

Groupe 5

Verriest Jordan
Infelta Elliot
Zielinski Lucas

Documentation Technique

Missions n°3 -Sprint 1

Vue d'ensemble

Afin de simplifier la mise à jour du site par les employés de l'agence immobilière, il a été décidé d'enregistrer les biens dans une bdd.

Les fonctionnalités actuelles du site doivent persister.

Tâches à réaliser Sprint 1

1. **Un agent immobilier doit pouvoir se connecter et voir "son" menu**
2. **Un agent immobilier doit pouvoir ajouter un bien (sans les images)**
3. **Un agent doit pouvoir visualiser tous les biens (sans les images)**

Langages utilisés

Pour effectuer nos différentes tâches, nous avons utilisé du PHP et du mysql.

Nous utilisons phpMyAdmin pour gérer notre base de données appelée "VIZImmo"

Voici le concepteur de notre bdd :



Nous utilisons trois tables dans notre bdd :

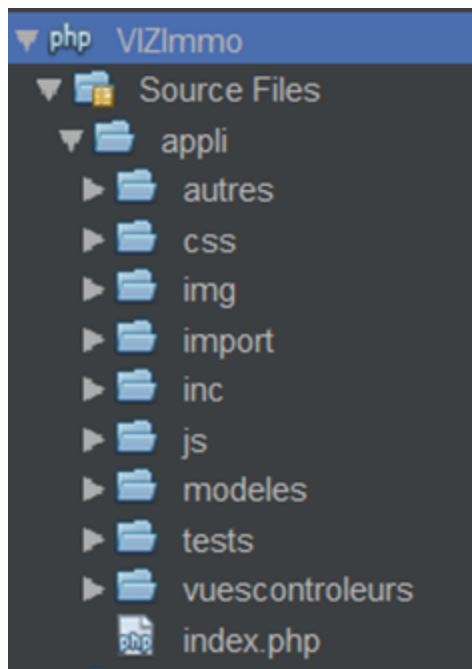
Dans la table utilisateur, il y a l'id (identifiant) en auto incrémentation et en clé primaire, le login représentant le pseudonyme de l'utilisateur ainsi que son mdp non affiché en clair (voir section sur la page php addUser.php).

Dans la table bien, il y a la ref (référence) représentant son identifiant, par exemple la maison n°5 correspondra à la référence M5, la rue composé du numéro et de la rue où se situe le bien, le cp (code postal), la ville, le prix du bien ainsi que son type variant ici de 1 à 5 qui est une clé étrangère relié à l'id de la table typebien.

Enfin dans la table typebien, il y a l'id qui est en clé primaire et relié à la clé étrangère type de la table bien, ainsi que le libellé représentant le type de bien comme un appartement, une maison, un local, un immeuble ou un terrain.

La colonne type dans la table bien est reliée avec l'id de la table typebien afin de permettre une meilleure compréhension à l'affichage d'un bien. En effet, au lieu d'afficher 1, il affichera appartement.

Arborescence des fichiers



Pour cette mission nous avons une arborescence imposée qui est montrée ci-dessus

Notre projet contient donc les fichiers : autres (qui contiennent les fonctions diverses pour l'instant absentes), le fichier css (qui contient les styles css), le fichier img (qui contient les images utilisées dans le site), le fichier import (qui contient les fichiers javascript importés), le fichier inc qui contient les includes, le fichier js qui contient les scripts js, modèles qui contiennent nos fonctions d'accès à la bdd, le fichier tests qui contiennent les différents tests de nos fonctions, vuescontroleurs qui contiennent toutes les pages du site et enfin la page index.php qui est la page d'accueil de notre site.

Nomenclature des fichiers et description

On garde les noms de fichiers de l'ancienne mission en remplaçant le .html par le .php

-La page ajoutBien.php est la page front-end permettant à un agent d'ajouter un bien dans la base de donnée. Cette page est visible que par les agents immobiliers se connectant. Elle comporte des explications de ce qu'il faut remplir afin que le bien soit ajouté. De plus, celle-ci comporte un formulaire permettant de saisir les différentes informations essentielles et nécessaires au bon fonctionnement de l'ajout d'un bien dans la base de données.

-La page controleSaisiAjout.php est une page essentiel au bon fonctionnement de l'ajout d'un bien depuis la page ajoutBien.php. Celle-ci permet d'effacer les espaces quand il n'y en a pas besoin, l'utilisation de la fonction htmlspecialchars, la fonction TailleMax permettant le contrôle de la saisit n'excède pas la taille imposé dans la base de donnée, la fonction testRef permettant de vérifier que la référence saisie n'est pas déjà utilisé ainsi que la fonction ajouteBien permettant d'ajouter un bien dans la base de donnée.

Toutes les fonctions saisies auparavant sont contenues dans le modeles dans mesFonctionsAccesAuxDonnees.php.

-La page testRef.php est une page contenu dans le fichier test qui permet de tester la fonction testRef contenu dans mesFonctionsAccesAuxDonnees.php, elle permet de vérifier si la référence à déjà été saisi ou non.

-La page `testdonneLesTypesDeBiens.php` est une page contenu dans le fichier `test` qui permet de tester la fonction `donneLesTypesDeBiens` contenu dans `mesFonctionsAccesAuxDonnees.php`, elle permet de récupérer non pas un chiffre correspondant à un libellé d'un type de bien mais bien le libellé associé au type du bien.

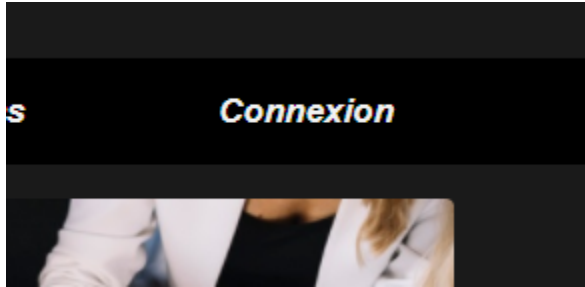
-La page `testajouterBien.php` est une page contenu dans le fichier `test` qui permet de tester la fonction `ajouterBien` contenu dans `mesFonctionsAccesAuxDonnees.php`, elle permet de vérifier que dès la saisie d'informations d'un bien valides par rapport à la base de donnée, alors celle-ci l'ajoute dans la base de donnée

-La page `addUser.php` permet d'ajouter un utilisateur dans la table `utilisateur`, elle renvoie à la bdd un id qui est implémenté automatiquement, un login de 20 caractère maximum et un mot de passe qui est hashé grâce à la fonction `password_hash()` (c'est pour ça que dans la bdd on a mis une grosse limite sur les caractère car un mot de passe hashé est beaucoup plus long).

-La page `testVerifieAuthentification` permet de tester la fonction qui vérifie les valeurs saisis lors de la connexion d'un agent.

Aperçu des pages et explication de celle-ci

Sur chaque page le menu contient désormais un bouton connexion qui permet à un agent de se connecter :

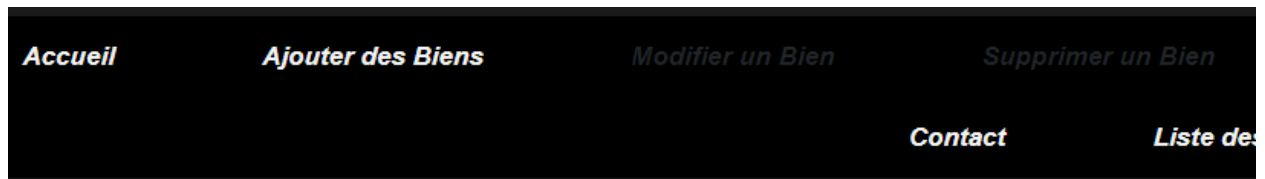


Ce bouton permet d'amener un agent voulant se connecter vers une page de connexion :

A screenshot of a login form titled 'Connexion' in a large, orange, italicized serif font. Below the title, there are two input fields. The first is labeled 'Nom d'utilisateur' and contains the placeholder text 'Entrer le nom d'utilisateur'. The second is labeled 'Mot de passe' and contains the placeholder text 'Entrer le mot de passe'. At the bottom of the form, there is a button labeled 'Connexion' in a white, italicized serif font.

Celle-ci permet de rentrer son login et son mot de passe (avec un contrôle de saisie)

Puis si la saisie est bonne elle renvoie vers l'accueil avec un menu différent pour l'agent connecté :



La page ajouter des Biens est maintenant disponible celle-ci permet d'ajouter des biens à la bdd grâce à un formulaire (avec contrôle de saisie grâce à la page controleSaisie.php) et la fonction ajouterBien :

Ajouter de nouveaux bien dans VIZIMMO !

Pour ajouter un nouveau bien dans notre nouvelle base de donnée, il vous faut y renseigner :

- La référence du bien, par exemple si c'est le local n°15, la référence serait L 15
- La rue du bien, qu'il y renseigne le numéro de rue ainsi que la rue où se situe le bien.
- Le code postal de la ville où se situe le bien.
- La ville où se situe le bien.
- Le prix du bien en euros (€)
- Le type du bien, allant de 1 à 5 :
 - Le numéro 1 pour un Appartement
 - Le numéro 2 pour un Immeuble
 - Le numéro 3 pour un local
 - Le numéro 4 pour une maison
 - Le numéro 5 pour un terrain

Form fields:

Reference:

Rue:

Code Postal:

Ville:

Prix:

Type de bien:

Valider l'ajout du bien

Coordonnées
N° de téléphone : 03 20 25 40 75

Et enfin tout utilisateurs doit pouvoir voir tous les biens grâce à la page Lister les biens et la fonction donneLesBiens(), celle-ci liste les biens qui se trouve dans la bdd

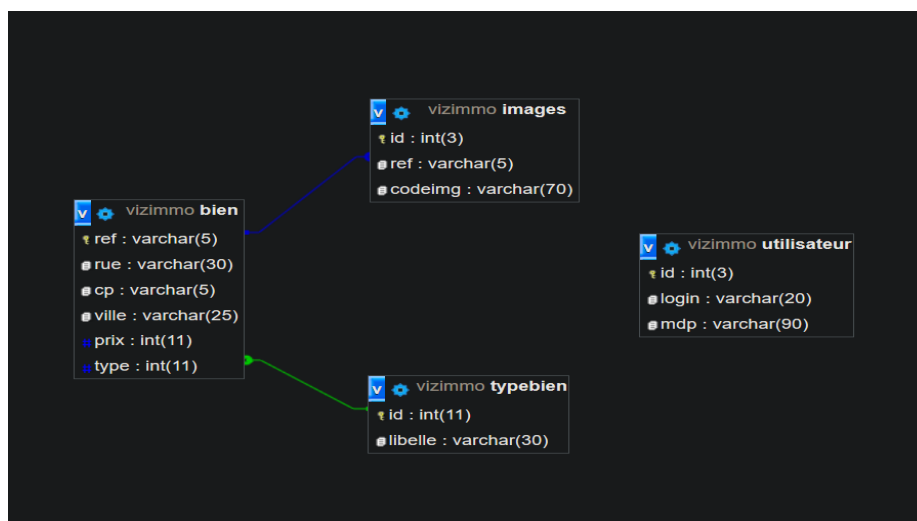
Tâches à réaliser Sprint 2

4. **Un utilisateur doit pouvoir faire une recherche de biens en fonction du type de bien et/ou d'une tranche de prix**
5. **Un utilisateur doit pouvoir visualiser tous les biens (réf, ville, type de bien, les images du bien)**
6. **Un agent immobilier doit pouvoir modifier les informations d'un bien existant (sans les images)**

Dans un premier temps, nous avons dû mettre à jour la base de données contenu dans phpMyAdmin. En effet, nous avons dû créer une nouvelle table, images.

Dans celle-ci, il y a l'id (identifiant) en clé primaire, la réf (référence) en clé étrangère faisant référence à la table bien qui a pour clé primaire réf, ainsi que codeimg représentant le lien de l'image qui va être affiché.

Ainsi dans cette table images, nous avons inscrits 59 liens d'image soit environ 3 images par bien présenté dans le site VIZImmo.



Modifier un bien existant

Pour cette user story, j'ai utilisé 3 fonctions contenu dans `mesFonctionsAccesAuxDonnees` dans le dossier `modeles` :

`modifierBien()`, celle-ci possède 7 attributs comprenant la connexion à la base de données, la rue, le code postal, la ville, le prix, son type ainsi que sa référence du bien, me permettant de modifier un Bien selon les informations saisies par l'utilisateur.


`donneLeBien()`, celle-ci possède 2 attributs comprenant la connexion à la base de données et la référence du bien, me permettant d'afficher un seul bien selon la référence choisie par l'utilisateur.

`donneLesRefs()`, celle-ci possède un seul critère, la connexion à la base de données, me permettant d'obtenir toutes les références de la base de données de la table bien.

Ces 3 fonctions sont accompagnées de 3 tests respectifs, `testmodifierBien.php`, `testDonneLeBien.php` et `testDonneLesRefs.php` toutes les 3 contenus dans le dossier `tests`.

Par la suite, j'ai créé 3 nouvelles pages PHP se prénommant `modifierBien.php`, `voirLeBienAModifier.php` et `controleSaisiModifier.php` contenu toutes les 3 dans le dossier `vuecontroleurs`.

Dans un premier temps, la page `modifierBien.php` va permettre à l'utilisateur de saisir une référence parmi toutes les autres références utilisées dans la base de données via la fonction `donneLesRefs()`.



Accueil Ajouter des Biens Modifier un Bien Supprimer un Bien Maison Appartement Locaux Immeuble

Terrain nu Contact Liste des Biens Recherche de Biens Déconnexion

Modifiez des biens dans VIZImmo !


Pour modifier un bien dans notre nouvelle base de donnée, il vous faut d'abord saisir la référence :

- A correspondant à Appartement
- I correspondant à Immeuble
- L correspondant à Local
- M correspondant à Maison
- T correspondant à Terrain

Reference :

Coordonnées

- N° de téléphone : 03.20.25.40.75
- Adresse e-mail : VIZImmo@gmail.com
- Notre agence est située au 67 rue du Marechal Leclerc à Lys-Lez-Lannoy (59390)



Après la saisie de la référence par l'utilisateur, la page `modifierBien.php` va envoyer l'utilisateur vers la page `voirLeBienAModifier.php` permettant de voir les informations du bien contenu dans un formulaire afin qu'ainsi, l'utilisateur puisse les modifier en utilisant la fonction `donneLeBien()`.

Modifiez des biens dans VIZImmo !

Bien vous avez saisi la référence, maintenant, il vous suffit de renseigner plusieurs informations :

- La nouvelle rue du bien, qui y renseigne le nouveau numéro de rue ainsi que la nouvelle rue où se situe le bien.
- La nouvelle ville où se situe le bien.
- Le nouveau code postale de la ville où se situe le bien.
- Le nouveau prix du bien en euros (€).
- Le nouveau type du bien, allant de 1 à 5 :
 - Le numéro 1 pour un Appartement
 - Le numéro 2 pour un Immeuble
 - Le numéro 3 pour un local
 - Le numéro 4 pour une maison
 - Le numéro 5 pour un terrain

Ref :

Rue :

Ville :

Code Postal :

Prix :

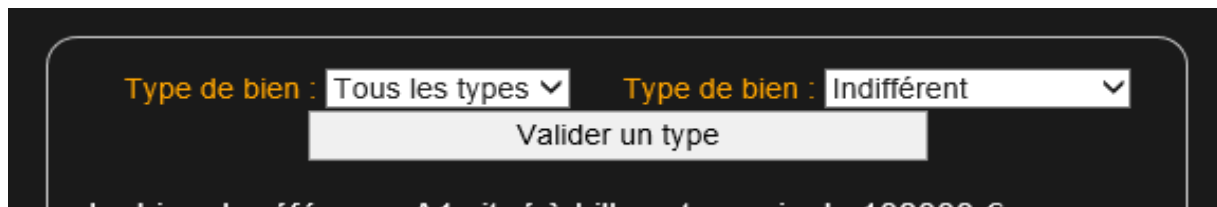
Type de bien :

Enfin, quand l'utilisateur va appuyer sur "Valider la modification du bien", celle-ci va va alors envoyer les saisies dans la page controleSaisiModifier.php qui va, comme son nom l'indique, va contrôler les saisies si elles respectent les informations de saisie de la base de données. Si toutes les saisies sont correctes, celle-ci va alors utiliser la fonction modifierBien().

Recherche de bien par critère

Pour cette user story j'ai utilisé une fonction rechercheBien(), celle-ci possède 3 attributs, la connexion à la BDD, le type de biens et le budget choisi.

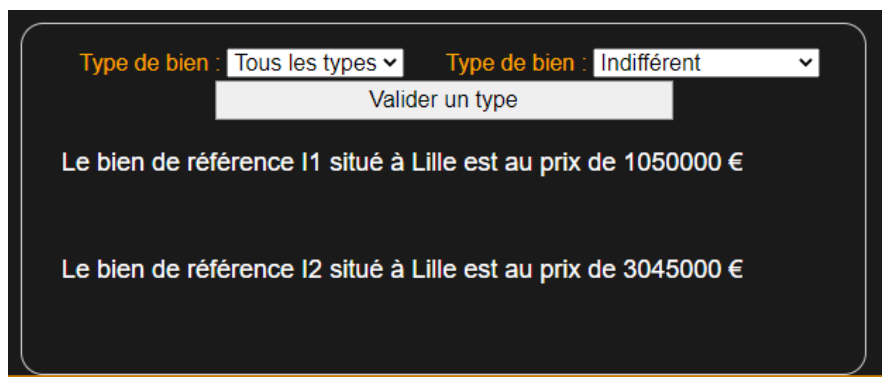
Pour récupérer ces informations nous avons une page avec formulaire dans lequel on doit sélectionner le type de bien et la tranche de prix que l'on souhaite afficher.



Ensuite les informations sont récupérées puis grâce à la fonction rechercheBien() on recherche les biens qui correspondent aux critères sélectionnés puis ils sont affichés.

De plus les types de biens sont récupérés via la fonction donneLesTypesDeBiens(),

Exemple (lorsque l'on choisit Immeubles avec prix indifférent).



Pour tester cette fonction il y a la fonction testRechercheBien qui teste les recherches avec 2,1 ou aucuns critères.

Visualiser tous les biens avec les images

Pour cette user story j'ai tout d'abord eu besoin de créer la fonction donnerLesImages(), celle-ci me permettait d'aller chercher le chemin de l'image que j'ai placé dans la base de donnée. La base de donnée contenait en clé primaire l'id, la référence du bien en clé étrangère et le codeimg qui correspondait au chemin de l'image. Sur la page ToutLesBiens, j'ai ajouté un href qui me permettait de m'emmener vers la page bien.php.

- [Le bien de référence A1, situé au 85 rue de Cambrai à Lille \(59000\) est un/une appartement au prix de 108000 €](#)

La page bien.php générait ensuite une page selon la ref de l'image.



Enfin sur la page bien.php, on affiche les informations du bien grâce à la fonction `donneLeBien()`. L'affichage des images se fait avec la fonction `donneLesImages()` et le défilement avec le script javascript.

Tâches à réaliser Sprint 3

7. **Un agent immobilier doit pouvoir ajouter un bien (réf, adresse, type de bien, prix, les images du bien)**
8. **Un utilisateur doit pouvoir faire une recherche de biens en fonction du type de bien et/ou d'une tranche de prix et/ou d'une surface minimum, et/ou de la présence d'un jardin ou pas, et/ou du nombre de pièces**
9. **Un agent immobilier doit pouvoir supprimer un bien existant**

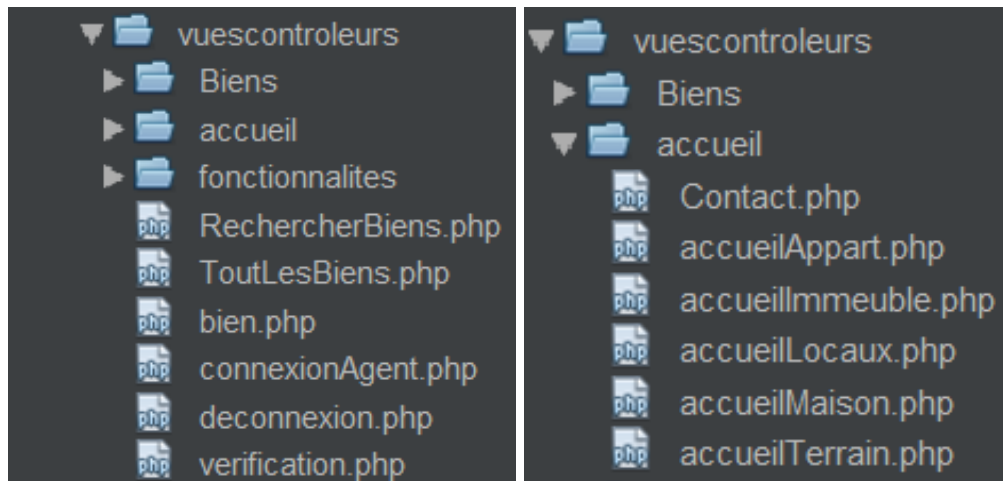
Pour ce sprint n°3, nous avons dû ajouter 3 colonnes dans la table bien de notre base de données. La colonne jardin, surface et nbpiece.

Jardin est en boolean, retournant 1 s'il ya un jardin sinon 0.

Surface est en integer, avec une taille maximum de 4 représentant la surface habitable du bien en mètre carré.

Nbpiece est en integer, avec une taille maximum de 3 représentant le nombre de pièce que contient le bien.

Nous avons aussi réorganisé l'arborescence du dossier vuescontroleurs. Nous avons créé 2 dossiers, l'un "accueil" regroupant toutes les pages d'accueil de bien ainsi que la page contact, l'autre "fonctionnalités" regroupant toutes les pages php permettant d'ajouter, modifier et supprimer.



Ajouter un bien avec les images

Pour cette user story, on devait pouvoir sélectionner des images lors de l'ajout d'un bien pour qu'elles s'ajoutent dans la base de données et qu'elles se transfèrent automatiquement dans les fichiers.

Tout d'abord, j'ai créé la fonction `ajouterImage`, celle-ci ajoute à la BDD la ref l'id et le codeimg.

Ensuite il fallait une autre fonction `getID` qui nous permettait de récupérer l'id MAX pour que l'image que l'on ajoute corresponde à l'id+1.

Pour pouvoir sélectionner les images, on ajoute dans le formulaire un bouton qui permet de sélectionner une ou plusieurs images. Lorsqu'on valide la saisie, cela nous renvoie vers la page `controleSaisiAjout`.

Dans le `controleSaisiAjout.php`,

On crée une boucle qui permet de traiter toutes les images sélectionnées,

```
$countfiles = count($_FILES['monfichier']['name']);
```

le \$countfiles permet de compter le nombre d'images ajoutés,

```
for($i=0;$i<$countfiles;$i++){
```

On vérifie s'il n'y a pas d'erreur dans le téléchargement des images,

```
if($_FILES['monfichier']['error'][$i] == 0){
```

S'il n'y en a pas alors on vérifie si la taille est inférieure à 500Ko,

```
// test taille
if($_FILES['monfichier']['size'][$i] > 500000){
    $erreur = 'Votre fichier est trop lourd';
}
```

On récupère alors les données de l'image grâce au formulaire, à

\$_FILES['monfichier']['donneesvoulu'] et à la fonction getID.php pour récupérer le dernier ID de la BDD,, puis on crée un fichier dans le dossier du type du bien avec comme nom la référence du bien, on y glisse ensuite l'image avec le move_uploaded_file

```
move_uploaded_file($_FILES['monfichier']['tmp_name'][$i], '../img/'.$type.'/'.$ref.'/'.$_FILES['monfichier']['name'][$i]);
```

et enfin on ajoute l'image à la base de donnée avec la fonction ajouterImage.

Effacer un bien

Pour cette user story, j'ai utilisé 2 fonctions contenu dans mesFonctionsAccesAuxDonnees dans le dossier modeles :

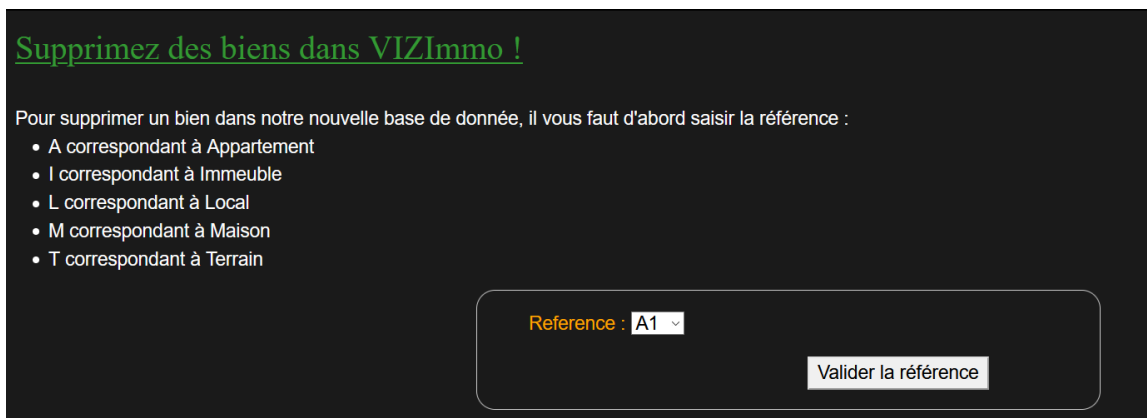
supprimerImages(), celle-ci possède 2 attributs comprenant la connexion à la base de données ainsi que la référence du bien.

supprimerBien(), celle-ci possède 2 attributs comprenant la connexion à la base de données ainsi que la référence du bien.

Ces 2 fonctions sont accompagnées de 2 tests respectifs, `testSupprimerImages.php` et `testSupprimerBien.php` toutes les 2 contenus dans le dossier `tests`.

Par la suite, j'ai créé 3 nouvelles pages PHP se prénommant `supprimerBien.php`, `voirLeBienASupprimer.php` et `controleSaisiSupprimer.php` contenu toutes les 3 dans le dossier `fonctionnalites`.

Dans un premier temps, la page `supprimerBien.php` va permettre à l'utilisateur de saisir une référence parmi toutes les autres références utilisées dans la base de données via la fonction `donneLesRefs()` créée dans le sprint n°2.



The screenshot shows a web form with a dark background. At the top, the title 'Supprimez des biens dans VIZImmo !' is written in green. Below it, a text prompt asks the user to enter a reference. A bulleted list provides options: 'A' for Apartment, 'I' for Building, 'L' for Local, 'M' for House, and 'T' for Land. At the bottom, there is a text input field with 'Reference : A1' and a 'Valider la référence' button.

Supprimez des biens dans VIZImmo !



Pour supprimer un bien dans notre nouvelle base de donnée, il vous faut d'abord saisir la référence :

- A correspondant à Appartement
- I correspondant à Immeuble
- L correspondant à Local
- M correspondant à Maison
- T correspondant à Terrain

Reference : A1

Valider la référence

Après la saisie de la référence par l'utilisateur, la page `supprimerBien.php` va envoyer l'utilisateur vers la page `voirLeBienASupprimer.php` permettant de voir les informations du bien contenu dans un formulaire sans qu'il puisse les modifier.

Ref :	A1
Rue :	85 rue de Cambrai
Ville :	Lille
Code Postal :	59000
Prix :	108000
Type de bien :	appartement
Jardin :	Non
Surface :	17
Nombre de pièces :	2
Images :	
	
Valider la suppression du bien	

Enfin, quand l'utilisateur va appuyer sur "Valider la suppression du bien", celle-ci va alors envoyer les informations de ce bien dans la page `controleSaisiSupprimer.php` qui va, dans un premier temps, vérifier la longueur de saisie des informations du bien.

Si ses informations sont à la taille requise, elle va alors se connecter à la base de données. Par la suite, elle teste si la référence du bien existe bien dans la base de données. Si elle existe alors, elle va utiliser la fonction `supprimerImages()` pour supprimer les images du bien saisi par sa référence. Si la fonction renvoie TRUE, alors elle va cette fois-ci utiliser `supprimerBien()` pour supprimer le bien ainsi que ses informations. Enfin si la fonction renvoie TRUE, alors la page `controleSaisieSupprimer.php` va renvoyer l'utilisateur à la page `supprimer.php`. Dans le cas contraire, un message d'erreur sera envoyé à l'utilisateur.

Il est primordial de supprimer les images du bien en premier car cette information est contenu dans une table faisant référence à la table bien. Il est donc obligatoire de supprimer toutes les données faisant référence à ce bien puis à le supprimer entièrement, sinon celui-ci ne pourra pas être supprimé.

Rechercher des biens avec plus de critères

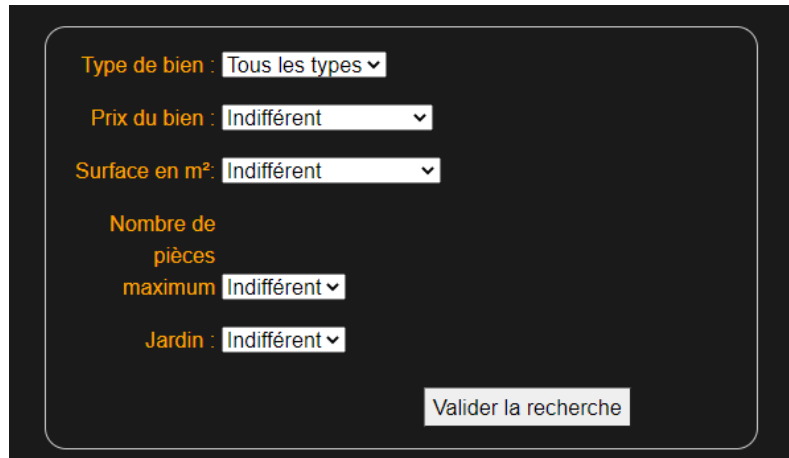
Cette user story avait pour but d'approfondir ce qui a été fait dans le sprint précédent, on reste sur de la recherche de biens mais avec plus de critères.

Pour faire cela, il a fallu modifier la table biens en ajoutant les nouveaux critères : la surface, le nombre de pièces et l'existence d'un jardin.

<input type="checkbox"/>	7 jardin	tinyint(1)
<input type="checkbox"/>	8 surface	int(6)
<input type="checkbox"/>	9 nbpiece	int(3)

Ensuite il a fallu modifier la fonction rechercheBien en ajoutant donc les nouveaux critères comme attribut, puis il a fallu juste reprendre la même logique que pour le sprint 2 en vérifiant que le critère a bien été sélectionner pour afficher les biens qui correspondent à la recherche, puis en fonction des critères choisis, ceux-ci seront permettront grâce aux différents IF de concaténer dans une variable la requête correspondant à la recherche.

Ensuite il a fallu modifier la page de recherche de biens pour pouvoir sélectionner les nouveaux critères :



Search form with the following filters:

- Type de bien : Tous les types ▾
- Prix du bien : Indifférent ▾
- Surface en m²: Indifférent ▾
- Nombre de pièces maximum Indifférent ▾
- Jardin : Indifférent ▾
- Button: Valider la recherche

Puis après que l'utilisateur ait sélectionné les critères qu'ils souhaitent ceux-ci s'afficheront comme cela :

Le bien de référence A1 situé à Lille est au prix de 108000 €
Le bien de référence A2 situé à Roubaix est au prix de 81750 €
Le bien de référence A3 situé à Wattrelos est au prix de 199000 €
Le bien de référence A4 situé à Croix est au prix de 155000 €
Le bien de référence A5 situé à Hem est au prix de 155000 €
Le bien de référence A6 situé à Roubaix est au prix de 78000 €
Le bien de référence A7 situé à Lille est au prix de 483000 €
Le bien de référence I1 situé à Lille est au prix de 1050000 €
Le bien de référence I2 situé à Lille est au prix de 3045000 €

Tâches à réaliser Sprint 4

10. **Un agent doit pouvoir ajouter un bien (réf, adresse, type de bien, prix, les images du bien, surface, le nombre de pièces, jardin)**
11. **Un utilisateur doit pouvoir visualiser tous les biens (réf, adresse, type de bien, prix, les images du bien, surface, le nombre de pièces, jardin)**
12. **Un agent immobilier doit pouvoir modifier toutes les informations d'un bien existant**

Pour ce sprint, l'arborescence a été modifiée car nous n'avons plus besoin des anciennes pages de biens ainsi que de l'accueil avec la liste de ceux-ci.

Ajouter un bien (surface, nombre de pièces et jardin)

Pour cette userstory, j'ai ajouté dans la fonction ajouterBien les variables jardin, surface et nbpiece.

```
$bvc7=$monObjPdoStatement->bindValue(':jardin',$jardin);  
$bvc8=$monObjPdoStatement->bindValue(':surface',$surface);  
$bvc9=$monObjPdoStatement->bindValue(':nbpiece',$nbpiece);
```

Il a fallu ensuite ajouter dans le formulaire de ajoutBien.php des cases pour le jardin la surface et le nombre de pièce avec la sélection du jardin sous forme de liste,

```
<div>
  <label for="jardin">Jardin :</label>
  <select id="jardin" title="Choisissez s'il y'a ou non un jardin"
    name="lesjardins">
    <option value="1">Il y a un jardin</option>
    <option value="0">Il n'y a pas de jardin</option>
  </select>
</div>
<div>
  <label for="surface">Surface :</label>
  <input type="text" id="surface" name="surface">
</div>
<div>
  <label for="nbpiece">Nombre de pièces :</label>
  <input type="text" id="nbpiece" name="nbpiece">
</div>
<div>
```

Enfin, j'ai ajouté dans le controleSaisiAjout.php la vérification du format des nouvelles données rentrées et l'ajout de ces données dans la BDD.

Visualiser tous les biens (surface,nombre de pièces et jardin)

Pour cette userstory, il a juste fallu ajouter dans la page bien.php les variables contenant les informations nécessaires (la surface, le nombre de pièces et l'existence d'un jardin), la seule différence avec le sprint 2 est que pour le jardin on utilise un if qui permet d'afficher ou non si il y a un jardin en fonction des informations dans la BDD.

Modifier toutes les informations d'un bien existant

Pour cette user story, j'ai utilisé 3 fonctions contenu dans mesFonctionsAccesAuxDonnees dans le dossier modeles :

modifierBien(), une fonction qui était déjà effective. Cependant, j'ai dû la modifier afin d'y ajouter 3 nouvelles informations, le jardin, la surface ainsi que le nombre de pièces du bien.

supprimerUnelImage(), celle-ci possède 2 attributs comprenant la connexion à la base de données ainsi que l'identifiant du bien.

donneUnelImage(), celle-ci possède 2 attributs comprenant la connexion à la base de données ainsi que l'identifiant du bien.

Ces 3 fonctions sont accompagnées de 3 tests respectifs, testmodifierBien.php, testSupprimerUnelImage.php et testDonneUnelImage.php toutes les 3 contenus dans le dossier tests.

Par la suite, j'ai modifié 2 pages PHP se prénommant , voirLeBienAModifier.php et controleSaisiModifier.php contenu toutes les 3 dans le dossier fonctionnalités afin d'y ajouter les nouvelles informations soit le jardin, la surface et le nombre de pièces, les images pouvant être supprimé ainsi qu'un bouton permettant d'ajouter une image..

Dans un premier temps, la page modifierBien.php va permettre à l'agent de saisir une référence parmi toutes les autres références utilisées dans la base de données via la fonction donneLesRefs() créée dans le sprint n°2.

Modifiez des biens dans VIZImmo !

Pour modifier un bien dans notre nouvelle base de donnée, il vous faut d'abord saisir la référence :

- A correspondant à Appartement
- I correspondant à Immeuble
- L correspondant à Local
- M correspondant à Maison
- T correspondant à Terrain

Reference :

Valider la référence

Cette page emmène par la suite l'agent vers la page voirLeBienAModifier.php affichant toutes les informations du bien.

The screenshot shows a web form with the following fields and elements:

- Ref : T4
- Rue : 146 rue du Chemin des Vaches
- Ville : Wargnies le petit
- Code Postal : 59144
- Prix : 126000
- Type de bien : terrain (dropdown menu)
- Jardin : Il y a un jardin (dropdown menu)
- Surface : 900
- Nombre de pièces : 0
- Images :
 - A photograph of a field with trees in the background. Below it is a button labeled "Supprimer la photo n°58".
 - A technical drawing or map. Below it is a button labeled "Supprimer la photo n°59".
 - A button labeled "Parcourir..." followed by the text "Aucun fichi...lectionné".
- A button at the bottom right labeled "Valider la modification du bien".

Cette page permet de modifier toutes les informations du bien. De plus, elle permet de supprimer une ou plusieurs images ainsi que l'ajout d'autres images pour le bien.

Dès l'appuies sur le bouton 'Supprimer la photo n°x', cette page va faire le traitement de l'information sur la page controleSaisiModifier.php où la page va utiliser la fonction supprimerUnelImage() selon l'id récupéré depuis la page voirLeBienAModifier.php.

De plus, dès l'appuies sur le bouton 'Parcourir...', l'agent va pouvoir ajouter une image en parcourant l'explorateur de fichier. Dès l'image validée, le nom de l'image va apparaître à la place du 'Aucun fichier sélectionné'.

Enfin, quand l'agent juge avoir fini la modification du bien, il doit appuyer sur 'Valider la modification du bien', un popup de confirmation va être demandé pour valider celle-ci. Si l'agent l'accepte, la page controleSaisiModifier.php va être appelée afin d'ajouter l'image via la fonction supprimerImage() s'il y a une nouvelle image de sélectionner ou va simplement modifier les informations du bien via la fonction modifierBien().

Par la suite, la page va rediriger l'agent vers le menu modifierBien.php.

Missions n°4

Un agent immobilier doit pouvoir réinitialiser son mot de passe :

Pour faire cela j'ai effectué des modifications dans la BDD (ajout d'une colonne mail dans la table utilisateur et une table recuperation)

Pour réinitialiser son mot de passe, l'agent immobilier peut cliquer sur mot de passe oublié sur la page de connexion, cela lui demandera son adresse mail puis après l'avoir saisi il pourra entrer le code de vérification et le nouveau mot de passe.

Le code de vérification ne peut pour l'instant pas être envoyer par mail à cause de soucis technique donc le code est affiché dans la barre de recherche du navigateur comme ceci :

← → ↻ localhost/V2Immo/appli/vuecontroleurs/verificationMdp.php?section=code&code=95725357

Applications Gmail YouTube Mon Compte Form... Account is disabled jor1c: OpenRISC O... tradu anglais - Rec... YureGieDHI Duel...

Réinitialiser votre mot de passe

Code de vérification

Nouveau Mot de passe

Créer une machine virtuelle permettant de visualiser le site VIZImmo :

Afin de répondre à cette problématique, nous avons dû dans un premier temps créer notre serveur linux permettant de stocker notre site ainsi que notre base de données.

Nous avons dû télécharger le module apache2, php, mysql mais aussi phpmyadmin.

Quand tout cela à été téléchargé, nous avons dû faire plusieurs manipulations :

- Aller dans dans : /var/www/html et faire le sudo git clone de notre projet
- Récupérer le sous dossier VIZImmo contenu dans le répertoire InfeltaVerriestZielinskiM3 et le déplacer dans le html donc sudo mv VIZImmo ../VIZImmo
- Supprimer le dossier InfeltaVerriestZielinskiM3

Par la suite nous avons dû mettre en place la base de données dans phpmyadmin ainsi que créer l'utilisateur :

- Dans le navigateur :

⚠ Non sécurisé | 192.168.0.30/phpmyadmin/

- Puis créer l'utilisateur APMission3 comme dans le sprint 1
- Créer la base de données vizimmo puis importer le fichier sql vizimmo contenu dans le dossier BDD

Enfin afin de visualiser le site, il faut saisir dans le navigateur :

⚠ Non sécurisé | 192.168.0.30/VIZImmo/appli/

Pour se connecter à la base de données :

login: APMission3

mdp: VIZImmo

Pour le site VIZImmo en tant qu'agent immobilier :

login: agent1

mdp: 020599

il faut d'abord l'ajouter le script avec addUser.php

NOTE:

Sur certaines pages il y a une bande blanche au bout de la page et malgré plusieurs essais impossible de la retirer.