

# Examen TPI - Gestion de prêt entre voisins

---



*FIGURE 1 IMAGE FREEPIK DE LA PAGE DE TITRE*



*FIGURE 2 LOGO SELENIUM IDE POUR PAGE DE TITRE*

Anthony Pfister  
CIN4A  
ETML  
Le 08.05.2023  
89h  
Gaël Sonney  
Daniel Berney  
Benjamin Wolf

# Table des matières

|          |   |           |
|----------|---|-----------|
| <b>I</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    | Fonctionnalités requises (du point de vue de l'utilisateur) .....                   | 4         |
| 1.5.2    | Contraintes .....   | 5         |
| 1.5.3    | Travail à réaliser par l'apprenti .....   | 5         |
| 1.5.4    | Méthodes de validation des solutions .....  | 5         |
| 1.6      | LES POINTS SUIVANTS SERONT ÉVALUÉS .....  | 5         |
| 1.7      | VALIDATION ET CONDITIONS DE RÉUSSITE .....  | 5         |
| 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 .....  | 9         |
| 3.2.4    | Plan de gestion des branches GIT .....  | 17        |
| 3.2.5    | Analyse des tables à créer dans la base des données : .....                         | 17        |
| 3.3      | CONCEPTION DES TESTS .....  | 18        |
| 3.3.1    | Test automatique avec Sélénium.....   | 18        |
| 3.3.2    | Test de fonctionnalité générale et final : .....                                    | 19        |
| 3.4      | PLANIFICATION DÉTAILLÉE .....   | 19        |
| <b>4</b> | <b>RÉALISATION .....</b>  | <b>19</b> |
| 4.1      | DOSSIER DE RÉALISATION .....  | 19        |
| 4.1.1    | Génération de la base de données du site, schémas MCD, MLD, MPD .....               | 19        |
| 4.1.2    | Développement des premières pages de base du site .....                             | 22        |
| 4.1.3    | Développement de la gestion de l'authentification .....                             | 23        |
| 4.1.4    | Développement Opération CRUD, Ajout, modifier, détail et supprimer un article ..... | 24        |
| 4.1.5    | Développement Barre de recherche liste d'article .....                              | 27        |
| 4.1.6    | Développement emprunter un article .....  | 28        |
| 4.1.7    | Développement lister les articles prêtés et empruntés d'un utilisateur .....        | 29        |
| 4.1.8    | Développement système de notification .....   | 29        |
| 4.1.9    | Développement communication SQL avec la base donnée .....                           | 30        |
| 4.1.10   | Mise en place de Selenium .....   | 33        |
| <b>5</b> | <b>TESTS.....</b>   | <b>34</b> |
| 5.1      | DOSSIER DES TESTS .....   | 34        |
| 5.1.1    | Bilan tests avec Selenium IDE.....  | 34        |
| 5.1.1    | Bilan des tests de fonctionnalités générales et finaux : .....                      | 35        |
| <b>6</b> | <b>CONCLUSION .....</b>   | <b>36</b> |
| 6.1      | BILAN DES FONCTIONNALITÉS DEMANDÉES.....  | 36        |
| 6.2      | BILAN DE LA PLANIFICATION.....  | 37        |
| 6.3      | BILAN PERSONNEL .....   | 37        |
| <b>7</b> | <b>DIVERS.....</b>  | <b>38</b> |

|          |                         |           |
|----------|-------------------------|-----------|
| 7.1      | JOURNAL DE TRAVAIL..... | 38        |
| 7.2      | BIBLIOGRAPHIE.....      | 38        |
| 7.3      | WEBOGRAPHIE.....        | 39        |
| <b>8</b> | <b>ANNEXES.....</b>     | <b>40</b> |

# I SPÉCIFICATIONS

## I.1 Titre

Gestion de prêt entre voisins

## I.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.

## I.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)

## I.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)

## I.5 Cahier des charges

### *1.5.1 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

### 1.5.2 Contraintes

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

### 1.5.3 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 2 fois 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)

### 1.5.4 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,

### 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 laquelle 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 des 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 documenté sur chaque tâche à implémenter. Je crée également une liste de tests 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 quelques minutes pour me renseigner spécifiquement sur sa 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 créerai 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 réalisées. Par la suite, je fais le bilan de ma gestion de mon projet, les défauts et les ratures. De voir quel serai les choses a améliorer pour un futur projet.

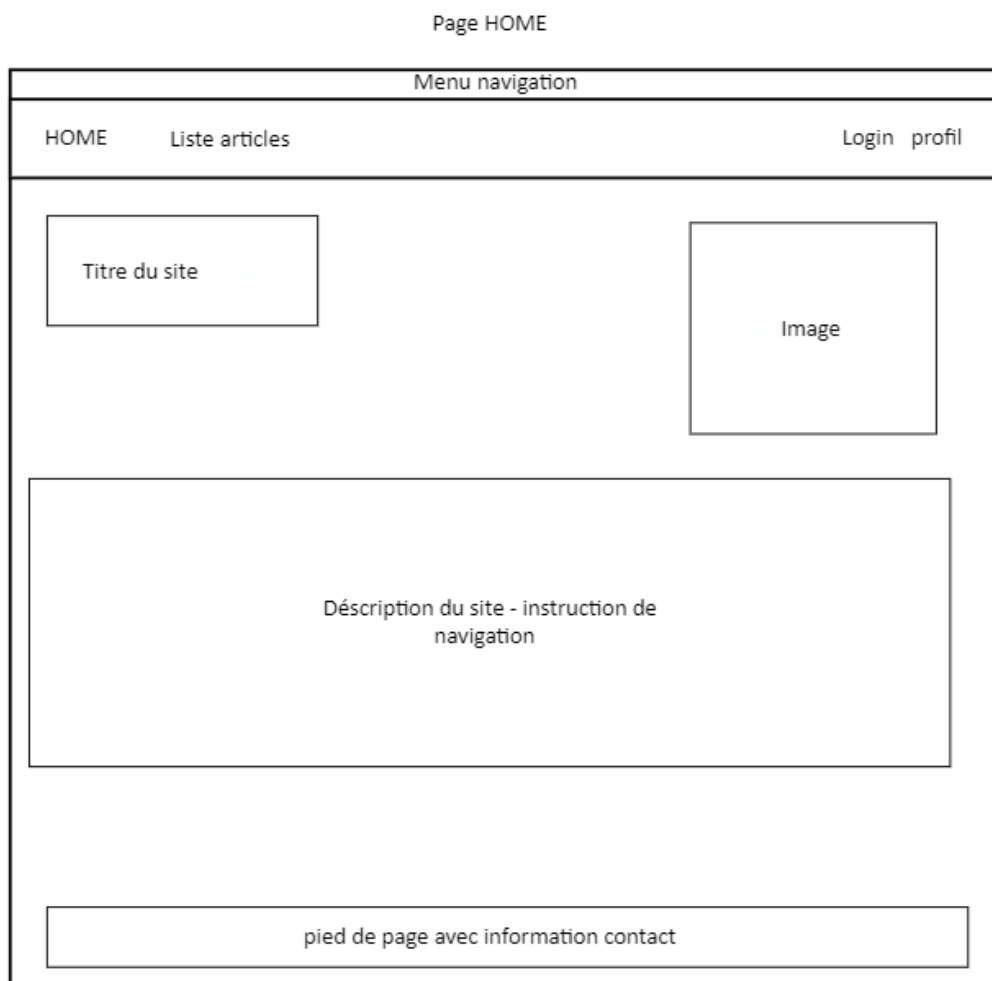
### 3.2.2 Liste de page de code a réalisé durant le projet :

- 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 userDetail - 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

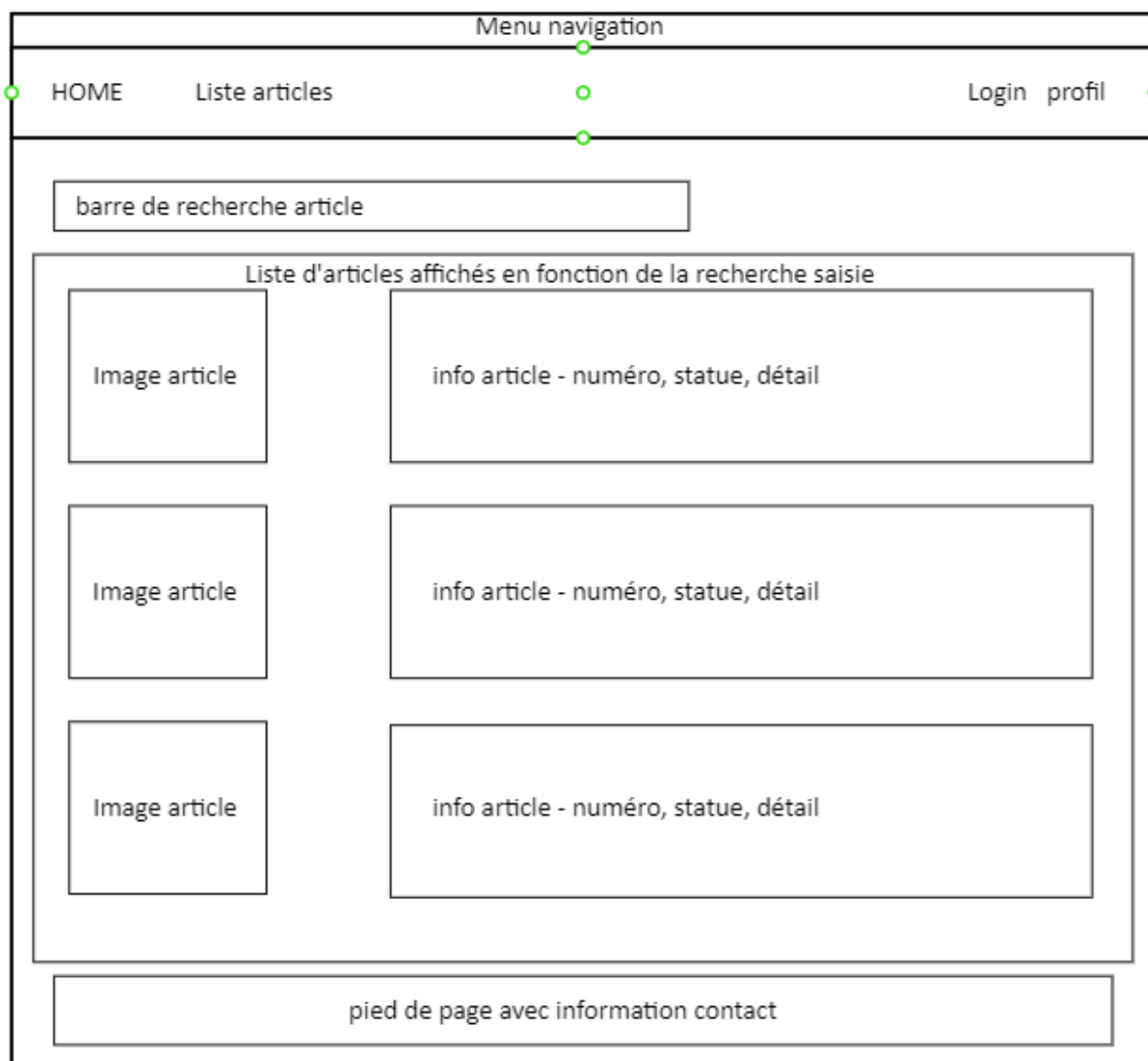


**FIGURE 3 MAQUETTE DE LA PAGE HOME DU SITE**

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



**FIGURE 4 MAQUETTE DE LA PAGE LISTE D'ARTICLES DU SITE**

Maquette de la page de la liste d'articles à emprunter. Tous les articles du site sont listés avec leurs informations. Avec une barre de recherche qui tri la liste en fonction de la localisation de l'utilisateur qui l'a créé. On retrouve également le menu de navigation et le pied de page.

- Page article

Page Article

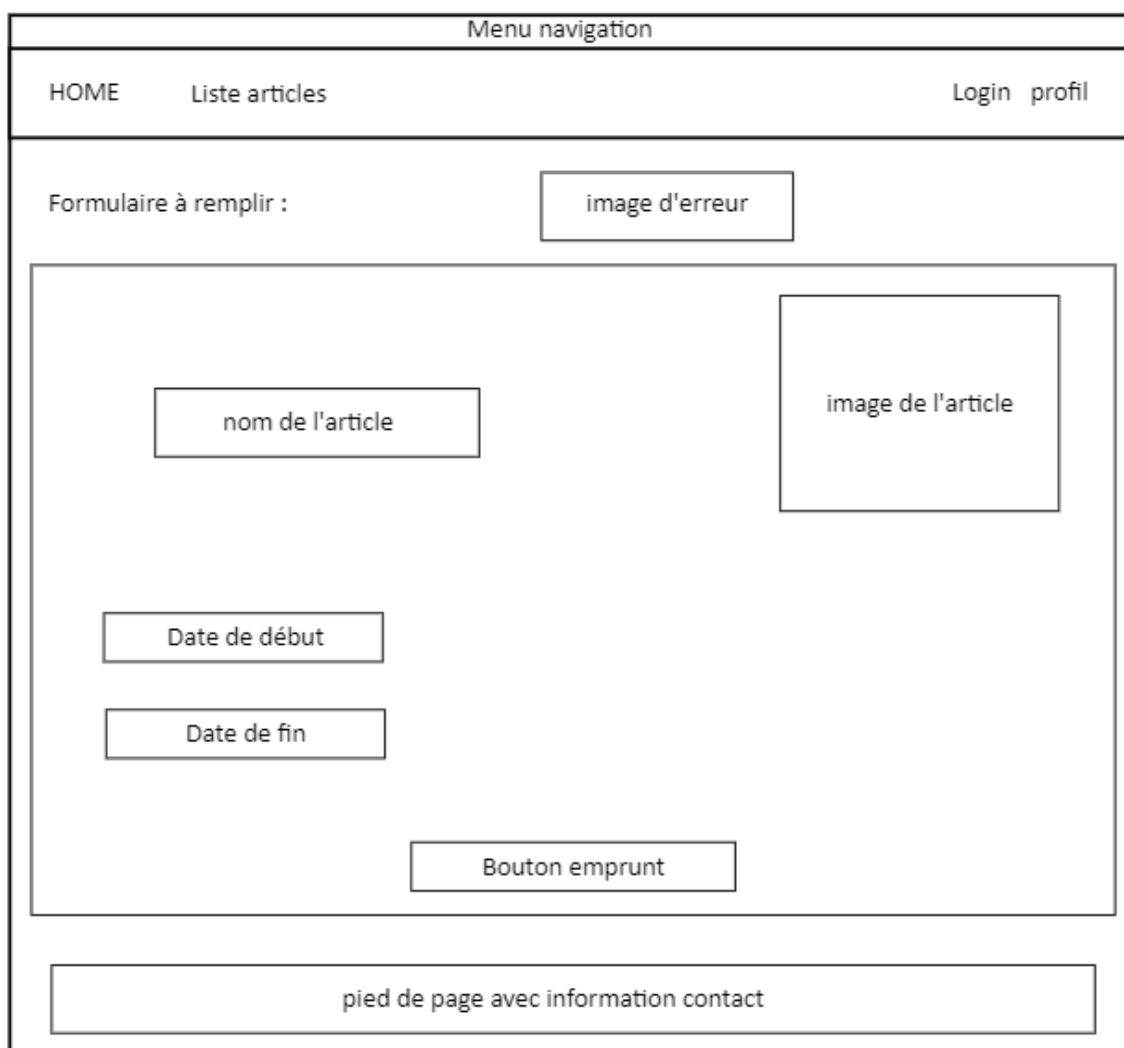
| Menu navigation  |                |              |
|--|----------------|--------------|
| HOME   | Liste articles | Login profil |
| <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 30%;"> <p>Information Article :</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Nom</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Auteur de l'article</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">lieu</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">statue - disponible/indisponible</div> <div style="border: 1px solid black; padding: 10px; margin-top: 10px; min-height: 100px;">Description de l'article</div> </div> <div style="width: 30%; text-align: center;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">image d'erreur</div> <div style="border: 1px solid black; padding: 10px; margin-top: 10px; min-height: 100px;">image de l'article</div> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px; width: 30%;">Bouton Emprunter l'article</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Bouton Modifier l'article</div> <div style="border: 1px solid black; padding: 5px; width: 30%;">Bouton supprimer l'article</div> </div> |                |              |
| <div style="border: 1px solid black; padding: 5px; min-height: 30px;">pied de page avec information contact</div>  |                |              |

**FIGURE 5 MAQUETTE DE LA PAGE ARTICLE DU SITE**

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 status disponible/indisponible. Si le statue 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 à accès au bouton « modifié » et « supprimé » pour 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

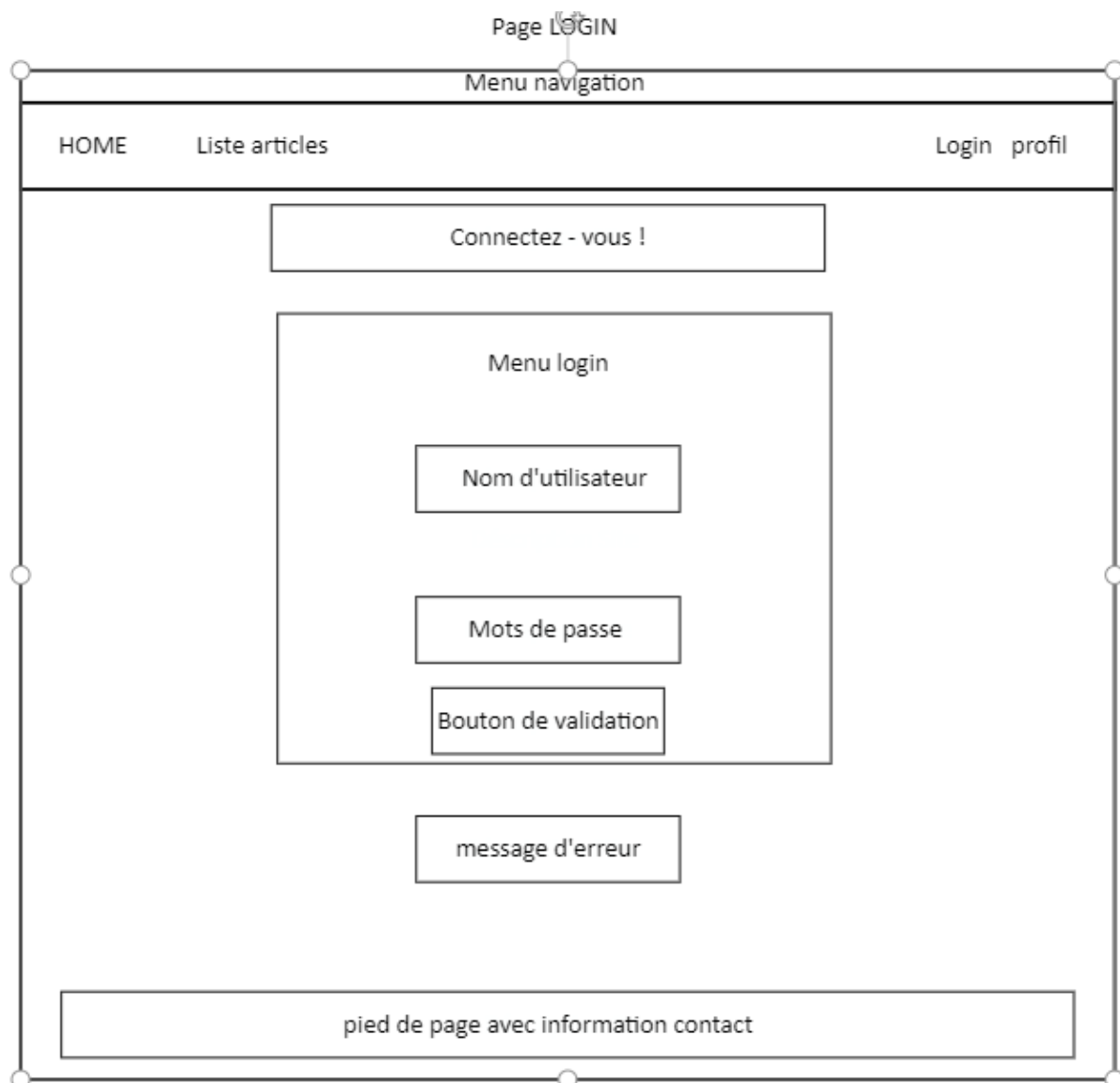
Bouton emprunt

pied de page avec information contact

**FIGURE 6 MAQUETTE DE LA PAGE EMPRUNT DU SITE**

Maquette de la page emprunt d'article. C'est un formulaire avec les dates de début et de fin à remplir. Ensuite, l'utilisateur appuis 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



**FIGURE 7 MAQUETTE DE LA PAGE LOGIN DU SITE**

Maquette de la page de login. L'utilisateur doit renseigner son nom d'utilisateur ainsi que son mot 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                             |                |   |         |
| <div style="border: 1px solid black; padding: 5px; display: inline-block;">Nom utilisateur</div>      |                | <div style="border: 1px solid black; padding: 5px; display: inline-block;">Bouton ajout article</div> |         |
| <div style="border: 1px solid black; padding: 5px; text-align: center;">information utilisateur</div> |                |   |         |
| Liste articles emprunter en cours avec information  |                |   |         |
| Article   | Article        | Article   | Article |
| Liste articles créer avec information   |                |   |         |
| Article   | Article        | Article   | Article |
| <div style="border: 1px solid black; padding: 5px;">pied de page avec information contact</div>       |                |   |         |

**FIGURE 8 MAQUETTE DE LA PAGE USER DÉTAIL DU SITE**

Maquette de la page de profil d'un utilisateur connecté. 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

**FIGURE 9 MAQUETTE DE LA PAGE AJOUT D'ARTICLE DU SITE**

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 rempli les champs de saisie, le message d'erreur apparaît 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

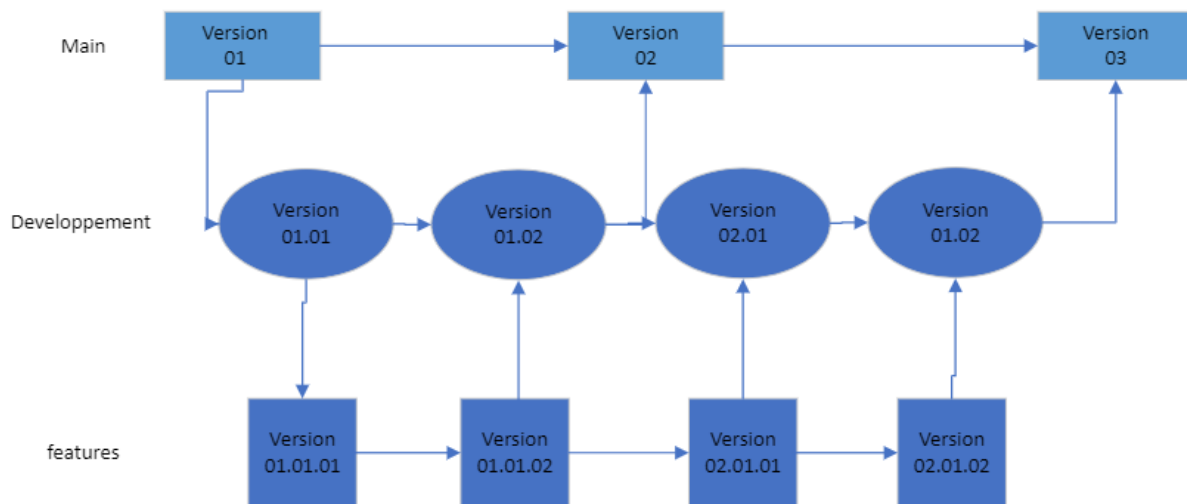
pied de page avec information contact

**FIGURE 10 MAQUETTE DE LA PAGE MODIFICATION D'ARTICLE DU SITE**

Maquette de la page de modification d'un article. L'utilisateur remplit le nouveau nom, insert 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



**FIGURE 1 | 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 versions principales et utilisables du code du site web. Les branches « développement » sont prévues pour les ajouts des nouvelles fonctionnalités finies. Les dernières branches de développement sont prévues pour les fonctionnalités en cours de développement et incomplètes.

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

### 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. S'identifier sur le site : ce test automatisé saisira les informations de connexion d'un utilisateur pour accéder aux fonctionnalités du site sur la page « login ». Ce test a pour but de vérifier le formulaire de connexion sur le site.
2. 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 téléchargé depuis le même répertoire du test, une description de l'article et un statut en mode « disponible ».
3. 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 ».
4. 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.
5. 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 et de voir si l'information de l'utilisateur est bien mise à jour.
6. 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.
7. 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'emprunts de l'utilisateur sont correctes.

8. Refaire une dernière fois l'emprunt de l'article précédemment supprimé : ce test automatisé va recréer le même emprunt avec les mêmes dates que l'emprunt précédemment supprimé.
  9. 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 créé, et de voir s'il arrive à le trouver même si le lieu n'a pas été écrit complètement.
  10. Supprimer l'article : ce test automatisé va supprimer l'article entièrement avec l'emprunt en cliquant sur le bouton « supprimé 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.
11. Se déconnecter du site : ce test automatisé va déconnecter l'utilisateur du site.

Les données que les tests automatiques vont être différents pour chaque exécution

### 3.3.2 Test de fonctionnalité générale et final :

Les tests à effectuer à la fin de la réalisation du projet sont là pour vérifier manuellement toutes les fonctionnalités demandées dans le cahier des charges pour les validées. Ils suivront cette liste :

1. S'identifier avec un utilisateur sans problème rencontrer. Voir si le message d'erreur est fonctionnel
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 recherche 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.
9. Voir si les tests automatiques créés dans Selenium IDE sont fonctionnel.

## 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 :

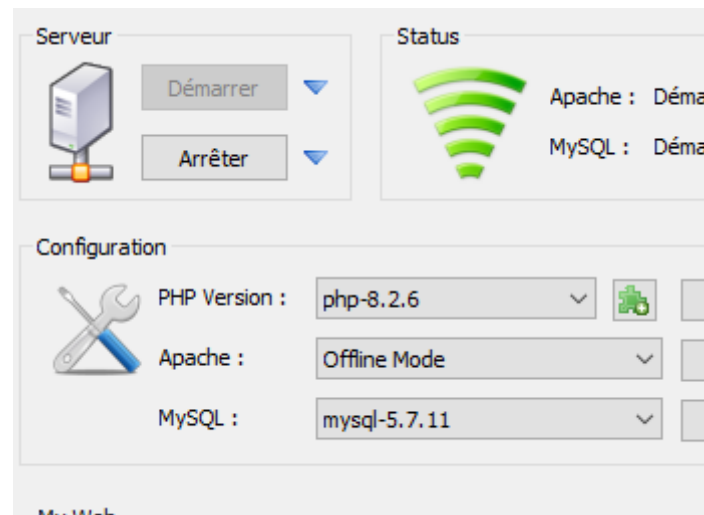


FIGURE 12 CAPTURE D'ÉCRAN DE LA CONFIGURATION DE SERVEUR UWAMP

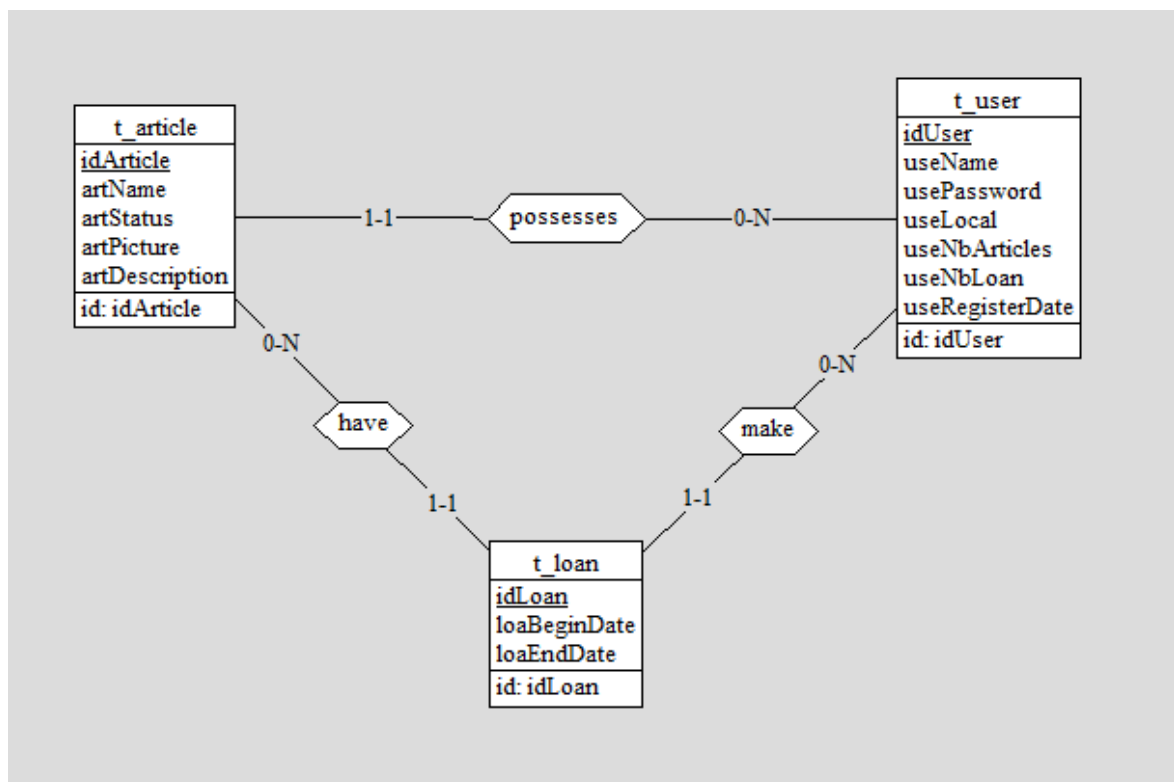


FIGURE 13 LE SCHÉMA MCD DE LA BASE DE DONNÉES

J'ai d'abord créé un schéma MCD sur le logiciel DB\_MAIN pour visualiser les tables que j'avais prévues 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 ».

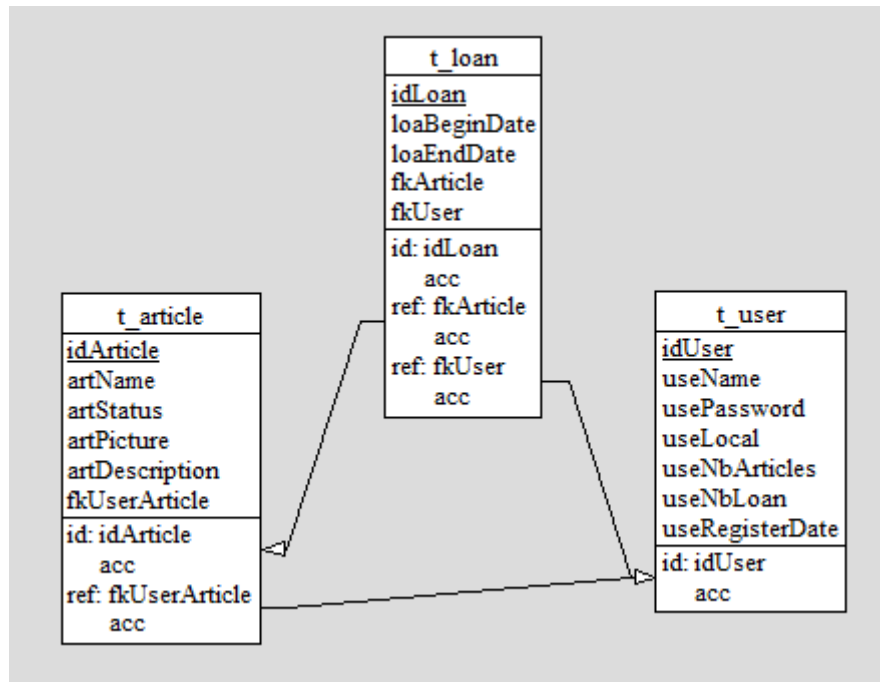


FIGURE 14 SCHÉMA MLD DE LA BASE DE DONNÉES

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.

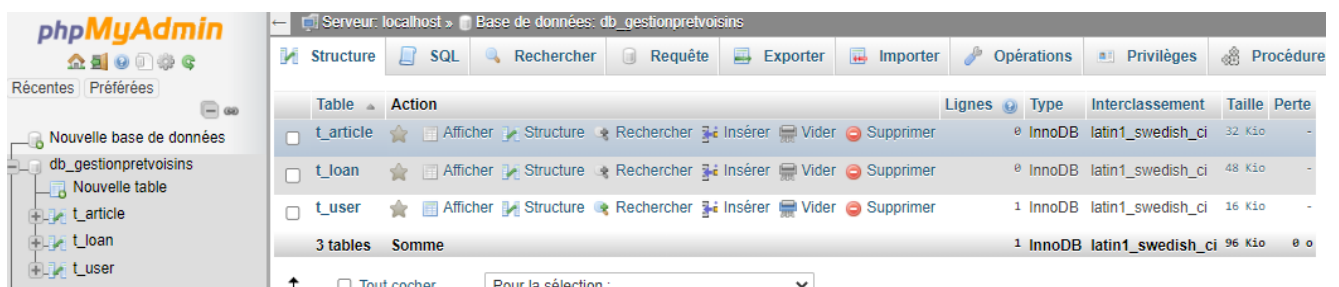


FIGURE 15 IMPORTATION DANS LA BASE DE DONNÉES ET VISUALISATION DANS PHPMYADMIN :

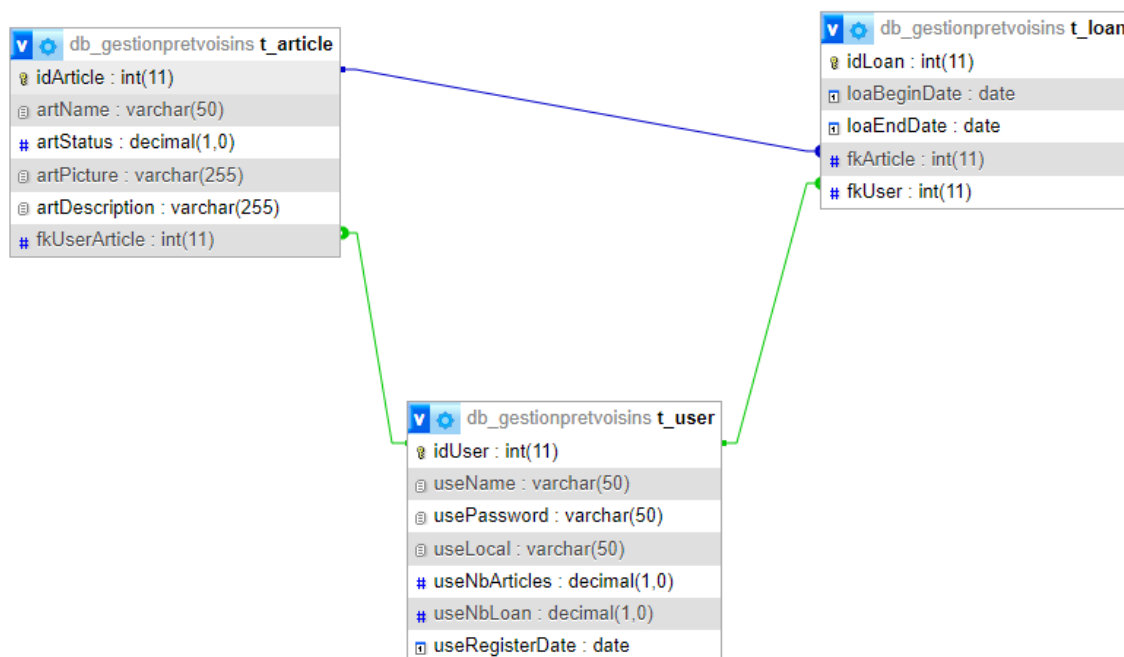


FIGURE 16 SCHÉMA DE LA BASE DE DONNÉES DANS LE CONCEPTEUR DE PHPMYADMIN

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

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

[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

**FIGURE 17 PREMIÈRE VERSION VISUELLE DE LA PAGE HOME SANS CODE CSS :**

Comme sur la maquette que j'ai créée 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 regroupe 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 « userDetail ».

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

## Connecter-Vous !!

## Renseigner vos information de connexion svp !

|                   |
|-------------------|
| Nom d'utilisateur |
| Mot de passe      |
| S'identifier      |

Créer par Anthony Pfister - Projet TPI

**FIGURE 18 PREMIÈRE VERSION VISUELLE DE LA PAGE LOGIN SANS CODE CSS**

Première version visuelle de la page « userDetail » 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 mot de passe (usePassword). Si l'utilisateur n'a pas saisi ses informations ou si elles sont fausses, alors la page est recharger avec un message en plus qu'il lui indique une erreur.

[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

**FIGURE 19 PAGE LOGIN SI L'UTILISATEUR A COMMIS UNE ERREUR DANS LA SAISIE DU FORMULAIRE DE CONNEXION**

Cependant si utilisateur a renseigné comme il faut ses données, alors utilisateur se retrouver sur la page home en étant connecté 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 sont corrects ou renvoie une erreur à la page login qui affiche un message d'erreur.

Cette partie de réalisation est enregistré 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 effectuée 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 rentrées sont correctes pour être entrées dans la base de données 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 saisie des informations.

Dans le cas où les informations seraient correctes, la date et l'heure est rajouté 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 source. Ensuite, les données sont enregistrées 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'articles par l'utilisateur connecté est mis à jour avec une unité en plus.

Cette page est accessible que depuis la page liste article et seulement si l'utilisateur est connecté.

[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

FIGURE 20 PREMIÈRE VERSION VISUELLE DE LA PAGE ADDARTICLE 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 leurs 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.

FIGURE 21 PREMIÈRE VERSION VISUELLE DE LA PAGE ARTICLELISTE SANS CODE CSS

Sur cette page d'info de l'article, l'utilisateur peut voir 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 développé 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)

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

## Article info - paire de jumelle



Disponible

c'est non

Créer par antpfister, se situe à Lausanne

[Supprimer l'article](#)  
[Modifier cette article](#)

**FIGURE 22 PREMIÈRE VERSION VISUELLE DE LA PAGE ARTICLE SANS CODE CSS**

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 remplit rien alors l'article n'est pas modifié et l'utilisateur est redirigé vers la page article. Les champs sont remplis avec les données initiales de l'article.

Ensuite, les informations remplies sont vérifiées 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 ajoutée. Ensuite toutes les nouvelles données sont envoyées et modifiées dans la ligne de la table article qui correspond via une requête SQL via la page Database. (Voir point 4.1.9). Si le statut d'un article modifié de « emprunter » à « disponible », alors l'emprunt lié à cet article est supprimé.

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 choix a été fait pour des raisons de cohérence du site.)

[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 statut de l'article : ☒ disponible ☐ indisponible

Créer par Anthony Pfister - Projet TPI

**FIGURE 23 PREMIÈRE VERSION VISUELLE DE LA PAGE MODIFARTICLE SANS CODE CSS**

#### 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 rempli, la page checkRechercheBarre renvoie un message d'erreur invitant l'utilisateur à bien remplir la barre

Page Liste des Articles avec la barre :

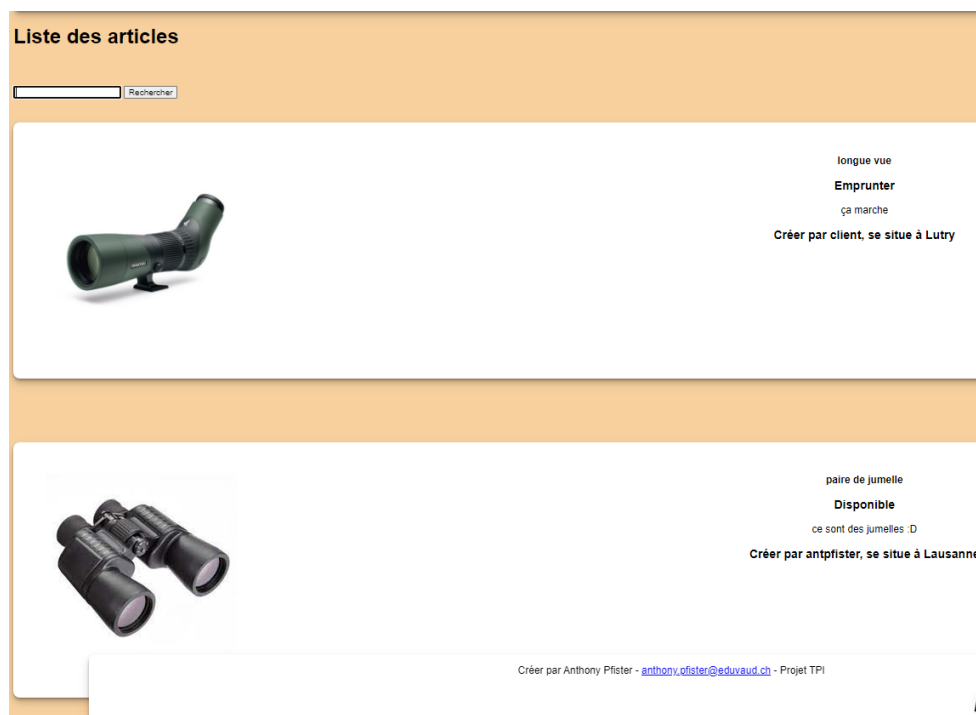


FIGURE 24 PAGE LISTE DES ARTICLES AVEC LA BARRE

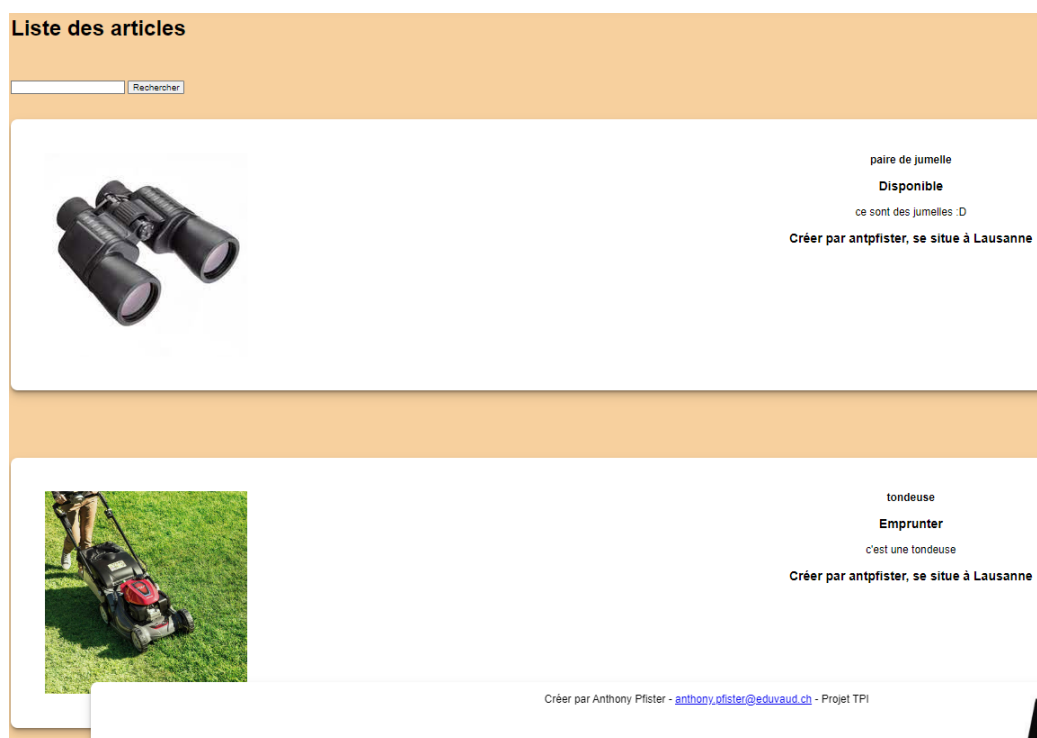
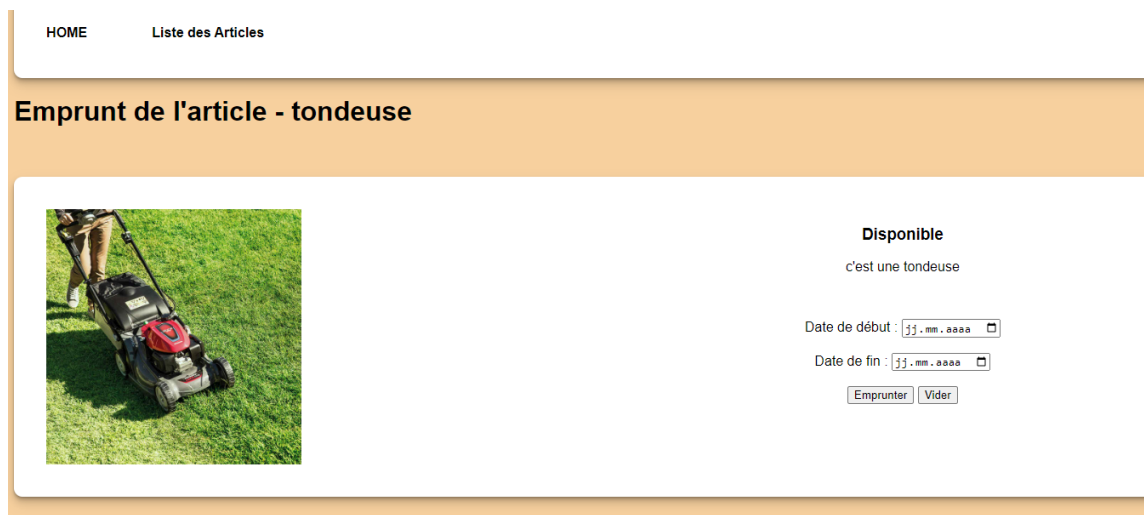


FIGURE 25 PAGE DE LISTE DES ARTICLES AVEC LA RECHERCHE « LAUSANNE » ENTRÉ


#### 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 renseigner les dates de début et de fin de son emprunt.



HOME    Liste des Articles

### Emprunt de l'article - tondeuse



**Disponible**  
c'est une tondeuse

Date de début :

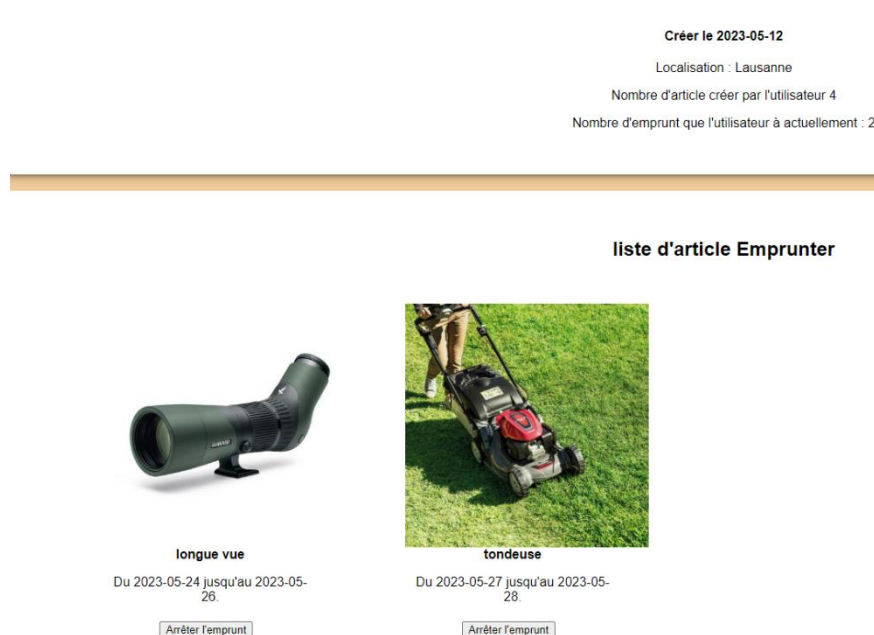
Date de fin :

**FIGURE 26 PREMIÈRE VERSION VISUELLE DE LA PAGE EMPRUNT ARTICLE**

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, c'est-à-dire si la date du début et de fin saisie n'est pas déjà passé par rapport à la date actuelle, et que la date du début est bien avant la date de fin pour des raisons de cohérences du site, et qu'également l'article soit 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'articles de l'utilisateur connecté et mis à jour avec une unité en plus et le statu de l'article change en indisponible.

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

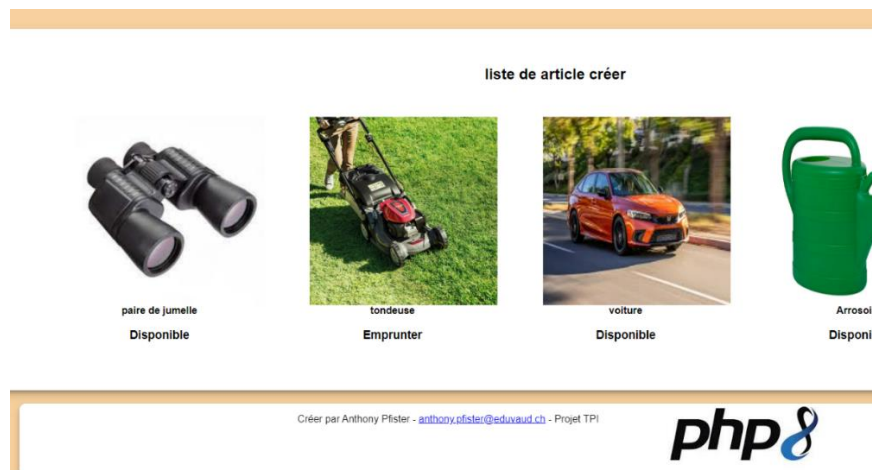


Su Sur la page profil du site. J'ai créé deux listes qui s'affichent en plus des informations détaillées de l'utilisateur. La première liste affiche tous les articles que l'utilisateur est en train d'emprunter. Il peut voir la date de début et de fin de l'article, et peut cliquer sur la photo pour voir 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 ».

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

**FIGURE 27 VISUELLE DE PAGE USERDÉTAIL AVEC LA LISTE DES EMPRUNT DE L'UTILISATEUR**

voir les détails de l'article en cliquant sur la photo. J'ai fait une modification avec la maquette que j'ai faites 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.



**FIGURE 28 VISUELLE DE PAGE USERDÉTAIL AVEC LA LISTE DES ARTICLES CRÉÉ DE L'UTILISATEUR**

#### 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.



FIGURE 29 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.

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

FIGURE 30 CODE JAVASCRIPT DE LA FONCTION ET DE L'APPEL DE LA FONCTION

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 ». Cette condition permet d'empêcher les incohérences sur le site.

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

Je communique avec la base de données en me connectant à l'utilisateur lié à la base de données du site dans phpMyAdmin. Cette connexion se fait automatiquement dans la page « Database » via les informations stocker sur la page « config ».

C'est sur la page « Database » que toutes les méthodes pour insérer, modifier, supprimer et afficher les données se trouvent.

```

/**
 * créer l'article dans la base de données
 * @param $artname-> nom de l'article
 * @param $artstatus-> status de l'article
 * @param $artimage-> nom de l'image de l'article
 * @param $artdescription-> description de l'article
 * @param $artuser-> l'identifiant de l'utilisateur qui l'a créé
 */
0 references | 0 overrides
public function insertArticle($artName,$artstatus,$artimage,$artdescription,$artuser) {
    $query = "INSERT INTO t_article(artName,artStatus,artPicture,artDescription,fkUserArticle)
    VALUES(:artName,:artstatus,:artimage,:artdescription,:artuser)";

    $binds = array(
        0 => array(
            'marker' => 'artName',
            'value' => $artName,
            'type' => PDO::PARAM_STR
        ),
        1 => array(
            'marker' => 'artstatus',
            'value' => $artstatus,
            'type' => PDO::PARAM_INT
        ),
        2 => array(
            'marker' => 'artimage',
            'value' => $artimage,
            'type' => PDO::PARAM_STR
        ),
        3 => array(
            'marker' => 'artdescription',
            'value' => $artdescription,
            'type' => PDO::PARAM_STR
        ),
        4 => array(
            'marker' => 'artuser',
            'value' => $artuser,
            'type' => PDO::PARAM_INT
        ),
    );

    $req = $this->queryPrepareExecute($query, $binds);

    $this->unsetData($req);
}

```

FIGURE 31 LA MÉTHODE POUR INSÉRER UN ARTICLE DANS LA BASE DE DONNÉES VIA REQUÊTE SQL

```

/**
 * retourne toutes les données de tous les articles dans la base de données avec les données de l'utilisateur qui la créé.
 * @return -> sous forme de tableau,
 * toutes les données de tous les articles de la base de données avec les données de l'utilisateur qui la créé.
 */
0 references | 0 overrides
public function getAllArticlesAndInfos(){
    $req = $this->querySimpleExecute('SELECT * FROM t_article
    INNER JOIN t_user ON t_article.fkUserArticle = t_user.idUser');
    $result = $this->formatData($req);

    $this->unsetData($req);

    return $result;
}

```

FIGURE 32 LA MÉTHODE POUR AFFICHER TOUTES LES DONNÉES DES ARTICLES ET DES UTILISATEURS

```

/**
 * Retourne toutes les données de l'article par rapport à l'identifiant de l'article donné.
 * @param $id -> l'identifiant de l'article
 * @return -> toutes les données de l'article sous forme de tableau
 */
0 references | 0 overrides
public function getArticle($id){
    $query = 'SELECT * FROM t_article
    INNER JOIN t_user ON t_article.fkUserArticle = t_user.idUser
    WHERE idArticle = :id';

    $binds = array(
        0 => array(
            'marker' => 'id',
            'value' => $id,
            'type' => PDO::PARAM_INT
        )
    );

    $req = $this->queryPrepareExecute($query, $binds);
    $result = $this->formatData($req);

    $this->unsetData($req);

    return $result[0];
}

```

**FIGURE 33 LA MÉTHODE POUR AFFICHER UN SEUL ARTICLE AVEC SON UTILISATEUR**

```

/**
 * créer l'article dans la base de données
 * @param $artname-> nom de l'article
 * @param $artstatus-> status de l'article
 * @param $artimage-> nom de l'image de l'article
 * @param $artdescription-> description de l'article
 * @param $artuser-> l'identifiant de l'utilisateur qui l'a créé
 */
0 references | 0 overrides
public function insertArticle($artName,$artstatus,$artimage,$artdescription,$artuser) {
    $query = "INSERT INTO t_article(artName,artStatus,artPicture,artDescription,fkUserArticle)
    VALUES(:artName,:artstatus,:artimage,:artdescription,:artuser)";

    $binds = array(
        0 => array(
            'marker' => 'artName',
            'value' => $artName,
            'type' => PDO::PARAM_STR
        ),
        1 => array(
            'marker' => 'artstatus',
            'value' => $artstatus,
            'type' => PDO::PARAM_INT
        ),
        2 => array(
            'marker' => 'artimage',
            'value' => $artimage,
            'type' => PDO::PARAM_STR
        ),
        3 => array(
            'marker' => 'artdescription',
            'value' => $artdescription,
            'type' => PDO::PARAM_STR
        ),
        4 => array(
            'marker' => 'artuser',
            'value' => $artuser,
            'type' => PDO::PARAM_INT
        )
    );

    $req = $this->queryPrepareExecute($query, $binds);

    $this->unsetData($req);
}

```

**FIGURE 34 LA MÉTHODE POUR MODIFIER L'ARTICLE**



```

/**
 * Retourne tous les article de la base de données par rapport à la localisation de l'utilisateur qui l'a créé.
 * @param $barrRecherche -> nom du lieu rechercher.
 * @return -> tous les identifiant des articles trouvés.
 */
0 references | 0 overrides
public function searchlocal($barrRecherche){
    $req = $this->querySimpleExecute('SELECT idArticle FROM t_article
    INNER JOIN t_user ON t_article.fkUserArticle = t_user.idUser
    WHERE useLocal LIKE "%'.$barrRecherche.'%"');

    $result = $this->formatData($req);







    $this->unsetData($req);

    return $result;
}

```

FIGURE 35 LA MÉTHODE POUR RECHERCHER UN ARTICLE VIA LA BARRE DE RECHERCHE

#### 4.1.10 Mise en place de Selenium

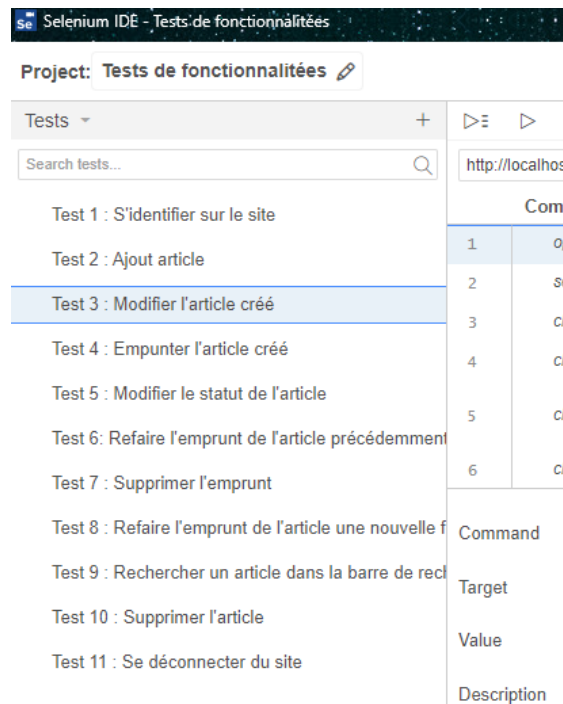
|   |   |
|---|---|
| Paramètres du site  |    |
| Autoriser en navigation privée<br>Avertissement : Google Chrome ne peut pas empêcher les extensions d'enregistrer l'historique de votre navigation. Pour désactiver cette extension en mode navigation privée, désélectionnez cette option. |    |
| Autoriser l'accès aux URL de fichier  |  |
| Ouvrir le site Web de l'extension   |  |
| Afficher sur le Chrome Web Store  |  |
| Source<br>Chrome Web Store  |   |
| Supprimer l'extension   |  |

La mise en place des tests automatiques avec Selenium IDE c'est d'abord fait par son installation de l'extension pour Chrome. Avec l'aide du chef de projet, nous avons trouvé qu'il fallait autoriser l'extension à pouvoir ouvrir l'explorateur de fichier pour télécharger une photo durant les tests automatisés. Ce paramètre s'active en cochant le paramètre « Autoriser l'accès aux URL de fichier » dans les paramètres de gestion de l'extension.

J'ai ensuite créé la liste des tests automatisée qui s'appelle « Tests complet des fonctionnalités du site « Gestion de prêt entre voisins ». Et j'y ai créé à l'intérieur les tests qui étaient planifiés dans la partie test de l'analyse.

(3.3.1).

FIGURE 36 PARAMÈTRE DE L'EXTENSION DE SELENIUM IDE



**FIGURE 37 LA LISTES COMPLÈTES DES TESTS AUTOMATISÉE DANS SELENIUM**

J'ai cependant rencontré un problème au niveau du chemin de l'image qu'on veut uploader. Selenium IDE ne semble pas avoir de solution pour pouvoir mettre des chemins relatifs pour le fichier à télécharger. Donc les tests automatisés que j'ai créés téléchargent les images de test avec un chemin absolu. Cela pose un problème pour exécuter ces tests via une autre machine, car à chaque fois, il faut changer le chemin absolu de la localisation des images de test pour que cela fonctionne.

## 5 TESTS

### 5.1 Dossier des tests

#### 5.1.1 Bilan tests avec Selenium IDE

La liste des bilans des tests effectués durant la phase de tests consiste à exécuter tous les tests automatisés à la suite avec des articles et des informations différentes afin de voir s'il y a des éventuels bugs à corriger.

Bilan des tests effectuer :

| Numéro Tests et Date :                  | Résultat :  | Données utilisées :  |
|---|---|--|
| <b>1<sup>er</sup> test, 31.05.2023</b>  | Erreur au niveau de la modification de l'article et de la suppression de l'article                                    | Article : nom moto, image moto rouge, description taille moyenne, statu en disponible.<br>Modification : données similaires. |
| <b>2<sup>ème</sup> test, 02.06.2023</b> | Erreur au moment de supprimer l'emprunt car l'emprunt c'est fait supprimer automatiquement par la page « userDetail » | Article : nom moto, image moto rouge, description taille moyenne, statu en disponible.<br>Modification : données similaires. |

|   |              |   |
|---|--------------|---|
| <b>3<sup>ème</sup> test, 02.06.2023</b> | Pas d'erreur | Article : nom moto, image moto rouge, description taille moyenne, statu en disponible.<br>Modification : données similaires.  |
| <b>4<sup>ème</sup> test, 02.06.2023</b> | Pas d'erreur | Article : nom moto, image moto rouge, description taille élever, statu en indisponible.<br>Modification : données similaires. |

Liste des bugs corrigés :

- 1<sup>er</sup> test, 31.05.2023 : problème au niveau de la modification de l'image d'un article. Le bug a été fixé sur la page « checkmodifarticle ». Le bug était dû à une erreur d'incrémentation d'une variable.
- 1<sup>er</sup> test, 31.05.2023 : problème au niveau de la suppression de l'article avec un emprunt. Le bug a été fixé en mettant la suppression de l'emprunt en premier et ensuite la suppression de l'article.
- 2<sup>ème</sup> test, 02.06.2023 : problème au niveau de la suppression de l'emprunt car l'emprunt s'est fait supprimer automatiquement par la page « userDetail ». Le bug a été corrigé en modifiant les conditions de suppression de l'emprunt.

### 5.1.1 Bilan des tests de fonctionnalités générales et finaux :

Les tests de fonctionnalités générales et finaux, qui ont été planifiées dans la partie analyse, ont été réalisés à la suite des tests automatiques par Selenium. La procédure des tests est de constater manuellement toutes les fonctionnalités implémentées sur le site et les validées dans un tableau.

| Fonctionnalités testées :  | Résultat : |
|--|------------|
| <b>Gestion de l'authentification</b>   | Validées   |
| <b>Ajouter des articles.</b>   | Validées   |
| <b>Modifier des articles.</b>  | Validées   |
| <b>Supprimer des articles.</b>   | Validées   |
| <b>Afficher les détails de tous les articles enregistrer.</b>                        | Validées   |
| <b>Utiliser la barre de recherche pour trouver un article par rapport à son lieu</b> | Validées   |
| <b>Faire des emprunts d'articles.</b>  | Validées   |
| <b>Lister les articles prêtés et empruntés d'un utilisateur</b>                      | Validées   |
| <b>Vérifier si la barre de notification est fonctionnelle</b>                        | Validées   |
| <b>Mettre en place des test automatiques</b>   | Validées   |

## 6 CONCLUSION

### 6.1 Bilan des fonctionnalités demandées

Toutes les fonctionnalités demandées dans le cahier des charges ont été réalisées et tester comme cela était prévu depuis le début dans la partie analyse.

Pour la gestion de l'authentification, ce fut la première fonctionnalité implémentée sur le site. Réaliser de manière rapide et sans problème. L'utilisateur ne peut pas accéder aux fonctionnalisées du site s'il ne s'est pas connecté. Par la suite, cette fonctionnalité s'est avérée très utile pour rajouter des droits spécifiques au créateur d'un article. Car seulement le créateur d'un article peut supprimer ou modifier l'article qui l'a créé, ce qui n'était pas prévu au tout début du projet.

Pour les opérations CRUD, un utilisateur connecté peut ajouter un article avec le nom, une image, une description et un statu qu'il peut définir en « disponible » ou « indisponible ». Il peut modifier toutes les données de l'article qui la créé, et même arrêter de cette façon manuelle un emprunt en cours de son article en changeant le status. Il peut également supprimer les articles qui l'a créé, ce qui supprimera également l'emprunt s'il y en a un en cours. Et enfin, l'affichage des données de l'article est accessible depuis la liste des articles, mais aussi depuis le profil de l'utilisateur en cliquant sur la photo.

Pour la recherche d'article sur le site, une barre de recherche a été rajoutée sur la page de liste des articles. L'utilisateur peut rechercher un article en fonction de la localisation du créateur de l'article. La requête SQL liée à cette recherche est capable de trouver la localisation d'un article même si l'utilisateur n'a pas exactement écrit comme il faut le nom de la localisation. Ce qui est confortable pour l'utilisation de cette barre de recherche.

Pour l'emprunt d'un article, l'utilisateur pour emprunter un article qui est « disponible » uniquement, et choisir les dates de début et de fin de son emprunt. Les dates doivent être cohérentes entre elles et avec la date actuelle. Un emprunt qui a dépassé la date de fin se fait supprimer automatiquement et un bouton pour arrêter l'emprunt d'un article est disponible sur la page profil.

Utilisateur a accès sur la page profil à tous les articles qu'il a créé, et aux articles qu'il est en train d'emprunter. Il peut visualiser leurs données en détail en cliquant sur leur image.

Un système de notification a été ajouté sur la page profil. Dès qu'un emprunt est à moins de 2 jours de la date de fin, l'utilisateur est informé par un bandeau d'affichage qui le lui indique en haut de la page.

Dès qu'un emprunt est à moins de 2 jours de la date de fin, l'utilisateur est informé par un bandeau d'affichage qui le lui indique en haut de la page. Avec Selenium IDE, l'extension m'a permis de fabriquer ces tests. Le seul problème rencontré que je n'ai pas pu résoudre est le fait qu'on ne peut pas mettre pour le moment de chemin relatif pour le téléchargement des images dans les tests automatisés. Ce qui pose problème si les tests sont exécutés sur un pc différent du mien. Le seul moyen que j'ai trouvé et de modifier manuellement les chemins absolus de la localisation des images de test. Malheureusement, en parcourant les forums, je n'ai pas vu de solution à ce problème.

La gestion des versions avec Git a été globalement bien respectée. Il y a eu un manque d'opportunité pour l'utilisation de la branche « feature », car il y a eu rarement des moments où une fonctionnalité n'a pas été réalisée à la fin de la séquence. Et les noms des versions ne sont pas vraiment cohérents, il aurait mieux fallu appeler la première version dans « main » la 00.01.00 plutôt que la 01.00.00 car la première version n'est pas ce qu'on peut appeler une version « fonctionnelle » du site. La version finale du site sera finalement la 02.00.00 ce qui est presque incohérent.

## 6.2 Bilan de la planification

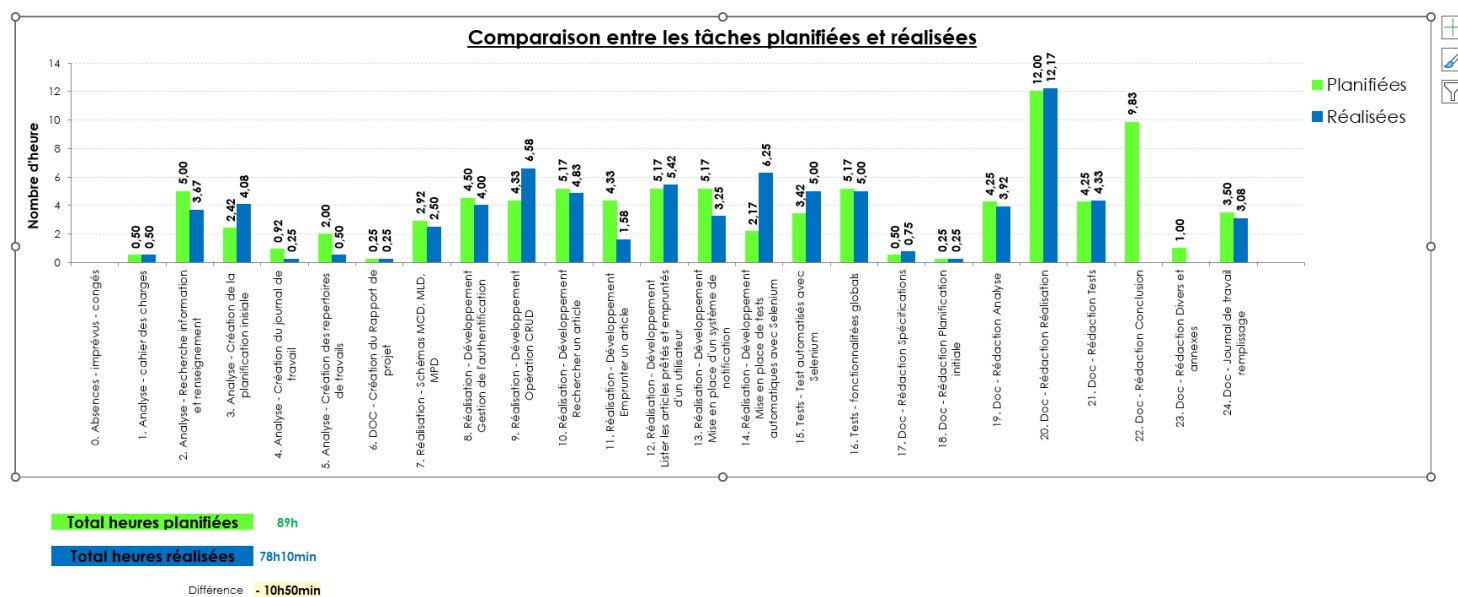


FIGURE 38 TABLEAU DE COMPARAISON ENTRE LA PLANIFICATION INITIALE ET LE JOURNAL DE TRAVAIL

(Cette capture d'écran de comparaison des tâches planifiées et réalisées a été faite le matin de la dernière journée de réalisation du projet)

L'ordre des tâches que j'avais prévu dès le début du projet dans la planification initiale a été respecté. La planification initiale a pris plus de temps que prévue à cause d'un problème de lisibilité en format PDF qu'il fallut régler. Ce temps pris a été rattrapé dans la création du journal de travail et de la création de répertoire de travail dans Github.

Pour la réalisation des fonctionnalités, le temps a été globalement bien respecté. Les opérations CRUD ont été un peu plus longues à réaliser que prévu, car cette partie englobe beaucoup de fonctionnalité (ajouter, modifier, supprimer, afficher) mais le temps a été rattrapé pour le système d'emprunt plus facile que prévu. La mise en place du système de tests automatiques avec Selenium a aussi pris du temps à cause du problème de chemin relatif des images de test que je n'ai pas pu résoudre.

Le temps pour remplir le journal de travail et à faire toutes les sauvegardes des avancements de mon projet à chaque fin de journée a été respecté à la lettre.

## 6.3 Bilan personnel

Personnellement, je suis plutôt satisfait du déroulement de ce projet en termes de timing. J'ai pu approfondir mes connaissances en développement Web et en gestion de base de données. J'ai pu pour la première fois utiliser Github pour la réalisation, et je me suis rendu compte de l'importance de l'outil pour ce genre de projet.

J'ai voulu faire un site Web en utilisant toutes les possibilités que j'avais à disposition (PHP, CSS, javascript) mais sans utiliser de Framework car je voulais approfondir mes connaissances en design avec le CSS.

Le site du projet est tout à fait utilisable et des améliorations sont envisageables sur le design, imaginées une fonctionnalité en plus pour le suivi d'un article, historique des utilisateurs qui a emprunté l'article par exemple.

Le rendu final du site est plutôt satisfaisant avec ce que j'avais imaginé à la base.

## 7 DIVERS

### 7.1 Journal de travail

Voir annexe.

### 7.2 Glossaire

- PHP : Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques.
- GIT : Git est un logiciel de gestion de versions décentralisé.
- Opération CRUD : (create, read, update, delete) (créer, lire, mettre à jour, supprimer) est un acronyme pour les façons dont on peut fonctionner sur des données stockées. C'est un moyen mnémotechnique pour les quatre fonctions de base du stockage persistant.
- Commit : capture un instantané des changements actuellement stagés du projet.
- SQL : est un langage informatique normalisé servant à exploiter des bases de données relationnelles.
- PDF : Format de fichier permettant de conserver les caractéristiques d'un document.
- JPEG : Format standardisé d'images numériques permettant de réduire la taille des fichiers.

### 7.3 Bibliographie

|  |           |
|--|-----------|
| <b>FIGURE 1 IMAGE FREEPIK DE LA PAGE DE TITRE</b>  | <b>1</b>  |
| <b>FIGURE 2 LOGO SELENIUM IDE POUR PAGE DE TITRE</b>   | <b>1</b>  |
| <b>FIGURE 3 MAQUETTE DE LA PAGE HOME DU SITE</b>   | <b>9</b>  |
| <b>FIGURE 4 MAQUETTE DE LA PAGE LISTE D'ARTICLES DU SITE</b>   | <b>10</b> |
| <b>FIGURE 5 MAQUETTE DE LA PAGE ARTICLE DU SITE</b>  | <b>11</b> |
| <b>FIGURE 6 MAQUETTE DE LA PAGE EMPRUNT DU SITE</b>  | <b>12</b> |
| <b>FIGURE 7 MAQUETTE DE LA PAGE LOGIN DU SITE</b>  | <b>13</b> |
| <b>FIGURE 8 MAQUETTE DE LA PAGE USERDÉTAIL DU SITE</b>   | <b>14</b> |
| <b>FIGURE 9 MAQUETTE DE LA PAGE AJOUT D'ARTICLE DU SITE</b>  | <b>15</b> |
| <b>FIGURE 10 MAQUETTE DE LA PAGE MODIFICATION D'ARTICLE DU SITE</b>  | <b>16</b> |
| <b>FIGURE 11 SCHÉMA DES BRANCHES DANS GITHUB À RÉALISER</b>  | <b>17</b> |
| <b>FIGURE 12 CAPTURE D'ÉCRAN DE LA CONFIGURATION DE SERVEUR UWAMP</b>                                      | <b>20</b> |
| <b>FIGURE 13 LE SCHÉMA MCD DE LA BASE DE DONNÉES</b>   | <b>20</b> |
| <b>FIGURE 14 SCHÉMA MLD DE LA BASE DE DONNÉES</b>  | <b>21</b> |
| <b>FIGURE 15 IMPORTATION DANS LA BASE DE DONNÉES ET VISUALISATION DANS PHPMYADMIN :</b>                    | <b>21</b> |
| <b>FIGURE 16 SCHÉMA DE LA BASE DE DONNÉES DANS LE CONCEPTEUR DE PHPMYADMIN</b>                             | <b>22</b> |
| <b>FIGURE 17 PREMIÈRE VERSION VISUELLE DE LA PAGE HOME SANS CODE CSS :</b>                                 | <b>23</b> |
| <b>FIGURE 18 PREMIÈRE VERSION VISUELLE DE LA PAGE LOGIN SANS CODE CSS</b>                                  | <b>23</b> |
| <b>FIGURE 19 PAGE LOGIN SI L'UTILISATEUR A COMMIS UNE ERREUR DANS LA SAISIE DU FORMULAIRE DE CONNEXION</b> | <b>24</b> |
| <b>FIGURE 20 PREMIÈRE VERSION VISUELLE DE LA PAGE ADDARTICLE SANS CODE CSS</b>                             | <b>25</b> |
| <b>FIGURE 21 PREMIÈRE VERSION VISUELLE DE LA PAGE ARTICLELISTE SANS CODE CSS</b>                           | <b>25</b> |
| <b>FIGURE 22 PREMIÈRE VERSION VISUELLE DE LA PAGE ARTICLE SANS CODE CSS</b>                                | <b>26</b> |
| <b>FIGURE 23 PREMIÈRE VERSION VISUELLE DE LA PAGE MODIFARTICLE SANS CODE CSS</b>                           | <b>26</b> |
| <b>FIGURE 24 PAGE LISTE DES ARTICLES AVEC LA BARRE</b>   | <b>27</b> |
| <b>FIGURE 25 PAGE DE LISTE DES ARTICLES AVEC LA RECHERCHE « LAUSANNE » ENTRÉ</b>                           | <b>27</b> |
| <b>FIGURE 26 PREMIÈRE VERSION VISUELLE DE LA PAGE EMPRUNT ARTICLE</b>                                      | <b>28</b> |
| <b>FIGURE 27 VISUELLE DE PAGE USERDÉTAIL AVEC LA LISTE DES EMPRUNT DE L'UTILISATEUR</b>                    | <b>29</b> |
| <b>FIGURE 28 VISUELLE DE PAGE USERDÉTAIL AVEC LA LISTE DES ARTICLES CRÉÉ DE L'UTILISATEUR</b>              | <b>29</b> |
| <b>FIGURE 29 PREMIER VISUEL DU BANDEAU DE NOTIFICATION</b>   | <b>30</b> |
| <b>FIGURE 30 CODE JAVASCRIPT DE LA FONCTION ET DE L'APPEL DE LA FONCTION</b>                               | <b>30</b> |
| <b>FIGURE 31 LA MÉTHODE POUR INSÉRER UN ARTICLE DANS LA BASE DE DONNÉES VIA REQUÊTE SQL</b>                | <b>31</b> |

|  |    |
|--|----|
| <b>FIGURE 32 LA MÉTHODE POUR AFFICHER TOUTES LES DONNÉES DES ARTICLES ET DES UTILISATEURS</b>    | 31 |
| <b>FIGURE 33 LA MÉTHODE POUR AFFICHER UN SEUL ARTICLE AVEC SON UTILISATEUR</b>                   | 32 |
| <b>FIGURE 34 LA MÉTHODE POUR MODIFIER L'ARTICLE</b>  | 32 |
| <b>FIGURE 35 LA MÉTHODE POUR RECHERCHER UN ARTICLE VIA LA BARRE DE RECHERCHE</b>                 | 33 |
| <b>FIGURE 36 PARAMÈTRE DE L'EXTENSION DE SELENIUM IDE</b>  | 33 |
| <b>FIGURE 37 LA LISTES COMPLÈTES DES TESTS AUTOMATISÉE DANS SELENIUM</b>                         | 34 |
| <b>FIGURE 38 TABLEAU DE COMPARAISON ENTRE LA PLANIFICATION INITIALE ET LE JOURNAL DE TRAVAIL</b> | 37 |

## 7.4 Webographie

- <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>
- <https://stackoverflow.com/questions/37280438/php-compare-two-dates-month-and-year-only>
- <https://github.com/SeleniumHQ/selenium-ide/issues/120>
- <https://codepen.io/huange/pen/bGqMqM?editors=1100>
- <https://markodenic.com/tools/buttons-generator/>
- <https://www.uwamp.com/fr/>
- <https://windows.php.net/download#php-8.2>
- <https://dev.mysql.com/downloads/mysql/5.7.html>
- <https://www.phpmyadmin.net/downloads/>
- <https://chrome.google.com/webstore/detail/selenium-ide/mooikfahbdckldjindioackbalphokd>
- <https://www.w3.org/>
- <https://sql.sh/cours/where>
- <https://grafikart.fr/forum/28211>
- <https://openclassrooms.com/forum/sujet/recuperer-tout-les-id-dune-table-avec-une-requete>
- <https://www.scaler.com/topics/button-disabled/>
- <https://www.conseil-webmaster.com/formation/php/09-variables-get-php.php#:~:text=Envoyer%20des%20variables%20dans%20l,une%20variable%20par%20l'url.>
- <https://forums.commentcamarche.net/forum/affich-31932526-phpmyadmin-ajouter-une-cle-etrangere>
- <https://sqlpro.developpez.com/cours/sqlaz/erreurs/>
- [https://fr.wikipedia.org/wiki/Dictionnaire\\_des\\_donn%C3%A9es](https://fr.wikipedia.org/wiki/Dictionnaire_des_donn%C3%A9es)
- <https://www.htmlcorrector.com/>
- <https://www.mysqltutorial.org/mysql-drop-database/>
- <https://help.talend.com/r/DRfUBNxs0UuxLf5xRafyA/v2hjs0Fr3Dt3JgmOMrcQg>
- <https://www.243tech.com/creer-une-barre-de-recherche-sur-son-site-php-mysql/#:~:text=Il%20suffit%20juste%20de%20se,cr%C3%A9er%20cette%20base%20de%20donn%C3%A9es.>
- [https://www.jetbrains.com/phpstorm/promo/?msclid=76aa88b8ec641676eb85ece3b24a3403&utm\\_source=bing&utm\\_medium=cpc&utm\\_campaign=EMEA\\_en\\_WEST\\_PhpStorm\\_Search&utm\\_term=php%20download&utm\\_content=php%20download](https://www.jetbrains.com/phpstorm/promo/?msclid=76aa88b8ec641676eb85ece3b24a3403&utm_source=bing&utm_medium=cpc&utm_campaign=EMEA_en_WEST_PhpStorm_Search&utm_term=php%20download&utm_content=php%20download)
- <https://cabotweb.fr/ajouter-des-versions-php-a-uwamp/>
- <https://cynoteck.com/fr/blog-post/what-is-selenium-getting-started-with-selenium-automation-testing/#:~:text=Selenium%20est%20une%20suite%20d,performances%20de%20votre%20application%20Web.>

## 8 ANNEXES

- Code source du projet : <https://github.com/Antpfigister/Code-source---TPI>
  - Base de données
  - Tests automatisés
  - Fichier README instruction installation
  - Code source du site Web
- Planification initiale
- Journal de travail
- Résumé du rapport de projet