

A l'arrivée sur le site on tombe sur le module information, sorte de page d'accueil avec infos généralistes  
Attendus du module :

- Un « free tour » sur la plateforme ;
  - Une recherche d'information contenant au moins deux (2) filtres ;
  - La possibilité de pouvoir s'inscrire sur la plateforme.
- (affichage météo, nombre d'objet, température du moment)

-----

**Inscription** (pop up ou nouvelle page?) :

Pseudonyme (login)

o Age

o Sexe / Genre

o Date de naissance

o Type de membre (mère, père, enfant...)

o Une photo

o Nom

o Prénom

Adresse mail

o Mot de passe

Une fois cliqué sur envoyé, un admin doit valider la demande, un mail est automatiquement envoyé pour confirmer l'adresse mail (gestion token...)

-----

Une fois inscrit, l'utilisateur peut se connecter (**login/mdp**) et a accès au module visualisation  
Il se voit automatiquement attribuer le niveau simple avec nb de point a definir

**Visualisation** (débutant et intermédiaire)

Il a acces a son profil (modif photo et mot de passe) mais aussi a la partie publique des autres profils

o Pseudonyme (login)

o Age

o Sexe / Genre

o Date de naissance

o Type de membre (mère, père, enfant etc.)

o photo

Consultation de tous les objets connectés (conso etc...) ET outils/service variés à définir  
mise en place moteur de recherche par mot clé (nom ou alors température pour le thermostat etc...)

Affiche liste de résultats, et au clic ouvre popup ou page (mise a jour nombre consultation, date etc...)  
Plein de détails sur chaque capteurs

---

**Exemple d'objet connecté** : Thermostat intelligent

ID unique : Thermo123

Nom : Thermostat Salon

Température actuelle : 21°C

Température cible : 23°C

Mode : Automatique

Connectivité : Wi-Fi, signal fort

État de batterie : 65%

---

Dernière interaction : Aujourd'hui, 10:00 AM

----

Définition de la méthode de calcul pour les points pour passer les niveaux

type simple avec les niveaux débutant ou intermédiaire. Si l'utilisateur passe au niveau avancé, il débloquent le module « Gestion », nous parlons d'un utilisateur du type complexe. S'il passe au niveau expert, il débloquent le module « Administration », nous parlons d'un utilisateur type administrateur.

Différence fonction entre débutant et intermédiaire pour le module visualisation ?

Possibilité de changer niveau utilisateur (si droit) : menu déroulant avec que les niveaux auquel il a accès ie une page par module?

Gérer le nombre d'accès des utilisateurs à la plateforme ;

•Gérer le nombre d'actions faites par les utilisateurs ;

•Mettre à jour les données précédentes (accès et actions) pour chaque utilisateur et chaque accès à la plateforme.

???

### 3.Module « Gestion » pour complexe mini

gestion avancée des objets : limite de température pour chauffage et clim, changement code alarme, etc...  
stats utilisations et génération rapportsAttendus du module :

#### Gestion des objets connectés

o Ajouter un nouvel objet connecté sur la plateforme ;

o Solliciter (à l'administrateur) la suppression d'un objet connecté ;

o Modifier les informations et attributs d'un objet connecté (nom, description, statut, paramètres, etc.) ;

o Contrôler l'état d'un objet (activer/désactiver, mettre à jour).

#### Configuration des services

o Associer des objets connectés à des pièces ou zones spécifiques ;

o Configurer les paramètres d'utilisation des objets connectés (par ex. Température cible pour un thermostat, horaire de fonctionnement pour une lumière).

#### Surveillance et optimisation des ressources

o Consulter et générer des rapports d'utilisation des objets (par ex. Consommation énergétique quotidienne ou hebdomadaire) ;

o Identifier les objets inefficaces (selon les paramètres d'utilisation) ou nécessitant une maintenance.

#### Rapports statistiques

o Générer des statistiques sur l'utilisation des objets et des services pour optimiser la gestion des ressources ;

o Générer des rapports sur l'utilisation des services et objets connectés ;

o Accéder aux historiques des données des objets connectés.

### 4.Module « Administration » (admin niveau expert)

Le profil contient également une partie privée, accessible seulement à l'utilisateur lui-même et à l'administrateur, qui contient les informations suivantes :

o Nom

o Prénom

o Mot de passe (s'il veut le modifier)

Attendus du module :

- 

Gestion des utilisateurs

- o Ajouter, modifier ou supprimer des utilisateurs ;
- o Attribuer ou révoquer des niveaux d'accès aux utilisateurs ;
- o Superviser les points accumulés par les utilisateurs et ajuster manuellement les niveaux en cas de besoin ;
- o Consulter les historiques de connexion et d'actions des utilisateurs.

- 

Gestion des objets et des outils/services

- o Ajouter ou supprimer des catégories d'objets et d'outils/services ;
- o Ajouter ou supprimer des objets et des outils/services ;
- o Définir les règles de fonctionnement globales des objets et des outils/services (par exemple, priorités énergétiques, gestion des alertes, etc.).

- 

Sécurité et maintenance

- o Mettre à jour le système de gestion des accès (par exemple, mise à jour des mots de passe administrateurs) ;
- o Effectuer des sauvegardes régulières de la base de données ;
- o Vérifier l'intégrité des données de la plateforme.

- 

Personnalisation de la plateforme

- o Modifier l'apparence et la structure des modules pour répondre aux besoins des utilisateurs ;
- o Configurer des règles de validation pour les inscriptions (par exemple, approbation manuelle).

- 

Rapports Avancés

- o Générer des rapports détaillés sur l'utilisation globale de la plateforme, avec exportation des données (CSV, PDF, etc.) ;
- o Statistiques sur la consommation énergétique totale, le taux de connexion des utilisateurs, et les services les plus utilisés.

strucure

page accueil

page inscription

page admin

page gestion

page visualisation

divers

Données générées pour chaque ojbets (trouver générateur en ligne)

Rapport des objets remplis automatiquement par les objets?

Journal utilisateur avec tout ce qui a été fait, connexion, consultation, modif avec le matériel lié

# Détail des Modules pour une Maison Connectée

---

## 1. Module « Information»

**Public :** Visiteurs (accès libre sans connexion).

**Fonctionnalités clés :**

- **Page d'accueil :**
  - Présentation visuelle de la maison connectée (bannières, vidéo de démonstration).
  - Section "Free Tour" avec aperçu des objets connectés (ex : thermostat, caméras).
- **Recherche d'informations :**
  - Moteur de recherche avec filtres :
    - Par **type d'objet** (ex : éclairage, sécurité).
    - Par **pièce** (salon, cuisine).
  - Résultats cliquables avec descriptions basiques (ex : "Thermostat Salon – 21°C").
- **Inscription :**
  - Formulaire avec : email, mot de passe.
  - Envoi d'un email de validation avec lien de confirmation.

**Exemple concret :**

Un visiteur recherche "éclairage cuisine" → affiche les ampoules connectées disponibles avec leurs statuts (allumé/éteint).

---

## 2. Module « Visualisation»

**Public :** Utilisateurs simples (membres vérifiés).

**Fonctionnalités clés :**

- **Authentification :**
  - Connexion via email/mot de passe + captcha.
  - Page de réinitialisation de mot de passe.
- **Gestion de profil :**
  - **Partie publique :**
    - Photo, pseudo, rôle (ex : "Parent", "Enfant").
    - Bouton "Voir profil" pour les autres membres (sans modification possible).
  - **Partie privée :**
    - Édition du nom/prénom/mot de passe.
    - Historique des actions (ex : "Consulté Thermostat le 10/05").
- **Recherche avancée :**
  - Filtres :

- **Statut** (connecté/déconnecté).
- **Marque** (ex : Philips Hue, Nest).
- Affichage des détails de l'objet (ex : batterie, dernière activité).
- **Système de niveaux** :
  - Points attribués pour :
    - Connexion quotidienne (+0.25 pts).
    - Consultation d'un objet (+0.5 pts).
  - Barrière à franchir :
    - **Débutant** → **Intermédiaire** : 10 pts (ex : 10 consultations).
    - **Intermédiaire** → **Avancé** : 30 pts (débloque le module Gestion).

#### **Exemple concret :**

Un utilisateur consulte 5 objets en une semaine → 2.5 pts. Après 4 semaines, il atteint 10 pts et passe "Intermédiaire".

---

### **3. Module « Gestion »**

**Public** : Utilisateurs complexes (niveau avancé/expert).

**Fonctionnalités clés** :

- **Gestion des objets** :
  - **Ajout manuel** :
    - Formulaire avec champs obligatoires (nom, type, pièce, IP).
    - Scan de QR code pour les objets compatibles.
  - **Modification** :
    - Changement de paramètres (ex : température max du thermostat).
    - Marquer un objet comme "Hors service".
- **Configuration** :
  - **Scénarios** :
    - "Mode Nuit" : éteindre les lumières à 23h.
    - "Départ" : activer les caméras et baisser le chauffage.
  - **Alertes** :
    - Notification si un objet est déconnecté > 24h.
- **Rapports** :
  - **Consommation énergétique** :
    - Graphique hebdomadaire des kWh par appareil.
    - Comparaison avec la moyenne du mois précédent.
  - **Export PDF/CSV** pour partage avec un technicien.

### Exemple concret :

L'utilisateur crée un scénario "Vacances" :

- Thermostat à 16°C.
  - Caméras activées.
  - Lumières aléatoires allumées le soir.
- 

## 4. Module « Administration »

**Public :** Administrateurs (niveau expert).

**Fonctionnalités clés :**

- **Gestion des utilisateurs :**
  - Tableau de bord avec liste des membres.
  - Droit de :
    - Supprimer un compte (ex : enfant quittant le foyer).
    - Forcer un niveau (ex : passer un utilisateur en "Expert").
- **Supervision globale :**
  - **Objets :**
    - Créer des catégories (ex : "Sécurité", "Divertissement").
    - Désactiver un objet défectueux pour tous les utilisateurs.
  - **Règles :**
    - Priorité énergétique (ex : éteindre le chauffage si fenêtre ouverte).
- **Maintenance :**
  - Sauvegarde automatique quotidienne de la BDD.
  - Journal des connexions suspectes (ex : 5 tentatives échouées).
- **Personnalisation :**
  - Changer le thème couleur de la plateforme.
  - Ajouter un message d'accueil personnalisé.

### Exemple concret :

L'administrateur reçoit une alerte "Surconsommation du lave-vaisselle" → vérifie l'historique et contacte le fabricant.

## Détail des Modules pour une Maison Connectée avec AngularJS et PHP

---

### 1. Module « Information »

**Public :** Visiteurs (accès libre sans connexion).

**Technologies :**

- **Frontend (AngularJS) :**

- Contrôleur : `InfoController` pour gérer la recherche et l'inscription.
- Service : `InfoService` pour les appels API vers le backend.
- Template :
  - Page d'accueil avec `ng-repeat` pour afficher les objets connectés.
  - Formulaire de recherche avec `ng-model` pour les filtres.
- **Backend (PHP) :**
  - Endpoint : `api/info.php` (méthode GET pour la recherche, POST pour l'inscription).
  - Validation : Vérification des données avant insertion en BDD (ex : email unique).

#### Exemple de code :

```
// AngularJS Controller
app.controller('InfoController', function($scope, InfoService) {
  $scope.search = { type: "", room: "" };
  $scope.results = [];
  $scope.searchObjects = function() {
    InfoService.getObjects($scope.search).then(function(response) {
      $scope.results = response.data;
    });
  };
});
```

```
// PHP (api/info.php)
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
  $type = $_GET['type'] ?? "";
  $room = $_GET['room'] ?? "";
  $query = "SELECT * FROM objects WHERE type LIKE '%$type%' AND room LIKE '%$room%'";
  // Exécution et retour JSON...
```

#### }2. Module « Visualisation »

**Public :** Utilisateurs connectés (niveau débutant/intermédiaire).

#### Technologies :

- **Frontend (AngularJS) :**
  - Contrôleur : `ProfileController` pour la gestion du profil.
  - Directive : `objectCard` pour afficher les objets avec leurs détails.
  - Service : `AuthService` pour la connexion (stockage du token en `localStorage`).
- **Backend (PHP) :**
  - Endpoint : `api/auth.php` (login/logout), `api/profile.php` (CRUD du profil).
  - Sécurité : Hashage des mots de passe avec `password_hash()`.

#### Exemple de code :

```
// AngularJS Service pour l'authentification
app.service('AuthService', function($http) {
  this.login = function(email, password) {
    return $http.post('api/auth.php', { email, password });
  };
});
```



```
// PHP (api/auth.php)
session_start();
if ($_POST['email'] === 'user@maison.com' && password_verify($_POST['password'],
$hashedPasswordFromDB)) {
    $_SESSION['user'] = ['id' => 1, 'level' => 'intermediate'];
    echo json_encode(['success' => true]);
}
```

### 3. Module « Gestion »

**Public :** Utilisateurs avancés (niveau expert).

**Technologies :**

- **Frontend (AngularJS) :**
  - Contrôleur : `ManageController` pour les scénarios et rapports.
  - Bibliothèque : `angular-chart.js` pour les graphiques de consommation.
- **Backend (PHP) :**
  - Endpoint : `api/manage.php` (gestion des objets/scénarios).
  - Logique : Calcul des statistiques côté serveur (ex : `AVG(energy_use)` en SQL).

**Exemple de code :**

```
// AngularJS : Création d'un scénario
$scope.createScenario = function() {
    ManageService.addScenario($scope.scenario).then(function() {
        $scope.showSuccessAlert();
    });
};
```

```
// PHP (api/manage.php)
if ($_POST['action'] === 'add_scenario') {
    $scenario = json_decode($_POST['data'], true);
    $stmt = $pdo->prepare("INSERT INTO scenarios (name, actions) VALUES (?, ?)");
    $stmt->execute([$scenario['name'], json_encode($scenario['actions'])]);
}
```

### 4. Module « Administration »

**Public :** Administrateurs.

**Technologies :**

- **Frontend (AngularJS) :**
  - Contrôleur : `AdminController` avec tableaux triables (`orderBy`).
  - Service : `AdminService` pour les actions critiques (ex : suppression d'utilisateur).
- **Backend (PHP) :**
  - Endpoint : `api/admin.php` avec vérification du rôle (`if ($_SESSION['user']['level'] !== 'admin')`).
  - Sécurité : Protection contre les injections SQL avec PDO.

**Exemple de code :**

```
// AngularJS : Liste des utilisateurs
$scope.users = [];
AdminService.getUsers().then(function(response) {
    $scope.users = response.data;
});

// PHP (api/admin.php)
if ($_SESSION['user']['level'] === 'admin') {
    $users = $pdo->query("SELECT id, email, level FROM users")->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($users);
}
```

## Workflow Complet

### 1. Base de données (MySQL) :

- Tables : users, objects, scenarios, logs.
- Schéma :

```
sql
Copy

CREATE TABLE users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    level ENUM('visitor', 'simple', 'complex', 'admin')
);
```

### 2. Intégration AngularJS-PHP :

- Les templates AngularJS (\*.html) appellent les API PHP via \$http.
- Le backend retourne du JSON pour une manipulation facile côté frontend.

### 3. Sécurité :

- Côté PHP :
  - Validation des entrées (filter\_var()).
  - Protection CSRF (tokens dans les formulaires).
- Côté AngularJS :
  - Interception des requêtes pour ajouter le token JWT.

### Exemple de route sécurisée :

```
php
Copy

// PHP
if (!isset($_SESSION['user']) || $_SESSION['user']['level'] !== 'admin') {
    http_response_code(403);
    die('Accès refusé');
}
```

---

## Optimisations

- **Performance :**
    - Cache des requêtes PHP avec APCu pour les données statiques.
    - Minification des assets AngularJS avec `gulp`.
  - **Responsive :**
    - Framework CSS comme Bootstrap ou Bulma intégré aux templates AngularJS.
  - **Déploiement :**
    - Backend PHP hébergé sur un serveur Apache (ex : OVH).
    - Frontend AngularJS déployé via `ng build --prod` sur Netlify.
- 

Avec cette architecture, l'équipe peut diviser les tâches :

- Un développeur sur le frontend AngularJS (modules).
- Un autre sur le backend PHP (API et sécurité).
- Un troisième sur la BDD et les rapports (requêtes SQL complexes).