

**Lazare Potier de La Manadière Terminal S, Thibault Colchen Terminal S, David Ekchajzer Terminal S**

**Présentation**

Les heures de Travail Autonome (TA) proposées par le lycée La Source permettent aux élèves de pouvoir se perfectionner dans la matière qu’ils ont choisie. Cependant, l’inscription dans tous les niveaux reste un moment problématique puisque c’est la politique du premier arrivé premier servi qui est appliquée. Ce type d’inscription nous a paru injuste et nous avons ainsi pensé que, par les outils numériques, nous pourrions la rendre plus égalitaire.

Aujourd’hui, les heures de TA étant la spécificité de La Source, aucun n’outil n’a été développé pour résoudre ce genre de problématique. Cependant, des programmes comme celui d’APB, a également pour but, de trier les vœux soit par mérite (note, entretiens, lettres, CV…) soit pour les universités par tirage au sort. Notre programme a pour vocation de mettre plus de justice dans les tris en permettant à chacun d’avoir son choix le plus élevé. Si un élève n’a pas eu son premier choix, il sera prioritaire pour l’heure de TA suivante grâce à l’implantions d’une variable justice.

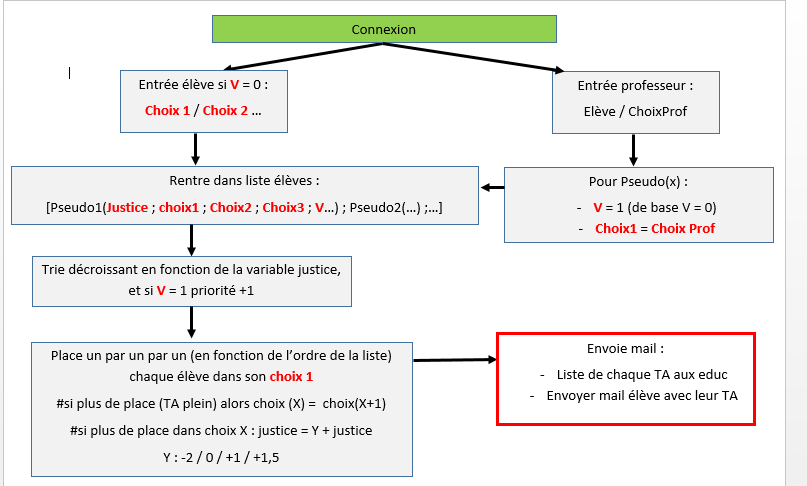
**But de départ**

**Cahier des charges**

Notre but était la création de deux espaces en ligne :

* Un espace élève (connexion + mise en ordre des 4 choix de TA) géré avec Django
* Un espace professeur (connexion + choix d’un élève à mettre dans son TA) géré avec Django
* Un algorithme permettant le tri des élèves dans leurs choix le plus haut en fonction gérée en python :
* De leur variable justice augmenté s’ils ont un de leur choix les plus hauts, baissés s’ils ont eu un choix bas
* De leur variable V (veto) qui indique la matière choisie par le professeur pour eux
* Des choix des autres élèves

**Infogrammes du programme**



**Planning des taches**

|  |  |  |
| --- | --- | --- |
| Apprendre HTML | 1 séance | Lazare / Thibault / David |
| Développer pages HTML | 1 séance | Lazare / Thibault / David |
| Apprendre Django | 3 séances | David |
| Apprendre Sqlite | 3 séances | Lazare |
| Développement de la base de données Sqlite | 3 séances | Lazare |
| Développement de l’algorithme python | 4 séances | Thibault |
| Développement de l’inscription avec Django | 4 séances | David |
| Assemblages des différents modules | 1 séance | Lazare / Thibault / David |

Pour les apprentissages, nous nous sommes appuyé sur des sites d’apprentissages tel que OpenClassroom ou les ressources officielles de Django et python, deux langages très documentés.

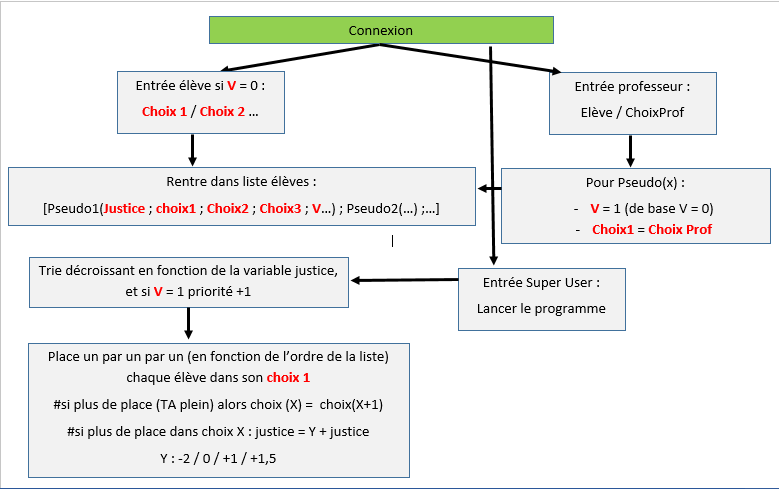
**Problèmes**

* L’apprentissage de Django pour faire un site d’identification et de formulaire a été très difficile. En effet, Django est un assemblage de fonction et pour apprendre un tel langage il est nécessaire d’avoir une connaissance globale de ces fonctions et nécessite donc beaucoup de pratique. Nous avons donc décidé de n’utiliser que le langage python à travers la librairie Tkinter, qui permet la mise en place d’une interface graphique. Cela permet finalement de faire la même chose que propose Django mise à part l’accès en ligne. Il s’agit donc d’un programme présent sur un ordinateur central où chaque élève peut s’identifier, Comme une fiche de TA informatique.

* Après de nombreux essai avec Tkinter, nous avons dû abandonner cette librairie car il nous était impossible de programmer un bouton quitter fonctionnel, nécessaire pour la sauvegarde des données.
* Nous avons essayé de créer une base de données avec Sqlite pour enregistrer les comptes des différents utilisateurs. Cependant, nous n’avons pas réussi à utiliser ce langage dans notre programme et après de multiples essaies, nous avons, découvert une librairie de python : pickle. Elle permet d’enregistrer n’importe qu’elle objet dans un fichier, rattaché au programme. Pickle nous as donc permis de sauvegarder les données de manière simple sans passé par une « base de donnée ».

Nous avons donc décidé d’utiliser seulement du python et nous avons refait notre planning des taches

**Infogrammes du programme 2**

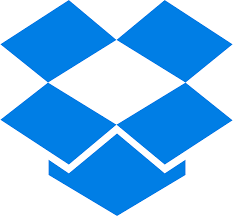


**Planning des taches 2**

En rouge les semaines déjà effectués avant les changements d’outils

|  |  |  |
| --- | --- | --- |
| Apprendre HTML | 1 semaine | Lazare / Thibault / David |
| Développer pages HTML | 1 semaine | Lazare / Thibault / David |
| Apprendre Django | 2 semaines | David |
| Apprendre Sqlite | 2 semaines | Lazare |
| Développement de la base de données Sqlite | 3 semaines | Lazare |
| Développement de l’algorithme python | 3 semaines | Thibault |
| Développement de l’inscription avec Django | 3 semaines | David |
| Développement de l’algorithme python | 3 semaines | David |
| Correction de l’algorithme python | 1 semaine | Lazare / Thibault |
| Mise en fonction de l’algorithme | 2 semaines | Lazare / Thibault |

Pour développer notre programme en collaboration, outre les séances d’ISN, nous avons collaborez grâce au partage d’écran de Skype lors de nos réunion et par Dropbox pour le développement au jour le jour. Nous avions par ailleurs un groupe Messenger pour pouvoir faire part de nos interrogations et nos problèmes puisque nous avons tous appris comment se servir des outils utilisés par chacun pour notre projet. Le fait de rester en lien toujours et d’avoir un regard « expert » sur la globalité du programme nous a permis de nous aider les uns les autres dans les différentes tâches que nous nous étions répartie. Github nous a servi à poster l’intégralité de notre programme dans le cloud pour une facilité de partage.







**Réalisation personnelle**

**Lazare :**

**Première partie : Django.**

Lorsqu’on a trouvé notre projet, nous ne savions pas tout de suite qu’il fallait apprendre des bases dans 4 langages. Le langage HTML et CSS s’est avéré indispensable pour la réalisation de la forme des pages internet de notre projet initial ; Nous avons commencé par l’apprentissage de ces deux-là pour faire la structure (HTML), puis la mise en forme de la structure (CSS). Personnellement, je me suis occupé de la page qui affichait les choix que les élèves pouvaient cocher dans l’ordre qu’ils voulaient. Pour cela j’ai suivi un cours d’OpenClassrooms sur la création d’une page en HTML/CSS. Nous avions 3 pages à réaliser : l’espace connexion principal suivit de l’espace professeur ou de l’espace élève selon les noms d’utilisateurs et mots de passe entrés. Puis l’espace de sélection des choix, que j’ai réalisé. Nous avions chacun une pages sur laquelle travailler. Je dirais que cette partie a été la plus simple pour moi car je me contentait d’appliquer ce que j’apprenais sur internet et ça marchais souvent du premier coup.

Une fois ces pages terminées nous avons commencé le développement de Django.

Django est un *Framework web* (espace de travail + outils, pour créer facilement des applications web)

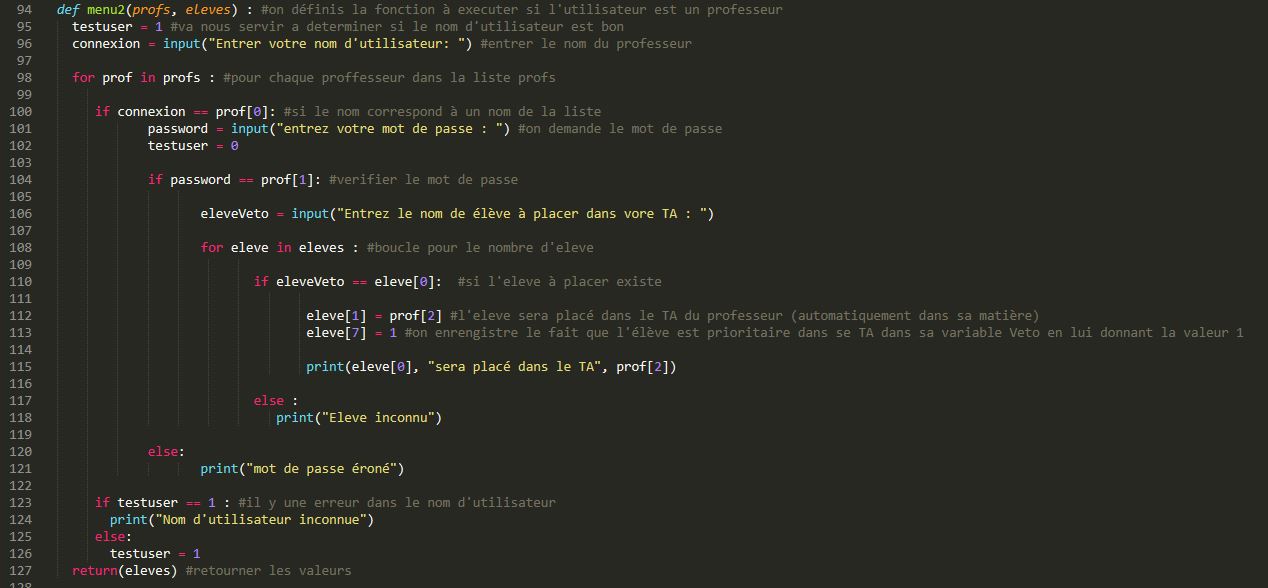
Il permet de regrouper un ensemble d’outils de base pour éviter de les réécrire et poser les bases de l’architecture du projet

Lorsque nous nous sommes répartit le travail, je me suis occupé de la base de donnée SQL Lite, partie intégrante de Django. Pour cela j’ai dû apprendre ce nouveau langage qui permet de stocker les données des élèves (pseudo, choix 1, choix 2, choix 3, choix 4, variable justice, variable veto) dans un tableau. Ce travail a été difficile, je manquais de connaissances général en fonctions complexes. Je n’avais avant cette année presque aucune connaissances informatique et notre projet était ambitieux. Avec SQL Lite, apprendre sur internet ne suffisait pas ou du moins mettait beaucoup de temps, notamment dans l’assimilation des concepts, des fonctions et du vocabulaire. J’ai trouvé néanmoins ce passage passionnant car j’avais vraiment l’impression de travailler avec un outil professionnelle de haut niveau. Je pense que c’est surtout le manque de temps ainsi que le manque d’expérience pratique qui nous a fait changer d’avis pour la suite de notre projet .

A partir de ce moment-là, après concertation, nous nous sommes mis d’accord sur le fait que nous manquerons de temps pour mener ce projet ambitieux, avec Django, SQL Lite.

**Deuxième partie : Python + Pickle et Tkinter :**

David à trouver à ce moment une librairie python appelée « Pickle » qui joue le rôle de base de donnée sous python. Apres que nous nous sommes détourné de Django pour réaliser notre projet avec python plus Pickle, Thibault et moi avons tenté de réaliser quand même une interface graphique à l’aide de Tkinter (module graphique de Python). Cependant, l’interface n’était pas indispensable car le code python sans Tkinter permettait déjà un certain affichage structuré.

**Fonctions détaillées :**

La fonction « menu2 » est la fonction destinée au professeur. C’est elle qui permet l’affectation d’un élève à un TA en passant outre les choix effectués auparavant d’où le nom de la variable « veto ». En premier, on initialise une variable « testuserprof = 1 » qui va nous permettre au programme si des informations sont erronées.

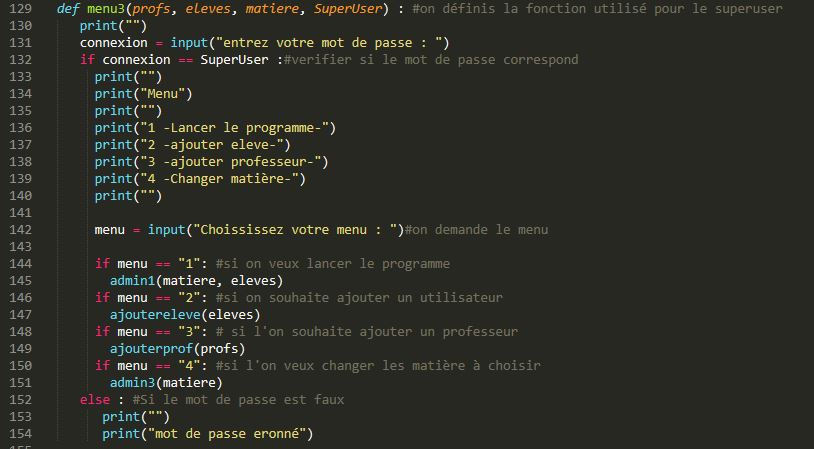
Ensuite une boucle est crée qui compte pour chaque professeur, le nom du professeur.

Puis avec de simples conditions on test le nom d’utilisateur entré puis le mot de passe (ceux-ci sont défini et modifiable par le Superuser (Admin)) .

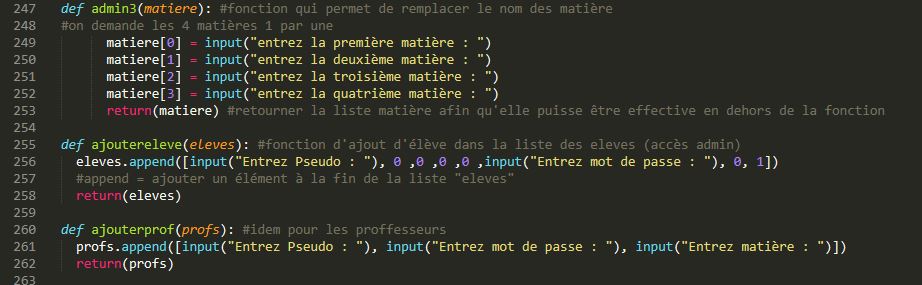
Une deuxième boucle indenté après les conditions compte pour chaque élève de la liste « eleve », le nom de l’élève à ajouter un TA souhaité. Les lignes qui suivent permettent

1° de vérifier que l’élève existe

2° d’affecter automatiquement l’élève souhaité dans le TA du professeurs connecté à l’espace.



La fonction « menu3 » est la fonction de connexion à l’espace SuperUser (Administrateur). D’ici, un administrateur peux 1° Lancer le programme 2° ajouter un élève/professeur à la liste des élèves/professeurs 3° changer une matière (en fonction des classes par exemple). L’Admin choisit l’action qu’il veut effectuer en tapant un chiffre (1,2,3 ou 4) dans la commande. Par la suite la fonction associé se déclenche.



Ces fonctions sont des sous-fonctions de l’espace SuperUser (« menu3 »). En effet, Ce sont les fonctions associées qui permettent ajouter une matière, un professeur ou un élève. On peut remarquer deux choses ici : la fonction « append », qui sert à ajouter une colonne à la fin d’une liste et la commande « return » qui sert simplement à « retourner » les variables (les rendre utilisable en dehors de la fonction : dans le reste du programme).

**Bilan et perspectives**

Je pense que si nous pouvions continuer notre projet j’aurai bien voulu essayer en premier de ea faire fonctionner avec Django, car je continue de croire qu’on pourrait y arriver. Django est utilisé par les développer des sites comme Instagram, la NASA… C’est vraiment un outil qui revient un peu à la mode aujourd’hui. Il simplifie beaucoup de taches. Notre projet Django aurai été vraiment « exportable », aussi bien à la Source qu’ailleurs, car c’est un outil professionnel. En continuant on pourrait aussi ajouter l’envoi de mail automatique aux éducatrices de la Source lorsque le programme est lancé par exemple. Pour finir, je pense que finalement de mon côté, ce n’est pas le code qui m’a passionnée mais les enjeux du projet. Réfléchir à comment résoudre les problème en groupe.

D’autre part, j’ai bien aimé une des dominantes de ce projet d’ISN qui est d’apprendre par soi-même. C’est un avantage des sciences du numérique qui permet une grande autonomie et une incroyable créativité. En effet, je préfère voir le codage comme un moyen pour développer des produits intéressant et innovateur qui répondent à de nouveaux problèmes.

**Annexe**

Lien Github :

• TA 2.0 : Exécutable pour accéder à au programme

• list.pickle : fichier enregistrant les listes du programme

• Visuels : fichier contenant les visuels en HTML que nous souhaitions utiliser