# PLAN DE REDACTION RAPPORT

## Introduction

Bienvenue dans notre projet d'application d'échange de livres, une initiative qui transcende les simples transactions littéraires pour créer une communauté dynamique de passionnés de lecture.

Notre application met à votre disposition une large sélection de livres, allant des romans contemporains aux classiques de la littérature, en passant par la science-fiction, la fantasy et la non-fiction et les ouvrages éducatifs.

Vous pouvez explorer les recommandations personnalisées, établir des connexions avec d’autres membres partageant vos centres d’intérêts littéraires, élargir votre horizon de lecture en découvrant de nouvelles œuvres captivantes, consulter les avis d’autres utilisateurs, créer votre propre liste et partager vos livres préférés avec une communauté passionnée de lecteurs.

Notre application vous permet de vivre une expérience interactive et conviviale, ou l’amour de la lecture est célébré et encouragé à chaque échange de livre.

Cette aventure répond à une problématique commune **: Comment créer un espace où chaque livre échangé devient une porte ouverte vers de nouvelles horizons, favorisant le partage d'expériences et l'enrichissement intellectuel au sein d'une communauté engagée ?**

Dans cette optique, notre approche se déploie en deux axes majeurs. Tout d'abord, nous explorons l'objectif central du projet, qui va bien au-delà de la simple logistique d'échange. Il s'agit de créer une plateforme où chaque interaction littéraire devient une invitation à la discussion, à l'exploration et à la construction de relations durables. En second lieu, nous abordons la population cible, invitant une communauté diversifiée de passionnés, à rejoindre cette aventure littéraire.

Rejoignez-nous pour une exploration collective où chaque page tournée devient une opportunité d'épanouissement collectif, et où l'amour des livres est le catalyseur de connexions profondes.

## Méthodologie

### Approche de développement

### Description des étapes du processus de conception et de développement

## Préconception

C’est la première etape de l’elaboration de notre projet

### Définition des besoins et des objectifs :

Nous avons constaté un manquement quant à l’accès aux différentes ressources que la plupart des gens sont confrontés à un probleme d’accés a certains livres notamment les editions limitées ou qui demandent un temps incalculables de recherches pour les trouvés :

* + Les étudiants et les lecteurs souhaitent échanger des livres sans tracas.
  + Les manuels scolaires sont souvent coûteux, ce qui peut être un fardeau pour les étudiants, La culture et l’éducation devraient être accessibles à tous.
  + Certaine personnes peuvent être confronté à des difficultés d’accès aux bibliothèques de manières récurrentes suite à une situations géographiques éloignés ou empêcher par des mauvaises conditions climatiques.
  + De nombreux livres sont jetés ou négligés, ce qui entraîne un gaspillage. Certains utilisateurs ont des livres dont ils n’ont plus besoin, tandis que d’autres cherchent des ouvrages spécifiques.

Dans le but de répondre aux besoins mentionnés ci-dessus l’application BiblioExchange a pour objectif de :

- **Faciliter l’échange de livres** : Créer une plateforme qui est simple d’utilisation et gratuite pour mettre en relation l’offre et la demande de livres et ainsi favoriserla solidarité et entraide entre les membres de la communauté en renforçant ainsi les liens sociaux,juste une application sans frais, basée sur le partage et la bienveillance.

**-Répondre aux difficultés d’accès aux bibliothèques pendant les intempéries** :

Offrir une solution numérique pour faciliter l’accès aux livres malgré les restrictions météorologiques et sociales.

**-Faciliter l’accès à la lecture** : Permettre à chacun d’accéder à une variété de livres sans barrières financières car l’éducation et la culture devrait être accessible a tous.

**-Réduire le gaspillage de livres** : Encourager la réutilisation et le partage de livres pour réduire l’empreinte écologique.

### Cahier de charge

### Cahier des charges

## Conception

### 4.1 Cycle de vie

#### 4.1.1 Définition

Dans le cycle de vie de notre projet, la conception représente une phase primordiale et déterminante pour produire un une application de haute qualité. C’est dans ce stade que nous devons clarifier la vue globale, en décrivant l’architecture générale que nous allons suivre dans la partie réalisation de notre projet.

Le cycle de vie d'une application comprend toutes les étapes depuis sa conception et sa réalisation jusqu’à sa mise en œuvre. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement du logiciel et la vérification de son processus de développement.

L'origine de ce découpage provient du constat que les erreurs ont un coût si élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs le plutôt possible.

#### 4.1.2 Les activités d’un cycle de vie

Le cycle de vie suivi pour réaliser une application, comprend généralement au minimum les activités suivantes :

- Spécification des besoins: elle consiste à définir la finalité du projet et son intégration dans une stratégie globale.

- Conception générale: dans cette activité, il s'agit de la préparation de l'architecture générale du logiciel.

- Conception détaillée: elle consiste à définir précisément chaque sous-ensemble du logiciel.

- Développement: (Implémentation ou programmation) il s’agit d’une traduction des fonctionnalités définies dans la phase de conception en langage de programmation.

- Tests unitaires: Ils permettent de vérifier individuellement que chaque sous-ensemble du l’application est implémenté conformément aux normes définies dans la conception.

- **Intégration**: dite aussi tests systèmes, elle consiste à vérifier que l’application correspond exactement au cahier des charges du projet en obtenant enfin un manuelle d’utilisation bien détaillé aux utilisateurs.

- Validation: c'est-à-dire la validation de conformité du site avec les buts spécifiés à la première étape du cycle de vie.

### 4.1.3 Modèle de cycle de vie en V

Afin de concevoir et développer application, nous avons opté pour le modèle de cycle de vie en V. Ce choix reviens au fait que ce cycle est le plus efficace avec son principe de travail qui nécessite la vérification de chaque étape et la possibilité de corriger les fautes avant de se lancer vers l’étape suivante.



***Figure1***

Le modèle de cycle de vie en V a été imaginé pour pallier le problème de réactivité du modèle en cascade. Ce modèle est une amélioration du modèle en cascade. Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel.

### 4.3 L’architecture logicielle MVC (Modèle - Vue – Contrôleur): Achitecture MVC (modèle-vue-contrôleur) : Cette architecture a pas mal d’avantages pour qu’elle reste toujours la plus utilisée dans le monde du développement Web étant donnée qu’elle se caractérise par :

- L’allégement du poste de travail.

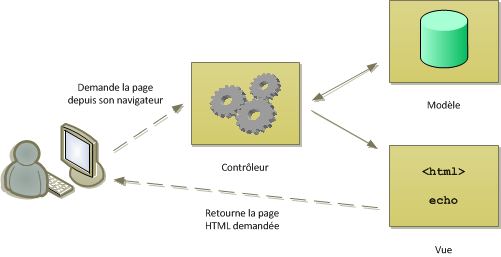
- La prise en compte de l'hétérogénéité des plates-formes (serveurs, clients, langages, etc.).

- Une meilleure répartition de la charge entre les différents entités clients et serveurs.

La figure suivante schématise le rôle de chacun de ses éléments :



L’illustration suivnante permet de comprendre la logique de cette architecture :



## 4.2 Conception détaillée

Dans cette partie, nous utiliserons la modélisation UML pour représenter la spécification des exigences grâce au diagramme de cas d’utilisation, analyser le domaine avec le diagramme de classe, mais aussi représenter graphiquement les interactions entre les acteurs et le système avec le diagramme de séquence. Par la suite, nous abordons la conception, d’un point de vue fonctionnel, technique et graphique.

1. **Diagrammes de Classes**

Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Ainsi dans cette partie, nous étudierons les entités statiques du système. Ceci est illustré par le diagramme de classes suivant :

**Utilisateur** :

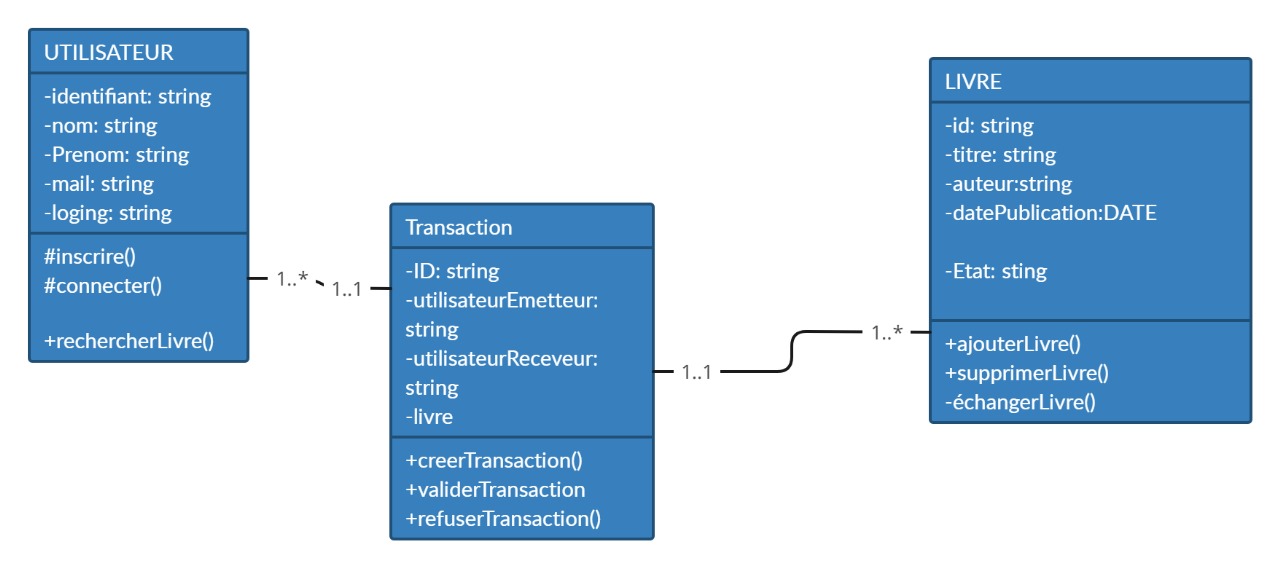
* + Attributs : ID, nom, adresse e-mail, mot de passe
  + Méthodes : seconnecter(), s’inscrire(), rechercherLivre()

**Livre** :

* + Attributs : ID, titre, auteur, année de publication, état (disponible, échangé, etc.)
  + Méthodes : ajouterLivre(), supprimerLivre(), échangerLivre()

**Transaction** :

* + Attributs : ID, utilisateurEmetteur, utilisateurReceveur, livre
  + Méthodes : créerTransaction(), validerTransaction()

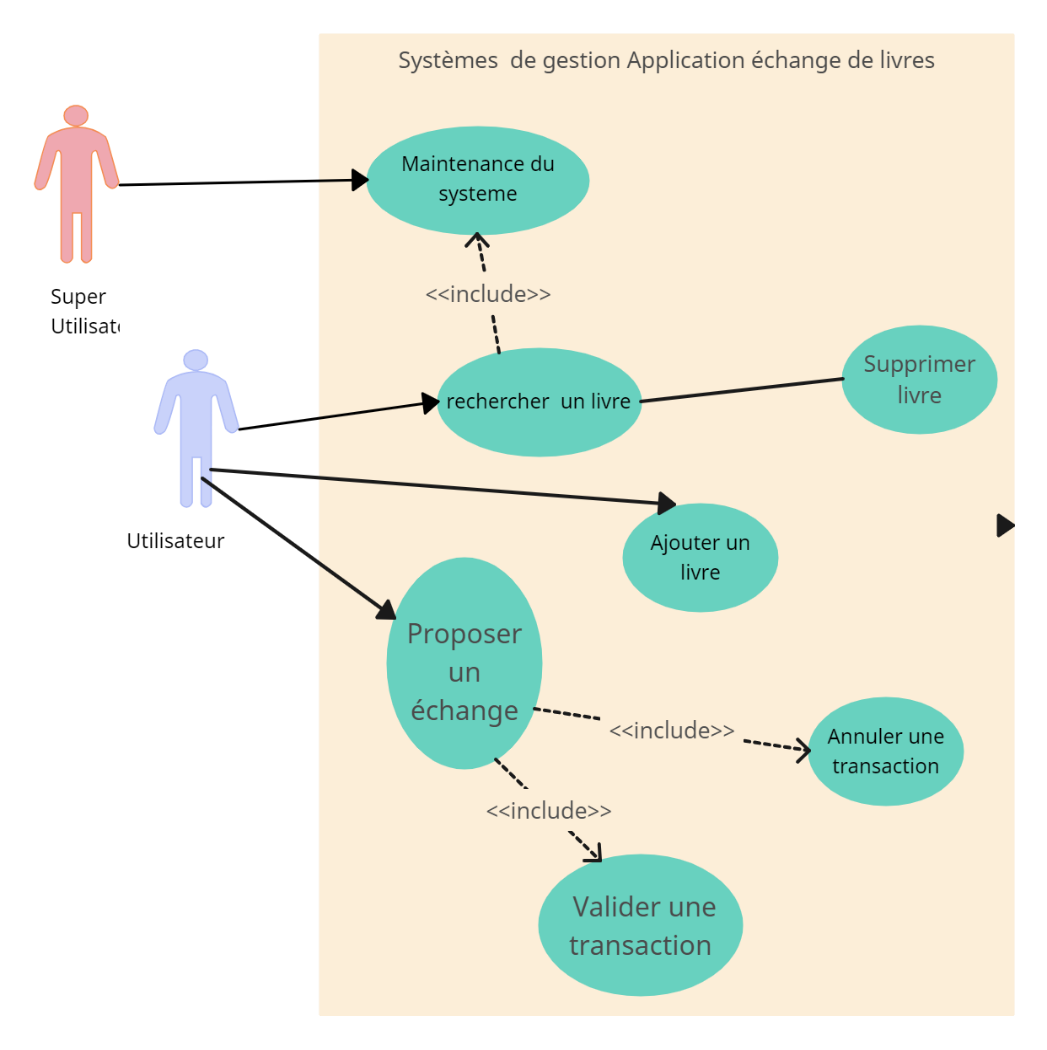
****

### Diagramme de Cas d'Utilisation

Les rôles des diagrammes de cas d’utilisation sont de recueillir, d’analyser et d’organiser les besoins, ainsi que de recenser les grandes fonctionnalités d’un système. Il s’agit donc de la première étape UML pour la conception d’un système.

Ainsi nous distinguons 2 types d’acteurs :

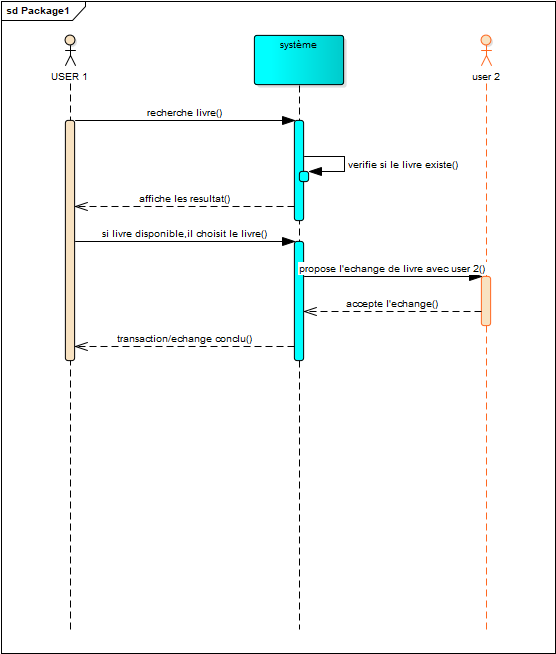
* + **Acteurs** :
    - Utilisateur
    - Super Utilisateurs
  + **Cas d’utilisation** :
    - Rechercher un livre
    - Ajouter un livre
    - Supprimer un livre
    - Proposer un échange



Pour chaque cas d’utilisation, on trouvera les différentes interactions entre les utilisateurs et le système, les scénarios des cas d’utilisation ainsi que les alternatives ou erreurs que peut rencontrer le système, dans le but de connaître les différentes étapes à développer pour chaque fonctionnalité.

### Diagrammes de Séquence :

Les diagrammes de séquences permettent de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

* + **Scénario : Rechercher un livre**
    - L’utilisateur entre des mots-clés dans la barre de recherche.
    - Le système recherche les livres correspondants dans la base de données.
    - Le système affiche les résultats à l’utilisateur.
  + **Scénario : Proposer un échange**
    - L’utilisateur sélectionne un livre qu’il souhaite échanger.
    - Le système vérifie si le livre est disponible.
    - L’utilisateur choisit un autre livre proposé par un autre utilisateur.
    - Le système crée une transaction entre les deux utilisateurs.
  + **Scénario : Valider une transaction**
    - L’utilisateur confirme l’échange.
    - ****Le système met à jour l’état des livres dans la base de données.

### Conception de l'interface utilisateur (UI/UX)

Une interface définit la frontière de communication entre deux entités, comme les éléments de logiciel, des composants de matériel informatique, ou un utilisateur.

Elle se réfère généralement à une image abstraite qu’une entité fournit d’elle-même à l’extérieur Cela permet de distinguer les méthodes de communication avec l'extérieur et les opérations internes, et autorise à modifier les opérations internes sans affecter la façon dont les entités externes interagissent avec elle, en même temps qu'elle en fournit des abstractions multiples

Une [interface humain-machine](https://fr.wikipedia.org/wiki/Interface_humain-machine) permet des échanges entre un humain et une machine ;

Une [interface de programmation](https://fr.wikipedia.org/wiki/Interface_de_programmation) permet des échanges entre plusieurs [logiciels](https://fr.wikipedia.org/wiki/Logiciel) ;

L’interface qui est présentée à l’utilisateur est nommée [interface utilisateur](https://fr.wikipedia.org/wiki/Interface_utilisateur) ;

Les interfaces entre des composants de [matériel informatique](https://fr.wikipedia.org/wiki/Mat%C3%A9riel_informatique) sont des interfaces physiques.

**IHM : Interface Homme Machine**

1. **Page d’Accueil** :
   * Affiche les livres disponibles pour l’échange.
   * Barre de recherche pour rechercher des livres spécifiques.
   * Bouton pour accéder au profil de l’utilisateur connecté.
2. **Liste des Livres** :
   * Affiche les livres disponibles avec leurs détails (titre, auteur, année de publication, état).
   * Bouton pour ajouter un livre à la liste.
3. **Détails du Livre** :
   * Affiche les informations détaillées d’un livre sélectionné.
   * Bouton pour proposer un échange avec ce livre.
4. **Profil Utilisateur** :
   * Affiche les informations sur l’utilisateur (nom, adresse e-mail, historique des transactions).
   * Bouton pour ajouter ou supprimer des livres de la liste de l’utilisateur.
5. **Page de Proposition d’Échange** :
   * Permet à l’utilisateur de sélectionner un livre qu’il souhaite échanger.
   * Affiche les livres disponibles pour l’échange.
   * Bouton pour valider la proposition d’échange.
6. **Notifications** :
   * Affiche les notifications pour les nouvelles propositions d’échange, les confirmations d’échange, etc.
7. **Barre de Navigation** :
   * Permet à l’utilisateur de naviguer entre les différentes sections de l’application (accueil, liste des livres, profil, etc.).

## Développement

Dans cette partie purement technique, nous allons aborder la présentation des interfaces de notre application, ainsi que le déroulement des actions à travers les acteurs afin de concrétiser notre travail.

6.INTERFACE :

### Technologies utilisées

## Les langages utilisés :

* HTML :

L'HyperText Markup Language, **HTML**, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web.

* CSS :

CSS est le langage que nous utilisons pour styliser un document HTML. Il décrit comment les éléments HTML doivent être affichés.

* PHP :

L’Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

* SQL :



SQL est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

* XML :

L'Extensible Markup Language, généralement appelé XML, « langage de balisage extensible » en français, est un métalangage informatique de balisage générique qui est un sous-ensemble du Standard Generalized Markup Language.

* Javascript :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web.

UML :

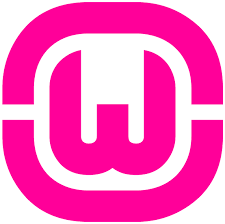
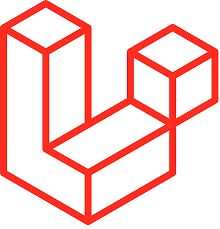
## Les outils de travail :

**ENTREPISE ARCHITECT :**



Enterprise Architect est un outil graphique multi-utilisateurs conçu pour aider à créer des systèmes robustes et maintenables. Et à l’aide de rapports et de documentations intégrés de haute qualité, vous pouvez fournir une vision véritablement partagée facilement et avec précision !

**WampServer :**

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs, un interpréteur de script, ainsi que phpMyAdmin pour l'administration Web des bases MySQL.Laravel est un framework web open-source écrit en PHP respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur GitHub.

**Sublime Text** est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

**VISUAL STUDIO CODE :**



Visual Studio Code est un éditeur de code source qui peut être utilise avec une variété de langage de programmation. Il est basé selon le model « Electron » utiliser pour développer des applications web. L’une des particularités les plus importantes de VS Code est sa capacité de débogage ; en effet il a été conçu de manière à déboguer des applications directement sur l’éditeur sans avoir recours aux navigateurs, grâce à un système de points d’arrêt et une console de débogage intégrée permettant de résoudre des problèmes via un terminal intégré directement dans l’éditeur de texte.

**BALSAMIQ : Balsamiq** est un **logiciel de conception de wireframes** qui permet aux équipes de créer des maquettes et des prototypes interactifs, ainsi que de réaliser des tests utilisateurs. Cet outil est destiné à tous ceux qui ont besoin de créer des wireframes, et son utilisation ne requiert pas d’expérience particulière dans le webdesign.

### Déroulement du développement

### Tests et validation

## Résultats et Discussion

### Évaluation de l'application

### Analyse des retours utilisateurs

## Conclusion

## Annexes

### Diagrammes, wireframes, maquettes

### Liste des tableaux