|  |
| --- |
| 4 MIN EAM |
| Projet architecture Web |
| Enseignant : Mr DEKIMPE |

|  |
| --- |
| Matthias Léonard 20308  2022/2023 |

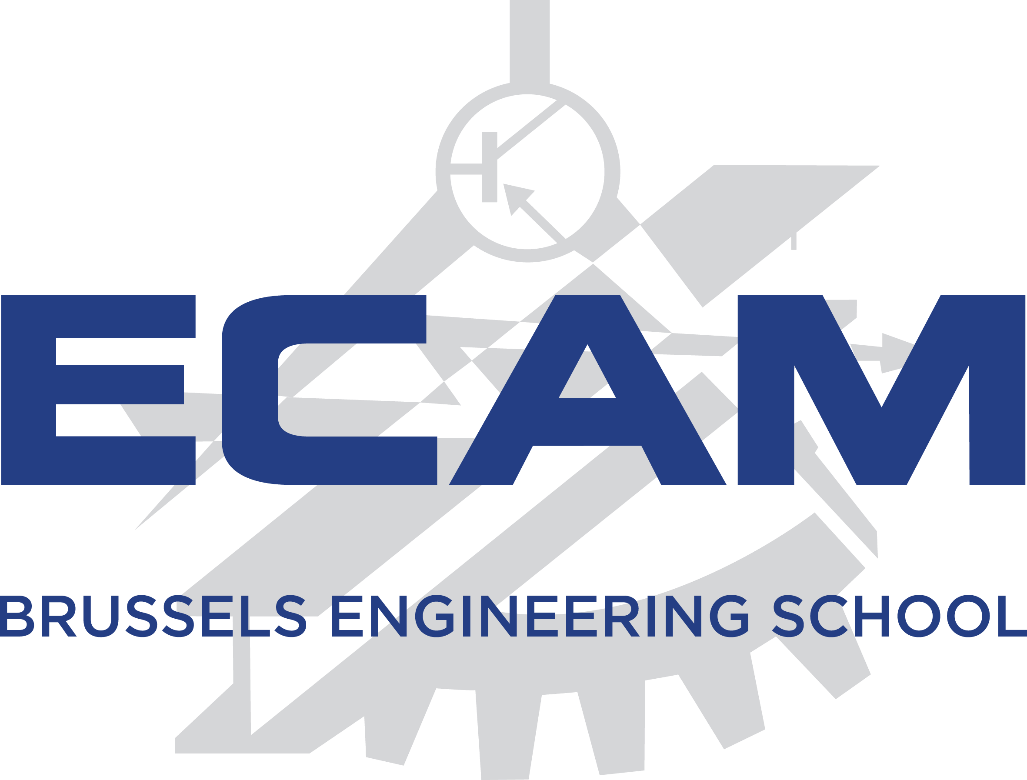


Table des matières

[Introduction : 2](#_Toc135129809)

[User stories : 2](#_Toc135129810)

[MockUP : 3](#_Toc135129811)

[Diagramme entité association : 5](#_Toc135129812)

[Diagramme relationnel : 5](#_Toc135129813)

[BackEnd : 5](#_Toc135129814)

[Description des champs de la base de données : 6](#_Toc135129815)

[Table appartements: 6](#_Toc135129816)

[Table renters: 6](#_Toc135129817)

[Tables repair\_estimates : 6](#_Toc135129818)

[Table tickets : 6](#_Toc135129819)

[Description des tables et des relations entre tables : 6](#_Toc135129820)

[ORM : 7](#_Toc135129821)

[API : 7](#_Toc135129822)

[Authentification : 8](#_Toc135129823)

[FrontEnd Angular: 8](#_Toc135129824)

[Architecture: 11](#_Toc135129825)

[FrontEnd App mobile hybride Ionic: 12](#_Toc135129826)

[Conclusion : 14](#_Toc135129827)

# Introduction :

Dans le cadre du cours de web architecture du cursus 4MIN il nous devons de réaliser un projet intégrant une API pour un développement front-end et back-end.

Le choix de mon projet est basé sur une agence immobilière, celle-ci qui doit pouvoir gérer sont portefeuille de biens immobilier mis à la location. La gestion des problèmes techniques et les locataires des biens.

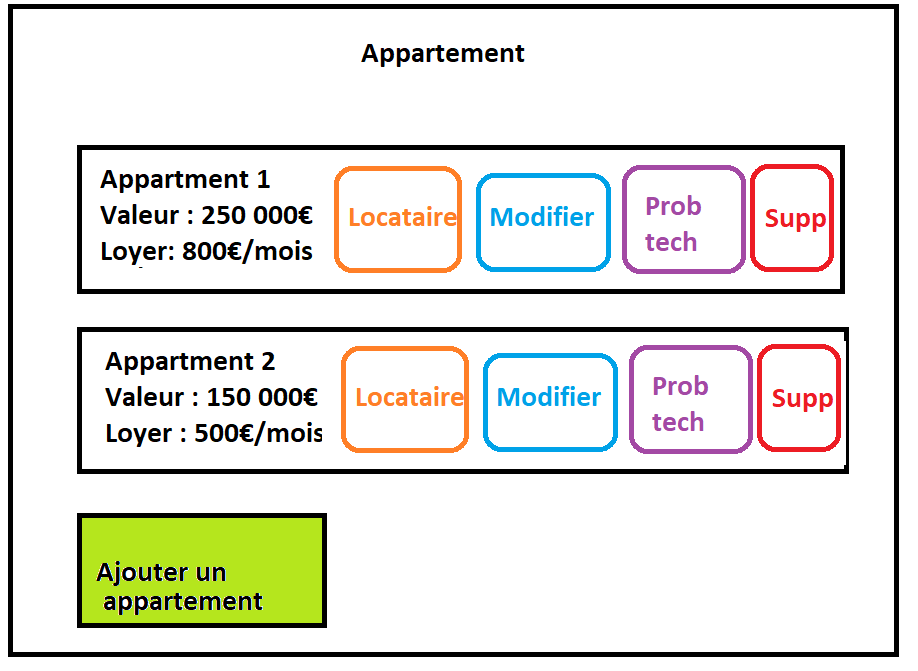
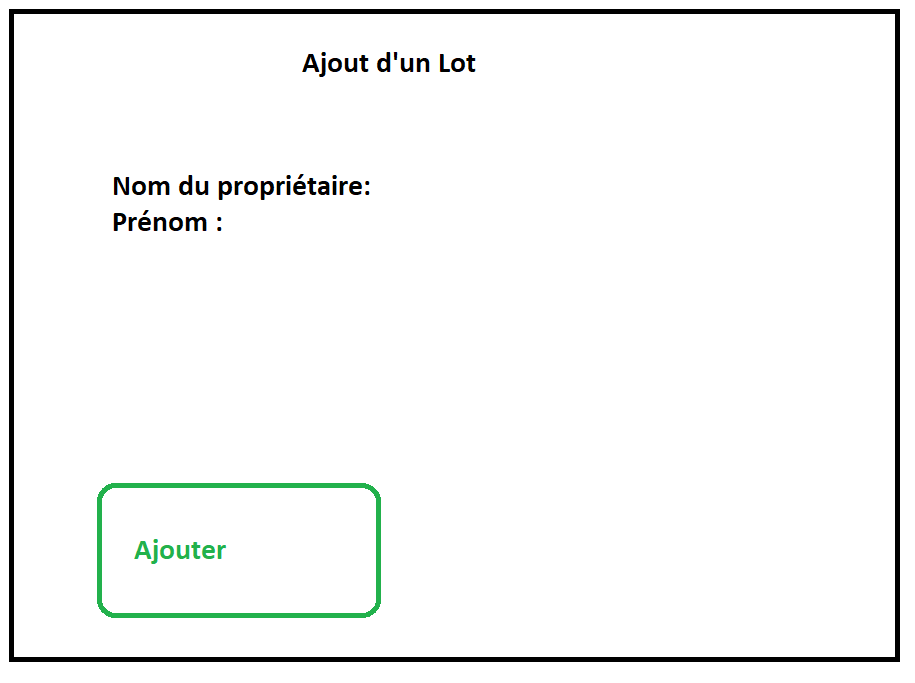
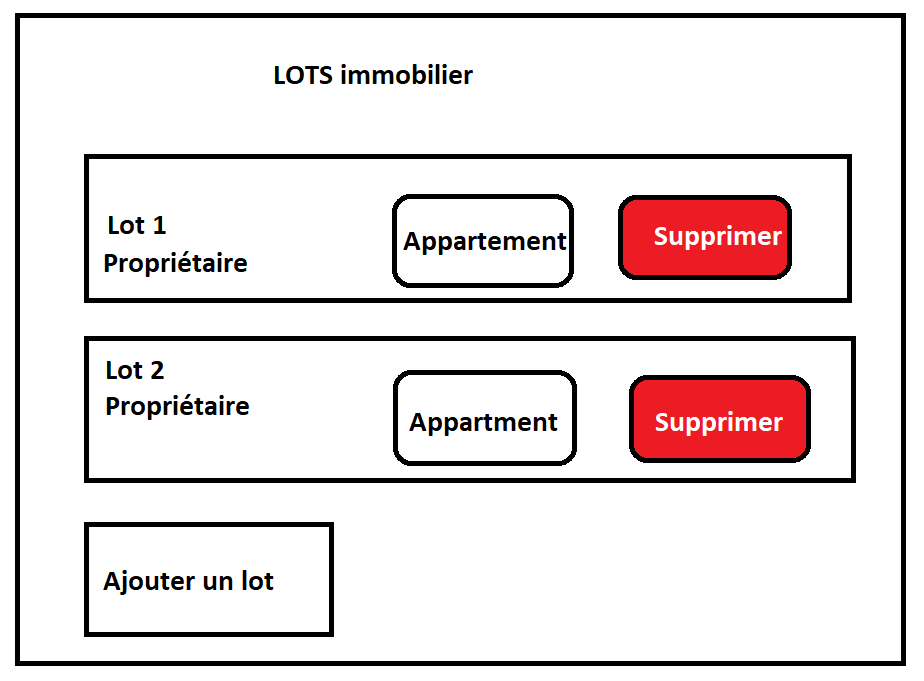
Dans ce projet nous nous limiterons aux problèmes techniques associer des devis, associer des locataires à des biens. Il est bien évidemment possible d’approfondir ce projet notamment par la gestion des loyers et l’exercice comptable, la gestion de plusieurs propriétaires d’un bien, ces exemples ne sont pas exhaustifs.

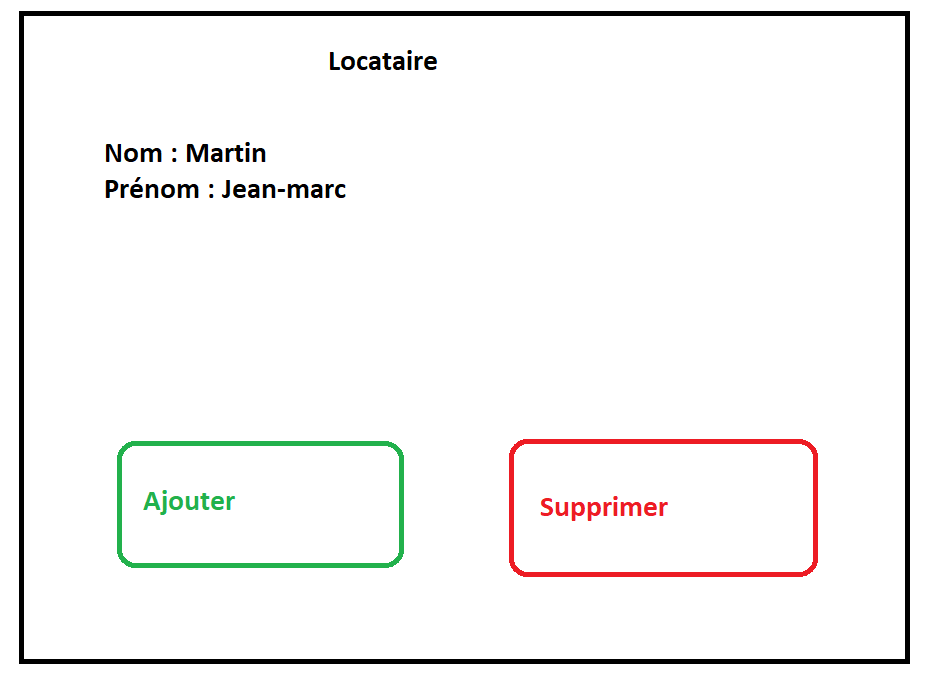
Lien GitHub : <https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere>

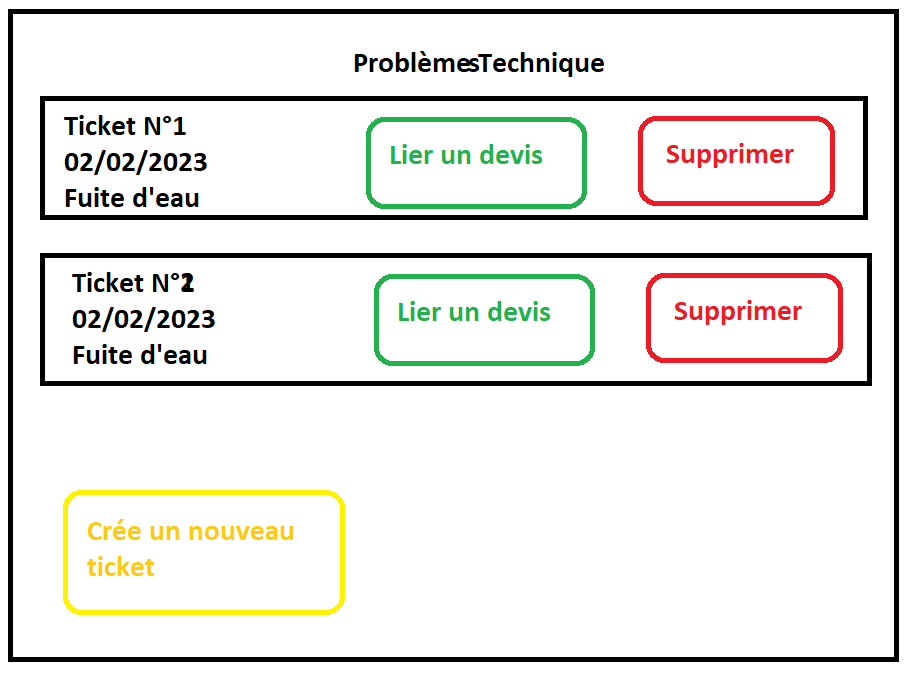
# User stories :

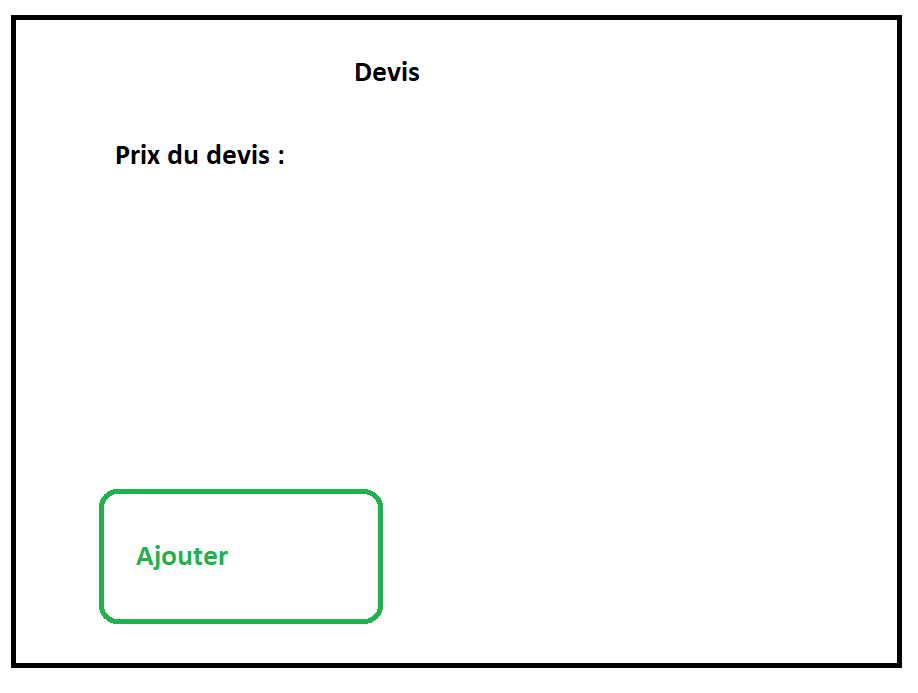
|  |  |
| --- | --- |
| **User stories** | **Priorité** |
| **Gestion des classes de biens** |  |
| En tant qu’agence immobilière, je veux pouvoir ajouter un bien d’un propriétaire dans un lot | 1 |
| En tant qu’agence immobilière, je veux pouvoir supprimer un bien d’un lot | 1 |
| **Gestion d’un bien** | |
| En tant qu’agence immobilière, je veux pouvoir crée un bien | 1 |
| En tant qu’agence immobilière, je veux pouvoir supprimer un bien | 1 |
| En tant qu’agence immobilière, je veux pouvoir modifier un bien | 3 |
| **Gérer les locataires** | |
| En tant qu’agence immobilière, je veux pouvoir ajouter un locataire à un bien | 4 |
| En tant qu’agence immobilière, je veux pouvoir supprimer un locataire d’un bien | 4 |
| **Gérer les problèmes technique** | |
| En tant qu’agence immobilière, je veux pouvoir crée un ticket pour un problème technique | 2 |
| En tant qu’agence immobilière, je veux pouvoir supprimer un ticket | 3 |
| En tant qu’agence immobilière, je veux pouvoir lister les tickets | 2 |
| En tant qu’agence immobilière, je veux pouvoir associer un devis à un ticket | 2 |

# MockUP :

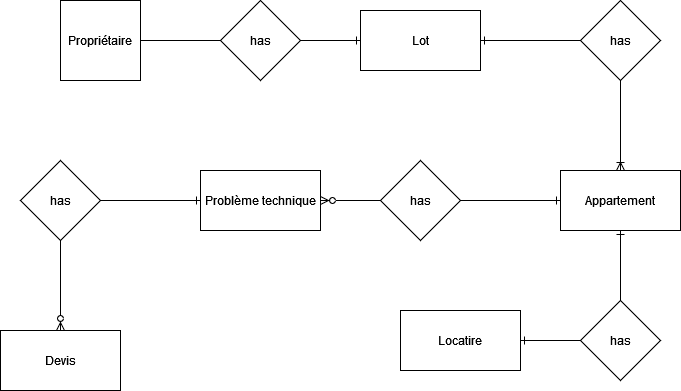








# Diagramme entité association :



(0,1)

(n,1)

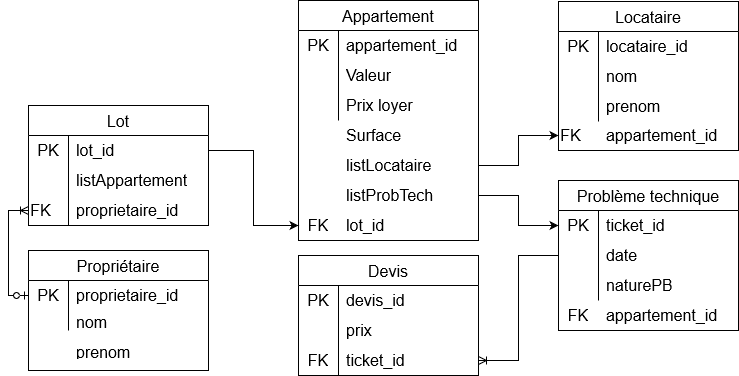
(1,1)

(1,1)

(0, n)

(1,1)

# Diagramme relationnel :

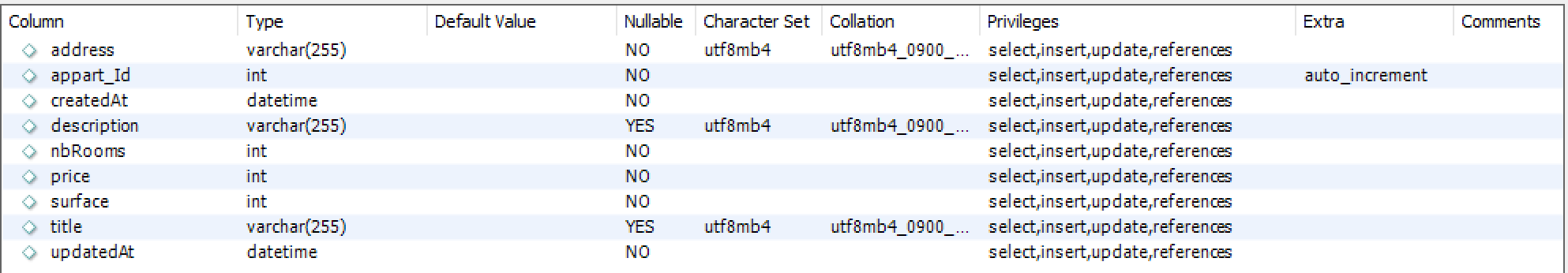


# BackEnd :

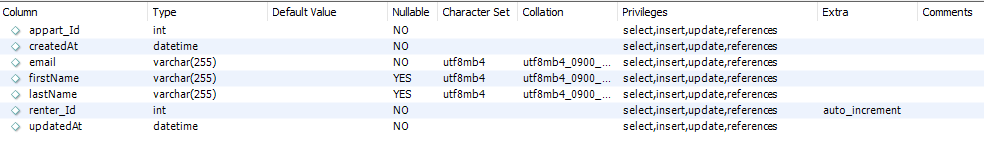
Github : <https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere/tree/main/WEB/BackEnd>

## Description des champs de la base de données :

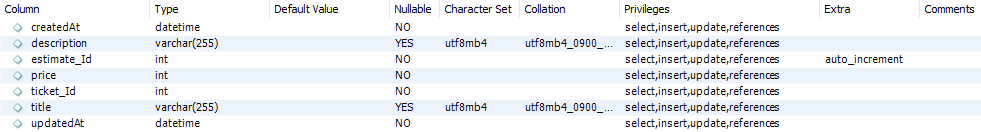
### Table appartements:



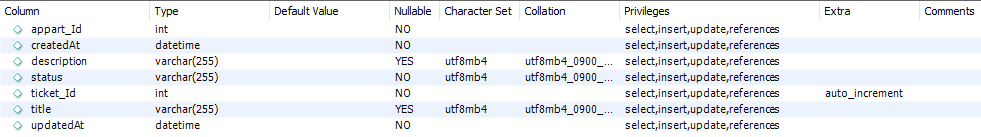
### Table renters:



### Tables repair\_estimates :



### Table tickets :



## Description des tables et des relations entre tables :

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Table | Type | Description |
| appartements | renters | 1 à 1 | Un appartement à un locataire et un locataire à un appartement. |
| appartements | tickets | 0 à N | Un appartement peut avoir aucun ticket ou plusieurs tickets, un ticket appartient à un appartement. |
| tickets | repair\_estimates | 1 à N | Un ticket peut avoir plusieurs devis et un devis appartient à un ticket. |

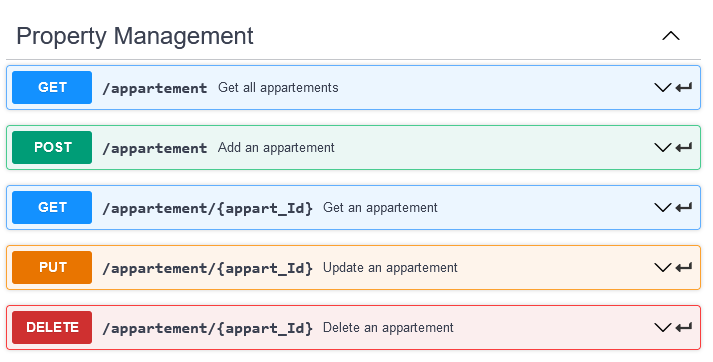
## ORM :

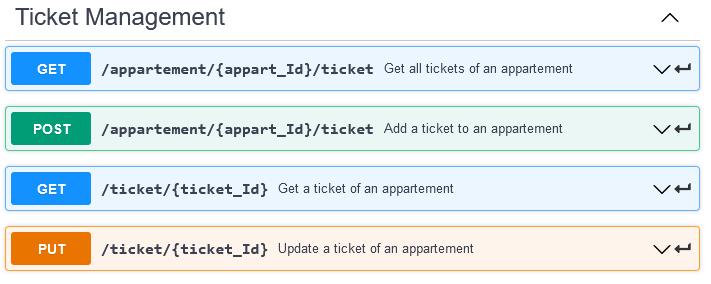
<https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere/blob/main/WEB/BackEnd/models/index.js>

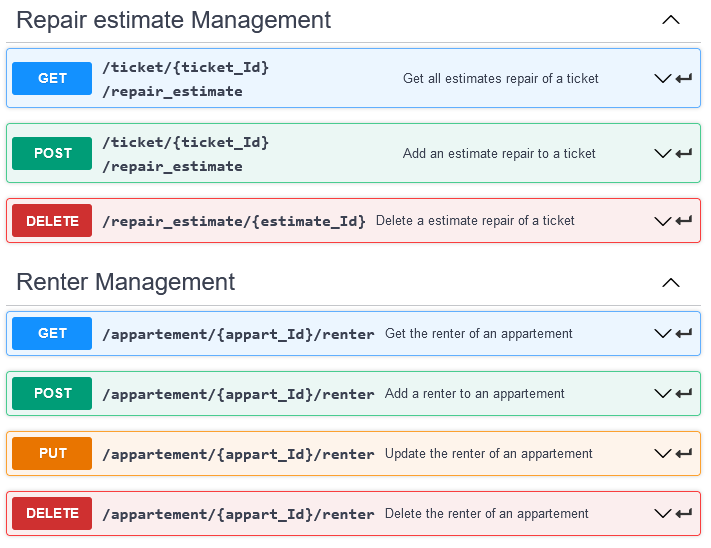
## API :

L’API est disponible en consultation sur le site de swaggerHub : <https://app.swaggerhub.com/apis/LeTouristeDeLECAM/Agence_Immobiliere/1.0>

Ou sur GitHub sous format YAML : <https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere/blob/main/WEB/API/API_estate_agency.yaml>







## Authentification :

L’authentification n’ayant pas été réfléchis lors de la phase de création du projet, celle-ci à été rapidement implémenté pour être en phase avec les exercices du cours plutôt qu’une réelle authentification. Ceci implique que dans le projet l’authentification est réalisée par une simple requête GET sur le chemin « /login », dans le monde réel il s’agit d’une faille de sécurité majeur.

Dans un monde idéal nous devrions crée dans la base de données une table users comportant une clé primaire, le nom d’utilisateur, et le Hash du mot de passe.

Lorsqu’un utilisateur souhaite s’identifier, le backEnd reçoit le nom d’utilisateur et le Hash du mot de passe qui sont placé dans les paramètres de la requête. Du coté backend il est recherché une correspondance entre le couple nom d’utilisateur et hash du mot de passe.

Lorsque la correspondance est effectuée le token d’identification est alors renvoyé, Sinon une erreur est envoyé pour un problème de correspondance.

# FrontEnd Angular:

Github : <https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere/tree/main/WEB/FrontEnd/real_estate_angular>

Les chemins possibles sont les suivants :

*Login fig1 -> Home (Property list) fig2 -> Ticket fig3 -> Estimate fig4*

*Home (Property list) fig2 -> Renter fig5*

Il est également possible de rechercher directement un Ticket par son ID dans la page principale.

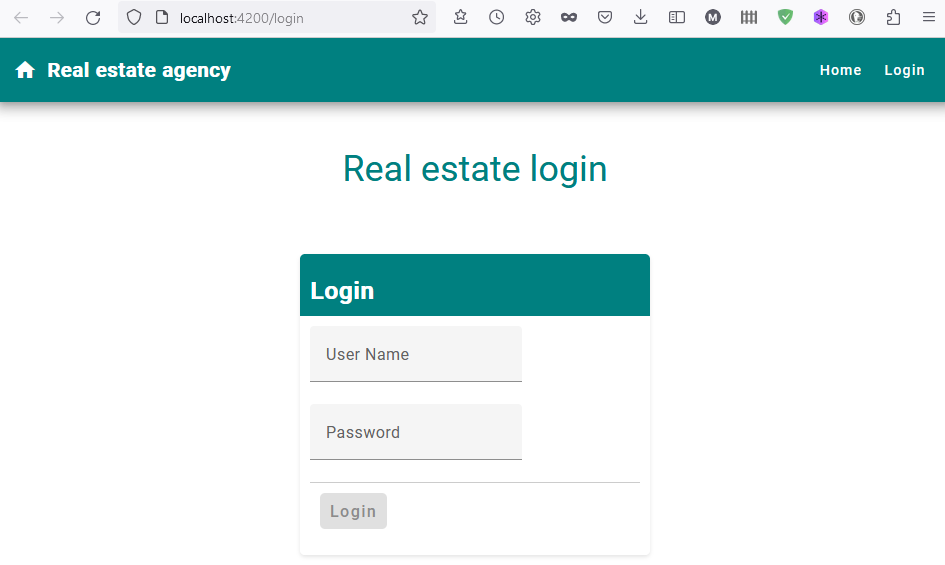


Figure : Page de login

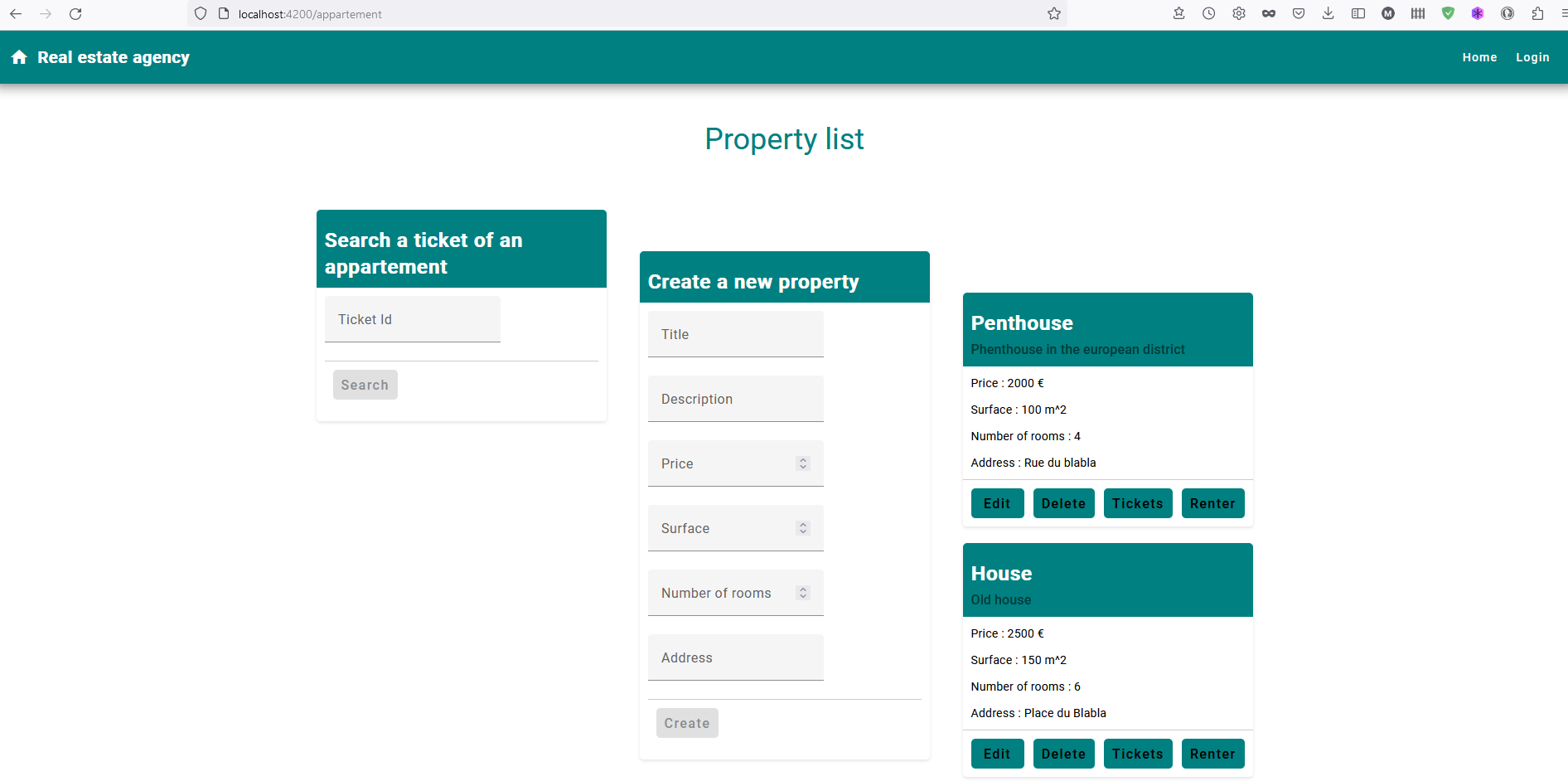


Figure : Home, liste des biens

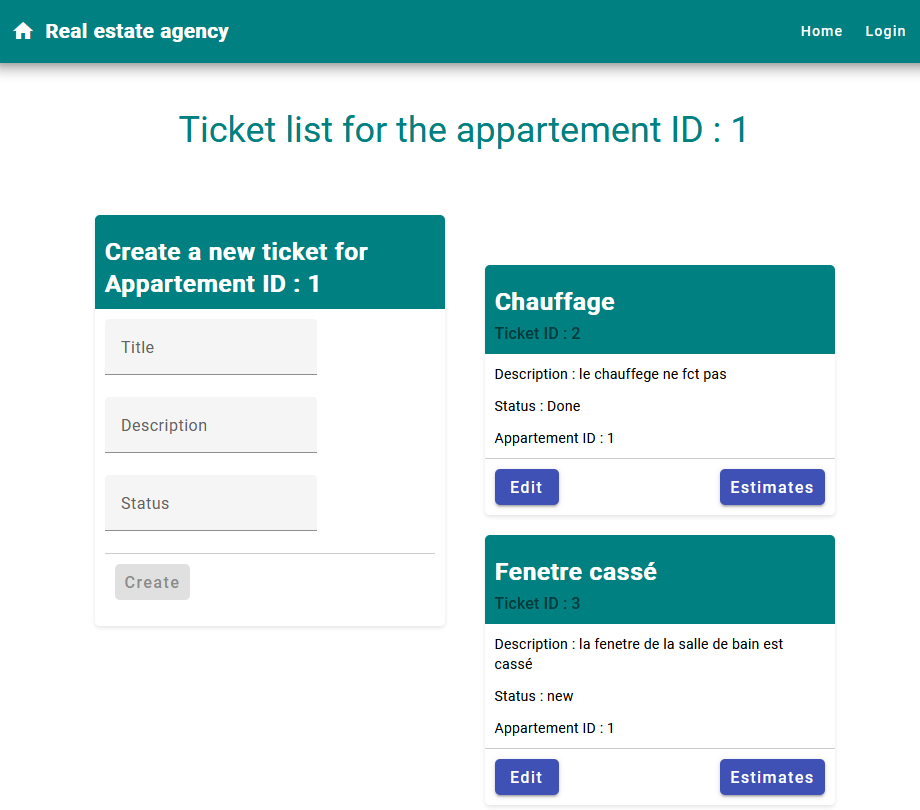


Figure : Liste des tickets

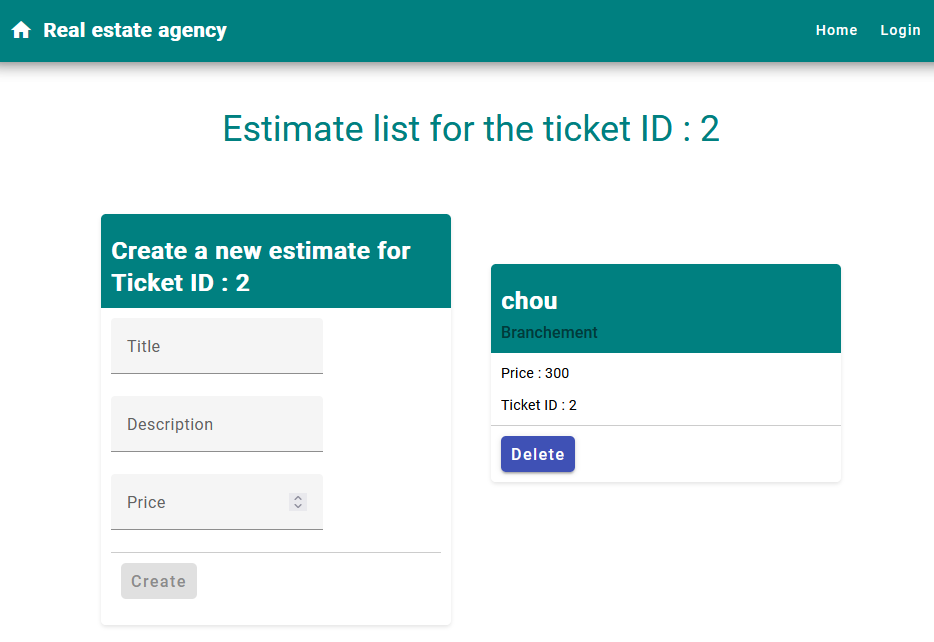


Figure : Liste des devis

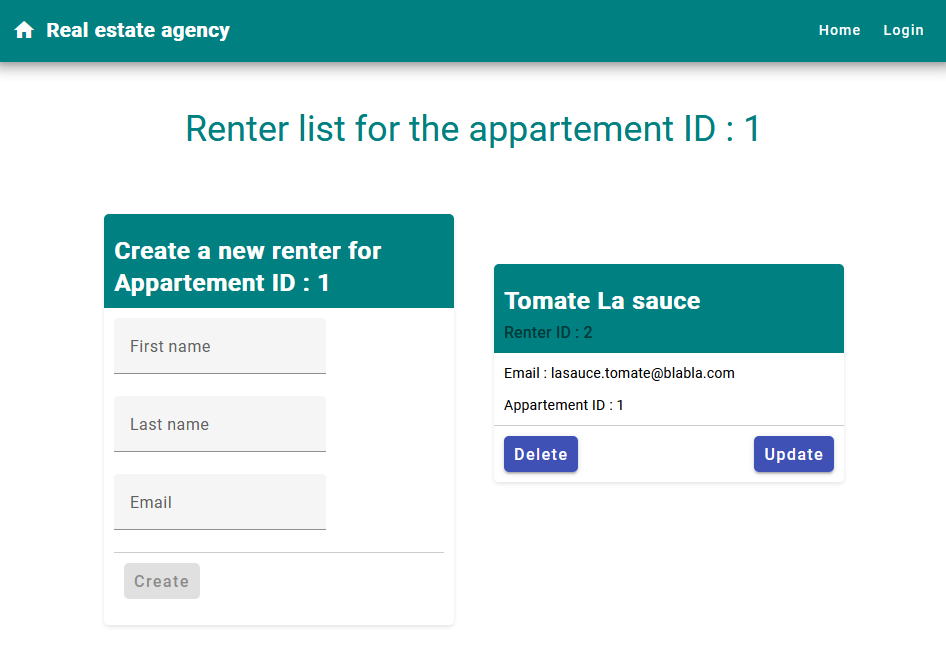


Figure : Information locataire

## Architecture:

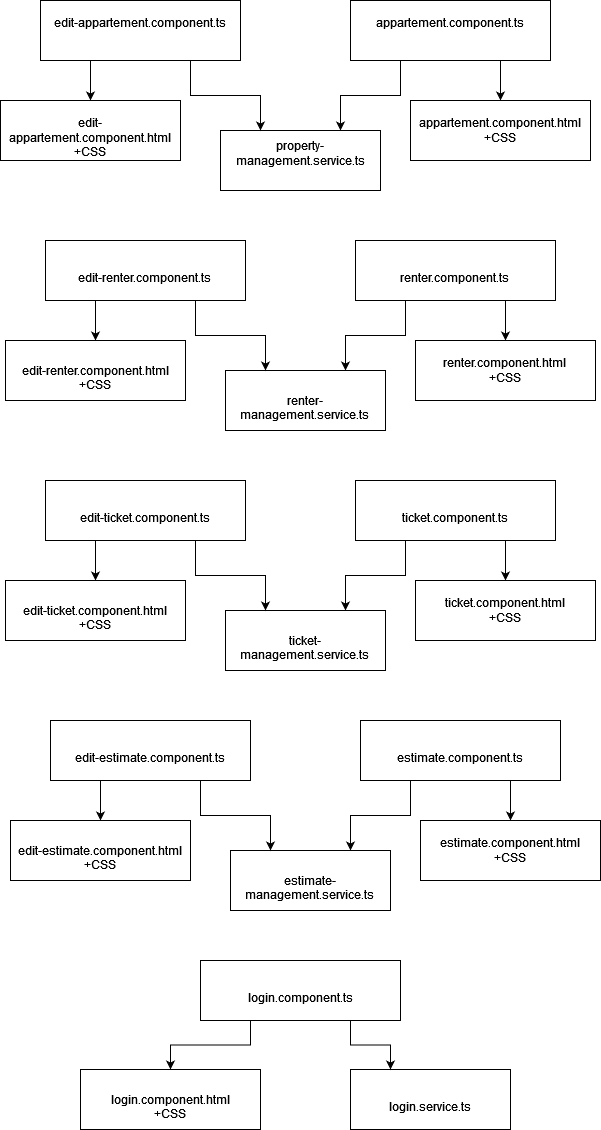


Figure : Architecture componant + service

# FrontEnd App mobile hybride Ionic:

Github: <https://github.com/LeTouristeDeLECAM/Projet-Web-architecture-Agence-Immobiliere/tree/main/WEB/FrontEnd/real_estate_ionic>

La partie Ionic comporte des plusieurs problèmes, ceux-ci sont uniquement sur la page appartement :

* La fenêtre est divisée en trois parties
* Les liens de ne fonctionnement pas

Comme pour la partie WEB conventionnelle les chemins sont les suivants :

*Home (Property list) fig6 -> Ticket fig9 -> Estimate fig10*

*Home (Property list) fig6 -> Renter fig7*

*Login fig8*

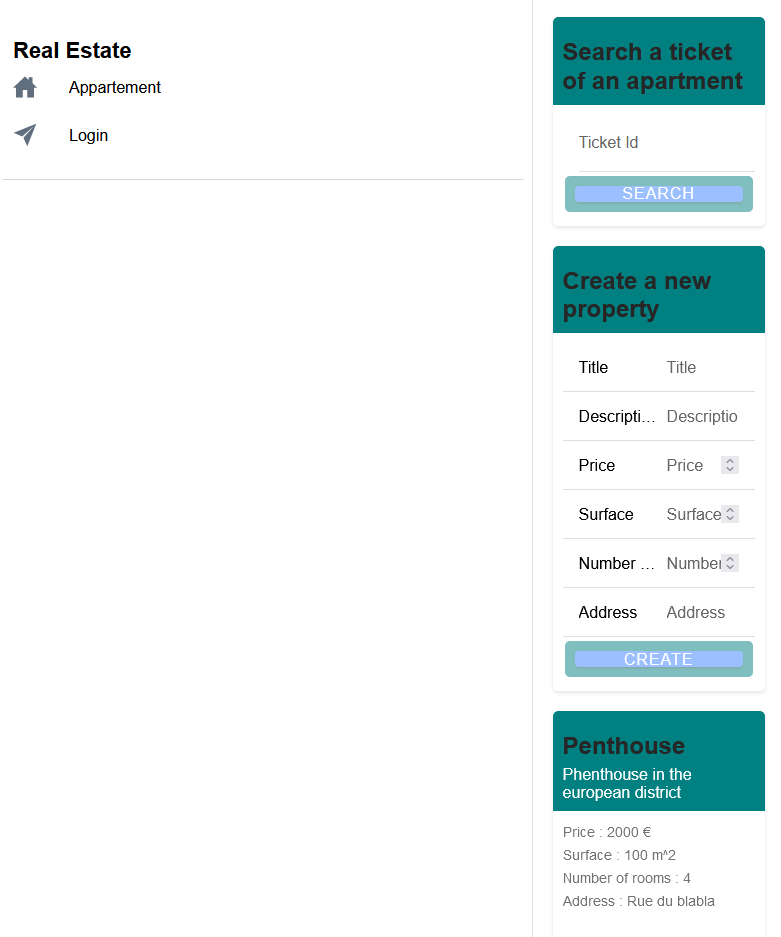


Figure : Home, liste des biens

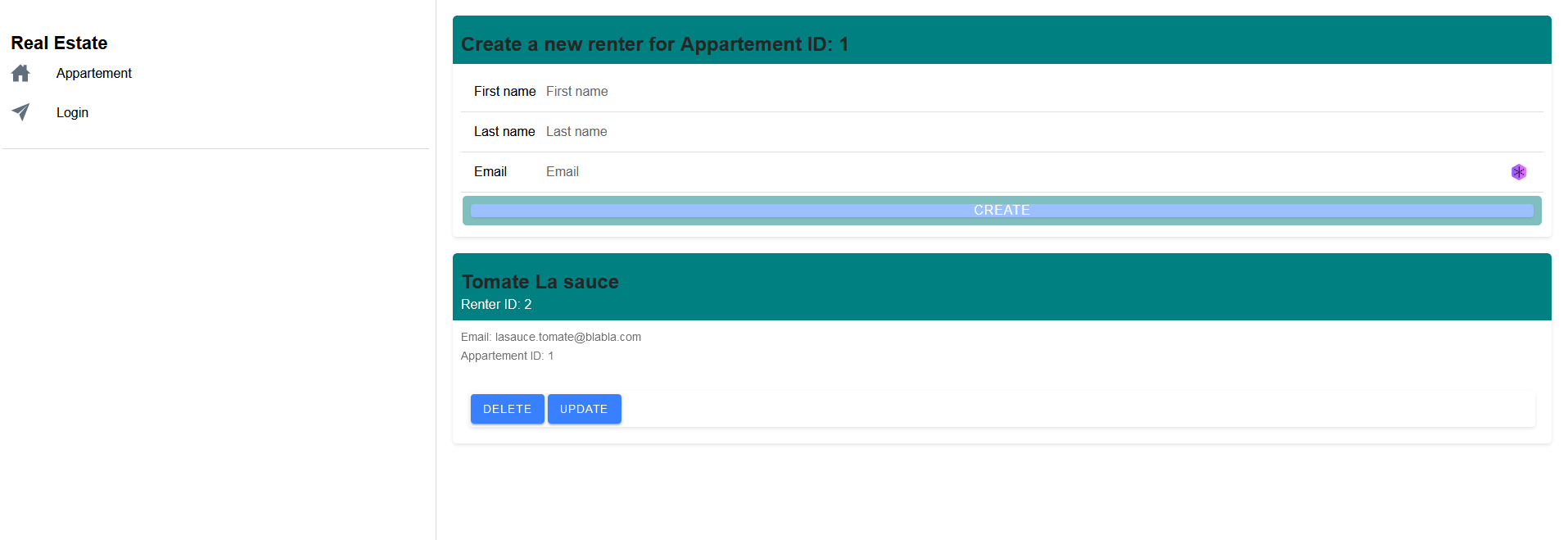


Figure : Information locataire

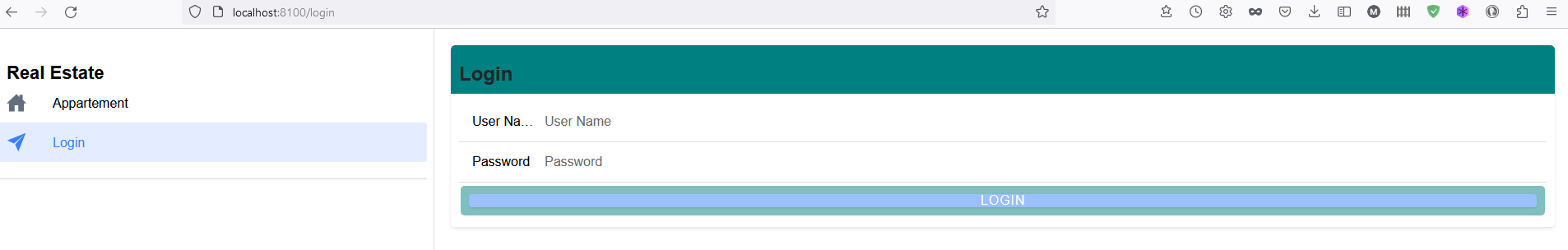


Figure : Page de Login

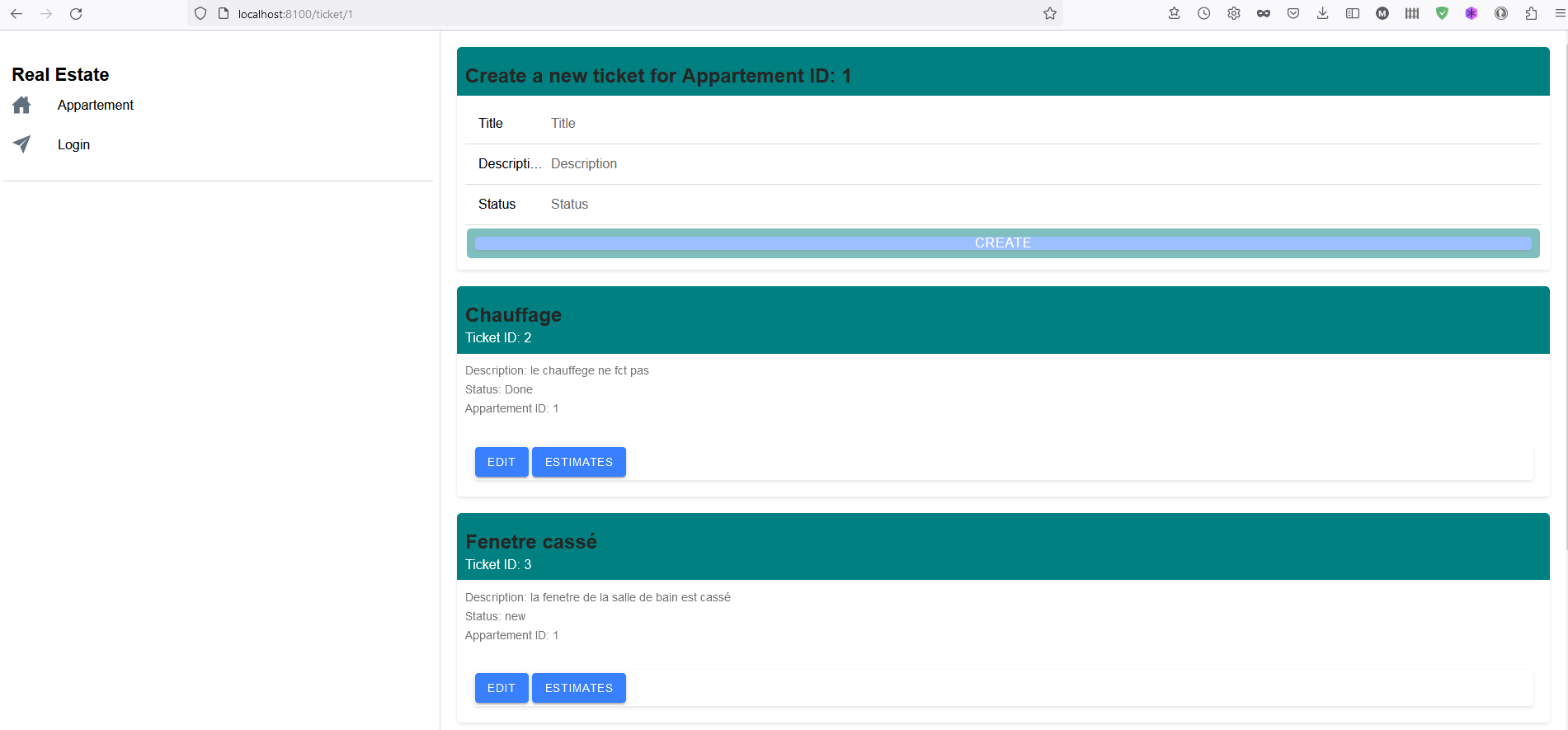


Figure : Liste des tickets

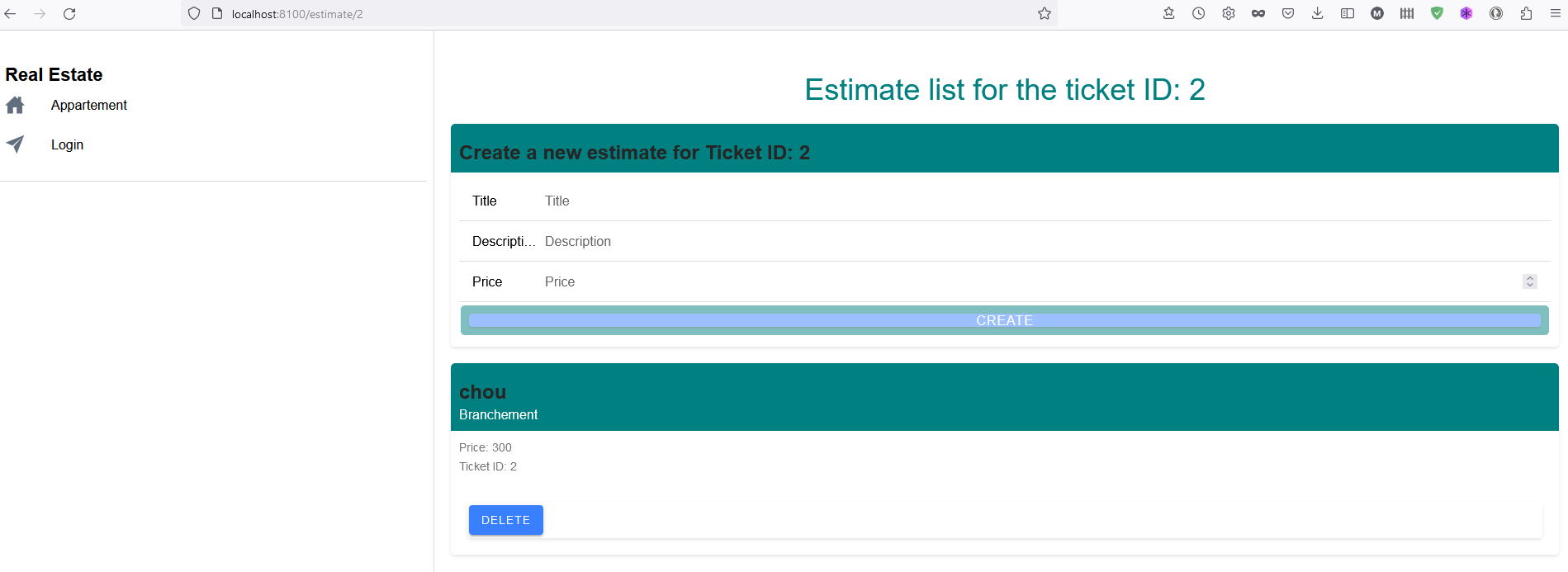


Figure : Liste des devis

# Conclusion :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User stories** | **Priorité** | **API doc et dev** | **Frontend Angular** | **Reste à faire/ changement** |
| **Gestion des classes de biens** |  |  |  |  |
| En tant qu’agence immobilière, je veux pouvoir ajouter un bien d’un propriétaire dans un lot | 1 | 0% | 0% | Ne fait plus partie du projet |
| En tant qu’agence immobilière, je veux pouvoir supprimer un bien d’un lot | 1 | 0% | 0% | Ne fait plus partie du projet |
| **Gestion d’un bien** | |  |  |  |
| En tant qu’agence immobilière, je veux pouvoir crée un bien | 1 | 100% | 100% |  |
| En tant qu’agence immobilière, je veux pouvoir supprimer un bien | 1 | 100% | 100% |  |
| En tant qu’agence immobilière, je veux pouvoir modifier un bien | 3 | 100% | 90% | Afficher les anciennes données |
| **Gérer les locataires** | |  |  |  |
| En tant qu’agence immobilière, je veux pouvoir ajouter un locataire à un bien | 4 | 100% | 100% |  |
| En tant qu’agence immobilière, je veux pouvoir supprimer un locataire d’un bien | 4 | 100% | 100% |  |
| En tant qu’agence, je veux pouvoir modifier un locataire |  | 100% | 90% | Ajout au projet , afficher les anciennes données |
| **Gérer les problèmes technique** | |  |  |  |
| En tant qu’agence immobilière, je veux pouvoir crée un ticket pour un problème technique | 2 | 100% | 100% |  |
| En tant qu’agence immobilière, je veux pouvoir supprimer un ticket | 3 | 0% | 0% | Ne fait plus partie du projet |
| En tant qu’agence immobilière, je veux pouvoir lister les tickets | 2 | 100% | 100% |  |
| En tant qu’agence immobilière, je veux pouvoir associer un devis à un ticket | 2 | 100% | 100% |  |
| **Gérer les devis** | | | | |
| En tant qu’agence je veux pouvoir supprimer un devis |  | 100% | 100% | Ajout au projet |
| En tant qu’agence je veux pouvoir lister les devis |  | 100% | 100% | Ajout au projet |
| **Authentification** | | | | |
| En tant qu’agence je veux pouvoir m’authentifier pour réaliser des modifications à la base de données |  | 25% | 90% | Coté Backend création d’une table User, Hash et recherche du match user et hash |
| **Rechercher** |  |  |  |  |
| Entant qu’agence je veux pouvoir chercher un ticket par son ID |  | 100% | 100% | Ajout au projet |

A mesure de l’avancement du projet beaucoup de nouveaux éléments ont été ajouté et implémenté au projet, il en est de même pour les éléments supprimés du projet, ceci est due à une meilleure compréhension du projet au fur et à mesure de son développement.

Pour exemple :

* Du coté backend il a été ajouté la modification des données d’un locataire, la suppression de devis et d’autres éléments. A contrario la création de lot et la modification des lots à été supprimé du projet.
* Du coté Frontend il à été implémenté une authentification via token envoyé par le backend. Il reste encore à construire les tables et la recherche de correspondance entre user et hash du coté backend.

Ce dynamisme a été rendu possible par la démarche agile suivie tout au long du projet.

La symbiose entre la méthode agile et mon propre apprentissage on permis de rediriger le projet pour qu’il corresponde aux exercices, avec du recul et des compétences accrues sur le sujet il est aujourd’hui plus facile d’imaginer crée un projet avec des bases solide en minimisant les changements et identifier les points prioritaires.