

Développement des applications mobiles

Module : Architecture et applications mobiles

Filière : Génie Logiciel

Application de gestion des finances personnelles



DirhamWay
Parce que les flouss,
ça file vite !

Réalisé par :

TAIK Kawtar
JAMYL Hanane
OUAZRI Khaoula
EL ALOUAN Wisal

Encadré par :

Pr. GUERMAH HATIM

Année académique : 2024/2025

Remerciements

Ce projet en développement des applications mobiles n'aurait pas pu être réalisé sans le soutien de certaines personnes à qui nous tenons à exprimer notre reconnaissance et notre gratitude.

Nous souhaitons adresser nos remerciements les plus sincères à notre encadrant, Pr. GUERMAH HATIM, pour son accompagnement précieux tout au long de ce projet. Sa disponibilité, ses conseils avisés et son expertise ont été d'une aide précieuse pour mener à bien ce projet.

Nous tenons également à remercier tous les professeurs, formateurs et encadrants de l'École Nationale Supérieure d'Informatique et d'Analyse des Systèmes pour la qualité de leur enseignement et leur disponibilité constante. Leur accompagnement a été déterminant dans la réalisation de ce projet et dans l'acquisition des compétences nécessaires à sa mise en œuvre.

Résumé

Dans un monde où la gestion des finances personnelles devient une nécessité, notre application mobile propose une solution intuitive et intelligente pour aider les utilisateurs à suivre leurs dépenses, gérer leurs budgets et obtenir des recommandations personnalisées pour économiser de l'argent. Grâce à une interface moderne et interactive, l'application permet l'enregistrement manuel des transactions, l'analyse des habitudes de consommation et la génération de conseils optimisés via un algorithme d'intelligence artificielle. L'utilisation de Jira pour la gestion Agile permet une organisation efficace des tâches entre les quatre membres de l'équipe, garantissant un développement structuré et une amélioration continue du produit.

Mots clés : Gestion financière, Budget, Recommandation IA, Dépenses, Android.

Abstract

In a world where personal finance management is essential, our mobile application offers an intuitive and smart solution to help users track their expenses, manage their budgets, and receive personalized recommendations to save money. With a modern and interactive interface, the app enables manual transaction logging, consumption habit analysis, and optimized advice generation through an artificial intelligence algorithm. The use of Jira for Agile management ensures an efficient task organization among the four team members, guaranteeing a structured development and continuous product improvement.

Keywords : Financial management, Budgeting, AI recommendation, Expenses, Android

Table des matières

Remerciements	1
Résumé	2
Abstract	3
Introduction générale	8
1 Contexte général du projet	10
1.1 Contexte général	10
1.2 Problématique	10
1.3 Étude de Marché	11
1.4 Objectifs du projet	11
1.5 Présentation du projet	12
1.6 Les enjeux du projet	12
2 Analyse fonctionnelle	13
2.1 Analyse des besoins fonctionnels	13
2.2 Analyse des besoins non fonctionnels	14
2.3 Priorisation des besoins	14
2.4 Méthodologie adoptée	15
2.4.1 Product Backlog priorisé	15
2.4.2 Organisation de l'équipe et rôles	16
2.4.3 Répartition des responsabilités techniques entre les membres	16
2.5 Diagramme des cas d'utilisation	19
3 Conception UML	20
3.1 Diagramme de classe	20
3.2 Diagrammes de séquence	21
3.2.1 Diagramme de séquence - Processus d'inscription	21

3.2.2	Diagramme de séquence – Authentification de l'utilisateur (Login)	23
3.2.3	Diagramme de séquence – Suggestion de la répartition budgétaire	24
3.2.4	Diagramme de séquence – Création d'un objectif financier	25
3.2.5	Diagramme de séquence – Ajout d'une transaction	26
3.2.6	Diagramme de séquence – Génération d'un rapport PDF	27
4	Implémentation	29
4.1	Technologies et outils utilisées	29
4.1.1	Environnement de développement intégré	29
4.1.2	Realm Database	30
4.1.3	Firebase	31
4.1.4	TensorFlow Lite	31
4.2	Architecture MVVM	31
4.2.1	Model	32
4.2.2	Repository	32
4.2.3	ViewModel	32
4.2.4	View	32
4.2.5	Utilitaires	33
4.2.6	Injection de dépendances	33
4.3	Exploration des interfaces utilisateur de DirhamWay	33
4.3.1	Interface d'authentification	33
4.3.2	Interfaces d'inscription	35
4.3.3	Interface de gestion de transactions	37
4.3.4	Interfaces de gestions des objectifs	38
4.3.5	Interface de la répartition intelligente des dépenses	41
4.3.6	Interface des rapports financiers	43
Conclusion		47

Table des figures

2.1	Diagramme des cas d'utilisation	19
3.1	Diagramme de classe	20
3.2	Diagramme de séquence - Processus d'inscription	22
3.3	Diagramme de séquence - Processus d'authentification	23
3.4	Diagramme de séquence - Suggestion de la répartition budgétaire	24
3.5	Diagramme de séquence - Création d'un objectif financier	25
3.6	Diagramme de séquence - Ajout d'une transaction	26
3.7	Diagramme de séquence - Génération d'un rapport PDF	27
4.1	Architecture MVVM de l'application	32

Liste des tableaux

2.3	Tableau des user stories priorisées	16
2.4	Répartition des tâches par membre de l'équipe	18

Introduction générale

À l'ère du smartphone et de l'instantané, la gestion de nos finances personnelles se réinvente pour devenir à la fois simple et captivante. Imaginez une application mobile qui transforme le suivi de vos dépenses en une expérience interactive : chaque transaction est une pièce du puzzle qui, une fois assemblée, révèle une vision claire de votre avenir financier.

Cette application ne se contente pas de comptabiliser votre argent, elle vous accompagne au quotidien en vous offrant des conseils personnalisés pour optimiser vos dépenses et réaliser vos projets. Grâce à une interface intuitive et élégante, vous pouvez visualiser facilement où va chaque centime, découvrir des tendances dans vos habitudes et recevoir des recommandations qui vous aident à mieux économiser.

Conçue pour être accessible à tous, même à ceux qui ne se considèrent pas comme des experts en chiffres, cette solution allie simplicité et efficacité. Elle propose une gestion ludique de votre budget, où l'analyse de vos dépenses devient une aventure motivante plutôt qu'une contrainte.

En adoptant cette approche innovante, notre application aspire à transformer votre rapport à l'argent, en faisant de la gestion financière non seulement un outil de suivi, mais également un véritable partenaire pour bâtir un futur plus serein et organisé.

Ce rapport est structuré en trois chapitres principaux :

- **Chapitre I : Contexte général du projet**

Ce chapitre présente le contexte du projet en mettant en avant la nécessité d'un outil efficace pour gérer les finances personnelles. Il souligne les difficultés rencontrées par les utilisateurs, comme le manque de visibilité sur leurs dépenses et l'absence de recommandations adaptées. L'objectif est de proposer une solution intuitive et intelligente pour optimiser la gestion budgétaire.

- **Chapitre II : Analyse fonctionnelle**

Ce chapitre présente les besoins fonctionnels et non fonctionnels de l'application DirhamWay. Ce chapitre est consacré à l'analyse fonctionnelle et non fonctionnelle du système. Il s'appuie sur la méthodologie Scrum pour organiser le développement du projet de manière itérative et collaborative. Il détaille les besoins des utilisateurs sous forme de user stories, ainsi que les cas d'utilisation principaux.

- **Chapitre III : Conception UML**

Ce chapitre se concentre sur la conception du système à l'aide du langage UML (Unified Modeling Language). Il comprend les diagrammes de cas d'utilisation, de classes, de séquence, d'activités et d'états, afin de modéliser de manière précise le comportement attendu du système. Ces diagrammes servent de base pour orienter le développement technique.

- **Chapitre IV : Implémentation**

Ce chapitre décrit en détail les technologies choisies (Java, Android Studio, Realm, Firebase,

TensorFlow Lite) et justifie leur pertinence par rapport aux contraintes du projet. Il illustre les principales interfaces développées (authentification, dashboard, ajout de transactions, définition d'objectifs, visualisation des recommandations). Les mécanismes d'interaction entre les composants MVVM sont également expliqués, ainsi que les défis rencontrés lors de l'intégration des modules d'IA pour la prédition budgétaire.

Chapitre 1

Contexte général du projet

Dans un environnement économique en évolution, maîtriser ses finances personnelles est devenu essentiel. Ce chapitre expose le contexte de notre projet, met en lumière la problématique rencontrée par les utilisateurs et définit les objectifs de l'application, afin de proposer une solution simple et personnalisée pour suivre et optimiser son budget.

1.1 Contexte général

Dans un environnement économique où les transactions se diversifient et les outils numériques se multiplient, gérer efficacement ses finances personnelles devient un enjeu majeur. Les utilisateurs recherchent des solutions simples, rapides et personnalisées pour centraliser leurs dépenses, suivre l'évolution de leurs revenus et recevoir des conseils adaptés. Cette demande croissante s'inscrit dans une tendance où la technologie mobile facilite l'accès à l'information financière, mais requiert également des fonctionnalités avancées pour véritablement aider les individus à optimiser leur budget et à améliorer leur santé financière.

1.2 Problématique

La gestion des finances personnelles reste un défi pour de nombreux utilisateurs, malgré l'existence de nombreuses applications et services bancaires numériques. La plupart de ces outils permettent d'enregistrer des transactions et de consulter ses soldes, mais ils offrent rarement une vision globale et centralisée des finances, en particulier pour ceux qui utilisent plusieurs comptes ou différents modes de paiement.

Par ailleurs, les fonctionnalités de recommandations financières restent limitées dans de nombreuses applications, ne proposant que des rapports basiques sans véritable accompagnement personnalisé. Pourtant, pour mieux gérer leur budget, les utilisateurs ont besoin d'alertes pertinentes, de conseils ciblés et d'analyses adaptées à leur comportement de consommation.

Ainsi, la problématique que nous abordons est la suivante : **comment concevoir une application mobile intuitive et accessible, capable de regrouper automatiquement toutes les transactions d'un utilisateur, de les analyser intelligemment et de lui fournir des recommandations personnalisées pour optimiser sa gestion financière ?**

1.3 Étude de Marché

Dans un marché en pleine expansion, marqué par une digitalisation croissante des services financiers, de nombreuses applications de gestion budgétaire ont émergé pour répondre aux besoins des utilisateurs. L'étude de marché permet d'identifier les forces et faiblesses des solutions existantes afin de positionner notre application de manière stratégique.

Fonctionnalité	Bankin'	Linxo	YNAB	Buddy
Authentification sécurisée	✓	✓	✓	✓
Gestion des transactions	✓	✓	✓	✓
Catégorisation automatique	✓	✓	✓	✗
Suivi budgétaire	✓	✓	✓	✓
Visualisation statistique	✓	✓	✓	✓
Notifications et alertes	✓	✓	✓	✗
Recommandations IA	✓	✗	✗	✓
Mode hors ligne	✓	✗	✗	✓
Sécurisation des données	✓	✓	✓	✓
Exportation des données	✓	✗	✗	✓

1.4 Objectifs du projet

L'objectif principal de ce projet est de développer une application mobile intuitive et efficace permettant aux utilisateurs de gérer leurs finances personnelles de manière centralisée et optimisée. Pour atteindre cet objectif global, plusieurs objectifs spécifiques sont définis :

1) Suivi et gestion des transactions :

- Permettre l'enregistrement manuel des dépenses et revenus.
- Offrir une catégorisation des transactions pour une meilleure visibilité.

2) Analyse et visualisation des finances :

- Fournir des statistiques détaillées sous forme de tableaux de bord interactifs.
- Identifier les tendances et les habitudes de consommation de l'utilisateur.

3) Aide à la prise de décision et recommandations personnalisées :

- Intégrer un moteur de recommandations basé sur un algorithme d'analyse des dépenses.
- Proposer des conseils financiers adaptés, tels que des alertes en cas de dépassement de budget ou des suggestions d'économies.

4) Gestion des budgets et des objectifs financiers :

- Permettre aux utilisateurs de définir et suivre des budgets mensuels par catégorie.
- Offrir un système d'alerte lorsque certaines limites de dépenses sont atteintes.

5) Expérience utilisateur fluide et ergonomique :

- Concevoir une interface intuitive et agréable, accessible même aux non-initiés en finance.
- Intégrer des animations et une navigation fluide pour améliorer l'engagement utilisateur.

1.5 Présentation du projet

Le projet consiste à développer une application mobile de gestion des finances personnelles permettant aux utilisateurs de suivre leurs dépenses et revenus, d'analyser leurs habitudes de consommation et de recevoir des recommandations personnalisées pour mieux gérer leur budget. L'application offrira une interface intuitive, un outil simple et efficace pour aider les utilisateurs à optimiser leur gestion financière et atteindre leurs objectifs d'épargne.

1.6 Les enjeux du projet

- 1) **Enjeux fonctionnels** : Répondre aux besoins des utilisateurs en proposant une gestion centralisée et intuitive de leurs finances, avec des outils adaptés pour suivre et optimiser leurs dépenses.
- 2) **Enjeux technologiques** : Concevoir une application performante et fluide en utilisant Android Studio (Java), Realm pour le stockage local et une API REST pour la synchronisation des données.
- 3) **Enjeux ergonomiques** : Offrir une expérience utilisateur fluide, avec une navigation intuitive et des visualisations claires des données financières.
- 4) **Enjeux économiques** : Permettre aux utilisateurs de mieux gérer leur budget, d'éviter les dépenses inutiles et de favoriser l'épargne grâce à un moteur de recommandations intelligent.

Ce chapitre a permis de définir le cadre global de notre projet en présentant le contexte, la problématique, les objectifs et les enjeux de l'application DirhamWay. En s'appuyant sur une analyse approfondie des besoins utilisateurs et du marché, nous avons identifié les fonctionnalités clés qui répondront aux attentes des utilisateurs tout en assurant la performance, l'ergonomie et la sécurité du système. Nous avons également défini les objectifs spécifiques qui orienteront le développement de l'application. Le prochain chapitre se concentrera sur l'analyse fonctionnelle détaillée de l'application, en mettant l'accent sur la méthodologie Scrum adoptée pour la gestion du projet et la priorisation des user stories afin de garantir un développement structuré et agile.

Chapitre 2

Analyse fonctionnelle

Ce chapitre est consacré à l'analyse fonctionnelle et conceptuelle du projet de gestion des finances personnelles. Il vise à identifier et structurer les différents aspects fonctionnels du système en tenant compte des besoins des utilisateurs et des contraintes techniques. Ce processus nous a permis de définir les fonctionnalités essentielles de l'application ainsi que les exigences non fonctionnelles liées à la performance, la sécurité, et l'ergonomie. À travers une analyse détaillée des besoins, nous avons mis en place une feuille de route pour garantir une conception et une implémentation optimales.

2.1 Analyse des besoins fonctionnels

Les besoins fonctionnels définissent ce que le système doit accomplir pour répondre aux attentes des utilisateurs. Dans notre cas, les principaux objectifs sont :

- Offrir une interface simple et intuitive pour la gestion des finances.
- Permettre l'ajout des transactions financières.
- Offrir un module complet de gestion des objectifs d'épargne, incluant la définition d'un montant cible, d'une date limite et d'une épargne actuelle.
- Générer des statistiques et rapports illustrant les tendances financières.
- Fournir des recommandations personnalisées pour optimiser les dépenses.

Les besoins fonctionnels sont détaillés sous forme de tableaux décrivant chaque fonctionnalité attendue.

ID	Fonctionnalité	Description
BF-01	Authentification	L'utilisateur doit pouvoir créer un compte et se connecter via un email et un mot de passe.
BF-02	Gestion des transactions	L'utilisateur peut ajouter une transaction (dépense ou revenu).
BF-03	Gestion des objectifs	L'utilisateur peut créer un objectif d'épargne en précisant un montant cible, une date limite et une somme déjà épargnée. Il peut également modifier ou supprimer ses objectifs en fonction de l'évolution de ses finances.

ID	Fonctionnalité	Description
BF-04	Suivi des budgets	L'utilisateur doit pouvoir définir un budget mensuel global et par catégorie, avec des indicateurs de progression.
BF-05	Visualisation des statistiques	Affichage des dépenses sous forme de graphiques pour analyser l'évolution des finances.
BF-06	Exportation des données	L'utilisateur doit pouvoir exporter son historique des transactions sous format PDF.
BF-07	Recommandations intelligentes	Un module d'IA doit proposer des conseils pour réduire les dépenses et optimiser le budget.
BF-08	Mode hors ligne	L'application doit permettre la consultation et l'ajout de transactions même sans connexion internet.

2.2 Analyse des besoins non fonctionnels

Les besoins non fonctionnels assurent que l'application respecte les contraintes de performance, de sécurité et d'ergonomie.

ID	Besoin	Description
BNF-01	Performance	L'application doit charger le tableau de bord et les transactions en moins de 3 secondes.
BNF-02	Accessibilité	L'interface doit être ergonomique.
BNF-03	Compatibilité	L'application doit être fonctionnelle sur les versions Android 14 et 15.
BNF-04	Évolutivité	L'architecture logicielle doit permettre d'ajouter de nouvelles fonctionnalités sans refonte majeure.
BNF-05	Interface utilisateur	L'application doit adopter un design moderne basé sur Material Design.
BNF-06	Maintenabilité	Le code source doit être structuré selon le modèle MVVM pour faciliter la maintenance.

2.3 Priorisation des besoins

Les besoins fonctionnels et non fonctionnels sont classés par priorité :

- **Priorité 1** : Fonctionnalités essentielles (ex. gestion des transactions, authentification).

- **Priorité 2** : Fonctionnalités améliorant l'expérience utilisateur (ex. notifications, IA, mode hors ligne).
- **Priorité 3** : Fonctionnalités avancées ou optionnelles (ex. exportation PDF).

2.4 Méthodologie adoptée

Nous avons choisi la méthodologie Agile, et plus spécifiquement Scrum, pour notre projet de développement mobile. Cette approche permet de gérer le projet de manière itérative et flexible, en assurant une adaptation continue aux besoins des utilisateurs et aux ajustements techniques. Chaque sprint d'une semaine garantit des retours fréquents, une validation rapide des livrables, et permet de mesurer régulièrement la vélocité de l'équipe. Ce rythme soutenu assure une concentration sur un périmètre précis, tout en favorisant l'amélioration continue grâce à des revues et des rétrospectives hebdomadaires.

2.4.1 Product Backlog priorisé

Dans le cadre de notre approche Scrum, nous avons collectivement priorisé les user stories en tenant compte de leur valeur métier, de leur complexité technique et de leur urgence. Le tableau ci-dessous présente l'ensemble des user stories ainsi priorisées.

ID	Titre
US01	En tant qu'utilisateur, je veux enregistrer mes transactions (dépenses et revenus) via un formulaire afin de suivre mes finances.
US02	En tant qu'utilisateur, je veux consulter la liste de mes transactions afin d'avoir une vue d'ensemble de mes données financières.
US03	En tant qu'utilisateur, je veux pouvoir définir un objectif financier (titre, montant cible et date d'échéance), afin de suivre et planifier mon épargne de manière structurée.
US04	En tant qu'utilisateur, je veux consulter la liste complète de mes objectifs financiers avec leur état d'avancement, afin d'avoir une vision centralisée de mes objectifs d'épargne.
US05	En tant qu'utilisateur, je veux pouvoir modifier un objectif (montant, date d'échéance, etc.), afin de garder mon plan d'épargne toujours aligné avec mes priorités financières.
US06	En tant qu'utilisateur, je veux accéder à mon compte via un système sécurisé, afin de protéger la confidentialité de mes données financières.
US07	En tant qu'utilisateur, je veux visualiser l'état de mes finances via des indicateurs graphiques afin de comprendre rapidement et d'analyser ma situation financière.

ID	Titre
US08	En tant qu'utilisateur, je veux recevoir des suggestions personnalisées d'économies, afin d'optimiser la gestion de mon budget plus efficacement.
US09	En tant qu'utilisateur, je veux recevoir des notifications lorsque j'approche de mes limites budgétaires, afin de pouvoir ajuster mes dépenses.
US10	En tant qu'utilisateur, je veux pouvoir supprimer un objectif, afin de ne conserver que mes objectifs en cours et garantir l'exactitude de mon plan d'épargne.
US11	En tant qu'utilisateur, je veux pouvoir rafraîchir toutes les pages de manière dynamique, afin que les formulaires se réinitialisent et que les champs remplis soient vidés.
US12	En tant qu'utilisateur, je veux exporter mes données financières sous forme de rapports structurés, afin de les conserver et de pouvoir les consulter ou les partager à tout moment.
US13	En tant qu'utilisateur, je veux que toutes mes données soient chiffrées, afin de prévenir tout accès non autorisé à mes informations sensibles.
US14	En tant qu'utilisateur, je veux pouvoir enregistrer et consulter mes transactions sans connexion internet, afin d'assurer la continuité de mon suivi financier.

TABLE 2.3: Tableau des user stories priorisées

2.4.2 Organisation de l'équipe et rôles

L'équipe est composée de quatre membres, chacun ayant un rôle défini pour optimiser le développement :

- Product Owner : Définit les besoins, établit le backlog produit et valide les développements selon les objectifs fixés. Cette tâche est partagée de manière commune par tous les membres de l'équipe.
- Scrum Master : Facilite le suivi du projet, organise les réunions et s'assure du bon respect de la méthodologie. Chaque sprint voit un membre différent de l'équipe endosser ce rôle.
- Développeurs : Réalisent les différentes fonctionnalités (Front-end, Back-end, Intelligence Artificielle pour les recommandations).

2.4.3 Répartition des responsabilités techniques entre les membres

Afin d'assurer une implication équilibrée de chaque membre dans les différentes parties du projet (Front-end, Back-end, IA), nous répartissons le développement de l'application sur les quatre

membres. Les tâches qui suivent concernent la contribution de chaque membre à travers l'ensemble des sprints.

Membre	Front-end Interface utilisateur	Back-end Logique et données	IA Recommandations
Membre 1	<ul style="list-style-type: none"> Conception et développement de l'interface de gestion des transactions (US01) Conception de l'interface de consultation des transactions (US02) Interface de visualisation des objectifs financiers (US03, US06) 	<ul style="list-style-type: none"> Gestion des transactions (US01) Enregistrement des objectifs financiers (US05) Implémentation de la logique de mise à jour et de suppression des objectifs (US05, US07) 	<ul style="list-style-type: none"> Définition des priorités de dépense (US09) Mise en place des recommandations intelligentes (IA) pour optimiser les dépenses (US08, US10)
Membre 2	<ul style="list-style-type: none"> Interface d'ajout de transaction (US01) Interface de connexion et inscription (US06) 	<ul style="list-style-type: none"> Authentification avec Firebase (US06) Stockage des transactions et des utilisateurs (Firebase/Realm) (US01) 	<ul style="list-style-type: none"> Intégration du modèle d'IA pour générer des recommandations d'économies (US08) Traitement des prévisions budgétaires en fonction des suggestions d'épargne (US08, US10)
Membre 3	<ul style="list-style-type: none"> Interface de consultation des rapports et synthèses (US12) Rappels concernant les limites budgétaires (US09) 	<ul style="list-style-type: none"> Traitement des transactions mensuelles (calcul des soldes) (US01) Gestion de l'historique des transactions (US01) 	<ul style="list-style-type: none"> Ajustement des recommandations intelligentes selon les dépenses réelles (US08) Prise en compte des charges fixes dans les calculs d'épargne (US10)
Membre 4	<ul style="list-style-type: none"> Personnalisation graphique des rapports financiers (US12) Écran de rapports financiers détaillés (US12) Interface de génération PDF pour l'exportation des rapports (US12) 	<ul style="list-style-type: none"> Sauvegarde des rapports en PDF (US12) Implémentation des filtres pour générer des rapports sur les transactions (US12) 	<ul style="list-style-type: none"> Statistiques et bilans des objectifs d'épargne (US03) Visualisation des performances d'épargne via des graphiques (US03)

TABLE 2.4: Répartition des tâches par membre de l'équipe

2.5 Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation, illustré ci-dessous, représente les interactions entre les utilisateurs et les fonctionnalités clés du système. Ce diagramme joue un rôle essentiel dans la compréhension des besoins des utilisateurs, en identifiant les actions principales qu'ils peuvent réaliser à travers l'application. Il permet ainsi de visualiser les scénarios d'utilisation, facilitant la définition des exigences fonctionnelles et contribuant à la conception de l'architecture du système en s'assurant que toutes les fonctionnalités essentielles sont bien prises en compte.

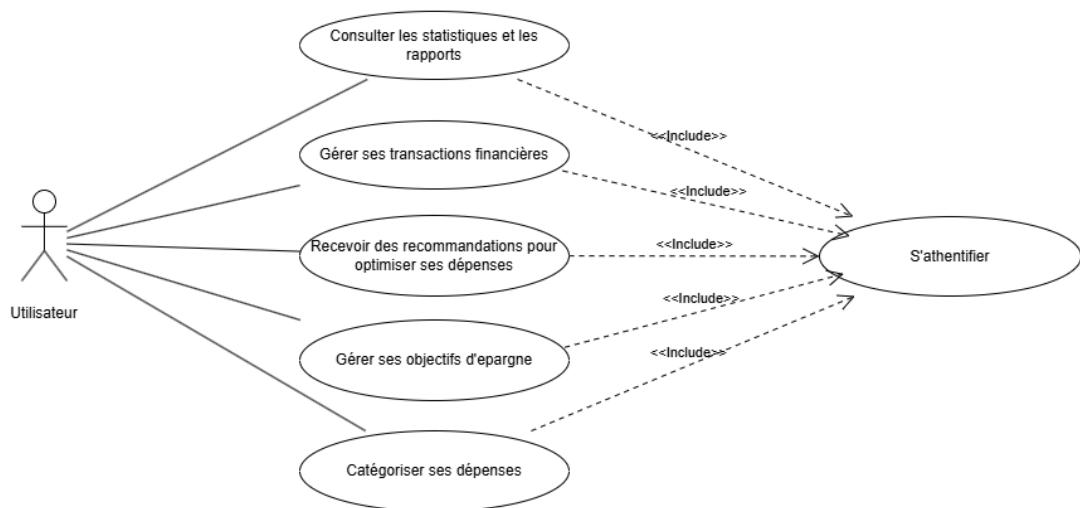


FIGURE 2.1: Diagramme des cas d'utilisation

Ce chapitre a permis d'identifier et de formaliser les besoins fonctionnels et non fonctionnels de l'application de gestion des finances personnelles. Nous avons détaillé les principales fonctionnalités nécessaires pour répondre aux attentes des utilisateurs, ainsi que les critères de performance et de sécurité afin de garantir une expérience utilisateur optimale. Cette analyse approfondie a conduit à la définition d'un Product Backlog priorisé, des tâches techniques et des responsabilités de chaque membre de l'équipe, assurant ainsi une gestion structurée du projet. La prochaine phase consistera à traduire ces exigences en une architecture logicielle robuste et évolutive, garantissant une mise en œuvre efficace et une utilisation fluide par les futurs utilisateurs.

Chapitre 3

Conception UML

Ce chapitre est dédié à la conception UML du projet DirhamWay, qui offre une vision structurée du système à travers des diagrammes statiques et dynamiques. L'objectif est de formaliser les interactions entre les différentes entités du système et de décrire les processus métiers essentiels. Le diagramme de classe présente la structure des entités principales et leurs relations, tandis que les diagrammes de séquence détaillent les flux d'interaction entre les utilisateurs et l'application pour des fonctionnalités clés telles que l'inscription, l'authentification, la gestion des transactions et la suggestion budgétaire. Ces diagrammes servent de base pour comprendre le fonctionnement interne du système et guider la phase de développement.

3.1 Diagramme de classe

Le diagramme ci-dessus représente la structure des principales entités de l'application de gestion budgétaire. Il met en évidence les relations entre les utilisateurs **User**, leurs catégories budgétaires **UserCategory**, les catégories globales **Category**, les options associées **Option** et les objectifs financiers **Goal**.

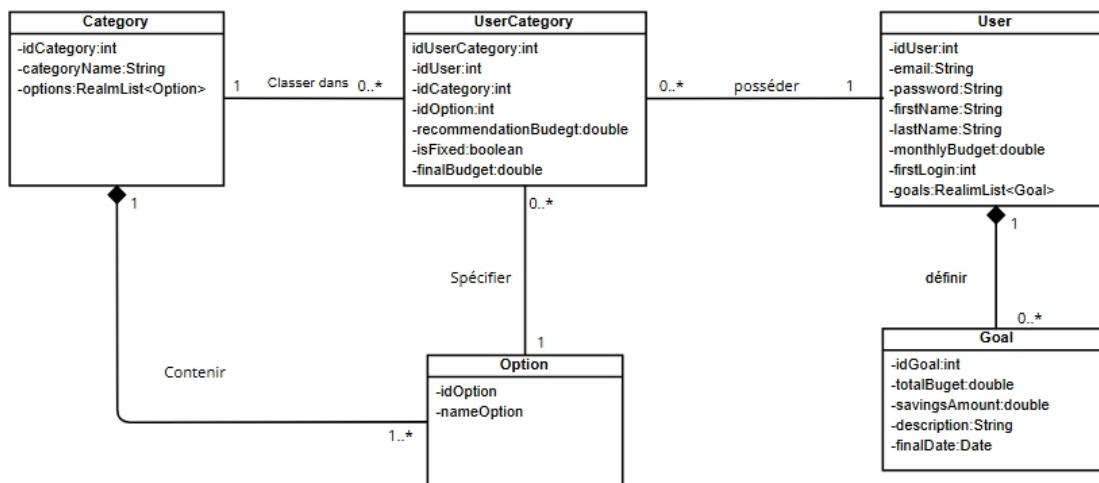


FIGURE 3.1: Diagramme de classe

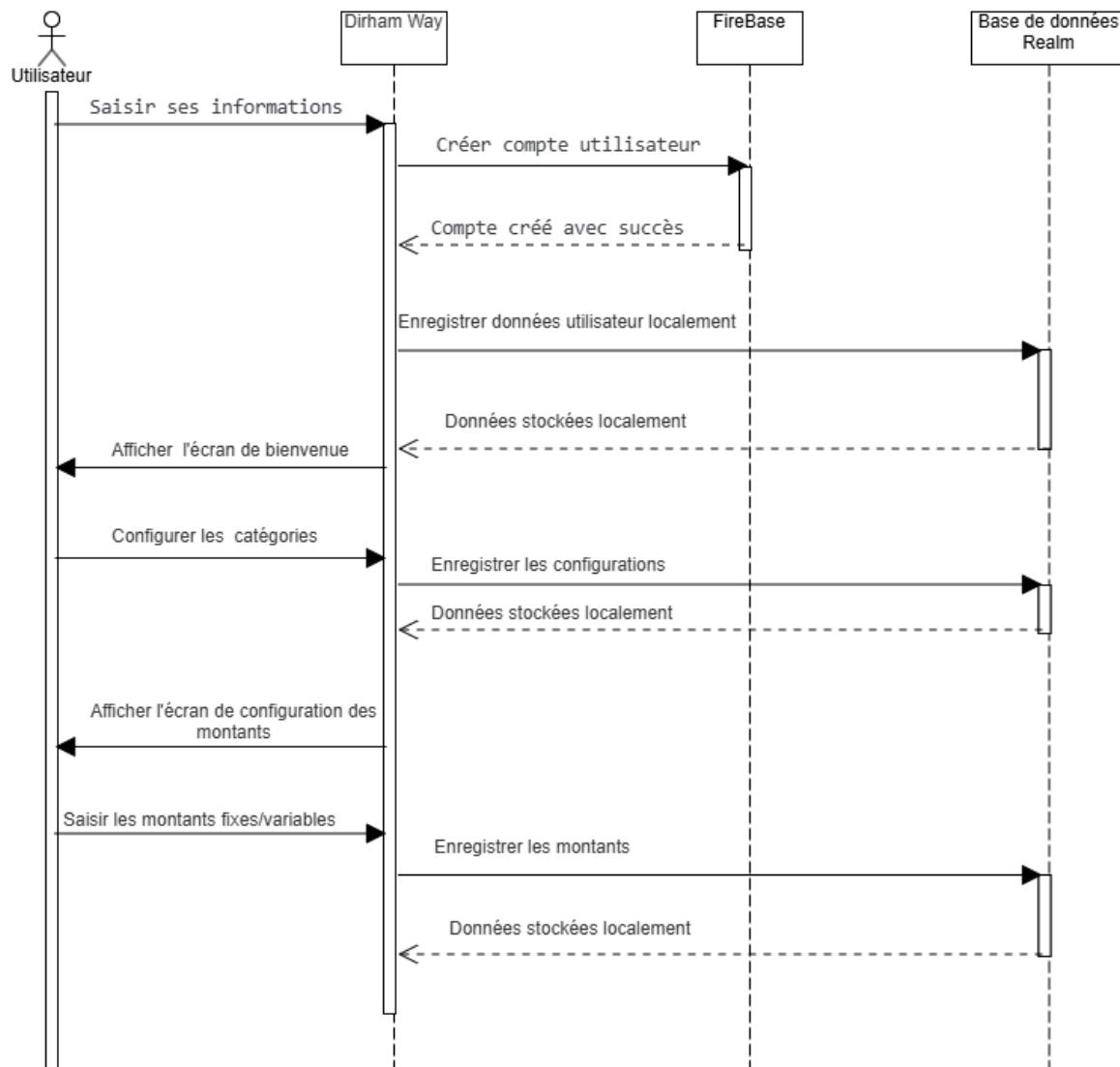
- Chaque utilisateur peut posséder plusieurs catégories personnalisées **UserCategory**, chacune étant classée dans une catégorie générale **Category** et spécifiée par une option (Option).

- Une catégorie peut contenir plusieurs options afin de proposer différents types de dépenses ou d'objectifs budgétaires.
- L'utilisateur peut également définir plusieurs objectifs financiers (Goal) en lien avec son budget mensuel.

3.2 Diagrammes de séquence

3.2.1 Diagramme de séquence - Processus d'inscription

Le diagramme de séquence ci-dessous illustre les différentes étapes fonctionnelles du processus d'inscription d'un utilisateur dans l'application **DirhamWay**. Il met en évidence les interactions entre l'utilisateur, l'application mobile, la base d'authentification Firebase et la base de données locale Realm.

**FIGURE 3.2:** Diagramme de séquence - Processus d'inscription

- **Saisie des informations** : l'utilisateur remplit les champs requis (nom, prénom, email, mot de passe) sur l'écran d'inscription.
- **Création du compte** : l'application contacte Firebase pour créer le compte utilisateur.
- **Enregistrement local** : une fois le compte créé avec succès, les données de l'utilisateur sont enregistrées localement dans la base Realm.
- **Affichage de l'écran de bienvenue** : l'utilisateur est redirigé vers l'écran d'accueil personnalisé.
- **Configuration des catégories** : l'utilisateur sélectionne les catégories et les options qui correspondent à ses habitudes de dépenses.
- **Enregistrement des préférences** : les choix effectués sont enregistrés localement.
- **Saisie des montants** : l'utilisateur renseigne les montants fixes et variables associés aux options sélectionnées.

- **Sauvegarde finale** : les montants sont également stockés dans la base locale.

3.2.2 Diagramme de séquence – Authentification de l'utilisateur (Login)

Le diagramme de séquence ci-dessous illustre le processus d'authentification d'un utilisateur dans l'application **DirhamWay**, en distinguant les cas de succès et d'échec. Il met en évidence les interactions entre l'utilisateur, l'application mobile, le service Firebase (authentification), et la base de données locale Realm.

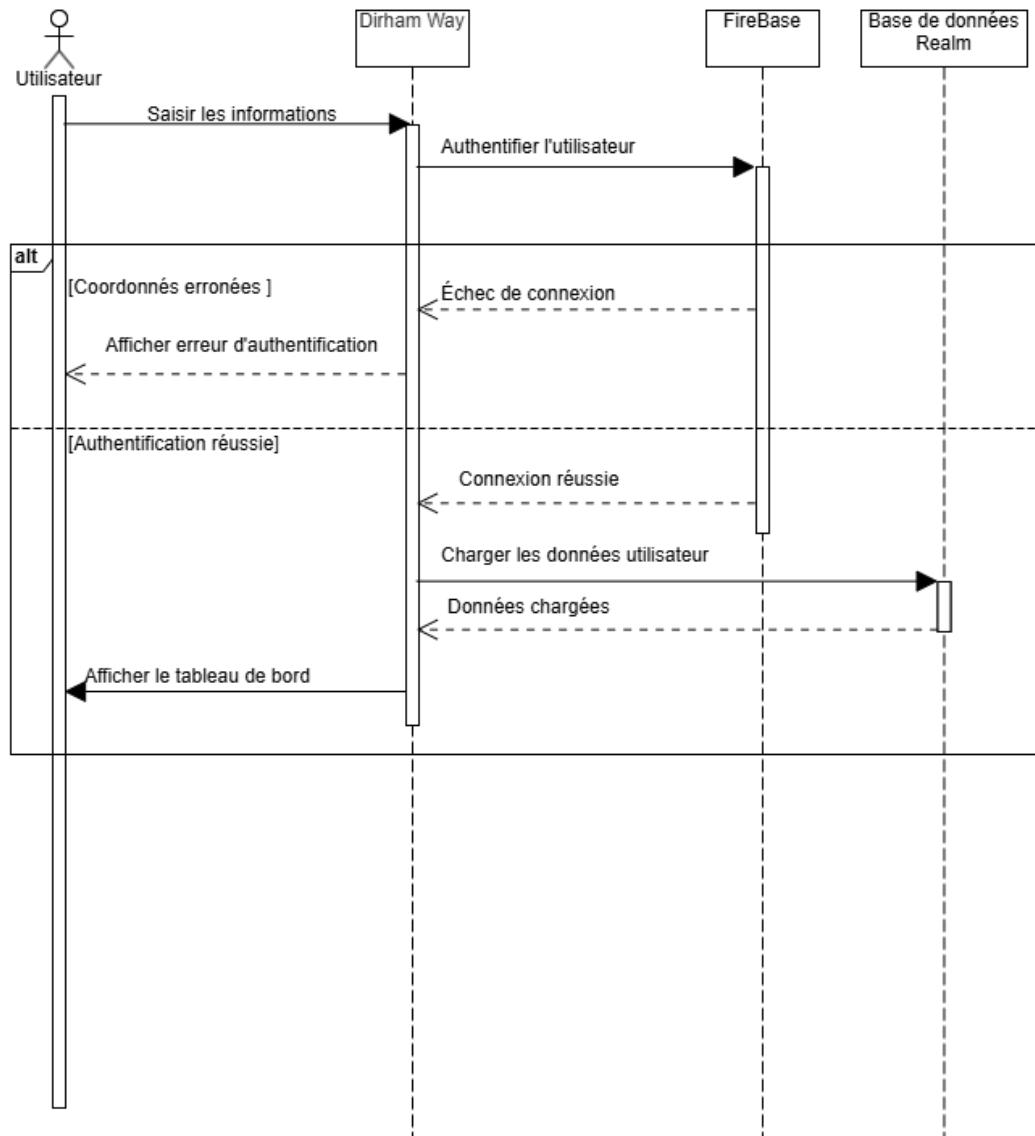


FIGURE 3.3: Diagramme de séquence - Processus d'authentification

- **Étape 1 : Saisie des identifiants** L'utilisateur saisit son adresse e-mail et son mot de passe via l'interface de l'application.

- **Étape 2 : Demande d'authentification** L'application transmet les identifiants à Firebase pour vérification.
- **Étape 3 : Deux scénarios possibles :**
 - **Succès** : Firebase valide l'identité de l'utilisateur. L'application interroge alors Realm pour retrouver ou créer le profil utilisateur. Les données sont chargées, puis le tableau de bord ou l'écran de bienvenue est affiché selon le contexte (première connexion ou non).
 - **Échec** : Firebase retourne une erreur. L'application affiche un message d'échec à l'utilisateur sans accéder à Realm.
- **Étape 4 : Accès au contenu** Si la connexion est réussie, l'utilisateur est redirigé vers l'écran principal de l'application (dashboard ou page de bienvenue).

3.2.3 Diagramme de séquence – Suggestion de la répartition budgétaire

Ce diagramme de séquence représente les étapes du processus de prédition budgétaire personnalisé dans l'application **DirhamWay**, basé sur un modèle d'intelligence artificielle embarqué avec TensorFlow Lite.

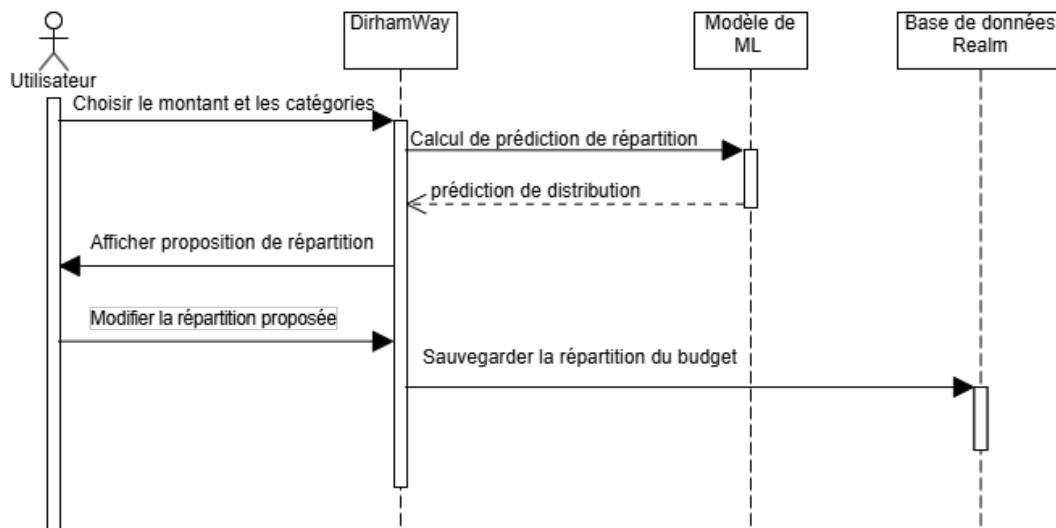


FIGURE 3.4: Diagramme de séquence - Suggestion de la répartition budgétaire

L'utilisateur saisit un montant global de budget ainsi qu'un ensemble de catégories sélectionnées. Le système va alors effectuer les actions suivantes :

- **1. Récupération des paiements fixes** L'application interroge la base de données locale (Realm) pour identifier les catégories marquées comme « fixes » (ex : Internet, Téléphone, Crédit) ainsi que les montants associés à ces catégories.
- **2. Calcul du budget variable** Le budget global saisi est réduit des paiements fixes pour obtenir le montant à allouer aux autres catégories variables.

- **3. Préparation des données d'entrée** Les catégories restantes sont encodées (par ID + date du jour, mois, etc.) et préparées pour être injectées dans le modèle TensorFlow Lite.
- **4. Prédiction des montants par catégorie variable** Le modèle embarqué (TFLitePredictor) prédit un poids relatif pour chaque catégorie variable. Ces prédictions sont utilisées pour effectuer une répartition proportionnelle du budget variable.
- **5. Fusion des montants fixes et prédictifs** Les allocations fixes et variables sont fusionnées dans un unique dictionnaire contenant les montants finaux à allouer à chaque catégorie.
- **6. Mise à jour de l'interface utilisateur** Les résultats sont affichés de manière visuelle (barres de couleurs, pourcentages, recommandations) avec possibilité de les modifier manuellement via des SeekBars pour chaque catégorie non fixe.
- **7. Option de sauvegarde** Une fois l'ajustement terminé, l'utilisateur peut sauvegarder les allocations dans Realm.

3.2.4 Diagramme de séquence – Crédation d'un objectif financier

Ce diagramme de séquence illustre les interactions entre les trois acteurs principaux du système : l'**Utilisateur**, l'**Application DirhamWay**, et la **base de données Realm**, dans le cadre de la création d'un nouvel objectif financier.

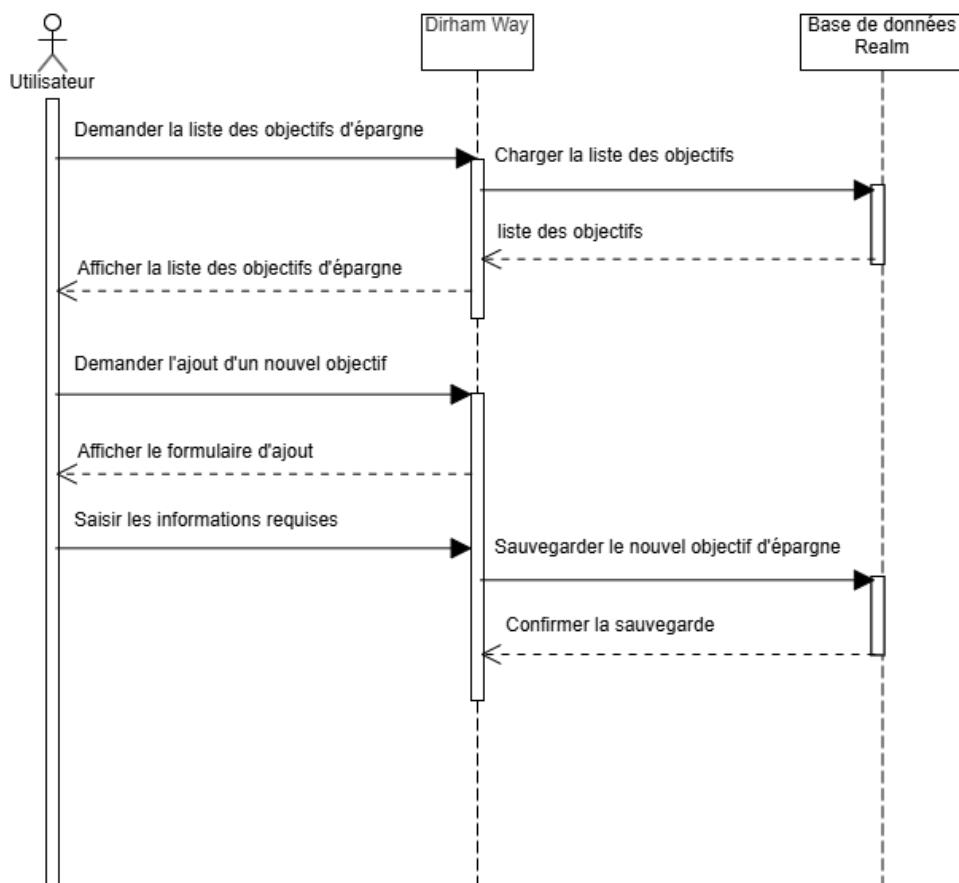


FIGURE 3.5: Diagramme de séquence - Crédation d'un objectif financier

- **1. Consultation des objectifs** : L'utilisateur commence par demander la liste de ses objectifs. L'application interroge Realm pour charger et afficher ces données à l'utilisateur.

- 2. Ajout d'un objectif** : L'utilisateur déclenche l'ouverture de l'interface de création en cliquant sur le bouton "Ajouter".
- 3. Saisie des données de l'objectif** : L'utilisateur remplit les champs du formulaire : description de l'objectif, montant total visé, montant déjà épargné, et la date limite.
- 4. Validation et sauvegarde** : Une fois les informations saisies, l'utilisateur clique sur "Sauvegarder". L'application crée alors une instance de la classe Goal, calcule l'ID approprié, puis envoie l'objectif à la base de données Realm pour persistance.
- 5. Confirmation** : Une fois l'objectif stocké, l'application informe l'utilisateur que l'objectif a été ajouté avec succès et ferme l'écran de création.

3.2.5 Diagramme de séquence – Ajout d'une transaction

Ce diagramme illustre les interactions entre l'utilisateur, l'application DirhamWay et la base de données Realm lors de la consultation et l'ajout d'une transaction financière, qu'elle soit un crédit (entrée d'argent) ou un débit (dépense).

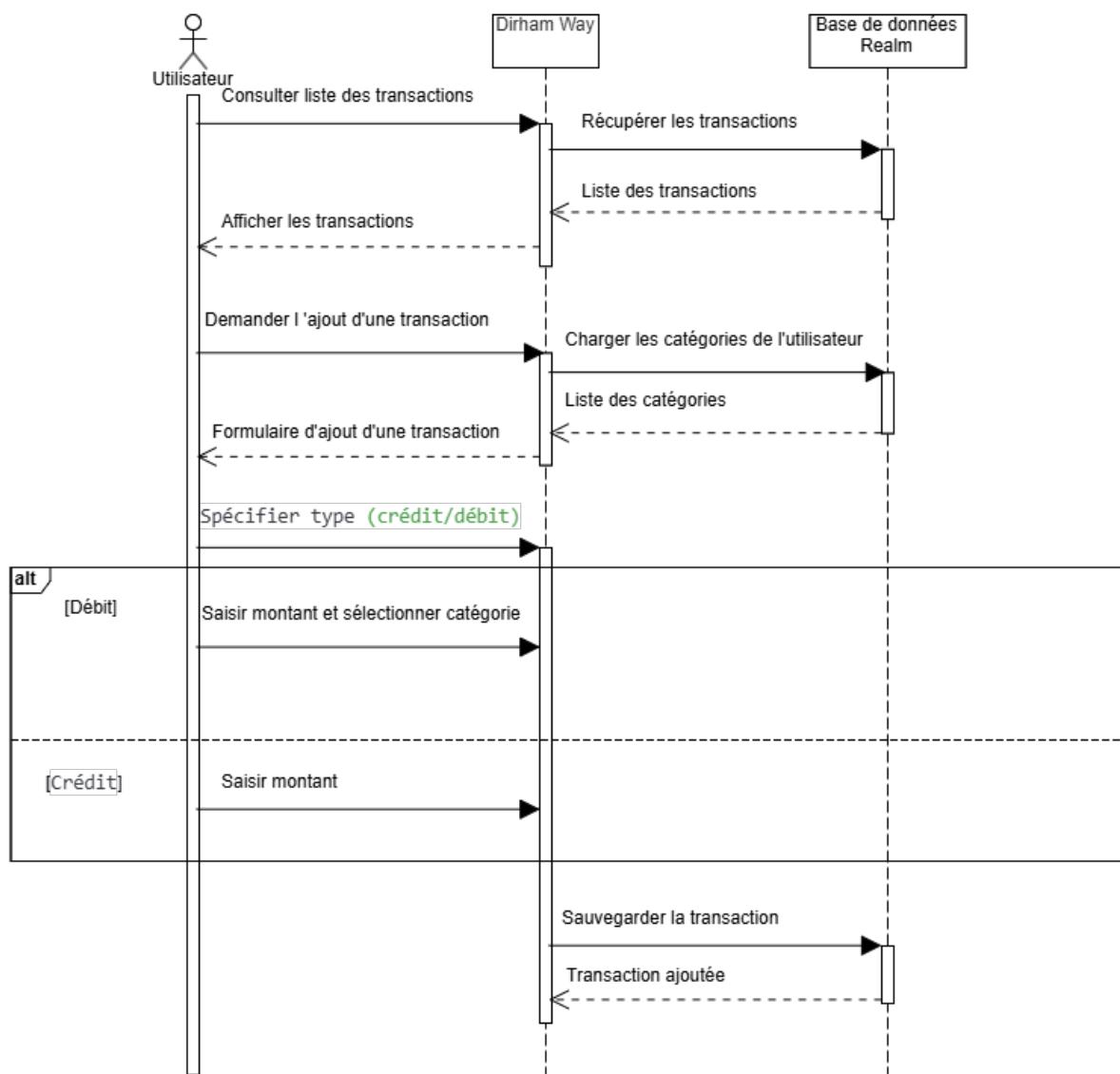


FIGURE 3.6: Diagramme de séquence - Ajout d'une transaction

- **1. Consultation initiale** : L'utilisateur demande à voir la liste de ses transactions existantes. L'application interroge la base de données Realm, qui retourne les données à afficher.
- **2. Ajout d'une transaction** : L'utilisateur déclenche le processus d'ajout en cliquant sur un bouton "Ajouter une transaction".
- **3. Chargement des catégories disponibles** : Pour associer la transaction à une catégorie, l'application demande à Realm la liste des catégories déjà sélectionnées par l'utilisateur.
- **4. Spécification du type** : L'utilisateur choisit si la transaction est un **crédit** (entrée) ou un **débit** (sortie).
- **5. Complétion du formulaire** :
 - Si **crédit**, l'utilisateur saisit uniquement le montant et la description.
 - Si **débit**, l'utilisateur doit en plus choisir une *catégorie* à associer (ex : Alimentation, Transport).
- **6. Enregistrement** : L'application enregistre la transaction dans Realm et confirme le succès à l'utilisateur.

3.2.6 Diagramme de séquence – Génération d'un rapport PDF

Ce diagramme de séquence illustre le processus de génération d'un rapport financier personnalisé à la demande de l'utilisateur dans l'application **DirhamWay**.

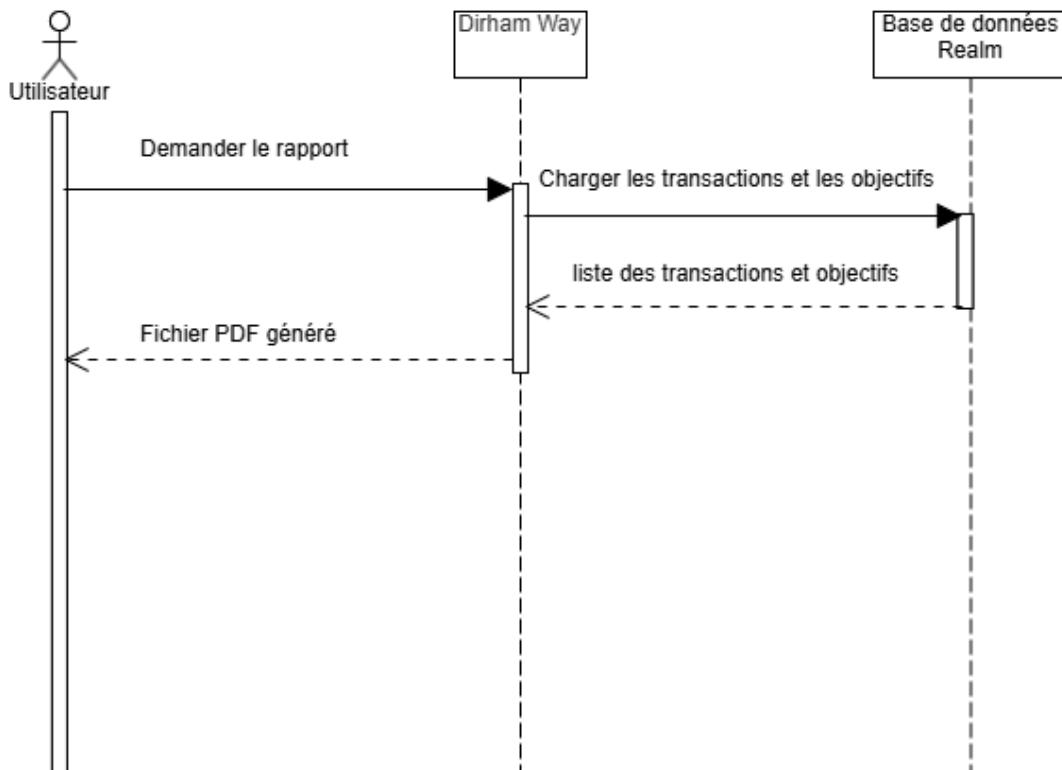


FIGURE 3.7: Diagramme de séquence - Génération d'un rapport PDF

- **Demande de l'utilisateur** : L'utilisateur initie l'action en appuyant sur le bouton d'exportation de rapport dans l'interface de l'application.
- **Chargement des données** : L'application envoie une requête à la base de données Realm pour récupérer toutes les transactions financières et les objectifs associés à l'utilisateur courant.

- **Récupération des informations** : Realm retourne la liste des transactions et des objectifs en cours ou passés.
- **Génération du rapport** : Une fois les données reçues, l'application assemble les informations dans un rapport formaté (PDF), incluant un résumé, des tableaux et des visualisations personnalisées.
- **Retour à l'utilisateur** : L'utilisateur est informé que le fichier PDF a été généré avec succès et sauvegardé localement.

À travers les diagrammes UML présentés dans ce chapitre, nous avons formalisé les aspects statiques et dynamiques du projet DirhamWay. Le diagramme de classe a permis de clarifier les relations entre les entités principales du système, tandis que les diagrammes de séquence ont illustré les différents processus fonctionnels du système. Ces éléments de conception offrent une vision précise des interactions et des flux de données nécessaires pour une implémentation correcte du système. La prochaine étape consistera à traduire ces modèles en une architecture logicielle concrète, prête à être mise en œuvre dans l'application.

Chapitre 4

Implémentation

Ce chapitre présente l'implémentation technique de l'application DirhamWay, en détaillant les technologies utilisées pour la conception, le développement et l'optimisation de l'application. Nous y décrirons les principaux outils et frameworks intégrés, et l'architecture MVVM. Cette phase vise à traduire les concepts fonctionnels en une solution technique robuste, évolutive et performante. L'implémentation repose sur des choix technologiques judicieusement sélectionnés pour répondre aux besoins de gestion des finances personnelles, en garantissant une expérience utilisateur fluide et des recommandations intelligentes adaptées. Nous allons également présenter les captures d'écran de l'application, illustrant les interfaces et les fonctionnalités mises en œuvre.

4.1 Technologies et outils utilisées

Dans le cadre du développement de notre application, nous avons intégré un ensemble cohérent de technologies, choisies avec soin afin de répondre aux exigences fonctionnelles des utilisateurs tout en assurant la fiabilité, la performance et la scalabilité du système. Chaque technologie a été sélectionnée en fonction de sa pertinence pour un besoin précis, contribuant ainsi à construire une solution robuste, moderne et évolutive. Ce qui suit présente un aperçu détaillé des outils utilisés et de leur contribution dans la mise en œuvre de notre application.

4.1.1 Environnement de développement intégré

L'ensemble des fonctionnalités de l'application a été développé en langage **Java**, qui reste l'un des langages de programmation les plus stables et largement adoptés pour le développement Android. Grâce à sa compatibilité native avec Android Studio, Java a permis une intégration fluide avec les bibliothèques Android, les services Firebase, et la base de données Realm utilisée dans ce projet. Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android. Il est basé sur IntelliJ IDEA et fourni par Google, offrant une plateforme puissante et spécialisée pour la création, le test et le déploiement d'applications mobiles sur les appareils Android.



Dans le cadre de ce projet, Android Studio a été choisi pour plusieurs raisons essentielles :

- **Support natif d'Android SDK et des bibliothèques Jetpack** : Android Studio intègre l'ensemble des outils nécessaires pour concevoir des interfaces modernes, gérer les permissions, la navigation, les bases de données, etc.
- **Interface graphique intuitive** : L'éditeur de layouts permet de visualiser et concevoir facilement les interfaces utilisateurs à l'aide du système de contraintes (ConstraintLayout).
- **Emulateur intégré** : Il permet de tester l'application sur différentes tailles et versions de smartphones sans avoir besoin de matériel réel.
- **Support complet de Gradle** : Gradle permet la gestion des dépendances, la compilation, l'optimisation et le packaging de l'application.
- **Outils de débogage et de profilage** : Android Studio offre une visualisation en temps réel de la mémoire, du CPU, des threads et de l'utilisation réseau.

4.1.2 Realm Database

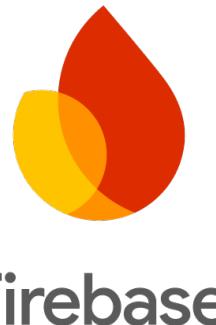
Realm est une base de données conçue spécifiquement pour les plateformes mobiles, à la différence de nombreuses bibliothèques qui reposent sur SQLite. Elle se distingue par le fait qu'elle ne dépend d'aucune base de données sous-jacente, offrant ainsi une solution native, légère et optimisée pour les environnements mobiles. Un des grands avantages de Realm réside dans sa fonctionnalité Realm Sync, qui permet de synchroniser automatiquement les données entre plusieurs appareils via le cloud. Cela garantit que les données restent disponibles même si l'utilisateur change de téléphone, assurant ainsi une continuité et une accessibilité des informations à travers différents appareils, sans nécessiter de gestion manuelle des sauvegardes ou de la synchronisation.

Realm utilise son propre moteur de stockage optimisé, permettant une manipulation fluide d'objets complexes, des requêtes avancées, ainsi qu'un chargement paresseux des données pour des performances élevées. Grâce à son API intuitive, sa compatibilité multi-langages (Java, Swift, Objective-C, React Native, Xamarin) et son intégration facile, Realm est aujourd'hui utilisé par de grandes entreprises comme Twitter, Starbucks ou Alibaba pour gérer efficacement des volumes importants de données sur mobile.



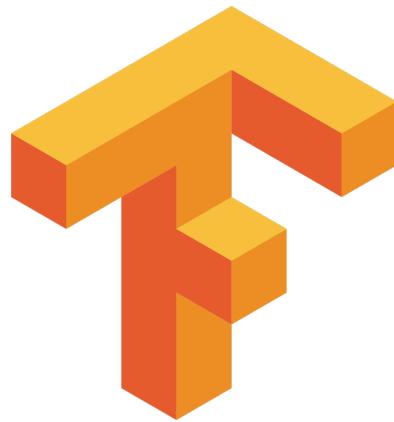
4.1.3 Firebase

Firebase Authentication est un service de Google qui permet d'authentifier les utilisateurs d'une application mobile ou web de manière sécurisée. Il fournit une solution clé en main pour gérer la connexion via email/mot de passe, Google, Facebook, ou numéro de téléphone, entre autres. Ce service est particulièrement utile pour stocker et gérer les informations des utilisateurs, facilitant ainsi l'accès sécurisé aux fonctionnalités de l'application.



4.1.4 TensorFlow Lite

TensorFlow Lite (TFLite) est une version allégée de la bibliothèque TensorFlow, optimisée pour les appareils mobiles et embarqués. Elle permet d'exécuter des modèles d'apprentissage automatique directement sur des smartphones Android, sans nécessiter une connexion constante à un serveur distant.

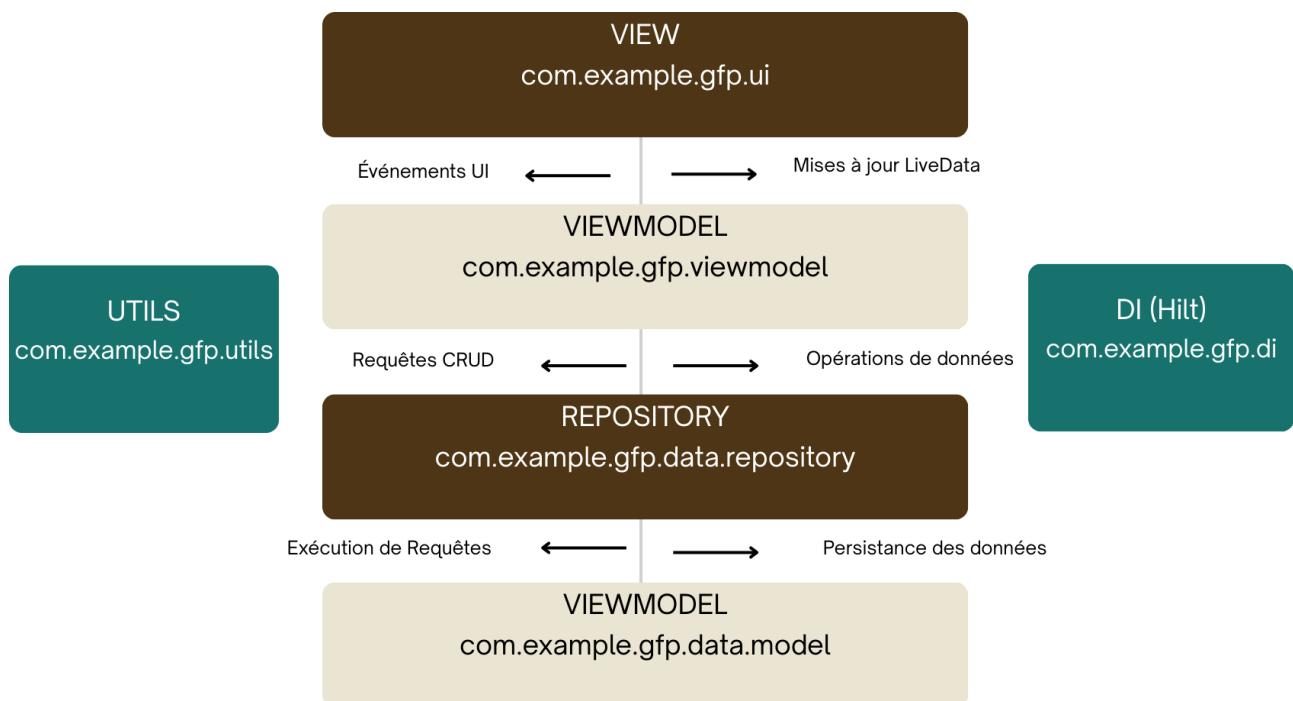


Dans notre projet, nous avons utilisé TensorFlow Lite pour intégrer un modèle pré-entraîné capable de fournir des recommandations budgétaires intelligentes. En fonction des catégories sélectionnées par l'utilisateur (alimentation, transport, santé, etc.), de la période (mois, jour) et d'autres paramètres, le modèle prédit des montants recommandés pour chaque catégorie de dépenses.

4.2 Architecture MVVM

L'architecture MVVM (Model–View–ViewModel) est un modèle architectural largement adopté dans le développement d'applications Android modernes. Elle vise à séparer les responsabilités entre les composants de l'interface utilisateur et la logique métier, afin d'améliorer la maintenabilité, la testabilité et la réutilisabilité du code.

Notre projet est structuré en plusieurs packages bien définis :

**FIGURE 4.1:** Architecture MVVM de l'application

4.2.1 Model

Cette couche contient toutes les entités métier persistées avec Realm, notre base de données locale. Les classes comme User, Transaction, Goal, Category, etc., y sont définies.

4.2.2 Repository

La couche Repository agit comme une interface entre les ViewModels et la base de données Realm. Chaque entité dispose de son propre repository (UserRepository, TransactionRepository, GoalRepository, etc.) permettant de centraliser les opérations CRUD et d'isoler la logique d'accès aux données.

4.2.3 ViewModel

Les ViewModels comme UserViewModel, CategoryViewModel, ChargesViewModel etc., exposent les données sous forme d'observables (LiveData) aux composants de l'interface utilisateur. Ils se chargent de la gestion de la logique métier et communiquent avec les Repositories pour obtenir ou persister les données. Cette couche garantit que les données survivent aux changements de configuration (rotation d'écran, etc.).

4.2.4 View

Les différentes Activity et Fragment composant l'interface utilisateur résident dans ce package : LoginActivity, AnalyseActivity, DefineGoalActivity, ReportsActivity, etc. Ces composants se contentent d'observer les ViewModel pour afficher les données à l'utilisateur, et de relayer ses interactions.

4.2.5 Utilitaires

On y retrouve des classes utilitaires comme PdfExporter, qui est responsable de la génération des rapports PDF à partir des transactions et objectifs.

4.2.6 Injection de dépendances

Le module AppModule gère l'injection des dépendances à l'aide de Hilt, permettant une instantiation centralisée et performante des Repository, SessionManager, Realm, etc.

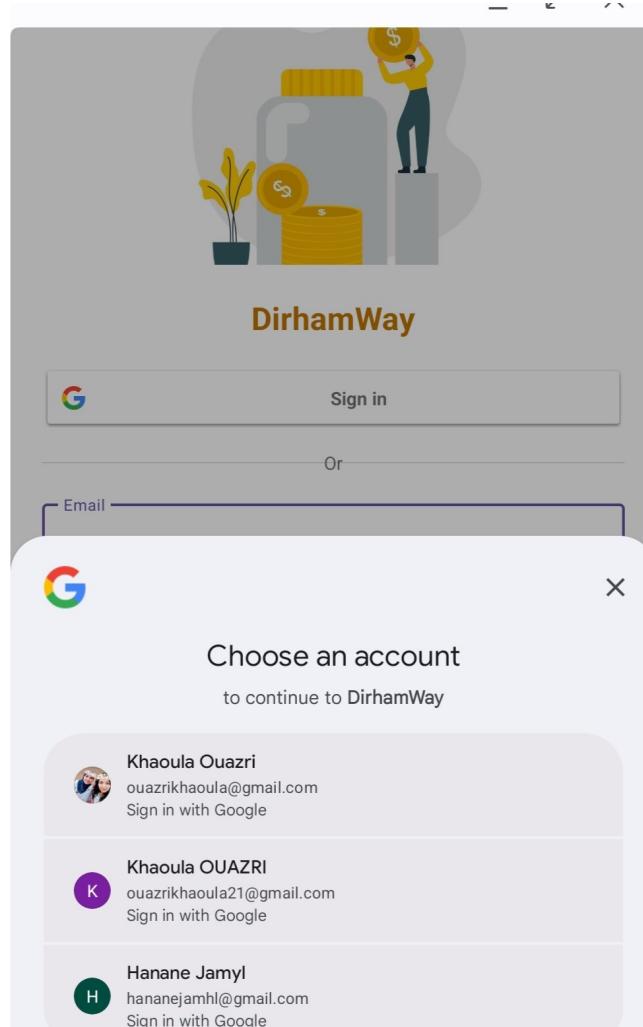
4.3 Exploration des interfaces utilisateur de DirhamWay

4.3.1 Interface d'authentification

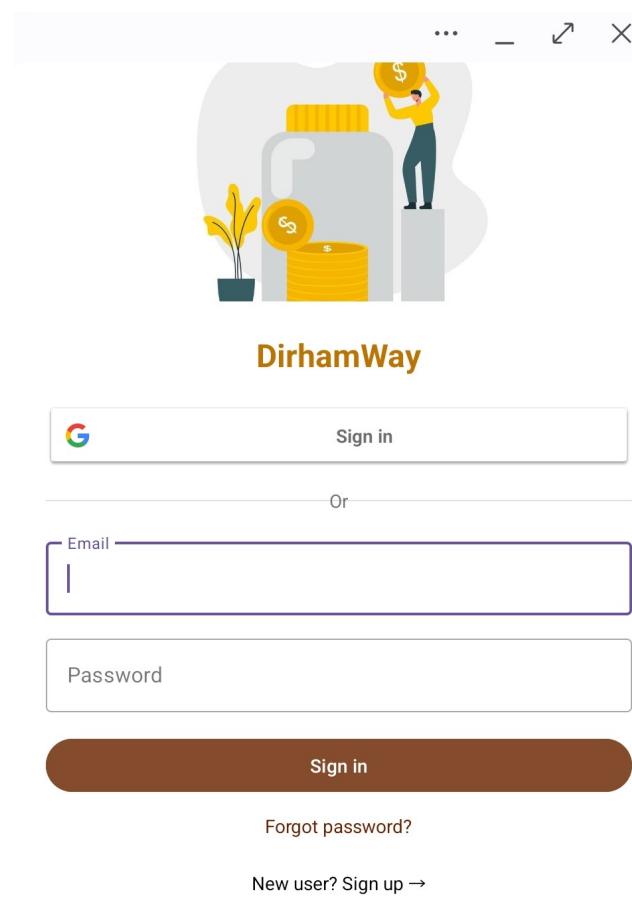
L'interface d'authentification de l'application **DirhamWay** constitue la porte d'entrée vers l'univers de gestion financière proposé par l'application. Elle est conçue pour être à la fois intuitive, moderne et accessible.

Deux modes d'authentification sont proposés à l'utilisateur :

- **Authentification via Google** : cette méthode permet à l'utilisateur de se connecter rapidement et en toute sécurité à l'aide de son compte Google. Elle repose sur l'intégration de *Firebase Authentication*, garantissant une gestion simplifiée de l'identité et une expérience utilisateur fluide.



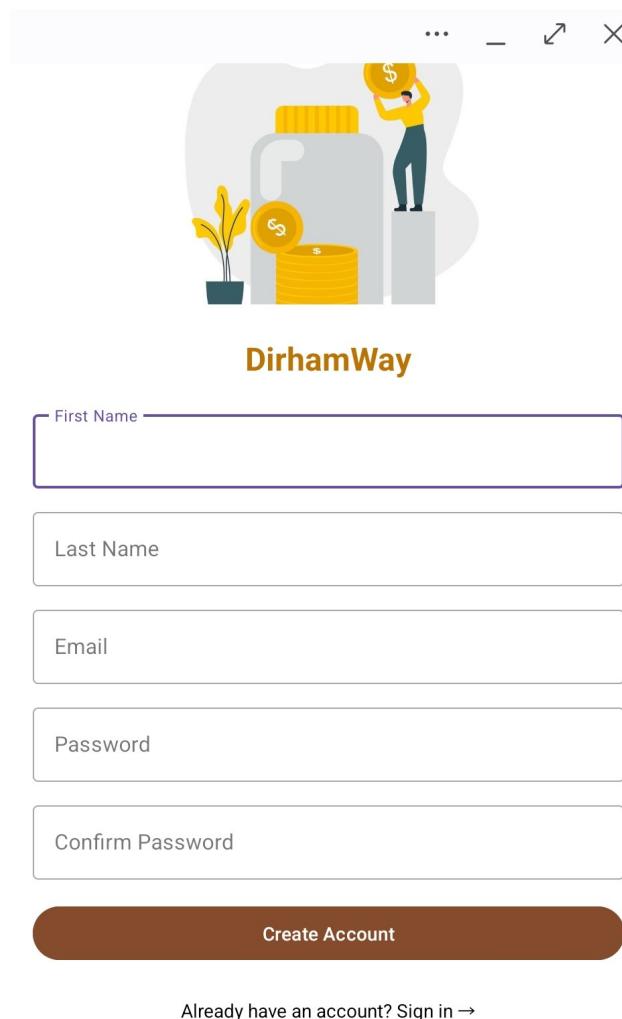
- **Authentification par email et mot de passe** : l'utilisateur peut également se connecter manuellement en renseignant son adresse email et son mot de passe. Ce mode convient aux utilisateurs préférant une méthode classique de connexion.



4.3.2 Interfaces d'inscription

L'application DirhamWay intègre une interface d'authentification moderne et épurée.

- **Authentification par email et mot de passe :** l'utilisateur peut créer un compte en renseignant son prénom, son nom, son adresse email et un mot de passe sécurisé. Lors de la connexion, ces informations sont vérifiées via Firebase Authentication. Cette méthode assure un contrôle strict de l'accès tout en restant simple à utiliser.

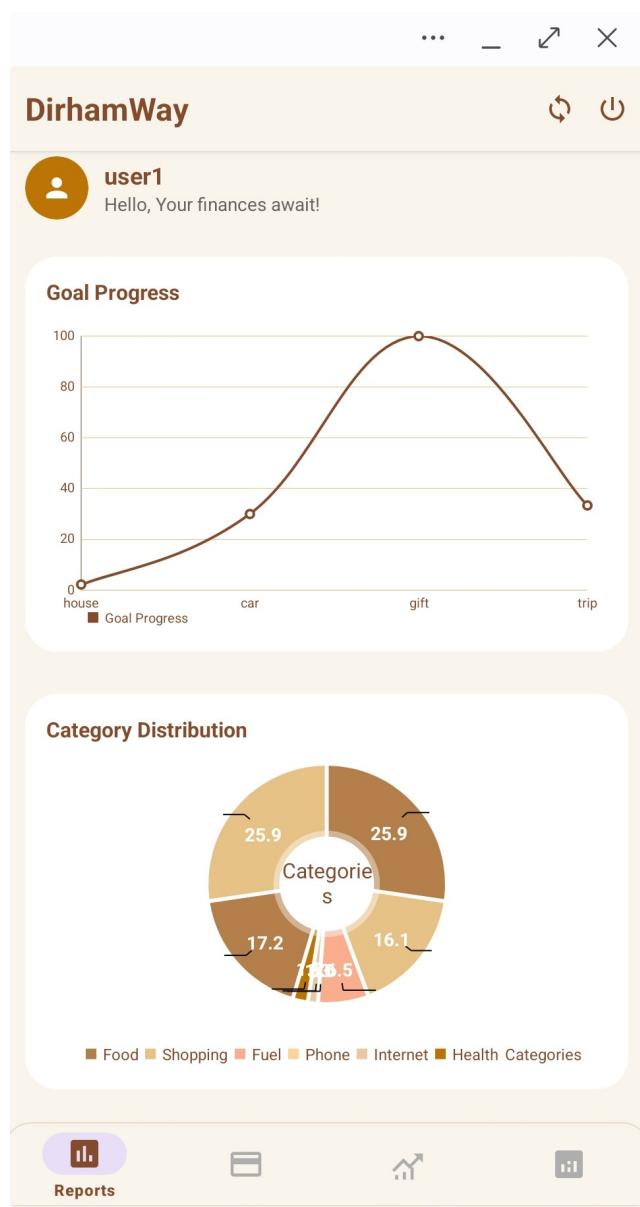


Une fois l'inscription terminée, l'utilisateur est accueilli par un écran de bienvenue personnalisé.

Ensuite, il est guidé à travers un processus de personnalisation de son profil financier :

- Sélection des catégories de dépenses pertinentes telles que la nourriture, le shopping, ou encore le logement.
- Indication de son budget mensuel global.
- Répartition de ce budget entre les différentes catégories sélectionnées, avec la possibilité de définir certaines dépenses comme fixes.

Ce parcours initial permet à l'application de configurer un tableau de bord personnalisé



Le tableau de bord de l'application DirhamWay offre à l'utilisateur un aperçu rapide et synthétique de sa situation financière :

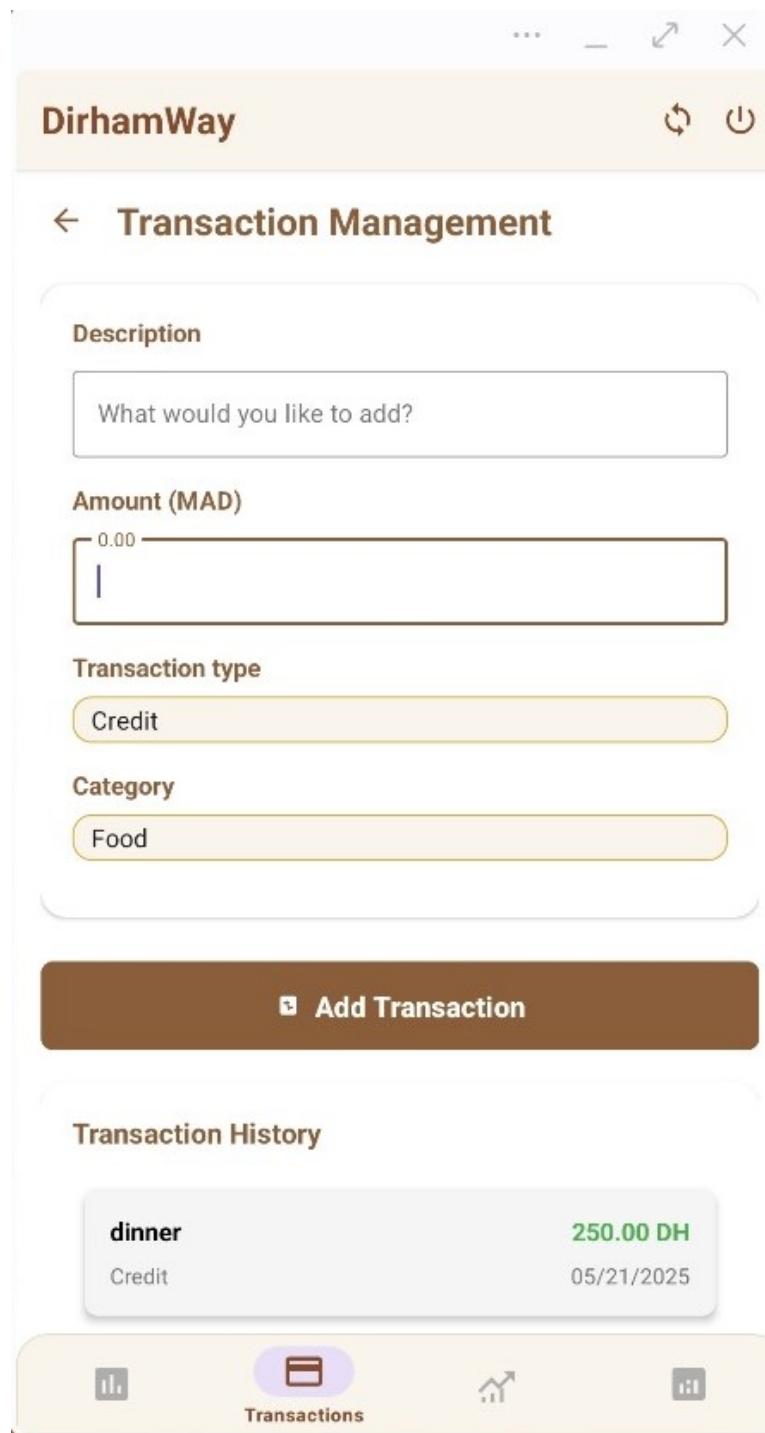
- **Progression des objectifs** : Une courbe illustrant l'évolution de l'épargne pour chaque objectif (ex. : maison, voiture, cadeau, voyage), permettant d'évaluer visuellement l'avancement.
- **Répartition par catégorie** : Un graphique en anneau présentant la distribution des dépenses par catégories (ex. : Alimentation, Shopping, Carburant, Téléphone, Internet, Santé). Cette vue aide à identifier les postes de dépenses les plus importants.
- **Message personnalisé** : Un message d'accueil dynamique affichant le nom de l'utilisateur.

4.3.3 Interface de gestion de transactions

L'interface **Transaction Management** permet aux utilisateurs de gérer manuellement leurs finances en toute simplicité. Elle leur offre la possibilité d'enregistrer aussi bien leurs revenus que leurs dépenses, en remplissant les champs suivants :

- **Description** : un champ libre pour décrire la nature de la transaction.
- **Montant (MAD)** : saisie du montant en dirhams.

- **Type de transaction :** choix entre une transaction de type revenu (crédit) ou dépense (débit).
- **Catégorie :** choix d'une catégorie prédéfinie (ex. : Alimentation, Transport, Loisirs, Santé, etc.) pour classer les dépenses.



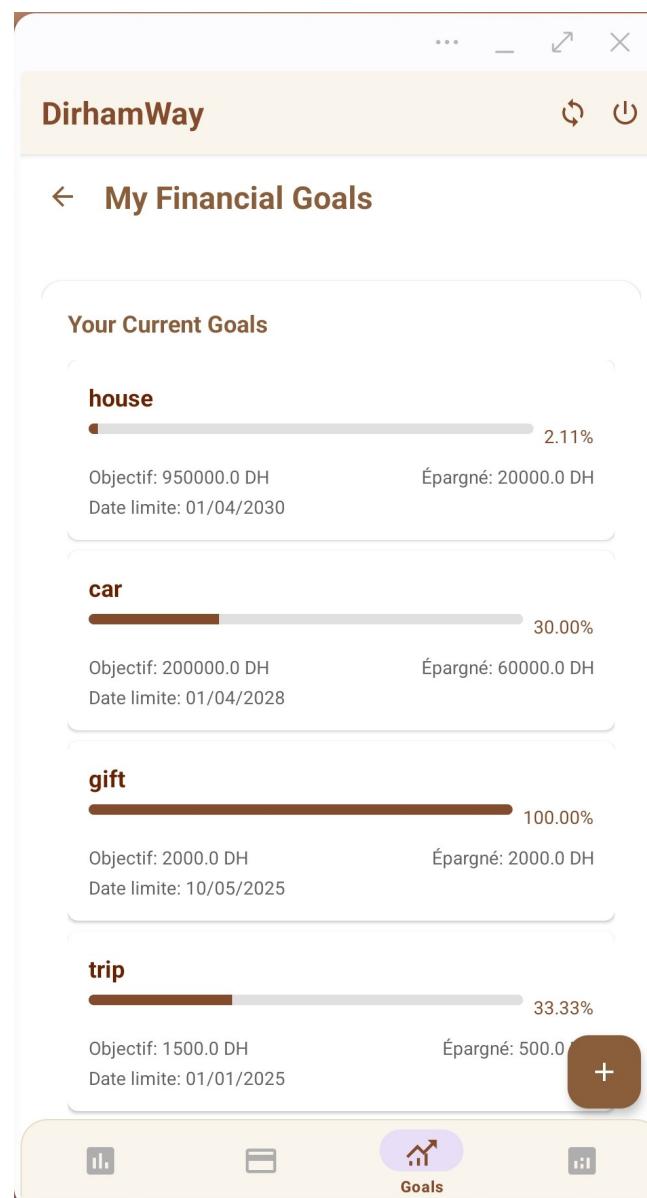
4.3.4 Interfaces de gestions des objectifs

La section **Gestion des objectifs** de l'application DirhamWay permet aux utilisateurs de définir, consulter, modifier et supprimer leurs objectifs d'épargne personnalisés.

Fonctionnalités principales :

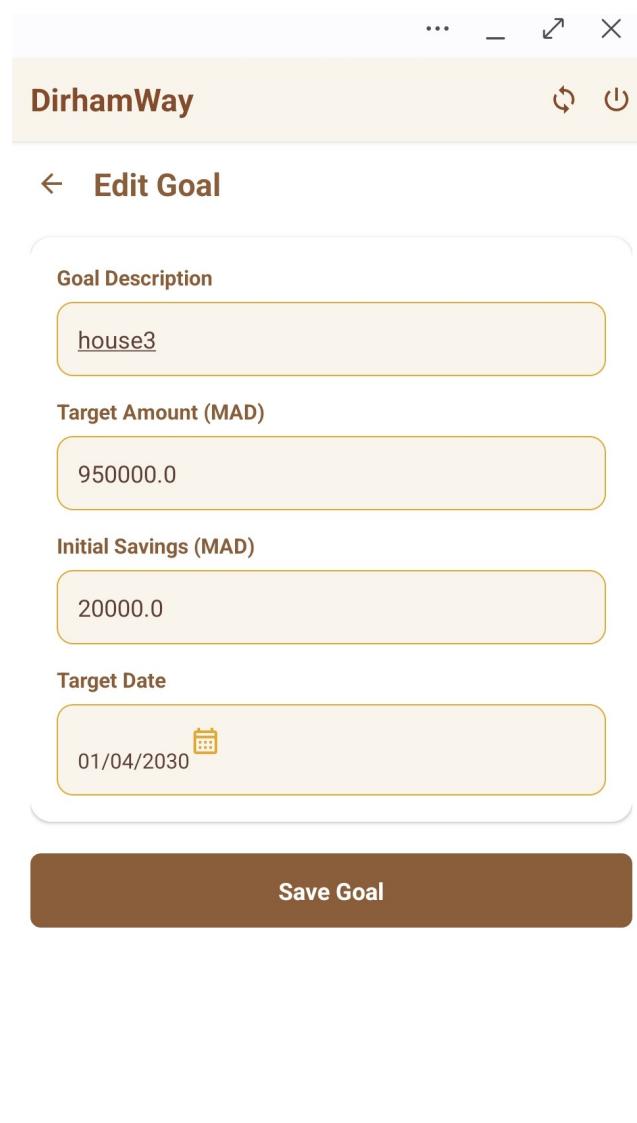
- **Affichage des objectifs existants :**

- La liste présente chaque objectif sous forme de carte ou ligne claire, incluant :
 - Le nom de l'objectif (ex. : "Voyage", "Achat PC"),
 - Le montant souhaité à épargner,
 - Le pourcentage d'évolution basé sur le montant déjà économisé,
 - La date limite définie par l'utilisateur.



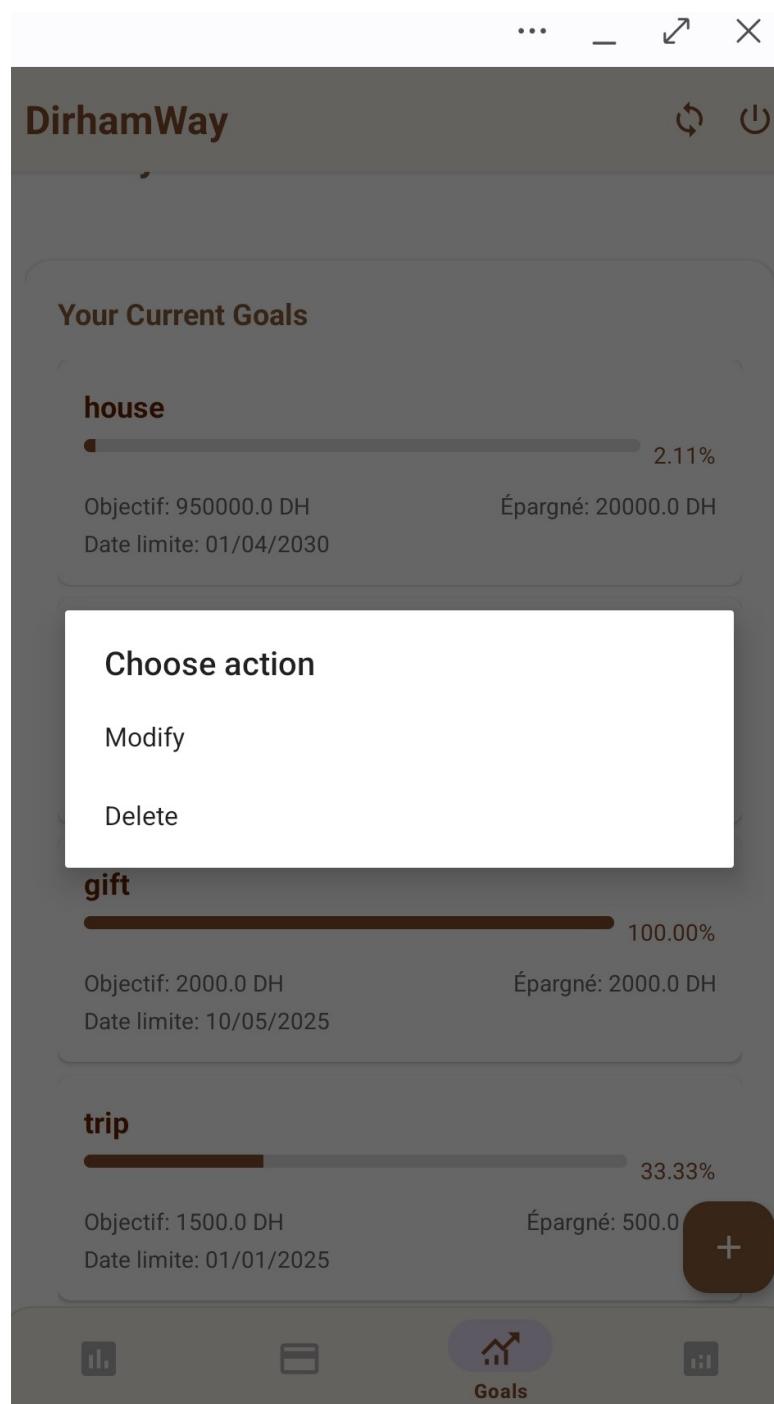
- **Modification d'un objectif :**

- En cliquant sur un objectif, l'application ouvre un formulaire prérempli contenant toutes les données de cet objectif.
- L'utilisateur peut alors modifier champ par champ selon ses besoins (ex. : changer le montant cible ou repousser la date limite).
- Après validation, les informations sont mises à jour en temps réel dans la base de données locale (Realm), et l'interface est automatiquement rafraîchie.



- **Suppression d'un objectif :**

- L'utilisateur peut également choisir de supprimer un objectif.
- Une confirmation est demandée pour éviter les suppressions accidentnelles.
- Une fois l'objectif supprimé, la liste est actualisée dynamiquement, supprimant l'entrée concernée sans recharger manuellement l'application.



4.3.5 Interface de la répartition intelligente des dépenses

La fonctionnalité Expense Distribution permet à l'utilisateur de répartir automatiquement son budget mensuel entre les catégories de dépenses sélectionnées, en s'appuyant sur un modèle prédictif basé sur TensorFlow Lite.

Fonctionnement :

- **Récupération du budget défini à l'inscription :**
 - Le budget mensuel de l'utilisateur, saisi lors de l'étape d'inscription, est automatiquement récupéré et affiché.
- **Sélection des catégories concernées :**

- L'utilisateur choisit les catégories de dépenses pertinentes pour lui (ex. : Nourriture, Assurance, Logement, etc.).
- Certaines catégories peuvent être marquées comme *fixes*, ce qui indique que leur montant ne doit pas être modifié par le modèle.

- **Prédiction automatique :**

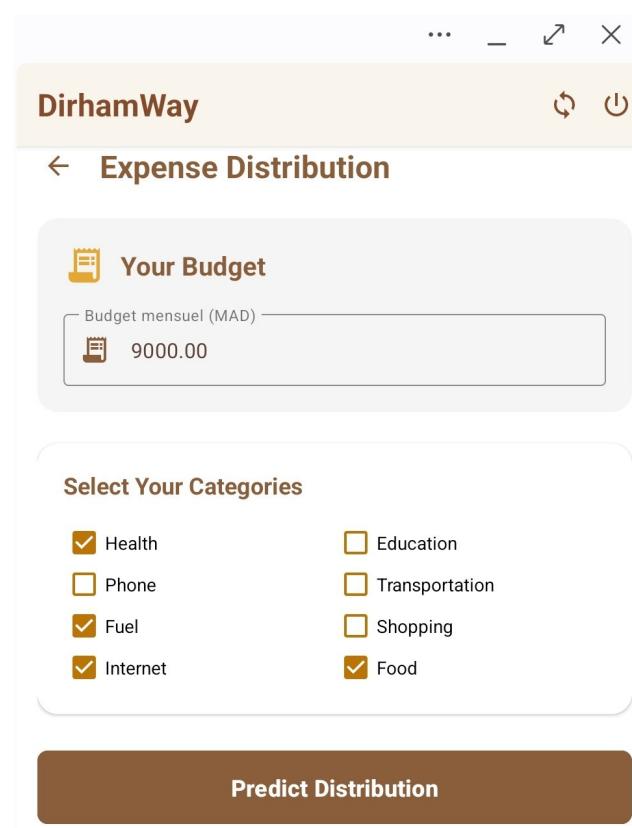
- En cliquant sur **Predict Distribution**, un modèle *TensorFlow Lite* répartit intelligemment le reste du budget entre les autres catégories, en tenant compte :
 - des catégories sélectionnées ;
 - des montants fixes imposés ;
 - des données historiques.

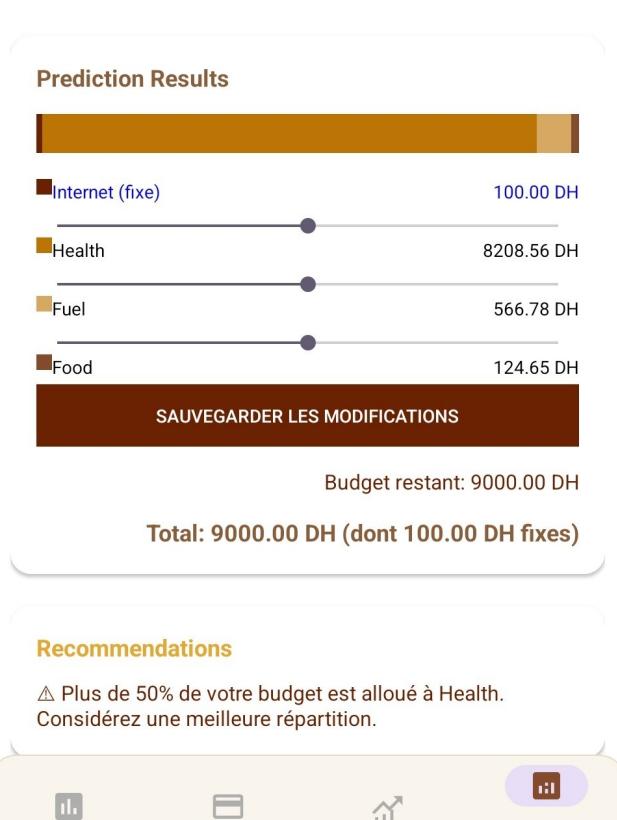
- **Ajustement manuel :**

- L'utilisateur garde la main sur la répartition : il peut modifier librement les montants prédits avant de valider.

- **Enregistrement de la répartition :**

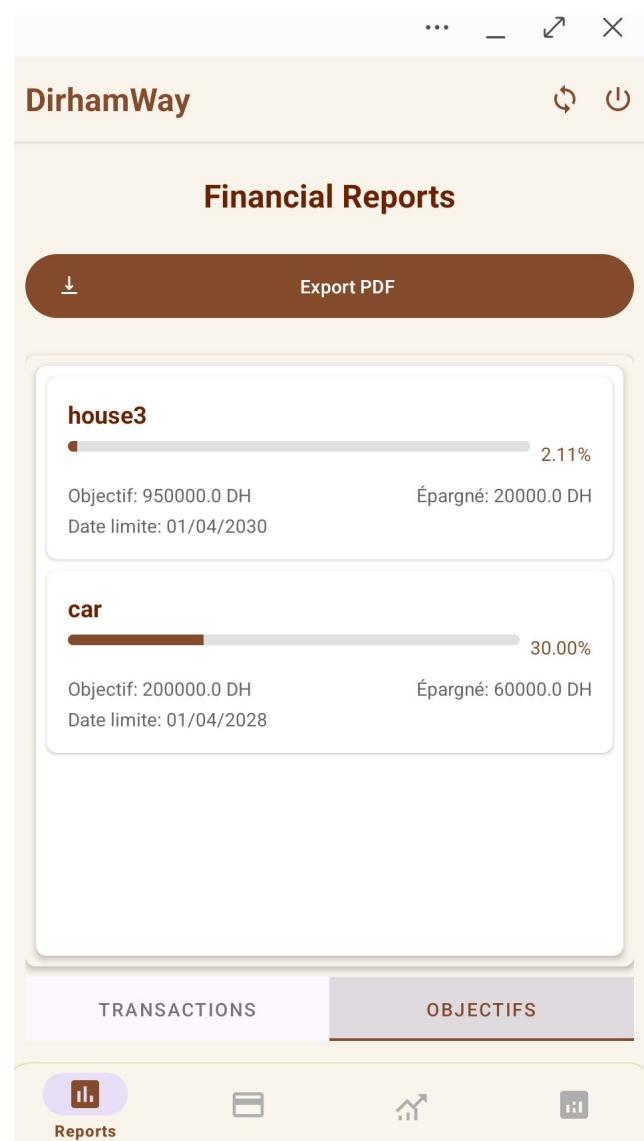
- La répartition validée est ensuite enregistrée dans la base de données *Realm*.





4.3.6 Interface des rapports financiers

La section **Financial Reports** de l'application **DirhamWay** fournit à l'utilisateur un aperçu global et détaillé de sa situation financière. Elle permet également l'exportation des rapports au format PDF.



L'utilisateur dispose d'un tableau de synthèse affichant les informations suivantes :

- **Total Income** : Somme des revenus enregistrés.
- **Total Expenses** : Total des dépenses.
- **Net Balance** : Solde net (revenus - dépenses).
- **Active Goals** : Nombre d'objectifs d'épargne actifs.

Aperçu Financier

Revenus

150.00 MAD

Dépenses

0.00 MAD

Solde Net

150.00 MAD

Objectifs Actifs

2

L'utilisateur peut suivre l'évolution de ses objectifs grâce à un tableau détaillé contenant :

- Le nombre total d'objectifs enregistrés.
- Le nombre d'objectifs complétés, actifs ou expirés.
- Le taux d'avancement global sous forme de barre de progression.

Objectifs Financiers

Objectif	Cible	Épargné	Progrès	Échéance	Statut
trip	1500.00 MAD	500.00 MAD	33.3%	01/01/2025	Expiré
car	200000.00 MAD	60000.00 MAD	30.0%	04/01/2028	Actif
house3	950000.00 MAD	20000.00 MAD	2.1%	04/01/2030	Actif
gift	2000.00 MAD	2000.00 MAD	100.0%	05/10/2025	Termin

L'utilisateur accède à l'historique détaillé de ses transactions, organisées par date. Chaque ligne contient :

- **Date** de l'opération,
- **Description**,
- **Montant**,
- **Catégorie** (ex. : Transport, Alimentation...),
- **Type** : crédit ou débit.

Historique des Transactions

Date	Description	Montant	Catégorie	Type
21 May	dinner	150.00 MAD	Non catégorisé	Credit

Cette fonctionnalité constitue un outil précieux pour une gestion budgétaire efficace, structurée et documentée.

Dans ce chapitre, nous avons détaillé les technologies et outils utilisés pour construire l'application DirhamWay, y compris l'intégration de base de données locale, l'authentification sécurisée des utilisateurs, et l'utilisation d'algorithmes de machine learning pour fournir des recommandations budgétaires. L'architecture MVVM garantit une séparation claire des responsabilités et une gestion optimale des données. Ce cadre technique permet à l'application de répondre efficacement aux exigences de performance et de scalabilité. Nous avons également présenté les captures d'écran de l'application, illustrant les interfaces et les fonctionnalités mises en œuvre grâce à cette architecture technique.

Conclusion

Ce projet d'application mobile de gestion financière personnelle a permis de répondre à un besoin croissant d'autonomie et de visibilité dans le suivi des dépenses quotidiennes. L'objectif principal était de fournir à l'utilisateur un outil simple, intuitif et intelligent lui permettant de gérer son budget, de planifier des objectifs d'épargne et de suivre ses transactions de manière personnalisée.

À travers une interface fluide et des fonctionnalités modulaires – telles que l'ajout de transactions, la catégorisation des dépenses, la définition d'objectifs financiers, ou encore la génération de rapports PDF – l'application offre une expérience utilisateur centrée sur la clarté, la personnalisation et la praticité.

Bien que les résultats obtenus soient satisfaisants, le projet n'a pas été exempt de difficultés. La gestion des composants liés à la base de données Realm, l'intégration de l'intelligence artificielle pour la prédition budgétaire via TensorFlow Lite, ainsi que la coordination de la navigation entre les différentes interfaces ont représenté des défis techniques majeurs, nécessitant rigueur, adaptation et esprit de synthèse.

Pour les perspectives d'évolution, plusieurs pistes peuvent être envisagées : l'ajout d'une synchronisation cloud entre appareils, une meilleure visualisation graphique des données, ou encore l'intégration de notifications intelligentes pour prévenir les dérives budgétaires. De plus, l'exploitation de l'IA pour recommander des actions d'épargne plus pertinentes en fonction du comportement utilisateur constituerait une évolution naturelle du système.

Ce projet a donc représenté une opportunité concrète d'appliquer les principes de développement mobile, de modélisation logicielle, et d'expérience utilisateur, tout en apportant une solution utile et réaliste aux défis de gestion financière individuelle.