ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

HONORIS EDUCATIONAL NETWORK

Rapport de projet

Application Mobile

Flutter

Sous le thème

APPLICATION QUIZ EDUCATIVE

Flutter App

Réalisé par :
HASSAR El Mehdi
FILALA Yassine

Encadrant :
Mr DEROUSS Anass

Table des matières

Table des figures

Résumé

Ce rapport présente le travail réalisé dans le cadre d'un projet académique portant sur le développement d'une application mobile éducative de type quiz, conçue à l'aide du framework **Flutter** et du langage **Dart**.

L'objectif principal de ce projet est de concevoir une application interactive permettant aux utilisateurs de répondre à des questions à choix multiples, d'évaluer leurs connaissances et d'obtenir un score final de manière intuitive et ergonomique.

La réalisation du projet a suivi plusieurs étapes, incluant l'analyse des besoins, la conception de l'architecture de l'application, le développement de l'interface utilisateur ainsi que l'implémentation de la logique de gestion des questions et du score. Une attention particulière a été portée à la simplicité d'utilisation, à la navigation entre les écrans et à la structuration du code.

Des tests fonctionnels ont été effectués afin de vérifier le bon fonctionnement de l'application et la conformité des fonctionnalités développées par rapport aux objectifs fixés. Les résultats obtenus montrent que l'application répond aux exigences du projet et constitue une base évolutive pour de futures améliorations, telles que l'ajout de niveaux de difficulté, l'intégration d'une base de données ou le support multi-plateformes.

Mots-clés : Application mobile, Quiz éducatif, Flutter, Dart, Projet académique, Développement mobile.

Introduction

Contexte et importance des applications mobiles éducatives

À l'ère de la transformation numérique, marquée par la généralisation des smartphones et des applications mobiles, les solutions éducatives numériques occupent une place de plus en plus importante dans l'apprentissage et l'évaluation des connaissances. Les applications mobiles éducatives permettent d'offrir des expériences interactives, accessibles et adaptées aux besoins des utilisateurs, favorisant ainsi un apprentissage ludique et progressif.

Le développement mobile multiplateforme, notamment à travers le framework **Flutter**, s'impose comme une solution moderne et efficace pour concevoir des applications performantes, maintenables et ergonomiques. Flutter permet de créer des interfaces riches et réactives tout en optimisant le temps de développement grâce à une base de code unique.

Objectifs du projet académique

Les objectifs principaux de ce projet académique sont les suivants :
— Concevoir et développer une application mobile éducative de type quiz.
— Mettre en pratique les concepts fondamentaux du développement mobile avec Flutter.
— Implémenter une interface utilisateur intuitive et ergonomique.
— Gérer la navigation entre les écrans et l'état de l'application.
— Mettre en œuvre un système de questions à choix multiples avec calcul du score.

Méthodologie générale

La réalisation de ce projet a suivi une démarche structurée, organisée en plusieurs étapes :
— **Analyse et documentation initiale** : étude des applications éducatives existantes et des fonctionnalités essentielles d'un quiz interactif.
— **Conception** : définition de l'architecture de l'application, organisation des écrans et conception de l'interface utilisateur.
— **Préparation de l'environnement** : installation et configuration de Flutter, des outils de développement et des émulateurs.
— **Développement** : implémentation des interfaces et de la logique du quiz en utilisant le langage Dart.
— **Tests et validation** : vérification du bon fonctionnement des fonctionnalités, correction des anomalies et amélioration de l'ergonomie.
— **Synthèse et restitution** : rédaction du rapport et présentation des résultats obtenus.

Chapitre 1

Contexte et problématique

Avec l'évolution rapide des technologies numériques et la démocratisation des smartphones, les applications mobiles sont devenues des outils incontournables dans de nombreux domaines, notamment l'éducation et la formation. Les méthodes d'apprentissage traditionnelles tendent à être complétées, voire remplacées, par des solutions numériques interactives permettant une meilleure accessibilité au contenu pédagogique et une plus grande implication des apprenants.

Dans ce contexte, les applications mobiles éducatives de type quiz se distinguent par leur capacité à rendre l'apprentissage plus ludique et dynamique. Elles permettent aux utilisateurs d'évaluer leurs connaissances de manière autonome, d'identifier leurs lacunes et de suivre leur progression à travers des mécanismes simples tels que le score et le feedback immédiat. Cependant, concevoir une application de quiz efficace ne se limite pas à l'affichage de questions et de réponses ; elle nécessite une réflexion approfondie sur l'ergonomie, la navigation, la gestion de l'état de l'application et l'expérience utilisateur globale.

Sur le plan technique, le développement d'applications mobiles pose plusieurs défis, notamment la diversité des plateformes, la performance, la maintenabilité du code et la rapidité de développement. Le choix d'un framework adapté devient alors un élément clé pour répondre à ces contraintes. **Flutter**, grâce à son approche multiplateforme et à son système de widgets, offre une solution moderne permettant de développer des applications performantes à partir d'une base de code unique.

La problématique principale de ce projet académique réside donc dans la conception et le développement d'une application mobile éducative capable d'offrir une expérience utilisateur fluide, intuitive et pédagogique, tout en respectant les bonnes pratiques du développement mobile. Il s'agit de répondre aux questions suivantes : comment structurer efficacement une application Flutter ? comment gérer la navigation et l'état de l'application ? et comment concevoir une interface claire facilitant l'interaction de l'utilisateur avec le quiz ?

Ainsi, ce projet vise à proposer une solution mobile simple et évolutive, permettant de mettre en pratique les concepts théoriques étudiés, tout en répondant aux exigences fonctionnelles et ergonomiques d'une application éducative moderne.

Chapitre 2

Présentation générale du projet

2.1 Contexte académique

Ce projet a été réalisé dans le cadre d'un contrôle académique visant à évaluer les compétences acquises en développement d'applications mobiles multiplateformes. Il s'inscrit dans le programme pédagogique de la formation dispensée à l'**EMSI**, et a pour objectif de mettre en pratique les notions théoriques étudiées en cours, notamment la programmation orientée objet, la conception d'interfaces utilisateur et le développement mobile avec Flutter.

La réalisation de ce projet permet également de renforcer l'autonomie de l'étudiant, sa capacité à analyser un besoin, à proposer une solution technique adaptée et à mettre en œuvre une application fonctionnelle répondant à des exigences précises.

2.2 Cahier des charges

Le cahier des charges définit les fonctionnalités et les contraintes techniques de l'application à développer. L'application mobile éducative de type quiz doit répondre aux exigences suivantes :

— Proposer une série de questions à choix multiples.
— Permettre à l'utilisateur de sélectionner une seule réponse par question.
— Afficher les questions de manière claire et lisible.
— Passer automatiquement à la question suivante après validation de la réponse.
— Calculer et afficher le score final à la fin du quiz.
— Offrir une interface utilisateur simple, intuitive et ergonomique.

Sur le plan technique, l'application doit être développée avec le framework Flutter et le langage Dart, et être compatible avec les plateformes mobiles, notamment Android.

2.3 Public cible

L'application est destinée principalement à un public étudiant ou à toute personne souhaitant tester et renforcer ses connaissances de manière ludique. Elle peut être utilisée dans un cadre éducatif, que ce soit pour l'auto-évaluation, la révision de notions théoriques ou l'apprentissage interactif.

Grâce à sa simplicité d'utilisation et à son interface intuitive, l'application peut également s'adresser à un large public, indépendamment de l'âge ou du niveau de connaissances techniques de l'utilisateur.

2.4 Objectifs fonctionnels

Les objectifs fonctionnels du projet définissent les fonctionnalités principales que l'application doit offrir à l'utilisateur :

- Démarrer un quiz à partir de l'écran principal.
- Afficher successivement les questions et leurs choix de réponses.
- Enregistrer les réponses sélectionnées par l'utilisateur.
- Calculer automatiquement le score en fonction des réponses correctes.
- Afficher un écran de résultat récapitulant le score obtenu.
- Permettre à l'utilisateur de relancer le quiz.

2.5 Objectifs pédagogiques

Au-delà de l'aspect fonctionnel, ce projet vise également plusieurs objectifs pédagogiques :

- Comprendre le fonctionnement du framework Flutter et du langage Dart.
- Appliquer les principes de la programmation orientée objet.
- Apprendre à structurer un projet Flutter de manière claire et maintenable.
- Développer une interface utilisateur respectant les principes de l'ergonomie.
- Renforcer la capacité à analyser un problème et à proposer une solution technique adaptée.

Chapitre 3

Analyse des besoins

3.1 Besoins fonctionnels

L'application mobile de quiz doit permettre les fonctionnalités suivantes :

— **Gestion des questions** : L'application doit pouvoir stocker, afficher et gérer un ensemble de questions avec leurs propositions de réponses associées.

— **Sélection des réponses** : L'utilisateur doit pouvoir sélectionner une réponse pour chaque question présentée.

— **Calcul du score** : À la fin du quiz, l'application doit calculer le score total obtenu par l'utilisateur en fonction des réponses sélectionnées.

— **Affichage des résultats** : Le résultat final (score, bonne/mauvaise réponse, etc.) doit être affiché clairement à l'utilisateur.

3.2 Besoins non fonctionnels

En plus des fonctionnalités, l'application doit répondre aux critères non fonctionnels suivants :

— **Performance** : L'application doit être réactive et fluide, avec des temps de réponse courts pour une bonne expérience utilisateur.

— **Ergonomie** : L'interface doit être intuitive et facile à utiliser, adaptée à un public large et varié.

— **Sécurité de base** : Les données utilisateur doivent être protégées contre toute forme d'accès non autorisé. Bien que ce soit une application simple, les bonnes pratiques de sécurité doivent être respectées.

— **Compatibilité Android / iOS** : L'application doit fonctionner correctement sur les plateformes mobiles Android et iOS, offrant une expérience homogène sur les deux systèmes.

Chapitre 4

Étude technique

4.1 Choix technologiques

Pour le développement de cette application mobile de quiz, les technologies suivantes ont été choisies :

- **Flutter** : Framework open-source développé par Google, permettant de créer des applications multiplateformes (Android, iOS) avec une seule base de code.
- **Dart** : Langage de programmation utilisé par Flutter, offrant une syntaxe moderne et performante, ainsi qu'une compilation native pour mobile.
- **IDE utilisé** : L'environnement de développement intégré utilisé est Visual Studio Code (ou Android Studio, selon ton cas), qui propose de nombreux plugins facilitant le développement Flutter.

4.2 Environnement de développement

Le développement a été réalisé dans l'environnement suivant :

- **Système d'exploitation** : Le système d'exploitation utilisé est Windows 10 (ou précise celui que tu utilises).
- **Émulateur** : Pour les tests, un émulateur Android a été utilisé (ex. : Android Emulator fourni avec Android Studio) ainsi qu'un simulateur iOS (si applicable).
- **Outils** : Flutter SDK, Dart SDK, ainsi que les outils en ligne de commande Flutter pour la gestion du projet et le déploiement.

4.3 Comparaison (optionnelle)

Une comparaison entre Flutter et d'autres technologies populaires de développement mobile a été effectuée :

- **Flutter vs React Native** : Flutter utilise un moteur de rendu graphique propre, ce qui permet une meilleure performance et une apparence uniforme sur toutes les plateformes. React Native repose sur les composants natifs du système, ce qui peut entraîner des différences visuelles et des problèmes de performance dans certains cas. Flutter offre également une compilation native et une meilleure intégration avec les outils Google.
- **Flutter vs Java Android** : Le développement natif Android avec Java offre une grande maîtrise et accès direct aux API Android, mais nécessite de développer une version spécifique pour iOS. Flutter permet un développement multiplateforme avec un seul code source, réduisant ainsi le temps et le coût de développement, tout en maintenant des performances proches du natif.

Chapitre 5

Conception et Implémentation de l'application

5.1 Conception de l'application

5.1.1 Architecture générale

— **Architecture Flutter** : L'application est développée avec Flutter, qui utilise une architecture basée sur des widgets. Chaque écran et élément graphique est un widget qui peut être stateless (sans état) ou stateful (avec état), permettant une gestion dynamique de l'interface.
— **Organisation des dossiers** : Le projet suit une organisation claire des fichiers avec des dossiers dédiés pour :
  — les écrans (screens),
  — les modèles (models),
  — les services (services) ou logique métier,
  — les widgets réutilisables,
  — les ressources (assets) comme images ou fichiers JSON.

5.1.2 Diagrammes UML

— **Diagramme de cas d'utilisation** : Ce diagramme illustre les interactions principales entre l'utilisateur et l'application, notamment la sélection du quiz, la réponse aux questions, et la consultation des résultats.

5.1.3 Conception de l'interface utilisateur

— **Maquettes des écrans** : Des maquettes ont été réalisées pour visualiser l'agencement des éléments : écran d'accueil, écran de quiz, écran de résultats.
— **Navigation entre écrans** : La navigation est gérée par le système de routes de Flutter, permettant de passer d'un écran à l'autre avec animation fluide et gestion du retour.

5.2 Implémentation

5.2.1 Structure du projet

— **Arborescence du projet Flutter** : L'arborescence suit la structure recommandée avec des dossiers tels que : `/lib/screens`, `/lib/models`, `/lib/widgets`, `/assets`, etc.

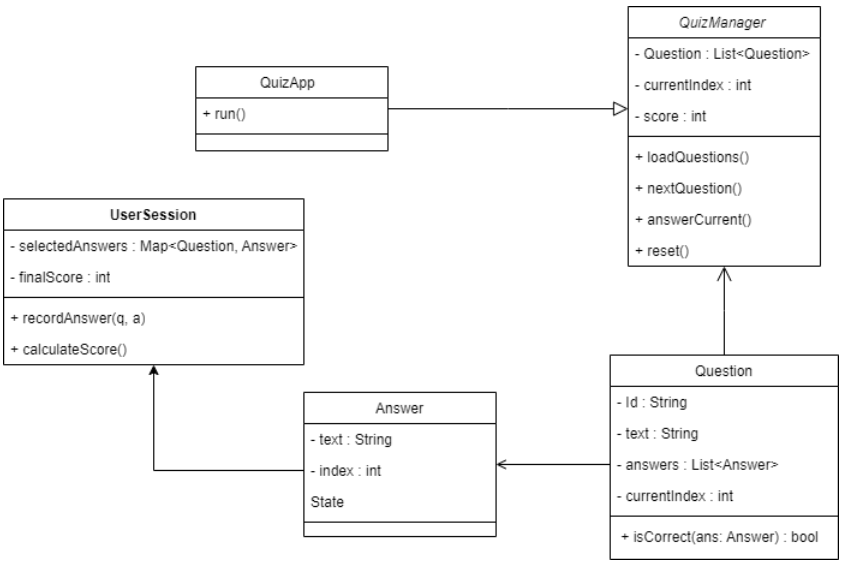

FIGURE 5.1 – Diagramme de Classe

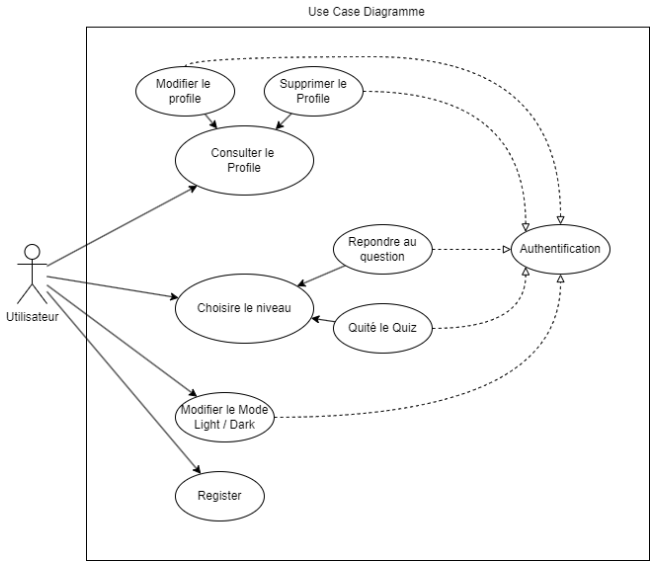

FIGURE 5.2 – Use Case Diagramme

5.2.2 Description des fichiers principaux

— `main.dart` : Point d'entrée de l'application, initialisant Flutter et configurant la navigation entre les écrans.
— Fichiers des écrans : Chaque écran est défini dans un fichier Dart dédié, par exemple $\texttt{quiz}_s creen.dart, \texttt{result}_s creen.dart$.
- `Logique du quiz : La gestion des questions, des réponses et du score est encapsulé dans des classes dédiées et des services.`

5.2.3 Extraits de code

— **Initialisation de l'application** : Exemple du code dans `main.dart` pour lancer l'application et définir la route initiale.
— **Gestion du score** : Code illustrant le calcul et la mise à jour du score en fonction des réponses de l'utilisateur.
— **Navigation** : Utilisation de `Navigator.push` et `Navigator.pop` pour gérer le passage

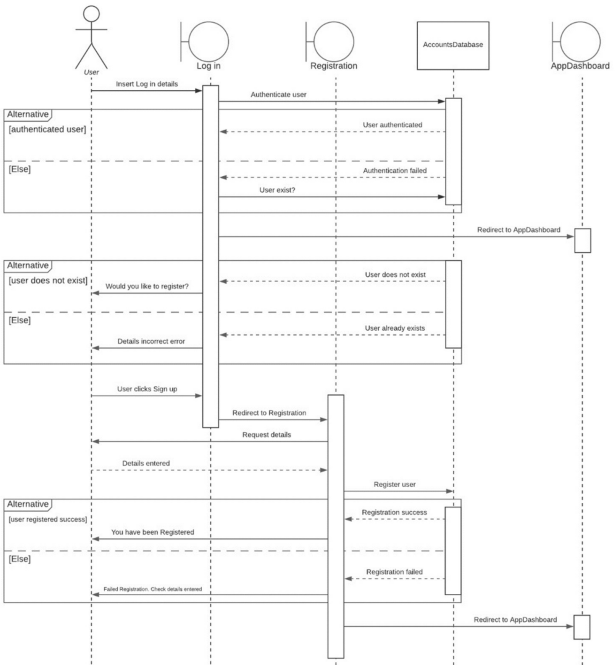

FIGURE 5.3 – Diagramme de Sequane

entre écrans.

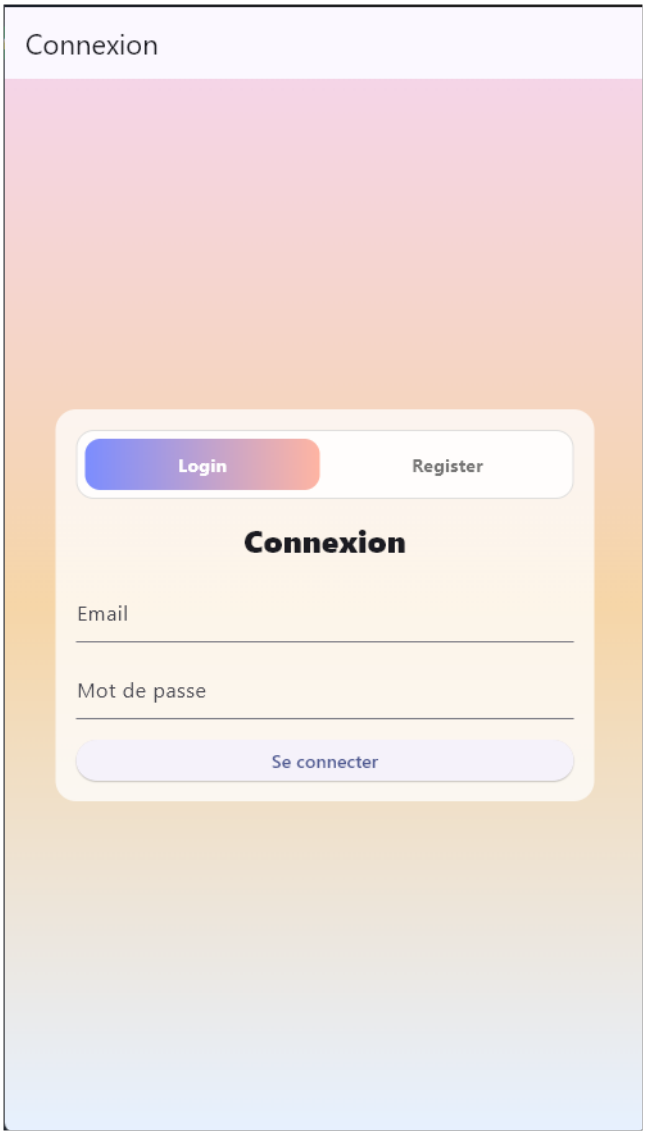

FIGURE 5.4 – Page Login

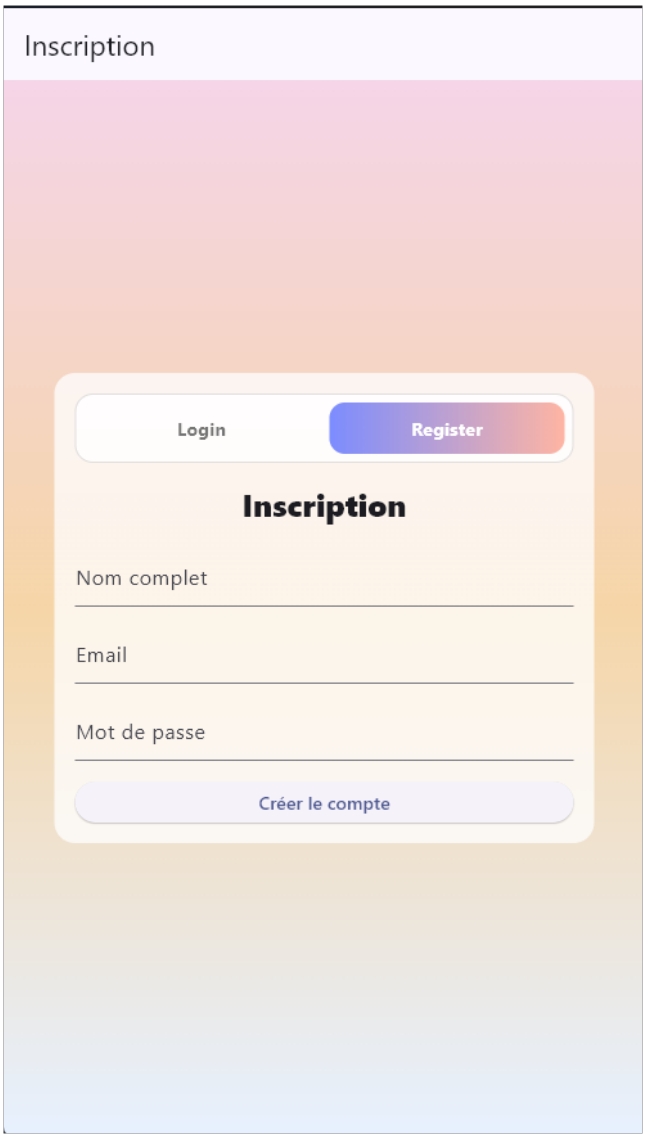

FIGURE 5.5 – Page Register

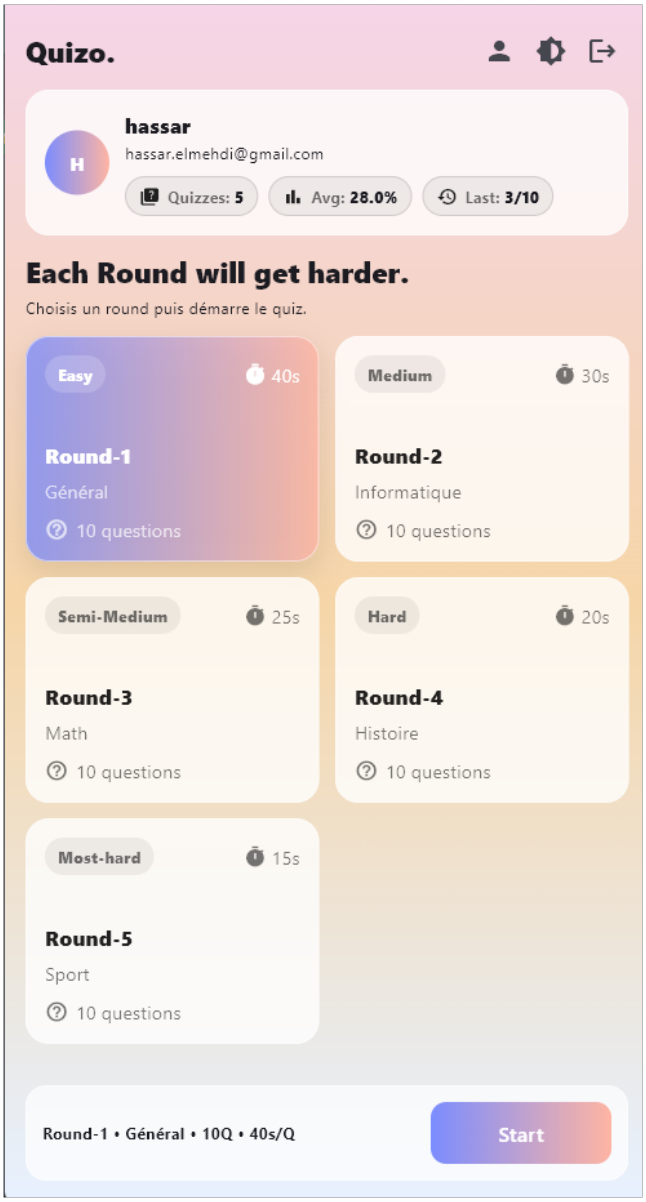

FIGURE 5.6 – Page Front

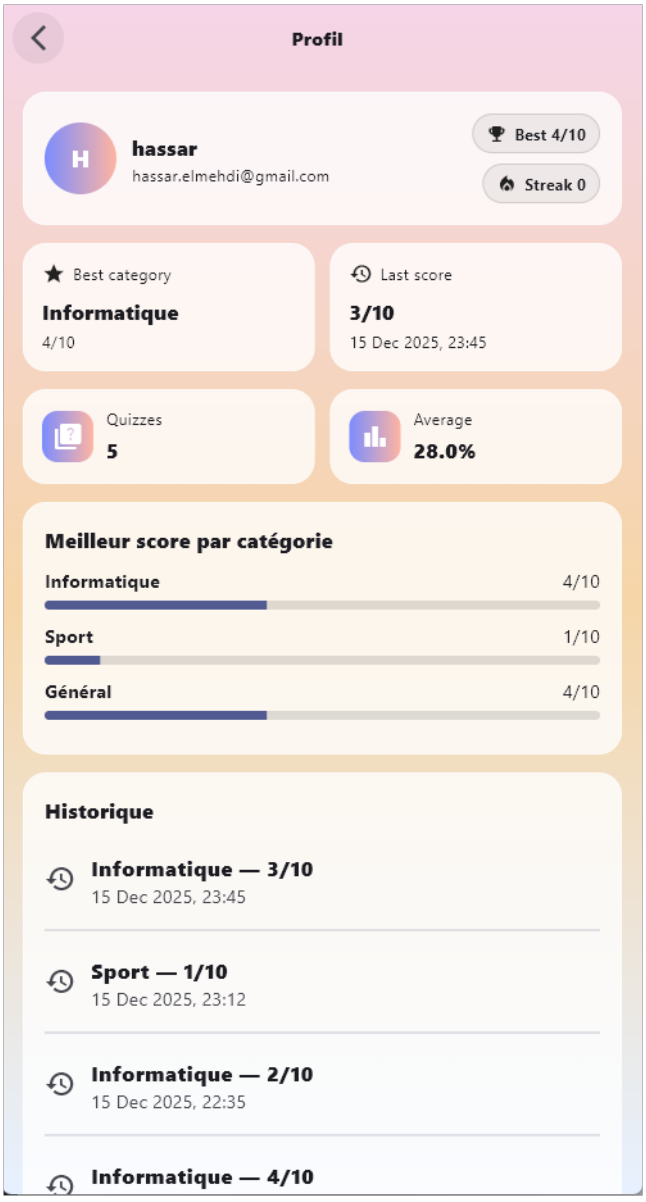

FIGURE 5.7 – Page Profile

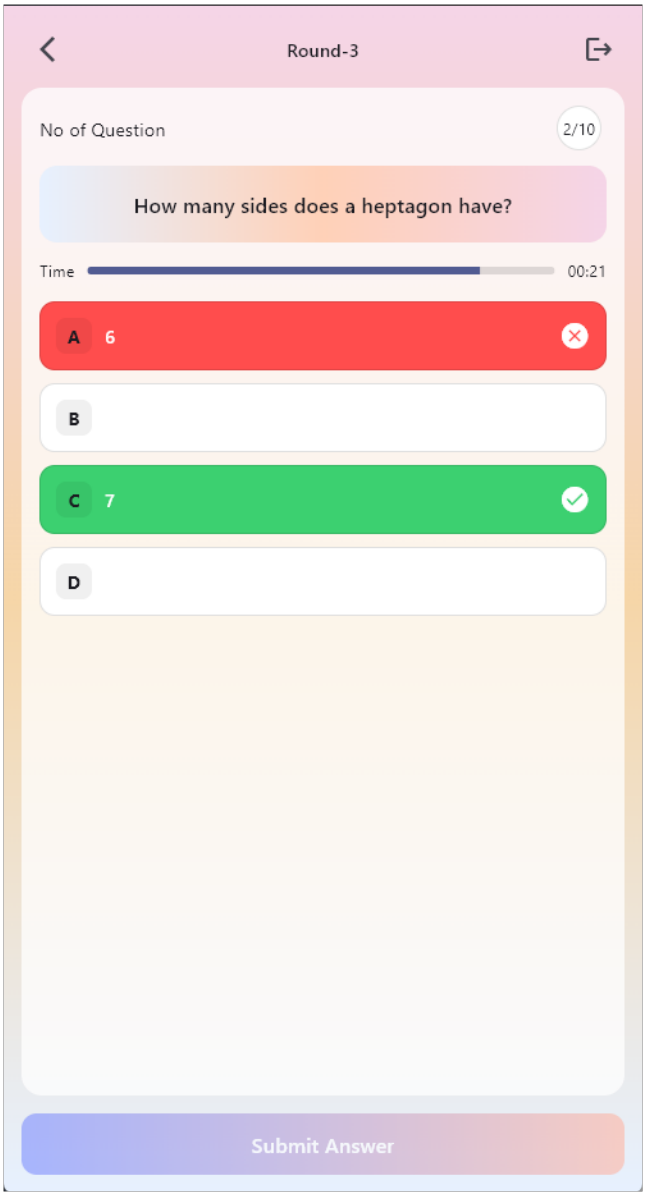

FIGURE 5.8 – Page Quiz

Chapitre 6

Difficultés rencontrées

6.1 Problèmes techniques

Durant le développement de l'application, plusieurs difficultés techniques ont été rencontrées, notamment :

— Gestion de l'état des widgets pour la sélection des réponses.
— Implémentation d'une navigation fluide entre les écrans.
— Optimisation des performances pour éviter les ralentissements sur certains appareils.
— Adaptation de l'interface aux différentes tailles d'écran et plateformes (Android/iOS).

6.2 Solutions apportées

Pour surmonter ces difficultés, les solutions suivantes ont été mises en place :

— Utilisation de widgets `Stateful` pour gérer dynamiquement les états.
— Adoption du système de navigation de Flutter avec des routes nommées.
— Tests réguliers sur plusieurs émulateurs et appareils physiques pour ajuster les performances.
— Mise en place de designs responsives avec des widgets adaptatifs.

6.3 Leçons apprises

Ce projet a permis d'acquérir plusieurs enseignements importants :

— L'importance d'une bonne gestion de l'état dans les applications Flutter.
— La nécessité de tester l'application sur plusieurs environnements pour garantir la compatibilité.
— La valeur d'une architecture claire et bien organisée pour faciliter le développement et la maintenance.

Chapitre 7

Conclusion et perspectives

7.1 Bilan général du projet

Ce projet de développement d'une application mobile de quiz en Flutter a permis de mettre en pratique les concepts de programmation mobile multiplateforme. L'application répond aux besoins fonctionnels attendus et offre une expérience utilisateur satisfaisante sur Android et iOS.

7.2 Compétences acquises

Au cours de ce projet, plusieurs compétences ont été développées, notamment :
— Maîtrise du framework Flutter et du langage Dart.
— Conception et organisation d'un projet mobile structuré.
— Gestion de la navigation et de l'interface utilisateur responsive.
— Résolution de problèmes techniques liés à la gestion de l'état et à la compatibilité multiplateforme.

7.3 Améliorations futures

Plusieurs pistes d'amélioration sont envisagées pour enrichir l'application :
— **Base de données** : Intégrer une base de données locale ou distante pour stocker les questions et les scores utilisateur.
— **Niveaux** : Ajouter différents niveaux de difficulté pour les quiz afin de diversifier les défis.
— **Authentification** : Permettre aux utilisateurs de se connecter afin de sauvegarder leurs progrès.
— **Chronomètre** : Implémenter un timer pour limiter le temps de réponse aux questions et rendre le jeu plus dynamique.