Projet Android MBDS 2018/2019

Objet : Création d'un Application de messagerie chiffrée de bout à bout

Environnement utilisé pour nos tests

Nous avons testé l'application sur des mobiles Android avec un API supérieur ou égale à 23. Ainsi que sur des émulateurs sous API 28.

Scénarios

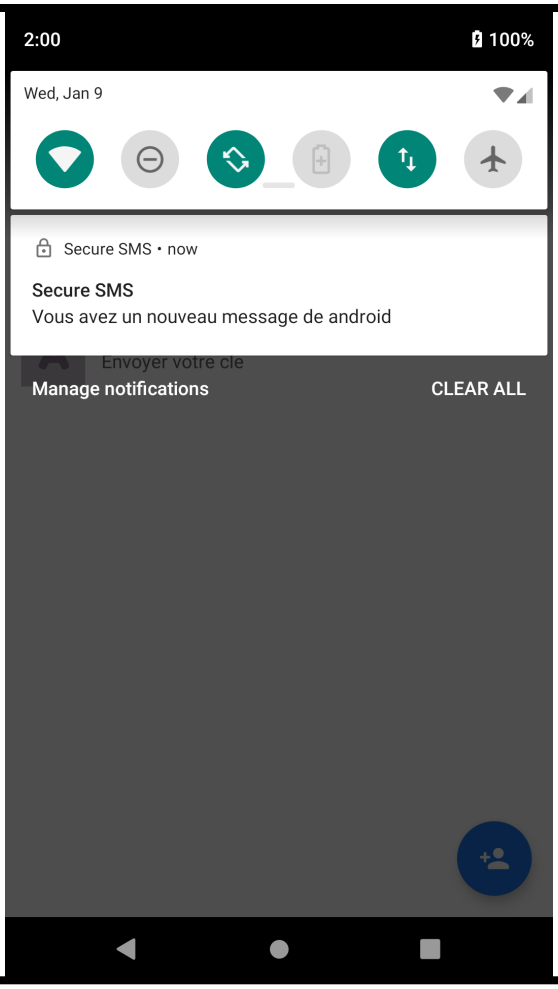
Au démarrage de l'application nous avons une première activité qui se lance (SplashScreen) dans cette activité on vérifie si il existe une session c'est-à-dire si nous avons un access_token (/!\ nous sauvegardons notre access_token dans un SharedPreference) est valide (URL : http://baobab.tokidev.fr/api/validate), si il y a une session alors on démarre notre service (MyServiceFetchMessage.java) ensuite on est redirigé vers la page principale (MainActivity). Si il y a pas de session en est redirigé vers la page de Login.

Pour l'envoi de messages (ALICE) il faut ajouter un nouveau contact (un contact qui possède aussi un compte). Lors de l'ajout d'un nouveau contact on génère une paire de clef (RSA) et un PING lui est automatiquement envoyé. de l'autre côté le nouveau contact (BOB) reçoit un message PING et il lui suffit de cliquer sur le BUTTON pour générer une clef secrete (AES) et renvoyer un PONG. Une fois le PING et le PONG envoyer les deux utilisateurs (ALICE et BOB) peuvent communiquer.

Le service tournant en arrière plan nous notifie de l'arrivée d'un nouveau message.

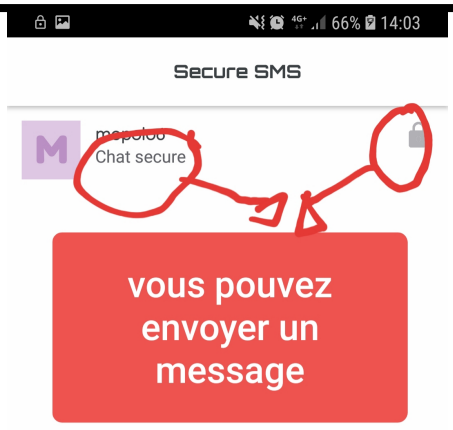
Les Ecrans

Notification

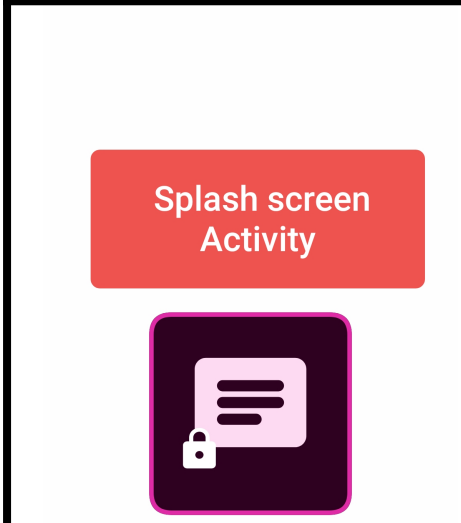


Status de la conversation

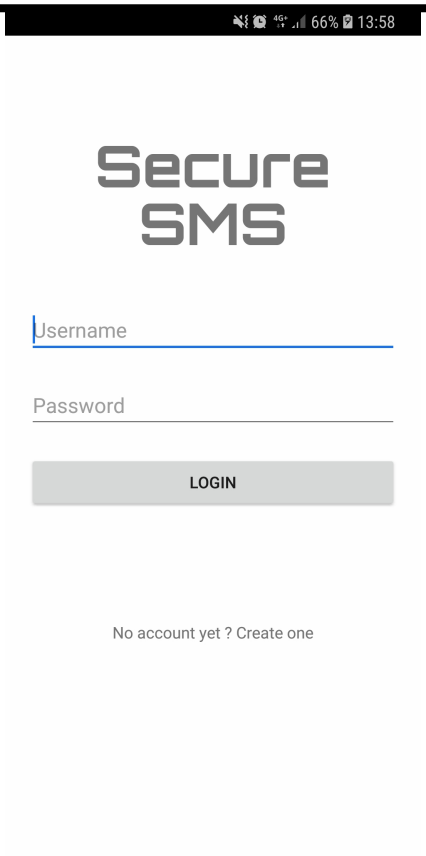
Status de la conversation



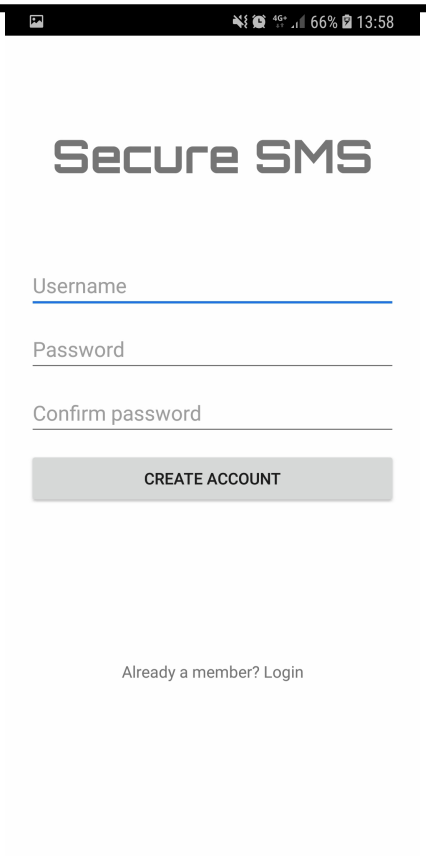
SplashScreen



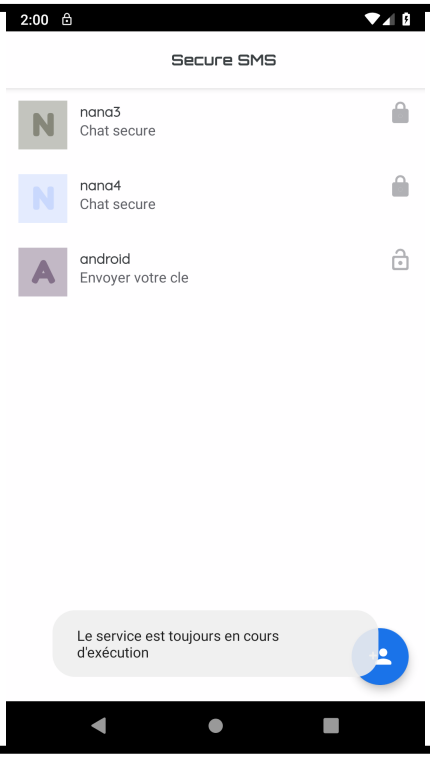
L'interface Login



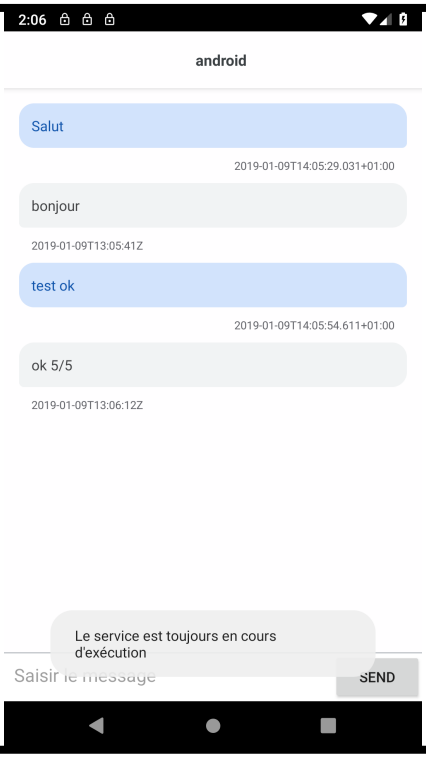
L'interface Register

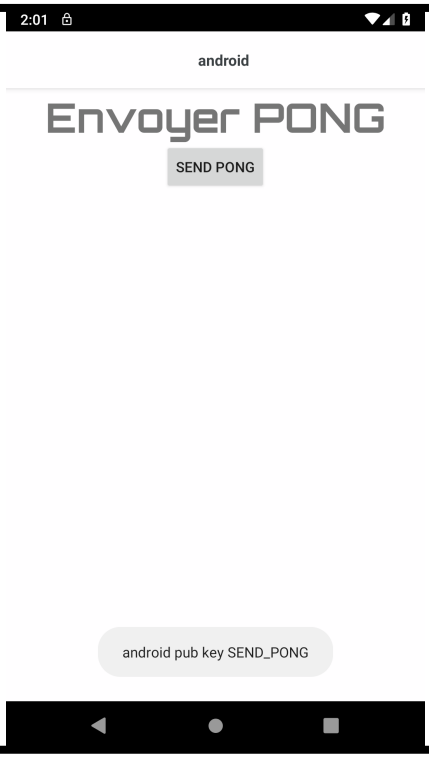


L'interface ContactList

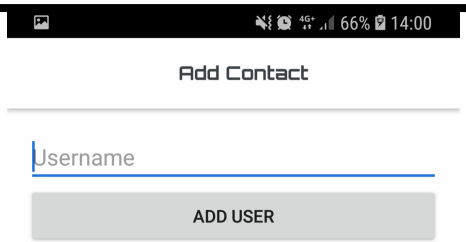


L'interface Messagerie

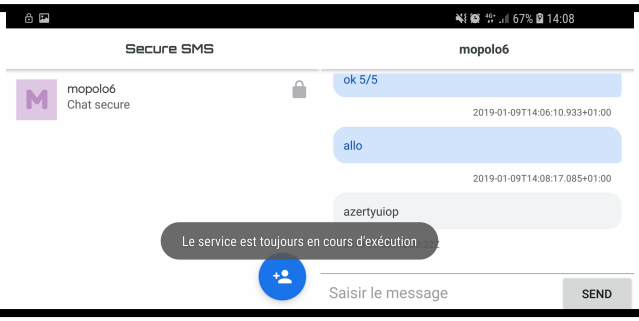




L'interface AjouterContact



Affichage mode paysage



Pourcentage de tache réalisé

- Login (username, password) (connexion) **FAIT à 100%**
- Register (username, password, ...) (créer un utilisateur) **FAIT à 100%**
- ContactList(usernames) (liste contactes avec nombre de messages à lire) **FAIT à 90%**

  A la place du nombre de messages à lire nous avons affichons en rouge le nom de l'utilisateur pour notifier d'un message non lu.
- Messagerie(conversation avec une personne, champ texte, bouton pour envoyer) **FAIT à 100%**
- AjouterContact(username,...) // envoie un message PING à l'utilisateur **FAIT à 100%**

Les Services

Le service est démarré lors de la connexion d'un utilisateur et fait des appel serveur toute les 5 secondes pour récupérer les nouveaux messages.

/!\ Voir la classe **MyServiceFetchMessage.java** dans le package **service**

Intent

Pour le partage il faut au préalable déjà avoir parlé avec une personne c'est-à-dire que l'action PING-PONG est déjà été réalisé et que la conversation est sécurisé.

Modèle de données

Ici nous avant utilisé 2 tables pour notre application.

User : pour sauvegarder les utilisateurs que nous ajoutons comme contact
Message : pour sauvegarder les messages en local

User		
uid	INT	identifiant d'un utilisateur
username	TEXT	le nom d'un utilisateur (unique)
thumbnail	TEXT	une image (pour l'instant représenté par une couleur aléatoire)
status	TEXT	pour notifier si la clé a été transmise et si la conversation est sécurisé
publicKey	TEXT	représente la clé publique généré par celui qui envoi le message en premier
privateKey	TEXT	représente la clé privée généré par celui qui envoi le message en premier
aesKey	TEXT	représente la clé Secrete généré par celui qui reçoit une demande de conversation

Message	
uid	INT
author	TEXT
authorKey	TEXT
message	TEXT
alreadyReturned	BOOLEAN
currentUser	BOOLEAN

Arborescence des Classes java

Membre :

- **BOUKOU Grace**
- **Askia Mohamed**