

REPUBLIQUE TOGOLAISE

Travail-Liberté-Patrie

PRESIDENCE DE LA REPUBLIQUE

**MINISTERE DE LA PLANIFICATION
DU DEVELOPPEMENT ET DE LA
COOPERATION**



**Institut Africain d'Informatique
Représentation du Togo
(IAI-TOGO)**
Téléphone : +228 22 20 47 00
Email : iaitogo@iai-togo.tg
Site Web: www.iai-togo.tg
07 BP 123456 Lome 07, TOGO

EDPAGE DATA SOLUTIONS
Téléphone : +228 92 07 67 67
Adresse : Agbalepedogan
Lome-Togo

RAPPORT DE STAGE

Type de Stage : Programmation

**MISE EN PLACE D'UNE PLATEFORME DE SUIVI
DE PROJET INFORMATIQUE**

Période : Du 18 Juin 2023 au 18 Août 2023

Rédigé et présenté par **AHONYO Kwami Mawuli Israel**

Etudiant en deuxième année Tronc commun

Année Académique : 2022-2023

Superviseur :

M. AKOMAGBE Innocent
Enseignant à IAI-TOGO

Maître de stage :

M. GNANIH Noé
Développeur Web à Edpage

REMERCIEMENTS

Grâce soit rendu à Dieu tout d'abord, pour tous ses bienfaits dans la vie de tout un chacun de nous, pour sa protection divine au cours de nos divers activités et dans tous nos déplacements.

Mes remerciements sont envoyés à l'égard de tous les acteurs qui ont permis le bon déroulement de mon stage et la réalisation de mon projet notamment les membres de ma famille pour leur soutien et leurs différents apports, mes camarades m'ayant prodigué des conseils par rapport à la réalisation de mon projet.

J'éprouve une profonde gratitude à l'endroit :

- du personnel et de la direction de l'IAI-TOGO notamment le Directeur des Affaires Académiques et Scolaires M. AMEYIKPO Kossi Nicolas, le représentant national de l'IAI-TOGO qui ont rendu disponible ce stage, ainsi que le corps enseignant.
- de la société **Edpage Data and Solutions** qui m'ont octroyé ce contrat de stage au sein de leur entreprise, qui de par leur accueil m'ont permis de m'intégrer aux pratiques de l'entreprise et d'être dans un bon état d'esprit pour travailler sur le dit projet ;
- de mon superviseur M. AKOMAGBE Innocent, enseignant à l'IAI-TOGO, qui de par ses conseils et son suivi m'ont appris les meilleurs pratiques pour la réalisation de ce projet conformément aux normes de l'IAI-TOGO ;
- de mon maître de stage M. GNANIH Noé, développeur Web, pour son aide et ses conseils lors de mon apprentissage des frameworks et de la réalisation de mon projet.

SOMMAIRE

REMERCIEMENTS	1
SOMMAIRE.....	2
LISTE DES FIGURES	3
LISTE DES TABLEAUX.....	4
INTRODUCTION	5
Partie 1 : CAHIER DES CHARGES.....	6
1.1 Présentation du sujet.....	7
1.2 Problématique du sujet.....	7
1.3 Intérêt du sujet.....	8
Partie 2: PREPROGRAMMATION	10
2.1 Etude de l'existant	11
2.2 Critique de l'existant	12
2.3 Planning prévisionnel	13
2.4 Etude détaillée de la solution.....	13
Partie 3 : REALISATION ET MISE EN OEUVRE	25
3.1 Matériels et logiciels utilisés.....	26
3.2 Sécurité de l'application.....	31
3.3 Evaluation financière de la solution	33
3.4 Présentation de l'application.....	34
CONCLUSION	48
BIBLIOGRAPHIE INDICATIVE	49
WEBOGRAPHIE INDICATIVE	50
TABLE DES MATIERES	51

LISTE DES FIGURES

Figure 1-Diagramme de cas d'utilisation du Chef Projet	15
Figure 2-Diagramme de cas d'utilisation du développeur	16
Figure 3-Diagramme de cas d'utilisation du client	17
Figure 4-Diagramme de cas d'utilisation général	18
Figure 5-Diagramme de séquence du cas d'utilisation « Ajouter un Client »	22
Figure 6-Diagramme de séquence du cas d'utilisation « Ajouter un Développeur sur une Tâche »	22
Figure 7-Diagramme d'activité du cas d'utilisation « Ajouter un Développeur sur une Tâche »	23
Figure 8-Diagramme de classe.....	24
Figure 9- Logo de MySQL.....	27
Figure 10-Logo de phpMyAdmin.....	27
Figure 11- Logo de Sequelize	28
Figure 12-Logo de JavaScript	29
Figure 13-Logo de React	29
Figure 14-Logo de Express.js	30
Figure 15-Logo de Visual Studio Code	31
Figure 16- Plan de navigation.....	42
Figure 17- Formulaire de connexion	43
Figure 18-Formulaire d'ajout d'un nouveau projet	43
Figure 19- Formulaire d'ajout de tâche	44
Figure 20- Gestion des statuts des tâches	44
Figure 21-Etats de la progression du développeur.....	45
Figure 22-Interface du client pour le suivi de ses projet	45
Figure 23- Tableau de bord pour la progression des tâches	46
Figure 24-Etats des tâches par rapport au statut et aux développeurs affectés	46
Figure 25-Liste des développeurs sur la plateforme	47

LISTE DES TABLEAUX

Tableau 1- Tableau du planning prévisionnel	13
Tableau 2- Tableau de cas d'utilisation	14
Tableau 3- Description textuelle du cas d'utilisation « Ajouter un Client »	19
Tableau 4- Description textuelle du cas d'utilisation « Ajouter un Développeur sur une Tâche »	20
Tableau 5-Tableau des matériels utilisés	26
Tableau 6- Evaluation du coût de conception	33
Tableau 7-Evaluation du coût d'hébergement	33
Tableau 8- Evaluation du coût de prestations	33
Tableau 9-Récapitulatif des coûts	34

INTRODUCTION

L'informatique occupe désormais une place centrale dans notre société moderne, apportant une multitude d'innovations technologiques qui transforment fondamentalement la manière dont les entreprises et les organisations fonctionnent. Dans cet univers numérique en constante évolution, la gestion des projets informatiques joue un rôle crucial pour assurer le succès des initiatives technologiques. En effet, elle permet de coordonner et de suivre toutes les activités liées à la réalisation d'un projet informatique, depuis sa conception initiale jusqu'à sa livraison finale.

Les risques liés à la réalisation des projets informatiques incluent les dépassements budgétaires, les retards, les erreurs de conception et les problèmes de sécurité. La numérisation des procédés de réalisation de projets permet d'améliorer la collaboration, de suivre efficacement les progrès, de réduire les erreurs et d'accéder à des données en temps réel, améliorant ainsi la réussite globale des projets. A cet effet nous avons effectué un stage du 18 Juin au 18 Août 2023 à EdPage, et nous avons été soumis au thème : « **MISE EN PLACE D'UNE PLATEFORME DE SUIVI DE PROJET INFORMATIQUE** ». Elle consiste à mettre en place une plateforme web pour permettre la coopération entre les différents acteurs pour la réalisation d'un projet informatique.

Dans ce document structuré en trois grandes parties pour une meilleure compréhension, nous exposerons en première partie les objectifs et les spécifications du projet, ensuite dans la deuxième nous étudierons l'existant, relèveront ses limites et définirons nos besoins pour la réalisation de cette solution, enfin dans la dernière partie, nous explorerons la partie réalisation et mise en œuvre, où nous implémenterons la plateforme web.

Partie 1 : CAHIER DES CHARGES

1.1 Présentation du sujet

La gestion efficace des projets informatiques est cruciale dans le contexte actuel, où les avancées technologiques et les initiatives numériques sont devenues des moteurs essentiels de la croissance et de la compétitivité des entreprises. La réussite d'un projet informatique repose sur la mise en œuvre d'une gestion rigoureuse, qui implique la coordination et l'interaction de plusieurs acteurs clés à savoir le client, le chef de projet et équipe de développement. Ayant donc reconnu l'importance capital de ces éléments nous nous assignons comme mission la conception d'une « **MISE EN PLACE D'UNE PLATEFORME DE SUIVI DE PROJET INFORMATIQUE** »

Ce projet consiste à développer une plateforme web dédiée pour gérer et suivre l'avancement des projets informatiques. Elle offre une vue d'ensemble en temps réel sur les tâches, échéances favorisant ainsi la collaboration, la prise de décision éclairée et l'efficacité du projet.

1.2 Problématique du sujet

La gestion de projet informatique est un défi complexe qui implique de nombreux acteurs, des exigences en constante évolution et des contraintes de temps et de ressources. Malheureusement, de nombreux projets informatiques échouent ou ne parviennent pas à atteindre tous leurs objectifs. Face à cette réalité, plusieurs questions se doivent d'être posées :

- Comment garantir une gestion efficace des projets informatiques et minimiser les risques d'échec ?
- Quels sont les facteurs clés de succès dans la gestion de projet informatique ?
- Comment peut-on optimiser la coordination entre les différents acteurs du projet ?
- Comment impliquer activement les clients tout au long du processus de gestion de projet ?

Au fil de notre étude, nous analyserons les avantages liés à l'utilisation d'une plateforme de suivi de projet informatique, en mettant en évidence son rôle dans

la gestion des risques, la coordination des acteurs et l'optimisation des résultats du projet.

1.3 Intérêt du sujet

1.3.1 Objectifs du projet

a- Objectif général

L'objectif général de notre thème consiste à mettre en place une plateforme de suivi de projet informatique efficace qui permettra d'améliorer la coordination, la communication et la collaboration entre les acteurs du projet.

b- Objectifs spécifiques

De façon spécifique, le projet doit permettre de :

- Concevoir une plateforme de suivi de projet informatique conviviale et intuitive ;
- Faciliter la coordination et la collaboration entre les acteurs du projet, tels que les chefs de projet, les développeurs et les clients ;
- Centraliser les informations liées au projet, y compris les tâches, les délais, les ressources et les documents ;
- Favoriser la communication efficace entre les membres de l'équipe, en facilitant les échanges d'informations ;
- Améliorer la satisfaction des clients en fournissant une visibilité accrue sur l'avancement du projet et en répondant à leurs besoins et attentes.
- Générer des rapports et des analyses pour évaluer la performance du projet et prendre des décisions basées sur des données.

1.3.2 Résultats attendus

Au terme de ce projet, nous désirons que :

- La plateforme de suivi de projet informatique soit fonctionnelle et opérationnelle ;
- Une meilleure coordination et collaboration entre les acteurs du projet soit effective ;

- Les informations du projet soient centralisées pour une meilleure visibilité et accessibilité ;
- La possibilité d'un suivi en temps réel de l'avancement du projet et des indicateurs clés de performance ;
- Une communication efficace entre les membres de l'équipe favorise l'échange d'informations et la prise de décisions ;
- La gestion des changements et des demandes des clients soit améliorée ;
- Les clients soient satisfaits grâce la visibilité accrue et à la prise en compte de leurs besoins et attentes.

Partie 2: PREPROGRAMMATION

2.1 Etude de l'existant

- **Communication initiale et planification :**

La première partie consiste en la prise de contact avec le client pour discuter du projet, ses besoins et objectifs. Cette communication peut se faire via un appel téléphonique ou un courrier électronique. Ensuite, une réunion est organisée pour comprendre en détail les exigences du projet. Une fois les besoins clairement définis, une offre technique et financière est préparée et soumise au client pour approbation. Pendant cette phase, les moyens de communication utilisés sont les appels téléphoniques, les e-mails et les réunions en face à face.

- **Développement, tests et révisions :**

La deuxième partie commence avec le développement du projet, selon le plan établi. Des réunions régulières sont organisées avec le client pour partager les progrès, recueillir des commentaires et effectuer des ajustements si nécessaire. Pour les clients disposant d'une équipe informatique, leurs membres sont mis en contact avec le chef de projet et inclus dans les plateformes de gestion de projet pour suivre l'avancement du travail et faire des rapports réguliers. Une fois le développement initial terminé, des tests de qualité sont effectués pour vérifier la performance, la sécurité et la fonctionnalité du produit. Sur la base des résultats des tests et des commentaires du client, des révisions et itérations peuvent être effectuées pour améliorer le produit final. La communication dans cette phase implique des réunions, des rapports de progrès, des échanges d'e-mails et des outils de gestion de projet partagés.

- **Livraison, déploiement et support :** La troisième partie concerne la livraison et le déploiement du produit final après avoir été approuvé par le client. Une fois livré, un support post-livraison est proposé pour fournir une assistance technique continue, des mises à jour logicielles et des correctifs de bogues. La communication durant cette phase peut se faire via des canaux tels que des appels téléphoniques, des e-mails, des tickets

de support et des réunions de suivi pour assurer le bon fonctionnement et la maintenance du produit

2.2 Critique de l'existant

La description de l'existant présente certaines lacunes qui pourraient potentiellement compromettre la satisfaction des clients et l'efficacité du processus de gestion de projet.

- **Visibilité en temps réel insuffisante :**

Une des lacunes majeures du processus de gestion de projet actuel réside dans le manque de visibilité en temps réel offerte aux clients. L'incapacité des clients à suivre activement l'évolution de leur projet peut engendrer de l'incertitude et de l'inconfort.

- **Historique des actions et tâches manquant :**

Le processus actuel souffre d'une absence d'historique des actions et tâches exécutées tout au long du projet. Cela peut entraver la traçabilité du travail accompli et compliquer le suivi de l'avancement global

- **Communication limitée et non structurée :**

Bien que des canaux de communication tels que les réunions, les e-mails et les appels téléphoniques soient actuellement utilisés, le processus pourrait bénéficier d'une communication plus structurée et efficace. L'intégration d'outils de collaboration en ligne, comme des tableaux de bord de projet partagés et des espaces de travail virtuels, permettrait une interaction plus fluide entre les équipes internes, les clients et les parties prenantes.

- **Engagement client limité :**

L'implication active des clients dans le processus semble être minimisée après la phase initiale de planification. Pour améliorer cette dynamique, il serait bénéfique de favoriser une participation continue des clients tout au long du projet.

2.3 Planning prévisionnel

Tableau 1- Tableau du planning prévisionnel

Numéro	Tâches	Date de début	Date de fin	Durée (jours)
1	Insertion	18 juin 2023	20 juin 2023	3
2	Choix et étude du thème	21 juin 2023	30 juin 2023	10
3	Maîtrise des frameworks	01 juillet 2023	08 juillet 2023	8
4	Rédaction du rapport de stage	18 juin 2023	20 août 2023	64
5	Analyses et rédaction du cahier des charges	7 juillet 2023	11 juillet 2023	5
6	Conception et mise à jour des diagrammes	12 juillet 2023	02 août 2023	22
7	Développement	21 juillet 2023	18 août 2023	29
8	Finalisation du rapport de stage	16 août 2023	20 août 2023	4

2.4 Etude détaillée de la solution

2.4.1 Identification des acteurs

Pour notre projet nous avons identifiés trois différents acteurs à savoir le Chef de projet, le Client et le Développeur.

a- Chef de Projet :

Il est responsable de la gestion globale du projet. Il assure la coordination de toutes les activités entre autres la création des projets et des tâches, la gestion des développeurs, la création des comptes des autres profils.

b- Client

Il est le bénéficiaire final du projet. Il définit les objectifs et les exigences, fournit les spécifications du projet et s'assure que le résultat final répond à ses attentes.

c- Développeur

Il est chargé de la mise en œuvre technique du projet. Il lui est assigné une ou plusieurs tâches. Il peut changer le statut des tâches qui lui sont accordées

2.4.2 Identification des cas d'utilisation

2.4.2.1 Tableau des cas d'utilisation

Les différents cas d'utilisation dans notre projet sont dans le tableau ci-après :

Tableau 2- Tableau de cas d'utilisation

CAS D'UTILISATION		ACTEURS
Gérer un projet	Ajouter un projet	Chef de projet
	Modifier les informations d'un projet	
	Modifier l'état d'un projet	
Gérer les utilisateurs	Ajouter un compte développeur	
	Ajouter un compte client	
Gérer les développeurs d'un projet	Ajouter un développeur sur un projet	
	Ajouter un développeur sur une tâche	
	Retirer un développeur d'une tâche	
	Retirer un développeur d'un projet	
Créer une section d'un projet		Chef de projet
Modifier l'état d'une tâche		Développeur
Gérer une tâche	Ajouter une tâche	Chef de projet
	Supprimer une tâche	
	Modifier les informations d'une tâche	
Interagir avec les autres acteurs	Envoyer un fichier	Chef de projet, Développeur, Client
	Consulter les fichiers reçus	
	Envoyer un commentaire	
Voir l'état d'un projet		Chef de projet, Développeur, Client
Connexion à la plateforme		Chef de projet, Développeur, Client

2.4.3 Diagrammes de cas d'utilisation

2.4.3.1 Diagramme de cas d'utilisation du chef projet



Figure 1-Diagramme de cas d'utilisation du Chef Projet

2.4.3.2 Diagramme de cas d'utilisation du développeur

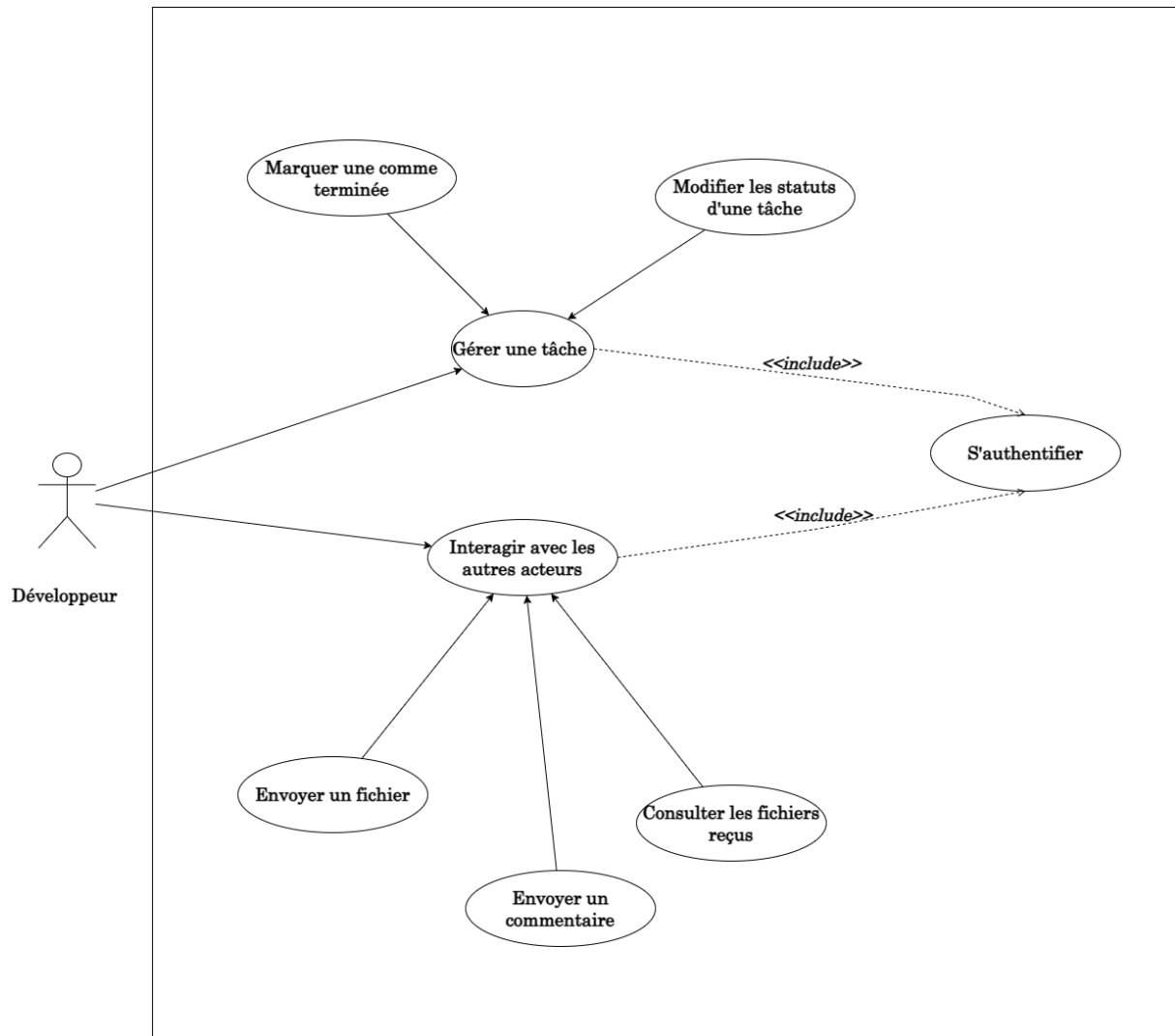


Figure 2-Diagramme de cas d'utilisation du développeur

2.4.3.3 Diagramme de cas d'utilisation du client

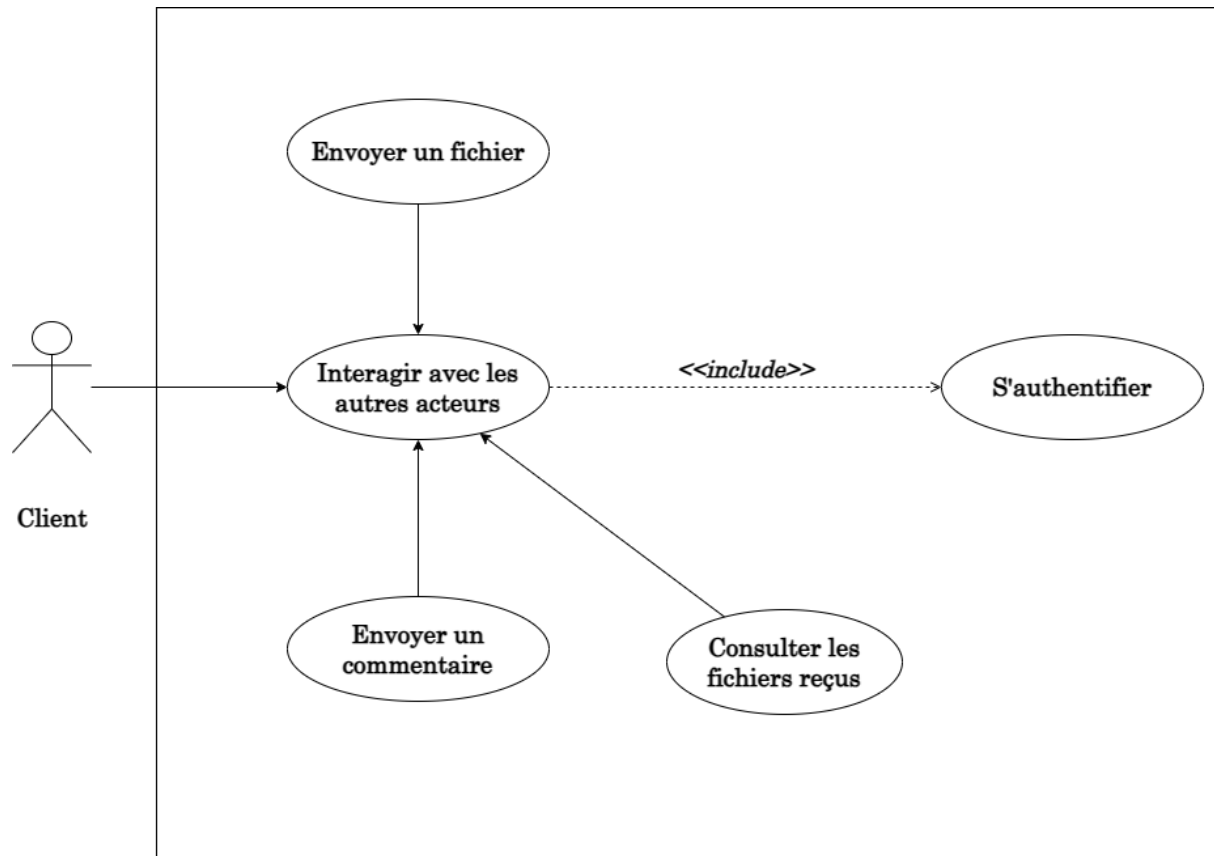


Figure 3-Diagramme de cas d'utilisation du client

2.4.3.4 Diagramme de cas d'utilisation général

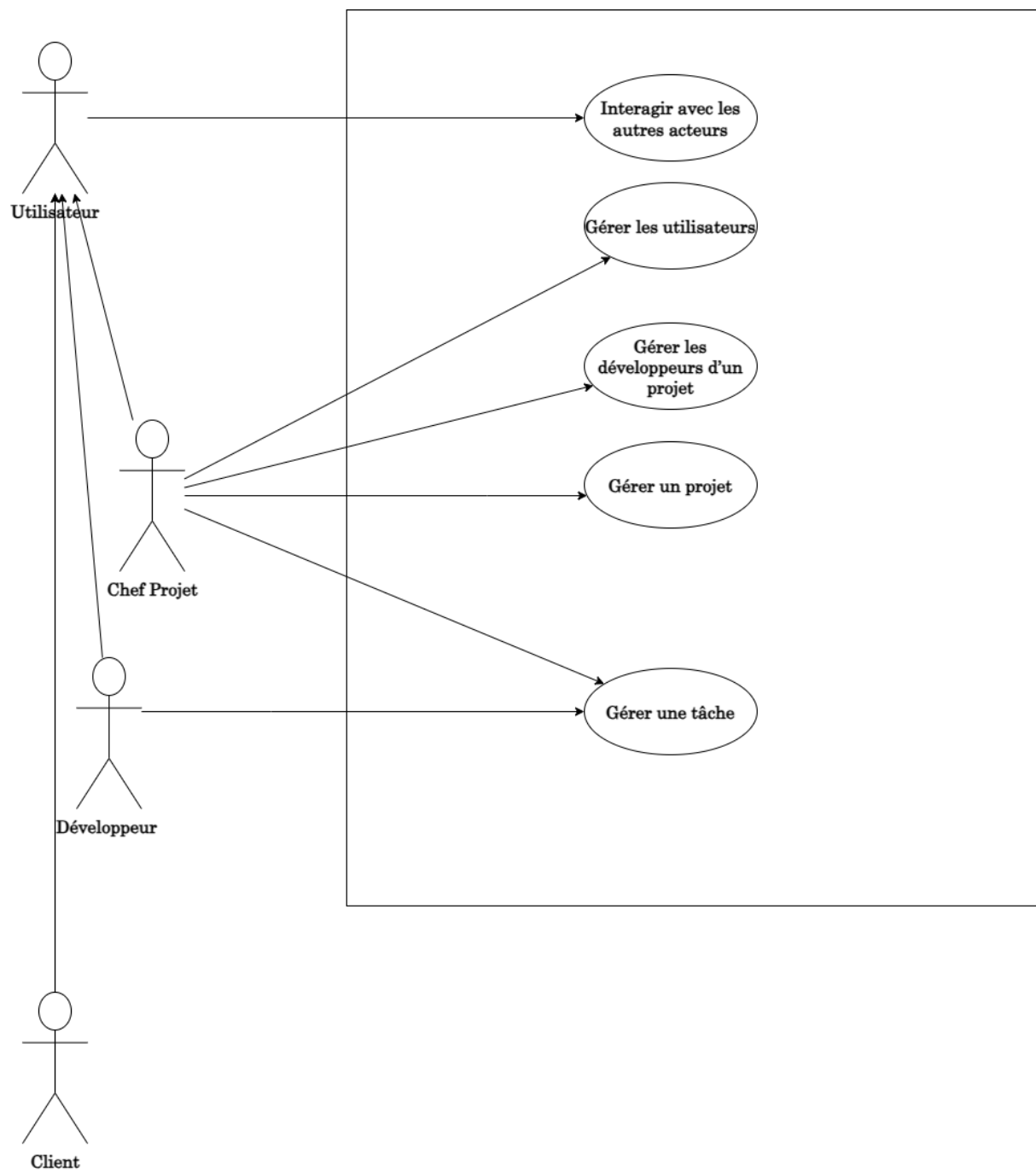


Figure 4-Diagramme de cas d'utilisation général

- **Cas d'utilisation « Ajouter un Client »**

Tableau 3- Description textuelle du cas d'utilisation « Ajouter un Client »

SOMMAIRE D'IDENTIFICATION	
<p>Titre : Ajouter un Client</p> <p>Résumé : Ce cas d'utilisation permet à un Chef de Projet d'ajouter un nouveau client à un projet en cours.</p> <p>Acteur principal : Chef de Projet</p> <p>Date de création : 03/08/2023</p> <p>Date de la dernière modification : 20/08/2023</p> <p>Version : 2.0</p> <p>Préconditions :</p> <ul style="list-style-type: none"> • Le Chef de Projet doit être connecté à la plateforme. 	
DESCRIPTION DES SCÉNARIOS	
Scénario nominal :	
<ol style="list-style-type: none"> 1. Le Chef de Projet navigue vers la page de gestion des clients en cliquant sur l'onglet Clients. 2. Le Chef de Projet sélectionne l'option « Ajouter un Client ». 3. Le système affiche un formulaire de saisie pour les informations du client 4. Le Chef de Projet remplit le formulaire avec les détails du client. 5. Le système vérifie si les champs obligatoires sont remplis (A1) 6. Le Chef de Projet confirme l'ajout du client (E1). 7. Le système vérifie si le nouveau client saisi n'est pas déjà enregistré(A2) 8. Le système enregistre les informations du client et envoie une notification de confirmation au Chef de Projet. 	
Scénario alternatif	
<p>A1 : Les données saisies sont invalides</p> <p>6. Le formulaire affiche que les données sont obligatoires</p> <p>Le déroulement reprend au point 4 du scenario nominal</p> <p>A2 : Le client est déjà enregistré</p> <p>8. Le système affiche que le client est déjà enregistré</p>	

Le déroulement reprend au point 4 du scenario nominal
Scénario d'exception :
<ul style="list-style-type: none"> • <u>E1</u> : Le Chef de Projet annule l'ajout du client : <p>Le scénario nominal est interrompu.</p> <p>Le formulaire d'ajout de client se ferme</p> <p>Post-conditions :</p> <p>Les informations du client sont enregistrées dans la base de données, permettant une communication et une collaboration efficaces tout au long du projet.</p>

- **Cas d'utilisation « Ajouter un Développeur sur une Tâche »**

Tableau 4- Description textuelle du cas d'utilisation « Ajouter un Développeur sur une Tâche »

SOMMAIRE D'IDENTIFICATION
<p>Titre : Ajouter un Développeur sur une Tâche</p> <p>Résumé : Ce cas d'utilisation permet à un Chef de Projet d'assigner un développeur spécifique à une tâche particulière au sein d'une section d'un projet.</p> <p>Acteur principal : Chef de Projet</p> <p>Date de création : 03/08/2023</p> <p>Date de la dernière modification : 20/08/2023</p> <p>Version : 2.0</p> <p>Préconditions :</p> <ul style="list-style-type: none"> • Le Chef de Projet est connecté à la plateforme. • Au moins un projet existe • Au moins une tâche existe • Le développeur est déjà enregistré sur la plateforme
DESCRIPTION DES SCÉNARIOS
Scénario nominal :
<ol style="list-style-type: none"> 1. Le Chef de Projet navigue vers la page de gestion des développeurs. 2. Le Chef de Projet accède à la liste des développeurs 3. Le Chef de Projet clique sur le bouton l'option « Assigner à une tâche » 4. Le système affiche la liste des différentes tâches enregistrées.

5. Le Chef de Projet sélectionne une tâche dans la liste.
6. Le système enregistre l'assignation du développeur à la tâche et envoie message de succès.

Post-conditions :

- Le développeur est assigné à la tâche spécifique dans le projet. Les informations sur cette assignation sont enregistrées dans le système.

• Diagramme de séquence du cas d'utilisation « Ajouter un Client »

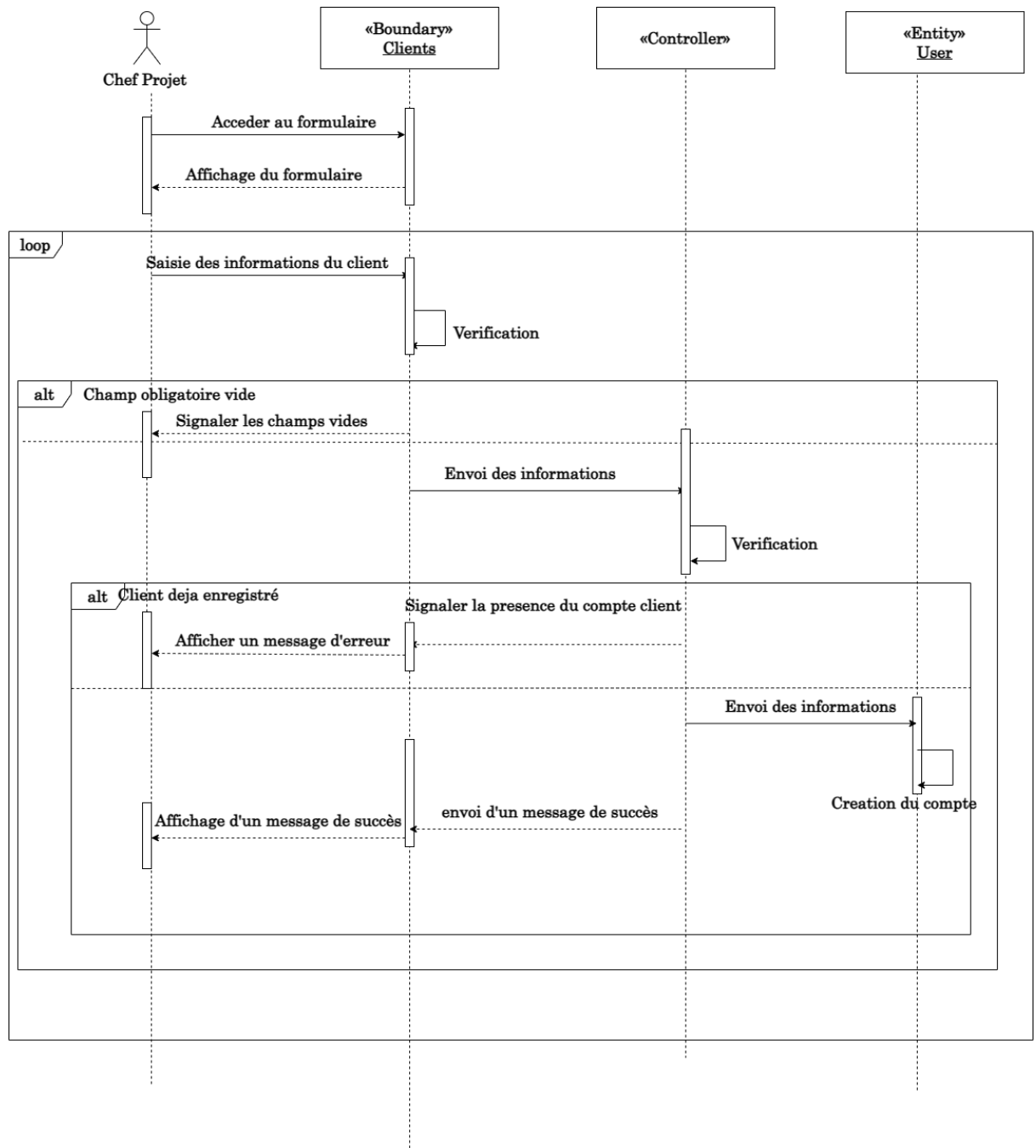


Figure 5-Diagramme de séquence du cas d'utilisation « Ajouter un Client »

- Diagramme de séquence du cas d'utilisation « Ajouter un Développeur sur une Tâche »

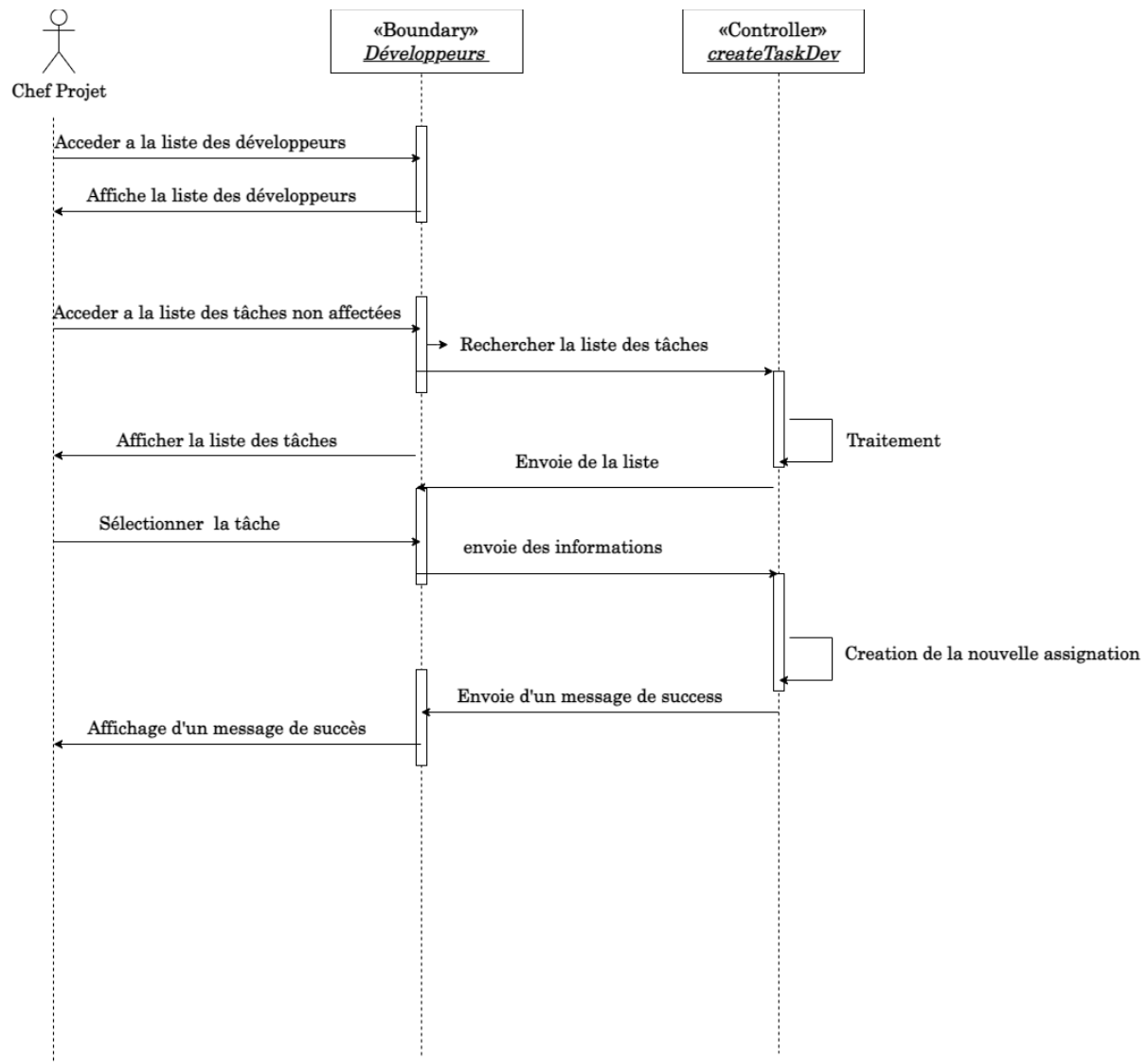


Figure 6-Diagramme de séquence du cas d'utilisation « Ajouter un Développeur sur une Tâche »

Diagramme d'activité du cas d'utilisation « Ajouter un Développeur sur une Tâche »

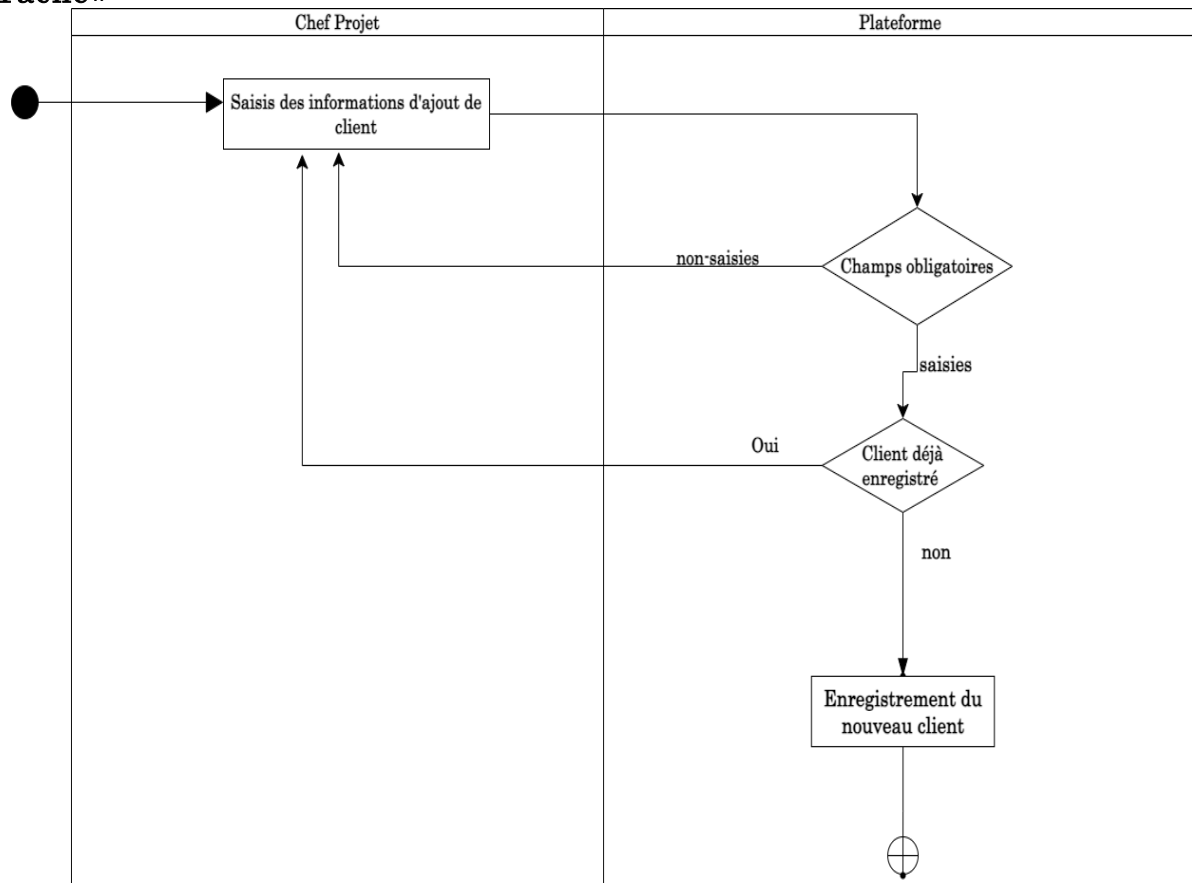


Figure 7-Diagramme d'activité du cas d'utilisation « Ajouter un Développeur sur une Tâche »

2.4.1 Diagrammes de classes

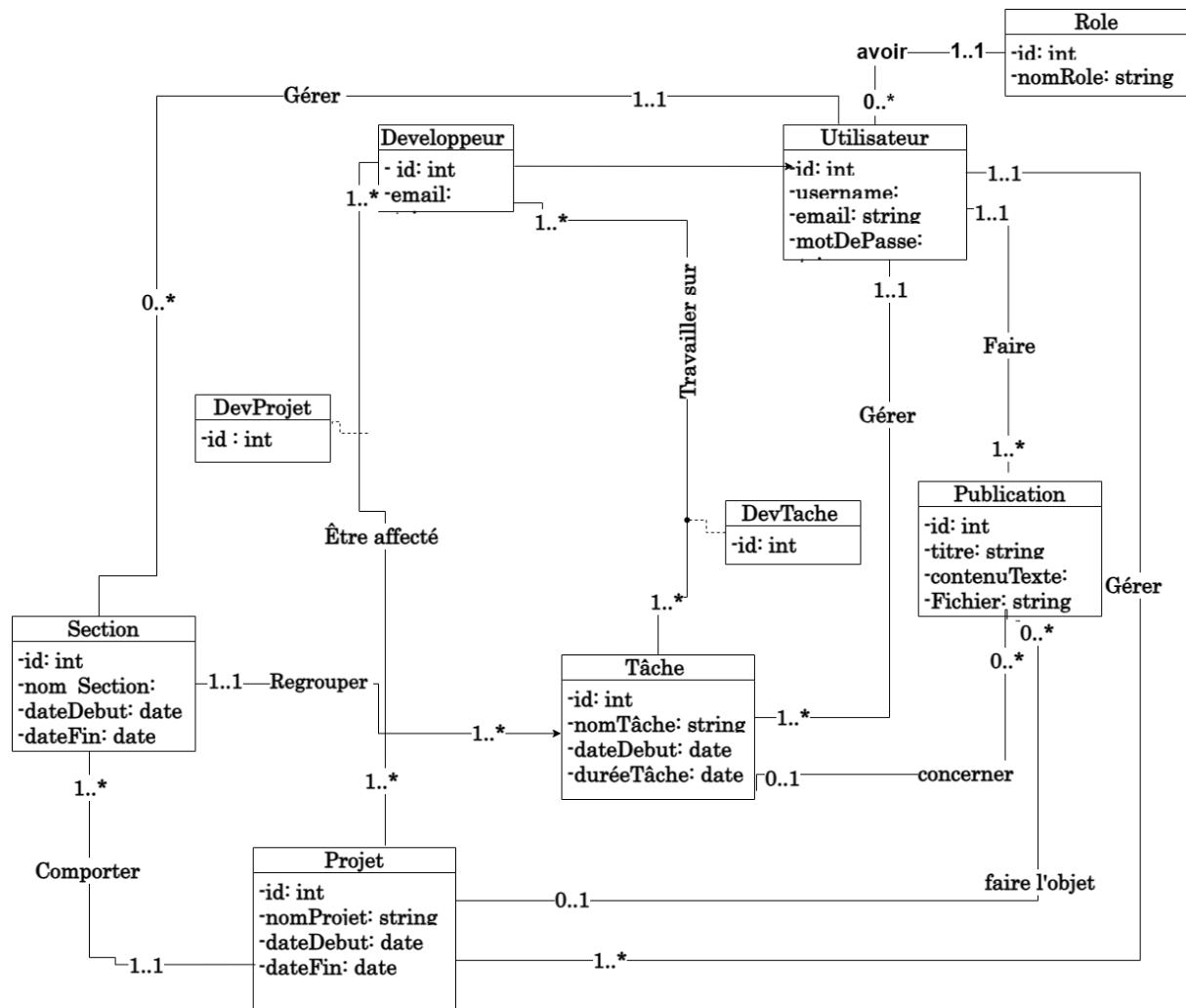


Figure 8-Diagramme de classe

Partie 3 : REALISATION ET MISE EN OEUVRE

3.1 Matériels et logiciels utilisés

3.1.1 Matériels utilisés

Dans le cadre de la réalisation de notre projet de stage, nous avons utilisé un ordinateur portable dont les caractéristiques sont les suivantes :

Tableau 5-Tableau des matériels utilisés

Description	Valeur
Modèle de l'appareil	HP
Processeur	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
Mémoire RAM installée	8.00 GB
Type du système	64-bit système d'exploitation, processeur x64
Taille du disque dur	1 To
Edition	Windows 10 Pro
Version	22H2
Installé le	14/4/2023

3.1.2 Logiciels utilisés

MySQL est un système de gestion de base de données relationnelle open-source largement utilisé pour stocker et gérer les données. Il offre une performance élevée, une évolutivité et une compatibilité avec de nombreuses applications.

Alternatives possibles :

- PostgreSQL : Un autre SGBDR open-source connu pour sa robustesse, sa conformité aux normes et ses fonctionnalités avancées.
- Microsoft SQL Server : Un SGBDR développé par Microsoft offrant une intégration étroite avec les technologies Microsoft et des fonctionnalités de sécurité avancées.

Nous avons opté pour MySQL en raison de sa popularité et de sa large adoption dans l'industrie, ce qui signifie qu'il existe une multitude de ressources, de documentation et de support en ligne. De plus, MySQL offre de bonnes performances pour les applications web et est bien adapté aux projets de taille moyenne à grande.



Figure 9- Logo de MySql

PhpMyAdmin est une application web open-source qui permet de gérer facilement des bases de données MySQL à l'aide d'une interface conviviale. Elle offre des fonctionnalités telles que la création de bases de données, la gestion de tables, l'exécution de requêtes SQL et bien plus encore.

Alternatives possibles :

- **Adminer** : Une autre application web open-source pour la gestion de bases de données, offrant une interface légère et rapide pour MySQL, PostgreSQL, SQLite et d'autres SGBDR.
- **HeidiSQL** : Une application de bureau pour Windows qui permet de gérer plusieurs types de bases de données, y compris MySQL, MariaDB et PostgreSQL.

Nous avons opté pour phpMyAdmin pour sa simplicité d'installation et d'utilisation. De plus, phpMyAdmin est bien établi et largement utilisé, ce qui signifie qu'il existe une abondance de ressources en ligne pour le support et la résolution de problèmes.



Figure 10-Logo de phpMyAdmin

Sequelize est une bibliothèque JavaScript basée sur les promesses, qui facilite la gestion des bases de données relationnelles, notamment avec Node.js. Il permet

de manipuler les données de manière asynchrone en utilisant des modèles et des requêtes SQL.

Alternatives possibles :

- **TypeORM** : Une autre bibliothèque JavaScript/TypeScript qui propose un ORM (Object-Relational Mapping) pour la gestion des bases de données relationnelles avec des fonctionnalités similaires à Sequelize.
- **Bookshelf.js** : Une bibliothèque ORM basée sur Node.js qui simplifie l'accès et la manipulation des bases de données en utilisant des modèles et des requêtes.

Nous avons choisi Sequelize pour sa simplicité d'utilisation et sa documentation bien fournie. Il offre une couche d'abstraction qui facilite la création et la manipulation de modèles de données, ainsi que la gestion des relations entre eux.



Figure 11- Logo de Sequelize

JavaScript (JS) est un langage de programmation largement utilisé dans le développement web pour ajouter des fonctionnalités interactives aux sites et aux applications. Il fonctionne côté client, ce qui signifie qu'il est exécuté dans le navigateur web de l'utilisateur. JavaScript est polyvalent et peut être utilisé pour manipuler le contenu HTML, gérer les événements, effectuer des requêtes réseau, créer des animations et bien plus encore. Il offre également la possibilité de créer des applications web complexes en utilisant des bibliothèques et des frameworks.

JSX (JavaScript XML) est une extension de syntaxe pour JavaScript utilisée principalement avec la bibliothèque React.



Figure 12-Logo de JavaScript

React est une bibliothèque JavaScript populaire pour la création d'interfaces utilisateur interactives et réactives. Elle permet de diviser l'interface en composants réutilisables, facilitant ainsi le développement et la maintenance.

Alternatives possibles :

- Angular : Un framework JavaScript développé par Google pour la création d'applications web dynamiques.
- Vue.js : Une autre bibliothèque JavaScript progressive pour la création d'interfaces utilisateur, offrant une courbe d'apprentissage plus douce.

Nous avons opté pour React dans la réalisation de l'interface utilisateur de notre application pour sa simplicité et sa flexibilité. Sa manière de gérer les composants facilite la réutilisation du code. React offre une abondance de ressources en ligne et de bibliothèques tierces pour résoudre différents problèmes.

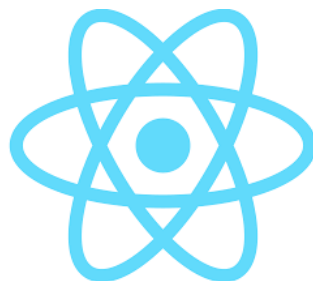


Figure 13-Logo de React

Express.js est un framework web minimaliste et rapide pour Node.js, conçu pour simplifier la création d'applications web et d'API. Il permet de gérer les routes, les middlewares et les requêtes de manière efficace.

Alternatives possibles :

- Koa.js : Un autre framework web pour Node.js axé sur la simplicité, la modularité et l'utilisation de générateurs pour gérer les flux asynchrones.
- Hapi.js : Un framework robuste pour la création d'applications web et d'API avec Node.js, axé sur la configuration, la sécurité et la documentation.

Son approche légère et sa syntaxe simple permettent de rapidement créer des routes et des middlewares, ce qui est idéal pour les projets de développement rapide. De plus, la présence de nombreuses ressources, tutoriels et packages tiers pour étendre ses fonctionnalités grâce à sa grande communauté ont fait de ce framework notre premier choix pour la réalisation de notre application web.



Figure 14-Logo de Express.js

Visual Studio Code est un éditeur de code léger et puissant développé par Microsoft. Il offre de nombreuses fonctionnalités utiles pour le développement, telles que la coloration syntaxique, les suggestions intelligentes, l'intégration Git et une vaste gamme d'extensions.

Alternatives possibles :

- Sublime Text : Un autre éditeur de code léger avec une interface épurée et une grande personnalisation.
- Atom : Un éditeur open-source développé par GitHub, également extensible via des packages.

Nous avons opté pour Visual Studio Code en raison de sa grande communauté d'utilisateurs, de son intégration transparente avec Git, de sa prise en charge d'extensions très utiles et de ses mises à fréquentes qui améliorent constamment l'expérience de développement.

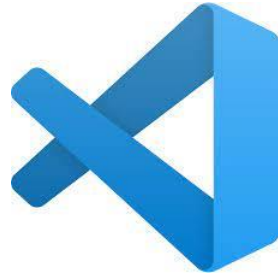


Figure 15-Logo de Visual Studio Code

3.2 Sécurité de l'application

Dans la réalisation de notre plateforme, nous avons pris des mesures de sécurité rigoureuses pour garantir la protection de notre application. Voici comment nous avons combiné les mesures de sécurité offertes individuellement par React, Sequelize et Express.js avec nos propres initiatives pour réduire au maximum les risques potentiels :

- **Mesures de sécurité fournies par React, Sequelize et Express.js :**
 - ***Protection contre les failles courantes (XSS, CSRF, injection SQL)*** : Nous avons intégré les fonctionnalités de sécurité inhérentes à React pour prévenir les attaques XSS grâce à l'utilisation de JSX et à la gestion appropriée des données dynamiques. De plus, en utilisant Sequelize, nous avons mis en œuvre des requêtes préparées pour empêcher les injections SQL. Parallèlement, Express.js nous aide à sécuriser notre application en évitant les vulnérabilités CSRF grâce à des mesures de protection appropriées.
 - ***Authentification (connexion, jetons JWT)*** : Grâce à Express.js, nous avons implémenté un système d'authentification robuste en utilisant des middlewares tels que Passport.js. Cela nous permet d'assurer

l'identification des utilisateurs de manière sécurisée. De plus, nous utilisons des jetons JWT (JSON Web Tokens) pour gérer les sessions utilisateur et les autorisations de manière sécurisée.

➤ Mesures de sécurité spécifiques à notre plateforme :

En plus des mesures de sécurité fournies par les technologies individuelles, nous avons mis en place les initiatives suivantes pour renforcer davantage la sécurité de notre application :

- **Validation des entrées utilisateur** : Nous avons mis en place des mécanismes de validation stricte des entrées utilisateur pour prévenir les attaques par injection et garantir que seules les données valides sont traitées.
-
- **Gestion fine des autorisations** : En plus de la gestion des rôles et des permissions, nous avons mis en place une stratégie d'autorisation granulaire pour nous assurer que les utilisateurs n'ont accès qu'aux ressources auxquelles ils sont autorisés.
- **Chiffrement des mots de passe** : Lorsqu'un utilisateur crée un compte ou modifie son mot de passe, nous appliquons un algorithme de hachage sécurisé avant de stocker le mot de passe dans notre base de données. Cela signifie que les mots de passe en texte brut ne sont jamais stockés, réduisant considérablement les risques en cas de violation de la base de données.

➤ Mesures de sécurité en cas de déploiement

Un serveur privé virtuel (VPS) propose un ensemble de mesures de sécurité pour assurer la protection des données et des applications hébergées. Grâce à l'isolation virtuelle, chaque VPS fonctionne indépendamment des autres, évitant ainsi la propagation des failles de sécurité. Les VPS sont généralement hébergés dans des centres de données hautement sécurisés, avec des systèmes de surveillance physique avancés et des contrôles d'accès stricts. Il est possible de mettre en place des pare-feux, des règles de sécurité personnalisées pour se prémunir contre les menaces en ligne. De plus, les fournisseurs de VPS offrent

souvent des sauvegardes automatisées et un accès sécurisé via des protocoles chiffrés, ce qui en fait une option fiable pour l'hébergement de notre application.

3.3 Evaluation financière de la solution

Tableau 6- Evaluation du coût de conception

Coût de conception de la plateforme					
Description		Unité	Quantité	Prix Unitaire	Total (FCFA)
1	Salaire du développeur	Heure	216	5000	1 080 000
					1 080 000

Tableau 7-Evaluation du coût d'hébergement

Coût d'hébergement (1)				
Plan		Stockage	RAM	Tarif Mensuel (FCFA)
1	V1-SSD (2)	20 Gb SSD	2 Gb	7 270 (3)
				7 270

Tableau 8- Evaluation du coût de prestations

Coût de prestations				
Description		Unité	Quantité	Prix Unitaire
1	Formation des utilisateurs	Heure	21	12 000
2	Maintenance	Heure	8	Offerte
				252 000

(1) Nous avons opté pour un VPS

(2) <https://shop.whois.com/virtualserverlinux-hosting.php>

(3) 12.08\$ en FCFA (20/08/2023)

Tableau 9-Récapitulatif des coûts

Récapitulatif des coûts	
Coût	Tarif (FCFA)
Coût de conception de la plateforme	1 080 000
Coût d'hébergement	7 270
Coût de prestations	252 000
	1 339 270

3.4 Présentation de l'application

3.4.1 Mise en place de la base de données

Le backend de notre application web est basé sur le framework Express js de Node. De ce fait nous avons fait usage de l'ORM « Sequelize » pour l'élaboration de notre base de données MySQL.

- Script SQL de création de la base de données

```

DROP TABLE IF EXISTS `developers`;
CREATE TABLE IF NOT EXISTS `developers` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `userId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `userId` (`userId`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

--DROP TABLE IF EXISTS `projectdevs`;
CREATE TABLE IF NOT EXISTS `projectdevs` (
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `projectId` int NOT NULL,
  `devId` int NOT NULL,
  PRIMARY KEY (`projectId`,`devId`),
  KEY `devId` (`devId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

```

```
DROP TABLE IF EXISTS `projects`;
CREATE TABLE IF NOT EXISTS `projects` (
  `id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(255) NOT NULL,
  `description` text,
  `startDate` datetime DEFAULT NULL,
  `endDate` datetime DEFAULT NULL,
  `status` varchar(255) DEFAULT 'Not Started',
  `clientId` varchar(255) DEFAULT NULL,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `userId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `name` (`name`),
  KEY `userId` (`userId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

DROP TABLE IF EXISTS `publications`;
CREATE TABLE IF NOT EXISTS `publications` (
  `id` int NOT NULL AUTO_INCREMENT,
  `title` varchar(255) NOT NULL,
  `content` text,
  `fileLink` text,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `projectId` int DEFAULT NULL,
  `sectionId` int DEFAULT NULL,
  `taskId` int DEFAULT NULL,
  `userId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `projectId` (`projectId`),
  KEY `sectionId` (`sectionId`),
  KEY `taskId` (`taskId`),
  KEY `userId` (`userId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
DROP TABLE IF EXISTS `roles`;
CREATE TABLE IF NOT EXISTS `roles` (
  `id` int NOT NULL AUTO_INCREMENT,
  `uuid` char(36) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin DEFAULT NULL,
  `roleName` varchar(255) NOT NULL,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
DROP TABLE IF EXISTS `sections`;
CREATE TABLE IF NOT EXISTS `sections` (
  `id` int NOT NULL AUTO_INCREMENT,
  `sectionName` varchar(255) NOT NULL,
  `description` text,
  `startDate` datetime DEFAULT NULL,
  `endDate` datetime DEFAULT NULL,
  `status` varchar(255) DEFAULT 'Not Started',
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `projectId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `projectId` (`projectId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
DROP TABLE IF EXISTS `taskdevs`;
CREATE TABLE IF NOT EXISTS `taskdevs` (
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `taskId` int NOT NULL,
  `devId` int NOT NULL,
  PRIMARY KEY (`taskId`,`devId`),
  KEY `devId` (`devId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
DROP TABLE IF EXISTS `tasks`;
CREATE TABLE IF NOT EXISTS `tasks` (
  `id` int NOT NULL AUTO_INCREMENT,
  `taskName` varchar(255) NOT NULL,
  `description` text,
  `durationDays` int DEFAULT NULL,
  `durationHours` int DEFAULT NULL,
  `startDate` datetime DEFAULT NULL,
  `endDate` datetime DEFAULT NULL,
  `status` varchar(255) DEFAULT 'Not Started',
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `sectionId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `sectionId` (`sectionId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `uuid` char(36) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin DEFAULT
  NULL,
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `createdAt` datetime NOT NULL,
  `updatedAt` datetime NOT NULL,
  `roleId` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `roleId` (`roleId`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;

ALTER TABLE `developers`
```

```
ADD CONSTRAINT `developers_ibfk_1` FOREIGN KEY (`userId`)
REFERENCES `users` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

```
ALTER TABLE `projectdevs`
```

```
ADD CONSTRAINT `projectdevs_ibfk_1` FOREIGN KEY (`projectId`)
REFERENCES `projects` (`id`) ON DELETE CASCADE ON UPDATE
CASCADE,
```

```
ADD CONSTRAINT `projectdevs_ibfk_2` FOREIGN KEY (`devId`)
REFERENCES `developers` (`id`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

```
ALTER TABLE `projects`
```

```
ADD CONSTRAINT `projects_ibfk_1` FOREIGN KEY (`userId`)
REFERENCES `users` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

```
ALTER TABLE `publications`
```

```
ADD CONSTRAINT `publications_ibfk_1` FOREIGN KEY (`projectId`)
REFERENCES `projects` (`id`) ON DELETE SET NULL ON UPDATE
CASCADE,
```

```
ADD CONSTRAINT `publications_ibfk_2` FOREIGN KEY (`sectionId`)
REFERENCES `sections` (`id`) ON DELETE SET NULL ON UPDATE
CASCADE,
```

```
ADD CONSTRAINT `publications_ibfk_3` FOREIGN KEY (`taskId`)
REFERENCES `tasks` (`id`) ON DELETE SET NULL ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `publications_ibfk_4` FOREIGN KEY (`userId`)
REFERENCES `users` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
```

```
ALTER TABLE `sections`
```

```
ADD CONSTRAINT `sections_ibfk_1` FOREIGN KEY (`projectId`)
REFERENCES `projects` (`id`) ON DELETE SET NULL ON UPDATE
CASCADE;
```

```
ALTER TABLE `taskdevs`
```

```
ADD CONSTRAINT `taskdevs_ibfk_1` FOREIGN KEY (`taskId`)
REFERENCES `tasks` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
ADD CONSTRAINT `taskdevs_ibfk_2` FOREIGN KEY (`devId`)
REFERENCES `developers` (`id`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

```
ALTER TABLE `tasks`
  ADD CONSTRAINT `tasks_ibfk_1` FOREIGN KEY (`sectionId`)
REFERENCES `sections` (`id`) ON DELETE SET NULL ON UPDATE
CASCADE;
ALTER TABLE `users`
  ADD CONSTRAINT `users_ibfk_1` FOREIGN KEY (`roleId`) REFERENCES
`roles` (`id`) ON DELETE SET NULL ON UPDATE CASCADE;
COMMIT;
```

- Connection a la base de données

```
const sequelize = new Sequelize(
  dbConfig.DB,
  dbConfig.USER,
  dbConfig.PASSWORD, {
    host: dbConfig.HOST,
    dialect : dbConfig.dialect,
    pool:{
      max: dbConfig.pool.max,
      min: dbConfig.pool.min,
      acquire: dbConfig.pool.acquire,
      idle: dbConfig.pool.idle
    }
  }
)

sequelize.authenticate()

.then(()=>{console.log('Connection to the database established successfully')
}).catch((err)=>{console.error(err)})
```

- Création de la table utilisateur

```
module.exports = (sequelize, DataTypes) => {
  const User = sequelize.define('user', {
    uuid:{
      type: DataTypes.UUID,
      defaultValue: DataTypes.UUIDV4,
    },
    username: {
      type: DataTypes.STRING,
```



```

    allowNull: false,
  },
  email: {
    type: DataTypes.STRING,
    allowNull: false,

    unique: true,
  },
  password: {
type: DataTypes.STRING,
    allowNull: false,
  },})
return User
}

User.hasMany(Project,{
  foreignKey:'userId'
})

```

- Exemple de création de deux tables
 - Création de la table Projet

```

module.exports=(sequelize, DataTypes)=>{
const Project = sequelize.define('project', {
  name: {
    type: DataTypes.STRING,
    allowNull: false,
    unique: true,
  },
  description: {
    type: DataTypes.TEXT,
    allowNull: true,
  },
  startDate: {
    type: DataTypes.DATE,
    allowNull: true,
    defaultValue: DataTypes.NOW,
  },
  endDate: {
    type: DataTypes.DATE,
    allowNull: true,
  },
  status: {
    type: DataTypes.STRING,
    defaultValue: 'Not Started',
    validate:{

```

```
        isIn :[['Not Started', 'In Progress', 'Completed', 'Suspended']]
    },
    },
    clientId:{
        type: DataTypes.STRING,
        allowNull: true,
    });
return Project
}
Project.belongsTo(User,{
    foreignKey: 'userId'
})
```

3.4.2 Plan de navigation

Le plan de navigation se présente, pour chaque utilisateur sur une barre verticale à gauche où se retrouvent les différents liens menant vers les différentes pages après la connexion.

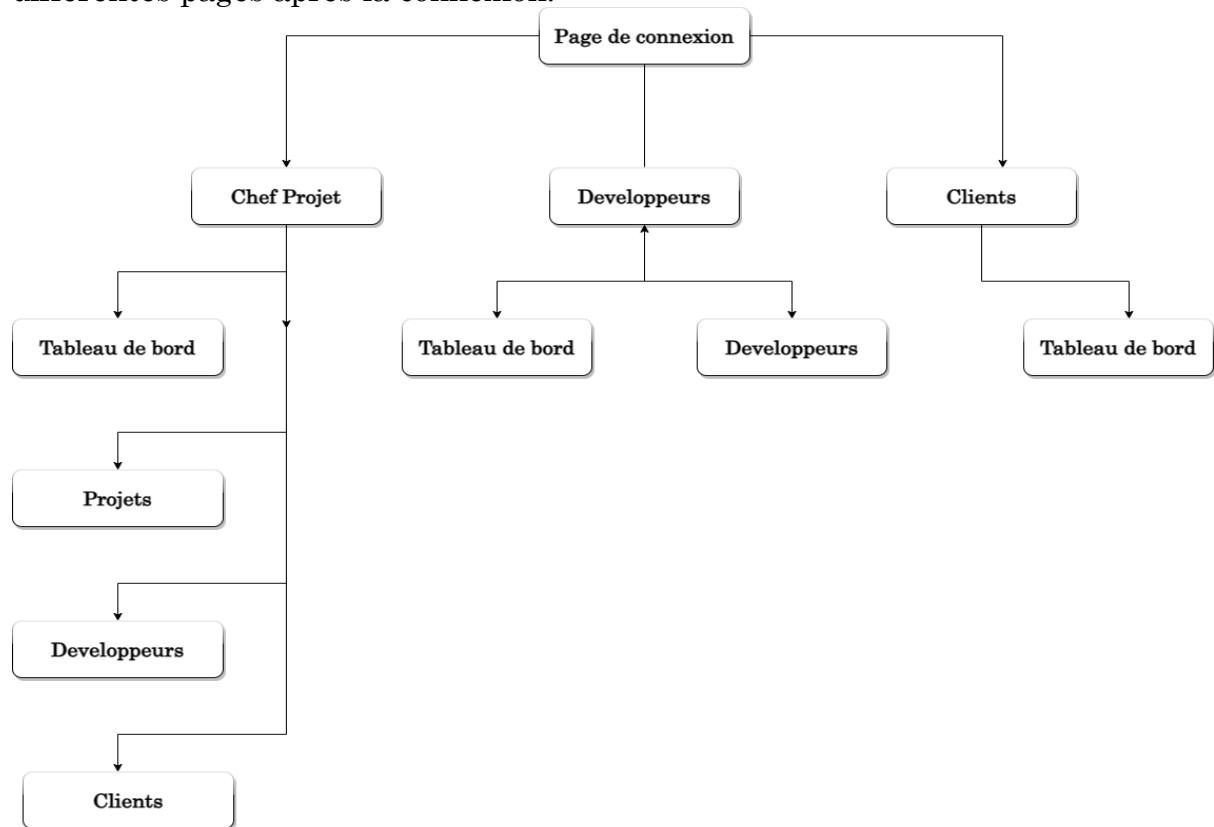


Figure 16- Plan de navigation

3.4.3 Quelques masques de saisies

-Connexion à la plateforme

Project Flow

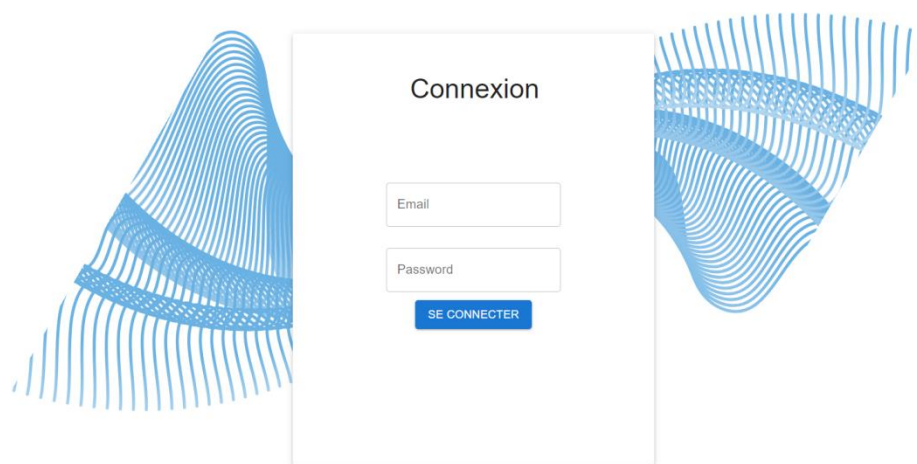
A login form titled "Connexion" is centered on a white background. It features two input fields: "Email" and "Password". Below these fields is a blue button labeled "SE CONNECTER". The form is flanked by abstract blue line art graphics.

Figure 17- Formulaire de connexion

-Ajout d'un nouveau projet

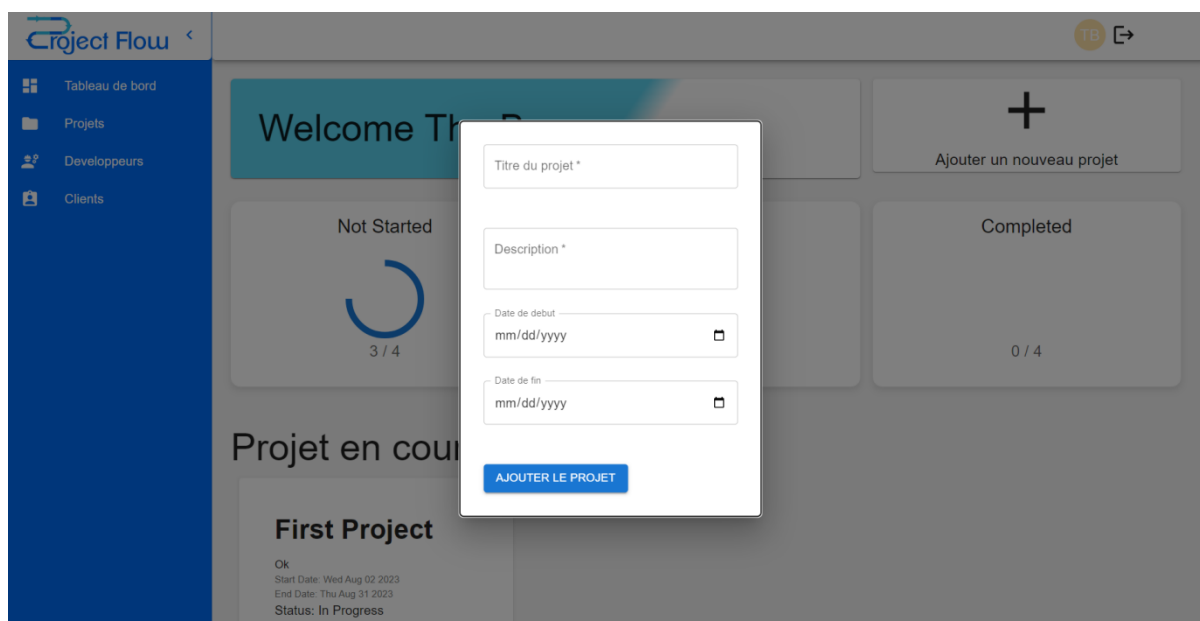
A screenshot of the Project Flow dashboard with a modal form for adding a new project. The dashboard has a blue sidebar with navigation links: "Tableau de bord", "Projets", "Developpeurs", and "Clients". The main area shows a "Welcome" message, a "Not Started" section with a circular progress indicator (3/4), and a "Completed" section (0/4). A "First Project" card is visible at the bottom. The modal form, titled "Ajouter un nouveau projet", contains fields for "Titre du projet *", "Description *", "Date de debut" (mm/dd/yyyy), and "Date de fin" (mm/dd/yyyy). A blue button labeled "AJOUTER LE PROJET" is at the bottom of the modal.

Figure 18-Formulaire d'ajout d'un nouveau projet

-Ajouter une tâche à une section

The image shows a modal form for adding a new task. The form is centered on a blurred background of a project management interface. The form has the following fields and elements:

- Nom de la tâche**: A text input field.
- Durée en jours**: A numeric input field with the value '0'.
- Durée en heures**: A numeric input field with the value '2' and a small up/down arrow icon.
- Description**: A text input field.
- AJOUTER**: A blue button with white text.

The background interface shows a sidebar with 'Authentication' and 'Security' sections, and a main area with a 'Not Started' tab and a 'Completed' tab. A 'New security' task card is visible in the background.

Figure 19- Formulaire d'ajout de tâche

-Gestion des statuts des différentes tâches

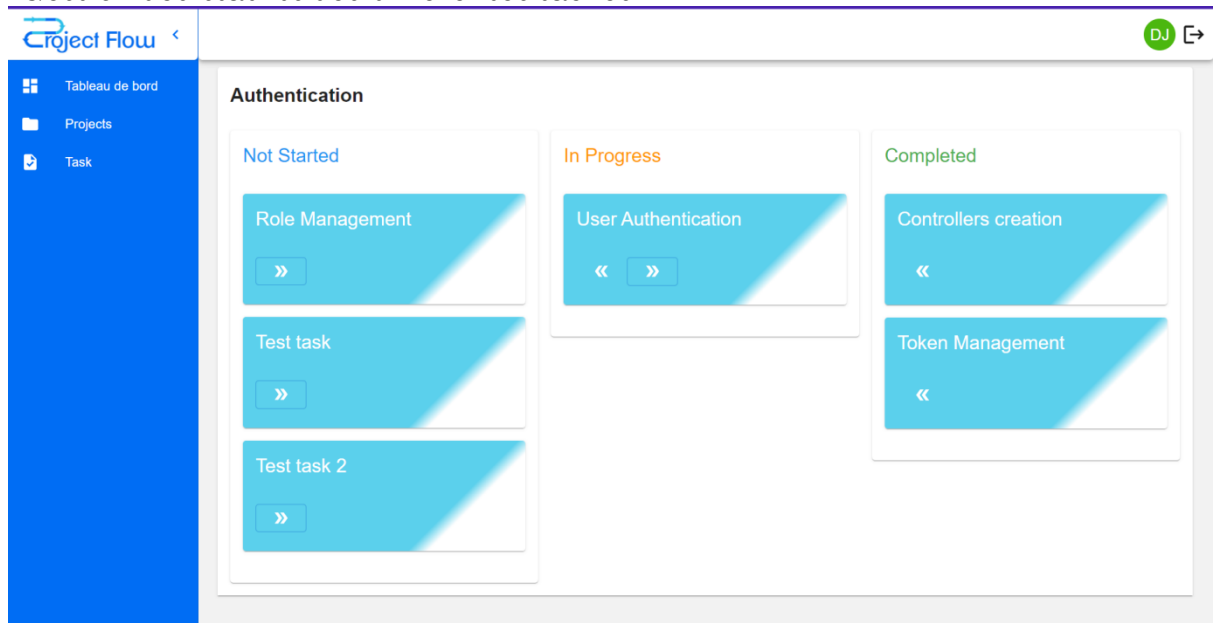


Figure 20- Gestion des statuts des tâches

3.4.4 Quelques états et statistiques

- Suivi de la progression du développeur

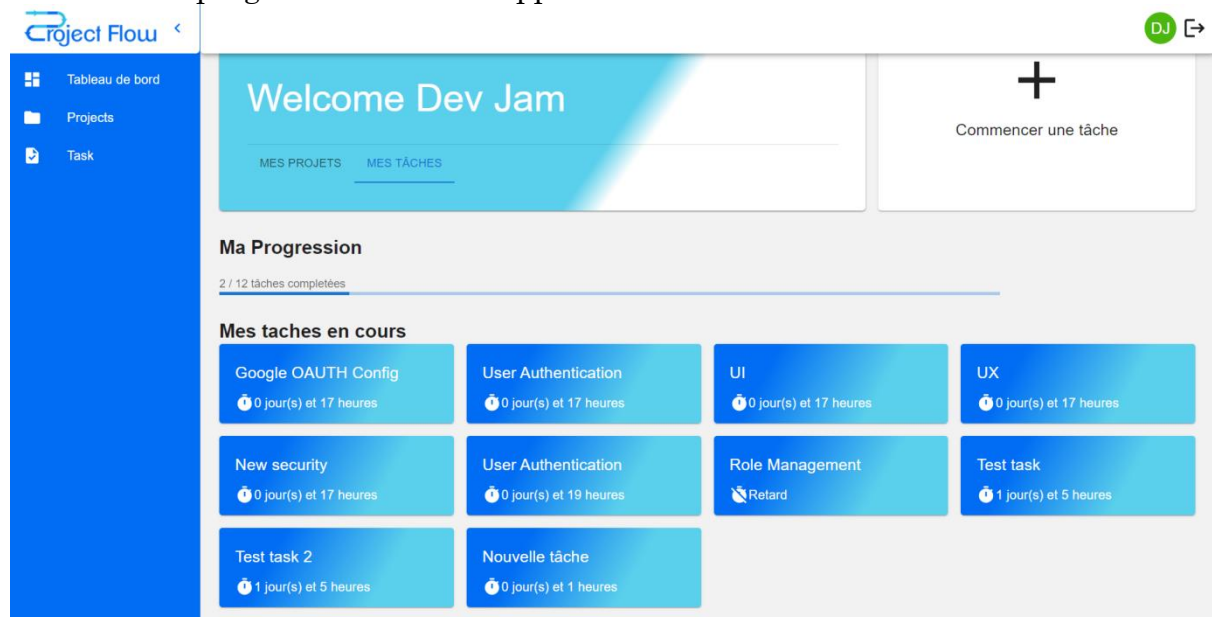


Figure 21- Etats de la progression du développeur

-Interface du client pour le suivi de ses projets

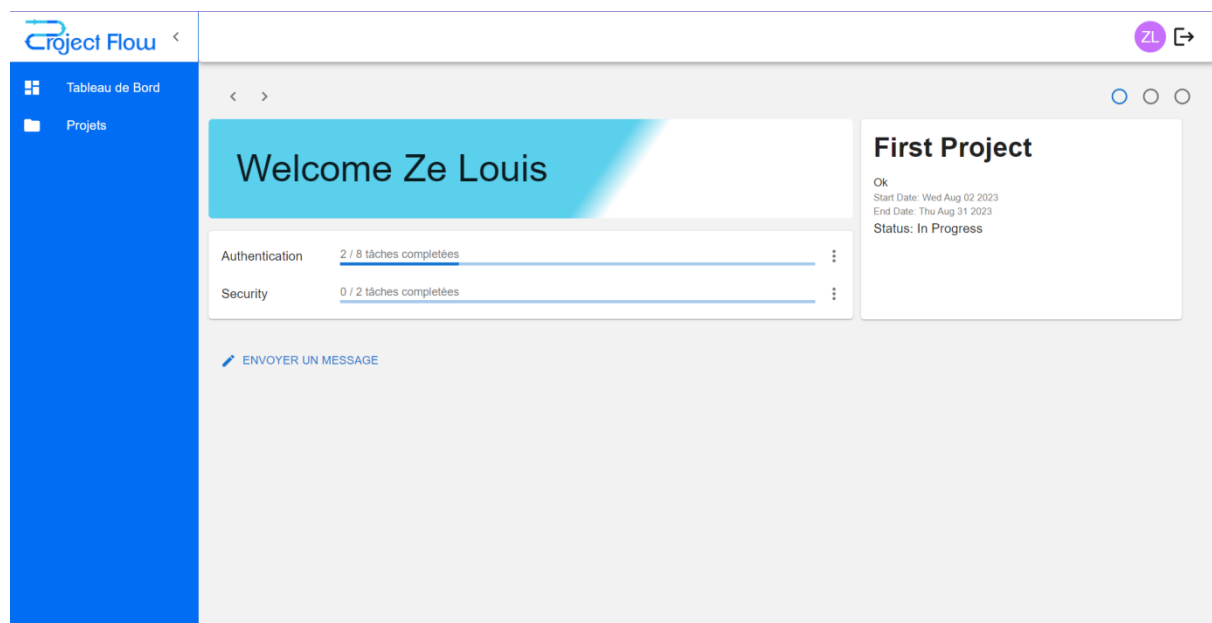


Figure 22- Interface du client pour le suivi de ses projet

-Statistiques de progression des tâches par rapport à leur statut

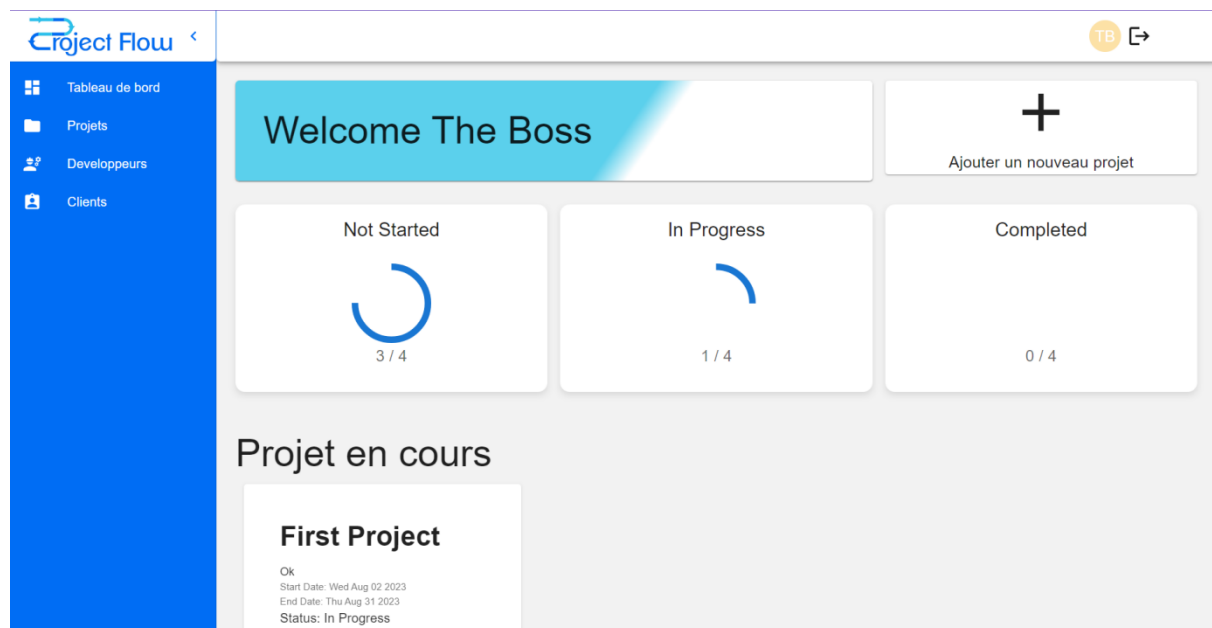


Figure 23- Tableau de bord pour la progression des tâches

-Etats des tâches par rapport au statut et aux développeurs affectés

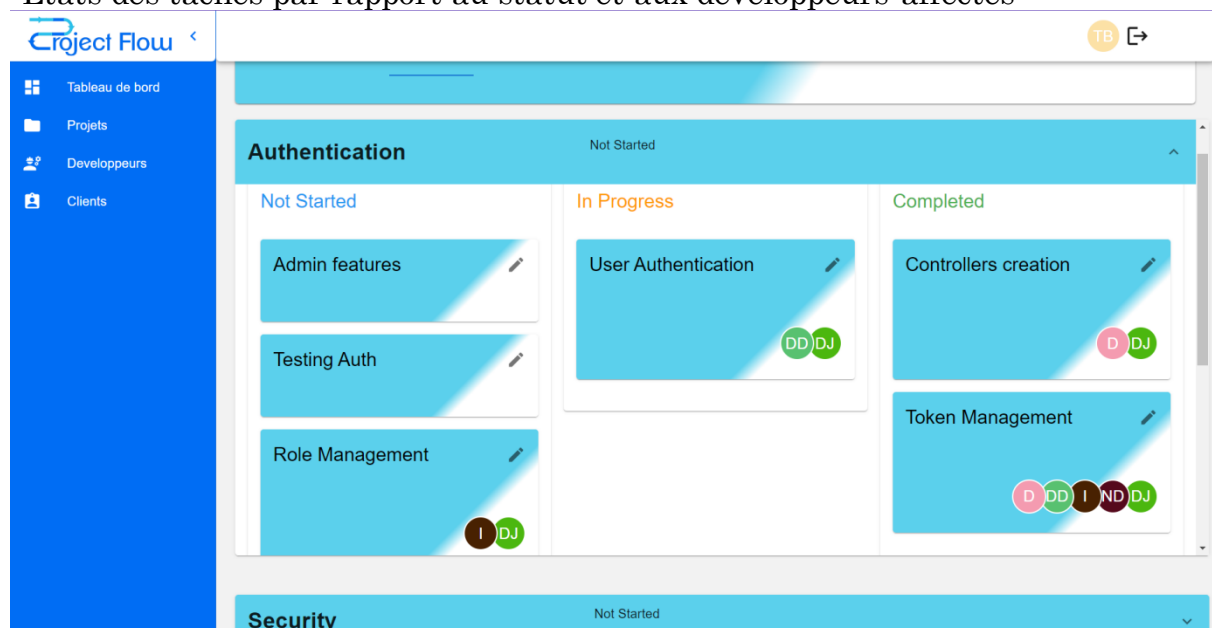
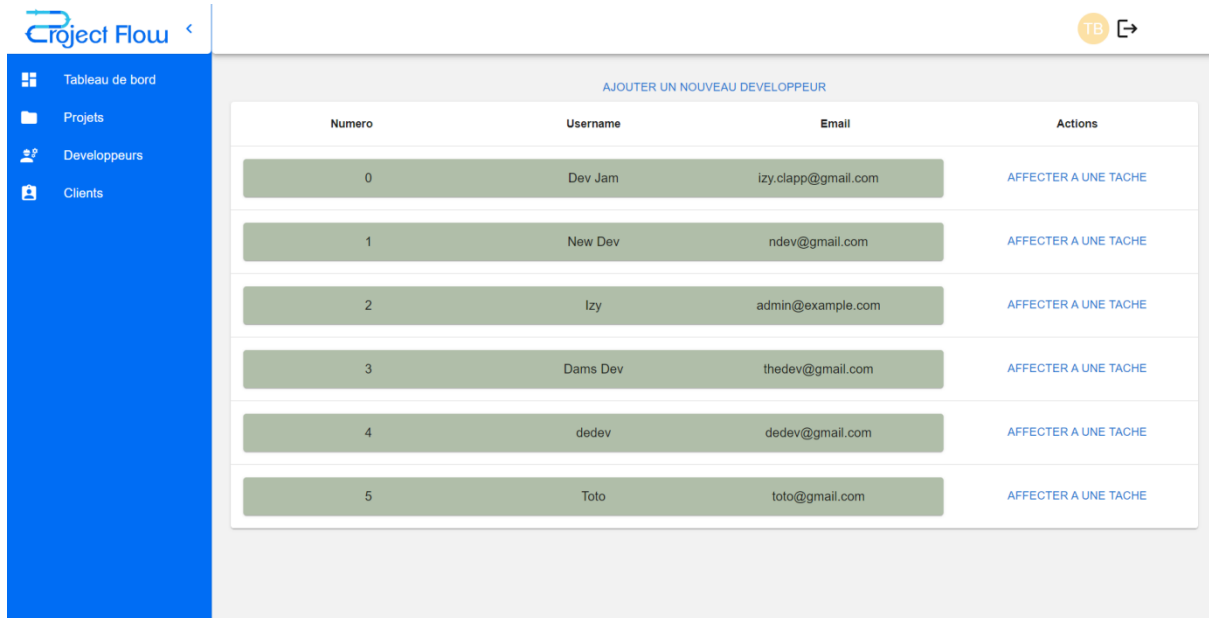


Figure 24- Etats des tâches par rapport au statut et aux développeurs affectés

-Liste des développeurs sur la plateforme



The screenshot displays the 'Project Flow' web application interface. On the left is a blue sidebar with navigation links: 'Tableau de bord', 'Projets', 'Developpeurs', and 'Clients'. The main content area has a header with the 'Project Flow' logo and a 'TB' icon. Below the header is a button labeled 'AJOUTER UN NOUVEAU DEVELOPPEUR'. The central part of the page features a table with the following data:

Numero	Username	Email	Actions
0	Dev Jam	izy.clapp@gmail.com	AFFECTER A UNE TACHE
1	New Dev	ndev@gmail.com	AFFECTER A UNE TACHE
2	Izy	admin@example.com	AFFECTER A UNE TACHE
3	Dams Dev	thedev@gmail.com	AFFECTER A UNE TACHE
4	dedev	dedev@gmail.com	AFFECTER A UNE TACHE
5	Toto	toto@gmail.com	AFFECTER A UNE TACHE

Figure 25-Liste des développeurs sur la plateforme

CONCLUSION

En conclusion, notre stage du 18 Juin au 18 Août 2023 au sein de l'entreprise EdPage nous a immergés dans le monde complexe et crucial de la gestion de projets informatiques. Notre mission était de mettre en place une plateforme web dédiée au suivi et à la coordination des projets informatiques, facilitant ainsi la collaboration entre les différents acteurs impliqués dans la réalisation de ces projets.

Notre travail au sein d'EdPage nous a permis de fusionner théorie et pratique en matière de gestion de projets informatiques. Nous avons pu constater de manière concrète comment une plateforme de suivi peut améliorer la coordination, le suivi des progrès et la communication entre les parties prenantes, aboutissant ainsi à une réalisation plus efficace et réussie des projets informatiques. Notre expérience renforce l'idée que la gestion de projets informatiques reste un pilier fondamental pour naviguer avec succès dans le paysage technologique en constante évolution.

BIBLIOGRAPHIE INDICATIVE

➤ Notes de cours

- Analyse (UML) M. **GLOKPO** Igor

➤ Anciens mémoires consultés

- Rapport de stage de M. **ANANI** Yaovi Caleb, mise en place d'une plateforme web pour uploader les fichiers lors des évaluations pratiques : cas de l'IAI-TOGO (2021-2022)
- Rapport de stage de M. **NOLITSE** Markovit, plateforme d'échange et de partage d'informations personnelles, cas du cabinet CAGECFI (2021-2022)

WEBOGRAPHIE INDICATIVE

- www.mui.com
- www.sequelize.org
- <https://dev.to/chandrapantachhetri/sending-emails-securely-using-node-js-nodemailer-smtp-gmail-and-oauth2-g3a>
- www.npmjs.com
- www.react.dev
- <https://redux.js.org/tutorials/essentials/part-2-app-structure>
- www.google.com

TABLE DES MATIERES

REMERCIEMENTS.....	1
SOMMAIRE.....	2
LISTE DES FIGURES	3
LISTE DES TABLEAUX.....	4
INTRODUCTION	5
Partie 1 : CAHIER DES CHARGES.....	6
1.1 Présentation du sujet.....	7
1.2 Problématique du sujet.....	7
1.3 Intérêt du sujet.....	8
1.3.1 Objectifs du projet	8
1.3.2 Résultats attendus	8
Partie 2: PREPROGRAMMATION	10
2.1 Etude de l'existant.....	11
2.2 Critique de l'existant	12
2.3 Planning prévisionnel	13
2.4 Etude détaillée de la solution.....	13
2.4.1 Identification des acteurs	13
2.4.2 Identification des cas d'utilisation	14
2.4.2.1 Tableau des cas d'utilisation	14
2.4.3 Diagrammes de cas d'utilisation	15
2.4.3.1 Diagramme de cas d'utilisation du chef projet	15
2.4.3.2 Diagramme de cas d'utilisation du développeur.....	16
2.4.3.3 Diagramme de cas d'utilisation du client	17
2.4.3.4 Diagramme de cas d'utilisation général	18
2.4.1 Diagrammes de classes	24
Partie 3 : REALISATION ET MISE EN OEUVRE	25
3.1 Matériels et logiciels utilisés.....	26
3.1.1 Matériels utilisés.....	26
3.1.2 Logiciels utilisés	26

3.2 Sécurité de l'application	31
3.3 Evaluation financière de la solution	33
3.4 Présentation de l'application.....	34
3.4.1 Mise en place de la base de données.....	34
3.4.2 Plan de navigation.....	42
3.4.3 Quelques masques de saisies	43
3.4.4 Quelques états et statistiques.....	45
CONCLUSION	48
BIBLIOGRAPHIE INDICATIVE	49
WEBOGRAPHIE INDICATIVE	50
TABLE DES MATIERES	51