**Royaume du Maroc**

**Université Mohammed Premier**

**Ecole Nationale des Sciences Appliquées Oujda**

**Filière Génie informatique**

Mémoire de projet de fin d’année

Soutenu le : 29/06/2024

Filière : **Génie Informatique**

Thème :

Réalisé par : Encadré par :

**Edrissi Mohamed Amine Mr. BOUCHENTOUF Toumi**

**Oustad Mohammed Mme. BELOUALI Saida**

**Mahtalah Mohammed Mr. BOULOUIZE Redouane**

Membres de Jury :

**Mr. BOUCHENTOUF Toumi**

**Mme. BELOUALI Saida**

**Mr. HAJA Zakaria**

Année Universitaire 2023-2024

**Remerciements**

Avant tout développement de cette expérience professionnelle, il parait important de commencer ce rapport par des remerciements à ceux et à celles qui nous ont beaucoup apprises, nous ont assistées et accompagnées tout au long de ce Projet

Nous tenons en particulier à remercier notre encadrant, **Mr. Bouchentouf Toumi**, **Mme.Belouali** **Saida et Mr. Boulouize Redouane** et pour son assistance et son accompagnement durant ce projet ainsi que pour les différentes connaissances qu'il nous a transmises. Nous tenons aussi les remercier pour leurs directives précieuses et leurs conseils pertinents qui nous ont été d'un appui considérable dans notre démarche.

**Résumé / Abstract**

L'objectif de notre projet de fin d'année au sein de la **Fac de Médecine spécifiquement au sein du Système CERBO *(Comité Ethique de Recherche Biomédical Oujda)*** est de concevoir une plateforme de gestion dont le but est la gestion des projets de recherche, qui permettrait à ce dernier d’automatiser la gestion des projets annuelles de recherche des investigateurs depuis la soumission jusqu'à la décision finale, tout en garantissant un accès sécurisé et une interface utilisateur conviviale.

Afin de parvenir à notre but nous avons opté pour **Spring / Réact** comme Framework de développement de l'application web, Java pour la partie backend, **Css Html Tailwind** pour la partie frontend, et UML pour la conception.

* + **Mots clés :** Gestion de projet, CERBO, automatisation, sécurité, Spring, React.

## **Table des matières**

**Table de matières 4**

**Table de figures 5**

**Liste d’abréviations 7**

**Introduction générale 8**

**Chapitre 1 : Contexte, Problématique, Objectifs et Méthodologie 9**

* 1. Introduction à CERBO 9
  2. Contexte et cadre du projet 11
  3. Problématique : 12
  4. Objectifs du projet : 13
  5. Méthodologie de gestion de projet  14

**Chapitre 2 : Conception, Modélisation et Planification du Projet 17**

2.1 Spécification et analyse des besoins : 17

### Planification du Projet 21

### 2.3 Conception du système 23

### Chapitre 3 : Réalisation, Implémentation et Tests du Projet 26

3.1 Outils et Technologies de développement : 26

3.2 Développement du projet : 32

3.3 Scénarios de tests et résultats obtenus 47

**Conclusion et Perspectives 49**

**Table des Figures**

*Figure1 : localisation de CERBO*

*Figure2 : Cycle de vie d’un projet*

*Figure 3 : use case ‘gestion des utilisateurs’*

*Figure 4 : use case ‘gestion des projets’*

*Figure 5: diagramme de classe*

*Figure  6:Architecture du backend*

*Figure 7 :la couche Modèle*

*Figure 8 :la couche Configuration*

*Figure 9 :la couche Service*

*Figure 10  :la couche DTO*

*Figure 11  :la couche Controller*

*Figure 12 :la couche Repository*

*Figure 13 : l’architecture du Frontend*

*Figure 14 :landing page*

*Figure 15  :page d’accueil (Génération du code d’inscription)*

*Figure 16 : page d’acceuil (Projets Examinés)*

*Figure 17 : listes des membres*

*Figure 18 : page des Investigateurs*

*Figure 19 : Button manage*

*Figure 20 : affichage du projet par l’admin*

*Figure 21:la page d’éxamination*

*Figure 22 : validation des commentaires (du fichier descriptif)*

*Figure 23 : l’étape finale de l’examination.*

*Figure 24 :la validation du projet*

*Figure 25 : affichage d’un projet validé*

*Figure 26 : la liste des projets*

*Figure 27 : l’affichage du projet*

*Figure 28 : rapport et début de la section des commentaires.*

*Figure 29 : section de commentaire pour le fichier descriptif.*

*Figure 30 : la liste des projets d’utilisateur.*

*Figure 31 : la première étape du soumission*

*Figure 32 : la deuxième étape du soumission*

*Figure 33: la troisième étape de soumission*

*Figure 34 : Voir le Profil*

*Figure 35 : Logout*

**Liste d’abréviations**

|  |  |
| --- | --- |
| **Abréviation** | **Signification** |
| **HTTP** | HyperText Transfer Protocol |
| **IOC** | Inversion of Control |
| **MVC** | Modèle Vue Contrôleur |
| **MySQL** | My Structured Query Language |
| **URL** | Uniform Resource Locator |
| **UML** | Unified Modeling Language |
| **API** | Application Programming Interface |

**Introduction générale**

Étant étudiants en deuxième année du cycle ingénieur en génie informatique, nous étions tenus d'effectuer, le long du second semestre de cette année, un Projet de fin d’année. Le but de ce projet est d'enrichir notre formation académique d'une expérience professionnelle aujourd'hui devenue nécessaire pour une bonne intégration de l'élève ingénieur dans le marché du travail à l'obtention de son diplôme. D'autre part, ce projet fut l'occasion pour nous de nous familiariser avec le travail en entreprise et d'appliquer nos connaissances théoriques pour résoudre une problématique réelle.

Notre but était de créer une application web qui facilite la tâche aux membres de la comité CERBO d’une part d'obtenir, consulter, et noter les projets postulés de leur investigateurs, d'autre part gérer la traçabilité de ces projets de façon permanente dès leur postulation.

Ce rapport est composé de 3 principaux chapitres :

* + **Premier chapitre :** Il présente l'organisme CERBO, et son domaine de fonctionnement. Ainsi le Contexte, Problématique, Objectifs et Méthodologie
  + **Deuxième chapitre :** Il est consacré à une étude de Conception, Modélisation et Planification du Projet
  + **Troisième chapitre :** Réalisation, Implémentation et Tests du Projet.
  + On clôture ce rapport par une conclusion, montrant la valeur ajoutée de notre application, et les perspectives que nous avons envisagées pour l'évoluer.

Chapitre 1 :

Contexte / Problématique/ Objectifs et Méthodologie

Dans ce premier chapitre nous présentons d’abord l'organisme CERBO, et son domaine de fonctionnement. Ainsi le contexte et cadre du projet, la problématique posée, les objectifs du projet et ainsi la méthodologie suivie à la gestion de ce projet.

**1.1 Introduction à CERBO**

Le **CERBO** est une commission d'examen établie pour aider à protéger les droits et le bien-être des sujets humains impliqués dans la recherche biomédicale. Ce comité a pour mission d’évaluer les projets de recherche impliquant des êtres humains déposés par les porteurs de projets affiliés ou non à la Faculté de Médecine et de Pharmacie d’Oujda.

**1.1.1 Composition de CERBO**

Le **CERBO** est une structure indépendante et pluridisciplinaire avec plusieurs caractéristiques qui reflètent les principes et objectifs de son travail. Le comité d’éthique est composé de 15 membres permanents :

• Un président

• Un vice-président

• 13 membres

Si la recherche implique des sujets vulnérables, le comité invite toute personne compétente à titre consultatif.

* + 1. **Fonctionnement de CERBO :**
  + **Réunions et Examen des Projets**
* **Objectif des Réunions** : Examiner les projets de recherche et évaluer leur acceptabilité éthique.
* **Fréquence** : Réunions mensuelles et supplémentaires si nécessaire.
* **Calendrier** : Annoncé à l'avance, temps suffisant pour l'étude des documents.
* **Quorum** : Présence du président ou remplaçant et quatre membres minimum. En cas de quorum non atteint, une deuxième réunion peut être tenue sans condition de quorum après huit jours.
  + **Groupes de Travail**
* **Formation** : Pour préparer les séances plénières par un travail préalable.
* **Présence du Président** : Obligatoire en dehors des réunions d’évaluation.
  + **Gestion des Conflits d'Intérêt**
* **Exclusion des Membres** : Membre avec conflit d'intérêt exclu de l'examen et du vote sur le projet concerné.
  + **Expertise Externe et Représentation**
* **Invitation d’Experts :** Sollicitation d'experts externes pour compétences spécifiques.
* **Représentants Religieux** : Invitation possible.
  + **Documentation et Confidentialité**
* **Procès-Verbaux** : Rédaction et archivage de chaque réunion.
* **Confidentialité** : Respect strict de la confidentialité des informations.
* **Archivage** : Documentation archivée pendant au moins trois ans après la fin de la recherche.
  + **Impartialité et Prise de Décision**
* **Neutralité** : Décisions prises uniquement en réunions annoncées et avec quorum.
* **Participation aux Délibérations** : Investigateurs et promoteurs entendus mais non impliqués dans la décision finale.
* **Archivage des Décisions** : Conservation et présentation des dossiers sur demande.
  + 1. **Situation géographique de CERBO**



***Figure1 : localisation de CERBO***

Le **CERBO** siège à la Faculté de Médecine et de Pharmacie d’Oujda (FMPO) mais il demeure

Indépendant de la hiérarchie et de l’organisation du personnel au sein de la FMPO. Le CERBO est En mesure de prendre des décisions sans subir de pressions institutionnelles, professionnelles, Politiques ou d’ordre économique.

**1.2 Contexte et cadre du projet**

**1.2.1 Contexte et choix du projet :**

Le Comité Ethique de Recherche Biomédical Oujda est un organisme essentiel à la Fac de Médecine, chargé d'assurer que les projets de recherche respectent les normes éthiques et scientifiques rigoureuses. Le choix de ce projet s'inscrit dans le cadre de la modernisation des outils de gestion du CERBO, visant à améliorer l'efficacité et la traçabilité des projets de recherche.

* + 1. **Pertinence du projet dans le cadre des études :**

Ce projet permet d'appliquer les compétences acquises en développement web et gestion de projet, tout en répondant à un besoin réel et concret.

**1.3 Problématique :**

**1.3.1 Analyse approfondie de la problématique ou du besoin à résoudre**

La gestion manuelle des projets de recherche au sein du Système **CERBO** pose plusieurs défis importants qui justifient la nécessité d'une solution automatisée. Ces défis incluent :

1. **Inefficacités Opérationnelles :**
   * Le processus manuel de soumission, de révision et d'approbation des projets de recherche est chronophage et exige un effort considérable de la part du personnel. Les tâches répétitives et administratives, telles que la saisie de données, la vérification de la conformité et la gestion des documents, ralentissent l'ensemble du processus.
2. **Risques d'Erreurs :**
   * Les erreurs humaines sont fréquentes dans les systèmes de gestion manuelle. Cela inclut des erreurs de saisie de données, des omissions de documents importants, et des erreurs de suivi des étapes du processus. Ces erreurs peuvent compromettre la qualité et l'intégrité des évaluations de projets.
3. **Délais Prolongés :**
   * La nature séquentielle et manuelle des étapes de gestion entraîne des délais importants. La communication entre les différents acteurs (investigateurs, évaluateurs, décideurs) peut être lente, et les retards dans la transmission d'informations cruciales sont courants. Ces délais prolongés peuvent avoir un impact négatif sur la réalisation des projets de recherche et sur l'innovation scientifique.
4. **Manque de Traçabilité et de Transparence :**
   * La gestion manuelle rend difficile le suivi précis et en temps réel de l'état des projets. Il est souvent complexe de maintenir une traçabilité claire des décisions prises, des modifications apportées et des communications échangées. Cela peut entraîner une confusion et une perte d'informations essentielles.
5. **Scalabilité Limitée :**
   * Avec l'augmentation du nombre de projets de recherche soumis chaque année, la capacité de gestion manuelle devient rapidement insuffisante. Il devient difficile de gérer efficacement un volume croissant de données et de projets sans une automatisation appropriée.

**1.3.2 Importance et pertinence de cette problématique :**

En réponse à ces défis, il est crucial de mettre en place une plateforme de gestion automatisée des projets de recherche. Cette plateforme doit non seulement rationaliser les processus opérationnels, réduire les erreurs et les délais, mais aussi garantir la sécurité des données et la transparence des opérations. Une telle solution permettrait au Système CERBO de gérer plus efficacement les projets de recherche, de faciliter la collaboration entre les parties prenantes et d'assurer une meilleure conformité aux normes éthiques et scientifiques.

* 1. **Objectifs du projet :**

Pour répondre aux problématiques identifiées et améliorer la gestion des projets de recherche au sein du CERBO les objectifs spécifiques du projet sont définis comme suit :

* 1. **Automatisation du Processus de Gestion :**

Concevoir une plateforme capable d'automatiser l'ensemble du processus de gestion des projets de recherche, depuis la soumission initiale jusqu'à la décision finale. Cela inclut l'automatisation de la soumission des projets, la révision par les membres de comité, la génération du rapport, le suivi des états, et la prise de décision.

* 1. **Réduction des Erreurs :**

Mettre en place des mécanismes pour minimiser les erreurs humaines courantes dans les systèmes manuels. Cela comprend la validation des formulaires et des contrôles de cohérence.

* 1. **Amélioration de l'Efficacité :**

Accélérer le processus de gestion des projets en réduisant et fixant les délais associés aux réunions, La plateforme doit permettre une communication rapide et fluide entre les membres de comité, l’administrateur et les investigateurs

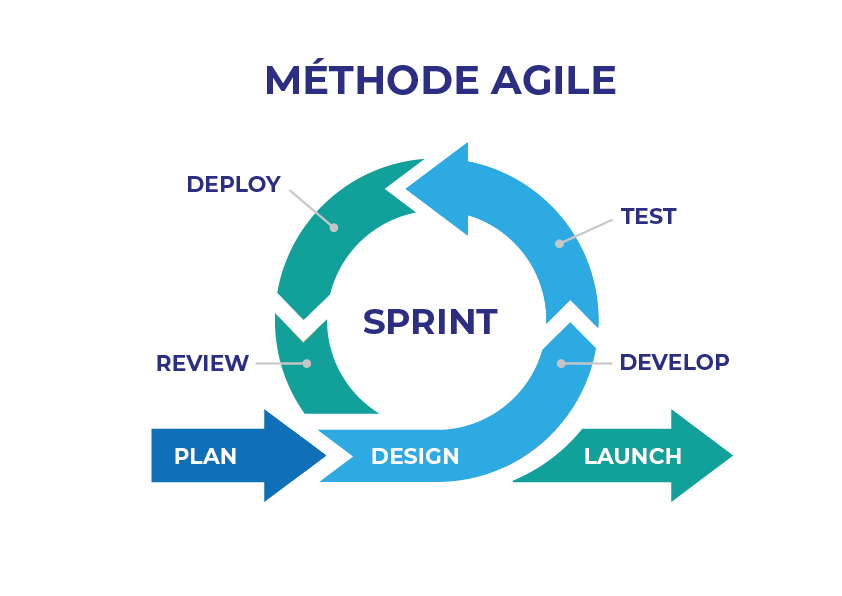
* 1. **Sécurité des Données :**

Assurer la sécurité et la confidentialité des données des projets de recherche. Cela inclut la mise en place d'un système d'authentification robuste et des permissions d'accès contrôlées.

* 1. **Interface Utilisateur Conviviale :**

Développer une interface utilisateur intuitive et facile à utiliser pour tous les acteurs impliqués (investigateurs, membres, administrateur). L'interface doit simplifier la navigation et l'exécution des tâches, tout en offrant une expérience utilisateur agréable.

* 1. **Méthodologie de gestion de projet :**

****

Pour la gestion de ce projet de conception d'une plateforme de gestion des projets de recherche au sein du Système CERBO, nous avons adopté la méthodologie **Agile**, en particulier le cadre **Scrum**.

Voici comment cette méthodologie a été mise en œuvre dans notre projet :

**1. Initialisation du projet :**

* **Définition de la vision du projet :**
  + Compréhension des objectifs globaux et des besoins des utilisateurs finaux : automatiser la gestion des projets de recherche, réduire les erreurs et les délais, assurer la sécurité des données.
  + Rencontre avec l’admin du CERBO pour clarifier les attentes et les résultats escomptés.
* **Planification initiale :**
  + Élaboration d'un plan de projet global avec des jalons clés et des objectifs à long terme.

**2. Planification des sprints :**

* **Product Backlog :**
  + Création et priorisation de la liste des fonctionnalités, des exigences et des améliorations souhaitées (ex. : soumission des projets, révision par les membres de comitté, gestion des états, sécurisation par authentification).
  + Le Product Owner (Admin du CERBO) est responsable de la gestion et de la priorisation du Product Backlog.
* **Sprint Planning :**
  + Organisation de réunions de planification de sprint pour sélectionner les éléments du Product Backlog à inclure dans le Sprint Backlog.
  + Définition des objectifs du sprint et des tâches nécessaires pour les atteindre.
  + Estimation du temps et des ressources nécessaires pour chaque tâche.

**3. Exécution des sprints :**

* **Daily Stand-up :**
  + Réunions quotidiennes pour synchroniser l'équipe, discuter des progrès, identifier les obstacles et planifier le travail de la journée.
* **Développement :**
  + Implémentation des fonctionnalités prévues pour le sprint en suivant les critères de définition de terminé (***Définition of Done*).**
  + Collaboration étroite entre les membres de l'équipe pour assurer la qualité et la cohérence du code.

**4. Revue et rétrospective des sprints :**

* **Sprint Review :**
  + Présentation des fonctionnalités développées à l’Admin du CERBO.
  + Recueil des feedbacks pour évaluer la satisfaction des parties prenantes et identifier les améliorations possibles.
* **Sprint Rétrospective :**
  + Identification des opportunités d'amélioration pour les prochains sprints.
  + Mise en place de plans d'action pour améliorer les processus et la collaboration.

**5. Répétition des sprints :**

* **Incrémentation :**
  + Ajout des nouvelles fonctionnalités et améliorations à chaque sprint, jusqu'à la finalisation du projet.
* **Adaptation :**
  + Ajustements et modifications basés sur les retours des parties prenantes et les leçons apprises.

**6. Clôture du projet :**

* **Finalisation et déploiement :**
  + Préparation de la version finale du produit pour le déploiement.
  + Exécution des tests finaux pour garantir la qualité et la conformité aux spécifications.
* **Documentation :**
  + Création de la documentation utilisateur et technique pour faciliter l'utilisation et la maintenance du produit.
* **Évaluation finale :**
  + Revue globale du projet pour identifier les réussites et les leçons apprises.
  + Préparation d'un rapport final détaillant les résultats obtenus, les défis surmontés et les recommandations pour les futurs projets.

Chapitre 2 :

Conception / Modélisation et Planification du Projet

Dans ce chapitre nous passerons à une étude d’analyse et de spécification des besoins et la conception de notre projet

**2.1** **Spécification et analyse des besoins :**

**2.1.1 Cycle de vie du projet :**



***Figure2 : Cycle de vie d’un projet***

Après avoir définir le cadre général de notre projet, nous passerons à une étude d'analyse et de spécification des besoins.

L'analyse et la spécification des besoins représentent la première étape du cycle de vie d'un logiciel informatique. Elle sert à identifier les acteurs réactifs du système ainsi que la source des données relatif à l'application, et ceci pour donner un résultat optimal et satisfaisant.

Ainsi, dans cette section, nous commençons par une spécification des besoins auxquels doit répondre l'application, passant ensuite à l'analyse de ces besoins à travers l'introduction des acteurs du système et la source des données.

**2.1.2 Les besoins fonctionnels :**

Les services proposés par notre application web se résument en **trois** acteurs majeurs :

***Administrateur / Membre de comité / Investigateur***

**1. Elle permet au (Membre de comité) de** *:*

* + **S’authentifier :**
* A travers un email et un mot de passe
  + **Voir / éditer son profile :**
* Voir ces informations personnelles (les infos soumis lors de l’inscription)
* Possibilité de modifier ses infos (structure de recherche, affiliation, password …)
* Enregistrer les modifications
  + **Gérer les projets déposés des investigateurs :**
* Vue de la liste des projets soumis et de leur état **(nouveau, en révision, validé).**
* Une recherche de projet selon leur id
* Possibilité de lire le projet (chaque fichier indépendamment dans une structure bien organisée)
* Examiner et évaluer les projets
* Attribuer des remarques concernant chaque fichier indépendamment du même projet.
  + **Gérer ses propres projets :**
* Interface conviviale pour soumettre ses propres projets de recherche à son rôle avec toutes les informations nécessaires (selon des étapes de soumission organisés)
* Voir la liste de ses projets soumis et de leur état (en attente, en révision, validé).
* Possibilité de modifier les détails des projets après la réception du rapport des remarques.*:*
  + 1. **Elle permet à (l’administrateur système) de :**

L’admin peut accéder aux mêmes fonctionnalités que celles d’un membre. De plus :

* + **S’authentifier :**
* A travers son email et son mot de passe
  + **Gérer les utilisateurs :**
* Création des users (donner la permission d’accéder à la plateforme a travers l’envoie d’un code de confidentialité à saisir lors de **Signup)**
* Modification et suppression des comptes utilisateurs.
* Attribution des rôles et des permissions
  + **Gestion des projets :**
* Vue de la liste des projets soumis et de leur état (nouveau, en révision, validé).
* Une recherche de projet selon leur id
* Possibilité de lire le projet (chaque fichier indépendamment dans une structure bien organisée)
* Examiner et évaluer les projets
  + **Reporting :**
* Examiner les remarques des membres à propos chaque projet
* Valider/ non valider les remarques
* Génération de rapports.
* Décision final de projet

**2.1.3 Les besoins non fonctionnels :**

Les besoins non fonctionnels sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système.

♦ Les principaux besoins non fonctionnels de notre application se résume dans les points suivants :

* **Performance :** L'application doit être rapide et réactive, même avec un grand nombre d'utilisateurs et de projets.
* **Sécurité :** Les données sensibles doivent être protégées contre les accès non autorisés.
* **Scalabilité :** Le système doit pouvoir évoluer pour gérer une augmentation du nombre d'utilisateurs et de projets.
* **Fiabilité :** Le système doit être stable et capable de fonctionner sans interruption.
* **Accessibilité :** L'application doit être accessible via différents appareils et navigateurs.

**2.1.4 Techniques de recueil et d'analyse des besoins**

#### Réunions avec les parties prenantes

Les réunions avec l’administrateur du CERBO ont été essentielles pour comprendre les processus existants et les défis rencontrés dans la gestion des projets de recherche. Ces réunions ont permis de :

* Identifier les acteurs clés du processus de gestion de projet, tels que les chercheurs, les responsables de projet et les membres du comité éthique.
* Discuter des flux de travail actuels, des points douloureux et des améliorations souhaitées.
* Recueillir des exigences spécifiques concernant la soumission des projets, la gestion des documents, et les critères de décision.

#### Analyse documentaire et études de cas

En complément des réunions et des enquêtes, une analyse documentaire a été menée pour :

* Examiner les documents existants relatifs aux processus de gestion de projet au sein du CERBO.
* Étudier les bonnes pratiques et les normes recommandées dans la gestion des projets de recherche biomédicale.
* Explorer des études de cas similaires dans d'autres institutions pour tirer des leçons et des comparaisons.

En intégrant ces techniques de recueil et d'analyse des besoins dans votre rapport, vous pouvez démontrer une approche méthodique et exhaustive pour comprendre les exigences du projet et formuler des solutions adaptées. Assurez-vous de discuter des résultats obtenus à partir de ces techniques et de la manière dont ils ont influencé la conception et la planification de votre projet.

### 2.2 Planification du Projet

Le planning prévisionnel du projet a été élaboré en utilisant un diagramme de Gantt détaillé, permettant de visualiser les différentes phases, les étapes clés, ainsi que les livrables attendus à chaque étape du développement de la plateforme de gestion des projets de recherche pour le CERBO.

#### Étapes du projet et livrables associés

1. **Phase de Conception :**
   * **Étapes :**
     + Analyse des besoins fonctionnels et non fonctionnels.
     + Conception de l'architecture du système.
     + Élaboration des modèles conceptuels (UML).
   * **Livrables :**
     + Document d'analyse des besoins.
     + Diagrammes UML (Diagramme de classes, use cases, etc.).
2. **Phase de Développement :**
   * **Étapes :**
     + Développement du backend avec Java et Spring Framework.
     + Développement du frontend avec React, HTML, CSS, et utilisation de Tailwind pour le design.
     + Intégration des modules et développement des fonctionnalités principales.
   * **Livrables :**
     + Code source documenté.
     + Interfaces utilisateur fonctionnelles pour les différentes fonctionnalités (soumission de projets, suivi des états, gestion des décisions).
3. **Phase de Tests :**
   * **Étapes :**
     + Réalisation de tests unitaires pour chaque module.
     + Tests d'intégration pour vérifier l'interopérabilité des différents composants.
     + Tests de performance pour évaluer la scalabilité du système.
   * **Livrables :**
     + Rapports de tests détaillés.
     + Correctifs et optimisations basés sur les résultats des tests.
4. **Phase de Déploiement et Mise en Production :**
   * **Étapes :**
     + Préparation de l'environnement de déploiement.
     + Déploiement initial sur un environnement de test.
     + Validation par les utilisateurs finaux et ajustements nécessaires.
   * **Livrables :**
     + Plateforme déployée et fonctionnelle.
     + Documentation d'utilisation et formation pour les utilisateurs finaux.

#### Utilisation du diagramme de Gantt

Le diagramme de Gantt a permis de planifier de manière détaillée chaque étape du projet, en attribuant des délais spécifiques à chaque tâche et en identifiant les dépendances entre les différentes activités. Cela a facilité la gestion des ressources et des priorités tout au long du développement, assurant une progression cohérente vers les objectifs fixés.

### 2.3 Conception du système

### 2.3.1 Diagramme de cas d’utilisation

### Le diagramme de cas d'utilisation représente les interactions entre les acteurs et le système à concevoir. Il décrit les différentes fonctionnalités (cas d'utilisation) offertes par le système et les acteurs impliqués dans chaque fonctionnalité. Ce type de diagramme est utile pour comprendre les exigences fonctionnelles du système du point de vue de l'utilisateur.

### a. Cas d’utilisation (gestion des utilisateurs)

### *F**igure 3 : use case ‘gestion des utilisateurs’*

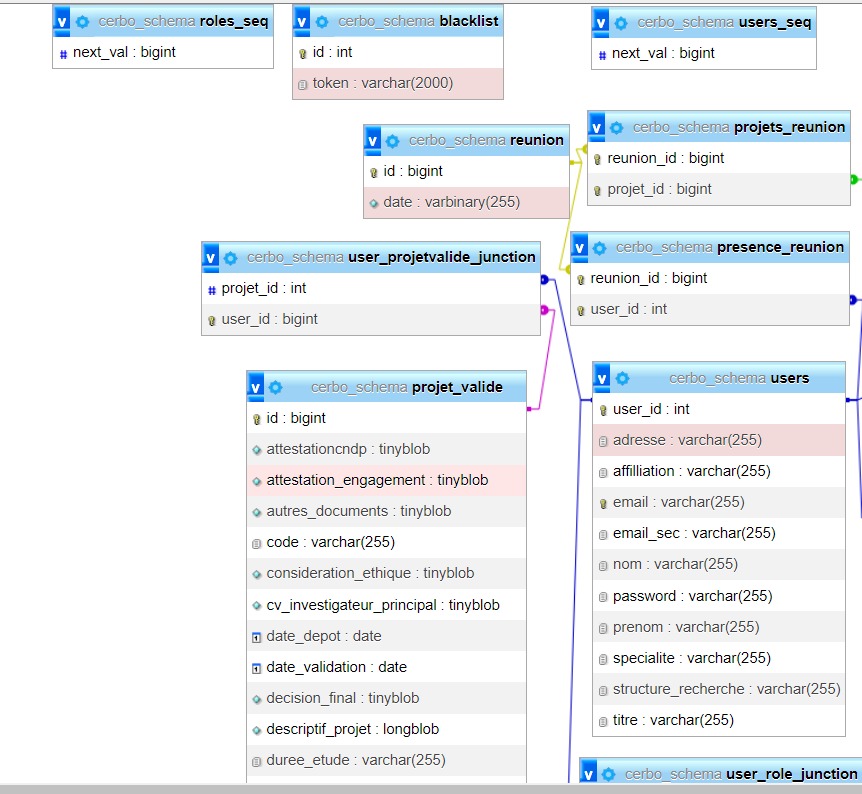
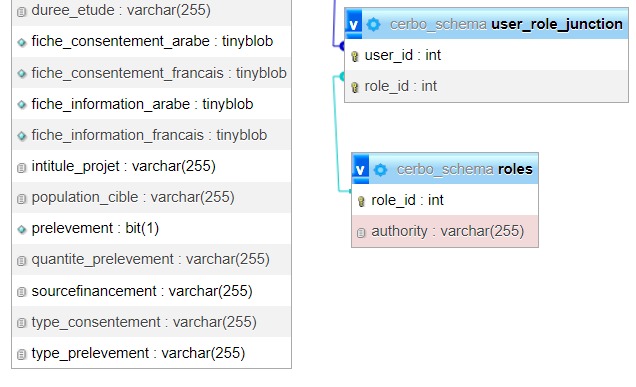
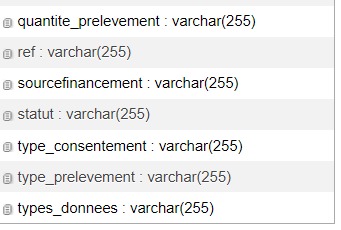
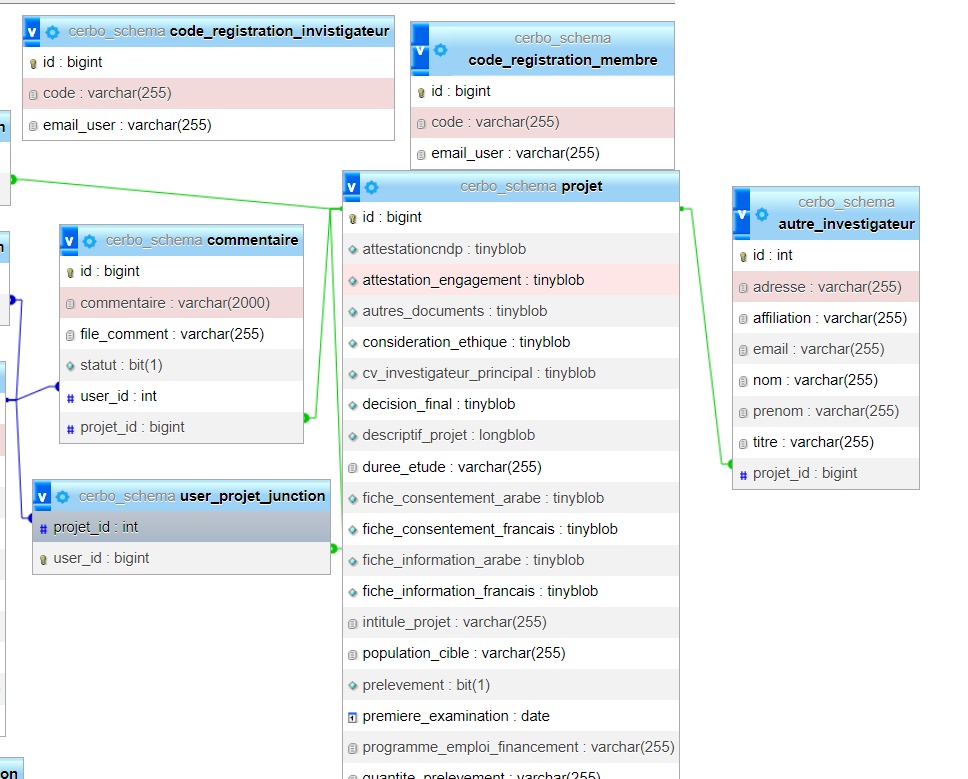
### b. Cas (Gestion des projets)

### 

### *Figure 4: use case ‘gestion des projets ’*

**2.3.2 Diagramme de classes**

Le diagramme de classes montre la structure interne du système. Il permet de fournir une présentation abstraite des objets du système, qui vont interagir ensemble pour réaliser les cas d'utilisation. La figure représente le diagramme de classes que nous avons élaboré après une étude détaillée des fonctionnalités du système.



*Figure5: diagramme de classe*

Chapitre 3 :

Réalisation / Implémentation et Tests du Projet

Dans ce troisième chapitre nous présentons, en premier lieu, l'environnement de travail et nous justifions les choix du langage de programmation et la technologie de développement appliqué ainsi le coté implémentation et Développement du projet.

**3.1 Outils et Technologies de développement :**

**3.1.1 Langage JAVA**

 Java 17 est une version LTS (Long-Term Support) du langage de programmation Java, publiée par Oracle en septembre 2021. Les versions LTS reçoivent des mises à jour de sécurité et des corrections de bugs pendant une période prolongée, ce qui les rend idéales pour les entreprises et les développeurs qui recherchent de la stabilité et un support à long terme.

Java est un langage de programmation orienté objet utilisable sur divers systèmes d'exploitation. Il est assez robuste, rapide, et fiable. Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.

**3.1.2 HTML**

HTML (HyperText Mark up Langage) est la Nouvelle révision majeure d'HTML, format de données conçu pour représenter les pages web Cette version est Devenue stable depuis 2012. Le langage comprend une couche application avec de nombreuses API

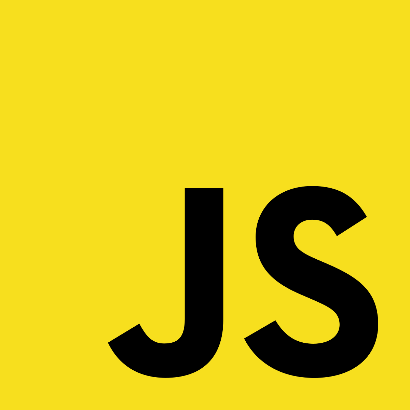
**3.1.3 CSS**

CSS est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade".



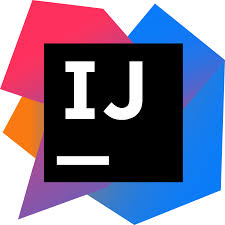
Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS.

**3.1.4 JAVA SCRIPT**

JavaScript est un langage de programmation interprété principalement utilisé pour créer et contrôler du contenu dynamique sur les sites web. Il permet d'ajouter des fonctionnalités interactives comme des animations, des formulaires dynamiques, des mises à jour de contenu sans rechargement de la page, et bien plus encore. JavaScript est exécuté côté client (dans le navigateur de l'utilisateur), mais peut également être utilisé côté serveur grâce à des environnements comme Node.js

**3.1.5 Environnements de développement :**

**a. IntelliJ IDEA**

IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains, principalement utilisé pour le développement de logiciels en Java, mais il prend également en charge de nombreux autres langages de programmation comme Kotlin, Groovy, Scala, et bien d'autres. IntelliJ IDEA propose des outils et des fonctionnalités avancées pour faciliter et améliorer le processus de développement logiciel, incluant la complétion de code intelligente, la refactorisation, le débogage, et bien plus encore. Parmi ses avantages : **Complétion de code intelligente / Refactorisation puissante / Débogage et test intégrés / Support multi-langages …**

**b. VS CODE**

VS Code est utilisé pour écrire et éditer du code dans divers langages de programmation développé par Microsoft. Il offre des fonctionnalités comme la complétion de code, le débogage intégré, le contrôle de version Git, et un vaste écosystème de plugins pour étendre ses capacités, facilitant ainsi le développement logiciel efficace et productif.

**c. Postman**

**Postman** est un outil de développement d'API qui permet aux développeurs de tester, documenter, et automatiser les API. Il offre une interface conviviale pour envoyer des requêtes HTTP, créer des tests automatisés, et organiser les requêtes en collections pour une gestion et une collaboration simplifiée.

**3.1.6 Outils et technologies de Base de données :**

**a. MySQL Workbench**



**MySQL Workbench** est un outil graphique intégré de gestion et de conception de bases de données pour MySQL. Développé par Oracle, il offre une interface utilisateur intuitive et riche en fonctionnalités pour les développeurs, les administrateurs de bases de données et les architectes de bases de données.

**3.1.7 Framework :**

**a. SPRING**

Spring est un Framework open source pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests. En 2004, Rod Johnson a écrit le livre Expert One-on-One J2EE Design and Développent qui explique les raisons de la création de Spring

* **Spring loC :**

Spring loc étant un Framework très complet et complexe, il ne sera décrit en détail qu'une des multiples manières de faire de l'injection de dépendance avec Spring.

Le fonctionnement de Spring loc repose sur 3 éléments clés :

• La configuration XML pour lier les implémentations à leurs interfaces

• Les classes sont des beans et possèdent donc des getters et setters pour les champs à injecter

• L'injection en même est effectuée par un conteneur de Spring

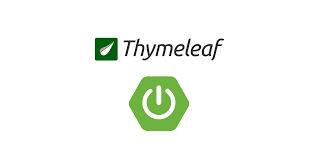
* **Spring Boot :**



**Spring Boot** est un Framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar, totalement autonome. Ce qui nous intéresse particulièrement, puisque nous essayons de développer des Micro services !

**c. Spring Security**

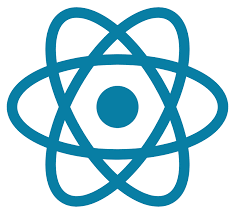
**Spring Security** est un Framework de sécurité complet et flexible pour les applications Java, permettant de gérer l'authentification et l'autorisation.**** Il offre des configurations simples via annotations et XML, protège contre les attaques courantes comme XSS et CSRF, et supporte divers mécanismes d'authentification, y compris OAuth2. Intégré de manière fluide avec d'autres modules Spring, il permet une personnalisation approfondie pour répondre aux besoins spécifiques des applications, assurant une sécurité robuste et prête pour la production.

*  **Spring Thymeleaf**

Thymeleaf est un moteur de template Java utilisé pour générer des pages web dynamiques. Il est conçu pour être utilisé avec des fichiers HTML valides, permettant ainsi de prévisualiser les templates directement dans un navigateur. Il s'intègre facilement avec le Framework Spring, en particulier Spring Boot.

Le rôle principal de Thymeleaf est de permettre la génération de contenu HTML dynamique dans les applications web Java. Il facilite la création de pages web en offrant une syntaxe intuitive pour lier les données du backend aux templates HTML, tout en permettant l'internationalisation, la réutilisation de fragments de template, et la gestion des formulaires HTML avec validation.

**b. React :**

React est une bibliothèque JavaScript open-source développée par Facebook pour créer des interfaces utilisateur interactives et dynamiques. Elle permet aux développeurs de construire des composants d'interface utilisateur réutilisables, facilitant la gestion de l'état et la mise à jour efficace du DOM (Document Object Model). React utilise une approche déclarative, ce qui signifie que les développeurs décrivent comment l'interface utilisateur doit apparaître en fonction de l'état de l'application, et React se charge de mettre à jour le DOM pour refléter ces changements de manière optimale. Grâce à son architecture basée sur les composants, React favorise une structure modulaire et maintenable pour le développement d'applications web modernes.

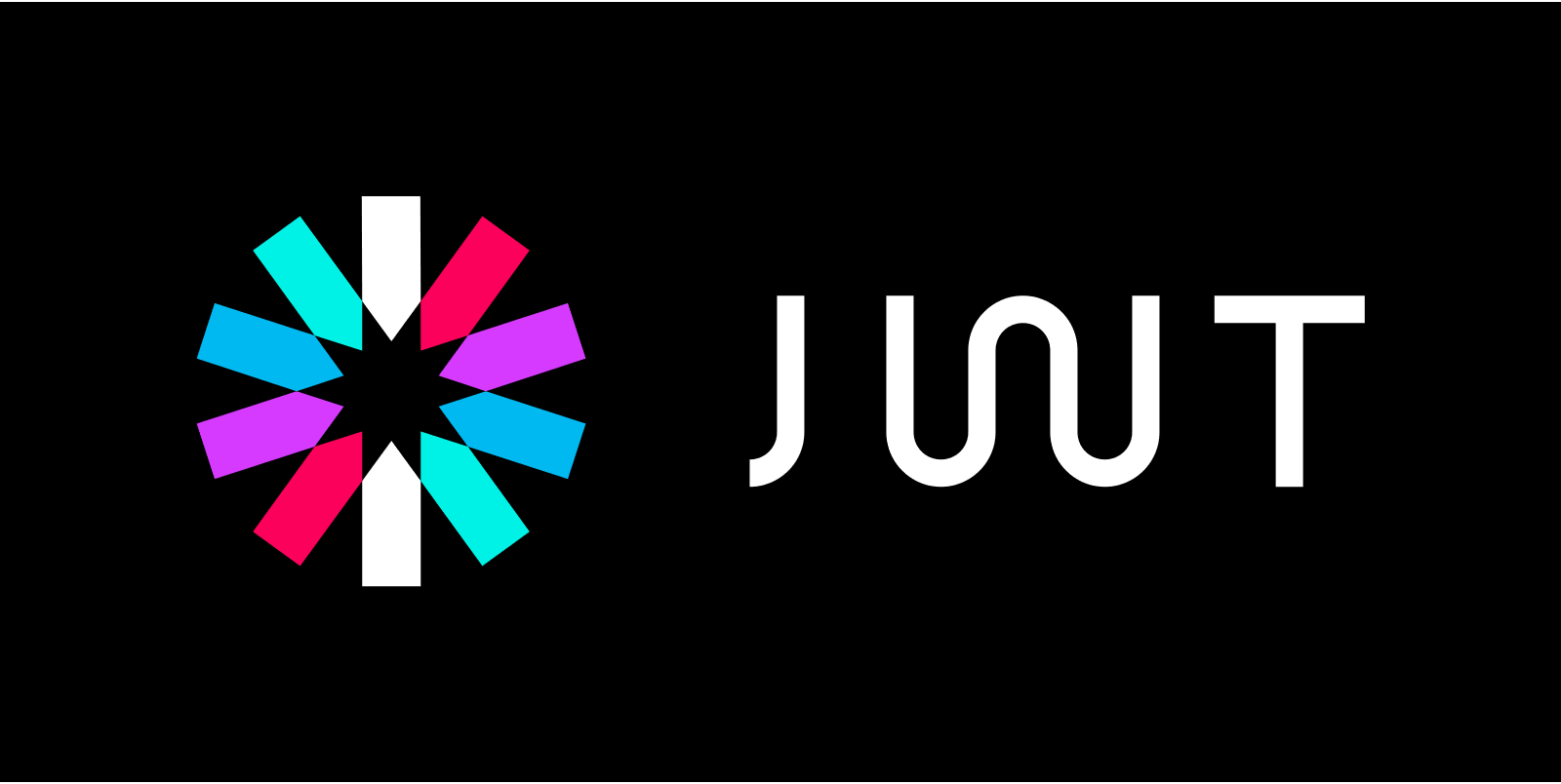
**c. Tailwind :**

**Tailwind CSS** est un Framework de conception utilitaire pour CSS qui permet de créer rapidement et efficacement des interfaces utilisateur personnalisées. Contrairement aux Frameworks CSS traditionnels qui offrent des composants prédéfinis, Tailwind se concentre sur des classes utilitaires de bas niveau, telles que p-4, text-center, et mt-2, permettant aux développeurs de combiner ces classes pour créer des designs uniques directement dans leur HTML.

**d. Hibernate**

Hibernate est un Framework de mapping Objet/Relationnel pour applications JAVA (et .NET avec Nhibernate). Supporter par Jboss/RedHat Hibernate, permet de créer une couche d'accès aux données (DAO) plus modulaire, plus maintenable, plus performante qu'une couche d'accès aux données HIBERNATE "classique" reposant sur l'API JDBC.

**3.1.8 Autre Outils :**

 **a. JSON WEB TOKEN**

JSON Web Token (JWT) est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité des données à l'aide d'une signature numérique. Elle s'effectue par l'algorithme HMAC ou RSA

**b. La bibliothèque MUI**

La bibliothèque MUI, anciennement connue sous le nom **de Material-UI**, est une bibliothèque populaire de composants d'interface utilisateur pour React. Elle implémente les principes du design Material Design de Google, offrant ainsi une collection de composants réutilisables et personnalisables pour créer des interfaces utilisateur modernes et cohérentes.

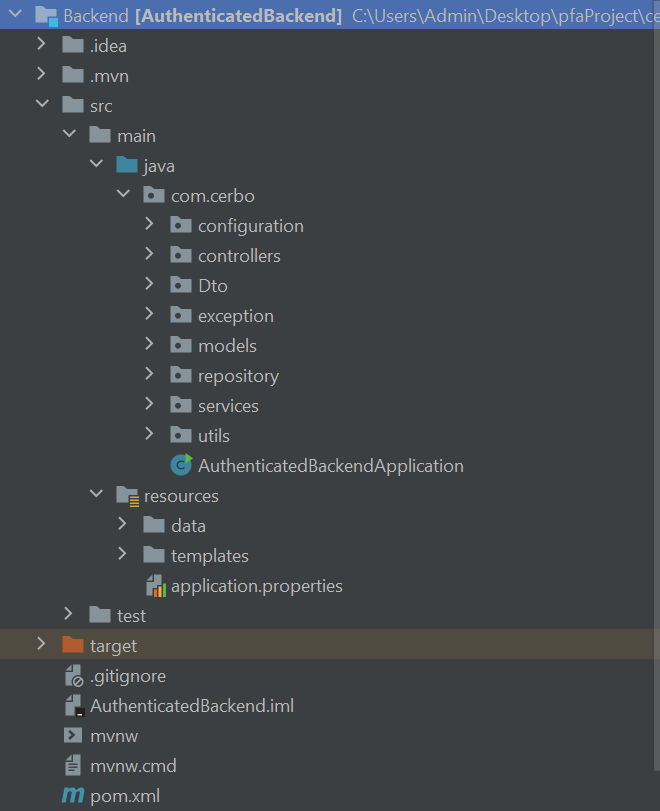
**c. La bibliothèque ‘react-pdf’**

**react-pdf** est une bibliothèque polyvalente et puissante pour les applications React, offrant des solutions pour afficher et créer des fichiers PDF. Que vous ayez besoin de visualiser des PDF ou de générer des documents PDF dynamiques, react-pdf fournit les outils nécessaires pour une intégration simple et efficace.

**3.2 Développement du projet :**

**3.2.1 Architecture du Backend :**

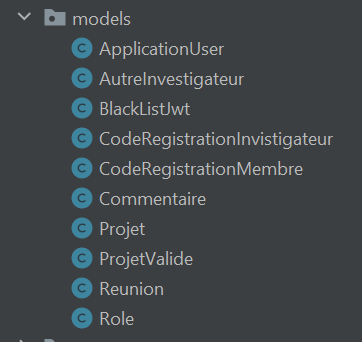
L’application est décomposée en couches applicatives, chaque couche a un rôle prédéfini dans l’architecture globale :



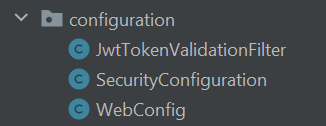
***Figure 6: Architecture du backend***

**a. La couche des modèles :**

Cette couche (figure 15) contient les entités, donc c'est une représentation des informations liées spécifiquement au domaine de l'application. La partie Modèle encapsule les données de l’application. Ces données sont en général définies et traitées par des simples classe java.



***Figure 7 : la couche Modèle***

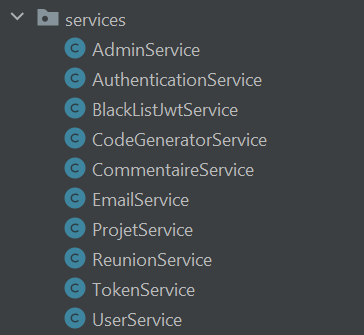
**b. La couche Configuration :**

***Figure 8 :la couche Configuration***

Cette couche (figure 16) a le rôle de gérer la sécurité cotée back end donc générer et vérifier les Json Web Token. Elle possède donc deux filtres, le premier s'exécute lors d'une authentification, donc il génère un JWT qui sera stocké dans le navigateur, et le deuxième c'est un filtre d'autorisation qui s'exécute lors de la demande d'une ressource, donc il valide la JWT qui se situe dans l'entête HTTP,et un filtre sert à bloquer les tokens qui sont dans le **BLACKLISTJWT** .

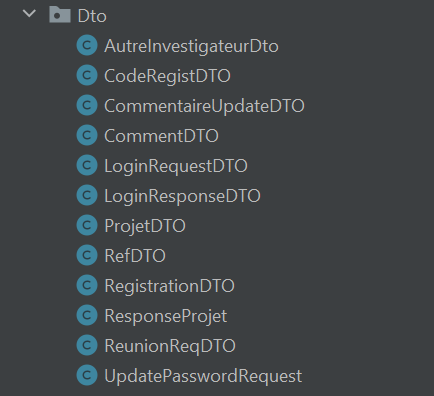
**c. La couche Services**

La couche de service est responsable de la logique métier de l’application. Elle interagit avec la couche de persistance repository et expose des méthodes que les contrôleurs peuvent appeler pour réaliser des opérations spécifiques.



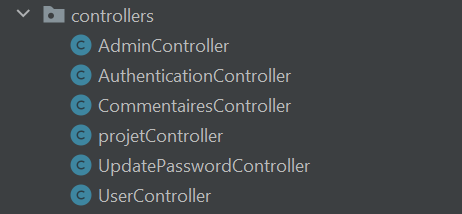
***Figure 9 :la couche Service***

**d. La couche DTO**

La couche DTO (Data Transfer Object) est utilisée pour transférer des données entre les différentes couches d'une application, souvent entre la couche de présentation (comme les contrôleurs) et la couche de service. Les DTO servent à encapsuler les données et à éviter que des entités de domaine soient directement exposées ou manipulées par la couche de présentation.

***Figure 10  :la couche DTO***

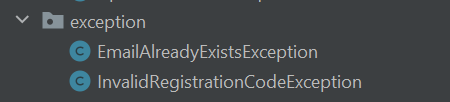
**e. La couche Controller**

 C'est la couche (figure ) supérieure d'une application Web. Il est responsable du traitement des entrées de l'utilisateur et du retour de la réponse correcte à l'utilisateur. La couche Controller doit également gérer les exceptions levées par les autres couches. La couche Controller étant le point d'entrée de notre application, elle doit prendre en charge l'authentification et agir comme première ligne de défense contre les utilisateurs non autorisés.

***Figure 11 :la couche Controller***

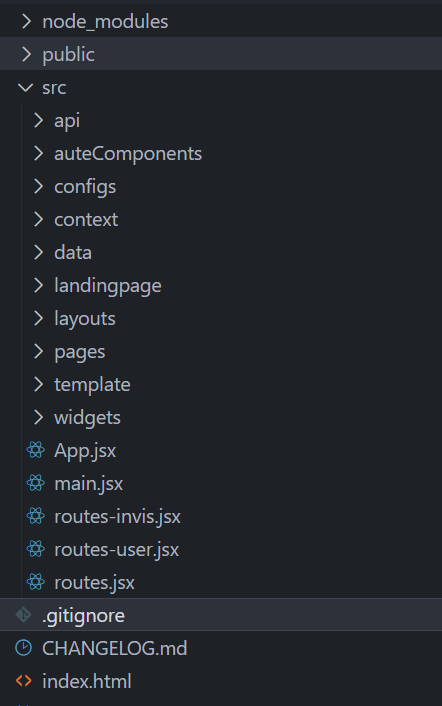
**e. La couche Repository**

La couche repository, également connue sous le nom de couche de persistance ou couche DAO (Data Access Object), est responsable de l'accès aux données et de leur gestion dans une application. Elle constitue une abstraction au-dessus du mécanisme de stockage des données, permettant aux autres parties de l'application d'interagir avec la base de données de manière plus simple et plus structurée.



***Figure 12 :la couche Repository***

**3.2.2 Architecture du Frontend :**



***Figure 13 : l’architecture du Frontend***

**3.2.3 Les interfaces de L’application :**

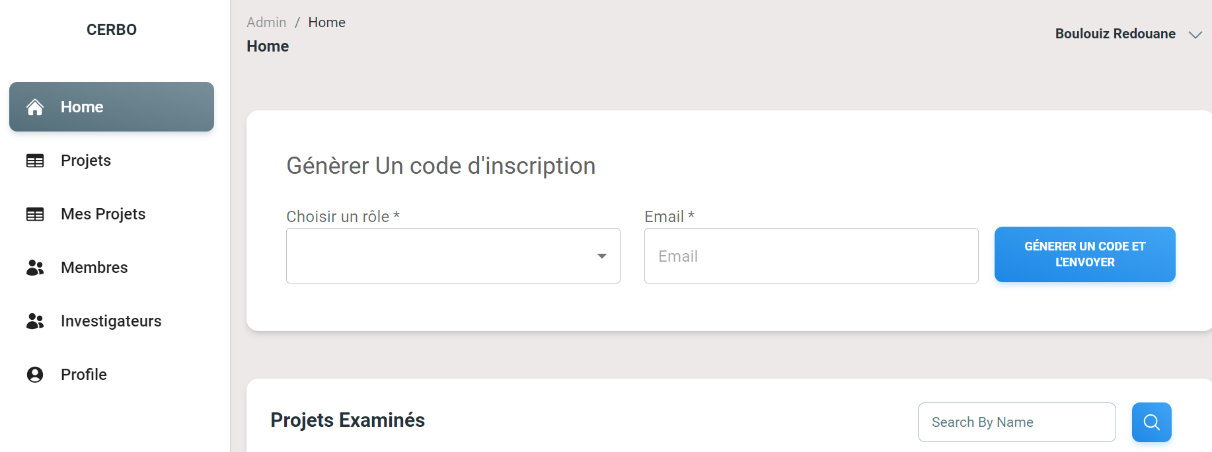


***Figure 14 :landing page***

La page affichée est une landing page pour le Comité d'éthique pour la Recherche Biomédicale d'Oujda (CERBO). Donne l’accès au page d’authentification ou d’inscription.

Pour accéder au Dashboard de l'application, l'utilisateur doit saisir son email et son mot de passe via l'interface d’authentification. Il existe trois types d’utilisateurs : **Admin / Membre /Investigateur.**

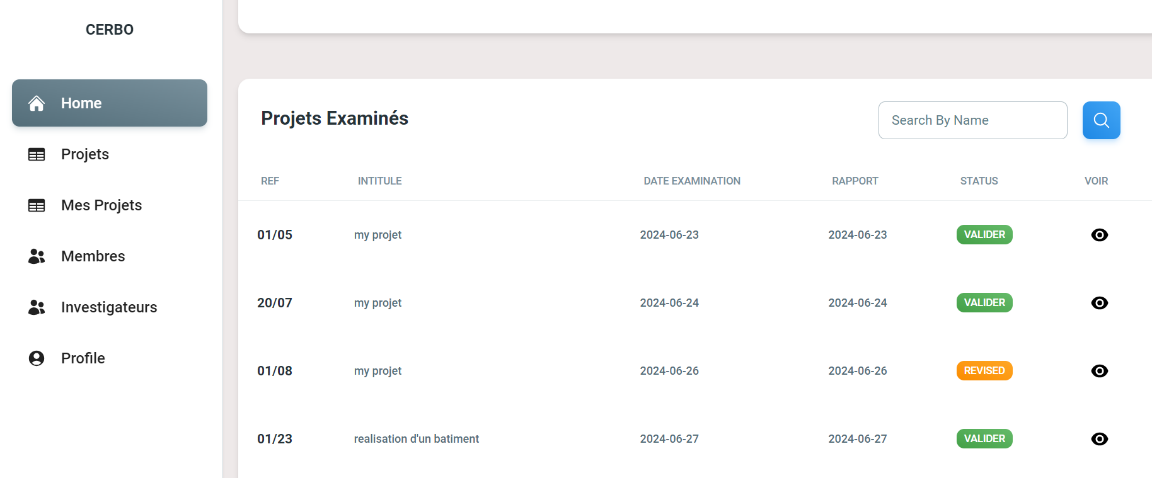
**3.2.3.1 Pour le cas d’un Admin**

 **a. la page d’accueil**

***Figure 15  :page d’accueil (Génération du code d’inscription)***

Après avoir être authentifié, la première interface qui s'affiche au début concerne la page d'accueil qui représente une partie permettant à **L’Admin d’envoyer un code de Registration** par email .Ce est généré automatiquement par la plateforme suivant le Rôle et Email entré par l’Admin.

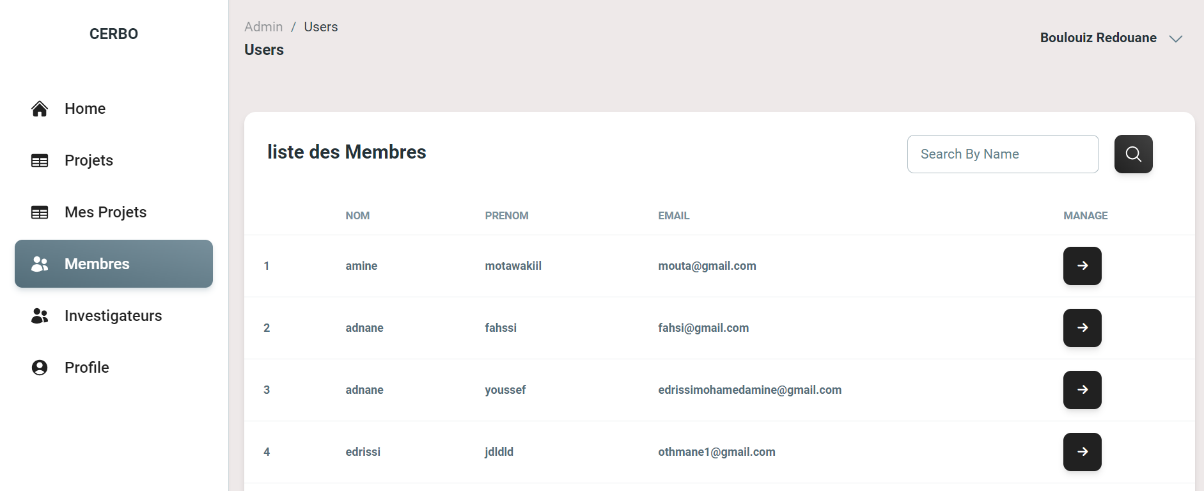
Ainsi un tableau de projets examiné présente les projets et leurs statuts (**révisé** : “déjà révisé”, **validé** : le comité a donné l’avis final du projet “ ou **nouveau**)



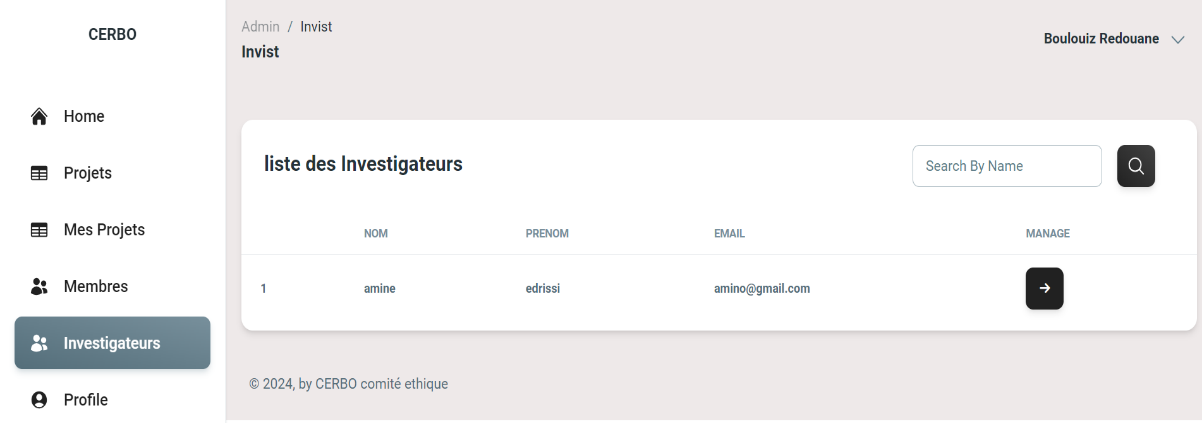
***Figure 16  : page d’acceuil (Projets Examinés)***

**b. gestion des membres et des investigateurs :**

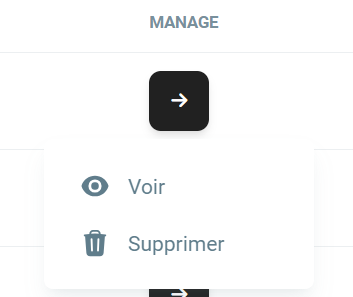
L’Admin a le droit de gérer le membres et les investigateurs, ajouter un utilisateur ( en générant un code de registration ) , supprimer les utilisateurs , consulter les informations des utilisateurs .

**Listes des membres :**

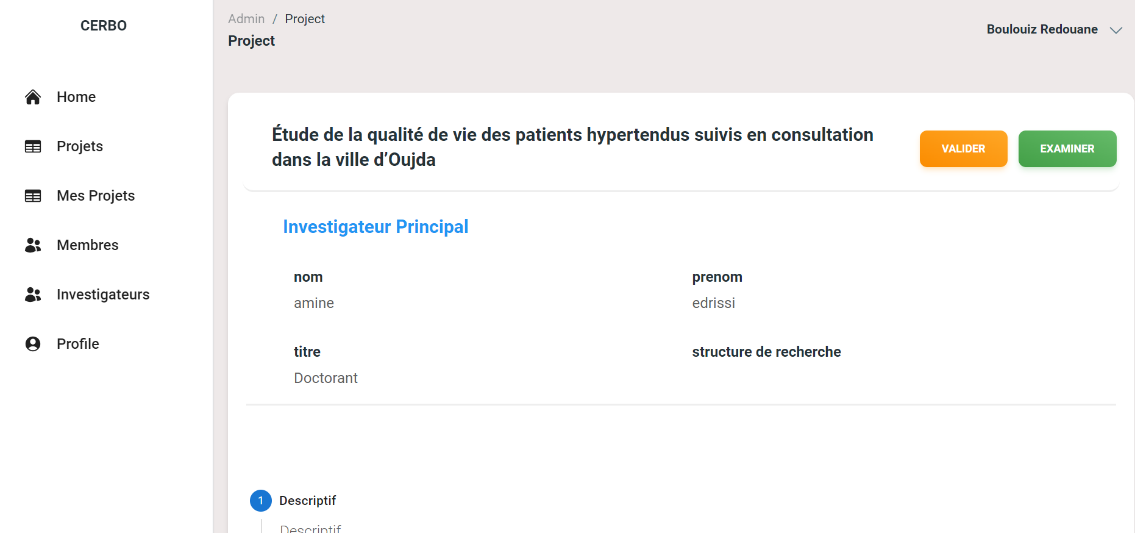
***Figure 17 : listes des membres***

 **Listes des investigateurs :**

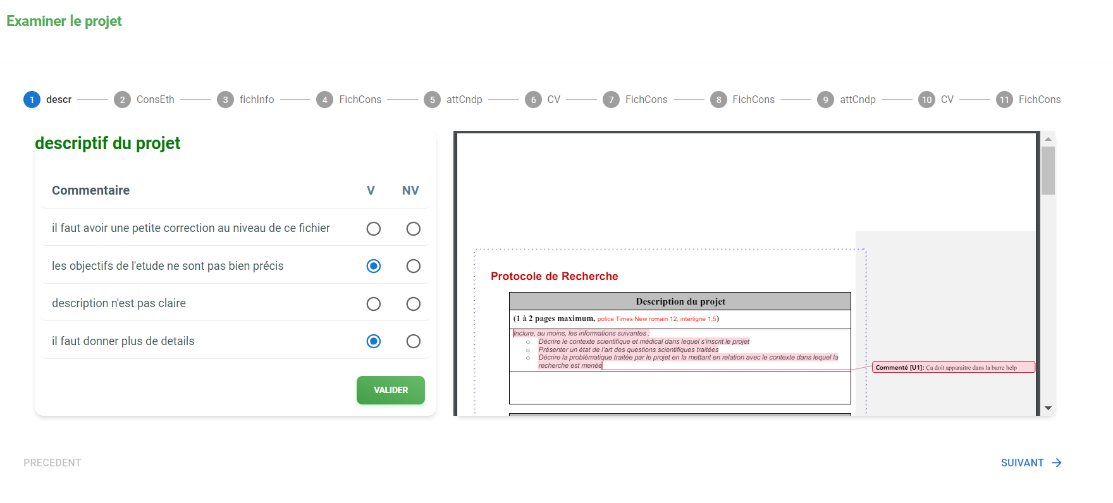
***Figure 18 : page des Investigateurs***



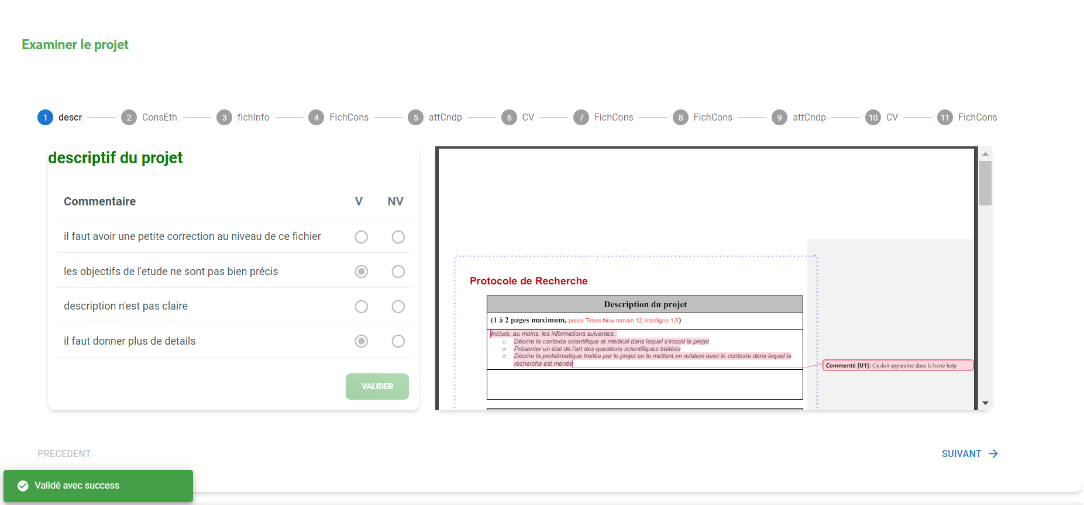
***Figure 19 : button manage***

**c. Examination du projet :**

***Figure 20 : affichage du projet par l’admin***

 En cliquant sur le bouton Examiner, une vérification aura lieu, permettant à l’administrateur de valider les commentaires saisis par les membres en tenant compte de leur avis lors de la réunion.

***Figure 21:la page d’éxamination***

Un **Button radio** et utilisé pour cocher V (si commentaire a été accepter par la comité ) et NV (si la comité ne prends pas en considération le commentaire ) . Par défaut le commentaire est NV.

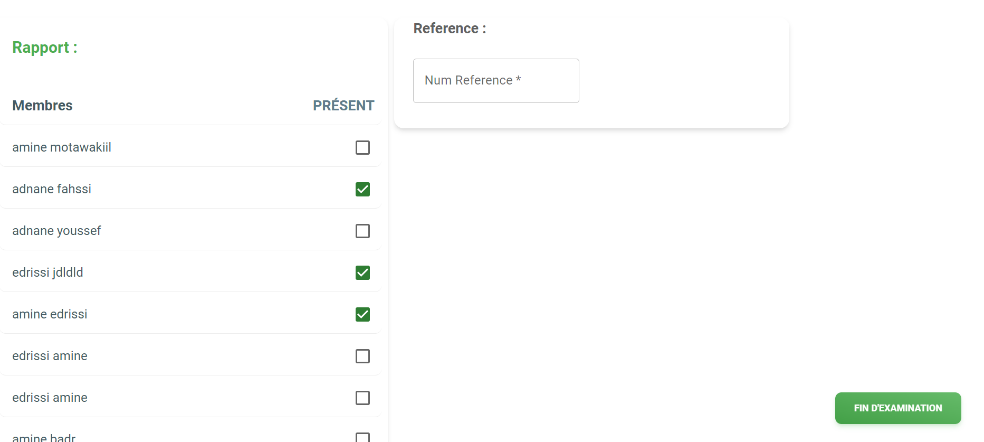
***Figure 22 : validation des commentaires (du fichier descriptif)***

**d.** [**Générer**](https://www.google.com/search?sca_esv=298f72215304aed5&sxsrf=ADLYWIIXnhz_m67ws1mE1TA8gTYAKYbjtg:1719531269303&q=G%C3%A9n%C3%A9rer&spell=1&sa=X&ved=2ahUKEwj0sfKT-fyGAxVn6gIHHeQ6Dp8QkeECKAB6BAgIEAE) **le Rapport  :**

Après que les étapes d’examen sont terminées, l’étape finale pour générer le rapport consiste à sélectionner la liste des membres présents à la réunion. Les membres du comité d’éthique se réunissent chaque mois, et l’application enregistre la liste de présence selon le mois et l’année de la réunion.

Un rapport sera généré contient les commentaires **validés**. Le statut du projet sera **“revisé”**

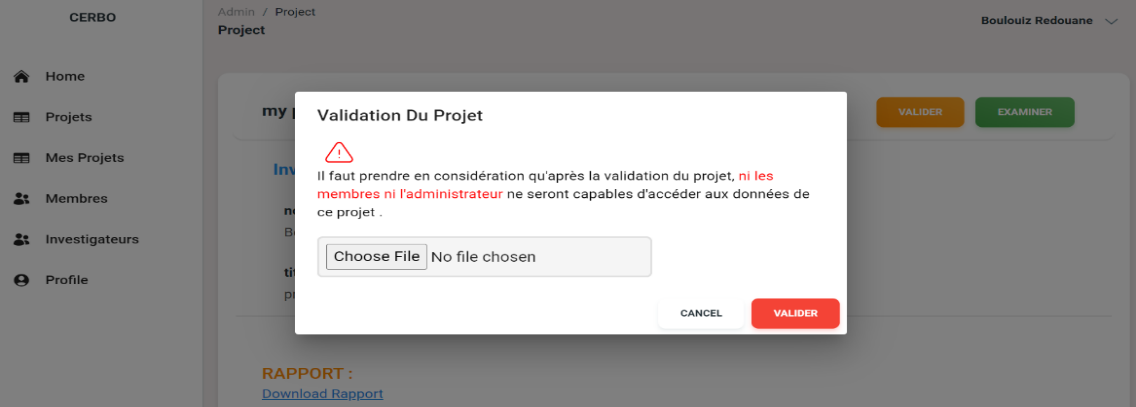
Pour donner possibilité a l’investigateur de modifier le projet



***Figure 23 : l’étape finale de l’examination.***

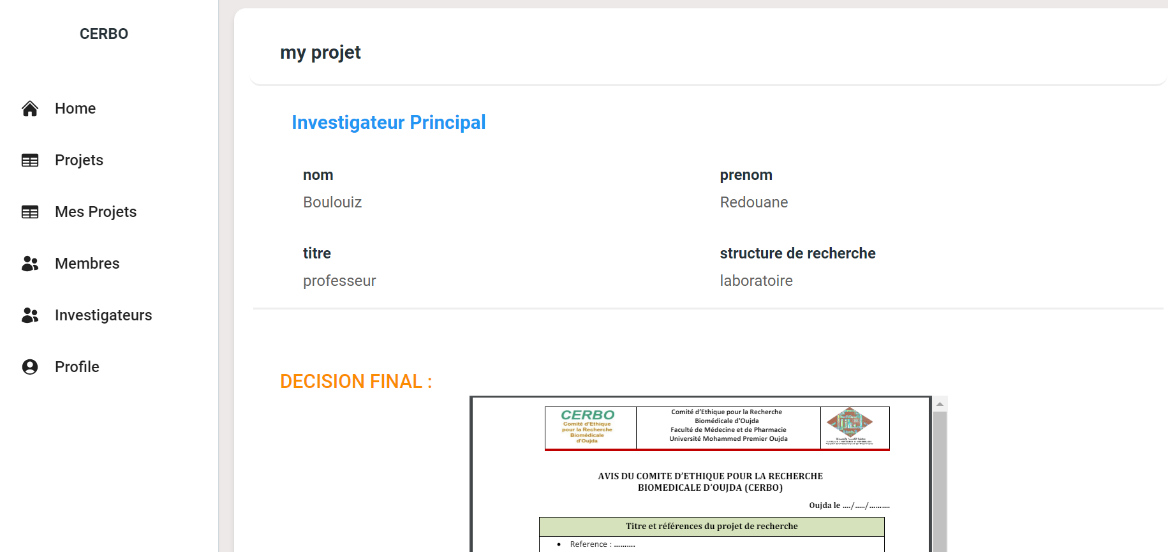
**e.** [**Générer**](https://www.google.com/search?sca_esv=298f72215304aed5&sxsrf=ADLYWIIXnhz_m67ws1mE1TA8gTYAKYbjtg:1719531269303&q=G%C3%A9n%C3%A9rer&spell=1&sa=X&ved=2ahUKEwj0sfKT-fyGAxVn6gIHHeQ6Dp8QkeECKAB6BAgIEAE) **l’Avis final :**

Pour valider le projet, l’administrateur doit entrer l’avis final du comité (fichier PDF). Après la validation, les fichiers du projet sont stockés dans une nouvelle table et plus personne, même l’administrateur, n’a accès aux données.



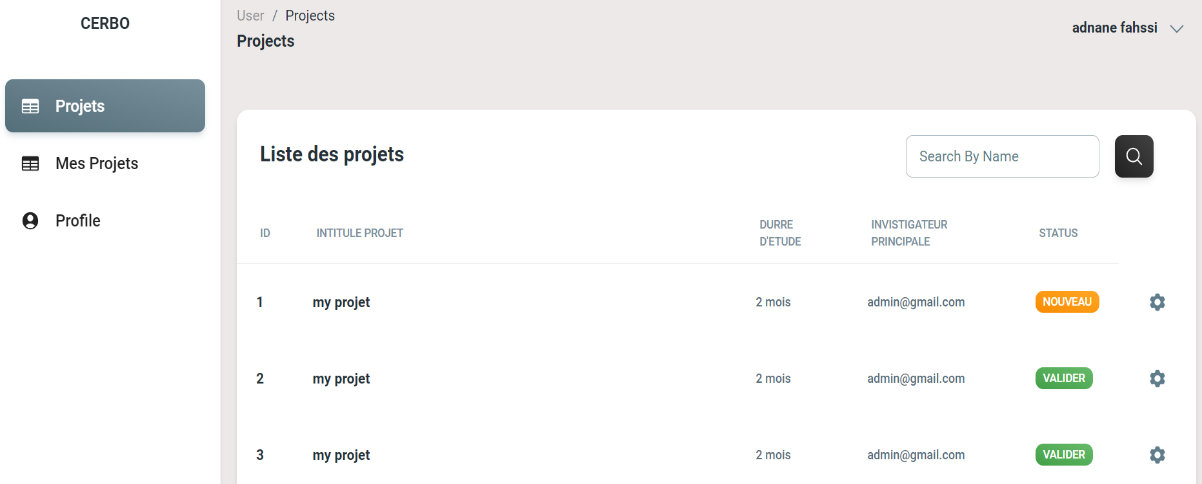
***Figure 24 :la validation du projet***

Après la validation le statut du projet devient “valideé”.



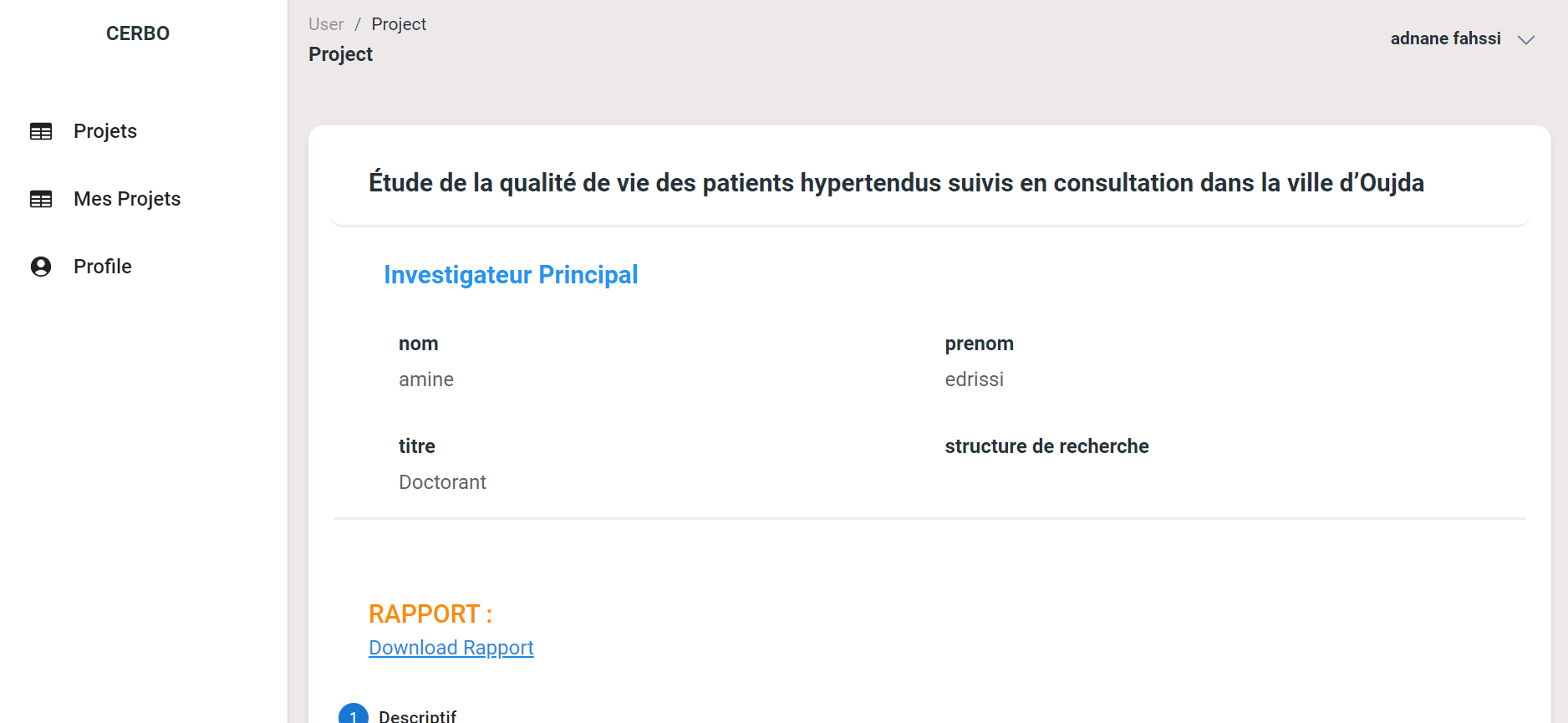
***Figure 25 : affichage d’un projet validé***

**3.2.3.2 Pour le cas d’un Membre (**Même l’admin ces les fonctionnalités)

**a. Liste des Projets :**

***Figure 26 : la liste des projets***

**b. Afficher le projet:**

**

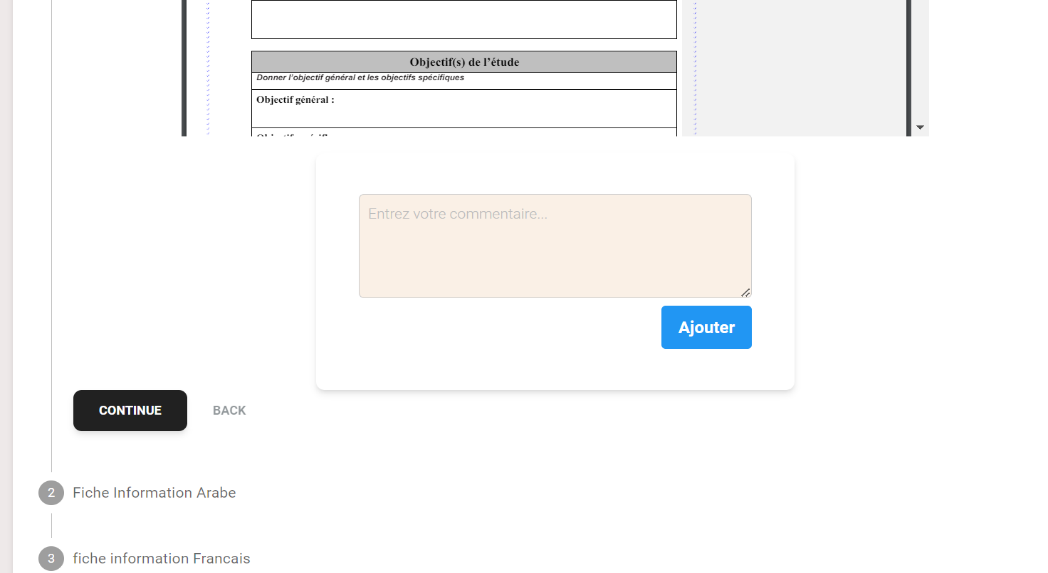
***Figure 27 : l’affichage du projet***

**c. commenter le projet :**

Le membre peut consulter chaque fichier de projets indépendamment et affecter de remarques et des commentaires ainsi télécharger le rapport :



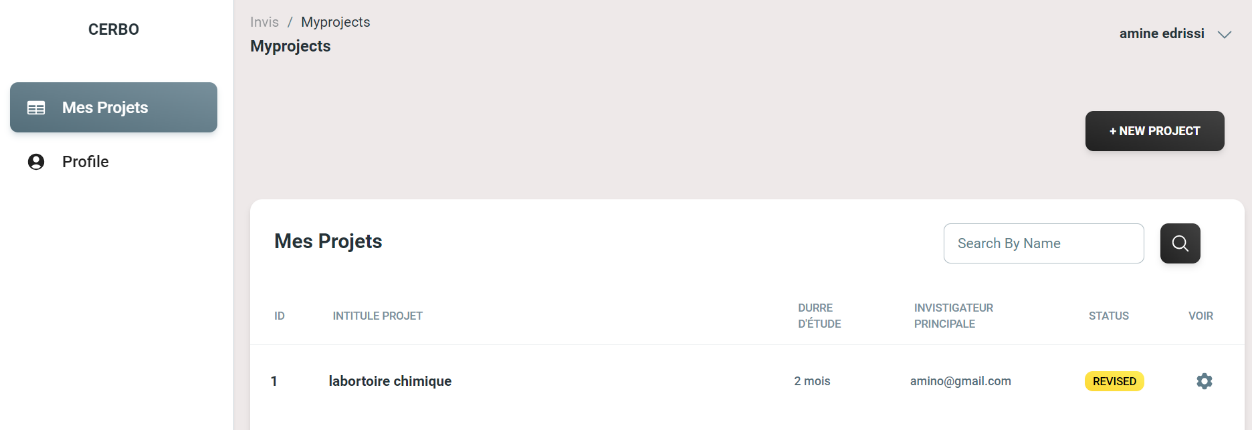
***Figure 28: rapport et début de la section des commentaires.***



***Figure 29 : section de commentaire pour le fichier descriptif.***

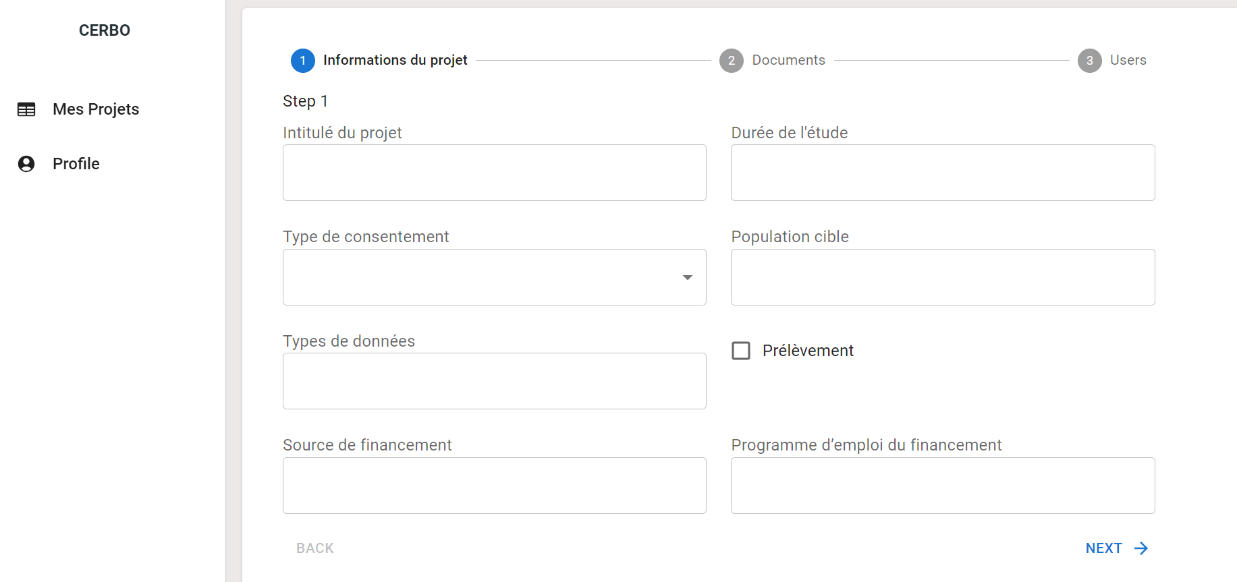
**3.2.3.3 Pour le cas d’un Investigateur (**Même les Membres ont ces fonctionnalités)

**a. Liste de projets de l’investigateur et soumission du projet :**

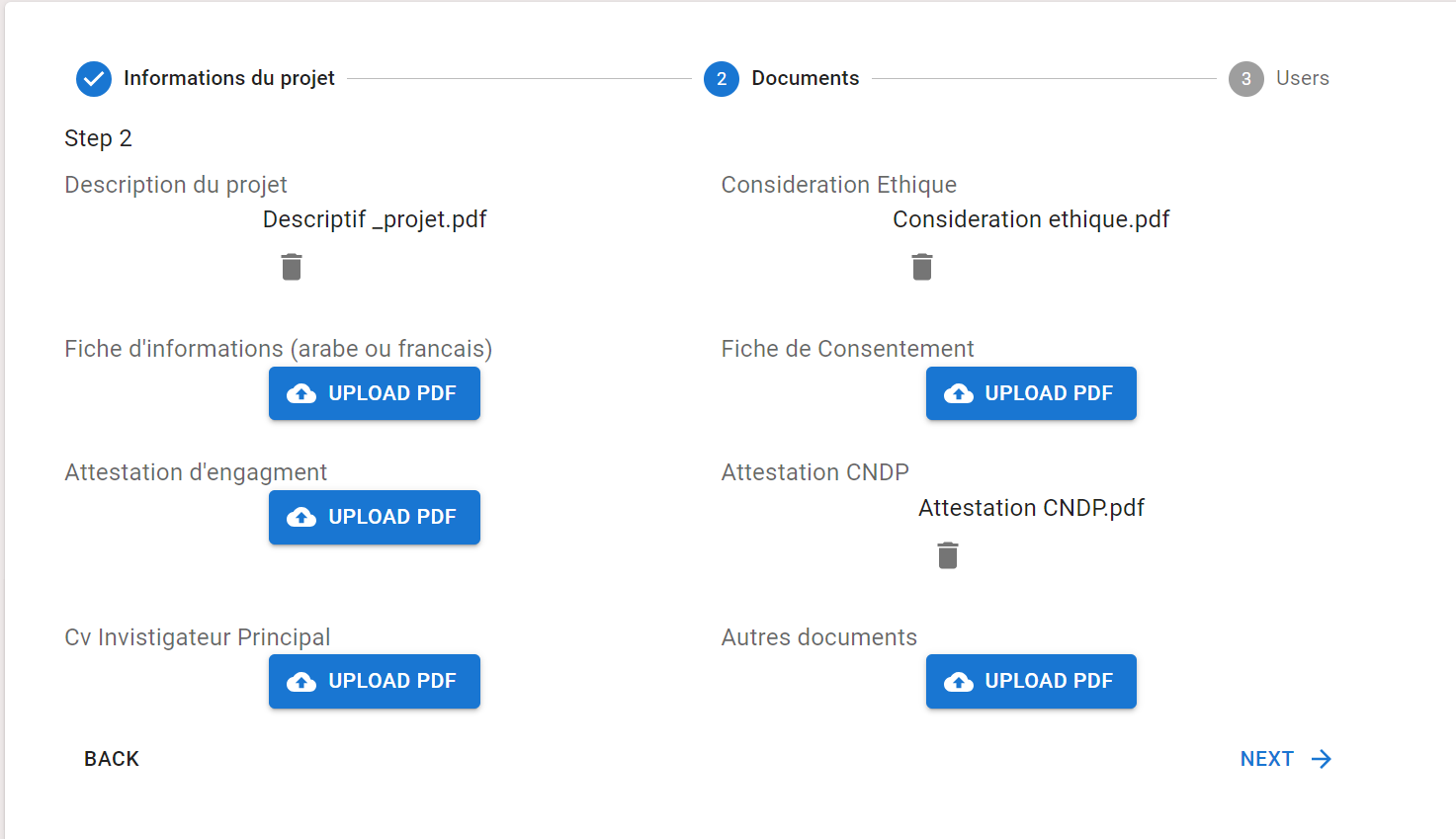
L'investigateur peut soumettre son projet et le modifier pour qu'il soit examiné par le comité, et suivre l’état de projet, et consulter le rapport, et l’avis final.

***Figure 30 : la liste des projets d’utilisateur.***

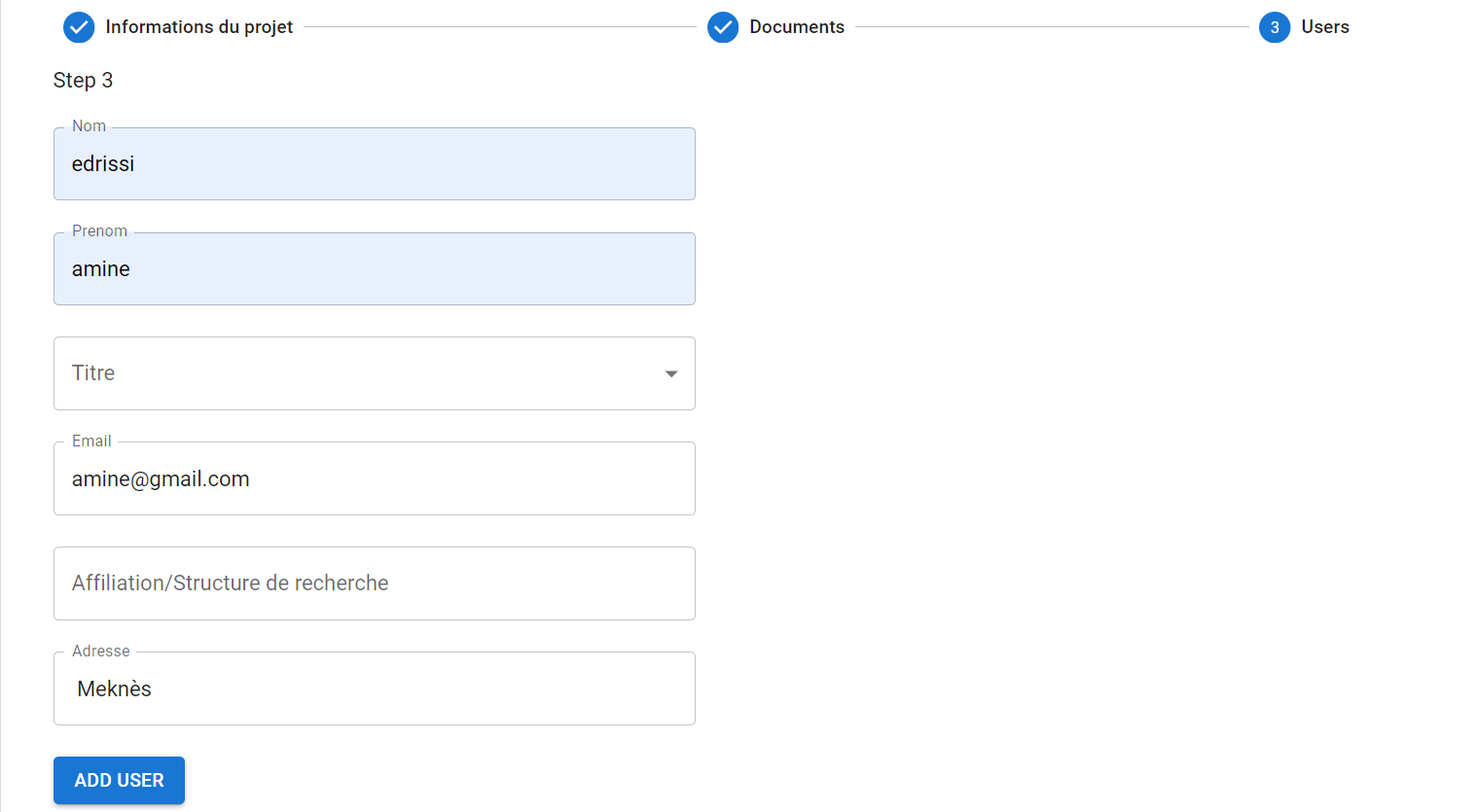
**Soumission du projet :**



***Figure 31 : la premiere étape du soumission***

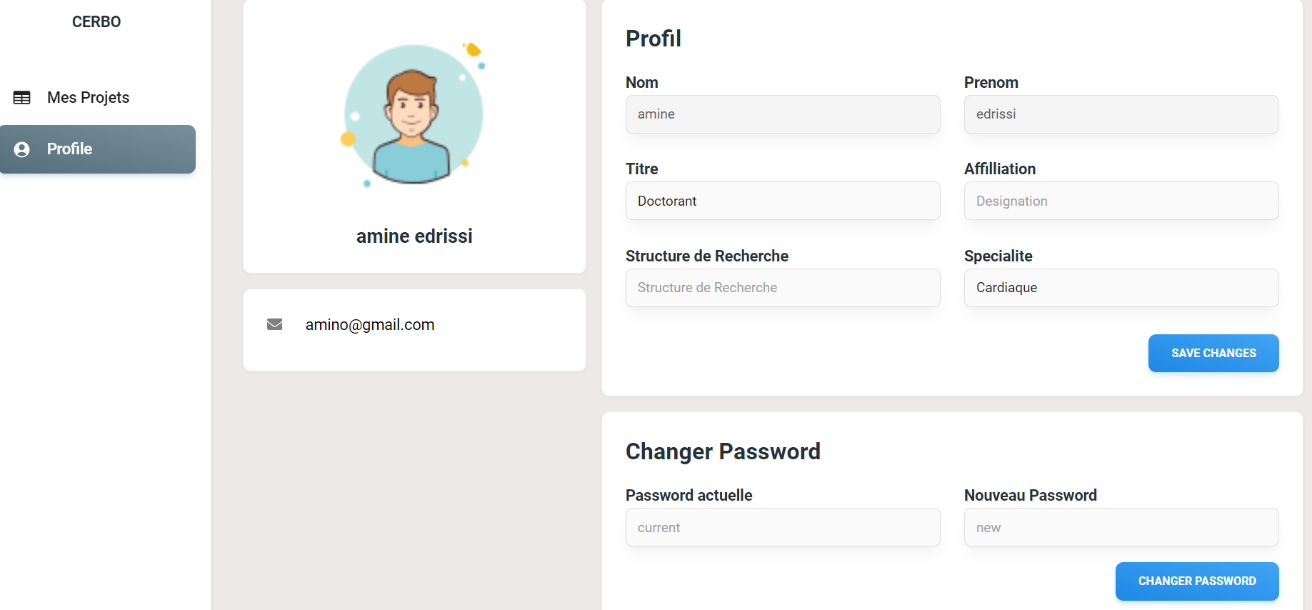


***Figure 32: la deuxième étape du soumission***

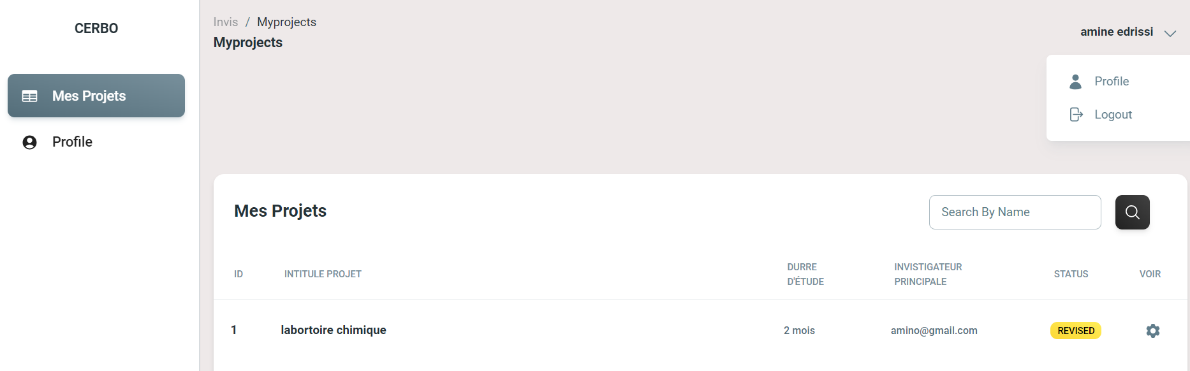


***Figure 33 : la troisième étape de soumission***

**b. voir et éditer le profil :**

L’investigateur peut voir et modifier les informations du profile et changer le Password.

***Figure 34 : Voir le Profil***

**c. Logout :**

***Figure 35 : Logout***

Après **logout**, l’utilisateur va se diriger vers la page **Se Connecter.**

**3.3 Scénarios de tests et résultats obtenus**

**3.3.1 Scénarios de tests réalisés :**

1. **Test d'authentification :**
   * **Objectif :** Vérifier la connexion avec des identifiants valides et invalides.
   * **Résultat :** Connexion réussie avec des identifiants valides, échec avec des identifiants invalides.
2. **Test de soumission de projet par un investigateur :**
   * **Objectif :** Assurer que les projets peuvent être soumis et enregistrés correctement.
   * **Résultat :** Soumission et enregistrement des projets réussis, confirmation envoyée.
3. **Test d'évaluation de projet par un membre du comité :**
   * **Objectif :** Vérifier la capacité d'évaluer et de commenter les projets.
   * **Résultat :** Évaluations enregistrées correctement, projets mis à jour.
4. **Test de gestion des utilisateurs par un administrateur :**
   * **Objectif :** Vérifier la création, modification et suppression des comptes utilisateurs.
   * **Résultat :** Gestion des comptes réussie.

**3.3.2 Résultats des tests et mesures prises :**

* **Authentification :**
  + **Problèmes :** Aucun.
  + **Mesures :** N/A.
* **Soumission de projet :**
  + **Problèmes :** Validation insuffisante de certains champs.
  + **Mesures :** Ajout de validations supplémentaires.
* **Évaluation de projet :**
  + **Problèmes :** Problèmes de synchronisation des évaluations.
  + **Mesures :** Amélioration de la synchronisation.
* **Gestion des utilisateurs :**
  + **Problèmes :** Erreurs de permission.
  + **Mesures :** Correction des règles de gestion des permissions.

Ces tests et les ajustements effectués ont garanti que la plateforme est fonctionnelle, sécurisée et répond aux besoins des utilisateurs.

Conclusion :

Dans ce rapport, nous avons détaillé les étapes de notre projet de fin d'année, consistant à développer une plateforme de gestion des projets de recherche pour le CERBO. Nous avons couvert :

* **Introduction :** Contexte et objectif général du projet.
* **Contexte, Problématique, Objectifs et Méthodologie :** Présentation de la problématique de la gestion manuelle des projets et des objectifs du projet, ainsi que la méthodologie Agile utilisée.
* **Conception, Modélisation et Planification :** Analyse des besoins, planification du projet et conception du système.
* **Réalisation, Implémentation et Tests :** Description des outils et technologies utilisés, les étapes de développement, les scénarios de tests réalisés, et les résultats obtenus.

Les résultats obtenus montrent que la plateforme répond aux exigences définies. Les fonctionnalités essentielles telles que l'authentification, la soumission de projets, l'évaluation par le comité, et la gestion des utilisateurs fonctionnent comme prévu. Les tests ont permis de corriger plusieurs erreurs et d'améliorer la performance et la fiabilité du système.

Les apprentissages tirés incluent une meilleure compréhension de la gestion de projet en méthodologie Agile, des compétences avancées en développement avec Spring/React, et l'importance des tests rigoureux pour garantir la qualité du produit final.

Pour l'avenir, des perspectives d'amélioration pourraient inclure l'intégration d'analyses de données pour évaluer les performances des projets, l'ajout de fonctionnalités de collaboration entre chercheurs, et l'amélioration continue de l'interface utilisateur pour une meilleure expérience.