Application Web pour vélos retrouvés



Michel Dos Santos Constantino – CIN4B

Lausanne, ETML

Durée

Nom du chef de projet

(Nom et adresse du mandant)

Table des matières

[1 Spécifications 4](#_Toc66446391)

[1.1 Titre 4](#_Toc66446392)

[1.2 Description 4](#_Toc66446393)

[1.3 Matériel et logiciels à disposition 4](#_Toc66446394)

[1.4 Prérequis 4](#_Toc66446395)

[1.5 Cahier des charges 5](#_Toc66446396)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 5](#_Toc66446397)

[1.5.2 Caractéristiques des utilisateurs et impacts 6](#_Toc66446398)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 6](#_Toc66446399)

[1.5.4 Contraintes 6](#_Toc66446400)

[1.5.5 Travail à réaliser par l'apprenti 7](#_Toc66446401)

[1.5.6 Si le temps le permet … 7](#_Toc66446402)

[1.5.7 Méthodes de validation des solutions 7](#_Toc66446403)

[1.6 Les points suivants seront évalués 7](#_Toc66446404)

[1.7 Validation et conditions de réussite 7](#_Toc66446405)

[2 Planification Initiale 7](#_Toc66446406)

[3 Analyse 8](#_Toc66446407)

[3.1 Opportunités 8](#_Toc66446408)

[3.2 Document d’analyse et conception 9](#_Toc66446409)

[3.2.1 Fonctionnement 9](#_Toc66446410)

[3.2.2 Schéma base de donnée 17](#_Toc66446411)

[3.2.3 Dictionnaire des données 20](#_Toc66446412)

[3.3 Conception des tests 26](#_Toc66446413)

[3.3.1 Login 26](#_Toc66446414)

[3.3.2 Système de recherche 26](#_Toc66446415)

[3.3.3 Système d’annonce 26](#_Toc66446416)

[3.3.4 Système de création de compte 26](#_Toc66446417)

[3.3.5 Création des receveurs et donneurs 27](#_Toc66446418)

[3.3.6 Modifications 27](#_Toc66446419)

[3.3.7 Statistiques 27](#_Toc66446420)

[3.4 Planification détaillée 27](#_Toc66446421)

[4 Réalisation 27](#_Toc66446422)

[4.1 Dossier de Réalisation 27](#_Toc66446423)

[4.2 Modifications 27](#_Toc66446424)

[5 Tests 28](#_Toc66446425)

[5.1 Dossier des tests 28](#_Toc66446426)

[6 Conclusion 30](#_Toc66446427)

[6.1 Bilan des fonctionnalités demandées 30](#_Toc66446428)

[6.2 Bilan de la planification 30](#_Toc66446429)

[6.3 Bilan personnel 30](#_Toc66446430)

[7 Divers 30](#_Toc66446431)

[7.1 Journal de travail 30](#_Toc66446432)

[7.2 Bibliographie 31](#_Toc66446433)

[7.3 Webographie 31](#_Toc66446434)

[8 Annexes 31](#_Toc66446435)

# Spécifications

## Titre

Application Web pour vélos retrouvés.

## Description

Création d’une plateforme commune aux villes/villages afin de permettre aux autorités de retrouver plus facilement des vélos perdus.

## Matériel et logiciels à disposition

* 1 ordinateur
* uWamp server
* Visual Studio Code
* NotePad++
* GitHub
* SQLdbm (MLD)
* Looping (MCD)
* Moqup.com (Mockups)

## Prérequis

* Modélisation de schéma de base de données
* Mise en place d’une base de données
* Mise en place modèle MVC
* PHP
* HTML
* CSS
* Javascript

## Cahier des charges

### Objectifs et portée du projet (objectifs SMART)

Combien ?

* Combien de temps ?
  + Ce projet prendra environ 6 semaines
* Pour combien d’utilisateurs ?
  + Cette application n’est pas pour le grand public mais pour les autorités communales, les utilisateurs seront donc peu nombreux (2 utilisateurs par communes maximum)

Qui ?

* Pour qui l’application est destinée ?
  + L’application est destinée aux communes
* Pour qui l’application est bénéfique ?
  + Pour toutes personnes ayant perdu son vélo ou se l’est fait volé, le vélo sera plus facilement retrouvé entre les communes.

Quoi ?

* Sur quel support sera disponible l’application ?
  + L’application sera uniquement disponible sur le Web
* Quelles sont les difficultés du projet ?
  + Principalement la mise en place du système de création de compte et les graphiques automatiques
* De quoi a besoin l’application ?
  + Un système de création de compte, de connexion, d’une database, des communes partenaires et un système de graphiques automatiques.

Comment ?

* Comment sera développé l’app ?
  + A l’aide de différents outils tel Visual Studio Code (HTML/CSS/PHP/JAVASCRIPT), uWamp (hébergement site et base de données), sqlDBM pour la modélisation de la base de données.
* Comment les tâches seront suivies ?
  + Je vais suivre les différentes tâches du projet à l’aide d’un Trello.

Où ?

* Ou sera développé l’app ?
  + A l’ETML
* Ou seront stocké les vélos ?
  + Cela est décidé par la commune ayant retrouvé le vélo.

Quand ?

* Quand est-ce que le projet doit être terminé ?
  + 19 mars 2021
* Quand est-ce que le projet peut être considéré comme terminé ?
  + Lorsque toutes les fonctions requises sont fonctionnelles et que le journal de travail et le rapport soient correctement remplis

Pourquoi ?

* Pourquoi le projet serait utile ?
  + Parce qu’il n’y a actuellement aucunes plateformes intercommunales permettant de retrouver facilement des vélos perdus/volés
* Pourquoi mettre en place un système de graphiques ?
  + Afin d’avoir une idée globale des vélos qui sont actuellement stockés et qui ont été rendus entre les différentes communes

### Caractéristiques des utilisateurs et impacts

Les utilisateurs seront des personnes travaillant dans différentes communes partenaires. L’application devra être simple d’utilisation pour la recherche et l’ajout de vélo dans la base de données car ces personnes ne sont pas formées pour cela.

### Fonctionnalités requises (du point de vue de l’utilisateur)

Il sera possible de faire des recherches sur les vélos stockés dans la base de données selon des critères descriptifs (couleur, taille, marque, etc…), tous vélos correspondant aux critères seront affichés même si le vélo n’est pas stocké dans la commune effectuant la recherche. Il sera également possible d’insérer des données dans la base de données à l’aide d’un compte « admin » unique à toutes communes (ce compte pourra rechercher ET insérer des données). Il faudra aussi avoir la possibilité de recevoir des graphiques trimestriels et annuels sur le nombre de vélo encore présent et rendus.

### Contraintes

Chaque commune aura au maximum 2 comptes (1 compte admin permettant des faire des recherches ET d’insérer des données et 1 compte qui ne pourra pas insérer de données mais pourra faire des recherches)

### Travail à réaliser par l'apprenti

Un site web avec un système de création de compte qui n’est pas automatique (le compte devra être approuvé par l’administrateur du site). Le compte créé aura un login et un mot de passe défini par l’administrateur. Je devrais également mettre en place un système de recherche et d’insertions des données pour les utilisateurs. Finalement des graphiques pourront être générés par les utilisateurs (ou automatiquement) afin de savoir le nombre de vélos stockés et rendus.

### Méthodes de validation des solutions

Je vais passer point par point sur les fonctionnalités requises du cahier des charges. Je vais tester le bon fonctionnement de la recherche en laissant volontairement des champs vides, en essayant des injections SQL, etc… et regarder si cela cause des problèmes et/ou affiche des mauvais résultats.

L’insertion des données sera testée en rentrant des données dans le mauvais format (des caractères à la place des chiffres par exemple) et en ne rentrant aucunes données

Le système de graphique sera testé en générant un graphique puis ensuite je modifierais des données dans la database puis finalement je recréerais un autre graphique afin de voir s’il se met correctement à jour.

Le système de login sera testé en essayant de me connecter sans rien écrire, en écrivant seulement le login ou seulement le mot de passe et en essayant des injections SQL.

La création des comptes sera testée en vérifiant la bonne réception du mail de confirmation et en vérifiant que le fait d’accepter la demande créée bel et bien le compte et la commune dans la base de données.

## 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

## 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

# Planification Initiale

Planification de mon [journal de travail](T-P_Appro-planification-jrnltrav.xlsm)

# Analyse

## Opportunités

La principale difficulté de ce projet est de réussir à rafraîchir suffisamment mes connaissances en PHP / Base de données ainsi que la mise en place d’un système de création de graphiques automatiques, ce que je n’ai jamais fait.

Compétences à acquérir/approfondir :

* PHP
* HTML / CSS
* MySQL
* JavaScript (pour la création de graphique automatique)
* Gestion de projet
* WebDesign

Etant donné que j’ai déjà fait des applications web utilisant du PHP, HTML/CSS et MySQL, cela est au niveau de mes compétences, il faut simplement que je rafraîchisse ces connaissances. La plus grande difficulté sera à mon avis la création des graphiques automatique car je n’ai pas beaucoup de connaissances en Javascript. Je devrais cependant réussir à me débrouiller avec les différentes documentations à ce sujet.

## Document d’analyse et conception

### Fonctionnement

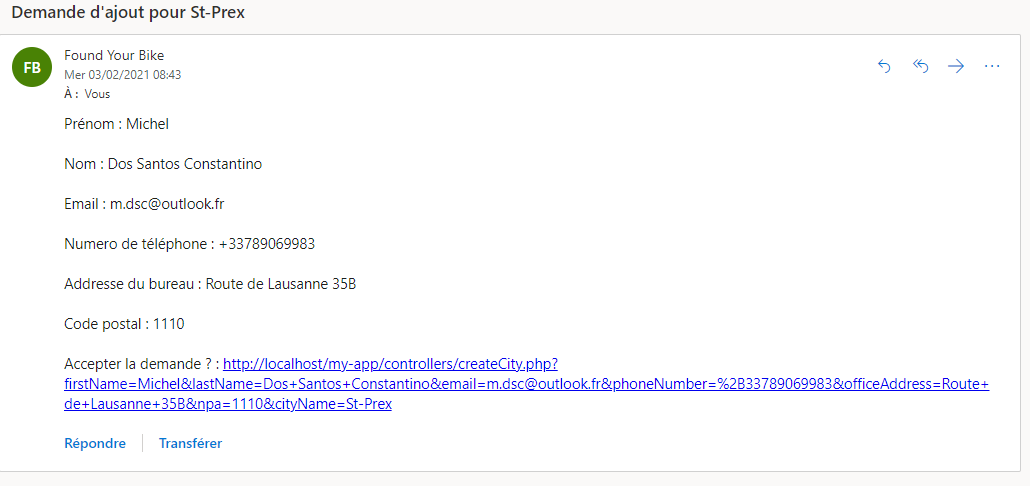
Les communes souhaitant s’inscrire sur le site devront d’abord faire une demande d’ouverture de compte à l’administrateur du site. Pour ce faire il faudra cliquer sur le lien en dessous du formulaire de login comme ci-dessous.



Figure 1: Formulaire de login (page index)

L’utilisateur sera ensuite redirigé sur un formulaire où il devra remplir tous les champs

Cette demande sera envoyée à l’adresse email « found.your.bike@outlook.fr » (adresse email créée pour le projet). Le mail de demande d’ouverture de compte d’une commune ressemblera à ça :



Le nom de la commune est dans l’objet du mail et les informations tel que prénom, nom, etc… représente la personne qui fait la demande et sera utilisé afin de créer automatiquement le nom d’utilisateur lié à la commune. Le nom d’utilisateur a comme format : les deux premières lettres du prénom + « . » + les cinq premières lettres du nom de famille.

Par exemple : Michel Dos Santos 🡺 mi.dossa

En cliquant sur le lien dans l’email, la commune en question sera automatiquement créée ainsi que l’utilisateur lié à la commune. Le mot de passe est défini par le système (généré aléatoirement) et toutes les informations relatives au login de l’utilisateur seront envoyées par mail à l’adresse email donnée par celui-ci (login + mot de passe). Le compte qui est créé automatiquement suite à la demande de la commune peut insérer des données dans la database et rechercher des infos. Si la commune souhaite ouvrir un compte qui peut uniquement rechercher des données dans la database, elle devra aller sur le deuxième formulaire d’ouverture de compte et faire une demande pour la commune en question.

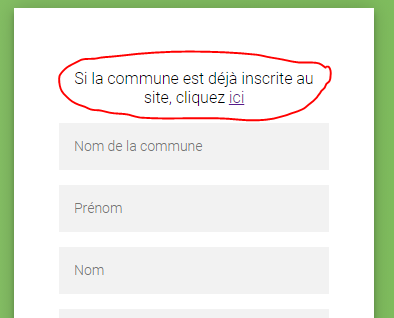


Figure 2: Pour accéder au 2ème formulaire pour compte de recherche

Il suffit de choisir pour quelle commune le compte doit être créé ainsi que remplir le reste du formulaire puis un mail sera envoyé afin d’être accepté ou non par l’administrateur. Si l’administrateur accepte la demande d’ouverture, un mail contenant le mot de passe ainsi que le login sera envoyé à l’adresse email fournie par l’utilisateur.

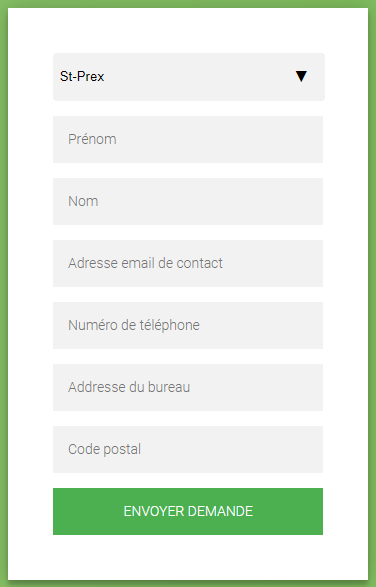


Figure 3: Formulaire pour compte de recherche

Maintenant que l’utilisateur a un compte ouvert il peut se connecter avec le login/mot de passe qu’il a reçu par mail. Lorsque l’utilisateur se connecte, il est redirigé sur la « main page » où il aura le choix de rechercher des vélos dans la DB et d’insérer un nouveau vélo retrouvé. Si le compte n’a pas les droits pour ajouter des vélos, le bouton pour ajouter ne sera tout simplement pas afficher. Sur cette page il y’a également des statistiques sur le contenu de la DB (le nombre de vélo actuellement présent dans la DB, le nombre de vélo rendu les 3 derniers mois, etc…).



Figure 4: Mockup de la page principale

(Tous les mockups réalisés sur moqups.com)

Concernant la recherche des vélos, l’utilisateur pourra remplir un formulaire avec les informations qu’il recherche (Marque du vélo, couleur, numéro de série, etc…) et les vélos correspondant aux critères de recherche seront affichés (si l’utilisateur ne mets rien dans le formulaire, tous les vélos de la DB seront affichés), à partir de là l’utilisateur pourra cliquer sur « détails » pour voir tous les attributs du vélo (lieu retrouvé, date de la trouvaille, etc…) ainsi que de modifier le vélo dans la base de donnée (si il a été rendu par exemple). Si l’utilisateur a les droits appropriés il pourra également cliquer sur « modifier » afin de pouvoir modifier les différents attributs du vélo (utile en cas d’erreur lors de l’insertion dans la base de données)

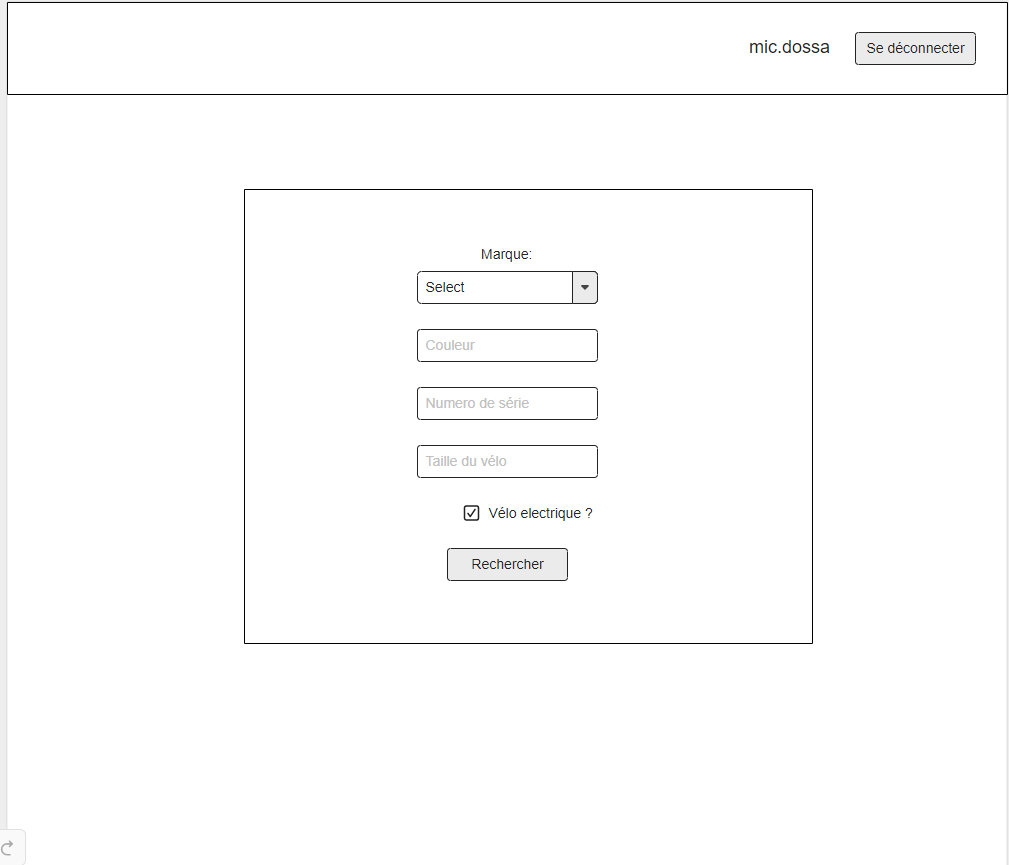


Figure 5: Formulaire de recherche

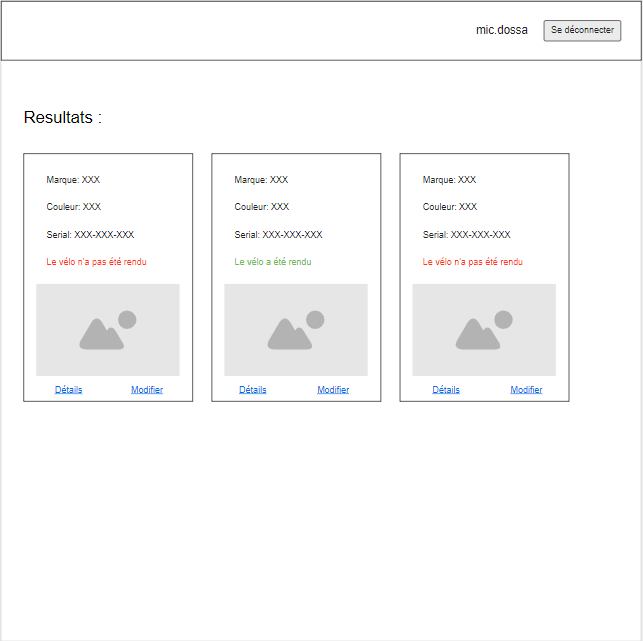


Figure 6: Page des résultats

En cliquant sur « détails » l’utilisateur pourra voir tous les attributs (également un slideshow des différentes photos) du vélo ainsi que de modifier les attributs « rendu à » et « donné par ». Si le vélo a déjà été rendu ces attributs ne seront pas modifiable directement depuis la page détail.

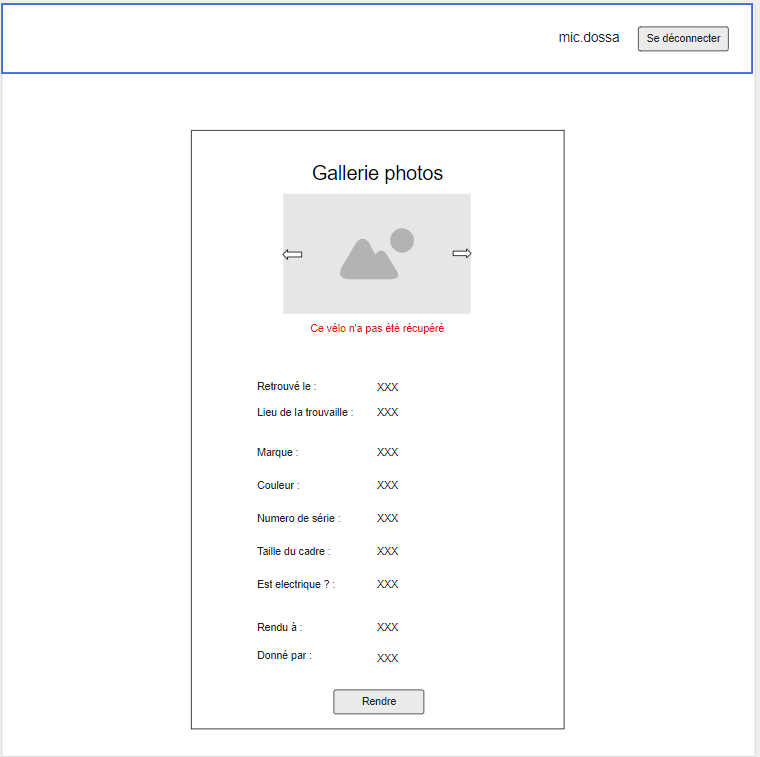


Figure 7: Mockup de la page détail d'un vélo qui n’a pas été rendu

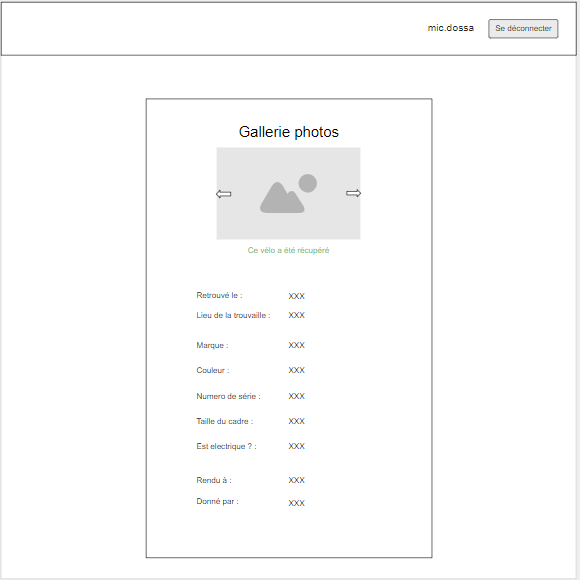


Figure 8: Mockup de la page détail d'un vélo qui a été rendu

Si le vélo n’a pas été rendu, l’utilisateur a la possibilité de cliquer sur le bouton « rendre » qui va le rediriger sur cette page.



Figure 9: Page permettant de définir le receveur/donneur du vélo

Depuis cette page il aura la possibilité de définir la personne qui reçoit le vélo (le receveur) et la personne qui lui rend (l’employé de la commune qui s’occupe de rendre le vélo a la personne). Si le receveur et/ou le donneur n’est pas déjà enregistré dans la base de données, il peut cliquer sur les liens pour rapidement les enregistrer.



Figure 10: Page de création du receveur



Figure 11: Page de création du donneur

Etant donné que le donneur et le receveur ont les mêmes attributs les deux pages sont identiques. Lorsque l’utilisateur appuie sur le bouton « créer », le donneur/receveur est créé dans la base de données et l’utilisateur est redirigé sur la page précédente (la page qui permet de définir le receveur/donneur).

L’utilisateur peut aussi cliquer sur « modifier » (si il a les accès) pour accéder à la page de modifications.

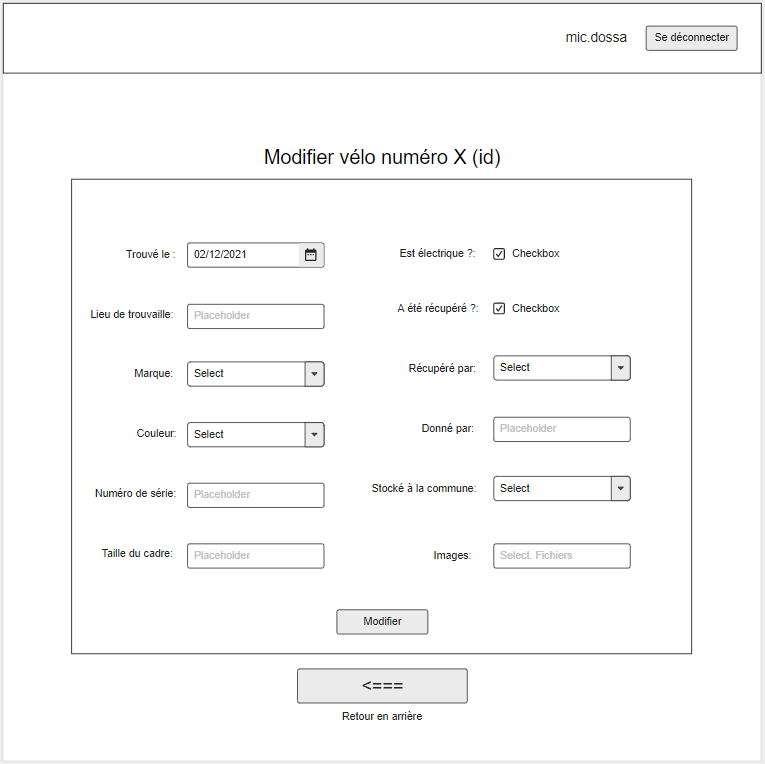


Figure 12: Mockup de la page de modification

Tous les attributs du vélo peuvent être modifié ici et les photos qui y sont liées peuvent être supprimées et changées par d’autres.

Pour ce qui est de la page permettant d’ajouter des vélos dans la base de données, elle ressemblera à cela :

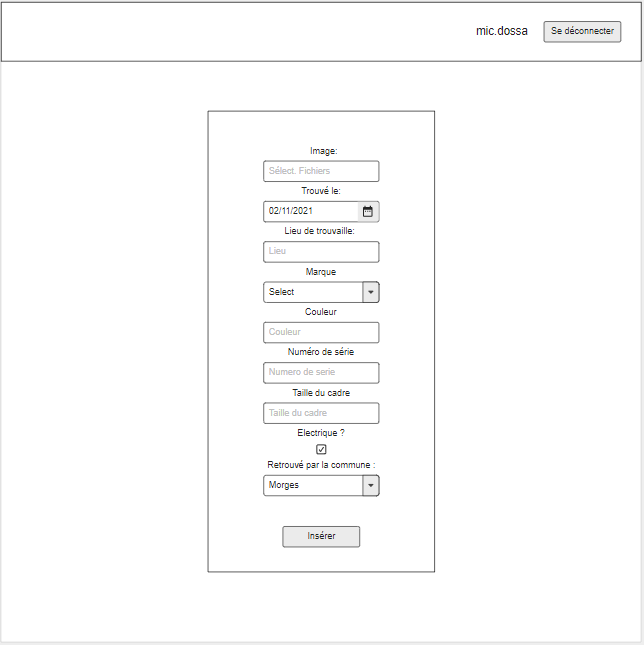
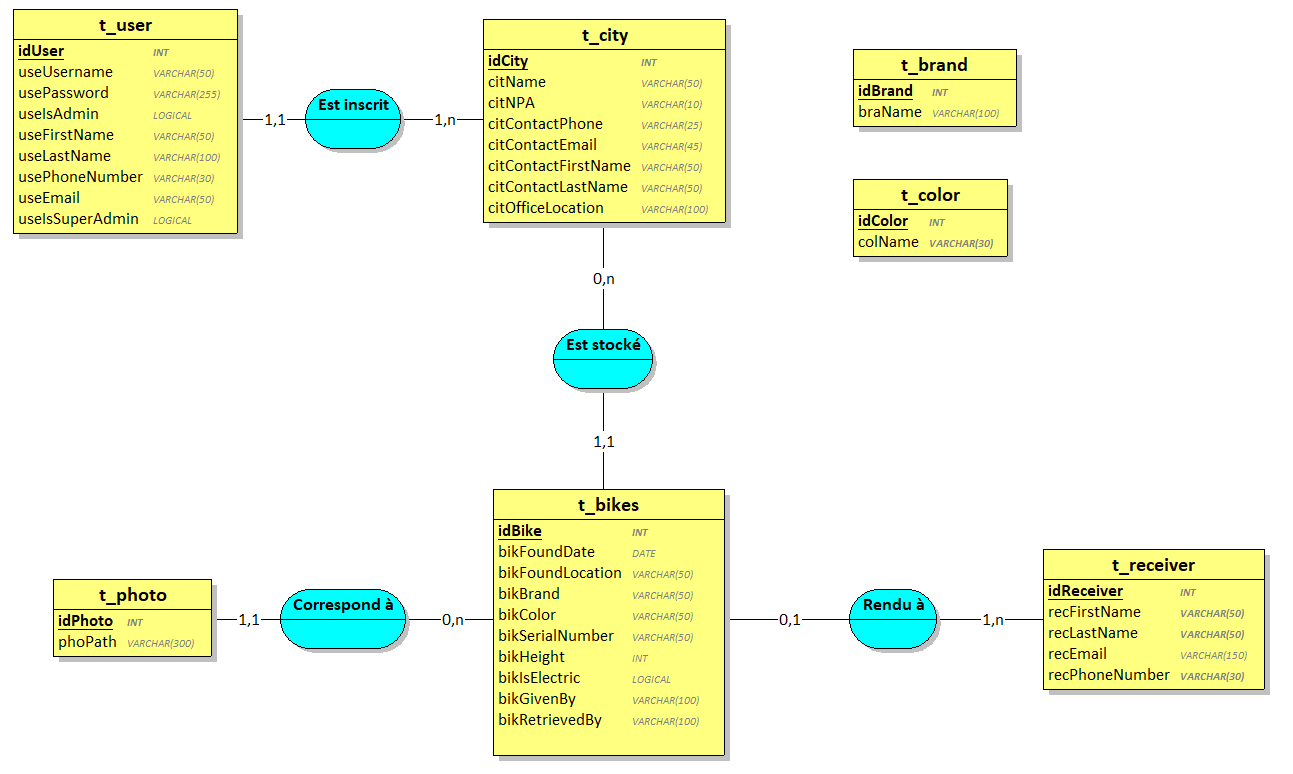


Figure 13: Mockup de la page pour ajouter un vélo

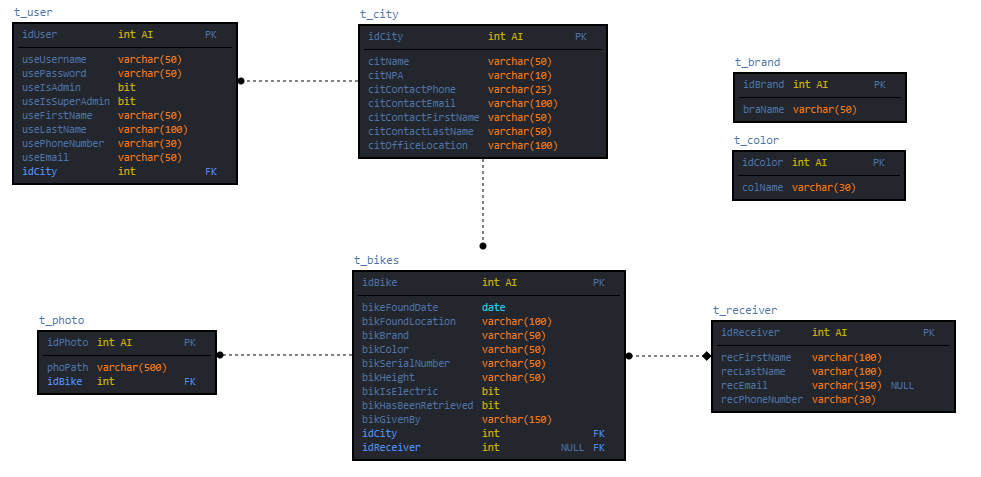
Sur ce formulaire il est possible d’uploader une ou plusieurs photos du vélo qui seront stockées dans le dossier « img/bike\_photos » à la racine du site. Après l’upload de ces photos sur le serveur, le chemin de celles-ci seront stockées dans la table t\_photo avec comme clé étrangère le vélo associé. Cela permet qu’un vélo aie plusieurs photos. L’utilisateur devra également remplir le reste du formulaire avant de pouvoir insérer le vélo dans la base de données.

### Schéma base de donnée

MCD :



MLD :



### Dictionnaire des données

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Nom de l’attribut | Table | Description | Type | Taille | NULL ? | Valeur par défaut | Remarque |
| idBike | t\_bikes | Identifiant du vélo | int | 11 | Non |  |  |
| bikeFoundDate | t\_bikes | Date auquel le vélo a été retrouvé | date |  | Non |  | Format : AAAA-MM-JJ |
| bikFoundLocation | t\_bikes | Lieu de la trouvaille | string | 100 | Non |  |  |
| bikBrand | t\_bikes | Marque du vélo | string | 50 | Non |  |  |
| bikColor | t\_bikes | Couleur du vélo | string | 50 | Non |  |  |
| bikSerialNumber | t\_bikes | Numéro de série du vélo | string | 50 | Non |  |  |
| bikHeight | t\_bikes | Taille du cadre du vélo | string | 50 | Non |  |  |
| bikIsElectric | t\_bikes | Bool qui définit si le vélo est électrique ou non | tinyint | 4 | Non |  |  |
| bikHasBeenRetrieved | t\_bikes | Bool qui définit si le vélo a été récupéré par le propriétaire | tinyint | 4 | Non |  |  |
| bikRetrieveDate | t\_bikes | Date à laquelle le vélo a été rendu a son propriétaire | date |  | Oui | NULL | Format : AAAA-MM-JJ |
| idCity (FK) | t\_bikes | Clé étrangère de la commune qui détient le vélo | int | 11 | Non |  |  |
| idReceiver (FK) | t\_bikes | Clé étrangère qui représente la personne qui a récupéré le vélo | int | 11 | Oui | NULL |  |
| idGiver (FK) | t\_bikes | Clé étrangère qui représente la personne (travaillant à la commune) qui a rendu le vélo | int | 11 | Oui | NULL |  |
| idBrand | T\_brand | Identifiant de la marque | int | 11 |  |  |  |
| braName | T\_brand | Nom de la marque | string | 50 | Non |  |  |
|  |  |  |  |  |  |  |  |
| idCity | T\_city | Identifiant de la commune | int | 11 | Non |  |  |
| citName | T\_city | Nom de la commune | string | 50 | Non |  |  |
| citNPA | T\_city | Code postal de la commune | string | 10 | Non |  |  |
| citContactPhone | T\_city | Numéro de téléphone de la personne de contact de la commune (la personne qui a fait la demande d’ouverture de compte) | string | 25 | Non |  |  |
| citContactEmail | T\_city | Adresse email de la personne de contact | string | 100 | Non |  |  |
| citContactFirstName | T\_city | Prénom de la personne de contact | string | 50 | Non |  |  |
| citContactLastName | T\_city | Nom de famille de la personne de contact | string | 50 | Non |  |  |
| citOfficeLocation | T\_city | Adresse du bureau de la commune | string | 100 | Non |  |  |
|  |  |  |  |  |  |  |  |
| idColor | T\_color | Identifiant de la couleur | int | 11 | Non |  |  |
| colName | T\_color | Nom de la couleur | string | 30 | Non |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| idGiver | T\_giver | Identifiant de la personne qui a rendu le vélo | int | 11 | Non |  |  |
| givFirstName | T\_giver | Prénom de la personne qui a rendu le vélo | string | 100 | Non |  |  |
| givLastName | T\_giver | Nom de famille | string | 100 | Non |  |  |
| givEmail | T\_giver | Adresse email de la personne qui a rendu le vélo | string | 150 | Oui | NULL |  |
| givPhoneNumber | T\_giver | Le numéro de téléphone de la personne qui a rendu le vélo | string | 30 | Non |  |  |
|  |  |  |  |  |  |  |  |
| idPhoto | T\_photo | Identifiant de la photo | int | 11 | Non |  |  |
| phoPath | T\_photo | Chemin de la photo | string | 500 | Non |  |  |
| idBike (FK) | T\_photo | Clé étrangère du vélo à laquelle est liée la photo | int | 11 | Non |  |  |
|  |  |  |  |  |  |  |  |
| idReceiver | T\_receiver | Identifiant de la personne qui a récupéré le vélo | int | 11 | Non |  |  |
| recFirstName | T\_receiver | Prénom de la personne qui a récupéré le vélo | string | 100 | Non |  |  |
| recLastName | T\_receiver | Nom de famille de la personne qui a récupéré le vélo | string | 100 | Non |  |  |
| recEmail | T\_receiver | Email de la personne qui a récupéré le vélo | string | 150 | Oui |  |  |
| recPhoneNumber | T\_receiver | Numéro de téléphone de la personne qui a récupéré le vélo | string | 30 | Non |  |  |
|  |  |  |  |  |  |  |  |
| idUser | T\_user | Identifiant de l’utilisateur | int | 11 | Non |  |  |
| useUsername | T\_user | Nom d’utilisateur de l’utilisateur | string | 50 | Non |  |  |
| usePassword | T\_user | Mot de passe hashé de l’utilisateur | string | 300 | Non |  | Mot de passe hashé |
| useIsAdmin | T\_user | Attribut qui définit si l’utilisateur est admin ou non (permissions d’ajouter dans la base de données) | tinyint | 4 | Non |  |  |
| useIsSuperAdmin | T\_user | Attribut qui définit si l’utilisateur est « super-admin », c’est-à-dire avoir les droits de créer les utilisateurs dans la base de données. | tinyint | 4 | Non |  | C’est le compte de l’administrateur du site, ce n’est pas possible de faire une demande d’ouverture de compte super-admin |
| useFirstName | T\_user | Prénom de l’utilisateur | string | 50 | Non |  |  |
| useLastName | T\_user | Nom de famille de l’utilisateur | string | 100 | Non |  |  |
| usePhoneNumber | T\_user | Numéro de téléphone de l’utilisateur | string | 30 | Non |  |  |
| useEmail | T\_user | Adresse email de l’utilisateur | string | 50 | Non |  |  |
| idCity (FK) | T\_user | Identifiant de la commune liée à l’utilisateur | int | 11 | Non |  |  |

## Conception des tests

Ce chapitre contient tous les tests que je vais faire tous le long du projet afin de vérifier le bon fonctionnement des différentes fonctionnalités.

### Login

Les différents tests que je vais faire pour m’assurer du bon fonctionnement du login et des permissions des utilisateurs (et l’accès aux pages qui nécessitent qu’un utilisateur soit connecté) :

* Ne rien mettre dans le login/mot de passe
* Mettre uniquement un nom d’utilisateur existant
* Mettre uniquement un mot de passe existant
* Essayer une injection de code SQL depuis le formulaire
* Essayer d’accéder à une page qui nécessite d’être login depuis l’URL
* Vérifier que les fonctionnalités réservées aux « admin » (compte recherche + annonce) ne s’affichent pas pour les comptes qui ne peuvent que rechercher

### Système de recherche

* Ne rien mettre dans le formulaire de recherche (si rien n’est mis, tous les vélos doivent de la base de données doivent s’afficher).
* Rentrer des critères de recherche non adaptés à l’attribut recherché (REGEX)
* Vérifier que les résultats obtenus correspondent bien aux attributs recherchés

### Système d’annonce

* Essayer d’insérer un vélo sans rien mettre dans les attributs
* Essayer d’insérer un vélo avec les attributs dans le mauvais format (REGEX et vérifications du type de variable renvoyé ainsi que les valeurs limites de celle-ci)
* Uploader des fichiers qui ne sont pas des photos (vidéo ou texte par exemple). Les formats autorisés sont .JPEG et .PNG.
* Remplir seulement en partie les attributs

### Système de création de compte

* Vérifier que le « controller » qui s’occupe de la création de la commune ainsi que des users ne soient accessible uniquement par un user « super-admin » (compte créé par l’administrateur du site)
* Vérifier que la création de la commune + le compte utilisateur ne se fasse pas si un ou plusieurs attributs manquent
* Essayer de créer un compte qui a le même nom qu’un autre utilisateur dans la base de données afin de vérifier que le compte ne se crée pas si le login existe déjà.

### Création des receveurs et donneurs

* Ne rien mettre dans les formulaires
* Essayer de créer un receveur/donneur sans être connecté et en étant connecté avec un compte n’ayant pas les permissions de création
* Essayer une injection SQL

### Modifications

* Essayer une injection SQL
* Modifier en étant connecté avec un compte sans les autorisations appropriées

### Statistiques

* Vider la table de tous les vélos et voir ce qui s’affiche
* Modifier la date à laquelle a été rendu un vélo et regarder si cela s’est correctement mis à jour

## Planification détaillée

Pour avoir accès à la planification, cliquez [ici](T-P_Appro-planification-jrnltrav.xlsm)

# Réalisation

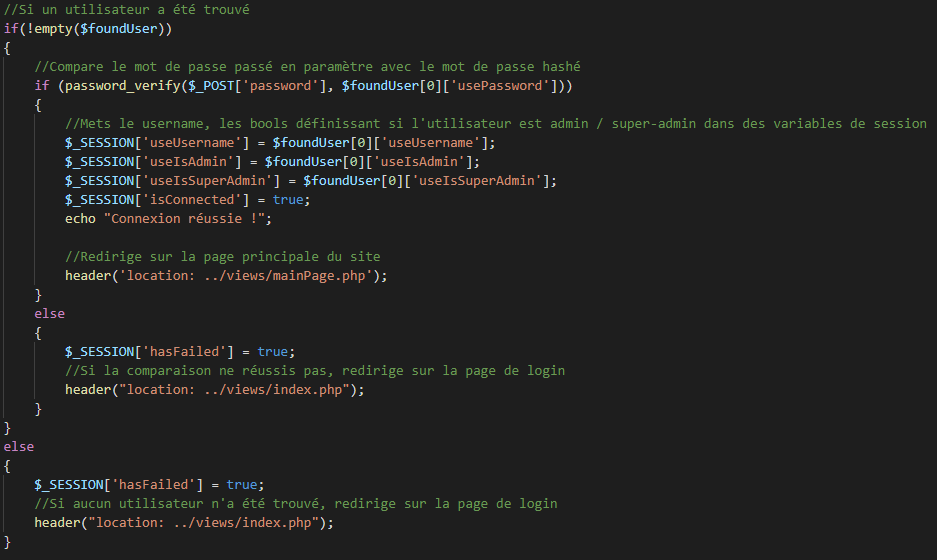
## Dossier de Réalisation

### Page de connexion

Pour la page de connexion j’ai utilisé [ce](https://colorlib.com/wp/template/creative-login-form/) template que j’ai ensuite utilisé pour la totalité du site.

Outils :

* Visual Studio Code v1.54.1
* uWamp v3.1.0
* php-5.6.18
* mysql-5.7.11

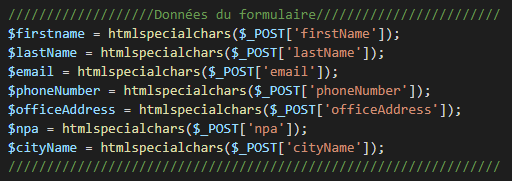
Et voici le controller qui est appelé lorsque l’utilisateur appuie sur « se connecter » sur la page de login.

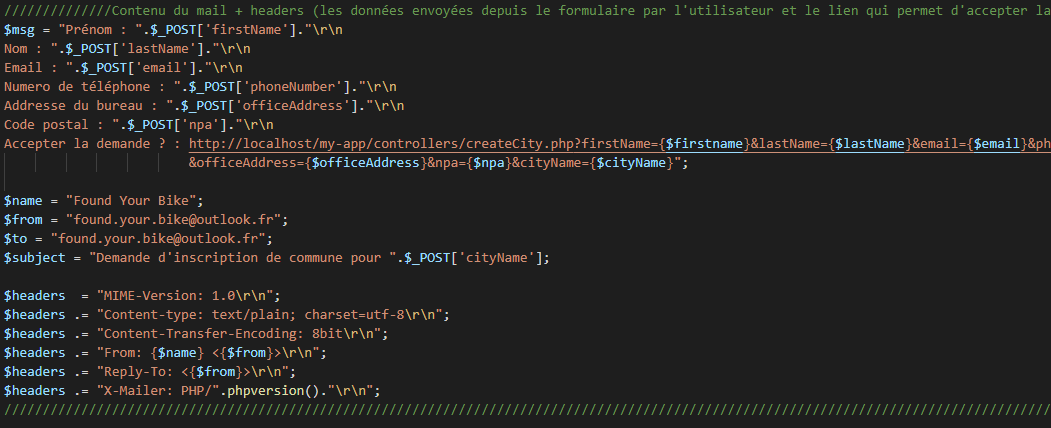
Je n’ai pas mis la totalité du code (il y’a des vérifications des données entrées par l’utilisateur plus haut). Cette partie du code vérifie qu’un utilisateur ayant le username passé par l’utilisateur et si c’est le cas vérifie le mot de passe hashé qui a été retourné, avec le mot de passe donné par l’utilisateur essayant de se connecter. Si le username ou le mot de passe n’est pas correct, redirige sur la page de login.

### Demande d’ouverture de compte

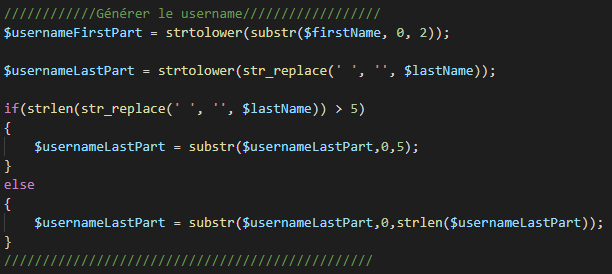
Lorsque l’utilisateur a rempli le formulaire avec les informations concernant sa demande d’ouverture de compte et qu’il appuie sur le bouton d’envoi, ce code va se lancer.

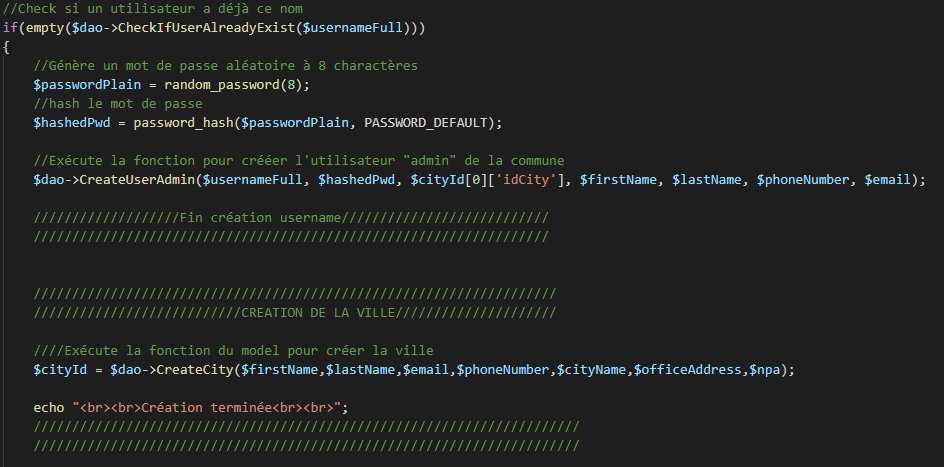
Tout d’abord on récupère les données stockées dans le $\_POST puis je les encode pour les insérer dans l’url (fonction urlencode() de php)



Ensuite je colle ces données dans le string qui correspond au mail.

Puis finalement j’envoie le mail. L’adresse email qui recevra ce mail est found.your.bike@outlook.fr , je l’ai créée spécialement pour le projet. L’admin du site verra ensuite les données de la personne qui a fait la demande d’ouverture de compte et pourra décider si oui ou non. Si l’admin du site accepte la demande d’ouverture en cliquant sur le lien il va lancer le controller s’occupant de la création de la commune et du compte qui y est automatiquement lié. Tout d’abord le controller va créer le nom d’utilisateur.

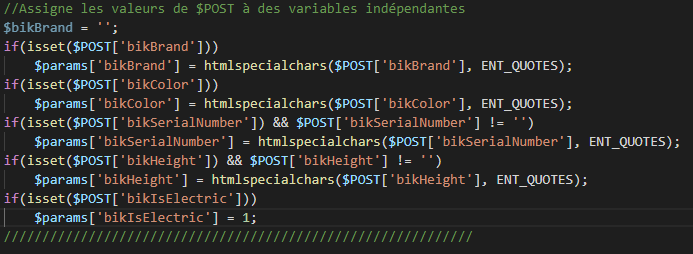


Ensuite il va vérifier si ce username existe déjà dans la base de données et si il n’existe pas, va créer un mot de passe aléatoire puis va créer la commune et le compte en même temps

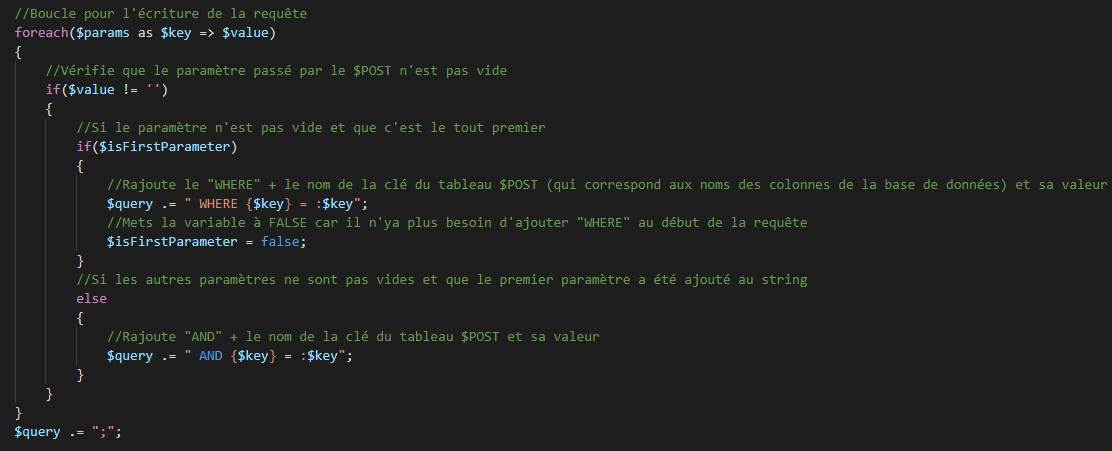
Puis les données de connexion sont envoyées par mail à l’adresse fournie par l’utilisateur qui a fait la demande. Le système est plus ou moins pareil pour la requête d’ouverture du compte à une commune déjà inscrite (compte recherche).

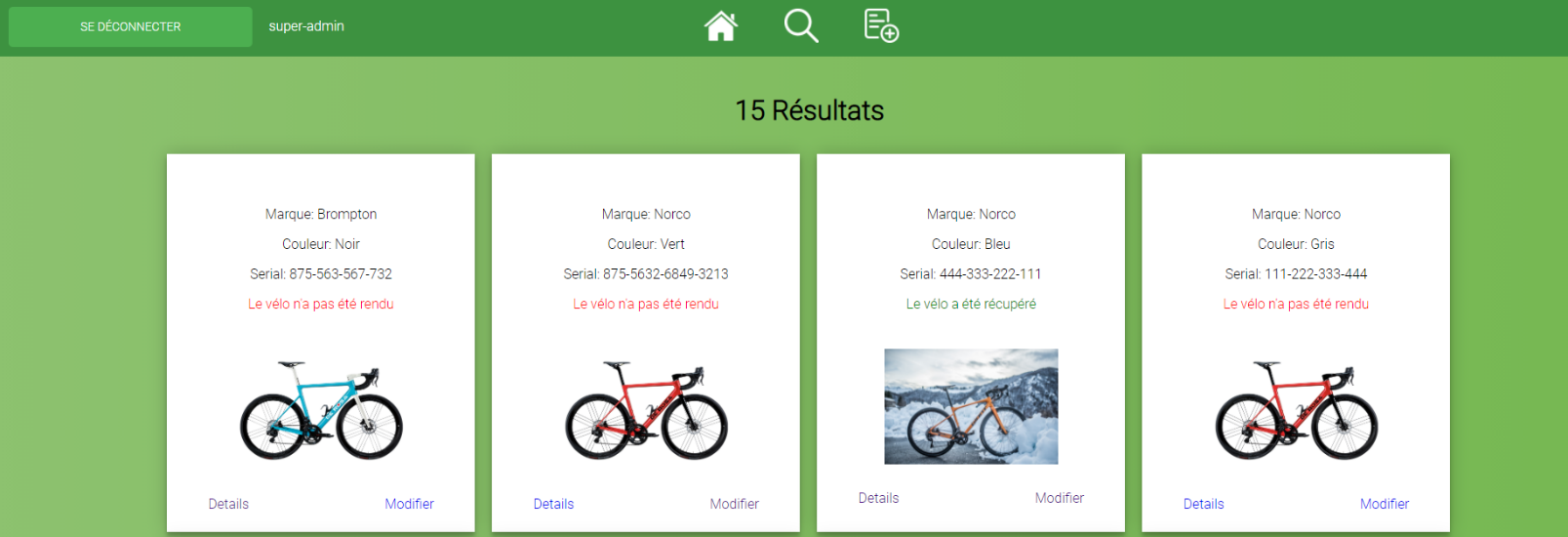
### Système de recherche

La partie du code la plus importante de cette fonctionnalité est la création de la requête en fonction de ce que l’utilisateur décide de rechercher. Voici le code :

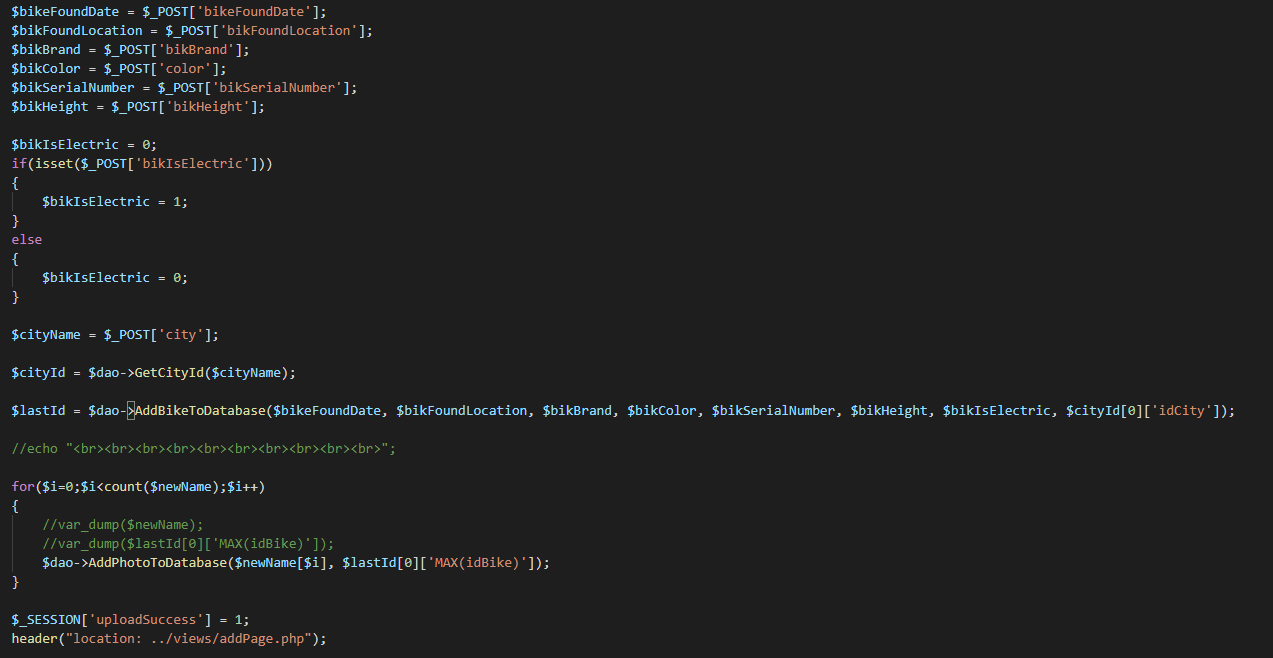


Tout d’abord on ajoute les variables dans un tableau de paramètres (pour le bindParam())

Puis on construit la requête en fonction de la clé du tableau associatif. Après avoir créé la requête, s’il y’a aucune erreur, exécute la requête.

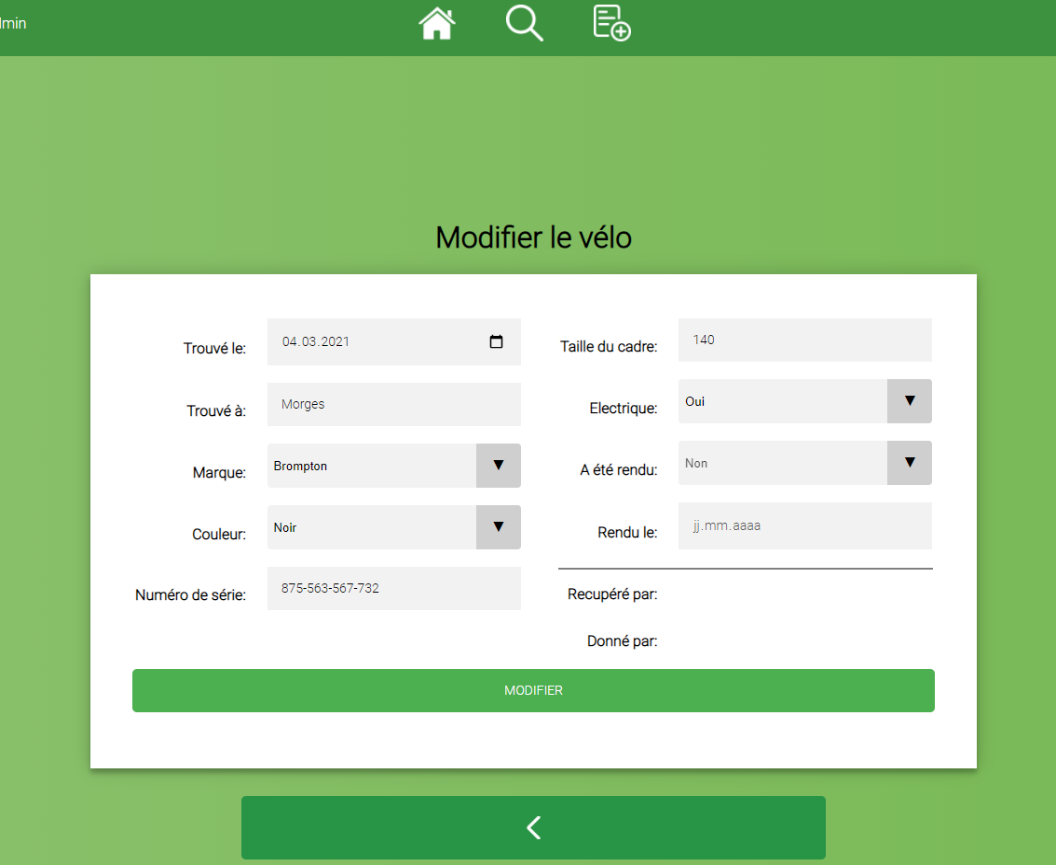
Page de résultats :

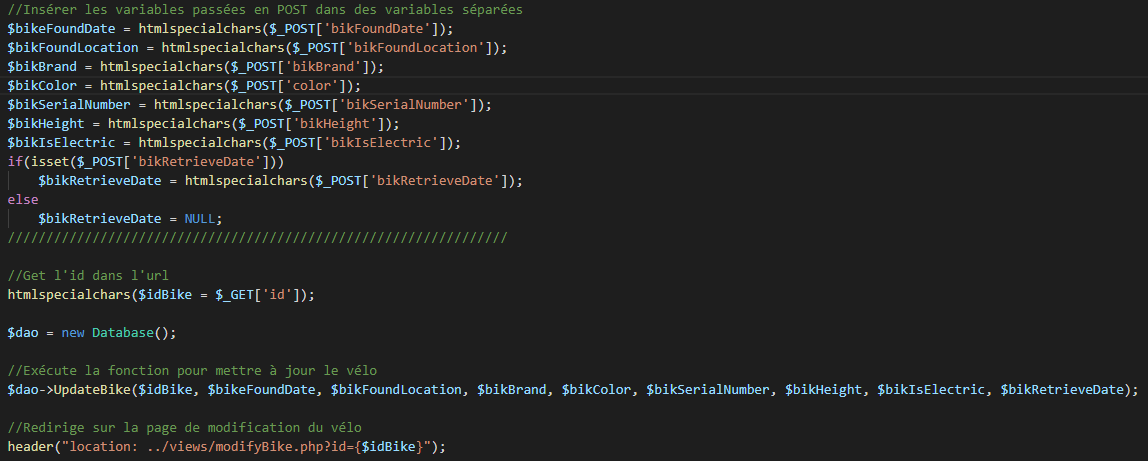
### Ajouter un vélo

Tout d’abord on va vérifier que tous les attributs ont été envoyé correctement ($\_POST et $\_FILES) puis les extensions, taille, nom de fichier, etc… Lorsque tout est bon on peut ajouter le vélo à la base de données ainsi que toutes les photos qui y sont liées.

### Modification des vélos

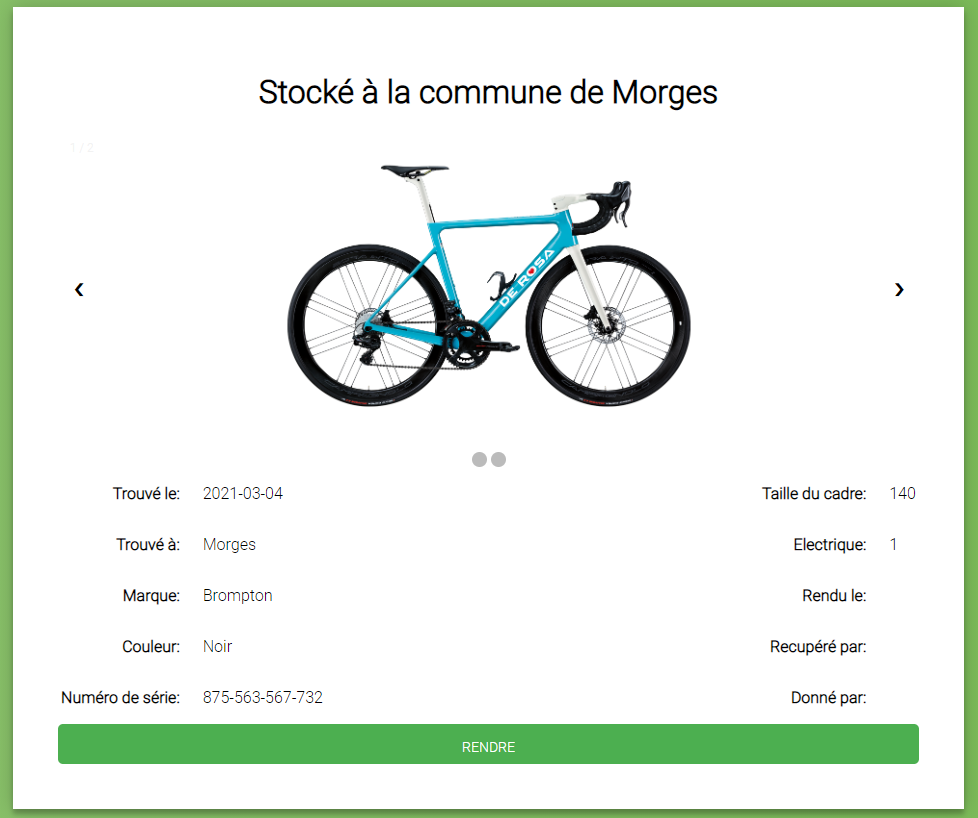
Voici la page de modification d’un vélo :

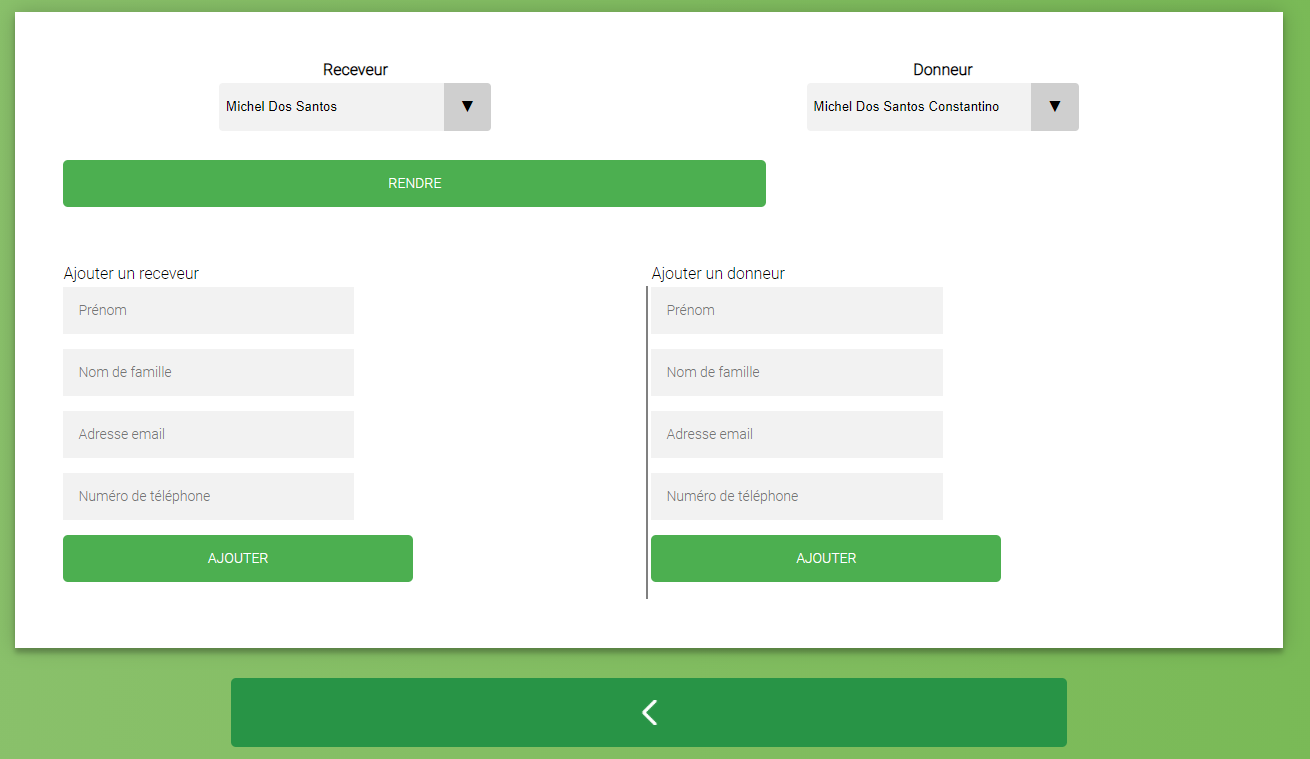


Après avoir cliqué sur « Modifier », les données de tous les <input> sont envoyées au controller (modifié ou non) puis le code va mettre ces données dans des variables puis les passer en paramètre au modèle pour qu’il mette le vélo à jour.

### Rendre un vélo

Page de détails :



Il est possible de rendre le vélo depuis la page de détails en cliquant sur « rendre ». Le bouton redirige sur la page qui permet de créer des « donneurs » et « receveurs »

On peut également choisir le receveur et le donneur puis mettre à jour le statut vélo (s’il a été rendu et à qui).

### Barre de navigation

Voici la barre de navigation :

L’utilisateur peut se déconnecter depuis la navbar ainsi qu’aller à la mainpage, la page de recherche et la page d’annonce.

# Tests

## Dossier des tests

|  |  |  |
| --- | --- | --- |
| Fonctionnalité testée | Test | Résultat |
| Login | Aucune donnée entrée dans le formulaire de login | Redirige sur la page de login |
| Login | Seulement le username | Redirige sur la page de login  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Login | Seulement le mot de passe | Redirige sur la page de login  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Login | J’ai essayé de rentrer cette requête dans le username : « ' OR 1=1;/\* » avec un mot de passe correct. | Redirige sur la page de login  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Login | Accès à une/des page(s) sans avoir les permissions nécessaires | Redirige sur la page de login  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Login | Je me suis connecté avec un compte non « admin » et j’ai regardé si les boutons modifier s’affichaient | Les boutons d’annonce et de modifications des vélos ne s’affichent pas  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système de recherche | Ne rien mettre dans les formulaire | Tous les vélos s’affichent (résultat voulu)  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système de recherche | Mettre des données au mauvais format | Aucuns résultats s’affichent. Les regex n’ont pas été mis en place donc la recherche se fait tout de même mais ne trouve rien. Aucune erreur ne s’affiche. |
| Système de recherche | Vérifié que les résultats qui s’affichent correspondent bien à ce que l’utilisateur a recherché | Les résultats correspondent bien à ce que j’ai recherché  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système d’annonce de vélo | J’ai supprimé le « required » des input et j’ai essayé d’annoncer le vélo. | Une erreur s’affiche et l’action est annulée  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
|  |  |  |
| Système d’annonce de vélo | Vérifier que les données sont envoyées au bon format | Les regex n’ont pas été mis en place, les input au format string peuvent contenir n’importe quoi  cancel |
| Système d’annonce de vélo | Mettre un fichier avec un format autre que .png, .jpeg, .jpg | J’ai essayé de mettre un pdf et l’extension a été correctement vérifiée  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système d’annonce de vélo | Je n’ai pas rempli un des attributs | L’attribut vide est détecté et une erreur s’affiche  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système de création de compte | J’ai essayé de créer un compte en étant deconnecté du user super-admin | Le compte ne se crée pas, je suis redirigé sur la page de login  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Système de création de compte | Essayer de créer un compte alors que les attributs ne sont pas tous remplis | Le compte peut être créé même si des attributs sont manquants, mais étant donné que chaque création de compte est faite manuellement ce n’est pas un gros problème  cancel |
| Système de création de compte | Essayer de créer un utilisateur qui existe déjà dans la base de données | Le compte ne se crée pas si un utilisateur du même nom existe  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Création receveurs / donneurs | Ne rien mettre dans le formulaire | Le contenu du formulaire est vérifié, si c’est vide, ne crée pas le donneur / receveur  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Création receveurs / donneurs | Essayer de créer un receveur / donneur sans être connecté | Le controller vérifie bien que l’utilisateur soit connecté. Aucunes permissions particulières n’est nécessaire.  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
|  |  |  |
| Création receveurs / donneurs | Injection SQL | Les paramètres sont insérés dans la requête avec un bindParam() donc l’injection SQL ne fonctionne pas  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Modifications des vélos | Injection SQL | Les paramètres sont insérés dans la requête avec un bindParam() donc l’injection SQL ne fonctionne pas  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Modifications des vélos | Essayer de modifier sans avoir les permissions ou en étant déconnecté | Les permissions sont vérifiées et la connexion de l’utilisateur également  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Statistiques | Voir ce qui se passe si aucuns vélos n’est dans la base de données | Aucunes statistiques ne sortent, pas d’erreur  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |
| Statistiques | Vérifier si les statistiques changent après avoir modifié la date à laquelle a été rendu le vélo | Les statistiques se modifient correctement  C:\Users\micdossantos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checked.png |

# Conclusion

## Bilan des fonctionnalités demandées

### Inscription d’une commune

Chaque commune qui souhaite rejoindre la plateforme, pourra s’inscrire en ligne en donnant les renseignements habituels tels que commune, nom, prénom, fonction de la personne, adresse complète, email, téléphone, etc... Cette inscription ne sera pas automatique, une validation manuelle par le responsable du site sera faite, puis un email en retour d’acceptation sera envoyé par le système. Le login et le mot de passe seront définis automatiquement par la plate-forme.

### Gestion des utilisateurs d’une commune

Chaque commune n’aura droit qu’à un seul compte de saisie et un seul compte de consultation/recherche. Ceci permet d’éviter la multiplication des comptes inutiles. Une personne par commune pourra donc tout faire (saisie, recherche, etc...) et un compte de consultation pourra être offert aux autres personnes de l’administration communale.

### Annonce d’un vélo trouvé

Chaque commune, une fois inscrite, pourra insérer dans le site toutes les informations relatives au vélo trouvé sur son territoire. Photo, date, lieu, marque du vélo, couleur, no de cadre, taille, etc... et toutes les informations utiles et réelles sur cet engin. Cette annonce va donc enrichir la BD du système avec un vélo de plus annoncé retrouvé. Il ne restera plus qu’à retrouver son propriétaire !

### Rechercher un vélo dans le système

Un habitant d’une des communes utilisatrices du système peut arriver aux bureaux communaux et annoncer que son vélo a été perdu/volé. Il peut donc alors décrire son vélo, donner la marque, la couleur, etc... L’employé communal peut alors, sans rien montrer à ce citoyen, lancer une recherche selon les critères donnés et voir, si dans toute l’application FindYourBike, son vélo s’y trouve. Si c’est le cas, l’employé peut alors lui annoncer si son vélo est ici dans sa commune, ou bien s’il se trouve dans une des autres communes adhérentes au système.

### Statistiques

Une fois tous les trimestres (= 3 mois), chaque commune doit pouvoir sortir une statistique de tous les vélos se trouvant dans la BD. Celle-ci doit donner tous les vélos encore présents, et tous ceux qui ont été rendus à leur propriétaire durant les 3 derniers mois. Une fois par année, la même statistique pour l’entier de l’année est demandée.

## Bilan de la planification

### Création des comptes

Cette fonctionnalité m’a pris beaucoup plus de temps que prévu parce que je pensais qu’il serait simple d’envoyer un mail avec la fonction « mail » qui est utilisable par défaut en PHP. Malheureusement l’envoi n’était pas si simple car j’utilisais uWamp pour héberger mon serveur et cette fonction ne marchait pas. Il fallait changer le fichier php.ini, définir le serveur de mail etc… Même après avoir fait les modifications nécessaires, l’envoi ne fonctionnait toujours pas. J’ai donc dû trouver une extension que je pouvais installer et qui permettait l’envoi des mails depuis un serveur uWamp. J’ai également perdu du temps sur cette fonctionnalité parce que je ne savais pas exactement comment faire la confirmation de la création de l’utilisateur/commune.

## Bilan personnel

Si ce projet était à refaire je passerai beaucoup plus de temps à l’analyse du projet avant de me lancer dans le code. Je me suis trop vite mit au code et cela m’a fait perdre beaucoup de temps car j’ai dû modifier plusieurs fois la base de données et le MVC n’est pas implémenté correctement. Je donnerai également plus de temps consacré au rapport, je m’y suis mis trop tard et donc je n’ai pas pu le faire correctement. En ce qui concerne le code, je ferais également plus attention à la vérification des données envoyées depuis les formulaires (mon code était vulnérables aux injections SQL) ainsi que mettre correctement en place le MVC. Finalement je n’ai pas pu non plus faire un design correct pour le site par manque de temps.

Cependant ce projet a pu rafraichir mes connaissances en HTML/CSS/PHP/SQL que j’avais bien perdu suite à mon stage d’1 an. Il m’a aussi appris l’importance de la vérification des données envoyées par l’utilisateur ainsi qu’à la sécurité du code en général.

Les améliorations que je souhaiterais sur ce projet sont :

* Responsive design
* Un MVC correct
* Plus de commentaires
* Des statistiques plus poussées
* Un code mieux organisé
* Un meilleur rapport
* Une meilleure planification de départ

# Divers

## Journal de travail

Journal de travail [ici](T-P_Appro-planification-jrnltrav.xlsm)

## Webographie

PHP :

<https://www.php.net/manual/fr/function.mail.php>

<https://stackoverflow.com/questions/14456673/sending-email-with-php-from-an-smtp-server>

<https://www.php.net/manual/fr/book.pdo.php>

<https://www.php.net/manual/fr/reserved.variables.session.php>

SQL :

<https://www.w3schools.com/sql/func_mysql_quarter.asp>

<https://stackoverflow.com/questions/11141100/how-to-get-calendar-quarter-from-a-date-in-tsql>

<https://www.w3schools.com/sql/sql_insert.asp>

<https://www.w3schools.com/sql/sql_update.asp>

HTML :

<https://www.w3schools.com/tags/tag_select.asp>

<https://stackoverflow.com/questions/3518002/how-can-i-set-the-default-value-for-an-html-select-element>

JAVASCRIPT :

<https://www.w3schools.com/howto/howto_js_slideshow_gallery.asp>