Grimmo Web

Repo GitHub :

Table des matières

[Diagramme de cas d’utilisation 3](#_Toc185353368)

[Modèle Conceptuel de Données (MCD) 4](#_Toc185353369)

[Règles de gestion 4](#_Toc185353370)

[USERS 4](#_Toc185353371)

[BIENS 5](#_Toc185353372)

[STATS 5](#_Toc185353373)

[Modèle Logique de Données (MLD) 5](#_Toc185353374)

[Fonctionnalités 6](#_Toc185353375)

[API 6](#_Toc185353376)

[add\_user 6](#_Toc185353377)

[add\_good 6](#_Toc185353378)

[delete\_good 6](#_Toc185353379)

[delete\_user 7](#_Toc185353380)

[edit 7](#_Toc185353381)

[get\_good\_stats 7](#_Toc185353382)

[Site web 8](#_Toc185353383)

[login 8](#_Toc185353384)

[signup 8](#_Toc185353385)

[logout 8](#_Toc185353386)

[get\_my\_goods 8](#_Toc185353387)

[get\_goods\_search 9](#_Toc185353388)

[get\_favorites\_goods 9](#_Toc185353389)

[check\_favorite 9](#_Toc185353390)

[add\_favorite\_good 9](#_Toc185353391)

[remove\_favorite\_good 10](#_Toc185353392)

[Table des illustrations 11](#_Toc185353393)

# Diagramme de cas d’utilisation

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

Une image contenant cercle, conception, illustration

Description générée automatiquement

Screenshot 1 : Diagramme de cas d'utilisation

Comme on peut le voir ci-dessus, un utilisateur hérite de prospecteur (le client qui veut acheter un bien) ou client (le client de l’agence qui met en vente ses propriétés). 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 pouvoir 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.

Une image contenant texte, diagramme, capture d’écran, Plan

Description générée automatiquement

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 associer 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 Conceptuelle de Données.

Une image contenant texte, capture d’écran, diagramme, Plan

Description générée automatiquement

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 le 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



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



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



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



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, ce 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 sur la base de données.

### signup



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 mit en ligne par l’agence et les sauvegarder dans ses favoris.

### logout



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



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 d’un détail d’un bien et permet de vérifier si le bien à 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 rouge 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



Screenshot 18 : Méthode Site web remove\_favorite\_good

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

# Table des illustrations

[Screenshot 1 : Diagramme de cas d'utilisation 3](#_Toc185353269)

[Screenshot 2 : Modèle Conceptuel de Données (MCD) 4](#_Toc185353270)

[Screenshot 3 : Modèle Logique de Données (MLD) 5](#_Toc185353271)

[Screenshot 4 : Méthode API add\_user 6](#_Toc185353272)

[Screenshot 5 : Méthode API add\_good 6](#_Toc185353273)

[Screenshot 6 : Méthode API delete\_good 6](#_Toc185353274)

[Screenshot 7 : Méthode API delete\_user 7](#_Toc185353275)

[Screenshot 8 : Méthode API edit 7](#_Toc185353276)

[Screenshot 9 : Méthode API get\_good\_stats 7](#_Toc185353277)

[Screenshot 10 : Méthode Site web login 8](#_Toc185353278)

[Screenshot 11 : Méthode Site web signup 8](#_Toc185353279)

[Screenshot 12 : Méthode Site web logout 8](#_Toc185353280)

[Screenshot 13 : Méthode Site web get\_my\_goods 8](#_Toc185353281)

[Screenshot 14 : Méthode Site web get\_goods\_search 9](#_Toc185353282)

[Screenshot 15 : Méthode Site web get\_favorites\_goods 9](#_Toc185353283)

[Screenshot 16 : Méthode Site web check\_favorite 9](#_Toc185353284)

[Screenshot 17 : Méthode Site web add\_favorite\_good 9](#_Toc185353285)

[Screenshot 18 : Méthode Site web remove\_favorite\_good 10](#_Toc185353286)