|  |  |
| --- | --- |
|  |  |

2020/2021 – ISEN 3 – Projet de Fin d’Année

RAPPORT D’ACTIVITE

(Département CSM)

|  |
| --- |
| **ISENINFO** |
| Groupe 10 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NOM Prénom étudiant 1 | DEMENEZ Hugo | CSI3 ✔ | CIR3 ☐ | CNB3 ☐ |
| NOM Prénom étudiant 2 | DANDOY Brieuc | CSI3 ✔ | CIR3 ☐ | CNB3 ☐ |
| NOM Prénom étudiant 3 | BENNOUNA Mohamed Amine | CSI3 ✔ | CIR3 ☐ | CNB3 ☐ |
| NOM Prénom étudiant 4 | GULLI Guillaume | CSI3 ✔ | CIR3 ☐ | CNB3 ☐ |
| Date de la soutenance | 29/06/2021 – 30/06/2021 | | | |

**Table des matières**

[1. Introduction 4](#_Toc74903677)

[1.1. Etat de l’art et problématique 4](#_Toc74903678)

[1.2. Solution envisagée 5](#_Toc74903679)

[1.2.1. Présentation de la solution 5](#_Toc74903680)

[1.2.1.1. Exposition de la proposition 5](#_Toc74903681)

[1.2.1.2. Périmètre fonctionnel 6](#_Toc74903682)

[1.2.1.3. Choix techniques (matériels et logiciels) 7](#_Toc74903683)

[1.2.1.4. Pilotage du projet 8](#_Toc74903684)

[1.2.1.4.1. Estimation des charges 8](#_Toc74903685)

[1.2.1.4.2. Planning prévisionnel 10](#_Toc74903686)

[1.2.1.4.3. Communication 11](#_Toc74903687)

[1.2.1.5. Coûts 12](#_Toc74903688)

[1.2.2. Analyse fonctionnelle 13](#_Toc74903689)

[1.2.2.1. Détail des fonctionnalités 13](#_Toc74903690)

[1.2.2.2. Maquettes 15](#_Toc74903691)

[1.2.3. Analyse technique 15](#_Toc74903692)

[1.2.3.1. Architecture technique (interfaçage entre les composants, flux…) 15](#_Toc74903693)

[1.2.3.2. Modélisation de la data (données, flux…) 15](#_Toc74903694)

[1.2.4. Réalisation 15](#_Toc74903695)

[1.2.5. Tests 15](#_Toc74903696)

[1.2.6. Documentation 15](#_Toc74903697)

[1.3. Résultats obtenus 15](#_Toc74903698)

[1.3.1. Difficultés rencontrées et solutions 15](#_Toc74903699)

[1.3.2. Respect des délais 15](#_Toc74903700)

[1.3.3. Respect du périmètre fonctionnel 15](#_Toc74903701)

[1.3.4. Mise en production 15](#_Toc74903702)

[1.4. Conclusion 16](#_Toc74903703)

[1.4.1. Montée en compétences de l’équipe 16](#_Toc74903704)

[1.4.2. Valeur ajoutée pour le client (Junia ou autre) 16](#_Toc74903705)

[1.4.3. Axes d’amélioration 16](#_Toc74903706)

[1.4.4. Et si c’était à refaire (prise de recul) ? 16](#_Toc74903707)

# Introduction

## Etat de l’art et problématique

Nous sommes 4 étudiants de l’école ISEN Lille. Depuis 3 ans, nous utilisons l’outil Aurion pour consulter notre planning, nos notes et parfois consulter nos bulletins.

Après chaque cours, nous nous connectons pour consulter la salle dans laquelle nous devons nous rendre. Cela implique de sortir notre téléphone, ouvrir le navigateur web, aller sur le site Aurion, entrer notre identifiant et notre mot de passe, cliquer sur le bouton pour se connecter, puis chercher notre planning sur le site.

Force est de constater que ces étapes sont une perte de temps conséquente dans la vie d’un(e) étudiant(e).

De surcroit, nous devons suivre la même démarche pour consulter nos notes.

Il est donc urgent de proposer un outil complémentaire à Aurion, pour réduire cette perte de temps au maximum.

## Solution envisagée

### Présentation de la solution

#### Exposition de la proposition

Avec notre solution, vous devrez ouvrir votre navigateur une seule fois par semaine.

Notre solution vous permet de recevoir par mail votre planning tous les dimanches pour ensuite le mettre dans votre Google Agenda.

De plus, vous recevrez une notification par mail dès qu'une nouvelle note est disponible où s’il y a eu un changement dans votre planning.

Pour cela, il faudra vous inscrire sur notre site iseninfo.fr.

Nous avons également mis en place une page d'accueil, avec certaines fonctionnalités, lorsque vous vous connecterez pour avoir des informations complémentaires.

#### Périmètre fonctionnel

Le site possède de nombreuses fonctionnalités :

* Fournir des informations météorologiques par rapport à la localisation des locaux du campus ISEN de Lille.
* Afficher un graphique reprenant toutes les notes de la scolarité de l’élève pour en voir l’évolution.
* Mettre à la disposition des utilisateurs un ChatBot\* pour donner l’emploi du temps de l’utilisateur ainsi que ses notes les plus récentes.
* Utiliser la reconnaissance vocale pour accéder à des informations similaires à celles fournies par le ChatBot.
* En complément du site, une infrastructure PYTHON gère l’envoie d’un mail avec le planning, ainsi qu’une notification dès lors qu’une nouvelle note apparaît sur Aurion ou qu’il y a un changement dans le planning de la semaine.
* Nous souhaitions ajouter le calendrier sans avoir à solliciter l’utilisateur cependant cette tâche est impossible à réaliser, puisqu’avec un fichier ICS les fuseaux horaires changeaient en fonction de la saison et les dates étaient donc corrompues. Nous avons donc contourné notre problème en utilisant un fichier CSV, l’utilisateur doit simplement l’ajouter à son calendrier type GoogleAgenda.

*\*ChatBot : assistance textuel, programmée pour répondre aux questions des utilisateurs*

#### Choix techniques (matériels et logiciels)

Nous avons décidé d’utiliser différents langages de programmation afin de pouvoir compenser les faiblesses de chaque langage.

Nous avons codé en **PYTHON** les parties correspondant à des algorithmes et des scripts car c’est un langage extrêmement flexible qui gère facilement l’analyse des données.

Nous avons également utilisé le **PHP**. Il permet de gérer la connexion entre les différentes pages HTML, et nous permet d’enregistrer les informations des utilisateurs pour pouvoir ensuite interagir avec la base de données.

Pour stocker les différentes informations récoltées, nous avions décidé d’utiliser **SQLite** qui permet d’avoir une base de données légère sous forme de fichier local. Cependant, nous sommes passées en **MySQL** afin de stocker les informations sur un serveur et cela permet également à plusieurs personnes de consulter les informations de la base de données.

De plus, l’interface web a été réalisé en **HTML** qui permet de structurer le contenu d’unepage et en **CSS** qui permet de définir le style de la page.

Pour dynamiser notre interface web, nous avons choisi d’employer du **JAVASCRIPT** ce qui permet de faire des animations et de l’interactivité.

Pour finir, notre projet fonctionne sur un serveur **LINUX** pour qu’on puisse accéder au serveur tous les jours et toutes le heures.

#### Pilotage du projet

##### Estimation des charges

Nous nous sommes efforcés à garder une répartition équitable des tâches.

Grâce à un tableur Excel, il était possible de compter le nombre de tâche que chacun d’entre nous effectuait.



##### Planning prévisionnel

##### Communication.

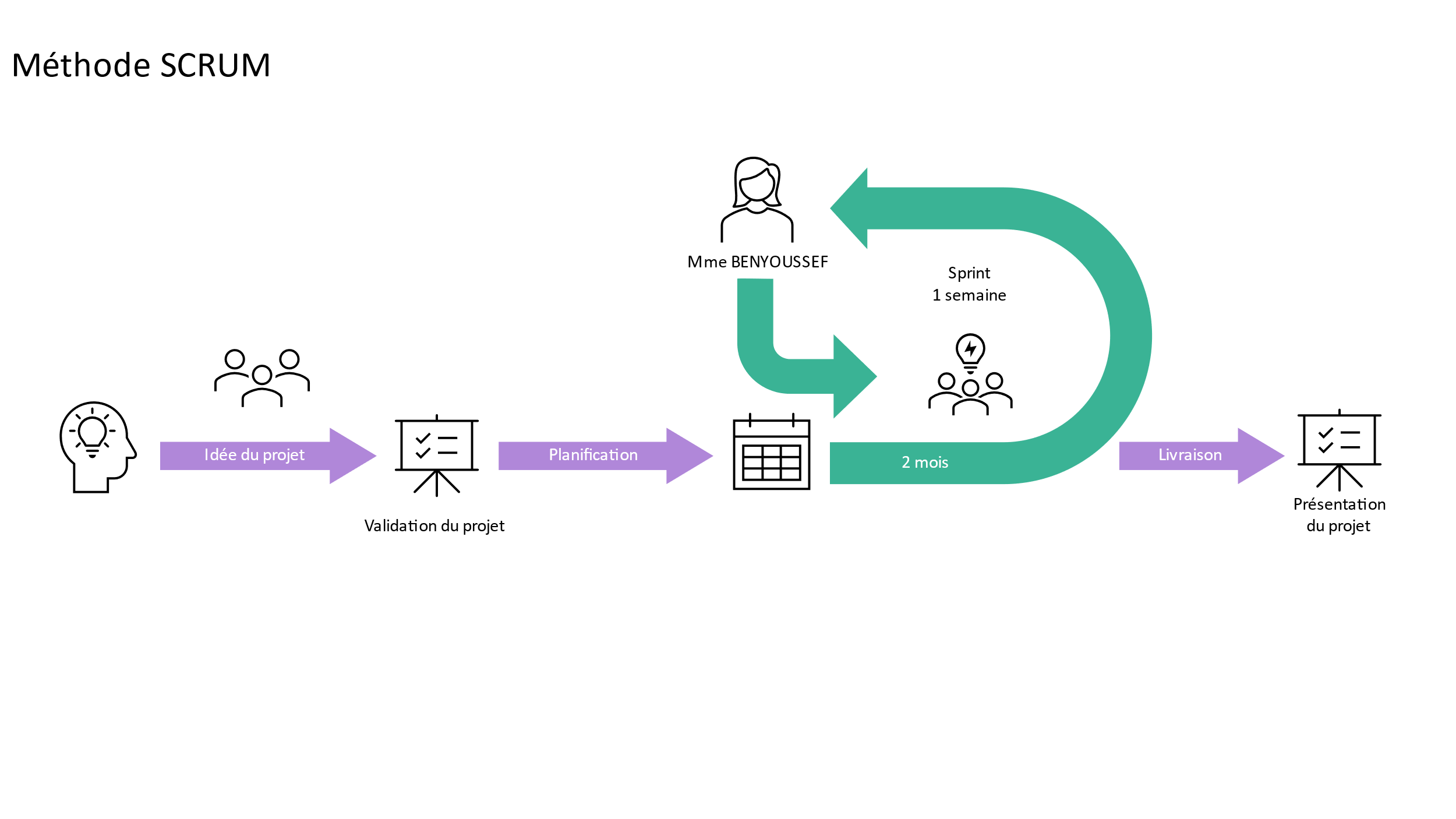
Les premiers jours ont permis d’établir les différentes tâches à effectuer, de réaliser un diagramme de Gantt ainsi qu’une To Do List (liste des choses à faire, en cours ou terminées).

Puis, nous avons souhaité mettre en place des réunions le plus souvent possible pour pouvoir avancer ensemble et demander de l’aide en cas de problèmes. Les réunions étaient organisées sur Teams tous les matins et tous les après-midis s’il n’y avait pas d’empêchement.

Au début de chaque journée, nous faisions un point sur l’avancement par rapport à la dernière réunion et à la fin de chaque semaine, nous faisions un point plus global grâce à un résumé des différentes activités de chaque personne sur un document word.

En plus des réunions quotidiennes, un répertoire GitHub a été créé pour qu’on puisse partager nos codes et avoir un suivi de l’avancement du projet.

Pour augmenter l’efficacité du projet, nous avons mis en place une méthode agile, la méthode SCRUM. Chaque semaine, nous effectuions des « Sprints » et à la fin de ces « Sprint », nous présentions ce que chaque personne a ajouté au projet.



#### Coûts

Vous trouverez ci-dessous deux devis :

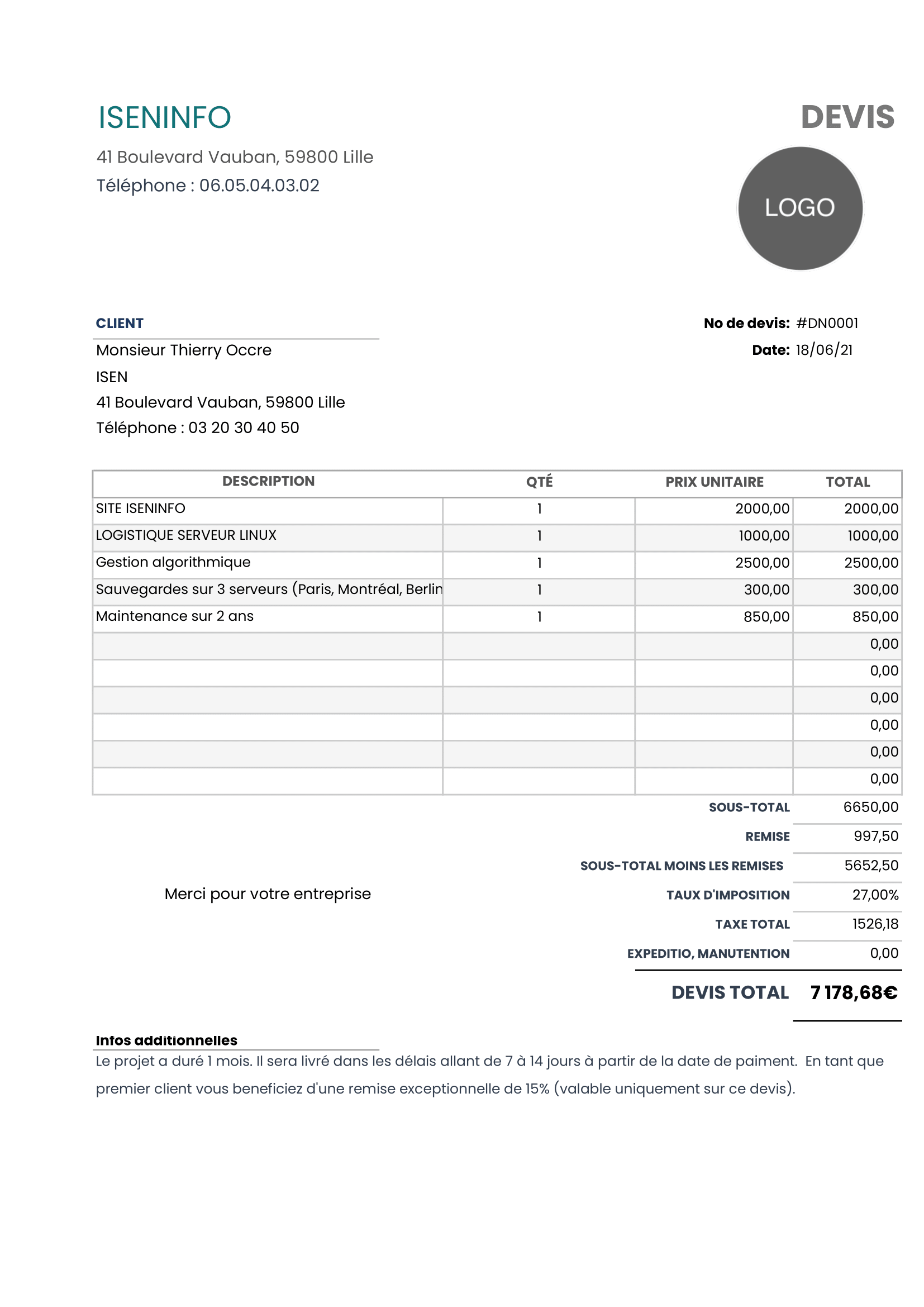
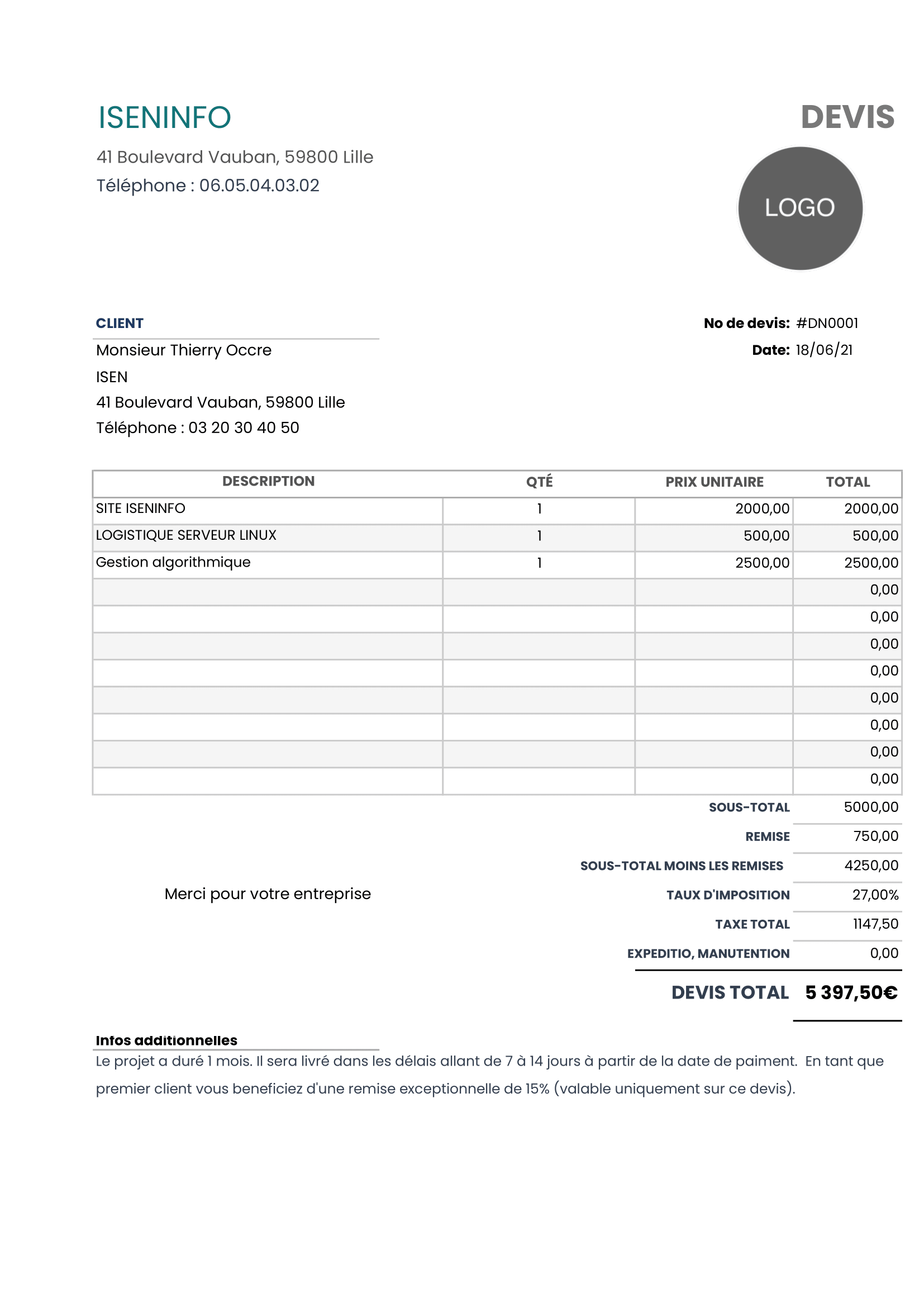
* Le premier est le plus abordable étant donné que le choix de l’hébergement est basique. Cependant il est moins sécurisé et donc plus sensible au piratage, et à la perte des données.
* Le second dispose d’un serveur sécurisé et d’un système de sauvegarde sur différents serveurs, il est donc plus onéreux.

Le SITE IsenINFO est vendu avec ses droits d’auteurs, ainsi que son nom de domaine. En faisant cet achat, vous bénéficiez du serveur linux mais également de toutes les ressources présentes sur ce dernier. C’est-à-dire : page PHP, script JavaScript et fichier CSS.

Le serveur LINUX est actuellement hébergé par un grand nom de l’hébergement. Pour engager l’économie française, nous avons choisi **OVH** qui est une société du Nord de la France, basée à Roubaix. Le serveur est localisé sur le site de Gravelines. Ainsi, la connexion au serveur est excellente pour les étudiants de l’ISEN Lille.

Enfin, nous vous vendons tous les algorithmes pythons qui vous permettront de gérer l’envoie des mails, la création du planning au format CSV et la population de la base de données.

Si vous choisissez le deuxième devis, vous bénéficiez d’une sécurité accrue grâce aux systèmes de sauvegarde mais également d’un accompagnement et d’une maintenance de la part de notre équipe d’une période de 2 ans.



### Analyse fonctionnelle

#### Détail des fonctionnalités

Notre site affiche, en haut à gauche, les informations météorologiques suivantes :

* La température actuelle de la localisation de ISEN LILLE
* Une icône dynamique qui change selon la météo indiqué grâce à une API.

Au milieu de la page, vous pouvez voir un graphique représentant l'évolution des notes depuis que vous êtes à ISEN. Sur le graphique, vous pourrez retrouver vos différentes notes ainsi que les dates associées. Cela est possible grâce à un programme PYTHON de Webscraping des notes sur Aurion qui les envois dans une base de données MySQL. De là, on les récupère pour faire un graphique grâce au JAVASCRIPT.

Nous avons eu l'idée d'ajouter un Chatbot afin que l'utilisateur puisse profiter d'une nouvelle expérience et cela permet d'obtenir d'autres services. Notre Chatbot pourra par exemple afficher le planning de la semaine, la dernière note obtenue ou les 5 dernières notes. L'utilisateur n'aura qu'à ouvrir la Chatbox en cliquant sur l'icône en bas à droite puis le bot posera une question et l'utilisateur choisira sa réponse en cliquant dessus. Le Chatbot a été codé en JAVASCRIPT pour la partie dynamique, en HTML pour la partie "dialogue" et en CSS pour la partie esthétique.

La reconnaissance vocale permet d'accéder à des informations similaires à celles fournies par le Chatbot. Elle a été réalisée en JAVASCRIPT puis intégrée sur le site grâce au PHP. Elle utilise l’api Webspeech qui permet de transformer un message vocal en texte. Ensuite nous allons chercher des mots clés dans le texte pour deviner les instructions de l’utilisateur. Pour cela nous utilisons des expressions régulières.

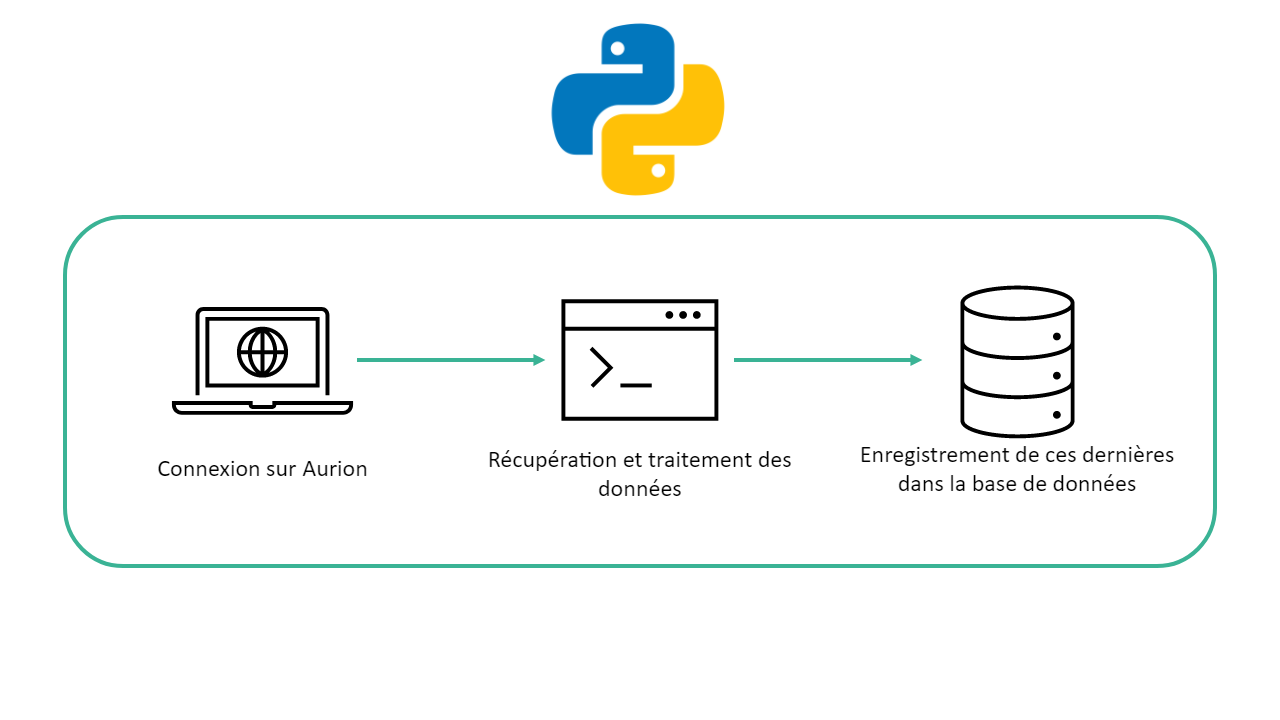
Une infrastructure PYTHON gère l’envoie d’un mail avec le planning, ainsi qu’une notification dès lors qu’une nouvelle note apparaît sur Aurion ou qu’il y a un changement dans le planning de la semaine. Pour cela, il faut commencer par un Webscraping du planning de la semaine et des notes. Puis, chaque dimanche, un programme PYTHON va envoyer un mail avec le planning de la semaine sous format .csv et il faudra le mettre dans votre calendrier Google. De plus, toutes les heures, un autre programme PYTHON va regarder si la nouvelle liste d'informations correspond à la liste précédente et si ça n'est pas le cas, vous allez recevoir un mail avec soit le planning modifié soit une notification pour vous indiquer qu'une nouvelle note est disponible.

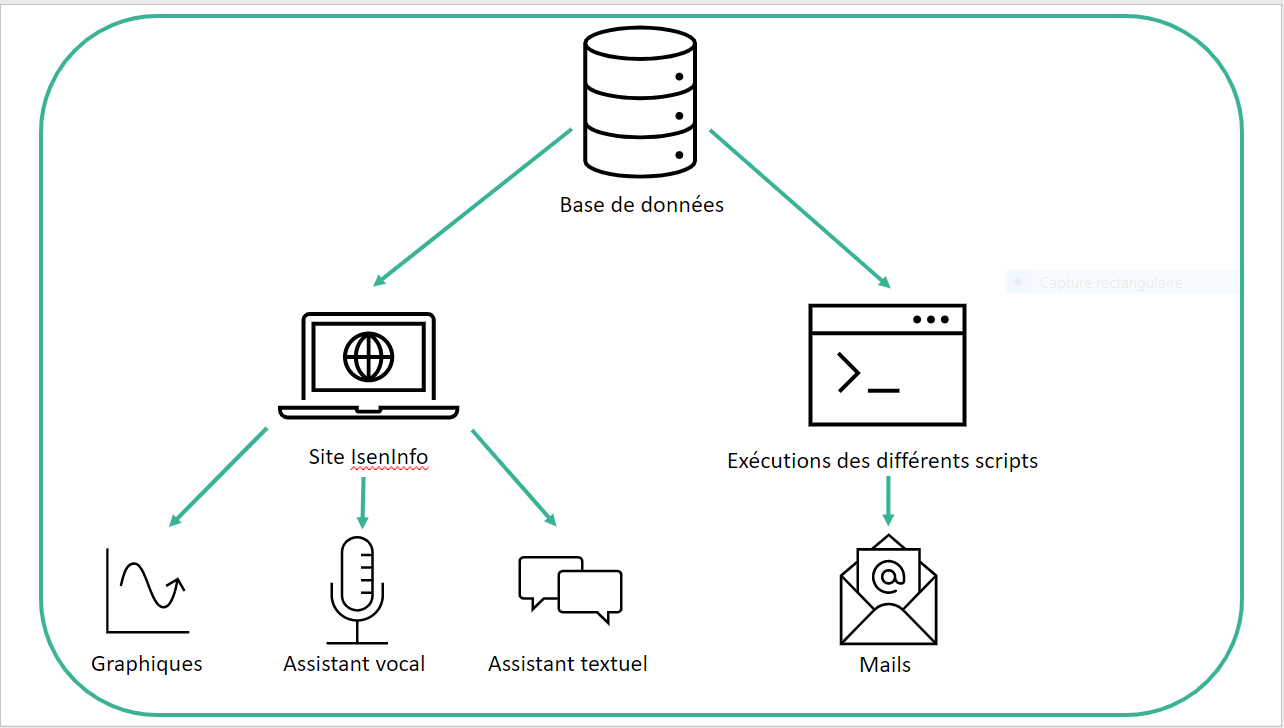
#### Maquettes

### Analyse technique

#### Architecture technique (interfaçage entre les composants, flux…)

Un bon schéma vaut mieux qu’un long discours…





#### Modélisation de la data (données, flux…)

Idem (UML ? Merise ? ...)

### Réalisation

Exposez dans cette partie tout ce que vous avez mis en œuvre pour traduire l’analyse technique et fonctionnelle en solution concrète, et surtout comment et pourquoi vous l’avez fait.

Si vous souhaitez exposer du code source, il n’est pas nécessaire de détailler tout le code, mais de mettre en exergue ce qui est important et remarquable (au sens « Qui est susceptible d'être remarqué, d'attirer l'attention ») pour vous.

### Tests

Mettez à profit la conférence « sensibilisation au test logiciel » du 01/06/2021.

En termes de tests, chaque personne était responsable de faire les tests unitaires de son code. Ensuite, une fois sûr que le code était fonctionnel, ce dernier était intégré au projet principal puis nous effectuions un test d’intégration du code.

### Documentation

Mentionnez la documentation que vous avez éventuellement produite : Guide utilisateur, guide du développeur, guide d’installation, aide en ligne…

## Résultats obtenus

### Difficultés rencontrées et solutions

### Respect des délais

### Respect du périmètre fonctionnel

### Mise en production

Votre solution est-elle ou va-t-elle être utilisée ?

## Conclusion

### Montée en compétences de l’équipe

Qu’ai-je appris individuellement ? Qu’ai-je appris en équipe ? Techniquement ? Humainement ? Financièrement ? Etc.

### Valeur ajoutée pour le client (Junia ou autre)

Qu’est-ce que votre projet et votre expertise ont apporté au client ?

### Axes d’amélioration

### Et si c’était à refaire (prise de recul) ?

Auriez-vous fait de la même manière ? Qu’auriez-vous changé ?