الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشهيد حمه لخضر - الوادي
كلية العلوم الدقيقة
قسم الإعلام الآلي

الميدان: رياضيات وإعلام آلي
الشعبة: إعلام آلي
التخصص: نظم معلوماتية

مذكرة نهاية تخرج ضمن متطلبات الحصول على شهادة ليسانس أكاديمية

الموضوع

Smart Expense Tracking with OCR

إعداد الطلبة:
شمسة محمد العيد
بن عتوس نور الدين
صهيب غزال

تحت إشراف:
الأستاذ: بقاص منير

السنة الجامعية: 2024/2025

شكر وتقدير

قال تعالى:

﴿رَبِّ أَوْزِعْنِي أَنْ أَشْكُرَ نِعْمَتَكَ الَّتِي أَنْعَمْتَ عَلَيَّ وَعَلَى وَالِدَيَّ وَأَنْ أَعْمَلَ صَالِحًا تَرْضَاهُ وَأَدْخِلْنِي بِرَحْمَتِكَ فِي عِبَادِكَ الصَّالِحِينَ﴾

سورة النمل، الآية 19

في البداية، نشكر الله عز وجل الذي وفقنا لإتمام هذا العمل المتواضع، كما نتوجه بالشكر الجزيل إلى كل من ساعدنا في إنجاز هذا البحث، سواء من قريب أو بعيد.

ويشرفنا أن نتقدم بأسمى عبارات الشكر والتقدير إلى:

- الأستاذ المشرف **بقاص منير** الذي تكرم علينا بقبول إشرافه على هذه المذكرة
- أساتذة كلية العلوم الدقيقة على دعمهم وتوجيهاتهم القيمة
- كل من ساهم في إنجاز هذا العمل، ولو بكلمة طيبة أو ابتسامة صادقة

إهداء

الحمد لله الذي وفقني لهذا، ولولا فضل الله علينا ما كنا لنصل إليه.

أما بعد، فإلى من نزلت في حقهم الآيتان الكريمتان، قال تعالى:

﴿ وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوا إِلَّا إِيَّاهُ وَبِالْوَالِدَيْنِ إِحْسَانًا إِمَّا يَبْلُغَنَّ عِندَكَ الْكِبَرَ أَحَدُهُمَا أَوْ كِلَاهُمَا فَلَا تَقُل لَّهُمَا أُفٍّ وَلَا تَنْهَرْهُمَا وَقُل لَّهُمَا قَوْلًا كَرِيمًا﴾

أُهدي هذا العمل المتواضع إلى:

- **أبي وأمي العزيزين** حفظهما الله لي، اللذين سهرا وتعبا على تعليمي، وساندني في إتمام هذا العمل.

- أفراد أسرتي، سندي في الحياة، ولا أحصي لهم فضلًا.

- كافة الأصدقاء والأحباب، كلٌّ باسمه.

المحتويات

1 المقدمة العامة **8**

1.1 خلفية المشروع 8

2.1 دوافع المشروع 8

3.1 أهمية المشروع 8

4.1 أهداف المشروع 8

2 دراسة الموجود **10**

1.2 أهمية دراسة الموجود 10

2.2 تطبيقات مشابهة 10

1.2.2 Money Manager 10

2.2.2 Manage Budget : Expense Tracker 11

3.2.2 مدير المال 12

3.2 تعريف تقنية OCR 13

4.2 نقد التطبيقات 14

5.2 الجهات الفاعلة (Actors) 14

6.2 مواصفات المتطلبات الوظيفية 14

7.2 UML: لغة النمذجة الموحدة 15

8.2 مخططات UML الأساسية 15

9.2 أداة إنشاء المخططات: Mermaid Live Editor 15

10.2 خاتمة 16

3 نمذجة وتصميم النظام **17**

1.3 الممثلون (Actors) 17

2.3 المتطلبات الوظيفية 17

3.3 المتطلبات غير الوظيفية 18

4.3 مخطط الحالة الاستخدامية (Use Case Diagram) 18

5.3 مخطط النشاط (Activity Diagram) 19

6.3 مخطط الحالة (State Diagram) 20

7.3 مخطط الفئات (Class Diagram) 21

8.3 مخطط التتابع (Sequence Diagram) 22

9.3 مخطط المكونات (Component Diagram) 23

10.3 أداة Mermaid Live Editor 23

11.3 خلاصة 23

4 الإنجاز والتطوير **24**
1.4 بيئة التطوير والأدوات 24
2.4 بناء التطبيق العملي 25
1.2.4 الواجهة الرئيسية للتطبيق 26
2.2.4 شاشة تسجيل الدخول 27
3.2.4 شاشة إنشاء حساب جديد 28
4.2.4 إضافة المصاريف والدخل يدوياً 29
5.2.4 واجهة عرض سجل المصاريف و الدخل 31
3.4 مسح الفواتير باستخدام OCR 32
1.3.4 التقاط صورة للفاتورة 32
4.4 استخدام واجهة API OCR Gemini 34
5.4 تخزين البيانات باستخدام Firebase 34
6.4 الإحصائيات وتحليل البيانات 36
1.6.4 عرض الرسوم البيانية 36
7.4 أمان البيانات وخصوصية المستخدم 37
8.4 خلاصة 37

5 النتائج والتقييم **38**
1.5 تجارب التطبيق 38
2.5 تحليل النتائج 38
3.5 الصعوبات والمشاكل 39
4.5 الحلول والتحسينات المقترحة 39
5.5 خلاصة 39

6 الخاتمة العامة **40**
1.6 مقدمة 40
2.6 تلخيص العمل المنجز 40
3.6 الاستنتاجات العامة 40
4.6 آفاق مستقبلية 40
5.6 كلمة ختامية 41

قائمة الأشكال

1 واجهة تطبيق Money Manager 11
2 واجهة تطبيق Manage Budget 12
3 واجهة تطبيق مدير المال 13

4 مخطط الحالة الاستخدامية لتطبيق تتبع المصاريف 18
5 مخطط النشاط لتسجيل مصروف باستخدام 19
6 مخطط الحالة لتطبيق تتبع المصاريف 20
7 مخطط الفئات لتطبيق تتبع المصاريف 21
8 مخطط التتابع لمعالجة مصروف باستخدام OCR 22
9 مخطط المكونات المعمارية للنظام 23

10 الواجهة الرئيسية للتطبيق 26
11 واجهة تسجيل الدخول 27
12 واجهة إنشاء حساب 28
13 إضافة مصروف يدوي 29
14 إضافة دخل يدوي 30
15 عرض سجل المصاريف و الدخل 31
16 التقاط صورة الفاتورة 32
17 إحصائيات رسومية للمصاريف 36

الملخص

تهدف هذه المذكرة إلى تصميم وتطوير تطبيق موبايل ذكي يسمح بمسح تذاكر الإنفاق باستخدام تقنية التعرف الضوئي على الحروف (OCR) المدعومة بالذكاء الاصطناعي عبر Google Gemini، وتحويلها إلى بيانات رقمية قابلة للتتبع والتحليل. تم بناء التطبيق باستخدام لغة Kotlin مع تكامل Firebase كخدمة خلفية لتخزين البيانات. يتميز المشروع بدمج OCR مع واجهات استخدام بسيطة لتقليل الإدخال اليدوي وتحسين تجربة المستخدم في إدارة مصاريفه اليومية.

الكلمات المفتاحية: تطبيق موبايل، تتبع المصاريف، Gemini OCR، AI، Firebase.

Abstract

This thesis aims to design and develop a smart mobile application that scans expense tickets using Optical Character Recognition (OCR) enhanced by Gemini AI, converting them into structured digital data. The app is developed using Kotlin and integrated with Firebase for backend services. The key feature of the project is the use of AI-powered OCR to minimize manual input and improve user experience in daily expense tracking.

Keywords: Mobile Application, Expense Tracking, OCR, Gemini AI, Firebase.

المقدمة العامة

1.1 خلفية المشروع

في عصر الرقمنة والتكنولوجيا المتسارعة، أصبحت إدارة المصاريف من التحديات اليومية التي يواجهها الأفراد. ومع تزايد عدد المعاملات المالية اليومية، ظهرت الحاجة إلى حلول ذكية تساعد المستخدم على تتبع نفقاته بطريقة بسيطة وفعالة.

2.1 دوافع المشروع

الاعتماد على الطرق التقليدية في تسجيل المصاريف كالمفكرات الورقية أو الإدخال اليدوي في التطبيقات يشكل عبئًا على المستخدم، كما أنه عرضة للنسيان أو الخطأ. من هنا جاءت فكرة تطوير تطبيق يُسهل هذه المهمة ويجعلها تلقائية بقدر الإمكان، خاصة مع توفر تقنيات ذكاء اصطناعي متقدمة مثل Gemini من Google.

3.1 أهمية المشروع

يساهم هذا المشروع في تحسين ثقافة التسيير المالي الشخصي، ويقلل من الجهد المطلوب لتوثيق المصاريف. كما أنه يستغل تقنيات حديثة مثل Gemini OCR وFirebase لتوفير تجربة استخدام ذكية وفعالة.

4.1 أهداف المشروع

- تطوير تطبيق موبايل باستخدام Kotlin.
- دمج تقنية Gemini OCR لمسح التذاكر واستخراج المعلومات منها تلقائيًا.
- استخدام Firebase لتخزين البيانات ومزامنتها.
- توليد تقارير وتحليلات للمصاريف لمساعدة المستخدم على اتخاذ قرارات مالية أفضل.

منهجية المذكرة

تتكون هذه المذكرة من خمسة فصول رئيسية. يُقدم الفصل الأول دراسة تحليلية للتطبيقات المشابهة في السوق. أما الفصل الثاني، فيتناول نمذجة وتصميم النظام باستخدام مخططات UML. ويُخصص الفصل الثالث للجانب التطبيقي، حيث نستعرض فيه بيئة العمل وخطوات بناء التطبيق. ويعرض الفصل الرابع نتائج التطبيق وتقييمه. وأخيرًا، يقدم الفصل الخامس الخاتمة والاستنتاجات مع الآفاق المستقبلية لتطوير المشروع.

دراسة الموجود

مقدمة

تُعد دراسة الموجود خطوة أساسية في أي مشروع برمجي، فهي تتيح للمطورين فهماً أعمق للمشاكل الموجودة، وتحليل الأنظمة الحالية التي تحاول حل هذه المشاكل. كما تسلط الضوء على أوجه النقص في هذه الأنظمة، مما يفسح المجال لتقديم حلول مبتكرة كتطبيقنا الحالي.

1.2 أهمية دراسة الموجود

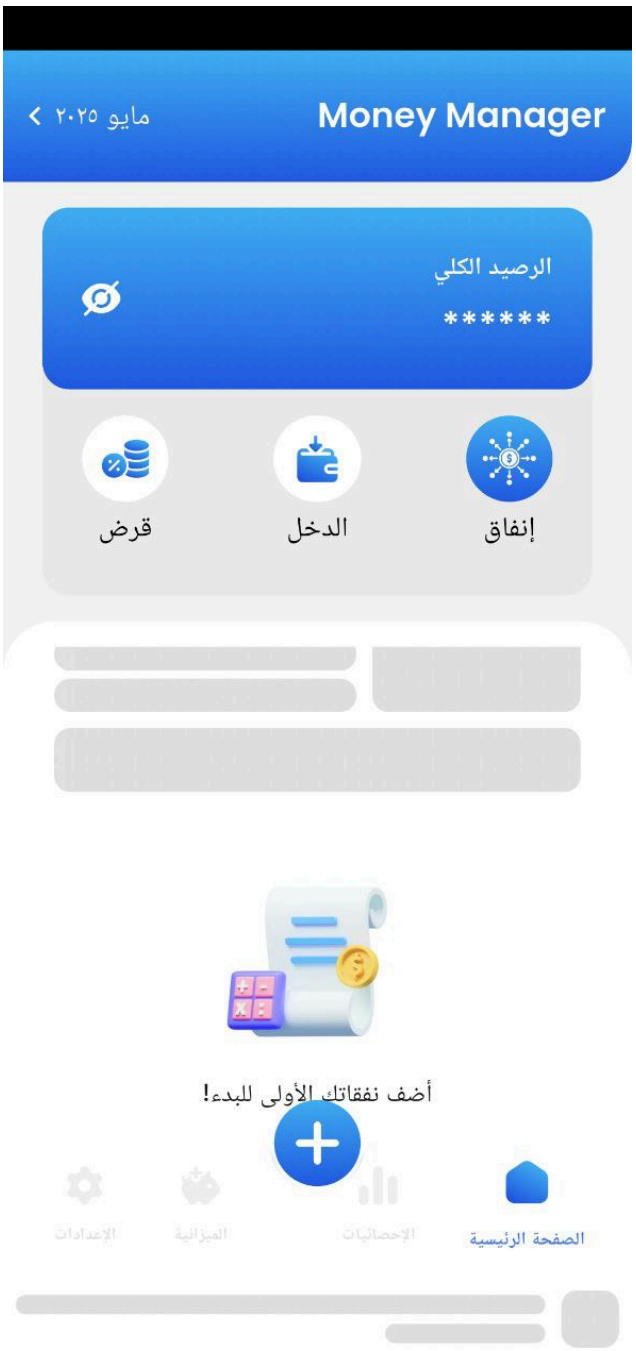
قبل الشروع في بناء أي نظام برمجي جديد، من الضروري فهم التطبيقات الحالية التي تُقدم خدمات مشابهة. هذه الدراسة تساعد في تحديد نقاط القوة للاستفادة منها، ونقاط الضعف لتفاديها، مما يُمكّننا من تقديم منتج يلبي حاجات المستخدمين بطريقة أكثر كفاءة وذكاء.

2.2 تطبيقات مشابهة

قمنا بمراجعة عدد من التطبيقات الشائعة في مجال تتبع المصاريف:

1.2.2 Money Manager

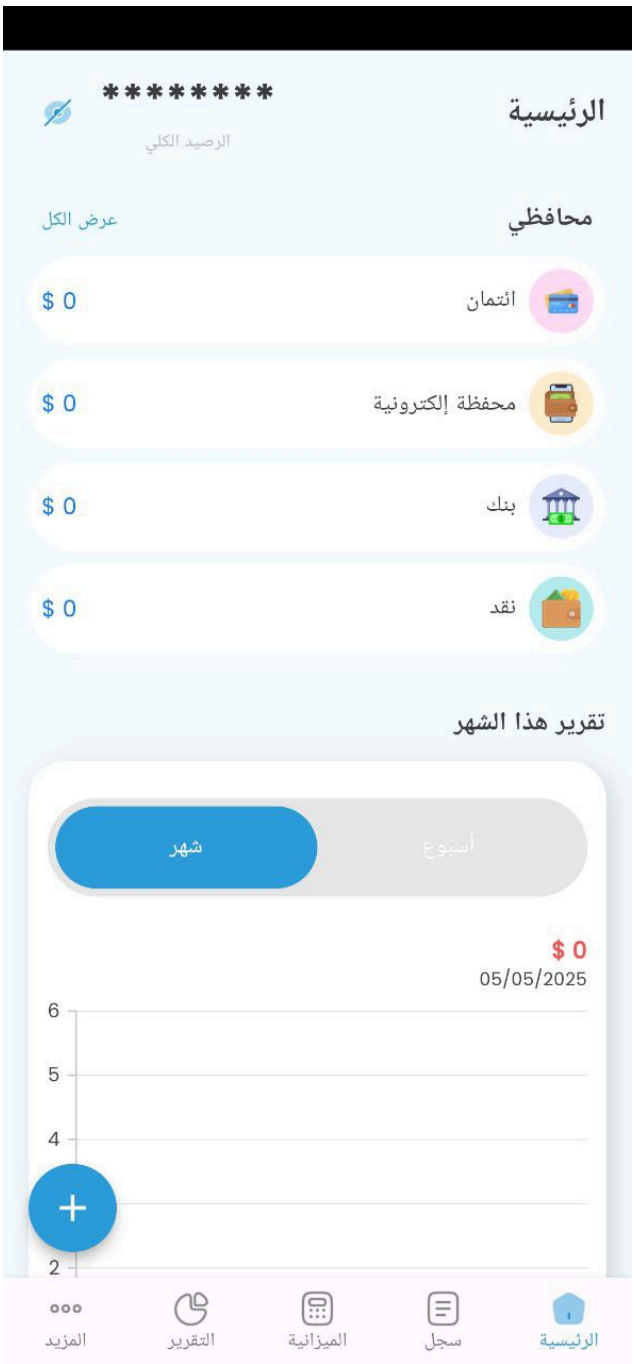
تطبيق شهير يسمح للمستخدمين بإدخال مصاريفهم يدويًا وتصنيفها يدويًا كذلك، مع تقديم رسوم بيانية وتقارير دورية. لكنه لا يدعم خاصية مسح الفواتير باستخدام OCR.



شكل 1: واجهة تطبيق Money Manager

2.2.2 Manage Budget : Expense Tracker

تطبيق يوفر ربطًا بالحسابات البنكية لتصنيف المصاريف آليًا، مع تنبيهات للميزانية وتحليلات شهرية. لكنه لا يتيح التقاط الفواتير أو استخراج محتواها بشكل ذكي.



شكل 2: واجهة تطبيق Manage Budget

3.2.2 مدير المال

يركز على تصميم جذاب وواجهة سهلة لإنشاء ميزانيات وتدوين المصروفات يدويًا. يفتقر تمامًا إلى التكامل مع تقنيات OCR أو AI.

شكل 3: واجهة تطبيق مدير المال

3.2 تعريف تقنية OCR

تقنية OCR (Optical Character Recognition) أو "التعرف الضوئي على الحروف"، هي تقنية تُستخدم لاستخراج النصوص من الصور والمستندات الورقية وتحويلها إلى نصوص رقمية قابلة للمعالجة والبحث. تعتمد هذه التقنية على خوارزميات في رؤية الحاسوب (Computer Vision) ومعالجة الصور لاكتشاف النص الموجود داخل الصور، ثم تتعرف على الحروف والكلمات وتحولها إلى بيانات يمكن التعامل معها برمجيًا.

تمر عملية OCR بعدة مراحل:

- **تحسين الصورة**: تصحيح الإضاءة والتباين والتخلص من الضوضاء.
- **اكتشاف النص**: تحديد أماكن النصوص داخل الصورة.
- **تحليل الحروف**: استخدام خوارزميات التعرف على الأنماط (Pattern Recognition) لتحويل الصور إلى حروف.
- **الإخراج الرقمي**: إنتاج النص بصيغة قابلة للنسخ والمعالجة.

وقد تطورت هذه التقنية بفضل الذكاء الاصطناعي، حيث أصبحت أكثر دقة وسرعة، خاصة عند دمجها مع تقنيات حديثة مثل Gemini AI الذي يُستخدم في تطبيقنا لاستخلاص معلومات المصاريف من صور الفواتير بشكل تلقائي وذكي.

4.2 نقد التطبيقات

لاحظنا أن معظم التطبيقات الحالية تفتقر إلى:

- دعم OCR لاستخلاص البيانات تلقائيًا من التذاكر.
- التفاعل الذكي القائم على الذكاء الاصطناعي.
- سهولة التتبع الفوري للمصروفات عبر الصور.

هذه الفجوات تُبرّر الحاجة لتطوير تطبيق جديد يُدمج Gemini AI-powered OCR لتحويل الصور إلى بيانات قابلة للتحليل، مما يختصر الوقت ويرفع من دقة التوثيق.

5.2 الجهات الفاعلة (Actors)

- **المستخدم**: يُسجّل دخوله، يلتقط صور الفواتير، يراجع المصاريف المصنفة.
- **النظام الذكي**: يستخلص البيانات من الصور باستخدام Gemini OCR، يُخزن البيانات في Firebase ويُنتج تحليلات بيانية.

6.2 مواصفات المتطلبات الوظيفية

- تسجيل المستخدمين وتوثيقهم.
- التقاط صورة تذكرة.
- استخدام OCR لاستخلاص البيانات.

- تخزين المصاريف وتصنيفها.
- عرض الرسوم البيانية والإحصائيات.

7.2 UML: لغة النمذجة الموحدة

تعريف UML

UML (Unified Modeling Language) هي لغة رسومية موحدة تُستخدم لتصور وتصميم وتوثيق مكونات الأنظمة البرمجية. تتيح للمطورين والمحللين فهم الأنظمة المعقدة من خلال مخططات مرئية تغطي بنية النظام وسلوكه.

أهداف UML

- تبسيط التعقيد البرمجي عبر تقديم عرض مرئي منطقي.
- تحسين التواصل بين أعضاء فريق التطوير.
- تسهيل توثيق جميع العلاقات والعمليات في النظام.
- دعم جميع مراحل تطوير البرمجيات من التحليل حتى التنفيذ.

8.2 مخططات UML الأساسية

- **مخطط حالة الاستخدام (Use Case Diagram)**: يُظهر التفاعلات بين المستخدمين (Actors) والنظام، ويستخدم لتحديد الوظائف الأساسية.
- **مخطط الفئات (Class Diagram)**: يوضح بنية النظام من حيث الكائنات (Classes) والعلاقات بينها مثل الوراثة والتكوين.
- **مخطط التتابع (Sequence Diagram)**: يصف كيفية تفاعل الكائنات المختلفة مع بعضها البعض بمرور الوقت لإنجاز مهمة معينة.

9.2 أداة إنشاء المخططات: Mermaid Live Editor

تم إنشاء مخططات UML الخاصة بنا باستخدام أداة **Mermaid Live Editor**، وهي أداة مفتوحة المصدر تدعم الرسم التخطيطي بلغة برمجية بسيطة تُكتب مباشرة وتُحوّل إلى رسم تفاعلي. تميزت بسهولة الاستخدام وسرعة توليد المخططات بدون الحاجة إلى بيئة معقدة.

10.2 خاتمة

من خلال هذا الفصل، قمنا بتحليل ودراسة أهم التطبيقات المنافسة واستخلصنا نقاط القوة والضعف فيها. الفجوة التي لاحظناها - وهي غياب OCR والذكاء الاصطناعي - كانت الدافع الأساسي لتطوير تطبيقنا الذي يعالج هذه النقطة بذكاء وسلاسة. كما عرضنا الجهات الفاعلة والمتطلبات الأساسية، وختمنا بتعريف UML، أنواع مخططاتها، وأداة Mermaid Live Editor التي استخدمناها للنمذجة. كل هذا ساعدنا على بناء تصور أولي واضح لمسار تطبيقنا في المراحل المقبلة.

نمذجة وتصميم النظام

مقدمة

في هذا الفصل، نُحلّل ونصمّم تطبيق تتبُّع المصاريف الذكي من خلال تقديم عرض شامل لمكوناته البنيوية والسلوكية، وذلك باستخدام مجموعة من مخططات UML. نبدأ بتحديد المتطلبات وتحليل الممثلين، ثم نمثّل مختلف السيناريوهات والمكونات والأنشطة باستخدام أدوات مثل Mermaid Live Editor وStarUML. تساعد هذه النماذج في بناء تصور عملي متكامل لكيفية عمل النظام.

1.3 الممثلون (Actors)

يتفاعل النظام مع نوعين من المستخدمين:

- **مستخدم مجهول (Anonymous User)**: يمكنه الوصول إلى واجهات تسجيل الدخول أو التسجيل فقط.
- **مستخدم موثَّق (Authenticated User)**: يمكنه استخدام جميع خصائص النظام، مثل تسجيل الدخل والمصاريف، والتقاط الفواتير، وعرض الإحصائيات.

2.3 المتطلبات الوظيفية

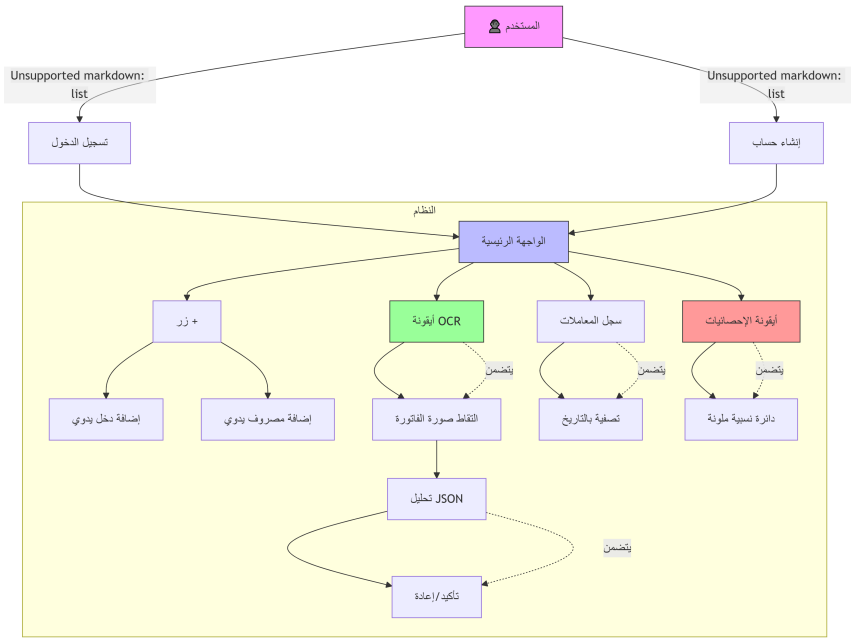
- تسجيل الدخول عبر البريد الإلكتروني أو باستخدام Google/Facebook/Apple.
- إنشاء حساب جديد.
- استعادة كلمة المرور.
- إدخال دخل يدويًا.
- إدخال مصروف يدويًا أو عبر التقاط صورة لإيصال الإنفاق.
- استخدام Gemini OCR لاستخلاص بيانات المصروف من صورة الفاتورة فقط.
- تصنيف المصروفات تلقائيًا.

- تخزين البيانات باستخدام Firebase Realtime Database.
- عرض سجل المصاريف والدخل بالإضافة إلى تحليلات رسومية.
- تعديل أو حذف العمليات المالية.

3.3 المتطلبات غير الوظيفية

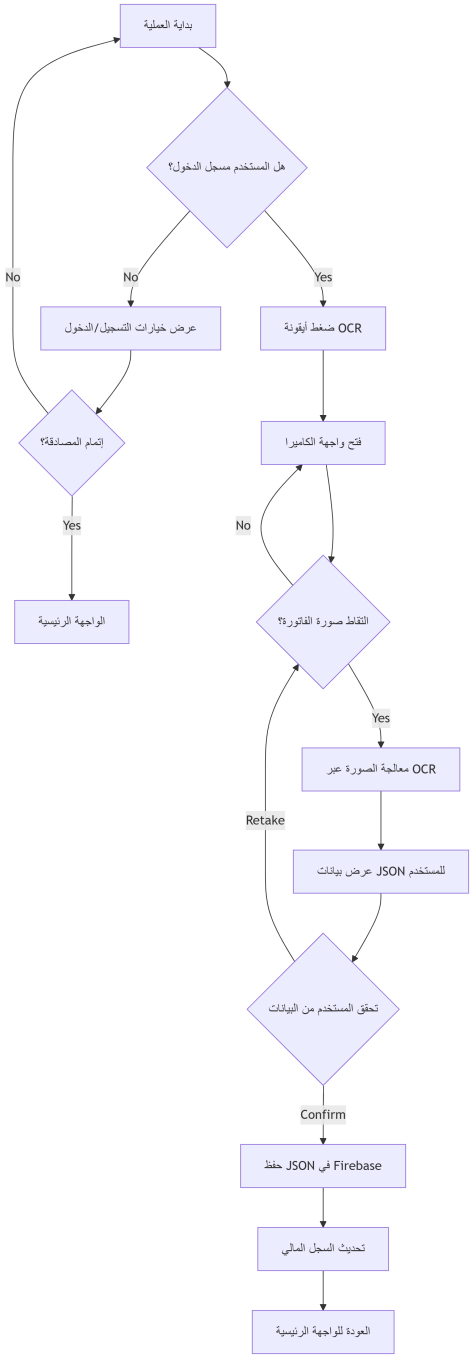
- واجهة استخدام مبنية باستخدام Jetpack Compose.
- استجابة فورية وتحليل سريع للصور.
- حماية بيانات المستخدم من خلال مصادقة Firebase.
- دعم لغات متعددة.

4.3 مخطط الحالة الاستخدامية (Use Case Diagram)



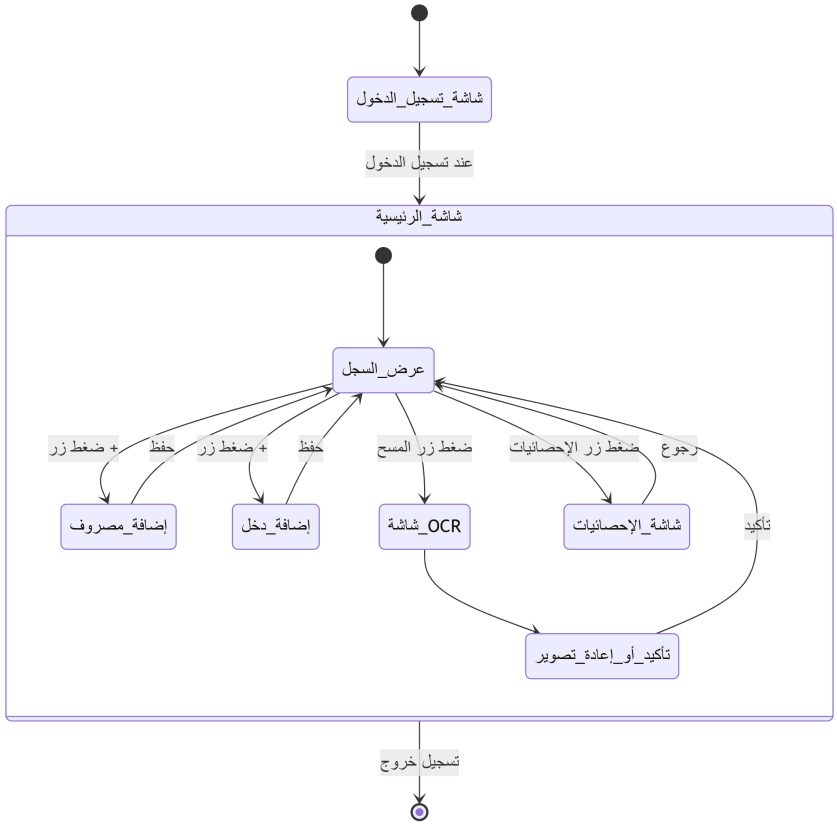
شكل 4: مخطط الحالة الاستخدامية لتطبيق تتبع المصاريف

5.3 مخطط النشاط (Activity Diagram)



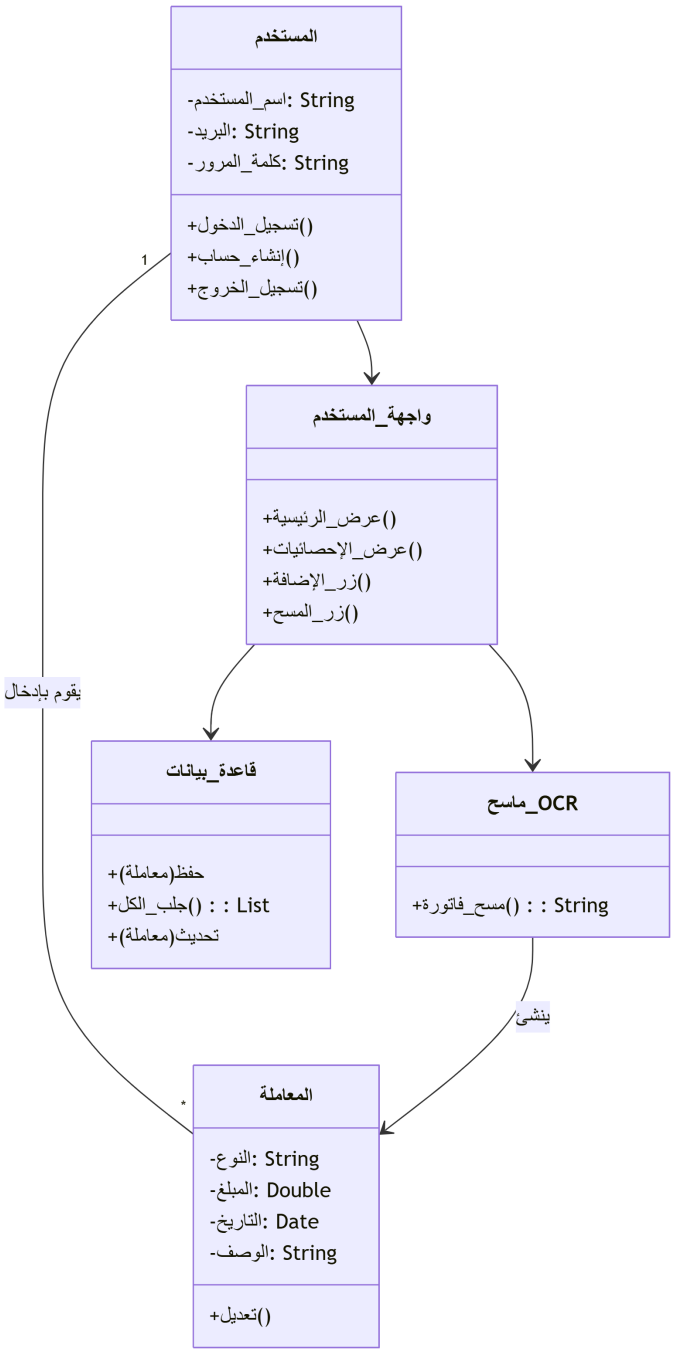
شكل 5: مخطط النشاط لتسجيل مصروف باستخدام

6.3 مخطط الحالة (State Diagram)



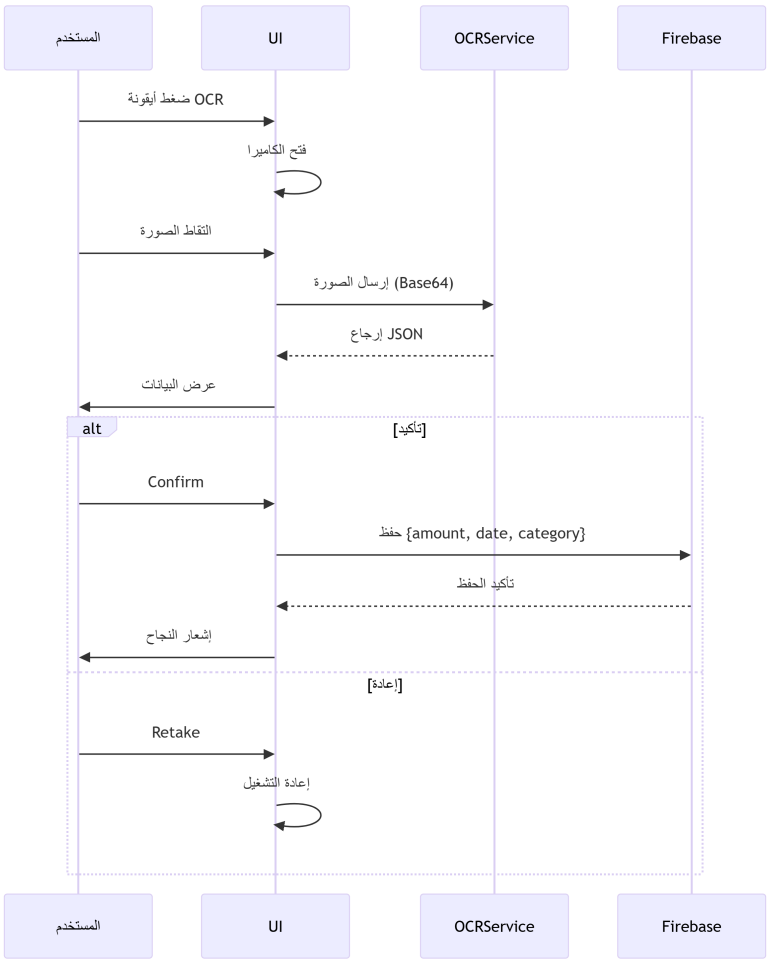
شكل 6: مخطط الحالة لتطبيق تتبع المصاريف

7.3 مخطط الفئات (Class Diagram)



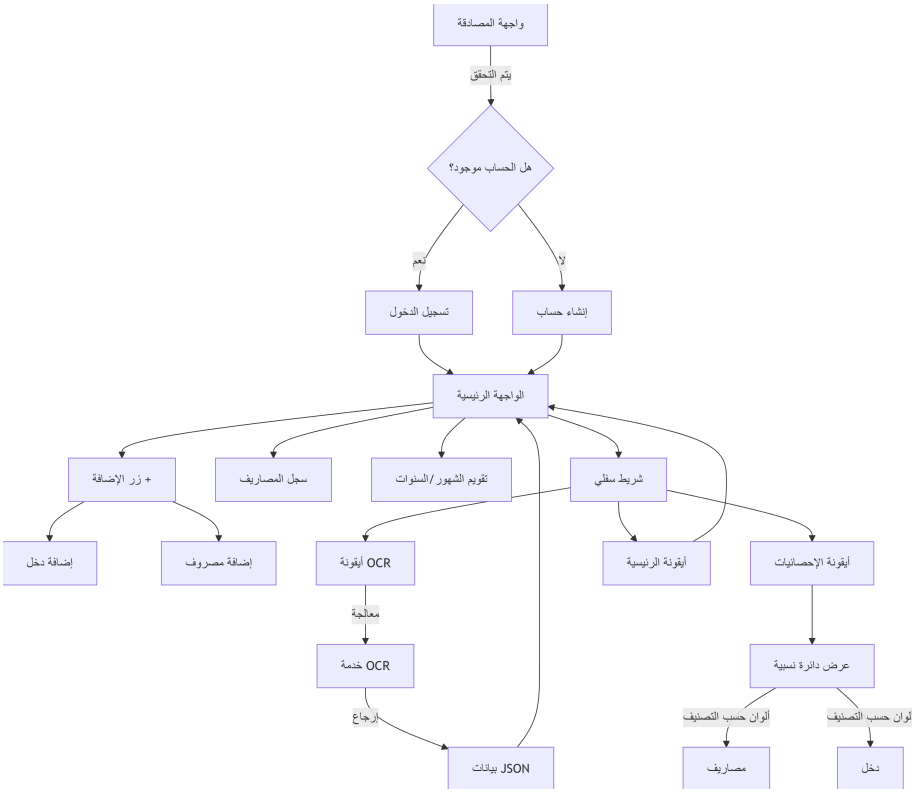
شكل 7: مخطط الفئات لتطبيق تتبع المصاريف

8.3 مخطط التتابع (Sequence Diagram)



شكل 8: مخطط التتابع لمعالجة مصروف باستخدام OCR

9.3 مخطط المكونات (Component Diagram)



شكل 9: مخطط المكونات المعمارية للنظام

10.3 أداة Mermaid Live Editor

تم استخدام أداة **Mermaid Live Editor** في إنشاء معظم المخططات بطريقة مرنة، حيث تتيح كتابة المخطط كنص برمجي وتحويله تلقائيًا إلى رسم تخطيطي قابل للتصدير بصيغ SVG/PNG.

11.3 خلاصة

استعرضنا في هذا الفصل التصور الكامل لتصميم النظام، بدءًا من تحديد المتطلبات وتحليل الأدوار، وصولاً إلى إنشاء المخططات البنيوية والسلوكية التي تصف آلية عمل النظام. يشكل هذا التصميم مرجعًا أساسيًا للمرحلة القادمة وهي مرحلة الإنجاز والتطوير.

الإنجاز والتطوير

مقدمة

بعد الانتهاء من مرحلة التحليل والنمذجة، بدأنا في تطوير التطبيق الذكي لتتبع المصاريف، بالاعتماد على تقنية التعرف البصري على الحروف (OCR) المدعومة بالذكاء الاصطناعي من خلال واجهة برمجية مقدمة من Gemini AI، بالإضافة إلى خدمات قاعدة البيانات والمصادقة السحابية من Firebase. في هذا الفصل نستعرض بيئة العمل، الأدوات والتقنيات المستخدمة، مراحل التطوير العملي، خوارزمية معالجة الفواتير، وآليات الأمان.

1.4 بيئة التطوير والأدوات

بيئة الأجهزة

تم تطوير التطبيق على حاسوب بمواصفات كافية لضمان تجربة تطوير سلسة:

- **نظام التشغيل**: Windows 10
- **المعالج**: Intel Core i5, 8th Gen
- **الذاكرة العشوائية (RAM)**: 8 GB
- **التخزين**: 512 GB SSD

بيئة البرمجيات والتقنيات

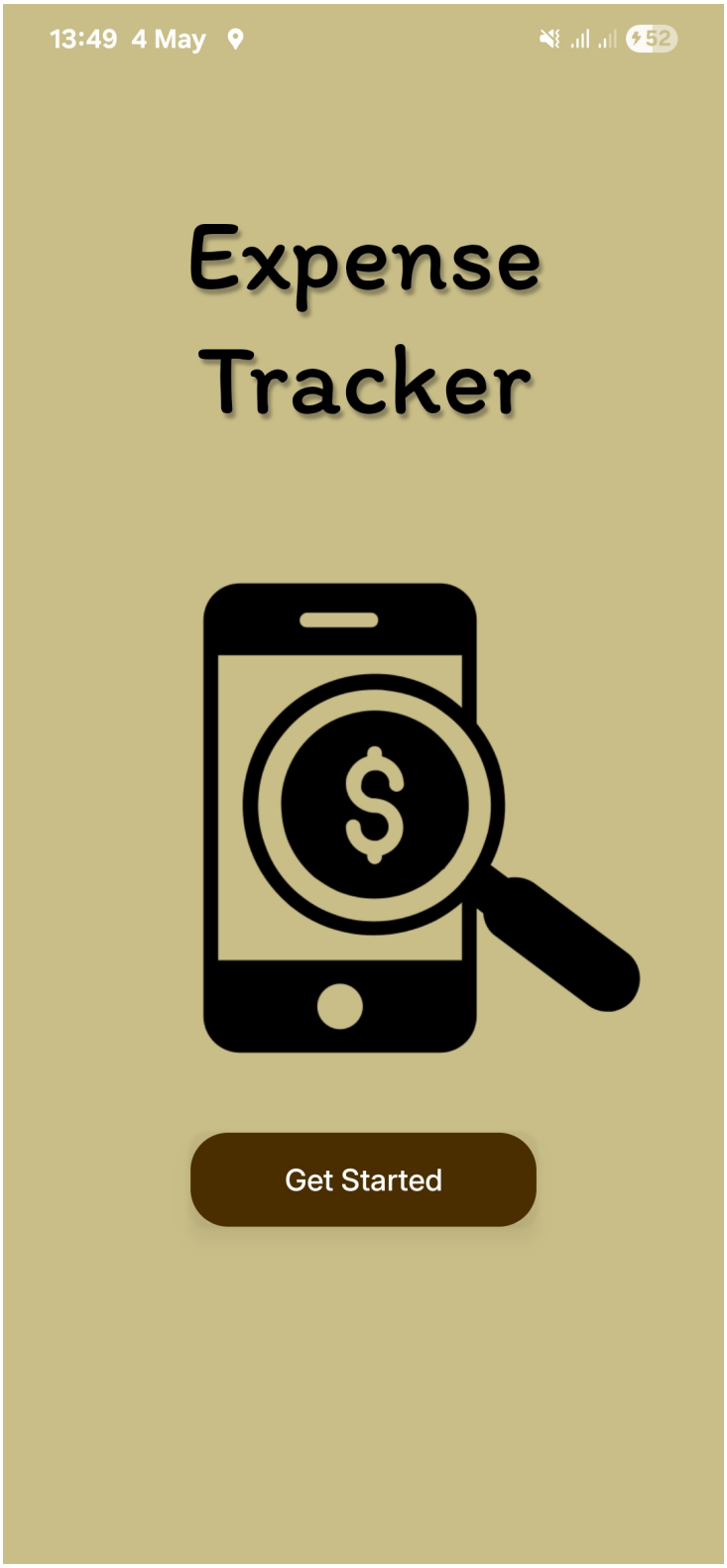
- Android Studio — بيئة تطوير متكاملة لبناء تطبيقات أندرويد.
- Kotlin — لغة برمجة رسمية مدعومة من جوجل لتطوير تطبيقات أندرويد.
- Jetpack Compose — إطار حديث لبناء واجهات المستخدم بطريقة تفاعلية ومرنة.
- Firebase — يقدم خدمات المصادقة وتخزين البيانات في الوقت الحقيقي.
- Gemini AI OCR API — تقنية ذكاء اصطناعي مقدمة من Google DeepMind لاستخلاص النصوص من الصور.

- Mermaid.js — لتوليد مخططات UML أثناء تحليل النظام.

2.4 بناء التطبيق العملي

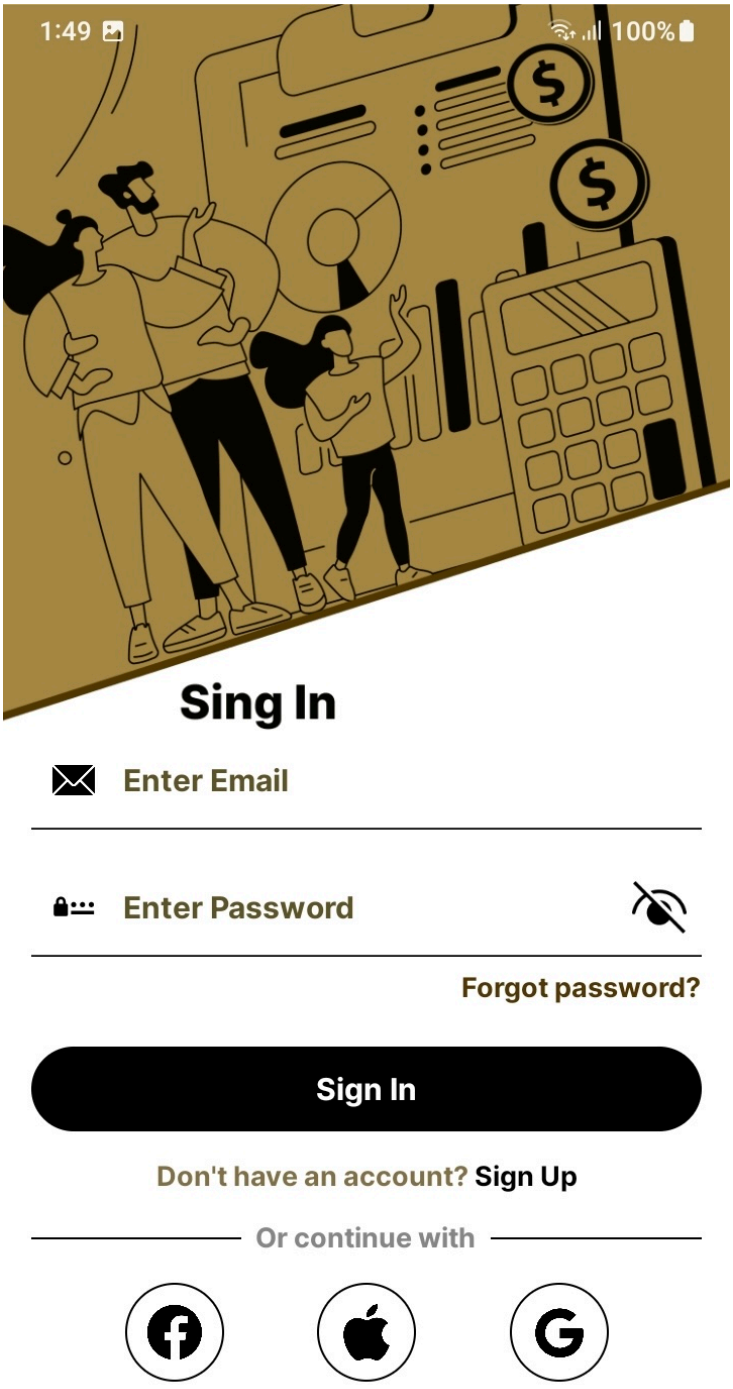
يحتوي التطبيق على عدة واجهات متكاملة، تُنفذ مهام متنوعة مثل تسجيل الدخول، إضافة المصاريف، تحليل الفواتير، وعرض الإحصائيات.

1.2.4 الواجهة الرئيسية للتطبيق



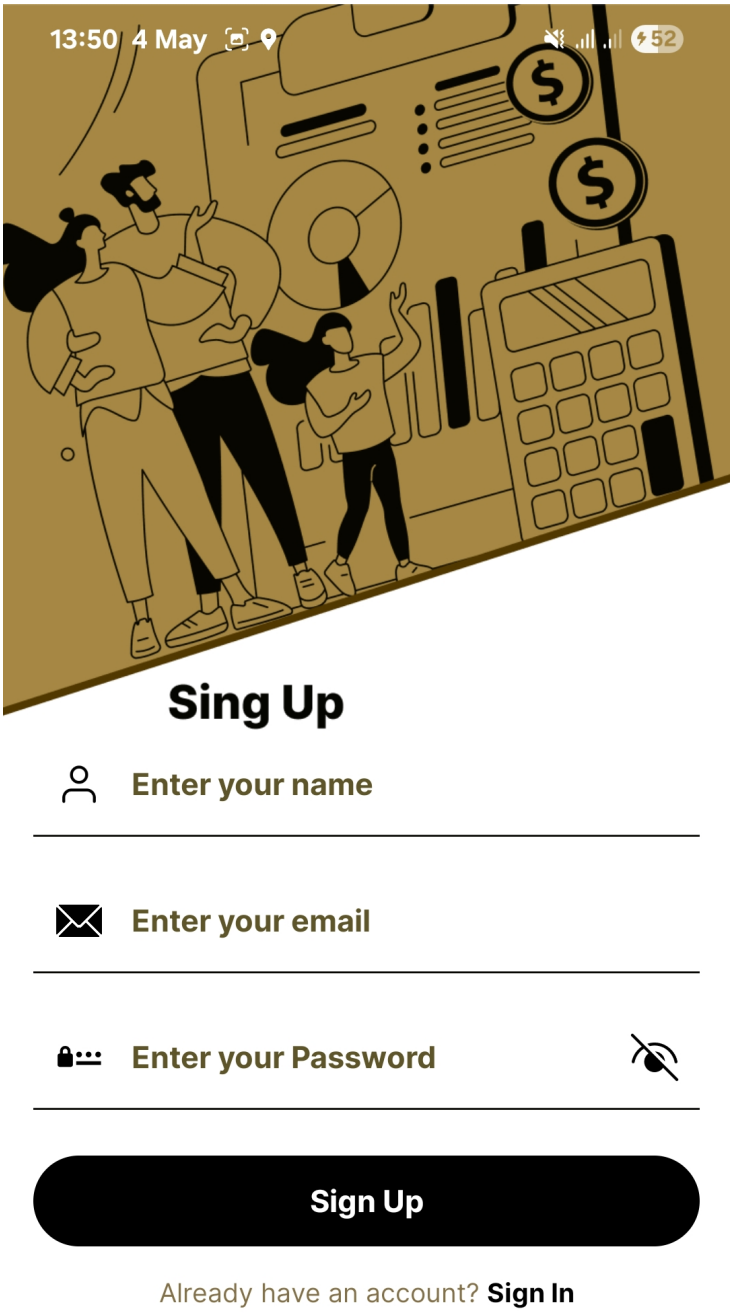
شكل 10: الواجهة الرئيسية للتطبيق

2.2.4 شاشة تسجيل الدخول



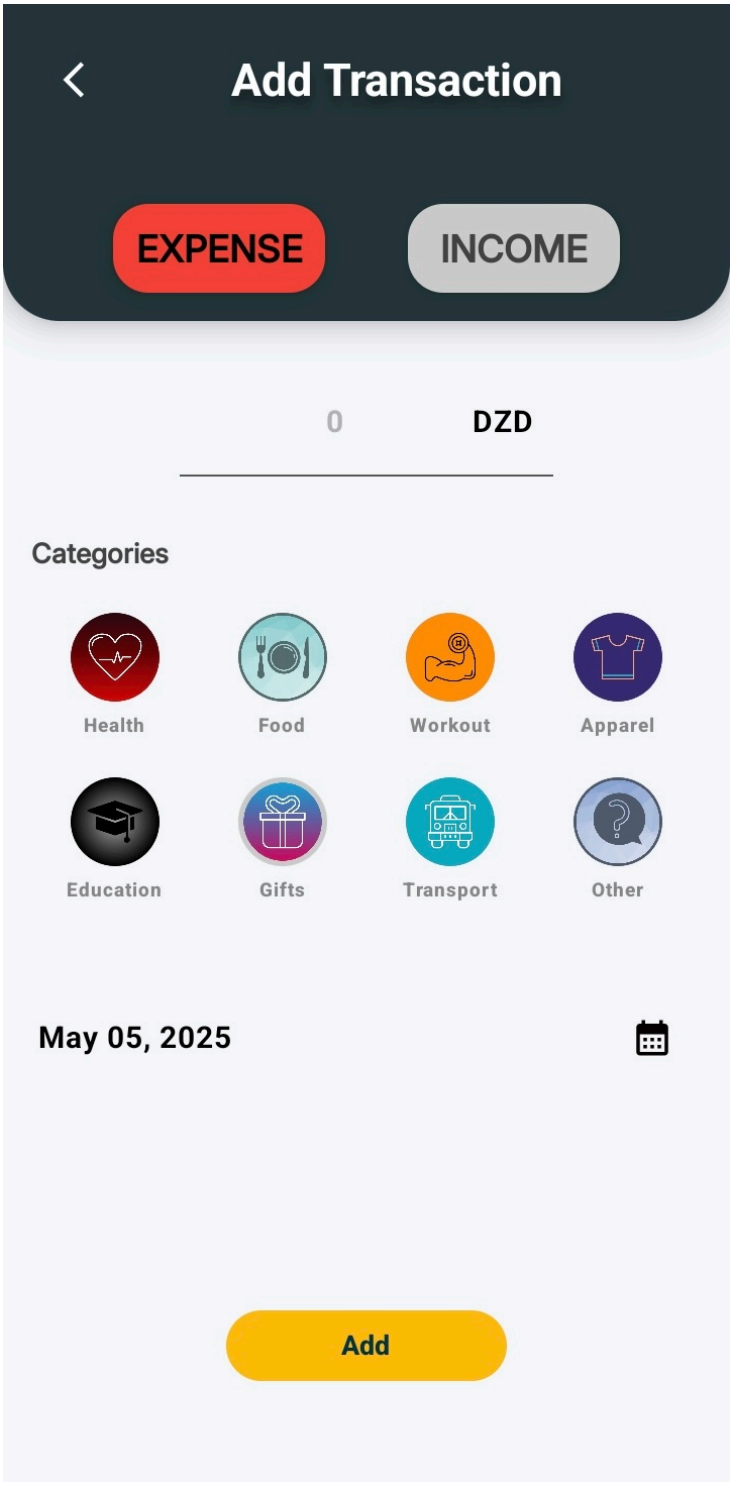
شكل 11: واجهة تسجيل الدخول

3.2.4 شاشة إنشاء حساب جديد

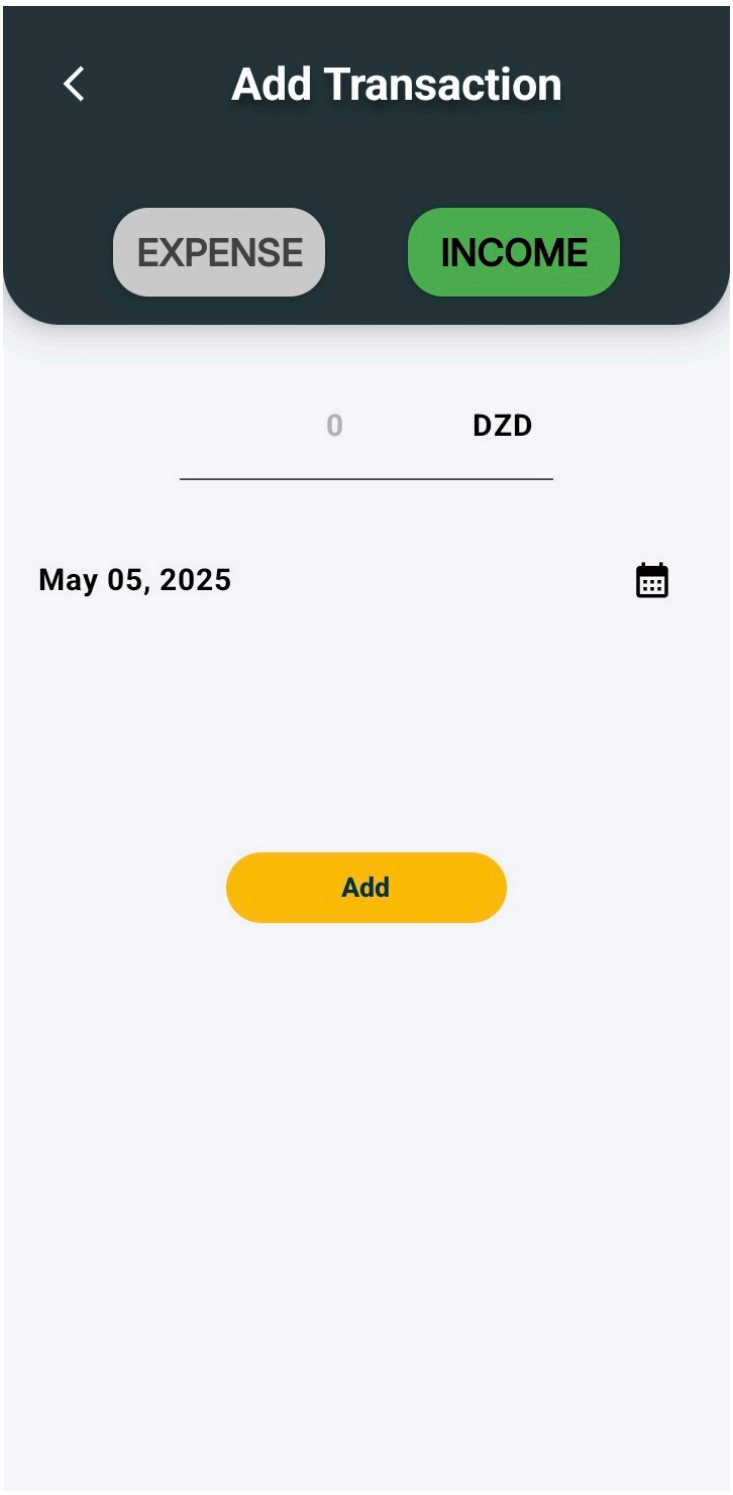


شكل 12: واجهة إنشاء حساب

4.2.4 إضافة المصاريف والدخل يدوياً

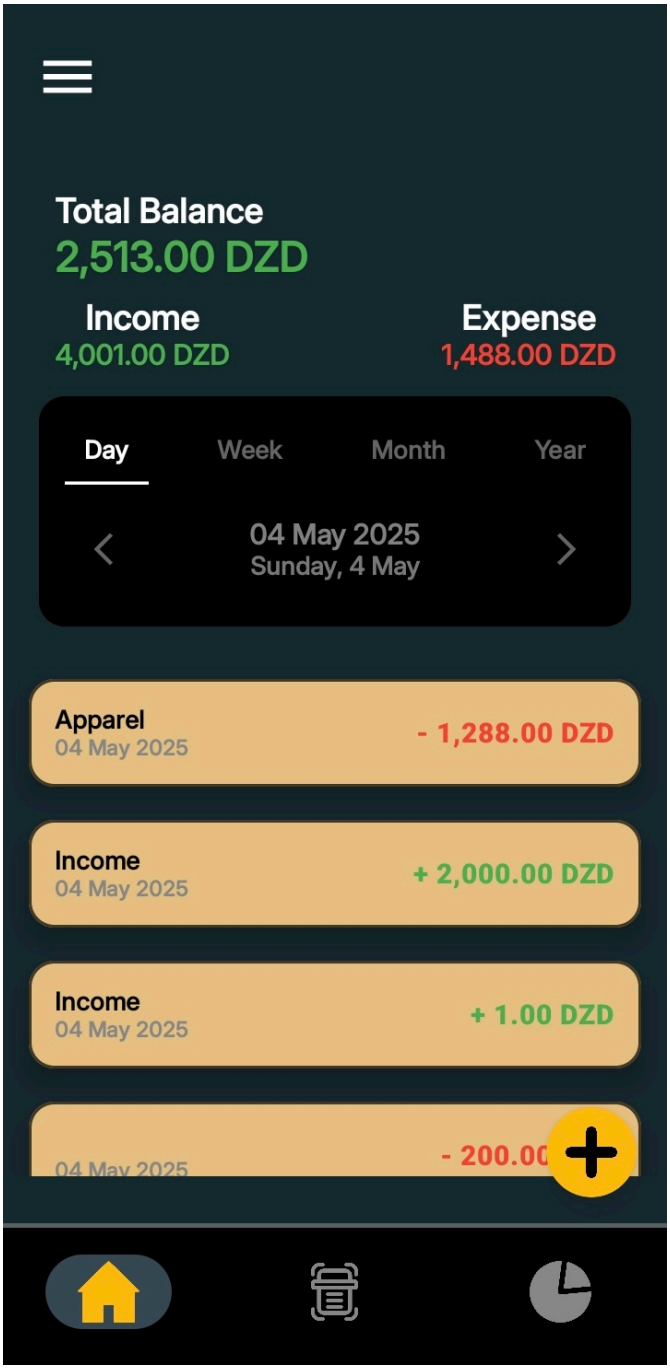


شكل 13: إضافة مصروف يدوي



شكل 14: إضافة دخل يدوي

5.2.4 واجهة عرض سجل المصاريف و الدخل



شكل 15: عرض سجل المصاريف و الدخل

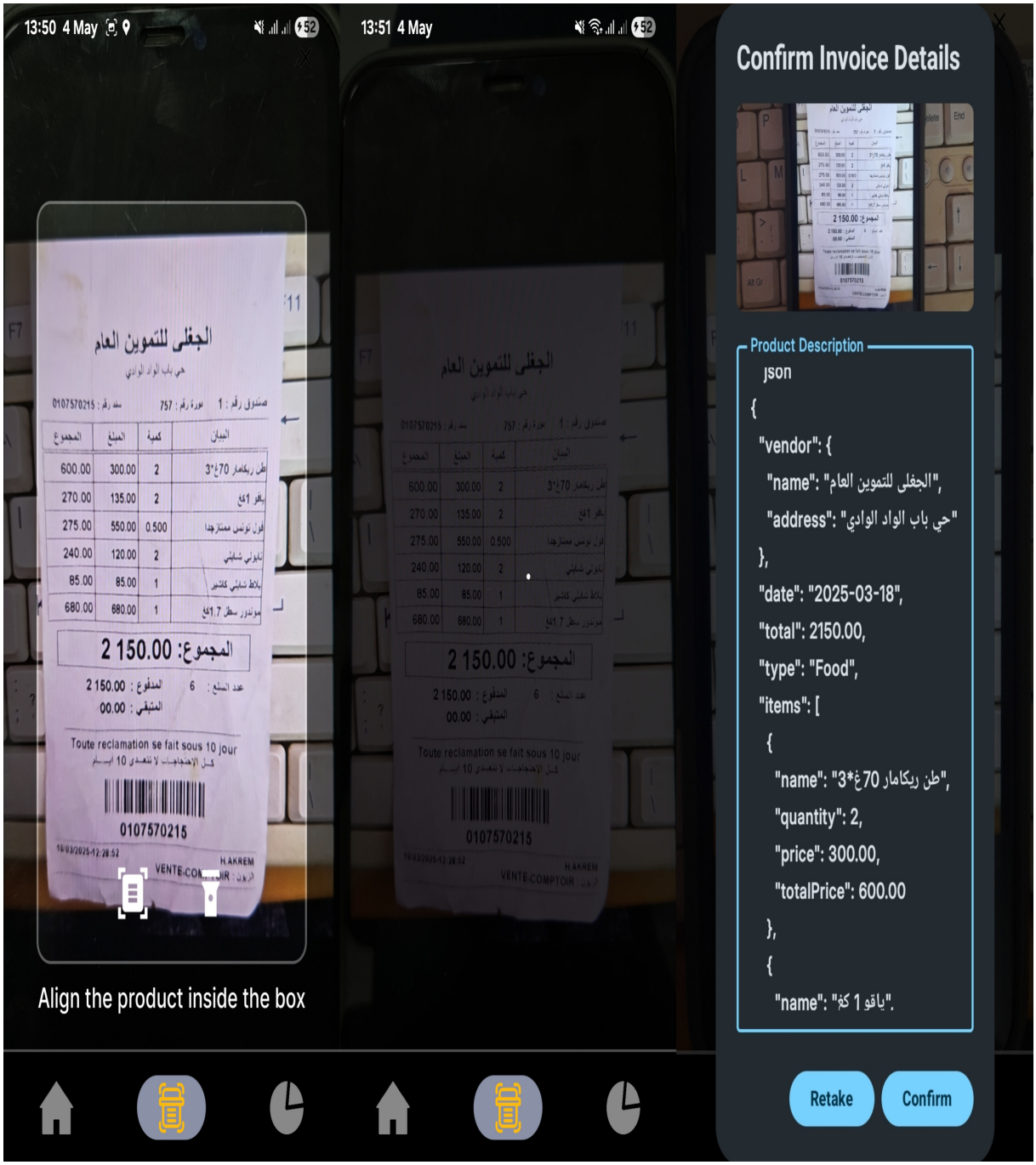
الوصف: عرض قائمة من المداخيل و المصاريف المدخلة مصنفة حسب التاريخ والتصنيف.

الوصف: يُمكن للمستخدم إدخال المصروف أو الدخل بشكل يدوي عبر ملء الحقول التالية:

- المبلغ (مثال: 2000 دينار)
- التصنيف (مثال: طعام، نقل، صحة)
- التاريخ (يُحدّد يدويًا أو تلقائيًا)

3.4 مسح الفواتير باستخدام OCR

1.3.4 التقاط صورة للفاتورة



شكل 16: التقاط صورة الفاتورة

بمجرد التقاط صورة للفاتورة، يتم إرسالها إلى واجهة برمجة التطبيقات الخاصة بـ Gemini OCR، والتي تقوم باستخلاص المعلومات النصية تلقائيًا.

نموذج بيانات مستخرجة (JSON)

المثال التالي يوضح شكل ناتج JSON واقعي بعد تحليل صورة فاتورة مشتريات.


```
"vendor": {
"name": "الجغلى للتموين العام",
"address": "حي باب الواد الوادي"
},
"date": "2025-03-18",
"total": "2150.00",
"type": "food",
"items": [
{
"name": "طن ريكامار 70غ*3",
"quantity": 2,
"price": 300.00,
"total_price": 600.00
},
{
"name": "ياقو 1 كغ",
"quantity": 2,
"price": 135.00,
"total_price": 270.00
},
{
"name": "فول تونس ممتاز جدا",
"quantity": 0.5,
"price": 550.00,
"total_price": 275.00
},
{
"name": "نابولي شايلي",
"quantity": 2,
"price": 120.00,
"total_price": 240.00
},
{
"name": "بلاط شايلي كاشير",
"quantity": 1,
"price": 85.00,
"total_price": 85.00
},
{
"name": "موندور سطل 7.1 كغ",
"quantity": 1,
"price": 680.00,
"total_price": 680.00
}
]
```


عرض النتائج للمستخدم

بعد التحليل، تظهر البيانات للمستخدم مع إمكانية مراجعتها قبل حفظها نهائيًا في قاعدة البيانات.

4.4 استخدام واجهة API OCR Gemini

طلب API لتحليل الصورة

```
curl "https://generativelanguage.googleapis.com/v1beta/models/
gemini0.2--flash:generateContent?key=AIzaSyB04Rd-g4N0Cio9NX1ok9w0j9QjL_6mMgc"
    -X POST  -H "Content-Type: application/json"  -d '
    "contents": [
    "parts": [
    "inlineData":
    "mimeType": "image/jpeg",
    "data": "/9j/4AAQSkZJRgABAQAAAQABAAD/..."
    ,
    "text": "Please extract the invoice information in structured JSON format,
    including: date (yyyy-MM-dd), total amount (DZD), item list (name,
    quantity, unit price, total price), and vendor details (name and address)."
    ]
    ]
    '
```

الدمج في كود Kotlin

```
suspend fun analyzeReceipt(imageBase64: String): JSONObject
    val url = "https://api.gemini.google.com/v1/ocr"
    val client = HttpClient()
    val response = client.post(url)
    headers
    append(HttpHeaders.Authorization, "Bearer YOUR_API_KEY")
    append(HttpHeaders.ContentType, "application/json")
    setBody(" ïmage_base64¨: $imageBase64¨")
    return JSONObject(response.bodyAsText())
```

5.4 تخزين البيانات باستخدام Firebase

نموذج حفظ المصاريف في قاعدة البيانات

```
val database = FirebaseDatabase.getInstance().reference

val expense = mapOf(
    "vendor" to mapOf(
        "name" to "Fashion Store",
        "address" to "123 Rue Didouche Mourad, Algiers"
    ),
    "date" to "202504-"15-,
    "total" to "1000 DA",
    "type" to "Apparel",
    "items" to listOf(
        mapOf(
            "name" to "T-shirt",
```

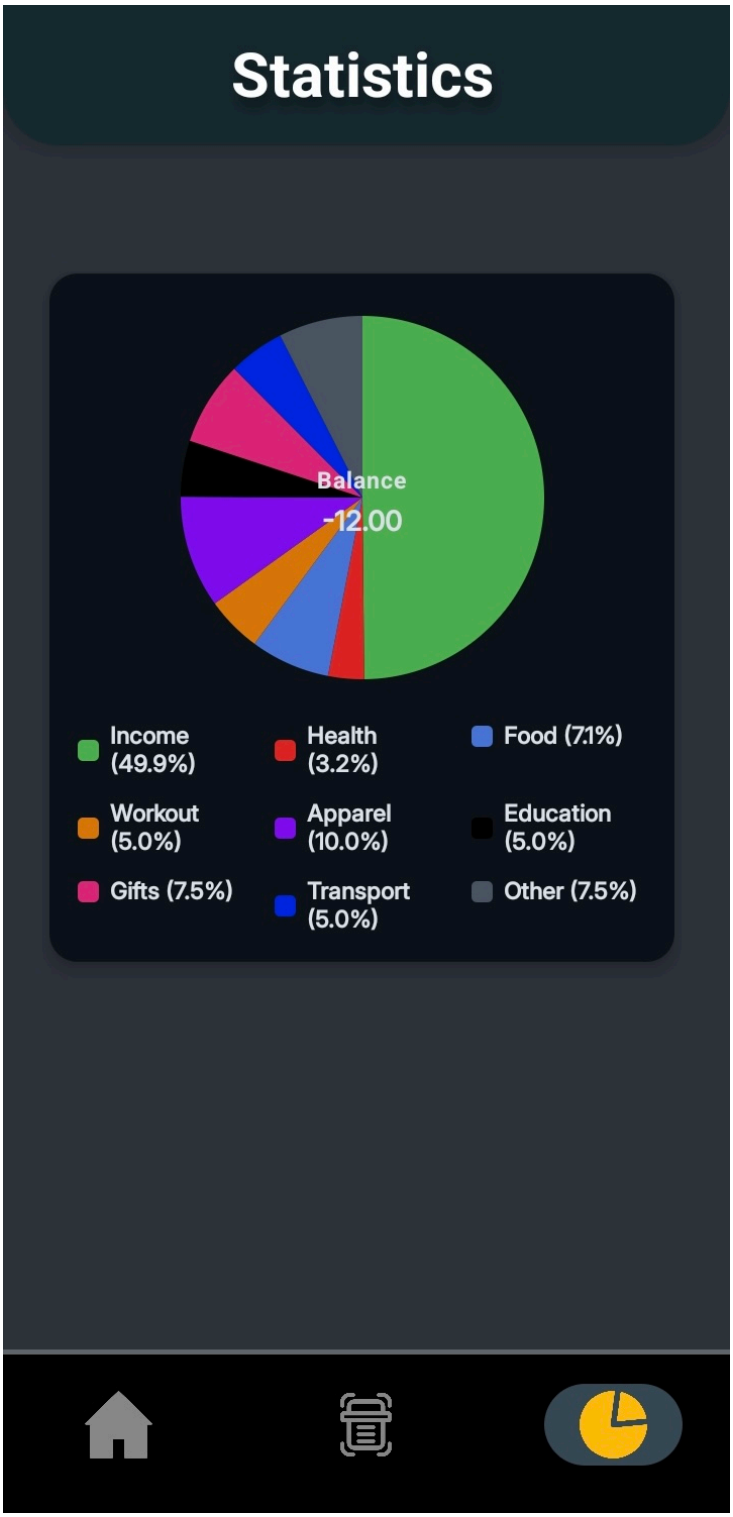
```
            "quantity" to 2,
            "price" to 0.300,
            "totalPrice" to 0.600
        )
    )
)

database.child("expenses").child(userId).push().setValue(expense)
```

تنظيم البيانات

يتم تخزين البيانات تحت كل مستخدم باستخدام UID المميز له لضمان الخصوصية.

6.4 الإحصائيات وتحليل البيانات

1.6.4 عرض الرسوم البيانية



شكل 17: إحصائيات رسومية للمصاريف

يستطيع المستخدم تتبع حجم المصاريف حسب التصنيفات (مثل الطعام، النقل...) والزمن (شهرياً أو أسبوعياً).

7.4 أمان البيانات وخصوصية المستخدم

الإجراءات المتبعة

- **المصادقة**: باستخدام Firebase Authentication.
- **تأمين البيانات**: ربط كل إدخال بـ UID وتطبيق قواعد وصول دقيقة.
- **الصور**: تُستخدم مؤقتًا وتحذف بعد معالجة النصوص.

مخاطر متوقعة

- الاعتماد على الاتصال بالإنترنت لإرسال الصور وتحليلها.
- تسريب معلومات تسجيل الدخول في حال سوء استخدام الجهاز من طرف المستخدم.

8.4 خلاصة

استعرضنا في هذا الفصل بيئة التطوير والتقنيات المستعملة، وطريقة بناء التطبيق وعرض واجهاته، وكيفية استخدام OCR لاستخلاص البيانات، مع ضمان الأمان والخصوصية. هذه المرحلة شكلت تتويجًا لكل العمل النظري السابق، مما أدى إلى إنتاج تطبيق ذكي وفعّال يساعد المستخدم في إدارة مصاريفه اليومية بدقة وسرعة.

النتائج والتقييم

مقدمة

يتناول هذا الفصل عرض نتائج التطبيق بعد الانتهاء من عملية الإنجاز والتطوير. كما يهدف إلى تقييم الأداء العملي للتطبيق من خلال مجموعة من التجارب، وتحليل النتائج المسجلة، ثم عرض أهم الصعوبات التي واجهتنا أثناء مرحلة التطوير، مع تقديم اقتراحات لتحسين المشروع مستقبلاً.

1.5 تجارب التطبيق

تم إجراء عدة اختبارات عملية على التطبيق للتحقق من وظائفه الأساسية، وشملت التجارب التالية:

- تسجيل الدخول باستخدام بريد إلكتروني وكلمة مرور حقيقيين عبر Firebase Authentication.
- التقاط صورة تذكرة باستخدام كاميرا الجهاز.
- استخراج بيانات التذكرة عبر تقنية Gemini OCR.
- تخزين البيانات المستخرجة في Firebase Realtime Database.
- عرض المصاريف المدخلة وتصنيفها.
- عرض التقارير البيانية الشهرية.

2.5 تحليل النتائج

- **دقة OCR**: بلغت نسبة نجاح استخراج المعلومات من الصور بشكل صحيح حوالي **90%**، مع نسبة أخطاء طفيفة في الحالات التي كانت فيها الإضاءة ضعيفة أو جودة الصورة منخفضة.
- **زمن المعالجة**: يستغرق التطبيق في المتوسط **2 إلى 4 ثواني** لمعالجة صورة التذكرة واستخلاص البيانات.
- **أداء Firebase**: أظهرت قاعدة البيانات أداءً ممتازًا في حفظ واسترجاع البيانات بدون تأخير ملحوظ.

- **واجهة المستخدم**: تبين من التجربة أن الواجهات بسيطة وسهلة الاستخدام حتى للمستخدمين غير التقنيين.

3.5 الصعوبات والمشاكل

خلال عملية التطوير والتجريب، واجهنا بعض التحديات، منها:

- صعوبة التعامل مع صور تذاكر غير واضحة أو ملتقطة بزوايا سيئة مما يؤثر على دقة OCR.
- الحاجة إلى الاتصال الدائم بالإنترنت لاستعمال Firebase وخدمات Gemini AI.
- بعض المشاكل البسيطة في توافق تنسيقات الصور مع مدخلات OCR.

4.5 الحلول والتحسينات المقترحة

- تحسين معالجة الصور قبل إرسالها لـ OCR عبر تقنيات تعزيز الصور (مثل تصحيح الإضاءة أو تعديل التباين).
- تطوير وضع "التخزين المؤقت" Offline Mode لتسجيل المصاريف دون الحاجة لاتصال فوري.
- إضافة خاصية التحقق اليدوي من نتائج OCR قبل الحفظ النهائي.

5.5 خلاصة

من خلال التجارب العملية والتحليل، تبين أن التطبيق يلبي المتطلبات الأساسية المرجوة منه بكفاءة عالية. رغم بعض التحديات التقنية البسيطة، إلا أن النتائج الإجمالية مشجعة، مما يفتح آفاقًا واسعة لتحسينات مستقبلية قد تجعل من النظام أداة أكثر فاعلية لإدارة المصاريف الشخصية.

الخاتمة العامة

الخاتمة العامة

1.6 مقدمة

تشكل هذه الخاتمة محطة أخيرة لاستعراض أهم المراحل التي مر بها المشروع، والنتائج التي تم التوصل إليها، مع اقتراح بعض التوجهات المستقبلية لتطوير العمل.

2.6 تلخيص العمل المنجز

تناول مشروعنا تطوير تطبيق موبايل ذكي يسمح للمستخدم بتسجيل وتتبع مصاريفه اليومية بسهولة، اعتمادًا على تقنيات حديثة مثل Gemini OCR وخدمات Firebase السحابية. انطلقت الدراسة من تحليل الموجود، مرورًا بمرحلة النمذجة، إلى الإنجاز العملي الذي شمل بناء الواجهات، ربط التطبيق بخدمات الذكاء الاصطناعي وقواعد البيانات، وانتهاءً بالتجريب والتحليل.

3.6 الاستنتاجات العامة

أثبتت نتائج التقييم العملي أن التطبيق يحقق معظم الأهداف المسطرة بكفاءة عالية، حيث وفر للمستخدمين تجربة سلسة ودقيقة في تسجيل المصاريف، مع تقليل الحاجة للإدخال اليدوي بفضل تقنية OCR. كما ساهم دمج Firebase في ضمان سهولة حفظ البيانات والوصول إليها عبر مختلف الأجهزة.

4.6 آفاق مستقبلية

انطلاقًا من الملاحظات المسجلة أثناء التطوير والتجريب، نقترح التوجهات التالية لتطوير التطبيق:

- تحسين خوارزميات معالجة الصور لتحسين نتائج OCR.
- دعم لغات متعددة لاستخلاص البيانات من مختلف أنواع التذاكر.
- إضافة خاصية التخزين المؤقت والعمل بدون اتصال (Offline Mode).

- توسيع النظام ليشمل ميزات إضافية مثل إعداد ميزانيات وتتبع الدخل.

5.6 كلمة ختامية

ختامًا، يمثل هذا المشروع خطوة عملية أولى في مسار تطوير حلول ذكية تساعد الأفراد على تحسين تسييرهم المالي اليومي. نأمل أن يكون هذا العمل قاعدة لمشاريع مستقبلية أكثر تطورًا تخدم المجتمع بشكل أوسع.

المصادر

[1] Firebase Documentation. *Firebase Docs*, Google. Available at: https://firebase.google.com/docs

[2] Gemini AI OCR Documentation. *Gemini AI Docs*. Available at: https://cloud.google.com/vision

[3] Jetpack Compose Basics. *Android Developers*, Google. Available at: https://developer.android.com/jetpack/compose

[4] Ktor Framework Documentation. *Kotlin by JetBrains*. Available at: https://ktor.io/docs/

[5] M. Smith, A. Johnson. *A Survey on Optical Character Recognition Systems*. Journal of Pattern Recognition, 2020.

[6] A. Rahman. *Comparison of Personal Budget Management Mobile Apps*, ACM Digital Library, 2021.