



# RAPPORT DE PROJET

## React Native

### DriftChronicles

Application Mobile de Messagerie Anonyme  
par Bouteilles Virtuelles

Realisée par : Ilyass Bentha

Projet académique  
Année universitaire : 2025-2026

## TABLE DES MATIÈRES

1. Introduction
2. Objectifs du Projet
3. Problématique
4. Solution Proposée
5. Analyse et Conception UML
6. Technologies Utilisées
7. Architecture du Projet
8. Captures d'Écran et Conclusion

# 1. INTRODUCTION

**DriftChronicles** est une application mobile innovante qui réinvente le concept romantique des "bouteilles à la mer" dans l'ère numérique. Cette plateforme de communication anonyme permet aux utilisateurs de partager leurs pensées les plus profondes avec des inconnus à travers le monde, créant ainsi des connexions humaines authentiques sans les contraintes des réseaux sociaux traditionnels.

## Contexte du Projet

Dans un monde où les réseaux sociaux sont omniprésents mais souvent superficiels, DriftChronicles propose une alternative rafraîchissante basée sur l'anonymat, la spontanéité et l'authenticité. L'application s'inspire de la tradition séculaire des bouteilles à la mer, où des marins confiaient leurs messages aux vagues de l'océan, espérant qu'un jour quelqu'un les découvre.

## Portée du Projet

Ce projet couvre le développement complet d'une application mobile cross-platform (iOS et Android) incluant :

- Système d'authentification sécurisé
- Création et découverte de messages anonymes (bouteilles)
- Messagerie en temps réel entre utilisateurs
- Interface utilisateur intuitive et poétique
- Backend scalable avec base de données en temps réel

## 2. OBJECTIFS DU PROJET

### Objectif Principal

Créer une application mobile permettant aux utilisateurs de communiquer de manière anonyme et spontanée à travers le concept de "bouteilles virtuelles", favorisant ainsi l'expression libre et les connexions authentiques.

### Objectifs Spécifiques

#### Fonctionnels

- **Permettre l'expression anonyme :** Les utilisateurs peuvent partager leurs pensées sans révéler leur identité.
- **Faciliter les découvertes aléatoires :** Un système de distribution aléatoire garantit que chaque message a une chance égale d'être découvert.
- **Établir des conversations privées :** Après avoir trouvé une bouteille, les utilisateurs peuvent engager une conversation avec son créateur.
- **Garantir l'éphémérité :** Les conversations peuvent être terminées à tout moment via le bouton "Release".

#### Techniques

- Développer une application cross-platform fonctionnant sur iOS et Android.
- Implémenter une architecture scalable et maintenable.
- Assurer des performances optimales avec synchronisation en temps réel.
- Garantir la sécurité des données et la confidentialité des utilisateurs.

## Expérience Utilisateur

- Créer une interface minimaliste inspirée de l'océan.
- Offrir une expérience fluide et intuitive.
- Minimiser la courbe d'apprentissage.
- Privilégier l'accessibilité et la simplicité.

## Vision du Projet

Créer un espace numérique où les utilisateurs peuvent s'exprimer librement, partager leurs pensées les plus profondes, et créer des connexions humaines authentiques sans les contraintes des réseaux sociaux traditionnels.

### 3. PROBLÉMATIQUE

#### Problèmes Identifiés dans les Réseaux Sociaux Actuels

##### Manque d'Authenticité

Les réseaux sociaux modernes encouragent la création de personas artificiels. Les utilisateurs sont constamment sous pression pour montrer une version "parfaite" de leur vie, ce qui conduit à :

- Des interactions superficielles et calculées.
- Une représentation déformée de la réalité.
- Un stress constant lié à l'image projetée.
- Une limitation de l'expression libre par peur du jugement.

##### Surcharge d'Information

Les plateformes actuelles souffrent d'une saturation de contenu :

- Flux infinis et addictifs qui capturent l'attention.
- Algorithmes créant des bulles de filtres et des chambres d'écho.
- Difficulté à établir des connexions significatives dans le bruit ambiant.
- Fatigue des réseaux sociaux ("social media fatigue") de plus en plus répandue.

##### Perte de Spontanéité

L'optimisation constante pour l'engagement a tué la spontanéité :

- Chaque publication est planifiée et optimisée.
- Les interactions deviennent prévisibles et formatées.

- Disparition de la surprise et de la découverte authentique.
- Relations limitées aux connexions existantes.

## Problèmes de Confidentialité et de Sécurité

Les utilisateurs font face à de nombreux risques :

- Collecte massive et exploitation des données personnelles.
- Profilage invasif pour la publicité ciblée.
- Manque de contrôle sur les informations partagées.
- Risques accrus de harcèlement, cyberbullying et usurpation d'identité.

## Besoin Non Satisfait

Il existe un besoin croissant pour une plateforme qui permet :

- L'expression anonyme et libre sans jugement social.
- Des connexions spontanées avec des personnes inconnues.
- Des conversations authentiques sans historique permanent.
- Une expérience simple, poétique et apaisante.
- Un contrôle total sur ses interactions et ses données.

**Question centrale :** Comment créer une plateforme sociale qui favorise l'authenticité et les connexions humaines tout en protégeant la vie privée et en évitant les pièges des réseaux sociaux traditionnels ?

## 4. SOLUTION PROPOSÉE

### Concept de DriftChronicles

DriftChronicles résout ces problématiques en proposant une approche radicalement différente basée sur quatre piliers fondamentaux.

#### Anonymat par Design

- **Aucune identité visible** : Pas de photo de profil, pas de nom réel.
- **Identifiants générés** : Chaque utilisateur reçoit un pseudonyme anonyme ("Stranger #123").
- **Pas de réseau social** : Aucun système de followers, d'amis ou de connexions permanentes.
- **Liberté d'expression** : Les utilisateurs peuvent s'exprimer sans crainte de conséquences sociales.

#### Découverte Aléatoire

- **Distribution équitable** : Chaque bouteille a la même chance d'être découverte.
- **Pas d'algorithme biaisé** : Contrairement aux réseaux sociaux, aucun algorithme ne favorise certains contenus.
- **Expérience de sérendipité** : La surprise de découvrir un message inconnu.
- **Égalité des chances** : Pas de popularité ni de métriques de vanité.

#### Éphémérité et Contrôle

- **Conversations temporaires** : Les chats peuvent être "libérés" à tout moment.

- **Pas d'archive permanente** : Focus sur le moment présent.
- **Bouton "Release"** : Contrôle total pour terminer une conversation indésirable.
- **Réduction de l'anxiété** : Pas de pression liée à un historique numérique permanent.

## Simplicité et Poésie

- **Interface minimaliste** : Design épuré inspiré de l'océan.
- **Métaphore des bouteilles** : Concept intuitif et universellement compréhensible.
- **Expérience apaisante** : Couleurs douces, animations fluides.
- **Pas de métriques** : Aucun like, vue, ou autre mesure de popularité.

## Fonctionnalités Principales

### Lancer une Bouteille (Cast Bottle)

L'utilisateur rédige un message anonyme (max 500 caractères) et le "lance" dans l'océan numérique. La bouteille devient alors disponible pour être découverte par d'autres utilisateurs de manière aléatoire.

### Trouver une Bouteille (Pick Bottle)

L'utilisateur peut rechercher des bouteilles à la dérive. Le système sélectionne aléatoirement une bouteille parmi celles disponibles, en excluant celles déjà vues par l'utilisateur.

### Ouvrir et Décider

Après avoir trouvé une bouteille, l'utilisateur a deux choix :

- **Keep & Reply** : Réclamer la bouteille et entamer une conversation privée avec son créateur.

- **Throw Back** : Rejeter la bouteille qui restera en circulation pour d'autres utilisateurs.

## Messagerie en Temps Réel

Une fois une bouteille réclamée, une conversation privée s'établit entre les deux utilisateurs.

Ils peuvent échanger des messages en temps réel tout en restant complètement anonymes.

## Chronicles (Liste des Conversations)

Un onglet dédié permet de gérer toutes les conversations actives avec :

- Aperçu du dernier message
- Indicateurs de messages non lus
- Recherche dans les conversations
- Timestamps relatifs

## Mécanismes de Protection

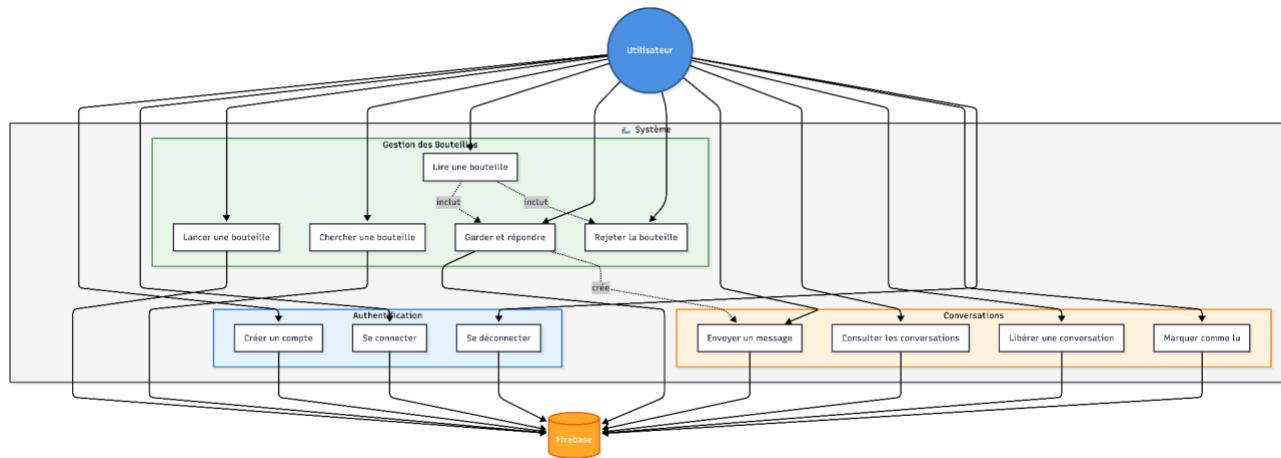
Pour garantir une expérience positive tout en préservant l'anonymat :

- Bouton "Release" : Permet de terminer instantanément toute conversation indésirable.
- Anonymat préservé : Aucune information personnelle n'est jamais révélée.
- Contrôle utilisateur : Chacun décide avec qui il souhaite interagir.
- Pas de spam : Le système de bouteilles uniques limite les abus.
- Modération possible : Architecture prête pour l'ajout de systèmes de signalement.

## 5. ANALYSE ET CONCEPTION UML

### Diagramme de Cas d'Utilisation

Acteur principal : Utilisateur (User)

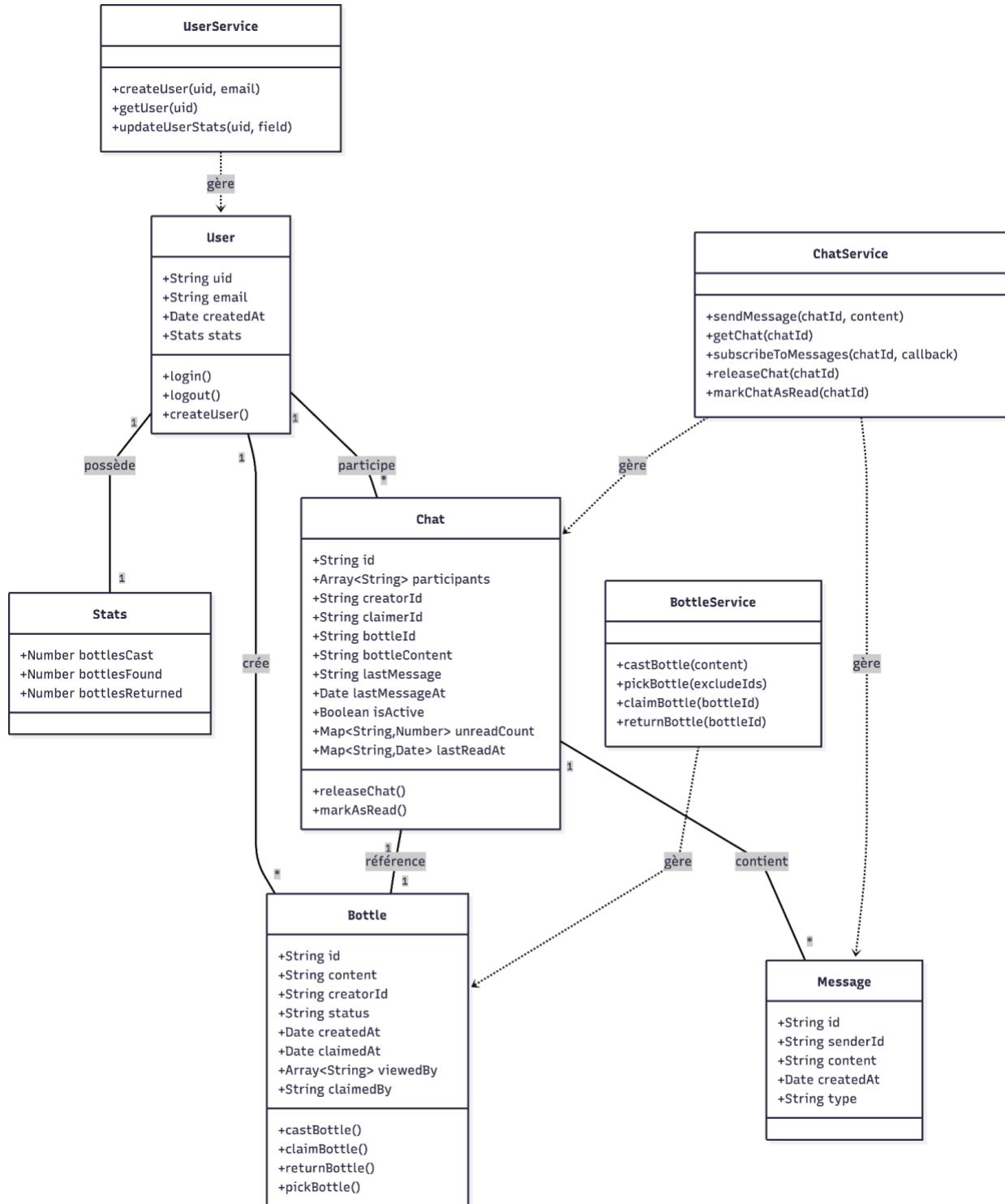


Cas d'utilisation principaux :

- **S'authentifier** : S'inscrire, Se connecter, Se déconnecter.
- **Gérer les Bouteilles** : Lancer une bouteille, Trouver une bouteille, Lire, Réclamer, Rejeter.
- **Gérer les Conversations** : Consulter Chroniques, Ouvrir une conversation, Envoyer/Recevoir messages, Libérer (Release).
- **Consulter le Profil** : Voir stats, historique.

### Diagramme de Classes

Classes Principales :

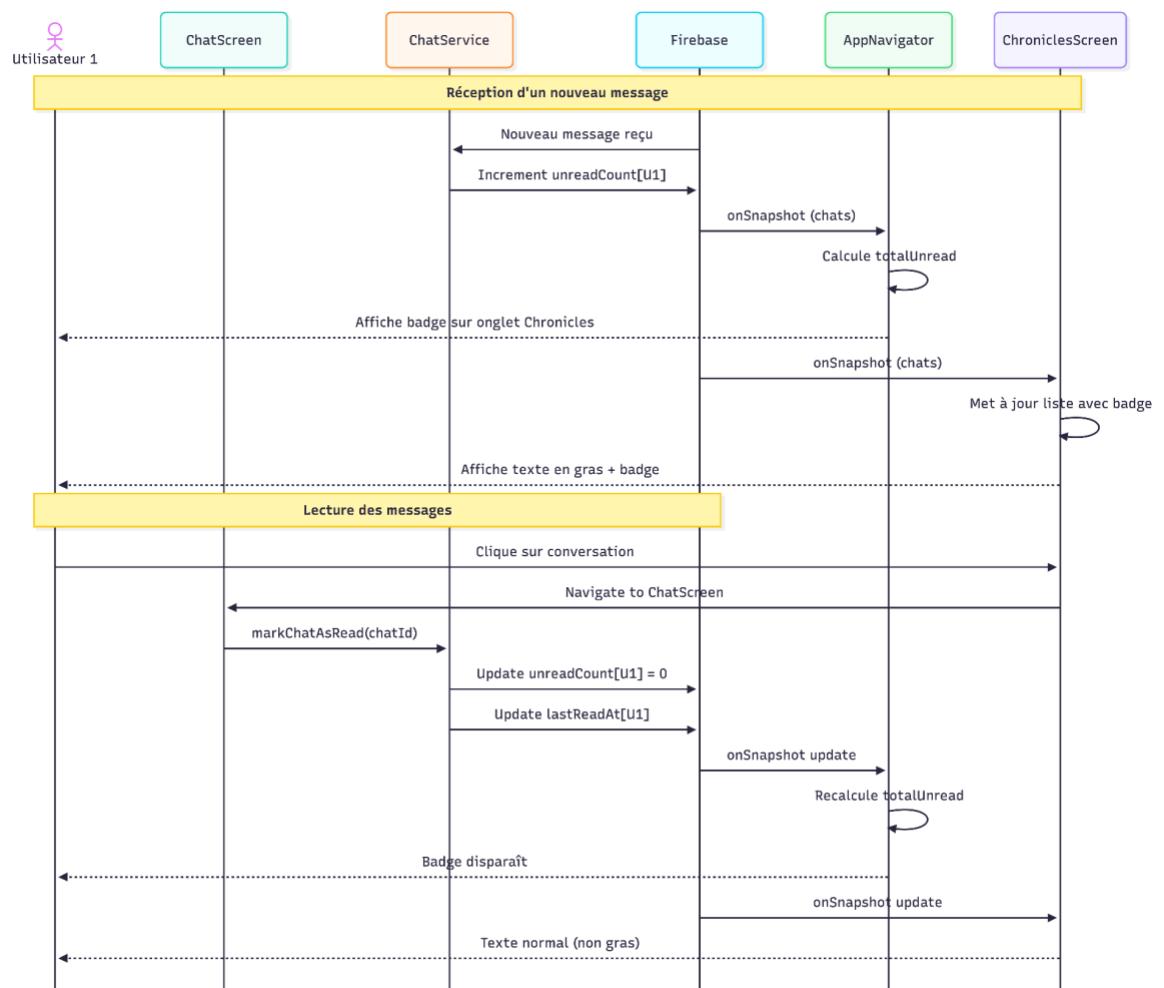


- **User** : uid, email, stats, méthodes(register, login...)
- **Bottle** : id, content, creatorId, status, méthodes(create, claim...)

- **Chat** : id, participants, bottleId, méthodes(addMessage, release...)
- **Message** : id, senderId, content, timestamp

## Diagramme de Séquence

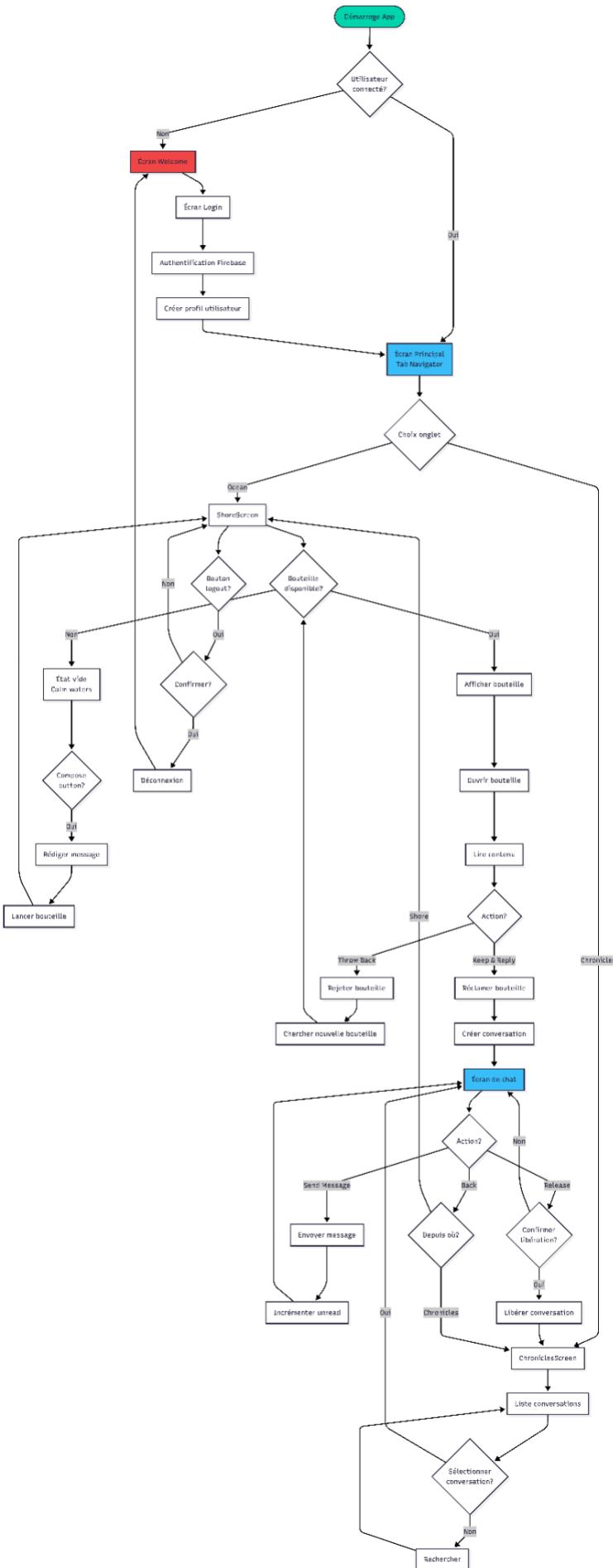
### Scénario 1 et 2 : Lancer et Réclamer une Bouteille & Messagerie en Temps Réel



## Diagramme d'Activité

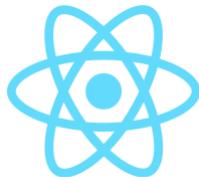
**Activité :** Parcours Complet de Découverte d'une Bouteille (Rechercher, Décision Throw

Back/Keep & Reply, Création chat).



## 6. TECHNOLOGIES UTILISÉES

### Frontend - React Native



React Native

**React Native** : Framework cross-platform permettant de développer des applications mobiles natives pour iOS et Android avec JavaScript et React.

**React** : Bibliothèque UI. Utilisation des Hooks (useState, useEffect, useRef, useCallback).

- Code unique pour deux plateformes.
- Composants natifs pour des performances optimales.
- Hot reload pour un développement rapide.



**Expo SDK 54** : Plateforme de développement simplifiant la création d'applications React Native (Builds simplifiés, Accès APIs natives).

### Backend - Firebase



**Firebase Authentication** : Email/Password.

**Cloud Firestore** : Base de données NoSQL temps réel.

- users/
- bottles/
- chats/
- messages/

Environnement de Développement

## Outils :

### npm (Node Package Manager)



Gestionnaire de paquets pour **JavaScript**.

Il permet d'installer, mettre à jour et gérer les bibliothèques et dépendances d'un projet (ex : React, Expo, etc.).

---

### Expo CLI



Outil en ligne de commande pour **créer, lancer et gérer** des projets **React Native avec Expo**. Il simplifie le développement mobile sans configuration complexe d'Android ou iOS.

---

### Expo Go

Application mobile qui permet de **tester une application Expo directement sur un téléphone**, sans compilation.

Le téléphone scanne un QR code généré par Expo CLI pour afficher l'application.

---

### VS Code (Visual Studio Code)



Éditeur de code léger et puissant.

Il offre la coloration syntaxique, l'autocomplétion, le débogage et des extensions pour faciliter le développement..

## 7. ARCHITECTURE DU PROJET

### Structure des Dossiers

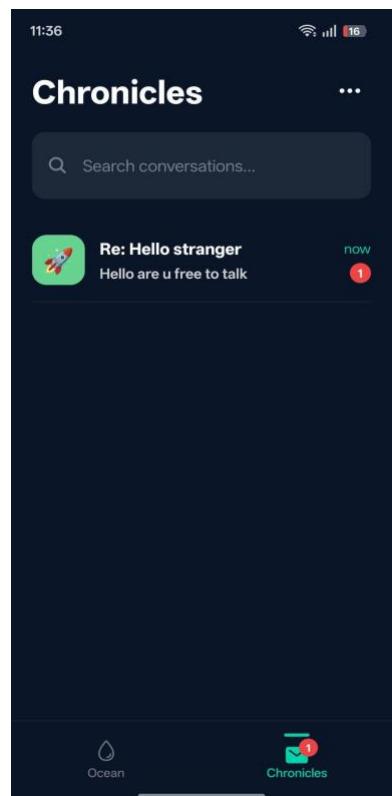
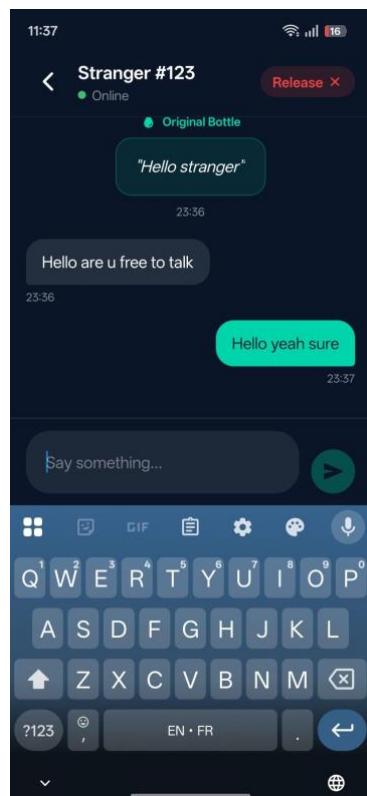
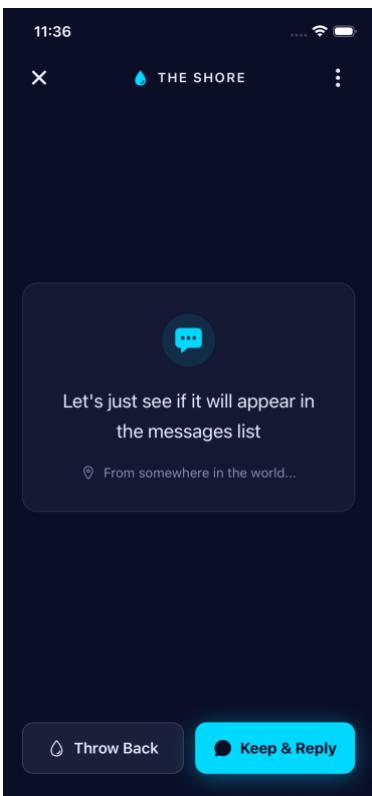
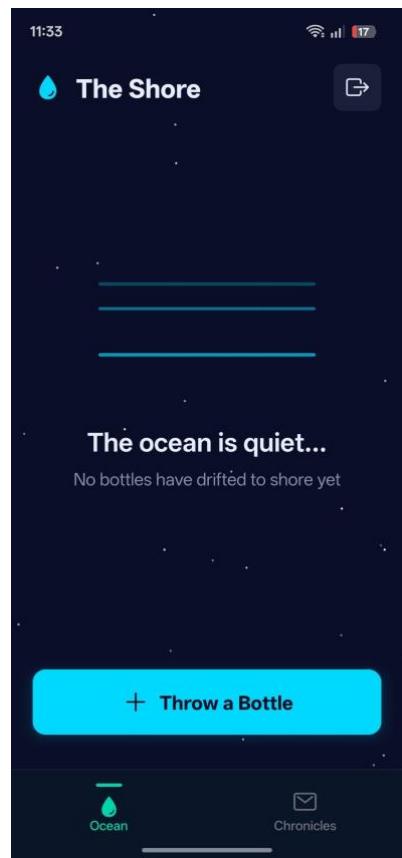
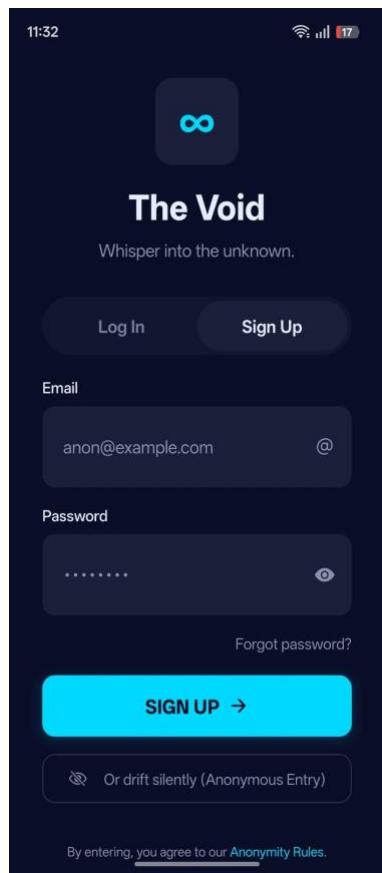
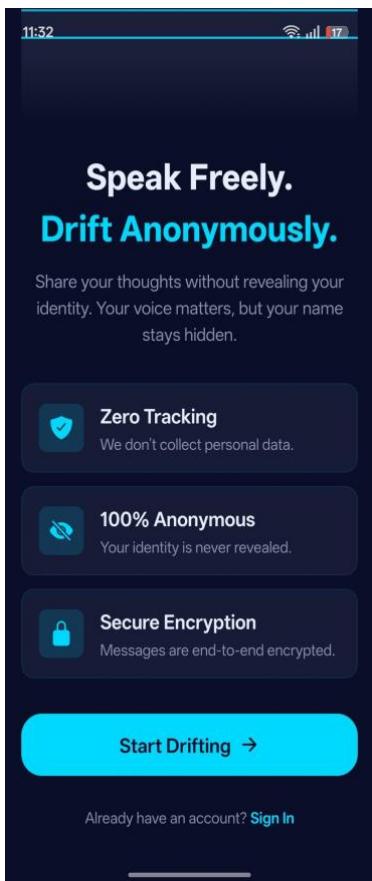
```
drift-chronicles/  
    └── src/  
        ├── components/    # Composants réutilisables  
        ├── screens/       # Ecrans (Shore, Chat, etc.)  
        ├── navigation/   # Configuration navigation  
        ├── services/     # Logique métier & Firebase  
        ├── hooks/        # Custom React hooks  
        ├── utils/        # Fonctions utilitaires  
        └── context/      # Contextes React (Auth)  
    └── assets/          # Images et icônes  
    └── App.js           # Point d'entrée  
    └── firestore.rules  # Règles de sécurité
```

### Pattern de Développement

Séparation stricte des responsabilités :

- **Screens** : Gestion de l'UI et interactions.
- **Services** : Logique métier et appels Firebase.
- **Components** : Composants réutilisables purs.

## 8. CAPTURES D'ÉCRAN ET CONCLUSION



## Conclusion

Le projet DriftChronicles a permis de développer une application mobile fonctionnelle et poétique, répondant à la problématique de la superficialité des réseaux sociaux modernes. Grâce à l'architecture React Native et Firebase, l'application est performante, scalable et sécurisée. Les objectifs d'anonymat, de spontanéité et de simplicité ont été atteints, offrant aux utilisateurs un espace unique pour des connexions humaines authentiques.