

**Documentation Technique AP WEB**

**Un livre près de chez vous**

unlivrepresdechezvous.ipssi-sio.fr - Daval Enzo et Madani Yanis

Créé en 2022

1. Arborescence

Cette partie a pour but de montrer l’arborescence de l’AP web « Un livre près de chez vous » ainsi que l’explication de chaque partie de celui-ci.

Une image contenant texte

Description générée automatiquement

Ce projet est composé à la racine de 3 dossiers ainsi que de 4 fichiers.

* Le dossier « membres » contient toutes les pages web sur lesquelles les différentes personnes peuvent accéder, selon leur rôle (administrateurs, visiteurs et membres) ainsi que les fichiers pour le frontend.
  + Le dossier « Categorie » contient toutes les images pour les genres de livres.
  + Le dossier « css » contient tout le css du site.
  + Le dossier « image » sert à stocker les images de profil ainsi que les couvertures des livres.
  + Le dossier « livres » contient les fichiers pdf de tous les livres libres de droits pouvant être lus directement sur le site.
  + Le dossier « script » contient le javascript du site.
  + Le dossier « webfonts » fait partie de la librairie Font Awesome.+

Une image contenant texte, équipement électronique, capture d’écran

Description générée automatiquement

* Le dossier modèle contient toutes les classes et méthodes chargées d’assurer la communication entre le client et la base de données.
* La classe Adresse répertorie les méthodes utilisées pour les adresses des utilisateurs. Cela permet de savoir où doivent être livrées les commandes.
* La classe Auteur répertorie les requêtes ayant un rapport aux auteurs des livres
* La classe Bibliothèque est la classe principale en ce qui concerne les livres. Elle récupère les infos des livres, leurs auteurs ainsi que leur genre.
* Comme son nom l’indique, les commandes sont les méthodes liées aux commandes des utilisateurs.
* Les Commentaires sont les avis liés aux livres.
* La classe Editeur contient les méthodes pour éditeurs des livres.
* La classe Genre concerne les genres de livre (Argumentatif, Narratif…).
* La classe Lectures sert pour l’affichage de livres libres de droit visionnables directement sur le site.
* La classe Livre récupère les livres et ses informations.
* La classe modèle sert à faire la connexion à la BDD et à lancer des requêtes dans les différents modèles.
* La classe Panier sert à tout ce qui concerne le panier des utilisateurs
* La classe stockage est une classe enfante de Panier permettant de récupérer les livres et leur quantité dans le panier.
* Type Genre concerne les sous genre des livres. (Roman, Fantasy…)
* La classe Utilisateur est l’ensemble des méthodes concernant les utilisateurs du site.
* Le dossier tests sont les tests unitaires réalisés avec le Framework PhpUnit. Il nous a permis de vérifier que toutes les méthodes de chaque modèle était fonctionnel en plus de vérifier certaines failles possibles.
* Le dossier Vendor contient toutes les ressources nécessaires au fonctionnement des différents Frameworks, il faut éviter d’y toucher afin de garder les Frameworks fonctionnels.
* Le dossier traitement contient tous les contrôleurs permettant de faire les vérifications de formulaires en backend ainsi que l’appel des méthodes permettant l’interaction entre le client et la base de données. Ils permettent l’affichage d’une donnée précise (avec un GET) ou l’envoi de données (méthode POST) dans une requête http.
* Le fichier .gitignore permet d’empêcher l’envoi sur GitHub de fichiers ou de dossiers que l’on ne veut pas envoyer au projet en ligne. Tout ce qui est mentionné dans le fichier reste donc localement dans le dossier du projet sur le pc.
* Le fichier .htaccess est un fichier texte qui permet de modifier certains paramètres du serveur apache. Ces modifications sont valables pour le dossier dans lequel le fichier se trouve ainsi que ses sous-dossiers. On utilise ce fichier principalement pour :
  + Contrôler l'accès à un répertoire ou un fichier par mot de passe.
  + Gérer le type mime des fichiers.
  + Définir des pages d'erreurs personnalisées.
  + Réécrire les URL.

Dans notre cas, nous l’utilisons afin d’empêcher des personnes malintentionnées d’accéder aux dossiers des contrôleurs ainsi que celui des modèles directement via l’url.

* Le fichier composer.json est le fichier contenant toutes les dépendances liées à Composer. Il liste tous les frameworks et librairies installés avec le gestionnaire de dépendance Composer.
* Le fichier composer.lock sert lui à lister toutes les versions des différentes dépendances du projet. Cela permet par exemple, de mettre à jour des dépendances et de garder la même version des dépendances que les autres personnes travaillant sur le projet.
* Index.php est un fichier php servant à la redirection vers la page membres/index.php. Cela permet à l’hébergeur (ici OVH) d’avoir lorsque l’on rentre l’url « unlivrepresdechezvous.ipssi-sio.fr » directement l’index.
* Le readme.md contient des informations supplémentaires concernant le projet, il est visible directement sur la page du projet sur GitHub si le projet en possède un.

1. Tableaux des risques

Ci-dessous pourront être retrouvés les tableaux des User stories, des événements redoutés ainsi que le tableau des risques.

Une image contenant table

Description générée automatiquement

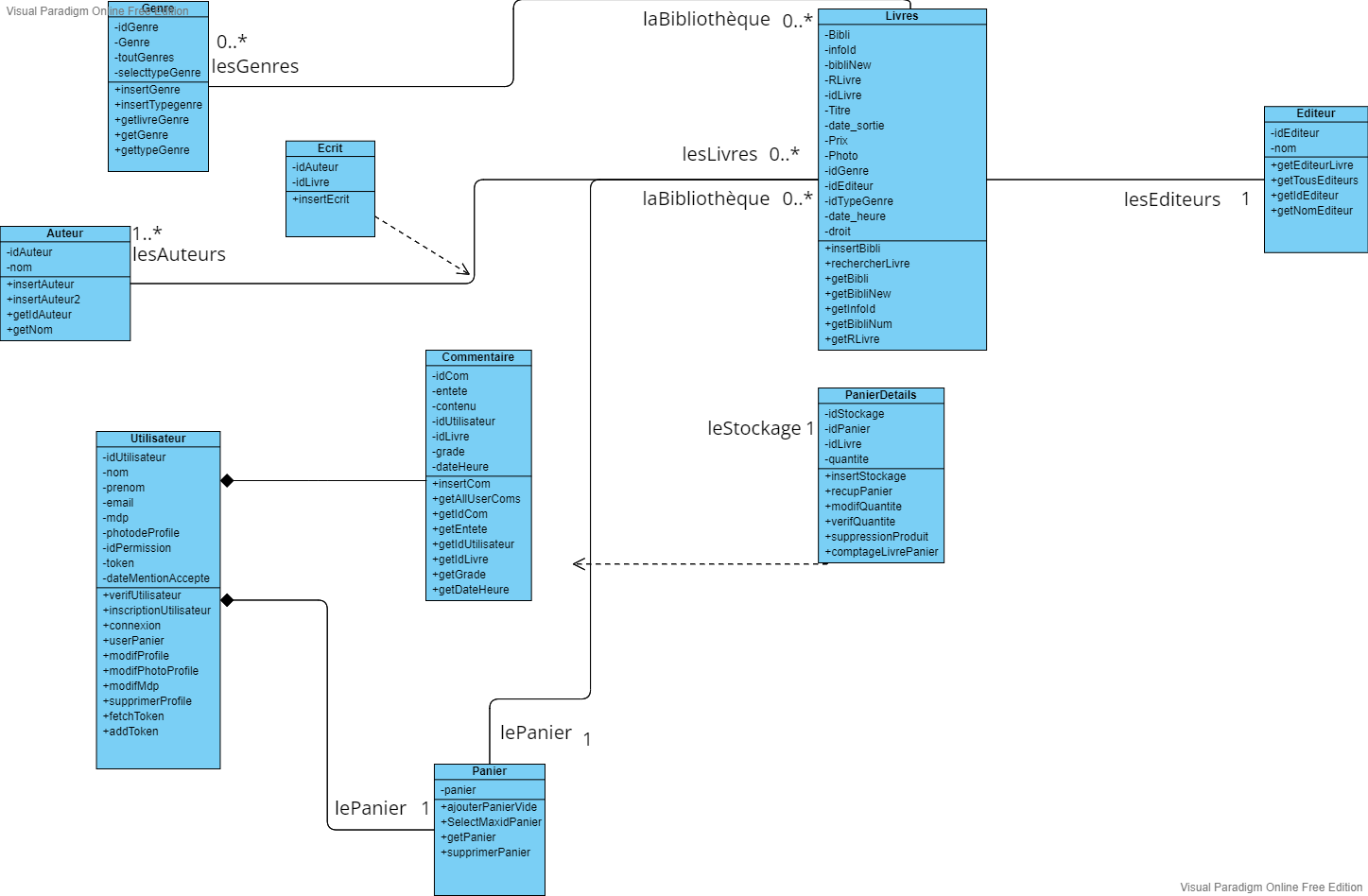
Une image contenant table

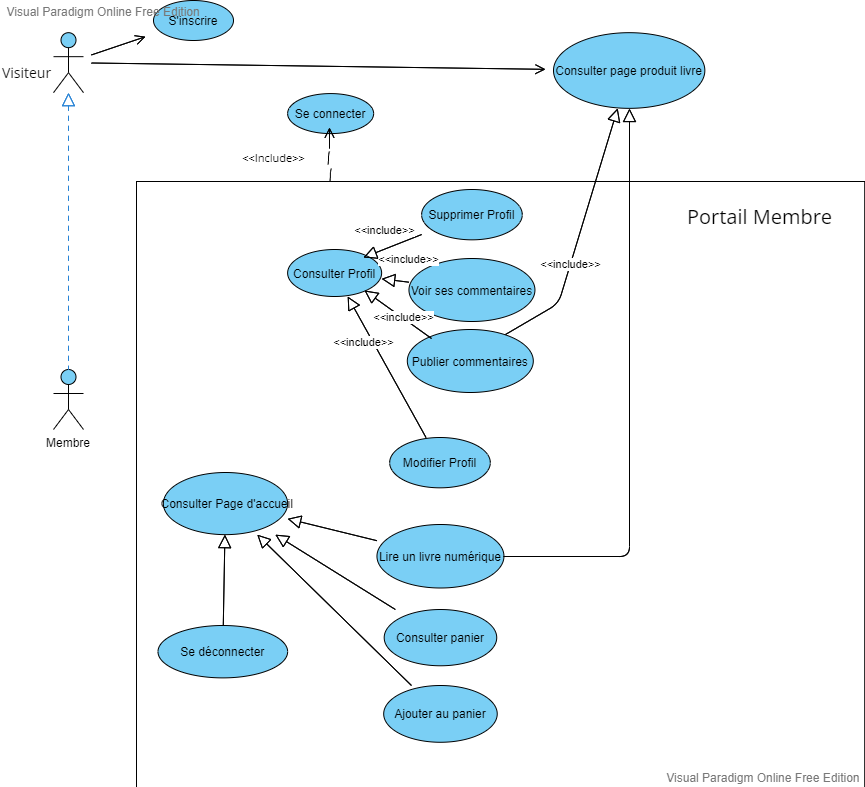
Description générée automatiquement

Une image contenant texte

Description générée automatiquement

1. Diagramme de UML et cas d’utilisation





1. Procédures stockées et déclencheurs

* insert\_livre\_with\_auteur : Insérer dans la table d’association le livre et l’auteur
* select\_allComs\_user : Sélectionner tous les commentaires de l’utilisateur
* select\_commentaires\_by\_user\_and\_livre : Sélectionner tous les commentaires d’un utilisateur sur un livre
* select\_commentaires\_non\_approuve : Sélectionner tous les commentaires en cous d’approbation
* select\_genre\_with\_livre\_and\_auteur : Sélectionner un livre, son genre et son auteur
* select\_panier : Sélectionner le panier de l’utilisateur et son contenu
* verif\_commentaire : Nombre de commentaires en cours d’approbation
* delete\_detailcommandes\_after\_delete\_commande : Les lignes de commande liées à une commande sont supprimée en même temps que la commande. Cela permet de ne pas avoir à faire une requête DELETE en code.
* delete\_tentatives\_connexion\_after\_insert\_access\_logs : Si la personne a réussi à se connecter (si une personne ne réussit pas à se connecter 3 fois, le captcha dans la page de connexion apparaît). Ses tentatives de reconnexions sont réinitialisées.

1. Librairies et frameworks utilisées

Bootstrap 5 : <https://getbootstrap.com/> (Librairie pour la partie css, permet la mise en forme du site)

AOS : <https://michalsnik.github.io/aos/> (utilisée pour faire des translations et fondus)

Jquery : https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js (Framework Javascript)

Slick Caroussel : <https://cdnjs.cloudflare.com/ajax/libs/slick-carousel/1.8.1/slick.css> (Utilisé pour faire le carrousel de la page d’accueil du site)

Font Awesome 5.13.3 : <https://fontawesome.com> (Librairies d’icônes)

PhpUnit : <https://phpunit.de> (Test unitaires)

1. Gestion du trafic de données



Google Analytics est utilisé afin de pouvoir analyser l’audience et le trafic sur le site. En insérant un tag avec un id donné par Analytics dans le header, il est possible de récolter des données concernant le nombre d’utilisateurs venant sur le site.

Code entré pour lier à Google Analytics :

Une image contenant texte

Description générée automatiquement

Une image contenant texte, capture d’écran, intérieur

Description générée automatiquementGoogle Analytics permet de voir le nombre de personnes ayant accédé à notre site dans un temps donné. On peut voir le temps passé sur celui-ci, le pays dans lequel la personne a eu accès.

Comme il est possible de le voir sur la sidebar, Google Analytics possède aussi d’autres fonctionnalités telle que l’analyse des données sur la publicité du site ainsi que le suivi de monétisation

1. Gestion des tickets

Afin de fournir une assistance en cas de panne de l’appli web ou C#, nous avons installé GLPI afin de pouvoir regrouper et traiter les demandes. Que ce soient des tickets pour des résolutions de problèmes ou des évolutions de services, GLPI nous permet d’être à l’affut de toutes demandes de la part du client.

