

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement Supérieur et de la Recherche Scientifique



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

Université de Béjaia  
Faculté des Sciences Exactes  
Département d'Informatique

## MÉMOIRE DE MASTER PROFESSIONNEL

En Informatique

Option : *Génie Logiciel*

### Thème

Développement d'une application web et  
mobile pour la gestion d'une agence  
immobilière

Présenté par :

LOUNIS Ines  
MEDDOUR Hadia

Soutenu le 02/07/2024 devant le jury composé de :

Présidente	Belkhiri Louiza	MCB	Université de Béjaia
Examineur	ZERARGA Loutfi	MCB	Université de Béjaia
Promotrice	OUYAHIA Samira	MCB	Université de Béjaia

Promotion :2023-2024

## *✧ Remerciements ✧*

*Nous tenons à remercier avant tout Dieu, qui nous a donné la force et le courage nécessaires pour accomplir ce modeste travail.*

*Notre profonde gratitude et nos sincères remerciements vont à notre promotrice, Mme Ouyahia Samira, pour nous avoir confié ce projet, pour son suivi, sa disponibilité, ses orientations et ses précieux conseils.*

*Nous souhaitons également exprimer notre reconnaissance à Mr Yous Athmane, directeur de l'entreprise où nous avons effectué notre stage, pour son aide précieuse, son soutien et ses encouragements tout au long de notre expérience professionnelle.*

*Nos remerciements s'adressent également aux membres du jury, qui ont accepté de juger notre travail.*

*Nous exprimons aussi notre gratitude à tous nos enseignants et aux membres du département informatique de l'Université Abderrahmane Mira.*

*Nous n'oublions pas de remercier chaleureusement tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce mémoire de fin d'études.*

***Merci à tous***

## ※ *Dédicaces* ※

### *Je dédie ce modeste travail :*

À mes chers parents, qui m'ont donné la force de continuer grâce à leur amour, leur patience et leur soutien. C'est grâce à eux que ce projet a été réalisé.

À mes frères Youcef, Saber et Chabane, et mes sœurs Soraya et Sabrina, qui ont toujours été à mes côtés avec leur soutien constant.

À la mémoire de ma grand-mère maternelle, qui continue de m'inspirer par ses prières et ses vœux de réussite. Son amour et ses bénédictions me portent toujours.

À ma grand-mère paternelle et mes oncles, qu'ils soient ici ou à l'étranger, pour leur soutien et encouragement inestimables.

À mon cher Lounis Bouyahmed, qui, malgré la distance en France, m'a toujours soutenu moralement et encouragé. Son amour et son soutien constant ont été une grande source de force pour moi.

À mes enseignants, qui m'ont guidée tout au long de ce parcours grâce à leurs conseils et leur enseignement précieux, je leur suis très reconnaissante pour leur soutien constant dans mon développement académique et personnel, m'aidant ainsi à atteindre mes objectifs.

Et enfin, à toutes les personnes qui ont contribué à la réalisation de ce projet, merci pour votre aide précieuse et votre soutien.

*Hadia*

## ※ *Dédicaces* ※

### *Je dédie ce modeste travail :*

À mes parents, pour leur amour inconditionnel, leur soutien sans faille et leurs encouragements constants. Vous avez été ma source d'inspiration et de motivation tout au long de ce parcours.

À mon frère Lyes, pour son soutien et ses encouragements. Ta présence et tes conseils ont été d'une grande aide..

À mes sœurs Lydia et Sonia, pour leur affection et leur soutien moral. Vous avez toujours su me redonner le sourire dans les moments difficiles.

À mes professeurs, pour leur dévouement et leur enseignement précieux tout au long de mon cursus.

À mon encadrante Samira Ouyahia, pour sa guidance, ses conseils avisés et son soutien indéfectible tout au long de ce projet.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, je vous exprime ma plus profonde gratitude.

*Ines*

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Liste des figures</b>	<b>v</b>
<b>Liste des tableaux</b>	<b>viii</b>
<b>1 Présentation de l'organisme d'accueil et étude de l'existant</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Présentation de l'EURL BEINX ALGERIE . . . . .	1
1.2.1 Effectifs de l'entreprise . . . . .	2
1.2.2 Activités commerciales . . . . .	2
1.3 Organigramme de l'entreprise . . . . .	3
1.4 Étude et critique de l'existant . . . . .	3
1.4.1 Ouedkniss : . . . . .	4
1.4.2 Elkeria : . . . . .	5
1.5 Problématique . . . . .	6
1.6 Présentation de l'application . . . . .	6
1.6.1 Objectifs . . . . .	9
1.6.2 Les besoins fonctionnels et non fonctionnels . . . . .	9
1.7 Conclusion . . . . .	12
<b>2 Modélisation et conception de projet</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Choix de la méthodologie de modélisation . . . . .	13
2.3 Processus de développement logiciel . . . . .	13
2.3.1 Définition Processus Unifié (UNIFIED PROCESS) . . . . .	14
2.3.2 Les caractéristiques du processus unifié . . . . .	14
2.3.3 Cycle de vie du processus unifié . . . . .	15

---

2.3.4	Les enchaînements d'activités D'UP . . . . .	16
2.3.5	Les phases et les disciplines du UP . . . . .	17
2.4	Analyse et conception . . . . .	19
2.4.1	Identification des acteurs . . . . .	19
2.5	Diagramme de contexte . . . . .	19
2.6	Diagramme de cas d'utilisation . . . . .	20
2.6.1	Plan des interactions pour «Utilisateur» . . . . .	21
2.6.2	Plan des interactions pour «Client» . . . . .	21
2.6.3	Plan des interactions pour «Gestionnaire» . . . . .	22
2.6.4	Plan des interactions pour «Agence» . . . . .	22
2.6.5	Plan des interactions pour «Agent immobilier» . . . . .	23
2.7	Fiche descriptive . . . . .	24
2.7.1	Fiche descriptive du cas d'utilisation «Ajouter un utilisateur» . . . . .	24
2.7.2	Fiche descriptive du cas d'utilisation «S'authentifier» . . . . .	25
2.7.3	Fiche descriptive du cas d'utilisation «Rechercher» . . . . .	26
2.7.4	Fiche descriptive du cas d'utilisation «Ajouter annonce» . . . . .	27
2.7.5	Fiche descriptive du cas d'utilisation «Modifier annonce» . . . . .	28
2.7.6	Fiche descriptive du cas d'utilisation «Supprimer annonce» . . . . .	29
2.7.7	Fiche descriptive du cas d'utilisation «Gérer les visites» . . . . .	30
2.8	Diagramme cas d'utilisation globale . . . . .	31
2.9	Les diagrammes de séquence système . . . . .	32
2.9.1	Diagramme de séquence «S'authentifier» . . . . .	32
2.9.2	Diagramme de séquence «ajouter un utilisateur» . . . . .	33
2.9.3	Diagramme de séquence «Modifier un utilisateur» . . . . .	34
2.9.4	Diagramme de séquence «Supprimer un utilisateur» . . . . .	35
2.9.5	Diagramme de séquence «Rechercher» . . . . .	36
2.9.6	Diagramme de séquence «Ajouter une annonce» . . . . .	37
2.9.7	Diagramme de séquence «Modifier une annonce» . . . . .	38
2.9.8	Diagramme de séquence «Supprimer une annonce» . . . . .	39
2.9.9	Diagramme de séquence «Contacter le gestionnaire» . . . . .	40
2.9.10	Diagramme de séquence «Effectuer une offre» . . . . .	41
2.10	Représentation des diagrammes d'activités . . . . .	42
2.10.1	Diagramme d'activité «S'authentifier» . . . . .	42
2.10.2	Diagramme d'activité «Ajouter un utilisateur» . . . . .	43
2.10.3	Diagramme d'activité «Modifier un utilisateur» . . . . .	44
2.10.4	Diagramme d'activité «Supprimer un utilisateur» . . . . .	45

---

2.11	Dictionnaire de données . . . . .	45
2.12	Diagramme de classes . . . . .	47
2.13	Modèle relationnel de données . . . . .	48
2.13.1	Passage d'un diagramme de classes UML à un schéma relationnel . . . . .	48
2.14	Conclusion . . . . .	50
<b>3</b>	<b>Réalisation et présentation de l'application</b>	<b>51</b>
3.1	Introduction . . . . .	51
3.2	Outils et langages de développements . . . . .	51
3.2.1	Les frameworks utilisés . . . . .	51
3.2.1.1	Laravel . . . . .	52
3.2.1.2	Bootstrap . . . . .	53
3.2.2	Langages de programmation utilisés . . . . .	54
3.2.2.1	HTML . . . . .	54
3.2.2.2	CSS . . . . .	54
3.2.2.3	JavaScript . . . . .	54
3.2.2.4	PHP . . . . .	54
3.2.3	Logiciels utilisés pour le développement . . . . .	55
3.2.3.1	Visual Studio Code . . . . .	55
3.2.3.2	Android Studio . . . . .	55
3.2.3.3	API REST . . . . .	55
3.2.3.4	XAMPP . . . . .	56
3.2.3.5	MYSQL . . . . .	56
3.2.3.6	StarUML . . . . .	57
3.2.3.7	Canva . . . . .	57
3.2.4	Matériel utilisé . . . . .	57
3.3	Présentation des interfaces de notre application web . . . . .	58
3.3.1	Page d'accueil . . . . .	58
3.3.2	Annonces récentes . . . . .	58
3.3.3	Agents immobiliers . . . . .	59
3.3.4	Footer de l'accueil . . . . .	59
3.3.5	Page contact . . . . .	60
3.3.6	Page d'authentification . . . . .	60
3.3.7	Page gestion des mots de passe . . . . .	61
3.3.8	Page vérification de l'adresse e-mail . . . . .	61
3.3.9	Espace d'accueil pour gestionnaire . . . . .	62

---

3.3.10	Page ajouter un utilisateur . . . . .	62
3.3.11	Page ajouter un bien . . . . .	63
3.3.12	Page gestion des utilisateurs . . . . .	63
3.3.13	Page gestion des biens . . . . .	64
3.3.14	Page des annonces . . . . .	64
3.3.15	Page détails et offres annonce . . . . .	65
3.3.16	Notification d'une offre . . . . .	65
3.3.17	Page prendre un rendez-vous . . . . .	66
3.3.18	Notification d'un rendez-vous . . . . .	66
3.3.19	Page gestion des visites . . . . .	67
3.3.20	Visites des agents immobiliers . . . . .	67
3.3.21	Page détail d'un contrat . . . . .	68
3.3.22	Impression page détail d'un contrat . . . . .	68
3.4	Présentation des interfaces de notre application mobile . . . . .	69
3.4.1	Page d'accueil . . . . .	69
3.4.2	Annonces récentes et liste des agents immobiliers . . . . .	69
3.4.3	Page des annonces . . . . .	70
3.4.4	Page contact . . . . .	70
3.4.5	Page offre . . . . .	71
3.5	Conclusion . . . . .	72
<b>Annexes</b>		<b>1</b>
3.6	Sélection du Langage de Modélisation . . . . .	1
3.7	Historique de l'UML . . . . .	1
3.8	Définition de UML . . . . .	2
3.8.1	Liste des types de diagrammes UML . . . . .	2
3.9	Description approfondie de certains diagrammes . . . . .	5
3.9.1	Diagramme de cas d'utilisation . . . . .	5
3.9.2	Diagramme de classes . . . . .	7
3.9.3	Diagramme de séquence . . . . .	7
3.9.4	Diagramme d'activités . . . . .	8
3.9.4.1	Avantages des diagrammes d'activités . . . . .	8
3.9.4.2	Pourquoi utiliser des diagrammes d'activités ? . . . . .	9
3.9.4.3	Composants de base d'un diagramme d'activités . . . . .	9
<b>Bibliographie</b>		<b>4</b>

# Table des figures

1.1	Présentation de l'entreprise EURL BEINX ALGERIE . . . . .	2
1.2	Organigramme de l'entreprise . . . . .	3
1.3	Site web Ouedkniss . . . . .	4
1.4	Site web Elkeria . . . . .	5
1.5	Logo . . . . .	7
1.6	La palette de couleur . . . . .	8
1.7	Carte de visite . . . . .	8
2.1	Déroulement D'UP . . . . .	14
2.2	différentes vues de l'architecture d'un projet . . . . .	14
2.3	L'architecture bidirectionnelle . . . . .	15
2.4	Les phases D'UP . . . . .	17
2.5	Diagramme de contexte . . . . .	19
2.6	Diagramme de cas d'utilisation de l'acteur «Utilisateur» . . . . .	21
2.7	Diagramme de de cas d'utilisation de l'acteur «Client» . . . . .	21
2.8	Diagramme de cas d'utilisation de l'acteur «Gestionnaire» . . . . .	22
2.9	Diagramme de cas d'utilisation de l'acteur «Agence» . . . . .	22
2.10	Diagramme de cas d'utilisation de l'acteur «Agent immobilier» . . . . .	23
2.11	Diagramme cas d'utilisation globale . . . . .	31
2.12	Diagramme de séquence «S'authentifier» . . . . .	32
2.13	Diagramme de séquence «ajouter un utilisateur» . . . . .	33
2.14	Diagramme de séquence «Modifier un utilisateur» . . . . .	34
2.15	Diagramme de séquence «Supprimer un utilisateur» . . . . .	35
2.16	Diagramme de séquence «Rechercher» . . . . .	36
2.17	Diagramme de séquence «Ajouter une annonce» . . . . .	37
2.18	Diagramme de séquence «Modifier une annonce» . . . . .	38
2.19	Diagramme de séquence «Supprimer une annonce» . . . . .	39
2.20	Diagramme de séquence «Contacter le gestionnaire» . . . . .	40

---

2.21	Diagramme de séquence «Effectuer une offre» . . . . .	41
2.22	Diagramme d'activité «S'authentifier» . . . . .	42
2.23	Diagramme d'activité «ajouter un utilisateur» . . . . .	43
2.24	Diagramme d'activité «Modifier un utilisateur» . . . . .	44
2.25	Diagramme d'activité «Supprimer un utilisateur» . . . . .	45
2.26	Diagramme de classes . . . . .	47
3.1	Architecture MVC . . . . .	53
3.2	«API REST» . . . . .	56
3.3	Page d'accueil . . . . .	58
3.4	Annonces récentes . . . . .	58
3.5	Agents immobiliers . . . . .	59
3.6	Footer de l'accueil . . . . .	59
3.7	Page contact . . . . .	60
3.8	Page d'authentification . . . . .	60
3.9	Page mot de passe oublié . . . . .	61
3.10	Page de réinitialisation . . . . .	61
3.11	Page vérification de l'adresse e-mail . . . . .	61
3.12	Espace d'accueil pour gestionnaire . . . . .	62
3.13	Page ajouter un utilisateur . . . . .	62
3.14	Page ajouter un bien . . . . .	63
3.15	Page gestion des utilisateurs . . . . .	63
3.16	Page gestion des biens . . . . .	64
3.17	Page des annonces . . . . .	64
3.18	Page détails et offres annonce . . . . .	65
3.19	Notification d'une offre . . . . .	65
3.20	Page prendre un rendez-vous . . . . .	66
3.21	Notification gestionnaire . . . . .	66
3.22	Notification agent . . . . .	66
3.23	Gestion des visites . . . . .	67
3.24	Gestion des visites . . . . .	67
3.25	Page détail d'un contrat . . . . .	68
3.26	Impression page détail d'un contrat . . . . .	68
3.27	Page d'accueil . . . . .	69
3.28	Annonces récentes . . . . .	69
3.29	Liste des agents immobiliers . . . . .	69

---

3.30	Page des annonces . . . . .	70
3.31	Page contact . . . . .	70
3.32	Page offre . . . . .	71
3.33	Histoire UML . . . . .	1
3.34	Diagrammes UML . . . . .	4
3.35	Acteur . . . . .	5
3.36	Un cas d'utilisation . . . . .	5
3.37	Interaction . . . . .	6
3.38	Inclusion . . . . .	6
3.39	Extension . . . . .	6
3.40	Généralisation . . . . .	7
3.41	Une classe avec ses attributs et ses opérations . . . . .	7
3.42	Diagramme de séquence . . . . .	8
3.43	Exemple d'un diagramme d'activités . . . . .	9

# Liste des tableaux

2.1	Tableau des cas d'utilisation . . . . .	20
2.2	Fiche descriptive du cas d'utilisation «Ajouter un utilisateur» . . . . .	24
2.3	Fiche descriptive du cas d'utilisation «S'authentifier» . . . . .	25
2.4	Fiche descriptive du cas d'utilisation «Rechercher» . . . . .	26
2.5	Fiche descriptive du cas d'utilisation «Ajouter annonce» . . . . .	27
2.6	Fiche descriptive du cas d'utilisation «Modifier annonce» . . . . .	28
2.7	Fiche descriptive du cas d'utilisation «Supprimer annonce» . . . . .	29
2.8	Fiche descriptive du cas d'utilisation «Gérer les visites» . . . . .	30
2.9	Dictionnaire de données pour les attributs . . . . .	46

# Introduction Générale

Dans un monde en constante évolution, où l'informatique est au cœur de nombreuses industries, les applications web et mobiles ont révolutionné la manière dont les entreprises gèrent leurs activités et interagissent avec leurs clients. Parmi ces secteurs d'activité, l'immobilier se distingue comme un pilier fondamental de l'économie, où la gestion efficace des biens et des transactions est cruciale pour le succès des agences immobilières.

C'est dans cette perspective que s'inscrit notre mémoire : la création d'une nouvelle application web et mobile dédiée à la gestion d'une agence immobilière. Notre objectif est de fournir une solution complète permettant aux agences immobilières de gérer leur portefeuille de biens immobiliers tout en offrant une expérience utilisateur optimale à leurs clients, qu'ils soient acheteurs, vendeurs ou locataires.

Notre application offre une gamme complète de fonctionnalités, allant de la publication d'annonces immobilières à la gestion des visites, des offres et des transactions, en passant par la gestion des demandes clients et leurs suivi. Les utilisateurs auront également accès à des outils avancés tels que la gestion des données clients et des contrats personnalisés.

La réalisation de ce mémoire est organisée en trois chapitres distincts, chacun abordant un aspect essentiel du développement de notre application :

- Le premier chapitre présente l'organisme d'accueil et propose une étude approfondie de l'existant, permettant ainsi de comprendre les besoins spécifiques des agences immobilières et des clients. Puis, nous avons présenté le projet et avons établi un cahier des charges, définissant les spécifications fonctionnelles et non fonctionnelles de notre application.
- Le deuxième chapitre explore la modélisation et la conception de notre application, en mettant en lumière les choix technologiques et architecturaux qui sous-tendent son développement.
- Enfin, le troisième et dernier chapitre présente la réalisation concrète de notre application, mettant en avant les différentes fonctionnalités implémentées et les défis rencontrés lors du développement.

# Chapitre 1

## Présentation de l'organisme d'accueil et étude de l'existant

### 1.1 Introduction

Le passage de la théorie à la pratique est un jalon essentiel dans le parcours de tout étudiant. Dans cette optique, ce rapport de projet de fin de cycle documente notre immersion dans le monde professionnel au sein de la EURL BEINX ALGERIE où nous avons eu l'opportunité de mettre en pratique les connaissances acquises au cours de nos études. Dans ce chapitre, après avoir présenté la EURL BEINX ALGERIE et critiqué l'existant dans le marché algérien, nous aborderons la problématique liée à la gestion des agences immobilières, la présentation de notre application dédiée à cette gestion, son titre, son logo, la conception de notre carte de visite ainsi que ses objectifs et fonctionnalités.

### 1.2 Présentation de l'EURL BEINX ALGERIE

EURL BEINX ALGERIE, dirigée par Mr Yous Athmane, Fondée en 2012 est une entreprise dynamique localisée à Béjaïa, en Algérie. Elle se distingue principalement dans le domaine des logiciels et propose une vaste gamme de solutions informatiques pour divers besoins professionnels. Voici un aperçu des services offerts par l'entreprise :

- Logiciels systèmes.
- Logiciels de bases de données et de gestion de l'information.
- Logiciels de création éditoriale.
- Logiciels de bureautique.
- Logiciels de gestion de l'entreprise.



FIGURE 1.1 – Présentation de l'entreprise EURL BEINX ALGERIE

- Logiciels pour l'industrie.
- Ingénierie informatique (SSII/ESN).

L'EURL BEINX ALGERIE se positionne ainsi comme un partenaire de confiance pour les entreprises cherchant à optimiser leurs processus grâce à des solutions logicielles efficaces et personnalisées [1].

### 1.2.1 Effectifs de l'entreprise

L'EURL BEINX ALGERIE compte entre 10 et 19 employés, constituant ainsi une équipe dynamique et compétente pour répondre aux besoins de ses clients.

### 1.2.2 Activités commerciales

- Commerce de détail de machines - matériel et mobilier de bureau.
- Vente au détail de matériel informatique : ordinateurs - périphériques - consommables et accessoires.
- Commerce de détail de la librairie et papeterie.
- Bureau d'étude et de conseil en informatique (consulting).
- Installation de réseaux et traitement de données [2].

### 1.3 Organigramme de l'entreprise

Un organigramme d'entreprise est un outil interne qui permet d'indiquer l'organisation hiérarchique du personnel [3].

Beinx est organisé comme suite :

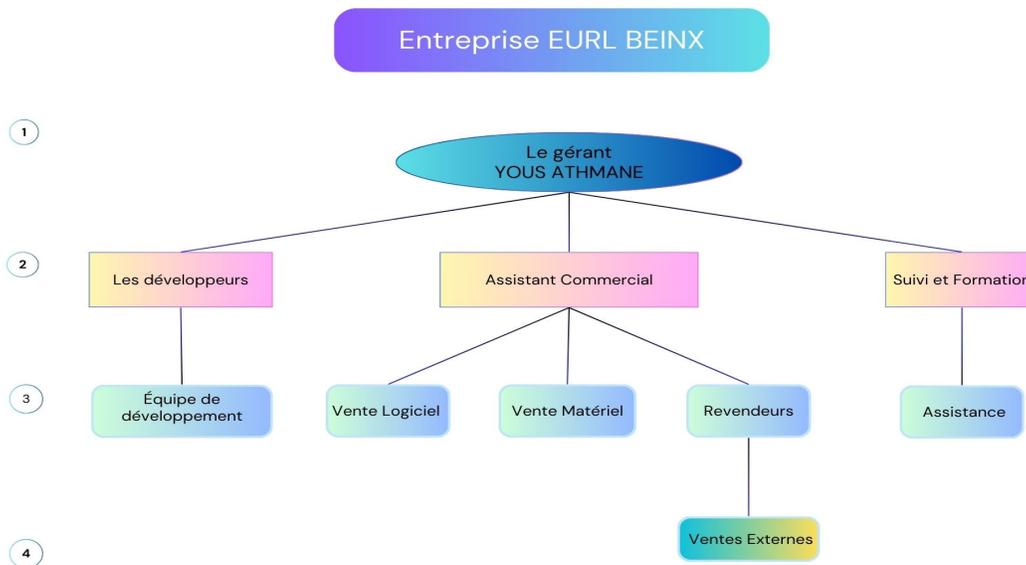


FIGURE 1.2 – Organigramme de l'entreprise

### 1.4 Étude et critique de l'existant

Avant d'aborder la conceptualisation de notre projet, il est essentiel d'examiner attentivement les projets similaires. Nous visons à exploiter leurs points forts, à atténuer leurs faiblesses et à proposer des solutions novatrices adaptées au contexte immobilier algérien.

Nous avons analysé plusieurs applications web de gestion des agences immobilières en Algérie, parmi lesquelles Ouedkniss, Elkeria, Darna, Dardjadida, Algeriahome et Algerieannonce ont attiré notre attention. Cette étude nous a permis d'identifier les pratiques existantes, leurs avantages et leurs lacunes.

En se fondant sur cette analyse, nous serons en mesure de concevoir une application qui répondra efficacement aux besoins du marché immobilier algérien, en proposant des fonctionnalités innovantes et une expérience utilisateur optimisée.

Après inscription et essai des fonctionnalités de Ouedkniss et Elkeria, voici nos observations :

### 1.4.1 Ouedkniss :

Les avantages d'utiliser Ouedkniss :

- Large sélection.
- Connexion avec les vendeurs locaux.
- Achat sécurisé.
- Conception adaptative.
- Annonces gratuites pour les utilisateurs individuels.

Les inconvénients d'utiliser Ouedkniss :

- Qualité incertaines des articles.
- Fraude potentielle.
- Limitation de livraison.
- Annonces payantes pour les entreprises [4].

Cette figure présente une capture d'écran de la section immobilière de Ouedkniss :

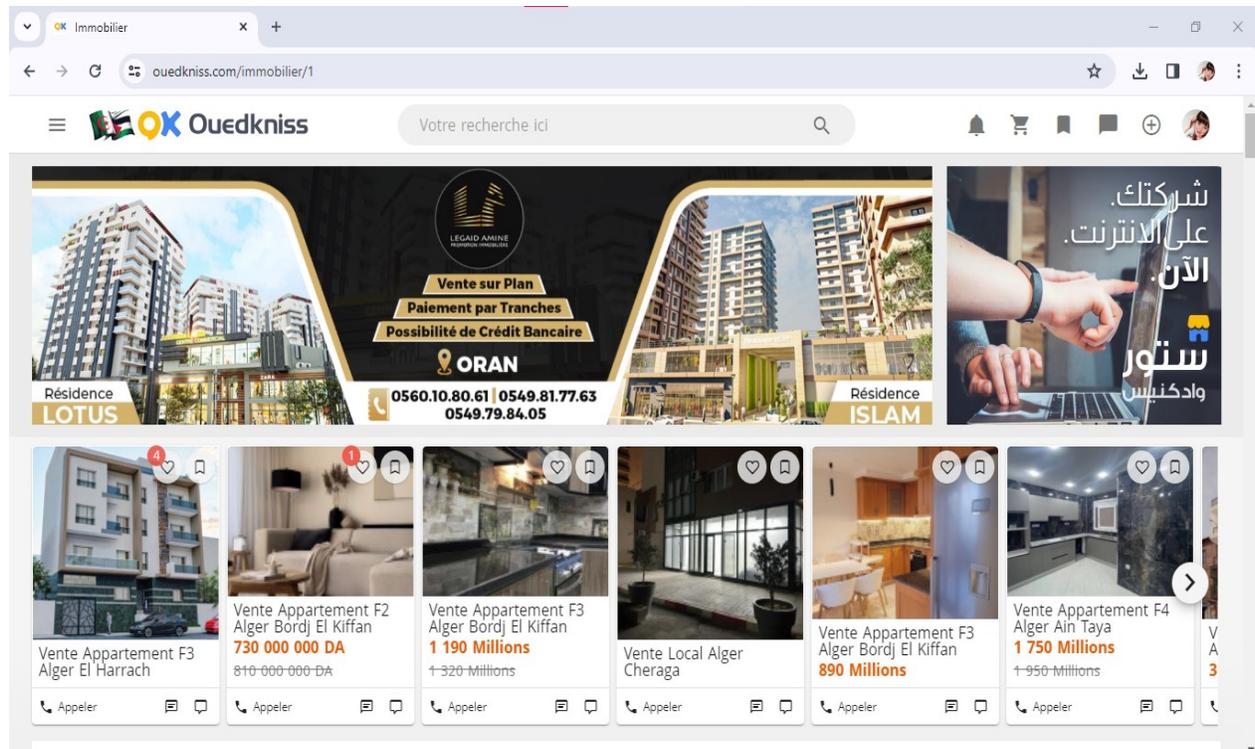


FIGURE 1.3 – Site web Ouedkniss

### 1.4.2 Elkeria :

Les avantages d'utiliser Elkeria :

- Premier portail d'annonce immobilière en Algérie.
- Design adaptative.
- Les particuliers peuvent publier des annonces sans avoir à s'inscrire.
- Transactions d'achat et de location en toute sécurité [5].

Les inconvénients d'utiliser Elkeria :

- Pour publier les annonces, les professionnels doivent payer un abonnement mensuel ou annuel avec deux offres : Essentiel ou Premium. Concernant les particuliers, ils doivent également payer la publication de leurs annonces ( l'annonce de 3 mois coûte 500 DA et celle de 6 mois à 1000 DA. Il faut effectuer un virement CCP ) [6].

Cette figure illustre une capture d'écran de Elkeria :

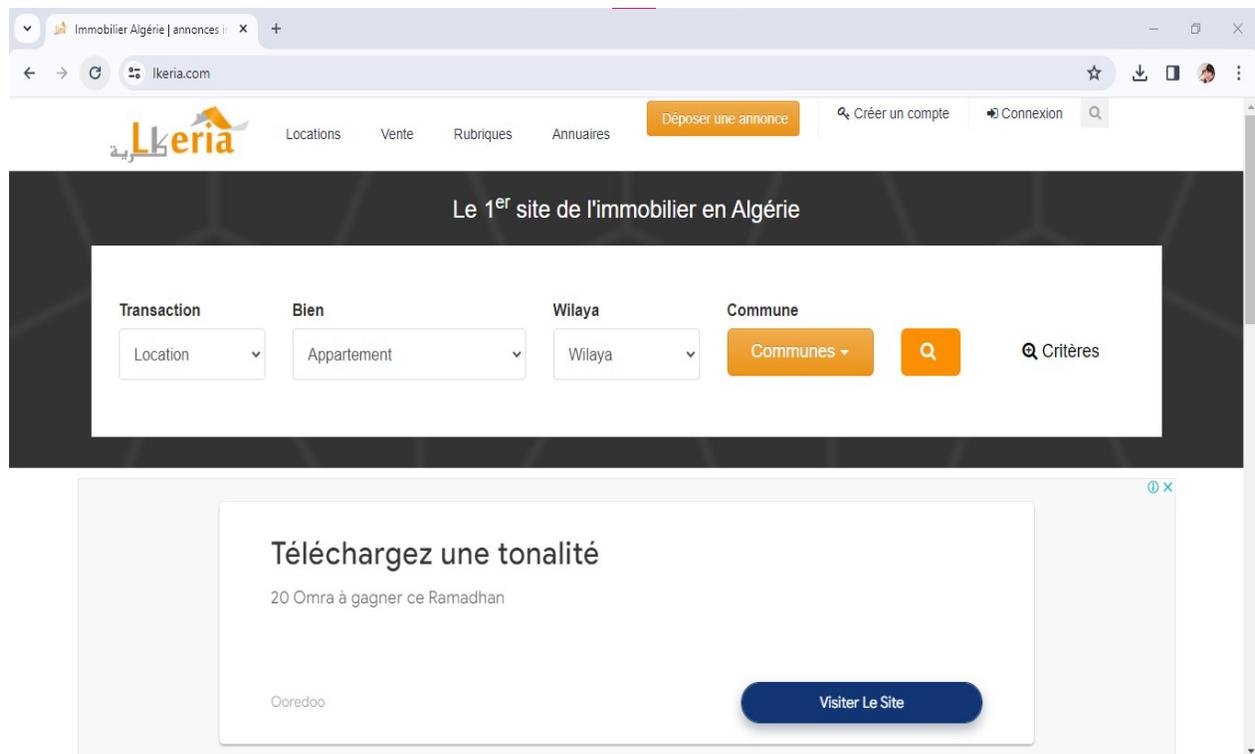


FIGURE 1.4 – Site web Elkeria

## 1.5 Problématique

Le marché immobilier est en constante évolution et la gestion des agences immobilières est devenue de plus en plus complexe. Dans ce contexte, les entreprises du secteur doivent faire preuve d'une grande efficacité pour rester compétitives.

Cette étude vise à pallier les lacunes des plateformes immobilières actuelles en Algérie en proposant une application web et mobile dédiée à la gestion des agences immobilières. Actuellement, les plateformes existantes imposent des frais élevés pour la publication d'annonces, rendant le service coûteux et limitant son accessibilité. De plus, la gestion des visites et des transactions n'est pas intégrée, ce qui complique le suivi pour les agences. Les systèmes de notification inefficaces nuisent à l'interaction et à la communication avec les utilisateurs, tandis que la recherche d'informations spécifiques reste difficile et laborieuse. L'archivage défaillant des données entraîne la perte d'informations importantes, et l'accès aux données sur les biens et les utilisateurs est souvent complexe et inefficace. Ainsi, la problématique de cette étude est de savoir comment développer une application web et mobile capable de surmonter ces défis.

C'est dans ce cadre que s'inscrit notre projet de gestion d'une agence immobilière, qui vise à proposer une solution innovante pour aider les entreprises à gérer efficacement leur activité.

## 1.6 Présentation de l'application

Nous proposons une application web et mobile dédiée à la gestion d'agences immobilières. Notre plateforme offre une expérience complète pour la publication, la recherche et la gestion d'annonces immobilières. Les utilisateurs peuvent publier différents types d'annonces, tels que la vente, l'achat, la location et l'échange de biens immobiliers. Pour faciliter la recherche et la gestion des annonces, notre application propose des fonctionnalités avancées de recherche, ainsi que des outils pour faire des offres sur une propriété et contacter directement les agences immobilières.

### Titre de l'application : **Dream House**

**Dream House** : Nous avons choisi le nom "Dream House", qui signifie "maison de rêve". Ce nom a été sélectionné pour évoquer l'idée de la maison idéale que chaque individu aspire à posséder. "Dream" souligne notre volonté d'aider les utilisateurs à réaliser leurs rêves immobiliers, tandis que "House" met en avant l'objectif concret de l'application dans le domaine de l'immobilier. Ensemble, ces termes visent à connecter les utilisateurs avec les biens immobiliers de leurs rêves.

### Logo :

Notre logo présente une maison élégante avec l'inscription "Dream House". Cette combinaison simple et élégante illustre parfaitement le thème immobilier de notre application, tout en reflétant l'aspiration de chaque individu à trouver sa maison idéale.



FIGURE 1.5 – Logo

### La palette de couleur :

Cette palette a été soigneusement choisie pour rappeler le monde de l'immobilier et garantir une cohérence visuelle à notre application de gestion d'agence immobilière.

**Noir (111111)** : Inspiré de la couleur du sol, le noir symbolise la fondation solide et la stabilité, la base solide sur laquelle repose chaque bâtiment et maison.

**Gris foncé (222222)** : Inspire du ciment, cette teinte représente la structure et la solidité, essentielles dans la construction des immeubles et des maisons.

**Gris clair (999999)** : Rappelle les nuances de béton, évoquant la modernité et la simplicité, reflétant le style moderne de l'architecture immobilière.

**Bleu (10B1D0)** : Il est largement utilisé dans le domaine immobilier pour évoquer la sérénité, la confiance et la sécurité, tout en rappelant le ciel sous lequel se dressent les bâtiments.

**Blanc (FFFFFF) :** Comme la couleur de nombreux intérieurs et façades, Représente la pureté, la clarté et la luminosité.

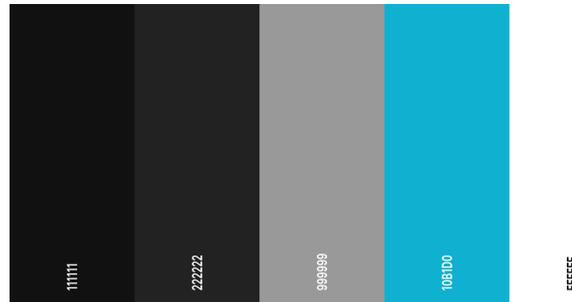


FIGURE 1.6 – La palette de couleur

### Carte de visite :

Dream House, votre partenaire pour vos besoins immobiliers en Algérie. Spécialistes en gestion immobilière, nous vous proposons des services adaptés et une bonne connaissance pour réaliser vos projets immobiliers.



FIGURE 1.7 – Carte de visite

### 1.6.1 Objectifs

- Faciliter la gestion des biens immobiliers pour les propriétaires et les gestionnaires.
- Améliorer la recherche de biens immobiliers pour les locataires et les acheteurs.
- Offrir un service personnalisé et de qualité aux utilisateurs.
- Améliorer la communication entre les différentes parties prenantes.
- Proposer la gratuité des annonces pour tous les utilisateurs, qu'ils soient individuels ou professionnels.
- Faciliter la gestion des visites et des offres pour les professionnels de l'immobilier.
- Offre un système de notification efficace pour une meilleure interaction avec les utilisateurs.
- Un accès rapide aux informations.
- Assure un archivage fiable des données.
- Un gain important en temps.
- Garantir la sécurité et la confidentialité des données personnelles et financières des utilisateurs.

### 1.6.2 Les besoins fonctionnels et non fonctionnels

**Les besoins fonctionnels :** Un besoin fonctionnel décrit ce qu'un produit ou service est censé faire, c'est-à-dire ses fonctions principales. Il s'agit de spécifications concrètes qui détaillent les opérations, les fonctionnalités ou les tâches que le produit ou le service doit être capable d'exécuter [7]. Ainsi, nous avons défini les besoins spécifiques pour chaque acteur en fonction de son rôle au sein de l'application :

**Utilisateurs :** Ceux qui visitent l'application pour trouver des biens immobiliers peuvent :

- Consulter et parcourir les annonces disponibles sur le site.
- Effectuer des recherches spécifiques selon leurs critères (type de bien, transaction, surface prix).
- Contacter le gestionnaire .
- Voir la liste des agents immobiliers travaillant dans l'agence.
- Effectuer une offre.

**Clients :** Il s'agit d'utilisateurs sur l'application, pouvant être des acheteurs, des vendeurs ou des locataires. Ils ont la possibilité de :

- Parcourir les annonces disponibles sur le site.
- Planifier des visites pour les propriétés qui l'intéressent.

- Effectuer des recherches de biens selon ses critères.
- Communiquer directement avec le gestionnaire ou l'agence pour publier son annonce et le fournir les informations nécessaires sur la propriété.
- Effectuer une offre sur un bien immobilier.

**Gestionnaire :** L'acteur responsable de la gestion des annonces et des propriétés au sein de l'application. Ses principales actions comprennent :

- Créer de nouvelles annonces, les mettre à jour ou les supprimer selon les besoins.
- Suivre le déroulement des visites.
- Échanger régulièrement avec les clients et les utilisateurs pour répondre à leurs questions et leurs besoins.
- Gérer les comptes des agences, des agents immobiliers et des clients, y compris leur création, leur modification et leur suppression.
- Générer le contrat.

**L'agence :** L'agence immobilière, sous la direction de son gestionnaire. Après s'être authentifiée, l'agence dispose des autorisations suivantes :

- L'ajout de nouveaux biens à la liste de l'agence permet d'avoir plus de choix et de variété.
- La modification des informations des biens existants offre une flexibilité essentielle pour mettre à jour et améliorer leur présentation.
- La suppression des biens qui ne sont plus disponibles ou pertinents garantit une base de données précise et actualisée.
- Générer le contrat.

**Agents immobiliers :** Les agents immobiliers facilitent les transactions entre les vendeurs de biens immobiliers et les acheteurs ou les locataires. Ils organisent les visites, négocient les prix et les conditions, et aident les clients à finaliser les transactions.

- Avoir un compte pour accéder aux fonctionnalités du site nécessaires à l'exercice de leurs fonctions d'agent immobilier.
- Organiser les visites des biens immobiliers pour les clients intéressés.

**Les besoins non fonctionnels :** Les spécifications non fonctionnelles sont toutes des spécifications qui n'expriment pas une fonction du logiciel. Elles caractérisent le système et sont des besoins en matière de performances, matériels, contraintes d'implémentation, systèmes d'exploitation [8]. Ci-dessous, les critères non fonctionnels de notre application :

- **La sécurité et fiabilité :** Les applications gérant des données sensibles doivent

respecter des normes de sécurité strictes (pour protéger les informations des utilisateurs contre tout accès non autorisé, résister aux tentatives de hacking, se remettre en état simplement. . .).

- **La performance** : La performance doit être choisie, et non subie. Voici quelques exemples de spécifications non fonctionnelles liées à la performance ( la capacité en nombre de visiteurs en simultané, le temps de chargement et la rapidité d'exécution, le nombre de langues supportées dans l'application. . .).
- **L'ergonomie** : L'utilisation de l'application doit être simple et claire. Une interface utilisateur intuitive et bien conçue améliore l'expérience globale des utilisateurs.
- **La disponibilité** : Selon les besoins des utilisateurs, certaines applications, telles que les plateformes bancaires en ligne, doivent être disponibles 24h/24 et 7j/7. Pensez à prévoir un contrat d'infogérance pour garantir une disponibilité optimale.
- **La maintenance** : Pour que votre application soit pérenne, sa maintenance est nécessaire (traçabilité des erreurs, mises à jour, testabilité. . .) elle doit être pensée pour éviter les bugs, crashes ou erreurs de chargement et pour mettre en place les évolutions logicielles [9].

## 1.7 Conclusion

Au cours de ce premier chapitre, nous avons procédé à une étude approfondie de l'existant en matière d'applications web dédiées à la gestion des agences immobilières. Cette analyse nous a permis d'identifier les lacunes et les insuffisances des plateformes existantes.

La présentation de l'EURL BEINX ALGERIE, notre organisme d'accueil, nous a offert un aperçu précieux de l'environnement professionnel dans lequel s'inscrit notre projet. Forts de ces constatations, nous avons pu approfondir la présentation détaillée de notre propre projet, où nous avons défini les spécifications et les fonctionnalités de l'application que nous envisageons de développer pour répondre aux besoins spécifiques du marché immobilier. Ainsi, nous avons dressé un aperçu complet des fonctionnalités et des spécifications de notre application, illustrant sa capacité à répondre aux exigences diversifiées des utilisateurs du secteur immobilier. En détaillant les fonctionnalités clés, telles que la gestion des propriétés, la recherche personnalisée, ainsi que les besoins non fonctionnels comme la sécurité des données, les performances et les contraintes d'implémentation, nous avons établi une fondation robuste pour notre solution. Cette analyse approfondie nous a permis de saisir les attentes et les exigences des utilisateurs.

Dans le prochain chapitre, nous introduirons la méthode agile pour le développement de notre application. Nous entamerons également la phase de conception, où nous concrétiserons nos idées en un design fonctionnel et intuitif, assurant ainsi une expérience utilisateur optimale.

# Chapitre 2

## Modélisation et conception de projet

### 2.1 Introduction

Ce deuxième chapitre se concentre sur la conception et la modélisation de notre application, en mettant en lumière notre approche basée sur le processus unifié (UP). Après avoir introduit la méthode Agile comme choix de développement pour notre projet, nous explorerons l'utilisation du langage de modélisation UML (Unified Modeling Language) et ses divers diagrammes. Nous débuterons par l'identification des acteurs clés du système et leurs interactions, illustrées à travers les scénarios d'utilisation. Cette démarche nous orientera dans l'élaboration des diagrammes de cas d'utilisation, de séquence et de classe, nous permettant ainsi de détailler les aspects fonctionnels, dynamiques et statiques de notre application.

### 2.2 Choix de la méthodologie de modélisation

Il existe plusieurs méthodes de développement logiciel construites sur UML comme la méthode : UP, RUP, TTUP, UP agile, XP, 2TUP ..... Parmi ses méthodes notre choix est basé sur la méthode UP (Unified Process).

### 2.3 Processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

L'objet 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.

### 2.3.1 Définition Processus Unifié (UNIFIED PROCESS)

Un processus unifié est un processus de développement logiciel construit sur UML ; il est itératif, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques. La gestion d'un tel processus est organisée suivant les 4 phases suivantes : pré-étude, élaboration, construction et transition. Ses activités de développement sont définies par 5 workflows fondamentaux qui décrivent la capture des besoins, l'analyse, la conception, l'implémentation et le test [10].

### 2.3.2 Les caractéristiques du processus unifié

**UP est itératif et incrémental** : Le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l'avancement globale. A la fin de chaque itération une partie exécutable du système finale est produite, de façon incrémentale.

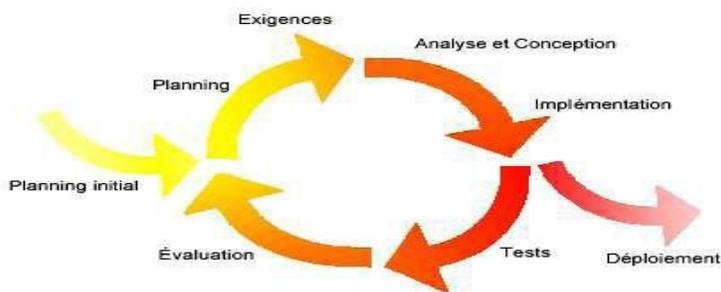


FIGURE 2.1 – Déroulement D'UP

**UP est centré sur l'architecture** : Tout système complexe doit être décomposé en partie modulaire afin d'en faciliter la maintenance et l'évolution. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modéliser en UML, et pas seulement documentée en texte.

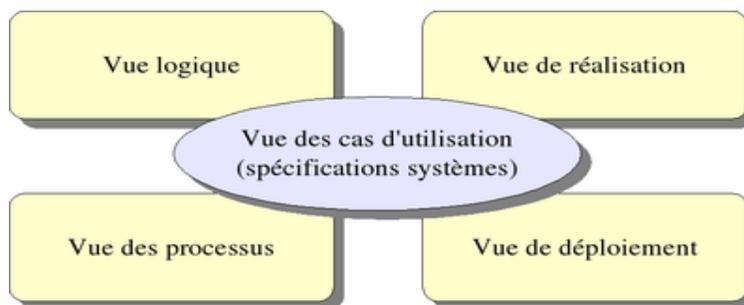


FIGURE 2.2 – différentes vues de l'architecture d'un projet

**UP est guidé par les cas d'utilisation d'UML :** 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.

**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 [11].

### 2.3.3 Cycle de vie du processus unifié

L'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. UP est un ensemble de principes génériques adapté en fonction des spécificités des projets.

**L'architecture bidirectionnelle :** UP gère le processus de développement par deux axes.

1. **L'axe vertical :** Représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en terme de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
2. **L'axe horizontal :** Représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons [12].

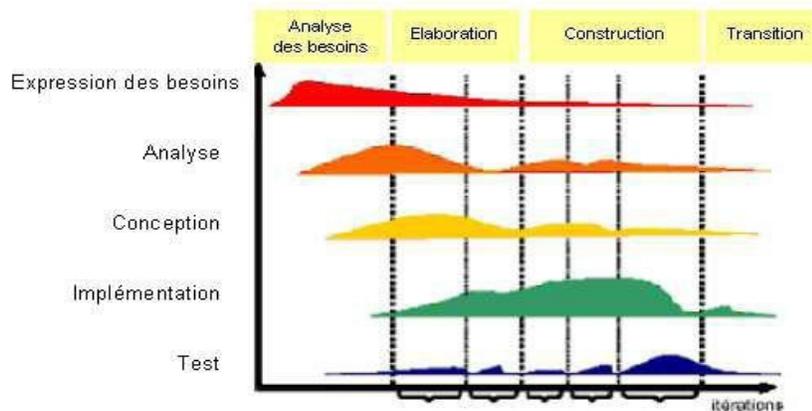


FIGURE 2.3 – L'architecture bidirectionnelle

UP répète un certain nombre de fois une série de cycles qui s'articulent autour de quatre phases :

- Analyse des besoins .

- Élaboration.
- Construction.
- Transition.

Pour mener efficacement un tel cycle, les développeurs ont besoin de toutes les représentations du produit logiciel :

- Un modèle de cas d'utilisation.
- Un modèle d'analyse : détailler les cas d'utilisation et procéder à une première répartition du comportement.
- Un modèle de conception : finissant la structure statique du système sous forme de sous-systèmes, de classes et interfaces.
- Un modèle d'implémentation : intégrant les composants.
- Un modèle de déploiement : définissant les nœuds physiques des ordinateurs.
- Un modèle de test : décrivant les cas de test vérifiant les cas d'utilisation.
- Une représentation de l'architecture.

### 2.3.4 Les enchaînements d'activités D'UP

**Expression des besoins** : L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

- Inventorier les besoins principaux et fournir une liste de leurs fonctions.
- Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation.
- Appréhender les besoins non fonctionnels (technique) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

**Analyse** : L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structures sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

**Conception** : La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation.

Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.

Elle constitue un point de départ à l'implémentation :

- Elle décompose le travail d'implémentation en sous-système.
- Elle crée une abstraction transparente de l'implémentation.

**Implémentation** : L'implémentation est le résultat de la conception pour implémenter le système sous forme de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous forme de codes sources.

**Test** : Les tests permettent de vérifier des résultats de l'implémentation en testant la construction.

Pour mener à bien ces tests, il faut les planifier pour chaque itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun [12].

### 2.3.5 Les phases et les disciplines du UP

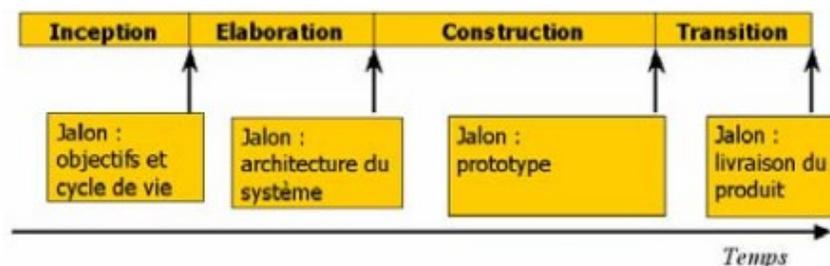


FIGURE 2.4 – Les phases D'UP

**Inception** : La phase d'inception a pour but de décrire la vision du produit final, de réaliser une étude de rentabilité et de risques et de définir le projet. Les activités principales de cette phase sont :

- Formuler l'ampleur du projet, c'est-à-dire capturer le contexte, les exigences et les contraintes les plus importantes de façon à pouvoir définir des critères acceptables pour le produit fini.
- Planifier et préparer un business case et évaluer les alternatives du point de vue du risque, du staff, de la planification, du coût, du profit, . . .
- Synthétiser une architecture et évaluer les coûts, plans et ressources nécessaires.

La phase d'inception se termine par le jalon « objectifs et cycle de vie ».

**Élaboration** : La phase d'élaboration a pour but d'analyser le domaine du problème, de construire l'architecture de base, de résoudre les éléments à haut risque et de définir la plupart des exigences. Il s'agit de la phase la plus critique du projet. A la fin de celle-ci, le travail d'engineering est terminé et l'on peut prendre la décision d'accomplir les phases de construction et de transition du projet. Les activités principales de cette phase sont :

- Elaborer une vision des cas d'utilisation les plus critiques.
- Le processus, l'infrastructure et l'environnement de développement sont élaborés et le processus, les outils et le support mis en place.
- L'architecture est élaborée et les composants sont sélectionnés et intégrés dans un scénario primaire. Les leçons tirées de ces activités peuvent résulter en un changement d'architecture.

La phase d'élaboration se termine par le jalon « architecture du système ».

**Construction** : Il s'agit de construire les fonctionnalités qui ne font pas partie de l'architecture de base et préparer le déploiement. Les activités principales de cette phase sont :

- La gestion de ressources, le contrôle de ressources et l'optimisation de processus.
- Le développement complet de composantes et le test par rapport aux critères d'évaluation définis .
- L'appréciation des produits par rapport aux visions définies.

La phase de construction se termine par le jalon « prototype ».

**Transition** : Le but de la phase de transition est de procurer le logiciel à la communauté d'utilisateurs. Après que le produit ait été donné à l'utilisateur, il est généralement nécessaire de développer de nouvelles versions, de corriger certains problèmes ou de finir certaines choses postposées. Il s'agit d'effectuer les bêtas tests pour valider le nouveau système auprès des utilisateurs et le déploiement de celui-ci. La phase de transition se termine par le jalon « livraison du produit » ou par une nouvelle itération [13].

## 2.4 Analyse et conception

### 2.4.1 Identification des acteurs

Dans notre projet de conception et de réalisation d'une application web de gestion d'agence immobilière, nous avons identifié les acteurs clés suivants :

**Utilisateurs :** Ce sont les personnes qui utilisent l'application pour trouver des biens immobiliers. Ils peuvent parcourir les annonces, effectuer des recherches selon leurs critères, soumettre une offre et contacter le gestionnaire.

**Clients :** Ce sont des personnes intéressées par des biens immobiliers qui souhaitent effectuer des offres sur des annonces, ils peuvent contacter le gestionnaire pour publier leurs annonces, consulter les annonces disponibles, planifier des visites avec les agents immobiliers après s'être authentifié.

**Gestionnaire :** Il s'agit de l'entité responsable de la gestion des biens immobiliers et la gestion des utilisateurs sur l'application.

**L'agence :** L'agence peut gérer les biens immobiliers en ajoutant, modifiant ou supprimant des annonces.

**Agents immobiliers :** Leur rôle comprend l'organisation des visites, la négociation des conditions et des prix de vente ou de location, ainsi que l'assistance aux clients pour conclure les transactions.

## 2.5 Diagramme de contexte

Le diagramme de contexte est une modélisation permettant de mettre en avant les acteurs à haut niveau du système [25].

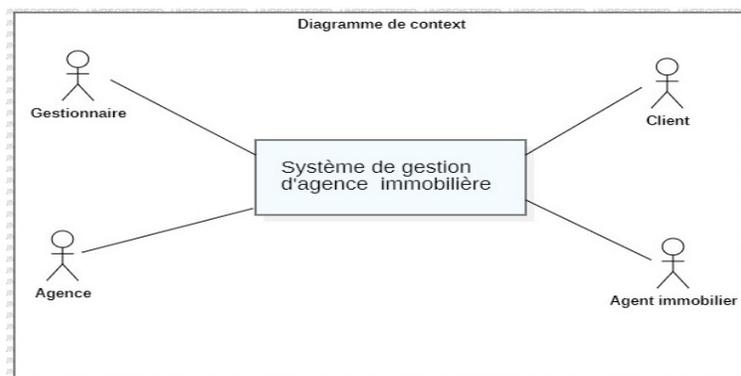


FIGURE 2.5 – Diagramme de contexte

## 2.6 Diagramme de cas d'utilisation

Cette section vise à modéliser les diagrammes de cas d'utilisation pour chacun des quatre acteurs du système, fournissant ainsi une vision complète de son fonctionnement. Nous avons commencé par identifier les différents cas d'utilisation pour chaque acteurs et nous les présentons dans le tableau 2.1.

N°	Cas d'utilisation	Acteurs
1	Consulter les annonces	Utilisateur
2	Consulter les agents immobiliers	Utilisateur
3	Rechercher	Utilisateur
4	Effectuer une offre	Utilisateur
5	Contacter le gestionnaire de l'agence	Utilisateur
6	Planifier des visites	Client
7	S'authentifier	Gestionnaire, Agence, Agent immobilier, Client
8	Ajouter annonce	Gestionnaire, Agence
9	Modifier annonce	Gestionnaire , Agence
10	Supprimer annonce	Gestionnaire, Agence
11	Ajouter un utilisateur	Gestionnaire
12	Modifier un utilisateur	Gestionnaire
13	Supprimer un utilisateur (agence, Agent immobilier, Client)	Gestionnaire
14	Recevoir des notifications	Gestionnaire, Agent immobilier
15	Gérer les visites	Agent immobilier
16	Générer détail contrat	Gestionnaire, agence

TABLEAU 2.1 – Tableau des cas d'utilisation

### 2.6.1 Plan des interactions pour «Utilisateur»

L'utilisateur peut découvrir le site et explorer ses fonctionnalités. De plus, il a la possibilité de faire des recherches pour trouver des informations spécifiques. La figure 2.6 représente le diagramme de cas d'utilisation de "Utilisateur".

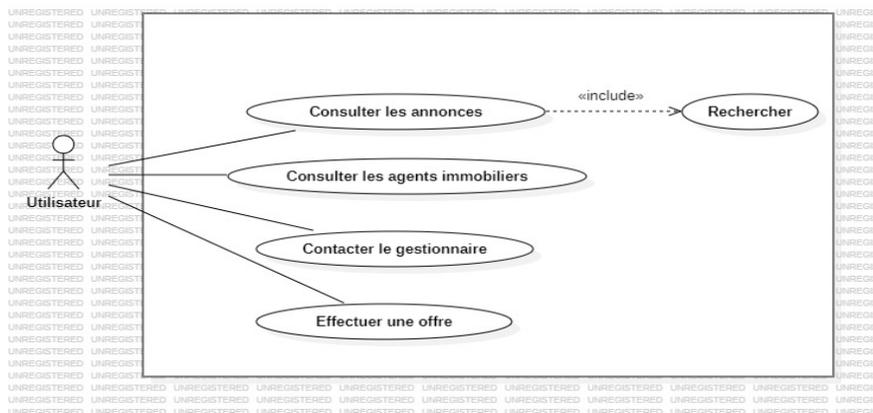


FIGURE 2.6 – Diagramme de cas d'utilisation de l'acteur «Utilisateur»

### 2.6.2 Plan des interactions pour «Client»

Les clients peuvent consulter les annonces, la liste des agents immobiliers sans authentification. Pour publier une annonce de location ou de vente, ils doivent contacter le gestionnaire, qui leur fournira les détails nécessaires sur la propriété avant de publier l'annonce. Les clients peuvent également effectuer une offre sur un bien immobilier et planifier des visites avec l'un des agents immobiliers. Ceci est illustré par le diagramme de la figure 2.7.

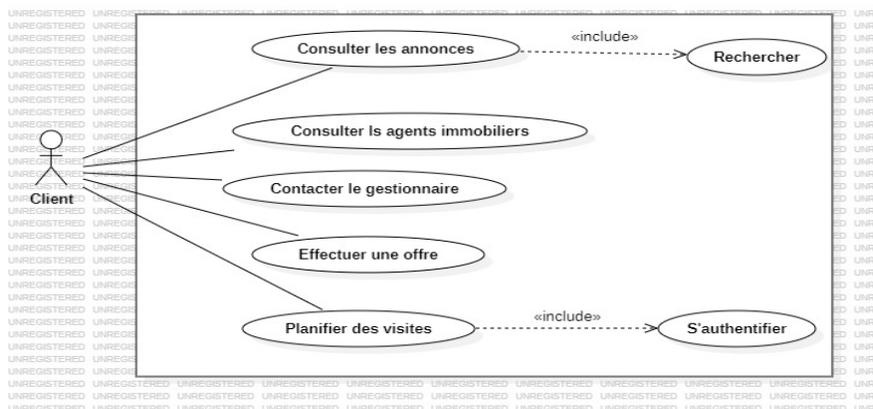


FIGURE 2.7 – Diagramme de cas d'utilisation de l'acteur «Client»

### 2.6.3 Plan des interactions pour «Gestionnaire»

Le gestionnaire assure une gestion efficace des annonces et des propriétés tout en restant en contact avec les clients. En outre, il administre les profils des utilisateurs. La figure 2.8 représente le diagramme de cas d'utilisation du gestionnaire.

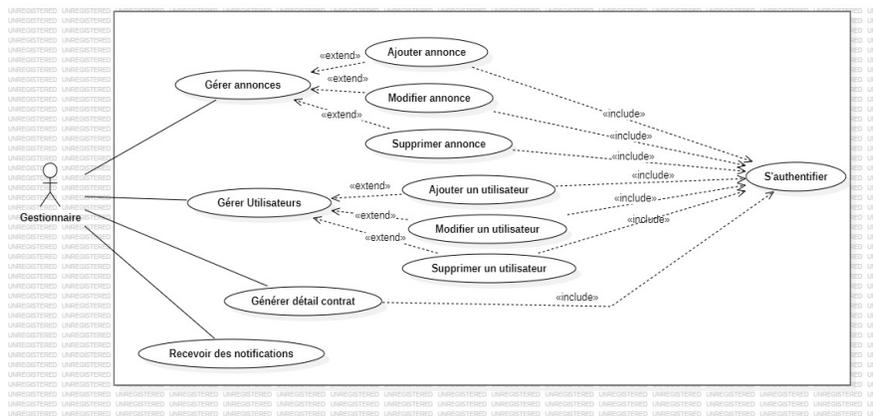


FIGURE 2.8 – Diagramme de cas d'utilisation de l'acteur «Gestionnaire»

### 2.6.4 Plan des interactions pour «Agence»

Sous la direction du gestionnaire, l'agence s'occupe des annonces sur le site. Elle les ajoute, les actualise et les retire si nécessaire. LA figure 2.9 montre ces interactions su le diagramme de cas d'utilisation.

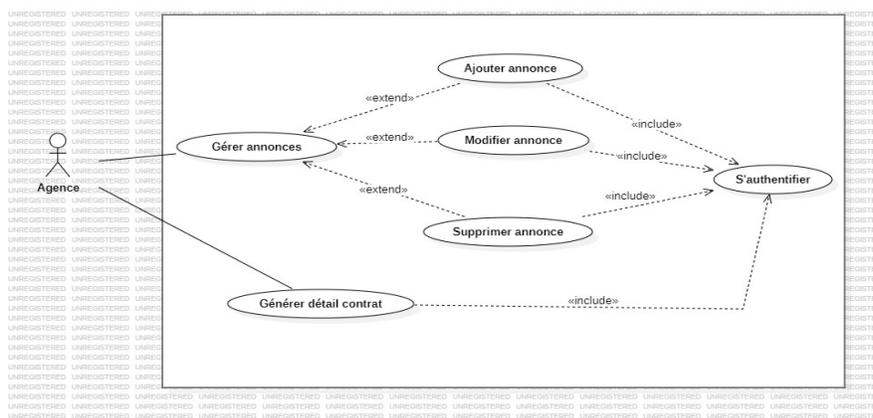


FIGURE 2.9 – Diagramme de cas d'utilisation de l'acteur «Agence»

### 2.6.5 Plan des interactions pour «Agent immobilier»

Les agents immobiliers prennent en charge les visites des biens immobiliers. Leur rôle consiste à accompagner les clients lors des visites, à répondre à leurs questions et à fournir des informations détaillées sur les propriétés. Ceci est illustré dans le digramme de la figure 2.10.

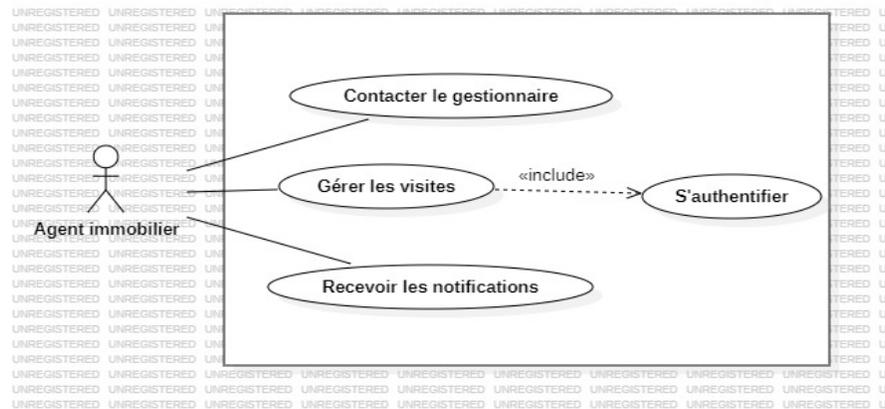


FIGURE 2.10 – Diagramme de cas d'utilisation de l'acteur «Agent immobilier»

## 2.7 Fiche descriptive

### 2.7.1 Fiche descriptive du cas d'utilisation «Ajouter un utilisateur»

<b>Ajouter un utilisateur</b>	
<b>Description</b>	Permet au gestionnaire de créer un compte pour un nouvel utilisateur (agence, agent immobilier, client) afin de lui donner accès à des fonctionnalités spécifiques du système.
<b>Acteur</b>	Gestionnaire du système.
<b>Pré-condition</b>	Le gestionnaire doit être authentifié et accéder à la page de gestion des utilisateurs.
<b>Post-condition</b>	Le nouvel utilisateur (agence, agent immobilier, client) possède un compte valide créé par le gestionnaire.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. Le gestionnaire accède à la page de gestion des utilisateurs.</li> <li>2. Le système présente un formulaire d'ajout d'utilisateur.</li> <li>3. Le gestionnaire saisit les informations requises dans le formulaire pour créer le compte de l'utilisateur.</li> <li>4. Le système valide les informations et crée le compte utilisateur.</li> <li>5. Le système confirme la création du compte et redirige le gestionnaire vers la page de gestion des utilisateurs.</li> </ol>
<b>Scénario Alternatif</b>	<ol style="list-style-type: none"> <li>1. Le gestionnaire saisit incorrectement les informations dans le formulaire.</li> </ol> <p>- Retour au scénario nominal étape 3.</p>
<b>Scénario d'exception</b>	Le gestionnaire quitte le processus d'ajout d'utilisateur avant sa finalisation.

TABLEAU 2.2 – Fiche descriptive du cas d'utilisation «Ajouter un utilisateur»

## 2.7.2 Fiche descriptive du cas d'utilisation «S'authentifier»

<b>S'authentifier</b>	
<b>Description</b>	Permet au gestionnaire, à l'agence, à l'agent immobilier et au client d'accéder au système.
<b>Acteur</b>	Gestionnaire, Agence, Agent immobilier, Client.
<b>Pré condition</b>	Chaque acteur doit posséder un compte valide.
<b>Post condition</b>	Chaque acteur est authentifié et a accès à son compte.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur saisit ses informations d'identification.</li> <li>2. Le système vérifie les informations et authentifie l'utilisateur.</li> <li>3. Le système affiche le compte de l'utilisateur avec succès.</li> </ol>
<b>Scénario Alternatif</b>	<p>Les informations saisies sont erronées.</p> <p>- Retour au scénario nominal 1.</p>
<b>Scénario d'exception</b>	L'acteur quitte le système sans s'authentifier.

TABLEAU 2.3 – Fiche descriptive du cas d'utilisation «S'authentifier»

### 2.7.3 Fiche descriptive du cas d'utilisation «Rechercher»

<b>Rechercher</b>	
<b>Description</b>	Permet au utilisateur de site de rechercher une fiche d'immobilier en fonction de critères spécifiques comme le type de propriété, la localisation et le prix.
<b>Acteur</b>	Utilisateur.
<b>Pré condition</b>	Des fiches d'immobilier doivent être disponibles dans le système.
<b>Post condition</b>	l'utilisateur visualise la liste des biens immobiliers correspondant aux critères spécifiés.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur accède à la page d'annonces.</li> <li>2. Le système affiche le formulaire de recherche.</li> <li>3. L'utilisateur choisit et saisit les critères de recherche.</li> <li>4. Le système affiche les résultats correspondants.</li> </ol>
<b>Scénario Alternatif</b>	<p>Aucun résultat de recherche ne correspond aux critères spécifiés.</p> <ul style="list-style-type: none"> <li>- Retour au scénario nominal étape 3.</li> </ul>
<b>Scénario d'exception</b>	L'utilisateur annule la recherche avant de lancer la recherche.

TABLEAU 2.4 – Fiche descriptive du cas d'utilisation «Rechercher»

## 2.7.4 Fiche descriptive du cas d'utilisation «Ajouter annonce»

<b>Ajouter annonce</b>	
<b>Description</b>	Permet au gestionnaire et agence de créer une nouvelle annonce sur le système.
<b>Acteur</b>	Gestionnaire, Agence.
<b>Pré condition</b>	L'acteur doit être authentifié.
<b>Post condition</b>	L'annonce n'est pas ajoutée et l'acteur quitte le système.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à la fonction "gestion des biens" dans le système.</li> <li>2. Le formulaire d'ajout est affiché.</li> <li>3. L'acteur remplit les champs obligatoires et valide.</li> <li>4. L'annonce est ajoutée avec succès dans le système.</li> </ol>
<b>Scénario Alternatif</b>	Ajout non effectué : - L'un des champs obligatoires est vide ou incorrect.
<b>Scénario d'exception</b>	L'acteur annule l'ajout et quitte le système.

TABLEAU 2.5 – Fiche descriptive du cas d'utilisation «Ajouter annonce»

### 2.7.5 Fiche descriptive du cas d'utilisation «Modifier annonce»

<b>Modifier annonce</b>	
<b>Description</b>	Permet au gestionnaire et agence de modifier une annonce sur le système.
<b>Acteur</b>	Gestionnaire, Agence.
<b>Pré condition</b>	L'acteur doit être authentifié.
<b>Post condition</b>	La modification est enregistrée par le système.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur va accéder a la page de gestions des biens .</li> <li>2. Le système affiche la liste des annonces.</li> <li>3. L'acteur sélectionne une annonce à modifier .</li> <li>4. Le système affiche le formulaire de modification.</li> <li>5. L'acteur modifie les informations et enregistre.</li> <li>6. Le système valide l'opération.</li> </ol>
<b>Scénario Alternatif</b>	Informations mal remplies ou les informations à ajouter existes déjà : - Retour au scénario nominal 3.
<b>Scénario d'exception</b>	L'acteur annule la modification.

TABLEAU 2.6 – Fiche descriptive du cas d'utilisation «Modifier annonce»

## 2.7.6 Fiche descriptive du cas d'utilisation «Supprimer annonce»

<b>Supprimer annonce</b>	
<b>Description</b>	Permet au gestionnaire et agence de supprimer une annonce du système.
<b>Acteur</b>	Gestionnaire, Agence.
<b>Pré condition</b>	L'acteur doit être authentifié.
<b>Post condition</b>	La suppression est effectuée par le système.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. L'acteur demande la suppression d'une annonce.</li> <li>2. Le système affiche la liste des annonces.</li> <li>3. L'acteur sélectionne une annonce à supprimer.</li> <li>4. Le système supprime l'annonce.</li> </ol>
<b>Scénario Alternatif</b>	<ol style="list-style-type: none"> <li>1. L'acteur accède à la liste des annonces publiées.</li> <li>2. Il sélectionne l'annonce à supprimer.</li> <li>3. Il clique sur "Supprimer", mais décide ensuite d'annuler l'action.</li> </ol>
<b>Scénario d'exception</b>	L'acteur annule la suppression.

TABLEAU 2.7 – Fiche descriptive du cas d'utilisation «Supprimer annonce»

### 2.7.7 Fiche descriptive du cas d'utilisation «Gérer les visites»

<b>Gérer les Visites</b>	
<b>Description</b>	Permet aux agents immobiliers de gérer les visites des biens immobiliers sur le système.
<b>Acteur</b>	Agent immobilier
<b>Pré condition</b>	Le client doit être authentifié.
<b>Post condition</b>	La gestion des visites est enregistrée par le système.
<b>Scénario Nominal</b>	<ol style="list-style-type: none"> <li>1. Le client demande à planifier une visite pour un bien immobilier.</li> <li>2. Le système affiche les dates disponibles pour la visite.</li> <li>3. Le client sélectionne une date et enregistre la visite.</li> <li>4. Le système envoie une notification à l'agent immobilier de la visite planifiée.</li> <li>5. Le système confirme la planification de la visite.</li> </ol>
<b>Scénario Alternatif</b>	Aucune date disponible pour la visite : - Informer Le client et proposer d'autres options.
<b>Scénario d'exception</b>	Le client annule la planification de la visite.

TABLEAU 2.8 – Fiche descriptive du cas d'utilisation «Gérer les visites»

## 2.8 Diagramme cas d'utilisation globale

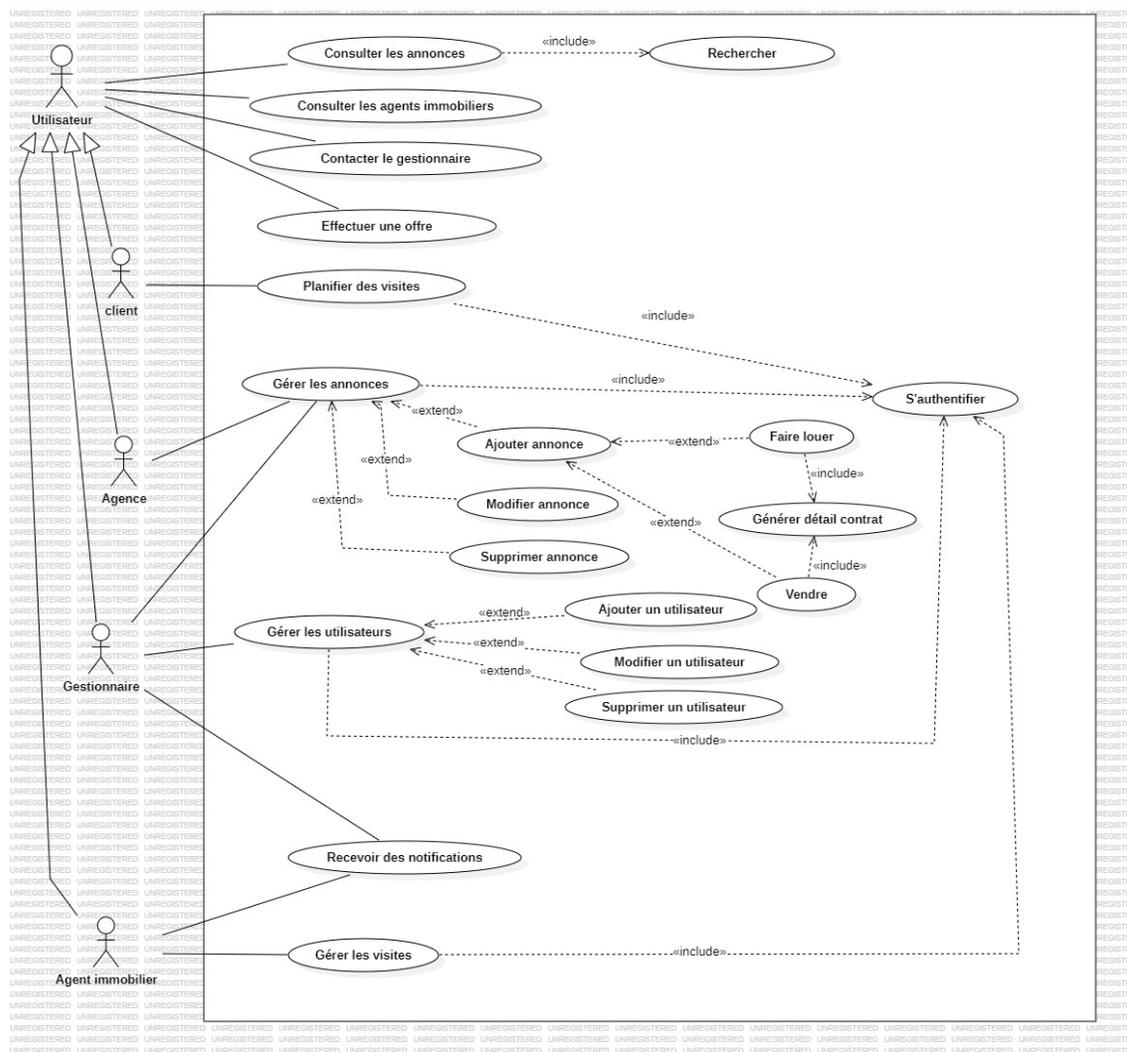


FIGURE 2.11 – Diagramme cas d'utilisation globale

## 2.9 Les diagrammes de séquence système

Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

### 2.9.1 Diagramme de séquence «S'authentifier»

Le gestionnaire, l'agence, l'agent immobilier ou le client envoie ses informations d'identification au système. Ce dernier vérifie les données, autorisant l'accès à l'espace personnel en cas de succès, sinon renvoyant un message d'erreur en cas d'échec.

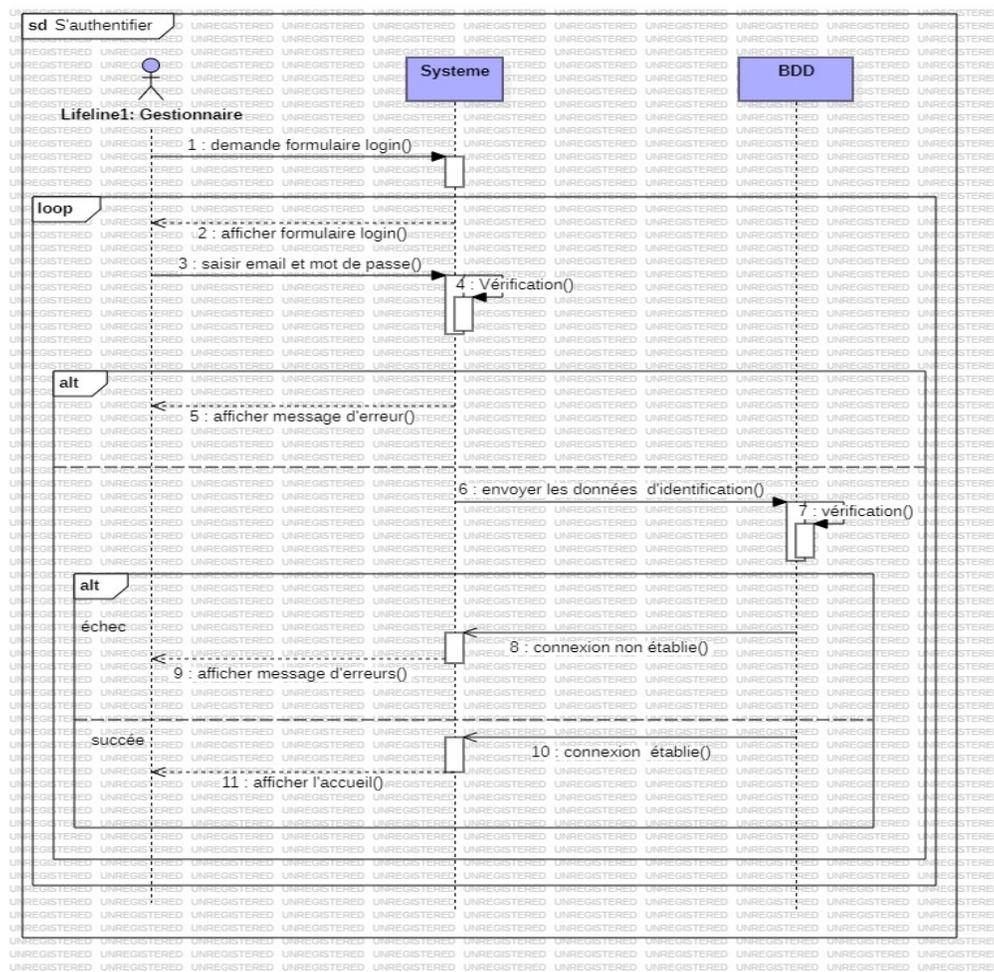


FIGURE 2.12 – Diagramme de séquence «S'authentifier»

## 2.9.2 Diagramme de séquence «ajouter un utilisateur»

Le diagramme de séquence "Ajouter un utilisateur" présente le processus par lequel un gestionnaire inscrit un nouvel utilisateur, qu'il s'agisse d'une agence, d'un agent immobilier ou d'un client, dans le système. Le gestionnaire initie le processus en saisissant les informations de l'utilisateur à ajouter. Le système valide ces données, autorisant ainsi l'inscription de l'utilisateur. Une fois inscrit, l'utilisateur peut accéder aux fonctionnalités qui lui sont attribuées dans le système.

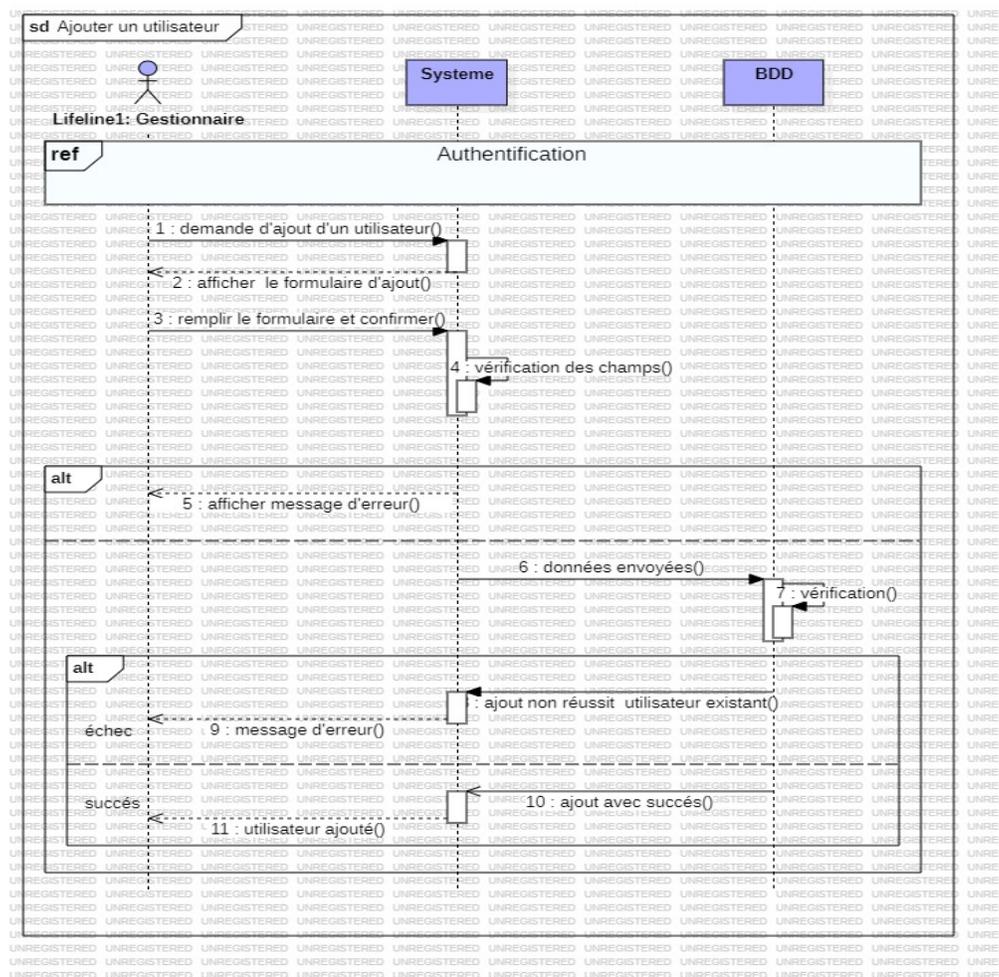


FIGURE 2.13 – Diagramme de séquence «ajouter un utilisateur»

### 2.9.3 Diagramme de séquence «Modifier un utilisateur»

Le diagramme de séquence "Modifier un utilisateur" décrit le processus par lequel un gestionnaire effectue des modifications sur les informations d'un utilisateur existant dans le système. Le gestionnaire initie cette action en sélectionnant l'utilisateur à modifier et en fournissant les nouvelles informations. Le système vérifie et met à jour les données de l'utilisateur, confirmant ainsi la modification.

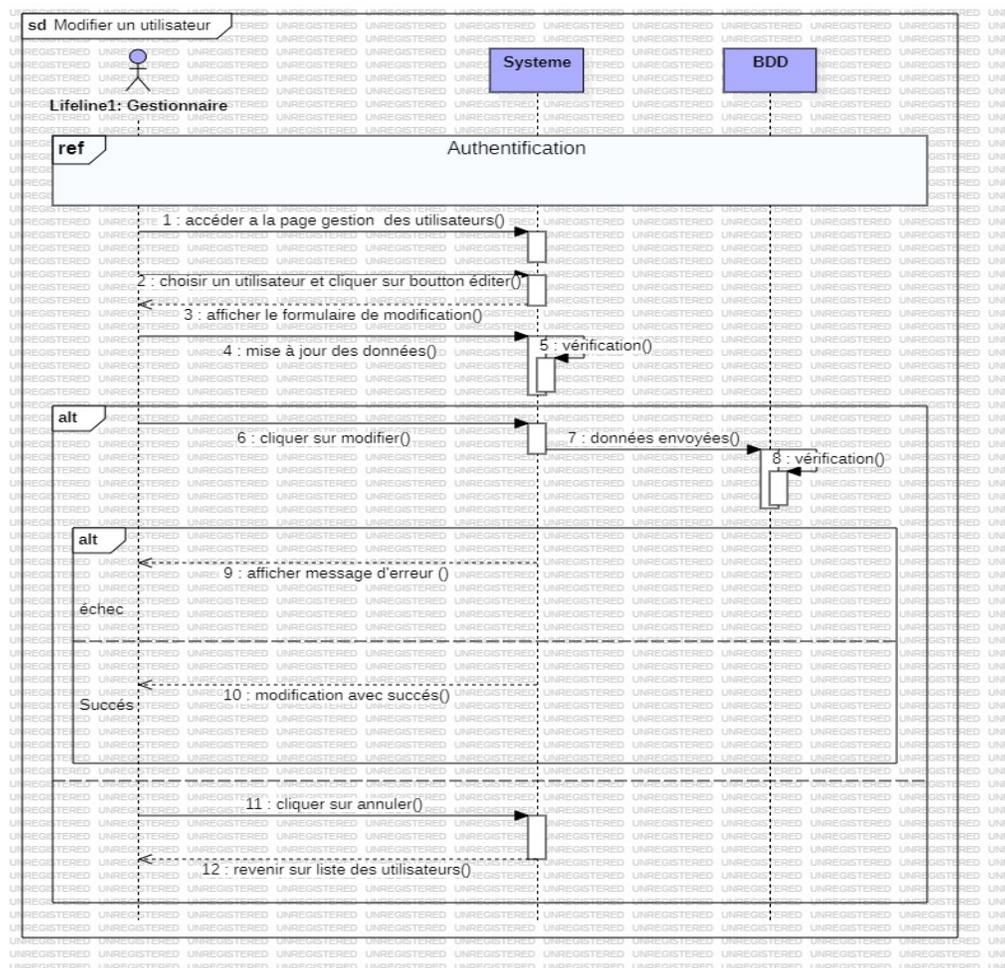


FIGURE 2.14 – Diagramme de séquence «Modifier un utilisateur»

## 2.9.4 Diagramme de séquence «Supprimer un utilisateur»

Le diagramme de séquence "Supprimer d'un utilisateur" montre comment un gestionnaire retire un utilisateur du système. Le gestionnaire choisit l'utilisateur à supprimer. Ensuite, le système confirme la suppression et efface les données de l'utilisateur, l'empêchant ainsi d'accéder au système.

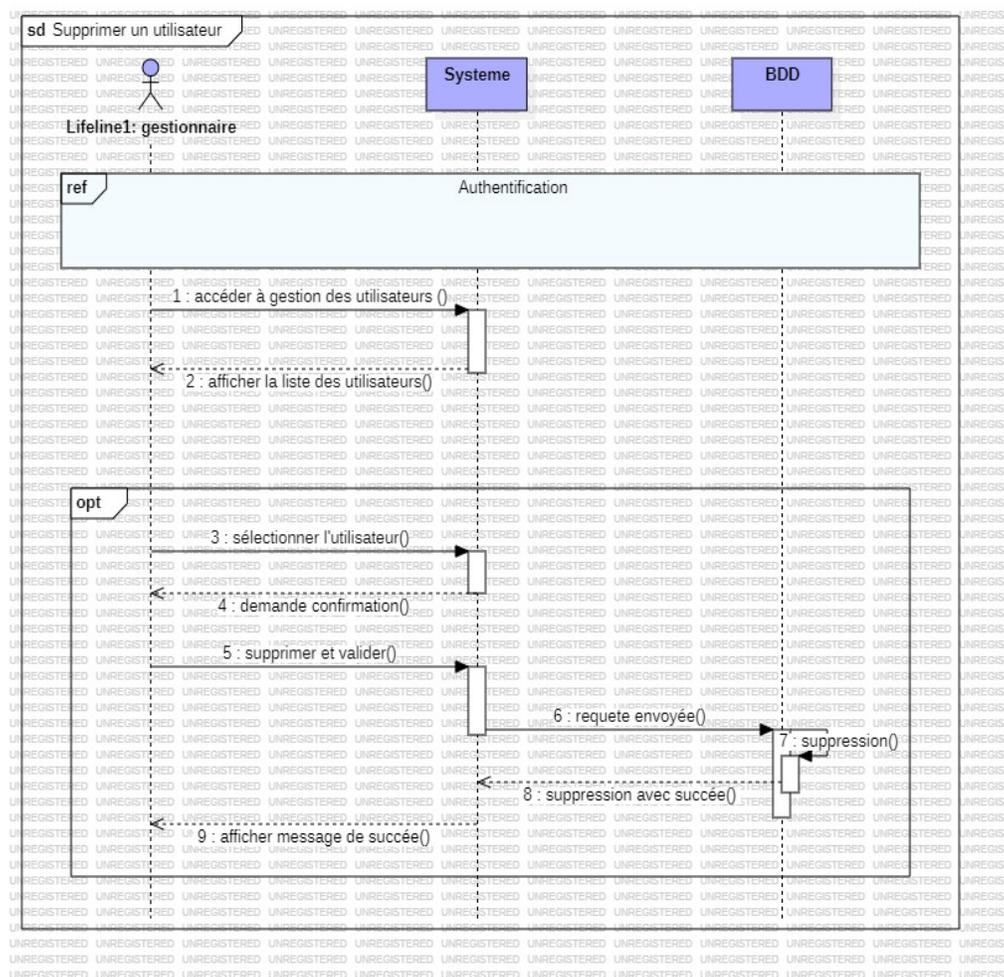


FIGURE 2.15 – Diagramme de séquence «Supprimer un utilisateur»

## 2.9.5 Diagramme de séquence «Rechercher»

Le diagramme de séquence "Rechercher" illustre comment un utilisateur peut rechercher des biens immobiliers dans le système, sans authentification préalable. L'utilisateur démarre la recherche en spécifiant les critères de recherche. Le système traite ces critères et renvoie les résultats correspondants.

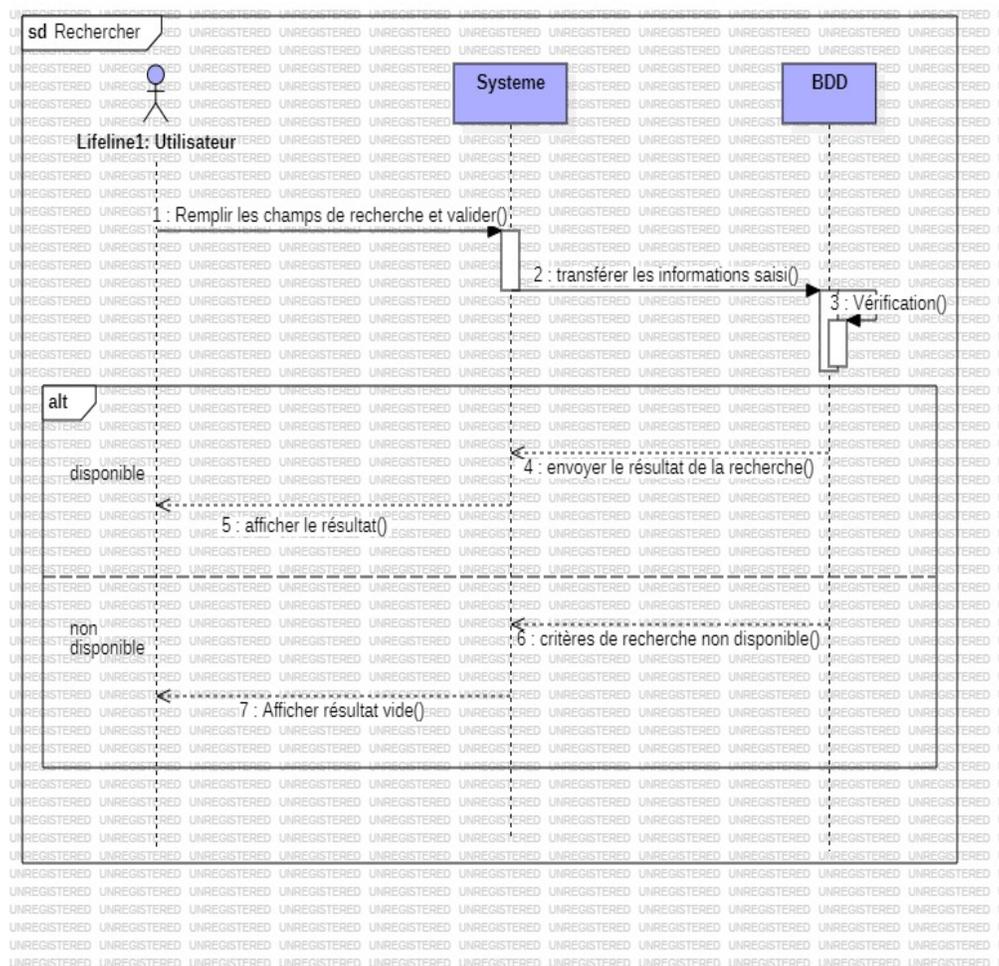


FIGURE 2.16 – Diagramme de séquence «Rechercher»

## 2.9.6 Diagramme de séquence «Ajouter une annonce»

Dans le diagramme de séquence "Ajouter annonce", le processus débute par l'authentification du gestionnaire ou de l'agence dans le système. Après s'être authentifié avec succès, le gestionnaire ou l'agence procède à l'ajout de l'annonce en saisissant les détails nécessaires. Ensuite, le système vérifie les informations fournies et enregistre l'annonce dans la base de données. Une fois l'annonce ajoutée avec succès, le système confirme l'ajout au gestionnaire ou à l'agence.

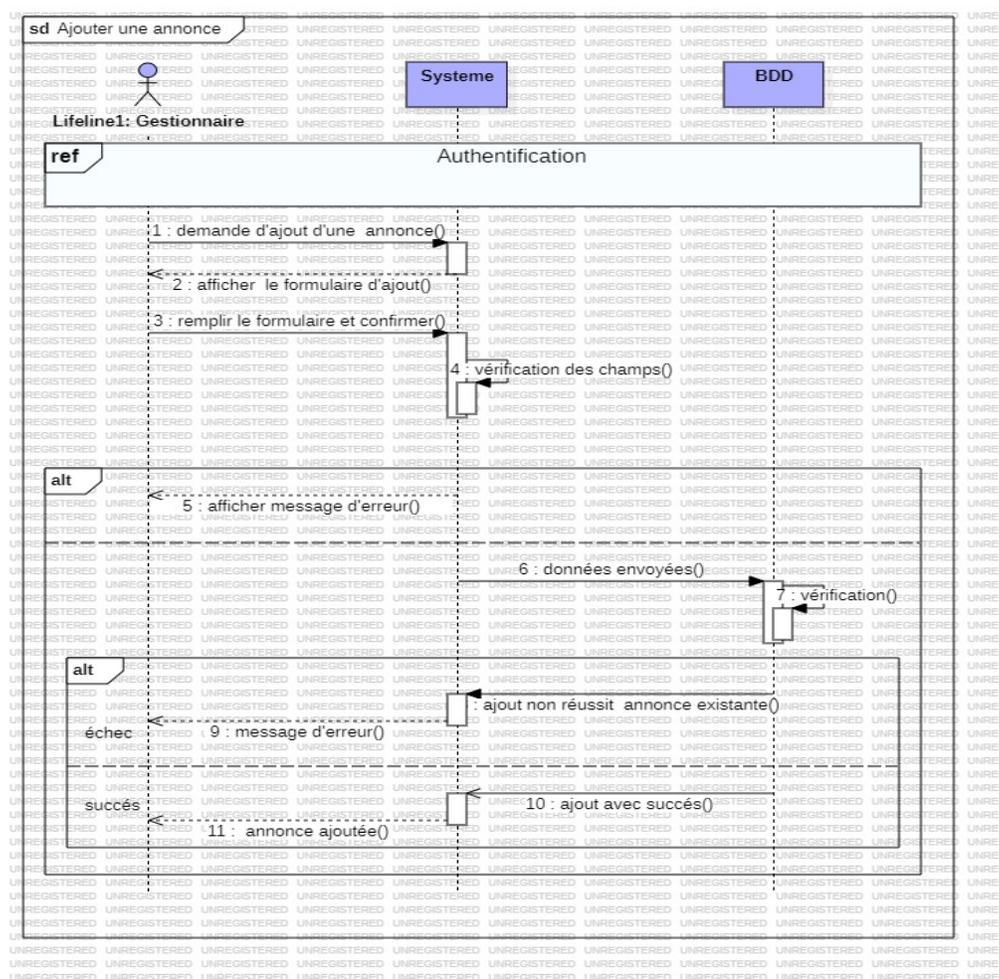


FIGURE 2.17 – Diagramme de séquence «Ajouter une annonce»

## 2.9.7 Diagramme de séquence «Modifier une annonce»

Dans le diagramme de séquence "Modifier une annonce", le processus débute par l'authentification du gestionnaire ou de l'agence dans le système. Après s'être authentifié avec succès, le gestionnaire ou l'agence sélectionne l'annonce à modifier et entre les nouvelles informations. Ensuite, le système vérifie ces modifications et met à jour les détails de l'annonce dans la base de données. Enfin, le système confirme la modification de l'annonce au gestionnaire ou à l'agence.

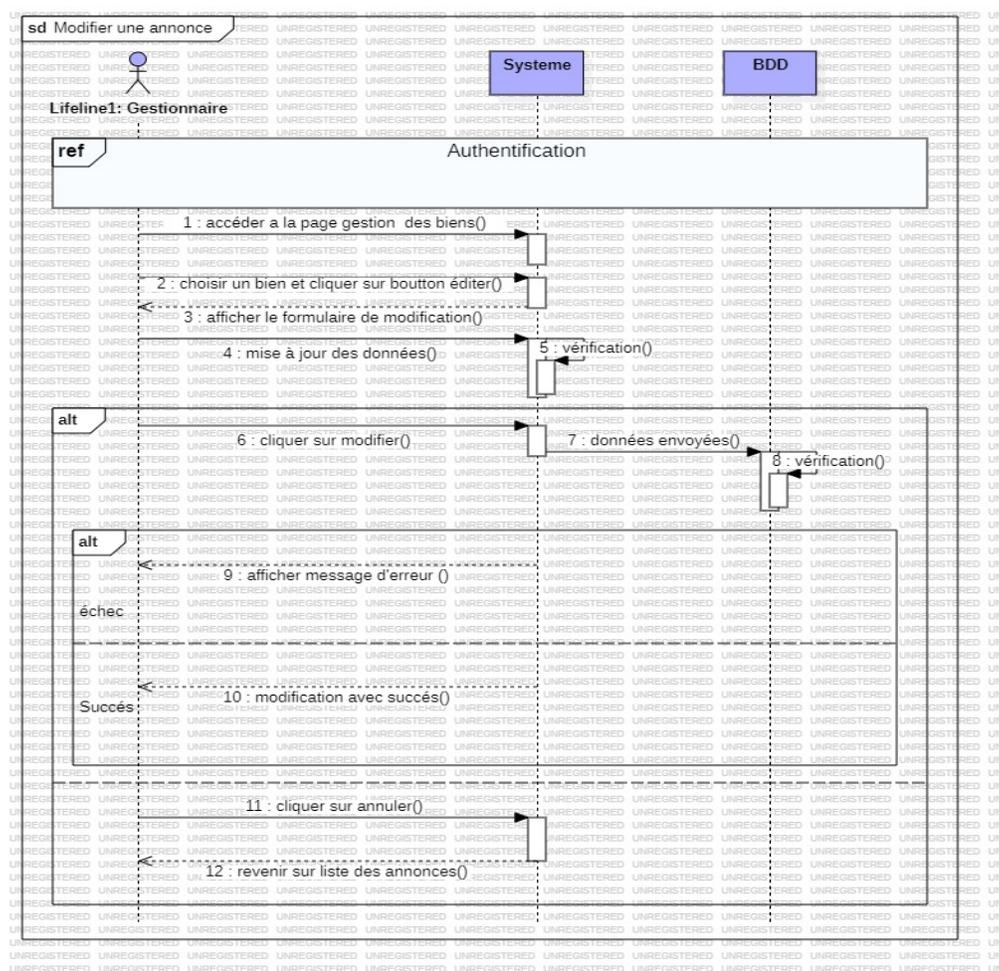


FIGURE 2.18 – Diagramme de séquence «Modifier une annonce»

## 2.9.8 Diagramme de séquence «Supprimer une annonce»

Dans le diagramme de séquence "Supprimer annonce", le processus commence par l'authentification du gestionnaire ou de l'agence dans le système. Après une authentification réussie, le gestionnaire ou l'agence navigue vers l'annonce à supprimer. Le système demande alors une confirmation de la suppression. Une fois la confirmation reçue, le système procède à la suppression de l'annonce de la base de données. Enfin, le système envoie une notification au gestionnaire ou à l'agence pour confirmer que l'annonce a bien été supprimée.

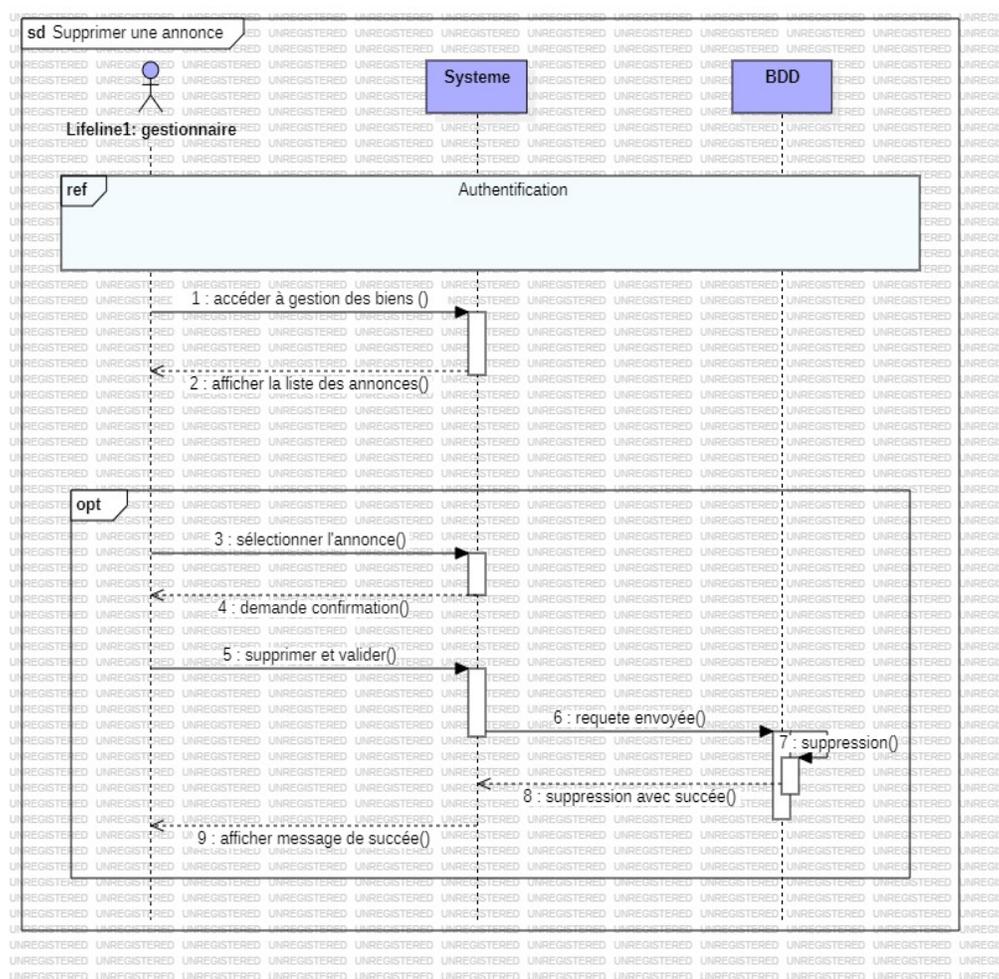


FIGURE 2.19 – Diagramme de séquence «Supprimer une annonce»

## 2.9.9 Diagramme de séquence «Contacter le gestionnaire»

Dans le diagramme de séquence "Contacter le gestionnaire", le processus débute lorsque l'utilisateur accède à l'accueil du système trouve le formulaire de contact. Ensuite, l'utilisateur remplit ce formulaire avec les informations nécessaires et confirme l'envoi du message. Si l'envoi réussit, le système transmet directement le message au gestionnaire à la boîte e-mail. Si l'envoi échoue, le système affiche un message d'erreur à l'utilisateur pour l'informer du problème.

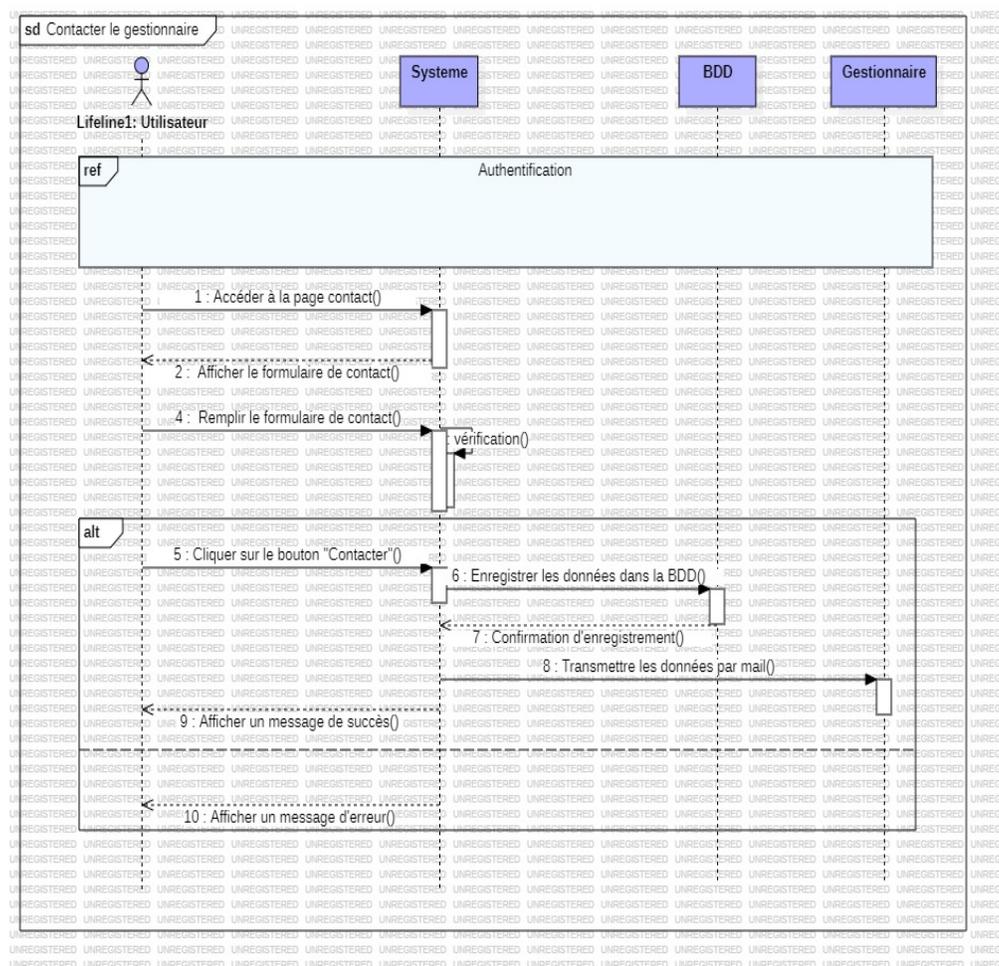


FIGURE 2.20 – Diagramme de séquence «Contacter le gestionnaire»

## 2.9.10 Diagramme de séquence «Effectuer une offre»

Dans le diagramme de séquence "Effectuer une offre", le processus décrit la façon dont un client, accède à la page de détails d'une annonce sur le système, il propose une offre via un formulaire dédié. Lorsque l'envoi de l'offre réussit, le système l'enregistre dans sa base de données et la transmet au gestionnaire à la boîte e-mail. En cas d'erreur lors de la proposition d'offre, le système affiche un message d'erreur à l'utilisateur.

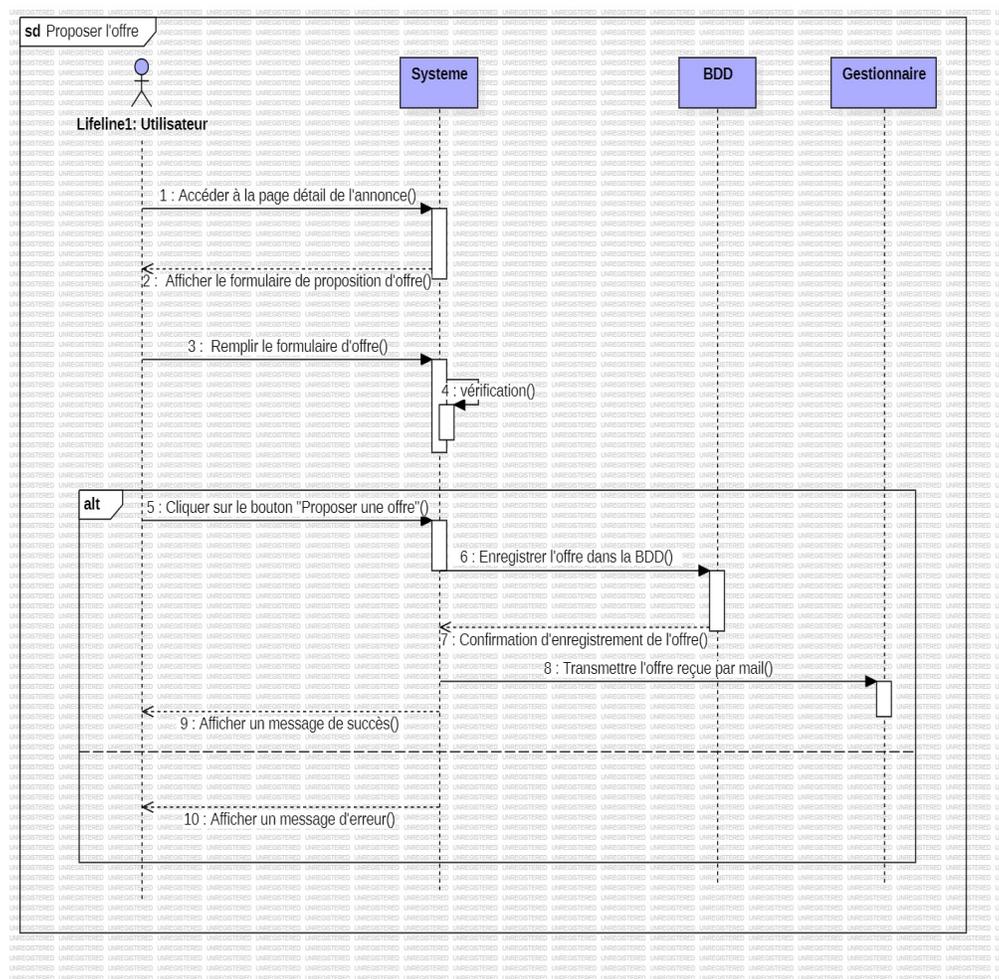


FIGURE 2.21 – Diagramme de séquence «Effectuer une offre»

## 2.10 Représentation des diagrammes d'activités

### 2.10.1 Diagramme d'activité «S'authentifier»

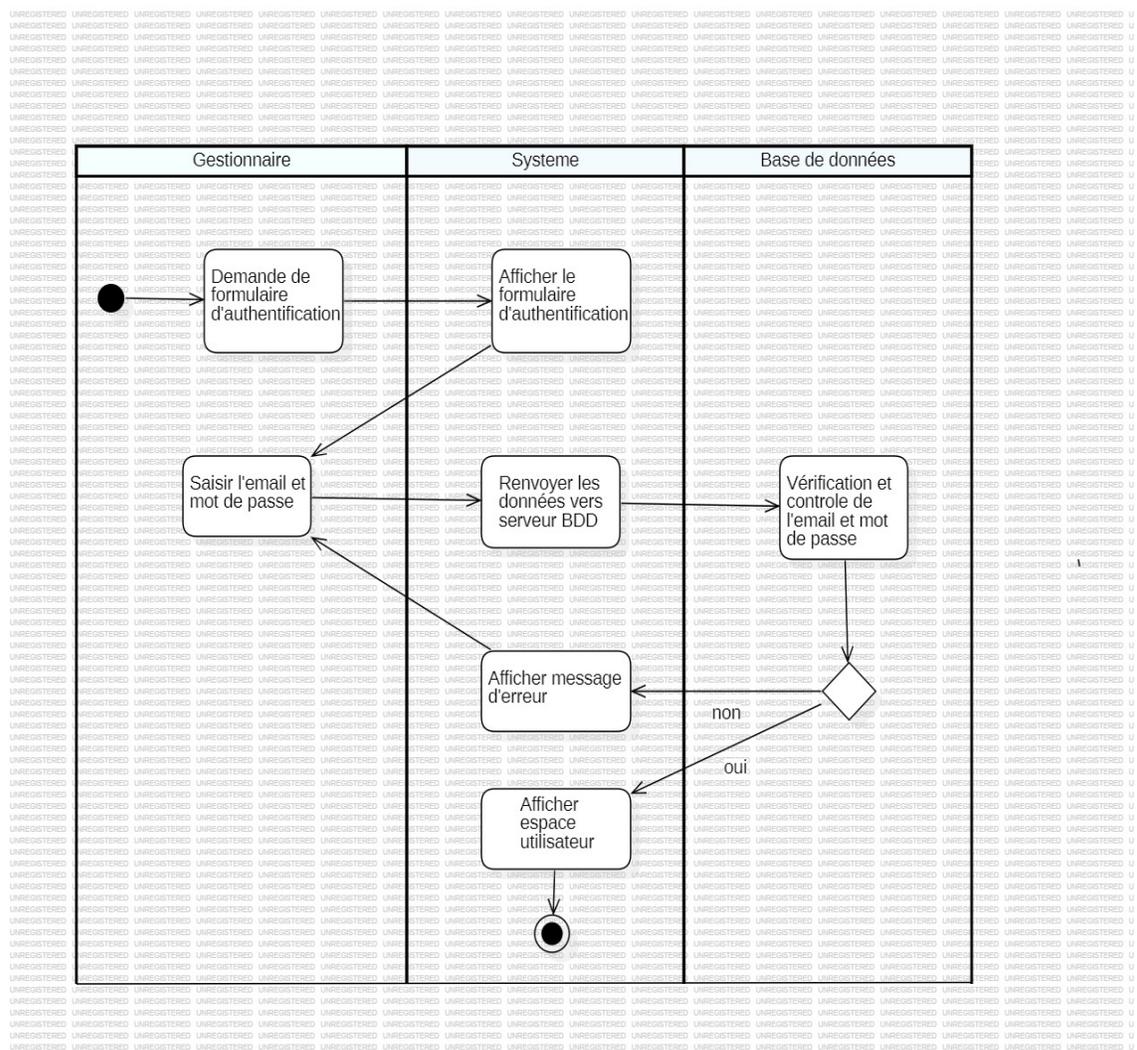


FIGURE 2.22 – Diagramme d'activité «S'authentifier»

## 2.10.2 Diagramme d'activité «Ajouter un utilisateur»

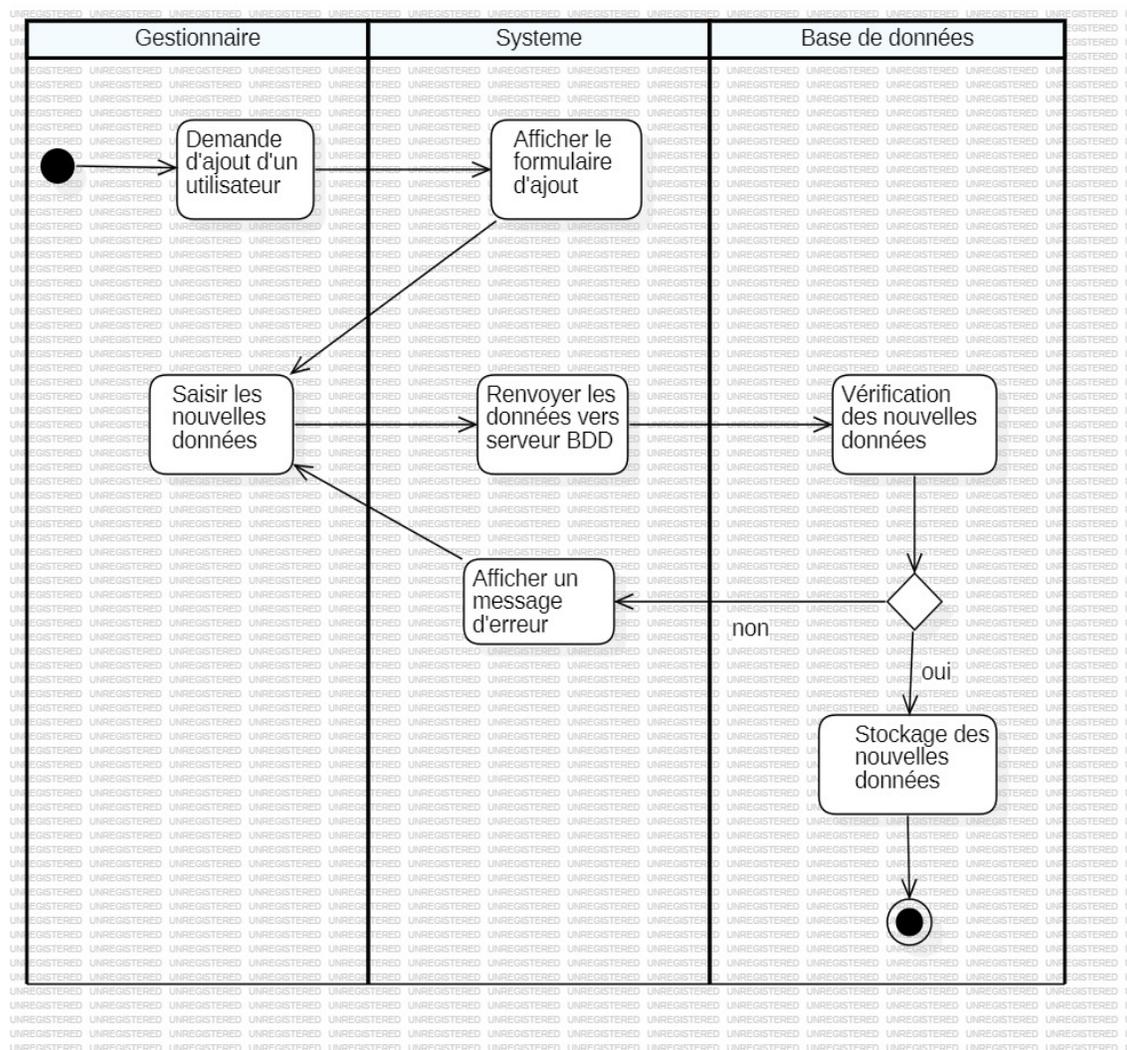


FIGURE 2.23 – Diagramme d'activité «ajouter un utilisateur»

### 2.10.3 Diagramme d'activité «Modifier un utilisateur»

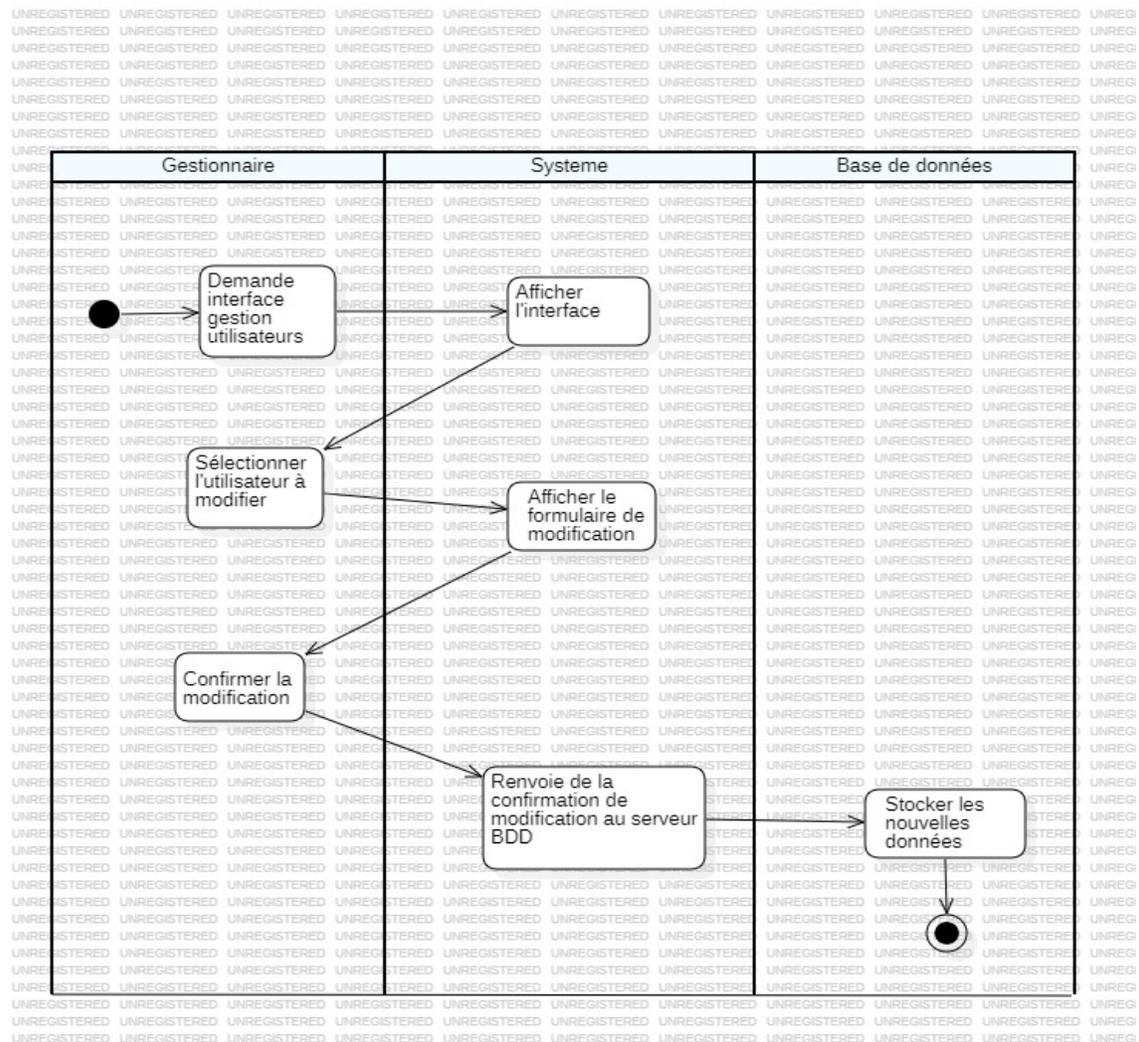


FIGURE 2.24 – Diagramme d'activité «Modifier un utilisateur»

## 2.10.4 Diagramme d'activité «Supprimer un utilisateur»

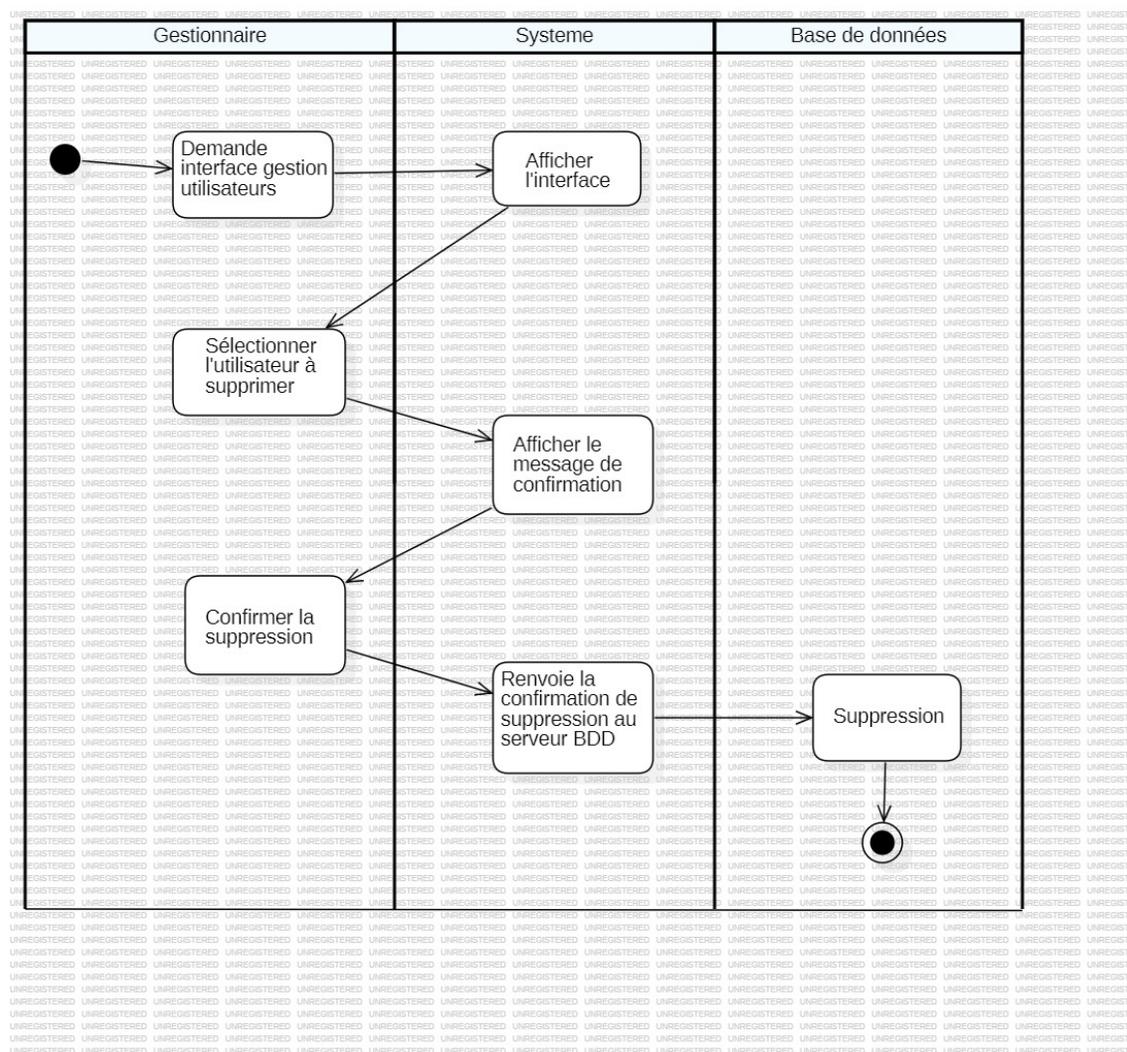


FIGURE 2.25 – Diagramme d'activité «Supprimer un utilisateur»

## 2.11 Dictionnaire de données

Le dictionnaire de données présenté organise et définit de manière structurée les différentes informations relatives aux annonces immobilières. Il fournit une vue d'ensemble des attributs nécessaires pour décrire les biens, les transactions, et les utilisateurs, facilitant ainsi la gestion et l'intégrité des données dans le système immobilier.

<b>Attribut</b>	<b>Signification</b>	<b>Type d'attribut</b>
idAnnonce	Identifiant de l'annonce	Entier
titre	Titre de l'annonce	Chaîne de caractères
image_index	Image d'une annonce	Chaînes de caractères
date	Date de dépôt de l'annonce	Date
idBien	Identifiant de bien immobilier	Entier
type	Type de bien immobilier	Chaîne de caractères
surface	Surface du bien (en mètres carrés)	Réel
prix	Prix du bien	Réel
wilaya	Wilaya du bien	Chaîne de caractères
adresse	Adresse du bien	Chaîne de caractères
description	Description de l'annonce	Texte
transaction	Type de transaction (vente, location,etc..)	Chaîne de caractères
imgs	Images pour les détails du bien	Liste de chaînes de caractères
etat	etat de l'annonce	Booléen
idUtilisateur	Identifiant de l'utilisateur	Entier
nom	Nom de l'utilisateur	Chaîne de caractères
prenom	Prénom de l'utilisateur	Chaîne de caractères
num_telephone	Numéro de téléphone de l'utilisateur	Chaîne de caractères
email	Adresse email de l'utilisateur	Chaîne de caractères
mot_passe	Mot de passe de l'utilisateur	Chaîne de caractères
image	Image d'utilisateur	Chaînes de caractères
adresse	Adresse de l'utilisateur	Chaîne de caractères
wilaya	Wilaya de l'utilisateur	Chaîne de caractères
role_utilisateur	Rôle d'un utilisateur(agent immobilier,etc..)	Chaîne de caractères
idOffre	Identifiant de l'offre	Entier
montant	Montant de l'offre proposée	Réel
idVisite	Identifiant de la visite	Entier
date_visite	Date de visite	Date
heure_visite	Heure de la visite	Heure
nombre_etages	Nombre d'étages(villa,local)	Entier
nombre_chambre	Nombre de chambres(villa,appartement,	Entier
numero_etage	Numéro d'étage(appartement)	Entier
id_contrat	Identifiant du contrat	Entier
duree_location	Durée de la location (en mois)	Entier
date	Date de contrat	Date
role_client	Rôle client(acheteur,vendeur,locataire...)	Chaîne de caractères

TABLEAU 2.9 – Dictionnaire de données pour les attributs

## 2.12 Diagramme de classes

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles

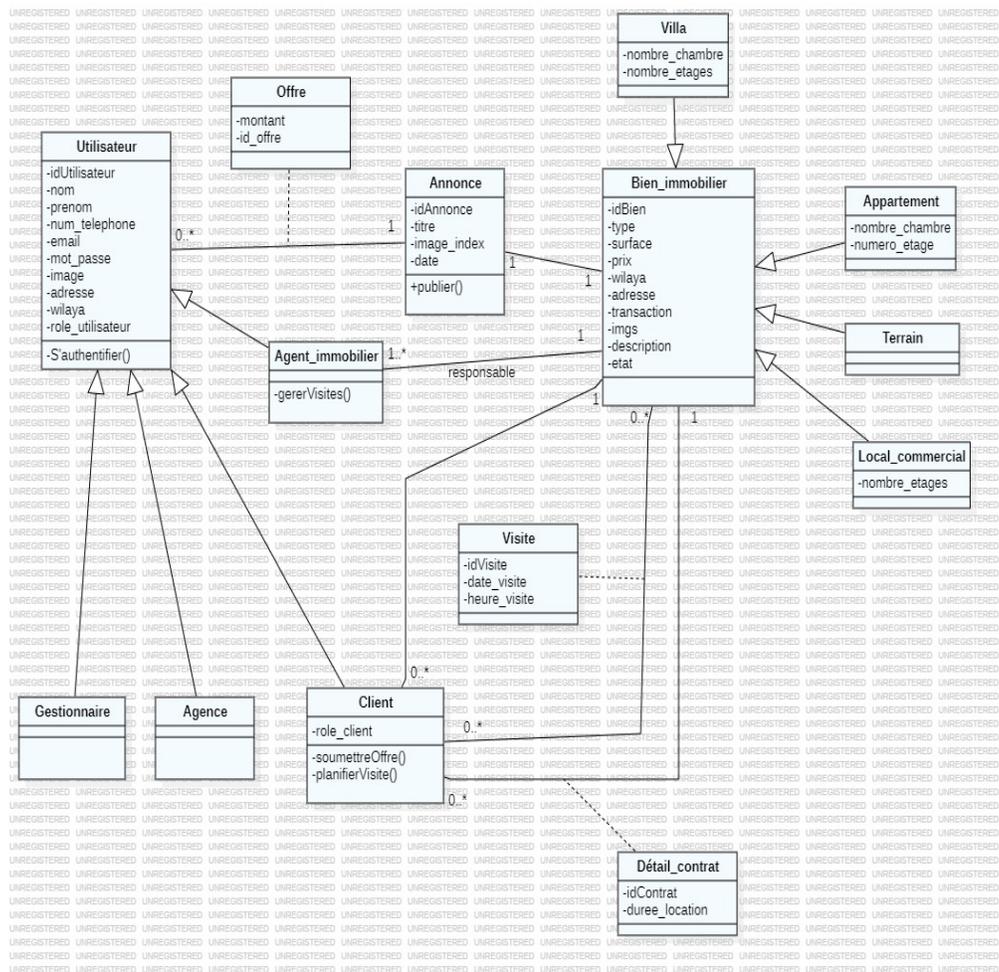


FIGURE 2.26 – Diagramme de classes

## 2.13 Modèle relationnel de données

Le modèle relationnel est le modèle logique de donnée qui correspond à l'organisation des données dans les bases de données relationnelles. Un modèle relationnel est composé de relations, encore appelée table. Ces tables sont décrites par des attributs ou champs. Pour décrire une relation, on indique tout simplement son nom, suivi du nom de ses attributs entre parenthèses. L'identifiant d'une relation est composé d'un ou plusieurs attributs qui forment la clé primaire. Une relation peut faire référence à une autre en utilisant une clé étrangère, qui correspond à la clé primaire de la relation référencée.

### 2.13.1 Passage d'un diagramme de classes UML à un schéma relationnel

Cette section présente les règles pour convertir un diagramme de classes UML en schéma relationnel. Nous présentons ci-dessous quatre règles principales (de R1 à R4) pour effectuer cette transformation. Bien qu'il existe d'autres méthodes, ces règles sont les plus simples et les plus efficaces pour assurer une conversion correcte et utilisable.

Règles :

#### R1 :Transformation des classes

Chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

**Transformation des associations :**Les règles de transformation que nous allons voir dépendent des cardinalités/multiplicités maximales des associations. Nous distinguons trois familles d'associations.

#### R2 :Association un à plusieurs

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association. La clé de la relation père migre dans la relation fils.

#### R3 :Association plusieurs à plusieurs

L'association (classe-association) devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectées à l'association. Chaque attribut devient clé étrangère si classe connectée dont il provient devient une relation en vertu de la

règle R1. Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

#### **R4 :Association un à un**

Il faut ajouter un attribut clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de UML, il faut ajouter un attribut clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de l'entité (classe) connectée à l'association. Si les deux cardinalités (multiplicités) minimales sont à zéro, le choix est donné entre les deux relations dérivées de la règle R1. Si les deux cardinalités minimales sont à un, il est sans doute préférable de fusionner les deux entités (classes) en une seule [26].

Voici un résumé des différentes tables créées pour la réalisation du diagramme de classe :

**Bien Immobilier**(idBien, type, surface, prix, wilaya, adresse, transaction, imgs,description,etat)

**Terrain**(idBien)

**Villa**(idBien, nombre\_chambre, nombre\_etages)

**Appartement**(idBien, nombre\_chambre,numero\_etage)

**Local Commercial**(idBien, nombre\_etage)

**Annonce**(idAnnonce, titre, image\_index,date,#idBien)

**Utilisateur**(idUtilisateur, nom, prenom, num\_telephone, email, mot\_passe,image,adresse, wilaya, role\_utilisateur)

**Client**(role\_client, #idUtilisateur)

**Agent Immobilier**(#idUtilisateur)

**Agence**( #idUtilisateur)

**Gestionnaire**(#idUtilisateur)

**Visite**( idVisite, date\_visite, heure\_visite, #idClient, #idBien)

**Offre**( idOffre, montant, #idUtilisateur, #idAnnonce)

**Contrat**(idContrat, duree\_location, date,#idClient, #idBien)

## 2.14 Conclusion

Après avoir approfondi dans la modélisation et la conception de notre projet, nous avons établi une base solide pour la suite de notre développement. En explorant différentes méthodologies de modélisation et en détaillant l'utilisation d'UML, nous avons jeté les fondations nécessaires pour comprendre et conceptualiser notre application de manière méthodique.

À travers l'identification des acteurs, la description des cas d'utilisation et la création de divers diagrammes, nous avons clarifié les fonctionnalités et les interactions clés de notre système. Ces étapes ont permis une compréhension approfondie de nos besoins et exigences, fournissant ainsi une orientation précieuse pour la suite du processus de développement.

Dans le prochain chapitre, nous plongerons dans les outils de développement et la conception des interfaces graphiques pour notre site web et notre application mobile. Cette étape nous rapprochera encore plus de la concrétisation de notre vision, en transformant nos concepts en une expérience utilisateur fonctionnelle et intuitive.

# Chapitre 3

## Réalisation et présentation de l'application

### 3.1 Introduction

Dans ce troisième chapitre, nous aborderons la phase de réalisation de notre projet, en présentant les outils de développement utilisés ainsi que la conception des interfaces graphiques. Cette étape cruciale traduit nos concepts en une application fonctionnelle, tant pour le site web que pour l'application mobile.

Nous commencerons par détailler les choix technologiques et les outils de développement qui nous ont permis de concrétiser notre vision. Ensuite, nous nous pencherons sur la création des interfaces utilisateur, en soulignant l'importance d'une expérience utilisateur fluide et intuitive. Chaque composant de notre système sera présenté avec précision, illustrant la transition des idées théoriques vers une solution pratique et utilisable.

Ce chapitre marquera une avancée significative dans notre projet, mettant en lumière le travail accompli et les efforts déployés pour développer une application robuste et conviviale.

### 3.2 Outils et langages de développements

#### 3.2.1 Les frameworks utilisés

### 3.2.1.1 Laravel



Laravel est un framework PHP créé par Taylor Otwell en 2011, qui permet de simplifier le développement d'applications web tout en gardant le code bien organisé. Depuis sa création, Laravel est devenu l'un des frameworks PHP les plus populaires et les plus utilisés, avec une communauté de développeurs en constante croissance.

Laravel est basé sur le modèle MVC (Modèle-Vue-Contrôleur), qui permet de séparer le code en trois couches distinctes pour une meilleure organisation et une maintenance plus facile. Le modèle s'occupe de la logique de l'application et de l'interaction avec la base de données, la vue est responsable de l'affichage de l'interface utilisateur, et le contrôleur agit comme une passerelle entre le modèle et la vue, en traitant les requêtes entrantes et en envoyant les données à la vue.

Laravel fournit également de nombreux composants prêts à l'emploi, tels que l'authentification, la validation, la gestion des fichiers, la mise en cache, la gestion des sessions, la gestion des tâches planifiées et bien d'autres [27].

**3.2.1.1.1 Les avantages de Laravel :** Voici quelques-uns des avantages spécifiques de Laravel :

- Laravel est facile à apprendre.
- Laravel simplifie le processus de développement.
- Laravel dispose d'outils pour les développeurs de tous les niveaux.
- Laravel évolue facilement.
- Laravel dispose d'un écosystème et d'une communauté massive.
- Laravel est largement utilisé[28].

**3.2.1.1.2 Découpage MVC :** Maintenant, voyons un peu plus en détails comment s'organise ce pattern de conception en découpant ses trois composantes : le modèle, la vue et le contrôleur.

**Modèle :**Le modèle, ou model, représente les données qui vont être utilisées dans l'application web.

**Vue :**La vue, ou view, c'est l'interface graphique de l'application. C'est via cet élément que vont se faire les interactions entre l'utilisateur et le code métier. Elle ne contient

presque aucune logique (contrairement à l'architecture concurrente de MVC dont on parlera plus loin), son but est de construire, à partir de ce que renvoie le serveur, une interface et de l'afficher à l'utilisateur.

**Contrôleur** :Le contrôleur, ou controller, est l'élément qui contient la logique métier. C'est ici que sont la plupart des algorithmes, calculs, etc. C'est aussi l'intermédiaire principal entre la vue et le modèle. Par exemple, la vue soumet un formulaire au contrôleur, qui gère sa validation via du code métier, et demande au modèle de faire des modifications dans la base de données [29].

La figure suivante illustre le modèle de conception MVC tel qu'implémenté dans Laravel :

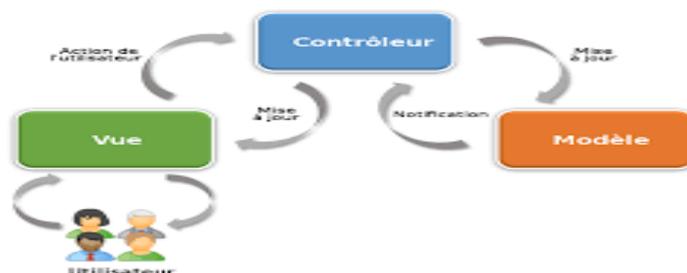


FIGURE 3.1 – Architecture MVC

### 3.2.1.2 Bootstrap

**B** Bootstrap est l'un des frameworks les plus populaires pour le développement de sites Web. L'outil fournit des modèles pour CSS et HTML avec lesquels vous pouvez mettre en place une conception Web moderne de manière relativement simple. La mise en page, les polices, les boutons et les divers éléments de navigation peuvent être créés et positionnés très simplement [30].

## 3.2.2 Langages de programmation utilisés

### 3.2.2.1 HTML



HTML (pour HyperText Markup Language, qu'on peut traduire en « langage de balisage hypertexte ») est le langage utilisé pour structurer une page web et son contenu. On peut par exemple organiser le contenu en un ensemble de paragraphes, une liste d'éléments, utiliser des images ou des tableaux de données [31].

### 3.2.2.2 CSS



Le CSS pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages HTML. On le traduit en français par feuilles de style en cascade. Grâce au CSS, vous pouvez en effet appliquer des règles de mise en forme (titrage, alignement, polices, couleurs, bordures, etc.) à plusieurs documents simultanément [32].

### 3.2.2.3 JavaScript



JavaScript est un langage de programmation utilisé par les développeurs pour concevoir des sites web interactifs. Les fonctions JavaScript peuvent permettre d'améliorer l'expérience utilisateur d'un site web, de la mise à jour des flux de médias sociaux à l'affichage d'animations et de cartes interactives [33].

### 3.2.2.4 PHP



PHP est un langage de script côté serveur. Il est utilisé pour développer des sites web statiques ou dynamiques ou des applications web. PHP signifie Hypertext Pre-processor, qui signifiait auparavant Personal Home Pages. Les scripts PHP ne peuvent être interprétés que sur un serveur sur lequel PHP est installé. Les ordinateurs clients qui accèdent aux scripts PHP ne nécessitent qu'un navigateur web [34].

## 3.2.3 Logiciels utilisés pour le développement

### 3.2.3.1 Visual Studio Code



Visual Studio Code (VSCode) est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C, Python, Java, etc. Il offre de nombreuses fonctionnalités comme la coloration syntaxique, l'auto-complétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l'intégration avec Git, et beaucoup d'autres. Il est également extensible à l'aide d'une grande variété d'extensions développées par la communauté, permettant aux développeurs de personnaliser l'éditeur selon leurs besoins [35].

### 3.2.3.2 Android Studio



Android Studio est l'environnement de développement intégré (IDE) officiel, conçu et distribué par Google pour le développement d'applications Android. Un IDE contient des outils qui permettent aux développeurs de concevoir, compiler, exécuter et tester des logiciels, dans ce cas, des applications pour la plate-forme Android. La plate-forme Android Studio est basée sur IntelliJ IDEA. Elle inclut le plug-in Android préinstallé ainsi que des modifications spécifiques à la plate-forme Android [36].

### 3.2.3.3 API REST

REST API (Representational State Transfer Application Program Interface) est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes HTTP pour récupérer et publier des données entre un périphérique client et un serveur.

En utilisant le protocole HTTP, les REST API permettent aux logiciels d'un appareil de communiquer avec les logiciels d'un autre appareil (ou du même appareil) même s'ils

utilisent des systèmes d'exploitation et des architectures différents. Le client peut demander des ressources avec un langage que le serveur comprend, et le serveur renvoie la ressource avec un langage que le client accepte. Le serveur renvoie la ressource au format JSON (JavaScript Object Notation), XML (Extensible Markup Language) ou texte, mais de nombreuses API prennent en charge d'autres langages [37].



FIGURE 3.2 – «API REST»

### 3.2.3.4 XAMPP



XAMPP est un ensemble de logiciels permettant de facilement, créer une interface web interagissant avec une base de données SQL. Le nom est un acronyme venant des initiales de tous les composants de cette suite. Ce dernier réunit donc le serveur Web Apache, la base de données relationnelle et système d'exploitation MySQL ou MariaDB ainsi que les langages scripts Perl et PHP. L'initiale X représente tous les systèmes d'exploitation possibles, à savoir Linux, Windows et Mac OS X.

### 3.2.3.5 MYSQL



MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

### 3.2.3.6 StarUML



StarUML est un outil de génie logiciel dédié à la modélisation UML et édité par la société coréenne MKLabs. Il est multiplateforme et fonctionne sous Windows, Linux et MacOS. La dernière version gère l'ensemble des diagrammes définis par UML 2, ainsi que plusieurs diagrammes SysML, les organigrammes, les diagrammes de flux de données, et les diagrammes entité-association [39].

### 3.2.3.7 Canva



Canva est un outil de design graphique disponible en ligne, mais également téléchargeable sur Mac, Windows, Android et iOS. Il s'agit d'une plateforme intuitive qui fonctionne selon le mode glisser/déposer. Elle permet de créer facilement une panoplie de publications Web et imprimées : des logos aux affiches, en passant par les bannières, dépliants, images pour les réseaux sociaux, etc [40].

## 3.2.4 Matériel utilisé

Nous avons utilisé un PC portable de marque HP (Hewlett-Packard) pour la réalisation de notre projet. Ses caractéristiques principales sont :

- Processeur : Intel Core i5 2.40GHz
- RAM : 8.00 Go
- Type du système : 64bits

## 3.3 Présentation des interfaces de notre application web

### 3.3.1 Page d'accueil

La page d'accueil est le premier point de contact avec notre application. Conçue pour une expérience fluide, elle offre un accès rapide aux annonces immobilières, aux informations sur nos agents, à la possibilité de contacter le gestionnaire, et à l'espace de connexion pour les utilisateurs enregistrés. Tout est centralisé pour faciliter la navigation dès l'arrivée.



FIGURE 3.3 – Page d'accueil

### 3.3.2 Annonces récentes

Découvrez les opportunités immobilières récentes, incluant maisons, terrains, appartements et locaux commerciaux, disponibles pour la vente, l'achat ou l'échange. Accédez à ces offres exceptionnelles directement depuis l'accueil de notre application.

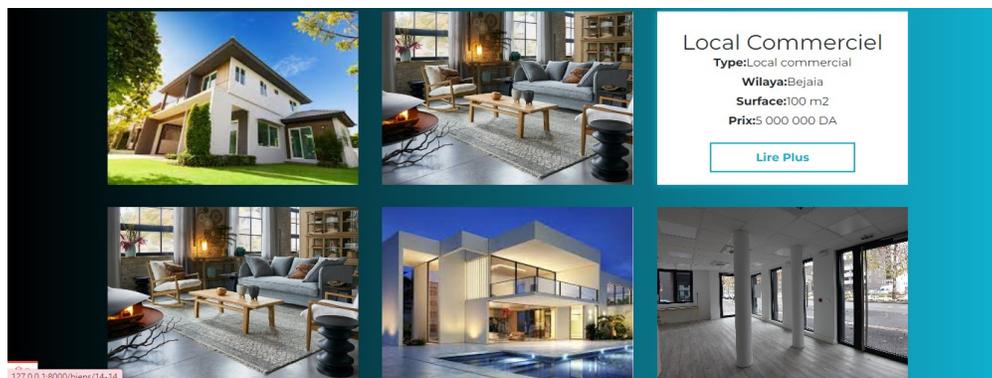


FIGURE 3.4 – Annonces récentes

### 3.3.3 Agents immobiliers

Dans cette section, découvrez les agents dédiés à la gestion des visites et transactions. Trouvez l'agent idéal pour vous accompagner dans toutes vos démarches immobilières.



FIGURE 3.5 – Agents immobiliers

### 3.3.4 Footer de l'accueil

Retrouvez des informations sur Dream House, nos branches, des liens utiles, et nos moyens de contact. Suivez-nous sur les réseaux sociaux pour les dernières offres et actualités.

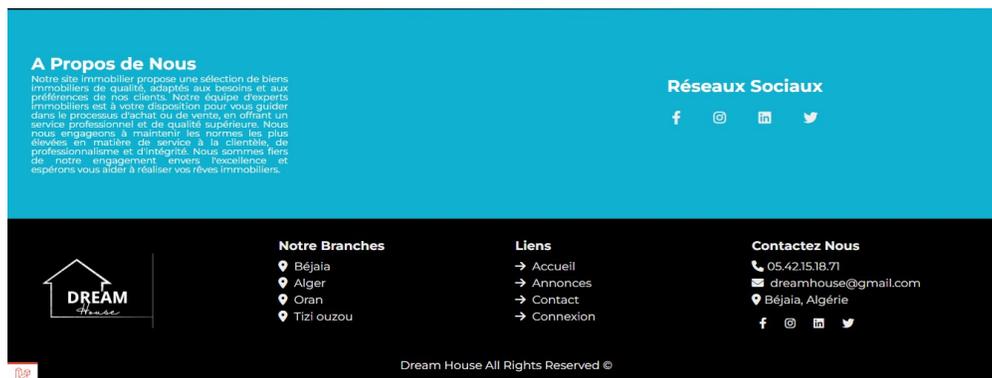
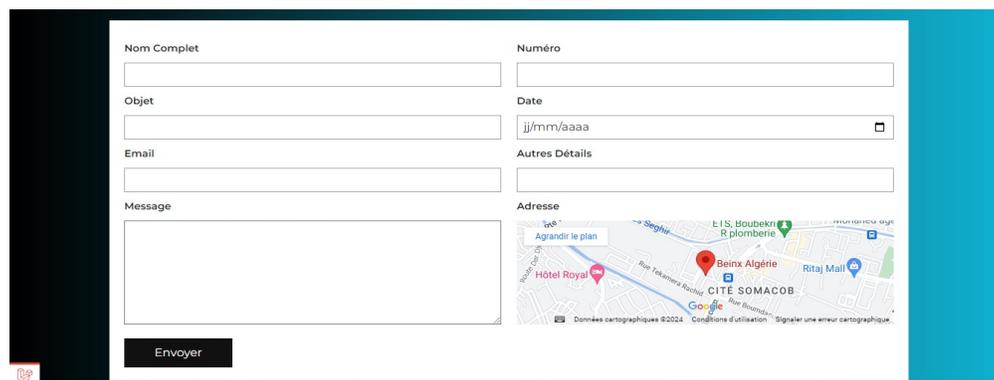


FIGURE 3.6 – Footer de l'accueil

### 3.3.5 Page contact

La page de contact permet aux utilisateurs de contacter directement l'agence en remplissant un formulaire. Les messages sont envoyés au gestionnaire qui les reçoit instantanément dans sa boîte e-mail, garantissant une communication rapide et efficace.



The screenshot shows a contact form with the following fields:

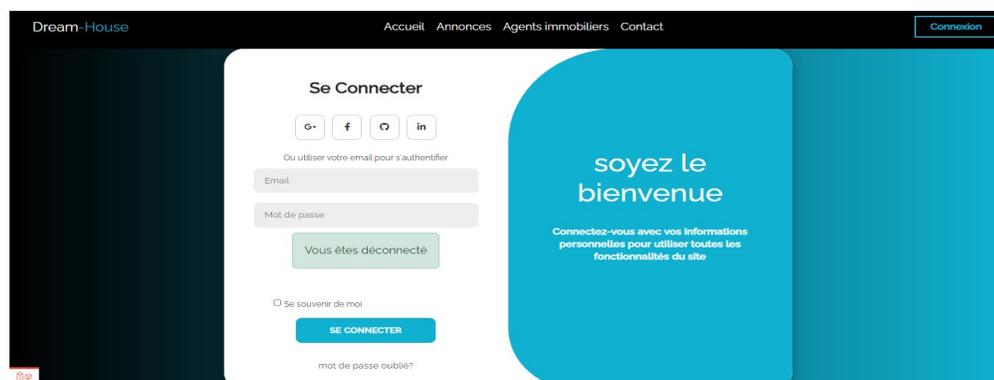
- Nom Complet
- Objet
- Email
- Message
- Numéro
- Date (with a calendar icon)
- Autres Détails
- Adresse (with a Google Maps preview showing locations like Hôtel Royal, Beinx Algérie, and CITE SOMACOB)

An "Envoyer" button is located at the bottom left of the form.

FIGURE 3.7 – Page contact

### 3.3.6 Page d'authentification

C'est l'interface principale de l'application, elle permet aux utilisateurs d'accéder à l'application en saisissant un email et un mot de passe pour s'authentifier. Depuis cette page, il pourra accéder aux autres interfaces.



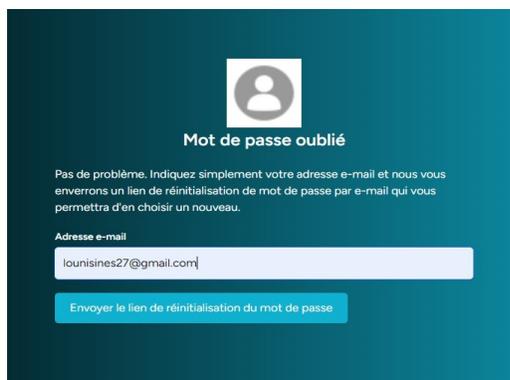
The screenshot shows the authentication page with the following elements:

- Header: Dream-House, Accueil, Annonces, Agents immobiliers, Contact, Connexion
- Section: Se Connecter
- Social login buttons: Google, Facebook, Twitter, LinkedIn
- Text: Ou utiliser votre email pour s'authentifier
- Form fields: Email, Mot de passe
- Buttons: Vous êtes déconnecté, SE CONNECTER
- Text: Se souvenir de moi
- Text: mot de passe oublié?
- Large blue graphic: soyez le bienvenue, Connectez-vous avec vos informations personnelles pour utiliser toutes les fonctionnalités du site

FIGURE 3.8 – Page d'authentification

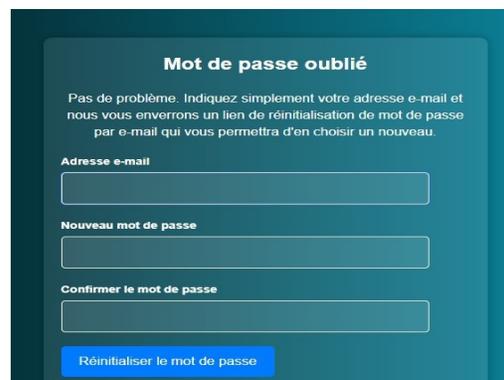
### 3.3.7 Page gestion des mots de passe

La page mot de passe oublié permet aux utilisateurs de récupérer leur mot de passe en saisissant leur adresse e-mail. Ils reçoivent ensuite un lien de réinitialisation dans leur boîte de réception pour choisir un nouveau mot de passe sécurisé, assurant ainsi un accès rapide et sécurisé à leur compte.



The screenshot shows a teal background with a white user icon at the top. Below it, the title 'Mot de passe oublié' is centered. A paragraph of text explains the process: 'Pas de problème. Indiquez simplement votre adresse e-mail et nous vous enverrons un lien de réinitialisation de mot de passe par e-mail qui vous permettra d'en choisir un nouveau.' Below this is a text input field for the email address, containing 'lounisines27@gmail.com'. At the bottom is a blue button labeled 'Envoyer le lien de réinitialisation du mot de passe'.

FIGURE 3.9 – Page mot de passe oublié



The screenshot shows a teal background with a white box containing the title 'Mot de passe oublié'. Below the title is a paragraph: 'Pas de problème. Indiquez simplement votre adresse e-mail et nous vous enverrons un lien de réinitialisation de mot de passe par e-mail qui vous permettra d'en choisir un nouveau.' There are three text input fields: 'Adresse e-mail', 'Nouveau mot de passe', and 'Confirmer le mot de passe'. At the bottom is a blue button labeled 'Réinitialiser le mot de passe'.

FIGURE 3.10 – Page de réinitialisation

### 3.3.8 Page vérification de l'adresse e-mail

La page de vérification d'e-mail est essentielle après l'inscription. Un e-mail est envoyé à l'utilisateur pour confirmer son adresse. En cliquant sur le lien, l'utilisateur active son compte, renforçant ainsi la sécurité.



The screenshot shows a teal background with a white box containing the title 'Vérification de l'adresse e-mail'. Below the title is a paragraph: 'Merci de vous être inscrit ! Avant de commencer, pourriez-vous vérifier votre adresse e-mail en cliquant sur le lien que nous venons de vous envoyer par e-mail ? Si vous n'avez pas reçu l'e-mail, nous vous en enverrons volontiers un autre.' At the bottom are two buttons: a blue button labeled 'Renvoyer l'e-mail de vérification' and a smaller, lighter blue button labeled 'Déconnexion'.

FIGURE 3.11 – Page vérification de l'adresse e-mail

### 3.3.9 Espace d'accueil pour gestionnaire

L'espace gestionnaire réservé à l'administrateur de Dream House permet une gestion facile des utilisateurs, agents immobiliers, agences et clients, incluant l'ajout de nouveaux biens immobiliers et la génération sécurisée de contrats pour les transactions.



FIGURE 3.12 – Espace d'accueil pour gestionnaire

### 3.3.10 Page ajouter un utilisateur

L'espace gestion utilisateurs de Dream House permet au gestionnaire d'ajouter et de gérer efficacement les agences, agents immobiliers et clients.

FIGURE 3.13 – Page ajouter un utilisateur

### 3.3.11 Page ajouter un bien

L'espace gestion biens de Dream House permet au gestionnaire ou à l'agence d'ajouter rapidement une variété de biens immobiliers pour la vente, la location ou l'échange, incluant appartements, locaux commerciaux, terrains et villas. Cette plateforme simplifie la gestion des biens immobiliers pour une commercialisation efficace.

FIGURE 3.14 – Page ajouter un bien

### 3.3.12 Page gestion des utilisateurs

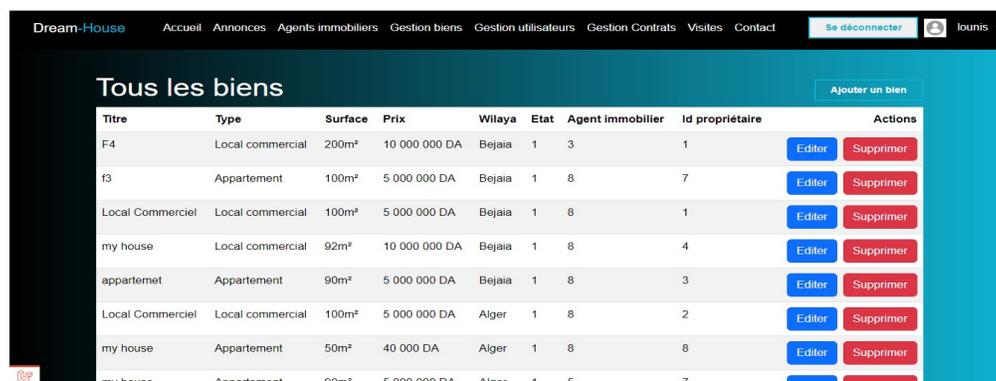
La page de gestion des utilisateurs est l'endroit où le gestionnaire peut visualiser l'ensemble des utilisateurs sous forme de tableau. Cette interface lui permet de supprimer ou de modifier les informations d'un utilisateur en toute simplicité.

id	Nom	Prénom	Email	Numéro de téléphone	Adresse	Rôle	Actions
10	Meddour	Hadia	hadiameddour28@gmail.com	541060975	Bejaia	agence	<a href="#">Modifier</a> <a href="#">Supprimer</a>
8	agent	agent	immobiliera208@gmail.com	541060975	Bejaia	agent	<a href="#">Modifier</a> <a href="#">Supprimer</a>
7	lounis	lydia	lounislydia06@gmail.com	541060975	bejaia	client	<a href="#">Modifier</a> <a href="#">Supprimer</a>
4	Lous ines	Lous ines	lounisines@gmail.com	541060975	Bejaia	agent	<a href="#">Modifier</a> <a href="#">Supprimer</a>
3	bouyahmed	lounis	bouyahmedlounis@gmail.com	541060975	Bejaia	agent	<a href="#">Modifier</a> <a href="#">Supprimer</a>
2	meddour	hadia	meddourhadia@gmail.com	541060977	bejaia	agence	<a href="#">Modifier</a> <a href="#">Supprimer</a>
1	lounis	ines	lounisines27@gmail.com	542151871	adekar	agent	<a href="#">Modifier</a> <a href="#">Supprimer</a>
11	lounis	ines	ines.lounis@se.univ-bejaia.dz	542151871	adekar	admin	<a href="#">Modifier</a> <a href="#">Supprimer</a>

FIGURE 3.15 – Page gestion des utilisateurs

### 3.3.13 Page gestion des biens

La page de gestion des biens présente toutes les annonces sous forme de tableau, offrant ainsi une vue complète pour le gestionnaire. À partir de cette interface conviviale, le gestionnaire peut facilement modifier ou supprimer des annonces selon les besoins de l'agence immobilière.



Titre	Type	Surface	Prix	Wilaya	Etat	Agent immobilier	Id propriétaire	Actions
F4	Local commercial	200m²	10 000 000 DA	Bejaia	1	3	1	<a href="#">Editer</a> <a href="#">Supprimer</a>
f3	Appartement	100m²	5 000 000 DA	Bejaia	1	8	7	<a href="#">Editer</a> <a href="#">Supprimer</a>
Local Commercial	Local commercial	100m²	5 000 000 DA	Bejaia	1	8	1	<a href="#">Editer</a> <a href="#">Supprimer</a>
my house	Local commercial	92m²	10 000 000 DA	Bejaia	1	8	4	<a href="#">Editer</a> <a href="#">Supprimer</a>
appartemet	Appartement	90m²	5 000 000 DA	Bejaia	1	8	3	<a href="#">Editer</a> <a href="#">Supprimer</a>
Local Commercial	Local commercial	100m²	5 000 000 DA	Alger	1	8	2	<a href="#">Editer</a> <a href="#">Supprimer</a>
my house	Appartement	50m²	40 000 DA	Alger	1	8	8	<a href="#">Editer</a> <a href="#">Supprimer</a>
my house	Appartement	90m²	5 000 000 DA	Alger	1	5	7	<a href="#">Editer</a> <a href="#">Supprimer</a>

FIGURE 3.16 – Page gestion des biens

### 3.3.14 Page des annonces

La page des annonces affiche toutes les annonces récentes et anciennes, avec une barre de recherche permettant de filtrer par type de bien, type de transaction, surface et prix maximum. Si aucune propriété ne correspond aux critères de recherche, un message "Aucun bien ne correspond à votre recherche" est affiché.

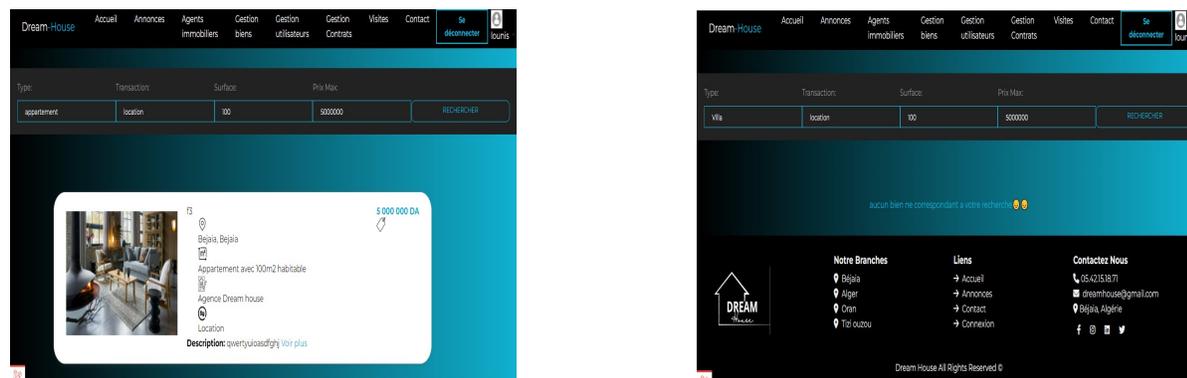


FIGURE 3.17 – Page des annonces

### 3.3.15 Page détails et offres annonce

Cette interface permet de consulter les détails complets d'une annonce immobilière ainsi que de soumettre facilement une offre via un formulaire dédié.

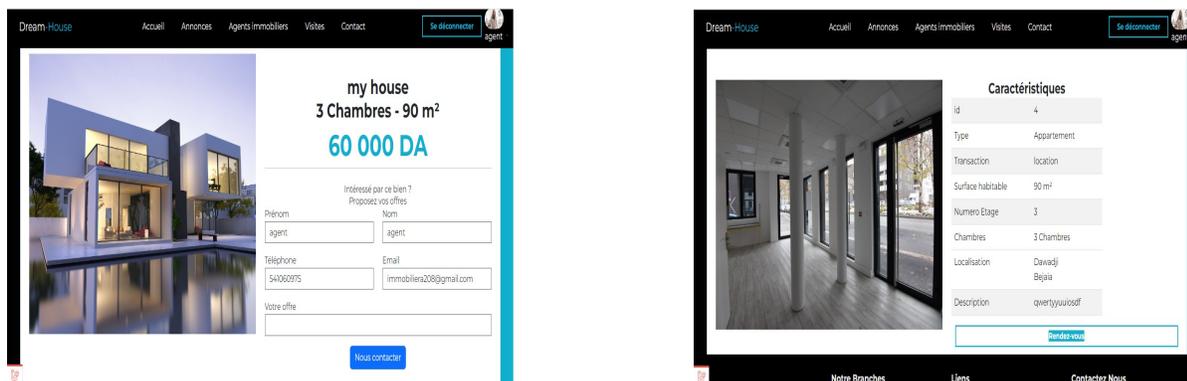


FIGURE 3.18 – Page détails et offres annonce

### 3.3.16 Notification d'une offre

Lorsqu'une offre est soumise, le gestionnaire reçoit immédiatement une notification par e-mail, l'informant de la nouvelle proposition d'achat ou de location sur une propriété.

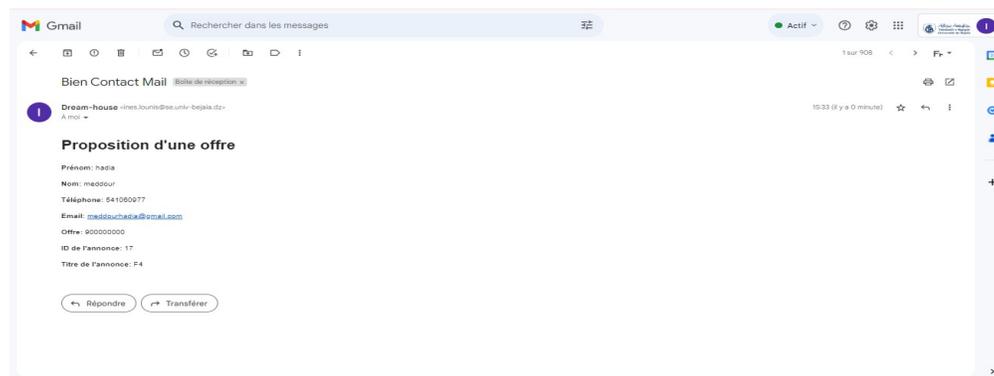


FIGURE 3.19 – Notification d'une offre

### 3.3.17 Page prendre un rendez-vous

Dans ce formulaire, les utilisateurs peuvent sélectionner la date et l'heure de leur rendez-vous parmi les disponibilités proposées. Les détails du client, de l'agent immobilier et du bien immobilier sont automatiquement préremplis à partir de la base de données, simplifiant ainsi la planification des visites.

FIGURE 3.20 – Page prendre un rendez-vous

### 3.3.18 Notification d'un rendez-vous

Lorsqu'un rendez-vous est pris, le gestionnaire et l'agent immobilier reçoivent une notification par e-mail contenant tous les détails de la visite, incluant la date, l'heure, et les informations du client.

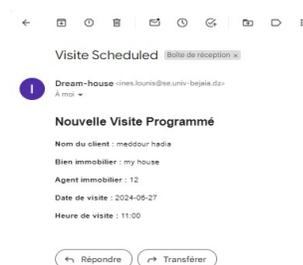


FIGURE 3.21 – Notification gestionnaire

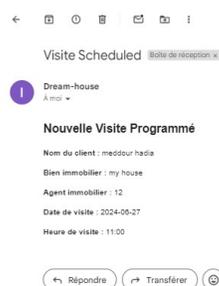


FIGURE 3.22 – Notification agent

### 3.3.19 Page gestion des visites

Cette interface présente un tableau récapitulatif où toutes les visites effectuées sont enregistrées. Le tableau affiche les détails de chaque visite, tels que la date, l'heure, l'agent immobilier et le bien immobilier concerné.



The screenshot shows a web interface titled 'Mes Visites'. It contains a table with the following data:

Bien	Date	Heure	Agent immobilier
Local Commercial	2024-06-07	08:00:00	8
Local Commercial	2024-12-26	08:00:00	8
Local Commercial	2024-09-20	08:00:00	8
Local Commercial	2024-09-27	09:00:00	8
Local Commercial	2024-06-02	08:00:00	8
Local Commercial	2024-06-02	09:00:00	8
Local Commercial	2024-06-02	10:00:00	8
Local Commercial	2024-09-27	10:00:00	8
Local Commercial	2024-09-27	11:00:00	8

FIGURE 3.23 – Gestion des visites

### 3.3.20 Visites des agents immobiliers

Cette interface présente un tableau listant toutes les visites programmées pour les agents immobiliers. Chaque entrée du tableau affiche le nom du client, la date et l'heure de la visite, ainsi que les détails du bien immobilier concerné. Cela permet aux agents de gérer et d'organiser leurs visites de manière efficace.



The screenshot shows a web interface titled 'Visites où je suis agent immobilier'. It contains a table with the following data:

Client	Bien	Date	Heure
Iounis Ines	Local Commercial	2024-06-07	08:00:00
Iounis Ines	Local Commercial	2024-12-26	08:00:00
Iounis Ines	Local Commercial	2024-09-20	08:00:00
Iounis Ines	Local Commercial	2024-09-27	09:00:00
Iounis Ines	Local Commercial	2024-06-02	08:00:00
Iounis Ines	Local Commercial	2024-06-02	09:00:00
Iounis Ines	Local Commercial	2024-06-02	10:00:00
Iounis Ines	Local Commercial	2024-09-27	10:00:00
Iounis Ines	Local Commercial	2024-09-27	11:00:00

FIGURE 3.24 – Gestion des visites

### 3.3.21 Page détail d'un contrat

Cette page contient quatre formulaires distincts. Le premier formulaire recueille les informations du propriétaire, le deuxième détaille les informations du bien immobilier, le troisième collecte les informations du client, et le quatrième concerne les détails du contrat, incluant la durée de la location si c'est une transaction de location et le prix final. Cela permet de créer des contrats complets et précis pour chaque transaction immobilière.

FIGURE 3.25 – Page détail d'un contrat

### 3.3.22 Impression page détail d'un contrat

En cliquant sur le bouton "Imprimer", vous pouvez générer une version imprimable du contrat. Cette fonctionnalité permet de facilement obtenir une copie physique du document pour signature ou archivage.

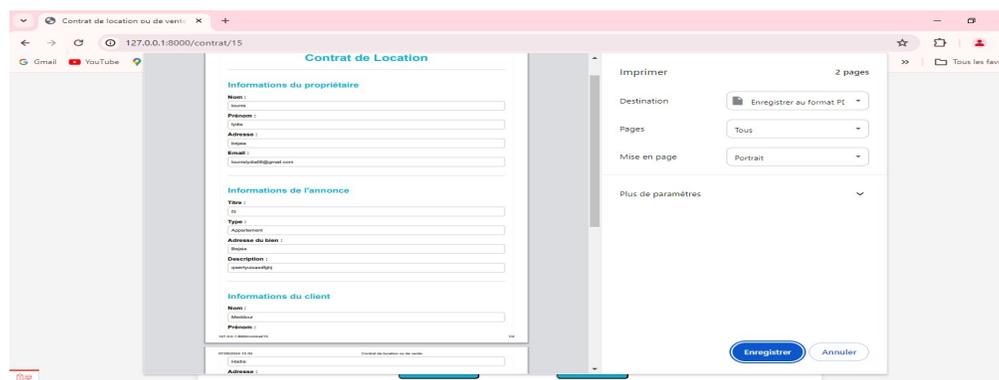


FIGURE 3.26 – Impression page détail d'un contrat

## 3.4 Présentation des interfaces de notre application mobile

### 3.4.1 Page d'accueil

L'écran d'accueil de l'application mobile Dream-House vous invite à découvrir l'élégance de l'immobilier avec une agence qui met l'accent sur la sécurité et la qualité de ses services.



FIGURE 3.27 – Page d'accueil

### 3.4.2 Annonces récentes et liste des agents immobiliers

Cette section présente les annonces récentes et la liste des agents immobiliers disponibles pour les visites. Ces informations sont accessibles directement sur la page d'accueil de l'application mobile.

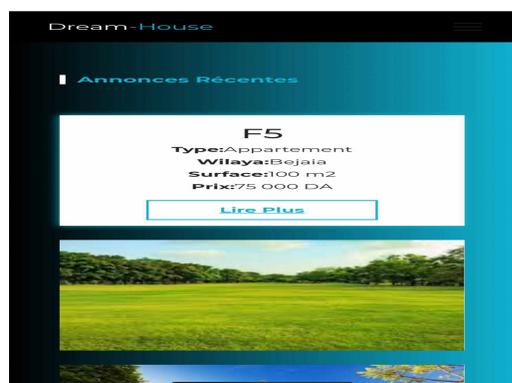


FIGURE 3.28 – Annonces récentes



FIGURE 3.29 – Liste des agents immobiliers

### 3.4.3 Page des annonces

La page des Annonces de notre application mobile vous permet de consulter l'ensemble des biens immobiliers disponibles qu'ils soient récents ou anciens. Vous y trouverez des appartements, des maisons, des terrains et des locaux commerciaux. Chaque annonce inclut une description détaillée, des photos, ainsi que des informations essentielles comme le prix, la surface, et la wilaya.

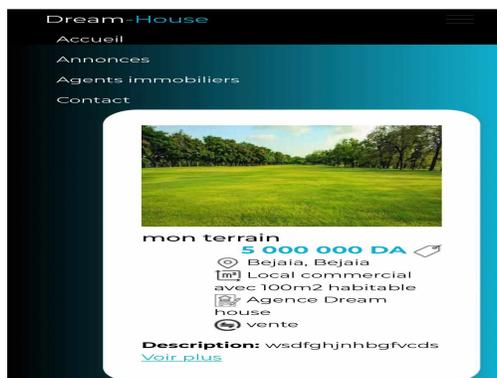


FIGURE 3.30 – Page des annonces

### 3.4.4 Page contact

La page contact facilite la communication avec le gestionnaire via un formulaire dédié, assurant ainsi une réponse prompte par courrier électronique.



FIGURE 3.31 – Page contact

### 3.4.5 Page offre

La page Offre de notre application mobile est conçue pour permettre aux clients de faire des offres sur les biens immobiliers qui les intéressent. En accédant à cette page, les utilisateurs peuvent soumettre une proposition en remplissant un formulaire dédié. Ce formulaire comprend des champs pour indiquer le montant de l'offre et toute autre information. Une fois l'offre soumise, le gestionnaire reçoit une notification par e-mail contenant tous les détails de la proposition.



Dream-House

**f4**

**12 Chambres - 200 m<sup>2</sup>**

**50 000 000 DA**

Intéressé par ce bien ?  
Proposez vos offres

Prénom  Nom

Téléphone  Email

Votre offre

[Nous contacter](#)

FIGURE 3.32 – Page offre

## 3.5 Conclusion

Ce chapitre de réalisation et de présentation de l'application marque une étape importante dans notre projet. En expliquant nos choix technologiques tels que l'utilisation de Laravel et Bootstrap, ainsi que la conception des interfaces graphiques, nous avons créé une application solide et conviviale. L'adoption de l'architecture MVC a permis une organisation claire du code et une expérience utilisateur fluide. Chaque élément, de la page d'accueil à la gestion des utilisateurs et des biens immobiliers, a été conçu pour simplifier les interactions et répondre aux besoins des utilisateurs.

# Conclusion Générale

La réalisation de ce projet de développement d'une application web et mobile pour la gestion d'une agence immobilière a été une expérience à la fois enrichissante et formatrice. En quatre chapitres distincts, nous avons exploré les différentes étapes essentielles à la concrétisation de notre solution intégrée.

Le premier chapitre a été consacré à une étude approfondie de l'organisme d'accueil et de l'existant, ce qui nous a permis de comprendre les besoins spécifiques des agences immobilières et de poser les bases de notre projet. Nous avons ensuite établi un cahier des charges détaillant les spécifications fonctionnelles et non fonctionnelles de notre application.

Le deuxième chapitre a porté sur la modélisation et la conception de l'application. En utilisant des méthodes de modélisation avancées, nous avons conçu une architecture robuste et évolutive.

Enfin, le troisième chapitre a décrit la réalisation concrète de notre application, mettant en avant les technologies utilisées, les fonctionnalités implémentées, et les défis surmontés.

Ce projet nous a permis de contribuer à la transformation digitale du secteur immobilier en proposant une solution innovante et efficace pour la gestion des agences immobilières. Bien que notre application soit désormais opérationnelle, nous sommes conscients des possibilités d'amélioration et envisageons l'intégration de nouvelles fonctionnalités pour une optimisation continue.

Nous espérons que ce travail apportera une contribution significative à l'évolution du secteur immobilier et facilitera le quotidien des professionnels et des clients. Nous remercions tous ceux qui ont contribué à la réalisation de ce projet et sommes confiants que notre application répondra aux attentes des utilisateurs.

# Annexes

## 3.6 Sélection du Langage de Modélisation

Nous avons opté pour UML en raison de sa capacité à offrir une modélisation de haut niveau de notre système, indépendamment des langages et des environnements utilisés, tout en permettant une documentation complète de notre projet.

## 3.7 Historique de l'UML

UML est issue du regroupement de 3 méthodes : la méthode OMT de James Rumbaugh : pour l'analyse et la conception de systèmes d'information, la méthode de Grady Booch pour l'implémentation des projets, de la méthode OOSE de Ivar Jacobson : pour l'expression des besoins. En 1994 Rumbaugh rejoint Booch chez Rational., puis en 1995 Jacobson rejoint Rational et le 14 Novembre 1997 : UML est adopté par l'OMG. La figure montre les différentes étapes sur les quelles est passé UML [14].

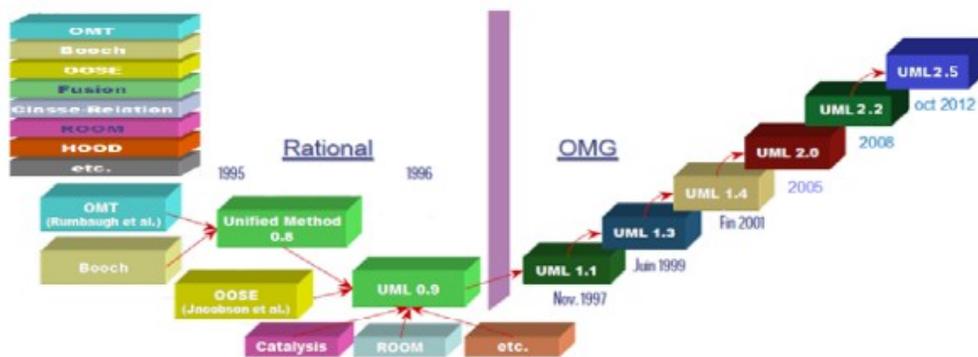


FIGURE 3.33 – Histoire UML

## 3.8 Définition de UML

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet [15].

### 3.8.1 Liste des types de diagrammes UML

#### Diagrammes de structure

- **Diagramme de classes** : Représente un système statique orienté objet. Il définit les projets par des classes, des attributs et des fonctions. Il s'agit donc d'un élément fondamental de toute solution orientée objet. Il montre les classes au sein d'un système et les opérations de chacune d'entre elles.
- **Diagramme de paquets** : Les diagrammes de paquets regroupent les classes en paquets. Ils montrent les différentes dépendances et relations entre les paquets au sein d'un système.
- **Diagramme d'objets** : Les diagrammes d'objets sont similaires aux diagrammes de classes en ce sens qu'ils montrent les relations entre les objets d'un logiciel. La différence est que les diagrammes d'objets utilisent des exemples du monde réel. Les diagrammes d'objets sont également appelés diagrammes d'instance, car ils montrent à quoi ressemble un système à un moment précis.
- **Diagramme de composants** : Permet aux développeurs d'avoir une compréhension globale des objets physiques d'un système. Ce type de diagramme UML montre la relation structurelle entre chaque composant et sous-composant physique d'un système logiciel complexe. Il aide les parties prenantes à comprendre comment les composants sont organisés et reliés entre eux.

- **Diagramme de structure composite** : Permettent de visualiser la structure interne d'une classe. Ces diagrammes décomposent le réseau de classes, d'interfaces et de composants. Ce type de diagramme UML montre également comment ces éléments interagissent entre eux et pourquoi ils sont essentiels à la structure globale d'un logiciel.
- **Diagramme de déploiement** : Un diagramme de déploiement montre la relation entre les composants logiciels et matériels d'un système. Il décrit la disposition physique des nœuds dans un système distribué. Ces diagrammes sont particulièrement utiles lorsque le logiciel en cours de développement fonctionnera sur plusieurs systèmes matériels différents.

### Diagrammes comportementaux

- **Diagramme d'activité** : Décompose de grands processus en activités plus petites. Il est utilisé pour mettre l'accent sur un composant d'un système. Ce diagramme de comportement UML offre une vue d'ensemble des aspects dynamiques d'un système.
- **Diagramme de séquences** : Montrent comment les différents objets sont liés et interagissent les uns avec les autres dans un système. Cet outil aide les développeurs à comprendre comment, pourquoi et dans quel ordre ces interactions se produisent.
- **Diagramme de communication** : Sont également appelés diagrammes de collaboration. Ils sont relativement similaires aux diagrammes de séquence en ce sens qu'ils se concentrent sur les messages transmis entre différents objets. Alors qu'un diagramme de séquence examine le traitement dans le temps, un diagramme de communication crée une cartographie complète et globale de votre produit.
- **Diagramme de synthèse d'interactions** : Comme les diagrammes d'activités, les diagrammes de synthèse d'interactions représentent visuellement un flux d'activités et la séquence de ces activités. La différence réside dans le fait que chaque activité d'un diagramme de synthèse d'interactions est représentée par un cadre. Les nœuds d'un diagramme de synthèse d'interactions représentent les interactions dans un système.
- **Diagramme temporel** : Sont utiles pour les déploiements, car ils décrivent le comportement d'objets spécifiques dans un intervalle de temps explicite. Ces diagrammes sont généralement assez simples, cependant, lorsqu'ils traitent de

plus d'un objet, ils montrent les interactions entre de nombreuses séquences différentes et importantes dans cet intervalle de temps donné.

- **Diagramme de cas d'utilisation** : Les diagrammes de cas d'utilisation fournissent une vue d'ensemble graphique des acteurs impliqués dans un système logiciel. Ils aident les développeurs à analyser les relations entre les cas d'utilisation et les personnes.
- **Diagramme de machine d'état** : Montrent les comportements des différents composants d'un système. Vous pouvez visualiser comment les éléments agissent différemment en fonction de l'état du programme en cours de développement.
- **Diagramme de profils** : Sont l'un des nouveaux diagrammes introduits par UML 2. Ces diagrammes sont rarement utilisés et comprennent tout diagramme créé dans un paquetage de profil [16].

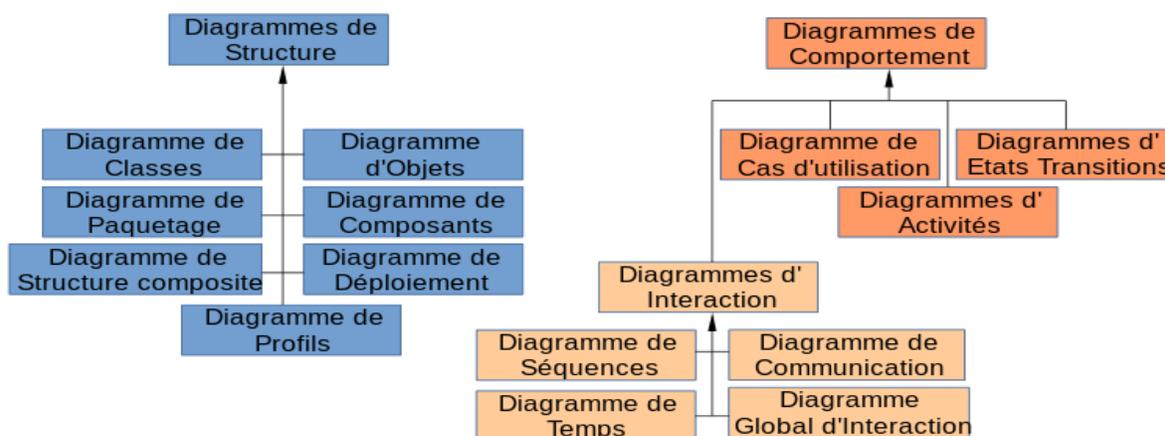


FIGURE 3.34 – Diagrammes UML

## 3.9 Description approfondie de certains diagrammes

### 3.9.1 Diagramme de cas d'utilisation

Les cas d'utilisation constituent un moyen de recueillir et de décrire les besoins des acteurs du système. La représentation d'un diagramme de cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation [17].

**Acteur :** Un acteur se représente par un petit bonhomme ayant son nom inscrit dessous.

Un acteur peut aussi être formalisé par un rectangle contenant le stéréotype « acteur » avec son nom juste en dessous [18].



FIGURE 3.35 – Acteur

**Cas d'utilisation :** Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur [19].

Un cas d'utilisation se représente par un ovale dans lequel figure son intitulé.

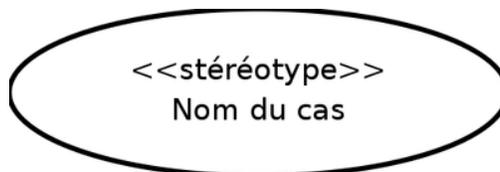


FIGURE 3.36 – Un cas d'utilisation

**Interaction :** Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation [19].

L'interaction entre un acteur et un cas d'utilisation se représente par une ligne (trait plein).

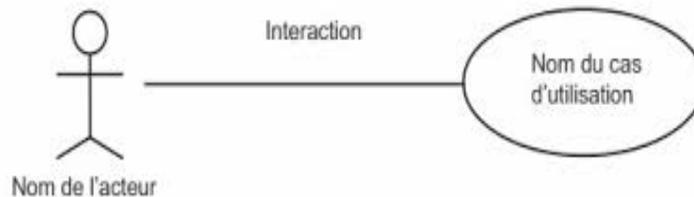


FIGURE 3.37 – Interaction

### Relations entre cas d'utilisation :

**Relation d'inclusion :** Une instance du cas d'utilisation source comprend (inclut) également le comportement décrit par le cas d'utilisation destination.

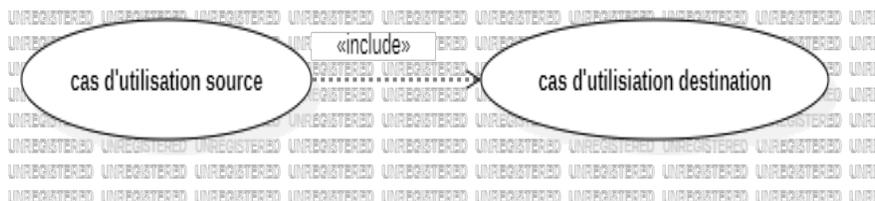


FIGURE 3.38 – Inclusion

**Relation d'extension :** Signifie que le cas d'utilisation source étend le comportement du cas d'utilisation destination.

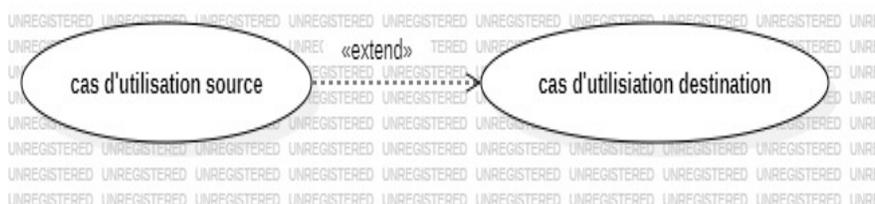


FIGURE 3.39 – Extension

**Relation de généralisation :** Le cas d'utilisation A est une généralisation de B si B est un cas particulier de A [20].



FIGURE 3.40 – Généralisation

### 3.9.2 Diagramme de classes

Le diagramme de classes est le plus important des diagrammes UML, c'est le seul qui soit obligatoire lors de la modélisation objet d'un système. Un diagramme de classes fournit une vue globale d'un système. Il permet de modéliser les classes du système et leurs relations. Le diagramme de classes standard est composé de trois sections :

**Section supérieure :** Contient le nom de la classe. Cette section est toujours nécessaire, que vous parliez du classifieur ou d'un objet.

**Section intermédiaire :** Contient les attributs de la classe. Utilisez-la pour décrire les qualités de la classe. Elle n'est nécessaire que lors de la description d'une instance spécifique d'une classe.

**Section inférieure :** Contient les opérations de la classe (méthodes), affichées sous forme de liste. Chaque opération occupe sa propre ligne. Les opérations décrivent la manière dont une classe interagit avec les données [21].

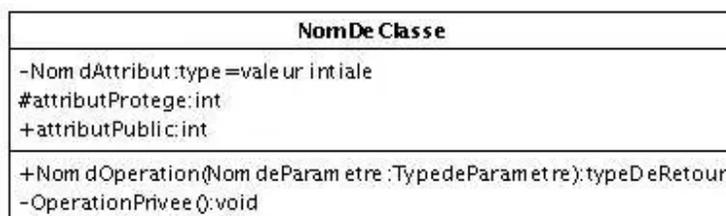


FIGURE 3.41 – Une classe avec ses attributs et ses opérations

### 3.9.3 Diagramme de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but est de décrire comment se déroulent les interactions entre les acteurs ou objets. Plusieurs types de messages (actions) peuvent transiter entre les acteurs et objets.

**message simple :** le message n'a pas de spécificité particulière d'envoi et de réception.

**message avec durée de vie** : l'expéditeur attend une réponse du récepteur pendant un certain temps et reprend ses activités si aucune réponse n'a lieu dans un délai prévu.

**message asynchrone** : le message est envoyé, l'expéditeur continue son activité que le message soit parvenu ou pris en compte ou non. Les messages asynchrones sont symbolisés par des demi-flèches.

**message déroband** : le message est mis en attente dans une liste d'attente de traitement chez le récepteur [22].

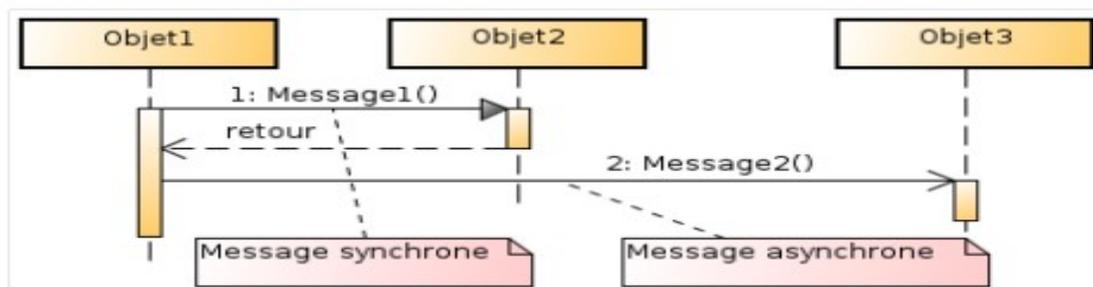


FIGURE 3.42 – Diagramme de séquence

### 3.9.4 Diagramme d'activités

Un diagramme d'activité UML permet de visualiser un certain cas d'utilisation à un niveau plus détaillé. Il s'agit d'un diagramme de comportement qui illustre le flux des activités dans un système.

Les diagrammes d'activité UML peuvent également être utilisés pour représenter un flux d'événements dans un processus commercial. Ils peuvent être utilisés pour examiner les processus d'entreprise afin d'identifier son flux et ses besoins [23].

#### 3.9.4.1 Avantages des diagrammes d'activités

- Démontrer la logique d'un algorithme.
- Décrire les étapes effectuées dans un cas d'utilisation d'UML.
- Illustrer un processus métier ou un flux de travail entre les utilisateurs et le système.
- Simplifier et améliorer n'importe quel processus en clarifiant les cas d'utilisation complexes.
- Modéliser des éléments de l'architecture de logiciels, tels que la méthode, la fonction et l'utilisation.

### 3.9.4.2 Pourquoi utiliser des diagrammes d'activités ?

Comme pour les organigrammes et les autres schémas, les diagrammes d'activité sont un moyen de représenter visuellement le comportement dynamique au sein d'un système. Les diagrammes d'activité représentent le flux des programmes à un niveau élevé, ce qui permet aux utilisateurs de comprendre plus clairement les conditions et les contraintes associées à des opérations spécifiques et de voir le flux de messages entre les activités.

### 3.9.4.3 Composants de base d'un diagramme d'activités

Avant de commencer à créer un diagramme d'activités, vous devez d'abord comprendre de quoi il est constitué. Voici quelques-uns des composants les plus courants d'un diagramme d'activités :

**Action** : Étape dans l'activité où les utilisateurs ou le logiciel exécutent une tâche donnée. Dans Lucidchart, les actions sont symbolisées par des rectangles aux bords arrondis.

**Nœud de décision** : Embranchement conditionnel dans le flux, qui est représenté par un losange. Il comporte une seule entrée et au moins deux sorties.

**Flux de contrôle** : Autre nom donné aux connecteurs qui illustrent le flux entre les étapes du diagramme.

**Nœud de départ** : Élément symbolisant le début de l'activité, que l'on représente par un cercle noir.

**Nœud de fin** : Élément symbolisant l'étape finale de l'activité, que l'on représente par un cercle noir avec un contour [24].

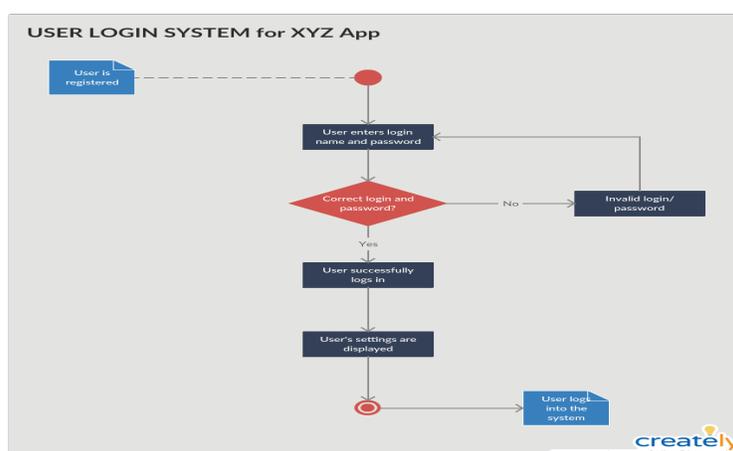


FIGURE 3.43 – Exemple d'un diagramme d'activités

# Bibliographie

- [1] <https://dz.kompass.com/c/beinx-algerie-eurl/dz269809/>.
- [2] <https://www.adresse-algerie.com/annuaire/eurl-beinx-algerie/>.
- [3] <https://www.com/fr/fiches-pratiques/organigramme-entreprise/#:~:text=Un%20organigramme%20d'entreprise%20est,la%20famille%20des%20indicateurs%20RH.>
- [4] <https://onlineshopdz.com/ouedkniss-comprendre-les-avantages-et-les-inconvenients/>.
- [5] <https://www.lkeria.com/conseils-immobilier-algerie>.
- [6] <https://www.lkeria.com/FAQ.html/>.
- [7] <https://www.advaloris.ch/gestion-de-projet/definir-besoins-gestion-projet>.
- [8] <https://mathieux51.gitbooks.io/howto/content/II.cahier-des-charges/2.expression-des-besoins/2.besoins-non-fonctionnels.html#>.
- [9] <https://www.axiocode.com/exigences-non-fonctionnelles/>.
- [10] <https://www.eyrolles.com/Informatique/Livre/uml-2-en-action-9782212121049/>.
- [11] [https://jsilab.ch/cfpa/ID/ref/UML/UML\\_2\\_ModeliserUneApplicationWeb.pdf](https://jsilab.ch/cfpa/ID/ref/UML/UML_2_ModeliserUneApplicationWeb.pdf).
- [12] <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>[.
- [13] [file:///C:/Users/VMI/Downloads/WP\\_109\\_Kolp%20\(2\).pdf](file:///C:/Users/VMI/Downloads/WP_109_Kolp%20(2).pdf).
- [14] [https://www.yumpu.com/fr/document/read/16739744/theme-modelisation-et-verification-des-processus-metiers-dans-les-#google\\_vignette](https://www.yumpu.com/fr/document/read/16739744/theme-modelisation-et-verification-des-processus-metiers-dans-les-#google_vignette).

- 
- [15] <https://www.lucidchart.com/pages/fr/langage-uml>.
- [16] <https://www.lucidchart.com/pages/fr/langage-uml>.
- [17] [https://www.memoireonline.com/09/18/10304/m\\_Developpement-d-une-application-web-pour-l-optimisation-du-processus-d-archivage-et-dacces-aux19.html](https://www.memoireonline.com/09/18/10304/m_Developpement-d-une-application-web-pour-l-optimisation-du-processus-d-archivage-et-dacces-aux19.html).
- [18] <https://lipn.univ-paris13.fr/~grozavu/CGP/data/Cours/GL2.pdf>.
- [19] [http://ibrahimguelzim.atspace.co.uk/\\_\\_\\_chap\\_05\\_cours\\_C00\\_UML2\\_diag\\_comp.pdf](http://ibrahimguelzim.atspace.co.uk/___chap_05_cours_C00_UML2_diag_comp.pdf).
- [20] <https://www.uv.es/nemiche/cursos/UML2.pdf>.
- [21] <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>.
- [22] [https://fr.wikipedia.org/wiki/Diagramme\\_de\\_s%C3%A9quence](https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence).
- [23] <https://creately.com/blog/fr/uncategorized-fr/tutorial-diagrammes-de-activite-uml/>.
- [24] <https://www.lucidchart.com/pages/fr/diagramme-dactivite-uml>.
- [25] <https://blog.ippon.fr/2020/10/16/diagramme-de-contexte-vs-hexagone/#:~:text=Le%20diagramme%20de%20contexte%20est%20une%20mod%C3%A9lisation%20permettant%20de%20mettre,et%20les%20POs%20du%20projet..>
- [26] [https://www.memoireonline.com/12/22/13543/m\\_Mise-en-place-d-une-application-web-mobile-pour-la-prise-en-charge-des-hypertensions36.html](https://www.memoireonline.com/12/22/13543/m_Mise-en-place-d-une-application-web-mobile-pour-la-prise-en-charge-des-hypertensions36.html).
- [27] <https://grafikart.fr/formations/laravel>.
- [28] [https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/#questce-que-laravel-%20\(consult%C3%A9%20le%2003/06/2021\)](https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-laravel/#questce-que-laravel-%20(consult%C3%A9%20le%2003/06/2021)).
- [29] <https://talks.freelancerepublik.com/comprendre-utiliser-architecture-mvc/>.
- [30] <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/bootstrap-5-les-nouveautes/>.
- [31] [https://developer.mozilla.org/fr/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics).

- 
- [32] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203277-css-cascading-style-sheets-definition-tra>.
- [33] <https://aws.amazon.com/fr/what-is/javascript/#:~:text=JavaScript%20est%20un%20langage%20de,animations%20et%20de%20cartes%20interactives>.
- [34] <https://jobphoning.com/dictionnaire/php#>.
- [35] <https://bility.fr/definition-visual-studio-code/>.
- [36] <https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=fr#0>.
- [37] <https://www.uptrends.fr/qu-est-ce-que/rest-api>.
- [38] [https://services.montefiore.uliege.be//verif/cours/bd/repet2014/tp7\\_slides.pdf](https://services.montefiore.uliege.be//verif/cours/bd/repet2014/tp7_slides.pdf).
- [39] <https://fr.wikipedia.org/wiki/StarUML#:~:text=StarUML%20est%20un%20outil%20de,sous%20Windows%2C%20Linux%20et%20MacOS>.
- [40] <https://www.patriciafiliatrault.com/graphisme/les-avantages-dutiliser-canva-et-canva-pro-pour-vos-contenus-visuels/>.

## RÉSUMÉ

Ce mémoire de master professionnel présente le développement d'une application web et mobile révolutionnaire dédiée à la gestion complète d'une agence immobilière. L'objectif principal est d'offrir aux professionnels de l'immobilier un outil robuste et convivial pour gérer efficacement leur portefeuille de biens, leurs clients et leurs transactions. L'application offre une gamme complète de fonctionnalités, de la publication d'annonces immobilières à la gestion des visites, en passant par le suivi des clients et des offres. En utilisant les dernières technologies du web et du mobile, cette solution offre une expérience utilisateur optimale tout en répondant aux besoins spécifiques du secteur immobilier. Ce projet représente une avancée significative dans le domaine du génie logiciel en combinant des technologies modernes avec une conception centrée sur l'utilisateur pour répondre aux défis uniques de l'industrie immobilière.

**Mots clés :** application web, application mobile, gestion agence immobilière, Agile, UML, Laravel, PHP, HTML, CSS, Bootstrap, JavaScript, MySQL, API REST.

## ABSTRACT

This professional master's thesis presents the development of a revolutionary web and mobile application dedicated to the complete management of a real estate agency. The main objective is to offer real estate professionals a robust and user-friendly tool to effectively manage their property portfolio, clients and transactions. The application offers a full range of features, from publishing real estate ads to managing visits, tracking customers and offers. Using the latest web and mobile technologies, this solution offers an optimal user experience while meeting the specific needs of the real estate sector. This project represents a significant advancement in the field of software engineering by combining modern technologies with user-centered design to address the unique challenges of the real estate industry.

**Keywords :** web application, mobile application, real estate agency management, Agile, UML, Laravel, PHP, HTML, CSS, Bootstrap, JavaScript, MySQL, REST API.