

Rapport de Projet – ChatApp Flutter

Date: 15/10/2025 · Plateformes: Web, Android, iOS

Dépendances clés: Firebase (Auth, Firestore, Storage), Provider, emoji_picker_flutter, grouped_list, intl

[Material 3](#) [PWA-ready](#) [Dark/Light](#)

Sommaire

1. Introduction & objectifs
2. Fonctionnalités
3. Architecture & conception
4. UI/UX & Thème
5. Points d'implémentation clés
6. Configuration & déploiement
7. Validation & qualité
8. Retours & améliorations
9. Roadmap
10. Annexes (extraits)

1. Introduction & objectifs

Le projet **ChatApp Flutter** est une application de messagerie temps réel, multi-plateformes, construite avec Flutter (Material 3) et Firebase. Elle vise à offrir une expérience moderne et fluide avec des fonctionnalités essentielles: authentification, discussions, envoi d'images, suppression de messages, regroupement par date, émojis, profils, et pagination.

2. Fonctionnalités

Authentification

- Email/Mot de passe
- Connexion avec Google (web via popup)

Messagerie

- Texte et **images** (Firebase Storage)
- Suppression: pour moi (local), pour tous (global/tombstone)
- Groupement par date: Aujourd'hui/Hier/Date FR
- Heure d'envoi sous chaque message (HH:mm)
- Lazy loading des anciens messages
- Émojis via `emoji_picker_flutter`

Profils

- Affichage du profil d'autrui depuis la discussion
- Edition du profil utilisateur (nom, bio)

Notifications

Le système FCM a été retiré. Le manifeste Web n'inclut plus `gcm_sender_id` et le `firebase-messaging-sw.js` a été supprimé.

3. Architecture & conception

L'application adopte une approche proche de MVVM:

- **Models:** Message, ChatUser
- **ViewModels:** AuthViewModel, ChatViewModel (Provider)
- **Pages:** Login, Signup, Home, Chat, Profile
- **Widgets:** MessageItem, ChatListItem, LoadingScreen
- **Constants:** AppColors/Spacing/Text, FirestorePaths

Stockage des messages dans Firestore sous `chats/{chatId}/messages`, où `chatId` est la concaténation ordonnée des deux `uid`. Les images sont stockées sous `storage: chats/{chatId}/`.

4. UI/UX & Thème

- Matériel 3 avec **blanc** en primaire et **bleu Flutter** en secondaire.
- Pages Auth modernisées (fond dégradé, carte, toggle mot de passe).
- **Home** revisité: barre de recherche, cartes de discussion, avatar et menu.
- **Chat**: GroupedListView (ASC) avec chip date au-dessus des messages; auto-scroll après envoi.
- Mode sombre supporté.

5. Points d'implémentation clés

- **Upload image**: `FirebaseStorage.putData` avec *contentType* et nom unique.
- **Suppression**:
 - Pour moi: ajout de l'utilisateur à `deletedFor`.
 - Pour tous: *tombstone* (`isDeleted`, contenu vide) + suppression du média.
- **Groupeement par date**: `grouped_list` avec labels FR (Aujourd'hui/Hier).
- **Heure HH:mm** sous chaque message via `intl`.
- **FCM**: suppression complète (dépendance, init, tokens, SW, manifeste).

6. Configuration & déploiement

Pré-requis

- Flutter 3.x
- Projet Firebase configuré: Auth (Email/Password + Google si souhaité), Firestore, Storage.

Initialisation

```
flutter pub get
flutterfire configure
flutter run -d chrome
```

Conseil web: après modification du manifeste, relancer dans une fenêtre privée pour éviter le cache.

Règles (développement)

Firestore

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /chats/{chatId}/messages/{msgId} {
      allow read, write: if request.auth != null;
    }
    match /users/{uid} {
      allow read: if request.auth != null;
      allow write: if request.auth != null && request.auth.uid == uid;
    }
  }
}
```

Storage

```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write:
    }
  }
}
```

7. Validation & qualité

- Analyse statique: `flutter analyze` sans erreurs.
- Contrôles visuels: Auth, Home, Chat (envoi de texte et images, suppression, émojis, regroupement et heures).
- Résolution de warnings (API Material3, context async, etc.).

8. Annexes (extraits)

Fichier	Rôle
<code>lib/models/message.dart</code>	Modèle message (texte/image, time, flags)
<code>lib/viewmodel/chat_view_model.dart</code>	Flux messages, envoi texte/image, suppression
<code>lib/widgets/message_item.dart</code>	Rendu bulle + heure HH:mm sous le message
<code>lib/pages/chat_page.dart</code>	GroupedListView, chip date, composer, émojis
<code>lib/pages/login_page.dart</code>	UI moderne + Google Sign-In (web)
<code>lib/pages/home_page.dart</code>	Liste utilisateurs, recherche, cartes

Fichier	Rôle
lib/pages/profile_page.dart	Profil et édition