



TELECOM Nancy

Projet Pluridisciplinaire d'Informatique Intégrative

---

BIRD

---

Bourdais Rémi  
Clériot Louis  
Serrand Corralie  
Tejedor Manon

Responsable de module :  
Olivier Festor



6 janvier 2022

# Table des matières

<b>1</b>	<b>Remerciement</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
2.1	Contexte du projet . . . . .	5
2.2	Organisation du document . . . . .	5
<b>3</b>	<b>État de l'art</b>	<b>6</b>
3.1	Introduction . . . . .	6
3.2	Définition des critères . . . . .	6
3.3	Concept et présentation des applications étudiées . . . . .	6
3.4	Comparatif des Civic Tech existantes . . . . .	8
3.5	Critique de programmes existants . . . . .	8
3.6	Conclusion . . . . .	8
<b>4</b>	<b>Gestion de projet</b>	<b>9</b>
4.1	Équipe de projet . . . . .	9
4.2	Outils de travail . . . . .	9
4.3	Déroulement . . . . .	9
4.3.1	Brainstorming . . . . .	9
4.3.2	TO DO LIST, matrice RACI, diagramme de GANTT . . . . .	9
4.3.3	Jalons et PDCA . . . . .	10
4.3.4	Matrice SWOT . . . . .	10
<b>5</b>	<b>Conception et implémentation</b>	<b>12</b>
5.1	Introduction générale . . . . .	12
5.2	Web . . . . .	13
5.3	Algorithmes de traitement . . . . .	15
5.3.1	Fil d'actualité . . . . .	16
5.3.2	Fonction allowed_image(filename) . . . . .	16
5.3.3	Vérification des contraintes . . . . .	16
5.3.4	Fonction test_login() . . . . .	16
5.3.5	Fonction test_verif() . . . . .	16
5.3.6	Fonction is_owner(id,user) . . . . .	16
5.3.7	Fonction est_abonne(id,user) . . . . .	16
5.3.8	Fonction est_participant(id,user) . . . . .	16
5.3.9	Fonction a_demande(id,user) . . . . .	17
5.3.10	Fonction_commentaire(i_com,user) . . . . .	17
5.3.11	Fonction likepost(id post,user) . . . . .	17
5.3.12	Fonction recommandation . . . . .	17
5.3.13	Autres fonctions . . . . .	17
5.4	Base de données . . . . .	18
<b>6</b>	<b>Tests et performances</b>	<b>20</b>
6.1	Complexité de l'algorithme de recommandation . . . . .	20
6.1.1	Analyse de complexité . . . . .	20
6.1.2	Graphiques . . . . .	20
6.2	Tests réalisés . . . . .	22
6.2.1	Liste des Tests . . . . .	22
6.2.2	Problèmes liés aux Tests . . . . .	22
6.2.3	Right BICEP . . . . .	23

<b>7</b>	<b>Bilan du projet</b>	<b>24</b>
7.1	Conclusion . . . . .	24
7.2	Conclusions personnelles . . . . .	25
7.2.1	Bourdais Rémi . . . . .	25
7.2.2	Clériot Louis . . . . .	25
7.2.3	Serrand Coralie . . . . .	26
7.2.4	Tejedor Manon . . . . .	26
<b>8</b>	<b>Annexes</b>	<b>28</b>
8.1	Comptes-rendus des réunions . . . . .	28
8.1.1	18 novembre 2021 . . . . .	28
8.1.2	23 novembre 2021 . . . . .	28
8.1.3	24 novembre 2021 . . . . .	30
8.1.4	25 novembre 2021 . . . . .	31
8.1.5	20 décembre 2021 . . . . .	31
8.1.6	22 décembre 2021 . . . . .	32
8.1.7	26 décembre 2021 . . . . .	33
8.1.8	28 décembre 2021 . . . . .	35
8.1.9	30 décembre 2021 . . . . .	36
8.2	Gestion de projet . . . . .	38
8.3	Sources . . . . .	39

# Table des figures

4.1	Matrice SWOT . . . . .	10
5.1	Schéma de conception des interactions partie 1 . . . . .	13
5.2	Schéma de conception des interactions partie 2 . . . . .	13
5.3	Photo de la barre de navigation 'layout' . . . . .	14
5.4	Photo de la barre de navigation dans sub . . . . .	15
5.5	Schéma relationnel de la base de données . . . . .	19
6.1	Graphique 3D de la complexité de recommandation . . . . .	20
6.2	Photo régression linéaire suivant le nombre de subs . . . . .	21
6.3	Régression linéaire du temps d'exécution en fonction du nombre de mots compris dans le titre et les mots-clés par sub . . . . .	21
6.4	Régression polynomiale du temps d'exécution en fonction du nombre de mots par sub . . . . .	22
8.1	Matrice RACI . . . . .	38
8.2	Diagramme de Gantt . . . . .	39

# Chapitre 1

## Remerciement

Tout d'abord nous souhaitons remercier Telecom Nancy et la ville de Nancy pour nous avoir permis de travailler sur le sujet trop méconnu mais intéressant qu'est la Civic Tech, nos encadrants et professeurs Mr Olivier Festor et Mr Gérard Oster pour les cours d'algorithmique et de Flask dispensés sans qui nous n'aurions pu mener ce projet à bien ainsi que pour l'encadrement lors du projet. Nous souhaitons également remercier Mme Heurtel pour ses conseils en gestion de projet pour la réalisation de la matrice RACI et du diagramme de Gantt. Finalement nous aimerions remercier Thibault Chenevière pour un cours particulier de Flask qui nous a permis de comprendre des subtilités de ce langage.

## Chapitre 2

# Introduction

### 2.1 Contexte du projet

Ce projet a été réalisé dans le cadre du Projet Pluridisciplinaire d'Informatique Intégrative en première année du cycle ingénieur à Telecom Nancy.

Depuis plusieurs années, on observe une diminution de l'intérêt des citoyens à l'égard de l'action publique. Pour lutter contre ce phénomène, différents acteurs cherchent à créer des solutions dans des domaines variés pour redynamiser l'engagement dans la vie politique à plusieurs échelles. C'est ce que l'on appelle la démocratie participative. En tant qu'élèves ingénieurs dans le numérique, il nous est demandé d'imaginer une application web appartenant à la Civic Tech. En d'autres termes, nous devons implémenter une application de démocratie participative pouvant être utile à une ville.

Notre travail est décomposé en quatre parties. Tout d'abord un état de l'art a été réalisé pour se rendre compte des technologies déjà existantes et s'en inspirer. Un deuxième temps a été dédié à la conception de notre application et à l'analyse des besoins. Suite à cela, la phase de programmation a été lancée. Dans un dernier temps, nous avons implémenté des fonctions de test, corrigé nos fonctions et évalué leurs performances. Enfin, tout au long du projet des méthodes de gestion de projet en équipe ont été mises en oeuvre.

### 2.2 Organisation du document

Dans le deuxième chapitre, nous présentons ce qu'est la Civic Tech et nous analysons et comparons les applications existantes.

Dans le troisième chapitre, nous présenterons les outils et les méthodes de gestion de projet mis en place. Dans le quatrième chapitre, nous détaillerons la conception et l'implémentation des différentes parties de notre application, à savoir la partie web, la partie base de données ainsi que la partie algorithmique. Dans le cinquième chapitre, nous verrons finalement les tests effectués pour s'assurer du bon fonctionnement de l'application web ainsi qu'une étude de leur performances.

## Chapitre 3

# État de l'art

### 3.1 Introduction

Les Civic Tech regroupent toutes les applications web et logiciels dédiés au développement de la démocratie participative, à la facilitation et la transformation de l'engagement citoyen ou encore à la réalisation de projets. La Civic tech est un domaine connaissant une grande croissance pendant les dernières années. De plus, ce domaine est très vaste et les applications très diverses.

L'objectif de cet état de l'art est de comparer les applications web de Civic-tech déjà existantes avant de commencer la conception de notre projet. Il a donc été nécessaire de définir des critères afin de comparer ces applications. Nous avons fait des recherches pour trouver les critères les plus pertinents. Il s'avère qu'il existe beaucoup de possibilités de classification.

### 3.2 Définition des critères

Suite à nos recherches, nous avons finalement choisi d'adopter cinq critères :

- Economie (marchand ou non marchand)
- Type de service (Co-production, Information, Mobilisation)
- Initiateur de l'application
- Orientation de la communication entre dirigeants et communautés
- Open Source ou non

Le critère sur l'open source a été initié par Valentin Chaput, co-fondateur de Open Source politics. Tandis que la Knight Foundation propose de répartir les applications selon leurs initiateurs et leur orientations vis-à-vis de la communication entre dirigeants et communautés. En d'autres termes, on cherche à savoir si l'application a été mise en place par les collectivités territoriales ou si elle est mise en œuvre par l'initiative des citoyens. Puis on regarde si l'application permet aux citoyens de proposer leurs initiatives participatives aux collectivités territoriales ou bien si, à l'inverse, l'application donne le moyen aux collectivités territoriales de réclamer une participation démocratique de la part de ses communautés.

L'économie et le type de service ont été définis par la cartographie de la Civic Tech de L'observatoire des Civic Tech et de démocratie numérique [10] ont réalisé une cartographie de la Civic Tech. C'est dans celle-ci que les trois types de services ont été définis. Nous avons trouvé cette répartition pertinente et l'avons donc intégrée à nos critères de comparaison. La co-production correspond aux applications qui permettent aux collectivités territoriales de consulter et d'impliquer les citoyens dans l'action publique. Les applications de mobilisation englobent les outils de création de communautés à différentes échelles comme les pétitions ou les réseaux sociaux. Enfin les applications au service de l'information correspondent à celles ayant pour finalité la transmission des données et l'accès à l'information de manière lisible et claire pour le plus grand nombre.

### 3.3 Concept et présentation des applications étudiées

Solucracy [6]

Solucracy est une association ayant pour but de fournir une méthode pour mobiliser les acteurs d'un territoire comme des citoyens, des collectivités territoriales ou des entreprises. Cette méthode est disponible gratuitement sur leur site internet. Ils proposent éga-



lement divers degrés d'accompagnement. L'association peut former une équipe et intervient tout au long du projet pour faciliter la communication. Elle peut aussi être présente seulement les deux premiers jours et proposer du coaching à distance pendant le reste du temps du projet. Des fiches explicatives de la méthode sont également disponibles et Solucracy se propose de faire des appels ponctuels pour faire le point sur l'avancement du projet.

#### DemocracyOS [4]

DemocracyOS est un logiciel open source créé en Argentine et maintenant présent dans de nombreux pays. Ce logiciel est une plateforme web qui permet à des associations, des entreprises ou des gouvernements de prendre des décisions transparentes basées sur les avis des membres des collectivités. En France, il est utilisé par le gouvernement, l'assemblée nationale et certaines villes. On peut noter que Nancy l'utilise depuis 2018 à travers des collectifs informels et des associations. L'application web permet de créer des espaces de discussion et de délibération sur des sujets choisis pour prendre des décisions collectivement grâce à des votes ou des sondages.

#### Cap Collectif [3]

Cap Collectif fournit différentes applications pour permettre aux collectivités territoriales et entreprises privées de développer leurs propres outils de civic tech. Pour cela six applications peuvent être utilisées :

- Consultation : permet au public de modifier des propositions fournies en amont pour en produire une version finale
- Budgets participatifs : permet d'allouer des budgets aux projets choisis par les habitants
- Questionnaire : recueillir l'opinion publique
- Boîte à idées : recueillir les idées du public
- Appels à projets : recueillir des projets du public correspondant à un problème donné
- Pétition : Permet au public que leurs propositions soient délibérées officiellement après un certain niveau de consensus

#### Wesignit [8]

L'application permet de créer des pétitions en ligne et de les partager sur les réseaux sociaux pour interpeller les acteurs du sujet visé. Il est également possible d'organiser des événements et des mobilisations pour faire vivre la pétition hors du web. L'application permet également de mettre en avant les plateformes de dons si besoin.

#### Stig [8]

L'application Stig peut être utilisée par les employés d'une entreprise ou par les citoyens d'une commune pour faire des propositions de tout type et voter pour celle des autres. Une intelligence artificielle permettra de classer les différentes notions et concepts automatiquement ce qui permettra ainsi aux décisionnaires d'observer les tendances et les dynamiques des propositions. On peut donc se servir de Stig dans de nombreux domaines tels que la gouvernance (préparer une assemblée générale, élaborer un programme, ...), le management (améliorer les conditions de travail). Pour les collectivités territoriales, l'application permet de sonder les habitants ou de construire un budget participatif.

#### Vooter [7]

L'application web permet d'organiser des votes à distance, de consulter et faire participer les membres du collectif aux décisions. L'application permet d'informer, directement sur leurs smartphones, les collaborateurs et peut être utilisée pour obtenir des informations en direct, pendant les réunions par exemple.

#### Regards citoyens [5]

C'est une association qui s'est donnée pour mission d'informer les citoyens sur le fonctionnement des institutions politiques françaises. Pour cela, elle a créé plusieurs sites. Par exemple, certains mettent en avant l'activité des sénateurs et des députés. On peut également retrouver de nombreuses données telles que les groupes de lobbyistes de l'assemblée nationale. Une analyse des données est présente pour comprendre ce que l'on voit. On retrouve également une section revue de presse mise à jour avec des actualités politiques.

3.4 Comparatif des Civic Tech existantes

Site	Economie	Catégorie	Initiateur	Orientation	Open source
Solucracy	non marchand	co-production	entreprise tier	haut-bas	oui
DemocracyOS	non marchand	co-production	dirigeants	haut-bas ou horizontale	oui
Citizenlab	marchand	co-production	dirigeants	haut-bas	oui
Capcollectif	marchand	co-production	dirigeants	haut-bas	non
Wesignit	marchand	mobilisation	citoyens	bas-haut	oui
Stig	marchand	mobilisation	dirigeants	bas-haut	non
Vooter	marchand	mobilisation	dirigeants	haut-bas	non
Regards citoyens	non marchand	information	citoyens	bas-haut	non

3.5 Critique de programmes existants

3.6 Conclusion

Suite à notre étude, nous avons décidé de nous tourner vers une application de type co-production. Cela nous a paru le plus utile vis-à-vis du renforcement de l’engagement citoyen ainsi que plus facile a mettre en place par des collectivités territoriales. Nous avons aussi décidé que la communication serait plus dirigée des citoyens vers les collectivités territoriales.

# Chapitre 4

## Gestion de projet

### 4.1 Équipe de projet

L'équipe que nous formons est constituée de quatre étudiants en première année :

- Bourdais Rémi
- Clériot Louis
- Serrand Coralie
- Tejedor Manon

Au vu de la dynamique du projet, de l'entente cordiale de ses membres et de la taille de l'équipe, nous avons opté pour une structure fonctionnelle, c'est à dire sans chef de projet. Chacun avait donc la tâche d'animer les réunions et d'y participer.

### 4.2 Outils de travail

Pendant toute la durée du projet, nous avons organisé plusieurs réunions. Le suivi de celles-ci peut se retrouver sous forme de comptes-rendus en annexe. Lorsque nous étions tous présent, les réunions ont eu lieu à Télécom Nancy. Pendant les vacances de Noël, celles-ci se sont déroulées via la plateforme Discord. Différentes to do lists ont été conçues tout au long du projet pour nous permettre de nous répartir le travail en fonction de l'importance de celui-ci. De plus un diagramme prévisionnel de Gantt élaboré au commencement nous a permis de nous projeter dans le temps même si quelques problèmes seront abordés plus bas dans ce rapport. Nous avons également communiqué sur la plateforme Messenger. Notamment pour établir les dates des réunions et s'aider en cas de blocage. Nous avons aussi utilisé un drive pour pouvoir se partager les documents.

Lors du développement de notre application web, nous avons utilisé le git mis à disposition par l'école pour notre projet afin d'y déposer régulièrement nos avancées. Le logiciel Visual studio code aura servi à l'implémentation et l'extension Microsoft Live Share aura permis la programmation simultanée pour les lots de travail effectués à plusieurs. La rédaction du rapport s'est faite via la plateforme OverLeaf.

### 4.3 Déroulement

Afin de se répartir au mieux la charge de travail, nous nous sommes servis de certains éléments de gestion de projets. [9]

#### 4.3.1 Brainstorming

Lors de nos premières réunions, nous avons organisé des séances de brainstorming. À tour de rôle, chacun exposait ses idées et ses recherches effectuées au préalable. Ainsi, au début de notre projet, nous avons beaucoup échangé et exploré différentes pistes. Nous avons discuté les idées de chacun, pointé celles intéressantes pour au final s'accorder sur le concept et les fonctionnalités de notre application web. Pendant cette phase, nous avons organisé plusieurs réunions toutes les semaines et parfois le délai entre celles-ci n'était que de vingt-quatre heures.

#### 4.3.2 TO DO LIST, matrice RACI, diagramme de GANTT

Dans la continuité des brainstormings, une liste des tâches a été réalisée. Chaque tâche est classée par ordre de priorité avec un code couleur. Nous nous sommes basés sur cette

TO DO LIST pour faire une matrice RACI (Réalise,Autorité,Conseille,Informe). On attribue les tâches de façon à ce que chaque participant travaille sur tous les domaines. Cette matrice RACI a été amenée à évoluer au court du projet dû a la découverte de la nécessité d'inclure de nouvelles tâches non prévues initialement. Vis-à-vis de la taille de notre équipe, et notre structure étant fonctionnelle, personne n'a été désigné comme "Autorité" ou "À informé". Néanmoins chaque membre de l'équipe qui ne réalise pas la tâche est présent pour conseiller et aider si besoin. En effet comme cité précédemment, nous avons utilisé l'application Messenger pour communiquer et s'entraider quand cela était nécessaire. Il est vrai que certaines tâches sont parfois similaires et les problèmes rencontrés lors de leurs réalisations le sont donc aussi. Enfin, en attribuant un délais prévisionnel pour chaque tâche, on transforme la matrice RACI en diagramme de Gantt. Il est important de noter que ce diagramme a évolué pendant toute la durée de ce projet en subissant plusieurs mises à jours. La matrice RACI et le diagramme de Gantt sont disponibles en annexes.

4.3.3 Jalons et PDCA

Afin d'éviter des blocages trop longs, nous avons mis en oeuvre plusieurs PDCA. Dans un premier temps, dans un cycle de PDCA,on prévoit grâce au diagramme de Gantt les tâches à réaliser jusqu'à la prochaine réunion. Puis on passe à la phase de développement. Enfin lors d'une nouvelle réunion, on contrôle l'avancement du projet en récapitulant ce qui a été fait par chaque membre de l'équipe. Chacun annonce ce qu'il a eu le temps de faire. En cas de blocage ou de problèmes techniques, les autres membres de l'équipe peuvent ainsi apporter leur aide. On procède ensuite à l'ajustement du diagramme de Gantt en fonction de ce qui a été fait. En effet, il nous est parfois arrivé d'être trop ambitieux sur la charge de travail sur la période du PDCA. Nottament au moment des fêtes de fin d'année. Néanmoins l'organisation de ces jalons tous les deux jours nous aura permis d'avoir un bon contrôle sur l'avancé du projet. Après avoir ajusté le diagramme de Gantt, on relance un nouveau PDCA en prévoyant au cours de cette même réunion les tâches à faire pendant les deux jours prochains.

4.3.4 Matrice SWOT

Une évaluation de nos forces et de nos faiblesses ainsi qu'une analyse des menaces et des opportunités ont été regroupées dans une matrice SWOT que l'on peut observer ci-dessous.

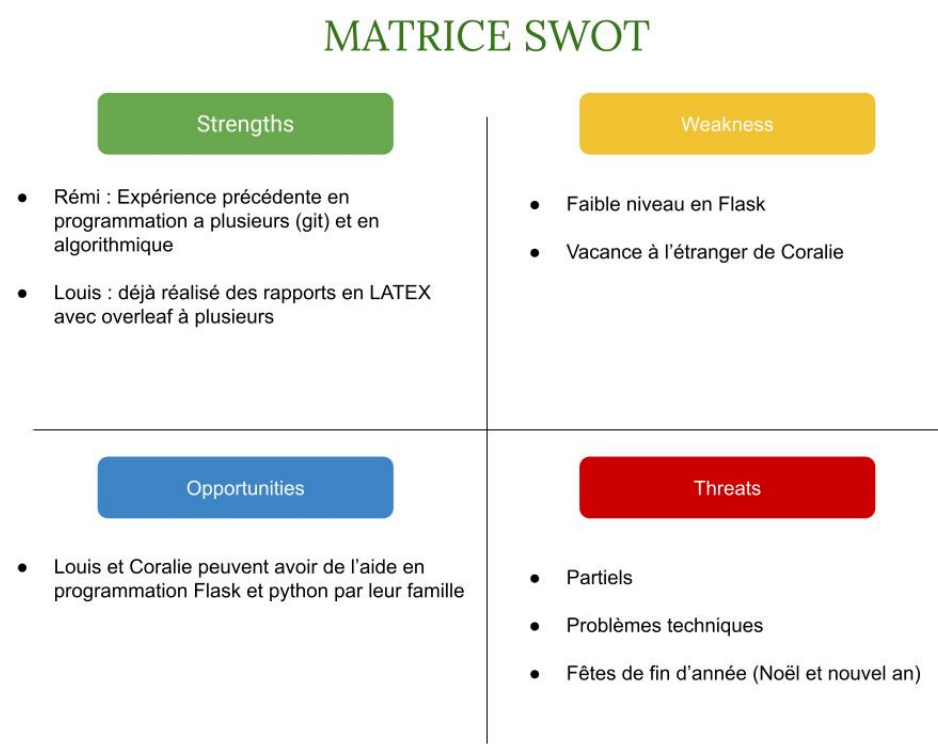


FIGURE 4.1 – Matrice SWOT

A partir de celle-ci, nous avons tenté de mettre en oeuvre des solutions pour pallier nos faiblesses. D'une part, Coralie a emmené son ordinateur lors de son voyage à l'étranger.

De plus, les journées en Suède étant très courtes, elle a pu avoir du temps pour travailler le projet en fin d'après-midi ou le soir. D'autre part, le faible niveau en Flask a été compensé suite à un cours particulier donné par un élève de l'école. Les TP et partiels de WEB ont aussi contribué à son amélioration. Par ailleurs, la phase d'implémentation aura été très formatrice.

Cependant, certaines menaces ont difficilement pu être évitées. En effet, la phase de programmation n'a pas pu être lancée avant la fin des partiels. D'autre part, les réunions prévues le 24 et le 31 Décembre n'ont pas pu avoir lieu. Nous avons aussi été confrontés à plusieurs reprises à des problèmes de fusion sur Git. En effet, jusqu'alors nous étions habitués à utiliser Git sur un travail personnel. Il nous a donc fallu appréhender cette nouvelle dimension de la programmation à plusieurs. On peut noter que ces problèmes de merge ont été beaucoup moins fréquents à la fin du projet. Notamment grâce à des conseils apportés par des membres de la famille comme signalé dans les opportunités de la matrice SWOT. En outre, Coralie aura subi des problèmes techniques rendant impossible l'accès à sa machine virtuelle. Néanmoins ceux-ci ont facilement pu être contrés en installant Visual studio code et Git bash sur Windows.

## Chapitre 5

# Conception et implémentation

### 5.1 Introduction générale

L'objectif de notre application web est d'offrir un espace d'échange et de débats aux citoyens ainsi qu'un espace de décision afin de faciliter la démocratie participative. Ainsi, en plus de permettre la liberté d'expression, de faciliter les échanges, et d'organiser les débats, l'application offre le moyen de participer aux prises de décision et facilite l'implication de ceux qui le souhaitent dans la vie citoyenne locale.

Chacun serait libre de proposer ses idées de projets (subs) ainsi que de donner son avis sur les projets. Un système de notation permet aux utilisateurs de montrer leur intérêt pour une idée. Ce système de notation permet de recommander plus facilement un projet à une personne.

L'utilisateur qui avait eu l'idée peut alors former son groupe d'organisateur – les autres utilisateurs peuvent ensuite choisir par exemple de participer au projet, ou bien de simplement s'abonner au projet afin d'être tenu au courant des dernières actualités le concernant sans pour autant y participer activement.

Un chat lié au projet permet aux organisateurs et participants de communiquer sur les tâches à faire, les organisateurs peuvent organiser des sondages pour recueillir les avis des participants sur des éléments tels que la date pour un événement lié au projet par exemple. Nous voulions également proposer un calendrier interactif, qui n'a pas été possible d'implémenter lié aux difficultés rencontrées.

Les organisateurs et participants peuvent aussi poster régulièrement sur le fil d'actualité du projet les dernières avancées de celui-ci. Ces posts seront visibles par tout ceux qui accéderont à la page liée au projet et apparaîtront sur le fil d'actualité de la page d'accueil des utilisateurs s'étant abonnés au projet.

Une idée de projet peut prendre différentes formes telles que le nettoyage d'une plage, l'installation d'un compost, la création d'une association, l'ajout d'un club sportif ou culturel etc. . .

Dans le cas des associations, clubs, conseils. . . le chat de discussion (à disposition des participants aux projets uniquement) pourra servir aux membres de ces derniers pour communiquer. La page liée au projet pourra servir à partager l'actualité.

Notre application web offre aussi un onglet recommandations dans lequel un utilisateur pourra voir des projets susceptibles de lui plaire en fonction des projets auxquels il s'est déjà abonné.

Cette application ayant une finalité démocratique, il est nécessaire de s'assurer que chaque utilisateur n'ait qu'un seul et unique compte. Pour cela on peut imaginer que la création du compte nécessite un certain code. Notre application pourra gérer le fait que lors de la création du compte, ce code doit être valide. Cependant les moyens qui permettent aux citoyens de récupérer ces codes ne seront pas gérés par l'application mais par exemple par la mairie.

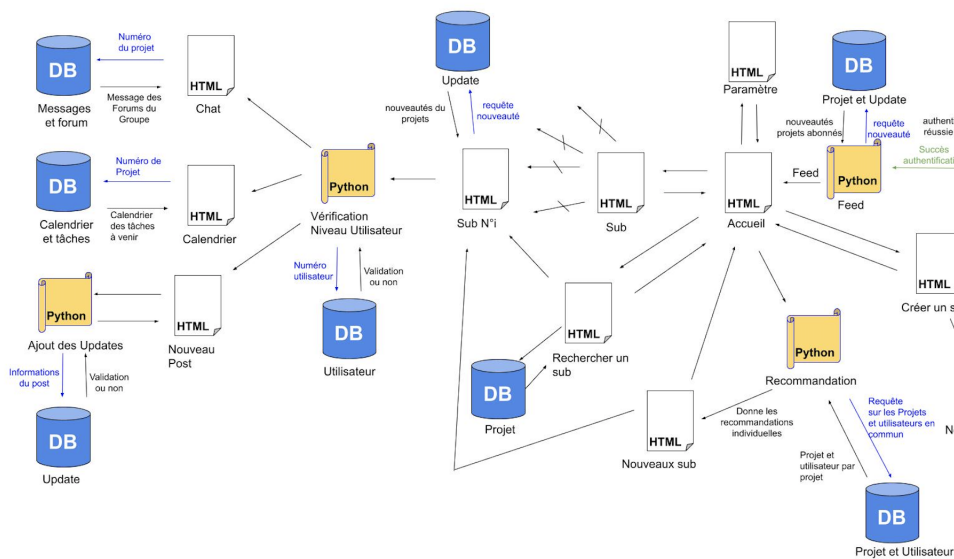


FIGURE 5.1 – Schéma de conception des interactions partie 1

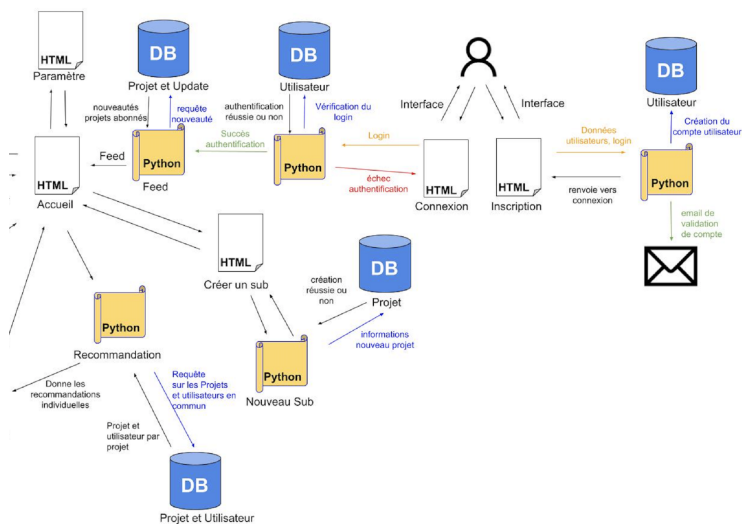


FIGURE 5.2 – Schéma de conception des interactions partie 2

Dans un premier temps, nous avons fait un schéma détaillant les interactions entre les constituants de notre application web.

La structure finale de notre application est finalement un peu différente de celle prévue au départ. En effet, au fur et à mesure du projet, nous avons été confrontés à de nouveaux besoins et il y’a certains points que nous n’avons malheureusement pas eu le temps d’aborder ou bien nous n’avons pas réussi à les implémenter.

Notre application est codée avec le langage Python et utilise le module flask. A la racine du projet on retrouve notre fichier app.py qui contient les routes de notre application web ainsi que notre base de données database.db. Pour implémenter cette base de données, on utilise sqlite3. Toujours à la racine de notre projet, on retrouve trois dossiers :

- templates (qui contient tous nos fichiers HTML)
- static
  - img (qui contient les images)
  - styles (qui contient les feuilles de styles sous format .css)
- python (qui contient nos fonctions pythons)
- pytest (qui contient les fichiers tests)

## 5.2 Web

En tout premier lieu, lors de l’ouverture de notre application web, apparaît une page d’authentification. Sur celle-ci est demandé l’adresse mail et le mot de passe de l’utilisateur. Si celui-ci ne possède pas de compte, un lien pour s’enregistrer est disponible en dessous de ces requêtes.

Est requis pour s'enregistrer :

- Une adresse mail
- Un Nom
- Un Prénom
- Un mot de passe

Nous avons fait une page html se prénomant 'layout.html'. Celle-ci s'affiche sur chaque page de l'application web. Nous avons utilisé une nav-bar proposée par Bootstrap pour l'implémentation. Nous retrouvons sur cette barre plusieurs fonctionnalités présentées ci-contre. Il nous a donc fallu étendre ce layout sur chaque fichier html.



FIGURE 5.3 – Photo de la barre de navigation 'layout'

L'accueil est constitué de la page 'accueil.html'. Sur cette page s'affiche un fil d'actualité qui correspond aux derniers posts créés auxquels l'utilisateur est abonné. Il est possible de cliquer sur les posts afin de visualiser le projet auquel il se rattache. Il correspond à `@app.route('/accueil')`.

L'onglet 'Parcourir' affiche l'ensemble des subs créés sans tenir compte des recommandations. De même, il est possible de cliquer sur un projet afin d'en avoir les détails. Le bouton est muni d'un lien renvoyant vers `@app.route('/parcourir')`.

L'onglet 'Projet' est un menu déroulant. Il est créé grâce à un dropdown-menu. On y retrouve les boutons 'Créer un sub' et 'Recommandations'. Ils mènent respectivement vers `@app.route('/form')` et `@app.route('/recommandation')`

'Créer un sub' mène vers une page sur laquelle on retrouve un form nous demandant de mettre :

- le nom du projet
- des mots-clés correspondants au projet
- une description

Afin de pouvoir soumettre ce projet, il est nécessaire d'être un utilisateur avec le statut 'administrateur' ou 'vérifié'. Pour ce faire, nous avons créé une page html : 'sub.html' dans laquelle nous avons fait un form. Lors de l'appui sur le bouton 'Submit', si l'utilisateur a le bon niveau. Cela le redirige ensuite sur `@app.route('/form')` Les informations du form sont alors insérées dans la base de données. L'utilisateur est ensuite redirigé vers l'accueil. Le bouton 'Recommandations' renvoie vers `@app.route('/recommandation')`. Il permet d'avoir accès à des projets qui ont été traités par un algorithme de recommandations. Ces recommandations dépendent des projets auxquels l'utilisateur est abonné.

Il y'a également un onglet déroulant 'Mes infos'. Nous y retrouvons les boutons :

- Mon Profil
- Mes Abonnements
- Mes Projets

Dans 'Mon Profil' (`@app.route('/profil')`) nous pouvons retrouver le Nom, Prénom, et l'adresse mail de l'utilisateur qui est connecté. On y retrouve également le mot de passe de l'utilisateur dont les lettres ont été remplacées par des étoiles afin de garder la confidentialité.

Comme expliqué précédemment. En fonction de ses droits, un utilisateur peut être sous trois statuts différents :

- administrateur
- validé
- basique

Lors de la création de son compte, l'utilisateur est sous le statut basique. Pour avoir accès à plus de fonctionnalités, il est nécessaire qu'un administrateur lui change son statut. Ainsi, c'est dans l'onglet "Mon Profil" qu'un administrateur peut avoir accès à la liste des utilisateurs enregistrés via un bouton "Gérer les niveaux des utilisateurs" (`@app.route('/validation')`). Sur cette page ('validation.html'), l'administrateur a possibilité grâce à des boutons de changer le niveau d'un utilisateur en "Validé" ou "Administrateur".

Dans 'Mes Abonnements' (`@app.route('/mesabonnements')`) s'affiche une liste de projets auxquels l'utilisateur est abonné.



Dans 'Mes projets' (@app.route('/mesprojets')) s'affichent tous les projets créés par l'utilisateur.

La barre 'Rechercher' permet de rentrer un ou plusieurs mots-clés. Et un algorithme de recherche permet de trouver les subs contenant ces mots-clés. Ces subs s'affichent ensuite dans une nouvelle page via @app.route('/results').

Lorsque l'on clique sur un projet, (@app.route('/sub/<id>') avec id représentant le numéro du projet), on est redirigé vers l'onglet 'Description' du sub.(viewsub.html') Sur cette page, on retrouve :

- la description du projet
- le créateur du projet
- le nombre d'abonnés
- un bouton pour s'abonner
- un bouton pour se désabonner
- les participants du projets
- un bouton pour participer
- un bouton pour ne plus participer.

Afin de cliquer sur 's'abonner' (@app.route("/<id>/abonnement")), il faut aussi être un utilisateur vérifié ou administrateur. De même, pour pouvoir participer à un projet (@app.route("/<id>/abonnement")), il est nécessaire d'y être abonné. Cliquer sur le bouton 'participer' (@app.route('/<id>/demande participation')) enverra une demande de participation au créateur du projet. Celui-ci peut ensuite accéder aux demandes reçues via l'onglet 'Demande de participations' (@app.route('/sub/<id>/demandes')) auquel lui seul a accès. Sur cette page, le créateur peut ensuite décider d'accepter ou de refuser un utilisateur parmi les participants. Le traitement de la requête enregistrée par les boutons se fait via les chemins @app.route('/<id>/accepter/<user>') et @app.route('/<id>/refuser/<user>') où id correspond au sub et user à l'id de l'utilisateur ayant fait la demande de participation.

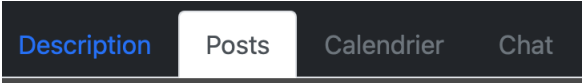


FIGURE 5.4 – Photo de la barre de navigation dans sub

Tout le monde a accès à l'onglet 'Posts'. À contrario seuls les utilisateurs qui participent au sub (ou l'ayant créé) peuvent avoir accès à l'onglet 'Chat' (app.route("/sub/<numsub>/chat", methods=["GET","POST"])). Le chat fonctionne comme un commentaire On peut remarquer que les onglets auxquels l'utilisateur n'a pas accès sont affichés en gris alors que les onglets auxquels il a accès sont affichés en bleu. Enfin l'onglet sur lequel il se trouve est affiché en blanc.

Si l'on clique sur l'onglet Post (app.route('/sub/<id>/post')), on retrouve l'ensemble des posts associés au projet avec les plus récent affichés en haut de la page. Pour chaque post, il s'affiche également :

- des boutons pour liker ou disliker un post ((@app.route('/like/<id>'))
- un form pour pouvoir laisser un commentaire (l'utilisateur doit être au moins vérifié, @app.route('/comment/<id>',methods=['post']))
- une case avec chaque commentaire que nous pouvons liker ou disliker.

De plus, si l'utilisateur est le créateur du projet ou si il fait parti des participants, il a la possibilité de créer un nouveau post via le menu déroulant de l'onglet 'Posts' qui donne accès à la page 'Créer un post' ('newpost.html'). Sur cette page, nous retrouvons un form nous demandant le titre et la description du post. Il y'a aussi possibilité d'insérer une image qui sera liée au post.

Nous voulions également implémenter un calendrier propre à chaque projet. Nous avons fait des recherches (page web expliquant comment faire un calendrier en boot strap [1]). Un début d'implémentation avait été effectué mais n'ayant pas complètement abouti, rien n'a été posté sur le Git.

### 5.3 Algorithmes de traitement

Lors de la création de notre projet, nous avons dû implémenter un certain nombre d'algorithmes sous le langage python afin de vérifier des conditions qui étaient nécessaires au bon fonctionnement de l'application.

Dans cette section, nous détaillerons le fonctionnement de nos algorithmes et leurs utilisations.

### 5.3.1 Fil d'actualité

Sur la page d'accueil de notre application se trouve un fil d'actualité. L'objectif est d'afficher les posts des subs auxquels l'utilisateur est abonné du plus récent au plus ancien. Nous avons donc implémenté un algorithme qui parcourt les subs dans la base de données et qui recherche ceux auxquels l'utilisateur est abonné. Il sélectionne alors tous les posts liés à ce sub et les trie par ordre décroissant de date de création.

De plus, nous avons souhaité donner la possibilité à l'utilisateur de le liker ou de le disliker à travers des boutons. En fonction du bouton cliqué, le programme modifie la base de donnée pour incrémenter ou décrémenter le compteur de likes. Ce vote est unique qu'il soit positif ou négatif.

En outre,

### 5.3.2 Fonction `allowed_image(filename)`

Cette fonction vérifie si l'image uploadé peut être accepté.

Pour cela, on vérifie deux critères, tout d'abord la présence d'un point et donc d'une extension dans le fichier et finalement si l'extension de ce fichier fait partie des extensions acceptables à savoir dans notre cas PNG, JPEG, JPG et GIF.

### 5.3.3 Vérification des contraintes

Les pages de subs sont caractérisées par un numéro unique. Afin de s'assurer qu'aucun utilisateur n'essaye d'accéder à une page qui n'existe pas, on vérifie

Test `id_sub` vérifie si le numéro de sub entré en paramètre existe bel et bien dans la base de données.

On peut ainsi empêcher les utilisateurs de rentrer des requêtes erronées manuellement en les redirigeant vers une page d'erreur.

### 5.3.4 Fonction `test_login()`

Le programme permet de savoir si l'on est connecté au site.

Pour cela, on regarde si l'id actuel de l'utilisateur existe dans la base de données.

### 5.3.5 Fonction `test_verif()`

Le programme vérifie si l'utilisateur connecté est vérifié ou non.

On regarde dans la base de données la valeur du niveau de l'utilisateur.

### 5.3.6 Fonction `is_owner(id,user)`

Le programme vérifie si pour un sub (ayant un numéro id) appartient à un utilisateur (ayant un numéro user).

Est sélectionné dans la table 'subs' les éléments ayant pour numéro de projet id et créés par user. Ils sont ceux qui appartiennent à l'utilisateur.

### 5.3.7 Fonction `est_abonne(id,user)`

Le programme regarde si un utilisateur de numéro user est abonné au sub de numéro id.

On sélectionne dans la table 'abonnements' les éléments ayant pour numéro de projet id et comme numéro de personne user.

### 5.3.8 Fonction `est_participant(id,user)`

Le programme regarde si un utilisateur de numéro user participe au projet de numéro id.

On regarde dans la table 'participants' les éléments ayant pour numéro de projet id et comme numéro de personne user. Si il existe, il correspond au numéro du sub auquel l'utilisateur s'est abonné et le numéro de l'utilisateur

### 5.3.9 Fonction a\_demande(id,user)

Le programme regarde si un utilisateur de numéro user a demandé à participer au projet de numéro id.

Sélectionne dans la table demande participation les éléments ayant pour numéro de projet id et comme numéro de personne user. Si cet élément existe, il correspond au numéro du sub et au numéro de la personne ayant demandé à s'abonner.

### 5.3.10 Fonction \_commentaire(i\_com,user)

La fonction regarde si un commentaire de numéro id\_com a déjà été likée ou dislikée par un utilisateur de numéro user.

Le programme prend le numéro du voteur dans la table 'Vote\_com' où l'id du commentaire vaut 'id\_com'.

Si la variable est non vide, alors on regarde si user est dans la variable.

- Si il y est, alors la fonction retourne (True, True) respectivement pour signifier qu'il y'a possibilité de mettre un like ou un dislike sous le commentaire.
- Sinon, la fonction sélectionne la valeur qui correspond au fait qu'un utilisateur ait liker ou disliker un commentaire dans la table Vote com avec le numéro de post qui vaut id post et le numéro du voteur qui vaut str(user). On rentre cela dans une deuxième variable.
- Si cette valeur est 'P' pour positive, la fonction retourne (False, True)
- Sinon elle retourne (True, False)

Sinon, la fonction retourne (True, True).

### 5.3.11 Fonction likepost(id post,user)

La fonction regarde si un post de numéro id post a déjà été likée ou dislikée par un utilisateur de numéro id.

Le programme se connecte premièrement à la base de données, créer un curseur, et sélectionne le numéro du voteur dans la table Vote post où l'id du post vaut id com. On met le résultat dans une variable.

Si la variable est non vide, alors on regarde si user est dans la variable.

- Si il y est, alors la fonction retourne (True, True) respectivement pour signifier qu'il y'a possibilité de mettre un like ou un dislike sous le post.
- Sinon, la fonction sélectionne la valeur qui correspond au fait qu'un utilisateur ait liker ou disliker un post dans la table Vote post avec le numéro de post qui vaut id post et le numéro du voteur qui vaut str(user). On rentre cela dans une deuxième variable.
- Si cette valeur est 'P' pour positive, la fonction retourne (False, True)
- Sinon elle retourne (True, False)

Sinon, la fonction retourne (True, True).

### 5.3.12 Fonction recommandation

La recommandation des subs se base sur le nombre de mots en commun avec ceux auxquels l'utilisateur est déjà abonné.

Pour cela, on récupère dans la base de données l'ensemble des mots, des titres, et des description d'un sub auquel un utilisateur est abonné. On retire de cette liste de mots les articles et déterminants usuels tels que "un,une,des,le,la,les" par exemple, pour éviter de fausser la recommandation et se concentrer sur les noms et les verbes qui portent le sens du projet. On regarde alors le nombre d'occurrences de ces termes dans le reste des subs du site, que l'on trie par fréquence d'apparition décroissante grâce à la fonction "sort" présente nativement dans python.

Si un utilisateur s'abonne à plusieurs projets l'algorithme procède ainsi, il fait une liste pour chaque sub, les fusionnes en une seule liste qui sera triée par la suite..

### 5.3.13 Autres fonctions

D'autres fonctions ont également été implémentée directement dans les routes. Elles ont le même fonctionnement que celles décrites précédemment.

## 5.4 Base de données

Nous avons fait plusieurs tables sur une base de données. Elles sont implémentées comme ci-dessous :

Nous avons été confrontés à certains problèmes avec la base de données. Nottament lors des commit sur le git. Nous avions une base de données commune et il y'avait certains conflits de merge lors des commit. Nous avons finalement fait un fichier python commun tel que tout le monde puisse initialiser la base de données et avoir une forme commune. Nous avons ensuite chacun fait ignorer par le git notre base de données.

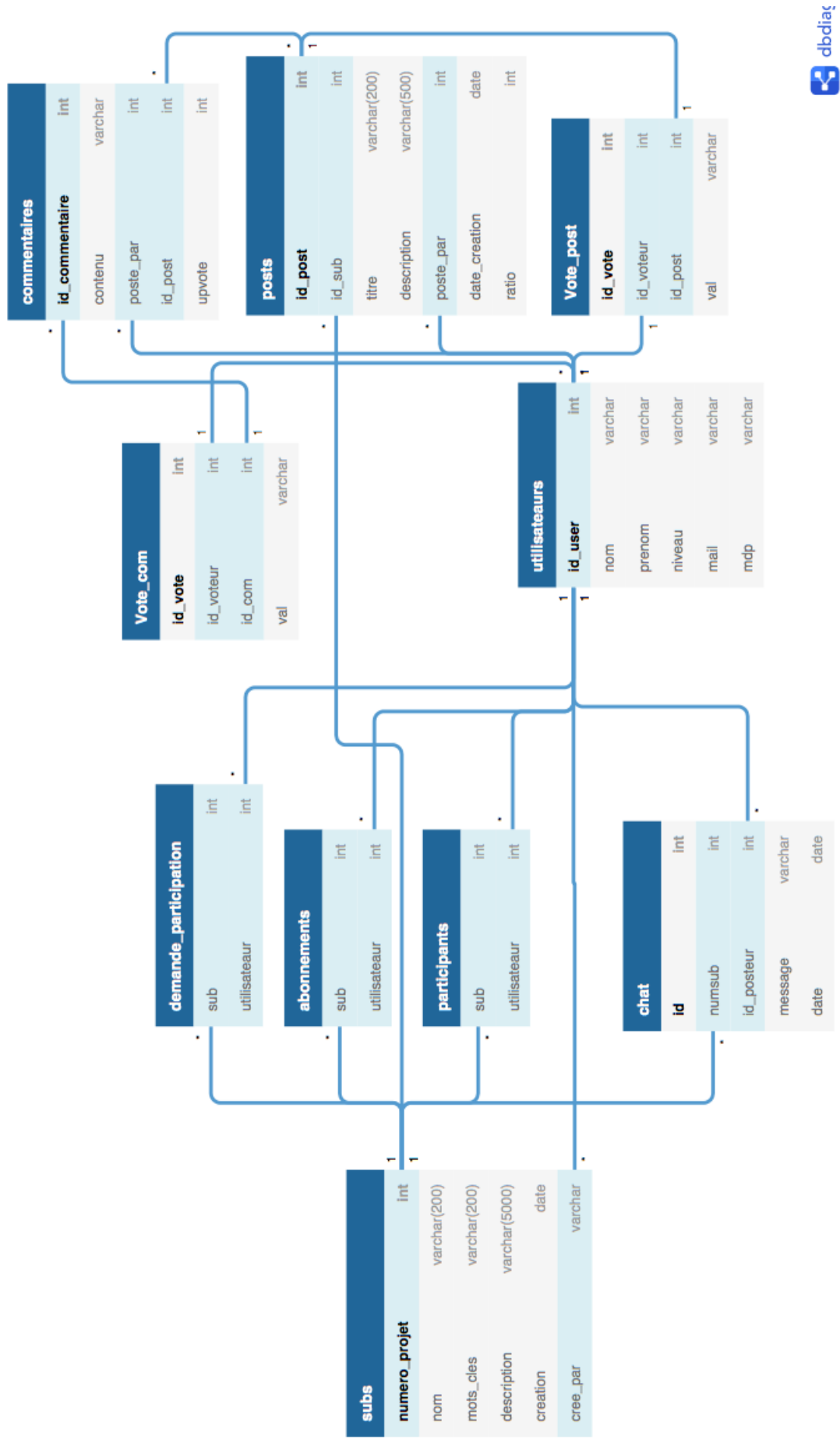


FIGURE 5.5 – Schéma relationnel de la base de données

# Chapitre 6

## Tests et performances

### 6.1 Complexité de l’algorithme de recommandation

#### 6.1.1 Analyse de complexité

L’algorithme de recommandation possède deux boucles for imbriquées. La première permet de passer par chaque sub alors que la deuxième permet de parcourir tous les mots clés et les mots du titre de chaque sub.

La complexité est donc linéaire suivant deux variables :

- le nombre de projets créés
- le nombre de mots utilisés dans les titres et mots clés

Le nombre de mots dans le titre et dans les mots clés étant limités à deux cents chacun, la complexité dépend surtout du nombre de sub créés. De plus, le nombre de sub augmente la complexité en temps alors que le nombre de mots ne le fait que très peu.

#### 6.1.2 Graphiques

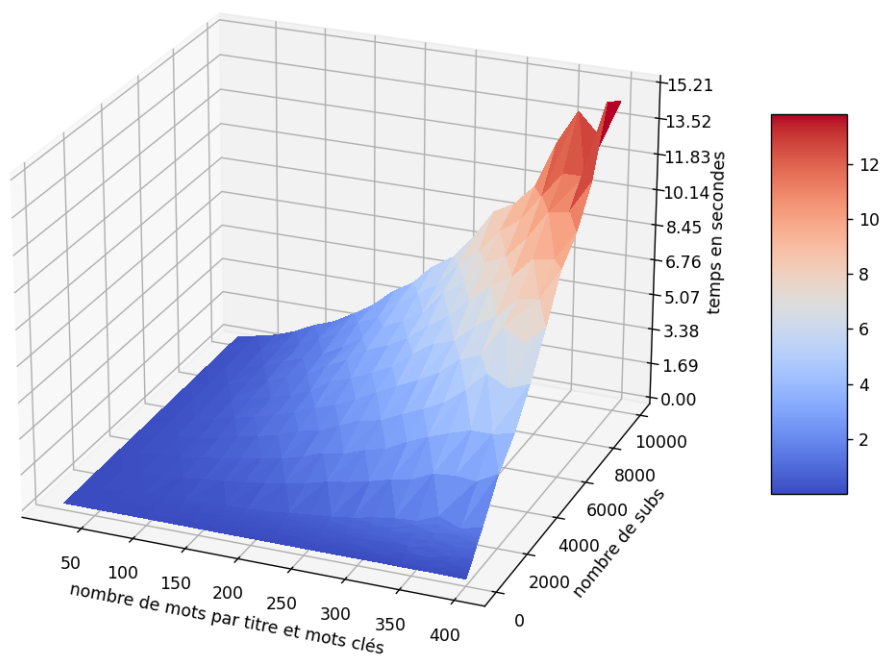


FIGURE 6.1 – Graphique 3D de la complexité de recommandation

On remarque sur le graphique ci-dessus que le temps d’exécution de l’algorithme en fonction du nombre de subs est linéaire. On le vérifie avec une régression linéaire. On obtient le graphique ci-dessous :

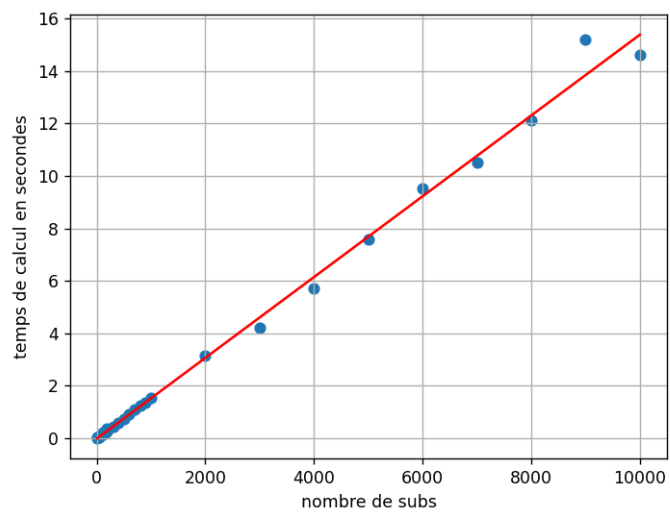


FIGURE 6.2 – Photo régression linéaire suivant le nombre de subs

Cependant, si l’on regarde le temps d’exécution en fonction du nombre de mots dans le titre et les mots clés, cela ne semble pas linéaire. On vérifie cela avec une régression linéaire sur le nombre de mots :

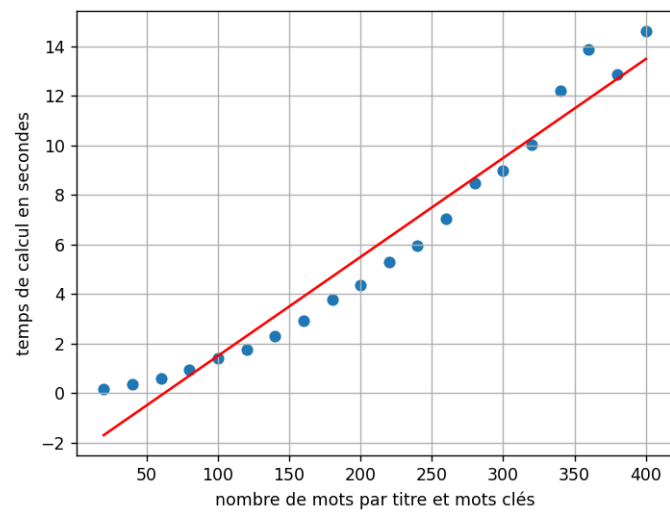


FIGURE 6.3 – Régression linéaire du temps d’exécution en fonction du nombre de mots compris dans le titre et les mots-clés par sub

La régression est éronnée, on ne suit pas une droite. Cela semble être plus proche d’un polynôme du second degré. On réalise donc la même régression mais sur un polynôme du deuxième degré, on obtient le résultat ci-dessous.

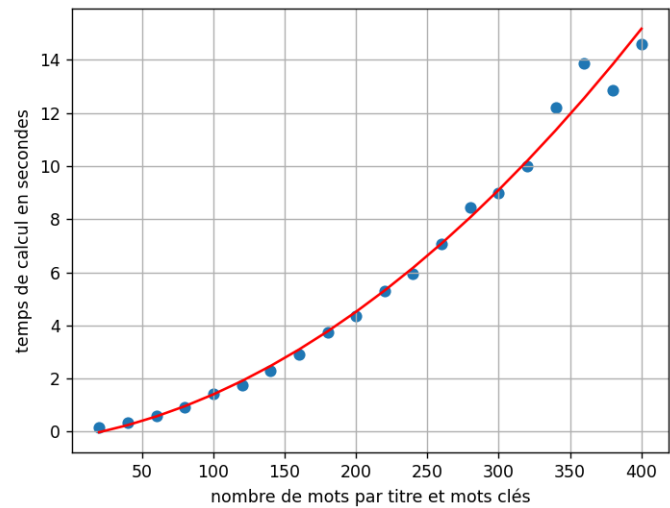


FIGURE 6.4 – Régression polynomiale du temps d’exécution en fonction du nombre de mots par sub

On en conclut donc que la complexité en temps suivant le nombre de mots se rapproche du polynôme :

$$-2.27 \times 10^{-1} \times x^2 + 8.92 \times 10^{-3} \times x + 7.39 \times 10^{-5}$$

## 6.2 Tests réalisés

### 6.2.1 Liste des Tests

Nous avons testé l’ensemble de nos fonctions Python grâce à Pytest en s’appuyant sur le principe du Right Biceps.

L’ensemble des fonctions suivantes ont été testées :

- a\_demande
- abonnement
- accepter
- accueil
- annuler\_participation
- chat
- commentaire
- demande\_participation
- demande
- desabonnement
- downlike
- downvote
- est\_abonne
- est\_participant
- is\_owner
- likepost
- newpost
- post\_commentaire
- recommandation
- refuser
- search\_results
- test\_id\_sub
- uplike
- upvote
- view\_sub
- viewpost

### 6.2.2 Problèmes liés aux Tests

Lors de ces tests, nous avons rencontré quelques difficultés sur les fonctions renvoyant à des pages web. Nous avons dû rechercher comment fonctionnait pytest dans ce genre de situations, notamment sur la documentation Flask [2])



### 6.2.3 Right BICEP

Afin de suivre le principe du Right BICEP, (Boundaries, Inverse Relationship, Crosscheck, Error, Performance). On veille à tester les fonctions qui nécessitent une connexion pour chaque type d'utilisateur (administrateur, valide, basique). De même, lorsque la fonction dépend d'un projet, on vérifie que chaque statut d'utilisateur (créateur, abonné, participant, aucun) est bien traité par la fonction.

On utilise des assertions sur les résultats de chaque fonction pour vérifier l'exactitude du résultat renvoyé est bien celui attendu. Cela permet de valider la partie Right de nos tests.

Concernant les tests de limites de nos fonctions (Boundaries), on insère des objets n'ayant pas le type prévu initialement ou n'existant pas dans nos bases de données.

Pour le critère d'inversabilité (Inverse Relationships), on souhaite s'assurer que chaque fonction est inversible. Cependant, nos fonctions utilisent des données parfois très similaires car rentrées par les utilisateurs. Ainsi, aucune fonction n'est inversible excepté la fonction affichant Mon Profil ou Mes Projets de par l'unicité du créateur et de l'adresse mail des utilisateurs.

Le critère de vérifications multiples (Cross-Check) est peu approprié à une application WEB, notre seul autre moyen de vérification est une vérification manuelle dans la base de données.

Nos fonctions possèdent toutes des erreurs forcées (Force Error). Pour gérer les cas où un paramètre ne correspond pas ou lorsqu'une requête ne termine pas, on renvoie l'utilisateur vers une page d'erreur correspondant à l'erreur.

Concernant les performances (Performance), nos fonctions ont un comportement adapté pour une utilisation par un nombre de personnes restreint. L'amélioration de cette performance est nécessaire pour un déploiement à plus grande échelle.

## Chapitre 7

# Bilan du projet

### 7.1 Conclusion

À l'issue de ce projet nous avons créé une application web de Civic tech qui doit être commanditée par les responsables de la commune pour pouvoir être mise en œuvre. Elle permettra aux habitants d'une ville de se regrouper autour de thématiques communes auto-déterminées. Elle favorise la discussion entre acteurs concernés grâce à un chat privé pour chaque projet. De plus elle permet d'informer les citoyens sur l'avancée des projets et des discussions grâce à la possibilité de créer des posts au sein des projets auxquels on peut joindre des images pour rendre le tout plus attractif. L'algorithme de recommandation permet également de proposer des projets similaires pour renforcer l'engagement citoyen sur divers sujets.

Pour mener ce projet à bien, nous avons employé divers outils de management. Le PDCA nous a permis d'organiser les grandes phases de notre projet là où les todo list élaborées à l'aide de brainstorming ont permis de dégager les fonctionnalités essentielles de notre application. De plus, la division en lots de celles-ci nous a permis de nous répartir les tâches à effectuer à l'aide d'une matrice RACI et de planifier ces tâches dans le temps grâce à un diagramme de Gantt.

Cependant tout ne s'est pas déroulé comme prévu. En effet, toutes les fonctionnalités prévues initialement n'ont pas pu être implémentées par manque de temps. Nous avons sous-estimé le temps que prendrait des composantes essentielles du projet telles que la phase de test et la rédaction du rapport. Certaines menaces identifiées dans notre matrice SWOT comme l'influence des fêtes de fin d'années nous ont plus impacté que prévu. Des réunions le 24 et 31 décembre ont été annulées et les journées du 1er et 2 janvier censées être travaillées ne l'ont pas été. Nous aurions dû commencer la phase de programmation plus tôt pour nous donner une plus ample marge de manœuvre. De plus, malgré une implication commune et sérieuse de tous les membres du groupe, les niveaux disparates en programmation ont conduit certains membres à être extrêmement présents sur cette partie. Pour compenser cela, les personnes ayant un niveau plus faible ont effectué une plus grosse part de la gestion de projet.

Malgré tout cela, notre application comporte la plupart des fonctionnalités prévues. Cependant elle peut être améliorée de plusieurs façons. Tout d'abord en intégrant du javascript à la partie chat pour la rendre plus fonctionnelle et agréable à utiliser. De façon plus générale, du javascript allié à un meilleur css rendrait le site plus beau. Du point de vue des fonctionnalités, la présence d'un calendrier dans chaque sub permettrait de garder une trace et d'informer plus facilement les utilisateurs des actions menées tout au long de la vie du projet. La possibilité de s'abonner à des utilisateurs pourrait également être implémentée. En effet, lorsqu'un citoyen est moteur et crée beaucoup de projets, on peut vouloir suivre les projets et idées de ce dernier. Pour améliorer la discussion entre utilisateurs et directeur d'un projet, un sondage pourrait être ajouté dans les posts. La possibilité de se déconnecter de sa session n'a pas été mise mais fait partie des fonctionnalités prioritaires en terme d'ajout. On peut également imaginer l'intégration d'une carte pour se renseigner plus simplement sur les lieux des événements à venir. L'adresse mail fournie pour vérifier le compte pourrait aussi être utilisée pour envoyer des notifications si besoin.

## 7.2 Conclusions personnelles

### 7.2.1 Bourdais Rémi

Je voudrais tout d'abord dire un grand merci à mes camarades pour avoir apporté de la bonne humeur tout au long du projet et pour l'entraide qui existe dans le groupe. En dépit des niveaux variés sur les différents aspects du projet chacun a su utiliser ses forces pour faire avancer le projet et aider les autres pour les faire progresser. L'organisation du code dans le projet n'a pas été optimal notamment le fichier app bien trop volumineux et difficile à comprendre par les personnes extérieures. Malgré cela tout le monde a su se retrouver dans cette organisation, c'est notre plus grande piste d'amélioration.

Le projet n'a commencé à avancer que tardivement car les examens sont passés en priorité pour tout les membres du groupe.

On a donc du être efficace et autonome pendant les fêtes pour continuer à développer l'application. Les réunions à distance se sont bien déroulées malgré les disponibilités limitées de chacun. Toutes fois demarrer la phase d'implémentation plus tôt n'aurait pu être que bénéfique. De même pour le temps prévu pour rédiger le rapport que nous avons sous estimé.

J'ai apprécié travailler sur ce projet : le sujet est original et très vaste, on a pu l'interpréter, modéliser et développer ensuite l'application. Ce projet m'a permis de progresser en programmation WEB que ce soit en HTML ou sur la partie serveur. Et de découvrir le Latex et l'application des principes de gestion de projet.

Ce fut pour moi une première expérience de développement en groupe et d'utilisation d'un git avec plusieurs personnes. Cela m'a amené bon nombre de problèmes de Merge et de doublons dans le code. Mais aussi dans la répartition du travail, j'ai dépassé accidentellement sur le travail à plusieurs reprises, c'est l'amélioration personnelle sur laquelle je dois travailler.

Je devrais aussi veiller à réaliser des commits moins conséquent et plus réguliers plutôt que juste en fin de session de travail.

### 7.2.2 Clériot Louis

Tout d'abord, le projet s'est déroulé sans problème d'entente au sein du groupe. Nous nous connaissions déjà avant le début du projet, ce choix a donc été effectué en toute connaissance de cause. À la suite de la première réunion quelques tensions sont apparues entre deux membres du groupe mais une discussion postérieure a permis d'éviter tout conflit et le projet a pu continuer dans de bonnes conditions jusqu'à la fin. Tous les membres sont restés sérieux et impliqués jusqu'à la fin malgré une mauvaise gestion du temps par moment qui a pu rendre stressante la réalisation des livrables.

D'un point de vue personnel, ce n'était pas mon premier projet étant en master l'année précédente. J'ai pu remarquer que pour certaines personnes il était compliqué de se cantonner à une liste de tâche prédéfinis à l'avance ce qui a entraîné des modifications des todo lists. Cela peut donc faire partie des améliorations possibles dans la gestion de projet du groupe. De mon côté, il ressortirait un léger manque de communication sur l'avancée de mon travail, comme peuvent en témoigner le faible nombre de commits. Pour la suite, il faudrait réaliser des commits plus régulièrement et non pas seulement à la fin d'une fonctionnalité. Ce n'était pas mon premier projet certes, mais mon premier en matière de programmation. J'ai découvert un type de rédaction différent de ce que j'avais l'habitude de produire (description de fonction), une ambiance différente également étant habitué à des groupes où tout le monde ne semble pas réellement concerné.

L'utilisation d'un git a plusieurs a pu se montrer plus difficile que je ne le pensais mais l'aide de Rémi et de mon frère m'a permis de surmonter ces difficultés. La réalisation de l'algorithme de recommandation et l'ajout d'image grâce à un formulaire m'a permis de fortement m'améliorer en html, en jinja et en python. De plus, la recherche de solution à un problème grâce à internet m'a permis de me familiariser avec stack overflow, formidable outil que je n'avais jamais utilisé auparavant. J'ai également appris de nombreuses choses sur la civic tech grâce à l'état de l'art. Les applications web produites dans le monde dans ce domaine sont très intéressantes mais malheureusement trop peu connues du grand public. J'ai appris que de tel dispositif existait dans la ville dans laquelle j'ai habité pendant 4 ans durant mes précédentes études sans jamais en avoir entendu parlé avant.

J'ai apprécié réaliser un projet complet ayant un réel objectif du début à la fin. Programmer et répartir les tâches associées à l'ambiance du groupe projet, m'ont confirmé mon choix pour ma réorientation dans le domaine de l'informatique.

### 7.2.3 Serrand Coralie

J'aimerais mettre en valeur la bonne entente de notre groupe qui aura rendu plaisante ce travail. J'ai par ailleurs beaucoup apprécié nos séances de travail de groupe qui m'ont permis d'être efficace et motivée pendant toute la durée du projet.

Pendant toute la durée du projet, nous avons organisé plusieurs réunions dans laquelle nous exposons nos idées. Il a donc fallu se confronter à l'avis des autres. Il était aussi nécessaire de trouver un terrain d'entente, d'écouter les autres et de parfois savoir abandonner ses idées. Par ailleurs, étant partie en voyage en Suède avec ma famille pendant les vacances de Noël qui représentait la plus grande partie de notre phase de programmation, il m'a fallu fournir une certaine charge de travail malgré un temps libre limité.

Néanmoins, observer le projet qui prend forme petit à petit, et se sentir monter en compétences motive grandement. Cependant cette motivation, alliée à une volonté de toujours vouloir faire plus ont pour conséquence un risque d'empiètement sur les tâches des autres. Je me suis aussi rendu compte qu'il faut parfois savoir s'arrêter sur certaine tâche et prioriser d'autres qui le sont plus importante malgré le fait qu'on aimerait pouvoir développer d'autres fonctionnalités.

En outre, au niveau des livrables, les programmes pourraient être plus propre avec plus de fonctions intermédiaires et de commentaires afin d'améliorer la clarté du code. Dans la même optique, il me semble que le design de l'application pourrait être grandement amélioré avec plus de CSS.

Au final, ce projet aura été très formateur. Il m'aura apporté de nouvelles compétences en web, en traitement de base de données et en gestion de projet. Ce fut ma première expérience de programmation en groupe et ce fut le premier projet pour lequel j'ai dû rédiger un rapport. Il m'aura appris à savoir m'adapter et à me former moi-même sur mes lacunes.

Enfin, ce projet m'aura aussi permis de me confronter dans mon choix d'orientation. Je suis plutôt satisfaite du travail que j'ai effectué et implémenter cette application aura été très plaisant.

### 7.2.4 Tejedor Manon

Premièrement, j'aimerais remercier mes camarades avec qui le projet fut agréable. Chacun a su placer ses points forts au service de celui-ci et les lacunes de chacun furent comblées par l'entraide qui régnait au sein du groupe.

Malgré le fait que le groupe ait eu du mal à se lancer, les semaines d'après furent dynamiques et chaque membre du groupe a su motiver les autres. D'où le fait que nous n'ayons pas ressenti le besoin de nommer de chef de projet et opté pour une organisation fonctionnelle.

Pour autant, nous avons eu quelques difficultés, nottament liées aux vacances. Juste avant celles-ci nous avons délaissé en partie le projet afin de nous concentrer sur nos partiels. Ce qui a eu pour conséquences qu'une partie du projet a dû se faire en distanciel. Cela a été assez régulier mais nous avons dû décaler quelques réunions du fait des fêtes. Un autre problème, technique cette fois ci, a quelque fois empêcher la progression. Celui-ci provient de la mauvaise maîtrise du git. Nous avons rencontrés des problèmes qui n'étaient pas forcément évident à résoudre.

Du point de vue des améliorations personnelles, les derniers rapports n'ont pas été rédigés toujours en temps et en heure, ceux dont j'étais chargés aurait dû être rédigés au bon moment.

Il aurait également fallu mieux répartir le temps tout au long du projet.

J'ai rencontré, de plus, quelques difficultés lors de la partie d'implémentation. Malgré mon implication et mon travail, j'avais l'impression de ne pas fournir assez de travail par rapport aux autres (ce qui s'est avéré être faux, malgré que certaines personnes aient effectivement plus de facilités), ce qui me stressais. Il faudrait que j'apprenne à gérer ce stress lors de prochains projets

Je pense avoir développé des compétences, à titre personnel, dans certains domaines notamment en programmation web en Flask. Je ne maîtrisais également pas du tout le code

en Latex et les tests sous Pytest. J'ai réussi à faire les tests et à rédiger une partie du rapport.

J'ai également apprécié le travail en groupe afin de réaliser un travail informatique.

# Chapitre 8

## Annexes

### 8.1 Comptes-rendus des réunions

#### 8.1.1 18 novembre 2021

##### Compte-rendu n°1 - PPII : groupe E.14

Type de réunion : Réunion technique	Lieu : Salle 0.10
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon</li></ul>	Le 18/11/21 De 17H00 à 18H00 Durée de la réunion : environ 1h00

**Ordre du jour :**

- Découvrir ce qui a déjà été fait en terme de Civitech
- Faire un état de l’art
- Brainstorming des premières idées

**Sujets abordés :**

- Chacun a regardé les différentes pages liées aux civitech trouvées par les autres et a fait part de ses impressions
- Les points positifs et négatifs de ces applications ont été soulevés
- Nous avons relié ces points avec le sujet
- À partir de cela, nous avons eu des idées que nous avons proposées et discutées.

**Prochaine réunion :**

Fixée pour le 23/11/21

**TO DO LIST :**

Tâches à faire	Par qui ?	Pour quand ?
Continuer les recherches de Civitech	Coralie Louis Rémi Manon	Prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h

#### 8.1.2 23 novembre 2021

##### Compte-rendu n°2 - PPII : groupe E.14

Type de réunion : Réunion technique	Lieu : Amphithéâtre Nord
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon</li></ul>	Le 23/11/21 De 16H00 à 17H30 Durée de la réunion : environ 1h30

Ordre du jour :

- Aborder les idées présentées lors de la dernière réunion
- Explorer de nouvelles possibilités
- Esquisser le format de notre application web

Sujets abordés :

L'idée de mettre en place sur l'application web de différentes échelles politiques pour un même utilisateur fut la première idée abordée. Le fait de faire un fil d'actualité a été posé comme possibilité. C'est à dire qu'en fonction des posts likés par l'utilisateur, des posts adaptés lui seraient proposés. L'idée de faire un forum a également été évoquée. Il serait possible à l'utilisateur de pouvoir poser des questions et répondre à d'autres. Nous avons également abordé la possibilité de s'abonner à certains projets pour un utilisateur dont il voudrait suivre l'avancement. Ceci a été relié à une idée de questionnaire lors de la première visite afin de savoir ce qui pourrait potentiellement intéresser le plus l'utilisateur. Une possibilité de dons a été évoquée mais rapidement abandonnée liée au fait que l'argent ne devrait pas interférer avec le projet au vue de sa visée de démocratie participative.

Nous avons finalement retenu l'idée d'une application web qui pourrait gérer les projets de différents dirigeants à différentes échelles (en lien avec les idées initiales). Nous voulions initialement l'appliquer à différents types de campagnes. Idée rapidement laissée tomber puisque les campagnes sont des moments assez courts et précis. Il faudrait que l'utilisateur de l'application puisse en avoir un usage régulier. De plus se poserait le problème d'impartialité quant à l'affichage équitable parmi les différents candidats. L'idée serait donc d'interroger l'utilisateur sur les points qui l'affectent (exemple : impôts, écologie...) particulièrement. Cela se ferait via une sélection, à la première ouverture de l'application, de plusieurs thèmes. Comme évoqué précédemment, l'application aurait une page principale sur laquelle défilerait un fil d'actualité. Celui-ci serait composé de projets pour lesquels l'utilisateur à un intérêt (recueillis via la sélection). Les projets dans lesquels les mots clés indiqués par l'utilisateur reviendraient seraient mis en avant. Il serait possible pour un utilisateur de ne plus vouloir que l'on lui recommande les projets d'une certaine personne publicatrice de projets. L'application devrait être capable de sélectionner les grands axes d'un projet pour pouvoir retransmettre au mieux à l'utilisateur les différents projets sur son fil d'actualité.

L'idée est venue de pouvoir échanger avec le créateur d'un projet sur les doutes émis par celui-ci. Il y aurait donc un espace sur lequel les différents utilisateurs puissent se répondre. Pour revenir sur l'espace d'échange, si un projet est adopté par une organisation, il y aurait une possibilité d'ouvrir un conseil de Sages (constitué de présidents d'associations par exemple). Cette fonctionnalité nécessiterait la mise en place d'un mot de passe pour un utilisateur.

<https://fvcs.fr/les-sages-cest-quoi/>

Enfin un système de recherche serait mis en place afin de permettre à un utilisateur de trouver un projet plus précis.

Est ensuite venue la problématique concernant les personnes qui postent les projets, une fonctionnalité serait nécessaire pour leur permettre de tout rentrer correctement. L'idée de faire un support vidéo afin de permettre aux gens de mieux comprendre la politique par exemple a également été citée.

Prochaine réunion :

Fixée pour le lendemain le 24/11/21

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Faire des recherches sur les outils utiles à la gestion de projet dans le cadre du projet	Coralie Louis	Jusqu'à la prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Recherche sur les algorithmes nécessaires aux fonctionnalités de l'application	Rémi	Jusqu'à la prochaine réunion

8.1.3 24 novembre 2021

Compte-rendu n°3 - PPII : groupe E.14

Type de réunion : Réunion technique	Lieu : Salle S.0.10
Présents : — Bourdais Rémi — Clériot Louis — Serrand Coralie — Tejedor Manon	Le 24/11/2021 De 16H00 à 17H00 Durée de la réunion : environ 1H00

Ordre du jour :

- Développer les points abordés précédemment.
- Commencer la feuille de conception
- Définir les besoins en base de données, algorithmique, et en web de notre application

Échanges :

Ont été présentés au tableau les différentes bases de données dont nous aurions besoin pour notre application web (définie dans le CR précédent). Plusieurs versions ont été proposées et ont abouti sur une version qui peut encore évoluer. Les tables sont les suivantes :

- Utilisateurs comprenant les attributs suivants : nom, login, mdp, le droit de poster des projets, le mail, le type d'utilisateur (), abonnements à des projets, organisateur de projets, participant à un projet, et les projets postés
- Projets : Nom, numéro du projet, Personne ayant posté le projet, type de projet (idée ou projet validé), et le domaine ou les mots clés concernant le projet.
- Un calendrier avec : le numéro de projet associé, la date, le nom, et l'heure de l'évènement.
- Une liste de tâche à faire pour un projet : numéro de projet, tâche, sujet
- Une table message :

La présentation de l'application web a également été discutée. Il a été convenu qu'il y aurait une grande partie de la page dédiée au fil d'actualité. Il a été discuté de la possibilité d'un volet avec les discussions de groupe qui seraient fixées sur la page principale ou alors qui défilerait avec celle-ci. L'utilisation d'un menu a été abordée mais nous n'avons pas su choisir entre cette idée ou alors une barre située en haut de l'écran avec les différentes fonctionnalités de l'application.

Il est apparu que nous devons modérer les messages des utilisateurs.

Finalement, nous avons contacté une personne syndicaliste spécialisée dans la démocratie participative. Nous lui avons exposé nos idées puis, elle nous a expliqué que notre idée d'application manquait d'un espace de prise de décision dans lequel les personnes qui suivent les projets pourraient prendre des décisions vis -à -vis de celui-ci. Ainsi cela ferait avancer le projet, voire le clôturerait.

Décisions :



Nous avons donc choisis de prendre en compte ses conseils et d’ajouter un espace sur lequel les gens concernés par le projet mettraient des notes aux différentes idées du projet.

Prochaine réunion :

Fixée pour le 25/11/21

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Recherche algorithme	Rémi	Jusqu’à la prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Rédiger la feuille de conception de l’application	Coralie	Jusqu’à la prochaine réunion

8.1.4 25 novembre 2021

Compte-rendu n°4 - PPII : groupe E.14

Type de réunion : Réunion technique	Lieu : Cafétéria
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon (partie plus tôt car cours d’espagnol)</li></ul>	Le 25/11/21 De 14H30 à 16H30 Durée de la réunion : environ 2h

Ordre du jour :

- Revenir sur la rédaction de la feuille de conception
- Commencer le schéma de l’application
- Refaire le point sur la rédaction de l’état de l’art

Sujets abordés :

- Relecture et reformulation de certains points de la production textuelle sur la conception de l’application web.
- Ajout de certains oublis comme l’onglet de recommandations dans la conception
- Définition au propre des bases de données nécessaires
- Esquisse du schéma au brouillon des relations entre les pages webs et les bases de données
- Rédaction au propre de l’état de l’art et finalisation de celui-ci

8.1.5 20 décembre 2021

Compte-rendu n°5 - PPII : groupe E.14

Type de réunion : Réunion de Chantier	Lieu : Distanciel (sur discord)
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon</li></ul>	Le 20/12/21 De 18H20 à 19H00 Durée de la réunion : environ 0H40

Ordre du jour :

- Vérifier l’avancement de chacun
- Résoudre les blocages

Sujets abordés :

Chacun a pris la parole tour à tour pour dire ce qui avait été fait.

Coralie	Fait : — Création du Layout En cours : — Page d’accueil
Louis	Fait : — Création du Layout — Algorithme de recommandation des projets En cours : — Page d’accueil
Rémi	Fait : — Création d’un Sub — Recherche d’un Sub ( à améliorer, recherche très basique) — Algorithme de recommandation des projets En cours : — Page de sub
Manon	Fait : — Création d’un Sub En cours : — Page parcourir

Prochaine réunion :

Fixée pour le 22/12/21

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Créer un nouveau post	Coralie	Prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Notation des posts	Louis Rémi	Jusqu’à la prochaine réunion
Finir le travail en cours	Tous	Jusqu’à la prochaine réunion

8.1.6 22 décembre 2021

Compte-rendu n°6 - PPII : groupe E.14

Type de réunion : Réunion de Chantier	Lieu : Distanciel (sur discord)
Présents : — Bourdais Rémi — Clériot Louis — Serrand Coralie — Tejedor Manon	Le 22/12/21 De 19H10 à 19H45 Durée de la réunion : environ 0H35

Ordre du jour :

- Vérifier l’avancement de chacun
- Résoudre les blocages

Sujets abordés :

Chacun a pris la parole tour à tour pour dire ce qui avait été fait.

Coralie	Fait : <ul style="list-style-type: none"><li>— Afficher les posts sur la page d’accueil</li><li>— Créer un nouveau post</li></ul>
Louis	Fait : <ul style="list-style-type: none"><li>— Page d’accueil</li><li>— Notation des posts</li></ul>
Rémi	Fait : <ul style="list-style-type: none"><li>— Notation des posts</li><li>— Page d’inscription</li><li>— Page de connexion</li></ul>
Manon	Fait : <ul style="list-style-type: none"><li>— Page parcourir</li></ul>

Ont été ajoutées également des tâches dans la matrice RACI en fonction des nouveaux besoins du projet et elles ont été réparties parmi les membres.

Prochaine réunion :

Fixée pour le 24/12/12

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Mettre des commentaires sous les posts	Rémi	Jusqu’à la prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Faire la page qui affiche le profil	Manon	Jusqu’à la prochaine réunion
Ajouter un bouton s’abonner	Coralie	Jusqu’à la prochaine réunion
Ajouter dans la base de données qui est abonné et qui participe au projet.	Coralie	Jusqu’à la prochaine réunion
Faire en sorte que le créateur du projet donne l’autorisation	Manon	Jusqu’à la prochaine réunion
Modifier la base de donnée pour savoir qui est administrateur ou utilisateur vérifié ou non	Rémi	Jusqu’à la prochaine réunion
Avoir accès au login sur toutes les pages de navigation	Rémi	Jusqu’à la prochaine réunion
Images sous les posts	Louis	Jusqu’à la prochaine réunion
Sondages sous les posts	Louis	Jusqu’à la prochaine réunion

8.1.7 26 décembre 2021

Compte-rendu n°7 - PPII : groupe E.14

Type de réunion : Réunion de Chantier	Lieu : Distanciel (sur discord)
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon</li></ul>	Le 26/12/21 De 20H30 à 21H00 Durée de la réunion : environ 0H30 (La réunion du 24 n'a pas pu avoir lieu liées aux fêtes de Noël)

Ordre du jour :

- Vérifier l'avancement de chacun
- Résoudre les blocages

Sujets abordés :

Chacun a pris la parole tour à tour pour dire ce qui avait été fait.

Coralie	Fait : <ul style="list-style-type: none"><li>— Ajouter dans la base de données qui est abonné et qui participe au projet.</li><li>— Ajouter un bouton s'abonner</li></ul>
Louis	Fait : <ul style="list-style-type: none"><li>— Images sous les posts</li></ul> Non fait : <ul style="list-style-type: none"><li>— Sondage sous les posts</li></ul>
Rémi	Fait : <ul style="list-style-type: none"><li>— Avoir accès au login sur les pages de navigation</li><li>— Modifier la base de donnée pour savoir qui est administrateur ou utilisateur vérifié ou non</li></ul> Non fait : <ul style="list-style-type: none"><li>— Mettre des commentaires sous les posts</li></ul>
Manon	Fait : <ul style="list-style-type: none"><li>— Faire la page qui affiche le profil</li></ul> Non fait : <ul style="list-style-type: none"><li>— Faire en sorte que le créateur du projet donne l'autorisation</li></ul>

Nous avons discuté des différents niveaux utilisateurs et des accès aux différentes parties du site en fonction du niveau utilisateur. (Admin, validé, basique).  
Les types d'utilisateurs au sein même d'un projet ont été discutés. (Abonné, participant, créateur).

Prochaine réunion :

Fixée pour le 28/12/12

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Sondages des posts	Louis	Jusqu'à la prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Afficher les abonnements personnalisés (en fonction du profil) sur la page d'accueil	Coralie	Jusqu'à la prochaine réunion
Afficher les projets auquel une personne est abonnée dans Mes abonnements	Manon	Jusqu'à la prochaine réunion
Afficher les projets créer par une personne dans Mes projets	Manon	Jusqu'à la prochaine réunion
Algorithme de recherche amélioré	Coralie Manon	Jusqu'à la prochaine réunion
Algorithme de recommandation	Louis	Jusqu'à la prochaine réunion
Gérer les accès en fonction des niveaux utilisateurs	Rémi	Jusqu'à la prochaine réunion
Attribuer les pages aux bons types d'utilisateurs (abonné, participant, créateur)	Rémi	Jusqu'à la prochaine réunion
Créer la page calendrier	Coralie Manon	Jusqu'à la prochaine réunion
Créer la page Chat	Louis Rémi	Jusqu'à la prochaine réunion

8.1.8 28 décembre 2021

Compte-rendu n°8 - PPII : groupe E.14

Type de réunion : Réunion de Chantier	Lieu : Distanciel (sur discord)
Présents : — Bourdais Rémi — Clériot Louis — Serrand Coralie — Tejedor Manon	Le 28/12/21 De 21H00 à 21H40 Durée de la réunion : environ 0H40

Ordre du jour :

- Vérifier l'avancement de chacun
- Résoudre les blocages

Sujets abordés :

- A tour de rôle chacun a dit ce qu'il a fait depuis la dernière fois
- Problème de Coralie avec sa VM qui ne fonctionne plus
- Problèmes d'affichage de mes abonnements
- Recherches sur l'implémentation d'un calendrier par Manon et Rémi
- Échanges sur le mail de M. Festor à propos du projet
- Problèmes des adresses mails multiples de Manon
- Un problème a été repéré dans mes subs

Coralie	Fait : — Afficher les abonnements personnalisés (en fonction du profil) sur la page d'accueil Recherches : — Algorithme de recherche amélioré
Louis	Fait : — Algorithme de recommandation En cours : — Créer la page Chat Non fait — Sondages des posts
Rémi	Fait : — Gérer les accès en fonction des niveaux utilisateurs — Attribuer les pages aux bons types d'utilisateurs (abonné, participant, créateur) En cours : — Créer la page Chat Recherches : — Créer la page calendrier
Manon	Fait : — Afficher les projets auquel une personne est abonnée dans Mes abonnements — Afficher les projets créer par une personne dans Mes projets Recherches : — Créer la page calendrier

Prochaine réunion :

Fixée pour le 30/12/12

TO DO LIST :

Tâches à faire	Par qui ?	Pour quand ?
Mettre les commentaires sur la page d'accueil	Louis	Jusqu'à la prochaine réunion
Rédiger le compte rendu	Manon	Dans les 48h
Mettre les photos pour chaque sub	Louis	Jusqu'à la prochaine réunion
Mettre les likes sur chaque sub	Rémi	Jusqu'à la prochaine réunion
Modifier la page d'acceuil pour que seulement les abonnements s'affichent	Coralie	Jusqu'à la prochaine réunion
Afficher le nombre de like à côté d'un post	Coralie	Jusqu'à la prochaine réunion
Voir d'où vient le problème dans Mes Subs.	Manon	Jusqu'à la prochaine réunion

8.1.9 30 décembre 2021

Compte-rendu n°9 - PPII : groupe E.14

Type de réunion : Réunion de Chantier	Lieu : Distanciel (sur discord)
Présents : <ul style="list-style-type: none"><li>— Bourdais Rémi</li><li>— Clériot Louis</li><li>— Serrand Coralie</li><li>— Tejedor Manon</li></ul>	Le 30/12/21 De 21H00 à 21H35 Durée de la réunion : environ 0H35

Ordre du jour :

- Vérifier l’avancement de chacun
- Résoudre les blocages

Sujets abordés :

Chacun a pris la parole tour à tour pour dire ce qui avait été fait.

Coralie	Fait : <ul style="list-style-type: none"><li>— Modifier la page d’accueil pour que seulement les abonnements s’af-fichent</li><li>— Afficher le nombre de like à côté d’un post</li></ul>
Louis	Fait : <ul style="list-style-type: none"><li>— Mettre les photos pour chaque sub</li></ul>
Rémi	Fait : <ul style="list-style-type: none"><li>— Mettre les likes sur chaque sub</li></ul>
Manon	Fait : <ul style="list-style-type: none"><li>— Voir d’où vient le problème dans Mes Subs.</li></ul> Recherches : <ul style="list-style-type: none"><li>— Recherches sur le Calendrier (n’ a pas complètement abouti)</li></ul>

TO DO LIST :

Faire les tests du site avec la méthode Right BICEP

8.2 Gestion de projet

LOT	sous-lot	Manon	Coralie	Rémi	Louis
Sub	créer	R		R	
	recherche			R	
	faire la page				R
	bouton abonnement		R		
	bouton participer		R		
	demande		R		
POST	créer		R	C	
	afficher		R		C
	notation			R	R
	images				R
	sondage				R
	like			R	R
COMMENTAIRES	sous poste			R	
	likes			R	
RECOMMANDATIONS	projet			R	R
	post	R	R	C	C
CHAT	accéder				R
	poster message				R
	recevoir message				R
CALENDRIER	accéder	R			
	poster événements	R			
LOGIN	vérif niveau			R	
	connexion			R	
	inscription			R	
	profil	R	C		
	autorisation	C	R		
TEMPLATE	page d'accueil		R		
	layout		R	C	R
PROFIL	abonnement	R	C		
	projet	R	C		
BD	créer	R	R	R	R
	remplir	R	R	R	R

FIGURE 8.1 – Matrice RACI



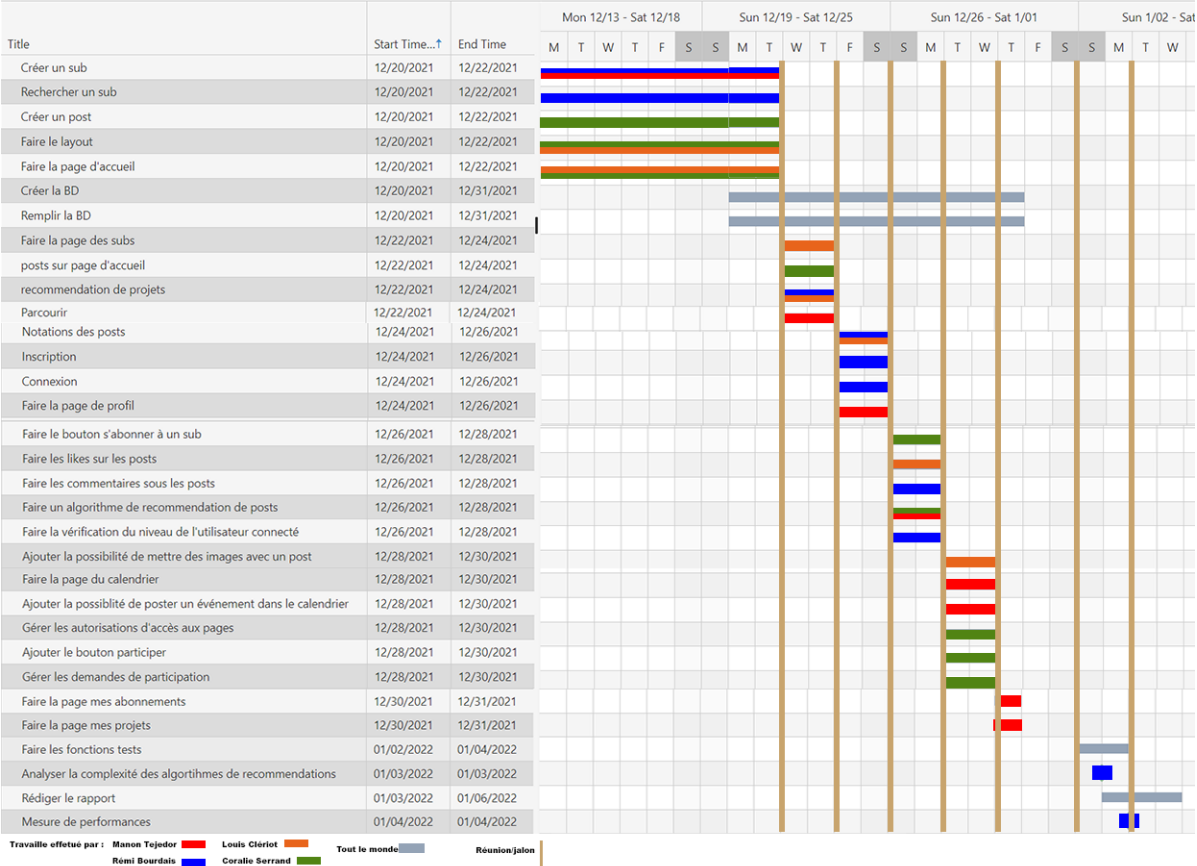


FIGURE 8.2 – Diagramme de Gantt

8.3 Sources

- <https://openclassrooms.com/fr>
- <https://stackoverflow.com>
- <https://roytuts.com>

# Bibliographie

- [1] Page expliquant comment faire un calendrier en bootstrap.  
<https://roytuts.com/bootstrap-calendar-events-demo-using-python-flask-mysql/>.
- [2] Site de la documentation flask. <https://flask.palletsprojects.com/>.
- [3] Site officiel de capcollectif. <https://cap-collectif.com/>.
- [4] Site officiel de democracyos. <https://democracyos.eu/>.
- [5] Site officiel de regards citoyens. <https://www.regardscitoyens.org/panell1-1/>.
- [6] Site officiel de solucracy. <https://solucracy.com/>.
- [7] Site officiel de vooter. <https://vooter.co/>.
- [8] Site officiel de wesignit. <https://www.wesign.it/>.
- [9] Site officiel du mooc. <https://mooc.gestiondeprojet.pm/>.
- [10] Tatiana de Feraudy. *Cartographie de la Civic Tech en France*. Observatoire des civic tech et de la démocratie numérique, 2019.