

# RAPPORT

## Projet de Programmation Web Avancée

### Groupe G

SEBASTIAO-ESTEVEES Kévin

MAFIA Koffi Guy-Serge Christian

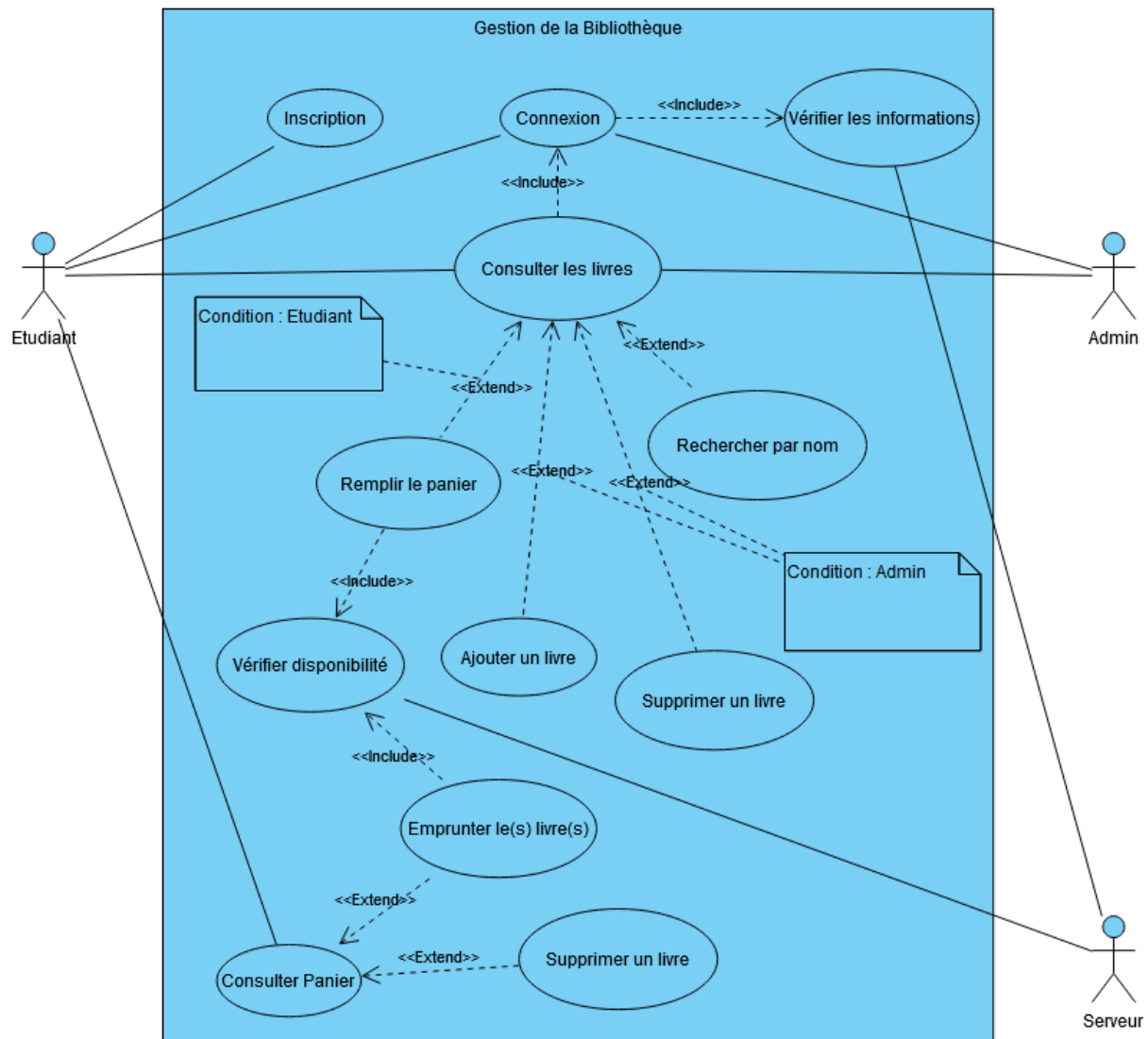
SAOUDI Aïda

PROVOST Nathan

Ce projet a pour objectif de créer une application web de gestion de la bibliothèque de l'EFREI en utilisant les langages de programmation web et les frameworks étudiés en cours. Le framework Vue.js a été utilisé pour le développement front end de l'application et Node.js pour la partie back end. Cette application a pour but de gérer les emprunts et les disponibilités des livres de la bibliothèque grâce à une base de données MySQL. La base de données et le client sont reliés via une API sur le serveur.

## I- Exigences fonctionnelles : liste des fonctionnalités pour chaque utilisateur

Deux types de comptes peuvent interagir avec l'application : les étudiants et les administrateurs. Les livres peuvent être ajoutés ou supprimés du catalogue par les comptes administrateurs. Les comptes étudiants peuvent emprunter des livres en fonction de leur disponibilité.



## II- Exigences non fonctionnelles : environnement de travail et frameworks, contraintes techniques, contraintes fonctionnelles.

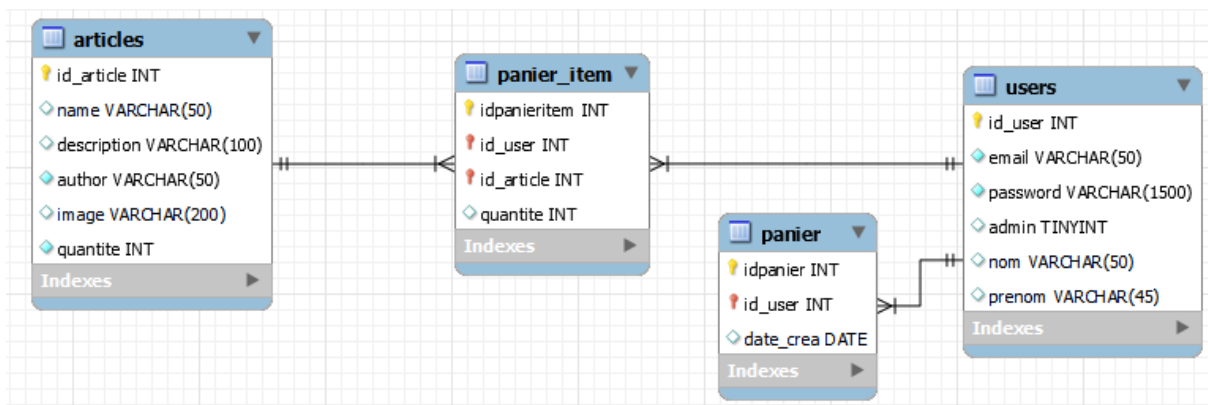
Ce projet a été réalisé avec les langages de programmation web HTML, CSS et JavaScript. Pour le côté client nous avons utilisé le framework Vue.js qui nous a permis d'afficher les différentes pages de notre application sur seul fichier HTML et selon les droits des utilisateurs grâce aux vues dynamiques.

Pour le côté serveur nous avons utilisé le framework Node.js et un serveur Express pour développer une api de gestion de l'application. Cette api fait aussi le lien avec la base de données. Nous avons utilisé le package axios pour faire transiter nos informations entre le serveur et le client et le package mysql pour faire les requêtes dans la base de données. Nous avons également utilisé le package bcrypt pour chiffrer les mots de passe et le package Json Web Token (JWT) pour protéger les données des utilisateurs.

Tout le projet a été réalisé en collaboration grâce à un dépôt en ligne sur GitHub où chacun a pu apporter sa contribution.

## III- Conception de la base de données (Modèle E/A)

La base de donnée est composée de quatre tables : users, panier, panier\_item et articles. La table users correspond aux logs et aux informations de chaque utilisateur. Elle permet l'authentification des comptes administrateurs et étudiants. La table panier permet la création d'une liste d'article correspondant à un utilisateur. Liste qui se retrouvera dans la page Panier. Cette liste renvoie à la table panier\_item qui contient tous les articles présents dans un panier ainsi que leur quantité. La dernière table est celles des articles disponibles, elle contient toutes les informations des articles dans le catalogue ainsi que leur quantité et est accessible à la modification par les comptes administrateurs.



## IV- Conception du projet

Pour réaliser ce projet nous sommes partis des fichiers du TP5 que nous avons modifiés et améliorés pour remplir le cahier des charges. Ce projet se compose de deux parties majeures : le client et le serveur. Celles-ci ont été développées en parallèles pour assurer la continuité des données avec la base de données. D'abord nous avons dû construire l'arborescence générale de l'application avec ses différentes pages. Puis nous nous sommes intéressés à chaque fonctionnalité une par une.

## V- Réalisation

Nous avons commencé par implémenter l'authentification de l'utilisateur qui est nécessaire pour accéder au service. Nous avons créé deux pages distinctes pour la connexion et l'inscription. La page d'inscription permet pour un étudiant d'enregistrer une nouvelle entrée dans la table users de notre base de données. Les administrateurs devaient être enregistrés par défaut dans la table. Nous avons ajouté une fonctionnalité permettant de créer un compte administrateur à l'aide d'un mot de passe. Les mots de passe sont chiffrés grâce à l'algorithme bcrypt avant d'être enregistrés. La page de connexion quant à elle permet de vérifier que les identifiants sont corrects grâce au système d'authentification JWT.

Ensuite nous avons créé la page du catalogue où nous pouvons consulter l'ensemble des livres disponibles ainsi que la page du panier. Ces deux pages ont été réalisées en parallèles dans un souci de cohérence de l'application. Nous avons d'abord développé la possibilité d'ajouter des livres via un formulaire et de les supprimer de la liste pour tous les utilisateurs pour ensuite limiter ces possibilités aux comptes administrateurs. Puis nous nous sommes penchés sur la fonction de recherche d'un livre dans le catalogue par son nom.

Pour finir nous avons synchroniser la quantité d'articles disponibles dans la base de données avec la quantité que l'étudiant peut placer dans le panier et commander.

## VI- Conclusion et perspective.

Dans la globalité ce projet est une réussite. Toutes les fonctionnalités ont pu être implémentées et sont fonctionnelles. Cependant certains défauts persistent et n'ont pu être corrigés. D'abord la base de données pourrait être améliorée pour être plus performante. Ensuite nous observons quelques problèmes d'actualisation lors de la navigation entre les pages, il faut parfois rafraîchir pour que les changements soient pris en compte. D'autre part, nous pourrions améliorer le design global du site avec une refonte complète des styles et de l'affichage des vues.