

Table des matières

Inti	oduction	3
I.	Conception	4
Act	eurs de l'application	4
Dia	gramme de cas utilisation	5
D	escription textuelle	5
S	cénarios	6
II.	Base de Données	7
Mo	dèle Conceptuel de Données (MCD)	7
D	ictionnaire des données correspondant au MCD :	7
Inse	ertions	11
Les	Déclencheurs	11
III.	Maquettage	12
Ma	quettes	12
IV.	JDBC	14
Fon	ctions	14
V.	Web	15
PH	P	15
Con	trôleur	15
Mét	:hode	16
Mo	deledele	17
CSS	5	18
VI.	Conclusion	10

Introduction

Nous sommes un groupe de développeurs sollicités par la médiathèque de notre université afin de concevoir une base de données de celle-ci, afin d'aider à la gestion des livres qu'elle possède.

À la suite de cela, nous devons réaliser une interface web permettant aux utilisateurs de pouvoir emprunter des ouvrages à la médiathèque, mais aussi de les rendre ; pour ce faire, l'utilisateur doit d'abord s'inscrire, puis s'authentifier (sous autorisation de la documentaliste).

L'administrateur doit quant à lui s'occuper de la gestion de la base de données, c'est-à-dire ajouter, supprimer ou modifier des ouvrages. Il possède quelques droits pour effectuer ces modifications, mais pas tous.

La base de données doit aussi permettre à la documentaliste de pouvoir faire les comptes, c'està-dire gérer les emprunts et les remises des différents utilisateurs. La documentaliste va donc gérer les autorisations chez les utilisateurs lors de l'inscription, emprunt et rendu du livre. Elle va également appliquer des sanctions si ces derniers sont rendus en retard.

La documentaliste a également comme rôle de gérer les stocks c'est-à-dire vérifier le nombre d'ouvrages restants et les réceptions de nouveaux livres.

Enfin, il y a les sous-traitants d'une petite entreprise externe, qui permettent d'assurer le fonctionnement quotidien du site dans le cas où ce dernier aurait des pannes ou autres. Les sous-traitants ont donc tous les droits car ils doivent posséder toutes les compétences requises pour résoudre les différents problèmes.

ı. <u>Conception</u>

<u>Acteurs de l'application</u>

Utilisateurs:

Ce sont les individus qui peuvent emprunter les livres, ils sont classés en plusieurs catégories (employé, professeur, étudiant). Ils interagissent avec l'interface web qui leur est mis à disposition. Ils peuvent emprunter et rendre des livres. Pour effectuer ces actions l'utilisateur doit s'authentifier au préalable.

Le documentaliste :

C'est la personne chargée de scanner les livres, et de les mettre à disposition des utilisateurs. C'est la documentaliste qui s'occupe de gérer les emprunts, les remises et les retards des différents utilisateurs. L'utilisateur est notifié par mail le jour de la date de rendu qu'il doit rendre le livre, il est également prévenu la semaine d'avant, et est notifié chaque fin de semaine lorsqu'il ne l'a pas rendu.

La documentaliste doit donc évidemment autoriser un utilisateur à emprunter les ouvrages, et c'est également elle qui va gérer et autoriser l'inscription d'un utilisateur. Elle s'occupe aussi de gérer les stocks en listant les ouvrages restants de chaque catégorie et également en s'occupant des réceptions de nouveaux livres.

Administrateurs:

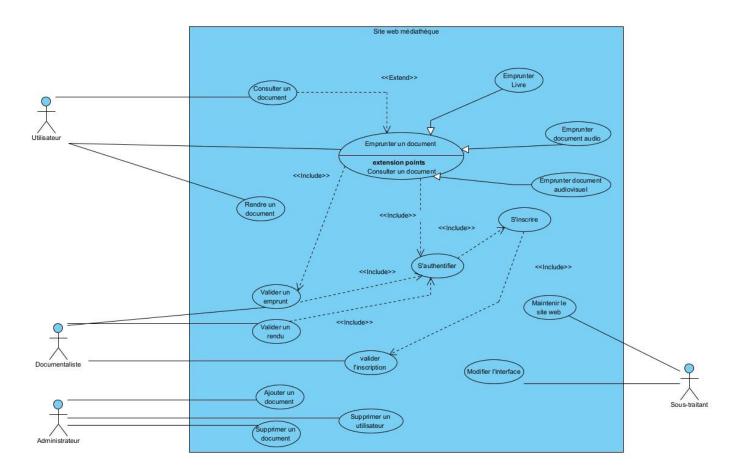
S'occupe de gérer la base de données, c'est-à-dire ajouter ou supprimer des livres. L'administrateur possède tous les droits afin d'effectuer les modifications (supprimer ou ajouter les tables, les utilisateurs), tout cela est historisé. C'est également l'administrateur qui s'occupe du délai des retards des différents utilisateurs, auquel cas il met une sanction.

Lorsqu'un utilisateur rend un livre en retard, il a une sanction et ne peut pas emprunter de nouveaux livres pendant le nombre de jours de retard du livre. C'est-à-dire que, s'il rend un livre 10j après sa date limite, il ne pourra pas emprunter de nouveaux livres pendant 10 jours. S'il a un autre livre chez lui qu'il n'a toujours pas rendu et qu'il le rend par exemple après 12 jours, le nombre de jour ne s'additionne pas, mais c'est la date de retard la plus grande qui l'emporte, c'est-à-dire qu'il ne pourra pas emprunter de livres pendant 2 jours supplémentaires, soit pendant 12 jours.

Sous-traitants:

S'occupe de la maintenance du site web, il permet d'assurer le bon fonctionnement du site dans le cas où ce dernier aurait des défaillances. Il va avoir tous les droits, de ce fait, cette personne doit avoir toutes les compétences informatiques requises. Nous ferons donc appel dans ce cas à une petite entreprise externe ayant des ingénieurs compétents dans chaque domaine.

Diagramme de cas utilisation



Description textuelle

Rendre/Emprunter un document : Ces deux cas représentent l'action d'emprunter un document dans le site web de la médiathèque et ensuite le remettre sur place à la médiathèque. Pour que l'emprunt s'effectue il faut une validation de la part de la documentaliste d'où l'ajout d'une relation d'inclusion dans le diagramme cas d'utilisation.

<u>Valider un emprunt/rendu</u>: Le documentaliste doit pouvoir indiquer si les emprunts ont été rendus, ou si ceux-ci ont été empruntés et donc s'ils sont disponibles.

<u>Valider l'inscription</u>: Une inscription nécessite une validation de la part de la documentaliste, c'est pour cela que nous avons mis une relation d'inclusion.

<u>Supprimer</u>: L'administrateur à le droit de supprimer de la base de données. Seul l'administrateur à ce droit. (L'utilisateur est ajouté lors de son inscription.) <u>Ajouter/Supprimer un document</u>: L'administrateur ajoute les livres dans la base de données et supprime ceux qui ne sont plus présents.

<u>Maintenir le site web</u>: Cette action est réservée au technicien, lorsque le site devient inaccessible ou défaillant, il a donc le rôle d'assurer son bon fonctionnement en résolvant les erreurs.

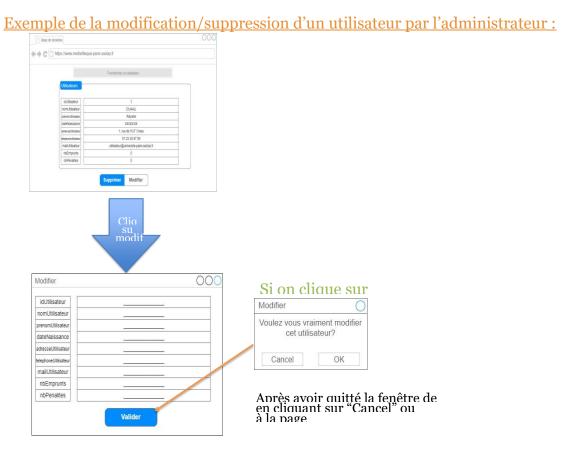
<u>Modifier l'interface</u>: Le technicien peut également modifier l'interface web vue de là par de l'utilisateur pour la simplifier par exemple.

Scénarios

Tout d'abord, l'utilisateur doit s'inscrire sur le site web de la médiathèque en inscrivant son nom, prénom et adresse et email, cette inscription doit ensuite être validée par le documentaliste. Une fois que le documentaliste a validé l'inscription, l'utilisateur est enregistré et peut alors accéder au site.

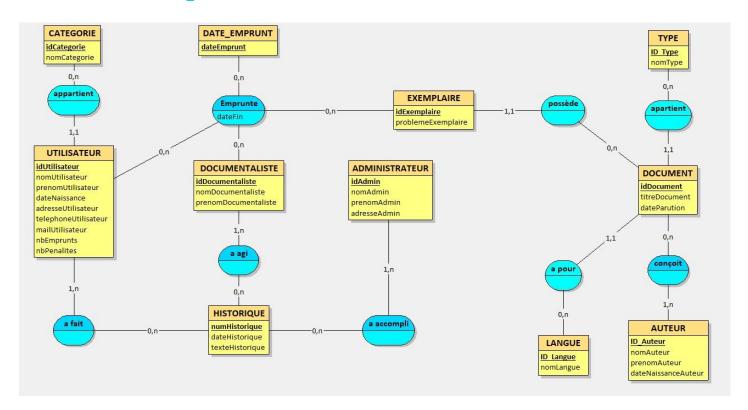
L'administrateur a accès à la base de données et peut supprimer un utilisateur en cas de besoin. L'administrateur a aussi la possibilité d'ajouter ou de supprimer des documents dans la base de données avec une insertion ou une suppression dans la base.

L'utilisateur peut rechercher un document souhaité et à la possibilité de le consulter en cliquant sur le bouton consulter. Suite à cette consultation, l'utilisateur a le choix d'emprunter le document ou non. Si oui, l'utilisateur clique sur le bouton d'emprunt et le documentaliste doit valider l'emprunt. Une fois le document rendu par l'utilisateur sur place, le documentaliste valide le rendu.



II. <u>Base de Données</u>

Modèle Conceptuel de Données (MCD)



Nous souhaitons avec la table Historique enregistrer les actions de l'administrateur et du documentaliste sur les Utilisateurs, en enregistrant la date de la modification et le texte résumant la modification.

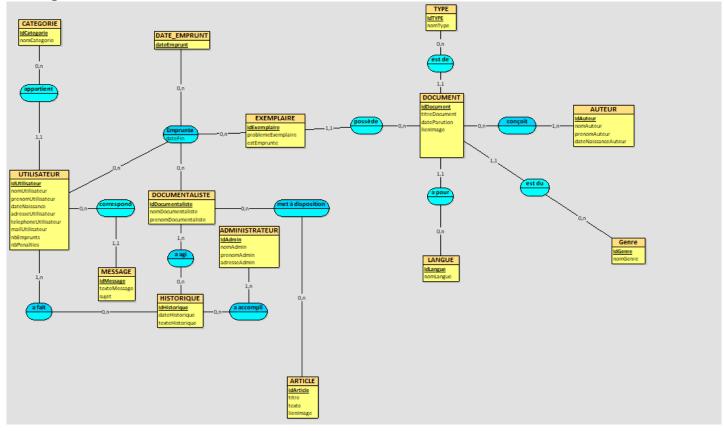
Dictionnaire des données correspondant au MCD :

	Libellé	Désignation	Туре	Nature	Longueur
A	idUtilisateur	Identifiant de l'utilisateur	N	Attribut	5
A	nomUtilisateur	Nom de l'utilisateur	A	Attribut	50
A	prenomUtilisateur	Prénom de l'utilisateur	A	Attribut	50
A	dateNaissance	Date de naissance de l'utilisateur	Date	Attribut	10
A	adresseUtilisateur	Adresse de l'utilisateur	AN	Attribut	50
A	telephoneUtilisateur	Téléphone de l'utilisateur	A	Attribut	50

A	mailUtilisateur	Mail de l'utilisateur	A	Attribut	50
A	nbEmprunts	Nombre d'emprunts de l'utilisateur	N	Attribut	5
A	nbPenalites	Nombre de pénalités que l'utilisateur a eu	N	Attribut	5
В	idCategorie	Catégorie de l'utilisateur	N	Attribut	5
В	nomCategorie	Nom de la catégorie	A	Attribut	50
С	idAdmin	Id de l'administrateur	N	Attribut	5
С	nomAdmin	Nom de l'administrateur	A	Attribut	50
С	prenomAdmin	Prénom de l'administrateur	A	Attribut	50
С	adresseAdmin	Adresse de l'administrateur	A	Attribut	50
D	dateDebut	Date de début de l'emprunt	DATE	Attribut	10
D	dateFin	Date de fin de l'emprunt	DATE	Attribut	10
D	Rendu	Savoir si le document est rendu ou non	BOOLEAN	Attribut	1
Е	idDocumentaliste	Numéro du documentaliste	N	Attribut	5
Е	nomDocumentaliste	Nom du documentaliste	A	Attribut	50
Е	prenomDocumentaliste	Prénom du documentaliste	A	Attribut	50
F	idExemplaire	Numéro de l'exemplaire	N	Attribut	5
F	problemeExemplaire	Problème que l'exemplaire peut avoir (déchiré, abîmé etc)	A	Attribut	50
G	idDocument	Numéro du document	N	Attribut	5
G	titreDocument	Titre du document	A	Attribut	50

G	dateParution	Date de parution du document	DATE	Attribut	10
Н	idLangue	Numéro de la langue du document	N	Attribut	5
Н	nomLangue	Nom de la langue du document	A	Attribut	50
I	idAuteur	Numéro de l'auteur	N	Attribut	5
I	nomAuteur	Nom de l'auteur	A	Attribut	50
I	prenomAuteur	Prénom de l'auteur	A	Attribut	50
I	dateNaissanceAuteur	Date de naissance de l'auteur	DATE	Attribut	10
J	IdType	Identifiant de l'utilisateurs	N	Attribut	5
J	nomType	Nom du type	A	Attribut	50
K	IdHistorique	Identifiant de l'historique	N	Attribut	5
K	dateHistorique	Date de l'ajout ou la suppression de l'utilisateurs	DATE	Attribut	10
K	LibelléHistorique		A	Attribut	50
	Durée	Durée de l'emprunt	DATE	Calculable	10

Dans le temps, nous avons apportés certaines modifications dans la base de données avec un nouveau MCD que voici :



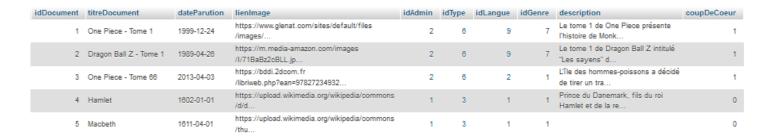
Nous avons donc rajouté les tables : Genre, Article et Message.

- La table Genre va nous permettre de pouvoir insérer des genres de documents, si ce sont des romans, des pièces de théâtres, des mangas etc.
- La table Article permet à la documentaliste d'ajouter des articles sur notre page d'accueil, ce seront principalement des articles sur l'actualité du monde, du pays, et de l'université.
- Enfin, la table Message, va permettre à l'utilisateur d'envoyer un message, qui lui servira de contact s'il a besoin d'aide. Ce message sera lu par la documentaliste.

De plus, nous avons modifiés quelques tables déjà existantes, comme avec l'ajout d'un lien image pour que chaque document soit associé à une page de couverture par exemple. La table exemplaire se voit recevoir un nouvel attribut « estEmprunte » qui est un booléen permettant de savoir si l'exemplaire d'un document est actuellement emprunté ou non.

Insertions

Nous pouvons donc ici voir des exemples d'insertions.



Les Déclencheurs

Nous avons fait des déclencheurs que voici :



Le déclencheur :

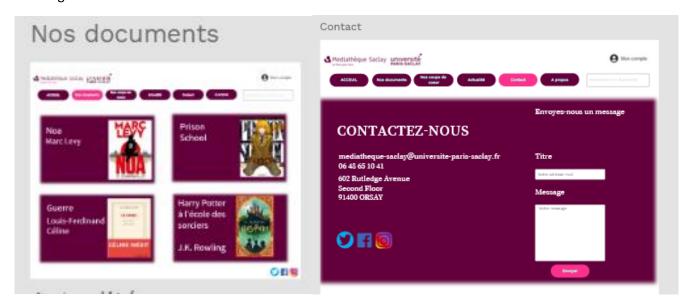
- AjoutEmprunt permet d'incrémenter le nombre d'emprunts quand un utilisateur fait un emprunt.
- SiSupprimeRetireEmprunt permet de mettre à zéro le statut d'un exemplaire lorsque l'emprunt de celui-ci est fini
- dateFinEmprunt permet de signaler une erreur si la date de fin dépasse 30 jours après la date emprunt.
- isEmprunte met le statut de l'exemplaire à 1 quand il est emprunté.

ш. <mark>Maquettage</mark>

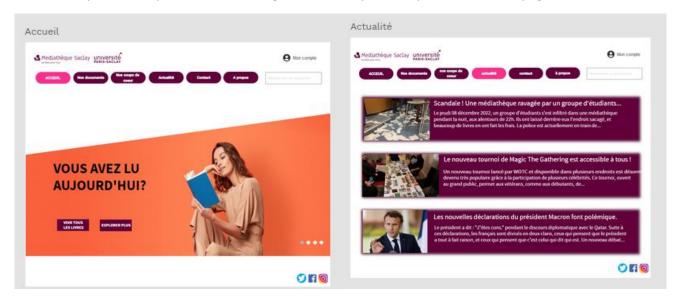
Pour pouvoir fournir un site web, il a fallu d'abord procéder à un maquettage, à partir de notre base de données et de nos fonctions, pour savoir comment le site allait être modéliser ou comment l'interface et l'expérience utilisateur pourrait être intuitive. Pour le maquettage, nous nous sommes aidés du site Penpot qui nous a permis de nous donner une idée de la modélisation de notre site web.

Maquettes

Nous pouvons voir ci-dessous un exemple de maquettage pour nos pages « Nos Livres » et « Contact », avec un menu de navigation



Nous avons par la suite procédé à des changements, voici par exemple si dessous les pages Accueil et Actualité



Que nous avons décidés finalement d'unir dans l'onglet Accueil :

Bienvenue!



Bienvenue sur le site de la Médiathèque Saclay! Vous pouvez consulter et emprunter des documents parmi notre liste de doncuments via la page "Nos documents"! Vous pouvez également consulter les documents coups de coeurs des documentalistes dans la page "Nos Coups De Coeur"! Pour plus d'informations consultez la page "A propos". Vous pouvez nous contacter via la page "Contact".

(Ceci est donc la version finale de notre site, avec le menu et les onglets que nous allons voir en détail par la suite.)

IV. JDBC

La partie JDBC nous a permis de rajouter certaines fonctions pour calculer des statistiques pouvant être intéressante pour le fonctionnement de la médiathèque.

Fonctions

Voici par exemple la fonction UtilisateurFidele, permettant de notifier quel utilisateur a fait le plus d'emprunts

Un autre exemple est la fonction pourcentageDoc, qui permet de renvoyer le pourcentage d'un type de document parmi tous les documents

La fonction documentsRecents permet de nous montrer les 10 documents ayant les dernières dates de parution.

v. Web

Pour donner vie à notre site, nous avons donc dû procéder à la mise en place de la partie web, où nous verrons le fonctionnement de tout ce qui permet à notre site de fonctionner en ligne, en se basant sur un routeur, des contrôleurs, des méthodes, et évidemment de l'HTML et du CSS, tout en respectant le schéma MVC.

PHP

Avec l'aide du php, nous avons pu faire nos fonctions dans des contrôleurs permettant de donner un sens au site pour les utilisateurs, la documentaliste et l'administrateur.

Contrôleur

```
// insertion du modèle Utilisateur
// insertion du contrôleur (ITIE ))."/modèle/document.php");
// définition du contrôleur Utilisateur
class ControleurDocument(
// la méthode de récupération et d'affichage de tous les utilisateurs
public static function la contempnement() {

// critre = "cous les documents() {

critre = "cous les documents() {

critre = "cous les documents() {

include ("vue/document.php");

include ("vue/document.php");

include ("vue/document.php");

public static function lireDocument() {

critre = "un document";

di = _oET["idDocument"];

critre = "un document";

di = _oET["idDocument"];

critre = "un document";

di = _oET["idDocument"];

include ("vue/document);

critre = "un document");

di = _oET["idDocument"];

include ("vue/document);

critre = "un document");

di = _oET["idDocument"];

include ("vue/document);

critre = "un document");

di = _oET["idDocument"];

include ("vue/document);

include ("vue/document);

include ("vue/document);

public static function rechercherDocuments();

critre = "resultata";

cretre = _oEOT["texte"];

if (ictype almmis(cretre));

header("Location: ./index.php?page=documents");

}

cretre = _oEOT["texte"];

if ('ctype almmis(cretre));

include ("vue/document/resultatDocuments.php");

include ("vue/documents.pritre php");

include ("vue/documents.pritre php");

include ("vue/
```

Voici un exemple de notre contrôleur de document, avec les fonctions :

- lireDocuments, qui permet d'afficher tous les documents
- lireDocument, qui permet d'afficher seulement un document,
- rechercherDocuments, que nous utiliseront dans une barre de recherche pour pouvoir taper le nom d'un document et le trouver plus facilement sans avoir à tout parcourir.
- lireCoupDeCoeur, qui permet d'afficher l'onglet coup de cœur avec les principaux documents que la médiathèque recommande.

Méthode

Pour que la documentaliste et l'administrateur puissent exercer certaines tâches, nous leur avons rajoutés la méthode suivante ajtArticle permet à la documentaliste d'ajouter un article sur la page d'accueil comme nous l'avons vu précédemment. Grâce à cette méthode, la documentaliste n'a qu'à ajouter un titre, une image, et une description d'un article pour que celui-ci soit fonctionnel.

```
public static function ajtArticle()
{
    //si le formulaire est vide
    if (empty($_POST)) {
        throw new HttpException();
    }
    require_once "config/connexion.php";

    //si tout les champs du formulaire sont fournis
    if (isset($_POST["titreArticle"], $_POST["texteArticle"], $_POST["image"])
    si lempty($_POST["titreArticle"], $_POST["texteArticle"]) $6 !empty($_POST["image"])) {
        $titre = strip_tags($_POST["titreArticle"]);
        $texte = htmlspecialchars($_POST["texteArticle"]);
        $lien = strip_tags($_POST["image"]);

        $requete = "INSERT INTO ARTICLE ('titre', 'texte', 'lienImage', 'idDocumentaliste') VALUES (:titre, :texte, :lien, l)";
        $req prep = Connexion::pdo()-"prepare($requete);
        $req prep-">bindValue(":titre", $titre, PDO::PARAM_STR);
        $req prep-">bindValue(":titre", $texte, PDO::PARAM_STR);
        $req prep-">bindValue(":titre", $texte, PDO::PARAM_STR);
        $req prep-">bindValue(":lien", $lien, PDO::PARAM_STR);
        $req prep-">bindValue(":lien", $lien, PDO::PARAM_STR);
        $red prep-">bindValue("
```

Mais également les méthodes suppArticle et ajtFav, qui permettent à la documentaliste de supprimer les articles ou d'en ajouter certains en favoris.

```
public static function suppArticle()
     //si le formulaire est vide
         (empty($_POST)) {
  throw new HttpException();
     require once "config/connexion.php";
     //si tout les champs du formulaire sont fournis
     if (isset($_POST["idArticle"]) && !empty($_POST["idArticle"])) {
    $id = strip_tags($_POST["idArticle"]);
         $requete = "DELETE FROM ARTICLE WHERE idArticle = :id";
         $req_prep = Connexion::pdo()->prepare($requete);
$req_prep->bindValue(":id", $id, PDO::PARAM_INT);
         if (!$req_prep->execute()) {
    die("Une erreur est survenue");
         } else {
               header('Location: index.php')
              //die("le livre $titre a bien été ajouté");
public static function aitFav()
     //si le formulaire est vide
    if (empty($_POST)) {
    throw new HttpException();
     require once "config/connexion.php":
     //si tout les champs du formulaire sont fournis
     if (isset($_POST["idArticleFav"]) && !empty($_POST["idArticleFav"])) {
         $id = strip_tags($_POST["idArticleFav"]);
          $requete = "UPDATE DOCUMENT SET coupDeCoeur = 1 WHERE idDocument = :id";
         $req_prep = Connexion::pdo()->prepare($requete);
$req_prep->bindValue(":id", $id, PDO::PARAM_INT);
         if (!$req_prep->execute()) {
         die("Une erreur est survenue");
} else {
   header('Location: index.php?page=documentaliste');
               //die("le livre $titre a bien été ajouté");
```

Voici donc ce que voit la documentaliste sur sa page, à savoir les champs pour ajouter/supprimer un article, ainsi que les messages des utilisateurs.



Modele

```
public static function getDocumentById($i) {
    // écriture de la requête
    $requetePreparee = "SELECT * FROM DOCUMENT WHERE idDocument = :id_tag;";
    // préparation de la requête
    $req prep = Connexion::pdo()->prepare($requetePreparee);
    // le tableau des valeurs
    $valeurs = array("id_tag" => $i);
    try {
        // envoi de la requête
        $req prep->execute($valeurs);
        // traitement de la réponse
    $req_prep->setFetchmode(PDO::FETCH_CLASS,'Document');
       // récupération de la voiture
        $1 = $req prep->fetch();
       // retour
        return $1;
    } catch(PDOException $e) {
        echo $e->getMessage();
```



Notre devoir final était donc de rendre toutes ces fonctionnalités accessibles simplement pour l'utilisateur. Nous pensons que le CSS est vraiment important pour l'expérience de l'utilisateur, car l'esthétique d'un site peut donner envie ou non à l'utilisateur d'utiliser le site. Le CSS est donc une tâche que nous n'avons sûrement pas sous-estimé pour le bien de l'utilisateur.

Voici le CSS du menu de navigation :

Pour avoir un menu de navigation stable et fixe peu importe la taille d'écran des utilisateurs, nous avons utilisés la méthode des grilles, permettant de mettre en place un cadrillage en ligne et en colonne, servant à placer notre logo, nos onglets et notre barre de recherche. Nous avons donc procédés à une méthode 2 lignes 7 colonnes, avec le logo sur la première ligne, qui prend en largeur les 3 premières colonnes. Les onglets sur la deuxième ligne juste en dessous du logo. La barre de recherche également sur la deuxième ligne, à l'opposé des onglets, et la page « Mon Compte » sur la première ligne juste au-dessus de la barre de recherche, ce qui a donné une barre de navigation de ce type :



vi. Conclusion

Nous pensons que nous avons répondu aux demandes essentielles des futurs clients pour ce site. Cependant, nous aurions aimés pouvoir faire plus de modifications tels que l'amélioration du CSS avec une police personnalisée par nous-même et un peu plus de décoration comme un défilement d'images de la médiathèque.

De plus, nous aurions voulu ajouter quelques fonctions permettant par exemple de sanctionner une personne qui a rendu des documents en retard, comme prévu avant le projet, mais cela a été impossible en raison du temps que nous avions pour le rendu. Une autre fonction était le fait que la documentaliste réussit à valider les emprunts réservés.