Rapport ANDROID

# Presentation du projet

Le but du projet fut de réaliser un chat bluethoot permettant à deux devixes de s'échanger des données aux travers d'une connexion et de garder l'historique de toutes les conversations passées.

Trois layouts sont donc essentiels à ce projet. L'historisations des données au travers d'un parser XML/JSON ou d'une base de donnée SQLite. L'affichage de ces données afin de pouvoir changer de layout et remettre à jour la ListView. Et enfin la connection en bluethoot avec un service bluethoot.

# Repartition des tâches

Afin de pouvoir travailler ensemble nous avons mis en place un depot Bitbucket et reparti les taches pour ne jamais pietiner sur les mêmes classes. La partie d'interface fut realiser rapidement grâce au storyboard qui nous a permis de voir que nous avions besoin que de trois layouts. **(c.f : storyboard)**

Puis pendant que un travaillais sur la base de donnée et rendre l'interface plus modulaire, l'autre mettais en place les classe pour la connection bluethoot.

La derniere étape fut de lier tout ensemble et obtenir la recette de notre projet.

# Problématiques rencontrés

Nous avons du prendre plusieurs choix durant l'élaboration de notre projet.

Ayant eu l'envie de travailler sur le coté design de l'application grâce à l'interface d'Android et le material design de Google. Nous avons du stopper cela lorsque nous nous sommes rendus compte qu'il ne s'agissait que de bonus et n'était pas pris en compte dans la note. Vous pourrez notamment voir le debut de ce travail dans les fichier de ressource style.xml pour le choix des couleurs et l'ajout d'ombre.

De plus les Listview etant de base non personnalisable, nous avons du mettre en place des Adapter contenant des LinearLayout permettant ainsi de changer l'interface en fonction du message reçu comme celui d'un chat. Nous avons aussi dû gerer la fermeture du clavier du device et forcer le rafraichissement de la ListView qui ne se faisait pas automatiquement.

Etant donné que nous gerons à la fois les personnes et les messages, nous avons aussi du mettre en place une base de donnée. Ce qui c'est avéré plus rapide de part la vitesse de traitement des requetes par rapport à un parseur. Mais aussi la gestion des donnée avec une DAO que nous avons mis en place. La base de donnée est tres simple avec une classe personne contenant un nom et un ID et la classe message avec un text, un ID, un type ( reçu ou envoyé) et une réference à une personne. Deux methodes sont mises en place pour la classe message. L'ajout d'un message en recevant l'ID de la personne et l'obtention de tous les messages avec un ID de personne. Pour la classe Personne aussi deux methodes, l'obtention de toutes les personnes et l'ajout d'une personne qui verifiera que ce nom n'existe pas.

# Réalisation final

Lors du lancement de l'application nous arrivons sur une interface qui va lister le noms de toutes les personnes avec qui nous avons communiquer. En cliquant sur un nom, la ListView va se transformer nous permettons de voir la communication. Nous pourrons retourner à la liste de personnes en cliquant sur le bouton retour de notre device.

En haut a droite, dans les parametres, nous avons la possibilité grâce à un menu entre joindre un chat ou créer un chat. L'un aura pour but juste de rendre notre bluethoot visible est en attente. Tandis que pour joindre nous allons ouvrir la liste de tous les bluethoots presents.

Une fois nos deux devices liés, nous pourrons communiquer au travers de l'interface de Chat de notre application.

Un scenario possible de test serait de lié deux devices, communiquer brievement puis retourner à l'ecran d'accueil à l'aide des boutons retour de notre device ou en fermant l'application. Et voir qu'une personne s'est rajouté à la liste de personne. En cliquant dessus nous pourrons alors voir les messages que nous avons envoyé dans l'ordre.

# Améliorations possible

Les améliorations possibles notamment pour la base de donnée serait d'ajouter l'ID bluethoot du device lié au nom. Ce qui permettra de faire une communication avec deux personnes ayant le meme nom bluethoot.

Il faudrait aussi pouvoir rajouter une date pour chaque messages.

Nous avions aussi pour objectif de mettre en place grace au SPEECH\_RECOGNIZER d'android de la reconnaissance vocal pour permettre d'ecrire des textes plus rapidement lorsque nos mains sont prises.

Et enfin nous pourrions améliorer l'interface graphique de l'application à l'aide d'animation et du material design de google afin de rendre notre application plus user friendly.