

Rapport de Stage

Développement d'un logiciel du chiffrage
eco-pilot

Stagiaire : Mayar Briki

Encadrant académique : Nom Prénom

Encadrant professionnel : Nom Prénom

Date : October 22, 2025

Chapter 1

Introduction

Durant mon stage de deux mois, j'ai eu l'opportunité de découvrir et de participer à des projets liés à la gestion de projets et à la gestion de budgets à travers un logiciel de chiffrage. Ce stage m'a permis de comprendre l'importance de l'organisation, du suivi des coûts et de l'optimisation des ressources pour assurer le succès des projets, tout en développant mes compétences pratiques dans un environnement professionnel. **eco-pilote**.

Chapter 2

Présentation générale du projet

2.1 Contexte et objectifs

Ce projet a été réalisé dans le cadre d'un stage de deux mois.

2.2 Utilisateurs cibles

L'application s'adresse principalement aux professionnels impliqués dans la gestion de projets et le chiffrage budgétaire. Les utilisateurs cibles incluent :

- **Chefs de projet** : responsables de la planification, du suivi et de la gestion des ressources et des coûts des projets.
- **Responsables financiers** : chargés de la validation des budgets, du contrôle des dépenses et de l'optimisation des coûts.
- **Membres d'équipes techniques** : intervenant dans la saisie des besoins, la consultation des articles et la remontée d'informations sur l'avancement des tâches.
- **Administrateurs** : assurant la gestion des utilisateurs, des droits d'accès et la configuration générale de l'application.
- **Clients ou partenaires externes** : pouvant accéder à certaines informations ou suivre l'avancement des projets selon les droits qui leur sont attribués.

Cette diversité d'utilisateurs implique une gestion fine des rôles et des permissions afin de garantir la sécurité et la pertinence des informations accessibles à chacun.

2.3 Fonctionnalités principales

Les principales fonctionnalités du projet incluent :

- **Gestion des utilisateurs** : Création, modification, suppression et gestion des rôles des utilisateurs. Chaque utilisateur dispose d'un profil personnalisé avec des droits d'accès adaptés à sa fonction (administrateur, chef de projet, membre technique, etc.). Un système d'authentification sécurisé permet de garantir la confidentialité des données et la traçabilité des actions.
- **Gestion des articles** : Ajout, mise à jour, suppression et consultation des articles. Les articles représentent les ressources, matériaux ou prestations nécessaires à la réalisation des projets. Un catalogue centralisé facilite la recherche, la comparaison et la sélection des articles, tout en permettant la gestion des stocks et des coûts associés.
- **Gestion des projets** : Suivi, planification et gestion des différents projets. Cette fonctionnalité permet de créer de nouveaux projets, d'y associer des utilisateurs, des articles et des budgets, et de suivre l'avancement des tâches. Des outils de reporting et de visualisation (tableaux de bord, graphiques) aident à piloter efficacement les projets et à anticiper les dérives budgétaires ou organisationnelles.

Chapter 3

Cahier des charges

3.1 Besoins fonctionnels

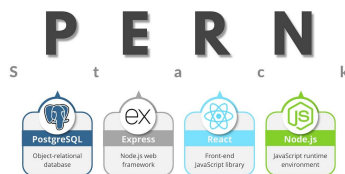
3.2 Besoins techniques

3.3 Contraintes

Chapter 4

Étude et choix technologiques

Présentation du choix de la stack **PERN** (PostgreSQL, Express, React, Node.js) et justification.



Justification du choix de la stack PERN

La stack **PERN** (PostgreSQL, Express, React, Node.js) a été choisie pour plusieurs raisons :

- **JavaScript full-stack** : Avec Node.js et React, le même langage est utilisé côté serveur et côté client, ce qui simplifie le développement et la maintenance.
- **Base de données relationnelle robuste** : PostgreSQL offre une gestion avancée des données, des relations complexes et une grande fiabilité pour les applications critiques.
- **Performance et scalabilité** : Node.js, grâce à son modèle non-bloquant et orienté événement, permet de gérer efficacement un grand nombre de requêtes simultanées.
- **Flexibilité et modularité** : Express.js facilite la création d'API RESTful flexibles, tandis que React permet de construire une interface utilisateur dynamique et réactive.

- **Écosystème riche :** Chaque composant de la stack dispose d'une communauté active et de nombreux packages, accélérant le développement et l'intégration de fonctionnalités avancées.

Chapter 5

Conception du projet

5.1 Architecture de l'application

Inclure un schéma de l'architecture PERN.

5.2 Diagrammes UML

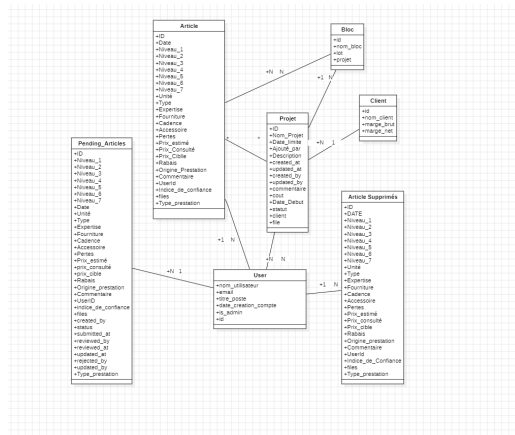


Figure 5.1: Enter Caption

5.3 Modèle de la base de données

Chapter 6

Implémentation

6.1 Backend (Node.js + Express)

6.2 Frontend (React)

6.3 Base de données (MongoDB)

6.4 Sécurité et authentification (JWT, hashage)

6.5 Tests et validation

Chapter 7

Déploiement

- 7.1 Environnement de développement
- 7.2 Environnement de production
- 7.3 Gestion de versions (Git/GitHub)
- 7.4 CI/CD (si applicable)

Chapter 8

Résultats obtenus

Inclure captures d'écran de l'application (ex. page de connexion, tableau de bord).

Chapter 9

Difficultés rencontrées et solutions

Lister les problèmes techniques et organisationnels rencontrés ainsi que leurs solutions.

Chapter 10

Perspectives d'amélioration

Améliorations techniques et fonctionnelles possibles.

Chapter 11

Conclusion

Bilan global du stage, apports professionnels et personnels.

Appendix A

Annexes

A.1 Extraits de code

A.2 Documentation API

A.3 Diagrammes complets

Bibliography

- [1] Documentation officielle MERN : <https://www.mongodb.com/mern-stack>
- [2] Documentation React : <https://react.dev>
- [3] Documentation Node.js : <https://nodejs.org>