Ecole Marocaine Des Sciences De L'Ingenieur (EMSI)

Département Informatique

Muslim App

Développement d'une application mobile complète sous Flutter

Auteur : Amine Içame // Salma Benomar

Professeur : M. Deroussi Anas

Année Universitaire : 2024 - 2025

Table des matières

Introduction

0.1 Contexte du projet

Dans le cadre de notre module de développement mobile, il nous a été demandé de réaliser une application complète intégrant l'authentification et plusieurs interfaces fonctionnelles.

L'objectif était de maîtriser le framework **Flutter** ainsi que l'intégration d'une base de données Cloud (**Firebase**).

0.2 Choix du sujet

J'ai choisi de développer une application utilitaire pour les musulmans, nommée "Muslim App". Ce choix est motivé par la complexité technique intéressante qu'il propose :

— Utilisation de la géolocalisation (GPS) pour les prières.
— Calculs astronomiques (Package Adhan).
— Gestion des capteurs (Boussole/Qibla).
— Base de données temps réel (Adhkar et Conseils).

Chapitre 1

Outils et Technologies

1.1 Flutter & Dart

L'application est développée avec **Flutter**, le kit de développement logiciel (SDK) d'interface utilisateur créé par Google. Il permet de créer des applications compilées nativement pour mobile, web et bureau à partir d'une seule base de code en langage **Dart**.

1.2 Firebase

Pour le Backend, nous avons utilisé la suite Firebase de Google :

— **Firebase Auth :** Pour la gestion sécurisée des utilisateurs (Inscription/Connexion par email).

— **Cloud Firestore :** Une base de données NoSQL hébergée dans le cloud pour stocker les Adhkar et les Conseils de manière dynamique.

1.3 Packages Clés

— `adhan` : Pour le calcul des horaires de prière.

— `geolocator` : Pour récupérer la position GPS.

— `flutter_compass` : Pour la fonctionnalité Qibla.

— `flutter_animate` : Pour les animations fluides.

Chapitre 2

Fonctionnalités et Réalisation

2.1 Authentification (Login/Register)

L'application est sécurisée. Au démarrage, une vérification est faite. Si l'utilisateur n'est pas connecté, il est redirigé vers une interface permettant de saisir son email et son mot de passe ou de créer un compte.

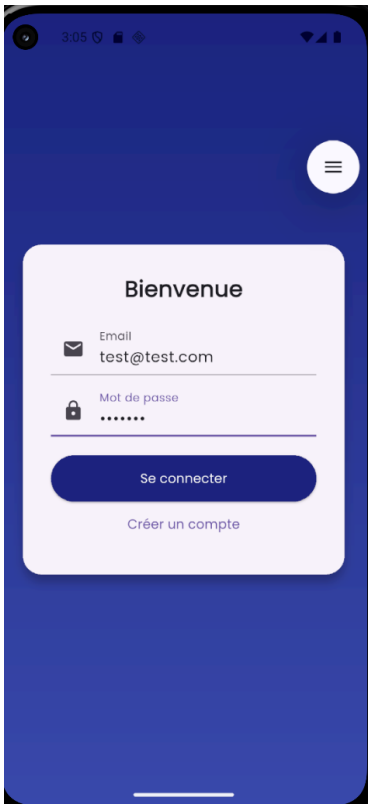

FIGURE 2.1 – Écran de Connexion

2.2 Horaires de Prière (Accueil)

C'est la fonctionnalité principale. L'application :

1. Détecte la position GPS de l'utilisateur.

2. En cas d'échec (ex : émulateur), bascule automatiquement sur les coordonnées de Casablanca.
3. Calcule les heures selon la méthode du Ministère des Habous (Maroc) : Fajr à 19°, Isha à 17°.

Le design utilise un style moderne avec un dégradé sombre et des informations claires sur la prochaine prière.

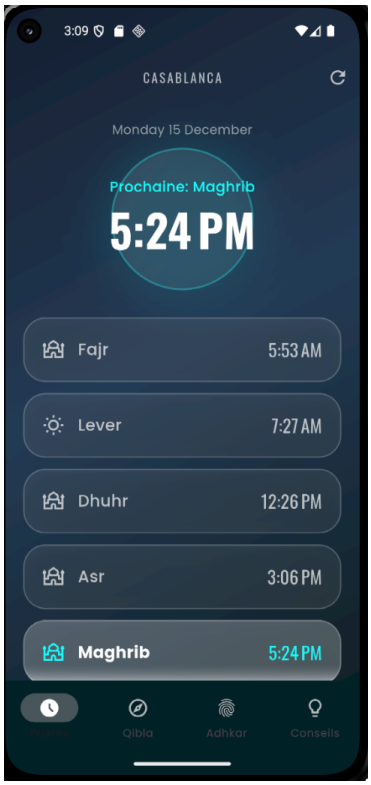

FIGURE 2.2 – Horaires de Prière

2.3 Direction de la Qibla

Une interface dédiée affiche une boussole qui indique la direction de la Mecque (Kaaba) en temps réel grâce aux capteurs magnétiques du téléphone.

2.4 Adhkar et Conseils (Dynamique)

Ces deux pages ne sont pas statiques. Elles sont connectées à **Cloud Firestore**.

— **Adhkar :** Une liste de compteurs (Tasbih) interactifs. L'utilisateur clique pour incrémenter.

— **Conseils :** Une liste de rappels religieux.

L'avantage est que l'administrateur peut ajouter de nouveaux contenus directement depuis la console Firebase sans mettre à jour l'application.

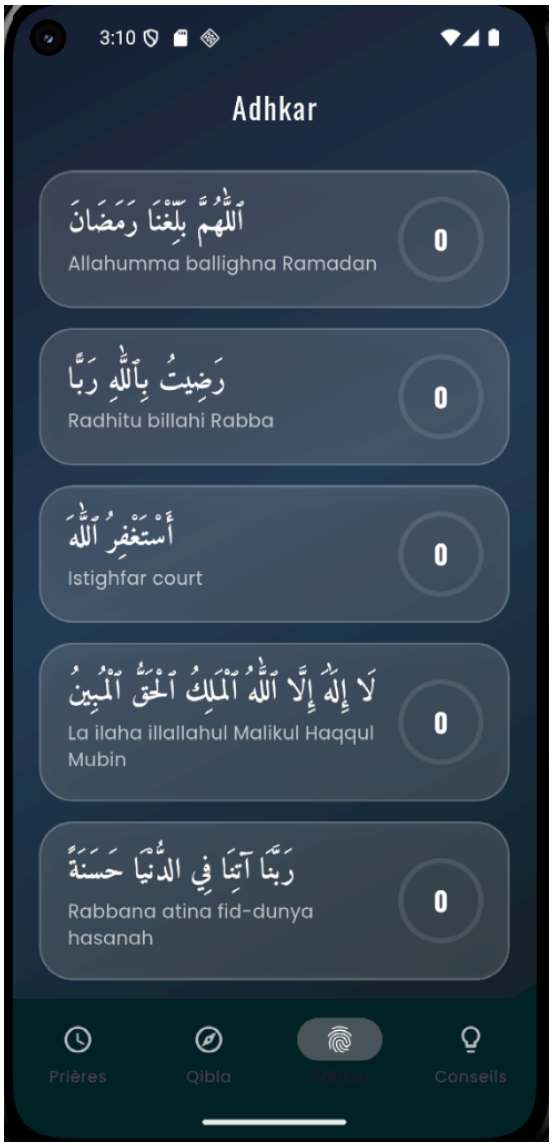

FIGURE 2.3 – Adhkar page

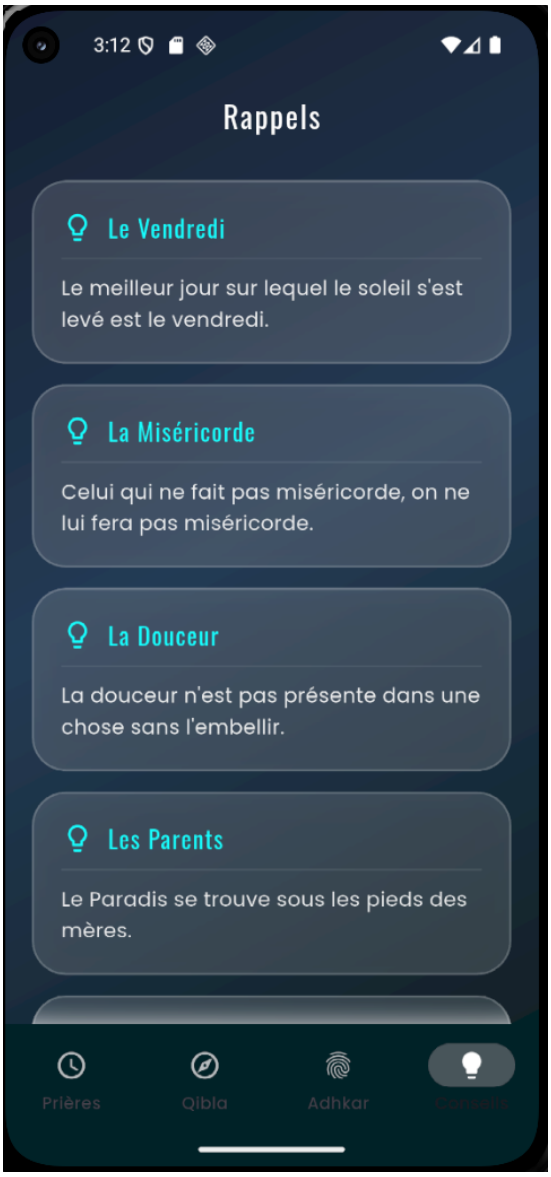

FIGURE 2.4 – Conseils page

Chapitre 3

Design et Implémentation

3.1 Architecture du Code

Le projet suit une structure propre et modulaire :
— `lib/main.dart` : Point d'entrée et logique de redirection (Auth).
— `lib/screens/` : Contient toutes les pages (Login, Home, Qibla, Adhkar...).
— `lib/widgets/` : Contient les composants réutilisables comme `GlassCard`.

3.2 Design "Glassmorphism"

Pour offrir une expérience utilisateur (UX) moderne, j'ai implémenté un design basé sur la transparence et le flou (Glassmorphism).

Voici un extrait du widget `GlassCard` utilisé pour les listes :

```dart
class GlassCard extends StatelessWidget {
  final Widget child;
  // ...
  @override
  Widget build(BuildContext context) {
    return ClipRRect(
      borderRadius: BorderRadius.circular(25),
      child: BackdropFilter(
        filter: ImageFilter.blur(sigmaX: 10, sigmaY: 10),
        child: Container(
          color: Colors.white.withOpacity(0.15), // Transparence
          child: child,
        ),
      ),
    );
  }
}
```

Listing 3.1 – Widget GlassCard

3.3 Gestion des Erreurs GPS

Un défi majeur a été la gestion du GPS sur l'émulateur. J'ai implémenté une logique de "Timeout" : si le GPS ne répond pas en 3 secondes, l'application charge les données par défaut pour ne pas planter.

Conclusion

Ce projet a été une excellente opportunité pour approfondir mes connaissances en développement mobile. J'ai pu :

— Mettre en place une authentification complète.
— Manipuler des capteurs matériels (GPS, Boussole).
— Connecter une application à une base de données Cloud en temps réel.
— Soigner l'interface utilisateur avec des animations et un design moderne.

L'application est fonctionnelle, robuste et évolutive grâce à Firebase. Pour les améliorations futures, nous pourrions ajouter les notifications locales pour l'Adhan (appel à la prière).