RAPPORT DE PROJET

Application Mobile de Gestion des Dépenses

Expense Tracker App

Titre du projet :

Application Mobile de Gestion des Dépenses Personnelles

Réalisé par :

Abderrahmane Ajili

Semestre :

Semestre 1 – Année universitaire 2025–2026

Établissement :

ISIMM – Institut Supérieur d'Informatique et de Mathématiques de Monastir

Résumé / Abstract :

Ce projet consiste à développer une application mobile de gestion des dépenses personnelles utilisant React Native et Expo Router. L'application permet aux utilisateurs d'enregistrer, gérer et analyser leurs dépenses, avec des statistiques hebdomadaires, mensuelles et annuelles.

Les fonctionnalités principales incluent l'authentification sécurisée, la gestion des dépenses, la visualisation des statistiques, et la gestion du profil utilisateur. Les données sont stockées et synchronisées en temps réel via Firebase, tandis que Cloudinary est utilisé pour le stockage des images.

L'objectif de ce projet est de proposer une solution intuitive, performante et multiplateforme (Android et iOS), tout en mettant en pratique les compétences acquises en développement mobile, architecture logicielle et intégration de services cloud. Les résultats démontrent la faisabilité d'une application efficace pour la gestion personnelle des finances et offrent des perspectives d'amélioration, telles que le mode hors ligne et les notifications intelligentes.

Table des matières

Introduction

Avec la généralisation des smartphones et l'évolution rapide des technologies numériques, les applications mobiles sont devenues des outils essentiels pour la gestion des activités quotidiennes. Parmi elles, les applications de gestion financière personnelle permettent aux utilisateurs de mieux suivre leurs dépenses et d'optimiser leur budget.

Ce projet a pour objectif de concevoir et de développer une application mobile de gestion des dépenses basée sur React Native et Expo Router. L'application permet aux utilisateurs d'enregistrer leurs dépenses, de consulter des statistiques détaillées (hebdomadaires, mensuelles et annuelles) et de gérer leur profil personnel. Les données sont stockées et synchronisées à l'aide de Firebase, tandis que Cloudinary est utilisé pour la gestion des images.

L'objectif principal de ce travail est de proposer une solution multiplateforme (Android et iOS), performante et intuitive, tout en mettant en pratique les compétences acquises en développement mobile, en conception logicielle et en intégration de services cloud. Ce projet constitue ainsi une expérience formatrice, tant sur le plan technique que méthodologique.

Analyse des besoins

1. Utilisateurs cibles

L'application s'adresse à toute personne souhaitant suivre et gérer ses dépenses personnelles, notamment :

- Étudiants, travailleurs ou indépendants.
- Utilisateurs de smartphones Android et iOS.
- Personnes recherchant une interface simple et intuitive pour gérer leur budget.

2. Fonctionnalités attendues

Les fonctionnalités principales identifiées pour répondre aux besoins des utilisateurs sont :

- Authentification et gestion des comptes utilisateurs : création, connexion et sécurisation des profils.
- Gestion des dépenses : ajout, modification, suppression et catégorisation des dépenses.
- Visualisation des statistiques : analyses hebdomadaires, mensuelles et annuelles.
- Profil utilisateur : personnalisation des informations et gestion des paramètres.
- Gestion des images : possibilité d'ajouter des reçus ou factures via Cloudinary.
- Interface fluide et responsive : expérience utilisateur optimisée sur tous les types d'appareils.

3. Contraintes techniques

Pour assurer la qualité et la faisabilité de l'application, les contraintes suivantes ont été prises en compte :

- Multiplateforme : l'application doit fonctionner sur Android et iOS.
- Stockage et synchronisation des données en temps réel : utilisation de Firebase.
- Sécurité des données utilisateurs : protection des informations sensibles.
- Performance et fluidité : interface responsive avec animations optimisées via React Native Reanimated.
- Extensibilité : architecture permettant l'ajout futur de nouvelles fonctionnalités (notifications, mode hors ligne, export de données).

Spécifications / Cahier des charges

1. Fonctionnalités minimales

Les fonctionnalités essentielles à implémenter pour que l'application soit opérationnelle sont :

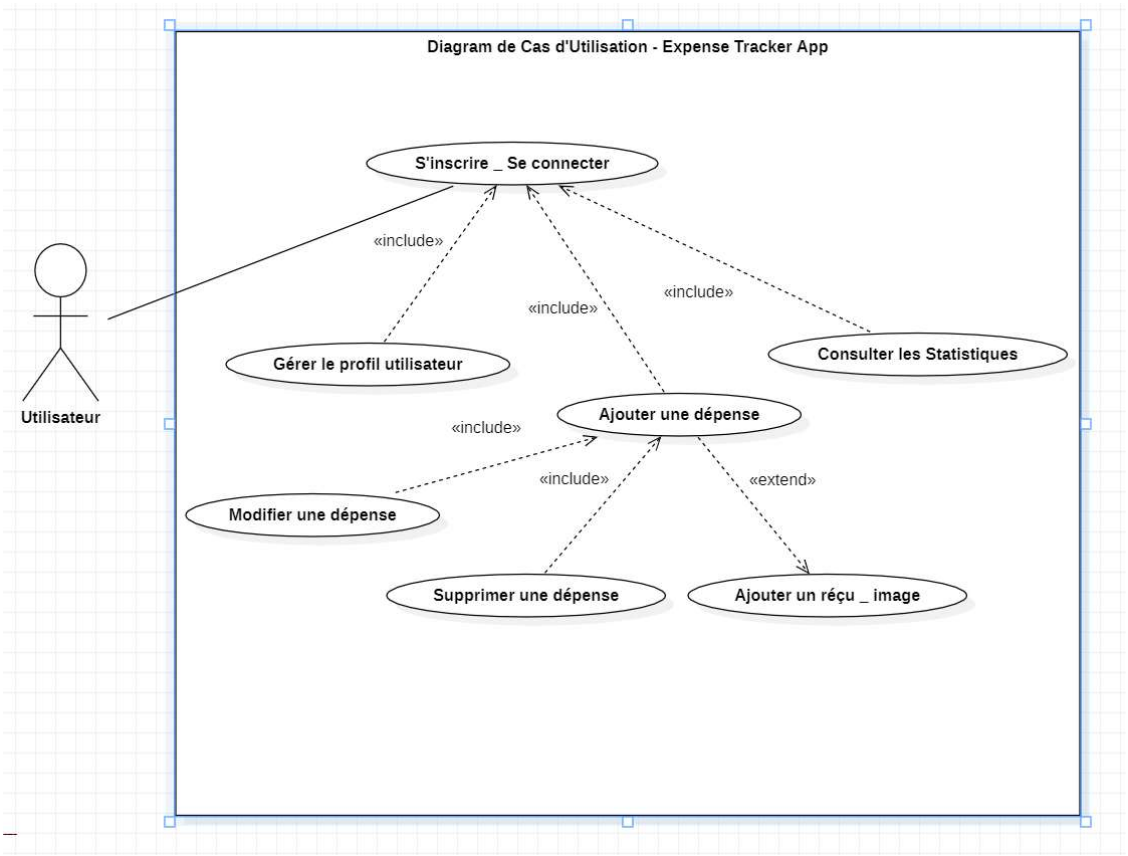
- Authentification : création de compte, connexion et déconnexion sécurisée.
- Gestion des dépenses : ajout, modification et suppression des dépenses.
- Visualisation des dépenses : liste des transactions avec tri et filtrage.
- Statistiques de base : affichage des dépenses par semaine, mois et année.
- Profil utilisateur : consultation et modification des informations personnelles.

2. Fonctionnalités optionnelles

Ces fonctionnalités améliorent l'expérience utilisateur et enrichissent l'application :

- Upload d'images : possibilité d'ajouter des reçus ou factures via Cloudinary.
- Animations : interface animée grâce à React Native Reanimated pour plus de fluidité.
- Design responsive : optimisation de l'affichage sur différents écrans.

3.Diagramme de cas d’utilisation



Conception et architecture

Choix technologique

Pour le développement de l'application Expense Tracker, nous avons choisi de créer une application hybride multiplateforme pour les raisons suivantes :

1. Multiplateforme :
    - Une seule base de code permet de déployer l'application à la fois sur Android et iOS, ce qui réduit le temps de développement et facilite la maintenance.
    - L'option Web pourrait être envisagée dans le futur grâce à des frameworks compatibles (comme React Native Web).
2. Framework choisi : React Native avec Expo Router
    - React Native (JavaScript / TypeScript) est particulièrement adapté pour les développeurs ayant des bases en web.
    - Offre une interface native et performante sur Android et iOS.
    - Permet l'intégration facile de bibliothèques modernes comme React Native Reanimated pour les animations et Phosphor Icons pour les icônes.
    - Compatible avec les services cloud comme Firebase pour la gestion des données et Cloudinary pour le stockage des images.

Diagrammes UML

1. Diagramme de classes

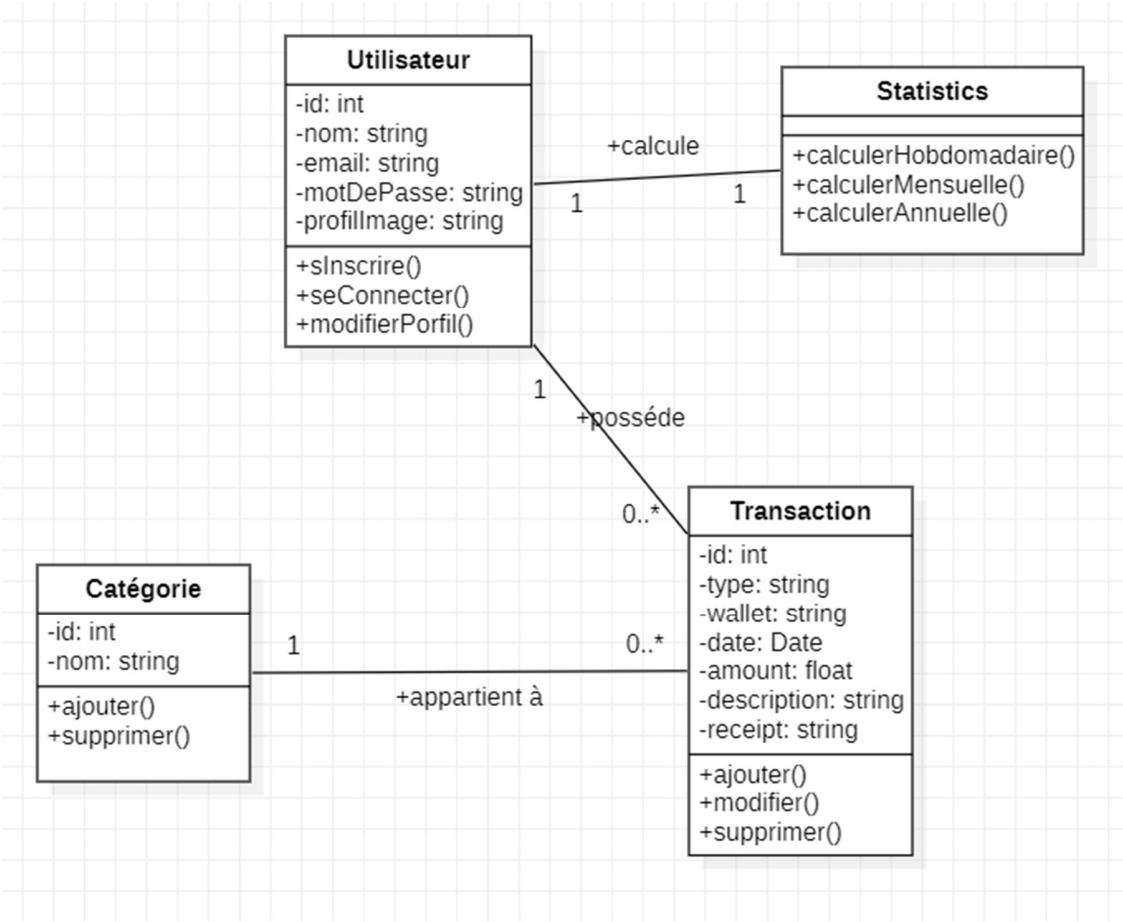
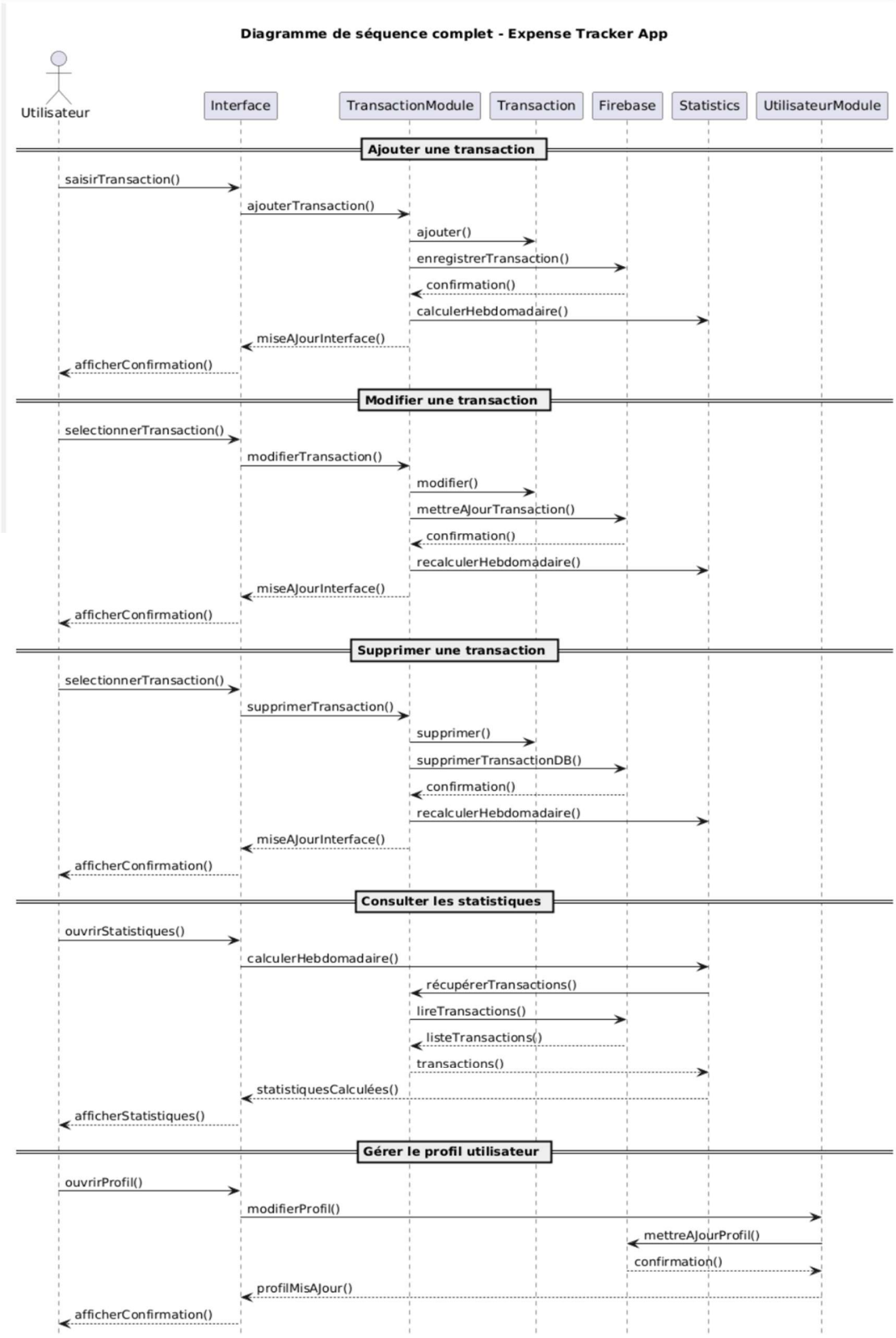


Diagramme de séquence



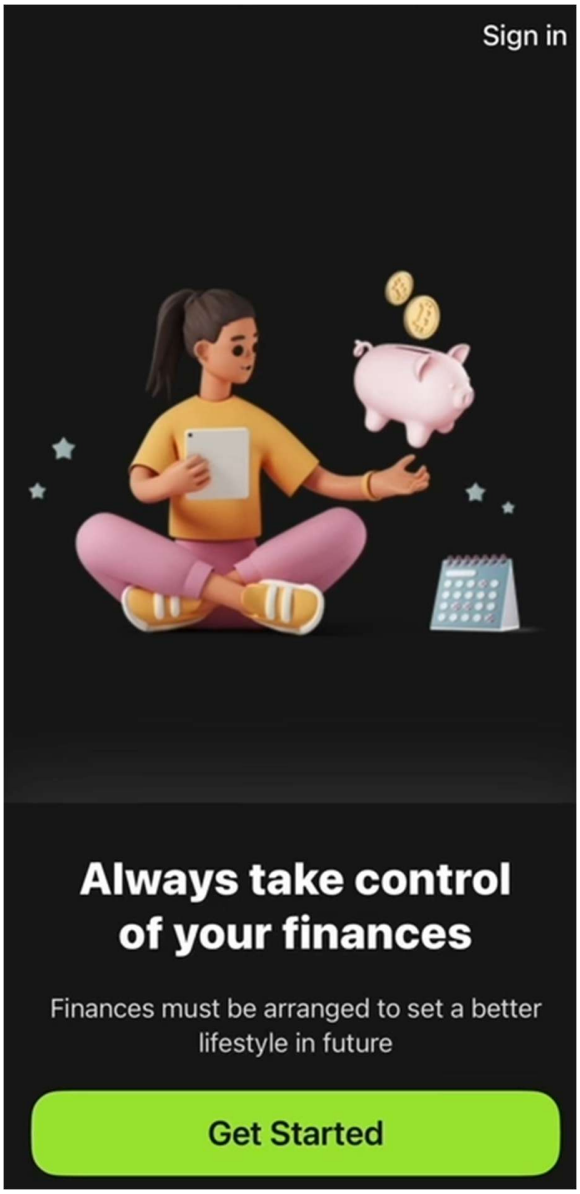
Développement et réalisation

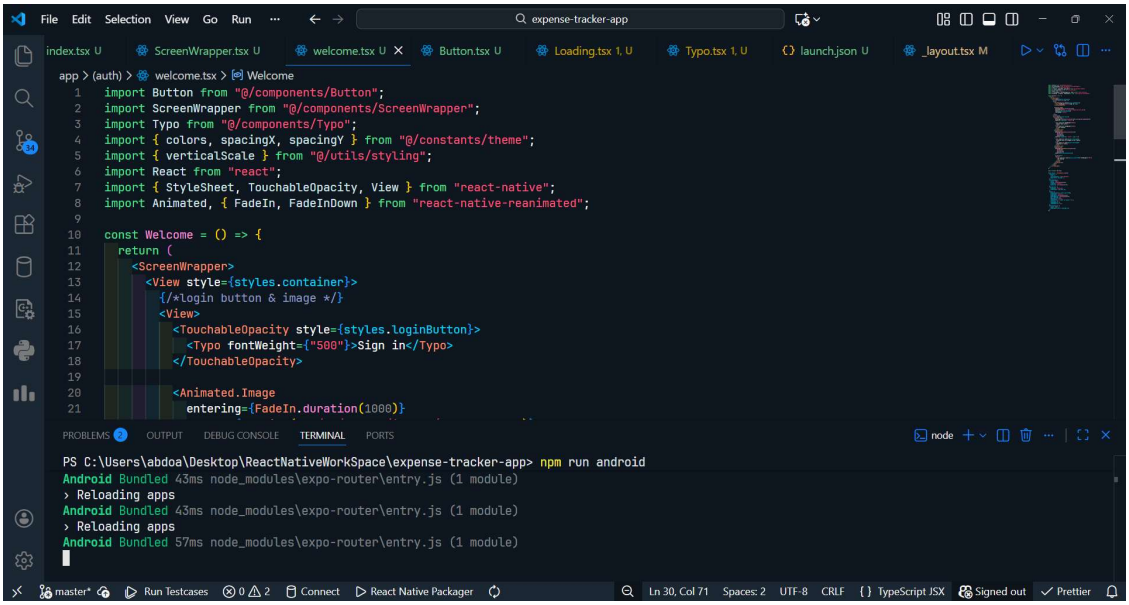
L'application Expense Tracker a été développée avec React Native et Expo Router, selon une architecture modulaire permettant une maintenance et une extension faciles.

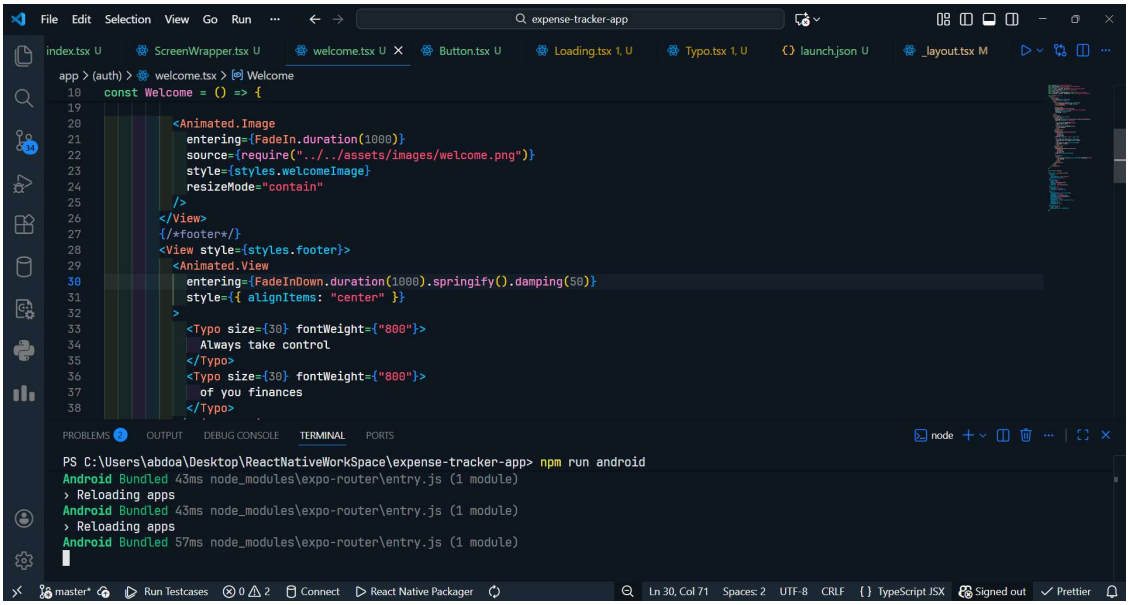
Modules principaux :

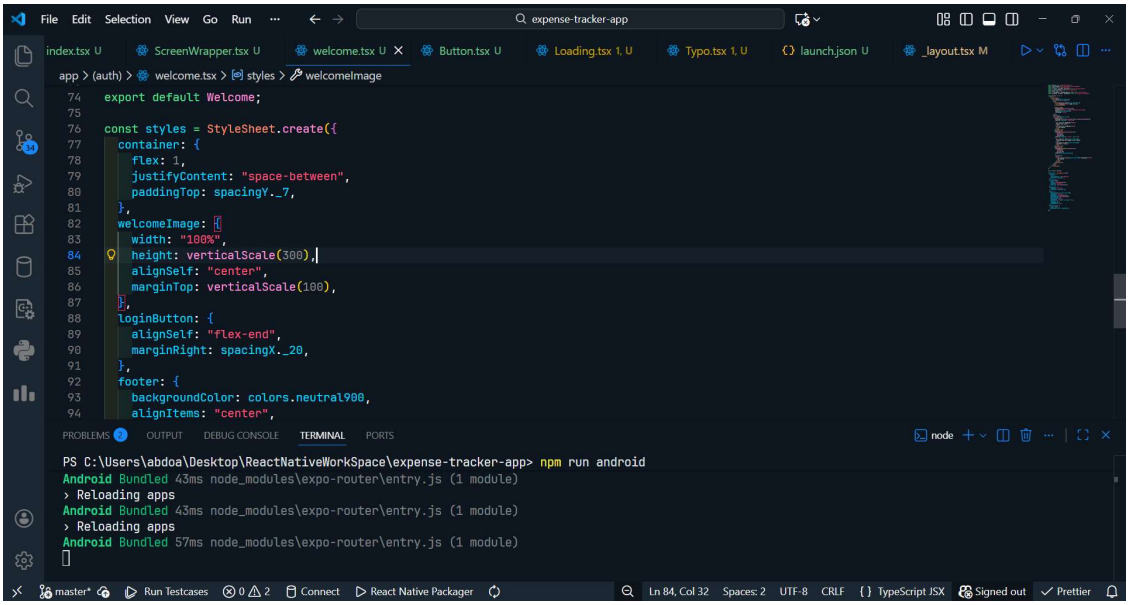
1. Authentification : inscription et connexion des utilisateurs via Firebase.
2. Gestion des transactions : ajout, modification et suppression des revenus et dépenses.
3. Statistiques : calcul des dépenses/revenus hebdomadaires, mensuelles et annuelles.
4. Profil utilisateur : modification des informations personnelles et photo de profil.
5. Catégories : création et suppression de catégories personnalisées

Captures d’écran de l’application, code source significatif (extraits) :




File Edit Selection View Go Run
expense-tracker-app
index.tsx U
ScreenWrapper.tsx U
welcome.tsx U
Button.tsx U
Loading.tsx 1, U
Typo.tsx 1, U
launch.json U
_layout.tsx M
app > (auth) > welcome.tsx > Welcome
import Button from "@/components/Button";
import ScreenWrapper from "@/components/ScreenWrapper";
import Typo from "@/components/Typo";
import { colors, spacingX, spacingY } from "@/constants/theme";
import { verticalScale } from "@/utils/styling";
import React from "react";
import { StyleSheet, TouchableOpacity, View } from "react-native";
import Animated, { FadeIn, FadeInDown } from "react-native-reanimated";
const Welcome = () => {
return (
<ScreenWrapper>
<View style={styles.container}>
{/*login button & image */}
<View>
<TouchableOpacity style={styles.loginButton}>
<Typo fontWeight={"500"}>Sign in</Typo>
</TouchableOpacity>
<Animated.Image
entering={FadeIn.duration(1000)}
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
PS C:\Users\abdoa\Desktop\ReactNativeWorkSpace\expense-tracker-app> npm run android
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 57ms node_modules\expo-router\entry.js (1 module)
master* Run Testcases 0 2 Connect React Native Packager
Ln 30, Col 71 Spaces: 2 UTF-8 CRLF { } TypeScript JSX Signed out Prettier


File Edit Selection View Go Run
expense-tracker-app
index.tsx U
ScreenWrapper.tsx U
welcome.tsx U
Button.tsx U
Loading.tsx 1, U
Typo.tsx 1, U
launch.json U
_layout.tsx M
app > (auth) > welcome.tsx > Welcome
const Welcome = () => {
<Animated.Image
entering={FadeIn.duration(1000)}
source={require("../../assets/images/welcome.png")}
style={styles.welcomeImage}
resizeMode="contain"
/>
</View>
{/*footer*/}
<View style={styles.footer}>
<Animated.View
entering={FadeInDown.duration(1000).springify().damping(50)}
style={{ alignItems: "center" }}
>
<Typo size={30} fontWeight={"800"}>
Always take control
</Typo>
<Typo size={30} fontWeight={"800"}>
of you finances
</Typo>
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
PS C:\Users\abdoa\Desktop\ReactNativeWorkSpace\expense-tracker-app> npm run android
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 57ms node_modules\expo-router\entry.js (1 module)
master* Run Testcases 0 2 Connect React Native Packager
Ln 30, Col 71 Spaces: 2 UTF-8 CRLF { } TypeScript JSX Signed out Prettier


File Edit Selection View Go Run
expense-tracker-app
index.tsx U
ScreenWrapper.tsx U
welcome.tsx U
Button.tsx U
Loading.tsx 1, U
Typo.tsx 1, U
launch.json U
_layout.tsx M
app > (auth) > welcome.tsx > styles > welcomeImage
export default Welcome;
const styles = StyleSheet.create({
container: {
flex: 1,
justifyContent: "space-between",
paddingTop: spacingY._7,
},
welcomeImage: {
width: "100%",
height: verticalScale(300),
alignSelf: "center",
marginTop: verticalScale(100),
},
loginButton: {
alignSelf: "flex-end",
marginRight: spacingX._20,
},
footer: {
backgroundColor: colors.neutral900,
alignItems: "center",
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
node
PS C:\Users\abdoa\Desktop\ReactNativeWorkSpace\expense-tracker-app> npm run android
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 43ms node_modules\expo-router\entry.js (1 module)
› Reloading apps
Android Bundled 57ms node_modules\expo-router\entry.js (1 module)
master*
Run Testcases
0 2
Connect
React Native Packager
Ln 84, Col 32
Spaces: 2
UTF-8
CRLF
{ } TypeScript JSX
Signed out
Prettier

Tests et validation

Afin de garantir la fiabilité et le bon fonctionnement de l'application Expense Tracker, plusieurs scénarios de test ont été réalisés sur les principales fonctionnalités.

Scénarios de test

- Authentification : test de l'inscription et de la connexion avec des données valides et invalides.
- Gestion des transactions : ajout, modification et suppression de transactions avec vérification de la mise à jour en temps réel.
- Statistiques : vérification du calcul correct des dépenses et revenus (hebdomadaire, mensuel et annuel).
- Profil utilisateur : modification des informations personnelles et mise à jour de la photo de profil.

Résultats

Les tests ont confirmé le bon fonctionnement global de l'application. Les données sont correctement enregistrées dans Firebase et les statistiques sont mises à jour automatiquement après chaque action.

Problèmes rencontrés et solutions

- Problème de synchronisation des données : résolu par l'utilisation de mises à jour en temps réel via Firebase.
- Gestion des erreurs de saisie : ajout de validations pour éviter les champs vides ou incorrects.

- Performance lors du chargement des données : amélioration grâce à l'optimisation des requêtes et au chargement conditionnel.

Conclusion et perspectives

Ce projet a permis de mettre en pratique les compétences acquises en développement mobile et en gestion de bases de données cloud.
L'application répond aux besoins essentiels de gestion financière personnelle.

Améliorations futures :

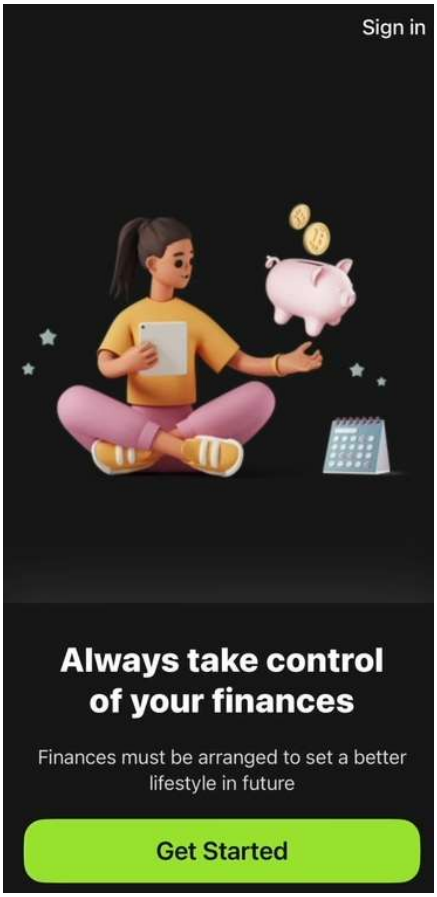
- Exportation des données en PDF / Excel.
- Mode hors ligne.
- Notifications intelligentes.
- Analyse prédictive des dépenses.

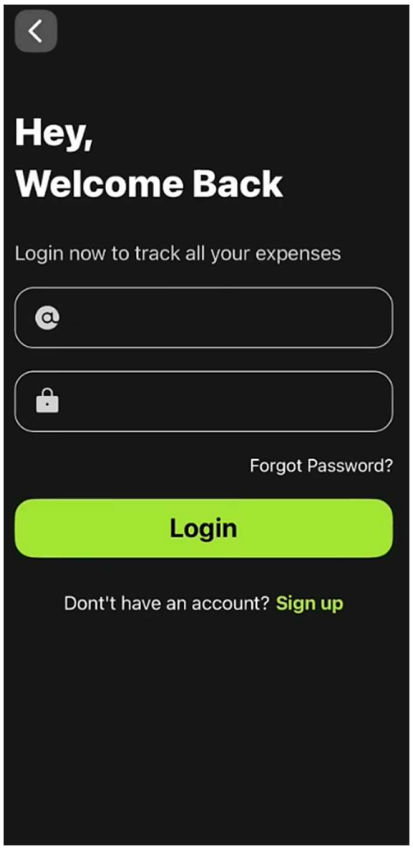
Bibliographie / Références

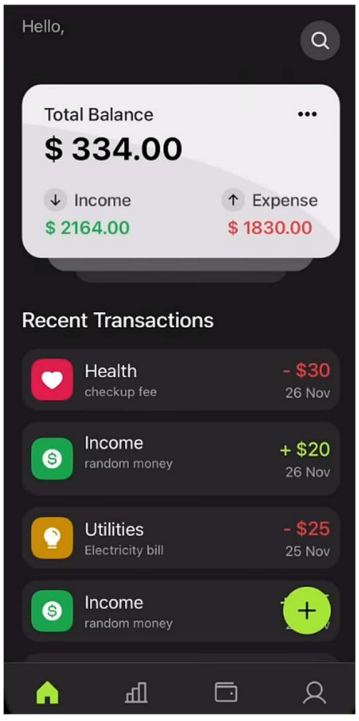
- https://reactnative.dev
- https://expo.dev
- https://firebase.google.com
- https://cloudinary.com
- Documentation officielle React Native

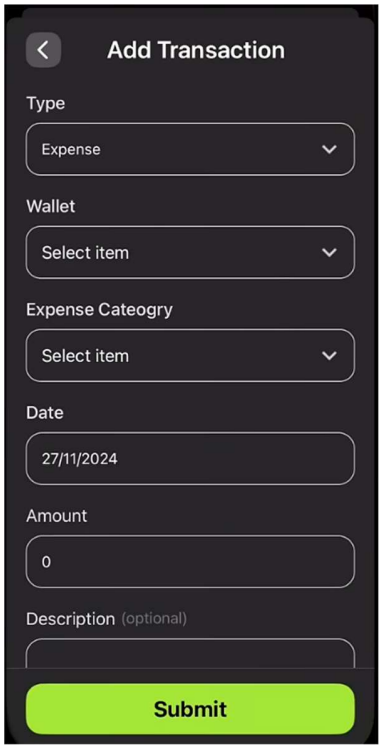
12. Annexes

- Code source complet (sera déposé ultérieurement).
- Diagrammes UML détaillés (déjà présentés précédemment).
- Captures d'écran supplémentaires.


Sign in
Always take control of your finances
Finances must be arranged to set a better lifestyle in future
Get Started


Hey,
Welcome Back
Login now to track all your expenses
Forgot Password?
Login
Dont't have an account? Sign up


Hello,
Total Balance
$ 334.00
Income
$ 2164.00
Expense
$ 1830.00
Recent Transactions
Health
checkup fee
- $30
26 Nov
Income
random money
+ $20
26 Nov
Utilities
Electricity bill
- $25
25 Nov
Income
random money


Add Transaction
Type
Expense
Wallet
Select item
Expense Cateogry
Select item
Date
27/11/2024
Amount
0
Description (optional)
Submit