

Accord

 Azure Pipelines failed [_build/latest?definitionId=8&branchName=master](https://dev.azure.com/raymsdev/Accord/_build/latest?definitionId=8&branchName=master)

Application de chat !



Le dossier de rendu contient une copy de notre repo git : accord.git

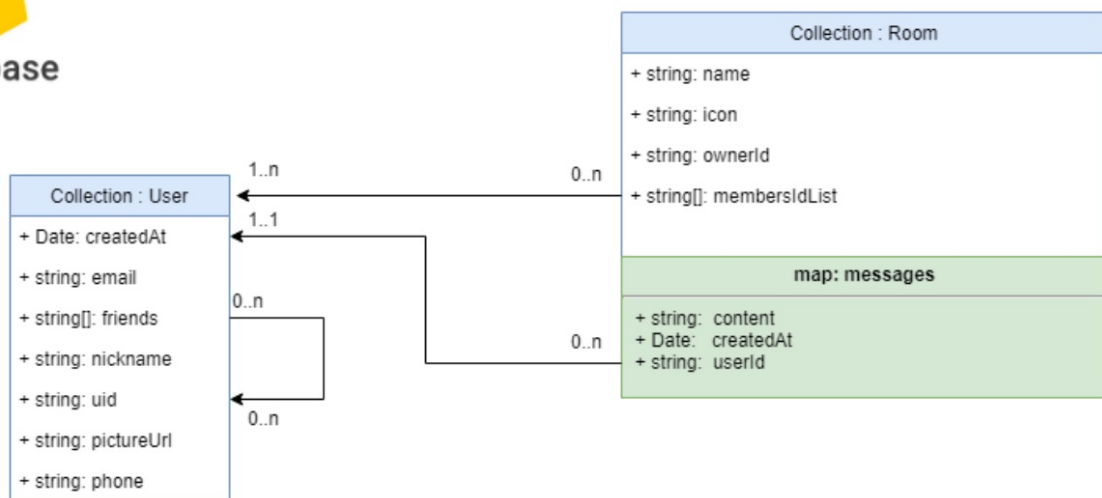
La branche master possède un README.md qui contient les infos nécessaire au rendu.

Cloner un repo local :

```
> git clone ./accord.git
```

BDD

Nous avons utilisé firestore de firebase pour stocker les données voici le schéma de celle-ci:

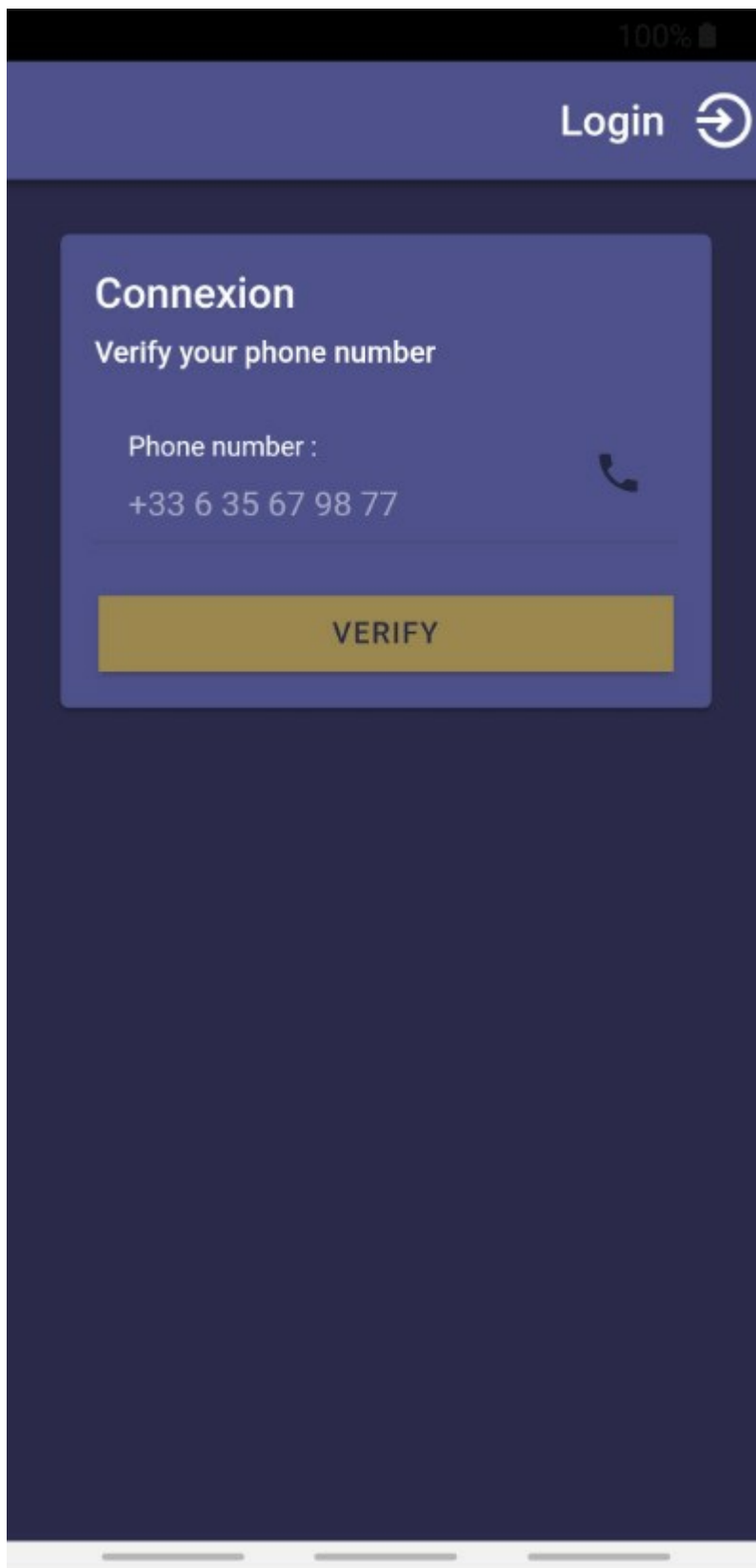


Authentication

Nous avons également utilisé firebase auth pour simplifier le backend d'authentification et accélérer le développement.

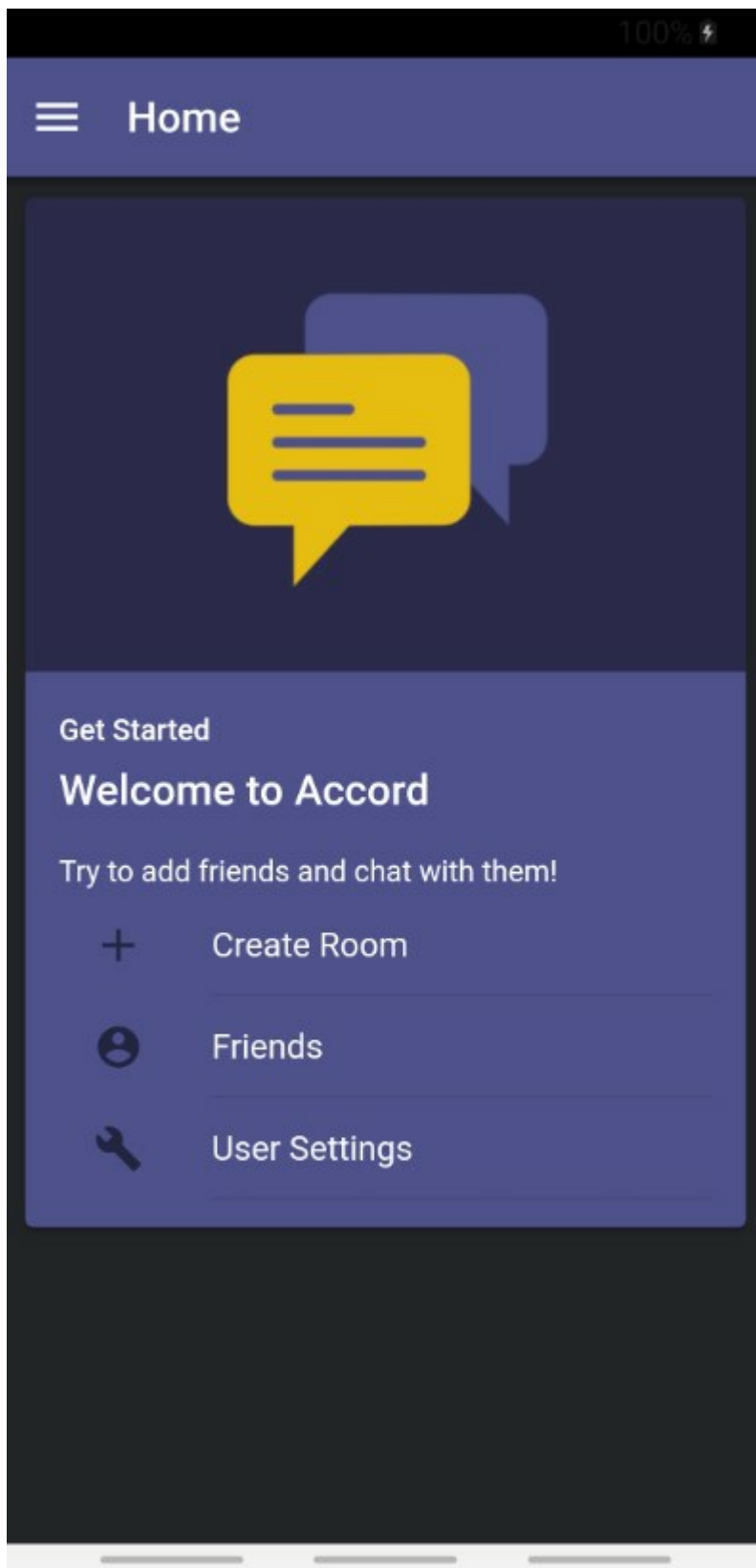
Nous avons utilisé le plugin cordova firebase pour s'authentifier de manière sécurisé depuis l'application.

L'utilisateur doit s'authentifier avec son numéro de téléphone, il reçoit alors un sms de validation qu'il entrera pour valider son numéro. Il sera alors authentifié.



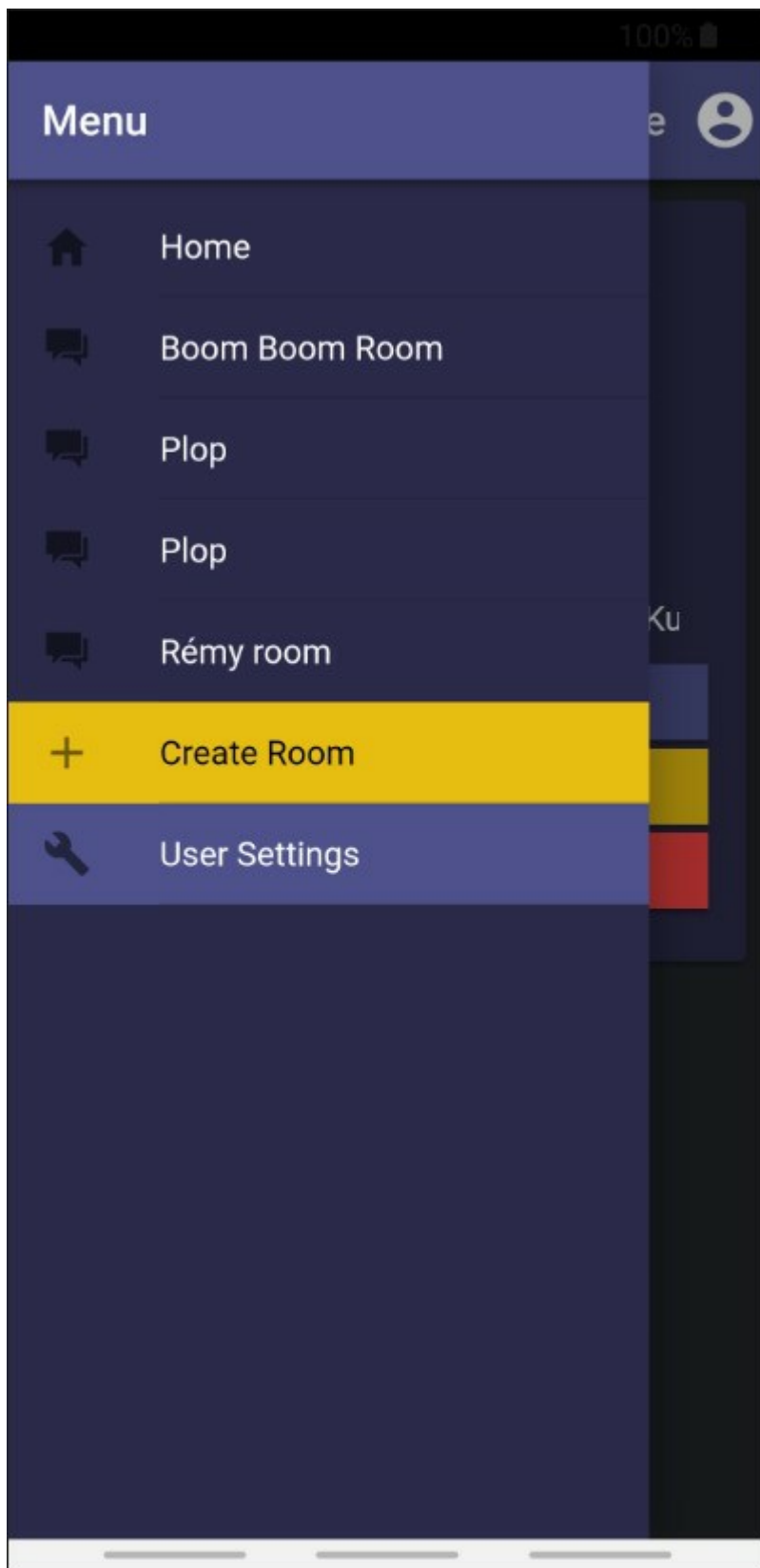
Home

L'utilisateur est alors redirigé sur la page home où il a accès à toutes les fonctionnalités de l'app.



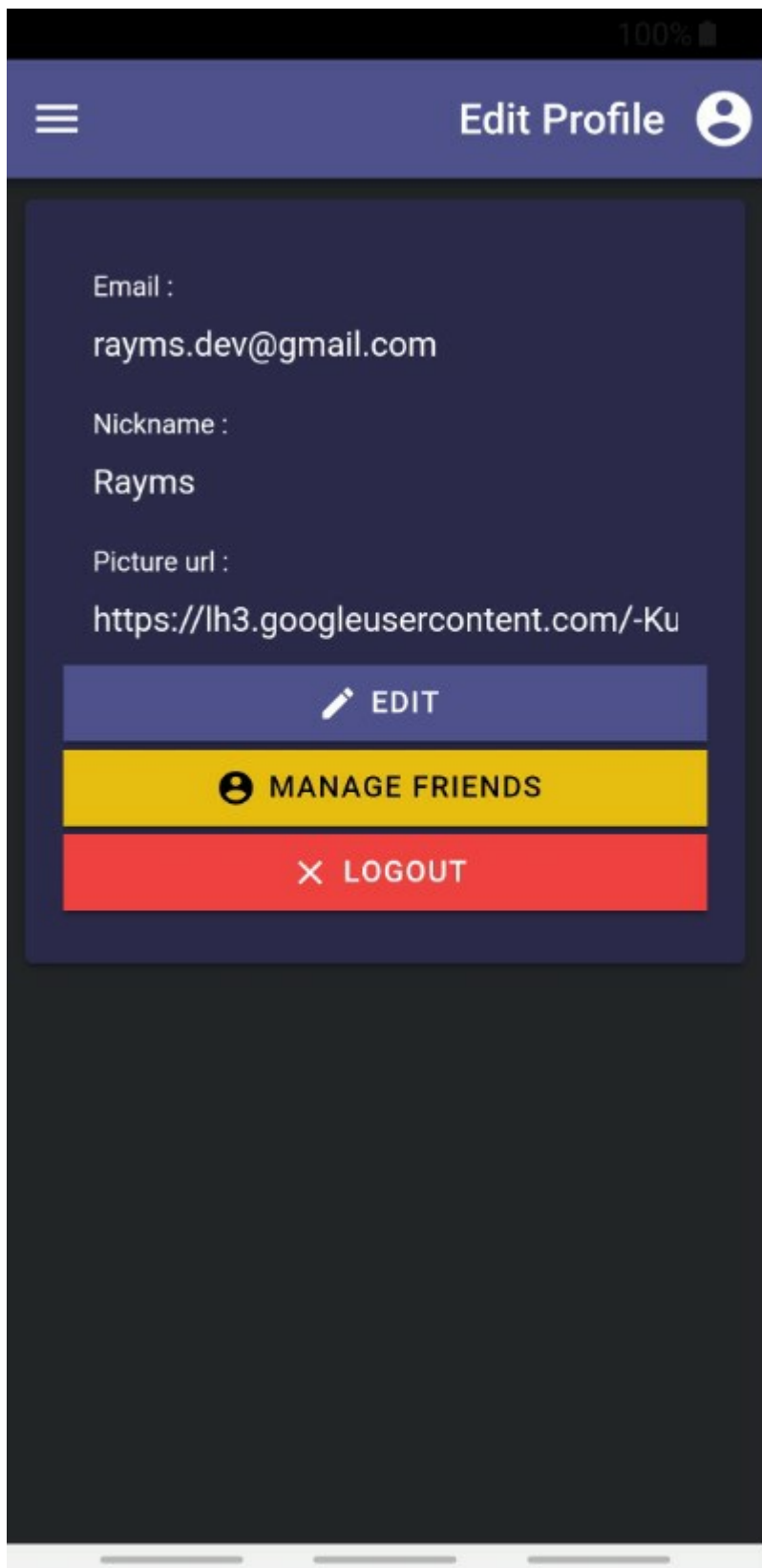
Navigation

La navigation dans l'app se fait dans le side menu



Réglage

L'utilisateur peut modifier ses informations dans la page d'édition de profil

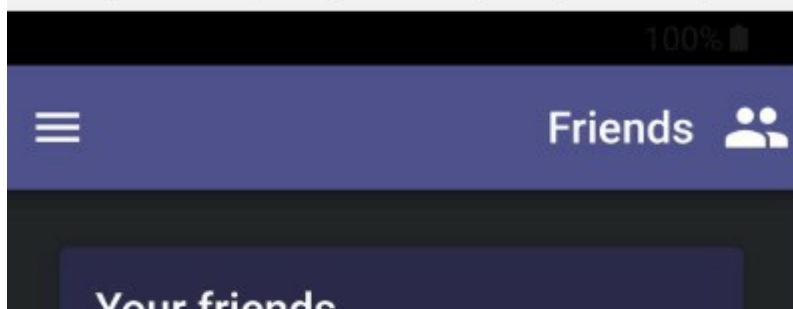
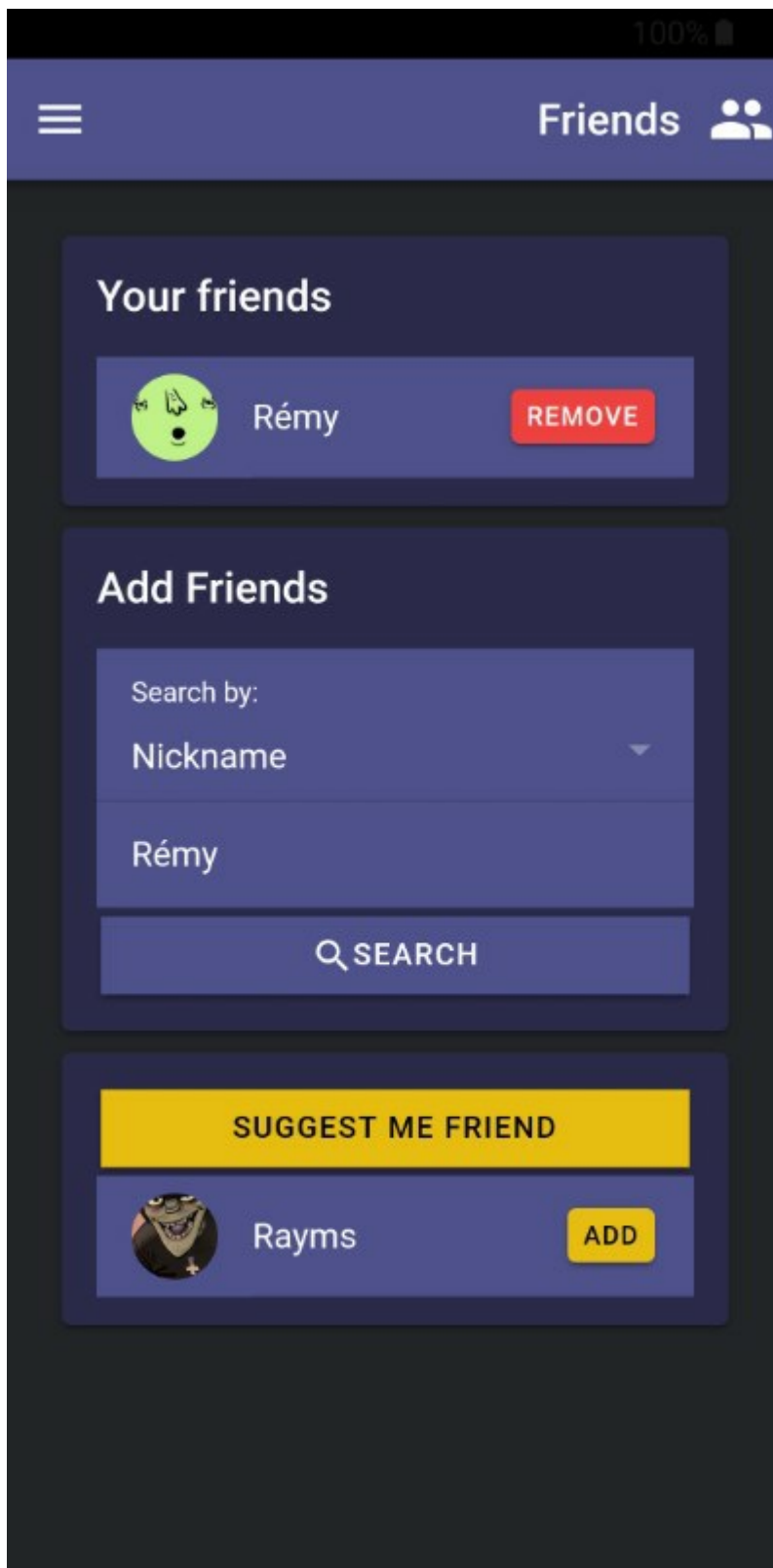


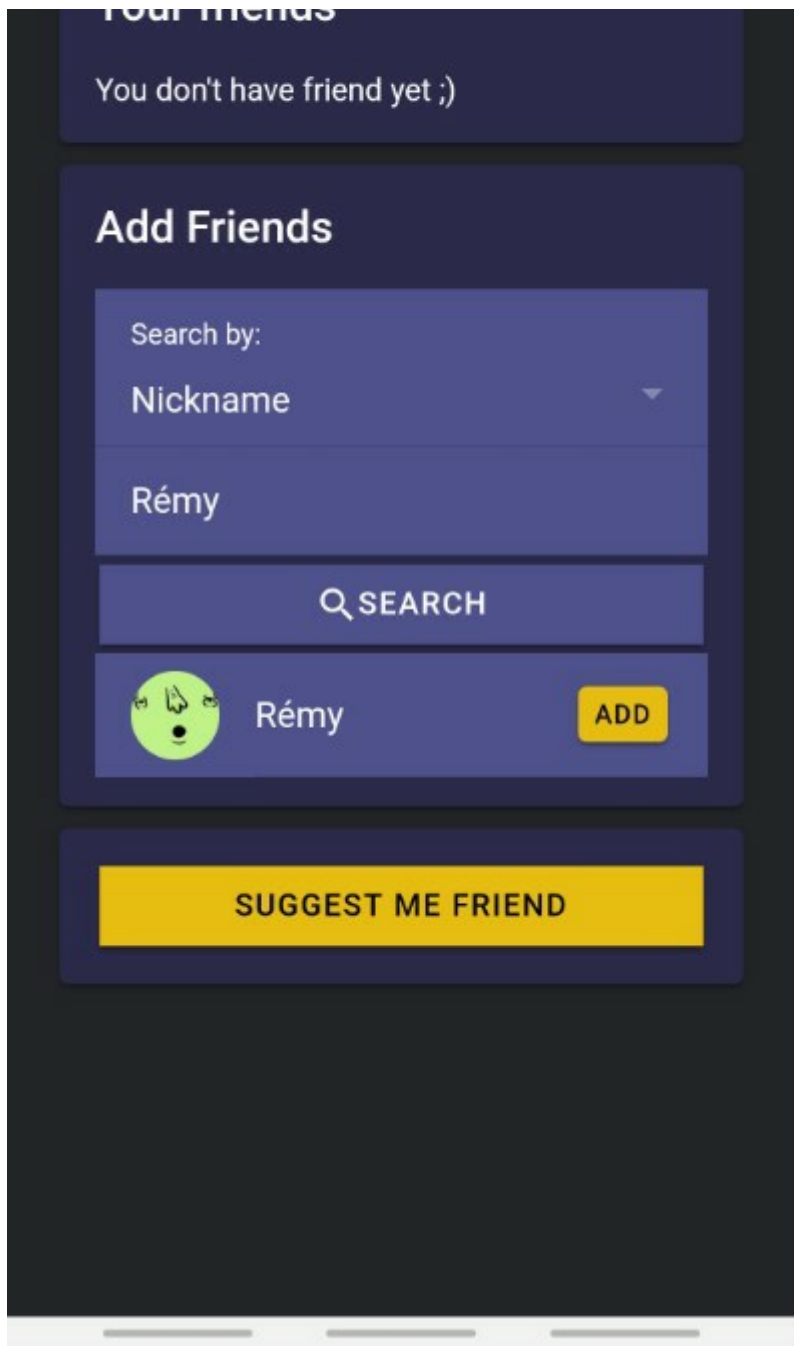
Friends

L'utilisateur peut gérer ses amis depuis cette page.

Il peut soit rechercher un utilisateur par numéro de téléphone ou par username.

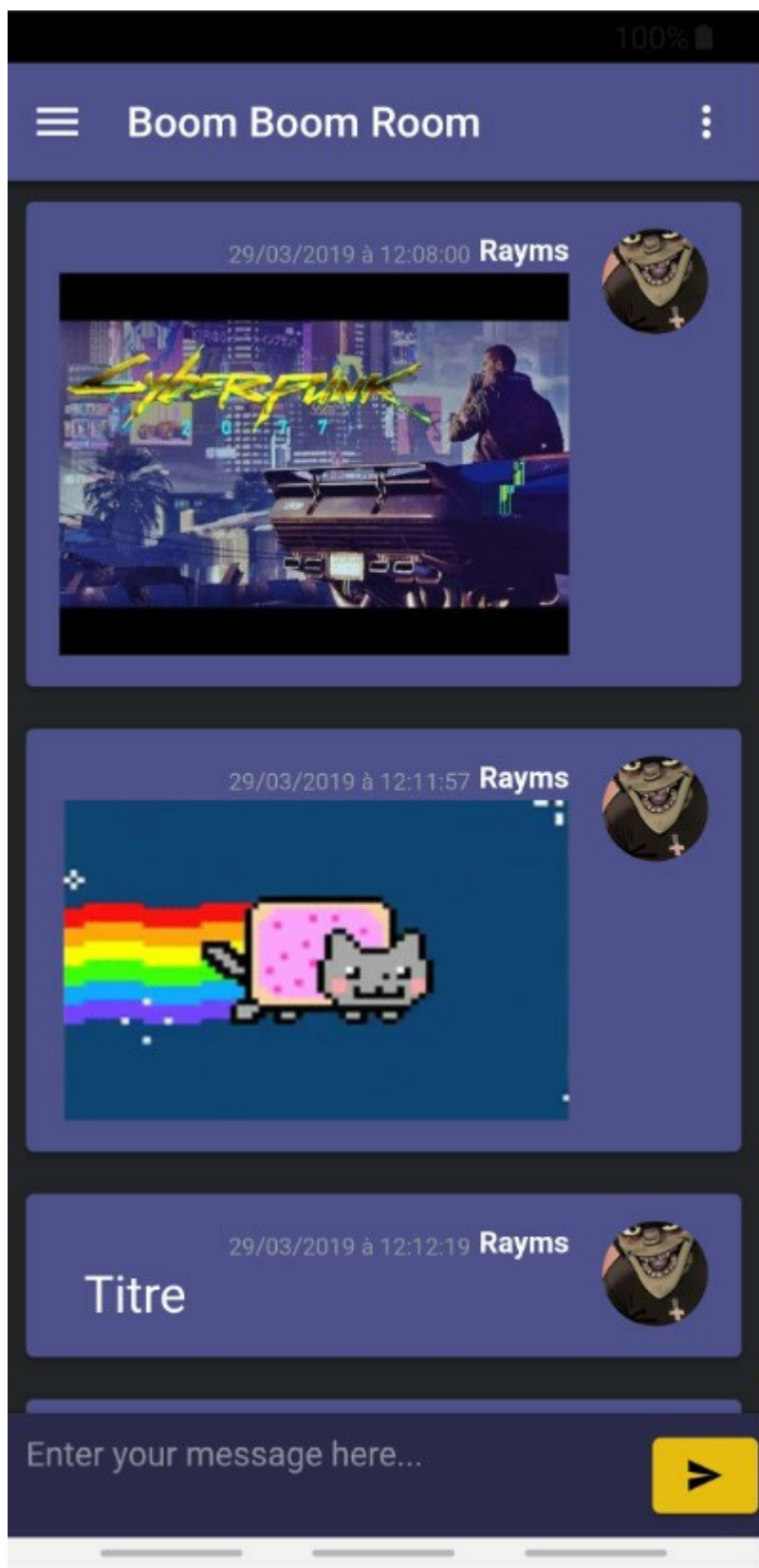
Nous avons également utilisé le plugin cordova permettant d'accéder aux contacts du téléphone pour proposer des amis en fonction des contacts de l'utilisateur via le bouton "SUGGEST ME FRIEND".



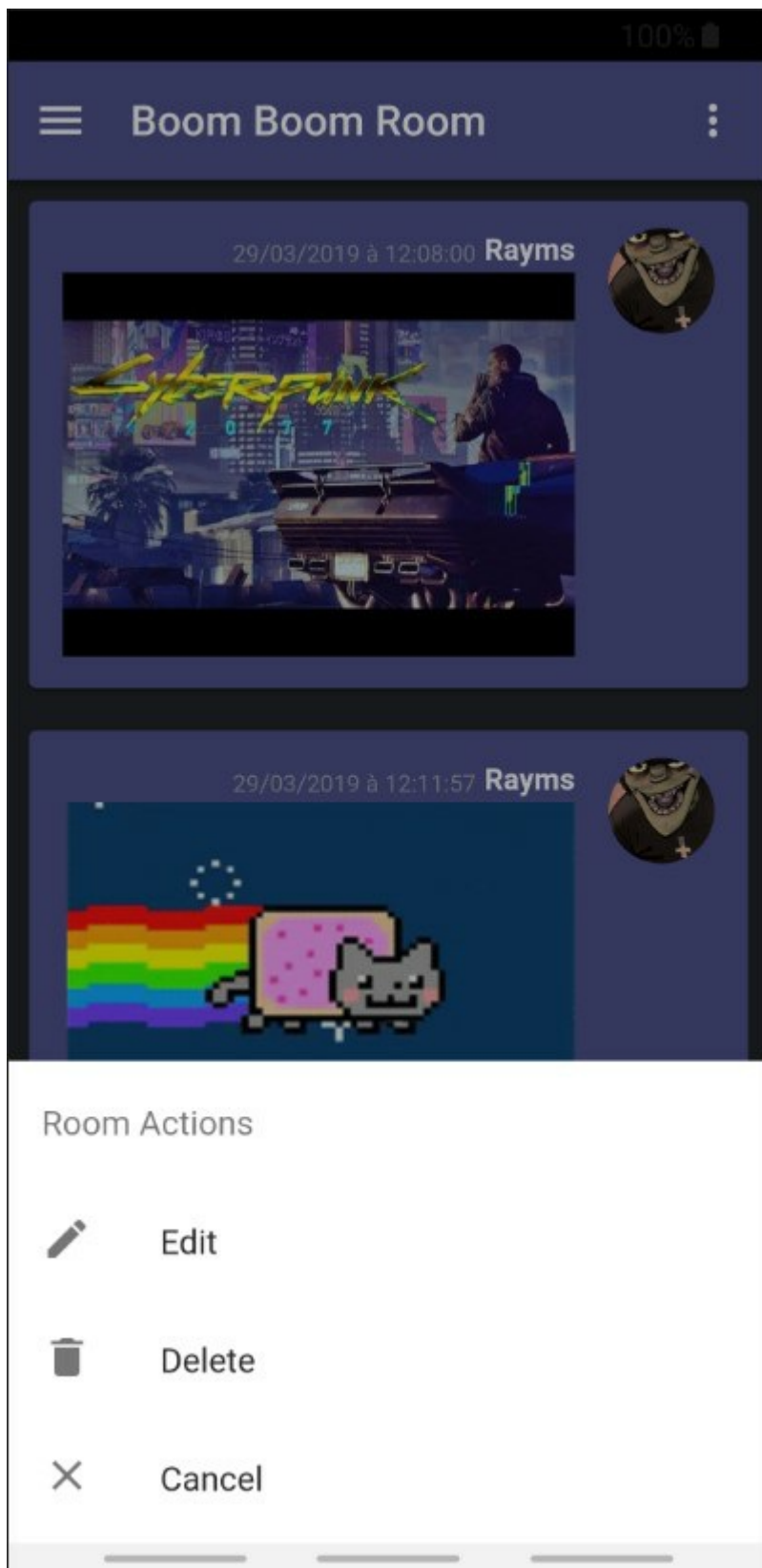


Room

L'utilisateur peut créer des rooms et inviter des amis afin de s'envoyer des messages. Le format Markdown est accepté et l'utilisateur peut envoyer des images , videos et autres via ce langage



Il peut administrer la room via les options:



Tester Accord !

Le projet nécessite node.js (lts) et npm (6.9.0)

Installation des dépendances:

```
> npm i
```

Initialisation de cordova

```
> npm run platforms
```

Test de l'application en mode Web

```
> npm start
```

Test sur un device (Android)

```
> npm run android
```

Création de l'apk signé et aligné (nécessite docker)

```
> npm run docker-apk
```

DevOps

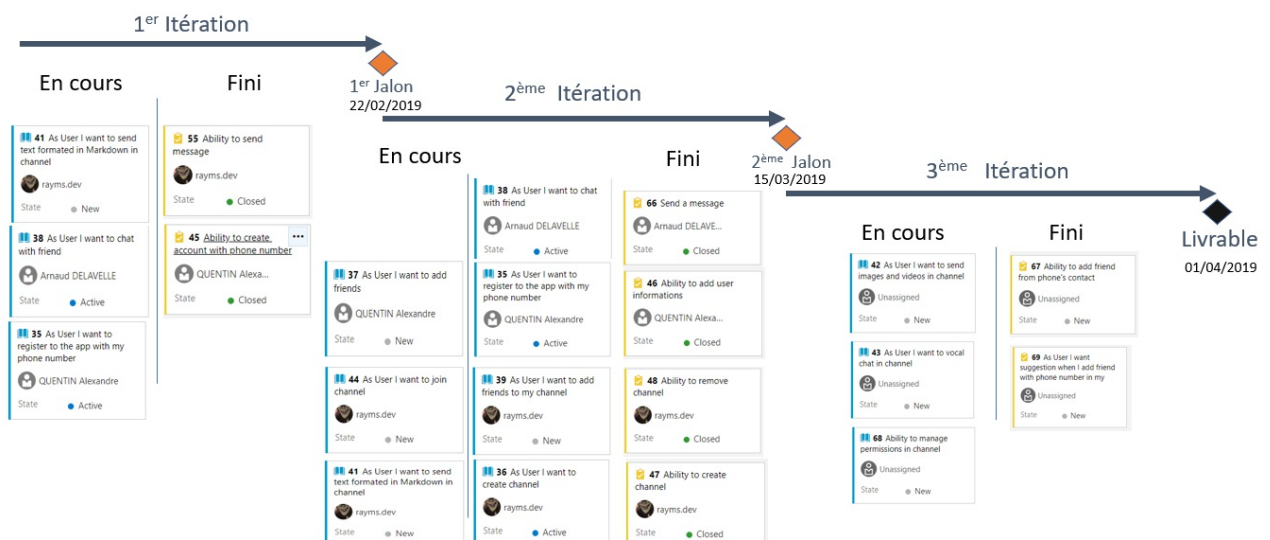


Hello Azure DevOps

Nous avons utilisé une image docker pour build l'apk ainsi nous avons mis en place un build. Chaque push sur la branche master. Lance un build de l'apk, une fois build, l'apk est envoyé à une liste d'utilisateur pour être testé directement ce qui nous a permis d'identifier rapidement les bugs.

Gestion du projet

Nous avons également utilisé Azure DevOps pour gérer notre projet en mode agile.



Gestion des sources

Nous avons utilisé git avec [gitflow \(http://danielkummer.github.io/git-flow-cheatsheet/index.fr_FR.html\)](http://danielkummer.github.io/git-flow-cheatsheet/index.fr_FR.html) dans Azure DevOps pour sauvegarder et gérer notre code.

Ce repo est privé, c'est pourquoi nous ne sommes pas souciés du commit des fichiers sensibles.

Diffcultés rencontrées

- Utilisation de l'API native des smartphones
- Mise en place de l'environnement JAVA/GRADLE/ANDROID
- Reactive programming

Merci de votre patience