|  |
| --- |
| Hehocom – Projet Pro |
| App Hehocom |
| Dossier technique |

|  |
| --- |
| Guillaume Durupt  17/01/2020 |

***Objet :***

*L’objectif de ce document est de décrire le cahier des charges du projet AppHehocom qui sera réalisé par Durupt Guillaume. Ce cahier des charges a pour vocation de démontrer la bonne compréhension du sujet, d’apporter une analyse sur le travail à effectuer et de proposer des solutions organisationnelles et techniques au sujet.*

Table des matières

[1. Cahier des charges 3](#_Toc61934512)

[1.1. Présentation de l’ensemble du projet 3](#_Toc61934513)

[1.1.1. Présentation du client 3](#_Toc61934514)

[1.1.2. Présentation du projet 3](#_Toc61934515)

[1.1.3. La cible visée par le projet 3](#_Toc61934516)

[1.1.4. Périmètre du projet 4](#_Toc61934517)

[1.2. Délais 4](#_Toc61934518)

[1.3. Analyse technique 4](#_Toc61934519)

[1.3.1. Description graphique et ergonomique 4](#_Toc61934520)

[1.3.1.1. Charte graphique 4](#_Toc61934521)

[1.3.2. Etude de besoins 4](#_Toc61934522)

[1.3.2.1. Besoins fonctionnels 4](#_Toc61934523)

[1.3.2.2. Besoins non fonctionnels 5](#_Toc61934524)

[1.3.3. Description fonctionnelle et technique 6](#_Toc61934525)

[1.3.3.1. Arborescence du site 6](#_Toc61934526)

[1.3.3.2. Ebauche d'une maquette 9](#_Toc61934527)

[1.3.4. Contraintes techniques 10](#_Toc61934528)

[1.3.4.1. Environnement de développement et Technologies utilisées 10](#_Toc61934529)

[1.3.4.2. Compatibilité des supports 10](#_Toc61934530)

[1.3.4.3. Intégration de l'emailing 10](#_Toc61934531)

1.3.4.4. Plan de sécurisation 11

[2. Modélisation de la structure de la base de données 12](#_Toc61934532)

[2.1. Modélisation de la base de données 12](#_Toc61934533)

[2.2. Architecture globale de l’application 13](#_Toc61934534)

# Cahier des charges

## Présentation de l’ensemble du projet

### Présentation du client

Hehocom est une agence qui a pour but de créer des sites web vitrine et e-commerce. Elle maintient à jour les sites web de ses clients, et permet même de faire un suivi des sites web afin de faire des modifications sur le long terme.

### Présentation du projet

Ayant pour objectif de permettre à leur client un suivi de la création de leurs sites web et de leur maintenance, l’agence Hehocom propose une application web qui permettra aux clients de suivre les actions réalisées sur leur site web, les « missions » qu’il reste à effectuer et leur progression en temps réel, de partager des commentaires/ressources avec l’équipe technique et d’avoir un Dashboard de statistiques.

Cette application permettra surtout de centraliser les demandes des clients, cela permettra un suivi plus facile et directement en liaison, de l’équipe technique au client et vice versa.

### La cible visée par le projet

Les cibles principales du projet sont les clients de l’agence Hehocom, tout client qui veux avoir une vue sur la création ou le maintien de son site pourra suivre cela sur l’application.

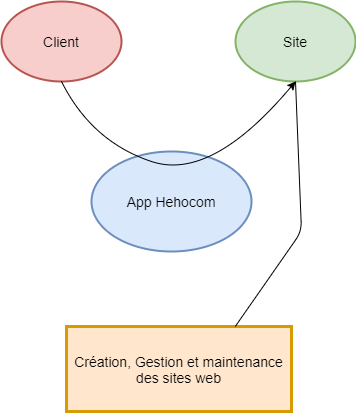


Diagramme d’analyse du besoin

### Périmètre du projet

L’application respectera les critères suivants :

* La langue principale devra être la langue Française.
* Le projet devra être accessible via Internet.

De manière à garantir une accessibilité maximale, les fonctions suivantes seront intégrées :

* Le projet sera pensé afin de faciliter l’ergonomie, c’est-à-dire lisibilité et simplicité des fonctionnalités.

## Délais

Etant un projet professionnel réalisé pour l’agence Hehocom en liaison avec l’école CESI, le projet devra être opérationnel pour début Juillet.

## Analyse technique

### Description graphique et ergonomique

#### Charte graphique

L’application utilisera les visuels créés pour le site web Hehocom.

Bientôt disponible

Logo prévisionnel

Le site devra allier confort visuel, simplicité d’accès aux fonctionnalités et originalité, de manière à dynamiser les contenus sans surcharger l’interface.

### Etude de besoins

#### Besoins fonctionnels

Fonctionnalités principales à assurer :

* Interface des sites web pour les clients (comportant Statistiques, missions et commentaires)
* Gestion des utilisateurs
* Gestion des missions
* Gestion des commentaires
* Option « mot de passe oublié » avec possibilité de le retrouver
* Création de missions PAR site web avec différents statuts pour le développeur (Urgent, A faire, Fait)
* Les sites web seront aussi affichés sous différents statuts sur l’interface client (En ligne, en panne, en cours de création)

Les priorités secondaires :

Un onglet privé pour l’équipe technique permettant de voir l’architecture et l’organisation des sites web en général :

* Sur quel serveur sont les données du site
* Où est la base de données ?

#### Besoins non fonctionnels

Non négligeables, les besoins non fonctionnels agissent de façon indirecte sur le résultat et sur le rendement de l’utilisateur. Il faut répondre aux exigences suivantes :

##### Fiabilité

L’application doit fonctionner de façon rapide, fluide, permanente et cohérente.

##### Les erreurs

Les ambigüités doivent être signalées par des messages d’erreurs explicites pour bien guider l’utilisateur et le familiariser avec notre site web sans le décourager.

##### Ergonomie et Interface intuitive

L’application doit être adaptée à l’utilisateur sans qu’il ne fournisse aucun effort (utilisation claire et facile) du point de vue navigation entre les différentes pages, couleurs, polices et mises en page utilisées.

##### Sécurité

Notre solution doit respecter la norme RGPD (confidentialité des données personnelles des utilisateurs). Un système de mot de passe et de cryptage des informations personnelles devra donc être mis en place.

##### Aptitude à la maintenance et la réutilisation

Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

### Description fonctionnelle et technique

#### Arborescence du site

Le site possèdera deux facettes un “Front office” et un “Back office” :

Le “Front Office” sera pour les personnes qui n’ont pas de compte utilisateur (Page login), ce qui servira à sécuriser l’application au travers d’un email et d’un mot de passe.

Le “Back office” sera pour les utilisatrices et utilisateurs connectés. Il comportera un menu qui permettra, selon le statut lié au compte utilisateur, de pouvoir utiliser ou non les différentes options de l’application.

Toutes les personnes qui ne seront pas connectées via la page login n’auront pas accès aux autres pages.

##### Fonctionnalités Client

*Les termes en bleu correspondent à des fonctionnalités prioritaires à implémenter. Les termes en vert correspondent à des fonctionnalités secondaires.*

Le client aura la possibilité de naviguer sur l’application, il pourra avoir accès aux différentes options selon ses sites :

La page de consultation principale comportera les fonctionnalités suivantes :

* Un Dashboard avec la liste du(des) site(s) accessible(s) à celui-ci :
  + Nom du site
  + URL du site
  + Type de site (Vitrine e-commerce ou app)
  + Statut du site
  + Nombre de commentaires
  + Nombre de missions
* Un clic sur le nom de la catégorie de tri sélectionnée donnera accès à l’intégralité des données :
  + Liste des commentaires avec possibilités de modifier les commentaires rédigés précédemment
  + Liste des missions avec leurs statuts d’avancement
  + Possibilité d’importer une ressource extérieure (PDF, image, docs, etc.)
  + Un formulaire de recherches pour rechercher les différents sites
  + Un aspect gestion de données sera disponible (ventes, visiteurs, pages les plus visités… à définir en fonction du statut du site)

##### Fonctionnalités équipe technique

* Gestion des utilisateurs (Création, modification et suppression).
* Gestion des missions seuls l’équipe technique peu gérer les missions (Création, modification et suppression).
* La possibilité de créer/modifier/supprimer les commentaires
* Accès à tous les sites web et de pouvoir modifier leur contenus (stats, ressources, mission, commentaire, statut du site, etc.)
* La possibilité d’ajouter des utilisateurs sur les différents sites.

##### Récapitulatif

|  |  |  |  |
| --- | --- | --- | --- |
| ***Type d’utilisateur*** | ***Fonctionnalité*** | ***Niveau de fonctionnalité*** | ***Page*** |
| *Client* | *Visualiser le(s) site(s) disponible(s) pour lui* | *Primaire* | *Page Dashboard* |
| *Client* | *Liste des sites avec descriptions de ceux-ci* | *Primaire* | *Page Dashboard* |
| *Client* | *Description complete d’un site web* | *Primaire* | *Page Site* |
| *Client* | *Listes des derniers commentaires* | *Primaire* | *Page Site* |
| *Client* | *Listes des dernières missions* | *Primaire* | *Page Site* |
| *Client* | *Listes des ressources liées au site* | *Secondaire* | *Page Site* |
| *Client* | *Statistiques du site* | *Primaire* | *PageSite* |
| *Equipe technique* | *Créer un compte* | *Primaire* | *Page Utilisateur* |
| *Equipe technique* | *Gestion des missions* | *Primaire* | *Page Site* |
| *Equipe technique* | *Gestion des utilisateurs* | *Primaire* | *Page Utilisateur* |
| *Equipe technique* | *Gestion des sites* | *Primaire* | *Page Site* |
| *Equipe technique* | *Gestion des commentaires* | *Primaire* | *Page Site* |
| *Equipe technique* | *Gestion des ressources* | *Primaire* | *Page Site* |
| *Equipe technique* | *Gestion des Statistiques* | *Primaire* | *Page Site* |
| *Equipe technique* | *Gestion du statut du site* | *Primaire* | *Page Site* |
| *Equipe technique* | *Architecture des sites web / base de données* | *Secondaire* | *Page Architecture* |

Liste des fonctionnalités prévues

L’ensemble de ces fonctionnalités peut être schématisé par un diagramme de cas d’usage UML, tel que présenté ci-dessous. Ce diagramme permet de représenter l’ensemble des fonctionnalités accessibles pour chaque acteur de l’application, et les relations de dépendance entre chacun d’entre eux.

La notion d’héritage, décrivant la hiérarchisation des types d’utilisateurs, est mise en évidence dans ce schéma par l’utilisation de flèches rouges reliant les différents utilisateurs. Elles modélisent le fait que le membre de l’équipe technique puisse accéder à toutes les fonctionnalités de l’utilisateur connecté.

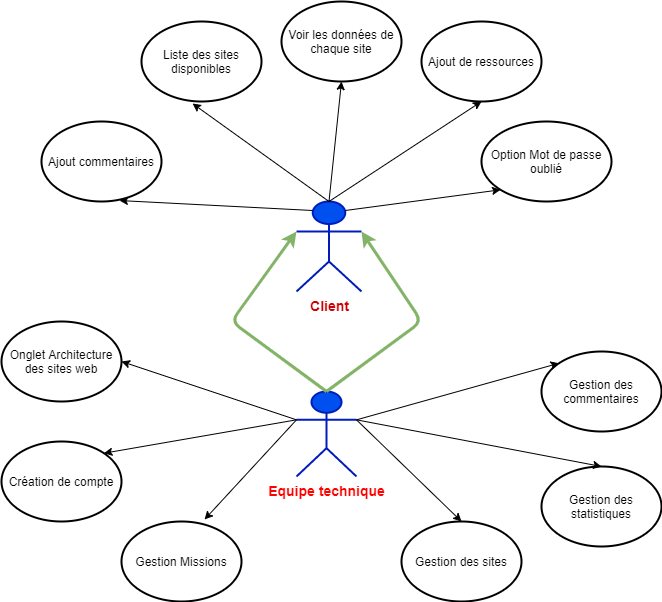
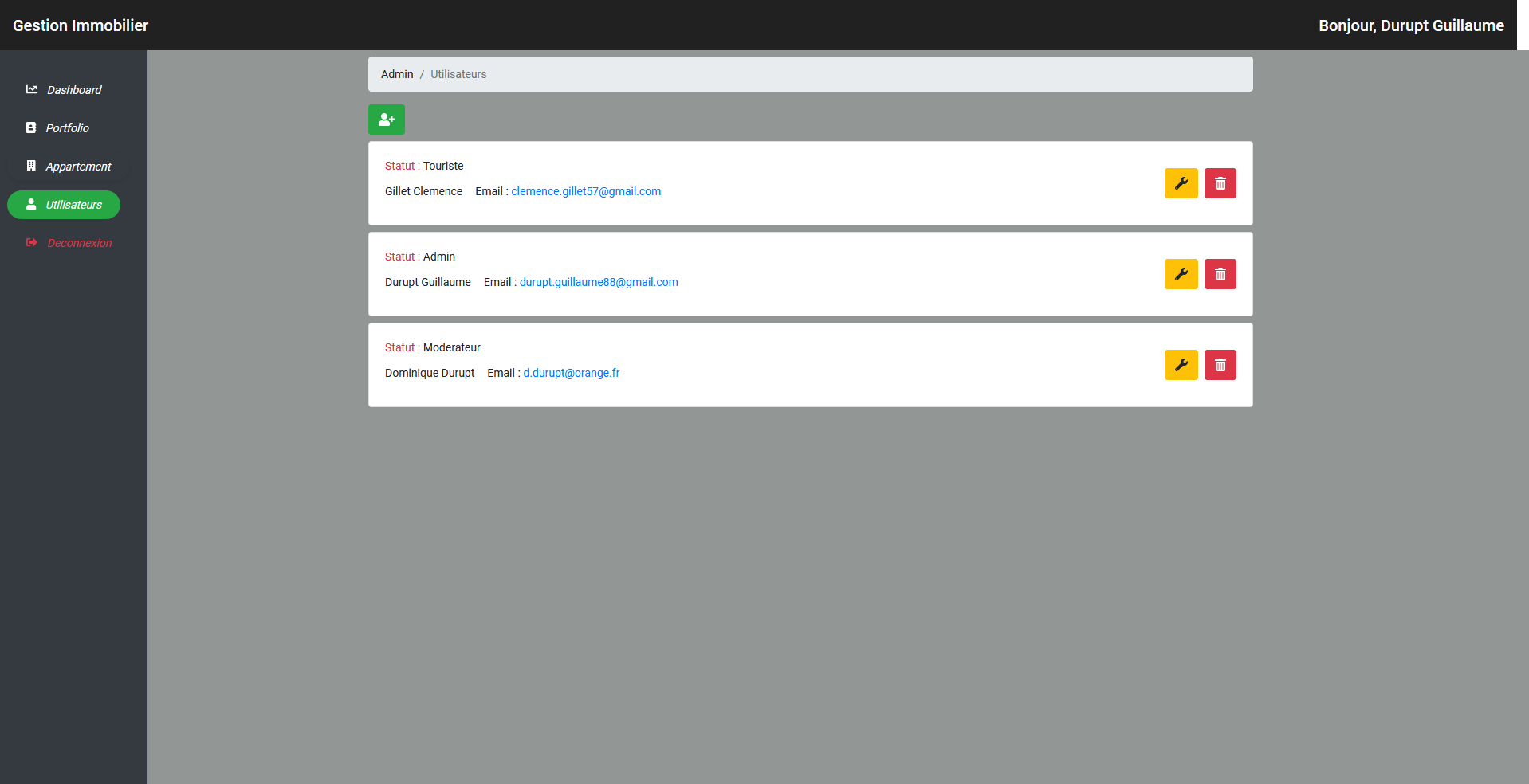
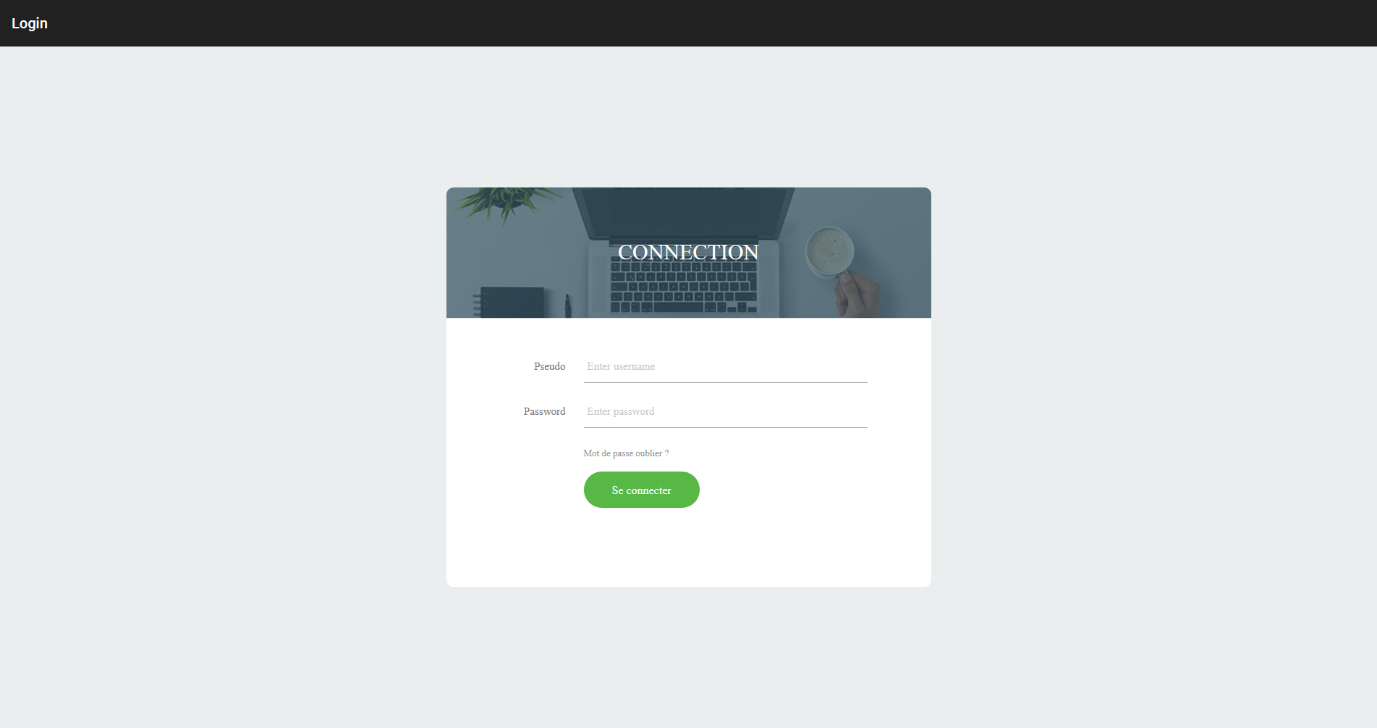


Diagramme de cas d’utilisation

#### Ebauche d'une maquette





### Contraintes techniques

#### Environnement de développement et Technologies utilisées

* + - Editeur : Visual Studio code / Eclipse
    - Langages : Html/CSS, Javascript Type script JEE
    - Base de données : MySQL
    - Framework: Angular / spring boot

#### Compatibilité des supports

Le projet devra être disponible sur tous les navigateurs les plus courants (au minimum Chrome, Firefox, Internet Explorer > 11, Safari) et être fonctionnel et opérationnel sur chacun d’eux.

Il serait bon d'avoir une compatibilité ascendante pour tous les navigateurs requis et d'étendre le support aux navigateurs tels que Microsoft Edge, Opera, Tor, etc.

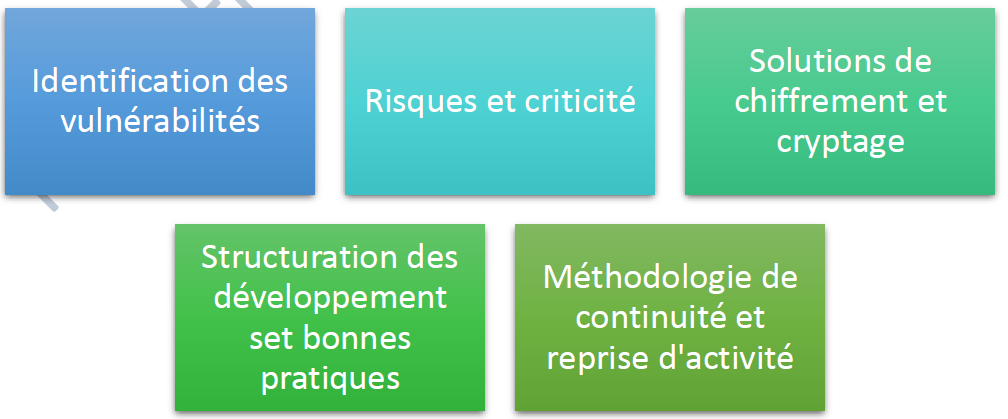
Du côté de l’application mobile, celle-ci sera compatible Android et iOS.

Normes de codage, normes techniques cohérentes et conformes aux meilleures pratiques seront pour la continuité et la maintenance du projet

#### Intégration de l'emailing

Le demandeur recevra après son inscription un mail à l’adresse rentrée afin de pouvoir se connecter à son compte. Cela permet une sécurité en plus pour l’utilisateur et le projet puisqu’il pourra aussi demander un mail indiquant la marche à suivre en cas d’oubli de mot de passe afin de pouvoir le changer.

* + - 1. *Plan de sécurisation*



Eléments de stratégie de sécurisation

1. *Identification des vulnérabilités*

Voici les vulnérabilités les plus courantes :

* Injection SQL
* Attaques XSS (Cross-site Scripting)

La planification de tests d'intrusions seront mises en place afin de détecter les vulnérabilités avant les pirates.

1. *Risques et criticité*

Les vulnérabilités seront évaluées et triées en fonction de son risque et de sa menace pour qu’elles soient traitées et sécurisées.

1. *Solutions de chiffrement et cryptage*

Dans la base de données, chaque mot de passe et données personnelles seront cryptés via une clé de chiffrement.

1. *Structuration des développements et bonnes pratiques*

Le développement de l’application sera structuré par un planning.

Une partie du temps de travail sera dédié à la sécurité.

Les bonnes pratiques de développements web seront suivis.

1. *Méthodologie de continuité et reprise d'activité*

Une documentation technique ainsi que l’historique des vulnérabilités sera mise à jour afin d’assurer la continuité et le suivi de l’application.

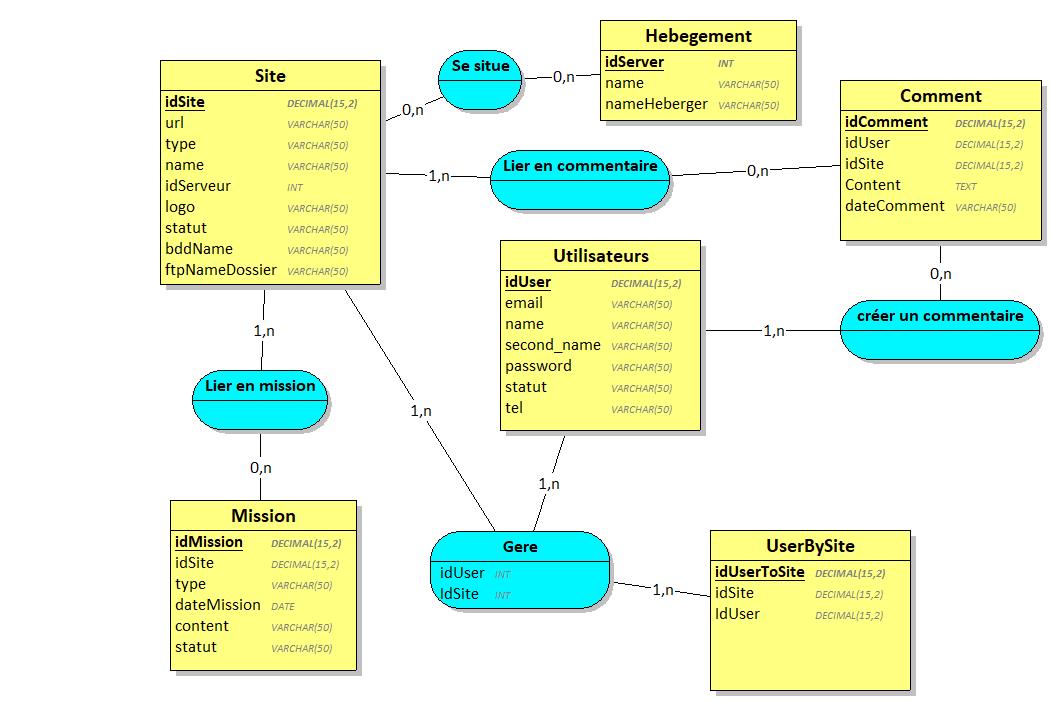
# Modélisation de la structure de la base de données

## Modélisation de la base de données

La base de données est de type relationnel, fondée sur la technologie MySQL. La base s’articule autour de deux tables principales, représentant les utilisateurs et les sites.

Plusieurs tables d’associations permettent de détailler les relations suivantes :

* Les sites liés à chaque utilisateur.
* Les commentaires liés à chaque utilisateur et site.
* Les missions liées à chaque utilisateur et site.
* Les ressources liées à chaque utilisateur et site.



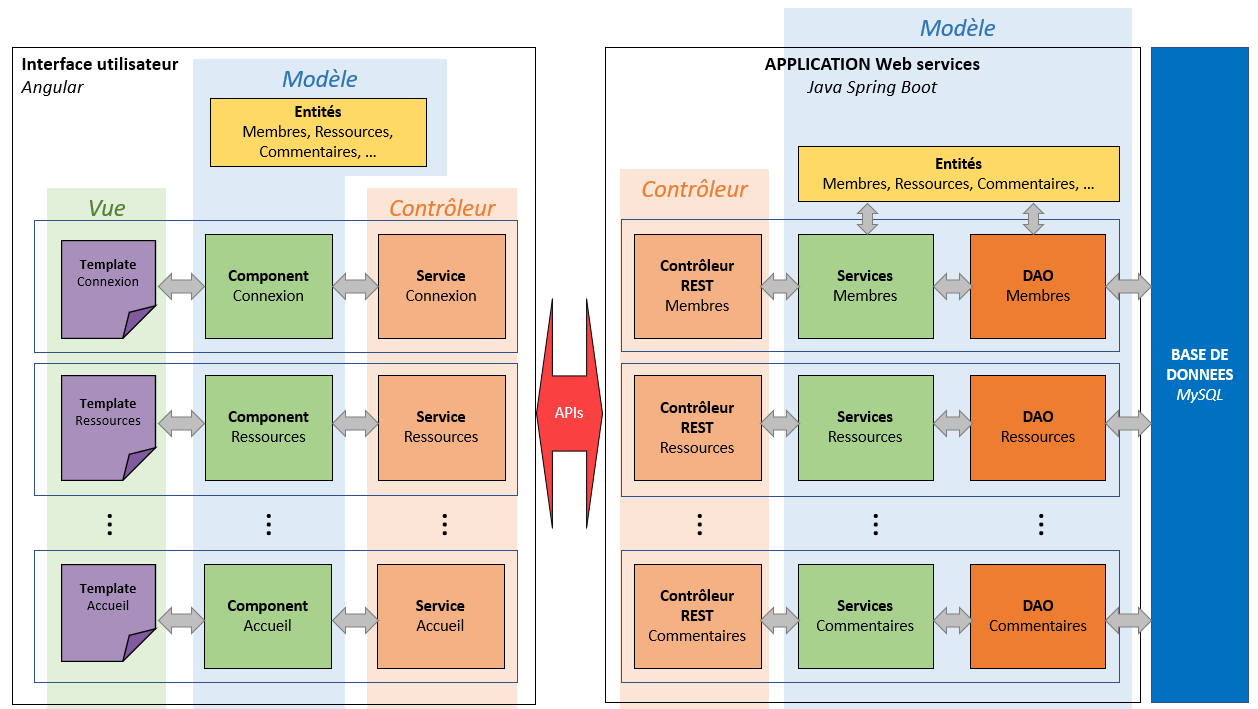
MODELISATION BASE DE DONNEE APP HEHOCOM

## Architecture globale de l’application

L’application est un ensemble de web services respectant la structuration MVC (Modèle Vue Contrôleur).

Côté interface utilisateur : l’utilisateur se connecte sur son navigateur via une interface utilisateur architecturée grâce à Angular. Chaque page du site est construite selon un assemblage de Template contrôlés par des composants. Les données à récupérer de la base de données sont appelées par les services, qui s’occupent de générer les requêtes http et de récupérer les réponses, grâce à des API (Application Programming Interface) au format json.

Côté web services, ces requêtes sont convoyées via les contrôleurs. Ceux-ci transmettent les données aux services (et les récupèrent). Chaque service va ensuite faire les requêtes nécessaires auprès de la base de données, en passant par des objets intermédiaires, les DAO (Data Access Object). Ces objets communiquent directement avec la base de données grâce à des méthodes qui sont autant de requêtes auprès de la base. Une fois les données récupérées, elles sont stockées dans des objets entités et éventuellement transformées et traitées dans les services, avant de repartir vers les contrôleurs puis vers l’interface utilisateur.



Architecture complète de l’application web (Re)sources Relationnelles

Chaque trio contrôleur / service / dao peut ainsi être considéré comme un microservice, regroupé au sein d’une API.

FIN DU DOCUMENT