BOTTON
Gaël

SIO

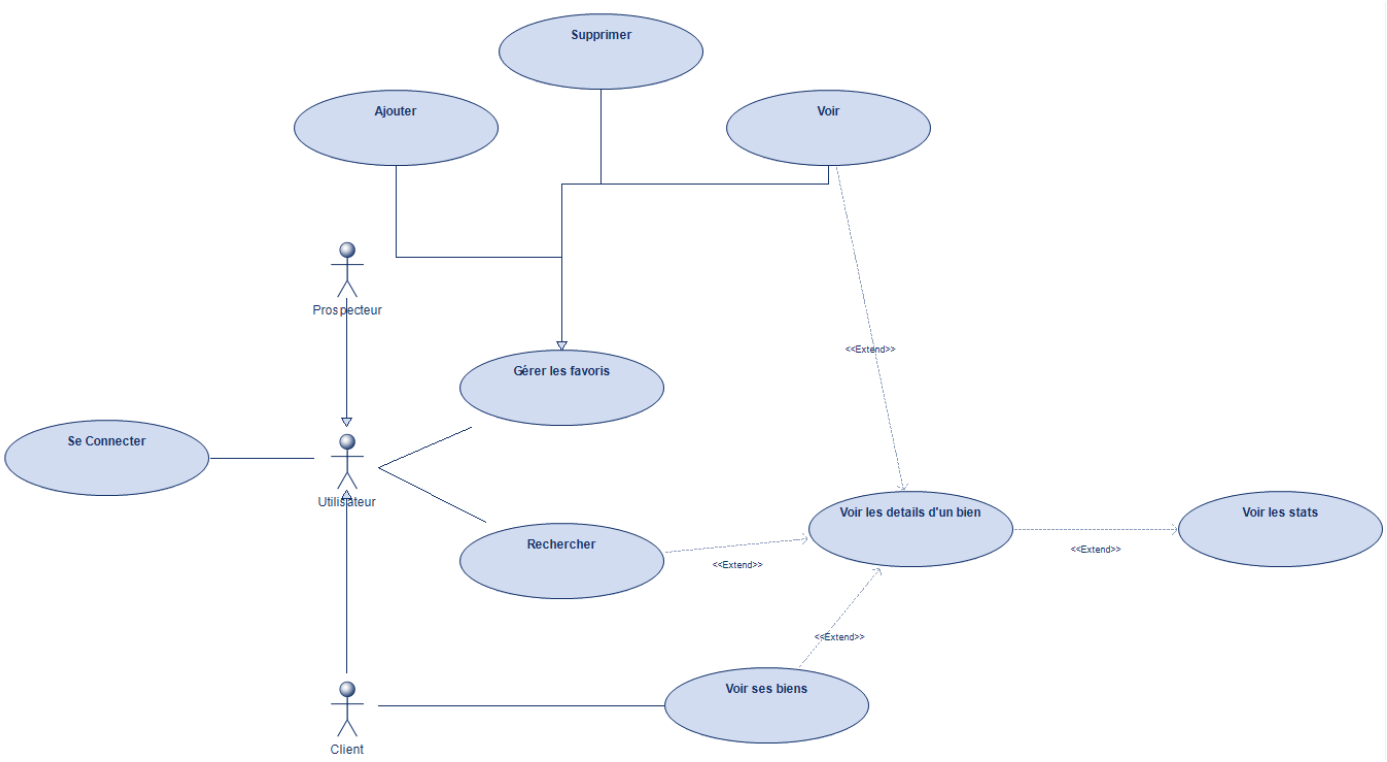
Grimmo Web

GitHub

Table des matières

Diagramme de cas d'utilisation

Voici le diagramme de cas d'utilisation réaliser pour la conception de site web Grimmo Web. Ce diagramme de cas représente les actions que pourra effectuer chaque utilisateur du site web.



Screenshot 1 : Diagramme de cas d'utilisation

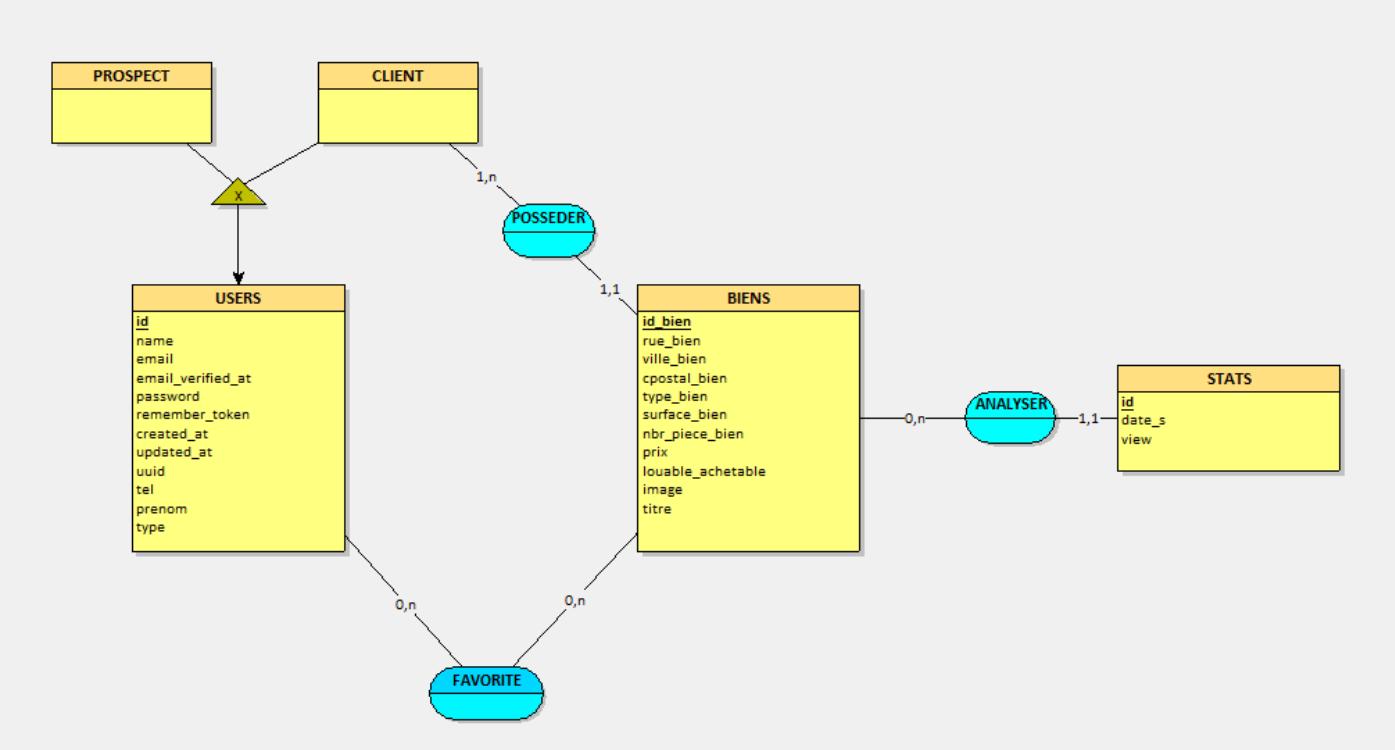
Comme on peut le voir ci-dessus, un prospecteur hérite d'utilisateur (le client qui veut acheter un bien) et client (le client de l'agence qui met en vente ses propriétés) hérite lui aussi d'utilisateur. Dans tous les cas l'utilisateur en général pourra :

- Se connecter
- Ajouter des favoris
- Gérer ses favoris
- Effectuer des recherches

La différence entre un utilisateur client et un utilisateur prospect qui est un client lambda, c'est que l'utilisateur client peut accéder à une page supplémentaire qui s'intitule « Mes biens » et qui permet de visualiser l'ensemble de ses biens mise en ligne par l'agence et de voir les statistiques de celui-ci.

Modèle Conceptuel de Données (MCD)

Pour réaliser ce projet, j'ai dû réaliser un MCD qui m'a permis de créer la structure de la base de données de manière correcte pour répondre aux précisément aux besoins de l'application.



Screenshot 2 : Modèle Conceptuel de Données (MCD)

Comme montrer ci-dessus, la base de données est constituée de 4 entités, dont une CIM (Common Information Model ou en français Modèle de Données Unifié). Sur ce MCD, nous pouvons voir que l'entité « USERS » possède deux enfants héritant d'elle. Cela permet de distinguer un utilisateur lambda et un client comme expliqué précédemment pour le diagramme de cas d'utilisation.

Règles de gestion

USERS

Un utilisateur (client) peut avoir un ou plusieurs biens.

Un utilisateur peut avoir zéro ou plusieurs favoris

BIENS

Un bien à un et un seul propriétaire (client).

Un bien peut être associé à zéro ou plusieurs favoris

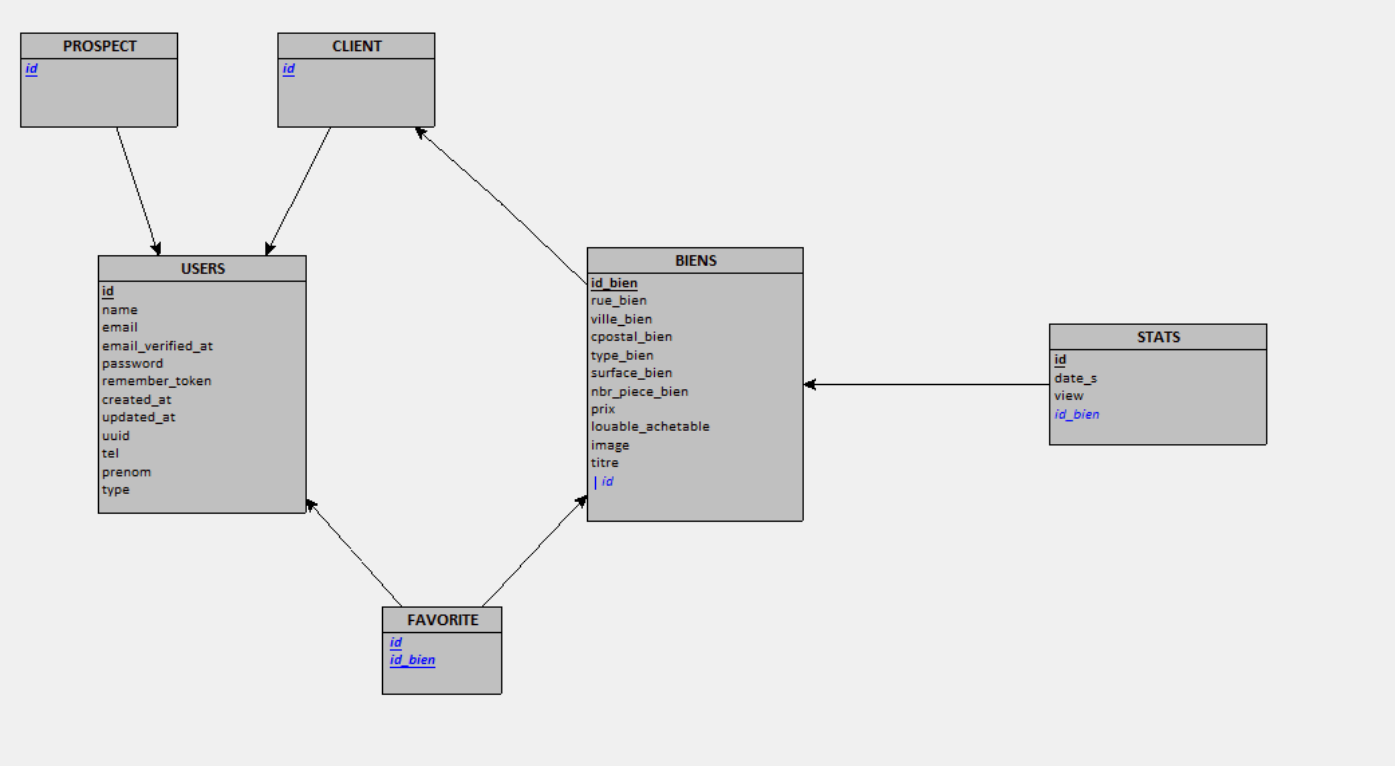
Un bien peut avoir zéro ou plusieurs statistiques

STATS

Une stat peut avoir un et un seul bien.

Modèle Logique de Données (MLD)

Voici le MLD associé au Modèle Conceptuel de Données.



Screenshot 3 : Modèle Logique de Données (MLD)

On peut voir sur ce MLD que l'entité BIENS est reliée à l'entité USERS par la clé étrangère ID de l'entité USERS. L'entité STATS est reliée à l'entité BIENS avec la clé étrangère ID_BIEN de l'entité BIENS. Enfin la CIM FAVORITE est la concaténation de la clé primaire de l'entité USERS et de l'entité BIENS.

Fonctionnalités

API

Toutes les routes api sont surveillées par le middleware Sanctum de Laravel pour assurer la confidentialité et l'intégrité des données présentes dans la base de données.

add_user

Screenshot 4 : Méthode API add_user

Cette méthode va permettre au client lourd d'ajouter un nouvel utilisateur sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /add_user »

add_good

Screenshot 5 : Méthode API add_good

La méthode « add_good » va permettre au client lourd d'ajouter un nouveau bien sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /add_good »

delete_good

Screenshot 6 : Méthode API delete_good

La méthode « delete_good » va permettre au client lourd de supprimer un bien sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /delete_good»

delete_user

```
public function delete_user(Request $request)
```

Screenshot 7 : Méthode API delete_user

La méthode « delete_user » va permettre au client lourd de supprimer un utilisateur sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /delete_good»

edit

```
public function edit(Request $request)
```

Screenshot 8 : Méthode API edit

Cette méthode va permettre au client lourd de modifier un bien ou un utilisateur sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /edit»

get_good_stats

```
public function get_good_stats(Request $request)
```

Screenshot 9 : Méthode API get_good_stats

Enfin cette méthode va permettre au client lourd de récupérer les statistiques d'un bien sur la base de données Laravel en envoyant une requête HTTP POST à la route api « /get_good_stats»

Site web

login

```
public function login(Request $request)
```

Screenshot 10 : Méthode Site web login

La méthode « login » permet à n'importe quel utilisateur de la base de données de se connecter à son compte. Pour la connexion, cette méthode va encrypter avec Bycrypt le mot de passe saisi en brut par l'utilisateur dans le formulaire de connexion et le comparer avec le hash de son mot de passe stocké dans la base de données.

signup

```
public function signup(Request $request)
```

Screenshot 11 : Méthode Site web signup

La méthode « signup » permet à un utilisateur de type prospect de se créer un compte pour consulter les biens mis en ligne par l'agence et les sauvegarder dans ses favoris.

logout

```
public function logout(Request $request)
```

Screenshot 12 : Méthode Site web logout

La méthode "logout" permet de se déconnecter de son compte. Celle-ci ferme la session Laravel en cours.

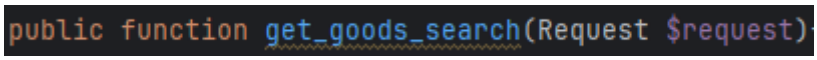
get_my_goods

```
public function get_my_goods()
```

Screenshot 13 : Méthode Site web get_my_goods

Cette méthode permet de récupérer tous les biens de l'utilisateur client lors du chargement de la vue « Mes biens ».

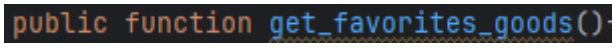
get_goods_search



Screenshot 14 : Méthode Site web get_goods_search

Cette méthode permet de récupérer tous les biens qui correspondent au critère de recherche saisi par l'utilisateur dans la barre de recherche.

get_favorites_goods



Screenshot 15 : Méthode Site web get_favorites_goods

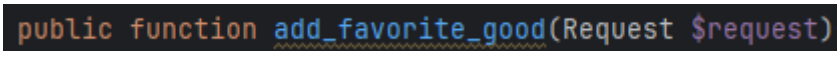
La méthode « get_favorites_goods » permet de récupérer tous les biens enregistrés en favoris de l'utilisateur connecté.

check_favorite

Screenshot 16 : Méthode Site web check_favorite

Cette méthode est appelée lors du chargement de la vue qui affiche les détails d'un bien et permet de vérifier si le bien a déjà été ajouté en favoris par l'utilisateur. Si c'est le cas, la page affiche le cœur en rouge sinon vide avec les contours rouges pour montrer l'absence dans les favoris.

add_favorite_good



Screenshot 17 : Méthode Site web add_favorite_good

Cette méthode permet simplement d'ajouter un bien dans les favoris de l'utilisateur.

remove_favorite_good

```
public function remove_favorite_good(Request $request)
```

Screenshot 18 : Méthode Site web remove_favorite_good

Cette méthode permet de supprimer un bien mis en favoris par l'utilisateur.

Table des illustrations