Rapport final de Projet Pluridisciplinaire d'Informatique Intégrative

Les jardins partagés

Alexis MARCEL
Lucas LAURENT
Noé STEINER
Mathias AURAND-AUGIER

Responsable du module :
Olivier FESTOR
Anne-Claire HEURTEL
Gerald OSTER

6 Janvier 2023

Contents

1 Base de donnée

1.1 Conception

1.1.1 Que faut-il dans notre base ?

Pour la base de donnée, nous aurons besoin de stocker plusieurs informations. Nous aurons besoin en premier lieu de stocker toutes les données concernant l'utilisateur (les informations de son compte à savoir son identifiant ou son mot de passe par ex), nous aurons également de stocker des données concernant les jardins que les utilisateurs vont créer (comme l'endroit où il se trouve par exemple, le nombre de personne qui en font partie). Lorsque les jardins seront crées, nous aurons besoin de caractériser les parcelles de jardin (les légumes qu'on y cultive ou encore l'état dans lequel elle est : a t'elle été labouré). Afin d'avoir une vue d'ensemble sur les différentes missions au sein des jardins, nous aurons également besoin de caracteriser les differentes tâches à faire (date limite, la personne qui s'en occupera...).

1.1.2 Schéma entité-association

Maintenant que nous savons ce que nous aurons à stocker, nous ponvons commencer la réalisation de notre schéma entité-asociation en respectant les contraintes logiques de cardinalités suivantes :

- Une personnes peut posséder un ou plusieurs jardins mais un jardin ne peut avoir qu'un propriétaire
- Un jardin peut avoir plusieurs parcelles mais une parcelle est completement associé à son jardin.
- Une personne peut faire autant de tache qu'elle veut, et de même, plusieurs personnes peuvent participer à une même tache.

Ainsi, on obtient le schéma suivant :

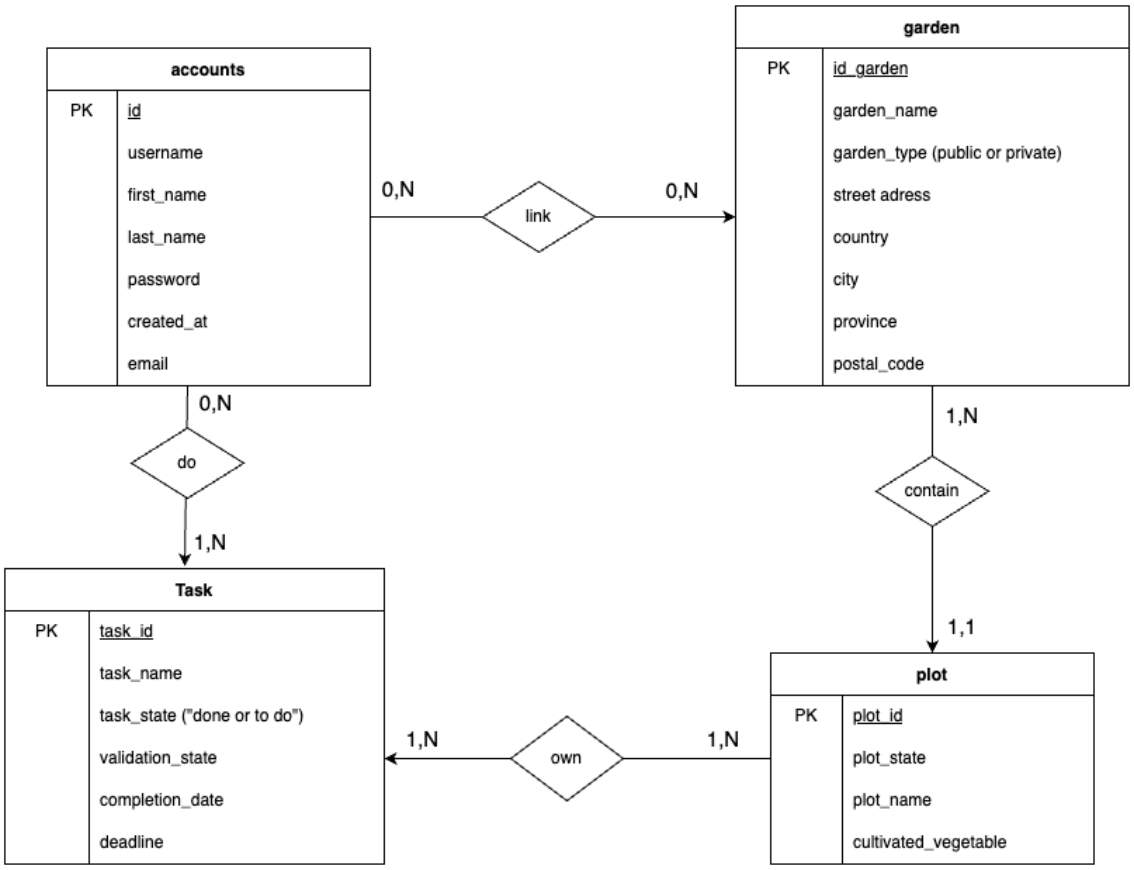

Figure 1: Schéma entité-association

1.1.3 Mise en 3ème forme normale

Pour finaliser la conception de notre base, il faut maintenant transformer les entités et associations décrites dans le schéma en relation, que nous mettrons ensuite sous troisieme forme normale : on obtient ainsi les relations suivantes :

- account(id, email, username, first name, last name, password, created at)
- garden(garden id, owner, garden name, manager, garden type, street adress, country, city, province, postal code)
- plot(plot id, garden id, plot state, plot name, cultivated vegetables)
- task(task id, plot id, task name, task description, task manager, task state, completion state, validation state, deadline)
- plot unit(plot id, unit)
- do(account id, task id)
- link(account id, garden id)

1.2 Implémentation

1.2.1 SQL alchemy

SQLAlchemy est un ORM (Object-Relational Mapping) permettant de manipuler la base de données via des objets python. Les requêtes en python sont ainsi "traduites" en SQL et la réponse reçu se présentera sous la forme d'un objet python avec lequel on peut interagir. SQLAlchemy constitue donc un pont entre la base de données et notre application. Pour que l'ORM puisse savoir où se situe notre base de données (pour pouvoir l'interroger), on doit définir des modèles. On doit ainsi lier les éléments de notre table avec des classes python dans lesquels on indiquera le type de chaque champ. Les sessions de SQLAlchemy permettent de gérer les transactions SQL, autrement dit un ensemble de requêtes. Si l'une d'elles échoue, l'ensemble de la transaction est annulée et aucune requête n'est communiquée à la base. L'avantage de ce système est la sécurité. Par exemple, si on crée un utilisateur et que le requête permettant la création des propriété de l'utilisateur dans une autre table (mot de passe) échoue,l'ensemble des requêtes est alors annulé et la base est corrompu par un utilisateur sans permission.

1.2.2 Création de la base

En utilisant le systeme de gestion de base de données sqlite, nous avons crée notre base dans un fichier data.db avec le script sauvegarder dans un fichier nommée "creation table.sql". Ces deux fichiers sont dans le dossier Data situé à la racine du Backend.

Dans le dossier models, situé à la racine du backend, nous avons crée les modèles, les classes python associé aux élements de la base. La connection à la base de donnée via SQLAlchemy est codé dans fichier intitulé bdd.py

2 Implémentation Partie Web

2.1 structure de l'application

Nous avons choisi de séparer la partie affichage coté utilisateur (Frontend) et la partie gestion de l'application coté serveur (Backend). Les deux communiquent ensembles via une API. Dans le backend, se trouvent les differentes fonctionnalités se déroulant coté serveur tel que : l'authentification ou encore la gestion de la base de données. Le backend est un serveur flask(et les fonctions sont donc codé en python) Dans le front end, on utilise une librairie Javascript appelé React.

2.2 Fonctionnalités de l'application

2.2.1 Authentification

Pour accèder aux fonctionnalités de l'application, une insciption est nécessaire. Nous avons ainsi besoin de creer une Authentification. L'utilisateur doit d'abord se créer un compte. Par la suite, si l'utilisateur veut se connecter, il devra rentrer les informations données lors de son inscription.

Nous avons ainsi séparé l'Authentification en plusieurs parties : les trois premieres situé dans le fichier Authentification.py situé la dossier Route (racine backend) et la dernieres dans le fichier auth.py situé dans middlewares.

- La fonction signup (inscription) récupère dans un permier temps les données rentrées par l'utilisateur. Ensuite, la fonction verifie si l'email est déjà enregister. Si il l'est alors l'utilisateur ne peut pas s'inscire. Sinon, la fonction hache le mot de passe (elle le crypte), avant de crée un nouveau compte. Pour finir, la fonction renvoie une réponse avec un cookie (jeton JWT).
- La fonction signin (connexion) récupère l'e-mail et le mot de passe d'un utilisateur, vérifie si l'utilisateur existe et si le mot de passe est correct, et si c'est le cas, il renvoie un cookie avec un jeton JWT.
- La fonction signout (déconnexion) déconnecte l'utilisateur en supprimant le cookies.
- La fonction authtest utilise la fonction authentificate se trouvant dans le fichier auth.py du dossier middlewares. Cette fonction récupère d'abord le token se trouvant dans le cookies (envoyé par la fonction signin ou signup), et verifie si le token existe avant de donner l'accès à l'application.

2.2.2 Gestion de compte et présentation des jardins

Dans le backend, la gestion du profil est dans le fichier appelé profile.py. Cette gestion se décompose en plusieurs fonctions :

- La fonction get information récupère les informations de la base de données et les renvoie sous la forme d'un objet json, la récupération de la photo de profil se fait via la fonction get image, qui va chercher la photo dans le dossier static du backend.
- La fonction modify profile permet à l'utilisateur de modifier son profil. Cette fonction récupère d'abord le tuple de la base associé à l'utilisateur en vue de le modifier. La fonction récupère ensuite les informations tapés par l'utilisateur à condition qu'elles existent, le tuple est ainsi modifié avec les nouvelles informations avant d'être réinjecté dans la base.

2.2.3 Gestion des jardins : creation, destruction, modification, management

Dans le backend, la gestion des jardins s'étend sur 2 fichiers. Au sein du fichier garden.py se trouve les fonctionnalités suivantes :

- La fonction get all garden : permet de montrer à l'utilisateur les jardins qu'il possède. Elle récupère les jardins associés à l'id de l'utilisateur (en effectuant une jointure entre 2 tables), elle utilise ensuite une fonction (issue du dossier lib, fichier garden.helper) qui va renvoyer les informations concernant chaque jardin sous la forme d'objet json.
- La fonction create permet de créer un jardin. Elle récupère dans un premier temps toutes les informations rentrés par l'utilisateur sur son jardin (photo, nom...). Les informations sont ensuite mise dans la base et la photo dans le dossier Static.
- La fonction

2.3

2.3.1

2.3.2

2.3.3

2.4 Test et performance

2.4.1

2.4.2

2.4.3

3 Algorithme

3.1 Principe de l'algorithme

3.2 Implémentation

3.3 Analyse en complexité

3.4 Test de validité de l'algorithme

3.5 Analyse de performance

4 Gestion de projet

4.1 Équipe de projet

Ce projet est un projet local réalisé en groupe de 4 personnes :

- Alexis MARCEL
- Lucas LAURENT
- Noé STEINER
- Mathias AURAND-AUGIER

Le comité de pilotage est constitué de :

- Anne-Claire HEURTEL
- Olivier FESTOR
- Gérald OSTER

Ces personnes constituent les parties prenantes de notre projet ainsi que les acteurs influents sur le livrables.

Ceux qui...	*Sont...*
... Demandent, financent le projet, propriétaires du livrable	Télécom Nancy.
... Pilotent le projet et assurent la coordination avec le client	Les professeurs gérants les différents groupes.
... Réalisent le projet	Les élèves de l'équipe projet.
... Sont également concernés	Les communautés derrière les jardins partagés/circuits-courts.

Figure 2: Parties prenantes

4.2 Organisation au sein de l'équipe projet

Nous avons réalisé plusieurs réunions, en présentiel dans les locaux de Télécom Nancy mais également sur en visio-conférence sur Discord. Ces réunions nous ont permis de mettre en commun nos avancés régulièrement, de partager nos connaissances sur des problématiques et de nous organiser de manière optimale. Les comptes rendus des réunions réalisés sont présents dans l'Annexe 1.

De plus, dès le début de notre projet nous avons mis en place un projet Trello. Trello est une application permettant d'organiser facilement un projet en reposant sur une organisation en planches listant des cartes, chacune représentant des tâches. Ces tâches peuvent ensuite être déplacées permettant de découper notre projet en plusieurs jalons dynamiquement.

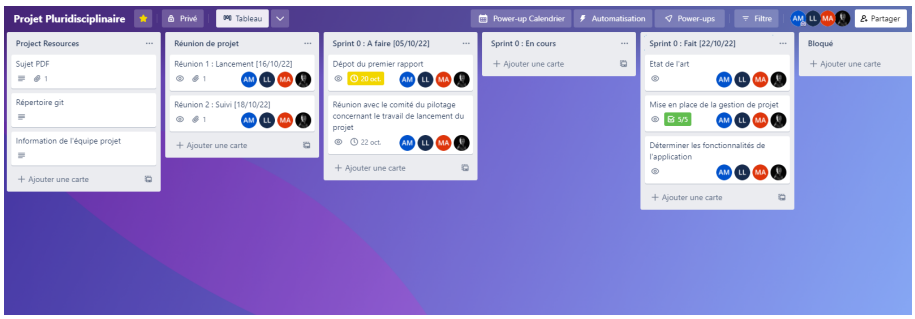

Figure 3: Organisation Trello

Ensuite, nous avons utilisé GitLab pour gérer les différentes versions du développement de notre application, ainsi que les différentes branches nous permettant de travailler simultanément sans conflit.

Enfin, la rédaction des differents comptes rendu de réunion et des rapports ont été rédigé en LaTeX.

4.3 Objectifs SMART

La méthode SMART que l'on rappelle ci-dessous nous a permis de définir nos différents objectifs :

	Critère	Indicateur
S	Spécifique	L'objectif est clairement défini.
M	Mesurable	On peut suivre et quantifier la progression de l'objectif.
A	Atteignable	L'objectif prend en compte la capacité des membres du projet à l'atteindre et des moyens mis à disposition.
R	Réaliste	L'objectif doit être réaliste, réalisable et pertinent par rapport à la situation.
T	Temporellement défini	Le projet doit être limité dans le temps, avec une date de fin.

Figure 4: Objectif SMART

4.4 Matrice des objectifs

Nous avons conçu, à l'aide de la méthode SMART, la matrice des objectifs suivante :

	LIVRABLE	DOCUMENTATION	VALORISATION DU PROJET	ACQUISITION DE COMPETENCES / FORMATIONS
INSUFFISANT	Créer un nouveau compte et utiliser l'application	Pas de recherche effectuée	Projet approuvé par tous les membres	Acquis superficiels
RÉUSSITE ACCEPTABLE	Reproduire son jardin sous forme numérique	Recherche sur les jardins partagés et les circuits-courts ainsi que ses possibilités	Conception du projet validé	
BON TRAVAIL	Créer un nouveau jardin et l'ouvrir aux autres	Étude des applications de jardins partagés		Être capable de réaliser une autre application similaire
EXCELLENT	Participer et gérer un jardin privé ou partagé	Tableau comparatif des applications étudiées		Avoir acquis intégralement toutes les notions utilisées durant le projet pour l'ensemble du groupe

Figure 5: Matrice des objectifs

4.5 Triangle qualité-cout-délai

Afin d'établir des objectifs cohérents, et réalisables dans les délais, nous avons réalisé le triangle qualité-coût-délai. On remarque ainsi, les délais étant courts, que nous avons tout intérêt à ne pas se fixer des objectifs trop ambitieux sous peine de devoir renoncer à certaines fonctionnalités et de ne pas rendre le livrable annoncé initialement.

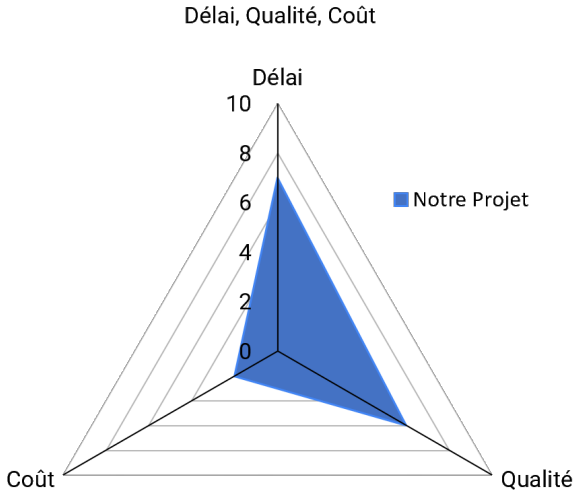

Figure 6: Triangle DQC

4.6 Matrice SWOT

Afin d'avoir une vision plus globale de nos ressources et des facteurs interne et externe agissant sur le projet, nous avons ensuite réalisé la matrice SWOT (Strengths, Weaknesses, Opportunities, Threats) de notre projet.

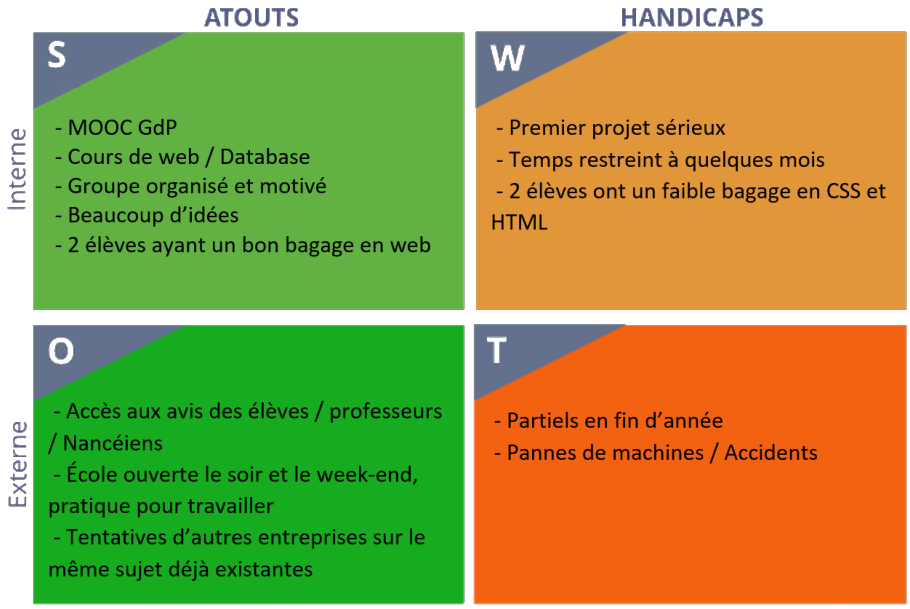

Figure 7: Matrice SWOT

On peut ainsi remarquer que notre projet présente de nombreux points fort notamment grâce aux connaissances acquises lors des cours de Télécom Nancy mais également de part l'expérience forte de deux des membres de l'équipe projet qui ont déjà réalisé des applications similaires. Cependant, plusieurs facteurs internes constituent nos faiblesses notamment les courts délais qui nous oblige à être concis et efficaces dans notre travail, ou encore le faible bagage informatique de deux des membres de l'équipe. Néanmoins, ces lacunes constituent pour eux l'opportunité d'apprendre, et de progresser avec l'aide des membres expérimentés de l'équipe.

De plus, nous devons anticiper les charges de travail dans le cadre de notre formation à Télécom Nancy qui s'avèrent être plus élevée en décembre lors des partiels de fin d'année. Nous allons donc devoir prendre cela en compte dans notre gestion des tâches.

4.7 Profil de projet

Afin d'avoir une vision plus globale sur notre projet, nous avons également réalisé le profil du projet (le budget étant égal à 0, nous avons choisi de ne pas le représenter dans notre profil). On remarque que, du fait des nombreuses fonctionnalités que nous avons l'intention d'implémenter dans notre application, que notre projet est de taille moyenne mais de complexité élevée.

Cependant, les enjeux du projet ne sont pas très importants (en dehors de la note finale qui compte dans notre moyenne) car l'échec du projet n'engendra pas la chute d'une organisation et le budget est négligeable.

De plus, au vu de l'état de l'art établi, l'innovation du projet est importante puisque nous avons choisi de combiner différentes fonctionnalités existantes de plusieurs applications et d'en rajouter de nouvelles.

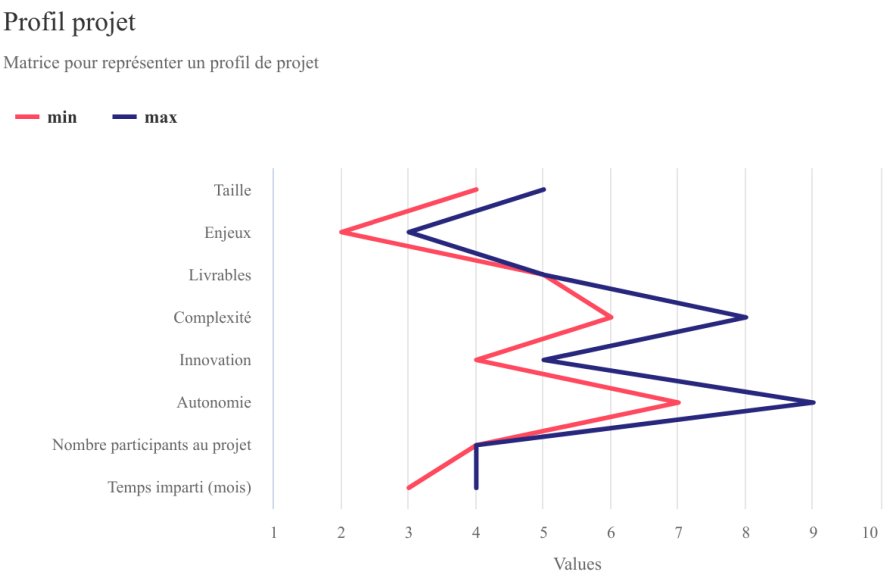

Figure 8: Profil du projet

4.8 WBS : comment concrétiser l'application

Ceci étant fait, nous avons maintenant choisi de détailler les lots de travail à effectuer pour fabriquer notre application. Nous avons ainsi réalisé le WBS (Work Breakdown Structure) de notre application : il apparait ainsi les grandes étapes de notre projet que sont : definition du cadre de l'application, développement des fonctionnalités de l'application et écriture du rapport.

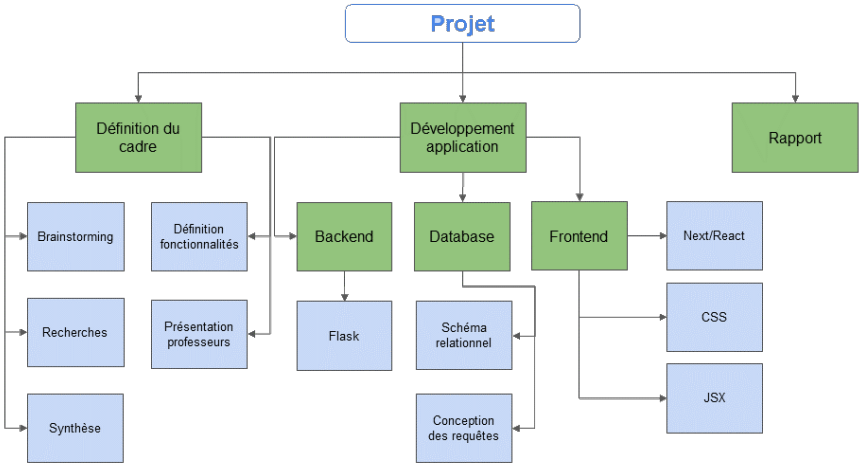

Figure 9: WBS

4.9 Diagramme de Gantt : planification

Maintenant que nous avons un détail des lots de travail qui constitue notre application, il faut maintenant les mettre en relation pour créer un planning efficace où chaque tâche est effectuée dans l'ordre.

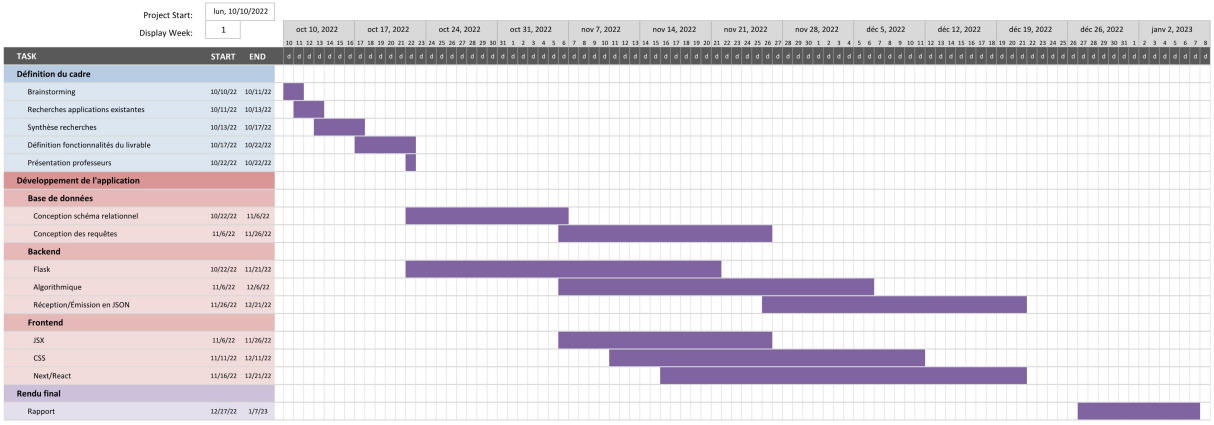

Figure 10: Diagramme de GANTT

Ce diagramme est une première version générale des tâches à effectuer, il sera modifié et détaillé davantage une fois la conception et les maquettes du projet réalisé.

4.10 Matrice RACI

Maintenant que toutes les étapes sont planifiées, nous devons répartir le travail entre les membres de l'équipe. On utilise ainsi une matrice RACI synthétisant les rôles de chacun.

Matrice RACI (R = Réalise ; A = Autorité ; C = Consulté ; I = Informé)	**Acteurs**				
Tâches	Lucas	Noé	Alexis	Matthias	Autres élèves
Base de données					
Schéma relationnel base de données	R	R	R	AR	
Création de la base de données	C	C	C	AR	
Ensemble des requêtes	R	R	R	AR	
Frontend					
JSX	R	C	AR	I	
CSS	AR	R	R	C	
React	C	AR	C	R	
Backend					
Algorithmique	R	R	R	AR	
Python/Flask	AR	R	R	C	
Réception/Émission en JSON	I	AR	R	I	
Rédaction du rapport	R	R	R	R	C

Figure 11: Matrice RACI

4.11 Gestion des risques

Nous avons également penser à prévoir une partie des risques pouvant se dresser sur notre route, les risques les plus classiques étant la gestion du temps et le manque de comprehension de certaines personnes de l'équipe

Description	Gravité 1-4	Fréquence 1-4	Criticité	Resp	Prévention	Réparation / Plan B
L'ordinateur ne se lance pas le jour de la soutenance	3,5	1	**3,5**	Alexis	*Chacun apportera son ordinateur pour que chaque personne soit en mesure de lancer le projet en cas de problème*	
Le projet est inutilisable par le prof (problème de dépendances, ...)	3	1	**3**	Lucas	*Expliquer au mieux dans un fichier README tout les modules à installer pour faire fonctionner l'appli*	Consolider la documentation utilisateur
Mésententes dans l'équipe	3	2,5	**7,5**	Mathias	*Ne jamais s'enerver. Ne pas comprendre n'est pas un tabou. Prendre le temps de discuter et privilégier l'explication plutot que la punition pour ceux qui ne comprennent pas.*	Aborder tous les problèmes, et en reparler tant qu'ils ne sont pas résolu. Les deux personnes devront prendre le temps de discuter pour s'entendre afin que le projet puisse continuer dans la joie et la bonne humeur.
Un des membres de l'équipe se démotive ou se désinteresse du projet	3,5	2,5	**8,75**	Noé	*A chaque réunion, faire le bilan de ce qui a été fait. La mise en commun de toutes les avancées du groupe est indispensable pour ne perdre personne.*	
Un des membres de l'equipe est incompétent(e)	3,5	3,2	**11,2**	Noé	*Réactualiser les connaissances nécessaires, et planifier les formations en fonction. Ne pas laisser la personne sans aide, elle n'y arrivera pas seule.*	Refaire les mesures dans des conditions de plus grande précision, réévaluer les performances finales du dispositif
Le résultat ne correspond pas aux attentes du client	3,5	1,5	**5,25**	Mathias	*Tenter de respecter au mieux le cahier des charges*	
Délai non respectés/Mauvaise gestion du temps	3	3	**9**	Alexis	*Faire un planning précis et organisé.*	

Figure 12: Plan de gestion des risques

5 Conclusion

6 Annexes

Compte rendu de réunion

Noé Steiner - Alexis Marcel - Lucas Laurent - Mathias Aurand-Augier

16 Octobre 2022

Projet PPII - Compte rendu n°01 - réunion de lancement

Motif / type de réunion:	Lieu:
• Alexis : Présent • Noé : Présent • Lucas : Présent • Mathias : Présent	• Le 16 Octobre 2022 • De 20h à 21h • Visioconférence sur Discord

Ordre du jour:

- Tâches à effectuer
- Répartition des tâches
- Vision globale de l'application

Information échangées

- Point de vue des membres sur les fonctionnalités de l'application

Remarques / Questions

Aucune question ou remarque spécifique relevée.

Décisions

- Fonctionnalités de l'application

Actions à suivre / Todo list

- Etudier les application similaire présentes sur le marché (Tout le monde)
- Etablir une listes des documents à produire relatifs à la gestion de projet (Tout le monde)
- Etablir une liste des fonctionnalités possibles de l'application (Tout le monde)

Date de la prochaine réunion

La prochaine réunion aura lieue le Mardi 18 Octobre 2022, de 20h à 21h.

Compte rendu de réunion

Noé Steiner - Alexis Marcel - Lucas Laurent - Mathias Aurand-Augier

18 Octobre 2022

Projet PPII - Compte rendu n°02 - réunion de suivi

Motif / type de réunion:	Lieu:
• Alexis : Présent • Noé : Présent • Lucas : Présent • Mathias : Présent	• Le 18 Octobre 2022 • De 20h à 21h • Visioconférence sur Discord

Ordre du jour:

- Point sur l'avancement des tâches
- Demande d'aide en cas de difficultés

Information échangées

- Principaux éléments de gestion de projet terminés
- Intégration du livrable en LaTeX en cours
- Liste des fonctionnalités détaillées de l'application

Remarques / Questions

- Comment créer le diagramme de GANTT ? Quelle application ?

Décisions

Actions à suivre / Todo list

- Intégration des documents vers un document LaTeX (Noé)
- Réalisation graphique des figures (Lucas)
- Rédaction de la partie gestion de projet et fonctionnalités de l'application (Alexis et Mathias)

Date de la prochaine réunion

La prochaine réunion aura lieue le Samedi 22 Octobre 2022, de 20h à 21h.

Compte rendu de réunion

Noé Steiner - Alexis Marcel - Lucas Laurent - Mathias Aurand-Augier

18 Novembre 2022

Projet PPII - Compte rendu n°06 - réunion d'avancement

Motif / type de réunion:	Lieu:
• Alexis : Présent • Noé : Présent • Lucas : Présent • Mathias : Présent	• Le 16 decembre 2022 • De 20h à 21h • Visioconférence sur Discord

Ordre du jour:

- Répartition des tâches pour le premier jalon de développement du projet
- Hébergement du projet
- Spécification de l'architecture de l'application

Informations échangées

- Répartition des tâches pour le premier jalon de développement du projet :
  - Alexis : Page de profil, creation de jardin
  - Lucas : Recherche jardin sur la map
  - Noé : Joindre un jardin
  - Mathias : Modification des informations du profils

Actions à suivre / Todo list

- Progression du développement du JSX
- Bon fonctionnement de la carte des jardins

Date de la prochaine réunion

La prochaine réunion aura lieue le mardi 27 decembre 2022, de 14h à 15h.