



بسم الله الرحمن الرحيم

جامعة الشيخ عبدالله البدرى



كلية علوم الحاسوب وتقانة المعلومات

بحث تكميلي لنيل درجة بكالوريوس الشرف في هندسة البرمجيات

تطبيق طبيبك

اعداد الطلاب :

- آيه عبد الناصر نعيم
- تيسير مرتضى مكي
- مازن كمال الدين ابراهيم الصافي

اشراف :

أ/خديجة محمد عبد الرحمن

ابريل 2026

اللاية

﴿يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ﴾

سورة المجادلة : 11

الشكر والعرفان

الحمد لله الذي بنعمته تتم الصالحات.

نتقدم بالشكر والتقدير إلى جامعة الشيخ عبدالله البدري لما وفّرت من دعم أكاديمي وبيئة تعليمية ساعدت في إنجاز هذا البحث. كما نتقدّم بالشكر الجزيل إلى مشرفة البحث على التوجيه والمتابعة والإرشاد العلمي القيم، وإلى جميع الأساتذة الكرام على ما قدّموه من علم ونصح.

ونشكر كذلك زملاءنا وكل من ساهم بمعلومة أو رأي أو دعم معنوي كان له أثر في إتمام هذا العمل، سائلين الله أن يجعل هذا الجهد نافعًا ومثمرًا.

الإهداء

نُهدي هذا العمل إلى والدينا وأسرنا الكريمة تقديرًا لدعمهم وتشجيعهم المستمر، وإلى أساتذتنا الأفاضل الذين كان لتوجيههم أثر كبير في إنجاز هذا البحث.

كما نُهدي هذا الجهد إلى جامعة الشيخ عبدالله البدري التي وفّرت لنا بيئة علمية ساعدتنا على التعلم والتطوير، وإلى كل من قدّم لنا يد العون خلال مراحل إعداد هذا البحث العلمي.

المستخلص

هدف هذا البحث إلى تحليل وتصميم وتنفيذ تطبيق إلكتروني لإدارة خدمات المستشفى في صورة تطبيق واحد يخدم ثلاث فئات رئيسية: المريض، الطبيب، والمسؤول. يوفر النظام للمريض إمكانية استعراض الأقسام والأطباء، الاطلاع على تفاصيل الطبيب، وحجز المواعيد ومتابعتها. كما يتيح للطبيب إدارة جدولته ومتابعة حجوزاته وتحديث بياناته ضمن صلاحيات محددة، ويمنح المسؤول أدوات لإدارة المستخدمين والصلاحيات، وتطوير ملفات الأطباء وربطها بحساباتهم، وإدارة الأقسام والجداول.

اعتمد البحث على إطار **Flutter** لتطوير واجهات المستخدم، مع استخدام خدمات **Firebase Authentication** و **Cloud Firestore** كخلفية للنظام، إضافة إلى تفعيل قواعد صلاحيات (**Security Rules**) لضمان حماية البيانات ومنع الوصول غير المصرح به. كما تم تطبيق آلية للحد من تعارض الحجوزات عبر تنظيم بيانات المواعيد بطريقة تقلل احتمالية تكرار الحجز لنفس الطبيب في نفس الوقت. أظهرت النتائج أن النظام يحقق المتطلبات الأساسية بكفاءة، مع تفعيل نظام الإشعارات اللحظية لتنبيه المستخدمين بحالة الحجوزات، مما يضمن دقة المتابعة، وتحسين إدارة الصور، ودعم السجل الطبي بشكل أوسع.

الكلمات المفتاحية: تطبيق مستشفى، حجز مواعيد، **Flutter**، **Firebase**، **Cloud Firestore**، صلاحيات، إدارة مستخدمين.

Abstract

This research aims to analyze, design, and implement a hospital service management system as a single application serving three main roles: patient, doctor, and admin. The system enables patients to browse departments and doctors, view doctor profiles, book appointments, and track their bookings. Doctors can manage schedules, view appointments, and update their profiles under controlled permissions. Admin users can manage accounts and roles, create doctor profiles and link them to user accounts, and manage departments and schedules.

Our team used Flutter to develop the user interface and adopted Firebase services (Authentication and Cloud Firestore) as the backend. Firestore Security Rules were configured to protect sensitive data and enforce role-based access control. In addition, an anti-conflict appointment strategy was applied by structuring appointment records in a way that reduces double-booking for the same doctor and time slot.

The results showed that the system efficiently meets the basic requirements, with the activation of the real-time notification system to alert users to the status of reservations, thus ensuring accurate follow-up, improving image handling, and extending medical record capabilities.

Keywords: Hospital application, Appointment booking, Flutter, Firebase, Cloud Firestore, Security Rules, Role-based access control.

الفهرس

الصفحة	العنوان	رقم العنوان
I	الاية	
li	شكر و عرفان	
lii	الاهداء	
lv	المستخلص	
Vi	الفهرس	
Viii	فهرس الأشكال	
الفصل الأول :الاطار العام للبحث		
1	مقدمة	1.1
1	مشكلة البحث	1.2
1	أهداف البحث	1.3
2	أهمية البحث	1.4
2	حدود البحث	1.5
2	منهجية البحث	1,6
2	ادوات البحث	1.7
3	هيكل البحث	1.8
الفصل الثاني : الاطار النظري والدراسات السابقة		
4	مقدمة	2.1
4	تطبيقات حجز المواعيد في المجال الصحي	2.2
5	التحول الرقمي في المستشفيات ودوره في تحسين الخدمة	2.3
6	قابلية الاستخدام وتجربة المستخدم في التطبيقات الصحية	2.4
7	تطبيقات الهاتف المحمول في المجال الصحي	2.5
7	أمن المعلومات والخصوصية في التطبيقات الصحية	2.6
8	قواعد البيانات السحابية ودورها في التطبيقات الصحية	2.7
9	إطار Flutter وأهميته في تطوير التطبيق	2.8

2.9	Firestore ودوره في التطبيق	9
2.10	تنظيم الصلاحيات داخل التطبيق	10
2.11	منع تعارض الحجوزات	11
2.12	الدراسات السابقة	11
الفصل الثالث: التحليل		
3.1	مقدمة	13
3.2	وصف التطبيق ونطاقه	13
3.3	أصحاب المصلحة (Stakeholders) والصلاحيات	15
3.4	المتطلبات الوظيفية (Functional Requirements)	16
3.5	المتطلبات غير الوظيفية (Non-Functional Requirements)	17
3.6	تحليل العمليات الأساسية في التطبيق	18
3.7	تحليل البيانات (Data Analysis)	20
3.8	مشكلة النظام الحالي	23
3.9	وصف النظام المقترح	24
3.10	تحليل مدخلات النظام	25
3.11	تحليل مخرجات النظام	26
3.12	مخططات UML لنظام طبيبك	27
3.13	مخطط حالة الاستخدام	27
3.14	مخطط التتابع	28
3.14	مخطط التفاعل	29
3.15	مخطط الاصناف	30
3.16	مخطط العلاقات والكيونة	31
3.17	مخطط سياق النظام	33
الفصل الرابع : التصميم		
4.1	مقدمة	28
4.2	تصميم معمارية التطبيق (Architecture)	28
4.3	التصميم التقني (Technology Stack)	28

4.4	تصميم قاعدة البيانات (Firestore Data Design)	28
4.5	قواعد الصلاحيات (Firestore Security Rules)	29
4.6	واجهات الادخال	31
4.7	واجهة المدير الرئيسية	
4.8	واجهة اضافة طبيب	32
4.9	واجهة إضافة الاقسام	33
4.10	تصميم شاشات التطبيق الطبي	34
4.11	واجهة الاقسام	34
4.12	واجهة المستخدم الرئيسية	34
4.13	واجهة المواعيد	34
4.14	واجهة سجلات المريض	35
4.15	واجهة حساب المستخدم	35
4.16	تعديل اعدادات المستخدم	35
الفصل الخامس : التنفيذ		
5.1	التنفيذ	33
5.2	الواجهات	34
الفصل السادس : النتائج والتوصيات		
6.1	مقدمة	50
6.2	النتائج (Results)	50
6.3	التوصيات (Recommendations)	52
6.4	المراجع (References)	53
6.5	الخاتمة	54

فهرس الجداول

الصفحة	العنوان	رقم الشكل
20	جدول مجموعة المستخدمين	3-1

3-2	جدول مجموعة الاقسام	20
3-3	جدول مجموعة الاطباء	21
3-4	مجموعة الجداول الزمنية	21
3-5	مجموعة الحجوزات	22
3-6	مجموعة السجلات الطبية	23
3-7	جدول تحليل المدخلات	25

فهرس الاشكال

3-8	مخطط الكينونة والعلاقات	27
5-1	واجهة المسؤول الرئيسية	34
5-2	واجهة اقسام المسؤول	35
5-3	واجهة الأطباء المسؤول	36
5-4	واجهة المستخدمين لدى المسؤول	37
5-5	واجهة الطبيب الرئيسية	38
5-6	واجهة جدول الطبيب	39
5-7	واجهة حجوزات الطبيب	40
5-8	واجهة حساب الطبيب	41
5-9	واجهة المستخدم الرئيسية	42
5-10	واجهة الاطباء والاقسام	43
5-11	واجهة عرض الطبيب والحجز	44
5-12	واجهة الحجوزات	45
5-13	واجهة السجل الطبي	46
5-14	واجهة الحساب	47
5-15	واجهة تسجيل الدخول	48
5-16	واجهة التسجيل الجديد	49

الفصل الأول:

الإطار العام للبحث

1.1 مقدمة

يشهد القطاع الصحي توسعًا متزايدًا في استخدام حلول التحول الرقمي لتحسين جودة الخدمة وتقليل زمن الانتظار وتسهيل الوصول للمعلومة والخدمة الطبية. من أكثر العمليات التي تتأثر بالازدحام وسوء التنظيم داخل المؤسسات الصحية هي عملية حجز المواعيد وإدارتها، حيث تؤدي الطرق التقليدية (الحضور المباشر/الاتصالات الهاتفية) إلى ضغط على موظفي الاستقبال وتأخر في الخدمة واحتمالات فقدان المعلومات أو تضارب المواعيد. لذلك يأتي هذا البحث لتطوير تطبيق طبيبك ويوفر إدارة للجدولة والحجوزات والملف الشخصي والسجل الطبي وفق صلاحيات واضحة.

1.2 مشكلة البحث

1. صعوبة وبطء إجراءات الحجز والمتابعة داخل المستشفيات عند الاعتماد على الطرق التقليدية.
2. طول زمن الانتظار وصعوبة الوصول للطبيب المناسب.
3. العبء الإداري الكبير على موظفي الاستقبال وإمكانية حدوث أخطاء في تسجيل المواعيد.
4. ضعف قدرة المريض على تتبع تاريخه الطبي ومواعيده بسهولة.

1.3 أهداف البحث

1.3.1 الهدف العام

تطوير تطبيق طبيبك لإدارة المستشفى يتيح للمريض استعراض الأطباء وحجز المواعيد، ويتيح للطبيب إدارة جدول حجوزاته، ويتيح للمسؤول إدارة المستخدمين والأطباء والأقسام.

1.3.2 أهداف فرعية

1. تطوير تطبيق حسابات وصلاحيات.(Admin/Doctor/Patient)
2. إدارة الأقسام والأطباء وربط الطبيب بحساب مستخدم.
3. إدارة الجدول الزمني للطبيب وتوليد مواعيد متاحة.
4. تطوير حجوزات مع منع التعارض .
5. عرض الحجوزات للمريض والطبيب حسب الصلاحية.
6. دعم سجل طبي مبدئي أو ملفات/ملاحظات .

1.4 أهمية البحث

تكمن الأهمية في توفير وسيلة رقمية تقلل الاعتماد على الإجراءات اليدوية وتسرع الحجز وتزيد رضا المستفيدين. وتؤكد أدبيات المجال أن أنظمة الحجز الإلكتروني تُستخدم لتجاوز مشكلات الحجز الحضوري وتحتاج تقييمًا مستمرًا لتحسين التجربة ومعالجة التحديات .

1.5 حدود البحث

- حدود موضوعية: الحجز، الجدولة، إدارة المستخدمين والصلاحيات، عرض/تحديث الملف الشخصي، سجل طبي مبسط.
- حدود مكانية: بربر – القدواب – مستشفى الشيخ البدري الجامعي .
- حدود زمنية: 2025 – 2026

1.6 منهجية البحث

تعتمد الدراسة على المنهج الوصفي لمسح الأدبيات السابقة والمنهج التحليلي لتصميم وبرمجة النظام:

- الدراسة النظرية (الإطار النظري): (مراجعة الدراسات السابقة والمفاهيم المتعلقة بالتحول الرقمي في القطاع الصحي، وفهم احتياجات إدارة المواعيد إلكترونياً.
- تحليل النظام (Analysis): دراسة المشكلة القائمة (الحجز التقليدي) وتحليل المتطلبات الوظيفية وغير الوظيفية للفئات المستهدفة (مريض، طبيب، مدير نظام).
- تصميم النظام (Design): تحويل المتطلبات إلى نماذج تقنية تشمل تصميم قاعدة البيانات (ERD) ، وتصميم واجهات المستخدم.
- التنفيذ البرمجي (Implementation): الجانب التطبيقي الذي يتم فيه تطوير التطبيق باستخدام الإطارات البرمجية المختارة وترجمتها إلى واقع تقني.
- الاختبار والتقييم (Testing & Evaluation): تشغيل التطبيق للتأكد من فاعلية بوابة الصلاحيات، ومنع تعارض الحجوزات، وتحقيق الأهداف المرجوة من البحث.

1.7 أدوات البحث

1. إطار العمل واللغة: تم استخدام إطار **Flutter** مع لغة **Dart** لتطوير واجهات التطبيق وبرمجته.
2. إدارة الحسابات: اعتمد التطبيق على **Firebase Authentication** لإدارة عمليات التسجيل والتحقق من الهوية

3. قاعدة البيانات: استخدمت **Cloud Firestore** كقاعدة بيانات سحابية لتخزين كافة بيانات المستخدمين، الأطباء، والحجوزات.
4. إدارة الحالة: تم استخدام أداة **Riverpod** لإدارة الحالة (State Management) داخل التطبيق.
5. التصميم: اعتمد التطبيق على مكتبة **Material 3** و **Google Fonts** لتصميم واجهات عصرية ومتجاوبة.
6. التوثيق: تم استخدام برنامج مايكروسوفت وورد للتوثيق وكتابته البحث.

1.8 هيكلية البحث

- الفصل الأول: الإطار العام.
- الفصل الثاني: الإطار النظري
- الفصل الثالث: تحليل النظام والمتطلبات.
- الفصل الرابع: التصميم.
- الفصل الخامس: التنفيذ والاختبار.
- الفصل السادس: النتائج والمصادر والمراجع

الفصل الثاني:

الإطار النظري

2.1 مقدمة

يشهد القطاع الصحي في السنوات الأخيرة تحولاً رقمياً متسارعاً نتيجة الحاجة إلى تحسين جودة الخدمات الصحية، وتقليل زمن الانتظار، ورفع كفاءة إدارة الموارد الطبية والبشرية. ومن بين أكثر المجالات التي استفادت من هذا التحول مجال إدارة المواعيد والحجوزات الطبية، حيث أسهمت التطبيقات الذكية في تسهيل وصول المرضى إلى الخدمات الصحية، وتحسين التنسيق بين المرضى والأطباء والإدارة داخل المؤسسات العلاجية.

وتبرز أهمية تطبيقات حجز المواعيد الطبية في كونها لا تقتصر على مجرد تسجيل موعد، بل تمثل بيئة خدمية متكاملة تساعد في تنظيم العمل داخل المؤسسة الصحية، وتمكين المستخدم من الوصول السريع إلى المعلومات، وتقليل الأخطاء الناتجة عن العمل اليدوي أو التواصل التقليدي عبر الهاتف أو الحضور المباشر. كما تساعد هذه التطبيقات في تحسين تجربة المستخدم من خلال توفير واجهات مرنة، وخدمات متاحة على مدار الساعة، وإمكانية متابعة الحجز وتعديله أو إلغائه بطريقة مباشرة.

وفي هذا الإطار، يعتمد هذا البحث على تطوير تطبيق طبيك لإدارة حجوزات مستشفى جامعة الشيخ عبدالله البدري، بحيث يخدم ثلاث فئات رئيسية: المريض، والطبيب، والمسؤول. ولأجل فهم الأساس العلمي والفني لهذا التطبيق، يتناول هذا الفصل المفاهيم النظرية المتعلقة بالحجوزات الإلكترونية في القطاع الصحي، وتجربة المستخدم في التطبيقات الصحية، والتقنيات البرمجية المستخدمة في التنفيذ، إضافة إلى أهم الدراسات السابقة المرتبطة بموضوع البحث.

2.2 تطبيقات حجز المواعيد في المجال الصحي

تُعد تطبيقات حجز المواعيد الطبية من أهم مخرجات التحول الرقمي في القطاع الصحي، إذ تهدف إلى أتمتة عملية الحجز وتنظيمها بصورة تقلل من الاعتماد على الإجراءات التقليدية. ففي كثير من المؤسسات الصحية، تتم إجراءات الحجز يدوياً أو عبر الهاتف، وهو ما يؤدي إلى ازدحام، وتأخير، وتكرار في المواعيد، وصعوبة في الوصول إلى معلومات دقيقة ومحدثة.

وتقوم فكرة تطبيقات الحجز الطبي على توفير منصة إلكترونية تسمح للمريض بالاطلاع على الأقسام الطبية، واستعراض الأطباء المتاحين، والتعرف على مواعيد العمل، ثم اختيار الموعد المناسب وتأكيد

الحجز. كما تمكّن الطبيب من متابعة جدول عمله والحجوزات المرتبطة به، وتمكّن الإدارة من متابعة المستخدمين وتنظيم البيانات الطبية والإدارية.

وتسهم هذه التطبيقات في تحقيق عدد من الفوائد، من أهمها:

- تقليل زمن انتظار المرضى.
- تخفيف الضغط على موظفي الاستقبال.
- الحد من التعارض بين المواعيد.
- تحسين استغلال وقت الأطباء.
- رفع مستوى رضا المرضى عن الخدمة.
- توفير بيانات دقيقة تساعد في الإدارة واتخاذ القرار.

ومن الناحية العملية، فإن نجاح أي تطبيق لحجز المواعيد الصحية يعتمد على توفر عناصر أساسية، مثل سهولة الاستخدام، ووضوح الواجهة، وسرعة الأداء، وأمن البيانات، ومرونة إدارة الجداول، وإمكانية التوسع مستقبلاً. لذلك فإن تصميم هذا النوع من التطبيقات يحتاج إلى دمج الجانب الصحي مع الجانب التقني بصورة مدروسة.

2.3 التحول الرقمي في المستشفيات ودوره في تحسين الخدمة

أصبح التحول الرقمي في المؤسسات الصحية ضرورة وليس خياراً، وذلك بسبب النمو المستمر في أعداد المراجعين، والحاجة إلى خدمات أسرع وأكثر دقة وشفافية. وتُعد تطبيقات الهاتف المحمول من أكثر الأدوات فاعلية في هذا المجال، لأنها تمنح المستخدم قدرة على الوصول إلى الخدمة من أي مكان وفي أي وقت.

وتنعكس أهمية التحول الرقمي في المستشفيات في عدد من الجوانب، منها:

- تحسين إدارة الوقت والموارد.
- تقليل المعاملات الورقية.
- تسهيل الوصول إلى البيانات الطبية والإدارية.
- دعم جودة القرار الإداري والطبي.

- تقليل الأخطاء الناتجة عن الإدخال اليدوي.
- تعزيز التواصل بين أطراف العملية الصحية.

كما أن التطبيقات الصحية تتيح للمريض الشعور بقدر أكبر من السيطرة على خدمته الصحية، حيث يمكنه متابعة مواعيده، والاطلاع على بياناته، ومعرفة تفاصيل الطبيب والقسم، بدلاً من الاعتماد الكامل على الموظف أو السجلات الورقية. ومن هنا فإن تطوير تطبيق مخصص لإدارة حجوزات المستشفى يمثل خطوة مهمة نحو تحديث العمل الصحي وجعله أكثر فاعلية وتنظيماً.

2.4 قابلية الاستخدام وتجربة المستخدم في التطبيقات الصحية

تُعد قابلية الاستخدام من أهم عوامل نجاح التطبيقات الصحية، لأن جمهور المستخدمين لا يقتصر على فئة تقنية متخصصة، بل يشمل المرضى، والأطباء، والإداريين، ولكل فئة احتياجات مختلفة وقدرات متفاوتة في التعامل مع التطبيقات. لذلك يجب أن يتسم التطبيق الصحي بالبساطة، والوضوح، وسهولة التنقل، وسرعة إنجاز المهام الأساسية.

ويقصد بقابلية الاستخدام مدى قدرة المستخدم على التعامل مع التطبيق بفعالية وكفاءة ورضا، مع أقل قدر ممكن من الأخطاء أو التعقيد. وفي التطبيقات الصحية تحديداً، تتضاعف أهمية هذا العنصر لأن أي تعقيد في الواجهة قد يؤدي إلى تأخير الحجز، أو إدخال بيانات خاطئة، أو عزوف المستخدم عن التطبيق بالكامل.

ومن أهم مبادئ تصميم تجربة المستخدم في التطبيقات الصحية:

- وضوح القوائم والعناصر الأساسية.
- تقليل عدد الخطوات اللازمة لتنفيذ الحجز.
- إظهار الرسائل والتنبيهات بصورة مفهومة.
- تنظيم الشاشات بحسب دور المستخدم.
- استخدام تسميات واضحة ومتوقعة.
- توفير تغذية راجعة فورية بعد كل عملية مهمة.

وفي هذا التطبيق، تم تقسيم الواجهات حسب الصلاحيات، بحيث تكون للمريض واجهة تناسب عملية الحجز والاستعراض، وللطبيب واجهة تركز على الجدول والحجوزات، والمسؤول واجهة مخصصة للإدارة والمتابعة. ويُعد هذا التقسيم من الأساليب الجيدة في تحسين قابلية الاستخدام وتقليل التشتت.

2.5 تطبيقات الهاتف المحمول في المجال الصحي

ازدادت أهمية تطبيقات الهاتف المحمول في تقديم الخدمات الصحية بسبب انتشار الهواتف الذكية واعتماد المستخدمين عليها في إنجاز كثير من الأنشطة اليومية. وأصبحت تطبيقات الصحة المتنقلة أداة فعالة في تقديم الخدمات، ومتابعة الحالات، وإرسال التنبيهات، وتنظيم المواعيد، وتسهيل التواصل.

وتتميز تطبيقات الهاتف في المجال الصحي بعدة خصائص، من أهمها:

- سهولة الوصول إلى الخدمة.
- إمكانية الاستخدام في أي وقت.
- السرعة في تنفيذ المهام.
- دعم التحديث الفوري للبيانات.
- المرونة في التوسع وإضافة مزايا جديدة.
- تحسين تجربة المستخدم مقارنة بالحلول الورقية أو التقليدية.

كما أن تطبيقات الهواتف تُعد مناسبة جدًا لبيئة مثل المستشفى الجامعي، لأن المستخدمين فيها متنوعون، والاحتياج إلى الحجز والمتابعة متكرر، والحاجة إلى السرعة والدقة مرتفعة. لذلك فإن اختيار تطوير تطبيق يعمل على الهاتف يمثل اختيارًا مناسبًا لطبيعة المشكلة موضوع البحث.

2.6 أمن المعلومات والخصوصية في التطبيقات الصحية

تُعد البيانات الصحية من أكثر أنواع البيانات حساسية، لأنها تتعلق بالمستخدمين، والمواعيد، والملاحظات الطبية، والبيانات الشخصية. ولهذا فإن أي تطبيق صحي يجب أن يراعي جوانب الأمن والخصوصية منذ مرحلة التحليل والتصميم، وليس فقط عند التنفيذ.

ويشمل أمن المعلومات في التطبيق الصحي عدة جوانب، منها:

- التحقق من هوية المستخدم قبل الدخول.
- التحكم في الصلاحيات بحسب الدور.
- حماية البيانات من الوصول غير المصرح به.
- تقليل تعارض الصلاحيات بين المستخدمين.
- حفظ البيانات بصورة منظمة وآمنة.
- ضمان أن كل مستخدم يرى فقط ما يخصه.

وفي التطبيق موضوع البحث تم الاعتماد على Firebase Authentication للتحقق من هوية المستخدم، وعلى Firestore Security Rules للتحكم في الصلاحيات، بحيث لا يستطيع المريض الاطلاع على حجوزات غيره، ولا يستطيع الطبيب الوصول إلى بيانات لا تخصه، بينما يمتلك المسؤول صلاحيات أوسع لإدارة البيانات. ويُعد هذا التوجه مناسباً لتطوير تطبيق صحي حديث يراعي الخصوصية والسرية.

2.7 قواعد البيانات السحابية ودورها في التطبيقات الصحية

تعتمد التطبيقات الحديثة بصورة متزايدة على قواعد البيانات السحابية بسبب ما توفره من مرونة وسهولة في الوصول والتحديث والتوسع. وفي التطبيقات الصحية، تكتسب قواعد البيانات أهمية خاصة لأنها تمثل الوعاء الذي تُخزن فيه بيانات المستخدمين، والأطباء، والأقسام، والجداول، والحجوزات، والسجلات الطبية.

ومن مزايا قواعد البيانات السحابية:

- الوصول السريع إلى البيانات من أكثر من جهاز.
- التحديث اللحظي للمعلومات.
- سهولة التوسع عند زيادة عدد المستخدمين.
- تقليل الحاجة إلى إدارة خوادم محلية معقدة.
- دعم النسخ الاحتياطي والاسترجاع.
- الربط المباشر مع التطبيقات المحمولة.

وقد تم في هذا ال استطبيق تخدام Cloud Firestore باعتبارها قاعدة بيانات NoSQL سحابية مناسبة لتطبيقات الهاتف، خاصة عندما تكون البيانات متنوعة البنية وتحتاج إلى تحديث سريع. كما أن Firestore تسمح بتقسيم البيانات إلى مجموعات ووثائق، وهو ما يسهل تنظيم بيانات المستخدمين والأطباء والجدول والحجوزات.

2.8 إطار Flutter وأهميته في تطوير التطبيق

يُعد Flutter من الأطر الحديثة والمفتوحة المصدر التي طورتها Google لتطوير تطبيقات متعددة المنصات باستخدام لغة Dart. وقد اكتسب Flutter انتشارًا واسعًا بسبب قدرته على تطوير واجهات تفاعلية جميلة وسريعة الأداء، مع تقليل الزمن والجهد اللازمين للتطوير.

ومن أهم أسباب اختيار Flutter في هذا التطبيق:

- سهولة تطوير واجهات حديثة ومتجاوبة.
- دعم مبدأ إعادة استخدام المكونات.
- السرعة في التطوير عبر خاصية Hot Reload.
- إمكانية تنظيم التطبيق في صورة Features أو Modules.
- التوافق الجيد مع Firebase.
- دعم مكتبات تصميم حديثة مثل Material Design.

كما أن Flutter مناسب للبحوث الأكاديمية والتطبيقية لأنه يجمع بين سهولة التطوير وقوة الأداء، ويُسهل تطوير واجهات متعددة لفئات مختلفة داخل التطبيق نفسه.

2.9 Firebase ودوره في التطبيق

Firebase عبارة عن منصة متكاملة تقدم خدمات سحابية تساعد في تطوير التطبيقات بسرعة وكفاءة. وتضم خدمات متعددة مثل المصادقة، وقواعد البيانات، والتخزين، والإشعارات، والتحليلات، وغيرها.

وتمثل Firebase خيارًا مناسبًا للتطبيقات التي تحتاج إلى سرعة في التطوير وربط مباشر بين الواجهة وقاعدة البيانات.

وفي هذا التطبيق تم الاعتماد على:

- Firebase Authentication لإدارة التسجيل والدخول.
- Cloud Firestore لتخزين البيانات.
- Security Rules للتحكم في الصلاحيات.

ويتميز استخدام Firebase في هذا التطبيق بأنه يقلل من التعقيد المرتبط بإدارة الخوادم، ويُسهل عملية التطوير والاختبار، كما يسمح بإضافة خدمات مستقبلية مثل التنبيهات والإشعارات ورفع الصور والتقارير.

2.10 تنظيم الصلاحيات داخل التطبيق

من الجوانب الأساسية في التطبيقات الخدمية متعددة المستخدمين وجود نظام صلاحيات وصلاحيات يحدد ما الذي يستطيع كل مستخدم الوصول إليه أو تنفيذه. وفي التطبيق محل الدراسة توجد ثلاث فئات رئيسية:

- المريض: يستعرض الأقسام والأطباء، ويحجز موعدًا، ويتابع حجوزاته.
- الطبيب: يطلع على جدولته وحجوزاته ويحدث بعض بياناته.
- المسؤول: يدير المستخدمين والأطباء والأقسام والجدول.

ويُعد الفصل بين الصلاحيات من الأمور المهمة جدًا، لأنه يمنع التداخل في الوظائف، ويحسن الأمن، ويساعد في تطوير واجهات مخصصة لكل مستخدم حسب طبيعة مهامه. كما يساهم في تبسيط تجربة الاستخدام، لأن كل فئة ترى ما تحتاج إليه فقط.

2.11 منع تعارض الحجوزات

واحدة من أهم المشكلات في تطبيقات المواعيد الطبية هي مشكلة تعارض الحجز، أي تسجيل أكثر من مريض في الوقت نفسه عند الطبيب نفسه. وإذا لم تُعالج هذه المشكلة فستفقد المنصة فائدتها الأساسية، لأن الهدف من التطبيق هو تنظيم المواعيد لا تكرارها أو إرباكها.

ولمنع هذا التعارض، لا بد من وجود آلية واضحة عند حفظ بيانات الحجز، مثل:

- التحقق من عدم وجود موعد سابق للطبيب في التاريخ والوقت نفسيهما.
- تنظيم السجلات بطريقة تمنع التكرار.
- اعتماد معرفات واضحة ومرتبطة بعناصر الحجز.
- التحقق من التوفر قبل تأكيد العملية.

وفي هذا التطبيق، تم اعتماد بنية بيانات تساعد في تقليل احتمالية تكرار الحجز، وذلك عبر تنظيم عناصر الموعد بصورة تميز بين الطبيب والتاريخ والوقت، وهو ما يدعم سلامة الحجز ويحسن موثوقية التطبيق.

2.12 الدراسات السابقة

توصل الباحثون من خلال استعراض الأدبيات العلمية المتعلقة بالتحول الرقمي والحجز الإلكتروني في القطاع الصحي إلى النتائج التالية:

1. دراسة المراجعة المنهجية لأنظمة المواعيد عبر الويب (2021): توصل الباحث peng zhao إلى أن أنظمة الحجز الإلكتروني تحقق أثراً إيجابية ملموسة، تشمل تقليل زمن انتظار المرضى، وتحسين مستوى الرضا العام، وتخفيف العبء الإداري عن كاهل الموظفين، مع ضرورة مراعاة اعتبارات الأمن والتكلفة.
2. دراسة تقييم تجربة المستخدم في الحجز الإلكتروني (2022): توصل الباحث qing ye إلى أن أنظمة الحجز الرقمية تتفوق على الطرق التقليدية (الحضور المباشر والاتصال الهاتفي) في معالجة تحديات الازدحام، لكنها تتطلب تقييماً وتحديثاً مستمراً لضمان معالجة التحديات التقنية التي قد تواجه المستخدمين.
3. دراسة الكفاءة التشغيلية وتدفق المراجعين (2023): توصل الباحث خالد يوسف إلى أن تطبيقات الحجز الطبي تساهم بشكل مباشر في تحسين الكفاءة التشغيلية للمؤسسات الصحية من خلال تنظيم تدفق المراجعين، وتقليل الضغط على مكاتب الاستقبال، ومنع تضارب المواعيد الذي كان يحدث في السجلات اليدوية.

4. دراسة معايير قابلية الاستخدام في التطبيقات الصحية (2024): توصل الباحث على محمود إلى أن نجاح التطبيقات الصحية لا يرتبط فقط بالوظائف البرمجية، بل يعتمد بشكل أساسي على "سهولة الاستخدام" ووضوح الواجهات، حيث أن أي تعقيد في الوصول للمعلومة قد يؤدي إلى عزوف المرضى عن استخدام النظام.
5. دراسة أمن المعلومات وصلاحيات الوصول السحابي (2025): توصل الباحث احمد عبد الله إلى أن التكامل بين قواعد البيانات السحابية ونظام الصلاحيات الدقيق يعد متطلباً جوهرياً في التطبيقات الصحية؛ وذلك لضمان حماية البيانات الشخصية الحساسة ومنع الوصول غير المصرح به، مما يعزز موثوقية النظام لدى المستخدمين.

الفصل الثالث:

تحليل التطبيق والمتطلبات

3.1

يهدف هذا الفصل إلى تحليل التطبيق المقترح من حيث الوظائف الأساسية، والفئات المستفيدة، وحدود التطبيق، والمتطلبات الوظيفية وغير الوظيفية، إضافة إلى تحليل البيانات والعمليات الرئيسية التي يعتمد عليها التطبيق. ويُعد التحليل خطوة أساسية لفهم ما يجب أن يقدمه التطبيق فعليًا للمستخدمين، وكيفية تنظيم البيانات والصلاحيات بطريقة تضمن سهولة الاستخدام وصحة الأداء.

ويستند هذا الفصل إلى أن تطبيق طبيبك عبارة عن تطبيق أندرويد لإدارة حجوزات مستشفى جامعة الشيخ عبدالله البدري الجامعي، وليس نظامًا مؤسسيًا عامًا أو منصة متعددة البيئات. لذلك فإن التحليل هنا يركز على خصائص التطبيق من حيث الشاشات، والصلاحيات، وتدفق العمليات، وآلية التفاعل بين المستخدم وقاعدة البيانات.

3.2 وصف التطبيق ونطاقه

3.2.1 وصف مختصر للتطبيق

طبيبك هو تطبيق أندرويد يهدف إلى تنظيم حجوزات المستشفى وإدارة بعض العمليات الأساسية المرتبطة بها، ويخدم ثلاث فئات رئيسية من المستخدمين:

أولاً: المريض

ويستطيع من خلال التطبيق:

- تطوير حساب أو تسجيل الدخول.
- استعراض الأقسام الطبية.
- استعراض الأطباء وبياناتهم.
- الاطلاع على المواعيد المتاحة.
- حجز موعد مع الطبيب المناسب.
- متابعة الحجوزات الخاصة به.
- إدارة بعض بياناته الشخصية.

ثانياً: الطبيب

ويستطيع من خلال التطبيق:

- تسجيل الدخول لحسابه.
- الاطلاع على جدولته.
- متابعة الحجوزات المرتبطة به.

- تحديث بعض بياناته المرتبطة بملفه.
- إدارة الأوقات المتاحة ضمن حدود الصلاحية.

ثالثاً: المسؤول

ويستطيع من خلال التطبيق:

- إدارة المستخدمين.
- تعديل الصلاحيات.
- تطوير ملفات الأطباء وربطها بالحسابات.
- إدارة الأقسام.
- إدارة الجداول.
- متابعة البيانات الأساسية الخاصة بالتطبيق.

3.2.2 نطاق التطبيق

يشمل نطاق التطبيق الوظائف الأساسية اللازمة لإدارة الحجز داخل المستشفى بصورة إلكترونية. ويركز التطبيق على الجوانب التالية:

- إدارة الحسابات وتحديد الصلاحيات.
- عرض الأقسام والأطباء.
- إدارة ملفات الأطباء.
- إدارة الجداول الزمنية.
- تطوير الحجوزات ومتابعتها.
- التحكم في الصلاحيات والوصول إلى البيانات.

أما الوظائف الخارجة عن نطاق التطبيق في هذه المرحلة فتشمل:

- الدفع الإلكتروني.
- الربط مع أنظمة مختبرات أو صيدليات.
- إدارة تقارير طبية متقدمة.
- التكامل مع أنظمة خارجية واسعة.
- الإشعارات المتقدمة وخدمات الرسائل في نسخته الحالية.

3.2.3 حدود التطبيق

داخل حدود التطبيق

- تسجيل الدخول والتحقق من الهوية.
- تحديد دور المستخدم.
- عرض الشاشات حسب الدور.
- إدارة بيانات الأطباء والأقسام.
- إدارة المواعيد والجدول.
- حفظ الحجوزات في قاعدة البيانات.
- حماية البيانات بواسطة قواعد الصلاحيات.

خارج حدود التطبيق

- أنظمة الدفع.
- الربط مع أجهزة طبية أو أنظمة خارجية.
- أنظمة محاسبية أو إدارية مستقلة.
- إدارة ملفات طبية موسعة بصورة كاملة.

3.3 أصحاب المصلحة والصلاحيات

3.3.1 المريض

يُعد المريض المستفيد الرئيسي من التطبيق، لأنه يستخدمه للوصول إلى خدمة الحجز بصورة مباشرة. وتتمثل احتياجاته في:

- واجهة بسيطة وواضحة.
- سهولة الوصول إلى الطبيب المناسب.
- معرفة الأوقات المتاحة.
- إتمام الحجز بسرعة.
- القدرة على متابعة حجوزاته السابقة والحالية.

3.3.2 الطبيب

يُعد الطبيب أحد المستخدمين الأساسيين للتطبيق، إذ يحتاج إلى وسيلة منظمة لمتابعة الحجوزات والجدول. وتتمثل احتياجاته في:

- الاطلاع على المواعيد المرتبطة به.

- معرفة بيانات المرضى المرتبطين بالحجوزات عند الحاجة.
- إدارة الأوقات أو الجدول المتاح.
- تحديث بعض البيانات المهنية أو الشخصية المرتبطة به.

3.3.3 المسؤول

المسؤول هو المسؤول الإداري عن متابعة التطبيق من ناحية إدارة البيانات والصلاحيات. وتتمثل مهامه في:

- إدارة المستخدمين.
- تعيين الصلاحيات.
- تطوير ملفات الأطباء.
- ربط الطبيب بحساب المستخدم.
- إضافة الأقسام وتعديلها.
- متابعة تنظيم البيانات داخل التطبيق.

3.3.4 المستشفى

تمثل المستشفى الجهة المستفيدة من التطبيق على المستوى المؤسسي، إذ يساعدها التطبيق في:

- تنظيم الحجوزات.
- تقليل العبء على الاستقبال.
- تحسين انسياب العمل.
- دعم التحول الرقمي.
- توفير صورة أفضل للخدمة الصحية المقدمة.

3.4 المتطلبات الوظيفية

يقصد بالمتطلبات الوظيفية الخدمات والعمليات التي يجب أن ينفذها التطبيق بصورة مباشرة.

3.4.1 متطلبات عامة

- يجب أن يتيح التطبيق تطوير حساب جديد للمستخدم.
- يجب أن يتيح تسجيل الدخول والخروج.
- يجب أن يحدد التطبيق دور المستخدم بعد تسجيل الدخول.

- يجب أن يوجّه التطبيق المستخدم إلى الواجهة المناسبة حسب دوره.
- يجب أن يخزن التطبيق بيانات المستخدمين بطريقة منظمة داخل قاعدة البيانات.

3.4.2 متطلبات المريض

- يجب أن يستطيع المريض استعراض الأقسام الطبية.
- يجب أن يستطيع استعراض قائمة الأطباء.
- يجب أن يستطيع تصفية الأطباء حسب القسم.
- يجب أن يستطيع الاطلاع على بيانات الطبيب.
- يجب أن يستطيع رؤية الأوقات المتاحة للحجز.
- يجب أن يستطيع حجز موعد.
- يجب أن يستطيع متابعة الحجوزات الخاصة به.
- يجب أن يستطيع تعديل بعض بياناته الشخصية.

3.4.3 متطلبات الطبيب

1. يجب أن يستطيع الطبيب تسجيل الدخول إلى حسابه.
2. يجب أن يستطيع عرض الجدول الخاص به.
3. يجب أن يستطيع متابعة الحجوزات المرتبطة به.
4. يجب أن يستطيع تحديث بعض بياناته.
5. يجب أن يستطيع إدارة الجدول أو الأوقات المتاحة وفق الصلاحية المحددة له.

3.4.4 متطلبات المسؤول

- يجب أن يستطيع المسؤول إدارة المستخدمين.
- يجب أن يستطيع تغيير دور المستخدم.
- يجب أن يستطيع تطوير ملف طبيب جديد.
- يجب أن يستطيع ربط ملف الطبيب بحساب المستخدم.
- يجب أن يستطيع إدارة الأقسام الطبية.
- يجب أن يستطيع إدارة جداول الأطباء.
- يجب أن يستطيع تفعيل أو تعطيل بعض السجلات حسب الحاجة.

3.5 المتطلبات غير الوظيفية

المتطلبات غير الوظيفية هي الخصائص التي تحدد جودة التطبيق وكفاءته ودرجة قبوله عند الاستخدام.

3.5.1 الأمان والخصوصية

يجب أن يضمن التطبيق عدم وصول أي مستخدم إلى بيانات لا تخصه، وأن يتم تقييد القراءة والكتابة حسب الدور والملكية.

3.5.2 سهولة الاستخدام

يجب أن تكون واجهات التطبيق بسيطة وواضحة، وأن تكون خطوات الحجز مفهومة وسهلة التنفيذ.

3.5.3 الأداء

يجب أن يستجيب التطبيق بسرعة مناسبة عند عرض البيانات أو تنفيذ الحجز أو تحميل القوائم.

3.5.4 الاعتمادية

يجب أن يعمل التطبيق بطريقة مستقرة، وأن يحافظ على صحة البيانات وعدم ضياعها أثناء العمليات الأساسية.

3.5.5 قابلية التوسع

يجب أن يكون تصميم التطبيق مرناً بما يسمح بإضافة مزايا جديدة مستقبلاً، مثل الإشعارات أو توسيع السجل الطبي.

3.5.6 التوافق

يجب أن يكون التطبيق مناسباً للعمل على بيئة أندرويد المستهدفة، مع واجهات متوافقة مع أحجام الشاشات المختلفة قدر الإمكان.

3.6 تحليل العمليات الأساسية في التطبيق

3.6.1 عملية التسجيل وتسجيل الدخول

تبدأ العملية بقيام المستخدم بتطوير حساب جديد أو تسجيل الدخول إذا كان الحساب موجوداً مسبقاً. وبعد نجاح التحقق من الهوية، يتم جلب بيانات المستخدم من قاعدة البيانات لمعرفة الدور الخاص به، ثم يتم توجيهه إلى الواجهة المناسبة.

3.6.2 عملية تحديد الدور

بعد تسجيل الدخول، يعتمد التطبيق على قيمة الدور المخزنة في ملف المستخدم لتحديد ما إذا كان المستخدم مريضاً أو طبيباً أو مسؤول، وتطوير على ذلك تظهر له الواجهة والوظائف المسموح بها.

3.6.3 عملية استعراض الأطباء والأقسام

يستطيع المريض الدخول إلى شاشة الأقسام أو قائمة الأطباء، ثم استعراض البيانات الأساسية لكل طبيب مثل الاسم والتخصص والقسم والوصف المختصر، ومن ثم اختيار الطبيب المناسب.

3.6.4 عملية عرض الجدول والأوقات المتاحة

بعد اختيار الطبيب، يعرض التطبيق الأوقات المتاحة تطوير على جدول الطبيب المخزن في قاعدة البيانات، بحيث يرى المستخدم الأيام والفترات التي يمكن الحجز خلالها.

3.6.5 عملية الحجز

عندما يختار المريض التاريخ والوقت المناسبين، يقوم التطبيق بتطوير سجل جديد للحجز يتضمن بيانات المريض والطبيب والقسم والتاريخ والوقت والحالة. وقبل تأكيد الحجز، يتم التحقق من عدم وجود تعارض مع حجز سابق لنفس الطبيب في الزمن نفسه.

3.6.6 عملية متابعة الحجوزات

يمكن للمريض عرض حجوزاته الخاصة فقط، كما يستطيع الطبيب عرض الحجوزات المرتبطة به فقط، بينما يتمتع المسؤول بقدرة أوسع على المتابعة والإدارة بحسب الصلاحية.

3.6.7 عملية إدارة الأطباء

يستطيع المسؤول تطوير ملف طبيب جديد وتخزين البيانات الأساسية الخاصة به، ثم ربط هذا الملف بحساب مستخدم حتى يتمكن الطبيب من الدخول إلى التطبيق بصفته المهنية.

3.6.8 عملية إدارة الأقسام

يقوم المسؤول بإضافة الأقسام الطبية أو تعديلها أو تعطيلها بحسب الحاجة، وذلك لتسهيل تصنيف الأطباء وتحسين تجربة التصفح والحجز.

3.6.9 عملية إدارة الجداول

تتعلق هذه العملية بتحديد أيام العمل والفترات الزمنية المتاحة للطبيب. وقد تتم هذه الإدارة بواسطة المسؤول أو الطبيب حسب التصميم المعتمد داخل التطبيق.

3.7 تحليل البيانات (Data Analysis)

3.7.1 قاموس البيانات Data Dictionary

1. مجموعة المستخدمين (users)

تحتوي هذه المجموعة على البيانات الأساسية لكل مستخدم في النظام.

الحقل (Field)	النوع (Type)	الوصف/الملاحظات
uid	Document ID	المعرف الفريد الخاص بالمستخدم.
displayName	String	الاسم الذي سيتم عرضه في التطبيق.
email	String	البريد الإلكتروني للمستخدم.
emailLower	String	البريد الإلكتروني بحروف صغيرة (لأغراض البحث).
role	String	دور المستخدم. (admin/doctor/patient)
createdAt	Timestamp	تاريخ ووقت إنشاء الحساب.
isActive	Boolean	حالة الحساب (نشط أم لا).

جدول رقم (1-3)

2. مجموعة الأقسام (departments)

تستخدم لتصنيف التخصصات الطبية داخل المستشفى.

الحقل (Field)	النوع (Type)	الوصف/الملاحظات
name	String	اسم القسم الطبي.
isActive	Boolean	حالة القسم (متاح للحجز أم لا).

جدول رقم (2-3)

3. مجموعة الأطباء (doctors)

تحتوي على الملفات المهنية للأطباء وربطها بحسابات المستخدمين.

الوصف/الملاحظات	النوع (Type)	الحقل (Field)
اسم الطبيب المعروض في القائمة.	String	name
الدرجة العلمية أو التخصص (مثل). Consultant :	String	title
معرف القسم الذي ينتمي إليه الطبيب.	String	departmentId
البريد الإلكتروني للطبيب (اختياري).	String?	email
البريد الإلكتروني بحروف صغيرة للبحث والربط.	String?	emailLower
المعرف (uid) من مجموعة المستخدمين عند ربط الحساب.	String?	userId
نبذة مختصرة عن الطبيب.	String?	bio
حالة ملف الطبيب في النظام.	Boolean	isActive

جدول رقم (3-3)

4. مجموعة الجداول الزمنية (schedules)

تحدد أوقات العمل المتاحة لكل طبيب.

الوصف/الملاحظات	النوع (Type)	الحقل (Field)

الوصف/الملاحظات	النوع (Type)	الحقل (Field)
معرف ملف الطبيب.	String	doctorId
معرف حساب الطبيب. (uid)	String?	doctorUserId
قائمة بأيام العمل (مثال: ["Mon", "Tue"]):	List<String>	days
الفترات الزمنية المتاحة (مثال: ["09:00", "09:30"]):	List<String>	slots
تاريخ آخر تحديث للجدول.	Timestamp	updatedAt

جدول رقم (4-3)

5. مجموعة الحجوزات (appointments)

تخزن بيانات عمليات حجز المواعيد بين المرضى والأطباء.

الوصف/الملاحظات	النوع (Type)	الحقل (Field)
معرف المريض. (uid)	String	patientId
معرف ملف الطبيب.	String	doctorId
معرف حساب الطبيب المستهدف.	String?	doctorUserId
معرف القسم الطبي.	String	departmentId
تاريخ الموعد بصيغة. (YYYY-MM-DD)	String	date
وقت الموعد بصيغة. (HH:mm)	String	time
حالة الحجز. (confirmed/cancelled/completed)	String	status

الوصف/الملاحظات	النوع (Type)	الحقل (Field)
تاريخ تسجيل عملية الحجز.	Timestamp	createdAt

جدول رقم (3-5)

6. مجموعة السجلات الطبية (medicalRecords)

تحتوي على ملاحظات الطبيب حول زيارات المريض.

الوصف/الملاحظات	النوع (Type)	الحقل (Field)
معرف المريض.	String	patientId
معرف الطبيب الذي قام بتسجيل السجل.	String	doctorUserId
الملاحظات الطبية أو التوصيات.	String	note
تاريخ ووقت إنشاء السجل الطبي.	Timestamp	createdAt

جدول رقم (3-6)

3.8 مشكلة النظام الحالي:

تتمثل مشكلة النظام الحالي (التقليدي) في مستشفى جامعة الشيخ عبدالله البدر، والذي يعتمد على الحضور المباشر أو الاتصالات الهاتفية، في مجموعة من التحديات التنظيمية والتقنية التي تؤثر على جودة الخدمة الطبية.

فيما يلي تفصيل لأبرز مشكلات النظام الحالي:

- **بطء وصعوبة الإجراءات:** تعاني إجراءات الحجز والمتابعة من التعقيد والبطء نتيجة الاعتماد الكلي على العمل اليدوي.
- **طول زمن الانتظار:** يؤدي سوء التنظيم التقليدي إلى هدر وقت المرضى وصعوبة الوصول إلى الطبيب المناسب في الوقت المطلوب.
- **العبء الإداري والأخطاء البشرية:** يقع ضغط كبير على موظفي الاستقبال، مما يرفع من احتمالية وقوع أخطاء في تسجيل البيانات أو تضارب المواعيد وفقدان المعلومات.

- **تضارب المواعيد (الازدواجية):** (غياب التنسيق الرقمي يؤدي في كثير من الأحيان إلى حجز أكثر من مريض عند نفس الطبيب في الوقت ذاته، مما يسبب إرباكاً للعملية الصحية).
- **صعوبة تتبع السجل الطبي:** يفتقر المريض في النظام الحالي إلى وسيلة سهلة تمكنه من الاطلاع على تاريخه الطبي أو متابعة مواعيده السابقة والحالية بشكل مباشر.
- **الازدحام وسوء التنظيم:** تؤدي الطرق التقليدية إلى تكديس المراجعين داخل أروقة المستشفى، مما يؤثر سلباً على انسيابية العمل ورضا المستفيدين.
- **محدودية الوصول للمعلومات:** يصعب في النظام اليدوي الحصول على معلومات دقيقة ومحدثة حول تخصصات الأطباء أو جداول عملهم المتاحة على مدار الساعة.

3.9 وصف النظام المقترح:

يعتمد النظام المقترح على بناء تطبيق هاتف ذكي يعمل بنظام أندرويد لتنظيم وإدارة حجوزات مستشفى جامعة الشيخ عبدالله البدري الجامعي، بهدف أتمتة العمليات التقليدية وتقليل زمن الانتظار. يوظف التطبيق إطار **Flutter** لتطوير واجهات المستخدم، مع الاعتماد على خدمات **Firebase** مثل **Authentication** و **Cloud Firestore** كخلفية برمجية سحابية لإدارة البيانات والمصادقة.

تحدد هيكلية النظام بناءً على **صلاحيات** محددة تضمن أمان البيانات وانسيابية العمل:

- **صلاحيات المريض:**
 - إنشاء حساب شخصي والتحقق من الهوية للدخول إلى النظام.
 - استعراض الأقسام الطبية وقائمة الأطباء المتاحين وتصفيتهم حسب التخصص.
 - الاطلاع على الملف المهني للطبيب والمواعيد المتاحة في جدول الزماني.
 - حجز موعد جديد ومتابعة حالة الحجوزات السابقة والحالية الخاصة به فقط.
- **صلاحيات الطبيب:**
 - الوصول إلى لوحة تحكم خاصة تتيح له رؤية جدول عمله اليومي.
 - متابعة الحجوزات المرتبطة به ومعرفة بيانات المرضى المسجلين في تلك المواعيد.
 - إدارة الأوقات المتاحة وتحديث بيانات ملفه الشخصي والمهني ضمن حدود معينة.
- **صلاحيات المسؤول (Admin):**
 - التحكم الكامل في إدارة المستخدمين وتعديل مستويات الوصول في النظام.
 - إنشاء ملفات الأطباء الجدد وربطها بحسابات المستخدمين المقابلة لها.
 - إضافة وتعديل الأقسام الطبية وإدارة الجداول الزمنية العامة للأطباء.

المزايا التقنية والأمنية للنظام:

- **بوابة الصلاحيات (Role Gate):** يتضمن التطبيق آلية برمجية تقوم بقراءة قيمة الصلاحية المخزنة للمستخدم وتوجيهه تلقائياً إلى الواجهة البرمجية المناسبة له فور نجاح عملية تسجيل الدخول.
- **منع تعارض الحجوزات:** يطبق النظام آلية للتحقق من التوفر قبل تأكيد العملية، حيث يتم استخدام معرف ثابت للحجز مشتق من بيانات الطبيب والتاريخ والوقت لمنع تكرار الحجز لنفس الطبيب في نفس اللحظة.
- **قواعد أمن البيانات:** يتم تفعيل قواعد (Security Rules) على مستوى قاعدة البيانات تمنع أي مستخدم من الوصول إلى بيانات لا تخصه، بحيث لا يستطيع المريض الاطلاع على حجوزات غيره، ولا يحق لأي مستخدم غير المسؤول تعديل البيانات الحساسة مثل مستويات الصلاحية.
- **التحول الرقمي المؤسسي:** يساهم النظام في تقليل المعاملات الورقية والعبء الإداري على موظفي الاستقبال، مما يرفع من كفاءة إدارة الموارد الطبية والبشرية داخل المستشفى.

3.10 تحليل مدخلات النظام (Inputs Analysis)

تتمثل المدخلات في كافة البيانات التي يتم إدخالها من قبل المستخدمين (مريض، طبيب، مسؤول) لتشغيل وظائف التطبيق المختلفة.

الوصف	البيانات المدخلة	جهة الإدخال
لإنشاء حساب جديد والتحقق من الهوية.	بيانات التسجيل (الاسم، البريد، كلمة المرور)	المريض
لتحديد تفاصيل الموعد المطلوب.	بيانات الحجز (التاريخ، الوقت)	
لتحديث الملف الشخصي للمريض.	البيانات الشخصية (الهاتف، تاريخ الميلاد)	
لتحديد الأوقات المتاحة للحجز من قبل المرضى.	الجدول الزمني (أيام العمل، فترات المواعيد)	الطبيب
لتحديث بيانات الطبيب التي تظهر للمرضى.	الملف المهني (التخصص، السيرة الذاتية)	
لتسجيل تفاصيل الزيارة في السجل الطبي للمريض.	الملاحظات الطبية	
لإضافة أو تعديل أسماء الأقسام داخل المستشفى.	الأقسام الطبية	المسؤول
لإنشاء ملفات الأطباء وربطها بحساباتهم.	بيانات الأطباء الجدد	
لتعديل صلاحيات الوصول لكل مستخدم (مريض/طبيب/مسؤول).	مستويات الصلاحية (Roles)	

جدول رقم (3-7)

3.11 تحليل مخرجات النظام (Outputs Analysis)

المخرجات هي المعلومات والنتائج التي ينتجها النظام بعد معالجة المدخلات، وتظهر في شكل شاشات، قوائم، أو رسائل تأكيد.

- **واجهات المستخدم المخصصة:** يتم عرض واجهة برمجية مختلفة لكل مستخدم بناءً على صلاحياته فور تسجيل الدخول.
- **مخرجات المريض:**
 - قوائم الأقسام الطبية والأطباء المتاحين للاستعراض.
 - عرض المواعيد المتاحة (Slots) بناءً على جدول الطبيب المختبأ في قاعدة البيانات.
 - قائمة الحجوزات الشخصية وتاريخها لمتابعة حالة المواعيد.
 - السجل الطبي المبسط الذي يحتوي على ملاحظات الطبيب.
- **مخرجات الطبيب:**
 - عرض الجدول الزمني الشخصي وأوقات العمل المحددة.
 - قائمة الحجوزات المؤكدة المرتبطة به مع تفاصيل المريض.
- **مخرجات المسؤول:**
 - لوحة تحكم (Dashboard) تعرض إحصائيات سريعة عن عدد المستخدمين، الأطباء، الأقسام، والحجوزات.
 - قوائم شاملة لإدارة كافة المستخدمين وملفات الأطباء والأقسام داخل النظام.
- **رسائل النظام:** تنبيهات فورية تؤكد نجاح عملية الحجز أو تمنع التعارض في حال وجود موعد سابق في نفس اللحظة.

UML3.12مخططات لتطبيق طبيبك

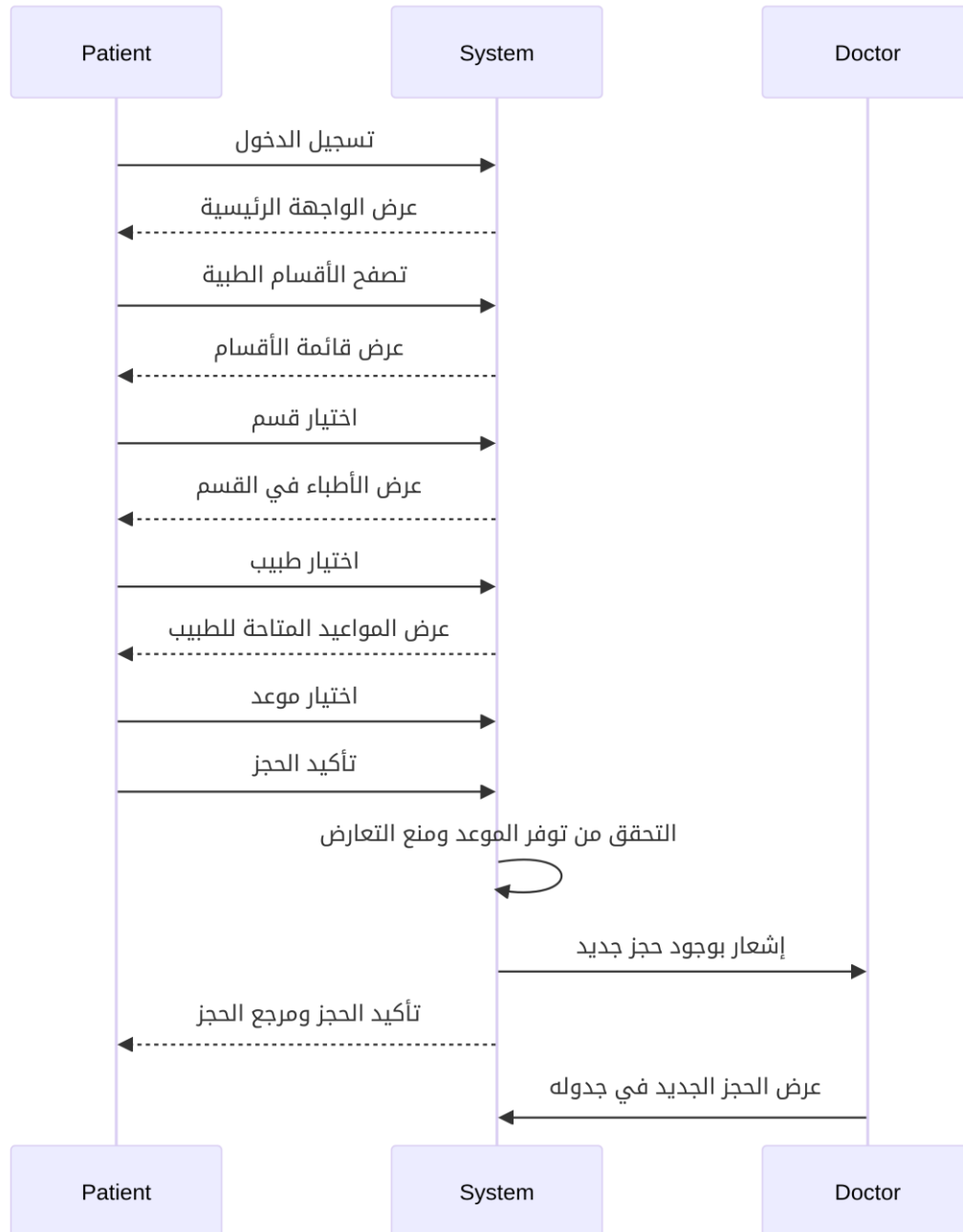
3.13 (Use Case Diagram) مخطط حالة الاستخدام

للنظام المختلفة والوظائف (المسؤول الطبيب، المريض)، المستخدمين بين التفاعلات المخطط هذا يوضح



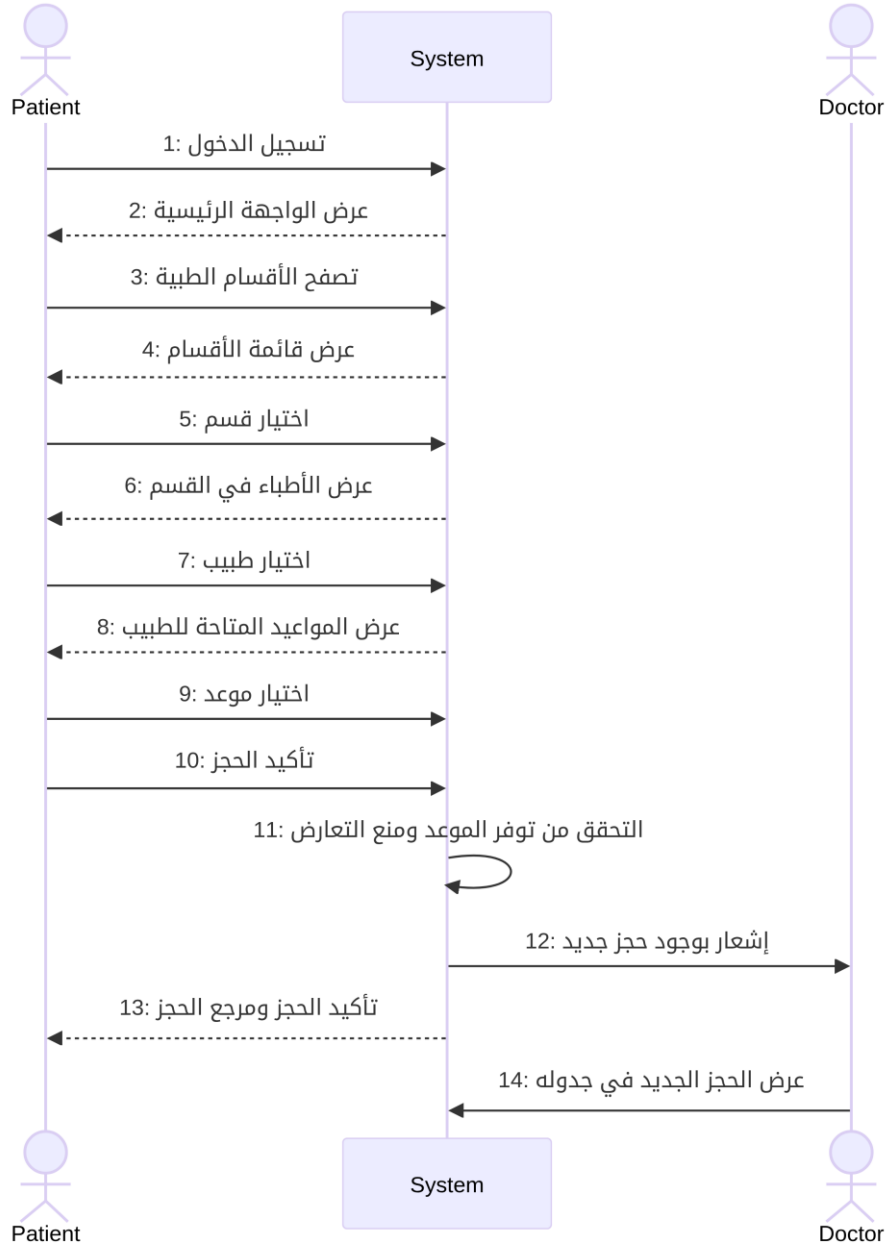
3.14 (Sequence Diagram) مخطط التتابع

موعد حجز عملية في الكائنات بين والتفاعلات العمليات تسلسل المخطط هذا يوضح



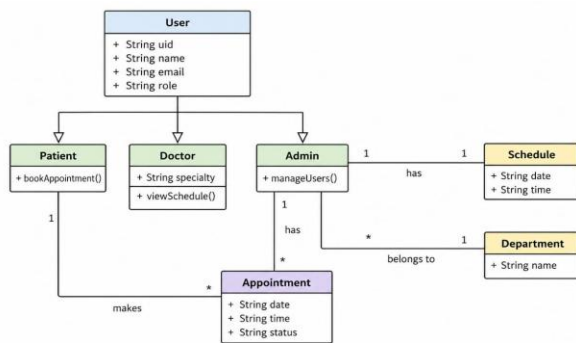
3.15 (Interaction Diagram) التفاعل مخطط

الموعد حجز مهمة لإنجاز البعض بعضها مع الكائنات تواصل كيفية المخطط هذا يوضح



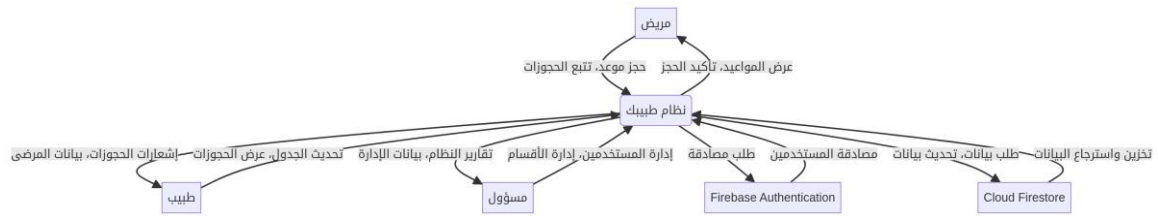
3.16 (Class Diagram) مخطط الاصناف

البعض ببعضها وعلاقتها وخصائصها (Classes) الأصناف عرض خلال من النظام هيكل المخطط هذا يوضح

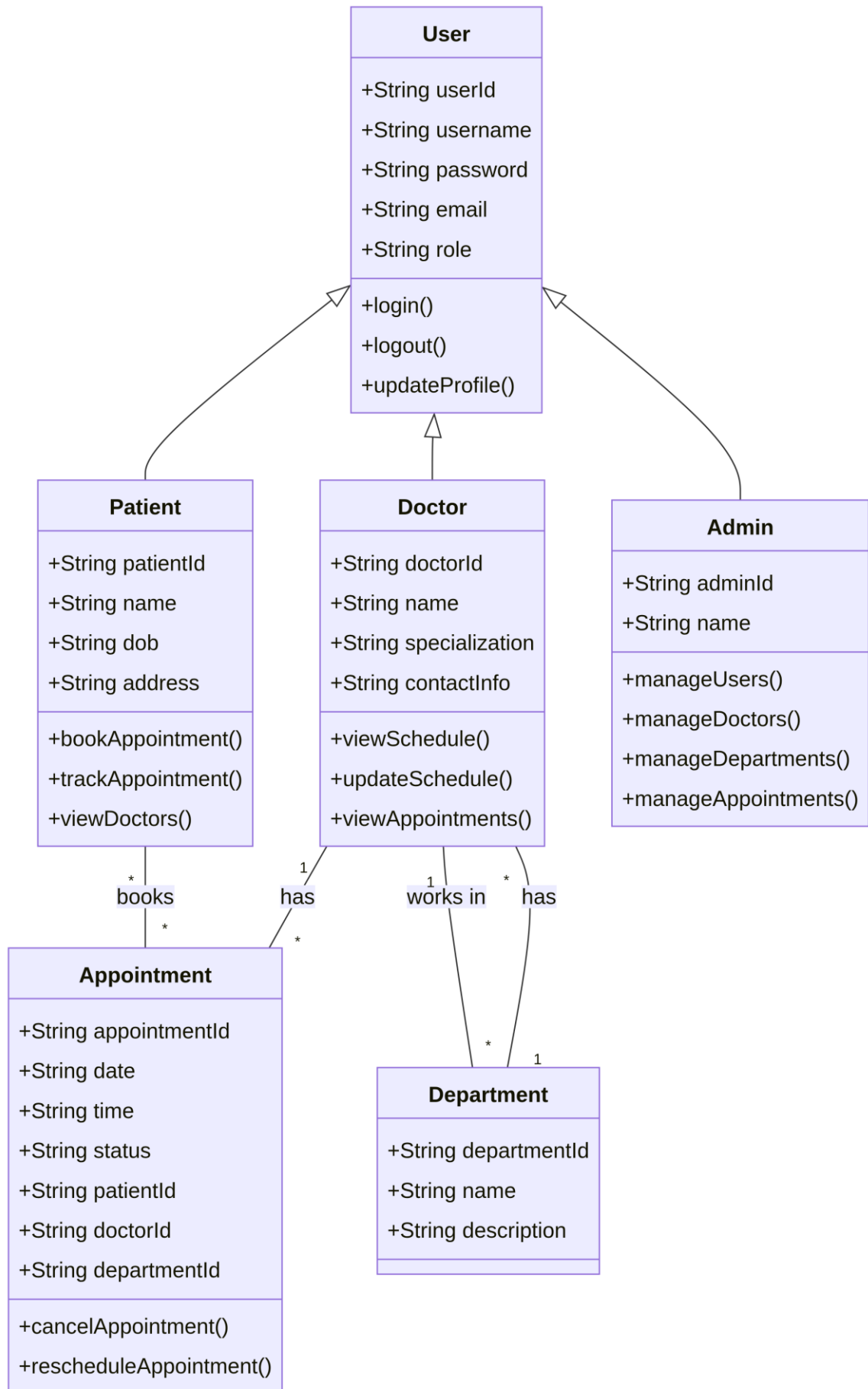


3.17 (Context Diagram) النظام سياق مخطط

(Firebase مثل الخارجية والأنظمة المستخدمين) الخارجية الكيانات مع وتفاعله النظام حدود المخطط هذا يوضح



3.18 مخطط الكينونة والعلاقات (ERD)



الفصل الرابع:

التصميم

4.1 تصميم معمارية التطبيق (Architecture)

تم اعتماد أسلوب **Feature-based Architecture** لتقسيم البحث حسب المزايا (Admin/Doctor/Patient/Auth/Gate) لتسهيل التطوير والصيانة.

4.2 التصميم التقني (Technology Stack)

- Flutter (واجهة)
- الخاتمة
-
- تُعد الرقابة الداخلية حجر الأساس في تحقيق الانضباط المالي والإداري داخل أي مؤسسة، حيث تسهم بشكل مباشر في حماية الأصول، والتأكد من دقة البيانات المالية، والالتزام بالسياسات والإجراءات المعتمدة.
-
- وقد أظهرت الدراسة أن وجود نظام رقابة داخلية فعال يؤدي إلى تحسين جودة التقارير المالية، وتقليل حالات الغش والأخطاء، ورفع كفاءة استخدام الموارد. هذه العوامل مجتمعة تنعكس إيجاباً على مؤشرات الأداء المالي مثل الربحية، السيولة، والعائد على الاستثمار.
-
- كما اتضح أن ضعف الرقابة الداخلية أو غيابها يفتح المجال أمام التلاعب والمخاطر المالية، مما يؤثر سلباً على سمعة المنشأة وقدرتها التنافسية. لذلك فإن الاستثمار في تطوير أنظمة الرقابة الداخلية وتدريب الكوادر عليها لم يعد خياراً، بل ضرورة حتمية لضمان الاستدامة والنمو.
-
- وفي الختام، توصي الدراسة بضرورة التقييم الدوري لنظم الرقابة الداخلية، ومواكبتها للتغيرات التقنية والقانونية، لضمان استمرار أثرها الإيجابي على الأداء المالي للمؤسسة. **Firebase**
- Authentication (تسجيل الدخول)
- Cloud Firestore (قاعدة بيانات)
- Riverpod (إدارة الحالة)
- Google Fonts + Material 3 (واجهات)

4.3 تصميم قاعدة البيانات (Firestore Data Design)

المجموعات Collections

1. users/{uid}

يحفظ ملف المستخدم + الدور:

- displayName, email, emailLower, role, createdAt, isActive

2. departments/{deptId}

الأقسام:

- name, isActive

3. doctors/{doctorId}

ملف الطبيب (ينشئه المسؤول ويربطه بالمستخدم):

- name, title, departmentId, email/emailLower, userId, bio, isActive

4. schedules/{doctorId}

جدول الطبيب:

- doctorUserId, days[], slots[], updatedAt

5. appointments/{appointmentId}

الحجوزات:

- patientId, doctorId, doctorUserId, departmentId, dateKey(or date), time, status, createdAt

6. medicalRecords/{id}

- patientId, doctorUserId, note, createdAt

4.4 قواعد الصلاحيات (Firestore Security Rules)

4.4.1 فكرة التصميم الأمني

- كل العمليات تعتمد على role داخل users/{uid}
- المسؤول: صلاحية كاملة (قراءة/كتابة)

- الطبيب: يقرأ بياناته/حجوزاته فقط، ويعدل فقط حقوله المسموح بها
- المريض: يحجز ويقرأ حجوزاته فقط

4.4.2 أهم منطق بالصلاحيات

- RoleGate يوجه المستخدم حسب `users/{uid}.role`
- في: Firestore Rules
 - حجوزات المريض `: resource.data.patientId == request.auth.uid`
 - حجوزات الطبيب `: resource.data.doctorUserId == request.auth.uid`
 - تعديل ملف الطبيب: فقط للطبيب المالك للـ `userId`

نقطة حساسة:

هل الحجز يخزن التاريخ في `date` أم `dateKey`؟

لازم تختار واحد وتلتزم به في الـ `Repo + Rules + Models + Queries + Indexes`.

4.4.3 مثل Rules

إذا قررت تستخدم `(dateKey)`

عدّل جزء `appointments` إلى:

```
match /appointments/{id} {
  allow create: ifsignedIn()
  &&request.resource.data.patientId == request.auth.uid
  &&request.resource.data.doctorId is string
  &&request.resource.data.dateKey is string
  &&request.resource.data.time is string;
  allow read: ifisAdmin()
    || (signedIn()&&resource.data.patientId == request.auth.uid)
    || (isDoctor()&&resource.data.doctorUserId == request.auth.uid);
  allow update: ifisAdmin()
    || (signedIn()&&resource.data.patientId == request.auth.uid)
    || (isDoctor()&&resource.data.doctorUserId == request.auth.uid);
  allow delete: ifisAdmin();
```

4.6 واجهات الاندخال

الرئيسية	Admin
لوحة التحكم User:2 Department:0 Doctor:0 Appotintment:0	
أدوات سريعة	
بيانات تجريبية seed انشاء+اقسام+اطباء+جداول بسيطة للتجربة بسرعة	
ملاحظة حول الأمان من الملف Firstore Rules بعد اختبار التطبيق firstore rules فعل	
حسابي	المستخدمين
الاطباء	الاقسام
الرئيسية	
واجهة المدير الرئيسية	

Admin					الأطباء												
إضافة طبيب																	
ahmed@gmail.com																	
uid:ENPckACVAmOTDD88YbP3GIBTB3																	
Doctor ID:																	
Doc_ ENPckA																	
Document ID داخل doctor سيكون هو																	
اسم الطبيب																	
الدرجة\الوظيفية																	
Specialist																	
القسم																	
باطنية																	
<table border="1"> <tr> <td colspan="2">حفظ</td> <td colspan="2">الغاء</td> </tr> <tr> <td>حسابي</td> <td>المستخدمين</td> <td>الاطباء</td> <td>الاقسام</td> </tr> <tr> <td colspan="2">الرئيسية</td> <td colspan="2"></td> </tr> </table>					حفظ		الغاء		حسابي	المستخدمين	الاطباء	الاقسام	الرئيسية				
حفظ		الغاء															
حسابي	المستخدمين	الاطباء	الاقسام														
الرئيسية																	
4.7 واجهة إضافة طبيب																	

Admin					الاقسام
إضافة قسم					
ID (مثال:cardiology)					
Cardiology(مثال)الاسم					
حفظ		الغاء			
حسابي	المستخدمين	الاطباء	الاقسام	الرئيسية	
4.8واجهة إضافة الاقسام					

تصميم شاشات التطبيق الطبي 4.9

4.10 واجهة الأقسام

10:58 AM Signal Battery
[بحث عن قسم]
 لا توجد أقسام بعد
حسابي السجل مواعيدي الأقسام الرئيسية

4.11 واجهة المستخدم الرئيسية

10:58 AM Signal Battery
👋 مرحباً كيف يمكننا مساعدتك؟ [ابحث عن طبيب] الأقسام: لا توجد أقسام بعد
حسابي السجل مواعيدي الأقسام الرئيسية

4.12 واجهة المواعيد

10:58 AM Signal Battery
 لا توجد مواعيد بعد
حسابي السجل مواعيدي الأقسام الرئيسية

4.13 واجهة سجلات المريض

10:58 AM Signal Battery
<div>  <p>لا توجد سجلات طبية</p> </div>
حسابي السجل مواعيدي الأقسام الرئيسية

4.14 واجهة حساب المستخدم

10:58 AM Signal Battery
<p>البريد: user@example.com</p> <p>الهاتف: 0123456789</p> <p>ذكر: الجنس</p> <p>[تسجيل الخروج]</p>
حسابي السجل مواعيدي الأقسام الرئيسية

4.15 تعديل اعدادات المستخدم

10:58 AM Signal Battery
<p>تعديل البيانات</p> <p>[0123456789]</p> <p>1 2 3</p> <p>4 5 6</p> <p>7 8 9</p> <p>* 0 <X></p>
حسابي السجل مواعيدي الأقسام الرئيسية

الفصل الخامس

التنفيذ

6.1 التنفيذ (Implementation)

وصف ما يحتويه الفصل

يستعرض هذا الفصل الجانب التطبيقي الفعلي للمشروع، موضحاً كيفية تهيئة وربط خدمات Firebase داخل إطار عمل Flutter. كما يشرح الآلية البرمجية لـ "بوابة الصلاحيات" المسؤولة عن توجيه المستخدمين تلقائياً، ويقدم عرضاً توثيقياً للواجهات النهائية التي تم تنفيذها لكل فئة من مستخدمي النظام (المسؤول، الطبيب، والمريض)، بالإضافة إلى واجهات تسجيل الحسابات.

5.1.1 تشغيل Firebase داخل Flutter

في `main.dart` يتم تهيئة Firebase قبل تشغيل التطبيق:

```
Future<void>main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  runApp(const ProviderScope(child: HospitalApp()));  
}
```

5.1.2 بوابة الصلاحيات Role Gate

الهدف: بعد تسجيل الدخول يقرأ `users/{uid}` ثم يفتح واجهة:

- AdminShell أو DoctorShell أو PatientShell

من `features/gate/role_gate.dart`:

```
final authState = ref.watch(authStateProvider);
```

```
if (user == null) return const AuthPage();
```

```
final me = ref.watch(currentAppUserProvider);
```

```
return switch (appUser.role) {
  'admin' => const AdminShell(),
  'doctor' => const DoctorShell(),
  _ => const PatientShell(),
};
```

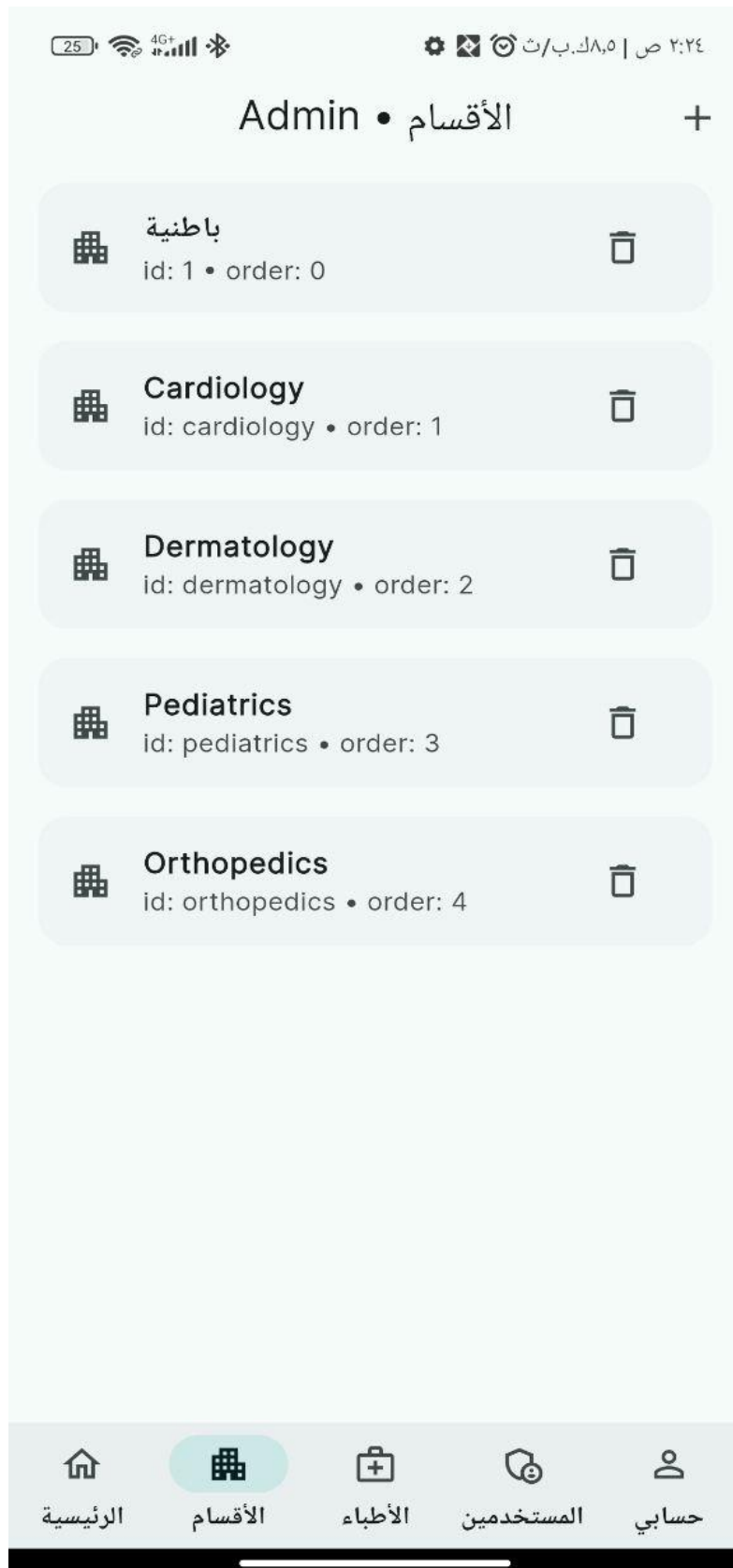
الفائدة: يمنع دخول أي واجهة بدون دور واضح.

5.2 الواجهات:

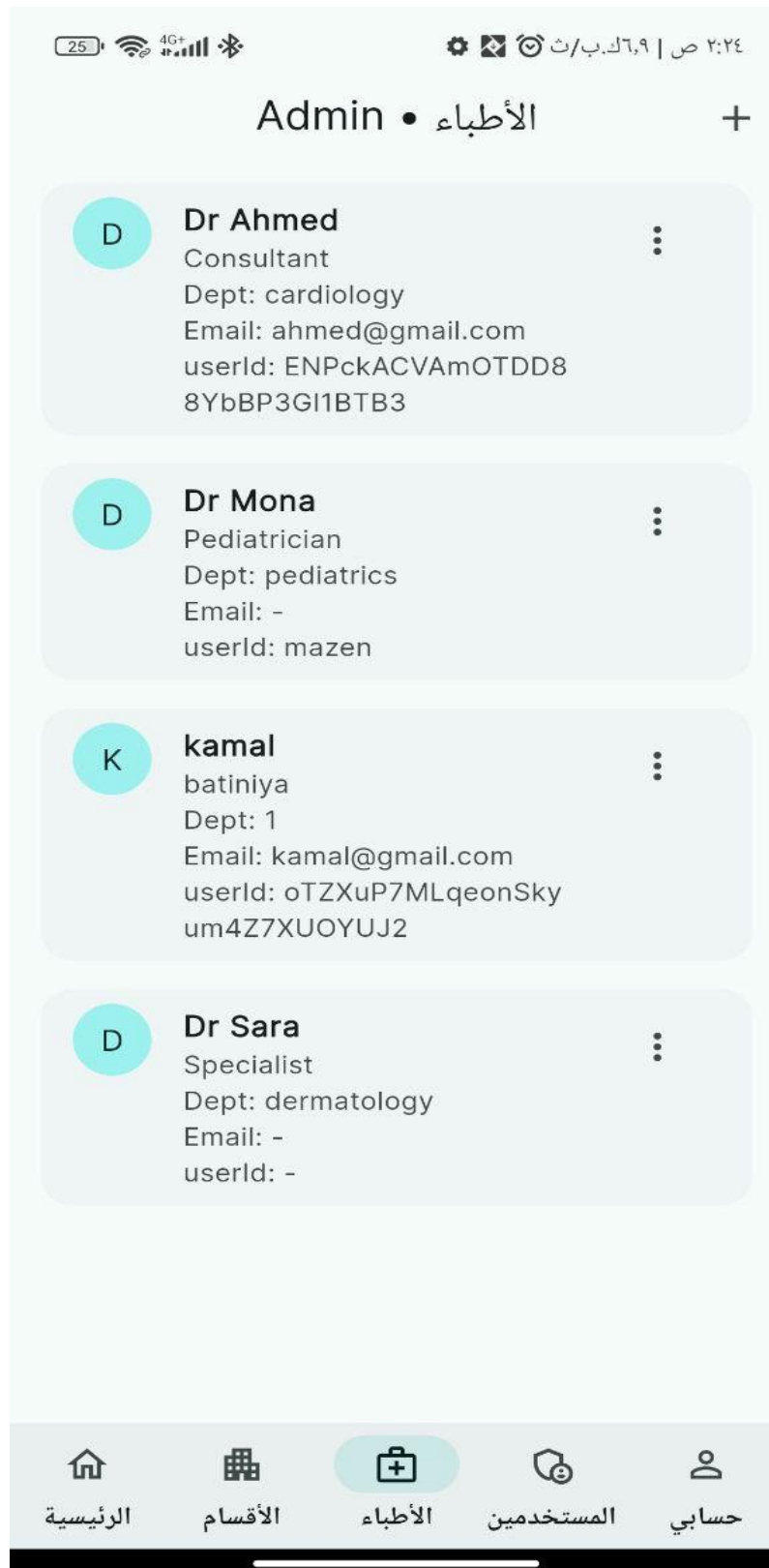
5.2.1 واجهات المسؤول



شكل رقم (5-1) واجهة المسؤول الرئيسية



شكل رقم (5-2) واجهة أقسام المسؤول



شكل رقم (3-5) واجهة الأطباء المسؤول

Admin • المستخدمين

🔍 UID... أو Email ابحث بالـ



ahmed@gmail.com

uid: ENPckACVAmOTD

D88YbBP3GI1BTB3

role: doctor



emad@gmail.com

uid: bqFpDRQxasR9DX

4bhwzzOt452q2

role: patient



الرئيسية



الأقسام



الأطباء



المستخدمين



حسابي

شكل رقم (5-4) واجهة المستخدمين لدى المسؤول

5.2.2 واجهات الطبيب



شكل رقم (5-5) واجهة الطبيب الرئيسية

25%

4G+

⚙️

🕒

٣٦,٠ ك.ب/ث

٢:٢٥ ص

لوحة الطبيب

أيام العمل

السبت

✓ الأحد

✓ الإثنين

✓ الثلاثاء

✓ الأربعاء

الخميس

الجمعة

الأوقات (Slots)

+

09:00 ×

09:30 ×

10:00 ×

10:25 ×

10:30 ×

11:00 ×

حفظ

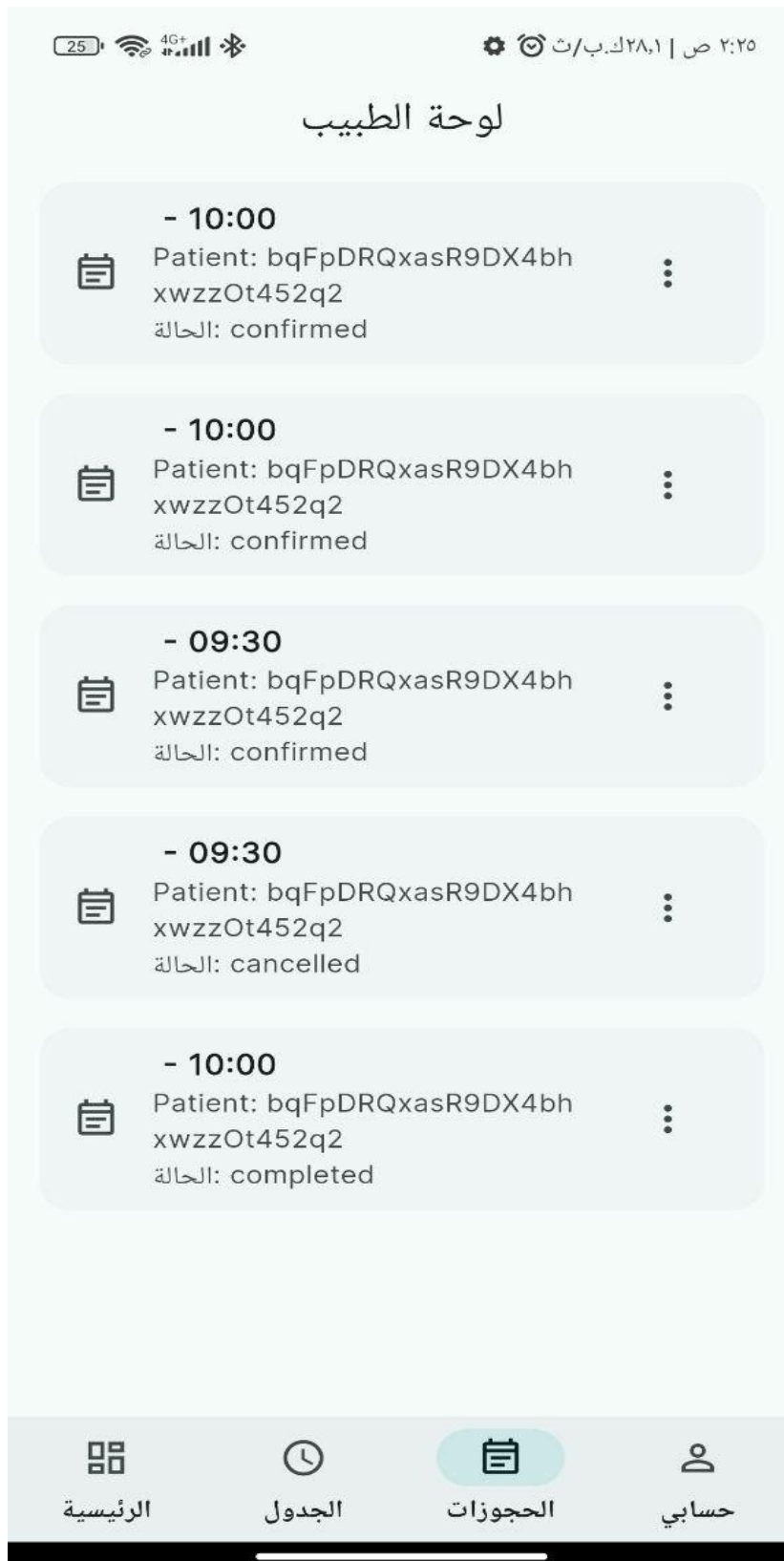
الرئيسية

الجدول

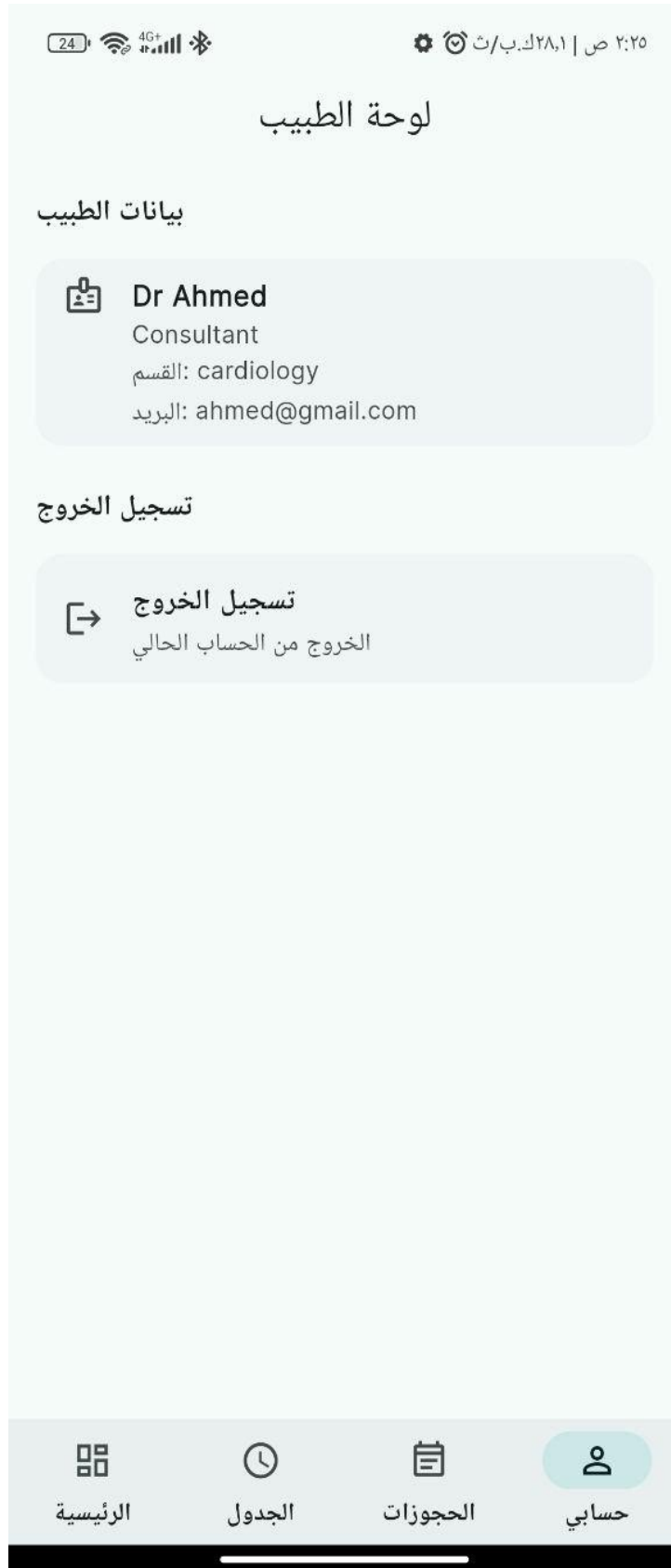
الحجوزات

حسابي

شكل رقم (5-6) واجهة جدول الطبيب

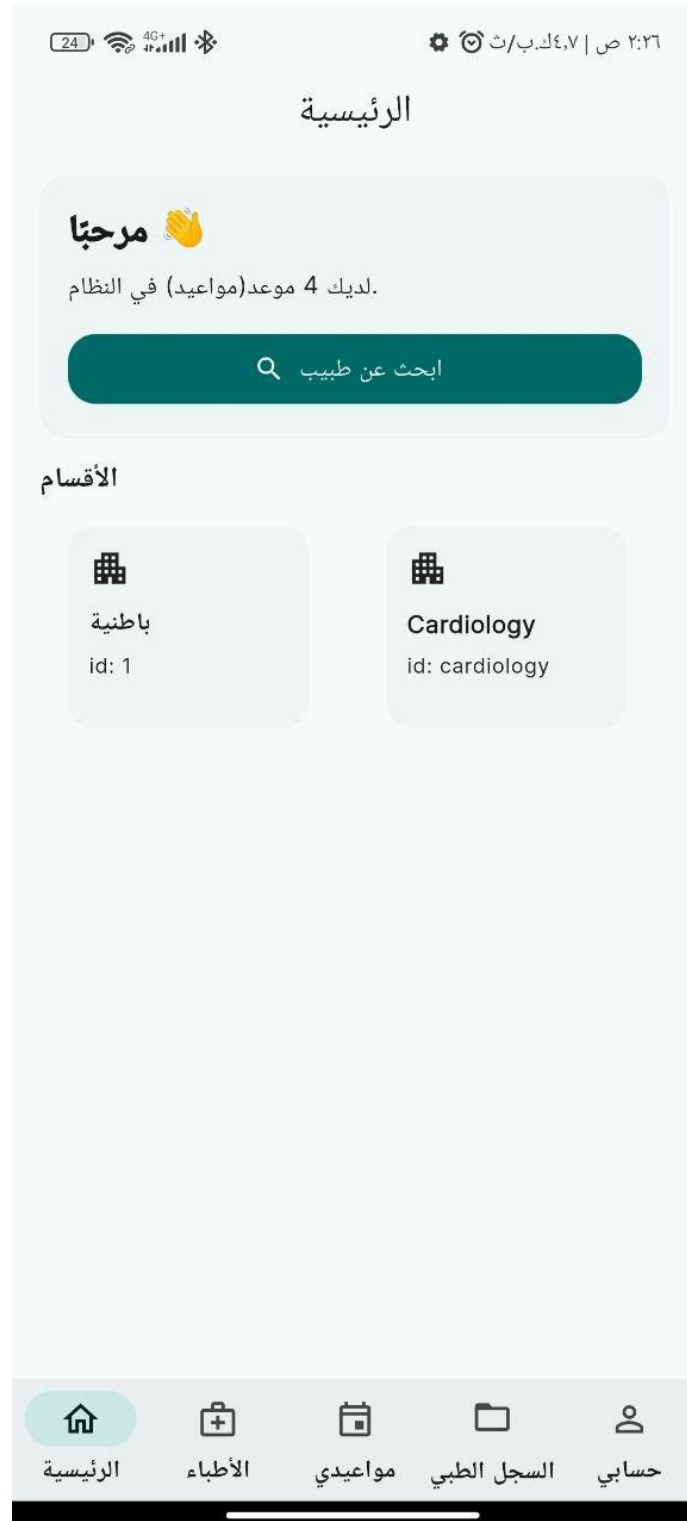


شكل رقم (5-7) واجهة حجوزات الطبيب

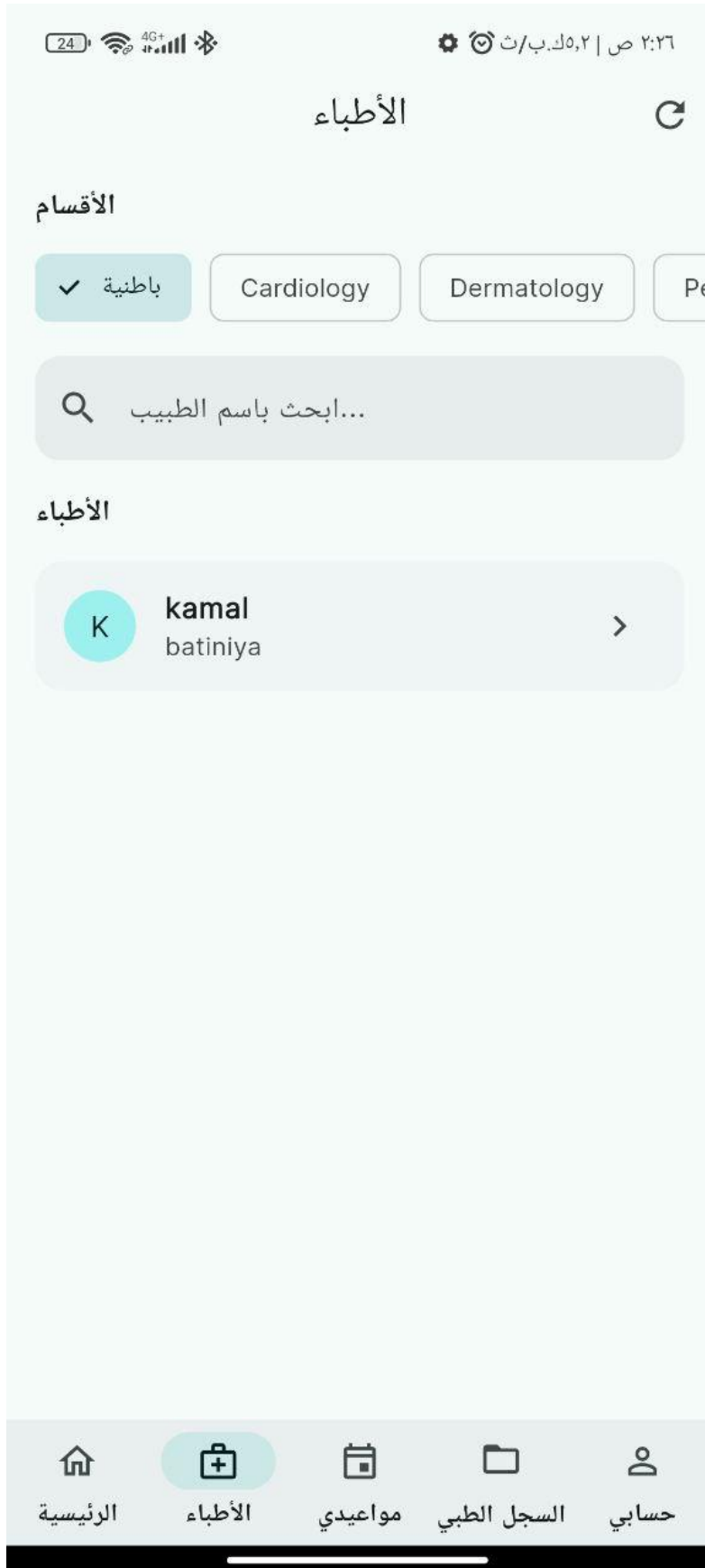


شكل رقم (5-8) واجهة حساب الطبيب

5.2.3 واجهات المستخدم



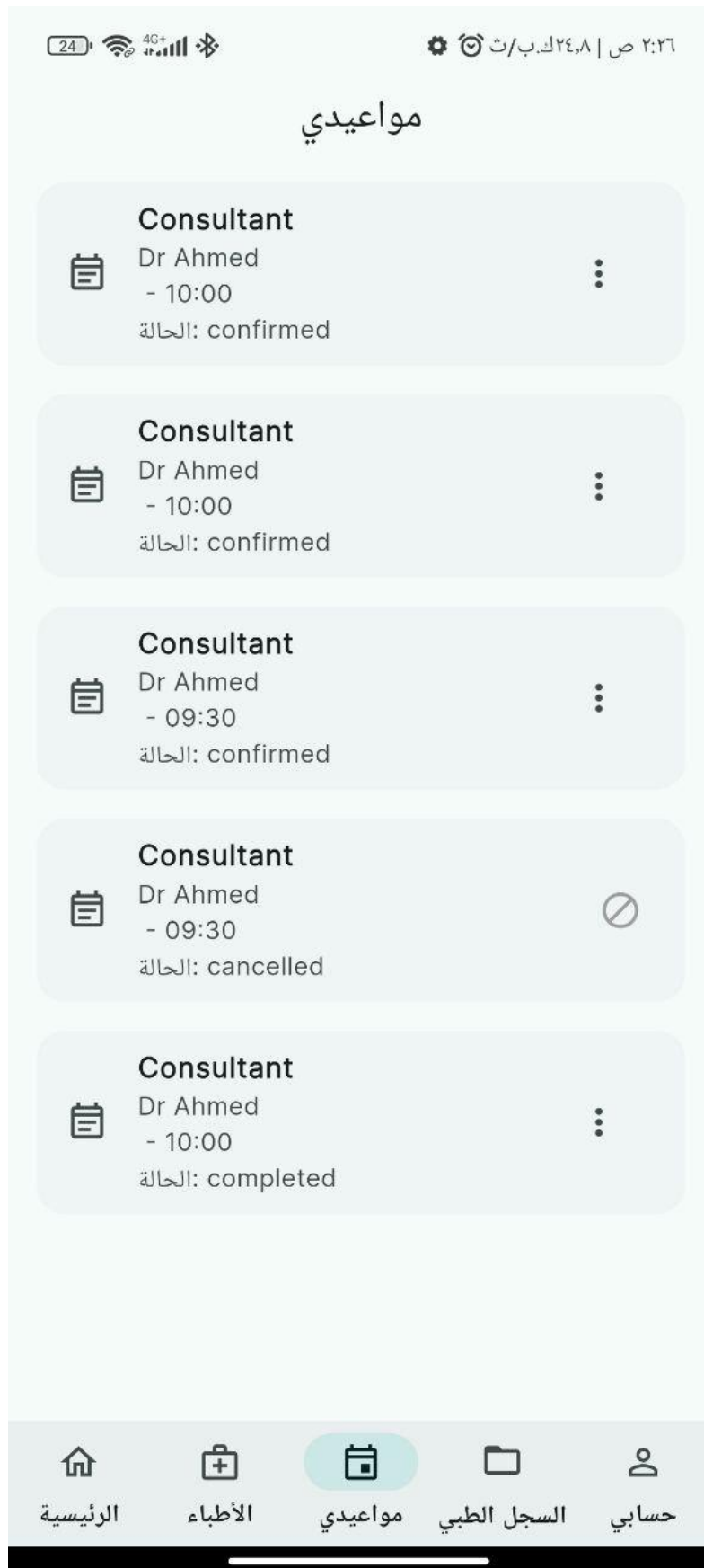
شكل رقم (5-9) واجهة المستخدم الرئيسية



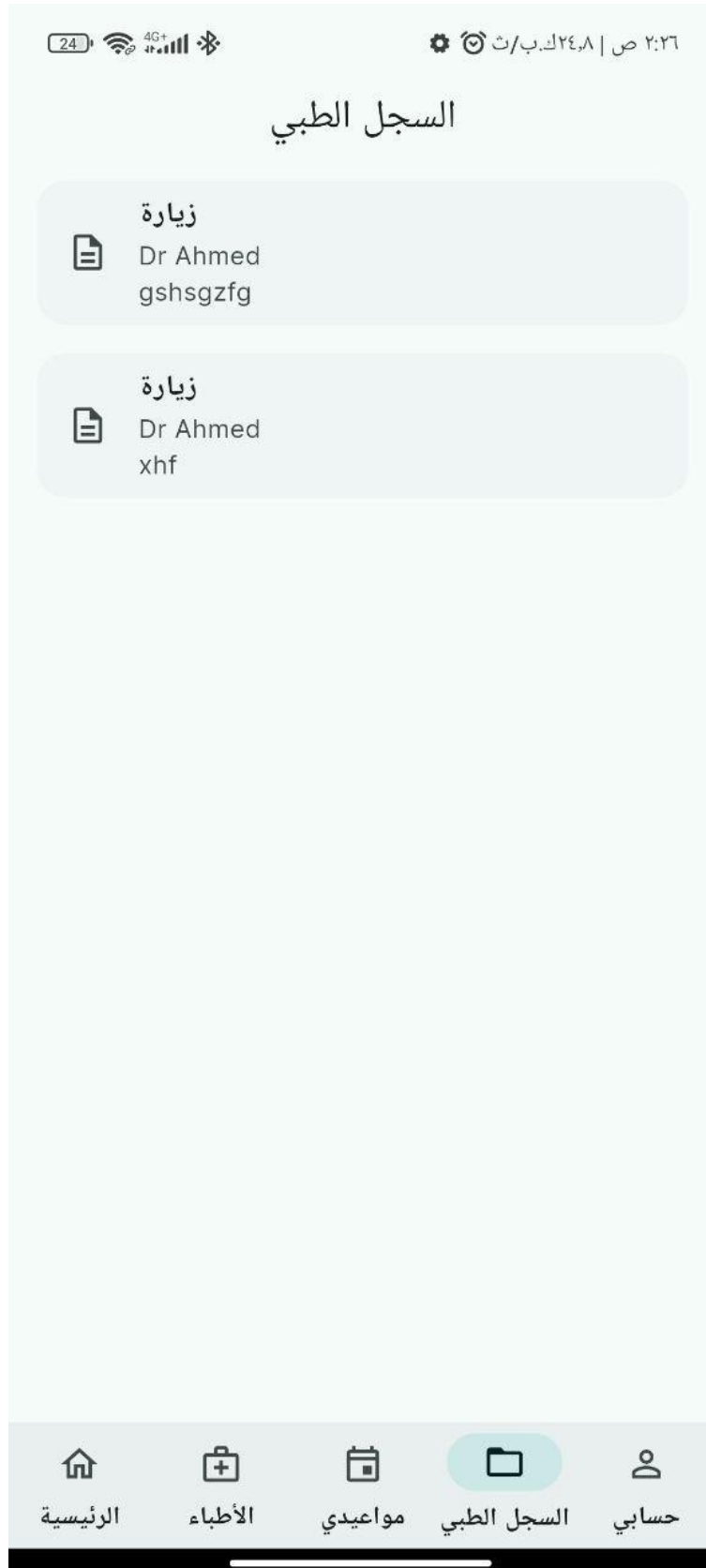
شكل رقم (10-5) واجهة الاطباء والاقسام



شكل رقم (5-11) واجهة عرض الطبيب والحجز



شكل رقم (12- 5) واجهة الحجوزات



شكل رقم (5-13) واجهة السجل الطبي

24 4G+

٢:٢٦ ص | ٩,٤ ك.ب/ث

حسابي

البيانات

Email: emad@gmail.com

الاسم

 emad

رقم الهاتف



النوع

 ▼

تاريخ الميلاد (YYYY-MM-DD)

 11-12-2002

حفظ



تسجيل الخروج

 Logout

الخروج من الحساب الحالي

 الرئيسية

 الأطباء

 مواعيدي

 السجل الطبي

 حسابي

شكل رقم (5-14) واجهة الحساب

5.2.4 واجهة تسجيل الدخول

The screenshot shows a mobile application interface for login. At the top, there is a status bar with icons for battery (26%), 4G+ signal, and a clock showing ٢:٢٤ ص | ١١,١.١. Below the status bar, the app's logo is a green square with a white cross. The text 'elsheikh abdalla albadry hospital' is displayed in a bold, black font. Below the name, the text 'تسجيل الدخول' (Login) is shown. There are two input fields: the first is labeled 'البريد الإلكتروني' (Email) with an '@' icon, and the second is labeled 'كلمة المرور' (Password) with a lock icon. A large green button with the text 'دخول' (Login) is positioned below the input fields. Below the button, the text 'إنشاء حساب' (Create Account) is visible. At the bottom, the text 'تطبيق مستشفى الشيخ عبدالله البدري الجامعي' (University of Sheikh Abdulla Albadry Hospital Application) is displayed.

شكل رقم (5-15) واجهة تسجيل الدخول



elsheikh abdalla albadry hospital

إنشاء حساب جديد

الاسم

البريد الإلكتروني

كلمة المرور

تسجيل

عندي حساب بالفعل

تطبيق مستشفى الشيخ عبدالله البدري الجامعي

شكل رقم (5-16) واجهة التسجيل الجديد

الفصل السادس:

النتائج والتوصيات

6.2 النتائج (Results)

6.2.1 تحقيق أهداف التطبيق

بعد تطوير التطبيق وتشغيله، تم تحقيق الأهداف الأساسية التالية:

1. تطبيق واحد بثلاث واجهات (مريض/طبيب/مسؤول)
تم اعتماد بوابة صلاحيات (Role Gate) تقوم بتحويل المستخدم تلقائيًا لواجهة مناسبة حسب قيمة role في مستند users/{uid} داخل Firestore.
2. إدارة المستخدمين والصلاحيات بواسطة المسؤول
 - المسؤول يستطيع تغيير الدور (patient/doctor/admin) حسب صلاحيات النظام.
3. إدارة الأطباء وربط الطبيب بحساب مستخدم
 - تم تنفيذ آلية ربط الطبيب بالمستخدم عبر doctors/{doctorId}.userId = uid.
 - بعد الربط يصبح الطبيب قادرًا على الدخول إلى لوحة الطبيب، ورؤية بياناته ومواعيده.
4. إدارة الأقسام الطبية
 - تم تطوير مجموعة departments لعرض الأقسام للمرضى وربط الطبيب بالقسم عبر departmentId.
5. إدارة جدول الطبيب (Schedules)
 - تم تخزين جدول الطبيب في schedules/{doctorId} ليشمل:
 - أيام العمل (days)
 - الفترات/الأوقات المتاحة (slots)
6. تنفيذ الحجز ومنع التعارض
 - تم تنفيذ الحجز في مجموعة appointments بحيث يمنع التعارض (Double booking) لنفس الطبيب في نفس التاريخ والوقت.
 - تم استخدام معرف ثابت للحجز (documentId) مشتق من doctorId + date + time لمنع تكرار نفس الموعد عمليًا.

7. صلاحيات (Firestore (Security Rules

تم تطبيق قواعد تمنع:

- المريض من قراءة حجوزات غيره
- الطبيب من قراءة بيانات لا تخصه
- أي مستخدم غير المسؤول من تعديل بيانات حساسة (roles / user management)

8. إضافة نظام إشعارات

- إشعار للمريض عند تأكيد الحجز/تغييره
 - إشعار للطبيب عند وجود حجز جديد
- يمكن تنفيذ ذلك عبر (Firebase Cloud Messaging (FCM.

9. إضافة تقويم لاختيار التاريخ بدل إدخاله نصيًا.

6.2.2 مخرجات التشغيل

عند تشغيل التطبيق تم التأكد من:

- نجاح تسجيل الدخول والخروج.
- انتقال المستخدم تلقائيًا لواجهة دوره.
- عرض قائمة الأطباء والأقسام للمريض.
- نجاح عملية حجز موعد للمريض.
- ظهور الحجز في صفحة حجوزات المريض.
- ظهور الحجز في صفحة حجوزات الطبيب باستخدام (doctorUserId) .
- إمكانية المسؤول تحديث بيانات الأطباء وربطهم بالمستخدمين.

6.2.3 نقاط قوة التطبيق

- أمان واضح: قواعد Firestore تمنع الوصول غير المصرح به.
- تقسيم معماري جيد: فصل (Models/Repos/Features) يسهل الصيانة.
- قابلية التوسع: يمكن إضافة ميزات دون تغيير جذري في الهيكل.

6.3 التوصيات (Recommendations)

6.3.1 توصيات تطوير تقنية

1. استخدام Cloud Functions للتحقق من الحجوزات
بدل الاعتماد على documentId فقط، يمكن إضافة Cloud Function تقوم بالتحقق والتسجيل وإرسال إشعارات، خصوصًا عند زيادة عدد المستخدمين.
- تحسين الأداء : إضافة Pagination لقوائم الأطباء والحجوزات عند زيادة البيانات.
2. إضافة مؤشرات تحميل (Loading States) في كل العمليات الشبكية.

6.3.2 توصيات إدارية وتشغيلية

1. النسخ الاحتياطي
- جدول نسخ احتياطي دوري من Firestore (Export) خاصة عند استخدام النظام فعليًا.

6.4 المراجع (References)

6.4.1 مراجع تقنية رسمية (Official Technical References)

- Flutter Documentation — Official Docs. .1
- Dart Language Documentation — Official Docs. .2
- Firebase Documentation — Overview. .3
- Firebase Authentication Documentation. .4
- Cloud Firestore Documentation. .5
- Cloud Firestore Security Rules Documentation. .6
- FlutterFire Documentation (Firebase for Flutter). .7
- Riverpod Documentation (State Management for Flutter). .8

6.4.2 كتب ومراجع أكاديمية (Academic References)

- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson. .1
- Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill. .2
- Fowler, M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Addison-Wesley. .3
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill. .4
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. .5

6.4.3 دراسات وأوراق/مقالات مرجعية

- Google Cloud — Best practices for Firestore data modeling and queries (Documentation/Guides). .1
- Material Design 3 Guidelines — Google (UI/UX Guidelines). .2

6.5 الخاتمة

تم في هذا البحث تحليل وتصميم وتنفيذ تطبيق طبيبك كتطبيق واحد يخدم ثلاث فئات رئيسية: المريض، الطبيب، والمسؤول، مع الاعتماد على Flutter في تطوير الواجهات و Firebase كخدمة خلفية + Authentication (Cloud Firestore لتوفير إدارة المستخدمين والبيانات في بيئة سحابية).

ركز التطبيق على تلبية المتطلبات الأساسية للمريض مثل استعراض الأقسام والأطباء، عرض التفاصيل، وحجز المواعيد، إضافة إلى تمكين الطبيب من إدارة جدولته ومتابعة حجوزاته وتحديث ملفه الشخصي ضمن صلاحيات محددة. كما أنجزت واجهة المسؤول لإدارة المستخدمين وتعديل الصلاحيات وتطوير ملفات الأطباء وربطها بحساباتهم وإدارة الأقسام والجدول.

وأظهرت نتائج التنفيذ أن التطبيق يحقق وظائفه الأساسية بكفاءة مع تطبيق قواعد صلاحيات لحماية البيانات ومنع الوصول غير المصرح به، إضافة إلى آلية تمنع تكرار الحجز لنفس الطبيب في نفس الوقت.