



RAPPORT DE PC3R

# Système de Gestion de Location

*Author :*

Qiwei XIAN  
Ruiwen WANG

*Professeur :*

Prof.ROMAIN DEMANGEON

10 juin 2020

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>API Web choisie</b>	<b>2</b>
<b>3</b>	<b>Fonctionnalités</b>	<b>2</b>
<b>4</b>	<b>Cas d'utilisations</b>	<b>2</b>
4.1	Les fonctionnalités du module Compte . . . . .	2
4.1.1	Inscription . . . . .	2
4.1.2	Authentification . . . . .	3
4.1.3	Consulter l'espace du client . . . . .	3
4.1.4	Créer l'espace du client . . . . .	3
4.1.5	Modifier l'espace du client . . . . .	3
4.2	Les fonctionnalités du module Ressource . . . . .	4
4.2.1	Consulter les ressources . . . . .	4
<b>5</b>	<b>Stockage des données</b>	<b>4</b>
<b>6</b>	<b>Structure du serveur</b>	<b>4</b>
<b>7</b>	<b>Structure du client</b>	<b>4</b>
<b>8</b>	<b>Requêtes et réponse</b>	<b>4</b>
<b>9</b>	<b>Conclusion</b>	<b>4</b>

# 1 Introduction

Système de gestion de location est une plateforme communautaire de location et de réservation de logements personnels. Les fonctionnalités du système est ressemblé à Airbnb. Il permet aux utilisateurs de louer leur propriétés immobilières et de réserver un logement de l'autre utilisateur. Chaque l'utilisateur doit inscrire sur le système afin d'obtenir un compte, il bénéficie de la service du système en utilisant ce compte. Il peut créer une espace de client et enregistrer les informations publiques, chaque utilisateur peut consulter les informations des autres utilisateurs.

Il peut trouver des logements qui satisfait à ses besoins, par exemple sous la condition de période de location, la ville, le nombre des locataires, ainsi que le droit de fumer. L'utilisateur peut aussi gérer leur propriétés par ce système, il d'abord ajoute ses logements à louer dans le compte et met les contraintes pour les locataires, par exemple le nombre des locataires ou l'interdiction de fumer, etc. En plus il peut les modifier ou supprimer comme il veut. Lors que l'utilisateur reçoit les demandes envoyées par les autres utilisateurs. Il a droit de la refuser ou accepter, mais si la demande a risque de causer un conflit, le système va le faire remarquer au propriétaire.

## 2 API Web choisie

## 3 Fonctionnalités

Le système est séparé en trois modules, **Account**, **Demand**, **Resource**.

Le module **Account** sert à gérer les données de l'authentification et l'espace de l'utilisateur. Le module **Resource** permet de gérer les données de propriété. Et le module **Demand** sert à gérer les données des demandes.

Cela permet de faciliter le développement parallèle et la maintenance de l'application. On a réalisé les fonctionnalités de l'ajoute, de la suppression et de la consultation, ainsi que la modification de ces trois modules.

## 4 Cas d'utilisations

### 4.1 Les fonctionnalités du module Compte

#### 4.1.1 Inscription

Acteur : L'utilisateur

Contexte : L'utilisateur crée un compte.

Scénario principal :

1. L'utilisateur entre dans le page d'inscription.
2. L'utilisateur saisit l'identifiant et le mot de passe.
3. L'utilisateur clique Signup pour soumettre les informations au serveur.
4. Le système répond le message de succès et affiche sur l'écran.

Cas particulier :

- 4a. L'identifiant est déjà utilisé dans le système, le système affiche le message d'erreur.

#### 4.1.2 Authentification

Acteur : L'utilisateur

Contexte : L'utilisateur s'authentifie dans le système par son compte. Scénario principal :

1. L'utilisateur entre dans le page de l'authentification.
2. L'utilisateur saisit l'identifiant et le mot de passe.
3. L'utilisateur clique le button Login pour soumettre les informations au serveur.
4. Le serveur vérifie l'identifiant et le mot de passe.
5. Le serveur rend le page web de **mainPage.jsp**, l'identifiant est stocké dans la session.

Cas particulier :

- 4a. L'identifiant n'est pas reconnu, le système affiche le message d'erreur.
- 4b. Le mot de passe n'est pas correspondant à l'identifiant, le système affiche le message d'erreur.

#### 4.1.3 Consulter l'espace du client

Acteur : L'utilisateur

Contexte : L'utilisateur connecté veut consulter son espace du client. Scénario principal :

1. L'utilisateur clique le button Profile dans mainPage.
2. Le serveur génère dynamiquement l'espace personnelle de l'utilisateur.

cas particulier :

- 2a. L'utilisateur n'a pas encore son espace du client, le serveur génère le page de creation d'espace, permet à l'utilisateur de créer son espace.

#### 4.1.4 Créer l'espace du client

Acteur : L'utilisateur

Contexte : L'utilisateur consulte sont espace première fois Scénario principal :

1. L'utilisateur clique le button Profile première fois dans le mainPage.
2. Le serveur repond un message et génère dynamiquement le formulaire pour créer l'espace du client.
3. L'utilisateur saisit les informations publiques.
4. L'utilisateur clique le button Créer afin de soumettre les information.
5. Le serveur crée l'espace du client et le stocke dans la base de données, ainsi que génère le page web de l'espace du client.

#### 4.1.5 Modifier l'espace du client

Acteur : L'utilisateur Contexte : L'utilisateur connecté se trouve dans son espace du client. Scénario principal :

1. L'utilisateur clique le button **here** afin de modifier son
2. Le serveur génère le page permet au client de modifier son espace.

## 4.2 Les fonctionnalités du module Ressource

### 4.2.1 Consulter les ressources

Acteur : L'utilisateur

Contexte : L'utilisateur s'authenticé se trouve à mainPage. Scénario principal :

1. L'utilisateur clique le button **Your house**.
2. Le serveur cherche dans la base de donnée et génère dynamiquement le page sert à afficher les informations de propriétés.

### 4.2.2 Ajouter la ressource

Acteur : L'utilisateur

Contexte : L'utilisateur connecté se trouve dans le page **Your House**. Scénario principal :

1. L'utilisateur clique le button **Add House**.
2. La page est générée dynamiquement par js permet à l'utilisateur de saisir les informations de la ressource.
3. L'utilisateur clique le button **Créer** pour soumettre.
4. Le serveur vérifie que le status de l'authentification de l'utilisateur est connecté.
5. Le serveur stocke les informations de la ressource.
6. Le serveur affiche un message de success sur l'écran.

## 5 Stockage des données

## 6 Structure du serveur

## 7 Structure du client

## 8 Requêtes et réponse

## 9 Conclusion