

Développement mobile

Projet 2019

Application de chat

V . Garruchet - H . Le Tarnec



Encadré par :

M. Bertrand
Mme. Clermont
M. Pesquet



Sommaire

Sommaire	1
Introduction	2
Spécification générale	3
Analyse fonctionnelle.	3
Analyse des besoins pour l'interface avec l'utilisateur.	3
Répartition des tâches dans le groupe.	4
Spécifications détaillées	5
Liste des fonctionnalités disponibles sur l'application	5
Justification des choix de stockage des informations	6
Résultats et Tests	7
Listes des exigences fonctionnelles respectées par le programme	7
Copies d'écran des pages de l'application.	8
Protocoles de tests pour vérifier le bon fonctionnement du programme.	9
Résultats des tests.	9
Bilan et Perspectives	10
Points positifs et négatifs.	10
Evolutions possibles.	10

I. Introduction

L'objectif de ce projet est la réalisation d'une application mobile de chat (messagerie instantanée) de type Slack ou Discord (en très simplifié). Cette application doit fonctionner sur le principe de channels dans lesquels se trouvent des messages. Afin d'être satisfaisante, l'application doit répondre à diverses exigences fonctionnelles que voici :

Description
L'utilisateur doit pouvoir choisir un pseudo
La liste des channels doit être affichées
L'utilisateur doit pouvoir créer un nouveau channel
L'utilisateur doit pouvoir entrer dans les channels et y trouver la liste des messages avec possibilité de remonter cette liste
L'utilisateur doit pouvoir envoyer des messages
Les listes des messages doivent être mises à jour en temps réel
L'utilisateur doit pouvoir sortir du channel dans lequel il se trouve

Nous sommes également invités à ajouter des nouvelles fonctionnalités telles que l'envoi de GIFs, l'intégration d'avatars, la possibilité de réagir aux messages etc.

De plus, les exigences techniques sont simples : l'environnement technique utilisé est celui du cours. A savoir, l'application passe par l'utilisation de React Native avec Expo, Firebase pour stocker les données et Prettier et ESLint pour le code. Le code réalisé en binôme est, quant à lui, partagé via Github.

Enfin, les livrables sont le code source sur Github et ce rapport.

II. Spécification générale

A. *Analyse fonctionnelle.*

Les exigences métier se divisent en 2 catégories d'importance :

Exigences Prioritaires	Exigences Optionnelles
<ul style="list-style-type: none">• Choix de son pseudo par l'utilisateur.• Affichage de la liste des channels.• Création d'un nouveau channel.• Entrée dans un channel et affichage de la liste des messages avec possibilité de remonter dans la liste.• Envoi de messages texte.• Mise à jour en temps réel des messages envoyés sur le channel.• Sortie d'un channel.	<ul style="list-style-type: none">• Intégration de GIFs via l'API Giphy• Intégration d'avatars via l'API Adorable Avatars ou DiceBear Avatars• Réaction ("like") à un message.• Intégration d'emojis.• ...

B. *Analyse des besoins pour l'interface avec l'utilisateur.*

N'ayant pas accès directement aux utilisateurs, nous nous sommes basés sur les exigences fournies dans le sujet du projet, ainsi que sur notre bon sens pour tenter de fournir un programme répondant au mieux aux besoins des potentiels utilisateurs finaux.

Ainsi, nous avons supposé que les utilisateurs finaux seraient des personnes souhaitant converser de n'importe quel sujet sans restriction d'accès. L'objectif de notre programme est dès lors devenu de leur offrir cette liberté d'interaction via une interface simple, intuitive et peu restrictive. En effet, elle leur permet de choisir un pseudo et d'en changer facilement, de créer autant de channels qu'ils le souhaitent et d'y participer librement.

C. Répartition des tâches dans le groupe.

Afin d'être efficaces sans se contraindre inutilement, nous avons opté pour une répartition des tâches assez libre. Un planning assez général a été mis en place en début de projet pour garantir une progression équilibrée tout au long du projet.

Nous avons premièrement essayé de satisfaire les exigences fonctionnelles principales sans regarder le design. Ainsi, nous avons avant tout d'abord pris en main firebase puis nous avons articulé nos différentes pages dans une navigation. Victor a mis en place ses premières étapes pendant qu'Hugo s'occupait de penser les pages à afficher (Login, Channels et Messages) avant de les designer une fois l'aspect fonctionnel garanti.

Nous avons ainsi pu coordonner notre travail et enchaîner les différentes étapes du développement. Nous nous concertions pour nous assurer que l'interface imaginée par l'un corresponde bien aux fonctions développées par l'autre. Ainsi, nous nous apportions mutuellement conseils, corrections et aides. Grâce à ces bilans et à cette forte communication, nous pouvions nous entraider si nécessaire et éviter ainsi de rester bloqués tout en partageant nos connaissances et méthodes respectives. De plus, cette façon de procéder nous permettait de ne pas travailler sur les mêmes éléments de l'interface simultanément afin d'éviter tout conflit.

Nous avons également une liste de travail à réaliser, de corrections ou d'améliorations à apporter, que nous actualisions en fonction de notre progression pour rester organisés et d'avancer efficacement.

Ce fonctionnement, bien que nécessitant une confiance mutuelle importante, nous a paru être le meilleur choix pour plusieurs raisons. Premièrement, il était moins contraignant qu'un emploi du temps strict. Ainsi, si l'un de nous souhaitait avancer le projet sur son temps libre, il pouvait le faire quand il le souhaitait. Cela nous a permis d'avancer à notre rythme et de rester motivés tout au long du projet.

Deuxièmement, la liste de travail en cours nous servait de support de communication car nous pouvions y inscrire un problème à corriger ou une nouvelle fonctionnalité à implanter. De plus, lister les tâches donne réellement l'impression d'être efficace et d'avancer lorsqu'on en achève une, ce qui est capital pour conserver l'envie de travailler.

En résumé, notre organisation assez souple était tout de même réfléchie et cadrée, ce qui nous a permis de progresser continuellement et d'achever ce premier projet de développement mobile.

III. Spécifications détaillées

A. *Liste des fonctionnalités disponibles sur l'application*

❑ Choisir un pseudo

En arrivant sur l'application, on nous propose d'inscrire un pseudo auquel sera rattaché un avatar. Il n'y a aucunement besoin de s'inscrire, on peut ainsi changer de pseudo dès qu'on le souhaite, cependant, on ne sera plus reconnu comme le même utilisateur. En effet, nous avons choisi de faire correspondre un utilisateur avec un pseudo afin de supprimer l'étape d'inscription. (Cependant, cela pose un petit problème sur lequel nous reviendrons en conclusion).

❑ Afficher les channels existants et les messages qu'ils contiennent

Une fois connecté, la liste des channels existants se présente à nous. En cliquant sur un channel, on peut observer la page de messages qui y sont liés.

❑ Créer un nouveau channel

Dans la page d'accès aux channels, on peut également créer un nouveau channel en lui donnant un nom. Il apparaîtra ensuite dans la liste des channels et sera accessible comme les autres, avec sa propre conversation.

❑ Rechercher un channel

Dans la page d'accès aux channels, on peut aussi rechercher un channel avec son nom. La liste des channels affichés évoluera en fonction de l'entrée dans la barre de recherche.

❑ Envoyer des messages

Lorsqu'on se trouve dans un channel, il est possible d'envoyer des messages. On voit ainsi les messages liés à son pseudo (ses messages si personne n'utilise le même pseudo) en vert sur la gauche. Sur la droite se trouvent en gris les messages de tous les autres utilisateurs. Dans les messages, il est possible de lire le contenu, le pseudo de celui qui l'a envoyé, la date et l'heure de l'envoi et sur le côté, l'avatar lié au pseudo. On peut également ajouter des emojis dans le contenu du texte, en effet ils sont supportés par défaut.

❑ Naviguer entre les différentes pages

Depuis n'importe quelle page, il est possible d'aller à la page qui la précède ou la suit directement dans la hiérarchie de l'application. Ainsi, depuis la page de connexion, on peut accéder à la page des channels, depuis laquelle on peut revenir ou avoir accès à celle des messages d'un channel dont on peut revenir.

B. Justification des choix de stockage des informations



Nous avons choisi deux différents objets de stockage. D'un côté, les channels, caractérisés par un id et un nom (name). Le nom est donné par son créateur et servira d'identifiant dans la liste des channels. De l'autre côté, nous avons les messages qui sont liés aux premiers via leur id (channelId). En effet, un message appartient à un channel, ainsi il paraît simple de les lier ainsi, ce qui nous permet d'afficher les bons messages dans le bon channel. Deuxièmement, on y trouve la date de l'envoi, l'identifiant du message, le texte contenu dans le message et le pseudo de l'utilisateur l'ayant envoyé. Ainsi, il nous est possible d'avoir notre affichage en accédant simplement aux différents paramètres de notre objet message. Le texte contenu dans text, le pseudo de l'utilisateur dans username, la date dans date tout en vérifiant que le message en question est bien dans le channel que nous consultons.

IV. Résultats et Tests

A. Listes des exigences fonctionnelles respectées par le programme

Nous avons fait en sorte de couvrir toutes les exigences métier demandées et avons répondu à quelques exigences optionnelles.

Exigences Prioritaires	Exigences Optionnelles
<ul style="list-style-type: none">✓ Choix de son pseudo par l'utilisateur✓ Affichage de la liste des channels✓ Création d'un nouveau channel✓ Entrée dans un channel et affichage de la liste des messages avec possibilité de remonter dans la liste✓ Envoi de messages texte✓ Mise à jour en temps réel des messages envoyés sur le channel✓ Sortie d'un channel	<ul style="list-style-type: none">✓ Intégration d'avatars via l'API Adorable Avatars ou DiceBear Avatars✓ Intégration d'emojis (supportés initialement)✓ Barre de recherche pour les channels

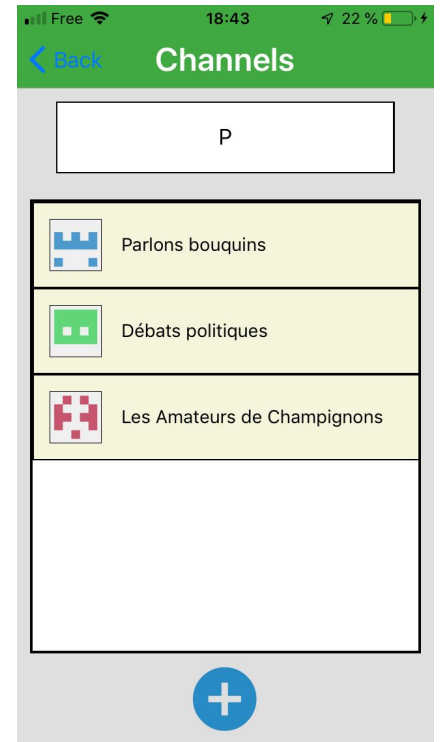
B. Copies d'écran des pages de l'application.



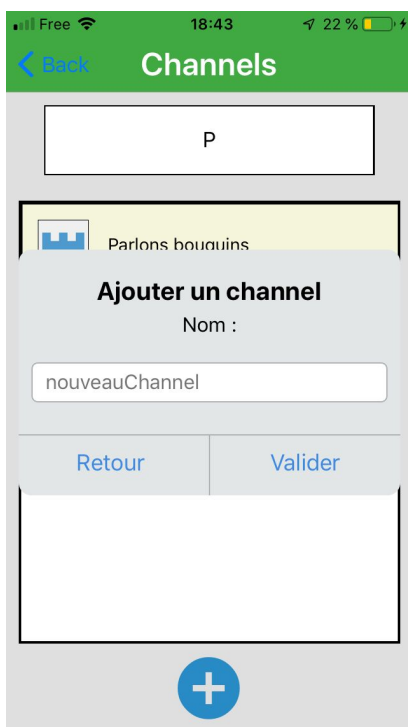
Écran d'accueil



Écran des Channels



Recherche



Ajout Channel



Écran des Messages



Écriture

C. Protocoles de tests pour vérifier le bon fonctionnement du programme.

❑ Test d'interface avec des scenarii d'utilisation

❑ Protocole de test 1

- . Choisissez le pseudo que vous voulez (de préférence autre que FrançoiseDu33)
- . Allez dans le channel nommé "Amis d'enfance"
- . Répondez à la question de Françoise

❑ Protocole de test 2

- . Choisissez le pseudo que vous voulez
- . Créez un channel nommé "FC Jizz"
- . Envoyez un message dans le channel

D. Résultats des tests.

L'application semble intuitive et les personnes testées ont très facilement accompli les scenarii. L'aspect graphique n'a pas engendré de remarques particulières, il semblait même être plutôt apprécié. Ainsi, nos tests semblent montrer que l'application est satisfaisante pour l'utilisation à laquelle elle est destinée (selon nos interprétations du sujet).

V. Bilan et Perspectives

A. *Points positifs et négatifs.*

L'application ChatBox remplit les exigences prioritaires et est très simple et rapide d'utilisation. Ainsi, elle répond à la demande du client que nous avons imaginé. Le design est simple et clair et l'aspect graphique, bien que n'étant pas extraordinaire, demeure correct. La possibilité d'envoyer des emojis, de voir qui a écrit quoi et à quel moment tout en discriminant les messages de l'utilisateur de ceux des autres offre une information assez riche tout en restant lisible. De plus, la possibilité de rechercher un channel par son nom facilite l'accès aux channels qui intéressent l'utilisateur. Pouvoir se connecter, créer et accéder à un channel et envoyer des messages rapidement et facilement offre un confort d'utilisation certain. Enfin, pouvoir changer de pseudo comme on le souhaite permet une réelle souplesse d'utilisation et peut par exemple éviter d'être la cible de harcèlement.

Cependant, cette même souplesse de pseudo peut entraîner quelques problèmes. En effet, la trop grande liberté de choix de pseudo peut occasionner que plusieurs utilisateurs choisissent le même pseudo, volontairement ou non et potentiellement à mauvais escient. En effet, cela offre la possibilité de se faire passer pour autrui, ce qui peut être problématique.

B. *Evolutions possibles.*

La première évolution qui vient à l'esprit est de sécuriser les comptes en ajoutant un mot de passe et en créant réellement un profil. On y perdrait certes l'avantage de rapidité et souplesse actuel, mais la problématique de vol d'identité ou de doublons dans les pseudos est trop importante.

Deuxièmement, on pourrait imaginer rajouter des fonctionnalités telles que les réactions aux messages, l'envoi de photos, gifs, vidéos ou autres comme le proposent des applications telles que Messenger.

Enfin, le design pourrait être un peu amélioré pour avoir un rendu un peu plus esthétique.