



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique
جامعة محمد بوضياف بالمسيلة
Université Mohamed Boudiaf de M'sila



كلية الرياضيات والإعلام الآلي
Faculté des Mathématiques et de l'Informatique

قسم الإعلام الآلي
Département de l'Informatique

Domaine: Mathématiques et Informatique

Une Thèse Présentée pour l'Obtention du diplôme
de Master en Informatique

Specialté: RTIC – Réseaux et Technologies de
l'Information et de la Communication

Réalisée Par: ARIOUA KHAYREDDINE , CHOUDAR ELBACHIR

Dirigée Par:

BOUNIF MOHAMMED

TITRE

Conception et réalisation d'un site web d'allocation immobilière

Membres de Jury

Prénom 1 Nom 1

Président

Prénom 2 Nom 2

Encadreur

Prénom 3 Nom 3

Examineur

Année Universitaire 2024/2025

Remerciements

Nous exprimons d'abord notre profonde gratitude envers Dieu, qui nous a guidés, accordé la patience et insufflé le courage tout au long de notre parcours académique. Nous remercions chaleureusement notre encadreur pour le temps qu'il nous a consacré, la richesse de ses conseils, ainsi que son soutien constant tout au long de la réalisation de ce travail. Nos remerciements vont également à l'ensemble des membres du jury, que nous remercions sincèrement pour l'honneur qu'ils nous ont fait en acceptant d'évaluer ce travail. Enfin, nous adressons nos remerciements les plus sincères à nos parents, ainsi qu'à tous ceux – famille et amis – qui nous ont encouragés et soutenus. Nous espérons que chaque lecteur de ce mémoire y trouvera le reflet de notre reconnaissance.

Table de Matière

Remerciements	
Table des Matière	
Liste des Figures	I
Introduction Générale.....	2
Chapitre 1 :Site Web et Architecture Client/Serveur.....	4
I.1. Introduction.....	4
I.2. Notion du site web	4
I.2.1. Le web	4
I.2.1.1. Définition du web	4
I.2.1.2. L'évolution du web.....	4
I.2.2. Le site web.....	5
I.2.2.1. Définition du site web.....	5
I.2.2.2. Types des sites web	5
I.3. Architecture client/serveur	5
I.3.1. Définition du l'architecture client/serveur	5
I.3.2. Fonctionnement du l'architecture client/serveur.....	6
I.3.3. Caractéristiques de l'architecture client/serveur	6
I.3.4. Type d'architectures client/serveur	7
I.3.5. Avantages et inconvénients de l'architecture.....	9
I.3.5.1. Avantages	9
I.3.5.2. Inconvénients.....	9
I.4. Conclusion :	10
Chapitre 2: E-commerce	11
II.1.Introduction	11
II 2. Définition et types de l'e-commerce :.....	11
II .2.1. Définition :	11
II.2.2 Types of E-Commerce: E-Commerce - Business Models	12
II 2.2.1. Entreprise à Entreprise (E2E).....	12
II 2.2.2. Entreprise à Consommateur (E2C).....	12
II 2.2.3. Consommateur à Consommateur (C2C).....	12
II 2.2.4. Consommateur à Entreprise (C2E).....	12
II 2.2.5. Entreprise à Gouvernement (E2G)	13

II .2.2.6.Gouvernement à Entreprise (G2E)	13
II .2.2.7Gouvernement à Citoyen (G2C).....	13
II.3. Composants du Commerce Électronique.....	15
II.4. Sécurité.....	17
II.6. Traitement des paiements	18
II.7.SET.....	18
II.8.COMPTE MARCHAND INTERNET	19
II.9.Conclusion :	19
Chapitre 3: Analyse et Conception.....	20
III.1. Introduction :	20
III.2. Présentation de l’UML :	20
III.2.1. Définition :	20
III.2.2. L’utilité de l’UML :.....	21
III.4. Identification des acteurs	21
III.5.1. Le visiteur :.....	22
III.5.2. Le Client :.....	22
III.5.3. Administrateur:.....	22
III.6. Identification des messages échangés :	22
III.7. Diagramme de contexte	23
III.8. Capture des besoins :	24
III.8.1. Identification des cas d'utilisations :.....	25
III.9. Description textuelle des cas d'utilisations.....	25
III.9.1. Description textuelle de cas d'utilisation «Effectuer une recherche ».....	26
III.9.2. Description textuelle de cas d'utilisation «Modifier son compte »	26
III.9.3. Description textuelle de cas d'utilisation « S'authentifie»	27
III.9.4. Description textuelle de cas d'utilisation « Gestion de biens immobiliers»	28
III.10. Diagramme de cas d'utilisation	29
III.10.1. Le diagramme de cas d'utilisation « interface visiteur » :... خطأ! الإشارة المرجعية غير معرفة.	
III.10.2. Le diagramme de cas d'utilisation « interface client »..... خطأ! الإشارة المرجعية غير معرفة.	
III.10.3. Le diagramme de cas d'utilisation global (E-commerce).....	29
III.11. Diagramme de séquence :	30
III.11.1. Diagramme de séquence du cas d'utilisation << Effectuer une recherche >>.....	30
III.11.2. Diagramme de séquence << S'authentifier>>	30
III.11.3. Diagramme de séquence du << Supprimer un compte >>.....	31
III.12. Diagramme d'activité	32
III.12.1. Les diagrammes d’activité utilisés :.....	33
III.12.1.1 Diagramme d'activité <<inscription >>	33
III.12.1.2. Diagramme d'activité <<Finaliser un achat>>	35
III.13. Diagramme de classe :	35

Chapitre 4: Realsation	37
IV.1. Introduction	37
IV.2.L'environnement de développement :	38
IV.2.1.1.L'environnement matériel :	38
IV.2.1.2.Environment Logiciel :	38
IV.2.1.3 - Bases de Données et Outils Associés	39
IV.2.1.4- Développement Web (Frontend et Backend)	39
IV.2.1.5- Sécurité et Authentification	40
IV.2.1.6- Services Cloud et Intégration	40
IV.3. Les principales interfaces graphiques	41
IV.3.1. Interface principale.....	41
IV.3.2. Interface des state	42
IV.4.3. Interface login	43
IV.5.Conclusion :	44
Conclusion Générale	45
Bibliographie.....	47

List des Figures

Figure I.1 : Architecture peer to peer	7
Figure I.2 : Architecture à 2 niveaux	8
Figure I.3 : Architecture à 3 niveaux	8
Figure I.4 : Architecture à N niveaux	9

List des Tableaux

Tableau.III. 1. Description des acteurs -----	21
Tableau.III. 2 Identification des messages échangés -----	23
Tableau.III. 3 Identification des cas d'utilisation -----	25
Tableau.III. 4 Description textuelle du cas d'utilisation « Effectuer une recherche » -----	26
Tableau.III. 5 Description textuelle du cas d'utilisation « Modifier son compte » -----	26
Tableau.III. 6 Description textuelle du cas d'utilisation « S'authentifier» -----	27
Tableau.III. 7 Description textuelle du cas d'utilisation Gestion de produit -----	28

Introduction Générale

La transition numérique, renforcée par l'essor des technologies de l'information et de la communication, a profondément modifié les dynamiques économiques et sociales à l'échelle mondiale. Parmi les secteurs les plus impactés par cette transformation figure celui de l'immobilier, qui connaît une mutation progressive vers des pratiques digitalisées, notamment en matière de location de biens. L'émergence de plateformes web spécialisées dans l'allocation immobilière s'inscrit dans cette logique de modernisation, en proposant des solutions innovantes visant à faciliter la mise en relation entre bailleurs et locataires, à optimiser les processus transactionnels, et à améliorer l'accessibilité à l'information.

En Algérie, cette tendance s'est accentuée à la faveur de la généralisation de l'usage des smartphones et de l'amélioration de la connectivité Internet, deux éléments qui favorisent l'appropriation des services numériques par une part croissante de la population. Selon les données officielles du Ministère de la Numérisation (2023), plus de 60 % des citoyens accèdent régulièrement aux services en ligne, ce qui modifie en profondeur leurs comportements de recherche, y compris dans le domaine du logement locatif. Toutefois, malgré ce potentiel prometteur, le marché local demeure caractérisé par un déficit en plateformes digitales adaptées aux spécificités culturelles, linguistiques et socio-économiques du pays.

Face à cette réalité, la présente étude vise à concevoir et à réaliser un site web dédié à l'allocation immobilière, en tenant compte à la fois des besoins des usagers, des contraintes techniques, et des exigences d'ergonomie et de sécurité. Il s'agit de proposer une solution numérique intégrée, capable de répondre efficacement aux attentes des propriétaires comme des locataires, tout en s'inscrivant dans une perspective de développement durable et de transformation numérique du secteur immobilier en Algérie.

Ce travail se propose ainsi d'articuler une réflexion théorique sur les enjeux de la digitalisation de l'immobilier avec une démarche pratique de développement web, afin d'explorer les apports, les limites et les perspectives d'un tel projet. À travers cette double approche, la recherche ambitionne de contribuer à l'enrichissement des connaissances sur l'intégration des technologies numériques dans les services immobiliers, tout en fournissant une base concrète pour la conception d'outils adaptés aux réalités du marché local.

Ce mémoire est structuré en quatre chapitres principaux qui abordent de manière progressive les aspects théoriques et pratiques du sujet.

- Le **premier chapitre**, intitulé « *Site web et architecture client/serveur* », présente les notions fondamentales du web et retrace son évolution. Il aborde ensuite la définition du site web et ses différents types, avant de se concentrer sur l'architecture client/serveur en exposant son fonctionnement, ses caractéristiques, ses variantes, ainsi que ses avantages et inconvénients.
- Le **deuxième chapitre** est consacré à *l'e-commerce*. Il en propose une définition claire, puis explore les différents modèles commerciaux existants (B2B, B2C, C2C, etc.). Le chapitre traite également des composants du commerce électronique, des enjeux liés à la sécurité, des mécanismes de paiement en ligne, du protocole SET et des comptes marchands sur Internet.
- Le **troisième chapitre** s'intéresse à *l'analyse et à la conception* du système en utilisant le langage UML. Il décrit les différents acteurs (visiteur, client, administrateur), identifie les cas d'utilisation et les messages échangés, puis les illustre à travers des diagrammes de contexte, de séquence, d'activités et de classes afin de modéliser le fonctionnement du système de manière détaillée.
- Le **quatrième chapitre** est dédié à *la réalisation de l'application web*. Il détaille l'environnement matériel et logiciel utilisé, les outils de développement web (frontend et backend), les mesures de sécurité mises en place, ainsi que l'intégration de services cloud. Enfin, ce chapitre présente les principales interfaces graphiques de l'application.

Une conclusion générale vient clore ce travail, en récapitulant les apports de l'étude et en proposant des pistes d'amélioration ou de perspectives futures.

CHAPITRE 1

Site web et architecture client/serveur

I.1. Introduction

Au cours des dernières années, compte tenu du progrès et de la présence d'Internet dans nos vies, de nombreuses entreprises ont mis en place et créé leurs propres sites Web. Ces sites permettent aux entreprises de se faire connaître dans le monde, d'attirer de nouveaux clients et donc d'améliorer leurs résultats économiques. Cependant, pour pouvoir naviguer sur ces sites, ils utilisent un modèle d'échange de données nommé 'Client/Serveur', le navigateur étant le client dans cette architecture. Dans ce chapitre, nous donnons un aperçu sur les notions de sites web et d'architecture Client/serveur. [1]

I.2. Notion du site web

I.2.1. Le web

I.2.1.1. Définition du web

Web en anglais, "la toile" désigne Internet et fait référence aux réseaux câblés qui parcourent le monde et relie les ordinateurs entre eux à la manière d'une toile d'araignée. Surfer sur le Web revient à dire : naviguer sur Internet.

I.2.1.2. L'évolution du web

Le World Wide Web, connu sous le nom de « www », a été lancé pour la première fois en 1991. Mais comme la technologie a avancé, de nouvelles versions sous forme de Web 2.0 et Web 3.0 ont été apparues. Web 2.0 et Web 3.0 sont évidemment considérés comme plus avancés et faciles à utiliser par rapport au Web 1.0. Mais la plupart d'entre nous sont confus quant aux changements et à l'évolution qui ont eu lieu.

A. Web 1.0 : Web 1.0 est simplement un portail d'informations où les utilisateurs reçoivent passivement des informations sans avoir la possibilité de publier des commentaires ou des réactions.

B. Web 2.0 : Le Web 2.0 encourage la participation, la collaboration et le partage d'informations. Youtube, Wiki, Flickr, Facebook, etc., sont des exemples d'applications Web 2.0.

C. Web 3.0 : Le Web 3.0 est un Web sémantique qui fait référence au futur. Dans le Web 3.0, les ordinateurs peuvent interpréter les informations comme des êtres humains et générer et distribuer intelligemment un contenu utile adapté aux besoins des utilisateurs. Tivo, un enregistreur vidéo numérique, est un exemple de Web 3.0. Son programme d'enregistrement peut rechercher sur le Web et lire ce qu'il vous trouve en fonction de vos préférences.

I.2.2. Le site web

I.2.2.1. Définition du site web

Un [site web](#) est l'ensemble des pages web et des ressources inter reliées entre elles par des liens hypertextes, auxquelles l'internaute peut accéder par une adresse web appelée Url, le tout enregistré sous le même nom de domaine. Un site web ou [site internet](#) englobe des textes et multimédia, il est hébergé sur un serveur web, auquel on peut accéder à travers le réseau internet ou intranet.

Une page web peut être défini comme étant l'ensemble des documents html structurés, stockés sur un serveur connecté au réseau mondial qu'est internet, cette dernière contient pour son essentiel du texte, que l'on enrichit et accompagne d'images, de vidéos animation, de son parfois et de liens reliant la présente page à d'autres pages web.

I.2.2.2. Types des sites web

On distingue deux types de sites : les sites statiques et les sites dynamiques.

A. Les sites statiques : sont des sites qui s'affichent tel qu'ils ont été codés. Ils manipulent des pages dont les contenus sont prédéterminés. En outre, les visiteurs peuvent seulement voir le contenu du site mais pas y participer. Pour les réaliser, seuls les langages dits d'interface utilisateur (frontend) sont nécessaires, c'est-à-dire HTML, [CSS](#) et [JavaScript](#) (en théorie, il est toutefois possible d'utiliser uniquement le HTML).

B. Les sites dynamiques : sont des sites qui interagissent avec le visiteur. Ses pages sont programmées pour être modifiables. Elles génèrent des contenus au moment de solliciter des informations à un serveur web par le biais de langages interprétés comme JavaScript). Il existe plusieurs langages pour créer ces pages: [PHP](#), [Java](#), [C#](#), [Ruby](#), voire d'autres comme [C++](#), [Python](#) et [Visual Basic](#) via l'interface [CGI](#).

I.3. Architecture client/serveur

I.3.1. Définition de l'architecture client/serveur

Le modèle client-serveur s'articule autour d'un réseau auquel sont connectés deux types d'ordinateurs le serveur et le client. Le client et le serveur communiquent via des protocoles. Les

applications et les données sont réparties entre le client et le serveur de manière à réduire les coûts. Le client-serveur représente un dialogue entre deux processus informatiques par l'intermédiaire d'un échange de messages. Le processus client sous-traite au processus serveur des services à réaliser. Les processus sont généralement exécutés sur des machines, des OS et des réseaux hétérogènes. [2]

I.3.2. Fonctionnement du l'architecture client/serveur

Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur. Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port. [2]

- **Le client** : Est un processus qui demande l'exécution d'une opération à un autre processus par un envoi de messages contenant le descriptif de l'opération à exécuter et attendant de cette opération par un message en retour.
- **Le serveur** : Est un processus accomplissant une opération sur demande d'un client et lui transmettant le résultat.
- **La requête** : Est un message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte de client.
- **La réponse** : Est un message transmis par un serveur à un client suite à l'exécution d'une opération contenant son résultat.

I.3.3. Caractéristiques de l'architecture client/serveur

Les éléments qui caractérisent une architecture client/serveur sont :

- **Service** : Le modèle client/serveur est une relation entre des processus qui tournent sur des machines séparées. Le serveur est un fournisseur de services. Le client est un consommateur de services. [2]
- **Partage de ressources** : Un serveur traite plusieurs clients et contrôle leurs accès aux ressources. [2]
- **Protocole asymétrique** : Conséquence du partage de ressources, le protocole de communication est asymétrique le client déclenche le dialogue ; le serveur attend les requêtes des clients. [2]
- **Transparence de la localisation** : L'architecture client/serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau). Transparence par rapport aux systèmes d'exploitation et aux plates-

formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments. [2]

- **Message** : Les messages sont les moyens d'échanges entre client et serveur.
- **Encapsulation des services** : Un client demande un service. Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique. [2]
- **Evolution** : Une architecture client/serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs). [2]

I.3.4. Type d'architectures client/serveur

Il existe 4 types d'architectures :

- **Architecture "Peer to Peer"** : Il existe deux types pour cette architecture, le centralisé et le décentralisé. Le réseau est dit pair à pair (Peer-to-Peer en anglais, ou P2P), lorsque chaque ordinateur connecté au réseau est susceptible de jouer tour à tour le rôle de client et celui de serveur. [3]

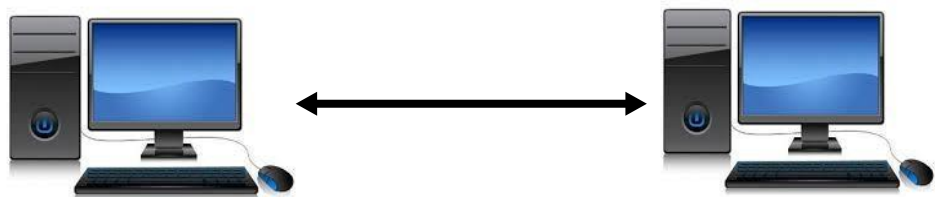


Figure I.1 : Architecture peer to peer

- **Architecture à 2 niveaux** : Ce type d'architecture (2-tier en anglais) caractérise l'architecture client-serveur où le poste client demande une ressource au serveur qui la fournit à partir de ses propres ressources.

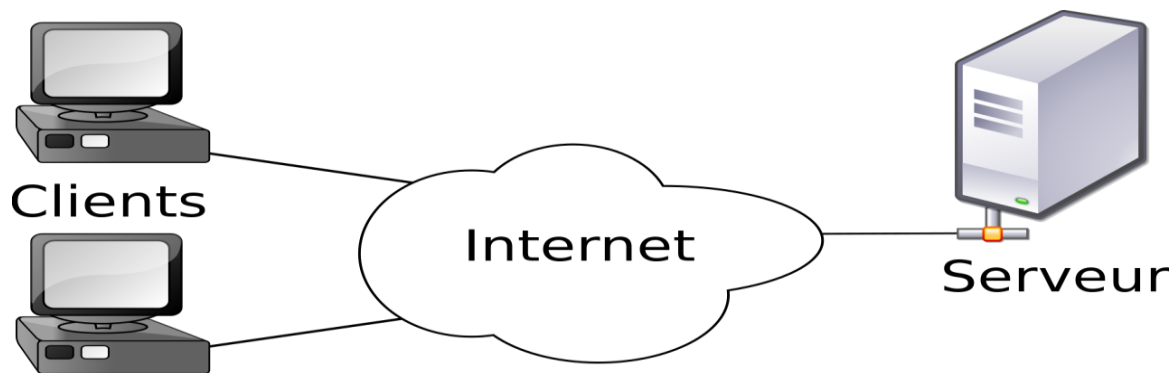


Figure I.2 : Architecture à 2 niveaux

- **Architecture à 3 niveaux** : Dans cette architecture (3-tier en anglais), existe un niveau supplémentaire. Un client (l'ordinateur demandeur de ressources) équipé d'une interface utilisateur (généralement un navigateur web) chargé de la présentation. Un serveur d'application (appelé middleware) qui fournit la ressource, mais en faisant appel à un autre serveur. Un serveur de données qui fournit au serveur d'application les données requises pour répondre au client. [3]

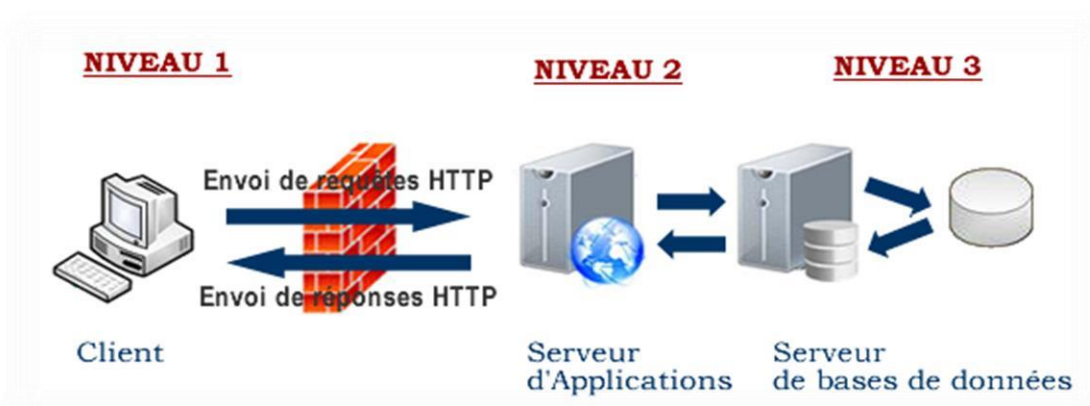


Figure I.3 : Architecture à 3 niveaux

- **Architecture à N niveaux** : On voit que l'architecture 3 niveaux permet de spécialiser les serveurs dans une tâche précise : Avantage de flexibilité, de sécurité et de performance. Potentiellement, l'architecture peut être étendue sur un nombre de niveaux plus important : On parle dans ce cas d'architecture à N niveaux (ou multi-tiers).[3]

N-Tier Architecture

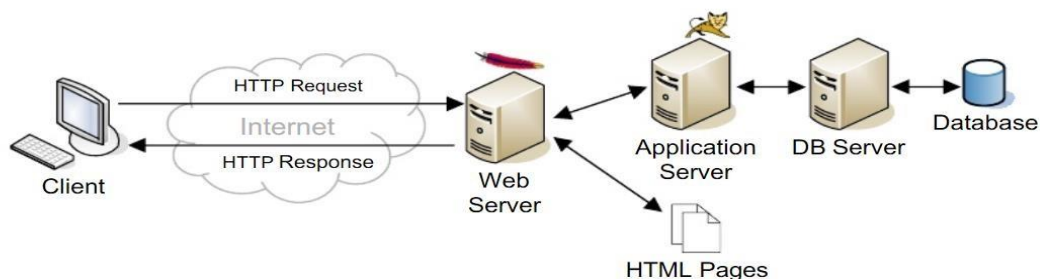


Figure I.4 : Architecture à N niveaux

I.3.5. Avantages et inconvénients de l'architecture

I.3.5.1. Avantages

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **Des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- **Une meilleure sécurité** : car le nombre de points d'entrée permettant l'accès aux données est moins important
- **Une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- **Un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.
- **Une économie d'espace mémoire** : tous les postes clients vont chercher l'information en fonction de leurs besoins et ne doivent pas la stocker durablement.

I.3.5.2. Inconvénients

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

- **un coût élevé** dû à la technicité du serveur.
- **un maillon faible** : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui, le serveur risque de ne pas supporter la charge s'il y a une communication de plusieurs machines avec le serveur au même moment.

I.4. Conclusion :

L'architecture client /serveur est la base de tous les services réseaux informatique, c'est pour cela nous sommes intéressés par l'étude de ce modèle. Le but de ce chapitre c'est définir l'architecture général de ce modèle, décrire le fonctionnement et les caractéristiques de cette architecture et citer les différents types de ce modèle afin de mieux gérer notre application qui est basée sur le recrutement en ligne.

CHAPITRE 2

E-commerce

II.1.Introduction

Le commerce électronique constitue une transformation majeure des échanges économiques contemporains, redéfinissant les modes d'interaction entre entreprises, consommateurs et institutions publiques dans un contexte numérique. Ce chapitre propose une analyse approfondie des fondements de l'e-commerce, dépassant la simple notion de « vente en ligne » pour en dévoiler la complexité et la dynamique en tant qu'écosystème. Il retrace notamment les origines historiques de cette pratique, qui trouvent leurs racines dès les réseaux pré-internet des années 1970.

Dans un premier temps, seront définies avec rigueur les différentes formes de commerce électronique, en distinguant les principaux modèles transactionnels (B2B, B2C, C2C, C2B, B2G, G2C), appuyés par des illustrations concrètes. Par la suite, l'étude s'attardera sur les éléments techniques essentiels à la mise en œuvre d'une plateforme e-commerce efficace, en détaillant notamment :

- L'architecture back-end, incluant la gestion des stocks, la logistique et le service client ;
- Le front-end, avec un focus sur la gestion des bases de données, la sécurité et le traitement des paiements ;
- Les protocoles de sécurité tels que les certificats numériques et le SSL, ainsi que les solutions de paiement sécurisées, comme le protocole SET.

Ce chapitre met en lumière l'impact de la technologie sur la transformation des transactions commerciales ainsi que sur l'évolution des relations économiques à l'échelle globale, offrant une lecture incontournable pour appréhender les mécanismes du commerce au XXI^e siècle.

II 2. Définition et types de l'e-commerce :

II .2.1. Définition :

Le e-commerce ou le commerce électronique, un sous ensemble de l'e-business, est l'achat, la vente et l'échange de biens et de services sur les réseaux informatiques (comme internet) par le biais duquel les opérations ou les conditions de vente sont exercées par voie électronique. Contrairement à la croyance populaire, le commerce électronique n'est pas seulement sur le web. En fait, le commerce électronique est bien vivant dans les transactions entre entreprise avant le web dans les années 70 par l'intermédiaire de l'EDI

(Electronic Data Interchange) a travers des VAN (Value-Added Networks). E-commerce peuvent être répartis en quatre catégories principales : **B2B**, **B2C**, **C2B** et **C2C**.

II.2.2 Types of E-Commerce: E-Commerce - Business Models

Les modèles commerciaux du commerce électronique (E-Commerce ou Commerce électronique) peuvent généralement être classés dans les catégories suivantes :

1. Entreprise à Entreprise (E2E)
2. Entreprise à Consommateur (E2C)
3. Consommateur à Consommateur (C2C)
4. Consommateur à Entreprise (C2E)
5. Entreprise à Gouvernement (E2G)
6. Gouvernement à Entreprise (G2E)
7. Gouvernement à Citoyen (G2C)

II 2.2.1. Entreprise à Entreprise (E2E)

Un site web suivant le modèle commercial E2E vend ses produits à un acheteur intermédiaire qui, à son tour, les revend au client final. Par exemple, un grossiste passe une commande sur le site web d'une entreprise et, après avoir reçu la marchandise, vend le produit final au client qui se rend dans le point de vente de détail du grossiste pour l'acheter. [4]

II 2.2.2. Entreprise à Consommateur (E2C)

Un site web suivant le modèle commercial E2C vend ses produits directement à un client. Le client peut consulter les produits affichés sur le site web de l'organisation commerciale. Il peut choisir un produit et passer commande. Le site web envoie une notification à l'organisation commerciale par e-mail, et l'organisation expédiera le produit/les marchandises au client. [5]

II 2.2.3. Consommateur à Consommateur (C2C)

Un site web suivant le modèle commercial C2C aide les consommateurs à vendre leurs biens, tels que des propriétés résidentielles, des voitures, des motos, etc., ou à louer une chambre en publiant leurs informations sur le site. Le site peut ou non facturer le consommateur pour ses services. Un autre consommateur peut choisir d'acheter le produit du premier client en consultant l'annonce/publicité sur le site. [6]

II 2.2.4. Consommateur à Entreprise (C2E)

Dans ce modèle, un consommateur se rend sur un site web qui présente plusieurs organisations commerciales offrant un service particulier. Le consommateur indique le montant qu'il/elle souhaite dépenser pour un service donné. Par exemple, la comparaison des taux d'intérêt des

prêts personnels/ prêts automobiles proposés par différentes banques via le site. L'organisation commerciale qui satisfait la demande du consommateur dans le budget spécifié contacte le client et lui propose ses services. [7]

II 2.2.5. Entreprise à Gouvernement (E2G)

Le modèle E2G est une variante du modèle E2E. De tels sites web sont utilisés par le gouvernement pour commercer et échanger des informations avec diverses organisations commerciales. Ces sites web sont accrédités par le gouvernement et offrent un moyen aux entreprises de soumettre des formulaires de demande au gouvernement.

II .2.2.6.Gouvernement à Entreprise (G2E)

Le gouvernement utilise des sites web suivant le modèle G2E pour approcher les organisations commerciales. Ces sites web soutiennent des fonctionnalités telles que les enchères, les appels d'offres et la soumission de candidatures.

II .2.2.7Gouvernement à Citoyen (G2C)

Le gouvernement utilise des sites web suivant le modèle G2C pour s'adresser aux citoyens en général. Ces sites soutiennent des enchères de véhicules, de machines ou de tout autre matériel. Ces sites offrent également des services tels que l'enregistrement des certificats de naissance, de mariage ou de décès. Les principaux objectifs des sites G2C sont de réduire le temps moyen nécessaire pour répondre aux demandes des citoyens pour divers services gouvernementaux. [8]

De : Fournisseur de contenu/service	Consommateur ou citoyen	Entreprise (organisation)	Gouvernement
Consommateur ou citoyen	Consommateur à Consommateur (C2C) <ul style="list-style-type: none"> - eBay - Peer-to-Peer (Skype) - Blogs et communautés - Recommandations de produits - Réseaux sociaux : MySpace, Bebo 	Entreprise à Consommateur (E2C) <ul style="list-style-type: none"> - Transactionnel : - Amazon - Création de relation : BP - Création de marque : - Unilever - Propriétaire de médias - News Corp - Comparateurs de prix : Kelkoo, Pricerunner 	Gouvernement à Consommateur (G2C)• <p>Services transactionnels nationaux :</p> <p>Impôts – finances publiques</p> <p>Informations gouvernementales nationales</p> <p>Services gouvernementaux locaux</p>
Entreprise (organisation)	Consommateur à Entreprise (C2E) <ul style="list-style-type: none"> - Priceline - Avis consommateurs, communautés ou campagnes 	Entreprise à Entreprise (E2E) <ul style="list-style-type: none"> - Transactionnel : - Eurooffice - Création de relation : BP - Médias spécialisés : Publications commerciales Emap - Places de marché B2B : EC21 	Gouvernement à Entreprise (G2E) <ul style="list-style-type: none"> - Services et transactions gouvernementales : impôts - Réglementations légales
Gouvernement	Consommateur à Gouvernement (C2G) <p>Retours vers le gouvernement via des groupes de pression ou sites individuels</p>	Entreprise à Gouvernement (E2G) <p>Retours vers le gouvernement des entreprises ou organisations non gouvernementales</p>	Gouvernement à Gouvernement (G2G) <p>Services intergouvernementaux•</p> <p>Échange d'informations</p>

II.3. Composants du Commerce Électronique

Cette section explique les éléments essentiels requis pour créer une application immobilière en ligne réussie, en partant des bases liées aux services immobiliers jusqu'aux technologies modernes utilisées dans le commerce électronique immobilier. Malgré l'échec de nombreuses plateformes en ligne, le marché immobilier électronique connaît une croissance rapide et continue.

De nombreuses entreprises immobilières cherchent à lancer des interfaces électroniques interactives, voire à créer des activités entièrement basées sur la vente et la location de propriétés via Internet.

Deux aspects principaux doivent être pris en compte lors de la création d'une application immobilière électronique : la (Back-End) et la (Front-End).

– BACK-END

Avant de travailler sur l'interface utilisateur front-end, il est impératif de s'assurer que le système de gestion back-end fonctionne de manière efficace et fluide. De nombreuses applications immobilières qui ont échoué souffraient d'une faiblesse à cet égard. La configuration du back-end comprend :

a) Gestion des Propriétés et des Services

Plusieurs options sont disponibles pour l'intégration des propriétés dans l'application :

- L'entreprise peut saisir manuellement les propriétés et mettre à jour leurs informations telles que les photos, les prix, et l'emplacement.
- Un système permettant aux propriétaires d'insérer directement les données de leurs biens (système d'auto-listage) peut être adopté.
- L'application peut être connectée à des systèmes externes tels que des agences ou des bureaux de courtage immobilier pour afficher automatiquement leurs propriétés.

b) Service Client

Quelle que soit la structure de l'application, offrir un service client exceptionnel est essentiel.

L'un des éléments clés d'un support efficace est de fournir à l'utilisateur l'état de sa demande ou de son enquête, que ce soit par e-mail ou via le tableau de bord de l'application.

– FRONT-END

Une fois que le back-end est en place, vous pouvez procéder à la configuration du front-end. Il y a cinq parties essentielles pour la mise en place de tout site de commerce électronique :

- Base de données
- HTML Dynamique (DHTML)

- Modèle Objet Document (DOM)
- Sécurité
- Traitement des paiements

Une base de données

La base de données doit contenir des informations détaillées sur les utilisateurs et les propriétés, telles que :

- Données de l'utilisateur (Nom, E-mail, Numéro de téléphone, Localisation)
- Préférences de l'utilisateur (types de propriétés préférées, fourchette de prix, zone géographique)
- Historique des recherches et des intérêts
- Historique des demandes ou des réservations

Ainsi que les informations sur les propriétés :

- Numéro de la propriété
- Description détaillée
- Photos de la propriété
- Localisation de la propriété sur la carte
- Prix
- Détails de location ou de vente
- Frais d'agence ou de services

La base de données doit être connectée aux systèmes back-end pour le traitement des réservations, des demandes ou des négociations directes avec le propriétaire ou l'agent.

– DHTML

Il est utilisé pour conférer à l'application interactivité et fluidité d'utilisation, comme :

- **Cartes interactives** des propriétés
- **Images de propriétés animées** au survol
- **Listes déroulantes** pour les options de filtre
- **Navigation fluide** entre les pages de propriétés et les résultats

– Modèle d'Objet Document (DOM)

Il permet aux développeurs de manipuler les éléments de l'interface comme des composants modifiables et interactifs, tels que la modification d'images ou de détails selon le choix de l'utilisateur.

Si la mise en œuvre d'un DOM avancé n'est pas possible, il peut être remplacé par l'utilisation de **CSS** pour le formatage et la conception interactive des éléments.

II.4. Sécurité

1. Certificats Numériques

Ils jouent un rôle clé dans la protection des données des utilisateurs lors de la connexion ou du paiement :

- **Certificat d'Autorité de Certification (CA)** : Émis par des entités telles que VeriSign pour garantir l'identité de l'entreprise.
- **Certificats de Serveur** : Pour sécuriser la connexion entre l'application et ses serveurs.
- **Certificats Personnels** : Pour les utilisateurs souhaitant des transactions fiables.
- **Certificats d'Éditeur** : Assurent que les mises à jour logicielles de l'application sont sécurisées et non modifiées.

2. Cryptographie à Clé Publique

Elle repose sur une paire de clés : une **clé publique** pour le chiffrement et une **clé privée** pour le déchiffrement, assurant ainsi la confidentialité des informations.

– SSL (Secure Sockets Layer)

SSL assure un chiffrement complet des données échangées entre l'application et l'utilisateur, empêchant l'espionnage ou la modification des informations pendant le transfert.

Par exemple, lorsqu'un utilisateur se connecte pour réserver une propriété ou envoyer un acompte, les données sont chiffrées et envoyées en toute sécurité au serveur, où elles sont déchiffrées sans qu'aucune partie externe ne puisse les intercepter ou les modifier.

-Comment fonctionne SSL

1. Un consommateur entre sur le site d'une entreprise et accède à une page sécurisée (indiquée par une URL commençant par "https:").
2. Le serveur hébergeant le site de l'entreprise envoie ensuite au consommateur son certificat de serveur, ce qui authentifie le site.
3. Le navigateur du consommateur génère une clé de session unique pour chiffrer toutes les communications.
4. Une session sécurisée est désormais établie. Selon le navigateur utilisé, le consommateur peut voir un cadenas dans la barre d'informations de son navigateur.

Les certificats de serveur SSL existent en deux niveaux de sécurité : le chiffrement à 40 bits et le chiffrement à 128 bits

Plus le nombre de bits est élevé, plus il est difficile de casser le code de chiffrement.

II.6. Traitement des paiements

Comme mentionné précédemment, il existe trois méthodes différentes pour traiter les paiements sur Internet : le modèle de paiement en espèces, le modèle de paiement par chèque et le modèle de paiement par carte de crédit.

a) Modèle de paiement en espèces (Cash Model)

Le modèle de paiement en espèces (e-cash) consiste en la création de monnaie électronique ou de jetons, généralement par une banque, que les acheteurs et les vendeurs échangent pour des biens et des services.

b) Modèle de paiement par chèque (Check Model)

Dans le modèle de paiement par chèque, le consommateur présente une version numérique de son chèque à une vitrine en ligne.

c) Modèle de paiement par carte de crédit (Credit Model)

Dans le modèle de paiement par carte de crédit, le consommateur saisit ses informations de carte de crédit et le paiement est traité par une organisation accréditée, généralement une banque, utilisant SET (Secure Electronic Transaction).

II.7.SET

SET, qui signifie transactions électroniques sécurisées, est utilisé avec le modèle de paiement par carte de crédit et est la méthode de paiement la plus courante sur Internet, car elle utilise le système de traitement des cartes de crédit existant.

Étape Un : Le consommateur passe une commande en fournissant ses informations de carte de crédit et les envoie à l'entreprise.

Étape Deux : L'entreprise envoie au consommateur une facture, son certificat ainsi que le certificat de sa banque.

Étape Trois : Le consommateur prend connaissance des informations, les approuve et les renvoie à l'entreprise.

Étape Quatre : L'entreprise génère ensuite une demande d'autorisation pour la carte de crédit du consommateur et l'envoie à sa banque.

Étape Cinq : La banque de l'entreprise envoie ensuite la demande d'autorisation de crédit à l'acquéreur.

Étape Six : L'acquéreur envoie un accusé de réception à la banque de l'entreprise après avoir reçu un accusé de réception de la banque du consommateur.

Étape Sept : Une fois que la banque du consommateur autorise le paiement, la banque de l'entreprise envoie un accusé de réception à l'entreprise avec le numéro d'autorisation..

Les étapes une à trois se produisent lors de la passation de votre commande sur le site web de l'entreprise. Les étapes quatre à sept sont les mêmes que celles suivies lorsque vous effectuez un achat en magasin. La seule différence est que ces étapes se déroulent sur Internet au lieu de se faire à un terminal de caisse.

II.8.COMPTE MARCHAND INTERNET

Après avoir sélectionné le type de traitement des paiements à utiliser, l'étape suivante consiste à créer un compte marchand Internet. Le compte marchand Internet doit être ouvert auprès d'une institution financière qui vous permet d'accepter les cartes de crédit ou les cartes d'achat pour les paiements en ligne et qui prend en charge les processeurs suivants :

- First Data Merchant Service (FDMS)
- Paymentech (Salem)
- Vital Processing Services
- Nova Information Systems

Obtenir un compte marchand Internet peut prendre entre deux jours et trois semaines.

II.9.Conclusion :

En conclusion, le commerce électronique apparaît comme un levier incontournable de la transformation économique à l'ère numérique, en modifiant profondément les pratiques commerciales traditionnelles et en favorisant l'émergence de nouveaux modèles d'affaires. L'analyse des différents types de transactions, couplée à une compréhension fine des composantes techniques et sécuritaires des plateformes e-commerce, révèle la complexité et la richesse de cet écosystème. Par ailleurs, l'évolution continue des technologies numériques invite à une adaptation permanente des acteurs pour répondre aux attentes croissantes des consommateurs et aux défis liés à la sécurité et à la confiance. Ainsi, maîtriser les enjeux du commerce électronique est essentiel pour saisir les dynamiques économiques actuelles et anticiper celles de demain.

CHAPITRE 3

Analyse et conception

III.1. Introduction :

Dans ce chapitre, nous présentons la solution conceptuelle que nous avons élaborée pour le développement de notre application immobilière. Autrement dit, ce chapitre vise à répondre à une question centrale : comment modéliser et concevoir de manière structurée notre système avant son implémentation ?

Afin de garantir une modélisation claire, cohérente et conforme aux standards actuels du développement logiciel, nous avons choisi d'adopter le langage **UML (Unified Modeling Language)**. Il s'agit d'un langage de modélisation unifié, orienté objet, largement utilisé dans la conception de systèmes informatiques. UML permet de représenter graphiquement les différentes composantes d'un système — ses acteurs, ses cas d'utilisation, sa structure interne ainsi que ses comportements dynamiques — ce qui facilite la compréhension, la communication et l'évolution du projet.

La phase de conception a ainsi été structurée autour d'une série de diagrammes UML (cas d'utilisation, classes, séquence, activités, etc.), chacun jouant un rôle précis dans la modélisation fonctionnelle et technique de notre application.

Ce chapitre détaille donc ces différents diagrammes et justifie leur utilité dans la conception d'une plateforme immobilière moderne, performante et évolutive.

III.2. Présentation de l'UML :

III.2.1. Définition :

UML (Unified Modeling Language) est un langage de modélisation standardisé à l'échelle internationale, largement utilisé dans l'industrie du développement logiciel. Il permet de visualiser, spécifier, construire et documenter les différents composants d'un système logiciel, en l'occurrence ici, ceux d'une application immobilière. Grâce à ses différents types de diagrammes (cas d'utilisation, classes, séquences, etc.), UML facilite la compréhension des interactions entre les utilisateurs (agents, acheteurs, administrateurs) et les fonctionnalités offertes par le système, tout en assurant une structure claire et évolutive du projet. [14]

III.2.2. L'utilité de l'UML :

UML (Unified Modeling Language) adopte une approche orientée objet en tant que langage universel de description. Il permet, à travers une série de diagrammes clairs et structurés, de représenter l'architecture et le fonctionnement d'un système informatique complexe – comme celui d'une application immobilière – en mettant en évidence les relations entre les différents concepts métiers (annonces, utilisateurs, transactions, etc.) et leur mise en œuvre fonctionnelle.

UML constitue avant tout un outil de communication puissant, facilitant l'expression, le partage et la compréhension des solutions orientées objet. Sa notation graphique normalisée permet une visualisation intuitive de la structure du système, ce qui aide à comparer, analyser et évaluer différentes alternatives de conception.

Par son caractère formel, UML réduit les ambiguïtés et les incompréhensions, tout en offrant une indépendance vis-à-vis des langages de programmation, des domaines métier et des méthodologies de développement, ce qui en fait un langage de modélisation universel et particulièrement pertinent pour la conception d'un système d'information immobilier. [15]

III.3. Description du projet

Notre projet vise à garantir que les abonnés à l'application achètent des produits et que ces clients aient la possibilité d'acheter des produits via un site Web.

III.4. Identification des acteurs

Acteur	Description
Visiteur	Utilisateur non enregistré qui consulte librement les annonces immobilières publiées sur la plateforme sans pouvoir effectuer d'actions nécessitant une authentification (comme enregistrer une annonce, envoyer une demande ou accéder à un espace personnel).
Client	Utilisateur enregistré (inscrit et connecté) pouvant interagir pleinement avec le système : créer et gérer un profil, publier ou réserver des biens, envoyer des messages, ou effectuer des demandes de contact ou de visite.
Administrateur	Utilisateur ayant des droits étendus de gestion du système. Il supervise l'ensemble de la plateforme, valide ou supprime des annonces, gère les utilisateurs (clients, agents immobiliers), modère les contenus, et s'assure du bon fonctionnement global de l'application.

Tableau.III. 1. Description des acteurs

III.5. Description des rôles de chacun des acteurs

III.5.1. Le visiteur :

Utilisateur anonyme ou non enregistré qui consulte activement les offres immobilières (biens à vendre ou à louer), compare les caractéristiques et les prix, et s'informe sur le marché via l'application. Bien qu'il ne soit pas encore client, il manifeste un intérêt réel et constitue une cible prioritaire pour la conversion.

III.5.2. Le Client :

Il s'agit d'un visiteur ayant déjà créé un compte sur l'application. Grâce à son authentification, il peut accéder à des fonctionnalités avancées telles que la gestion de ses favoris, le suivi de ses démarches (prises de rendez-vous, offres soumises, historiques de recherche), et effectuer des actions sécurisées tout au long du processus d'achat ou de location d'un bien immobilier.

III.5.3. Administrateur:

Dans le cadre de l'application, l'administrateur (souvent appelé "webmaster" sur les sites web) est chargé de garantir le bon fonctionnement de la plateforme. Il supervise la mise à jour des annonces immobilières (ajout, modification, suppression de biens), la gestion des tarifs, la disponibilité des biens, ainsi que le suivi des demandes, des paiements et, le cas échéant, des processus liés à la réservation ou à la location. Il veille également à la sécurité et à la fluidité de l'expérience utilisateur.[16]

III.6. Identification des messages échangés :

Un message est une spécification de communication unidirectionnelle entre des objets qui transmet des informations destinées à déclencher une activité de la part du destinataire.

Acteur	Message	Services
Visiteur	Rechercher sur le site	Afficher les résultats de recherche
	Consulter les produits	Afficher une page dynamique qui contient les produits
	Inscription	Afficher un formulaire de l'inscription
Client	Ajouter, Modifier ou supprimer un une annonce	Sauvegarder, modifier ou supprimer le une annonce
	Finaliser l'achat	
	Noter	Sauvegarder la note
	Demande de modifier ou supprimer son compte	Demande de confirmation, après il peut modifier ou supprimer son compte

Tableau.III. 2 Identification des messages échangés

III.7. Diagramme de contexte

Le diagramme de contexte d'un système représente les flux d'informations, de services ou de produits échangés entre le système (vu comme une boîte noire) et les acteurs externes de son environnement.

Ce diagramme permet de visualiser les limites du système, de définir les interactions principales avec l'extérieur et d'identifier les différents acteurs (utilisateurs, clients, administrateurs, partenaires, etc.) qui interagissent avec la plateforme.

Dans le cas de notre application immobilière, ce diagramme nous aide à représenter :

- Les utilisateurs non inscrits (visiteurs) qui consultent ou recherchent des biens.
- Les clients inscrits qui peuvent enregistrer des biens favoris, faire une demande de visite, ou finaliser une transaction.
- L'administrateur qui gère les annonces, les utilisateurs et le contenu de la plateforme.

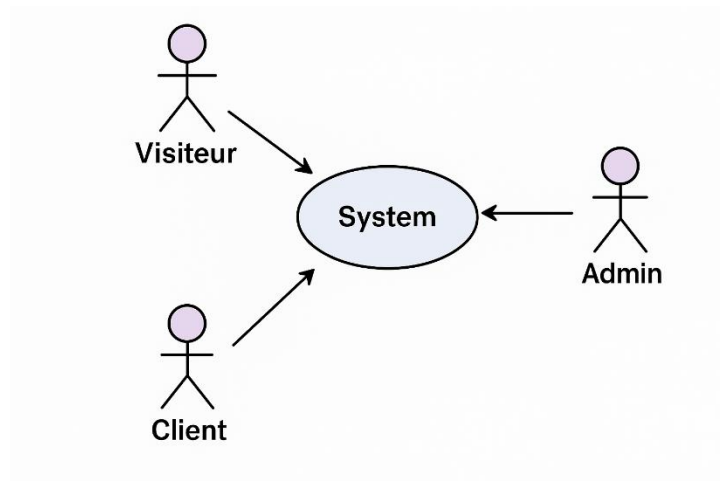


Figure III. 1 Diagramme de contexte

III.8. Capture des besoins :

Le système prend en charge trois types d'utilisateurs : particuliers, agences immobilières et entreprises de construction.

- ✓ Chaque utilisateur peut :
- ✓ Publier une annonce de bien immobilier.
- ✓ Modifier ou supprimer ses propres annonces.
- ✓ Consulter et évaluer les annonces publiées par d'autres utilisateurs.
- ✓ Signaler les annonces frauduleuses ou suspectes. [17]

III.8.1. Identification des cas d'utilisations :

N°	Cas d'utilisations		Acteurs
1	- Effectuer une recherche		-Visiteur-Client -Administrateur
2	- Consulter les		
3	- Inscription		Visiteur
4	- S'authentifier		Administrateur
5	- Modifier un compte		
6	- Supprimer un compte		
7	- Gestion de Publication	Ajouter	
		Modifier	
		Supprimer	
8	- Gestion de commentaire		
9	- S'authentifier		Client
10	- Commenter		
	-Noter un annonce		
11	- Gestion de compte	Modifier	
		Désabonner	

Tableau.III. 3 Identification des cas d'utilisation

III.9. Description textuelle des cas d'utilisations

Ce qui suit décrit une série de cas d'utilisation.

II.9.1. Description textuelle de cas d'utilisation «Effectuer une recherche »

Titre	Effectuer une recherche
Acteur	Visiteur / Client / Administrateur
Description	L'utilisateur a la possibilité de chercher un immobilière
Pré-condition	Le visiteur doit d'abord s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur s'authentifie sur la plateforme. 2- Il accède à la barre de recherche. 3- Il saisit un mot-clé relatif au produit recherché. 4- Il valide la recherche. 5- Le système affiche les résultats correspondants au mot-clé saisi.

Tableau.III. 4 Description textuelle du cas d'utilisation « Effectuer une rechercher »

III.9.2. Description textuelle de cas d'utilisation «Modifier son compte »

Titre	Modifier son compte
Acteur	Client
Description	L'utilisateur a la possibilité de modifier son compte.
Pré-condition	L'utilisateur doit d'abord s'authentifier
Scénario nominal	<ol style="list-style-type: none"> 1- L'utilisateur se connecte au site. 2- L'utilisateur demande à modifier son compte. 3- Le système affiche un formulaire. 4- L'utilisateur effectue les modifications désirée et valide. 5- Le système sauvegarde les modifications et affiche un message de confirmation.

Tableau.III. 5 Description textuelle du cas d'utilisation « Modifier son compte »

III.9.3. Description textuelle de cas d'utilisation « S'authentifier»

Titre	S'authentifier
Acteur	Client
Description	Le client peut s'authentifier pour accéder à son compte.
Pré-condition	Le client doit être connecté au site.
Scénario nominal	<ol style="list-style-type: none">1- Le client saisie les informations nécessaires (Email et un mot de passe) et valider.2- Le système vérifie le mot de passe et l'Email (pour confirmer l'existence du compte).3- Le système ouvre la session.4- L'utilisateur accède à son compte.
Scénario alternatif	<p>2.(A1) : - Si un champ de saisis et vide, alors le système indique qu'un champ est incomplet et demande de ressaisir.</p> <p>3.(A2) : Si l'Email et/ou le mot de passe est incorrect, le système affiche un message d'erreur.</p>

Tableau.III. 6 Description textuelle du cas d'utilisation « S'authentifier»

III.9.4. Description textuelle de cas d'utilisation « Gestion de biens immobiliers »

Titre	Gestion de immobilière
Acteur	Administrateur
Description	L'administrateur a la possibilité de gérer les annonces immobilières du site. Il peut ajouter de nouveaux biens, modifier les informations existantes ou supprimer des annonces.
Pré-condition	L'administrateur doit être authentifié au préalable pour accéder à cette fonctionnalité.
Scénario nominal	<p>1- L'administrateur se connecte à son espace d'administration.</p> <p>2- Il sélectionne l'action souhaitée : ajouter, modifier ou supprimer un bien immobilier.</p> <p>3- Le système affiche un formulaire adapté à l'opération choisie.</p> <p>4- L'administrateur saisit ou met à jour les informations nécessaires, puis confirme l'opération.</p> <p>5- Le système valide et enregistre les modifications, puis affiche un message de confirmation.</p>
Scénario alternatif	5 (A1) : Si les données saisies sont incomplètes ou incorrectes (ex. : champ vide, format non conforme), le système affiche un message d'erreur et invite l'administrateur à corriger les informations.

Tableau.III. 7 Description textuelle du cas d'utilisation Gestion de produit

III.10. Le diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des outils essentiels du langage UML (Unified Modeling Language) utilisés lors de la phase d'analyse d'un projet pour identifier les principales fonctionnalités d'un système. Ils illustrent les interactions entre les acteurs (utilisateurs ou systèmes externes) et le système étudié. Sans entrer dans les détails techniques ou fonctionnels, ces diagrammes mettent en évidence les relations fondamentales entre les acteurs, les cas d'utilisation et le système. Leur objectif principal est de définir les services ou objectifs que le système doit offrir.

III.10.1. Le diagramme de cas d'utilisation global (E-commerce)

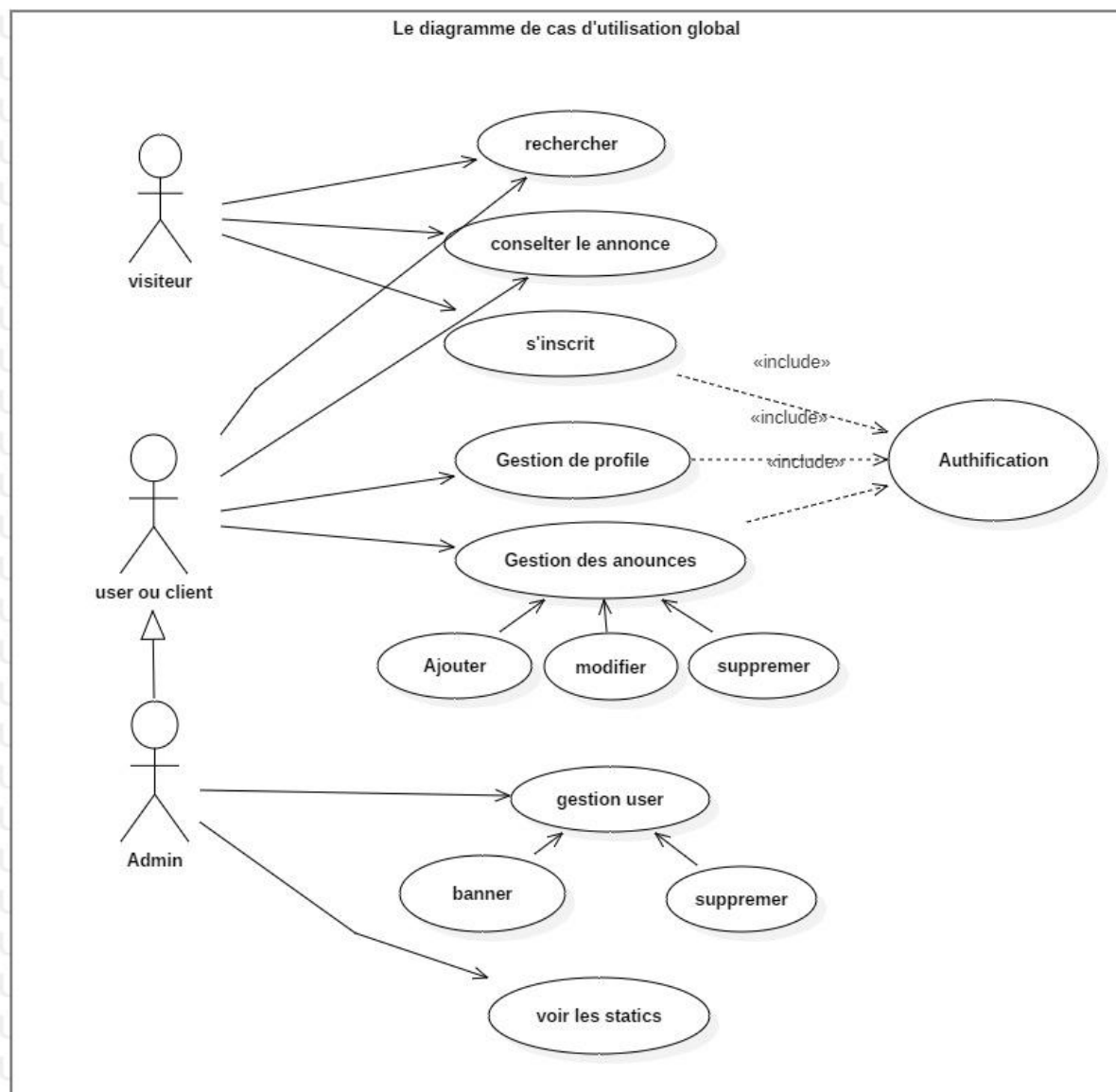


Figure III. 2. Le diagramme de cas d'utilisation global

III.11. Diagramme de séquence :

Pour effectuer une recherche, les utilisateurs doivent entrer des mots-clés spécifiques au résultat souhaité. S'il y a des résultats, le système les affichera dans un formulaire de recherche avancée.

Cela permet aux utilisateurs d'effectuer différentes recherches pour mieux atteindre les résultats souhaités. BDD dans ce cas fait référence au produit.

III.11.1. Diagramme de séquence du cas d'utilisation << Effectuer une recherche >>

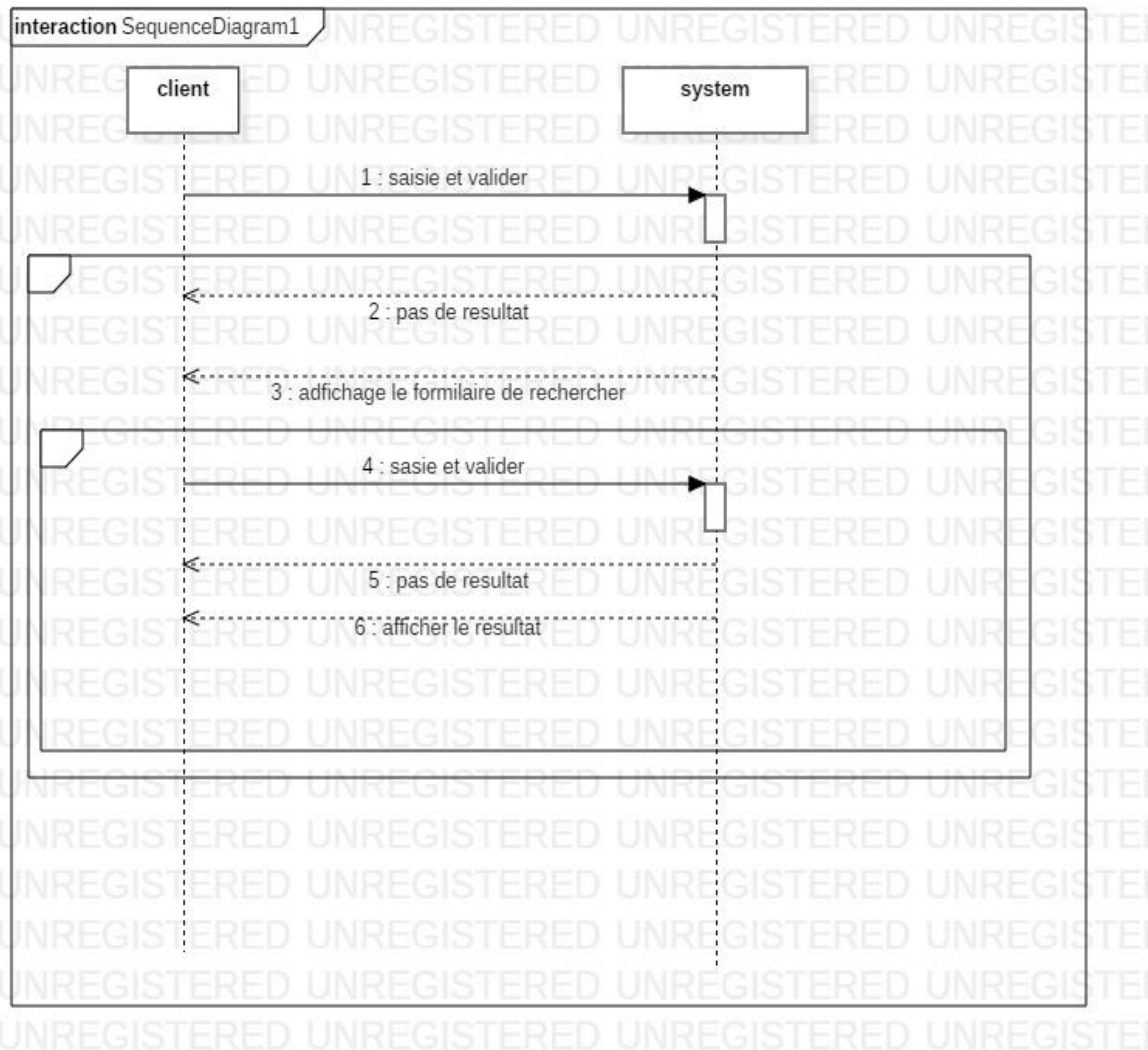


Figure III. 3. Diagramme de séquence du cas d'utilisation << Effectuer une recherche >>

III.11.2. Diagramme de séquence << S'authentifier >>

Les utilisateurs ont la possibilité de s'authentifier en saisissant leur identifiant et leur mot de passe dans les champs qui leur sont réservés. Après validation, si les données s'affichent correctement dans BDD, l'utilisateur poursuit sa session. Sinon, le système renverra un message d'erreur.

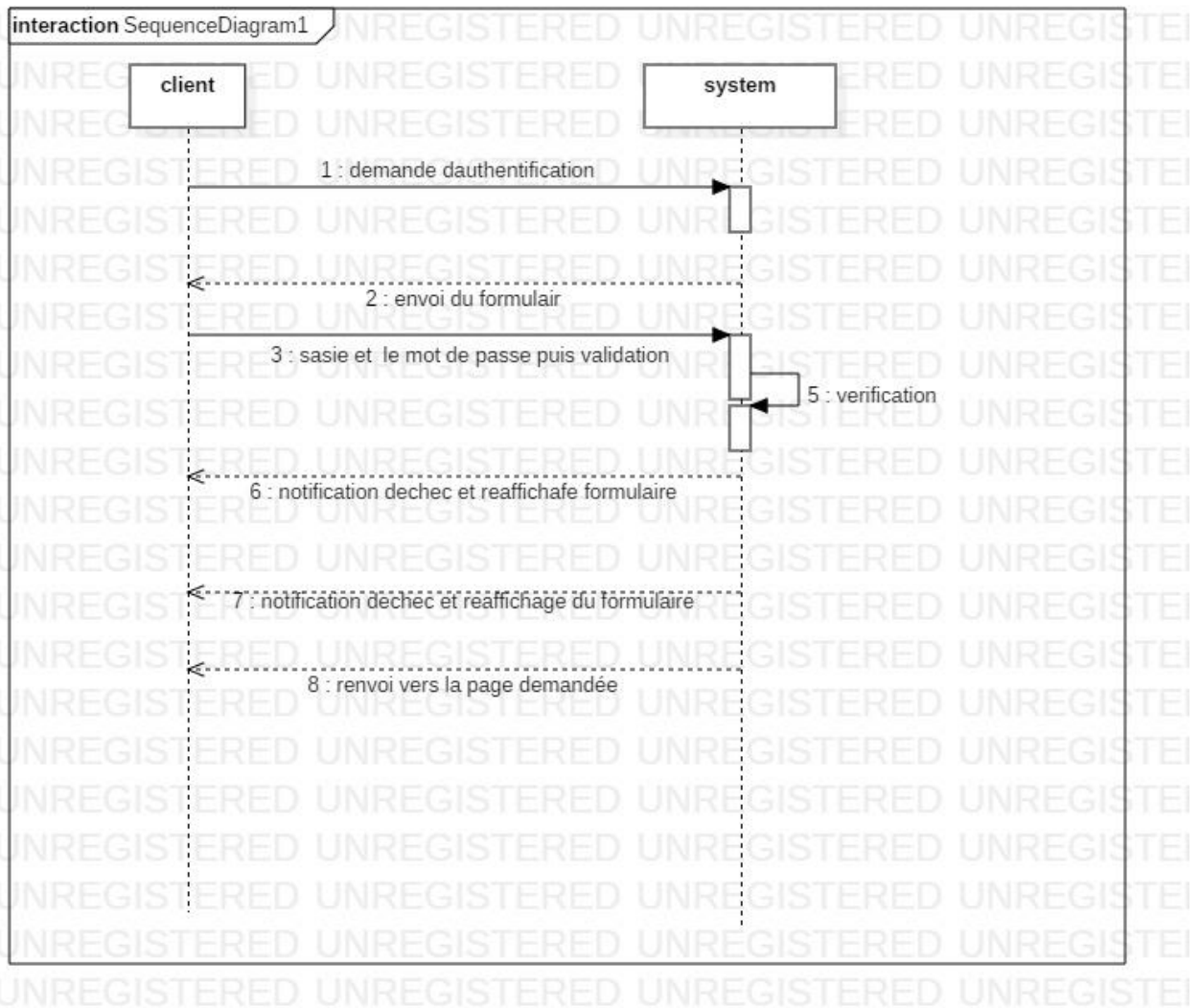


Figure III. 4. Diagramme de séquence du cas d'utilisation << S'authentifier>>

III -11.3. Diagramme de séquence du << Supprimer un compte >>

Les utilisateurs ont la possibilité de supprimer leur compte lorsqu'il n'est plus nécessaire ou lorsqu'un administrateur décide que ce n'est pas souhaitable. Pour ce faire, l'utilisateur demande la suppression du compte et le système demande à l'utilisateur une confirmation.

Après confirmation, le compte sera supprimé de la base de données et le système enverra un message de confirmation à l'utilisateur.

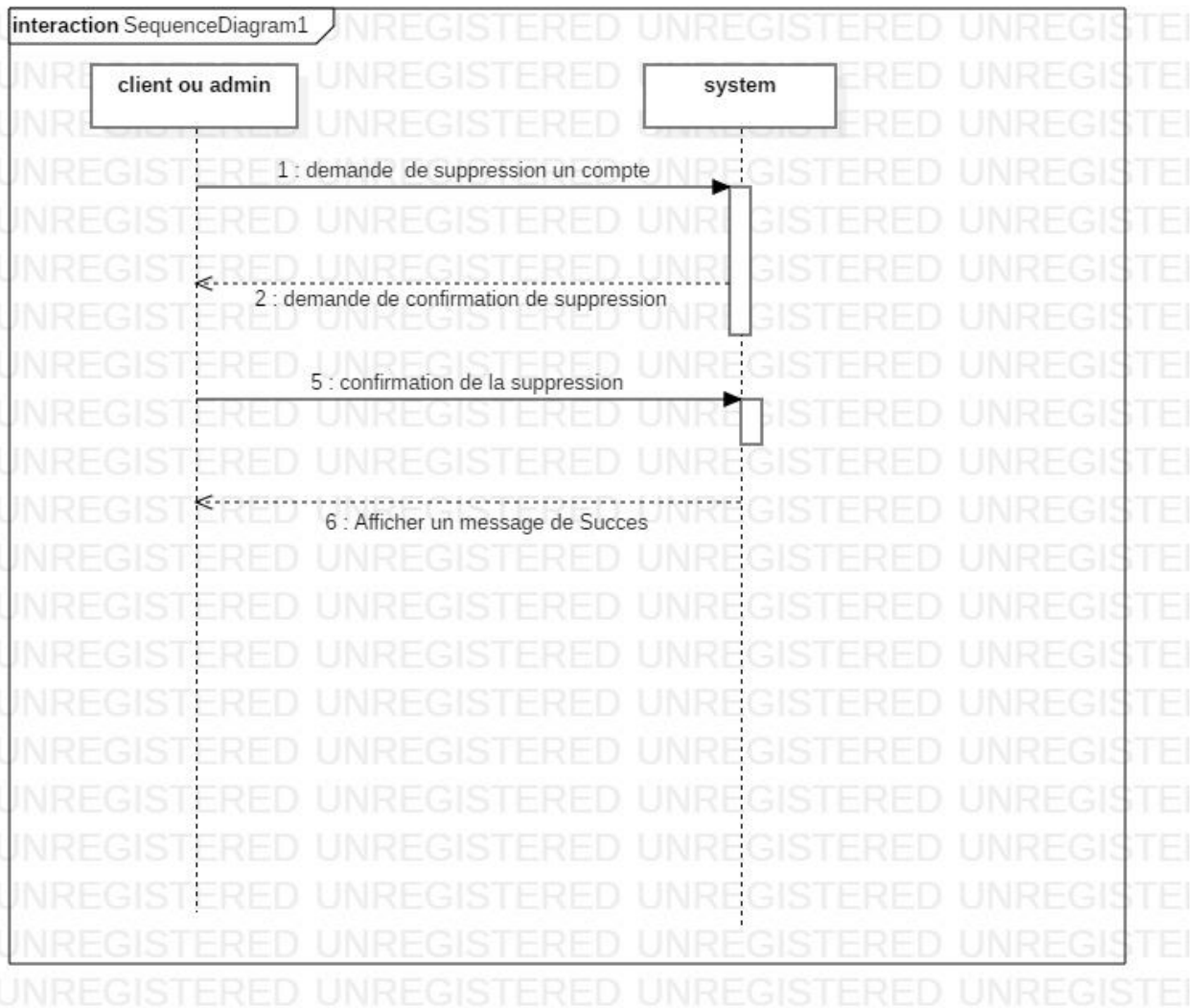


Figure III. 5 Diagramme de séquence du cas d'utilisation << Supprimer un compte >>

III.12. Diagramme d'activité

Le diagramme d'activité a pour objectif principal d'illustrer le flux de contrôle ou le déroulement des actions au sein d'une activité. Il est couramment utilisé pour modéliser la manière dont les différentes étapes d'un processus ou d'une opération sont enchaînées et coordonnées. Ce type de diagramme permet notamment de :

- visualiser la séquence des actions,
- représenter les services fournis au cours de l'activité,
- identifier les événements nécessaires à l'achèvement d'une tâche,
- et décrire les relations logiques ou temporelles entre les différentes étapes d'un cas d'utilisation spécifique.

Ainsi, le diagramme d'activité joue un rôle essentiel dans la compréhension du comportement dynamique du système, notamment dans le cadre de l'analyse fonctionnelle et de la modélisation des processus métier.

III.12.1. Les diagrammes d'activité utilisés :

III.12.1.1 Diagramme d'activité <<inscription >>

La phase d'inscription est essentielle pour attendre qu'un produit soit livré, depuis un simple visiteur du site qui n'a que le droit de voir le produit et son prix, jusqu'à un client qui peut acheter le produit et régler la facture.

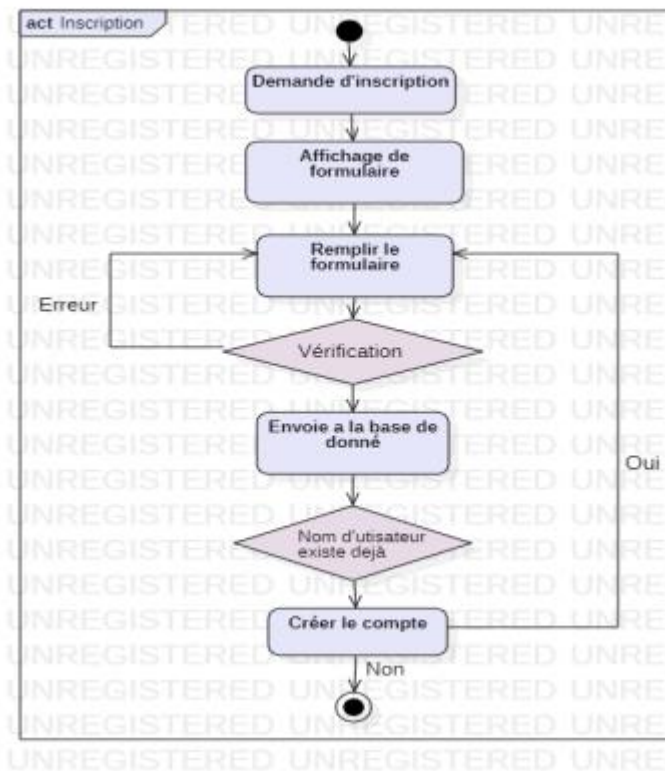


Figure III. 6. Diagramme d'activité << inscription >>

Nous expliquons ce diagramme par le processus suivant :

- Le visiteur demande l'inscription.
- Le formulaire d'inscription s'affiche sur l'écran.
- Le visiteur remplit les champs demandés dans le formulaire.
- Le système vérifie les données entrées.
- Si les données sont acceptées, le système les envoie à la base, sinon, il revient à l'étape précédente.
- Le serveur vérifie l'existence du client dans la base.

- Si le client existe déjà, le feedback renvoyé est un message d'erreur qui s'affiche.
- Si le client n'existe pas, l'inscription se termine avec succès suivant d'un feedback pour notifier à l'utilisateur que son inscription s'est effectuée. [18]

III.12.1.2 Diagramme d'activité <<authentication >>

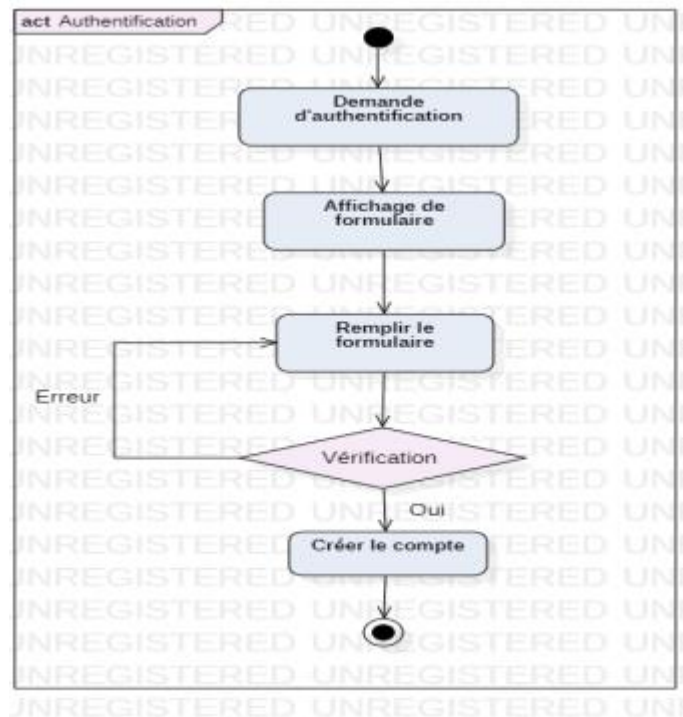


Figure III. 7 . Diagramme d'activité << authentication >>

- *Le formulaire d'authentification s'affichera à l'écran.*
- *Les clients entrent leur nom d'utilisateur et leur mot de passe.*
- *Le système vérifie les coordonnées du client sur la base*
- *Une confirmation de réussite ou d'échec est envoyée au client.*

III.12.1.3. Diagramme d'activité <<Finaliser un achat>>

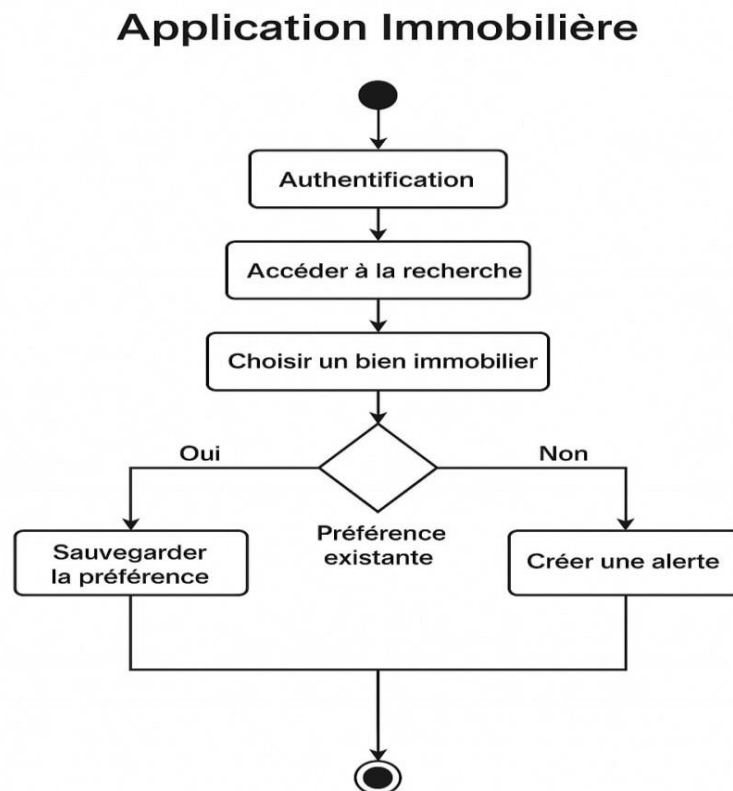
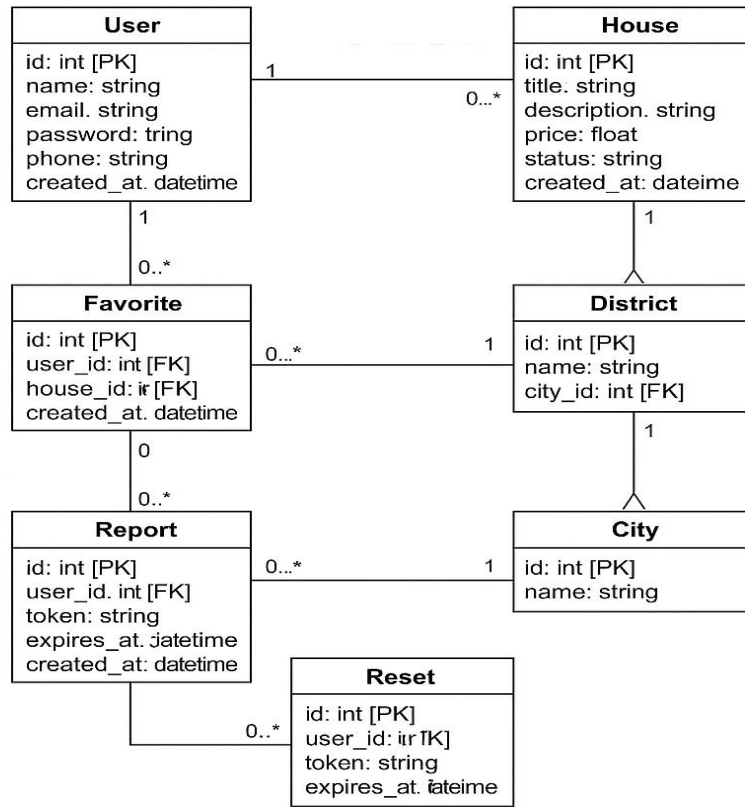


Figure III. 8. Diagramme d'activité <<Finaliser un achat>>

III.13. Diagramme de classe :

Le diagramme de classe est un type de diagramme utilisé en UML pour représenter la structure statique d'un système. Il montre les différentes classes du système, leurs attributs, leurs méthodes, ainsi que les relations entre elles. Ce diagramme est essentiel lors de la phase de modélisation et de conception d'une application, car il permet de visualiser comment les objets interagissent et s'organisent dans le système.



CHAPITRE 4

Realsation

IV.1. Introduction

Ce chapitre décrit en détail la phase de réalisation de notre application, en mettant l'accent sur les aspects techniques et pratiques du développement. Il commence par la présentation de l'environnement de développement, qui comprend à la fois l'environnement matériel (matériels utilisés pour coder, tester et déployer l'application) et l'environnement logiciel (systèmes d'exploitation, langages de programmation, bibliothèques, frameworks, etc.). Ensuite, nous exposerons les principaux logiciels et outils ayant facilité le développement, comme les éditeurs de code, les systèmes de gestion de base de données, les outils de versionnage ou encore les simulateurs de tests. La suite du chapitre sera consacrée à la présentation des principales interfaces graphiques de l'application, accompagnée de captures d'écran et d'explications fonctionnelles illustrant les interactions utilisateur. Enfin, ce chapitre permettra de mettre en évidence les choix techniques opérés, les contraintes rencontrées, ainsi que la manière dont les besoins des utilisateurs ont été traduits en solutions concrètes et efficaces sur le plan visuel et fonctionnel.

IV.2.L'environnement de développement :

IV.2.1.1.L'environnement matériel :

- ✓ Machine : ASUS
- ✓ Mémoire : 8Go.
- ✓ Disque Dur : 500 Go ,256 GO .
- ✓ Processeur : RAYZEN 5.
- ✓ Type de système : Windows 11

IV.2.1.2.Environment Logiciel :

- HTML
- CSS
- Bootstrap
- Javascript
- NodeJS
- Expressjs
- EJS
- PostgreSQL
- Sequelize
- JWT
- Bcryptjs
- Cookie Parser
- SendGrid
- Cloudinary

1. Visual Studio Code (VS Code)

- Éditeur de code léger et puissant, développé par Microsoft.
- Supporte JavaScript, Node.js, HTML, CSS, et bien d'autres langages.
- Extensions utiles : Prettier, ESLint, Live Server, etc.

2. Node.js

- Environnement d'exécution JavaScript côté serveur.
- Permet de développer des applications web rapides et évolutives.
- Utilisé avec Express.js pour gérer les routes, requêtes HTTP, etc.

IV .2.1.3 - Bases de Données et Outils Associés

PostgreSQL

- Système de gestion de base de données relationnelle (SGBDR) robuste et open source.
- Supporte les requêtes SQL complexes, la sécurité, et l'intégrité des données.

pgAdmin

- Interface graphique pour gérer les bases de données PostgreSQL.
- Permet la création, la modification et la visualisation des bases de données.

Sequelize (et Sequelize CLI)

- ORM (Object-Relational Mapping) pour Node.js.
- Facilite les interactions avec la base PostgreSQL à travers des objets JavaScript.
- Sequelize CLI permet de gérer les migrations, les modèles et la base de données depuis le terminal.

IV.2.1.4- Développement Web (Frontend et Backend)

1. HTML

- Langage de balisage utilisé pour structurer les pages web.

2. CSS

- Langage de style utilisé pour la présentation visuelle (couleurs, polices, disposition...).

3. Bootstrap

- Framework CSS prêt à l'emploi.
- Facilite la création d'interfaces web responsives et modernes.

4. JavaScript

- Langage de programmation utilisé pour rendre les pages web interactives.

5. EJS (Embedded JavaScript Templates)

- Moteur de templates pour générer dynamiquement du HTML avec JavaScript côté serveur.

6. Express.js

- Framework minimaliste basé sur Node.js.
- Gère les routes, les middlewares et la logique serveur.

14. Cookie-Parser

- Middleware utilisé pour parser (analyser) les cookies HTTP dans Express.

IV.2.1.5- Sécurité et Authentification

1. JWT (JSON Web Token)

- Utilisé pour gérer l'authentification et les sessions utilisateur de manière sécurisée.

2. Bcryptjs

- Bibliothèque permettant de chiffrer (hasher) les mots de passe avant de les stocker.

IV.2.1.6- Services Cloud et Intégration

1. Cloudinary

- Plateforme cloud pour stocker et gérer des images, vidéos, et autres médias.
- Permet de transformer les fichiers à la volée et de les intégrer facilement dans une app web.

2. SendGrid

- Service d'envoi d'e-mails sécurisé et fiable.
- Utilisé pour envoyer des e-mails de confirmation, de récupération de mot de passe, etc.

Tests et API

1. Postman

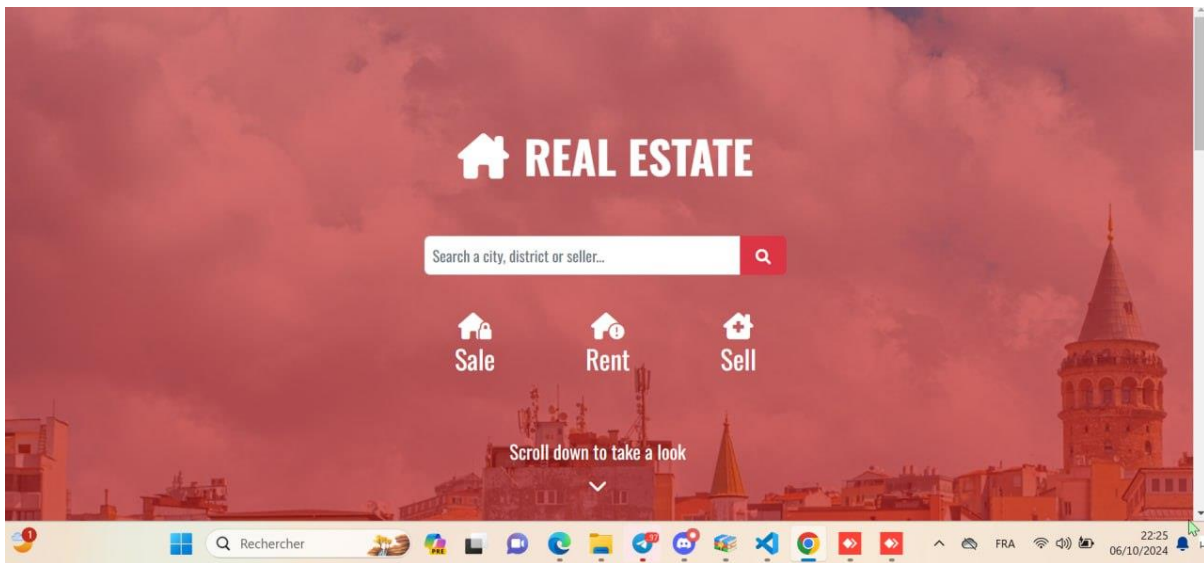
- Outil pour tester les requêtes HTTP (GET, POST, PUT, DELETE).
- Utile pour tester les routes API pendant le développement.

2. Insomnia (optionnel)

- Alternative à Postman pour tester les APIs avec une interface plus légère.

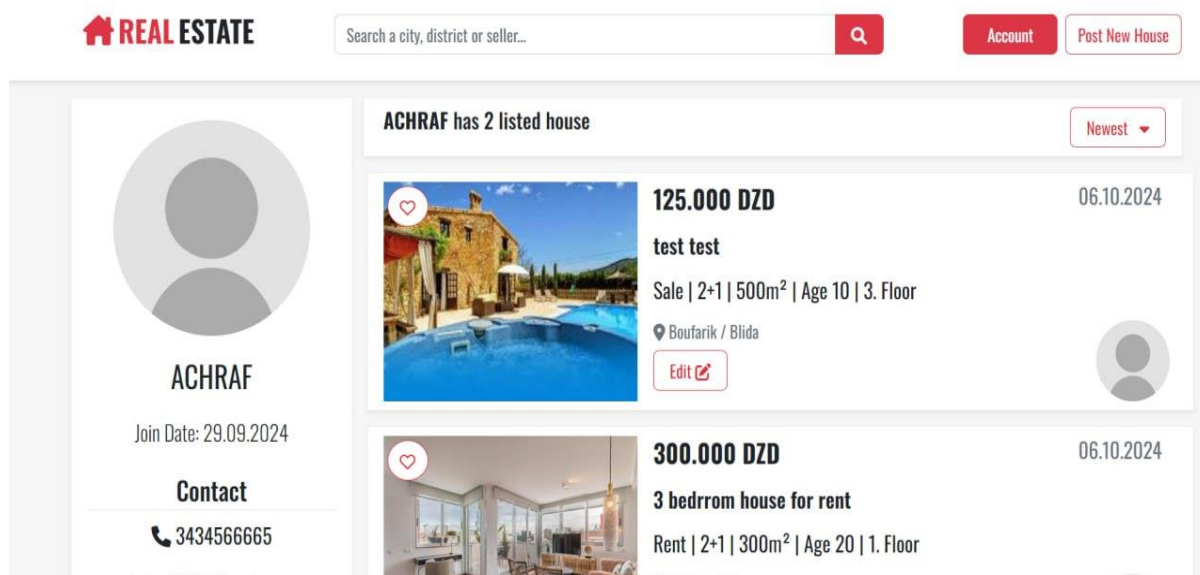
IV.4. Les principales interfaces graphiques

IV.4.1. Interface principale



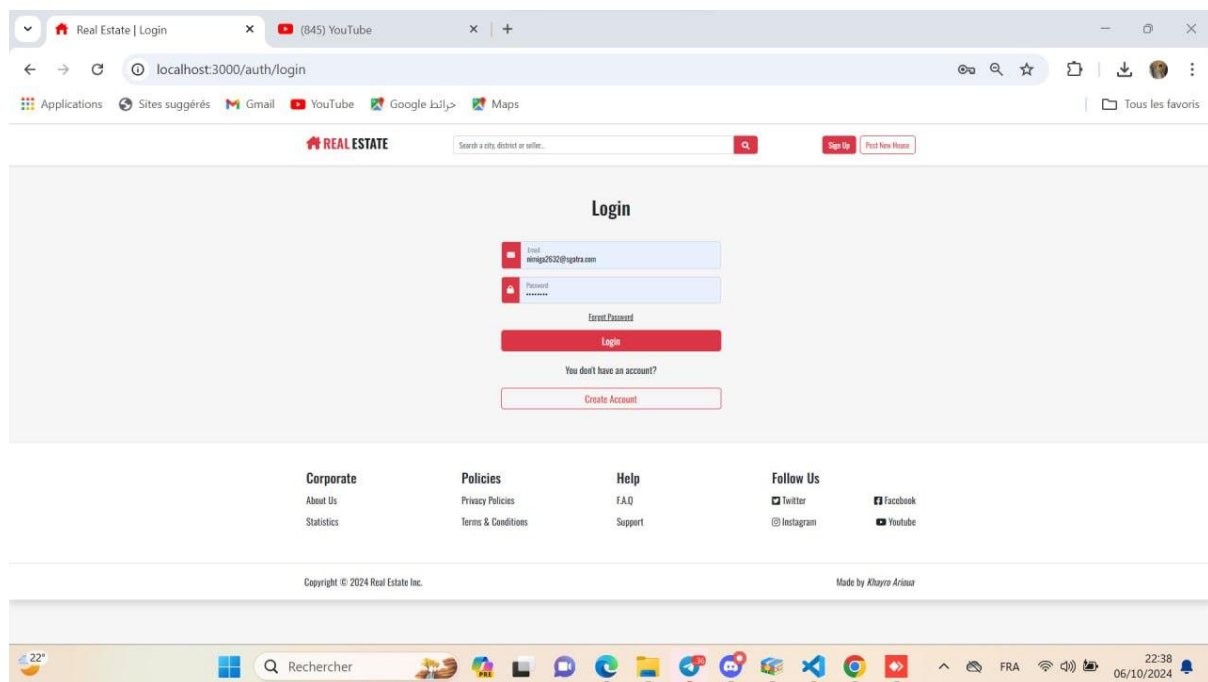
L'interface principale de l'application REAL ESTATE adopte une approche centrée sur l'utilisateur, privilégiant la simplicité et l'accessibilité. La présence d'une barre de recherche bien positionnée répond aux attentes des utilisateurs en matière de rapidité et de précision dans la recherche immobilière, conformément aux principes d'UX modernes qui valorisent l'efficacité des parcours utilisateurs . Les options de filtrage (« Sale », « Rent », « Sell ») permettent une segmentation claire des intentions, réduisant ainsi la charge cognitive et facilitant la prise de décision.

IV.4.2. Interface des state



L'interface analysée présente une structure fonctionnelle mais souffre de lacunes en matière de cohérence visuelle et d'ergonomie. Bien que les informations essentielles (prix, caractéristiques du bien, contact) soient présentes et partiellement hiérarchisées, leur agencement manque de clarté, notamment en raison de l'absence de séparation visuelle nette entre les différentes annonces et sections. Des éléments superflus (ex. « test test ») et une mise en forme inégale (polices, espacements) nuisent à la lisibilité et au professionnalisme perçu. L'inclusion de données structurées (superficie, nombre de pièces) répond aux besoins utilitaire, mais l'interface gagnerait à adopter des principes de design plus rigoureux (ex. grille typographique, contraste visuel) pour améliorer l'expérience utilisateur et faciliter la comparaison entre biens. Une intégration d'icônes ou de visuels minimaux pourrait par ailleurs renforcer l'attractivité sans compromettre la minimalisme. Cette analyse souligne l'importance d'un équilibre entre fonctionnalité et esthétique dans les interfaces transactionnelles.

IV.4.3. Interface login



L'interface de connexion de REAL ESTATE présente une structure fonctionnelle mais souffre de lacunes significatives en matière d'ergonomie et de sécurité, compromettant son efficacité. Bien que l'organisation générale suive une logique minimaliste conforme aux standards UX plusieurs éléments posent problème : l'affichage inapproprié d'un placeholder "[Email: huawei]" dans le champ mot de passe révèle des failles dans l'implémentation technique et soulève des questions sur la gestion des données sensibles . L'absence totale de feedback visuel pour les erreurs de saisie ou de confirmation de connexion viole les principes fondamentaux de l'interaction homme-machine (Norman, 1988), tandis que la disposition dense des éléments secondaires (mentions légales, liens utiles) en bas de page nuit à leur découvrabilité. Ces défauts d'affordance (Gibson, 1979) et de guidage utilisateur sont particulièrement préjudiciables dans un contexte transactionnel où la confiance et la clarté sont primordiales. La présence simultanée d'artefacts de développement ("huawei") et d'une structuration rigide suggère un processus de conception inabouti, où les impératifs techniques auraient pris le pas sur l'expérience utilisateur. Une refonte intégrant (1) des mécanismes de validation en temps réel, (2) une hiérarchie visuelle plus claire et (3) l'élimination des éléments superflus serait nécessaire pour répondre aux standards académiques en matière d'interfaces sécurisées et centrées-utilisateur.

IV.5.Conclusion :

L'environnement logiciel mis en place pour ce projet web repose sur une architecture moderne, complète et cohérente, combinant des technologies robustes côté client (HTML, CSS, Bootstrap, JavaScript) et côté serveur (Node.js, Express.js). L'intégration de PostgreSQL avec l'ORM Sequelize assure une gestion structurée et performante des données. Des outils comme JWT et Bcryptjs garantissent la sécurité des utilisateurs, tandis que des services externes comme SendGrid et Cloudinary renforcent les fonctionnalités de communication et de gestion des médias. Enfin, l'utilisation d'éditeurs professionnels (Visual Studio Code), de systèmes de contrôle de version (Git/GitHub), et d'outils de test (Postman) permet un développement fluide, collaboratif et efficace. Cet ensemble de logiciels permet ainsi de répondre aux exigences fonctionnelles, techniques et sécuritaires d'une application web moderne.

Conclusion Générale

Cette application propose une plateforme numérique complète, spécialement conçue pour répondre aux besoins croissants du marché immobilier moderne. Elle permet aux utilisateurs, qu'ils soient vendeurs ou acquéreurs potentiels, de publier facilement des annonces de maisons, tout en offrant aux acheteurs la possibilité de consulter, trier, et évaluer une vaste base de données d'offres immobilières en temps réel.

Grâce à un système de recherche doté de filtres avancés, les utilisateurs peuvent affiner leurs critères selon une multitude de paramètres : localisation géographique, fourchette de prix, type de bien (villa, appartement, maison traditionnelle, etc.), superficie, nombre de pièces, commodités (jardin, garage, piscine, etc.) ou encore proximité de services (écoles, commerces, transports). Cette fonctionnalité améliore considérablement la pertinence des résultats et permet une recherche ciblée, intuitive et rapide.

Lorsqu'un bien immobilier suscite l'intérêt d'un utilisateur, ce dernier peut l'ajouter à sa liste de favoris afin de le retrouver facilement lors de connexions ultérieures. De plus, l'application intègre un système de messagerie sécurisé qui facilite la prise de contact directe entre acheteurs et vendeurs, réduisant ainsi les intermédiaires et favorisant des échanges transparents, rapides et efficaces.

En matière d'expérience utilisateur, l'application se distingue par son interface ergonomique et conviviale, accessible aussi bien aux débutants qu'aux utilisateurs expérimentés. Elle met l'accent sur la fluidité de la navigation, la clarté des informations, et la personnalisation de l'expérience selon les profils et les besoins de chacun. Ce positionnement stratégique permet à l'application de se hisser parmi les outils numériques les plus performants dans le domaine de la vente immobilière, à l'image des grandes plateformes concurrentes du secteur.

Perspectives de développement futur :

Dans une logique d'évolution continue, l'application prévoit d'intégrer prochainement plusieurs fonctionnalités innovantes pour renforcer encore davantage son attractivité et sa compétitivité.

Parmi ces améliorations figurent :

- **L'intelligence artificielle** pour proposer des suggestions de biens personnalisés selon le comportement de recherche des utilisateurs.
- **La réalité virtuelle (VR)** permettant des visites immersives des maisons à distance, ce qui constituera un gain de temps considérable pour les acheteurs.
- **L'ajout de systèmes d'évaluation et de notation** des vendeurs et des annonces, afin de renforcer la fiabilité et la transparence sur la plateforme.
- **La possibilité de programmer des visites physiques directement via l'application**, avec synchronisation de calendrier.
- **Un tableau de bord pour les vendeurs**, offrant des statistiques avancées sur les performances de leurs annonces (vues, clics, temps de consultation).
- **Une intégration bancaire** facilitant les simulations de prêt ou les contacts avec des partenaires financiers agréés.

En somme, cette application ne se limite pas à servir de vitrine immobilière numérique, mais ambitionne de devenir un véritable écosystème intégré dédié à l'achat et à la vente de biens immobiliers. Grâce à ses outils actuels et à ses évolutions futures, elle entend s'imposer durablement comme un acteur incontournable du secteur immobilier digitalisé.

Bibliographie

[1]: <https://web.maths.unsw.edu.au/~lafaye/CCM/cs/csintro.htm/>

[2]: BELKHOUCHE S., Etude et Administration des Systèmes de Supervision dans un Réseau, Mémoire pour l'Obtention d'un Diplôme d'Ingénieur d'Etat en Informatique soutenu à l'Université Abou Bakr Belkaid-Tlemcen, 2011

[3]: <https://fr.wikipedia.org/wiki/>

[4]: dictionnaire Larousse

[5]: HASSANI S., E-Recrutement, Mémoire de fin de cycle, soutenu à l'Université Mohamed V-Soussi, 2014.

[6] <https://www.eyerys.com/articles/types-e-commerce-models>

[7] https://www.tutorialspoint.com/e-commerce/e-commerce_edi.htm

[8] <https://www.journals.elsevier.com/electronic-commerce-research-and-applications>

[9] <https://www.tandfonline.com/doi/abs/10.1080/15332861.2010.529052>

[10] Journal-Initial trust and intentions to buy: The effect of vendor-specific guarantees, customer reviews and the role of online shopping experience

[11] Consumer-to-Consumer e-Commerce Research in Information Systems Journals

[12] <https://www.clickz.com/the-ultimate-guide-to-ecommerce-business-models/101740/>

13 <https://www.africmemoire.com/part.5-ii-3-1-le-langage-de-modelisation-uml-949.html>

14 <https://ocabdrs.blogspot.com/p/i.html>

15 Mr BENACHOUR Kouceila, Mr TOUATI Mourad «Conception et réalisation d'une application Web ECommerce sous Java EE». Master, Bejaia, 2014.

[16] Mr. Adel RAISSI «Conception et développement d'un site web de e-commerce pour le compte de LSAT_Nokia ».Master, Tunis, 2012.

[17] <https://www.techno-science.net/glossaire-definition/Visual-Studio.html>

Résumé :

Ce mémoire examine l'impact des applications immobilières sur la digitalisation du marché immobilier en Algérie, dans un contexte marqué par la généralisation des smartphones et l'amélioration de l'accès à Internet. Ces plateformes numériques facilitent la mise en relation entre acheteurs et vendeurs grâce à des fonctionnalités innovantes telles que les annonces en temps réel, la géolocalisation et les visites virtuelles. À travers une approche analytique, ce travail évalue l'efficacité de ces outils, identifie leurs avantages et défis, et propose des recommandations pour une intégration réussie dans le contexte algérien. La recherche s'appuie sur une étude technique incluant la conception UML, l'architecture client/serveur, les aspects du e-commerce et la mise en œuvre d'une application web sécurisée utilisant des technologies modernes côté client et serveur. L'objectif est de contribuer à une transformation numérique durable et adaptée aux besoins du marché local.

Mots-clés :

Applications immobilières, digitalisation, Algérie, e-commerce, UML, architecture client/serveur, technologies web.

Abstract :

This thesis explores the impact of real estate applications on the digitalization of the property market in Algeria, within a context shaped by widespread smartphone usage and improved internet access. These digital platforms streamline interactions between buyers and sellers through innovative features such as real-time listings, geolocation, and virtual tours. Using an analytical approach, the study assesses the effectiveness of these tools, highlights their benefits and challenges, and offers recommendations for their successful integration into the Algerian market. The technical aspect includes UML-based system design, client/server architecture, e-commerce components, and the development of a secure web application using modern technologies on both the client and server sides. The goal is to support a sustainable digital transformation tailored to local market needs.

Keywords:

Real estate applications, digitalization, Algeria, e-commerce, UML, client/server architecture, web technologies.

ملخص :

يتناول هذا البحث تأثير التطبيقات العقارية على رقمنة سوق العقار في الجزائر، في ظل الانتشار الواسع للهواتف الذكية وتحسن الولوج إلى الإنترنت. تساهم هذه المنصات الرقمية في تسهيل التواصل بين البائعين والمشتريين من خلال ميزات مبتكرة كإعلانات الفورية، وتحديد الموقع الجغرافي، والجولات الافتراضية. ومن خلال منهج تحليلي، يقيم هذا العمل مدى فعالية هذه الأدوات، ويبرز مزاياها وتحدياتها، مع تقديم توصيات لضمان إدماجها الناجح في السياق الجزائري. كما يشتمل الجانب التطبيقي على تصميم النظام باستخدام لغة UML، وهيكلية العميل/الخادم، ومكونات التجارة الإلكترونية، بالإضافة إلى تطوير تطبيق ويب آمن بالاعتماد على تقنيات حديثة في الواجهة الأمامية والخلفية. ويهدف هذا المشروع إلى دعم التحول الرقمي المستدام بما يتماشى مع متطلبات السوق المحلي.

الكلمات المفتاحية

التطبيقات العقارية، الرقمنة، الجزائر، التجارة الإلكترونية، UML، بنية العميل/الخادم، تقنيات الويب.