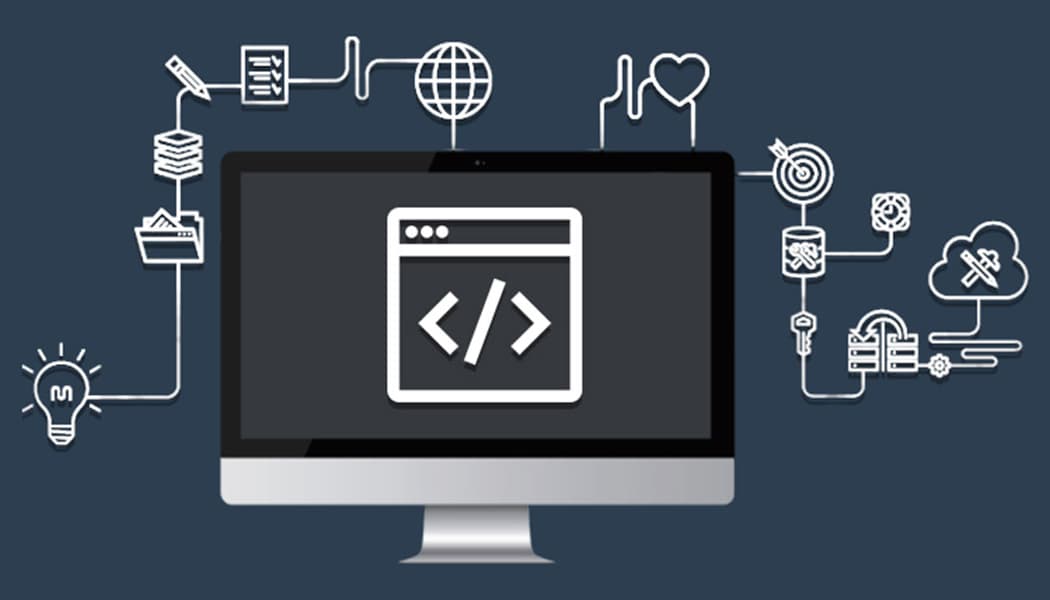
**P\_WEB295**



(Logo Backend)

Antoine Piguet  
Yosef Nademo  
Charles-Henri Moser  
 CID2B

Vennes - Lausanne

24 périodes

Chef de Projet – Gregory Charmier

Table des matières

[1 Spécifications 3](#_Toc192663871)

[1.1 Titre 3](#_Toc192663872)

[1.2 Description 3](#_Toc192663873)

[1.3 Matériel et logiciels à disposition 3](#_Toc192663874)

[1.4 Prérequis 3](#_Toc192663875)

[1.5 Cahier des charges 3](#_Toc192663876)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 3](#_Toc192663877)

[1.5.2 Fonctionnalités requises (techniques) 3](#_Toc192663878)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 4](#_Toc192663879)

[1.5.4 Contraintes 4](#_Toc192663880)

[1.5.5 Détail des vues 4](#_Toc192663881)

[1.5.6 Travail à réaliser par l'apprenti 5](#_Toc192663882)

[1.6 Les points suivants seront évalués 6](#_Toc192663883)

[1.7 Validation et conditions de réussite 6](#_Toc192663884)

[2 Planification Initiale 7](#_Toc192663885)

[3 Analyse 7](#_Toc192663886)

[3.1 Analyse API REST 7](#_Toc192663887)

[3.2 Conception de la base de données 8](#_Toc192663888)

[3.3 Conception des tests 9](#_Toc192663889)

[3.4 Planification détaillée 9](#_Toc192663890)

[4 Réalisation 9](#_Toc192663891)

[4.1 Gestion de l’authentification et des rôles 9](#_Toc192663892)

[4.2 Sécurité de l’application 9](#_Toc192663893)

[5 Tests 9](#_Toc192663894)

[5.1 Dossier des tests 9](#_Toc192663895)

[6 Conclusion 9](#_Toc192663896)

[6.1 Bilan des fonctionnalités demandées 9](#_Toc192663897)

[6.2 Bilan de la planification 10](#_Toc192663898)

[6.3 Bilan personnel 10](#_Toc192663899)

[7 Divers 10](#_Toc192663900)

[7.1 Journal de travail 10](#_Toc192663901)

[7.2 Bibliographie 10](#_Toc192663902)

[7.3 Webographie 10](#_Toc192663903)

[8 Annexes 10](#_Toc192663904)

# Spécifications

## Titre

Passion lecture

## Description

Réaliser le backend d’une application permettant de partager sa passion pour la lecture

## Matériel et logiciels à disposition

• Microsoft Windows 7

• VSCode

• Serveur local

• Navigateurs Web

• Accès à Internet

## Prérequis

Modules 106, 162, 164, 231, 293, 319, 320, 322 et 426

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

Ce projet vise à mettre en œuvre les connaissances apprises dans le module 295. Au final, l’application réalisée devra être exploitable et livrable. Dès lors, on attend un rendu professionnel et un soin particulier dans la documentation du projet.

### Fonctionnalités requises (techniques)

L’application backend consiste à la mise en place d’une API REST permettant à un consommateur de développer le frontend de l’application. Cette API REST doit être la plus complète possible avec toutes les notions vues en cours, à savoir :

• Un ensemble de routes permettant la gestion des livres, des catégories, des utilisateurs, etc

• Une validation de toutes les données fournies par le consommateur de l’API

• Une gestion des statuts http (200, 3xx, 4xx, 5xx) et des erreurs

• Une recherche sur les livres, catégories, etc

• Un système d’authentification basé sur les jetons JWT

• Une documentation Swagger la plus complète possible

• Des tests de votre API avec Insomnia ou Postman

• Des tests automatisés avec vitest

• Une intégration continue dans github actions

• Une « dockerisation » du backend

### Fonctionnalités requises (du point de vue de l’utilisateur)

Le frontend de l’application devra avoir :

• Une page d’accueil comprenant une explication de l’utilité du site ainsi que les cinq derniers ouvrages ajoutés (accès tout public).

• Une page comprenant la liste des ouvrages par catégorie (accès tout public avec restrictions sur les liens).

• Une page d’ajout d’un ouvrage (accès utilisateur).

• Une page de modification d’un ouvrage (accès utilisateur pour ses ouvrages)

• Une page de suppression d’un ouvrage (accès utilisateur pour ses ouvrages)

• Une page (vue détail d’un livre) permettant d’ajouter une appréciationet un commentaire à un ouvrage (accès utilisateur).

• L’utilisateur admin peut réaliser toutes les actions

PS : le pied-de page du site doit faire mention de la personne qui a créé l’application ainsi que le moyen de la contacter.

### Contraintes

* **Ecoconception Web** Votre application comprendra au moins un point d'écoconception Web. Il devra être présenté dans le rapport.
* La documentation est écrite en police Century Gothic et avec une taille de 11. Les titres des différentes parties sont écrits en police Century Gothic et avec une taille de 14. Les sous-titres sont écrits en police Century Gothic et avec une taille de 12. Les notes de bas de page seront utilisées à bon escient. Toutes les sources seront citées.
* Le nombre de page en () est le minimum à effectuer.

### Détail des vues

Ce site web permet de visualiser la liste des ouvrages stockés dans une base de données, de pouvoir en ajouter, modifier et en supprimer.

Un ouvrage est représenté par diverses informations, notamment :

• Un titre

• Une catégorie (bande dessinée, manga, roman, livre etc.)

• Un nombre de pages

• Un extrait (lien relatif vers un fichier pdf d’une page de l’ouvrage)

• Un résumé de l’ouvrage

• Les nom et prénom de l’écrivain qui est l’auteur de l’ouvrage

• Le nom de l’éditeur

• L’année d’édition

• La moyenne des appréciations des utilisateurs

• Les commentaires

• Une image de couverture

Un utilisateur est défini par un pseudo, une date d'entrée dans le site, du nombre d’ouvrages proposés et du nombre d'appréciations et de commentaires faits.

Une appréciation pour un ouvrage consiste à lui attribuer une note parmi la liste suivante : {0, 1, 2, 3, 4, et 5}

Vue liste

Sur la page de la liste des ouvrages, il doit être possible de visualiser le titre, l’auteur ainsi que le pseudo de la personne qui a posté l’ouvrage. Ceci doit être possible par catégorie.

Pour chaque ouvrage, il est possible en cliquant sur le titre ou le pseudo, de voir les détails de l’ouvrage ou de la personne.

Vue détail (accessible seulement pour les membres connectés ou les admin)

Pour l’ouvrage, il est possible de voir les informations, le nombre d'appréciations ainsi que la moyenne obtenue.

Pour la personne, il est possible de voir les informations, le nombre d’ouvrages proposés.

Sur la page de la liste des ouvrages, il doit être possible de visualiser le titre, l’auteur ainsi que le pseudo de la personne qui a posté l’ouvrage. Ceci doit être possible par catégorie.

### Travail à réaliser par l'apprenti

Les parties suivantes doivent figurer dans le rapport1 du projet :

• Introduction

Comprend une brève explication du projet (½ page)2

• Analyse

Planification de toutes les tâches à réaliser : Vous devez utiliser un outil permettant au chef de projet de voir l’évolution des tâches (github projet ou Trello).

Contiendra une analyse de l’API REST présentant pour chaque route :

Le verbe http, l’URI et si nécessaire le json ou autre information à transmettre (1 page).

Contiendra une analyse de de la base de données à réaliser (MCD, MLD, MPD) (1 page).

Contiendra une analyse de la structure du code qui sera effectuée (Organisation du code, …) (1 page)

Schéma de l’architecture représentant les interactions entre le frontend et le backend tout en situant les différents composants (API REST, la base de données, l’ORM, etc)

• Réalisation

Comprend une explication de l'algorithme utilisé pour gérer l’authentification et la gestion des rôles (1 page)

Comprend une explication des mesures prises pour les aspects de sécurité. (1 page)

Comprend une explication sur chaque fonctionnalité technique demandée.

• Test : Comprend une explication des tests réalisés (test automatique) (1 page)

• Conclusion : Comprend une explication de l’organisation du groupe quant à la gestion du code avec git/GitHub. Comprend une conclusion générale sur le projet (½ page). Comprend une conclusion personnelle sur le projet (½ page). Comprend une critique constructive sur la planification du projet (½ page).

• Webographie / Bibliographique / Glossaire :

Concernant la méthode de projet, elle devra être annexée au rapport et sera imposée par le chef de projet.

**Remarque :** Le recours à des outils en ligne d’intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d’inspiration à la réalisation. En cas d’abus, l’évaluation du projet en tiendra compte.

## Les points suivants seront évalués

* Le rapport
* Les planifications (initiale et détaillée)
* Le journal de travail
* Le code et les commentaires
* Les documentations de mise en œuvre et d’utilisation

## Validation et conditions de réussite

* + - Compréhension du travail
    - Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
    - Etat de fonctionnement du produit livré

# Planification Initiale

Le projet a été structuré selon un plan détaillé utilisant GitHub Projects pour suivre l'évolution des tâches. Les étapes principales incluent :

* Conception de l’architecture backend
* Développement des endpoints API REST
* Intégration de l’authentification JWT
* Validation des données et gestion des erreurs
* Tests unitaires et automatisés
* Documentation et intégration continue

# Analyse

## Analyse API REST

Ce paragraphe énumère la liste des difficultés potentielles de tout ordre :

Liste des compétences à acquérir ou approfondir

Liste du matériel à exploiter

Recherche d’informations particulières

Gestion du travail en équipe & collaboration

Ainsi que les solutions possibles

Si les spécifications de départ ne laissent pas de doutes sur la manière de réaliser un projet, ce chapitre ne fera que renvoyer le lecteur aux spécifications.

## Conception de la base de données

Le modèle relationnel de données a été conçu avec MySQL et Sequelize :

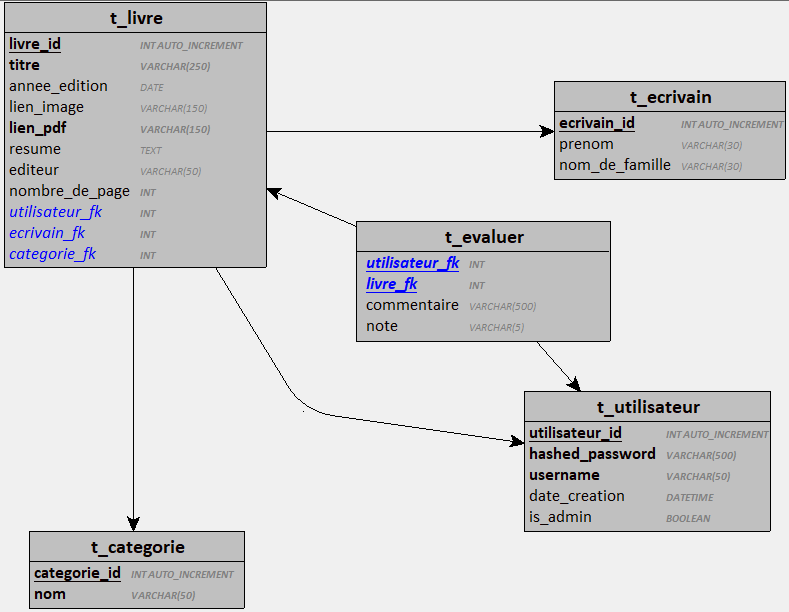
Tables principales : Utilisateur, Livre, Categories, Ecrivain

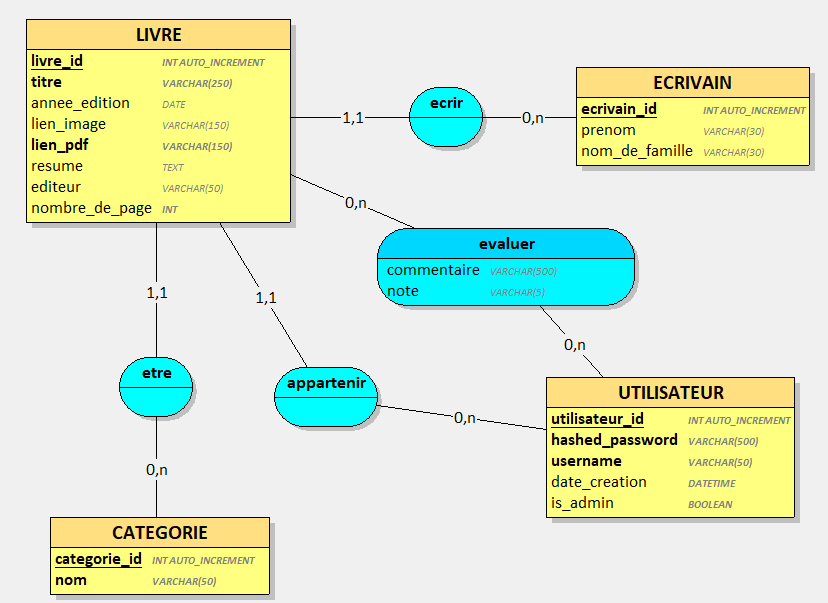
Relations :

Un utilisateur peut ajouter plusieurs livres

Un livre appartient à une catégorie

Un utilisateur peut noter et commenter plusieurs livres





## Planification détaillée

Le projet a été structuré à l’aide de **GitHub Projects**, permettant un suivi efficace des tâches et des jalons clés. Les grandes étapes sont les suivantes :

* Conception de l’architecture backend
* Développement des endpoints API REST
* Intégration de l’authentification JWT
* Validation des données et gestion des erreurs
* Tests unitaires et automatisés
* Documentation et intégration continue

# Réalisation

## Gestion de l’authentification et des rôles

L'authentification repose sur JWT (JSON Web Token) pour assurer une gestion sécurisée des accès utilisateur

## Sécurité de l’application

Les mesures de sécurité mises en place incluent :

* **Hashage des mots de passe** avec bcrypt
* **Validation des entrées utilisateur** avec Joi
* **Protection contre les injections SQL** via Sequelize ORM
* **Gestion des erreurs et statuts HTTP**

# Tests

## Dossier des tests

Les tests sont réalisés à l’aide de Insomnia pour les tests manuels.

# Conclusion

## Bilan des fonctionnalités demandées

Mon avons pus conclure la plupart des fonctionnalités demandée mais il manque les tests avec Vitest car nous n’avons pas eu le temps.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différence entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

## Bibliographie

## Webographie

Documentation Sequelize : <https://sequelize.org/docs/v6/>

Documentation Express.js : <https://expressjs.com/>

Swagger API Docs : <https://swagger.io/docs/>

Tutoriels JWT et sécurité des API : <https://jwt.io/introduction/>

# Annexes