

Rapport De Projet

Application Android de messagerie instantanée géopositionnée

Sommaire

1 Introduction	2
2 Fonctionnement global	2
2.1 Identification	
2.2 Liste des contact	
2.3 Status	
2.4 Messagerie instantanée	
2.5 Envoi de fichier	
2.6 Inviter un contact	
2.7 Géopositionnement	
3 Technologies utilisées	4
3.1 Compatibilité 2.2	
3.2 Protocole de communication	
3.2.1 Bonjour	
3.2.2 Typa	
4 Difficultés rencontrées	6
4.1 Ce qui n'a pas pu être réalisé	
5 Conclusion	7

1 Introduction

Ce projet a pour objectif de développer une application Android de A à Z permettant de discuter instantanément avec d'autres personnes possédant l'application. Elle doit également permettre de s'échanger des fichiers, musiques, images et de géolocaliser la personne avec laquelle on discute.

Dans ce rapport nous allons expliquer comment fonctionne notre application, quelles sont les technologies que nous avons utilisées afin de rendre chaque fonctionnalité possible. Nous présenterons ensuite les problèmes que nous avons rencontrés et finirons par une conclusion.

2 Fonctionnement global

2.1 Identification

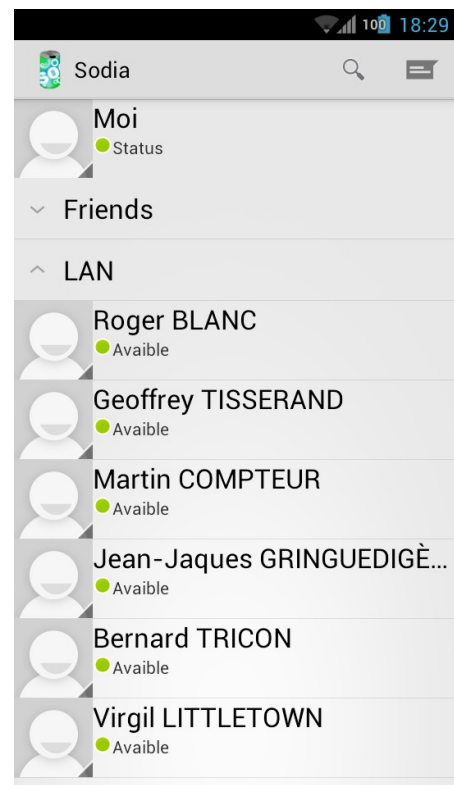
Lorsqu'on lance l'application pour la première fois, il est demandé de renseigner un compte de type Typa (protocole fait maison à l'aide de Bonjour). On est alors dirigé vers les réglages de compte et synchronisation.



2.2 Liste des contacts

Une fois connecté, nous arrivons directement sur la liste des contacts présents sur le réseau. Les pseudos des contacts, ainsi que leur statut et leur image perso sont affichés dans cette liste.

En appuyant longtemps sur un contact, un menu contextuel apparaît proposant la localisation ou la conversation. Il est également possible de discuter avec un contact avec un simple clic sur celui-ci. Au-dessus de cette liste, on affiche de la même façon que les autres contacts, notre propre fiche.



2.3 Status

En cliquant sur notre compte, une boîte de dialogue apparaît permettant de modifier son Pseudo, son image et son status.

2.4 Messagerie instantanée

En cliquant sur un contact, on arrive dans la fenêtre de chat. En haut de l'activité se trouve le nom de la personne avec qui l'on discute, si il y a plusieurs personne il est marqué "Group". En haut de l'écran se trouve également un menu (actionbar) permettant d'accéder aux fonctionnalités d'ajout de contacts à la conversation, d'envoi de fichiers et de localisation.

Le centre de l'activité est destiné à l'affichage des messages du chat.

En bas se trouve le champ permettant d'envoyer un message.

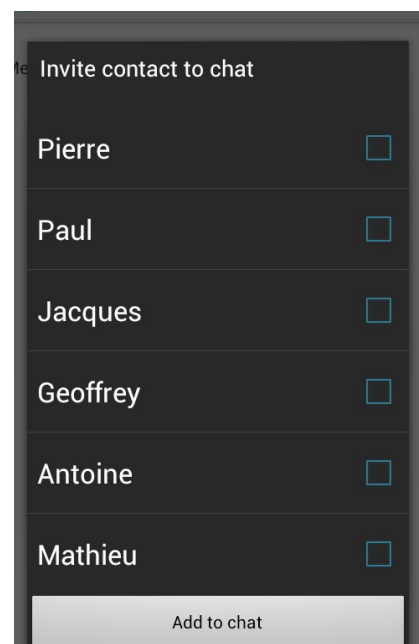
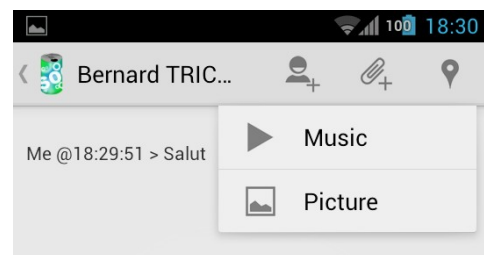
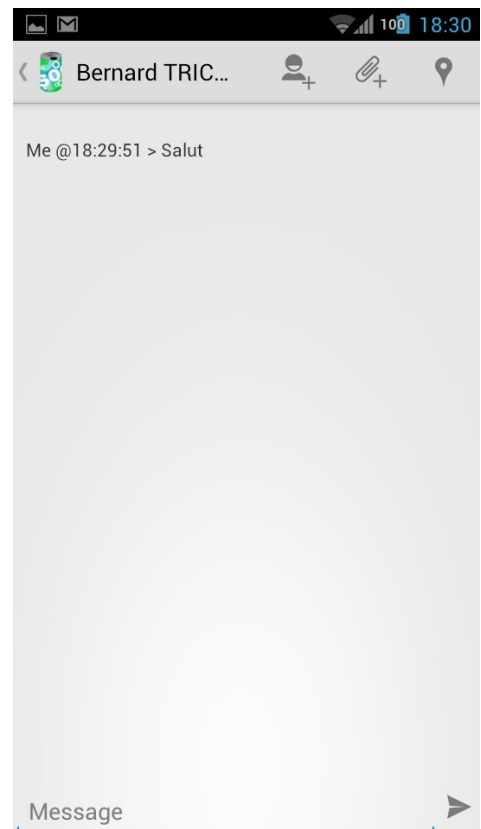
2.5 Envoi de fichier

Lorsque l'on clic sur l'icône d'envoi de fichier, un sous-menu s'ouvre et propose d'envoyer une image ou une musique. Lorsque l'on clic sur l'un de ces items, on est redirigé sur la galerie d'images ou de musiques d'Android.

Une fois le fichier, l'image ou la musique sélectionnée, il est envoyé à tous les membre du chat.

2.6 Inviter un contact

Lorsque l'on clic sur l'icône ajouter un contact, une fenêtre de dialogue apparaît. Celle-ci permet de sélectionner un ou plusieurs contact parmi la liste des contacts en ligne. Un bouton permet d'ajouter ces contacts à la conversation.



2.7 Géopositionnement

Lorsque l'on clic sur l'icône permettant de localiser le contact, une nouvelle fenêtre apparaît proposant une map déjà centrée sur le contact en question et sur laquelle toutes les positions des contact sont affiché à l'aide de pins.

Cette activité utilise les librairies de google maps. Lorsque l'on clic sur un pins positionnés sur la map, une boîte de dialogue apparaît permettant de connaître le pseudo de l'utilisateur et les distances par rapport à notre position.

3 Technologies utilisées

Le projet a été réalisé entièrement dans la **version 4.0.3** d'Android grâce à des IDE comme Eclipse, NetBeans ou encore Geany possédant les plugins nécessaires au développement d'applications Android.

Nous avons également utilisée Git afin de travailler collaborativement et de gérer le versioning du projet.

Afin que notre application fonctionne correctement sur les tablettes proposées par l'UV et tournant sur une version obsolète d'Android, la 2.2.1, nous avons utilisé deux librairies spécifiques.

3.1 Compatibilité 2.2



La première librairie utilisée permettant la rétro-compatibilité est ActionBarSherlock. Cette librairie permet de rendre compatible les composants graphiques appelés ActionBar, très utilisés sous Android dans sa dernière version.

ActionBarSherlock est disponible à cette adresse :

<http://actionbarsherlock.com/>

Nous avons également utilisé la librairie Holo-Everywhere afin de porter le thème populaire d'Android 4.0 sur les versions précédentes.

Holo-Everywhere est disponible à cette adresse :

<http://android-developers.blogspot.fr/2012/01/holo-everywhere.html>

L'utilisation de ces librairies à imposé quelques adaptations spécifiques du code ainsi que la lecture et l'appropriation de la documentation de ces modules.

3.2 Protocole de communication

3.2.1 Bonjour



Pour cette application, un protocole a été écrit afin de communiquer sur le réseau Wi-Fi. Ce protocole repose sur Bonjour, protocole libre proposé par Apple et permettant de découvrir les services disponibles sur le réseau. Pour cela il a fallu utiliser la bibliothèque jmdns pour pouvoir découvrir les services de type "_typa._tcp._local.".

Une fois découvert le service est ajouté à une liste de clients. Chaque client conserve une liaison avec la partie serveur sur le port 4242. Une fois établie la connexion est utilisé afin de demander la liste des contacts présents sur l'hôte distant. Le serveur répond à la demande en envoyant la liste des comptes hébergés sur l'appareil.

La découverte de services fonctionne étonnamment bien à tel point qu'il nous a été possible de se découvrir d'un bout à l'autre de l'UTBM en utilisant le réseau "eduroam". Cela peut aussi poser problème car cela produit une découverte de services locaux (le téléphone se détecte lui-même) qu'il faut alors filtrer.

3.2.2 Typa



Pour les besoins du projet un protocole de communication simple a été conçu. Ce protocole est basé sur un formatage CSV (Comas Separated Values) qui permet d'envoyer des données délimitées sur un réseau. Le protocole est défini de la manière suivante :

REQUETTE;TAILLE;TYPE;DESC;DESC;DESC... où :

- **REQUETTE** est de type **GET** pour une demande d'information (taille à 0) et **RET** pour une déclaration d'informations.
- **TAILLE** est le nombre d'éléments envoyés
- **TYPE** est soit **CONTACT**, soit **MESSAGE**
- **DESC** dépend de **TYPE** :
 - **CONTACT;ADRESSE;NOM**
 - **MESSAGE;EXPEDITEUR;NB_DEST;DEST;DEST;DEST;TYPE;DONNEE**
 - S'il n'y a pas de destinataires le message doit être envoyé à tout le monde
 - **TYPE** est soit **PRESENCE** (pour notifier des changements de statut), **ROSTER**, **POSITION** (pour donner sa localisation GPS), **TEXT**, **PICTURE**, **SOUND**, **ARTICLE**, ou **RAW** (pour des fichiers)

Ce protocole a été utilisé avec succès entre appareils ainsi qu'à l'aide d'une console telnet (requêtes manuelles). Le décodage et encodage de divers types de requêtes fonctionnent.

4 Difficultés rencontrées

Pour tous les membres de l'équipe il était question de développer une application Android pour la première fois.

Ayant envie de commencer à développer sur une version non obsolète d'Android nous avons été optimiste en choisissant de développer pour la dernière version en espérant ensuite obtenir une compatibilité totale pour les anciennes versions. Cette compatibilité n'étant pas assuré cela a nécessité du code supplémentaire.

Malgré nos nombreuses heures passées à déboguer étape par étape notre application, nous n'avons pas réussi à obtenir le résultat que l'on souhaitait. Un problème de communication n'a pas pu être résolu ne nous laissant pas la possibilité de répondre à la plupart des requêtes. En effet le serveur obtient une voie de communication avec le client qui ne peut pas traiter les réponses, le client doit alors chercher le serveur correspondant dans une liste et envoyer la réponse au serveur. Cette dernière étape pose problème et n'a pas pu être surmontée. Le serveur est résolu avec le bon port mais lors de l'envoi le serveur semble, la plupart du temps ne rien recevoir. En revanche le protocole et son décodage et affichage fonctionne, on a réussi à afficher un contact reçu en mettant la liste des contacts à jour, et les messages sont correctement décodés et affichés.

Une autre difficulté fut la contrainte sur la taille de l'écran. Nous avons fait tout notre possible pour rendre cette application un maximum ergonomique et adaptable à tous types d'écrans.

Enfin ne pouvant pas emprunter ces dites-tablettes il était assez contraignant d'utiliser l'émulateur Android pour ceux qui ne possédaient pas d'appareil compatibles. Cet émulateur est en effet très lent et rend difficile voire impossible certaines fonctionnalités (connectivités, fonctionnalités hardware). Il était aussi nécessaire d'avoir plusieurs appareils à disposition, et l'étendue de la diversité Android a fait que sur certains de nos appareils, le même code avait un comportement différent.

4.1 Ce qui n'a pas pu être réalisé

Devant la multitude de projets proposés durant ce semestre ils nous a été impossible de développer toutes les fonctionnalités que l'on imaginait.

En effet nous voulions développer un système de vraies notifications Android permettant de notifier l'utilisateur d'un nouveau message.

Aussi la possibilité d'envoyer une photo directement prise de l'appareil photo en utilisant celui du device n'a pas pu être implémentée.

Il était aussi prévu de filtrer les contacts à l'aide du champ de recherche, et d'implémenter un système de photo pour les contacts. De même un affichage dynamique des divers types de messages dans la liste de discussions n'a pas été réalisé.

5 Conclusion

Nous sommes clairement déçus de ne pas avoir pu résoudre ce problème de communication. En effet l'interface et la couche métier sont complets et n'attendaient plus que cette partie communication pour être utilisés.

Ce projet aura tout de même été très bénéfique pour tous les membres de l'équipe. Nous avons en effet pu découvrir cette technologie Android qui est l'avenir du développement mobile.

Avoir une expérience sur un projet de cette taille et de cette utilité est un vrai plus pour nos CV.

Il est également possible de continuer ou de reprendre ce projet afin d'en faire une application de messagerie assez pratique.

De plus l'aspect utile de l'application nous a donné à tous l'envie de faire un projet bien fait respectant les concepts d'Android et ayant une ergonomie certaine.

Nous avons également pu perfectionner notre capacité de travailler en équipe avec notamment l'utilisation de Git qui était nouveau pour certains membres de l'équipe.