

Telecom Nancy
Université de Lorraine
Groupe 21

Flavien Ledoux
Guilhèm Roura
Céline Zhu

PROJET PPII



Sébastien Da Silva
Gérald Oster

Villers-lès-Nancy
Avril 2021

Introduction

*“We don’t just build websites, we
build websites that SELLS”*

— Christopher Dayagdag

Le concours Mines-Telecom, en raison du nombre de participants et d’épreuves, comporte de nombreuses données différentes. Cela nécessite donc des outils adaptés afin de pouvoir les vérifier, les traiter et les exploiter.

Afin de pouvoir stocker ces données, il faut de plus avoir une base de données adaptée, que l’on pourra remplir avec les données issues de ce concours.

Ainsi, il nous a été demandé pour ce projet de concevoir une base de données devant conserver l’ensemble des données du concours. Mais il a aussi été demandé de concevoir et réaliser une application permettant de remplir et d’exploiter cette base de données. Le remplissage doit pouvoir être fait à partir de fichiers Excel comportant des données issues du concours.

Le but précis de

cette application est, à partir des fichiers Excel, de remplir la base de données. De plus, l’application doit permettre de retrouver les données selon divers critères ou filtres.

Une certaine liberté est laissée sur la conception de l’application et de la base de données, la seule contrainte étant d’utiliser au moins un des 3 langages imposés (C, Python, Java) pour l’application.

Table des matières

1	État de l’art	4
1.1	Technologies utilisable pour le projet	4
2	Conception et mise en œuvre d’une base de données relationnelle	7
2.1	Normalisation	7
2.2	Choix de conception	8
3	Composants logiciels informatiques	10
3.1	Script	11
3.2	Traitement des fichiers	12
4	Statistiques	22
4.1	Généralités	22
4.2	Analyse descriptive	22
4.3	Modélisation et représentation des résultats statistiques	26
5	Conception et développement de l’application	27
5.1	Généralités	27
5.2	Partie Front-end de l’application	28
5.3	Partie Back-end de l’application	31
6	Gestion de Projet	33
6.1	Généralités	33
6.2	Outils de gestion de projet	34
6.3	Prendre connaissance du sujet	39
6.4	Établir les outils de GdP	39
6.5	Établir le Cahier des charges	39
6.6	Latex (?)	40
6.7	Planifier la semaine d’après	40
6.8	Organigramme projet	40

7 Conclusion	43
7.1 Conclusion personnelle	44
8 Annexe	49
8.1 Schéma de la base de données A	49

Chapitre 1

État de l’art

1.1 Technologies utilisable pour le projet

Contents

1.1	Technologies utilisable pour le projet	4
1.1.1	Présentation des SGBD (système de gestion de base de données) considérés	4
1.1.1.1	SQLite :	4
1.1.1.2	MySQL :	5
1.1.2	Présentation des frameworks considérés	5
1.1.2.1	Flask :	5
1.1.2.2	JavaFX :	5
1.1.2.3	Qt :	5
1.1.3	Conclusion	6

1.1.1 Présentation des SGBD (système de gestion de base de données) considérés

1.1.1.1 SQLite :

SQLite est un SGBD utilisant le langage SQL. La base de données consiste en un fichier dans lequel toutes les informations sont stockées. En plus de son incorporation trivial dans un projet, le SGBD ne demande pas beaucoup de mémoire. Enfin, tous les membres du groupe ont utilisé SQLite au cours des TPs de WeBD. [14] [10]

1.1.1.2 MySQL :

MySQL est un SGBD utilisant le langage SQL. La base de données est située sur un serveur. Pour consulter des informations, il faut être connecté en réseau avec le serveur dans lequel la base est hébergée. Cela demande donc la configuration d'un serveur qui doit héberger la base de données tout au long du projet ainsi que de pallier aux problèmes de serveur introuvable et de connexion non sécurisée. Enfin, tous les membres du groupe n'ont pas d'expérience avec MySQL. [13]

1.1.2 Présentation des frameworks considérés

1.1.2.1 Flask :

Flask est un framework permettant de développer des sites internet grâce à sa génération de pages HTML. Le framework est basé sur Python, langage que nous avons vu ou revu au cours du premier semestre. De plus, ce framework est facile à prendre en main car il possède moins de fonctionnalité que les frameworks standards. D'autre part, Flask intègre directement la gestion de SQLite. Enfin, Flask est utilisé dans le cadre du module WEBD, dans lequel nous avons pu nous initier à son utilisation. [4]

1.1.2.2 JavaFX :

JavaFX est un framework permettant de développer des interfaces graphiques pour des applications bureautiques. Le framework est basé sur Java, langage que nous avons vu ou revu au cours du deuxième semestre lors de nos cours de programmation orientée objet (OOP). JavaFX n'intègre pas de gestion de SGBD. Cette fonction est déléguée à d'autres APIs à installer en plus du framework. Enfin, tous les membres du groupe n'ont pas d'expérience avec ce framework. [12]

1.1.2.3 Qt :

Qt est un framework permettant de développer des interfaces graphiques pour des applications bureautiques. Le framework est basé sur du C++, langage que nous n'avons pas vu en cours mais qui ressemble à du C orienté objet. Le langage C a cependant été vu en cours. Qt intègre la gestion d'un grand nombre de SGBD. Cependant, en fonction du SGBD choisi, il se peut qu'il soit nécessaire de compiler Qt. Enfin, aucun membre du groupe n'a d'expérience avec ce framework. [1]

1.1.3 Conclusion

Nous avons décidé de choisir le MySQL comme SGBD car ce dernier fonctionne en local et qu'il ne demande pas la mise en place d'un serveur, que nous aurions dû héberger à nos frais.

Le framework Qt a été exclu d'emblée car nous n'avions pas assez d'expérience dans son utilisation, malgré sa grande variété d'outils pour se connecter à des SGBD. D'autre part, JavaFX a été aussi exclu en faveur de Flask car ce dernier n'intègre pas directement les outils pour se connecter au SGBD que nous avons choisi.

Ainsi, nous avons choisi d'utiliser pour le projet SQLite et Flask de part leur aisance d'utilisation quand ces derniers sont utilisés en conjonction.

Chapitre 2

Conception et mise en œuvre d'une base de données relationnelle

Contents

2.1	Normalisation	7
2.1.1	1FN	8
2.1.2	2FN	8
2.1.3	3FN	8
2.1.4	Autre pratique	8
2.2	Choix de conception	8

Le sujet nous demandait de créer une base de données sous forme normal 3 (3FN), puis de dé-normaliser si cela était voulu ou nécessaire. Dans un premier temps nous présenterons les 3 formes normales obligatoire du sujet. Ensuite nous expliqueront nos choix de conception.

2.1 Normalisation

Les formes normales des bases de données sont définies récursivement, c'est à dire qu'une forme plus évolué de normalisation dépend des formes plus basses. Ainsi la forme 3FN dépend de la forme 2FN qui dépend elle même de la forme 1FN, normalisation la plus basse des bases de données. [11]

2.1.1 1FN

: La forme 1FN correspond à l'atomicité des informations de la base de données. C'est à dire que chaque élément de la base contient au plus une donnée. Cela nous empêche donc d'avoir des listes d'éléments dans une entrée de la base de données

2.1.2 2FN

La forme 2FN correspond, en plus de la norme 1FN, au principe d'avoir chaque élément d'une table ne dépendant que de sa ou ses clés primaires. C'est-à-dire qu'un élément, pour être dans la même table que la clé primaire, doit uniquement dépendre de cette dernière ou d'éléments dépendant de cette dernière, et si elle dépend partiellement de cette dernière, alors une nouvelle table doit être créée avec une combinaison de plusieurs clés étrangères comme clé primaire.

2.1.3 3FN

La forme 3FN correspond, en plus de la norme 2FN, au principe de retirer la transitivité des dépendances au sein d'une même table. C'est-à-dire que si un élément A dépend de B et que B dépend C alors A ne peut pas être dans la même table que C. Nous devons donc avoir deux tables, une où A dépend de B et une autre où B dépend de C.

2.1.4 Autre pratique

Un principe courant dans les bases de données est de donner des codes à des données qui n'en possèdent pas dans le jeu sur lequel notre modèle se base. Cela permet d'économiser de la mémoire en évitant de stocker des informations redondantes en entière à l'aide de chaînes de caractères mais de leur assigner une valeur bien moins gourmande, généralement un entier, pour les représenter. Cependant cela a pour conséquence, dans certains cas, d'augmenter le temps des requêtes.

2.2 Choix de conception

Concrètement, nous avons centré notre schéma autour d'une table *candidat*, pour laquelle chaque entrée représente un candidat. Cette table comporte donc de nombreux attributs, puisqu'elle contient les informations sur les candidats fournis par le fichier *Inscription*. Nous y avons aussi ajouté des données

liés au candidats, mais obtenu ou extrapolé depuis d'autre fichier. La grande majorité des attributs reprennent des colonnes du fichier inscription. Cependant, afin d'éviter la redondance et d'être normale, si le fichier "Inscription" contenant un code et le libellé associé, alors la table "candidat" ne contient que le code. De même, pour des données répétées, tel que la qualité du candidat (Boursier/Pupille) ou encore un nom de pays ou de ville, la table contient alors un entier qui référence cette donnée.

Ces données répétées se trouvent dans des tables auxiliaires, contenant en clé primaire le code associé, et en autre attribut le libellé de la donnée.

Enfin, il y a des tables auxiliaires qui contiennent plusieurs données, par exemple celles pour les établissements.

Chapitre 3

Composants logiciels informatiques

Contents

3.1	Script	11
3.1.1	Script python	11
3.1.1.1	BDDCreation	11
3.1.1.2	ImportRepository	11
3.1.2	Autre Script	12
3.1.2.1	installer	12
3.2	Traitement des fichiers	12
3.2.1	Lecture des fichiers	12
3.2.1.1	Lecture de fichier tableur	14
3.2.2	Fonction d'insertion des données	14
3.2.2.1	Fonction Uploadnotes	14
3.2.2.2	Fonction UploadEcole	14
3.2.2.3	Fonction UploadOralEcrit	15
3.2.2.4	Fonction UploadAdm	15
3.2.2.5	Fonction UploadListVoeux	16
3.2.2.6	Fonction UploadListReponse	16
3.2.2.7	Fonction UploadSCEI	16
3.2.2.8	Fonction UploadClasse	16
3.2.2.9	Fonction UploadEtabli	17
3.2.2.10	Fonction UploadInscription	17
3.2.2.11	Fonction InsertData / InsertOrUpdateData	18
3.2.2.12	Fonction AddXXXX	19
3.2.3	Test de l'insertion	20

3.2.3.1	Test unitaire	20
3.2.3.2	Test de cohérence	21

3.1 Script

3.1.1 Script python

3.1.1.1 BDDCreation

Ce script permet de créer une base de données depuis à l'aide d'un fichier texte qui liste les différentes tables, le tout noté au format SQL.

La script prend deux arguments :

- **file_path** : le chemin vers le fichiers qui contient le code SQL
- **database_path** : le chemin où la base de données sera créé

Le script prend aussi en compte les commentaires SQL placé dans le fichier texte. Ces dernier doivent être marqué par la chaîne de caractère suivante : `'//'`.

Le script prend prend aussi en compte le cas ou le chemin du la futur base de données est déjà occupé par un autre fichier. Dans ce cas, le script laisse la main à l'utilisateur pour choisir s' il veut continuer le script et ainsi supprimer le fichier problématique.

Le script requiert l'extension python click pour fonctionner.

3.1.1.2 ImportRepository

Ce script permet d'insérer les données de tous les fichiers présents dans un répertoire vers une base de données. Le script prend deux arguments :

- **repository_path** : le chemin vers le répertoire où sont stocké les fichiers
- **database_path** : le chemin de la base de données qui va subir l'opération

Le script prend uniquement en compte les fichiers directement présents dans le répertoire, c'est-à-dire que les fichiers présents dans un répertoire situé dans le répertoire passé en paramètre seront ignorés.

Dans un premier temps, le script récupère le chemin de tous les fichiers. Une fois les fichiers récupérés, le script va chercher une chaîne de caractères clés (3.1 page 13) dans leur nom pour sélectionner la méthode de traitement approprié. Si un fichier ne contient aucune chaîne de caractères clé, il sera ignoré.

Enfin, le script exécute la méthode de traitement associée à chaque fichier retenu.

Comme le temps du script est proportionnel à la quantité de données à insérer dans la base de données, un indicateur est affiché pour permettre à l'utilisateur d'avoir une idée de la progression du script.

Le script requiert l'extension python click pour fonctionner.

3.1.2 Autre Script

3.1.2.1 installer

Le script Installer, disponible en .bat(Windows) et en .sh(linux) détecte si Python3 est installé sur l'ordinateur. Si Python3 n'est pas installé le script l'installe. Ensuite le script crée un environnement virtuel nommé ProjectEnvironnement où les dépenses du projet seront installées.

La version Linux fonctionne avec les distributions possédant les gestionnaires de package suivant : DPKG, DNF, et YUM.

3.2 Traitement des fichiers

3.2.1 Lecture des fichiers

La lecture des fichiers s'effectue à l'aide de la fonction ReadFile du fichier PolyMorph_Lecture. Cette fonction renvoie une liste dont le premier élément est le nom du fichier ouvert, et tous les éléments suivants étant les lignes du fichier. Les lignes sont modélisées par une liste où chaque case de la ligne correspond à une information trouvée sur la ligne.

La lecture des fichiers a été implémentée avec le polymorphisme en tête. Ce concept de programmation orienté objet consiste à créer une classe interface qui définit les entêtes fonctions pour être sûr que toutes les classes qui implémentent cette interface utilisent la même syntaxe. Ainsi, nous avons 3 classes définies :

- FileReading : la classe interface
- XLRS : la classe qui gère les fichiers .xlrs
- CSV : la classe qui gère les fichiers .csv

Cela permet au projet d'évoluer plus simplement au cours du temps si il faut ajouter une nouvelle extension à traiter ou de devoir placer les données de la base de données dans un fichier d'une extension gérée.

Dans un premier temps, la fonction vérifie si le fichier existe. Ensuite, elle va regarder l'extension du fichier pour connaître son type. Dans notre cas,

description	mot clé associé / type	exemple de fichier du jeu de test
liste des écoles	listeecoles	listeEcoles.xlsx
liste des établissement	listeetablissement	listeEtablissements.xlsx
liste des états possible d'un voeux	listeetatsreponsesappel	listeEtatsReponsesAppel.xlsx
liste des personnes inscrites	inscription	Inscription.xlsx
liste des informations de multiples candidats admissibles	admissible_	ADMISSIBLE_ATS.xlsx
liste des informations d'un candidat lié à sa classe	classes_	Classes_MP_CMT_spe_XXXX.xlsx
liste des informations d'un candidat lié à son code scei	scei	Classes_MP_CMT_spe_XXXX_SCEI.csv
liste des information d'un candidat et de son rang écrit	ecrit_	Ecrit_MP.xlsx
liste des information d'un candidat et de son rang oral	oral_	Oral_MP.xlsx
liste des informations de multiples candidats admis	admis_	ADMIS_ATS.xlsx
liste des voeux de multiples candidats	listevoeux_	listeVoeux_ATS.xlsx
liste des résultats des oraux CMT de multiples candidats	cmt_oraux	CMT_Oraux_YYYY_MP.xlsx
liste des résultats écrit de multiples candidats	resultatecrit_	ResultatEcrit_DD_MM_YYYY_ATS.xlsx
liste des résultats oraux de multiples candidats	resultatoral_	ResultatOral_DD_MM_YYYY_ATS.xlsx

TABLE 3.1 – *Tableau des chaînes de caractères clés*

seul deux types de fichier sont autorisés : .xlsx et .csv. La fonction va appeler la méthode read de la classe adéquate pour obtenir les données. Si cette méthode n'existe pas, alors la fonction s'arrête. Enfin, la fonction retourne la liste contenant le nom du fichiers et les données.

3.2.1.1 Lecture de fichier tableau

La lecture des fichiers se base sur 2 modules python :

- **openpyxl** : Pour la lecture des fichiers .xlsx
- **csv** (standard) : Pour la lecture des fichiers .csv

Ceci permet l'ouverture et la lecture de fichier .xlsx et .csv, ce qui couvre tous les formats de fichier attendus pour les fichiers de données.

3.2.2 Fonction d'insertion des données

3.2.2.1 Fonction Uploadnotes

Cette fonction prend deux arguments : les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile et d'une chaîne de caractères correspondant au type d'examen passé. La fonctions est a utiliser avec les fichiers de types "resultatoral_", "resultatecrit_" et "cmt_oraux"(3.1 page 13). La fonction a pour but d'ajouter des notes ainsi que d'autres informations relatifs aux notes tel que le nom du jury dans la table *notes* ainsi que dans que dans deux autres tables lié : *matiere* et *typeExam*.

La fonctions appel n-1 fois la fonction AddMatiere, avec n étant le nombre de données sur la première ligne, pour ajouter à la base de données les différentes matières et avoir leur code. La fonction AddTypeExam est aussi appelé pour ajouté un type d'examen et obtenir son code. Enfin, pour chaque donnée de chaque ligne, on associe dans la base de données, la note ou valeur à un candidats, une matière et un type d'examen en utilisant leur code.

3.2.2.2 Fonction UploadEcole

Cette fonction prend en unique argument les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile. Ce fichier doit être du type de "listeecoles" (3.1 page 13) La fonction a pour but d'ajouter de multiples écoles à la base de données dans la tables *ecole*. Pour chaque ligne du fichier, on ajoute le nom de l'école ainsi que son code dans la base de données.

3.2.2.3 Fonction UploadOralEcrit

Cette fonction prend deux arguments : les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile et d'une chaîne de caractères correspondant au type d'examen associé au rang, c'est à dire soit oral, soit écrit. La fonction est à utiliser avec les fichiers de types "oral_", "ecrit_" (3.1 page 13). La fonction a pour but d'ajouter des candidats ainsi que d'autres informations relatifs à ces derniers dans la table *candidats* ainsi que dans la table : *pays*, *commune*, *civilite*, *voie*, *matiere*, *typeExam* et *notes*.

Dans un premier temps, la fonction trouve la filière associée aux candidats à l'aide du nom du fichier de provenance. Pour chaque ligne nous appelons les fonctions AddCommune, AddCountry pour obtenir et enregistrer différentes informations liées aux candidats et en obtenir leur code pour y faire référence. Ensuite, nous ajoutons un nouveau candidat dans la table *candidat* ou nous le mettons à jour si il existe en utilisant les données de la ligne et des codes des communes, des pays et des civilités. Enfin, nous ajoutons son rang dans la table *notes* à l'aide du code du candidat, du type d'examen, et de la matière rang créée spécialement.

3.2.2.4 Fonction UploadAdm

Cette fonction prend deux arguments : les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile et d'une chaîne de caractères correspondant au type du candidat, c'est à dire soit admissible, soit admis. La fonction est à utiliser avec les fichiers de types "admis_", "admissible_" (3.1 page 13). La fonction a pour but d'ajouter des candidats ainsi que d'autres informations relatifs à ces derniers dans la table *candidats* ainsi que dans la table : *pays*, *commune*, *civilite*, *voie*, *resultat*.

Dans un premier temps, la fonction trouve la filière associée aux candidats à l'aide du nom du fichier de provenance. Ensuite, la fonction, à l'aide du nom du fichier dont proviennent les données, regarde si la liste des candidats est de type spécifique ou non et utilise la fonction AddResultat pour obtenir le code du résultat correspondant. Pour chaque ligne nous appelons les fonctions AddCommune, Addcountry, pour obtenir et enregistrer différentes informations liées aux candidats et en obtenir leur code pour y faire référence. Ensuite, nous ajoutons un nouveau candidat dans la table *candidat* ou nous le mettons à jour si il existe en utilisant les données de la ligne et des codes des communes, des pays, des résultats et des civilités.

3.2.2.5 Fonction UploadListVoeux

Cette fonction prend en unique argument les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile. Ce fichier doit être du type de "listevoeux_" (3.1 page 13) La fonction a pour but d'ajouter de multiples voeux provenant de multiples candidats à la base de données dans la tables *voeux_ecole*. Pour chaque ligne du fichier, on associe à un candidat, une école et un entier correspondant l'ordre du voeux un rang et le statuts du voeux .

3.2.2.6 Fonction UploadListReponse

Cette fonction prend en unique argument les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile. Ce fichier doit être du type de "listeetatsreponsesappel" (3.1 page 13) La fonction a pour but d'ajouter les possibles réponses à un voeux à la base de données dans la tables *reponse*. Pour chaque ligne du fichier, on associe à un code à ce à quoi il correspond.

3.2.2.7 Fonction UploadSCEI

Cette fonction prend en unique argument les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile. Ce fichier doit être du type de "scei" (3.1 page 13) La fonction a pour but d'ajouter d'associer a un candidat des informations grâce a son rang dans sa filière en modifiant les tables *ranginfo*, *notes*, *typeExam* et *matiere*. Pour chaque ligne du fichier, on associe à un rang et à une filière de multiples information. Ensuite, si un candidat de cette filière et avec ce rang existe, on ajoute le total de point qu'il a reçu lors de épreuve oral à l'aide des méthodes Add et du code candidat trouvé.

3.2.2.8 Fonction UploadClasse

Cette fonction prend deux arguments : les données d'un fichier obtenue préalablement à l'aide de la fonction ReadFile et d'une chaîne de caractères correspondant au type du candidat, c'est à dire soit admissible, soit admis. La fonctions est a utiliser avec les fichiers de types "admis_", "admissible_" (3.1 page 13). La fonction a pour but d'ajouter des candidats ainsi que d'autres informations relatifs a ces derniers dans la tables *candidats* ainsi que dans que dans les table : *pays*, *commune*, *civilite*, *voie*, *resultat*.

Dans un premier temps, la fonction trouve la filière associé aux candidats à l'aide du nom du fichier de provenance. Ensuite, la fonctions appel n fois la fonction AddMatiere, avec n étant le nombre de matières présentes sur la première ligne, pour ajouter à la base de données les différentes matières et

avoir leur code. La fonction `AddResultat` est aussi appelé pour obtenir les codes des résultats associable au candidats du fichier. De même on appelle la fonction `AddExamType` pour obtenir le code pour l'examen oral et l'examen écrit.

Dans un second temps, pour chaque ligne, nous appelons les fonctions `AddCommune`, `AddCountry`, `AddVoie`, pour obtenir enregistrer différentes informations lié au candidats et en obtenir leur code pour y faire référence. Ensuite, nous ajoutons un nouveau candidat dans la table *candidat* ou nous le mettons à jour si il existe en utilisant les données de la ligne et des codes des communes, des pays, des résultats et des civilités. Puis, nous ajoutons dans la table *notes* les notes du candidat associé à la ligne

3.2.2.9 Fonction UploadEtabli

Cette fonction permet d'ajouter les données un fichier de la forme de "listeEtablissement"

Elle prend pour unique paramètre le fichier sous forme de liste, obtenu avec la fonction `ReadFile`.

Pour chaque ligne du fichier excel, elle crée un dictionnaire correspondant à l'entrée que l'on veut ajouter dans la base de donnée. Pour cela, on fait une première requête basé sur le RNE de l'établissement, pour tester si l'établissement a déjà été ajouté. Il a en effet pu avoir été ajouté précédemment depuis le fichier "Inscription". Dans ce cas certains champs seront non renseigné.

Si l'établissement est déjà présent, on met à jour tous les champs avec la valeur venant de "listeEtablissement".

Autrement, on ajoute une nouvelle entrée correspondant à l'établissement.

3.2.2.10 Fonction UploadInscription

Cette fonction prend **en unique argument** un fichier sous forme d'une liste, obtenu par une des fonctions de lecture de fichier. Ce fichier doit **être du type** de *Inscription* ((3.1 page 13). Elle a pour but de remplir la grande majorité de attributs de la table *candidat*, ainsi que différentes tables auxiliaires, notamment celles liées directement à la table *candidat*.

Cette fonction va parcourir chaque ligne pour à chaque fois faire les insertion nécessaire suivant la ligne lu.

Pour une ligne, **elle appelle d'abord une dizaine de fonction *Add[...]***. Ces dernières permettent d'ajouter la donnée à une table auxiliaire, et renvoient l'identifiant associé pour la référencer. Si la donnée se trouvait déjà

dans la table, elle donne l'identifiant sans avoir fait d'insertion.

Ensuite, un dictionnaire correspondant à la table *candidat* est rempli. La clé correspond au nom de l'attribut, et la valeur correspond à la valeur associé. Ce dictionnaire comprends ainsi la quasi totalité des attributs de *candidat*.

Pour chaque valeur, il y a **trois possibilités** :

- la valeur du fichier est **inséré tel quel**. Cela correspond notamment aux valeurs entière ne nécessitant pas de traitement, où aux valeurs unique (ou presque) comme l'adresse
- **La valeur est transformé**, au moyen d'une **fonction de traitement**, puis insérez. Cela correspond en particulier aux numéro de téléphone, qui sont formatés autant que possible sous un même format
- La valeur rentrée est **l'identifiant référençant la vrai valeur** dans une autre table. C'est l'identifiant obtenu plus tôt avec les fonctions *Add[...]*

Enfin, à partir du dictionnaire, les données sont inséré ou mise à jour, suivant qu'un entrée pour ce numéro de candidat existe déjà ou pas. Cette insertion/mise à jour est faite au moyen de la fonction d'ajout *InsertOrUpdateData*

3.2.2.11 Fonction *InsertData* / *InsertOrUpdateData*

Ces 2 fonctions sont assez semblables, *InsertOrUpdateData* n'étant qu'une version augmentée de *InsertData*

InsertData

Cette fonction à pour but de généraliser l'ajout de données dans une table. Elle prend **4 paramètres** et **1 valeur de retour** :

- **param1 = data** : C'est le paramètre principal de la fonction, un dictionnaire contenant toutes les données qui seront éventuellement ajoutées
- **param2 = name_id** : C'est le nom de l'attribut qui sera renvoyé à la fin. Il s'agit de la valeur permettant de référencer l'entrée depuis une autre table
- **param3 = name_table** : C'est le nom de la table dans laquelle faire l'insertion
- **param4 = name_select** : C'est le nom de l'attribut qui sera utilisé pour vérifier si l'entrée existe déjà ou pas
- **retour** : La valeur associée à l'attribut *name_id*

Le paramètre *data* est un **dictionnaire** contenant l'entrée à ajouter. Chaque couple clé/valeur est un attribut à remplir dans la table. La *clé* est le nom de l'attribut dans la base de donnée, et la *valeur* est la valeur de l'attribut pour cette entrée.

La fonction va ouvrir la connexion avec la base de donnée et faire une première requête. Celle-ci essaye de récupérer la valeur de l'attribut *name_id* pour l'entrée dont la valeur de l'attribut *name_select* coïncide avec la valeur passé en paramètre. Si on a un résultat, l'entrée existe déjà, et on ferme la connexion. On renvoie alors la valeur de *name_id* obtenu.

Si la première requête ne donne rien, alors l'entrée n'existe pas encore. On va alors, avec le dictionnaire et le nom de la table, effectuer une requête qui va ajouter l'entrée. On effectue alors une 3ème requête, identique à la première, afin de récupérer la valeur de l'attribut *name_id*, qui sera alors la valeur renvoyée.

InsertOrUpdateData

Cette fonction a les mêmes paramètres et valeur de retour, et est quasi identique. Cependant, si la première requête nous renvoie un résultat, alors au lieu de ne rien faire on va insérer les données en remplaçant éventuellement celle qui était déjà présente. On ne fait cependant pas de 3ème requête et on renvoie la valeur obtenue dès le départ.

L'intérêt de cette deuxième fonction est d'éviter des requêtes et des changements de valeurs inutiles, dans le cas où on sait que si l'entrée existe déjà, alors il n'y a rien à rajouter.

3.2.2.12 Fonction AddXXXX

Les fonctions de la forme *Add[...]* sont toutes très courtes, puisqu'elles se basent principalement sur la fonction *InsertData*.

Elles sont chacune liées à une table, et permettent d'insérer un ou plusieurs attributs.

Paramètre(s) : le ou les attributs à insérer. Certains paramètres peuvent être optionnels.

Retourne : l'attribut qui permet de référencer l'entrée depuis une autre table.

Ces fonctions peuvent être appelées dans 2 cas :

1. On lit un fichier et on a obtenu une ou plusieurs données allant ensemble. On appelle alors la fonction pour les insérer si elles n'étaient pas déjà présentes. On peut éventuellement utiliser la valeur de retour pour référencer ces données depuis une autre table (en majorité depuis la table "candidat").

2. On veut uniquement la valeur permettant de référencer une donnée, qu'on sait être déjà présente.

Le principal cas où on utilise 2 paramètres est celui où l'on connaît un code et son libellé associé, par exemple pour le pays. Or, le code associé à un pays n'est présent que dans le fichier *Inscription*, mais il y a des pays dans d'autres fichiers.

Le paramètre du numéro de code est alors optionnel, pour 2 raisons. Tout d'abord, pour pouvoir ajouter un nouveau pays si on ne connaît pas le code. Le code sera éventuellement ajouté lors d'un appel ultérieur de la fonction. En deuxième, pour pouvoir obtenir la clé primaire auto-incrémentée associée au pays en ayant uniquement à spécifier le libellé, et ce avec la même fonction.

De manière générale, cette série de fonction se contente de créer un dictionnaire et d'appeler *InsertData* avec les bons paramètres, puis de renvoyer sa valeur de retour.

3.2.3 Test de l'insertion

3.2.3.1 Test unitaire

Pour tester l'exactitude de notre insertion de données, chaque fonction possède des tests unitaires. Les tests unitaires sont gérés avec la librairie *pytest*. Pour nos tests unitaires, nous avons essayé d'appliquer le concept RIGHT-BICEP :

- Right : notre test est-il bon ?
- B : Les limites sont-elles correctes ?
- i : La relation inverse est-elle possible ?
- C : Peut-on croiser nos résultats avec des fonctions développées par des tiers ?
- E : Peut-on forcer une erreur ?
- P : Les performances de la fonction sont-elles acceptables ?

Bien que certaines parties du concept ne peuvent s'appliquer dans notre cas, les principes des limites et des performances ont pu être testés sur nos fonctions.

Un test standard correspond à l'insertion dans la base de données de quelques lignes d'un fichier excel compatible à la fonction. Dans ce dernier, nous essayons d'aller aux limites de nos fonctions à l'aide de notre jeu de données. Pour vérifier que tout s'est passé comme prévu, nous effectuons un *SELECT* pour récupérer le jeu de données et le comparer à ce que nous aurions théoriquement dû trouver.

3.2.3.2 Test de cohérence

Une première choses à noter est qu'en SQLite, bien que l'on spécifie des types pour les attributs, ces types ne sont pas nécessairement contraignants. [10]

Il est donc pertinent de vérifier le type des valeurs réellement enregistrées. Par exemple que les attributs sensés contenir des entiers contiennent bien des entiers. Cela concerne notamment les attributs qui référence une autre table via un identifiant, mais aussi les valeurs supposé entière, comme certaines date, ou encore des départements.

Il faut ensuite vérifier autant que possible que les données contenues sont celles attendues. Par exemple, pour toutes les entrées, le champ "email" doit avoir le format d'un email. Ce genre de vérification peut notamment être effectuée au moyen de Regex. Dans python, cela se fait par le module standard *re*. Ce module offre des opérations sur les expressions régulières similaires aux regex Perl.[3]

Chapitre 4

Statistiques

Contents

4.1	Généralités	22
4.2	Analyse descriptive	22
4.2.1	Algorithmes et explication pédagogiques	23
4.2.1.1	Dispersion : Variance	23
4.2.1.2	Dispersion : Ecart-type	24
4.2.1.3	Médiane	25
4.2.1.4	Premier Quartile	25
4.3	Modélisation et représentation des résultats statistiques	26

4.1 Généralités

Dans le cadre de notre projet, nous allons effectuer quelques analyses statistiques avec l'ensemble des données fournies.

Dans cette perspective nous allons clarifier quelques termes nécessaires à la bonne compréhension de notre analyse.

4.2 Analyse descriptive

L'analyse descriptive est un sous-domaine des méthodes quantitatives (méthodes de recherches scientifiques utilisant l'analyse mathématique et scientifique). Dans la perspective d'appliquer cette science à notre base de données, nous allons par la suite nous inspirer de la statistique descriptive, afin de caractériser différentes catégories d'élèves

4.2.1 Algorithmes et explication pédagogiques

Afin de choisir les outils statistiques pertinentes aux données fournies, on s'est inspiré des données fournies par le site **www.scei-concours.fr**. Nous allons donc nous intéresser à l'implémentation de fonctions Python permettant le calcul de ces données ainsi que des fonctions complémentaires dont nous allons montrer l'algorithme ci-dessous.

Concours Mines - Télécom															
Ecole - Concours	Inscrits			Admissibles			Classés			Intégrés					Places
	Nb	Filles	5/2	Nb	Filles	5/2	Nb	Filles	5/2	Nb	Filles	5/2	Rg médian	Rg moyen	
IMT Mines Albi										49	34,7%	38,8%	2026	1972	53
IMT Mines Alès										58	25,9%	24,1%	1802	1731	64
ENSIIE Evry										78	17,9%	15,4%	2031	1981	72
ENSTA Bretagne (filère sous statut étudiant)										46	23,9%	17,4%	1257	1197	50
IMT Lille Douai										60	20,0%	31,7%	2171	2050	72
Mines Saint Etienne-ISMINE										32	21,9%	12,5%	1671	1624	38
Télécom Physique Strasbourg Informatique et Réseaux										20	35,0%	20,0%	2509	2351	20
Télécom Physique Strasbourg TI-Santé										7	71,4%	14,3%	2302	2050	8
TELECOM SudParis										85	20,0%	29,4%	1226	1171	86
TELECOM SudParis - cursus Sophia-Antipolis										9	22,2%	44,4%	1264	1207	4
Concours Mines-Télécom MP	5424	24,4%	21,1%				2986	22,2%	22,5%	444					467
ENSTA Bretagne (filère sous statut IETA - Direction Générale de l'Armement)	5424	24,4%	21,1%				2500	21,7%	25,3%	14	14,3%	57,1%	1120	1048	14
ENSG-Géomatique Marne la Vallée Civil										6	16,7%	16,7%	3096	3142	10
ENSG Nancy										7	57,1%	28,6%	2884	2886	5
ENSSAT Lannion										23	21,7%	30,4%	2902	2740	25
TELECOM Nancy										55	20,0%	21,8%	2605	2523	55
TELECOM St Etienne - Statut étudiant										27	22,2%	33,3%	2788	2661	30
TELECOM St Etienne - Statut apprenti										2	0,0%	50,0%	2768	2768	2
Concours Mines-Télécom Spécifique MP	5424	24,4%	21,1%				4067	23,7%	21,5%	120					127

Figure1 : Tableaux des statistiques 2020 sur scei (ici filière MP)

4.2.1.1 Dispersion : Variance

La dispersion représente la variabilité des différentes valeurs que peut prendre une variable. En statistiques, il existe différentes mesures de la dispersion. Les plus courantes sont la variance, l'écart-type ou encore l'intervalle inter-quartile. C'est une mesure peu influencée par la présence de valeurs extrêmes.

La variance est la moyenne des carrés des écarts à la moyenne :

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$$

Il s'agit alors d'appliquer l'algorithme ci-dessous :

Algorithm 1: Variance

Result: Variance

$l = \text{list}();$

for $i=0$ jusqu'à la longueur de la liste l avec un pas de 1 **do**

 compteur=0;

for $j=0$ jusqu'à la longueur de la liste l avec un pas de 1 **do**

if $l(i)(j)=n$ **then**

 compteur=compteur+1;

end

end

$l = l + \text{compteur};$

end

compteur=0;

for $i=0$ jusqu'à la longueur de la liste l avec un pas de 1 **do**

 compteur=compteur+l(i);

end

moyenne=compteur/longueur(l);

compteur=0;

for $i=0$ jusqu'à la longueur de la liste l avec un pas de 1 **do**

 compteur=compteur+(l(i) - moyenne)²

end

Variance=compteur/longueur(l);

4.2.1.2 Dispersion : Ecart-type

L'Ecart-type est la moyenne quadratique des écarts par rapport à la moyenne ou plus simplement la racine carrée de la Variance :

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$$

En admettant la viabilité de l'algorithme précédent, on définit une unique instruction : l'attribution de la racine carrée de la valeur absolue du résultat obtenu par l'algorithme précédent.

Algorithm 2: Ecart-type

Result: Et

Et= $\sqrt{|Variance|}$;;

4.2.1.3 Médiane

La médiane d'un ensemble de valeurs (échantillon, population, distribution de probabilités) est une valeur m qui permet de couper l'ensemble des valeurs triées en deux parties égales : mettant d'un côté une moitié des valeurs, qui sont toutes inférieures ou égales à m et de l'autre côté l'autre moitié des valeurs, qui sont toutes supérieures ou égales à m (s'il y a un nombre pair de valeurs, la médiane sera la moyenne des 2 valeurs "centrales" de la distribution) :

Si l'effectif total de la série est impair : $N = 2p + 1$, la médiane est la $(p + 1)^{\text{ème}}$ valeur.

Si l'effectif est pair : $N = 2p$, on prend en général pour médiane la moyenne de la $p^{\text{ème}}$ et de la $(p + 1)^{\text{ème}}$ valeur.

Il s'agit alors d'appliquer l'algorithme ci-dessous :

Algorithm 3: Mediane

Result: Mediane

Med=list();

for $i=0$ jusqu'à la longueur de la liste l avec un pas de 1 **do**

\lfloor med=med+liste(i);

end

$N=\text{entier}((\text{longueur}(\text{med})+1/2)$;

med=sorted(med);

mediane=0 **if** $\text{longueur}(\text{Mediane}) \equiv 0 \pmod{2}$ **then**

 mediane=med(longueur(med)+1/2)+longueur((med)-1/2)/2;

else

\lfloor mediane=med(N);

end

4.2.1.4 Premier Quartile

Les quartiles sont les 3 valeurs qui divisent les données triées en 4 parts égales, de sorte que chaque partie représente 1/4 de l'échantillon de la population. Il existe donc trois quartiles : Q1, Q2 (égal à la médiane) et Q3. Par exemple, Q1 est la valeur telle que 25 % des valeurs de l'échantillon lui sont inférieures, 75 % supérieures.

Algorithm 4: Premier Quartile

Result: Premier Quartile

premierquart=list();

for $i=0$ jusqu'à la longueur de la liste avec un pas de 1 **do**

 premierquart=premierquart+liste(i);

end

N=entier((longueur(premierquart))/4)

premierquart=sorted(premierquart);

Premier Quartile=0

if longueur(premierquart) $\equiv 0 \pmod{4}$ **then**

 Premier Quartile=premierquart(longueur(premierquart)/4-1);

else

 Premier Quartile=premierquart(N);

end

4.3 Modélisation et représentation des résultats statistiques

Ils existent plusieurs façons de modéliser et d'exprimer les résultats des statistiques qui nous intéressent. Que ça soit des données pures organisées sous forme de tableau ou encore des cartes interactives. Nous avons décidé de garder le format scvi qui semble être optimale, pour le service rendu.

Chapitre 5

Conception et développement de l'application

Contents

5.1	Généralités	27
5.2	Partie Front-end de l'application	28
5.2.1	Présentation de l'interface utilisateur	28
5.2.1.1	Interface Élève	29
5.2.1.2	Interface Professeur	29
5.2.1.3	Interface Administrateur	29
5.2.1.4	Interface Curieux	29
5.2.2	Architecture du Front-end	30
5.2.2.1	Automatisation avec un script SH	30
5.2.3	Interface Graphique Générale	30
5.2.3.1	Bulma et Css	31
5.3	Partie Back-end de l'application	31
5.3.1	Architecture du Back-end	31

5.1 Généralités

Le développement de l'application s'est fait principalement à partir du langage Python et du framework Flask que nous avons rencontré durant cette année scolaire dans le cadre de la WebBD et du TAPS. Le module développé présente donc deux parties :

- La partie Front-end, c'est à dire la partie client qui permet au client d'appliquer des fonctionnalités CRUD à la base de données à partir de méthodes FORM

- La partie Back-end, c'est à dire la partie serveur qui gère ces modifications dans la base de données

Un schéma résumant l'ensemble des interactions serveur/client sera fourni à la fin de ce chapitre.

On utilise dans cette partie un nombre de librairies Python pour parvenir à nos fins :

- Flask qui permet d'utiliser le framework Flask
- pdfkit qui permet de convertir des urls en pdf
- sqlite3 qui permet d'accéder et de modifier la base de données en utilisant le langage SQL

5.2 Partie Front-end de l'application

La problématique du sujet nous demandais de créer une application permettant à l'utilisateur d'effectuer des requêtes (SQL ou autres) qui interagissent avec la base de données et d'en conserver le résultat sous forme HTML ou PDF.

5.2.1 Présentation de l'interface utilisateur

Les requêtes des utilisateurs varient grandement selon l'identité de l'utilisateur, en effet, on peut classer les profils utilisateurs dans quatre grandes catégories compte tenue de la nature des données :

- Élèves qui seront intéressés uniquement par leur profil candidat
- Professeurs qui seront intéressés par les étudiants de leur établissement
- Administrateurs qui savent utiliser le SQL et veulent uploader des fichiers .xlsx ou .csv pour mettre à jour les données
- Curieux qui veulent tout simplement connaître des informations plus générales sur l'ensemble du concours de l'institut Mines-Télécom.

L'objectif ici étant de permettre à tout le monde d'accéder à l'ensemble des données qui leur sont pertinentes le plus rapidement possible. De cette manière, un utilisateur non-identifié doit alors déclarer son identité sur la page principale de l'application. Une fois identifié il se retrouvera sur une page de connexion (à l'exception du profil "Curieux"), qui permettra à l'application de confirmer son identité en comparant les informations fournies avec celles données par la partie serveur de l'application.

FIGURE

5.2.1.1 Interface Élève

L'utilisateur "Élève" qui est identifié par son code candidat (unique) et son nom. Une fois, la connexion établie, il arrivera à sa page personnelle où il aura accès à plusieurs information :

- Les informations personnelles le concernant tirés de la table candidat, modifiable
- Les notes obtenues aux écrits et aux oraux (si ces derniers ont eu lieux)
- Les vœux des candidats classés dans l'ordre, modifiable

5.2.1.2 Interface Professeur

L'utilisateur "Professeur" est identifié par le nom de son établissement ainsi le code de son établissement "unique", il a donc accès à toutes les notes des étudiants de son établissement avec les noms et prénoms des candidats.

Logiquement les professeurs ne sont pas sensés avoir accès aux informations personnelles des étudiants

5.2.1.3 Interface Administrateur

L'utilisateur " Administrateur" est identifié par un nom et un mot de passe arbitraire qu'on a décidé pour le projet :

- nom : "Groupe21"
- mot de passe : "L3 M31LL3UR GR0UP3"¹

Une fois la connexion établie, l'administrateur aura le choix entre trois options :

- Une recherche SQL avec le schéma de la base de données en référence
- Une barre de recherche dynamique
- Une option pour ajouter et supprimer les fichiers .xls et .csv et en conséquence modifier la base de données

5.2.1.4 Interface Curieux

Il s'agit de la visualisation des résultats des fonctions statistiques établies dans le chapitre 4. Une première visualisation des données générale sera fournie à l'utilisateur, puis des propositions plus précises seront offertes à l'utilisateur selon différents critères.

1. Les développeurs de ce projet sont conscients que le mot de passe n'est pas très sécurisé

5.2.2 Architecture du Front-end

Cette partie est développée en Python et HTML dans le framework Flask, l'installation de nouveaux packages se fait au travers d'un environnement Python 3.0 virtuel et de la commande pip3. l'architecture du Front-end est détaillée dans la figure suivante de manière simplifiée :

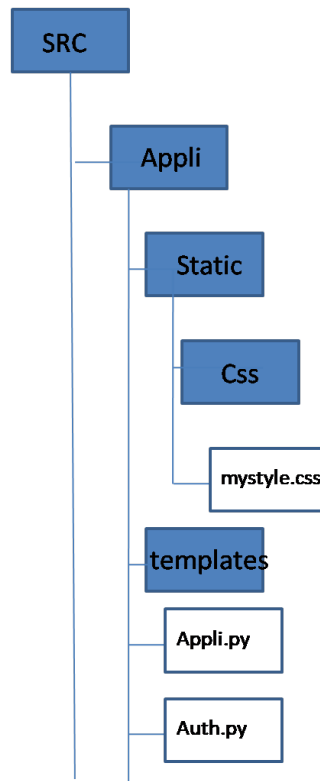


Figure :Architecture du Front-end

Les fichiers `auth.py` et `appli.py` quiinstancient les formulaires contenant leur paramètre d'entrée ainsi que les templates HTML associés.

5.2.2.1 Automatisation avec un script SH

`launchapp.sh` permet de lancer l'application dans l'environnement virtuel de manière simplifiée afin de maximiser l'expérience de l'utilisateur.

5.2.3 Interface Graphique Générale

Il tient à préciser que le design graphique de notre application a pour inspiration le site officiel de l'Institut Mines-Télécom. La bar de navigation

est commune à toutes les pages du sites et possèdent un bouton download qui à l'aide de URL obtenue par une fonction Javascript (request.url)

5.2.3.1 Bulma et Css

Pour le design de notre site Web, nous avons choisi le Flat Design plutôt que le Skeuomorphisme, qui permet de supprimer toutes les informations non-essentiels à l'utilisateur. Le choix de Bulma s'est donc imposé comme une évidence non seulement grâce à son utilisation pratique mais aussi grâce au design de ses assets, connus pour son flat design. Ainsi nous avons adapté à nos besoins les divers assets du sites ainsi pris pour inspiration en plus du color scheme et de la forme triangulaire.



Figure : Icône de l'IMT modifiée pour signifier la notification

De la même manière, nous avons implémenté un carrousel pour la page d'accueil à partir du fichier mystyle.css dans lequel sont implémentés un At-Rules keyframes modifiés ainsi qu'un At-Rules media adapté.

5.3 Partie Back-end de l'application

Il s'agit principalement des fonctions de création de base de données que l'on a présenté dans le chapitre précédent ainsi que l'installation de l'ensemble des dépendances de l'application. Nous avons également créé un script SH config.sh pour installer et configurer l'environnement du Back-end.

5.3.1 Architecture du Back-end

L'architecture est donc détaillée dans la figure suivante de manière simplifiée :

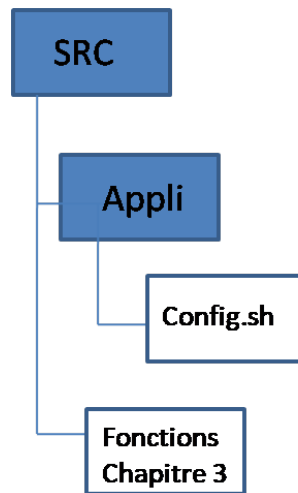


Figure :Architecture du Back-end

Le Back-end utilise alors le principe de microservices, c'est à dire qu'ils utilisent plein de petits services différents plutôt qu'un code monolithique. Il y a plusieurs avantages à utiliser des microservices : mais celle qui ressort le plus c'est l'indépendance. En effet ces différents services sont testés et utilisés de façon indépendantes.

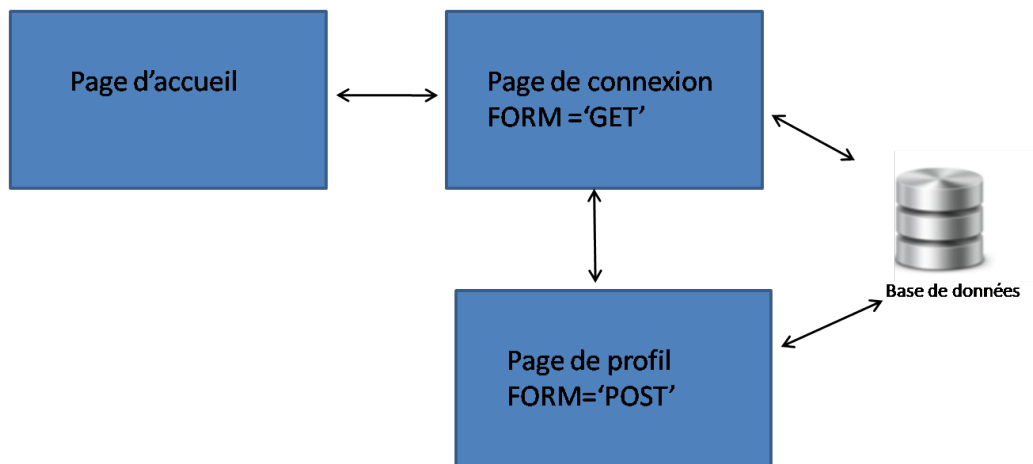


Figure :Architecture du site

Chapitre 6

Gestion de Projet

Contents

6.1	Généralités	33
6.2	Outils de gestion de projet	34
6.2.1	Charte de projet	34
6.2.2	Matrice SWOT	37
6.2.3	Matrice RACI	37
6.2.4	Méthode Agile	37
6.2.5	Comptes-rendus	38
6.3	Prendre connaissance du sujet	39
6.4	Établir les outils de GdP	39
6.5	Établir le Cahier des charges	39
6.6	Latex (?)	40
6.7	Planifier la semaine d'après	40
6.8	Organigramme projet	40

6.1 Généralités

Ce projet se développe dans le cadre du module du P2II, il vise la mobilisation des connaissances acquises durant les différents modules de l'année scolaire, y compris la Gestion de Projet.

Les algorithmes et leur étude développés lors de ce projet doivent répondre à l'énoncé fourni par Sébastien Da Silva et Gérald Oster. Tous les élèves participant au projet doivent travailler dans le déroulement de chacune de ses parties, autant comme développeurs que managers. Le projet

s'est déroulé suivant le principe de la méthode Agile.

6.2 Outils de gestion de projet

Le bon outil pour le bon usage.

Pour bien gérer notre projet et pour respecter les consignes de l'énoncé, nous nous sommes appuyés sur plusieurs outils de gestion de projet :

- Charte de projet
- Matrice SWOT
- Matrice RACI
- Comptes-rendus rédigés lors de chaque réunion
- Méthode Agile

6.2.1 Charte de projet

- **Cadrage/Finalités et importance du projet**

- **Objectifs et résultats opérationnels**

Tous les élèves participant au projet doivent maîtriser les différentes parties de l'application de tel sorte qu'ils soient capables de répondre à des questions sur ces dernières. Le projet se déroule par jalons définis à partir des sous-parties de l'énoncé.

Le projet vise à livrer une application, incluant une base de données et une méthode d'insertion de données depuis des fichiers excel et csv, qui sera notée. Le projet cherche à répondre à la totalité des questions de son énoncé. On s'intéresse à la création d'une application qui permet l'utilisation et la modification d'une base de données créée à partir de fichiers fournis. Le développement et l'étude des fonctions algorithmiques, l'application résultante ainsi que la gestion du projet seront conçus par tous les participants et le compte-rendu final comportera un document rédigé sur Latex ainsi qu'une présentation orale accompagnée d'une démonstration. À l'issue du compte-rendu, le module sera

soumis à une validation.

— **Déroulement du projet/Organisation/ressources**

— Dans le déroulement de ce projet, les participants sont :


LEDEUX Flavien
ROURA Guilhèm
ZHU Céline

Chacun des membres du groupe ont le même rôle en tant que développeur et manager.

— Les moyens à disposition sont multiples :

- Logiciels de traitement de texte tel Latex
- Logiciels permettant travail en équipe tel Gitlab
- Connaissances acquises lors de l'année scolaire 2020-2021 à TELECOM Nancy pour le développement et l'étude d'algorithmes
- Connaissances acquises lors du module de MOOC-GdP pour la gestion de projet
- Éventuelle aide des professeurs de TÉLÉCOM Nancy

— **Jalons : échéancier / événements importants**

<u>Jalon</u>	Description	Date
Étape 1 : Exigences opérationnelles	Validation des spécifications : cahier des charges techniques.	17/04
Étape 2 : Préparation pour démarrer le projet	Réalisation de la matrice SWOT, charte de Projet	24/04 
Étape 3 : Développement du livrable	Récupération des données	08/05
Étape 4 : Développement du livrable	Création et étude des fonctions nécessaires à la BDD de même que leur rédaction dans le compte rendu.	15/05
Étape 5 : Développement du livrable	Création de scripts d'installation et de lancement de l'application	15/05
Étape 6 : Développement du livrable	Création et étude de l'application	14/06
Étape 7 : Compte rendu	Valider compte rendu final et présentation orale.	15/06

— Risques et opportunités

Après analyse du déroulement du projet, on dégage :

Plusieurs risques tels :

- Un potentiel mauvais accès à internet
- Une absence de date butoir
- Le sujet n'est pas entièrement défini

Plusieurs opportunités telles :

- La possibilité de solliciter une personne extérieure au projet pour s'entraîner à la présentation orale
- La possibilité de demander de l'aide à un professeur en cas de blocage
- Accès à l'école et à son matériel
- Beaucoup de documentations en ligne

6.2.2 Matrice SWOT

	INCIDENCE POSITIVE	INCIDENCE NÉGATIVE
ORIGINE INTERNE	<ul style="list-style-type: none">- Les membres de l'équipe savent programmer en python et expérience Flask par le module WebBD- Les membres de l'équipe sont motivés et travailleurs- Les membres de l'équipe ont des parcours scolaires différents et complémentaires- accès à des machines installé uniquement pour tester l'application	<ul style="list-style-type: none">- La majorité des membres de l'équipe n'avaient jamais crée d'application avec front et back complète- Les membres de l'équipe ne se connaissait pas et n'avait pas travaillé ensemble auparavant
ORIGINE EXTERNE	<ul style="list-style-type: none">- La possibilité de solliciter une personne extérieure au projet pour s'entraîner à la présentation orale- La possibilité de demander de l'aide à un professeur en cas de blocage- Des horaires flexibles	<ul style="list-style-type: none">- Un possible mauvais accès à internet- Une absence de date butoire- Le sujet n'est pas entièrement définie

6.2.3 Matrice RACI

Nous avons réalisé une matrice RACI dans le but de formaliser le rôle de chacun lors des différents lots de travail et avoir une idée claire de nos responsabilités.

La matrice RACI est illustrée par la figure ci-dessus.

6.2.4 Méthode Agile

En ingénierie informatique, la méthode Agile se base sur un cercle vertueux d'auto-discipline centrée sur la confiance en autrui et sur un feedback régulier. Nous avons décidé d'utiliser cette méthode plutôt que la méthode linéaire traditionnelle, car la méthode Agile permet des réponses très flexibles aux changements et aux imprévus. Étant donné la liberté donnée par le sujet, et le problème ouvert, nous avons donc décidé de ne pas s'embarquer dans une méthode trop rigide (Diagramme de Gantt), dans la situation où une grande modification du sujet ait lieu. Néanmoins, afin de se fixer une struc-

ture dans notre organisation, nous avons tout de même imposé des réunions dits de "vérification d'avancement". Ce sont des réunions non de travail, mais d'observation de l'avancement de l'équipe ainsi que de partage des difficultés rencontrées. Ces réunions ont donc été fixées à moins d'une heure et cette contrainte a été bien respectée à l'exception de deux réunions. Elles ont été plus focalisées sur la création de documents de gestion de projet.

6.2.5 Comptes-rendus

À l'issue de chaque réunion, un compte-rendu était rédigé pour résumer les propos traités, formaliser les décisions prises et la liste des responsabilités désignées à chaque membre de l'équipe pour la prochaine réunion. Sa rédaction était prise en charge par chacun d'entre nous de manière cyclique.

Chaque compte-rendu a été rédigé d'après le modèle donnée ci-dessous, la totalité des compte-rendus peuvent être retrouvés dans l'Annexe et sur le dépôt GitLab :

Motif de la réunion : Réunion de démarrage du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : Flavien LEDEUX Guilhèm ROURA Céline ZHU Absent : Personne Rédaction du compte-rendu : Céline ZHU	Date : 03/04/2021 Heure : 10h01 Durée : 45 minutes

Contents

6.1	Généralités	33
6.2	Outils de gestion de projet	34
6.2.1	Charte de projet	34
6.2.2	Matrice SWOT	37
6.2.3	Matrice RACI	37
6.2.4	Méthode Agile	37
6.2.5	Comptes-rendus	38
6.3	Prendre connaissance du sujet	39

6.4	Établir les outils de GdP	39
6.5	Établir le Cahier des charges	39
6.6	Latex (?)	40
6.7	Planifier la semaine d'après	40
6.8	Organigramme projet	40

6.3 Prendre connaissance du sujet

Rapport sur la Gdp et la réflexion
 Objets à rendre :
 Base de données SQL standard + schéma
 Rapports formats html (rédigé en fonction de quoi à demander à M Da Silva ou M Oster)
 Afficher les résultats des requêtes que ce soit en schéma ou par texte
 L'état des fichiers final ou pas.

6.4 Établir les outils de GdP

Fichier Readme.md
 Matrice RACI
 Gantt
 SWOT

6.5 Établir le Cahier des charges

Base de données SQL :
 script Python rendant une base de données cohérente
 Avec des informations
 Trois étapes : Design, construction, tests boîte noire/blanche
 - Application Python (html) :
 - Tous Flask
 - Définir tous les liens qui sont liés à notre application
 - Définir les conventions Html
 - Fiches de style

6.6 Latex (?)

Création du latex sur Overleaf et séparation en plusieurs fichiers Tex plus tard.

Programmation

Flavien : Python(flask) et Sqlite sont plus pratiques pour des amateurs. et on peut s'aider de la WebBD.

6.7 Planifier la semaine d'après

- Faire les scripts de la base de données pour les créer et le schéma.
- Schéma : Flavien/Guilhèm
- Scripts : Céline
- Tests : Flavien/Céline
- Commencer à réaliser les classes représentant les différents éléments à ajouter dans la BDD.
- Documenter vos actions en latex.

Prochaine réunion :

Samedi à 10h01 10/04/21 :

odj :

- Avancement du projet
- Construction de la matrice RACI
- Faire le Gantt
- Dire les endroits qui nous ont posés problèmes

6.8 Organigramme projet

Selon leur localisation dans l'organigramme, il existe 3 types de projets :

- Au sein d'un même service : c'est un *projet local*.
- Impliquant plusieurs services différents : des *projets transversaux*.
- *projets sortis* : de taille importante et font appel à des intervenants détachés spécialement.

Notre projet est clairement un projet transversal au sein d'une structure fonctionnelle.

En effet, la structure fonctionnelle est une structure où il n'y a pas de chef de projet, ce qui encourage l'initiative, favorise l'adaptation aux évolutions de

l'environnement et facilite la circulation des informations. De plus, la division du travail se fait par fonctions ce qui favorise l'efficacité du travail.

Cette structure a permis à chacun d'entre nous de décider quelles tâches leur a été confiées :

— **Flavien Ledoux**

- réalisation des script de création de la base de données (BDDCreation.py)
- réalisation du script d'upload des données vers la base de données depuis un répertoire (ImportRepository.py)
- réalisation des script d'installation automatique installer (Windows et Linux)
- mise en place du polymorphisme sur les fonctions de lecture de fichier et création d'une fonction l'utilisant pour retourner les informations du fichier
- création des tables notes, voeux_ecole et ranginfo ainsi que les tables qui dépendent de ces dernières
- mise a jour et correction du schéma de la base de données
- Réalisation des fonctions Add sous leur forme initial
- réalisation des fonctions qui insèrent des informations vers la base de données sauf pour les fichiers inscription et listeEtablissement (3.1 page 13)

— **Guilhèm Roura**

- Réalisation d'une partie du schéma de la BDD, notamment la table candidat.
- Réalisation de l'insertion des données provenant du fichier Inscription.
- Fonction d'insertion dans la BDD (*InsertData* et *InsertOrUpdateData*)
- Modification des fonctions Add pour supprimer le redondance du code d'insertion des tables, en utilisant les 2 fonctions ci-dessus
- Fonction de statistiques et de test.
- Relecture et corrections diverses dans les différentes fonctions

— **Céline Zhu :**

- Réalisation d'une partie du schéma de la BDD, notamment la création des tables adjacentes à la table candidats et de leurs jointures
- réalisation des scripts de lecture des fichiers Excel et CSV

- réalisation de la fonction d'upload des données pour listeEtablissement
- réalisation de l'application Flask
- réalisation des fichier .sh pour automatiser Flask et automatiser la création de l'environnement virtuel nécessaire

Chapitre 7

Conclusion

Tout d'abord, nous avons pris du temps à démarrer. Bien que nous nous soyons rapidement mis d'accord sur les technologies à utiliser, nous avons pris beaucoup de temps à concevoir la base de données.

Pour 2 raisons : tous d'abords, il y a plusieurs fichier, et en particulier "Inscription", qui contiennent beaucoup de colonnes.

Il fallait donc être rigoureux pour ne rien oublier, mais tout en veillant à n'avoir que des tables et des attributs pertinents. En effet, il fallait respecter la contrainte de normalisation de la table, et éviter aux maximum les redondances.

De plus, il y a globalement de nombreux fichiers, et de nombreuses données dans chacun d'eux. Tout était plus ou moins liés, ce qui augmentait la complexité. De plus, cette intrication entre les fichiers et les tables rendaient plus difficile de travailler à plusieurs simultanément sur le schéma de la base de données.

Pour l'application, nous avons bien réussi à travailler ensemble et à avoir un code aussi propre que possible. Cela est passé notamment en ayant de nombreux fichiers différents pour séparer les différents domaines de l'application. Nous avons de plus veillé à créer des fonctions simples et assez courtes ne faisant qu'une chose à la fois. Nous les avons documentées, afin de savoir ce qu'elle font, prennent en argument et renvoie. Cela à permis de facilement travailler à plusieurs sur la même chose en parallèle, et de pouvoir comprendre et corriger le code des autres avec aisance.

Pour la gestion de projet, les différents documents fait au début du projet,

et complété éventuellement par la suite, ont permis de savoir quoi faire. Nous avons ainsi évité de nous disperser sur des idées ou des conceptions différentes.

De plus, nous étions tous les 3 à l'aise avec la méthodologies que nous avons suivi, consistant à travailler de notre côté. Nous mettions en commun notre avancé et nos problèmes lors de réunion hebdomadaire de généralement 30 à 45min. Elles ont rarement dépassé 1h, lors de réflexions et travaux en commun, notamment pour des documents de gestion de projet ou modification de la base de données.

7.1 Conclusion personnelle

7.1.1 LEDEUX Flavien

J'ai trouvé ce projet intéressant car les informations vis à vis du projet venaient petit à petit et que le sujet restait vague mais laissant la possibilité d'être précisé à l'aide de questions, ce qui est quelque chose qui ressemble à un exemple concret de projet que l'on risque de rencontrer dans le futur.

D'autre part, ce projet m'a permis de me familiariser avec un langage qui m'était presque inconnu : le langage des fichiers batch ainsi que de revoir les scripts bash que j'avais vue au tout début de ma formation d'IUT.

De plus le projet m'a permis de travailler avec des personnes qui m'étaient inconnues confrontant ainsi mes méthodes de travailles à celles des autres forçant ainsi une adaptation enrichissante dans le but de garder le meilleur de chacun.

Je trouve cependant que notre projet n'as malheureusement pas atteint toutes les attentes que j'avais vis à vis de ce dernier. Cela est pour de multiples raisons telles que le manque de temps pour approfondir certaines thématiques, la présence des partiels lors de la fin de projet limitant le temps disponible a consacrer a ce dernier et un investissement plus important possible vis à vis du projet.

Si je devais refaire le projet de zéro, j'essaierais de mieux gérer la l'utilisation de notre temps en nous planifiant mieux et en séparant mieux les différentes tâches. De plus, j'essaierais aussi de nous faire travailler avec une cadence plus importante sur le début du projet.

D'autre part, la chose que j'essaierais le plus d'appliquer à d'autres projets est notre méthode rigoureuse de planification et d'exécution des réunions qui nous a permis d'éviter de rester bloquer trop longtemps. D'autre part, le contenu des réunions était toujours pertinent et le travail était au rendez-vous lors de ces dernières ce qui ce reflète dans des temps de réunion ne

dépassant qu'occasionnellement une heure.

Enfin, ce projet m'as forcé a prendre du recul vis a vis de ma méthode de travailler car j'ai du développer un rôle plus en retrait vis à vis de la création de la base de données pour pouvoir laisser mes collègues gagner de l'expérience avec cette notion au lieu de me concentrer la faire a ma façon au détriment de mes collègues.

Pour conclure, j'ai trouvé ce projet enrichissant bien qu'il laisse un arrière goût amer en quand on réfléchit à ce qui aurait pue être amélioré où rajouter.

7.1.2 ROURA Guilhèm

Ce projet fut mon premier projet complet, allant de la conception à la réalisation. J'ai d'abord pu m'entraîner et m'améliorer à la conception d'une base de données, chose que je ne maîtrisais pas forcément au départ. J'ai pu bien prendre conscience et mettre e pratique des concepts tel que la normalisation d'une base de données, ou encore la manière d'éviter la redondance par des tables auxiliaires.

Pour ce qui est de la programmation, cela ma permis de faire mon premier projet complet en python. Je me sentais déjà assez à l'aise avec, mais je n'avais jamais fait de projet de cette ampleur.

De manière général, j'ai pu me rendre compte de la difficulté et la complexité d'entière concevoir une application qui soit solution à un problème précis. La liberté de conception permet de se sentir réellement impliqué et de faire des choix en fonction de ses préférences et de ce qu'on juge le plus pertinent.

Cette liberté est cependant à double tranchant : Il faut savoir se fixer des limites pour ne pas rester bloqué sur la phase de conception, tout en étant aussi rigoureux que possible. Tous les mauvais choix auront des répercussions par la suite, ils peuvent obliger à s'adapter lors de la phase de réalisation, par exemple en raison du langage et des technologies utilisées. Mais la conception même peut être à revoir, par exemple pour le schéma de la base de données, si cette dernière ne répond finalement pas à toutes les problématiques posés. On peut aussi la revoir si une solution bien meilleur est trouvée par la suite.

Enfin, ce projet fut le premier que je fais avec des personnes que je ne connais pas, et ayant des parcours différents. Ces différences m'ont cependant semblé être plus des forces que des faiblesses, puisque chacun à ses propres points forts et connaissance à apporter. Le fait de devoir adapter sa méthode de travail m'a semblé valoir le coup, puisqu'on peut tenter de garder le meilleur de chacun en terme de méthode. De plus, chacun s'est investi et s'est senti responsable du projet, dans l'échec comme dans la réussite, ce qui constitue une bonne synergie, et nous a permit de rester motivé.

Pour finir, mon ressenti sur le projet est que le temps pris au début pour concevoir la base de données et commencer les scripts python fut trop important. Cela a laissé moins de temps pour avoir un résultat satisfaisant. Cependant, la grande complexité des données à analyser a obligé à prendre du temps pour bien tout comprendre et tout concevoir correctement.

7.1.3 ZHU Céline

La différence entre théorie et pratique m'a parfois surprise durant la réalisation de ce projet que ce soit en gestion de projet ou en informatique. L'approfondissement du Flask par exemple a été une expérience enrichissante allant au delà des cours fournis et allant dans le domaine de la recherche personnelle. C'était toujours très gratifiant de faire réussir à coder un nouveau bout du site et d'interagir très rapidement avec.

La réalisation de mon premier projet complet ne vient cependant pas sans défaut. En effet, nous avons eu la chance d'avoir un sujet qui nous laissez beaucoup de liberté et parfois ne maîtrisant pas bien mes limites, je me lançait dans des projets trop ambitieux pour mon niveau : par exemple le carrousel qui a été très chronophage pour un rendu finalement qui n'était peut-être pas tellement intéressant.

C'est aussi mon premier projet en méthode Agile avec des gens que je ne connaissais pas. Heureusement, nous avons eu des complémentaires grâce à des parcours scolaires très différents nous permettant ainsi sur beaucoup de niveau de nous combler nos lacunes.

Qu'est-ce qu'on pourrait améliorer ? Beaucoup de choses, mais je citerai trois points clés : un démarrage en gestion de projet plus rapide, une sauvegarde plus automatique des fonctions codées et enfin être moins optimiste quant aux dates buttoirs. Ce premier projet d'informatique m'a permis de vérifier la bonne acquisitions des compétences acquise durant cette première année à TELECOM Nancy. En conclusion, ce fut une expérience enrichissante qui s'est très bien passée dû à la synergie de travail du groupe.

Bibliographie

- [1] DEVELOPPEZ.COM. *Qt - WikiDeveloppez*. URL : <https://wiki.developpez.com/Qt>.
- [2] DEVHINTS.IO. *Bash scripting cheatsheet*. URL : <https://devhints.io/bash>.
- [3] DOCS.PYTHON. *Documentation module python re*. URL : <https://docs.python.org/fr/3/library/re.html>.
- [4] FLASK.COM. *Welcome to Flask — Flask Documentation (2.0.x)*. URL : <https://flask.palletsprojects.com/en/2.0.x/>.
- [5] PYTHON-SIMPLE.COM. *openpyxl (lecture et ecriture xlsx)*. URL : <http://www.python-simple.com/python-autres-modules-non-standards/openpyxl.php>.
- [6] PYTHON.ORG. *3.9.5 Documentation*. URL : <https://docs.python.org/3/>.
- [7] PYTHON.ORG. *sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.9.5 documentation*. URL : <https://docs.python.org/3/library/sqlite3.html>.
- [8] PYTHONEXAMPLES.ORG. *python convert html to pdf*. URL : <https://pythonexamples.org/python-convert-html-to-pdf/>.
- [9] SQL.SH. *Cours et Tutoriels sur le Langage SQL*. URL : <https://sql.sh/>.
- [10] SQLITE.ORG. *SQLite In 5 Minutes Or Less*. URL : <https://www.sqlite.org/quickstart.html>.
- [11] WIKIPEDIA.ORG. *Database normalization - Wikipedia*. URL : https://en.wikipedia.org/wiki/Database_normalization.
- [12] WIKIPEDIA.ORG. *JavaFX — Wikipédia*. URL : <https://fr.wikipedia.org/wiki/JavaFX>.
- [13] WIKIPEDIA.ORG. *MySQL - Wikipédia*. URL : <https://fr.wikipedia.org/wiki/MySQL>.

- [14] WIKIPEDIA.ORG. *SQLite* - *Wikipédia*. URL : <https://fr.wikipedia.org/wiki/SQLite>.

Chapitre 8

Annexe

8.1 Schéma de la base de données A



P2II 1A - Compte-rendu n°1

Motif de la réunion : Réunion de démarrage du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Céline ZHU	Date : 03/04/2021 Heure : 10h01 Durée : 45 minutes

● **Ordre du jour :**

- **Prendre connaissance du sujet**
- **Établir les outils de GdP**
- **Établir le Cahier des charges**
- **Planifier la semaine d'après**
- **Latex (?)**
- **Langage de programmation/ SGBD**

● **Prendre connaissance du sujet**

- **Rapport sur la Gdp et la réflexion**
- **Objets à rendre:**
 - **Base de données SQL standard + schéma**
 - **Rapports formats html (rédigé en fonction de quoi à demander à M Da silva ou M Oster)**

- Afficher les résultats des requêtes que ce soit en schéma ou par texte
- L'état des fichiers final ou pas.
- Établir les outils de GdP
 - Fichier Readme.md
 - Matrice RACI
 - Gantt
 - SWOT
- Établir le Cahier des charges
 - Base de données SQL:
 - script Python rendant une base de données cohérente
 - Avec des informations
 - Trois étapes: Design, construction, tests boîte noire/blanche
 - Application Python (html):
 - Tous Flask
 - Définir tous les liens qui sont liés à notre application
 - Définir les conventions Html
 - Fiches de style
- Latex (?)

Création du latex sur Overleaf et séparation en plusieurs fichiers Tex plus tard.
- Programmation

Flavien: Python(flask) et Sqlite sont plus pratiques pour des amateurs. et on peut s'aider de la WebBD.
- Planifier la semaine d'après
 - Faire les scripts de la base de données pour les créer et le schéma.
 - Schéma: Flavien/Guilhèm
 - Scripts: Céline/
 - Tests: Flavien/Céline
 - Commencer à réaliser les classes représentant les différentes éléments à ajouter dans la BDD.

- Documenter vos actions en latex.

Prochaine réunion:

- **Samedi à 10h01 10/04/21:**

odj:

- **Avancement du projet**
- **Construction de la matrice RACI**
- **Faire le Gantt**
- **Dire les endroits qui nous ont posés problèmes**

P2II 1A - Compte-rendu n°2

Motif de la réunion : Réunion d'avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Flavien LEDEUX	Date : 10/04/2021 Heure : 10h01 Durée : 1 heure

- **Ordre du jour :**

- Avancement du projet
- Construction de la matrice RACI
- Faire le Gantt
- Dire les endroits qui nous ont posés problèmes

- **Avancement du projet**

- début conception
- fichier qui recense les fichiers de data implémentés
- début de l'écriture du script de la base de données
- vérification de la cohérence des noms dans la base de données
- élimination de certaines redondances

- **Construction de la matrice RACI**

- début de la création de la matrice RACI en fonction des informations déjà fournies dans le sujet.

- **Faire le Gantt**

- début de la réalisation du diagramme de Gantt

- **Dire les endroits qui nous ont posés problèmes**

- redondance des données dans les fichiers
- ambiguïté sur les informations

- **Planifier la semaine d'après**

- **Faire les scripts de la base de données pour les créer et le schéma.**
 - **Schéma: Céline/Flavien**
 - **Scripts: Guilhèm/Céline**
- **Rédaction d'information sur le projet: Flavien/Guilhèm**
- **Documenter vos actions en latex.**

Prochaine réunion:

- **Samedi à 10h01 10/04/21:**
 - odj:**
 - **Avancement du projet**
 - **Compléter la matrice RACI**
 - **Compléter le Gantt**
 - **Dire les endroits qui nous ont posés problèmes**
 - **Répartition des tâches**

P2II 1A - Compte-rendu n°3

Motif de la réunion : Réunion d'avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Guilhèm ROURA	Date : 17/04/2021 Heure : 10h01 Durée : 1 heure

• **Ordre du jour :**

- Avancement du projet
- Compléter la matrice RACI
- Compléter le Gantt
- Dire les endroits qui nous ont posés problèmes
- Répartition des tâches

• **Avancement du projet**

- **Flavien:** rédaction de 2 pages du rapport (chapitre 1: étude biblio). Elle mentionnent les technologies utilisées et le choix effectués suivant les pour et le contres.
- **Céline:** A continué le schéma de la base de la base de données. La table "candidat" est maintenant complète. M. Da Silva aussi pu lui éclaircir un point quand aux communes. Elle a aussi ajouté de

nombreuses tables pour éliminer les redondances. On va finalement fusionner les deux tables "villecrit" et "commune".

Dédoublage de champs/tables : "voie" et "filière" stocke la même chose, il va falloir en supprimer un des deux

Début d'une table pour les notes: Questionnement sur est-ce qu'il faut

- **Guilhèm** : A continué le fichier de création de la base de données en SQLite, en adaptant le schéma de base de données de Céline.

● **Dire les endroits qui nous ont posés problèmes**

- Céline: Le fichier d'inscription a été difficile à adapter en base de données (nombreux champs et beaucoup de données potentiellement redondantes)

- **Planifier la semaine d'après**

- **Faire les scripts de la base de données pour les créer et le schéma.**
 - **Schéma: Céline/Flavien**
 - **Scripts: Guilhèm/Céline**
- **Rédaction d'information sur le projet: Flavien/Guilhèm**
- **Début du script Python (parsing): Flavien**
- **Documenter vos actions en latex.**

Prochaine réunion:

- **Samedi après-midi le 24/04/21:**
 - odj:**
 - **Avancement du projet**
 - **Compléter la matrice RACI**
 - **Compléter le Gantt**
 - **Dire les endroits qui nous ont posés problèmes**
 - **Répartition des tâches**

P2II 1A - Compte-rendu n°4

Motif de la réunion : Réunion de démarrage du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Céline ZHU	Date : 24/04/2021 Heure : 19h01 Durée : 45 minutes

• **Ordre du jour :**

- **Avancement du projet**
- **Compléter la matrice RACI**
- **Compléter le Gantt**
- **Dire les endroits qui nous ont posés problèmes**
- **Répartition des tâches**

Flavien: gitlab fichier python: script qui crée la base de donnée en fonction du fichier passé en paramètre, base de données sans commentaire

Céline: Continuer la base de données

Guilhèm: Vérification base de données et écriture du script de création de la table.

Pour la semaine prochaine: finir la base de données, aborder le Flask.

- Planifier la semaine d'après

- Faire les scripts de la base de données pour les créer et le schéma.

- Schéma: Guilhèm/Flavien

- Scripts: Céline
- Rédaction d'information sur le projet: Flavien/Guilhèm
- Documenter vos actions en latex.

Prochaine réunion:

- Samedi à 16h00 01/05/21:
 - odj:
 - Avancement du projet
 - Construction de la matrice RACI
 - Faire le Gantt
 - Dire les endroits qui nous ont posés problèmes

P2II 1A - Compte-rendu n°5

Motif de la réunion : Réunion de démarrage du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Flavien LEDEUX	Date : 01/05/2021 Heure : 15h Durée : 1 heure 20 minutes

• **Ordre du jour :**

- **Avancement du projet**
- **Construction de la matrice RACI**
- **Faire le Gantt**
- **Dire les endroits qui nous ont posés problèmes**

Flavien gitlab fichier batch: Fichier batch pour installer python et les dépendances du projet de façon automatique pour windows

Guilhèm : Continuation du schéma de la BDD

Céline : réalisation des scripts de réalisation de table en fonctions du schéma

Note : Réunion longue pour discuter de différents points à améliorer dans la conception de la base de données et surtout de point posant problème. Notamment, création de tables auxiliaires pour éviter la redondance.

Pour la semaine prochaine: finir la base de données, aborder le Flask.

- Planifier la semaine d'après
 - Faire les scripts de la base de données pour les créer et le schéma.
 - Schéma: Céline/Flavien
 - Scripts: Guilhèm
 - code de lecture des données : Céline
 - Fichier test excel : Guilhèm
 - Liste des contraintes des données : Guilhèm
 - Faire les scripts d'installation automatique pour Linux et Mac : Flavien
 - Documenter l'installation
 - Documenter vos actions en latex.

Prochaine réunion:

- Samedi à 15h00 08/05/21:
 - odj:
 - Avancement du projet
 - Dire les endroits qui nous ont posés problèmes
 - Discuter le cahier des charges du terme application
 - fixer la deadline de l'application

P2II 1A - Compte-rendu n°6

Motif de la réunion : Avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Guilhèm ROURA	Date : 08/05/2021 Heure : 15h20 Durée : 30 min

• **Ordre du jour :**

- **Avancement du projet**
- **Construction de la matrice RACI**
- **Faire le Gantt**
- **Dire les endroits qui nous ont posés problèmes**

Flavien : Ajout de commentaire dans le fichier .bat d'installation de python sur Windows pour tout expliquer.

Création du même fichier pour Linux (.sh) : beaucoup plus simple à gérer sur linux que sur windows, notamment grâce aux paquets manageurs. Gère actuellement les 3 packages managers les plus courants (pourra en rajouter après).

Va utiliser une VM Linux et Mac pour tester ses fichiers d'installation

Fichier .py qui coupe le serveur Flash (Pour à terme tout automatiser)

Céline : Création de 3 fichiers : Une qui lit un fichier .xlsx (Excel), une qui lit un .csv, et une qui lit tous les fichiers du répertoire en choisissant la fonction suivant l'extension du fichier.

Les 2 premières fonctions renvoient une liste : le premier éléments est le nom du fichier, puis chaque élément correspond à une ligne du fichier : Structure particulière stockant le type de donnée et la valeur

Guilhèm : Écritures des contraintes de valeurs pour toutes les données du fichier Inscription. Exemple : Quelles valeurs possibles pour CSP, département, etc....

Écritures des contraintes entre diverses valeurs du fichier Inscription. Exemple: Les valeurs de "Code_csp" et "libellé_csp" sont liés: l'une définit l'autre

Tout ceci permettra des vérifications de cohérence des fichiers

Pour la semaine prochaine: Faire des fonctions remplissant la BDD à partir des fichiers

- Planifier la semaine d'après

- Faire les scripts de la base de données pour les créer et le schéma.

(Flavien/Guilhèm/Céline)

=> Chacun s'occupe d'une partie des fichiers : Une fonction par fichier ou famille de fichier. Répartition à adapter suivant le travail pour chaque fichier

- Finir les scripts d'installation automatique pour Linux et Mac : Flavien
 - Documenter l'installation
 - Documenter vos actions en latex.

Prochaine réunion:

- **Samedi à 15h00 15/05/21:**

- odj:**

- Avancement du projet
 - Dire les endroits qui nous ont posés problèmes
 - Discuter le cahier des charges du terme application
 - fixer la deadline de l'application

P2II 1A - Compte-rendu n°7

Motif de la réunion : Avancement du projet	Lieu de la réunion : Réunion par écrit sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Céline ZHU	Date : 15/05/2021 Heure : 15h20 Durée : 55 min

- **Ordre du jour :**

- **Avancement du projet**
- **Dire les endroits qui nous ont posés problèmes**
- **Discuter le cahier des charges du terme application**
- **fixer la deadline de l'application**

Note: La réunion fu par écrit pour raison de problème technique.

- **Avancement du projet**

Guilhèm: 1)Modifié le script de lecture de fichier, afin d'utiliser un module compatible avec les ".xlsx". J'ai aussi légèrement modifié le reste pour avoir un code plus "direct". J'ai sauvegardé ça dans un fichier "V2".

2) Création de 2 fichiers: un avec les fonctions qui remplissent la BDD à partir d'un fichier (grâce au fichier de lecture). Et un fichier avec les fonctions qui parse des données précises.

a) Pour ce dernier, il a fait une fonction qui gère les numéros de téléphone pour les passer en format "+33 X XX XX XX XX" à partir des 4 formats trouvable dans le fichier.

b) Pour le 1er fichier, j'ai commencé une fonction qui remplit la table "candidat" via le fichier "Inscription". Elle a 2 problèmes: actuellement elle crée l'entrée, mais ne gère pas si il existe déjà une entrée (Il suffit de reprendre le code qu'a fait Flavien, donc pas trop gênant)

Mais de plus, elle rentre toutes les données d'une ligne dans un dictionnaire, puis utilise ce dictionnaire pour remplir la table, ce qui ne va pas marcher (certaines données sont à mettre dans une des tables auxiliaire par exemple). Il faut donc que je l'adapte pour que le dictionnaire ne contiennent que les données à mettre dans les candidats, puis utilise le reste pour remplir les autres tables.

3) en voulant remplir la table candidat, je me suis rendu compte que la BDD a des problèmes. Déjà, il faudrait enlever les "not null". Certains n'ont pas de sens (comme "rang oral"), et en général ça crée des problèmes pour remplir la table.

Mais surtout, je crois que la BDD n'avait pas été mis à jour après une des dernières réunions puisqu'il n'y a pas les 2 tables devant gérer les matières, et qu'il y a encore "voix et filière" (qui sont la même chose). De plus, je crois qu'il y a des erreurs dans les candidats (p-ê ma faute). Je crois qu'il manque des données (par ex. "Epreuve 2 à 4" ou "qualité")

Pour ce dernier point, il faudrait donc à priori corriger la table suivant ce qui avait été dit dans une des précédents réunion, et relire la table candidats pour vérifier les données manquantes par rapport à "Inscription"

Céline Zhu — 15/05/2021

A un peu contribué à la fonction lié à 'inscription'. A fait une fonction 'établissement' et 'parse établissement' et a ajouté un !/bin/python au fichier.

Flavien — 15/05/2021

Travail fait :

finitions des script d'installation des dépendances (il faut juste ajouter 2-3 commandes en fonctions des nouvelles dépendances qui vont arriver ex le module pour les fichiers excel)

mise en place du polymorphisme pour la lecture des fichiers + création
d'une fonction renvoie les donnée en utilisant le polymorphisme

création de script pour le chargement des fichiers Admis et Admissible,
plus création de pays, commune et résultat

- **Dire les endroits qui nous ont posés problèmes**

Céline Zhu — 15/05/2021

N'arrive pas à faire marcher le fichier polymorphisme de Flavien,
mais elle suppose que c'est juste un problème de format.

- **Discuter le cahier des charges du terme application**

cf docu cahier des charges

- **fixer la deadline de l'application**

Prochaine réunion:

- **Samedi à 15h00 22/05/21:**

odj:

- Avancement du projet
- Dire les endroits qui nous ont posés problèmes
- Discuter le cahier des charges du terme application
- fixer la deadline de l'application

P2II 1A - Compte-rendu n°8

Motif de la réunion : Avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Flavien LEDEUX	Date : 21/05/2021 Heure : 15h00 Durée : 45 min

- **Ordre du jour :**

- **Avancement du projet**
- **Dire les endroits qui nous ont posés problèmes**
- **Discuter le cahier des charges du terme application**
- **fixer la deadline de l'application**

- **Avancement du projet**

Guilhèm:

- continuation de la fonction de chargement du fichier inscription
- clarification du code
- fonction ajoute csp

Celine :

- début de la création de l'appli
- accès à des données à partir d'info d'un candidat
- création de candidat depuis l'application
- professeur : consulter des profils où uploader un fichier
- création de fichiers bash pour de la qualité de vie

- Tentative d'édition de profil

Flavien :

- réalisation du fichier de gestion d'upload des notes et des voeux
- modification de la base de données pour accommoder l'insertion des notes et des voeux
- **Dire les endroits qui nous ont posés problèmes**
 - Problème de conception de la base de données demandant des modifications
 - Manque d'information vis à vis du projet pour progresser

Prochaine réunion:

- **Samedi à 15h00 29/05/21:**

odj:

- Avancement du projet (finitions du chargement des fichiers et continuation de l'application flask)
- Dire les endroits qui nous ont posés problèmes
- avancement de la gestion de projet (faire la charte de projet)

P2II 1A - Compte-rendu n°9

Motif de la réunion : Avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Guilhèm ROURA	Date : 30/05/2021 Heure : 15h00 Durée : 1h20 min

- **Ordre du jour :**

- **Avancement du projet**
- **Dire les endroits qui nous ont posés problèmes**

- **Avancement du projet**

Guilhèm:

- Continuer le fichier Inscription : Quelque modif nécessaire pour que la table candidat corresponde complètement au fichier Inscription.
- Modification de 2 fonction lié à l'établissement du candidat, pour gérer le fait que les infos sont présentes dans 2 fichiers différents (listeEtabl et Inscription)

Celine :

- Avancement sur l'appli Flask:
 - 4 catégorie sur l'accueil "Je suis Candidat/Professeur/Nouveau candidat/ Administration"

- je suis candidat : utilise nom de famille/numéro de candidat.
Permet de voir ses notes, ses voeux et ses informations
- Je suis professeur:
- Je suis nouveau candidat : Permet d'ajouter une entrée à la table candidat
- Administrateur : utiliserais un compte : Permet de faire des requêtes SQL / Permet de rechercher des données

Flavien :

- Fixe upload des fichiers Admi et Admissible
- Fonction qui importe les fichier CMT_Oraux
- MAJ Schéma BDD : Quasi à jour (à 2-3 détail près)
- **Dire les endroits qui nous ont posés problèmes**
 - Flavien : zone d'ombre (attente de réunion) : liste etat réponse appel
- **Gestion de projet ensemble**
 - Complète la charte de projet ensemble

Prochaine réunion:

- **Samedi à 15h00 29/05/21:**

odj:

- Avancement du projet
- Dire les endroits qui nous ont posés problèmes
- Continuer le rapport LaTeX

A faire: Céline:

- Finir l'appli Flask, Trouver comment faire une conversion HTML -> PDF

P2II 1A - Compte-rendu n°10

Motif de la réunion : Avancement du projet	Lieu de la réunion : Réunion en visioconférence sur Discord
Présents : <ul style="list-style-type: none">• Flavien LEDEUX• Guilhèm ROURA• Céline ZHU Absent : <ul style="list-style-type: none">• Personne Rédaction du compte-rendu : <ul style="list-style-type: none">• Céline ZHU	Date : 05/06/2021 Heure : 15h00 Durée : 1h20 min

- **Ordre du jour :**

- **Avancement du projet**
- **Dire les endroits qui nous ont posés problèmes**

- **Avancement du projet**

Flavien:

- Continuer les fichiers de remplissages, insertion de données, problème de csv. Problème d'insertion de données des fichiers oraux.
- fichier classe: max physique, max oral, total interclassement.
- fichier total écrit, fichier total oral.
- A fini l'implémentation des fichiers oraux et écrits dans la bdd, mp cmt classe cmt spé sei.

Celine :

- Avancement sur l'appli Flask:
 - 4 catégorie sur l'accueil "Je suis Candidat/Professeur/Nouveau candidat/ Administration"

- je suis candidat : utilise nom de famille/numéro de candidat.
Permet de voir ses notes, ses vœux et ses informations
- Je suis professeur:
- Je suis nouveau candidat : Permet d'ajouter une entrée à la table candidat
- Administrateur : utiliserais un compte : Permet de faire des

Guilhèm :

- Insère toutes données par les données auxiliaire, remplace toutes les données, fonction insère_data(), une version qui fait que installer et qui installe aussi, code pays, clé auto-incrémentée; add country, le nom .
- A séparé les fichiers upload et add.
- au remplissage, vérifie les tests
- **Dire les endroits qui nous ont posés problèmes**
 - Flavien : zone d'ombre (attente de réunion) : liste etat réponse appel
- **Gestion de projet ensemble**
 - Complète la charte de projet ensemble

Moyenne et occupation, des stats

Faire un fichier python qui regroupe plusieurs fonctions, chaque fonction, une fonction sql, les différentes fonctions, les at qui récupère les différentes jointures, récupère la liste qui affiche les différentes, dans un table html.

Prochaine réunion:

- **Samedi à 15h00:**
odj:
 - Préparation soutenance

Charte de projet

Auteurs

Nom / mél	Qualité / rôle
Flavien Ledoux / flavien.ledoux@telecomnancy.eu	Projet / Participant
Guilhèm Roura / guilhem.roura@telecomnancy.eu	Projet / Participant
Céline Zhu / celine.zhu@telecomnancy.eu	Projet / Participant

Historique des modifications et révisions de ce document

n° de version	Date	Description et circonstances de la modification
V 0.1	01/05/2020	Brouillon : première version suite à la réunion initiale de validation du cahier des charges fonctionnel
V 1	30/05/2020	Demande de compléments d'informations
V 1.1	10/06/2021	Validation

TABLE DES MATIÈRES

Résumé
Cadrage
Finalités et importance du projet
Contexte / hypothèses de départ
Objectifs et résultats opérationnels
Déroulement du projet
Organisation / ressources, budget
Jalons : échéancier / événements importants
Risques et opportunités

Résumé

Ce projet se développe dans le cadre du PII, il vise la mobilisation des connaissances acquises durant notre première année d'école d'ingénieur à Télécom Nancy.

Les algorithmes et leur étude développés lors de ce projet doivent répondre à l'énoncé fourni par les enseignants :

- Sébastien Da Silva <sebastien.dasilva@telecomnancy.eu>
- Gérald Oster <gerald.oster@telecomnancy.eu>.

Tous les élèves participant au projet doivent maîtriser les différentes parties de l'application de tel sorte qu'ils soient capables de répondre à des questions sur ces dernières. Le projet se déroule par jalons définis à partir des sous parties de l'énoncé.

Le projet vise à livrer une application, incluant une base de données et une méthode d'insertion de données depuis des fichiers excel et csv, qui sera notée.

Cadrage

Finalités et importance du projet

Ce projet cherche à répondre à la totalité des questions de son énoncé. On s'intéresse à la création d'une application qui permet l'utilisation et la modification d'une base de données créée à partir de fichiers fournis. Le développement et l'étude des fonctions algorithmiques, l'application résultante ainsi que la gestion du projet seront conçus par toutes les participants et le compte rendu final comportera un document rédigé sur Latex ainsi qu'une présentation orale accompagnée d'une démonstration. À l'issue du compte-rendu, le module sera soumis à une validation.

Contexte / hypothèses de départ

	INCIDENCE POSITIVE	INCIDENCE NÉGATIVE
ORIGINE INTERNE	<ul style="list-style-type: none"> - Les membres de l'équipe savent programmer en python et expérience Flask par le module WebBD - Les membres de l'équipe sont motivés et travailleurs - Les membres de l'équipe ont des parcours scolaires différents et complémentaires - accès à des machines installé uniquement pour tester l'application 	<ul style="list-style-type: none"> - La majorité des membres de l'équipe n'avaient jamais crée d'application avec front et back complète - Les membres de l'équipe ne se connaissait pas et n'avait pas travaillé ensemble auparavant
ORIGINE EXTERNE	<ul style="list-style-type: none"> - La possibilité de solliciter une personne extérieure au projet pour s'entraîner à la présentation orale - La possibilité de demander de l'aide à un professeur en cas de blocage - Des horaires flexibles 	<ul style="list-style-type: none"> - Un possible mauvais accès à internet - Une absence de date butoire - Le sujet n'est pas entièrement définie

Objectifs et résultats opérationnels

Les livrables de ce projet sont les compte rendus finales :

Une application permettant d'importer des fichier excel dans une base de donnée, et générer des rapports en HTML ou PDF

Un rapport sur Latex et probablement une présentation orale suivie d'une démonstration.

Déroulement du projet

Organisation / ressources

- Dans le déroulement de ce projet, les participantes sont :

- Flavien Ledoux
- Guilhèm Roura
- Céline Zhu

Chacune d'entre elles aura le même rôle en tant que développeur et manager.

- Les moyens à disposition sont multiples :
 - Logiciels de traitement de texte tel Latex
 - Logiciels permettant travail en équipe tel Gitlab
 - Logiciel de création de machine virtuelle pour effectuer des tests
 - Connaissances acquises lors de l'année pour le développement et l'étude d'algorithmes
 - Connaissances acquises lors du module de MOOC pour la gestion de projet
 - Éventuelle aide des professeurs de TÉLÉCOM Nancy

Jalons : échéancier / événements importants

Jalon	Description	Date
Étape 1 : Exigences opérationnelles	Validation des spécifications : cahier des charges techniques.	17/04
Étape 2 : Préparation pour démarrer le projet	Réalisation de la matrice SWOT, charte de Projet	24/04
Étape 3 : Développement du livrable	Récupération des données	08/05
Étape 4 : Développement du livrable	Création et étude des fonctions nécessaires à la BDD de même que leur rédaction dans le compte rendu.	15/05
Étape 5 : Développement du livrable	Création de scripts d'installation et de lancement de l'application	15/05
Étape 6 : Développement du livrable	Création et étude de l'application	14/06
Étape 7 : Compte rendu	Valider compte rendu final et présentation orale.	15/06

Risques et opportunités

Comme déjà spécifié dans la matrice SWOT présentée précédemment, après analyse du déroulement du projet, on dégage :

Plusieurs risques tels:

- Un potentiel mauvais accès à internet
- Une absence de date butoire
- Le sujet n'est pas entièrement définie

Plusieurs opportunités telles:

- La possibilité de solliciter une personne extérieure au projet pour s'entraîner à la présentation orale
- La possibilité de demander de l'aide à un professeur en cas de blocage
- Accès à l'école et à son matériel
- Beaucoup de documentations en ligne

Installation et utilisation de l'application

Installation

Installation automatique:

L'installation automatique fonctionne uniquement avec des ordinateurs Windows ou Linux. Si votre ordinateur ne suit pas ce pré-requis, veuillez suivre l'installation manuelle.

Exécutez le script installer.bat pour Windows ou installer.sh pour Linux depuis le repertoire où il est placé.

Une fois le script fini, un repertoire nommé ProjectEnvironnement devrait avoir été créé . Vous pouvez regarder son contenu pour vous assurer du bon fonctionnement du script.

Installation manuelle:

- Installer python3 si non présent :
 - Windows et mac : télécharger et installer la version voulu depuis ce site <https://www.python.org/downloads/>
 - Linux : Ubuntu (Pensez à "sudo" si besoin)
 - apt-get update
 - apt-get install python3
 - apt-get install python3-venv
 - Linux : Fedora
 - yum clean all
 - yum install python3
 - yum install python3-venv
 - Linux : autre
 - Utilisez le package manager de votre distribution
- (Optionnel) Création de l'environnement virtuel
 - <venv> : nom du repertoire où l'environnement virtuel sera créé
 - Windows :
 - python -m venv ./<venv>
 - <venv>\Scripts\activate.bat
 - .python -m pip install --upgrade pip
 - Linux :
 - python3 -m venv ./<venv>
 - ./<venv>/bin/activate
 - python3 -m pip install --upgrade pip
- Installation des dépendances
 - pip install -r ./requirements.txt