

Projet Tutoré 2016 - Groupe M

**Dossier d’analyse et conception**

**Réalisé par :**

Guillaume Robert

Yoann Gathignol

Titouan Bouete-Giraud

**Destinataire :**

Isabelle Clavel

**Superviseur :**

Laurence Redon

I Hear, I Say

Table des matières

[Introduction 3](#_Toc469606205)

[I. Organisation actuelle et future 4](#_Toc469606206)

[II. Analyse 5](#_Toc469606207)

[III. Conception 7](#_Toc469606208)

[IV. Spécificités techniques 7](#_Toc469606209)

[V. Plan d’assurance qualité 8](#_Toc469606210)

[VI. Organisation mise en œuvre dans l’étape DAC 8](#_Toc469606211)

[Conclusion 10](#_Toc469606212)

[Annexes 11](#_Toc469606213)

[I. Compte-rendu P-TUT 29/09/16 « I Hear, I Say » 11](#_Toc469606214)

[II. Compte-rendu 07/12/2016 « I Hear, I Say » 11](#_Toc469606215)

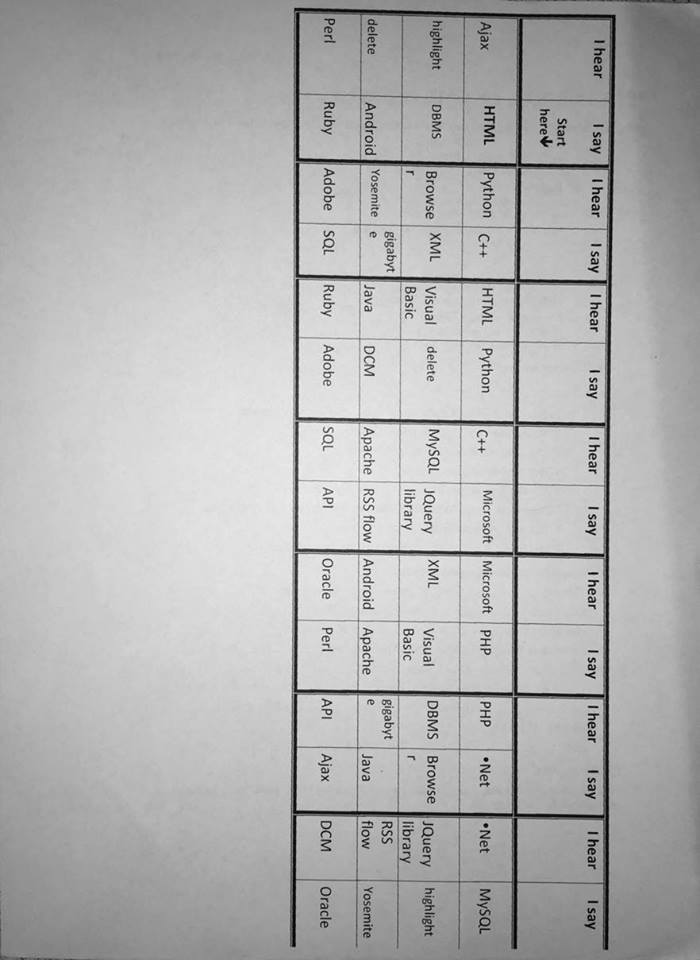
# Introduction

Ce projet tutoré est une demande d’Isabelle Clavel, professeure d’anglais au département Informatique de l’Institut Universitaire Technologique de Blagnac. Laurence Redon, également enseignante à l’Institut, supervise ce projet.

Dans le cadre de son enseignement, Mme Clavel utilise des grilles d’exercices de prononciation qu’elle crée elle-même à l’aide de logiciel de tableur tel que Microsoft Office Excel. Bien que très complet, ce type de logiciel n’a pas pour but premier ce type d’exercice, et peut donc présenter diverses contraintes pour l’utilisateur :

* De temps : l’absence de format standard force l’utilisateur à dédier une partie non négligeable de son temps à gérer la forme de la grille.
* De suivis : La modification de grille peut s’avérer très fastidieuse à cause des contraintes sur les données saisies, à tel point qu’il peut s’avérer préférable de commencer une nouvelle grille.

Le client a donc demandé une solution alternative qui lui permettrait un gain de temps non seulement à la réalisation, mais aussi lors de la modification éventuelle de ces grilles. Pour de plus amples informations, nous invitons le lecteur à consulter le Cahier Des Charges Utilisateurs (CDCU) fournis avec ce dossier.



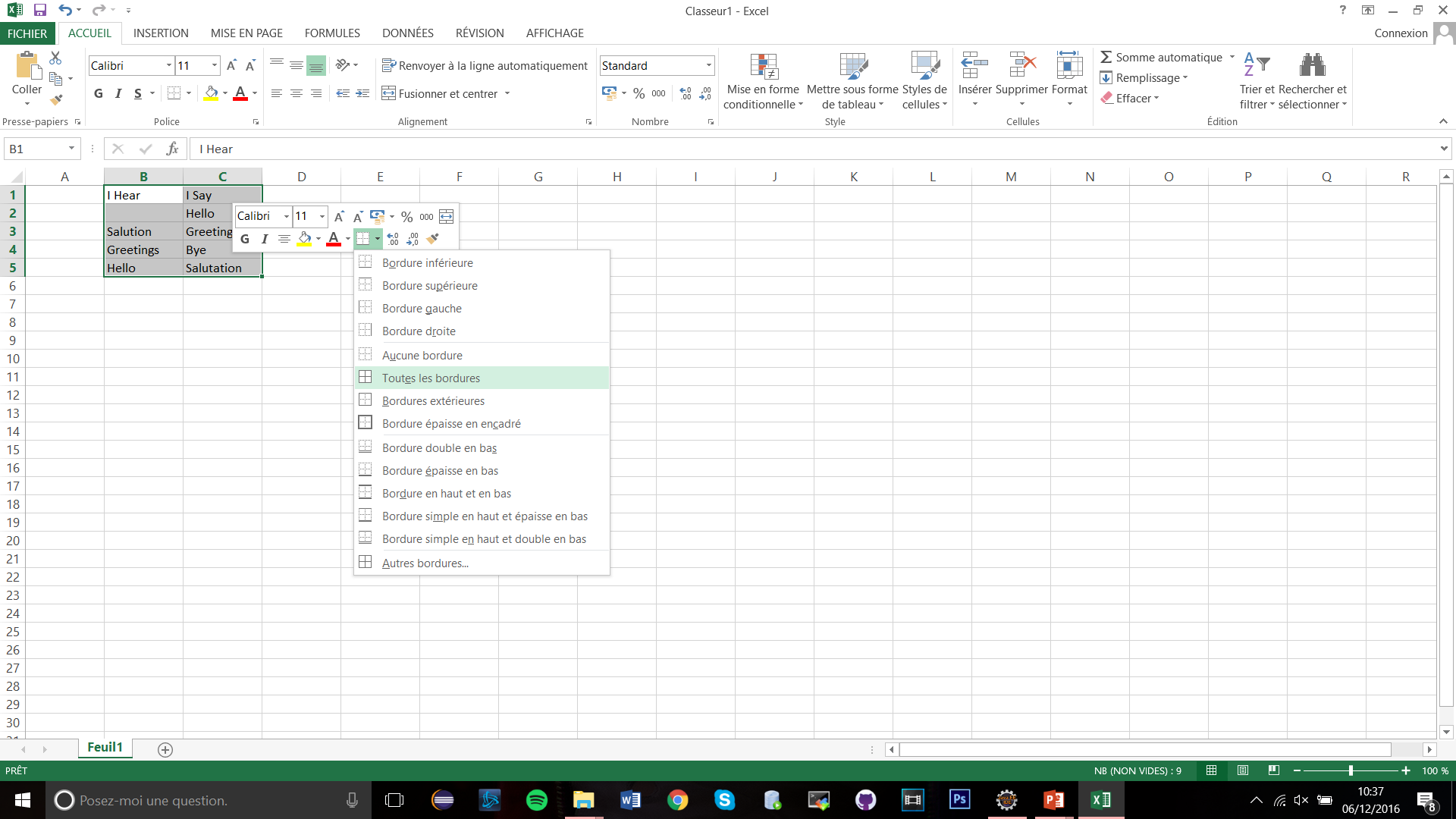
*Figure 1 : Exemple de résultat*

Sur cette figure on peut voir ce qui est attendu au final. Ce que nous devons permettre au client de générer grâce à notre application. En effet l’activité consiste à prononcer des mots de la colonne « I Say », celui qui a ce mot dans la colonne « I Hear » devra dire le mot correspondant et ainsi de suite. Cela forme une chaine jusqu’à ce que tous les mots aient été prononcés.

**Comment faciliter la création de cette activité grâce à un logiciel dédié à cette tâche ?**

# Organisation actuelle et future

Actuellement, pour réaliser son activité, notre cliente doit passer par plusieurs étapes plus ou moins longues. Tout d’abord elle doit faire de la mise en forme afin de créer un tableau contenant deux colonnes : « **I Hear** » et « **I** **Say** ». Ensuite, elle doit entrer les mots un par un, en double puisque chaque mot doit apparaître et dans la colonne I Hear et dans la colonne I Say. Cependant comme le but de l’exercice est de dire tous les mots présents dans la grille, les mots ne sont pas entrés dans l’ordre mais dans un ordre aléatoire. C’est pourquoi la création est prend beaucoup de temps et la modification est aussi très longue. Par exemple, pour générer cette petite capture d’écran il aura fallu presque 5 minutes juste pour remplir correctement les grilles.



*Figure 2 : Exemple de génération de grille*

Grâce à l’application que nous produisons, il y aura une automatisation de plusieurs étapes lors de la création des grilles.

La modification d’une case entraînera automatiquement la modification de celle qui lui est associée (si on modifie une case de la colonne « I Say » contenant le mot « Now » alors la case contenant ce mot dans la colonne « I Hear » sera modifié).

Lors de la création d’une grille, une option permettra de créer la grille automatiquement sans avoir à faire la mise en forme et en aillant juste à entrer les mots les uns à la suite des autres.

Les grilles sous format .csv seront générées directement grâce à l’application, c’est-à-dire que la mise en forme sera faite automatiquement.

Ces choix ont été faits dans le but de faciliter la production de cette activité par notre cliente.

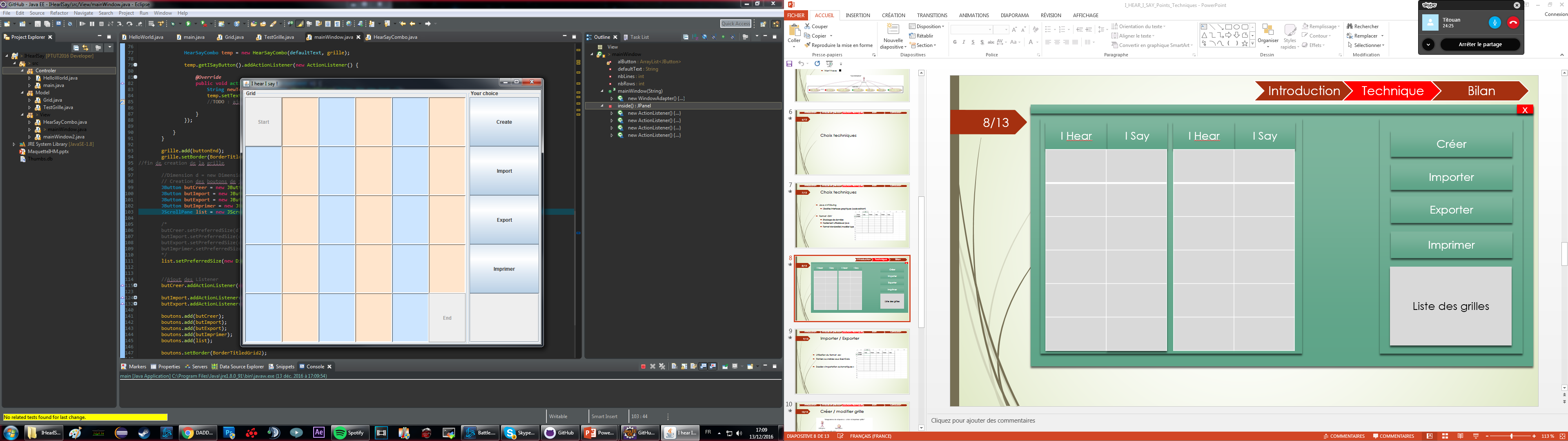
# Analyse

//TODO

**Jidbanana** pour UC/DSS/DC

****

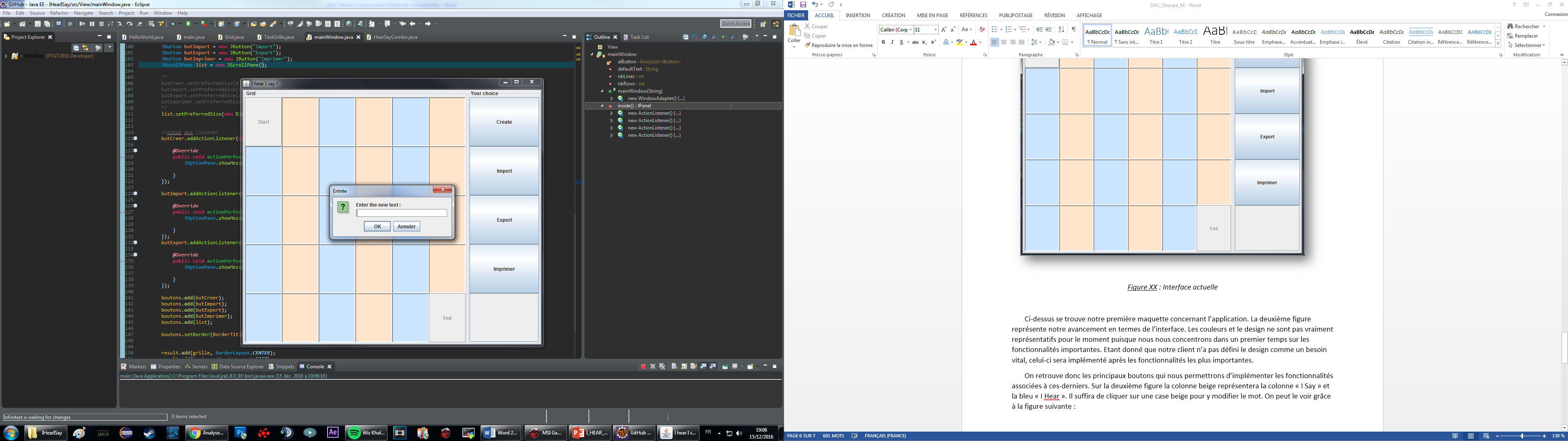
*Figure XX : Maquette*

**

*Figure XX : Interface actuelle*

Ci-dessus se trouve notre première maquette concernant l’application. La deuxième figure représente notre avancement en termes de l’interface. Les couleurs et le design ne sont pas vraiment représentatifs pour le moment puisque nous nous concentrons dans un premier temps sur les fonctionnalités importantes. Etant donné que notre client n’a pas défini le design comme un besoin vital, celui-ci sera implémenté après les fonctionnalités les plus importantes.

On retrouve donc les principaux boutons qui nous permettrons d’implémenter les fonctionnalités associées à ces-derniers. Sur la deuxième figure la colonne beige représentera la colonne « I Say » et la bleu « I Hear ». Il suffira de cliquer sur une case beige pour y modifier le mot. On peut le voir grâce à la figure suivante :



*Figure XX : Interface lors du clic sur une case*

# Conception

//TODO

# Spécificités techniques

Afin de mette en place, nous allons développer une application Java. Cela demandera donc l’installation du logiciel *Java* chez l’utilisateur de l’application. Ce logiciel peut se télécharger gratuitement à l’adresse suivante :

[**https://www.java.com/fr/**](https://www.java.com/fr/)

Dans un premier temps l’utilisateur nécessitera également un logiciel permettant de lire et d’imprimer des fichiers de format csv, tel que *Microsoft Office Exel*, ou *Libre Office Calc.* Dans un second temps, la fonctionnalité « imprimer » sera implémentée.

Pour réaliser cette application, nous utiliserons l’environnement *Eclipse* pour son aide très précieuse durant l’écriture du code. Aussi, nous utiliserons les librairies « java AWT/Swing» qui offre de nombreuses fonctionnalités pour créer une interface.

Le partage du travail au sein du groupe se fera grâce à *GitHub* qui permet une mise en commun des données, notamment du code. L’application offre également d’autres fonctionnalités très appréciables pour le développement, tel que le *versionning* des fichiers, ou encore différentes branches pour ranger les informations que nous partageons.

La communication au sein de l’équipe se fera grâce à *Facebook*/*Messenger*, ainsi que *Skype* qui offre une fonctionnalité de partage d’écran.

Enfin la réalisation de la documentation se fera grâce aux différents logiciels du pack *Microsoft Office*.

Nous utilisons PlantUML afin de générer nos diagrammes tels que « Diagramme de classe », « diagramme de séquence »…

# Plan d’assurance qualité

Pour s’assurer de la pertinence de cette application, nous avons choisis de mettre en place la méthode agile. Cette méthode vise à réaliser régulièrement des rendez-vous avec le client et le superviseur afin d’obtenir un maximum de retour sur le projet et son avancement. De cette manière, nous nous assurons que l’application que nous développons répond bien aux exigences du client.

Aussi, nous envisageons le fait de pouvoir partager notre application avec quelques professeurs et quelques élèves afin qu’ils puissent nous faire des retours constructifs nous permettant d’améliorer notre rendu.

Nous implémenterons régulièrement des tests afin de garantir le bon fonctionnement des fonctionnalités implémentées.

# Organisation mise en œuvre dans l’étape DAC

Concernant la communication durant l’étape DAC, elle a été réalisée par plusieurs moyens. Tout d’abord au sein de l’équipe, grâce à plusieurs outils à savoir, Messenger, Skype, GitHub, réunion à l’IUT. Toutes nos données ont été échangés via la plateforme GitHub, notre projet est en ligne afin qu’on puisse y accéder depuis n’importe quel PC.

Dans un second temps, la communication avec le client et le superviseur à majoritairement été faite par mail et mais aussi grâce à des réunions à l’IUT. Il y a eu plusieurs réunions :

* Jeudi 22 Septembre 2016 (client)
* Jeudi 29 Septembre 2016 (superviseur + client)
* Lundi 17 Octobre 2016 (superviseur)
* Mercredi 07 Décembre 2016 (client)
* Une autre réunion a eu lieu (une avec le client)

**Jidbanana 🡺** Gantt initiale + prévisionnel + difficultés rencontrés

# Conclusion

L’un des enjeux principaux de ce sujet est de faciliter la production d’une activité pour notre client. En proposant une application plus ciblée sur les besoins du client nous espérons permettre à celui-ci de gagner du temps. Nous avons donc décidé de créer une application en Java permettant de répondre à ses besoins de manière efficace. L’application doit être le plus ergonomique possible et doit permettre une modification des grilles d’une manière très simple.

# Annexes

## Compte-rendu P-TUT 29/09/16 « I Hear, I Say »

**Groupe M**

Personnes présentes : - Isabelle Clavel

- Laurence Redon

- Guillaume Robert

- Yoann Gathignol

- Titouan Bouëte-Giraud

Sujet de la réunion : Discussion à propos des fonctionnalités du projet

Lors de notre dernière réunion nous avons ajouté les besoins suivants :

* L’implémentation d’un historique qui garde une trace des anciennes grilles générées, contenant également le nom d’utilisateur, le groupe, la date et la thématique de la grille,
* Filtrer l’historique pour retrouver les grilles.

Avant le rendu final nous devons rendre une version finale du cahier des charges au professeur référent. Le projet pourra potentiellement se terminer en avance, c’est pourquoi il pourrait y avoir des modifications. Aussi, il faudra expliquer la structure de notre algorithme.

## Compte-rendu 07/12/2016 « I Hear, I Say »

**Groupe M**

Personnes présentes : - Isabelle Clavel

- Guillaume Robert

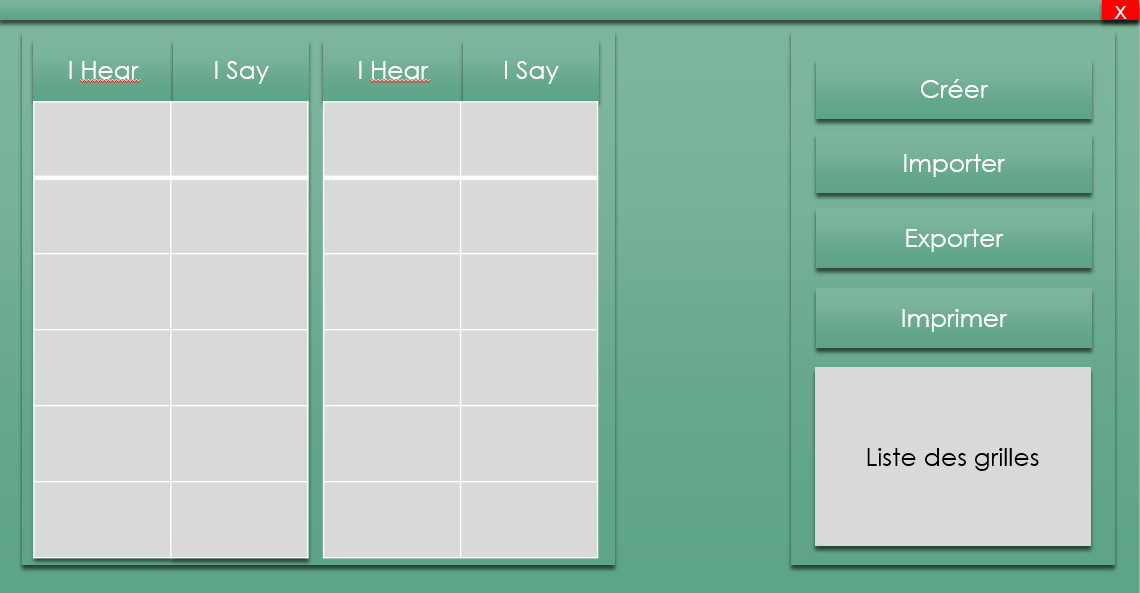
- Yoann Gathignol

- Titouan Bouëte-Giraud

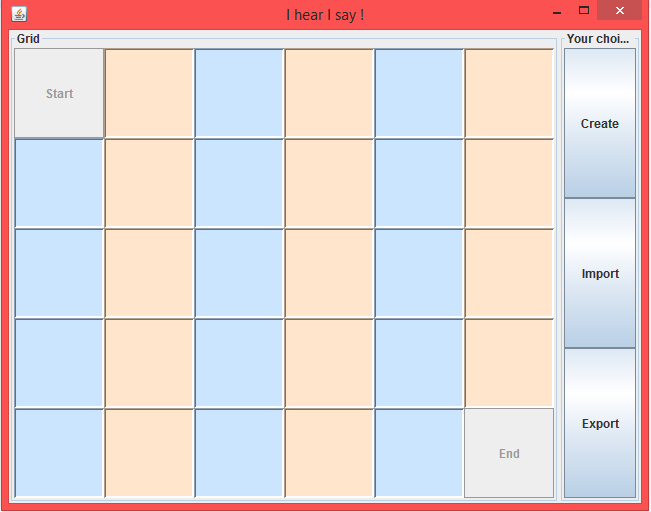
Sujet de la réunion : Présentation d’un premier prototype

Sujets évoqués :

* Présentation de la maquette



* Présentation de l’existant



Colonne bleu : I hear

Colonne beige : I say

Paire de boutons bleu/beige (même contenu)

* Un format type de csv sera fourni (ce format devra être respecté pour l’importation).
* Présente d’un dossier « Importation Automatiques » pour charger des grilles au lancement de l’application.
* Nous avons aussi discuté de la possibilité d’entrer tous les mots d’un coup sans avoir à cliquer sur les cases.