

# Gestion de prêt entre voisins

---



Anthony Pfister – CIN4A

ETML

Le 08.05.2023

89h

Gaël Sonney

Daniel Berney

Benjamin Wolf

# Table des matières

<b>1</b>	<b>SPÉCIFICATIONS.....</b>	<b>4</b>
1.1	TITRE.....	4
1.2	DESCRIPTION.....	4
1.3	MATÉRIEL ET LOGICIELS À DISPOSITION .....	4
1.4	PRÉREQUIS .....	4
1.5	CAHIER DES CHARGES.....	4
1.5.1	Objectifs et portée du projet .....	4
1.5.2	Fonctionnalités requises (du point de vue de l'utilisateur).....	4
1.5.3	Contraintes.....	5
1.5.4	Travail à réaliser par l'apprenti.....	5
1.5.5	Méthodes de validation des solutions .....	5
1.6	LES POINTS SUIVANTS SERONT ÉVALUÉS .....	6
1.7	VALIDATION ET CONDITIONS DE RÉUSSITE.....	6
1.8	POINT TECHNIQUE ÉVALUÉS SPÉCIFIQUES AU PROJET .....	6
<b>2</b>	<b>PLANIFICATION INITIALE.....</b>	<b>6</b>
<b>3</b>	<b>ANALYSE.....</b>	<b>7</b>
3.1	OPPORTUNITÉS .....	7
3.1.1	Liste des compétences à acquérir ou approfondir :.....	7
3.1.2	Liste du matériel à exploiter :.....	7
3.1.3	Recherche d'informations particulières .....	7
3.2	DOCUMENT D'ANALYSE ET CONCEPTION .....	7
3.2.1	Méthode de gestion de projet :.....	7
3.2.2	Liste de page de code a réalisé durant le projet :.....	8
3.2.3	Ergonomie des interfaces du site .....	10
3.2.4	Plan de gestion des branches GIT.....	18
3.2.5	Analyse des tables à créer dans la base des données :.....	18
3.3	CONCEPTION DES TESTS .....	19
3.3.1	Test automatique avec Sélénium .....	19
3.3.2	Test de fonctionnalité générale : .....	20
3.4	PLANIFICATION DÉTAILLÉE .....	20
<b>4</b>	<b>RÉALISATION .....</b>	<b>20</b>
4.1	DOSSIER DE RÉALISATION.....	20
4.1.1	Génération de la base de données du site, schémas MCD, MLD, MPD .....	20
4.1.2	Développement des premières pages de base du site .....	23
4.1.3	Développement de la gestion de l'authentification.....	24
4.1.4	Développement Opération CRUD, Ajout, modifier, détail et supprimer un article 26	
4.1.5	Développement Barre de recherche liste d'article.....	29
4.1.6	Développement emprunter un article .....	30
4.1.7	Développement lister les articles prêtés et empruntés d'un utilisateur .....	31
4.1.8	Développement système de notification .....	32
4.1.9	Développement communication SQL avec la base donnée .....	33
4.1.10	Mise en place de Selenium.....	33
4.2	MODIFICATIONS .....	33
<b>5</b>	<b>TESTS.....</b>	<b>33</b>
5.1	DOSSIER DES TESTS.....	33
<b>6</b>	<b>CONCLUSION.....</b>	<b>33</b>

6.1	BILAN DES FONCTIONNALITÉS DEMANDÉES.....	33
6.2	BILAN DE LA PLANIFICATION .....	33
6.3	BILAN PERSONNEL .....	33
<b>7</b>	<b>DIVERS.....</b>	<b>33</b>
7.1	JOURNAL DE TRAVAIL .....	33
7.2	BIBLIOGRAPHIE .....	33
7.3	WEBOGRAPHIE.....	34
<b>8</b>	<b>ANNEXES .....</b>	<b>34</b>

# 1 SPÉCIFICATIONS

## 1.1 Titre

Gestion de prêt entre voisin

## 1.2 Description

Le but de ce projet est de réaliser une application web permettant la gestion de prêt et emprunt entre voisins.

Cette application est destinée à toute personne voulant mettre à disposition et emprunter des articles dans le cadre d'un voisinage. Ces articles sont principalement des objets de bricolage, de jardinage et de cuisine qui ne sont utilisés que temporairement par le propriétaire. Des personnes du voisinage ont la possibilité d'emprunter les objets mis à disposition par un propriétaire. Il y a donc plusieurs utilisateurs qui peuvent être à la fois propriétaire et emprunteur.

Chaque utilisateur peut ajouter des articles et les mettre à disposition. Chaque utilisateur peut emprunter des articles disponibles. Un article est disponible s'il n'est pas emprunté. Chaque utilisateur peut consulter sa liste d'articles prêtés et empruntés.

## 1.3 Matériel et logiciels à disposition

- Un PC standard de l'ETML (Windows 10)
- Visual Studio code avec environnement PHP installé
- Serveur web local (uWamp ou autre)
- Suite Microsoft Office pour la documentation
- Un dépôt GIT (GITHUB, BitBucket ou autre)

## 1.4 Prérequis

- Connaissances en programmation PHP et en POO (Modules ICT 403, 404, 226, 120, 326, 411, 133)
- Connaissances en modélisation et implémentation de bases de données relationnelles (Modules ICT 104, 105, 153)

## 1.5 Cahier des charges

### 1.5.1 Objectifs et portée du projet

### 1.5.2 Fonctionnalités requises (du point de vue de l'utilisateur)

Les fonctionnalités présentes dans l'application seront les suivantes :

- Gestion de l'authentification
  - Opération CRUD
  - Ajout d'un article

- Modification d'un article
  - Suppression d'un article
- Affichage des détails d'un article
- Rechercher un article
- Emprunter un article
- Lister les articles prêtés et empruntés d'un utilisateur
- Mise en place d'un système de notification
- Mise en place de tests automatiques avec Selenium

### 1.5.3 Contraintes

Le projet devra se faire en utilisant Git (GitHub).

### 1.5.4 Travail à réaliser par l'apprenti

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale à la fin du premier jour
- Un rapport de projet contenant au minimum:
  - L'analyse du projet (un ou des schémas de principe sont souhaitables pour illustrer les réflexions rédigées).
  - L'analyse devra contenir des informations sur les objectifs, le contenu et le public cible avec les conséquences.
  - La réalisation répondant au cahier des charges (recherches de solutions, fonctionnalités, argumentation sur les choix effectués, obstacles rencontrés, mentions des aides extérieures apportées, etc.), ainsi qu'une charte graphique et une maquette du design. Les schémas de base de données (MCD, MLD, MPD) sont aussi attendus.
  - Les tests planifiés et effectués avec les résultats et analyses de ceux-ci.
  - La conclusion contenant, en outre, des considérations personnelles, des considérations techniques et des considérations de réalisation du projet (comparaison entre ce qui devait être fait et ce qui a été réellement fait, etc.).
- Le rapport de projet en l'état sera livré 2fois par semaine, le mercredi et le vendredi en fin de journée. Il sera livré sous forme de PDF dans le repo Git dans un dossier dédié à la documentation (/doc par exemple).
- Le journal de travail avec mentions de ce qui a été accompli, les suites à donner chaque fois qu'elles sont envisagées, liens et références des informations collectées ou retranscrites.
- Le code source de l'application et les scripts de base de données (liens vers le dépôt GIT fourni au CDP et aux experts)

### 1.5.5 Méthodes de validation des solutions

Une partie des tests devront se faire avec Sélénium de manière automatique.

## 1.6 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

## 1.7 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

## 1.8 Point technique évalués spécifiques au projet

- Choix, modélisation et intégration de la base de données
- Qualité et lisibilité du code, respect des conventions de nommages standards
- Les opérations CRUD sur un article (ajout, modification, suppression et afficher les détails, d'un article).
- L'utilisateur doit pouvoir rechercher un article qu'il veut louer, l'emprunter et terminer l'emprunt.
- Un système de notification permettant d'avertir l'utilisateur que la date de fin de prêt est proche.
- Des tests automatisés sont implémentés avec le Framework Sélénium

# 2 PLANIFICATION INITIALE

- Début du projet le 8 mai 2023 et fin au 7 juin 2023.
- Il y a une période de congé le 29 mai (Pentecôte), le 18 mai et 19 mai (Ascension). Et l'examen ECG qui est le 1<sup>er</sup> juin.
- 89h pour la réalisation du projet.
- Dans l'ordre des tâches du projet :
  1. Analyse - cahier des charges,
  2. Analyse - Recherche information et renseignement,
  3. Analyse - Création de la planification initiale,
  4. Analyse - Création du journal de travail,
  5. Analyse - Création des répertoires de travaux,
  6. DOC - Création du Rapport de projet,
  7. Réalisation - Schémas MCD, MLD, MPD,
  8. Réalisation - Développement Gestion de l'authentification,
  9. Réalisation - Développement Opération CRUD,
  10. Réalisation - Développement Rechercher un article,
  11. Réalisation - Développement Emprunter un article,
  12. Réalisation - Développement Lister les articles prêtés et empruntés d'un utilisateur,

13. Réalisation - Développement Mise en place d'un système de notification,
14. Réalisation - Développement Mise en place de tests automatiques avec Selenium,
15. Tests - Test automatisés avec Selenium,
16. Tests - fonctionnalités globales,
17. Doc - Rédaction Spécifications,
18. Doc - Rédaction Planification initiale,
19. Doc - Rédaction Analyse,
20. Doc - Rédaction Réalisation,
21. Doc - Rédaction Tests,
22. Doc - Rédaction Conclusion,
23. Doc - Rédaction Divers et annexes,
24. Doc - Journal de travail remplissage

## 3 ANALYSE

### 3.1 Opportunités

#### 3.1.1 Liste des compétences à acquérir ou approfondir :

- Savoir mettre en place une gestion d'authentification
- Pouvoir mettre en place des opérations CRUD pour un article
- Mettre un système de barre de recherche pour un article sur le site
- Savoir faire une gestion d'emprunt d'un article sur le site
- Pouvoir lister les emprunts d'un utilisateur en cours
- Mettre en place un système de notification sur le profil.
- Comprendre Selenium et l'utiliser pour faire des tests de fonctionnalités

#### 3.1.2 Liste du matériel à exploiter :

- Un serveur Web local uWamp
- Un dépôt GitHub

#### 3.1.3 Recherche d'informations particulières

- Recherche conception barre de recherche php
- Recherche conception notification

### 3.2 Document d'analyse et conception

#### 3.2.1 Méthode de gestion de projet :

J'ai utilisé la méthode des 6 pas, car c'est la méthode avec la quel j'ai le plus de facilité en faisant ces raisonnements :



- S'informer :

Je me suis informé des fonctionnalités demandées dans le cahier des charges. J'ai recherché les conditions et prérequis avant le commencement. Et aussi, me renseigner qu'elle était les fonctionnalités à prioriser pour le bon déroulement du projet.

- Planifier :

J'ai planifié le projet en me basant sur le cahier des charges. L'ordre de tâches à effectuer et leurs durées plus ou moins précise pour pouvoir une vue globale du projet. La répartition des tâches est de 10% pour la partie analyse soit ~17h, 40% pour la partie réalisation soit ~35h, 10% pour la partie test soit ~9h et 30% pour la partie documentation soit ~26h

- Décider :

J'ai décidé ensuite de la méthode de travail, de quelle tâche effectuer en premier par priorité, après m'être documenter sur chaque tâche à implémenter. Je crée également une liste de test à faire à la fin de projet.

- Réaliser :

Je réalise le projet en me basant sur la planification initiale le plus possible. À chaque commencement d'une nouvelle tâche, je prends quelque minute pour me renseigner spécifiquement sur ça réalisation et ensuite je l'effectue. Pour chaque fin de journée, je documente les avancements dans le rapport et le journal de travail pour ne rien oublier

- Contrôler :

J'effectue les tests que j'ai décidé de faire afin de bien vérifier le fonctionnement global de projet que je vais rendre. À la fin de cette phase, je crée la version finale du site que je dois rendre.

- Evaluer :

J'analyse le projet globalement en fonction de la planification initiale et tu journal de travail. Je fais le bilan des fonctionnalités qui ont été réalisés. Par la suite, je fais le bilan de ma gestion de mon projet, les défauts et les ratés. De voir quel sera les choses à améliorer pour un futur projet.

### 3.2.2 Liste de page de code a réalisé durant le projet :

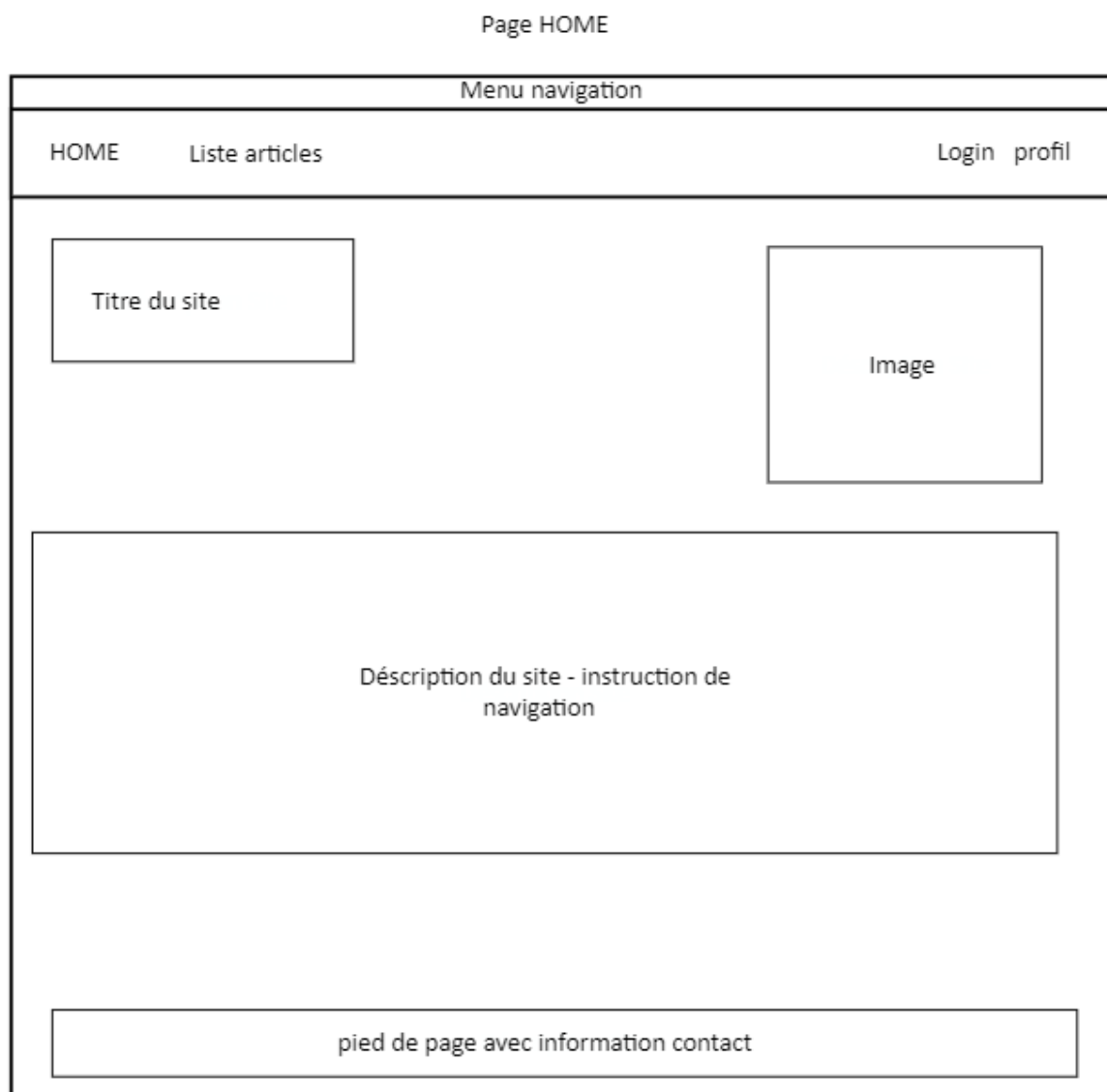
- Page home - page d'accueil
- Page menu - onglet de navigation des pages, home, articleListe, login
- Page login - page de login, utilisateur rentre son nom et mots de passe
- Page checkLogin - check si le nom et le mots de passe correspond à la base de données, si non -> page login
- Page checkConnection - check si l'utilisateur est connecter avant d'utiliser une fonctionnalité, si non -> page login
- Page disconnect - page de déconnexion
- Page addArticle - page d'ajout d'un article



- Page checkInsertArticle - check si l'article est conforme à la base de données
- Page article - page qui affiche toutes les infos d'un article,
- Page empruntArticle - page de l'emprunt d'un article. D'utilisateur indique les dates de son emprunts.
- Page checkEmprunt - check si l'emprunt est possible
- Page suppArticle - supprime l'article
- Page articleListe - liste tous les article
- Page rechercheBarre - barre de recherche pour trouver un article, exclusivement sur la page articleListe
- Page checkRechercheBarre - check les informations saisie dans la barre de recherche, et affiche le resultat
- Page modifArticle - page modification d'un article
- Page checkModifArticle - check les modification de l'article
- Page userDetails - page de détail d'un utilisateur, information, liste d'article, notif
- Page notifier - affiche un bandeau en dessous du menu si y fin d'un prêt proche
- Page Database - page de communication à la Database et le site
- Page config - Configuration de la connexion à la db
- Page footer - Bas de page sur tout le site

### 3.2.3 Ergonomie des interfaces du site

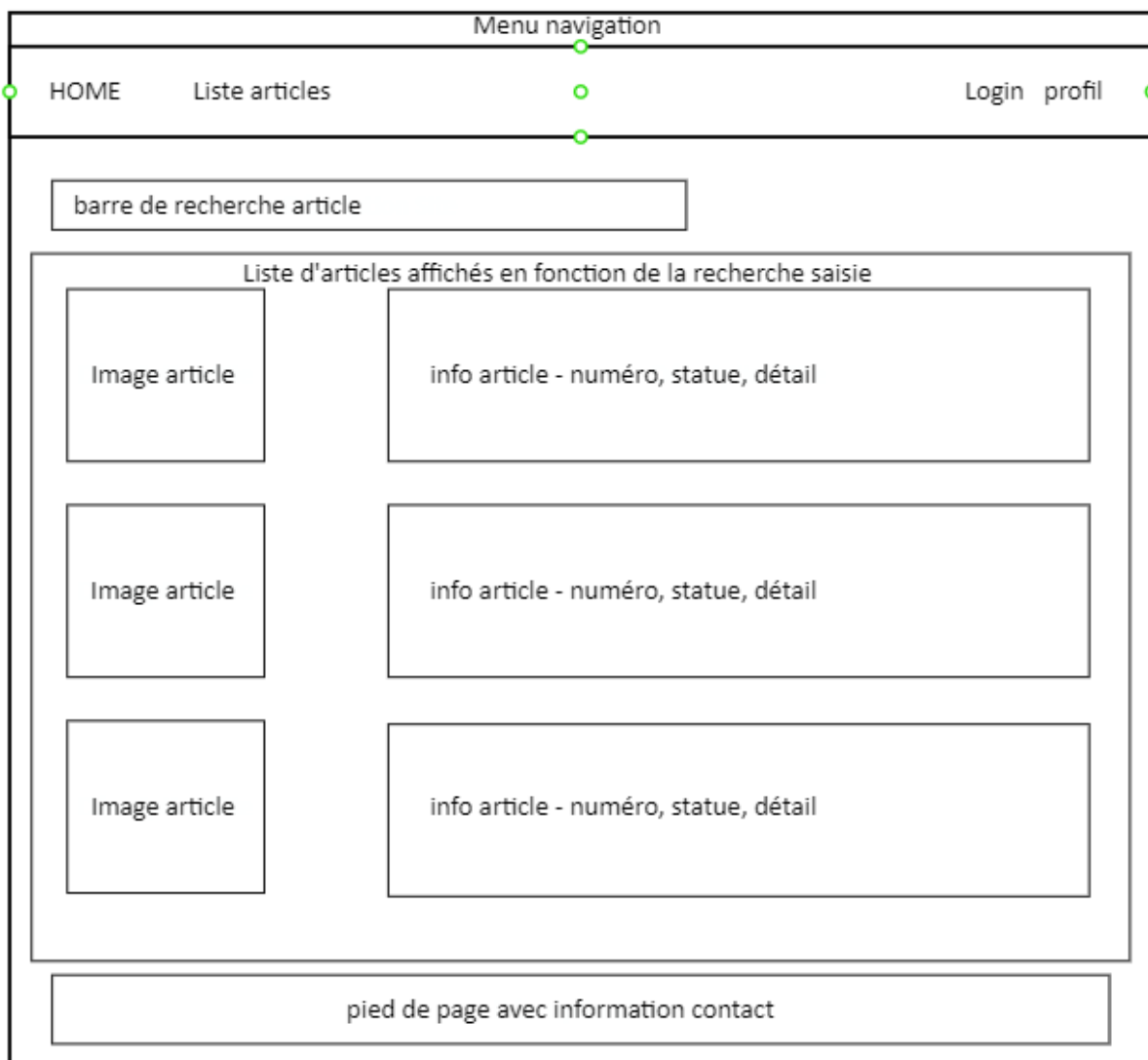
- Page HOME



Maquette de la page d'accueil du site. Sur la page on retrouve le menu de navigation pour pouvoir accéder aux différentes pages du site. Le titre du site ainsi qu'une description avec des instructions de navigation et d'utilisation des fonctions du site. Et le pied de page avec des informations de contact du créateur.

- Page Liste d'article

### Page Liste Article



Maquette de la page de la liste d'article à emprunter. Avec une barre de recherche qui tri la liste en fonction du lieu. On retrouve également le menu de navigation et le pied de page.

- Page article

Page Article

Menu navigation

HOME   Liste articles   Login   profil

Information Article :

image d'erreur

Nom

Auteur de l'article

lieu

statue - disponible/indisponible

Description de l'article

Bouton Emprunter l'article

Bouton Modifier l'article

Bouton supprimer l'article

image de l'article

pied de page avec information contact

Maquette de la page d'un article. L'utilisateur peut voir le nom, l'auteur, le lieu, la description ainsi qu'une image de l'article. Il y a un statut disponible/indisponible. Si le statut de l'article est en indisponible, l'utilisateur ne pourra pas l'emprunter et un message d'erreur s'affichera avec l'information. L'utilisateur a accès au bouton modifier et supprimer de l'article. On retrouve également le menu de navigation et le pied de page.

- Page emprunt article

#### Page emprunt

Menu navigation

HOME    Liste articles    Login    profil

Formulaire à remplir :

image d'erreur

nom de l'article

image de l'article

Date de début

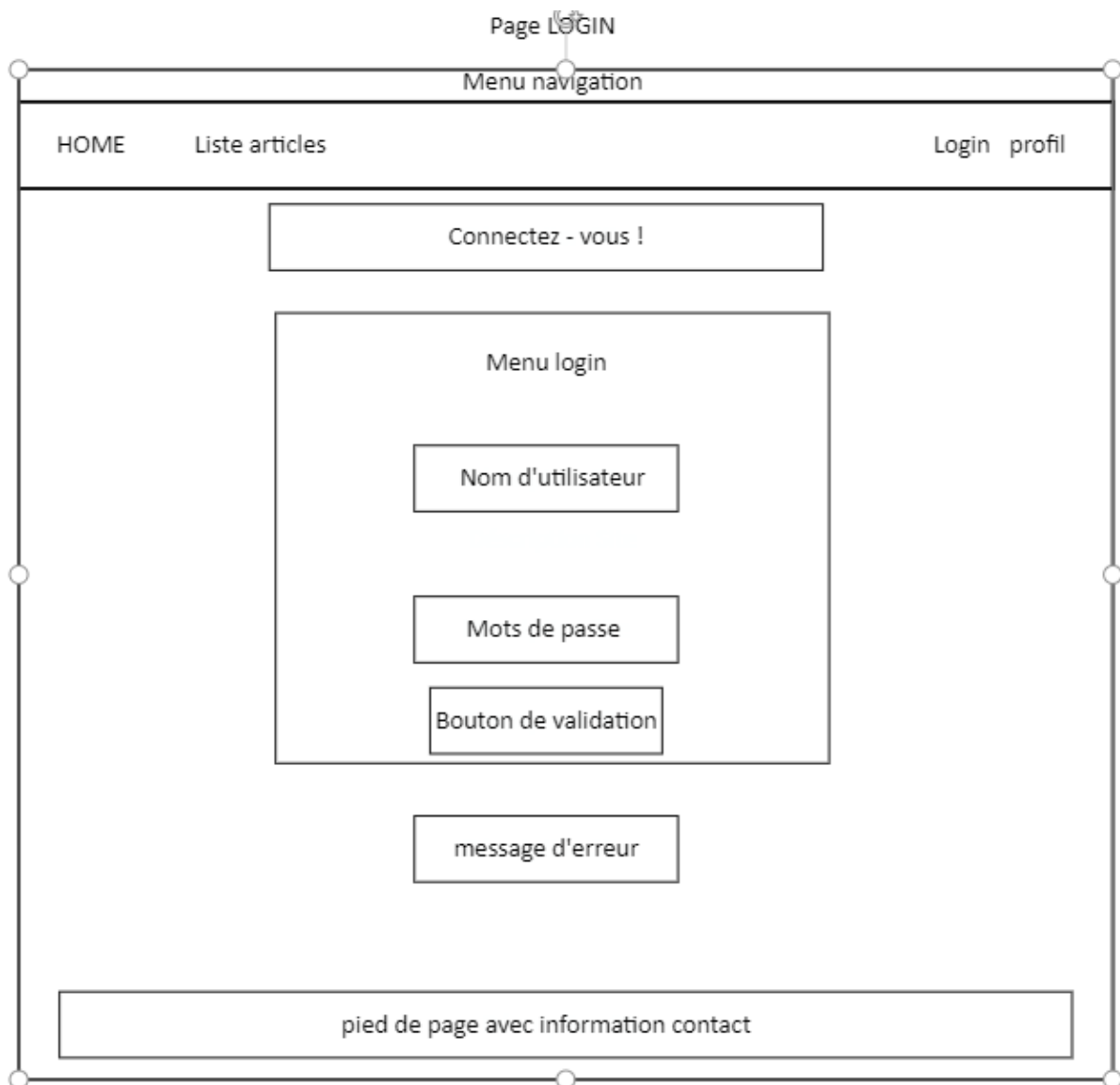
Date de fin

Bouton emprunt

pied de page avec information contact

Maquette de la page emprunt d'article. C'est un formulaire avec les dates de début et de fin à remplir. Ensuite, l'utilisateur appuie sur le bouton qui valide l'emprunt. Un message d'erreur s'affiche si l'utilisateur n'a pas rempli les dates ou mal remplis. On retrouve également le menu de navigation et le pied de page.

- Page Login



Maquette de la page de login. L'utilisateur doit renseigner son nom d'utilisateur ainsi que sont mots de passe. Si les informations ne sont pas juste ou pas remplis, un message d'erreur s'affiche et l'utilisateur doit recommencer. On retrouve également le menu de navigation et le pied de page.

- Page Profil

### Page Profil

Menu navigation			
HOME	Liste articles	Login profil	
Bandeau de notification - quelque jours avant la date de fin d'un emprunt			
Nom utilisateur	Bouton ajout article		
information utilisateur			
Liste articles emprunter en cours avec information			
Article	Article	Article	Article
Liste articles créer avec information			
Article	Article	Article	Article
pied de page avec information contact			

Maquette de la page de profil d'un utilisateur connecter. Il peut voir son nom d'utilisateur ainsi que ses informations. La liste des articles qu'il est en train d'emprunter s'affiche sur sa page et Il peut visualiser les détails de l'article en cliquant dessus, c'est un renvoi à la page article du site. Il y a aussi une liste des articles que l'utilisateur a créé, il peut également cliquer dessus pour voir les détails. L'utilisateur a, depuis cette page, accès aux boutons d'ajout d'article où il est redirigé. En dessous du menu de navigation, il y a un bandeau de notification qui l'informe d'une date de fin prochain d'un de ses articles en cours. On retrouve également le menu de navigation et le pied de page.



- Page ajout article

Page Ajout Article

Menu navigation

HOME   Liste articles   Login   profil

Formulaire d'ajout d'article

message d'erreur

à remplir :

Nom

téléchargement image

statue - disponible/indisponible

Description de l'article

Bouton Créer l'article

pied de page avec information contact

Maquette de la page d'ajout d'article. Ici, un utilisateur connecté peut ajouter un article en remplissant son nom, une image, une description et son statue. Ensuite l'utilisateur valide la saisie en appuyant sur le bouton. Si l'utilisateur n'a pas remplis les champs de saisie, le message d'erreur apparait et il est invité à recommencer. On retrouve également le menu de navigation et le pied de page.

- Page modification article

### Page modification article

Menu navigation

HOME    Liste articles    Login profil

Formulaire modification d'article

Message d'erreur

à remplir :

Nom

téléchargement image

statue - disponible/indisponible

Description de l'article

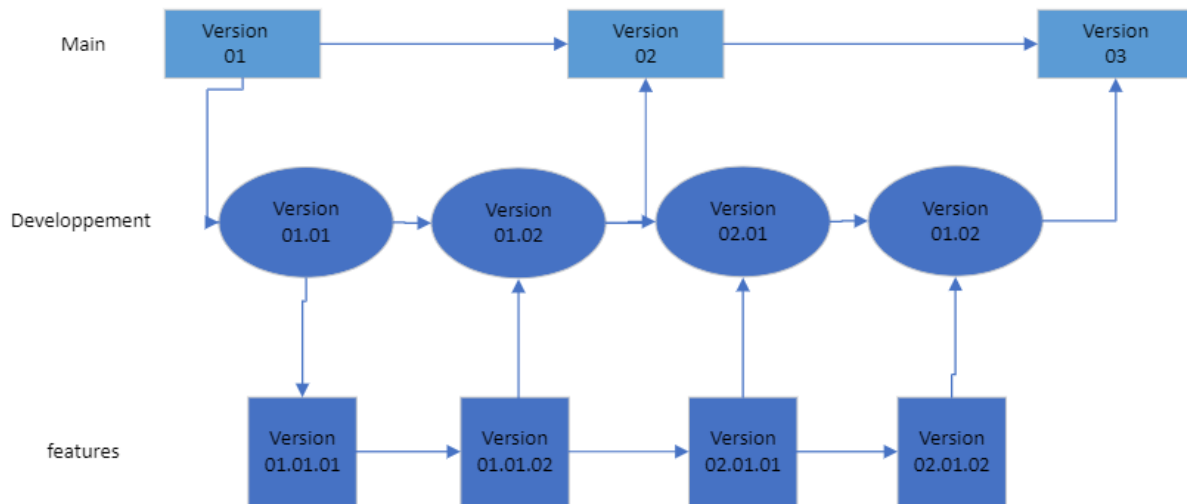
Bouton Modifier l'article

pied de page avec information contact

Maquette de la page de modification d'un article. L'utilisateur remplit le nouveau nom, insère une nouvelle image, change la statue ou change la description. Ensuite, l'utilisateur valide la modification en cliquant sur le bouton. Si les champs de saisie ne sont pas remplis ou incorrects, un message d'erreur s'affiche et l'utilisateur est invité à recommencer.

### 3.2.4 Plan de gestion des branches GIT

Schéma des branches dans GitHub à réaliser :



Ce schéma montre les différentes branches que je vais créer pour la réalisation du développement du site web. Les premières branches « main » seront les version principale et utilisables du code du site web. Les branches « développement » sont prévus pour les ajouts des nouvelles fonctionnalités finit. Les dernière branche de développement sont prévus pour les fonctionnalités en cours de développement et incomplète.

Un «Commit » devra s'effectuer à chaque fin de séquence dans la branche adapté.

### 3.2.5 Analyse des tables à créer dans la base des données :

Le site aura besoin d'une base de données avec trois tables pour son fonctionnement :

Table utilisateur :

- iduser
- userName
- usePassword
- useLocal
- useNbArticle
- useNbEmprunt
- useRegisterDate

La table utilisateur renseigne le nom, le mot de passe, la localisation, le nombre d'article qu'il a posté sur le site et également le nombre d'emprunt d'article qu'il a actuellement.

Table article :

- idArticle
- artName
- artStatue

- artPicture
- artDescription
- fkUserArticle

La table pour les articles répertorie le nom, son statut disponible ou indisponible, une image ainsi qu'une description. La table est reliée à la table utilisateur pour savoir de qui elle a été créée.

Table emprunt :

- idLoan
- loaBeginDate
- loaEndDate
- fkUser
- fkArticle

La table des emprunts renseigne la date de début et de fin de l'emprunt. Elle relie les tables utilisateur et article pour savoir de quel article cet emprunt est lié et à quel utilisateur du site.

### 3.3 Conception des tests

#### 3.3.1 Test automatique avec Sélénium

Des tests automatisés vont devoir être créés dans Selenium IDE et être exécutés après la fin de la réalisation de toutes les fonctionnalités. Dans l'ordre des tests à créer et à exécuter :

1. Ajouter un article : ce test automatisé saisira dans le formulaire de la page ajout d'article le nom d'un article simple avec une photo .JPG qui est téléchargée depuis le même répertoire du test, une description de l'article et un statut en mode « disponible ».
2. Modifier l'article créé : ce test automatisé va aller modifier un par un les données de l'article créé par le précédent test sur la page « Modifier l'article ». Saisir un nouveau nom, télécharger une nouvelle image .JPG, une nouvelle description et changer le statut en « indisponible ».
3. Emprunter l'article créé : ce test automatisé va emprunter l'article créé sur la page « Emprunt d'article ». Il saisira une date de début et de fin en cohérence avec la date actuelle où le test est effectué.
4. Modifier le statut de l'article : ce test automatisé va modifier juste le statut de l'article en « disponible » pour voir si l'emprunt créé par le précédent test va être supprimé par cette action.
5. Refaire l'emprunt de l'article précédemment supprimé : ce test automatisé va recréer le même emprunt avec les mêmes dates que l'emprunt précédemment supprimé.

6. Supprimer l'emprunt : Ce test automatique va supprimer l'emprunt avec le bouton qui se trouve sur la page Profil du site. Ce test a pour but de vérifier si les données du nombre d'emprunt de l'utilisateur est correcte.
7. Refaire une dernière fois l'emprunt de l'article précédemment supprimer : ce test automatisé va recréer le même emprunt avec les mêmes dates que l'emprunt précédemment supprimer.
8. Rechercher l'article dans la barre de recherche : ce test automatisé va saisir le lieu de l'article de test, écrit à moitié dans la barre de recherche. Ce test a pour but de vérifier si la barre de recherche est capable de trouver l'article en fonction du lieu de l'utilisateur qui l'a créer, et de voir si il arrive à le trouver même si le lieu n'a pas été écrit complètement.
9. Supprimer l'article : ce test automatiser va supprimer l'article entièrement avec l'emprunt en cliquant sur le bouton « supprimer l'article » sur la page « article ». Ce test a pour but de vérifier si les données de l'utilisateur sont bien mises à jour par rapport à l'action effectuer.

### 3.3.2 Test de fonctionnalité générale :

Les tests à effectuer à la fin de la réalisation du projet suivront cette liste :

1. S'identifier avec un utilisateur sans problème rencontrer.
2. Ajouter des articles.
3. Modifier des articles.
4. Supprimer des articles.
5. Afficher les détails de tous les articles enregistrer.
6. Utiliser la barre de rechercher pour trouver un article par rapport à son lieu
7. Faire des emprunts d'articles.
8. Vérifier si la barre de notification est fonctionnelle en mettant un article en fin d'une date d'emprunt.

## 3.4 Planification détaillée

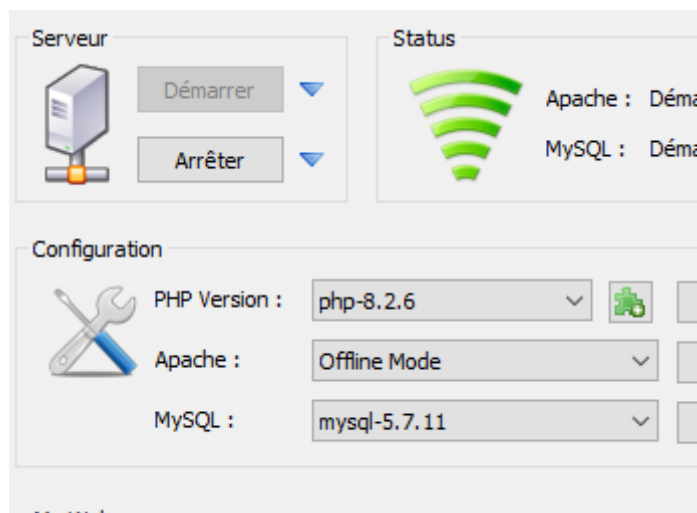
Voir annexe.

# 4 RÉALISATION

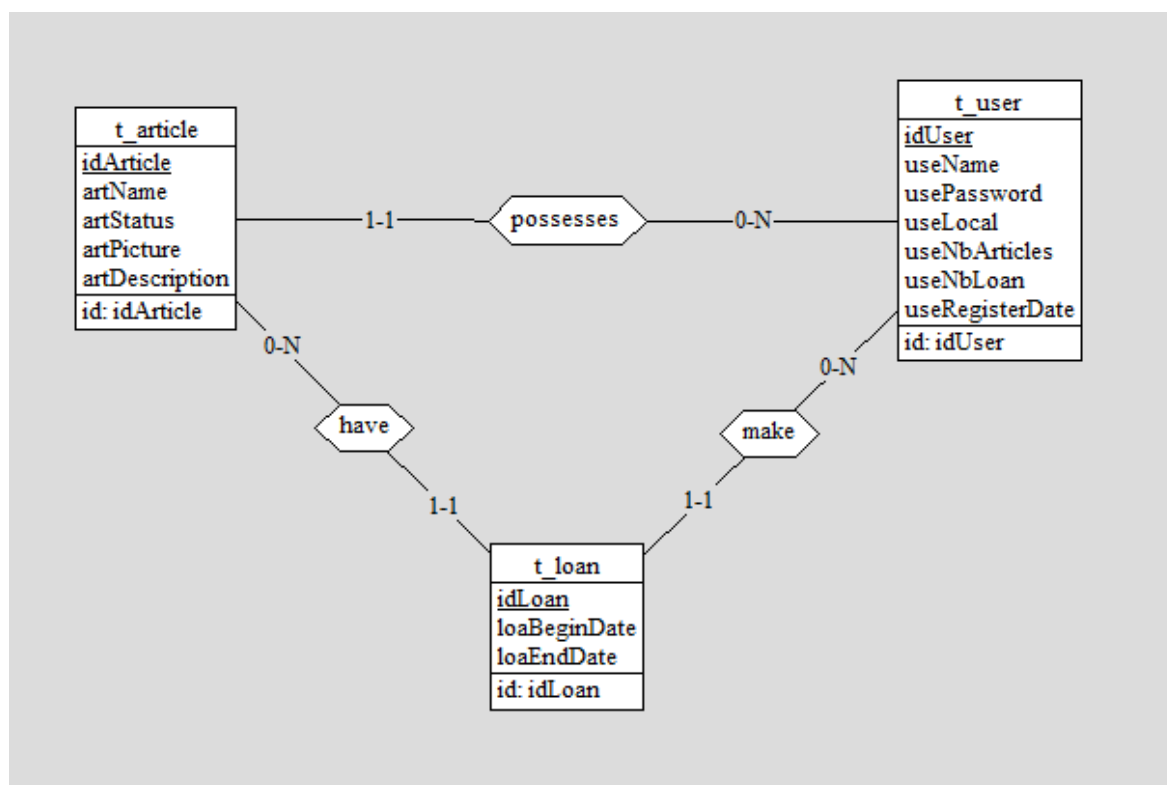
## 4.1 Dossier de Réalisation

### 4.1.1 Génération de la base de données du site, schémas MCD, MLD, MPD

Avant de commencer le projet, j'ai dû changer de version php de mon ServeurWeb local uWamp des versions de base (5.6,7.0) en 8.2.6 :



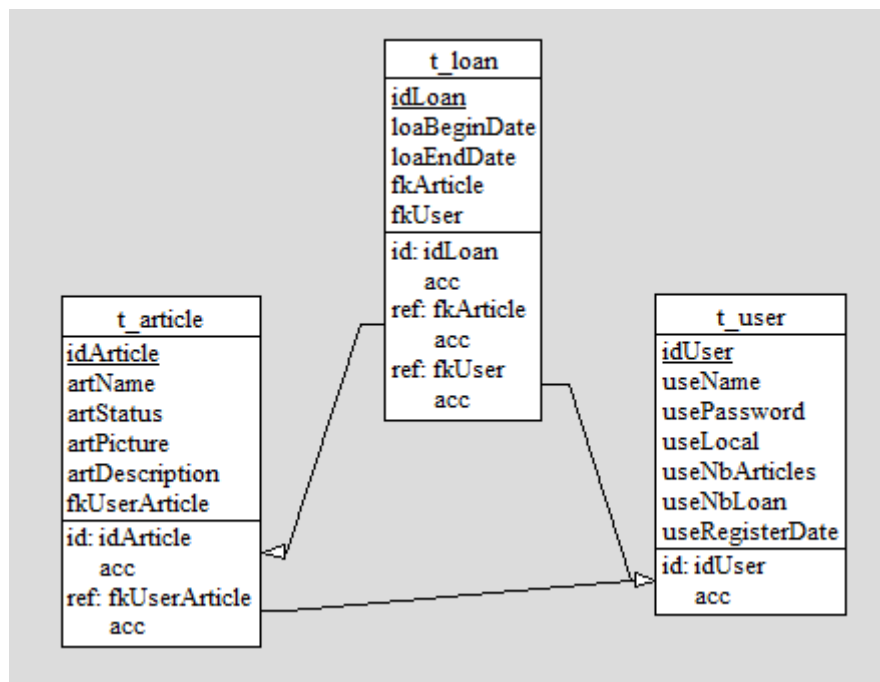
Le schéma MCD de la base de données :



J'ai d'abord créé un schéma MCD sur le logiciel DB\_MAIN pour visualiser les tables que j'avais prévu dans la partie analyse. Pour les relations entre les tables, la table « article » possède un utilisateur comme référence pour sa création, et pour partager le lieu. La table « emprunt » contient les références de l'article qui va se faire emprunter et l'utilisateur qui fait cet emprunt.

Pour le type des attributs des tables, la plupart sont en « varchar » pour pouvoir rentrer dans la base de données un nombre suffisant de données sans prendre trop de place. Pour les attributs que ne devons posséder que des numéro comme « artStatus », « useNbArticles » ou « useNbLoan », elle sont en « numeric ». Et les attributs date de la tables « emprunt » sont de type « date ».

Schéma MLD de la base de données :



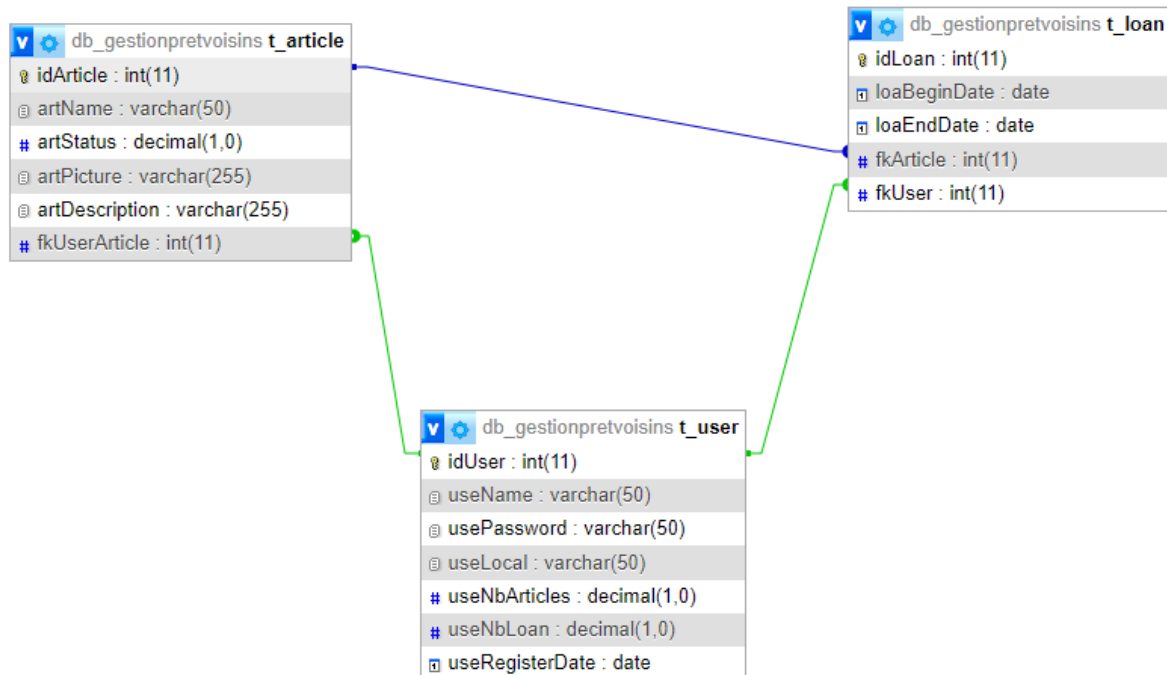
Ensuite j'ai pu générer le schéma MLD sur DB\_MAIN pour pouvoir voir le résultat. Après avoir fini de vérifier si les relations étaient correctes. J'ai généré le fichier SQL pour l'importer ensuite sur mon serveur uWamp.

Importation dans la base de données et visualisation dans phpmyadmin :

Table	Action	Lignes	Type	Interclassement	Taille	Perte
t_article	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	32 Kio	-
t_loan	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	48 Kio	-
t_user	Afficher Structure Rechercher Insérer Vider Supprimer	1	InnoDB	latin1_swedish_ci	16 Kio	-
<b>3 tables</b>	<b>Somme</b>	<b>1</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>96 Kio</b>	<b>0</b>



Schéma de la base de données dans le concepteur de phpMyAdmin :



#### 4.1.2 Développement des premières pages de base du site

Pour commencer le développement des fonctionnalités demandé, j'ai du d'abord créer les page home, menu, footer pour avoir une base graphique.

Première version visuelle de la page home sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Login](#)  
[Profil](#)

## Page Home



image freepik

Bienvenu sur la page d'accueil du site "Gestion de prêt entre voisins". Ici, vous pouvez retrouver des articles utile que vous pouvez emprunter au date que vous voulez.

Créer par Anthony Pfister - Projet TPI

Comme sur la maquette que j'ai créé dans la partie analyse, la page home inclue la page menu qui est le navigateur qu'on retrouve sur tous les pages du site. Et la page footer qui rajoute le pied de page avec les informations de contact. Je n'ai pas encore rajouté de style pour l'ensemble des pages pour le site, je réserve cette partie pour la fin. La page Home regoure pour le moment une image et une description provisoire pour l'accueil de l'utilisateur sur le site

### 4.1.3 Développement de la gestion de l'authentification

J'ai commencé le développement de la gestion de l'authentification par créer la page login et la page userDetails.

Première version visuelle de la page login sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Login](#)  
[Profil](#)

**Connecter-Vous !!**

**Renseigner vos information de connexion svp !**

Nom d'utilisateur
Mot de passe
<input type="button" value="S'identifier"/>

Créer par Anthony Pfister - Projet TPI

Première version visuelle de la page userDetails sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Login](#)  
[Profil](#)

## Détails de l'utilisateur antpfister

Créer par Anthony Pfister - Projet TPI

Pour la page login, l'utilisateur est invité à renseigner son nom d'utilisateur (userName) et son mots de passe (usePassword). Si l'utilisateur n'a pas saisi ses informations ou si elles sont fausse, alors la page est recharger avec un message en plus qu'il lui indique une erreur.

Page login si l'utilisateur a fait une erreur dans la saisie du formulaire de connexion :

[HOME](#)  
[liste des articles](#)  
[Login](#)  
[Profil](#)

## Connecter-Vous !!

### Renseigner vos information de connexion svp !

Nom d'utilisateur
Mot de passe
S'identifier

Vous avez mal rempli le formulaire de connexion !! s'il vous plaie recommencer.

Créer par Anthony Pfister - Projet TPI

Par contre si utilisateur a renseigné comme il faut ses données, alors utilisateur se retrouver sur la page home en étant connecter et peut accéder à tous le site et notamment à la page profil (userDetail) qui n'est pas possible s'il n'est pas connecté.

Le processus de vérification de la saisie de formulaire de connexion se trouve sur la page checklogin qui compare la saisie à la base de données du serveur local. La page connecte l'utilisateur si tous est correcte ou renvoie une erreur à la page login qui affiche un message d'erreur.

Cette partie de réalisation est enregistrer dans GitHub comme la version 01.00.00 du site de la branche main. Car, c'est à ce moment que le site est utilisable à ce niveau là.

#### 4.1.4 Développement Opération CRUD, Ajout, modifier, détail et supprimer un article

La première étape que j'ai effectué pour la confection des opération CRUD est l'ajout d'un article dans la base de données du site. J'ai créé la page « addArticle » où l'utilisateur est invité à renseigner le nom de l'article (artName), choisir une image .JPG(artPicture), écrire une description correspondant au produit (artDescription) ainsi que le Status qu'on peut choisir à sa création.

Ensuite, ces informations sont envoyées sur la page checkInsertArticle qui va vérifier si les informations rentrer sont correcte pour être entrer dans la base de donner et aussi voir si elles ne sont pas vides. Si ce n'est pas le cas, l'utilisateur est redirigé vers la page ajout d'article avec un message d'erreur en plus qui l'invite à recommencer sa saisit des informations.

Dans le cas où les informations sont correctes, la date et l'heure est rajouter au nom de l'image .jpg téléchargé, afin de ne jamais avoir le même nom dans le dossier images et d'enlever les erreurs éventuelles que cela pourrait causer. Et seulement ensuite, placé l'image dans le dossier images du code sources. Ensuite, les données sont enregistrer dans la table article de la base de données via une requête SQL qui se trouve sur la page Database. (voir point 4.1.9). Le nombre d'article par l'utilisateur connecté est mis à jour avec une unité en plus.

C'est page est accessible que depuis la page Liste article et seulement si l'utilisateur est connecté.

Première version visuelle de la page addArticle sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Profil](#)

## Ajouter un Article

Nom de l'article :

Photo de l'article (.jpg) :  Aucun fichier choisi

Description de l'article :

Le status de l'article : ☒ disponible ☐ indisponible

Créer par Anthony Pfister - Projet TPI

Première version visuelle de la page articleListe sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Profil](#)

### Liste des articles



long vue

Disponible

c'est une long vue

Créer par client, se situe à Lutry



Ensuite, je suis passé à la page de liste des articles où tous les articles sont affichés avec leur informations. L'utilisateur doit être connecté pour accéder à cette page. Et depuis cette page, il a accès à la page ajout d'article via un bouton en haut. Les articles sont affichés dans l'ordre de création dans la base de données. Une barre de recherche est prévue sur cette page pour rechercher les articles en fonction de leur lieu et sera développée dans la partie 4.1.5.

Les informations sont affichées grâce à une requête SQL qui récupère les données des tables articles et utilisateur correspondant. (Voir point 4.1.9).

L'utilisateur peut cliquer sur l'image de l'article pour voir les détails sur la page article.

Sur cette page d'info de l'article, l'utilisateur peut voir des les détails de cet article et peut s'il le souhaite le supprimer ou le modifier. Un bouton d'emprunt est également prévu et sera développer dans la partie 4.1.6. Toutes les informations de l'article sont affichées, grâce à une requête SQL depuis la page Database. (Voir point 4.1.9).

Première version visuelle de la page article sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Profil](#)

### Article info - paire de jumelle



**Disponible**

c'est non

**Créer par antpfister, se situe à Lausanne**

Depuis la page article, un bouton est prévu pour modifier cet article.

L'utilisateur est redirigé sur la page modification de l'article où l'utilisateur est invité à renseigner les nouvelles informations qu'il veut ajouter. Aucun champ n'est obligatoire, si l'utilisateur ne remplis rien alors l'article n'est pas modifié et l'utilisateur est rediriger vers la page article. Les champs sont remplis avec les données initial de l'article.

Ensuite les information remplis sont vérifier dans la page checkModifArticle qui renvoie un message d'erreur si les informations ne sont pas conformes. S'il y a une modification de l'image, alors l'image initiale est supprimée sur le répertoire et la nouvelle est ajouter. Ensuite toutes les nouvelles données sont envoyées et modifier dans la ligne de la table article qui correspond via une requête SQL via la page Database. (Voir point 4.1.9). Si le Statue d'un article modifier de « emprunter » à

« disponible », alors l'emprunt lier à cet article est supprimer.

J'ai rajouté après discussion avec le chef de projet le fait qu'uniquement le créateur de l'article a accès au bouton supprimer et modifier l'article. Ce choit a été fait pour des raison de cohérence du site.

Première version visuelle de la page modifArticle sans code CSS :

[HOME](#)  
[liste des articles](#)  
[Profil](#)

## Modification de l'article "Tondeuse"

Nom de l'article :

Photo de l'article (.jpg) :  Aucun fichier choisi

Description de l'article :

Le status de l'article : ☒ disponible ☐ indisponible


Créer par Anthony Pfister - Projet TPI

### 4.1.5 Développement Barre de recherche liste d'article


Pour le développement de la barre de recherche sur la page de liste des articles. J'ai créé une page checkRechercheBarre qui va vérifier la saisie que l'utilisateur a mis dans la barre de recherche. La page envoie ensuite cette donnée dans la page Database et ressort via une requête SQL la liste des Identifiant des articles rechercher via uniquement le lieu où le créateur de l'article se situe. (Voir point 4.1.9). Si l'utilisateur n'a rien remplie, la page checkRechercheBarre renvoie un message d'erreur invitant l'utilisateur a bien remplir la barre

Page Liste des Articles avec la barre :

**Liste des articles**



longue vue  
**Emprunter**  
ça marche  
Créer par client, se situe à Lutry



paire de jumelle  
**Disponible**  
ce sont des jumelles :D  
Créer par antpfister, se situe à Lausanne


Créer par Anthony Pfister - [anthony.pfister@eduvaud.ch](mailto:anthony.pfister@eduvaud.ch) - Projet TPI




Page de Liste des articles avec la recherche « Lausanne » entré :

**Liste des articles**

Rechercher



paire de jumelle  
**Disponible**  
 ce sont des jumelles : D  
**Créer par antpfister, se situe à Lausanne**



tondeuse  
**Emprunter**  
 c'est une tondeuse  
**Créer par antpfister, se situe à Lausanne**

Créer par Anthony Pfister - [anthony.pfister@eduvaud.ch](mailto:anthony.pfister@eduvaud.ch) - Projet TPI

**F**


#### 4.1.6 Développement emprunter un article

Pour la partie « emprunt d'un article » sur le site, j'ai d'abord créé la page emprunt Article qui est un formulaire où l'utilisateur est invité de renseigné les dates de début et de fin de son emprunt.

Première version visuelle de la page Emprunt Article :

HOME Liste des Articles

**Emprunt de l'article - tondeuse**



**Disponible**  
 c'est une tondeuse

Date de début :

Date de fin :

Par la suite, l'utilisateur appuie sur le bouton pour emprunter l'article. Les informations sont alors envoyées sur la page checkEmprunt qui vérifie si les dates sont conformes et que l'article est

bien disponible. Si ce n'est pas le cas, l'utilisateur est renvoyé sur la page du formulaire avec un message d'erreur lui indiquant qu'il ne peut pas emprunter cet article.

Si l'emprunt est correct il est enregistré dans la base de données. Le nombre d'article de l'utilisateur connecté et mis à jour avec une unités en plus et le statu de l'article change en indisponible.

#### 4.1.7 Développement lister les articles prêtés et empruntés d'un utilisateur

Créer le 2023-05-12

Localisation : Lausanne

Nombre d'article créer par l'utilisateur 4

Nombre d'emprunt que l'utilisateur à actuellement : 2

Sur la page profil du site. J'ai créé deux listes qui s'affichent en plus des information détaillés de l'utilisateur. La première liste affiche tous les articles que l'utilisateur est en train d'emprunter. Il peut voir la date de début et de fin de l'article, et peut cliquer sur la photo pour voir les détails sur la page « article ». Il dispose également d'un bouton qui supprime l'emprunt de l'article, cette partie se fait sur la page « suppEmprunt ».

##### liste d'article Emprunter



longue vue

Du 2023-05-24 jusqu'au 2023-05-26

Arrêter l'emprunt



tondeuse

Du 2023-05-27 jusqu'au 2023-05-28

Arrêter l'emprunt

La deuxième liste affiche quant à elle tous les articles créés par l'utilisateur, avec leur photo, noms et statuts. L'utilisateur a la possibilité de voir les détails de l'article en cliquant sur la photo. J'ai fait une modification entre la maquette que j'ai faite dans la partie analyse au niveau des boutons. J'ai choisi de remplacer le bouton « ajouter un article » qui se trouve toujours sur la page liste article par un bouton « se Déconnecter ». Pour des raisons de cohérence et pour faciliter l'utilisation du site.

##### liste de article créer



paire de jumelle  
Disponible



tondeuse  
Emprunter



voiture  
Disponible



Arrosoirs  
Disponible

Créer par Anthony Pfister - [anthony.pfister@eduyvaud.ch](mailto:anthony.pfister@eduyvaud.ch) - Projet TPI

php8

#### 4.1.8 Développement système de notification

Pour le Développement du système de notification, j'ai ajouté un bandeau qui informe l'utilisateur quand un de ces emprunts est à moins de 2 jours de la date de fin.

Premier visuel du bandeau de notification :



L'activation de cette notification se fait via une fonction en javascript qui rend visible le conteneur du message quand elle est appelée.

Code javascript de la fonction et de l'appel de la fonction :

```
<script>
function activerConteneur() {
    var conteneur = document.getElementById('notifMessage');
    conteneur.style.display = 'block';
}
activerConteneur();
</script>
```

J'ai également créé en même temps une condition qui fait que si la date de fin d'un emprunt est la même ou avant la date actuelle. Alors l'emprunt est supprimé dans la page « suppEmprunt ».

#### 4.1.9 Développement communication SQL avec la base donnée

#### 4.1.10 Mise en place de Selenium

- Cette partie permet de reproduire ou reprendre le projet par un tiers.
- Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :
  - Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)
  - Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)
  - Code source commenté des éléments logiciels développés.
  - Modèle physique d'une base de données.
  - Arbres des documents produits.
- Il faut décrire le parcours de réalisation et justifier les choix.

### 4.2 Modifications

- Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.
- Date, raison, description, etc.

## 5 TESTS

### 5.1 Dossier des tests

- On dresse le bilan des tests effectués (qui, quand, avec quelles données...) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats obtenus et les actions à entreprendre en conséquence (et une estimation de leur durée).
- Si des tests prévus dans la stratégie n'ont pas pu être effectués :
  - raison, décisions, etc.
- Liste des bugs répertoriés avec la date de découverte et leur état:
  - Corrigé, date de correction, corrigé par, etc.

## 6 CONCLUSION

### 6.1 Bilan des fonctionnalités demandées

- Il s'agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.
- Si ce n'est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu'il reste à accomplir pour terminer le tout.

### 6.2 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 différences entre les planifications initiales et détaillées avec le journal de travail.

### 6.3 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-ce que ce projet m'a appris ?
- Suite à donner, améliorations souhaitables, ...
- Remerciements, signature, etc.

## 7 DIVERS

### 7.1 Journal de travail

Voir annexe.

### 7.2 Bibliographie

- Références des livres, revues et publications utilisés durant le projet.

## 7.3 Webographie

➤ Références des sites Internet consultés durant le projet.

- <https://waytolearnx.com/2019/07/comment-verifier-le-format-dune-date-en-php.html>
- [https://fr.wikipedia.org/wiki/Merise\\_\(informatique\)](https://fr.wikipedia.org/wiki/Merise_(informatique))
- <https://fr.freepik.com/>
- <https://github.com/Antpfister/>
- <https://www.codexworld.com/how-to/sort-results-order-by-best-match-using-like-in-mysql/>
- <https://www.primfx.com/tuto-php-barre-recherche-337/>
- <https://openclassrooms.com/forum/sujet/mettre-un-array-en-session-59486>
- <https://waytolearnx.com/2019/07/comment-ajouter-des-jours-a-une-date-en-php.html>
- 

## 8 ANNEXES

- Listing du code source (partiel ou, plus rarement complet)
- Guide(s) d'utilisation et/ou guide de l'administrateur
- Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).
- Extraits de catalogue, documentation de fabricant, etc.