Documentation Technique

Application de WecodeForChat

Version : 1.0.0
Date : 3 mai 2025
Auteur: Kra christian leonce yao

Sommaire

1. Introduction

L'application de chat mobile est une plateforme de messagerie instantanée développée avec Flutter et Firebase. Elle offre des fonctionnalités de messagerie en temps réel, de partage de médias, de chat de groupe dans une interface utilisateur intuitive et réactive.

1.1 Objectifs du projet

- Offrir une expérience de messagerie instantanée fluide et fiable
- Permettre la communication individuelle et de groupe
- Assurer la sécurité et la confidentialité des conversations
- Optimiser les performances sur diverses qualités de connexion réseau

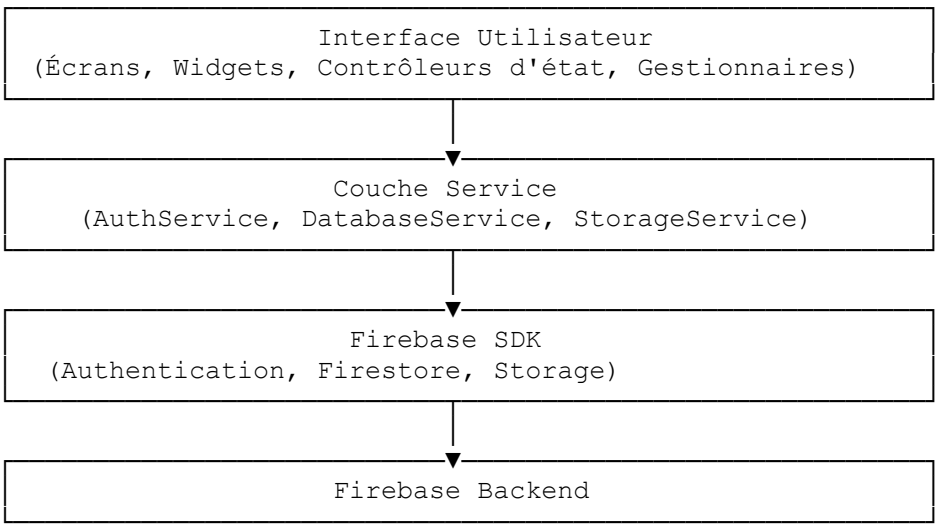
1.2 Technologies utilisées

Technologie	Version	Utilisation
Flutter	3.10+	Framework de développement multiplateforme
Dart	3.0+	Langage de programmation
Firebase Authentication	10.0+	Authentification utilisateur
Cloud Firestore	10.0+	Base de données NoSQL en temps réel
Firebase Storage	10.0+	Stockage de fichiers médias
Google Sign-In	6.0+	Authentification via Google

2. Architecture technique

L'application suit une architecture en couches avec séparation des préoccupations et utilise le pattern Service pour l'accès aux API externes.

2.1 Diagramme d'architecture



2.2 Principes architecturaux

- **Séparation des préoccupations** : Chaque service a une responsabilité unique
- **Programmation réactive** : Utilisation de streams pour les données en temps réel
- **Gestion d'état** : Approche cohérente pour la gestion de l'état de l'application
- **Isolation** : Les changements dans une couche impactent minimalement les autres couches

3. Services

3.1 Service d'authentification

`AuthService` gère tout ce qui concerne l'authentification des utilisateurs, y compris l'inscription, la connexion, la récupération de mot de passe et l'intégration avec les fournisseurs d'authentification tiers.

3.1.1 Fonctionnalités

- Authentification par email/mot de passe
- Authentification via Google
- Gestion de session utilisateur
- Réinitialisation de mot de passe
- Suivi de statut en ligne/hors ligne

3.1.2 Méthodes principales

Méthode	Description	Paramètres	Retour
signUpWithEmailAndPassword	Crée un nouveau compte utilisateur	`email`, `password`, `displayName`	`UserCredential`
signInWithEmailAndPassword	Authentifie un utilisateur existant	`email`, `password`	`UserCredential`
signInWithGoogle	Authentifie via Google	-	`UserCredential`
resetPassword	Envoie un email de réinitialisation	`email`	`void`
signOut	Déconnecte l'utilisateur	-	`void`

3.2 Service de base de données

DatabaseService gère toutes les opérations liées à Firestore, y compris la gestion des utilisateurs, des chats, des messages et des groupes.

3.2.1 Collections Firestore

Collection	Description

users	Profils utilisateurs et statuts
chats	Métadonnées des conversations entre deux utilisateurs
messages	Messages individuels
groups	Informations sur les groupes de discussion
group_messages	Messages dans les groupes
calls	Enregistrements des appels audio/vidéo
reports	Signalements d'utilisateurs

3.2.2 Fonctionnalités

- Gestion des profils utilisateurs
- Messagerie en temps réel
- Création et gestion de conversations individuelles
- Création et gestion de groupes
- Blocage d'utilisateurs
- Recherche de messages et utilisateurs

3.2.3 Méthodes principales

Méthode	Description	Paramètres	Retour
createUserProfile	Crée un profil utilisateur	UserModel user	void
getUserById	Récupère un utilisateur par ID	String uid	Future<UserModel?>
createOrGetChatId	Trouve ou crée une conversation	String userId1, String userId2	Future<String>
sendMessage	Envoie un message	chatId, senderId, receiverId, content, type	Future<MessageModel>
messagesStream	Stream de messages	String chatId	Stream<List<MessageModel>>
createGroup	Crée un groupe	name, creatorId, memberIds, description, photoUrl	Future<String>
blockUser	Bloque un utilisateur	currentUserId, blockedUserId	void
markMessagesAsRead	Marque les messages comme lus	chatId, currentUserId	void

3.3 Service de stockage

StorageService gère le téléchargement et la récupération des fichiers médias dans Firebase Storage.

3.3.1 Fonctionnalités

- Téléchargement d'images de profil

- Téléchargement d'images dans les conversations
- Téléchargement de fichiers audio
- Suppression de fichiers

3.3.3 Méthodes principales

Méthode	Description	Paramètres	Retour
uploadProfileImage	Télécharge une image de profil	userId, imageFile	Future<String> (URL)
uploadChatImage	Télécharge une image pour le chat	chatId, imageFile	Future<String> (URL)
uploadChatAudio	Télécharge un fichier audio	name, file	Future<String> (URL)
deleteFile	Supprime un fichier	fileUrl	Future<void>

4. Modèles de données

4.1 UserModel

Représente un utilisateur de l'application.

```
class UserModel {
  final String uid;
  final String email;
  final String displayName;
  final String photoUrl;
  final String status;
  final DateTime lastSeen;
  final bool isOnline;
  final List<String> blockedUsers;

}
```

4.2 MessageModel

Représente un message individuel.

```
enum MessageType { text, image, audio, video, file }

class MessageModel {
  final String id;
  final String senderId;
  final String receiverId;
  final String content;
  final MessageType type;
  final DateTime timestamp;
  final bool isRead;

}
```

4.3 ChatModel

Représente une conversation entre deux utilisateurs.

```
class ChatModel {
  final String id;
  final List<String> participants;
  final DateTime createdAt;
  final DateTime updatedAt;
  final List<String> mutedBy;
  final List<String> blockedBy;
  final MessageModel lastMessage;
}
```

5. Fonctionnalités principales

5.1 Système de messagerie

L'application offre un système de messagerie robuste avec:

- **Messages en temps réel** : Synchronisation instantanée sur tous les appareils
- **Types de messages** : Texte, images, audio, vidéo
- **Accusés de lecture** : Indicateurs visuels pour les messages lus
- **Indicateurs de frappe** : Notification quand un utilisateur est en train d'écrire
- **Pagination** : Chargement optimisé des messages pour les longues conversations
- **Recherche** : Possibilité de rechercher dans l'historique des messages

5.2 Gestion des utilisateurs

- **Profils utilisateurs** : Photos, statuts, informations personnelles
- **Statut en ligne** : Indicateurs de présence en temps réel
- **Blocage** : Possibilité de bloquer des utilisateurs indésirables

5.3 Groupes de discussion

- **Création de groupes** : Avec nom, description et photo
- **Gestion des membres** : Ajout et suppression de participants
- **Administration** : Rôles d'administrateur avec privilèges spécifiques
- **Messagerie de groupe** : Support pour les conversations à participants multiples

6. Flux utilisateur

6.1 Inscription et connexion

1. L'utilisateur ouvre l'application
2. Choisit une méthode d'authentification (email, Google)
3. Complète les informations de profil nécessaires
4. Accède à l'écran principal de l'application

6.2 Création d'une conversation

1. L'utilisateur accède à la liste des contacts
2. Sélectionne un contact pour démarrer une conversation
3. Le système vérifie l'existence d'une conversation ou en crée une nouvelle
4. L'utilisateur est redirigé vers l'écran de chat

6.3 Envoi de messages

1. L'utilisateur entre du texte ou sélectionne un média à envoyer
2. Appuie sur le bouton d'envoi
3. Le message est envoyé au serveur et stocké dans Firestore
4. Le destinataire reçoit une notification et le message en temps réel

6.4 Création de groupe

1. L'utilisateur accède à la section groupes
2. Sélectionne "Créer un groupe"
3. Entre le nom, la description et sélectionne une photo
4. Ajoute des participants au groupe
5. Confirme la création du groupe

7. Sécurité

7.1 Authentification

- Utilisation des mécanismes sécurisés de Firebase Authentication
- Support de l'authentification multifacteur
- Gestion sécurisée des tokens et de la session

7.2 Règles de sécurité Firestore

Des règles de sécurité sont mises en place pour garantir que:

- Les utilisateurs ne peuvent accéder qu'à leurs propres données
- Les messages ne sont accessibles qu'aux participants de la conversation
- Les données de groupe ne sont modifiables que par les administrateurs

7.3 Protection des données sensibles

- Aucun stockage de mot de passe en clair
- Informations de contact protégées
- Mécanismes de blocage pour prévenir le harcèlement

8. Performance

8.1 Optimisations

- **Pagination** : Limite le nombre de messages chargés à la fois
- **Opérations par lots** : Utilisation de batches pour les opérations multiples

- **Mécanisme de retry** : Gestion intelligente des échecs de requêtes réseau
- **Indexation** : Création d'index Firebase pour optimiser les requêtes fréquentes

8.2 Gestion du cache

- Mise en cache locale des conversations récentes
- Stratégies de chargement différé pour les médias
- Synchronisation intelligente des données lors de la reconnexion

Documentation Flutter

Documentation Firebase

Règles de sécurité Firestore
