



جامعة بوليتكنك فلسطين

كلية تكنولوجيا المعلومات وهندسة الحاسوب

دائرة علم الحاسوب وتكنولوجيا المعلومات

المشروع:

" نظام محوسب لأتمته خدمات الرعاية الصحية "

فريق العمل:

إبراهيم يوسف معبد

محمد باسم بدر

محمود نسيم سعيد

إشراف:

د. موسى إرفاعية

2016-2017

قدم هذا البحث استكمالاً لإنهاء متطلبات مشروع التخرج

الإهداء

بسم الله الرحمن الرحيم
(قل اعملوا فسيرى الله عملكم ورسوله والمؤمنون)
صدق الله العظيم

إلهي لا يطيب الليل إلا بشرك ولا يطيب النهار إلى بطاعتك .. ولا تطيب
حظات إلا بذكرك .. ولا تطيب الآخرة إلا بعفوك .. ولا تطيب الجنة إلا برويتك

"الله جل جلاله"

إلى من بلغ الرسالة وأدى الأمانة .. ونصح الأمة .. إلى نبي الرحمة ونور العالمين

"سيدنا محمد صلى الله عليه وسلم"

إلى من كلله الله بالهيبة والوقار .. إلى من علمني العطاء بدون انتظار .. إلى
من أحمل أسمه بكل افتخار .

والدي العزيز

إلى حكمتيوعلمي
إلى أدبيوحلمي
إلى طريقي المستقيم
إلى طريق..... الهداية
إلى ينبوع الصبر والتفؤل والأمل
إلى كل من في الوجود بعد الله ورسوله

أمي الغالية

إلى سندي وقوتي وملاذي بعد الله
إلى من آثروني على أنفسهم
إلى من علموني علم الحياة
إلى من أظهروا لي ما هو أجمل من الحياة

إخوتي

إلى من كانوا ملاذي وملجئي
إلى من تذوقت معهم أجمل اللحظات
إلى من سأفتقدهم وأتمنى أن يفتقدوني
إلى من جعلهم الله أخوتي بالله و من أحببتهم بالله

طلاب كلية تكنولوجيا المعلومات وهندسة الحاسوب

شكر و تقدير

نشكر الله العليّ القدير الذي أنعم علينا بنعمة العقل و الدين . القائل بمحكم تنزيل " وفوق كل ذي علم عليم " سورة وسف، آية 76.

و قال رسول الله (صلى الله عليه وسلم) " من صنع اليكم معروفًا فكافئوه ,فان لم تجدوا ما تكافئونه به فادعوا له حتى تروا انكم كافأتموه."

إلى صاحب التميّز والأفكار النيرة الدكتور موسى ارفاعية, أذكى التحيّات وأجملها وأنداها وأطيبها .. أرسلها لك بكلّ ودّ وحب وإخلاص

تعجز الحروف أن تكتب ما يحمل قلبي من تقدير واحترام .. وأن تصف ما اختلج بملئ فؤادي من ثناء وإعجاب .

فما أجمل أن يكون الإنسان شمعة تُنير دروب الحائرين , فشكراً لك على عطائك.

ولا ننسى ان نتقدم بجزيل الشكر الى جميع المعلمين و المعلمات الذين قاموا بتوجيهنا و دعمنا طيلة فترة الدراسة الجامعية.

كما و نشكر كل من مد لنا يد العون لانتظام و انجاح و اخراج هذا النظام على أكمل وجه.

فريق العمل

المخلص

المشروع عبارة عن تطبيق للهواتف الذكية المحمولة حيث يمكن من حفظ المعلومات الطبية بطريقة إلكترونية، فكرة المشروع تولدت من المشاكل الناجمة عن فقدان التقارير الطبية و المعلومات الطبية ، أيضا المشاكل مواعيد الطبيب والإنتظار لمدة طويلة .علاوة على ذلك ، يمكن التطبيق المرضى من إضافة أفراد العائلة و إضافة التقارير و المعلومات الطبية الخاصة بهم تحت خانة إضافة أفراد العائلة . إضافة الى ذلك هنالك صفحة ويب خاصة بالأطباء لتمكنهم من مراقبة المرضى الخاصة بهم . ولذلك، يعتبر التطبيق وسيلة سهلة لحفظ و تبادل و توزيع المعلومات الطبية الخاصة بك ، حيث يمكن ايجاد التطبيق و تحميله من خلال متجر الاندرويد .

وقد اتبع فريق المشروع منهجية تطوير دورة حياة النظام؛ من أجل تحقيق النواتج المتوقعة من النظام. و تبدأ من مرحلة تخطيط النظام، وتحليل المتطلبات، وتصميم النظام، وتطوير النظام وتشغيله، واختبار النظام. المرحلة الأخيرة من هذا النهج هي تشغيل وصيانة النظام.

وأخيرا، قام فريق المشروع ببناء تطبيق مصمم خصيصا للهواتف الذكية. والذي يقوم بمساعدة المرضى على حفظ المعلومات الخاصة بهم دون اي فقد ، ومع ذلك، فإنه يوفر القدرة على ارسال التقارير الطبية الى اطباء معينين ، و يعطي الحالة الصحية و بعض المعلومات عن الحالة الصحية الخاصة بكل مريض من خلال الضغط على ايقونة الحالة الصحية .

Abstract

Our project is an application that can be used for smart phones. Which aims save medical and health information in electronic way. This project's idea comes to solve the problems with loss reports and medical information, Also the problems than comes from doctor's appointments. Furthermore, the patients can add their families and their health information under family member tab. As well as there is web page for doctors to follow up patients Therefore, this application considered to be an easy way to save, share and distribute your medical information's that could be found in the Android store.

The project team followed the methodology of developing the system life cycle; In order to achieve the expected outputs of the system. It starts from the system planning phase, requirement analysis, system design, system developing and operating, and system testing. The last phase of this approach is running and maintaining the system.

Finally, the project team had built an application designed especially to smart phones. Which is able to give the users help patients to save their information's without losing it nevertheless, It provides the ability for share medical reports form specific doctors, and check how your healthy today by check your health status.

جدول المحتويات

2	الإهداء.....
3	شكر و تقدير.....
4	الملخص.....
5	Abstract.....
7	جدول الجداول.....
8	جدول الاشكال.....
9	الفصل الأول: الإطار العام لمشروع.....
9	الفكرة العامة للمشروع.....
9	مشكلة المشروع.....
9	أهداف مشروع.....
9	دوافع المشروع.....
10	توثيق الزيارة.....
10	أهمية مشروع.....
10	الفئة المستهدفة.....
10	الهيكل التنظيمي بالمشروع.....
11	قائمة المهام للمشروع.....
14	دراسة الجدوى الاقتصادية.....
16	الفصل الثاني متطلبات النظام.....
16	تحليل النظام.....
29	الوظائف والمتطلبات.....
30	تحليل البيانات.....
31	الفصل الثالث التصميم.....
31	أجزاء النظام.....
46	الفصل الرابع البناء و التنفيذ :.....
52	الفصل الخامس: فحص النظام.....
54	Unit testing.....

جدول الجداول

رقم الصفحة	العنوان	رقم الجدول
8	جدول الاشكال	(1.1)
12	(1.2)توضيح مهام المشروع	(1.2)
13	التقسيم الزمني لمهام المشروع	(1.3)
14	المصادر التطويرية الفيزيائية	(1.4)
14	المصادر التطويرية البرمجية	(1.5)
14	المصادر التطويرية البشرية	(1.6)
15	المصادر التشغيلية البرمجية	(1.7)
15	تكاليف المصادر الإجمالية	(1.8)
24-18	Use Case الخاصة بالمريض	-
27-24	Use Case الخاصة بالطبيب	-
28-27	Use Case الخاصة بالطبيب المسؤول	-
36-33	جداول قاعدة البيانات	-

رقم الصفحة	العنوان	رقم الشكل
11	طريقة العمل الخاصة بالمشروع (Agile Development).	(1.1)
17	Use case Diagram	(2.1)
30	Class diagram خاص بالتطبيق	(2.2)
31	الهيكلية الاساسية للنظام	(3.1)
32	قواعد بيانات النظام	(3.2)
42-37	شاشات التطبيق	-
45-43	شاشات الموقع الإلكتروني	-
48-46	التعليمات البرمجية الخاصة بالموقع	-
51-49	التعليمات البرمجية الخاصة بالتطبيق	-
52	فحص الموقع الإلكتروني	-
53	فحص تطبيق الاندرويد	-
54	لتعليمات البرمجية الخاصه بـ Units Testing	-

الفصل الأول: الإطار العام لمشروع

الفكرة العامة للمشروع

منذ بداية القرن الحادي والعشرون شهد العالم تطور كبير في شتى مجالات الحياة خصوصاً في المجال التكنولوجي مما أدى إلى تطور وسائل الاتصال والتواصل بين البشر وتوالى الاختراعات واحدة تلو الأخرى مما أدى إلى راحة الإنسان وتوفير الوقت وسرعة إنجاز العمل، ومن هذه الاختراعات تطور الهاتف من الشكل القديم التقليدي إلى الهاتف المحمول الذكي وتعددت البيئات المشغلة لهذه الأجهزة الذكية منها بيئة عمل أندرويد فقد تولدت فكرة لأعضاء الفريق بإنشاء تطبيق للهواتف الذكية الذي يندرج تحت فكرة ربط ووصل الطبيب مع المريض وإضافة التقارير والوصفات والتحليل والفحوصات الطبية لكل مريض. ومن ثم عرضها بشكل إلكتروني على الطبيب والجهات ذات الاختصاص، إضافة إلى ذلك؛ إضافة أفراد العائلة لكل مريض وإضافة خانة بطاقة التأمين الصحي الخاصة بكل مريض.

نحن نهدف كفريق، تطوير أنفسنا ودمج قدراتنا وخبراتنا العلمية والعملية داخل هذا المشروع كطلاب تكنولوجيا معلومات وعلم الحاسوب للخروج بفكرة تخدم المجتمع الفلسطيني بشكل خاص والمجتمعات الأخرى ككل وهذا جزء من مسؤوليتنا المجتمعية كأفراد مجتمع وضمن رؤية الجامعة واستراتيجيتها كجامعة تخدم مجتمع.

مشكلة المشروع

نظراً لكثرة الضغط على الأطباء والوسائل التقليدية المتبعة عند مراجعة المرضى للأطباء، مما أدى إلى ضياع العديد من التقارير الطبية والوصفات والتحليل الخاصة بكل مريض سواء من قبل الطبيب أو المريض. قام فريق المشروع بزيارة إلى أحد الأطباء واستمع من خلالها إلى المشاكل التي تواجه الأطباء واستمع أيضاً إلى المرضى واقترح عليهم فكرة المشروع، ومن حسن الحظ أن الفكرة لقيت دعماً بين الطبيب والمريض.

أهداف مشروع

1. بناء تطبيق android يساعد الأشخاص في حفظ المعلومات الطبية الخاصة بهم والخاصة بأفراد عائلاتهم.
2. بناء موقع إلكتروني يساعد الأطباء بمتابعة المرضى ومتابعة ملفاتهم الطبية وإجراء التعديلات والإضافات الطبية اللازمة لكل مريض.
3. تمكين الأطباء والجهات ذات الاختصاص بمعرفة جميع المعلومات والفحوصات والتقارير الطبية وكل ما يخص المرضى من المعلومات الصحية.
4. إضافة قائمة بأسماء جميع الأدوية ومعلومات حول كل دواء.
5. يقوم التطبيق بإعطاء نتيجة أولية للمريض بعد ادخال بعض المعلومات عن حالته الصحية.

دوافع المشروع

مساعدة المريض أو الأشخاص الطبيعيين بالمحافظة على جميع الفحوصات الطبية بجميع أنواعها والتي تتم عند الأطباء وحفظها بشكل إلكتروني.

1. المحافظة على بيانات الشخص والسماح للأطباء والجهات المتخصصة للاطلاع عليها.
2. تم استهداف جميع الأشخاص لما للمشروع دور كبير في مساعدتهم في حفظ بياناتهم الطبية والرجوع إليها متى يشاء ومعرفة ما قام به من فحوصات وتقارير طبية أخرى خاصة بالمريض وعائلته.

توثيق الزيارة

تم اجراء عدة مقابلات مع بعض الاطباء في مشافي وعيادات عدة وذلك لجمع المتطلبات الاساسية للاستكمال اهداف المشروع وتبعه لقاء مع عدد من المستخدمين المحتمل استخدامهم لهذه الخدمة لتطبيق, و ذلك استيفاءً للمتطلبات الخاصة بالمستخدم .

أهمية مشروع

1. أهمية المشروع بالنسبة للمرضى:
 - أن تطبيق سيساعدهم في الرجوع إلى أي فحص طبي قاموا به.
 - تسهيل معرفة معلومات الفحوصات الطبية الخاصة بعائلاتهم.
 - إضافة تأمين وإضافة معلومات عن التأمين.
 - مشاركة المعلومات الخاصة بهم وعائلاتهم للأطباء.
 - الاتصال على الإسعاف والطوارئ في الحالات الطارئة.
2. أهمية المشروع بالنسبة للأطباء:
 - معرفة جميع الفحوصات التي قام بها المراجعين لديهم.
 - إضافة تقارير جديدة للمرضى.
 - المحافظة على الوصفة الطبية المعطاة للمريض.

الفئة المستهدفة

يستهدف التطبيق المرضى من جميع الفئات العمرية المختلفة المستخدمة للأجهزة الذكية بالإضافة الى الجهات الطبية ذات العلاقة، بحيث يمكن للمريض تحميل السجلات الطبية الشخصية المختلفة مثل وصفات الأدوية، تقارير المختبرات، التطعيمات الطبية وغيرها من الصيغ المختلفة كمدخلات للتطبيق بحيث يقوم بالحفاظ عليها بشكل إلكتروني وإمكانية الوصول اليها أثناء زيارة الطبيب أو في حالة الطوارئ. ويقوم الطبيب باستعراض السجلات الخاصة بالمريض بشكل إلكتروني بطريقة سهلة وسريعة من اجل تشخيص الحالة ومن ثم كتابة الملاحظات من قبل الاطباء.

الهيكل التنظيمي بالمشروع

إن طريقة العمل التي قد تم اختيارها للقيام بالمشروع هي طريقة (Agile development) استنادا إلى أسباب عدة منها:

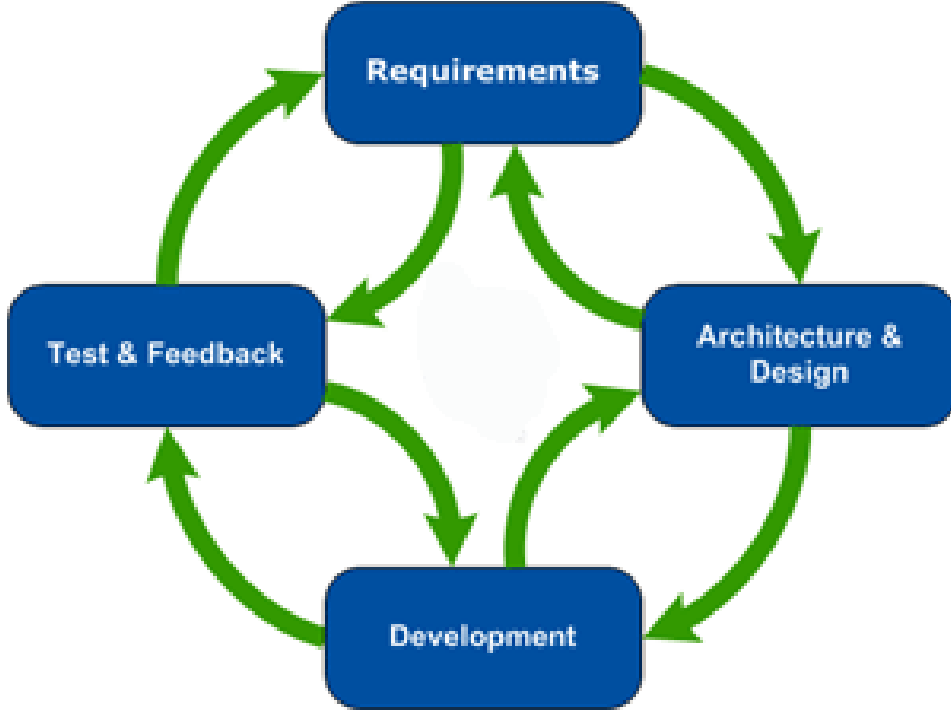
1. التقليل من خطر عدم موافقة الزبون على المشروع: طالما ان المستفيد من المشروع (stockholder) مشارك في المشروع وعلى علم بما يحدث طوال الوقت خلال العملية التطويرية فان هذه الطريقة (Agile) تقلل من احتمالية عدم رضى الزبون على المنتج النهائي.
2. امكانية الشروع في العمل قبل تحديد المتطلبات: في بعض الاحيان الزبون لا يعرف ما الذي يريده بالضبط من البرنامج، ولكن عن طريق اعتماد (Agile development) ليس من الضروري الانتظار للبدء بالعمل الى حين جمع كل الافكار والمتطلبات، حيث ان الزبون مشارك باستمرار في العملية التطويرية للمشروع وبأماكنه اعطاء وتنفيذ الافكار عند اي نقطة.
3. توافر جميع الموارد البشرية والمادية لهذا المشروع: بما ان الزبون يشارك في عملية التطوير فان امكانية حدوث سوء تواصل او عدم التوافقية تكون قليلة نسبياً، وبالتالي حدوث اخطاء اقل وهذا بدوره يؤدي الى اختصار المال والجهد.
4. أسهل لإدارة المخاطر، حيث يتم تحديد المخاطر والتعامل معها أثناء حدوثها وبالتالي القدرة على التغيير والتكيف السريعان.

قائمة المهام للمشروع

يجب على الفريق العمل على إنجاز هذا المشروع وفقاً للمهام التالية:

1. متطلبات المشروع
2. تصميم المشروع
3. تطوير المشروع
4. فحص المشروع

وشكل التالي يمثل طريقة عمل ال (Agile Development) وكيفية عملها وانتقالها من جزء إلى آخر:



الشكل (1.1) يوضح طريقة العمل الخاصة بالمشروع (Agile Development).

فيما يلي جدول بقائمة المهام للمشروع، والوقت الخاص في كل مهمة:
في هذا الجدول سيتم توضيح المهام التي ترافق حياة عمل المشروع وسيتم تحديد فترة زمنية معينة لإنجاز كل مهمة:

Dependencies	Duration (Days)	Task	No index entries found.
	10	إيجاد فكرة المشروع	T1
T1	2	البحث على الإنترنت	T2
T1	7	جمع المعلومات	T3
T1	4	بحث في مشاريع سابقة	T4
T1	14	مقابلة الأطباء والمستشفيات	T5
T2,T3,T4,T5	4	تحديد نطاق المشروع وملخص المشروع	T6
T1,T2,T3,T4,T5	15	التخطيط للمشروع	T7
T7	7	إيجاد مصادر المشروع وتحديد التكلفة وتحديد المخاطر وتحليلها	T8
T7,T8	30	تحليل النظام	T9
T10	20	تصميم النظام	T10
T11	7	التحسينات و التحقق من الصحة	T11
T11,T12	60	تنفيذ المشروع (الموقع الإلكتروني + التطبيق)	T12
T13	10	اختبار وتصحيح النظام	T13
T14	150	توثيق المشروع	T14

الجدول (1.2) يوضح المهام المشروع

فيما يلي جدول (Bar chart) يبين التقسيم الزمني لمهام المشروع مقسم على فترتين:

الفصل الثاني							الفصل الأول							المهمة
14	12	10	8	6	4	2	14	12	10	8	6	4	2	
■	■	■	■	■	■	■			■	■	■	■	■	التواصل مع المستخدمين
											■	■		التخطيط
										■	■			التحليل
									■	■	■			التصميم
	■	■	■	■	■	■								تطوير النظام
	■	■	■	■	■									فحص وصيانة المشروع
■	■	■	■	■	■	■	■	■	■	■	■	■	■	التوثيق

الجدول (1.3) يوضح التقسيم الزمني لمهام المشروع

الفترة التي سوف يتم العمل ضمنها
فترة استراحة / العطلة بين الفصلين

دراسة الجدوى الاقتصادية

سيتم توضيح تكلفة النظام الإلكتروني في مرحلتي التطوير والتشغيل، وتشمل تكلفة المصادر الفيزيائية والبشرية.

• المصادر التطويرية للنظام:

• المصادر التطويرية الفيزيائية

التكلفة الكلية	العدد	المصادر الفيزيائية
2800\$	3	جهاز حاسوب
350\$	1	هاتف محمول ذكي
2050\$		المجموع

الجدول (1.4)

• المصادر التطويرية البرمجية

التكلفة الكلية	العدد	المصادر برمجية
0\$	3	WAMP, MAMP Servers
0\$	3	Android studio
0\$	3	Sublime Text
0\$		المجموع

الجدول (1.5)

• المصادر التطويرية البشرية

التكلفة الشهرية	العدد	المصادر البشرية
2200\$	3	مطور
0\$	0	مدير
2200\$		المجموع

الجدول (1.6)

• مصادر التشغيلية للنظام:

• المصادر التشغيلية البرمجية:

التكلفة السنوية	العدد	المصادر برمجية
0\$	3	WAMP,MAMP Servers
11\$	1	Server and Domain
25\$	1	Google play developer Account
0\$	3	MySQL
75\$	المجموع	

الجدول (1.7)

• تكاليف المصادر الإجمالية:

التكاليف	التكلفة الكلية
التطويرية	4200\$
التشغيلية	75\$
المجموع	4275\$

الجدول (1.8)

الفصل الثاني متطلبات النظام

تحليل النظام

في هذه المرحلة سيتم شرح دورة حياة المشروع من إدارة وتطوير، بحيث يتم توضيح كل المراحل والعمليات التي يتضمنها النظام، والتعرف على المستخدمين للمشروع وكيفية استخدامهم للنظام بحيث يتم تحليل النظام بشكل عام كالتالي:

1 - المريض:

- يقوم بتسجيل الدخول الى التطبيق.
- إضافة سجل طبي¹ له ولعائلته.
- عرض وتعديل وحذف السجلات الطبية الخاصة به وعائلته.
- ارسال السجلات الطبية الى جهات الاختصاص.
- معرفة بعض المعلومات عن حالته الصحية من خلال التطبيق.
- تصفح الأدوية ومعرفة النشرة الطبية عن كل دواء.
- تسجيل الخروج من التطبيق.
- اجراء مكالمة طوارئ.

2 - الطبيب:

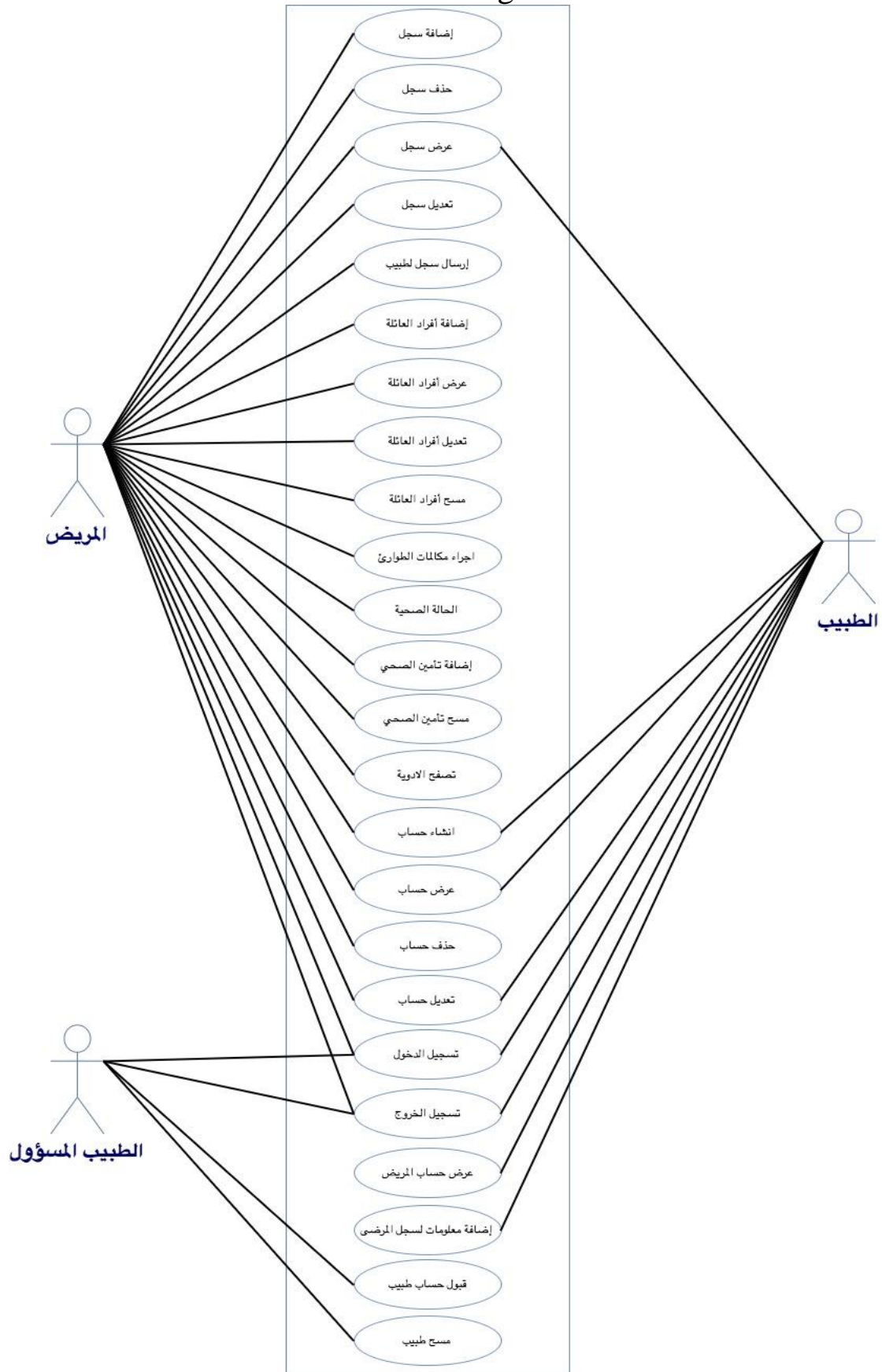
- تسجيل الدخول الى النظام.
- تصفح جميع المرضى المدرجين له.
- عرض جميع السجلات التي تم إرسالها له من قبل المرضى.
- إضافة سجل جديد لمريض معين والتعديل على سجلات المرضى.
- تسجيل الخروج من النظام.

3 الطبيب المسؤول:

- تسجيل الدخول الى النظام.
- تأكيد حسابات الأطباء , من خلال المرجع الاساسي الا وهو وزارة الصحة.
- تسجيل الخروج من النظام.
- بما ان المسؤول هو طبيب معترف به من قبل وزارة الصحة فيمكنه اضافة الأدوية و النشرة الطبية لكل دواء.

¹السجل الطبي: هو عبارة عن مجموعة الوثائق الطبية التي تم انشاءها لتوثيق الحالة الطبية للشخص, تم تسجيلها من قبل أناس مؤهلين للقيام بذلك. ويتضمن سجل المعلومات تفاصيل تعريف وتحديد المريض والمعالج , ومعلومات طبية حول العلاج الطبي الذي حصل عليه المريض, تاريخ المريض الطبي كما هو مبين في السجل, وتشخيص الوضع الحالي وتعليمات العلاج.

Use case Diagram



الشكل (2.1) يوضح الوظائف الخاصة بالنظام

الوظائف المتعلقة بالمريض:

الممثل الرئيسي	الهدف	الشروط السابقة	الدوافع	السيناريو	الاستثناءات
المريض	إنشاء حساب للمريض.	أن يقوم المريض بتعبئة مجموعة من البيانات لإنشاء حساب له.	إنشاء حساب للمريض لكي يستطيع استخدام التطبيق.	1. يقوم المريض بإضافة مجموعة من البيانات الخاصة به. 2. يقوم بالنقر على تسجيل.	عدم تعبئة بعض البيانات الخاصة به.

الممثل الرئيسي	الهدف	الشروط السابقة	الدوافع	السيناريو	الاستثناءات
المريض.	تعديل الحساب الخاص بالمريض.	أن يكون مسجلاً دخوله الى التطبيق.	امكانية تعديل معلومات الحساب الخاص بالمريض.	1. يقوم المستخدم بالدخول للتطبيق. 2. الدخول الى القسم الخاص بمعلومات الحساب. 3. تعديل المعلومات.	عدم انشاء المريض حساب خاص به.

الممثل الرئيسي	الهدف	الشروط السابقة	الدوافع	السيناريو	الاستثناءات
المريض.	عرض الحساب الخاص بالمريض.	إن يكون مسجلاً دخوله الى التطبيق.	عرض معلومات الحساب الخاص بالمريض.	1. يقوم المريض بتسجيل الدخول الى التطبيق. 2. يقوم المريض بعرض بياناته بالدخول الى الصفحة الخاصة به.	عدم تسجيل دخول المستخدم.

الممثل الرئيسي	الهدف	الشروط السابقة	الدوافع	السيناريو	الاستثناءات
المريض.	حذف الحساب الخاص بالمريض.	إن يكون مسجلاً دخوله الى التطبيق.	حذف معلومات الحساب الخاص بالمريض.	1. يقوم المريض بتسجيل الدخول الى التطبيق. 2. يقوم المريض بحذف حسابه بالدخول الى الصفحة الخاصة به.	عدم تسجيل دخول المستخدم.

Use case	تسجيل دخول النظام
الممثل الرئيسي	المريض.
الهدف	لكي يستخدم التطبيق.
الشروط السابقة	أن يكون قد قام بإنشاء حساب.
الدوافع	- ليتمكن المستخدم من إضافة المعلومات الطبية والتقارير وارسالها للأطباء.
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بفتح التطبيق. 2. إدخال اسم المستخدم الخاصة به. 3. إدخال كلمة المرور الخاصة به. 4. الضغط على زر تسجيل الدخول. 5. التأكد من البيانات المدخلة. 6. إرجاع المستخدم الى صفحة تسجيل الدخول إذا وجد أي خطأ في البيانات المدخلة. 7. الدخول الى تطبيق إذا كانت كلمة المرور واسم المستخدم صحيحين.
الاستثناءات	عدم انشاء المريض حساب خاص به.

Use case	إضافة سجل
الممثل الرئيسي	المريض.
الهدف	إضافة سجل جديد.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	أن يقوم بإضافة بياناته الطبية على التطبيق وإرسالها للأطباء.
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى صفحة إضافة سجل جديد. 3. يقوم باختيار نوع السجل المراد إضافته. 4. يقوم بتعبئة البيانات المطلوبة. 5. ثم يقوم باختيار الطبيب. 6. يقوم بالضغط على زر إضافة السجل.
الاستثناءات	عدم إدخال بعض البيانات الضرورية في صفحة إضافة السجل.

Use case	عرض سجل
الممثل الرئيسي	المريض.
الهدف	عرض سجل طبي.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	أن يقوم بعرض معلومات السجل الخاص المريض..
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى صفحة عرض سجل طبي. 3. يقوم باختيار نوع السجل المراد عرضه.
الاستثناءات	عدم تسجيل دخول المستخدم..

تعديل سجل	Use case
المريض.	الممثل الرئيسي
إمكانية التعديل على السجلات.	الهدف
- أن يكون مسجلاً دخوله الى التطبيق - أن يكون على الأقل لدية سجل قد تم أضافته من قبل.	الشروط السابقة
التعديل على السجل بعد حفظها من قبل المستخدم.	الدوافع
1. يقوم المستخدم بالدخول للتطبيق. 2. الذهاب الى صفحة السجلات. 3. اختيار السجل المطلوب من خلال الضغط عليه. 4. الذهاب الى صفحة عرض جميع البيانات الخاصة بالسجل يقوم بتعديل بيانات السجل. 5. في النهاية الضغط على زر تم والعودة الى صفحة السجلات.	السيناريو
- عدم وجود أي سجل طبي للمستخدم. - تغيير على البيانات وعدم حفظها.	الاستثناءات

حذف سجل	Use case
المريض.	الممثل الرئيسي
حذف سجل خاص بالمستخدم.	الهدف
- أن يكون مسجلاً دخوله الى التطبيق. - أن يكون على الأقل لدية سجل قد تم أضافته من قبل.	الشروط السابقة
حذف سجل للمستخدم لا يريده او يريد الانتهاء منه.	الدوافع
1. يقوم المستخدم بالدخول للتطبيق. 2. الذهاب الى جميع السجلات. 3. الدخول الى السجل المطلوب والضغط على زر حذف السجل. 4. تأكيد عملية الحذف.	السيناريو
- عدم وجود أي سجل طبي للمستخدم. - أن يتم حذف سجل غير المراد.	الاستثناءات

ارسال سجل للطبيب	Use case
المريض.	الممثل الرئيسي
مشاركة سجل طبي خاص به مع طبيب.	الهدف
إن يكون مسجلاً دخوله الى التطبيق.	الشروط السابقة
اخذ استشارة طبية عن بعض التقارير الطبية.	الدوافع
1. يقوم المريض بتسجيل الدخول الى التطبيق. 2. يقوم المريض بالدخول الى صفحة السجلات الطبي . 3. يقوم بالضغط على السجل المراد ارسالة سواء له او لاحد من افراد عائلته. 4. يقوم بتحديد الطبيب من قائمة الاطباء المعتمدين 5. يفوم بالضغط على رز ارسال.	السيناريو
- عدم تسجيل دخول المستخدم. - ارسال سجل خاطيء.	الاستثناءات

Use case	أضافة أفراد العائلة
الممثل الرئيسي	المريض.
الهدف	إضافة افراد العائلة لمستخدم التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	إضافة افراد العائلة لكي يستطيع إضافة معلوماتهم الطبية.
السيناريو	1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى إضافة افراد العائلة 3. يقوم بإدخال البيانات الخاصة بأفراد عائلة المستخدم. 4. ثم الضغط على زر تم.
الاستثناءات	عدم حفظ البيانات التي تم اضافتها لأفراد العائلة.

Use case	عرض أفراد العائلة
الممثل الرئيسي	المريض.
الهدف	عرض افراد العائلة لمستخدم التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	إضافة افراد العائلة لكي يستطيع إضافة معلوماتهم الطبية.
السيناريو	1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى عرض افراد العائلة 3. يقوم بالضغط على عرض افراد العائلة. 4. ثم الضغط على زر تم.
الاستثناءات	عدم تسجيل دخول المستخدم..

Use case	تعديل أفراد العائلة
الممثل الرئيسي	المريض.
الهدف	امكانية التعديل افراد العائلة لمستخدم التطبيق.
الشروط السابقة	- أن يكون مسجلاً دخوله الى التطبيق - أن يكون على الأقل لديه سجل قد تم أضافته من قبل
الدوافع	تعديل افراد العائلة بعد حفظها من قبل المستخدم.
السيناريو	1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى صفحة عرض جميع البيانات الخاصة بأفراد يقوم بتعديل بيانات احد افراد العائلة. 3. في النهاية الضغط على زر تم والعودة الى صفحة السجلات.
الاستثناءات	- عدم وجود أي احد من افراد العائلة للمستخدم. - تغيير على البيانات وعدم حفظها

Use case	مسح أفراد العائلة
الممثل الرئيسي	المريض.
الهدف	مسح افراد العائلة لمستخدم التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	مسح فرد من افراد العائلة بسبب إنشاء حساب خاص به .
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى افراد العائلة. 3. يقوم بالتحديد احد افراد العائلة لكي يتم مسحه. 4. يقوم بالضغط على مسح افراد العائلة. 5. ثم الضغط على زر تم.
الاستثناءات	عدم تسجيل دخول المستخدم..

Use case	أضافة تأمين الصحي
الممثل الرئيسي	المريض.
الهدف	إضافة بطاقة التأمين الصحي لمستخدم التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	إضافة بطاقة التأمين الصحي و حفظها لكي يتمكن من عرضها عند استشارة الطبيب .
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى إضافة بطاقة التأمين الصحي 3. يقوم بإدخال البيانات الخاصة ببطاقة التأمين 4. ثم الضغط على زر تم.
الاستثناءات	عدم حفظ البيانات التي تم اضافتها لبطاقة التأمين بشكل صحيح.

Use case	مسح بطاقة تأمين الصحي
الممثل الرئيسي	المريض.
الهدف	مسح بطاقة التأمين الصحي لمستخدم التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	مسح بطاقة التأمين الصحي بسبب انتهاء او تجديدها.
السيناريو	<ol style="list-style-type: none"> 1. يقوم المستخدم بالدخول الى التطبيق. 2. الذهاب الى مسح بطاقة التأمين الصحي 3. يقوم بالضغط على زر مسح البيانات الخاصة ببطاقة التأمين 4. ثم الضغط على زر تم.
الاستثناءات	عدم تسجيل دخول المستخدم.

Use case	إجراء مكاملة طوارئ
الممثل الرئيسي	المريض.
الهدف	إمكانية الاتصال بالطوارئ.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	إمكانية الاتصال بالطوارئ من قبل المستخدم.
السيناريو	1. يقوم المستخدم بتسجيل الدخول للتطبيق. 2. يقوم المستخدم بالنقر على ايقونة الاتصال بالطوارئ. 3. يتم الاتصال على رقم الطوارئ.
الاستثناءات	عدم تسجيل دخول المريض.

Use case	توقع الحالة الصحية
الممثل الرئيسي	المريض.
الهدف	إعطاء نتائج توقعية عن الحالة الصحية للمريض ، بناءً على ادخال بعض المعلومات الطبية من قبل المستخدم.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	إعطاء بعض المعلومات عن الحالة الصحية الخاصة بالمريض.
السيناريو	1. يقوم المستخدم بتسجيل الدخول للتطبيق 2. يقوم المستخدم بالنقر على ايقونة الحالة صحية. 3. يقوم المستخدم بملء المعلومات الخاصة بحالته الصحية. 4. يقوم المستخدم بالنقر على زر اظهار النتيجة.
الاستثناءات	إدخال بيانات خاطئة.

Use case	تصفح الأدوية
الممثل الرئيسي	المريض.
الهدف	إمكانية الاطلاع على الادوية من خلال التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	- التعرف والاطلاع على الادوية والعقاقير الطبية من قبل المستخدم. - يحتاج المستخدم الى اتصال بالإنترنت.
السيناريو	1. يقوم المستخدم بتسجيل الدخول للتطبيق 2. يقوم المستخدم بالنقر على تصفح الادوية. 3. يقوم المستخدم بالبحث عن الدواء المطلوب. 4. نقر على الدواء المراد. 5. ثم الانتقال الى صفحة الدواء وعرض معلومات عن دواء.
الاستثناءات	عدم وجود الدواء المراد.

Use case	تسجيل الخروج من النظام
الممثل الرئيسي	المريض.
الهدف	السماح للمستخدم الخروج من التطبيق.
الشروط السابقة	أن يكون مسجلاً دخوله الى التطبيق.
الدوافع	السماح بالخروج من التطبيق بعد إجراء المطلوب.
السيناريو	1. أن يضغط المستخدم على زر تسجيل الخروج. 2. خروج المستخدم من التطبيق.
الاستثناءات	عدم تسجيل دخول المستخدم.

الوظائف المتعلقة بالطبيب :

Use case	إنشاء حساب
الممثل الرئيسي	الطبيب
الهدف	إنشاء حساب للطبيب.
الشروط السابقة	يحتاج الطبيب أن يقوم بتعبئة مجموعة من البيانات لإنشاء حساب له.
الدوافع	إنشاء حساب للطبيب لكي يستطيع متابعة بيانات المرضى.
السيناريو	1. يقوم الطبيب بإضافة مجموعة من البيانات الخاصة به. 2. يقوم بالنقر على تسجيل.
الاستثناءات	عدم تعبئة بعض البيانات الخاصة به.

Use case	عرض حساب
الممثل الرئيسي	الطبيب.
الهدف	عرض الحساب الخاص بالطبيب.
الشروط السابقة	إن يكون مسجلاً دخوله الى النظام.
الدوافع	عرض معلومات الحساب الخاص بالطبيب.
السيناريو	1. يقوم الطبيب بتسجيل الدخول الى النظام. 2. يقوم الطبيب بعرض بياناته بالدخول الى الصفحة الخاصة به.
الاستثناءات	عدم تسجيل دخول المستخدم.

Use case	تعديل حساب
الممثل الرئيسي	الطبيب.
الهدف	تعديل الحساب الخاص بالطبيب.
الشروط السابقة	أن يكون مسجلاً دخوله الى النظام.
الدوافع	امكانية تعديل معلومات الحساب الخاص بالطبيب.
السيناريو	1. يقوم المستخدم بالدخول للصفحة الويب. 2. الدخول الى القسم الخاص بمعلومات الحساب. 3. تعديل المعلومات.
الاستثناءات	عدم انشاء الطبيب حساب خاص به.

Use case	تسجيل الدخول للنظام
الممثل الرئيسي	الطبيب.
الهدف	لكي يستخدم الموقع الالكتروني.
الشروط السابقة	- أن يكون مسجلاً دخوله الى النظام - أن يتم قبوله من قبل الطبيب المسؤول
الدوافع	ليتمكن الطبيب من إضافة المعلومات الطبية للمستخدمين وعرض سجلات المرضى.
السيناريو	1. يقوم الطبيب بالدخول الى الموقع الالكتروني. 2. إدخال اسم المستخدم الخاص به. 3. إدخال كلمة المرور الخاصة به. 4. الضغط على زر تسجيل الدخول. 5. التأكد من البيانات المدخلة. 6. إرجاع الطبيب الى صفحة تسجيل الدخول إذا وجد أي خطأ في البيانات المدخلة. 7. الدخول الى النظام اذا كانت كلمة المرور واسم المستخدم صحيحين.
الاستثناءات	عدم قبول حساب الطبيب من قبل الطبيب المسؤول.

Use case	تصفح أسماء المرضى
الممثل الرئيسي	الطبيب.
الهدف	تصفح اسماء المرضى.
الشروط السابقة	- إن يكون مسجلاً دخوله الى النظام. - إن يكون هناك على الأقل واحد من المرضى قام بمشاركته بتقرير طبي.
الدوافع	تصفح أسماء المرضى من اجل متابعة حالاتهم.
السيناريو	1. يقوم الطبيب بتسجيل الدخول الى النظام. 2. يقوم الطبيب بعرض أسماء المرضى. 3. النقر على اسم المريض المراد لكي يتم عرض البيانات الخاصة به.
الاستثناءات	عدم وجود أي مريض قد قام بإرسال له أي سجل طبي.

Use case	إضافة سجل
الممثل الرئيسي	الطبيب.
الهدف	إضافة سجل جديد.
الشروط السابقة	أن يكون مسجلاً دخوله الى النظام.
الدوافع	أن يقوم بإضافة السجلات الطبية على النظام وتحديد ما قام بإضافتها لمن من المرضى.
السيناريو	<ol style="list-style-type: none"> 1. يقوم الطبيب بالدخول الى النظام. 2. الذهاب الى صفحة إضافة سجل جديد. 3. يقوم باختيار نوع السجل المراد إضافته. 4. يقوم بتعبئة البيانات المطلوبة. 5. ثم يقوم باختيار المريض الذي له هذا السجل. 6. يقوم بالضغط على زر إضافة السجل.
الاستثناءات	<ul style="list-style-type: none"> - عدم إدخال بعض البيانات الضرورية في صفحة إضافة السجل. - عدم وجود حساب للمريض المراد.

Use case	إضافة معلومات للمرضى
الممثل الرئيسي	الطبيب.
الهدف	إضافة معلومات على الفحوصات وسجلات المرضى.
الشروط السابقة	يحتاج الطبيب الدخول الى النظام.
الدوافع	إضافة معلومات عن السجلات الطبية حتى تكون بشكل مفهوم لدى المرضى.
السيناريو	<ol style="list-style-type: none"> 1. يقوم الطبيب بالدخول الى النظام. 2. الذهاب الى قائمة أسماء المرضى. 3. النقر على المريض المراد لعرض السجل. 4. إضافة معلومات الى السجلات الخاصة بالمرضى. 5. ثم يقوم الطبيب بالنقر على ايقونة تم والعودة الى صفحة السجلات.
الاستثناءات	<ul style="list-style-type: none"> - أن يتم إضافة معلومات الى مريض آخر. - إضافة معلومات الى تقرير آخر.

Use case	عرض حسابات للمرضى
الممثل الرئيسي	الطبيب.
الهدف	عرض المعلومات الطبية وسجلات المرضى.
الشروط السابقة	يحتاج الطبيب الدخول الى النظام.
الدوافع	عرض معلومات الحسابات الخاصة بالمرضى.
السيناريو	<ol style="list-style-type: none"> 1. يقوم الطبيب بالدخول الى النظام. 2. الذهاب الى قائمة أسماء المرضى. 3. النقر على المريض المراد لعرض حسابه. 4. ثم يقوم الطبيب بالنقر على ايقونة تم والعودة الى صفحة السجلات.
الاستثناءات	أن يتم عرض معلومات الى مريض آخر.

Use case	تسجيل الخروج من النظام
الممثل الرئيسي	الطبيب.
الهدف	السماح للطبيب الخروج من النظام.
الشروط السابقة	أن يكون مسجلاً دخوله الى النظام.
الدوافع	السماح بالخروج من التطبيق بعد إجراء المطلوب.
السيناريو	1. أن يضغط الطبيب على زر تسجيل الخروج. 2. خروج الطبيب من النظام.
الاستثناءات	عدم تسجيل دخول الطبيب.

الوظائف المتعلقة بالطبيب المسؤول:

Use case	تسجيل الدخول للنظام
الممثل الرئيسي	الطبيب المسؤول.
الهدف	لكي يضيف و يحذف الأطباء.
الشروط السابقة	لديه اسم مستخدم وكلمة مرور.
الدوافع	لكي يتأكد من موثوقية الأطباء.
السيناريو	1. يقوم الطبيب المسؤول بفتح النظام. 2. ادخال اسم المستخدم الخاص به. 3. ادخال كلمة المرور الخاصة به. 4. الضغط على زر تسجيل الدخول. 5. إرجاع المسؤول الى صفحة تسجيل الدخول إذا وجد أي خطأ في البيانات المدخلة. 6. الدخول الى النظام اذا كانت كلمة المرور واسم المستخدم صحيحين.
الاستثناءات	خطء في ادخال الحساب الخاص به.

Use case	قبول حساب الطبيب
الممثل الرئيسي	الطبيب المسؤول.
الهدف	قبول حساب الطبيب من خلال النظام.
الشروط السابقة	يحتاج الطبيب المسؤول الدخول الى النظام.
الدوافع	قبول حسابات الأطباء والعمل على التأكد أن طبيب هو الطبيب المزاوول للمهنة.
السيناريو	1. يقوم الطبيب المسؤول بتسجيل الدخول الى النظام. 2. الذهاب الى صفحة الاطباء. 3. ظهور له جميع الأطباء. 4. يقوم بتأكيد حساب الطبيب عن طريق إضافة الطبيب.
الاستثناءات	قبول حساب طبيب غير مزاوول لمهنة الطب.

Use case	مسح طبيب
الممثل الرئيسي	الطبيب المسؤول.
الهدف	مسح طبيب من النظام.
الشروط السابقة	يحتاج الطبيب المسؤول الدخول الى النظام.
الدوافع	مسح طبيب من النظام بسبب أي شيء له علاقة بتوقف الطبيب عن عمله في مهنته.
السيناريو	<ol style="list-style-type: none"> 1. يقوم طبيب المسؤول بتسجيل الدخول الى النظام. 2. الذهاب الى قائمة الأطباء. 3. الضغط على ايقونة حذف التي تظهر بجانب كل طبيب. 4. يضغط المسؤول على ايقونة حذف وبعدها يتم الحذف بعد ظهور رسالة تأكيد الحذف.
الاستثناءات	أن يتم حذف طبيب غير الطبيب المراد.

Use case	تسجيل الخروج من النظام
الممثل الرئيسي	المسؤول.
الهدف	السماح للمسؤول الخروج من النظام.
الشروط السابقة	أن يكون مسجلاً دخوله الى النظام.
الدوافع	السماح بالخروج من النظام بعد إجراء المطلوب.
السيناريو	<ol style="list-style-type: none"> 1. أن يضغط المسؤول على زر تسجيل الخروج. 2. خروج المسؤول من التطبيق.
الاستثناءات	عدم تأكيد الخروج من النظام.

الوظائف والمتطلبات

تقسم الوظائف التي يقوم بها المشروع الى قسمين: المتطلبات الوظيفية، والمتطلبات غير الوظيفية.

المتطلبات الوظيفية:

وهي المتطلبات والوظائف التي سيقوم بها المشروع بشكل أساسي وتتمثل بالآتي:

- حفظ السجلات الخاصة بالمستخدم على الإنترنت.
- إمكانية الوصول إلى السجلات الخاصة بالمستخدم وعرضها مع إمكانية التعديل عليها او حذفها.
- النفاذ وتحميل السجلات باستخدام كاميرا الهاتف عن طريق التطبيق أو من خلال الملفات الموجودة على الهاتف مسبقاً.
- إمكانية مشاركة السجلات مع الأطباء وجهات الاختصاص.
- إمكانية إجراء مكالمة طوارئ من خلال التطبيق.
- إمكانية تصفح الأدوية الطبية.
- إمكانية تقدير الحالة الصحية للمستخدم من خلال قسم تحليل الفحوصات في تطبيق الهاتف.

المتطلبات الغير وظيفية:

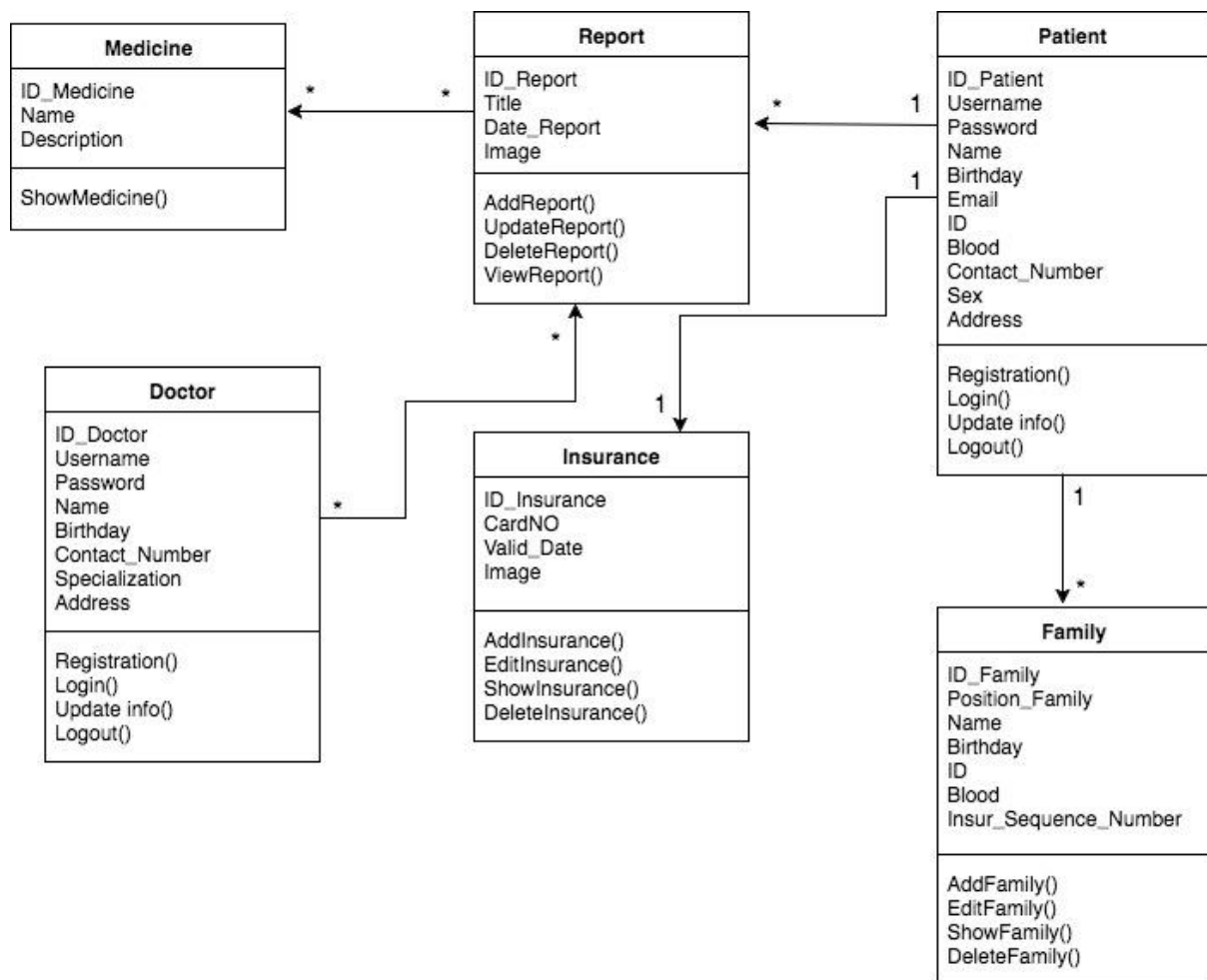
- سهولة التعامل مع التطبيق والموقع الإلكتروني:
يفضل أن تكون الواجهة سهلة التعامل والاستخدام لمستخدمي التطبيق والموقع حيث لا يوجد أي صعوبة في إجراء العمليات داخل النظام.
- الكفاءة العالية للنظام وسرعة الاستجابة:
تتضح كفاءة النظام من سرعة استقبال المدخلات الخاصة بالمستخدم بالإضافة الى سرعة تحليل الفحوصات الطبية الخاصة به وعمليات الإضافة والحذف، لا تستغرق أكثر من ثواني من عملية الضغط على الأزرار.
- ثقة المستخدمين بالنظام:
حيث أن النظام (الموقع الإلكتروني + التطبيق) يعمل على حفظ بيانات المستخدمين، مما يتطلب من تأمين تلك المعلومات وعدم افشائها، بالإضافة الى تقديم تحليل جيد للفحوصات الخاصة بالمستخدمين من أجل كسب ثقتهم.

تحليل البيانات

:Class Diagram

في هذا الجزء سيتم شرح Class Diagram التي تستخدم لوصف classes وتوضح العلاقات فيما بينهم، بحيث تمثل أو تصف هذه classes بناء العمليات وسلوكها داخل use cases وتقوم بتزويدنا بنموذج مفاهيمي حول الأجزاء الموجودة في النظام.

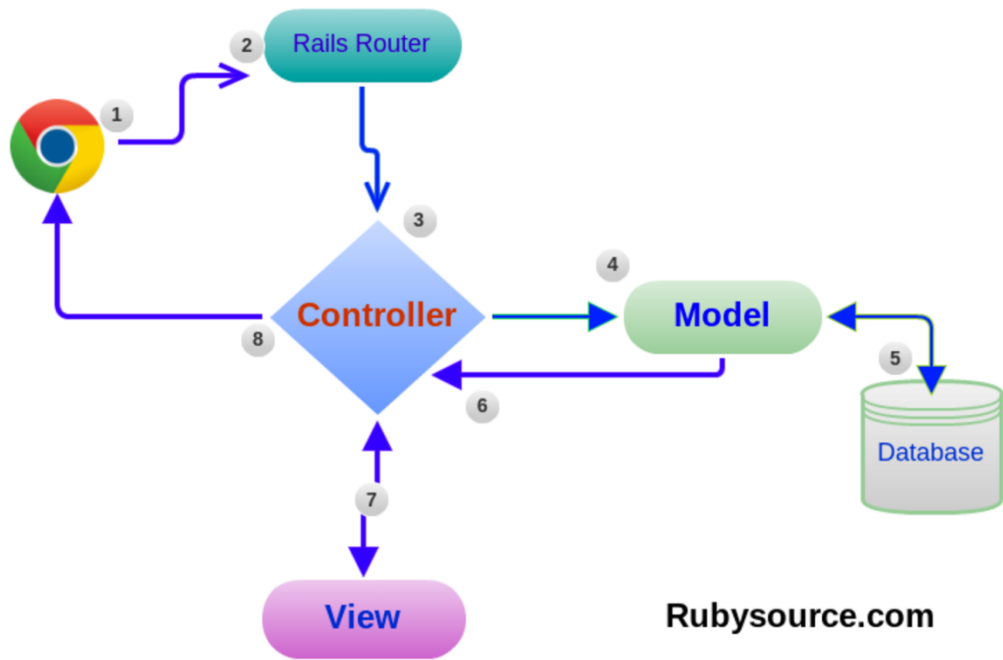
Classes الخاصة في التطبيق:



الشكل (2.2) يوضح Class diagram خاص بالتطبيق

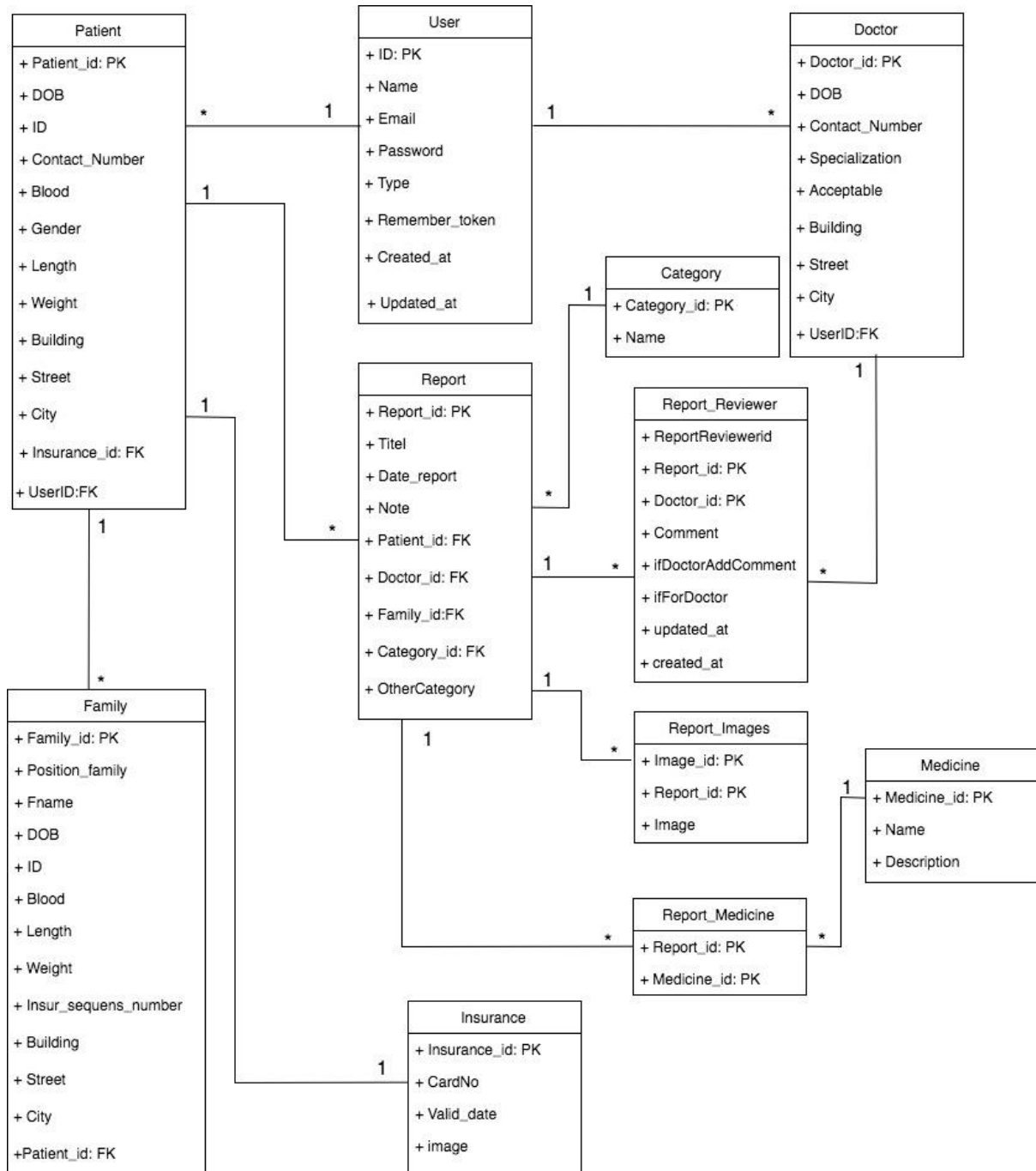
أجزاء النظام

MVC نمط يعتمد عليه برمجة الويب بشكل واسع في الأونة الاخيرة ، ويهدف MVC الى فصل النظام الى ثلاثة عناصر وتتفاعل مع بعضها البعض. حيث ان model يعمل على إدارة البيانات والعمليات المرتبطة بالبيانات، View يعمل على إدارة وعرض الصفحات لكل مستخدم، Controller يعمل على التحكم وإدارة التواصل بين Model، View.



الشكل (3.1) يوضح الهيكلية الأساسية للنظام

Database Design



الشكل (3.2) يوضح قواعد بيانات النظام

قواعد البيانات :

من خلال هذا القسم سيتم عرض قواعد بيانات النظام، و وصف الحقول، وسيتم عرض العلاقات بين الجداول على مخطط قاعدة البيانات :

User Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>User-ID</u>	Int	No	User Number	11	Yes	-
<u>Name</u>	Varchar	No	User Name	255	No	-
<u>Email</u>	Varchar	No	User Email	255	No	-
Password	Varchar	No	User Password	255	No	-
Type	Int	No	User Type	11	No	0
Remember-token	Varchar	Yes	User Token	100	No	Null
Update-at	Time stamp	Yes	Update Date	-	No	Null
Create -at	Time stamp	Yes	Creation Date	-	No	Null

Family Member Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Family-ID</u>	Int	No	Family Member Number	11	Yes	-
Patient-ID	Int	Yes	Patient Number	11	No	Unique
Family_ Position	Varchar	No	Family Member Position	30	No	-
Full Name	Varchar	No	Family Member Full Name	100	No	-
DOB	Varchar	No	Family Member Birthdate	30	No	-
ID	Int	Yes	Family Member ID	11	No	-
Blood	Int	No	Family Member Blood	10	No	-
Length	Varchar	No	Family Member Length	3	No	-
Weight	Tiny int	No	Family Member Weight	3	No	0
Insurance- sequence-number	Varchar	No	Family Member Insurance number	3	No	-
Building	Varchar	No	Family Member Address "building"	40	No	-
Street	Varchar	No	Family Member Address "street"	40	No	-
City	Varchar	No	Family Member Address "city"	40	No	-

Category Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
Category-Id	Int	No	Category Number	11	Yes	-
Name	Varchar	No	Category Name	30	No	-

Doctor Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Doctor -ID</u>	Int	No	Doctor Number	11	Yes	-
DOB	Varchar	No	Doctor birthdate	30	No	-
Contact-Number	Int	Yes	Doctor Contact-Number	11	No	-
Specialization	Varchar	No	Doctor Specialization	50	No	-
Acceptable	Tiny int	No	Doctor Acceptable	4	No	0
Building	Varchar	No	Doctor Address "Building "	30	No	-
Street	Varchar	No	Doctor Address "Street"	30	No	-
City	Varchar	No	Doctor Address "City"	30	No	-
User ID	Int	No	Doctor User ID	10	No	-

Patient Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Patient-ID</u>	Int	No	Patient Number	11	Yes	-
DOB	Varchar	No	Patient birthdate	30	No	-
ID	Int	No	Patient ID	11	No	-
Contact-Number	Int	Yes	Patient Contact-Number	11	No	Null
Blood	Varchar	No	Patient Blood	11	No	-
Gender	Varchar	No	Patient Gender	20	No	-
Length	Int	Yes	Patient Length	3	No	Null
Weight	Int	Yes	Patient Weight	3	No	Null
Building	Varchar	Yes	Doctor Address "Building "	30	No	Null
Street	Varchar	Yes	Doctor Address "Street"	30	No	Null
City	Varchar	Yes	Doctor Address "City"	30	No	Null
Insurance ID	Int	Yes	Patient Insurance	11	No	Null
User ID	Int	No	Doctor User ID	10	No	-

Insurance Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Insurance-ID</u>	Int	No	Insurance Number	11	Yes	-
Card NO	Int	No	Insurance Card Number	11	No	-
Valid-date	Varchar	No	Insurance Valid Date	30	No	-
Insurance Image	Varchar	Yes	Insurance Card Image	255	Yes	-

Medicine Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Medicine-ID</u>	Int	No	Medicine Number	11	Yes	-
Name	Varchar	No	Medicine Name	30	No	-
Description	Text	No	Medicine Description	-	No	-

Report Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Report-ID</u>	Int	No	Report Number	11	Yes	-
Title	Varchar	No	Report Title	100	No	-
Report Date	Varchar	No	Report Date	30	No	-
Note	Text	No	Report Note	-	No	-
Patient ID	Int	No	Patient Number	11	No	-
Family ID	Int	Yes	Family Number	11	No	Null
Doctor ID	Int	No	Doctor Number	11	No	-
Category ID	Int	Yes	Report Category	11	No	Null
Other Category	Varchar	Yes	Other Category	100	No	Null

Report Image Table

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Image -ID</u>	Int	No	Image Number	11	Yes	-
<u>Report-ID</u>	Int	No	Report Number	11	No	-
Image	Varchar	No	Report Image	255	No	-

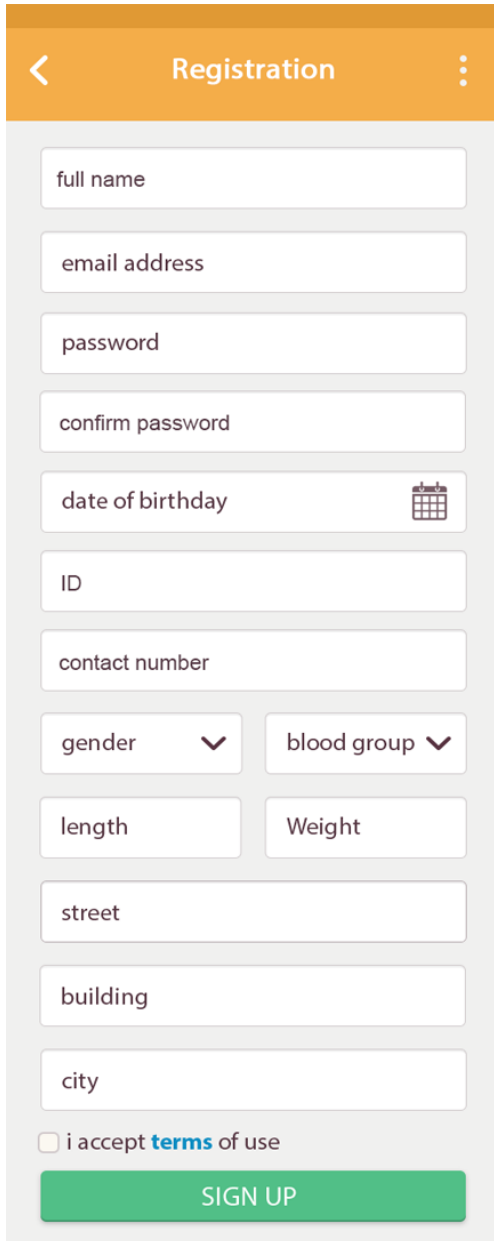
Report Reviewer

Name	Type	Can be null	Description	Length	Auto-increment	Default
<u>Report-review-ID</u>	Int	No	Report Review Number	11	Yes	-
<u>Report-ID</u>	Int	No	Report Number	11	No	-
<u>Doctor-ID</u>	Int	No	Doctor Number	11	No	-
Comment	Text	Yes	Doctor Comment	-	No	Null
If Doctor Add Comment	Tiny Int	No	If Doctor Add Comment	4	No	0
IF For Doctor	Tiny Int	No	IF For Doctor	4	No	0
Update-at	Time stamp	No	Update Date	-	No	-
Create -at	Time stamp	No	Creation Date	-	No	-

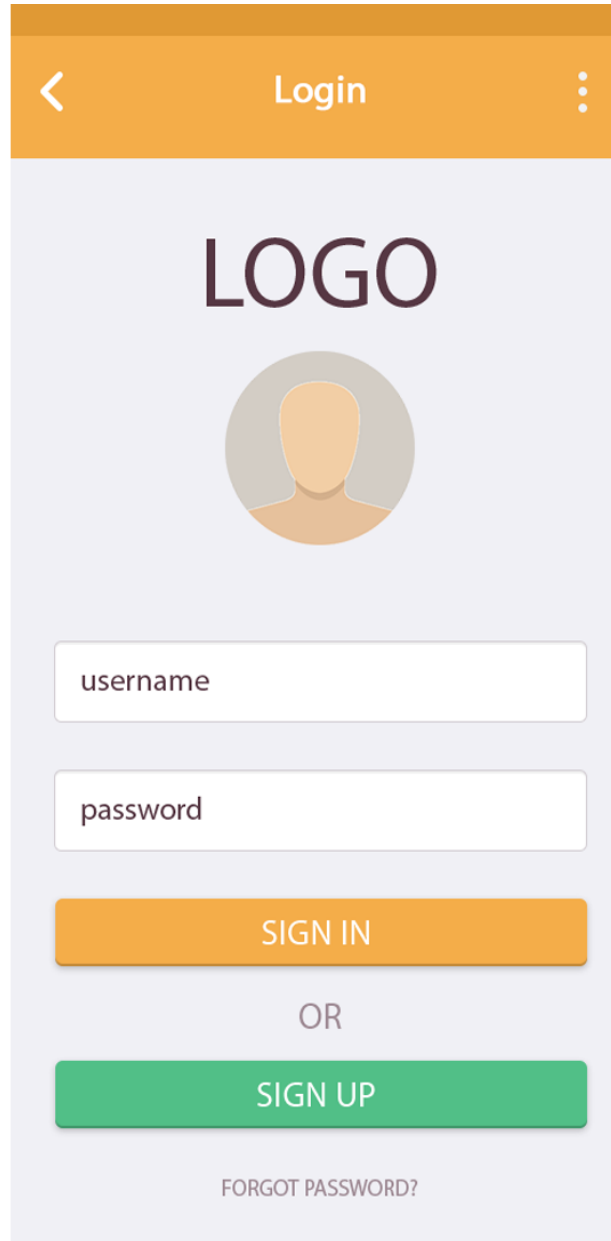
User Interface design

قمنا بتصميم الشاشات التي سوف يتعامل معها مستخدم النظام بشكل أولي في هذه المرحلة حيث تعتبر هذه مرحلة مؤقتة وأبدائية لبناء المشروع، يتم من خلالها توضيح العملية التي يسير بها النظام.

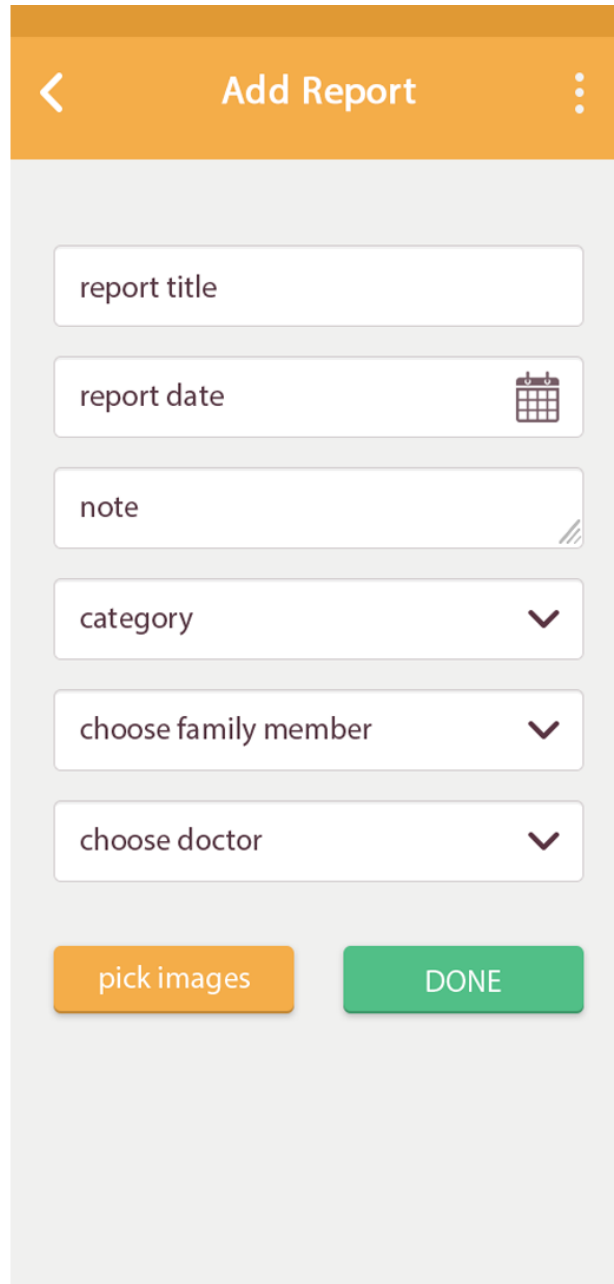
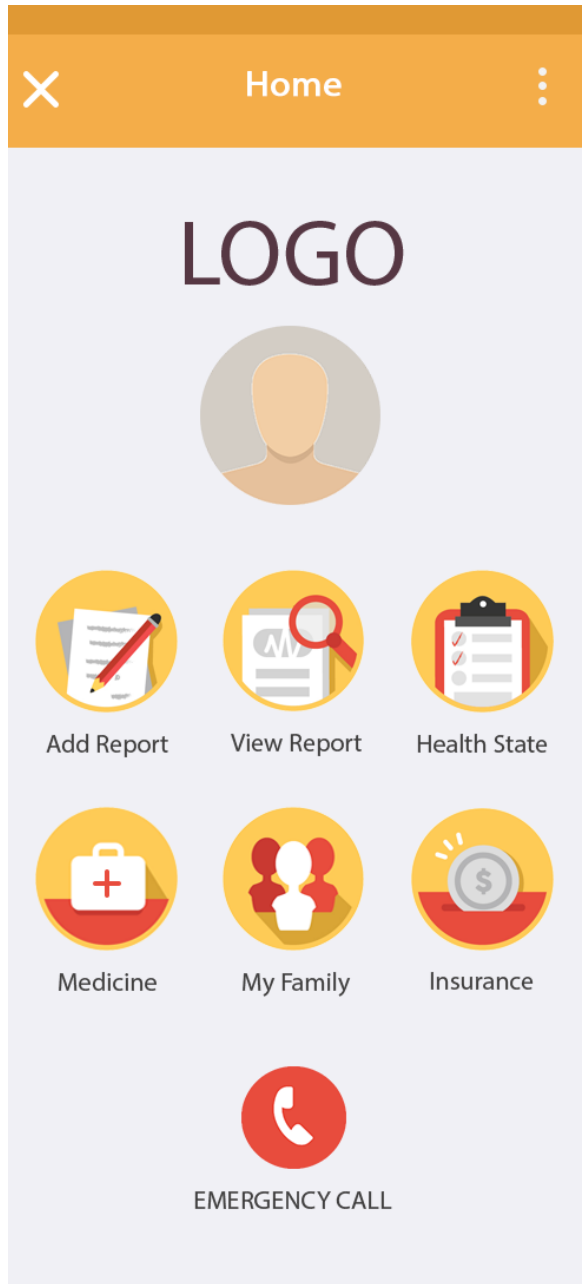
● شاشات التطبيق:



The registration screen features an orange header with a back arrow and a menu icon. The form includes fields for full name, email address, password, confirm password, date of birthday (with a calendar icon), ID, contact number, gender, blood group, length, and weight. It also has fields for street, building, and city. A checkbox for terms of use is present, followed by a green SIGN UP button.



The login screen features an orange header with a back arrow and a menu icon. It displays a large LOGO placeholder and a user profile icon. The form includes fields for username and password. Below the fields are two buttons: an orange SIGN IN button and a green SIGN UP button. The text 'OR' is centered between the buttons. At the bottom, there is a link for 'FORGOT PASSWORD?'.



< View Report ⋮

View as

Me ▼

SHOW

< View Report ⋮

Report Title number ONE

Report Title number TWO

Report Title number THREE

Report Title number FOURE

Report Title number FIVE

Report Title number SEX

Report Title number SEVEN

Report Title number EIGHT

Report Title number NINE


Report Title number TFN

View Report

Report Title Post Date

Category

Report Images



Patient Note

there no any note yet

Doctors Note

Ibrahim Youssef Mabad

Need more rest

08:18:52 2017-05-05

Edit Report

Share with another doctors

Delete Report

Health Status

check your Health status today

gender ▼

length

weight

pulses heart

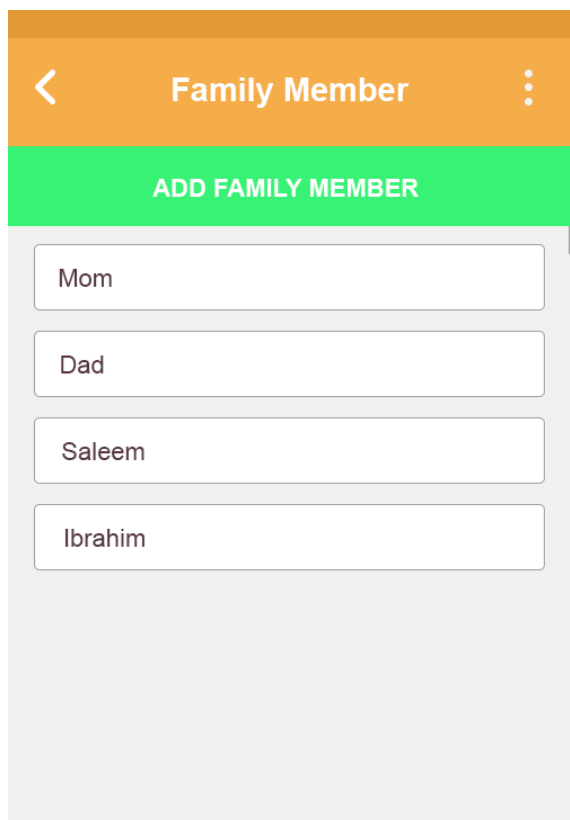
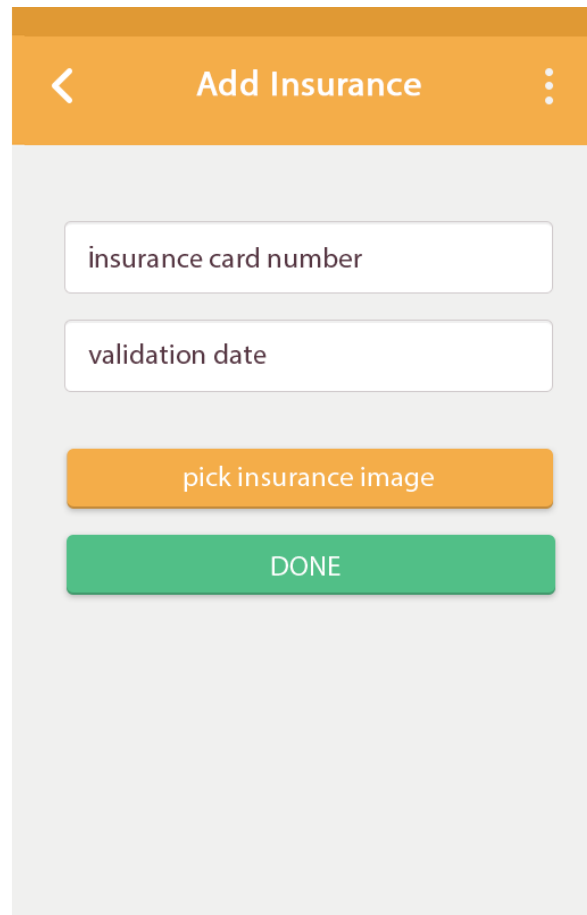
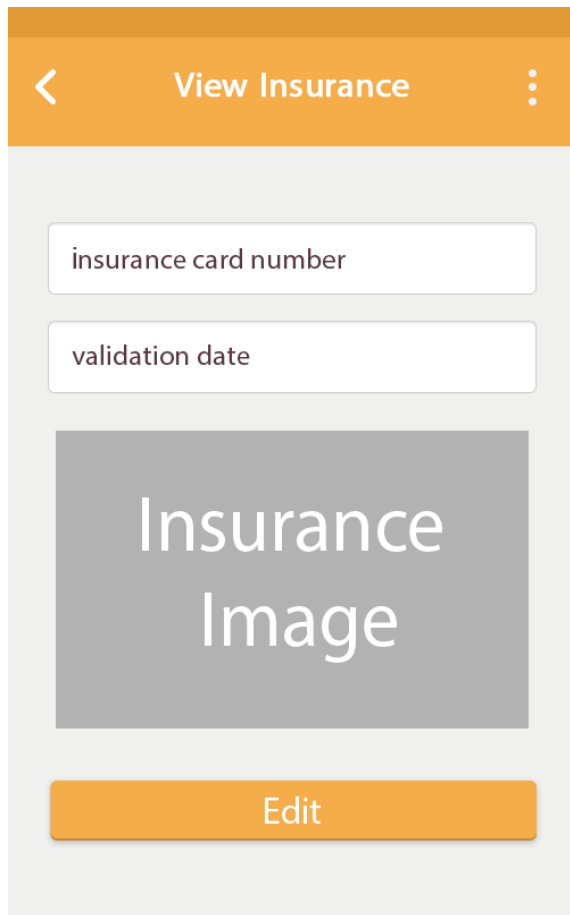
blood pressure

high blood pre .. low blood pre ..

fast blood sugar (FBS)

hemoglobin Hb

check your health




< Family Member ⋮

position in the family ▼

full name

insurance sequance number

ID

date of birthday 

blood ▼

length Weight

street

building

city

add member

< Family Member ⋮

position in the family

full name

insurance sequance number

ID

date of birthday

blood

length Weight

street

building

city

edit family member

delete family member

● شاشات الموقع الإلكتروني:

ECare Login Sign UP

Login

E-Mail Address

Password

Remember Me

[Forgot Your Password?](#)

ECare Login Sign UP

LOGO

This Page Just for Doctors

Register

Name

E-Mail Address

Password

Confirm Password

Date Of Birthday

Contact Number

Specialization

Building

Street

City



Dr.loay naseem said

Name: loay naseem said

Specialization: GP

Contact number.: 599887766

DOB: 2017-05-16

Building: bbb

Street: ssss




City: ccccc بلات

Edit

Patient Report



- Patient: Mahmoud naseem said 05-05-2017

Note: note from patient


[View](#)
- Patient: mohanad said 07-05-2017

Note: note1 note 1 note 1 note1 no...


[View](#)
- Patient: Mahmoud naseem said 03-05-2017

Note: note note note note







[View](#)
- Patient: Mahmoud naseem said 01-05-2017

Note: bbbbbb




[View](#)
- Patient: Mahmoud naseem said 07-05-2017

Note: noooooo nooot nooote notee...

[View](#)
- Patient: 555 06-05-2017

Note: ccccc



[View](#)

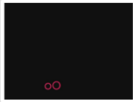
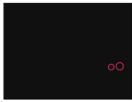
ECare Reports Shared for you Hi, Dr.loay naseem said

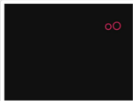
Titel: report5

Patient Name: [Mahmoud naseem said](#)

Category: categorycategory

Report Image



Note By Patient: Date: 05-05-2017
note from patient

Your Comments


[You](#) 2017-05-07 13:30:53
my comment comment comment comment comment comment comment
comment comment comment comment

[Dr.sh](#) 2017-05-07 13:35:28
comment from dr.sh


[You](#) 2017-05-07 13:36:20
asdaaaa

localhost:8000/drreport

ECare Reports Shared for you Hi, Dr.loay naseem said




Account Patient: Mahmoud naseem said

Gender: Female	DOB: 19-03-1997	Contact Number: 597554023
Blood: B-	Building: bbbss	Street: dfghjkl
City: cccfghj	Length :123	Weight :123
Insurance NO:4443	Insurance Valid Date:30-08-2017	Image Insurance: 

Patient Report:





Patient: 555 06-05-2017

Note: ccccc



Patient: Mahmoud naseem said 07-05-2017

Note: noooooo nooot noooote noteeeeeeee...

الفصل الرابع البناء و التنفيذ :

يحتاج المشروع إلى متطلبات لكي يتم تطويره و بناؤه بالشكل الكامل، وهذه المتطلبات إما أن تكون مادية أو برمجية، إن هذه المتطلبات تنقل مرحلة المشروع من الجانب النظري إلى الجانب العملي والتطبيقي لذلك فهي مهمة جداً في مرحلة بناء النظام ولا يمكن الاستغناء عنها، وتعتبر كلها ضرورية لإنهاء مرحلة بناء النظام.

المصادر البرمجية والتقنيات التي تم استخدامها في بناء مشروع:

في هذا الجزء سيتم تحديد الطريقة التي تم إتباعها في بناء النظام ولغات البرمجة التي تم استخدامها مع البرامج التي تم بناء المشروع عليها حيث أن النظام ككل يقسم إلى ثلاث أقسام وهي :

1. طبيب المسؤول.
2. الطبيب.
3. المريض.

بحيث تم استخدام لغة ال (PHP & Blade) من خلال استخدام تقنية الحديثة Laravel حيث تم انشاءها لتطوير تطبيقات الويب من خلال الاعتماد على MVC تم استخدامها من اجل بناء موقع الالكتروني لكي يمكن الطبيب المسؤول وطبيب من استخدام النظام.

وتم انشاء تطبيق يعمل على هواتف أندرويد حيث تم بناء من خلال استخدام لغة ال (JAVA) وتم بناء لكي يستطيع المريض من استخدام النظام .

وفيما يلي بعض من التعليمات البرمجية الخاصة بالموقع:

تعليمات البرمجية التالية هي لطبيب المسؤول توضح عملية عرض جميع الاطباء المعتمدين ولغير معتمدين لاستخدام النظام بعد قيامهم بعملية التسجيل في الموقع.

```
public function index()
{
    $doctorAccepted = DB::table('users')
    ->join('doctors', 'users.id', '=', 'doctors.UserID')
    ->select('users.name', 'doctors.Specialization', 'doctors.Doctor_id')
    ->where('doctors.Acceptable',1)
    ->orderBy('doctors.Doctor_id','DESC')
    ->get();
    $doctorsNotaccepted = DB::table('users')
    ->join('doctors', 'users.id', '=', 'doctors.UserID')
    ->select('users.name', 'doctors.Specialization', 'doctors.Doctor_id')
    ->where('doctors.Acceptable',0)
    ->orderBy('doctors.Doctor_id','DESC')
    ->get();
    return response()->view('doctor_dradminprofile', ['doctorsNotaccepted'=>$doctorsNotaccepted, 'doctorAccepted'=>$doctorAccepted]);
}
```

تعليمات البرمجة التالية هي لطبيب الذي تم قبول حساب الخاص به من قبل الطبيب المسؤول:

حيث تم استخدام Middleware لكي يتم توجيه كل مستخدم عند تسجيل دخوله الى النظام سواء كان مريض او طبيب مسؤول او طبيب.

```
<?php
namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Auth;
use App\Doctor;
class midForNormalDr
{
    /**
     * Handle an incoming request.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        $userID = Auth::id();
        $getDoctorData = Doctor::where('UserID', $userID)->first();
        if (Auth::user()->type==0){
            return response()->view('patient.patientweb');
        }elseif (Auth::user()->type==1){
            if($getDoctorData->Acceptable==0) {
                return response()->view('doctor.doctorWaitingAccept',['drData'=>$getDoctorData]);
            }
        }elseif(Auth::user()->type==2){
            return response()->redirectTo('dradmin');
        }
        return $next($request);
    }
}
```

تعليمات البرمجة التالية هي لطبيب:

من خلال Resource Controller التالي يتم عرض جميع التقارير الخاصة بالطبيب المسجل دخوله الى النظام وعند دخول الى صفحة الرئيسية للطبيب يتم تأكيد في البداية من انه قام بعملية تسجيل الدخول وتأكد من أن طبيب مقبول حسابه من خلال Middleware.

```
<?php
namespace App\Http\Controllers\web;
use App\ReportImage;
use App\ReportReviewers;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Auth;
use App\Doctor;
use App\Report;
use DB;
use Illuminate\Support\Facades\Input;

class contDrReport extends Controller
{
    public function __construct()
    {
        $this->middleware('checkForNormalDr');
        $this->middleware('auth');
    }
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $UserID = Auth::user()->id;
        $getDoctorID = DB::table('users')
            ->join('doctors', 'users.id', '=', 'doctors.UserID')
            ->where('users.id', $UserID)
            ->select('doctors.Doctor_id', 'users.name')
            ->get();
        foreach ($getDoctorID as $gdoctorID){
            $idDoctor = $gdoctorID->Doctor_id;
        }
        $getReportForDoctor = DB::table('reports')
            ->join('patients', 'reports.Patient_id', '=', 'patients.Patient_id')
            ->join('users', 'patients.UserID', '=', 'users.id')
            ->where('reports.Doctor_id', $idDoctor)
            ->orderBy('reports.Report_id', 'DESC')
            ->select('reports.Report_id', 'users.name', 'reports.Family_id', 'reports.Note', 'reports.Date_report')
            ->get();
        return response()->view('index', ['allReportForDoctor'=>$getReportForDoctor]);
    }
}
```

• يتم من خلال التعليمات التالية عرض التقرير الطبي للطبيب:

```
public function show($id)
{
    $getReport = Report::find($id);
    if (count($getReport)>0) {
        $idCategoryforreport = Report::find($id);
        if (!$idCategoryforreport->Category_id == null) {
            $allDateReport = DB::table('reports')
                ->join('patients', 'reports.Patient_id', '=', 'patients.Patient_id')
                ->join('users', 'patients.UserID', '=', 'users.id')
                ->join('categories', 'reports.Category_id', '=', 'categories.Category_id')
                ->where('reports.Report_id', $id)
                ->select('reports.Report_id', 'reports.Title', 'reports.Date_report',
                    'reports.otherCategory', 'reports.Category_id', 'patients.Patient_id', 'users.name', 'reports.Note', 'reports.Family_id',
                    'categories.Name', 'reports.Date_report')
                ->get();
            return view('doctor.showReport', ['datareport' => $allDateReport]);
        } else {
            $allDateReport = DB::table('reports')
                ->join('patients', 'reports.Patient_id', '=', 'patients.Patient_id')
                ->join('users', 'patients.UserID', '=', 'users.id')
                ->where('reports.Report_id', $id)
                ->select('reports.Report_id', 'reports.Title', 'reports.Date_report', 'reports.otherCategory', 'reports.Category_id',
                    'users.name', 'reports.Note', 'patients.Patient_id', 'reports.Family_id', 'reports.Date_report')
                ->get();
            return view('doctor.showReport', ['datareport' => $allDateReport]);
        }
    } else {
        return view('missing');
    }
}
```

تعليمات البرمجة التالية هي للمريض:

من خلال Volley في تطبيق الاندرويد يتم ارسال بيانات التقرير الطبي الى Server حيث يتم تخزين البيانات في Database.

تعليمات البرمجة الخاصة بتطبيق الاندرويد:

```
private void addReport(){
    strReportTitel = reportTitel.getText().toString();
    strDateReport = DateReport.getText().toString();
    strNote = reportNote.getText().toString();
    Integer IDCATEGORY = reportCategory.getSelectedItemsPosition();
    if (IDCATEGORY==6) {strOtherCategory = otherCategory.getText().toString();strIDCategory="";}
    Integer IDFamily = chooseFamilyMember.getSelectedItemsPosition();
    if (IDFamily==0){
        strIDFamilyIFExist="";
    }
    loading.show();
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    StringRequest stringRequest = new StringRequest(StringRequest.Method.POST,
        AppConfig.URL_FOR_Report, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            loading.dismiss();
            Log.d("res", response);
            strReportID = response+"";
            addImages();
        }
    },
        new Response.ErrorListener(){
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText(getApplicationContext(), "Please Connect to the Internet", Toast.LENGTH_LONG).show();
                loading.dismiss();
            }
        }){
        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> headers = new HashMap<>();
            // String credentials = "username:password";
            String auth = "Bearer "+access_token_string.toString();
            headers.put("Content-Type", "application/json");
            headers.put("Authorization", auth);
            headers.put(KEY_REPORTTITEL, strReportTitel);
            headers.put(KEY_DATEREPORT, strDateReport);
            headers.put(KEY_NOTE, strNote);
            headers.put(KEY_IDCATEGORY, strIDCategory);
            headers.put(KEY_IDFAMILYIFEXIST, strIDFamilyIFExist);
            headers.put(KEY_IDDOCTOR, strIDDoctor);
            headers.put(KEY_OTHERCATEGORY, strOtherCategory);
        }
    }
```

تعليمات البرمجة الخاصة بعملية حفظ البيانات في Database من خلال Server :

```
public function store(Request $request)
{
    $newReport = new Report();
    $UserID = Auth::id();
    $patDate = Patient::where('UserID', $UserID)->first();
    $newReport->Titel = $request->header("reporttitel");
    $newReport->Date_report = $request->header("datereport");
    $newReport->Note = $request->header("note");
    $newReport->Patient_id = $patDate->Patient_id;
    $newReport->Family_id = $request->header("familyid");
    if ($request->header("familyid") == "") {
        $newReport->Family_id = null;
    }
    $newReport->Doctore_id = $request->header("doctorid");
    $newReport->Category_id = $request->header("categoryid");
    if ($request->header("categoryid") == "") {
        $newReport->Category_id = null;
    }
    $newReport->otherCategory = $request->header("othercategory");
    $newReport->save();
    $newReportReviwer = new ReportReviewers();
    $newReportReviwer->Report_id = $newReport->Report_id;
    $newReportReviwer->Doctor_id = $request->header("doctorid");
    $newReportReviwer->save();
    if ($newReport) {
        return $newReport->Report_id;
    } else {
        return "not";
    }
}
```

- تعليمات البرمجية التالية ليتم عرض جميع افراد العائلة الذين تم اضافتهم من قبل وعرضهم في Spinner ليتمكن من عرض التقارير الخاصه لاحد افراد العائلة :

```

public void getAllFamily(){
//for spinner Choose FamilyMember
RequestQueue queue = Volley.newRequestQueue(getBaseContext());
final StringRequest stringRequest = new StringRequest(Request.Method.GET, AppConfig.URL_FamilyMember, new
@Override
public void onResponse(String response) {
Log.d("fffffffff", response);
try {
JSONArray data = new JSONArray(response);
familyMemberList.add(0, "Me");
idsFamilyMember.add(0, 0+ "");
for (int i = 0; i < data.length(); i++) {
familyMemberList.add(data.getJSONObject(i).getString("Fname"));
String id = data.getJSONObject(i).getString("Family_id");
idsFamilyMember.add(id);
}
ArrayAdapter<String> adapterFamilyMember = new ArrayAdapter<String>(ChooseFamilyViewReport.this,
choosePerson.setAdapter(adapterFamilyMember);
// adapterFamilyMember.notifyDataSetChanged();
} catch (JSONException e) {
e.printStackTrace();
}
}
}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
Toast.makeText(getBaseContext(), "Please Connect to the Internet", Toast.LENGTH_LONG).show();
}
}) {
@Override
public Map<String, String> getHeaders() throws AuthFailureError {
Map<String, String> headers = new HashMap<>();
// String credentials = "username:password";
String auth = "Bearer "
+access_token_string.toString() ;
headers.put("Content-Type", "application/json");
headers.put("Authorization", auth);
return headers;
}
};
queue.add(stringRequest);

```

```

choosePerson.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
if (choosePerson.getSelectedItemId() == 0) {
strIDPerson = 0+ "";
} else {
strIDPerson = idsFamilyMember.get(i);
}
}
@Override
public void onNothingSelected(AdapterView<?> adapterView) {
}
});
}

```

- تعليمات الخاصة لعملية حذف اي تقرير خاص بالمريض من خلال المريض :

من خلال API يتم ارسال من تطبيق الى server ومن ثم الى Resource Controller ليتم عملية حذف تقرير الخاص بالمريض .

```
private void setDeleteReport(){
    RequestQueue requestQueue = Volley.newRequestQueue(this);
    StringRequest stringRequest = new StringRequest(StringRequest.Method.DELETE,
        AppConfig.URL_FOR_Report+"/"+idReport, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if (response.equals("done")){
                Toast.makeText(getApplicationContext(),"Yes! Delete Report",Toast.LENGTH_LONG).show();
                new Handler().postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        Intent intent = new Intent(getApplicationContext(),MainActivity.class);
                        DetailsReport.this.finish();
                        startActivity(intent);
                        overridePendingTransition(R.anim.fade_in,R.anim.fade_out);
                    }
                },AppConfig.W_ANIT);
            }else{
                Toast.makeText(getApplicationContext(),"NOT Delete Insurance",Toast.LENGTH_LONG).show();
            }
        }
    },
    new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), "Please Connect to the Internet", Toast.LENGTH_LONG).show();
        }
    }){
        @Override
        public Map<String, String> getHeaders() throws AuthFailureError {
            Map<String, String> headers = new HashMap<>();
            String auth = "Bearer "
                +access_token_string.toString();
            headers.put("Content-Type", "application/json");
            headers.put("Authorization", auth);
            return headers;
        }
    };
    requestQueue.add(stringRequest);
}
```

```
public function destroy($id)
{
    $getReport = Report::find($id);
    $getReport->delete();
    $getImagesReport = ReportImage::where('Report_id', $id)->get();
    foreach ($getImagesReport as $idReportImage){
        ReportImage::where('Report_id',$idReportImage->Report_id )->delete();
    }
    $getImagesReport = ReportReviewers::where('Report_id', $id)->get();
    foreach ($getImagesReport as $idReportImage){
        ReportReviewers::where('Report_id',$idReportImage->Report_id )->delete();
    }
    return "done";
}
```

الفصل الخامس: فحص النظام

بعد ان تم بناء النظام بالكامل، أصبح من الضروري فحص أجزاء ومكونات للتأكد من أن النظام يعمل بالشكل المطلوب، وتعتبر هذه المرحلة من أهم مراحل بناء النظام وأكثرها تكلفة من حيث الوقت والجهد، وتكمن أهمية هذه المرحلة في ترابطها بمراحل بناء النظام.

فيما يلي الاشكال التالية توضح فحص شاشات النظام :

- الشكل التالي يوضح فشل تسجيل في الموقع الالكتروني:

The screenshot shows the 'Register' page of the E-Care system. The page title is 'This Page Just for Doctors'. The form contains the following fields and error messages:

- Name:** mahmoud naseem said
- E-Mail Address:** mahmoud310@mahmoud.com
- Password:** Password. Error: **The password confirmation does not match.**
- Confirm Password:** Password Confirmation
- Date Of Birthday:** dd/mm/yyyy
- Contact Number:** Contact number
- Specialization:** Specialization. Error: **The specialization may not be greater than 50 characters.**
- Building:** Building. Error: **The building may not be greater than 30 characters.**
- Street:** Street. Error: **The street may not be greater than 30 characters.**
- City:** City. Error: **The city may not be greater than 30 characters.**

- الشكل التالي يوضح فشل تسجيل الدخول للموقع :

The screenshot shows the 'Login' page of the E-Care system. The form contains the following fields and error messages:

- E-Mail Address:** mahmoud@mahmoud.com. Error: **These credentials do not match our records.**
- Password:** Password
- Remember Me
- Login** button
- [Forgot Your Password?](#)

• الشكل التالي يوضح فشل تسجيل في تطبيق:

The image shows a registration form with several input fields and error messages. The fields are: Full Name (with a red exclamation mark icon), Email (with a tooltip that says "Please fill the field"), Password (with a red exclamation mark icon), Confirm Password (with a red exclamation mark icon), Date of Birthday (with a calendar icon), and ID (with a red exclamation mark icon). A red error message at the bottom left states "Confirm Password did not match!".

• الشكل التالي يوضح فشل تسجيل دخول الى النظام

The image shows a login screen with a dark background. At the top is a "LOGO" and a silhouette of a person. Below this is a white error message box that says "Not valid Email or Password" with a red "OK" button. Below the error box are three buttons: "SIGN IN" (brown), "OR" (grey), and "SIGN UP" (green).

• الشكل التالي يوضح فشل اضافة تقرير جديد

The image shows a mobile app screen titled "Add Report". It has two input fields: "Report Title" (with a red exclamation mark icon) and "Date of Report" (with a tooltip that says "Please fill the field" and a red exclamation mark icon).

فيما يلي التعليمات البرمجية الخاصة بـ Units Testing :

تم عمل unit Testing للنظام من خلال laravel حيث يتيح لنا امكانية عمل TestCase لأجزاء النظام وفيما يلي التعليمات البرمجية الخاصة بـ TestCase :

```
class RoleTest extends BaseTestCase
{
    use CreatesApplication;
    public $baseUrl = 'http://localhost';
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testExample()
    {
        $this->seeInDatabase('users', ['email' => 'loay@loay.com']);
        $this->seeInDatabase('users', ['email' => 'naseem@naseem.com']);
        $user = factory(User::class)->create([
            'name' => 'mahmoud333333',
            'email' => 'mahmoudmahmoud@gg.com',
            'type' => 1,
        ]);
    }
}
```

```
public function testExample()
{
    $this->visit('/login')
        ->type('mahmoud@mahmoud.com ', 'email')
        ->type('123456', 'password')
        ->press('Login')
        ->see('Download');
    $this->visit('/doctor/id')
        ->see('Name');
}
```

```
public function testBasicTest()
{
    $this->visit('/home')
        ->seePageIs('/login');
    $response = $this->call('GET', '/drreport/11');
    $this->assertEquals(200, $response->status());
    $this->visit('register')
        ->see('for');
    $this->visit('drreport/7?id=3')
        ->see('Logout');
}
```