

بسم الله الرحمن الرحيم



جامعة بوليتكنك فلسطين

كلية تكنولوجيا المعلومات وهندسة الحاسوب

مشروع التخرج

(تطبيق المجمع الذكي)

فريق العمل :

كنان محمد عيايده      يزن معن قاضي

إشراف:

د. ضياء أبو زينة

## الإهداء

قال تعالى: { يرفع الله الذين آمنوا منكم والذين أوتوا العلم درجات

والله بما تعملون خبير } المجادلة .11

إلى شهدائنا الأبرار، الذين قدموا أرواحهم الزكية من أجل أرضنا وكرامتنا  
إلى الأسرى الأبطال الذين يحملون عبء الوطن في قلوبهم وقيود الظلم على أيديهم  
إلى أسرتي الحنونة التي كانت دعمًا لا ينضب، وركنًا ثابتًا في رحلتي إلى والديّ  
الذين علموني قيم العزيمة والإصرار إلى كل الأمهات اللواتي يشكلن قوة وجمال  
الحنان وإلى الآباء الذين يعملون بصمت وإخلاص إلى أساتذتي الكرام، الذين أضاءوا  
دروب العلم ليسطع نوره في حياتي إلى أصدقائي الأوفياء الذين مروا بجانبني في كل  
المحطات إلى كل من عرفنا وتعامل معنا

## الشكر والتقدير

قال رسول الله ﷺ : " من لا شكر الناس لا شكر الله "  
نحمد الله سبحانه وتعالى على إحسانه وتوفقه على ما أسداه علينا من نعم لا تعد ولا تحصى  
ونصلّ ونسلم على نبي هذه الأمة الذي جاء رحمة للعالمين. وبعد  
يسرنا أن نتقدم بوافر شكرنا، وعظيم امتناننا إلى الدكتور ضياء أبو زينة على ما غمرنا به من  
وافر علمه، وسديد توجيهاته؛ للخروج بهذا العمل، فجزاه الله عنا كل خير  
ونشكر جميع الأساتذة المحترمين اللذين رافقونا في رحلة العلم ولم يدخروا جهداً، كما  
نشكر جميع أصدقائنا وزملائنا الذين كانوا لنا خير مساعد ومعين  
وفي الختام، نسأل الله تعالى أن يجعل هذا الجهد المتواضع في ميزان حسناتنا وأن يكون لنا  
عوناً في مسيرتنا المستقبلية. ونرجو من الله العلي القدير دوام التوفيق والنجاح

## المخلص:

يقدم هذا المشروع لتخصص علم الحاسوب تطبيقًا مبتكرًا للهاتف المحمول مصممًا لتعزيز نظام النقل في فلسطين. سيتم تطوير هذا التطبيق باستخدام **Android Studio**، ويهدف إلى تسهيل التفاعل بين الركاب وسائقي النقل ومديري الطوابق، وبالتالي تحسين الكفاءة العامة لوسائل النقل العام. بالنسبة للركاب، يوفر التطبيق تحديثات في الوقت الفعلي عن حالة النقل، مما يسمح لهم باتخاذ قرارات سفر مدروسة وتقليل أوقات الانتظار. يستفيد سائقو النقل من التطبيق من خلال الحصول على معلومات حول عدد الركاب الذين ينتظرون الخدمات، مما يمكنهم من تحسين مساراتهم وجدولهم الزمنية. من الميزات المهمة لهذا النظام دور مديري الطوابق، الذين يقومون بمراقبة حالة المجمع والتطبيق للتأكد من دقة عدد الركاب والسائقين المتاحين لكل وجهة. وهذا يضمن دقة وموثوقية بيانات التطبيق. يقتصر نطاق التطبيق على فلسطين، ويتناول تحديات النقل المحلي ويقدم نموذجًا لحلول مماثلة في سياقات أخرى، على سبيل المثال واجهة الشكاوي، وواجهة مخصصة للمفقودات التي تعرض الأغراض التي قد ينساها الراكب في احد المركبات او المجمع. يعد اختيار قاعدة البيانات المناسبة لإدارة وتخزين البيانات عنصرًا أساسيًا في هذا المشروع، والذي سيتم تحديده مع تقدم المشروع. لا يعد هذا التطبيق بإحداث ثورة في طريقة تفاعل الركاب والسائقين داخل نظام النقل الفلسطيني فحسب، بل يوضح أيضًا إمكانات التكنولوجيا في تعزيز البنية التحتية للنقل العام للمناطق المجاورة.

## **Abstract**

This graduation project in computer science introduces an innovative mobile application designed to enhance the transportation system in Palestine. Developed using Android Studio, this app aims to streamline the interaction between passengers, transport drivers, and floor managers, thereby improving the overall efficiency of public transport. For passengers, the application provides real-time updates on transport status, allowing them to make informed travel decisions and minimize waiting times. Transport drivers benefit from the app by gaining insights into the number of passengers waiting services, enabling them to optimize their routes and schedules. A crucial feature of this system is the role of floor managers, who monitors the app with current information about passenger numbers and available drivers for each destination. This ensures the accuracy and reliability of the app's data, which is pivotal for its effectiveness. The application's scope is confined to the Palestine, addressing local transportation challenges and providing a model for similar solutions in other contexts , such as a complaints tab and a lost-found tab to show missing / forgotten items by the passengers. The selection of an appropriate database for managing and storing data is a key component of this project, which will be determined as the project progresses. This application not only promises to revolutionize the way passengers and drivers interact within the Palestinian transport system but also illustrates the potential of technology in enhancing public transportation infrastructure.

## المحتويات

9	الفصل الأول: المقدمة
10	1.1 المقدمة:
11	1.2 فكرة المشروع:
11	1.3 دوافع بناء المشروع :
11	1.4 نطاق المشروع:
12	1.5 أهداف المشروع:
12	1.6 الأسباب وراء قرار عمل تطبيق بدلا من موقع:
13	1.7 الجدول الزمني:
15	1.8 مخطط السياق (CONTEXT DIAGRAM):
15	1.9 ملخص عام عن الفصل الأول:
15	الفصل الثاني: مواصفات المتطلبات
17	2.1 مقدمة الفصل
17	2.2 الجهات الفاعلة في النظام
17	2.3 متطلبات النظام
20	2.4 مخطط حالة الاستخدام (USE-CASE DIAGRAM)
35	الفصل الثالث: تصميم النظام
36	3.1 مقدمة الفصل
36	3.2 معمارية النظام
38	3.3 النماذج (MODELS)
40	3.4 جداول قاعدة البيانات
45	3.5 واجهات التطبيق (USER INTERFACE)
47	الفصل الرابع: البناء والتنفيذ
48	4.1 مقدمة الفصل
48	4.2 لغات البرمجة والتقنيات
50	4.3 تشغيل النظام
51	4.4 التعليمات البرمجية
53	4.5 ملخص الفصل
54	الفصل الخامس: فحص النظام
54	5.1 مقدمة الفصل
54	5.2 الآلية المتبعة في فحص النظام
58	5.3 تقنيات التحقق
59	5.4 ملخص الفصل

## قائمة الجداول

- 13 ..... جدول التقييم الزمني (1.1)
- 14 ..... جدول توزيع المهام على الوقت (1.2)
- 23 ..... جدول (2.1) إضافة سائقين ومديري طوابق
- 24 ..... جدول (2.2) تسجيل الدخول
- 25 ..... جدول (2.3) تسجيل خروج
- 25 ..... جدول (2.4) تغيير كلمة المرور
- 26 ..... جدول (2.5) تغيير الصورة الشخصية
- 27 ..... جدول (2.6) عرض عدد الركاب
- 28 ..... جدول (2.7) تعديل حسابات المستخدمين
- 29 ..... جدول (2.8) حذف حسابات المستخدمين
- 30 ..... جدول (2.9) عمل شكوى
- 30 ..... جدول (2.10) عرض الشكوى
- 31 ..... جدول (2.11) التواصل مع السائقين
- 31 ..... جدول (2.12) عرض المركبات القادمة
- 32 ..... جدول (2.13) عرض موقع المركبات القادمة
- 32 ..... جدول (2.14) عرض المركبات المتاحة
- 33 ..... جدول (2.15) تأكيد التواجد
- 33 ..... جدول (2.16) إضافة مفقودات
- 34 ..... جدول (2.17) عرض مفقودات
- 34 ..... جدول (2.18) حجز دور
- 40 ..... جدول (3.1) جداول قاعدة البيانات
- 41 ..... جدول (3.2) جدول السائقين
- 42 ..... جدول (3.3) جدول مديري الطوابق
- 42 ..... جدول (3.4) جدول المسؤول
- 43 ..... جدول (3.5) جدول عدد الركاب
- 43 ..... جدول (3.6) جدول الشكاوي
- 44 ..... جدول (3.7) جدول المفقودات

## قائمة الأشكال

15	الشكل (1.1) مخطط السياق
22	الشكل (2.1) مخطط حالة الاستخدام USE CASE
36	الشكل (3.1) معمارية الطبقات LAYERED ARCHITECTURE
37	الشكل (3.2) معمارية نموذج عرض نموذج العرض MVVM
38	الشكل (3.3) CLASS DIAGRAM
39	الشكل (3.4) DATABASE MAPPING
45	الشكل (3.7) واجهة اضافة سائق
45	الشكل (3.5) واجهة تسجيل الدخول
45	الشكل (3.6) واجهة الصفحة الرئيسية
46	الشكل (3.10) واجهة الإدارة للمسؤول
46	الشكل (3.9) واجهة الشكاوي
46	الشكل (3.8) واجهة تقديم شكوى
46	الشكل (3.13) السائقين للادمن
46	الشكل (3.11) المركبات القادمة
46	الشكل (3.12) مدراء الطوابق للادمن
46	الشكل (3.15) واجهة تغيير كلمة المرور
46	الشكل (3.14) واجهة إضافة مفقودات
46	الشكل (4.1) FUNCTION ADD DRIVER
46	الشكل (4.2) جزء من كود طلب صلاحيات الموقع
46	الشكل (5.1) INPUT VALIDATION
46	الشكل (5.2) طلب صلاحيات الموقع

# الفصل الأول

## المقدمة

- 1.1 المقدمة
- 1.2 فكرة المشروع
- 1.3 دوافع بناء المشروع
- 1.4 نطاق المشروع
- 1.5 أهداف المشروع
- 1.6 الأسباب وراء القرار
- 1.7 الجدول الزمني
- 1.8 مخطط السياق
- 1.9 ملخص عن الفصل الأول

## 1.1 المقدمة:

بناء على إحصائيات يوجد ما يقارب 7.33 مليار مستخدم للهواتف الذكية، ومن بينهم 49% مستخدم أندرويد حول العالم، أما في فلسطين عدد مستخدمي الهواتف الذكية 4,4 مليون مستخدم للهواتف الذكية ونسبة مستخدمي الأندرويد منهم هو 80% لسهولة استخدامهم وإمكانية الحصول عليهم بسبب أسعارها المعقولة. هناك ما يقارب 3.5 مليون تطبيق على متجر بلاي التابع لجوجل. والفضل كله يعود إلى التطور التقني وتقدم الإنسان السريع في تحريك العمليات للعمل عن طريق الإنترنت ولكن التغيير لم يحدث في ليلة إنما حدث بشكل تدريجي متصاعد مستمر وما زال يحدث.

وهذا التطور بدأ في تسعينات القرن الماضي عندما بدأ الناس باستخدام الإنترنت على نحو رسمي لأنه في بداية ظهوره كان للاستخدام العسكري فقط. ومنذ ذلك الوقت التطور التقني في مجال الويب بدأ بالنمو على نحو سريع.

في عالم اليوم سريع الخطى، تعد كفاءة أنظمة النقل أمرًا بالغ الأهمية لحسن سير الحياة اليومية، خاصة في المناطق المكتظة بالسكان. وإدراكًا لهذه الحاجة، يقدم مشروع التخرج لدينا في علوم الكمبيوتر تطبيقًا رائدًا للهاتف المحمول، مصممًا خصيصًا لتحسين تجربة النقل في فلسطين. يعمل هذا التطبيق، الذي تم تطويره باستخدام Android Studio ، بمثابة جسر ديناميكي بين الركاب وسائقي النقل، مما يضمن تجربة سفر سلسة وفعالة لجميع المشاركين.

في قلب هذا التطبيق المبتكر تكمن وظيفته المزدوجة. وبالنسبة للركاب، فهو يوفر تحديثات في الوقت الفعلي عن حالة خدمات النقل المختلفة، مما يمكنهم من اتخاذ قرارات مستنيرة بشأن خطط سفرهم. لا توفر هذه الميزة الوقت الثمين فحسب، بل تقلل أيضًا من عدم اليقين والضغط المرتبط بانتظار النقل. على الجانب الآخر، يستفيد سائقو النقل بشكل كبير من التطبيق لأنه يتيح لهم عرض عدد الركاب المنتظرين وتخطيط مساراتهم وجدولهم وفقًا لذلك. ويعزز هذا التبادل المتبادل للمعلومات نظام نقل أكثر تنظيمًا وكفاءة، مما يعود بالنفع على الركاب والسائقين على حد سواء.

بالإضافة إلى ذلك، يتضمن التطبيق دورًا فريدًا لمديري الطوابق، الذين يلعبون دورًا محوريًا في الحفاظ على فعالية النظام. هؤلاء المديرون مسؤولون عن الحفاظ على دقة البيانات في التطبيق من حيث أعداد الركاب والسائقين المتاحين لكل وجهة، مما يضمن بقاء البيانات الموجودة على التطبيق حديثة. ومع تطور المشروع، قمنا باختيار Firebase كقاعدة بيانات ، نظرًا لدورها في إدارة وتخزين هذه المعلومات المهمة. ومن خلال التركيز على احتياجات النقل المحددة داخل فلسطين، لا يعالج هذا التطبيق التحديات المحلية فحسب، بل يشكل أيضًا سابقة لكيفية الاستفادة من التكنولوجيا لتعزيز أنظمة النقل العام في مناطق مماثلة

## 1.2 فكرة المشروع:

تقوم فكرة المشروع على إنشاء تطبيق خاص بمجمع البلدية للسيارات بحيث يساعد الركاب والسائقون , يستطيع الركاب عن طريق التطبيق رؤية السيارات المتوفرة والقادمة وعدد الركاب، ويستطيع السائق معرفة عدد الركاب، وإذا كان يوجد سيارات كافية. ويقوم مدير الطابق بالحرص على التواصل مع السائقين في حالات الضرورة. ويحجز السائق دور في التطبيق بحيث يرى الركاب إن هنالك سيارات متوفرة أو قادمة إلى المجمع .

## 1.3 دوافع بناء المشروع :

نعرض في هذا البند الدوافع لبناء المشروع:

- الفكرة جاءت من عدم وجود تطبيق او موقع إلكتروني لتنظيم الحركة في المجمع
- حل مشكلة وقت الانتظار الطويل والتأخير
- يعد مجمع البلدية مركز النقل الرئيسي بين المدينة والقرى والمدن الأخرى حيث يستقبل المجمع عدد كبير من الأفراد واصبح وجود تطبيق يسهل على الركاب والسائقين حاجة ضرورية
- حاجة السائقين لوجود تطبيق يعطي معلومات عن المجمع في حال عدم وجودهم داخل المجمع

## 1.4 نطاق المشروع:

يتركز نطاق مشروع تخرجنا هذا على قطاع النقل في فلسطين، ومعالجة التحديات والمتطلبات الفريدة لوسائل النقل العام المحلية. وهو يركز على تطوير وتنفيذ تطبيق شامل للهاتف المحمول يعمل على سد الفجوة بين الركاب وسائقي المركبات، بتسهيل من مديري الطوابق الذين يتأكدون من دقة البيانات في التطبيق. سيُصمَّم التطبيق حصريًا لمنصة Android ، مع استخدام Android Studio و Flutter و Dart للتطوير، مع اتخاذ قرار مستقبلي بشأن قاعدة البيانات الأكثر ملاءمة لإدارة البيانات. يهدف هذا المشروع بشكل طموح إلى إحداث ثورة في تجربة النقل في فلسطين من خلال تقديم حل قائم على التقنية يعمل على تحسين تخطيط الطريق، ويقلل أوقات انتظار الركاب، ويحسن التواصل العام والكفاءة في نظام النقل. ومن خلال هذا النهج المحلي، لا يلبي المشروع الاحتياجات المحددة للركاب ومقدمي خدمات النقل الفلسطينيين فحسب، بل يشكل أيضًا سابقة للحلول الرقمية في وسائل النقل العام ضمن سياقات إقليمية مماثلة.

## 1.5 أهداف المشروع:

- راحة الركاب: توفير تحديثات لحظية عن حالة النقل للركاب، مما يقلل من أوقات الانتظار وعدم اليقين المرتبط بالنقل العام.
- تعزيز كفاءة مسارات السائقين: تمكين سائقي النقل من الوصول إلى معلومات لحظية حول أعداد الركاب في المجموع، مما يسمح لهم بتحسين مساراتهم وجداولهم لتحقيق أقصى قدر من الكفاءة وجودة الخدمة.
- تبسيط إدارة النقل: بمساعدة مديري الطوابق في التنسيق الفعال بين الركاب والسائقين من خلال توفير منصة لتحديث وإدارة البيانات اللحظية حول توافر السائقين والتواصل مع السائقين إذا لزم الأمر.
- تعزيز إنتاجية نظام النقل العام: زيادة إنتاجية النظام العام للنقل من خلال تقليل وقت الفراغ للسائقين ووقت انتظار الركاب، مما يؤدي إلى تجربة نقل عام أكثر كفاءة وسهولة في الاستخدام.
- تشجيع الاندماج التقني في النقل: إظهار الاستخدام الفعال للتقنية في إدارة النقل العام، وتحديد نموذج لكيفية تطبيق الحلول الرقمية في سياقات مماثلة.
- ضمان قابلية التوسع والمرونة: تصميم التطبيق بطريقة تسمح بتوسيعه وتكييفه بسهولة مع الاحتياجات المستقبلية، مثل الدمج مع أنواع مختلفة من وسائل النقل أو التوسع إلى مناطق أخرى.
- تصميم واجهة مستخدم سهلة الاستخدام: تطوير واجهة مستخدم بديهية وسهلة التنقل تلبي احتياجات المستخدمين جميعهم، بما في ذلك الركاب والسائقون ومديرو الطابق.
- أمان البيانات والخصوصية: إعطاء الأولوية لأمان وخصوصية بيانات المستخدم، مع ضمان حماية جميع البيانات الشخصية والتشغيلية ومعالجتها وفقاً للوائح حماية البيانات ذات الصلة.

## 1.6 الأسباب وراء قرار عمل تطبيق بدلاً من موقع:

- سهولة الوصول: تطبيق على الهاتف يسهل عملية الوصول إلى المعلومات بمجرد الاتصال بالإنترنت، مما يجعل الاطلاع على التطبيق أسهل ما يمكن.
- سهولة الاستخدام: تطبيق بواجهات مستخدم سهلة الفهم والتعامل معها، هو ضروري للمستخدمين المستهدفين.
- الفعالية من حيث التكلفة: عملية تطوير تطبيق الأندرويد خاصتنا يعتبر أقل تكلفة من صفحة ويب لعدم الحاجة إلى منصة استضافة، وفي حالة قررنا استخدام استضافة ستكون أقل تكلفة بكثير لأن البيانات المستخدمة قليلة نوعاً ما.
- سرعة الوصول للبيانات؛ نظراً لأن التطبيق حُمّل على الجهاز مسبقاً، وليس بحاجة إلى طلب بيانات كثيرة.
- عملية التطوير والصيانة على التطبيق أسهل وأقل تكلفة من صفحة الويب

## 1.7 الجدول الزمني:

رقم المهمة	اسم المهمة	الوقت اللازم بالأسابيع
1	تخطيط النظام وجمع المعلومات	4
2	تحديد متطلبات النظام	3
3	وصف متطلبات النظام وتحليلها	3
4	تصميم النظام	4
5	برمجة النظام وتطويره	7
6	فحص النظام	3
7	توثيق النظام	طوال فترة العمل

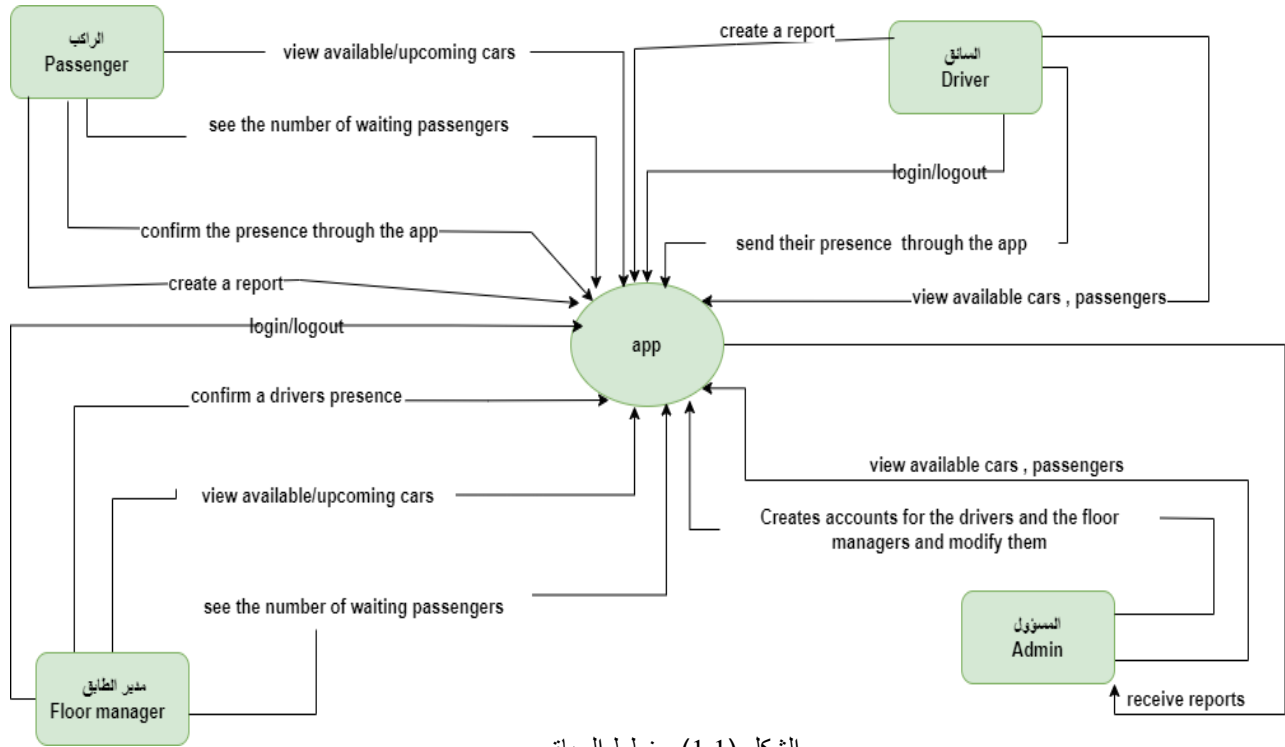
جدول التقييم الزمني (1.1)

الفصل الثاني			الفصل الأول			الأسابيع	المهمة
16-13	13-6	5-1	14-11	10-7	6-2		
							تخطيط النظام وجمع المعلومات
							تحديد متطلبات النظام
							وصف متطلبات النظام وتحليلها
							تصميم النظام
							برمجة النظام وتطويره
							فحص النظام
							توثيق النظام

جدول توزيع المهام على الوقت (1.2)



## 1.8 مخطط السياق (Context diagram):



مخطط السياق والمعروف أيضا باسم (Context diagram) ويعرف أيضا بالمستوى 0 لحركة البيانات والمعلومات، ويعتبر مستوى مرتفع لعرض البيانات. يعمل على عرض وتوضيح حركة البيانات بين النظام وباقي ال Entities من الشائع استخدامه من قبل محلي الأعمال والمشاريع لفهم منظور النظام، شروطه ومتطلباته. يقوم مخطط السياق بإظهار الشكل العام للنظام مع إبقاء البنية التحتية مخفية .

## 1.9 ملخص عام عن الفصل الأول:

يقدم الفصل الأول لمحة عامة عن المشروع، بما في ذلك الفكرة، والدافع، النطاق والأهداف والبدائل. يبدأ الفصل بمناقشة الحجم الكبير للإنترنت وتطور تقنية الويب. ثم يعرض فكرة مشروعنا، تطبيق يمكّن الركاب من عرض المركبات المتاحة. يشرح الفصل الدافع وراء المشروع، وهو مساعدة الركاب والسائقين معرفة عدد المركبات والركاب المتاحين. كما يناقش نطاق المشروع وأهدافه. يعرض الفصل بعد ذلك قرار استخدام التطبيق عوضا عن موقع ويب، ويوفر جدولاً زمنياً لإنجاز المشروع. وأخيراً، يختتم الفصل بمناقشة مخطط سياق (context diagram) الخاص بالمشروع وفائدته لفهم تدفق معلومات المشروع

## الفصل الثاني

### مواصفات المتطلبات

2.1 مقدمة الفصل

2.2 ممثلو النظام

2.3 متطلبات النظام

2.4 مخطط حالة الاستخدام

2.5 ملخص الفصل

## 2.1: مقدمة الفصل

يركز هذا الفصل على مواصفات المتطلبات لمشروعنا. يبدأ مع لمحة عامة عن النظام العام الذي سيعمل فيه مشروعنا، بما في ذلك مزايا وعيوب. ثم يوضح الفصل أهداف المشروع التي ذكرناها في الفصل السابق. يشمل تحليل المتطلبات ووصف المشروع نموذج السياق (Use Case diagram) لخاص بالنظام، بالإضافة إلى تحليل تفصيلي لمتطلباته. وسيقدم هذا التحليل لمحة موجزة عن المتطلبات اللازمة للتنفيذ الناجح للمشروع، ويشمل ذلك تحديد المتطلبات الوظيفية، غير الوظيفية، المتطلبات العامة وعمل مخطط سياق للمشروع.

## 2.2: ممثلو النظام

- الركاب: المستخدمون الذين يبحثون عن تحديثات في الوقت الفعلي حول خدمات النقل لاتخاذ قرارات السفر.
- السائقون : المستخدمون الذين يمكنهم تسجيل الدخول في التطبيق وعرض عدد الركاب المنتظرين وتخطيط مساراتهم وجدولهم وفقاً لذلك.
- مديرا الطوابق: الجهات الممثلة الرئيسية المسؤولة عن الحفاظ على دقة أعداد الركاب وضمان توفر سائقين لكل وجهة، مما يضمن بقاء بيانات التطبيق دقيقة وحديثة.
- مسؤولو النظام: الإشراف على وظائف التطبيق الشاملة، إدارة المستخدم وسلامة البيانات وإضافة حسابات السائقين ومديري الطوابق.

## 2.3: متطلبات النظام

من أهم الخطوات في عملية بناء المشاريع هي عملية جمع المتطلبات وتقييمها، ويتم تقسيم العمل الذي يتعين على ممثلي النظام القيام به إلى المهام التي يمكن إدارتها بطريقة متكاملة تفي بوظائفها الأساسية. متطلبات النظام تقسم إلى قسمين رئيسيين وهما

- 1-متطلبات وظيفية
- 2-متطلبات غير وظيفية

1- المتطلبات الوظيفية للمشروع : وهي مواصفات تحدد مميزات كل وظيفة في النظام.

الركاب:

- عرض المركبات المتاحة: يمكن للركاب رؤية المركبات المتاحة في الوقت الفعلي.
- عرض موقع المركبات القادمة الى المجمع
- يستطيع تحديد الوجهة المراد الذهاب اليها لعرض البيانات المتعلقة بها ( عدد الركاب المنتظرين , عدد المركبات المتاحة , عدد المركبات القادمة)
- عرض عدد الركاب: يمكنهم عرض العدد الحالي للمسافرين المنتظرين.

- يستطيع تسجيل حضوره في المجمع
- يستطيع عمل شكوى
- يستطيع عرض منشورات المفقودات

#### السائقين:

- تسجيل الدخول/الخروج: يمكن للسائقين تسجيل الدخول / الخروج من حساباتهم التي تم انشائها من قبل مسؤول النظام.
- يستطيع تغيير كلمة المرور و الصورة الشخصية
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار.
- عرض المركبات المتاحة: يمكنه رؤية عدد المركبات المتاحة في الوقت الفعلي
- عرض موقع المركبات القادمة الى المجمع
- يستطيع حجز دور
- يستطيع عمل شكوى
- يستطيع اضافة منشور على صفحة المفقودات و عرض المفقودات

#### مديري الطوابق:

- تسجيل الدخول/الخروج من النظام: الوصول إلى بوابة الإدارة الخاصة بهم.
- يستطيع تغيير كلمة المرور و الصورة الشخصية
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار
- عرض المركبات المتاحة: يمكن لمدير الطابق رؤية المركبات المتاحة في الطابق المسؤول عنه
- عرض موقع المركبات القادمة الى المجمع
- التواصل مع السائقين
- يستطيع عمل شكوى
- يستطيع اضافة منشور على صفحة المفقودات و عرض المفقودات

#### مدراء النظام:

- إدارة المستخدمين: الإشراف وإدارة حسابات السائقين ومديري الطوابق ومخالفة المخالفين حيث يستطيع تعديل , حذف حسابات المستخدمين (السائقين ومديري الطوابق).
- يستطيع تغيير كلمة المرور والملف الشخصي
- اضافة سائقين و مديري الطوابق
- إدارة حسابات المستخدمين :

- يستطيع عرض الشكاوي والتعامل معها.
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار
- عرض المركبات المتاحة: يمكن لمدير الطابق رؤية المركبات المتاحة في الطابق المسؤول عنه
- عرض موقع المركبات القادمة الى المجمع
- التواصل مع السائقين
- يستطيع اضافة منشور على صفحة المفقودات وعرض المفقودات

2-متطلبات غير وظيفية: هي المواصفات التي تصف القدرات والقيود التشغيلية للنظام. المتطلبات غير الوظيفية للمشروع: تحدد هذه المتطلبات خصائص النظام غير المرتبطة بشكل مباشر بوظيفة النظام ولكنها ضرورية للنظام بشكل عام لزيادة فعاليته.

تشمل المتطلبات غير الوظيفية ما يلي:

- سهولة الاستخدام: يتمتع النظام بواجهة سهلة الاستخدام سهل التنقل، مع تعليمات واضحة ورسائل خطأ لتقليل الأخطاء او الارتباك للمستخدم.
- الموثوقية: التطبيق موثوقاً به للغاية، مع الحد الأدنى من التوقف أو المشكلات الفنية التي قد تؤثر سلباً على المستخدمين.
- قابلية التوسع: تصميم النظام قابل للتعامل مع عدد كبير من المستخدمين والبيانات، ويجب وقابل للتطوير بسهولة لاستيعاب النمو المستقبلي.
- الأمان: لدى النظام إجراءات أمنية قوية مطبقة لحماية بيانات المستخدم ومنع الوصول غير المصرح به أو اختراق البيانات.

**المتطلبات العامة للمشروع:**

- تطبيق بسيط وسهل الاستخدام , مع واجهة مستخدم سهلة الاستخدام تتيح للمستخدم التنقل بسهولة بين المعلومات
- التطبيق متاح للوصول في جميع الأوقات ووقت توقف قليل لضمان وصول المستخدم له عندما يريد
- التطبيق قابل للتطوير والتكيف مع الاحتياجات و بنية قابلة لعمل تحديثات وتطويرات بسهولة
- تصميم التطبيق مريح ويتيح لمن ليس لهم خبرة عالية في التعامل مع الهواتف استخدامه

## 2.4: مخطط حالة الاستخدام (Use-Case Diagram)

يوفر مخطط حالة الاستخدام عرضاً عالي المستوى لوظائف النظام من وجهة نظر مستخدميه، مع التركيز على ما يجب على النظام فعله بدلاً من التركيز على كيف ينبغي أن يتم فعله. سيتضمن مخطط حالة الاستخدام الجهات الفاعلة وحالات الاستخدام و سيكون مصمم ليكون بسيط وسهل الفهم.

ممثلو النظام واستخدامها:

### الركاب:

- عرض المركبات المتاحة: يمكن للركاب رؤية المركبات المتاحة في الوقت الفعلي.
- عرض موقع المركبات القادمة الى المجمع
- يستطيع تحديد الوجهة المراد الذهاب اليها لعرض البيانات المتعلقة بها (عدد الركاب المنتظرين , عدد المركبات المتاحة , عدد المركبات القادمة)
- عرض عدد الركاب: يمكنهم عرض العدد الحالي للركاب المنتظرين لكل وجهه.
- يستطيع تسجيل حضوره في المجمع
- يستطيع عمل شكوى
- يستطيع عرض منشورات المفقودات

### السائقين:

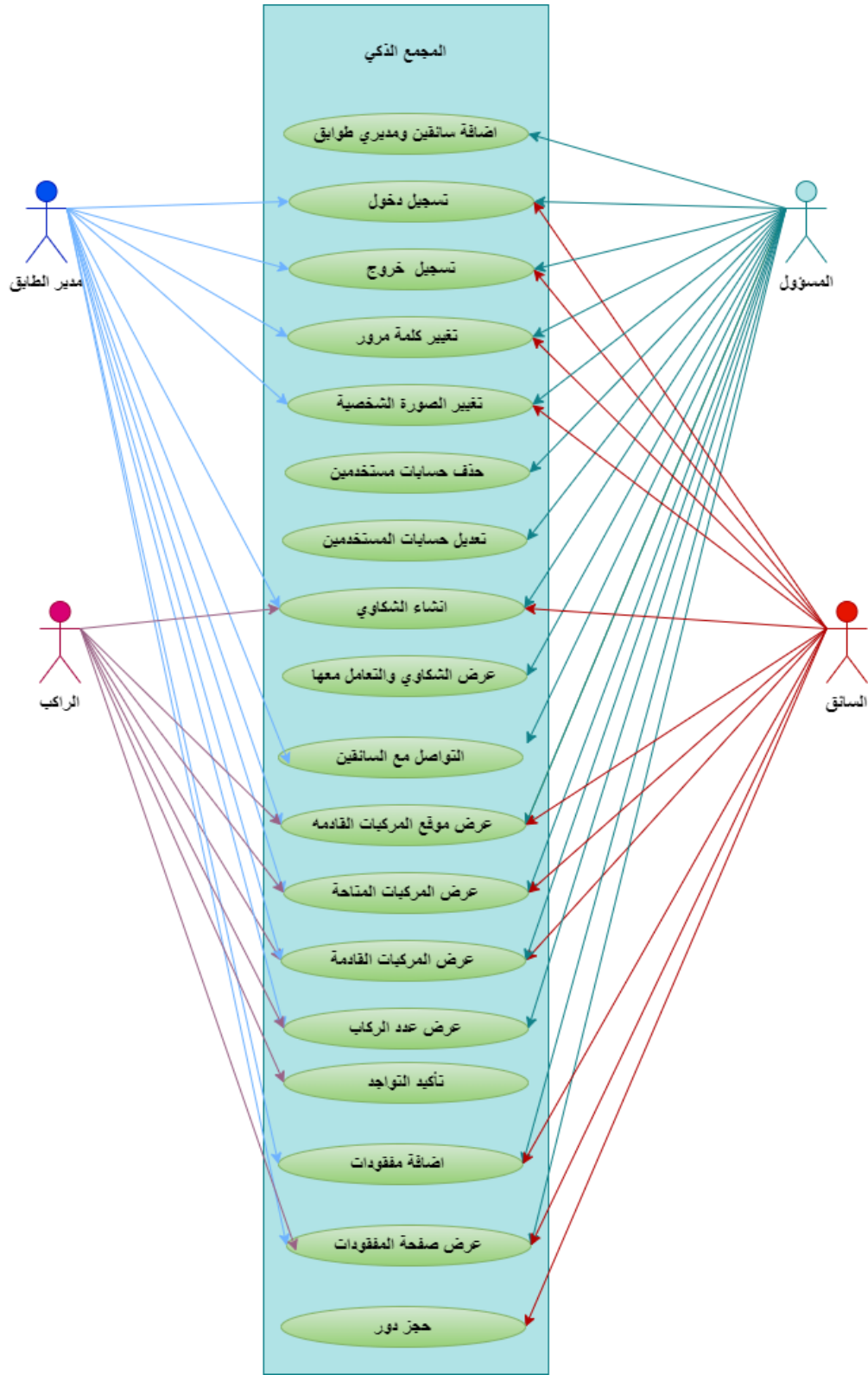
- تسجيل الدخول/الخروج: يمكن للسائقين تسجيل الدخول / الخروج من حساباتهم التي تم انشائها من قبل مسؤول النظام.
- يستطيع تغيير كلمة المرور و الصورة الشخصية
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار.
- عرض المركبات المتاحة: يمكنه رؤية عدد المركبات المتاحة في الوقت الفعلي
- عرض موقع المركبات القادمة الى المجمع
- يستطيع تسجيل حضوره في المجمع
- يستطيع عمل شكوى
- يستطيع اضافة منشور على صفحة المفقودات و عرض المفقودات

### مدراء الطوابق:

- تسجيل الدخول/الخروج من النظام: الوصول إلى بوابة الإدارة الخاصة بهم.
- يستطيع تغيير كلمة المرور , الصورة الشخصية وتعديل الملف الشخصي
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار
- عرض المركبات المتاحة: يمكن لمدير الطابق رؤية المركبات المتاحة في الطابق المسؤول عنه
- عرض موقع المركبات القادمة الى المجمع
- التواصل مع السائقين
- يستطيع عمل شكاوى
- يستطيع اضافة منشور على صفحة المفقودات وعرض المفقودات

### مسؤول النظام:

- تسجيل الدخول/الخروج: يمكن للسائقين تسجيل الدخول / الخروج من حساباتهم التي تم انشائها من قبل مسؤول النظام.
- يستطيع تغيير كلمة المرور والملف الشخصي
- اضافة سائقين و مديري الطوابق
- إدارة حسابات المستخدمين : حيث يستطيع تعديل , حذف حسابات المستخدمين (السائقين ومديري الطوابق)
- يستطيع عرض الشكاوي والتعامل معها.
- عرض عدد الركاب: يمكنه الوصول إلى البيانات في الوقت الفعلي حول عدد الركاب في ساحة الانتظار
- عرض المركبات المتاحة: يمكنه رؤية عدد المركبات المتاحة في الوقت الفعلي
- عرض موقع المركبات القادمة الى المجمع
- التواصل مع السائقين
- يستطيع اضافة منشور على صفحة المفقودات وعرض المفقودات




الشكل (2.1) مخطط حالة الاستخدام Use Case

إضافة سائقين ومديري طوابق	اسم الوظيفة
المسؤول	ممثل النظام
إضافة سائقين و مديري طوابق إلى قاعدة البيانات	هدف الوظيفة
يجب أن يكون ممثل النظام مسؤول	الشروط المسبقة
1- فتح التطبيق 2- التوجه الى صفحة الإدارة 3- الضغط على خيار سائقين او مديري الطوابق 4-الضغط على ايقونة الاضافة 5- يقوم المسؤول بإدخال البيانات اللازمة 6- التأكد من البيانات المدخلة وانهاء العملية	السيناريو
اذا كان المستخدم موجود مسبقا يظهر رسالة خطأ ان المستخدم موجود مسبقا	الاستثناءات


جدول (2.1) إضافة سائقين ومديري طوابق

اسم الوظيفة	تسجيل دخول
ممثل النظام	المسؤول , مدير الطابق , السائق
هدف الوظيفة	منح حق الوصول إلى التطبيق
الشروط المسبقة	يجب أن يكون لكل من المسؤول و مدير الطابق والسائق حساب مسجل
السيناريو	<p>1- يقوم المستخدم بفتح التطبيق</p> <p>2- النقر على أيقونة الملف الشخصي</p> <p>3- يقوم بإدخال بيانات الحساب (اسم المستخدم وكلمة المرور)</p> <p>4- يتم التحقق من بيانات المستخدم في قاعدة البيانات</p> <p>5- يمنح التطبيق حق الوصل إذا كانت المعلومات صحيحة</p>
الاستثناءات	إذا كانت بيانات تسجيل الدخول غير صحيحة يتم رفض الوصول الى التطبيق و يتلقى المستخدم رسالة خطأ ويطلب منه اعادة المحاولة


جدول (2.2) تسجيل الدخول

اسم الوظيفة	تسجيل خروج
ممثل النظام	المسؤول , مدير الطابق , السائق
هدف الوظيفة	تسجيل الخروج من حساب المستخدم
الشروط المسبقة	يجب أن يكون لكل من المسؤول و مدير الطابق والسائق حساب مسجل
السيناريو	1- يقوم المستخدم بفتح التطبيق 2- النقر على أيقونة  3- يقوم بالضغط على تسجيل خروج
الاستثناءات	

جدول (2.3) تسجيل خروج

اسم الوظيفة	تغيير كلمة المرور
ممثل النظام	المسؤول , مدير الطابق , السائق
هدف الوظيفة	تغيير كلمة المرور
الشروط المسبقة	يجب أن يكون لكل من المسؤول و مدير الطابق والسائق حساب مسجل
السيناريو	1- يقوم المستخدم بفتح التطبيق 2- النقر على أيقونة  3- يقوم بالضغط على تغيير كلمة المرور 4- يقوم المستخدم بإدخال كلمة المرور القديمة ثم الجديدة ثم تأكيد كلمة المرور الجديدة 5- الضغط على زر تحديث كلمة المرور او الغاء
الاستثناءات	

جدول (2.4) تغيير كلمة المرور

اسم الوظيفة	تغيير الصورة الشخصية
ممثّل النظام	المسؤول , مدير الطابق , السائق
هدف الوظيفة	تغيير الصورة الشخصية
الشروط المسبقة	يجب أن يكون لكل من المسؤول و مدير الطابق والسائق حساب مسجل
السيناريو	<p>1- يقوم المستخدم بفتح التطبيق</p> <p>2- النقر على أيقونة </p> <p>3- يقوم بالضغط على الصورة الشخصية</p> <p>4- يقوم برفع صورة او التقاط صورة</p>
الاستثناءات	


جدول (2.5) تغيير الصورة الشخصية

عرض عدد الركاب	اسم الوظيفة
الراكب , السائق , مدير الطابق , المسؤول	ممثّل النظام
عرض عدد الركاب في منطقة الانتظار	هدف الوظيفة
	الشروط المسبقة
1- فتح التطبيق 2--بعد فتح التطبيق يتم عرض الصفحة الرئيسية التي تحتوي المركبات المتاحة 4-اختيار الوجهة المراد الذهاب إليها 5- يقوم التطبيق بعرض عدد الركاب	السيناريو
إذا لم يكن هناك ركاب في قاعة الانتظار يظهر عدد الركاب 0	الاستثناءات


جدول (2.6) عرض عدد الركاب

اسم الوظيفة	تعديل حسابات المستخدمين
ممثل النظام	المسؤول
هدف الوظيفة	تعديل بيانات المستخدمين
الشروط المسبقة	يجب أن يكون مسؤول في النظام
السيناريو	<p>1- فتح التطبيق</p> <p>2- التوجه الى صفحة الادارة</p> <p>3- الضغط على خيار سائقين او مديري الطوابق</p> <p>4- الضغط على  بجانب المستخدم المراد التعديل على بياناته او حذف حسابه</p> <p>5- تظهر ثلاث خيارات (تعديل , حذف , الغاء)</p> <p>6- يقوم المسؤول باختيار عملية تعديل</p> <p>7- بعد الضغط على تعديل يتم التحويل الى صفحة تحتوي على بيانات المستخدم</p> <p>8- ادخال البيانات المراد تحديثها</p> <p>9- الضغط على زر تحديث</p>
الاستثناءات	

جدول (2.7) تعديل حسابات المستخدمين

اسم الوظيفة	حذف حسابات مستخدمين
ممثّل النظام	المسؤول
هدف الوظيفة	حذف مستخدم
الشروط المسبقة	يجب أن يكون مسؤول في النظام
السيناريو	<p>1- فتح التطبيق</p> <p>2- التوجه الى صفحة الإدارة</p> <p>3- الضغط على خيار سائقين او مديري الطوابق</p> <p>4- الضغط على  بجانب المستخدم المراد التعديل على بياناته او حذف حسابه</p> <p>5- تظهر ثلاث خيارات (تعديل , حذف , الغاء)</p> <p>6- يقوم المسؤول باختيار عملية حذف</p> <p>7- بعد الضغط على حذف تظهر صفحة لتأكيد الحذف</p>
الاستثناءات	


جدول (2.8) حذف حسابات المستخدمين

اسم الوظيفة	عمل شكوى
ممثل النظام	الراكب , السائق , مدير الطابق
هدف الوظيفة	تقديم شكوى للمسؤول
الشروط المسبقة	
السيناريو	1- يقوم المستخدم بفتح التطبيق والذهاب للصفحة الرئيسية 2- يقوم المستخدم بالضغط على ايقونة  ابلاغ 3- تظهر صفحة تقديم شكوى 4- يقوم المستخدم بكتابة اسمه ورقم هاتفه ووصف للشكوى
الاستثناءات	اذا كان المستخدم سائق او مدير طابق لا يطلب منه كتابة اسم او رقم هاتف

جدول (2.9) عمل شكوى

اسم الوظيفة	عرض الشكاوي
ممثل النظام	المسؤول
هدف الوظيفة	عرض الشكاوي التي تم تقديمها من المستخدمين والركاب
الشروط المسبقة	أن يكون ممثل النظام مسؤول
السيناريو	1- فتح التطبيق 2- الضغط على ايقونة الملف الشخصي 3- الضغط على خيار شكاوي 4- تظهر الشكاوي التي تم تقديمها
الاستثناءات	

جدول (2.10) عرض الشكوى

اسم الوظيفة	التواصل مع السائقين
ممثل النظام	المسؤول و مدير الطابق
هدف الوظيفة	التواصل مع السائقين عند الحاجه
الشروط المسبقة	ان يكون ممثل النظام مسؤول او مدير طابق ومسجل دخوله
السيناريو	1- يقوم ممثل النظام بفتح التطبيق 2- يقوم الضغط على ايقونة الملف الشخصي 3- يقوم بالضغط السائقين 4- يقوم بالضغط على ايقونة  بجانب السائق المراد التواصل معه 5- يقوم التطبيق بفتح سجل المكالمات مع وضع رقم السائق
الاستثناءات	

جدول (2.11) التواصل مع السائقين

اسم الوظيفة	عرض المركبات القادمة
ممثل النظام	الراكب , السائق , مدير الطابق , المسؤول
هدف الوظيفة	عرض المركبات القادمة
الشروط المسبقة	
السيناريو	1- فتح التطبيق 2- في الصفحة الرئيسية يقوم بالضغط على المركبات القادمة 3- تظهر معلومات المركبات القادمة
الاستثناءات	اذا كان لا يوجد مركبات قادمة يظهر لا يوجد مركبات قادمة

جدول (2.12) عرض المركبات القادمة

اسم الوظيفة	عرض موقع المركبات القادمة
ممثل النظام	الراكب , السائق , مدير الطابق , المسؤول
هدف الوظيفة	عرض موقع المركبات القادمة
الشروط المسبقة	
السيناريو	1- فتح التطبيق 2- في الصفحة الرئيسية يقوم بالضغط على المركبات القادمة 3- تظهر معلومات المركبات القادمة 4- بجانب كل مركبة يقوم بالضغط على ايقونة 9 5- يتم فتح الموقع في واجهة الخرائط
الاستثناءات	اذا كان لا يوجد مركبات قادمة يظهر لا يوجد مركبات قادمة

جدول (2.13) عرض موقع المركبات القادمة

اسم الوظيفة	عرض المركبات المتاحة
ممثل النظام	الراكب , السائق , مدير الطابق , المسؤول
هدف الوظيفة	عرض المركبات المتاحة للمستخدمين
الشروط المسبقة	يجب أن تكون المركبات متاحة وجاهزة للاستخدام
السيناريو	1- فتح التطبيق 2- بعد فتح التطبيق يتم عرض الصفحة الرئيسية التي تحتوي المركبات المتاحة 4- اختيار الوجهة المراد الذهاب إليها 5- يقوم التطبيق بعرض المركبات المتاحة للوجهه
الاستثناءات	إذا لم تكن هناك مركبات متاحة، يظهر لا يوجد مركبات متاحة للوجهه

جدول (2.14) عرض المركبات المتاحة

اسم الوظيفة	تأكيد التواجد
ممثل النظام	الراكب
هدف الوظيفة	تسجيل الحضور داخل المجمع
الشروط المسبقة	يجب التواجد داخل المجمع
السيناريو	1- فتح التطبيق 2- بعد فتح التطبيق يتم عرض الصفحة الرئيسية 3- يقوم الراكب باختيار الوجهه المراد الذهاب اليها 4-الضغط على مربع تأكيد التواجد
الاستثناءات	إذا لم يكن المستخدم في المجمع لا تكون قابلة للضغط

جدول (2.15) تأكيد التواجد

اسم الوظيفة	إضافة مفقودات
ممثل النظام	السائق , مدير الطابق , المسؤول
هدف الوظيفة	في حال قام راكب بفقدان شيء يتم نشره على صفحة المفقودات
الشروط المسبقة	يجب ان يكون مسجل بالنظام
السيناريو	1- فتح التطبيق 2- الذهاب على صفحة المفقودات 3-الضغط على إضافة منشور 4- ملئ البيانات (الوصف , صورة , الوجهة التي تم العثور فيها) 5- الضغط على نشر
الاستثناءات	

جدول (2.16) إضافة مفقودات

عرض مفقودات	اسم الوظيفة
الراكب, السائق, مدير الطابق, المسؤول	ممثل النظام
عرض المفقودات في واجهه خاصه حتى يستطيع صاحبها الوصول اليها	هدف الوظيفة
	الشروط المسبقة
1- فتح التطبيق 2- الذهاب على صفحة المفقودات 3- تظهر المنشورات	السيناريو
	الاستثناءات

جدول (2.17) عرض مفقودات

حجز دور	اسم الوظيفة
السائق	ممثل النظام
ان يقوم السائق بحجز دور في التطبيق	هدف الوظيفة
ان يكون حساب سائق	الشروط المسبقة
1- فتح التطبيق 2- الذهاب الى الصفحة الشخصية 3- الضغط على حجز دور	السيناريو
	الاستثناءات

جدول (2.18) حجز دور

## الفصل الثالث

### تصميم النظام

3.1 مقدمة الفصل

3.2 معمارية العمل

3.3 النماذج (Models)

3.4 جداول قاعدة البيانات

### 3.1: مقدمة الفصل

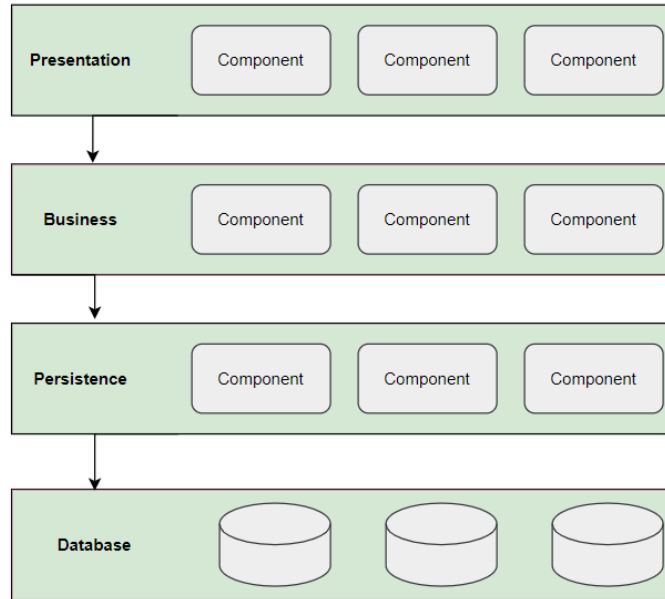
في هذا الفصل، سنستعرض مرحلة تصميم التطبيق، حيث سنفصل جميع أجزاء ومكونات النظام بالضبط ويعتبر هذا الجزء من المشروع ذا أهمية كبيرة، حيث يقدم نظرة شاملة عن جميع جوانب التطبيق. سيتم ذلك من خلال استخدام رسومات توضح وظيفة كل جزء ومكوناته، مما يُسهّل على المبرمج والقارئ فهم التطبيق وطريقة البناء بطريقة صحيحة. تأخذ مرحلة التصميم في الاعتبار تفضيلات المستخدم حيث يُؤكّد على تصميم النظام بما يتناسب مع توقعات المستخدمين من خلال النظر في واجهات المستخدم الرئيسية والاعتبارات المتعلقة بقاعدة البيانات. تعتبر مرحلة تصميم النظام جسراً بين تحليل المتطلبات ومرحلة التنفيذ، حيث يقدم خطة تنموية تُلبي المتطلبات، وتعزز تجربة المستخدم.

### 3.2: معمارية النظام

في بحثنا عن معماريات الأنظمة، قمنا بعمل تقييم لخيارين: معمارية الطبقات (LA) و نموذج عرض نموذج العرض (MVVM)

#### معمارية الطبقات (Layered Architecture)

معمارية الطبقات تقسم النظام الى عدة طبقات، كل طبقة تمثل مجموعة جزئية أو جزء من النظام. وعلى حسب بحثنا تبين ان هذه المعمارية تناسب تطبيقات الموبايل والجهاز الشخصي.



الشكل (3.1) معمارية الطبقات Layered Architecture

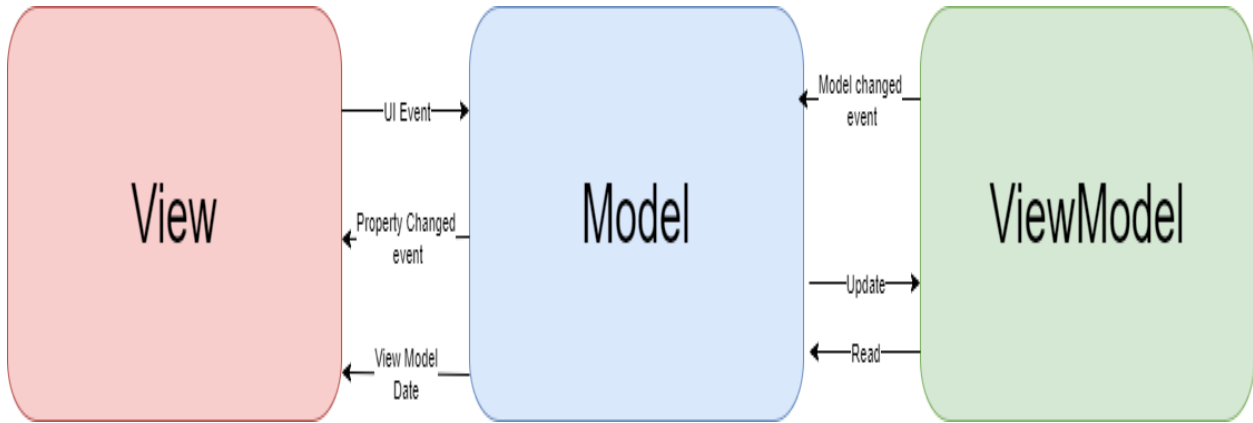
## معمارية نموذج عرض نموذج العرض او : Model-View-ViewModel

بعد الأخذ بالاعتبار المعماريين وجدنا أن معمارية ال ( MVVM ) مناسبة أكثر لمشروعنا لأنها تقدم عدة مزايا ومنها:

1. يمكن أن تعزز بشكل كبير عملية التطوير. من خلال فصل مكونات واجهة المستخدم عن كود التطبيق، مما يمكن المطورين التركيز على بناء التطبيقات دون التورط في تعقيدات تصميم واجهة المستخدم.

2. لا يؤدي هذا الفصل في التطبيق إلى تعزيز إمكانية قراءة التعليمات البرمجية وقابلية صيانتها فحسب، بل يسهل أيضًا التعاون الفعال بين أعضاء الفريق، حيث يمكن العمل على مكونات مختلفة في وقت واحد بواسطة أعضاء مختلفين في الفريق.

3. يعد (MVVM) أحد أنماط التصميم المستخدمة على نطاق واسع في تطوير تطبيقات Android. فهو يساعد في إدارة واجهة المستخدم الخاصة بالتطبيق، مما يجعله أكثر تنظيمًا وقابلية للصيانة. وتعمل MVVM على تعزيز فصل ال (UI) عن ال (Backend). فهو يجعل الكود أكثر قابلية للقراءة، وموحدة، وأسهل في تصحيح الأخطاء، وبالتالي تحسين الجودة الإجمالية للتطبيق.



الشكل (3.2) معمارية نموذج عرض نموذج العرض MVVM

## فيما يلي وصف لمكونات النظام حسب بنيته.

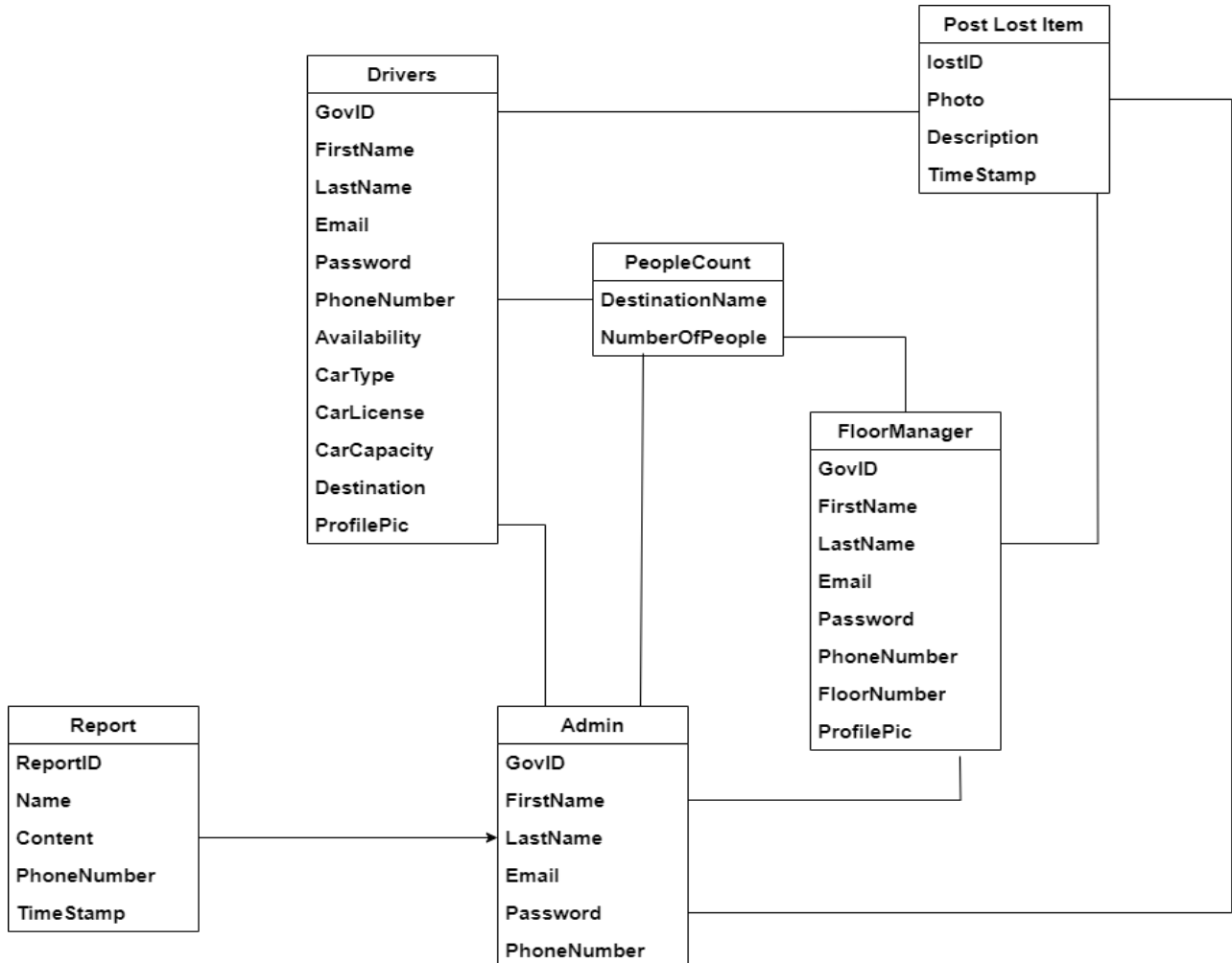
يعتبر مكون الموديل للنظام مسؤول عن إدارة البيانات والتفاعل مع قاعدة البيانات. ويشمل مختلف الفئات و يعتني بالعمليات مثل استرجاع البيانات وحفظها من وإلى قاعدة البيانات. الموديل يلعب المكون دورًا حاسمًا في بنية النظام الشاملة لأنه بمثابة جوهر البرنامج.

### 3.3 النماذج (Models)

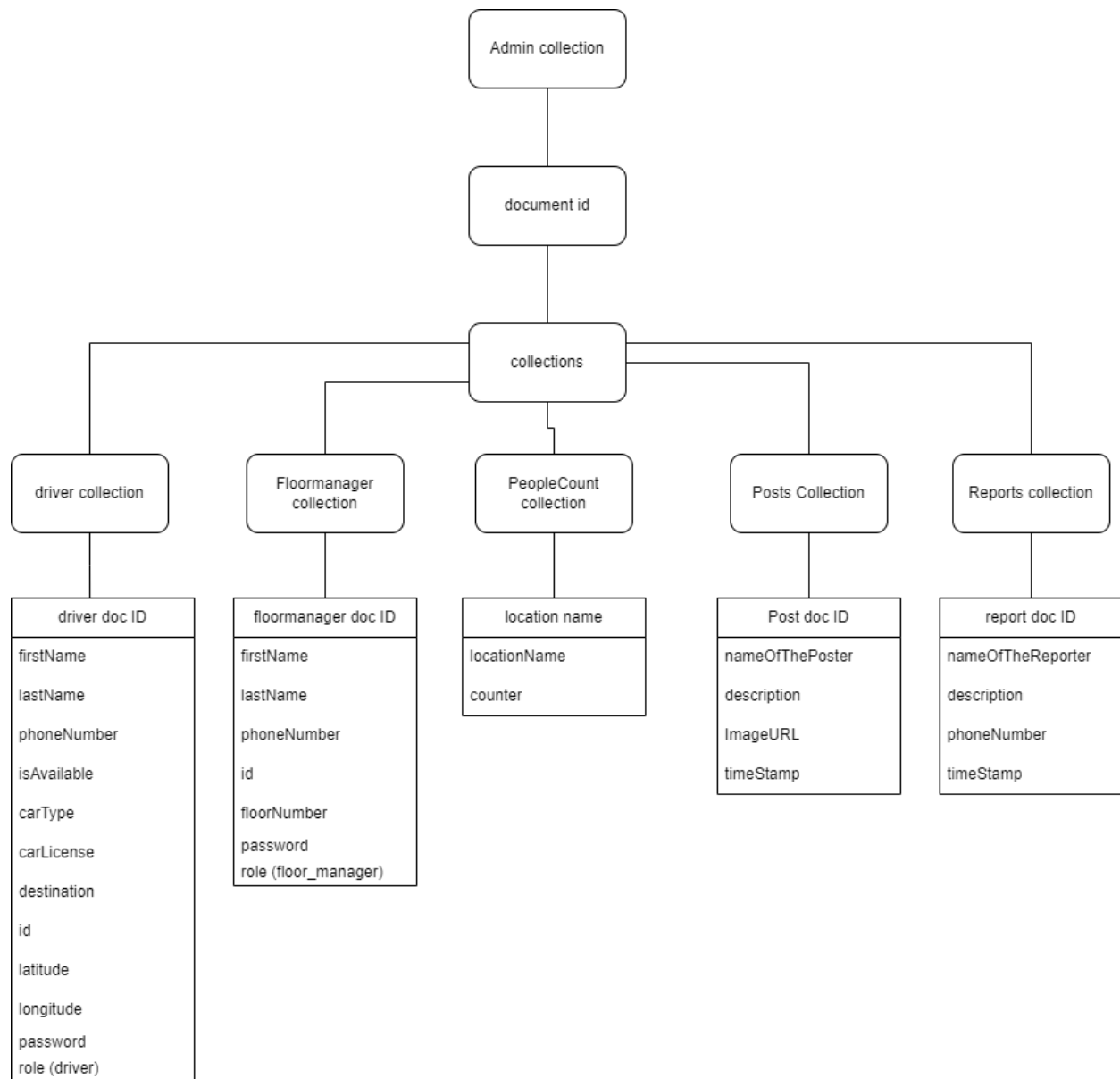
تتكون الطبقة النموذجية لنظامنا من عدة عناصر أساسية، مثل ما هو مبين في الرسم البياني الطبقة التالية.

### Class Diagram Model

يمثل Class Diagram Model التي تشكل جزءًا من طبقة النموذج في مشروعنا، وتلتزم بالنمط المعماري MVVM.



الشكل (3.3) Class Diagram



الشكل (3.4) Database Mapping

## 3.4 جداول قاعدة البيانات

يرتبط التطبيق بقاعدة بيانات مكونة من عدة جداول متصلة بعلاقات. في هذا القسم سيتم استخدام معلومات قاعدة البيانات حول مدخلات النظام والروابط بين جداوله لشرح مكونات النظام العديدة.

### جداول قاعدة البيانات

Name	Description
Admin	Store information about the Admins of the system
Driver	Store information about each driver in the system
FloorManager	Store information about each Floor Manager of the system
Destination	Store information about each Destination in the system
Report	Stores Floor Managers reports
Post Lost item	Stores the lost items details
PeopleCount	Stores the number of people present in the parking spot

جدول (3.1) جداول قاعدة البيانات

## Drivers Table

Field	Data Type	Null	Description
DriverID	Number		Unique Identifier for each driver
FirstName	String	NO	Admin first name
LastName	String	NO	Admin last name
Password	String	NO	Driver's password
PhoneNumber	Number	NO	Driver's Phone Number
CarTyp	String	NO	Unique Identifier for each photo
CarLicense	Number	NO	The Destination unique number
CarCapacity	Number	NO	Unique Identifier for each Admin
Email	String	No	The Driver Email
availability	Boolean		The availability of the driver

جدول (3.2) جدول السائقين

## FloorManager

Field	Data Type	Null	Description
FloorManagerID	Number	NO	Unique Identifier for each Manager
FirstName	String	NO	Manager's First Name
LastName	String	NO	Manager's Last Name
Password	String	NO	Manager's password
PhoneNumber	Number	NO	Manager's Phone Number
Email	String	NO	The FloorManager Email
FloorNumber	Number	NO	The FloorManager Phone Number

جدول (3.3) جدول مديري الطوابق

## Admin

Field	Data Type	Null	Description
AdminID	Number	NO	Unique Identifier for each Admin
FirstName	String	NO	Admin's First Name
LastName	String	NO	Admin's Last Name
Email	String	NO	Admin's Email
Password	String	NO	Admin password
PhoneNumber	Number	NO	Admin's Phone Number

جدول (3.4) جدول المسؤول

## PeopleCount

Field	Data Type	Null	Description
DestinationName	String	NO	The Name of the destination
NumberOfPeople	Integer	NO	The Number of people that goes to each desination

جدول (3.5) جدول عدد الركاب

## Report

Field	Data Type	Null	Description
ReportID	String		Unique Identifier for each Report
Title	String	NO	The Report Title (Subject)
content	String	NO	The information that the Reports cover
PhoneNumber	Number	NO	The phone number of the person reporting
TimeStamp	timestamp	NO	The date and time for the report

جدول (3.6) جدول الشكاوي

## Post Lost Item

Field	Data Type	Null	Description
DocID	String		Unique Identifier for each lost item
Description	String	NO	Short description of the item
Photo	Integer		Unique Identifier for each photo
TimeStamp	timestamp	NO	Unique Identifier for each driver

جدول (3.7) جدول المفقودات

## 3.6 واجهات التطبيق (User interface)

الشكل (3.7) واجهة اضافة سائق

الشكل (3.6) واجهة الصفحة الرئيسية

الشكل (3.5) واجهة تسجيل الدخول



الشكل (3.10) واجهة الإدارة للمسؤول



الشكل (3.9) واجهة الشكاوي



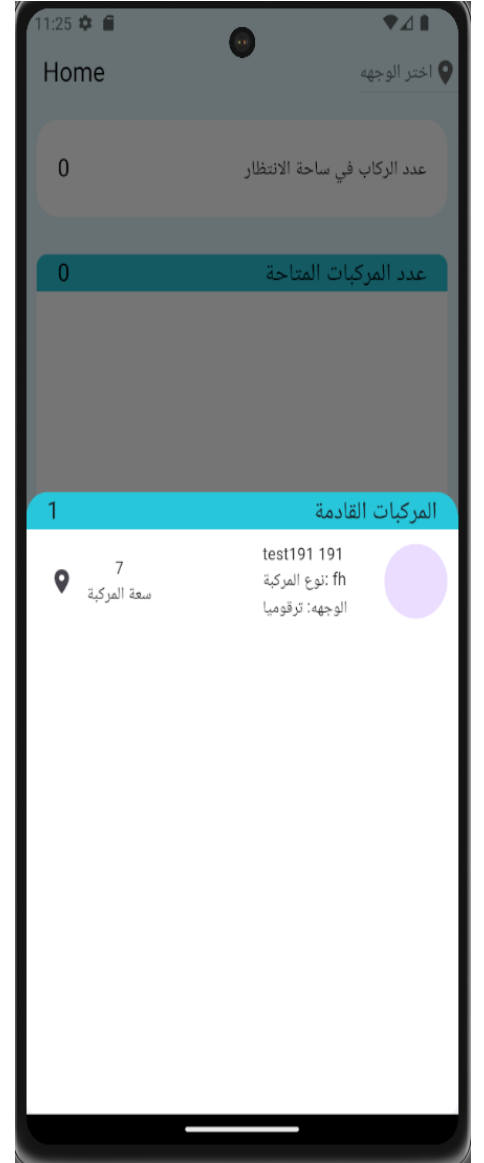
الشكل (3.8) واجهة تقديم شكوى



الشكل (3.13) السائقين للادمن



الشكل (3.12) مدراء الطوابق للادمن



الشكل (3.11) المركبات القادمة



الشكل (3.15) واجهة تغيير كلمة المرور



الشكل (3.14) واجهة إضافة مفقودات

## الفصل الرابع

### البناء والتنفيذ

4.1 مقدمة الفصل

4.2 لغات البرمجة والتقنيات

4.3 تشغيل النظام

4.4 التعليمات البرمجية

4.5 ملخص الفصل

## 4.1 مقدمة

في هذا الفصل، سنتعمق في مرحلة التنفيذ ضمن عملية تطوير النظام. هنا، سنركز على الجوانب العملية لتحويل التصميم والمتطلبات إلى نظام وظيفي. سيغطي هذا الفصل مواضيع مختلفة تتعلق بالتنفيذ، بما في ذلك لغات البرمجة، والأطر، والأدوات، والمنهجيات المستخدمة خلال عملية التطوير.

## 4.2 لغات البرمجة والتقنيات

في أثناء مرحلة التنفيذ، استخدمنا لغات البرمجة والتقنيات التالية:

- **Dart** : اخترنا لغة Dart لتطوير تطبيقنا؛ بسبب كفاءتها وأدائها العالي، وكذلك دعمها الممتاز لتطوير تطبيقات متعددة المنصات. مكنت لغة Dart من كتابة كود واحد يعمل بشكل ممتاز على كل من أنظمة التشغيل iOS و Android.
- **Flutter**: لتطوير الواجهة الأمامية لتطبيقنا، اخترنا إطار العمل Flutter ، وهو مكتبة مفتوحة المصدر تابعة لشركة Google. قدم لنا Flutter نهجًا معتمدًا على المكونات، مما جعل عملية تطوير وصيانة واجهة المستخدم لتطبيقنا أكثر سهولة وفاعلية.
- **Firestore**: استخدمنا Firestore كقاعدة بيانات ومنصة إدارة لتطبيقنا. وفرت Firestore مجموعة من الأدوات والخدمات مثل مصادقة المستخدمين، قاعدة بيانات لحظية، وخدمات التخزين السحابي، مما ساهم في تسهيل عملية تطوير التطبيق وتحسين أدائه.
- **JSON** : استخدمنا (JavaScript Object Notation) JSON كصيغة لتبادل البيانات بين مكونات الواجهة الأمامية والخلفية في تطبيقنا. قدمت JSON صيغة خفيفة وسهلة القراءة لتمثيل البيانات.
- **IDE** : استخدمنا Visual Studio Code و Android Studio

### نهج التنفيذ

في أثناء مرحلة التنفيذ للمشروع، اتبعنا نهجًا منهجيًا لإحياء تصميم النظام. كان هدفنا من هذا النهج ضمان تطوير نظام وظيفي وموثوق يلبي متطلبات المشروع. إليك نظرة عامة على نهج التنفيذ الذي اعتمدناه:

#### 1. اختيار التكنولوجيا:

**بيئة تشغيل الخلفية (Backend Runtime Environment)**: اخترنا لغة Dart كبيئة تشغيل خلفية. تُعرف Dart بأدائها وإنتاجيتها، مما يجعلها خيارًا مناسبًا لنظامنا. زيادة على ذلك، تدعم Dart التكامل مع Firebase ، مما يسمح لنا بالعمل بسلاسة مع خدمات Firebase.

هيكل الواجهة الأمامية (Frontend Framework) :استخدمنا Flutter لتطوير الواجهة الأمامية، وهو إطار عمل شهير لبناء تطبيقات مدمجة محليًا للهواتف المحمولة والويب وسطح المكتب من قاعدة شيفرة واحدة. قدم لنا Flutter نهجًا قائمًا على المكونات، مما سهل تطوير وصيانة الواجهة الأمامية لنظامنا.

قاعدة البيانات :اخترنا Firebase Firestone كقاعدة بيانات أساسية لنا. تُعرف Firebase Firestone بمرونتها وقدراتها على مزامنة البيانات في الوقت الفعلي. سمح لنا هذا الاختيار بإدارة وتخزين البيانات بكفاءة لتطبيقنا.

## 2. إعداد المشروع:

تجهيز بيئة العمل :أعدنا بيئة التطوير اللازمة لكل من الخلفية والواجهة الأمامية. تضمن ذلك تثبيت Dart و Flutter و Firebase والأدوات التابعة لها.

التحكم في الإصدار :استخدمنا نظام التحكم في الإصدار، مثل Git ، لتتبع وإدارة تغييرات الشيفرة المصدرية للمشروع.

## 3. تطوير خلفية المشروع (Backend Development)

المعمارية :اعتمدنا بنية طبقية للخلفية، حيث قسمناها إلى مكونات مختلفة مثل وحدات التحكم والنماذج والخدمات. تعزز هذه البنية فصل المكونات وتسهيل صيانة التطبيق.

برمجة الخلفية وتضمين Backend API :صممنا وطبقنا واجهات برمجة التطبيقات RESTful باستخدام Dart لتمكين التواصل بين مكونات الواجهة الأمامية والخلفية. حددت هذه "APIs" نقاط النهاية وصيغ البيانات لوظائف النظام المختلفة.

إضافة قاعدة البيانات :دمجنا Firebase Firestone في نظامنا لضمان تخزين واسترجاع البيانات بكفاءة. صممنا مخطط قاعدة البيانات، وطبقنا منطق الوصول إلى البيانات الضرورية في الخلفية لتتعامل بسهولة مع Firebase Firestore

## 4. تطوير الواجهة الأمامية (Frontend Development)

نهج قائم على المكونات :باستخدام Flutter ، طورنا الواجهة الأمامية باستخدام مكونات قابلة لإعادة الاستخدام والتعديل. سهل هذا النهج إعادة استخدام الكود وعرض واجهة المستخدم بكفاءة.

تصميم واجهة المستخدم :استخدمنا مجموعة واسعة من الأدوات المصممة مسبقًا في Flutter لتحسين تصميم وتجربة المستخدم للتطبيق حيث يقدم Flutter تصاميم مستجيبة وضمان سهولة الاستخدام.

## 5. الدمج والاختبار (Integration and Testing)

دمج الواجهة الأمامية والخلفية: قمنا بدمج مكونات الواجهة الأمامية والخلفية، وضمان التواصل السلس من خلال الـ APIs .  
شمل هذا الدمج اختبار نقاط النهاية والتعامل مع تبادل البيانات بين الطبقتين.

الاختبار وضمان الجودة: أجرينا العديد من أنشطة الاختبار، بما في ذلك اختبار الوحدات واختبار الدمج ، للتحقق من وظائف النظام وموثوقيته. استخدمنا أدوات مثل محاكي الاندرويد وهواتف ذكية لاختبار والتحقق من الـ APIs، وضمان دقتها ووظائفها الصحيحة.

## 6. الخاتمة:

في نهاية مرحلة التنفيذ، نجحنا في تحويل تصميم النظام إلى حل برمجي وظيفي. حققنا التكامل المطلوب بين الواجهة الأمامية والخلفية، وضمان التشغيل السلس للنظام.

## 4.3 تشغيل التطبيق

في هذا القسم، سنتناول كيفية تشغيل التطبيق بشكل كامل بدءًا من تحميل التطبيق حتى الاستخدام الفعلي له.

- تحميل التطبيق: يمكن تحميل التطبيق على شكل APK وتنصيبه على الأجهزة التي تعمل بنظام Android.
- التسجيل والدخول: عملية تسجيل الدخول مقصورة على المستخدمين وهم المسؤول والسائق ومدير الطابق وبعد تسجيل الدخول يتم تحويل كل مستخدم إلى صفحة المهام المخصصة له
- استخدام الميزات الأساسية: عند فتح التطبيق تفتح الصفحة الرئيسية التي يمكن من خلالها اختيار الواجهة وعرض المركبات المتاحة ومعرفة عدد الركاب

## 4.4 التعليمات البرمجية

في هذا القسم، سنعرض مقتطفات من التعليمات البرمجية الرئيسية المستخدمة في تطوير التطبيق.

```
Future<void> addDriver() async {
  if (formKey.currentState?.validate() ?? false) {
    Position position = await _determinePosition();
    final success = await remoteDataSource.addDriver(
      firstName: firstNameController.text,
      lastName: lastNameController.text,
      carType: carTypeController.text,
      carCapacity: carCapacityController.text,
      carLicense: carLicenseController.text,
      phone: phoneNumberController.text,
      email: emailController.text,
      password: passwordController.text,
      latitude: position.latitude,
      longitude: position.longitude,
    );

    if (success) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Driver added successfully')),
      );
      Navigator.pop(context); // Go back to the previous screen
    } else {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Error adding driver')),
      );
    }
  }
}
```

الشكل (4.1) Function Add Driver

```

getCurrentLocation(BuildContext context) async {
  bool serviceEnabled;
  LocationPermission permission;

  serviceEnabled = await Geolocator.isLocationServiceEnabled();
  if (!serviceEnabled) {
    await showDialog(
      context: context,
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('تنبيه'),
          content: Text('خدمات الموقع غير مفعله. لا يمكن استخدام التطبيق.'),
          actions: <Widget>[
            TextButton(
              child: Text('موافق'),
              onPressed: () {
                SystemNavigator.pop(); // ايقاف التطبيق
              },
            ), // TextButton
          ], // <Widget>[]
        ); // AlertDialog
      },
    );
    return;
  }
  permission = await Geolocator.checkPermission();
  if (permission == LocationPermission.denied) {
    permission = await Geolocator.requestPermission();
    if (permission == LocationPermission.denied) {
      await showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(

```

الشكل (4.2) جزء من كود طلب صلاحيات الموقع

## 4.5 ملخص الفصل

في هذا الفصل، تطرقنا إلى مرحلة التنفيذ في عملية تطوير النظام. غطينا جوانب مختلفة تتعلق بالتنفيذ العملي للنظام، بما في ذلك لغات البرمجة، والتقنيات، والمنهجيات المستخدمة خلال عملية التطوير.

بدأنا بمناقشة لغات البرمجة والتقنيات التي استخدمناها في مشروعنا. اخترنا لغة Dart كبيئة تشغيل الخلفية نظرًا لأدائها وإنتاجيتها. استخدمنا Flutter لتطوير الواجهة الأمامية، مما وفر لنا نهجًا قائمًا على المكونات لبناء واجهات المستخدم (APIs)، و JSON كصيغة لتبادل البيانات، و Visual Studio Code و Android studio كمحرر نصوص، و Flutter widgets لتحسين التصميم.

بعد ذلك، استعرضنا نهج التنفيذ الذي اتبعناه، والذي شمل اختيار التقنية، وإعداد المشروع، وتطوير الخلفية، وتطوير الواجهة الأمامية، والتكامل والاختبار. أبرزنا أهمية اختيار التقنيات المناسبة، وتكوين بيئة التطوير، وإعداد هيكل المشروع. كما ناقشنا الهيكل المعماري، والوظائف الرئيسية التي نُفِّذت في مراحل تطوير الخلفية والواجهة الأمامية.

بالإضافة إلى ذلك، غطينا مواصفات واجهات برمجة التطبيقات وكيفية توثيق نقاط النهاية المتاحة، وصيغ الطلب/الاستجابة، وآليات المصادقة، ومعالجة الأخطاء. ناقشنا أيضًا أهمية التصميم المستجيب لضمان وصول النظام وتجربة المستخدم المثلى عبر الأجهزة المختلفة.

أخيرًا، قدمنا نظرة عامة على الفصل، ملخصين النقاط الرئيسية التي نُوقِشت في كل قسم. أبرزنا أهمية تحويل تصميم النظام إلى حل برمجي وظيفي والدمج الناجح بين مكونات الواجهة الأمامية والخلفية. أسست مرحلة التنفيذ الأساس للاختبار اللاحق، والنشر، والتطويرات المستقبلية.

بشكل عام، ركز هذا الفصل على الجوانب العملية لتنفيذ النظام، مقدمًا رؤى حول لغات البرمجة، والتقنيات، ونهج التنفيذ، والتكامل، وتطوير واجهات برمجة التطبيقات، وتطوير الواجهة الأمامية.

## الفصل الخامس

### فحص النظام

5.1 مقدمة الفصل

5.2 الآلية المتبعة في فحص النظام

5.3 تقنيات التحقق ( Validation Techniques )

5.4 نظرة عامة على الفصل

## 5.1 مقدمة الفصل

في هذه المرحلة، يتم التأكد من عمل النظام بشكل صحيح وخالي من المشاكل، وذلك من خلال التحقق من المتطلبات الوظيفية وغير الوظيفية للمشروع. يتم أيضاً التأكد من أن النظام يعمل بكفاءة عالية في إتمام المهام، سواء من حيث الدقة أو السرعة. تأتي مرحلة فحص النظام بعد تنفيذه وتطبيقه، حيث يتم فحص كافة جوانب النظام لضمان تلبية جميع الاحتياجات والمتطلبات المحددة.

## 5.2 الآلية المتبعة في فحص النظام

عند القيام بفحص الأنظمة الإلكترونية، يجب اتباع خطوات معينة لضمان شمولية الفحص واكتماله لجميع أجزاء النظام. تشمل مراحل فحص النظام الأساسية الآتية:

- **فحص وحدات النظام:** يتم في هذه المرحلة فحص كل وحدة من وحدات النظام على انفراد لضمان عملها بشكل صحيح وفعال، وتأكيد توافقها مع المواصفات المطلوبة .

### مثال على اختبار الوحدة:

في مرحلة اختبار الوحدة، ركزنا على اختبار المكونات الفردية لنظامنا. على سبيل المثال، أجرينا اختبارات وحدة لوحة معينة مسؤولة عن مصادقة المستخدم. أنشأنا حالات اختبار للتحقق من وظائف الدوال والأساليب داخل هذه الوحدة، مثل تسجيل الدخول، وإعادة تعيين كلمة المرور. من خلال محاكاة سيناريوهات ومدخلات مختلفة، تمكنا من اكتشاف وإصلاح أي أخطاء أو مشاكل في مرحلة مبكرة، مما يضمن موثوقية وصحة وحدة المصادقة.

- **فحص دمج النظام (Frontend and backend integration):** يتم في هذه المرحلة اختبار تكامل وحدات

النظام معاً للتأكد من أنها تعمل بشكل سليم مع بعضها البعض وتنسجم بشكل لا يُثير الشكوك.

### مثال على اختبار الدمج:

خلال مرحلة اختبار الدمج، هدفنا إلى ضمان التفاعل السلس وتوافق المكونات المختلفة للنظام. على سبيل المثال، اختبرنا الدمج بين مكونات الواجهة الأمامية والخلفية لنظامنا. قمنا بمحاكاة تفاعلات مستخدم مختلفة على الواجهة الأمامية، مثل إرسال نموذج أو تقديم طلب، وتحققنا من أن البيانات تم نقلها بشكل صحيح إلى الواجهة الخلفية عبر (APIs). كما تحققنا من اتصال قاعدة البيانات وتأكدنا من أن البيانات تم تخزينها واسترجاعها بشكل صحيح. ساعدنا هذا الاختبار في اكتشاف أي مشاكل أو تناقضات في التواصل بين الواجهة الأمامية والخلفية، مما يضمن تبادل البيانات السلس ووظائف النظام.

اسم الوحدة	المدخلات	المخرجات المتوقعة من الفحص	المخرجات الفعلية للنظام
وحدة تسجيل الدخول	ايميل المستخدم / كلمة المرور	التأكد من هوية المستخدم	وصول المستخدم للنظام
وحدة اضافة سائق	بيانات السائق	اضافة سائق الى النظام	بعد اضافة سائق يستطيع السائق الوصول للنظام
وحدة اضافة مدير طابق	بيانات مدير الطابق	اضافة مدير طابق الى النظام	بعد اضافة مدير طابق يستطيع مدير الطابق الوصول للنظام
وحدة تحديث او حذف سائق / مدير طابق	--	تعديل بيانات او حذف المستخدمين	تعديل بيانات المستخدم وتخزينها في قاعدة البيانات , حذف مستخدم من قاعدة البيانات

جدول (5.1) Testing

تتم عملية إنشاء الحسابات بواسطة المسؤول عن النظام ، بما في ذلك حسابات السائقين ومديري الطوابق حيث يترتب عليه إدخال جميع الحقول المطلوبة كما هو موضح بالصورة التالية، وفي حال لم يتم إدخال البيانات في الحقول سيظهر له تنبيه وهذا مثال على فحص المدخلات "validation input"

11:37

← إضافة سائق

الاسم الأول  
الرجاء ادخال الاسم الاول

الاسم الأخير  
الرجاء ادخال الاسم الأخير

رقم الهاتف  
الرجاء ادخال رقم الهاتف

نوع المركبة  
الرجاء ادخال نوع المركبة

سعة المركبة  
الرجاء ادخال سعة المركبة

لوحة المركبة  
الرجاء ادخال رقم لوحة المركبة

بريد الكتروني  
الرجاء ادخال بريد الكتروني

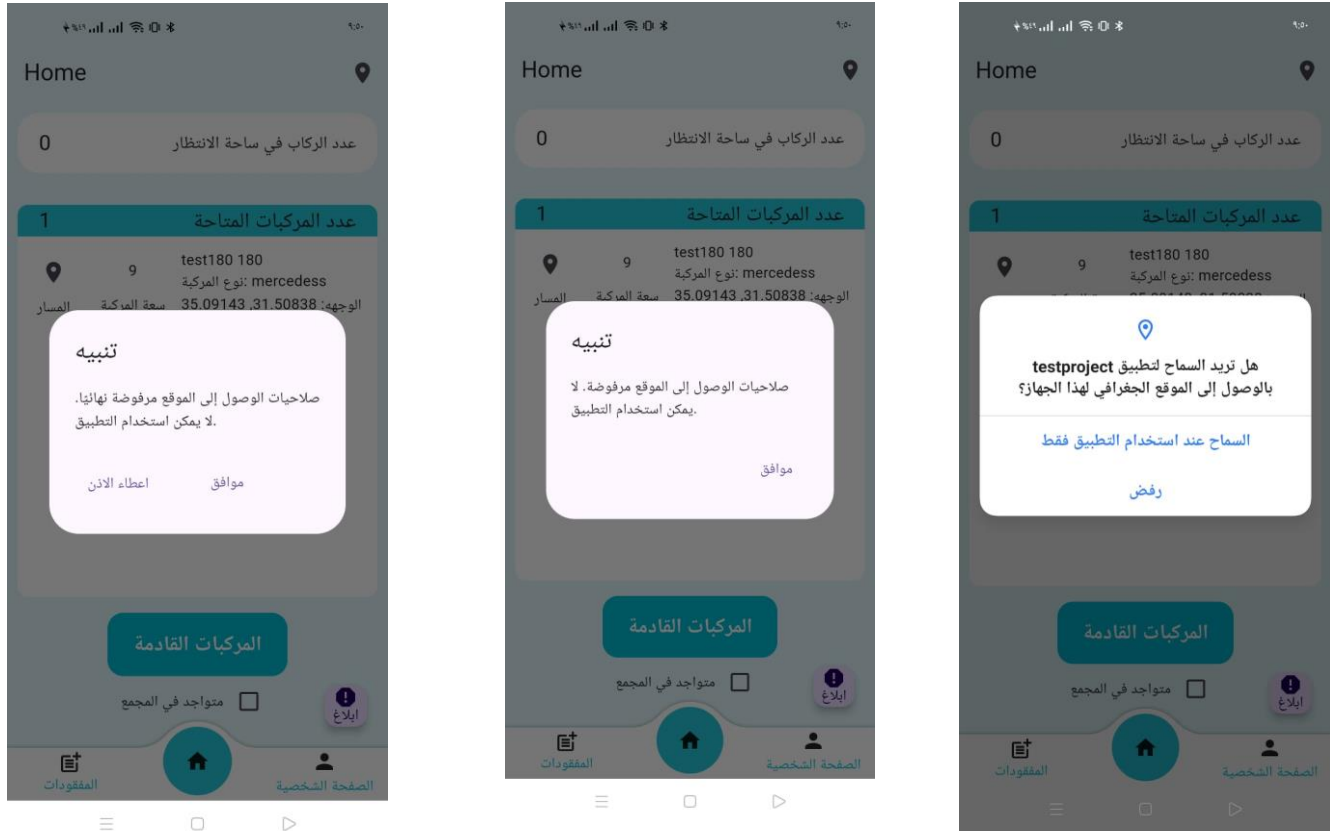
كلمة مرور  
الرجاء ادخال كلمة مرور

أختار وجهه  
الرجاء اختيار وجهه

إضافة سائق

الشكل (5.1) Input validation

يعتبر الموقع احد الأساسيات في التطبيق ولا يمكن عمل التطبيق بدونه لذا عند فتح التطبيق يطلب صلاحيات الوصول للموقع وفي حال الرفض تظهر رسالة تنبيه حتى يتم إعطاء الاذن من الاعدادات الخاصة بالجهاز



الشكل (5.2) طلب صلاحيات الموقع

### 5.3 تقنيات التحقق:

بالإضافة إلى الاختبار، استخدمنا تقنيات التحقق لضمان أن النظام يلبي الغرض المقصود ويلبي متطلبات المستخدمين. تضمنت هذه التقنيات:

#### **التحقق من المتطلبات:**

الهدف من التحقق من المتطلبات : هو مراجعة وتأكيد متطلبات النظام لضمان أنها كاملة ودقيقة ومتوافقة مع أهداف المشروع.

العملية:

- مراجعة المتطلبات: فحص المتطلبات الموثقة بدقة، لضمان وضوحها وإمكانية تحقيقها وملاءمتها لأهداف المشروع. مثال: أثناء مراجعة المتطلبات، قمنا بتقييم المتطلب الذي ينص على "يجب أن يسمح النظام للمستخدمين بالتسجيل باستخدام بريدهم الإلكتروني وكلمة المرور". هذا التحقق يؤكد توافقه مع رؤية المشروع لتمكين التسجيل عن طريق البريد الإلكتروني.

#### **تحقق من طرف المستخدم (User validation):**

الهدف: يشمل تحقق المستخدم إشراك المستخدمين النهائيين لجمع ملاحظاتهم والتأكد من أن النظام يلبي احتياجاتهم وتوقعاتهم.

العملية:

- اختبار قبول المستخدم ( User acceptance testing ) إشراك مجموعة ممثلة من المستخدمين النهائيين لاختبار النظام في بيئة محكمة، واتباع سيناريوهات محددة مسبقاً، وتقييم أدائه وقابليته للاستخدام ووظائفه.

- جلسات التغذية الراجعة للمستخدمين: إجراء جلسات تغذية راجعة منظمة مع المستخدمين النهائيين لجمع رؤيتهم واقتراحاتهم ومخاوفهم بشأن قابلية استخدام النظام

مثال: أثناء اختبار قبول المستخدم، يقوم المستخدمون النهائيون المحددون بأداء مهام داخل النظام، مثل تبادل حجز دور ، تأكيد التواجد ، تغيير كلمة المرور . تساعد ملاحظاتهم في تقييم سهولة الاستخدام ووضوح التعليمات والرضا العام عن العملية.

#### **التحقق من البيانات:**

الهدف: يضمن تحقق البيانات دقة البيانات تخزينها و تخزينها و معالجتها بواسطة التطبيق.

العملية:

- التوفيق بين البيانات: مقارنة البيانات في التطبيق مع مصادر البيانات الموثوقة أو السجلات الموجودة لتحديد التباينات أو التناقضات.

مثال: يتضمن تحقق البيانات التحقق من بيانات المستخدمين أثناء عملية تسجيل المستخدم. عندما يحاول المسؤول اضافة مستخدم جديد التسجيل، يتحقق النظام مما إذا كان المستخدم موجود بالفعل في قاعدة البيانات يقوم النظام بإشعار المسؤول ان المستخدم موجود بالفعل في النظام ولا حاجة لاضافته مجددا لتجنب التعارضات.

## 5.4 ملخص الفصل

ركز الفصل الخامس على المرحلة الحيوية من الاختبار والتحقق في عملية تطوير النظام. استخدمنا مجموعة من منهجيات الاختبار وتقنيات التحقق لضمان جودة النظام ووظائفه وموثوقيته. تناول الفصل مختلف المواضيع المتعلقة بالاختبار، بما في ذلك منهجيات الاختبار وتقنيات التحقق وأمثلة على تطبيقها في مشروعنا.

بدأنا بمناقشة منهجيات الاختبار المختلفة التي استخدمناها، بما في ذلك اختبار الوحدة، واختبار الدمج، واختبار النظام، واختبار الأداء، واختبار قبول المستخدم (UAT) كل منهجية تخدم هدفاً محدداً وساعدتنا على تقييم جوانب مختلفة من التطبيق. قدمنا أهدافاً واضحة وعمليات وأمثلة لكل منهجية اختبار، موضحين كيفية تطبيقها في مشروعنا.

بعد ذلك، استعرضنا تقنيات التحقق التي تضمنت أن التطبيق يحقق الغرض المقصود ويلبي متطلبات المستخدمين. ناقشنا تقنيات التحقق من المتطلبات، والتحقق من المستخدم، والتحقق من البيانات. لكل تقنية، حددنا الأهداف والعمليات وقدمنا أمثلة على كيفية تطبيقها في مشروعنا

## References

( 1 ) Flutter Course

<https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/>

( 2 ) Youtube Tutorial

[https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ&ab\\_channel=NetNinja](https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ&ab_channel=NetNinja)

( 3 ) <https://pub.dev/>

( 4 ) Android Doc

<https://developer.android.com/develop>

( 5 ) Firebase Doc

<https://firebase.google.com/docs>

( 6 ) Firebase auth doc

<https://firebase.google.com/docs/auth>

( 7 ) Firebase Api doc

<https://firebase.google.com/docs/reference>

( 8 ) Flutter & Firebase Tutorial

[https://www.youtube.com/watch?v=sfA3NWDBPZ4&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&ab\\_channel=NetNinja](https://www.youtube.com/watch?v=sfA3NWDBPZ4&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&ab_channel=NetNinja)

( 9 ) [Draw.io](https://draw.io)