



THÈME

***CONCEPTION ET RÉALISATION
D'UNE APPLICATION WEB
D'UNE AGENCE IMMOBILIÈRE***

Membres du groupe :

BELAIOUAR Anis(chef de projet)
OUTERBAH Mouhammed
SEKIOU Doua
CHOUAKI Ines
AISSAOUI AMANI

Encadrant :

Ms AZOUAOU Faical

2023/2024

Remerciements

Nous remercions Allah pour Sa guidance et Son soutien tout au long de ce projet. Sa grâce et Sa miséricorde nous ont permis de surmonter les défis, d'apprendre de précieuses leçons et d'atteindre nos objectifs.

Un grand merci à notre équipe pour son travail acharné, sa collaboration et son dévouement. Chacun a apporté une contribution essentielle, faisant de ce projet une réussite collective grâce à notre complémentarité et notre esprit d'équipe.

Enfin, nous exprimons notre gratitude à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce projet.

Votre soutien a été d'une valeur inestimable.

Table des matières

1	Introduction Générale	5
2	Diagramme de Cas d'Utilisation	6
2.1	Description du Diagramme de Cas d'Utilisation	8
3	Diagrammes de Séquence	10
3.1	Créer un compte	10
3.2	Déposer une annonce	12
3.3	Faire une recherche	13
3.4	Authentification	14
4	Diagrammes de Classes	15
5	Diagrammes de Composants	16
6	Choix Technologiques Retenues	17
6.1	Introduction	17
6.2	Next.js	17
6.3	Express	17
6.4	LaTeX	17
6.5	Visual Studio Code	18
6.6	Figma	19
6.7	GitHub	19
6.8	Overleaf	19
6.9	Trello	19
6.10	Draw.io	20
6.11	Postman	20
7	Environnement de Collaboration et de Développement	21
7.1	Gestion de Tâches, Documents et Planning de Réunions avec Trello, Discord et la Méthode Scrum	21
7.2	Gestion du Code	21
7.2.1	GitHub	21
7.2.2	Visual Studio Code (VSCoDe)	22
8	Les Tests Unitaires	23
8.1	Tests de recherche d'un Estate	23

Table des figures

2.1	Diagramme de Cas d'utilisation	7
3.1	Diagramme de Sequence -Créer un compte-	11
3.2	Diagramme de Sequence -Déposer une annonce-	12
3.3	Diagramme de Sequence -Faire une recherche-	13
3.4	Diagramme de Sequence -Authentification-	14
4.1	Diagramme de Classes	15
5.1	Diagramme de Composants	16
6.1	Le Logo de Next.js	17
6.2	Le Logo de Express	17
6.3	Le Logo de LaTeX	18
6.4	Le Logo de VISUAL STUDIO CODE	18
6.5	Le Logo de FIGMA	19
6.6	Le Logo de GITHUB	19
6.7	Le Logo de OVERLEAF	19
6.8	Le Logo de Trello	19
6.9	Le Logo de Draw.io	20
6.10	Le Logo de POSTMAN	20

Chapitre 1

Introduction Générale

Actuellement, le monde connaît un avancement technologique considérable dans tous les secteurs, et cela grâce à l'informatique qui joue un rôle important dans le développement de nombreuses entreprises et organisations. L'informatique propose des solutions nombreuses, et une d'elles sont les applications web qui sont très répandues sur le marché technologique de notre temps. Dans cette évolution constante, le secteur immobilier ne fait pas exception, Il connaît une transformation digitale majeure, avec l'émergence de plateformes web dédiées à la publication d'annonces et à la mise en relation entre acquéreurs, vendeurs et bailleurs. En Algérie, si certaines solutions existent déjà, le marché reste demandeur d'outils plus performants et répondant mieux aux besoins des acteurs du secteur. C'est dans ce contexte que s'inscrit le projet de création de **DZ-Estate**, une nouvelle plateforme immobilière dans le but de simplifier le processus immobilier. Le présent rapport se focalise sur la conception et la réalisation de l'application web beity qui a comme objectif principal : fournir une plateforme complète où les utilisateurs peuvent publier des annonces de vente et de location de biens immobiliers, ainsi que soumettre des demandes de visite pour les propriétés disponibles. Avec une vision de simplifier et d'accélérer le processus de transaction immobilière en offrant une interface conviviale et des fonctionnalités intuitives, en mettant l'accent sur la simplicité et l'efficacité de la publication et de la gestion des annonces, ainsi que sur la facilitation des demandes de visite pour les utilisateurs intéressés.

Chapitre 2

Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation décrit les interactions entre les acteurs du système et les différentes fonctionnalités offertes par l'application. Quatre acteurs principaux sont identifiés dans ce système : le **User**, l'**Admin**, l'**Member**, et l'**Premium Member**.

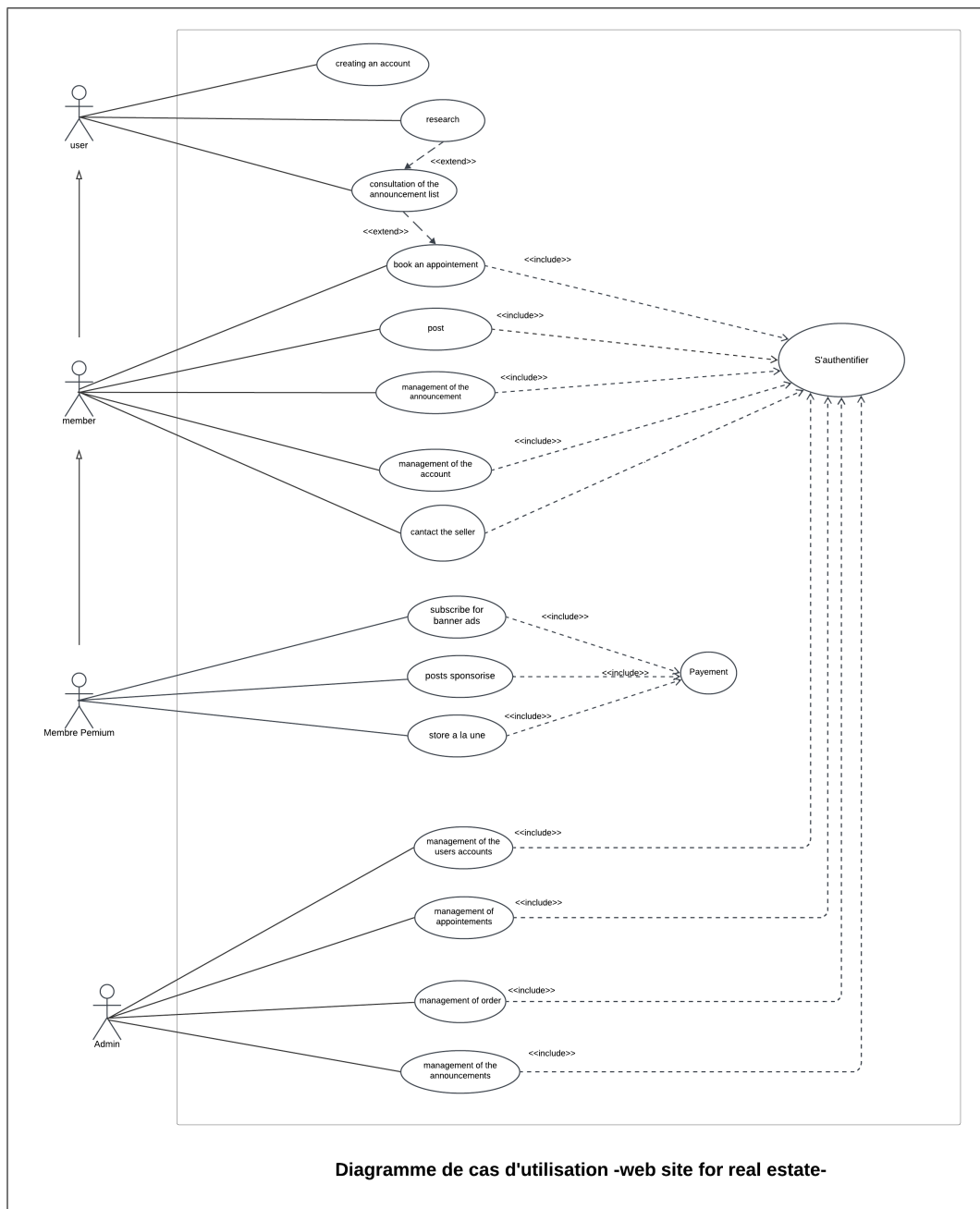


FIGURE 2.1 – Diagramme de Cas d'utilisation

2.1 Description du Diagramme de Cas d'Utilisation

Créer un compte

Sommaire d'identification	
Titre	Créer un compte
Acteur	Visiteur
Résumé	Le visiteur remplit un formulaire pour créer un compte.
Description des scénarios	
Précondition	Être sur la page d'accueil du site.
Scénario nominal	1. Le visiteur demande à créer un compte. 2. L'application affiche le formulaire d'inscription. 3. Le visiteur remplit tous les champs obligatoires et valide. 4. Un message de succès est affiché et son espace personnel est ouvert.
Scénario alternatif	Inscription non effectuée : 1. L'un des champs obligatoires est vide ou incorrect. 2. L'utilisateur existe déjà.
Postcondition	Le nouveau compte est enregistré dans la base de données et l'espace personnel est créé.

TABLE 2.1 – Cas d'utilisation « Créer un compte »

Faire une recherche

Sommaire d'identification	
Titre	Faire une recherche
Acteur	Visiteur, Membre
Résumé	L'acteur remplit les champs du formulaire de recherche et le valide afin de lancer la recherche.
Description des scénarios	
Précondition	Être sur la page de recherche du site.
Scénario nominal	1. L'acteur remplit les champs de recherche et valide. 2. Les résultats de recherche sont affichés.
Scénario alternatif	Critère de recherche non disponible : * Afficher un message résultat vide.
Postcondition	L'utilisateur a trouvé l'annonce qu'il cherche.

TABLE 2.2 – Cas d'utilisation « Faire une recherche »

Consulter la liste des annonces

Sommaire d'identification	
Titre	Consulter la liste des annonces
Acteur	Visiteur, Membre
Résumé	L'acteur consulte la liste des annonces disponibles.
Description des scénarios	
Précondition	Être sur la page d'accueil ou de recherche du site.
Scénario nominal	1. L'acteur accède à la liste des annonces. 2. Les annonces sont affichées.
Scénario alternatif	Aucune annonce disponible : * Afficher un message indiquant qu'il n'y a pas d'annonces.
Postcondition	L'utilisateur a consulté la liste des annonces.

TABLE 2.3 – Cas d'utilisation « Consulter la liste des annonces »

Réserver un rendez-vous

Sommaire d'identification	
Titre	Réserver un rendez-vous
Acteur	Membre
Résumé	Le membre réserve un rendez-vous pour visiter un bien immobilier.
Description des scénarios	
Précondition	Être connecté et avoir sélectionné une annonce.
Scénario nominal	1. Le membre sélectionne une annonce. 2. Le membre choisit une date et une heure pour le rendez-vous. 3. Le membre confirme la réservation.
Scénario alternatif	Aucune disponibilité : * Afficher un message indiquant qu'il n'y a pas de disponibilité.
Postcondition	Le rendez-vous est enregistré dans le système.

TABLE 2.4 – Cas d'utilisation « Réserver un rendez-vous »

Chapitre 3

Diagrammes de Séquence

3.1 Créer un compte

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

- Consulter l'application.
- Demander à s'inscrire.
- Le système répond en affichant le formulaire d'inscription.
- Remplir le formulaire et le valider.
- Le système vérifie si les champs ne sont pas vides, puis vérifie si les informations sont correctes :
 - Si un champ est vide, il renvoie un message d'erreur.
 - Sinon, il envoie les informations à la base de données (si les informations sont incorrectes alors il envoie un message d'erreur / sinon il met à jour la base de données puis affiche l'espace membre créer).

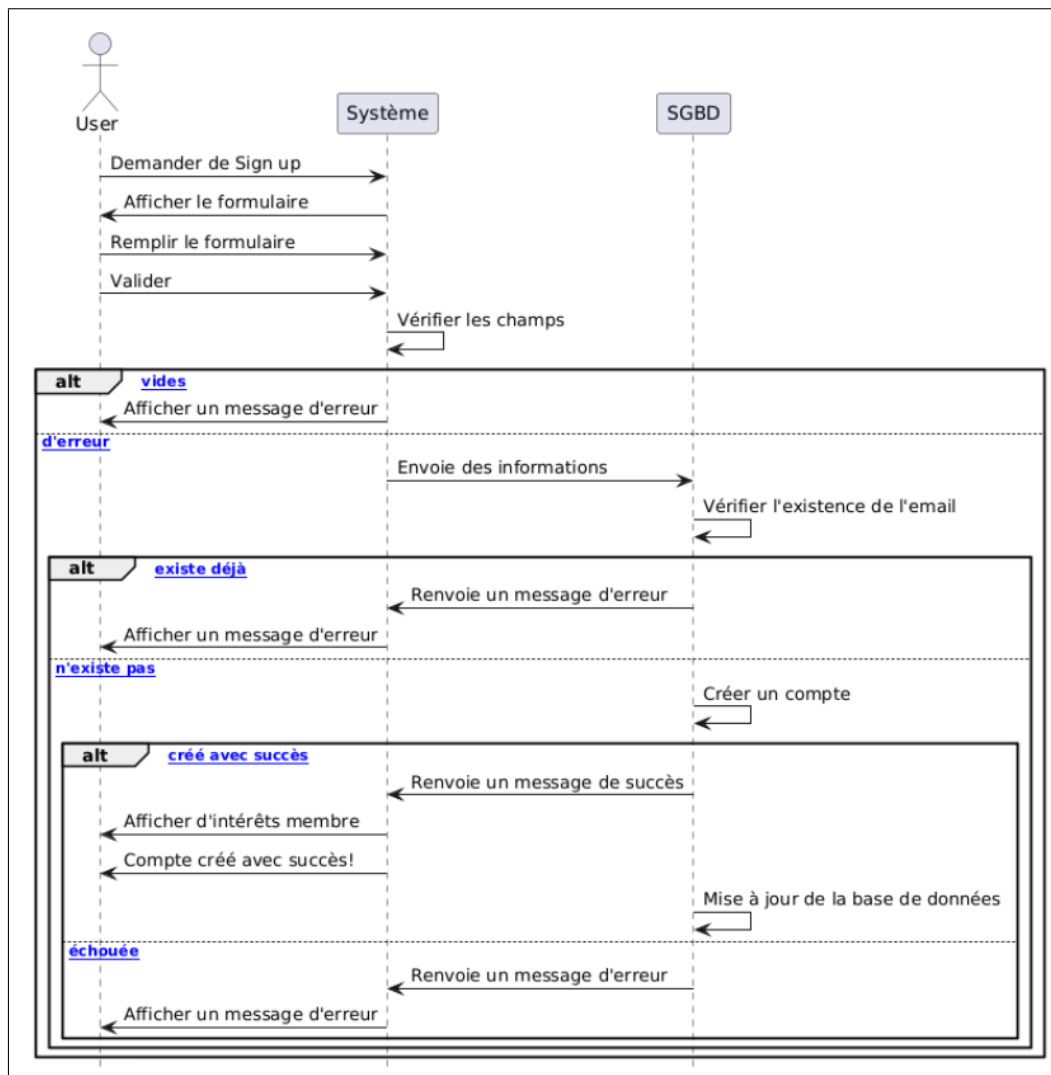


FIGURE 3.1 – Diagramme de Sequence -Créer un compte-

3.2 Déposer une annonce

Demande d'ajout d'une annonce.

- vérification de l'authentification :
 - Si il n'est pas authentifié(demande d'inscription).
 - Sinon :
 - Le système répond en affichant un formulaire à remplir.
 - Le membre remplit le formulaire et le valide.
 - Le système procède à la vérification de tous les champs du formulaire :
 - Si l'un des champs obligatoires est vide ou l'information saisie est incorrecte alors le système retourne un message d'erreur.
 - Si les champs sont tous bien remplis sans fausses informations alors le système valide le formu

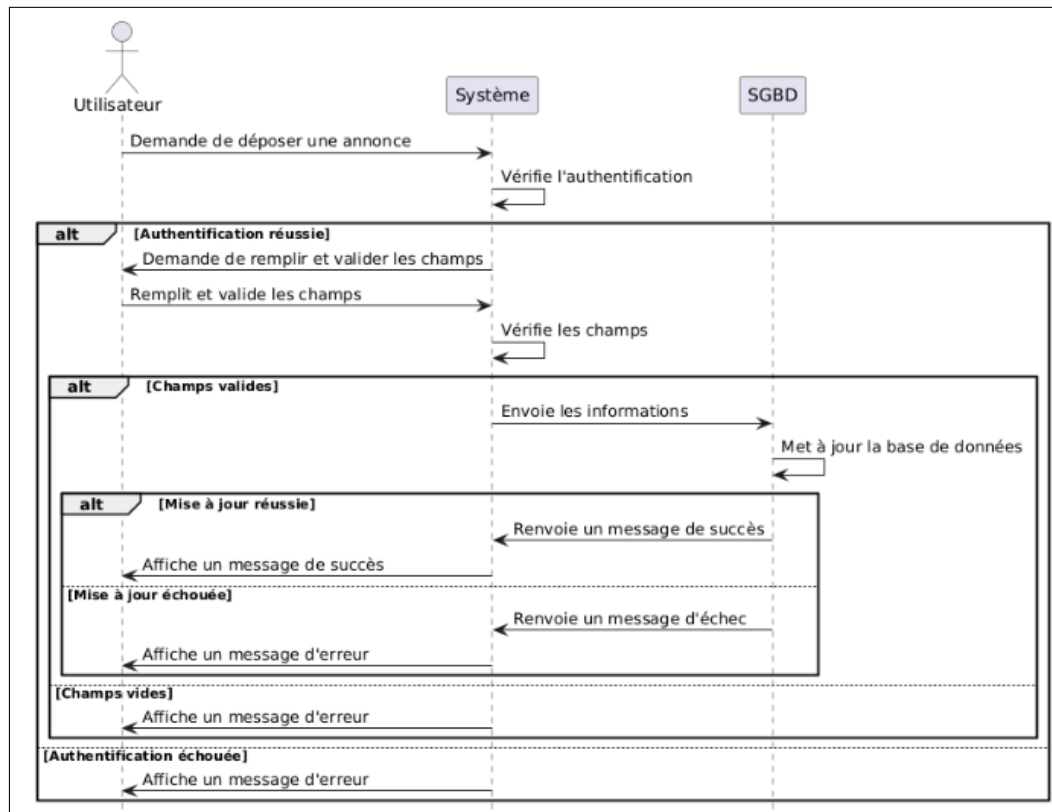


FIGURE 3.2 – Diagramme de Sequence -Déposer une annonce-

3.3 Faire une recherche

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

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

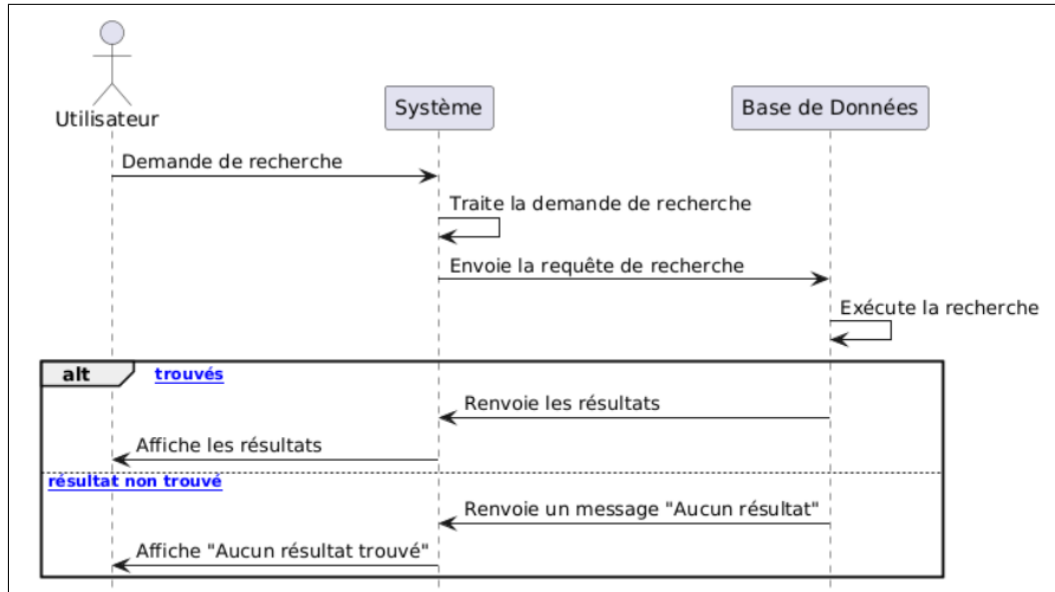


FIGURE 3.3 – Diagramme de Sequence -Faire une recherche-

3.4 Authentification

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

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

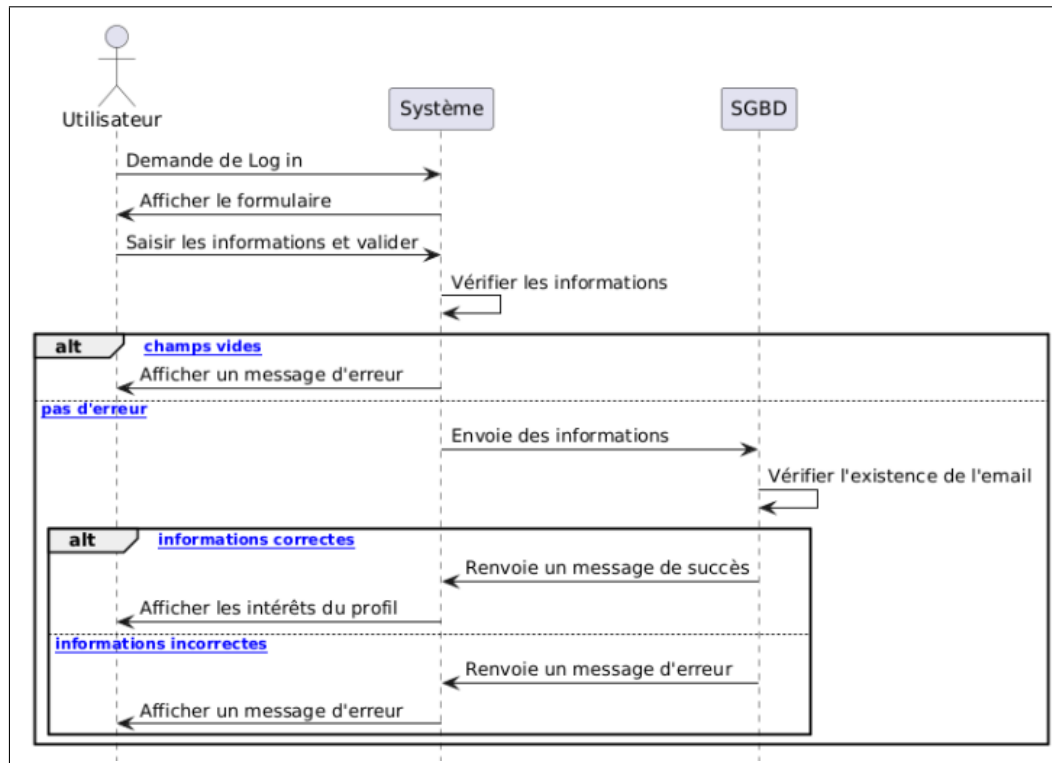


FIGURE 3.4 – Diagramme de Sequence -Authentification-

Chapitre 4

Diagrammes 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. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction

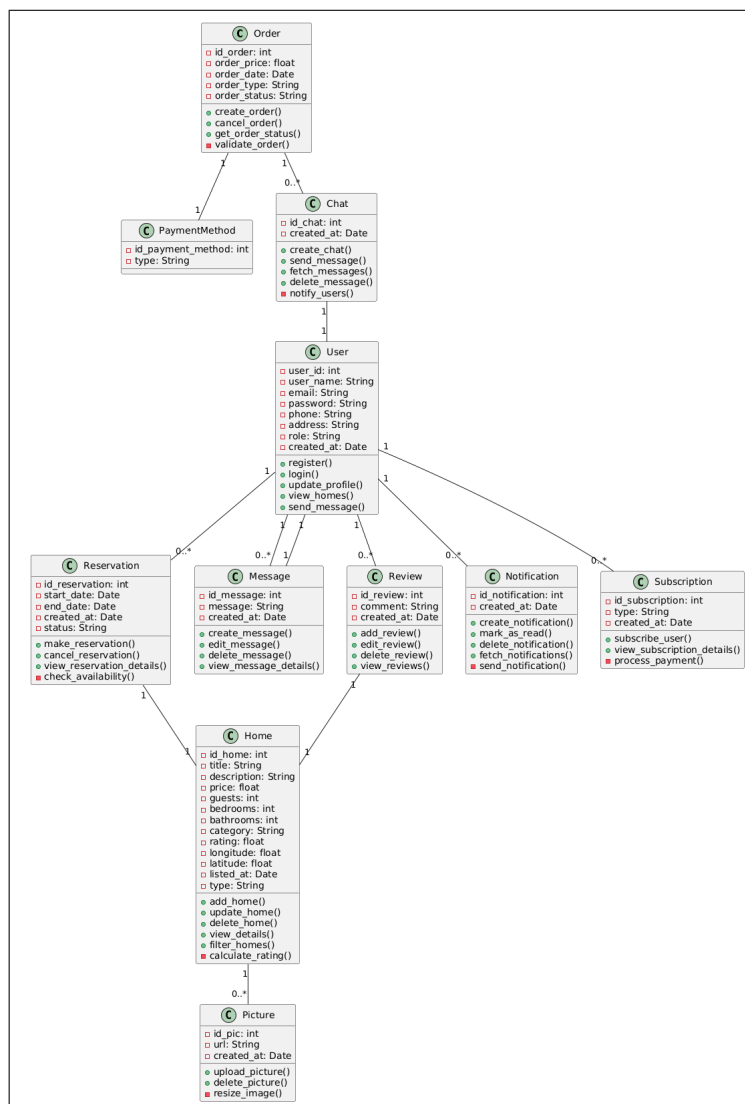


FIGURE 4.1 – Diagramme de Classes

Chapitre 5

Diagrammes de Composants

Un diagramme de composants modélise l'architecture d'un système en montrant ses composants logiciels, leurs interfaces et leurs interactions. Il met en avant les modules fonctionnels (bibliothèques, services, applications) et leurs dépendances. Chaque composant expose des interfaces pour communiquer avec d'autres. Les relations entre composants illustrent leur collaboration pour réaliser les fonctionnalités du système.

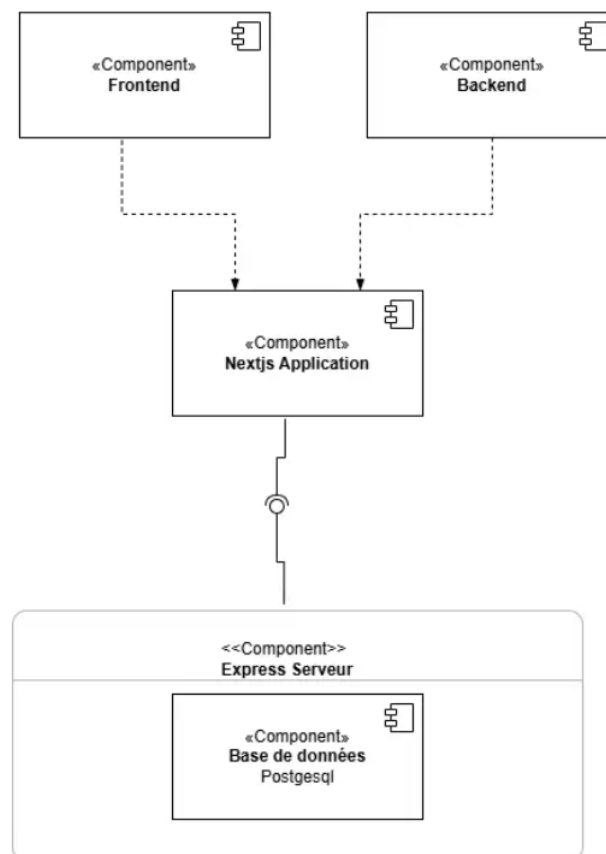


FIGURE 5.1 – Diagramme de Composants

Chapitre 6

Choix Technologiques Retenues

6.1 Introduction

Dans cette section, nous explorerons en détail La phase de réalisation qui marque la dernière étape cruciale du cycle de vie de notre projet de développement d'une application web de gestion d'une agence immobilière **DZ-Estate**. Après avoir passé par toutes les étapes précédentes, nous sommes maintenant prêts à présenter l'environnement logiciel ainsi que les différentes technologies que nous avons utilisées pour la création de l'application. Nous allons également vous montrer les différentes interfaces que nous avons conçues pour permettre une utilisation facile et intuitive de l'application.

6.2 Next.js

Next.js est un framework React open source qui permet de créer des applications web modernes avec des fonctionnalités telles que le rendu côté serveur (SSR), la génération de sites statiques (SSG), et le routage dynamique. Il est conçu pour offrir des performances optimales et une expérience de développement fluide, ce qui en fait un choix populaire pour les développeurs front-end.



FIGURE 6.1 – Le Logo de Next.js

6.3 Express

Express est un framework web minimaliste et flexible pour Node.js, conçu pour faciliter la création d'applications web et d'API. Il offre une structure simple pour gérer les routes, les requêtes HTTP, et les middlewares, ce qui permet aux développeurs de créer des serveurs backend robustes et évolutifs.



FIGURE 6.2 – Le Logo de Express

6.4 LaTeX

LaTeX est un système de composition de documents largement utilisé dans le domaine universitaire et scientifique pour la création de documents de haute qualité, notamment des articles, des rapports, des thèses

et des livres. Il permet aux utilisateurs de produire des documents bien structurés et professionnels avec une grande variété de fonctions, telles que la gestion des références bibliographiques, la création de tableaux et de graphiques, et la mise en forme de l'équation mathématique complexe.



FIGURE 6.3 – Le Logo de LaTeX

6.5 Visual Studio Code

Visual Studio Code est un éditeur de code source développé par Microsoft pour Windows, Linux et macOS. Il prend en charge plusieurs langages de programmation et fournit des fonctionnalités telles que la coloration syntaxique, l'achèvement de code, le débogage et la gestion de code source. Il dispose également d'une grande variété d'extensions développées par la communauté pour améliorer les fonctionnalités de l'éditeur.



FIGURE 6.4 – Le Logo de VISUAL STUDIO CODE

6.6 Figma

Figma est un outil de conception d'interface utilisateur (UI) et de conception d'expérience utilisateur (UX) basé sur le cloud. Il permet aux équipes de conception de collaborer en temps réel, de créer des maquettes d'interface utilisateur, des prototypes interactifs et des designs de haute qualité pour les applications mobiles et web.



FIGURE 6.5 – Le Logo de FIGMA

6.7 GitHub

GitHub est une plateforme de développement collaboratif de logiciels permettant aux utilisateurs de stocker, gérer et partager du code source avec d'autres développeurs. Elle offre une variété d'outils et de fonctionnalités pour faciliter la collaboration entre les membres d'une équipe de développement, y compris le contrôle de version, le suivi des problèmes, la gestion de projet, l'hébergement de code source et bien plus encore.



FIGURE 6.6 – Le Logo de GITHUB

6.8 Overleaf

Overleaf est un éditeur de texte collaboratif en ligne conçu spécialement pour la création de documents scientifiques tels que des articles de recherche, des thèses, des présentations et des posters.



FIGURE 6.7 – Le Logo de OVERLEAF

6.9 Trello

Trello est un outil de gestion de projet basé sur le système de tableaux Kanban. Il permet aux équipes de visualiser leurs tâches sous forme de cartes organisées en colonnes, ce qui facilite la planification, le suivi et la collaboration. Trello est particulièrement utile pour gérer les workflows et suivre l'avancement des projets en temps réel.



FIGURE 6.8 – Le Logo de Trello

6.10 Draw.io

Draw.io est un outil de diagrammation en ligne gratuit qui permet de créer des diagrammes de flux, des diagrammes de séquence, des diagrammes de classes, et bien d'autres. Il est largement utilisé pour la conception visuelle de systèmes et la documentation technique. Draw.io offre une interface intuitive et une intégration facile avec des plateformes comme Google Drive et GitHub.



FIGURE 6.9 – Le Logo de Draw.io

6.11 Postman

Postman est l'un des outils de test de logiciels les plus populaires utilisés pour les tests d'API. Avec l'aide de cet outil, les développeurs peuvent facilement créer, tester, partager et documenter des API.



FIGURE 6.10 – Le Logo de POSTMAN

Chapitre 7

Environnement de Collaboration et de Développement

7.1 Gestion de Tâches, Documents et Planning de Réunions avec Trello, Discord et la Méthode Scrum

Pour la gestion de notre projet, nous avons adopté une approche structurée en utilisant Trello pour l'organisation des tâches, Discord pour la communication, et la méthodologie Scrum pour la gestion agile du projet. Voici comment nous avons combiné ces outils et méthodes pour maximiser l'efficacité de notre équipe :

Gestion des Tâches avec Trello et Scrum : Trello a été notre outil principal pour la gestion des tâches, aligné sur les principes de Scrum. Nous avons organisé nos tableaux Kanban en colonnes reflétant les étapes du workflow Scrum : Backlog, À faire, En cours, et Terminé. Chaque carte représente une user story ou une tâche technique, avec des détails tels que les échéances, les responsables (Developers), et les fichiers joints. Le Product Owner priorise les tâches dans le Backlog, tandis que le Scrum Master veille à ce que l'équipe respecte les délais et les objectifs des sprints.

Documentation Centralisée avec Trello : Trello a également servi de plateforme centralisée pour la documentation. Chaque carte peut contenir des descriptions détaillées, des fichiers joints (comme des diagrammes, des documents ou des liens), et des commentaires. Cela a permis à tous les membres de l'équipe, y compris le Product Owner et les Developers, d'accéder facilement aux informations partagées et de collaborer de manière transparente.

Planification des Réunions avec Discord et Scrum : Discord a été notre outil de choix pour la communication et la planification des réunions Scrum. Nous avons créé des salons dédiés pour les Daily Stand-ups, les Sprint Plannings, les Sprint Reviews, et les Sprint Retrospectives. Ces réunions sont essentielles pour maintenir une communication fluide entre le Product Owner, le Scrum Master, et les Developers. Les fonctionnalités de messagerie instantanée et de vocal de Discord ont permis une communication en temps réel, favorisant la transparence et la responsabilité au sein de l'équipe.

Rôles dans l'Équipe Scrum :

- Product Owner : Responsable de la priorisation des tâches dans le Backlog et de la définition des objectifs du produit.

- Scrum Master : Garant du respect des pratiques Scrum, facilitateur des réunions, et soutien pour l'équipe de développement.

- Developers : En charge de la réalisation des tâches techniques et des user stories pendant les sprints.

En résumé, la combinaison de Trello, Discord, et la méthodologie Scrum a offert une solution complète pour la gestion de tâches, la documentation, et la planification des réunions. Ces outils et méthodes ont grandement contribué à l'efficacité opérationnelle de notre équipe, en facilitant la collaboration, en assurant une communication claire, et en respectant les principes agiles de Scrum.

7.2 Gestion du Code

7.2.1 GitHub

Dans le cadre de notre projet, GitHub a été notre plateforme principale pour la gestion du code source. Nous avons créé un dépôt (repository) privé pour le projet, permettant ainsi à tous les membres de l'équipe d'accéder au code de manière sécurisée. Les fonctionnalités de contrôle de version de Git sur GitHub nous ont permis de

gérer efficacement les différentes versions du code, de suivre les modifications apportées par chaque membre de l'équipe et de fusionner les contributions de manière transparente.

7.2.2 Visual Studio Code (VSCode)

Nous avons intégré Visual Studio Code directement avec GitHub pour une expérience de développement plus fluide. L'extension GitHub pour VSCode nous a permis de cloner le dépôt, créer des pull requests, et effectuer d'autres opérations directement depuis l'éditeur. VSCode a été notre choix d'éditeur de code en raison de sa légèreté, de sa polyvalence et de son support étendu pour différents langages de programmation. Les fonctionnalités telles que l'autocomplétion, le débogage intégré et la navigation aisée entre les fichiers ont amélioré notre productivité.

Chapitre 8

Les Tests Unitaires

8.1 Tests de recherche d'un Etat