



الجمهورية الجزائرية الديمقراطية الشعبية  
وزارة التعليم العالي والبحث العلمي  
جامعة الشهيد حمدة لخضر الوادي



كلية العلوم الدقيقة

قسم الإعلام الآلي

مذكرة نهاية السنة

ليسانس أكاديمي

الميدان: رياضيات وإعلام آلي

الموضوع:

تطوير موقع تسيير ومراقبة حظيرة مركبات

مذكرة تخرج تدخل ضمن متطلبات الحصول على شهادة ليسانس في شعبة الإعلام الآلي. تخصص: أنظمة معلوماتية.

تحت إشراف:

- بن علي عبد الكامل

من انجاز الطلبة:

- أيمن كروش

- هزله هارون

نوقشت يوم 08 / 06 / 2024 أمام اللجنة المكونة من الأساتذة:

رئيسا

يعقوب محمد الأمين

مقررا

بن علي عبد الكامل

عضوا

خالد سلطاني

السنة الجامعية: 2023 / 2024

# الإهداء

الحمد لله وكفى والصلاة على الحبيب المصطفى وأهله ومن وفى أما بعد

الحمد لله الذي وفقنا لتتئين هذه الخطوة في مسيرتنا الدراسية بمذكرتنا

هذه ثمرة الجهد والنجاح بفضلته تعالى

مهداة إلى والدينا الكرام حفظهم الله وأدامهم نورا لدرينا

لكل عائلاتنا الكريمة التي ساندتنا ولا تزال، من إخوة وأخوات

إلى رفقاء المشوار الذين قاسمونا كطاتهم رعاهم الله ووفقهم،

إلى قسم الإعلام الآلي وجميع دفعة السنة الثالثة LMD 2024

جامعة الشهيد حمه نخضر الوادي

إلى كل من كان لهم أثر على حياتنا، وإلى كل

من أحبهم قلبنا ونسيم أقلامنا.

# شكر وتقدير

نشكر الله عز وجل الذي بتوفيقه وفضل منه تمكنا من إنجاز هذه المذكرة  
تتقدم بالعرفان والشكر الجزيل إلى الأستاذ الفاضل بن علي عبد الكامل  
على كل التوجيهات والملاحظات التي وجهها لنا،  
وكذلك على صبره طيلة إشرافه على هذه المذكرة .  
كما نشكر كثيرا جميع الأساتذة الذين قدموا لنا المساعدة مهما كانت طبيعتها،  
وإلى كل من قدم لنا تشجيعا مهما بلغت درجته.  
كما تتوجه بخالص الشكر إلى كافة أساتذتنا الكرام بقسم الإعلام الآلي  
بجامعة الشهيد حمه لخضر الوادي على كل ما قدموه لنا طيلة فترة تكويننا.

## الملخص:

الهدف من مشروعنا الوارد في هذا التقرير هو تصميم وتنفيذ بسيط ومحسن لتسيير ومراقبة حضيرة السيارات وقد أجريت عملية الإنجاز من خلال ثلاثة مراحل وهي:

أولا قمنا بدراسة وجمع المعلومات والإحصائيات المختلفة لخدمة التسيير والمراقبة لحظيرة المركبات، ثم المرحلة الثانية المتمثلة في النمذجة والتي قمنا فيها بوصف أجزاء ومكونات الموقع باللغة الموحدة للنمذجة UML، أما المرحلة الأخيرة اعتمدنا في تطوير هذا الموقع على العديد من التقنيات:

- قواعد بيانات MySQL لتخزين البيانات
- لغة ال PHP للتعامل مع قواعد بيانات / BOOTSTAP لإنشاء واجهة تفاعلية حديثة.
- الكلمات المفتاحية: حظيرة مركبات، صيانة، سائق...

## Summary:

The objective of our project, as outlined in this report, is to design and implement a simple and optimized system for managing and monitoring a vehicle fleet. The project was carried out in three phases:

First, we conducted a study and gathered various data and statistics relevant to the management and monitoring of vehicle fleets. The second phase involved modeling, during which we described the components and parts of the site using the Unified Modeling Language (UML). In the final phase, we developed the site using several technologies:

- MYSQL databases for data storage
- PHP language for database interaction
- BOOTSTRAP for creating a modern, interactive interface.

Keywords: vehicle fleet, maintenance, driver...

الصفحة	الموضوع
1	المقدمة العامة
	<b>الفصل الأول : دراسة النظرية</b>
4-3	1. المشاكل المطروحة.....
3	مراقبة سيارات الشركة.....
3	صيانة مركبات الشركة بشكل دوري.....
4	جمع وتحليل البيانات.....
5-4	2. فوائد المشروع.....
4	تقليل وقت التوقف عن العمل.....
4	مراقبة سلوكيات القيادة.....
5	زيادة الإنتاجية.....
5	3. الإحصائيات.....
5	التحديات في إدارة الحظيرة.....
5	زيادة تكلفة الصيانة.....
5	تأثير الصيانة غير المخططة.....
6	تأثير التوقف عن العمل.....
6	التحديات في التواصل والتنسيق.....
	<b>الفصل الثاني : التصميم</b>
8	المقدمة.....
8	1. لغة النمذجة الموحدة.....
9	2. أهمية استخدام لغة النمذجة الموحدة UML.....
10-9	3. مخططات UML.....
10	4. أداة UML المستعملة.....
14-10	5. مخطط حالة الاستخدام (Diagram Case Ues).....
11	المخطط العام لحالة الاستخدام.....
12	مخطط حالة الاستخدام "مسير حظيرة".....
13	مخطط حالة الاستخدام "السائق".....
13	مخطط حالة الاستخدام "منسق المهام".....
14	مخطط حالة الاستخدام "فريق الصيانة".....
14	6. مخطط النشاط (Diagram activity).....
15	مخطط النشاط "تسجيل الدخول".....
16	مخطط النشاط "المهمة".....
17	مخطط النشاط "التزود بالوقود".....
18	مخطط النشاط "الصيانة".....
19	7. مخطط الفئات (Diagramme de Classes) :.....
19	المخطط العام للفئات.....
21-20	النموذج العلائقي و ترجمة المعطيات و قاموس المعطيات.....

22	..... قاموس المعطبات
22	..... جدول قاموس المعطبات
23	..... الخاتمة
<b>الفصل الثالث : الانجاز و التنفيذ</b>	
24	..... المقدمة
25-24	..... 1. لغات التنفيذ المستخدمة
24	..... MySQL
25	..... PHP
25	..... JavaScript
29-26	..... 2. أدوات التطوير
26	..... BOOSTRAP
27	..... XHTML
28-27	..... CSS
29-28	..... Laravel
35-30	..... 3. واجهات الموقع
30	..... واجهة تسجيل الدخول
30	..... واجهة السائقين
31	..... واجهة إضافة سائق
31	..... واجهة السيارات
32	..... واجهة إضافة سيارة
32	..... واجهة التزويد بالوقود
33	..... واجهة عملية التزود بالوقود
33	..... واجهة المهام
34	..... واجهة إضافة المهمة
34	..... واجهة الصيانة
35	..... واجهة إضافة الصيانة
35	..... واجهة لمستخدمين
	..... الخاتمة العامة

فهرس الأشكال

الصفحة	الشكل
10	الشكل 1 : تقسيم مخططات لغة النمذجة الموحدة
11	الشكل 2: مخطط العام لحالة الاستخدام
12	الشكل 3 : مخطط حالة الاستخدام مسير حظيرة
13	الشكل 4 : مخطط حالة الاستخدام السائق
13	الشكل 5 : مخطط حالة الاستخدام منسق المهام
14	الشكل 6 : مخطط حالة الاستخدام فريق الصيانة
15	الشكل 7 : مخطط حالة النشاط تسجيل الدخول
16	الشكل 8 : مخطط حالة النشاط المهمة
17	الشكل 9 : مخطط حالة النشاط التزود بالوقود
18	الشكل 10 : مخطط حالة النشاط الصيانة
19	الشكل 11 : مخطط العام للفئات
30	الشكل 12 : السائقين
30	الشكل 13: واجهة سائق
31	الشكل 14: واجهة السيارات
31	الشكل 15: واجهة اضافة سيارة
32	الشكل 16: التزود بالوقود
32	الشكل 17: واجهة عملية التزود
33	الشكل 18: واجهة المهام
33	الشكل 19: واجهة اضافة مهمة
34	الشكل 20: واجهة الصيانة
34	الشكل 22: واجهة اضافة الصيانة
35	الشكل 23: واجهة المستخدمين

# الفصل الأول: دراسة النظرية

## 1. المشاكل المطروحة:

### 1-1 مراقبة سيارات الشركة:

- نقص المعلومات حول حالة المركبات بدون بيانات حول مستوى الوقود أو حالة المحرك والأجزاء الأخرى من السيارة، يمكن أن يكون من الصعب اتخاذ القرارات الفعّالة بشأن الصيانة وإدارة استهلاك الوقود.
- صعوبة مراقبة سلوكيات القيادة تتسبب هذه المشكلة في عدم القدرة على تقييم سلوك السائقين، مما يزيد من مخاطر الحوادث ويؤثر سلباً على سمعة الشركة. [1]

### 2-1 صيانة مركبات الشركة بشكل دوري:

- صعوبة في جدولة مواعيد الصيانة الدورية وتذكرها: يمكن أن تؤدي هذه المشكلة إلى تأخير في الصيانة اللازمة، مما يزيد من خطر حدوث الأعطال ويؤثر سلباً على عمر السيارة.
- صعوبة في إدارة قطع الغيار والمخزون بشكل فعال: قد يتسبب ذلك في تأخيرات في عمليات الصيانة وزيادة التكاليف بسبب عدم توفر القطع البديلة في الوقت المناسب.
- تأخيرات في إصلاح الأعطال وزيادة تكاليف الصيانة: عدم القدرة على التعرف على الأعطال في الوقت المناسب وتأخير توفير قطع الغيار اللازمة يمكن أن يؤدي إلى زيادة التكاليف وتعطيل عمليات الشركة. [1]

### 3-1 جمع وتحليل البيانات:

- صعوبة في جمع البيانات حول استخدام المركبات وكفاءة استهلاك الوقود: من دون بيانات كافية حول استخدام السيارات وكيفية استهلاك الوقود، يصعب تحديد الفرص لتحسين الكفاءة وتقليل التكاليف.
- صعوبة في تحليل البيانات وتحديد فرص تحسين العمليات: بدون أدوات تحليل البيانات الصحيحة أو الخبراء المناسبين، يمكن أن تظل البيانات غير مستغلة بشكل كامل، مما يمنع الشركة من تحسين عملياتها بشكل فعال.
- معالجة هذه المشكلات يتطلب تنفيذ نظام شامل وفعال لإدارة حظيرة الشركة وتحليل بياناته بشكل فعال لتحقيق أهداف الكفاءة والتوفير في التكاليف. [1]

## 2. فوائد المشروع:

### 3-1 تقليل وقت التوقف عن العمل:

- تحسين استهلاك الوقود: من خلال مراقبة وتحليل أنماط القيادة وتوجيه السائقين نحو القيادة الاقتصادية، يمكن تقليل استهلاك الوقود وبالتالي تقليل مواقف التزود بالوقود وتوقفات غير مخطط لها.
- تقليل تكاليف الصيانة: من خلال الصيانة الدورية والتحقق المنتظم من حالة المركبات، يمكن اكتشاف المشاكل المحتملة مبكرًا وتصحيحها قبل أن تؤدي إلى توقفات طويلة وتكاليف إصلاح باهظة.

### 3-2 مراقبة سلوكيات القيادة:

- تقليل الحوادث: من خلال مراقبة سلوك القيادة وتحليله، يمكن تحديد العادات السيئة مثل القيادة بسرعة مفرطة أو التشتت أثناء القيادة، وبالتالي تقليل حوادث الطرق والتكاليف المرتبطة بها.

### 3-3 زيادة الإنتاجية:

- تحسين تخطيط العمليات: من خلال معرفة مواقع وحالات المركبات في الوقت الفعلي، يمكن تحسين تخطيط العمليات وتوجيه المركبات بشكل أكثر فعالية لتنفيذ المهام وتقليل تأخيرات التسليم.
- زيادة الكفاءة والإنتاجية: بفضل تحسين استخدام الوقت وتقليل التوقفات غير المخطط لها وتحسين كفاءة القيادة، يمكن زيادة الإنتاجية العامة للشركة وتحسين أداء العمل.

هذه الفوائد تساهم في تحسين أداء الشركة، وتقليل التكاليف، وزيادة الكفاءة والإنتاجية بشكل عام. [1]

## 3. الإحصائيات:

### 1-2 التحديات في إدارة الحظيرة:

يشير 60% من الشركات إلى وجود مشاكل في إدارة وصيانة الحظيرة المركبات. هذا يعكس تحديات متعددة تواجهها الشركات في مجال إدارة الحظيرة بشكل فعال.

### 2-2 زيادة تكلفة الصيانة:

يُظهر التقرير الإحصائي أن متوسط تكلفة إدارة حظيرة المركبات يمكن أن ترتفع بنسبة تتراوح بين 20% إلى 30% بسبب التحديات في الصيانة والتواصل. هذا يعني أن هناك حاجة ملحة لحلول تساعد في تقليل هذه التكاليف الزائدة.

### 3-2 تأثير الصيانة غير المخططة:

وفقًا لدراسة أخرى تشير إحصائيات إلى أن نسبة كبيرة من التكاليف المتعلقة بصيانة حظيرة المركبات تأتي من الصيانة غير المخططة أو الإصلاحات الطارئة. يمكن أن تصل هذه التكاليف إلى نسبة معينة من ميزانية الشركة، مما يؤثر على الأداء المالي بشكل عام.

### 4-2 تأثير التوقف عن العمل:

تظهر بيانات أخرى أن كل ساعة من التوقف غير المخطط له لأحد المركبات في الحظيرة يمكن أن تكلف الشركة مبالغ كبيرة من المال، سواء بسبب فقدان الإيرادات أو تكاليف الإصلاح الطارئة.

### 5-2 التحديات في التواصل والتنسيق:

إحدى التحديات الرئيسية التي تواجهها الشركات هي ضعف التواصل والتنسيق بين مختلف الأقسام المعنية بإدارة الحظيرة، مما يؤدي إلى تأخير في الصيانة وتكلفة إضافية ناتجة عن تشتت الجهود وتكرار الأعمال.

هذه الإحصائيات تسلط الضوء على عدة جوانب تتعلق بإدارة وصيانة حظيرة المركبات، وتظهر أهمية تطوير حلول تقنية تساعد في تحسين الكفاءة وتقليل التكاليف للشركات. [2]



## الفصل الثاني : التصميم

## المقدمة:

في الجزء السابق تطرقنا إلى دراسة الموجود في النظام الحالي والتي سمحت لنا بتحليل النظام ومعرفة أكبر قدر من المعلومات حوله، حيث استخلصنا منها جملة من المعارف التي تسمح بإقامة تصور عن النظام المستقبلي، والذي يستجيب للاحتياجات المستخدمين. في هذه المرحلة سنستخدم لغة UML النمذجة الموحدة والتي من خلالها سنتعرف على أهم جزئيات التصميم.

سنستخدم أداة StarUML للتعبير عن التطبيق الذي سيتم إنجازه وتصميم نظام معلوماتي وفق الخطوات التالية:

- إنشاء مخطط حالة الاستخدام (Diagram Case Use)
- إنشاء مخطط الفئات (Diagram Class)
- إنشاء مخطط النشاط (diagram Activity)

## 1. لغة النمذجة الموحدة:

- UML- Language Modelling Unified هي لغة النمذجة الموحدة وهي لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية، وتعتمد الUML على البرمجة كائنية التوجه -Object Oriented Programming-OOP-
- تستخدم لعمل نموذج مبدئي لبرنامج سهل الفهم، وتحديد هيكل وسلوك النظام بصيغة مجردة عن أي تفاصيل تقنية بحيث تصبح مقروءة من قبل أي شخص بغض النظر عن أي تقنية برمجية يألفها.

## 2. أهمية استخدام لغة النمذجة الموحدة UML:

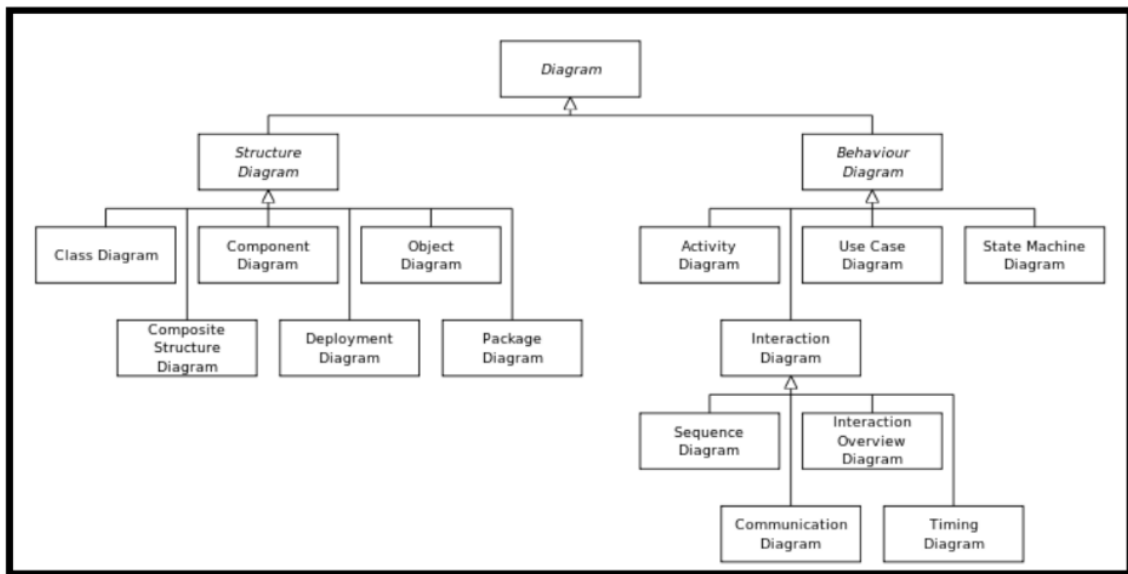
- تصميم البرمجيات بشكل احترافي
- سهولة التعديل والصيانة وبكلفة منخفضة.
- مخططات UML تساعد المطورين على فهم النظام بسهولة وسرعة.
- لغة تواصل بين المطورين والمصممين.
- توثيق التصميم قبل البدء بالبرمجة.
- الاستقلالية عن أي لغة برمجة معينة وعمليات التطوير.

### 3. مخططات UML:

يتكون UML من أربعة عشر مخططا تستخدم دلالات دقيقة، واستخدامها الباحثون لتصميم نماذج يمكن الاعتماد عليها للحد من الأخطاء البرمجية الكبيرة، تمكننا لغة البرمجة الموحدة من تحويل المشروع في الواقع إلى تصميم أقرب للفهم وترجمته بإحدى لغات البرمجة (C ++,C; Java...). المخطط يعطي للمستخدم وسيلة للعرض والتعامل مع عناصر النمذجة.

وتنقسم هذه المخططات إلى مجموعتين:

مخططات البنية ( Diagrammes Structures ) والمخططات السلوكية ( Diagrammes Behaviors ) كما هو موضح في (الشكل 1 )



الشكل ( 1 ) : تقسيم مخططات لغة النمذجة الموحدة

## 4. أداة UML المستعملة:

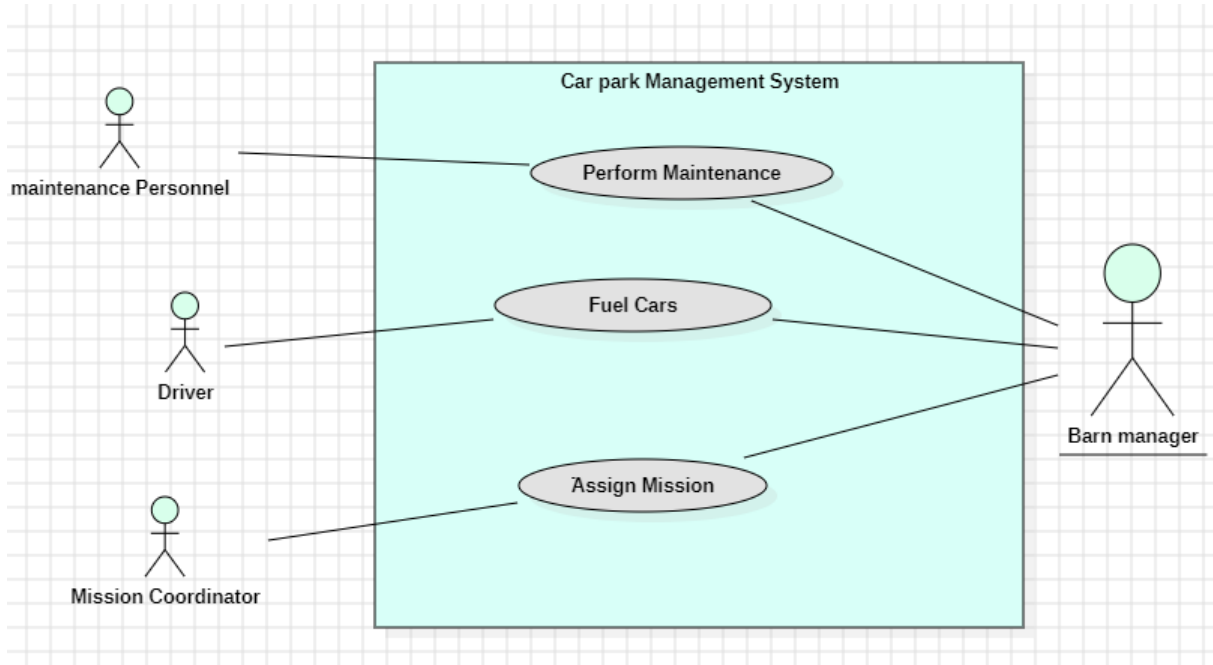
برنامج رسم المخططات StarUML هو برنامج يساعد على رسم المخططات المستخدمة في تصميم البرامج ويقوم برسم الجداول والأشكال بكل سهولة. هذه الأداة تتيح حفظ العمل بصيغ متعددة (صور، ملفات نصية... الخ). لقد استخدمنا في مشروعنا هذا ثلاثة مخططات وهي كالتالي:

## 5. مخطط حالة الاستخدام (Diagram Case Ues):

هو عبارة عن مخطط يظهر التفاعل بين المستخدمين (Actors) وحالات الاستخدام (Case Use) للنظام المراد إنجازه.

## 5-1 المخطط العام لحالة الاستخدام:

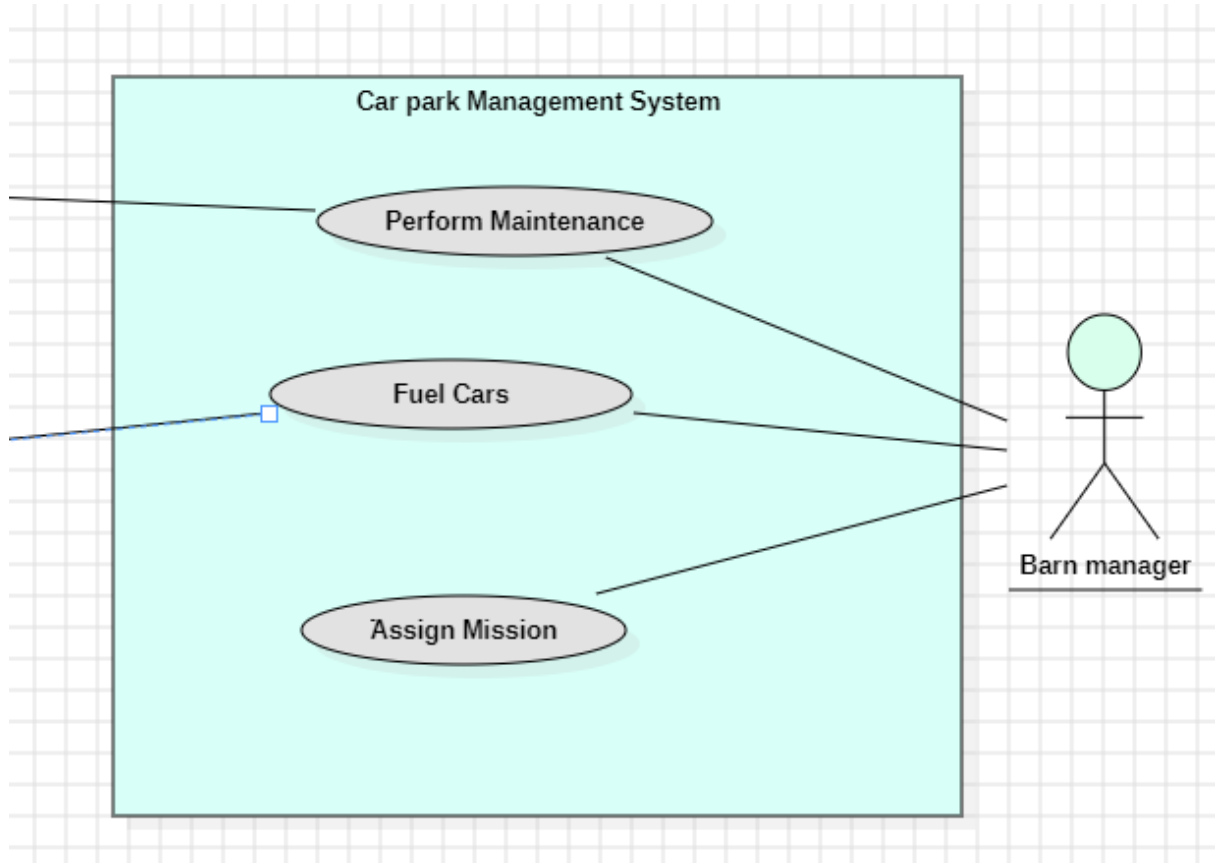
رسم تخطيطي لحالة الاستخدام يوضح التفاعلات بين الجهات الفاعلة (المستخدمين) والنظام من حيث وظائف النظام. في هذا السيناريو، لدينا عدة جهات فاعلة: "السائق"، "منسق المهام" و"فريق الصيانة" و"مسير حظيرة".



الشكل (2): مخطط العام لحالة الاستخدام

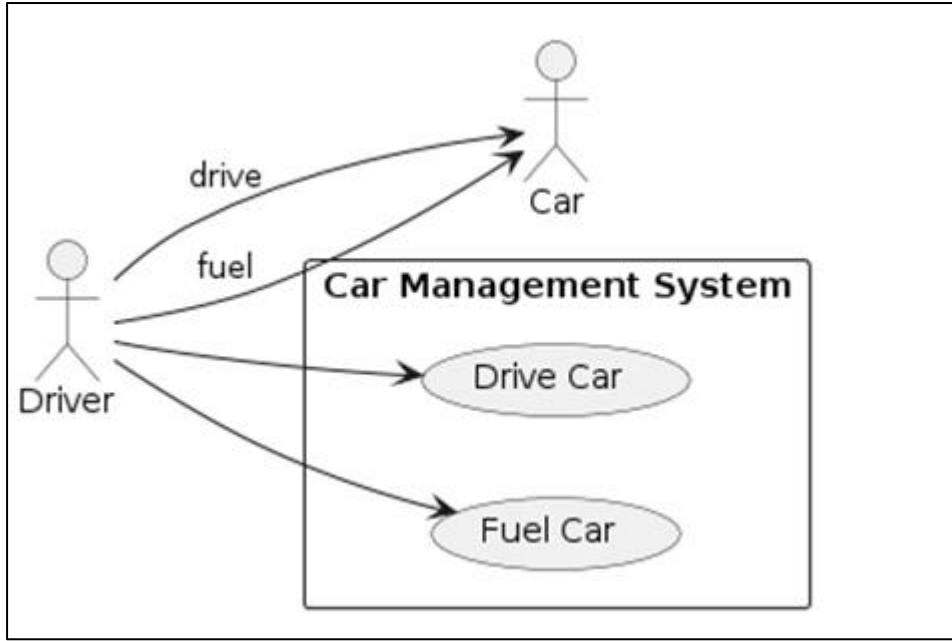
## 5-2 مخطط حالة الاستخدام "مسير حظيرة":

يركز هذا الرسم التخطيطي بشكل خاص على الإجراءات التي يمكن لـ "السائق" القيام بها في النظام، وهي تعين مهمة و تسجيل صيانة و تزود بالوقود و تسجيل سائقين و سيارات



الشكل ( 3 ) : مخطط حالة الاستخدام " مسير بالحظيرة "

## 5-2 مخطط حالة الاستخدام "السائق":

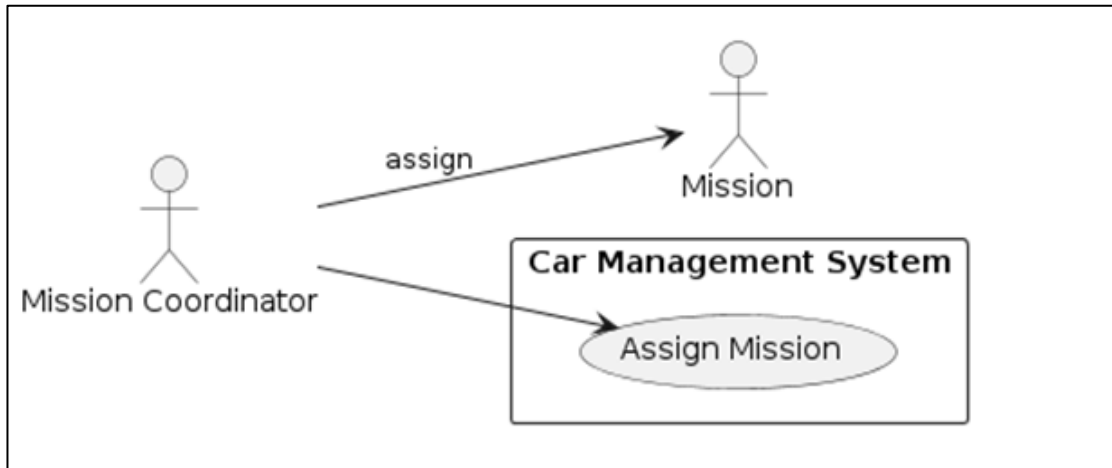


الشكل (3) : مخطط حالة الاستخدام " السائق "

يركز هذا الرسم التخطيطي بشكل خاص على الإجراءات التي يمكن لـ "السائق" القيام بها في النظام، وهي قيادة السيارة وتزويد السيارة بالوقود.

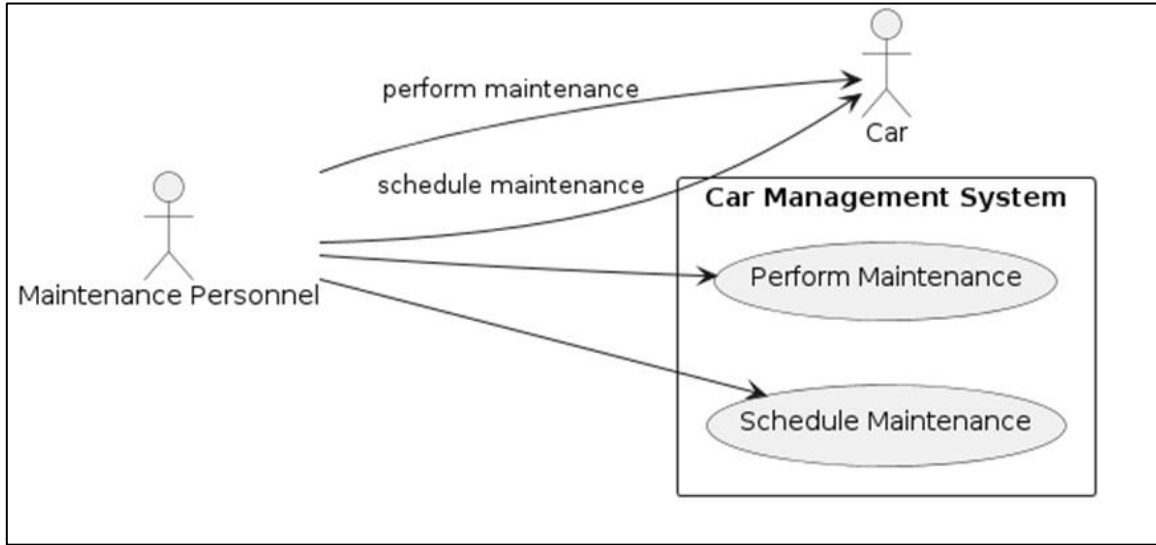
### 3-5 مخطط حالة الاستخدام "منسق المهام":

يوضح هذا الرسم التخطيطي إجراء تعيين مهمة للسائق، الذي يقوم به الفاعل "منسق المهام".



الشكل (4) : مخطط حالة الاستخدام " منسق المهام "

## 4-5 مخطط حالة الاستخدام "فريق الصيانة":



الشكل (5) : مخطط حالة الاستخدام "فريق الصيانة"

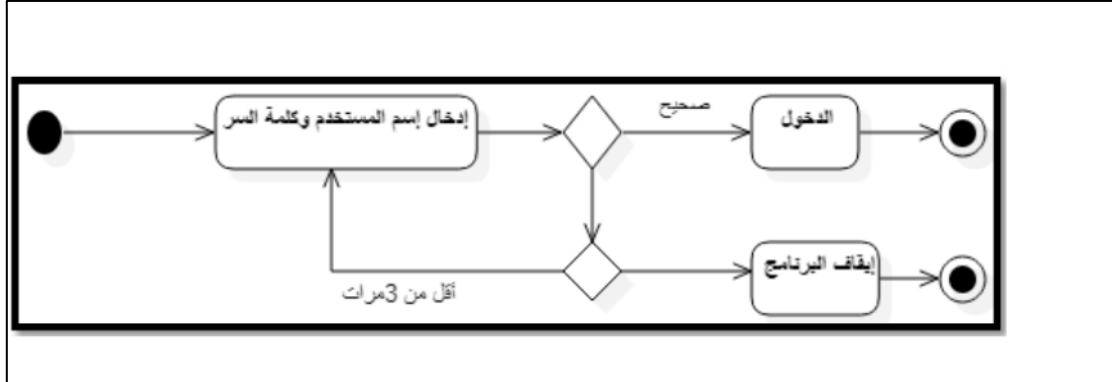
يركز هذا الرسم التخطيطي على الإجراءات المتعلقة بمهام الصيانة، مثل تنفيذ الصيانة وجدولة الصيانة، التي يقوم بها الفاعل "فريق الصيانة".

## 6. مخطط حالة النشاط (Diagram activity):

مخططات النشاط عبارة عن تمثيلات بيانية لسير عمل الأنشطة والإجراءات التدريجية مع دعم الاختيار والتكرار والتزامن في لغة النمذجة الموحدة، تهدف الرسوم البيانية للنشاط إلى نمذجة كل من العمليات الحسابية والتنظيمية.

### 6-1 مخطط حالة النشاط " تسجيل الدخول ":

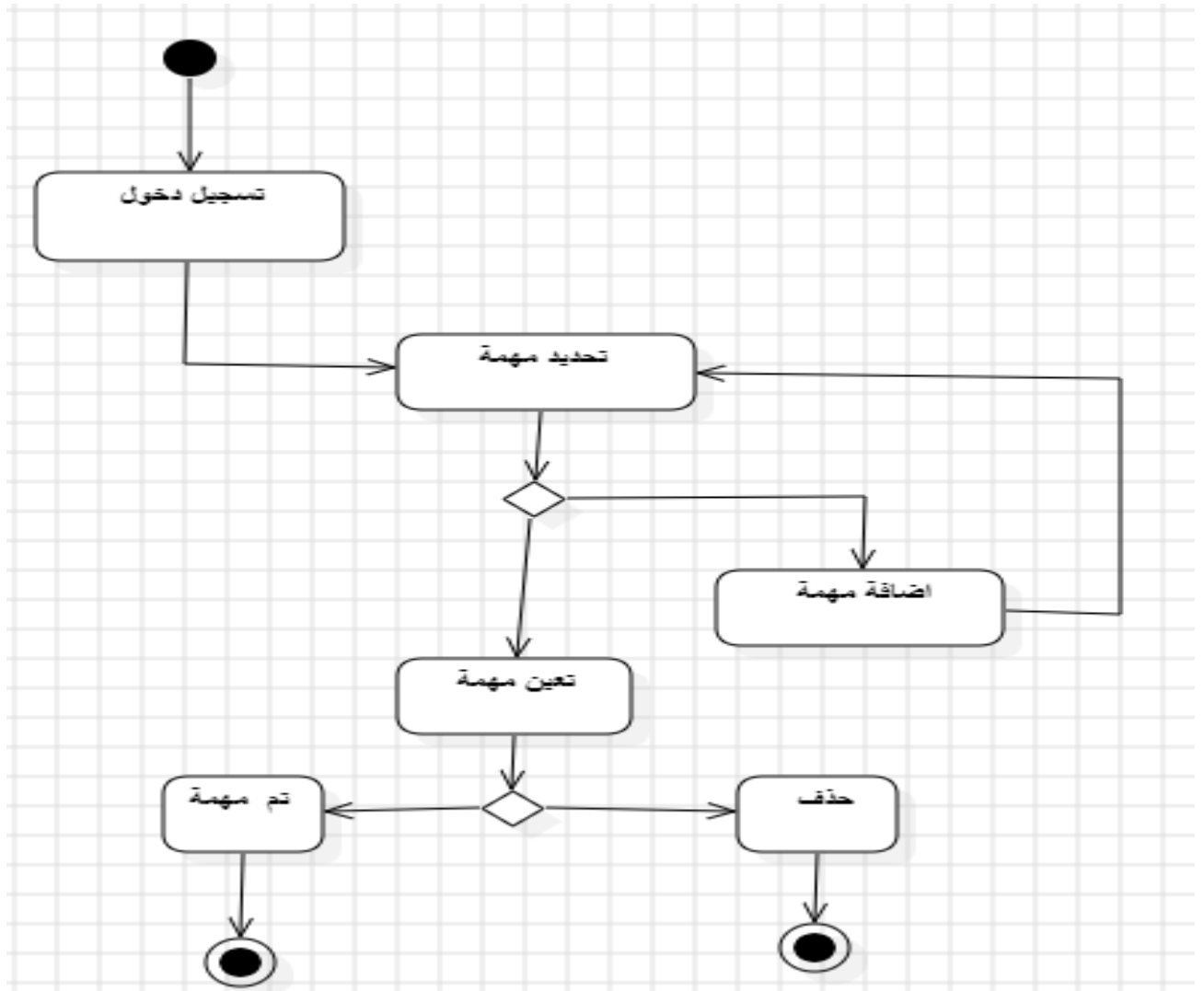
يقوم المستخدمميين إدخال اسم المستخدم وكلمة السر إذا كان صحيح يدخل في البرنامج أو يكرر 3 مرات.



الشكل ( 6 ) : مخطط النشاط تسجيل الدخول

2-6 مخطط حالة النشاط " المهمة ":

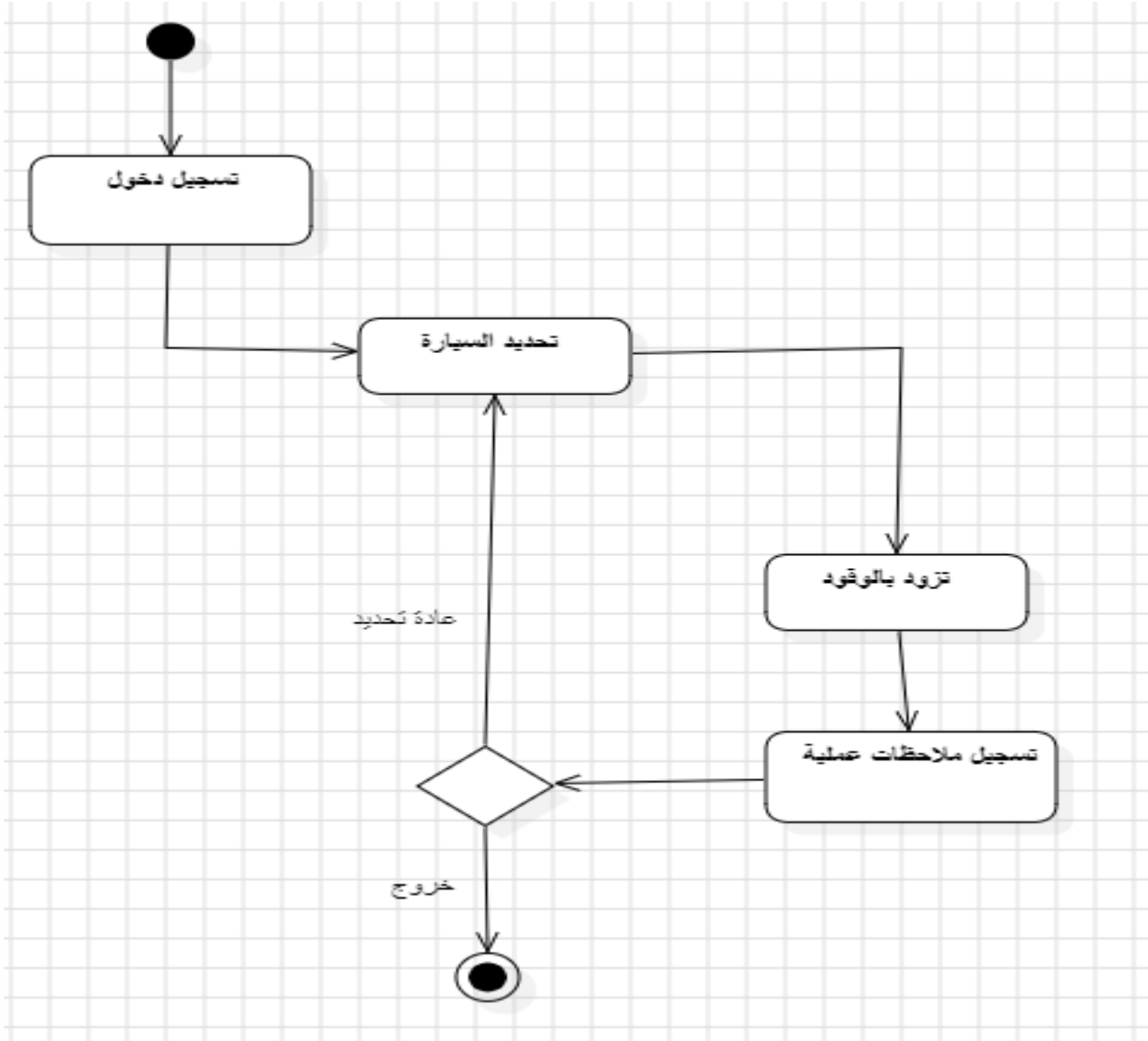
تعين المهمة من طرف المنسق المهام أو حيث يقوم تحديد المهمة (معرف سائق، معرف مركبة، مدة، الوجهة)



الشكل ( 10 ): مخطط النشاط المهمة

## 3-6 مخطط حالة النشاط " التزود بالوقود ":

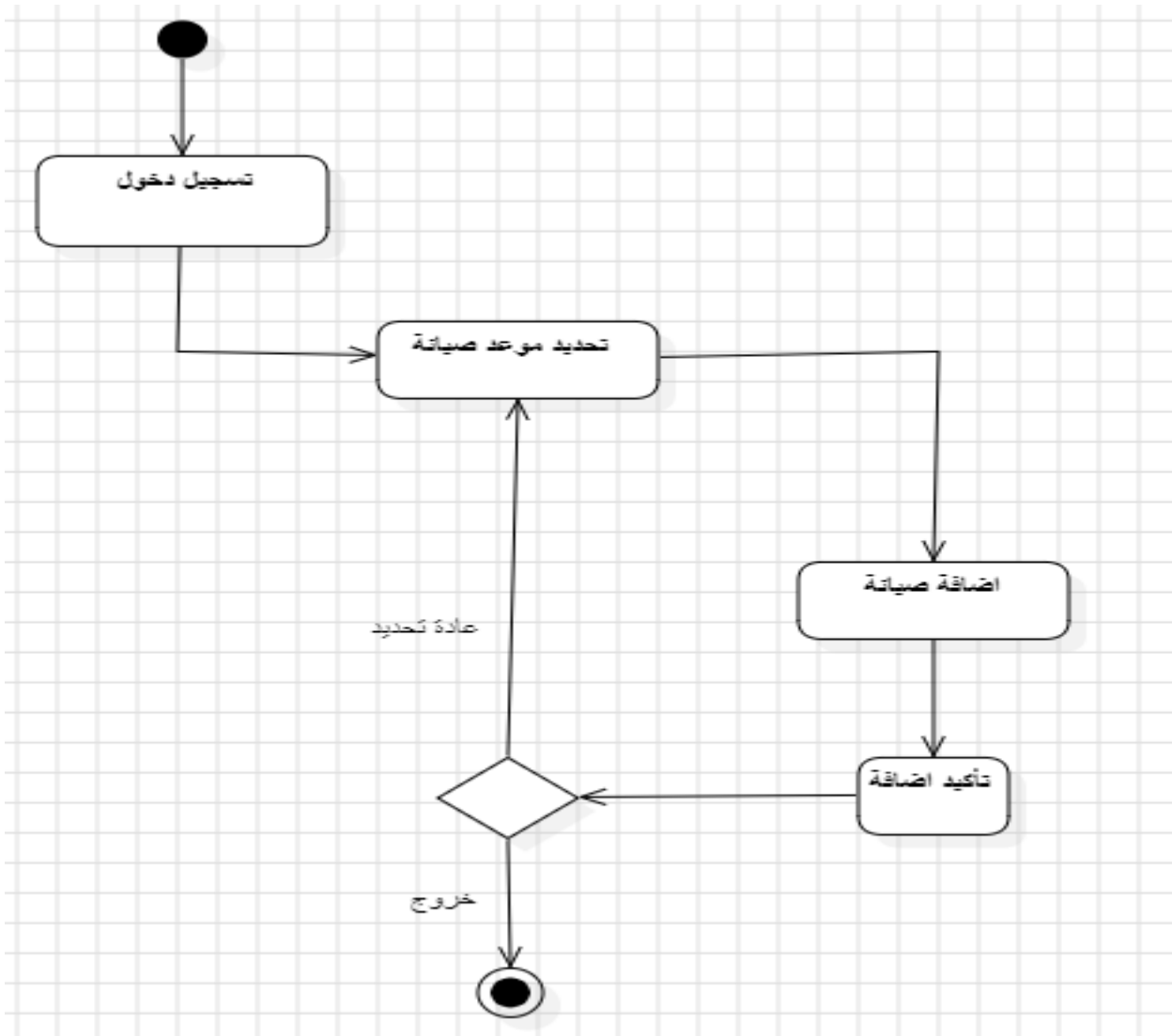
تنفذ عملية التزود بالوقود من طرف فريق صيانة أو سائق من خلال تحديد (معرف السيارة، تاريخ التزويد بالوقود ،موقع التزويد بالوقود ،كمية الوقود) ثم تأكيد معلومات إن وجد خطأ تعاد عملية إدخال المعلومات



الشكل ( 11 ) : مخطط النشاط التزود بالوقود

4-6 مخطط حالة النشاط " الصيانة " :

تنفذ عملية الصيانة من فريق صيانة من خلال تحديد (معرف الصيانة، معرف السيارة، تاريخ الصيانة، نوع الصيانة)

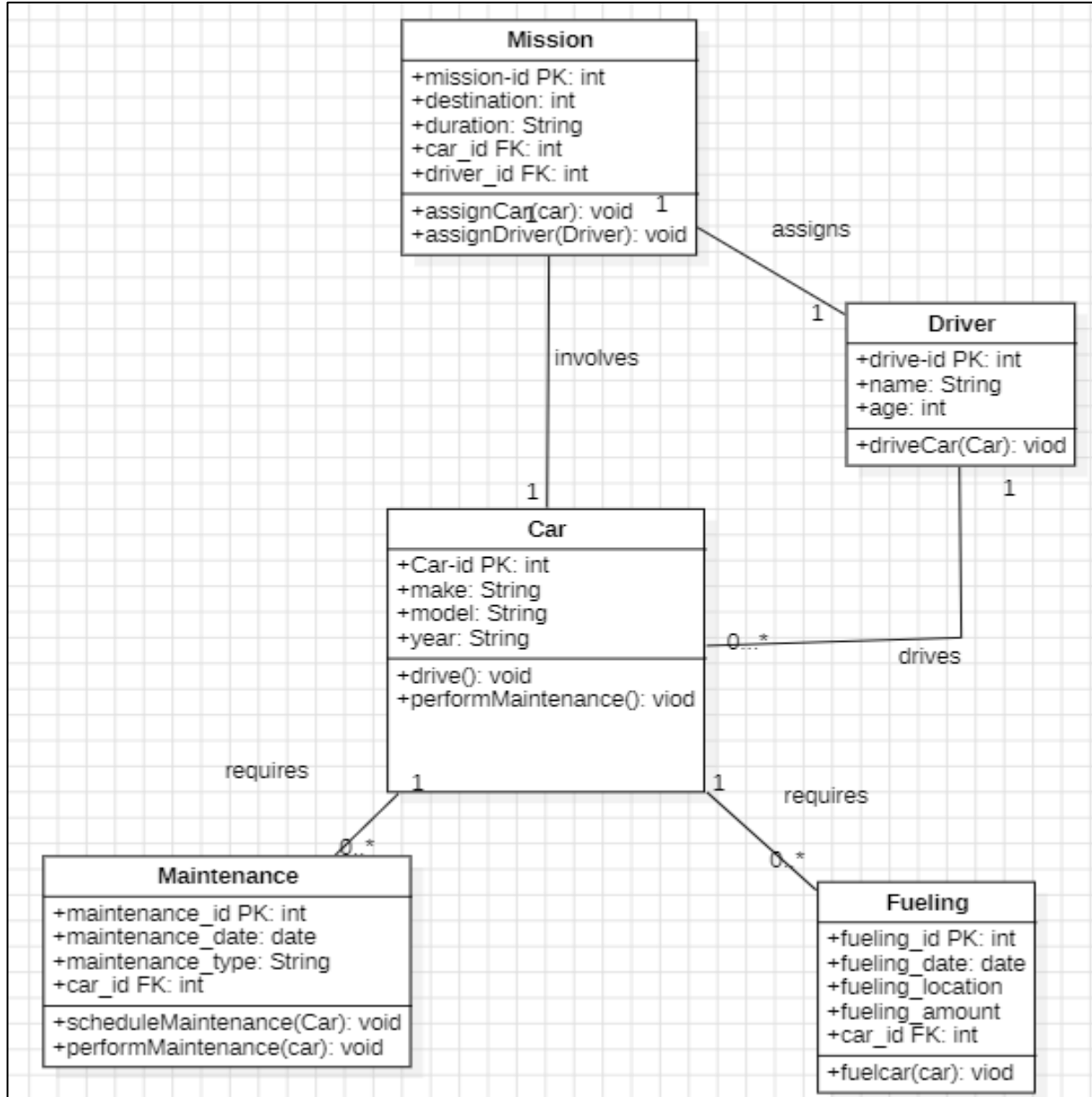


الشكل ( 12 ) : مخطط النشاط الصيانة

7. مخطط الفئات (Diagramme de Classes) :

هو من أحد أنواع مخططات الفئة في لغة النمذجة الموحدة (UML) سهل الرسم نسبيا ، و هو نوع من المخططات البنوية (Structurals Diagrammes) التي تصف بنية النظام من خلال إظهار طبقاته وخصائصه، والعمليات والعلاقات بين الكائنات.

7-1 المخطط العام للفئات:



الشكل (13) : مخطط العام للفئات

## 2-7 النموذج العلائقي و ترجمة المعطيات و قاموس المعطيات :

## • النموذج العلائقي:

- Car (car\_id PK, make, model, year)
- Driver (driver\_id PK, name, age)
- Mission (mission\_id PK, destination, duration, car\_id FK, driver\_id FK)
- Maintenance (maintenance\_id PK, maintenance\_date, maintenance\_type, car\_id FK)
- Fueling (fueling\_id PK, fueling\_date, fueling\_location, fueling\_amount, car\_id FK)

## ▪ In this schema:

- PK stands for Primary Key.
- FK stands for Foreign Key.
- car\_id and driver\_id in the Mission table are foreign keys referencing the Car and Driver tables, respectively.
- car\_id in the Maintenance and Fueling tables are foreign keys referencing the Car table.

## • ترجمة المعطيات :

## ▪ السيارة (Car)

- car\_id (المعرف الرئيسي PK)
- make (الماركة)
- model (الطراز)
- year (السنة)
- 

## ▪ السائق (Driver)

- driver\_id (المعرف الرئيسي PK)
- name (الاسم)
- age (العمر)
- 

## ▪ المهمة (Mission)

- mission\_id (المعرف الرئيسي PK)
- destination (الوجهة)
- duration (المدة)
- car\_id (المعرف الخارجي FK)
- driver\_id (المعرف الخارجي FK)

## ▪ الصيانة (Maintenance)

- maintenance\_id (المعرف الرئيسي PK)
- maintenance\_date (تاريخ الصيانة)

- maintenance\_type (نوع الصيانة)
- car\_id (المعرف الخارجي FK)

▪ التزود بالوقود (Fueling)

- fueling\_id (المعرف الرئيسي PK)
- fueling\_date (تاريخ التزويد بالوقود)
- fueling\_location (موقع التزويد بالوقود)
- fueling\_amount (كمية الوقود)
- car\_id (المعرف الخارجي FK)

• قاموس المعطيات :

rgLes classes	L'attribut	Signification	type
Car	car_id PK	Number of Car	Integer
	make	Make of Car	String
	model	Model of Car	String
	year	Year of Car	Integer
Driver	driver_id PK	Number of Driver	Integer
	name	Name of Driver	String
	age	Age of Driver	Integer
Mission	mission_id PK	Number of Mission	Integer
	destination	Destination of Mission	String
	duration	Duration of Mission	date
Maintenance	maintenance_id PK	Number of Maintenance	Integer
	maintenance_date	Date of Maintenance	date

	maintenance_type	Type of Maintenance	String
Fueling	fueling_id PK	Number of Fueling	Integer
	fueling_date	Date of Fueling	date
	fueling_location	Location of Fueling	String
	fueling_amount	Fueling of amount	Integer

الجدول ( 1 ): جدول قاموس المعطيات

**الخاتمة:**

حاولنا في هذا الفصل تقديم موجز كاف لشرح طريقة عمل النظام وهذا بعد إنجاز المخططات اللازمة (مخطط حالات الاستخدام، مخطط النشاط) والنموذج التصوري للمعطيات (مخطط الفئات)، والتي من خلالها أصبح لدينا تصور كاف عن الموقع الجاري إنجازه. بوصولنا لهذه المرحلة لم يبق لنا سوى الشرط الأخير ألا وهو الدراسة التقنية، والتي من خلالها نصل إلى تحقيق الهدف الذي هو البرنامج النهائي لتسيير حظيرة سيارات.

## الفصل الثالث: الانجاز والتنفيذ

## المقدمة

أصبحت لغات البرمجة عالية المستوى قريبة جدا من لغة الإنسان بحيث أدى تطورها من يوم لآخر إلى جعل أغلبها سريعة وقوية وبسيطة خاصة المبنية من قبل شركات كبيرة كجوجل وفيس بوك وغيرها، في نفس السياق يسعى جميع مطوري اللغات إلى تحسين وتطوير تقنيات جديدة تنافس وتتحدى بقية اللغات والتقنيات من كل هذا يحتار مطور البرمجيات في اختيار لغة مناسبة له ولمشاريعه ومن هنا حاولنا في هذا المشروع البحث عن أبرز وأهم التقنيات الحديثة الآمنة والتي تشهد تطور ملفت والتي وظفناها في مشروعنا الحالي ووضعتنا لها شرح مبسط في هذه الفصل، وسنقوم بدراسة تفصيلية وانجاز قاموس معطيات يساعدنا على إنشاء قاعدة بيانات دقيقة وذات جودة عالية نستعملها لإتمام إنشاء النظام تسير السيارات

## 1. لغات التنفيذ المستخدمة:

## : MySQL 1-1

اعتمدنا في تطوير النظام على قاعدة البيانات مفتوحة المصدر على MySQL وهو نظام إدارة قواعد بيانات علائقي يعتمد التعامل معه على لغة إس كيو إل يعتبر MySQL نظام إدارة قواعد البيانات العلائقية الأشهر عالميا نظرا لكونها متاحة تحت ترخيص مفتوح المصدر GPL GUN، تم تصميم MySQL حول ثلاث مفاهيم رئيسية وهي السرعة والثبات وسهولة الاستخدام.

تعتبر أهم سمات نظم قواعد البيانات MySQL السرعة والثبات، الشيء الذي يفسر كثرة استخدامها من طرف المطورين والمدراء والمستخدمين حول العالم، وسوف نستعرض الآن ما يميز هذه القاعدة بشيء من التفصيل. [4]

**: PHP 2-1**

لغة PHP هي لغة برمجة لإنشاء تطبيقات وصفحات مواقع الإنترنت وهي اختصار لـ Pages Home Personale أو بصيغة أخرى هي لغة لبرمجة المواقع الإلكترونية.

بيئة تطوير لغة PHP هو نظام Linux هي لغة سهلة التعلم لأن المنطق البرمجي واحد وأوامرها تشبه إلى حد كبير أمها لغة C. [5]

**: JavaScript 3-1**

JavaScript هي لغة برمجة تُستخدم على نطاق واسع في تطوير تطبيقات الويب والتفاعلية. وُجدت JavaScript أصلاً كلغة برمجة متصفح لتمكين التفاعل مع صفحات الويب، ولكنها تطورت بشكل كبير لتصبح لغة برمجة متعددة الاستخدامات. تتيح JavaScript للمطورين إضافة وظائف ديناميكية وتفاعلية لمواقع الويب، مثل التحقق من البيانات المدخلة، وتغيير محتوى الصفحة دون إعادة تحميل الصفحة، وإنشاء تطبيقات الويب الثرية.

يتم تنفيذ JavaScript عادةً في متصفح الويب الخاص بالمستخدم، ولكن يمكن أيضًا تشغيله على الخوادم باستخدام تقنيات البرمجة إلى جانب الخادم مثل JavaScript. Node.js لديها مجموعة كبيرة من المكتبات والإطارات مثل React.js و AngularJS و Vue.js التي تساعد في تسهيل وتسريع عملية تطوير تطبيقات الويب.

بفضل شيوعها وقوتها، أصبحت JavaScript واحدة من أهم لغات البرمجة في عصر الويب الحديث، وتُستخدم في تطوير مجموعة واسعة من التطبيقات بما في ذلك تطبيقات الويب، وتطبيقات الجوال، وتطبيقات سطح المكتب، وحتى تطبيقات الأجهزة المنزلية مثل الأجهزة الذكية. [6]

## 2. أدوات التطوير:

### : BOOSTRAP 1-2

هي مجموعة من الأدوات مفتوحة المصدر لإنشاء مواقع الويب وتطبيقات الانترنت. قامت شركة تويتر بتطويرها مع Thornton Jacob وتم إنشاؤها لمساعدة المصممين والمطورين على بناء منتجات مذهلة بسرعة وفعالية، الهدف منها هو توفير مكتبة واسعة مرنة وموثقة بشكل جيد للتصاميم "مكررة الاستعمال" المبنية باستخدام HTML و CSS و JavaScript كي يقوم المطورون باستعمالها والإبداع باستخدامها.

توسع المشروع ليصبح يعرف ب Bootstrap ويستقل عن تويتر لأنه كما صرح المسؤولين أصبح مشروع كبير الآن يتضمن العشرات من العناصر حيث أصبح المشروع الأكثر شعبية على موقع Hub Git بعد أن حصل على أكثر من 13 ألف متابع و 2000 تطوير فرعي. [7]

**: XHTML 3-2**

Language Markup HyperText ، ويشار إليها اختصاراً بـ HTML، وتعرف بلغة ترميز النص التشعبي، هي عبارة عن لغة ترميز خاصة تستخدم لغايات تصميم وإنشاء صفحات الويب، كما يمكننا القول بأنها عبارة عن الهيكل الرئيسي لصفحات الويب والبنية التحتية لها؛ إذ تقدم وصفا مفصلاً حول الكيفية التي ستكون عليها آلية عرض محتويات الموقع الإلكتروني ويكون ذلك بتقسيمه إلى عنوان وفقرات، ويتم ذلك كله بالاعتماد على ما يعرف بالوسوم.

**: CSS 4-2**

CSS هو اختصار لـ "Cascading Style Sheets"، وهو لغة تنسيق تُستخدم لتحديد شكل وتنسيق المستندات الإلكترونية، وبشكل رئيسي صفحات الويب. يُعتبر CSS أحد الأسس الثلاثة لبناء صفحات الويب، إلى جانب HTML (لتحديد الهيكل) وJavaScript (لتحديد السلوك والتفاعل).

تسمح لغة CSS بتعريف الأنماط والمظاهر الخاصة بالعناصر المختلفة في صفحة الويب، مثل الألوان والخطوط والهوامش والتنسيقات الخاصة بالحاويات والترتيبات، بالتالي يمكن استخدام CSS لتحسين جاذبية ومظهر صفحات الويب وجعلها تبدو أكثر احترافية وجاذبية للمستخدمين.

يتم استخدام CSS عن طريق كتابة مجموعة من القواعد (Rules) التي تحدد كيفية تنسيق العناصر المختلفة في صفحة الويب. تتألف كل قاعدة من محدد (Selector) الذي يحدد العنصر المراد تنسيقه، ومجموعة من الخصائص (Properties) التي تحدد التنسيق الذي سيتم تطبيقه على هذا العنصر على سبيل المثال، يمكن استخدام CSS لتحديد اللون والخط لنص العناوين،

وتحديد الحدود والألوان لصور، وتحديد الألوان والخلفيات لخلفية الصفحة، وغير ذلك الكثير.

باستخدام CSS بشكل فعال، يمكن للمطورين تخصيص وتنسيق مظهر صفحات الويب بشكل كامل وتنفيذ تصميمات مختلفة وجذابة بسهولة. [9]

#### :Laravel 3-4

Laravel هو إطار عمل مفتوح المصدر لتطوير تطبيقات الويب بلغة PHP. يهدف Laravel إلى جعل عملية تطوير التطبيقات أكثر سلاسة وفعالية من خلال توفير مجموعة من الأدوات والميزات الجاهزة التي تساعد المطورين على كتابة كود نظيف ومنظم بسرعة وسهولة.

#### بعض الميزات الرئيسية لـ: Laravel

1. **سهولة الاستخدام:** يتيح Laravel بناء تطبيقات الويب بسهولة بفضل توثيقه الشامل والأدوات القوية التي يوفرها، مما يجعله مناسباً للمطورين من جميع المستويات.

2. **نظام التوجيه: (Routing)** يوفر Laravel نظام توجيه مرن وسهل الاستخدام، مما يسمح بتحديد المسارات المختلفة للتطبيق بطريقة بسيطة ومنظمة.

3. **نظام القوالب Blade: (Blade)** هو محرك القوالب المدمج في Laravel، الذي يساعد على إنشاء واجهات مستخدم ديناميكية عن طريق دمج كود PHP داخل HTML بسهولة.

4. **نظام Eloquent: (Eloquent) ORM** هو نظام ORM مدمج في Laravel، الذي يسهل التعامل مع قواعد البيانات من خلال استخدام نماذج كائنية التوجه بدلاً من كتابة استعلامات SQL معقدة.

5. **نظام الهجرة (Migrations):** يساعد نظام الهجرة في Laravel على إدارة هيكل قاعدة البيانات، مما يتيح إنشاء وتعديل الجداول والعلاقات بينها بطريقة منضبطة ومنظمة.
6. **الأمان:** يتضمن Laravel العديد من ميزات الأمان المدمجة مثل الحماية من هجمات SQL Injection ، XSS ، و CSRF ، بالإضافة إلى إدارة كلمات المرور بطريقة آمنة.
7. **التوثيق والصلاحيات:** يأتي Laravel بنظام متكامل لإدارة التوثيق وصلاحيات المستخدمين، مما يسهل التحكم في من يمكنه الوصول إلى أجزاء معينة من التطبيق.
8. **المكتبات والباكجات:** يوفر Laravel بيئة غنية بالمكتبات والباكجات التي يمكن إضافتها بسهولة إلى المشروع لتوسيع وظائفه. [10]

### 3. واجهات الموقع:

#### 3-1 واجهة تسجيل الدخول:

تسجيل الدخول

البريد الإلكتروني

admin@gmail.com

كلمة المرور

.....

تسجيل للدخول ←

الشكل ( 12 ) : تسجيل الدخول

#### 3-2 واجهة السائقين :

حديقة سيارات

السائقين

السيارات

التزويد بالوقود

الصيانة

المهمة

الاعدادات

المستخدمين

تسجيل خروج

يبحث

اضافة جديد +

المعرف	الاسم	العنوان	العمر	الأدوات
# 2	خارون	جزلة	22	
# 3	أيمن	كرويش	21	

الشكل (13) : السائقين

### 3-3 واجهة اضافة سائق:

الشكل (14): واجهة سائق

### 4-3 واجهة السيارات:

المعرف	رقم اللوحة	مودال	سنة اصدار	الحالة	الأنواع
1 #	11111111	FH Aero	2024	تمت	

الشكل (15): السيارات

### 3-5 واجهة اضافة سيارة:

### الشكل (16): واجهة اضافة سيارة

### 3-6 واجهة التزود بالوقود:

المعرف	السيارة	مكان التزويد	الكمية	تاريخ التزويد	الأنواع
1 #	Volvo - FH Aero	الوادي	25.00	2024-06-04	

### الشكل (17): التزود بالوقود

### 7-3 واجهة علمية التزود بالوقود:

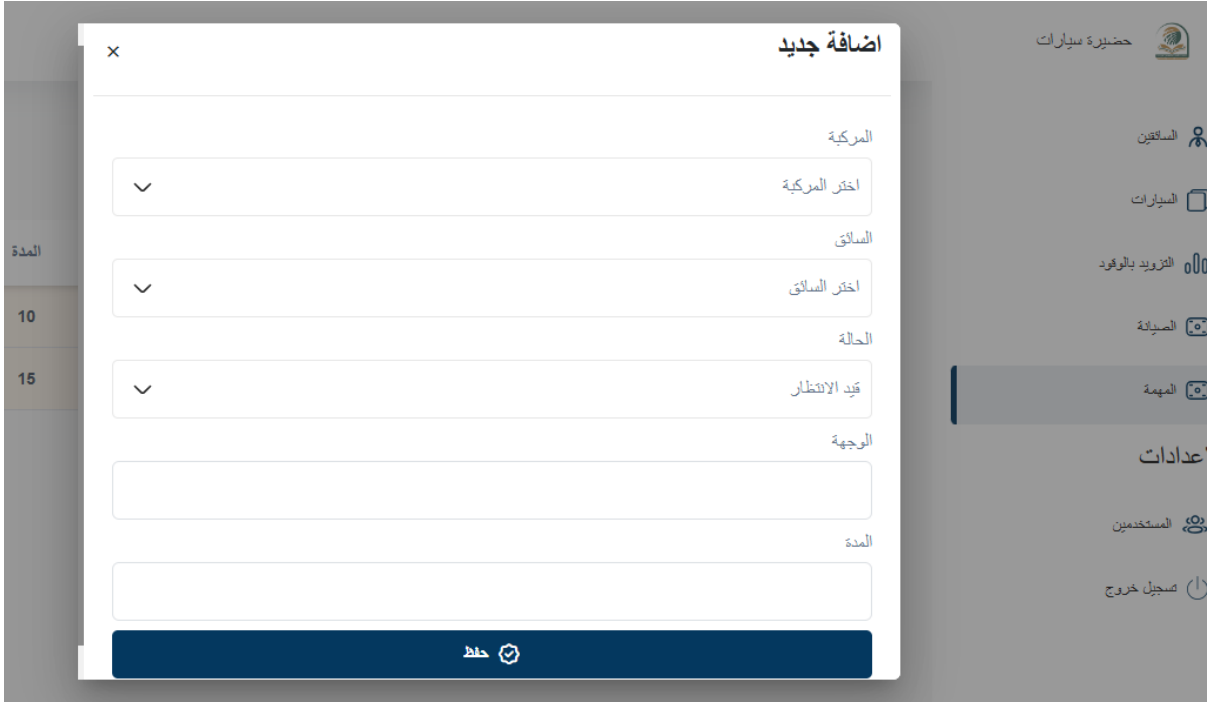
الشكل (18): واجهة عملية التزود

### 8-4 واجهة المهام :

المعرف	الموظف	السيارة	الوجهة	المدة	الأموات
1 #	أيمن كروش	Volvo - FH Aero	وهران	10	
2 #	هارون مزلة	Volvo - FH Aero	تلمسان	15	

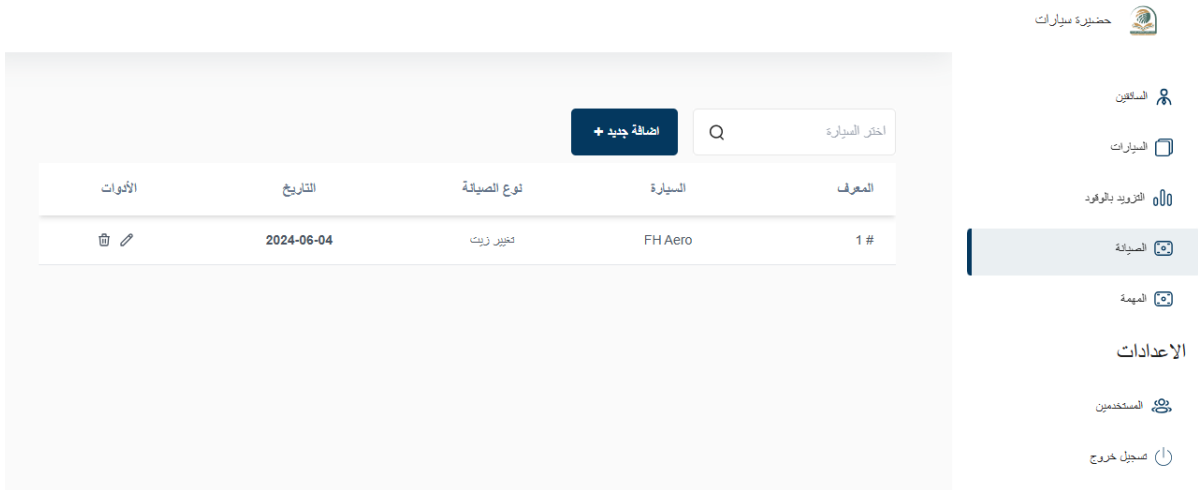
الشكل (19): واجهة المهام



### 9-4 واجهة اضافة المهمة :



الشكل (20): واجهة اضافة مهمة

### 10-4 واجهة الصيانة :



المعرف	السيارة	نوع الصيانة	التاريخ	الأدوات
1 #	FH Aero	تغيير زيت	2024-06-04	 

الشكل (21): واجهة الصيانة

#### 11-4 واجهة اضافة الصيانة :

الشكل (22) : واجهة اضافة الصيانة

#### 12-4 واجهة المستخدمين :

المعرف	الاسم	البريد الالكتروني	رقم الهاتف	الحالة	الأكوات
1 #	admin	admin@gmail.com	0553868769	نشط	🗑️ ✎️

الشكل (23) : واجهة المستخدمين

## الخاتمة العامة

في الختام، يمكن القول أن تطوير موقع لمراقبة وتسيير حظيرة مركبات يعد خطوة هامة ومؤثرة في مجال تكنولوجيا المعلومات وإدارة الأعمال. من خلال هذا الموقع، سيتمكن أصحاب الشركات والمؤسسات من تحسين الكفاءة التشغيلية لمركباتهم، وتقليل التكاليف، وزيادة العمر الافتراضي للمركبات من خلال صيانة دورية ومتابعة دقيقة. في النهاية، نأمل أن يكون هذا المشروع إضافة قيمة للشركات والمؤسسات، وأن يساهم في تحسين إدارتها للمركبات بطرق أكثر فعالية وأماناً، مما ينعكس إيجاباً على الأداء العام للشركة ومستوى الرضا لدى العملاء والمستخدمين.

## قائمة المصادر والمراجع:

- [1]: [The Role of Vehicle Tracking in Fleet Management: Enhancing Efficiency and Safety - GoFleet Tracking](#)
- [2]: [The Role of Vehicle Tracking in Fleet Management: Enhancing Efficiency and Safety - GoFleet Tracking](#)
- [3]: [Welcome To UML Web Site!](#)
- [4]: <https://www.mysql.com>
- [5]: [PHP: Hypertext Preprocessor](#)
- [6]: [Learn JavaScript Online - Courses for Beginners - javascript.com](#)
- [7]: [Bootstrap · The most popular HTML, CSS, and JS library in the world. \(getbootstrap.com\)](#)
- [9]: [CSS: Cascading Style Sheets | MDN \(mozilla.org\)](#)
- [10]: [Laravel - The PHP Framework For Web Artisans](#)