

Projet Tuteuré 2018

Adeline DUMAS – Johan LANNIER – Nicolas QUILLON



Rapport Technique Projet Temps de Parole

Université Claude Bernard – Lyon 1

Institut Universitaire de Technologie



Licence Professionnelle Informatique Et Mobilité
71 rue Peter Fink – 01 000 BOURG-EN-BRESSE
Téléphone : 04 74 45 50 65 – Télécopie : 04 74 45 50 51
Courriel : iutbourg.info@univ-lyon1.fr
URL : <http://iut.univ-lyon1.fr>

TABLE DES MATIERES

I.	Présentation du projet	3
1.	Présentation du sujet	3
2.	Objectif à atteindre	3
3.	Répartition des différentes tâches à réaliser	3
II.	Présentation des différents projets effectués.....	5
1.	Application mobile.....	5
A.	Fonctionnement	5
B.	Résultats	6
2.	Application bureau	7
A.	Introduction à Recognito :	7
B.	Fonctionnement de l'application	7
C.	Résultats :	8
	Conclusion	9
	Sitographie.....	10

I. PRESENTATION DU PROJET

1. Présentation du sujet

Lors de ce projet tuteuré, il nous a été demandé de réaliser en quinze jours une application mobile sous Android qui permettrait de calculer en **temps réel le temps de parole d'une personne dans une réunion**. Par exemple, dans une réunion comportant quatre personnes, nous devons être capable d'afficher en temps réel un chronomètre qui s'incrémenterait lorsque la personne référente parlerait. Si nous arrivions à faire cela, nous devons ajouter à cette application des chronomètres pour calculer le temps de parole des autres personnes de la réunion. Il serait alors possible d'afficher dans l'application le temps de parole du référent versus le temps de parole des autres personnes de la réunion (incrémenté dans un même chronomètre).

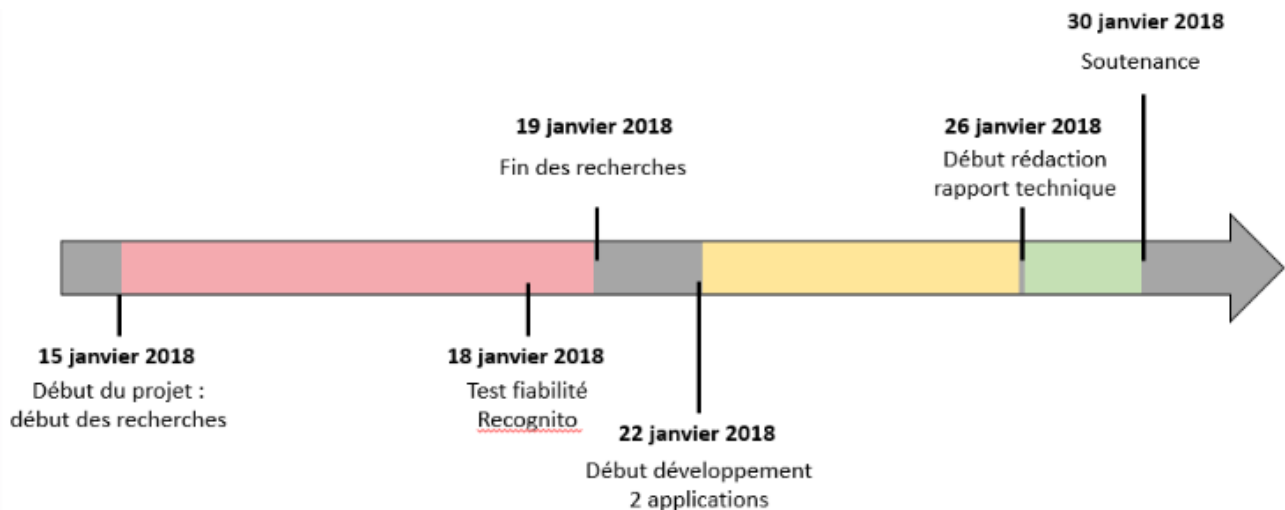
Nous devons réaliser cette application avec l'aide de la société **Break-first** et plus exactement **Benoit Auguin** et **Elia Cocchi**. Nous pouvons également demander plus d'informations concernant ce sujet à **Adam Mebarki**, également étudiant dans la licence professionnelle IEM.

2. Objectif à atteindre

Les objectifs à atteindre étaient que l'application soit tout d'abord fonctionnelle ainsi que la reconnaissance vocale fiable.

3. Répartition des différentes tâches à réaliser

Afin de satisfaire le client et de réaliser l'application dans les temps impartis, nous avons décidé de mettre en place un planning. Voici donc le planning :



Nous avons durant la première semaine, cherché tous les trois des librairies permettant de faire de la reconnaissance vocale compatible sur Android. Cependant nous n'en avons pas trouvé de fiable.

Comme la reconnaissance vocale est une nouvelle technologie, il y a peu de librairies fiables. Nous avons tout d'abord trouvé des **API** notamment celle de **Google** ou de **Microsoft** qui sont très chères et qui par ailleurs ne permettent pas de faire du temps réel.

Nous avons également testé **Alizé** qui est une librairie portable sur Android. Celle-ci fonctionne mais était très difficile à installer. Nous avons donc abandonné cette librairie.

Nous nous sommes aussi intéressés à **SHoUT** qui est une librairie en C++ créée par un étudiant : Marijn Huijbregts. Celle-ci était assez vieille pour la technologie (2010) et était compliquée à installer. Comme le montre cet exemple, la reconnaissance vocale est une nouvelle technologie étudiée par différents étudiants niveau bac+5 et plus. Les librairies pour la plupart, sont développées comme sujet de thèse et n'aboutissent pas à une librairie fiable.

Enfin, nous avons testé VeriSpeak qui est un SDK lié à un serveur. Les résultats n'étaient pas très fiables et nous nous sommes rendus compte par la suite que ce n'était pas de la reconnaissance vocale à proprement parlé mais plutôt un service d'authentification vocal. Il fallait enregistrer une certaine phrase comme mot de passe et le service derrière pouvait matcher suivant l'empreinte vocale et la phrase entendue.

Après avoir discuté avec Benoit Auguin et Elian Cocchi, ils nous ont demandé d'utiliser la librairie Java **Recognito** qui permet de faire de la reconnaissance vocale. Celle-ci, comparé aux autres librairies, ne prend pas en compte des mots comme « clé de reconnaissance », mais reconnaît bien une personne par rapport à sa voix. Recognito cherche dans sa base de données si la personne qui est en train de parler s'y trouve et donne un pourcentage de ressemblance. Pour que la librairie fonctionne correctement il ne faut qu'il y ait pas de bruit de fond et que les personnes parlent fort et distinctement.

Parallèlement, nos professeurs Mr Farrugia et Mr Buathier, nous ont conseillé de réaliser une application Android permettant de faire de la reconnaissance vocale, basée sur le volume.

Durant la deuxième semaine, nous avons développé les deux applications. Johan s'est occupé de l'application mobile et Nicolas a développé l'application bureau avec l'aide d'Adeline, qui enregistrait notamment les cinquante voix. Enfin, le rapport technique a été rédigé par nous trois.

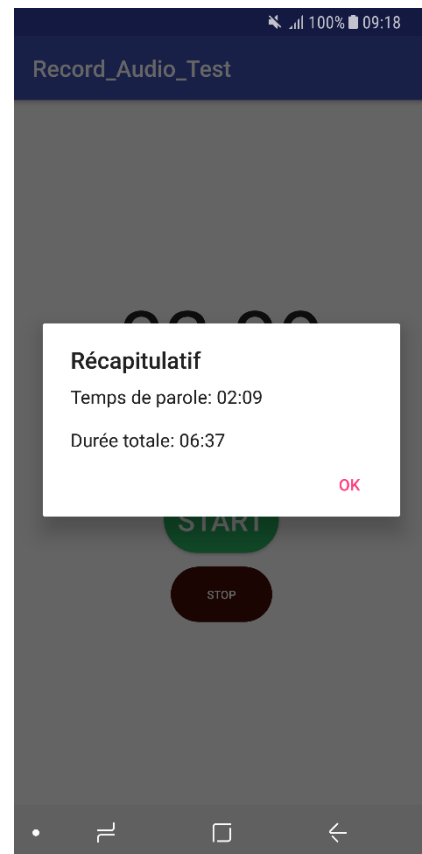
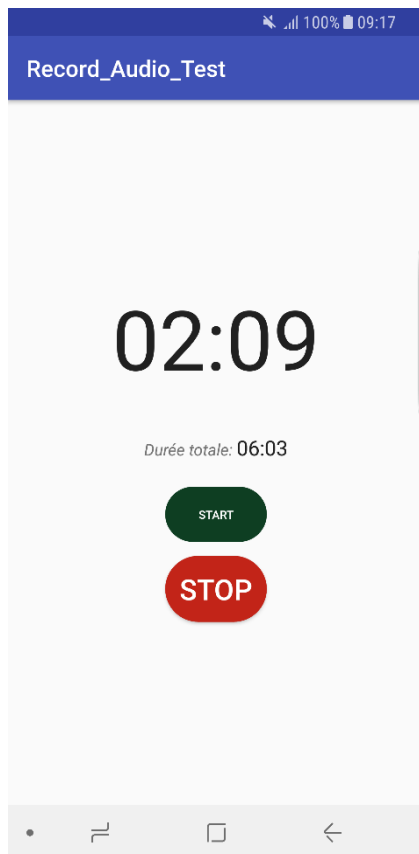
II. PRESENTATION DES DIFFERENTS PROJETS EFFECTUES

1. Application mobile

A. Fonctionnement

L'application mobile permet de chronométrer le temps de parole d'une personne suivant le volume. Comme les micros des smartphones sont très orientés, il est facile de déterminer la personne qui est en train de parler.

L'application est constituée de deux boutons : un bouton start et un bouton stop. Lors d'une réunion, la personne référente pose le téléphone en face de lui et appuie sur le bouton start. Lorsqu'il va parler, le chronomètre du haut s'incrémentera. Si c'est une autre personne, il ne s'incrémentera pas. Il y a également un chronomètre qui détermine la durée totale de la réunion. Lorsque la personne appuie sur stop, une fenêtre récapitulative s'affiche montrant le temps de parole de la personne et la durée totale.



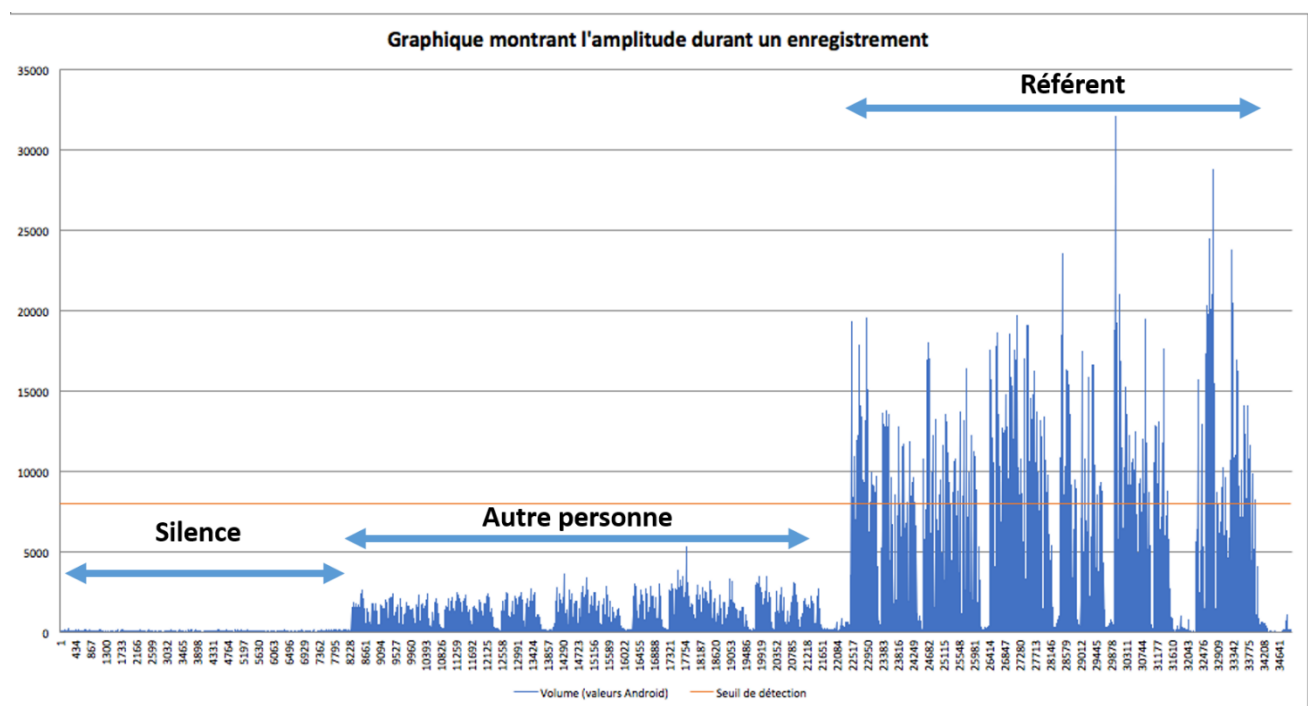
B. Résultats

Pour effectuer cette expérience, il y avait deux personnes, l'une en face de l'autre, le téléphone (Samsung Galaxy S8) tourné vers la personne référente.

Comme nous pouvons le constater dans le graphique ci-dessous, les résultats de cette application sont plutôt très fiables.

Lorsque que personne ne parle, l'application ne détecte rien. Lorsque c'est l'autre personne, il détecte un faible volume en dessous du seuil de 8000. Enfin lorsque c'est le référent qui parle, l'application détecte très bien.

Nous avons établi un seuil de 8000, qui permet de déterminer à partir de quelle valeur l'application doit comptabiliser du temps de parole pour le référent. Ce seuil a été déterminé après plusieurs enregistrements.



2. Application bureau

A. Introduction à Recognito :

L'application bureau est une application Java utilisant la librairie Recognito. Recognito fonctionne ainsi : on enregistre une voix pendant une minute puis il cherche dans sa base de données la personne la plus ressemblante et ressort un pourcentage de ressemblance.

Nous nous sommes rendus compte que pour que Recognito soit fiable il fallait que les **enregistrements se fassent dans le silence**. Nous nous sommes également rendus compte que pour que le pourcentage soit le plus juste possible il fallait **le plus de voix possible dans sa base de données**.

Une fois ces informations connues, nous avons pu effectuer différents tests de fiabilité.

Nous avons tout d'abord enregistré différentes personnes pendant une minute grâce à un Samsung Galaxy S7. Nous avons converti les fichiers enregistrés en .wav puis ajouté dans la base de données de Recognito. Ensuite nous avons effectué d'autres enregistrements de différentes durées. Enfin nous avons lancé Recognito pour qu'il détermine quelles étaient les personnes qui parlaient sur les enregistrements.

Nous avons commencé par des enregistrements de trente secondes. Les résultats étaient très fiables. Nous avons conclu que si le pourcentage était supérieur à 90% il s'agissait bien de la bonne personne. Puis nous avons testé avec des enregistrements de vingt secondes, quinze secondes pour finir à six secondes. Nous avons conclu qu'à partir d'un enregistrement de six secondes Recognito était fiable.

B. Fonctionnement de l'application

Une fois que nous avons effectué différents tests de fiabilité de Recognito, nous avons commencé à développer l'application bureau pour qu'elle fasse du temps réel.

Voici à quoi ressemble l'application :



Nous enregistrons notre voix pendant une minute grâce au bouton d'initialisation. L'enregistrement s'arrête automatiquement à partir d'une minute. Ensuite nous initialisons Recognito. Cette étape est assez longue, plus il y a de voix dans la base et plus l'initialisation est longue (environ trente secondes pour cinquante voix).

Une fois l'initialisation terminée, nous pouvons commencer l'enregistrement de la conversation, en appuyant sur le bouton start. Le programme va enregistrer sept secondes puis à l'issue de ces sept secondes, il va déterminer si c'est bien la voix enregistrée avant ou non. Si oui, il va incrémenter le compteur en dessous du bouton start et nous dire le nom de la personne (enregistrée dans la base, par exemple référent). Pendant qu'il fait ses étapes, il continue à enregistrer et ainsi de suite.

Les boutons en dessous déterminent avec un enregistrement préalable d'Adeline et de Johan la personne qui ressemble le plus dans la base de données. Ils correspondent aux tests précédemment évoqués.

C. Résultats :

Les résultats divergent en fonction du test effectué.

Pour les enregistrements réalisés avec le Samsung Galaxy S7, les résultats sont très fiables. Concernant, les résultats de l'enregistrement en temps réel, ils ne sont pas du tout fiables. Après tous les tests que nous avons réalisés, Recognito ne reconnaissait jamais la bonne personne. Il ne distinguait même pas si c'était une femme ou un homme qui parlait.

Nous nous sommes donc demandés pourquoi les résultats étaient si différents entre les deux tests. Nous en sommes venus à la conclusion qu'il fallait tester l'application en temps réel avec un très bon micro. Les premiers enregistrements ont été effectués avec des micros de smartphones qui comme énoncé plus tôt, sont orientés et enlèvent le bruit de fond. Les enregistrements en temps réel étaient effectués avec un ordinateur qui lui prend en compte le fond sonore.

Nous pensons donc qu'il est possible d'utiliser la librairie Recognito et l'application bureau en utilisant un très bon micro.

CONCLUSION

Ces deux semaines de projet tuteuré nous ont permis de nous renseigner sur une nouvelle technologie qui est la reconnaissance vocale. C'est une technologie encore naissante et il n'y a pas encore beaucoup de librairies à ce sujet en open source.

Toutefois la librairie Recognito est assez fiable. Cependant, il y a encore quelques contraintes qui sont que les enregistrements doivent se faire sans bruits, qu'il faut beaucoup de voix pour que Recognito soit fiable ce qui rend le temps d'exécution long. Hormis ces points, nous pensons que Recognito pourrait être fiable avec des très bons micros ou des micros de smartphones.

Actuellement, l'application est une application bureau non compatible avec Android. Toutefois nous pensons qu'il est possible de passer l'application sous Android mais que cela sera assez compliqué car il faudrait recoder une bonne partie de la librairie. De plus, si l'application met trente secondes à initialiser Recognito sur des ordinateurs performants, nous pensons que cela sera beaucoup plus long sur smartphone.

Enfin, nous sommes très satisfaits du travail que nous avons réalisé. A partir d'un sujet, nous avons pu donner deux solutions fiables pour répondre à cette problématique.

SITOGRAPHIE

Recognito : <https://github.com/amaurycrickx/recognito>

Alizé : <https://github.com/ALIZE-Speaker-Recognition/android-alize>

SHoUT : <http://shout-toolkit.sourceforge.net/index.html>

VeriSpeak : <http://www.neurotechnology.com/cgi-bin/biometric-components.cgi?ref=vs#matcher>

Api Google : <https://cloud.google.com/speech/?hl=fr>

Api Microsoft : <https://azure.microsoft.com/en-us/services/cognitive-services/speaker-recognition/>)