

Rapport Technique - DA50

FreelyForm - Outil de création de formulaires web no-code

Projet réalisé par :

- Ayoub MOUSTAKIM
- Lilian BOUVIER
- Malsor HAMITI
- Théo COCAGNE
- Victor RAHUEL
- Yacine BEN-AHMED

Projet encadré par :

- Olivier RICHARD

Introduction

Dans le cadre de l'U.V DA50 Projet développement informatique avancé, nous avons eu comme consigne de reprendre un projet existant dont le sujet était une application permettant la création, la personnalisation et la visualisation de formulaires web, le tout, en no-code.

L'application s'adresse à tout type d'organisation comme par exemple une entreprise, un établissement d'enseignement, une association, un particulier... L'objectif principal à respecter était donc la simplicité et la facilité d'utilisation pour assurer que n'importe quel utilisateur sans connaissances particulières puisse réaliser des formulaires et les personnaliser.

Cet outil avait été commencé par le groupe du semestre précédent et nous nous sommes donc basés sur leur réalisation pour continuer de développer cette application en ajoutant les fonctionnalités manquantes pour arriver un produit fonctionnel.

Table des matières

Introduction.....	2
Table des matières.....	3
Sujet du projet.....	4
1. Objectif du projet.....	4
2. Problématique.....	4
Cahier des charges.....	5
I. Besoins Fonctionnels.....	5
1. Interface No-code pour la Création et l'Édition des Formulaires.....	5
2. Interface de Réponse au Formulaire.....	5
3. Interface de Visualisation des Réponses.....	5
4. Différentes règles.....	5
II. Besoins Techniques.....	6
1. Stockage des Formulaires et des Données.....	6
2. Affichage des Formulaires sur l'Interface.....	6
3. Exécution de l'Application sur Github Codespaces.....	6
Analyse du projet.....	7
Diagramme de Gantt.....	8
Organisation.....	11
Séparation des tâches.....	11
Conception - Diagramme de classes.....	13
Réalisation.....	14
Réalisation du Frontend.....	14
Réalisation du Backend.....	15
Base de données.....	15
Application Backend.....	16
Conclusion, Avis des membres.....	18

Sujet du projet

1. Objectif du projet

L'objectif du projet est de mettre en place une application web qui permet la création de formulaires avec les fonctionnalités suivantes :

- Une interface no-code (low-code pour les utilisateurs avancés) de création et d'édition de formulaires.
- Une interface de réponse au formulaire
- Une interface de visualisation des réponses d'un formulaire

Nous avons également des besoins techniques, définis au début du projet, à respecter qui sont les suivants :

- Stocker les formulaires et des données des formulaires dans une base de données MongoDB
- Gérer l'affichage des formulaires sur l'interface
- Permettre l'exécution de l'application sur Github Codespaces

2. Problématique

Un outil permettant la réalisation de formulaires avec des champs personnalisés pour créer des questionnaires personnalisés, des enquêtes de satisfaction ou des retours produits est un produit très demandé et touche plusieurs secteurs et plusieurs catégories de profils.

Le but est donc de faire cet outil en ayant une vision de l'ensemble de ces catégories afin qu'il soit adapté au plus grand nombre.

Certes, il y a beaucoup de solutions sur le marché, mais aucune ne répond suffisamment aux besoins précis d'un utilisateur tout en étant simple d'utilisation.

Ce projet répond favorablement à cette problématique en proposant une solution no-code qui facilite la prise en main et la personnalisation de formulaires sans connaissances particulières requises.

Cahier des charges

I. Besoins Fonctionnels

1. Interface No-code pour la Création et l'Édition des Formulaires

- Fournir une interface utilisateur intuitive pour la création et l'édition de formulaires sans nécessiter de compétences en codage.
- Permettre l'utilisation de divers champs de formulaire, y compris :
 - Texte
 - Nombre
 - Date
 - Case à cocher
 - Booléen (Vrai/Faux)
- Intégrer des règles de validation pour chaque champ, telles que :
 - Règle de maximum/minimum
 - Expression régulière (Regex) pour la validation de format
 - Validation d'adresse e-mail
 - Choix multiple ou simple

2. Interface de Réponse au Formulaire

- Créer une interface conviviale permettant aux utilisateurs de répondre aux formulaires.
- Assurer que les réponses soient enregistrées de manière fiable et accessible pour une analyse ultérieure.

3. Interface de Visualisation des Réponses

- Développer une interface pour visualiser les réponses collectées des formulaires.

4. Différentes règles

- Les données saisies doivent être vérifiées lors de la saisie du formulaire.
- La validation peut être customisable (vérifier qu'un champ contient un entier compris entre 10 et 20).
- Un champ peut être spécifié comme « facultatif » (un champ vide est valide).
- Un formulaire invalide ne peut pas être soumis.
- Les données saisies doivent être vérifiées après la soumission du formulaire avant d'être ajoutées aux enregistrements de données.
- L'affichage de l'interface de réponse à un formulaire doit être mobile-first.
- L'affichage et la saisie des formulaires et des données doivent être à la fois pratiques et agréables à utiliser.

II. Besoins Techniques

1. Stockage des Formulaires et des Données

- Utiliser MongoDB pour stocker les formulaires et les données des réponses de manière sécurisée et efficace.
- Assurer l'intégrité, la disponibilité et la confidentialité des données stockées.

2. Affichage des Formulaires sur l'Interface

- Implémenter des mécanismes pour un affichage dynamique et réactif des formulaires sur l'interface utilisateur.
- Garantir la compatibilité et la réactivité de l'affichage sur différents appareils et navigateurs.

3. Exécution de l'Application sur Github Codespaces

- Configurer et optimiser l'application pour une exécution fluide sur Github Codespaces.
- Assurer la compatibilité avec les outils et les environnements de développement intégrés dans Github Codespaces.

Analyse du projet

Antécédents

Lors de la première présentation de ce projet, nous avons été introduits à la version réalisée par le groupe qui avait choisi ce projet l'année précédente. Le groupe chargé de ce projet avait lui aussi été introduit à un existant et a commencé la réalisation du projet par une analyse de l'existant. Par contre, eux on fait le choix de ne pas repartir du code existant mais de tout re-développer.

Ils avaient différents diagrammes ainsi que des documents permettant de justifier certains choix technologiques donc nous les avons récupéré pour continuer le développement du projet

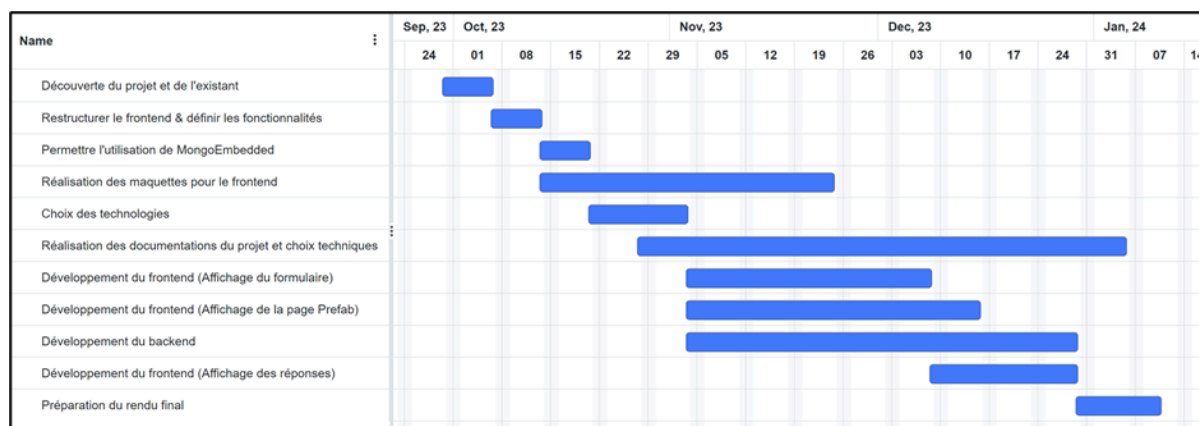
La priorité de ce groupe a donc été, selon leur rapport, de réaliser en priorité des documentations, des diagrammes, autant pour le code source que pour la description de la structure du projet. Nous avons donc pu récupérer cette documentation pour pouvoir partir sur les mêmes idées que celles établies par ce groupe.

En revanche, concernant le code source de l'application réalisé par le groupe précédent, il était très peu développé puisqu'en effet, le frontend de l'application ne contenait que des données statiques, sans connexions avec le backend.

Concernant le backend, il était fonctionnel mais il a été nécessaire de réaliser quelques modifications. Il manquait également certaines fonctionnalités clés pour arriver à une application fonctionnelle.

Notre premier objectif a donc été de déterminer les différentes fonctionnalités qui nous semblaient essentielles pour ce projet pour, par la suite, les planifier afin de s'assurer que notre application soit fonctionnelle et capable d'être en mesure d'être améliorée.

Diagramme de Gantt



Nous avons divisé notre diagramme GANTT en 11 parties principales différentes qui concernent les différents membres du groupe (d'où la superposition de différentes tâches). Comme certains sujets étaient plus importants d'un point de vue du besoin initial, nous avons à mi-parcours décidé de changer la formation des différentes équipes en créant deux équipes de deux membres qui ont travaillé sur des parties différentes du frontend. Cela nous a permis de finir à temps les différentes fonctionnalités que nous avions prévues.

Nous allons maintenant entrer plus en détail sur les différentes parties en expliquant quel a été le travail réalisé pour chacune d'entre-elles :

- **Découverte du projet et de l'existant** : Comme mentionné un peu plus haut dans ce rapport, ce projet a déjà été proposé à des groupes des années précédentes. Nous avons donc eu la tâche dans un premier temps de découvrir le travail réalisé par les autres groupes et ainsi définir la direction que nous voulions prendre pour le développement de ce projet. Nous avons également défini différents axes possibles de développement que nous ne pourrions pas réaliser mais qui pourront l'être en cas de reprise de notre projet par des groupes des années qui suivront.
- **Restructurer le frontend & définir les fonctionnalités** : Après avoir récupéré un projet qui ne se lançait pas lors de la réception de ce dernier, il a été défini avec notre responsable de projet, M. Oliver RICHARD, qu'il serait plus adapté de repartir de zéro pour le front-end. Nous avons donc initié un projet en React avec Typescript.
- **Exécuter l'application dans Github Codespaces avec l'utilisation de Mongo Embedded** : Le besoin exprimé par M. RICHARD était l'exécution du projet dans un environnement d'exécution défini et unique. Ainsi, notre premier objectif a été de permettre l'exécution de ce projet directement sur l'outil Codespaces de GitHub.

Ainsi, l'environnement serait fixé et permettrait une exécution garantie du projet.

Nous avons donc dû utiliser une base de données avec des données de test. Nous avons utilisé MongoEmbedded (qui nous à été conseillé par M. RICHARD) qui est un outil permettant l'exécution d'une base de données MongoDB sans utiliser docker. Cela permet de simplifier l'exécution de l'application dans GitHub Codespaces

- **Réalisation des maquettes pour le frontend** : Cette étape concerne la réalisation des maquettes afin que l'on s'accorde tous sur un design et une charte graphique commune. Nous avons dans un premier temps réalisé les maquettes pour l'affichage du formulaire, puis nous avons continué en réalisant les maquettes pour la création de PreFab, et enfin les maquettes qui définissent le design de l'affichage des réponses. Ces maquettes ont été réalisées sur Figma et sont disponibles pour chacun des membres du groupe. Nous avons également décidé à ce moment d'utiliser DaisyUI comme librairie de composants afin d'uniformiser la charte graphique de l'application web.

- **Choix des technologies** : En ce qui concerne cette étape, nous avons été assez rapide pour faire notre choix notamment dû au fait que nous avons récupéré une partie du backend qui nous a donc laissé un seul choix pour cette partie mais également parce que la technologie frontend qui a été choisie par le groupe précédent, bien que le frontend n'était pas implémenté, est celle qui est la plus maîtrisée au sein de notre groupe. Ainsi pour la partie frontend, nous avons décidé d'utiliser le framework React, et pour la partie backend nous avons donc décidé de réutiliser l'existant et donc de la réaliser en utilisant le framework Java Spring Boot

- **Réalisation des documentations du projet et des choix techniques** : Pour garantir l'amélioration et la continuité du projet, il est essentiel de tenir à jour des fiches de documentations sur le projet et sur les choix techniques qui ont été faits lors des phases de conception des différentes tâches. Certaines fiches de documentations avaient déjà été réalisées par les groupes précédents, mais certaines décisions ont été prises depuis la reprise de ce projet et nous avons donc dû actualiser le dossier contenant les documentations existantes.

- **Développement du frontend** : Pour cette phase, il s'agit tout simplement de la phase de réalisation du frontend du projet. Cette phase a été réalisée par une partie de l'équipe et l'objectif était d'implémenter les fonctionnalités définies auparavant lors des phases de conception.

- **Développement du backend** : Comme pour la phase précédente, il s'agit de la phase de réalisation du backend du projet. Nous nous sommes basés sur l'existant, contrairement à la phase précédente où l'existant n'existait pas pour le frontend. Ainsi, nous avons dû séparer le développement du backend en différentes parties. Dans un premier temps, nous avons dû modifier le code réalisé par l'équipe précédente pour qu'il puisse correspondre à

notre vision de notre projet. Dans un second temps, nous avons dû réaliser de nouvelles fonctionnalités, par la création de nouvelles routes d'API, ou par la complétion de fonctions incomplètes.

Organisation

Nous avons organisé des réunions à hauteur d'une réunion par semaine pour assurer une coordination efficace entre les équipes. Pour la gestion des tâches, la documentation des réunions, l'élaboration du produit minimum viable (MVP) et la rédaction des fiches techniques, nous avons utilisé la plateforme Notion comme outil collaboratif.

Notre projet a été divisé en trois équipes principales : la première était responsable de la création du formulaire, la deuxième s'est occupée de la gestion des réponses au formulaire, et la troisième s'est concentrée sur le développement backend qui communiquait avec les deux premières équipes.

Séparation des tâches

En tant qu'équipe, nous avons collaboré étroitement pour mener à bien notre projet, chacun apportant son expertise unique dans son domaine.

Théo Cocagne, en tant que chef de projet, a joué un rôle central dans la gestion et la répartition du travail. Il s'est chargé de la rédaction des documentations et des différents fichiers, y compris ce rapport technique et les vidéos de présentation. Durant la phase de conception, Théo a organisé l'équipe, validé les choix technologiques, et assigné les tâches pour une division efficace du travail.

Victor Rahuel, notre développeur backend, a pris en charge des aspects techniques cruciaux. Il a travaillé sur l'analyse technique, la refonte du code, l'intégration de Lombok, la mise en œuvre de règles de validation, l'ajout de données de test, la gestion des exceptions, et l'utilisation appropriée des codes HTTP. Son objectif était d'optimiser la robustesse de notre application.

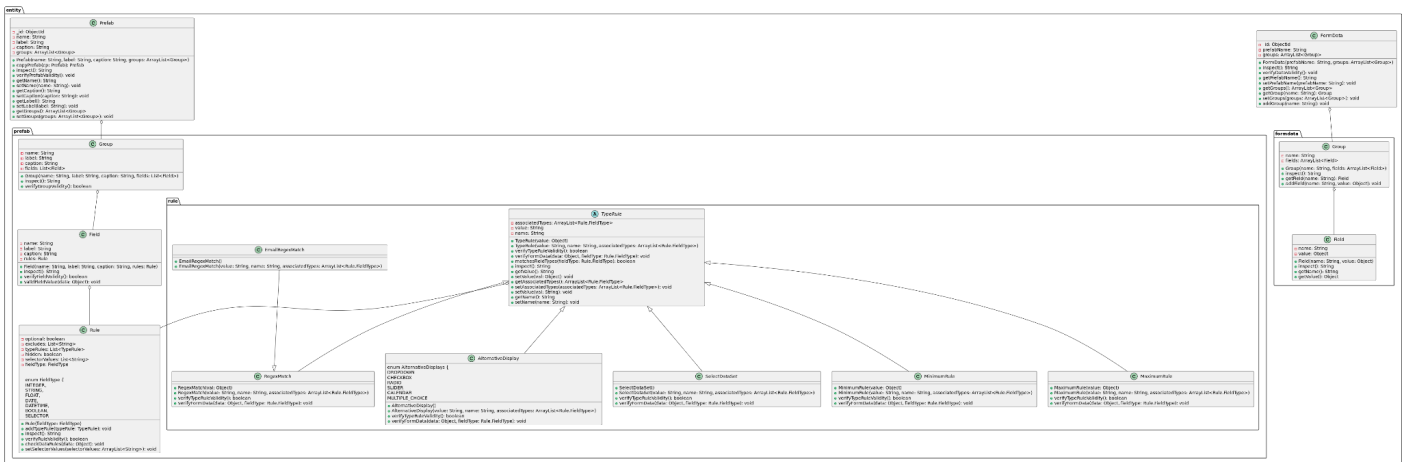
Ayoub Moustakim, en tant que développeur frontend, a conçu les pages de création de formulaire, en se concentrant sur une interface intuitive et esthétique. Il a également assuré une connexion efficace entre le frontend et le backend, grâce à l'utilisation d'un contrat OpenAPI, contribuant ainsi au succès de l'intégration.

Lilian Bouvier, également développeur frontend, a créé l'interface pour l'affichage des formulaires. Son travail a mis l'accent sur une présentation claire et attrayante, en veillant à une intégration harmonieuse des données avec le backend.

Yacine Ben Ahmed, un autre membre de notre équipe frontend, a assisté Lilian dans l'affichage des formulaires et a travaillé avec Théo sur le business plan, une étude de marché et une analyse des coûts. Il a également soutenu Victor sur certains aspects du backend.

Enfin, Malsor Hamiti a joué un rôle clé dans la conception des maquettes et la définition de la charte graphique de l'application. Il a assisté Ayoub dans la construction des formulaires, veillant à ce que notre application soit facile d'utilisation et accessible à un large éventail d'utilisateurs, conformément à notre vision d'une application no-code.

Conception - Diagramme de classes

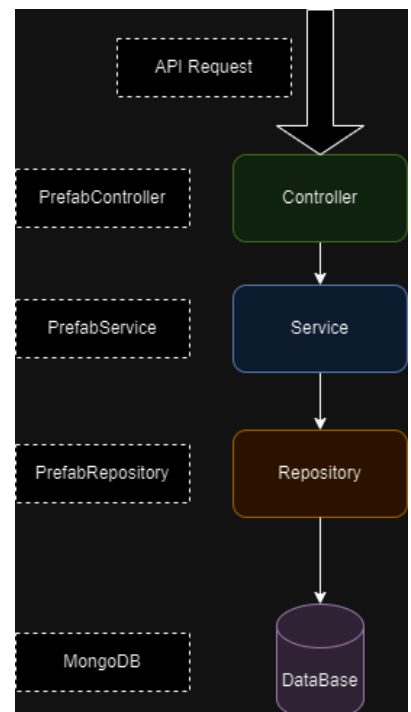


Avant de passer à la phase de réalisation de notre projet, nous avons dû prendre connaissance du backend qui avait été réalisé par le groupe précédent, et se baser sur cette conception afin de réaliser les différentes fonctionnalités imaginées.

Ce diagramme étant assez large, il est préférable pour vous, lecteur de ce rapport technique, de le visualiser directement dans le repository GitHub du projet dans la partie documentation.

En ce qui concerne ce schéma, il représente les différentes couches du backend en indiquant quels sont les noms des différentes classes associés à chaque couche.

On peut voir via cette représentation que le contrôleur passe toujours par une couche de service avant de pouvoir communiquer avec le repository qui fait la liaison avec la base de données. Cela nous permet ainsi de séparer les différentes fonctions des couches dans des fichiers distincts.



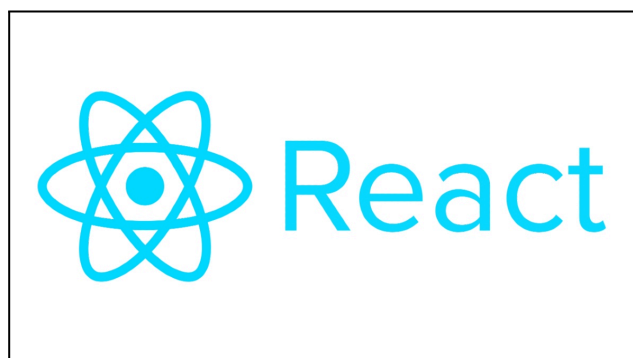
Réalisation

Réalisation du Frontend

L'application frontend permet l'interaction directe avec l'utilisateur final. Elle rassemble les fonctionnalités et permet aux utilisateurs d'interagir avec les données en offrant des interfaces graphiques qui rendent possibles la création, modification, suppression etc des données.

Dans la réalisation de notre application web de création et de gestion de formulaires, nous avons opté pour une approche moderne et efficace. Pour le frontend, notre choix s'est porté sur React avec TypeScript, ce qui nous a permis de créer une interface utilisateur réactive et fortement typée.

Nous avons également utilisé Daisy UI comme librairie de composants afin de simplifier l'intégration du frontend. Ce choix a été fait pour réduire le temps de développement et permettre de normaliser les différents composants de notre application.



Nous avons utilisé Jotai pour la gestion de l'état, offrant une simplicité et une performance accrues par rapport aux solutions traditionnelles. Cette bibliothèque légère a été essentielle pour maintenir notre application structurée et réactive, en facilitant la gestion des états globaux et locaux des composants.

Pour assurer une interaction fluide avec le backend Java Spring Boot, nous avons généré un contrat OpenAPI. Cela nous a permis de créer automatiquement des types TypeScript correspondants pour notre client API grâce à typed-openapi, garantissant ainsi un développement "typesafe" et une intégration facile avec notre API.

L'utilisation de React Router a été cruciale pour la navigation dans l'application. Nous avons défini des routes spécifiques pour les différentes fonctionnalités, telles que la création de formulaires, la réponse aux formulaires, et la visualisation des réponses, assurant ainsi une navigation intuitive et une organisation claire du contenu.

Réalisation du Backend

Base de données

Le stockage des fichiers de configuration et des fichiers de données entrées par les utilisateurs ne peut s'effectuer de façon fluide dans une base de données relationnelle classique.

En effet, celles-ci s'attendent à recevoir des données de types précis et dans une structure rigide. Les fichiers de configuration et de données de formulaire peuvent cependant avoir des structures très variables, suivant la configuration du formulaire et la présence ou non de données optionnelles.

Les bases de données non relationnelles, plus flexibles dans leur structure et capables de gérer rapidement d'importants volumes de données non structurées sont plus intéressantes pour notre cas d'usage.

L'équipe précédente avait opté pour MongoDB comme système de gestion de base de données pour le projet et nous avons également décidé que c'était un choix cohérent. C'est un système relativement mature parmi les systèmes NoSQL, populaire et solidement supporté par les frameworks Java.

Pour ce projet, nous avons uniquement implémenté Mongo Embedded pour pouvoir l'intégrer et tester dans GitHub Codespaces



Application Backend

Pour des raisons de cohérence avec nos autres réalisations du semestre et avec le groupe précédent, le framework Java Spring Boot nous était imposé pour l'application backend.

Ce n'est pas pour autant que ce framework est un mauvais choix. Spring Boot est, tout comme MongoDB, un système éprouvé et populaire.

Son utilisation est rendue plus facile par la richesse de la documentation disponible, la multitude de tutoriels expliquant ses fonctionnalités et l'abondance de détails disponibles sur toute erreur inattendue que l'on peut rencontrer lors de son utilisation, de part une connaissance incomplète du framework.

Spring Boot est un framework très puissant qui offre un environnement de développement solide et efficace.

Spring Boot est utilisé pour ce projet en conjonction avec Apache Maven, un outil de gestion et d'automatisation de production des projets Java, qui permet de gérer plus aisément les dépendances externes utilisées dans le projet et d'établir des tâches spécifiques à effectuer lors de la production d'un livrable.



- L'application backend est un service REST qui communique directement avec la base de données et qui présente plusieurs routes API qui peuvent être appelées pour obtenir des informations, des données en base, ou modifier l'état des données en base.
 - L'application backend permet de renvoyer les données au format JSON, dans un format facilement exploitable et manipulable par l'application frontend.
 - C'est l'application backend qui définit le format des données échangées.

Sur le plan du backend, nous avons réalisé une série d'améliorations techniques et fonctionnelles significatives. L'analyse technique et fonctionnelle a guidé notre processus de refonte partielle et d'amélioration du code existant. Cette approche a permis de structurer le projet de manière plus efficace et durable.

L'intégration de Lombok a joué un rôle clé dans la simplification du code. En réduisant la quantité de code boilerplate, nous avons augmenté la lisibilité et la maintenabilité de notre codebase.

Nous avons également mis en place des règles de validation pour assurer la fiabilité et la précision des données traitées. L'ajout de données de test dès l'initialisation de l'application a facilité les phases de développement et de test, en permettant une évaluation rapide des nouvelles fonctionnalités et des modifications.

La gestion des exceptions et l'utilisation de codes HTTP adaptés ont été essentielles pour garantir une communication efficace et fiable entre le backend et le frontend, ainsi qu'une expérience utilisateur cohérente et sans erreur.

Conclusion, Avis des membres

Théo Cocagne :

Au cours de ce projet j'ai eu l'occasion de découvrir de nouvelles choses. N'étant pas familier autant que mes collègues sur les technologies utilisées, je me suis proposé pour être chef de projet et cela m'a plu. Notre groupe était assez bien structuré et les membres qui m'accompagnaient étaient tous compétents ce qui était plus simple à gérer en tant que chef de projet.

Les fonctionnalités clés que nous avons définies lors des phases de conception ont été développées lors de ce semestre et fonctionnent correctement ce qui prouve donc que ce projet est une réussite. Il reste néanmoins quelques améliorations possibles à apporter à notre application mais, dans son état actuel, elle pourrait correspondre à un outil que l'on pourrait proposer à de nombreux profils d'utilisateurs.

Si je devais changer ma manière de faire dans une situation similaire avec une fonction similaire, je pense que la seule modification que j'aurais effectuée par rapport à ce qui a été réalisé ce semestre est la mise en place plus rapide des groupes et de la répartition des tâches. Cela aurait pu être réalisé plus tôt et nous aurait permis de commencer le développement plus tôt pour nous laisser plus de temps pour tester et valider l'application finale.

Lilian Bouvier :

Au cours de ce projet j'ai majoritairement travaillé sur le frontend, et donc sur React, qui est l'un de mes langages de prédilection. Cette expérience de travail collaboratif a été intéressante. En effet, j'ai eu l'occasion de partager différentes manières de faire et également ma vision de la partie frontend avec les autres développeurs de l'équipe. J'ai également eu l'occasion de travailler en collaboration avec Victor pour déterminer des fonctionnalités et une communication front-back pertinente.

Ce projet m'a beaucoup plu et je pense que la version actuelle permettra aux groupes suivants de continuer sur notre lancée et produire une véritable application de qualité.

Victor Rahuel :

Pour ce projet, j'ai eu l'occasion de travailler sur le backend. Les technologies utilisées que ce soit MongoDB ou bien Spring Boot sont des technologies que je connaissais et c'est donc naturellement que je me suis proposé pour réaliser l'ensemble du backend de l'application en reprenant l'existant laissé par l'équipe du semestre précédent.

En ce qui concerne ma vision sur la réalisation de ce projet, elle est assez partagée. La base existante du projet n'était pas réellement conforme ni adaptée à ce que j'aurais imaginé pour un projet similaire. En revanche, le fait de se baser sur des documentations et des explications pour comprendre les décisions qui ont été prises était un vrai plus.

J'ai essayé de réaliser un backend de telle sorte à ce qu'il soit facile à comprendre et rapidement modifiable et adaptable par les prochaines équipes qui seront amenées à travailler sur ce projet.

Ayoub Moustakim :

En ce qui me concerne, j'ai travaillé en collaboration avec différents membres de l'équipe tout au long de ce projet. J'ai eu l'occasion de réaliser différentes interfaces permettant l'implémentation des différentes fonctionnalités tout en veillant à ce qu'elles soient faciles à prendre en main et compréhensibles pour n'importe quel profil d'utilisateur.

Concernant les choix technologiques, ils correspondaient à mon profil, je n'ai donc pas été perdu lorsque des tâches sur le frontend m'ont été confiées.

Ce projet était très complet. Nous avons eu l'occasion de travailler sur différentes choses tout en gardant une ligne directrice bien claire.

Malsor Hamiti :

Au cours de ce projet, j'ai eu l'occasion de travailler sur différents sujets. J'ai notamment travaillé sur les maquettes qui ont servi aux développeurs frontend de réaliser correctement les interfaces des différentes fonctionnalités. J'ai également travaillé sur le frontend mais, React n'étant pas ma technologie de prédilection, j'ai uniquement assisté Ayoub sur les différentes tâches qu'il devait réaliser.

J'ai apprécié travailler sur ce projet et je trouve que le rendu final est conforme aux attentes qui nous ont été demandées.

Yacine Ben Ahmed :

Pour ce projet, j'ai travaillé sur le frontend avec Lilian notamment pour implémenter la page complexe permettant l'affichage des formulaires. Lilian étant plus expérimenté que moi, je l'ai laissé terminer cette partie pour me concentrer sur la réalisation du budget et du business plan de notre projet.

Ce changement de fonction au cours du projet ne m'a pas dérangé et cela m'a permis de me concentrer pleinement sur la réalisation des nouvelles tâches qui m'ont été confiées.

Je pense que le projet que nous avons réalisé répond parfaitement à la vision que l'on avait imaginé lors de la phase de conception, mais également à la vision qu'avait le groupe précédent sur ce projet. Nous espérons que le groupe qui reprendra ce projet sera en mesure de garder cette même vision pour l'améliorer sans remettre en cause le travail effectué jusqu'à présent.