

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

جامعة بوليتكنك فلسطين



كلية تكنولوجيا المعلومات و هندسة الحاسوب

علم حاسوب

مشروع تخرج :

مستكشف العقارات

عمل الطالبة :

ميسون إخليل

المشرف :

الدكتور نبيل عرمان

توقيع رئيس القسم:

.....

توقيع المشرف:

.....

الاهداء

الى والدي براً واحساناً

الى اخوتي حباً واعتزازاً

الى من اضاء ذاكرتي بفيض من علمه الدكتور نبيل عرمان

أهدي هذا البحث

مستكشف العقارات

الشكر والتقدير :

انقدم بجزيل الشكر الى جميع أعضاء الهيئة التدريسية في كلية تكنولوجيا المعلومات وهندسة الحاسوب، لما زودوه لنا من المعرفة والعلم خلال السنين.

كما اوجه الشكر والتقدير الى الدكتور الفاضل د.نبيل عرمان المشرف على هذا المشروع، لما قدمه لي من توجيهات ونصائح لإنجاز هذا العمل.

كل الشكر والاحترام والتقدير الى كل من ساهم في انجاز هذا العمل المتواضع.

ميسون اخليل

خلاصة

يواجه الباحثين عن عقارات صعوبة في ايجاد هذه العقارات من بيوت أو قطعة ارض بغرض الاستئجار أو الشراء، ناهيك عن التكلفة الاضافية التي يتحملونها بسبب الوسطاء، ولذلك نسعى الى حل المشكلة من خلال ايجاد طريقة سهلة لمساعدة الاشخاص على ايجاد العقارات المناسبة لهم ووصلهم باصحابها من اجل اتمام عملية البيع أو الشراء، لذلك سيتم بناء موقع الكتروني يوفر للباحثين عن عقارات امكانية البحث عن العقارات من خلال موقعها أو اسعارها واطهار النتيجة مباشرة على خرائط جوجل (Google Maps)، وايضا يوفر الموقع الالكتروني لأصحاب العقارات والوسطاء إمكانية عرض عقاراتهم للإيجار أو البيع مع امكانية التواصل بين الاشخاص وصاحب العقار من خلال نظام رسائل داخلي. إن سبب اختيار هذا المشروع هو وجود مشكلة لدى الاشخاص الراغبين بالبيع أو الشراء في عرض عقاراتهم التي يمتلكونها والاشخاص الراغبين بالشراء أو الاستئجار في البحث عن العقار المناسب .

Abstract

People searching for real estate encounter many difficulties in finding the appropriate properties to lease or buy. In addition, they may need to pay additional charges to intermediaries. To reduce these difficulties and charges, the project team decides to build a web application/site that makes it easier for people to find suitable properties. In addition, the web application may link these people directly to properties' owners to lease or buy. the project team decides to develop a web application that helps people finding proper properties based on the location and/or price by displaying results on Google maps directly. In addition, the web application will help properties' owners to post their properties to be viewed by visitors and communicate with them through website messaging, phone numbers and emails.

فهرس المحتويات

I.....	صفحة عنوان المشروع وتوقيع المشرفين ورئيس القسم
II.....	الإهداء
III.....	صفحة العنوان الرئيسية
IV.....	شكر وتقدير
V.....	ملخص المشروع
VII.....	جدول المحتويات
1.....	الفصل الأول
7.....	الفصل الثاني
12.....	الفصل الثالث
26.....	الفصل الرابع
56.....	الفصل الخامس
99.....	الفصل السادس
113.....	الفصل السابع
124.....	المصادر و المراجع

الفصل الأول: مقدمة

2.....	1.1 الفكرة العامة
3.....	1.2 وصف فكرة المشروع
3.....	1.3 أصل المشكلة
4.....	1.4 نطاق المشروع
5.....	1.5 وصف الأجزاء اللاحقة

الفصل الثاني: الدراسات السابقة

8.....	2.1 المقدمة
8.....	2.2 مشاريع سابقة
11.....	2.3 الاستنتاجات
11.....	2.4 طرق جمع المعلومات

الفصل الثالث: تخطيط النظام

13.....	3.1 المقدمة
13.....	3.2 مهام المشروع
15.....	3.3 المخاطر
16.....	3.4 تحليل المخاطر
19.....	3.5 تكلفة المشروع
19.....	3.5.1 تكلفة تطوير النظام
19.....	3.5.1.1 تكاليف المعدات
20.....	3.5.1.2 تكاليف البرمجيات
21.....	3.5.1.3 تكاليف فريق العمل

22.....	3.2.2 تكلفة تشغيل النظام.....
23.....	3.6 شبكة المهام.....
24.....	3.7 الجدول الزمني.....

الفصل الرابع: تحليل النظام

27.....	4.1 المقدمة.....
27.....	4.2 وصف المتطلبات.....
28.....	4.3 وظائف النظام.....
41.....	4.4 CRC(Class Responsibilities Collaboration) Modeling.....
45.....	4.5 Activity Diagram.....
45.....	4.5.1 مسؤول الموقع و الزبون والزائر.....
47.....	4.5.2 مستخدم الموقع(الزائر).....
47.....	4.5.3 مستخدم الموقع(الزبون).....
53.....	4.5.4 مسؤول الموقع.....
54.....	4.6 Class Hierarchies and System.....

الفصل الخامس: تصميم النظام

57.....	5.1 المقدمة.....
57.....	5.2 Object-Relational Model.....
59.....	5.3 State Behavioral Modeling.....
61.....	5.4 Subsystem Design (Tasks and Components).....
63.....	5.5 Class and Object Design.....
84.....	5.6 Interface Design (System Messages).....
84.....	5.6.1 Object Interface.....
87.....	5.6.2 تصميم صفحات المستخدم.....

الفصل السادس: تنفيذ النظام

99.....	تنفيذ النظام.....
---------	-------------------

الفصل السابع: اختبار النظام

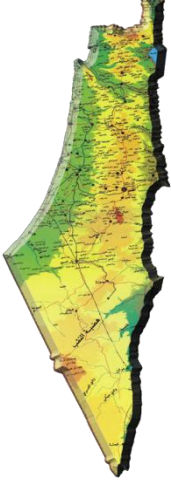
114.....	7.1 المقدمة.....
114.....	7.2 فحص العمليات.....
114.....	7.2.1 عمليات الزبون.....
118.....	7.2.2 عمليات الزائر.....

- شكل 2.2.1 (الموقع الإلكتروني شو بذك من فلسطين) 9.....
- شكل 2.2.2 (الموقع الإلكتروني عقار استيت) 10.....
- شكل 3.7.1 شبكة المهام 24.....
- شكل 3.7.2 الجدول الزمني لتنفيذ المهام 25.....
- شكل 4.3.1 use case diagram system 29.....
- بطاقة 4.4.A المستخدم 42.....
- بطاقة 4.4.B مجموعة المستخدمين 42.....
- بطاقة 4.4.C العقارات 43.....
- بطاقة 4.4.D المنزل 43.....
- بطاقة 4.4.E الارض 44.....
- بطاقة 4.4.F المدينة 44.....
- بطاقة 4.4.G نوع العقار 44.....
- بطاقة 4.4.H نوع الارض 44.....
- بطاقة 4.4.I الرسالة 45.....
- شكل 4.5.1.1 تصفح واستعراض العقارات باستخدام القائمة 46.....
- شكل 4.5.1.2 تصفح واستعراض العقارات باستخدام الخريطة 46.....
- شكل 4.5.1.3 البحث عن العقارات 46.....
- شكل 4.5.2.1 تسجيل حساب 47.....
- شكل 4.5.3.1 تسجيل دخول المستخدم الى الموقع 48.....
- شكل 4.5.3.2 تسجيل خروج المستخدم من الموقع 48.....
- شكل 4.5.3.3 عملية استرجاع كلمة المرور 49.....
- شكل 4.5.3.4 تعديل المستخدم لبيانات في الموقع 49.....
- شكل 4.5.3.5 اضافة قطعة أرض 50.....
- شكل 4.5.3.6 اضافة منزل 51.....
- شكل 4.5.3.7 حذف عقار 51.....
- شكل 4.5.3.8 تعديل معلومات العقار 52.....
- شكل 4.5.3.9 تصفح الرسائل وادارتها 52.....
- شكل 4.5.3.10 ارسال رسالة الى صاحب العقار 53.....
- شكل 4.5.3.11 الرد على رسالة 53.....
- شكل 4.5.4.1 عملية حذف مستخدم من الموقع 54.....
- شكل 4.5.4.2 اضافة مستخدم الى الموقع 54.....
- شكل 5.2.1 مخطط قاعدة البيانات 58.....
- شكل 5.3.1 تسجيل دخول مدير النظام 60.....
- شكل 5.3.2 عملية اضافة/حذف عقار 60.....
- شكل 5.3.3 تسجيل دخول الزبون 61.....
- شكل 5.4.1 نظام مستكشف العقارات 62.....
- شكل 5.4.1.1 حزمة العقارات 62.....
- شكل 5.4.1.2 حزمة مستخدم النظام 63.....
- شكل 5.6.1 تفاعل مسؤول الموقع مع النظام 85.....
- شكل 5.6.2 تفاعل الزبون مع النظام 86.....
- شكل 5.6.3 تفاعل الزائر مع النظام 87.....

16	جدول 3.3.1 المخاطر
16	جدول 3.4.1 مخاطر ناجمة عن فريق العمل
17	جدول 3.4.2 مخاطر ناجمة عن الادارة
17	جدول 3.4.3 مخاطر التسليم
18	جدول 3.4.2 مخاطر رفض النظام من قبل المواطنين
18	جدول 3.4.5 مخاطر ناجمة عن الاجهزة
19	جدول 3.4.6 مخاطر ناجمة عن عدم توفر خدمة الانترنت لجميع المواطنين
20	جدول 3.5.1.1 تكاليف المعدات
21	جدول 3.5.1.2 تكاليف البرمجيات
22	جدول 3.5.1.3 تكاليف تطوير النظام
22	جدول 3.5.2 تكاليف تشغيل النظام
23	جدول 3.6.1 جدول المهام
30	جدول (4.3.1) تصفح واستعرض العقارات باستخدام القائمة
30	جدول (4.3.2) تصفح واستعراض العقارات باستخدام الخريطة
31	جدول (4.3.3) البحث عن العقارات
32	جدول (4.3.4) تسجيل حساب
33	جدول (4.3.5) تسجيل الدخول الى الموقع
33	جدول (4.3.6) تسجيل الخروج من الموقع
34	جدول (4.3.7) استرجاع كلمة المرور
34	جدول (4.3.8) التعديل على البيانات
35	جدول (4.3.9) اضافة قطعة أرض
36	جدول (4.3.10) اضافة منزل
37	جدول (4.3.11) حذف عقار
37	جدول (4.3.12) تعديل معلومات العقار
38	جدول (4.3.13) تصفح الرسائل وادارتها
38	جدول (4.3.14) ارسال رسالة الى صاحب العقار
39	جدول (4.3.15) الرد على رسالة
40	جدول (4.3.16) حذف مستخدم من الموقع
40	جدول (4.3.17) اضافة مستخدم الى الموقع
114	جدول (7.2.1.1) فحص عمليات الزبون
118	جدول (7.2.2.1) فحص عمليات الزائر

الفصل الأول

المقدمة



- ❖ الفكرة العامة.
- ❖ وصف فكرة المشروع.
- ❖ أصل المشكلة.
- ❖ نطاق المشروع.
- ❖ وصف الأجزاء اللاحقة.

1.1 الفكرة العامة

شهدت التكنولوجيا خلال العقدين الماضيين نمواً متزايداً، فأدى ذلك إلى تغيير بعض الوسائل التقليدية إلى وسائل تكنولوجية حديثة، وقد امتدت التكنولوجيا إلى نواحي مختلفة من الحياة، تاركة خلفها سلسلة من التحولات الحياتية، لقد أصبح الاهتمام بالتكنولوجيا في مجتمعنا يتزايد ويأخذ ابعاداً أكثر عمقاً وشمولاً وأهمية وتأثيراً، ولكن بعد دراسة معمقة تم القيام بها، تم توصل إلى أنّ هناك قطاع مهم لم تصله هذه التكنولوجيا بشكل كبير، وهو القطاع التجاري من عمليات بيع وشراء للعقارات من خلال استخدام خرائط جوجل (Google Maps).

في الوضع الراهن، ان عملية البحث عن العقارات عملية مضيئة وتحتاج إلى وقت كبير، حيث يقوم الأشخاص بالبحث عن العقارات عن طريق التوجه إلى مالكي العقارات أو الوسطاء في المنطقة، فيذهبون إلى موقع العقار للتعرف عليه من حيث الموقع والسعر، فيكلفه ذلك الوقت والجهد والمال، وما تقدمه بعض المواقع الإلكترونية في عملية البحث عن العقارات غير كافية لوصف العقارات أو التعرف على ميزاتها بصورة دقيقة.

لذلك تم العمل على بناء موقع إلكتروني يوفر للمعنيين إمكانية البحث عن العقارات من خلال موقعها أو أسعارها وإظهار النتيجة مباشرة على خرائط جوجل (Google Maps)، كما يمكن مالكي العقارات من عرض عقاراتهم للإيجار أو البيع.

1.2 وصف فكرة المشروع

تقوم فكرة المشروع على إنشاء موقع إلكتروني يوفر للمعنيين امكانية البحث عن العقارات من خلال موقعها او اسعارها واطهار النتيجة مباشرة على خرائط جوجل (Google Maps)، وايضا اظهار النتائج على القائمة، حيث تظهر العقارات في القائمة مرتبه من أقرب عقار الى ابعد عقار عن موقعك او اي موقع يتم اختيار البحث بالنسبة له، وايضا يوفر الموقع الالكتروني لأصحاب العقارات والوسطاء إمكانية عرض عقاراتهم للإيجار أو البيع مع امكانية التواصل بين الاشخاص وصاحب العقار من خلال نظام رسائل داخلي .

1.3 أصل المشكلة

نظرا لما يعانيه المواطنون في فلسطين من صعوبة في ايجاد بيوت أو قطع اراض بغرض الاستئجار او الشراء، ناهيك عن التكلفة الاضافية التي يتحملونها بسبب الوسطاء، و صعوبة تنقل المواطنين بين المناطق للتعرف على العقارات المناسبة لهم من حيث موقعها و اسعارها، نظراً لهذه الأسباب التي هي حقاً مشاكل واقعية، وموجودة حالياً و تحتاج الى حلول أنسب، تم النفاخي بطريقة عملية لتخفيف هذا العبء عن المواطنين، وأولئك الذين يملكون هذه العقارات. حيث تم التوصل إلى فكرة تصميم موقع إلكتروني يوفر للمستخدمين طريقة سهلة للبحث عن عقارات معروضة من خلال الموقع على خرائط جوجل (Google Maps) كحل لهذه المشاكل.

1.4 نطاق المشروع

ان فكرة المشروع تستهدف المواطنين الفلسطينيين الذين يبحثون عن عقارات بغرض الاستئجار او الشراء، وأولئك الذين يملكون هذه العقارات لذلك هذا المشروع موجه الى كافة المواطنين ليستفيد منه كل شخص يملك عقار او يود البحث عن العقار المناسب. كما أن النظام الذي سيتم تطويره يمكن أن يعاد استخدامه وتكييفه لأي (مواطنين اخرين).

ان المدخلات لهذا النظام تتمثل في الاشخاص الذين يودون عرض عقاراتهم للبيع او الاستئجار والذين يودون الشراء أو الاستئجار، معلومات عن صاحب العقار و عن من يودون شراء أو استئجار العقارات، معلومات عن العقارات من حيث المساحة والمكان وسعرها، ومخرجات النظام تتمثل في الحصول على فرصة لعرض العقار على الموقع بالنسبة لمن يريد عرض عقاره للبيع أو الاستئجار، والحصول على معلومات عن العقارات وأماكن توفرها ومساحتها، وأسعارها بالنسبة لمن يريد الشراء .

يمكن تلخيص وظائف المشروع الأساسية التي سيتم اجرائها كما يلي:

- البحث عن العقارات بهدف الشراء او الاستئجار من خلال الموقع.
- عرض العقارات بهدف البيع او الاستئجار على الموقع.
- التواصل مع صاحب العقار ومع من يودون شراء او استئجار العقارات من خلال نظام رسائل داخلي في الموقع.
- عرض نتائج البحث على القائمة او خرائط جوجل.

1.5 وصف الأجزاء اللاحقة

فيما يلي سيتم عرض للفصول المختلفة التي سيتكون منها التقرير، وهي عبارة عن أربعة فصول:

• الفصل الأول

يعرض فكرة عامة عن المشروع، ووصف لما يحويه، مبينا أهدافه، وسبب اختيار الفريق لهذا المشروع ونطاق المشروع، وأهميته وفوائده للمواطنين .

• الفصل الثاني

عرض دراسات لمشاريع سابقة قامت بعمل مشابه له، والنتائج التي تم التوصل إليها من خلال تلك الدراسات، وعرض لما تم انجازه سابقا، وأهم ما يميز مشروعنا عن مشروعهم.

• الفصل الثالث

هو مرحلة التخطيط للمشروع، حيث يتم عرض توضيحي لوظائف المشروع، والمتطلبات التي يجب توافرها في النظام، وحساب التكاليف المتوقعة للمشروع والمتعلقة بالتكلفة التشغيلية والتطويرية، وعرض المخاطر التي يمكن الوقوع فيها خلال مرحلة تنفيذنا للمشروع.

• الفصل الرابع

مرحلة تحليل النظام، وهي مرحلة الوصف التفصيلي لوظائف النظام، بحيث يتم عرض تفصيلي لكل مرحلة ولكل عملية يتوجب توافرها بالمشروع، حيث يتم عرض الوظائف والخطوات التي سيتم العمل عليها، كما سيتم شرح متطلبات المشروع لكل جزء من أجزائه، والعمليات التي يقوم بتنفيذها.

• الفصل الخامس

سيتم في هذا الفصل توضيح ووصف تصميم النظام من حيث التصميم الوظيفي لكل جزء من اجزاء النظام، ويحوي هذا الفصل أيضا على نموذج الأصناف، وتصميم واجهات النظام و وصف قاعدة بيانات النظام.

• الفصل السادس

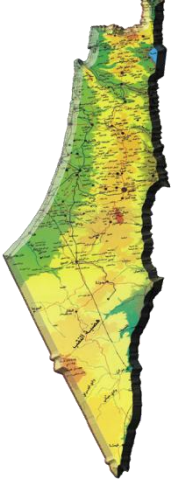
سيتم في هذا الفصل كتابة الكود الخاص بلبرنامج .

• الفصل السابع

سيتم في هذا الفصل إجراء عدة اختبارات على النظام ، تشمل اختبار أحد مكونات البرنامج واختبار مجموعة من مكونات البرنامج، بالإضافة إلى اختبار النظام بشكل كامل .

الفصل الثاني

الدراسات السابقة



- ❖ المقدمة.
- ❖ مشاريع سابقة.
- ❖ الاستنتاجات.
- ❖ طرق جمع المعلومات.

2.1 المقدمة

إن الدراسات السابقة من أهم المراحل التي يمر بها الباحث، حيث أنها تساعد الباحثين في إعداد دراساتهم، وفهم موضوع البحث، كما أنها توجه الباحث الى طرق أخرى لم يتم التطرق اليها في المشاريع السابقة، ويتم تجنب الاخطاء التي واجهت هذه المشاريع.

في هذا الفصل سيتم عرض بعض المشاريع السابقة التي تتشابه في الفكرة ولكنها تختلف في طريقة تنفيذها.

2.2 مشاريع سابقة

❖ الموقع الالكتروني شو بدك من فلسطين www.shobiddak.com

هو عبارة عن موقع شامل لكل شئ يخص البيع والشراء، ويحتوي على ستة اقسام رئيسية وهي:

1. الإعلانات العامة
2. الوظائف الشاغرة
3. معرض السيارات
4. دليل الشقق والمنازل
5. دليل الاراضي
6. دليل المحلات والمخازن

وهذه الاقسام يمكن اضافة الاعلان فيها مجاناً، وكما انه يحتوي على اشياء اضافية مثل المناهج الفلسطينية

و استمع الى الاذاعات وشاهد الفضائيات والطقس والعملات، عند عملية البحث عن العقار يتم عرض

النتائج في جداول كما يوضح شكل (2.2.1).

شبكة
www.shabaka.com
شو بديك من فلسطين

الرئيسية | اصف اعلانك مجاناً | خدمة الرسائل القصيرة | لتعلم المزيد من نحن | إسرائيل | تسجيل للدخول

128 | 10 | 10 | 1.5 | 1.5 | 1.5

رصيدك مفلس؟
تذكرة "جول" بتفديتك إذا فلس رصيدك!

شغل فلسطيني ناعم | الشام | سورماركت العاتك لكل شيء شهري

خدمات عامة

تاريخ الاصل	المنطقة	النوع	معلومات الاصل
2012-10-02	الخليل	عقارات	عروض مميزة للمساكن الخاصة لدى عقار مول
2012-10-03	رام الله والبيرة	حلوليات ولوازمها	جالكسي إس 2 = 1800 شغل مستعمل لأسبوع
2012-10-03	جنين	خدمات جامعية	ملازم لطلبة جامعة القدس المفتوحة
2012-10-02	رام الله والبيرة	دورات تعليمية	فرصة ذهبية لطلبة التوجيه دورات مجاناً 100%
2012-10-02	طولكرم	اتصالات	للبيع 0569-30-70-10
2012-10-03	رام الله والبيرة	خدمات عامة	شركة المصري للمساكن العامة

المزيد من الاعلانات

شغل فلسطيني ناعم | شغل فلسطيني نظيف | الإعلانات العامة | فرقناك لشايف | معرض السيارات | دليل الشقق والمساكن | عن الخدمات والمساكن | عن اثار فلسطين | شغل موقع الاقمار | استعان لتقريبها | تصاميم فلسطينية | تسع في الاعمال | شاهد تقنيات | نظيف ورائع | مفرات

Wohn Trends

Directory-www.com

+970-599-471991

شكل (2.2.1)

❖ الموقع الإلكتروني عقار استيت www.aqar_estate.com

هو عبارة عن موقع يعرض العقارات بأنواعها من أراضي أو شقق سكنية للبيع أو الشراء أو الايجار،

حيث يمكن اضافة الاعلان فيه بسهولة ومعرفة ما هو معروض من العقارات، وهو بالتالي متخصص في

مجال العقارات، على عكس الموقع الاول (شو بديك من فلسطين) الذي يعرض أكثر من مجال (العقارات)

والشكل (2.2.2) يبين موقع عقار استيت.



شكل (2.2.2)

اما فيما يتعلق بموقع مستكشف العقارات فهو عبارة عن موقع الكتروني يتخصص في مجال العقارات (قطع ارض أو شقق أو عمارات) فقط من حيث البيع أو الشراء أو الاستئجار، وهو بالتالي أقرب للموقع الثاني الذي يختص في مجال العقارات، لكنه لا يعرض العقارات على خرائط جوجل (Google Maps)، اما بالنسبة لهذا المشروع فعند عملية البحث عن العقار المناسب يتم عرض النتائج مباشرةً على خرائط جوجل (Google Maps)، كما انك تستطيع البحث عن العقار حسب السعر او الموقع، فيساعدك هذا على معرفة موقع وسعر العقار الذي يناسبك .

2.3 الاستنتاجات

تم التوصل من الدراسات السابقة، بأن الموقع (شو بدك من فلسطين) منح المواطنين البحث عن اي شئ، حيث يحتوي الموقع على دليل فلسطين الشامل، دليل فلسطين التجاري، دليل فلسطين الطبي، سيارات، عقارات، وظائف، اراضي، شقق، تعليم لغة عبرية، الاذاعات الفلسطينية، فلم يختص في مجال واحد، وأيضا تظهر نتيجة بحث المواطنين في جداول، اما الموقع الثاني (عقار استيت) فقد كان تركيزه على العقارات، ولكن كلا الموقعين لم يستخدم خرائط جوجل (Google Maps)، التي تساعد الاشخاص على معرفة موقع وسعر العقار المناسب لهم.

2.4 طرق جمع المعلومات

إن الطريقة التي تم اتباعها لجمع المعلومات كانت من خلال الإستفادة من مواقع الإنترنت والمواقع التعليمية، للتعرف على اهم الدراسات حول موضوع العقارات ، والإستعانة ببعض المدرسين في الجامعة عن متطلبات العمل في المشروع.

الفصل الثالث

تخطيط النظام



- ❖ المقدمة.
- ❖ مهام المشروع.
- ❖ المخاطر.
- ❖ تحليل المخاطر.
- ❖ تكلفة المشروع.
- ❖ شبكة المهام.
- ❖ الجدول الزمني.

3.1 المقدمة

سوف يتم في هذا الفصل تحديد كافة المتطلبات الضرورية للقيام بالمشروع، من خلال معرفة التكلفة المتوقعة له وعرض الوظائف الخاصة به وتحليلها، من ثم سيتم تحديد المخاطر وعمل تحليل لها، بالإضافة إلى تحضير جدول المهام الذي يبين الخطوات التي قمنا بها خلال الفترة الزمنية اللازمة لانتهاء المشروع.

3.2 مهام المشروع

يمكن تقسيم مهام المشروع الى:

أ. التواصل: ويشمل:

1. ايجاد فكرة المشروع: تطلب منا البحث عن الفكرة وايجادها حوالي 14 يوماً.
2. البحث على الانترنت: قمنا بعملية بحث عن المشروع في الانترنت، بهدف التعرف على مشاريع مشابهة له والاستفادة منها، وتطلب منا البحث حوالي 7 ايام.
3. جمع المعلومات: قمنا بجمع المعلومات اللازمة عن المشروع حتى نتمكن من فهمه فهما جيداً، ولنتمكن من معرفة ما علينا فعله، وقد تطلب منا 7 ايام لجمع المعلومات الضرورية.
4. دراسة مشاريع سابقة: قمنا بدراسة مشاريع سابقة لهذا المشروع، حتى نتمكن من معرفة ما يجب علينا فعله، ماذا يجب ان يحتوي النظام وما يمكننا تطويره عليه لتقديم الافضل، وقد تطلب ذلك منا حوالي 14 يوماً.

ب. التخطيط: ويشمل:

1. تخطيط المشروع: بعد ان قمنا بجمع المعلومات اللازمة، وأدركنا ما الذي يجب توفره في هذا المشروع من معدات واجهزة وبرامج واشخاص، وبناءا على ذلك قمنا بتجهيز الخطة التي ستساعدنا على ادارة عملنا وانجازه بشكل جيد، وقد تطلب منا تحضيرها حوالي 15 يوماً.
2. ايجاد المخاطر وتحليلها: هو معرفة ما قد يواجهنا من مخاطر وكيف يمكننا تجنبها، و تحليل لهذه المخاطر، وقد تطلب ذلك منا حوالي 10 ايام.

3. تحليل النظام: ويشمل:

1. تحليل النظام: بعد ان قمنا بكل المهام السابقة وأجريناها بنجاح، سيتم عملية تحليل للمتطلبات التي توصلنا اليها والتي يحتويها النظام، وسيتم عمل ذلك في الفصل القادم من المشروع، وقد توصلنا الى انه سيحتاج منا حوالي 48 يوماً اذا تدرجنا على الخطة بالشكل الصحيح.

ت. التصميم: ويشمل:

1. تصميم النظام: في هذا الفصل سيتم توضيح أجزاء النظام على شكل كيانات والعلاقات بين أجزاء النظام الأساسية والفرعية، بحيث سيتم وضع هذه الأجزاء على شكل مغلفات تدعى (Deployments) ومكونات المغلف الفرعية تسمى (Components)، كما وسيتم إعطاء تفصيل توضيحي لكل جزء فرعي من المغلف من حيث العمليات التي ستتم عليه وخصائصه.

ث. التنفيذ: ويشمل:

1. التنفيذ: في هذا الجزء سيتم برمجة النظام.

ج. تقييم: ويشمل:

1. اختبار وتصحيح النظام: سيتم فحص العمليات النظام التي يقوم بها مستخدمي النظام، حيث سيتم فحص كل عملية بشكل منفصل عن الأخرى، وفحص الشاشات التي تبين التأكد من صحة العمليات التي يقوم بها كل مستخدم.

2. توثيق المشروع: عملية توثيق المشروع هي الناتج النهائي من المهام السابقة، وبناء على تقسيمات الايام التي وضعناها سابقا سوف نحتاج الى 240 يوماً لتجهيزه.

3.3 المخاطر

في هذا الجزء يتم تحديد المخاطر التي يمكن أن تواجه فريق العمل خلال تنفيذ المشروع، حيث تعني المخاطر تسليط الضوء على أهم المشكلات التي قد يعاني منها هذا المشروع، ويمكن تصنيف المخاطر التي تؤثر في المشاريع ضمن مجموعتين، ترتبط المجموعة الأولى بالأوضاع السياسية والاقتصادية والاجتماعية التي يمر بها المجتمع، ولذلك يصعب تغييرها من قبل إدارة المشروع، وهي مجموعة المخاطر الخارجية. أما الثانية فهي المخاطر الداخلية وترتبط بالتوجهات العامة لسياسات المشروع، ويمكن معالجتها من قبل إدارة المشروع، وسيتم توضيح بعض الحلول والاقتراحات المناسبة لهذه المخاطر.

3.4 تحليل المخاطر

جدول رقم (3.3.1) المخاطر

المخاطر	الإحتمالية	التأثير	رقم الجدول
مخاطر ناجمة عن فريق العمل	50%	حرج	3.4.1
مخاطر ناجمة عن الإدارة	20%	حرج	3.4.2
مخاطر التسليم	20%	جدي	3.4.3
رفض النظام من قبل المواطنين	40%	وسط	3.4.4
مخاطر ناجمة عن الأجهزة	25%	وسط	3.4.5
عدم توفر خدمة الإنترنت لجميع المواطنين	20%	وسط	3.4.6

في هذا الجزء يتم توضيح لتحليل المخاطر التي تم ذكرها:

جدول 3.4.1 (مخاطر ناجمة عن فريق العمل)

Risk Information Sheet			
رقم الخطر: 1	اليوم: 2012-10-24	الإحتمالية: 50%	التأثير: حرج
الوصف: مخاطر ناجمة عن فريق العمل .			
تحسين الوصف:			
قد لا يكون فريق العمل في المشروع مستعدا في الوقت المناسب لعدة اسباب قد تكون صحية، نفسية، اجتماعية، اقتصادية في أثناء تنفيذ المشروع، بسبب وجود خلافات بين أفراد الفريق وعدم القدرة على انهاء المشروع بالوقت المطلوب و عدم التزام الفريق بالجدول الزمني للتنفيذ المشروع .			
مراقبة/تخفيف الخطر:			
التزام الفريق بالجدول الزمني للمشروع اثناء التنفيذ، وتوثيق كل خطوة يقوم بها اعضاء الفريق . بالاضافة الى الاجتماعات المتواصلة لتخفيف من المخاطر في حال تم تغير احد الاعضاء الفريق .			

<p>ادارة المخاطر/خطة الطوارئ:</p> <p>يجب على كل فرد من أفراد الفريق الإحاطة بجميع مهام المشروع، بحيث يستطيع كل منهم القيام بمهام العضو الآخر، وعقد اجتماع في نهاية كل أسبوع لمناقشة ما تم انجازه.</p> <p>الوضع الحالي للخطر: تم العمل على تخفيف الخطر عند حدوثه .</p>

جدول 3.4.2 (مخاطر ناجمة الادارة)

Risk Information Sheet			
رقم الخطر: 2	اليوم: 2012-10-24	الإحتمالية: 20%	التأثير: حرج
الوصف: مخاطر ناجمة عن الادارة .			
تحسين الوصف:			
عدم اتخاذ القرارات اللازمة في الوقت المناسب، مثل مفاجأة اعضاء الفريق بقرارات يجب الالتزام بها وعدم الحياد عنها .			
مراقبة/تخفيف الخطر:			
الالتزام باعطاء القرارات في اوقاتها المناسب، وعمل الفريق على تنفيذها في اوقاتها.			
ادارة المخاطر/خطة الطوارئ:			
يجب على كل فرد من أفراد الفريق الإحاطة بجميع القرارات، والالتزام بالجدول الزمني للعمل على انجازها.			
الوضع الحالي للخطر: تم العمل على تخفيف الخطر عند حدوثه .			

جدول 3.4.3 (مخاطر التسليم)

Risk Information Sheet			
رقم الخطر: 3	اليوم: 2012-10-24	الإحتمالية: 20%	التأثير: ووسط
الوصف: مخاطر التسليم .			
تحسين الوصف:			
قد لا يسلم المشروع في الوقت المحدد بشكل دقيق، واحتياجات المشروع قد تكون أعلى من الإمكانيات المتاحة والنظام قد يفشل في تلبية كل الوظائف المطلوبة.			
مراقبة/تخفيف الخطر:			
الالتزام بالجدول الزمني ، والعمل على (Process Models) المناسب للوقت المسموح للمشروع (Incremental ,RAD ,Spiral , ...)			

ادارة المخاطر/خطة الطوارئ: مساعدة اشخاص من خارج فريق العمل على انجاز المشروع .
الوضع الحالي للخطر: العمل على تخفيف الخطر قبل حدوثه .

جدول 3.4.4 (رفض النظام من قبل المواطنين)

Risk Information Sheet			
رقم الخطر: 4	اليوم: 2012-10-24	الإحتمالية: 40%	التأثير: وسط
الوصف: رفض النظام من قبل المواطنين .			
تحسين الوصف: رفض النظام من قبل المواطنين، ويرجع ذلك لعدة أسباب منها: العادات والتقاليد(الطريقة المعتاد عليها من قبل المواطنين في عمليات بيع واسنجر للعقارات)، عدم الثقة بالمواقع الإلكترونية، وعدم الرغبة في التعامل مع المواقع الإلكترونية، وعدم القدرة على استخدام الوسائل التكنولوجية، عدم القدرة على فهم المواطنين كيفية التسجيل في الموقع و التعامل معه.			
مراقبة/تخفيف الخطر: العمل على اقناع المواطنين بوضع اعلانات تفيد مزايا وايجابيات الموقع وسهولة التعامل معه.			
ادارة المخاطر/خطة الطوارئ: مساعدة اشخاص في العمل على اعلان عن الموقع، ونشر أهميته بالنسبة للمواطنين .			
الوضع الحالي للخطر: العمل على تخفيف الخطر قبل حدوثه .			

جدول 3.4.5 (مخاطر ناجمة عن الأجهزة)

Risk Information Sheet			
رقم الخطر: 5	اليوم: 2012-10-24	الإحتمالية: 25%	التأثير: وسط
الوصف: مخاطر ناجمة عن الأجهزة .			
تحسين الوصف: قد تتعرض الأجهزة للأعطال المفاجئة، مما يؤدي الى تأخر المشروع أو ضياعه .			
مراقبة/تخفيف الخطر: عمل الفريق على عمل نسخ اضافية للمشروع في اكثر من مكان، وتوفير الامان له .			

<p>ادارة المخاطر/خطة الطوارئ:</p> <p>ايجاد أجهزة بديلة لها مواصفات الاجهزة التي يتم العمل عليها (حاسوب ، طابعة ، كاميرا ، او اي جهاز سيتم التعامل معه في المشروع كجزء أساسي أو ثانوي) .</p> <p>الوضع الحالي للخطر: العمل على تخفيف الخطر قبل حدوثه .</p>

جدول 3.4.6 (عدم توفر خدمة الإنترنت لجميع المواطنين)

Risk Information Sheet			
رقم الخطر: 6	اليوم: 2012-10-24	الاحتمالية: 50%	التأثير: وسط
الوصف: عدم توفر خدمة الإنترنت لجميع المواطنين .			
تحسين المحتوى/الوصف:			
عدم توفر خدمة الإنترنت لدى المواطنين، فذلك يصعب عملية التسجيل في الموقع واستخدامه، وتتبع عمليات الشراء والبيع، وضعف القدرة على استخدام الحاسوب أو الإنترنت .			
مراقبة/تخفيف الخطر:			
العمل على توفر مصادر أخرى لتوفير خدمات الإنترنت مثل مقاهي الإنترنت .			
إدارة المخاطر/خطة الطوارئ:			
في حال عدم توفر الانترنت لن يكون هناك مجال لاستخدام الموقع إلا في حال استعمال خدمة الانترنت المتوفرة في احد مراكز الحاسوب .			
الوضع الحالي للخطر: العمل على تخفيف الخطر قبل حدوثه.			

3.5 تكلفة المشروع

3.5.1 تكلفة تطوير النظام

تقسم تكاليف تطوير النظام الى ثلاثة أقسام :-

3.5.1.1 تكاليف المعدات

مثل أي مشروع يقام ويتم بناؤه وعمله ، يلزم هذا المشروع بعض المعدات التي يجب مراعاة حساب تكاليفها، ومعرفة القيمة المستهلكة لهذه المعدات، حيث سيتم تقسيم مدة العمل في المشروع على العمر الافتراضي لهذه المعدات التي لا تقتصر تكاليفها على هذا المشروع (يمكن الاستفادة من هذه المعدات وتكاليفها في مشاريع واعمال أخرى).

$$\text{تكلفة الجهاز للمشروع لمدة سنة} = \text{تكلفة الكلية} * \frac{1}{\text{العمر الافتراضي للجهاز}}$$

تكلفة Computer Compaq = 700
العمر الافتراضي Computer Compaq هي 3 سنوات

$$\text{تكلفة الجهاز Computer Compaq للمشروع لمدة سنة} = 700 * \frac{1}{3}$$

$$\approx 233.3$$

جدول 3.5.1.1 (تكاليف المعدات)

Num	Name	Quantity	Price \$	Notes
1	Computer Compaq 610 Core 2 Due	1	233.3	Processor 2.1GHz RAM 3GB , HD 240 GB
3	Printer	1	80	

Total ≈ =	314\$
-----------	-------

3.5.1.2 تكاليف البرمجيات

تشمل جميع البرمجيات الضرورية لتطوير المشروع، وحساب تكلفة هذه البرمجيات المتوقع استخدامها

لتطوير المشروع، حيث تم افتراض شراء هذه البرمجيات كما يلي:

جدول (3.5.1.2) تكاليف البرمجيات

Num	Name	Price \$	Note
1	Eclipse	Free	Professional Edition
2	XAMPP	Free	الطبعة المهنية
3	Windows 7	90	Ultimate Edition
4	Microsoft Office 2007	120	Editors and Presentation
5	Microsoft Office Visio 2007	70	Drawing tool
Total		280\$	

3.5.1.3 تكلفة فريق العمل

المصادر البشرية القائمة على تطوير هذا المشروع تتمثل في فريق العمل، الذي يتكون من طالبين بحيث يتم العمل بمعدل 3 ساعات يوميا في 4 أيام في الأسبوع، وعلى مدار 30 أسبوع، حيث يتقاضى كل شخص في الفريق على الساعة الواحدة مبلغ \$4، وبذلك يصبح الناتج لتكلفة العمل خلال أسبوع كالتالي :

التكلفة الأسبوعية : $48 = 4 * 12 * 1$
التكلفة الكلية : $1440 = 30 * 48$

مما سبق يمكن إجمال تكلفة المشروع كما هو موضح بالجدول التالي:

جدول (3.5.1.3) تكاليف تطوير النظام

الرقم	تكاليف تطوير النظام	التكلفة \$
1	تكاليف المعدات	\$580
2	تكاليف البرمجيات	\$314
3	تكاليف فريق العمل	\$1440
المجموع		\$2334

3.5.2 تكلفة تشغيل النظام

تحديد المعدات اللازمة لتشغيل النظام في صورته النهائية بما يلي:

- جهاز حاسوب Pentium Dual Core 2.0 GHz(Server) , RAM 1GB
- اشتراك خط إنترنت DSL بسرعة لا تقل عن 256kbps
- حجز مدى (Domain) للموقع الإلكتروني على شبكة الإنترنت .

جدول 3.5.2 (تكلفة تشغيل النظام) / سنويا

Num	Name	Quantity	Price \$	Notes
1	Computer Dual Core(Server)	1	450	RAM 1GB , CPU 2.0 GHz
2	DSL Network Link	1	12	Monthly
3	Web Hosting Domain	1	15	Yearly

Total	\$477
-------	-------

3.6 شبكة المهام

في الجدول التالي سيتم توضيح مجموعة المهام التي قمنا بها، وتوزيع هذه المهام على الايام، كل مهمة حسب الفترة التي يمكن من خلالها انجاز تلك المهمة:

جدول 3.6.1 (جدول المهام)

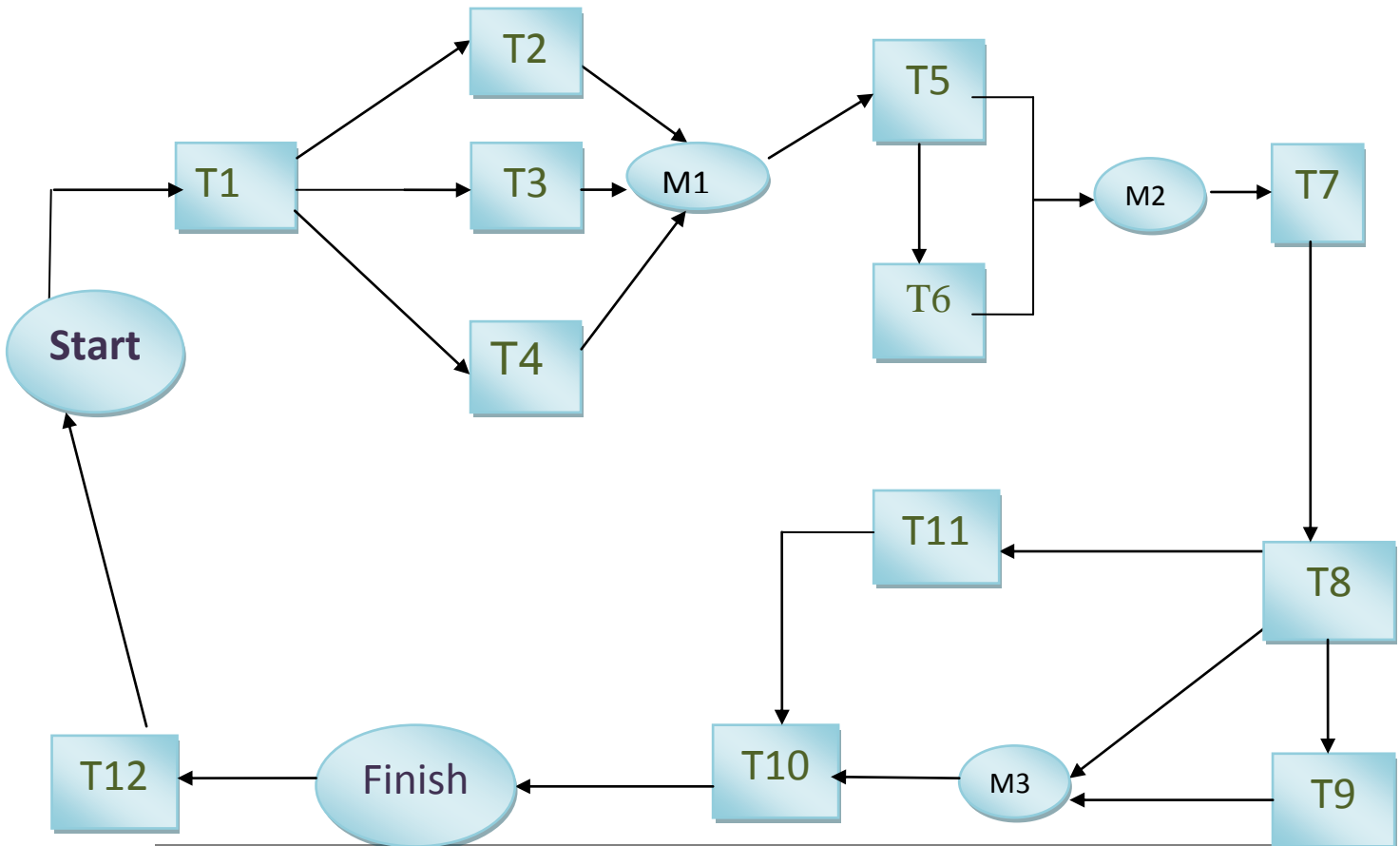
Task Symbol	Task	Duration (Days)	Dependencies
T1	ايجاد فكرة المشروع (Create project idea)	14	
T2	البحث على الانترنت (Browse websites)	7	T1
T3	جمع المعلومات (Collecting information)	7	T1
T4	دراسة مشاريع سابقة (Studying previous projects)	14	T1
T5	تخطيط المشروع (Project planning)	15	T2,T3,T4,(M1)
T6	ايجاد المخاطر وتحليلها (Risk defining and analyzing)	10	T5
T7	تحليل النظام (System analysis)	48	T5,T6,(M2)
T8	تصميم النظام وتصميم واجهة المستخدم System Design and User Interface Design	55	T7
T9	تحسينات وتحقق من الصحة Enhancement and validation	10	T8
T10	تنفيذ المشروع	20	T8,T9

	Implementation of Website		
T11	اختبار وتصحيح النظام Testing and correcting the System	40	T8
T12	توثيق المشروع Review and Documentation	240	T11

3.7 الجدول الزمني

يمثل الشكل التالي مجموعة المهام بالترتيب:

شكل 3.7.1 (شبكة المهام)



يمثل الجدول الزمني مجموعة المهام التي قمنا بها خلال فترة مقدمة المشروع من خلال توزيع تلك

المهام على الفترات الزمنية:

ID	Task Name	Sep 2012			Oct 2012				Nov 2012				Dec 2012				Jan 2013				Feb 2013			Mar 2013			Apr 2013			May 2013																																	
		2/9	9/9	16/9	23/9	30/9	7/10	14/10	21/10	28/10	4/11	11/11	18/11	25/11	2/12	9/12	16/12	23/12	30/12	6/1	13/1	20/1	27/1	3/2	10/2	17/2	24/2	3/3	10/3	17/3	24/3	31/3	7/4	14/4	21/4	28/4	5/5	12/5	19/5																								
1	Task 1	■																																																													
2	Task 2											■																																																			
3	Task 3											■																																																			
4	Task 4											■																																																			
5	Task 5											■																																																			
6	Task 6											■																																																			
7	Task 7											■																																																			
8	Task 8																					■																																									
9	Task 9																															■																															
10	Task 10																																									■																					
11	Task 11																																																			■											
12	Task 12	■																																																													

شكل 3.7.2 (الجدول الزمني لتنفيذ المهام)

الفصل الرابع

تحليل النظام



❖ المقدمة

❖ وصف المتطلبات

❖ وظائف النظام

❖ CRC Modeling

❖ Activity Diagram

❖ Class Hierarchies and Subsystems

4.1 المقدمة

تعتبر مرحلة تحليل النظام من أهم مراحل تطوير المشروع، حيث يتم تناول المشروع من عدة جوانب لتوضيح هيكلية عمل النظام، حيث يتم وصف أجزاء المشروع وصفاً تفصيلياً باستخدام

(Object Oriented Analysis) ولغة النمذجة الموحدة (Unified Modeling Language(UML)، من حيث مدخلات المشروع ومخرجات المشروع وايضا خصائصه، والعمليات التي سيقوم بها مستخدم النظام بحيث يكون هذا التفصيل عن طريق وصف العمليات لكل مستخدم النظام باستخدام (Use Case) ليوضح بشكل مفصل سير العمليات التي سيقوم مستخدم النظام أثناء جلسة العمل وكيفية تفاعل المستخدم الحالي مع النظام، كما ويتم رسم هذه العمليات باستخدام رسومات تفصيلية (Use Case, Class Hierarchies, Activity Diagram, State Transition Diagram).

4.2 وصف المتطلبات

مستكشف العقارات هو عبارة عن موقع الالكتروني يمكن أي مواطن من التسجيل والدخول اليه اما لعرض عقاراتهم للبيع، أو الاستئجار، و تمكن المواطنين الذين يبحثون عن عقارات بهدف الشراء أو الاستئجار من البحث عن العقار المناسب لهم، حيث يوجد عدة مستخدمين لهذا النظام وهم مسؤول الموقع والزبون والزائر.مسؤول الموقع هو الذي يستطيع القيام بجميع المهام التي يقوم بها كل من الزبون والزائر بالاضافة الى امكانية حذف او اضافة مستخدم، و الزبون هو الذي يستخدم النظام بهدف تصفح أو بحث عن عقار باستخدام القائمة او الخريطة ، حيث يمكنه النظام من حذف و تعديل و اضافة عقاراته في الموقع، اما الزائر هو الذي يستخدم النظام بهدف تصفح أو بحث عن عقار باستخدام القائمة او الخريطة، وفيما يلي توضيح الوظائف والعمليات لكل مستخدم النظام.

4.3 وظائف النظام

الوظائف التي يقدمها النظام تقسم إلى ثلاث وظائف:

➤ الوظائف التي تتعلق بزوار (متصفح) الموقع (Visitor).

هم الأشخاص الذين يستطيعون ان يستخدموا الموقع كضيوف، حيث يستطيع الزائر القيام بعدة مهام وهي تصفح العقارات و استعراض عقار و البحث عن العقارات، حيث تظهر النتائج البحث مباشرة على الخريطة وفي القائمة أيضا.

➤ الوظائف التي تتعلق زبائن الموقع (Customer).

هم الأشخاص لديهم حساب في الموقع، ويستطيعون القيام بجميع المهام التي يقوم بها الزائر بالإضافة الى تسجيل الدخول إلى الموقع و تسجيل الخروج من الموقع و استرجاع كلمة المرور و التعديل على بياناته الشخصية و تصفح وإدارة قائمة عقاراته في اضافة عقار و حذف عقار و تعديل معلومات العقار، وايضا قدرة الزبائن على تصفح الرسائل وادارتها، و ارسال رسالة الى صاحب العقار، و الرد على رسالة.

➤ الوظائف التي تتعلق بمسؤول الموقع (Administrator).

مسؤول الموقع هو شخص يستطيع القيام بجميع المهام التي يقوم بها كل من الزائر و الزبون بالإضافة الى حذف مستخدم من الموقع و اضافة مستخدم الى الموقع .

➤ يوضح الشكل التالي شكل (4.3.1) مستخدم نظام مستكشف العقارات وهم مسؤول الموقع

والزبون والزائر، كما يوضح الوظائف المشتركة بينهم:



شأن (4.3.1) Use Case Diagram System

❖ الوظائف التي تتعلق بنوار (متصفح) الموقع (Visitor).

جدول (4.3.1) تصفح واستعراض العقارات باستخدام القائمة

Use Case	تصفح واستعراض العقارات باستخدام القائمة.
Primary Actor	المستخدم (الزائر).
In Context Goal	السماح للمستخدم بتصفح العقارات في الموقع باستخدام القائمة.
Preconditions	الدخول الى عنوان الموقع.
Trigger	تصفح العقارات الموجودة في الموقع من خلال القائمة.
Scenario	<ul style="list-style-type: none"> ➤ يقوم الزائر بالدخول الى عنوان الموقع. ➤ التنقل بين صفحات النظام، لتصفح العقارات. ➤ الذهاب الى الجزء المتعلق بعرض العقارات في القائمة. ➤ يقوم النظام بإظهار معلومات مختصرة عن كل عقار. ➤ يقوم الزائر بالضغط الزائر على صورة عقار معين أو على المعلومات المختصرة الخاصة للعقار في القائمة يقوم النظام بالانتقال الى صفحة معلومات العقار، حيث تظهر جميع معلومات العقار المختار من قبل المستخدم (الزائر).
Exceptions	عدم معرفة استخدام الموقع، او الضغط على في مكان اخر.

جدول (4.3.2) تصفح واستعراض العقارات من خلال الخريطة

Use Case	تصفح واستعراض العقارات باستخدام الخريطة.
Primary Actor	المستخدم (الزائر).
In Context Goal	السماح للمستخدم بتصفح العقارات في الموقع باستخدام الخريطة.
Preconditions	الدخول الى عنوان الموقع.
Trigger	تصفح العقارات الموجودة في الموقع من خلال الخريطة.
Scenario	<ul style="list-style-type: none"> ➤ يقوم الزائر بالدخول الى عنوان الموقع. ➤ التنقل بين صفحات النظام، لتصفح العقارات. ➤ الذهاب الى الجزء المتعلق بعرض العقارات على الخريطة. ➤ يقوم النظام بإظهار معلومات مختصرة عن كل عقار. ➤ يقوم الزائر بالضغط على المعلومات المختصرة الخاصة للعقار في الخريطة

	يقوم النظام بالانتقال الى صفحة معلومات العقار، حيث تظهر جميع معلومات العقار المختار من قبل المستخدم(الزائر).
Exceptions	عدم معرفة استخدام الموقع.

جدول(4.3.3) البحث عن العقارات

Use Case	البحث عن العقارات
Primary Actor	المستخدم(الزائر).
In Context Goal	السماح للمستخدم بالبحث عن العقارات في الموقع.
Preconditions	الدخول الى عنوان الموقع.
Trigger	البحث في العقارات الموجودة في الموقع من خلال القائمة والخريطة.
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم (الزائر) بالدخول الى عنوان الموقع. ➤ التنقل بين صفحات النظام، للبحث عن العقارات. ➤ الذهاب الى الواجهة الخاصة بالبحث عن العقارات. ➤ يقوم المستخدم بادخال معلومات العقار المراد البحث عنه، حيث يستطيع المستخدم البحث حسب نوع العقار(منزل او قطعة أرض) وايضا مكان العقار(في اي مدينة)، باضافة الى امكانية البحث حسب حالة العقار(بيع/استئجار)، وامكانية تحديد منطقة البحث (مثال اريد اظهار كل العقارات التي تبعد عني 1 كم) وامكانية البحث حسب سعر ومساحة العقار، وامكانية تحديد موقع على الخريطة للبحث على العقارات القريبة للموقع المحدد من قبل المستخدم(الموقع قد يكون موقع الشخص او اي موقع اخر) ➤ يضغط المستخدم على ايقونة بحث. ➤ يقوم النظام باظهار نتائج البحث مباشرة على الخريطة و في القائمة
Exceptions	ادخال معلومات خاطئة، مثل تحديد المكان غير صحيح على خرائط جوجل للبحث على العقارات القريبة من الموقع المحدد من قبل المستخدم.

جدول (4.3.4) تسجيل حساب.

Use Case	تسجيل حساب.
Primary Actor	المستخدم (الزائر).
In Context Goal	تسجيل حساب في الموقع.
Preconditions	الدخول الى عنوان الموقع.
Trigger	امكانية اضافة عقارات وادارتها.
Scenario	<ul style="list-style-type: none"> ➤ يقوم الزائر بالدخول الى عنوان الموقع. ➤ اختيار تسجيل. ➤ يقوم النظام بالانتقال الى صفحة تسجيل ➤ يقوم المستخدم بتعبئة نموذج التسجيل، حيث يقوم بادخال بياناته، ثم يضغط على ايقونة ارسال ➤ في حال كانت البيانات المدخلة صحيحة، تخزن البيانات في قاعدة البيانات ويقوم النظام بتحويل الشخص الى صفحة تسجيل الدخول. ➤ في حال كانت البيانات المدخلة غير صحيحة، يطلب النظام من الشخص اعادة ادخال بيانات صحيحة مكان البيانات الغير صحيحة.
Exceptions	ادخال بيانات غير صحيحة.

❖ الوظائف التي تتعلق الزبائن (Customer).

يستطيع الزبون القيام بعدة مهام مشتركة مع الزائر وهي

- جدول (4.3.1) تصفح واستعراض العقارات باستخدام القائمة.
- جدول (4.3.2) تصفح واستعراض العقارات باستخدام الخريطة.
- جدول (4.3.3) البحث عن العقارات.

وكما انه يستطيع القيام بالامور التالية :

جدول (4.3.5) تسجيل الدخول إلى الموقع

Use Case	تسجيل الدخول.
Primary Actor	المستخدم (الزبون) .
Goal In Context	السماح للمستخدم بالدخول إلى الموقع.
Preconditions	إدخال اسم مستخدم وكلمة مرور.
Trigger	السماح للمستخدم بالدخول للمشاركة في الموقع كمشارك.
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بفتح صفحة الموقع. ➤ إدخال اسم المستخدم في المكان المخصص لاسم المستخدم وهو عبارة عن البريد الإلكتروني. ➤ إدخال كلمة المرور في المكان المخصص لكلمة المرور. ➤ الضغط على أيقونة تسجيل دخول. ➤ التأكد من البيانات المدخلة. ➤ يقوم الموقع بارجاع المستخدم إلى الصفحة تسجيل الدخول اذا تم ادخال اسم المستخدم وكلمة المرور غير فعالات. ➤ يقوم الموقع بدخول المستخدم إلى الصفحة الرئيسية اذا تم ادخال اسم المستخدم وكلمة المرور فعالات. ➤ تتغير حالة المستخدم " الأيقونة " من تسجيل دخول إلى تسجيل الخروج.
Exceptions	عدم القدرة على تسجيل الدخول الى الموقع (ادخال اسم المستخدم أو كلمة المرور غير صحيحة).

جدول (4.3.6) تسجيل الخروج من الموقع

Use Case	تسجيل الخروج .
Primary Actor	المستخدم (الزبون).
Goal In Context	السماح للمستخدم بالخروج من الموقع .
Preconditions	تسجيل الدخول إلى الموقع .
Trigger	السماح لمستخدم بالخروج من الموقع بعد إجراء المطلوب من الموقع .
Scenario	<ul style="list-style-type: none"> ➤ تأكد المستخدم من حفظ بياناتهم واعمالهم قبل الخروج. ➤ الضغط على أيقونة تسجيل الخروج من الموقع . ➤ تحويله الى الصفحة الرئيسية حيث تُحول ايقونة تسجيل الخروج من حالة خروج الى حالة دخول .

	➤ يقوم النظام بالانتقال الى الصفحة الرئيسية.
Exceptions	عدم تأكيد الخروج من الموقع .

جدول (4.3.7) استرجاع كلمة المرور

Use Case	استرجاع كلمة المرور
Primary Actor	المستخدم(الزبون).
In Context Goal	السماح للمستخدم باسترجاع كلمة المرور .
Preconditions	الدخول الى عنوان الموقع.
Trigger	القدرة على استرجاع كلمة المرور عند فقدانها.
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بالدخول الى عنوان الموقع. ➤ الضغط على ايقونة استرجاع كلمة المرور . ➤ يقوم الموقع بارساله الى صفحة استرجاع كلمة المرور . ➤ يُدخل المستخدم البريد الالكتروني، في المكان المخصص له، ثم يضغط على ايقونة ارسال. ➤ يذهب الشخص الى بريده الالكتروني، ويفتح الرسالة التي اتته من الموقع ويعمل على تسجيل كلمة مرور جديدة.
Exceptions	ادخال بريد الكتروني خاطئ.

جدول (4.3.8) التعديل على البيانات

Use Case	تعديل بيانات المستخدم في النظام .
Primary Actor	المستخدم(الزبون) .
Goal In Context	تغيير بيانات المستخدم والتعديل عليها.
Preconditions	دخول المستخدم المشترك بالموقع إلى صفحة التعديل .
Trigger	إدخال تعديلات على بيانات المستخدم على قاعدة البيانات الخاصة بالمستخدمين فتعدل البيانات ويتم التعامل في الموقع مع البيانات المعدلة بدلاً من البيانات القديمة .
Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول الى الموقع . ➤ يقوم المستخدم بالذهاب الى ملفه الشخصي. ➤ يقوم النظام بارساله الى صفحة ملفه الشخصي. ➤ يضغط المستخدم على أيقونة تعديل حساب .

	<ul style="list-style-type: none"> ➤ يقوم النظام بإرساله الى صفحة تعديل الحساب. ➤ يقوم النظام باظهار بياناته في النافذة الخاصة بمعلومات الحساب. ➤ التعديل على البيانات المراد التعديل عليها. ➤ الضغط على ايقونة حفظ. ➤ يقوم النظام بتأكد من البيانات المدخلة . ➤ في حال كانت البيانات المدخلة صحيحة، يتم حفظ التعديل. ➤ في حال كانت البيانات المدخلة غير صحيحة، يطلب النظام من المستخدم باعادة ادخال البيانات الصحيحة.
Exceptions	عدم حفظ التغييرات بعد التعديل .

جدول (4.3.9) اضافة قطعة أرض

Use Case	اضافة المستخدم قطعة أرض
Primary Actor	المستخدم (الزبون) .
Goal In Context	اضافة قطعة ارض جديدة.
Preconditions	تسجيل الدخول للموقع .
Trigger	ظهور العقار المضاف في الموقع امام جميع مستخدمي النظام (اعلان عن هذا العقار)
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بالدخول الى عنوان الموقع. ➤ ادخال اسم المستخدم وكلمة المرور . ➤ الذهاب الى ايقونة اضافة قطعة ارض. ➤ يقوم النظام بإرسال المستخدم الى صفحة اضافة أرض جديد. ➤ يقوم المستخدم بادخال معلومات عن قطعة الارض من حيث تحديد حالتها (بيع/استئجار) وايضا ادخال نوع الارض (زراعية /صناعية / تجارية) ➤ وادخال المساحة و سعرها و المدينة التي تقع بها وتحديد موقع العقار(الارض) على خرائط جوجل (Google Maps)، وادخال وصف عن العقار وتحميل المرفقات الخاصة بالعقار(الصور). ➤ يضغط المستخدم على ايقونة حفظ. ➤ يقوم النظام بتأكد من المعلومات المدخلة. ➤ اذا كانت المعلومات المدخلة صحيحة يقوم النظام بحفظ البيانات.

	➤ اذا كانت المعلومات المدخلة غير صحيحة يقوم النظام بالطلب من المستخدم اعادة ادخال البيانات.
Exceptions	خطأ في الاضافة مثل اعطاء معلومات خاطئة عن العقار المنوي اضافته.

جدول (4.3.10) اضافة منزل.

Use Case	اضافة المستخدم منزل.
Primary Actor	المستخدم (الزبون) .
Goal In Context	اضافة منزل جديد.
Preconditions	تسجيل الدخول للموقع .
Trigger	ظهور العقار المضاف في الموقع امام جميع مستخدمي النظام (اعلان عن هذا العقار)
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بالدخول الى عنوان الموقع. ➤ ادخال اسم المستخدم وكلمة المرور. ➤ الذهاب الى ايقونة اصف منزل. ➤ يقوم النظام بارسال المستخدم الى صفحة اضافة منزل جديد. ➤ يقوم المستخدم بادخال معلومات المنزل (عدد الغرف وعدد الشرفات وعدد الحمامات وعدد المطابخ ورقم الطابق ومساحته وسعره وتحديد المدينة التي يقع بها العقار وتحديد موقع العقار على خرائط جوجل (Google Maps)، وتحديد حالة العقار (بيع/استئجار) وادخال وصف عن العقار وتحميل المرفقات الخاصة بالعقار(الصور). ➤ يضغط المستخدم على ايقونة حفظ. ➤ يقوم النظام بتأكد من المعلومات المدخلة. ➤ اذا كانت المعلومات المدخلة صحيحة يقوم النظام بحفظ البيانات المدخلة. ➤ اذا كانت المعلومات المدخلة غير صحيحة يقوم النظام بالطلب من المستخدم اعادة ادخال البيانات.
Exceptions	خطأ في الاضافة مثل اعطاء معلومات خاطئة عن العقار المنوي اضافته.

جدول (4.3.11) حذف عقار

Use Case	حذف عقار مضاف سابقا في الموقع .
Primary Actor	المستخدم (الزبون) .
Goal In Context	حذف العقار .
Preconditions	تسجيل الدخول للموقع .
Trigger	حذف العقار المضاف في الموقع امام جميع مستخدمي النظام .
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بالدخول الى عنوان الموقع. ➤ ادخال اسم المستخدم وكلمة المرور. ➤ الضغط على ايقونة عقاراتي. ➤ يرسله النظام الى صفحة عقاراته. ➤ يختار المستخدم العقار المراد حذفه و يضغط على ايقونة حذف عقار . ➤ يقوم النظام بارسال رسالة تأكيد عملية الحذف. ➤ في حال تم تأكيد المستخدم عملية الحذف، يتم حذف العقار. ➤ في حال لم يتم تأكيد المستخدم عملية الحذف، يرسله النظام الى صفحة عقاراته.
Exceptions	حذف عقار اخر .

جدول (4.3.12) تعديل معلومات العقار

Use Case	تعديل معلومات عقار مضاف مسابقا في الموقع .
Primary Actor	المستخدم (الزبون) .
Goal In Context	تعديل معلومات العقار .
Preconditions	تسجيل الدخول للموقع .
Trigger	تعديل معلومات العقار المضاف في الموقع امام جميع مستخدمي النظام .
Scenario	<ul style="list-style-type: none"> ➤ يقوم المستخدم بالدخول الى عنوان الموقع. ➤ ادخال اسم المستخدم وكلمة المرور ➤ الضغط على ايقونة عقاراتي. ➤ يرسله النظام الى صفحة عقاراته. ➤ يضغط المستخدم على ايقونة تعديل . ➤ يرسله النظام الى صفحة تعديل العقار المختار ويقوم الموقع باظهار معلومات العقار القديمة في النافذة.

	<ul style="list-style-type: none"> ➤ يقوم المستخدم بتعديل البيانات. ➤ يضغط المستخدم على حفظ البيانات. ➤ يتأكد النظام من البيانات المدخلة . ➤ في حال كانت البيانات المدخلة صحيحة، يرسله النظام الي صفحة العقار (صفحة يظهر بها جميع معلومات العقار)،وسالة في اعلى الصفحة (تم حفظ العقار). ➤ في حال كانت البيانات المدخلة غير صحيحة، يطلب النظام ادخال بيانات صحيحة.
Exceptions	عدم الضغط على ايقونة حفظ معلومات .

جدول(4.3.13) تصفح الرسائل وادارتها.

Use Case	تصفح الرسائل وادارتها.
Primary Actor	المستخدم (الزبون).
Goal In Context	ادارة الرسائل.
Preconditions	تسجيل الدخول للموقع .
Trigger	تصفح الرسائل والتعامل معها.
Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول الى الموقع. ➤ الذهاب الى ايقونة رسائلي. ➤ يقوم الموقع بارساله الى صفحة الرسائل. ➤ يقوم المستخدم بتصفح الرسائل المرسله او المستقبله(حذف).
Exceptions	حذف رسالة بالخطأ .

جدول(4.3.14) ارسال رسالة الى صاحب العقار .

Use Case	ارسال رسالة الى صاحب العقار .
Primary Actor	المستخدم (الزبون).
Goal In Context	التواصل مع اصحاب العقارات.
Preconditions	تسجيل الدخول للموقع .
Trigger	ارسال رسالة الى صاحب العقار والتواصل معه.

Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول الى الموقع. ➤ فتح صفحة معلومات العقار . ➤ الضغط على ايقونة ارسال رسالة. ➤ كتابة الرسالة. ➤ اضغط على ايقونة ارسال. ➤ يقوم النظام بارسال الرسالة الى صاحب العقار .
Exceptions	عدم وصول الرسالة.

جدول(4.3.15) الرد على رسالة.

Use Case	الرد على رسالة.
Primary Actor	المستخدم (الزبون).
Goal In Context	التواصل مع اصحاب العقارات.
Preconditions	تسجيل الدخول للموقع .
Trigger	الرد على رسالة مرسله سابقا.
Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول في الموقع ➤ الذهاب الى ايقونة رسائلي. ➤ فتح رسالة مرسله. ➤ كتابة الرد. ➤ الضغط على ايقونة ارسال.
Exceptions	عدم وصول الرسالة.

❖ الوظائف التي تتعلق بمسؤول الموقع (Administrator).

يستطيع مسؤول الموقع القيام بجميع المهام التي يقوم بها مستخدم النظام (لزائر او الزبون) بالاضافة

الى المهام التالية:

جدول (4.3.16) حذف مستخدم من الموقع

Use Case	الحذف مستخدم من الموقع .
Primary Actor	مسؤول الموقع .
Goal In Context	حذف حساب مستخدم من الموقع .
Preconditions	وجود المستخدم المراد حذف حسابه .
Trigger	وجود حساب لمستخدم يريد تعطيله .
Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول الى الموقع. ➤ الدخول الى صفحة المستخدمين. ➤ يقوم النظام بالانتقال الى صفحة المستخدمين. ➤ الضغط على ايقونة حذف بجانب المستخدم المراد حذفه. ➤ يقوم النظام بارسال رسالة تأكد عملية الحذف أو التراجع عن عملية الحذف. ➤ في حال تم تأكيد عملية الحذف، يتم حذف المستخدمين النظام. ➤ في حال لم يتم تأكيد عملية الحذف، يتم ارجاع مسؤول الموقع الى صفحة المستخدمين.
Exceptions	أن يتم حذف مستخدم اخر .

جدول (4.3.17) اضافة مستخدم الى الموقع

Use Case	اضافة مستخدم الى الموقع.
Primary Actor	مسؤول الموقع .
Goal In Context	امكانية اضافة مستخدم الى الموقع.
Preconditions	وجود مستخدمين لاضافتهم.
Trigger	وجود حساب لمستخدم، تمكنهم من اضافة عقاراتهم.
Scenario	<ul style="list-style-type: none"> ➤ تسجيل الدخول الى الموقع. ➤ الدخول الى صفحة المستخدمين. ➤ الضغط على ايقونة اضافة. ➤ تعبئة معلومات المستخدمين، والضغط على ايقونة ارسال. ➤ يقوم النظام بتفعيل حسابات المستخدمين، وحفظ بياناتهم في قاعدة البيانات.
Exceptions	أن لا يتم اضافة المستخدم .

4.4 CRC Modeling (Class-Responsibilities-Collaboration)

(بطاقة نموذج CRC)

وهي عبارة عن طريقة بسيطة وقوية في تحليل الأنظمة التي تستخدم الكائنات الموجهة (Object Oriented) باستخدام البطاقات، حيث تتضمن عملية انشاء هذه البطاقات تحليل مستخدم النظام وأجزاءه، والعمليات التي يتم تنفيذها في النظام ذاته، كما ويتم تحديد العلاقات والروابط وما بين اجزاء المشروع وخصائص كل جزء من اجزاءه.

حيث يتكون هذا النظام (CRC) من مجموعة بطاقات، بحيث تقسم كل بطاقة الى ثلاثة أجزاء كالتالي:

1) Class (الفئة).

2) Responsibilities (المسؤولية).

3) Collaboration (التعاون).

والموضحة كالتالي:

1) Class (الفئة): وتمثل مجموعة من الكيانات المتشابهة، حيث أن هذه الكيانات تكون من

ضمن خصائص ومكونات النظام، وقد تكون اما شخص أو جزء من أجزاء النظام، بحيث

يظهر اسم الفئة على رأس البطاقة.

2) Responsibilities (المسؤولية): وتشمل جميع الخصائص التابعة لنفس الفئة بالإضافة

الى العمليات التي سيقوم مستخدم الفئة بتنفيذها، سواء كان هذا المستخدم شخص أو جزء

من أجزاء النظام، بحيث يتم وضع هذا الجزء في المنطقة اليسرى للبطاقة.

3) Collaboration (التعاون): وهي الفئات الأخرى التي تستخدم للحصول على معلومات أو

للقيام بمهام أخرى تابعة للفئة التي قامت باستدعائها، حيث يتم وضع الأجزاء المعاونة على

انها المهمة في المنطقة اليمنى للبطاقة.

وفيما يلي يتم توضيح جميع الفئات المستخدمة في المشروع الحالي:
بطاقة (4.4.A) المستخدم

Class Name: المستخدم	
Responsibilities	Collaboration
الاسم الاول اسم العائلة البريد الالكتروني كلمة المرور العنوان رقم الهاتف ----- تسجيل حساب تسجيل الدخول تسجيل الخروج ارجاع كلمة المرور اظهار بيانات المستخدم تعديل بيانات المستخدم	الرسائل العقارات مجموعات المستخدمين

بطاقة (4.4.B) مجموعات المستخدمين

Class Name: مجموعات المستخدمين	
Responsibilities	Collaboration
الرقم نوع المستخدم ----- ارجاع نوع المستخدم	الزبون مسؤول الموقع

بطاقة (4.4.C) العقارات

Class Name:العقارات	
Responsibilities	Collaboration
الوصف	المدينة
السعر	المنزل
المساحة	الارض
خطوط الطول	نوع العقار
خطوط العرض	المستخدم
رقم المستخدم	
رقم المدينة	

حفظ العقارات	
عرض العقارات	
عرض عقاراتي	
بحث عن عقار	

بطاقة (4.4.D) المنزل

Class Name:المنزل	
Responsibilities	Collaboration
رقم المنزل	العقارات
رقم الطابق	
المدينة	
عدد الغرف	
عدد الشرفات	
عدد الحمامات	
عدد المطابخ	

اضافة منزل	
تعديل معلومات منزل	
حذف منزل	

بطاقة (4.4.E) الأرض

Class Name:الأرض	
Responsibilities	Collaboration
رقم العقار نوع الارض ----- اضافة قطعة ارض تعديل معلومات قطعة الارض حذف قطعة الارض	العقارات نوع الارض

بطاقة (4.4.F) المدينة

Class Name:المدينة	
Responsibilities	Collaboration
رقم المدينة اسم المدينة	العقارات

بطاقة (4.4.G) نوع العقار

Class Name:نوع العقار	
Responsibilities	Collaboration
رقم العقار نوع العقار	العقارات

بطاقة (4.4.H) نوع الارض

Class Name:نوع الأرض	
Responsibilities	Collaboration
رقم الأرض نوع الأرض	الأرض

بطاقة (4.4.1) الرسالة

الرسالة: Class Name	
Responsibilities	Collaboration
رقم الرسالة رقم المرسل رقم المستقبل عنوان الرسالة محتوى الرسالة ----- ارسال رسالة الرد على الرسائل	المستخدم

Activity Diagram 4.5

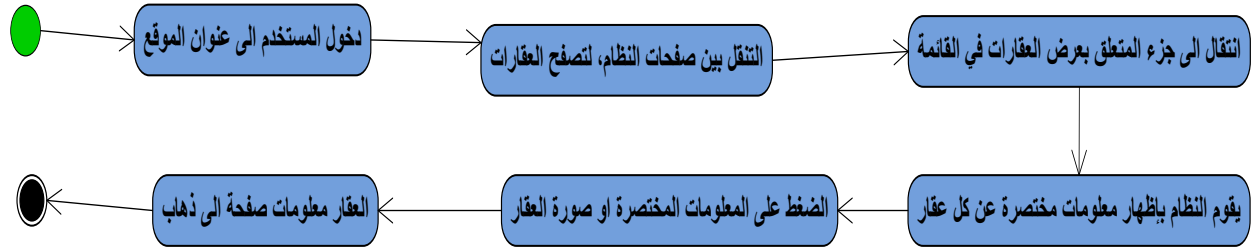
في هذا الجزء سنقوم بشرح حالات النظام وكيفية الانتقال من حالة الى أخرى، حيث يمكننا من معرفة كيفية تصرف أجزاء النظام مع بعضها البعض وكيف تنتقل من حالة الى أخرى.

4.5.1 مستخدم الموقع (مسؤول الموقع و الزبون والزائر)

هنا سيتم توضيح (Activity Diagram) المشتركة بين مسؤول الموقع و مستخدم النظام (الزبون) و

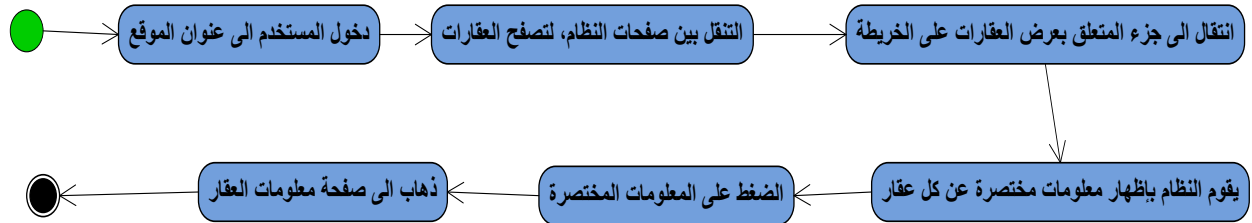
مستخدم النظام (الزائر).

يوضح شكل (4.5.1.1) كيفية تصفح واستعراض العقارات باستخدام القائمة، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون) ومستخدم النظام (الزائر).



شكل (4.5.1.1) تصفح واستعراض العقارات باستخدام القائمة

يوضح شكل (4.5.1.2) كيفية تصفح واستعراض العقارات باستخدام الخريطة، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون) ومستخدم النظام (الزائر).



شكل (4.5.1.2) تصفح واستعراض العقارات باستخدام الخريطة

يوضح شكل (4.5.1.3) كيفية البحث عن العقارات، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون) ومستخدم النظام (الزائر).



شكل (4.5.1.3) البحث عن العقارات

4.5.2 مستخدم الموقع (الزائر)

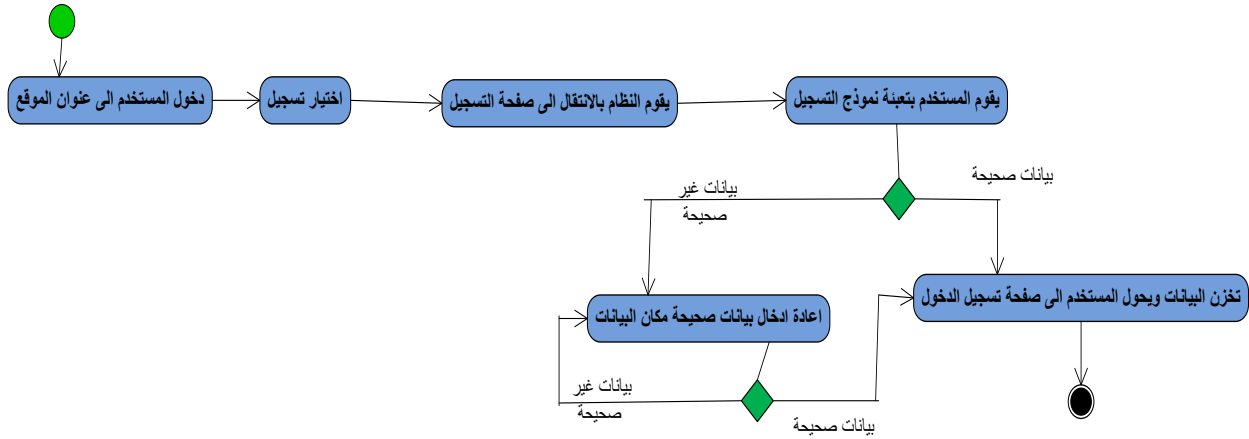
يستطيع مستخدم الموقع (الزائر) بالقيام بجميع المهام المشتركة مع مسؤول الموقع ومستخدم النظام

(الزبون) وهي:

- شكل (4.5.1.1) تصفح واستعراض العقارات باستخدام القائمة.
- شكل (4.5.1.2) تصفح واستعراض العقارات باستخدام الخريطة.
- شكل (4.5.1.3) البحث عن العقارات.

بالإضافة الى:

يوضح شكل (4.5.2.1) كيفية تسجيل حساب في الموقع، حيث يستطيع القيام بهذه العملية الزائر



شكل (4.5.2.1) تسجيل حساب

4.5.3 مستخدم الموقع (الزبون).

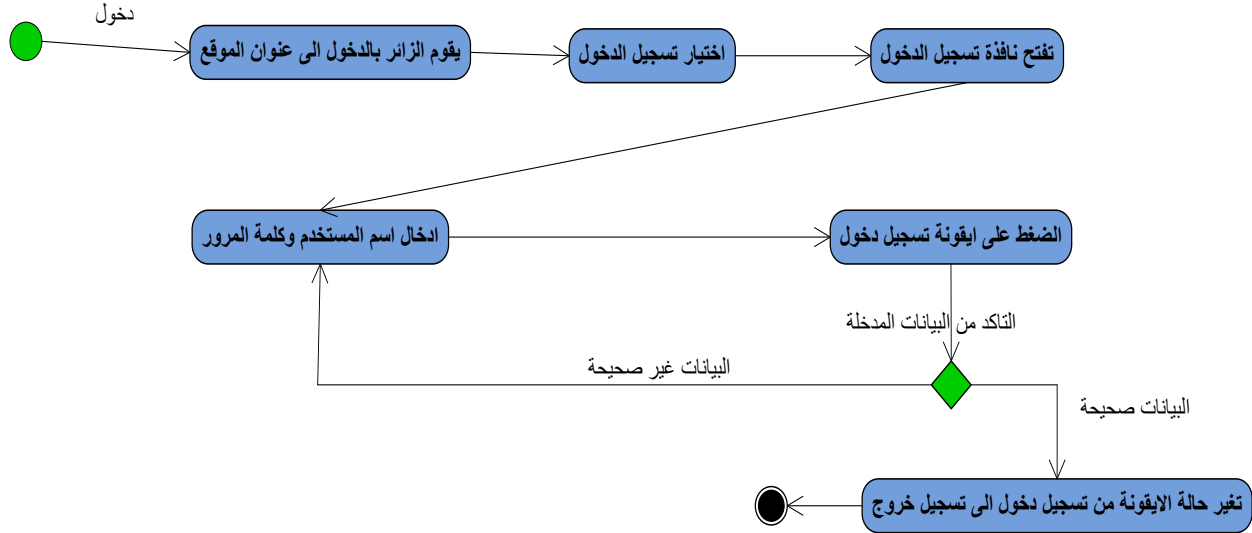
يستطيع مستخدم الموقع (الزبون) بالقيام بجميع المهام المشتركة مع مستخدم النظام (الزائر) ومسؤول

الموقع وهي:

- شكل (4.5.1.1) تصفح واستعراض العقارات باستخدام القائمة.
- شكل (4.5.1.2) تصفح واستعراض العقارات باستخدام الخريطة.
- شكل (4.5.1.3) البحث عن العقارات.

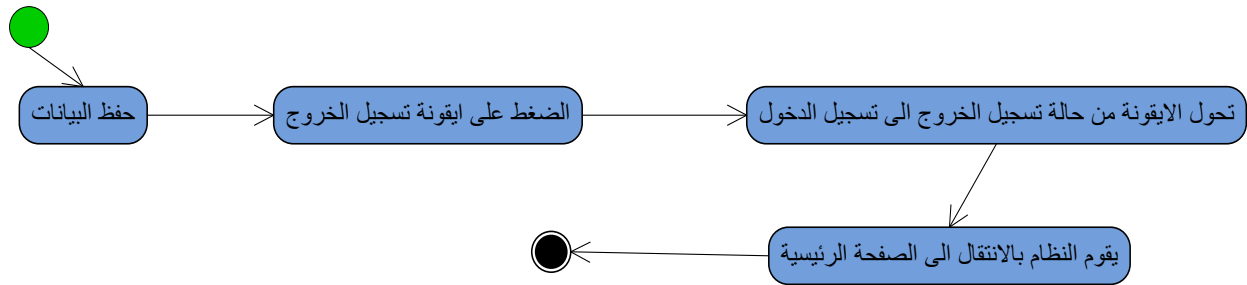
بالإضافة الى المهام التالية:

يوضح شكل (4.5.3.1) عملية تسجيل الدخول إلى الموقع، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



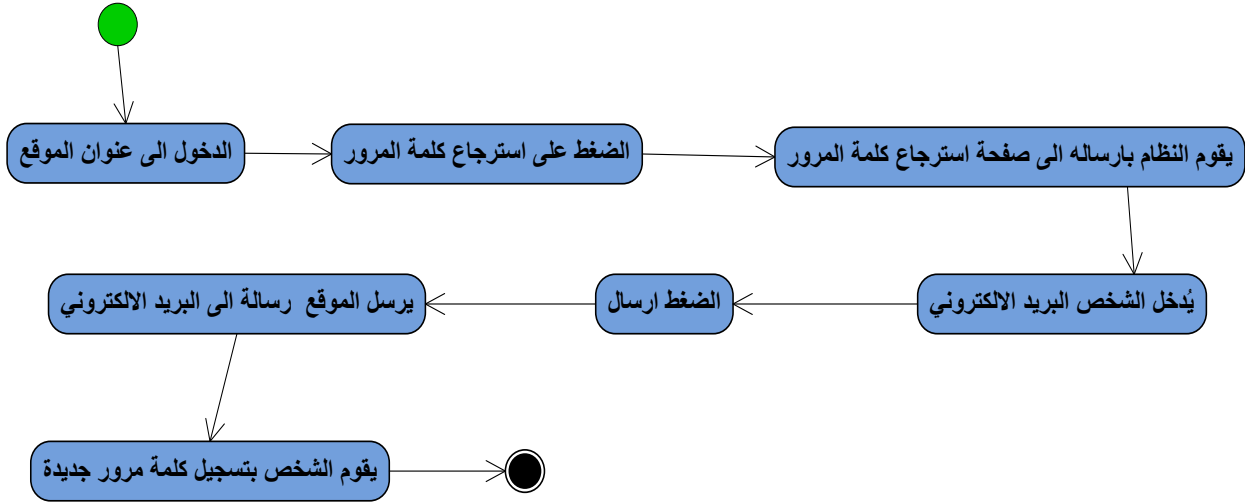
شكل (4.5.3.1) تسجيل دخول المستخدم إلى الموقع

يوضح شكل (4.5.3.2) عملية تسجيل الخروج من الموقع، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



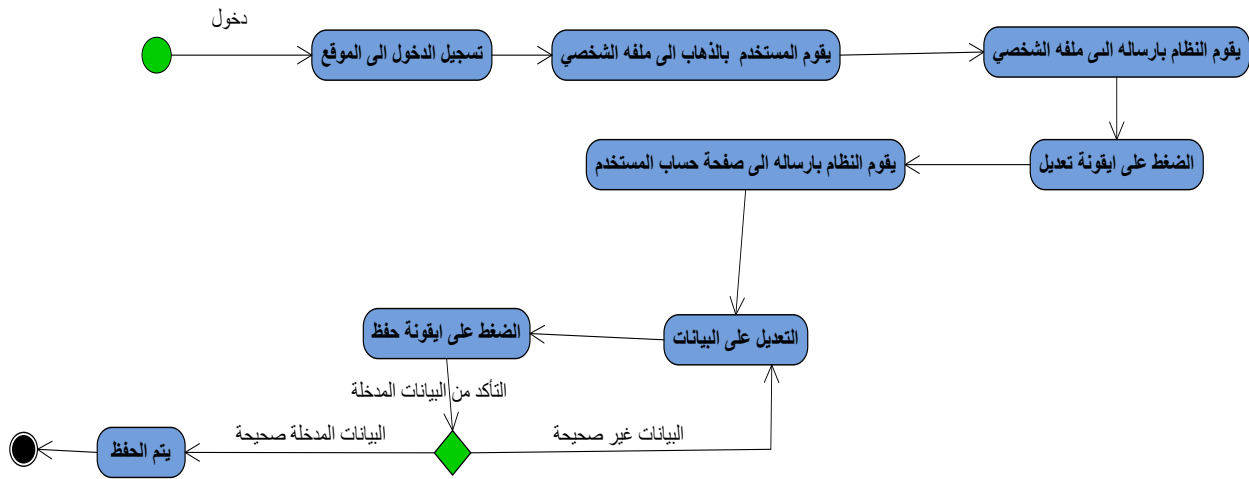
شكل (4.5.3.2) تسجيل خروج المستخدم من الموقع

يوضح شكل (4.5.3.3) عملية استرجاع كلمة المرور، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



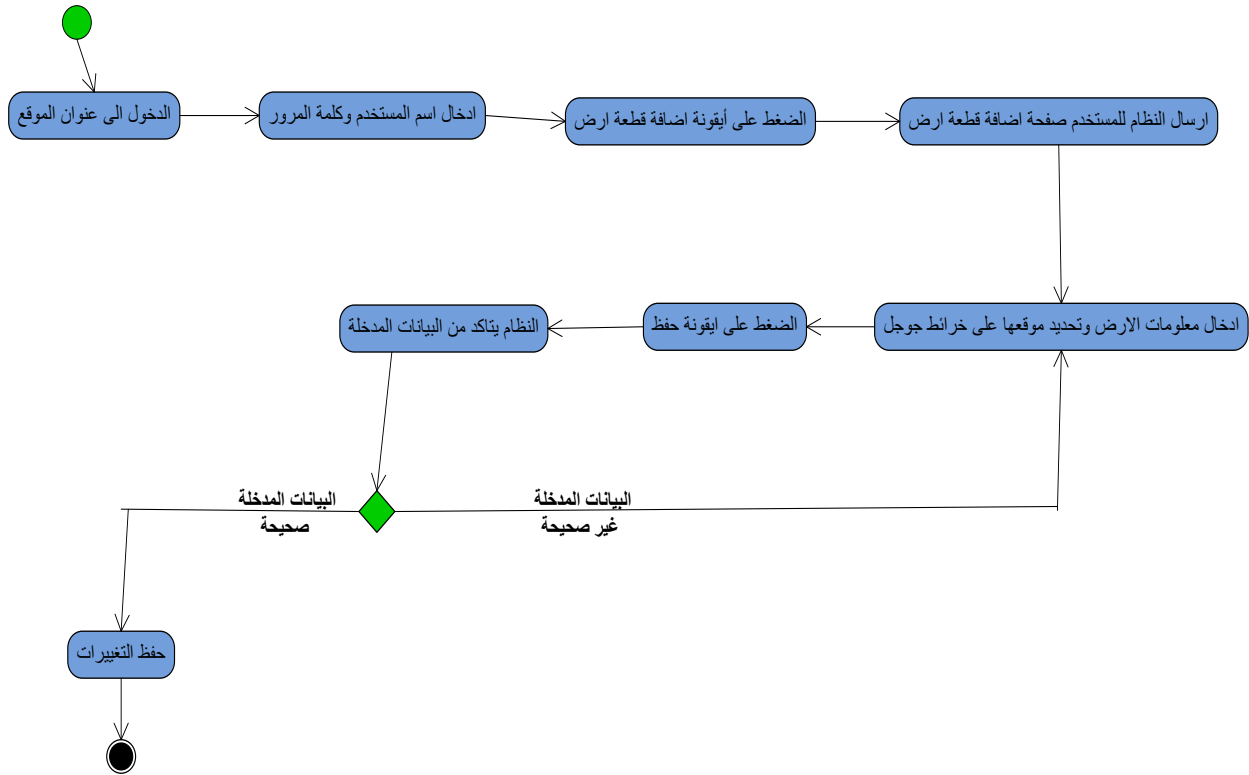
شكل (4.5.3.3) عملية استرجاع كلمة المرور

يوضح شكل (4.5.3.4) عملية تعديل البيانات في الموقع، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



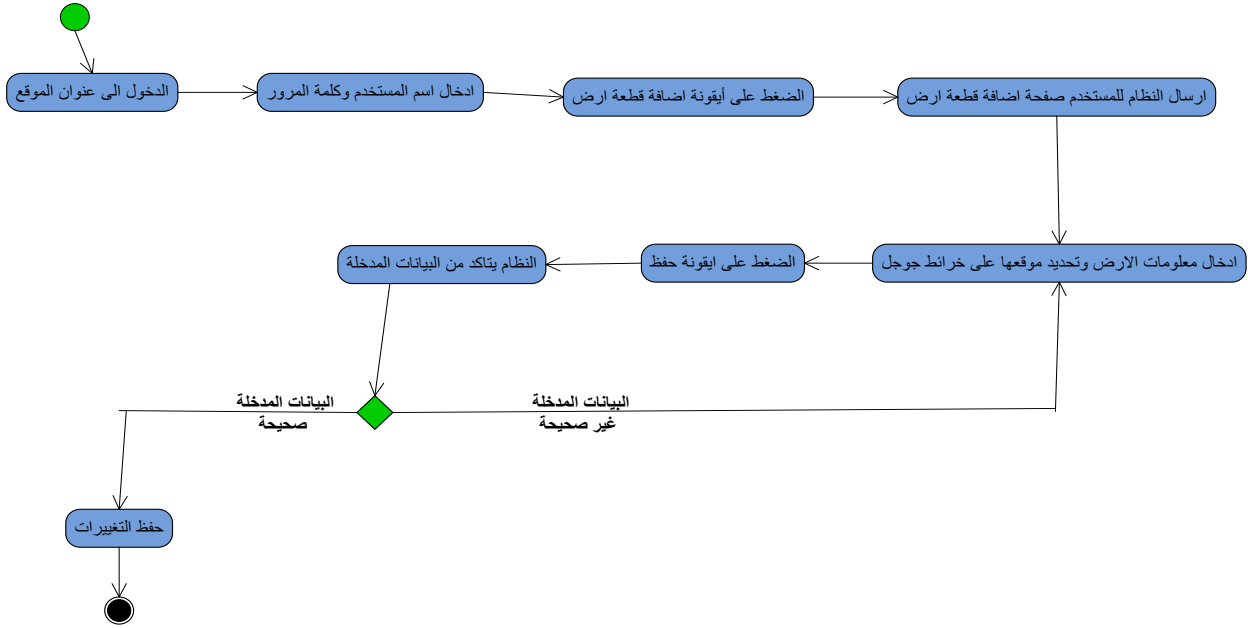
شكل (4.5.3.4) تعديل المستخدم لبياناته في الموقع

يوضح شكل (4.5.3.5) عمليّة اضافة قطعة أرض ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



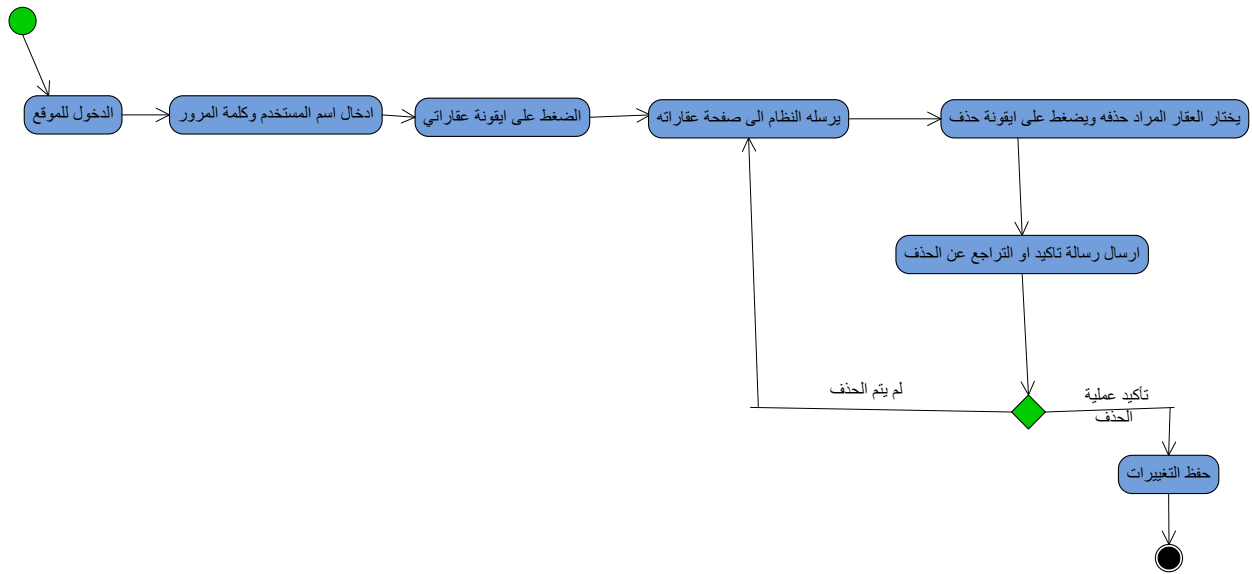
شكل (4.5.3.5) اضافة قطعة أرض

يوضح شكل (4.5.3.6) عمليّة اضافة منزل ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



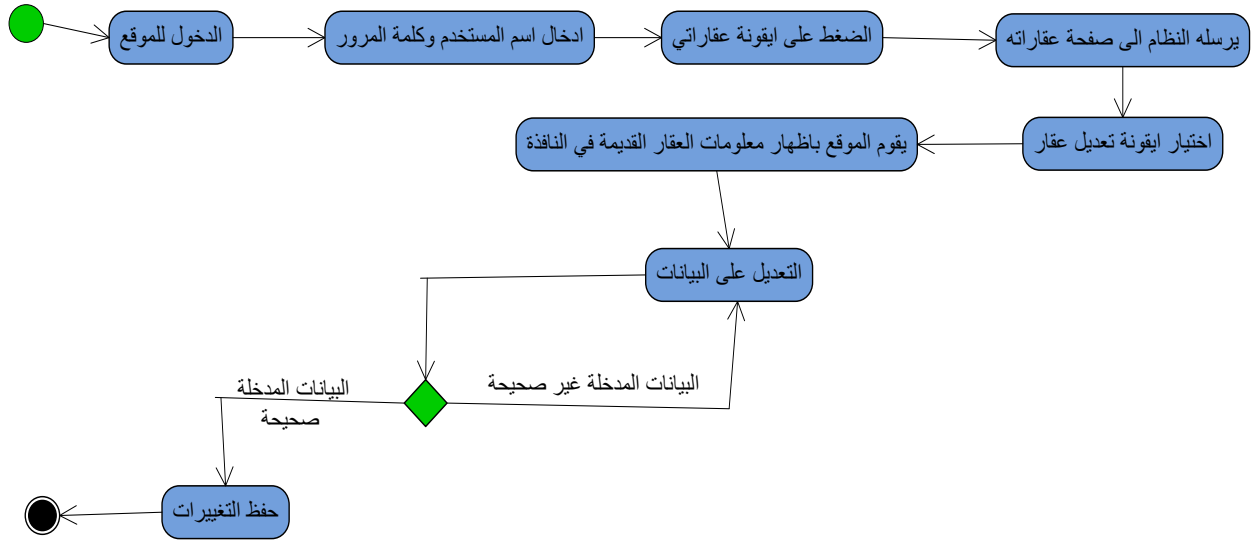
شكل (4.5.3.6) اضافة منزل

يوضح شكل (4.5.3.7) عملي حذف عقار ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



شكل (4.5.3.7) حذف عقار

يوضح شكل (4.5.3.8) عملية تعديل معلومات العقار ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



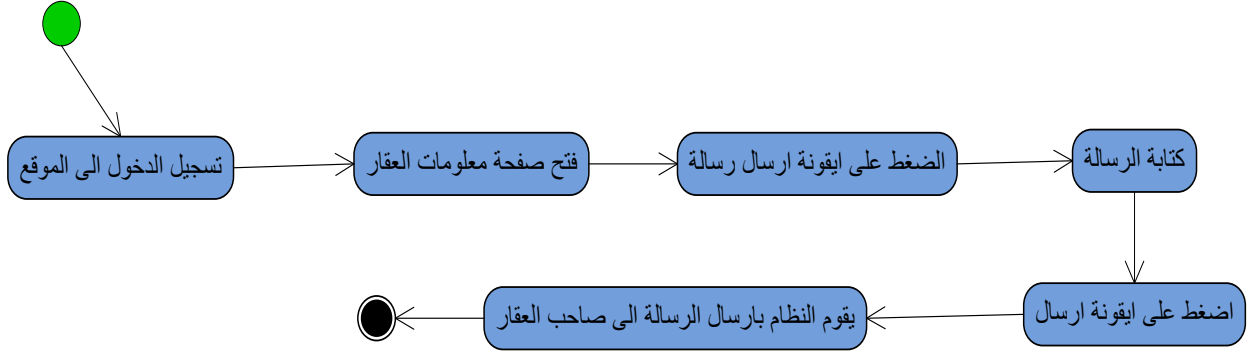
شكل (4.5.3.8) تعديل معلومات العقار

يوضح شكل (4.5.3.9) عملية تصفح الرسائل وادارتها ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



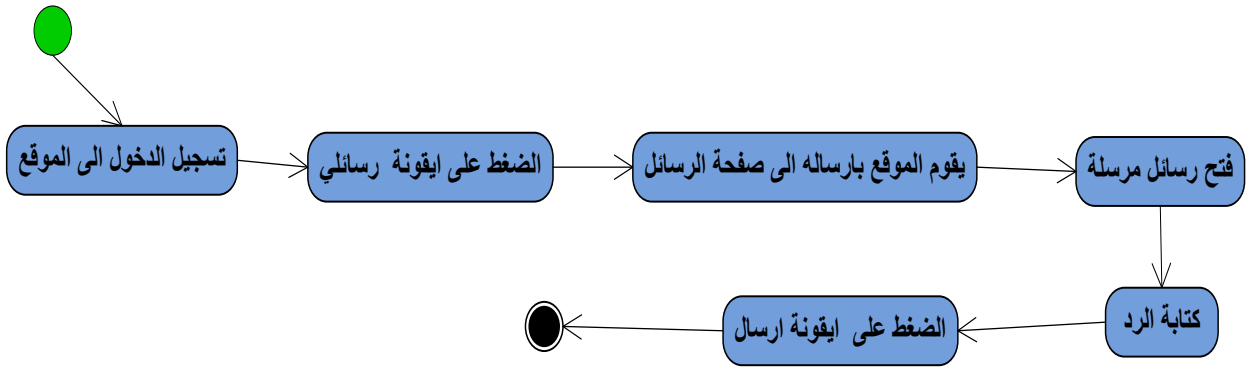
شكل (4.5.3.9) تصفح الرسائل وادارتها.

يوضح شكل (4.5.3.10) عملية ارسال رسالة الى صاحب العقار ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



شكل(4.5.3.10) ارسال رسالة الى صاحب العقار .

يوضح شكل (4.5.3.11) عمليّة الرد على رسالة ، حيث يستطيع القيام بهذه العملية كل من مسؤول النظام ومستخدم النظام (الزبون).



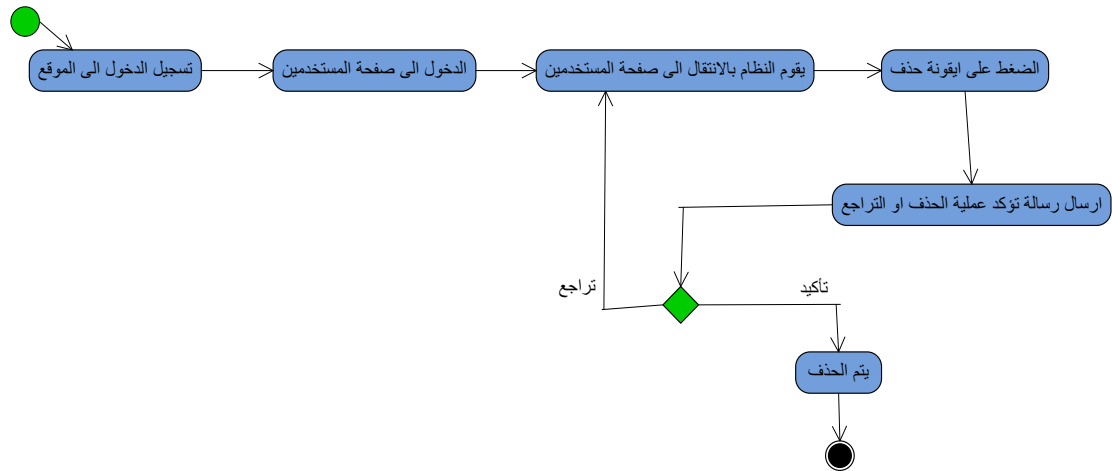
شكل(4.5.3.11) الرد على رسالة.

4.5.4 مسؤول الموقع

يستطيع مسؤول الموقع بالقيام بجميع المهام التي يقوم بها مستخدم النظام (الزائر او

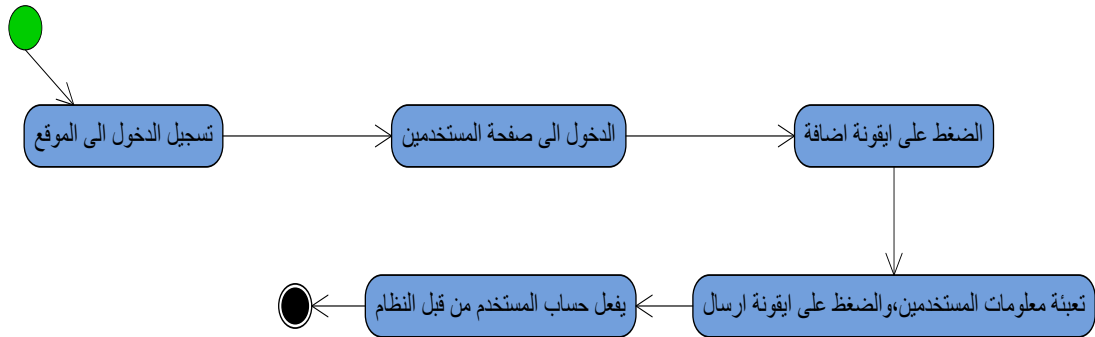
الزبون) بالاضافة الى المهام التالية:

يوضح شكل (4.5.4.1) عمليّة حذف مسجلين بالموقع ، حيث يستطيع القيام بهذه العملية مسؤول النظام.



شكل (4.5.4.1) عملية حذف مستخدم من الموقع

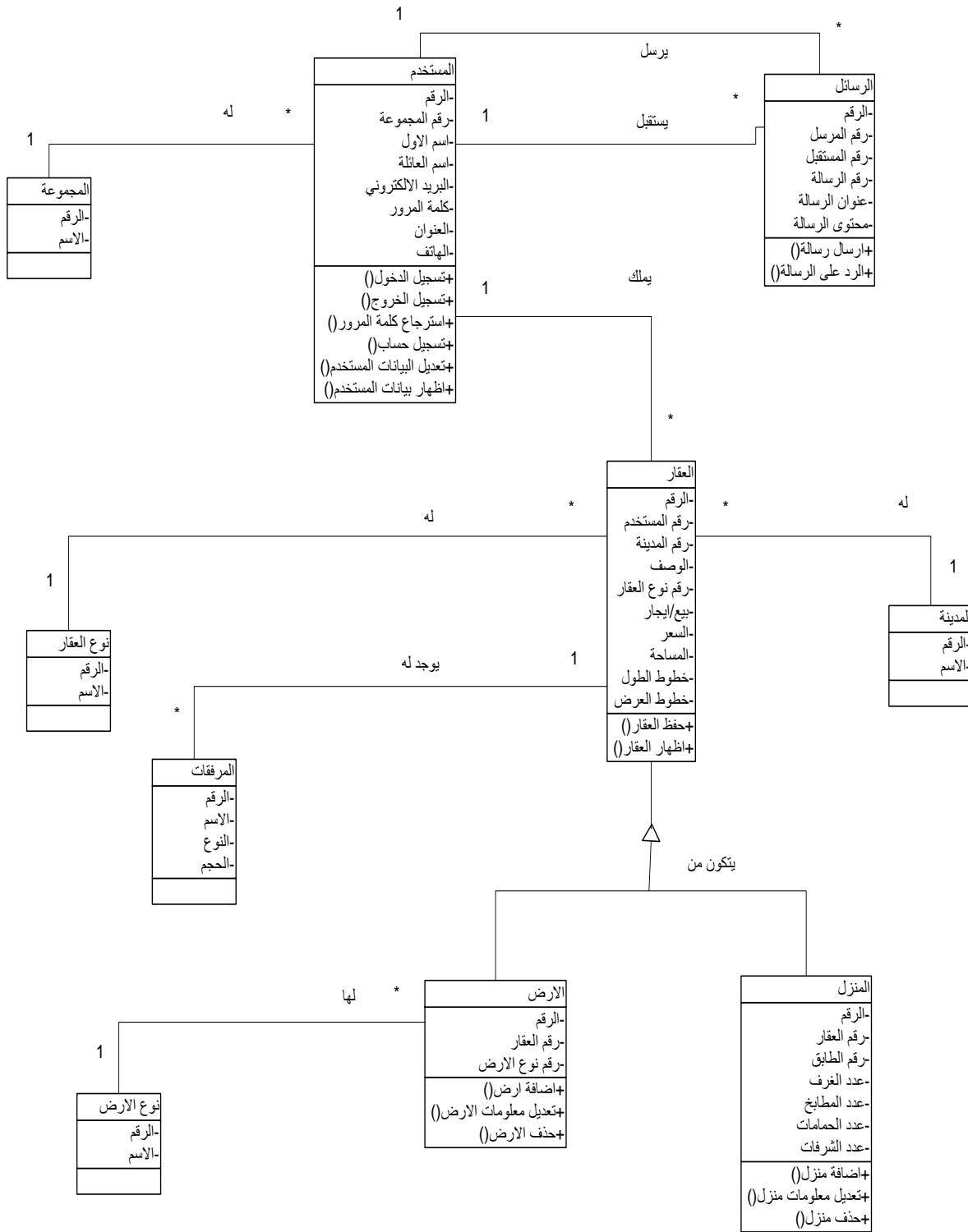
يوضح شكل (4.5.4.2) عملية اضافة مستخدمين الى الموقع، حيث يستطيع القيام بهذه العملية مسؤول النظام.



شكل (4.5.4.2) اضافة مستخدم الى الموقع

4.6 Class Hierarchies and Subsystems

يوضح الشكل التالي Class Hierarchies ويوضح طبيعة العلاقات بينها.



الفصل الخامس

تصميم النظام

❖ مقدمة.

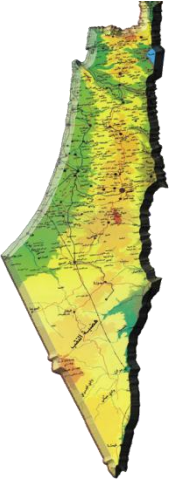
❖ Object- Relational Model

❖ State Behavioral Modeling

❖ Subsystem Design (Tasks and Components)

❖ Class and Object Design

❖ Interface Design (System Messages)

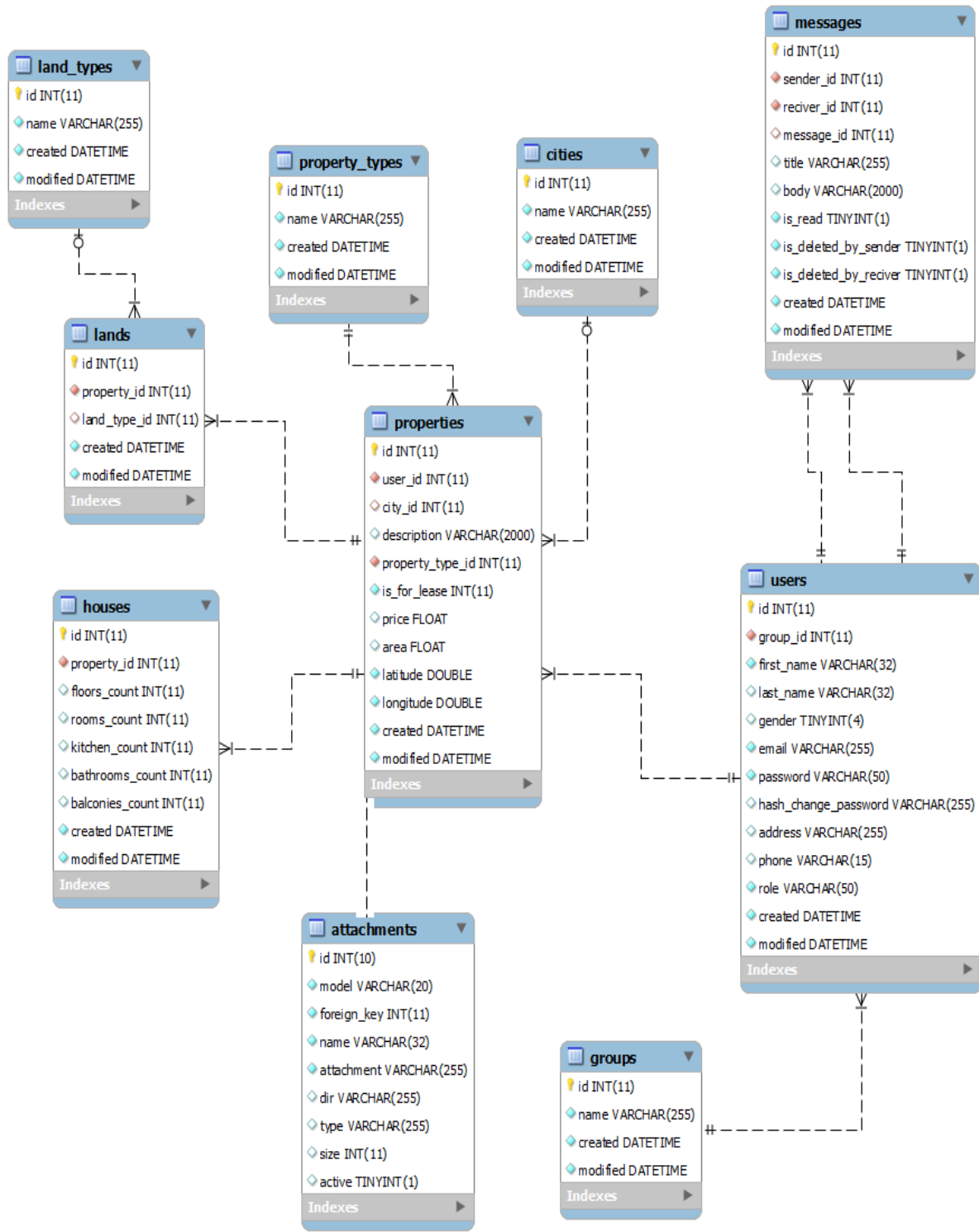


5.1 المقدمة

في هذا الفصل سيتم توضيح أجزاء النظام على شكل كيانات والعلاقات بين أجزاء النظام الأساسية والفرعية، بحيث سيتم وضع هذه الأجزاء على شكل مغلفات تدعى (Deployments) ومكونات المغلف الفرعية تسمى (Components)، كما وسيتم إعطاء تفصيل توضيحي لكل جزء فرعي من المغلف من حيث العمليات التي ستتم عليه وخصائصه.

Object-Relational Model 5.2

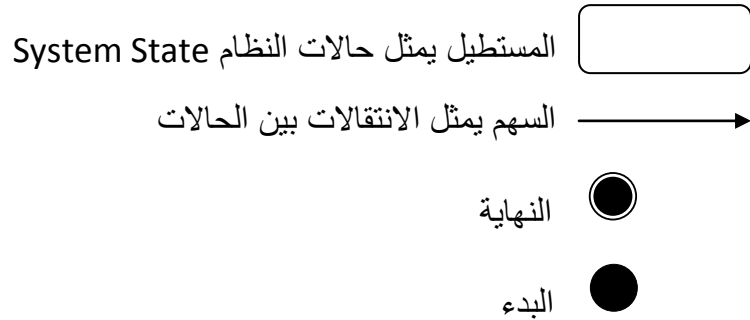
يسمح للمبرمجين بإنشاء قواعد بيانات مع البيانات الخاصة فيها وأشكالها وأنواعها. سيتم في هذا الجزء توضيح ارتباطات الاصناف مع بعضها البعض كما في الشكل الآتي:



شكل 5.2.1 مخطط قاعدة البيانات

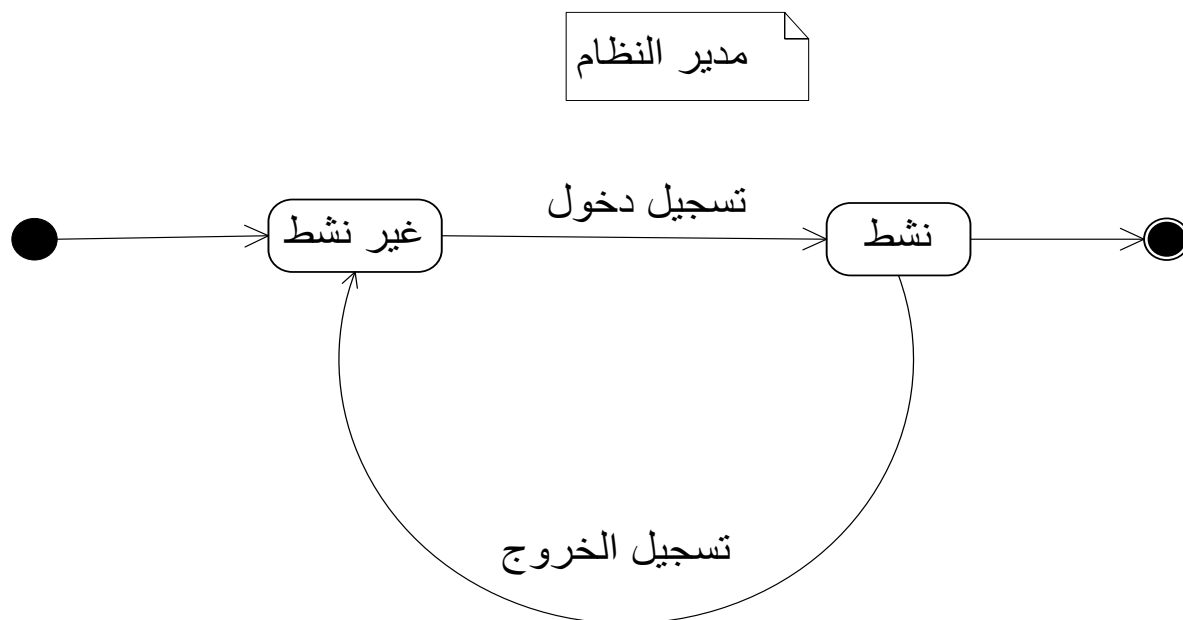
State Behavioral Modeling 5.3

هو مبدأ عمل اساسي لجميع طرق تحليل المتطلبات، حيث ينتج عنه نموذج يوضح ردة فعل الـ Software للأحداث الخارجية. ان الـ State Transition Diagram يمثل سلوك النظام وذلك برسم حالاته والأحداث التي تسبب تغيير في تلك الحالات. إضافة لذلك فهو يشير إلى ما سيكون الفعل التالي كنتيجة لحدث معين.



مدير النظام (تسجيل الدخول والخروج من النظام) ➤

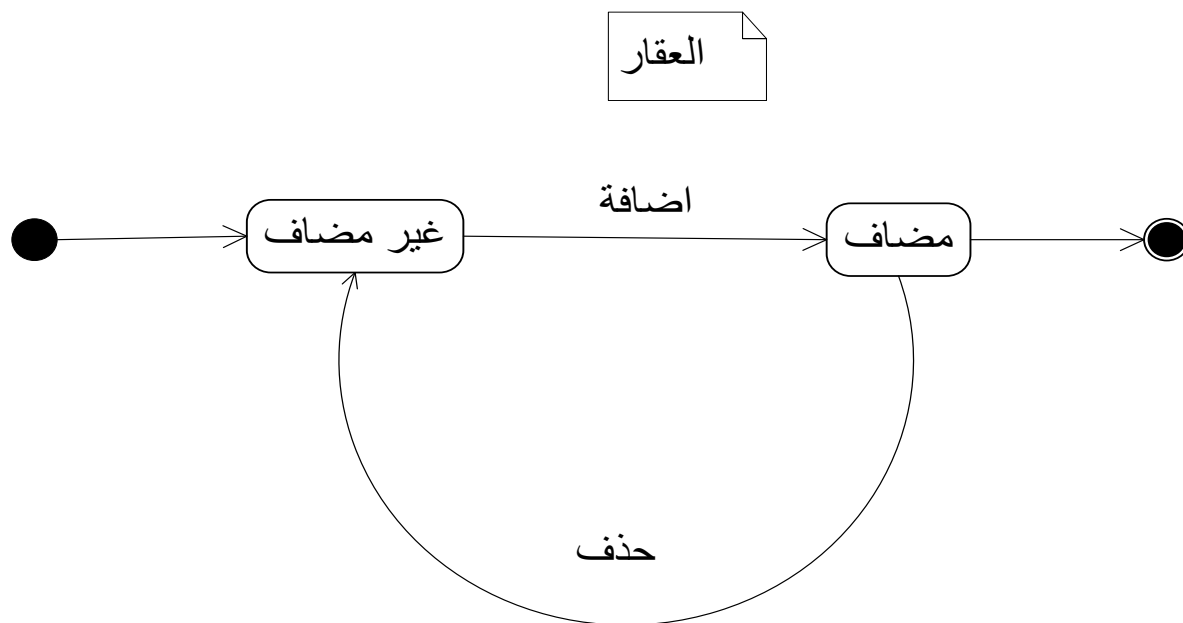
في حال كون حال مدير النظام غير نشط يستطيع تسجيل الدخول والانتقال إلى حالة النشط
أما في حال كون حالة مدير النظام نشط يستطيع تسجيل الخروج والانتقال إلى حالة غير النشط.



شكل (5.3.1) تسجيل دخول مدير النظام

العقار (إضافة وحذف عقار) ➤

في حال كون حالة العقار غير مضاف يمكن اضافته لينتقل الى حالة العقار المضاف.
 اما في حال كون حالة العقار مضاف يمكن حذفه لينتقل الى حالة العقار الغير مضاف.

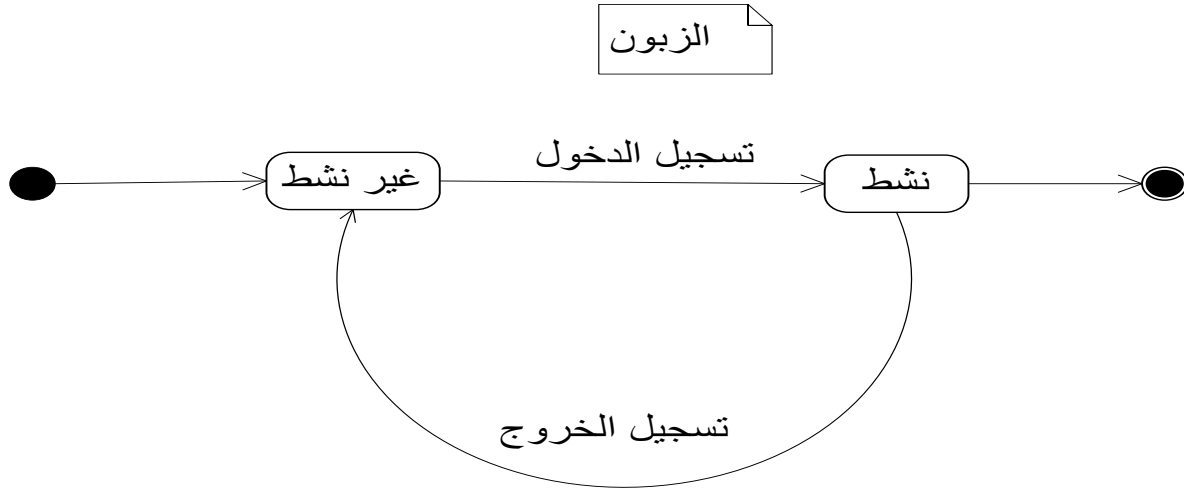


شكل (5.3.2) عملية اضافة/حذف عقار

الزبون (تسجيل الدخول والخروج من النظام) ➤

في حال كون حال الزبون غير نشط يستطيع تسجيل الدخول والانتقال الى حالة النشط

اما في حال كون حالة الزبون نشط يستطيع تسجيل الخروج والانتقال الى حالة غير النشط.



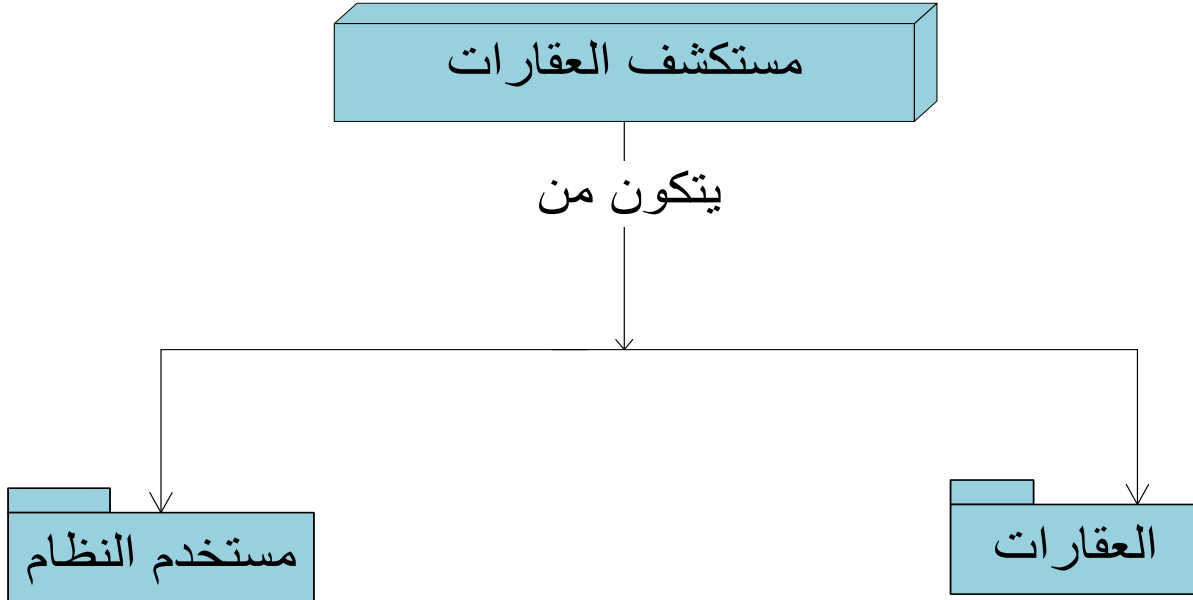
شكل (5.3.3) تسجيل دخول الزبون

Subsystem Design (Tasks and Components) 5.4

سيتم في هذا الفصل توضيح النموذج الذي يوضح ال (packages) والأصناف (Classes)

والعلاقات ما بين هذه الأصناف (Classes) وما تحويه من سمات (Attributes) للنظام، حيث أن

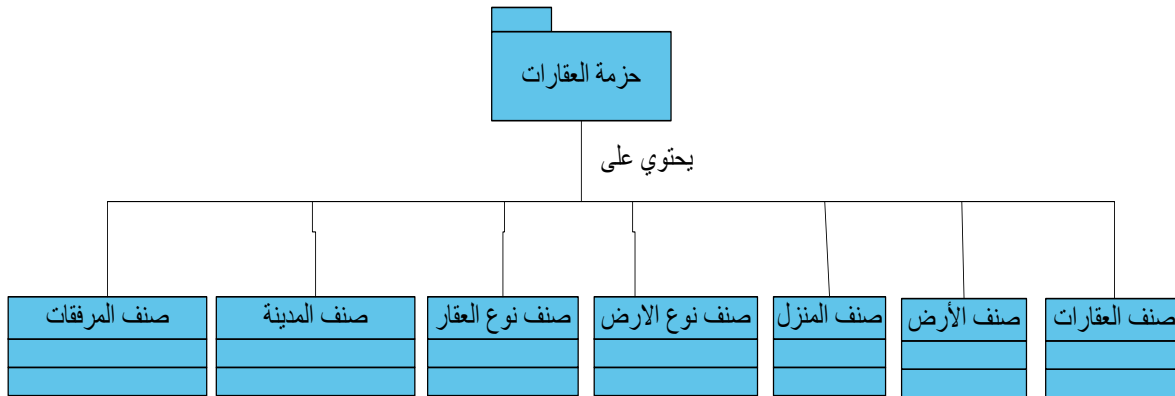
النظام يحوي عدد من ال (Packages) بداخلها الأصناف (Classes) .



شكل (5.4.1) نظام مستكشف العقارات

5.4.1 حزمة العقارات (Aqars Package)

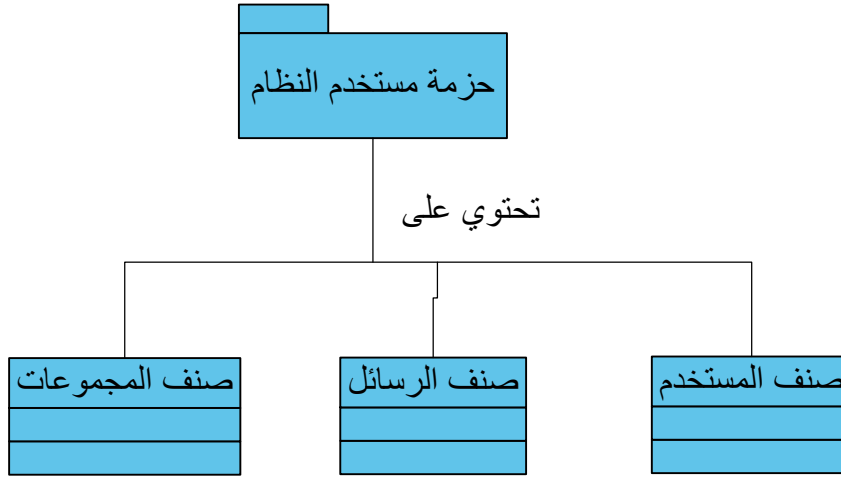
يحتوي هذا الجزء على مجموعة من (Class) وهي صنف العقارات وصنف الأرض وصنف المنزل وصنف نوع الارض وصنف نوع العقار وصنف المدينة وصنف المرفقات. سيتم التعامل في هذا الجزء مع العمليات المختلفة التي يتم تطبيقها على العقار من ناحية الإضافة والتعديل والبحث عن بيانات العقار التي تم تسجيلها في قاعدة البيانات.



شكل (5.4.1.1) حزمة العقارات

5.4.2 حزمة المستخدم (Users Package)

يتم التعامل في هذا الجزء مع العمليات المختلفة التي يتم تطبيقها على مستخدمي النظام (المسؤول النظام، الزبون، الزائر)، من ناحية الإضافة والتعديل والبحث عن معلومات مستخدمي النظام التي تم تسجيلها والتي سيتم تسجيلها أو حذفها من قاعدة البيانات، وايضا التواصل بين المستخدمين عن طريق الرسائل (ارسال رسالة والرد على رسالة وحذف رسالة).



شكل (5.4.2.1) حزمة مستخدم النظام

Class and Object Design 5.5

في هذا الفصل سيتم شرح كل الفئات التي تم استخدامها في هذا المشروع، وكما سيتم تحديد العلاقات بين هذه الفئات والتغيرات المستخدمة فيها والعمليات التابعة لها.

Users Class ➤

يسمح لزبون او مستخدم النظام تعديل او اضافة او حذف عقار، ويسمح ايضا بتعديل معلومات المستخدم او اضافة مستخدم الى النظام.

user
-id : int -group_id : int -first_name : varchar -last_name : varchar -email : varchar -password : varchar -hash_change_password : varchar -address : varchar -phone : varchar -created :date -modified :date
+Login() +Logout() +remember_password() +register() +View() +edit()

◆ **صنف المستخدمين موضح كالتالي:**

: User Attributes (1

User_ID : Integer, Private •

هو عبارة عن رقم الزبون، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر

لتمييز كل زبون عن الآخر، إضافة إلى تسهيل عملية البحث والتعديل على بيانات الزبون.

User_Group_ID : Integer, Private •

المجموعة التي ينتمي لها المستخدم حيث يحدد كل من الزبون ومسؤول الموقع ويحدد صلاحيات

كل منهما .

• User_First Name : Varchar, Private

هو عبارة عن الاسم الاول للمستخدم، يقوم المستخدم بادخاله عند عملية التسجيل في الموقع، ويظهر هذا اسم بعد نجاح عملية الدخول الى الموقع على الصفحة الرئيسية وايضا يظهر في ملفه الشخصي في الموقع .

• User_Last Name : Varchar, Private

هو عبارة عن اسم العائلة للمستخدم، حيث يقوم المستخدم بادخاله عند عملية التسجيل في الموقع، ويظهر هذا اسم بعد نجاح عملية الدخول الى الموقع على الصفحة الرئيسية وايضا يظهر في ملفه الشخصي في الموقع .

• User_Email : Varchar, Private

هو عبارة عن اسم المستخدم (البريد الالكتروني)، حيث يقوم المستخدم في ادخاله عند عملية التسجيل في الموقع، وهو المفتاح الاساسي الذي يجب أن لا يتكرر، بحيث لا يمكن ان يكون لمستخدم النظام أكثر من بريد الكتروني واحد متشابهين في نفس الجدول في قاعدة البيانات، لضمان نجاح عملية تسجيل الدخول الى الموقع من خلال قراءة اسم المستخدم، يستخدم البريد الالكتروني ايضا في مساعدة المستخدم في استبدال كلمة المرور الخاصة به في حال تم فقدانها، وايضا يساعد في التواصل بين اصحاب العقارات عن طريق الرسائل الخاصة في الموقع.

• User_Password : Varchar, Private

هو مجموعة من الرموز يعينها كل مستخدم (زبون او مسؤول نظام) عند تسجيله في النظام، والتي تستخدم عند عملية تسجيل الدخول، تخزن في قاعدة البيانات، وتستطيع ان تستبدلها بمجموعة من الرموز الجديدة في حال تم فقدانها.

User_Hash_Change_Password : Varchar, Private •

هي عبارة عن مجموعة من الرموز الجديدة الخاصة في المستخدم، يقوم المستخدم في ادخاله عند عملية استرجاع كلمة المرور في الموقع، يتم ادخال الرموز الجديدة وايضا ادخال تأكيد هذه الرموز، حيث يتم تخزينها في قاعدة البيانات مكان الرموز السابقة، بعد ذلك يطلب النظام ادخالها في عملية تسجيل الدخول الى الموقع، لضمان نجاح عملية تسجيل الدخول الى الموقع من خلال قراءة كلمة المرور، وايضا في حال تم فقدانها يتم استرجاعها بنفس الطريقة.

User_Address : Varchar, Private •

هو عبارة عن موقع المستخدم الحالي، حيث يسهل عملية التواصل بين مستخدمي النظام في حال اراد احد مستخدمي النظام ان يتعرف على صاحب العقار وان يتواصل معه.

User_Phone : Varchar, Private •

هو من احد الوسائل التي يستخدمها الشخص لتواصل مع اصحاب العقارات في الموقع.

User_Created : DateTime, Private •

هو عبارة عن تاريخ تسجيل الشخص في الموقع، حيث يملك كل زبون تاريخ تسجيله في الموقع.

User_Modified: DateTime, Private •

هو عبارة عن تاريخ اخر تعديل تم على معلومات الزبون.

System Users Operation (2)

Login() : void ➤

يقوم الزبون بالدخول الي الموقع من خلال الواجهات الخاصة بالدخول الى الموقع، ويتم ادخال البيانات التي تلزم لتسجيل الدخول، ويقوم النظام بفحص البيانات المدخلة من قبل الشخص، في حال تم تطابق البيانات المدخلة مع البيانات الموجودة في قاعدة البيانات يتم تسجيل الدخول الى الموقع.

Logout() : void ➤

يقوم الزبون بالضغط على ايقونة تسجيل الخروج ليتحول من حالة نشط الى حالة غير نشط.

remember_Password() : void ➤

يستخدم لاستعادة كلمة المرور المفقودة.

register() : void ➤

يقوم الزائر باضافة حساب جديد في الموقع من خلال الواجهات الخاصة باضافة حساب جديد،

ويتم حفظ البيانات في قاعدة البيانات في حال كانت البيانات المدخلة صحيحة، اما في حال كون

البيانات المدخلة غير صحيحة، يطلب النظام من الزائر ادخال البيانات الصحيحة.

View() : void ➤

يستخدم لاطهار معلومات المستخدم في الصفحة الشخصية.

edit () : void ➤

يقوم الزبون في استرجاع البيانات القديمة التي تم سابقا حفظها في قاعدة البيانات، ثم يقوم الزبون

بتعديل معلوماته الخاصة فيه، من خلال الواجهات الخاصة بتعديل معلومات الزبون، حيث يتم حفظ

تعديلات معلومات الزبون في قاعدة البيانات.

Aqars Class 

يساعد المستخدم على اضافة عقار، واسترجاع عقار عن طريق البحث عنه.

Aqars
-id : int
-user_id : int
-city_id : int
-description : varchar
-Aqar_type_id : int
-is_for_lease : int
-price : float
-area : float
-latitude : double
-longitude : double
-created : DateTime
-modified : DateTime
+myproperty()
+saveProperty()
+explore()

❖ صنف العقارات موضح كالتالي:

(1 : Aqar Attributes

• Aqars_ID : Integer, Private

هو عبارة عن رقم العقار، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر

لتمييز كل عقار عن الآخر، إضافة إلى تسهيل عملية البحث والتعديل على بيانات العقار.

• Aqars_user_id : Integer, Private

هو عبارة عن رقم تسلسلي يربط العقار مع صاحب العقار، يسهل عمليات البحث والتعديل .

• Aqars_city_id : Integer, Private

هو عبارة عن رقم، حيث أنه رقم تسلسلي يربط العقار مع المدينة التي يوجد بها العقار، حيث

يسهل عملية البحث على العقارات ضمن مدينة محددة .

• **Aqars_description : Integer, Private**

هو عبارة عن وصف العقار، حيث يتم وصفه وصفا تفصيليا، حيث يطلب النظام اضافة الوصف للعقار عند اضافة العقار، من خلال الواجهة الخاصة باضافة العقارات، ويتم تخزينها في قاعدة البيانات، عند البحث عن العقارات يظهر وصف العقار بالواجهة الخاصة بإظهار معلومات العقار، يستطيع صاحب العقار التعديل على الوصف المدخل من قبله.

• **Aqars_Aqar_type_id : Integer, Private**

هو عبارة عن رقم تسلسلي يربط العقار مع نوع العقار (قطعة أرض، منزل)، يساعد على ارجاع نوع العقار بسهولة .

• **Aqars_is_for_lease : Integer, Private**

هو عبارة عن تحديد العقار المعروض في الموقع، العقار للبيع ام الاستئجار، حيث يتم تخزينه في قاعدة البيانات، وهذه احد البيانات التي يطلب من الزبون ادخالها عند عملية اضافة العقار، لتسهيل عملية البحث عن العقارات المناسبة لكل من الزبون والزائر.

• **Aqars_price : Float, Private**

هو عبارة عن سعر العقار حيث يقوم المستخدم بادخاله عند عملية اضافة العقار، يتم تخزينه في قاعدة البيانات، يظهر للمستخدم عند عملية البحث عن العقار، يستطيع صاحب العقار التعديل على السعر في حال أراد ذلك .

• **Aqars_area : Float, Private**

هي عبارة عن مساحة العقار حيث يقوم المستخدم بادخالها عند عملية اضافة العقار، يتم تخزينها في قاعدة البيانات، تظهر للمستخدم عند عملية البحث عن العقار.

Aqars_latitude : Double, Private •

هي عبارة عن خطوط العرض، تستخدم لتحديد موقع العقارات، وكما تساعد في تحديد موقعك الحالي.

Aqars_longitude : Double, Private •

هي عبارة عن خطوط الطول، تستخدم لتحديد موقع العقارات، وكما تساعد في تحديد موقعك الحالي.

Aqars_created : DateTime, Private •

هو عبارة عن تاريخ اضافة العقار، حيث كل عقار يملك تاريخ انشاء له .

Aqars_modified : DateTime, Private •

هو عبارة عن اخر تاريخ تم فيه التعديل على العقار.

System Aqars Operation (2)

myproperty() : void ➤

يستطيع الزبون رؤية جميع عقاراته في صفحة عقاراته الشخصية، ويستطيع التعديل على بيانات عقاراته او حذف العقار اذا اراد ذلك .

saveProperty () : void ➤

يقوم مستخدم النظام بادخال معلومات العقار وهي المدينة التي يقع بها العقار و وصف العقار ونوع العقار وسعره ومساحته وموقعه على خرائط جوجل، وتحديد العقار هل هو للبيع ام الايجار وارقاق مرفقات العقار(صور عن العقار) من خلال الواجهات الخاصة باضافة هذه المعلومات، ويتم تخزين البيانات في قاعدة البيانات.

explore () : void ➤

يقوم مستخدم النظام بادخال معلومات العقار وهي المدينة و وصف العقار ونوع العقار وسعره ومساحته وموقعه على خرائط جوجل، وتحديد العقار هل هو للبيع ام الايجار من خلال الواجهات الخاصة باضافة هذه المعلومات، ويتم البحث عن البيانات المدخلة في قاعدة البيانات، في حال تم تطابق البيانات المدخلة من قبل مستخدم النظام مع بيانات مخزنة في قاعدة البيانات، تظهر نتائج البحث عن العقار على القائمة وعلى خرائط جوجل.

Houses Class ➤

يساعد المستخدم على اضافة عقار من نوع منزل مع امكانية التعديل على العقار

او حذفه.

Houses
-id : int
-Aqar_id : int
-floors_count : int
-rooms_count : int
-kitchen_count : int
-bathrooms_count : int
-balconies_count : int
-created : DateTime
-modified : DateTime
+addHouse()
+editHouse()
+deleteHouse()

❖صنف المنازل موضح كالتالي:

(1 Houses Attributes

Houses _ID : Integer, Private •

هو عبارة عن رقم المنزل، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر لتميز كل منزل عن الآخر، إضافة إلى تسهيل عملية البحث والتعديل على بيانات المنزل.

Houses _ Aqar_id: Integer, Private •

هو عبارة عن ربط رقم العقار بنوعه (منزل)، حيث يتم اختيار نوع العقار بناء على رقم العقار المخزن في جدول قاعدة البيانات.

Houses _ floors_count : Integer, Private •

هو عبارة عن رقم الطابق الذي يحتوي هذا المنزل، يطلب من الزبون ادخاله عند اضافة العقار (المنزل)، ويظهر امام المستخدمين بعد عملية البحث.

Houses _ rooms_count : Integer, Private •

هو عبارة عن عدد الغرف في المنزل، يطلب من الزبون ادخالها عند اضافة العقار (المنزل)، ويظهر امام المستخدمين بعد عملية البحث.

Houses _ kitchen_count : Integer, Private •

هو عبارة عن عدد المطابخ في المنزل، يطلب من الزبون ادخالها عند اضافة العقار (المنزل)، ويظهر امام المستخدمين بعد عملية البحث.

Houses _ bathrooms_count : Integer, Private •

هو عبارة عن عدد الحمامات في المنزل، يطلب من الزبون ادخالها عند اضافة العقار (المنزل)، ويظهر امام المستخدمين بعد عملية البحث.

Houses _ balconies_count : Integer, Private •

هو عبارة عن عدد البالكونات (الشرفات) في المنزل، يطلب من الزبون ادخالها عند اضافة العقار (المنزل)، ويظهر امام المستخدمين بعد عملية البحث.

Houses _ created : DataType, Private •

هو عبارة عن تاريخ انشاء هذا العقار (عقار المنزل).

Houses _ modified : DataType, Private •

هو عبارة عن اخر تعديل تم على هذا العقار.

System Properties Operation (2)

addHouse () : void ➤

يقوم مستخدم النظام (الزبون او مسؤول الموقع) بوضع معلومات العقار (المنزل)، وهي رقم الطابق وعدد الغرف وعدد المطابخ وعدد الحمامات وعدد البالكونات، وتحديد العقار (المنزل) هل هو للبيع ام الايجار، باضافة الى اختيار المدينة و وصف العقار ونوع العقار وسعره ومساحته، من خلال الواجهات الخاصة باضافة هذه المعلومات، حيث تخزن هذه المعلومات في قاعدة البيانات.

editHouse() : void ➤

يستطيع مستخدم النظام (الزبون أو مسؤول الموقع أو الزائر) الحصول على معلومات العقار وهي رقم الطابق وعدد الغرف وعدد المطابخ وعدد الحمامات (الشرفات)، وتحديد العقار (المنزل) هل هو للبيع ام الايجار، وايضا اختيار المدينة و وصف العقار ونوع العقار وسعره ومساحته من خلال الواجهات الخاصة بعرض هذه المعلومات، حيث يستطيع المستخدم التعديل على هذه المعلومات والضغط على ايقونة حفظ ، وبذلك تحفظ البيانات الجديدة في قاعدة البيانات، وتظهر هذه البيانات امام مستخدمي النظام.

deleteHouse () : void ➤

يعمل على حذف العقار من نوع منزل، حيث يتم ازالة معلومات العقار المضاف سابقا من قاعدة البيانات.

Lands Class ➤

يساعد المستخدم على اضافة عقار من نوع ارض مع امكانية التعديل على العقار او حذفه.

Lands
-id : int
-Aqar_id : int
-land_type_id : int
-created : DataType
-modified : DataType
+addLand()
+editLand()
+deleteLand()

❖صنف الأراضي موضح كالتالي:

: Lands Attributes (1)

Lands _ID : Integer, Private •

هو عبارة عن رقم الأرض، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن

يتكرر لتميز كل قطعة ارض عن الأخرى، إضافة إلى تسهيل عملية البحث والتعديل على بيانات هذه القطعة.

Lands _ Aqar_id: Integer, Private •

هو عبارة عن ربط رقم العقار في نوعه (قطعة أرض)، حيث يتم اختيار نوع العقار بناء على رقم

العقار المخزن في جدول قاعدة البيانات.

Lands _ land_type_id : Integer, Private •

هو عبارة عن ربط رقم العقار (الارض) مع نوع الارض، قد يكون نوع الارض زراعية او تجارية او صناعية .

Lands _ created: DataType, Private •

هو عبارة عن تاريخ انشاء هذا العقار(عقار قطعة الارض).

Lands _ modified: DataType, Private •

هو عبارة عن اخر تعديل تم على هذا العقار .

System Properties Operation (2)

addLand() : void ➤

يقوم مستخدم النظام (الزبون او مسؤول الموقع) بوضع معلومات العقار (قطعة الارض)، وهي نوع قطعة الارض، وتحديد العقار(قطعة الارض) هل هي للبيع ام الايجار، باضافة الى اختيار المدينة و وصف العقار ونوع العقار وسعره ومساحته من خلال الواجهات الخاصة باضافة هذه المعلومات، حيث تخزن هذه المعلومات في قاعدة البيانات.

editLand () : void ➤

يستطيع مستخدم النظام (الزبون أو مسؤول الموقع أو الزائر) الحصول على معلومات العقار، وهي نوع قطعة الارض، وتحديد العقار(قطعة الارض) هل هي للبيع ام الايجار، وايضا اختيار المدينة و وصف العقار ونوع العقار وسعره ومساحته من خلال الواجهات الخاصة لعرض هذه المعلومات، حيث يستطيع المستخدم التعديل على هذه المعلومات والضغط على ايقونة حفظ، وبذلك تحفظ البيانات الجديدة في قاعدة البيانات، وتظهر هذه البيانات امام مستخدم النظام.

deleteLand () : void ➤

يعمل على حذف العقار من نوع منزل، حيث يتم ازالة معلومات العقار المضاف سابقا من قاعدة البيانات.

Cities Class ➤

يساعد المستخدم على اختيار المدينة التي يبحث بها عن وجود عقار ضمن حدودها.

Cities
-id : int
-name : varchar
-created : DataType
-modified : DataType

❖ صنف المدن موضح كالتالي:

Cities Attributes (1)

• Cities _ID : Integer, Private

هو عبارة عن رقم المدينة، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر لتميز كل مدينة عن الأخرى، يساعد على تسهيل عمليات البحث.

• Cities _name : Varchar, Private

هو عبارة عن اسم المدينة، يطلب من الزبون ادخاله عند ملء المعلومات الخاصة بالعقار، تخزن هذه المعلومات في قاعدة البيانات، تستخدم لتسهيل عملية البحث عن العقار المناسب.

• Cities _created : DataType, Private

هو تاريخ انشاء هذه الفئة (Cities).

Cities _modified : DataType, Private •

هو تاريخ اخر تعديل تم على هذه الفئة (Cities).

Aqar_type Class

يساعد على اختيار نوع العقار (قطعة أرض او منزل)

Aqar_types
-id : int
-name : varchar
-created : DataType
-modified : DataType

❖صنف نوع العقار موضح كالتالي:

Aqar_type Attributes (1

Aqar_type _ID : Integer, Private •

هو عبارة عن رقم نوع العقار، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر

لتمييز نوع العقار (قطعة ارض ام منزل)، إضافة إلى تسهيل عملية البحث على بيانات العقار.

Aqar_type _name : Varchar, Private •

هو عبارة عن نوع العقار (قطعة أرض أو منزل)، تخزن هذه المعلومات في قاعدة البيانات،

تستخدم لتسهيل عملية البحث عن المناسب.

Aqar_type _created : DataType, Private

هو تاريخ انشاء هذه الفئة (نوع العقار).

Aqar_type _modified : DataType, Private •

هو اخر تعديل تم على هذه الفئة (نوع العقار).

Land_Type Class

يساعد على اختيار نوع الارض (أرض تجارية، أرض زراعية، أرض صناعية).

land_types
-id : int
-name : varchar
-created : DataType
-modified : DataType

❖ صنف نوع الارض موضح كالتالي:

land_types Attributes (1)

land_types_ID : Integer, Private •

هو عبارة عن رقم (نوع الأرض)، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن

يتكرر لتميز نوع قطعة الارض .

land_types_name : Varchar, Private •

هو عبارة عن نوع قطعة الأرض، يطلب من الزبون ادخاله عند ملئ المعلومات الخاصة بقطعة

الارض، قد تكون نوع الارض زراعية او تجارية او صناعية، تخزن هذه المعلومات في قاعدة البيانات،

تستخدم لتسهيل عملية البحث عن قطع الارض.

land_types_created : DataType, Private •

هو تاريخ انشاء هذه الفئة (نوع قطعة الارض).

land_types_modified : DataType, Private •

هو اخر تعديل تم على هذه الفئة (نوع قطعة الارض).

Groups Class

يساعد على تحديد نوع المستخدم وتحديد صلاحياته.

Groups
-id : int
-name : varchar
-created : DataType
-modified : DataType

❖صنف المجموعات موضح كالتالي:

Groups Attributes (1)

Groups _ID : Integer, Private •

هو عبارة عن رقم يمثل فئة المستخدم (مسؤول النظام، زبون)، يميز الزبون من مسؤول النظام.

Groups _name : Varchar, Private •

هو عبارة عن وصف اسم فئة من المستخدمين (مسؤول النظام، زبون)،

Groups _created : DataType, Private •

هو تاريخ انشاء هذه الفئة (Groups).

Groups _modified : DataType, Private •

هو اخر تعديل تم على هذه الفئة (نوع المستخدم).

Messages Class

هذه (فئة الرسائل) تعمل على تواصل الزبائن، حيث يستطيع الزبون ان يرسل رسائل الى صاحب

العقار او مسؤول الموقع ويستطيع الزبائن ايضا التواصل مع مسؤول الموقع .

Messages
-id : int
-sender_id : int
-reciver_id : int
-message_id : int
-title : varchar
-body : varchar
-is_read : tinyint
-is_deleted_by_sender : tinyint
-is_deleted_by_reciver : tinyint
-created : DateTime
-modified : DateTime
+send_message()
+Replay_message()

❖صنف الرسائل موضح كالتالي:

Message Attributes (1

Message_ID : Integer, Private •

هو عبارة عن رقم الرسالة، حيث أنه رقم تسلسلي وهو المفتاح الأساسي الذي لا يمكن أن يتكرر لتمييز الرسائل.

Message_sender_id: Integer, Private •

هو عبارة عن ربط رقم مرسل الرسالة مع الرسائل، يستخدم لتمييز الشخص الذي قام بإرسال الرسالة، يتم تخزين رقم مرسل الرسالة في جدول قاعدة البيانات.

Message_reciver_id : Integer, Private •

هو عبارة عن ربط رقم مستقبل الرسالة مع الرسائل، يستخدم لتمييز الشخص الذي قام باستقبال الرسالة، يتم تخزين رقم مستقبل الرسالة في جدول قاعدة البيانات.

Message _ message_id : Integer, Private •

يستخدم هذا الرقم لتمييز اذا كانت الرسالة هي رد على رسالة اخرى ، في حال كانت الرسالة عبارة عن رد، يكون هذا الرقم يعبر عن الرسالة التي تم الرد عليها.

Message _ title : Varchar, Private •

هو عبارة عن عنوان الرسالة، يقوم مستخدم النظام بادخاله عند كتابة الرسالة، يتم تخزين عنوان الرسالة في جدول قاعدة البيانات.

Message _ body : Varchar, Private •

هو عبارة عن جسم الرسالة، يقوم مستخدم النظام بادخاله عند كتابة نص الرسالة، حيث يتم الاستفسار عن العقار بشكل تفصيلي، يتم تخزين عنوان الرسالة في جدول قاعدة البيانات.

Message _ is_read : Tinyint, Private •

هو عبارة عن تحديد الرسالة هل تم قرائتها من قبل المستخدم ام لا.

Message _ is_deleted_by_sender : Tinyint, Private •

هو عبارة عن تحديد الرسالة هل تم حذفها من قبل المرسل الرسالة ام لا.

Message _ is_deleted_by_reciver : Tinyint, Private •

هو عبارة عن تحديد الرسالة هل تم حذفها من قبل المستقبل الرسالة ام لا.

Message _ created : DataType, Private •

هو عبارة عن تاريخ انشاء هذه الرسالة.

Message _ modified : DataType, Private •

هو عبارة عن اخر تعديل تم على هذه ارسالة.

System Properties Operation (2)

send_message () : void ➤

يقوم مستخدم النظام (الزبون او مسؤول الموقع) بكتابة الرسالة، حيث يدخل عنوان الرسالة (يجب ان يكون موحى لنص الرسالة، وان يكون قصير لا تفاصيل فيه)، ثم يقوم المستخدم بإدخال جسم الرسالة، هنا يستطيع المستخدم (الزبون او مسؤول الموقع) أن يفصل ويستفسر عن العقارات، ويستطيع أن يتعرف على صاحب العقار وأن يتواصل معه، يتم ارسال الرسالة من خلال الواجهات الخاصة بارسال الرسائل، حيث تخزن هذه الرسائل في قاعدة البيانات.

Replay_message () : void ➤

هو الرد مستخدم النظام (الزبون أو مسؤول الموقع) على الرسائل التي تم ارسالها من قبل مستخدمين آخرين في النظام (الزبون أو مسؤول الموقع).

Attachments Class ➤

هذه فئة المرفقات) تعمل على اضافة مرفقات للعقار الذي يتم الاعلان عنه.

Attachments
-id : int
-model : varchar
-foreign_key : int
-name : varchar
-attachment : varchar
-dir : varchar
-type : varchar
-size : int
-active : tinint

❖ **صنف المرفقات موضح كالتالي:**

: Attachment Attributes (1

Attachment _id : Integer, Private •

رقم تسلسلي يستخدم لتمييز المرفقات من بعضها البعض وربطها مع الجداول الاخرى.

Attachment _model : Varchar, Private •

اسم ال Class للجدول الذي يحوي العلاقة مع هاي المرفقات ، مثال اذا كان هذا المرفق عبارة

عن صورة لعقار معين فان ال class هو Property.

Attachment _foreign_key : Integer, Private •

يمثل الرابط الذي يربط المرفقات بالعقار والجدول الاخرى.

Attachment _attachment: Varchar, Private •

اسم الملف التابع للمرفقات.

Attachment _dir : Varchar, Private •

عبارة عن المكان المخزن به الصور.

Attachment _type : Varchar, Private •

نوع المرفقات (الصور).

Attachment _size : Integer, Private •

عبارة عن حجم المرفقات.

Attachment _active : Tinyint, Private •

يرجع هل المرفقات فعالة ام غير فعالة.

Attachment _created : DataType, Private •

هو تاريخ رفع المرفقات (صور العقار).

• Attachment _ modified : DataType, Private

هو عبارة عن اخر تعديل تم على هذه المرفقات.

5.6 Interface Design (System Messages)

5.6.1 Object Interfacing

في الشكل التالي توضيح لدور الشخص الذي يضيف عقار (منزل) وتفاعله مع النظام، وكيفية سير عمليات النظام. حيث تتكون الرسمة التالية من ثلاث كائنات يتفاعل معها ال شخص المتقدم لإضافة عقار (منزل) وهي :

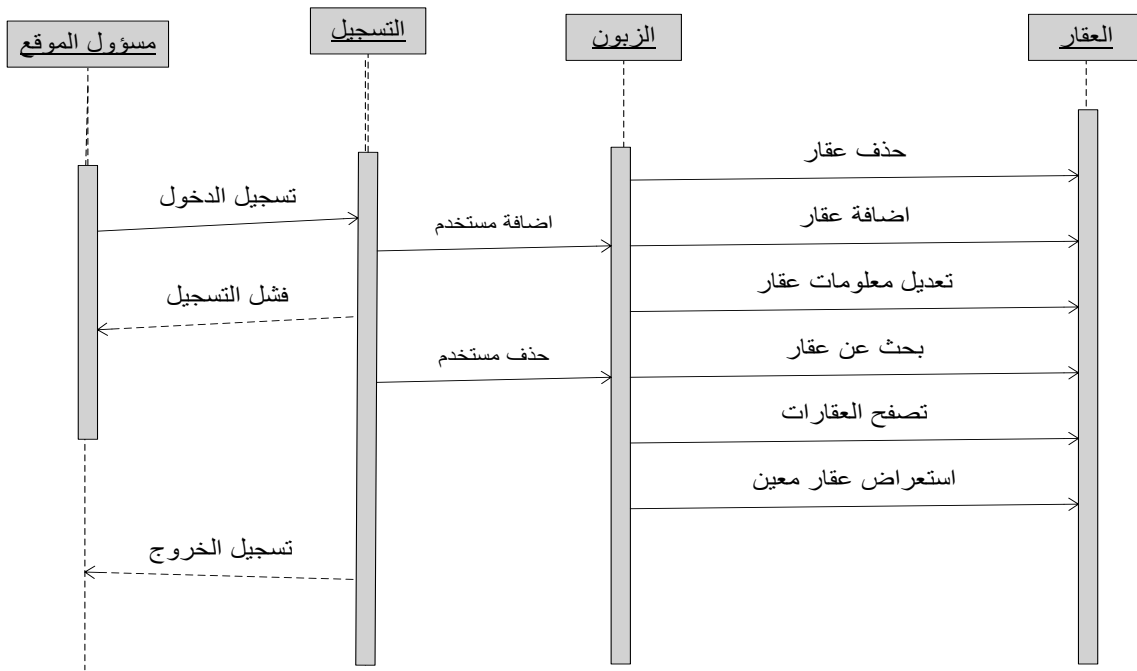
تفاعل مسؤول النظام مع النظام

في الشكل التالي توضيح لدور مسؤول النظام و تفاعله مع النظام ، وكيفية سير عمليات النظام. حيث تتكون الرسمة التالية من ثلاث كائنات يتفاعل معها مسؤول النظام:

التسجيل : وهو الجزء الاول في تفاعل مسؤول النظام مع النظام، بحيث يتم التحقق من اسم المستخدم وكلمة المرور.

عقار : في هذا الجزء يتم توضيح العمليات التي يقوم بها مسؤول الموقع وهي اضافة العقار (قطعة الارض، منزل) أو حذف عقار أو تعديل معلومات العقار او البحث عن عقار او تصفح عقار أو استعراض عقار معين.

الزبون : امكانية اضافة مستخدم الى النظام او حذفه من النظام.



شكل (1-6-5) تفاعل مسؤول الموقع مع النظام

تفاعل الزبون مع النظام ➤

في الشكل التالي توضيح لدور الشخص (الزبون) و تفاعله مع النظام ، وكيفية سير عمليات

النظام. حيث تتكون الرسمة التالية من كائنين يتفاعل معها الشخص الزبون :

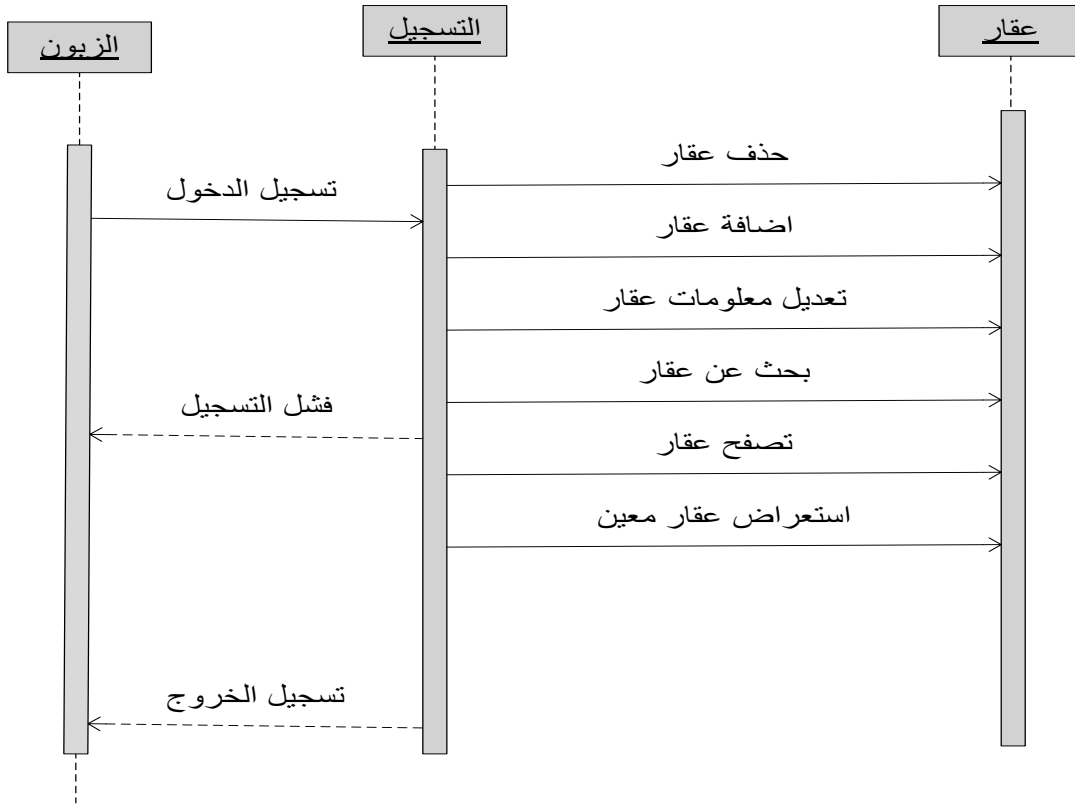
التسجيل : وهو الجزء الاول في تفاعل الزبون مع النظام، بحيث يتم التحقق من اسم المستخدم وكلمة

المرور.

عقار : في هذا الجزء يتم توضيح العمليات التي يقوم بها بها الزبون وهي اضافة العقار (قطعة الارض،

منزل) أو حذف عقار أو تعديل معلومات العقار او البحث عن عقار او تصفح عقار أو استعراض

عقار معين.



شكل (2-6-5) تفاعل الزبون مع النظام

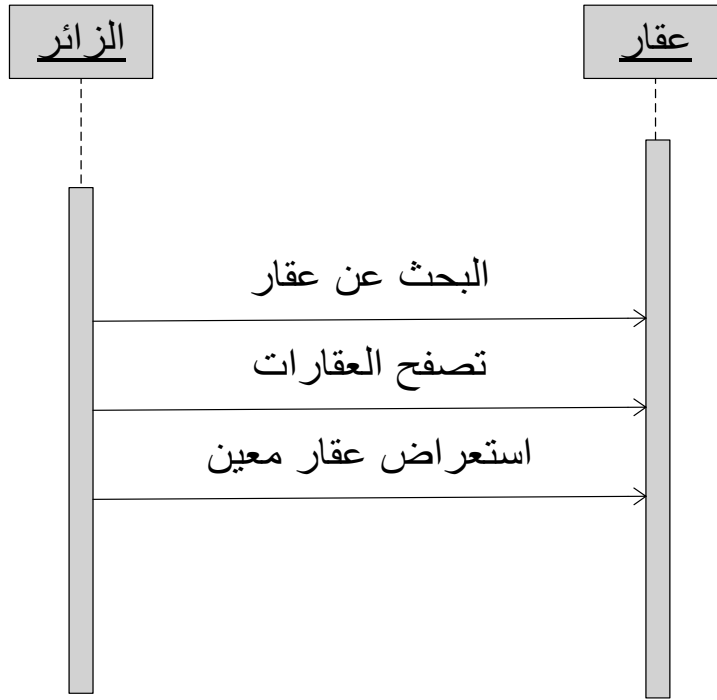
تفاعل الزائر مع النظام ➤

في الشكل التالي توضيح لدور الشخص (الزائر) و تفاعله مع النظام ، وكيفية سير عمليات

النظام. حيث تتكون الرسمة التالية من كائ العقار الذي يتفاعل معه الزائر :

عقار : في هذا الجزء يتم توضيح العمليات التي يقوم بها بها الزائر وهي البحث عن عقار، أو تصفح

عقار او استعراض عقار معين.



شكل (3-6-5) تفاعل الزائر مع النظام

5.6.2 تصميم صفحات المستخدم

سيتم في هذا الجزء وضع الواجهات الرسومية للبرنامج والصفحات التي يحتويها الموقع الإلكتروني التابع للنظام، والتي تعد الواجهات المباشرة لتفاعل المستخدم مع النظام، فيما يلي سيتم وضع شرح مختصر عن كل واجهه وطبيعة عمله.

5.6.2.1 واجهة انشاء حساب

يتم من خلال هذه الصفحة انشاء حساب عضو جديد على الموقع بحيث يقوم بتعبئة البيانات

المطلوبة كما هو مبين في الشكل التالي :

تسجيل

اضافة مستخدم

العنوان أدخل عنوانك	البريد الإلكتروني xample : jhon@example.com	الاسم الأول الاسم الأول
	كلمة السر كلمة السر	اسم العائلة اسم العائلة
	تأكيد كلمة المرور اعادة ادخال كلمة المرور	هاتف 05XXXXXXXX

أرسل

Register

Add User

First Name First Name	Email example : jhon@example.coi	Address Enter Your address
Last Name Last Name	Password Password	
Phone 05XXXXXXXX	Confirm Password re enter your password	

Submit

5.6.2.2 واجهة تسجيل الدخول

تحتوي هذه الواجهة على حقلين الاول يحتوي اسم المستخدم وهو عبارة عن البريد الالكتروني و الحقل

الثاني عبارة عن كلمة المرور الخاصة به:

The image shows two side-by-side screenshots of a login form. The left screenshot is in English, with a header containing a user icon, a dropdown menu with 'English', and a 'تسجيل الدخول' (Login) button. The form fields are labeled 'البريد الالكتروني' (Email) and 'كلمة السر' (Password). The email field contains 'example: jhon@example.com' and the password field contains 'أدخل كلمة المرور' (Enter your password). Below the fields is a blue 'تسجيل الدخول' button and a link 'نسيت كلمة المرور?' (Forgot my password?). The right screenshot is in Arabic, with a header containing 'Arabic' and a 'Login' dropdown menu. The form fields are labeled 'Email' and 'Password'. The email field contains 'example: jhon@example.com' and the password field contains 'Enter your password'. Below the fields is a blue 'Login' button and a link 'Forgot My password?'.

5.6.2.3 واجهة استعادة كلمة المرور

بحيث يتم في هذه الصفحة ادخال البريد الالكتروني من اجل ان يتم استبدال كلمة المرور للمستخدم على البريد الالكتروني المدخل :

The image shows a screenshot of a 'Forgot your password?' form. The title is 'Forgot your password?'. Below the title is a message: 'Please enter email you used to register in website name, an email will sent to you to reset your password.' There is a text input field with the placeholder 'Enter email'. Below the input field is a blue 'Submit' button.

5.6.2.4 واجهة البحث عن عقار

يتم في هذه الواجهة ادخال معلومات العقار المراد البحث عنه، حيث يطلب اعطاء معلومات عن العقار وهي: اولاً في اي مدينة يقع العقار وهل العقار للبيع ام للإستجار باضافة الى نوع العقار المراد البحث عنه هل هو منزل ام قطعة أرض وايضا يطلب تحديد موقع على الخريطة للبحث عن العقارات بالنسبة له، وبعد العقار عن هذا الموقع الذي تم تحديده على الخريطة (مثال : تحديد 50 كم، تعني أريد جميع العقارات التي تبعد عني 50 كم او أقل) وأخيراً تحديد أكبر وأقل (مساحة و السعر).

Search For Properties

City

Lease / Sell

Land/House

Search Radius From (34.90,34.67)

Minmum Price \$

Maximum Price \$

Minmum Area M

Maximum Area M

البحث عن العقارات

المدينة

بيع / إيجار

أرض/منزل

منطقة البحث

لا يوجد حد

أقل سعر
 \$

أكبر مساحة
 M

أكبر سعر
 \$

أقل مساحة
 M

5.6.2.5 واجهة اضافة عقار من نوع ارض

في هذه الصفحة يتم ادخال معلومات قطعة الارض المراد عرضها في الموقع، حيث يطلب

تحديد العقار هل هو للبيع ام الاستئجار و ادخال نوع العقار ومساحته وسعره والمدينة التي يقع بها

ويطلب من المستخدم كتابة وصف عام عن العقار، ورفاق المرفقات لهذا العقار(صور).


New Land

Lease / Sell: Land Type: Area: M

Price: \$ City:

Description:

Locate marker on your property:



Attachment: No file chosen

Attachment: No file chosen


Attachment: No file chosen

Attachment: No file chosen

أرض جديدة

المدينة: سعر: \$ مساحة: M نوع الأرض: بيع / إيجار:

ضع العلامة على موقع عقارك



مرفقات:

مرفقات:

مرفقات:

مرفقات:

5.6.2.6 واجهة اضافة عقار من نوع منزل

في هذه الصفحة يتم ادخال معلومات المنزل المراد عرضه في الموقع، في هذه الواجهة يطلب من المستخدم ادخال عدد الغرف وعدد المطابخ و رقم الطابق وعدد البلكونات(الشرفات) وعدد


الحمامات وايضا يطلب تحديد المدينة التي يقع بها العقار ويطلب ادخال السعر والمساحة للعقار
وتحديد هل هو للبيع ام الاستئجار، بالإضافة الى ادخال وصف عن العقار وايضا المرفقات.

New House

Bathrooms Count 1	Balconies Count 1	Floors Count 1	Rooms Count 1	Kitchen Count 1
Lease / Sell Lease / Sell	Area M	Price 0.00 \$	City	

Description

Locate marker on your property.




Attachment Choose File No file chosen	Attachment Choose File No file chosen	Attachment Choose File No file chosen	Attachment Choose File No file chosen
--	--	--	--

[Save](#)

منزل جديد

عدد المطابخ 1	عدد الغرف 1	عدد الطوابق 1	عدد الشرفات 1	عدد الحمامات 1
المدينة Hebron	سعر \$ 0.00	مساحة M	بيع / إيجار بيع / إيجار	الوصف

ضع العلامة على موقع عقارك



مرفقات No file chosen Choose File	مرفقات No file chosen Choose File	مرفقات No file chosen Choose File	مرفقات n.jpg_8263...1015_61583 Choose File
--------------------------------------	--------------------------------------	--------------------------------------	---

[حفظ](#)

5.6.2.7 واجهة عرض العقارات باستخدام الخريطة

في هذه الواجهة يتم عرض موقعك وعرض موقع العقارات ايضا.



في هذه الواجهة يظهر ثلاث ألوان الأحمر والأخضر والأزرق حيث يدل اللون الأحمر على الموقع الذي قمت باختياره عند عملية البحث، أما اللون الأخضر فيدل على قطع الأرض واللون الأزرق يدل على البيوت.



5.6.2.8 واجهة عرض العقارات باستخدام القائمة.

تظهر واجهة عرض العقار باستخدام القائمة، وتظهر بض التفاصيل عن العقار مثل سعر العقار وبعد العقار عن الموقعك او الموقع الذي تم اختياره من قبل المستخدم، وتظهر بعض المعلومات القليلة عن العقار، في حال تم الضغط على هذه المعلومات، سوف ينقلك النظام الى صفحة العقار حيث تظهر المعلومات المفصلة عن هذا العقار، تتميز واجهة العرض باستخدام القائمة انها ترتب العقارات حسب البعد، حيث يظهر العقار الاقرب ثم الابعد ثم الابعد وهكذا.

Page 1 of 2, showing 6 records out of 10 total, starting on record 1, ending on 6

 <p>0.00 KM</p> <p>منزل للاستئجار في مدينة دورا مساحة 250 متر</p> <p>\$250.00</p>	 <p>5.62 KM</p> <p>ارض زراعية تقع في الخليل</p> <p>\$1,000.00</p>	 <p>6.36 KM</p> <p>بيت جديد للإيجار</p> <p>\$1,000.00</p>
 <p>6.46 KM</p>	 <p>6.54 KM</p>	 <p>9.58 KM</p>

5.6.2.9 واجهة تعديل معلومات الزبون

يتم ادخال البيانات المراد تعديلها، ثم الضغط على ايقونة ارسال، حيث تخزن المعلومات المعدلة في قاعدة البيانات.

Edit user

Edit User

First Name	Email	Address
<input type="text" value="mayson"/>	<input type="text" value="✉️ maysn1991@hotmail.com"/>	<input type="text" value="Doura"/>
Last Name	Password	
<input type="text" value="khalil"/>	<input type="password" value="\$ Password"/>	
Phone	Confirm Password	
<input type="text" value="0598111111"/>	<input type="password" value="\$ re enter your password"/>	

5.6.2.10 واجهة الرسائل

تحتوي هذه الواجهة على اسم المرسل اليه، حيث يتم في هذه الواجهة ادخال عنوان

الرسالة ونص الرسالة ثم ارسال الرسالة.

Send Message



To: Aaa S

Title

Body

Close

Send Message

5.6.2.11 واجهة ادارة الرسائل

يظهر في هذه الواجهة الرسائل الموجهه من الاشخاص الى صاحب العقار، حيث يستطيع صاحب

العقار الرد على هذه الرسائل ويستطيع ايضا حذفها.

Messages

مرحبا
mayson khali 3 Jun, 2013

test
aqar aqar 3 Jun, 2013

test
aqar aqar 3 Jun, 2013

< 1 >

مرحبا



mayson mayson

09:31 AM

Enter your reply

Send

5.6.2.12 واجهة تحديد الموقع.

في هذه الواجهة يتم تحديد المكان المراد البحث عن العقارات بالنسبة له (قد يكون المكان موقعك الحالي

او اي مكان اخر ترغب به).

×

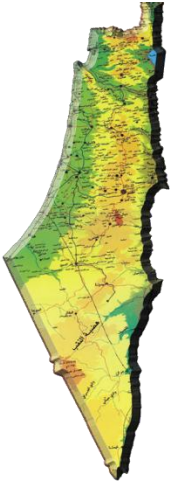
Locate Me

قمر صناعي خريطة

Save changes Close

الفصل السادس

تنفيذ النظام



في هذا الجزء سنقوم بوضع بعض الكود المهم الذي تم استخدامه في البرمجة:

اعداد البروجكت

تركيب الداتا بيس

```
#app/Config/database.php
public $default = array(
    'datasource' => 'Database/Mysql',
    'persistent' => false,
    'host' => 'localhost',
    'login' => 'root',
    'password' => '',
    'database' => 'aqar',
    'encoding' => 'utf8',
);
```

اعدادات الایمیل

```
#app/Config/email.php
public $gmail = array(
    'host' => 'ssl://smtp.gmail.com',
    'port' => 465,
    'timeout' => 30,
    'username' => 'palestinefinder@gmail.com',
    'password' => '1050926',
    'transport' => 'Smtplib',
    'from' => array('palestinefinder@gmail.com' =>
        'PropertyFinder' )
);
```

کود تسجيل مستخدم جديد

```
#app/Cotroller/UsersController.php
public function register(){

    if ($this->request->is('post')){
        $this->User->create();
        $this->request->data['User']['group_id'] = 4;
        if ($this->User->save( $this->request->data )){
            $this->Session->setFlash(__('The user has been
            saved'), 'flash_success');
            $this->redirect(array('controller' =>
                'pages', 'action' => 'home'));
        }
        else{
            # Create a loop with validation errors
            $this->Error->set( $this->User->validationErrors );
        }
    }
}
```

كود تركيب Auth Component من اجل عملية تسجيل الدخول وتسجيل الخروج

```
#app/Cotroller/AppController.php
public $components = array(
    'Auth' => array(
        'loginRedirect' => array('controller' =>
            'users', 'action' => 'home') ,
        'logoutRedirect' => array('controller' =>
            'pages', 'action' => 'home'),
        'authenticate' => array(
            'Form' => array( 'fields' =>
                array('username' => 'email')
            ),
            'authError' => 'You have to login.',
        )
    )
);

#app/Cotroller/UsersController.php
public function beforeFilter()
{
    parent::beforeFilter();
    $this->Auth-
    >allow('register', 'logout', 'change_password', 'remember_password', 'rem
    ember_password_step_2');
}

#app/Cotroller/UsersController.php
public function login(){
    if ($this->request->is('post')){
        if ($this->Auth->login()){
            $this->redirect($this->Auth->redirect());
        }
        else{
            $this->Session->setFlash(__('Invalid username or
            password, try again'),'flash_fail');
        }
    }
}

#app/Cotroller/UsersController.php
public function logout(){
    $this->redirect($this->Auth->logout());
}
}
```

```

#app/Cotroller/UsersController.php
public function remember_password()
{
    if( $this->request->is('post') )
    {
        $user = $this->User->findByEmail( $this->request-
        >data['User']['email'] );

        if( empty($user) )
        {
            $this->Session->setFlash('This email does not exist
            in our database.','flash_fail');
            $this->redirect(array('action' => 'login'));
        }

        $hash = $this->User->generateHashChangePassword();

        $data = array( 'User' => array( 'id' =>
        $user['User']['id'], 'hash_change_password' => $hash ) );
        $this->User->save($data);

        $link_url = Router::url(array( 'controller' => 'users',
        'action' => 'remember_password_step_2', $hash ), true);

        $email = new CakeEmail('gmail');
        $email->template('remember_password')
        ->emailFormat('html')
        ->subject(__('Lost password -
        '.Configure::read('Application.name')))
        ->to( $user['User']['email'] )
        ->viewVars(array('hash' => $link_url))
        ->send();

        //ALTER TABLE `users` ADD `hash_change_password` VARCHAR(
        255 ) NULL AFTER `password`

        $this->Session->setFlash('Check your e-mail to continue the
        process of recovering password.','flash_success');

    }
}

/**
 * Step 2 to change the password.
 * This step verifies that the hash is valid, if it is, show the form
 to the user to inform your new password
 */
public function remember_password_step_2( $hash = null )
{
    $user = $this->User->findByHashChangePassword( $hash );
}

```



```

if( !empty( $user ) && $user['User']['hash_change_password'] !=
    $hash || empty($user)){
    throw new NotFoundException(__('Invalid Link'));
}

# Sends the hash to the form to check before changing the
password
$this->set('hash',$hash);
$this->render('/Users/change_password');
}

```

كود اظهار معلومات المستخدم في الصفحة الشخصية

```

#app/Cotroller/UsersController.php
public function _view($id = null) {
    if (!$this->User->exists($id)) {
        throw new NotFoundException(__('Invalid user'));
    }

    $options = array('conditions' => array('User.' . $this-
    >User->primaryKey => $id));
    $this->set('user', $this->User->find('first', $options));
}

```

كود اضافة قطعة ارض وكود اضافة منزل

```

#app/Cotroller/PropertiesController.php
public function addLand() {

    if ($this->request->is('post')) {

        $this->request->data['Property']['property_type_id'] = 2;
        $this->_saveProperty();
    }

    $cities = $this->Property->City->find('list');
    $landTypes = $this->Property->Land->LandType->find('list');
    $this->set(compact( 'landTypes','cities' ));
}
}

```

```

#app/Cotroller/PropertiesController.php
public function addHouse() {

    if ($this->request->is('post')) {
        $this->request->data['Property']['property_type_id'] = 1;
        $this->_saveProperty();
    }
    $cities = $this->Property->City->find('list');
    $this->set(compact( 'cities' ));
}

```

```
}  
}
```

```
#app/Cotroller/PropertiesController.php
```

```
function _saveProperty(){  
  
    $dbs = $this->Property->getDataSource();  
    $dbs->begin( $this->Property );  
  
    $this->request->data['Property']['user_id'] =  
    AuthComponent::user('id');  
  
    $attachments = $this->request->data['Attachment'];  
    unset($this->request->data['Attachment']);  
  
    $this->Property->create();  
    if ($this->Property->saveAll($this->request->data)) {  
  
        $this->request->data['Attachment'] = $attachments;  
  
        if( $this->_saveImages( $this->Property->id ) ){  
            $dbs->commit( $this->Property );  
            $this->Session->setFlash(__('The property has been  
            saved'));  
            $this->redirect(array('action' => 'view', $this->  
            >Property->id ));  
        }  
        else {  
  
            $dbs->rollback( $this->Property );  
            $this->Session->setFlash(__('The property could not  
            be saved. Please, try again.'));  
        }  
    }  
    else {  
        $dbs->rollback( $this->Property );  
        $this->Session->setFlash(__('The property could not be  
        saved. Please, try again.'));  
    }  
}
```

كود تعديل معلومات المستخدم

```
#app/Cotroller/UsersController.php
```

```
public function edit($id = null)  
{  
    if( $id != null ){  
        $this->User->id = $id;  
        if (!$this->User->exists()){  
            throw new NotFoundException(__('Invalid user'));  
        }  
    }
```

```

    }
    else{
        $id = AuthComponent::user('id');
    }

    $this->User->recursive = -1;
    $user = $this->User->read(null, $id);
    $this->set('user', $user);

    if ($this->request->is('post') || $this->request->is('put'))
    {
        if( empty($this->request->data['User']['password']) )
        {
            unset($this->request->data['User']['password']);
        }

        if ($this->User->save($this->request->data)) {
            $this->Session->setFlash(__('The user has been
            saved'), 'flash_success');
            $this->redirect(array('action' => 'home'));
        }
        else{
            $this->Session->setFlash(__('The user could not be
            saved. Please, try again.'), 'flash_fail');
        }
    }
    else{
        $this->request->data = $user;
        unset($this->request->data['User']['password']);
        unset($this->request->data['User']['confirm_password']);
    }
}

```

كود تعديل معلومات العقار (منزل او قطعة ارض)

```

#app/Cotroller/PropertiesController.php
    public function editLand( $id = null ) {

        if (!$this->Property->exists($id)) {
            throw new NotFoundException(__('Invalid Land'));
        }

        $this->request->data['Property']['user_id'] =
        AuthComponent::user('id');
        $this->request->data['Property']['property_type_id'] = 2;

        if ($this->request->is('post') || $this->request->is('put')) {

            $this->_saveProperty();
        }
        else {

```

```

        $options = array('conditions' => array('Property.id' =>
            $id, "Property.user_id" =>
AuthComponent::user('id')));
        $this->request->data = $this->Property->find('first',
            $options);
    }
    $cities = $this->Property->City->find('list');
    $landTypes = $this->Property->Land->LandType->find('list');
    $this->set(compact( 'landTypes', 'cities' ));
}

```

```

#app/Cotroller/PropertiesController.php
public function editHouse( $id = null ) {

    if (!$this->Property->exists($id) ) {
        throw new NotFoundException(__('Invalid House'));
    }

    $this->request->data['Property']['user_id'] =
AuthComponent::user('id');
    $this->request->data['Property']['property_type_id'] = 1;

    if ($this->request->is('post') || $this->request->is('put')) {

        $options = array(
            'conditions' => array(
                "Property.id" =>$this->request-
>data['Property']['id'],
                "Property.user_id" =>
AuthComponent::user('id'),
                'House.id' => $this->request-
>data['House']['id'] ));
        $house = $this->Property->find( 'first', $options );

        if( !empty( $house ) ){
            $this->_saveProperty();
        }
        else{
            throw new NotFoundException(__('Invalid House'));
        }
    }
    else {
        $options = array('conditions' => array('Property.id' =>
            $id, "Property.user_id" => AuthComponent::user('id')));
        $this->request->data = $this->Property->find('first',
            $options);
    }
    $cities = $this->Property->City->find('list');
    $this->set(compact( 'cities' ));
}

```

كود تركيب Upload Plugin لرفع المرفقات

```
#app/Model/Attachment.php
public $actsAs = array(
    'Upload.Upload' => array(
        'attachment'=>array(
            'thumbnailSizes' => array(
                'slide_show' => '770x480',
                'explore' => '270x200',
                'xvga' => '1024x768',
                'vga' => '640x480',
                'thumb' => '80x80'
            ),
            'thumbnailMethod' => 'php'
        )
    ),
);
```

كود البحث عن عقارات

```
#app/Cotroller/PropertiesController.php
public function explore() {

    $this->Property->recursive = 1;

    $latitude = 30;
    $longitude = 30;

    $conditions = array();

    //for mile 3959

    //for km 3959
    $distance_eq = "( 6372.797 * acos( cos( radians( $latitude ) )
    * cos( radians( Property.latitude ) ) * cos( radians(
Property.longitude ) - radians( {$longitude} ) ) + sin(
radians( {$latitude} ) ) * sin( radians( Property.latitude ) )
) )";

    if( $this->request->is('post') ){

        $latitude = $this->request->data['User']['latitude'];
        $longitude = $this->request->data['User']['longitude'];

        $distance_eq = "( 6372.797 * acos( cos( radians(
$latitude ) ) * cos( radians( Property.latitude ) ) *
cos( radians( Property.longitude ) - radians(
{$longitude} ) ) + sin( radians( {$latitude} ) ) * sin(
radians( Property.latitude ) ) ) )";
```

```

        $conditions = array(
            'Property.price BETWEEN ? AND ?' => array( $this-
>request->data['Property']['minmum_price'], $this-
>request->data['Property']['maxmum_price'] ),
            'Property.area BETWEEN ? AND ?' => array( $this-
>request->data['Property']['minmum_area'], $this-
>request->data['Property']['maxmum_area'] ),
        );

        if( !empty( $this->request->data['Property']['city_id'] ) )
    ){
            $conditions['Property.city_id'] = $this->request-
>data['Property']['city_id'];
        }

        if( !empty( $this->request-
>data['Property']['property_type_id'] ) ){
            $conditions['Property.property_type_id'] = $this-
>request->data['Property']['property_type_id'];
        }

        if( !empty( $this->request-
>data['Property']['is_for_lease'] ) ){
            $conditions['Property.is_for_lease'] = $this-
>request->data['Property']['is_for_lease'];
        }

        if( !empty( $this->request-
>data['Property']['search_radius'] ) ){
            $conditions["Property.distance <="] = $this-
>request->data['Property']['search_radius'];
        }

        CakeSession::write( 'request_data' , $this->request->data
);

        CakeSession::write( 'filter_condition' , $conditions );
    }
    else{

        $this->request->data['User']['latitude'] = 30;
        $this->request->data['User']['longitude'] = 30;

    }

    CakeSession::write('lat',$latitude);
    CakeSession::write('lng',$longitude);

    if( CakeSession::check( 'filter_condition' , $conditions )){
        $this->request->data = CakeSession::read('request_data');
        $conditions = CakeSession::read( 'filter_condition');
    }

    $this->Property->virtualFields['distance'] = "$distance_eq";

```

```

$this->paginate = array(
    'fields' => array( "*" ),
    'limit' => 12,
    'conditions' => $conditions,
    //'group' => array( 'Property.id' ),
    'order' => array(
        "Property.distance" => "ASC",
    )
);

$properties = $this->paginate();
$this->log($properties);
$cities = $this->Property->City->find('list');

$propertyTypes = $this->Property->PropertyType->find('list');
$this->set(compact('properties', 'propertyTypes', 'cities'));

}

```

كود ارسال رسالة جديدة

```

#app/Cotroller/MessagesController.php
public function compose( $user_id = null ) {

    $this->layout = 'ajax';

    if( $user_id != null ){

        $this->loadModel('User');

        $this->User->recursive = -1;
        $user = $this->User->findById( $user_id );

        if( empty( $user ) ){
            throw new NotFoundException(__( 'Invalid User' ));
        }
        else{
            $this->set('user', $user );
        }
    }

    if ( $this->request->is('post') ) {

        $this->request->data['Message']['sender_id'] =
AuthComponent::user('id');
        $this->Message->create();

        if ( $this->Message->save( $this->request->data ) ) {
            $this->redirect( array( 'action' =>
'sendMessageSuccess' ) );

```

كود الرد على رسالة

```
#app/Cotroller/MessagesController.php
```

```
public function replyMessage() {  
  
    if ($this->request->is('post') ) {  
        $this->request->data['Message']['sender_id'] =  
AuthComponent::user('id');  
  
        $valid = true;  
  
        if( isset($this->request->data['Message']['message_id'])  
) {  
  
            $this->Message->recursive = -1;  
            $message = $this->Message->findById( $this->  
>request->data['Message']['message_id'] );  
  
            if( empty( $message ) ){  
                $valid = false;  
            }  
  
            if( $message['Message']['sender_id'] ==  
AuthComponent::user('id') ){  
                $this->request->data['Message']['reciver_id']  
= $message['Message']['reciver_id'];  
            }  
            else if( $message['Message']['reciver_id'] ==  
AuthComponent::user('id') ){  
                $this->request->data['Message']['reciver_id']  
= $message['Message']['sender_id'];  
            }  
            else {  
                $valid = false;  
            }  
  
        }  
        else if( !$this->Message->Sender->exist( $this->request->  
>data['Message']['reciver_id'] ) ){  
            $valid = false;  
        }  
  
        $this->Message->create();  
  
        if ($this->Message->save($this->request->data)) {  
  
            $this->Session->setFlash(__('The message has been  
saved'));  
  
        } else {
```



```

                $this->Session->setFlash(__('The message could not
be saved. Please, try again.));
            }
        }
        $this->redirect(array('action' => 'view', $this->request-
>data['Message']['message_id'] ));
    }
}

```

كود حفظ العقار

```

#app/Cotroller/PropertiesController.php
function _saveProperty() {

    $dbs = $this->Property->getDataSource();
    $dbs->begin( $this->Property );

    $this->request->data['Property']['user_id'] =
    AuthComponent::user('id');

    $attachments = $this->request->data['Attachment'];
    unset($this->request->data['Attachment']);

    $this->Property->create();
    if ($this->Property->saveAll($this->request->data)) {

        $this->request->data['Attachment'] = $attachments;

        if( $this->_saveImages( $this->Property->id ) ){
            $dbs->commit( $this->Property );
            $this->Session->setFlash(__('The property has been
            saved'));
            $this->redirect(array('action' => 'view', $this-
            >Property->id ));
        }
        else {

            $dbs->rollback( $this->Property );
            $this->Session->setFlash(__('The property could not
            be saved. Please, try again.));

        }
    }
    else {
        $dbs->rollback( $this->Property );
        $this->Session->setFlash(__('The property could not be
        saved. Please, try again.));
    }
}
}

```

```
#app/Cotroller/AppController.php

public function beforeFilter() {
    $this->_setLanguage();
}

private function _setLanguage() {
    //if the cookie was previously set, and Config.language has not
    been set
    //write the Config.language with the value from the Cookie

    if ($this->Cookie->read('lang') && !$this->Session-
>check('Config.language')) {
        $this->Session->write('Config.language', $this->Cookie-
>read('lang'));
    }
    //if the user clicked the language URL
    else if ( isset($this->params['language']) && ( $this-
>params['language'] != $this->Session->read('Config.language')) ) {
        //then update the value in Session and the one in Cookie
        $this->Session->write('Config.language', $this-
>params['language']);
        $this->Cookie->write('lang', $this->params['language'],
false, '20 days');
    }
    else{
        $this->Session-
>read('Config.language',Configure::read("Config.language"));
    }
}

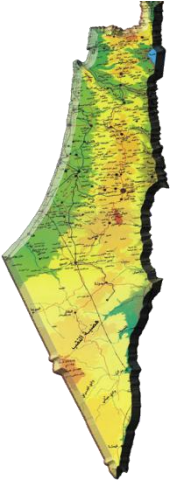
//override redirect

public function redirect( $url, $status = NULL, $exit = true ) {
    if (!isset($url['language']) && $this->Session-
>check('Config.language')) {
        $url['language'] = $this->Session->read('Config.language');
    }
    parent::redirect($url,$status,$exit);
}

#app/View/Helper/MyHtmlHelper.php
class MyHtmlHelper extends HtmlHelper {
    public function url($url = null, $full = false) {
        if(!isset($url['language']) && isset($this->params['language'])) {
            $url['language'] = $this->params['language'];
        }
        return parent::url($url, $full);
    }
}
}
```

الفصل السابع

اختبار النظام



❖ المقدمة

❖ فحص العمليات

7.1 المقدمة:

بعد مرحلة تصميم وتشغيل النظام، تأتي عملية فحص النظام والتي تعتبر من أهم المراحل التي يمر بها تطوير النظام، وذلك للتأكد من أنه يحقق المواصفات والمتطلبات ويعمل حسب ما هو متوقع .

7.2 فحص العمليات:

سيتم فحص عمليات النظام التي يقوم بها مستخدمي النظام، حيث سيتم فحص كل عملية بشكل منفصل عن الأخرى، كما هو مبين في الجداول التالية، والشاشات التي تبين التأكد من صحة العمليات التي يقوم بها كل مستخدم.

7.2.1 عمليات الزبون

فحص العمليات التي يقوم بها الزبون، وهي عملية تسجيل الدخول و اضافة العقارات والتعديل على بيانات العقارات والتعديل على بياناته الشخصية.

عمليات الزبون	
اسم العملية	تمام العملية
تسجيل الدخول للنظام	تمت بنجاح
اضافة عقار من نوع الارض	تمت بنجاح
اضافة عقار من نوع منزل	تمت بنجاح
تعديل معلوماته (ملفه الشخصي)	تمت بنجاح
ادارة عقاراته (تعديل، حذف)	تمت بنجاح

تمت بنجاح	البحث عن عقار
تمت بنجاح	تصفح الرسائل وادارتها
تمت بنجاح	ارسال الرسالة الى صاحب العقار
تمت بنجاح	الرد على الرسائل
تمت بنجاح	تسجيل الخروج من النظام

جدول (7.2.1.1) فحص عمليات الزبون

عملية تسجيل الدخول للنظام

هنا يتم التحقق من البريد الالكتروني المستخدم ومن كلمة المرور .

Error!

Invalid username or password, try again

Login

Email

Password


[Login](#)

[Forgot My password?](#)

عملية اضافة عقار من نوع الارض

يتم التأكد من البيانات المدخلة، والتأكد من ادخال بعض البيانات التي يجب ان تكون موجودة في معلومات العقار.

New Land

Lease / Sell Lease / Sell	Land Type Select Land Type	Area <input type="text"/> M	Price 0.00 \$	City <input type="text"/>
Description <input type="text"/>		<div>Please fill in this field.</div>	<div>Please fill in this field.</div>	
Attachment Choose File No file chosen	Attachment Choose File No file chosen	Attachment Choose File No file chosen	Attachment Choose File No file chosen	<input type="button" value="Save"/>

عملية اضافة عقار من نوع منزل

يتم التأكد من ان المستخدم قام بادخال البيانات في الحقول الخاصة بها، حيث تظهر هذه البيانات في معلومات العقار المضاف للموقع.

The property could not be saved. Please, try again.


New House

Bathrooms Count	Balconies Count	Floors Count	Rooms Count	Kitchen Count
1	1	1	1	1
Lease / Sell	Area	Price	City	
Lease / Sell	M	0.00 \$		

Please select Type

Description

Locate marker on your property.



Attachment

Choose File	No file chosen	Choose File	No file chosen	Choose File	No file chosen	Choose File	No file chosen
-------------	----------------	-------------	----------------	-------------	----------------	-------------	----------------

Save

تعديل معلوماته (ملفه الشخصي)

Edit user

Edit User











First Name	Email	Address
mayson	✉ maysonhotmail.com	this is me
Last Name	Password	
Khalil AAAAAAAAAAAAAAAAAA	\$ **	
max length allowed is 64	A password must be more than 6 characters	
Phone	Confirm Password	
0598111111	\$ **	
	password doesn't match confirmation.	

Submit

ادارة عقاراته (تعديل، حذف)

My Properties

Page 1 of 10, showing 5 records out of 48 total, starting on record 1, ending on 5

Id	Property Type	Description	Price	Area	Created	Actions
87	House	House for sell	500	500	22 Apr,2013	 
86	Land	land for sell			013	 
84	Land				013	 
80	House	this is me , here i am			013	 
80	House	this is me , here i am	350.668	301.943	15 Apr,2013	 

< 1 2 3 >

7.2.2 عمليات الزائر

فحص العمليات التي يقوم بها الزائر، وهي عملية البحث عن العقار وعملية تصفح واستعراض العقارات.

عمليات الزائر	
اسم العملية	تمام العملية
البحث عن عقار	تمت بنجاح
تصفح العقارات	تمت بنجاح
استعراض العقارات	تمت بنجاح
تسجيل (Register)	تمت بنجاح

جدول (7.2.2.1) فحص عمليات الزائر

البحث عن عقار

يتم ادخال البيانات المراد البحث عنها لتظهر النتيجة امامك على القائمة والخريطة

Search For Properties

City
Select City

Lease / Sell
Lease

Land/House
Land

Search Radius From (34.90,34.67)
100 KM

Minimum Price 450 \$ Maximum Price 500 \$

Minimum Area 450 M Maximum Area 500 M

Search

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

70.55 KM

this is land in Hebron Price for thi...

\$500.00

Search For Properties

City

Lease / Sell

Land/House

Search Radius From **(34.90,34.67)**

Minimum Price \$ \$

Maximum Price

Minimum Area M M

Maximum Area



Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

70.55 KM

this is new house Kitchen Count...

\$450.00

Search For Properties

City

Lease / Sell

Land/House

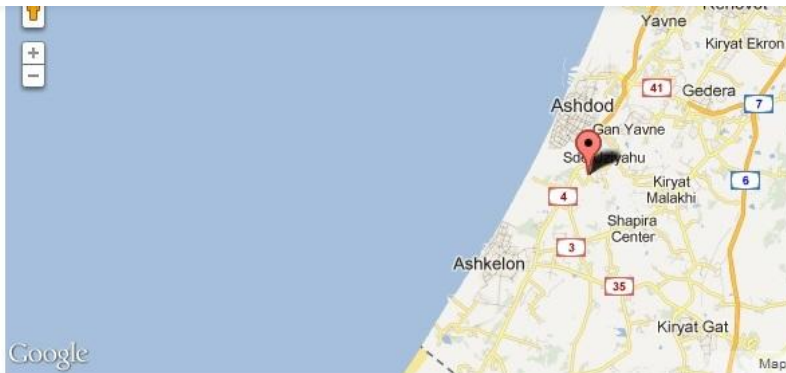
Search Radius From **(34.90,34.67)**

Minimum Price \$ \$

Maximum Price

Minimum Area M M

Maximum Area



Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

70.55 KM

this is land in Hebron Price for thi...

\$500.00

70.55 KM

this is new house Kitchen Count...


\$450.00

استعراض عقار


عرض مثال على استعراض معلومات عقار تم اضافته سابقا من قبل المستخدمين، حيث يظهر في هذه الواجهة صور العقار باضافة الى وصف العقار ويظهر ايضا اسم صاحب العقار، في حال تم الضغط على اسم صاحب العقار تظهر لك معلومات صاحب العقار، ويظهر في هذه الواجهة ايضا ايقونة الرسائل في حال تم الضغط عليها تفتح واجهة الرسائل لترسل رسالة لصاحب العقار، ويظهر ايضا موقع العقار على خرائط جوجل.

Since: 02/06/2013

For Lease 1000\$ 100M²



Location

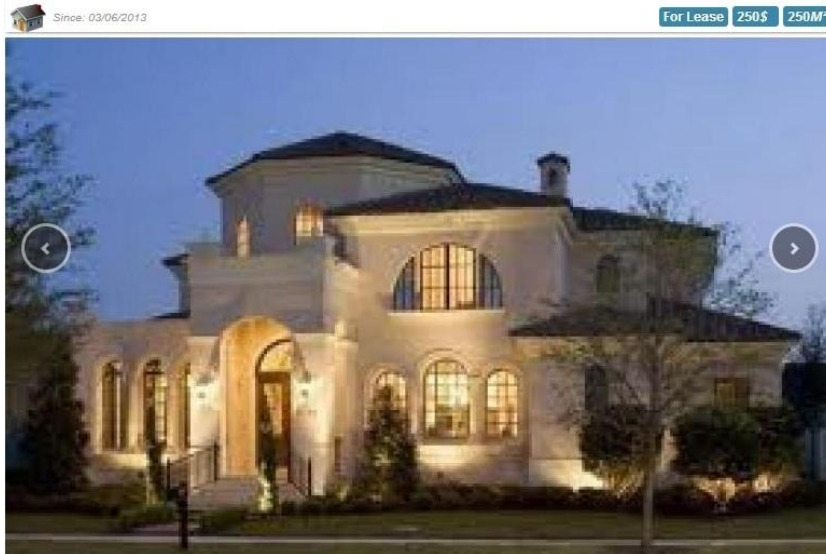


Owner

aqar aqar

Send aqar aqar Message

Description
Land in Hebron



Description

House in Doura

Features

- 1 Floors
- 5 Rooms
- 1 Kitchen
- 2 Balconies
- 2 Bathrooms
- 250 Meter

التسجيل

يتم هنا التأكد من ادخال معلومات المستخدم (الزائر) في المكان المخصص لها .

Add User

First Name

Last Name

Email example : jhon@example.co

Phone

Address

Password

 Confirm Password

Add User

First Name

Last Name

Email

Gender

Password

Please fill in this field.

Phone

Confirm Password

Address

Submit

المصادر والمراجع

<http://www.mysql.com/products/workbench/>

<http://twitter.github.io/bootstrap/>

<http://jquery.com/>

<https://developers.google.com/maps/documentation/javascript/>

<https://github.com/josegonzalez/upload>

<http://stackoverflow.com/questions/7915318/cakephp-2-0-how-can-i-send-email-from-a-gmail-account-using-cakemail-and-smtp-s>

www.shobiddak.com

<http://www.aqar-estate.com/misc/countryredirect?originalurl=http://www.aqar-estate.com/>

تم بحمد الله