رقم: .....

مذكرة تخرج مقدمة في

جامعة محمد بوضياف – المسيلة

كلية الرياضيات والإعلام الآلي

قسم الإعلام الآلي

من أجل الاستيفاء الجزئي لمتطلبات شهادة

ماستر في الإعلام الآلي

تخصص :

ذكاء الاعمال والتحسين

من طرف

ساعد زيد

عنوان المذكرة

منصة لتسيير العقارات

تحت إشراف الأستاذ

لكحل مفتاح

أعضاء لجنة المناقشة

لكحل مفتاح	جامعة المسيلة	**مشرفا**
شترة محمد	جامعة المسيلة	**رئيسا**
بونيف محمد	جامعة المسيلة	**ممتحنا**

19، جوان 2025

إهداء

إلى من كانت دعواتها سلاحي في كل الأوقات،
...إلى من سهرت وتعبت من أجلي دون كلل أو ملل
.إلى أمي، نبع الحنان ومصدر القوة

...إلى أبي العزيز، شفاه الله ، كنت وما زلت قدوتي ومصدر إلهامي
.إلى إخوتي، سندي في الحياة، من شاركوني كل المراحل بحب وصبر

إلى أصدقائي الأوفياء، من كانوا إلى جانبي في كل خطوة،
.من شجعوني وآمنوا بي حين شككت في نفسي

.أهديكم هذا العمل المتواضع، عربون وفاء وامتنان

تشكرات

.أتوجه بجزيل الشكر والامتنان إلى كل من ساهم في دعمي خلال فترة إنجاز هذا المشروع

أشكر أستاذي المشرف على توجيهاته القيمة وصبره الكريم،
.وأشكر أساتذتي وكل من ساعدني بكلمة، فكرة، أو دعم

كما لا أنسى كل من ساندني من قريب أو بعيد،
.لكم مني كل الاحترام والتقدير

الفهرس

إهداء 3

تشكرات 4

الفهرس 5

فهرس الاشكال والمخططات : 7

فهرس الصور : 8

المقدمة 10

الفصل الاول : دراسة الموجود 11

1.1.مقدمة 11

2.1. اهم المصطلحات 11

1.2.1. تعريف العقار 11

2.2.1. انواع العقار 11

3.2.1. السوق العقاري 11

4.2.1. القطاع العقاري 12

5.2.1. اهداف القطاع العقاري 12

6.2.1.المستخدمين للقطاع العقاري 12

7.2.1. الرقمنة 13

8.2.1. المنصة الرقمية 13

3.1 .القطاع العقاري في الجزائر 13

لمحة عن القطاع 13

احصائيات القطاع في الجزائر[6] 13

احصائيات السوق العقاري 14

توزيع الإعلانات حسب نوع البائع 14

الاستثمار العقاري 14

4.1. المشاكل المطروحة 14

5.1.الحلول المقترحة 15

6.1. التعريف بالمنصة 15

7.1. أهداف المنصة 16

8.1. الخاتمة 16

الفصل الثاني : الانجاز والتصميم 17

1.2. مقدمة 17

2.2. اهم المصطلحات 17

لغة النمذجة UML 17
Star UML 17
PlantUML 18
مميزات PlantUML : 18
3.2. مخططات UML :..... 18
1.3.2. مخطط الاستخدام Use Case Diagram..... 18
2.3.2. مخطط التسلسل Sequence diagrams : 24
3.3.2 مخطط الفصل Diagram de class : 38
4.2.الخاتمة 43
الفصل الثالث : التطوير..... 44
1.3. مقدمة..... 44
2.3. بيئة العمل..... 44
1.2.3.العتاد المستخدم 44
2.2.3. لبرمجيات والأدوات المستخدمة 44
3.3. واجهات التطبيق..... 47
1.3.3. واجهات المنصة :..... 47
2.3.3. واجهات لوحة التحكم (flutter app) :..... 52
4.3. قواعد البيانات : 59
1.4.3. قواعد البيانات NoSQL : 59
2.4.3 أنواع قواعد NoSQL : 59
3.4.3. سبب اختيار Firebase 60
4.4.3.من مخطط الصنف إلى نموذج Firebase :..... 61
5.4.3 قواعد الأمان في Firebase (Firebase Rules) : 62
5.3. خاتمة..... 64
الخاتمة..... 65
المراجع 66
ملخص..... 67
Abstract..... 67

فهرس الاشكال والمخططات :

شكل 1.1. اعمدة بيانية لاعلانات العقار في الجزائر 14

شكل 1.2: مخطط الاستخدام (Use Case Diagram) للمستخدم (User). 19

شكل 2.2: مخطط الاستخدام (Use Case Diagram) للمهندس (Engineer). 20

شكل 3.2: مخطط الاستخدام (Use Case Diagram) للمقاول (Contractor). 21

شكل 4.2: مخطط الاستخدام (Use Case Diagram) للوكالة العقارية (Real Estate Agency)..... 22

شكل 5.2: مخطط التسلسل (Sequence Diagram) لعملية طلب حساب. 25

شكل 6.2: مخطط التسلسل (Sequence Diagram) لعملية تسجيل الدخول والخروج..... 26

شكل 7.2: مخطط التسلسل (Sequence Diagram) لعملية إدارة العقار. 28

شكل 8.2: مخطط التسلسل (Sequence Diagram) لعملية طلب شراء العقار. 30

شكل 9.2: مخطط التسلسل (Sequence Diagram) لعملية إضافة، حذف، وترويج خبر. 32

شكل 10.2: مخطط التسلسل (Sequence Diagram) لعملية إرسال رسالة..... 33

شكل 11.2: مخطط التسلسل (Sequence Diagram) لعملية التعديل على الحساب..... 35

شكل 12.2: مخطط التسلسل (Sequence Diagram) لعملية المناقصة..... 37

شكل 13.2: مخطط الصنف (Class Diagram)..... 40

فهرس الصور :

صورة 1.3. الصفحة الرئيسية 47
صورة 2.3. طلب حساب 48
صورة 3.3. صفحة المشاريع 48
صورة 4.3. تفاصيل العقار 49
صورة 5.3. صفحة الاخبار 50
صورة 6.3. صفحة التواصل 50
صورة 7.3. صفحة الملف الشخصي 51
صورة 8.3. صفحة مشاريع المناقصة 51
صورة 9.3. تقديم العرض 52
صورة 10.3. صفحة تسجيل الدخول 52
صورة 11.3. لوحة التحكم 53
صورة 12.3. صفحة ادارة الممتلكات او العقارات 54
صورة 13.3. اضافة عقار 54
صورة 14.3. تعديل العقار 55
صورة 15.3. الاخبار 55
صورة 16.3. صفحة المدفوعات 56
صورة 17.3. الرسائل 56
صورة 18.3. الملف الشخصي 57
صورة 19.3. المناقصات 57
صورة 20.3. اضافة مشاريع المناقصة 58
صورة 21.3. عروض المناقصة 58
صورة 22.3. قاعدة بيانات firestore 62
صورة 23.3. قواعد الأمان (Firebase Rules) 63

المقدمة

شهد العالم خلال السنوات الأخيرة تحولًا كبيرًا نحو الرقمنة، واستعمال المنصات الرقمية والويب في مختلف القطاعات الاقتصادية والاجتماعية وغيرها. ومن بين هذه القطاعات التي بدأت تشهد تطورًا تدريجيًا يبرز القطاع العقاري، الذي لا يزال يعاني من الركود وسوء التسيير، بالإضافة إلى التأخر في اعتماد الرقمنة والمنصات الذكية.

فعلى الرغم من وجود آليات متعددة للتسويق الرقمي، ومنصات البيع والشراء العقارية، إلا أن القطاع لا يزال يعاني من ضعف التواجد الرقمي، خاصة في الجزائر، وذلك نتيجة غياب التنسيق وسوء التسيير بين مختلف الجهات الفاعلة، مثل المهندسين، والمقاولين، ووكالات الترقية العقارية، وغيرها من الأطراف ذات الصلة.

وفي ظل هذا الركود، وانتشار استخدام الإنترنت والهواتف الذكية والتصفح اليومي للويب، بات من الضروري البحث عن حلول رقمية للنهوض بالسوق العقارية، سواء في الجزائر أو عالميًا، ومواكبة التحول الرقمي المستمر.

تم تنظيم هذا العمل في ثلاثة فصول رئيسية، تناول كل منها جانبًا من مراحل إنجاز المشروع:

الفصل الأول: خُصص لتحليل الواقع الحالي للقطاع العقاري في الجزائر، واستعراض أبرز الأدوات والمنصات الرقمية المشابهة، مع التطرق إلى التقنيات المعتمدة فيها.

الفصل الثاني: تناولنا فيه مرحلة التصميم والنمذجة، حيث تم تحديد المتطلبات الأساسية للمنصة، وبناء هيكل قاعدة البيانات، وتصميم الواجهات باستخدام المخططات المناسبة من.UML

الفصل الثالث: تناول الجوانب التقنية لتطوير المنصة، مع استعراض الواجهات والوظائف، وشرح كيفية استخدام النظام من وجهة نظر المستخدم.

الفصل الاول : دراسة الموجود

1.1.مقدمة :

القطاع العقاري يعتبر من القطاعات الفعّالة اقتصاديًا واجتماعيًا، إذ يعتبر أحد أهم القطاعات المحركة لعجلة التنمية، والسير والتنظيم الحسن للمجتمع. رغم ذلك، ما زال يعاني من عدة مشاكل وعراقيل، بالأخص في الجزائر، ما يستدعي حلولًا رقمية ومتطورة لمواجهة هذه المشاكل.

في هذا الفصل سنتعرف على أهم المصطلحات الواجب معرفتها، كما سنتعرف على مشاكل القطاع العقاري، ونقدم حلولًا مقترحة لها، وسنتعرف على المنصة الخاصة بنا في تسيير العقارات والقطاع بشكل عام، وأهم أهدافها، وكيف تساهم في السوق والقطاع، وبعض المنافسين في هذا المجال.

2.1. اهم المصطلحات :

1.2.1. تعريف العقار:

عرفه المشرع الجزائري في المادة 683 ق م بأنه "كل شيئ مستقر بحيز وتابت فيه ولا يمكن له نقله دون تلف فهو عقار كل ما عدا ذلك من شيئ فهو منقول" [1]

2.2.1. انواع العقار :

عقارات سكنية: شقق، فيلات، قصور، شاليهات.

عقارات صناعية: مصانع، مخازن.

عقارات تجارية: مبانى إدارية، مراكز تجارية، محال ... إلخ.

عقارات زراعية: مزارع، بساتين.

عقارات ذات ملكية عامة تستخدم لأغراض محددة: أراضى حكومية، مساجد، كنائس، مستشفيات، مدارس ."[9]

3.2.1. السوق العقاري :

" يشير سوق العقارات إلى شراء وبيع العقارات، بما في ذلك الأراضي والمباني السكنية والتجارية والصناعية. ويشمل السوق ديناميكيات العرض والطلب، واتجاهات الأسعار، والظروف الاقتصادية العامة التي تؤثر على معاملات العقارات. ويراقب مستثمرو العقارات، والراغبون في الاستثمار، السوق عن كثب لاتخاذ قرارات مدروسة بشأن شراء أو بيع أو الاستثمار في العقارات." [2]

4.2.1. القطاع العقاري :

ومنه يمكننا القول عن القطاع العقاري " يُعد القطاع العقاري من أهم القطاعات الاقتصادية التي تلعب دورًا رئيسيًا في تحقيق التنمية المستدامة وتحفيز النمو الاقتصادي. فهو يشمل التطوير العمراني، والمشاريع السكنية والتجارية، والبنية التحتية، والاستثمارات العقارية التي تدعم توسع المدن وتعزيز جودة الحياة." [3]

5.2.1. اهداف القطاع العقاري :

"يلعب القطاع العقاري دورًا محوريًا في الاقتصاد، حيث يُسهم في تنويع مصادر الدخل ورفع كفاءة استخدام الأراضي. كما يوفر بيئة استثمارية قوية تسهم في جذب رؤوس الأموال المحلية والأجنبية. من ناحية أخرى، فإن تطوير المشاريع العقارية يعزز قطاع البناء والتشييد، ويدعم نمو قطاعات مرتبطة مثل التمويل العقاري، والتجارة، والخدمات".[3]

6.2.1.المستخدمين للقطاع العقاري :

الجهات الحكومية والمؤسسات العمومية :

الوكالة الوطنية لتحسين السكن وتطويره :(AADL) تُعنى بتوفير سكنات اجتماعية لفائدة الطبقة المتوسطة .

الشركة الوطنية للترقية العقارية :(ENPI) تنفذ مشاريع سكنية ترويجية .

وزارة السكن والعمران والمدينة: تشرف على تنفيذ السياسة العامة للسكن.

البلديات وأمانات المدن: تمنح التراخيص وتدير المخطط العمراني.

المطورون العقاريون :

شركات البناء والمقاولات: تشارك في تنفيذ المشاريع العقارية.

وكالات الترقية العقارية : صاحبة املاك مستقلة ومطورة للعقار (بناء و بيع ,,,الخ) ,

المستثمرون العقاريون: يشترون العقارات لأغراض الاستثمار.

المواطنون :

المشترون والمستأجرون.

المالكون الفرديون.

الوسطاء والمنصات :

الوكلاء العقاريون.

المنصات الرقمية العقارية (مواقع، تطبيقات).

الفئات التقنية والميدانية :

المهندس المعماري: يصمم المباني.

المهندس المدني: يشرف على الهيكل والبنية التحتية.

مهندس كهرباء/ميكانيك: لتجهيزات المبنى التقنية.

العمال المهنيون (البناؤون، الكهربائيون، النجارون...).

7.2.1. الرقمنة :

"الرقمنة هي تحويل شيء غير رقمي إلى تمثيل رقمي أو قطعة أثرية. يمكن للأنظمة المحوسبة بعد ذلك استخدامها في حالات استخدام مختلفة. ومن الأمثلة على ذلك في مجال التصنيع تحويل القياس من قراءة يدوية أو ميكانيكية إلى قراءة إلكترونية.

الرقمنة أساسية. إنها الصلة بين العالم المادي والبرمجيات. هذا ما نقوم به منذ ستينيات القرن الماضي. إنها تُمكّن جميع العمليات التي تُقدم قيمة للأعمال نظرًا للحاجة إلى بيانات قابلة للاستهلاك." [4]

8.2.1. المنصة الرقمية :

وهي عبارة "بنية تحتية عبر الإنترنت تعتمد على البرمجيات لتسهيل التفاعلات والمعاملات بين المستخدمين " [5]

3.1 .القطاع العقاري في الجزائر:

لمحة عن القطاع :

يعد القطاع العقاري في الجزائر من القطاعات المساهمة في الاقتصاد والتنمية، رغم ذلك يعاني تراجع كبير سواء من ناحية التسيير او الرقمنة او حتى السوق

احصائيات القطاع في الجزائر[6] :

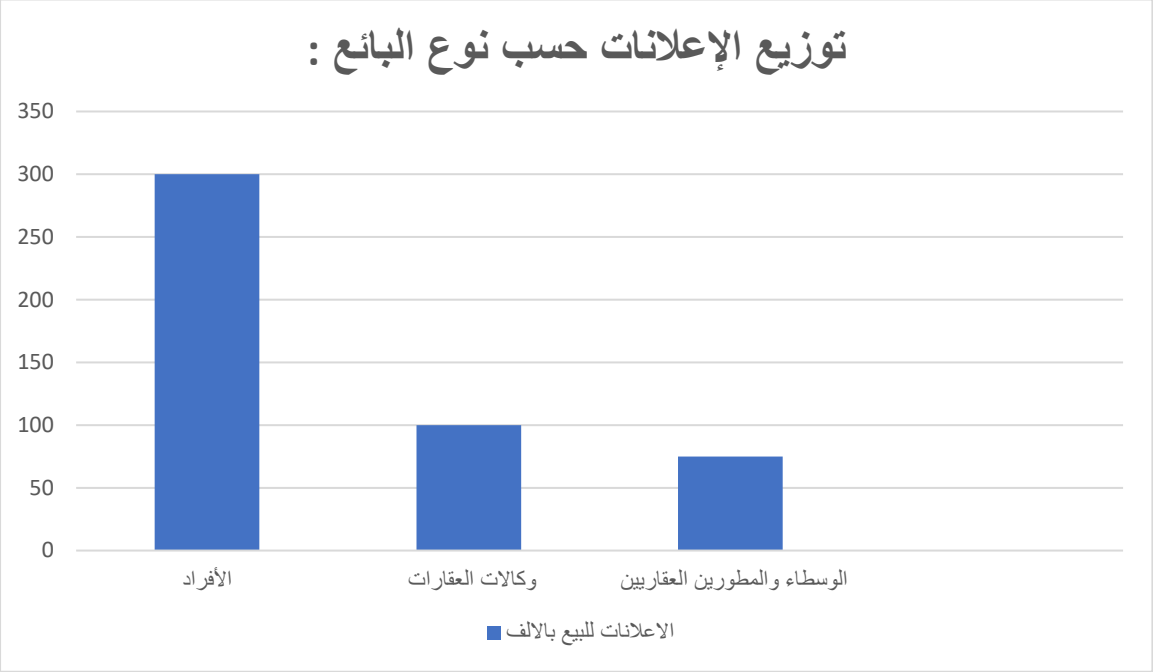
	العدد الاجمالي	سنة
الوكالات العقارية	حوالي 5500	2018 vitaminedz.com
المقاولون	حوالي 14,000	2024 news.radioalgerie.dz
المهندسون	اكثر من 652	2017 https://www.djazairess.com/
عدد المؤسسات في مجال البناء	حوالي 2480	2025 https://rentechdigital.com/
اليد العاملة في قطاع البناء	اكثر من 1.9 مليون	2022 echoroukonline.com

احصائيات السوق العقاري :

البيع : حوالي 300 الف اعلان

الايجار : 200 ألف إعلان.

توزيع الإعلانات حسب نوع البائع [7] **:**



شكل 1.1. اعمدة بيانية لاعلانات العقار في الجزائر

الاستثمار العقاري:

"الوكالة الجزائرية لترقية الاستثمار (AAPI) - (أغسطس 2024): المدير العام للوكالة يكشف عن تسجيل حوالي 9 آلاف مشروع بقيمة 400 ألف مليار دينار خلال عامين (بشكل عام، وليس فقط عقاري)."[8]

4.1. المشاكل المطروحة :

من خلال الدراسات الميدانية التي تم إجراؤها خلال التربص الميداني في وكالة الترقية العقارية Eural Uertec، وجدنا أن القطاع العقاري في الجزائر يعاني من عدة تعقيدات ومشاكل، بالأخص من جانب الرقمنة واستعمال التكنولوجيا.

من أبرز هذه المشاكل:

- ضعف التواصل بين جهات القطاع العقاري (بائعين، مشترين، وكالات عقارية، وغيرها).
- مشاكل في العثور على عقار مناسب للبيع، أو الاستئجار، أو حتى للإنشاء.
- مشاكل بيروقراطية، خاصة بالنسبة للوكالات والمهندسين والمقاولين، في إجراءات التسجيل.
- ضعف كبير لدى الوكالات في العثور على عمّال، ومهندسين، ومقاولين للمشاريع.
- عدم وجود قاعدة بيانات للوكالات ومختلف الأطراف الفعّالة في القطاع، للبحث والتواصل فيما بينهم.
- صعوبة في إنشاء المشاريع والمناقصات، وإيجاد العروض المناسبة بالنسبة للوكالات والمقاولين وغيرهم.
- ضعف في الشفافية، وفي استخدام المنصات الرقمية والتكنولوجيا في التسيير، والتسويق، والتواصل، والبيع، والكراء، خصوصًا لدى الوكالات.

5.1.الحلول المقترحة :

إنشاء منصة رقمية أو موقع ويب لأجل رقمنة القطاع، والتواصل بين مستخدميه، لمحاولة التقليل من المشاكل التي يواجهها، مع إضافة جميع الخصائص والمميزات المساعدة للوكالات، المهندسين، المقاولين، وأيضًا المشترين والمستأجرين.

6.1. التعريف بالمنصة :

منصة عقارية ويب رقمية متكاملة تجمع بين عمليات بيع، شراء، وتأجير العقارات، مع خدمات تواصل فعالة. بين المستخدمين تهدف إلى تسهيل المعاملات العقارية وتقديم تجربة فريدة للمستخدمين والمهنيين .

أهم الميزات :

- متجر إلكتروني للعقارات: عرض العقارات مع صور توضيحية، مع دعم تصنيفات بحث متقدمة (الموقع، السعر، عدد الغرف).
- ملفات شخصية للمستخدمين :متوفرة لكل من وكالات الترقية العقارية، المهندسين، والمقاولين، لتخصيص تجربة الاستخدام وإدارة القوائم بفعالية.
- نظام تواصل مدمج: يتضمن دردشة فورية، ومشاركة الملفات بين المستخدمين والمهنيين.
- رفع مشاريع المناقصات: يمكن للوكالات العقارية رفع مشاريع للمناقصة، وتقديم العروض من طرف المهندسين والمقاولين.
- نشر الأخبار العقارية: إمكانية رفع الأخبار من طرف الوكالات والمقاولين وغيرهم، للاطلاع الدائم على مستجدات السوق.

7.1. أهداف المنصة :

- تحسين فعالية السوق العقاري وعملياته من بيع، وشراء، ومناقصات.
- تعزيز الشفافية، وتسهيل الوصول إلى المعلومات الموثوقة.
- تمكين المستخدمين من اتخاذ قرارات عقارية مدروسة.
- تسهيل الوصول إلى المهندس، والمقاول، والوكالة عبر ملفاتهم الشخصية، وتعزيز التواصل بينهم.
- دمج التكنولوجيا الحديثة في القطاع العقاري.

8.1. الخاتمة :

في الختام، استعرضنا في هذا الفصل أهم المصطلحات الخاصة بالقطاع العقاري وأهميته، مع ذكر المشاكل التي يعاني منها، وأبرز إحصائياته في الجزائر. كما قدّمنا تصورًا لحل هذه المشاكل من خلال منصتنا الإلكترونية، وتطرقنا إلى تعريفها وأهم ميزاتها.

الفصل الثاني : الانجاز والتصميم

1.2. مقدمة :

"تُعد النمذجة جزءًا أساسيًا من مشاريع البرمجيات الكبيرة، وهي مفيدة أيضًا للمشاريع المتوسطة وحتى الصغيرة. يلعب النموذج في تطوير البرمجيات دورًا مشابهًا لدور المخططات والمخططات الأخرى (مثل خرائط المواقع، والواجهات، والنماذج المادية) في بناء ناطحة سحاب. باستخدام نموذج، يمكن للمسؤولين عن نجاح مشروع تطوير برمجيات التأكد من اكتمال وظائف العمل ودقتها، وتلبية احتياجات المستخدم النهائي، ودعم تصميم البرنامج لمتطلبات قابلية التوسع، والمتانة، والأمان، وقابلية التوسع، وغيرها من الخصائص، قبل أن يجعل التنفيذ في الكود التغييرات صعبة ومكلفة." [10]

في هذا الفصل، سوف نُعرّف بلغة النمذجة UML، كما سنتعرف على البيئات المستخدمة في نمذجة المخططات (Diagrams)، ونُقدّم المخططات اللازمة لشرح المنصة، بالإضافة إلى هيكلة قاعدة البيانات وشرح العمليات الخاصة بالمنصة.

2.2. اهم المصطلحات :

لغة النمذجة UML :

"تساعدك لغة النمذجة الموحدة (UML®) من OMG على تحديد نماذج أنظمة البرمجيات وتصورها وتوثيقها، بما في ذلك بنيتها وتصميمها، بطريقة تلبي جميع هذه المتطلبات. (يمكنك استخدام UML لنمذجة الأعمال ونمذجة أنظمة أخرى غير برمجية أيضًا.) باستخدام أيٍّ من الأدوات العديدة القائمة على UML في السوق، يمكنك تحليل متطلبات تطبيقك المستقبلي وتصميم حلٍّ يلبيها، مع تمثيل النتائج باستخدام أنواع المخططات القياسية الثلاثة عشر في UML 2.0.

يمكنك نمذجة أي نوع من التطبيقات تقريبًا، سواءً كان يعمل على أي نوع أو مجموعة من الأجهزة أو أنظمة التشغيل أو لغات البرمجة أو الشبكات، باستخدام UML. تتيح لك مرونتها نمذجة التطبيقات الموزعة التي تستخدم أي برنامج وسيط متوفر تقريبًا. بُنيت UML على مفاهيم OO الأساسية، بما في ذلك الصنف والعملية، وهي مناسبة تمامًا للغات وبيئات البرمجة كائنية التوجه مثل C++ وJava وC# الحديثة، ولكن يمكنك استخدامها أيضًا لنمذجة التطبيقات غير كائنية التوجه في، على سبيل المثال، Fortran وVB وCOBOL. تساعدك ملفات تعريف UML (أي مجموعات فرعية من UML مصممة لأغراض محددة) على نمذجة الأنظمة التفاعلية والأنظمة في الوقت الفعلي والأنظمة المقاومة للأخطاء بطريقة طبيعية."[10]

Star UML :

"هي منصة نمذجة برمجية تدعم لغة النمذجة الموحدة (UML). تعتمد على الإصدار 1.4 من UML، وتوفر أحد عشر نوعًا مختلفًا من المخططات، وتتوافق مع ترميز UML 2.0. تدعم المنصة بفعالية نهج MDA (الهندسة المعمارية الموجهة بالنموذج) من خلال دعم مفهوم ملف تعريف UML. تتميز StarUML™ بسهولة تخصيصها لبيئة المستخدم، كما تتميز بإمكانية توسعة وظائفها. يضمن استخدام StarUML™، إحدى أفضل أدوات نمذجة البرمجيات الرائدة، تحقيق أقصى قدر من الإنتاجية والجودة في مشاريعك البرمجية."[11]

PlantUML :

" هي أداة متعددة الاستخدامات تُسهّل إنشاء مجموعة واسعة من المخططات بسرعة وسهولة. باستخدام لغة بسيطة وبديهية، يُمكن للمستخدمين رسم أنواع مختلفة من المخططات بسهولة. للاطلاع على تفاصيل إمكانيات اللغة وقواعدها النحوية،" [12]

كيف تعمل PlantUML :

تكتب النصّ التوصيفي بلغة PlantUML مثلاً @startuml, Alice -> Bob : Hello :

تقوم الأداة بتحليل هذا النص وتحويله إلى رسم بياني أو مخطط UML تلقائيًا.

مميزات PlantUML :

_مجانية ومفتوحة المصدر.(Apache License 2.0)

_تدعم عددًا كبيرًا من أنواع المخططات:

مخططات التسلسل(Sequence diagrams) ،مخططات الاستخدام(Use Case diagrams) ،المخططات الفصلية (Class diagrams)،مخططات الأنشطة(Activity diagrams) ،مخططات الحالة، المكونات، النشر، التوقيت، خرائط التفكير، Gantt، WBS، واجهات رسومية، والمزيد.

_سهلة التعلم والاستخدام.

3.2. مخططات UML :

1.3.2. مخطط الاستخدام Use Case Diagram :

التعريف :

"الغرض من مخطط حالة الاستخدام في لغة النمذجة الموحدة (UML) هو توضيح الطرق المختلفة التي قد يتفاعل بها المستخدم مع النظام. "[13]

مميزاته :

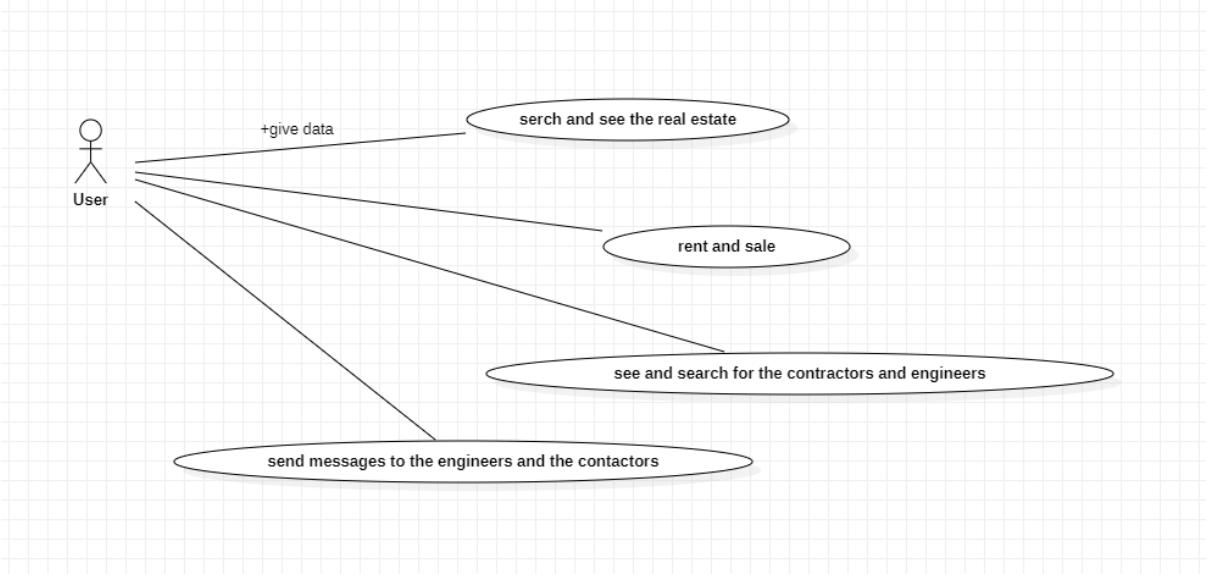
"في لغة النمذجة الموحدة (UML)، يُمكن لمخطط حالة الاستخدام أن يُلخص تفاصيل مستخدمي نظامك (المعروفين أيضًا بالجهات الفاعلة) وتفاعلاتهم مع النظام. لإنشاء مخطط حالة استخدام، ستستخدم مجموعة من الرموز والموصلات المتخصصة. يُمكن لمخطط حالة الاستخدام الفعّال أن يُساعد فريقك على مناقشة وتمثيل ما يلي:

_السيناريوهات التي يتفاعل فيها نظامك أو تطبيقك مع الأشخاص أو المؤسسات أو الأنظمة الخارجية

_الأهداف التي يُساعد نظامك أو تطبيقك هذه الجهات (المعروفة بالجهات الفاعلة) على تحقيقها

نطاق نظامك"[13]

مخططات USE case للمنصة :



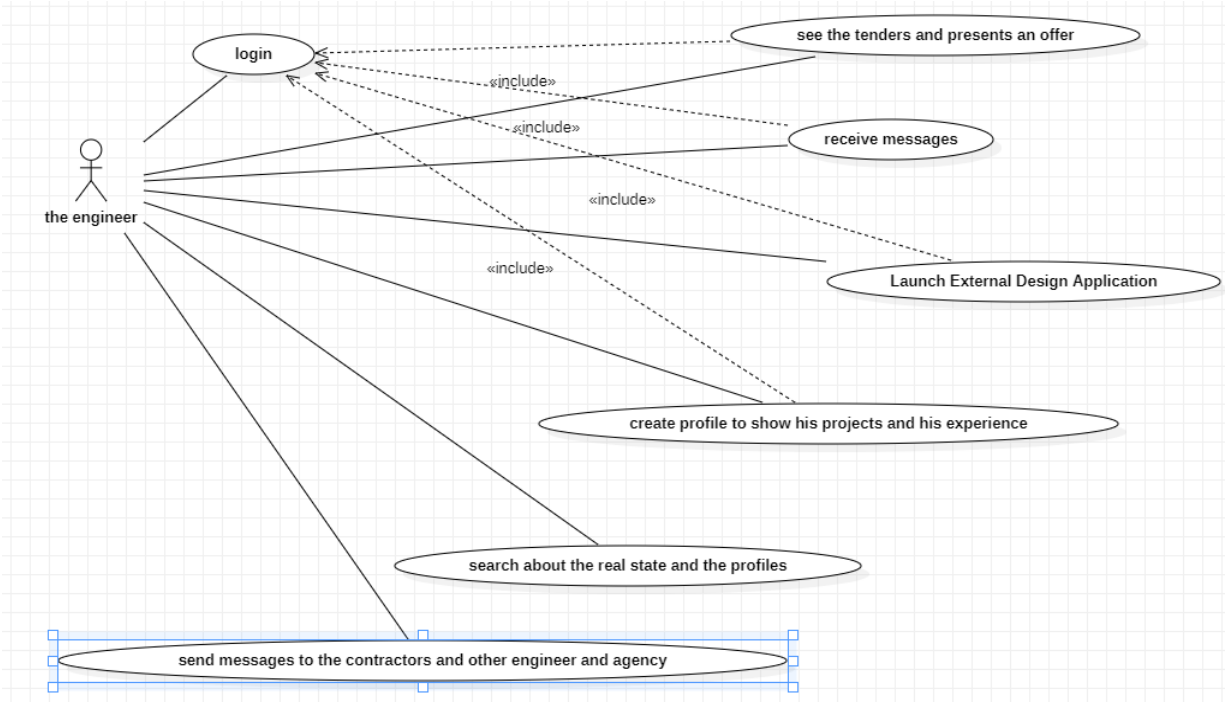
شكل 1.2: مخطط الاستخدام (Use Case Diagram) للمستخدم (User).

شكل 1.2: مخطط الاستخدام (Use Case Diagram) للمستخدم(User)

يعرض هذا المخطط الوظائف الأساسية التي يمكن للمستخدم تنفيذها داخل المنصة بعد تسجيل الدخول. تتضمن هذه الوظائف ما يلي:

- البحث عن العقارات :(Search for real estate) استعراض العقارات المتاحة للبيع أو الإيجار.
- الشراء أو الكراء :(Buy, rent) تنفيذ العمليات العقارية داخل المنصة.
- البحث عن المهنيين :(Search for engineers and contractors) استكشاف ملفات المهندسين والمقاولين والتواصل معهم.
- إرسال الرسائل :(Send messages) تمكين المستخدم من التواصل المباشر مع المهنيين المسجلين في المنصة.

يعكس المخطط بشكل شامل التفاعل المباشر بين المستخدم والمنصة، ويبرز الأدوار الرئيسية التي يمكنه القيام بها، ما يسهم في تحسين تجربة الاستخدام الرقمية.



شكل 2.2: مخطط الاستخدام (Use Case Diagram) للمهندس (Engineer).

شكل 2.2: مخطط الاستخدام (Use Case Diagram) للمهندس(Engineer)

يعرض هذا المخطط مختلف الوظائف التي يُمكن للمهندس القيام بها داخل المنصة، سواء من خلال الواجهة العامة أو من خلال لوحة التحكم المخصصة له. يوضح المخطط التفاعل المباشر بين المهندس والنظام، ويبرز أهم العمليات المهنية التي تتيح له التفاعل الفعّال مع بقية المستخدمين والمنصة .تشمل هذه العمليات:

تسجيل الدخول :(Login) للوصول إلى الحساب الشخصي والوظائف المرتبطة به.

عرض المناقصات وتقديم العروض :(View tenders and submit offers) يتيح النظام للمهندس المشاركة في المشاريع المفتوحة من طرف الوكالات.

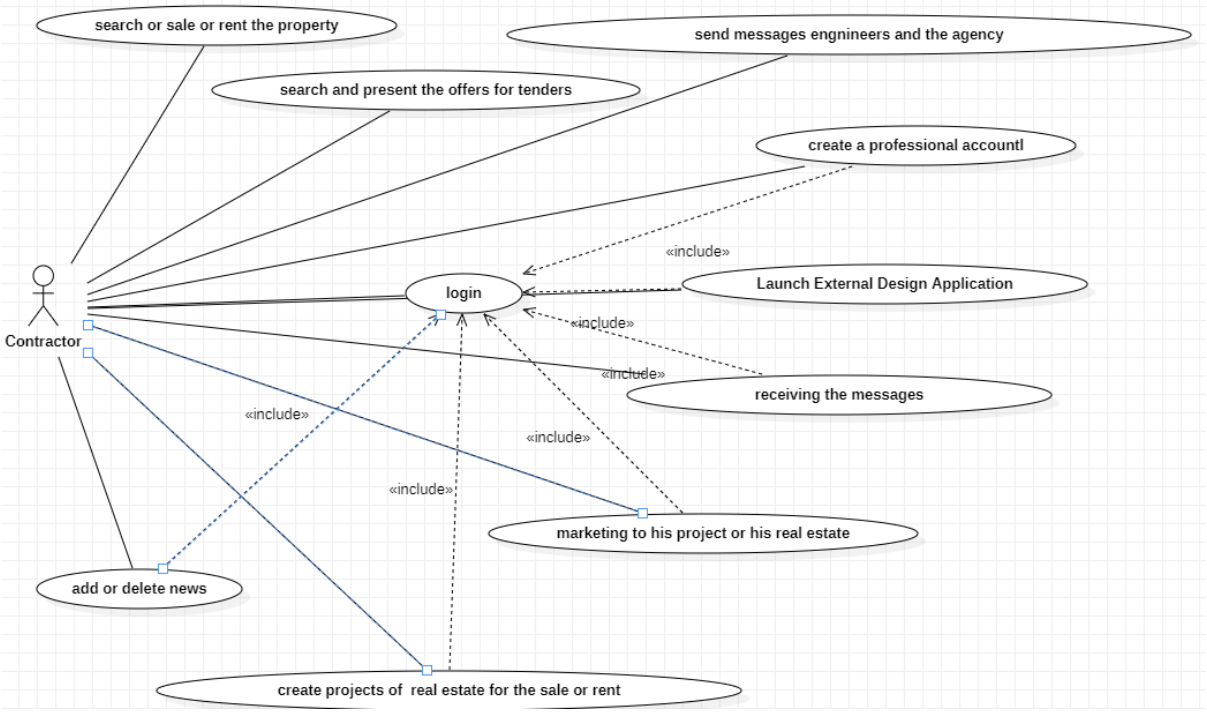
استقبال الرسائل :(Receive messages) لاستلام استفسارات أو عروض من مستخدمين آخرين.

استخدام أداة تصميم خارجية :(Use external design tool) الربط معHomestyler لإنشاء مخططات المشاريع.

إنشاء ملف شخصي :(Create profile) لعرض خبراته ومشاريعه السابقة.

البحث عن العقارات والملفات الشخصية :(Search real estate and user profiles) للوصول إلى المعلومات اللازمة في إطار مهامه.

إرسال الرسائل (Send messages): للتواصل مع المقاولين، المهندسين، أو الوكالات العقارية.



شكل 3.2: مخطط الاستخدام (Use Case Diagram) للمقاول (Contractor).

شكل 3.2: مخطط الاستخدام (Use Case Diagram) للمقاول(Contractor)

يعرض هذا المخطط المهام الرئيسية التي يمكن للمقاول تنفيذها داخل المنصة، سواء من خلال الواجهة العامة أو من خلال لوحة التحكم الخاصة به بعد تسجيل الدخول. يوضّح المخطط التفاعل الديناميكي بين المقاول والنظام، ويُبرز الوظائف التي تُمكّنه من أداء دوره بفعالية ضمن البيئة الرقمية للعقار والمناقصات.

تشمل هذه المهام:

تسجيل الدخول :(Login) للوصول إلى حسابه وإدارة محتوى حسابه وعملياته داخل النظام.

البحث عن العقارات، المناقصات، وحسابات المستخدمين الآخرين: لاستكشاف فرص العمل أو الاستثمار، أو للتعاون مع أطراف أخرى داخل النظام.

إنشاء المخططات وحفظها باستخدام منصة خارجية :Homestyler وذلك بعد تسجيل الدخول إلى تلك الأداة.

إرسال عروض للمناقصات :(Submit tenders) يمكن للمقاول الاطلاع على المشاريع المفتوحة وتقديم عروضه رسميًا.

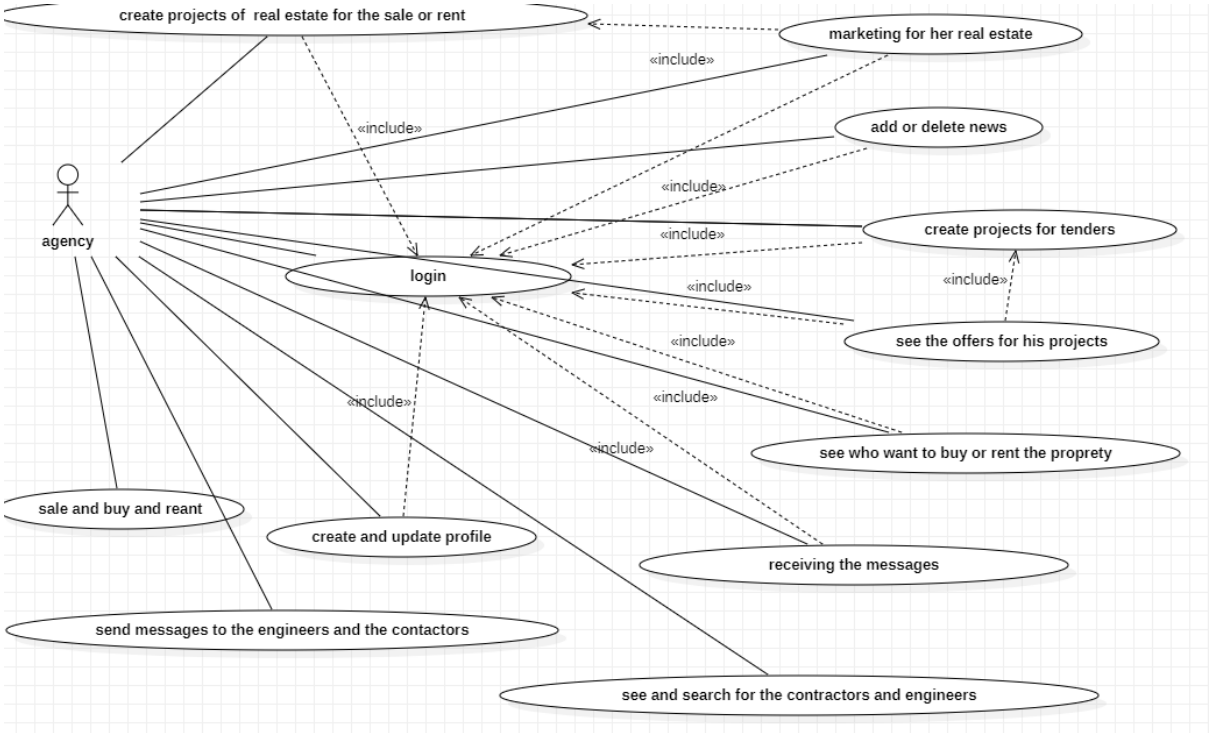
تعديل الملف الشخصي ورفع المشاريع الخاصة به: لتقديم نبذة مهنية محدثة وموثوقة للعملاء المحتملين.

رفع العقارات للبيع أو الكراء، وتعديلها أو حذفها: تُتيح المنصة للمقاول إدخال بيانات المشاريع التي يملكها أو يديرها، وتحديثها حسب الحاجة.

رؤية العروض التي يُرسلها المستخدمون بخصوص عقاراته، والتواصل معهم: لتسهيل عمليات التفاوض والإجابة على الاستفسارات مباشرة عبر المنصة.

نشر الأخبار المهنية الخاصة به أو بمشاريعه داخل المنصة، مع إمكانية حذفها: كوسيلة للتواصل المستمر مع السوق.

إرسال الرسائل إلى المستخدمين الآخرين داخل المنصة: لتيسير التواصل المهني مع الوكالات، المهندسين.



شكل 4.2: مخطط الاستخدام (Use Case Diagram) للوكالة العقارية (Real Estate Agency).

شكل 4.2: مخطط الاستخدام (Use Case Diagram) للوكالة العقارية(Real Estate Agency)

يعرض هذا المخطط مختلف العمليات التي يمكن للوكالة العقارية تنفيذها داخل المنصة الرقمية، سواء عبر الواجهة العامة أو من خلال لوحة التحكم الخاصة بها. وتُمثّل هذه العمليات مجمل الوظائف التي تساعد الوكالة على إدارة أنشطتها التجارية، والتفاعل مع باقي المستخدمين.

تشمل المهام الأساسية للوكالة:

- تسجيل الدخول:(Login) يُعد شرطًا أساسيًا للوصول إلى النظام وتنفيذ مختلف المهام، ويُستخدم كمدخل إلزامي لجميع حالات الاستخدام الأخرى عبر علاقة.<<include>>
- إنشاء وتحديث الملف الشخصي:(Create and update profile) تتيح هذه الوظيفة للوكالة إنشاء حساب رسمي خاص بها داخل المنصة، وتحديث بياناتها أو معلوماتها المهنية حسب الحاجة.
- إنشاء مشاريع عقارية للبيع أو الإيجار:(Create projects of real estate for the sale or rent) تمثل الوظيفة الأساسية لإدراج عقارات جديدة في النظام بغرض عرضها للبيع أو الكراء، وتُعد من أبرز العمليات التي تقوم بها الوكالة.
- التسويق للعقارات:(Marketing for her real estate) تستخدم لترويج العقارات المنشورة والوصول بها إلى شريحة أوسع من المستخدمين، مما يُسهم في تحسين فرص البيع أو الإيجار.
- إدارة الأخبار:(Add or delete news) تسمح للوكالة بنشر الأخبار أو حذفها، سواء كانت تتعلق بمشاريعها أو بعروض خاصة، مما يعزز التفاعل داخل المنصة.
- إنشاء مشاريع مناقصات:(Create projects for tenders) تُتيح للوكالة فتح مناقصات مهنية، بهدف تلقي عروض من مهندسين أو مقاولين لتنفيذ مشاريع بناء أو ترميم.
- عرض العروض على المشاريع:(See the offers for his projects) تمكن الوكالة من الاطلاع على العروض المُقدّمة من طرف المهنيين فيما يخص المناقصات التي تم طرحها.
- الاطلاع على الأشخاص المهتمين بشراء أو استئجار العقارات (See who want to buy or rent the property):تعرض للوكالة استفسارات وطلبات العملاء المحتملين المهتمين بالعقارات المنشورة، وتساعد على تقييم الطلب.
- استقبال الرسائل:(Receiving the messages) تُتيح للوكالة استقبال الرسائل من مستخدمين آخرين ضمن النظام، مثل المهنيين أو العملاء.
- إرسال رسائل إلى المهندسين والمقاولين:(Send messages to the engineers and the contractors) وظيفة تواصل مباشر تُستخدم لتنسيق المشاريع مع المهنيين، سواء ضمن إطار مناقصة أو مشروع قائم.
- البحث عن المهندسين والمقاولين:(See and search for the contractors and engineers) توفر للوكالة إمكانية البحث في قاعدة بيانات المهنيين المسجلين، والتفاعل معهم حسب الحاجة.
- البيع، الشراء، والإيجار:(Sale and buy and rent) تمثل هذه الحالة جميع المعاملات التجارية التي يمكن للوكالة إدارتها داخل النظام، مثل إتمام صفقات بيع أو تأجير العقارات.

2.3.2. مخطط التسلسل Sequence diagrams :

التعريف :

"تُعدّ مخططات التسلسل حلاً شائعًا للنمذجة الديناميكية في لغة النمذجة الموحدة (UML)، إذ تُركّز تحديدًا على خطوط الحياة، أي العمليات والكائنات التي تعمل في آنٍ واحد، والرسائل المتبادلة بينها لأداء وظيفة قبل انتهاء خط الحياة. "[13]

مميزاته :

"تُعدّ مخططات التسلسل مراجع مفيدة للشركات والمؤسسات الأخرى. جرّب رسم مخطط تسلسلي من أجل:

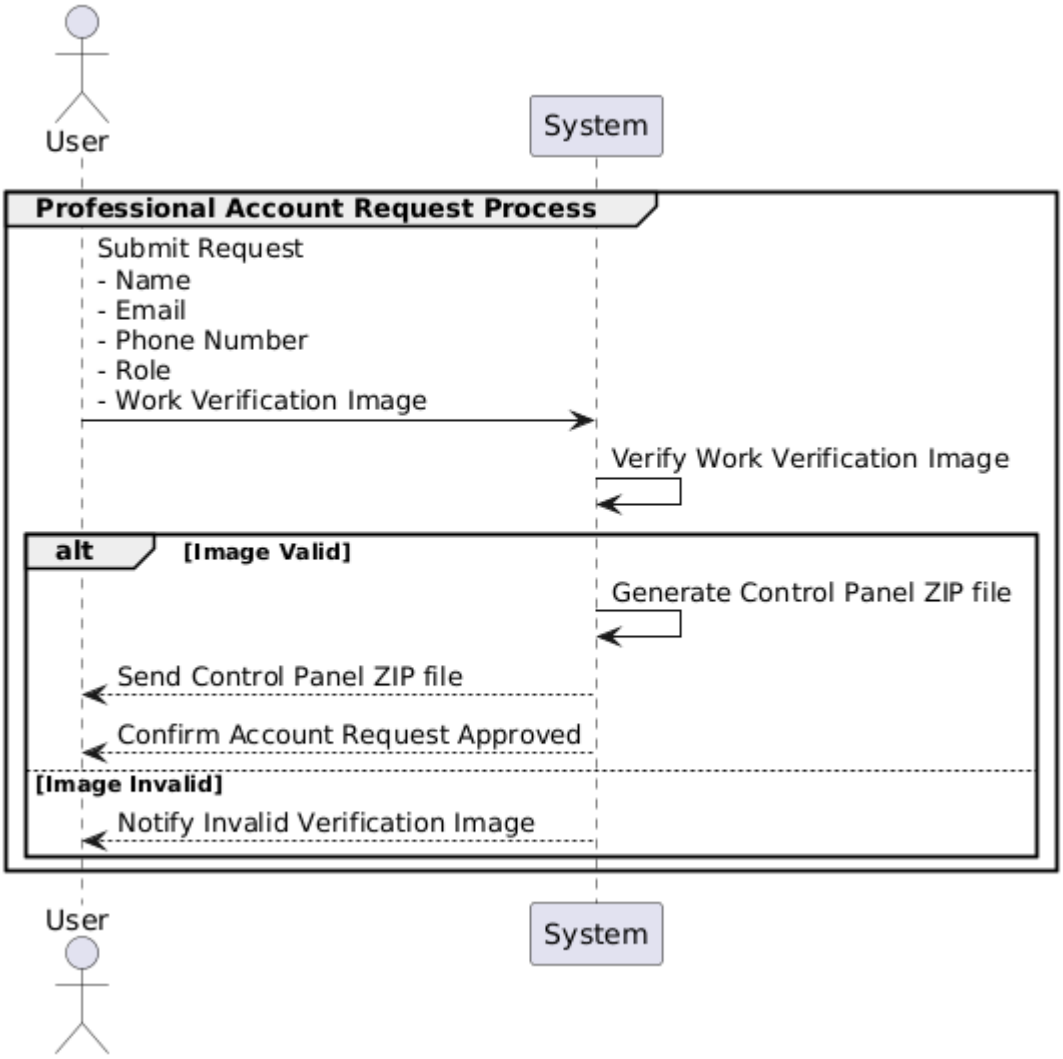
_تمثيل تفاصيل حالة استخدام UML.

_نمذجة منطق إجراء أو وظيفة أو عملية معقدة.

_رؤية كيفية تفاعل الكائنات والمكونات مع بعضها البعض لإكمال عملية.

_تخطيط وفهم الوظائف التفصيلية لسيناريو حالي أو مستقبلي. "[13]

مخططات SEQUENCE للمنصة :



شكل 5.2: مخطط التسلسل (Sequence Diagram) لعملية طلب حساب.

يُبرز هذا المخطط التفاعل المتسلسل بين المستخدم والنظام أثناء طلب إنشاء حساب مهني داخل المنصة، مثل حساب مهندس أو مقاول. تمثل هذه العملية خطوة محورية لضمان مصداقية المهنيين قبل منحهم صلاحيات خاصة عبر لوحة تحكم مخصصة. يتضمن هذا التفاعل المراحل التالية:

1 .إرسال طلب الحساب المهني:

يبدأ المستخدم بتعبئة نموذج مخصص يتضمن البيانات التالية:

الاسم الكامل،البريد الإلكتروني،رقم الهاتف،الدور المهني (مهندس/مقاول/وكالة)،صورة أو وثيقة إثبات المهنة (Work Verification Image)

يُرسل النموذج كاملًا إلى النظام لمعالجة الطلب.

2 .التحقق من وثيقة إثبات العمل:

يقوم النظام داخليًا بتحليل وتقييم صورة إثبات المهنة للتحقق من صحتها ومطابقتها للمعايير المطلوبة.

3 .التفرع وفق صلاحية الوثيقة:

في حال كانت الوثيقة صحيحة:(Image Valid)

ينشئ النظام تلقائيًا ملفًا مضغوطًا (ZIP) يحتوي على لوحة التحكم المناسبة لدور المستخدم.

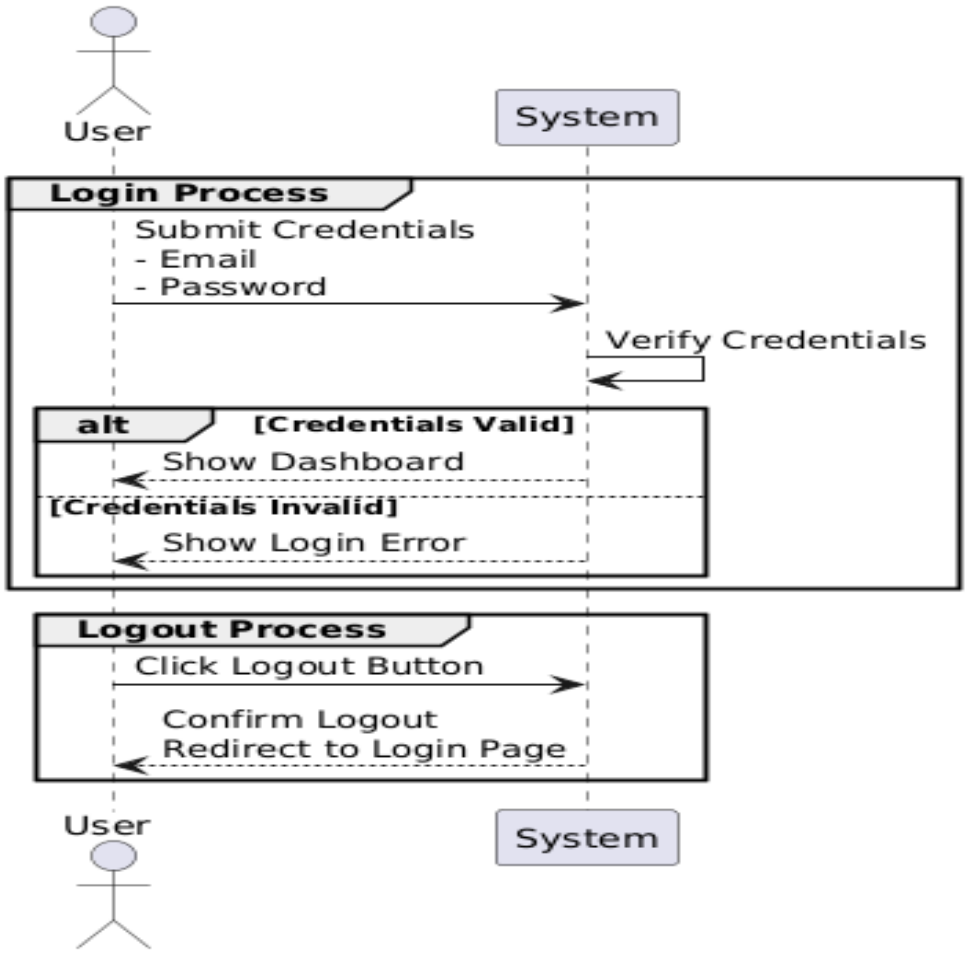
يتم إرسال الملف إلى المستخدم.

يبلّغ النظام المستخدم بنجاح الموافقة على طلب الحساب المهني.

في حال كانت الوثيقة غير صحيحة:(Image Invalid)

يقوم النظام بإشعار المستخدم بأن الوثيقة غير مقبولة، ويطلب منه إعادة المحاولة أو تصحيح الخطأ.

يعكس هذا المخطط حرص النظام على ضبط جودة الحسابات المهنية عبر عملية تحقق دقيقة، مما يعزز الثقة بين المستخدمين والمهنيين داخل المنصة، ويضمن بيئة أكثر احترافية وتنظيمًا.



شكل 6.2: مخطط التسلسل (Sequence Diagram) لعملية تسجيل الدخول والخروج.

شكل 6.2: مخطط التسلسل (Sequence Diagram) لعملية تسجيل الدخول والخروج.

يعرض هذا المخطط التفاعل الديناميكي بين المستخدم والنظام خلال تنفيذ عمليتين أساسيتين ضمن المنصة الرقمية: تسجيل الدخول (Login) وتسجيل الخروج .(Logout) تُمثل هاتان العمليتان أساس الوصول إلى مختلف الوظائف والخدمات التي تقدمها المنصة، وتشكلان نقطة الانطلاق والنهاية لأي تفاعل وظيفي داخل النظام.
ويتضمن المخطط ما يلي:

1 .تسجيل الدخول:(Login Process)

يبدأ المستخدم بإدخال بيانات الاعتماد الخاصة به، والتي تتضمن البريد الإلكتروني وكلمة المرور.

يقوم النظام داخليًا بالتحقق من صحة البيانات المُدخلة عبر آلية تحقق آمنة.

نتيجتان محتملتان:

إذا كانت البيانات صحيحة: يعرض النظام لوحة التحكم الخاصة بالمستخدم(Dashboard) ، مما يسمح له بالوصول إلى كافة الخدمات.

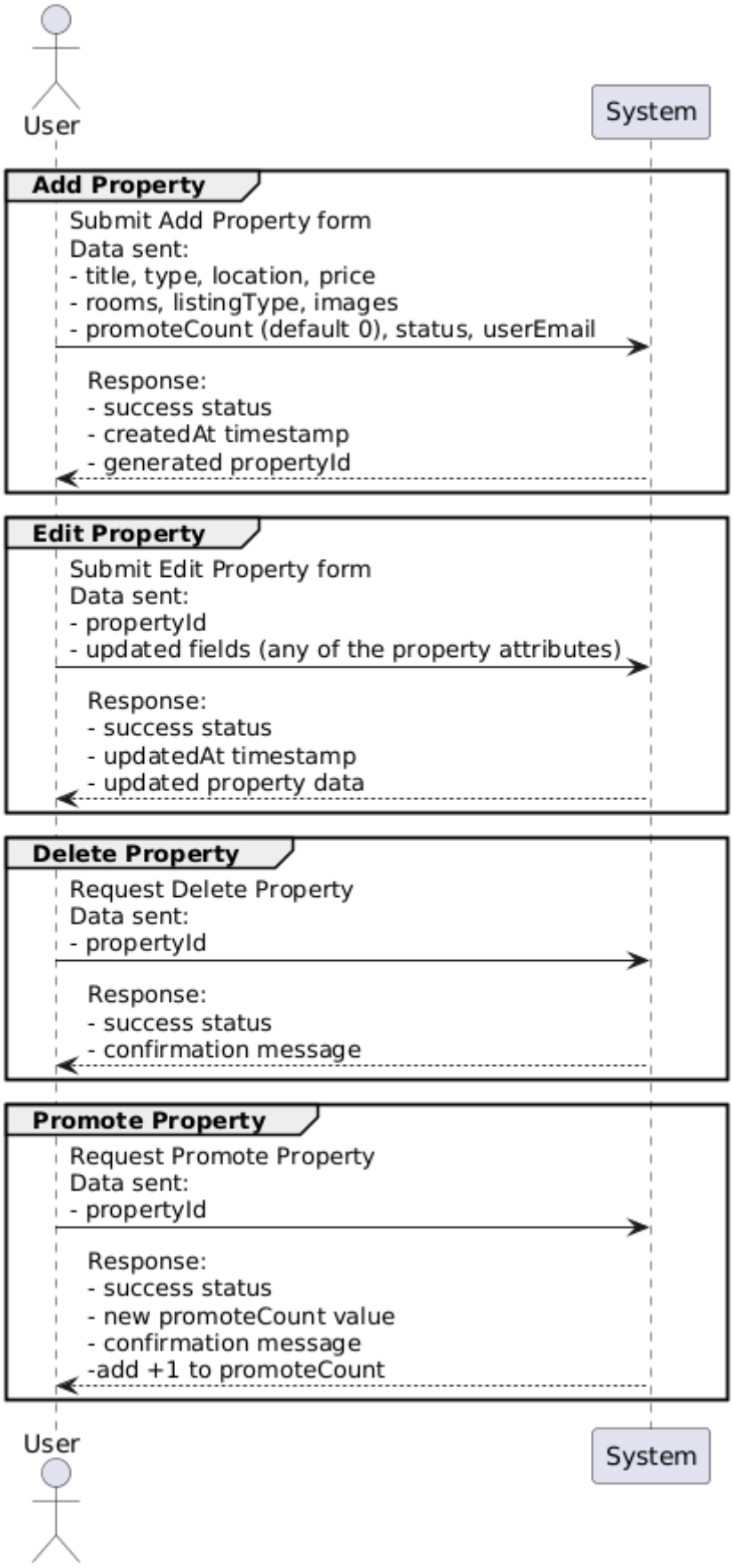
إذا كانت البيانات غير صحيحة: يُظهر النظام رسالة خطأ تُعلم المستخدم بأن بيانات الدخول غير صحيحة.

2 .تسجيل الخروج:(Logout Process)

بعد الانتهاء من استخدام المنصة، ينقر المستخدم على زر "تسجيل الخروج."

يستجيب النظام مباشرة بتأكيد عملية تسجيل الخروج، ويقوم بإعادة توجيه المستخدم إلى صفحة تسجيل الدخول.

يعكس هذا المخطط آلية التحكم في الوصول داخل المنصة، ويوضح كيف تضمن المنظومة حماية البيانات والخصوصية من خلال دورة مصادقة متكاملة تبدأ بتسجيل الدخول وتنتهي بالخروج.



شكل 7.2: مخطط التسلسل (Sequence Diagram) لعملية إدارة العقار.

شكل 7.2: مخطط التسلسل (Sequence Diagram) لعملية إدارة العقار.

يوضح هذا المخطط التفاعل الديناميكي بين المستخدم والنظام خلال تنفيذ أربع حالات استخدام رئيسية متعلقة بإدارة العقارات، وهي: إضافة عقار، تعديل بياناته، حذفه، أو الترويج له. يتمثل التفاعل في إرسال بيانات معينة من طرف المستخدم إلى النظام، ثم استلام استجابة توضح حالة العملية. وتفصيلًا:

1_إضافة عقار:(Add Property)

يبدأ المستخدم بإرسال نموذج يحتوي على بيانات العقار الجديد، مثل العنوان، النوع، الموقع، السعر، عدد الغرف، نوع الإدراج (بيع/كراء)، الصور، وحالة العقار. يتم أيضًا إرسال البريد الإلكتروني لصاحب العقار وعدد مرات الترويج.

يرد النظام بتأكيد نجاح العملية، مع تحديد تاريخ الإنشاء (createdAt) ومعرّف العقار .(propertyId)

2_تعديل عقار:(Edit Property)

يقوم المستخدم بإرسال طلب تعديل يتضمن معرف العقار (propertyId) والحقول التي يريد تعديلها.

يرد النظام بتأكيد النجاح، وتاريخ التحديث(updatedAt) ، إلى جانب البيانات المحدثة للعقار.

3_حذف عقار:(Delete Property)

يطلب المستخدم حذف عقار معين من خلال إرسال معرف العقار فقط.

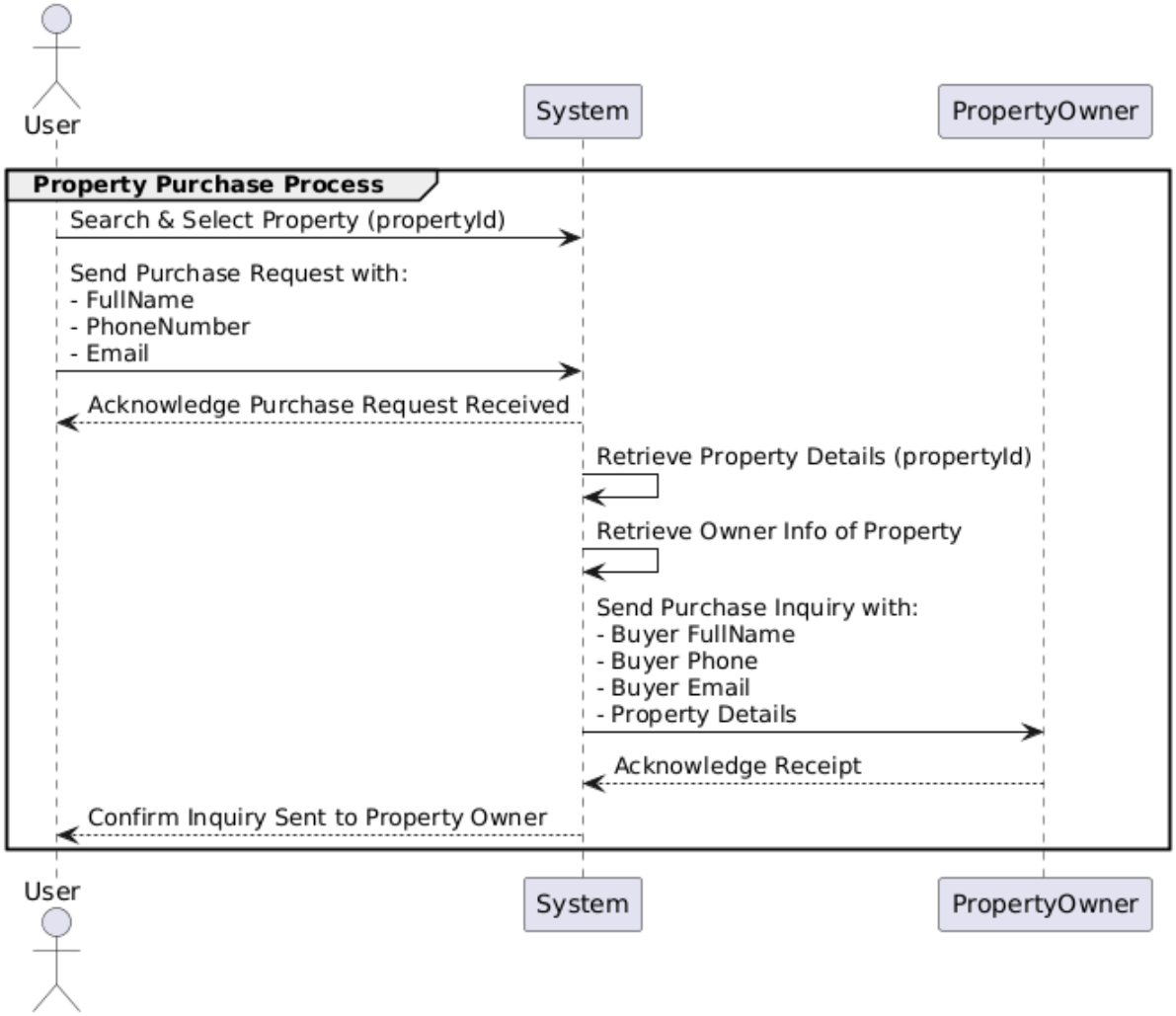
يؤكد النظام نجاح العملية ويعيد رسالة تأكيد.

4_ترويج عقار:(Promote Property)

يرسل المستخدم طلبًا لترويج عقار معين عبر معرفه.

يقوم النظام بزيادة قيمة promoteCount بمقدار 1، ثم يعيد للمستخدم رسالة تأكيد، بالإضافة إلى القيمة الجديدة لمؤشر الترويج.

يُظهر هذا المخطط بوضوح كيفية تفاعل المستخدم مع النظام بطريقة متسلسلة ومنظمة، ويعكس مدى تكامل العمليات المرتبطة بإدارة العقارات داخل المنصة الرقمية.



شكل 8.2: مخطط التسلسل (Sequence Diagram) لعملية طلب شراء العقار.

شكل 8.2: مخطط التسلسل (Sequence Diagram) لعملية طلب شراء العقار.

يعرض هذا المخطط التفاعل التتابعي بين المستخدم، النظام، وصاحب العقار خلال تنفيذ عملية طلب شراء عقار عبر المنصة.

تُظهر الخطوات تسلسل تبادل الرسائل من لحظة بحث المستخدم عن العقار إلى غاية إيصال طلب الشراء لصاحب العقار.

1 .البحث واختيار العقار:

يبدأ المستخدم بالبحث ضمن قائمة العقارات المتاحة، ثم يختار عقارًا معينًا باستخدام معرّفه.(propertyId)

2 .إرسال طلب الشراء:

بعد اختيار العقار، يُرسل المستخدم طلب شراء يحتوي على البيانات التالية:

الاسم الكامل(FullName) ،رقم الهاتف(PhoneNumber) ،البريد الإلكتروني(Email) .

يرد النظام بتأكيد استلام الطلب الأولي من المستخدم.

3 .معالجة الطلب داخليًا:

يقوم النظام بالعمليات التالية:

استخراج تفاصيل العقار المختار باستخدام معرفه.

جلب معلومات صاحب العقار المرتبط بذلك العقار.

4 .إخطار صاحب العقار:

يرسل النظام إشعارًا إلى صاحب العقار يتضمن:

معلومات المشتري (الاسم، الهاتف، البريد الإلكتروني).

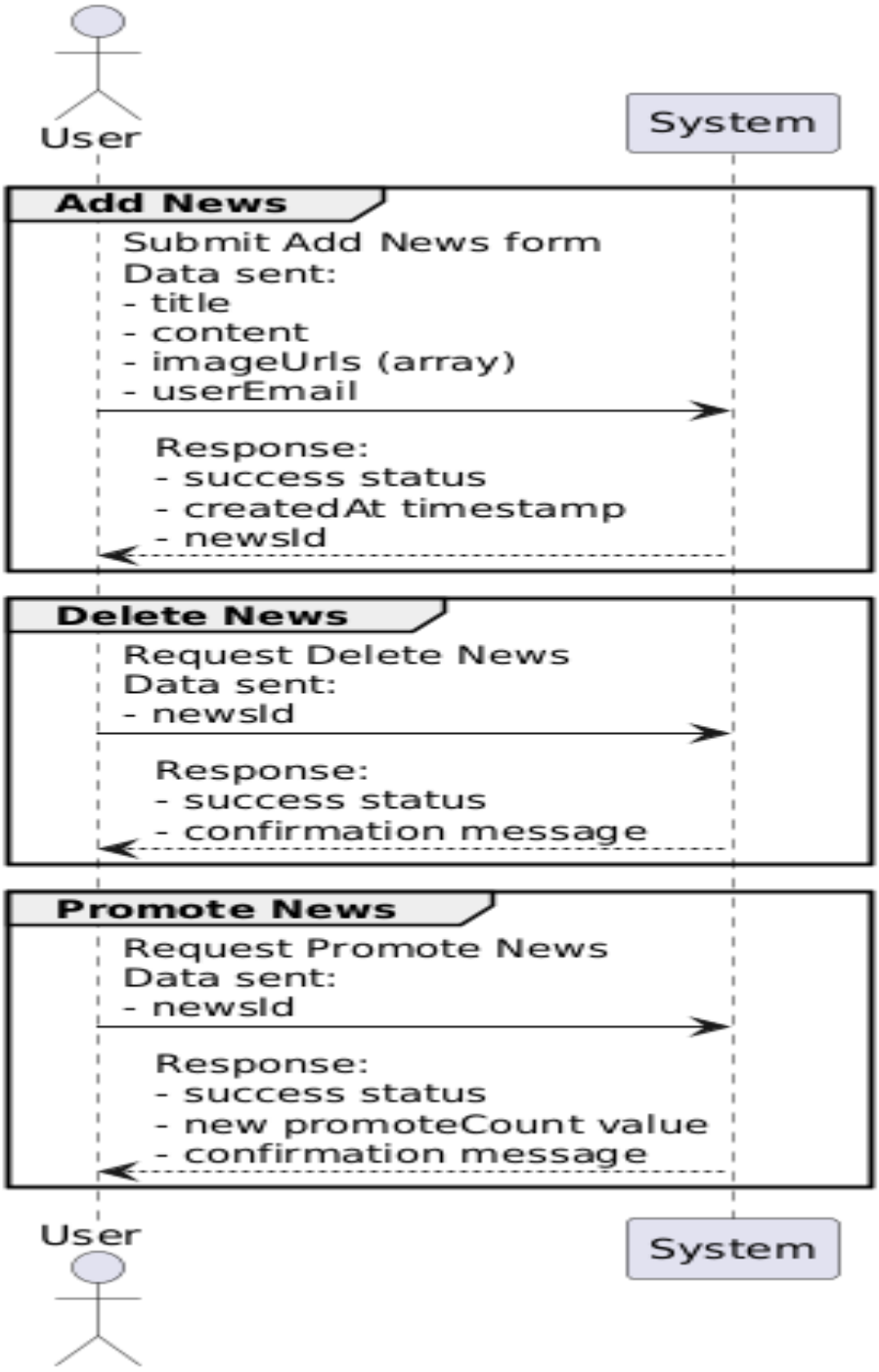
تفاصيل العقار المطلوب شراؤه.

يُجيب صاحب العقار بتأكيد استلام الطلب.

5 .الرد النهائي على المستخدم:

بعد تأكيد وصول الطلب إلى صاحب العقار، يُرسل النظام للمستخدم رسالة تفيد بأن استفساره قد تم توصيله بنجاح.

هذا المخطط يُبرز التكامل بين الجهات المتفاعلة داخل المنصة، ويُوضح كيف تضمن المنظومة الرقمية سلاسة التواصل بين المشترين وأصحاب العقارات، مما يُعزز من فاعلية عمليات البيع داخل النظام العقاري.



شكل 9.2: مخطط التسلسل (Sequence Diagram) لعملية إضافة، حذف، وترويج خبر.

شكل 9.2: مخطط التسلسل (Sequence Diagram) لعملية إدارة الأخبار

يعرض هذا المخطط التفاعل الديناميكي بين المستخدم والنظام أثناء تنفيذ ثلاث حالات استخدام رئيسية متعلقة بإدارة الأخبار داخل المنصة الرقمية، وهي: إضافة خبر، حذف خبر، وترويج خبر.

تتمثل هذه الوظائف في إرسال المستخدم لمجموعة من البيانات إلى النظام، والذي يستجيب بتأكيدات مختلفة حسب طبيعة العملية.

1 .إضافة خبر:(Add News)

يقوم المستخدم بإرسال نموذج يحتوي على المعلومات التالية: عنوان الخبر، محتوى الخبر ، الصور(imageUrls) ،البريد الإلكتروني للمستخدم.

يرد النظام بتأكيد نجاح العملية، مع إرسال تاريخ الإضافة (createdAt) ومعرّف الخبر الجديد.(newsId)

2 .حذف خبر:(Delete News)

يرسل المستخدم طلب حذف خبر معين عبر معرف الخبر.(newsId)

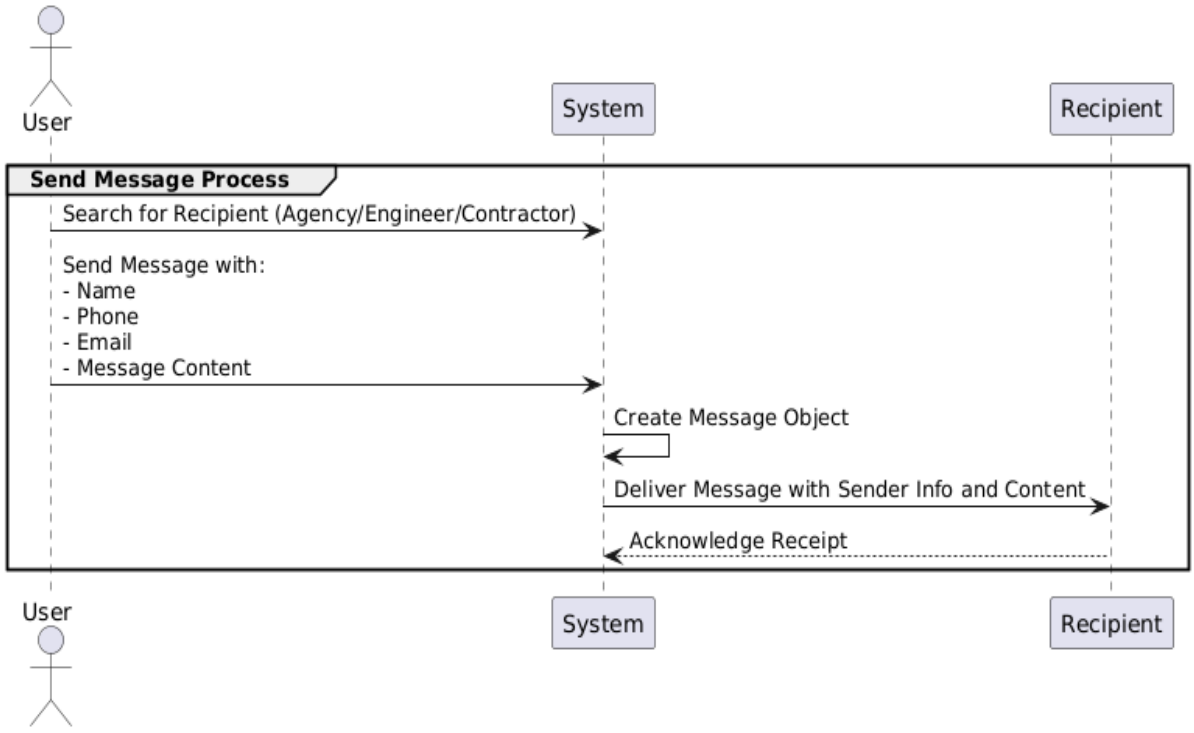
يرد النظام بتأكيد نجاح الحذف، ويُرجع رسالة تأكيد توضح ذلك.

3 .ترويج خبر:(Promote News)

عند طلب الترويج، يُرسل المستخدم معرف الخبر المطلوب.

يقوم النظام بزيادة عدد الترويجات (promoteCount) بمقدار 1، ثم يُرجع رسالة تأكيد تتضمن القيمة الجديدة لمؤشر الترويج.

يُوضح هذا المخطط كيف يمكن للمستخدم إدارة محتوى الأخبار بطريقة متسلسلة ومنظمة، بما يُعزز من الحضور الإعلامي للوكالة العقارية داخل المنصة الرقمية، ويساهم في تحسين التواصل مع جمهور العملاء.



شكل 10.2: مخطط التسلسل (Sequence Diagram) لعملية إرسال رسالة.

شكل 10.2: مخطط التسلسل (Sequence Diagram) لعملية إرسال رسالة.

يُظهر هذا المخطط التفاعل التتابعي بين المستخدم، النظام، والمستلم (سواء كان وكالة عقارية، مهندسًا، أو مقاولًا) أثناء تنفيذ عملية إرسال رسالة داخل المنصة. تمثل هذه العملية أداة تواصل رئيسية تسهّل التفاعل بين مختلف الفاعلين في النظام العقاري الرقمي.

1 .البحث عن المستلم:

يبدأ المستخدم بالبحث عن الطرف الذي يرغب في التواصل معه (الوكالة، المهندس، أو المقاول) عبر واجهة البحث الخاصة بالمنصة.

2 .إرسال الرسالة:

بعد تحديد المستلم، يقوم المستخدم بإرسال رسالة تتضمن:

الاسم الكامل،رقم الهاتف،البريد الإلكتروني،محتوى الرسالة

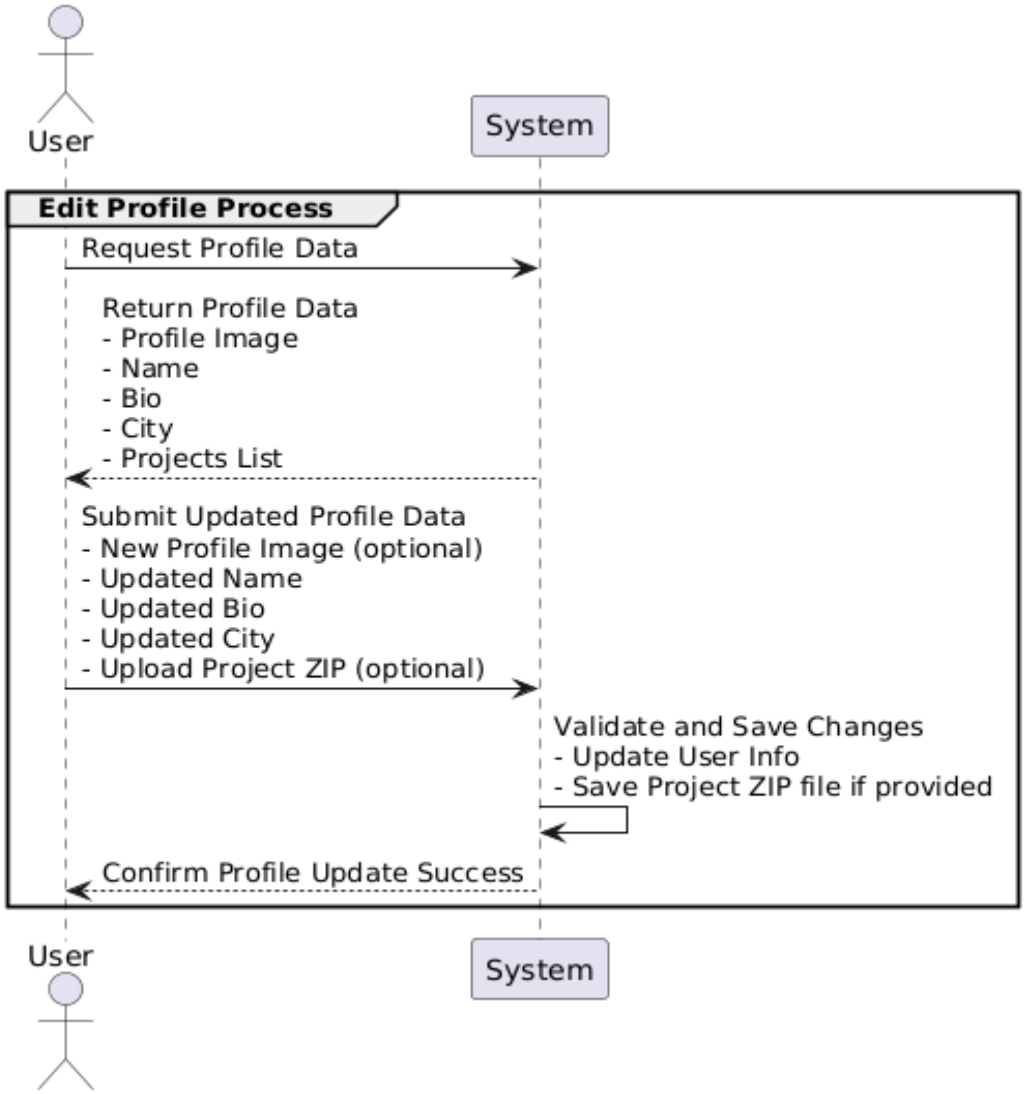
3 .معالجة الرسالة من طرف النظام:

يعمل النظام على إنشاء كائن خاص بالرسالة (Message Object) يتضمن بيانات المرسل ومحتوى الرسالة، ويُجهّزها للإرسال إلى المستلم.

4 .تسليم الرسالة:

يقوم النظام بإرسال الرسالة إلى المستلم المحدد مرفقة بجميع المعلومات الضرورية حول المرسل.

هذا المخطط يُبرز أهمية التفاعل الديناميكي والآني بين المستخدمين داخل المنصة، كما يُبيّن كيف تُمكّن خاصية المراسلة من تعزيز العلاقات المهنية وتسهيل التفاهم بين جميع الأطراف المعنية بإدارة المشاريع العقارية.



شكل 11.2: مخطط التسلسل (Sequence Diagram) لعملية التعديل على الحساب.

شكل 11.2: مخطط التسلسل (Sequence Diagram) لعملية تعديل الحساب

يعرض هذا المخطط التسلسلي التفاعل بين المستخدم والنظام خلال تنفيذ عملية تعديل الملف الشخصي، والتي تُعد من العمليات الجوهرية في المنصة العقارية، نظرًا لأهميتها في إبراز هوية المستخدم المهنية (مهندس، مقاول، أو وكالة). تتكون العملية من عدة مراحل متسلسلة كما يلي:

1 .طلب بيانات الملف الشخصي:

يبدأ المستخدم بإرسال طلب إلى النظام لاسترجاع معلومات ملفه الشخصي الحالية، بهدف تعديلها.

2 .استلام بيانات الملف:

يستجيب النظام للطلب بإرسال جميع بيانات الملف الشخصي الحالية، والتي تشمل:

صورة الملف الشخصي،الاسم،النبذة التعريفية(Bio) ،المدينة،قائمة المشاريع السابقة أو الحالية.

3 .إرسال البيانات المحدّثة:

يقوم المستخدم بإرسال التعديلات الجديدة على البيانات، والتي قد تشمل:

صورة جديدة (اختيارية)

اسم جديد،نبذة محدثة،تغيير المدينة،رفع ملف مضغوط (ZIP) يحتوي على مشاريع إضافية (اختياري)،

4 .معالجة البيانات داخل النظام:

يتولى النظام التحقق من صحة البيانات المُرسلة ثم يقوم بـ:

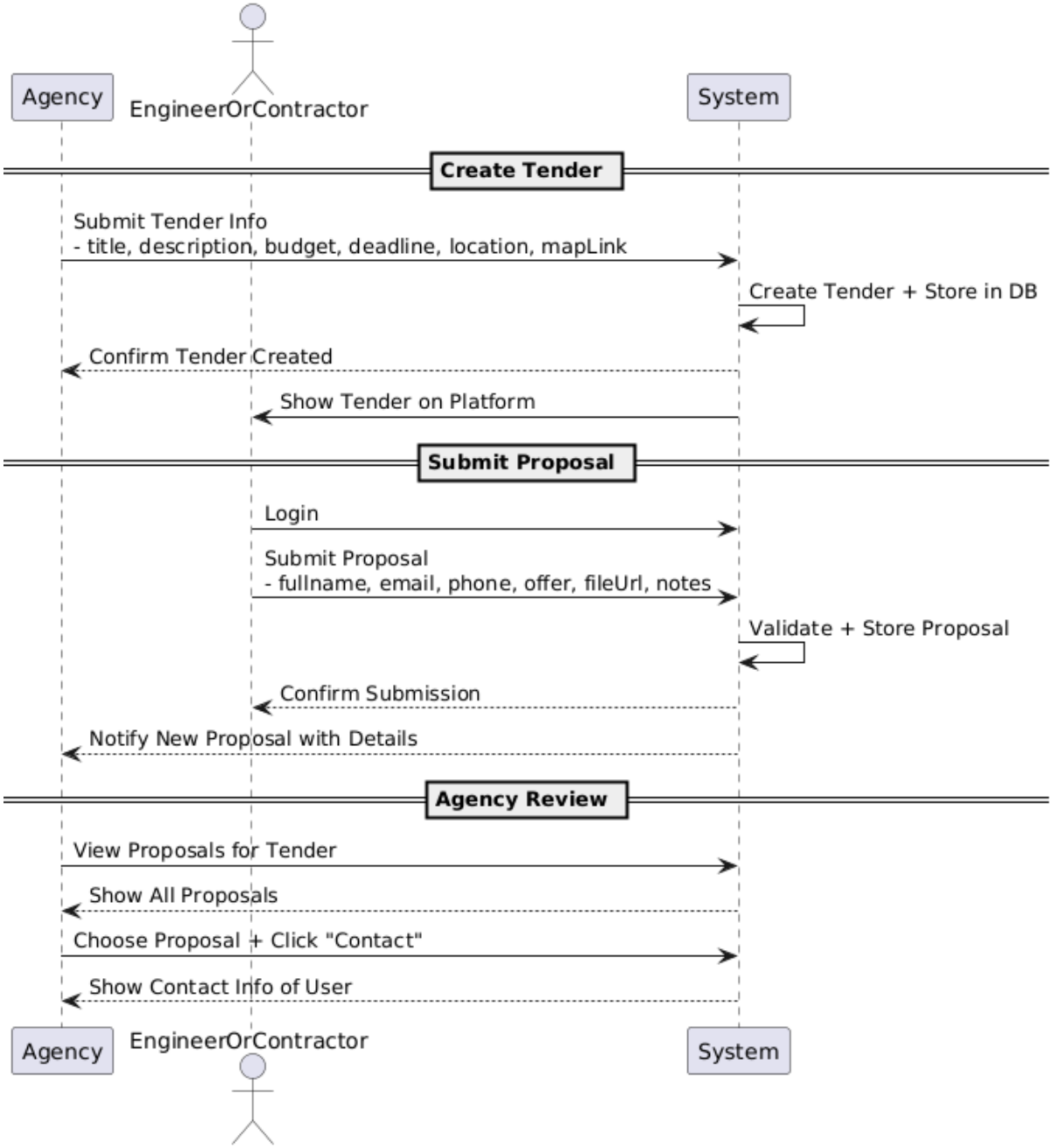
تحديث المعلومات الشخصية للمستخدم في قاعدة البيانات

حفظ الملف المضغوط للمشاريع إذا تم إرساله

5 .تأكيد نجاح التحديث:

يرسل النظام رسالة تأكيد للمستخدم تفيد بنجاح عملية التعديل.

يعكس هذا المخطط بشكل منهجي التفاعل بين المستخدم والمنصة في سياق تحديث بياناته المهنية، ويُظهر كيف تُمكّن هذه العملية المستخدم من الحفاظ على ملف احترافي محدث يعزز من فرصه في التفاعل داخل البيئة الرقمية العقارية.



شكل 12.2: مخطط التسلسل (Sequence Diagram) لعملية المناقصة.

شكل 12.2: مخطط التسلسل (Sequence Diagram) لعملية المناقصة.

يعرض هذا المخطط تسلسل التفاعل بين الوكالة العقارية(Agency) ، والنظام(System) ، والمهني (مهندس أو مقاول) خلال مناقصة داخل المنصة، ويشمل ذلك ثلاث مراحل رئيسية: إنشاء المناقصة، تقديم العرض، ومراجعة العروض من طرف الوكالة. يوضح المخطط كيف تتم هذه العمليات بشكل ديناميكي ومنظم لضمان شفافية المعاملات وسهولة التواصل بين الأطراف.

1 .إنشاء المناقصة:(Create Tender)

تقوم الوكالة بإرسال نموذج يحتوي على بيانات المناقصة، مثل: العنوان، الوصف، الميزانية، الموعد النهائي، الموقع، ورابط الخريطة.

يقوم النظام بتخزين بيانات المناقصة في قاعدة البيانات ويؤكد للوكالة نجاح العملية.

مباشرة بعد ذلك، يتم عرض المناقصة على المنصة ليتمكن المهندسون والمقاولون من الاطلاع عليها.

2 .تقديم العرض من طرف المهني:(Submit Proposal)

يبدأ المستخدم (المهندس أو المقاول) بتسجيل الدخول إلى النظام.

بعد المصادقة، يقوم بتقديم عرض يشمل: الاسم الكامل، البريد الإلكتروني، رقم الهاتف، رابط الملف التقني، وملاحظات إضافية.

يقوم النظام بالتحقق من صحة البيانات وتخزين العرض، ثم يُبلغ المستخدم بنجاح العملية.

في ذات الوقت، يتم إخطار الوكالة بوجود عرض جديد مرفقًا بجميع التفاصيل.

3 .مراجعة العروض من طرف الوكالة:(Agency Review)

تدخل الوكالة إلى النظام لاستعراض العروض المقدمة على المناقصة المحددة.

يعرض النظام كافة العروض المتعلقة بالمناقصة.

عند اختيار أحد العروض، تقوم الوكالة بالنقر على زر "اتصال"، ليتم عرض معلومات الاتصال الخاصة بالمهني الذي قدم العرض.

يعكس هذا المخطط آلية التعاون بين الأطراف داخل المنصة، كما يُبرز كيفية تنظيم عمليات المناقصة والعروض بما يضمن الكفاءة والاحترافية في إدارة المشاريع العقارية. كما يسهم هذا التفاعل في تسهيل اتخاذ القرار من قبل الوكالة عبر منصة موحدة وشفافة.

3.3.2 مخطط الفصل Diagram de class :

التعريف:

" مخطط الفصل في لغة النمذجة الموحدة (UML) هو نوع من مخططات البنية الثابتة التي تصف بنية النظام من خلال إظهار فئات النظام، وسماتها، وعملياتها (أو أساليبها)، والعلاقات بين الكائنات."[13]

مميزاته :

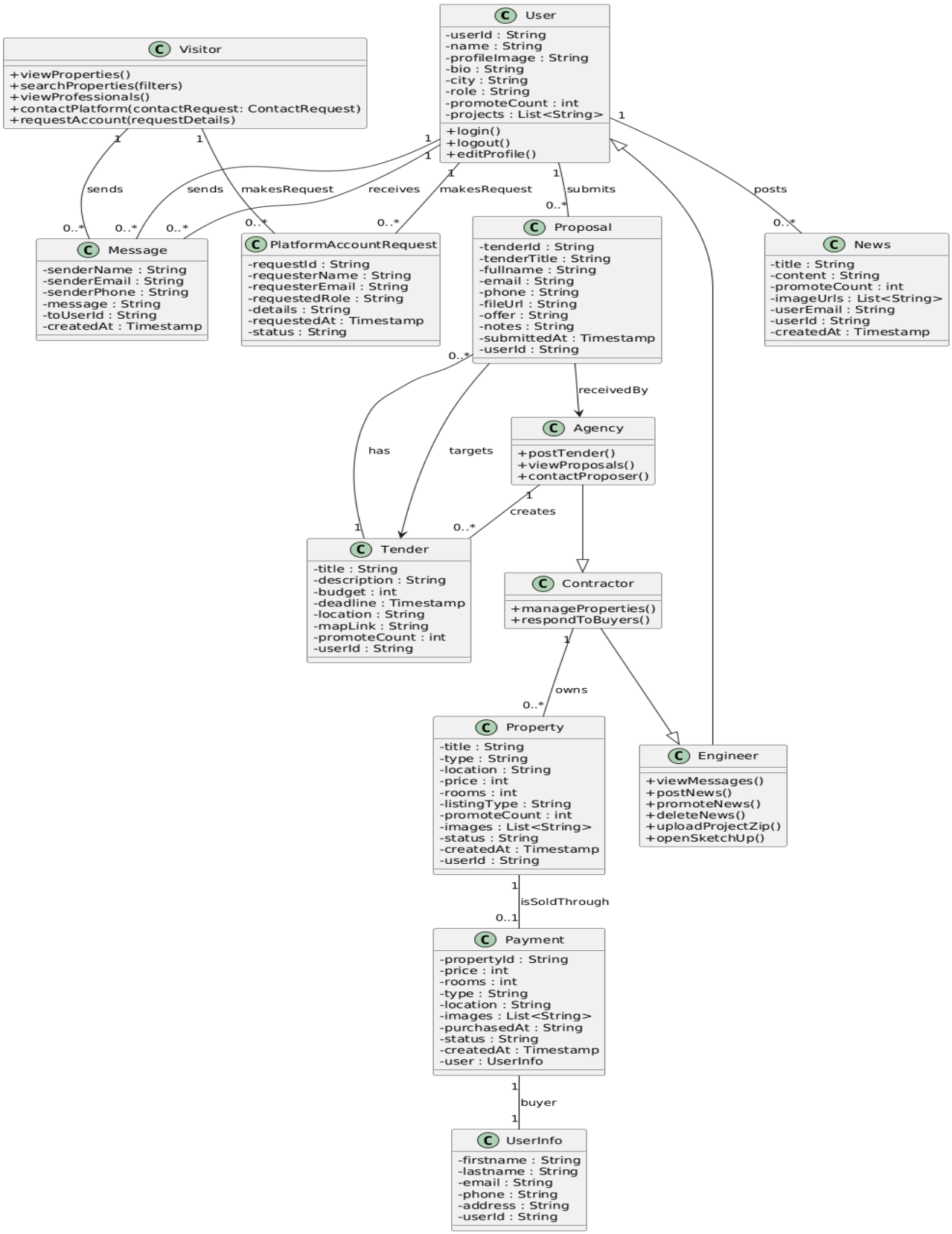
_يُظهر هيكلًا ثابتًا للمصنفات في النظام.

_يوفر الرسم التخطيطي تدوينًا أساسيًا لمخططات الهياكل الأخرى التي تحددها لغة النمذجة الموحدة (UML).

_مفيد للمطورين وأعضاء الفريق الآخرين أيضًا.

_يمكن لمحللي الأعمال استخدام مخططات الفئات لنمذجة الأنظمة من منظور الأعمال." [13]

مخطط class للمنصة :



شكل 13.2: مخطط الصنف (Class Diagram).

شكل 13.2: مخطط الصنف (Class Diagram).

يوضح هذا المخطط البنية العامة للفئات (Classes) في النظام العقاري الرقمي، بالإضافة إلى العلاقات التي تربط بينها. يمثل المخطط الكيانات الأساسية التي يتعامل معها النظام، مثل المستخدمين، العقارات، المناقصات، الرسائل، وغيرها. ويمكن تصنيف هذه الفئات كما يلي:

1. الكيانات الرئيسية:

المستخدم (User) :

يمثل الحساب الرئيسي في المنصة، ويحتوي على معلومات مثل:

الاسم، المدينة، الصورة الشخصية، السيرة الذاتية، وعدد مرات الترويج.

الوظائف المتوفرة: تسجيل الدخول، تسجيل الخروج، تعديل الملف الشخصي.

المهندس (Engineer) :

يرث خصائص المستخدم، ويُضاف إليه القدرة على:

نشر الأخبار، الترويج للأخبار، حذف الأخبار، رفع ملفات المشاريع، واستخدام أدوات تصميم خارجية مثل Homestyler.

المقاول (Contractor):

يرث من Engineer ويُضاف إليه وظائف مثل:

إدارة العقارات والرد على استفسارات المشترين.

الوكالة (Agency) :

ترث من Contractor وتضيف إمكانية:

نشر المناقصات، عرض المقترحات، والتواصل مع أصحاب العروض.

2 .الزائر غير المسجل:(Visitor)

يُتيح للمستخدمين غير المسجلين:

تصفح العقارات، البحث باستخدام المرشحات، عرض الملفات الشخصية للمهنيين، التواصل مع المنصة، وطلب فتح حساب احترافي.

3 .الكيانات الداعمة في النظام:

العقار (Property) :

يحتوي على بيانات مثل: العنوان، النوع، الموقع، السعر، عدد الغرف، نوع العرض (بيع/كراء)، الصور، الحالة، وتاريخ النشر.

عملية الشراء (Payment) :

تخزن بيانات عمليات شراء العقارات، مثل:

معرف العقار، السعر، عدد الغرف، نوع العقار، صور، تاريخ الشراء، والحالة.

معلومات المستخدم (UserInfo) :

تُستخدم داخل العمليات المالية والعروض، وتشمل:

الاسم الكامل، البريد الإلكتروني، الهاتف، العنوان، ومعرف المستخدم.

الأخبار (News) :

تمثل المنشورات الإخبارية للمستخدمين المهنيين، وتشمل:

العنوان، المحتوى، الصور، البريد الإلكتروني، ومعرف الناشر.

الرسائل (Message) :

تتضمن بيانات مثل: اسم المرسل، بريده، هاتفه، محتوى الرسالة، المستلم، وتاريخ الإرسال.

المناقصة (Tender) :

تُنشر من قبل الوكالات العقارية، وتشمل:

العنوان، الوصف، الميزانية، تاريخ الانتهاء، الموقع، الرابط الجغرافي، وعدد الترويج.

عرض المشروع (Proposal) :

تُرسل كرد على المناقصات، وتحتوي على:

اسم صاحب العرض، بريده، رقم هاتفه، المبلغ المقترح، رابط الملف، ملاحظات، تاريخ التقديم، ومعرف المستخدم.

طلب حساب (PlatformAccountRequest) :

يحتوي على بيانات مثل: اسم مقدم الطلب، بريده الإلكتروني، الدور المهني المطلوب، تفاصيل إضافية، وتاريخ الطلب.

4 .العلاقات بين الكيانات:

كل Contractor يمكن أن يملك عدة.Property

كل User يمكنه نشر عدة News وتقديم عدة Proposal واستلام عدة Message.

كل Tender يمكن أن يُقابَل بعدة Proposal.

Paymentترتبط بـ Property وبـ UserInfo كمشتري.

Proposalمرتبطة بـ Tender و Agency.

كل Visitor أو User يمكنه تقديم عدة PlatformAccountRequest.

4.2.الخاتمة :

وفي الختام، قمنا في هذا الفصل باستعراض أهم المفاهيم الخاصة بلغة نمذجةUML ، كما قمنا بشرح العمليات التي تتم داخل المنصة ولوحة التحكم الخاصة بها من خلال مخططات التسلسل (Sequence Diagrams). كذلك عرضنا أدوار جميع المستخدمين وتفاعلاتهم مع النظام من خلال مخططات الاستخدام(Use Case Diagrams) ، كما شرحنا هيكلة قاعدة البيانات والعلاقات بين الكائنات البرمجية باستخدام مخطط الصنف.(Class Diagram)

الفصل الثالث : التطوير

1.3. مقدمة :

بعد أن تطرقنا في الفصول السابقة إلى دراسة الوضع الحالي وتحليل المشاكل المرتبطة بالقطاع العقاري، نصل في هذا الفصل إلى المرحلة الأهم من المشروع، وهي مرحلة التطوير. في هذا الجزء، سنوضح كيف تم تحويل الفكرة النظرية إلى تطبيق عملي ملموس، مع بيان الأدوات والبرمجيات التي تم استخدامها لبناء المنصة الرقمية.

نبدأ بعرض بيئة التطوير، سواء من حيث العتاد (الأجهزة) أو البرمجيات المعتمدة، ثم ننتقل إلى شرح واجهات المستخدم الخاصة بالمنصة، موضحين كيفية تصميمها والأسباب التي دفعتنا لاختيار هذا الشكل من التصميم. كما سنتناول أهم التقنيات والأدوات التي تم الاعتماد عليها، والدور الذي لعبته في تسهيل عملية التطوير وتحسين تجربة الاستخدام بشكل عام.كما سنقدم قواعد البيانات المستخدمة ، وايضا كيفية العمل بنموذج NoSql و firebase .

2.3. بيئة العمل :

1.2.3.العتاد المستخدم :

تم استخدام حاسوب محمول بمواصفات تقنية متوسطة إلى جيدة، كالتالي:

المعالج : Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz 1.90 GHz .

الذاكرة العشوائية : 16 (RAM) جيغابايت .

قرص التخزين 512 :(SSD) جيغابايت .

نظام التشغيل: Windows 10 x64 .

توفر هذه المواصفات بيئة مناسبة لتشغيل برامج التصميم والبرمجة مثل VS Code، StarUML ولغات البرمجة مثل flutter وغيرها بدون بطء أو مشاكل تقنية.

2.2.3. لبرمجيات والأدوات المستخدمة :

1.2.2.3 Microsoft Visual Studio Code :

" هو محرر أكواد مجاني، قوي، وخفيف الوزن، يعمل على أنظمة ويندوز، وماك، ولينكس. مبني على نظام مفتوح المصدر، ويتميز بقابلية تخصيص عالية، مع أكثر من 25,000 إضافة، تناسب جميع المطورين وجميع لغات البرمجة." [14]

مميزاته :

_ دعم متعدد المنصات (macOS، Linux، Windows).

_ تصحيح الأخطاء التفاعلي.

_ قابلية التخصيص من خلال الإضافات والسمات. و دعم لمجموعة واسعة من لغات البرمجة.

3.2.2.4. Flutter :

" هي مجموعة أدوات واجهة مستخدم متعددة المنصات، مصممة للسماح بإعادة استخدام الكود عبر أنظمة التشغيل مثل iOS وAndroid والويب وسطح المكتب، مع تمكين التطبيقات من التفاعل مباشرةً مع خدمات المنصة الأساسية. الهدف هو تمكين المطورين من تقديم تطبيقات عالية الأداء تعمل بسلاسة على منصات مختلفة، مع مراعاة الاختلافات أينما وجدت، مع مشاركة أكبر قدر ممكن من الكود." [15]

Dart :

"لغة برمجة مُحسّنة للعملاء لتطوير تطبيقات سريعة على أي منصة. هدفها توفير لغة برمجة أكثر إنتاجية للتطوير متعدد المنصات، مقترنة بمنصة تشغيل مرنة لأطر عمل التطبيقات." [15]

المميزات :

_تعمل تطبيقات Flutter في آلة افتراضية توفر إعادة تحميل سريعة للتغييرات مع الحفاظ على الحالة دون الحاجة إلى إعادة تجميع كاملة.

_ (على الويب، يدعم Flutter إعادة التشغيل السريع وإعادة التحميل السريع مع وجود علامة). للإصدار، تُجمّع تطبيقات Flutter مباشرةً إلى الكود الآلي، سواءً بتعليمات Intel x64 أو ARM، أو إلى JavaScript إذا كانت تستهدف الويب.

_الإطار مفتوح المصدر، مع ترخيص BSD متساهل، ويضم منظومة متكاملة مزدهرة من حزم الجهات الخارجية التي تُكمل وظائف المكتبة الأساسية. [15]

3.2.2.3. HTML (HyperText Markup Language) :

"هي اللبنة الأساسية للويب. فهي تُعرّف معنى وبنية محتوى الويب. تُستخدم تقنيات أخرى إلى جانب HTML عادةً لوصف مظهر/عرض صفحات الويب (CSS) أو وظائف/سلوك صفحات الويب (JavaScript). " [16]

4.2.2.3. Cascading Style Sheets (CSS) :

"هي لغة أوراق أنماط تُستخدم لوصف عرض مستند مكتوب بلغة HTML أو XML (بما في ذلك لغات XML مثل SVG وMathML وXHTML). تصف CSS كيفية عرض العناصر على الشاشة، أو الورق، أو الكلام، أو الوسائط الأخرى."[16]

JavaScript .5.2.2.3 :

"هي لغة برمجة خفيفة مُفسَّرة (أو مُجمَّعة آنيًا) ذات دوال من الطراز الأول. ورغم شهرتها الواسعة كلغة برمجة نصية لصفحات الويب، إلا أن العديد من البيئات غير المُخصصة للمتصفحات تستخدمها أيضًا، مثل Node.js وApache CouchDB وAdobe Acrobat. جافا سكريبت لغة ديناميكية، متعددة النماذج، أحادية الخيط، قائمة على النماذج الأولية، تدعم أنماط البرمجة الكائنية التوجه، والبرمجة الإلزامية، والبرمجة التصريحية (مثل البرمجة الوظيفية)."[16]

FIREBASE .6.2.2.3 :

" عبارة عن مجموعة من أدوات التطوير المستندة إلى السحابة والتي تساعد مطوري تطبيقات الأجهزة المحمولة على إنشاء تطبيقاتهم ونشرها وتوسيع نطاقها."[17]

مميزات FIREBASE :

يوفر Firebase مجموعة متنوعة من الميزات، بما في ذلك:

- المصادقة. يوفر Firebase طريقة آمنة وسهلة للمستخدمين لتسجيل الدخول إلى تطبيقاتهم. يمكن للمطورين استخدام مصادقة Firebase لدعم تسجيل الدخول باستخدام البريد الإلكتروني وكلمة المرور، وتسجيل الدخول عبر جوجل، وتسجيل الدخول عبر فيسبوك، والمزيد.
- قاعدة بيانات آنية. قاعدة بيانات Firebase آنية هي قاعدة بيانات NoSQL مستضافة على السحابة، تتيح للمؤسسات تخزين البيانات ومزامنتها آنيًا عبر جميع أجهزة مستخدميها. هذا يُسهّل بناء تطبيقات مُحدّثة باستمرار، حتى في حالة عدم اتصال المستخدمين بالإنترنت.
- المراسلة السحابية. خدمة Firebase Cloud Messaging (FCM) هي خدمة تُتيح للشركات إرسال رسائل إلى أجهزة مستخدميها، حتى في حالة عدم استخدامهم للتطبيق. يمكن للمطورين استخدام FCM لإرسال إشعارات فورية، وتحديث محتوى التطبيق، والمزيد.
- Crashlytics. خدمة Firebase Crashlytics هي خدمة تُساعد المؤسسات على تتبع الأعطال وإصلاحها في تطبيقاتها. تُوفر Crashlytics تقارير مُفصلة عن الأعطال، حتى يتمكنوا من تحديد السبب الجذري وحل المشكلة بسرعة.
- مراقبة الأداء. توفر خدمة مراقبة أداء Firebase رؤىً حول أداء تطبيقاتها. يمكن للمؤسسات استخدام خدمة مراقبة الأداء لتتبع مقاييس مثل استخدام وحدة المعالجة المركزية (CPU) واستخدام الذاكرة وحركة مرور الشبكة.
- مختبر الاختبار: Firebase Test Lab هي خدمة سحابية تتيح للمطورين اختبار تطبيقاتهم على مجموعة متنوعة من الأجهزة والتكوينات. هذا يساعدهم على ضمان عمل التطبيق بكفاءة على مجموعة متنوعة من الأجهزة وفي ظروف الشبكة المختلفة."[17]

7.2.2.3. Cloudinary :

" هو حل شامل لإدارة الصور والفيديوهات لمواقع الويب وتطبيقات الجوال، يغطي كل شيء بدءًا من تحميل الصور والفيديوهات، والتخزين، والمعالجات، والتحسينات، وصولًا إلى التسليم." [18]

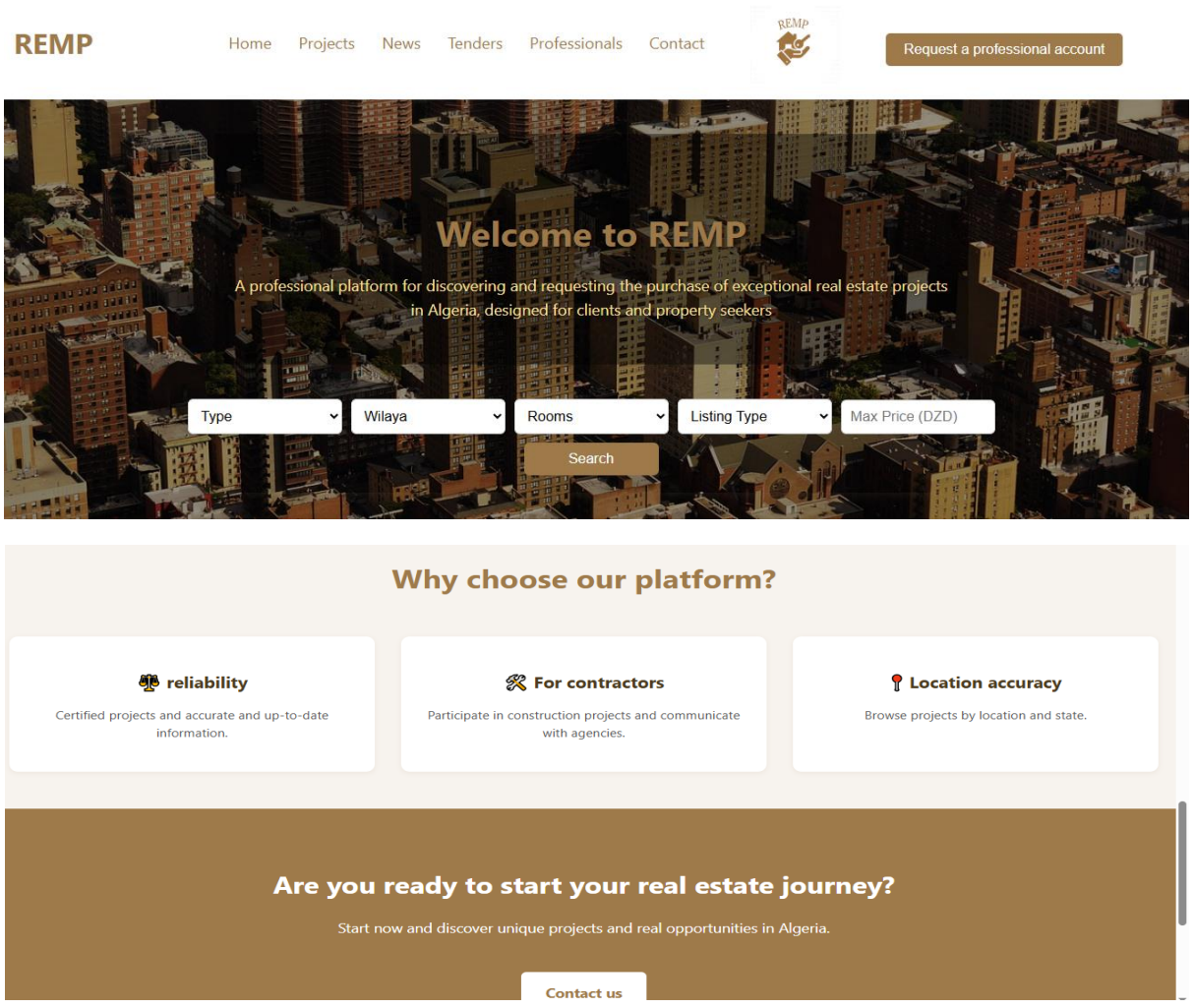
مميزاته :

"مع Cloudinary، يمكنك بسهولة تحميل الصور والفيديوهات إلى السحابة وأتمتة عمليات المعالجة الذكية لهذه الوسائط دون الحاجة إلى تثبيت أي برنامج آخر. ثم يُسلّم Cloudinary وسائطك بسلاسة عبر شبكة توصيل محتوى سريعة (CDN)، مُحسّنة وفقًا لأفضل الممارسات في هذا المجال.

بالإضافة إلى ذلك، يوفر Cloudinary واجهات برمجة تطبيقات شاملة وقدرات إدارة، يمكنك دمجها بسهولة مع تطبيقات الويب والجوال."[18]

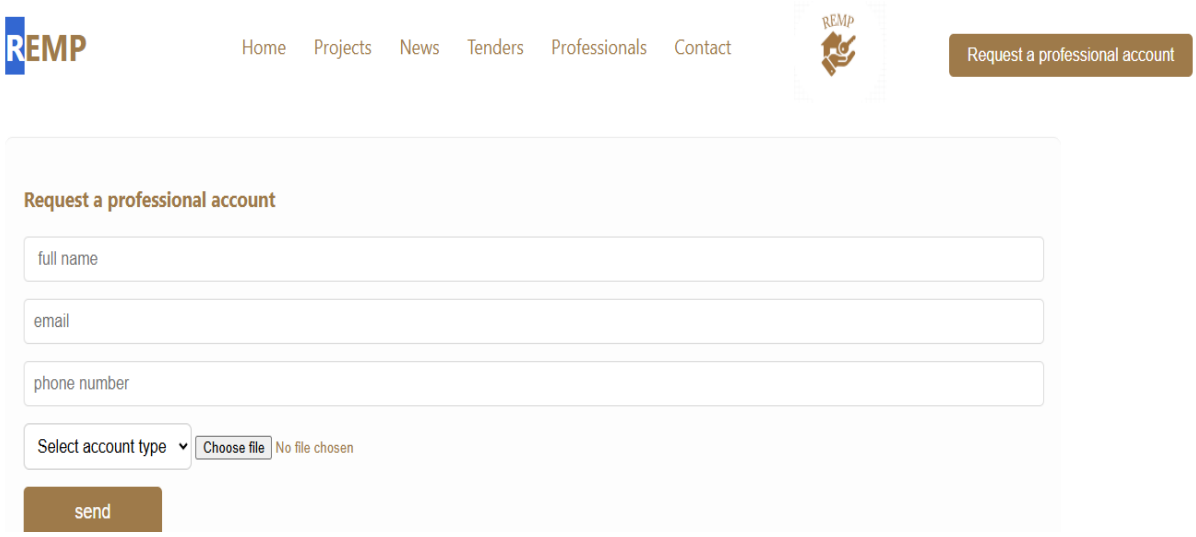
3.3. واجهات التطبيق :

1.3.3. واجهات المنصة :



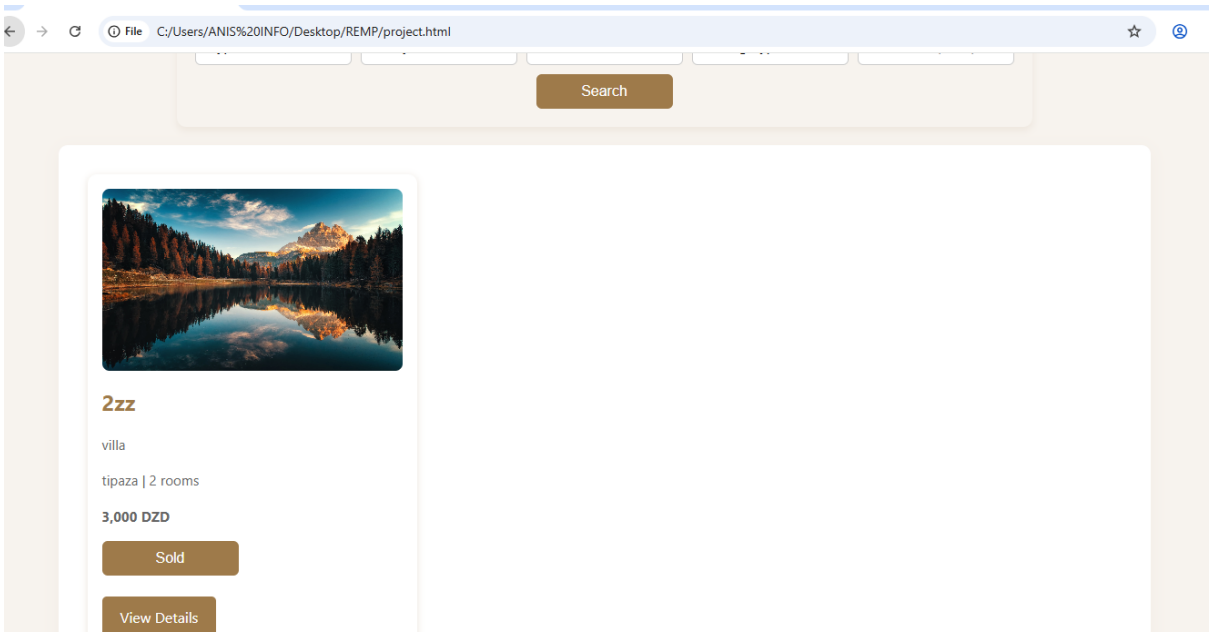
صورة 1.3. الصفحة الرئيسية

صفحة ال home حيث يجد المستخدم نفسه امام العديد من المزيا والصفحات الاخرى مع التعريف بالمنصة وغيرها ، كما يمكنه التنقل بين الصفحات الاخرى .



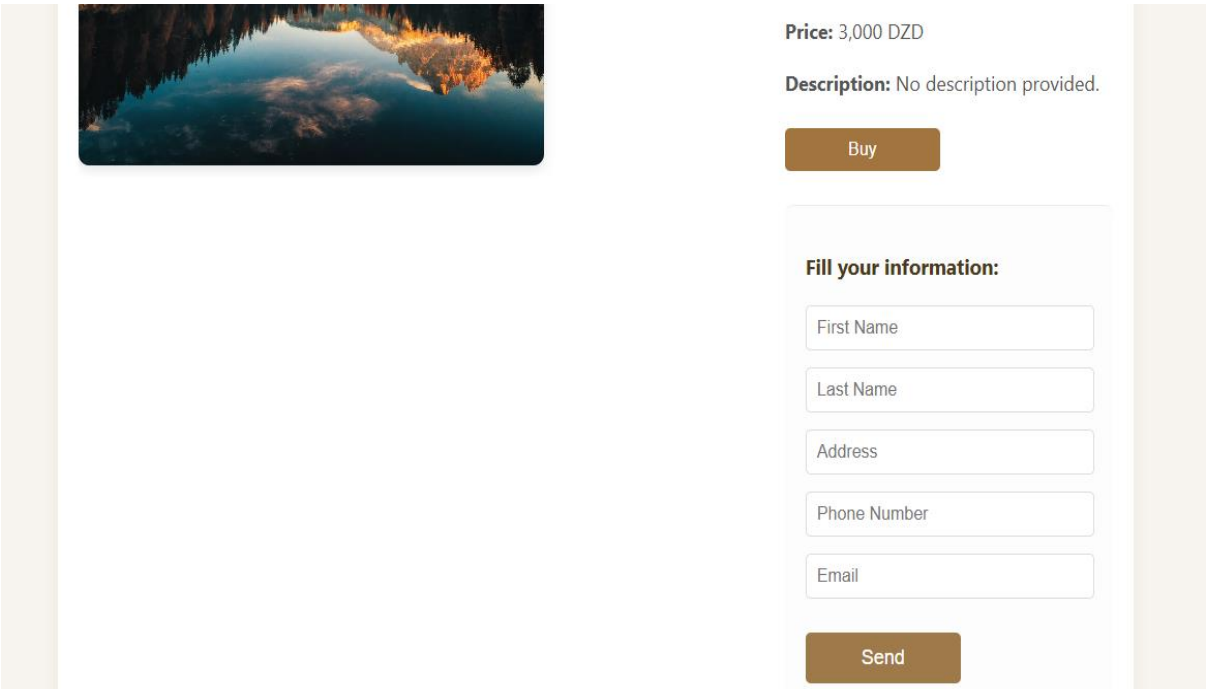
صورة 2.3. طلب حساب

هنا يمكن للمستخدم ان يقوم بطلب الحساب الرئيسي لاجل ان يستخدم control panel الخاصة بالمنصة حيث يدخل الاسم الكامل ،الايميل ،رقم الهاتف ونوع الحساب (مهندس،مقاول او وكالة)وصورة تثبت وضعيته ولاحقا يتم التحقق منها من قبل الادارة وارسال تطبيق flutter للوحة التحكم مع كلمة السر واسم المستخدم له في الايميل



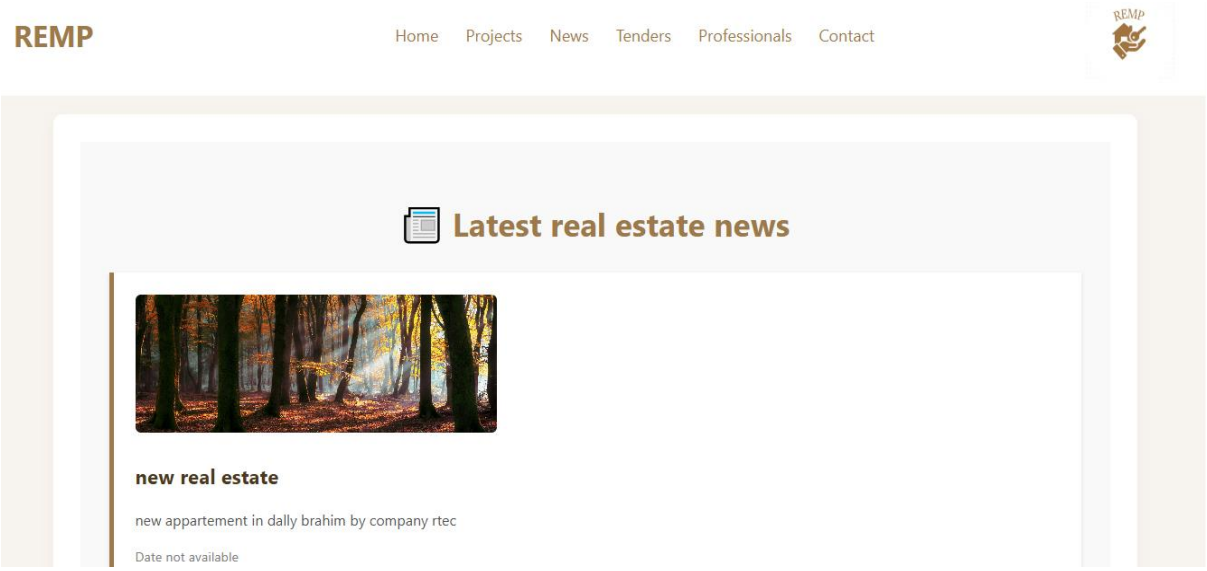
صورة 3.3. صفحة المشاريع

في هاته الصفحة يمكن للمستخدم ان يرى العقارات مع صورتها ومعلوماتها كما لديه بحث متقدم (النوع ، السعر ، عدد الغرف ، الولاية ، للكراء او البيع) ويرى حالة العقار اذا كان متوفر ام مباع ثم يضغط زر view details لرؤوية او طلب الشراء .



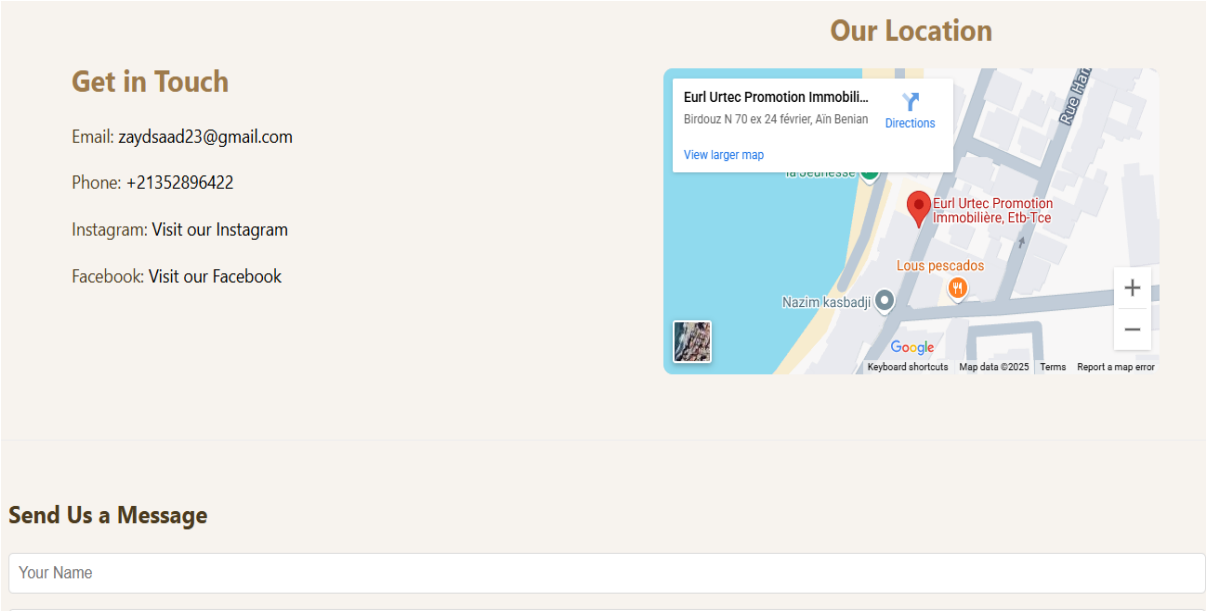
صورة 4.3. تفاصيل العقار

هنا يرى المستخدم تفاصيل العقار مع كل صوره وايضا اذا كان متوفر يضغط buy لتقديم طلب شراء ثم يملئ معلوماته ويضغط send لتقديم الطلب وفور الارسال تتحول الحالة ل sold حتى يغيرها صاحب العقار ولا يمكن تقديم طلب شراء اخر .



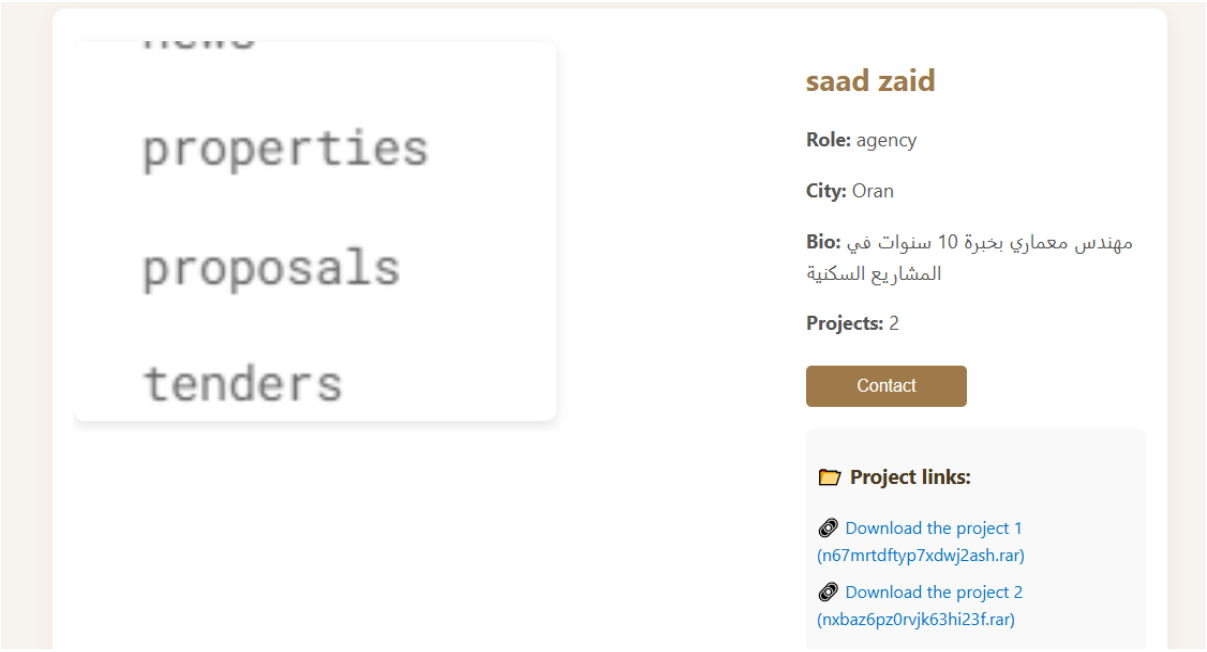
صورة 5.3. صفحة الاخبار

هنا يرى المستخدم جميع الاخبار التي تم رفعها سواء من قبل وكالات ام مقاولين ويتصفح على حسب موعد صدور الخبر



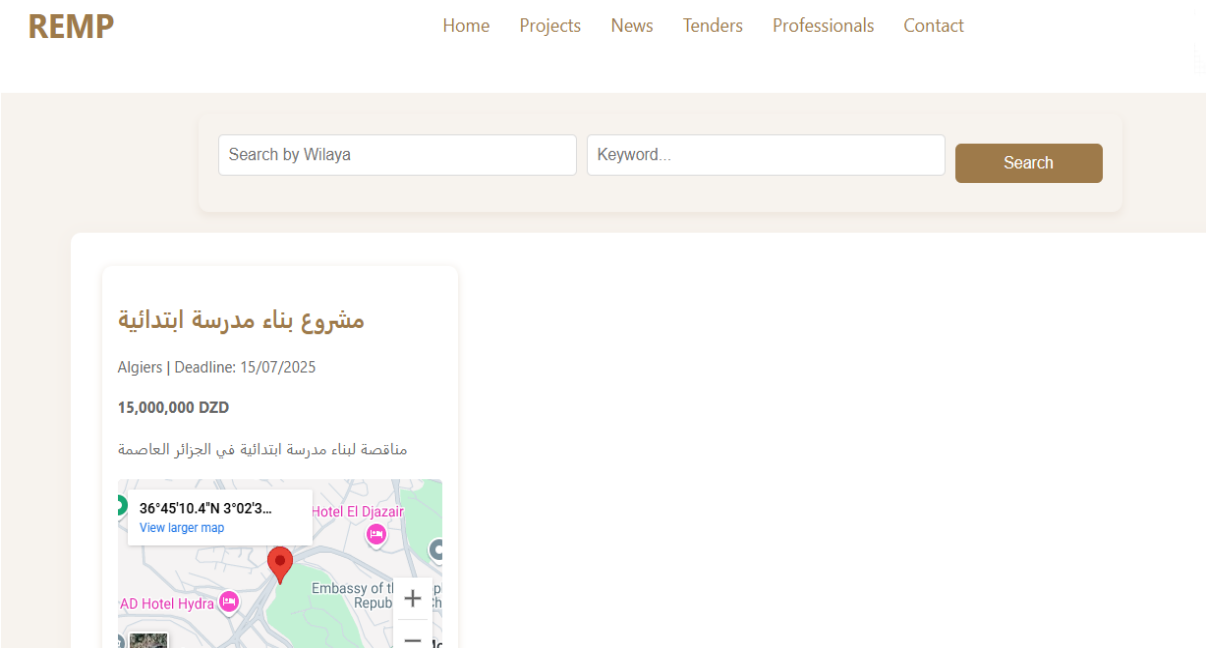
صورة 6.3. صفحة التواصل

هنا يرى المستخدم معلومات صاحب المنصة مع موقعه ويمكن ان يرسل له رسالة استفسار او غيرها .



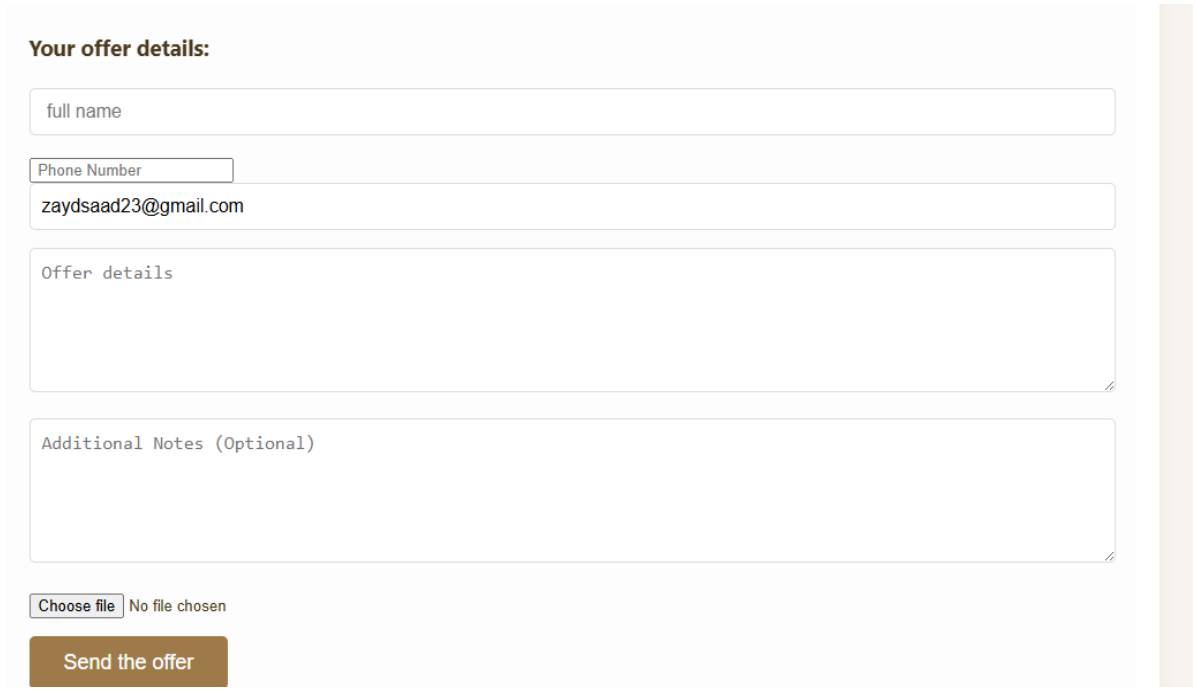
صورة 7.3. صفحة الملف الشخصي

هنا يمكن للمستخدم رؤية معلومات اي حساب بعد البحث عنه او تصفحه سواء كان لوكالة او مهندس او مقاول كم يتصفح مشروعاته ويستطيع ارسال رسالة اليه .



صورة 8.3. صفحة مشاريع المناقصة

هنا يرى المستخدم اي مشاريع خاصة بالمناقصة التي طرحتها الوكالات مع الموقع بالضبط و اخر اجل لها و الميزانية الخاصة بها .



صورة 9.3. تقديم العرض

بعد رؤية المشروع يضغط المهندس او المقاول على صفحة تقديم عرض ويسجل الدخول اولا ثم يرسل المعلومات الخاصة به بالاضافة لملف مضغوط فيه معلومات عرضه ويرسلها للوكالة صاحبة العرض .

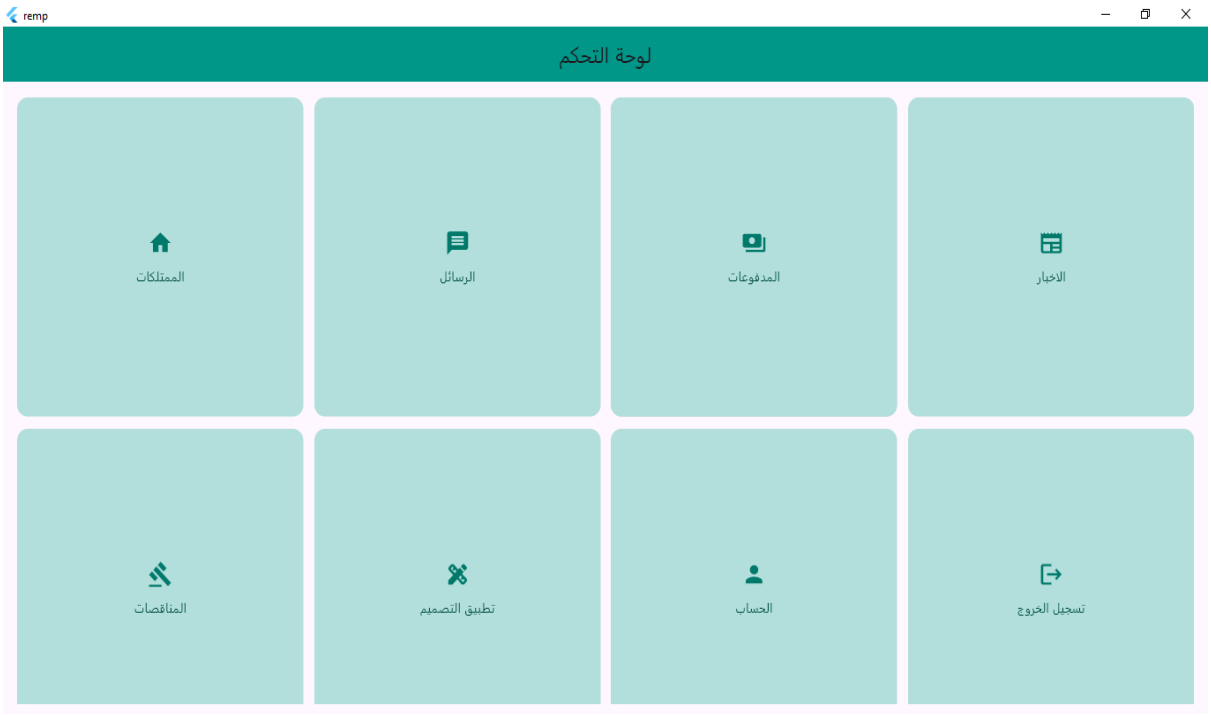
2.3.3. واجهات لوحة التحكم (flutter app) :

صورة 10.3. صفحة تسجيل الدخول

واجهة بسيطة لتسجيل الدخول خاصة باصحاب الحسابات (مقاول مهندس او وكالة)

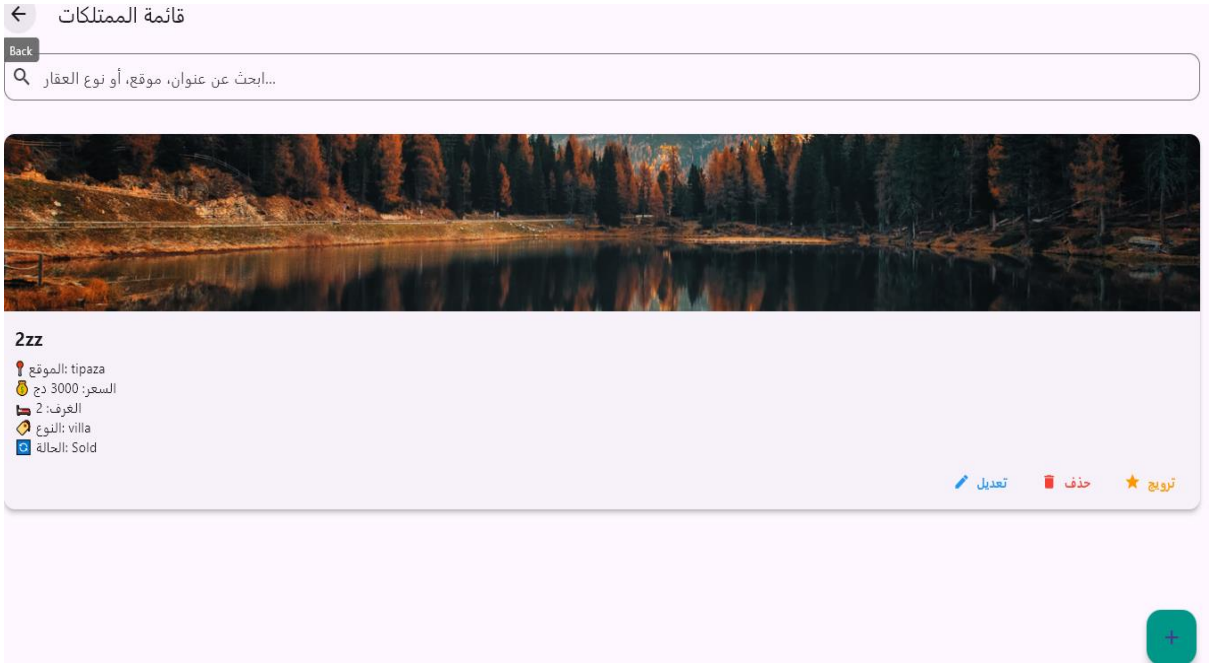
حيث تملك الوكالة جميع الميزات (ادارة الممتلكات ، الاخبار ،الرسائل ، ادارة المناقصات ، ادارة الحساب وغيرها) .

ويملك المقاول ايضا جميع الميزات عدى ادارة المناقصات كما يملك المهندس ايضا الميزات المذكورة سابقا عدى ادارة الممتلكات و المناقصات .



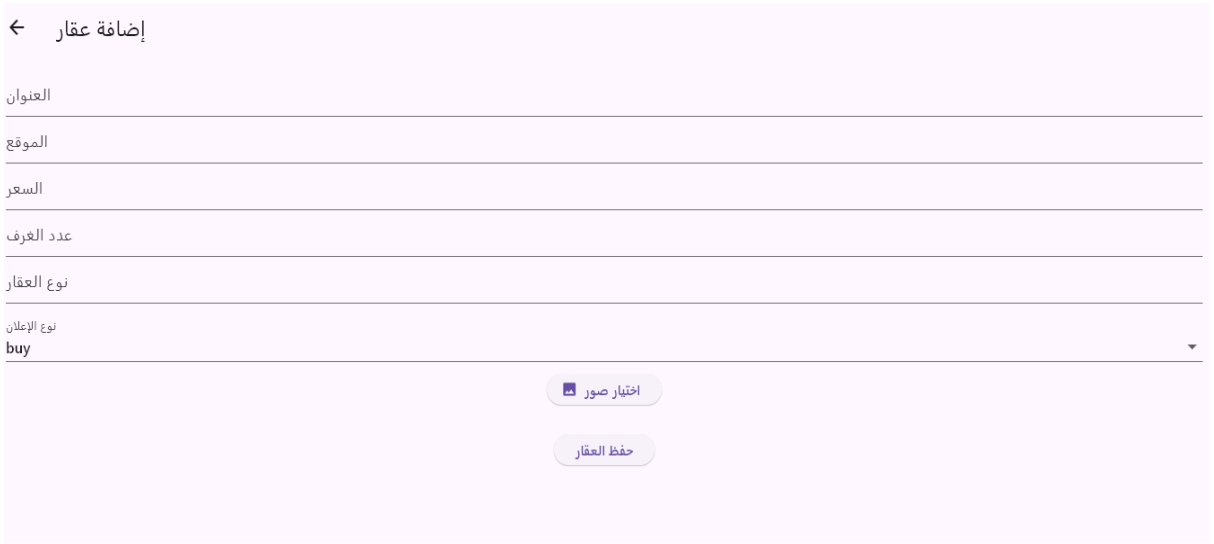
صورة 11.3. لوحة التحكم

لوحة التحكم مع واجهة سهلة للميزات الخاصة بكل مستخدم تعمل على جميع الانظمة windos android mac ios بسبب دعم ال flutter لاكثر من منصة بكود موحد.



صورة 12.3. صفحة ادارة الممتلكات او العقارات .

واجهة بسيطة مع ازرار واضحة وسهلة لاضافة او حذف او تعديل العقار او حتى الترويج له مع دعم بحث متقدم لاجل تسهيل عملية ادارة العقارات .

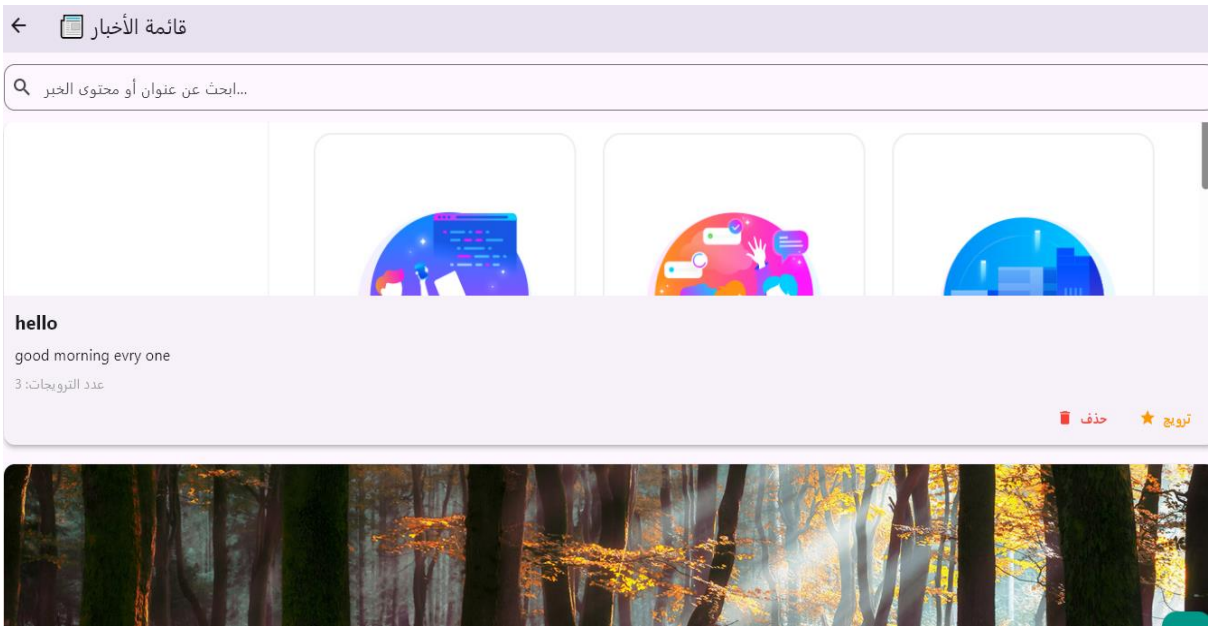


صورة 13.3. اضافة عقار

بعد ضغط زر الاضافة يمكن ان يضيف المستخدم عقار بادخال معلوماته ونوعه اذا كان للشراء او التاجير مع صورة او اكثر له .

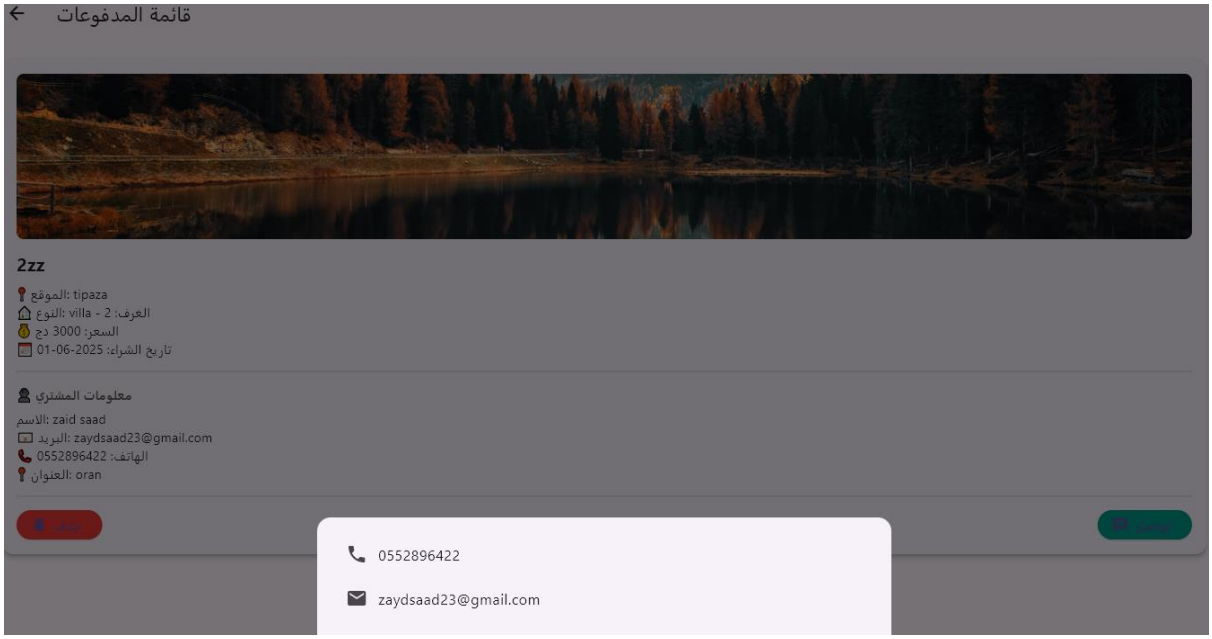
صورة 14.3. تعديل العقار

يمكن للمستخدم هنا ان يعدل على العقار بالاخص حالته اذا كان مباع ام متوفر ويضيف او يحذف الصور الخاصة به .



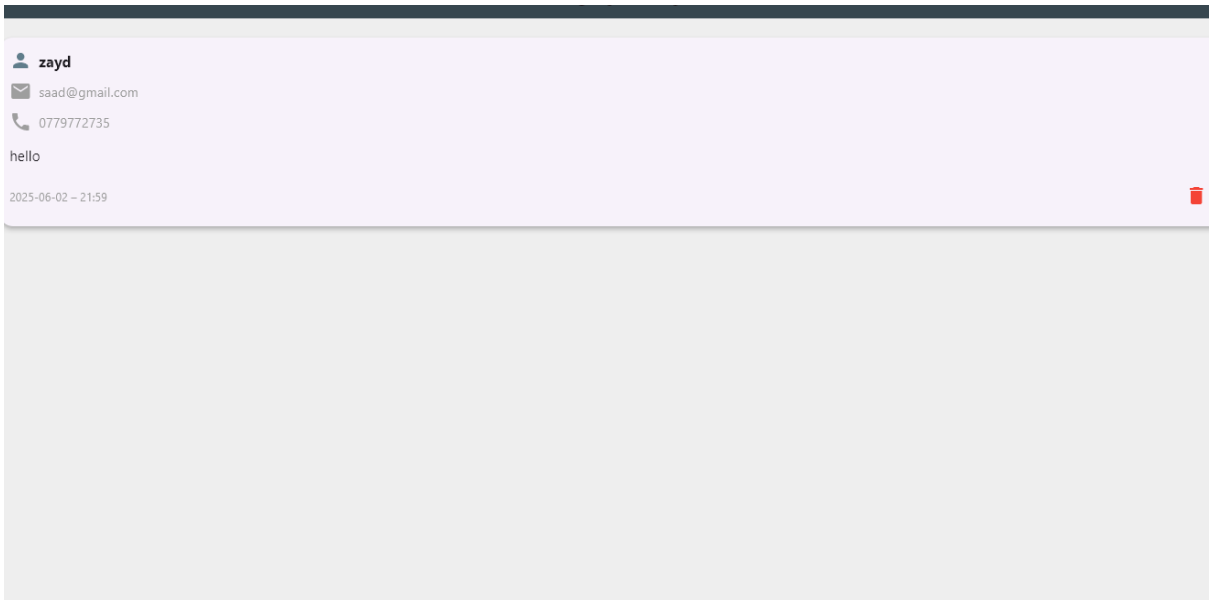
صورة 15.3. الاخبار

يمكن للمستخدم ان يضيف او يحذف او يروج لخبر ما كما لديه بحث متقدم للاخبار



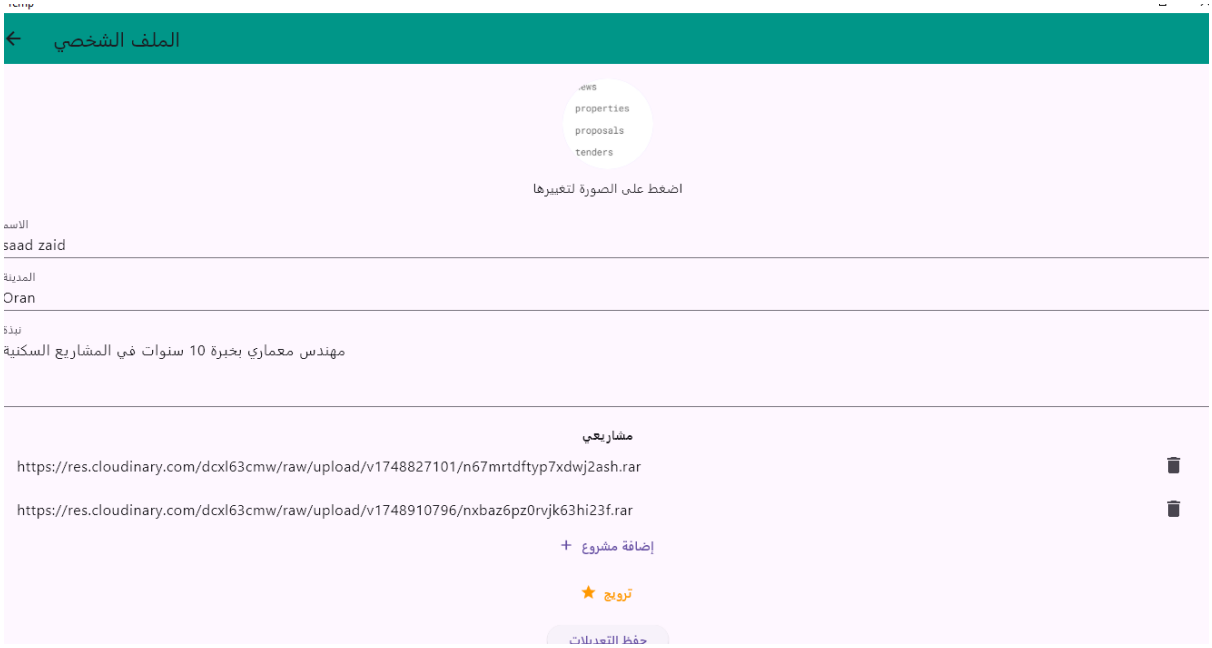
صورة 16.3. صفحة المدفوعات

هنا يرى المستخدم العقارات التي تم طلب شراءها ويرى معلومات المستخدم صاحب الطلب كم بمكنه الاتصال به او ارسال رسالة في الايميل له .



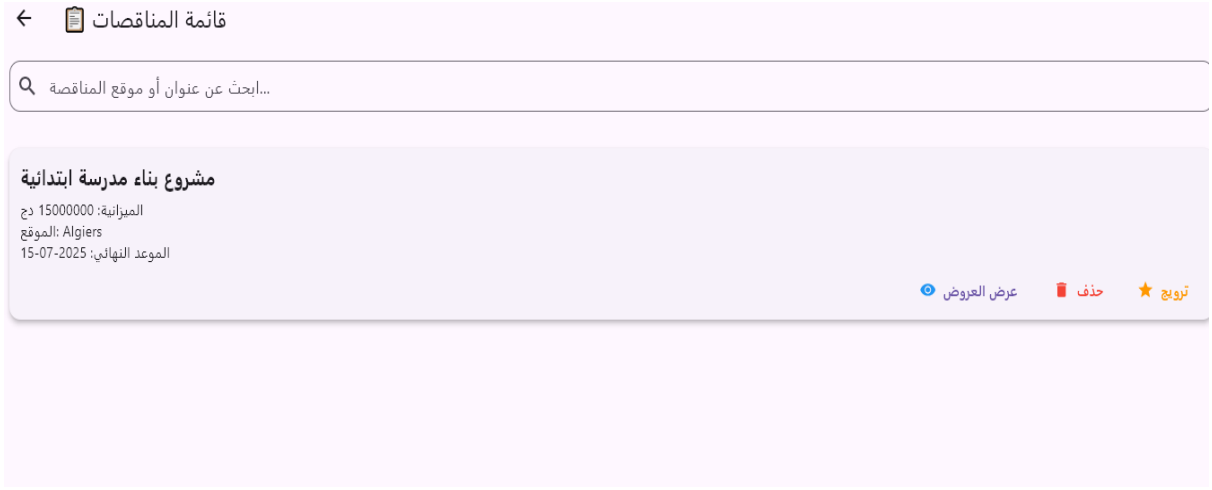
صورة 17.3. الرسائل

هنا يرى المستخدم الرسائل التي وصلته من قبل مستخدمي المنصة .



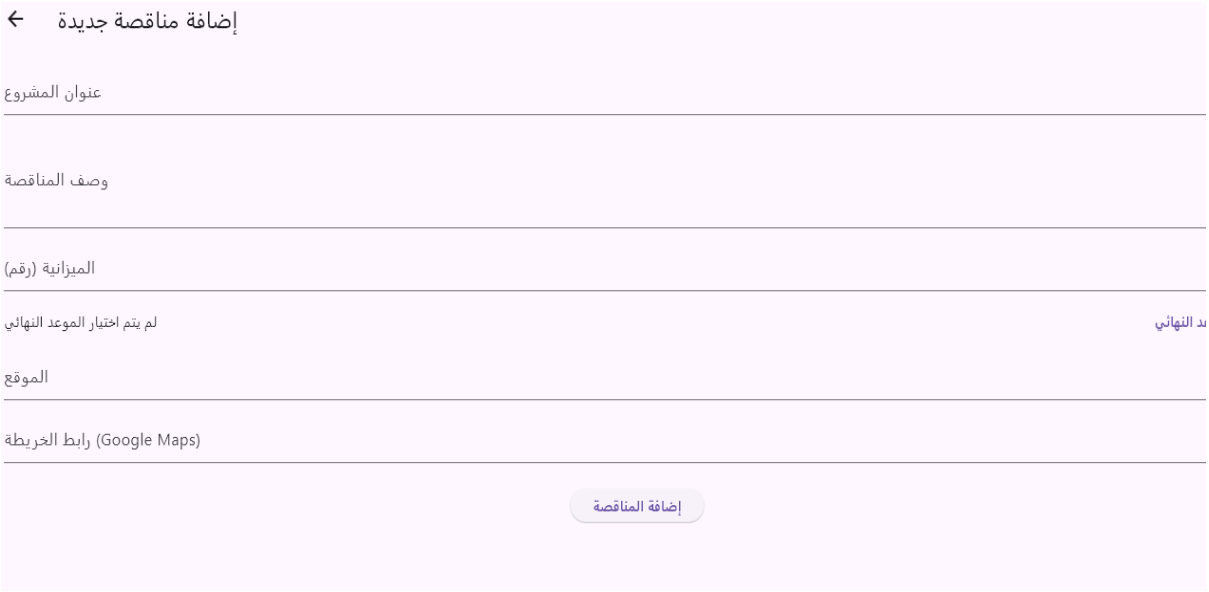
صورة 18.3. الملف الشخصي

هنا يمكن للمستخدم رؤية المعلومات الخاصة به (صورة شخصية ، الاسم ، الولاية ، الوصف ، المشاريع) وتعديلها كما يشاء او يرفع ملف لمشاريعه لاستعراض مهاراته وخبراته . كما يمكنه الترويج لملفه لزيادة فرص ظهوره.



صورة 19.3. المناقصات

هنا صفحة المناقصات (خاصة بالوكالة)، حيث يمكنها رفع مشروع للمناقصة او حذفه او الترويج له والبحث داخل مشاريعها



صورة 20.3. اضافة مشاريع المناقصة

تقوم الوكالة باضافة مشروع من خلال ادخال معلومات المشروع والميزانية ورابط للخريطة عن مكان المشروع واخر وقت لتقديم العروض .

صورة 21.3. عروض المناقصة

يمكن للوكالة رؤية العروض على شكل ملفات مضغوطة ثم اختيار العرض الافضل والاتصال بصاحب العرض ثم حذف المناقصة من المنصة .

4.3. قواعد البيانات :

1.4.3. قواعد البيانات NoSQL :

"يُستخدم مصطلح "قاعدة بيانات NoSQL" عادةً للإشارة إلى أي قاعدة بيانات غير علائقية. يرى البعض أن مصطلح "NoSQL" يعني "غير SQL"، بينما يرى آخرون أنه يعني "ليس SQL فقط". مع ذلك، يتفق معظم الناس على أن قواعد بيانات NoSQL تخزن البيانات بطريقة أكثر طبيعية ومرونة. بخلاف SQL، تُعدّ NoSQL نهجًا لإدارة قواعد البيانات، بينما SQL هي ببساطة لغة استعلام، تُشبه لغات الاستعلام في قواعد بيانات NoSQL.

تخزن البيانات بشكل مختلف عن الجداول العلائقية. تتوفر قواعد بيانات NoSQL بأنواع مختلفة حسب نموذج بياناتها، وأهمها: المستند، والمفتاح-القيمة، والعمودي، والرسومي. توفر هذه القواعد مخططات مرنة، وتتسع بسهولة لتستوعب كميات كبيرة من البيانات الضخمة وأحمال المستخدمين العالية."[19]

2.4.3 أنواع قواعد NoSQL :

مع مرور الوقت، ظهرت أربعة أنواع رئيسية من قواعد بيانات NoSQL:

1_ قواعد البيانات المستندة إلى المستندات (Document-Based Databases) :

تخزن قواعد البيانات المستندة إلى المستندات البيانات في المستندات بطريقة مشابهة لكائنات JavaScript Object Notation (JSON). يحتوي كل مستند على أزواج من الحقول والقيم. يمكن أن تكون القيم من أنواع مختلفة، بما في ذلك السلاسل النصية، والأرقام، والقيم المنطقية، والمصفوفات، أو غيرها من الكائنات. توفر قواعد بيانات المستندات نموذج بيانات مرنًا، مثاليًا لمجموعات البيانات شبه المنظمة وغير المنظمة عمومًا. كما أنها تدعم الهياكل المتداخلة، مما يسهل تمثيل العلاقات المعقدة أو البيانات الهرمية. Firebase مثال على ذلك .

2_مخزن المفتاح-القيمة (Key-Value Databases) :

هو نوع أبسط من قواعد البيانات، حيث يحتوي كل عنصر على مفاتيح وقيم. كل مفتاح فريد ومرتبط بقيمة واحدة. تُستخدم هذه القواعد للتخزين المؤقت وإدارة الجلسات، وتوفر أداءً عاليًا في القراءة والكتابة لأنها عادةً ما تخزن العناصر في الذاكرة. Amazon DynamoDB وRedis مثالان على ذلك.

3_المخازن ذات الأعمدة (Column-Family Databases) :

تخزن المخازن ذات الأعمدة البيانات في جداول ديناميكية، وصفوف، وأعمدة. تُخزَّن البيانات في جداول. ومع ذلك، بخلاف قواعد بيانات SQL التقليدية، تتميز المخازن ذات الأعمدة بالمرونة، حيث يمكن أن تحتوي الصفوف المختلفة على مجموعات مختلفة من الأعمدة. تستخدم قواعد البيانات هذه تقنيات ضغط الأعمدة لتقليل مساحة

التخزين وتحسين الأداء. تتيح أحجام الصفوف والأعمدة الكبيرة استرجاعًا فعالًا للبيانات الكبيرة والمتباينة. يُعد كل من Apache Cassandra وHBase مثالين على المخازن ذات الأعمدة.

4_قواعد بيانات الرسوم البيانية (Graph Databases) :

تخزن قواعد بيانات الرسوم البيانية البيانات على شكل عقد وحواف. عادةً ما تخزن العقد معلومات حول الأشخاص والأماكن والكائنات (مثل الأسماء)، بينما تخزن الحواف معلومات حول العلاقات بين العقد. تعمل هذه الآلية بشكل جيد مع البيانات عالية الترابط، حيث لا تكون العلاقات أو الأنماط واضحة دائمًا.

من الأمثلة على ذلك قواعد بيانات الرسوم البيانية Neo4J وAmazon Neptune. كما توفر MongoDB إمكانيات مسح الرسوم البيانية باستخدام خطوة $graphLookup من مسار التجميع.

3.4.3. سبب اختيار Firebase

Firebaseعبارة عن نظام قواعد بيانات سحابي من نوع NoSQL وثائقي(Document-Based) ، حيث يتم تخزين البيانات داخل وثائق (Documents) ضمن مجموعات(Collections) ، بدلًا من الجداول والعلاقات التقليدية الموجودة في قواعد البيانات العلائقية.

السبب وراء اختيارFirebase :

1_التكامل الممتاز معFlutter :

بما أن واجهات لوحة التحكم طُورت باستخدامFlutter ، كان من المنطقي الاعتماد علىFirebase ، كونه يتكامل بسلاسة مع هذا الإطار، ويُسهّل ربط الواجهة الأمامية بقاعدة البيانات دون الحاجة إلى إعدادات معقدة.

2_الدعم للتحديث الفوري(Realtime) :

واحدة من أبرز ميزات Firebase هي قدرته على مزامنة البيانات بين جميع المستخدمين في الوقت الحقيقي، مما يسمح ببناء ميزات تفاعلية مثل الرسائل، متابعة العروض الجديدة، أو تحديثات العقارات بشكل لحظي.

3_سهولة التوسع والإدارة:

بما أن Firebase يعمل ضمن بيئة سحابية بالكامل، لا يحتاج المطور لإدارة الخوادم أو إعدادات معقدة، وهو ما يوفر وقتًا وجهدًا كبيرين أثناء التطوير والصيانة.

4_نظام الأمان المدمج:

من خلالFirebase Rules ، يمكن تحديد من يمكنه الوصول إلى أي جزء من البيانات بناءً على دوره (وكالة، مهندس، مستخدم عادي...)، مما يضيف طبقة أمان مهمة لحماية المعلومات.

4.4.3.من مخطط الصنف إلى نموذج Firebase :

في هذا المشروع، لم يتم اعتماد قاعدة بيانات علائقية (SQL) تقليدية تحتوي على جداول ومفاتيح أجنبية، بل تم تصميم الهيكل العام للبيانات انطلاقًا من مخطط الصنف (Class Diagram) بلغةUML ، والذي يُمثّل الكيانات الرئيسية في النظام والعلاقات بينها، مثل User, Property, Tender, Proposal, Message, :وغيرها.

انطلاقًا من هذا النموذج المفاهيمي، تم تحويل كل (Class) إلى مجموعة (Collection) في قاعدة بيانات Cloud Firestore التابعة لFirebase ، والتي تعتمد على النموذج الوثائقي (Document-Based Model) ضمن بيئة NoSQL.

تم اتباع عدد من القواعد والممارسات، وهي:

1_تحويل كل Class إلى Collection مستقلة:

مثلًا، User إلى Collection باسمusers ، Tender إلى tenders ، وهكذا.

2_تخزين خصائص الكائن كحقول داخل الوثيقة:

كل خاصية معرفة في الكلاس مثل(name, role, city) تمثل مفتاحًا داخل مستند JSON داخل.Firestore

3_تمثيل العلاقات المرجعية باستخدام معرفات:(IDs)

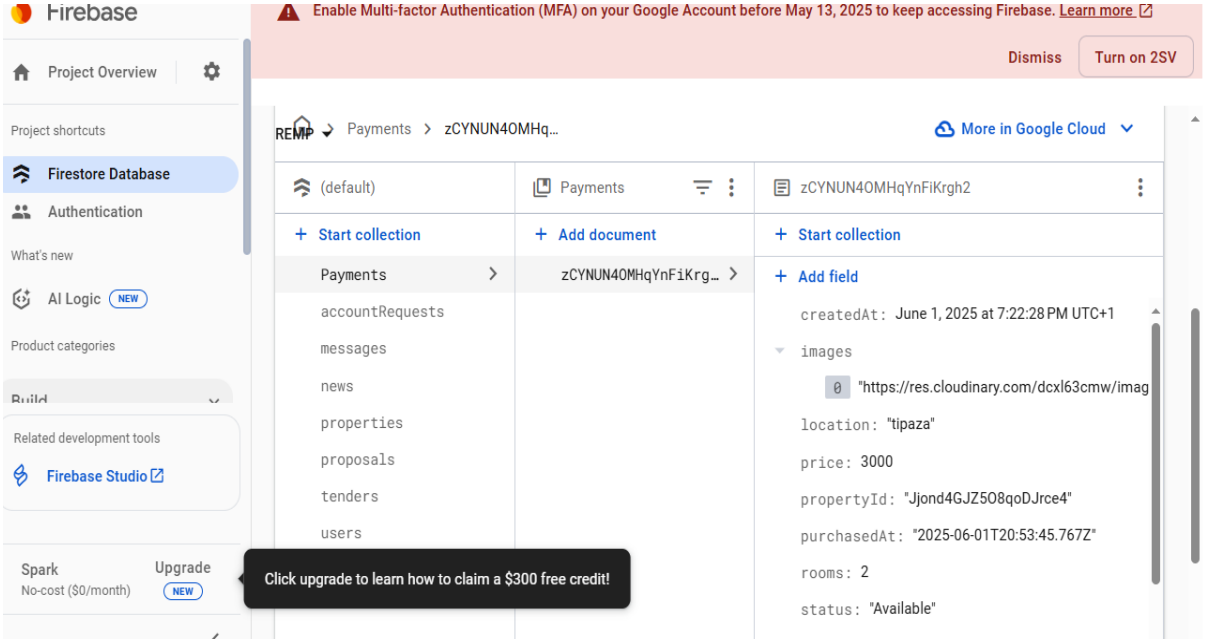
بدلًا من استخدام علاقات مباشرة كما فيSQL ، يتم تضمين معرف userId داخل مستند Property أو Tender للدلالة على المالك أو الجهة المرتبطة.

4_استخدام التكرار الجزئي للبيانات:(Denormalization)

في بعض الحالات، تم نسخ بعض البيانات الأساسية مثل اسم المستخدم أو بريده داخل مستندات أخرى مثل (Proposal) لتسهيل عمليات القراءة وتقليل التعقيد في الاستعلام.

5_تنظيم الوثائق بطريقة مسطحة:(Flat Structure)

تم تجنب التداخل العميق للبيانات داخل المستندات، والاعتماد على مجموعات واضحة ومستقلة لكل نوع من البيانات.



صورة 22.3. قاعدة بيانات firestore

5.4.3 قواعد الأمان في Firebase (Firebase Rules) :

توفر Firebase آلية فعّالة تُعرف بـ Firebase Security Rules، وهي عبارة عن مجموعة من القواعد تُستخدم لتحديد من يمكنه قراءة أو تعديل البيانات المخزّنة في Firestore .

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /users/{userId} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.uid == userId;
    }
    match /properties/{propertyId} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.uid == request.resource.data.userId;
      allow update: if request.resource.data.diff(resource.data).affectedKeys().hasOnly(['status']);
    }

    match /news/{newsId} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.uid == request.resource.data.userId;
    }

    match /accountRequests/{requestId} {
      allow create: if true;
      allow read, update, delete: if false;
    }
```

```
  match /messages/{messageId} {
    allow create: if true;
    allow read, delete: if request.auth != null && request.auth.uid == resource.data.toUserId ;
  }

  match /payments/{paymentId} {
    allow create: if true;
    allow read, delete: if request.auth != null && request.auth.uid == resource.data.userId;
  }
 match /tenders/{tenderId} {
    allow read: if true;
    allow write: if request.auth != null && request.auth.uid == request.resource.data.userId ;
  }

match /proposals/{proposalId} {
    allow create: if request.auth != null;
    allow read, update, delete: if request.auth != null && request.auth.uid == resource.data.userId;
  }

  match /contact/{contactId} {
    allow create: if true;
    allow read, update, delete: if false;
```

صورة 23.3. قواعد الأمان (Firebase Rules)

تعتمد قاعدة بيانات Firestore في هذا المشروع على قواعد أمان تتحكم في من يمكنه قراءة أو كتابة البيانات، حسب نوع المجموعة وصلاحيات المستخدم.

فيما يلي شرح مبسط لكل مجموعة:

1. مجموعة المستخدمين(users)

القراءة: مسموحة للجميع.

الكتابة: مسموحة فقط لصاحب الحساب (المستخدم يكتب بياناته فقط).

2. مجموعة العقارات(properties)

القراءة: مسموحة للجميع.

الكتابة: مسموحة فقط لصاحب العقار (المستخدم الذي يملك العقار).

التعديل: مسموح فقط لتغيير حالة العقار) حقل status فقط.(

3. مجموعة الأخبار(news)

القراءة: مسموحة للجميع.

الكتابة: مسموحة فقط لصاحب الخبر (الناشر الأصلي).

4. مجموعة طلبات الحساب الاحترافي(accountRequests)

الإنشاء: مسموح لأي مستخدم.

القراءة والتعديل والحذف: غير مسموحة داخل التطبيق، وتُدار من خلال لوحة خارجية.

5 .مجموعة الرسائل(messages)

الإرسال (الإنشاء): متاح لأي شخص.

القراءة والحذف: مسموحة فقط للمرسل.

6 .مجموعة المدفوعات(payments)

الإنشاء: متاح لأي مستخدم.

القراءة والحذف: مسموحة فقط لصاحب عملية الدفع.

7 .مجموعة المناقصات(tenders)

القراءة: متاحة للجميع.

الكتابة: مسموحة فقط للوكالة المالكة للمناقصة.

8 .مجموعة العروض على المناقصات(proposals)

الإنشاء: مسموح لأي مستخدم مسجل.

القراءة والتعديل والحذف: مسموحة فقط لصاحب العرض.

9 .مجموعة نموذج اتصل بنا(contact)

الإنشاء: متاح للجميع.

القراءة والتعديل والحذف: غير مسموحة، وتُدار من جهة خارجية.

5.3. خاتمة :

وفي الختام، قمنا في هذا الفصل بشرح جميع الأدوات والتقنيات التي تم استخدامها في تطوير المنصة، سواء من حيث البرمجيات أو العتاد أو الخدمات السحابية، مثل Firebase وCloudinary. كما استعرضنا الواجهات الرئيسية للمنصة، بالإضافة إلى لوحة التحكم الخاصة بالمستخدمين، موضحين آلية تصميمها وطريقة تفاعل المستخدم معها.

الخاتمة

في ختام هذا العمل، نكون قد قدّمنا مقاربة شاملة لواقع القطاع العقاري في الجزائر، مع تسليط الضوء على أبرز التحديات التي يعاني منها، خاصة تلك المرتبطة بنقص الرقمنة وضعف التنسيق بين مختلف الفاعلين. وقد سعينا من خلال هذه الدراسة إلى تقديم حل رقمي عملي يتمثل في منصة عقارية رقمية تهدف إلى تسهيل التفاعل بين الوكالات العقارية، المهندسين، المقاولين، والمستخدمين، بما يساهم في تحسين أداء هذا القطاع الحيوي.

كما استعرضنا في هذا المشروع أهم المخططات التي توضح العمليات داخل المنصة، وأدوار المستخدمين، وهيكلة قاعدة البيانات، بالإضافة إلى كيفية تسيير المنصة والتفاعلات التي تتم داخلها.

وقد تطرّقنا أيضًا إلى الجانب التقني من خلال عرض الأدوات والتقنيات التي تم استخدامها، مثلFlutter ، وFirebase، وCloudinary، مع توضيح كيفية توظيف هذه الأدوات لتحقيق نظام رقمي متكامل، سهل الاستخدام، وفعّال من حيث الأداء.

كما تم تقديم نظرة تفصيلية على المنصة المطوّرة، بما في ذلك واجهات الاستخدام ولوحة التحكم الخاصة بالمستخدمين.

نتمنى أن يُشكّل هذا العمل خطوة أولى نحو تطوير حلول مبتكرة تساهم في رقمنة القطاع العقاري بشكل أوسع، وتفتح المجال أمام مبادرات مستقبلية توظّف التكنولوجيا في خدمة الاقتصاد الوطني وتحسين جودة الخدمات.

المراجع :

[1] القانون المدني الجزائري، المادة 683، الأمر رقم 75-58 المؤرخ في 26 سبتمبر 1975، الجريدة الرسمية للجمهورية الجزائرية، عدد خاص.

[2] Real Estate Market | New Western (دون تاريخ). تم الاسترجاع في 3 يونيو 2025

[3] القطاع العقاريnnovative Partnerships for Sustainable Development – (دون تاريخ). القطاع العقاري. تم الاسترجاع في 5 جوان 2025

[4] What is Digitization, Digitalization, and Digital Transformation | ARC Advisory في 24 مارس 2020

[5] https://journals.sagepub.com/ في 2018

[6] https://www.djazairess.com/ في 2018

[7] تحليل شامل لسوق العقارات في الجزائر 2024/2023 - الاتجاهات والإحصائيات الحديثة Lamacta | Immobilier في 2024/2023

[8] https://www.aps.dz/ في , 07 جانفي 2025

[9] ما هي أنواع العقارات؟ وكيف تختار العقار المناسب لك؟ | مدونة عقارماب في 8 جوان 2021

[10] What is UML | Unified Modeling Language في جويلية 2005

[11] What is StarUML - StarUML Documentation (تاريخ غير موجود) تم الاسترجاع في 5 جوان 2025

[12] Open-source tool that uses simple textual descriptions to draw beautiful UML diagrams. في جوان 2005

[13] https://www.lucidchart.com/ (تاريخ غير موجود) تم الاسترجاع في 5 جوان 2025

[14] https://learn.microsoft.com/en-us/shows/visual-studio-code/ (تاريخ غير موجود) تم الاسترجاع في 5 جوان 2025

[15] Flutter architectural overview | Flutter في *2025-05-19*

[16] https://developer.mozilla.org/ في *2025-05-14*

[17] What is Google Firebase? في *2024-05-19*

[18] Cloudinary Frequently Asked Questions (تاريخ غير موجود) تم الاسترجاع في 5 جوان 2025

[19] Qu'est-ce Que NoSQL ? Tout Savoir Sur Les Bases De Données NoSQL | MongoDB (تاريخ غير موجود) تم الاسترجاع في 5 جوان 2025

ملخص :

في ظل التحولات الرقمية المتسارعة، أصبح من الضروري رقمنة القطاعات الحيوية لتحقيق فعالية أكبر وتسهيل المعاملات. يهدف هذا المشروع إلى دراسة واقع القطاع العقاري في الجزائر، وتسليط الضوء على أبرز التحديات التي تواجهه، مثل ضعف التنسيق بين الفاعلين، نقص الشفافية، وصعوبة الوصول إلى المعلومات والخدمات.

بناءً على ذلك، تم اقتراح منصة رقمية متكاملة تجمع بين الوكالات العقارية، المهندسين، المقاولين، والمستخدمين، من أجل تحسين التفاعل بينهم وتسهيل عمليات البيع، الشراء، والإيجار، بالإضافة إلى تنظيم المناقصات والعروض.

وقد تم اعتماد تقنيات حديثة في تطوير هذه المنصة، مثل: Flutter، Firebase، Cloudinary، وHTML وCSS وJavaScript، مع اعتماد بنية تعتمد على واجهات تفاعلية وتجربة مستخدم سهلة.

يساهم هذا المشروع في دعم التحول الرقمي للقطاع العقاري، ويمهّد الطريق نحو إدارة أكثر شفافية وكفاءة في هذا المجال.

Abstract

Amid rapid digital transformation, digitizing vital sectors has become essential to enhance efficiency and streamline transactions. This project aims to examine the current state of the real estate sector in Algeria and highlight its main challenges, such as weak coordination among stakeholders, lack of transparency, and difficulty accessing reliable information and services.

In response to these issues, an integrated digital platform has been proposed to connect real estate agencies, engineers, contractors, and users. The platform seeks to improve interaction among these actors and facilitate property sales, purchases, and rentals, as well as organize tenders and offers.

The platform was developed using modern technologies including Flutter, Firebase, Cloudinary, HTML, CSS, and JavaScript. It is designed with an interactive architecture to provide a seamless and user-friendly experience.

This project contributes to the digital transformation of the real estate sector and paves the way for more transparent and efficient management practices in the field.