الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة الشهيد حمه لخضر الوادي
كلية العلوم الدقيقة
قسم الاعلام الآلي

مذكرة نهاية التخرج

تدخل ضمن متطلبات الحصول على الشهادة

ليسانس أكاديمي

الميدان: رياضيات وإعلام آلي

الشعبة: الإعلام آلي

التخصص: نظم معلوماتية

إنشاء تطبيق اندرويد لتسيير ومراقبة النفقات الشخصية

من إعداد الطلبة:

- زتون هيثم
- بالمسعود يوسف الصديق
- ونيسي معتز

نوقشت يوم 2023/06/18 أمام اللجنة المكونة من الأساتذة:

عثماني سمير	أستاذ	رئيسا
بقاص منير	أستاذ	مناقشا
غربي قدور	أستاذ	مشرفا ومقررا

الموسم الجامعي: 2023/2022.

بسم الله الرحمن الرحيم

إهداء

قال تعالى:

"وقل اعملوا فسيرى الله عملكم ورسوله والمؤمنون"

إلهي لا يطيب الليل إلا بشكرك، ولا يطيب النهار إلا بطاعتك، ولا تطيب اللحظات إلا بذكرك، ولا تطيب الآخرة إلا بعفوك، ولا تطيب الجنة إلا برؤيتك جل جلالك.

إلى من بلغ الرسالة وأدى الأمانة نبي الرحمة ونور العالمين سيدنا محمد ﷺ

نحمد الله عز وجل لإتمام هذا البحث، وإلى الذي وهبنا كل ما يملك حتى نحقق له آماله، إلى من كان يدفعنا قدما نحو الأمام لنيل المبتغى، إلى الإنسان الذي امتلك الإنسانية بكل قوة، إلى الذي سهر على تعليمنا بتضحيات جسام مترجمة في تقديسه للعلم، آبائنا الأعزاء على قلوبنا، إلى مدرستنا الأولى في الحياة التي رعتنا، إلى التي صبرت على كل شيء، إلى التي وهبت فلذة كبدها كل العطاء والحنان حق الرعاية وكانت سندنا في الشدائد، وكانت دعواها لنا بالتوفيق، تتبعتنا خطوة بخطوة في عملنا، إلى من ارتحنا كلما تذكرنا ابتسامتها في وجهنا نبع الحنان أمهاتنا أعز ملاك على القلب والعين جزاها الله عنا خير الجزاء في الدارين، إليهما نهدي هذا العمل المتواضع لكي ندخل على قلبهما شيئا من السعادة، إلى إخوتنا وأخواتنا اللذين تقاسموا معنا عبء الحياة، كما نهدي ثمرة جهدنا لأستاذنا الكريم غربي قدور وإلى كل أساتذة قسم الإعلام الآلي، إلى كل من ساهم في هذا العمل من بعيد أو قريب وإلى من يؤمن بأن بذور نجاح التغيير هي في ذواتنا وفي أنفسنا قبل أن تكون في أشياء أخرى.

عن أبي هريرة رضي الله عنه قال سمعت رسول الله صلى الله عليه وسلم يقول:

"من سلك طريقا يلتمس فيه علما سهل الله له به طريقا إلى الجنة"

رواه مسلم

شكـــــر وتقدير

أول مشكور هو الله عز وجل، ثم الوالدان على كل مجهوداتهم منذ ولادتنا إلى هذه اللحظات، أنتم كل شيء لكم منا كل الحب.

يسرنا أن نوجه شكرنا لكل من نصحنا وأرشدنا أو وجهنا أو ساهم معنا في إعداد هذا البحث بإيصالنا للمراجع والمصادر المطلوب في أي مرحلة من مراحله، ونشكر على وجه الخصوص أستاذنا الفاضل غربي قدور على مساندتنا وإرشادنا بالنصح والتصحيح، كما أن شكرنا موجه لإدارة كلية العلوم الدقيقة بجامعة حمه لخضر قسم الإعلام الآلي، وبعدها الشكر موصول لكل أساتذتنا اللذين تتلمذنا على أيديهم في كل مراحل دراستنا حتى تشرفنا بوقوفنا أمام حضراتكم اليوم.

مخلص

ملخص

وصل التطور التكنولوجي عالميا مستويات غير مسبوقة، حيث أصبحت التكنولوجيا تمثل العمود الفقري لحياة الناس ومن بينها تطبيقات الهواتف الذكية ومن خلال هذه الأهمية التي تمثلها هذه التطبيقات تأتي أهمية مشروعنا المتمثل في برنامج محمول لتسيير النفقات الشخصية.

ومن خلال هذا المشروع سيتم معالجة المشاكل التي تواجه المستخدم يوميا حيث سيقدم البرنامج الحلول الملائمة للمستخدم العادي في بيئتنا البسيطة بعيدا عن تعقيد البرامج العالمية والإقليمية الموجهة لفئات تختلف اهتماماتها عن اهتمامات بيئتنا.

وقد استعننا في انجاز هذا المشروع بمحرر Star UML لتصميم البرامج وبيئة فلاتر لبرمجة المشروع.

Abstract

The technological advancement has reached unprecedented levels worldwide, where technology has become the backbone of people's lives, including smartphone applications.

Through the significance of these applications, the importance of our project emerges, which is represented by a mobile program for personal expense management.

Through this project, we aim to address the daily challenges faced by users by providing suitable solutions for the average user in our simple environment, away from the complexity of global and regional programs that target different interests than our environment.

To accomplish this project, we have utilized the Star UML editor for program design and the Flutter environment for project programming.

Résumé

Le développement technologique a atteint des niveaux sans précédent dans le monde entier, où la technologie est devenue l'épine dorsale de la vie des gens, notamment grâce aux applications pour smartphones.

À travers l'importance de ces applications, l'importance de notre projet se manifeste, qui consiste en un programme mobile de gestion des dépenses personnelles.

Grâce à ce projet, nous visons à résoudre les défis quotidiens auxquels les utilisateurs sont confrontés en proposant des solutions adaptées à l'utilisateur moyen dans notre environnement simple, loin de la complexité des programmes mondiaux et régionaux qui ciblent des intérêts différents de notre environnement.

Pour réaliser ce projet, nous avons utilisé l'éditeur Star UML pour la conception du programme et l'environnement Flutter pour la programmation du projet.

الفهرس

إهداء.....
شكـر وتقدير.....
ملخص.....
الفهرس.....
قائمة الصور .....
قائمة المخططات.....
مقدمة عامة.....

الفصل الأول: دراسة الموجود

1- مقدمة..... 4
2- تطبيق إدارة النفقات الشخصية..... 4
3- التطبيقات المتشابهة..... 4
3-1- المميزات..... 4
3-2- العيوب 5
4-الحل المقترح..... 5
5- خاتمة..... 5

الفصل الثاني: نموذج وتصميم التطبيق

1-مقدمة..... 7
2-تقديم الطريقة المستعملة..... 7
2-1 تعريف لغة النمذجة الموحدةUML..... 7
2-2 تصنيف مخططات UML 7
2-3 شرح المخططات المستعملة..... 8

3- مخططات UML..... 9
3-1 مخطط حالات الاستخدام العام Use Case Diagram..... 9
3-2 مخطط التتابع Sequence Diagram.....10
3-3 مخطط النشاط Activity Diagram.....12
3-4 مخطط الفئات Class Diagram.....13
4- خاتمة.....13

الفصل الثالث: التنفيذ

1- مقدمة.....15
2- التعريف ببيئة العمل.....15
2-1 الوسائل المادية.....15
2-2 الوسائل البرمجية 16
2- الانجاز 19
2-1 تطبيق android الخاص بالمشروع.....19
3- خاتمة.....36
خاتمة عامة 37

قائمة الصور

صورة1 Dependencies of Project 18
صورة 2 Programing code of chose the local Currency 20
صورة 3 Programing code of add current balance 21
صورة 4 Programing code of start now 22
صورة 5 Start now user interface 23
صورة 6 Programing code of transaction 24
صورة 7 home page user interface 25
صورة 8 Programing code of add amount 27
صورة 9 Add transaction user interface 28
صورة 10 Programing code of wallet 29
صورة 11 The wallet user interface 30
صورة 12 Programing code of user profile 31
صورة 13 User profile interface 32
صورة 14 Programing code of edit profile 33
صورة 15 Edit profile interface 34

قائمة المخططات

مخطط 1 Use Case Diagram..... 9
مخطط 2 Séquence Diagram.....10
مخطط 3 Séquence Diagram.....11
مخطط 4 Activity Diagram.....12
مخطط 5 Class Diagram 13

مقدمة عامة

مقدمة عامة

تعتبر التكنولوجيا الأكثر تأثيرا في حياتنا اليومية حيث ارتقت في شتى المجالات من مستوى الاستغلال التصوري البسيط إلى المستوى التطبيقي الآلي وذلك نظرا لما يقدمه من الخدمات التي تتميز بالدقة وسرعة معالجة المعلومات، وهذا الارتقاء توافق طرديا وبشكل ملحوظ مع زيادة الطلب على البرمجيات والخدمات الإلكترونية بأنواعها، أصبحت حاجة الإنسان الى العديد من التطبيقات والتي من بينها تطبيق تسيير ومراقبة النفقات الشخصية، والذي يتمثل في تسهيل المعاملات المالية بأفضل طريقة لتوفير المزيد من الأموال وإدارة المصاريف ويكمن الهدف من ذلك في اتخاذ القرارات التي لا تضطر إلى إضاعة الكثير من الوقت والجهد في إدارة الأموال.

نظرا للجهود المبذولة لكسب العديد من الأموال، وعند الادخار نرى أن جزءا كبيرا منها قد ضاع لعدم تنظيم أولويات مثل:(أساسي– عادي– ترفيهي)، وعدم معرفة المعاملات اليومية والاسبوعية والشهرية، وكذلك الديون المدفوعة والمستلمة، لهذا فإن فكرة التطبيق تقوم بحلول جميع المشاكل المقترحة أي انشاء تطبيق أندرويد يقوم بتسيير ومراقبة النفقات الشخصية.

فهل من الممكن تطوير تطبيق سهل الاستعمال يوفر هذه المهام؟

الفصل الأول:

دراسة الموجود

1- مقدمة

تعتبر دراسة الموجود المرحلة الأولى من هذا البحث، إلى أنها تهدف إلى جمع أكبر قدر ممكن من المعلومات حول النظام المدروس، ومن خلالها نتوصل إلى فهم دقيق للواقع نفسه، ونستطيع بواسطتها القيام بحل تنظيمي وإنشاء تصوري مستقبلي للنظام. وتعتمد هذه الدراسة أساسا على جملة من التحقيقات الميدانية والمجسدة في الحوارات مع المسؤولين والمعنيين في تسيير حقل الدراسة، حتى نتمكن من الإلمام بكافة جوانبه، وبالتالي ضمان السير الحسن لعملية النقل من الواقع إلى النموذج التصوري.

2- تطبيق إدارة النفقات الشخصية

تعريفه:

هو عبارة عن تطبيق يساعد على إدارة النفقات وإعداد ميزانية بشكل فعال، ويسهل على المستخدم تتبع الإنفاق والادخار، وكذلك يساعد على إيجاد طرق للحد من الإنفاق لتوفير الأموال والبقاء على الاطلاع الدائم عما أنفقت أموالك.

3- التطبيقات المتشابهة

3-1- المميزات

- الدعم اللغوي.
- النسخ الاحتياطي.
- الإشعارات.
- الأمان.

3-2- العيوب

- السعر: يتطلب دفع مبلغ مالي لشراء التطبيق كاملا.
- تصميم معقد.
- صعوبة وتعقيد التطبيقات الموجودة في السوق.
- الواجهة.
- الإعلانات.
- غير متوفر لجميع بلدان العالم.

4-الحل المقترح

من أجل تسهيل عملية إدارة المصاريف لدى المستخدمين، وللحد من السلبيات والعراقيل التي سبق عرضها، لذلك قمنا بإنشاء تطبيق هاتف مبسط مع قابلية التحديث، بحيث يستطيع جميع الأشخاص استعماله بطريقة سلسة وسهلة دون صعوبات وعراقيل، حيث يوفر التطبيق عدة خدمات كإمكانية معرفة تفاصيل المعاملات وإدارتها، وكذا عرض قائمة محفظة المستخدم، كما يسمح للمستخدم إضافة معاملات جديدة في أي قطاع كان.

5- خاتمة

بعد مجموعة من الأبحاث التي قمنا بها من بعض المواقع والمراجع أنجزنا هذا الجزء الذي عرفنا فيه خصائص ومحاسن ومساوئ مختلف التطبيقات المتوفرة في الوقت الراهن، بالإضافة إلى الصعوبات التي يعاني منها المستخدم في التعامل معها، ومن هذه الدراسة اقترحنا الحلول اللازمة.

الفصل الثاني:

نموذج وتصميم التطبيق

1- مقدمة

بعد تحليل ودراسة الاحتياجات وتحديد مختلف العمليات اللازمة لإدارة المصاريف المنزلية.

سيشمل هذا الفصل وصف مهام التطبيق المختلفة باستخدام لغة النمذجة الموحدة UML حيث يتضمن الدراسة التفصيلية من خلال انجاز المخططات اللازمة لوصف العمليات والمهام على مستوى التطبيق والمستخدم.

2-تقديم الطريقة المستعملة

2-1 تعريف لغة النمذجة الموحدةUML

لغة النمذجة الموحدة (Language Modelling Unified) او ((UML) هي لغة تحليل وتصميم متخصصة في مجال هندسة البرمجيات تهدف الى توفير طرق وآليات منهجيه تساعد على تصور انظمة البرمجيات، وتتجه لغة UML بطبيعتها الخاصة في بناء أنظمة البرمجيات كائنية المنحنى (Object Oriented).

في الواقع، هي عبارة عن الآراء المختلفة من اشخاص مختلفين. بمعنى ان لغة النمذجة هذه جاءت نتيجة للآراء المختلفة التي لدى الاشخاص حول أفضل منهجية لبناء برامج ناجحة وفعالة.

2-2 تصنيف مخططات UML

تنقسم مخططات UML إلى قسمين اساسين:

- **المخططات الهيكلية: Diagrams Structure**

هي مجموعة من المخططات التي تظهر الهيكل العام للتطبيق على عدة مستويات وكيف ان كل هذه المستويات مرتبطة ببعضها البعض.

❖ **المخططات السلوكية: Diagrams Behavior**

هي مجموعة من المخططات التي تظهر السلوك الديناميكي الذي تسلكه مختلف اجزاء التطبيق، يتكون هذا الصنف من عدة مخططات " Diagrams" مثل:

(Use Case, Sequence, Activity, Timing … etc.)

2-3 شرح المخططات المستعملة

❖ **مخطط حالات الاستخدام Use Cas Diagram:**

هو مخطط لتحليل وصف سلوك التطبيق من وجهة نظر المستخدم، وهو ذو فائدة خلال مراحل التحليل والتطوير.

❖ **مخطط التتابع Sequence Diagram:**

هو مخطط يوضح كيفية تفاعل الأجواء المختلفة من التطبيق مع بعضهم البعض لتنفيذ كل حالة من الحالات استخدام، ويظهر أجزاء مختلفة من التطبيق تعمل في "تتابع"(بطريقة تسلسلية) من أجل انجاز وظيفة محددة.

❖ **مخطط النشاط Activity Diagram:**

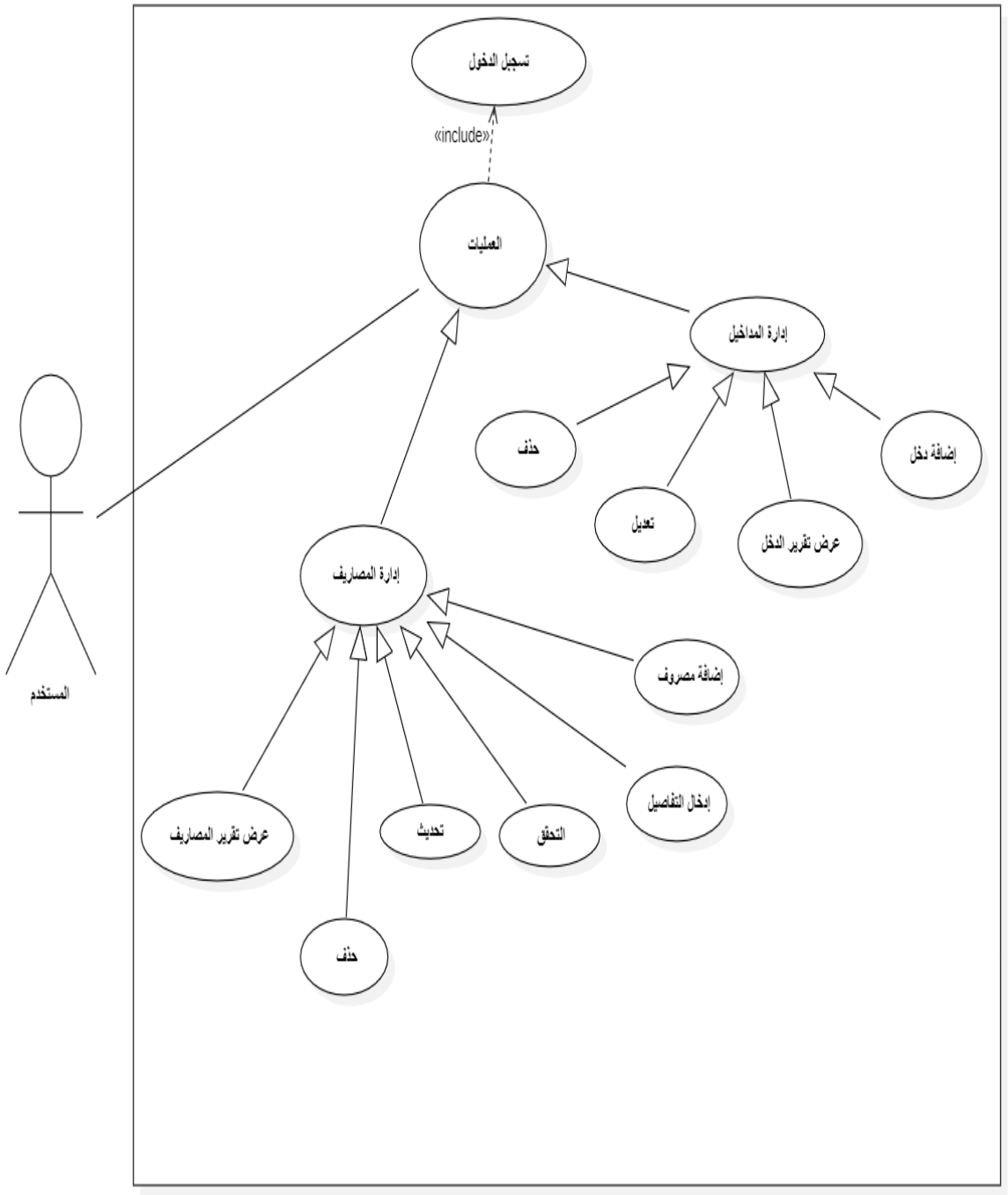
هو مخطط انسيابي لتمثيل التدفق من نشاط إلى اخر، حيث يمكن وصف النشاط كعملية للنظام.

❖ **مخطط الفئات Classes Diagram :**

هو مخطط يساعد في استخراج جميع الفئات المندرجة داخل النظام الواحد، ويشرح انواع العلاقات التي تربط بين مختلف هذه الفئات، كما يوضح الوظائف والسمات التي تتمتع بها كل فئة.

3- مخططات UML

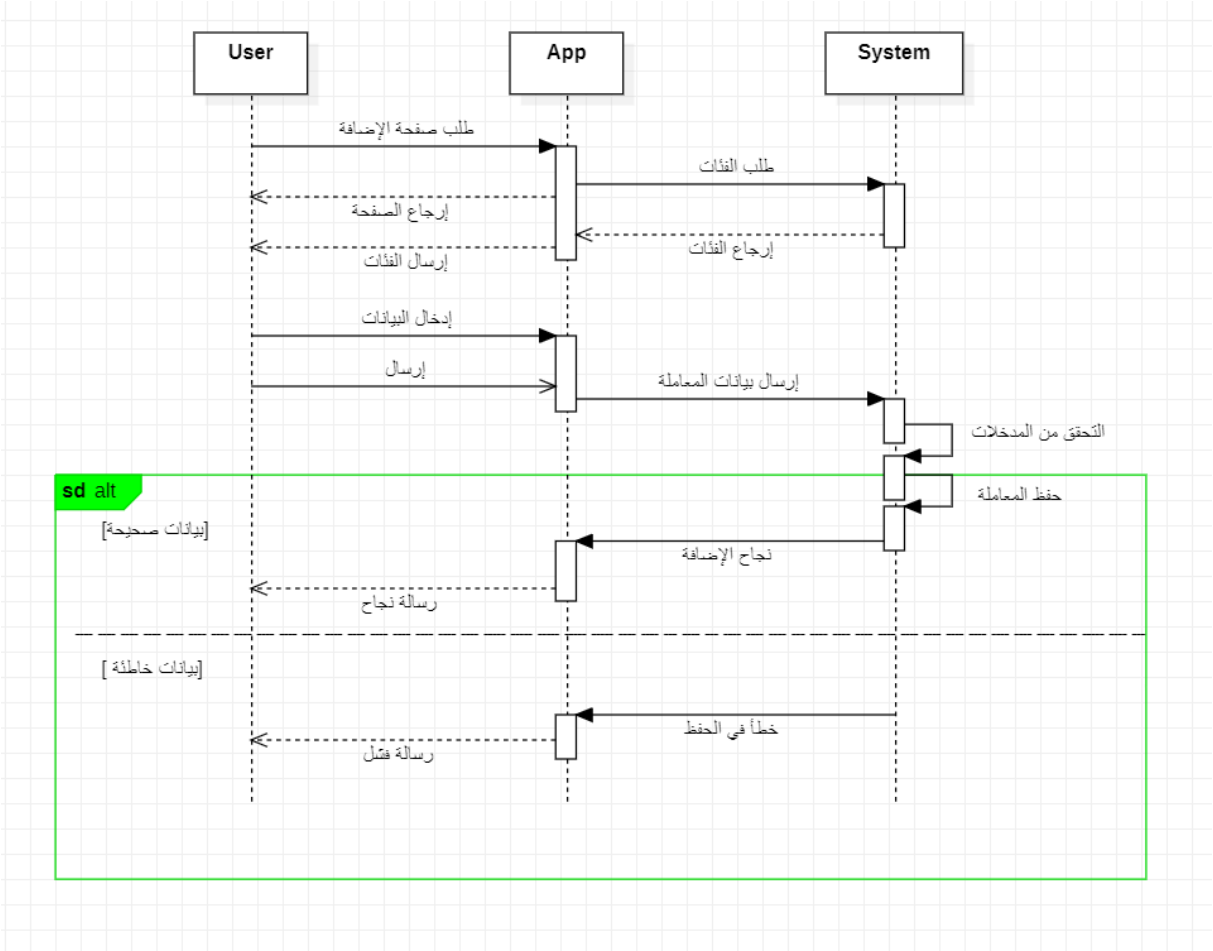
3-1 مخطط حالات الاستخدام العام Use Case Diagram



مخطط 1 Use Case Diagram

3-2 مخطط التتابع Séquence Diagram

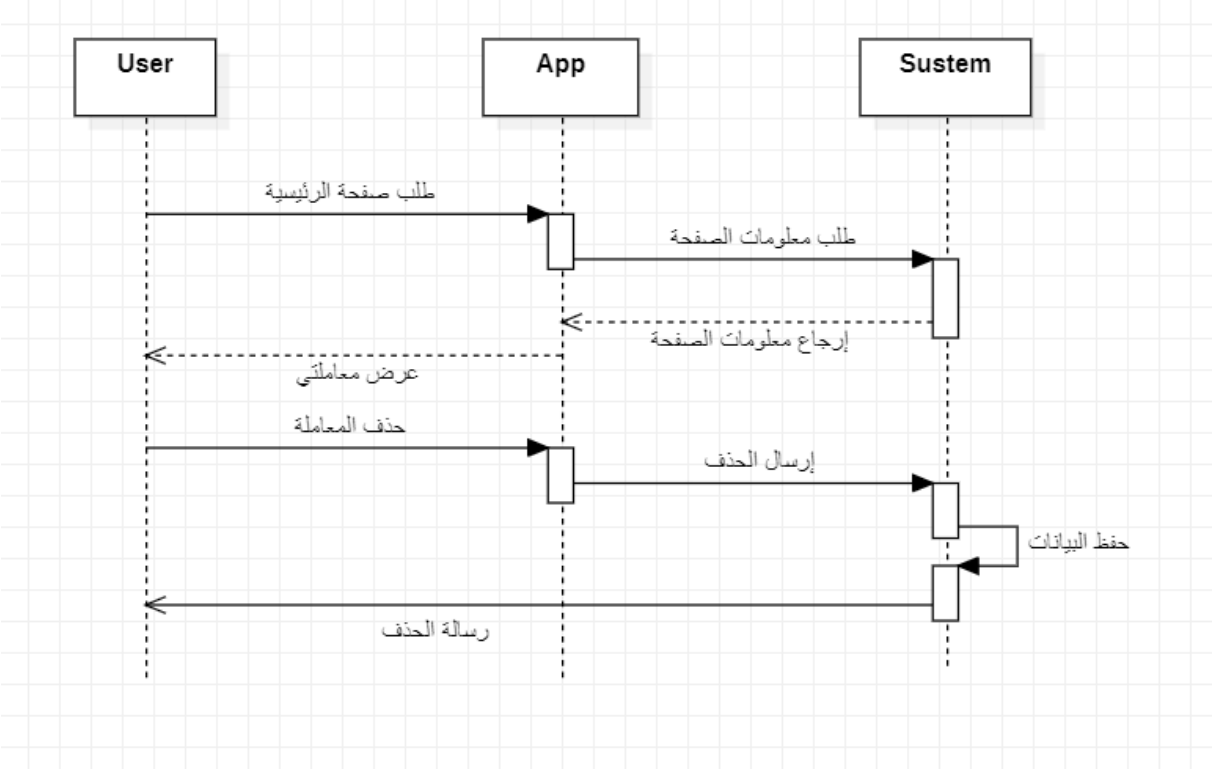
3-2-1 مخطط التتابع إضافة معاملة



مخطط 2 Séquence Diagram

من خلال مخطط (2) بيّنا طريقة تفاعل التطبيق والنظام مع المستخدم أثناء عملية الإضافة انطلاقا من اختيار صفحة المعاملة ثم الإضافة وذلك في حالتين إذا كان المستخدم قيد تسجيل الدخول أو ليس قيد تسجيل الدخول.

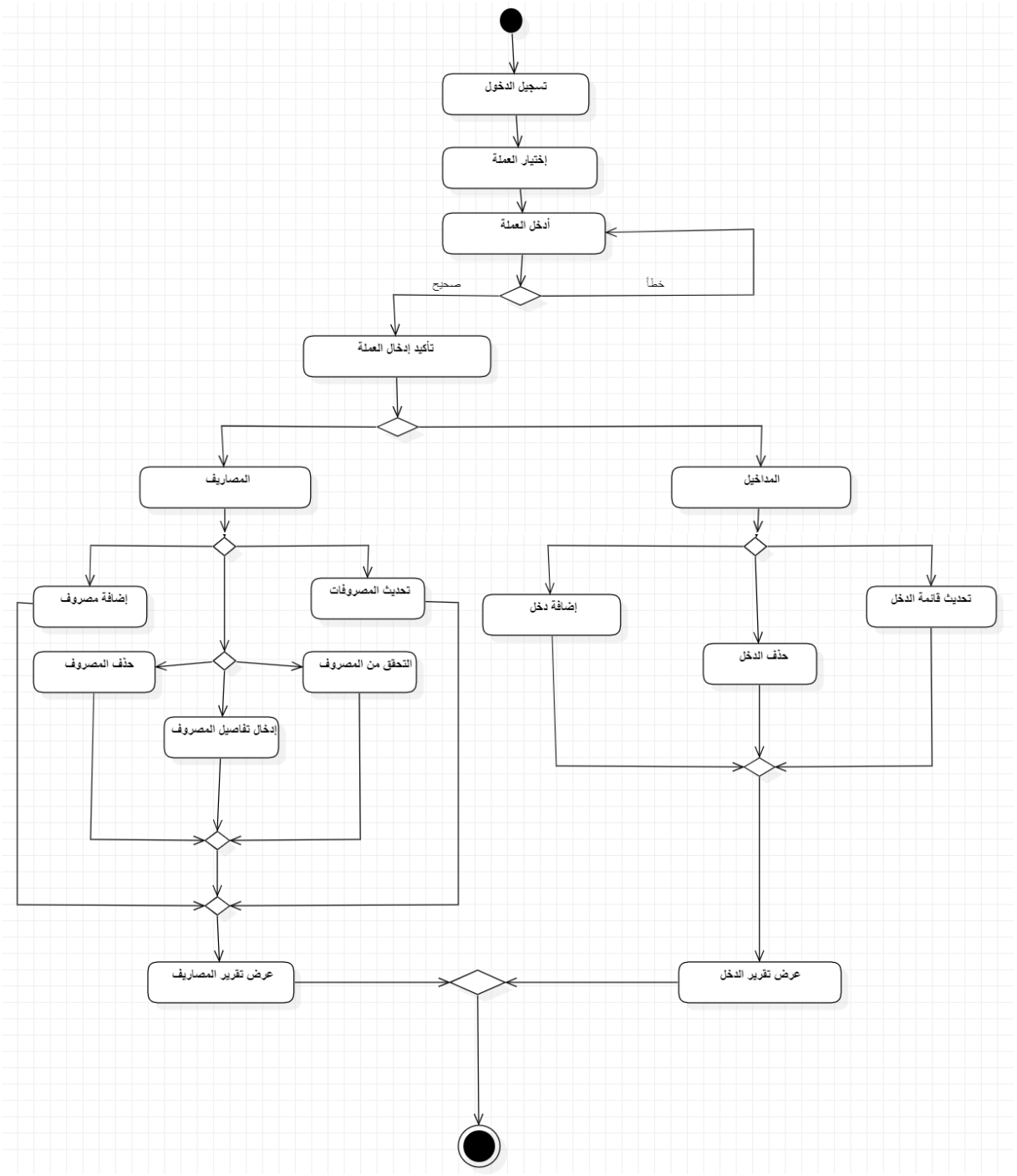
3-2-2 مخطط التتابع لحذف معاملة



مخطط 3 Séquence Diagram

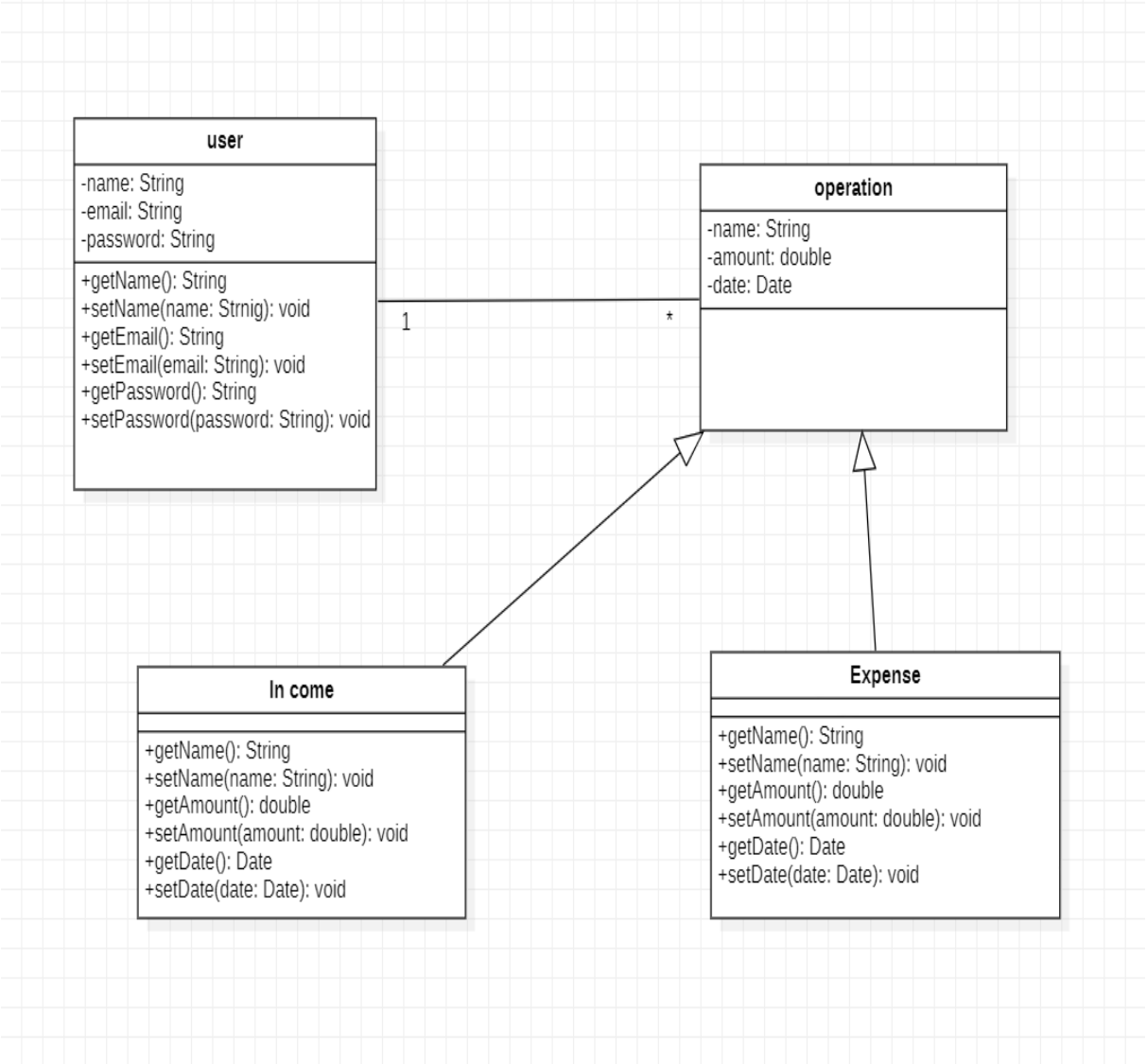
من خلال مخطط (3) بيّنا طريقة تفاعل التطبيق والنظام مع المستخدم أثناء عملية حذف المعاملة وذلك بطلبه لصفحة معاملاته ومن ثم يحذف معاملته.

3-3 مخطط النشاط Activity Diagram



مخطط 4 Activity Diagram

4-3 مخطط الفئات Class Diagram



مخطط 5 Class Diagram

4- خاتمة

مع الانتهاء من هذا الفصل وانجاز المخططات اللازمة (Use Cas Diagram- Classes Diagram-Activity Diagram) أصبح النموذج التصوري للمعطيات واضح المعالم ومن خلاله أمكننا وضع تصور كافي عن التطبيق والشروع في انجازه.

الفصل الثالث:

التنفيذ

1- مقدمة

بعد إتمام العمل النظري ومرحلة التصميم، نأتي الي مرحلة التجسيد العملي للمشروع، في هذا الفصل.

سيتم التعريف بالأدوات المستخدمة ولغات البرمجة الخاصة المطلوبة في عملية الانجاز والتنفيذ وبيئة التطوير المعتمدة وكذا شرح لمختلف مهام التطبيق.

2- التعريف ببيئة العمل

تم تطوير هذا التطبيق الذي يعمل بنظام Android في بيئة تطوير التطبيقات المجانية Visual Studio Code باستخدام لغة البرمجة Dart المهيئة من قبل منصة Flutter

2-1 الوسائل المادية

للبيئة التجريبية استعملنا هاتف ذكي يعمل على نظام التشغيل Androidوجهاز حاسوب محمول HP يعمل على نظام Windows لتنصيب بيئة العمل وكابل USB Drive لتوصيل بينهما.

2-2 الوسائل البرمجية

❖ **Android Studio (V-11.0.15)**

هي منصة تطوير تطبيقات الاندرويد تسهل على المطورين كتابة الشيفرة بشكل فوري لتطبيقات اندرويد وتمتاز بسهولة الاستخدام وكما تعطي معاينة فورية لمختلف أنواع الأجهزة اللوحية او المحمولة ومختص للبرمجة.

❖ **Android**

Android هو نظام تشغيل مفتوح المصدر مطور من قبل شركة Google استنادا للنسخة المعدلة من توزيعة Linux Kernel 1.6 من اجل استخدامه للهواتف التي تعمل باللمس والأجهزة المحمولة، والأجهزة اللوحية كما تم توظيفه حديثا ليناسب أجهزة التلفاز الذكية السيارات وساعات اليد.

❖ **Visual Studio Code(v-1.78.2)**

بيئة تطوير متكاملة integrated development environment (IDE) أنشأتها شركة Microsoft وقدمتها لأول مرة في 29 ابريل 2015.

وهي مجانية صغيرة الحجم ومفتوحة المصدر تعمل على نظام التشغيل ويندوز وماك أو إس ولينكس. تم تطوير code VS باستخدام إطار العمل إلكترون(electron)

وNode.JS،Runtime ، يدعم بشكل أساسي العديد من اللغات البرمجية المتنوعة حيث إنه لا يقتصر على مجال محدد من اللغات المدعومة لغة C و C++ كما وأنه يدعم لغات الويب الشهير(HTML) .

❖ **Flutter(v-3.10.2)**

هي برنامج مفتوح المصدر لمجموعة أدوات تطوير البرمجيات تم تطويرها من قبل جوجل لبناء واجهات المستخدم لتطبيقات نظام الأندرويد ونظام iOS ونظام الويندوز وتطبيقات الويب.

- من مكونات إطار فلاتر الرئيسية:

لغة برمجة Dart: لغة برمجة مصممة لتطوير تطبيقات الويب وتطبيقات الاندرويد والـiOS من تطوير شركة جوجل والتي تستهدف فيها مطوري الويب وتطبيقات الهاتف.

محرك فلاترFlutter SDK .

المكتبات الأساسية.

```yaml
dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.2
  get: ^4.6.5
  intl:
  hive_flutter: ^1.1.0
  dropdown_button2: ^2.1.0
  iconsax:
  share_plus: ^7.0.0
  image_picker:
  path_provider: ^2.0.15
  flutter_simple_calculator:
  url_launcher:
  flutter_secure_storage: ^8.0.0
  flutter_localization: ^0.1.9
  collection: ^1.15.0
dev_dependencies:
  flutter_test:
    sdk: flutter
  flutter_lints: ^2.0.0
  hive_generator: ^2.0.0
  build_runner: ^2.3.3
```

صورة1 Dependencies of Project

❖ **StarUML**

يعتبر هذا البرنامج أفضل حل برمجي وعملي موثوق به وعملي مصمم خصيصا للمستخدمين الذين يحتاجون الى انشاء رسوم بيانية لتحليل تدفق البيانات ويستخدم كتطبيق نمذجة UML مصمم لتسهيل عمل مطوري البرامج.

2- الانجاز

2-1 تطبيق android الخاص بالمشروع

يدعم تطبيق مصروفك طريقة واحدة للمصادقة (اختيار العملة المحلية – إضافة الرصيد الحالي – إضافة الدخل الشهري)

I- المصادقة

1-إختيار العملة المحلية

كود البرمجة:

```
      children: [
        Icon(Icons.attach_money),
        Text(
          'local currency'.tr,
          style: Themes.Theme3,
        ), // Text
      ],
    ), // Row
    SizedBox(
      height: Get.height * 0.03,
    ), // SizedBox
    DropdownButtonFormField(
      hint: Text(
        'choose local currency'.tr,
        style: Themes.Theme4,
      ), // Text
      decoration: const InputDecoration(
        filled: true,
        fillColor: Color.fromARGB(255, 229, 234, 238),
        contentPadding: const EdgeInsets.only(
            top: 13, bottom: 13, right: 10, left: 10), // EdgeInsets.only
        hintStyle:
            TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
        focusedBorder: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.white),
          borderRadius: BorderRadius.all(Radius.circular(10.0)),
        ), // OutlineInputBorder
        enabledBorder: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.white),
          borderRadius: BorderRadius.all(Radius.circular(10.0)),
        ), // OutlineInputBorder
      ), // InputDecoration
```

صورة 2 Programing code of chose the local Currency

2-إضافة الرصيد المحلي

كود البرمجة:

```
text_field(
  hint: 'current balance'.tr,
  controller: currentBalanceController,
  typekey: TextInputType.number,
), // text_field
SizedBox(
  height: Get.height * 0.03,
), // SizedBox
Row(
  children: [
    Icon(Icons.attach_money),
    Text(
      'قيمة الراتب بالعملة المحلية'.tr,
      style: Themes.Theme3,
    ), // Text
  ],
), // Row
SizedBox(
  height: Get.height * 0.02,
), // SizedBox
```

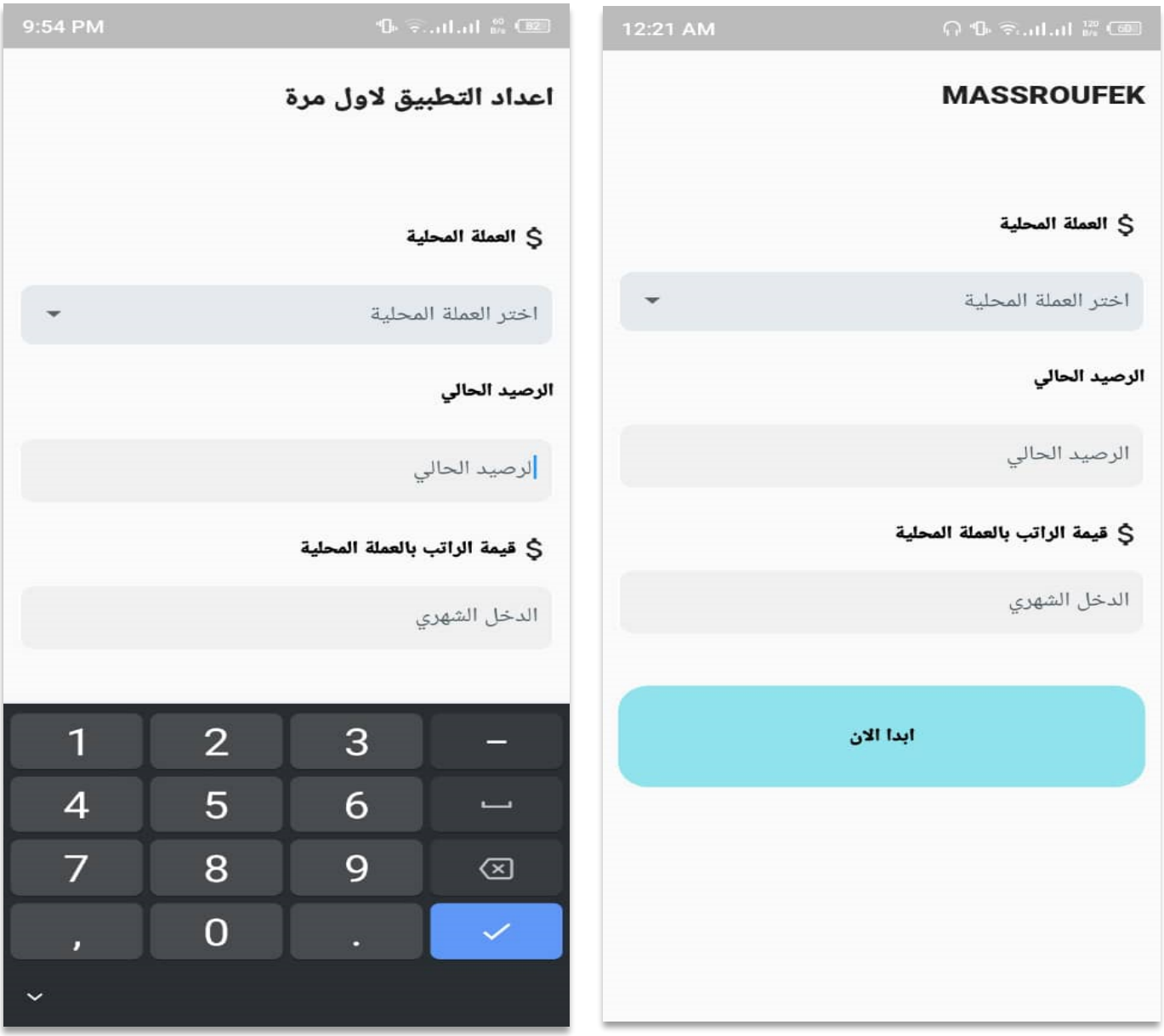
صورة 3 Programing code of add current balance

زر ابدأ الان:

```
customButton(
  height: Get.height * 0.1,
  width: Get.width,
  color: PrimaryColorBlue,
  raduis: 20,
  hint: 'start now'.tr,
  ontap: () {
    if (currentBalanceController.text == '' &&
        MonthlyIncomeController.text == '') {
      Get.snackbar('خطأ'.tr, 'The fields are empty'.tr,
          backgroundColor: PrimaryColorBlue);
    } else {
      controller.information_setup(
          currentBalanceController.text,
          MonthlyIncomeController.text);

      Get.offAllNamed(Routes.MainPageView);
    }
  },
) // customButton
```

صورة 4 Programing code of start now

واجهة المستخدم:



صورة 5 Start now user interface

II- الصفحة الرئيسية

كود البرمجة:

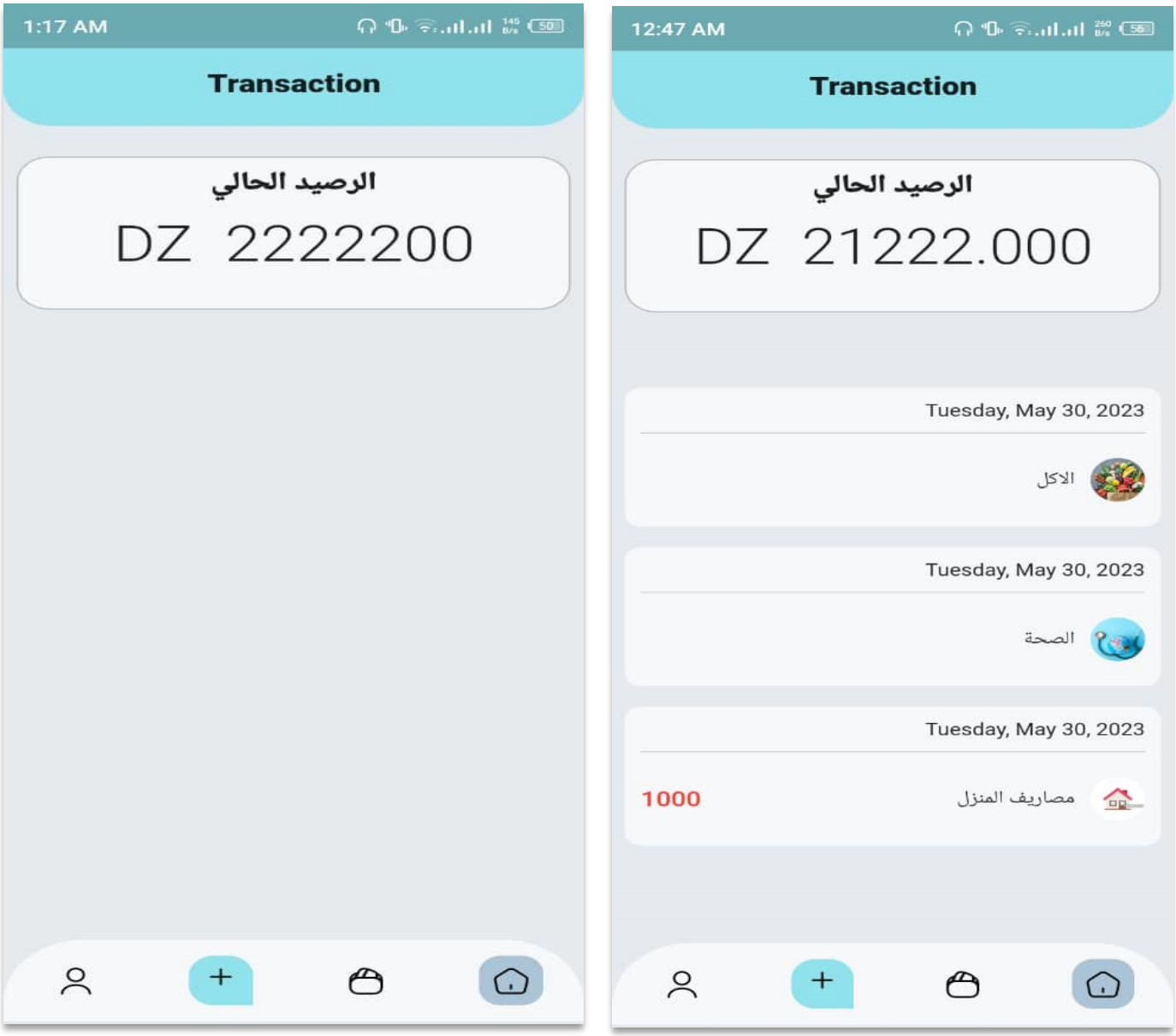
```dart
class Homepageview extends StatelessWidget {
  const Homepageview({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      //backgroundColor: colorfield,
      body: Container(
        color: colorfield,
        height: Get.height,
        width: Get.width,
        child: Column(
          children: [
            Container(
              padding: EdgeInsets.only(top: 30),
              decoration: BoxDecoration(
                color: PrimaryColorBlue,
                borderRadius: BorderRadius.only(
                    bottomLeft: Radius.circular(30),
                    bottomRight: Radius.circular(30)), // BorderRadius.only
              ), // BoxDecoration
              height: Get.height * 0.12,
              width: Get.width,
              child: Center(
                child: Text(
                  'Transaction'.tr,
                  style: Themes.Theme1,
                ), // Text
              ), // Center
            ), // Container
            SizedBox(
              height: Get.height * 0.02,
            ), // SizedBox
```

```dart
Container(
  margin: EdgeInsets.all(Get.height * 0.01),
  padding: EdgeInsets.all(4),
  height: Get.height * 0.15,
  width: Get.width,
  decoration: BoxDecoration(
    color: colorscreen,
    borderRadius: BorderRadius.circular(20),
    border: Border.all(color: Colors.black26),
  ), // BoxDecoration
  child: GetBuilder<Homepagecontroller>(builder: (context) {
    return Column(
      children: [
        Text(
          'current balance'.tr,
          style: Themes.Theme1,
        ), // Text
        SizedBox(
          height: Get.height * 0.01,
        ), // SizedBox
        Text(
          context.currency + '00  DZ',
          style:
              TextStyle(fontSize: 40, fontWeight: FontWeight.w300),
        ) // Text
      ],
    ); // Column
  }), // GetBuilder
), // Container
```

Programing code of transaction 6 صورة

واجهة المستخدم:

هناك واجهتان رئيسيتان لهما نفس تصميم واجهة المستخدم.



صورة 7 home page user interface

1- إضافة معاملة:

كود البرمجة:

```
class Add_transaction_view extends StatelessWidget {
  const Add_transaction_view({super.key});

  @override
  Widget build(BuildContext context) {
    TextEditingController amountController = TextEditingController();
    TextEditingController NotesController = TextEditingController();
    CalcController calcController = CalcController();

    return Scaffold(
      body: GetBuilder<Add_transaction_controller>(
       init: Add_transaction_controller(),
        builder: (controller) {
          return Container(
            height: Get.height,
            width: Get.width,
            child: SingleChildScrollView(
              child: Column(
                children: [
                  Container(
                    padding: EdgeInsets.only(top: 30),
                    decoration: BoxDecoration(
                      color: PrimaryColorBlue,
                      borderRadius: BorderRadius.only(
                          bottomLeft: Radius.circular(30),
                          bottomRight: Radius.circular(30)), // BorderRadius.only
                    ), // BoxDecoration
                    height: Get.height * 0.12,
                    width: Get.width,
                    child: Center(
                      child: Text(
                        'اضافة معاملة'.tr,
                       style: Themes.Theme1,
                      ), // Text
                    ), // Center
                  ), // Container
```

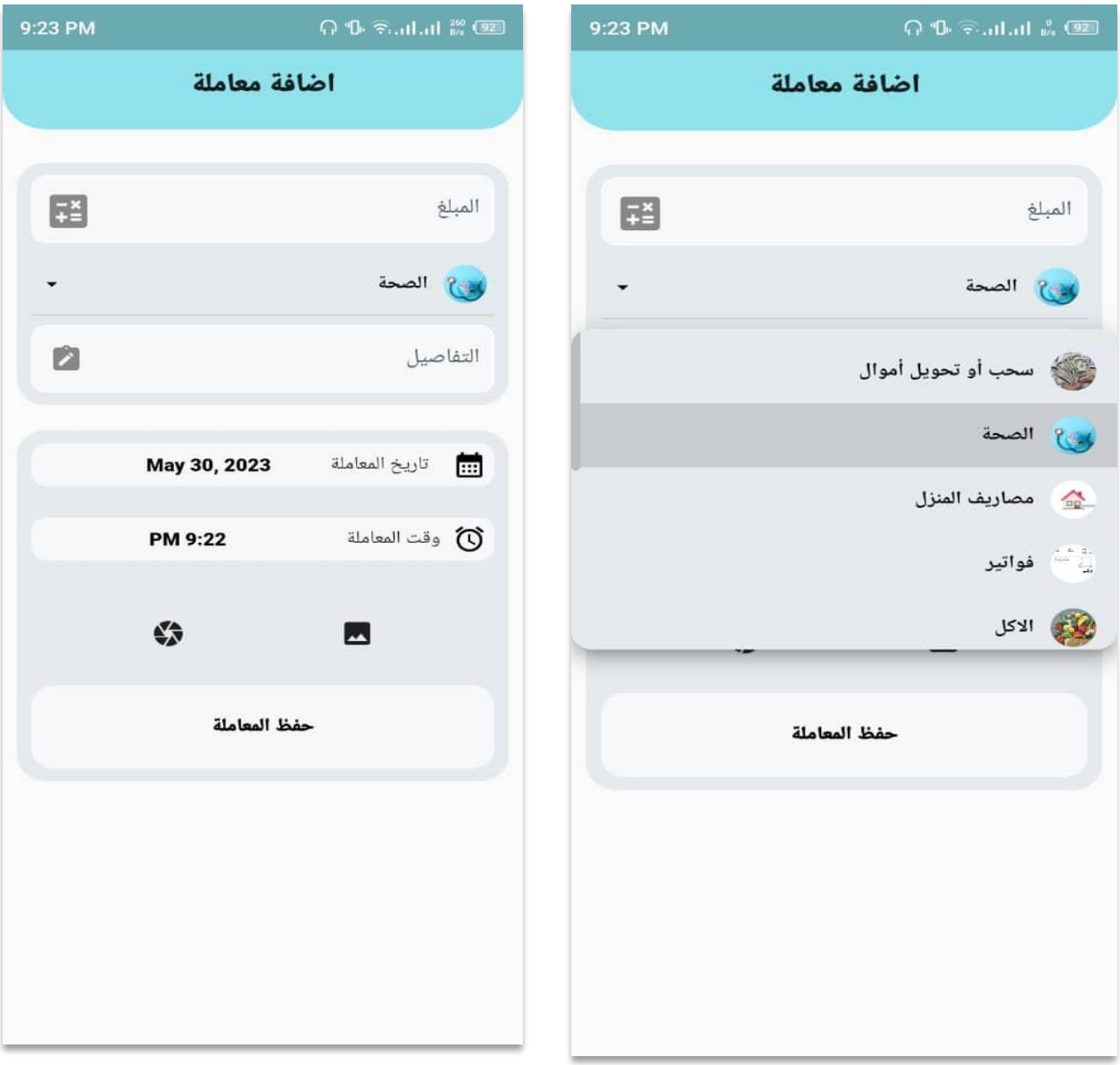
```
Container(
  margin: EdgeInsets.all(10),
  padding: EdgeInsets.all(10),
  decoration: BoxDecoration(
      color: colorfield, borderRadius: BorderRadius.circular(20)), // BoxDecoration
  child: Column(
    children: [
      text_field(
        typekey: TextInputType.number,
        color: colorscreen,
        hint: 'المبلغ'.tr,
        controller: amountController,
        icon: IconButton(
          onPressed: () => Get.bottomSheet(BottomSheet(
              onClosing: () {},
              builder: (BuildContext) {
                return SimpleCalculator(
                  controller: calcController,
                  value: 0,
                  hideExpression: true,
                  onChanged: (key, value, expression) {
                    /*...*/
                    if (calcController.value != 0.0) {
                      amountController.text =
                          calcController.value.toString();
                    }
                  },
                  theme: const CalculatorThemeData(
                    displayColor: Colors.black,
                    displayStyle: const TextStyle(
                        fontSize: 80, color: Colors.yellow), // TextStyle
                  ), // CalculatorThemeData
                ); // SimpleCalculator
              })), // BottomSheet
```

Programing code of add amount 8 صورة

واجهة المستخدم:

لإضافة معاملة جديدة، يمكن الضغط على علامة الزائد أيقونة تقوم بإظهار صندوق العملية.

يؤدي لمس زر حفظ المعاملة الى نقل البيانات.



صورة 9 Add transaction user interface

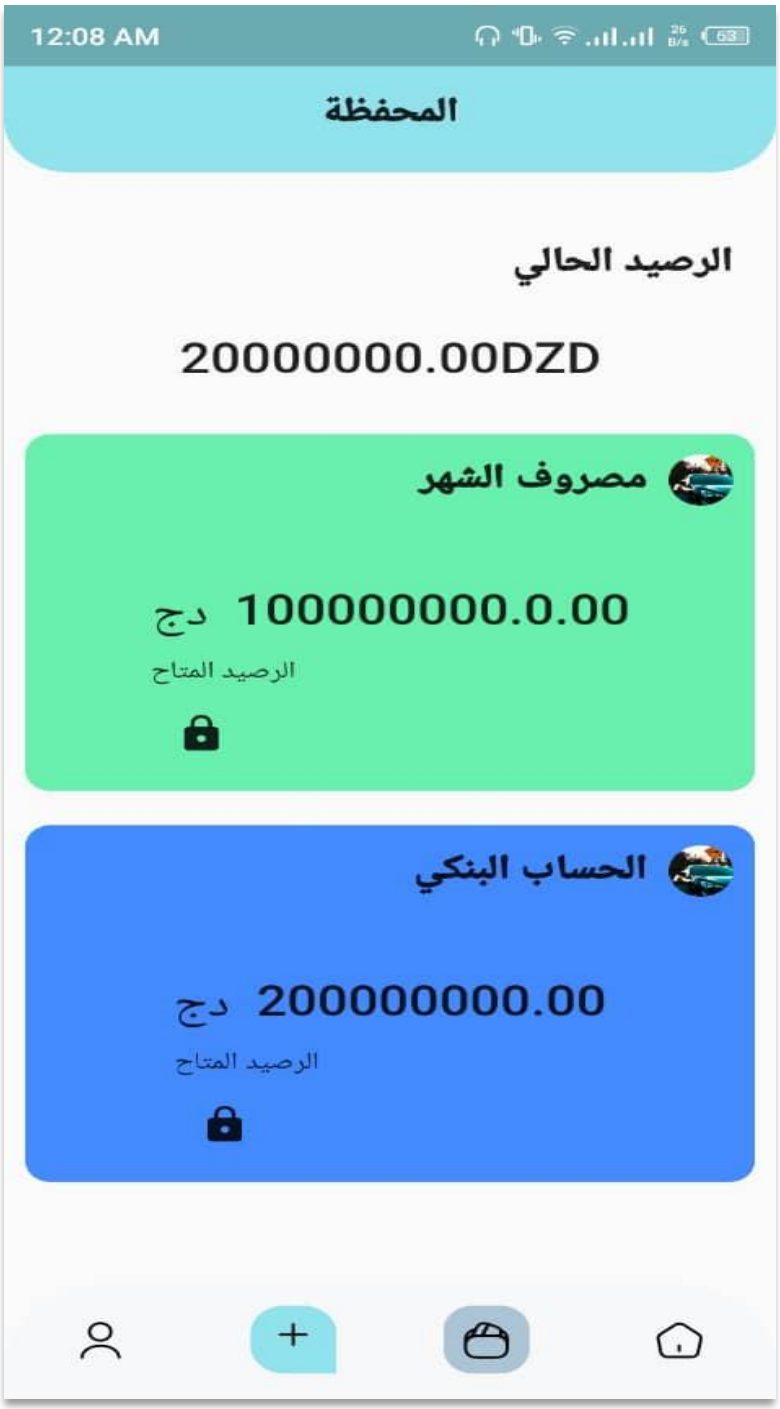
2- المحفظة:

كود البرمجة:

```
class money_view extends GetView<money_controller> {
  const money_view({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      //backgroundColor: colorfield,
      body: Container(
        // color: colorfield,
        height: Get.height,
        width: Get.width,
        child: Obx( () {
          return Column(
            children: [
              Container(
                padding: EdgeInsets.only(top: 30),
                decoration: BoxDecoration(
                  color: PrimaryColorBlue,
                  borderRadius: BorderRadius.only(
                      bottomLeft: Radius.circular(30),
                      bottomRight: Radius.circular(30)), // BorderRadius.only
                ), // BoxDecoration
                height: Get.height * 0.12,
                width: Get.width,
                child: Center(
                  child: Text(
                    'المحفظة'.tr,
                    style: Themes.Theme1,
                  ), // Text
                ), // Center
              ), // Container
              SizedBox(
                height: Get.height * 0.02,
              ), // SizedBox
```

صورة 10 Programing code of wallet

واجهة المستخدم:



صورة 11 The wallet user interface

3- ملف المستخدم:

كود البرمجة:

```
class profileClientView extends GetView<ProfileController> {
  const profileClientView({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color.fromRGBO(247, 248, 250, 1),
      body: GetBuilder<ProfileController>(
        builder: (buildContext) => Column(
          children: [
            Container(
              padding: EdgeInsets.only(top: 50),
              decoration: BoxDecoration(
                color: PrimaryColorBlue,
                borderRadius:const BorderRadius.only(
                    bottomLeft: Radius.circular(50),
                    bottomRight: Radius.circular(50)), // BorderRadius.only
              ), // BoxDecoration
              height: Get.height * 0.27,
              width: Get.width,
              child: Center(
                child: Column(
                  children: [
                    Container(
                      decoration: BoxDecoration(
                        borderRadius: BorderRadius.circular(94),
                        color: colorscreen,
                      ), // BoxDecoration
                      height: 94,
                      width: 94,
                      child: ClipRRect(
                        borderRadius: BorderRadius.circular(94),
                        child: controller.image != ''
                            ? Image.file(
                                File(controller.image!),
                                fit: BoxFit.cover,
                              ) // Image.file
```

صورة 12 Programing code of user profile

واجهة المستخدم:

صورة 13 User profile interface

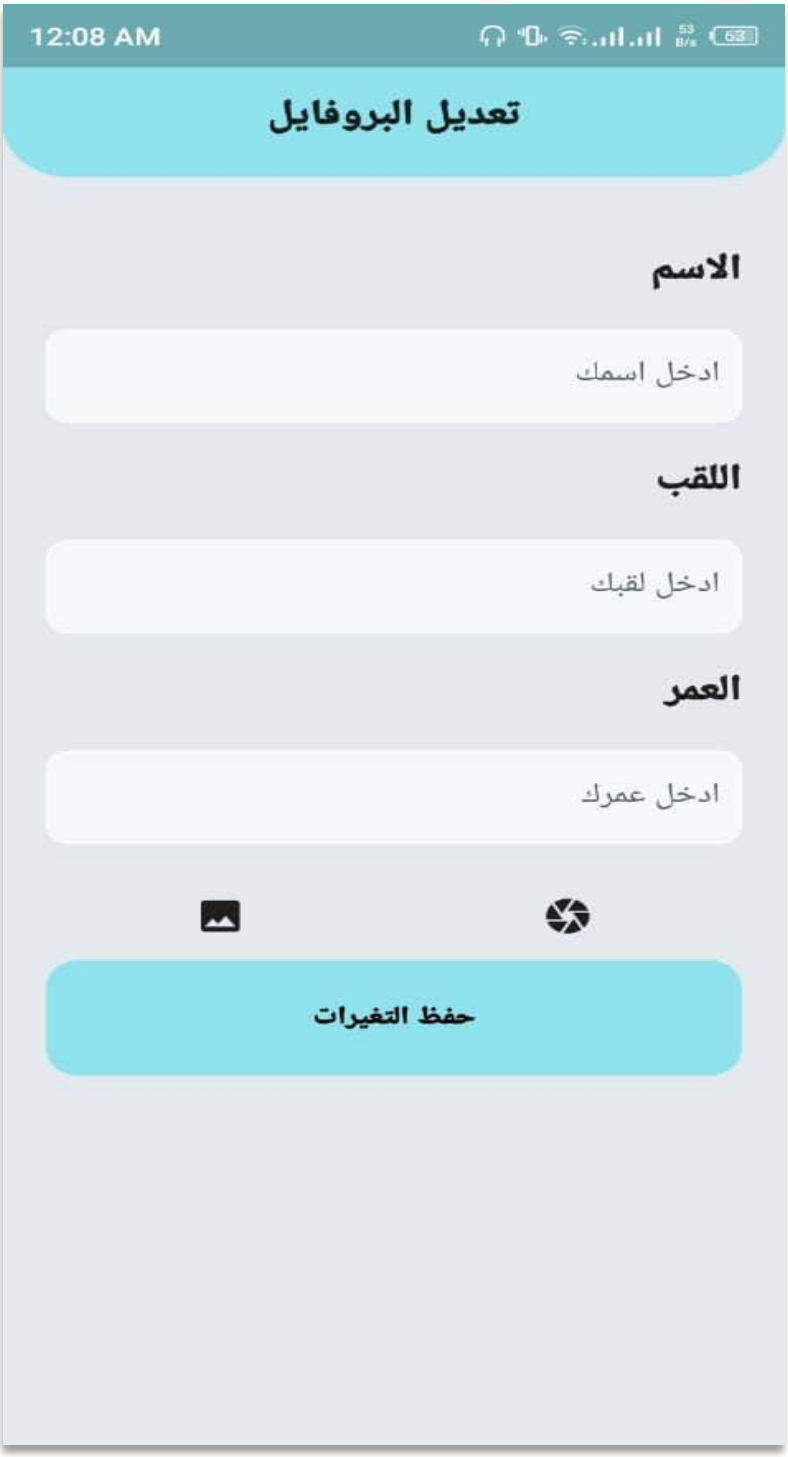
4- تعديل الملف:

كود البرمجة:

```
class EditProfileView extends GetWidget<editProfileController> {
  const EditProfileView({super.key});

  @override
  Widget build(BuildContext context) {
    TextEditingController firstNameController = TextEditingController();
    TextEditingController lastNameController = TextEditingController();
    TextEditingController ageController = TextEditingController();
    return Scaffold(
      //backgroundColor: colorfield,
      body: Container(
        color: colorfield,
        height: Get.height,
        width: Get.width,
        child: SingleChildScrollView(
          child: Column(
            children: [
              Container(
                padding: EdgeInsets.only(top: 30),
                decoration: BoxDecoration(
                  color: PrimaryColorBlue,
                  borderRadius: BorderRadius.only(
                      bottomLeft: Radius.circular(30),
                      bottomRight: Radius.circular(30)), // BorderRadius.only
                ), // BoxDecoration
                height: Get.height * 0.12,
                width: Get.width,
                child: Center(
                  child: Text(
                    'تعديل البروفايل'.tr,
                    style: Themes.Theme1,
                  ), // Text
                ), // Center
              ), // Container
              SizedBox(
                height: Get.height * 0.02,
              ), // SizedBox
```

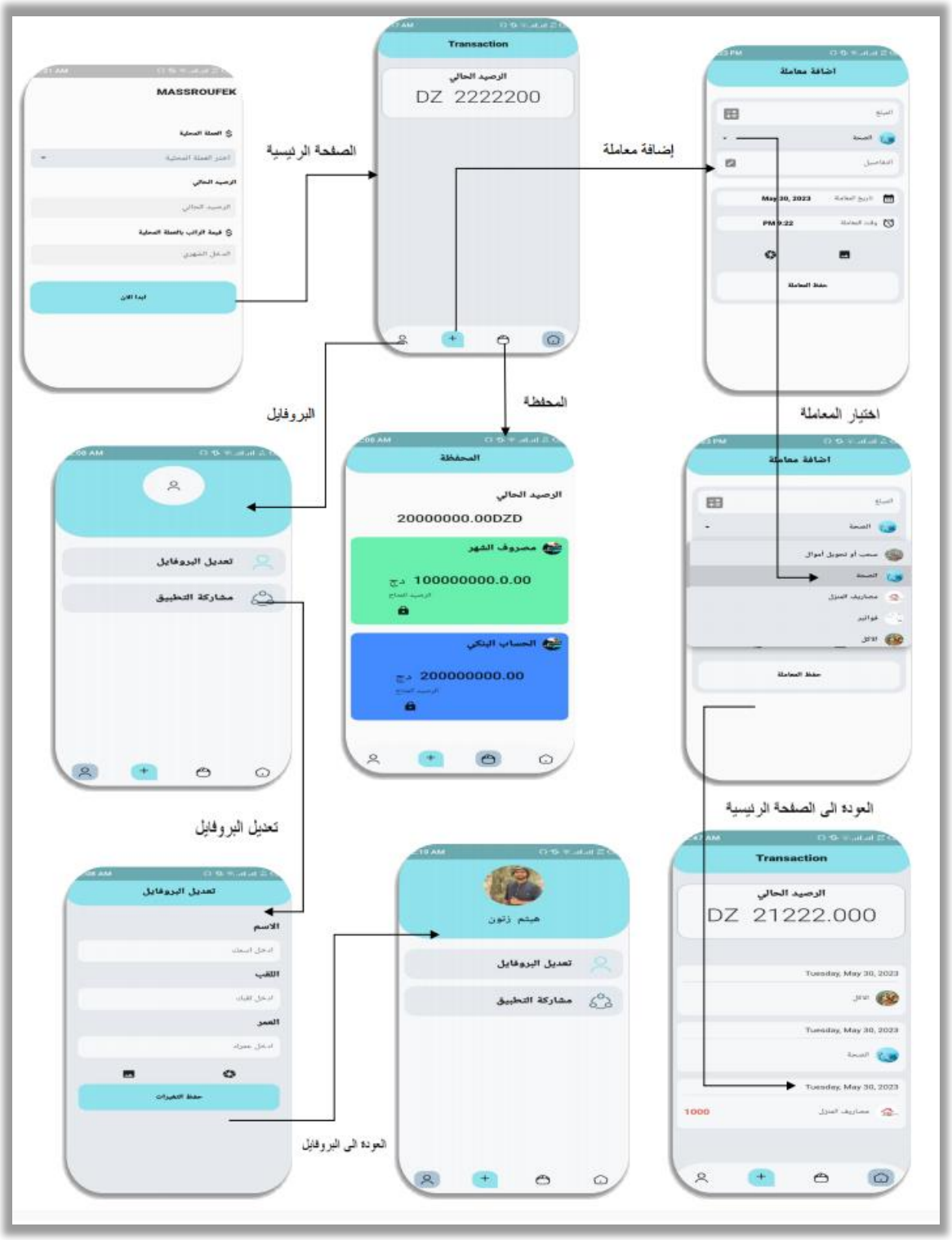
صورة 14 Programing code of edit profile

واجهة المستخدم:



صورة 15 Edit profile interface

III- تسلسل واجهات التطبيق :



3- خاتمة

هذا الفصل احتوى على أهم وأبرز التقنيات المستعملة في انجاز هذا المشروع سواءا من ناحية العتاد او من ناحية البرمجيات المستعملة، كما ضم واجهات التطبيق المنجز وأسطر برمجية وشرح لجميع وظائفه.

خاتمة عامة

خاتمة عامة

احتوى هذا المشروع مراحل تصميم وإنجاز تطبيق هاتف محمول يساعد على تسيير ومراقبة النفقات الشخصية، يوفر هذا التطبيق عمليات حفظ وعرض لهذه المعاملات وكذا تعديلها وفلترتها في قوائم حسب أي فئة يتم إدخالها بشكل منفصل، وكذا عرض شكل المشروع النهائي.

قائمة المصادر والمراجع

قائمة المصادر والمراجع

المواقع:

1. https://pub.dev/packages/hive?fbclid=IwAR3M0U0XmBzZSBpDnpKUMVWKpn1SY8YYRYhyzensfSAaS8jSvGSJfO451Z8

2. https://docs.flutter.dev/ui/widgets?fbclid=IwAR2JSnLFKQuKnPpLqdtw2zqJyMa8Zr0WLvD04NRy9mLaIquAQFJEokGE180

الروابط:

1. https://www.youtube.com/watch?v=lRercKJaAes&list=PLw6Y5u47CYq47oDw63bMqkq06fjuoK_GJ

2. https://www.youtube.com/watch?v=OaSZYA-K8HM

3. https://www.youtube.com/watch?v=z6j6yJvx3J4

الحمد لله