

Licence 3 Informatique
Projet de programmation Web

.
Immob Porto

Par
Donatien Yeto

11 Mai 2021 au 01 Juin 2021

Chargé du cours
Dr. Amadou Tidjani SANDA MAHAMA

Sommaire

I	Introduction	3
0.1	Introduction	5
II	Étude préalable	6
0.2	La description du contexte du projet	8
0.3	L'identification des besoins	8
0.3.1	Les besoins fonctionnels	8
0.3.2	Les besoins non fonctionnels	8
0.4	L'analyse des besoins	9
0.4.1	L'identification des acteurs et des cas d'utilisations de l'application . . .	9
0.4.2	Le diagramme des cas d'utilisations.	10
0.4.3	Description textuelle de quelques cas	10
III	Conception	12
0.5	Diagramme des classes	14
0.6	Passage au relationnel	14
IV	Réalisation	16
0.7	Environnement et outils de développement de notre application	18
0.8	Présentation de quelques interfaces de l'application	19
V	Hébergement	22
0.9	Notion d'hébergement	24
0.10	Hébergement	24
VI	Conclusion	25
0.11	Conclusion	27

Partie I

Introduction

0.1 Introduction

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela à l'aide de l'informatique, qui joue un rôle important dans le développement de nombreuses entreprises et organisations.

En quelques années, des outils informatiques performants se sont imposés dans les agences immobilières, que ce soit pour répertorier des biens, dynamiser les échanges entre confrères, déterminer un prix de vente... Leur atout : favoriser le travail des professionnels. Les agents immobiliers peuvent aujourd'hui gérer leurs activités grâce à des logiciels d'immobilier performants.

Nous avons pu constater, en effet, pendant notre observation, que la plupart du temps que les particuliers, pour trouver un logement à louer ou acheter passent par des démarcheurs ou négociateurs, et ne sont pas souvent satisfaits de leurs services. En plus le délai de retour de ceux-ci est souvent long.

L'objectif de notre projet présenté dans ce rapport est la conception et la réalisation d'une application web (**Immob Porto**) pour faciliter la recherche de logements à louer ou à vendre.

Après avoir présenté la problématique, identifié et analysé les besoins en présentant les cas d'utilisation sous forme de diagramme, nous passerons à l'implémentation et la réalisation de notre application, où nous présenterons l'environnement de développement, la structure de notre application et quelques interfaces de celle-ci. Ensuite nous aborderons l'hébergement avant de conclure.

Partie II

Étude préalable

0.2 La description du contexte du projet

ImmobPorto vient répondre aux problèmes ci-haut mentionnés en facilitant la recherche de logements que ce soit pour l'achat ou la location en répondant aux exigences des utilisateurs d'une part et en permettant à ces derniers d'y publier des annonces de demandes ou d'offres. Ainsi, un utilisateur ayant accès à la plateforme pourra directement parcourir les annonces sur son fil d'actualité, publier des annonces d'offres en proposant des logements à vendre ou à louer, des annonces de demandes s'il est dans le besoin urgent.

0.3 L'identification des besoins

0.3.1 Les besoins fonctionnels

Les besoins fonctionnels sont les besoins spécifiant un comportement d'entrée/sortie du système. L'application doit permettre de :

- Tout internaute doit pouvoir parcourir les annonces publiées
- Rechercher une annonce
- Visiter une annonce
- Souscrire pour une annonce et prendre un rendez-vous pour la visite des lieux
- Gérer son compte
 - Modifier son profil
 - Compléter son profil
- Gérer les annonces d'offre ou de demandes
 - Publier les annonces
 - Modifier les annonces
 - Supprimer les annonces
 - Activer une annonce
 - Désactiver une annonce
- Contacter un utilisateur en cas de besoin
- Administrer les comptes utilisateurs
 - Supprimer un compte
 - Supprimer une annonce

0.3.2 Les besoins non fonctionnels

Pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques, on doit prendre en considération les contraintes qui caractérisent ce système. L'application doit nécessairement assurer les besoins suivants :

- Les informations de consultation des offres sont accessibles aux internautes mais toute autre opération n'est accessible que par authentification.

- Chaque utilisateur ne doit pas accéder à des services s'il n'a pas les permissions de ceux-ci.
- L'application doit respecter les principes d'IHM.
- L'application doit être facile à prendre en main et sécurisée.
- L'application doit être portable.

0.4 L'analyse des besoins

Nous traitons dans cette partie l'analyse fonctionnelle de notre projet. D'abord nous identifions les acteurs impliqués, ensuite nous spécifions les cas d'utilisations de notre application.

0.4.1 L'identification des acteurs et des cas d'utilisations de l'application

Les différents acteurs qu'on va trouver dans nos diagrammes de cas d'utilisation sont :

1. **Anonyme** : c'est l'utilisateur sans compte. Il peut

- Parcourir les annonces
- Rechercher une annonce
- Consulter une annone
- S'inscrire

2. **Utilisateur** : l'acteur principal dans notre application, c'est lui qui

- Gérer une annonce (publier, modifier, supprimer, activer, désactiver)
- Gérer son compte
- Souscrire pour une annonce
- Prendre rendez-vous de visite
- Contacter un autre utilisateur

3. **Administrateur** : le responsable de la gestion depuis la conception jusqu'à la maintenance de l'application web. Il peut

- Supprimer un compte
- Supprimer une annonce

0.4.2 Le diagramme des cas d'utilisations.

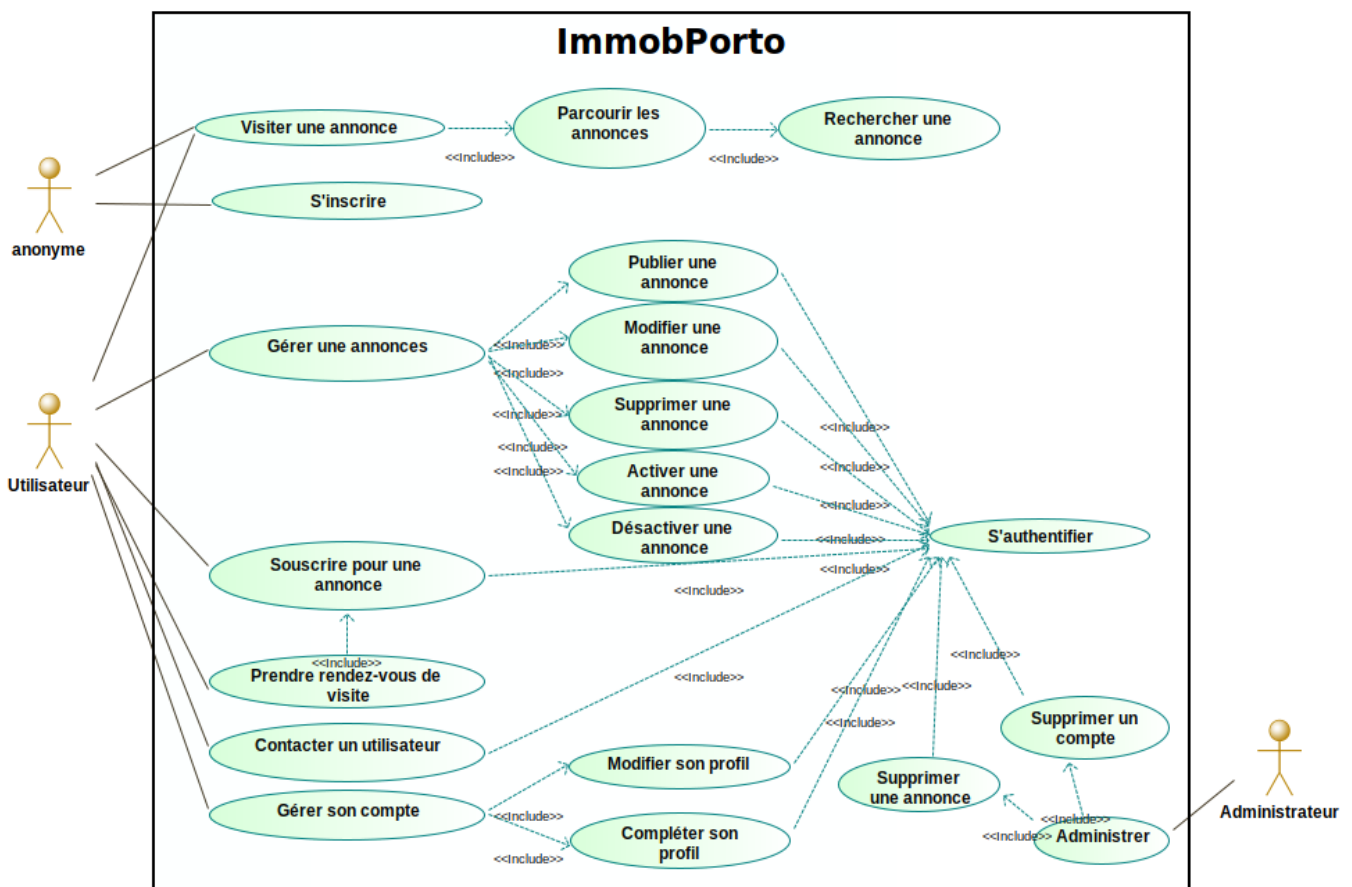


Figure 1: Diagramme des cas d'utilisations

0.4.3 Description textuelle de quelques cas

Le cas d'utilisation s'authentifier

1. Identification du cas

- **Acteur principal** : utilisateur, administrateur
- **Objectif** : S'assurer que l'utilisateur est bien celui qui prétant être.

2. Les enchaînements

- **Les préconditions**
 - L'acteur a un compte
 - Le système est fonctionnel
- **Scenario nominal**
 1. L'acteur saisit son email et son mot de passe.
 2. Le système vérifie l'email et le mot de passe.
 3. Le système affiche l'espace approprié pour chaque acteur.
- **Scenario alternatif** :
 - A1. L'email ou mot de passe incorrects
L'enchaînement démarre au point 2. du scenario nominal.

3. Le système notifie à l'acteur que l'email ou le mot de passe est incorrect

L'enchaînement redémarre au point 1.

Le cas d'utilisation publier une annonce

1. Identification du cas

- **Acteur principal** : utilisateur
- **Objectif** : Permettre à l'utilisateur de proposer ou de demander un service de location ou de vente immobilière.

2. Les enchaînements

- **Les préconditions**

- L'utilisateur s'est authentifié
- Le système est fonctionnel

- **Scenario nominal**

1. L'utilisateur clique sur publier annonce
2. Le système demande le type d'annonce (offre ou demande)
3. L'utilisateur fournit le type
4. Le système affiche le formulaire correspondant
5. L'utilisateur remplit le formulaire
6. L'utilisateur soumet le formulaire
7. Le système vérifie la validité des données du formulaire
8. Le système sauvegarde les données du formulaire
9. Le système notifie à l'utilisateur que l'annonce a été bien publiée.

- **Scenario alternatif :**

- A1. Quelques données du formulaire sont non valide
L'enchaînement démarre au point 7. du scenario nominal.
8. Le système le notifie à l'utilisateur
L'enchaînement redémarre au point 4.

Partie III

Conception

0.5 Diagramme des classes

Un diagramme des classes décrit le type des objets ou données du système ainsi que les différentes formes de relations statiques qui existent entre eux.

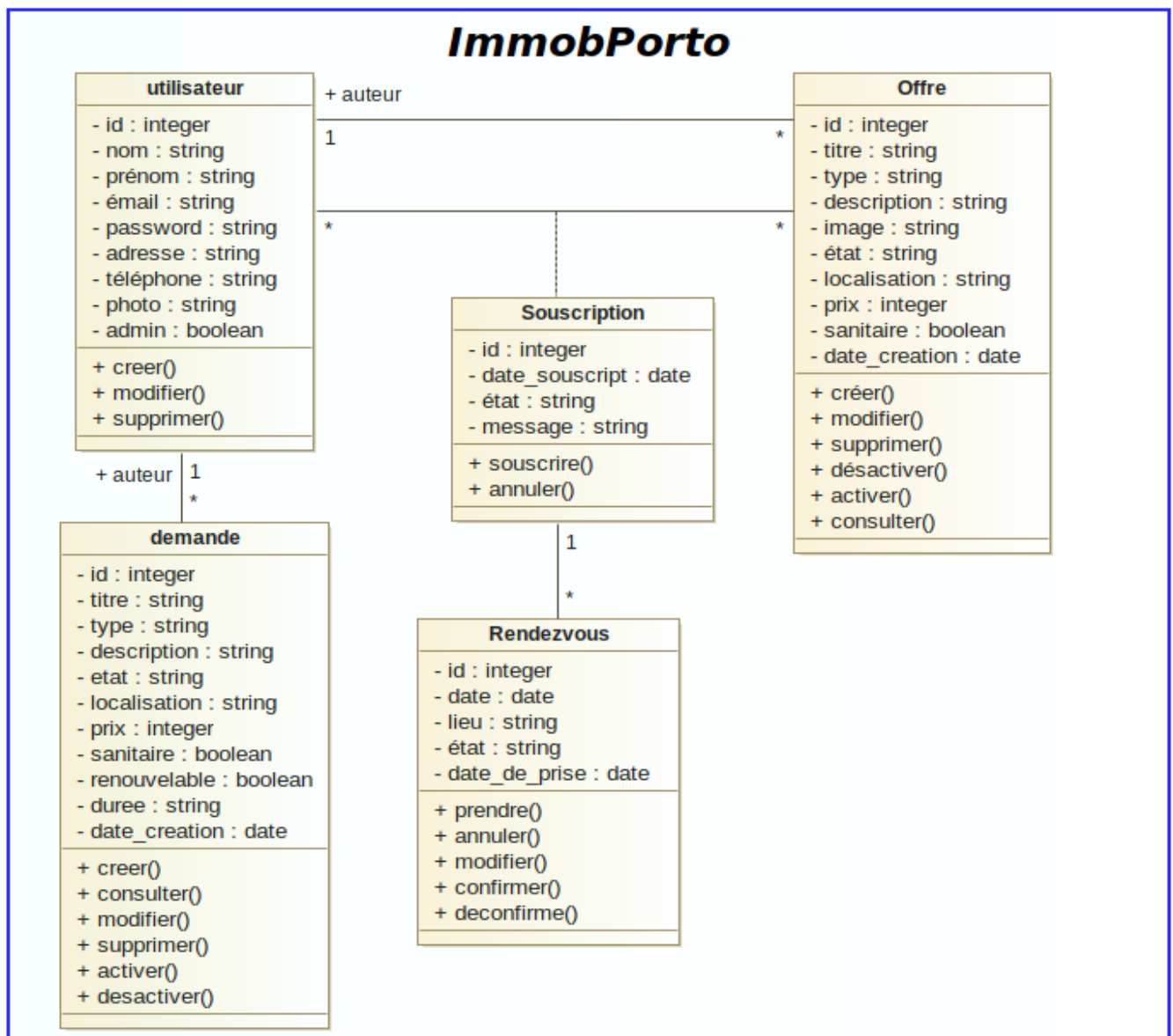


Figure 2: Diagramme des classes

0.6 Passage au relationnel

Le modèle relationnel doit refléter le diagramme de classes issu de l'analyse, et donc les éléments présents dans le diagramme des classes (classe, attribut, association, classe association) doivent se retrouver dans le modèle relationnel. En procédant par l'application des règles de passage du diagramme de classes au modèle relationnel, nous avons généré le modèle relationnel suivant :

- **utilisateur**(utilisateur-id, nom ,prenom, email, password, adress, telephone, photo, admin)
- **demande**(demande-id, titre, type, description, etat, localisation, prix, sanitaire, renouvelable, duree, date-creation, #utilisateur-id)

- **offre**(offre-id, titre, type, description, etat, localisation, prix, sanitaire, date-creation, #utilisateur-id)
- **souscription**(souscription-id, date-souscription, etat, message, #utilisateur-id, #offre-id)
- **rendezvous**(rendezvous-id, date, lieu, etat, date-de-prise, #souscription-id)

Partie IV

Réalisation

0.7 Environnement et outils de développement de notre application

Pour pouvoir bien réaliser notre application nous avons opté pour quelque outils que nous allons définir ci-dessous.

1. HTML

HTML est un langage de balisage servant à écrire des pages structurées pour le World Wide Web(www).

2. CSS(CASCADING STYLE SHEETS)

Les feuilles de style dites CSS permettent de gérer la présentation d'une page web. CSS permet de créer des règles de styles et de mise en pages. Ces règles portent sur le positionnement des éléments, l'alignement, les polices de caractères, les couleurs, les marges et espacement, les bordures, les images de fond, etc.

3. JavaScript

C'est un langage de programmation qui offre la possibilité d'implémenter des traitements élaborés dans des pages web, et permet d'apporter des améliorations au langage HTML en permettant d'exécuter des commandes côté client

4. PHP(Hypertexte Préprocesseur)

PHP est un langage de scripts libre principalement utilisé pour produire des pages web dynamiques.

Nous avons aussi utilisé des frameworks qui sont :

1. Laravel pour le backend

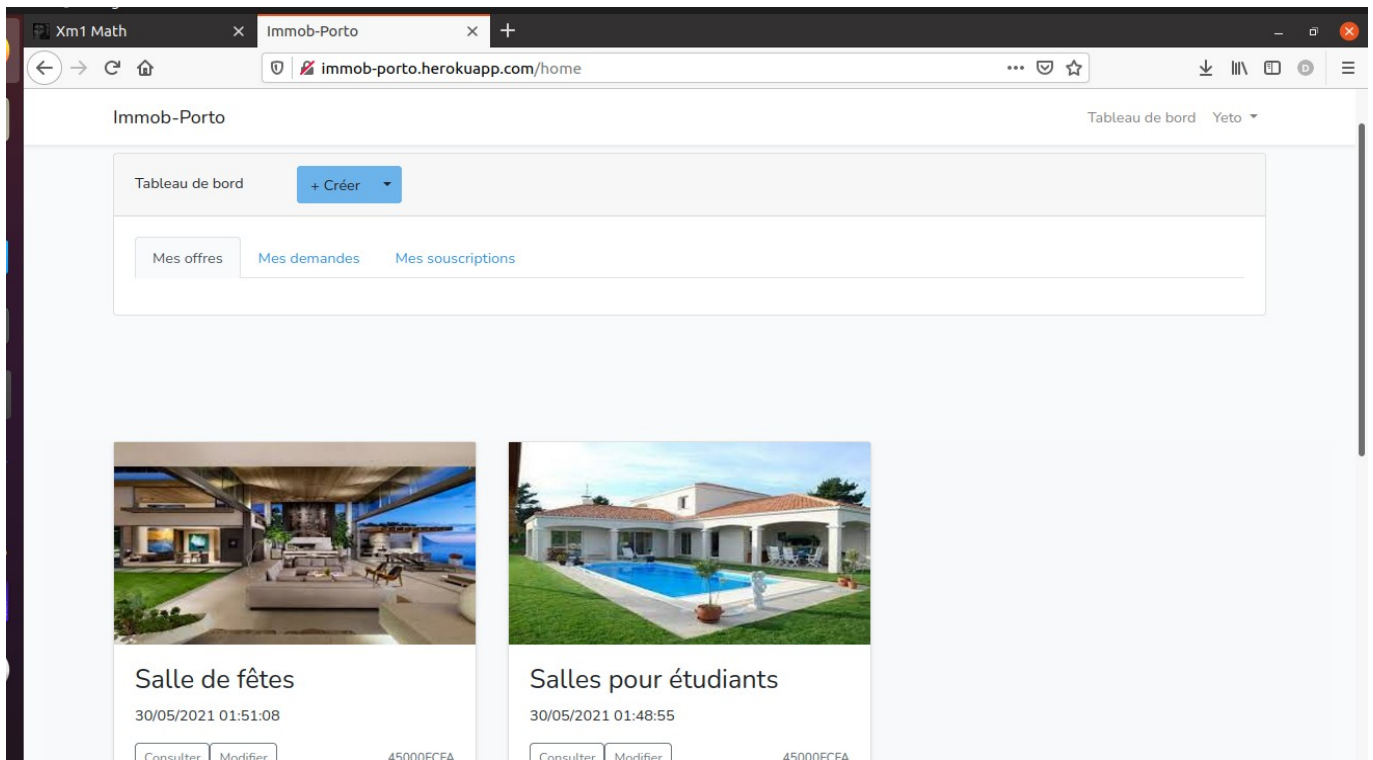
Laravel est un framework backend d'application Web basé sur **PHP** avec une syntaxe expressive et élégante. Il s'efforce de fournir une expérience de développement incroyable, tout en fournissant des **fonctionnalités puissantes**. Il utilise le modèle **MVC**.

2. Bootstrap pour le frontend

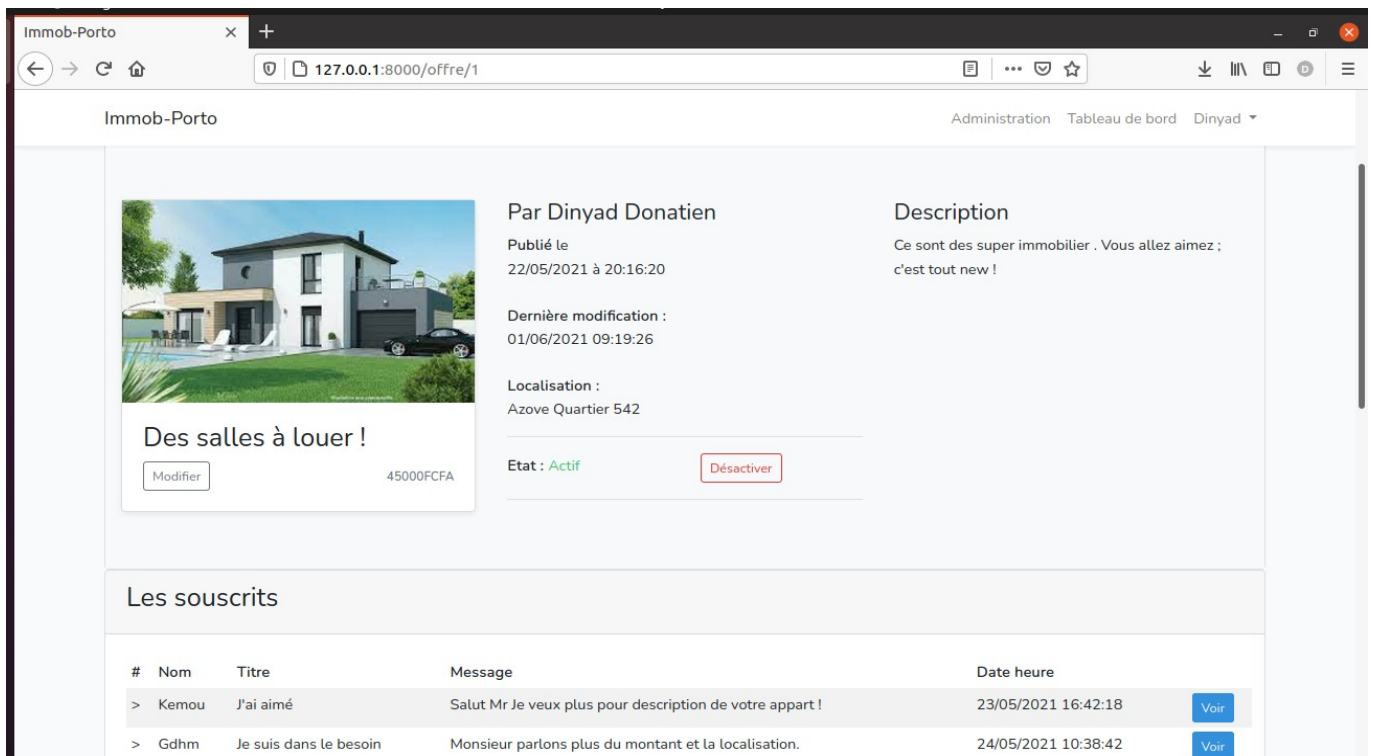
Bootstrap est un framework CSS et JavaScript qui organise et gère la mise en page d'un site web. Bien que Bootstrap fasse gagner du temps aux développeurs en leur évitant de devoir gérer les modèles de façon répétitive, son objectif premier est de créer des sites réactifs. Il permet à l'interface utilisateur d'un site web de fonctionner de manière optimale sur toutes les tailles d'écran, que ce soit sur des téléphones à petit écran ou des ordinateurs de bureau à grand écran. Les développeurs n'ont donc pas besoin de créer des sites spécifiques à un appareil et de limiter leur public.

0.8 Présentation de quelques interfaces de l'application

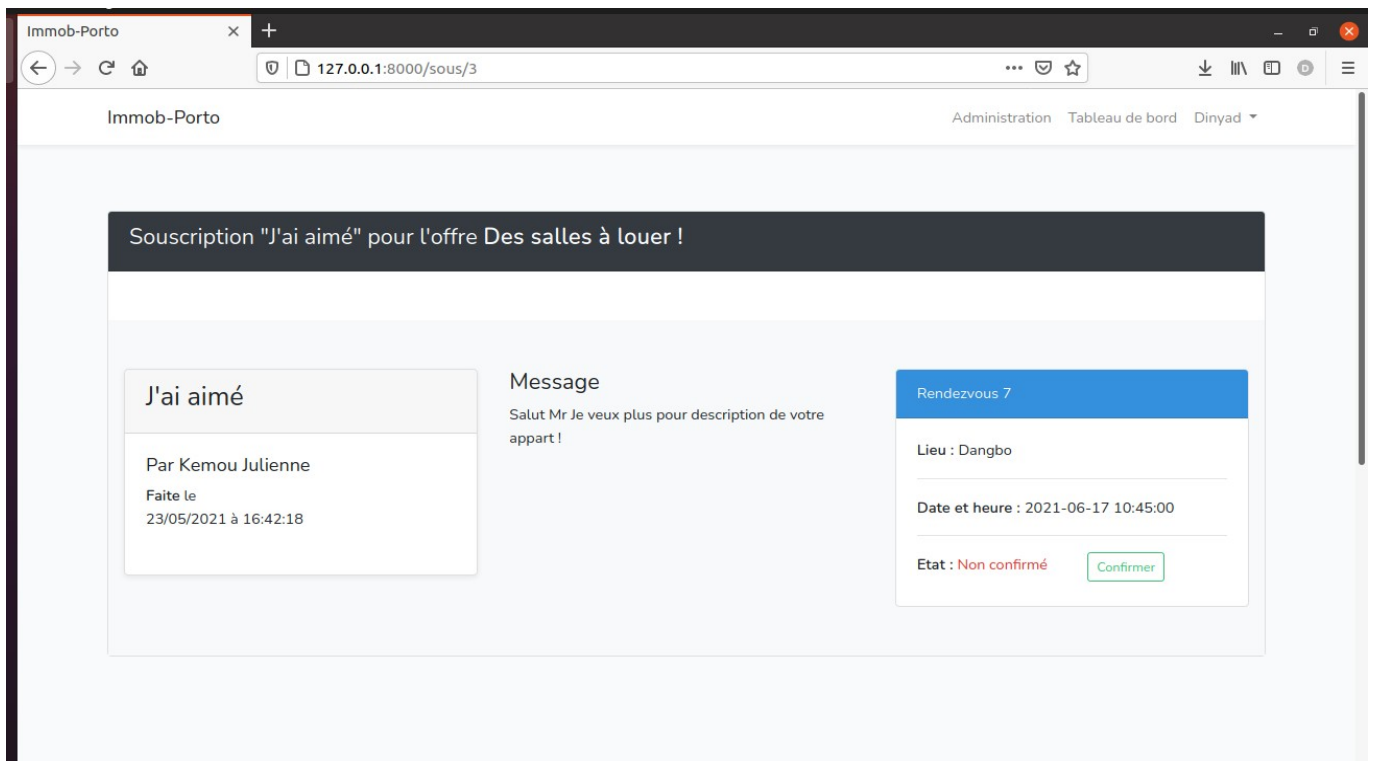
Tableau de bord



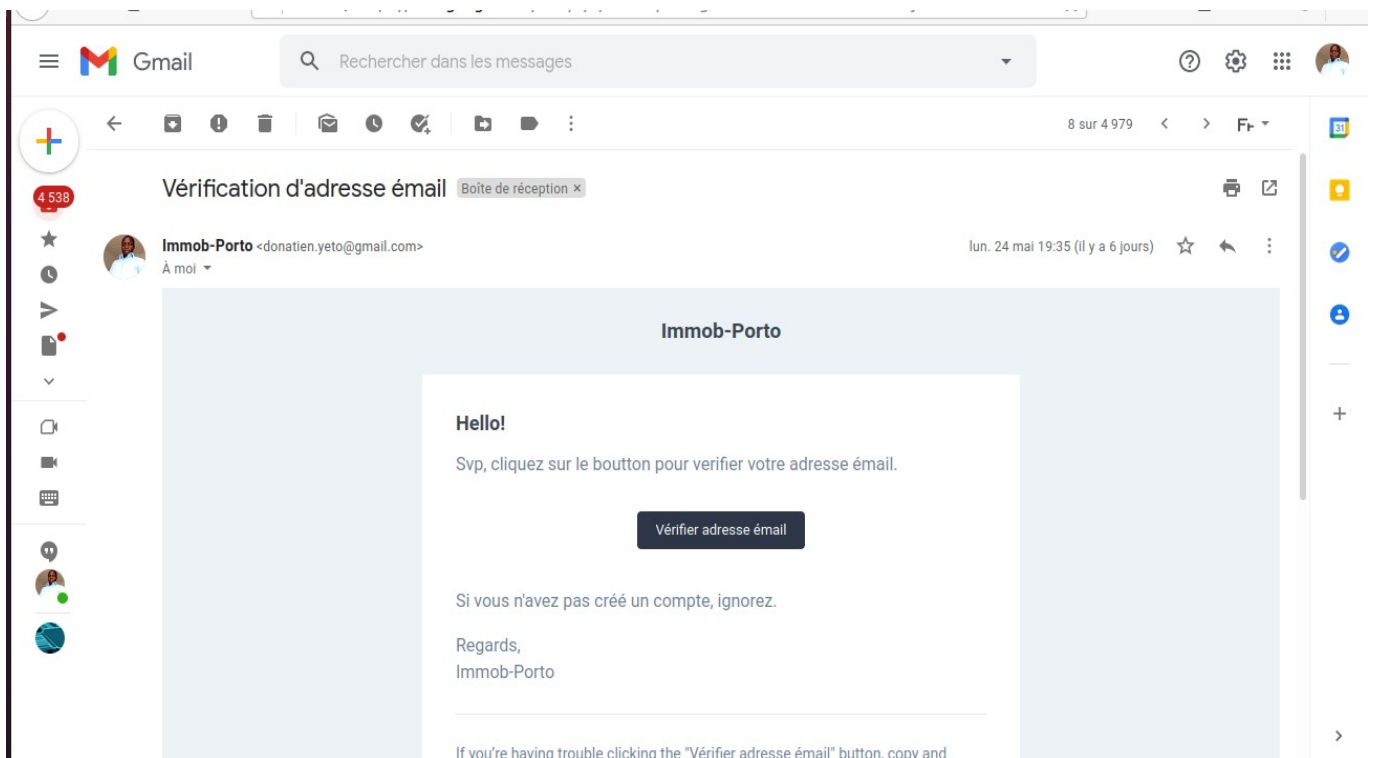
Consultation d'un offre par son auteur



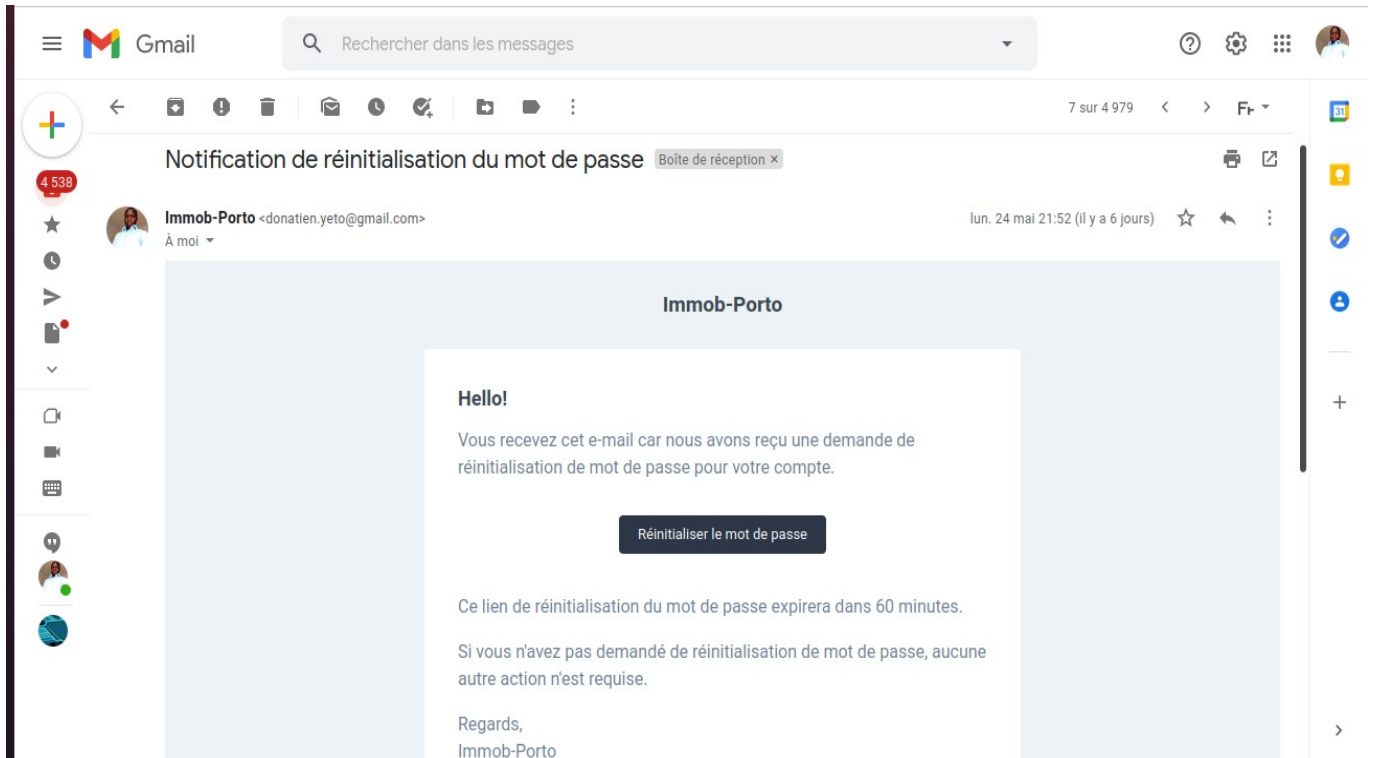
Consultation d'une souscription



Email de vérification d'adresse email



Notification de réinitialisation du mot de passe



Partie V

Hébergement

0.9 Notion d'hébergement

L'hébergement est un service visant à rendre un site ou une application web accessible sur Internet. Afin que les pages soient visibles par tout le monde, il faut qu'elles soient stockées sur un ordinateur connecté en permanence à l'internet (serveur).

Un hébergeur est comme une société spécialisée dans l'hébergement de sites web sur des serveurs, sa principale activité sur internet consiste à installer des serveurs, à les sécuriser, à les tenir à jour en installant les mises à jour de sécurité pour éviter les attaques malveillantes et à les réparer en cas de panne.

Il existe plusieurs types d'hébergements dont :

- **Hébergement gratuit** : ce type d'hébergement offre un service basique, avec un espace de stockage assez limité, avec une fiabilité qui n'est pas toujours optimale.
- **Hébergement payant** : ce type d'hébergement propose un service de qualité, avec de bonnes capacités de stockage ainsi qu'une grande fiabilité. Le support sera également plus disponible et plus réactif en cas de problème et aidera avec des réponses personnalisées.

0.10 Hébergement

Pour ce projet nous avons opté pour l'hébergeur **Heroku**. Heroku est une entreprise créant des logiciels pour serveur qui permettent le déploiement d'applications web.

Nous nous sommes limités à **l'hébergement gratuit**.

L'application est donc hébergée et accessible à l'adresse :
immob-porto.herokuapp.com

Partie VI

Conclusion

0.11 Conclusion

L'application réalisée permet d'assurer plusieurs fonctionnalités de base à savoir publication des annonces d'offres ou de demandes de vente ou d'achat de logement, souscription à une annonce, prise de rendez-vous pour la visite du terrain...

Le projet qui nous a été attribué est très instructif sur le plan pédagogique et très intéressant sur le plan technologique et développement. Nous en tant qu'étudiants en fin de cycle, ils nous a permis de :

- Accroître nos connaissances.
- Initier aux différentes technologies de développement (css, php. . .).
- Améliorer nos compétence dans la programmation orienté objet.

List of Figures

1	Diagramme des cas d'utilisations	10
2	Diagramme des classes	14