقدم تحليلاً عميقاً لحالات الاستخدام الأساسية لمنصة كراج السيارات لإدارة الحجوزات، مع التركيز على تحديد الفاعلين (Actors) وال العلاقات الوظيفية المعقدة بين حالات الاستخدام، ممثلاً بعلاقتي التضمين include والتوضيع extend.

و قبل الخوض في تفاصيل حالات الاستخدام، من الضروري استعراض المفاهيم الأساسية التي تحكم هذا التحليل:

حالة الاستخدام (Use Case): تمثل وصفاً مكتوباً لسيناريوهات تفاعل النظام مع الكيانات الخارجية (الفاعلين) لإنجاز هدف وظيفي محدد. تركز حالات الاستخدام على المتطلبات الوظيفية للنظام، وتصف ماذا يفعل النظام من منظور المستخدم.

الفاعل (Actor): هو دور يقوم به شخص أو شيء (مثل نظام آخر أو جهاز) يتفاعل مع النظام لإنجاز هدف معين. الفاعلون هم كيانات خارج حدود النظام، ويمكن تصنيفهم إلى فاعل أساسى (يبدأ حالة الاستخدام وله هدف يسعى لتحقيقه) وفاعل داعم (يقدم خدمة للنظام).

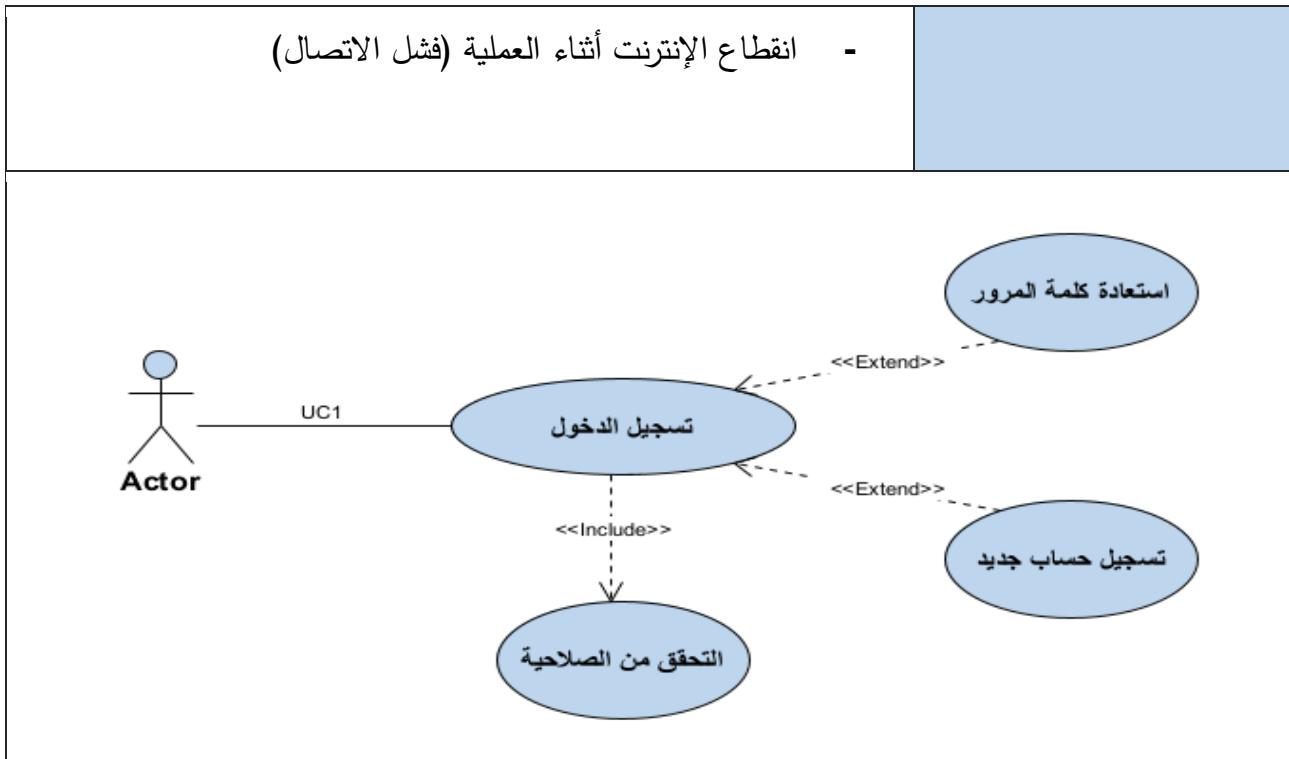
علاقة التضمين (include): علاقة موجهة بين حالتى استخدام توضح أن سلوك حالة الاستخدام المتضمنة يتم إدخاله بشكل إلزامي في سلوك حالة الاستخدام الأساسية. تُستخدم لتبسيط حالات الاستخدام الكبيرة أو لاستخراج الأجزاء المشتركة من السلوك التي تُعد ضرورية لإكمال الوظيفة الأساسية. السلوك المتضمن مطلوب وغير اختياري.

علاقة التوضيع (extend): علاقة موجهة تُستخدم لإظهار كيف ومتى يمكن إدخال حالة استخدام اختيارية (مُوسعة) في حالة استخدام أساسية. السلوك الموسع اختياري ومكمل، وقد لا يكون ذات معنى بمفرده، بينما تكون حالة الاستخدام الأساسية كاملة وذات معنى بمفردها حتى بدون التوضيع. تحدث حالة الاستخدام الموسعة فقط عندما يتم استيفاء معايير أو شروط معينة

فيما يلي تحليل مفصل لكل حالة استخدام من الوظائف الأساسية للنظام، مع تحديد الفاعلين المشاركين وشرح العلاقات «include» و «extend» وفقاً للمبادئ الموضحة سابقاً.

جدول (٤،٢) : حالة الاستخدام ١ تسجيل الدخول للنظام

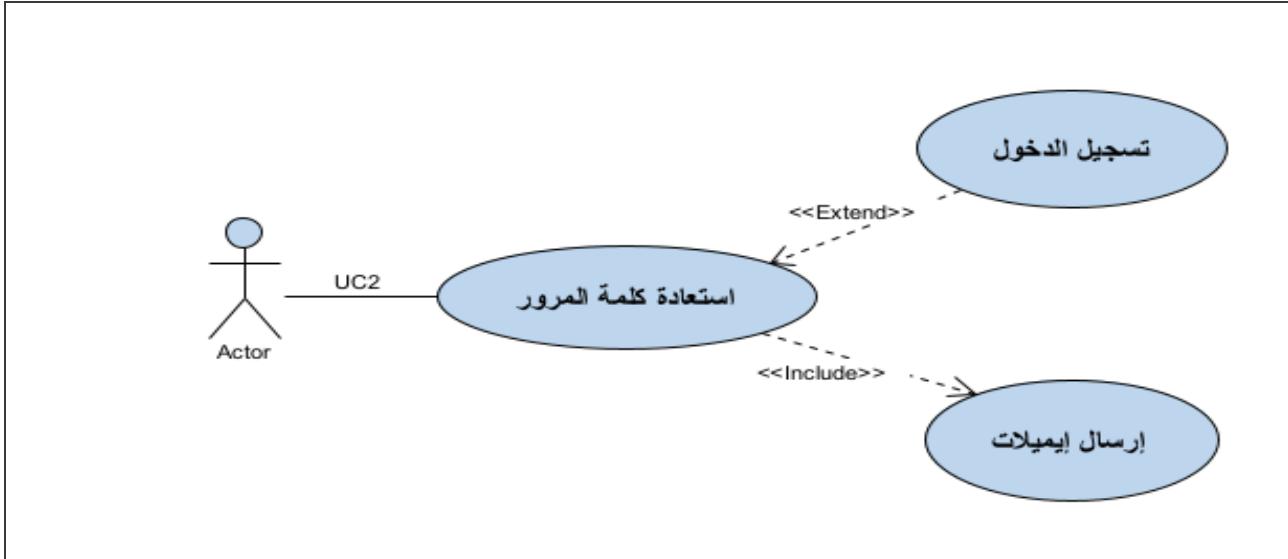
مواصفات حالة الاستخدام	
UC1	معرف حالة الاستخدام
تسجيل الدخول الى النظام	اسم حالة الاستخدام
- الفاعل الرئيسي: المستخدم (عميل/موظفي مدير كراج/مدير نظام) - الفاعل الداعم: النظام (خدمة التوثيق)	الجهات الفاعلة
تمكين أي مستخدم (عميل، موظف، مدير كراج، أو مسؤول نظام) من الوصول الآمن إلى النظام من خلال إدخال بيانات الاعتماد الخاصة به.	الوصف
- يجب أن يكون لدى المستخدم حساب مفعل في النظام. - توفر اتصال بالإنترنت	الشروط المسبقة
- يتم توجيه المستخدم إلى الواجهة المناسبة حسب نوع الحساب - إنشاء جلسة دخول للمستخدم.	الشروط اللاحقة
١. يفتح المستخدم الموقع ٢. ينقر على "تسجيل الدخول" ٣. يدخل البريد الإلكتروني ٤. يدخل كلمة المرور ٥. ينقر "دخول" ٦. يتحقق النظام من صحة البيانات ٧. البيانات صحيحة، يتم توجيه المستخدم للواجهة المناسبة لصلاحياته	السيناريو الرئيسي (Happy Path)
- اسم مستخدم/كلمة مرور غير صحيحة (يعيد النظام رسالة خطأ) - حساب غير مفعل (يظهر تنبيه "الحساب غير نشط")	الاستثناءات (Critical Scenario)



جدول (٣٠.٤): حالة الاستخدام _ استعادة كلمة المرور

مواصفات حالة الاستخدام	UC2	معرف حالة الاستخدام
اسم حالة الاستخدام	استعادة كلمة المرور	
<ul style="list-style-type: none"> - فاعل أساسي: جميع الأدوار (مدبرون، موظفون، عمالء) - فاعل داعم: النظام والبريد الإلكتروني يوفر خدمة إرسال رمز التحقق أو رابط إعادة التعيين للمستخدم. 		الجهات الفاعلة
عملية تسمح للمستخدم باستعادة كلمة المرور عند نسيانها عبر البريد الإلكتروني المسجل		الوصف
<ul style="list-style-type: none"> - وجود حساب مسجل للمستخدم - البريد الإلكتروني المسجل صالحًا ويمكن الوصول إليه. 		الشروط المسبقة

<ul style="list-style-type: none"> - توفر اتصال بالإنترنت 	
<ul style="list-style-type: none"> - تحديث كلمة المرور في قاعدة البيانات - إرسال تأكيد تغيير كلمة المرور بالبريد الإلكتروني 	الشروط اللاحقة
<ol style="list-style-type: none"> ١. ينقر المستخدم على "تسيت كلمة المرور" في صفحة تسجيل الدخول ٢. يدخل البريد الإلكتروني المسجل ٣. ينقر "إرسال رابط الاستعادة" ٤. يتلقى المستخدم بريداً إلكترونياً يحتوي على رابط فريد ٥. ينقر المستخدم على زر "إعادة تعين كلمة المرور" ٦. تفتح صفحة إعادة تعين كلمة المرور في المنصة ٧. يدخل كلمة مرور جديدة ٨. يؤكّد كلمة المرور الجديدة ٩. ينقر "تغيير كلمة المرور" ١٠. يتحقق النظام من صحة الرابط ويحدث كلمة المرور 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - البريد الإلكتروني غير مسجل (رسالة خطأ: "هذا البريد غير مسجل") - انتهاء صلاحية الرابط (رسالة: "انتهت صلاحية الرابط، أطلب رابطاً جديداً") - عدم تطابق كلمة المرور الجديدة مع التأكيد (رسالة: "كلمتا المرور غير متطابقتين") 	الاستثناءات (Critical Scenario)



جدول (٤،٤): حالة الاستخدام _٣ _إنشاء حساب جديد

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	معرف حالة الاستخدام
UC3	إنشاء حساب جديد
- الفاعل الرئيسي: العميل - الفاعل الداعم: النظام، خدمة البريد الإلكتروني	اسم حالة الاستخدام
إنشاء حساب مستخدم جديد في النظام	الجهات الفاعلة
عدم وجود حساب سابق بنفس البريد الإلكتروني أو اسم المستخدم.	الوصف
- إنشاء الحساب وإرسال بريد ترحبي للمستخدم. - تحديث users isActive=true في جدول	الشروط المسبقة
users	الشروط اللاحقة

<ol style="list-style-type: none"> ١. يقوم العميل بالدخول الى المنصة ٢. يضغط على زر إنشاء حساب جديد ٣. يقوم بتبينه البيانات المطلوبة في الاستماراة ٤. يضغط على زر إرسال ٥. يظهر تنبية باكتمال عملية إنشاء الحساب بنجاح ٦. تحديث حالة الحساب إلى "مفعل" ٧. توجيه المستخدم لصفحة تسجيل الدخول 	السيناريو الرئيسي (Happy Path)
الحساب موجود مسبقاً (توجيه لتسجيل الدخول)	الاستثناءات (Critical Scenario)

جدول (٤،٥) : حالة الاستخدام ؛ عرض قائمة الكراجات

مواصفات حالة الاستخدام	
UC4	معرف حالة الاستخدام
عرض قائمة الكراجات	اسم حالة الاستخدام
- الفاعل الرئيسي: العميل، الزائر - الفاعل الداعم: النظام	الجهات الفاعلة
عرض قائمة الكراجات المتوفرة في النظام مع تفاصيلها الأساسية.	الوصف
المستخدم مسجل دخول أو زائر. يجب أن تكون هناك كراجات مسجلة في جدول garages.	الشروط المسبقة
- تظهر قائمة الكراجات مع تفاصيلها.	الشروط اللاحقة

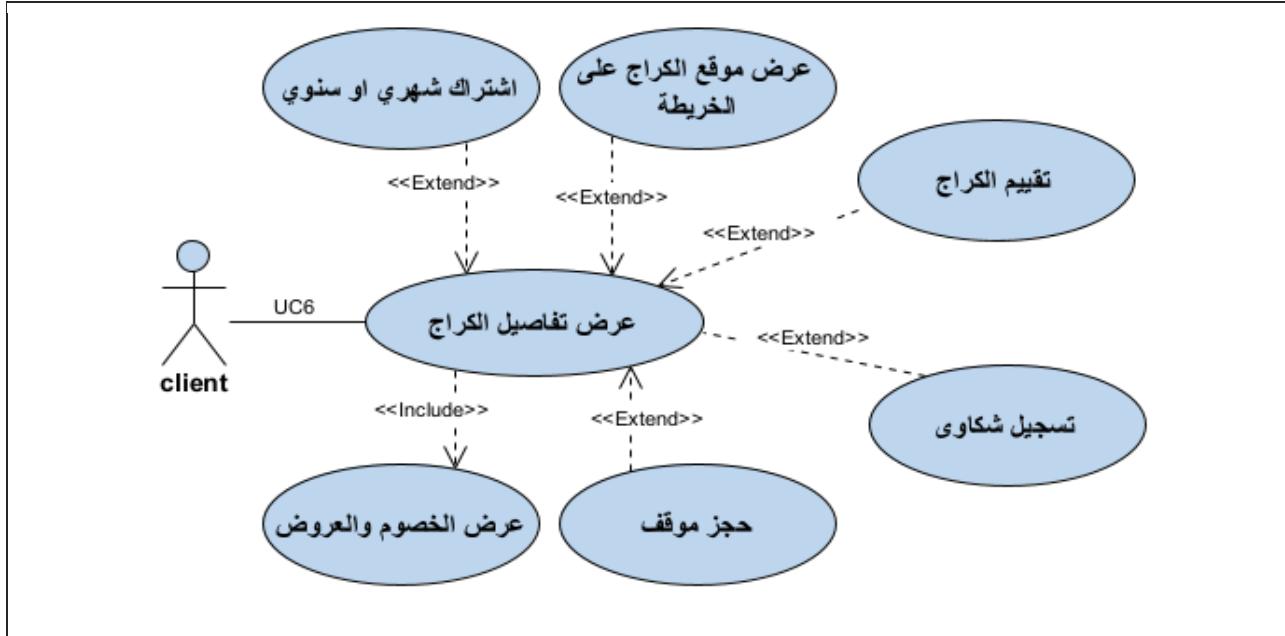
<p>- تمكين المستخدم من تصفية/ترتيب النتائج</p> <ol style="list-style-type: none"> ١. يفتح المستخدم قسم "استعراض الكراجات" ٢. يستلم النظام إحداثيات الموقع الحالي تلقائياً ٣. يعرض النظام قائمة الكراجات القريبة مرتبة حسب المسافة ٤. يعرض لكل كraj (الاسم والصورة المصغرة- متوسط التقييم- السعر/الساعة- الموقع على الخريطة - عدد الأماكن المتاحة) ٥. يمكن للمستخدم الفلترة والترتيب بالاعتماد على معيار آخر ٦. يمكن للمستخدم النقر على كراج لمشاهدة التفاصيل الكاملة 	<p>السيناريو الرئيسي (Happy Path)</p>
<p>-</p>	<p>الاستثناءات (Critical Scenario)</p>
<pre> classDiagram actor clientvisitor useCase UC4 useCase ShowParkingList useCase ShowParkingDetails useCase BookParking useCase ShowParkingPlaces clientvisitor -- UC4 : UC4 --> ShowParkingList ShowParkingList --><<Extend>>--> ShowParkingDetails ShowParkingList --><<Extend>>--> BookParking ShowParkingList --><<Extend>>--> ShowParkingPlaces </pre>	

جدول (٤): حالة الاستخدام ٥ _ عرض أماكن الكراجات

مواصفات حالة الاستخدام	
UC5	معرف حالة الاستخدام
عرض أماكن الكراجات	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام، خدمة الخرائط 	الجهات الفاعلة
عرض موقع الكراجات المتاحة مع تفاصيلها على الخريطة	الوصف
<ul style="list-style-type: none"> - تفعيل حساب المستخدم - وجود كراجات في النظام 	الشروط المسبقة
- تظهر موقع الكراجات للمستخدم.	الشروط اللاحقة
١. المستخدم يضغط على زر خيارات البحث ويختار عرض على الخريطة ٢. النظام يعرض الكراجات على الخريطة	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - لا توجد كراجات متاحة (رسالة "لا نتائج") - خطأ في تحميل البيانات (إعادة المحاولة) 	الاستثناءات (Critical Scenario)

جدول (٤،٨) : حالة الاستخدام ٦ _ عرض تفاصيل الكراج

مواصفات حالة الاستخدام	
UC6	معرف حالة الاستخدام
عرض تفاصيل الكراج	اسم حالة الاستخدام
- الفاعل الرئيسي: العميل - الفاعل الداعم: النظام	الجهات الفاعلة
عرض تفاصيل الكراج المحدد	الوصف
- تفعيل حساب المستخدم - وجود كراجمات في النظام	الشروط المسبقة
تظهر تفاصيل الكراج للمستخدم.	الشروط اللاحقة
١. يضغط المستخدم على زر الكراجمات ٢. النظام يعرض قائمة الكراجمات (الاسم، الموقع، التقييم) ٣. المستخدم يحدد كراجاً ٤. النظام يعرض التفاصيل (السعر/ساعة، السعة، الصور)	السيناريو الرئيسي (Happy Path)
- لا توجد كراجمات متوافحة (رسالة "لا نتائج") - خطأ في تحميل البيانات (إعادة المحاولة)	الاستثناءات (Critical Scenario)



جدول (٤،٩) : حالة الاستخدام ٧ _ جزء موقف

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	اسم حالة الاستخدام
UC7	جزء موقف
- الفاعل الرئيسي: العميل - الفاعل الداعم: النظام، خدمة الدفع	الجهات الفاعلة
جزء موقف شاغر وتأكيد العملية	الوصف
- وجود موقف شاغر - رصيد كافٍ في المحفظة	الشروط المسبقة
- خصم المبلغ من حساب المستخدم - تحديث حالة الموقف إلى "محجوز" وإرسال رمز QR للمستخدم.	الشروط اللاحقة

<ol style="list-style-type: none"> ١. يختار المستخدم الكراج ويدخل الوقت وطريقة الدفع ورقم اللوحة ٢. يضغط على إنشاء الحجز ٣. يتحقق النظام من صلاحية البيانات ٤. يدفع المبلغ المطلوب ٥. يتم تثبيت الحجز وإنشاء رمز QR 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - موقف غير متاح: رسالة خطأ - رصيد غير كافي: رسالة خطأ 	الاستثناءات (Critical Scenario)
<pre> graph LR Client((Client)) -- UC8 --> UC8[حجز موقف] UC8 -- "<<Extend>>" --> UCE1[لغاء الحجز] UC8 -- "<<Extend>>" --> UCE2[تعديل الحجز] UC8 -- "<<Include>>" --> UCI1[رسالة إشعارات تأكيد] UC8 -- "<<Include>>" --> UCI2[دفع إلكتروني] UC8 -- "<<Include>>" --> UCI3[توليد رموز فريدة] </pre>	

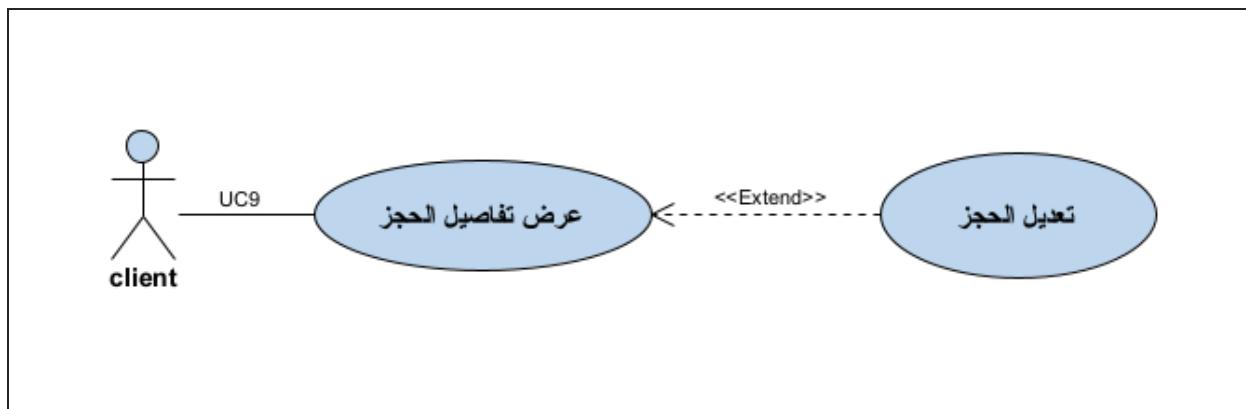
جدول (٤) : حالة الاستخدام ٨ _ عرض الحجوزات

مواصفات حالة الاستخدام	
UC8	معرف حالة الاستخدام
عرض الحجوزات	اسم حالة الاستخدام
-	الجهات الفاعلة

<ul style="list-style-type: none"> - الفاعل الداعم: النظام 	
عرض جميع الحجوزات للعميل مع تفاصيلها	الوصف
<ul style="list-style-type: none"> - تسجيل دخول العميل - وجود حجوزات سابقة 	الشروط المسبقة
-	الشروط اللاحقة
<ol style="list-style-type: none"> ١. يضغط المستخدم على زر الحجوزات من القائمة الجانبية ٢. يجلب النظام الحجوزات السابقة ٣. يعرض النظام قائمة الحجوزات مع بعض التفاصيل لكل حجز ٤. وبحسب حالة الحجز تظهر أزرار إجراءات إضافية 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - لا توجد حجوزات سابقة (رسالة: "لا توجد سجلات") - فشل تحميل البيانات (رسالة خطأ) 	الاستثناءات (Critical Scenario)
<pre> graph TD client((client)) -- UC8 --> UC8([عرض الحجوزات]) UC8 -- "<<Extend>>" --> UCTF([عرض تفاصيل الحجز]) UC8 -- "<<Extend>>" --> Tfu([تفع]) UC8 -- "<<Extend>>" --> UTD([تعديل]) UC8 -- "<<Extend>>" --> UGA([الغاء]) UC8 -- "<<Include>>" --> UCCS([تحقق من الصلاحية]) UC8 -- "<<Extend>>" --> UQR([عرض QR]) </pre>	

جدول (٤،١١): حالة الاستخدام ^٩ عرض تفاصيل الحجز

مواصفات حالة الاستخدام	
UC9	معرف حالة الاستخدام
عرض تفاصيل الحجز	اسم حالة الاستخدام
- الفاعل الرئيسي: العميل - الفاعل الداعم: النظام	الجهات الفاعلة
عرض معلومات الحجز (الموقع، التكلفة، الحالة وباقى التفاصيل)	الوصف
وجود حجز للمستخدم	الشروط المسبقة
-	الشروط اللاحقة
١. المستخدم يفتح "الحجوزات" ٢. النظام يعرض قائمة الحجوزات ٣. المستخدم يختار حجزاً ٤. النظام يُظهر كافة التفاصيل مثل (اسم الكراج، التوقيت، الموقع، الرمز ، التكلفة)	السيناريو الرئيسي (Happy Path)
خطأ في تحميل البيانات (إعادة المحاولة)	الاستثناءات (Critical Scenario)



جدول (٤،١٢): حالة الاستخدام ١٠ _ إدارة الحجوزات

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	معرف حالة الاستخدام
اسم حالة الاستخدام	إدارة الحجوزات
الجهات الفاعلة	- الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام
الوصف	إدارة جميع الحجوزات مع عرض تفاصيلها
الشروط المسبقة	- تسجيل دخول الموظف - وجود حجوزات سابقة
الشروط اللاحقة	-
السيناريو الرئيسي (Happy Path)	١. يضغط الموظف على زر الحجوزات من القائمة الجانبية ٢. يجلب النظام الحجوزات السابقة ٣. يعرض النظام قائمة الحجوزات مع بعض التفاصيل لكل حجز ٤. ولكل حجز يظهر زر الإجراءات الذي يعرض تفاصيل الحجز ومجموعة من الإجراءات تتغير بحسب حالة الحجز مثل

(تمديد، الغاء، تفعيل دخول يدوي، تفعيل خروج يدوي، تسجيل حادث، إعادة إصدار QR) - لا توجد حجوزات سابقة (رسالة: "لا توجد سجلات") - فشل تحميل البيانات (رسالة خطأ)	الاستثناءات (Critical Scenario)
<pre> classDiagram participant employee useCase UC10 useCase ManagementOfReservations useCase CancelReservation useCase QRGeneration useCase Search useCase Filtering employee --> UC10 : UC10 UC10 --> ManagementOfReservations : <<Extend>> UC10 --> CancelReservation : <<Extend>> UC10 --> QRGeneration : <<Extend>> UC10 --> Search : <<Include>> UC10 --> Filtering : <<Include>> ManagementOfReservations --> CancelReservation : <<Extend>> ManagementOfReservations --> QRGeneration : <<Extend>> ManagementOfReservations --> Search : <<Extend>> ManagementOfReservations --> Filtering : <<Extend>> CancelReservation --> HandwrittenEntryActivation CancelReservation --> HandwrittenExitActivation CancelReservation --> RecordingIncident </pre>	

جدول (١٢,٤) : حالة الاستخدام ١١ _ إلغاء الحجز

مواصفات حالة الاستخدام	
UC11	معرف حالة الاستخدام
إلغاء الحجز	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: المستخدم (عميل) والموظف - الفاعل الداعم: النظام 	الجهات الفاعلة
إلغاء حجز موجود وإعادة المبلغ وحسم نسبة محددة	الوصف

الشروط المسبقة	وجود حجز وقابل للإلغاء
الشروط اللاحقة	- تحرير الموقف - إعادة المبلغ مع حسم محدد
السيناريو الرئيسي (Happy Path)	١. المستخدم يعرض تفاصيل الحجز (UC05) ٢. ينقر "إلغاء" ٣. يؤكّد إلغاء ٤. النظام يُعيد المبلغ إلى المحفظة مع حسم رسوم الإلغاء ٥. إرسال تأكيد بإلغاء
الاستثناءات (Critical Scenario)	- تجاوز مهلة الإلغاء (استقطاع رسوم) - فشل في استرداد المبلغ (تنكير بالتواصل مع الدعم)

Sequence diagram illustrating the cancellation process:

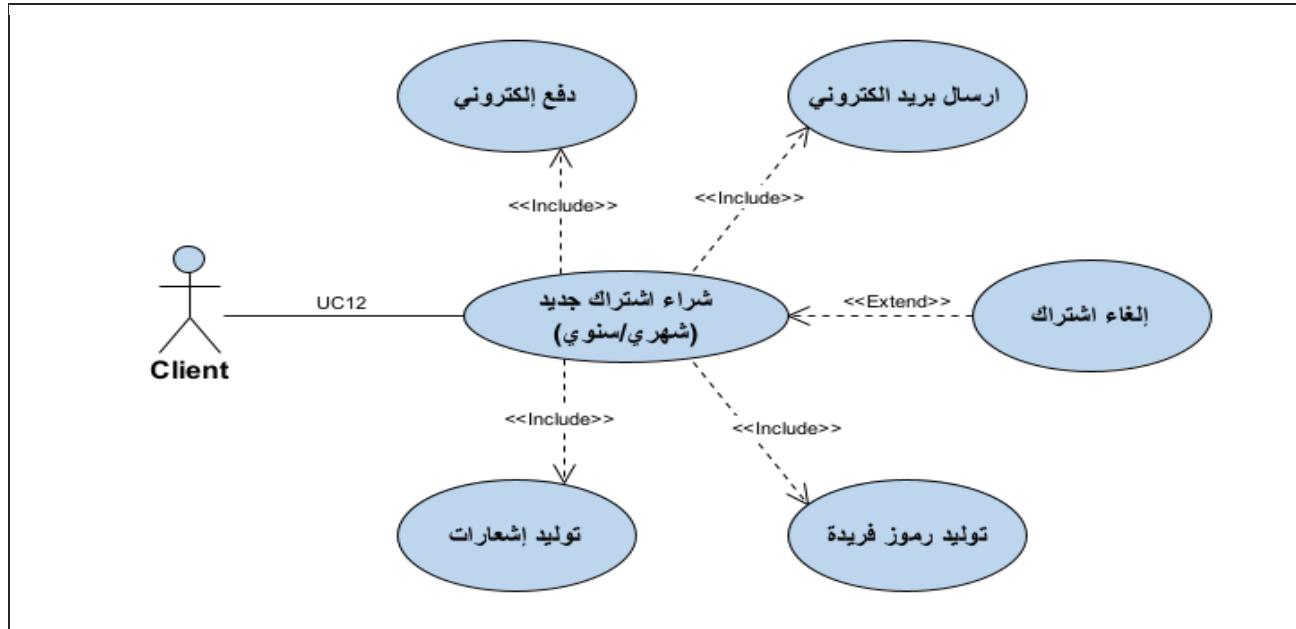
```

sequenceDiagram
    actor clientEmployee as client,employee
    participant UC11 as UC11
    participant cancel as إلغاء الحجز
    participant refund as دفع الكتروني (ارجاع المبلغ)
    client->>UC11: 
    activate UC11
    UC11->>cancel: 
    activate cancel
    cancel-->>refund: <<Include>>
    deactivate cancel
    deactivate UC11
    refund->>client: 
  
```

جدول (٤،١٣) : حالة الاستخدام ١٢ _ شراء اشتراك جديد

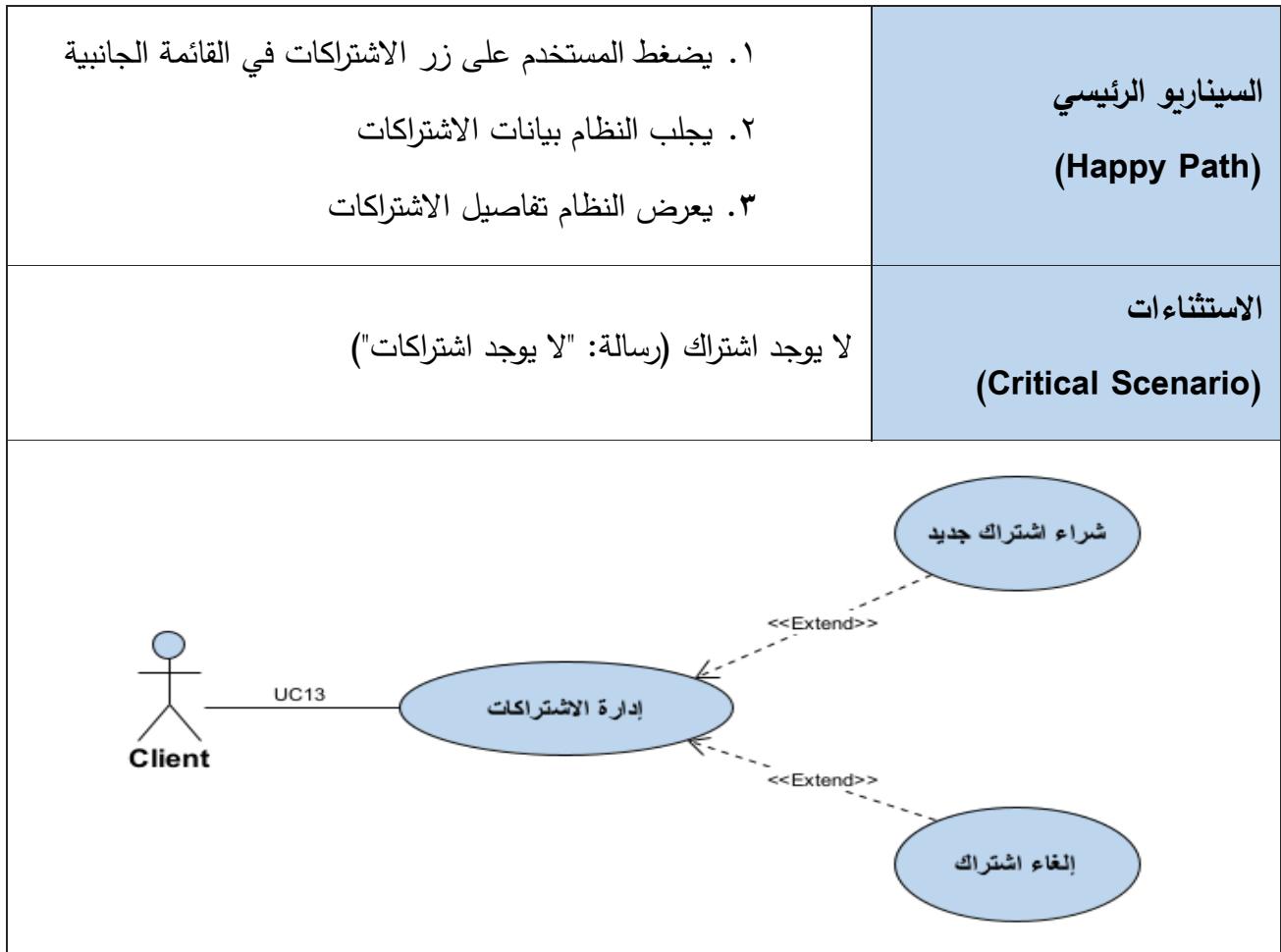
مواصفات حالة الاستخدام	
معرف حالة الاستخدام	UC12
اسم حالة الاستخدام	شراء اشتراك جديد

الجهات الفاعلة	
الوصف	
الشروط المسبقة	
<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام، خدمة الدفع 	
<ul style="list-style-type: none"> شراء اشتراك جديد في موقف معين 	
<ul style="list-style-type: none"> - تسجيل دخول العميل - رصيد كافي في المحفظة 	
<ul style="list-style-type: none"> - إرسال رمز QR. - خصم المبلغ من wallets - تحديث garages available_spots في available_spots - إرسال بريد الكتروني للمستخدم - توجيه اشعار للمستخدم ضمن النظام 	
<ol style="list-style-type: none"> ١. يختار المستخدم الكراج الذي يريد الاشتراك فيه ٢. يضغط على شراء اشتراك جديد ٣. يختار المستخدم نوع الاشتراك (شهري/سنوي) ٤. يتحقق النظام من الرصيد ٥. يدفع المستخدم المبلغ المطلوب ٦. يتم إنشاء الاشتراك وإرسال رمز QR 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - رصيد غير كاف (رسالة خطأ) - عدم توفر أماكن (رسالة: "الاشتراك غير متاح") 	الاستثناءات (Critical Scenario)



جدول (١٤،٤): حالة الاستخدام ١٣ _ إدارة الاشتراكات

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	UC13
اسم حالة الاستخدام	إدارة الاشتراكات
الجهات الفاعلة	<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام
الوصف	عرض وإدارة جميع الاشتراكات
الشروط المسبقة	<ul style="list-style-type: none"> - تسجيل دخول العميل - وجود اشتراك
الشروط اللاحقة	-



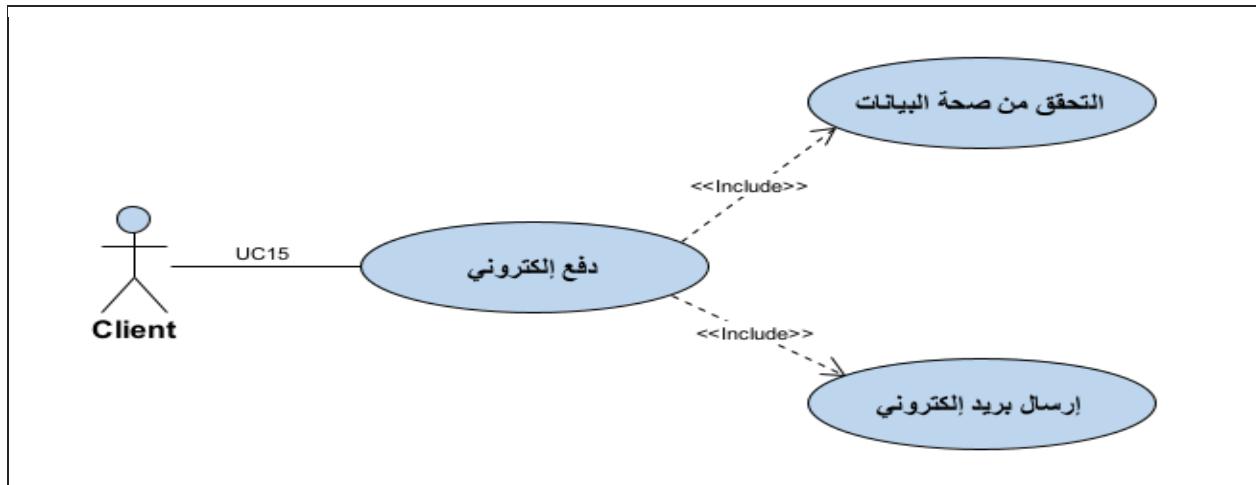
جدول (٤): حالة الاستخدام ٤ _ إلغاء اشتراك

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	نوع
UC14	تعريف
إلغاء اشتراك	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام 	الجهات الفاعلة
إلغاء الاشتراك النشط مع إرجاع المبلغ المتبقى.	الوصف

<ul style="list-style-type: none"> - تسجيل دخول العميل - وجود اشتراك نشط 	الشروط المسبقة
<ul style="list-style-type: none"> - تحديث subscriptions في status إلى "ملغي" - تحديث available_spots في garages - ارسال بريد الكتروني للمستخدم - توجيهه اشعار للمستخدم ضمن النظام 	الشروط اللاحقة
<ol style="list-style-type: none"> ١. العميل يفتح الاشتراكات من القائمة الجانبية ٢. يختار الاشتراك المطلوب ويضغط "إلغاء الاشتراك" ٣. يؤكّد الإلغاء ٤. النظام يحدّث الحالة ويرسل تأكيداً 	السيناريو الرئيسي (Happy Path)
-	الاستثناءات (Critical Scenario)
<pre> sequenceDiagram participant Client participant UC14 participant Cancel as إلغاء اشتراك participant Filter as الفترة والبحث Client->>UC14: UC14 activate UC14 UC14->>Cancel: activate Cancel UC14-->>Filter: <<Include>> deactivate Cancel deactivate UC14 </pre>	

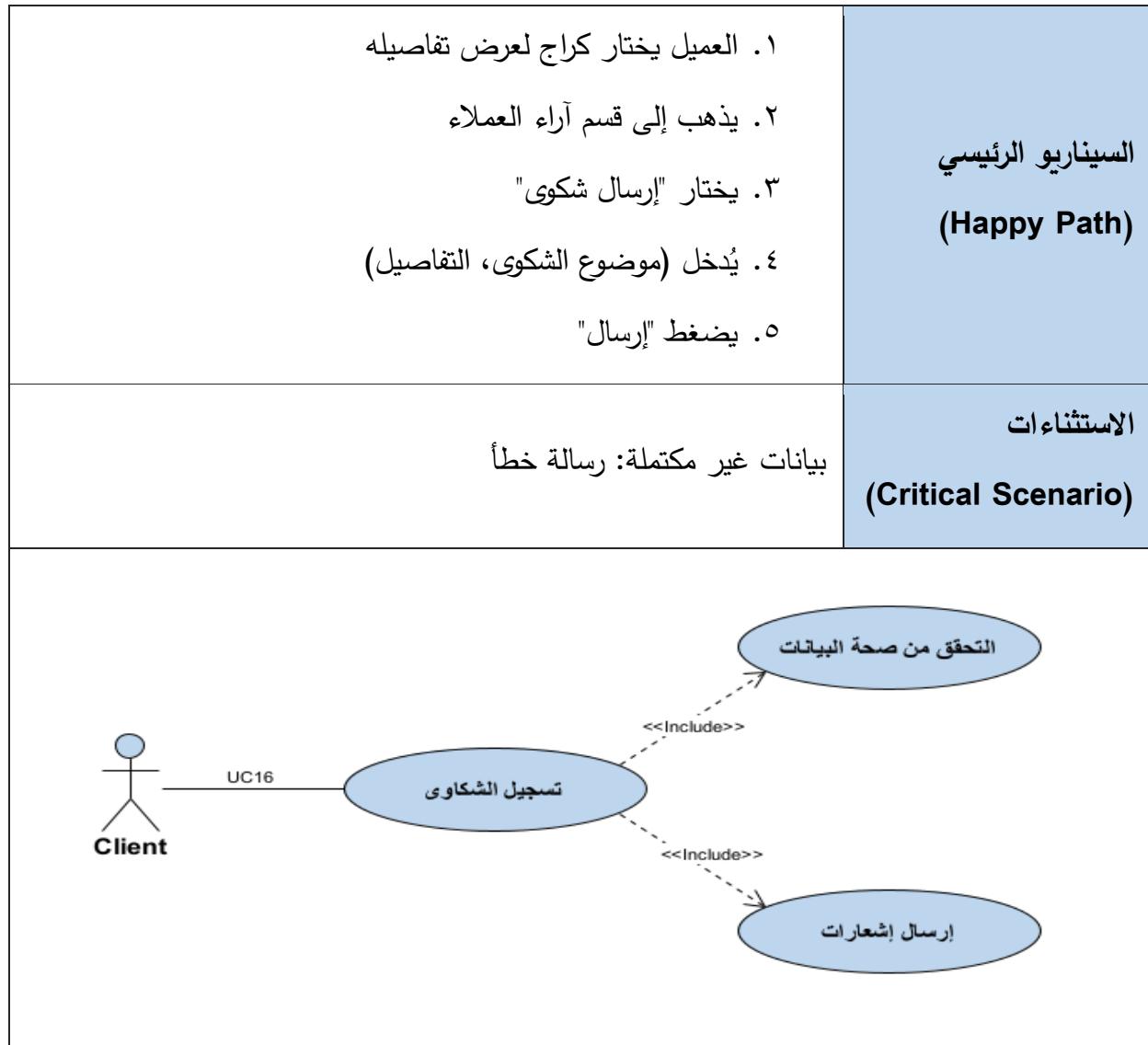
جدول (١٦،٤) : حالة الاستخدام ١٥ _ دفع إلكتروني

مواصفات حالة الاستخدام	
UC15	معرف حالة الاستخدام
دفع إلكتروني	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام، خدمة الدفع، خدمة البريد الإلكتروني 	الجهات الفاعلة
إنمام معاملات الدفع للحجوزات/الاشتراكات	الوصف
<ul style="list-style-type: none"> - تسجيل دخول العميل - وجود عملية شراء 	الشروط المسبقة
<ul style="list-style-type: none"> - تحديث payment_status - تسجيل المعاملة في transactions 	الشروط اللاحقة
<ol style="list-style-type: none"> ١. عند القيام بأي عملية تحتاج دفع يظهر زر الدفع ٢. يضغط العميل على زر الدفع ٣. يقوم بتأكيد العملية ٤. يتحقق النظام من وجود رصيد كافي في المحفظة ٥. النظام يُحدث الحالة ويرسل بريد إلكتروني للمستخدم 	السيناريو الرئيسي (Happy Path)
فشل المعاملة (إعادة المحاولة)	الاستثناءات (Critical Scenario)



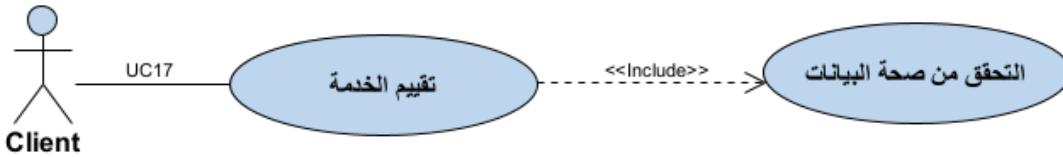
جدول (١٦،٤): حالة الاستخدام _ تسجيل الشكاوى

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	UC16
اسم حالة الاستخدام	تسجيل الشكاوى
الجهات الفاعلة	<ul style="list-style-type: none"> - الفاعل الرئيسي: العميل - الفاعل الداعم: النظام
الوصف	تقديم شكوى أو اقتراح حول الخدمة
الشروط المسبقة	تسجيل دخول العميل
الشروط اللاحقة	<ul style="list-style-type: none"> - إنشاء سجل شكوى في complaints - إخطار فريق الدعم



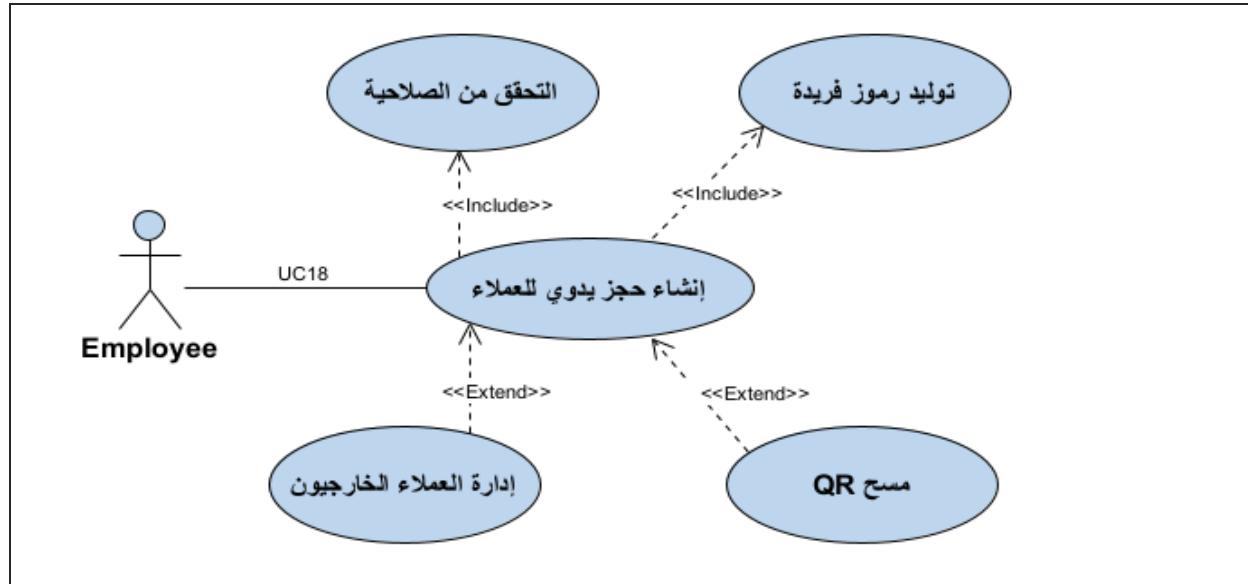
جدول (٤): حالة الاستخدام ١٧ _ تقييم الخدمة

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	اسم حالة الاستخدام
UC17	تعريف حالة الاستخدام
تقييم الخدمة	اسم حالة الاستخدام
- الفاعل الرئيسي: العميل - الفاعل الداعم: النظام	الجهات الفاعلة

تقييم تجربة المستخدم بعد انتهاء الحجز/الاشتراك	الوصف
<ul style="list-style-type: none"> - تسجيل دخول العميل - وجود حجز أو اشتراك منتهٍ 	الشروط المسبقة
<ul style="list-style-type: none"> - تسجيل التقييم في ratings - تحديث rating_average في garages 	الشروط اللاحقة
<ol style="list-style-type: none"> ١. العميل يختار كراج لعرض تفاصيله ٢. يذهب إلى قسم آراء العملاء ٣. يضغط على زر "قيم الكراج" ٤. العميل يختار حجز منتهي ٥. يعطي تقييماً (٥-١ نجوم) ٦. يدخل تعليقاً (اختياري) ٧. يضغط "إرسال" ٨. النظام يُحدث متوسط التقييمات 	السيناريو الرئيسي (Happy Path)
حجز غير مكتمل: رسالة خطأ	الاستثناءات (Critical Scenario)
 <pre> sequenceDiagram participant Client participant UC17 participant TC as تتحقق من صحة البيانات Client->>UC17: UC17 activate UC17 UC17-->>TC: <<Include>> deactivate UC17 </pre>	

جدول (٤١٩): حالة الاستخدام _١٨_ إنشاء حجز يدوٍ للعملاء

مواصفات حالة الاستخدام	
UC18	معرف حالة الاستخدام
إنشاء حجز يدوي للعملاء	اسم حالة الاستخدام
- الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام	الجهات الفاعلة
إنشاء حجز نيابة عن العميل عبر واجهة الموظفين.	الوصف
- توفر مواقف شاغرة.	الشروط المسبقة
- توليد QR فوري.	الشروط اللاحقة
١. يختار الموظف الحجوزات ٢. يضغط على إنشاء حجز يدوي ٣. البحث عن عميل موجود في النظام (الاسم/رقم الهاتف/البريد...) او إدخال اسم عميل خارجي. ٤. اختيار الموقف والزمن. ٥. إصدار الحجز للعميل مع الأخذ بعين الاعتبار عند حساب التكفة حالي العميل الخارجي والداخلي	السيناريو الرئيسي (Happy Path)
- عدم وجود صلاحية: رسالة خطأ	الاستثناءات (Critical Scenario)



جدول (٤): حالة الاستخدام ١٩ _ مسح QR

مواصفات حالة الاستخدام	
UC19	معرف حالة الاستخدام
مسح QR	اسم حالة الاستخدام
- الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام	الجهات الفاعلة
مسح رمز QR للتحكم في دخول/خروج المركبات.	الوصف
- وجود حجز نشط. - توفر كاميرا/قارئ QR.	الشروط المسبقة
- تسجيل وقت الدخول/الخروج. - احتساب الرسوم.	الشروط اللاحقة

١. يختار الموظف مسح QR من القائمة الجانبية. ٢. التتحقق من صلاحية الحجز.	السيناريو الرئيسي (Happy Path)
- QR غير صالح (عرض "الحجز منتهٍ أو غير موجود"). - محاولة دخول قبل الموعود (عرض تنبية).	الاستثناءات (Critical Scenario)

```

sequenceDiagram
    participant Employee
    participant QR
    participant CheckValidity
    Employee->>QR: UC19
    activate QR
    QR-->>CheckValidity: <<Include>>
    deactivate QR
    deactivate CheckValidity
  
```

جدول (٤) : حالة الاستخدام _٢٠ _إعادة إصدارQR

مواصفات حالة الاستخدام	
UC20	معرف حالة الاستخدام
إعادة إصدار QR	اسم حالة الاستخدام
- الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام	الجهات الفاعلة
إعادة إصدار رمز QR مفقود للعميل.	الوصف
وجود حجز صالح.	الشروط المسبقة
- إبطال الرمز القديم.	الشروط اللاحقة

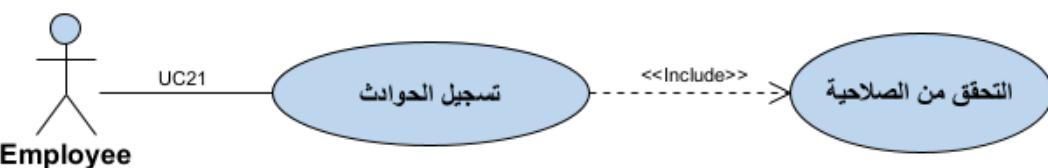
<ul style="list-style-type: none"> - إرسال رمز جديد. 	
<ol style="list-style-type: none"> ١. العميل يبلغ الموظف عن فقدان QR أثناء الدخول أو الخروج. ٢. يتحقق الموظف من صحة بيانات العميل. ٣. يضغط الموظف على زر إعادة إصدار QR 	السيناريو الرئيسي (Happy Path)
<p>الاستثناءات (Critical Scenario)</p> <p>الجز منتهٍ أو ملغى (عدم إصدار بديل).</p>	

```

    graph LR
        Employee((Employee)) --- UC20[UC20]
        UC20 -- "<<Include>>" --> UC21[إعادة إصدار QR]
        UC21 -- "<<Include>>" --> Check[التحقق من الصلاحية]
        UC21 -- "<<Include>>" --> Generate[توليد رموز فريدة]
    
```

جدول (٤،٢): حالة الاستخدام ٢١ _ تسجيل الحوادث

مواصفات حالة الاستخدام	
UC21	معرف حالة الاستخدام
تسجيل الحوادث	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام 	الجهات الفاعلة

الوصف	
توثيق الحوادث داخل الكراج (تلف مركبة، مشاكل أمنية).	
وجود حجز مرتبط بالحادث.	الشروط المسبقة
<ul style="list-style-type: none"> - إشعار الإدارة. - حفظ السجلات لأغراض التأمين. 	الشروط اللاحقة
<ol style="list-style-type: none"> ١. اختيار "تسجيل حادث" ٢. إدخال وصف الحادث (نص/صورة). 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - بيانات غير مكتملة (الزامية الحقول). 	الاستثناءات (Critical Scenario)
 <pre> sequenceDiagram participant Employee participant UC21 participant UC22 participant UC23 Employee->>UC21: UC21 activate UC21 UC21-->>UC22: <<Include>> activate UC22 UC22-->>UC23: deactivate UC22 deactivate UC21 </pre>	

جدول (٤) : حالة الاستخدام ٢٢ _ تفعيل الدخول اليدوي

مواصفات حالة الاستخدام	
UC22	معرف حالة الاستخدام
تفعيل الدخول اليدوي	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام 	الجهات الفاعلة

الوصف	
تفعيل دخول يدوی للعميل في حال مشاكل QR.	الشروط المسبقة
التحقق من صلاحية الحجز .	الشروط اللاحقة
تغيير حالة الحجز .	السيناريو الرئيسي (Happy Path)
١. يختار الموظف الحجز الذي يحتاج إلى تدخل ٢. يضغط على "تفعيل الدخول اليدوي".	الاستثناءات (Critical Scenario)
<pre> sequenceDiagram participant Employee participant UC22 participant UC23 Employee->>UC22: UC22 activate UC22 UC22-->>UC23: <<Include>> deactivate UC22 UC23-->>Employee: Response </pre>	

جدول (٤): حالة الاستخدام _ ٢٣ _ إدارة المواقف

مواصفات حالة الاستخدام	
UC23	معرف حالة الاستخدام
إدارة المواقف	اسم حالة الاستخدام
- الفاعل الرئيسي: الموظف - الفاعل الداعم: النظام	الجهات الفاعلة

الوصف	تغيير حالة موقف معين (متاح/تحت الصيانة/مغلق).
الشروط المسبقة	الموقف غير مشغول بحجز نشط.
الشروط اللاحقة	- تحديث حالة الموقف.
السيناريو الرئيسي (Happy Path)	1. يضغط الموظف على زر إدارة المواقف في القائمة الجانبية 2. يغير حالة الموقف المحدد
الاستثناءات (Critical Scenario)	- عدم وجود صلاحية: رسالة خطأ
<pre> sequenceDiagram participant Employee participant UC23 participant UC24 Employee->>UC23: UC23 activate UC23 UC23-->>UC24: <<Include>> deactivate UC23 deactivate UC24 </pre>	

جدول (٤، ٢٥): حالة الاستخدام ٢٤ ضبط ساعات العمل

مواصفات حالة الاستخدام		معرف حالة الاستخدام
UC24		اسم حالة الاستخدام
ضبط ساعات العمل		الفاعل الرئيسي: مدير الكراج
<ul style="list-style-type: none"> - الفاعل الداعم: النظام 		الجهات الفاعلة

الوصف	تحديد أو تعديل ساعات عمل الكراج اليومية
الشروط المسبقة	صلاحية إدارة الكراج.
الشروط اللاحقة	تطبيق التغيير على نظام الحجز.
السيناريو الرئيسي (Happy Path)	<ol style="list-style-type: none"> ١. يضغط مدير الكراج على زر ساعات العمل في القائمة الجانبية ٢. تعديل الجداول الزمنية. ٣. حفظ الإعدادات.
الاستثناءات (Critical Scenario)	<ul style="list-style-type: none"> - تعارض مع حجوزات مستقبلية (عرض قائمة بالحجوزات المتأثرة).
<pre> graph LR participant[garage admin] --- UC24[UC24] UC24 --> UC25[ضبط ساعات العمل] UC25 -.-> UC26[تحقق من الصلاحية] style UC25 fill:#e0f2fd style UC26 fill:#e0f2fd </pre>	

جدول (٤): حالة الاستخدام _ ٢٥ _ إدارة المحافظ

مواصفات حالة الاستخدام	
UC25	معرف حالة الاستخدام
إدارة المحافظ	اسم حالة الاستخدام
- الفاعل الرئيسي: مدير الكراج	الجهات الفاعلة

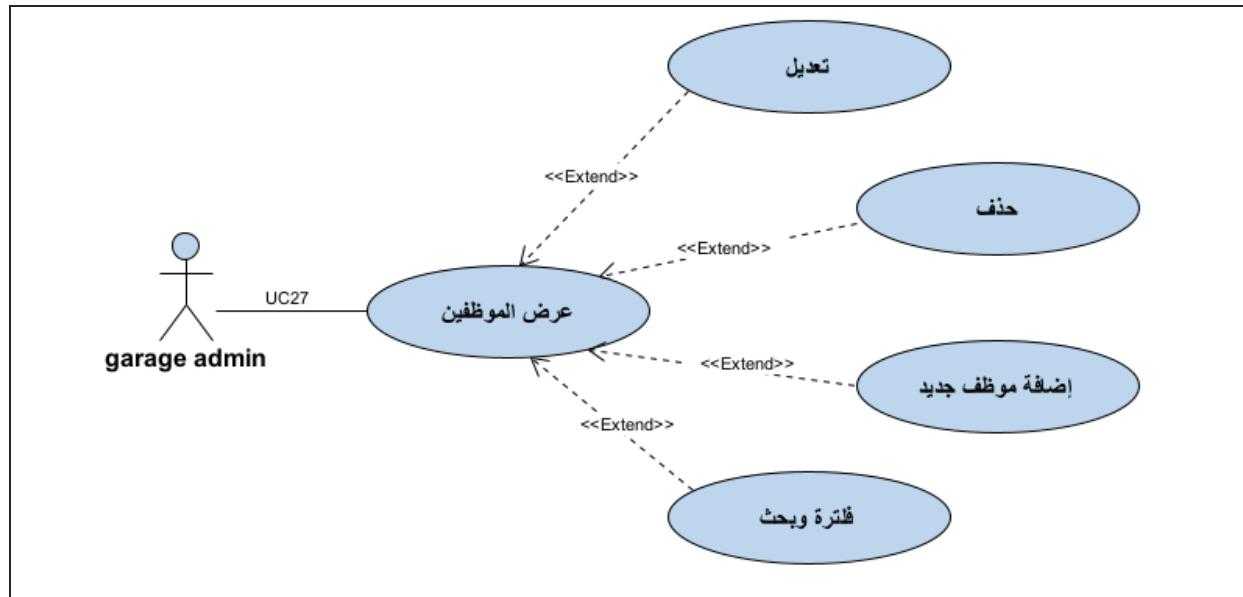
-	الفاعل الداعم: النظام	
	إنشاء محفظة إلكترونية للعميل.	الوصف
	تسجيل دخول لمدير الكراج	الشروط المسبقة
	-	الشروط اللاحقة
	١. يختار مدير الكراج إدارة المحافظ من القائمة الجانبية. ٢. يتم عرض محفظة العميل ٣. يمكن للمدير البحث عن محفظة باسم العميل او البريد الإلكتروني ويمكن له إنشاء محفظة جديدة أو تعطيل محفظة موجودة أو إيداع رصيد فيها	السيناريو الرئيسي (Happy Path)
	-	الاستثناءات (Critical Scenario)
	<pre> useCaseDiagram participant GarageAdmin as garage admin participant ManagementOfWallet as "ادارة المحفظة" participant CreateNewWallet as "إنشاء محفظة جديدة" participant DeactivateWallet as "تعطيل محفظة" participant DepositAmount as "إيداع رصيد" GarageAdmin -- UC25 --> ManagementOfWallet ManagementOfWallet -- <<Extend>> -- CreateNewWallet ManagementOfWallet -- <<Extend>> -- DeactivateWallet ManagementOfWallet -- <<Extend>> -- DepositAmount </pre>	

جدول (٤): حالة الاستخدام ٢٦ _ تحديد الأسعار (الاشتراكات المؤقتة)

مواصفات حالة الاستخدام	
UC26	معرف حالة الاستخدام
تحديد الأسعار (الاشتراكات المؤقتة)	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: مدير الكراج - الفاعل الداعم: النظام 	الجهات الفاعلة
تحديد أسعار الاشتراكات والحجوزات المؤقتة.	الوصف
عدم وجود اشتراكات نشطة تتأثر بالتغيير.	الشروط المسبقة
تطبيق الأسعار الجديدة على الحجوزات الجديدة.	الشروط اللاحقة
<ol style="list-style-type: none"> ١. الدخول إلى "تسغير الموقف". ٢. اختيار نوع الموقف. ٣. إدخال السعر الجديد. ٤. تحديد نطاق زمني (إن وجد). ٥. حفظ التعديلات. 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - أسعار غير منطقية (تنبيه بالمقارنة مع المنطقة). - تعارض مع عروض سابقة. 	الاستثناءات (Critical Scenario)
<pre> sequenceDiagram actor garageAdmin as garage admin useCase uc26 as UC26 useCase auth as "تحقق من الصلاحية" garageAdmin->>uc26: UC26 activate uc26 uc26-->>auth: <<Include>> deactivate uc26 deactivate garageAdmin </pre>	

جدول (٤،٢٨): حالة الاستخدام _ عرض الموظفين

مواصفات حالة الاستخدام	
UC27	معرف حالة الاستخدام
عرض الموظفين	اسم حالة الاستخدام
- الفاعل الرئيسي: مدير الكراج - الفاعل الداعم: النظام	الجهات الفاعلة
عرض الموظفين في الكراج	الوصف
مدير الكراج مسجل دخول مسبقا. صلاحيّة إدارة الموظفين.	الشروط المسبقة
-	الشروط اللاحقة
١. من الواجهة الرئيسية لمدير الكراج يختار " الموظفين" من القائمة الجانبية. ٢. يتم عرض موظفي الكراج على شكل قائمة مع تفاصيلهم	السيناريو الرئيسي (Happy Path)
-	الاستثناءات (Critical Scenario)



جدول (٤،٢٩): حالة الاستخدام _٢٨ _ إضافة موظفين جدد

مواصفات حالة الاستخدام	
UC28	معرف حالة الاستخدام
إضافة موظفين جدد	اسم حالة الاستخدام
- الفاعل الرئيسي: مدير الكراج - الفاعل الداعم: النظام	الجهات الفاعلة
تسجيل موظف جديد في النظام وربطه بکراج معين.	الوصف
صلاحيّة إدارة الموظفين.	الشروط المسبقة
- إنشاء وتفعيل الحساب.	الشروط اللاحقة

<ol style="list-style-type: none"> ١. اختيار "الموظفين" من القائمة الجانبية. ٢. يضغط على إضافة موظف جديد ٣. تعبئة النموذج (الاسم، البريد، الجوال...). ٤. تعيين الصلاحيات. ٥. الضغط على زر إضافة. 	السيناريو الرئيسي (Happy Path)
<ul style="list-style-type: none"> - بيانات غير مكتملة (إبراز الحقول المطلوبة). - البريد مستخدم مسبقاً (طلب إدخال جديد). 	الاستثناءات (Critical Scenario)
<pre> graph LR garageAdmin[garage admin] -- UC28 --> addEmployee[اضافة موظف جديد] addEmployee -- <<Include>> --> auth[تحقق من الصلاحية] addEmployee -- <<Include>> --> email[إرسال إيميلات] </pre>	

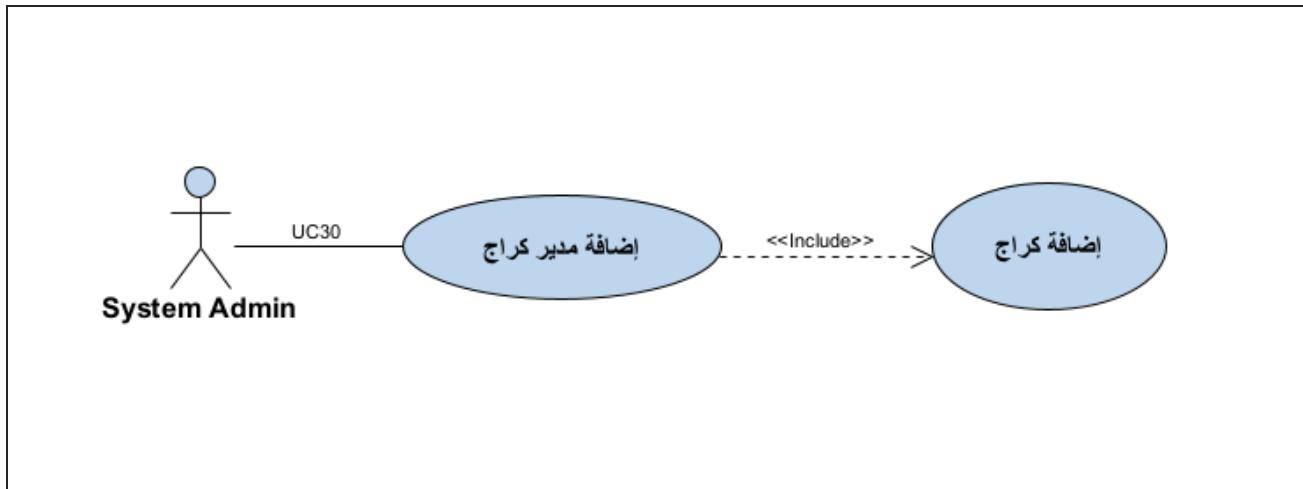
جدول (٤٠، ٤٩) : حالة الاستخدام _ ٢٩ إدارة حسابات المستخدمين

مواصفات حالة الاستخدام	
UC29	معرف حالة الاستخدام
ادارة حسابات المستخدمين	اسم حالة الاستخدام
<ul style="list-style-type: none"> - الفاعل الرئيسي: مدير النظام - الفاعل الداعم: النظام 	الجهات الفاعلة

الوصف	إنشاء/تعديل/تعطيل حسابات المستخدمين في النظام
الشروط المسبقة	<ul style="list-style-type: none"> - صلاحية مدير عام - تسجيل الدخول
الشروط اللاحقة	<ul style="list-style-type: none"> - تحديث بيانات users في قاعدة البيانات - تسجيل العملية في audit_logs
السيناريو الرئيسي (Happy Path)	<ol style="list-style-type: none"> ١. فتح "إدارة المستخدمين" في القائمة الجانبية. ٢. يتم عرض المستخدمين الحاليين في النظام على شكل قائمة مع إمكانية التعديل والحذف ٣. يمكن للمدير إضافة مستخدم جديد باختيار "إضافة مستخدم جديد" ٤. يقوم بتبثة بيانات الحساب الجديد (...role, isActive) ٥. يضغط على زر حفظ
الاستثناءات (Critical Scenario)	<ul style="list-style-type: none"> - بيانات غير مكتملة (إظهار أخطاء التحقق) - محاولة تعديل صلاحيات غير مصرحة - الضغط على زر إلغاء
<pre> graph LR Admin[system admin] --- UC29[UC29] UC29 --> UserManagement([ادارة حسابات المستخدمين]) UserManagement -.-> AuthorityCheck([التحقق من الصلاحية]) </pre>	

جدول (٤، ٣١) : حالة الاستخدام _ إضافة كراج

مواصفات حالة الاستخدام	
UC30	معرف حالة الاستخدام
إضافة كراج	اسم حالة الاستخدام
- الفاعل الرئيسي: مدير النظام، مدير الكراج - الفاعل الداعم: النظام	الجهات الفاعلة
تسجيل كراج جديد في النظام مع تفاصيل الموقع والسعنة.	الوصف
توفر معلومات الكراج (الاسم، الموقع، السعة). صلاحيّة إدارة الكراجات.	الشروط المسبقة
- تفعيل الكراج للحجوزات.	الشروط اللاحقة
١. يقوم المدير العام بالضغط على إدارة المستخدمين في القائمة الجانبيّة. ٢. يقوم بالضغط على زر إضافة مستخدم جديد. ٣. يقوم بملء البيانات ويحدد الدور مدير كراج. ٤. يقوم مدير الكراج بتسجيل الدخول من حسابه الخاص حسب الدور. ٥. عند النجاح في تسجيل الدخول إلى حسابه تظهر واجهة تفید بان المدير لا يملك كراج وعليه ملء النموذج لثبت بيانات الكراج الخاص به. ٦. ثم يضغط على زر إنشاء الكراج.	السيناريو الرئيسي (Happy Path)
-	الاستثناءات (Critical Scenario)



جدول (٤، ٣٢): حالة الاستخدام _٣١ _إضافة صورة للكراج

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	اسم حالة الاستخدام
UC31	إضافة صورة للكراج
الفاعل الرئيسي: مدير الكراج الفاعل الداعم: النظام	الجهات الفاعلة
رفع صور للكراج لعرضها في واجهة العميل.	الوصف
- وجود الكراج في النظام. - توفر الصور بصيغ مدعومة.	الشروط المسبقة
ظهور الصور في واجهة حجز العميل.	الشروط اللاحقة

١. يضغط مدير الكراج على إدارة الصور في القائمة الجانبية. ٢. يضغط على "اختيار ملف". ٣. تحميل الصور من الجهاز. ٥. يضغط على رفع.	السيناريو الرئيسي (Happy Path)
- صيغة غير مدعومة (عرض قائمة الصيغ المقبولة). - حجم ملف كبير (ضغط تلقائي). - عدم وجود صلاحية: رسالة خطأ	الاستثناءات (Critical Scenario)
 <pre> graph LR GA((Garage Admin)) -- UC31 --> AP((إضافة صورة للكراج)) AP -- "<<Include>>" --> CP((تحقق من الصلاحية)) </pre>	

جدول (٤) : حالة الاستخدام ٣٢ _ الاطلاع على التقارير المقدمة

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	تعريف حالة الاستخدام
UC32	الاطلاع على التقارير المقدمة
الفاعل الرئيسي: مدير الكراج الفاعل الداعم: النظام	اسم حالة الاستخدام الجهات الفاعلة
توليد تقارير إحصائية عن أداء الكراج (الإيرادات، الإشغال، الحوادث).	الوصف

الشروط المسبقة	وجود بيانات تاريخية.
الشروط اللاحقة	عرض التقارير.
١. يختار المدير "التقارير المتقدمة" من القائمة الجانبية. ٢. يمكن أن يقوم بالفلترة حسب الفترة (يومي/أسبوعي/شهري/سنوي). ٣. تظهر قائمة بالنتائج حسب نوع التصنيف للمعلومات (معدل الإلغاء-نوع الموقف-الإيرادات.....)	السيناريو الرئيسي (Happy Path)
- عدم وجود بيانات (عرض رسالة). - معايير غير صالحة (تببيه).	الاستثناءات (Critical Scenario)
 <pre> graph LR GA[Garage Admin] --- UC32([التقرير المتقدمة]) UC32 --> UC33([تحقق من الصلاحية]) style UC32 fill:#ADD8E6 style UC33 fill:#ADD8E6 </pre>	

جدول (٤.٣): حالة الاستخدام ٣٣ _ مخططات الحجوزات والإيرادات

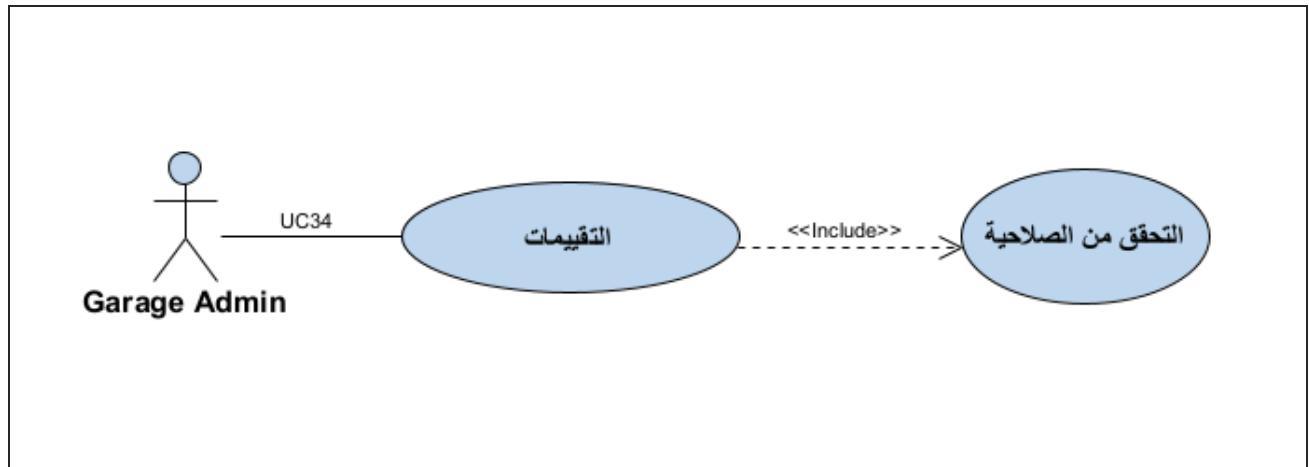
مواصفات حالة الاستخدام	
UC33	معرف حالة الاستخدام
مخططات الحجوزات والإيرادات	اسم حالة الاستخدام
- الفاعل الرئيسي: مدير الكراج	الجهات الفاعلة

-	الفاعل الداعم: النظام	
عرض مخطط يوضح النسب للحجوزات والإيرادات مما يساهم في تحليل النظام		الوصف
وجود حجوزات مسجلة في النظام.		الشروط المسبقة
تحديث المخططات حسب آخر إحصائية للحجوزات.		الشروط اللاحقة
١. اختيار "المخططات" من القائمة الجانبية. ٢. تحديد الفترة التي يمكن عرض المخطط على حسب نتائجها ٣. (يومي/ أسبوعي/ شهري/ سنوي) ويمكن تحديد مدة زمنية. ٤. تعرض مخططات تحليلية توضح النتائج حسب الفلترة المطلوبة.		السيناريو الرئيسي (Happy Path)
-	لا توجد بيانات في الفترة المحددة (توسيع النطاق الزمني).	الاستثناءات (Critical Scenario)



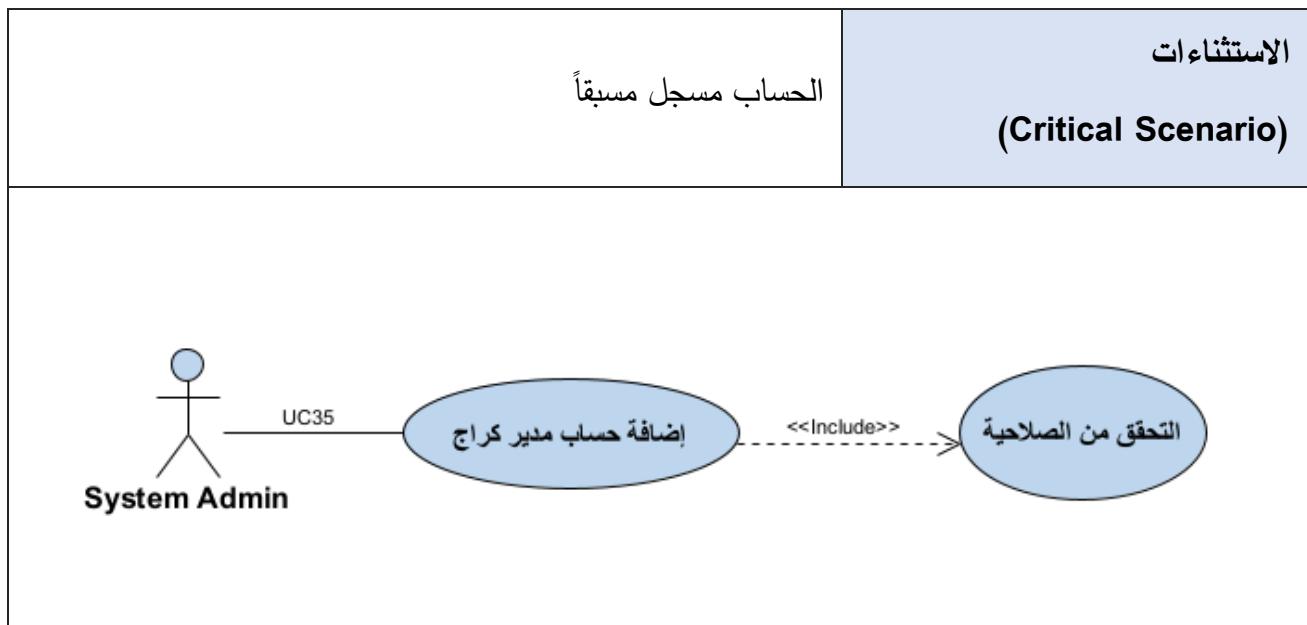
جدول (٤) : حالة الاستخدام ٣٤ _ عرض التقييم للخدمات

مواصفات حالة الاستخدام	
UC34	معرف حالة الاستخدام
عرض التقييم للخدمات	اسم حالة الاستخدام
الفاعل الرئيسي: مدير الكراج الفاعل الداعم: النظام	الجهات الفاعلة
عرض متوسط تقييمات الخدمات لكل كراج وتفاصيل التقييمات الفردية	الوصف
- تسجيل الدخول - وجود تقييمات مسجلة في النظام	الشروط المسبقة
تحديث متوسط التقييمات تلقائياً	الشروط اللاحقة
١. يفتح صفحة "التقييمات" من القائمة الجانبية ٣. يعرض قائمة الكراجات مع متوسط التقييم ٤. عند اختيار كراج، يعرض التقييمات التفصيلية من جدول ratings	السيناريو الرئيسي (Happy Path)
لا توجد تقييمات (عرض رسالة "لا توجد بيانات")	الاستثناءات (Critical Scenario)



جدول (٣٦.٤) : حالة الاستخدام ٣٥ _ إضافة حساب مدير كراج

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	عنوان
UC35	إضافة حساب مدير كراج
الفاعل الرئيسي: مدير النظام الفاعل الداعم: النظام	الجهات الفاعلة
إنشاء حساب خاص بمدير كراج جديد	الوصف
- صلاحية مدير عام	الشروط المسبقة
- إضافة مستخدم جديد في users	الشروط اللاحقة
١. يختار مدير النظام "إدارة المستخدمين" من القائمة الجانبية يضغط على إضافة مستخدم جديد ٢. تعبئة بيانات الحساب ويحدد الدور "مدير كراج" يضغط على حفظ	السيناريو الرئيسي (Happy Path)



جدول (٤) : حالة الاستخدام ٣٦ _ تسجيل الخروج من النظام

مواصفات حالة الاستخدام	
معرف حالة الاستخدام	بيانات حالة الاستخدام
UC36	اسم حالة الاستخدام
تسجيل الخروج من النظام	الجهات الفاعلة
الفاعل الرئيسي: جميع المستخدمين الفاعل الداعم: النظام	الوصف
إنهاء جلسة المستخدم بشكل آمن	الشروط المسبقة
- جلسة مستخدم نشطة - تسجيل دخول مسبق	الشروط اللاحقة
- إنهاء الجلسة. - تسجيل التاريخ في audit_logs	

١. النقر على "تسجيل الخروج" ٢. إلغاء تفويض الوصول ٣. توجيه المستخدم للواجهة الرئيسية	السيناريو الرئيسي (Happy Path)
عدم وجود جلسة نشطة: توجيه لصفحة تسجيل الدخول	الاستثناءات (Critical Scenario)
<pre> graph LR Actor((Actor)) -- UC36 --> UC36([تسجيل الخروج من النظام]) UC36 -- <<Extend>> --> Login([تسجيل الدخول]) UC36 -- <<Extend>> --> CreateAccount([إنشاء حساب]) </pre>	

٤- مخططات التدفق:

سنقوم بعرض بعض المخططات التدفقية التي توضح سير العمليات داخل النظام.

تعتبر المخططات التدفقية أدوات فعالة في تحليل العمليات وتبسيطها، ولها العديد من الفوائد المهمة، منها:

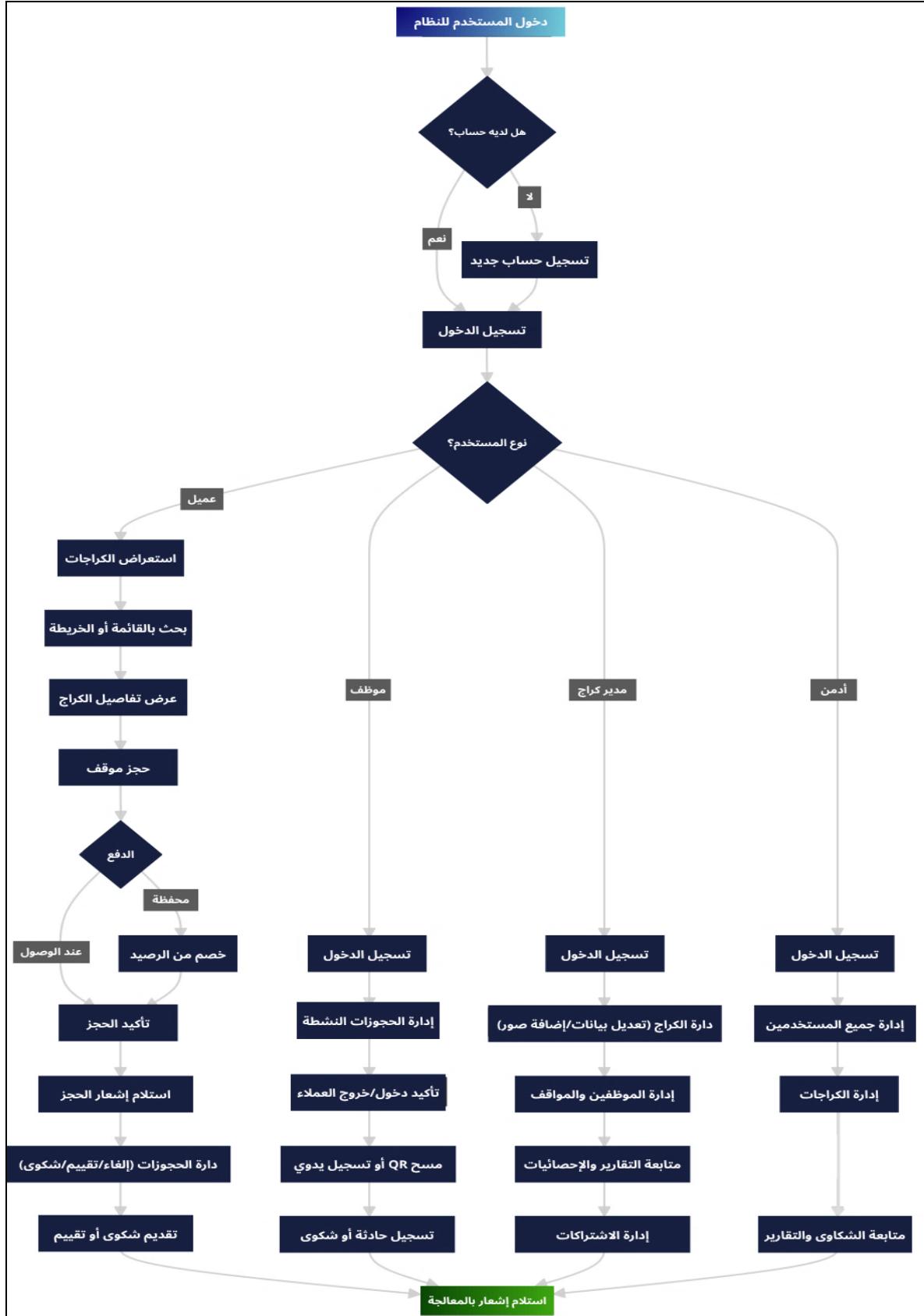
- تحسين الفهم: تساعد المخططات التدفقية في توضيح العمليات المعقدة من خلال تقديم تمثيل مرجعي يسهل فهمه، مما يعزز من قدرة الأفراد على استيعاب الخطوات المختلفة في العملية.

- تسهيل التواصل: تُستخدم المخططات كأداة تواصل فعالة بين الفرق المختلفة، حيث تتيح للأعضاء فهم العمليات بشكل أسرع وأوضح، مما يقلل من الحاجة للاجتماعات المطولة.

- تحديد المشكلات: من خلال تحليل المخطط، يمكن تحديد نقاط الضعف أو الأخطاء المحتملة في العملية، مما يسهل اتخاذ القرارات المناسبة لتحسين الأداء.
- زيادة الكفاءة: تساعد المخططات في تحديد الخطوات غير الضرورية أو المكررة، مما يساهم في تحسين كفاءة العمليات وتقليل الهدر.
- تسهيل اتخاذ القرارات: توفر المخططات تدفقاً منطقياً للعمليات، مما يساعد الأفراد على اتخاذ قرارات مستنيرة بناءً على الخيارات المتاحة.

١٧.٤ - مخطط تفصيلي للعلاقات بين مكونات النظام:

إعداد مخطط تفصيلي يوضح العلاقات بين المكونات المختلفة للنظام حسب دور كل مستخدم:



الشكل (٤ .٤) – العلاقات بين المكونات المختلفة للنظام حسب دور كل مستخدم

- تحليل تدفق العمليات وفق المخطط السابق:

يتمثل المخطط التدقي نظاماً متكاملاً لإدارة كراجات السيارات يخدم أربعة أدوار مستخدم رئيسية: العميل، مدير الكراج، الموظف، ومدير النظام. يبدأ المسار بدخول أي مستخدم للنظام، حيث يُوجه المستخدمون الجدد إلى عملية تسجيل حساب جديد بينما يقوم المستخدمون المسجلون بتسجيل الدخول مباشرة.

دور العميل: بعد تسجيل الدخول، يبدأ المستأجر بالبحث عن كراجات متاحة باستخدام القوائم أو الخرائط. عند تحديد كراج معين، يعرض تفاصيله ثم ينتقل إلى عملية حجز الموقف. يتضمن الحجز اختيار طريقة الدفع، حيث يظهر خيار استخدام المحفظة الإلكترونية لخصم المبلغ مباشرةً من رصيد المستخدم. بعد إتمام الدفع بنجاح، يتلقى المستأجر تأكيداً بالحجز وإشعاراً رقمياً. عند الوصول الفعلي للكراج، يقوم بإتمام عملية الوصول إما بمسح رمز QR الخاص به أو عبر التسجيل اليدوي في نقطة الاستقبال. في نهاية التجربة، يُتاح للعميل تسجيل تقييم أو تقديم شكوى.

دور الموظف: يتمحور دور الموظف حول المهام التنفيذية اليومية داخل الكراج. حيث يقوم بتأكيد دخول أو خروج المعدات المرتبطة بعمل الكراج مثل البوابات الآلية أو أنظمة الدفع. يشرف بشكل مباشر على تنفيذ عملية الوصول للمستأجرين عند حضورهم، سواء عبر مسح رمز QR الخاص بهم أو من خلال التسجيل اليدوي في النظام. كما يلتزم الموظف بتسجيل أي حوادث أو شكاوى تبلغ مباشرةً في الموقع ويتلقى هو أيضاً إشعارات بالحوادث الطارئة للتعامل المباشر معها.

دور مدير الكراج: بعد تسجيل الدخول، يركز مدير الكراج على المهام الإدارية حيث يقوم بإدارة بيانات الكراج الأساسية وتحديثها أو إضافة الصور التوضيحية. يدير أيضاً الاشتراكات الدورية للعملاء المنتظمين ويستقبل الشكاوى والتقارير المقدمة من المستأجرين لمعالجتها.

دور مدير النظام: يمتلك مدير النظام صلاحيات الرقابة الشاملة على النظام ككل. حيث يدير حسابات المستخدمين في النظام من إنشاء وتعديل وتعطيل ويطبع على التقارير.

٤-٢.٧.٤ - مخطط تدقي لعملية تسجيل الدخول:

إعداد مخطط تفصيلي يوضح العلاقات بين المكونات المختلفة لتسجيل الدخول:



الشكل (٤.٥) – مخطط تدفقي لتسجيل الدخول للمستخدم

- تحليل تدفق عملية تسجيل الدخول وفق المخطط السابق:

يمثل المخطط التدفقى آلية تسجيل الدخول الموحدة لنظام إدارة الكراجات، والتي تخضع لها جميع الأدوار الأربع: العميل، مدير الكراج، الموظف، ومدير النظام.

تبدأ العملية عندما يفتح المستخدم صفحة تسجيل الدخول الخاصة بالنظام، حيث يقوم بإدخال بيانات الاعتماد المطلوبة وهي البريد الإلكتروني وكلمة المرور في الحقول المخصصة.

بعد الضغط على زر "تسجيل الدخول"، تقوم واجهة المستخدم الأمامية (Frontend) بتنشيط خدمة (apiService.login(credentials)) التي ترسل بدورها طلباً من نوع POST إلى المسار /auth/login/ في الخادم (Backend). يستقبل هذا الطلب في الخلفية وحدة التحكم AuthController التي تستدعي دالة login المخصصة للمصادقة.

تضمن عملية المعالجة في الخادم مرحلتين أساسيتين:

- أولاً: التحقق من صحة البيانات (Validation) للتأكد من تطابق تنسيق البريد الإلكتروني ومتطلبات كلمة المرور.
- ثانياً: التتحقق من مطابقة البيانات في قاعدة البيانات، حيث يقارن البريد الإلكتروني مع الحسابات المسجلة، وتحفّص كلمة المرور المقابلة للتأكد من صحتها.

إذا كانت المعلومات غير متطابقة مع السجلات، يرد الخادم برسالة خطأ واضحة تقيد بعدم صحة البيانات، والتي تعرضها واجهة المستخدم للمستخدم مباشرةً. أما في حال نجاح التتحقق، ينشئ الخادم رمز دخول (JWT) مشفرًا يحتوي على بيانات المستخدم الأساسية، ويرسله مع تفاصيل الدور (العميل/مدير كراج/موظفي/إداري) كاستجابة ناجحة.

تقوم الواجهة الأمامية باستلام الاستجابة الناجحة وت تخزين رمز الدخول (Token) في مكان آمن مثل localStorage أو سياق التطبيق (Context). أخيراً، يُوجّه المستخدم تلقائياً إلى لوحة التحكم المناسبة لدوره في النظام، حيث يبدأ كل مستخدم بمهامه الخاصة حسب صلاحياته المحددة.

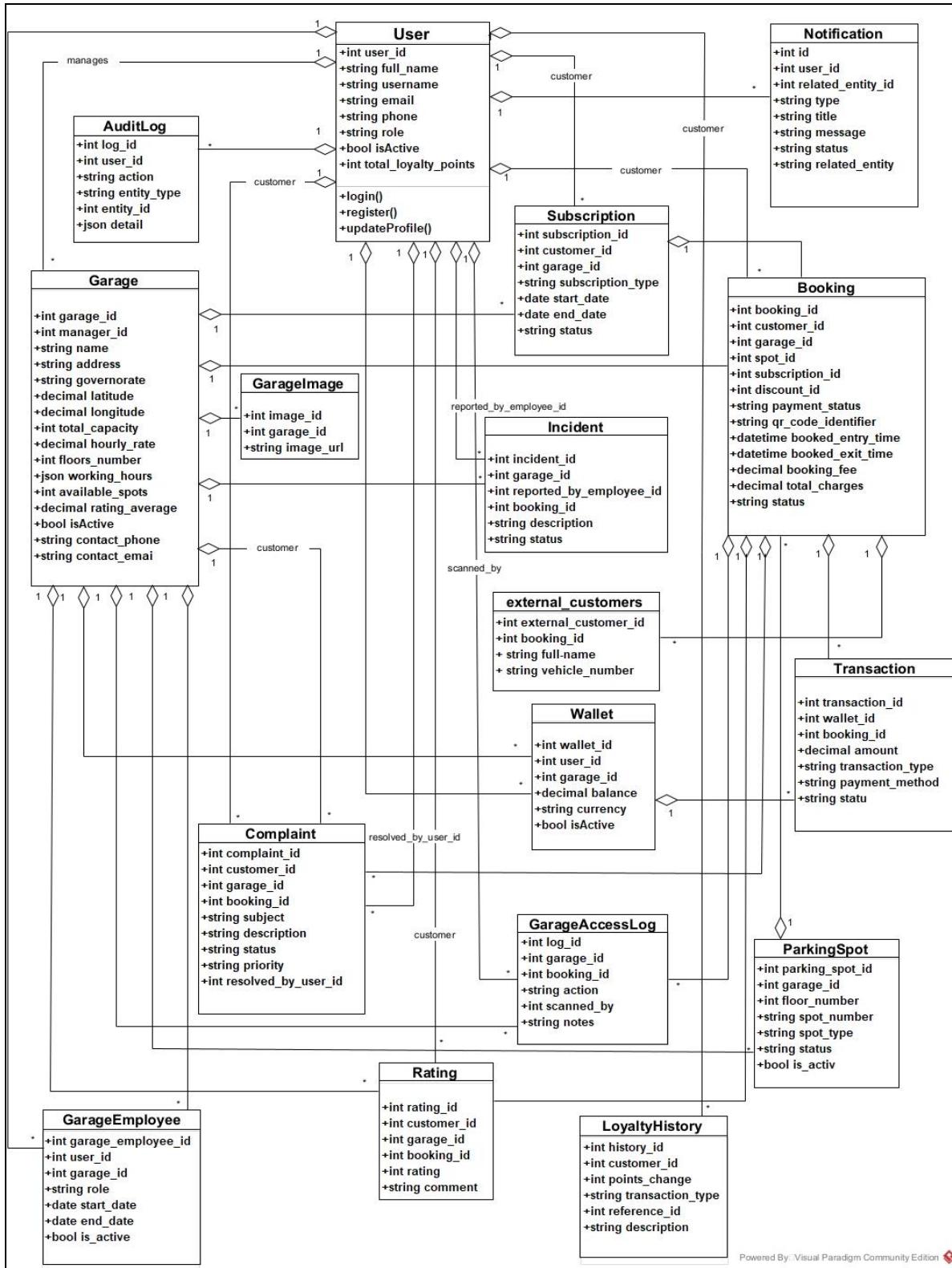
تضمن هذه الآلية الموحدة أماناً وفعالية في عملية المصادقة لجميع المستخدمين، مع الحفاظ على خصوصية البيانات من خلال التشفير، وتوفير تجربة مستخدم سلسة عبر توجيهه فوري للمسار الوظيفي الصحيح بعد التتحقق.

الفصل الخامس: تصميم النظام

١.٥ - مقدمة:

يمثل تصميم النظام مرحلة محورية في البناء المنهجي للمشروع، حيث يتم فيه الانتقال من مرحلة تحليل المتطلبات إلى صياغة الحلول التقنية في صورة نماذج ورسوم تخطيطية توضح الهيكل العام للنظام وأليات تفاعلاته الداخلية. يهدف هذا الفصل إلى عرض الرؤية المعمارية للمكونات البرمجية وال العلاقات التي تربطها، بما يشمل النماذج الكائنية التي تجسد البنية المفاهيمية للنظام، والمخططات التسلسالية التي توضح تدفق العمليات بين العناصر المختلفة، إضافة إلى مخططات العلاقات التي تبين الترابط بين الكيانات، ومن خلال هذه التصاميم، يتم وضع الأساس الذي يربط المفاهيم النظرية بالتطبيق العملي، ممهداً الطريق لمرحلة التنفيذ الفعلي وفق معايير فنية مدققة تضمن وضوح البنية وقابلية التطوير مستقبلاً.

٢.٥ - مخطط الفئات (Class Diagram) :



Powered By: Visual Paradigm Community Edition

(Class Diagram) - مخطط الفئات (١.٥)

تحليل مخطط الفئات لنظام إدارة الكراجات:

يُمثل مخطط الفئات المعرض الهيكل الأساسي لنظام إدارة كراجات السيارات، حيث تُحدد الكيانات الرئيسية وعلاقتها لتلبية احتياجات الأدوار الأربع: العميل، مدير الكراج، الموظف، ومدير النظام. تعكس الفئات التصميم المفاهيمي لقاعدة البيانات مع تحديد السمات وال العلاقات بدقة.

• الفئات الأساسية وتفاعلاته الأدوار

- فئة المستخدم (User):

تُعتبر النواة المركزية للنظام، حيث تحتوي السمات المشتركة لجميع المستخدمين: المعرف الفريد، الاسم الكامل، اسم المستخدم، البريد الإلكتروني، الهاتف، الدور (العميل/مدير النظام/موظف/مدير الكراج)، حالة الحساب، ونقاط الولاء. تُمكن العمليات الأساسية مثل تسجيل الدخول وتحديث الملف الشخصي جميع الأدوار من التفاعل مع النظام.

- فئة الكراج (Garage):

تمثل الكيانات الرئيسية التي يديرها مدير الكراج، وتشمل تفاصيل مثل: الموقع الجغرافي (خطوط الطول والعرض)، السعة الكلية، السعر بالساعة، عدد الطوابق، ساعات العمل، الأماكن المتاحة، ومعلومات الاتصال. يتحكم مدير الكراج في تحديث هذه البيانات وإضافة الصور عبر فئة GarageImage المرتبطة بعلاقة تكوين.

- فئة الحجز (Booking):

ترتبط بين العميل والكراج والموقف، حيث تُسجل تفاصيل الحجز: أوقات الدخول/الخروج المتوقعة، حالة الدفع، رمز QR، والرسوم. يتفاعل المستأجر مع هذه الفئة عند إجراء الحجز، بينما يتولى الموظف التحقق من الحجوزات عبر مسح QR المُخزن في هذه الفئة.

- فئة المحفظة (Wallet):

تدير المعاملات المالية لجميع المستخدمين، حيث تحافظ بالرصيد والعملة. يستخدم المستأجر المحفظة لدفع رسوم الحجز، بينما يتتبع المدير الإداري حركة الأموال عبر فئة المعاملات (Transaction) المرتبطة بها.

• فئات العمليات الإدارية

- فئة الموظف (GarageEmployee):

ترتبط بين المستخدم والكراج، محددةً دور الموظف وتاريخ بدء/انتهاء الخدمة. يستطيع مدير الكراج إدارة موظفيه من خلال هذه الفئة، بينما يستخدم الموظف صلاحياته لتسجيل الدخول/الخروج والبلاغات.

- فئة الشكوى (Complaint):

تسمح للعميل بتقديم الشكاوى المتعلقة بالخدمة، حيث تحدد الأولوية والحالة والموظف الذي حلّها. يتبع مدير الكراج معالجة الشكاوى.

- فئة الحادث (Incident) :

تُستخدم من قبل الموظف للإبلاغ عن الحوادث داخل الكراج، حيث تسجل وصف الحادث والمعرف الخاص بالموظف المبلغ.

- فئات التتبع والمراقبة

- فئة سجل التدقيق (AuditLog) :

ترصد جميع الإجراءات المهمة في النظام مثل تعديل البيانات أو الحذف، حيث تخزن نوع الإجراء والكيان المستهدف. يستخدمها مدير الكراج لمراقبة سلامة النظام واكتشاف أي نشاط غير طبيعي.

- فئة سجل الدخول (GarageAccessLog) :

تسجل عمليات دخول وخروج السيارات بالتفصيل، بما في ذلك الموظف المسؤول عن المسح والملاحظات. توفر لمدير الكراج تقارير دقيقة عن حركة المركبات.

- فئة الإشعار (Notification) :

ترسل تنبيهات للمستخدمين حسب الأحداث، كتأكيد الحجز أو إغلاق الشكوى. تُستخدم من قبل جميع الأدوار لضمان متابعة التحديثات الفورية.

- العلاقات البنوية الرئيسية

- الكراج يتكون من مواقف سيارات (ParkingSpot) وصور (GarageImage)
- الحجز يرتبط بمعاملة مالية (Transaction) وتقييم (Rating)
- العميل يمتلك محفظة (Wallet) واشتراكات (Subscription)
- الموظف يسجل حوادث (Incident) ويدير سجلات الدخول (GarageAccessLog)
- مدير الكراج يراقب سجلات التدقيق (AuditLog) والشكوى (Complaint)

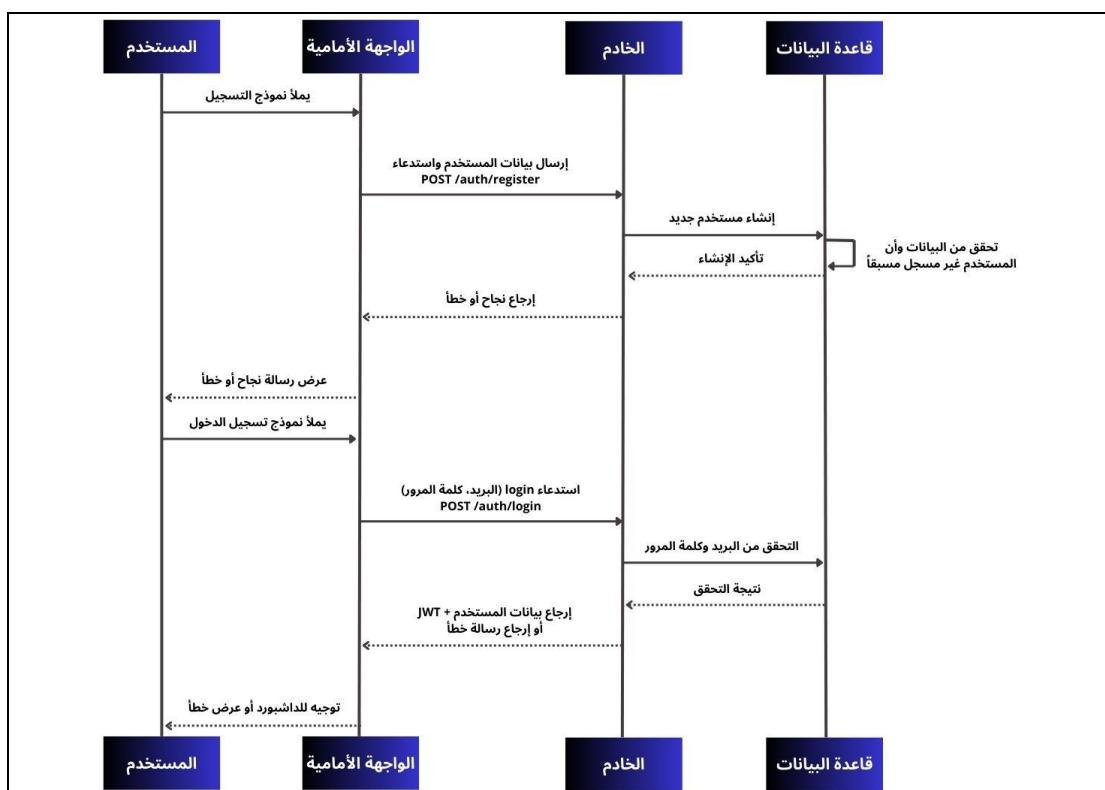
يُظهر هذا التصميم المتكامل كفاءة في توزيع المسؤوليات بين الأدوار المختلفة، مع ضمان سلامة البيانات وسهولة التتبع عبر العلاقات الواضحة بين الكيانات.

٣.٥ - مخططات التسلسل:

- إعداد مخطط تسلسل (Sequence Diagram) لتوضيح تسلسل تبادل الرسائل بين الكائنات: يُعد مخطط التسلسل أداة حيوية لنمذجة تفاعل العناصر في النظام، حيث يُقدم عدة فوائد رئيسية مثل:

- توضيح تدفق العمليات: يُظهر الخطوات الزمنية للتقاعلات بين المكونات
- توثيق الاتصالات: يُحدد البروتوكولات والرسائل بين الواجهة والخادم وقاعدة البيانات
- كشف الاختراقات: يُحدد نقاط الفشل المحتملة (مثل فشل التحقق من بيانات الاعتماد)
- تبسيط التصميم: يقدم تمثيلاً مرجياً للهيكل البرمجي قبل التنفيذ

١.٣.٥ - مخطط sign in



الشكل (٢.٥) - مخطط تسلسل (Sequence Diagram) - مخطط sign in

تحليل عملية تسجيل الدخول وفق المخطط السابق:

- مسار المستخدم:

يبداً العميل/مدير الكراج/الموظف/مدير النظام (جميع الأدوار) بتفاعل مع الواجهة الأمامية، حيث يُدخل بيانات الاعتماد (البريد الإلكتروني وكلمة المرور) في نموذج تسجيل الدخول. تقوم الواجهة بإرسال هذه البيانات عبر طلب API إلى الخادم باستخدام بروتوكول آمن.

- معالجة الخادم:

يستقبل الخادم الطلب وينشئ استعلاماً إلى قاعدة البيانات للتحقق من صحة بيانات الاعتماد.

يقوم بثلاث عمليات جوهرية:

- التحقق من تطابق البريد الإلكتروني مع السجلات
- مطابقة كلمة المرور المشفرة
- التتحقق من حالة الحساب (نشط/موقوف)

- قرارات النظام

- عند النجاح: تسترجع قاعدة البيانات بيانات المستخدم (الدور، الصلاحيات، المعلومات الأساسية) وترسلها للخادم، الذي يولد رمز دخول (Token) ويرسله لواجهة
- عند الفشل: تُرجع قاعدة البيانات رسالة خطأ محددة (بريد غير مسجل/كلمة مرور خاطئة/حساب موقوف)

- استجابة الواجهة: تعرض الواجهة الأمامية للمستخدم:

- حالة النجاح: توجيه فوري إلى لوحة التحكم المناسبة للدور (لوحة العميل/لوحة مدير الكراج/لوحة الموظف/لوحة مدير النظام)
- حالة الفشل: عرض رسالة خطأ سياقية مع إمكانية إعادة المحاولة

- التفاعل مع الأدوار:

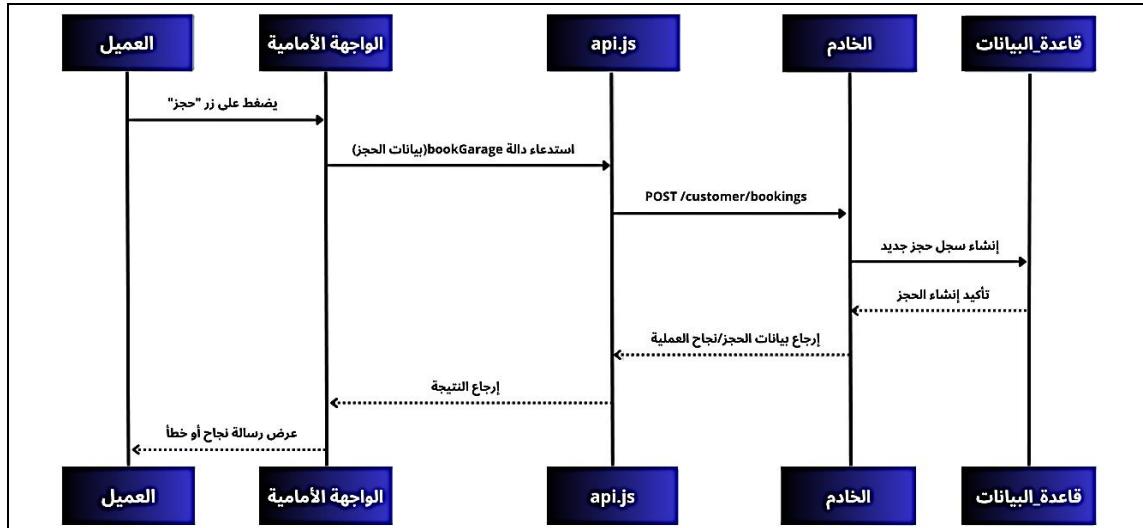
- العميل: يُوجه إلى صفحة البحث عن كراجات
- مدير الكراج: يُوجه إلى لوحة إدارة الكراجات
- الموظف: يُوجه إلى واجهة مسح QR
- مدير النظام: يُوجه إلى لوحة التحكم الشاملة

- ضمانات الأمان:

- التشفير: نقل البيانات ببروتوكولات مشفرة (HTTPS)
- التحقق الثنائي: توثيق الهوية عبر طبقتين (البريد + كلمة المرور)
- عزل الصلاحيات: تحديد الصلاحيات حسب الدور فور المصادقة

يُظهر هذا التدفق كفاءة النظام في التعامل الموحد مع جميع المستخدمين مع الحفاظ على تخصيص التجربة حسب الدور، حيث تُشكل عملية المصادقة الناجحة البوابة الرئيسية لجميع الوظائف اللاحقة في نظام إدارة الكراجات.

٢.٣.٥ - مخطط booking



الشكل (٣.٥) - مخطط تسلسل (Sequence Diagram) - مخطط booking

تحليل مخطط تسلسل عملية حجز موقف في كراج:

يقدم هذا المخطط التسلسلي وصفاً دقيقاً لآلية حجز موقف في كراج من خلال النظام الإلكتروني، حيث يظهر التفاعل بين أربع مكونات رئيسية: العميل، الواجهة الأمامية، `api.js` (طبقة الخدمات)، الخادم، وقاعدة البيانات. يبدأ التفاعل عندما العميل يضغط على زر "حجز" في واجهة التطبيق، مما يطلق سلسلة من الأحداث المتسلسلة.

تقوم الواجهة الأمامية باستدعاء دالة `bookGarage()` مع تمرير بيانات الحجز المدخلة من قبل العميل. هذه الدالة تمثل نقطة الربط بين الواجهة البصرية وطبقة الخدمات. يتلقى مكون `api.js` الاستدعاء ويحوله إلى طلب HTTP من نوع `POST` موجه إلى المسار `/customer/bookings/` على الخادم، حاملاً معه تفاصيل الحجز المدخلة.

يستقبل الخادم الطلب وينفذ عملية إنشاء سجل حجز جديد في قاعدة البيانات. هذه العملية تشمل التحقق من توفر الموقف، وتوليد معرف فريد للحجز، وتخزين معلومات العميل ومواعيد الحجز. عند اكتمال العملية بنجاح، ترسل قاعدة البيانات تأكيداً إلى الخادم يفيد بإنشاء السجل الجديد.

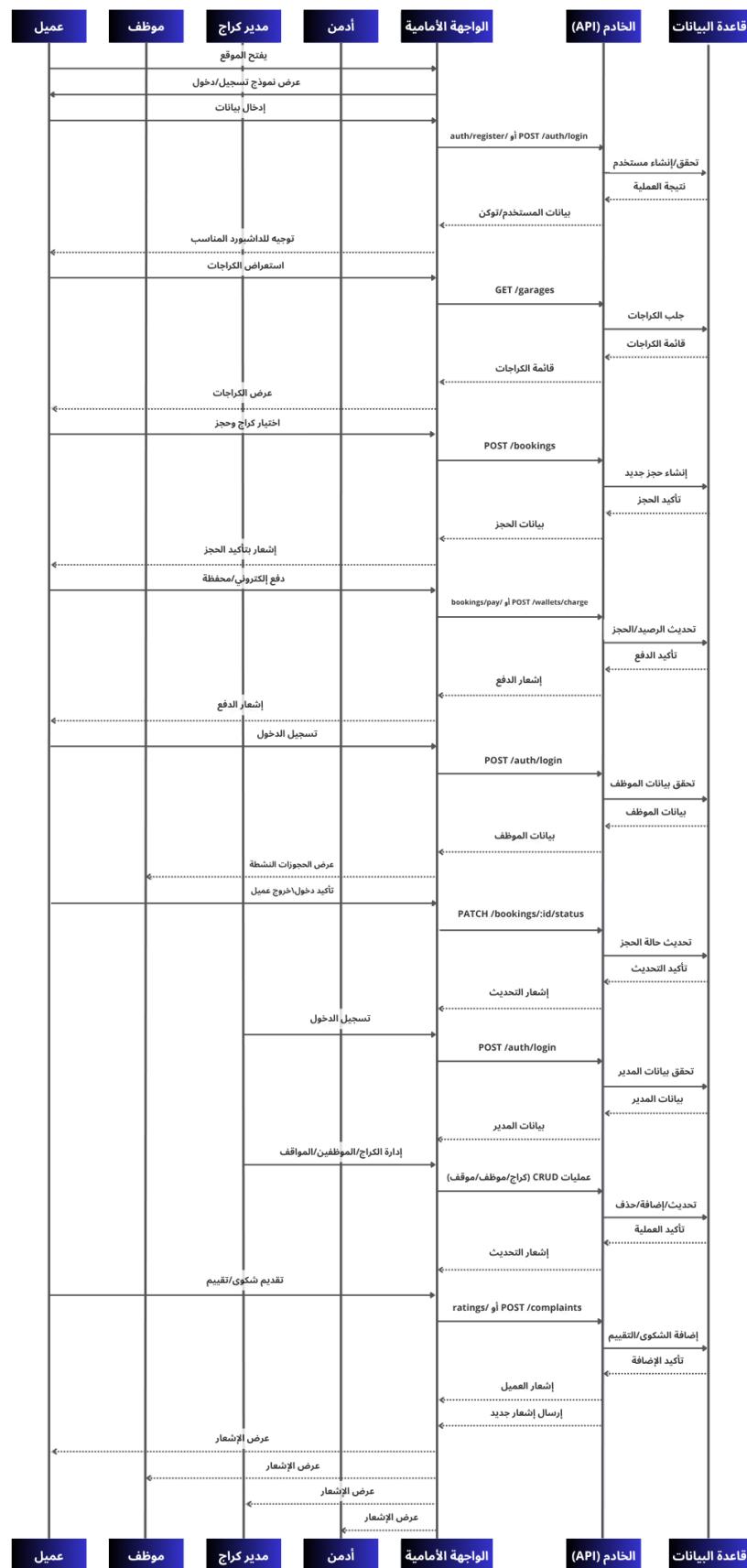
يعيد الخادم بعد ذلك استجابة تحتوي على بيانات الحجز المكتملة أو رسالة نجاح العملية إلى طبقة `api.js`، التي بدورها تمرر هذه النتيجة إلى الواجهة الأمامية. أخيراً، تعرض الواجهة الأمامية رسالة توضيحية للعميل - إما رسالة نجاح تؤكد إتمام الحجز، أو رسالة خطأ في حال وجود مشكلة في البيانات أو التوافر.

يبين هذا المخطط دقة التكامل بين طبقات النظام:

- الواجهة الأمامية ك وسيط تفاعلي مع المستخدم
- api.js ك جسر اتصال مع الخادم
- الخادم ك معالج للمنطق التشغيلي
- قاعدة البيانات ك مستودع موثوق للتخزين

تضمن هذه الآلية المتسلسلة تنفيذ العملية بشكل منظم، حيث ينتقل التحكم والبيانات بشكل أحادي الاتجاه من الطبقة العليا (العميل) إلى الطبقة السفلية (قاعدة البيانات) ثم تعود بشكل تصاعدي مع نتائج التنفيذ، مما يحافظ على وضوح التدفق وموثوقية النظام

٣،٣،٥ - مخطط Sequence For All Actors



الشكل (٤,٥) - مخطط تسلسل (Sequence Diagram) - مخطط تسلسل العمليات في نظام إدارة الكراجات

تحليل مخطط تسلسل العمليات في نظام إدارة الكراجات:

يمثل مخطط التسلسل هذا البنية التحتية التفاعلية لنظام إدارة كراجات متكامل، حيث يتعاون أربعة مستفيدين رئيسيين (العميل، الموظف، مدير الكراج، مدير النظام) مع طبقات النظام التقنية لتنفيذ عمليات متناسقة. يوضح المخطط تدفق الأحداث بين المكونات بالترتيب التالي:

- العميل يبدأ التفاعل بفتح الموقع عبر الواجهة الأمامية، التي تعرض له نموذج تسجيل الدخول أو الاشتراك. بعد إدخال بياناته، تُرسل المعلومات عبر طلب POST إلى الخادم (API) الذي يتواصل مع قاعدة البيانات للتحقق من صحة المستخدم أو إنشاء حساب جديد. عند نجاح العملية، يتلقى العميل بيانات المستخدم ورمز الوصول (Token) ويتم توجيهه إلى لوحة التحكم المناسبة حيث يمكنه استعراض الكراجات المتاحة عبر طلبات GET.
- عند اختيار كراج معين، يُنشئ العميل حجزاً جديداً عبر طلب POST، فيقوم الخادم بتسجيل الحجز في قاعدة البيانات ويعيد تأكيداً للعملية. يلي ذلك عملية الدفع الإلكتروني إما عبر شحن المحفظة أو الدفع المباشر، حيث يُحدث الخادم رصيد المحفظة أو حالة الحجز في قاعدة البيانات. وأخيراً، يمكن للعميل تقديم شكوى أو تقييمات تخزن في النظام وتؤخذ إشارات لجميع الأطراف المعنية.
- الموظف يدخل النظام عبر بوابة مستقلة ببيانات اعتماده الخاصة. بعد التحقق من هويته، يتلقى قائمة بالحجوزات النشطة في كراجه. عند تأكيده دخول أو خروج عميل ما، يُرسل طلب تحديث حالة الحجز (PATCH) إلى الخادم، الذي يُعدل سجلات قاعدة البيانات ويعيد إشعاراً بتأكيد التحديث للموظف.
- مدير الكراج يمثل لمسار تسجيل دخول مشابه، لكن صلاحياته تمتد إلى إدارة الكراج بالكامل. من خلال لوحة تحكم متخصصة، ينفذ عمليات إنشاء/قراءة/تحديث/حذف (CRUD) للكراجات والمواقف وسجلات الموظفين عبر طلبات متعددة إلى الخادم. كل عملية تحديث تؤكدها قاعدة البيانات، ويرسل الخادم إشارات بالتغييرات لضمان اتساق البيانات عبر النظام.
- مدير النظام يتفاعل مع النظام بشكل غير مباشر عبر الإشارات التلقائية. لا يُظهر المخطط إجراءات نشطة من مدير النظام، بل يتلقى إشارات بنفس الآلية التي تصل للعميل والموظف ومدير الكراج عند حدوث أحداث مهمة مثل إضافة تقييمات أو شكوى جديدة أو تحديثات في حالة الحجوزات.

الطبقة التقنية تعمل ك وسيط حيوي:

- الواجهة الأمامية تقدم واجهات المستخدم وتجمع المدخلات
- الخادم (API) يعالج المنطق التشغيلي ويتحقق من الصلاحيات
- قاعدة البيانات تخزن وتسترد جميع السجلات بدءاً من بيانات المستخدمين إلى الحجوزات والدفعات.

تلحق هذه التفاعلات نظاماً متربطاً حيث يضمن تدفق المعلومات في كلا الاتجاهين (طلبات/ردود) تحديداً فورياً لكافة الأطراف. آلية الإشعارات الموحدة تمثل العمود الفقري للتسيير بين المستخدمين، مما يمكن كل مستفيد من أداء دوره ضمن دورة حياة الخدمة بدءاً من الحجز حتى التقييم النهائي، مع الحفاظ على اتساق البيانات عبر جميع طبقات النظام.

٤.٤ - تصميم قاعدة البيانات:

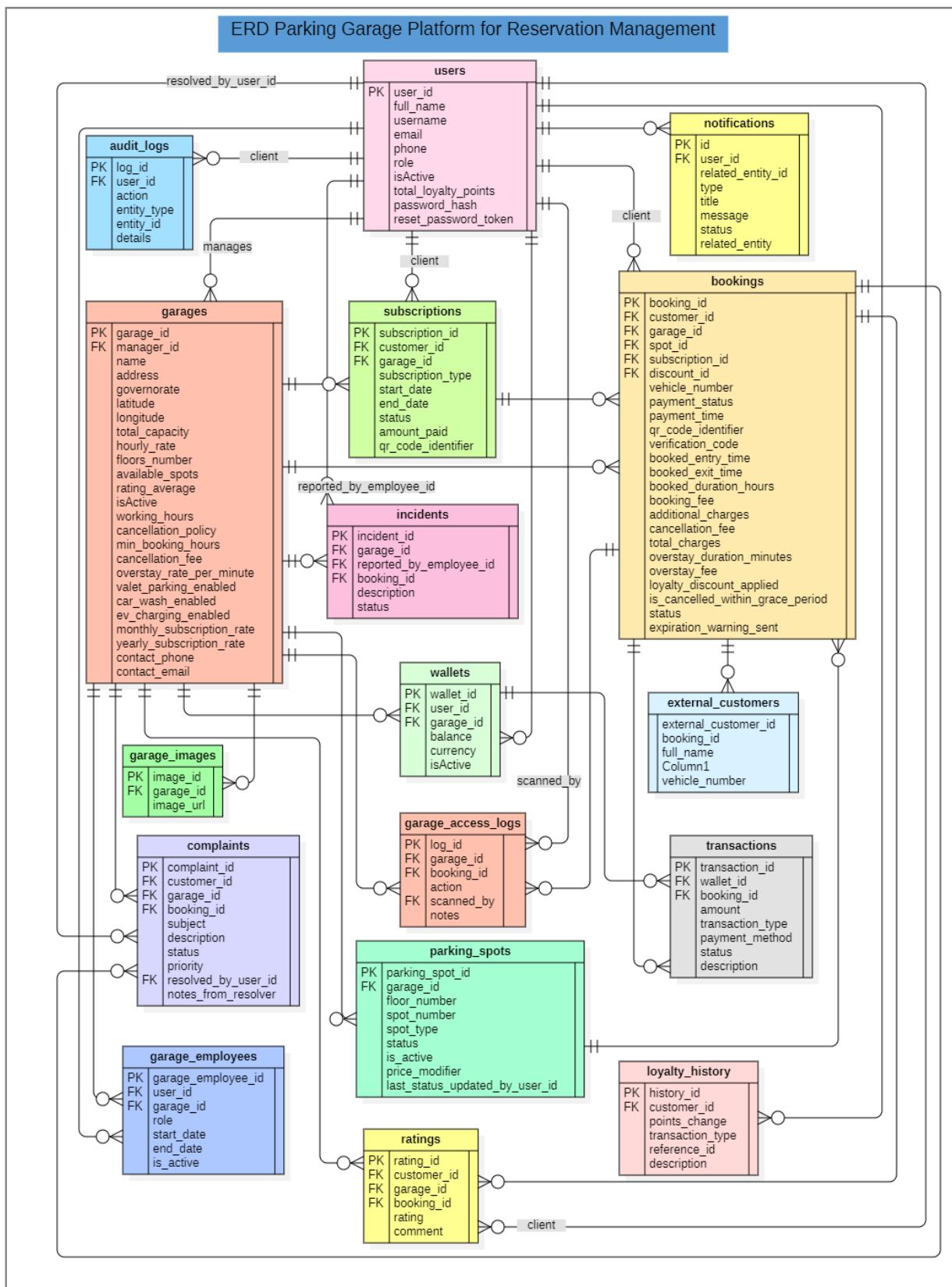
مخطط ERD:

يعد مخطط علاقـة الكيانـات (ERD) أداة مـهمـة في تصـمـيم قـوـاعـدـ الـبـيـانـاتـ، وـلـهـ العـدـيدـ مـنـ الـفوـائـدـ،ـ منهاـ:

- تحسين الفهم: يساعد ERD في توضيح كيفية ارتباط الكيانات المختلفة ببعضها البعض، مما يسهل فهم العلاقات المعقدة بين البيانات.
- تسهيل التواصل: يوفر المخطط وسيلة فعالة للتواصل بين الفرق المختلفة، حيث يمكن للجميع فهم الهيكل العام للبيانات والعلاقات دون الحاجة إلى معرفة تقنية عميقـةـ.
- دعم التصميم: يعد ERD نقطة انطلاق مرئية لتصميم قاعدة البيانات، مما يساعد في تحديد متطلبات النظام وتحليل البيانات بشكل أفضل.
- تحديد الأخطاء: يمكن أن يكشف ERD عن الأخطاء في تصميم قاعدة البيانات، مما يسهل عملية تصحيح الأخطاء وتحسين جودة البيانات.
- تعزيز التعاون: يسهل ERD التعاون بين الفرق المختلفة، حيث يمكن لفرق العمل معاً على المخطط، مما يعزز من فعالية التواصل ويقلل من سوء الفهم.
- تسهيل إعادة هندسة العمليات: يستخدم ERD في مشاريع إعادة تصميم العمليات التجارية، حيث يوفر نظرة شاملة على البيانات داخل أنظمة المعلومات، مما يساعد في تطوير حلول بيانات أكثر كفاءة.

يُعتبر مخطط علاقـة الكيانـات أداة حـيـوـية لـتـحـلـيل وـتـصـمـيم قـوـاعـد الـبـيـانـات، مـا يـسـهـل فـهـم الـبـيـانـات وـتـحـسـين التـواـصـل بـيـن الفـرـق المـخـتـلـفة.

إـعـادـة مـخـطـط ERD لـتـحـدـيد الجـداـول وـالـعـلـاقـات بـيـنـهـا، مـثـل جـدـول الـمـسـتـخـدـمـين، جـدـول الـحـجـوزـات، وـجـدـول الـخـدـمـات كـمـا فـي الشـكـل التـالـي:



الشكل (٥.٥) - مخطط (ERD) - الشكل (٥.٥) - مخطط (ERD)

تحليل مخطط Erd لنظام إدارة مواقف السيارات:

يمثل مخطط الكيانات وال العلاقات (Erd) السابق البنية المفاهيمية لنظام متكملا لإدارة مواقف السيارات، مصمم وفق معايير التطبيع القياسية (Normalization). إذ يعكس المخطط التكمال الوظيفي بين المكونات الأساسية للنظام عبر علاقات هرمية واضحة.

- الكيان المركزي (Users):

يحتوي على السمات الأساسية للمستخدمين (المعرف، الاسم الكامل، البريد الإلكتروني، الدور) يرتبط بـ ٩ كيانات فرعية عبر علاقات "واحد إلى متعدد" (1:N) السمة isActive تمكن إدارة حالة الحساب ديناميكياً

- نظام المواقف (Garages):

يحدد المواصفات الفنية للكراجات (السعة، الموقع الجغرافي، معدل الساعة) يحتوي على مفاتيح مرجعية لـ manager_id (Users) يرتبط به:

Parking_Spots -

Garage_Images -

Employees -

- إدارة الحجوزات (Bookings):

تمثل النواة التشغيلية للنظام وتحتوي على ١٣ سمة تعطي دورة الحجز الكاملة حيث تربط بين:

- العملاء (Customer_id)

- الكراجات (Garage_id)

- المواقف (Spot_id)

- الاشتراكات (Subscription_id)

- النظام المالي المتكمال:

- المحافظ (Wallets): تدير الأرصدة المالية

- المعاملات (Transactions): تسجل التدفقات المالية
 - الاشتراكات (Subscriptions): تدير الباقات الدورية
 - أنظمة الدعم:
 - التقييمات (Ratings): تقيس تجربة المستخدم
 - سجلات المراجعة (Audit_Logs): تتبع التغييرات
 - نظام الولاء (Loyalty_History): يدير نقاط المكافآت
 - سجلات الدخول (Garage_Access_Logs): تراقب الوصول الفعلي
 - التكامل الوظيفي:
 - يضمن التصميم تدفقاً متاغماً للبيانات عبر ثلاث طبقات رئيسية:
 - طبقة المستخدم: (العملاء + الموظفون + المدراء)
 - طبقة العمليات: (الحجوزات + المعاملات + الوصول)
 - طبقة الدعم: (التقييمات + السجلات + نظام المكافآت)
 - الضوابط التصميمية:
 - جميع الجداول تحقق التصميم الطبيعي الثالث (NF³)
 - استخدام isActive في كيانات متعددة لإدارة الحالة
 - وجود سمات مراقبة زمنية في (Bookings, Subscriptions)
 - استخدام JSON في Audit_Logs.details لمرونة التسجيل.
- فيما يلي توضيح لمحتوى كل جدول وتوصيف الكيانات:

١. جدول المستخدمين (users):

يخزن بيانات المستخدمين الأساسية (مثل الاسم، البريد، الهاتف)، تفاصيل الحساب (كلمة المرور، أدوار الصلاحيات)، إعدادات الأمان (توكن إعادة التعيين)، وحالة النشاط مع تتبع نقاط الولاء. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٥): جدول قاعدة البيانات users_1

users			
القيود واللاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للمستخدم	INT	user_id
NULL	الاسم الكامل للمستخدم	VARCHAR	full_name
UNIQUE, NOT NULL	اسم المستخدم الفريد (إجباري).	VARCHAR	username
NOT NULL	التجزئة المشفرة لكلمة المرور (إجباري)	VARCHAR	password_hash
UNIQUE, NOT NULL	البريد الإلكتروني الفريد للمستخدم (إجباري).	VARCHAR	email
	رقم الهاتف (اختياري، يمكن أن يكون فارغاً).	VARCHAR	phone
NOT NULL	دور المستخدم: ,garage_admin ,admin customer ,employee (إجباري).	ENUM	role
DEFAULT: TRUE	حالة الحساب (true/false)	BOOLEAN	isActive
NULL	تاريخ/وقت آخر تسجيل دخول	TIMESTAMP	last_login
UNIQUE يمكن أن يكون فارغاً	رمز إعادة تعيين كلمة المرور	VARCHAR	reset_password_token
	تاريخ/وقت انتهاء صلاحية رمز إعادة التعيين (اختياري).	TIMESTAMP	reset_password_expires
DEFAULT: 0	نقاط الولاء التراكمية للمستخدم.	INT	total_loyalty_points

يضاف تلقائياً عند الإنشاء	تاريخ/ وقت إنشاء الحساب	TIMESTAMP	created_at
يُحدث تلقائياً عند التعديل	تاريخ/ وقت آخر تحديث الحساب	TIMESTAMP	updated_at

٢. جدول الكراجات (garages)

يحتوي على معلومات التشغيل الأساسية للمرائب تشمل الموقع الجغرافي، السعة، الأسعار، سياسات العمل، ومعلومات الاتصال مع مؤشرات الأداء. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٢٥): جدول قاعدة البيانات garages_2

garages			
القيود واللاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للكراج	INT	garage_id
NOT NULL, FK (users.user_id)	مدير الكراج	INT	manager_id
NOT NULL	اسم الكراج (إجباري).	VARCHAR	name
NOT NULL	العنوان التفصيلي (إجباري).	TEXT	address
NOT NULL	المحافظة/المنطقة (إجباري).	VARCHAR	governorate
-	خط العرض الجغرافي (اختياري).	DECIMAL	latitude
-	خط الطول الجغرافي (اختياري).	DECIMAL	longitude
NOT NULL	السعة الكلية لمواقف السيارات (إجباري).	INT	total_capacity
NOT NULL	سعر الساعة (إجباري).	DECIMAL	hourly_rate

NOT NULL	عدد الطوابق (إجباري).	INT	floors_number
-	ساعات العمل	JSON	working_hours
DEFAULT: 0	الأماكن المتاحة حالياً	INT	available_spots
DEFAULT: 0.00	متوسط التقييم	DECIMAL	rating_average
نص وصفي	سياسة الإلغاء	TEXT	cancellation_policy
DEFAULT: 1.00	الحد الأدنى لساعات الحجز	DECIMAL	min_booking_hours
DEFAULT: 0	رسوم الإلغاء	DECIMAL	cancellation_fee
DEFAULT: 0.00	سعر التجاوز لكل دقيقة	DECIMAL	overstay_rate_per_minute
DEFAULT FALSE	خدمة صف السيارات (الافتراضي: غير مفعل).	BOOLEAN	valet_parking_enabled
DEFAULT FALSE	خدمة غسيل السيارات (الافتراضي: غير مفعل).	BOOLEAN	car_wash_enabled
DEFAULT FALSE	خدمة شحن السيارات الكهربائية (الافتراضي: غير مفعل).	BOOLEAN	ev_charging_enabled
NULL	سعر الاشتراك الشهري (اختياري).	DECIMAL (10, 2)	monthly_subscription_rate
NULL	سعر الاشتراك السنوي (اختياري).	DECIMAL (10, 2)	yearly_subscription_rate

DEFAULT TRUE	حالة الكراج (نشط/معطل).	BOOLEAN	isActive
-	هاتف التواصل (اختياري).	VARCHAR	contact_phone
-	بريد إلكتروني التواصل (اختياري).	VARCHAR	contact_email
DEFAULT: يُسجل تلقائياً	تاريخ الإنشاء	TIMESTAMP	created_at
DEFAULT: يُحدث تلقائياً عند التعديل	تاريخ التحديث	TIMESTAMP	updated_at

٣. جدول الإشعارات (notifications)

يدير إرسال التنبيةات للمستخدمين (بريد/منصة) مع تفاصيل المحتوى، الحالة، وارتباطها بالعمليات المختلفة مثل الحجوزات أو المعاملات. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٣،٥): جدول قاعدة البيانات notifications^٣

notifications			
القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للإشعار	INT	id
NOT NULL, FK (users.user_id)	المستخدم المستهدف	INT	user_id
NOT NULL	المعرف المرتبط بالكيان (مثل رقم الحجز، المعاملة، التقييم، إلخ).	INT	related_entity_id
NOT NULL	نوع الإشعار: in_app أو email (إجباري).	ENUM	type

NOT NULL	عنوان الإشعار (إجباري).	VARCHAR	title
NOT NULL	محتوى الإشعار (إجباري).	TEXT	message
pending:DEFAULT	حالة الإشعار: pending (بالانتظار) sent (قيد الإرسال) read (تم القراءة).	ENUM	status
NULL	الكيان المرتبط: booking (حجز) transaction (معاملة) register (تسجيل) rating (تقييم) complaint (شكوى) promotion (عرض) loyalty (ولاء).	ENUM	related_entity
DEFAULT: يسجل تلقائياً عند الإرسال	تاريخ/وقت الإرسال	TIMESTAMP	sent_at
DEFAULT: يسجل تلقائياً عند الإنشاء	تاريخ الإنشاء	TIMESTAMP	created_at

٤. جدول صور الكراجات (garage_images):

يربط الصور التوضيحية بالمرائب المحددة عبر روابط تخزين خارجية. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٤،٥) : جدول قاعدة البيانات ؛_garage_images

garage_images

القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للصورة	INT	image_id
NOT NULL, FK (garages. garage_id) ON DELETE CASCADE	المعرف المرتبط بالكراج	INT	garage_id
NOT NULL	رابط الصورة (إجباري).	VARCHAR	image_url
DEFAULT: يُسجل تلقائياً	تاريخ رفع الصورة	TIMESTAMP	created_at

٥. جدول أماكن وقوف السيارات (parking_spots):

يحدد الموصفات التشغيلية لكل مكان وقوف (النوع، الحالة، السعر) وتاريخ الصيانة مع تتبع آخر تحديث. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٥٥) : جدول قاعدة البيانات parking_spots

parking_spots			
القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للمكان (توليد تلقائي).	INT	parking_spot_id
NOT NULL, FK (garages. garage_id) ON DELETE CASCADE	الكراج التابع له المكان	INT	garage_id
NOT NULL	رقم الطابق (إجباري).	INT	floor_number
NOT NULL	رقم/رمز المكان (إجباري).	VARCHAR	spot_number
DEFAULT: normal	: النوع	ENUM	spot_type

	normal (عادي) electric (كهربائي) vip (مميز).		
available:DEFAULT	الحالة: available (متاح) occupied (مشغول) under_maintenance (صيانة) closed (مغلق).	ENUM	status
DEFAULT: 1.00	مُعدّل السعر	DECIMAL	price_modifier
DEFAULT: TRUE	المكان نشط/غير نشط	BOOLEAN	is_active
DEFAULT: تسجيل تلقائي	تاريخ آخر حجز للمكان.	TIMESTAMP	last_booked_at
NULL	بداية فترة الصيانة (اختياري).	TIMESTAMP	maintenance_start_date
NULL	نهاية فترة الصيانة (اختياري).	TIMESTAMP	maintenance_end_date
NULL, FK (users.user_id)	آخر مستخدم عَدَّل الحالة (مرتبط بجدول المستخدمين).	INT	last_status_updated_by_user_id
DEFAULT: تسجيل تلقائي	تاريخ الإنشاء	TIMESTAMP	created_at
DEFAULT: تحديث تلقائي عند التعديل	تاريخ التحديث	TIMESTAMP	updated_at
UNIQUE (garage_id,	يضمن عدم تكرار نفس المكان داخل نفس الطابق والكراج.	-	UNIQUE Constraint

floor_number, spot_number)			
-------------------------------	--	--	--

٦. جدول موظفي الكراجات (garage_employees):

يوثق ارتباط الموظفين بالمرائب التي يعملون فيها، مع تحديد فترات العمل والأدوار الوظيفية.
وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٦,٥): جدول قاعدة البيانات `garage_employees`

garage_employees			
القيود والملاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد لسجل الموظف	INT	garage_employee_id
NOT NULL, FK (users.user_id)	المستخدم المرتبط	INT	user_id
NOT NULL, FK (garages. garage_id)	الكراج الذي يعمل به الموظف	INT	garage_id
	دور الموظف: supervisor (مشرف) scanner (مسح).	ENUM	role
-	تاريخ بداية العمل (اختياري).	DATE	start_date
-	تاريخ نهاية العمل (اختياري).	DATE	end_date
DEFAULT TRUE	حالة التوظيف: نشط/غير نشط	BOOLEAN	is_active
DEFAULT: تسجيل تلقائي	تاريخ الإنشاء	TIMESTAMP	created_at
DEFAULT: تحديث تلقائي عند التعديل	تاريخ التحديث	TIMESTAMP	updated_at

UNIQUE (user_id, garage_id)	يمنع تكرار نفس المستخدم في نفس الكراج.	-	UNIQUE Constraint
-----------------------------	--	---	-------------------

٧. جدول المحافظ (wallets):

يدير الأرصدة المالية للمستخدمين حسب المرأب والعملة، مع تتبع التحديثات والتشغيل. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول قاعدة البيانات wallets (٧، ٥):

wallets			
القيود واللاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للمحفظة	INT	wallet_id
NOT NULL, FK (users.user_id) ON DELETE CASCADE	المستخدم المالك للمحفظة	INT	user_id
NOT NULL, FK (Garages. garage_id) ON DELETE CASCADE	الكراج المرتبط بالمحفظة	INT	garage_id
DEFAULT :0	الرصيد الحالي	DECIMAL	balance
DEFAULT: SYR	العملة	VARCHAR	currency
DEFAULT: TRUE	حالة المحفظة: نشطة/معطلة	BOOLEAN	isActive
تسجيل تلقائي	تاريخ الإنشاء	TIMESTAMP	created_at
تحديث تلقائي عند تعديل الرصيد	تاريخ التحديث	TIMESTAMP	updated_at
UNIQUE (user_id, garage_id)	يمنع إنشاء أكثر من محفظة لنفس المستخدم في نفس الكراج.	-	UNIQUE Constraint

٨. جدول الحجوزات (bookings)

يسجل التفاصيل الكاملة للحجوزات تشمل: الأوقات (المخطط/الفعلي)، الرسوم، الحالة، وارتباطها بالخصومات والاشتراكات. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٨,٥) : جدول قاعدة البيانات [bookings_٨](#)

bookings			
القيود واللاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للحجز	INT	booking_id
NOT NULL, FK (users.user_id)	العميل صاحب الحجز	INT	customer_id
NOT NULL, FK (Garages.garage_id)	الكراج المحجوز	INT	garage_id
NOT NULL, FK (parking_spots. spot_id)	مكان الوقوف المحجوز	INT	spot_id
NULL, FK (subscriptions. subscription_id)	الاشتراك المستخدم	INT	subscription_id
NULL, FK (discounts_and_offers. offer_id)	الخصم المستخدم (إن وجد).	INT	discount_id
DEFAULT: unpaid	حالة الدفع: (مدفوع) paid غير مدفوع) unpaid	ENUM	payment_status
UNIQUE NOT NULL	رمز QR فريد لتأكيد الحجز (إجباري).	VARCHAR	qr_code_identifier

NULL	يستخدم لتخزين رمز التحقق/التفعيل الخاص بالحجز (مثل: كود تأكيد الحجز).	VARCHAR	verification_code
NULL	تاريخ انتهاء صلاحية QR (للحجوزات المؤقتة).	TIMESTAMP	qr_code_expiry
NULL	وقت الدخول الفعلي للسيارة.	TIMESTAMP	entry_time_actual
NULL	وقت الخروج الفعلي للسيارة.	TIMESTAMP	exit_time_actual
NOT NULL	وقت الدخول المخطط للحجز (إجباري).	DATETIME	booked_entry_time
NOT NULL	وقت الخروج المخطط للحجز (إجباري).	DATETIME	booked_exit_time
NOT NULL	المدة المحجوزة (بالساعات)	DECIMAL	booked_duration_hours
NOT NULL	تكلفة الحجز الأساسية (قبل الخصومات).	DECIMAL	booking_fee
DEFAULT :0.00	رسوم إضافية (خدمات إضافية).	DECIMAL	additional_charges
DEFAULT: 0.00	رسوم الإلغاء (تحسب عند الإلغاء).	DECIMAL	cancellation_fee
DEFAULT: 0.00	إجمالي المبلغ المستحق (بعد الخصومات والإضافات).	DECIMAL	total_charges
DEFAULT 0	مدة التجاوز عن وقت الخروج (بالدقيقة).	INT	overstay_duration_minutes
DEFAULT 0.00	رسوم التجاوز الزمني.	DECIMAL	overstay_fee
DEFAULT 0.00	قيمة الخصم من نقاط الولاء.	DECIMAL	loyalty_discount_applied

NULL	وقت إلغاء الحجز.	TIMESTAMP	cancellation_time
DEFAULT NULL	هل تم الإلغاء خلال الفترة المسموحة؟	BOOLEAN	is_cancelled_within_grace_period
NOT NULL	حالة الحجز: (بانتظار pending_payment) (دفع confirmed) (نشط active) (مكتمل completed) (ملغى cancelled) (تجاوز الوقت overstayed).	ENUM	status
-	ملاحظات إضافية عن الحجز.	TEXT	notes
DEFAULT: تسجيل تلقائي	تاريخ إنشاء الحجز.	TIMESTAMP	created_at
DEFAULT: تحديث تلقائي عند تعديل الحجز	تاريخ آخر تحديث للحجز.	TIMESTAMP	updated_at

٩. جدول المعاملات المالية (transactions)

يرصد الحركات المالية (مدفوعات، استرداد، غرامات) مع وسائل الدفع وحالة التنفيذ. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (٩،٥): جدول قاعدة البيانات transactions^٩

transactions			
القيود والملاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للمعاملة	INT	transaction_id
NOT NULL,	المحفظة المرتبطة بالمعاملة.	INT	wallet_id

FK (wallets. wallet_id)			
NULL, FK (bookings. booking_id)	الحجز المرتبط (إن وجد).	INT	booking_id
NOT NULL	قيمة المعاملة (موجب: إيداع، سالب: سحب).	DECIMAL	amount
NOT NULL	النوع: (رسوم حجز) booking_fee (شحن) top_up (استرداد) refund (تجاوز) overstay_charge (إلغاء) cancellation_fee (تعديل نظام) system_adjustment (دفع اشتراك) subscription_payment	ENUM	transaction_type
NOT NULL	وسيلة الدفع: (محفظة إلكترونية) e_wallet (نقدى) cash	ENUM	payment_method
NOT NULL, DEFAULT: pending	الحالة: (مكتملة) completed (معلقة) pending (فاشلة) failed	ENUM	status
-	وصف تفصيلي	VARCHAR	description
تسجيل تلقائي	يمنع إنشاء أكثر من محفظة لنفس المستخدم في نفس الكراج.	TIMESTAMP	created_at

DEFAULT: يُحدث عند كل تعديل	تاريخ التحديث	TIMESTAMP	updated_at
-----------------------------	---------------	-----------	------------

١٠. وصف جدول التقييمات (ratings):

يجمع آراء العملاء في المرائب مرتبطة بالحجوزات مع التقييمات الرقمية والتعليقات. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٠,٥): جدول قاعدة البيانات ratings_١٠

ratings			
القيود والملاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للتقييم	INT	rating_id
NOT NULL, FK	معرف العميل الذي قام بالتقييم	INT	customer_id
NOT NULL, FK	معرف الكراج الذي تم تقييمه	INT	garage_id
FK	معرف الحجز المرتبط بالتقييم (اختياري)	INT	booking_id
NOT NULL, CHECK (rating 1-5)	قيمة التقييم (من ١ إلى ٥ نجوم)	INT	rating
NULL	تعليق نصي حر يضيفه العميل (اختياري)	TEXT	comment
تسجيل تلقائي	تاريخ ووقت إنشاء التقييم	TIMESTAMP	created_at
DEFAULT: يُحدث عند كل تعديل	تاريخ ووقت آخر تحديث للتقييم	TIMESTAMP	updated_at

١١. وصف جدول سجلات التدقيق (audit_logs):

يوثق تغييرات النظام الهامة التي يقوم بها المستخدمون مع تفاصيل الإجراء والكيان المتأثر. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١١,٥): جدول قاعدة البيانات audit_logs_١١

audit_logs

القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد لسجل التدقيق	INT	log_id
FK (users.user_id)	معرف المستخدم الذي قام بالإجراء	INT	user_id
NOT NULL	نوع الإجراء المسجل (مثلاً: تحديث، حذف، إنشاء)	VARCHAR	action
-	نوع الكيان المستهدف	VARCHAR	entity_type
-	معرف الكيان المستهدف	INT	entity_id
-	تفاصيل التغيير	JSON	details
DEFAULT: تسجيل تلقائي	تاريخ ووقت حدوث الحدث	TIMESTAMP	created_at

١٢. وصف جدول الاشتراكات (subscriptions):

يدير خطط الاشتراك الشهرية/السنوية للعملاء مع فترات الصلاحية والحالة. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٢,٥) : جدول قاعدة البيانات subscriptions_١٢

subscriptions			
القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للاشتراك	INT	subscription_id
NOT NULL, FK (user.user_id)	معرف العميل المشترك (مرتبط بجدول المستخدمين)	INT	customer_id
NOT NULL, FK (Garage. Garages_id)	معرف المرآب (الجراج) الذي تم الاشتراك فيه	INT	garage_id
NOT NULL	نوع الاشتراك (شهري أو سنوي)	ENUM	subscription_type
NOT NULL	تاريخ بدء الاشتراك	DATE	start_date
NOT NULL	تاريخ انتهاء الاشتراك	DATE	end_date

DEFAULT: active	حالة الاشتراك: active نشط expired منتهي cancelled ملغى	ENUM	status
NOT NULL	المبلغ المدفوع للاشتراك	DECIMAL	amount_paid
NULL	تاريخ ووقت إلغاء الاشتراك (إن وجد)	TIMESTAM P	cancellation_date
DEFAULT: تسجيل تلقائي	تاريخ ووقت إنشاء الاشتراك	TIMESTAM P	created_at
DEFAULT: يُحدث عند كل تعديل	تاريخ ووقت آخر تحديث للاشتراك	TIMESTAM P	updated_at
NULL	معرف فريد لرمز الاستجابة السريعة (QR) المرتبط بالاشتراك، يستخدم للتحقق من صحة الاشتراك.	varchar	qr_code_identifier

١٣. جدول سجل الولاء (loyalty_history):

يسجل هذا الجدول كل حركة لنقاط الولاء للعملاء سواء إضافة نقاط (مثل مكافآت الحجوزات) أو خصم نقاط (مثل استبدالها بخصومات) مع توضيح سبب الحركة وتاريخها. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٣,٥): جدول قاعدة البيانات loyalty_history_١٣

loyalty_history			
القيود والملاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد لسجل نقاط الولاء	INT	history_id
FK (user.user_id)	المعرف الفريد للعميل المرتبط بالحدث	INT	customer_id
NOT NULL	مقدار التغيير في نقاط الولاء (إيجابي/سلبي)	INT	points_change

الوقت الحالي: DEFAULT	نوع العملية المسببة للتغيير: إكمال حجز: booking_completion شراء اشتراك: subscription_purchase استبدال النقاط: redeem_for_credit تعديل يدوي: manual_adjustment استرداد: refund	ENUM	transaction_type
NULL	المعرف المرجعي للعملية الأصلية	INT	reference_id
NULL	وصف تفصيلي للحدث	VARCHAR	description
تسجيل تلقائي: تسجيل	تاريخ ووقت حدوث التغيير.	TIMESTAMP	created_at

١٤. جدول external_customers (العملاء الخارجيين):

جدول (١٤،٥) : جدول قاعدة البيانات ١٤ external_customers

external_customers			
القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للعميل الخارجي	INT	external_customer_id
FK (booking.Bookings_id)	المعرف المرتبط بالحجز في جدول الحجوزات	INT	booking_id
NOT NULL	الاسم الكامل للعميل الخارجي	VARCHAR	full_name
NOT NULL	رقم مركبة العميل (لوحة السيارة).	VARCHAR	vehicle_number
الوقت الحالي: DEFAULT	تاريخ ووقت إنشاء السجل	TIMESTAMP	created_at

١٥. جدول سجلات دخول/خروج الكراج(garage_access_logs):

يسجل أحداث دخول/خروج المركبات مع تحديد الوقت والمسؤول عن المسح. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٥,٥) : جدول قاعدة البيانات _garage_access_logs_١٥

garage_access_logs			
القيود والملحوظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للسجل	INT	log_id
NOT NULL, FK (garages. Garage_id)	معرف الكراج الذي حدث فيه الدخول/الخروج	INT	garage_id
NULL, FK (booking. Booking_id)	معرف الحجز المرتبط بهذا الدخول/الخروج	INT	booking_id
NOT NULL,	نوع الحدث: entry (دخول) أو exit (خروج)	ENUM	action
DEFAULT: تسجيل تلقائي	تاريخ ووقت حدوث الدخول/الخروج	TIMESTAMP	timestamp
FK (user.user_id)	معرف المستخدم الذي قام بعملية المسح	INT	scanned_by
-	ملحوظات إضافية عن الحدث (مثل سبب عدم المسح، أو ملاحظات أخرى)	TEXT	notes

١٦. وصف جدول الشكاوى (complaints) :

يتعقب تقارير المشكلات المقدمة من العملاء مع حالة المعالجة وأولوياتها. وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٦,٥) : جدول قاعدة البيانات _complaints_١٦

complaints			
القيود والملحوظات	الوصف	نوع البيانات	العمود

PK, AUTO_INCREMENT	المعرف الفريد للشكوى (مفتاح أساسي)	INT	complaint_id
FK (user.user_id)	معرف العميل صاحب الشكوى	INT	customer_id
FK (garages. Garage_id)	معرف المرآب المشكو ضده (اختياري)	INT	garage_id
FK (booking. Booking_id)	معرف الحجز المرتبط بالشكوى (اختياري)	INT	booking_id
NOT NULL	موضوع الشكوى (عنوان)	VARCHAR	subject
NOT NULL	تفاصيل الشكوى	TEXT	description
DEFAULT: New	حالة الشكوى الحالية: New In_progress Resolved Closed Rejected	ENUM	status
DEFAULT: Medium	أولوية معالجة الشكوى: Low Medium Hight	ENUM	priority
DEFAULT: تلقي	تاريخ ووقت تقديم الشكوى	TIMESTAMP	filed_at
قد يكون NULL	تاريخ ووقت حل الشكوى	TIMESTAMP	resolved_at

NULL, FK (user.user_id)	معرف المستخدم الذي حل الشكوى	INT	resolved_by_user_id
NULL	ملاحظات من الشخص الذي حل الشكوى	TEXT	notes_from_resolver
DEFAULT: تسجيل تلقي	تاريخ ووقت إنشاء السجل	TIMESTAMP	created_at
DEFAULT: يُحدّث عند كل تعديل	تاريخ ووقت آخر تحديث للسجل	TIMESTAMP	updated_at

١٧. وصف جدول الحوادث (incidents)

يسجل هذا الجدول جميع الحوادث التي تحدث في مرائب الشركة (مثل أضرار السيارات، مشاكل المعدات، حوادث السلامة) وصف الأعمدة ونوع البيانات والخصائص كالتالي:

جدول (١٧،٥) : جدول قاعدة البيانات incidents_١٧

incidents			
القيود والملاحظات	الوصف	نوع البيانات	العمود
PK, AUTO_INCREMENT	المعرف الفريد للحادث	INT	incident_id
NOT NULL, FK (garages. Garage_id)	معرف المرأب الذي وقع فيه الحادث	INT	garage_id
FK (user.user_id)	معرف الموظف الذي أبلغ عن الحادث	INT	reported_by_employee_id
FK	معرف الحجز المرتبط	INT	booking_id

	بالحادث (اختياري)		
NOT NULL	وصف تفصيلي للحادث	TEXT	description
NOT NULL	تاريخ ووقت حدوث الحادث الفعلي	TIMESTAMP	incident_time
Reported:DEFAULT	حالة معالجة الحادث: Reported under_review Resolved	ENUM	status
تسجيل تلقائي DEFAULT:	تاريخ ووقت تسجيل الحادث في النظام	TIMESTAMP	created_at

٥.٥ - تصميم الواجهات:

١.٥.٥ - الواجهة الرئيسية للمنصة:



الشكل (٦.٥) – الواجهة الرئيسية للمنصة

٢.٥.٥ - واجهة العميل الأساسية:

لوحة التحكم
Client1-Garage1
مرحبا بك.

5 نقاط الولاء
الرصيد ٧٩,٨٣٥ ل.س.
١٠ عدد الحجوزات

الكراجات المتاحة

النعناع ١ موقف متاح
دمشق -شارع مسلم بارودي ١٠٠٠ ليرة سورية/ساعة

العنوان ١ موقف متاح
دمشق -شارع مسلم بارودي ١٠٠٠ ليرة سورية/ساعة

الحجوزات

الإشعارات

ملف الشخصي

الاشتراك

Park Easy مشروع إدارة كراج ذكي

الشكل (٧.٥) – واجهة العميل الأساسية

١.٢.٥.٥ - واجهة الكراجات لدى العميل:

لوحة التحكم
Client1-Garage1
مرحبا بك.

الكراجات المتاحة

النعناع ١ موقف متاح
دمشق -شارع مسلم بارودي ١٠٠٠ ليرة سورية/ساعة

الحرمرات ٣ موقف متاح
طرطوس، شارع الكرايجة ٢٥٠٠ ليرة سورية/ساعة

الحضارة ٤ موقف متاح

الحجوزات

نقطة الولاء

الإشعارات

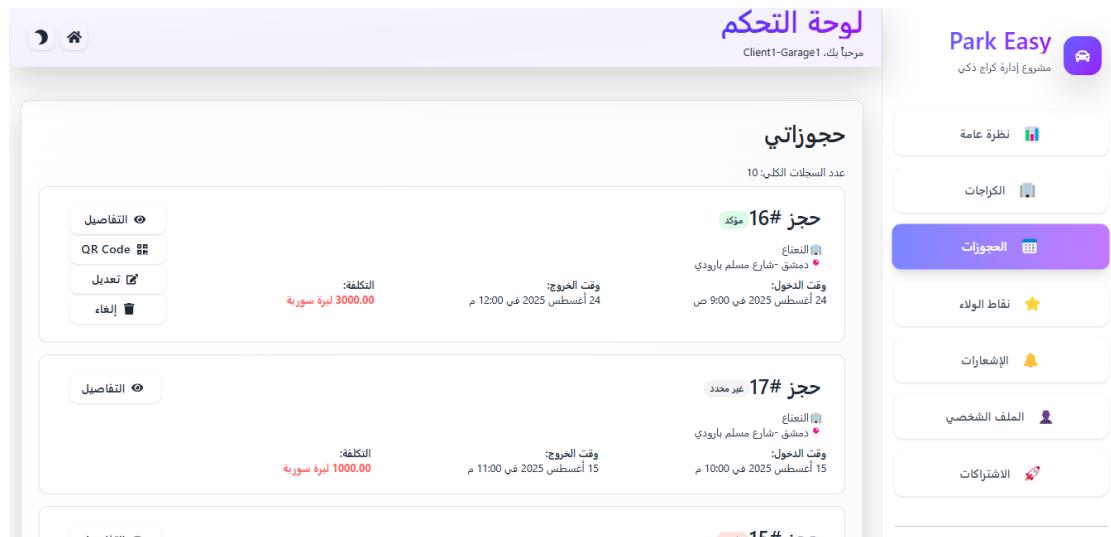
ملف الشخصي

الاشتراك

Park Easy مشروع إدارة كراج ذكي

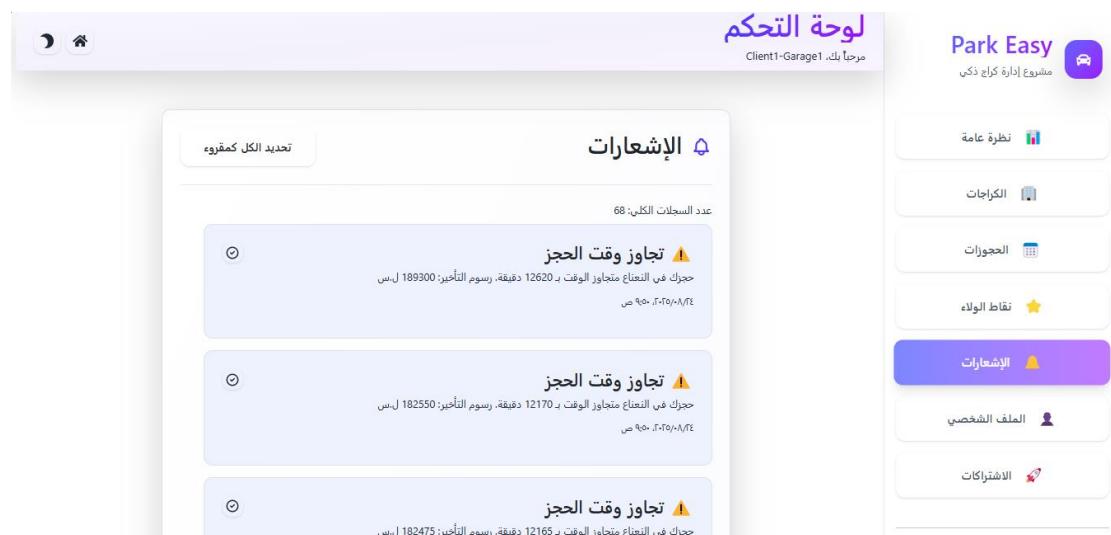
الشكل (٨.٥) – واجهة الكراجات لدى العميل

٢.٥.٥ - واجهة الحجوزات عند العميل:



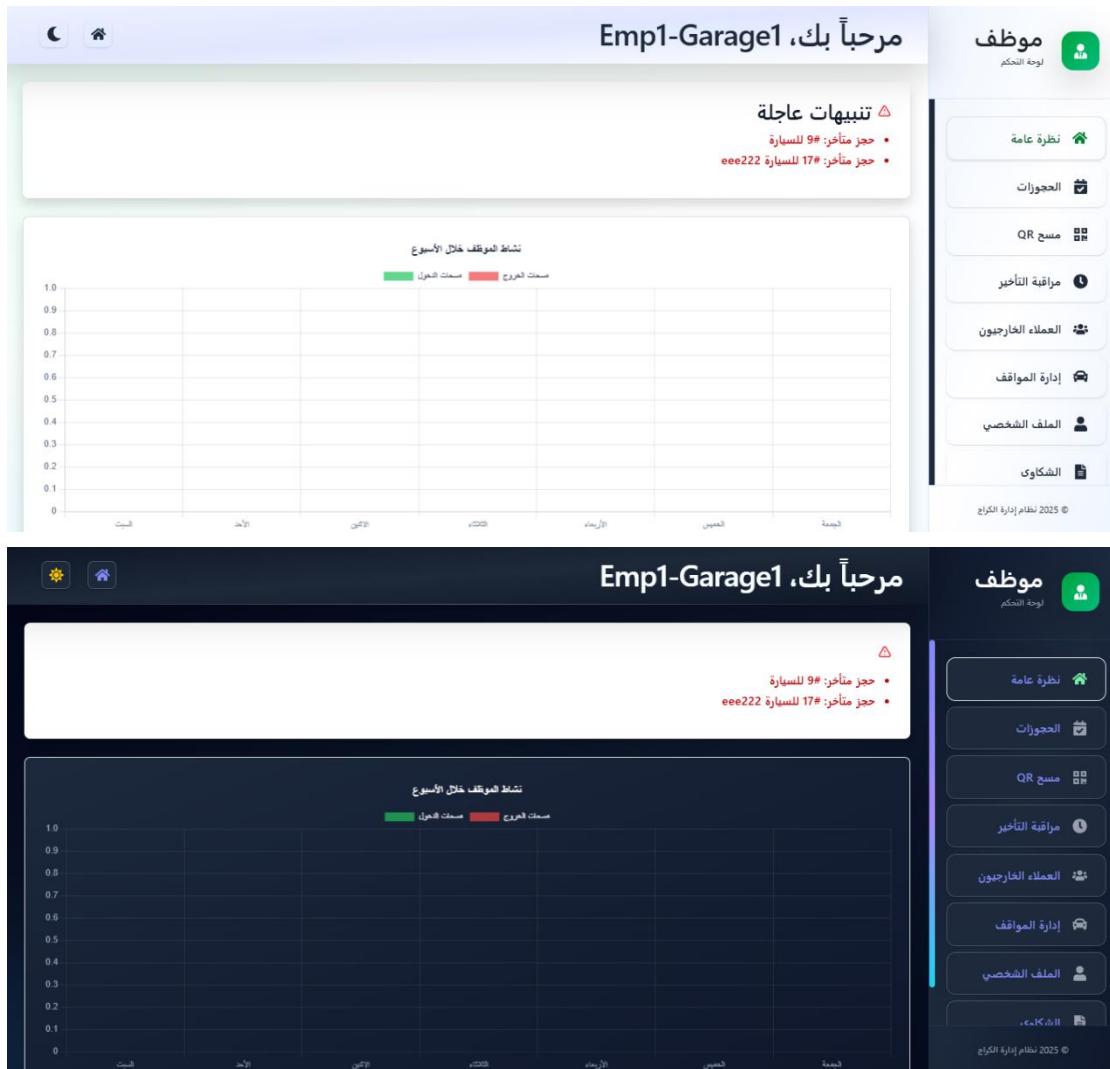
الشكل (٩.٥) - واجهة الحجوزات لدى العميل

٣.٢.٥.٥ - واجهة الاشعارات عند العميل:



الشكل (١٠.٥) - واجهة الاشعارات لدى العميل

٣.٥.٥ - واجهة الموظف الأساسية:



الشكل (١١.٥) - واجهة الموظف الأساسية

٤.٣.٥.٥ - واجهة الحجوزات عند الموظف:

إدارة الحجوزات						
الرقم	النوع	وقت الارجاع	وقت الدخول	رقم السيارة	رقم الهاتف	اسم العميل
عدد الحالات الكلية: 14						
...	موتو	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٧	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٧	ssd233	0111999111	Client1-Garage1
...	تجاوز الوقت	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٨	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٨	eee222	0111999111	Client1-Garage1
...	ملاطف	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٩	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٣٩	ssd233	0111999111	Client1-Garage1
...	ملاطف	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤٠	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤٠	ssd233	0222999111	Client2-Garage1
...	ملاطف	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤١	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤١	ssd233	0222999111	Client2-Garage1
...	ملاطف	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤٢	٢٠٢٣-٠٦-١٢ ٢٣:٥٩:٤٢	ssd233	0222999111	Client2-Garage1

الشكل (١٢.٥) - واجهة الحجوزات عند الموظف

٢.٣.٥.٥ - واجهة مسح رمز QR عند الموظف:



الشكل (١٣.٥) - واجهة مسح رمز QR عند الموظف

٣.٣.٥.٥ - واجهة مراقبة التأخير عند الموظف:



الشكل (١٤.٥) - واجهة مراقبة التأخير عند الموظف

٤.٣.٥.٥ - واجهة الملف الشخصي عند للموظف:



الشكل (١٥.٥) - واجهة الملف الشخصي عند للموظف

٥.٣.٥.٥ - واجهة الشكاوى عند الموظف:



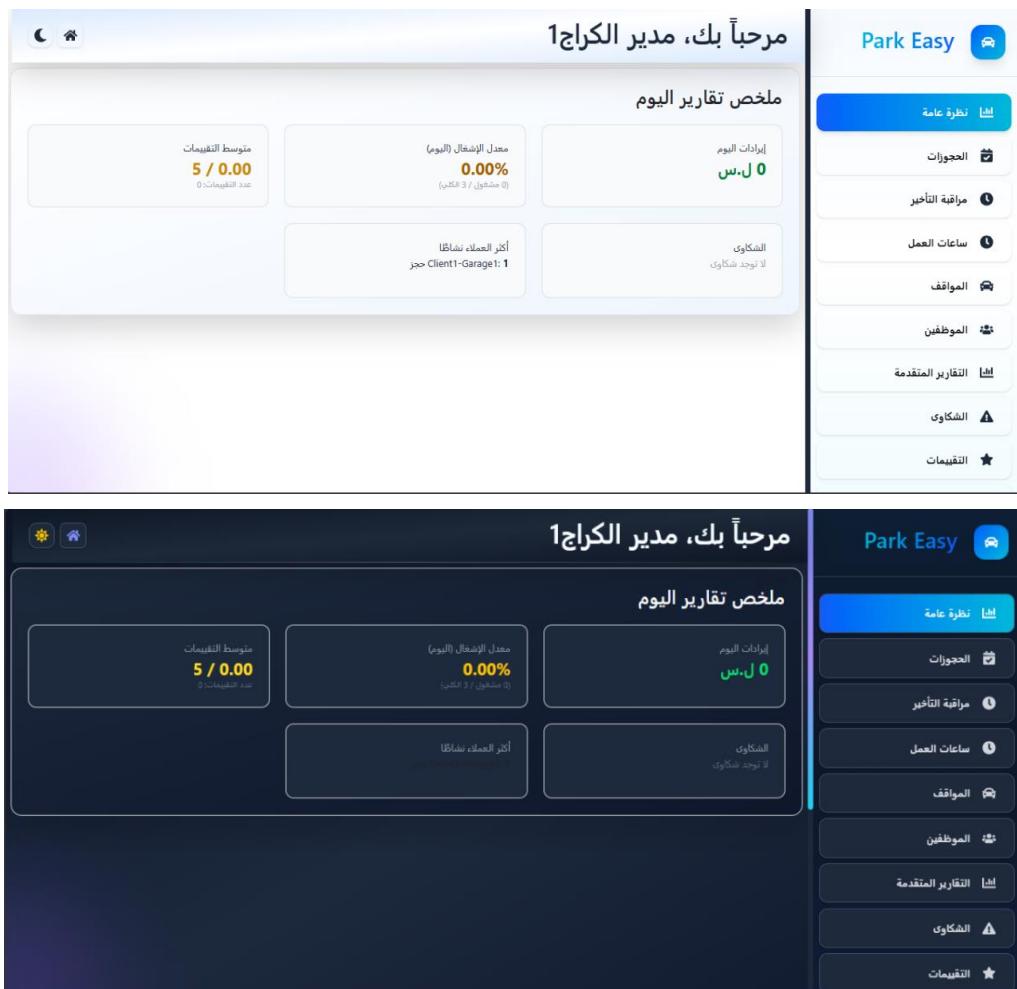
الشكل (١٦.٥) - واجهة الشكاوى عند الموظف

٦.٣.٥.٥ - واجهة عرض تفاصيل الشكوى عند الموظف:



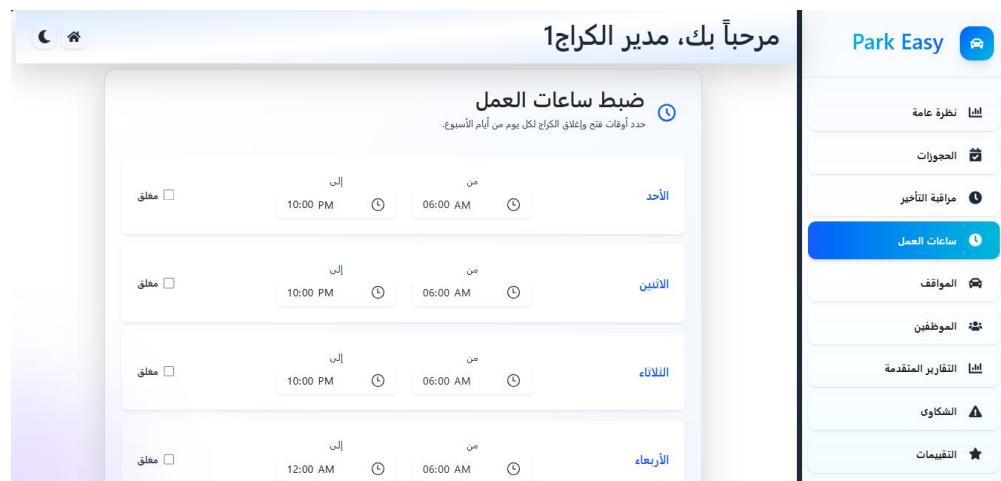
الشكل (١٧.٥) - واجهة عرض تفاصيل الشكوى عند الموظف

٤.٥.٥ - الواجهة الأساسية لمدير الكراج:



الشكل (١٨.٥) - الواجهة الأساسية لمدير الكراج

٤.٥.٦- واجهة ضبط ساعات العمل عند مدير الكراج:



الشكل (١٩.٥) - واجهة ضبط ساعات العمل عند مدير الكراج

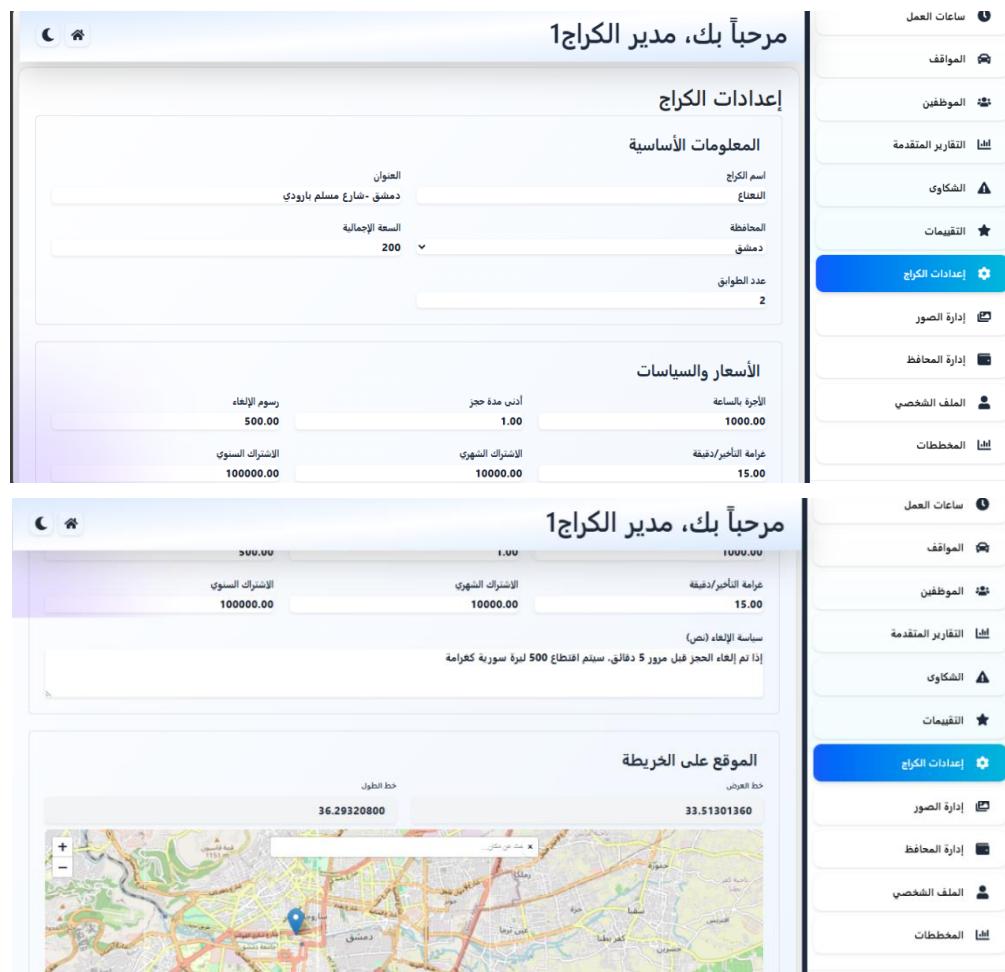
٢٠٤٥٥- واجهة إدارة الموظفين عند مدير الكراج:

الشكل (٢٠٥)- واجهة إدارة الموظفين عند مدير الكراج

٣٠٤٥٥- واجهة التقارير المقدمة لمدير الكراج:

الشكل (٢١٥)- واجهة التقارير المقدمة لمدير الكراج

٤٠٤٥٥- واجهة إعدادات الكراج عند مدير الكراج:



الشكل (٢٢.٥) – واجهة إعدادات الكراج عند مدير الكراج

٤.٥.٤-٥.٥- واجهة إدارة الصور مدير الكراج:



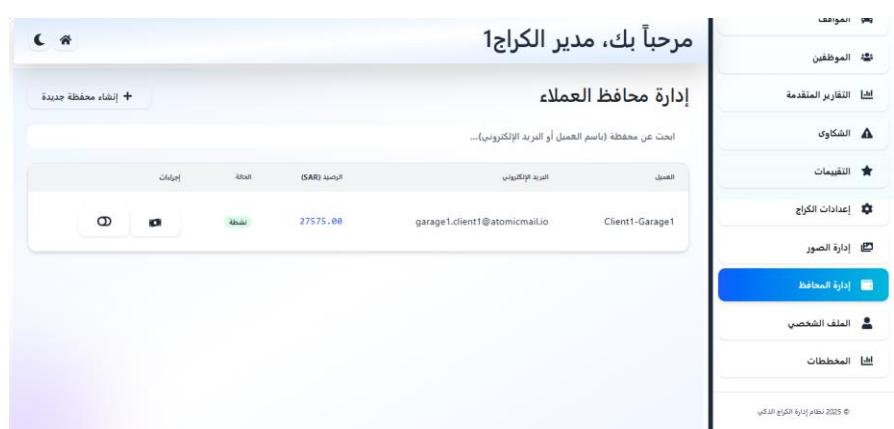
الشكل (٢٣.٥) – واجهة إدارة الصور مدير الكراج

٦.٤.٥.٥-٦.٥.٤- واجهة المخططات عند مدير الكراج:



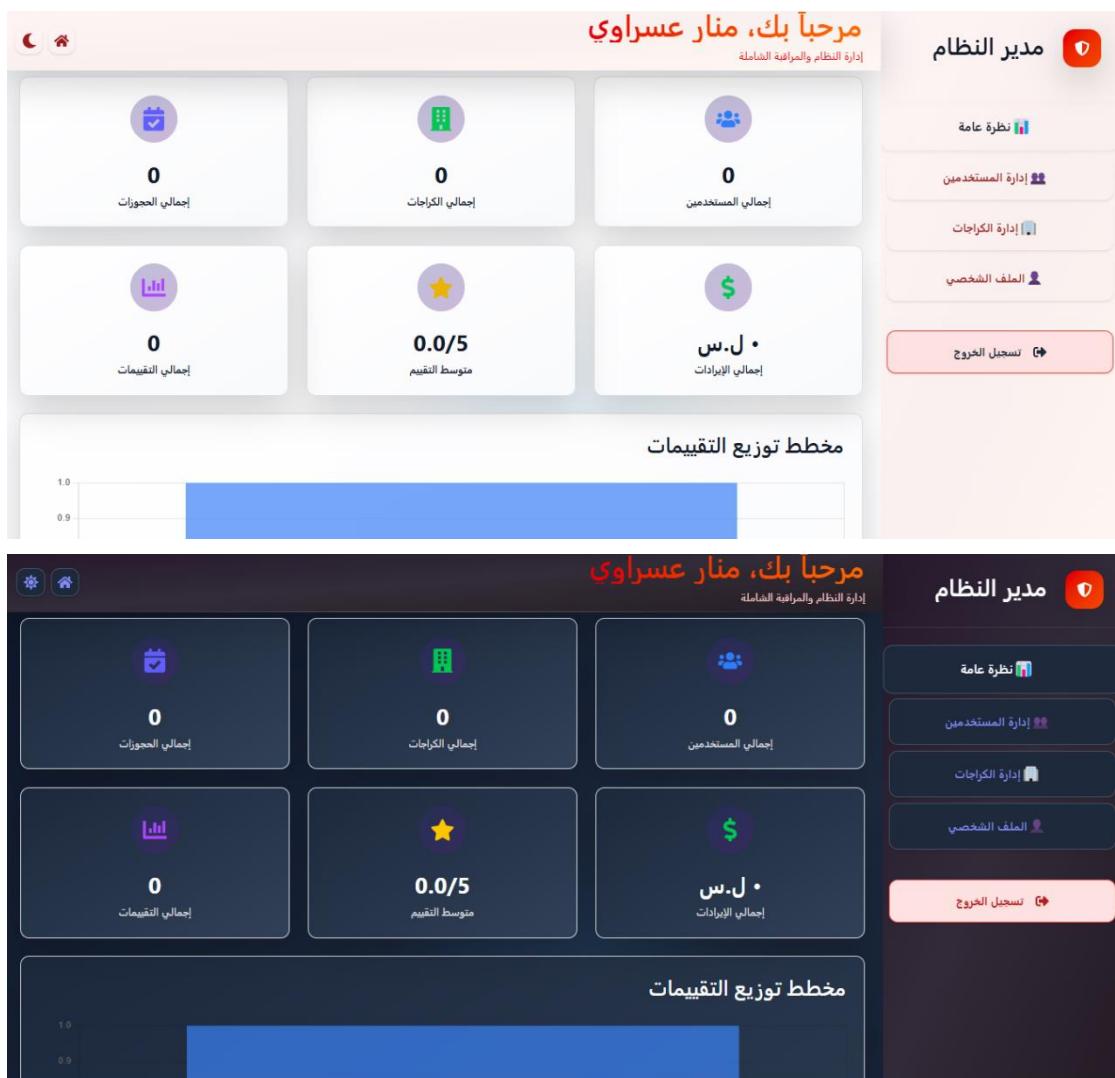
الشكل (٢٤.٥) – واجهة المخاططات عند مدير الكراج

٧.٤.٥.٥ - واجهة إدارة المحفظة مدير الكراج:



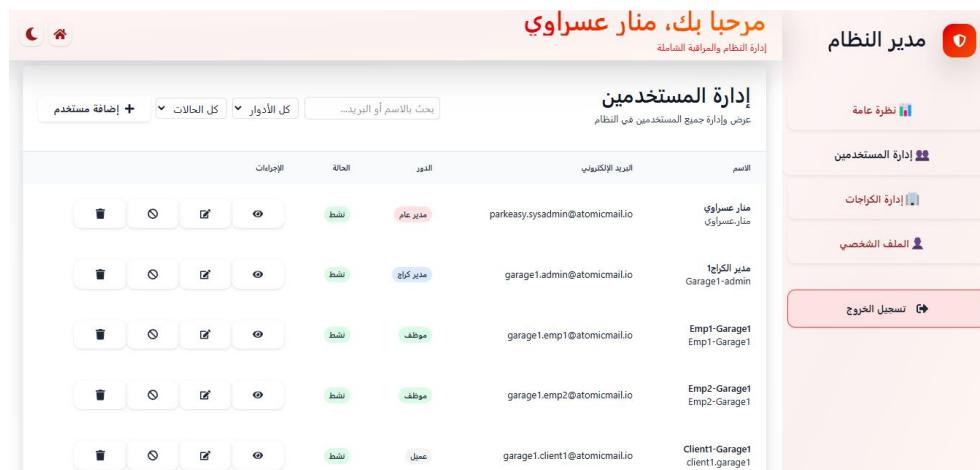
الشكل (٢٥.٥) – واجهة إدارة المحفظة مدير الكراج

٥.٥.٥ - الواجهة الأساسية لمدير النظام:



الشكل (٢٦.٥) – الواجهة الأساسية لمدير النظام

١٥.٥.٥ - واجهة إدارة المستخدمين مدير النظام:



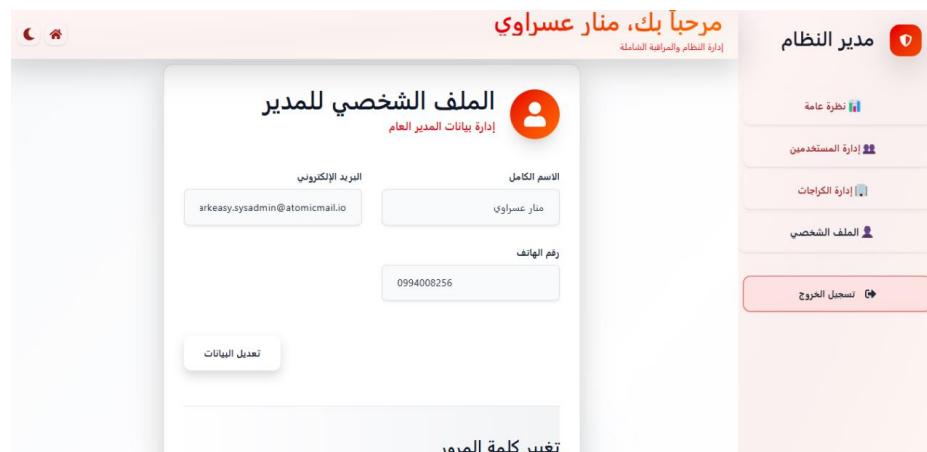
الشكل (٢٧.٥) - واجهة إدارة المستخدمين مدير النظام

٢٠.٥.٥.٥ - واجهة إدارة الكراجات عند مدير النظام:



الشكل (٢٨.٥) - واجهة إدارة الكراجات عند مدير النظام

٣٠.٥.٥.٥ - واجهة الملف الشخصي لمدير النظام:



الشكل (٢٩٠.٥) - واجهة الملف الشخصي لمدير النظام

٦٠٥.٥ - واجهة تسجيل الدخول:

الشكل (٣٠٠.٥) - واجهة تسجيل الدخول

٧.٥.٥ - واجهة إنشاء حساب:

الشكل (٣١.٥) – واجهة إنشاء حساب

٨.٥.٥ - دورة حياة طلب الحجز من قبل العميل:

العميل يختار من القائمة الجانبية في لوحة التحكم الخاصة به على زر الكراجات:



الشكل (٣٢.٥) - واجهة فتح صفحة الكراجات

يختار الكراج الذي يريد الحجز به:

النوع	الكمي	المتاح	السعر/ساعة	الميزات
عادى	2	2	2000.0000 ل.س	دخول وخروج غير محدود صالح لمدة 30 يوم QR code للدخول السريع
كهربائى	1	1	2400.0000 ل.س	شحن مجاني أثناء التوقف
VIP	1	1	3000.0000 ل.س	موقع مميز قريب من المدخل خدمة صف السيارات هزقة بالكاميرات

مجموع المواقف الكلى: 4

الشكل (٣٣.٥) - واجهة اختيار كراج للحجز

يضغط على زر الانتقال إلى الحجز:

تفاصيل الحجز

العنوان حمص - شارع الحضارة	اسم الكراج الحضارة	السعر الأساسي لليمة 2000.00 ليرة سورية/ساعة
نوع الموقف *		
اختر نوع الموقف		
0 المواقف المشغولة	4 المواقف المتاحة	4 [صافي المواقف]
طريقة الدفع		
وقت الخروج *		وقت الدخول *
mm/dd/yyyy --::-- --	mm/dd/yyyy --::-- --	رقم اللوحة
ABC-123 متاليا		
<input checked="" type="checkbox"/> إنشاء الحجز <input type="button" value="إلغاء"/>		

الشكل (٣٣.٥) – واجهة نموذج بيانات الحجز

يختار نوع الموقف الذي يريد الحجز به حيث لكل موقف سعره وميزاته:

تفاصيل الحجز

السعر الأساسي 2000.00 ليرة سورية/ساعة	العنوان حمص - شارع الحضارة	اسم الكراج الحضارة
--	-------------------------------	-----------------------

* نوع الموقف

اختر نوع الموقف

عادي- 2000.0000 ل.س/ساعة (كلي: 2، متاح: 2، مشغول: 0)
 كهربائي- 2400.0000 ل.س/ساعة (كلي: 1، متاح: 1، مشغول: 0)
 VIP- 3000.0000 ل.س/ساعة (كلي: 1، متاح: 1، مشغول: 0)

نقطات (10 نقطه/ساعة) طريقة الدفع

* وقت الخروج * وقت الدخول

رقم اللوحة مثال: ABC-123

إنشاء الحجز

الشكل (٣٤.٥) – واجهة اختيار نوع الموقف

يقوم بملء النموذج بالمعلومات كاملة من وقت الدخول ووقت الخروج وطريقة الدفع ورقم السيارة:

تفاصيل الحجز

السعر الأساسي ليرة سورية/ساعة 2000.00	العنوان الطلوان حصص - شارع الحضارة	اسم الكراج الحضارة
نوع الموقف *		
VIP - 3000.0000 ل.س/ساعة (كلي: 1، متاح: 1، مشغول: 0)		
0 المواقف المشغولة	4 المواقف المتاحة	4 إجمالي المواقف
نقطة (10 نقطه/ساعة) <input checked="" type="checkbox"/> نعم		طريقة الدفع
وقت الخروج * 08/26/2025 03:30 PM		وقت الدخول * 08/26/2025 03:00 PM
رقم اللوحة 5sd233		
ملخص الحجز المدة المحددة: 30 دقيقة نوع الموقف: VIP التكلفة الإجمالية: 1500.00 ل.س		
<input checked="" type="checkbox"/> إنشاء الحجز		الغاء

الشكل (٣٥.٥) – واجهة بيانات الحجز مع حساب التكلفة الإجمالية

وعندما يضغط على إنشاء الحجز تظهر الواجهة التالية لتأكيد الحجز :



الشكل (٣٦.٥) - واجهة سؤال تأكيد الحجز

بالضغط على زر تأكيد يظهر إشعار تم إنشاء الحجز بنجاح وينتقل لواجهة الحجوزات الحالية والسابقة:

الشكل (٣٧.٥) - واجهة تأكيد الحجز

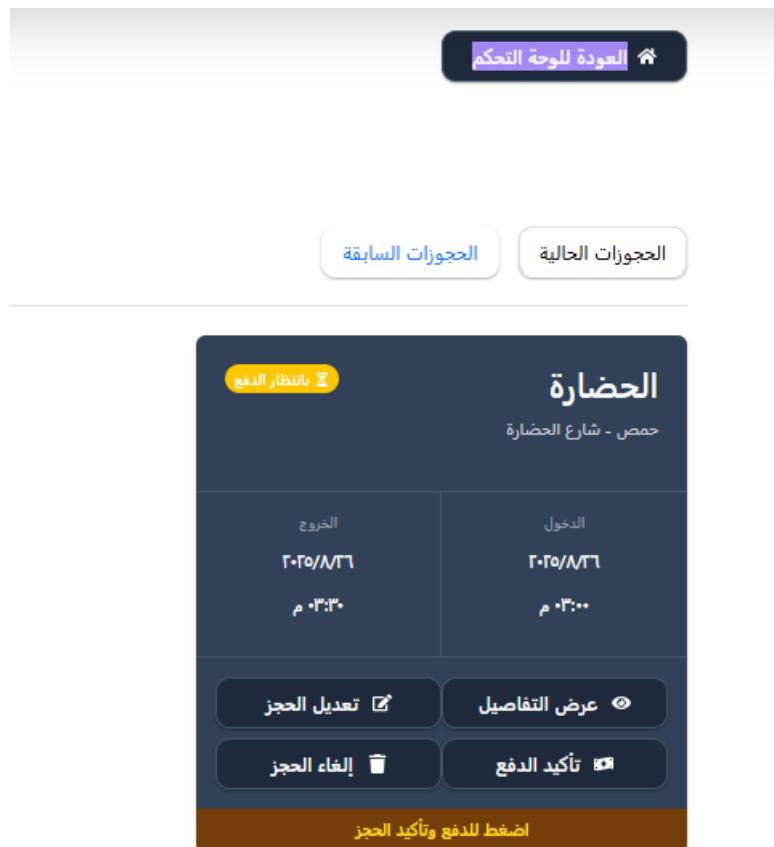
في حال كان لديه حجز آخر نشط في نفس الكراج يرسل إشعار بوجود حجز آخر نشط وينبه من الحجز:

The screenshot displays a mobile application interface for managing car reservations. At the top, there is a message in a red-bordered box: "لديك حجز تنشط أو مؤكدة في هذا الكراج، لا يمكن إضافة حجز جديد." (You have an active or confirmed booking in this garage, cannot add a new booking). Below this, the form fields include:

- العنوان:** حمص - شارع الحضارة
- اسم الكراج:** الحضارة
- السعر الأساسي:** 2000.00 ليرة سورية/ساعة
- نوع الموقف:** عادي - 2000.0000 ل.س/ساعة (كلي: 2, متاح: 2, مشغول: 0)
- المواقف المنشورة:** 1 (الموقف المشفورة)
- المواقف المتاحة:** 3 (الموقف المتاحة)
- إجمالي الموقف:** 4 (إجمالي الموقف)
- طريقة الدفع:** (선택) (선택/ساعة)
- وقت الدخول:** 08/26/2025 03:15 PM
- وقت الخروج:** 08/26/2025 04:15 PM
- رقم اللوحة:** www345
- ملخص الحجز:**
 - المدة المحددة: 1 ساعة
 - نوع الموقف: عادي
 - التكلفة الإجمالية: 2200.00 ل.س
- Buttons:** [إنشاء الحجز] (Create Booking) and [الغاء] (Cancel).

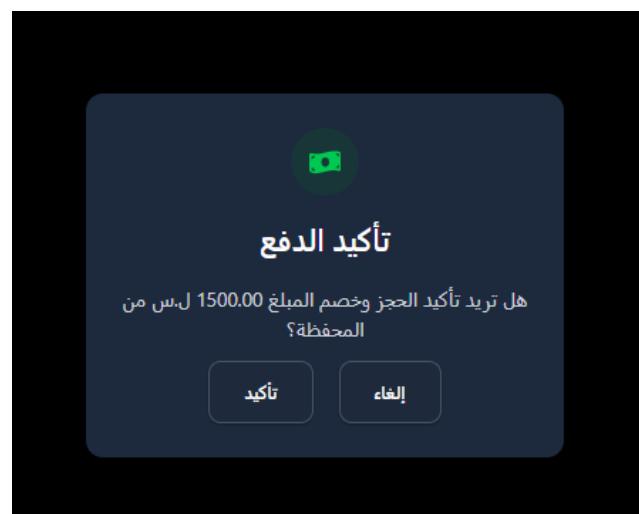
الشكل (٣٨.٥) – واجهة وجود حجز نشط

يكون الحجز في حالة انتظار الدفع:



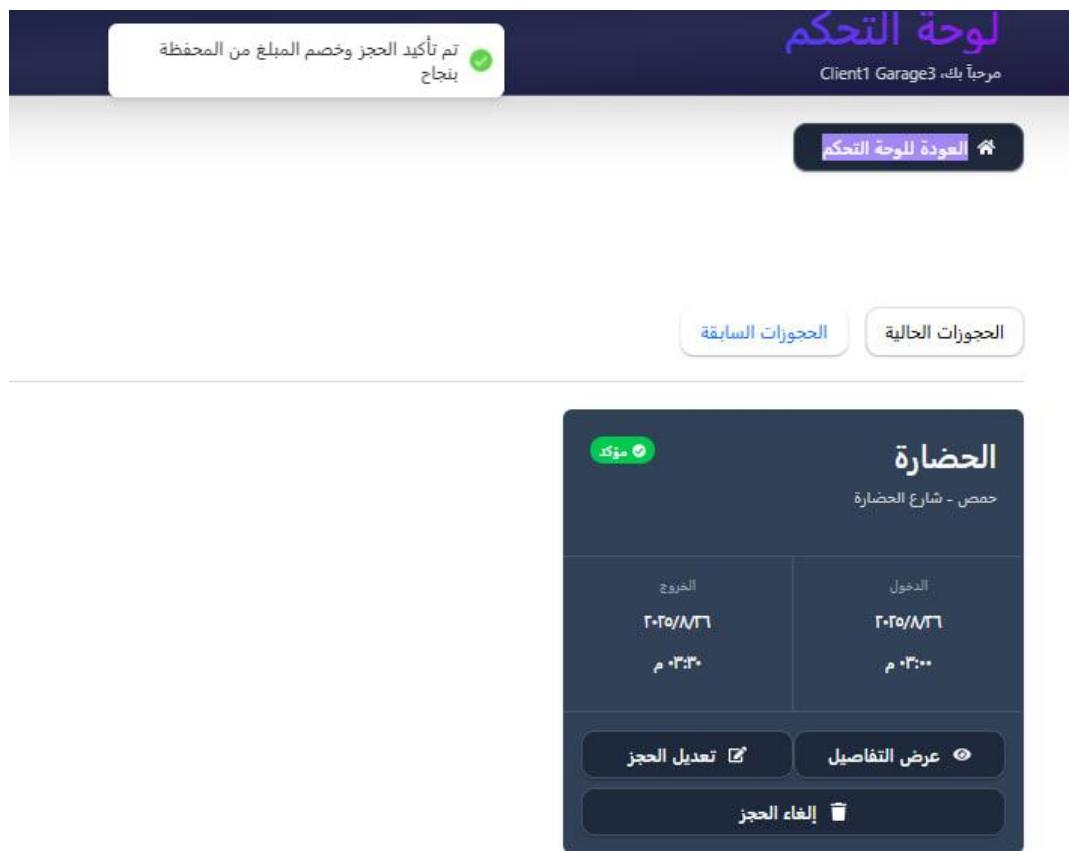
الشكل (٣٩.٥) – واجهة حالة الحجز قبل الدفع

يختار الخيار تأكيد الدفع ليتم عملية الحجز فيظهر الإشعار التالي:



الشكل (٤٠.٥) – واجهة تأكيد الدفع

وعند الضغط على زر تأكيد يظهر إشعار بإتمام العملية بنجاح وخصم الرصيد من المحفظة الخاصة بالعميل وتصبح حالة الحجز مؤكداً:



الشكل (٤١.٥) - واجهة حالة الحجز بعد الدفع

بالعودة إلى لوحة التحكم واختيار الحجوزات من القائمة الجانبية يظهر الحجز المؤكد مع خيارات التعديل أو الحصول على رمز QR للدخول:



الشكل (٤٢.٥) - واجهة معلومات الحجز في قائمة الحجوزات

عند الضغط على زر يحصل على رمز QR الخاص بهذا الحجز:



الشكل (٤.٣٥) – واجهة رمز QR

وبعد مسح الخول عند الموظف لهذا الحجز تصبح حالة الحجز نشط بالشكل التالي:

 A screenshot of a mobile application interface titled "لوحة التحكم" (Control Panel) in Arabic. The title bar also includes "Client1 Garage3" and "مرحباً بك". The main screen shows booking details for "تفاصيل الحجز #25". There are two main sections: "معلومات الكراج" (Parking Lot Information) and "المعلومات الأساسية" (Basic Information).

- معلومات الكراج:**
 - اسم الكراج: **الحضارة**
 - الموقع: **العنوان**
 - سعر الساعة: **2000 ليرة سورية**
- المعلومات الأساسية:**
 - وقت الدخول الممحوز: **26 أغسطس 2025 في 3:00 م**
 - وقت الخروج الممحوز: **26 أغسطس 2025 في 3:30 م**
 - وقت الدخول الفعلي: **26 أغسطس 2025 في 3:01 م**
 - وقت الخروج الفعلي: **لم يتم الخروج بعد**
 - المدة الممحوزة: **30 دقيقة**
 - المدة الفعلية: **غير متوفر**

الشكل (٤.٤) – واجهة تفاصيل الحجز بعد مسح الدخول عند العميل

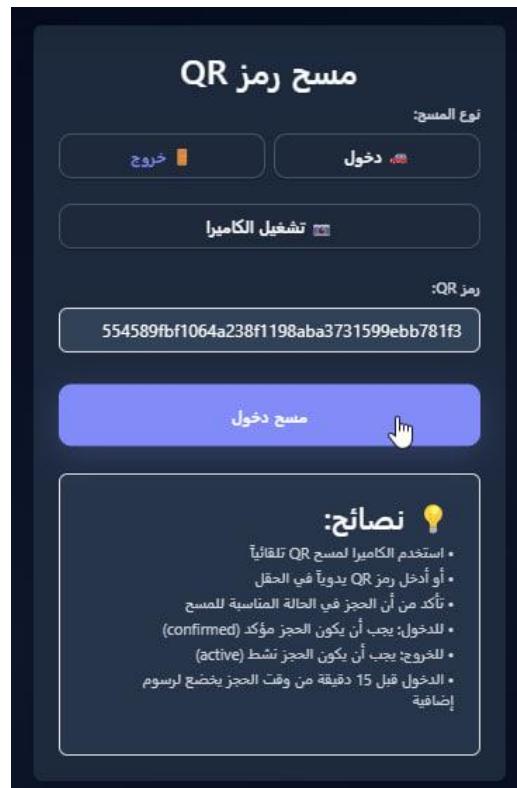
٩.٥.٥ - دورة حياة طلب الحجز عند الموظف:

الموظفي في لوحة التحكم الخاصة به يختار مسح QR ويختار دخول:



الشكل (٤٥.٥) - واجهة مسح QR عند الموظف

يمسح رمز QR من جوال العميل على بوابة الكراج إما عن طريق الكاميرا أو عن طريق إرسال الرمز:



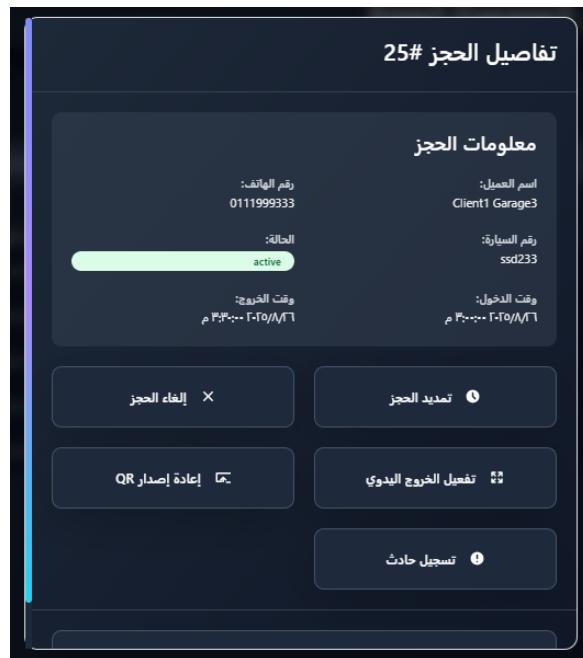
الشكل (٤٥) - واجهة مسح QR دخول

عند الضغط على مسح الدخول يعطي رسالة تأكيد الدخول بنجاح:



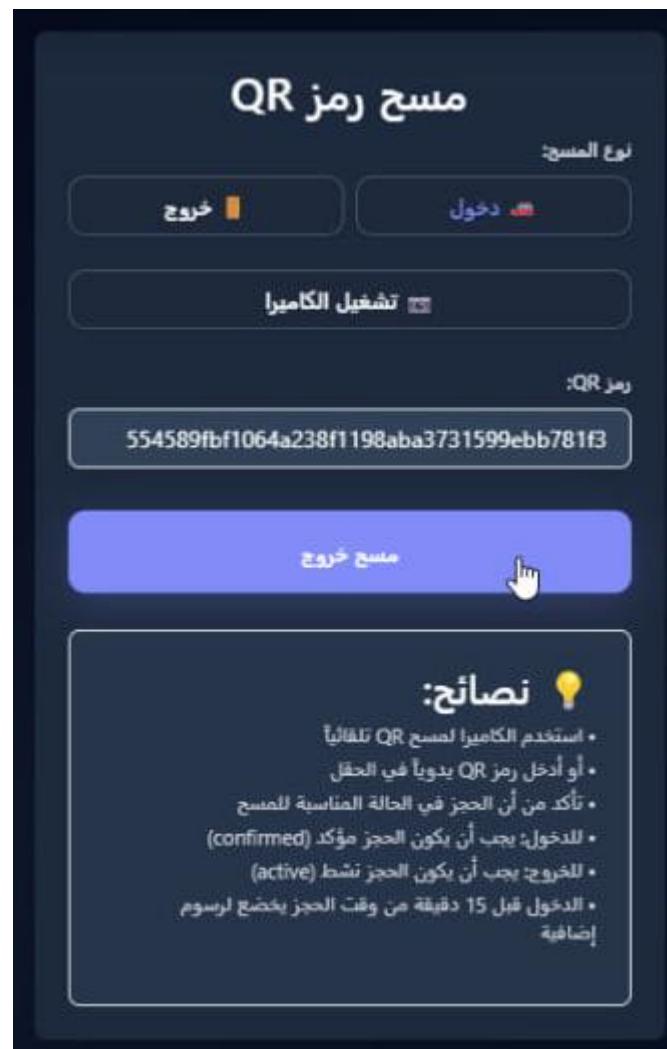
الشكل (٤٦.٥) – واجهة تأكيد الدخول للكراج

وبالعودة إلى قائمة الحجوزات يصبح الحجز نشط:



الشكل (٤٧.٥) – واجهة تفاصيل الحجز بعد مسح الدخول عند الموظف

وبعد انتهاء الحجز ويريد العميل المغادرة يقوم الموظف بمسح QR الخروج:



الشكل (٤٨.٥) - واجهة مسح QR خروج

تظهر رسالة تأكيد بإتمام العملية بنجاح:



الشكل (٤٩.٥) – واجهة تأكيد الخروج من الكراج

وتصبح حالة الحجز مكتمل عند كل من العميل والموظف:



الشكل (٥٠.٥) – واجهة تفاصيل الحجز بعد مسح الخروج عند الموظف

١٠٠.٥.٥ - حالات ثانوية للحجز:

١،١٠،٥,٥-دخول مبكر:

إذا قام العميل بإنشاء حجز وأتى مبكراً أكثر من ٥ دقائق تظهر رسالة خطأ تمنعه من الدخول:



الشكل (٥١.٥) – واجهة دخول مبكر أكثر من ٥ دقائق

١،١٠،٥,٥-الدخول مبكر أقل من ٥ دقائق:

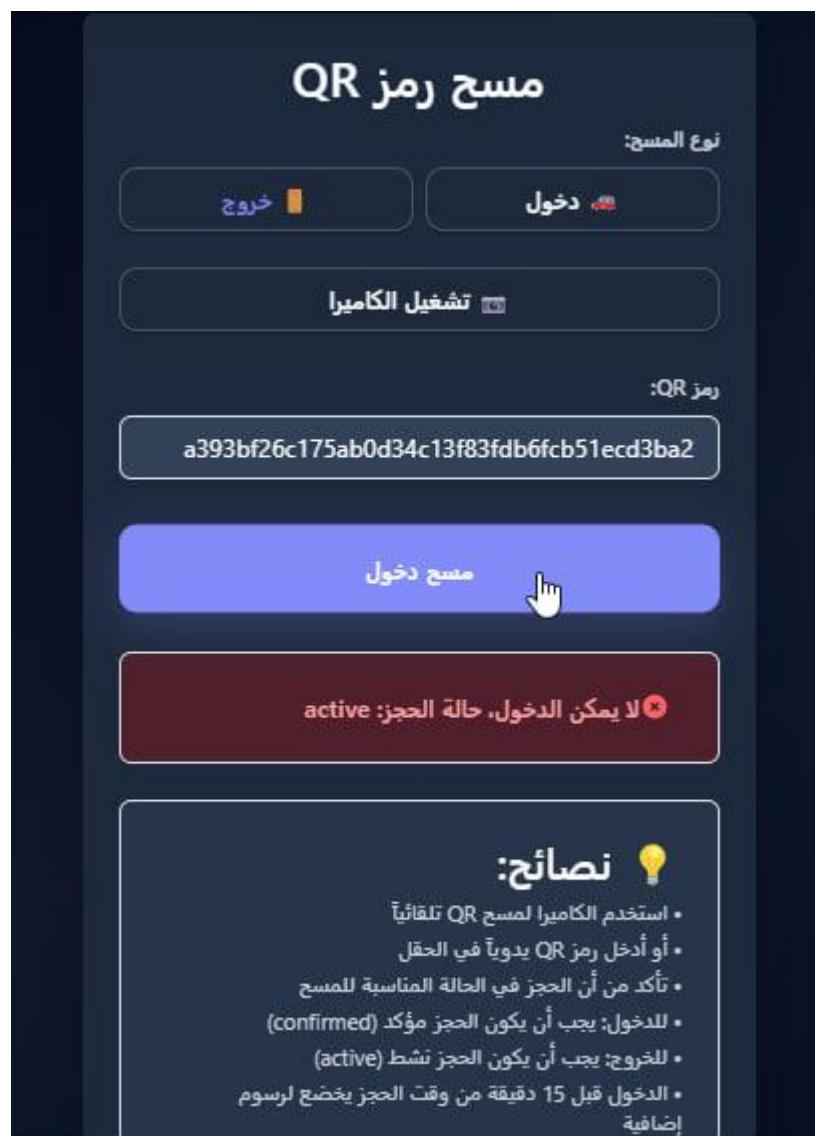
يتم مسح الكود بنجاح مع ارسال تتبّيه دخول مبكر واحتساب الدقائق وخصم سعرها من المحفظة تلقائياً:



الشكل (٥٢.٥) – واجهة دخول مبكر أقل من ٥ دقائق

٣-١٠،٥-مسح QR لحجز نشط:

تظهر رسالة خطأ تفيد بأنه لا يمكن الدخول لأن الحجز نشط كالتالي:



الشكل (٥٣.٥) - واجهة مسح دخول لحجز نشط

٤-تأخر عن الموعد المحجوز:

أولاً يتم إرسال إشعار تتبّيه تلقائي وإيميل إلى بريد العميل عند بقاء ربع ساعة من وقت الحجز ويظهر عند الموظف بقائمة مراقبة التأخير بأنه حجوزات قادمة



الشكل (٥٤.٥) - واجهة حجوزات قادمة

وإذا انتهت المدة المحددة يظهر إشعار في واجهة الموظف بوجود حجز متاخر وعند عرض التفاصيل يظهر زر إرسال تحذير عند الضغط عليه يتم إيميل تحذير للعميل بأنه تجاوز المدة المحددة ويتم احتساب عدد دقائق التأخير والرسوم المرتبطة على ذلك واقطاعها من المحفظة تلقائياً:



الشكل (٥٥.٥) - واجهة حجوزات متاخرة

١١.٥.٥ - البريد الإلكتروني الذي يصل للعميل من النظام:

١.١١.٥.٥ - بريد ترحيب بالانضمام للمنصة عند إنشاء الحساب:



الشكل (٥٦.٥) - واجهة بريد ترحبي

٢.١١.٥.٥ - بريد تأكيد الحجز عند إتمام الدفع متضمن رمز QR:



الشكل (٥٧.٥) – واجهة بريد تأكيد الحجز

٣.١١.٥.٥ - بريد إلغاء الحجز مع تفاصيل المبلغ المسترد والشخص الذي حصل حسب سياسة الإلغاء:



الشكل (٥٨.٥) – واجهة بريد إلغاء حجز

٤.١١.٥.٥ - بريد تتبّيه الحجز تلقائي في آخر ربع ساعة من مدة الحجز:



الشكل (٥٩.٥) – واجهة بريد تنبيه حجز

٥.١١.٥.٥- بريد تجاوز الوقت المحدد عند انتهاء مدة الحجز وعدم خروج العميل:



الشكل (٦٠.٥) – واجهة بريد تجاوز الوقت

٦.١١.٥.٥ - بريد استعادة كلمة المرور في حال نسيانها:



الشكل (٦١.٥) - واجهة بريد إعادة تعيين كلمة المرور

الخاتمة

يتمثل هذا المشروع خطوة نوعية نحو تطوير منظومة متكاملة لإدارة حجوزات مواقف السيارات، تجمع بين الكفاءة التشغيلية وسلامة تجربة المستخدم، مستندة إلى أسس علمية وتقنية راسخة. فقد تناول العمل مراحل التطوير بدءاً من تحديد المشكلة البحثية وتحليل المتطلبات، مروراً بدراسة النظم المشابهة واستخلاص أفضل الممارسات، وصولاً إلى تصميم وظيفي متكامل للمنصة يشمل البنية البرمجية، وقاعدة البيانات، وآليات التفاعل بين المكونات.

أثبتت النتائج التحليلية أن دمج تقنيات الويب الحديثة (HTML, CSS, React.js, Node.js, Express.js) مع أنظمة الدفع الإلكتروني، ورموز الاستجابة السريعة (QR)، وخدمات الإشعارات، يوفر بيئه تشغيلية مرنة وقابلة للتتوسيع، قادرة على تلبية احتياجات مختلف الأطراف المعنية. كما أن اعتماد منهجية الشلال التكراري أتاح ضبط جودة المخرجات وتقليل المخاطر عبر التكرار المرحلي ومعالجة الأخطاء مبكراً.

إن القيمة المضافة للمشروع لا تقتصر على تحسين إدارة المواقف فحسب، بل تمتد لتسهم في دعم توجهات المدن الذكية، من خلال تقليل الازدحام المروري، وخفض الانبعاثات، وتعزيز الاستدامة الحضرية.

وبذلك، يمكن القول إن المنصة المقترحة تمثل نموذجاً عملياً قابلاً للتطبيق والتوسيع، يجمع بين الرؤية المستقبلية والجدى التقنية، ويسس لمرحلة جديدة من التحول الرقمي في قطاع إدارة مواقف السيارات، مع قابلية التكيف مع التطورات التقنية والاحتياجات المتغيرة للمستخدمين.

الآفاق المستقبلية

يمكن بلوحة آفاق المشروع المستقبلية في النقاط التالية:

- تطوير تطبيق موبايل متكامل للمنصة يوفر بيئة تفاعلية متقدمة للمستخدمين عبر واجهة استخدام بديهية، وإشعارات فورية، وخدمات تحديد الموقع الجغرافي (GPS) لعرض المواقف المتاحة وحجزها وتتبعها لحظياً، مع إمكانية دمج تقنيات الواقع المعزز في المستقبل لتحسين تجربة المستخدم.
- دمج تقنيات الذكاء الاصطناعي وإنترنت الأشياء (IoT) لتجهيز المواقف بحساسات ذكية متصلة بالشبكة، قادرة على رصد حالتها (شاغر/مشغول) ومعالجة البيانات محلياً لزيادة سرعة الاستجابة وتقليل استهلاك الموارد، مع إمكانية استخدام خوارزميات تنبؤية لتقدير نسب الإشغال مستقبلاً بناءً على أنماط الاستخدام.
- تطوير نموذج رؤية حاسوبية للتعرف على لوحات المركبات تدريب نموذج ذكاء اصطناعي متقدم باستخدام تقنيات التعلم العميق (Deep Learning) للتعرف بدقة على أرقام اللوحات، بما يمكن من أتمتة عمليات الدخول والخروج، وتسجيل الإحصاءات، وربطها بملفات المشتركين، مع تعزيز أمن المنظومة عبر اكتشاف المركبات غير المصرح لها.
- إدارة المحافظ الرقمية وتوسيع خيارات الدفعربط المنصة بأنظمة الدفع الإلكتروني مثل Syriatel Cash والحسابات البنكية المحلية، مع تطبيق بروتوكولات تشفير متقدمة لضمان أمان وسرية المعاملات، وإضافة خدمات تعبئة الرصيد والتحويل المباشر لتسهيل عمليات الدفع للمستخدمين.

الملحقات

١. الصف : (audit_logs)

```
CREATE TABLE `audit_logs` (
  `log_id` int(11) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `action` varchar(255) NOT NULL,
  `entity_type` varchar(50) DEFAULT NULL,
  `entity_id` int(11) DEFAULT NULL,
  `details` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_bin DEFAULT
NULL CHECK (json_valid(`details`)),
  `created_at` timestamp NOT NULL DEFAULT current_timestamp()
)
```

٢. الصف : (bookings)

```
CREATE TABLE `bookings` (
  `booking_id` int(11) NOT NULL,
  `customer_id` int(11) DEFAULT NULL,
  `garage_id` int(11) NOT NULL,
  `spot_id` int(11) NOT NULL,
  `subscription_id` int(11) DEFAULT NULL,
  `discount_id` int(11) DEFAULT NULL,
  `vehicle_number` varchar(20) DEFAULT NULL,
  `payment_status` enum('paid','unpaid') DEFAULT 'unpaid',
  `payment_time` datetime DEFAULT NULL,
  `qr_code_identifier` varchar(255) NOT NULL,
  `verification_code` varchar(10) DEFAULT NULL,
  `qr_code_expiry` timestamp NULL DEFAULT NULL,
  `entry_time_actual` timestamp NULL DEFAULT NULL,
  `exit_time_actual` timestamp NULL DEFAULT NULL,
  `booked_entry_time` datetime NOT NULL,
  `booked_exit_time` datetime DEFAULT NULL,
  `booked_duration_hours` decimal(10,2) NOT NULL,
  `booking_fee` decimal(10,2) NOT NULL,
  `additional_charges` decimal(10,2) DEFAULT 0.00,
  `cancellation_fee` decimal(10,2) DEFAULT 0.00,
  `total_charges` decimal(10,2) DEFAULT 0.00,
  `overstay_duration_minutes` int(11) DEFAULT 0,
  `overstay_fee` decimal(10,2) DEFAULT 0.00,
```

```

`loyalty_discount_applied` decimal(10,2) DEFAULT 0.00,
`cancellation_time` timestamp NULL DEFAULT NULL,
`is_cancelled_within_grace_period` tinyint(1) DEFAULT NULL,
`status` enum('pending_payment','confirmed','active','completed','cancelled','overstayed') NOT NULL,
`expiration_warning_sent` tinyint(1) DEFAULT 0,
`notes` text DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)

```

٢. الصف (complaints):

```

CREATE TABLE `complaints` (
`complaint_id` int(11) NOT NULL,
`customer_id` int(11) NOT NULL,
`garage_id` int(11) DEFAULT NULL,
`booking_id` int(11) DEFAULT NULL,
`subject` varchar(255) NOT NULL,
`description` text NOT NULL,
`status` enum('new','in_progress','resolved','closed','rejected')
DEFAULT 'new',
`priority` enum('low','medium','high') DEFAULT 'medium',
`filed_at` timestamp NOT NULL DEFAULT current_timestamp(),
`resolved_at` timestamp NULL DEFAULT NULL,
`resolved_by_user_id` int(11) DEFAULT NULL,
`notes_from_resolver` text DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)

```

٤. الصف (external_customers):

```

CREATE TABLE `external_customers` (
`external_customer_id` int(11) NOT NULL,
`booking_id` int(11) NOT NULL,
`full_name` varchar(255) NOT NULL,
`vehicle_number` varchar(20) NOT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp()
)

```

٥. الصنف (garages)

```
CREATE TABLE `garages` (
  `garage_id` int(11) NOT NULL,
  `manager_id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `address` text NOT NULL,
  `governorate` varchar(255) NOT NULL,
  `latitude` decimal(10,8) DEFAULT NULL,
  `longitude` decimal(11,8) DEFAULT NULL,
  `total_capacity` int(11) NOT NULL,
  `hourly_rate` decimal(10,2) NOT NULL,
  `floors_number` int(11) NOT NULL,
  `working_hours` longtext CHARACTER SET utf8mb4 COLLATE utf8mb4_bin
  DEFAULT NULL CHECK (json_valid(`working_hours`)),
  `available_spots` int(11) DEFAULT 0,
  `rating_average` decimal(3,2) DEFAULT 0.00,
  `cancellation_policy` text DEFAULT NULL,
  `min_booking_hours` decimal(4,2) DEFAULT 1.00,
  `cancellation_fee` decimal(10,2) DEFAULT 0.00,
  `overstay_rate_per_minute` decimal(10,2) DEFAULT 0.00,
  `valet_parking_enabled` tinyint(1) NOT NULL DEFAULT 0,
  `car_wash_enabled` double NOT NULL DEFAULT 0,
  `ev_charging_enabled` double NOT NULL DEFAULT 0,
  `monthly_subscription_rate` decimal(10,2) DEFAULT NULL,
  `yearly_subscription_rate` decimal(10,2) DEFAULT NULL,
  `isActive` tinyint(1) DEFAULT 1,
  `contact_phone` varchar(20) DEFAULT NULL,
  `contact_email` varchar(255) DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
  current_timestamp()
)
```

٦. الصنف (garage_access_logs)

```
CREATE TABLE `garage_access_logs` (
  `log_id` int(11) NOT NULL,
  `garage_id` int(11) NOT NULL,
  `booking_id` int(11) DEFAULT NULL,
  `action` enum('entry','exit') NOT NULL,
```

```

`timestamp` timestamp NOT NULL DEFAULT current_timestamp(),
`scanned_by` int(11) DEFAULT NULL,
`notes` text DEFAULT NULL
)

```

٧. الصف : (garage_employees)

```

CREATE TABLE `garage_employees` (
`garage_employee_id` int(11) NOT NULL,
`user_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`role` enum('supervisor','scanner') DEFAULT NULL,
`start_date` date DEFAULT NULL,
`end_date` date DEFAULT NULL,
`is_active` tinyint(1) DEFAULT 1,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)

```

٨. الصف : (garage_images)

```

CREATE TABLE `garage_images` (
`image_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`image_url` varchar(255) NOT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp()
)

```

٩. الصف : (incidents)

```

CREATE TABLE `incidents` (
`incident_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`reported_by_employee_id` int(11) NOT NULL,
`booking_id` int(11) DEFAULT NULL,
`description` text NOT NULL,
`incident_time` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp(),
`status` enum('reported','under_review','resolved') DEFAULT
'reported',
`created_at` timestamp NOT NULL DEFAULT current_timestamp()
)

```

١٠. الصف : (loyalty_history)

```

CREATE TABLE `loyalty_history` (
    `history_id` int(11) NOT NULL,
    `customer_id` int(11) NOT NULL,
    `points_change` int(11) NOT NULL,
    `transaction_type` enum('booking_completion','subscription_purchase','redeem_for_credit','manual_adjustment','refund') NOT NULL,
    `reference_id` int(11) DEFAULT NULL,
    `description` varchar(255) DEFAULT NULL,
    `created_at` timestamp NOT NULL DEFAULT current_timestamp()
)

```

١١. الصنف (notifications)

```

CREATE TABLE `notifications` (
    `id` int(11) NOT NULL,
    `user_id` int(11) NOT NULL,
    `related_entity_id` int(11) NOT NULL,
    `type` enum('email','in_app') NOT NULL,
    `title` varchar(255) NOT NULL,
    `message` text NOT NULL,
    `status` enum('pending','sent','read') DEFAULT 'pending',
    `related_entity` enum('booking','transaction','register','rating','complaint','promotion','loyalty') DEFAULT NULL,
    `sent_at` timestamp NOT NULL DEFAULT current_timestamp(),
    `created_at` timestamp NOT NULL DEFAULT current_timestamp()
)

```

١٢. الصنف (parking_spots)

```

CREATE TABLE `parking_spots` (
    `parking_spot_id` int(11) NOT NULL,
    `garage_id` int(11) NOT NULL,
    `floor_number` int(11) NOT NULL,
    `spot_number` varchar(50) NOT NULL,
    `spot_type` enum('normal','electric','vip') DEFAULT 'normal',
    `status` enum('available','occupied','under_maintenance','closed') DEFAULT 'available',
    `price_modifier` decimal(5,2) DEFAULT 1.00,
    `is_active` tinyint(1) DEFAULT 1,
    `last_booked_at` timestamp NOT NULL DEFAULT current_timestamp(),
    `maintenance_start_date` timestamp NULL DEFAULT NULL,
)

```

```

`maintenance_end_date` timestamp NULL DEFAULT NULL,
`last_status_updated_by_user_id` int(11) DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)

```

:الصف (ratings) .١٣

```

CREATE TABLE `ratings` (
`rating_id` int(11) NOT NULL,
`customer_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`booking_id` int(11) DEFAULT NULL,
`rating` int(11) NOT NULL CHECK (`rating` >= 1 and `rating` <= 5),
`comment` text DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)

```

:الصف (subscriptions) .١٤

```

CREATE TABLE `subscriptions` (
`subscription_id` int(11) NOT NULL,
`customer_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`subscription_type` enum('monthly','yearly') NOT NULL,
`start_date` date NOT NULL,
`end_date` date NOT NULL,
`status` enum('active','expired','cancelled') DEFAULT 'active',
`amount_paid` decimal(10,2) NOT NULL,
`cancellation_date` timestamp NULL DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp(),
`qr_code_identifier` varchar(255) DEFAULT NULL
)

```

:الصف (transactions) .١٥

```

CREATE TABLE `transactions` (
`transaction_id` int(11) NOT NULL,

```

```

`wallet_id` int(11) DEFAULT NULL,
`booking_id` int(11) DEFAULT NULL,
`amount` decimal(10,2) NOT NULL,
`transaction_type` enum('booking_fee','top_up','refund','overstay_charge','cancellation_fee','system_adjustment','subscription_payment') NOT NULL,
`payment_method` enum('e_wallet','cash') NOT NULL,
`status` enum('pending','completed','failed') NOT NULL DEFAULT 'pending',
`description` varchar(255) DEFAULT NULL,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
)

```

: (users) . ١٦ . الصف

```

CREATE TABLE `users` (
`user_id` int(11) NOT NULL,
`full_name` varchar(255) DEFAULT NULL,
`username` varchar(255) NOT NULL,
`password_hash` varchar(255) NOT NULL,
`email` varchar(255) NOT NULL,
`phone` varchar(20) DEFAULT NULL,
`role` enum('admin','garage_admin','employee','customer') NOT NULL,
`isActive` tinyint(1) DEFAULT 1,
`last_login` timestamp NULL DEFAULT NULL,
`reset_password_token` varchar(255) DEFAULT NULL,
`reset_password_expires` timestamp NULL DEFAULT NULL,
`total_loyalty_points` int(11) DEFAULT 0,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
)

```

: (wallets) . ١٧ . الصف

```

CREATE TABLE `wallets` (
`wallet_id` int(11) NOT NULL,
`user_id` int(11) NOT NULL,
`garage_id` int(11) NOT NULL,
`balance` decimal(10,2) DEFAULT 0.00,

```

```
`currency` varchar(3) DEFAULT 'SYR',
`isActive` tinyint(1) DEFAULT 1,
`created_at` timestamp NOT NULL DEFAULT current_timestamp(),
`updated_at` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE
current_timestamp()
)
```

المراجع

- ١) الموسوعة الحرة Wikipedia (٢٠٢٢)، "الإدارة الإلكترونية" متاح على:
<https://en.wikipedia.org>
[تاريخ الاطلاع ٢٠٢٥/٥/١٢]
- ٢) أبو شنب، ع. أ., حرب، ي., وأبو البصل، و. (٢٠١١) "الخدمات الإلكترونية". إربد، الأردن: دار الكتاب الثقافي. متاح على: <https://books.google.com/books/about> [تاريخ الاطلاع: ٩ مايو ٢٠٢٥].
- 3) Auckland Airport Parking (2024) Auckland Airport Limited [Online]. Available at: <https://www.aucklandairport.co.nz/airport-parking> (Accessed: 3 May 2025).
- 4) ParkWhiz (2024) ParkWhiz, Inc., Chicago, IL [Online]. Available at: <https://www.parkwhiz.com/> (Accessed: 4 May 2025).
- 5) SpotHero (2024) SpotHero, Inc., Chicago, IL [Online]. Available at: <https://www.spothero.com/> (Accessed: 12 May 2025).
- 6) JustPark (2024) JustPark Limited, London [Online]. Available at: <https://www.justpark.com/> (Accessed: 5 May 2025).
- 7) ParkMe (2024) ParkMe, Inc., Santa Monica, CA [Online]. Available at: <https://www.parkme.com/accounts/login/?next=/> (Accessed: 8 May 2025).
- 8) Parkopedia (2025) Parking solutions. London: Parkopedia Limited. Available at: <https://www.parkopedia.com/> (Accessed: 22 May 2025)
- 9) Karunamurthy, A., Chander, M., & Vigneshwaran, R. (2024). Enhancing Urban Parking Management Through an Online Reservation System: A Step Towards Smarter Cities. Quing: International Journal of Innovative Research in Science and Engineering, 3(4), 27-33. <https://doi.org/10.54368/qijirse.3.4.0135> (Accessed: 15 May 2025)
- 10) Ogbiti, John Temitope, and William Aaron. "Development of a web-based car rental management system." Science World Journal 19.3 (2024): 797-807. (Accessed: 5 June 2025)
- ١١) الشرق الأوسط للسفر والسياحة (٢٠٢٥). "دبي تتبّنى مواقف السيارات الذكية من خلال إطلاق تطبيق باركن عالي التقنية". (عبر الانترنت) متاح على:
<https://www.travelandtourworld>
[تاريخ الاطلاع: ٤ مايو ٢٠٢٥]

- 12) QRCode (2024) QR code generator [npm package]. Available at: <https://www.npmjs.com/package/qrcode> (Accessed: 8 June 2025).
- 13) Nodemailer (2024) Nodemailer [Online]. Available at: <https://nodemailer.com/> (Accessed: 2 June 2025)
- (١٤) API Club (٢٠٢٥) المنصة المعرفية للمطوريين العرب: المنصة العربية الأولى التي توفر توثيقاً شاملاً للتقنيات البرمجية وخدمات API متکاملة. متاح على: <https://apiclub.site> [تاريخ الاطلاع: ١٨ مايو ٢٠٢٥]
- (١٥) الشهابي، طلال، ٢٠٢٣، مقرر مبادئ هندسة البرمجيات، الجامعة الافتراضية السورية
- (١٦) الخطيب، باسل، ٢٠٢٢، مقرر برمجة الويب ١، الجامعة الافتراضية السورية
- (١٧) الخطيب، باسل، ٢٠٢٤، مقرر برمجة الويب ٢، الجامعة الافتراضية السورية
- (١٨) الخوري، عبدو، ٢٠٢٢، مقرر مبادئ قواعد البيانات، الجامعة الافتراضية السورية