



GESTION DE PROJET

Мережа

Préparé par

MAXIME OBRY, BENJAMIN BORELLO, LAURIE JEHAM

SOMMAIRE



Équipe

Présentation de l'équipe



Décomposition du projet

documentation,
développement, tests



Répartition charge de travail



Gestion du temps et priorité



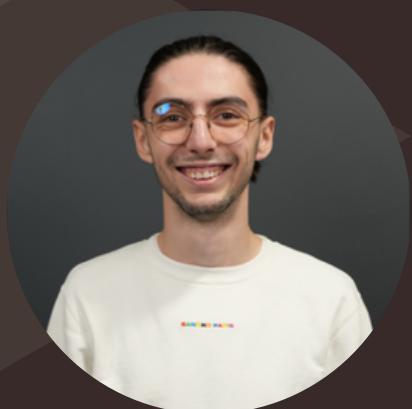
Stratégie de documentation



Documents liés au projet

Annexes, Wireframe, Schéma...

ÉQUIPE



BENJAMIN BORELLO

Chef de projet



MAXIME OBRY

Dev fullstack



LAURIE JEHAM

Dev fullstack

Décomposition du projet



Mise en place d'un outil collaboratif

Avant de commencer n'importe quel projet, il est important de tout poser sur la table afin de pouvoir se retrouver facilement quand il y a une charge de travail importante. Pour le projet Mapexa, nous avons tout d'abord commencer par ouvrir un Miro. Grâce à cet outil nous avons pu définir les éléments principaux du projet : identité visuelle, plan du site, tableau de tâche, fonctionnalités du site ainsi que les wireframes.

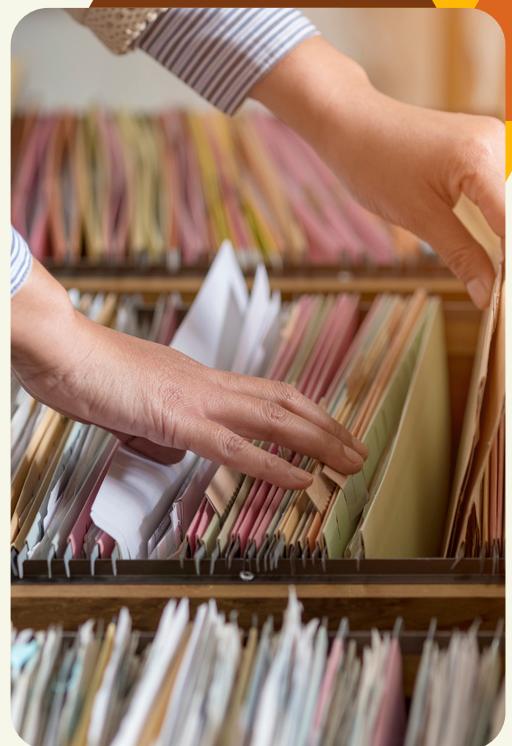
Nous avons choisi Miro car c'est une plateforme de gestion de projet complexe et élaboré que nous maitrisions au préalable.

Documentation

Une fois que tout a été listé et que les tâches ont bien été distribuées, il est temps de se lancer dans l'étape de recherche avant de commencer le développement de ce réseau social. Étant nouveaux sur PHP et ayant entendu parler des framework, c'est tout naïvement que nous avions essayé de mettre dans notre projet le framework Symphonie. Après étude de ce dernier, nous avons décidé de ne pas l'utiliser. En effet nous avons jugé le célèbre framework trop important et volumineux comparé aux besoins du projet. Le temps d'apprentissage et de prise en main aurait pu être trop laborieux.

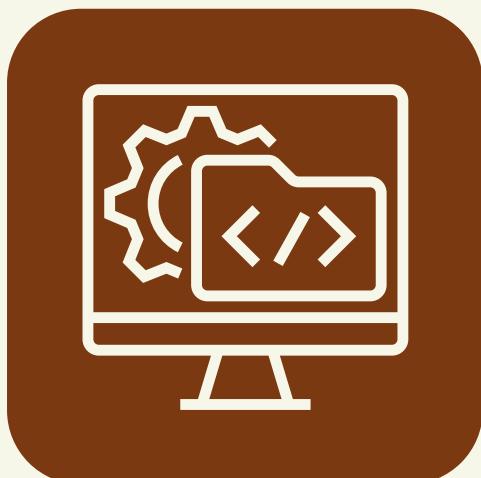
DOCUMENTATION (PARTIE 2)

La gestion du temps est une priorité qu'il ne faut pas négliger. Afin d'avoir les ressources nécessaires à l'élaboration du projet, nous avons déniché différentes ressources et documentation: notamment la document officiel de PHP, ou différents projets présent sur Github (cf. bibliographie). Nous avons aussi eu recours aux forums d'aide communautaire tel que StackOverflow et d'autres tutoriels présent sur Youtube



PHASE DE DÉVELOPPEMENT (FRONT)

La phase de développement a débuté avec la création visuel des éléments principaux du site. Nous avions besoin des éléments graphiques afin de pouvoir y attacher la logique et les fonctionnalités sous-jacentes liées notamment à la persistance des données et aux interactions des utilisateurs.



PHASE DE DÉVELOPPEMENT (BACK)

Une fois les éléments visuels intégrés, nous avons commencé à intégrer le PHP au projet. Nous avons produit et appliquer la base de données préparée au préalable depuis nos diagramme MCD/MLD.

Nous simultanément créé chacun des modèles. Ensuite nous avons créer chacun des contrôleurs et leurs logique sous-jacente. Nous les avons ensuite liés aux différentes vues.

Tout les fonctionnalités ont été créées par ordre de priorité définis lors de la phase de documentation afin de gérer plus facilement notre gestion du temps en cas d'ajustements.

PHASE DE TEST

Test unitaire :

Les phases de tests ont débutés à chaque production de fonctionnalités pour voir si nous obtenions le résultat attendu.

Pour cela nous avons notamment utilisé (abusé) la fonction var_dump afin de cibler chaque retour. Le test unitaire est une approche dans la programmation qui s'assure que les fonctions, les classes et les méthodes écrites fonctionnent comme prévu tout le long du cycle de développement. En vérifiant les valeurs en entrée et en sortie des différentes fonctions, nous nous assurons du bon fonctionnement de la logique interne.

Test d'intégration :

Une fois les tests unitaires passés, c'est le moment de faire la mise en commun du travail de chacun et vérifié le bon fonctionnement de cet ensemble.

Test fonctionnel :

Les tests fonctionnels permettent de vérifier que chaque fonction est correctement implémentée, c'est-à-dire qu'elle est conforme aux exigences et aux spécifications. Chaque fonction est généralement vérifiée indépendamment en termes d'entrées/sorties. Nous avons donc examiner le produit dans son ensemble, souvent en conjonction avec des scénarios d'utilisation réels.



Répartition des tâches



MAXIME

Connexion - Setup projet - Git - Convention - Reformat du code - DB master



BENJAMIN

Réactions - Crédit Post - Amitié - Page de profil - Crédit de post



LAURIE

Pages statiques - Recherche - Pagination - Feed - Documentation



ALL

Wireframes - Charte graphique - Liste des features - Test - Documentation

Chaque tâche a été distribué par rapport à l'ordre de priorité de ces dernières. Ensuite, elles ont aussi été donné par ordre de difficulté et de préférences. Cela avait pour but de nous confronter aux tâches sur lesquels nous n'étions pas les plus à l'aise afin de nous améliorer sur de nouvelles missions tout en s'entraînant en fonction de nos facilités.

Maxime préférait travailler sur le back, alors que Laurie a une attirance pour le dev-front. Benjamin lui était multi-casquettes. Évidemment, la conception du site a été réfléchi avec l'ensemble des membres du groupe pour une meilleure cohésion. C'est plus agréable de travailler sur un projet si chacun a eu son mot à dire sur son développement et l'idée qui allait en ressortir.

Pour rentrer dans le détail des tâches assignées :

- Maxime a été désigné pour faire le système de connexion étant donné que ce dernier avait déjà réalisé celui sur un ancien projet que nous avions déjà réalisé ensemble. Il était donc plus à l'aise pour le faire.
- Benjamin étant le chef du projet s'est désigné afin de faire le squelette du site internet afin de mieux nous aiguiller et répartir la charge de travail. Il possède de meilleure connaissance et est la personne la plus à l'aise sur ce projet.
- Laurie s'est chargée du front + documentation par préférence mais s'est aussi mis sur du back pour améliorer ses capacités étant donné qu'elle n'était pas très à l'aise dessus.

Gestion du temps et des priorités



Afin de travailler plus efficacement, il est important de définir des ordres de priorités au sein des tâches. En effet, il y a des degrés d'importance qui permettront de mieux gérer son temps par la suite. Elles sont importantes dans le sens où leur état permet de faire avancer d'autres tâches, et peuvent être cruciales pour plusieurs membres de l'équipe. Pour faire cela nous avons utilisé l'outil Miro qui nous a permis de nous organiser plus efficacement, répartir les tâches, avoir une vue générale du projet, connaître l'avancement de chacun, et mieux se réorganiser si besoin. Il a fallu mettre de côté les réponses aux commentaires pour pouvoir finir à temps, malgré que tout soit prêt dans la base de donnée. C'est le sens des priorités qui doit primer.

Priorités importantes :

- Tableau des tâches

Permet à chacun de savoir ce qu'il doit faire et avec quelle priorité

- Wireframes

Permet de construire chaque page suivant une ligne directive pour le front et le back

- Github + Structure du site

Meilleur partage des avancements, centralisations du code et rangement des fonctions

- Base SQL

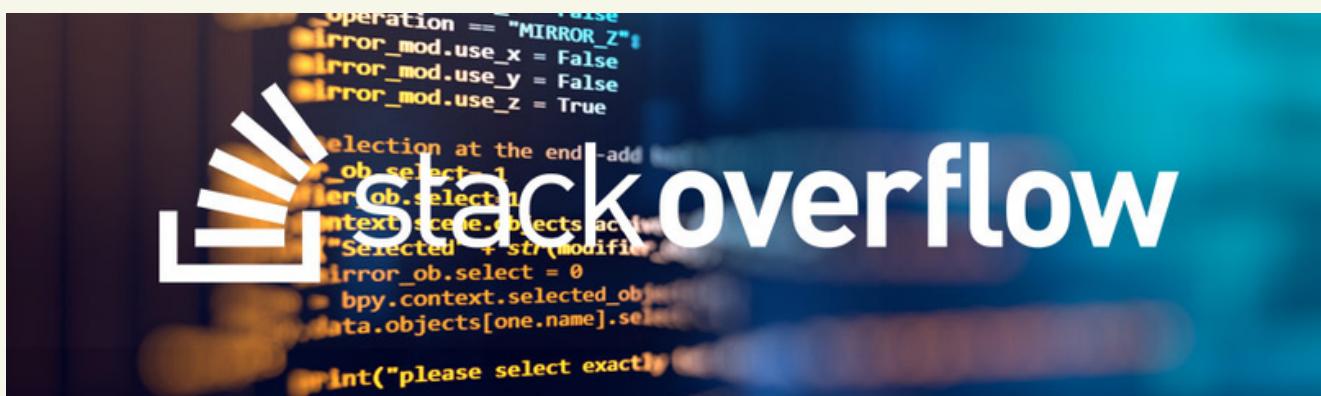
Très important, peut analyser et manipuler les données. Sans ça le site ne fonctionne tout simplement pas.

Stratégie de documentation

Pour se documenter sur la création d'un réseau social en PHP, nous avons commencé par consulter des tutoriels et des guides en ligne qui traitent de ce sujet. Il existe de nombreux tutoriels gratuits et de qualité sur les différents aspects de la création d'un réseau social en PHP, allant de l'installation et la configuration de PHP à la conception et la mise en œuvre des fonctionnalités du réseau social.



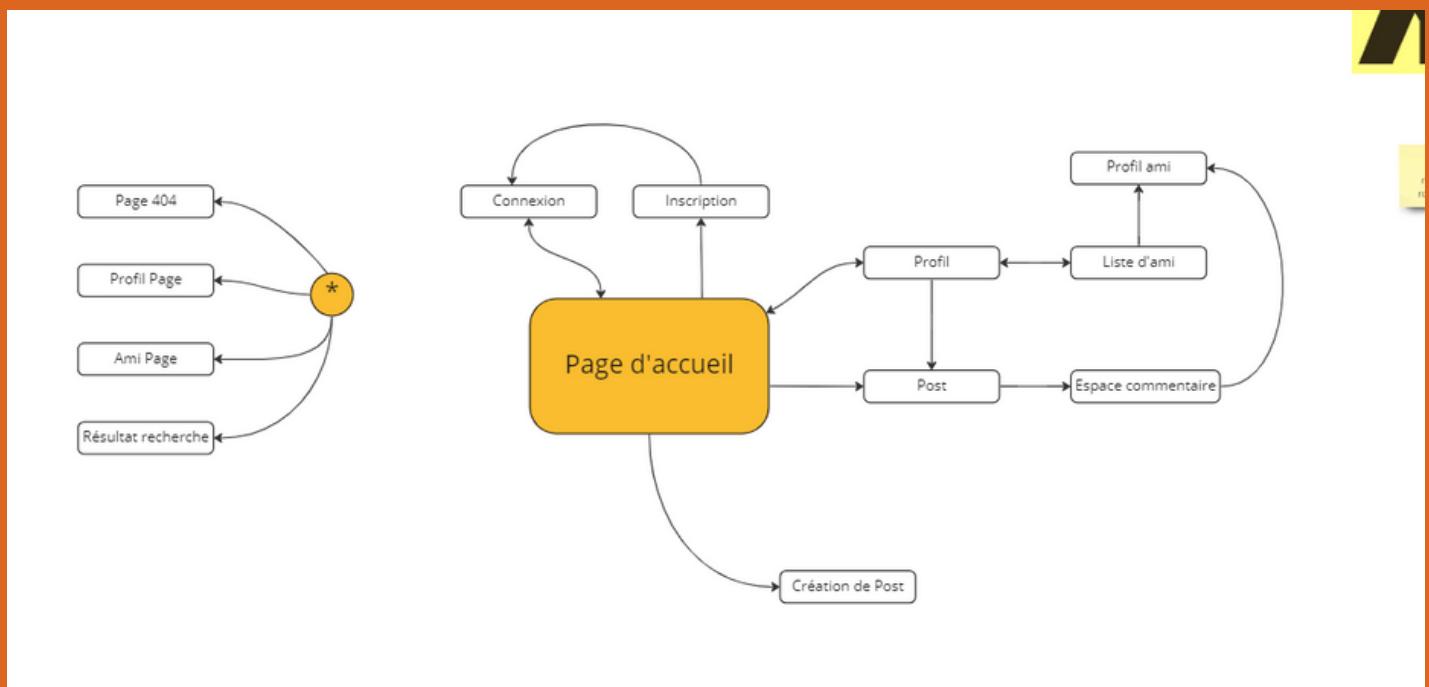
Aussi, il a été utile de se connecter à des communautés en ligne de développeurs PHP et de réseaux sociaux, où nous avons pu poser des questions et échanger avec d'autres personnes qui ont déjà créé des réseaux sociaux en PHP. Ces communautés sont une source précieuse d'informations, de conseils et de soutien. Enfin, nous avons pu compter sur l'aide et les cours de notre intervenant. Cela a permis d'avoir des réponses plus rapides et une aide personnalisée sur un problème.



Documents de compréhension de sujet

Comme expliqué précédemment, nous avons utiliser l'outil Miro pour l'avancement de notre projet. C'est là-dessus que nous avons réalisé toute notre documentation utile et intrinsèque au projet. Nous y retrouvons, les Wireframes, Liste des tâches, MVC de la bdd, les features présents sur le site, la Charte graphique, le plan du site ainsi qu'un diagramme de Gantt.

PLAN DU SITE



CHARTE GRAPHIQUE

Charte graphique

Color palette:

bg	red	green	yellow	blue	purple	aqua	gray
#282828	#cc211d	#38871a	#d7992c	#455588	#f15186	#68996a	#3a3998a
gray	red	green	yellow	blue	purple	aqua	gray
#929292	#d7d934	#888826	#f1f121	#455588	#f15186	#68996a	#3a3998a
bg0_h	bg0	bg1	bg2	bg3	bg4	gray	fg
#1d1d21	#282828	#3c3836	#565495	#665c54	#7c76f4	#3a3998a	#ebdbb2
bg0_s	fg4	fg3	fg2	fg1	fg0	orange	orange
#37320f	#a89984	#bd9993	#d5cda1	#ebdbb2	#fbf1c7	#fe8018	#fe8018

Contrast alternative: 8+ -- contrast alternative

Idée logo:

Nom du site: Mepexa
Police d'écriture: Noto sans (peut être temporaire)

Colors table:

Couleur	Hex	Utilisation
#FABD2F	U+2764	Couleur de marque
#ebdbb2	U+1F44D	Couleur text 1
#D5C4A1	U+1F44E	Couleur text 2
#1d2021	BG1	
#282828	BG2	
#3c3836	BG3	
#665c54	Border	

Emojis:

- Heart: U+2764
- Thumbs up: U+1F44D
- Thumbs down: U+1F44E
- Laughing face: U+1F602
- Crying face: U+1F622

WIREFRAMES

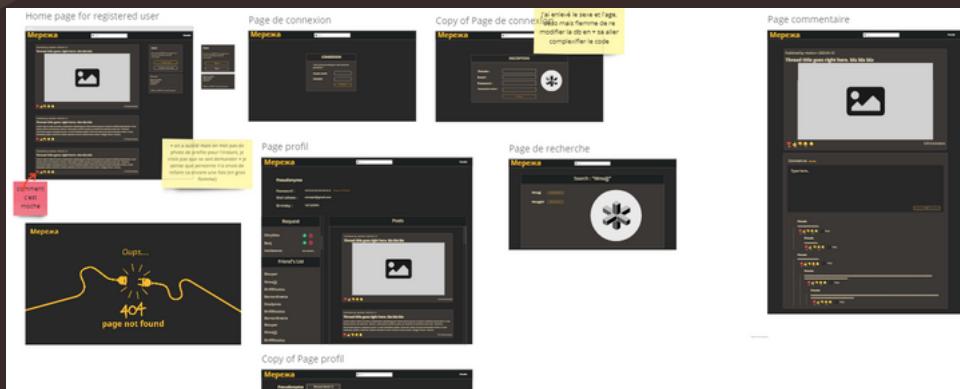
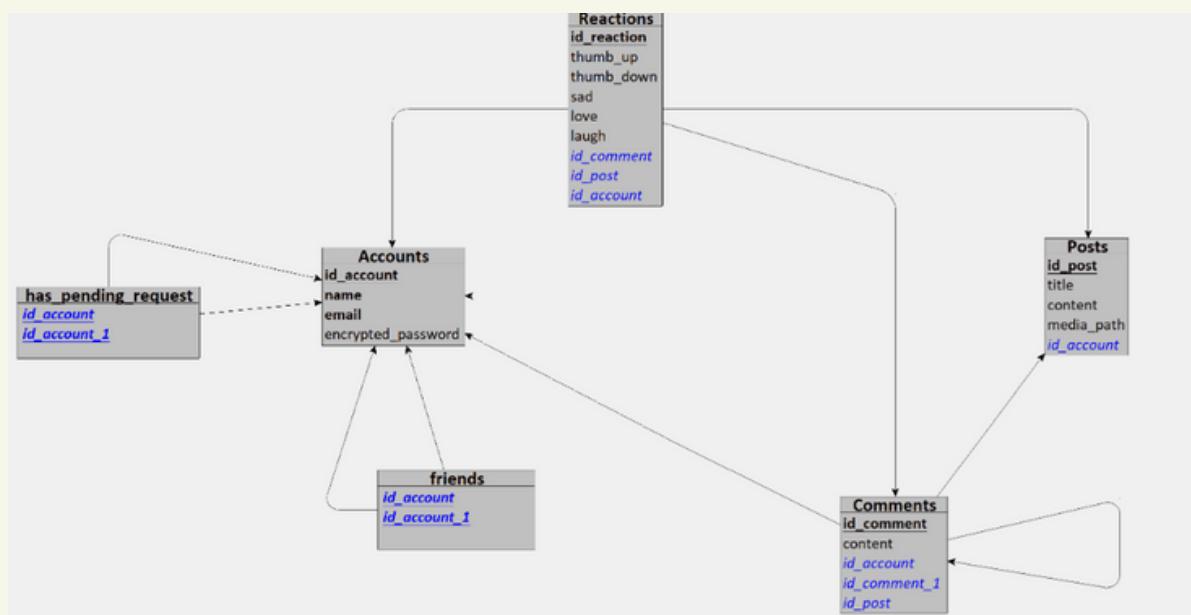
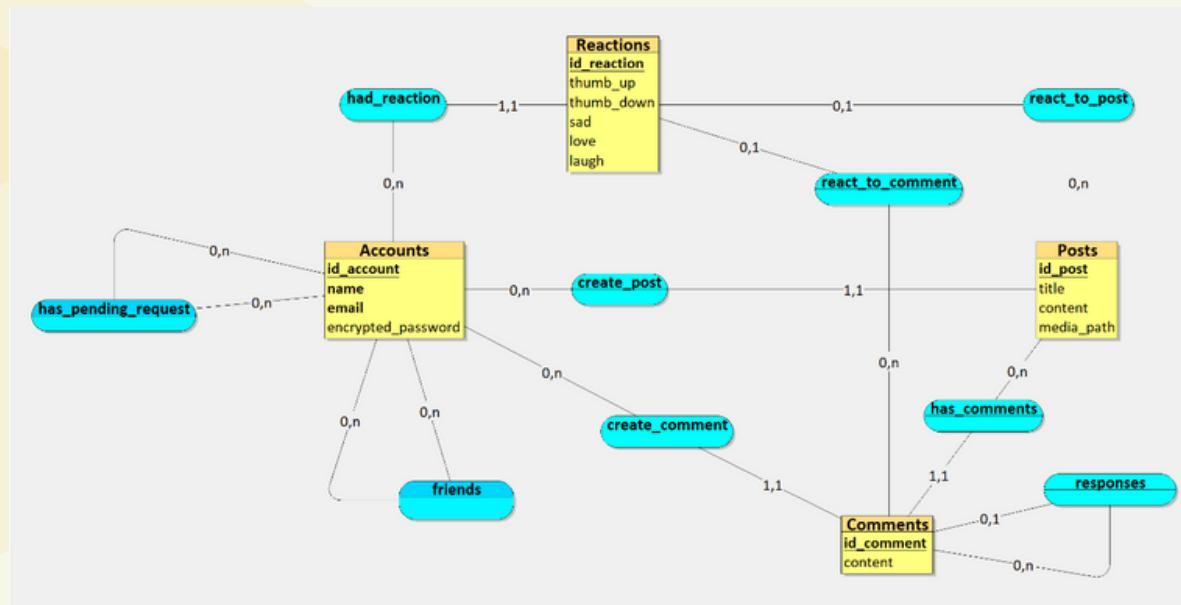


TABLEAU DES TÂCHES

Back-end				
To do 6	In Progress 1	Some shits to fix 2	Done 15	Done 2 14
reactions system	comments and shits	payer miro	limit post description to 400	test pages
commentaire = max 120	+ +	create post	feeds	search page back
delete comment		+ +	delete pending request	wireframes
Move fonction dégueulasse dans controller			secure db login	friend system
mysql dump			pagination (10 par pages)	search page front
create log for each controller + each db call + all shits in			login with mail	back 404
+ +			delete posts	

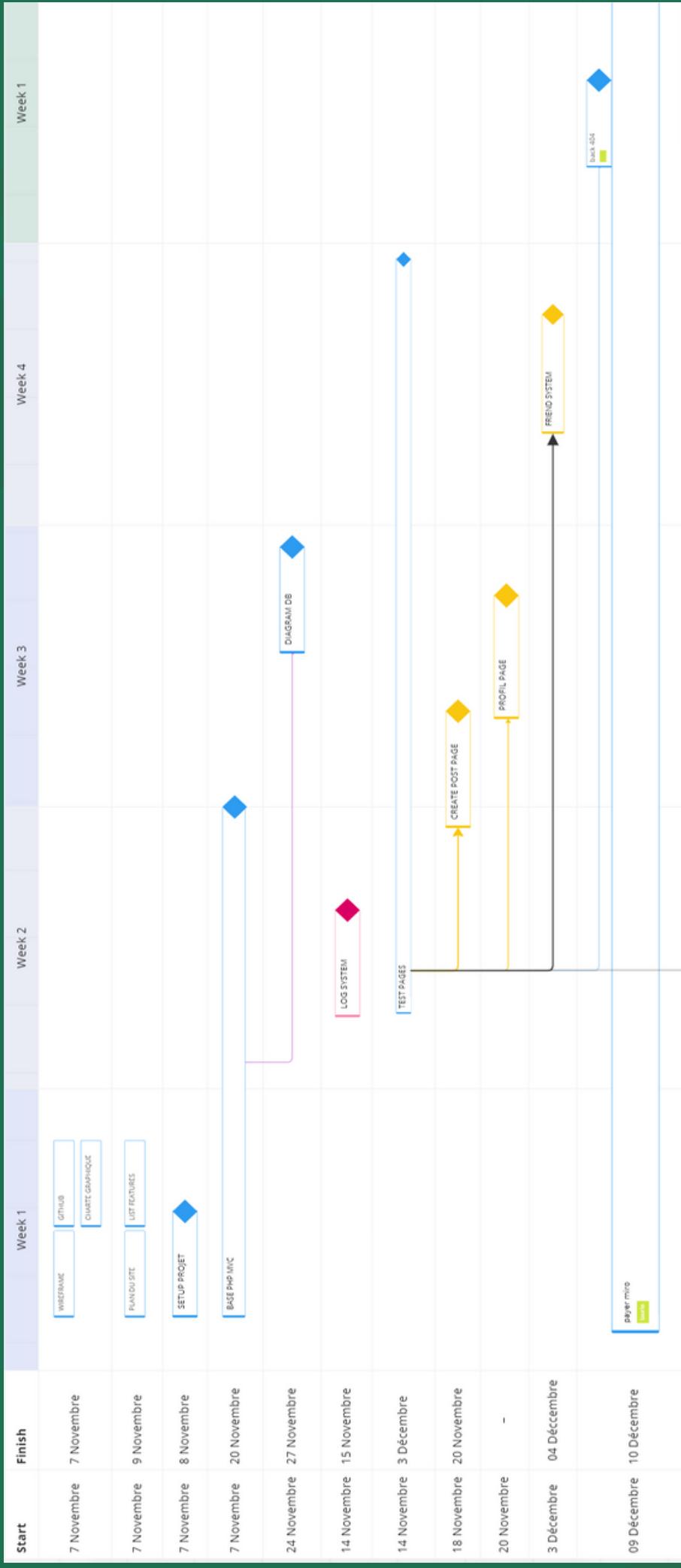
DIAGRAMME DB (MCD / MLD)



FEATURES DU SITE

FONCTIONNALITÉS	CONTRAINTEs
Inscription	Adresse Email : RFC 2822 Pseudo : entre 1 et 20 caractères Deux utilisateurs ne peuvent pas avoir le même Pseudo
Connexion/Déconnexion	L'utilisateur doit pouvoir se connecter soit avec la combinaison pseudo + mot de passe, soit avec la combinaison email + mot de passe
Requête d'ami	Un utilisateur peut envoyer une requête d'ami à un autre utilisateur et si l'autre utilisateur refuse cette requête, le premier utilisateur ne pourra pas voir le refus.
Consultation Liste Ami	L'Utilisateur peut à tout moment supprimer un Ami. Les deux Utilisateurs concernés ne sont alors plus liés.
Consultation page profil	L'utilisateur doit pouvoir consulter toutes les informations d'un profil.
Profil public / privé	Une fonctionnalité pour offrir la possibilité à l'utilisateur de rendre son profil visible à d'autre utilisateur ou non.
Création Post	L'utilisateur doit pouvoir créer un post. Le champ doit être de 2 caractères jusqu'à 400 caractères. Tout utilisateur doit pouvoir réagir à ce Post ou le commenté.
Suppression de Post	Seulement le créateur du Post doit pouvoir le Supprimer.
Consultation du feed	L'utilisateur doit pouvoir consulter son feed. Un feed contient seulement des Post qui sont susceptibles de l'intéressé comme les Post publiés par ses amies . Les Post seront ordonnés par date de publication. Le feed pourra contenir seulement 10 Post par page.
Réactions post	L'Utilisateur doit pouvoir réagir au post ou au commentaire avec la liste d'emoji dans le site. Tout Utilisateur a accès à la liste des réactions d'un Post ou d'un commentaire.
Commentaire	Tout Utilisateur doit pouvoir créer un commentaire d'un post ou un commentaire d'un commentaire avec une taille minimale de 1 caractère et une taille maximale de 120 caractères
Suppression commentaire	Le créateur d'un commentaire doit pouvoir supprimé ce commentaire et par conséquent supprimer les commentaires annexé à ce commentaire.
Réaction commentaire	L'Utilisateur doit aussi pouvoir réagir à un commentaire.
Publication de photo	L'Utilisateur peut, s'il le souhaite, associer (uploader) une photo à un Post au moment de la publication.
Utilisateur Anonyme	Si un Utilisateur n'est pas connecté, il est considéré anonyme, et dans ce cas, il aura accès seulement à l'inscription et à la connexion/déconnexion.
Logs	Un administrateur système accédant à la machine qui héberge le Site doit pouvoir consulter une trace de toutes les actions effectuées sur le Site depuis sa première mise en service.
Page de post	

DIAGRAMME DE GANTT



Bibliographie

- **Documentation PHP :**

<https://www.php.net/docs.php/>

- **Stack Overflow** : *Résolution de bugs*

<https://stackoverflow.com/>

- **GitHub** : *Comparaison projet similaire*

<https://github.com/>

- **Miro** : *Gestion de projet*

<https://miro.com/>