

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A. Mira de Bejaïa
Faculté des Sciences Exactes
Département d'Informatique



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Génie Logiciel

Thème

Conception et réalisation d'une application web pour la publication d'annonces immobilières

Réalisé par

M^r. MOKRANE Lounis

Évalué par le jury :

M ^{me} Leila CHELOUAH	M.C.B, université de Bejaïa	Examinatrice
M ^{me} Nawel YESSAD	M.C.B, université de Bejaïa	Examinatrice
M ^{me} Souaad LAHLAH	M.C.B, université de Bejaïa	Encadrante

Promotion 2020 – 2021

Remerciements

Mes remerciements s'adressent en premier lieu à ma promotrice M^{me} Souaad LAHLAH, pour avoir accepté de diriger ce travail. Son suivi, ses conseils et orientations.

Je tiens également à remercier sincèrement les membres du jury ; M^{me} Leila CHELOUAH et M^{me} Nawel YESSAD; qui nous font l'honneur d'évaluer ce travail.

Je remercie également les proches famille et amis pour leurs soutiens et leurs conseils.

Dans l'impossibilité de cité tous les noms, que tous ce qui mon soutenu de près ou de loin trouvent ici ma sincère gratitude.

Table de matières

Liste des figures	v
Liste des tableaux	vii
Liste des abréviations	viii
Introduction générale	1
Chapitre 1 : Etude préalable	3
1. Introduction	3
2. Définition et caractéristiques d'une application web	3
3. Etude et critique de l'existant	4
3.1. Analyse via Similareweb	4
3.1.1. Cas Ouedkniss	4
3.1.2. Cas de Lkeria	7
3.2. Critique de l'existant	9
3.2.1. Cas de ouedkinss	9
3.2.2. Cas de Lkeria	11
4. Problématique	12
5. Présentation de notre application	12
5.1. Les besoins fonctionnels et non fonctionnels de l'application	13
5.1.1. Les besoins fonctionnels	13
5.1.2. Les besoins non-fonctionnels	14
6. Conclusion	15
Chapitre 2 : Etude conceptuelle	16
1. Introduction	16
2. Processus de développement	16
2.1. Un processus simplifié pour les applications web	16
2.2. Les principes fondamentaux du Processus Unifié (PU)	16
2.2.1. Les phases et les disciplines du PU	17
3. Choix du langage de conception	17
3.1. C'est quoi le UML ?	18

3.1.1.	Six diagrammes structurels	18
3.1.2.	Sept diagrammes comportementaux	18
3.2.	Définition détaillée de quelques diagrammes	19
3.2.1.	Diagramme de classes	19
3.2.2.	Diagramme de cas d'utilisation	20
3.2.3.	Diagramme de séquence	23
4.	Identification des acteurs	24
5.	Identification des cas d'utilisation	24
6.	Diagramme de cas d'utilisation	25
6.1.	Diagramme de cas d'utilisation de « Visiteur »	25
6.2.	Diagramme de cas d'utilisation de « Membre »	26
6.3.	Diagramme de cas d'utilisation de « Administrateur »	27
7.	Description des cas d'utilisation	27
7.1.	Cas d'utilisation « S'inscrire »	27
7.2.	Cas d'utilisation « Faire une recherche »	28
7.3.	Cas d'utilisation « S'authentifier »	28
7.4.	Cas d'utilisation « Modifier son compte »	29
7.5.	Cas d'utilisation « Publier une annonce »	30
7.6.	Cas d'utilisation « Signaler une annonce »	30
7.7.	Cas d'utilisation « Modifier son annonce »	31
7.8.	Cas d'utilisation « Supprimer son annonce »	31
7.9.	Cas d'utilisation « Supprimer un membre »	32
8.	Diagramme de séquence	32
8.1.	Diagramme de séquence « S'inscrire »	32
8.2.	Diagramme de séquence « Faire une recherche »	34
8.3.	Diagramme de séquence « S'authentifier »	35
8.4.	Diagramme de séquence « Modifier son compte »	36
8.5.	Diagramme de séquence « Publier une annonce »	37
8.6.	Diagramme de séquence « Signaler une annonce »	39
8.7.	Diagramme de séquence « Modifier son annonce »	40
8.8.	Diagramme de séquence « Supprimer son annonce »:	41
8.9.	Diagramme de séquence « Supprimer un membre »	43

9. Diagramme de classe	44
10. Dictionnaire de données	45
11. Les règles de passage de diagramme de classe vers le schéma de base de données	46
12. Schéma relationnel	47
13. Conclusion	48
Chapitre 3 : Réalisation et présentation de l'application	49
1. Introduction	49
2. Outils de développement utilisés	49
2.1. Frameworks	49
2.1.1. Laravel	49
2.1.1.1. L'architecture MVC	50
2.1.2. Bootstrap	50
2.2. Persistance de données	51
2.2.1. Eloquent ORM	51
2.2.2. MySQL	51
2.3. Logiciels	52
2.3.1. Visual Studio	52
2.3.2. Adobe Illustrator	52
2.3.3. Draw.io	52
2.3.4. POSTMAN	53
2.4. Langages et techniques	54
2.4.1. PHP	54
2.4.2. JavaScript	54
2.4.3. HTML	54
2.4.4. CSS	55
2.4.5. AJAX	55
3. Présentation des interfaces	55
3.1. Page d'accueil	56
3.2. Page d'inscription	56
3.3. Page Se connecter	57

3.4.	Page consulter les annonces	60
3.5.	Page publier une annonce	61
3.6.	Espace membre montrant l'arrivé de notifications	62
3.7.	Page Mes annonces	63
3.8.	Formulaire de modification d'une annonce	64
4.	Conclusion	65
	Conclusion générale et perspectives	66
	Bibliographie	68
	Résumé	71

Liste des figures

Figure 1. 1 : Résultat d'analyse du site ouedkniss.com sur similarweb [11].	5
Figure 1. 2 : Résultat d'analyse du site ouedkniss.com sur similarweb [11].	6
Figure 1. 3 : Résultat d'analyse du site ouedkniss.com sur similarweb [11].	6
Figure 1. 4: Résultat de l'analyse de site lkeria.com sur similarweb [12].	7
Figure 1. 5 : Résultat d'analyse du site lkeria..com sur similarweb [12].	8
Figure 1. 6 : Résultat d'analyse du site lkeria..com sur similarweb [12].	8
Figure 1. 7 : Capture d'écran du site ouedkniss.com [3].	10
Figure 1. 8 : Capture d'écran du site ouedkniss.com [13].	10
Figure 1. 9 : Capture du site lkeria.com [4].	11
Figure 1. 10 : Logo de notre application	12
Figure 1. 11 : Palette de couleurs utilisé pour notre logo.	13
Figure 2. 1 : Les diagrammes UML [16]	19
Figure 2. 2 : Représentation graphique d'une classe	20
Figure 2. 3 : Représentation graphique d'un acteur	21
Figure 2. 4 : Représentation graphique d'un cas d'utilisation	21
Figure 2. 5 : Représentation graphique de relation d'inclusion «include» dans le diagramme de cas d'utilisation	22
Figure 2. 6 : Représentation graphique de relation d'extension «extend» dans le diagramme de cas d'utilisation	22
Figure 2. 7 : Représentation graphique de relation généralisation	22
Figure 2. 8 : Représentation graphique d'un message dans le diagramme de séquence	23
Figure 2. 9 : Représentation graphique d'un objet dans le diagramme de séquence	24
Figure 2. 10 : Diagramme de cas d'utilisation associé à l'acteur « Visiteur ».	25
Figure 2. 11 : Diagramme de cas d'utilisation associé à l'acteur « Membre ».	26
Figure 2. 12 : Diagramme de cas d'utilisation associé à l'acteur « Administrateur ».	27
Figure 2. 13 : Diagramme de séquence du cas d'utilisation «S'inscrire ».	33
Figure 2. 14 : Diagramme de séquence du cas d'utilisation «Faire une recherche ».	34
Figure 2. 15 : Diagramme de séquence du cas d'utilisation «S'authentifier ».	35
Figure 2. 16 : Diagramme de séquence du cas d'utilisation «Modifier son compte ».	37
Figure 2. 17 : Diagramme de séquence du cas d'utilisation «Publier une annonce ».	38
Figure 2. 18 : Diagramme de séquence du cas d'utilisation «Signaler une annonce ».	39

Figure 2. 19 : Diagramme de séquence du cas d'utilisation « Modifier son annonce ».	40
Figure 2. 20 : Diagramme de séquence du cas d'utilisation «Supprimer son annonce ».	42
Figure 2. 21 : Diagramme de séquence du cas d'utilisation «Signaler une annonce ».	44
Figure 2. 22 : Diagramme de classe	45
Figure 3. 1 : Logo de Framework Laravel [21].	49
Figure 3. 2 : Logo de la bibliothèque Bootstrap [23]	50
Figure 3. 3 : Schéma démentant l'emplacement et le rôle de Eloquent ORM [32]	51
Figure 3. 4 : Logo de SGBD MYSQL [25]	51
Figure 3. 5 : Logo de l'éditeur de code source Visual Studio Code[27]	52
Figure 3. 6 : Logo de logiciel Adobe Illustrator [28]	52
Figure 3. 7 : Logo de l'application Draw.io [30]	53
Figure 3. 8 : Logo de logiciel POSTMAN [32]	53
Figure 3. 9 : Logo de langage PHP [34]	54
Figure 3. 10 : Logo de langage JAVASCRIPT [36]	54
Figure 3. 11 : Logo de langage HTML [38]	55
Figure 3. 12 : Logo de langage CSS [40]	55
Figure 3. 13 : Logo de langage AJAX [42]	55
Figure 3. 14 : Interface page d'accueil.	56
Figure 3. 15: Interface d'inscription.	57
Figure 3. 16 : Interface Se connecter.	58
Figure 3. 17 : Interface espace membre	59
Figure 3. 18 : Interface espace administrateur.	60
Figure 3. 19 : Interface consulter les annonces	61
Figure 3. 20 : Formulaire de publication d'une nouvelle annonce.	62
Figure 3. 21 : Espace membre avec la présence de nouvelles notifications.	63
Figure 3. 22 : Page Mes annonces.	64
Figure 3. 23 : Formulaire de modification d'une annonce.	65

Liste des tableaux

Tableau 2. 1 :	Identification des cas d'utilisation.....	25
Tableau 2. 2 :	Description du cas d'utilisation « S'inscrire ».....	28
Tableau 2. 3 :	Description du cas d'utilisation « Faire une recherche ».....	28
Tableau 2. 4 :	Description du cas d'utilisation « S'authentifier ».....	29
Tableau 2. 5 :	Description du cas d'utilisation «Modifier son compte».....	29
Tableau 2. 6 :	Description du cas d'utilisation «Publier une annonce».....	30
Tableau 2. 7 :	Description du cas d'utilisation «Signaler une annonce».....	30
Tableau 2. 8 :	Description du cas d'utilisation «Modifier son annonce».....	31
Tableau 2. 9 :	Description du cas d'utilisation «Supprimer son annonce».....	31
Tableau 2. 10 :	Description du cas d'utilisation «Supprimer un membre».....	32
Tableau 2. 11 :	Dictionnaire de données.....	46

Liste des abréviations

UP	: Unified Process
UML	: Unified Modeling Language
MVC	: Model View Controller
ORM	: Object-Relational Mapping
HTML	: HyperText Markup Language
CSS	: Cascading Style Sheets
AI	: Adobe Illustrator
PHP	: HyperText Preprocessor
JS	: JavaScript
AJAX	: Asynchronous JavaScript and XML
HTTP	: HyperText Transfer Protocole
URL	: Uniform Resource Locator

Introduction générale

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

L'informatique propose des solutions nombreuses, et une d'elles sont les applications web qui sont très répandues sur le marché technologique de notre temps.

Avec les recommandations des gens de l'immobilier en Algérie, on trouve que le marché Algérien est toujours dans l'état de demande de nouvelle solution malgré la présence de certaines applications de cette tendance.

Dans cette optique, ce projet est dédié à mettre en place une nouvelle application de publication d'annonces immobilières en Algérie.

Notre application a pour but d'offrir et de répondre à plus de besoins. Elle va disposer d'une fonctionnalité dont les autres ne disposent pas, cette dernière est un système de notifications à temps réel. C'est-à-dire, que lors de l'arrivée d'une nouvelle annonce sur notre application, le système de notification se charge de notifier tous les membres de l'application de cette nouvelle.

Pour la réalisation de ce projet, nous avons structuré l'ensemble de notre travail en trois chapitres :

Le premier chapitre est consacré à l'étude préalable où nous avons réalisé une étude du marché Algérien en ce qui concerne les applications web dédiées à la publication d'annonces immobilières. Après cette étude, nous avons déduit la problématique et nous avons présenté notre application coté nom et logo de marque ainsi que les besoins fonctionnels et non fonctionnels.

Le deuxième chapitre décrit la partie conception où nous avons présenté le processus de développement PU puis le langage de modélisation UML. Après ça, nous avons identifié les acteurs de l'application. Ensuite, nous avons cité les cas d'utilisations pour chaque acteur afin qu'on modélise les différents diagrammes de cas d'utilisations et de séquences, et vers la fin le diagramme de classe ainsi que les règles de passage de ce dernier vers schéma de la base de données.

Introduction générale

Le troisième chapitre présente la partie réalisation de notre application. Nous avons exposé les technologies que nous avons utilisées pour la réalisation et le développement de cette application, et présenté les différentes interfaces de l'application.

Enfin, nous concluons ce travail avec une conclusion générale et quelques perspectives

Chapitre 1 : Etude préalable

1. Introduction

Dans ce chapitre, nous allons commencer tout d'abord par définir une application web et ses caractéristiques. Puis, nous présenterons notre étude et critique de l'existant sur les applications web dédiées à la publication d'annonces immobilières en Algérie tout en s'aidant du site similarweb [1]. Ensuite, nous exposerons la problématique à traiter. Enfin, nous présenterons notre application ainsi que les besoins fonctionnels et non fonctionnels.

2. Définition et caractéristiques d'une application web

Une application Web est un logiciel applicatif manipulable grâce à un navigateur Web (Google Chrome, Mozilla Firefox, Safari). De la même manière que les sites Web, une application Web est généralement placée sur un serveur et se manipule en interaction sur le navigateur. Parmi ses caractéristiques [2]:

- Elle est disponible partout, cela veut dire qu'il vous suffit d'utiliser un navigateur internet pour y accéder sur n'importe quelle machine (tablette, téléphone, ordinateur).
- Vous pouvez vous connecter depuis n'importe quel poste et retrouver toutes vos données synchronisées. Concrètement, on suppose qu'il y'a deux utilisateurs A et B, le A accède à l'application depuis sa machine et fera des modifications, suppressions ou ajouts. Quand le B accède depuis sa machine, il trouve les mêmes informations disponibles sur la machine de A.
- Il n'y a pas d'installation préalable à réaliser sur son poste de travail ou sur sa tablette (contrairement à un logiciel ou une application mobile).
- Les sauvegardes sont centralisées sur le serveur et il n'y a plus besoin de réaliser des mises à jour sur tous les postes de travail.
- En cas de remplacement de poste, la mise en place sera beaucoup plus simple et rapide car elle ne nécessite pas une installation.

3. Etude et critique de l'existant

Avant de se lancer dans la réalisation de tout projet, il est indispensable de bien étudier et analyser des projets similaires pour profiter de leurs avantages, essayer de réduire leurs inconvénients, et proposer des solutions pour combler les manques ou des solutions complètement différentes avec de nouveaux concepts et plus d'originalités.

D'après nos recherches sur les applications web de publication d'annonces immobilières existantes en Algérie nous avons trouvé : Ouedkniss [3], Elkeria [4], Nbatou [5], Darna [6], Dardjadida [7], Essekena [8], Algeriahome [9] et Algerieannonce [10].

Pour une analyse efficace, nous avons utilisé l'un des sites d'analyse en ligne le plus utilisé qui est SimilarWeb [1]. Ce dernier, nous permet de connaître le classement d'une application dans un pays et à l'échelle mondiale ainsi que le classement de ces concurrents, et le nombre de visiteurs par mois.

Après l'analyse via SimilarWeb, nous avons spécifié les applications que nous allons prendre en considération dans notre étude. En termes des meilleurs résultats, les applications sont lkeria et ouedkniss.

3.1. Analyse via Similareweb

SimilarWeb est un outil d'analyse en ligne, il permet une estimation du trafic et donc une meilleure compréhension du marché et des tendances. Ci-dessous les résultats d'analyse pour les deux applications ouedkniss et lkeria :

3.1.1. Cas Ouedkniss

Les figures 1.1, 1.2 et 1.3 montrent des captures d'écran des résultats de l'analyse du site ouedkniss via le site Similarweb.

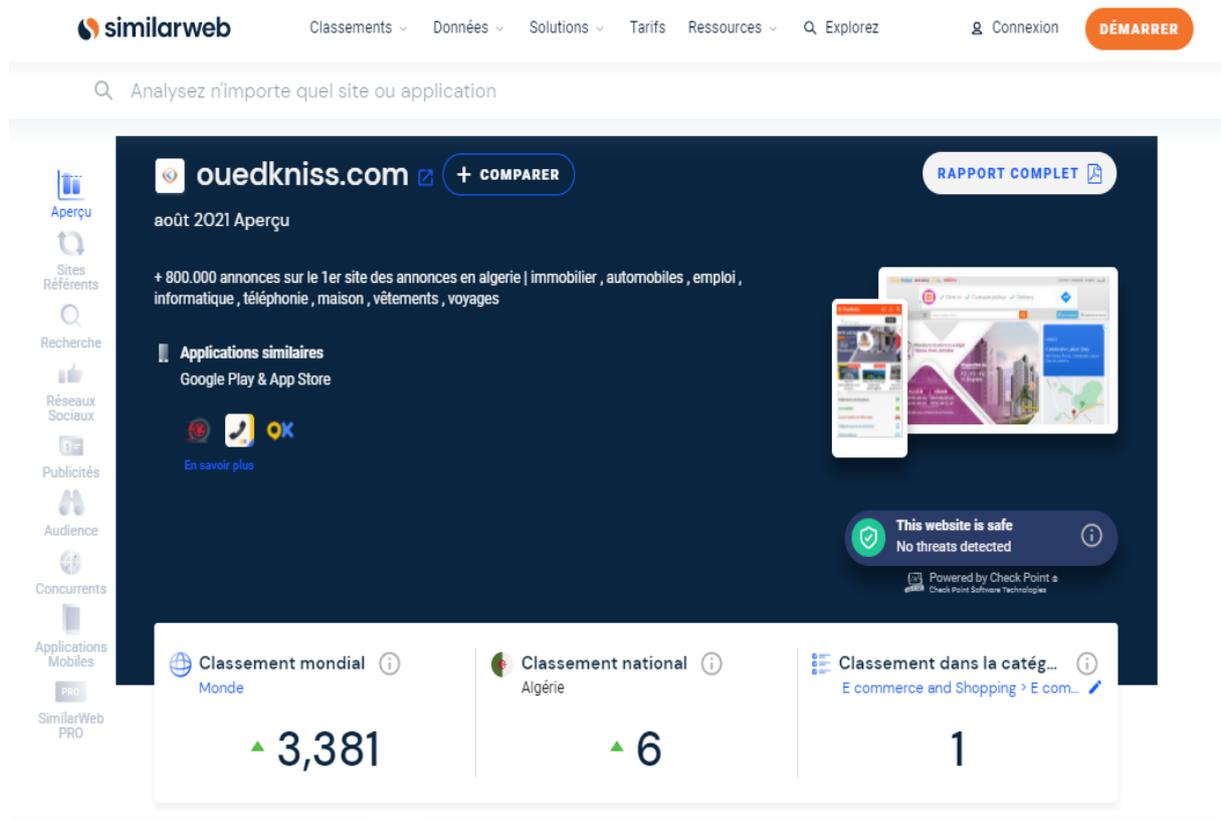


Figure 1. 1 : Résultat d’analyse du site ouedkniss.com sur similarweb [11].

D’après l’analyse via Similarweb du site Ouedkniss, nous avons obtenu les résultats suivants :

- C’est la première plateforme dans sa catégorie.
- C’est la 6^e plateforme au niveau national.
- Elle est classée 3.381 au niveau international.
- Le nombre de visites était de 12.93M (1M=1000 000visites) pendant le mois d’août 2021.
- Le pourcentage de visiteurs en Algérie pour l’application était de 95.55% en mois d’août 2021.

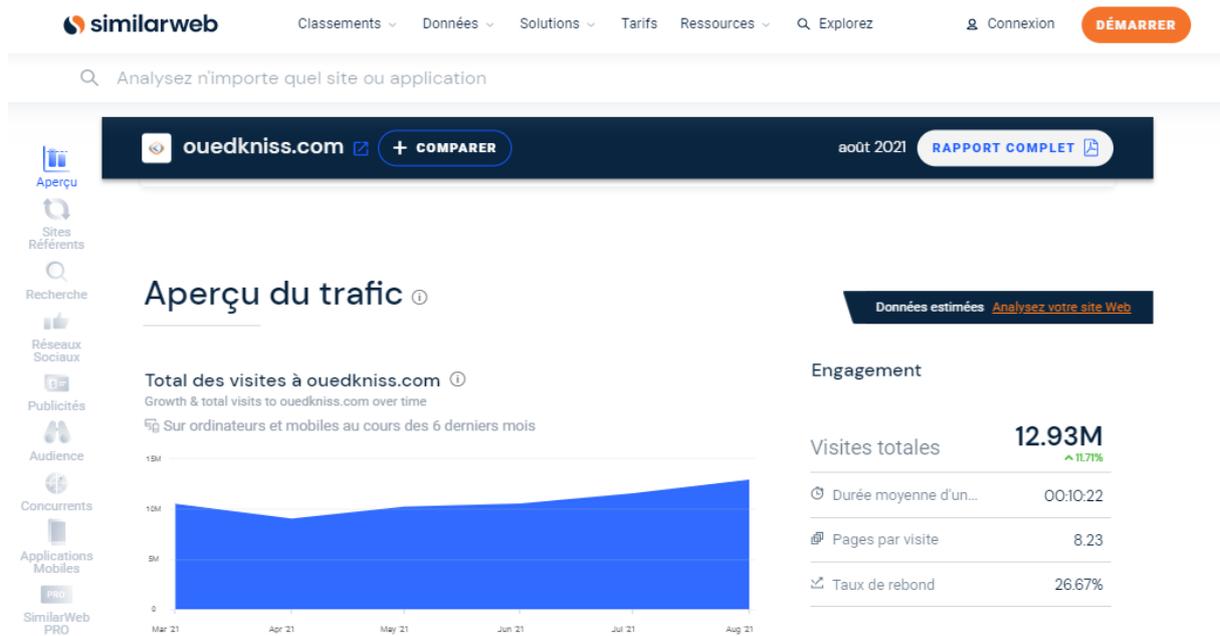


Figure 1. 2 : Résultat d’analyse du site ouedkniss.com sur similarweb [11].



Figure 1.3 : Résultat d’analyse du site ouedkniss.com sur similarweb [11].

3.1.2. Cas de Lkeria

Les figures 1.4, 1.5 et 1.6 montrent des captures d'écran des résultats de l'analyse du site Lkeria via le site Similarweb.

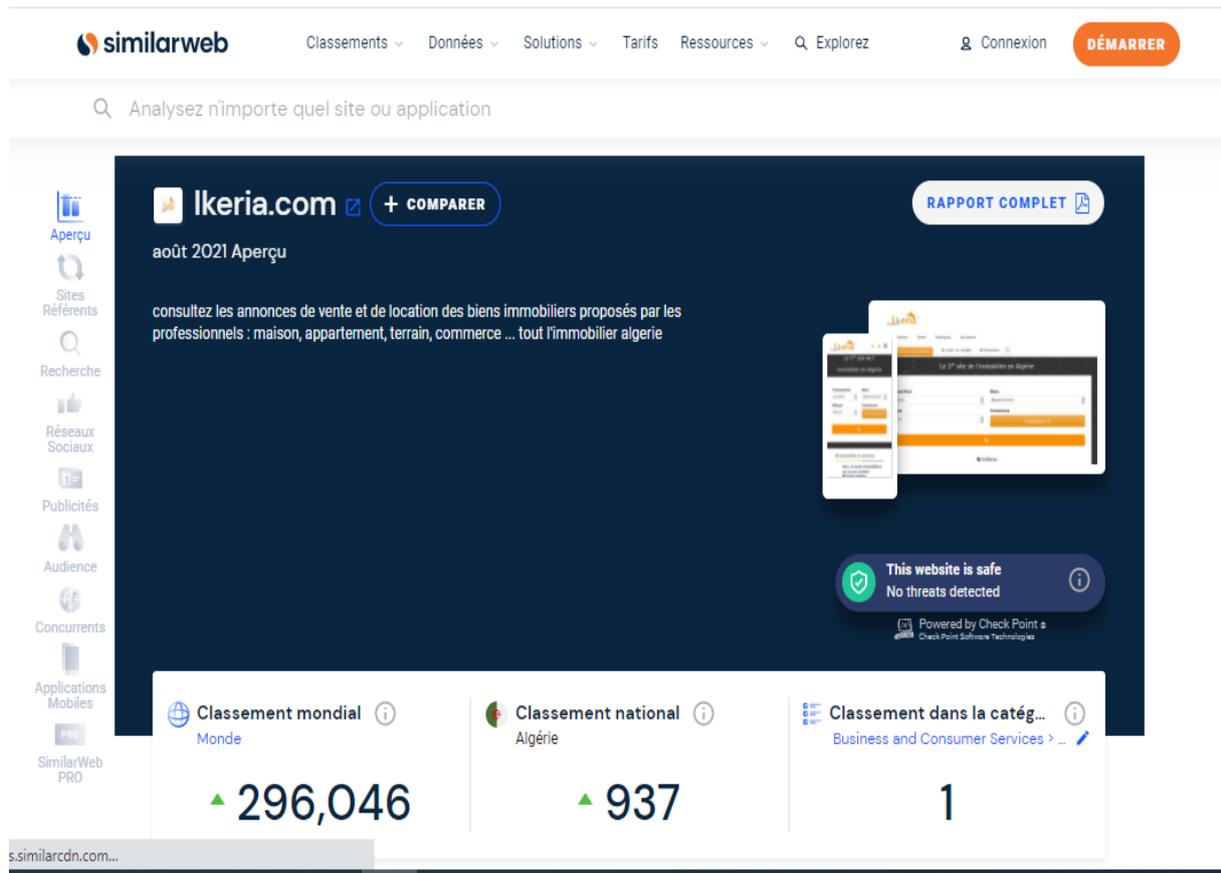


Figure 1.4 : Résultat de l'analyse de site lkeria.com sur similarweb [12].

L'analyse du site Lkeria via similarweb nous a montré les résultats suivants :

- C'est la première plateforme dans sa catégorie.
- Elle est classée 937 plateforme au niveau national.
- Elle est classée 296.046 au niveau international.
- Le nombre de visites était de 126.97K (1K=1000visites) pendant le mois d'août 2021.
- Le pourcentage de visiteurs en Algérie pour l'application était de 85.51% en mois d'août 2021.

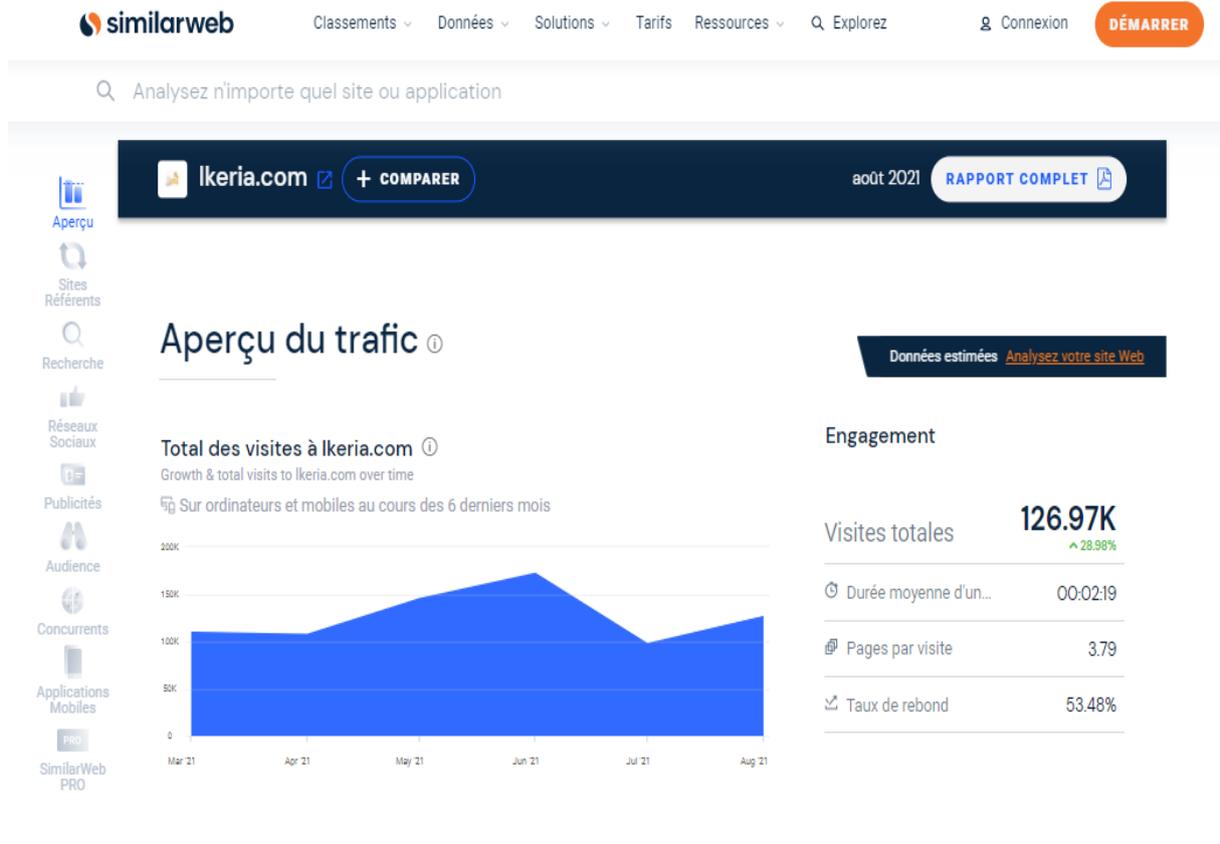


Figure 1.5 : Résultat d'analyse du site Ikeria..com sur similarweb [12].

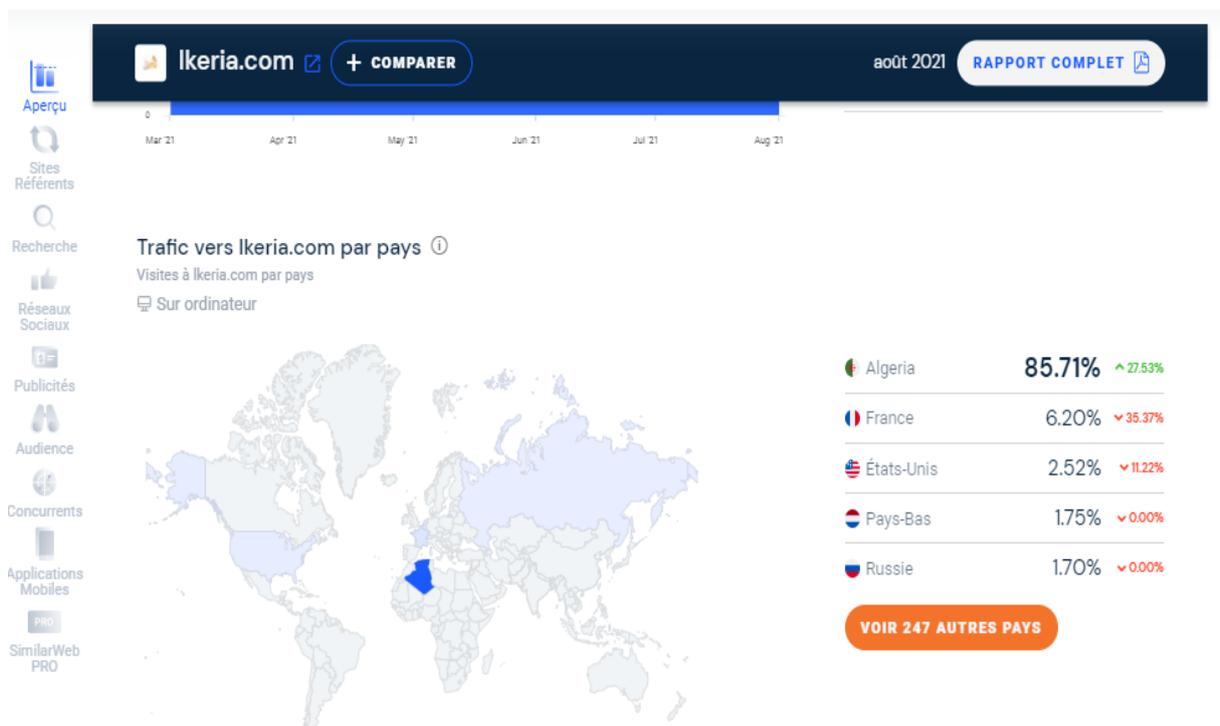


Figure 1.6 : Résultat d'analyse du site Ikeria..com sur similarweb [12].

3.2. Critique de l'existant

Afin de faire une critique des deux applications Ouedkniss et Lkeria, nous nous sommes inscrits afin de devenir des membres de ces applications, ce qui nous a permis d'avoir le droit d'accéder aux fonctionnalités de ces applications et de les essayer.

Après être inscrit sur les deux applications, et pu essayer les fonctionnalités disponibles sur ces dernières, nous avons déduis les points suivants :

3.2.1. Cas de ouedkinss

La figure 1.7 montre une capture d'écran du site ouedkniss, et la figure 1.8 montre une capture d'écran de la partie immobilier sur ouedkniss.

Points positifs :

- Première plateforme de petites annonces.
- Design responsive.
- Gratuité des annonces pour les particuliers.

Points négatifs :

- Beaucoup d'annonces publicitaires.
- Design obsolète depuis des années.
- Annonces payantes pour les entreprises.
- Pas d'annonces promotionnelles.
- Absence de notification pour les nouvelles annonces.

Etude préalable

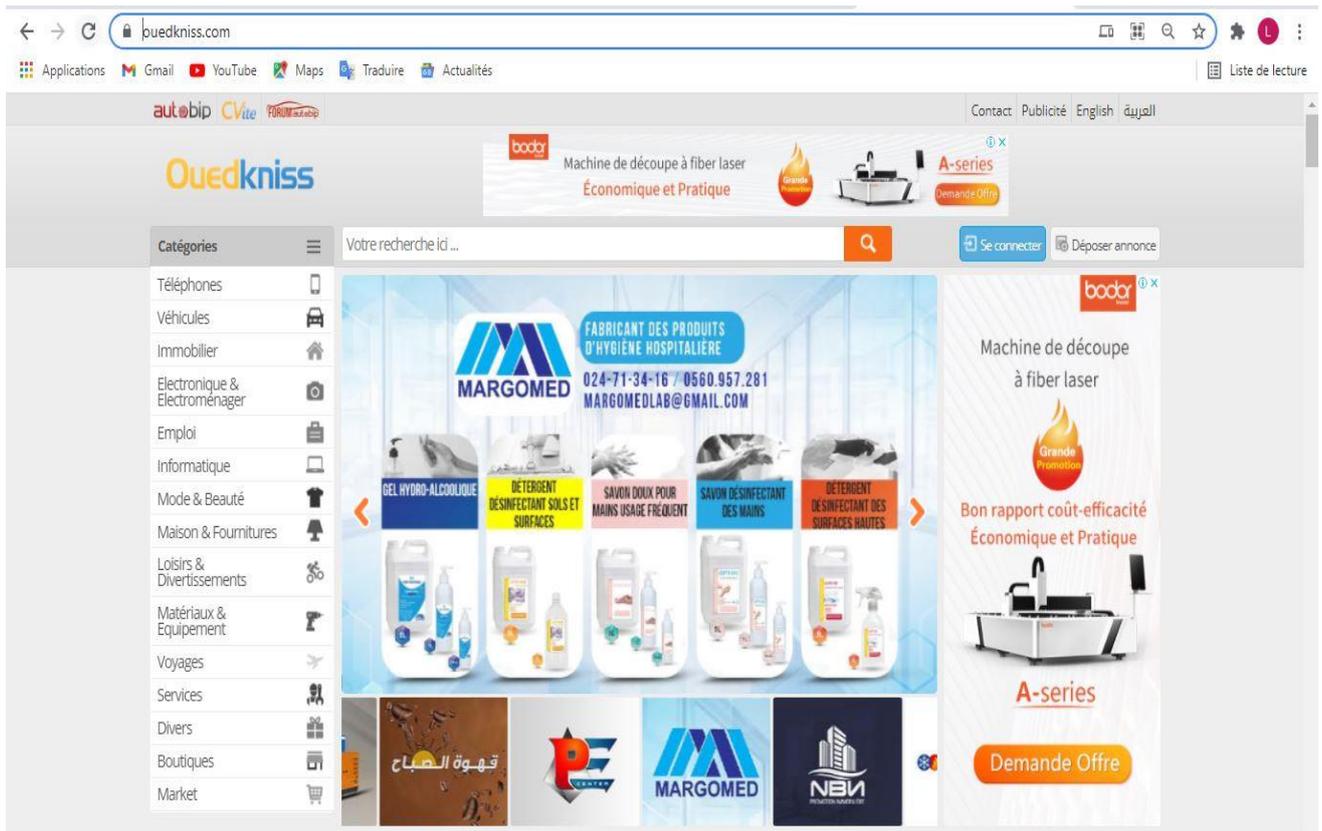


Figure 1.7 : Capture d'écran du site ouedkniss.com [3].

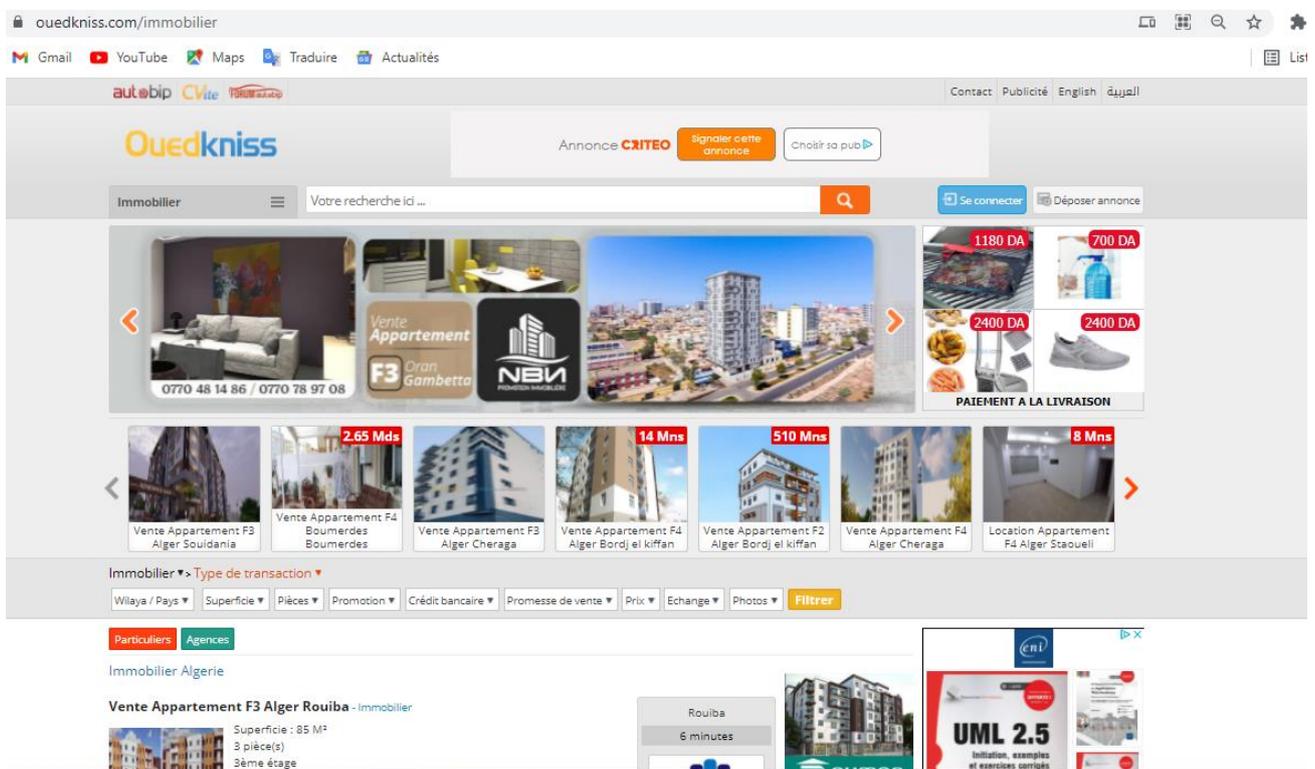


Figure 1.8 : Capture d'écran du site ouedkniss.com [13].

3.2.2. Cas de Lkeria

La figure 1.9 montre une capture d'écran du site Lkeria.

Points positifs :

- Premier site d'annonces immobilières en Algérie.
- Design responsive.
- Publicité promotionnelle.
- Pour le cas de particulier nous pouvons publier sans nous inscrire.

Points négatifs :

- Annonces payantes pour les particuliers et les entreprises.
- Absence de notification pour les nouvelles annonces.

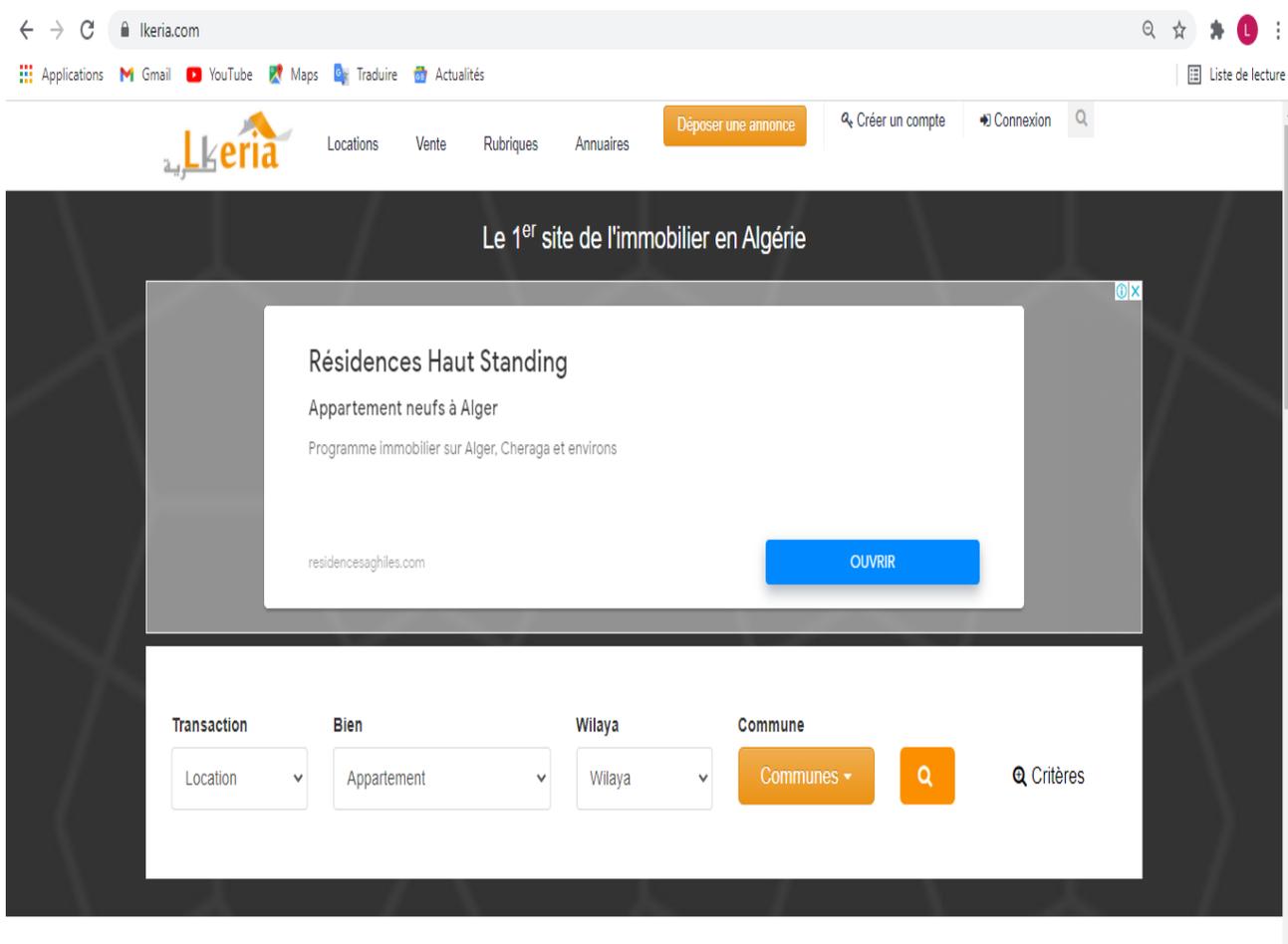


Figure 1.9 : Capture du site lkeria.com [4].

4. Problématique

A travers l'analyse de l'existant que nous avons réalisé, nous avons pu déduire que les applications existantes répondent uniquement à quelques besoins mais pas à tous. Par exemples, la publication d'annonce de bien immobilier sur l'application ouedkniss est placée dans une catégorie et n'est donc pas bien mise en évidence alors que c'est quelque chose de très recherchés par les utilisateurs et devrait être sous forme d'une application à part entière. Sur Ikeria, le gros point négatif c'est que les publications sont payantes ce qui peut faire renoncer quelqu'un à publier son annonce.

Dans ce travail de mémoire, nous proposons une application web pour la publication d'annonces immobilières tout en prenant en compte les avantages des applications similaires existantes et en rajoutant une fonctionnalité clé que nous avons jugé importante pour ce genre d'application et qui est la **notification** d'un bien. En effet, cette fonctionnalité permettra au système de mettre au courant les membres de l'application de l'arrivée de chaque nouvelle annonce sur l'application.

5. Présentation de notre application

Nous proposons une application web de publication d'annonces immobilières qui inclura, en plus des fonctionnalités de base, une fonctionnalité de notification qui informera les membres de l'application des nouveaux biens qui viennent d'être publiés. Le type d'annonces de bien immobilier qu'on pourrait retrouver sur notre application sont : vente, achat, location et échange d'un bien immobilier.

Nous avons choisi de nommer notre application **immo.store** que nous pouvons traduire par immobilier Storage, qui fait référence à un coin qui contient un stock de publications d'annonces immobilières.

La figure 1.10 montre le logo que nous avons créé pour notre application.



Figure 1.10 : Logo de notre application.

Etude préalable

Pour la réalisation de notre logo de marque, nous avons utilisé le logiciel de dessin vectorielle Adobe Illustrator [14]. Notre logo est une combinaison de deux termes : *Immobilier* Et *Storage*. Pour le premier mot '*Immobilier*', nous avons utilisé la partie '*immo*' et pour le deuxième mot '*Storage*,' nous avons utilisé la partie '*store*.' Ensuite, nous avons séparé les deux parties par un point pour former le mot **immo.store**.

Ensuite, nous avons pensé à transformer le point I et M en charpente pour donner un sens plus proche au domaine de l'immobilier.

Pour les couleurs du logo, nous avons choisi le cyan grisâtre très foncé et le rouge vif, comme sur la figure 1.11.



Figure 1.11 : Palette de couleurs utilisée pour notre logo.

- **Le rouge vif** E94E1B : cette couleur est inspirée de la couleur des tuiles qui fait partie de certaines catégories d'immeubles et de maisons.
- **Le cyan grisâtre très foncé** 4A4F50 : cette couleur est inspirée de la couleur du ciment qui est la base dans la construction de la majorité des immeubles et des maisons.

5.1. Les besoins fonctionnels et non fonctionnels de l'application

5.1.1. Les besoins fonctionnels

Les besoins fonctionnels représentent les attentes par rapport à notre application dépendant de la nature et du statut de l'acteur. Pour cela, nous avons spécifié les besoins pour chaque acteur en prenant en compte son statut :

- ❖ **Visiteur** : c'est toute personne qui consulte l'application. Il pourra :
 - Consulter le site.
 - Faire une recherche.
 - Créer un compte utilisateur.
- ❖ **Membre** : c'est un utilisateur déjà inscrit sur notre application. Il pourrait s'agir soit d'un particulier soit d'un professionnel de l'immobilier. Il pourra :
 - Consulter le site.

- Faire une recherche.
- Publier/modifier/supprimer une annonce.
- Ajouter/retirer un commentaire.
- Signaler une annonce ou un commentaire.

❖ **Administrateur** : c'est la personne qui gère l'application. Il pourra :

- Supprimer un compte.
- Supprimer une annonce.
- Supprimer un commentaire.

5.1.2. Les besoins non-fonctionnels

Ce sont des besoins qui caractérisent le système, ils sont liés aux contraintes pesant sur les fonctionnalités. La liste suivante représente les besoins non fonctionnels de notre système :

Performance :

L'application doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur.

Sécurité :

- Aucune opération (ajout, modification, suppression) ne doit être possible sans authentification.
- Les mots de passe doivent être hachés puis stockés et jamais stockés en clair.

Ergonomie :

- Les informations doivent être lisibles et facilement interprétables.
- Des graphiques résumant les informations pertinentes doivent être présentés à l'utilisateur.
- La navigation doit être simple et efficace.
- L'utilisation d'icônes explicites permettant de comprendre le sens des actions que l'utilisateur s'apprête à effectuer.
- Le système ne doit pas nécessiter une formation longue durée pour être utilisé, il doit offrir une prise en main rapide et facile sans beaucoup d'efforts.

Modularité du code :

Etude préalable

Compte tenu du contexte et de la problématique traitée, on se doit de penser à l'évolution potentielle du système. Dans cette optique, il est primordial d'écrire un code lisible, modulaire et bien documenté pour faciliter la maintenance, la personnalisation, et l'évolutivité de la solution proposée.

6. Conclusion

Dans ce chapitre, nous avons présenté une étude préalable sur les applications web dédiées à la publication d'annonces immobilières existantes en Algérie. Cela nous a permis de déterminer les acteurs qui interagissent avec le système ainsi que les besoins à prendre en considération lors de la conception de notre application, qui sera l'objet du chapitre suivant.

Chapitre 2 : Etude conceptuelle

1. Introduction

Ce deuxième chapitre est dédié à la phase de conception de notre application. Nous avons commencé par la présentation de la méthodologie de développement que nous avons suivi dans notre projet, à savoir le processus de développement PU (Processus Unifié) [15], que nous avons jugé le plus adapté à sa réalisation. Puis, nous sommes passés à la description du langage de conception qui est UML (Unified Modeling Language) [16] et de ses différents diagrammes.

Nous avons par la suite identifié les différents acteurs du système ainsi que leurs actions représentées en cas d'utilisations. Ceci nous a permis de modéliser les diagrammes de cas d'utilisations, de séquence et de classe afin de spécifier de façon détaillée les aspects fonctionnels, dynamiques et statiques du système.

2. Processus de développement

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles. Plus simplement, un processus doit permettre de répondre à la question fondamentale : « Qui fait quoi et quand ? ».

2.1. Un processus simplifié pour les applications web

Le processus que nous avons choisi de suivre pour le développement de notre application web est le Processus Unifié (PU). Il représente un cadre général très complet de processus de développement, et les méthodes agiles en vogue actuellement, telles que XP (eXtreme Programming) et Scrum. Il s'inspire également des bonnes pratiques prônées par les tenants de la modélisation agile (Agile Modeling) [15].

2.2. Les principes fondamentaux du Processus Unifié (PU)

Le PU est un processus de développement logiciel itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques [15] :

Etude conceptuelle

- ✓ **Itératif et incrémental** : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global du projet. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- ✓ **Centré sur l'architecture** : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- ✓ **Piloté par les risques** : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- ✓ **Conduit par les cas d'utilisation** : Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

2.2.1. Les phases et les disciplines du PU

La gestion d'un tel processus est organisée suivant les quatre phases suivantes [17] :

- **La phase d'initialisation** : Conduit à définir la « vision » du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.
- **La phase d'élaboration** : Poursuit trois objectifs principaux en parallèle :
 - Identifier et décrire la majeure partie des besoins des utilisateurs,
 - Construire (et pas seulement décrire dans un document !) l'architecture de base du système,
 - Lever les risques majeurs du projet.
- **La phase de construction** : Consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.
- **La phase de transition** : Permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

3. Choix du langage de conception

Nous avons choisi UML [16] car il permet l'obtention d'une modélisation de très haut niveau de notre système indépendant des langages et des environnements, et de documenter notre projet.

3.1. C'est quoi le UML ?

UML se définit comme un langage de modélisation graphique et textuel, il décrit des besoins, spécifier et documenter des systèmes.

UML unifie les notations et les concepts orientés objet. Il se n'agit pas d'une simple notation graphique car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots de langage.

UML unifie également les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis l'expression de besoin jusqu'au codage [16].

UML 2.2 s'articule autour de treize types de diagrammes (figure 2.1), chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes :

3.1.1. Six diagrammes structurels

Diagramme de classes : Il montre les briques de base statiques comme les classes, les associations, les interfaces, les attributs, les opérations, les généralisations, etc.

Diagramme de packages : Il montre l'organisation logique du modèle et les relations entre packages.

Diagramme de structure composite : Il montre l'organisation interne d'un élément statique complexe.

Diagramme de composants : Il montre des structures complexes, avec leurs interfaces fournies et requises.

Diagramme de déploiement : Il montre le déploiement physique des « artefacts » sur les ressources matérielles.

3.1.2. Sept diagrammes comportementaux

Diagramme de cas d'utilisation : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

Diagramme de vue d'ensemble des interactions : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.

Etude conceptuelle

Diagramme de séquence : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.

Diagramme de communication : Il montre la communication entre objets dans le plan au sein d'une interaction.

Diagramme de temps : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.

Diagramme d'activité : Il montre l'enchaînement des actions et décisions au sein d'une activité.

Diagramme d'états : Il montre les différents états et transitions possibles des objets d'une classe.

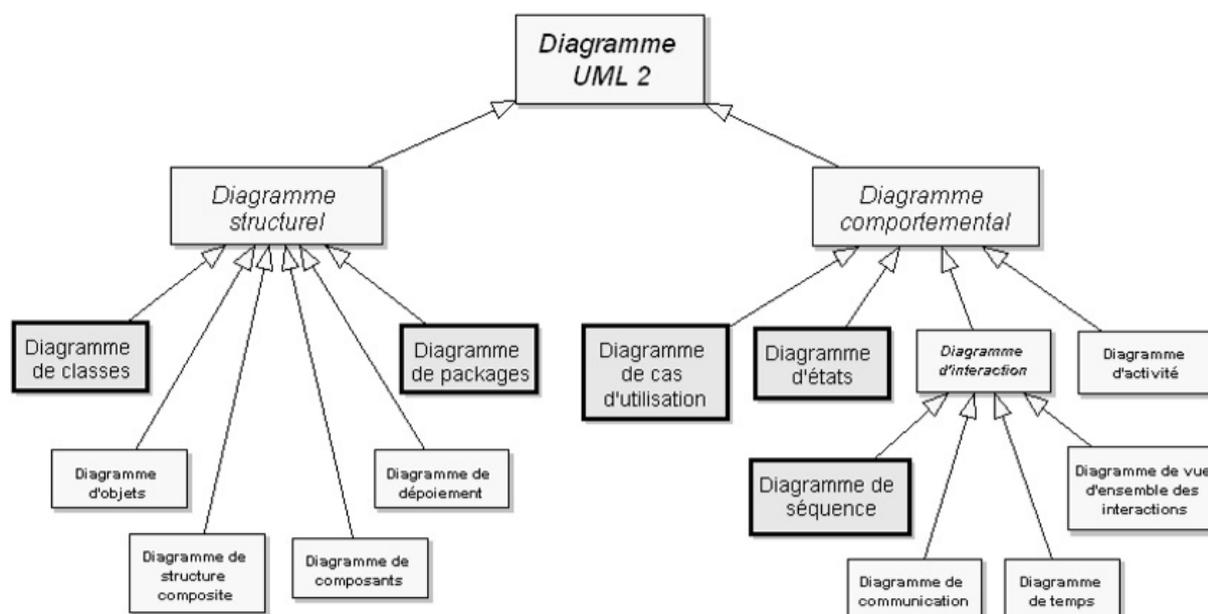


Figure 2. 1 : Les diagrammes UML 2.2 [16].

3.2. Définition détaillée de quelques diagrammes

3.2.1. Diagramme de classes

Ce diagramme permet de donner la représentation statique du système à développer, cette représentation est basée sur les concepts de classe et association. Chaque classe se décrit par des données et les traitements dont elle est responsable [18].

Etude conceptuelle

Classe : elle définit la description abstraite, servant de modèle, est d'un ensemble d'objets ayant un état similaire, un comportement identique, mais une identité différente. Une classe comporte 3 éléments :

Nom de la classe : description de modèle objet.

Attributs : informations qualifiant l'objet, variables associées à la classe.

Méthodes : compétences, opérations, actions et réactions de l'objet, fonctions associées à la classe.

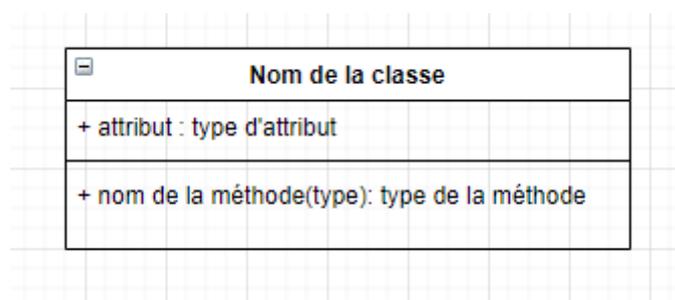


Figure 2. 2 : Représentation graphique d'une classe.

3.2.2. Diagramme de cas d'utilisation

Il est utilisé dans l'activité de spécification des besoins. Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

✚ **Acteur** : Un acteur représente un rôle joué par une entité qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [18].

On peut distinguer trois types d'acteurs :

- **Humain** : utilisateur du système au travers des différentes interfaces.
- **Logiciel** : entité logicielle existante et fonctionnelle qui communique avec le système grâce à une interface logicielle.
- **Matériel** : entité matériel qui exploite les données du système, ou est piloté par le système.

Représentation graphique :

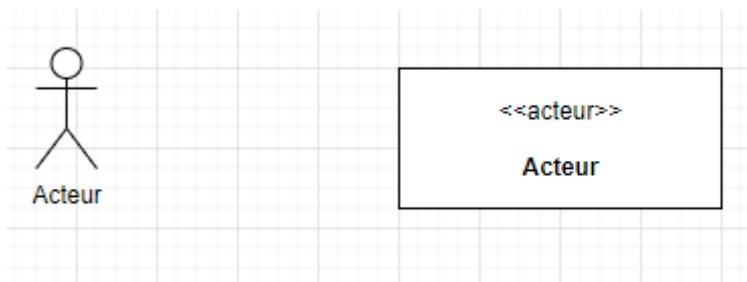


Figure 2. 3 : Représentation graphique d'un acteur.

- ✚ **Cas d'utilisation :** Un cas d'utilisation (use case en anglais) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné.

Représentation graphique :

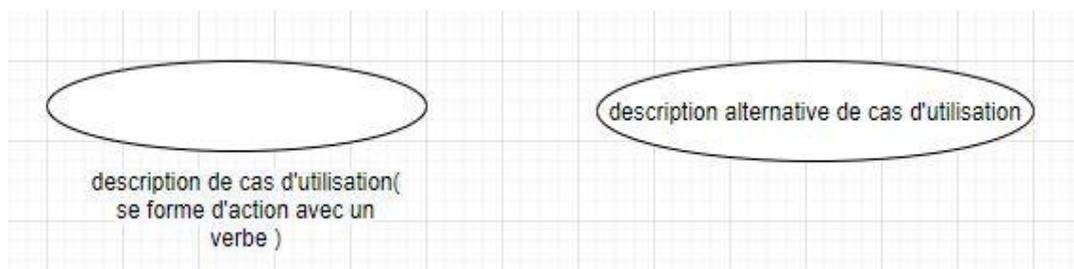


Figure 2. 4: Représentation graphique d'un cas d'utilisation.

- ❖ **Relation :** Précise une interaction unidirectionnelle ou bidirectionnelle mettant en jeu deux éléments de type acteur ou cas d'utilisation, celle-ci peut être deux type :

Entre acteur et cas d'utilisation : représente l'interaction du système avec l'extérieur, appelée association.

Entre deux entités de même type : représente le comportement entre les entités (généralisation, dépendance, inclusion ou extension).

- **Une association :** utiliser pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement : « participe à ».

Etude conceptuelle

- **Une relation d'inclusion** : Formalisée par le mot-clé <<include>> ; le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire.

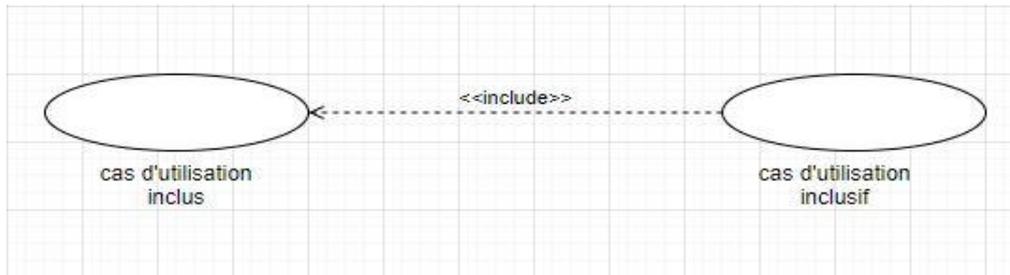


Figure 2. 5 : Représentation graphique de relation d'inclusion «include» dans le diagramme de cas d'utilisation.

- **Une relation d'extension** : Formalisée par le mot-clé <<extend>> ; le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle.

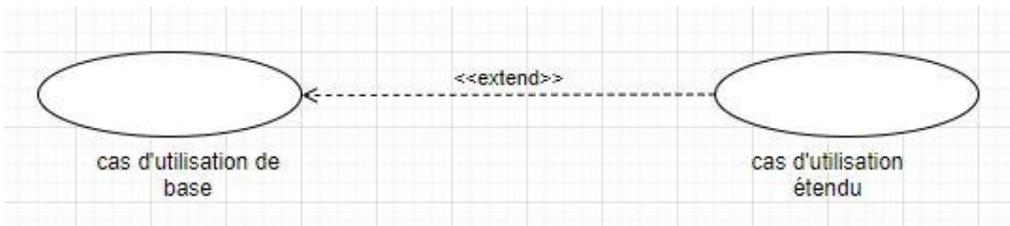


Figure 2. 6 : Représentation graphique de relation d'extension «extend» dans le diagramme de cas d'utilisation.

- **Une relation de généralisation/spécialisation** : Les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des interactions spécifiques supplémentaires.



Figure 2. 7 : Représentation graphique de relation généralisation.

3.2.3. Diagramme de séquence

Il décrit le déroulement de chaque cas d'utilisation, il montre le séquençage vertical des interactions passées entre objets dans le cadre d'un scénario. Le diagramme de séquence permet de visualiser les messages par une lecture de haut en bas. L'axe vertical représente l'enchaînement des actions dans le temps et l'axe horizontal les objets qui collaborent. Une ligne verticale pointillée est attachée à chaque objet et représente sa durée de vie [18].

Message : Un message définit une communication particulière entre des lignes de vie des objets. Ainsi, un message est une communication d'un objet vers un autre objet. La réception d'un message est considérée par l'objet récepteur comme un événement qu'il faut traiter ou pas.

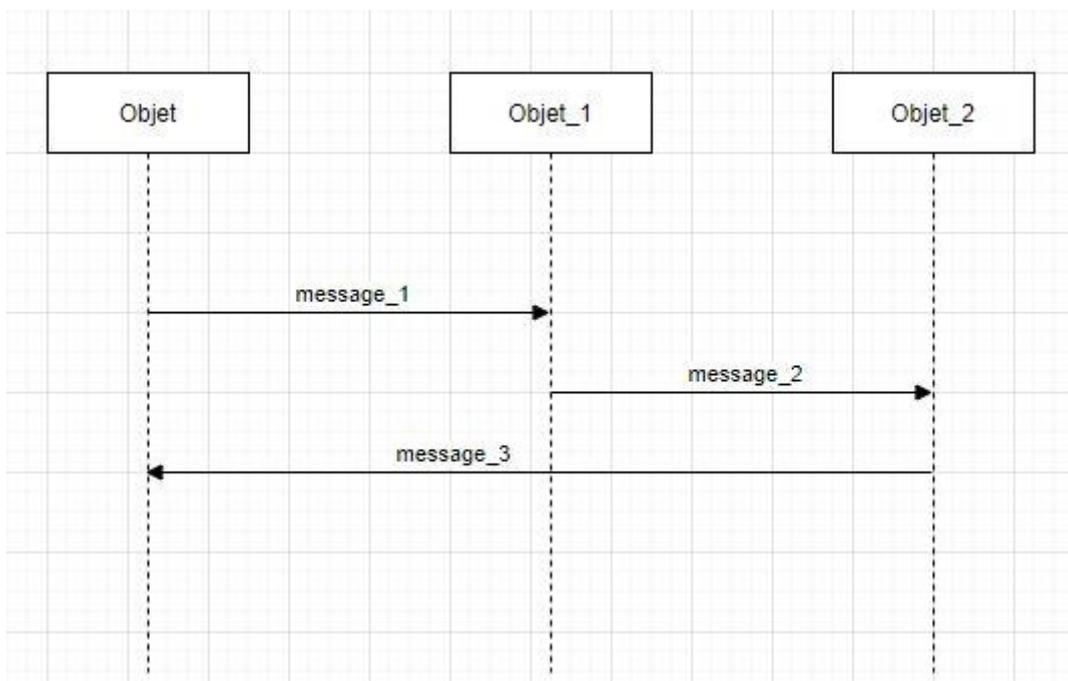


Figure 2. 8 : Représentation graphique d'un message dans le diagramme de séquence.

Objet : Les objets sont des entités appartenant au système <<instance d'une classe>> ou se trouvant à ses limites <<acteurs>>. Un objet est représenté par un rectangle (ou acteur), ainsi qu'une ligne pointillée vertical partant de ce rectangle et dirigée vers le bas qui représente la ligne de vie de l'objet. Ils représentent soit des concepts abstraits, soit des acteurs, ou des objets d'implantation.

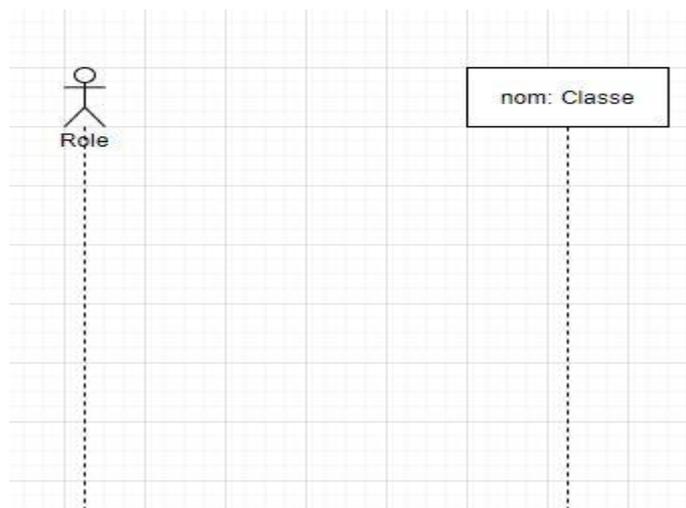


Figure 2. 9: Représentation graphique d'un objet dans le diagramme de séquence.

4. Identification des acteurs

Notre application fait intervenir trois acteurs :

Administrateur : C'est l'acteur qui a pour rôle d'assurer le bon fonctionnement du système.

Membre : C'est l'acteur qui a pour rôle la consultation du site, la gestion de son compte sur le site, ...etc.

Visiteur : C'est l'acteur qui a pour rôle de consulter le site et de s'inscrire.

5. Identification des cas d'utilisation

Un cas d'utilisation c'est un ensemble d'interactions d'un acteur avec le système. Le tableau suivant représente les différents cas d'utilisations et leurs acteurs concernés :

N° :	Cas d'utilisation	Acteurs
1	S'inscrire	Visiteur
2	Consulter l'application	Visiteur/ Membre
3	Faire une recherche	
4	S'authentifier	Membre/ Administrateur
5	Modifier son compte	
6	Publier une annonce	Membre
7	Signaler une annonce	
8	Modifier son annonce	
9	Supprimer son annonce	

10	Consulter ses notifications	
11	Ajouter un commentaire	
12	Supprimer un commentaire	
13	Signaler un commentaire	
14	Consulter ses annonces	Administrateur
15	Supprimer un membre	
16	Supprimer une annonce	
17	Supprimer un commentaire	

Tableau 2. 1 : Identification des cas d'utilisation.

6. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente l'interaction de l'utilisateur avec le système, le diagramme de cas d'utilisation peut identifier les différents utilisateurs d'un système et les différents cas d'utilisation [19].

Nous allons ici modéliser les diagrammes de cas d'utilisation des 3 acteurs de notre système pour avoir une vue globale de ce système.

6.1. Diagramme de cas d'utilisation de « Visiteur »

Le visiteur peut accéder au site et consulter les offres disponibles, comme il lui est possible de faire une recherche, et de devenir membre en s'inscrivant à l'application.

La figure 2.10 montre le diagramme de cas d'utilisation 'visiteur'.

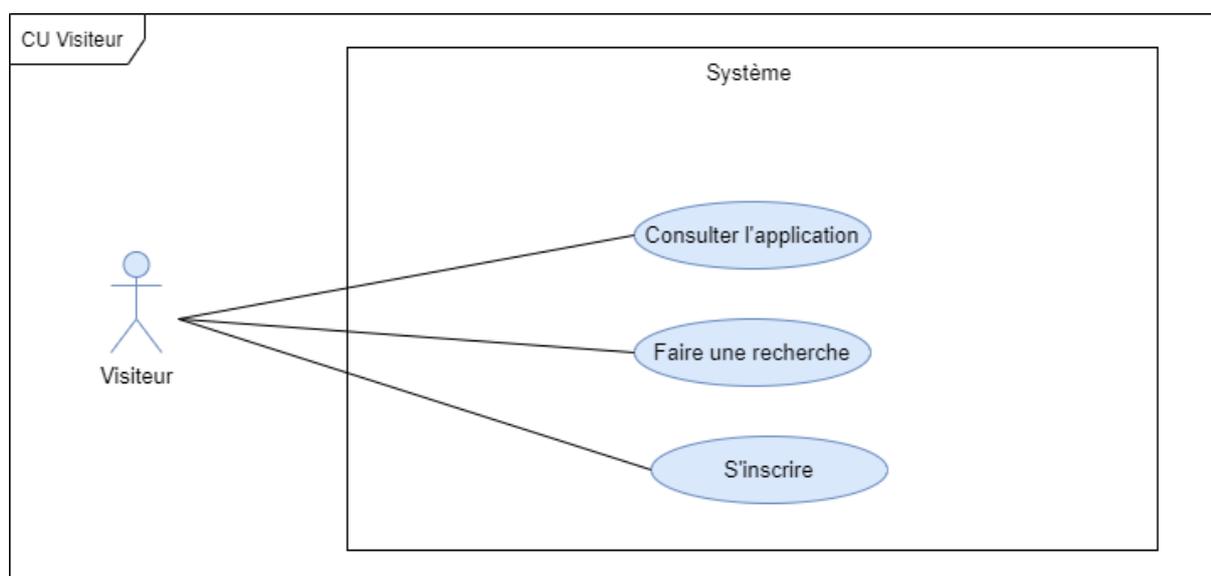


Figure 2. 10 : Diagramme de cas d'utilisation associé à l'acteur « Visiteur ».

6.2. Diagramme de cas d'utilisation de « Membre »

L'utilisateur membre doit être déjà inscrit, il peut faire une recherche ou bien consulter les offres disponibles sans authentification, néanmoins, il doit s'authentifier pour publier, modifier ou supprimer ses annonces dans son panier de publication. Il peut aussi consulter son journal de notification, modifier les paramètres de son compte, ajouter ou supprimer son commentaire et signaler les publications qui ne respectent pas les règles de la plateforme.

La figure 2.11 montre le diagramme de cas d'utilisation 'Membre'.

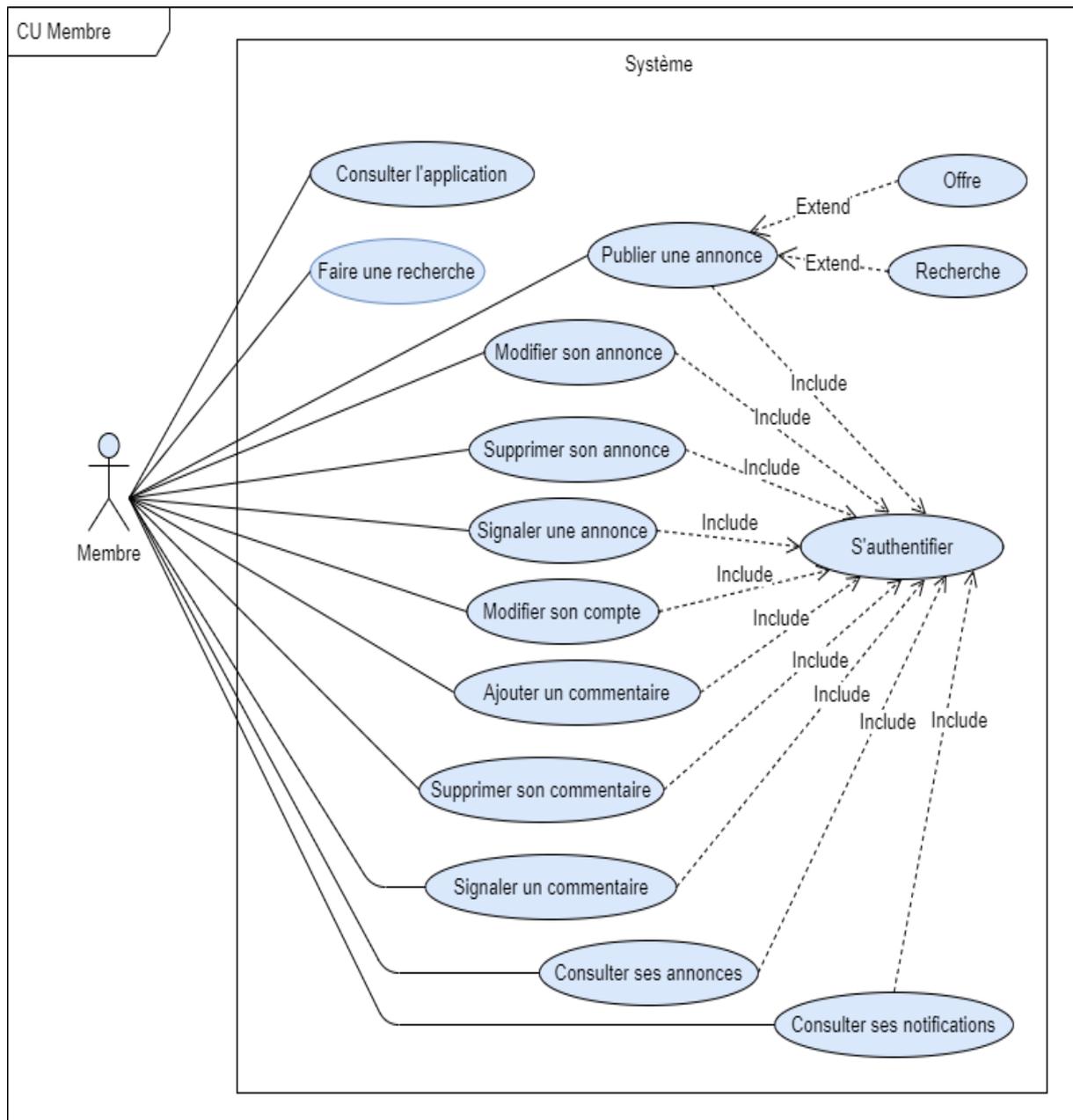


Figure 2. 11: Diagramme de cas d'utilisation associé à l'acteur «Membre ».

6.3. Diagramme de cas d'utilisation de « Administrateur »

Le rôle de l'administrateur est de garder le contrôle sur le bon fonctionnement de l'application. Il doit supprimer les membres, les publications et les commentaires qui ne respectent pas les règles de la plateforme.

La figure 2.12 montre le diagramme de cas d'utilisation 'Administrateur'.

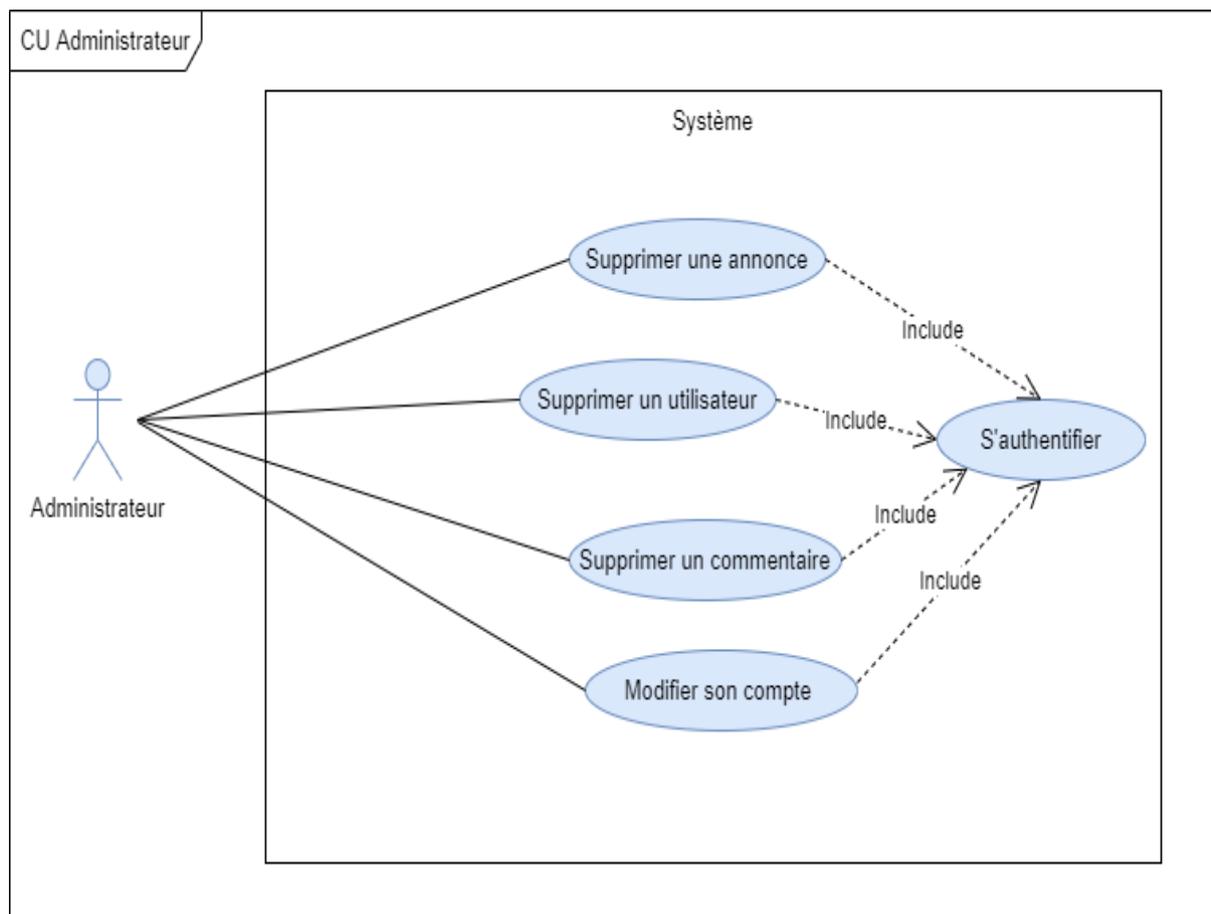


Figure 2. 12 : Diagramme de cas d'utilisation associé à l'acteur « Administrateur ».

7. Description des cas d'utilisation

7.1. Cas d'utilisation « S'inscrire »

Sommaire d'identification	
Titre	S'inscrire
Acteur	Visiteur
Résumé	Le visiteur doit remplir le formulaire d'inscription.
Description des scénarios	
Pré condition	Etre sur la page d'accueil du site.

Scénario nominal	<ol style="list-style-type: none"> 1. Le visiteur demande à s'inscrire. 2. L'application affiche le formulaire d'inscription. 3. Le visiteur remplit tous les champs obligatoires et valide. 4. Un message de succès est affiché et son espace personnel est ouvert.
Scénario alternatif	Inscription non effectuée : <ol style="list-style-type: none"> 1. L'un des champs obligatoires est vide ou incorrecte. 2. L'utilisateur existe déjà.
Poste conditions	Le nouveau compte est enregistré dans la base de données et l'espace personnel sera créé.

Tableau 2. 2 : Description du cas d'utilisation «S'inscrire».

7.2. Cas d'utilisation « Faire une recherche »

Sommaire d'identification	
Titre	Faire une recherche
Acteur	Visiteur, Membre
Résumé	L'acteur doit remplir les champs du formulaire de recherche et le valide afin de lancer la recherche.
Description des scénarios	
Pré condition	Etre sur la page d'accueil du site.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur remplit les champs de recherche et valide. 2. Les résultats de recherche sont affichés.
Scénario alternatif	Critère de recherche non disponible : <ol style="list-style-type: none"> 1. Afficher un message résultat vide.
Poste conditions	L'utilisateur a trouvé l'annonce qu'il cherche

Tableau 2. 3: Description du cas d'utilisation « Faire une recherche ».

7.3. Cas d'utilisation « S'authentifier »

Sommaire d'identification	
Titre	S'authentifier
Acteur	Membre, Administrateur

Résumé	Membre/Administrateur doit saisir son e-mail et son mot de passe
Description des scénarios	
Pré condition	L'acteur existe dans la base de données.
Scénario nominal	<ol style="list-style-type: none"> 1. L'acteur saisit son e-mail et son mot de passe. 2. Le système vérifie si les informations saisies sont correctes. 3. Le système affiche l'espace de l'utilisateur.
Scénario alternatif	Authentification non effectuée : <ol style="list-style-type: none"> 1. email ou mot de passe incorrecte. 2. l'utilisateur (membre) n'est pas encore inscrit.
Poste conditions	Membre/Administrateur peut accéder à son compte.

Tableau 2. 4 : Description du cas d'utilisation « s'authentifier ».

7.4. Cas d'utilisation « Modifier son compte »

Sommaire d'identification	
Titre	Modifier son compte
Acteur	Membre, Administrateur
Résumé	L'utilisateur accède aux informations de son profil, il effectue les modifications souhaitées.
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. Membre/Administrateur accède à son profil 2. Il effectue ses modifications 3. Il clique sur sauvegarder. 4. Modification réalisée avec succès.
Scénario alternatif	Modifications non effectuées : <ol style="list-style-type: none"> 1. Modifications non enregistrées, les informations saisies sont incorrectes.
Poste conditions	Les informations de profil sont modifiées.

Tableau 2. 5: Description du cas d'utilisation «Modifier son compte».

7.5. Cas d'utilisation « Publier une annonce »

Sommaire d'identification	
Titre	Publier une annonce
Acteur	Membre
Résumé	L'utilisateur accède à une interface qui lui permet de créer une nouvelle annonce sur son espace personnel.
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur ajouter une annonce dans son compte. 2. Le formulaire d'ajout est affiché 3. Remplir les champs obligatoires et valider 4. Ajout effectué avec succès et l'annonce est publiée.
Scénario alternatif	Ajout non effectué : <ol style="list-style-type: none"> 1. L'un des champs obligatoires est vide ou incorrecte.
Poste conditions	l'utilisateur ajoute une annonce dans son journal de publications.

Tableau 2. 6 : Description du cas d'utilisation «Publier une annonce».

7.6. Cas d'utilisation « Signaler une annonce »

Sommaire d'identification	
Titre	Signaler une annonce
Acteur	Membre
Résumé	L'utilisateur accède aux annonces dans le site, s'il rencontre une annonce qui ne respecte pas les conditions de l'application, il signale l'annonce à l'administrateur.
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur consulte les annonces. 2. Si l'annonce ne respecte pas les conditions de la plateforme : <ul style="list-style-type: none"> - Il signale l'annonce à l'administrateur.
Scénario alternatif	Aucun.
Poste conditions	L'annonce est signalée à l'administrateur

Tableau 2. 7 : Description du cas d'utilisation «Signaler une annonce».

7.7. Cas d'utilisation « Modifier son annonce »

Sommaire d'identification	
Titre	Modifier son annonce
Acteur	Membre
Résumé	L'utilisateur accède au journal de ses annonces, il sélectionne et modifie l'annonce souhaité
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à ses annonces publiées. 2. Il sélectionne l'annonce qu'il veut modifier. 3. Il clique sur modifier, puis il réalise ses modifications et valide. 4. Modification réalisée avec succès.
Scénario alternatif	Modification non effectuée : <ol style="list-style-type: none"> 1. champs obligatoire vide ou les informations saisies sont incorrectes.
Poste conditions	Les critères de l'annonce sont modifiés.

Tableau 2. 8: Description du cas d'utilisation «Modifier son annonce».

7.8. Cas d'utilisation « Supprimer son annonce »

Sommaire d'identification	
Titre	Supprimer son annonce
Acteur	Membre
Résumé	L'utilisateur accède au journal de ses annonces, il sélection et supprime l'annonce souhaité.
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à ses annonces publiées 2. Il sélectionne l'annonce qu'il veut supprimer 3. Il clique sur supprimer, puis valide. 4. Suppression réalisée avec succès.
Scénario alternatif	<ol style="list-style-type: none"> 1. L'utilisateur accède à ses annonces publiées. 2. Il sélectionne l'annonce qu'il veut supprimer. 3. Il clique sur supprimer, puis annulé.
Poste conditions	l'utilisateur supprime une annonce de son journal d'annonces.

Tableau 2. 9: Description du cas d'utilisation «Supprimer son annonce».

7.9. Cas d'utilisation « Supprimer un membre »

Sommaire d'identification	
Titre	Supprimer un membre
Acteur	Administrateur
Résumé	L'administrateur accède à son espace, puis il consulte la liste des membres signalés et procède à la suppression.
Description des scénarios	
Pré condition	Etre authentifié
Scénario nominal	<ol style="list-style-type: none"> 1. L'administrateur accède à la liste des membres signalés. 2. Il sélectionne le membre signalé. 3. Il clique sur supprimer, puis valide. 4. Suppression réalisée avec succès.
Scénario alternatif	<ol style="list-style-type: none"> 1. L'administrateur accède à la liste des membres signalés. 2. Il sélectionne le membre signalé. 3. Il supprime le membre de la liste seulement. 4. Il retourne sur la liste des membres signalés.
Poste conditions	L'administrateur supprime un membre signalé.

Tableau 2. 10 : Description du cas d'utilisation «Supprimer un membre».

8. Diagramme de séquence

Il représente la succession chronologique des opérations réalisées par un acteur en interaction avec le système, il indique les différentes opérations qui font passer d'un objet à un autre.

8.1. Diagramme de séquence « S'inscrire »

Pour devenir un membre de l'application, il faut d'abord s'inscrire, pour cela il faut :

- Consulter l'application ;
- Demander à s'inscrire ;
- Le système répond en affichant le formulaire d'inscription ;
- Remplir le formulaire et le valider ;
- Le système vérifie si les champs ne sont pas vides, puis vérifie si les informations sont correctes :
 - Si un champ est vide, il renvoie un message d'erreur.

- Sinon, il envoie les informations à la base de données :
 - o Si les informations sont incorrectes alors il renvoie un message d'erreur.
 - o Sinon il met à jour la base de données puis affiche l'espace membre créé,

La figure 2.13 illustre le diagramme de séquence de cas d'utilisation 'S'inscrire'.

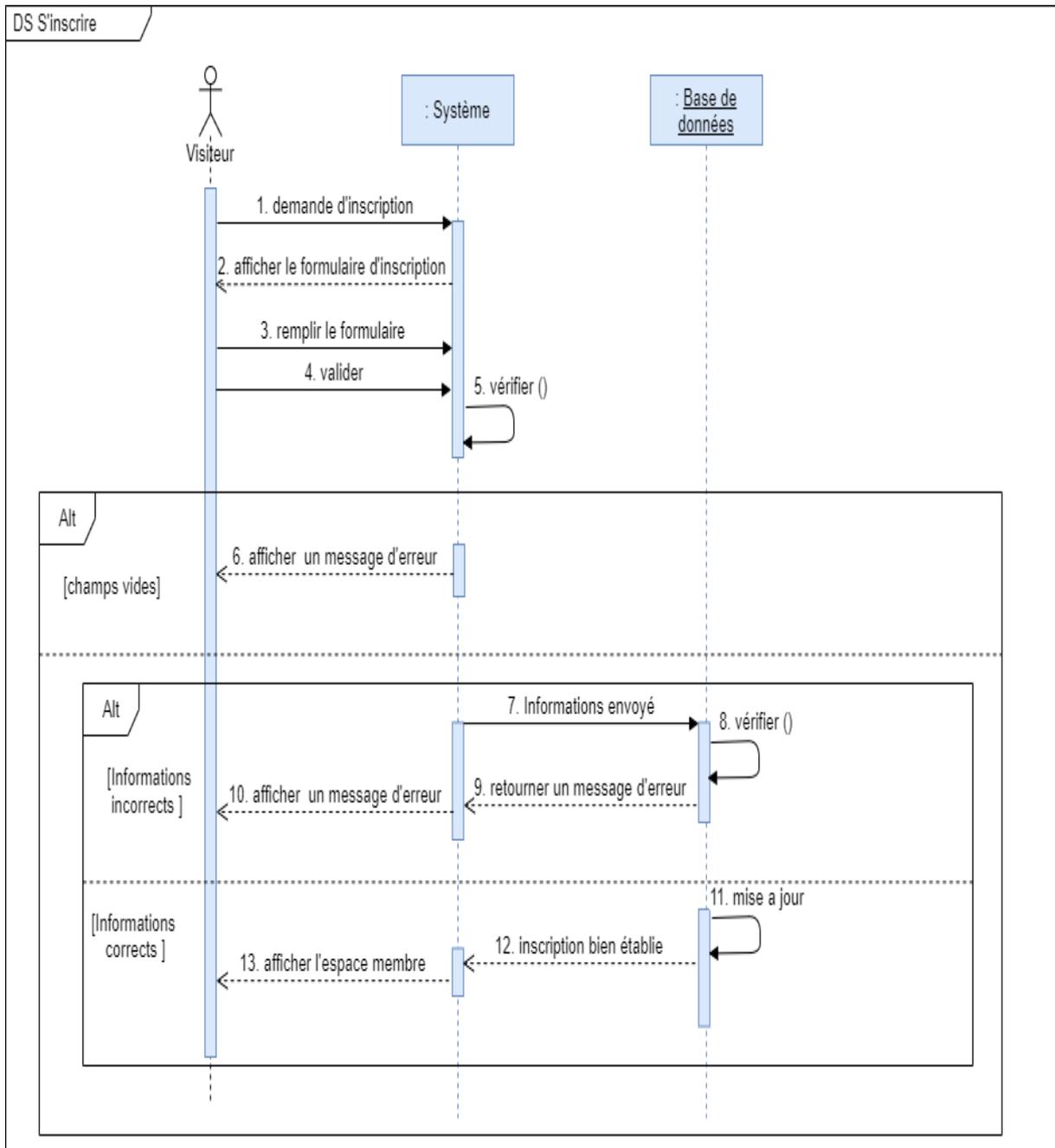


Figure 2. 13 : Diagramme de séquence du cas d'utilisation « S'inscrire ».

8.2. Diagramme de séquence « Faire une recherche »

Pour qu'un utilisateur (visiteur/ membre) puisse faire une recherche sur notre application, il faut:

- Qu'il consulte l'application ;
- Qu'il remplisse le formulaire de recherche et le valide ;
- Que le système vérifie si les informations saisies sont disponibles sur la base de données :
 - Si oui, il affiche les résultats correspondants aux informations saisies.
 - Sinon, il affiche un message d'erreur (les critères recherchés ne sont pas disponibles).

La figure 2.14 illustre le diagramme de séquence de cas d'utilisation 'Faire une recherche'.

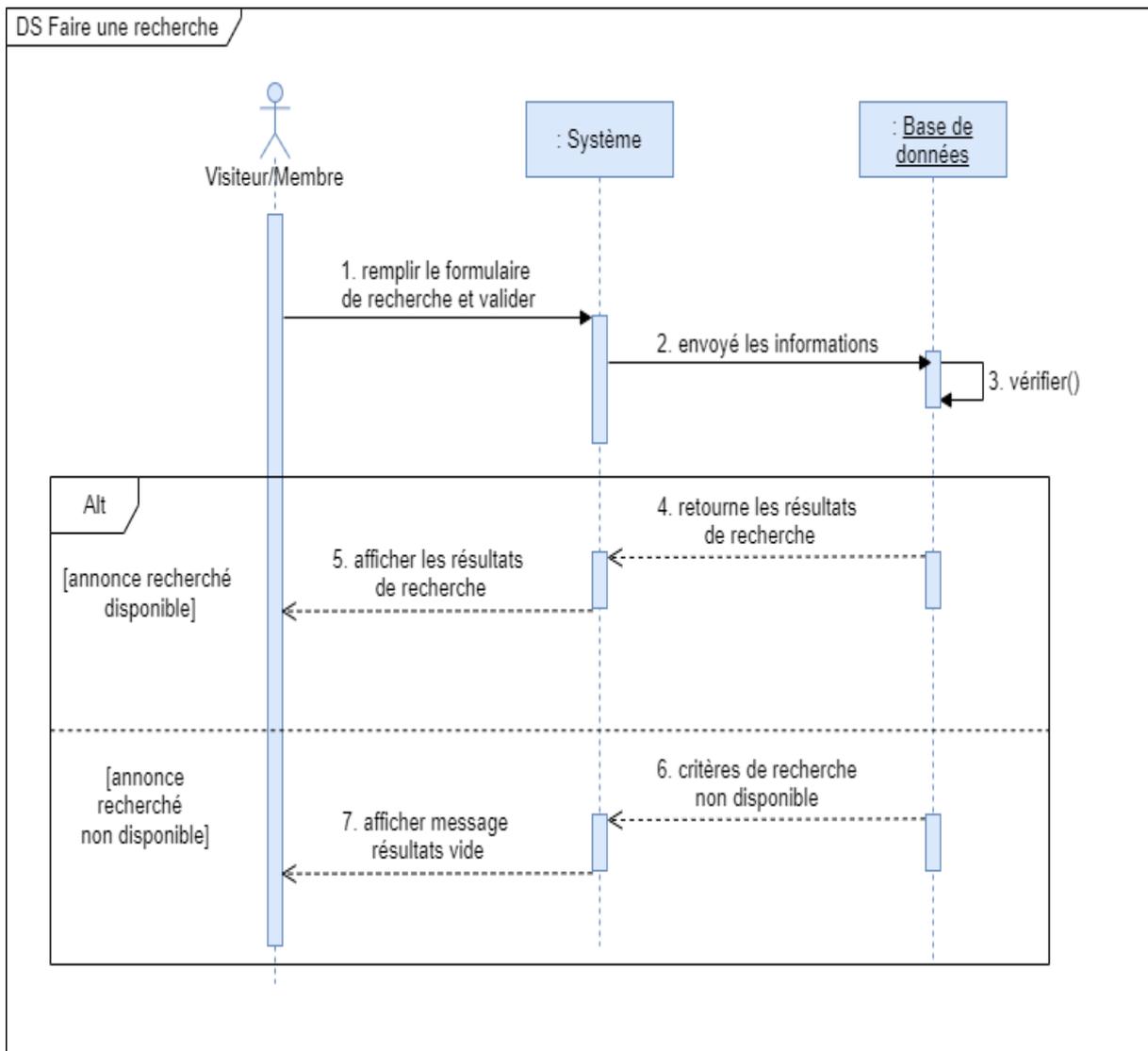


Figure 2. 14: Diagramme de séquence du cas d'utilisation «Faire une recherche ».

8.3. Diagramme de séquence « S'authentifier »

Pour qu'un membre/administrateur puisse accéder à son espace personnel, il doit d'abord s'authentifier, pour cela il faut :

- Consulter le site ;
- Demander la connexion à son espace personnel ;
- Le système répond par l'affichage du formulaire de connexion ;
- Le membre remplit le formulaire et le valide ;
- Le système vérifie si les champs ne sont pas incorrects, puis vérifie si les informations sont déjà enregistrées dans la base de données :
 - Si oui, il affiche l'espace personnel du membre,
 - Sinon, il affiche un message d'erreur : email ou mot de passe incorrecte.

La figure 2.15 illustre le diagramme de séquence de cas d'utilisation 'S'authentifier'.

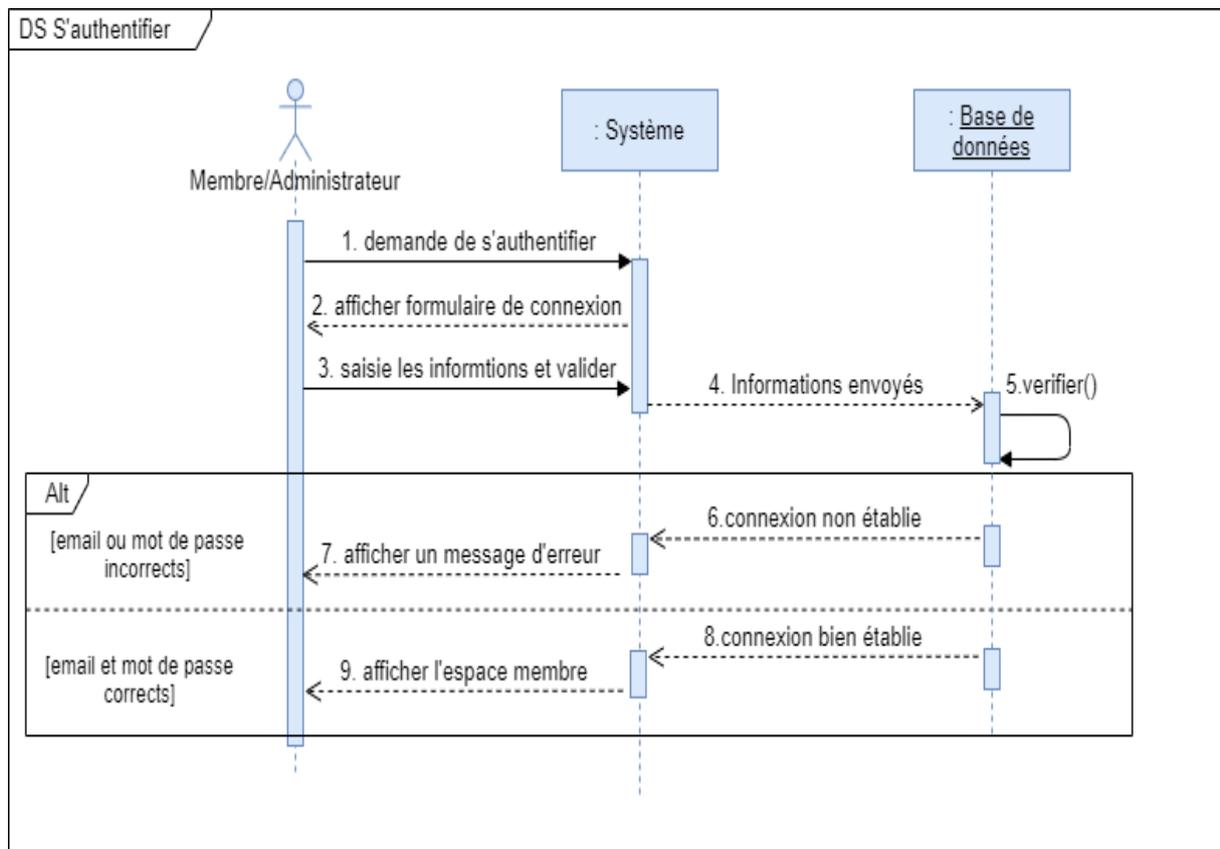


Figure 2. 15 : Diagramme de séquence du cas d'utilisation «S'authentifier ».

8.4. Diagramme de séquence « Modifier son compte »

Pour qu'un membre puisse faire des modifications sur son compte, il doit d'abord s'authentifier, puis une fois connecté, il procède aux étapes de modification suivantes :

- Accès aux paramètres ;
- Etablir les modifications et valider ;
- Si un champ est vide, le système renvoie un message d'erreur
- Sinon le système renvoie les informations à la base de données;
 - Le système valide les modifications et met à jour la base de données,
 - Sinon, il affiche un message d'erreur indiquant que les modifications n'ont pas été enregistrées.

La figure 2.16 illustre le diagramme de séquence de cas d'utilisation 'Modifier son compte'.

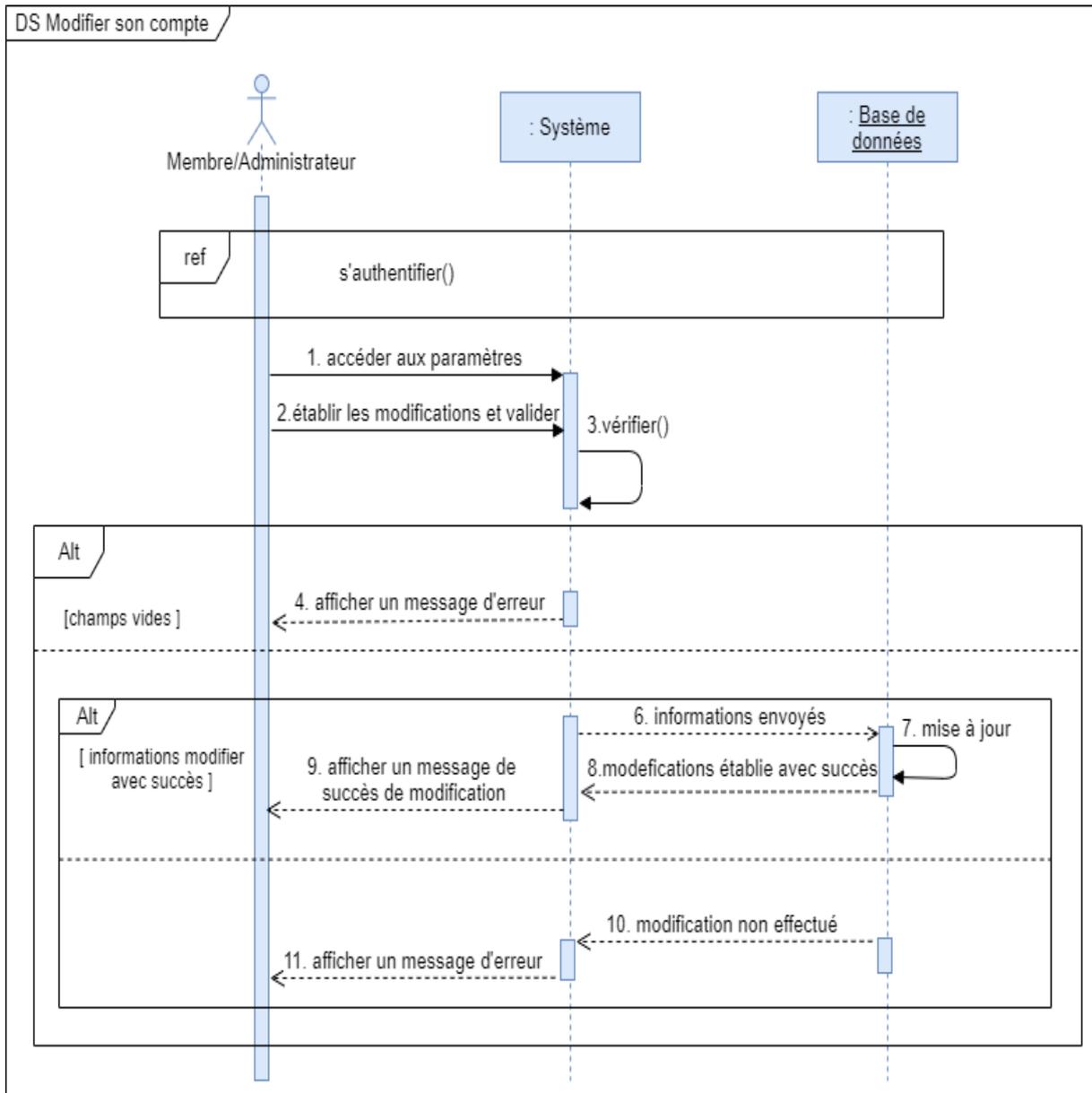


Figure 2. 16 : Diagramme de séquence du cas d'utilisation «Modifier son compte ».

8.5. Diagramme de séquence « Publier une annonce »

Pour que le membre puisse faire une publication sur son espace personnel, il doit s'authentifier, puis une fois connecté, il procède aux étapes de publication d'une annonce qui sont :

- Demande d'ajout d'une annonce ;
- Le système lui répondre en affichant un formulaire à remplir ;
- Le membre remplit le formulaire et le valide ;
- Le système procède à la vérification de tous les champs du formulaire :

Etude conceptuelle

- Si l'un des champs obligatoires est vide ou l'information saisie est incorrecte alors le système retourne un message d'erreur.
- Si les champs sont tout bien remplis sans fausses informations alors le système valide le formulaire et met à jour la base de données en ajoutant l'annonce.

La figure 2.17 illustre le diagramme de séquence de cas d'utilisation 'Publier une annonce'.

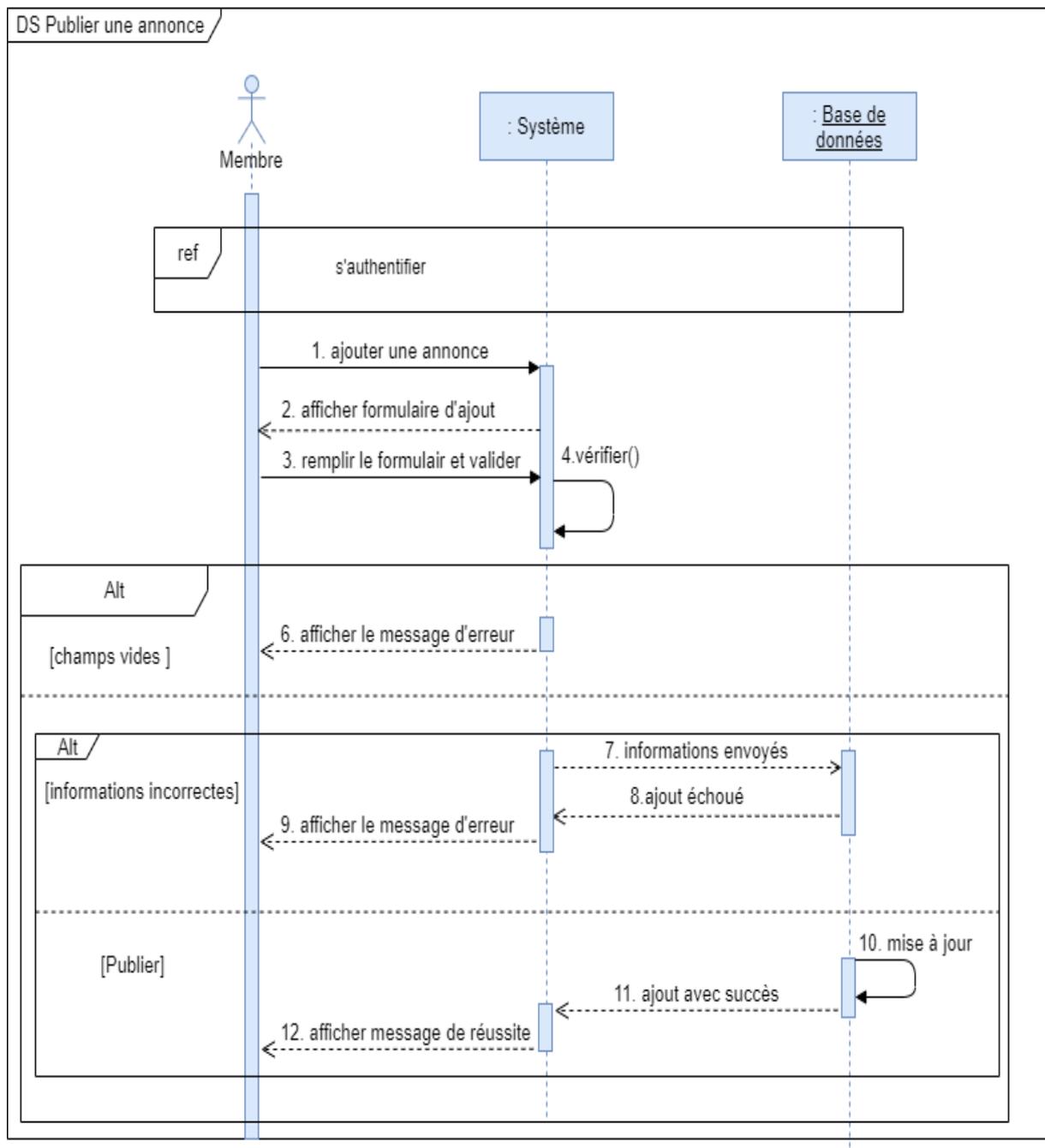


Figure 2. 17: Diagramme de séquence du cas d'utilisation «Publier une annonce ».

8.6. Diagramme de séquence « Signaler une annonce »

Pour qu'un membre puisse signaler une annonce, il doit d'abord s'authentifier, puis une fois connecté il consulte les annonces sur l'application. S'il constate qu'une annonce ne respecte pas les conditions de l'application alors :

- Il clique sur le boutons signaler.
- Le système envoie un rapport avec l'identifiant de l'annonceur à l'administrateur.

La figure 2.18 illustre le diagramme de séquence de cas d'utilisation 'Signaler une annonce'.

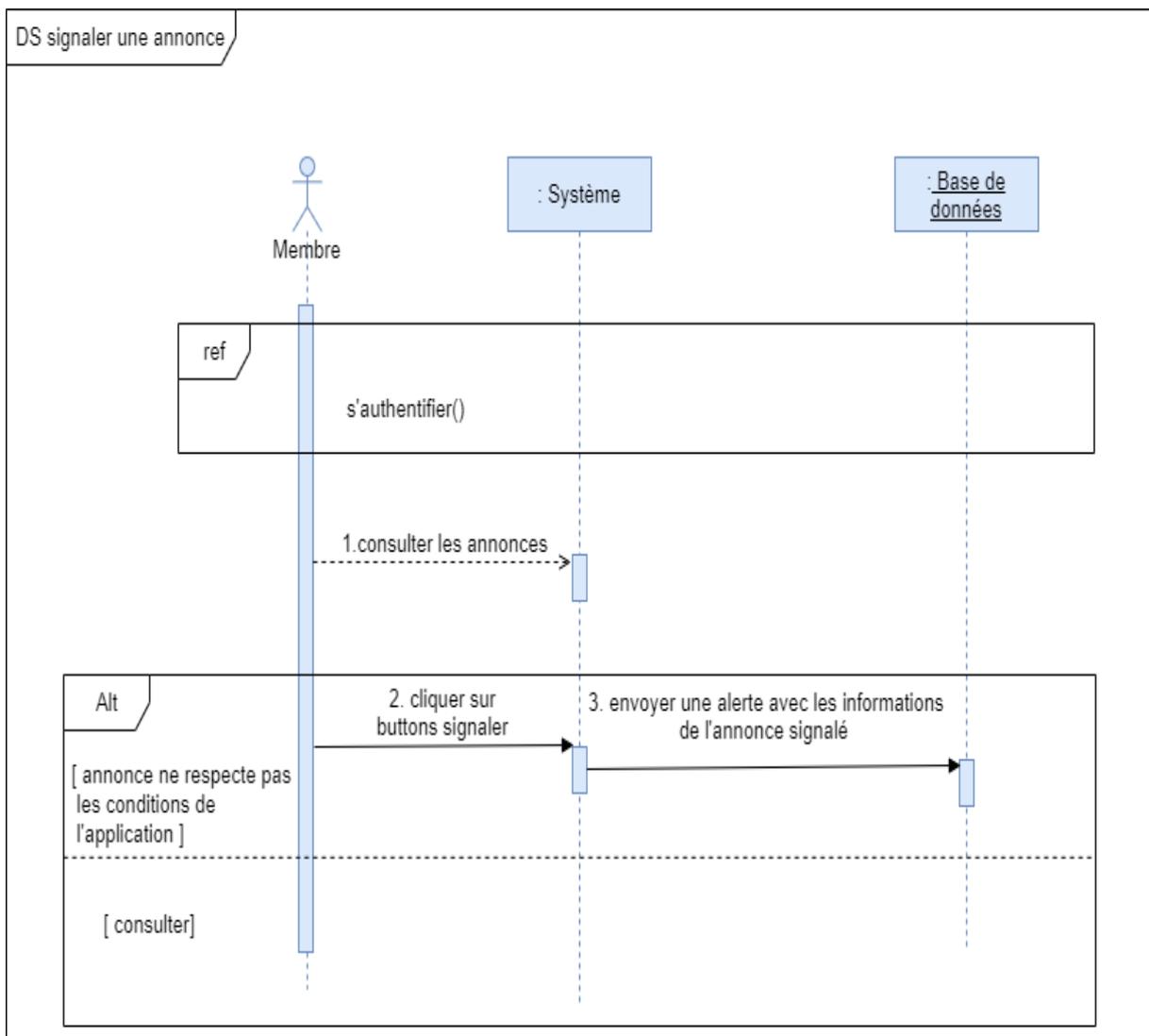


Figure 2. 18: Diagramme de séquence du cas d'utilisation «Signaler une annonce ».

8.7. Diagramme de séquence « Modifier son annonce »

Pour qu'un membre puisse modifier l'une de ses annonces, il doit d'abord s'authentifier, puis une fois connecté il consulte la page **Mes annonces** sur l'application. Il sélectionne l'annonce souhaité puis il clique sur le bouton modifier, puis :

- Saisit les modifications souhaitées et clique sur modifier.
- Sinon, il clique sur revenir au journal sans faire de modification.

La figure 2.19 illustre le diagramme de séquence de cas d'utilisation 'Modifier son annonce'.

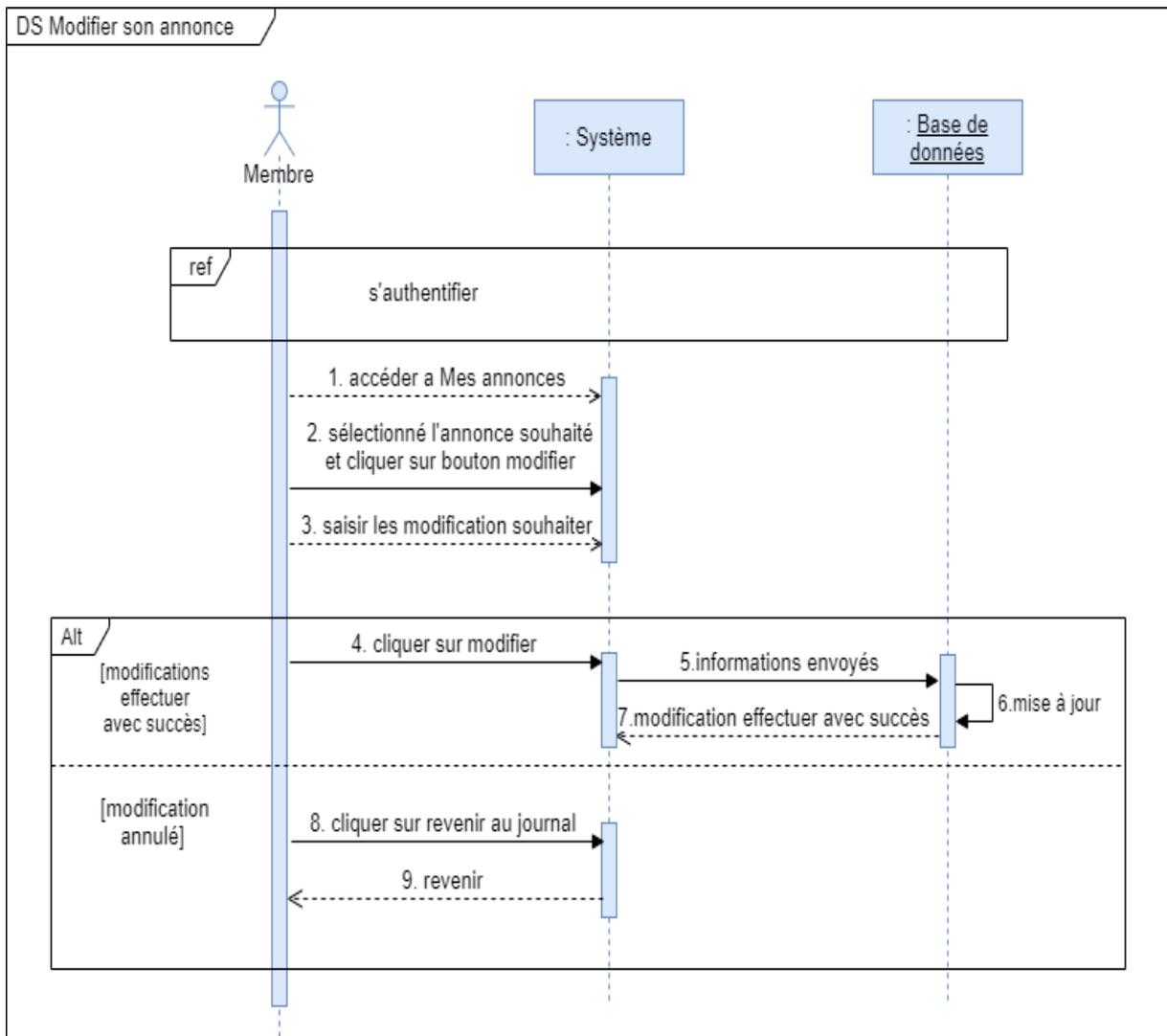


Figure 2. 19: Diagramme de séquence du cas d'utilisation « Modifier son annonce ».

8.8. Diagramme de séquence « Supprimer son annonce »:

Pour que le membre puisse faire une suppression d'une annonce sur son espace personnel, il doit d'abord s'authentifier, puis une fois connecté il procède aux étapes de suppression qui sont :

- Demande d'accès au journal d'annonces ;
- Le système répond en affichant le journal d'annonces.

Une fois le journal affiché, le membre peut :

- Sélectionner directement l'annonce.
- Supprimer et valider la suppression.
- Le système met à jour la base de données, et affichera un message de succès de suppression.

Ou bien :

- Faire une recherche de l'annonce que l'on souhaite supprimer ;
- Sélectionner l'annonce ;
- Supprimer et valider la suppression ;
- Le système met à jour la base de données, et affichera un message de succès de suppression.

La figure 2.20 illustre le diagramme de séquence de cas d'utilisation 'Supprimer son annonce'.

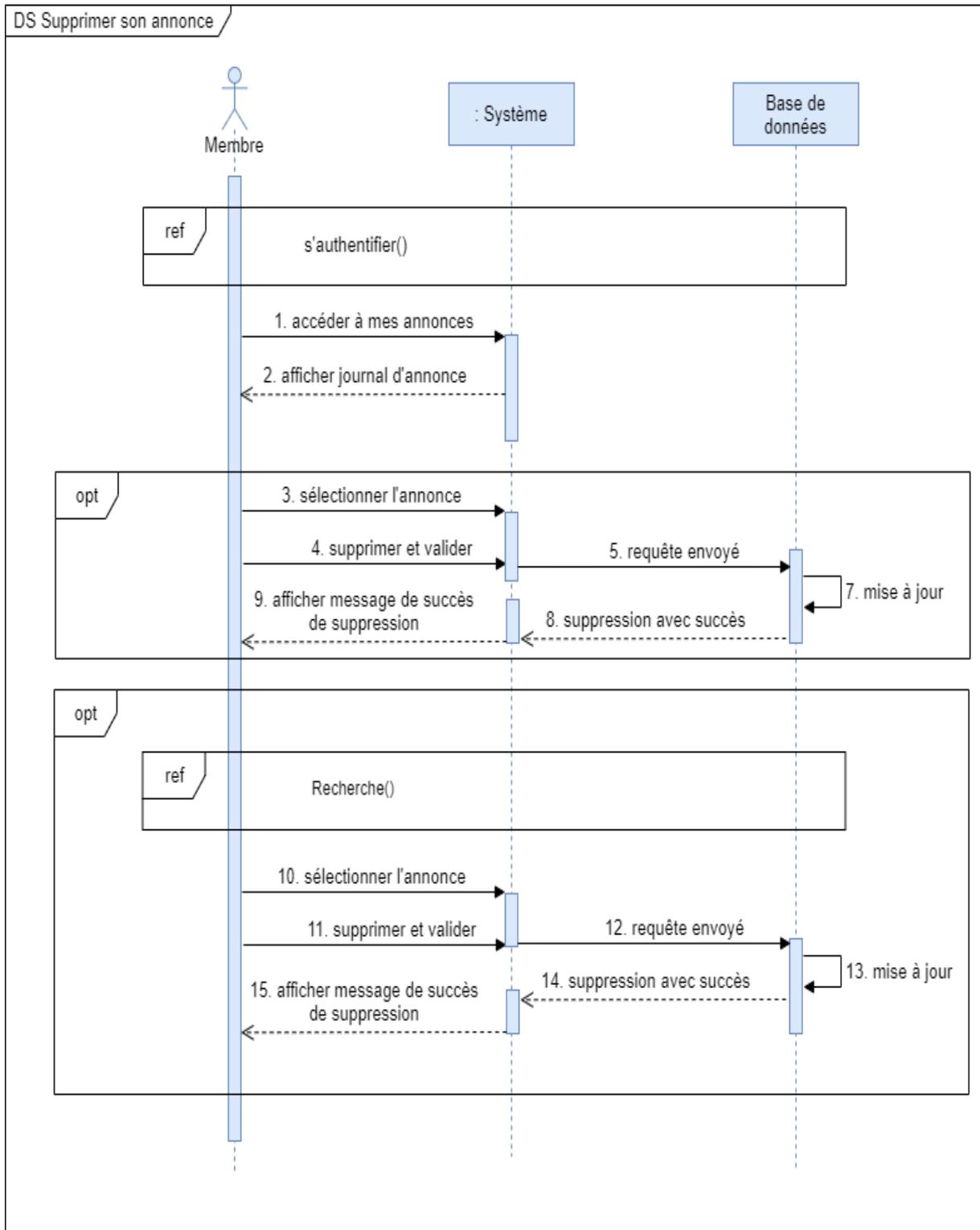


Figure 2. 20 : Diagramme de séquence du cas d'utilisation «Supprimer son annonce ».

8.9. Diagramme de séquence « Supprimer un membre »

Pour que l'administrateur puisse supprimer un membre signalé, il doit d'abord s'authentifier, puis une fois connecté il consulte la page **Les membres signalés** sur l'application. Il sélectionne le membre puis :

Il vérifie le motif de signalement :

- ✓ Si, le motif est valide alors il :
 - Clique sur supprimer et valider.
- ✓ Sinon :
 - Retire le membre de la liste.
 - Revient sur la liste **Les membres signalés**.

La figure 2.21 illustre le diagramme de séquence de cas d'utilisation 'Supprimer un membre'.

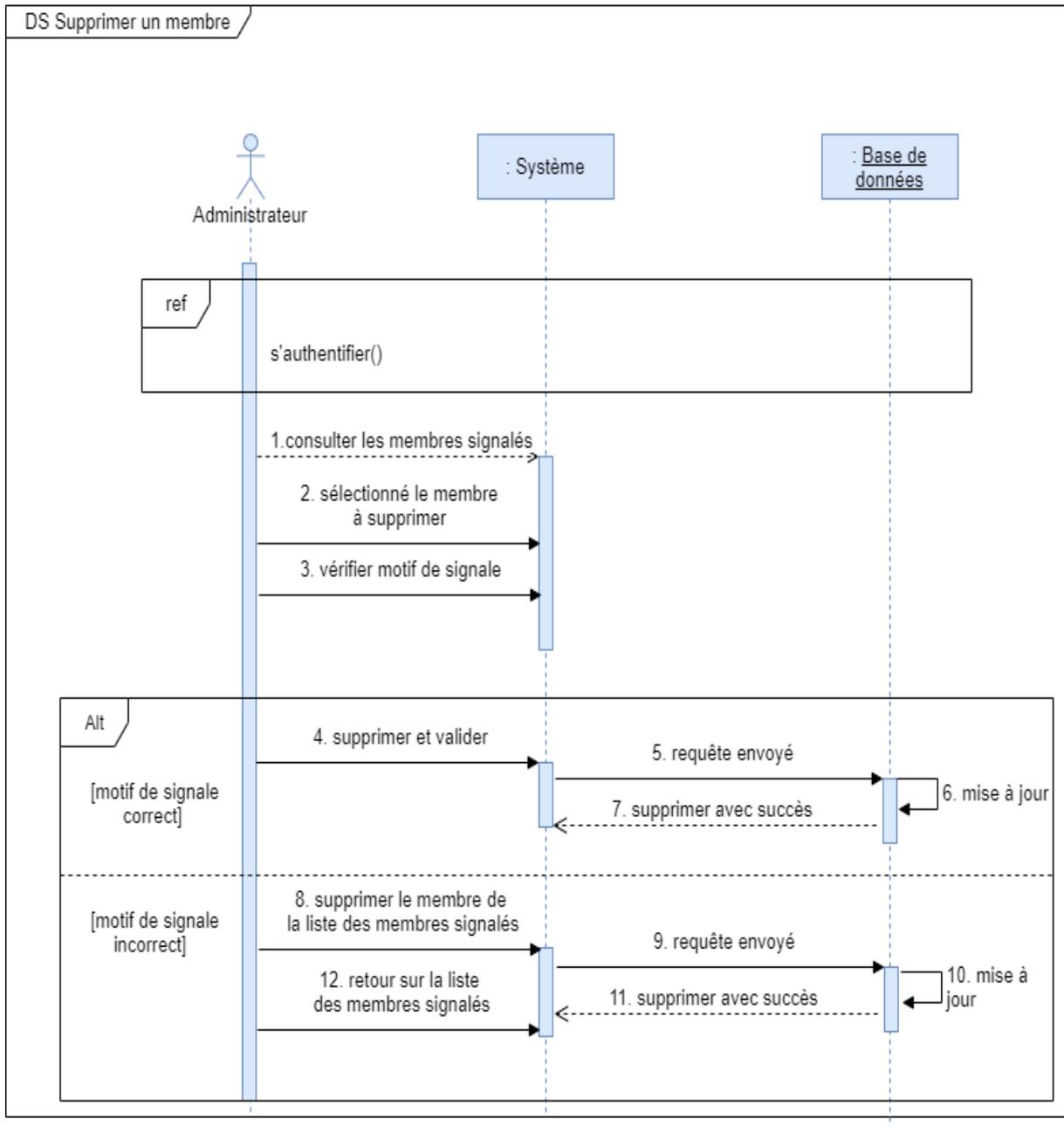


Figure 2. 21: Diagramme de séquence du cas d'utilisation «Supprimer un membre ».

9. Diagramme de classe

Il représente l'architecture conceptuelle du système, il décrit l'ensemble des classes existantes et leur structure interne et définit les relations entre les classes [19].

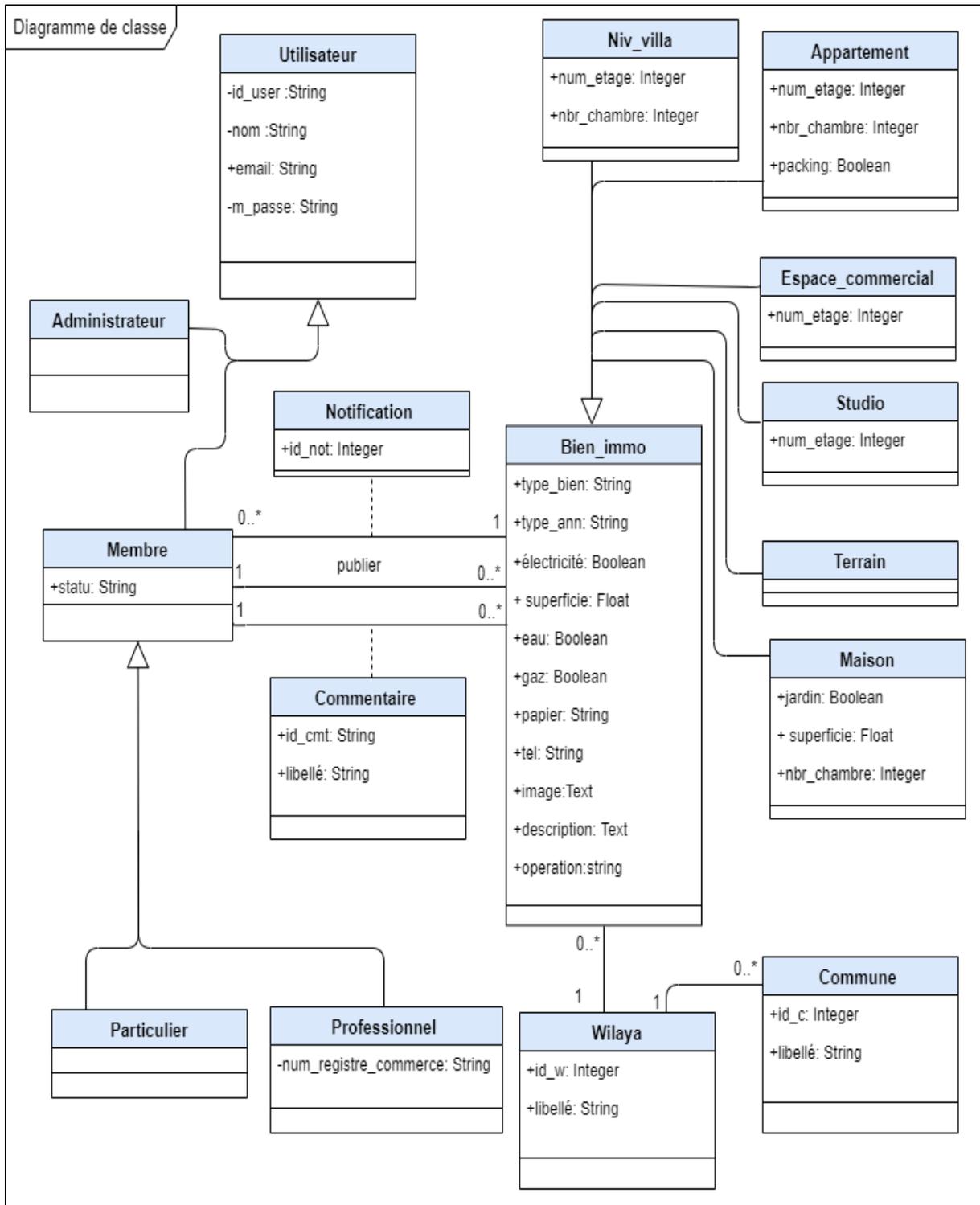


Figure 2. 22: Diagramme de classe.

10. Dictionnaire de données

Attribut	Signification	Type d'attribut
id_user	Identifiant de membre	Entier
Type_bien	Type de bien immobilier	Chaîne de caractères

Etude conceptuelle

Type_ann	Type d'annonce (offre ou recherche)	Chaîne de caractères
opération	Opération	Chaîne de caractères
description	Description	Texte
image	Image	Texte
Num_etage	Numéro d'étage	Entier
superficie	Superficie	Réel
Nbr_chambre	Nombre de chambre	Entier
papier	Papier	Chaîne de caractères
meubler	Meubler	Booléen
gaz	Gaz	Booléen
eau	Eau	Booléen
électricité	Electricité	Booléen
parking	Parking	Booléen
jardin	Jardin	Booléen
tel	Numéro de téléphone	Chaîne de caractères
Id_w	Identifiant de la wilaya	Entier
Id_c	Identifiant de la commune	Entier
Libellé	Libellé	Chaîne de caractères
statu	Statut de membre	Chaîne de caractères
E_mail	Email	Chaîne de caractères
M_passe	Mot de passe	Chaîne de caractères
nom	Nom	Chaîne de caractères
Num_register_commerce	Numéro de registre de commerce	Chaîne de caractères
Id_not	Identifiant de la notification	Entier
Id_cmt	Identifiant de commentaire	Entier
id_bien	Identifiant de bien	Entier

Tableau 2. 11: Dictionnaire de données.

11. Les règles de passage de diagramme de classe vers le schéma de base de données

Pour réaliser un passage du diagramme de classe vers un schéma de base de données, nous sommes obligés de suivre une logique et des règles qui sont les suivantes :

✚ Présence de la cardinalité 0..1 dans un coté de l'association

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'identifiant de la classe qui est associée à la cardinalité 0..1 devient la clé de l'autre classe.

✚ Présence de la cardinalité 0..* des deux côtés de l'association

- Chaque classe se transforme en une table.
- Chaque attribut de classe se transforme en un champ de table.
- L'association se transforme en une table. Cette table a comme champs l'identifiant de chacune des deux classes, plus d'éventuels autres attributs.

✚ Présence d'une généralisation

Méthode 1 :

- Créer une table avec tous les attributs des classes.
- Ajouter un attribut pour distinguer les types des objets.

Méthode 2 :

- Créer une table pour chaque sous type, chaque table se compose d'attributs génériques et d'attributs spécifiques.

Méthode 3 :

- Créer une table par classe et des associations.

12. Schéma relationnel

Pour une implémentation d'une base de données, il faut traduire le modèle conceptuel en modèle logique, cela veut dire qu'il faut pouvoir convertir un modèle UML en modèle relationnel, Les résultats de la traduction de notre diagramme de classe sont les suivants :

Commune (id_c, libellé, #idw).

Wilaya (id_w, libellé).

Etude conceptuelle

Bien_immo (id_bien, type_bien, type Ann, opération, description, image, num_etage, superficie, nbr_chambre, papier, meubler, gaz, eau, électricité, parking, jardin, tel, #idw, #idc, #id_user).

Notification (id_not, #id_bien, #id_user).

Commentaire (id_cmt, libellé, #id_user, #id_bien).

Membre (statu,num_registre_commerce).

Utilisateur (id_user, nom,mail, m_passe,).

13. Conclusion

Dans ce chapitre, nous avons modélisé les différents diagrammes nécessaires pour le développement correct de notre application, ce qui nous a permis de générer les tables de la base de données et d'avoir une vue générale du comportement théorique des fonctionnalités offertes par notre application. Cette base théorique va nous guider dans l'étape suivante qui est la réalisation de notre application.

Chapitre 3 : Réalisation et présentation de l'application

1. Introduction

Dans ce chapitre, nous allons entamer la réalisation de notre application. Nous allons commencer tout d'abord par présenter les outils de développement et les langages de programmation que nous avons exploités pour réaliser ce projet. Puis, nous présenterons le fonctionnement de notre application web à travers quelques interfaces graphiques.

2. Outils de développement utilisés

2.1. Frameworks

Un Framework (ou infrastructure logicielle en français) est un ensemble de bibliothèques, contenant des packages et des classes plus un modèle de programmation, que le développeur utilise dans le but de simplifier et d'uniformiser le développement.

2.1.1. Laravel

Laravel [20] est un Framework d'application web du langage de programmation PHP. Il est gratuit, open source et multiplateforme. Il a été conçu pour rendre le développement d'applications Web plus facile et plus rapide grâce à des fonctionnalités intégrées.

Laravel est basé sur l'architecture MVC (M : Modèle, V : Vue, C : Contrôleur), son objectif est de faciliter et d'accélérer le processus de développement d'application web, car il dispose de bibliothèque de fonctionnalités préprogrammées.

Laravel offre un environnement de développement fonctionnel, ainsi que des interfaces de ligne de commande intuitive et expressives. Pour l'accès et la manipulation des données, il utilise la Cartographie Objet Relationnel (ORM : Object-Relational Mapping.) créé en 2005 par TAYLOR Otwell.



Figure 3. 1 : Logo de Framework Laravel [21].

Réalisation et présentation de l'application

2.1.1.1. L'architecture MVC

Laravel utilise le modèle MVC, il y a donc trois parties du cadre qui fonctionnent ensemble: les modèles, les vues et les contrôleurs. Les contrôleurs sont la partie principale de la majeure partie du travail. Ils se connectent aux modèles pour obtenir, créer ou mettre à jour des données et afficher les résultats sur les vues, qui contiennent la structure HTML réelle de l'application. [21].

✓ **Modèle**

Le modèle est la source de toutes données avec laquelle un utilisateur travail. Il peut représenter les données en cours de traitement ou stockés. Chaque Modèle correspond à une table dans la base de données.

✓ **Vue**

La vue s'occupe principalement de faire afficher ce que le modèle renvoie. Ensuite, il s'occupe de recevoir toutes interactions de l'utilisateur. La vue contient la présentation graphique à renvoyer à l'utilisateur.

✓ **Contrôleur**

Les Contrôleurs agissent comme intermédiaire entre les Vues et les Modèles, il gère les interactions de l'utilisateur sur la vue, et envoie de commande au modèle pour manipulation de données (insertion, suppression, modification...etc.), et interagissent avec les vues pour restituer la sortie finale.

2.1.2. Bootstrap

C'est un Framework frontal CSS gratuit le plus populaire au monde pour la création des sites responsifs adaptés au mobile et des conceptions réactives. Il dispose des modèles de conception basé sur HTML et CSS [22].



Figure 3. 2 : Logo de la bibliothèque Bootstrap [23]

Réalisation et présentation de l'application

2.2. Persistance de données

Pour les données, Laravel contient en couche ORM par défaut qui peut être utilisée pour interagir avec les données d'applications de diverses bases de données relationnelles telles que MySQL, SQLite et PostgreSQL.

2.2.1. Eloquent ORM

ORM est une technique qui permet de manipuler et d'interroger les données d'une base de données.

Laravel propose son propre ORM Eloquent avec un large éventail de fonctions avec lesquelles travailler. Le Framework fournit également la migration et l'amorçage et propose également des restaurations [23].

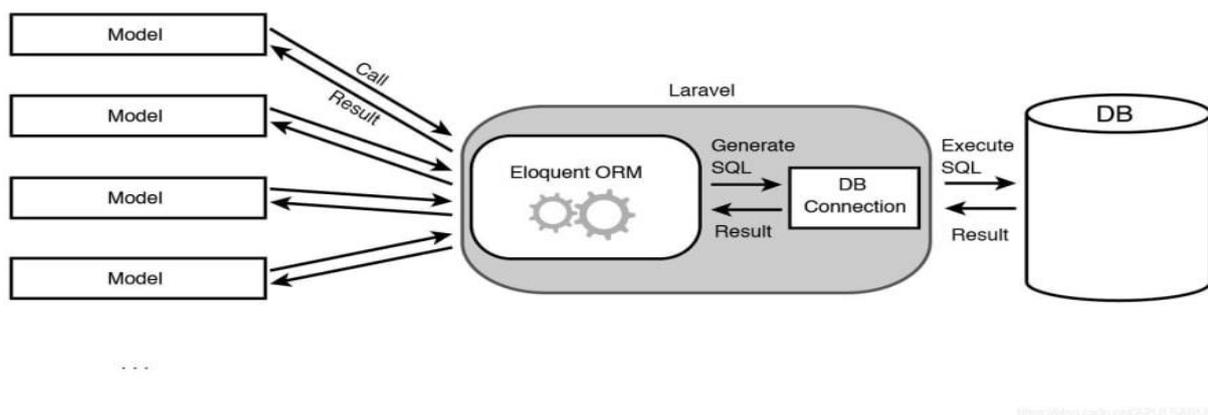


Figure 3. 3: Schéma démontant l'emplacement et le rôle de Eloquent ORM [32]

2.2.2. MySQL

MySQL est un système de gestion de base de données relationnelle open source développé par Oracle. Il est multiplateforme basé sur le langage de requête structuré SQL, son approche relationnelle lui permet de structurer les données dans des tables à deux dimension [24].



Figure 3. 4 : Logo de SGBD MYSQL [25].

Réalisation et présentation de l'application

2.3. Logiciels

Dans cette section, nous allons présenter différents logiciels utilisés pour le développement de notre application web.

2.3.1. Visual Studio

Visual Studio code est un éditeur de code source léger, puissant et multiplateforme. Il est livré avec une prise en charge intégrée de JavaScript, Type script et Nodejs et dispose de plusieurs écosystème d'extension pour d'autres langages (C++, C#, Java...etc.) et des environnements d'exécution tel que .Net et Unity [26].



Figure 3. 5: Logo de l'éditeur de code source Visual Studio Code [27].

2.3.2. Adobe Illustrator

C'est un logiciel de la création graphique vectorielle. Il fait partie de la gamme Adobe, peut être utilisé indépendamment ou en complément Photoshop, comme il offre des outils de dessins vectorielle puissants [28].



Figure 3. 6: Logo de logiciel Adobe Illustrator [28].

2.3.3. Draw.io

Draw.io est une application gratuite en ligne, accessible via son navigateur, qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre [29].

Réalisation et présentation de l'application



Figure 3. 7 : Logo de l'application Draw.io [30].

2.3.4. POSTMAN

Postman sert à exécuter des requêtes HTTP (Hyper Text Transfer Protocol) directe via son interface graphique. Il suffit de renseigner l'URL (Uniform Resource Locator) et la méthode http [19]:

GET : les requêtes GET sont celles effectuées par un navigateur lorsque vous entrez une URL dans la barre de navigation, dans le but de chercher une page ou une donnée.

POST : les requêtes POST ont pour but d'envoyer les informations contenues dans le body de la requête vers le serveur.

PUT : les requêtes PUT vont écraser une ressource avec une nouvelle donnée, là aussi présenter dans le body de la requête. Elle est utilisée pour mettre à jour la donnée à condition qu'on soit capable de fournir la ressource mise à jour dans son intégralité

PATCH : les requêtes PATCH servent également à mettre à jour une ressource mais en ne modifiant que l'élément envoyé en body de la requête.

DELETE : comme son nom l'indique, la requête DELETE sert à effacer une ressource.



Figure 3. 8 : Logo de logiciel POSTMAN [32].

Réalisation et présentation de l'application

2.4. Langages et techniques

Dans cette section nous allons représenter l'ensemble des langages de programmation et techniques que nous avons utilisé pour la réalisation de notre application web.

2.4.1. PHP

Le PHP signifie Hyper Text Preprocessor, c'est un langage de programmation ou avec un autre thème un langage de script. C'est un langage sous licence libre. Il s'utilise pour la création des sites web dynamique, il génère du code HTML, CSS ou XHTML, JavaScript...etc.

Ce langage est créé début des années 1990 par le Canadien Grenlandais Rasmus Lerdorf [33].



Figure 3. 9 : Logo de langage PHP [34].

2.4.2. JavaScript

JavaScript est un langage de script léger, orienté objet. Le langage JavaScript est interprété ou compilé à la volée, c'est un langage à objet utilisant le concept de prototype. Il dispose d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation [35].

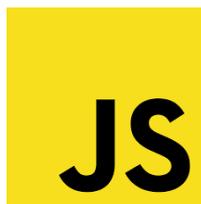


Figure 3. 10 : Logo de langage JAVASCRIPT [36].

2.4.3. HTML

HTML signifie Hyper Text Marckup Language, c'est un langage de balisage standard conçu pour la création des pages web [37].



Réalisation et présentation de l'application

Figure 3. 11 : Logo de langage HTML [38].

2.4.4. CSS

CSS (Cascading Style Sheets) signifie feuilles de style en cascade, c'est un langage utilisé pour décrire le style des pages HTML, il décrit comment les éléments doivent être affichés (couleurs, tailles, espacements, ...etc) [39].



Figure 3. 12 : Logo de langage CSS [40].

2.4.5. AJAX

Ajax est l'abréviation des Asynchrones JavaScript and XML, qui fait référence à un ensemble de techniques de développement Web plutôt qu'à un véritable langage de programmation. Cependant, Ajax est largement utilisé dans la programmation côté client (par exemple JavaScript) pour permettre l'envoi et la réception de données vers et depuis une base de données / un serveur. La particularité de l'utilisation de la programmation Ajax est que vous pouvez échanger des données en arrière-plan sans réellement perturber l'expérience utilisateur [41].



Figure 3. 13 : Logo de langage AJAX [42].

3. Présentation des interfaces

Dans cette section, nous allons présenter l'ensemble des interfaces graphiques de notre application web.

Réalisation et présentation de l'application

3.1. Page d'accueil

La page d'accueil est la première interface qui sera affichée lors du premier accès à notre application, elle est à accès public. Sur cette page, on peut :

- ✓ Effectuer une recherche directe d'un type de bien et de sa localisation.
- ✓ Se rediriger vers les pages suivantes : *publier une annonce, se connecter, inscription et consulter les annonces.*
- ✓ Changer la langue de l'application grâce à l'onglet qui se trouve en haut de la page à l'extrême droite.
- ✓ Etc.

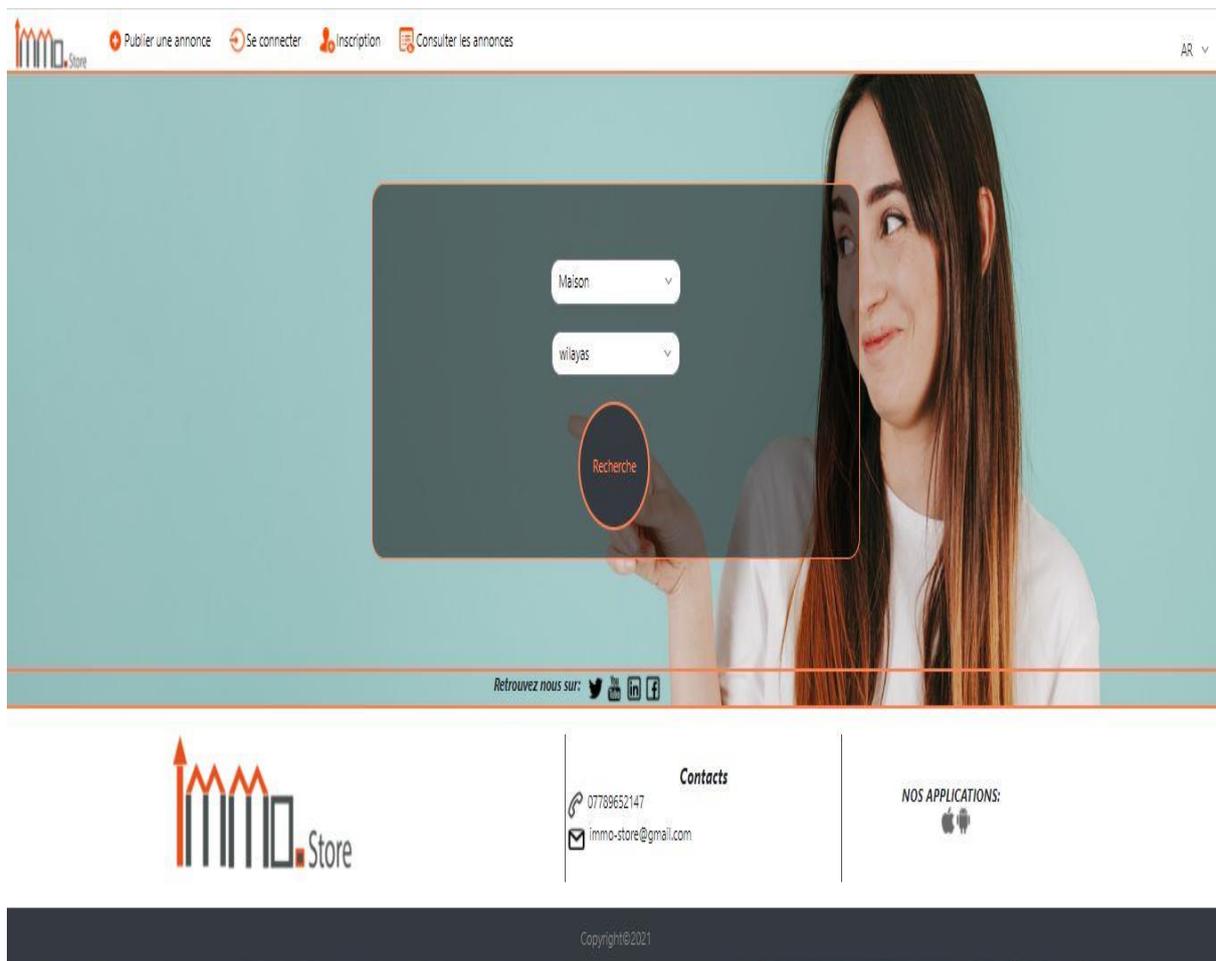


Figure 3. 14 : Interface page d'accueil.

3.2. Page d'inscription

Cette interface est consacrée à l'inscription à notre application pour les visiteurs qui souhaitent devenir membres. Ils doivent remplir des champs, dont certains sont obligatoires, afin de fournir des informations tels que : nom, e-mail, mot de passe, et numéro de registre de commerce s'il

Réalisation et présentation de l'application

a le statu de professionnel de l'immobilier. Cette interface est accessible via l'interface d'accueil en cliquant sur **Se connecter**.

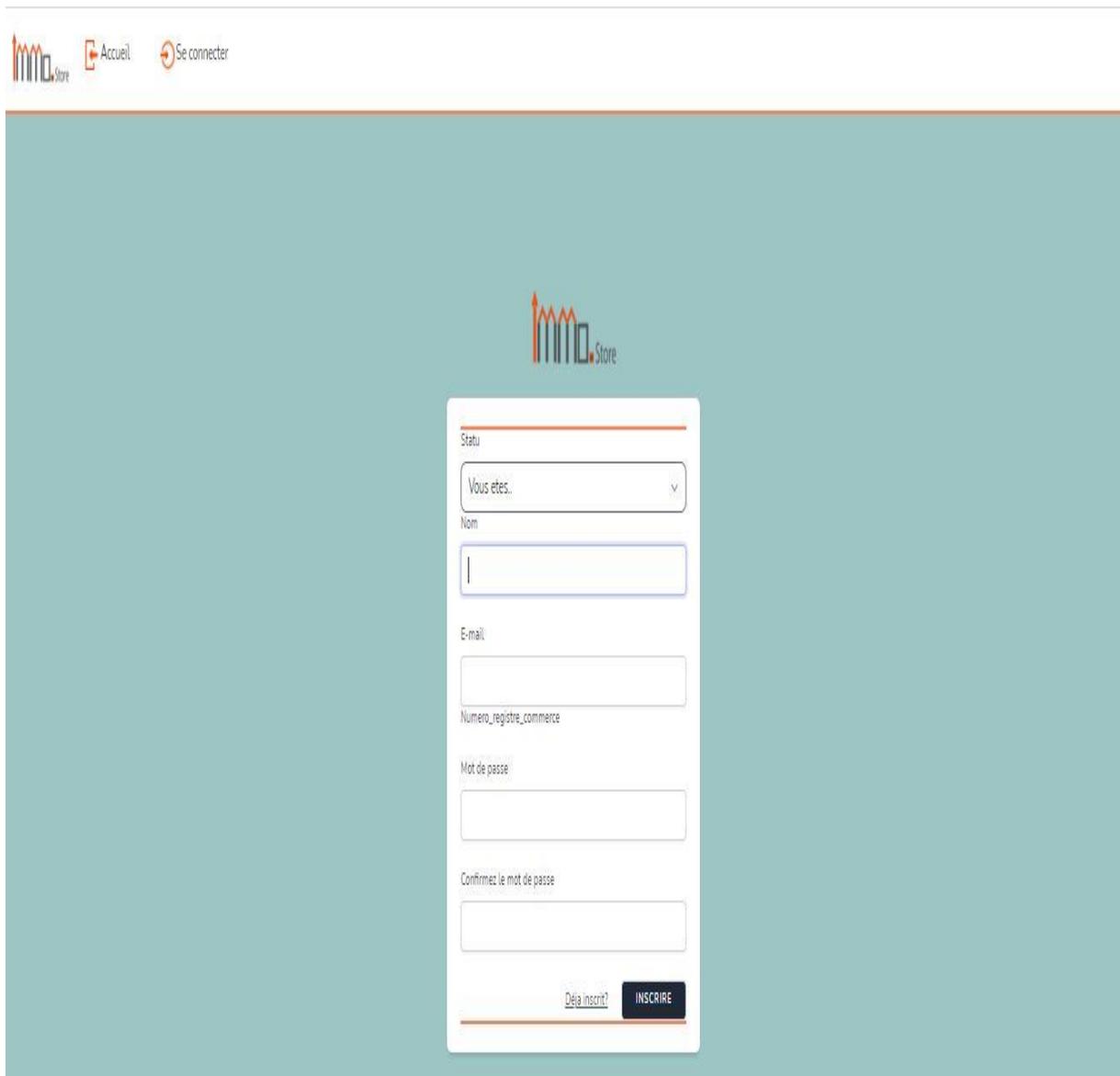


Figure 3. 15: Interface d'inscription.

3.3. Page Se connecter

La page se connecter, pour s'authentifier, concerne les personnes qui sont déjà inscrit dans notre application, c'est-à-dire qui sont déjà membres.

Réalisation et présentation de l'application

Sur cette page (voir la figure 3.16), il doit entrer son e-mail ainsi que son mot de passe. Si l'utilisateur ne se souvient pas de son mot de passe il clique sur le lien **Mot de passe oublier ?** pour qu'il reçoit un e-mail et procède à la création d'un nouveau mot de passe.

De plus via cette interface on peut revenir vers la page d'accueil en cliquant sur **Accueil** en haut comme il est possible de s'orienter vers la page d'inscription en cliquant sur le lien en bas **inscrivez-vous** pour les personnes qui ne sont pas encore membre.

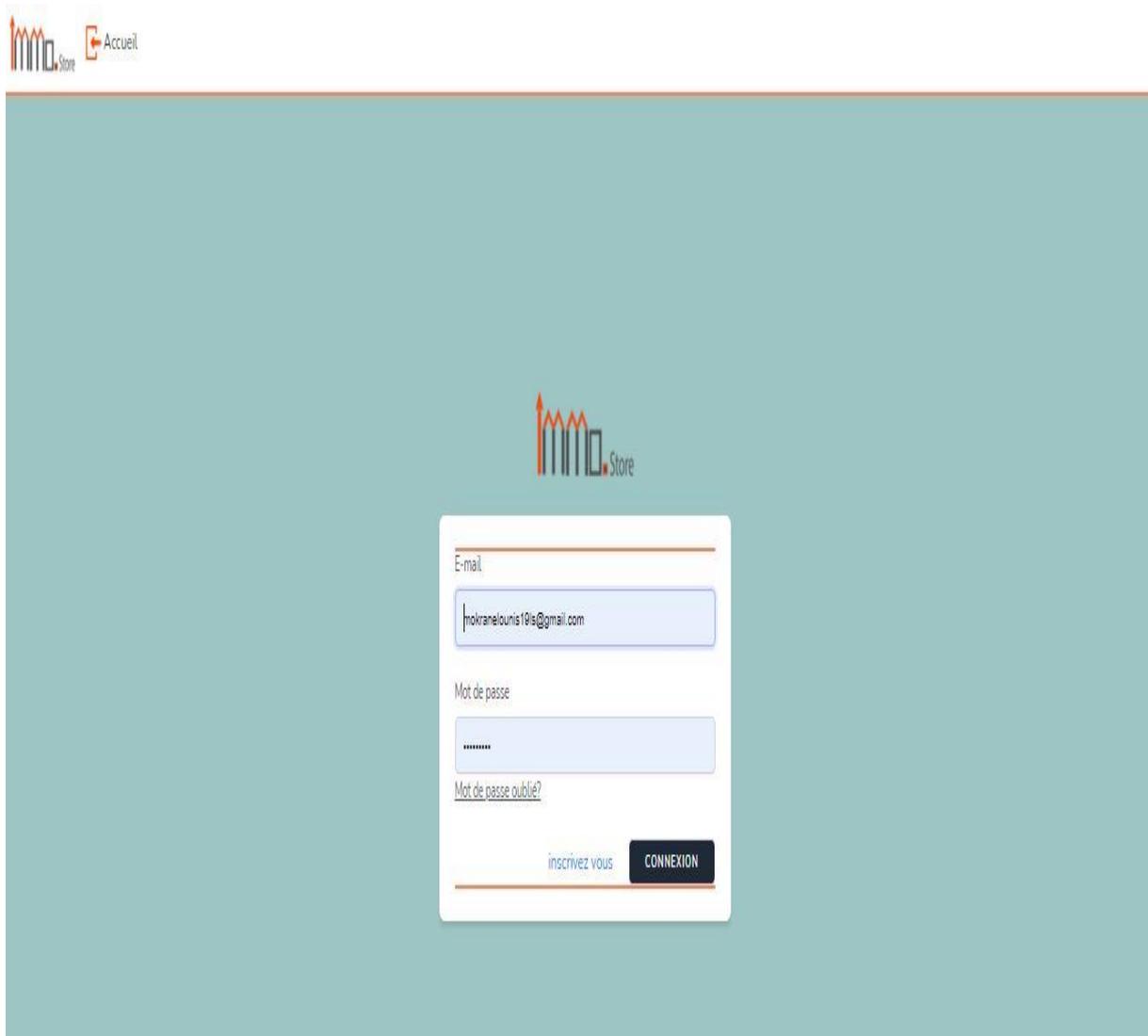


Figure 3. 16 : Interface Se connecter.

En cliquant sur connexion, on peut se retrouver dans l'un des deux cas suivants :

- ✓ E-mail ou mot de passe incorrecte.

Réalisation et présentation de l'application

- ✓ Dans le cas authentification effectuée elle va afficher un espace selon le statu membre ou administrateur.

Dans le cas membre :

Elle affiche l'espace membre comme le montre la figure 3.17, à partir de cette interface il peut :

- ✓ Revenir vers la page accueil.
- ✓ S'orienter vers formulaire de publication d'une annonce en cliquant sur **Publier une annonce**.
- ✓ Consulter l'ensemble des annonces qui la publié en cliquant sur **Mes annonces**.
- ✓ Consulter toutes les annonces disponibles sur l'application en cliquant sur **Consulter les annonces**.
- ✓ Consulter l'ensemble des notifications en cliquant sur l'icônes se forme de cloche et visiter chaque annonce concerné par le lien avec une simple clique sur le lien.
- ✓ Consulter et modifier son profile en accédant via l'anglet qui porte le nom de membre (sur la figure 3.17 c'est mokrane) qui se trouve en haut à droite.

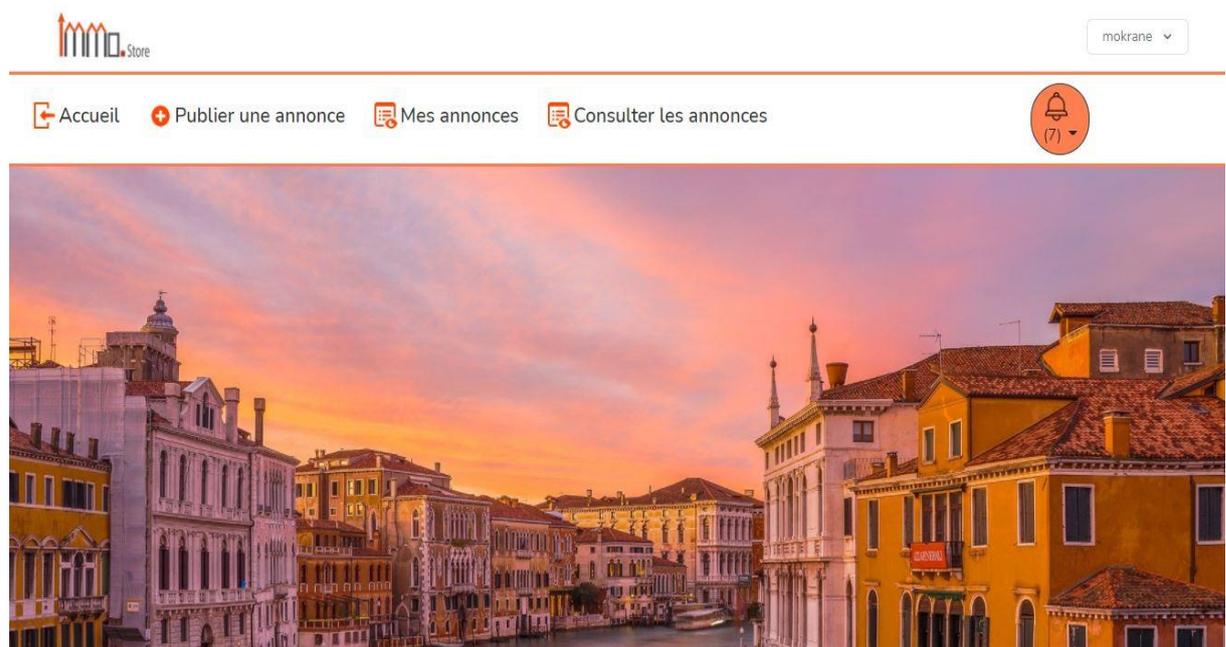


Figure 3. 17: Interface espace membre.

Dans le cas administrateur :

L'administrateur sur son espace peut comme le montre la figure 3.18 :

Réalisation et présentation de l'application

- ✓ Revenir vers la page accueil.
- ✓ Consulter et modifier son profil en cliquant en haut sur **Admin** (ce nom dépend du nom de l'administrateur).
- ✓ Consulter toutes les annonces disponibles sur l'application en cliquant sur **Consulter les annonces**.
- ✓ Consulter l'ensemble des annonces et membres et commentaires signalés et réagir avec une suppression.

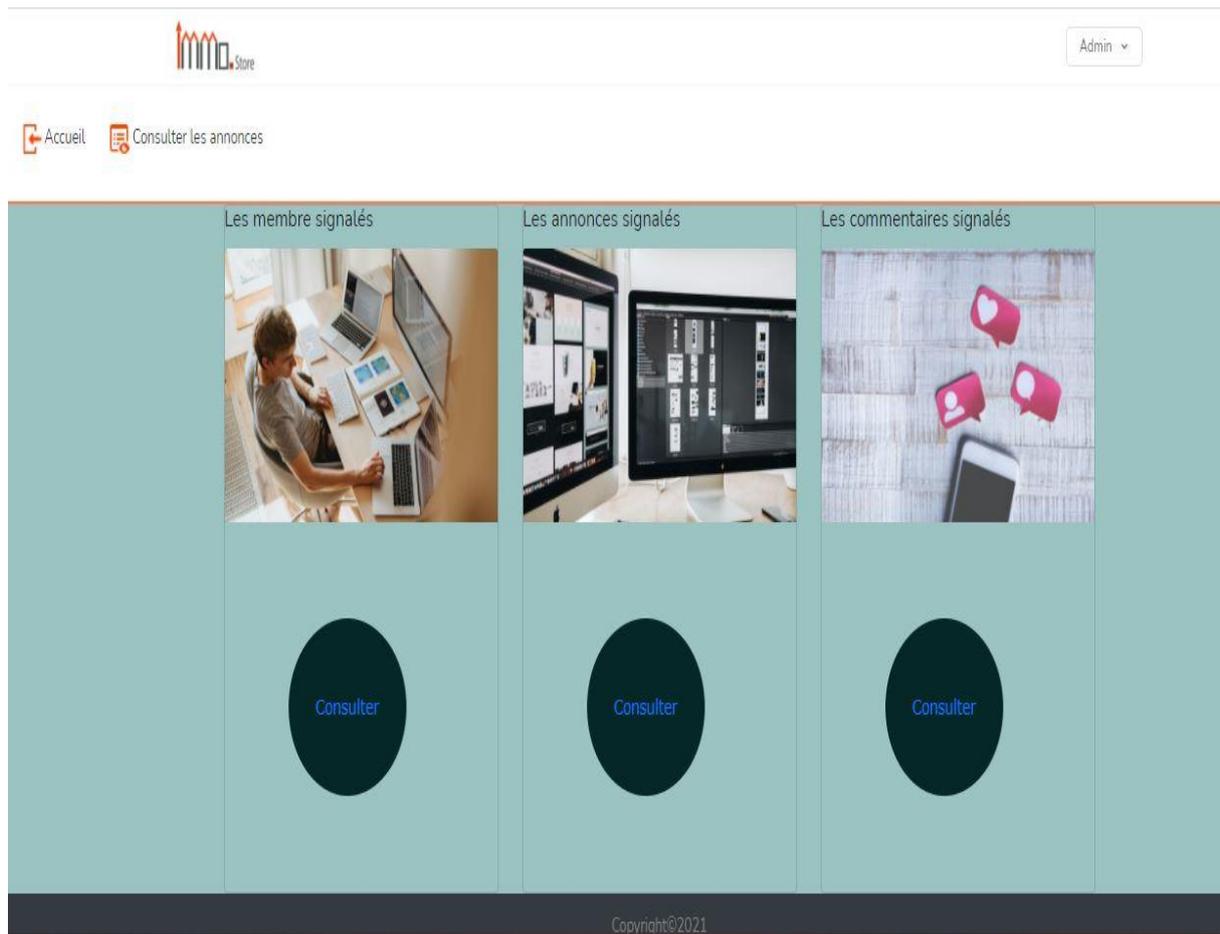


Figure 3. 18: Interface espace administrateur.

3.4. Page consulter les annonces

Cette page constitue (voir la figure 3.19) l'ensemble des publications disponibles sur l'application, accessible via la page d'accueil ou l'espace personnel (membre ou administrateur) ainsi que via la page **Mes annonces**.

On peut consulter les différentes publications de cette page sans une authentification préalable mais on ne peut pas réagir en commentant ou en signalant une annonce sans être authentifié.

Réalisation et présentation de l'application

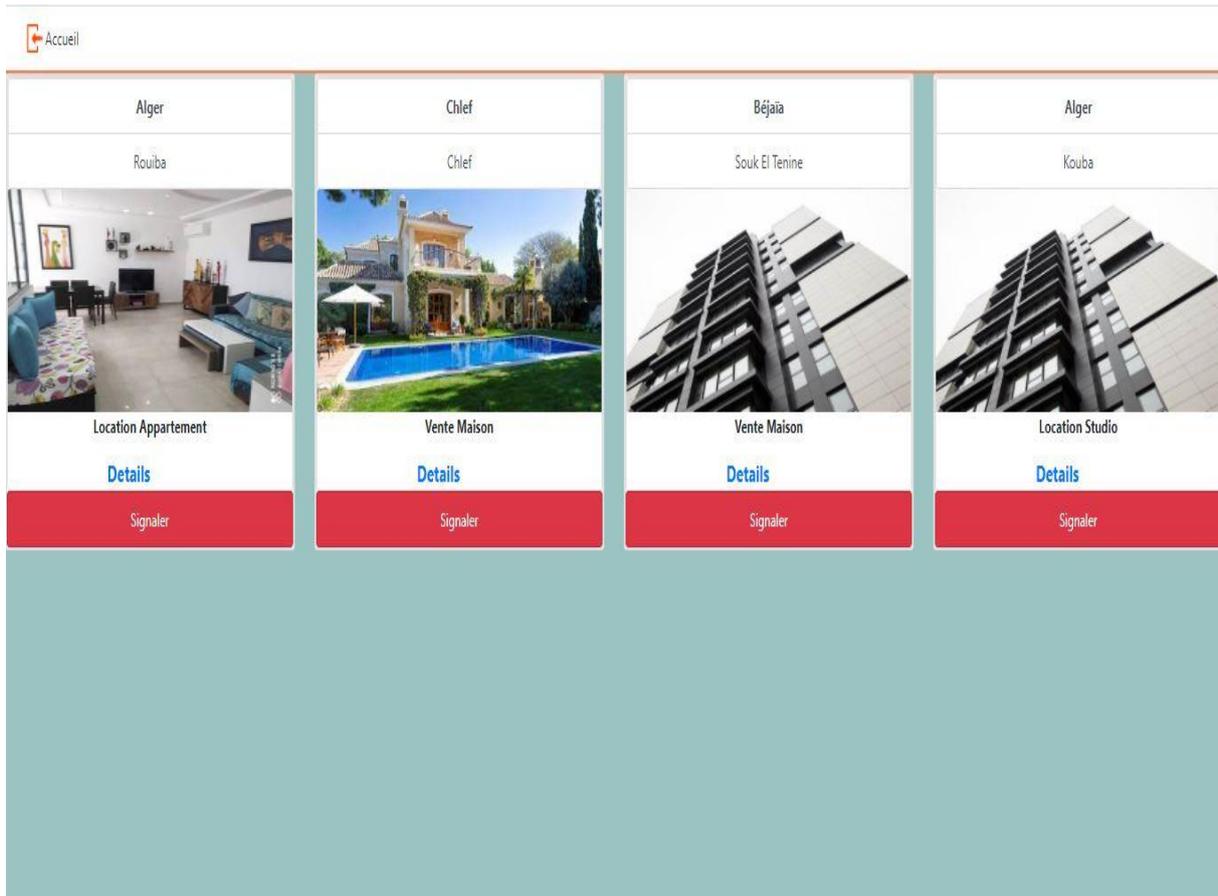


Figure 3. 19 : Interface consulter les annonces.

3.5. Page publier une annonce

Cette page constitue (voir la figure 3.20) un formulaire de publication d'une nouvelle annonce accessible seulement pour les membres via l'espace personnel ou via l'interface **Mes annonces**.

Si vous n'êtes pas authentifié et que vous tentez de publier une annonce, le système vous redirigera vers la page **Se connecter**.

Une fois sur la page **Publier une annonce**, on peut revenir vers l'accueil ou vers l'espace personnel.

Pour publier on doit remplir tous les champs et cliquer sur publier en bas. Une fois tous les champs bien remplis, on est redirigé vers notre espace personnel, et voir sa publication en cliquant sur **Mes annonces**.

Réalisation et présentation de l'application

The form is titled 'ACCUEIL' and 'mon compte'. It is divided into two main columns. The left column contains the following sections:

- Champs obligatoires (*)**: Three dropdown menus for 'Offre un bien', 'Vente', and 'Maison', each with a red asterisk.
- Emplacement**: Two dropdown menus for 'wilayas' and 'communes', each with a red asterisk.
- Papiers**: A dropdown menu for 'Acte notarié'.
- Spécifications**: A row of radio buttons for 'Meublé', 'Gaz', 'Eau', 'Parking', 'Jardin', and 'Electricité'.

The right column contains the following sections:

- Details**: Three input fields for 'Superficiel', 'Etage', and 'Nombre de chambre', with a red asterisk on the first.
- Informations Complémentaire**: A large text area for 'discription...' with a red asterisk.
- Contacts**: A file selection field 'Sélect. fichiers' with 'Aucun fichier choisi' and a red asterisk, and a 'telephone' input field with a red asterisk.

A 'Publier' button is located at the bottom center of the form.

Figure 3. 20 : Formulaire de publication d'une nouvelle annonce.

3.6. Espace membre montrant l'arrivé de notifications

Les notifications sont disponibles sur l'espace personnel, accessible après une authentification.

On peut les consulter avec un clique sur l'icône sous forme d'une cloche, où on voit apparaître un journal de notification, comme le montre la figure 3.21.

Pour consulter la publication concernée par la notification arrivée, on clique sur le lien du journal des notifications.

Réalisation et présentation de l'application

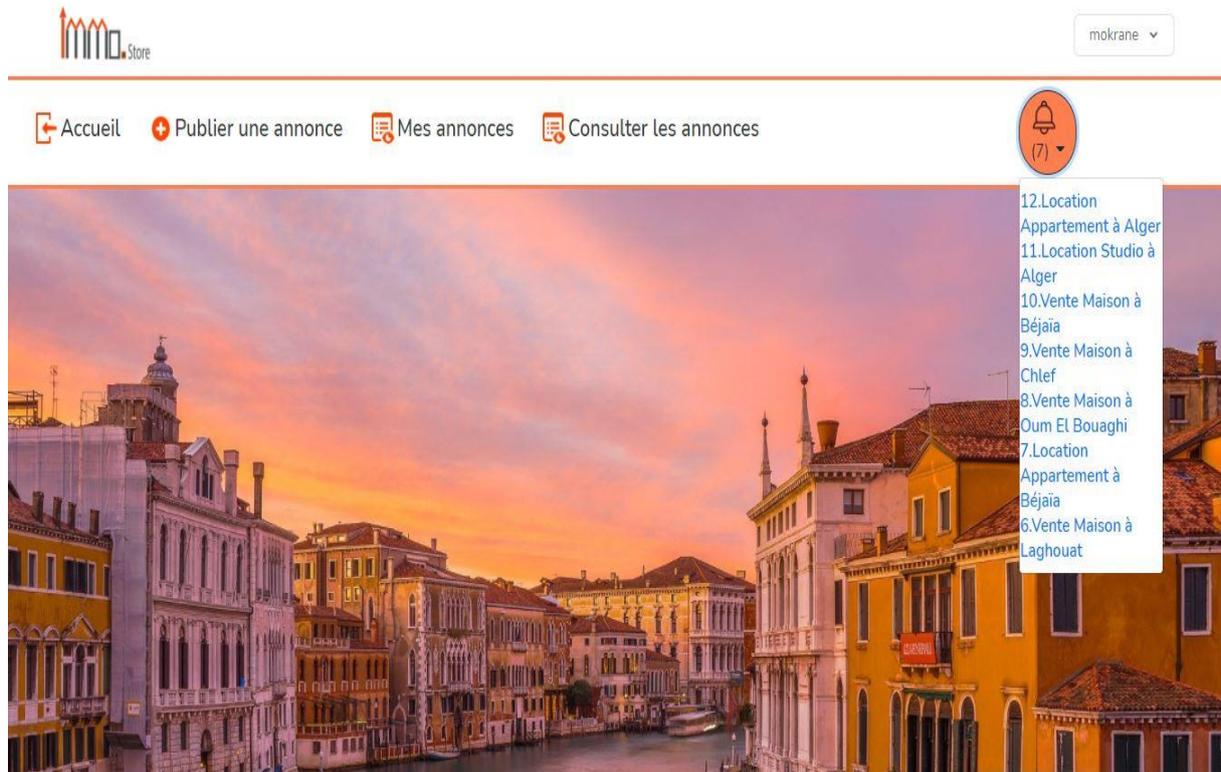


Figure 3. 21: Espace membre avec la présence de nouvelles notifications.

3.7. Page Mes annonces

Cette page (voir la figure 3.22) regroupe l'ensemble des annonces publiées par un membre. Elle est accessible via son espace personnel.

A travers cette page, il peut :

- ✓ Revenir sur son espace personnel en cliquant sur **Mon compte** qui se trouve en haut à gauche.
- ✓ Voir l'ensemble de ses annonces cliquant sur **Mes annonces**.
- ✓ Supprimer l'une de ses annonces.
- ✓ Modifier les informations de ses annonces.
- ✓ Commenter son annonce en cliquant sur le lien **Details**.

Réalisation et présentation de l'application

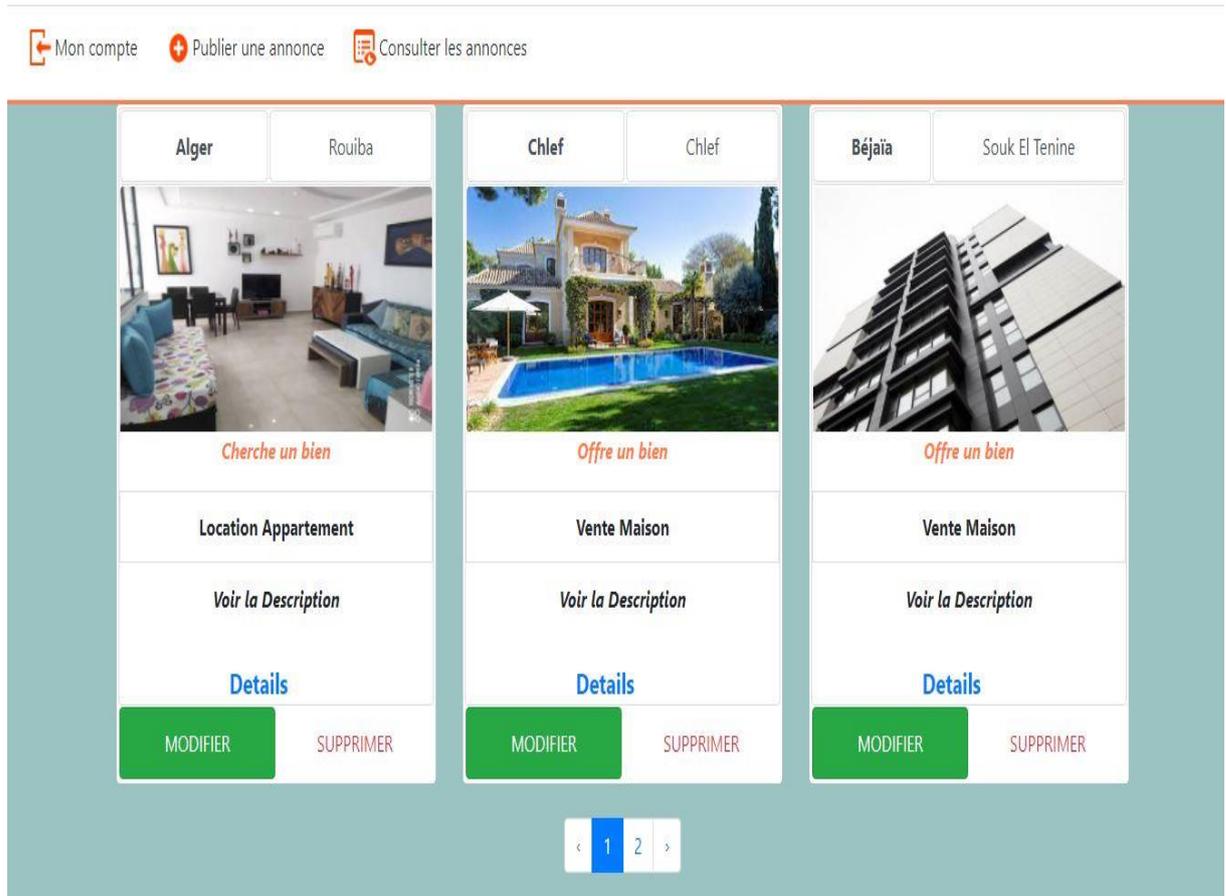


Figure 3. 22 : Page Mes annonces.

3.8. Formulaire de modification d'une annonce

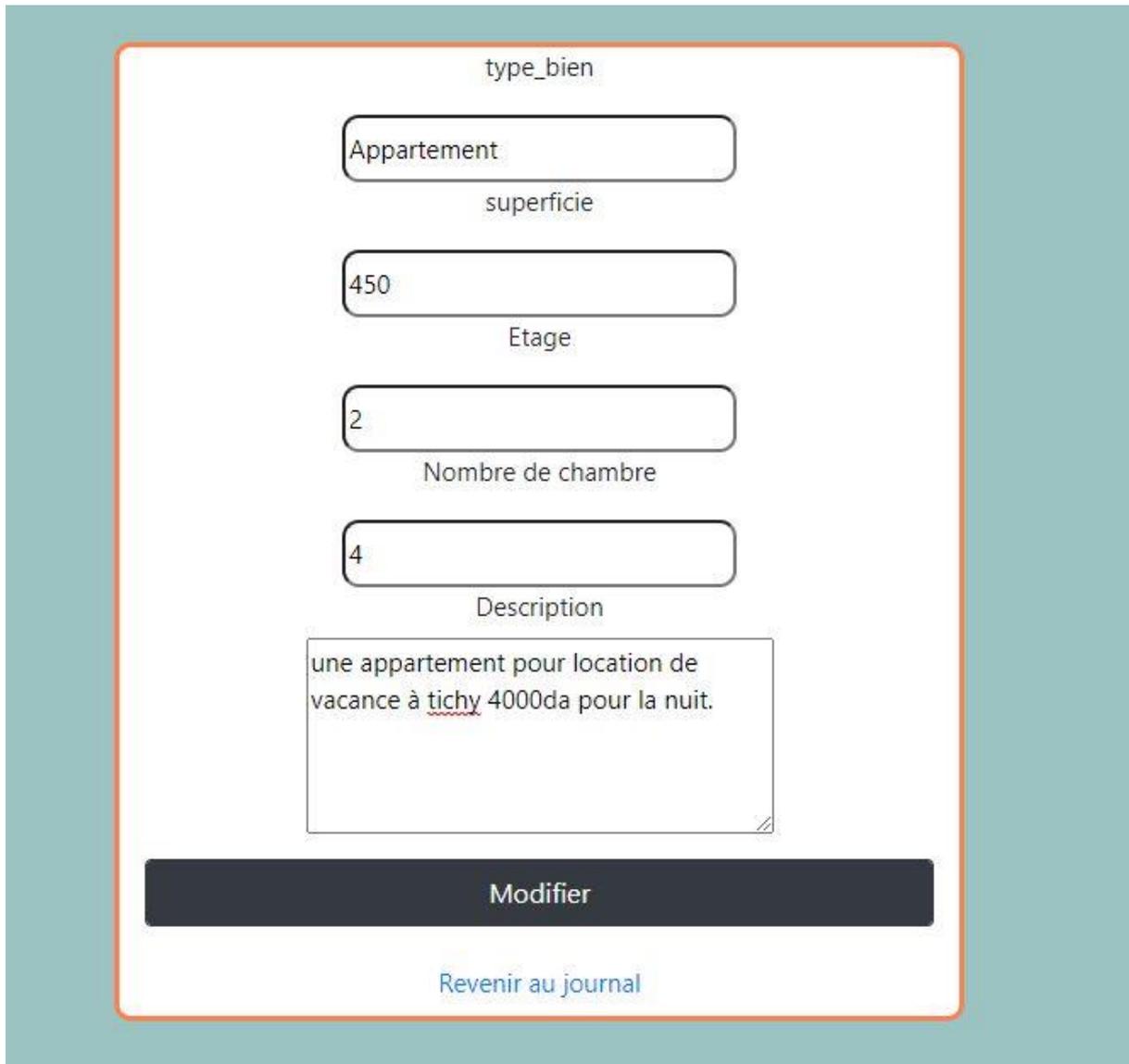
Ce formulaire (figure 3.23) a pour but de donner la main à un membre de modifier des informations d'une annonce qu'il a déjà publiée, comme par exemple : sa superficie, sa description, etc.

Ce formulaire est accessible via la page **Mes annonces** après un clique sur bouton **MODIFIER**.

Une fois le formulaire affiché on peut :

- ✓ Saisir les modifications souhaitées et cliquer sur modifier.
- ✓ Revenir sur la page **Mes annonces** en cliquant sur le lien en bas revenir au journal.

Réalisation et présentation de l'application



type_bien

Appartement

superficie

450

Etage

2

Nombre de chambre

4

Description

une appartement pour location de vacance à tichy 4000da pour la nuit.

Modifier

[Revenir au journal](#)

Figure 3. 23 : Formulaire de modification d'une annonce.

4. Conclusion

Dans ce chapitre, nous avons présenté les différents outils de développement que nous avons utilisé pour la réalisation de notre application notamment le Framework Laravel qui nous a offert une bonne persistance de données grâce à l'ORM Eloquent.

Après nous avons montré les résultats de réalisation représentant notre application, en illustrant les différentes interfaces graphiques de notre application.

Conclusion générale et perspectives

Conclusion générale et perspectives

Ce travail a été réalisé dans le cadre de notre projet de fin de cycle master en génie logiciel.

Il consiste en la conception et réalisation d'une application web pour la publication d'annonces immobilière en Algérie. Cette application est destinée aux particuliers et aux professionnels de l'immobilier afin d'atteindre une amélioration importante en ce qui concerne la vente et l'achat, l'échange et la location de biens immobiliers.

La réalisation de ce projet a commencé par une large étude du marché algérien exactement pointée sur les applications dédiées aux publications d'annonces immobilières et à leurs analyses pour en tirer notre point de vue, et de dégager les besoins de notre application.

Après avoir dégagé les besoins de notre application, nous avons fait notre choix pour l'UML comme langage de modélisation, afin qu'on représente les différents diagrammes (diagrammes de cas d'utilisation, diagrammes de séquences, diagrammes de classe) de notre application.

Après avoir terminé la partie conception, nous avons entamé la partie réalisation de l'application. Avant de commencer cette dernière, nous avons spécifié l'environnement de développement, pour le côté frontal (front-end en anglais) nous avons utilisé HTML, CSS, BOOTSTRAPS et JS. Pour tous ce qui concerne la programmation d'arrière-plan des interfaces (back-end en anglais), nous avons utilisé le langage de programmation PHP et le Framework Laravel, sans oublier la base de données qui nous avons implémenté sur le SGBD WAMPSEVER.

Ce projet représente une occasion pour mettre en pratique nos connaissances dans le domaine et exploiter de nouvelle technologies, et de tirer de plus surtout ce qui concerne l'analyse.

A la fin de ce travail, nous avons conscience qu'il y'a toujours une possibilité d'améliorer et de continuer le développement de cette application notamment :

- ✓ Activer la fonctionnalité de signalisation d'un membre et d'annonce ainsi que les commentaires.
- ✓ Rajouter un système de messagerie intégré dans notre application pour une communication directe via l'application.
- ✓ Rajouter un tableau graphique représentant le nombre de visiteur de l'application.

Conclusion générale et perspectives

- ✓ Inclure l'option qui compte le nombre de vues pour chaque annonce.
- ✓ Introduire l'option de 'j'aime' pour chaque annonce.
- ✓ Mettre en place l'API Google chrome pour un meilleur géolocalisation.

Bibliographie

- [1] <https://www.similarweb.com> (consulté le 25/02/2021)
- [2] <https://www.syloe.com/glossaire/application-web/>(consulté le 15/04/202).
- [3] <https://www.ouedkniss.com>(consulté le 25/02/2021)
- [4] <https://www.lkeria.com>(consulté le 25/02/2021)
- [5] <https://nbatou.com>(consulté le 26/02/2021).
- [6] <https://darna.com> (consulté le 28/02/2021)
- [7] <https://darjadida.com>(consulté le 28/02/2021)
- [8] <https://www.essekna.com/>(consulté le 02/03/2021).
- [9] <https://www.algeriahome.com>(consulté le 02/03/2021).
- [10] <https://www.algerieannonces.com>(consulté le 04/03/2021).
- [11] <https://www.similarweb.com/fr/website/oudkniss.com/> (consulté le 07/09/2021).
- [12] <https://www.similarweb.com/fr/website/lkeria.com/> (consulté le 07/09/2021).
- [13] <https://www.ouedkniss.com/immobilier>(consulté le 23/02/2021)
- [14] <https://www.adobe.com/fr/products/illustrator.html> (consulté le 30/07/2021)
- [15] <http://dSPACE.univ-tlemcen.dz/bitstream/112/5500/5/chapitre1.pdf>
- [16] P. Roques, UML 2 Modéliser une Application Web, 4e édition, Eyrolles, Paris, 2008
- [17] <http://projet.eu.org/pedago/sin/term/3-UML.pdf>
- [18] http://info.arqendra.net/Files/_UML_cours.pdf(consulté le 15/04/2021)
- [19] P. Roques & Frank Vallée, UML 2 en action de l'analyse des besoins à la conception, 4e Eyrolles, 2007
- [20] <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/#questce-que-laravel->
(consulté le 03/06/2021)
- [21] <https://laravel.com/> (consulté le 20/08/2021)

Bibliographie

- [22] https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
(consulté le 10/07/2021)
- [24] https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp(consulté le 04/06/2021)
- [25] <https://laravel.com/docs/8.x/eloquent> (consulté le 22/04/2021)
- [24] <https://getbootstrap.com/> (consulté le 14/08/2021)
- [26] <https://www.mysql.com/fr/>(consulté le 06/05/2021)
- [27] <https://www.wampserver.com/> (consulté le 24/04/2021)
- [28] <https://code.visualstudio.com/docs>(consulté le 01/07/2021)
- [29] <https://visualstudio.microsoft.com/> (consulté le 24/05/2021)
- [28] https://fr.wikipedia.org/wiki/Adobe_Illustrator(consulté le 11/07/2021).
- [29] <https://www.tice-education.fr/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>(consulté le 15/06/2021)
- [30] <https://drawio-app.com/> (consulté le 22/09/2021)
- [31] <https://www.postman.com/> (consulté le 20/09/2021)
- [32] <https://arabicprogrammer.com/article/8979228350/>(consulté le 09/06/2021)
- [33] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/>(consulté le 10/07/2021)
- [34] <https://www.php.net/> (consulté le 28/09/2021)
- [37] <https://www.javascript.com/> (consulté le 10/09/2021)
- [38] https://www.w3schools.com/html/html_intro.asp(consulté le 03/06/2021)
- [39] <https://www.w3.org/html/logo/> (consulté le 14/09/2021)
- [40] https://www.w3schools.com/css/css_intro.asp(consulté le 03/06/2021)
- [41] https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg (consulté le 20/09/2021)

Bibliographie

[42] <https://www.keycdn.com/support/ajax-programming>(consulté le 25/05/2021)

[43] <https://brandeps.com/logo/A/AJAX-Language-01> (consulté le 21/09/2021)

Résumé

Ce document est rédigé en vue de l'obtention du diplôme de master génie logiciel.

Le projet que nous avons réalisé consiste à une application web pour la publication d'annonces immobilières en Algérie, dédiée aux particuliers et aux professionnels de l'immobilier. Notre application se nomme **immo.store** En plus d'être gratuite et facile à utiliser, elle intègre une fonctionnalité, que l'on ne retrouve pas dans d'autres applications similaires, et qui est la notification des nouvelles annonces.

Pour la création de cette application, nous avons utilisé le processus de développement PU et le langage de modélisation UML pour la conception, et pour la réalisation nous avons opté pour le Framework Laravel, qui est un Framework PHP. Nous avons utilisé pour le coté frontal (front-end en anglais) le HTML, CSS, BOOTSTRAPS, JS et pour la programmation des arrières plans des interfaces graphiques (back-end en anglais) le langage PHP ainsi que le MYSQL pour les bases de données.

Mots clés : application web, PU, UML, Laraval, PHP, HTML, CSS, BOOTSTRAPS, JS, MYSQL.

Abstract

This document is written with a view to obtaining the Master's degree in Software Engineering. The project that we developed consisted of a web application for the publication of real estate ads in Algeria, dedicated to individuals and real estate professionals. Our application is named immo.store, besides being free and easy to use, it also integrates a functionality that cannot be found in other similar applications including the notification of new announcements.

To create this application we used the UP developement process and the UML modeling language for the design, for the realization we opted for the Laravel Framework which is a php framwork.we used the HTML, CSS, BOOTSTRAPS, JS for the front-end , for the back-end.We used the php language and MYSQL for the database.

Keys words: web application, UP, UML, Laravl, PHP, HTML,CSS, BOOTSTRAPS, JS, MYSQL.