

**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.

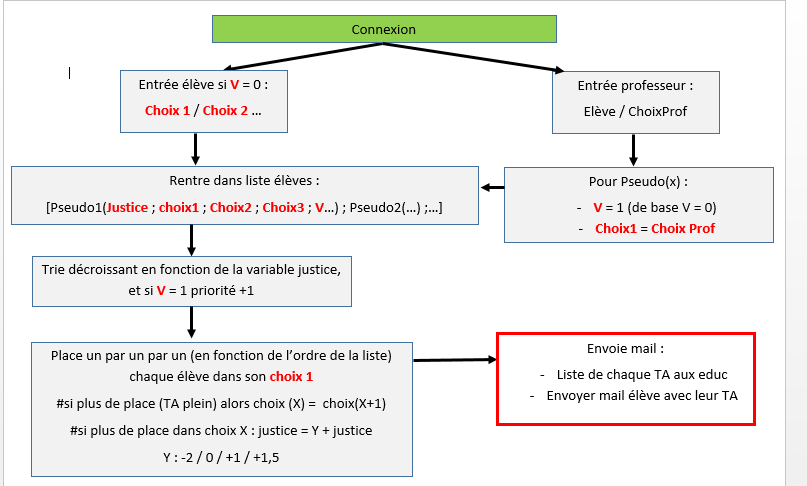
**But de départ**

**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 appuyés sur des sites d’apprentissages tels qu’OpenClassroom ou les ressources officielles de Django et python, deux langages très documentés :

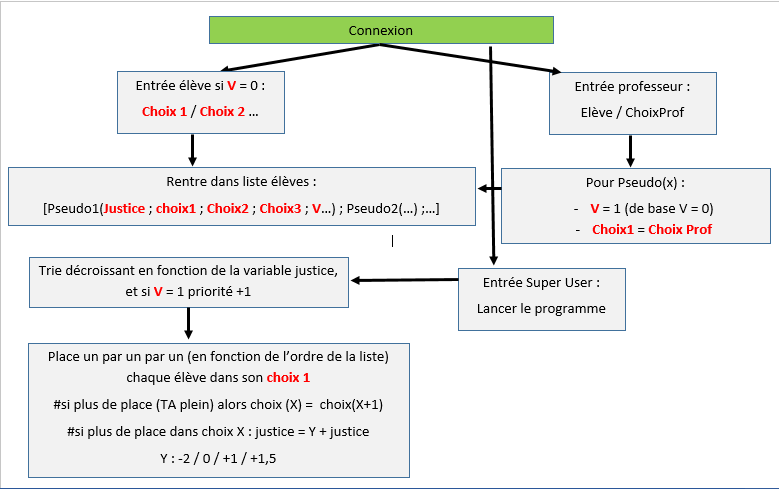
**Problèmes**

* 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**

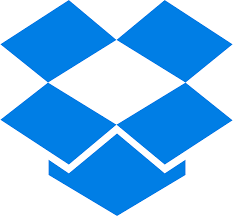


**Planning des taches 2**

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.







**Partie personnel de Thibault :**

**Première partie, utilisation de Django, Html/Css, sqlite3 :**

Notre projet initiale était le code d’une page ayant une interface graphique en HTML, j’ai donc commencé à apprendre via un site internet (OpenClassrooms) le codage en HTML/CSS(ces deux outils sont interdépendant pour créer une interface agréable : HTML sert pour la structure et CSS pour la mise en forme de celle-ci). Au fur et à mesure de l’apprentissage j’ai développé des “petits projets” intermédiaires pour finaliser une page de connexion simple avec une entrée d’utilisateur et mot de passe qui donne accès à la page élève ou la page professeur après avoir été traitée par Django.

Après avoir codé l’interface graphique nous devions tous les trois apprendre Django, l’apprentissage de cet outil passe par la création de page internet en local. J’ai donc suivis la formation en me concentrant sur ce dont nous avions réellement besoin pour gagner du temps et espérer pouvoir rendre un projet fonctionnel à temps. Il apparut au fur et à mesure que j’avais besoin de plus en plus de temps pour assimiler les connaissances nécessaires. En se répartissant les chapitres à apprendre nous avons pu réussir à finir l’apprentissage. Cependant il restait un problème de taille : le stockage des données.

L’utilisation de Django nécessite un accès à une base de données qui stocks les données en attendant qu’elles soient traitées par le “servomoteur”, c’est à dire le code qui classe les élèves et modifie leurs variable justice (codé en python). Ainsi j’ai appris l’utilisation des base de données en sqlite3, ce ne fut pas une tâche simple et a pris du temps. Il fallait coder plusieurs sortes de données : les données inchangeable comme le nom et le mot de passe des utilisateurs puis les variables dans les tables correspondantes afin de les utiliser et les modifier. De plus il fallut en raison du temps qu’il restait commencer le code du servomoteur en python, j’ai donc entamé le code de celui-ci ce qui ne fut pas une tâche facile étant donnée l’absence de base de données fonctionnel : il fallait créer des tableaux directement sur le code pour pouvoir le tester, sachant qu’à chaque fois que l’on lance le code les données d’avant n’était pas enregistrées.

Lorsque les résultats des bases de données commençaient à devenir satisfaisants nous avons réalisé que nous n’avions pas le temps de finaliser les bases de données pour ensuite pouvoir les traiter via Django. Nous avons donc décidé d’abandonner Django et la base de données SQL Lite

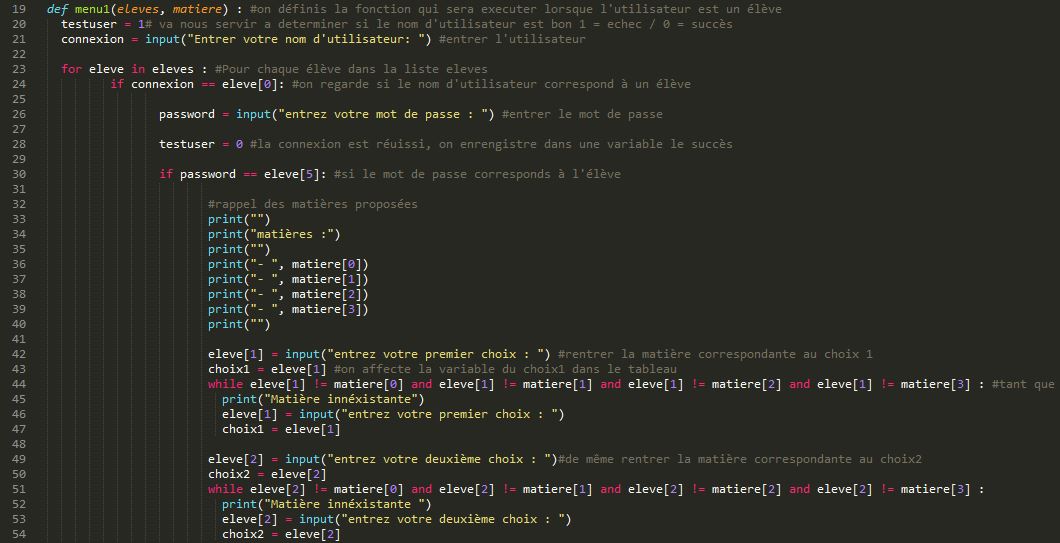
**Deuxième partie Python,Tkinter :**

Après la finalisation d’un servomoteur fonctionnel, il fallait élaborer une interface graphique simple grâce à Tkinter qui permet d’ouvrir une fenêtre avec différents boutons. Il fallut remodeler le code principale pour pouvoir le mêler à l’interface graphique en créant de nombreuse fonctions et de les assigner aux boutons. Il y a eu de nombreuses erreurs d’indentations et d’affectation de fonctions mais après une réunion à trois nous avons tout réglé sauf une chose : le bouton quitter ne quitte pas la fenêtre. Après beaucoup de recherche aucune solutions n’est apparue, les fonctions “quitter” ne fermait pas la fenêtre, une nouvelle remise en question est venue : le seul choix de lancer le programme après l’entrée des variables est d’avoir un bouton quitter (permettant de sauvegarder les données). L’autre choix est d’abandonner l’interface Tkinter.

**Détail d’une fonction : Menu1**

La fonction Menu1 permet de collecter les choix d’un élève selon un rang : il possède 4 choix de matières à hiérarchiser. Tout d’abord nous vérifions les noms d’utilisateur et mot de passe.

* On définit tout d’abord la variable “testuser” qui va nous permettre de vérifier selon sa valeur si le nom d’utilisateur est correct
* La boucle for permet de tourner pour le nombre d’élève selon la liste élève
* On demande à l’utilisateur d’entrer son nom contenu dans la variable “connexion”
* Si le nom existe dans la liste on demande à l'utilisateur de rentrer son mot de passe “password”, l’identifiant est correct donc “testuser” est égale à 0.
* Si le “password” correspond au mot de passe  de l’élève de la liste alors on commence à traiter les choix de l’utilisateur. On rappelle les matières proposées en TA pour éviter le nombre d’erreur simplement en les affichant



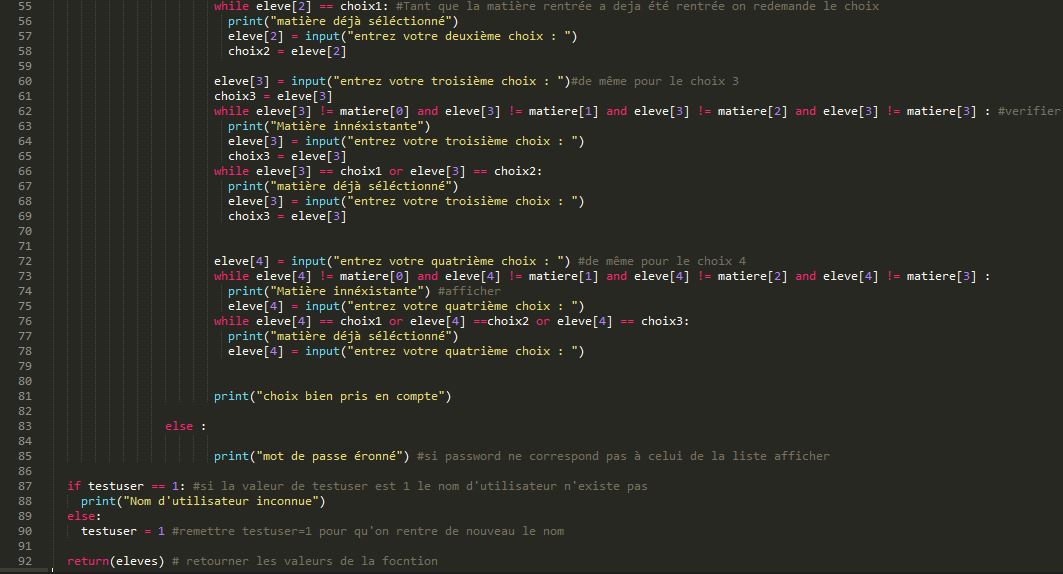
Ensuite nous traitons les choix de l’élève :

* Nous allons faire une succession de boucle “while” afin de rentrer les choix :

Tant que le choix est différent de toutes les matières proposées (l.44) nous affichons “choix inexistant (print). Puis re-demander de rentrer le choix jusqu'à ce qu’il existe.

⇒ Cette action est réitérée à chaque entrée de choix avec la même boucle “while”

* A partir du deuxième choix on introduit à la suite de la première boucle “while” une seconde boucle, qui tant que le choix a déjà été rentré demande de rentrer de nouveau un autre choix (exemple l.55)
* On fait les mêmes boucles jusqu’au choix4, en affectant lorsque la matière existe le choix correspondant dans la liste qui sert de base de donnée” (exemple l.54)
* Lorsque l’on est sorti  des boucles “while”, on “print” que le choix a été pris en compte
* le “else” (l.83) correspond au “if” vérifiant le mot de passe, il permet de “print” que le mot de passe est faux.
* si la variable “testuser” c’est que l’on est pas passer par la boucle et que le nom d’utilisateur n'existe donc pas.
* On réinitialise ensuite “testuser” à la valeur 1 pour passer à un autre élève.



**Bilan et perspectives**

Le projet de TA 2.0 en groupe m’a appris beaucoup de choses à commencer par les langages de programmations d’interface graphique comme je l’ai expliqué auparavant. Au-delà des connaissances accumulées le projet m’a permis de m’immiscer dans des apprentissages et dans des codages qui ne sont pas forcément de mon niveau, je n’avais pas écrit une ligne de code avant cette année ! Le fait de tout apprendre tout seul sur un site permet d’acquérir une certaine autonomie qui je pense sera bénéfique pour les études supérieurs.

Le projet de groupe et le partage des taches n’ont pas été simple mais ont aboutie à une belle cohérence au niveau du travail, nous avons pu rendre un dossier complet et malgré les échecs nous avons su rebondir.

Il y a différentes perspectives envisageable pour notre projet en effet on pourrait le mettre en réseau pour que les élève puissent s’inscrire depuis plusieurs ordinateurs ou encore sur internet pour qu’ils puissent le faire depuis leurs propre ordinateur. Il faudrait aussi le faire tester en réalité par exemple au lycée de La Source.

Nous pourrions faire une interface graphique ce qui était notre projet de départ, et qui manque à celui actuel. Cela permettra d’améliorer l’apparence de notre projet.

Pour finir je dirai que cette année d’ISN a su me donné gout au codage et à l’informatique, j’envisage d’ailleurs de poursuivre un apprentissage de la matière en école d’ingénieur l’année prochaine.

**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

**Comptes**

|  |  |  |
| --- | --- | --- |
| Type de compte | Nom d’utilisateur | Mot de passe |
| SuperUser | / | 123 |
| Elève | David | 123 |
| Elève | Thibault | 123 |
| Elève | Lazare | 123 |
| Elève | Elena | 123 |
| Elève | Elise | 123 |
| Elève | Violette | 123 |
| Elève | Pauline | 123 |
| Elève | Andrien | 123 |
| Elève | Vincent | 123 |
| Elève | Nicolas | 123 |
| Elève | Mathis | 123 |
| Elève | Raphael | 123 |
| Elève | Solena | 123 |
| Elève | Dana | 123 |
| Elève | Ines | 123 |
| Elève | Maxime | 123 |
| Elève | Quentin | 123 |
| Elève | Leo | 123 |
| Professeur | Léo (histoire) | 123 |
| Professeur | Delphine (anglais) | 123 |
| Professeur | Florence (maths) | 123 |
| Professeur | David (histoire) | 123 |