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 3](#_Toc66361851)

[1.1 Titre 3](#_Toc66361852)

[1.2 Description 3](#_Toc66361853)

[1.3 Matériel et logiciels à disposition 3](#_Toc66361854)

[1.4 Prérequis 3](#_Toc66361855)

[1.5 Cahier des charges 4](#_Toc66361856)

[1.5.1 Objectifs et portée du projet (objectifs SMART) 4](#_Toc66361857)

[1.5.2 Caractéristiques des utilisateurs et impacts 5](#_Toc66361858)

[1.5.3 Fonctionnalités requises (du point de vue de l’utilisateur) 5](#_Toc66361859)

[1.5.4 Contraintes 5](#_Toc66361860)

[1.5.5 Travail à réaliser par l'apprenti 6](#_Toc66361861)

[1.5.6 Si le temps le permet … 6](#_Toc66361862)

[1.5.7 Méthodes de validation des solutions 6](#_Toc66361863)

[1.6 Les points suivants seront évalués 6](#_Toc66361864)

[1.7 Validation et conditions de réussite 6](#_Toc66361865)

[2 Planification Initiale 6](#_Toc66361866)

[3 Analyse 7](#_Toc66361867)

[3.1 Opportunités 7](#_Toc66361868)

[3.2 Document d’analyse et conception 8](#_Toc66361869)

[3.2.1 Fonctionnement 8](#_Toc66361870)

[3.2.2 Schéma base de donnée 16](#_Toc66361871)

[3.2.3 Dictionnaire des données 18](#_Toc66361872)

[3.3 Conception des tests 20](#_Toc66361873)

[3.3.1 Login 20](#_Toc66361874)

[3.3.2 Système de recherche 20](#_Toc66361875)

[3.3.3 Système d’annonce 20](#_Toc66361876)

[3.3.4 Système de création de compte 21](#_Toc66361877)

[3.4 Planification détaillée 21](#_Toc66361878)

[4 Réalisation 21](#_Toc66361879)

[4.1 Dossier de Réalisation 21](#_Toc66361880)

[4.2 Modifications 21](#_Toc66361881)

[5 Tests 21](#_Toc66361882)

[5.1 Dossier des tests 21](#_Toc66361883)

[6 Conclusion 22](#_Toc66361884)

[6.1 Bilan des fonctionnalités demandées 22](#_Toc66361885)

[6.2 Bilan de la planification 22](#_Toc66361886)

[6.3 Bilan personnel 22](#_Toc66361887)

[7 Divers 22](#_Toc66361888)

[7.1 Journal de travail 22](#_Toc66361889)

[7.2 Bibliographie 22](#_Toc66361890)

[7.3 Webographie 22](#_Toc66361891)

[8 Annexes 22](#_Toc66361892)

# 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.

### Si le temps le permet …

Objectifs complémentairesau cas où le projet n’est pas assez ambitieux dans le temps imparti…

### 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

Ce paragraphe présente le planning d'origine (date de début, date de fin, vacances et congés, liste hiérarchique des tâches ou GANTT, jalons, durée totale)

Toutes les mises à jour subies par le planning sont à reporter (avec date de mise à jour) et peuvent déboucher sur plusieurs versions de plannings.

# 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

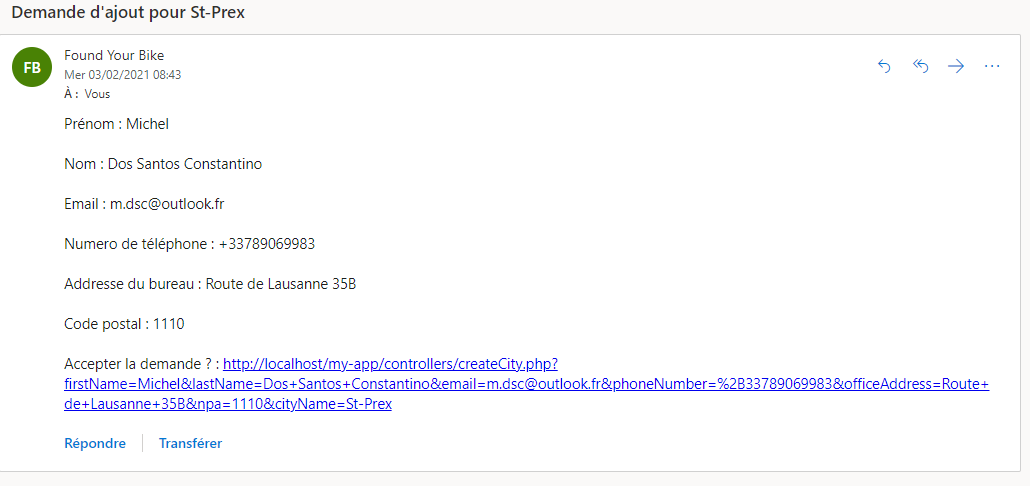
Les communes souhaitant s’inscrire sur le site devront d’abord faire une demande d’ouverture de compte à l’administrateur du site. Pour ce faire il faudra cliquer sur le lien en dessous du formulaire de login comme ci-dessous.



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.

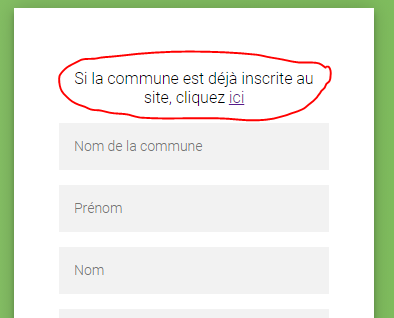


Figure 2: Pour accéder au 2ème formulaire pour compte de recherche

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.

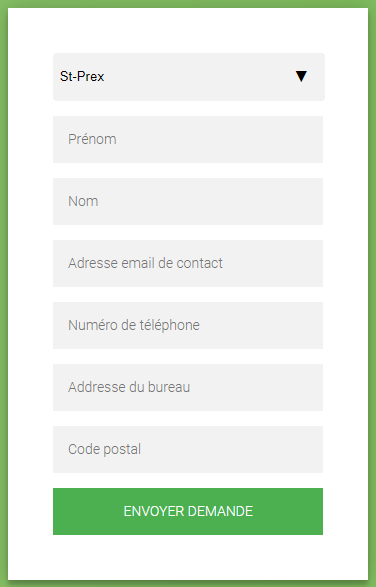


Figure 3: Formulaire pour compte de recherche

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)

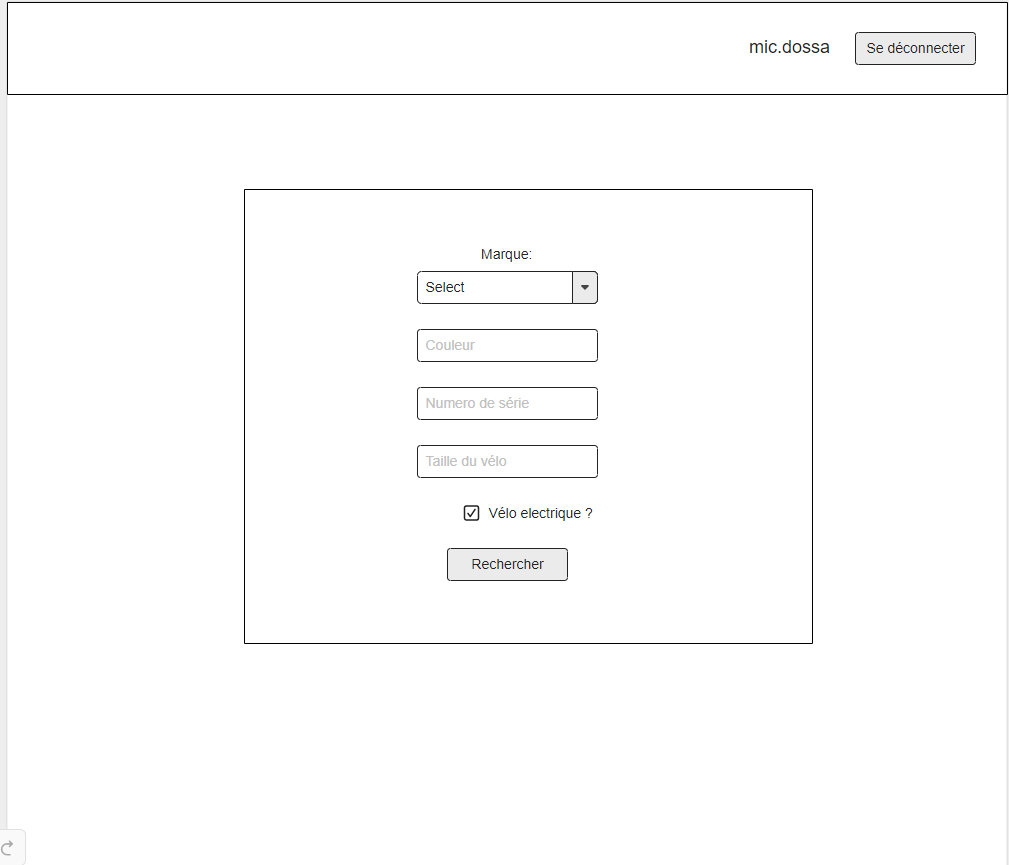


Figure 5: Formulaire de recherche

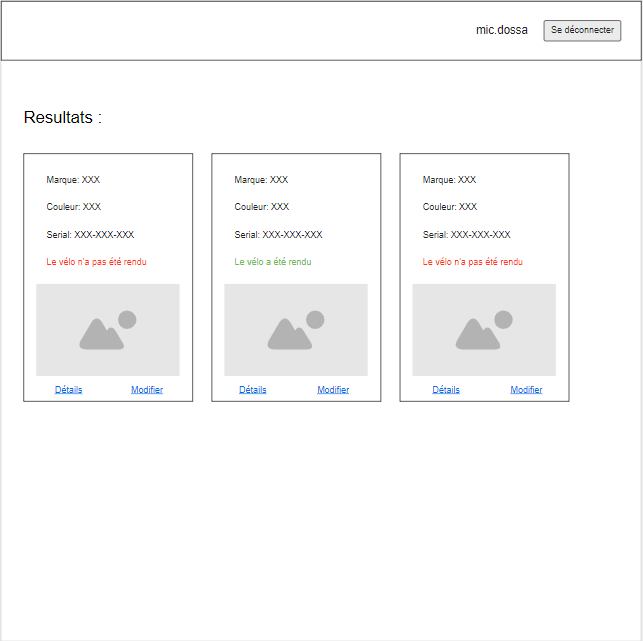


Figure 6: Page des résultats

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.

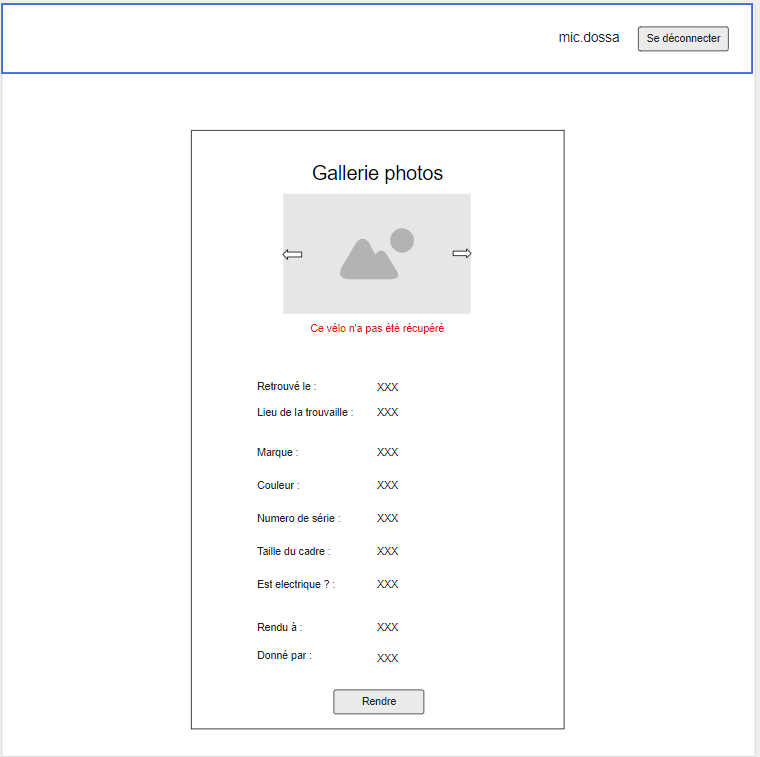


Figure 7: Mockup de la page détail d'un vélo qui n’a pas été rendu

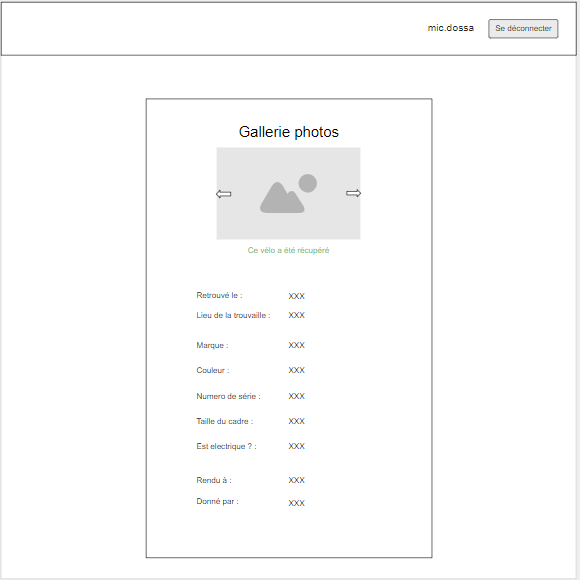


Figure 8: Mockup de la page détail d'un vélo qui a été rendu

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.

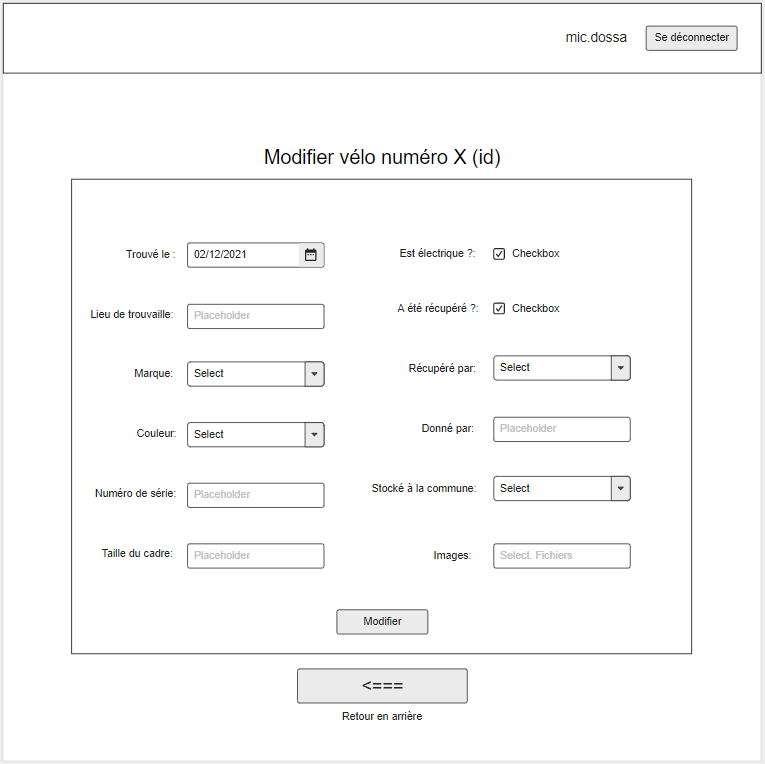


Figure 12: Mockup de la page de modification

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 :

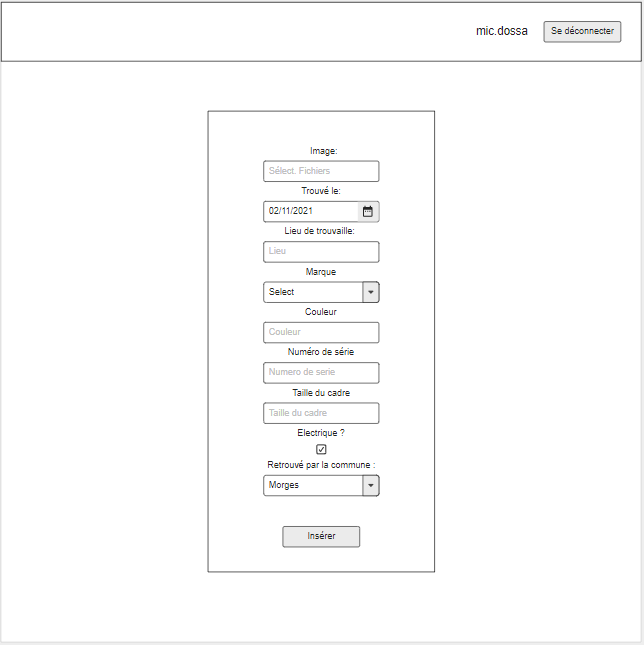
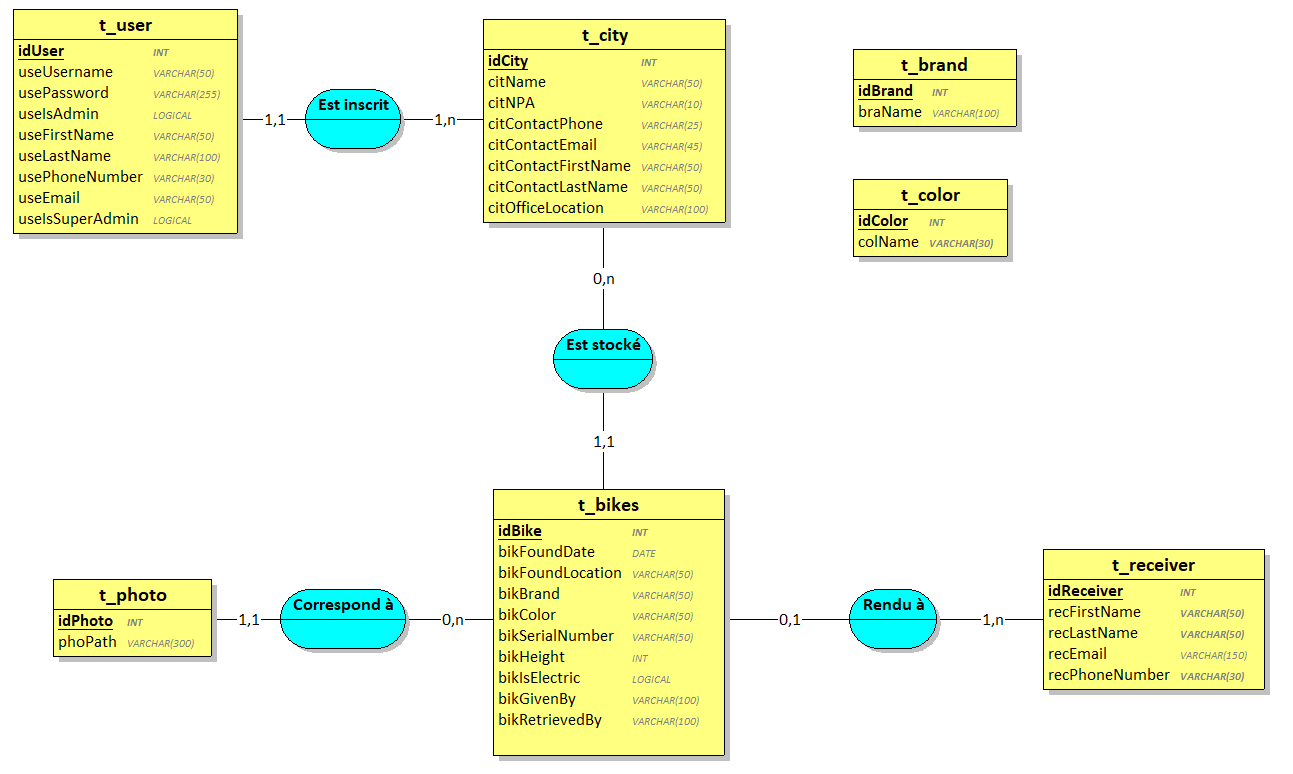


Figure 13: Mockup de la page pour ajouter un vélo

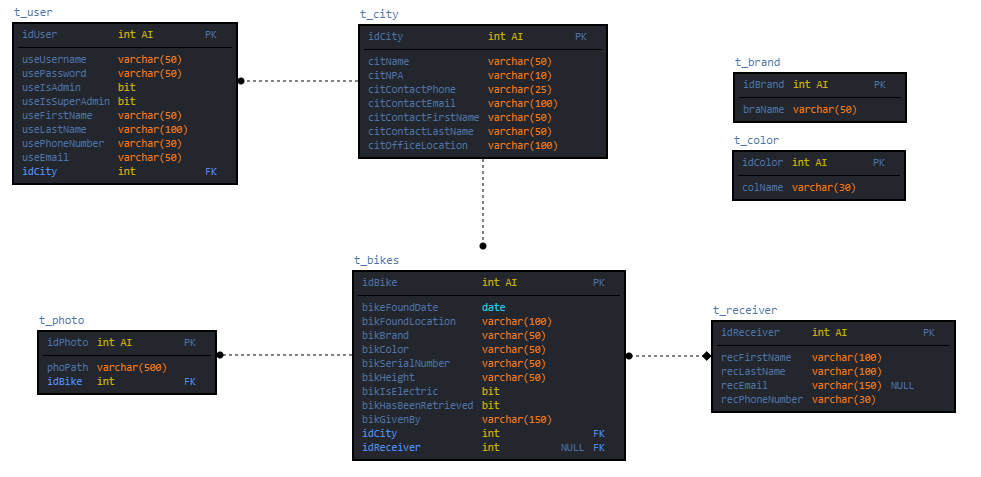
Sur ce formulaire il est possible d’uploader une ou plusieurs photos du vélo qui seront stockées dans le dossier « img/bike\_photos » à la racine du site. Après l’upload de ces photos sur le serveur, le chemin de celles-ci seront stockées dans la table t\_photo avec comme clé étrangère le vélo associé. 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 |  |  |

Ce paragraphe décrit le fonctionnement de manière détaillée.

Autant que possible de manière graphique, imagée, tableaux, etc.

Tous les cas particuliers devraient y être spécifiés…

Il s’agit d’y présenter les fonctionnalités à développer :

Découpage en étapes, en modules, en fonctionnalités, etc.

Formulaires, interfaces graphiques, pages web, etc.

Schémas de navigation, schémas événementiels, structogramme, pseudocode, etc.

Si le projet inclut une base de données :

Dictionnaire des données

Modèle conceptuel des données, modèles logique des données

## 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

Ce paragraphe permet de spécifier la stratégie de test qui sera menée au point 5.1

Qui, quand, avec quelles données, dans quel ordre, etc.

## Planification détaillée

A ce stade, après l’analyse complète du projet, un planning détaillé et complet (avec tâches, sous-tâches, dépendances, durée, …) peut être finalisé.

Le planning détaillé doit s’inscrire dans le planning initial. Il faut que l’on puisse situer cette planification détaillée par rapport à la planification initiale.

# Réalisation

## Dossier de Réalisation

Cette partie permet de reproduire ou reprendre le projet par un tiers.

Pour chaque étape, il faut décrire sa mise en œuvre. Typiquement :

Versions des outils logiciels utilisés (OS, applications, pilotes, librairies, etc.)

Configurations spéciales des outils (Equipements, PC, machines, outillage, etc.)

Code source commenté des éléments logiciels développés.

Modèle physique d’une base de données.

Arborescences des documents produits.

Il faut décrire le parcours de réalisation et justifier les choix.

## Modifications

Historique des modifications demandées (ou nécessaires) aux spécifications détaillées.

Date, raison, description, etc.

# Tests

## Dossier des tests

On dresse le bilan des tests effectués (qui, quand, avec quelles données…) sous forme de procédure. Lorsque cela est possible, fournir un tableau des tests effectués avec les résultats obtenus et les actions à entreprendre en conséquence (et une estimation de leur durée).

Si des tests prévus dans la stratégie n'ont pas pu être effectués :

raison, décisions, etc.

Liste des bugs répertoriés avec la date de découverte et leur état:

Corrigé, date de correction, corrigé par, etc.

# Conclusion

## Bilan des fonctionnalités demandées

Il s’agit de reprendre point par point les fonctionnalités décrites dans les spécifications de départ et de définir si elles sont atteintes ou pas, et pourquoi.

Si ce n’est pas le cas, estimer en « % » ou en « temps supplémentaire » le travail qu’il reste à accomplir pour terminer le tout.

## Bilan de la planification

Distinguer et expliquer les tâches qui ont généré des retards ou de l'avance dans la gestion du projet. Indiquer les différence entre les planifications initiales et détaillées avec le journal de travail.

## Bilan personnel

Si c’était à refaire:

Qu’est-ce qu’il faudrait garder ? Les plus et les moins ?

Qu’est-ce qu’il faudrait gérer, réaliser ou traiter différemment ?

Qu’est que ce projet m’a appris ?

Suite à donner, améliorations souhaitables, …

Remerciements, signature, etc.

# Divers

## Journal de travail

Date, activité (description qui permet de reproduire le cheminement du projet), durée, liens et références sur des documents externes. Lorsqu’une activité de recherches a été entreprise, il convient d’énumérer ce qui a été trouvé, avec les références.

## Bibliographie

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

## Webographie

Références des sites Internet consultés durant le projet.

# Annexes

Listing du code source (partiel ou, plus rarement complet)

Guide(s) d’utilisation et/ou guide de l’administrateur

Etat ou « dump » de la configuration des équipements (routeur, switch, robot, etc.).

Extraits de catalogue, documentation de fabricant, etc.