

Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Constantine 2



Faculté des Nouvelles Technologies de l'Information et la de Communication  
Département des Technologies des Logiciels et Systèmes d'Information

---

Projet de fin d'études pour l'obtention du diplôme de  
Licence en Informatique

Option : Génie Logiciel

Thème

Conception et réalisation d'un site web dynamique pour un "Gestion des ventes pour une société immobilière"

Dirigé par :

-Mohamed Ali Sandel.

Réalisé par :

- Mohamed Amine Griche.  
- Boudiar Houssam.

- Session Juin 2018 -

## *Remerciement*

*Avant tout, nous remercions dieu le tout puissant qui nous a donné la force et le courage pour qu'on puisse accomplir ce modeste travail.*

*On remercie spécialement Monsieur M.Sandeli notre encadreur, qui nous a donné l'aide et l'assistance par sa disponibilité, ses Conseils et ses orientations durant toute la période de la réalisation de notre projet.*

*Nous exprimons nos reconnaissances à toute personne qui a contribué de près ou de loin à l'achèvement de ce travail, nos enseignants, nos amis, nos collègues et toute la promotion de GL 2018.*

*Nous remercions également les membres de jury d'avoir accepté de juger ce modeste travail.*

*Enfin, nous espérons que ce travail aura la valeur souhaitée.*

## *Dédicace*

## *Dédicace*

# Sommaire

## Table des matières

Introduction général :	7
Chapitre 1 : Etude préliminaire et spécification Présente des besoins .....	8
1- Introduction :	9
2- Etude préliminaire :	9
2.1 Présentation général du Project :	9
2.2 Définition des grands choix techniques :	9
2.3 Recueil des besoins fonctionnels :	10
2.4 Définition rapide sur les diagrammes :	10
3- Diagramme de cas d'utilisation :	11
4- Les Fiches Descriptives et leurs diagrammes de séquence :	14
5- Les Maquettes	28
6- Conclusion.....	30
Chapitre 2 Analyse et conception.....	31
1- Introduction :	32
2- Modelé de domaine :	32
3- Les Diagrammes des classes participantes :	33
4- Les Diagrammes de séquence :	36
5- Les Diagrammes des classes conception :	43
6-Conclusion :	44
Chapitre 3 : Implémentation .....	45
1- Introduction .....	46
2- Plateformes :	46
2.1 Plateforme matérielle :	46
2.2 Plateforme logicielle :	47
3- Langages de programmation :	47
4- Outils de développement :	50
5- Présentation de notre application :	52
3.1 Réalisation de la base de données :	52
3.2- Exemple Interface :	52

6- Conclusion.....	56
Chapitre 4 : Les Tests Unitaires.....	57
1- Introduction : .....	58
2- Les Unités :.....	58
Conclusion Générale.....	60

## Introduction général :

L'internet et l'ordinateur sont les nouveaux outils d'intégration planétaire, ils multiplient les contacts entre individus, nations, cultures et économies. Ce phénomène d'interaction offre des possibilités considérables et accroît les chances de réaliser des progrès sans précédents, ces outils révolutionnaires ont donné naissance à des cybercommunautés rassemblent des milliers de participants (particuliers ou entreprises) sur des vastes champs d'activités dans différents domaines notamment l'astronomie, les industries, la médecine. L'internet a libéré aujourd'hui les sociétés des contraintes de temps et d'espaces d'où ses avantages en matière de vitesse et de coût sont évidents. Le secteur de l'immobilier s'est engagé dans une importante transformation numérique en vue de répondre aux nouvelles attentes des particuliers comme des professionnels, pour qui la recherche d'annonces immobilières s'effectue désormais majoritairement en ligne. Cette transformation se traduit notamment par l'apparition de nouveaux services et donc par l'arrivée de nouveaux acteurs, qui permettent d'approfondir la dématérialisation.

L'objectif de mon projet de fin d'étude est la conception et la réalisation d'un système pour améliorer la gestion des ventes pour une société immobilière.

Pour l'analyse et le développement de notre site web, on applique le processus Pasqual Roques et on utilise des outils comme les notations du langage et de modélisation UML.

Dans ce mémoire nous allons organiser notre travail comme suivant :

- 1- Chapitre 1(Etude préliminaire et spécification Présente des besoins) : C'est là où nous allons présenter en détails le résultat de l'étude effectuée sur le système (Cahier de charge et les diagrammes obtenus par les différentes phases du processus de développement).
- 2- Chapitre 2 (Analyse et conception) : Décrit les différentes étapes d'analyse et de conception de notre système selon le processus Pasqual Roques.
- 3- Chapitre 3 (Implémentation) : Nous allons définir les outils utilisées dans la réalisation de notre site web dynamique, Aussi nous donnons quelques exemples d'interfaces.
- 4- Chapitre 4 (Les Testes Unitaire) : Nous allons tester des différentes méthodes dans notre implémentation on utilise Junit.
- 5- Conclusion général.

# Chapitre 1 : Etude préliminaire et spécification Présente des besoins

## 1- Introduction :

Dans ce chapitre, nous allons présenter la première étape du processus de développement Pascal Roques. Cette étape consiste à identifier et modéliser les besoins des différents utilisateurs à partir les diagrammes des cas d'utilisations. Les interactions entre les acteurs et le système (au sein des cas d'utilisation) seront explicitées sous forme textuelle (description textuelle), et sous forme graphique au moyen des diagrammes de séquence qui représentent exactement le déroulement de la tâche courante.

## 2- Etude préliminaire :

L'étude préliminaire est un processus de développement consiste à positionner précisément le champ du système étudié et à faire un premier repérage des besoins dans cette phase nous allons donner une représentation générale de projet :

### 2.1 Présentation général du Project :

L'objectif fondamentale de notre site est de réaliser une application web qui consiste à informatiser les différents aspects du fonctionnement d'une gestion des ventes pour une société immobilière et qui permet de :

- Gérer les comptes du clients, agents et operateurs (crée, modifie, supprimer . . .).
- Gérer les inscriptions des différents utilisateurs (accepter et refuser les inscriptions).
- Gérer les rendez-vous (prendre, proposer, annuler . . .).

### 2.2 Définition des grands choix techniques :

Pour la réussite de notre projet, nous avons fixé les choix techniques les mieux adaptés :

- A. Le processus de développement : Pascal Roques.
- B. L'outil de modélisation : UML 2.0.
- C. Langages de programmation : J2EE (JSP, SERVLET), Spring Framework, JAVA, JAVA SCRIPT, HTML, CSS, AJAX, JQuery.
- D. L'architecture de l'application : basée sur le model Client / Serveur.
- E. Le serveur de base de données : MYSQL
- F. La plateforme : Ubuntu 16.04.

## 2.3 Recueil des besoins fonctionnels :

Le site à été implémenté dans le but de réaliser les grandes fonctionnalités. Nous avons classé ces fonctionnalités par type d'utilisateur de la manière suivante :

- ✓ L'internaute peut :
  - ✓ Incrire.
  - ✓ Consulter les logements.
- ✓ Le Client peut :
  - ✓ Consulter les logements.
  - ✓ Consulter les Rendez-vous (Proposer un nouveau RDV, Annuler un RDV ou bien Signaler un RDV).
- ✓ L'Agent peut :
  - ✓ Consulter les logements.
  - ✓ Noter les opinions des clients.
- ✓ L'Opérateur peut :
  - ✓ Consulter les logements.
  - ✓ Gérer les comptes des clients.
- ✓ L'Administrateur peut :
  - ✓ Consulter les Statistique.
  - ✓ Gérer les comptes des Employées.
  - ✓ Gérer les logements.

## 2.4 Définition rapide sur les diagrammes :

### I- Le diagramme de cas d'utilisation :

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

### II- Les diagrammes de séquence système :

Les diagrammes de séquences mettent en valeur les échanges de messages (déclenchant des événements) entre acteurs et objets (ou entre objets et objets) de manière chronologique, l'évolution du temps se lisant de haut en bas. Chaque colonne correspond à un objet (décrit dans le diagramme des classes), ou éventuellement à un acteur, introduit dans le diagramme des cas. La ligne de vie de l'objet représente la durée de son interaction avec les autres objets du diagramme.

### 3- Diagramme de cas d'utilisation :

Puis que le diagramme de cas d'utilisation est très large on va le diviser en dépend sur les acteurs :

**Internaute :**

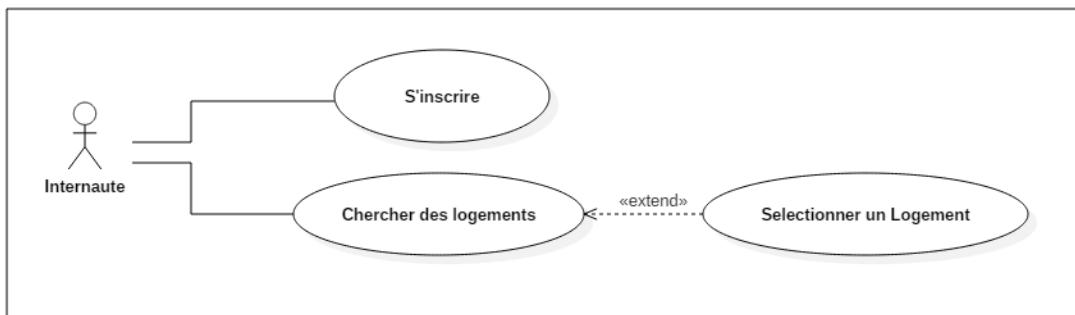


Figure 1 - Cas d'utilisation Internaute

**Client :**

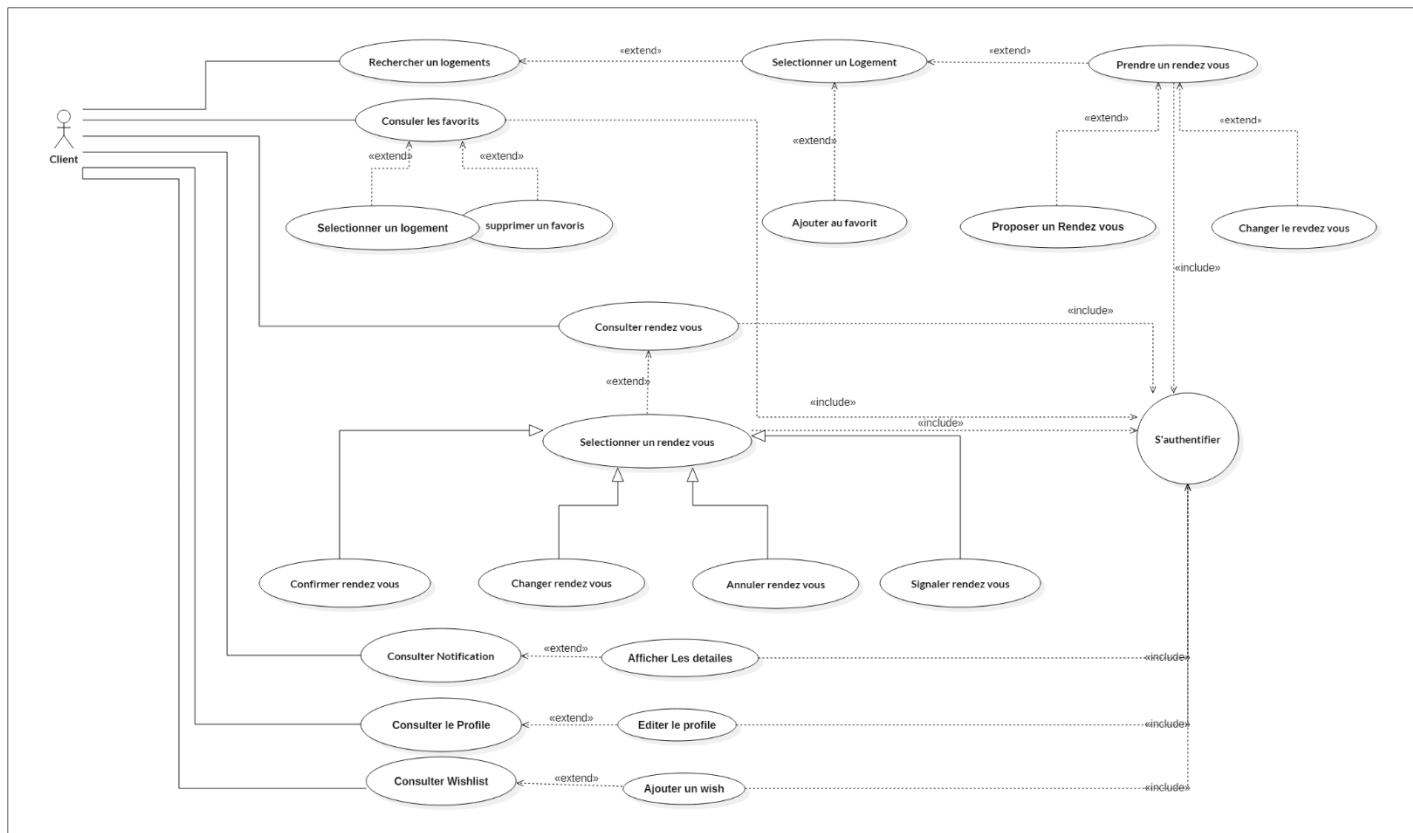


Figure 2- Cas d'utilisation Client

## Operateur :

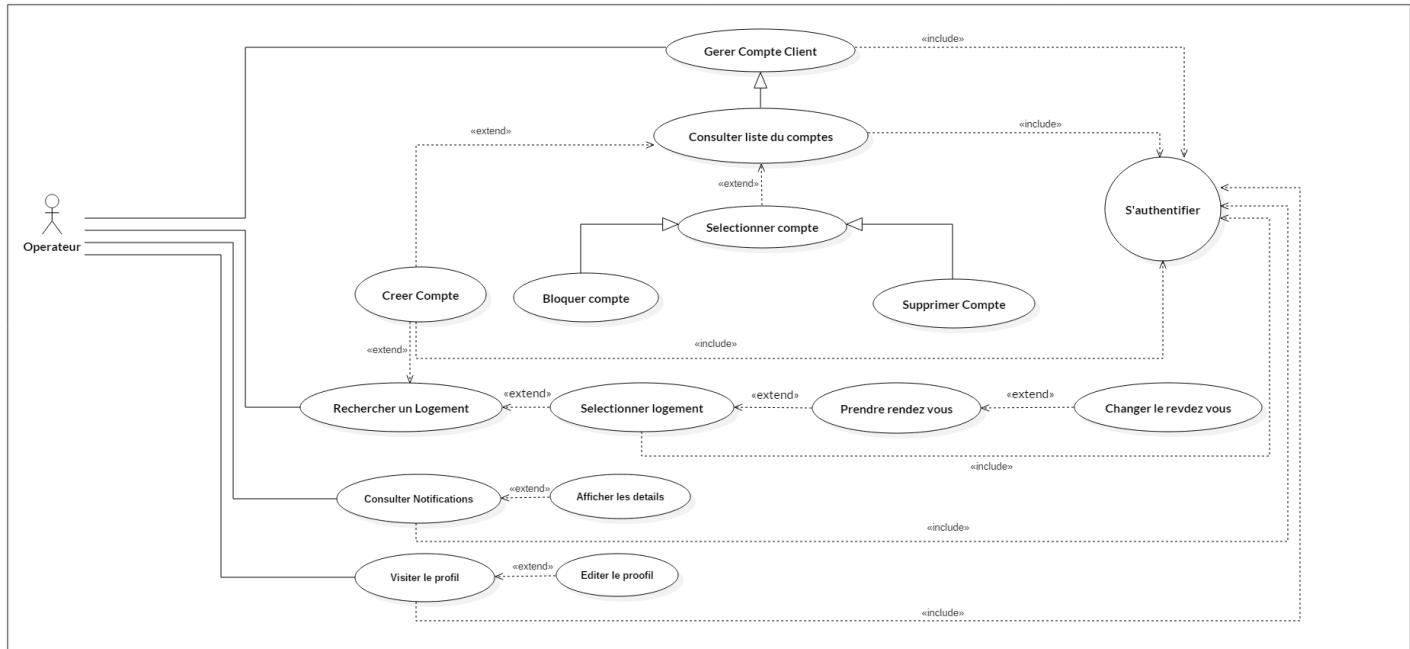


Figure 3- Cas d'utilisation Operateur

## Agent :

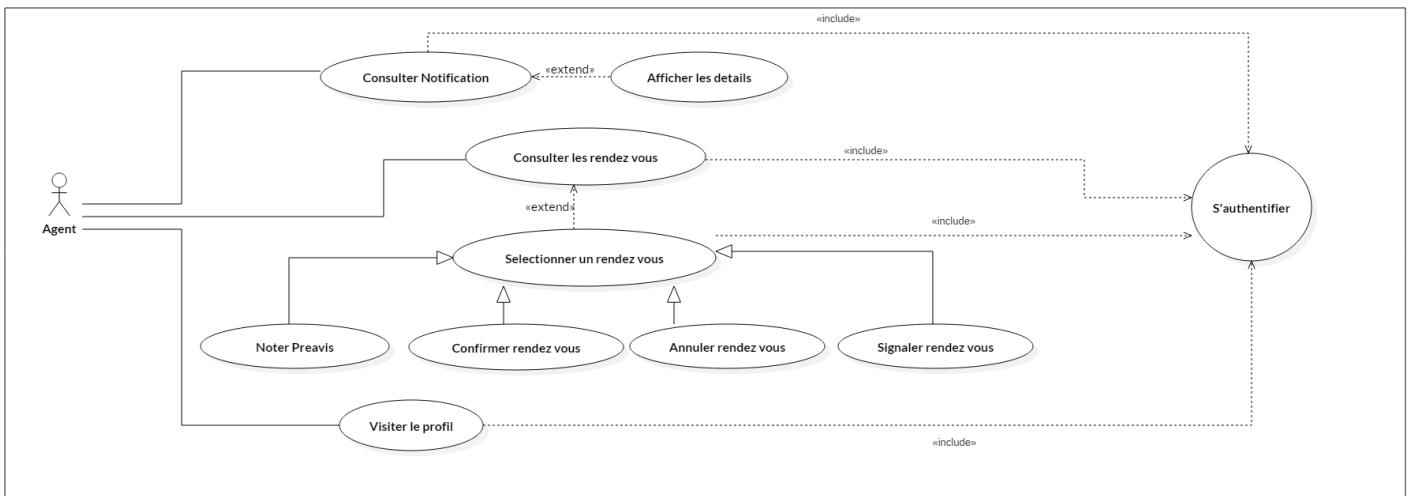


Figure 4- Cas d'utilisation Agent

## Admin :

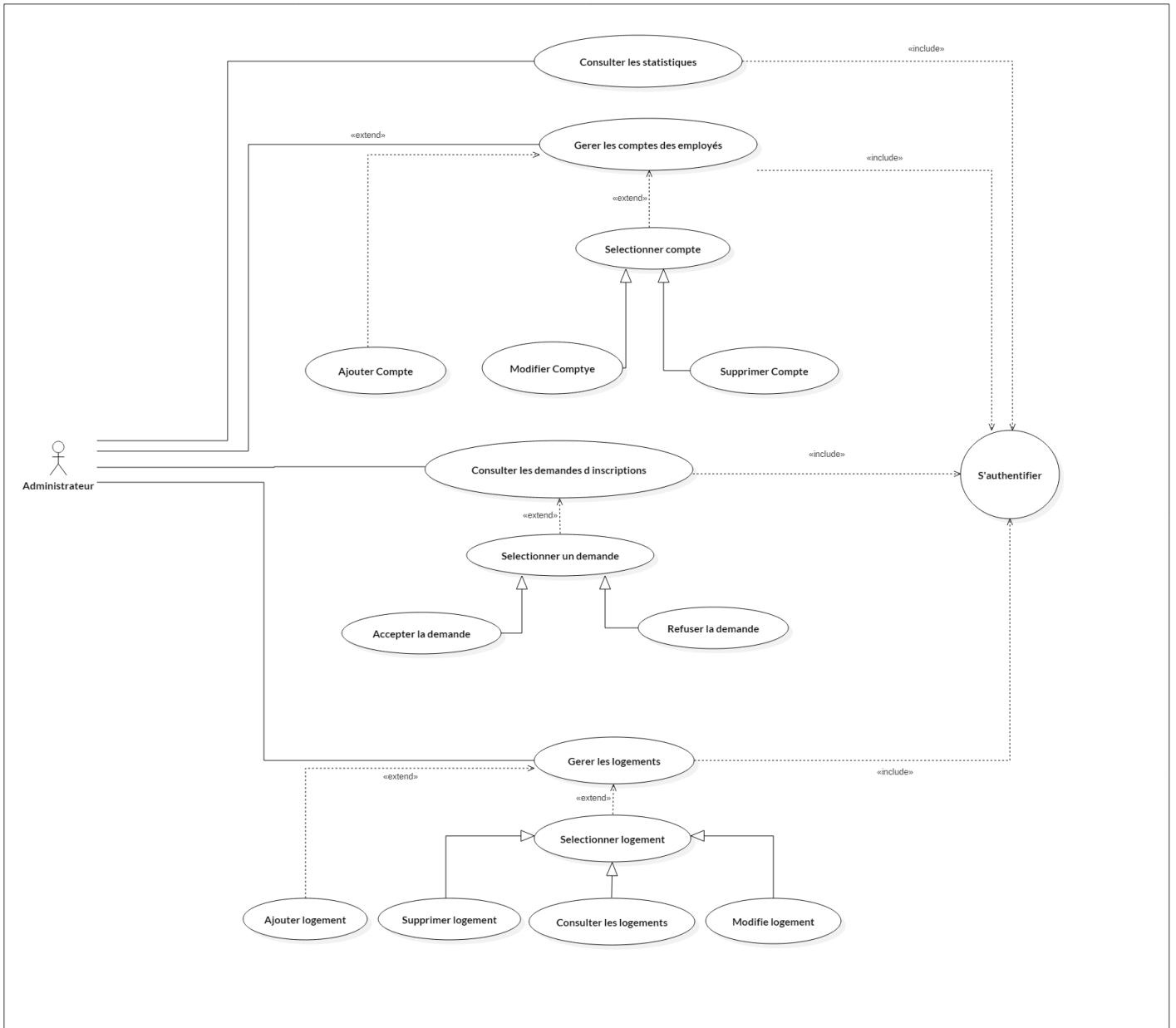


Figure 5 - Cas d'utilisation Admin

#### 4- Les Fiches Descriptives et leurs diagrammes de séquence :

- Cas d'utilisation : Incrire

- Fiche descriptive :

Cas d'utilisation	Incrire
Acteur	Internaute
Type	Primaire
Objectif	Permet au visiteur de s'inscrire dans le site
Préconditions	Néant
Post conditions	L'internaute dispose un compte
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur demande le formulaire d'inscription.</li> <li>2. Le système affiche le formulaire d'inscription.</li> <li>3. L'internaute remplit le formulaire.</li> <li>4. L'internaute envoie le formulaire.</li> <li>5. Le système enregistre le nouveau compte.</li> <li>6. Le système affiche la page d'accueil du nouveau l'utilisateur.</li> </ol>
Scénario Alternatif	<p>3.a Information invalide.</p> <p>a.1. Le système affiche un message d'erreur. Reprendre à l'étape 2</p>
Scénario Exception	<ul style="list-style-type: none"> <li>- Annuler l'envoie.</li> <li>- Compte existe déjà.</li> </ul>

- Diagramme de séquence :

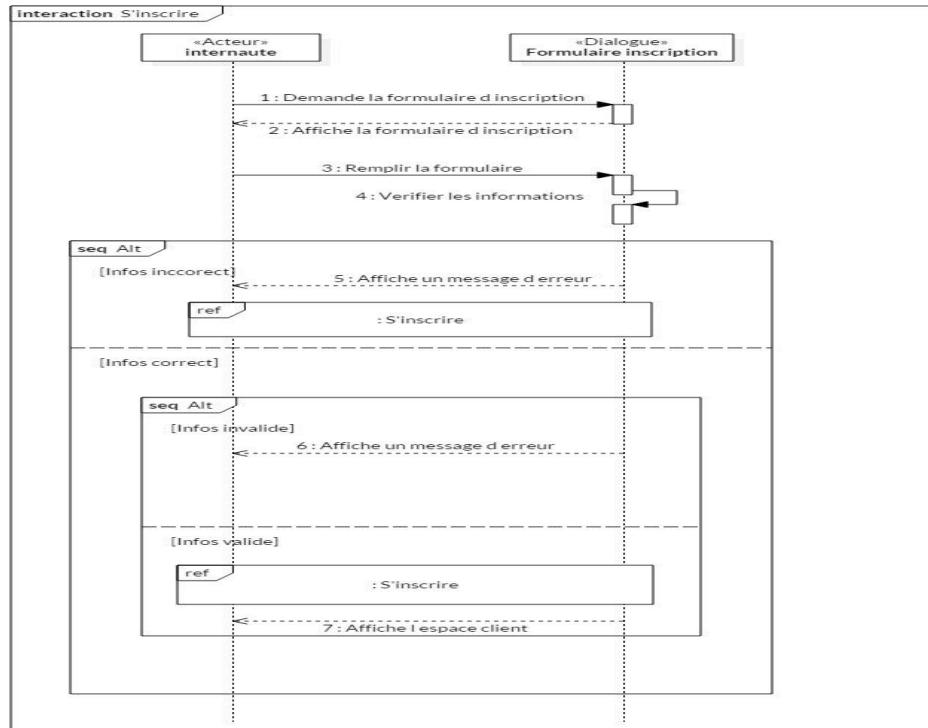


Figure 6 – Diagramme de séquence « Incrire »

## - Cas d'utilisation : Authentifier

### - Fiche descriptive :

Cas d'utilisation	S'authentifier
Acteur	Client , Operateur , Agent , Administrateur
Type	Secondaire
Objectif	Accéder à son propre espace.
Préconditions	Avoir un pseudo (email ou nom d'utilisateur)et un mot de passe .
Post conditions	Le client accédé a son espace de travail.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur demande le formulaire d'authentification.</li> <li>2. Le système affiche le formulaire d'authentification.</li> <li>3. L'utilisateur saisit son pseudo et mot de passe.</li> <li>4. Le système vérifie son existence.</li> <li>5. Le système affiche à l'utilisateur son espace de travail.</li> </ol>
Scénario Alternatif	
Scénario Exception	<ul style="list-style-type: none"> <li>- Pseudo ou mot de passe invalide.</li> <li>- Annulation.</li> </ul>

### Diagramme de séquence :

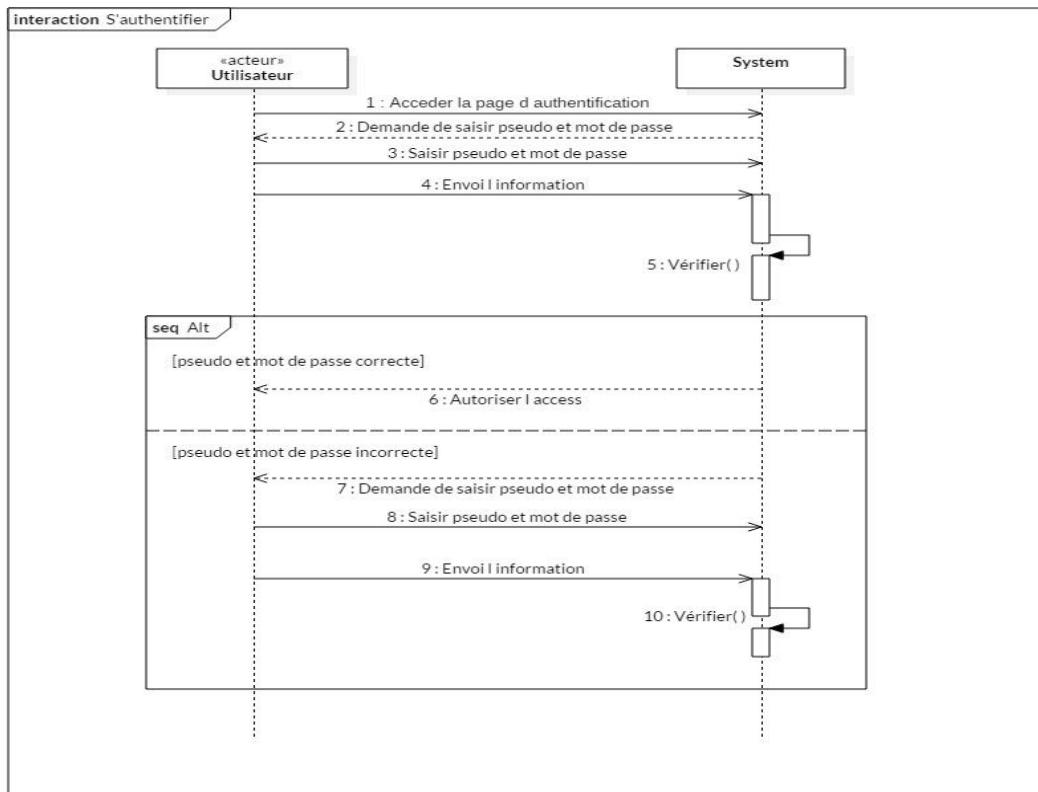


Figure 7 - Diagramme de séquence « Authentification »

- Cas d'utilisation : Consulter les logements

- Fiche descriptive :

Cas d'utilisation	Rechercher un logement
Acteur	Internaute , Client , Operateur
Type	Primaire
Objectif	Permet au Client de voir les logements disponible
Préconditions	Néant
Post conditions	Afficher les logements disponible
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur accéder l'espace de recherche.</li> <li>2. Le système affiche l'espace de recherche.</li> <li>3. L'utilisateur remplit les critères de recherche pour filtrer les logements.</li> <li>4. Le système renvoie la liste des logements qui convient les critères.</li> </ol>
Scénario Alternatif	<ol style="list-style-type: none"> <li>4.a Aucun logement était trouvé.             <ol style="list-style-type: none"> <li>a1. Le système affiche une liste vide.</li> <li>Reprendre à l'étape 2</li> </ol> </li> </ol>
Scénario Exception	<ul style="list-style-type: none"> <li>- L'utilisateur Annule la demande.</li> </ul>

- Diagramme de séquence :

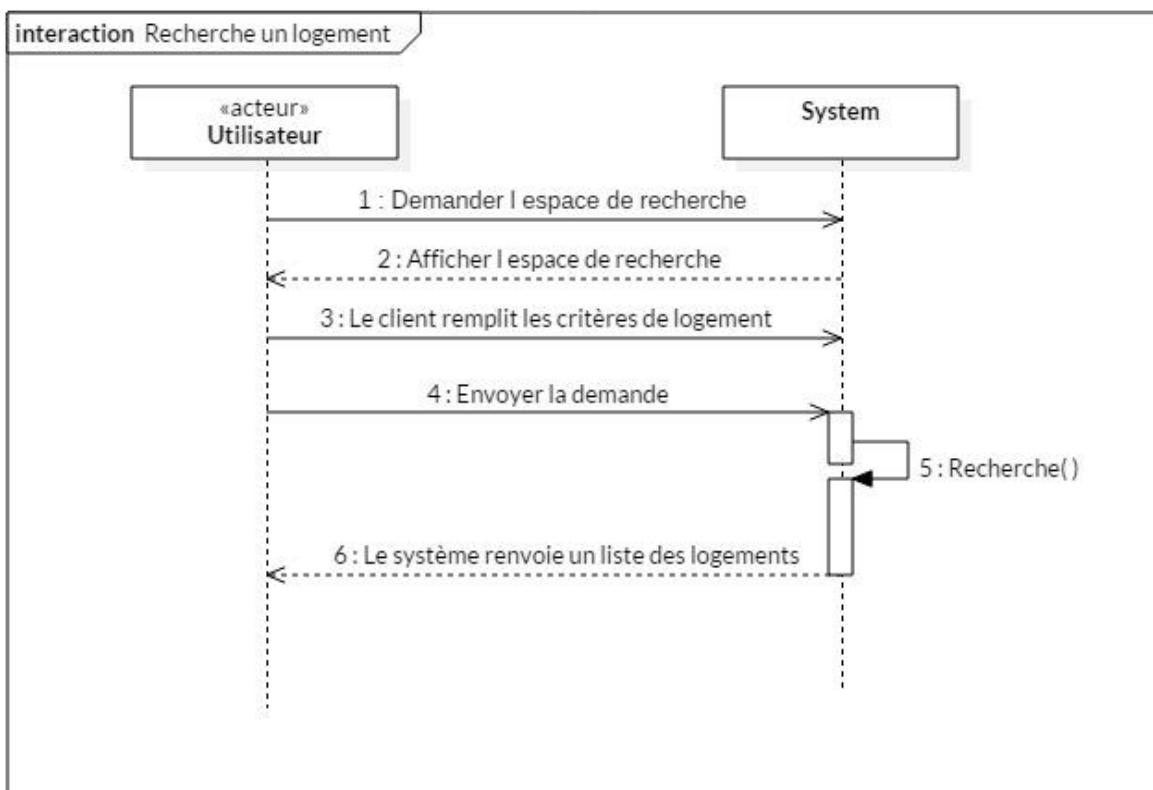


Figure 8 - Diagramme de séquence « Rechercher un logement »

- Cas d'utilisation : Sélectionner un logement

- Fiche descriptive :

Cas d'utilisation	Sélectionner un logement
Acteur	Internaute , Client , Operateur
Type	Primaire
Objectif	Permet au Client de voir les détails d'un logements
Préconditions	Consulter les logements
Post conditions	Afficher les détails du logement choisi.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur demande la liste des logements.</li> <li>2. Le système affiche la liste des logements.</li> <li>3. Le client choisi un logement de la liste.</li> <li>4. Le système affiche les détails du logement choisi.</li> </ol>
Scénario Alternatif	
Scénario Exception	- L'utilisateur Annule la demande.

Diagramme de séquence :

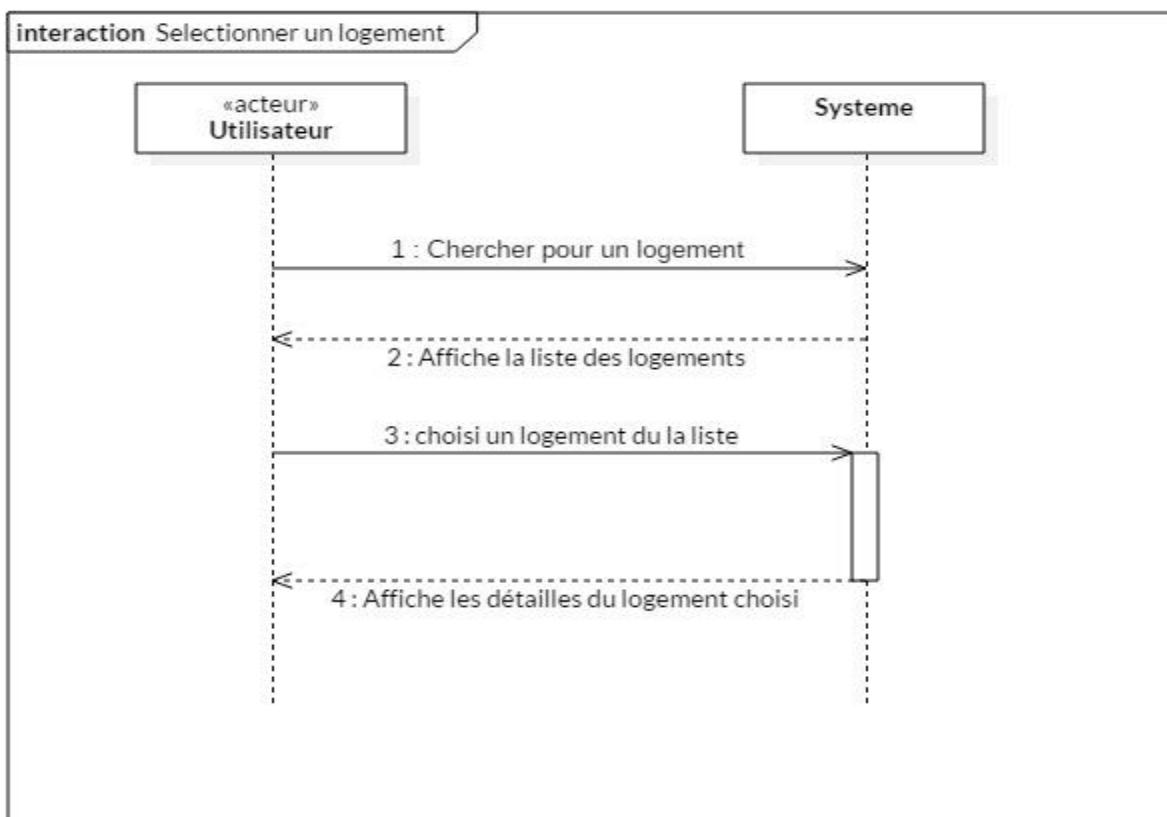


Figure 9 - Diagramme de séquence « Selectionner un logement »

- Cas d'utilisation : Prendre un rendez-vous

- Fiche descriptive :

Cas d'utilisation	Prendre un RDV
Acteur	Client , Operateur
Type	Primaire
Objectif	Permet au Client de prendre un rendez-vous.
Préconditions	<ul style="list-style-type: none"><li>- Sélectionner un logement.</li><li>- S'authentifier.</li></ul>
Post conditions	Afficher les détails d'un rendez-vous.
Scénario Nominal	<ol style="list-style-type: none"><li>1. Le client demande un rendez-vous.</li><li>2. Le système lui génère un rendez-vous.</li><li>3. Le client accepte le rendez-vous.</li></ol>
Scénario Alternatif	
Scénario Exception	<ul style="list-style-type: none"><li>- L'utilisateur Annule l'opération.</li></ul>

- Diagramme de séquence :

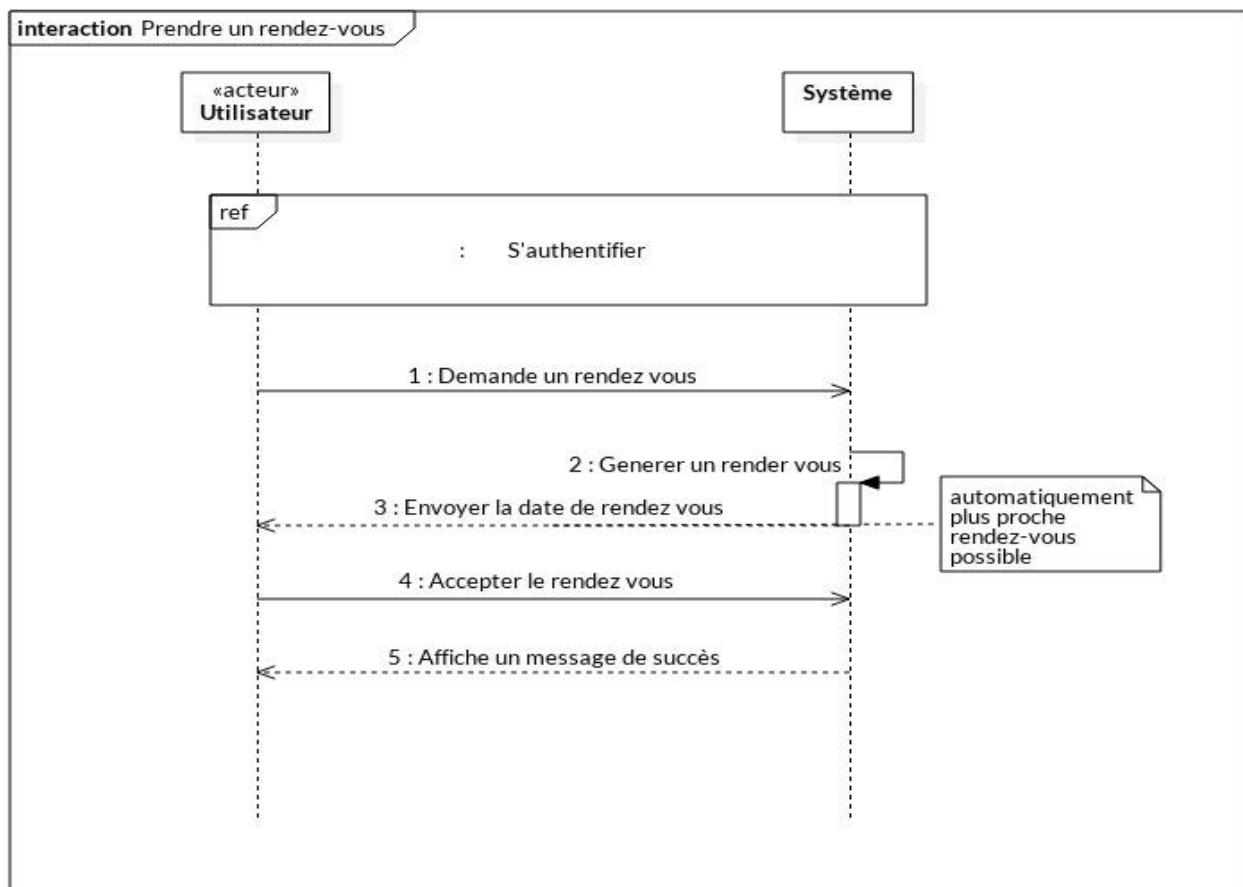


Figure 10 - Diagramme de séquence « Prendre Rendez-vous »

- Cas d'utilisation : Confirmer un rendez-vous

- Fiche descriptive :

Cas d'utilisation	Confirmer rendez-vous
Acteur	Client , Agent
Type	Primaire
Objectif	Permet au utilisateur de valider son présence .
Préconditions	S'authentifier
Post conditions	Le rendez-vous a été confirmé par l'utilisateur
Scénario Nominal	<ol style="list-style-type: none"><li>1. Le système demande aux utilisateurs de confirmer le rendez-vous.</li><li>2. L'utilisateur confirme le rendez-vous.</li><li>3. Le système enregistre l'opération.</li><li>4. Le système informe les utilisateurs concernés.</li></ol>
Scénario Alternatif	
Scénario Exception	- Annulation.

- Diagramme de séquence :

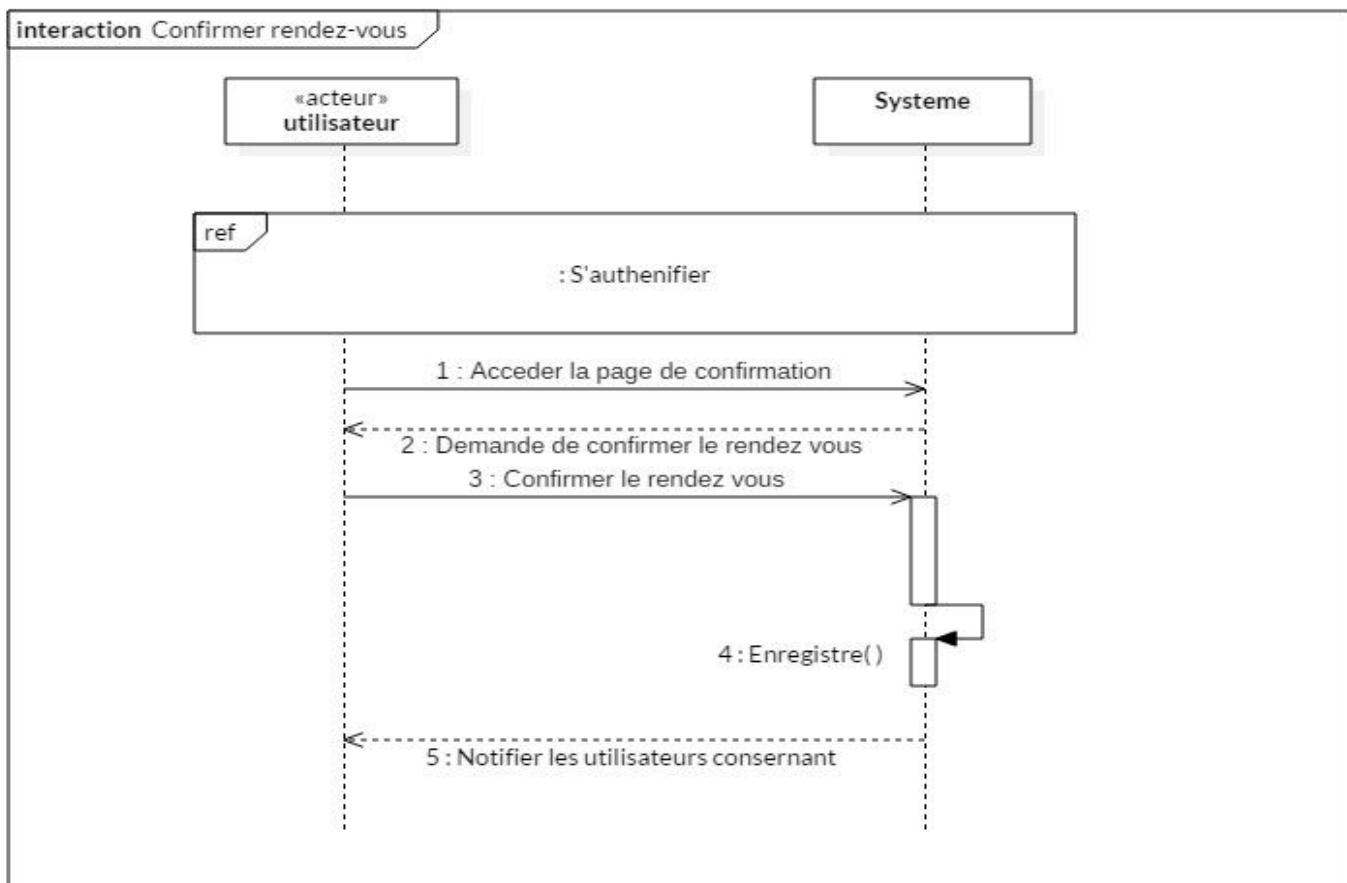


Figure 11 - Diagramme de séquence « Confirmer Rendez-vous »

- Cas d'utilisation : Noter les préavis

- Fiche descriptive :

Cas d'utilisation	Noter les préavis.
Acteur	Agent.
Type	Secondaire.
Objectif	Prendre l'avis du client.
Précondition	S'authentifier.
Post-Condition	Obtient l'avis finale d'un client.
Scenario nominal	<ol style="list-style-type: none"> <li>1. L'agent consulte un RDV.</li> <li>2. L'agent demande de noter un préavis.</li> <li>3. Le système affiche un formulaire du préavis.</li> <li>4. L'agent remplir le formulaire.</li> <li>5. L'agent envoi le formulaire au système.</li> <li>6. Le système enregistre la préavis.</li> </ol>
Scenario Alternatif	<p>2.a.1 Les informations invalide      a.2 Le system renvoi un message de l'erreur      Reprendre à l'étape 2.</p>
Scenario Exception	L'agent annulé l'opération.

- Diagramme de séquence :

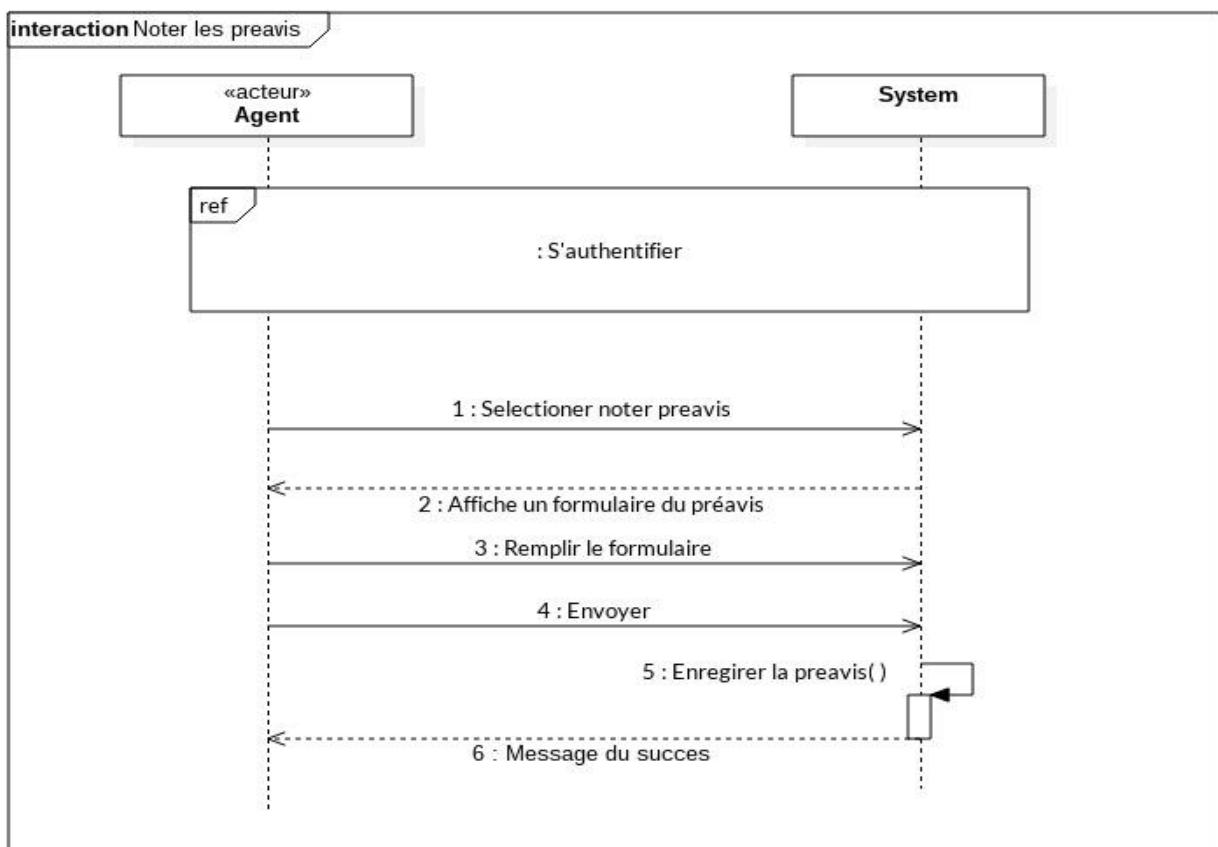


Figure 12 – Diagramme de séquence « Noter préavis »

- Cas d'utilisation : Signaler un rendez vous

- Fiche descriptive :

Cas d'utilisation	Signaler le RDV
Acteur	Client , Opérateur , Agent
Type	Primaire
Objectif	Permet au utilisateur de signaler le RDV sélectionnée.
Préconditions	- S'authentifier.
Post conditions	Envoyer un formulaire de la proclamation d'un RDV.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'Utilisateur consulter un RDV.</li> <li>2. L'Utilisateur demande de signaler le RDV.</li> <li>3. Le système enregistre le signal.</li> <li>4. Le système informe l'opérateur au signal.</li> </ol>
Scénario Alternatif	
Scénario Exception	- L'utilisateur Annule la demande.

- Diagramme de séquence :

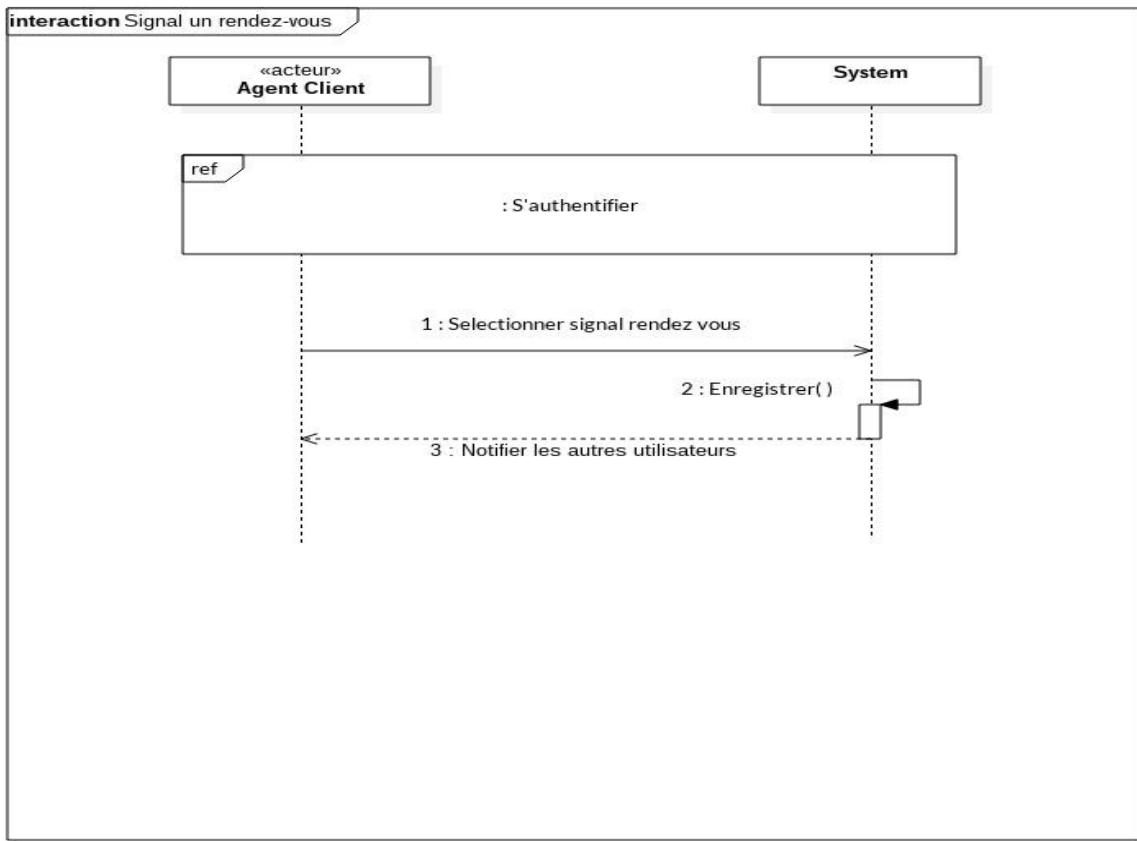


Figure 13 - Diagramme de séquence « Signal Rendez-vous »

- Cas d'utilisation : Annuler un rendez-vous

- Fiche descriptive :

Cas d'utilisation	Annuler le RDV
Acteur	Client , Opérateur
Type	Primaire
Objectif	Permet au utilisateur d'annuler le RDV sélectionnée.
Préconditions	- S'authentifier.
Post conditions	L'annulation d'un RDV.
Scénario Nominal	<ol style="list-style-type: none"><li>1. L'Utilisateur demande d'annuler le rendez-vous.</li><li>2. Le système demande de confirme l'annulation.</li><li>3. L'utilisateur confirme l'annulation du rendez-vous.</li><li>4. Le système valide l'annulation.</li><li>5. Le système notifier les autres utilisateurs concertants.</li></ol>
Scénario Alternatif	
Scénario Exception	- L'utilisateur Annule la demande.

- Diagramme de séquence :

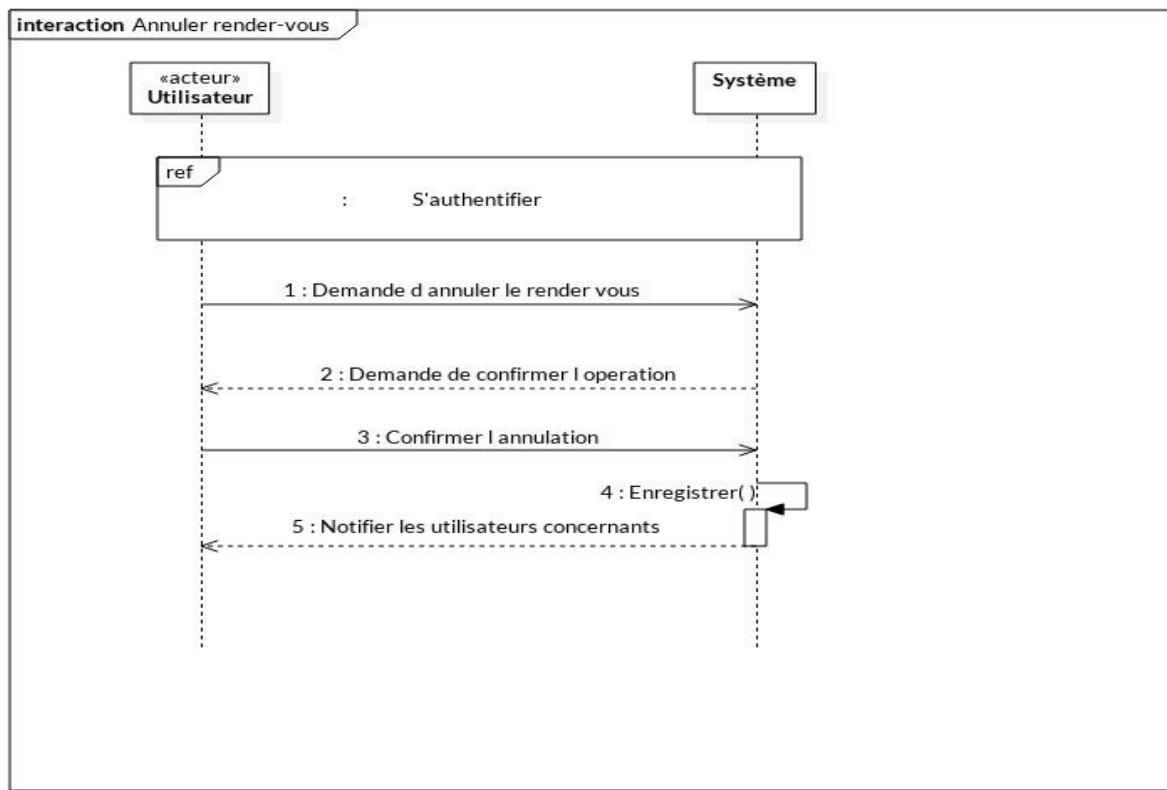


Figure 14 - Diagramme de séquence « Annuler rendez-vous »

- Cas d'utilisation : Créer un compte client

- Fiche descriptive :

Cas d'utilisation	Créer un compte
Acteur	Operateur
Type	Primaire
Objectif	Permet à l'opérateur d'inscrit un nouveau client
Préconditions	<ul style="list-style-type: none"> <li>-S'authentifier.</li> <li>-L'opérateur reçoit l'appelle du client.</li> <li>-Gere les comptes.</li> </ul>
Post conditions	Avait un nouveau compte après la demande d'un client.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. L'opérateur demande de créer un compte.</li> <li>2. Le système affiche un formulaire de la création.</li> <li>3. L'opérateur saisit les informations du compte.</li> <li>4. L'opérateur envoier le formulaire.</li> <li>5. Le système enregistre le nouveau compte.</li> </ol>
Scénario Alternatif	<ol style="list-style-type: none"> <li>3.a Information invalide.             <ol style="list-style-type: none"> <li>a.1. Le système affiche un message d'erreur.</li> </ol> </li> </ol> <p>Reprendre à l'étape 2</p>
Scénario Exception	<ul style="list-style-type: none"> <li>- Annuler l'envoie.</li> <li>- Compte existe déjà.</li> </ul>

- Diagramme de séquence :

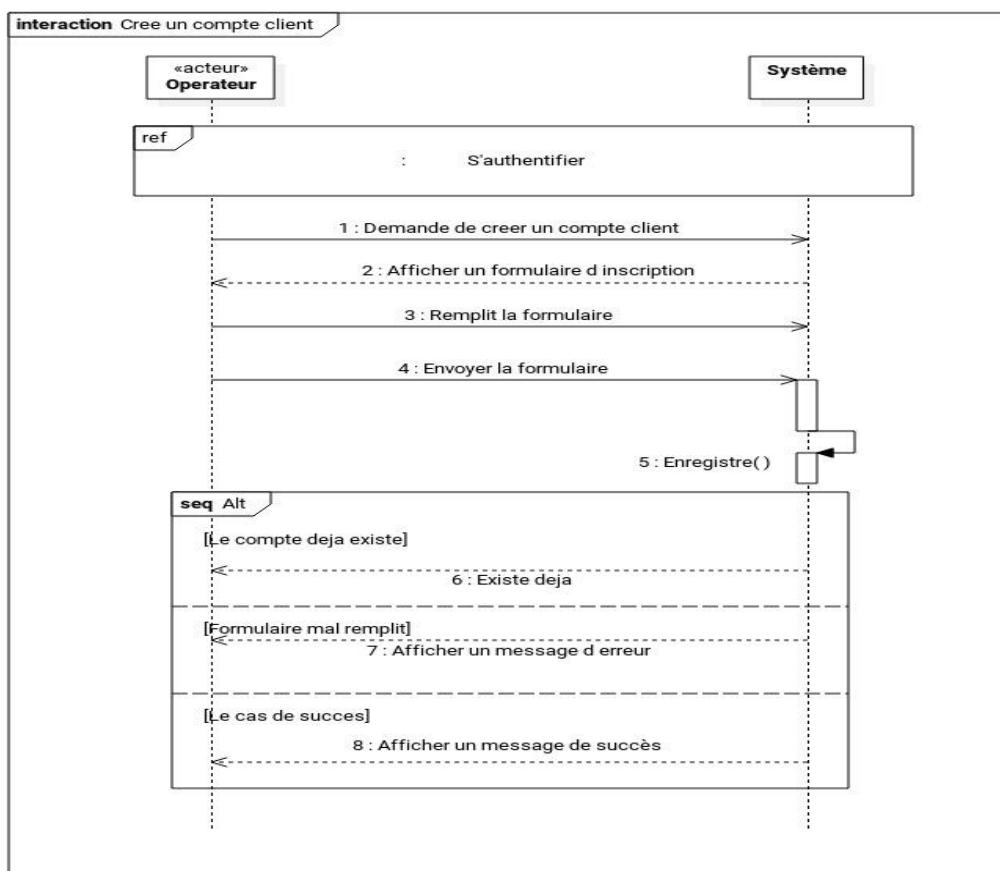


Figure 15 - Diagramme de séquence « Cree compte »

- Cas d'utilisation : Accepter l'inscription

- Fiche descriptive :

Cas d'utilisation	Accepter les inscriptions
Acteur	Administrateur
Type	Secondaire
Objectif	Valider l'inscriptions des opérateur et agents
Précondition	<ul style="list-style-type: none"> <li>- S'authentifier</li> <li>- Gérer les comptes des employés</li> </ul>
Post-Condition	Vérifier tous les nouveau comptes des employées dans le système.
Scenario nominal	<ol style="list-style-type: none"> <li>1. L'utilisateur demande de valider les inscriptions.</li> <li>2. Le system affiche la liste des comptes qui non pas encore valide.</li> <li>3. L'admin sélectionne un compte de la liste.</li> <li>4. Le système affiche les détails de compte.</li> <li>5. L'admin valide l'inscription de cet utilisateur.</li> <li>6. Le système affiche un message de succès.</li> </ol>
Scenario Alternatif	
Scenario Exception	- Annuler l'envoie.

- Diagramme de séquence :

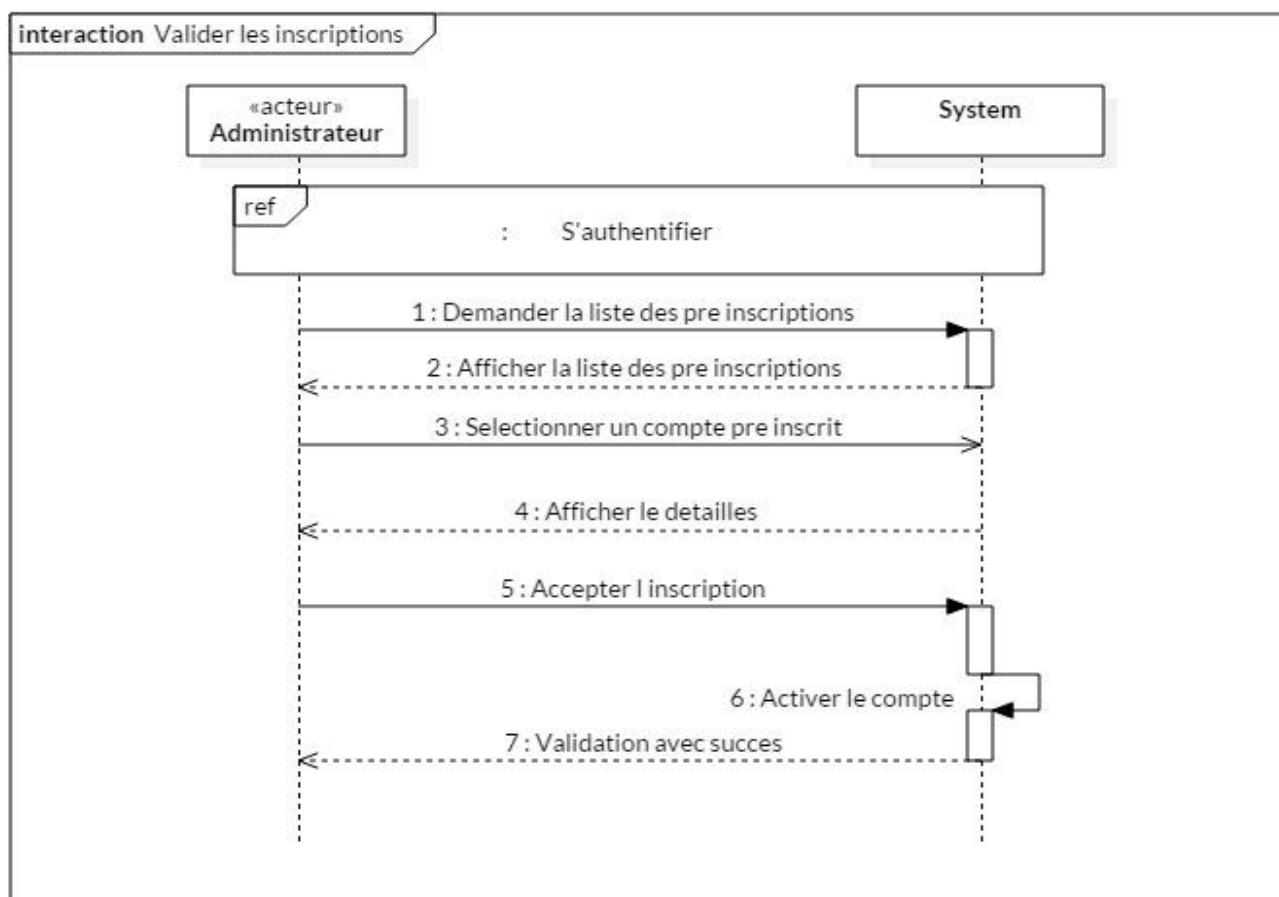


Figure 16 - Diagramme de séquence « Valider les inscriptions »

- Cas d'utilisation : Consulter les statistiques

- Fiche descriptive :

Cas d'utilisation	Consulter les statistique
Acteur	Administrateur
Type	Secondaire
Objectif	Permet à l'admin de voir tous les statistique des logements et employées
Préconditions	S'authentifie
Post conditions	Afficher les statistique de l'agence.
Scénario Nominal	<ol style="list-style-type: none"> <li>1. Le système demande une date de début et fin.</li> <li>2. L'administrateur remplit les dates.</li> <li>3. Le système renvoie la liste des statistique dans cette période.</li> </ol>
Scénario Alternatif	<ol style="list-style-type: none"> <li>4.a Date invalide.             <ol style="list-style-type: none"> <li>a.1. Le système affiche un message d'erreur.</li> </ol> </li> </ol> <p>Reprendre à l'étape 2</p>
Scénario Exception	- L'utilisateur Annule la demande.

- Diagramme de séquence :

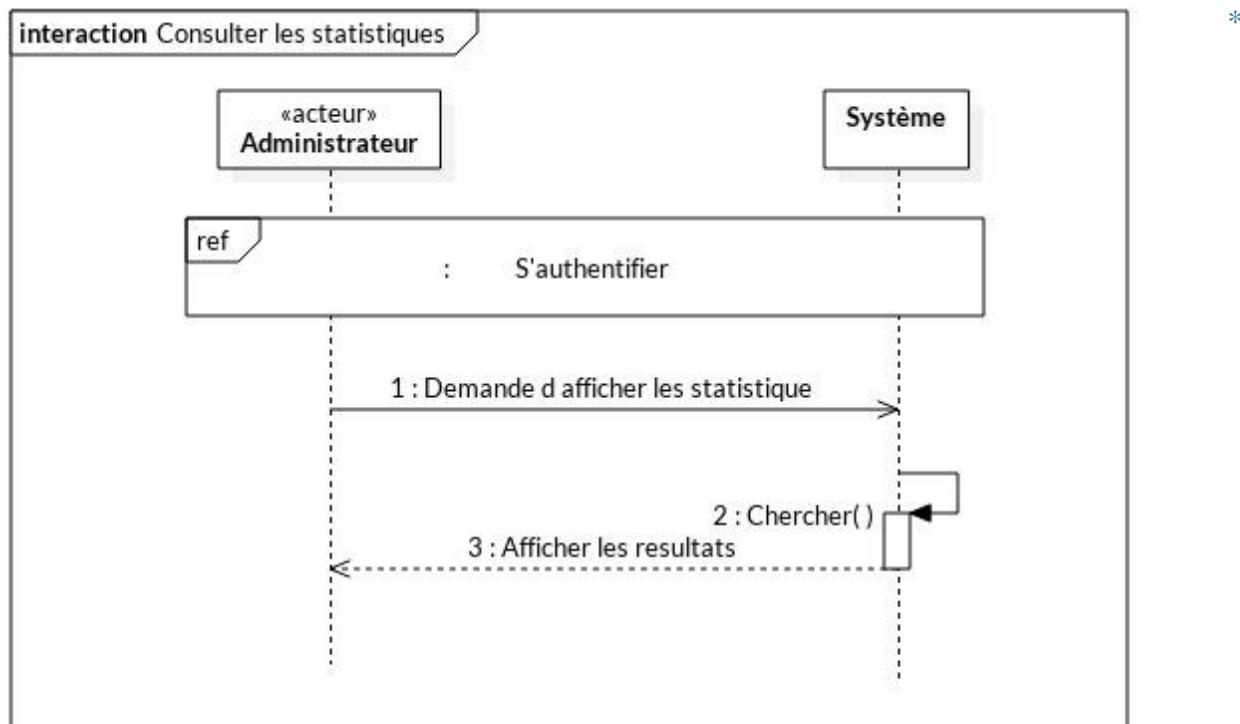


Figure 17 - Diagramme de séquence « Consulter les statistiques »

- Cas d'utilisation : Ajouter un logement

- Fiche descriptive :

Cas d'utilisation	Ajouter un logement
Acteur	Administrateur BD
Type	Secondaire
Objectif	Ajouter un logements au base de données.
Précondition	- S'authentifier
Post-Condition	un nouveau logements est ajoutée dans le base de données.
Scenario nominal	<ol style="list-style-type: none"> <li>1. Le system affiche une formulaire d'ajout.</li> <li>2. L'admin remplit le formulaire et envoier là.</li> <li>3. Le système enregistre le logement.</li> <li>5. Le système affiche un message de succès.</li> </ol>
Scenario Alternatif	
Scenario Exception	- Annuler l'envoie.

- Diagramme de séquence :

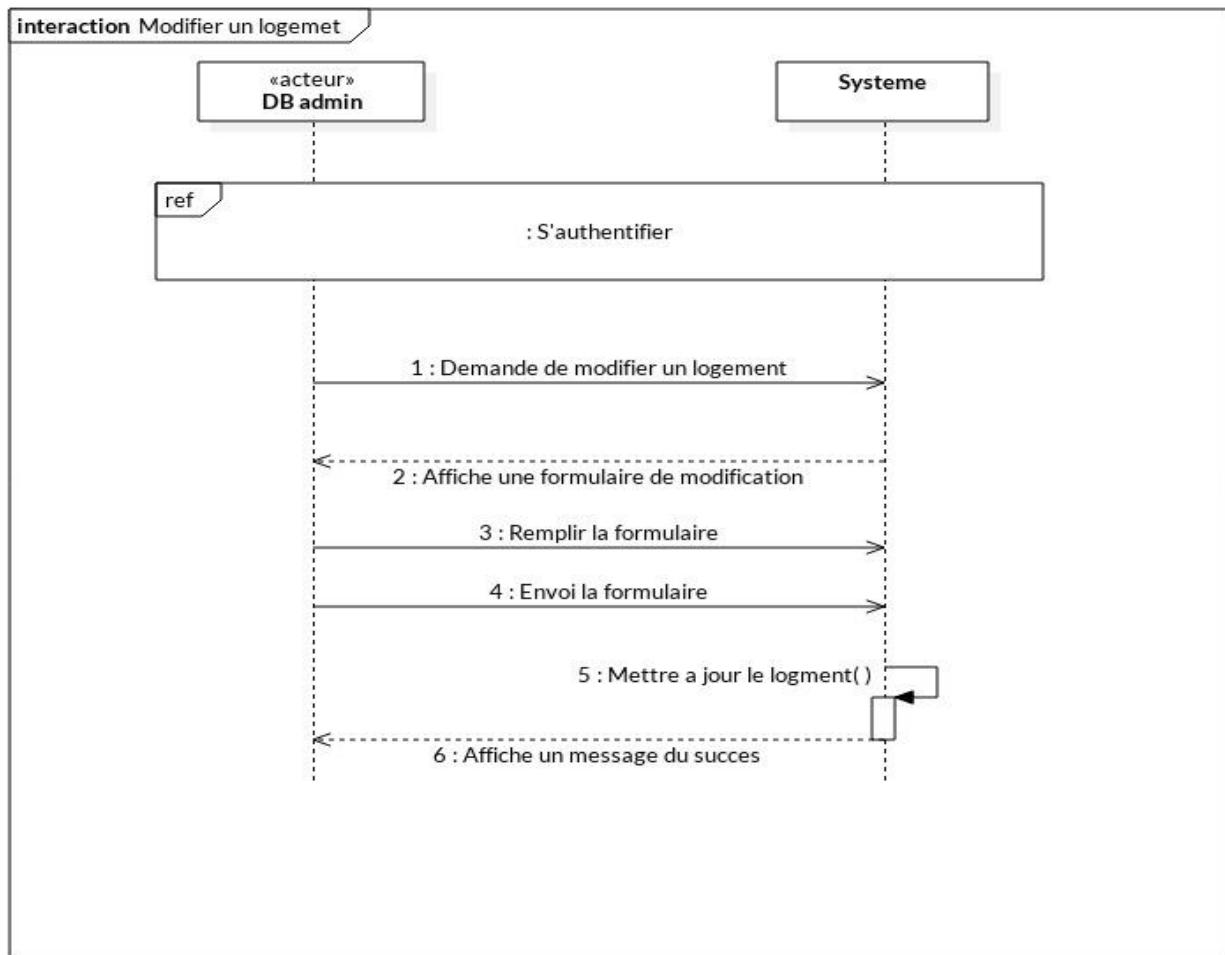


Figure 18 - Diagramme de séquence « Modifier un logement »

- Cas d'utilisation : Modifier un logement

- Fiche descriptive :

Cas d'utilisation	Modifier un logement
Acteur	Administrateur BD
Type	Secondaire
Objectif	Permet de modifier des critères d'un logements dans base de données.
Précondition	- S'authentifier
Post-Condition	un logement a est modifier dans le base de données.
Scenario nominal	<ol style="list-style-type: none"> <li>1. Le system affiche un formulaire de modification.</li> <li>2. L'admin remplit le formulaire et envoier là.</li> <li>3. Le système enregistre le logement.</li> <li>5. Le système affiche un message de succès.</li> </ol>
Scenario Alternatif	
Scenario Exception	- Annuler l'envoie.

- Diagramme de séquence :

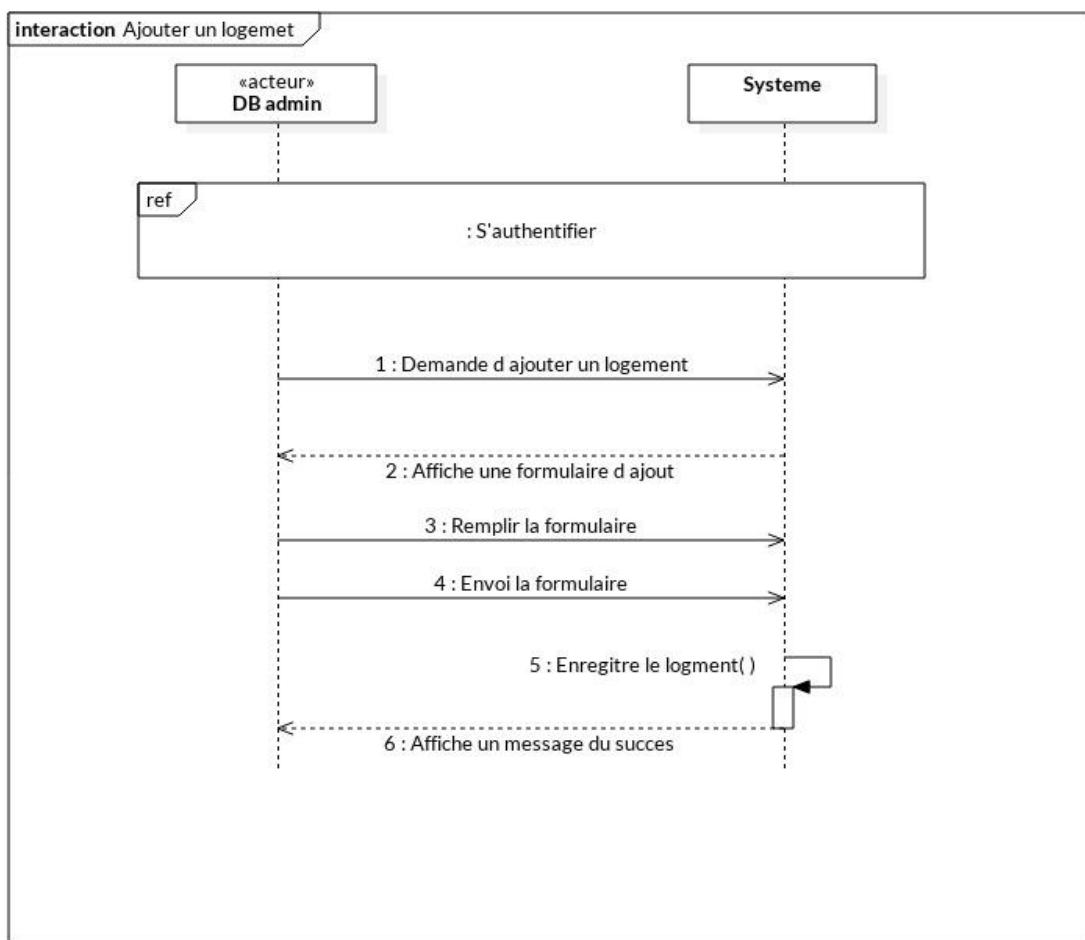


Figure 19 - Diagramme de séquence « Ajouter un logement »

## 5- Les Maquettes

Ici présent vous verrez quelque maquette d'écran prise dans le progiciel du Système d'Information Hospitalière et de la Gestion du centre de santé de.

### - L'inscription :

The wireframe shows a web browser window for 'www.real-estate.com'. The header includes a back/forward button, a search bar, and a refresh button. The menu bar has links for Home, Services, Our Agents, Contact us, and About, along with Sign-in and Sign-up buttons. The main content area features a city skyline background. On the right, there's a logo with a '#' symbol followed by 'Real Estate'. The form fields include Email, Password, Re-Pass, Name, Fam Name, and Phone, each with an associated input field. A checkbox for 'Agree Policy' is present, and a 'Subscribe' button is at the bottom.

Figure 21 - Incrire Wireframe

### L'authentification :

The wireframe shows a web browser window for 'www.real-estate.com'. The header includes a back/forward button, a search bar, and a refresh button. The menu bar has links for Home, Services, Our Agents, Contact us, and About, along with Sign-in and Sign-up buttons. The main content area features a city skyline background. On the right, there's a logo with a '#' symbol followed by 'Real Estate'. The form fields include Email and Password, each with an associated input field. A checkbox for 'Remeber Me' is present, and a 'Login' button is at the bottom.

Figure 20 - Authentification Wireframe

## - La Recherche :

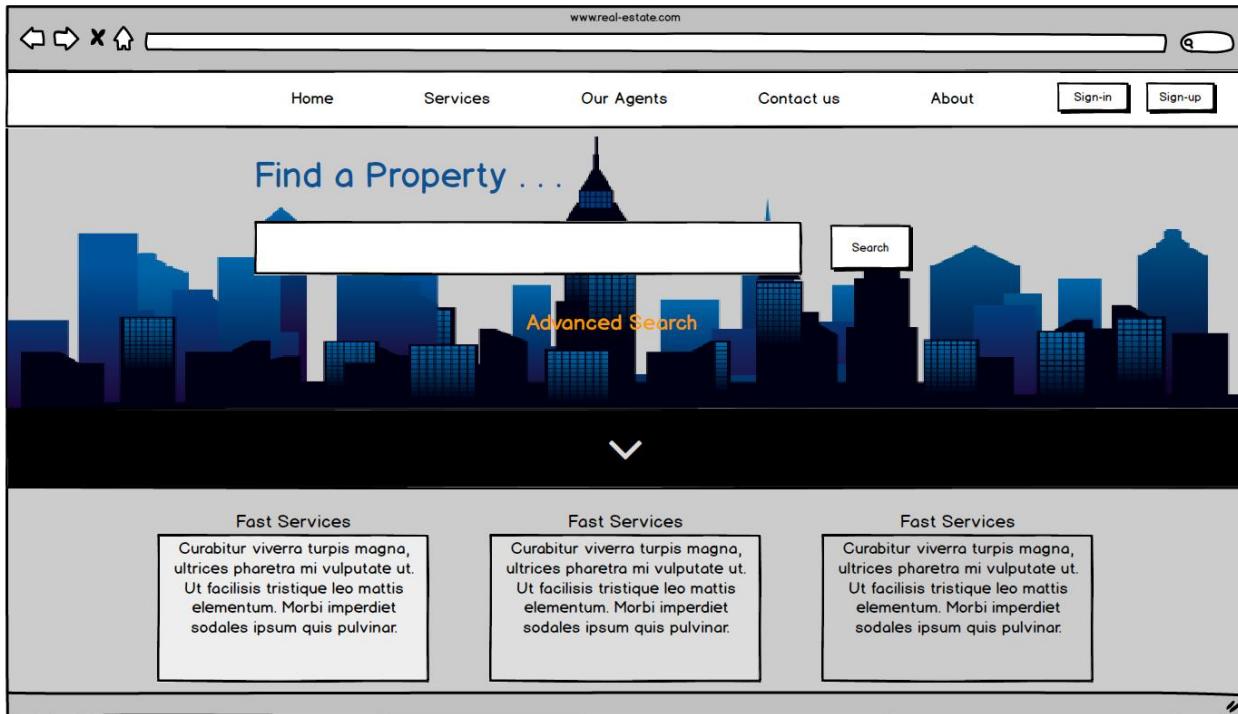


Figure 22 -Rechercher un logement Wireframe

## - Les Résultats :

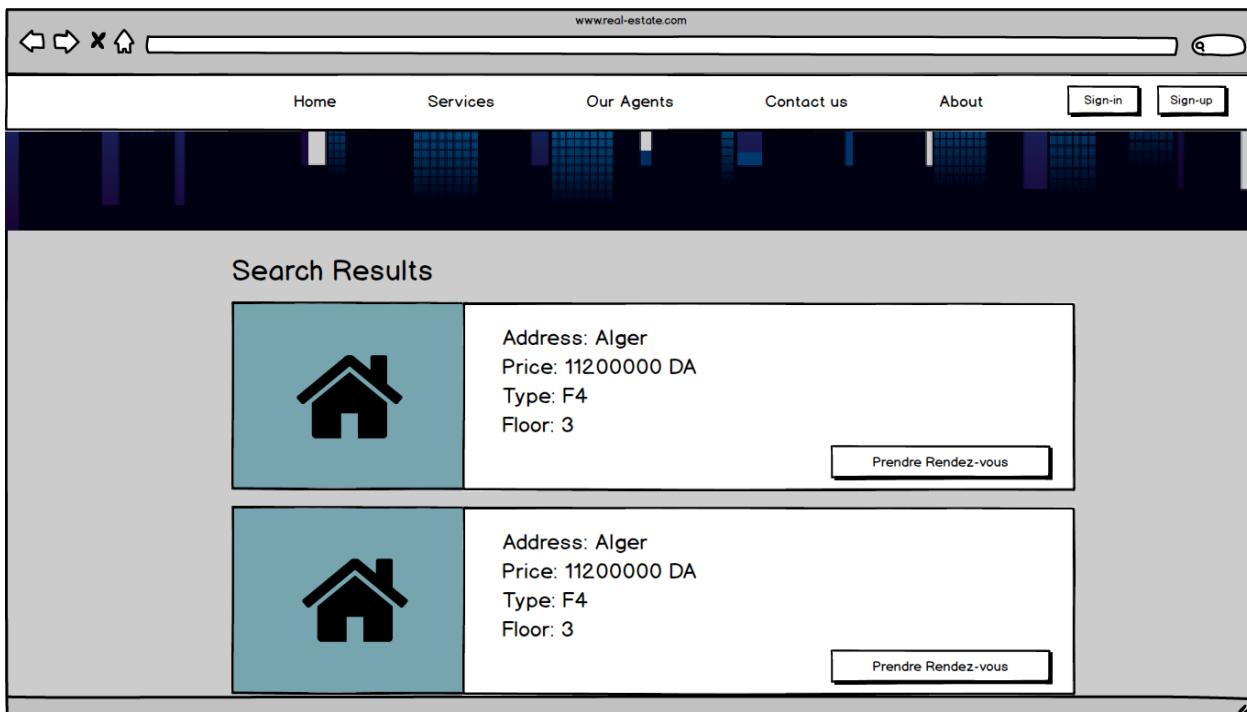
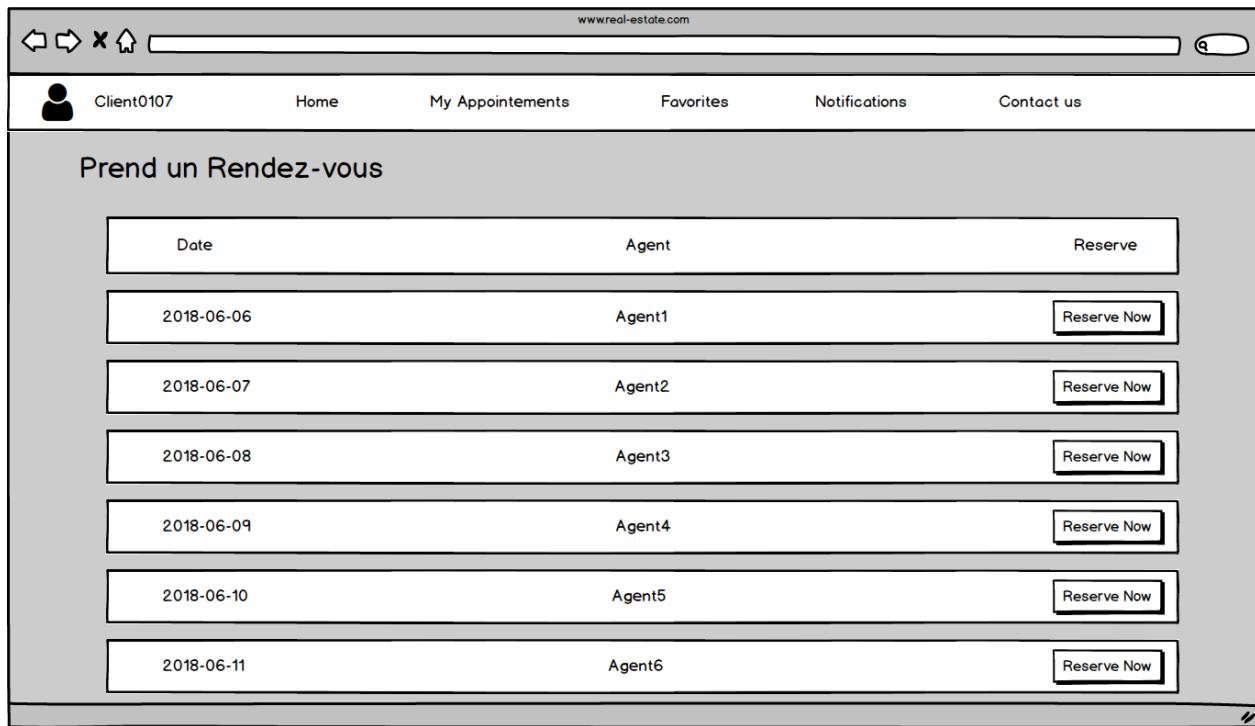


Figure 23 - Resultat de recherche

- Prendre Rendez-vous :



## 6- Conclusion

Ce chapitre a été consacré à la présentation de notre conception en appliquant la démarche Pasqual Roques et les fiches descriptives avec les différents diagrammes UML (DCU, DSE) avec les différents maquettes ces derniers représentent une vue nécessaire pour notre projet, de plus les représentations graphiques nous permettront d'entamer l'implémentation de notre système, que nous allons voir dans le chapitre suivant.

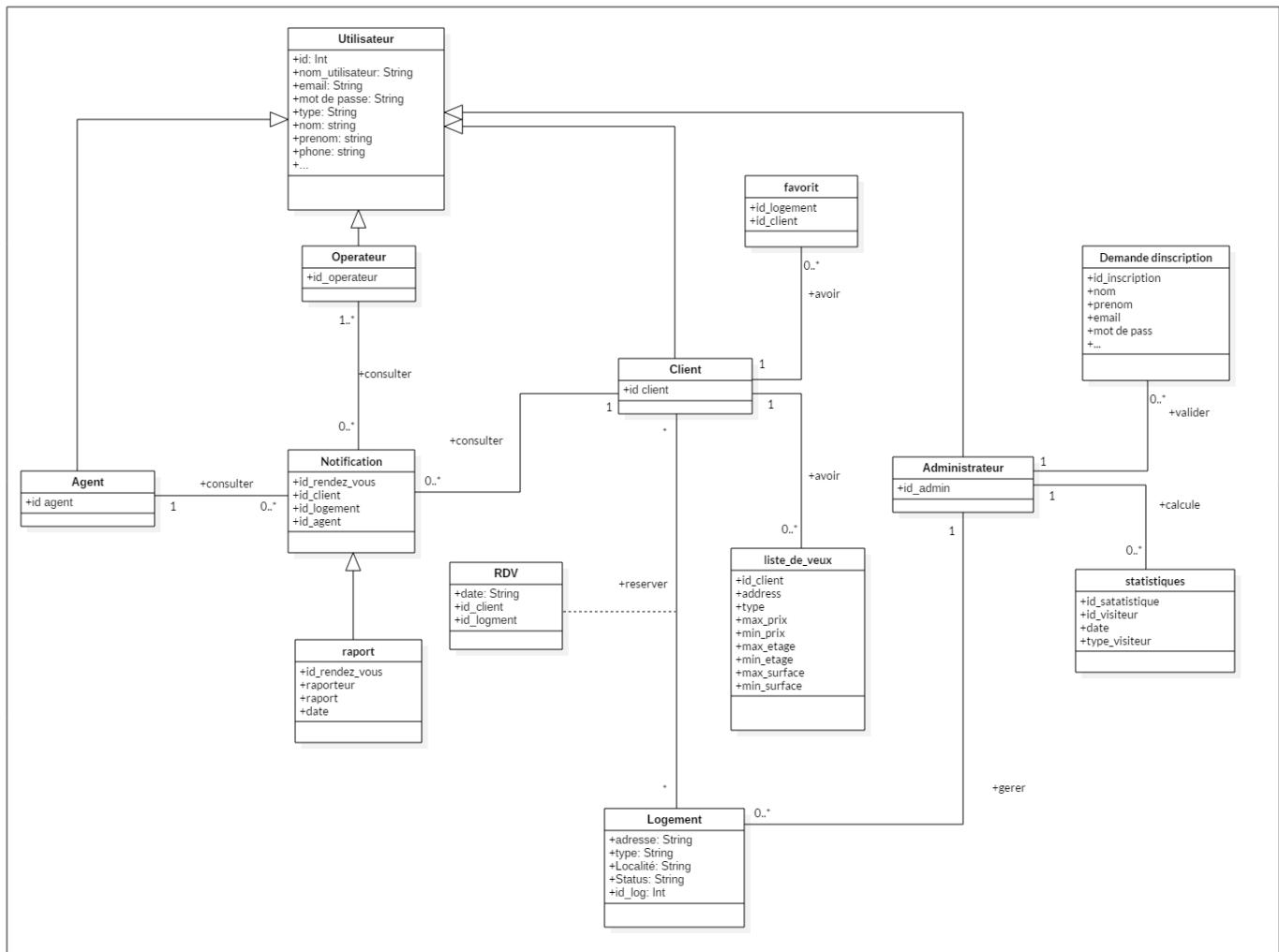
# Chapitre 2 Analyse et conception

## 1- Introduction :

La réalisation d'un site dynamique (Gestion des ventes pour une société immobilière) doit être impérativement précédée d'une méthodologie de conception permettant d'offrir une vision claire des différents éléments de l'application et de l'interaction entre ces derniers. Pour notre conception, nous allons utiliser Pasqual Roques qui est un processus de développement.

## 2- Modèle de domaine :

Le modèle du domaine issu de notre analyse comprend différentes classes et associations comme le suivant :



### 3- Les Diagrammes des classes participantes :

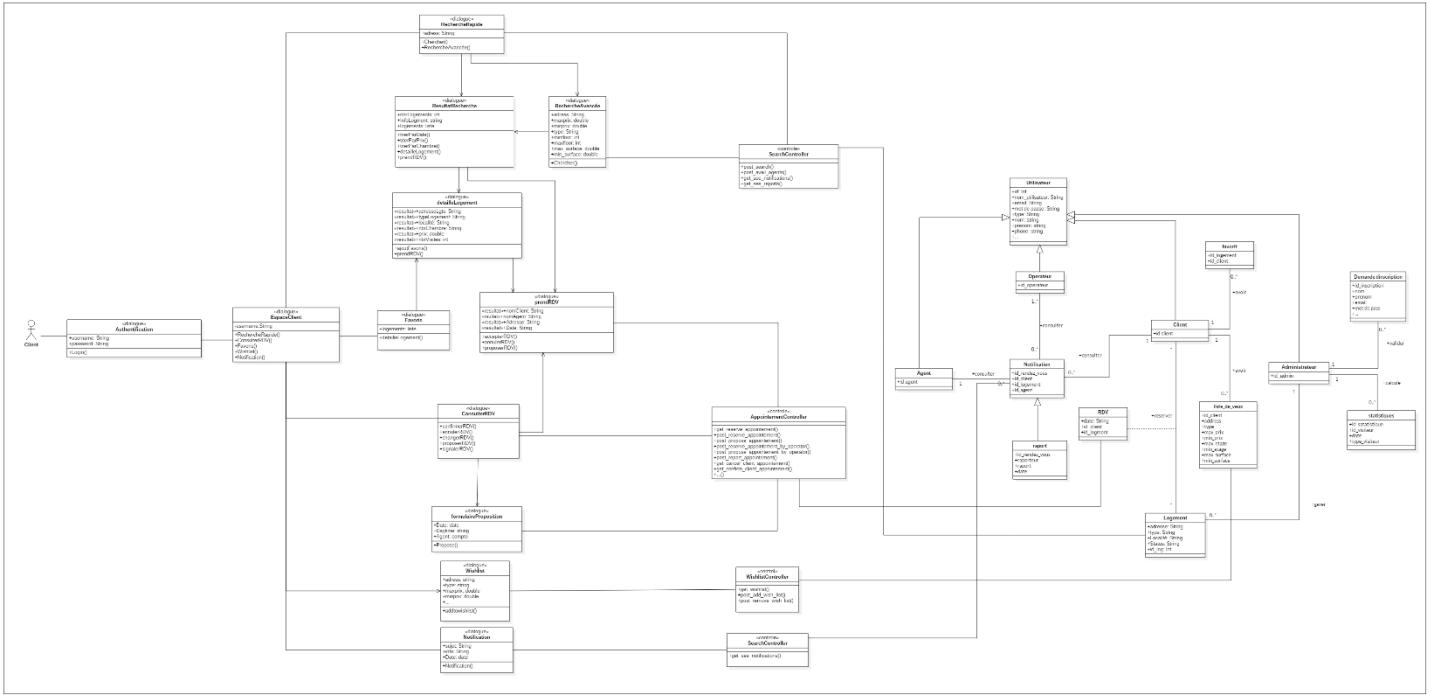
Le diagramme de classes participantes est un diagramme de classes UML qui décrit cas d'utilisation par cas, les trois principales classes d'analyse et leurs relations :

- **Les classes dialogues** possèdent des attributs et des opérations. Les attributs représentent des champs de saisie ou des résultats. Les opérations elles, représentent des actions de l'utilisation sur l'Interface Homme Machine (IHM).

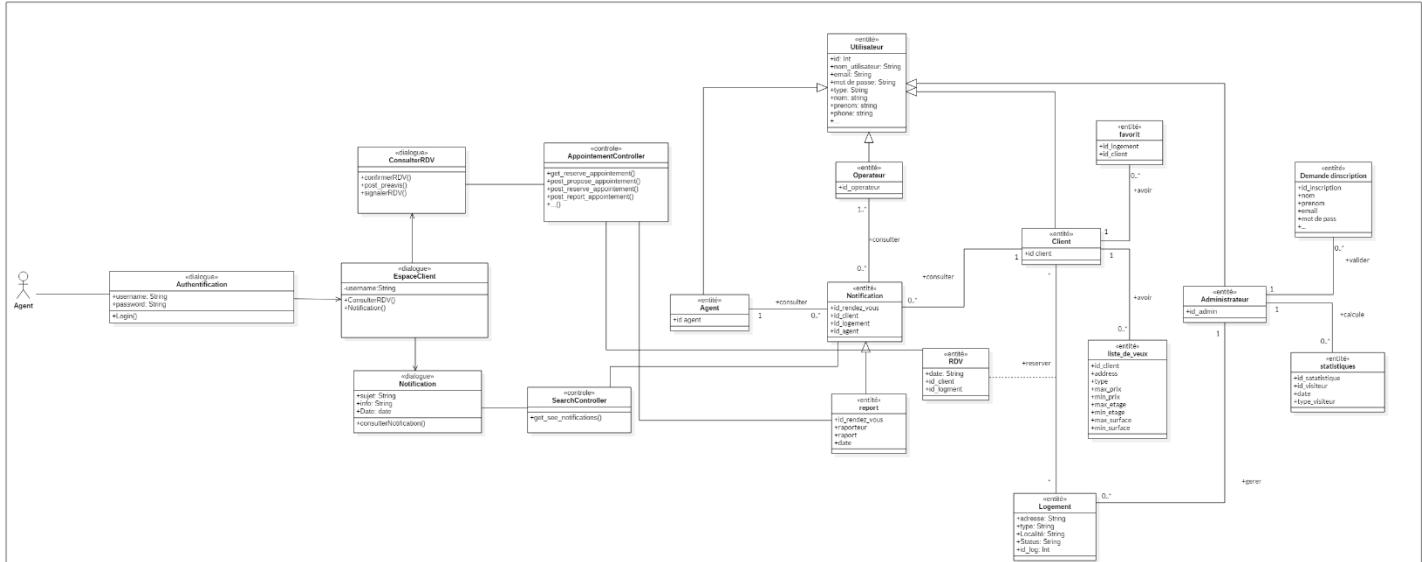
- **Les classes contrôles** contiennent des opérations représentant la logique applicative de l'application, les règles métiers ou les comportements du système informatique.

- **Les classes entités** possèdent en général des informations persistantes de l'application.

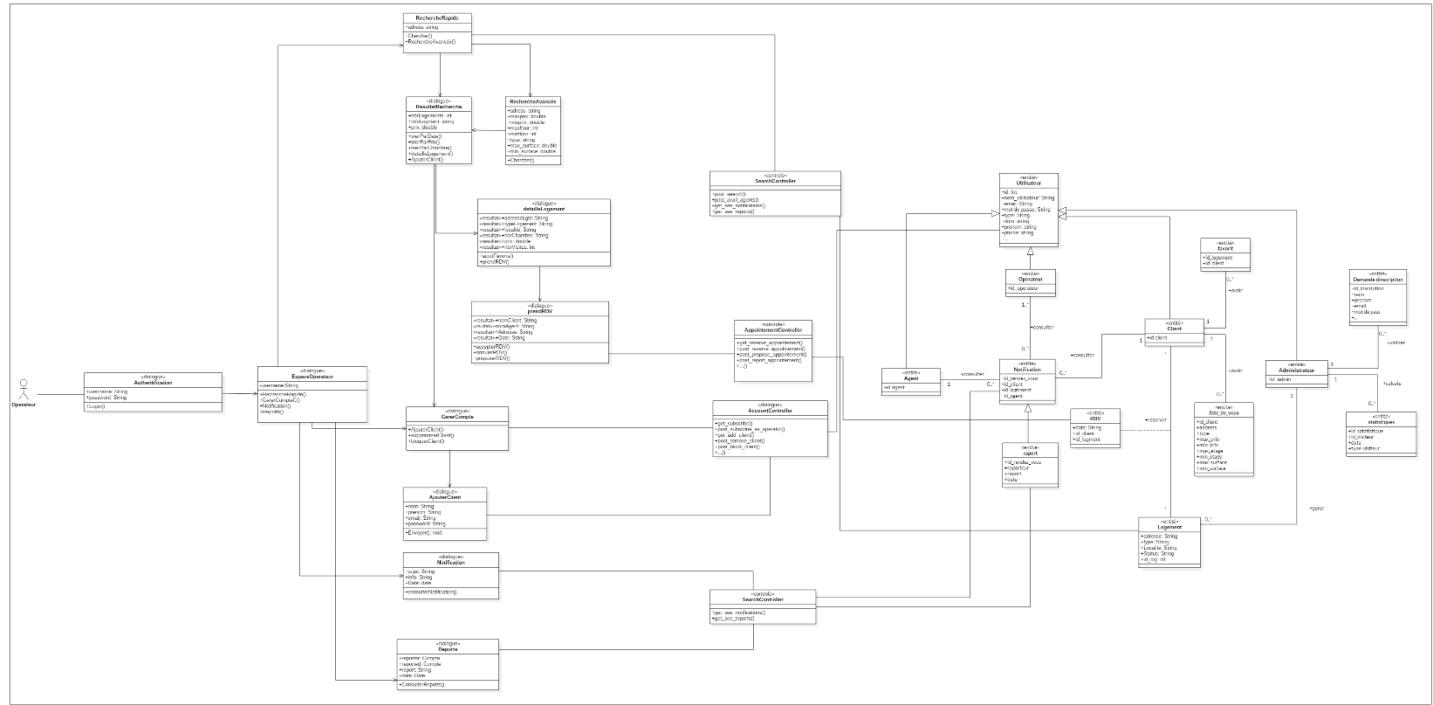
## 2-1 Client :



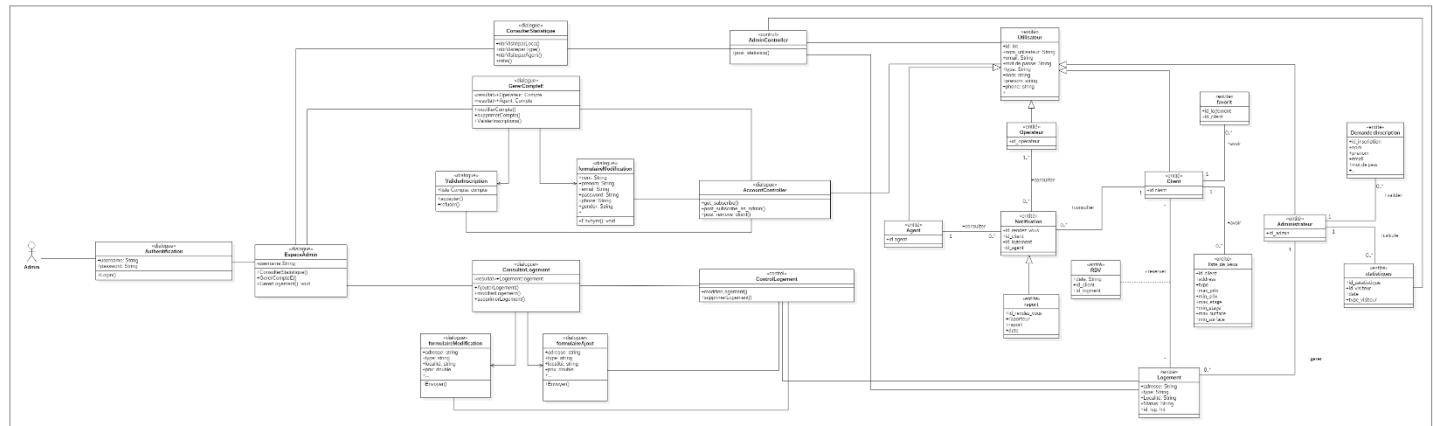
## 2.2- Agent :



## 2.3- Operateur :



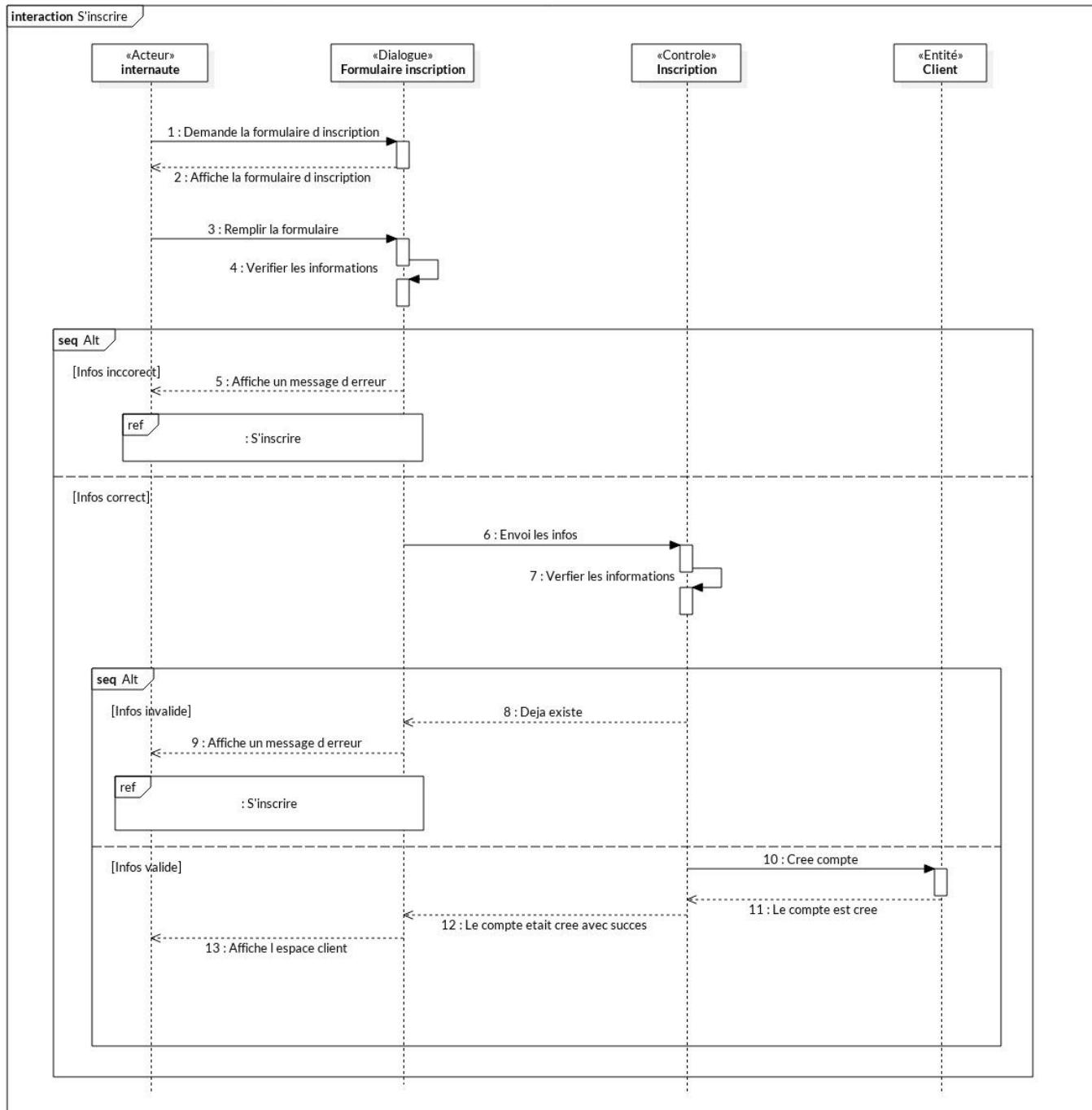
## 2.4- Administrateur :



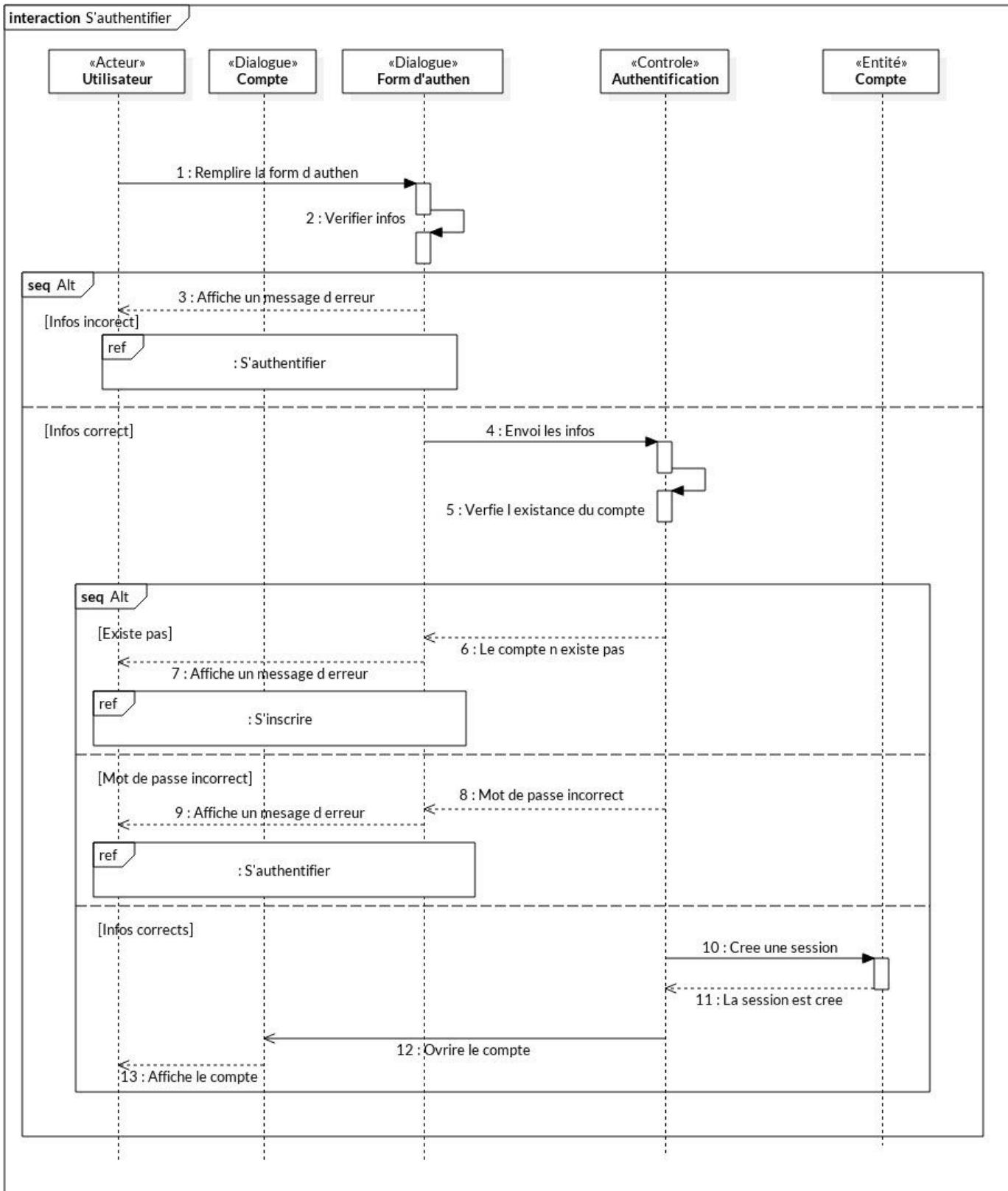
## 4- Les Diagrammes de séquence :

Un diagramme de séquence décrit les actions élémentaires en interactions pour chaque cas d'utilisation. En effet, il élucide les cas d'utilisation un par un en détaillant chaque opération qui en découle ; ainsi il est plus facile de comprendre chaque cas d'utilisation.

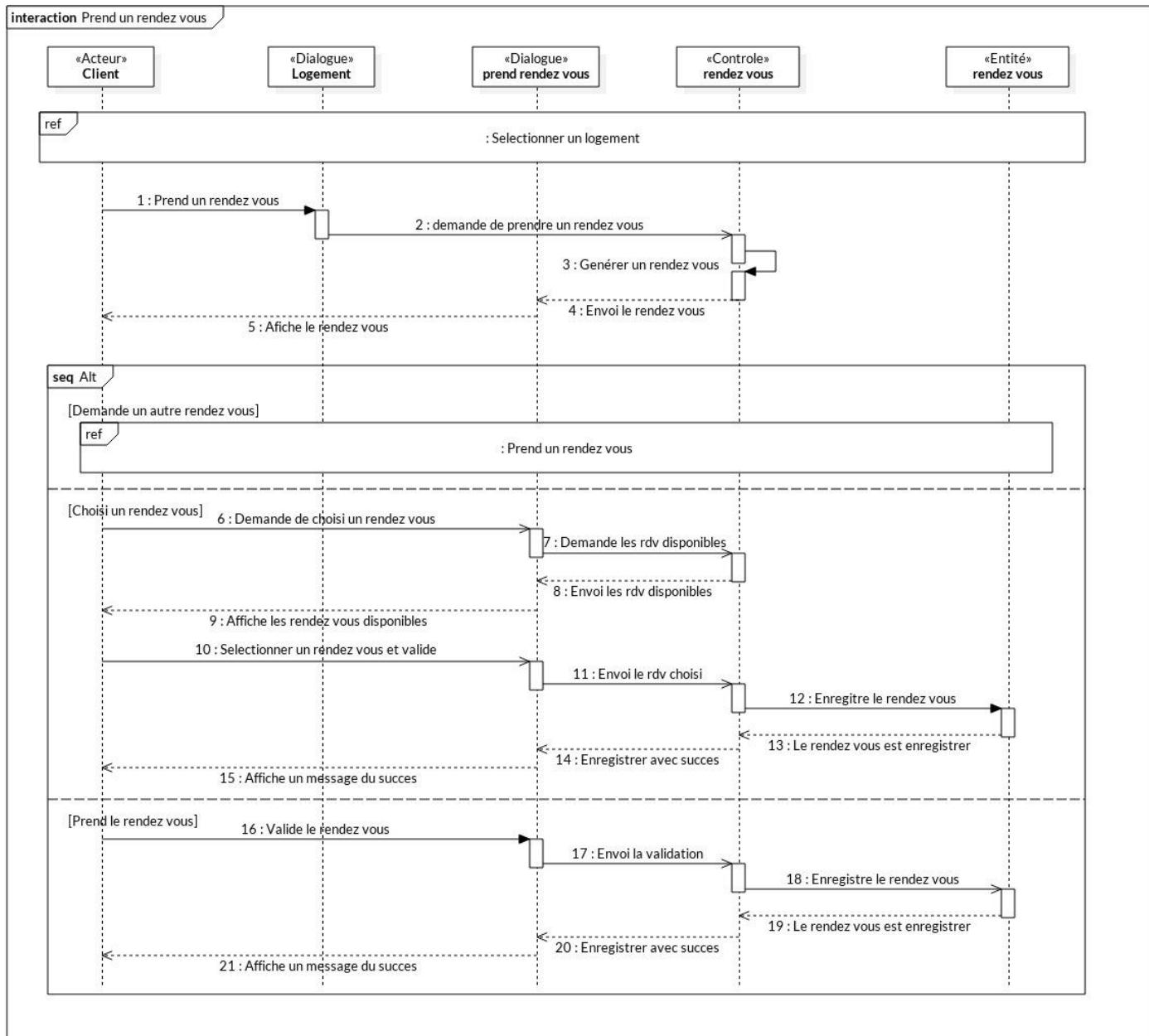
### 3.1- Incrire :



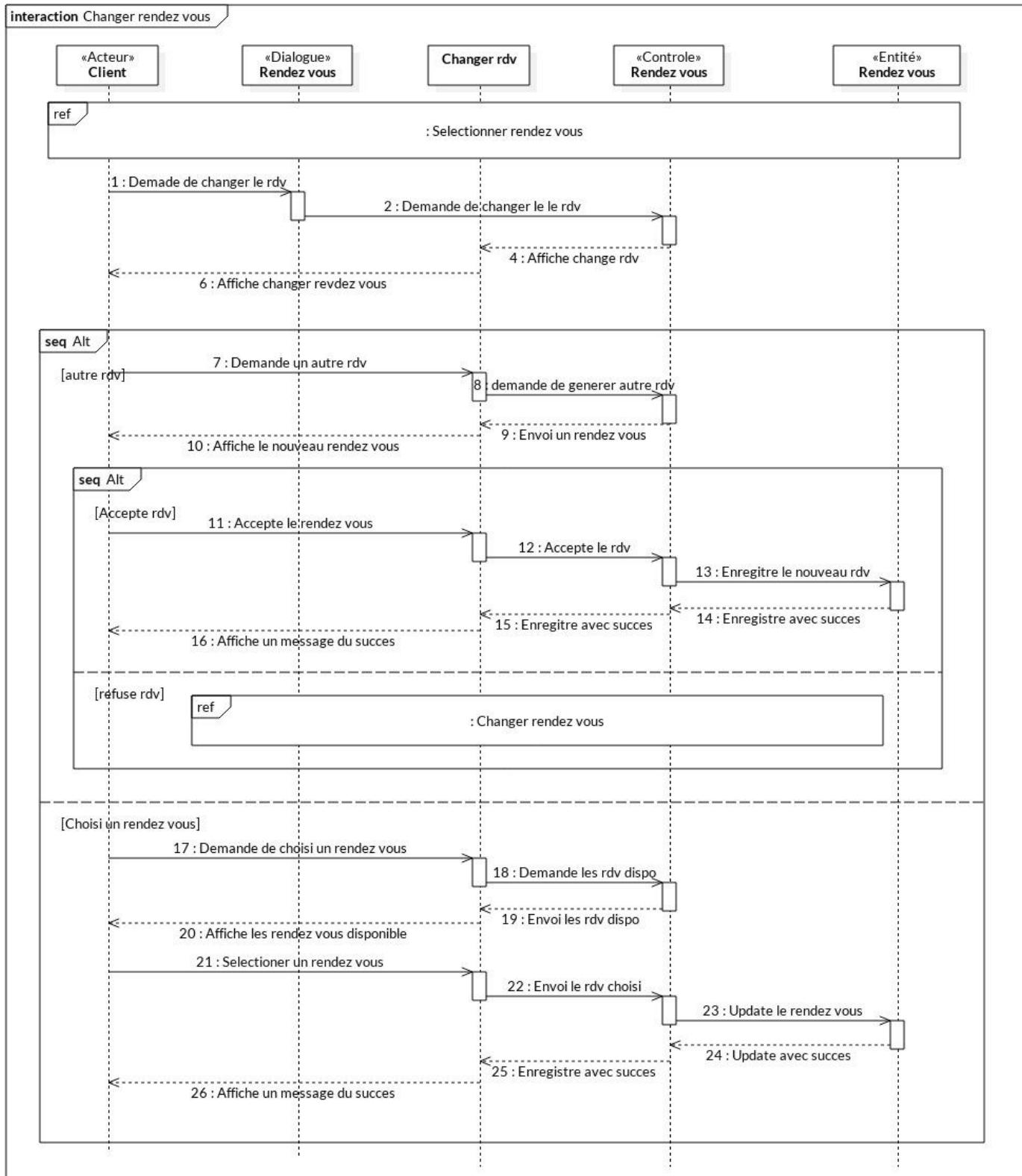
### 3.2-Authentification :



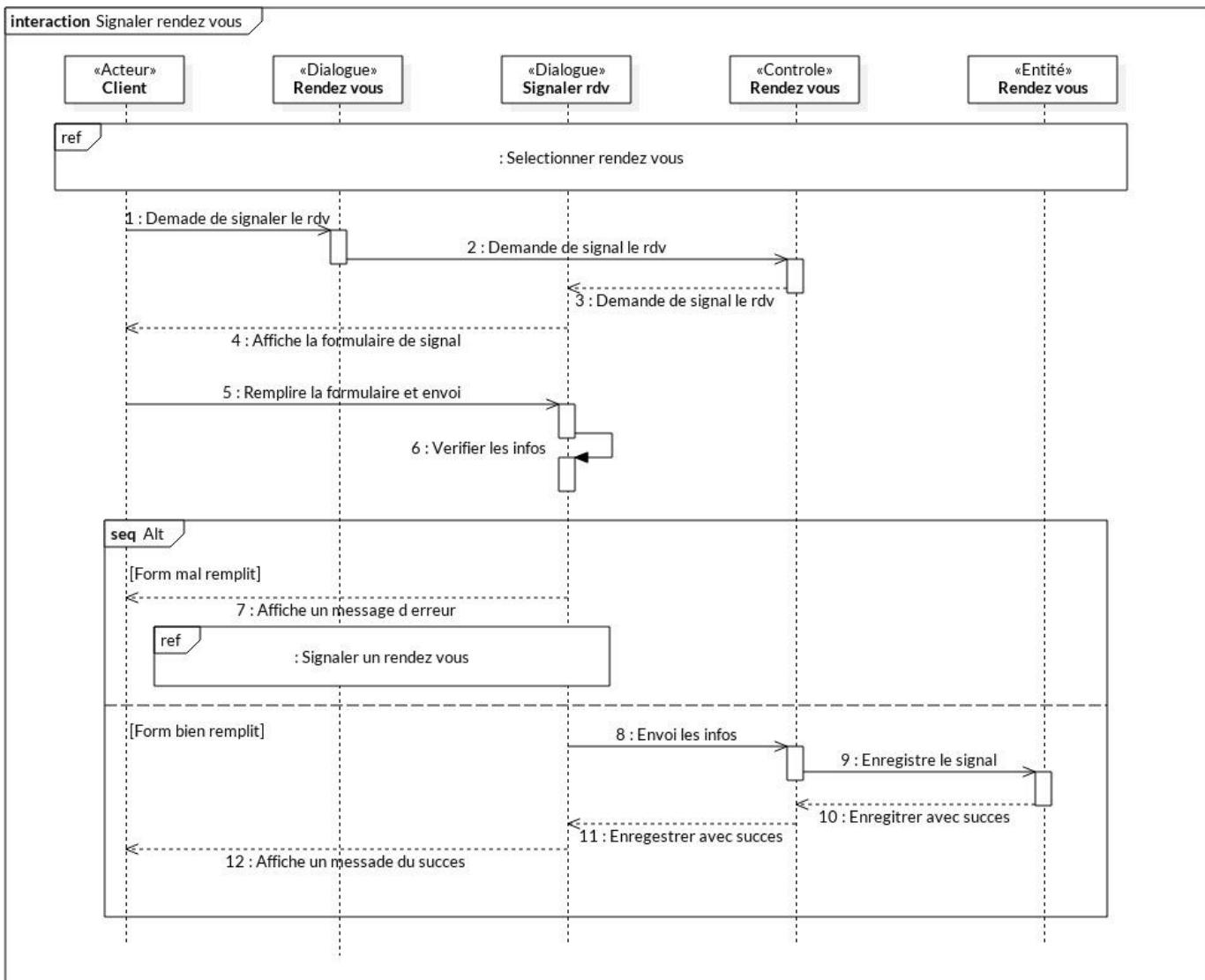
### 3.3- Prend rendez-vous :



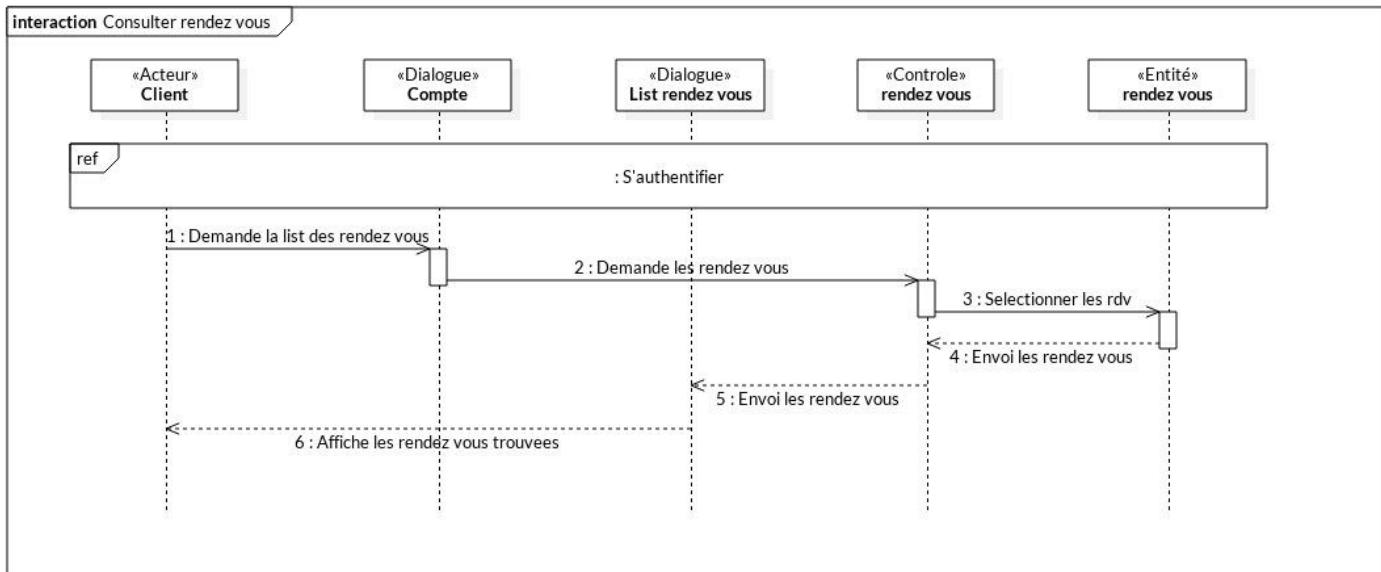
### 3.4-Changer rendez-vous



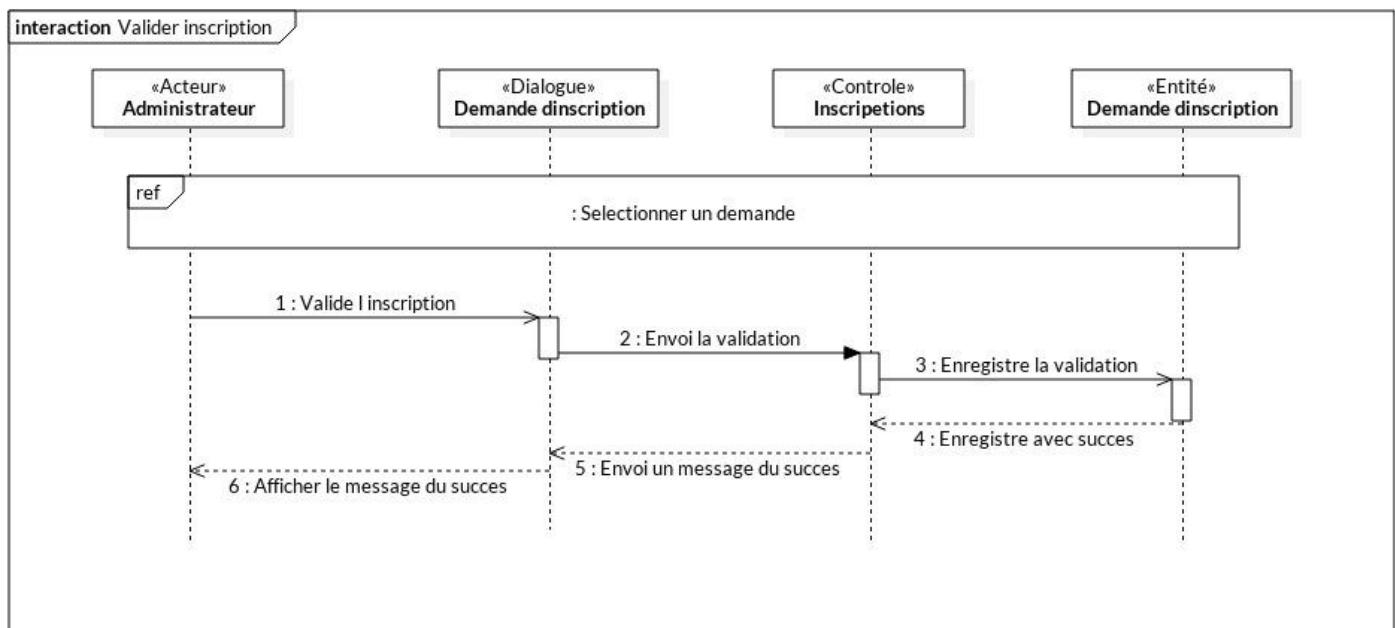
### 3.5-Singaler rendez-vous :



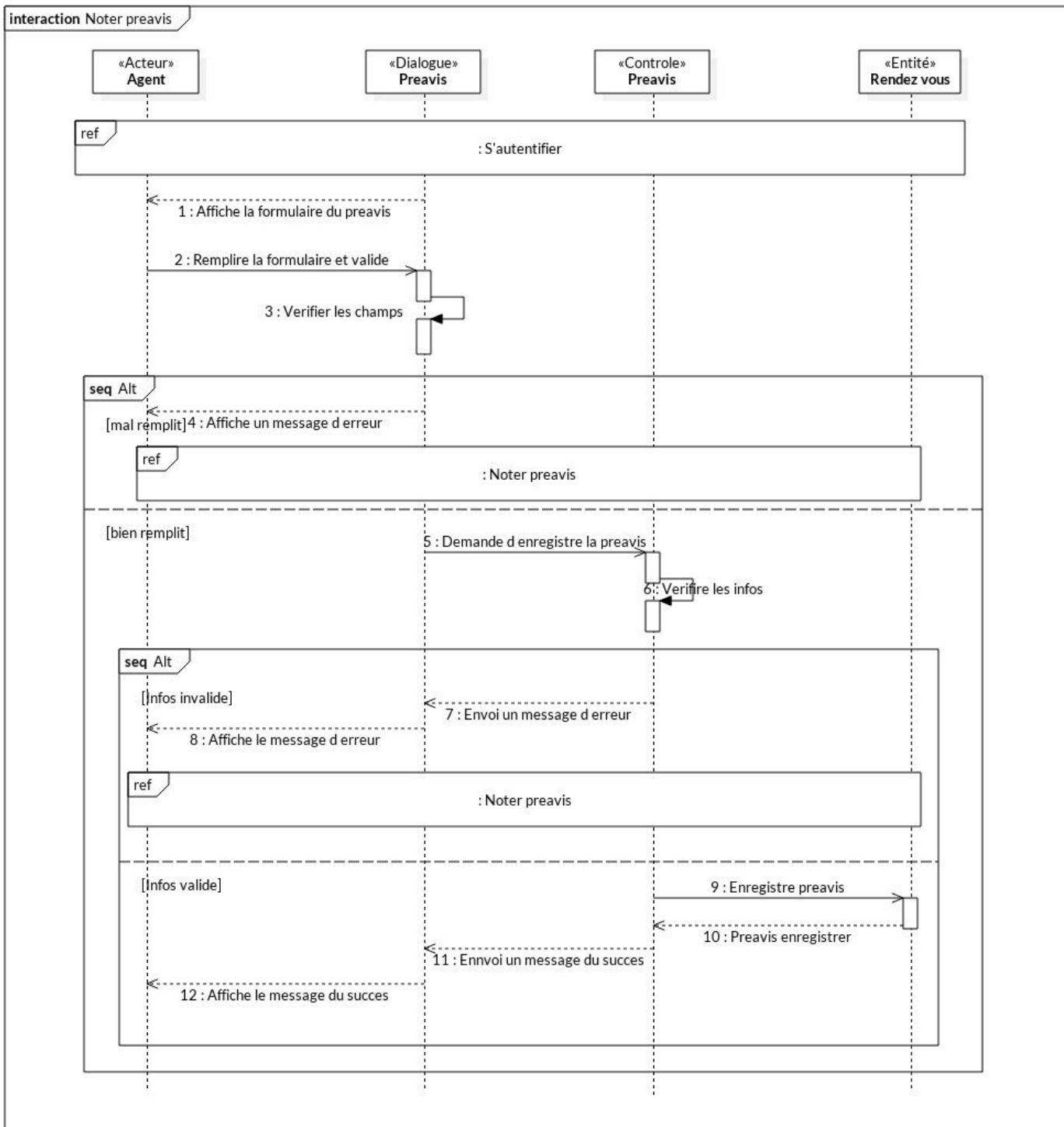
### 3.6- Consulter rendez-vous :



### 3.8- Valider l'inscription :

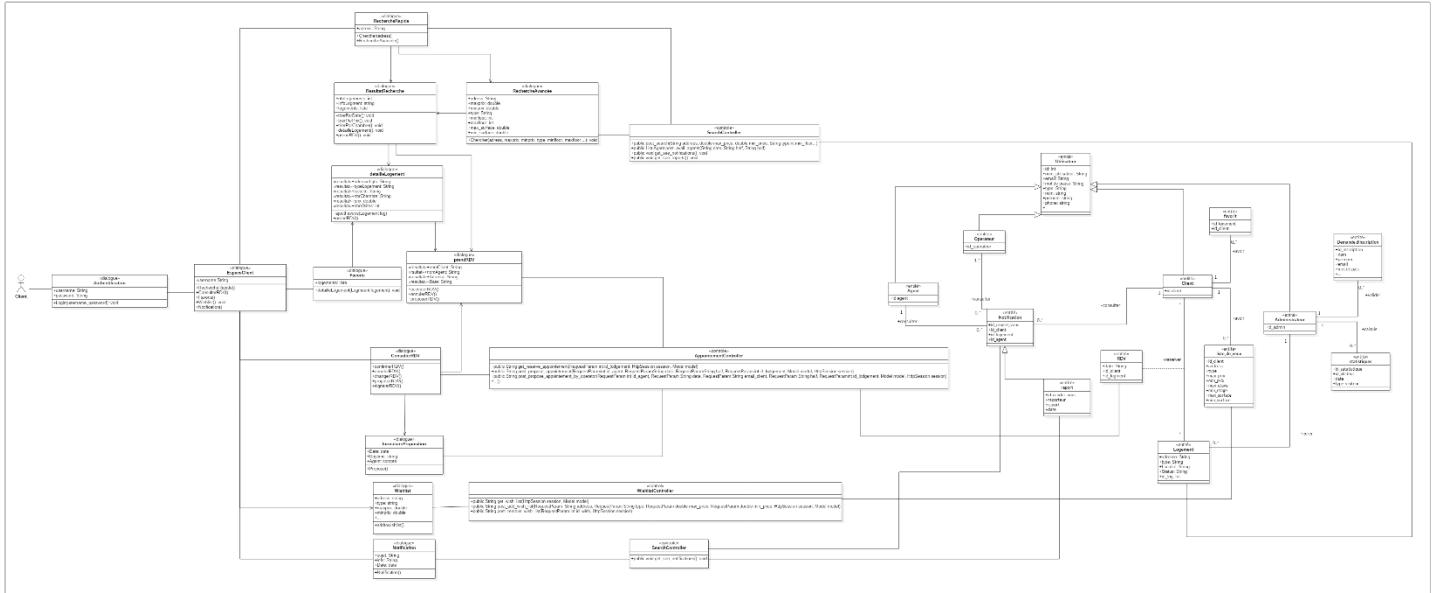


### 3.9- Noter préavis :

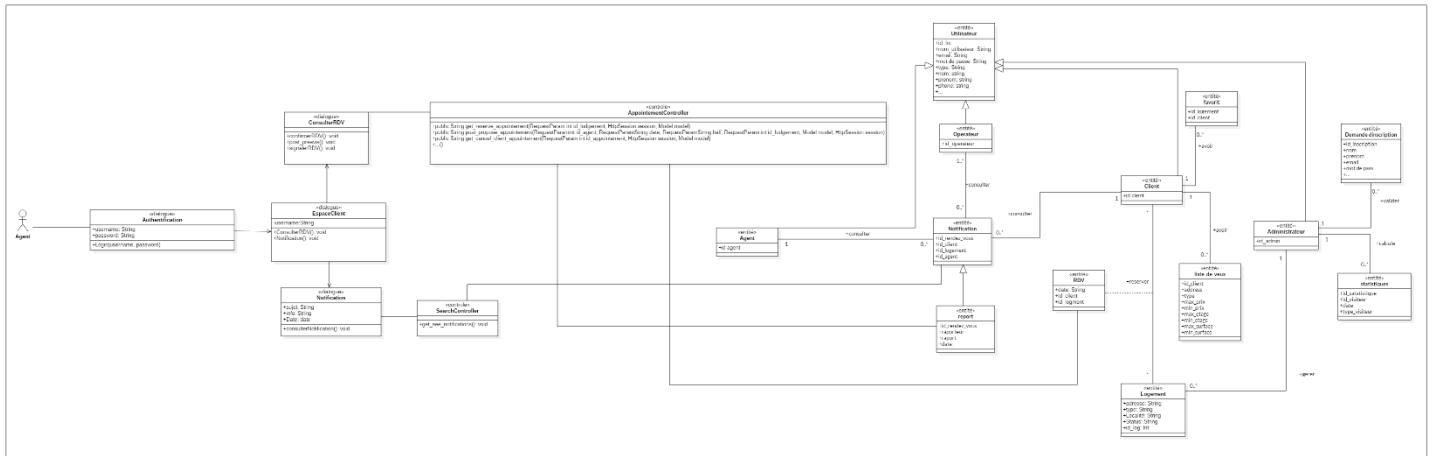


## 5- Les Diagrammes des classes conception :

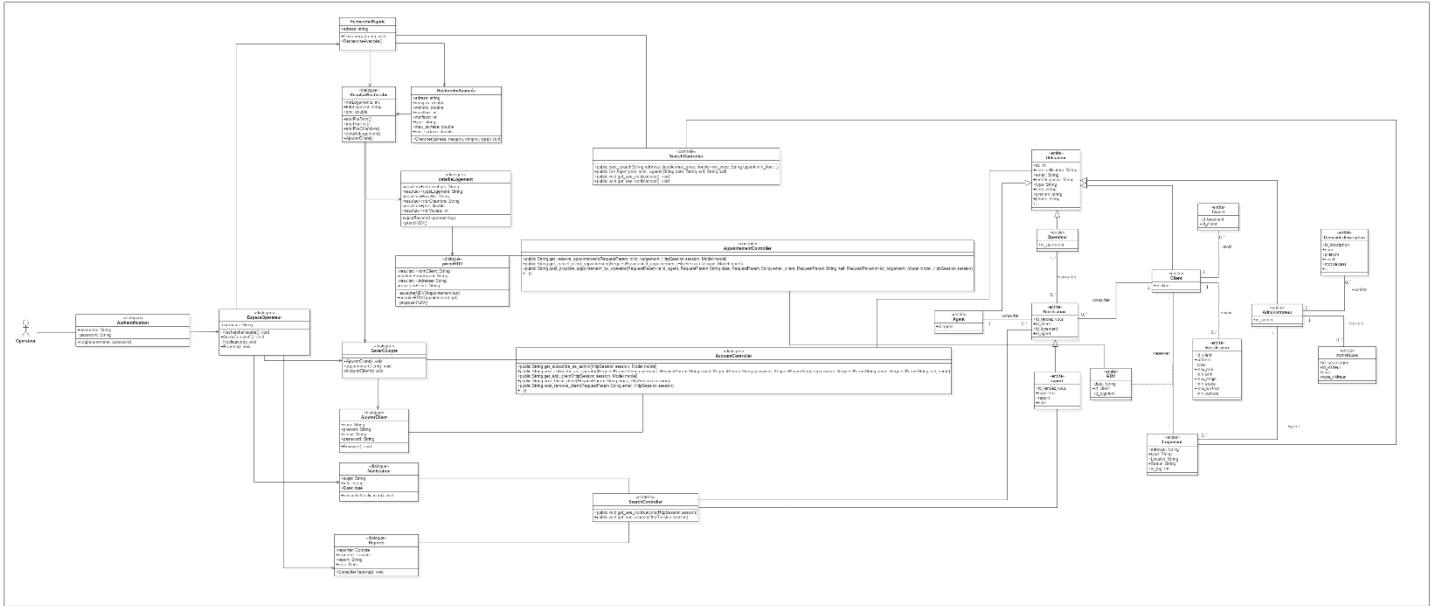
## 2-1. Client :



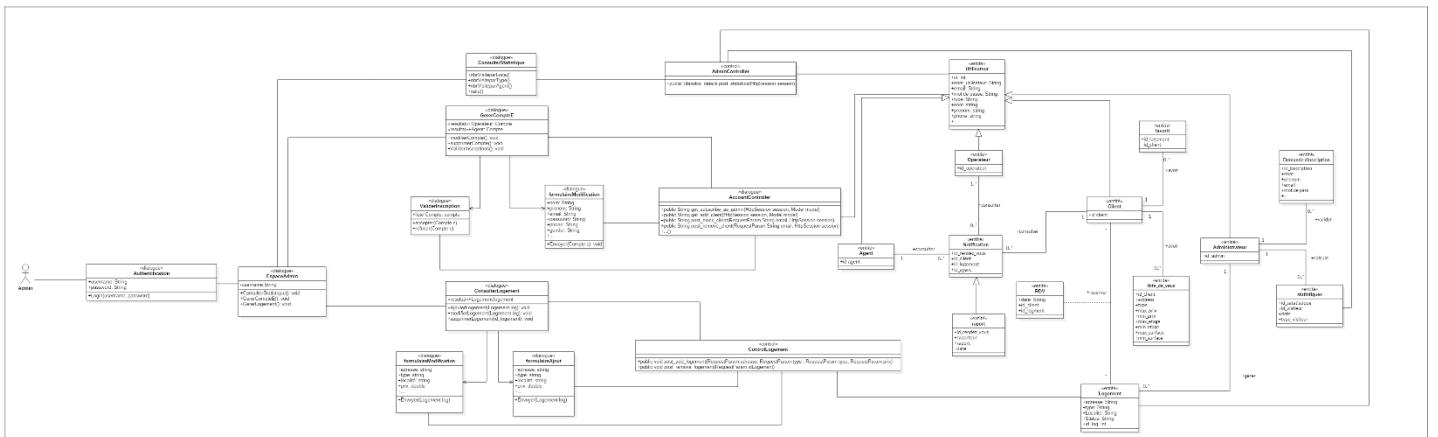
## 2.2- Agent :



## 2.3- Operateur :



## 2.4-Administrateur :



## 6-Conclusion :

L'analyse et la conception d'un logiciel exige un travail délicat. Ce chapitre montre déjà ce qu'est la structure de données qui vont être stockées dans la base de données, il reste le codage.

# Chapitre 3 : Implémentation

## 1- Introduction

A Ce stade de processus, il reste que la programmation et l'implémentation du site.

Dans ce chapitre nous allons décrire les différents logiciels et outils, ainsi que les langages de programmation utilisés dans la création et la mise en œuvre des différentes parties de notre application. Ainsi nous décrivons l'environnement de développement du système et de la base de données.

## 2- Plateformes :

Il y a deux types de plateformes sont à présenter ici :

- Plateforme matérielle, nous allons présenter les machines sur lesquelles nous avons réalisé et testé notre site.
- Plateforme logicielle représente les outils, logiciels et les langages de programmation orientés web utilisés.

### 2.1 Plateforme matérielle :

#### - Machine Serveur :

PC	DESKTOP-Toshiba
Type du système	Ubuntu 16.04
Processeur	Intel Coré i3-3300U
Mémoire installe(RAM)	4GB
Disque dur	500GB
Carte Graphique	Intel IV bridge Mobile
Carte réseaux	Athéros Ar5006X

#### - Machine Client :

PC	DESKTOP-3GVVBHB
Type du système	Windows 10 Pro 64-bit
Processeur	Intel Core i5-6200U
Mémoire installe(RAM)	8GB
Disque dur	500GB
Carte Graphique	AMD Radeon M330
Carte réseaux	Realtek RTL8723BE

## 2.2 Plateforme logicielle :

Pour développer un système qui répond aux besoins, il faut utiliser des systèmes et des logiciels complexes. On présente ici les outils de développement que nous avons utilisé.

Système d'exploitation : Ubuntu 16.04

### 3- Langages de programmation :

Dans le web, un seul langage de programmation ne suffit pas pour faire fonctionner un site ou une application web de qualité, on a besoin donc d'utiliser plusieurs langages chacun son rôle.

HTML 5 :

HTML est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. C'est le langage responsable sur la structure de la page web.



CSS 3

Les feuilles de style en cascade, généralement appelées CSS de l'anglais « *Cascading Style Sheets* », forment un langage informatique qui décrit la présentation des documents HTML. Introduit en HTML 4, CSS est le langage responsable sur le style du site (tous ce qui concerne les couleurs et le design).



JS :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs. Son rôle est de rendre le site ou l'application plus fluide et plus facile à utiliser.



AJAX :

L'architecture informatique Ajax permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client en se servant de différentes technologies ajoutées aux navigateurs web. Ajax est très essentiel surtout pour les formulaires dynamiques.



XML :

Langage extensible de balisage, est un langage de description des données. Il permet de structurer et échanger des données de type texte. Il est dit extensible car il permet à l'utilisateur de personnaliser ses balises.



## Java EE :

Le terme « Java EE » signifie *Java Enterprise Edition*, et était anciennement raccourci en « J2EE ». Il est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise. Il fait quant à lui référence à une extension de la plate-forme standard. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées, déployées et exécutées sur un serveur d'applications.



## Spring :

Est un Framework libre pour construire et définir l'infrastructure d'une application java<sup>4</sup>, dont il facilite le développement et les tests. Il est effectivement un conteneur dit "léger", c'est-à-dire une infrastructure similaire à un serveur d'applications J2EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets. Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le Framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».



## Spring Web MVC :

Spring Web MVC Framework fournit une architecture MVC (Model-View-Controller) et des composants prêts à l'emploi qui peuvent être utilisés pour développer des applications Web flexibles et faiblement couplées. Le modèle MVC permet de séparer les différents aspects de l'application (logique d'entrée, logique métier et logique UI) tout en assurant un couplage lâche entre ces éléments.



- Un modèle (Model) contient les données à afficher.
- Une vue (View) contient la présentation de l'interface graphique.
- Un contrôleur (Controller) contient la logique concernant les actions effectuées par l'utilisateur.

## Spring Boot :

Est un micro Framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet.



## Java Server Pages :

JSP est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédefinies d'être ajoutés dans un contenu statique.



## Java Server Pages Standard Tag Library :

JSTL est un composant de la plate-forme JEE de développement. Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle,



## Expression langage :

Une EL permet d'accéder simplement aux beans des différents scopes de l'application web (page, requête, session et application). Utilisé conjointement avec des librairies de tags, elles permettent de se passer totalement des scriptlets.

## SQL :

Afin d'exploiter notre BD, on utilise SQL qui signifie « Structured Query Language » (langage de requête structurée). C'est un langage qui permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.



## Hibernate :

Est un Framework open source gérant la persistance des objets en base de données relationnelle. Il est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE complet.



## La Java Persistance :

JPA est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java.



## 4- Outils de développement :

MySQL :

Est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par les professionnels.



Interface d'SGBD > PhpMyAdmin :

PhpMyAdmin est une application web qui permet de gérer un serveur de bases de données MySQL. Dans un environnement multi-utilisateur, cette interface écrite en PHP permet également de donner à un utilisateur un accès à ses propres bases de données.



Serveur Web > Apache Tom Cat :

Est un conteneur web libre de servlets et JSP Java EE. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Fondation. Il implémente les spécifications des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et des propriétés, et inclut des outils pour la configuration et la gestion. Il comporte également un serveur HTTP.



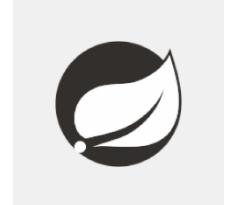
IDE > Spring Tool Suite :

Est un environnement de développement basé sur Eclipse qui est personnalisé pour le développement d'applications Spring. Il fournit un environnement prêt à l'emploi pour implémenter, déboguer, exécuter et déployer vos applications Spring.



Spring Initialiser :

Fournit une API extensible pour générer des projets Quickstart. Il offre également un service configurable « <https://start.spring.io> ». Il fournit une interface utilisateur Web simple pour configurer le projet à générer et les points de terminaison que vous pouvez utiliser via Https simples.



Logiciel de gestion de dépendances > Apache Maven :

Est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il utilise un paradigme connu sous le nom de *Project Object*



## Logiciel de gestion de versions > Git :

Est un logiciel de gestion de versions décentralisé. Un système qui permet le développement facile des projets informatiques. Avec ce système, plusieurs développeurs peuvent travailler sur un projet sans qu'il y'aurait des ambiguïtés, ni de problèmes de versions de code. Le code sera open source afin que les développeurs du monde entier puissent le consulter, commenter et l'améliorer en proposant une nouvelle branche de développement.



## GitHub :

Est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git



## Chrome :

Est un navigateur web propriétaire développé par Google basé sur le projet libre Chromium fonctionnant sous Windows, Mac, Linux, Android et iOS.



## StarUML

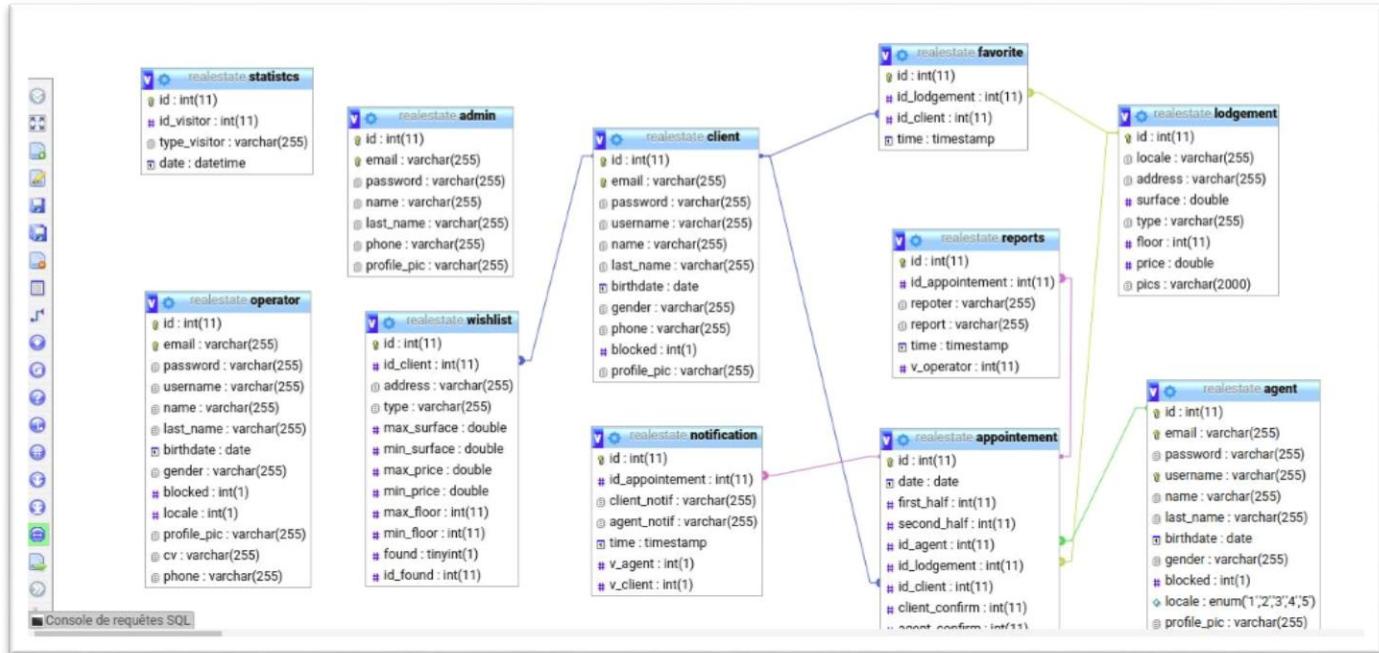
Est un Logiciel de modélisation UML



## 5- Présentation de notre application :

### 3.1 Réalisation de la base de données :

La base de données que nous allons utiliser contient tables pour conserver les différentes informations nécessaires pour l'invocation des fonctionnalités du site.



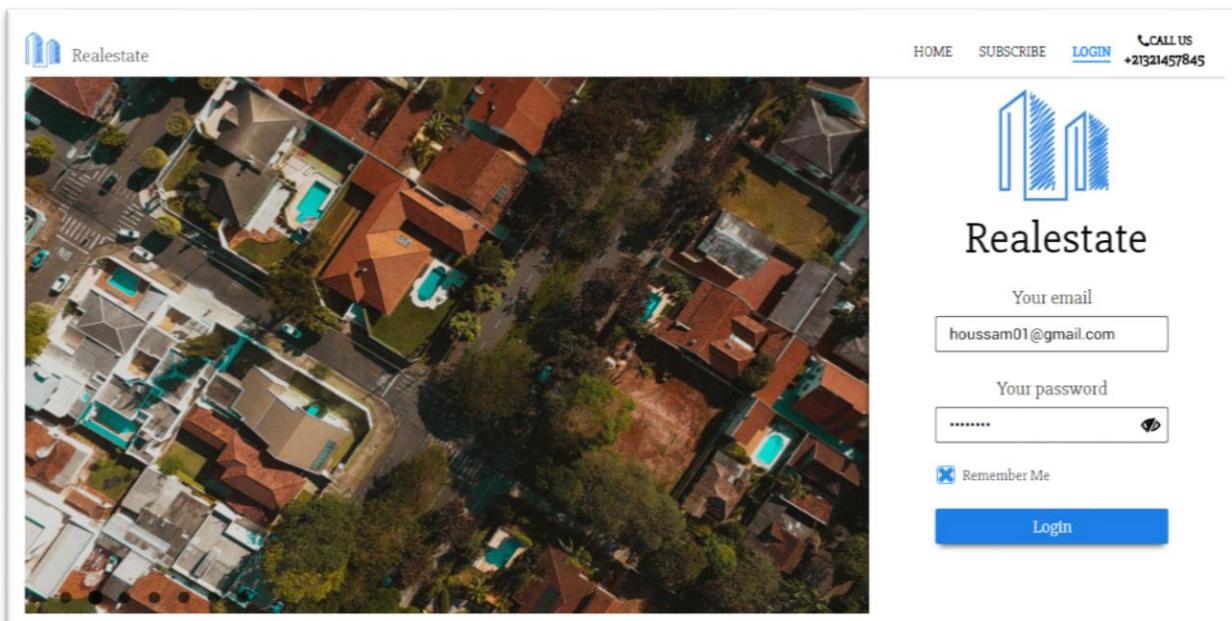
### 3.2- Exemple Interface :

#### - Cas d'utilisation « Prendre Rendez-vous » :

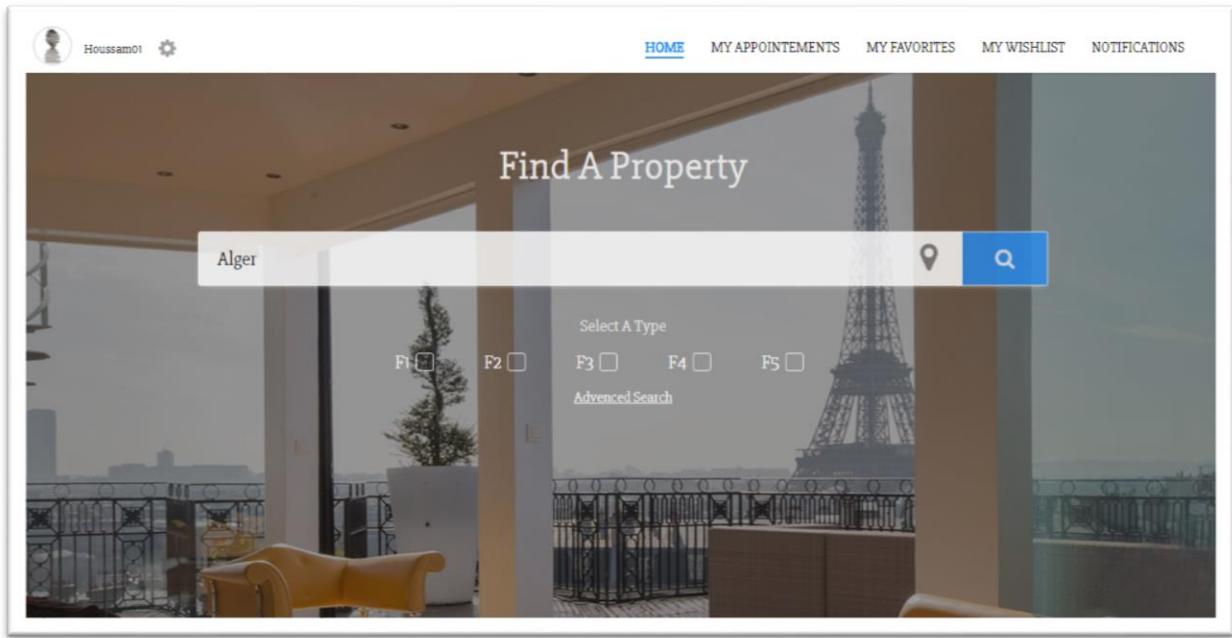
On va présenter le processus de réservation d'un du point de vue d'un client et d'un agent :

**Client :**

- Se connecter en tant que client :



- On Cherche une propriété par état :



- Nous choisissons un logement parmi les résultats :

A screenshot of a search result page. At the top, there are type selection buttons for F1 through F5. Below them is a link to "Advanced Search". The main content area shows a large image of a modern apartment interior with a kitchen and a balcony. Below the image, the text "The Search Result :" is displayed. To the left of the image is a thumbnail of the same apartment with arrows for navigation and a "More Details" button. To the right of the image, under the heading "Appartement's Informations:", are the following details: Address: Alger, Price: 15000000 DA, Type: F1, Floor: 2, Surface: 50 M<sup>2</sup>. At the bottom right is a blue button labeled "Reserve An Appointement".

- Nous réservons un rendez-vous après avoir choisi l'heure et l'agent préférés :

The screenshot shows a user profile "Houssam01" at the top left. The top navigation bar includes links for HOME, MY APPOINTEMENTS, MY FAVORITES, MY WISHLIST, and NOTIFICATIONS. A green success message box displays "The appointment has been saved." Below this, a section titled "Propose An Appointment:" contains fields for Date (jj/mm/aaa), Time (Morning dropdown), Agent (No Agent is available dropdown with a "visit profile" link), and a blue "Propose" button. Another section titled "Choose An Appointment:" lists available slots from June 6 to June 9, 2018, with columns for Date, Time, Agent (Agent26 dropdown with "visit profile" link), and a "Reserve Now" button.

Date	Time	Agent	Reserve
2018-06-06	Evening	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>
2018-06-07	Morning	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>
2018-06-07	Evening	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>
2018-06-08	Morning	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>
2018-06-08	Evening	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>
2018-06-09	Morning	Agent26	<a href="#">visit profile</a> <a href="#">Reserve Now</a>

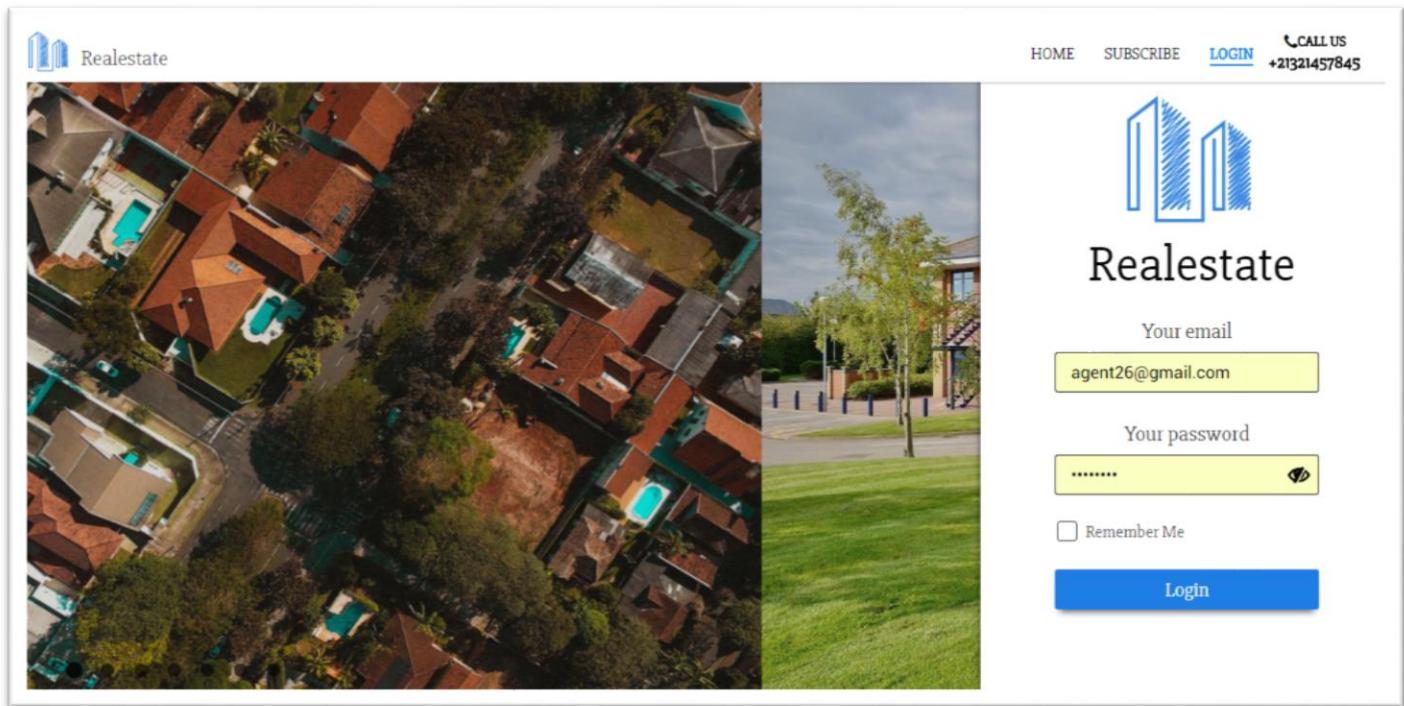
- Notre travail est fait quand la notification verte apparaît, nous pouvons également vérifier nos Rendez-vous.

The screenshot shows a user profile "Houssam01" at the top left. The top navigation bar includes links for HOME, MY APPOINTMENTS (which is underlined in blue), MY FAVORITES, MY WISHLIST, and NOTIFICATIONS. A section titled "My appointments:" shows a table of booked appointments. The table columns are Date, Time, Agent (Agent26 dropdown with "visit profile" link), Lodgement ("F1 in alger" dropdown), Agent Confirmation (Not confirmed.), Your Confirmation (Not confirmed.), The review (No review), Report a probleme (You can't report yet.), and Cancel. The first row shows an appointment for June 6, Morning, with all fields populated except for the lodgement which is currently empty.

Date	Time	Agent	Lodgement	Agent Confirmation	Your Confirmation	The review	Report a probleme	Cancel
2018-06-06	Morning	agent26 <a href="#">visit profile</a>	F1 in alger <a href="#">visit profile</a>	Not confirmed.	Not confirmed.	No review	You can't report yet.	<a href="#">Cancel</a>

Maintenant la partie de l'agent

- L'authentification :



-Comme vous pouvez le voir, l'agent a un nouveau rendez-vous.

A screenshot of the agent's dashboard. At the top left is a user icon and the text "Agent26". At the top right are links for "MY APPOINTEMENTS", "NOTIFICATIONS", and "LOGOUT". Below the header is a section titled "My appointments:" with a subtitle "Find An Appointment:". It includes a date input field "jj/mm/aaaa", a "Search" button, and two tabs: "All My Appointments" (selected) and "Valid Appointments". A table below lists the agent's upcoming appointment details:

Date	Time	Client	Lodgement	Your Confirmation	Client Confirmation	The review	Report a problem	Cancel
2018-06-06	Morning	houssam01	in alger	Not confirmed.	Not confirmed.	You can't write a review yet.	You can't report yet.	<a href="#">Cancel</a>

A ce stade, tout ce qu'il à faire est d'attendre avant 24 heure de le rendez-vous pour valider le, après le meeting l'agent peut écrire le préavis si le client est satisfait de l'appartement ou non.

- Cas d'utilisation « Consulter les statistiques » > Admin :



## 6- Conclusion

Dans ce dernier chapitre, nous avons présenté la structure de notre système, les outils utilisés et la façon dont on a intégré les nouvelles notions et les bonnes pratiques pour produire un résultat propre réutilisable et maintenable. Cela a été aussi l'occasion de montrer les différentes parties de l'application développée par quelques captures d'écran.

# Chapitre 4 : Les Tests Unitaires

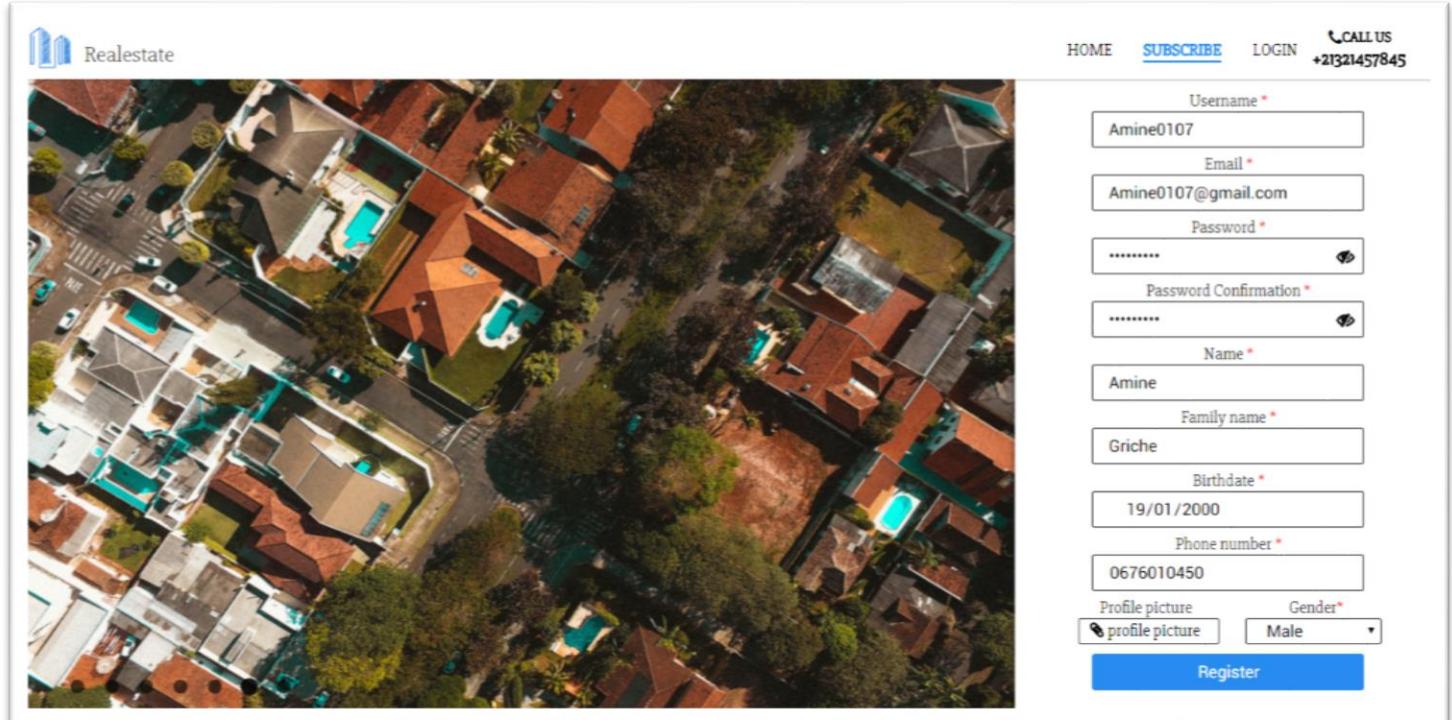
## 1- Introduction :

Le test d'un logiciel est une phase très importante car il permet de voir si ce logiciel répond aux exigences lui concernant. Ces tests permettent de vérifier individuellement que chaque sous-ensemble du logiciel est implémenté conformément aux spécifications. En effet, Pour tester notre application nous avons sélectionné les données que nous jugeons représentatives des situations cernées dans notre travail. Cette activité consiste à tester les résultats de l'implémentation pour s'assurer du bon déroulement des fonctionnalités du système.

## 2- Les Unités :

Nous allons tester le fonctionnement de l'ajout d'utilisateur :

- La page d'inscription sur le navigateur



The screenshot shows a web page for a real estate platform. At the top left is the logo 'Realestate'. At the top right are links for 'HOME', 'SUBSCRIBE', 'LOGIN', and a phone number '+21321457845'. Below these are several input fields for user registration, each marked with a red asterisk indicating it is required:

- Username\*: Amine0107
- Email\*: Amine0107@gmail.com
- Password\*: (redacted)
- Password Confirmation\*: (redacted)
- Name\*: Amine
- Family name\*: Griche
- Birthdate\*: 19/01/2000
- Phone number\*: 0676010450
- Profile picture: (button to upload)
- Gender\*: Male (dropdown menu)

A large blue 'Register' button is located at the bottom right of the form area. The background of the form is an aerial photograph of a residential neighborhood with houses, streets, and pools.

- La table des clients dans la base de données après l'inscription :

	id	email	password	username	name	last_name	birthdate	gender	phone	blocked	profile_pic
1	amine@gmail.com	00000000	amine77	amine	griche		1995-06-27	male	0559377270	1	default_profile_picture.jpg
2	amine.griche77@gmail.com	00000000	amine77	amine	griche		2018-04-23	male	0559767676	0	default_profile_picture.jpg
4	houssam01@gmail.com	00000000	houssam01	houssam	boudiar		2002-12-11	male	0559767676	0	default_profile_picture.jpg
5	amine02@gmail.com	00000000	amine02	amine	amine		2002-12-04	male	0559767676	0	default_profile_picture.jpg
6	amine03@gmail.com	00000000	amine03	amine	amine		1995-06-27	male	0559677270	0	default_profile_picture.jpg
7	amine05@gmail.com	00000000	amine05	amine	amine		2002-12-18	male	0559559598	0	default_profile_picture.jpg
8	amine06@gmail.com	00000000	amine06	amine	amine		2002-12-09	male	0559559598	0	default_profile_picture.jpg
9	amine07@gmail.com	00000000	amine07	amine	amine		2002-12-09	male	0559559598	0	default_profile_picture.jpg
10	amine08@gmail.com	00000000	amine08	amine	amine		2002-12-09	male	0559559598	0	default_profile_picture.jpg
11	amine09@gmail.com	00000000	amine09	amine	amine		2002-12-09	male	0559559598	0	default_profile_picture.jpg
12	amine10@gmail.com	00000000	amine10	amine	amine		2002-12-09	male	0559559598	0	default_profile_picture.jpg
14	amine11@gmail.com	00000000	amine11	amine	amine		1993-09-24	male	0559767676	0	1527548527111.jpg
16	amine12@gmail.com	00000000	amine12	amine	amine		1994-01-01	male	0559767676	0	1527545271828.jpg
17	Amine0107@gmail.com	123456789	Amine0107	Amine	Griche		2000-01-19	Male	0676010450	0	

- Le test on utilisons J-unit :

« Ajouter un utilisateur puis essayer de l'obtenir en utilisant l'email et le mot de passe »

```

84
85
86     // should be true , Subscribe test ... adding a user then trying to get it using "getuserByEmailAndPassword" method
87 @Test
88 public void testADDUser(){
89
90     User user = new User();
91
92     user.setUsername("amine");
93     user.setEmail("amine@gmail.com");
94     user.setPassword("123456789");
95     user.setType("client");
96     user.set fname("amine");
97     user.set f_name("amine");
98     user.setBirthdate("2002-12-14");
99     user.setPhone("0606040558");
100    user.setAddress("alger");
101    user.setGender("Male");
102
103    UserRepository mock = org.mockito.Mockito.mock(UserRepository.class);
104    Mockito.when(mock.save(user)).thenReturn(user);
105    assertEquals(not(userRepository).getuserByEmailAndPassword("amine@gmail.com", "123456789").getUsername(), user.getUsername());
106 }
107 //should be false , adding a user already exists on our database
108 @Test
109 public void falsetestADDUser(){
110
111     User user = new User();
112
113     user.setUsername("amine100");
114     user.setEmail("amine100@gmail.com");
115     user.setPassword("00000000");
116     user.setType("00000000");
117     user.set fname("amine");
118     user.set f_name("amine");
119     user.setBirthdate("2018-04-24");
120     user.setPhone("0559677270");
121     user.setAddress("algeria");
122     user.setGender("Male");
123
124    UserRepository mock = org.mockito.Mockito.mock(UserRepository.class);
125    Mockito.when(mock.save(user)).thenReturn(user);
126    assertEquals(not(userService).adduser(user), isEqualTo(true));
127 }
128

```

JUNIT 35  
Finished after 8.243 seconds  
Runs: 2/2 Errors: 0 Failures: 1

service.userServiceTest [Runner: JUnit 4] (1.182 s)  
falsetestADDUser (0.977 s)  
testADDUser (0.201 s)

Failure Trace  
Junit.ComparisonFailure: expected:<[true]> but was:<[false]>  
at service.userServiceTest.falsetestADDUser(userServiceTest.java:126)  
at org.springframework.test.context.junit4.statements.RunBeforeTestMethodPart.  
at org.springframework.test.context.junit4.statements.RunAfterTestMethodPart.  
at org.springframework.test.context.junit4.statements.RunBeforeTestClassPart.  
at org.springframework.test.context.junit4.statements.RunAfterTestClassPart.  
< at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChild  
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.runChildren  
at org.springframework.test.context.junit4.SpringJUnit4ClassRunner.  
<

« Modification des données du client puis tester s'il est bloqué ou non »

The screenshot shows an IDE interface with two main panes. The left pane displays the Java code for `userServiceTest.java`. The right pane shows the JUnit test results for `service.userServiceTest`.

**Code (userServiceTest.java):**

```
136     //Should be true , blocking user then "blocked = 1" then comparing getBlocked() it to "1"
137 @Test
138 public void testupdateuser(){
139
140     User user = new User();
141
142     user.setId(43);
143     user.setUsername("amine1111");
144     user.setPassword("00000000");
145     user.setEmail("amine1111@gmail.com");
146     user.setType("Client");
147     user.setName("amine");
148     user.setF_name("amine");
149     user.setBirthdate("2002-11-14");
150     user.setPhone("0605040550");
151     user.setAddress("alger");
152     user.setGender("Male");
153     UserRepository mock = org.mockito.Mockito.mock(UserRepository.class);
154     Mockito.when(mock.updateUserStatus(user.getId(), 1)).thenReturn(user.getBlocked());
155     assertEquals(user.getRepository().getByEmailAndPassword("amine1111@gmail.com", "00000000").getBlocked(),1);
156 }
157 //Should be false , updating "blocked to 0(keeping it the same)" then comparing it to 1
158 @Test
159 public void falsetestupdateuser(){
160
161     User user = new User();
162     user.setId(44);
163     user.setUsername("houssam");
164     user.setPassword("1234567");
165     user.setEmail("hou@gmail.com");
166     user.setType("Client");
167     user.setName("boudair");
168     user.setF_name("houssam");
169     user.setBirthdate("2002-12-04");
170     user.setPhone("0605040550");
171     user.setAddress("bou el ater");
172     user.setGender("Male");
173     UserRepository mock = org.mockito.Mockito.mock(UserRepository.class);
174     Mockito.when(mock.updateUserStatus(user.getId(), 0)).thenReturn(user.getBlocked());
175     assertEquals(user.getRepository().getByEmailAndPassword("hou@gmail.com", "1234567").getBlocked(),0);
176     //System.out.println("Blocked : "+ user.getBlocked());
177 }
178
179
180 }
```

**JUnit Test Results:**

Finished after 8.185 seconds

Runs: 2/2	Errors: 0	Failures: 1
-----------	-----------	-------------

service.userServiceTest [Runners: JUnit 4] [1.015 s]

- testUpdateUser (0.922 s)
- falseTestUpdateUser (0.093 s)

**Failure Trace:**

```
java.lang.ComparisonFailure: expected:<[]> but was:<[0]>
  at service.userServiceTest.falsetestupdateuser(service.userServiceTe
  at org.springframework.test.context.junit4.statements.RunBefor
  at org.springframework.test.context.junit4.statements.RunAft
  at org.springframework.test.context.junit4.statements.RunBefor
  at org.springframework.test.context.junit4.statements.RunAft
  at org.springframework.test.context.junit4.statements.RunBefor
  at org.springframework.test.context.junit4.statements.RunAft
  at org.springframework.test.context.junit4.statements.Spring
  at org.springframework.test.context.junit4.SpringJUnit4Class
  at org.springframework.test.context.junit4.SpringJUnit4Class
  at org.springframework.test.context.junit4.statements.RunBefor
  at org.springframework.test.context.junit4.statements.RunAft
```

« Tester le type de retourne de la fonction get\_Users() , Users list length et la fonction FindUserByID() »

The screenshot shows an IDE interface with several windows:

- Left Panel (Code View):** Displays the Java code for `userServiceTest.java`. The code contains three test methods: `TestgetAllUsersNotNull()`, `TestgetUsersListLength()`, and `testFindUserById()`.
- Top Right Panel (JUnit Results):** Shows the JUnit results with a summary: "Finished after 7.738 seconds", "Runs: 3/3", "Errors: 0", and "Failures: 1". It lists the three test cases with their execution times.
- Bottom Right Panel (Failure Trace):** Displays the stack trace for the failed assertion in `testFindUserById()`.
- Bottom Left Panel (Console Log):** Shows the terminal output of the JUnit run, including timestamp, log level, and message details.

## Conclusion Générale

Ce projet de fin d'étude consiste à réaliser un site web dynamique pour la prise des rendez-vous pour un mieux gestion d'une société immobilière.

Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de notre site. Dans le but de satisfaire les utilisateurs nous avons commencé la conception en utilisant le langage de modélisation UML 2.0 et le processus de développement Pasqual Roques pour ensuite faire la mise en œuvre des bases de données à l'aide du SGBD MYSQL, après nous avons entamé l'implémentation de toutes les requêtes SQL pour la manipulation des données et enfin la concrétisation de l'application sous langage de programmation J2E.

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine de la programmation. Nous avons appris à mieux manipuler les langages J2E et Java Script.

Finalement, on est arrivé à réaliser notre site avec une simple et attrayante interface graphique qui est de nos jours, un atout important dans les sites web.

En effet, ce travail étant une œuvre humaine, n'est pas un modèle unique et parfait, c'est pourquoi nous restons ouverts à tous les critiques et nous sommes prêts à recevoir toutes les suggestions et remarques tendant à améliorer d'avantage cette étude. Etant donné que tout travail informatique a été toujours l'œuvre d'une équipe.