



RAPPORT DU PROJET DU MODULE : TECHNOLOGIE WEB

SEMESTRE S3
OPTION GÉNIE LOGICIEL

Intitulé :

Application Web pour la gestion des achats et des locations des hébergements

Réalisé par :

GHAZLANE Mohammed (GL 2)

NACHIT Btissam (GL 3)

SABOUR Ilham (GL 3)

Sous la direction de :

Pr. EL HAMLAOUI Mahmoud

Année Universitaire : 2019 / 2020

Remerciements

Nous tenons tout d'abord à remercier Monsieur Mahmoud EL HAMLAOUI pour son encadrement qui nous a bien orienté et aidé à voir les choses au clair. Ainsi pour son cours concis et bien organisé qui nous a permis d'améliorer nos connaissances et compétences du côté conceptuel et analytique.

Ensuite, nous remercions chacun d'entre nous d'avoir participé à ce projet sérieusement et réalisé pleinement son travail en respectant les délais et les consignes qui lui sont attribués.

Enfin nous dirigeons notre grande reconnaissance à nos familles et à tout le comité pédagogique qui nous a offert cette opportunité de travailler sous de bonnes circonstances, avec tant de fierté et à qui nous dédions ce travail.

Résumé

Dans le cadre de notre formation au sein de l'Ecole Supérieure d'Informatique et d'Analyse des Systèmes nous sommes amenés à réaliser une application web JEE. Par conséquent, nous avons décidé que cette application portera sur le marché de l'immobilier et sera une plateforme d'échange d'offres et de demandes.

Cette application sera une interface entre tous ceux qui veulent mettre en vente ou en location leurs propriétés, et ceux qui désirent en profiter sans avoir à se déplacer ou à fournir beaucoup d'effort.

A travers cette plateforme, le client choisit l'offre qui lui convient, et suit l'état de sa demande jusqu'à la confirmation ou l'annulation du propriétaire.

Abstract

As part of our training at the Higher School of Computer Science and Systems Analysis we are led to create a JEE web application. Therefore, we have decided that this application will focus on the real estate market and will be a platform for the exchange of offers and requests.

This application will be an interface between all those who want to sell or rent their properties, and those who want to enjoy them without having to travel or make a lot of effort.

Through this platform, the customer chooses the offer that suits him, and follows the status of his request until the owner confirms or cancels it.

Table des figures

1.1	Le modèle de cycle de vie en V	11
2.1	Maquette accueil	13
2.2	Maquette Inscription	14
2.3	Maquette Connexion	14
2.4	Maquette Accueil client	15
2.5	Maquette offres client	15
2.6	Maquette Détails offre	16
2.7	Maquette Demandes client	16
2.8	Maquette Réservations	17
2.9	Maquette Recherche offre	17
2.10	Maquette Reservation Voyage	18
2.11	Maquette Message	18
2.12	Maquette Ajout offre	19
2.13	Maquette Liste des offres	19
2.14	Maquette Détails demande	20
2.15	Maquette liste des clients	20
2.16	Maquette Détails client	21
2.17	Maquette Liste des messages	21
2.18	Diagramme de cas d'utilisation globale	24
2.19	Diagramme de classe	25
2.20	Diagramme de package	26
2.21	Diagramme de séquence « Se connecter »	27
2.22	Diagramme de séquence « faire une recherche »	27
2.23	Diagramme de séquence « Nouveau utilisateur »	28
2.24	Diagramme de séquence « Gestion offre par vendeur »	28
3.1	Présentation du modèle d'accès aux fichiers JSP	31
3.2	Eclipse logo	33
3.3	XAMPP logo	33
3.4	Tomcat logo	34
3.5	Entreprise Architect logo	34
3.6	Git Bash logo	35
3.7	HTML logo	35
3.8	CSS logo	36
3.9	javascript logo	36
3.10	UML logo	37

3.11 Bootstrap logo	37
3.12 jQuery logo	38
3.13 Architecture MVC2	39
3.14 Interface accueil	40
3.15 Interface inscription	41
3.16 Interface connexion	42
3.17 Interface contact	43
3.18 Interface "Menu client"	44
3.19 Interface "Liste des tous les offres"	45
3.20 Interface "Liste des réservations"	46
3.21 Interface "Liste des réservations confirmées"	47
3.22 Interface pour cherche une offre	48
3.23 Menu vendeur	49
3.24 Interface pour créer une offre	50
3.25 Interface "Liste des offres créées"	51
3.26 Interface "Liste des demandes"	52
3.27 Menu administrateur	53
3.28 Interface "Liste des clients"	54
3.29 Interface "Liste des propriétaires"	55

Table des matières

1	Contexte Général du Projet	9
1.1	Présentation du projet	9
1.1.1	Contexte du projet	9
1.1.2	Objectif du projet	9
1.1.3	Résultat attendu du projet	9
1.2	Planning du projet	10
1.2.1	Découpage de projet en tâche	10
1.2.2	Cycle de vie en V	10
2	Analyse et conception	12
2.1	Analyse	12
2.1.1	Besoins fonctionnels	12
2.1.2	Besoins non-fonctionnels	12
2.2	Conception	13
2.2.1	Maquettes	13
2.2.2	Vue fonctionnelle des besoins	21
2.2.3	Vue statique des besoins	25
2.2.4	Vue dynamique des besoins	26
3	Réalisation	29
3.1	Choix de la plateforme JavaEE	29
3.1.1	Contexte générale	29
3.1.2	La technologie JavaEE	29
3.1.3	L'Architecture JavaEE	30
3.1.4	Composants de base de JavaEE	30
3.2	Architecture applicative	31
3.2.1	Couche présentation	31
3.2.2	Couche métier	32
3.2.3	Couche persistance	32
3.3	Boîte à outils	33
3.3.1	IDE eclipse	33
3.3.2	XAMPP	33
3.3.3	Apache Tomcat 9.0	34
3.3.4	Entreprise architecte	34
3.3.5	Github	35
3.4	Langages, technologies et API utilisées	35

3.4.1	HTML5	35
3.4.2	CSS3	36
3.4.3	JavaScript	36
3.4.4	UML	37
3.4.5	Bootstrap	37
3.4.6	JQuery	38
3.5	Modèle MVC2	38
3.5.1	Définition	38
3.5.2	Avantages	38
3.5.3	Architectures MVC2	39
3.6	Mise en œuvre	39
3.6.1	Accueil	39
3.6.2	Espace client	44
3.6.3	Espace vendeur	49
3.6.4	Espace administrateur	53

Introduction

L'informatique étant une science de traitement automatique de données s'avère bénéfique dans tous les domaines qu'ils soient scientifiques ou professionnels, privés et/ou publics. En observant les grandes entreprises dans le monde, on se rend vite compte qu'elles réalisent des travaux complexes en fractions de temps très réduit à l'aide des machines, ce qui leur coûterait des journées manuellement.

En plus, le commerce électronique connaît une formidable croissance et succès grâce à la sûreté, la disponibilité et l'excellence de temps de réponse qui est fourni par ces applications.

En se référant à la gestion des Locations et des Ventes de logement. Ce genre de travaux ne s'effectuent plus à la main, mais par les machines et les applications web afin de faciliter la communication entre l'acheteur et le vendeur ou bien entre le locataire et le locateur et la rendre plus facile et plus sûre.

Donc, le but principal de ce projet consiste à développer une application web qui met en place les services qui permettent la gestion électronique des locations et des ventes de logement.

Dans ce contexte, ce présent rapport sera structuré en trois chapitre.

Le premier chapitre sera consacré à une présentation du contexte générale du projet.

La phase d'analyse des besoins et de conception fonctionnelle et technique sera détaillée dans le deuxième chapitre.

Ensuite, on abordera la partie de la réalisation du projet.

Et pour finir, nous concluons avec les perspectives du travail réalisé.

Chapitre 1

Contexte Général du Projet

Dans ce chapitre, nous présenterons une vue globale sur notre projet. Nous aborderons dans une première partie une présentation du projet, dans la deuxième section les services fournis par notre projet et dans la troisième section le planning du projet.

1.1 Présentation du projet

1.1.1 Contexte du projet

Ce projet est destiné pour la gestion de location et vente des hébergements, et il rentre dans le cadre de réalisation d'une application web qui assure le bon fonctionnement de cette gestion. Le projet est découpé en 3 parties : espace client, espace propriétaire et espace administration.

1.1.2 Objectif du projet

Il vise à assurer toutes les gestions : gestion des réservations, gestion de location, gestion d'achat, gestion de vente, gestion des offres, gestion des demandes et la gestion des clients et des propriétaires

La communication entre les utilisateurs de l'application (clients ou propriétaires) et l'administrateur constitue une chose primordiale c'est pour cela que nous avons aussi mis en place un service d'envoi de mail.

1.1.3 Résultat attendu du projet

Après la réalisation de cette application, On souhaite d'obtenir les résultats honorables. Parmi les résultats attendus on trouve :

1. Être facile au propriétaire d'ajouter des offres et de les gérer et de gérer les demandes envoyées par le client.
2. Être facile au client de lister les offres et de faire une demande de location ou d'achat d'un hébergement.
3. Être facile à l'administrateur de gérer les utilisateurs, les offres et les demandes.

4. Le système doit stocker les données qui concernent les utilisateurs et les offres de manière fiable.
5. Assurer un accès protégé et rapide.
6. Avoir une bonne gestion des différents composants de l'application.

1.2 Planning du projet

1.2.1 Découpage de projet en tâche

La planification de projet consiste à effectuer un découpage en phases chronologiques. Pour chaque phase, il faut ensuite déterminer la liste des tâches à accomplir, les charges à prévoir et les ressources nécessaires.

Le projet va connaître différentes phases :

1. **Etude préalable** : il s'agit de déterminer l'analyse des besoins et fonctionnelle de projet.
2. **Conception** : cette phase correspond à la conception technique de la solution. Des choix qui sont faits à ce stade dépendent l'estimation de la durée de réalisation des différentes tâches identifiées.
3. **Réalisation** : développement de la solution.
4. **Tests** : tests unitaires, techniques et fonctionnels de la solution.

1.2.2 Cycle de vie en V

Le projet en question utilise un mode de développement en cycle en V du fait de son périmètre fonctionnel important. Ce mode comprend les phases habituelles : Analyse, Conception, Réalisation/développement, Intégration, Validation.

Le cycle en V est un paradigme du développement d'un logiciel, le cycle de vie du projet. Il est représenté par un V dont la branche descendante contient toutes les étapes de la conception du projet, et la branche montante toutes les étapes de tests du projet. La pointe du V, quant à elle, représente la réalisation concrète du projet. Chaque étape d'une branche a son pendant dans l'autre branche, c'est-à-dire qu'une étape de conception correspond à une étape de test qui lui est spécifique. A tel point d'ailleurs, qu'une étape de test peut être élaborée dès que la phase de conception correspondante est terminée, indépendamment du reste du projet.

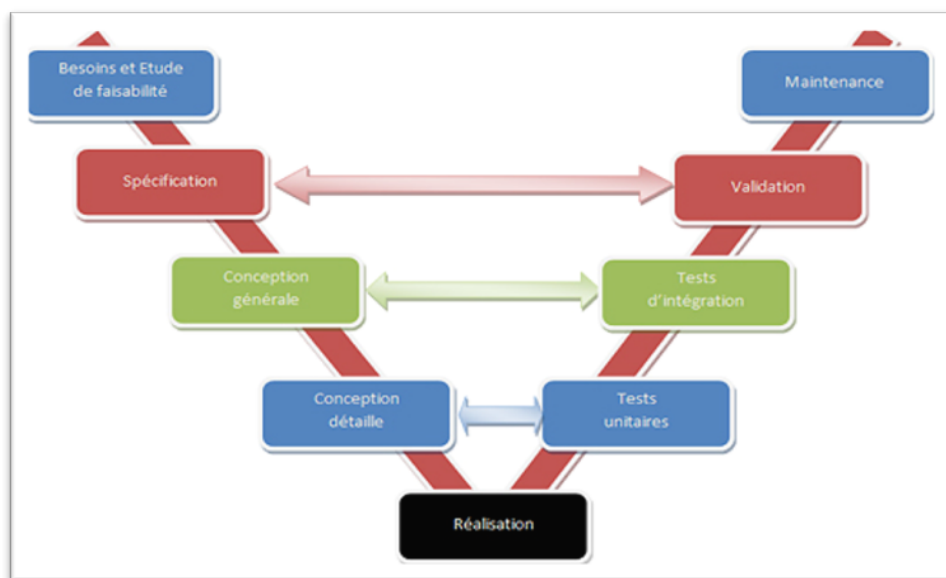


FIGURE 1.1 – Le modèle de cycle de vie en V

Chapitre 2

Analyse et conception

Dans le but de développer un système cohérent et complet, les phases de spécification de besoins, de conception fonctionnelle et de conception technique sont jugées très importante. En effet, elles permettent de recenser les fonctionnalités du système et de définir son architecture fonctionnelle et de déterminer les technologies utilisées, et c'est l'objectif du présent chapitre.

2.1 Analyse

2.1.1 Besoins fonctionnels

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Selon notre application, nous avons identifié les besoins fonctionnels :

- **La gestion des offres**
Elle doit être assurer : Ajout, suppression, modification et consultation des offres proposées par le vendeur.
- **La gestion des demandes**
Elle doit être assurer : Ajout, suppression, modification et consultation de ces demandes effectuées par le client.
- **La Gestion des comptes des utilisateurs de l'application**
Elle doit être assurer : Suppression et consultation des informations caractérisant chacun des utilisateurs.

2.1.2 Besoins non-fonctionnels

Les besoins non fonctionnels présentent des exigences internes au système et cachées aux utilisateurs :

- **Exigences d'ergonomie**
 - Une interface conviviale, lisible et facile à utiliser.

- Rapidité du temps de réponse car il joue un rôle fondamental. Il doit être court.
- Utiliser des listes contenant des valeurs correctes afin d'éviter la saisie et par conséquent les erreurs de saisie.

- **La Sécurité**

- Traçabilité et journalisation des transactions et des opérations effectuées
- Le système doit être sécurisé avec l'obligation pour chaque utilisateur de saisir un mot de passe et un login.
- L'authentification : Elle est assurée par un module d'authentification et d'autorisation, il gère l'accès entre les utilisateurs et l'application par un login et un mot de passe.

2.2 Conception

2.2.1 Maquettes

Cette partie présentera les maquettes de l'application. Celles-ci sont réalisées à partir de la plateforme « balsamiq.cloud » qui facilite la création des différentes composantes de l'application.

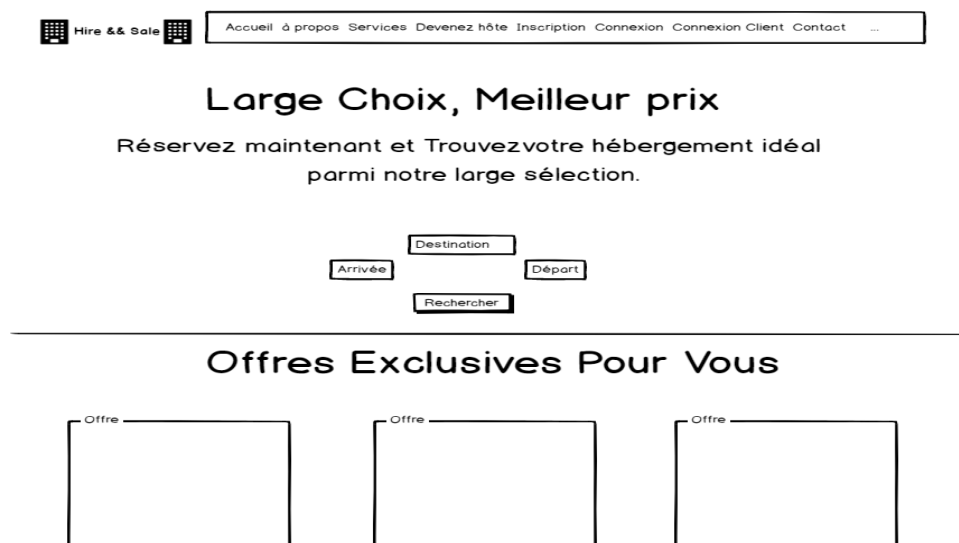




FIGURE 2.1 – Maquette accueil


Hire & Sale


Accueil
à propos
Services
Devenez hôte
Inscription
Connexion
Connexion Client
Contact
...

CRÉER VOTRE COMPTE
Remplir le formulaire suivant pour créer votre compte

Prénom * Prénom	Nom * Nom
Date de naissance : jj/mm/aaaa	Cin : Bk62...
Sexe : Homme	Email : exemple@domain.com
Adresse : St street ...	
Pays/Région : Maroc	Ville : Rabat
Téléphone : 0649...	

Inscription

FIGURE 2.2 – Maquette Inscription



Email

Mot de passe

Se connecter

Mot de passe oublié

FIGURE 2.3 – Maquette Connexion



FIGURE 2.4 – Maquette Accueil client

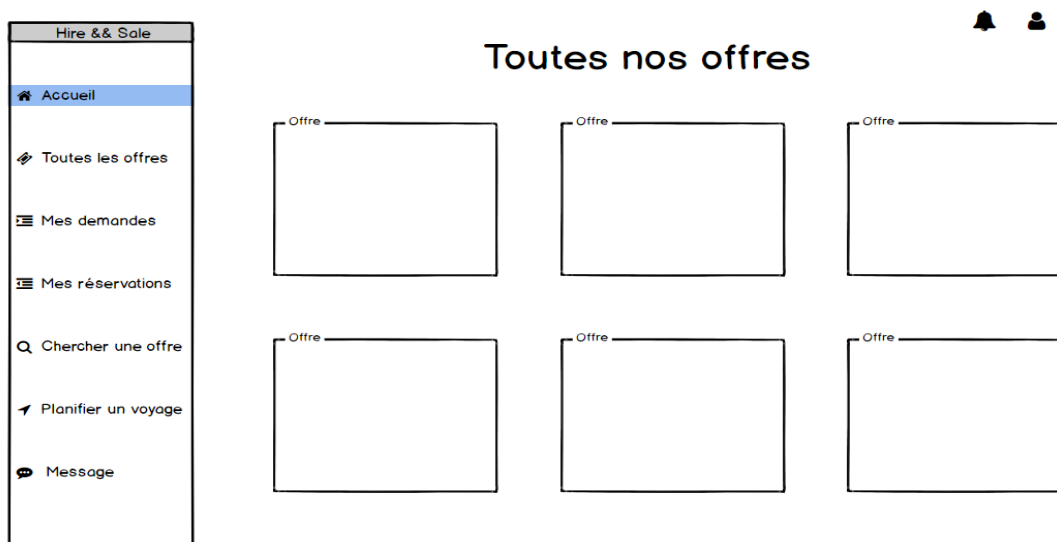


FIGURE 2.5 – Maquette offres client

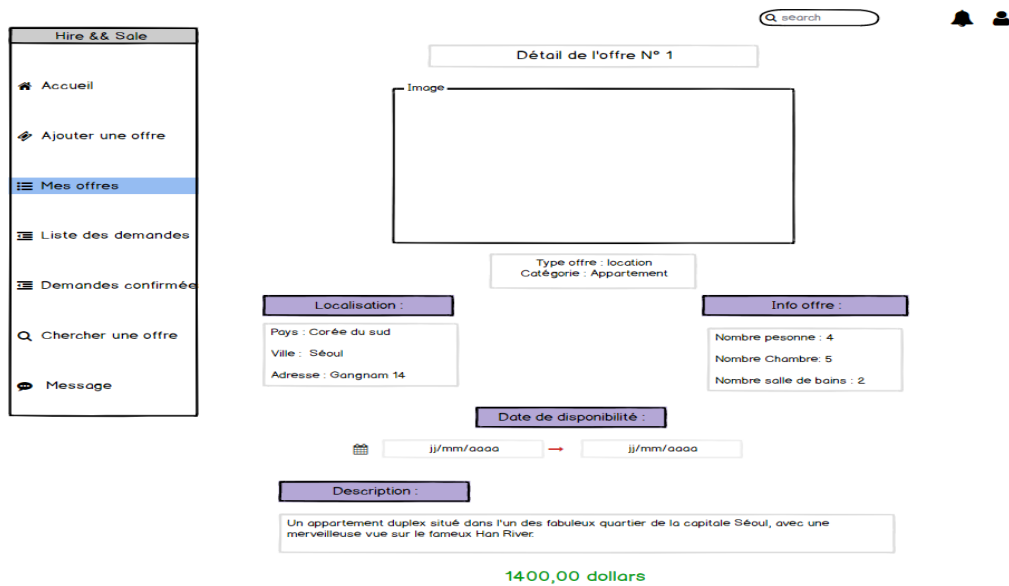


FIGURE 2.6 – Maquette Détails offre

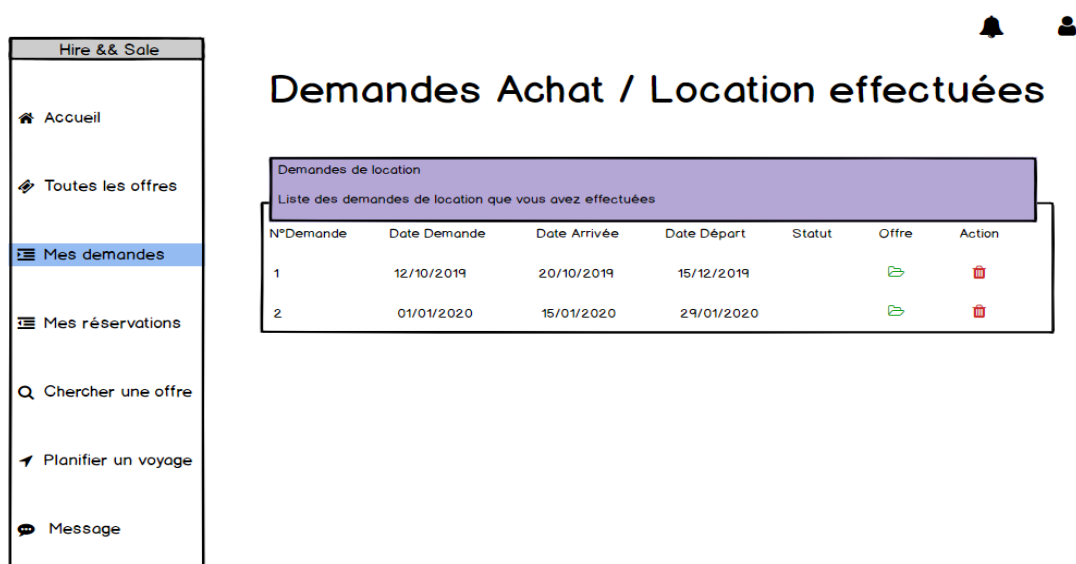


FIGURE 2.7 – Maquette Demandes client

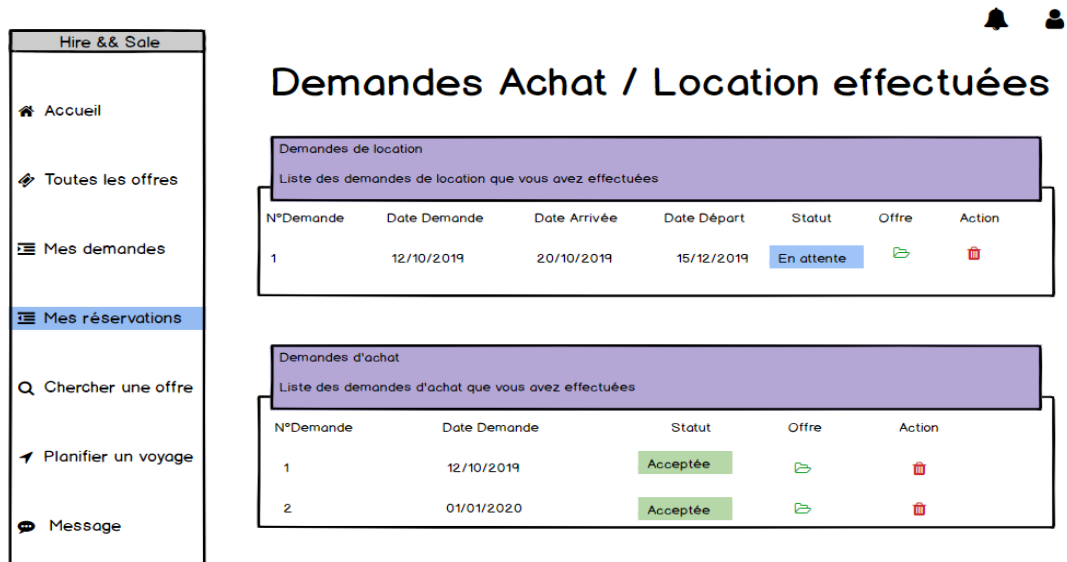


FIGURE 2.8 – Maquette Réservations

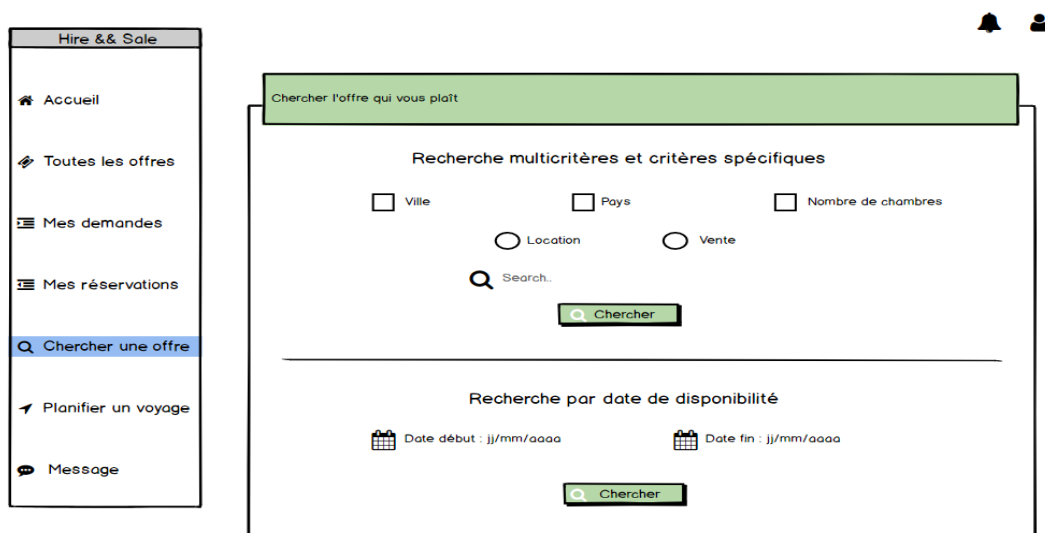


FIGURE 2.9 – Maquette Recherche offre

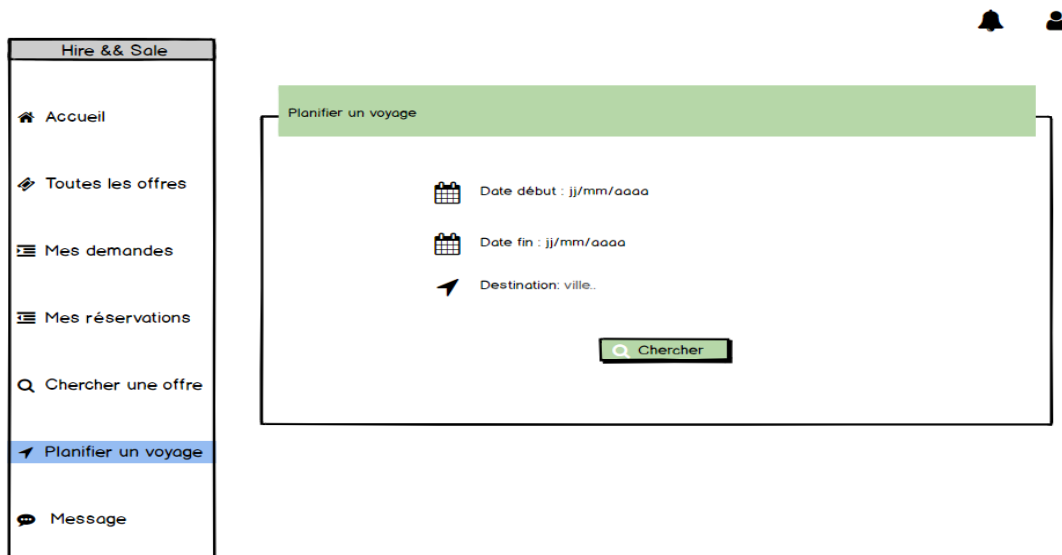


FIGURE 2.10 – Maquette Reservation Voyage



FIGURE 2.11 – Maquette Message

Hire && Sale

Accueil

Ajouter une offre

Mes offres

Liste des demandes

Demandes confirmées

Chercher une offre

Message

search

1

2

3

4

Catégorie

Emplacement

Informations

Description

CATEGORIE DE L'IMMOBILIER

Choisissez la catégorie de votre immobilier

Transaction :

Vente

Location

Location vacances

Catégorie :

Appartement

Maison

Villas & maisons de luxe

Riyad

Next

FIGURE 2.12 – Maquette Ajout offre

Hire && Sale

Accueil

Ajouter une offre

Mes offres

Liste des demandes

Demandes confirmées

Chercher une offre

Message

search

1

2

3

4

Catégorie

Emplacement

Informations

Description

Liste des offres

la liste de tous vos offres effectuées

Catégorie	Vente/Location	Pays	Ville	Adresse	Date Arrivée	Date Départ	Etat	Action
appartement	vente	France	Toulon		20/10/2019	15/12/2019	En attente	<div></div> <div></div> <div></div>

FIGURE 2.13 – Maquette Liste des offres

Q search

Détail de la demande N° 11

Type offre : location

Catégorie : Appartement

Status : En attente

Info demande :

jj/mm/aaaa

→

jj/mm/aaaa

Nombre de nuits : 10

Info Client :

Nom : Fadil

Prenom : Zakaria

Téléphone : 0647851447

Email : fzakaria@gmail.com

Accepter la demande

Refuser la demande

Hire & Sale

Accueil

Ajouter une offre

Mes offres

Liste des demandes

Demandes confirmées

Chercher une offre

Message

FIGURE 2.14 – Maquette Détails demande

Q search

Clients

La liste de tous les clients

Nom	Prénom	Date de naissance	Sexe	Pays	Téléphone	Action
Fadil	Zakaria	16/11/1989	Homme	Maroc	0614785412	<div></div> <div></div>
Rachid	Youssef	15/07/1991	Femme	France	0748141225	<div></div> <div></div>

Hire & Sale

Accueil

Liste des propriétaires

Liste des Clients

Liste des offres

Liste des demandes

Demandes confirmées

Message

FIGURE 2.15 – Maquette liste des clients

20

🔔
👤

Hire & Sale

- 🏠 Accueil
- 👤 Liste des propriétaires
- 👤 Liste des Clients
- 🔍 Liste des offres
- 📄 Liste des demandes
- ✅ Demandes confirmées
- 💬 Message

Données du client N° 5

Nom :

Prénom :

Cin :

Date de naissance:

Email :

Pays :

Ville :

FIGURE 2.16 – Maquette Détails client

🔔
👤

Hire & Sale

- 🏠 Accueil
- 👤 Liste des propriétaires
- 👤 Liste des Clients
- 🔍 Liste des offres
- 📄 Liste des demandes
- ✅ Demandes confirmées
- 💬 Message

Messages
 Liste de tous les messages que vous avez reçus

Type émetteur	Nom	Sujet	Action
Client	Rachid	Demande de clarification	<input type="checkbox"/> <input type="checkbox"/>
Propriétaire	Fadil	Réclamation	<input type="checkbox"/> <input type="checkbox"/>

FIGURE 2.17 – Maquette Liste des messages

2.2.2 Vue fonctionnelle des besoins

2.2.2.1 Identification des acteurs

Un acteur est une entité externe qui agit sur le système. Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes. Les acteurs de notre application sont :

1. L'administrateur

2. Le propriétaire : Vendeur ou Locateur
3. Le client

Administrateur

L'administrateur a la possibilité de :

- **Gérer les comptes**

1. Supprimer un utilisateur
2. Modifier un utilisateur
3. Consulter la liste des comptes de tous les utilisateurs

- **Gérer les offres**

1. Supprimer une offre
2. Consulter la liste des offres et leurs détails

- **Gérer les demandes**

1. Supprimer une demande
2. Consulter la liste des demandes

- **Gérer les messages**

1. Consulter les messages envoyés par les utilisateurs

Propriétaire

Le vendeur ou le locataire a la possibilité de :

- **Gérer ses offres**

1. Ajouter une offre selon son type (vente ou location) et sa catégorie (chambre, maison, hôtel ...)
2. Modifier une offre si elle n'est pas encore demandée par un client
3. Supprimer une offre
4. Rechercher une offre selon le type, la catégorie ou le pays ...

- **Gérer les demandes**

1. Consulter la liste des demandes envoyées par les clients
2. Accepter ou refuser une demande
3. Consulter la liste des demandes confirmées par le client après l'acceptation

- **Envoyer des messages**

1. Envoyer une réclamation à l'administrateur sous forme de message

Client

Le client a la possibilité de :

- **Voir les offres**

1. Consulter la liste des offres et leurs détails
2. Rechercher les offres selon leur type, leur catégorie et leur localisation ...

- **Gérer ses demandes**

1. Réserver une offre
2. Consulter la liste de ses demandes
3. Supprimer une demande
4. Confirmer une demande

- **Envoyer des messages**

1. Envoyer une réclamation à l'administrateur sous forme de message

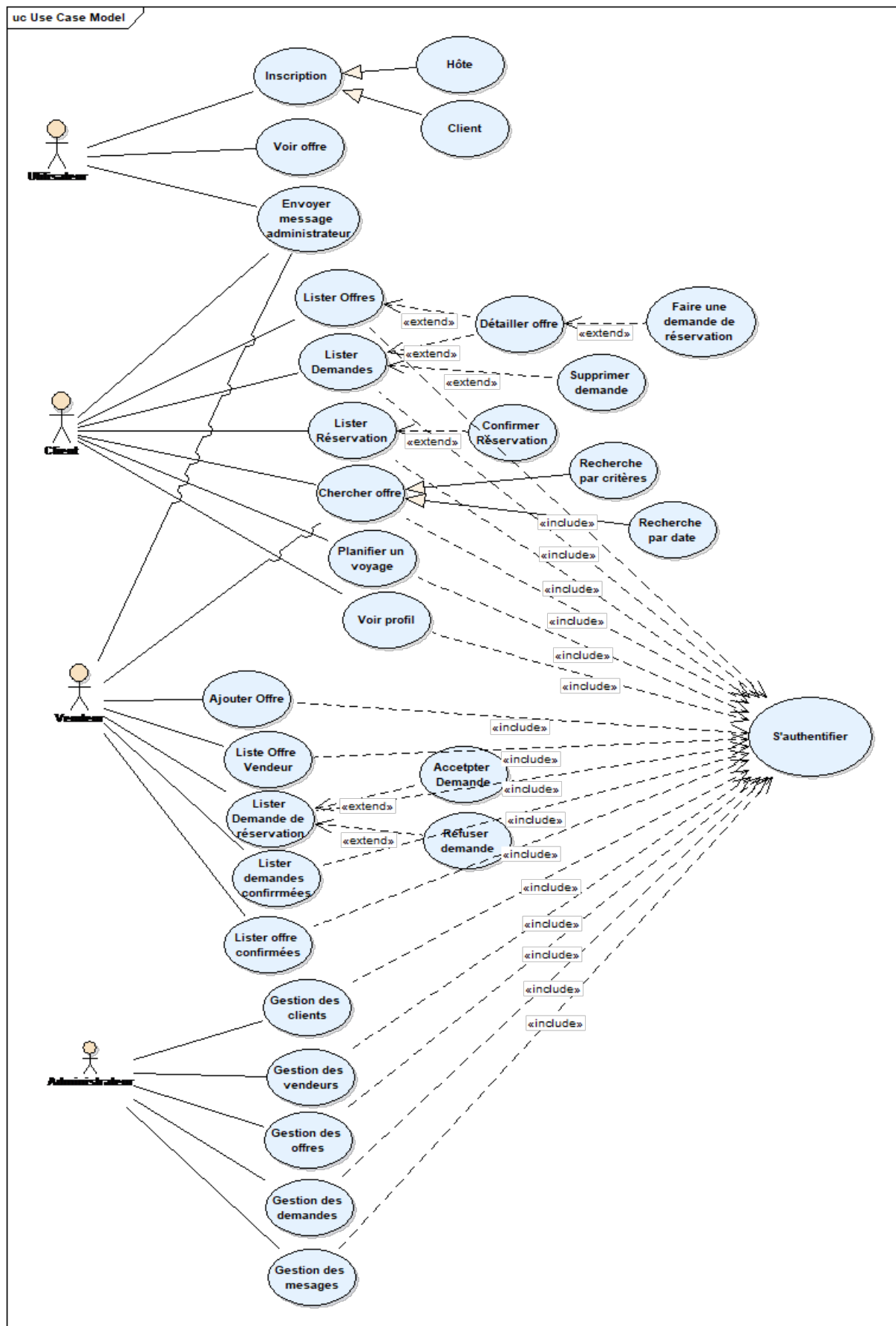
- **Planifier un voyage**

- **Voir ses réservations**

2.2.2.2 Diagramme de cas d'utilisation

Les cas d'utilisation décrivent le comportement du système du point de vue de l'utilisateur sous la forme d'actions et de réactions. Un cas d'utilisation indique une fonctionnalité du système déclenché par un acteur externe au système.

Dans cette partie, nous allons présenter le cas d'utilisation globale de l'application.



2.2.3 Vue statique des besoins

2.2.3.1 Diagramme de classe

Le diagramme de classes représente les classes constituant le système et les associations entre elles.

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classe et de relations entre ces classes.

De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens ; les objets sont des instances de classes et les liens sont des instances de relations.

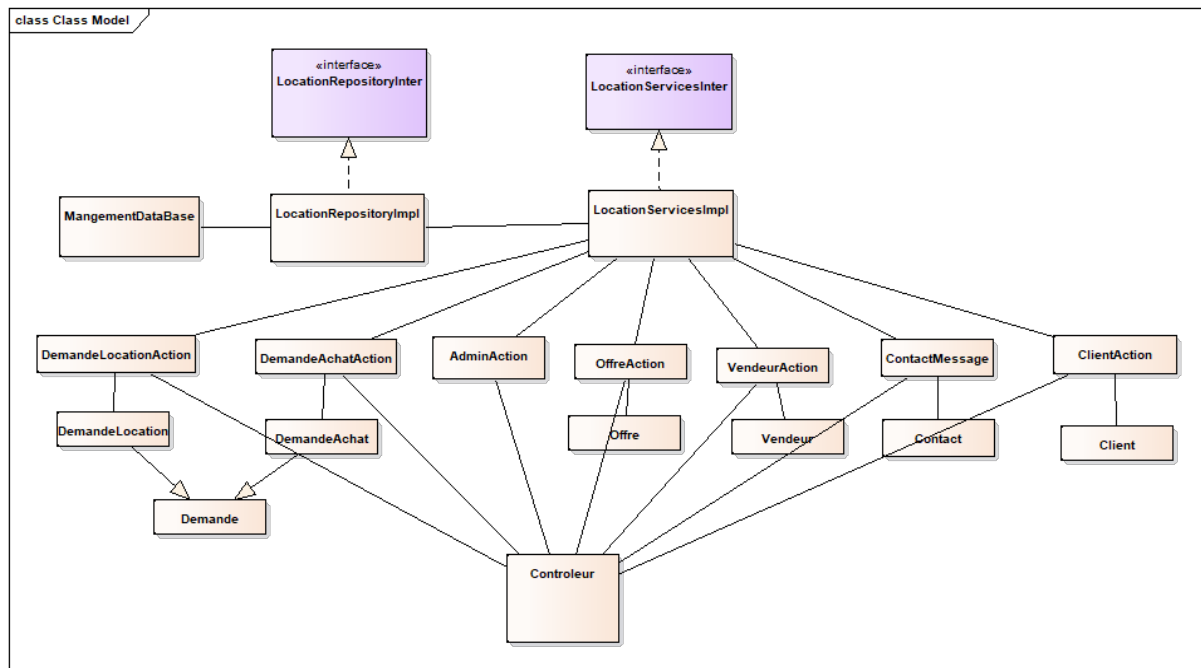


FIGURE 2.19 – Diagramme de classe

2.2.3.2 Diagramme de package

Un diagramme de packages est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de l'application, et il aide à identifier les liens de généralisation et de dépendance entre les packages.

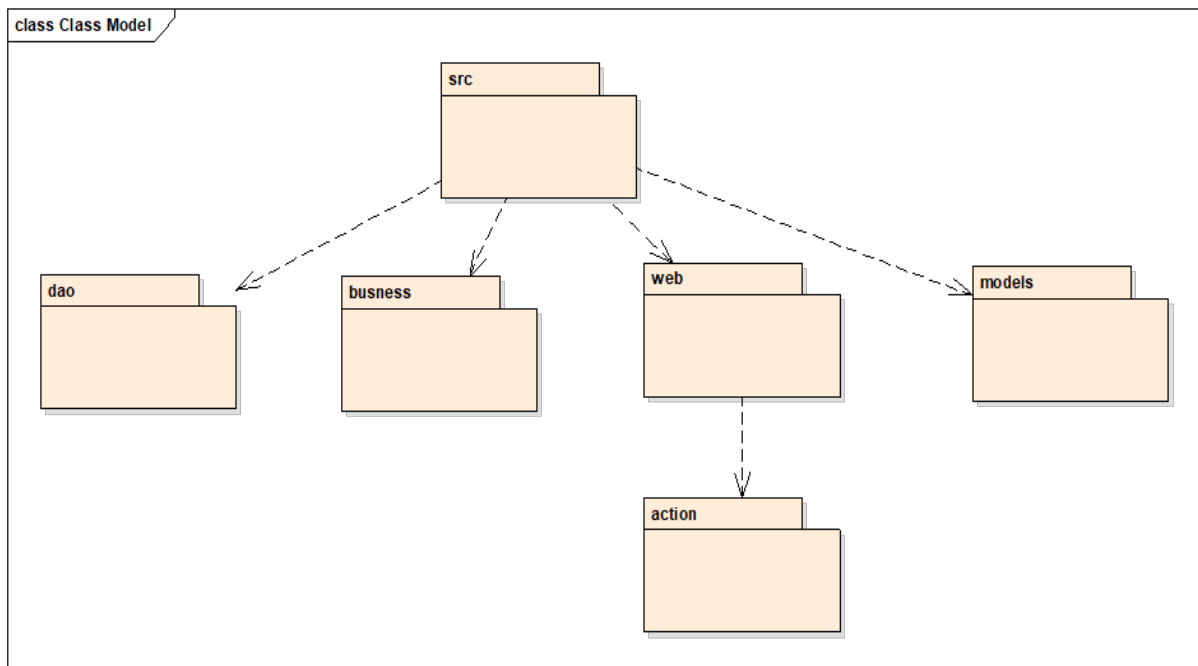


FIGURE 2.20 – Diagramme de package

2.2.4 Vue dynamique des besoins

2.2.4.1 Diagramme de séquences

Afin de bien illustrer les cas d'utilisations déjà élaborés, et dans le but de mieux représenter les interactions entre les objets de notre projet selon un point de vue temporel, nous avons utilisé les diagrammes de séquences.

L'objectif du diagramme de séquence est de représenter graphiquement les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.

Cette section alors présente les diagrammes de séquences dégagés pour décrire la communication du point de vue temporel entre les acteurs et le système. Ces diagrammes illustrent, clairement et avec des schémas simples, le comportement des objets constituant le système vis-à-vis des requêtes issues de ces acteurs.

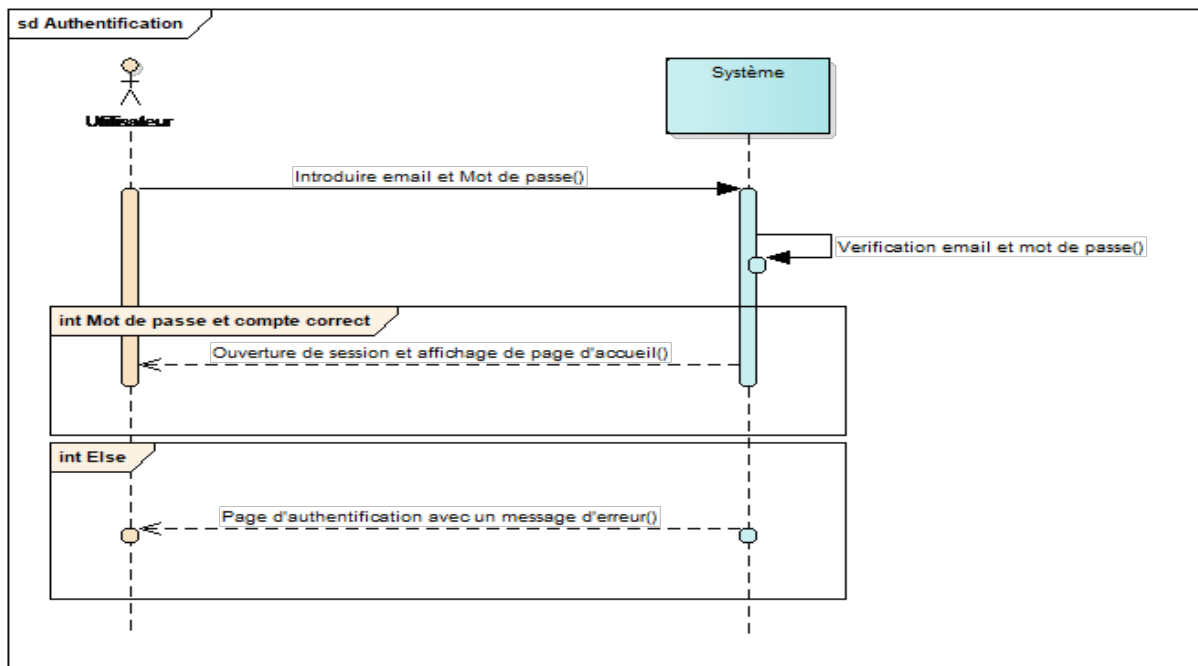


FIGURE 2.21 – Diagramme de séquence « Se connecter »

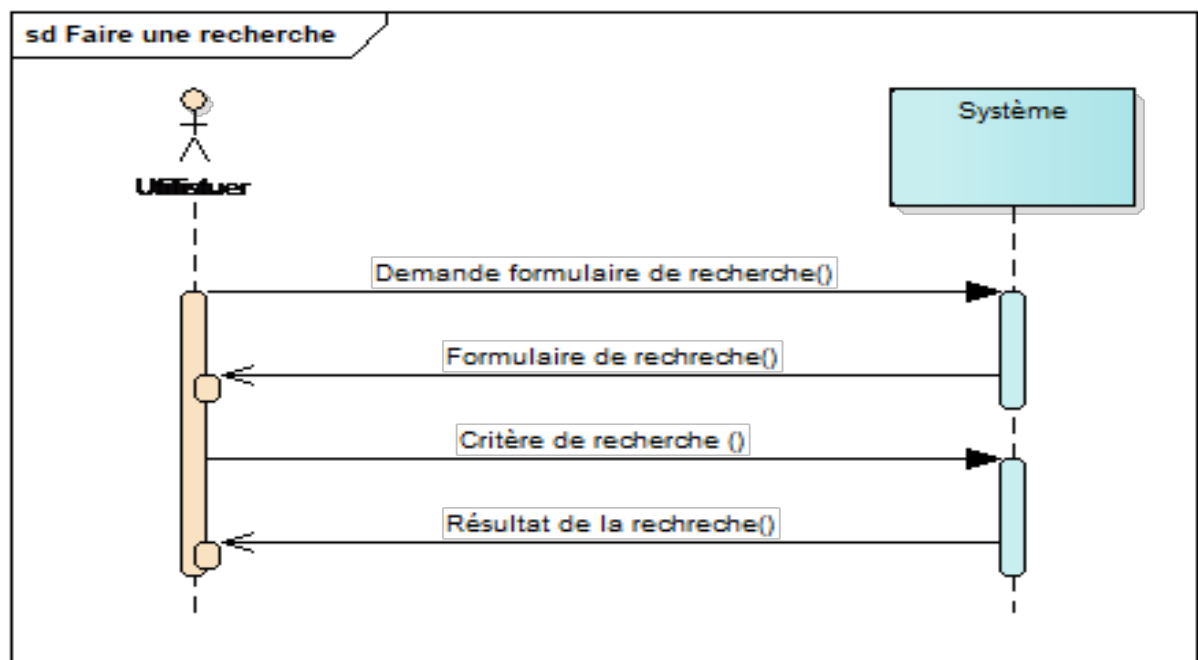


FIGURE 2.22 – Diagramme de séquence « faire une recherche »

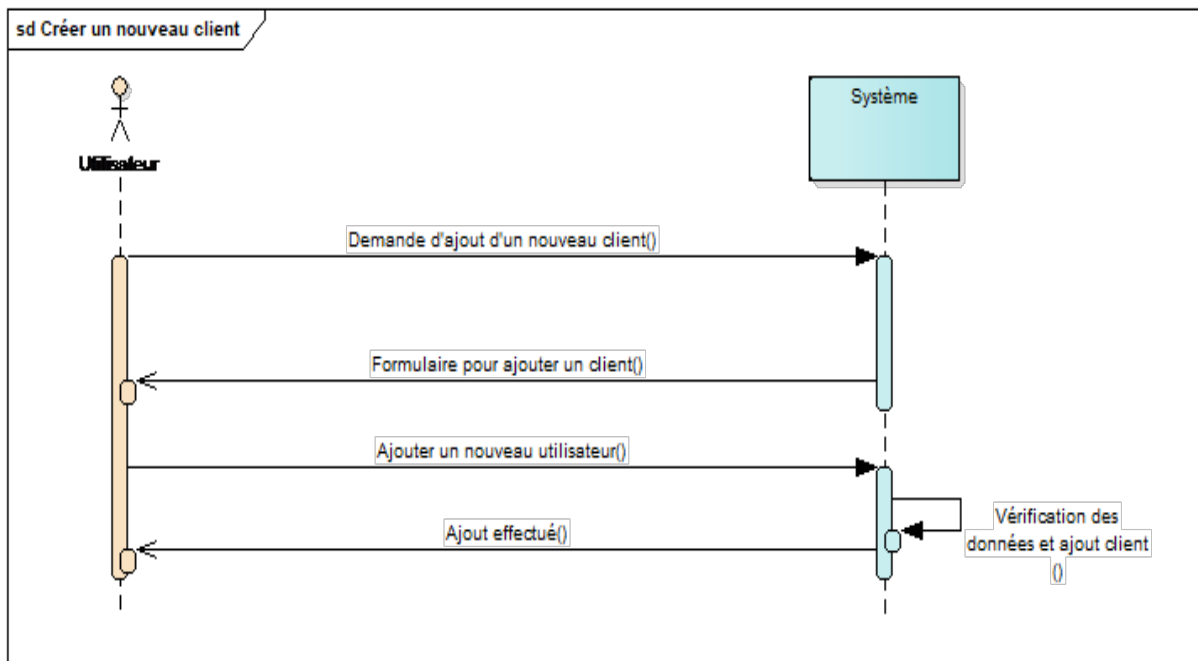


FIGURE 2.23 – Diagramme de séquence « Nouveau utilisateur »

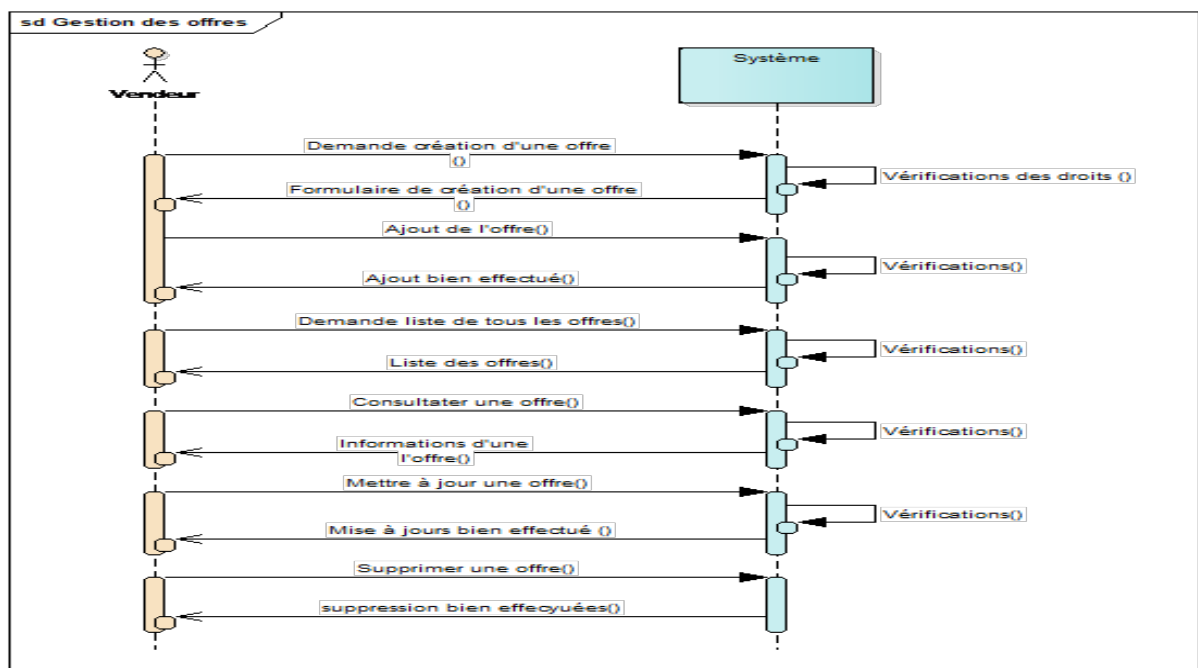


FIGURE 2.24 – Diagramme de séquence « Gestion offre par vendeur »

Chapitre 3

Réalisation

Dans ce chapitre nous allons décrire les différentes technologies adoptées et utilisées pour la réalisation de ce projet, nous expliquerons éventuellement nos choix techniques relatif aux langages de programmation, donc nous avons eu recours aux éléments suivants.

3.1 Choix de la plateforme JavaEE

3.1.1 Contexte générale

Les entreprises recherchent, de plus en plus, des solutions qui permettent la diminution des coûts et la réduction du temps de réponse, tout en respectant les standards de qualité logicielle.

Typiquement, les applications qui répondent à ces besoins doivent combiner entre les systèmes d'information existants et les nouvelles fonctions métiers qui apportent des services à un large intervalle des utilisateurs. Ces services doivent être :

- Fortement disponibles, pour répondre aux besoins actuels de l'environnement métier.
- Sécurisés, pour protéger les utilisateurs et l'intégrité des données d'entreprise.
- Fiables et extensibles, pour s'assurer que les transactions sont exécutées d'une manière exacte et prompte.

3.1.2 La technologie JavaEE

La technologie Java Entreprise Edition (JavaEE) définit la norme de développement des applications d'entreprise distribuées et multi tiers. Elle permet de simplifier les applications d'entreprise en les basant sur des composants modulaires normalisés tout en fournissant un ensemble complet de services et en gérant automatiquement un grand nombre de détails sur le comportement des applications.

Elle est composée de plusieurs technologies qui sont regroupées en trois catégories composantes, service et communication.

3.1.3 L'Architecture JavaEE

La plateforme Java Entreprise Edition est une norme qui vise à définir un standard de développement d'applications d'entreprises, et qui permet de réduire de manière significative le coût et la complexité du développement, du déploiement et la gestion de ces applications.

C'est une architecture promue par SUN® MICROSYSTEMS qui définit et fournit un modèle de programmation multi tiers basé sur des composants. Elles permettent de simplifier les applications d'entreprise en les basant sur des composants modulaires normalisés tout en fournissant un ensemble complet de services et en gérant automatiquement un grand nombre de détails sur le comportement des applications. Avec ce modèle, des applications légères peuvent interagir facilement avec le système d'information et les composants implantant la logique d'entreprise sur des serveurs d'applications.

L'architecture JavaEE consiste en trois parties :

- **Components** : les composants qui prennent en charge la présentation et la logique métier.
- **Containers** : fournit un contexte aux composants.
- **Connectors** : fournit l'accès aux sources de données de l'entreprise.

3.1.4 Composants de base de JavaEE

- **Les servlets**

Une servlet est un programme Java qui tourne sur la machine où est installé le serveur JEE, et qui est invoqué lorsqu'un navigateur client appelle l'URL liée à ce programme.

Les Servlets HTTP s'exécutent sur un serveur Web compatible Java pour en étendre les fonctionnalités. Les Servlets sont des programmes qui utilisent l'interface API Servlet Java, ainsi que les méthodes et les classes associées.

Les Servlets présentent de nombreuses fonctions, par exemple :

- Créer et retourner une page HTML entière comportant un contenu dynamique basé sur la nature de la requête client.
- Communiquer avec d'autres ressources du serveur, y compris les bases de données et les applications Java.
- Fournir un traitement personnalisé à n'importe quelle routine standard du serveur.
- Filtrer des données par type MIME pour des traitements spéciaux, tels que la conversion d'image et la fonction SSI (Server-Side Includes).
- Gérer les connexions à plusieurs clients en acceptant leurs données d'entrée et en leur diffusant les données de sortie.

- **Java Server Pages (JSP) :**

JSP est Une technologie Java qui permet la génération de pages web dynamiques. Les pages JSPs permettent de séparer efficacement le codage HTML de la logique applicative des pages Web. Elles sont utilisées pour accéder à des composants réutilisables, tels que des Servlets, des JavaBeans et des applications compatibles avec le langage Java. Elles prennent également en charge l'intégration de code Java en ligne dans les pages Web.

Il est possible d'accéder aux fichiers JSP de la manière suivante :

- La requête est envoyée à une Servlet qui génère du contenu dynamique et appelle un fichier JSP afin qu'il envoie le contenu au navigateur. Le schéma suivant illustre ce modèle d'accès.

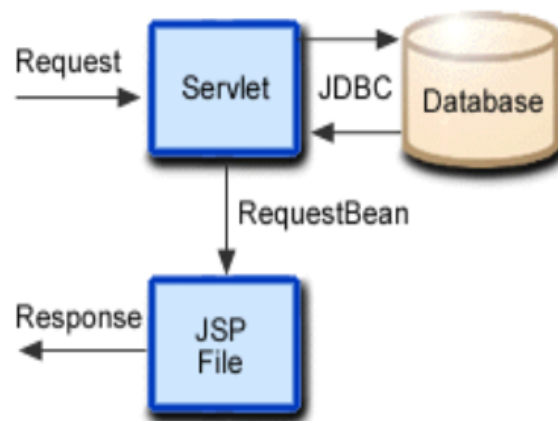


FIGURE 3.1 – Présentation du modèle d'accès aux fichiers JSP

3.2 Architecture applicative

Concevoir une application professionnelle nécessite obligatoirement une architecture basée sur le principe de la séparation des couches. Cela garantit une bonne identification des tâches, une simplification de la maintenance et un avancement simultané au niveau de toutes les couches tout en ayant un résultat cohérent.

Dans cette application, on compte trois couches principales, à savoir, Présentation, Métier et Persistance.

3.2.1 Couche présentation

C'est la partie Interface Homme Machine qui représente la couche frontale de l'application, et qui sera présentée aux utilisateurs finaux de cette dernière. Elle doit être riche, ergonomique et multilingue.

Cette couche est adoptée principalement pour gérer l'aspect visuel des applications et pour gérer les interactions avec les utilisateurs. Chargée de dessiner les fenêtres et

les autres composants graphiques, elle intercepte les événements et appelle la couche Application. De plus, elle vérifie les autorisations des utilisateurs auprès du service de sécurité.

Les rôles de la couche Présentation sont comme suit :

- Gestion du domaine de l'écran.
- Affichage des pages Web.
- Gestion de l'interaction avec l'utilisateur.
- Validation syntaxique.
- La couche Présentation établit deux relations avec la couche Application.
- Demandes de contenu à la couche Application pour affichage et édition.
- Ordres de traitement à la couche Application pour création, mise à jour, suppression, etc.

3.2.2 Couche métier

Comme son nom l'indique, cette couche implémente la logique Métier décrite dans les spécifications fonctionnelles. C'est l'intermédiaire entre la couche citée auparavant et la couche Persistance.

Cette couche est concentrée sur le métier de l'application, c'est-à-dire les règles métier, sémantique et logiques. C'est l'espace où réside le modèle du domaine. Sa charge principale consiste à garantir la validation sémantique de l'information métier. Cette couche sera basée sur le modèle objet.

Les responsabilités et les relations de la couche métier sont :

- Comportement métier.
- Validation sémantique.
- Echange de l'état des composants entités métier avec la couche persistance.

3.2.3 Couche persistance

Cette couche est certainement l'une des plus importantes. C'est ici que l'on trouve les fonctionnalités de base qui permettent de créer, rechercher et supprimer des entités métier dans le respect des propriétés transactionnelles. C'est également dans cette couche que les mécanismes de conversion objet/relationnel peuvent prendre place.

Le rôle de la couche Persistance est défini comme suit :

- Assurer les services basiques de création, lecture, mise à jour et suppression.
- Permettre la conversion Objet/Relationnel.

Ainsi, l'architecture choisie est basée sur la logique des couches. Chaque couche fournit aux autres couches des services et utilise les services des autres.

3.3 Boîte à outils

3.3.1 IDE eclipse



FIGURE 3.2 – Eclipse logo

Eclipse est un Environnement de développement Intégré (EDI) développé par IBM, il permet de développer les applications se basant sur l'Architecture JEE. Cet EDI permet le déploiement des applications dans les serveurs d'applications. Eclipse est composé de plusieurs « plug-ins », chaque plug-in permet l'ajout d'une fonctionnalité à Eclipse (Exemple le plug-in « Lombok » permet à Eclipse de supporter les fichiers « JSP »).

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment au moyen d'un environnement de développement intégré) mais aussi de modélisation, de conception, de test et de reporting.

Son environnement de développement intégré vise à supporter tout langage de programmation.

3.3.2 XAMPP



FIGURE 3.3 – XAMPP logo

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : OpenSSL, Expat (parseur XML) PNG, SQLite, zlib, etc. Ainsi que différents modules Perl et Tomcat.

3.3.3 Apache Tomcat 9.0



FIGURE 3.4 – Tomcat logo

Apache Tomcat est un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. Il implémente les spécifications des servlets et des JSP du Java Community Process⁶, 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.

De plus, les version Tomcat 9.x sont annoncés par la fondation Apache :

- Implémente Servlet 4.0 et WebSocket 1.1
- Support de Java SE 8 et +.
- Améliorations de la stabilité de nombreuses API (javax, EL 3.0, Jasper 2, JSP 2.3...)

3.3.4 Entreprise architecte



FIGURE 3.5 – Enterprise Architect logo

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems.

Enterprise Architect permet le développement d'applications selon le schéma d'architecture orientée modèle ainsi que le schéma d'architecture orientée services.

Enterprise Architect couvre tous les aspects du cycle de développement d'applications depuis la gestion des exigences, en passant par les phases de conception, la construction, tests et maintenance. Ces aspects sont appuyés par des fonctions de support tels que la traçabilité, la gestion de projet, ou encore le contrôle de version.

Le produit est destiné aux analystes, développeurs, architectes, urbanistes de toutes structures : de petites et moyennes entreprises aux multinationales, ainsi que les organisations gouvernementales.

3.3.5 Github



FIGURE 3.6 – Git Bash logo

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Les logiciels de gestion de versions sont des logiciels qui permettent de sauvegarder toutes les modifications apportées sur un fichier. Ce qui est utile lorsque plusieurs personnes travaillent sur un même fichier. On peut ainsi voir qui a fait telle ou telle modification et quand elle a été faite. À tout moment, il est possible de revenir à une version ultérieure du fichier, afin de corriger d'éventuelles erreurs par exemple.

3.4 Langages, technologies et API utilisées

3.4.1 HTML5



FIGURE 3.7 – HTML logo

Est un langage informatique utilisé sur internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage.

3.4.2 CSS3



FIGURE 3.8 – CSS logo

Les CSS permettent de définir un aspect graphique réutilisable pour des éléments d'une page Web. Dans une application Ajax, l'interface utilisateur peut être modifiée dynamiquement grâce à ces feuilles de style.

Le concept de feuilles de styles repose sur le principe de la séparation du contenu de la présentation, dans l'élaboration de documents basés sur le HTML.

3.4.3 JavaScript



FIGURE 3.9 – javascript logo

JS est un langage de programmation de scripts basé sur la norme ECMAScript. Il s'insère dans le code HTML d'une page web, ou séparément dans un fichier .js et permet d'augmenter considérablement le spectre des possibilités grâce à sa robustesse, sa facilité et le grand nombre de ses bibliothèques.

Ce langage de programmation OO, dont la syntaxe est simple, est faiblement typé, et exécuté généralement côté client par le navigateur web. Il est utilisé pour interagir avec le DOM, réaliser des services dynamiques, ou simplement pour des fins ergonomiques.

Les avantages du JavaScript sont nombreux :

- **Vitesse** : Les fonctions du JavaScript ne doivent pas attendre pour des réponses de leurs serveurs pour agir, ce qui accélère l'ouverture des sites web.
- **Simplicité** : le JavaScript est relativement simple et facile à apprendre.

- **Versatilité** : Le JavaScript ne nécessite pas un programme spécial pour l'interpréter (Flash Player, "plug-ins"), ni pour l'écrire. De plus, JavaScript n'occupe pas un grand espace sur les sites web.

3.4.4 UML



FIGURE 3.10 – UML logo

Nous avons choisi de travailler avec UML parce qu'il exprime mieux la vue statique et dynamique du système d'information et, pour notre application Web, il est nécessaire de faire une analyse très approfondie pour pouvoir dégager les nécessités de développement ainsi que quelques scénarios d'exécution.

3.4.5 Bootstrap



FIGURE 3.11 – Bootstrap logo

Est un Framework utilise les éléments HTML et propriétés CSS pour ajouter des styles optimisés a une application web. Il permet de gérer le contenu interactif d'une application web.

3.4.6 JQuery



FIGURE 3.12 – jQuery logo

C'est une bibliothèque JavaScript contenant un ensemble cohérent de fonctions qui permettent le parcours et la modification du DOM, la gestion des événements, la création des effets visuels et les appels AJAX. Ces fonctions sont conçues pour minimiser le code JS sans nuire à la productivité « write less, do more ».

3.5 Modèle MVC2

3.5.1 Définition

Le Model-View-Controller (MVC2) est un modèle de conception logicielle très répandu et fort utile. Aujourd'hui il est fortement recommandé dans l'univers JEE.

Créé dans les années 80 par Xerox PARC pour Smalltalk-80, il est indépendant du langage de programmation.

La différence entre MVC et MVC2 c'est que ce dernier il a un seul contrôleur qui se charge de rediriger la requête vers le bon traitement.

L'application se retrouve segmentée en trois composants essentiels :

- Le **modèle** représente les données et les règles métiers. C'est dans ce composant que s'effectuent les traitements liés au coeur du métier. Les données peuvent être liées à une base de données, des EJBs, des services Web, ... Il est important de noter que les données sont indépendantes de la présentation. En d'autres termes, le modèle ne réalise aucune mise en forme. Ces données pourront être affichées par plusieurs vues. Du coup le code du modèle est factorisé : il est écrit une seule et unique fois puis réutilisé par chaque vue.
- La **vue** correspond à l'IHM. Elle présente les données et interagit avec l'utilisateur. Dans le cadre des applications Web, il s'agit d'une interface HTML, mais n'importe quel composant graphique peut jouer ce rôle.
- Le **contrôleur**, quant à lui, se charge d'intercepter les requêtes de l'utilisateur, d'appeler le modèle puis de rediriger vers la vue adéquate. Il ne doit faire aucun traitement. Il ne fait que de l'interception et de la redirection.

3.5.2 Avantages

Plusieurs vues peuvent utiliser le même modèle.

- Le modèle est totalement indépendant de la vue. Si l'application a besoin d'un nouveau mode d'accès, le modèle restera inchangé. Il suffit juste de changer la partie IHM.
- Le modèle étant totalement autonome, il peut être modifié beaucoup plus facilement.

3.5.3 Architectures MVC2

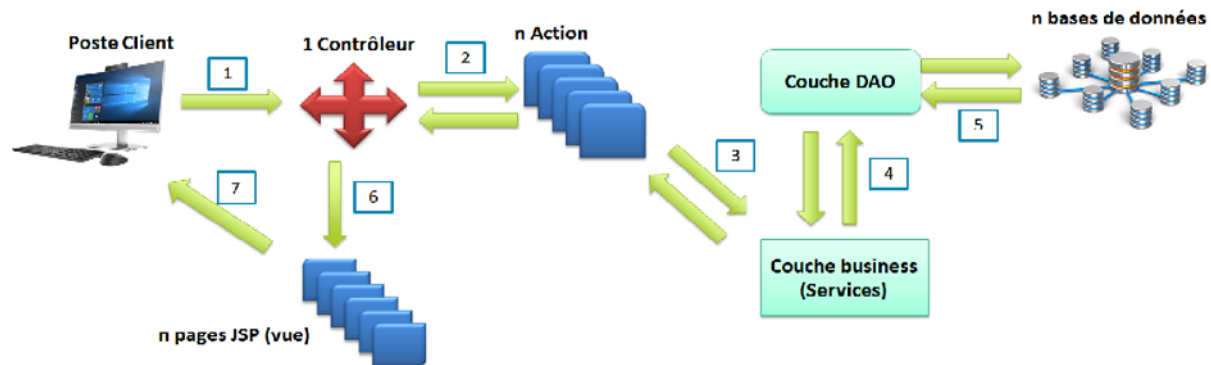


FIGURE 3.13 – Architecture MVC2

3.6 Mise en œuvre

Notre application se scinde en trois parties : une partie pour les clients, une deuxième pour les propriétaires et une troisième pour l'administrateur. Ces parties vont être détaillées dans ce qui suit.

3.6.1 Accueil

- Accueil

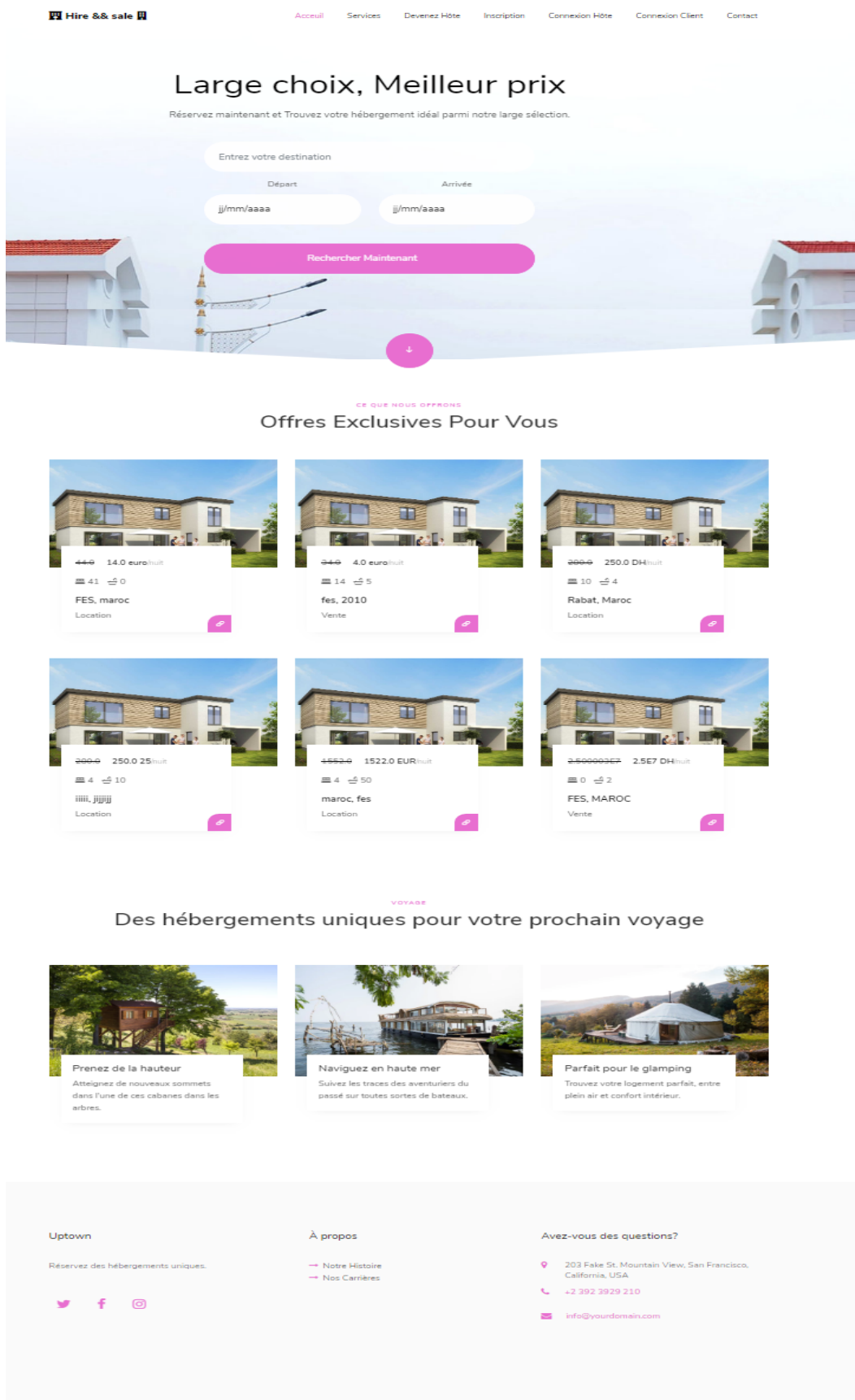




FIGURE 3.14 – Interface accueil

• Inscription

 Hire & sale 

[Accueil](#)

[Services](#)

[Devenez Hôte](#)

[Inscription](#)

[Connexion Hôte](#)

[Connexion Client](#)

[Contact](#)

— CRÉER VOTRE COMPTE —

Remplir le formulaire suivant pour créer votre compte

Prénom *	Nom *
<input type="text" value="Prénom"/>	<input type="text" value="Nom"/>
Date de naissance *	Numéro CIN *
<input type="text" value="jj/mm/aaaa"/>	<input type="text" value="CD5...."/>
Sexe *	Email *
<input type="text" value="Homme"/>	<input type="text" value="email@gmail.com"/>
Mot de passe *	Confirmer le mot de passe *
<input type="password" value="*****"/>	<input type="password" value="*****"/>
Pays / Région *	Ville
<input type="text" value="Maroc"/>	<input type="text" value="Fés"/>
Numéro de téléphone *	
<input type="text" value="065568...."/>	
<input type="button" value="Inscription"/>	

Uptown


Réservez des hébergements uniques.



À propos

[→ Notre Histoire](#)
[→ Nos Carrières](#)

Avez-vous des questions?

 203 Fake St. Mountain View, San Francisco, California, USA

 +2 392 3929 210

 info@yourdomain.com

FIGURE 3.15 – Interface inscription

- Connexion

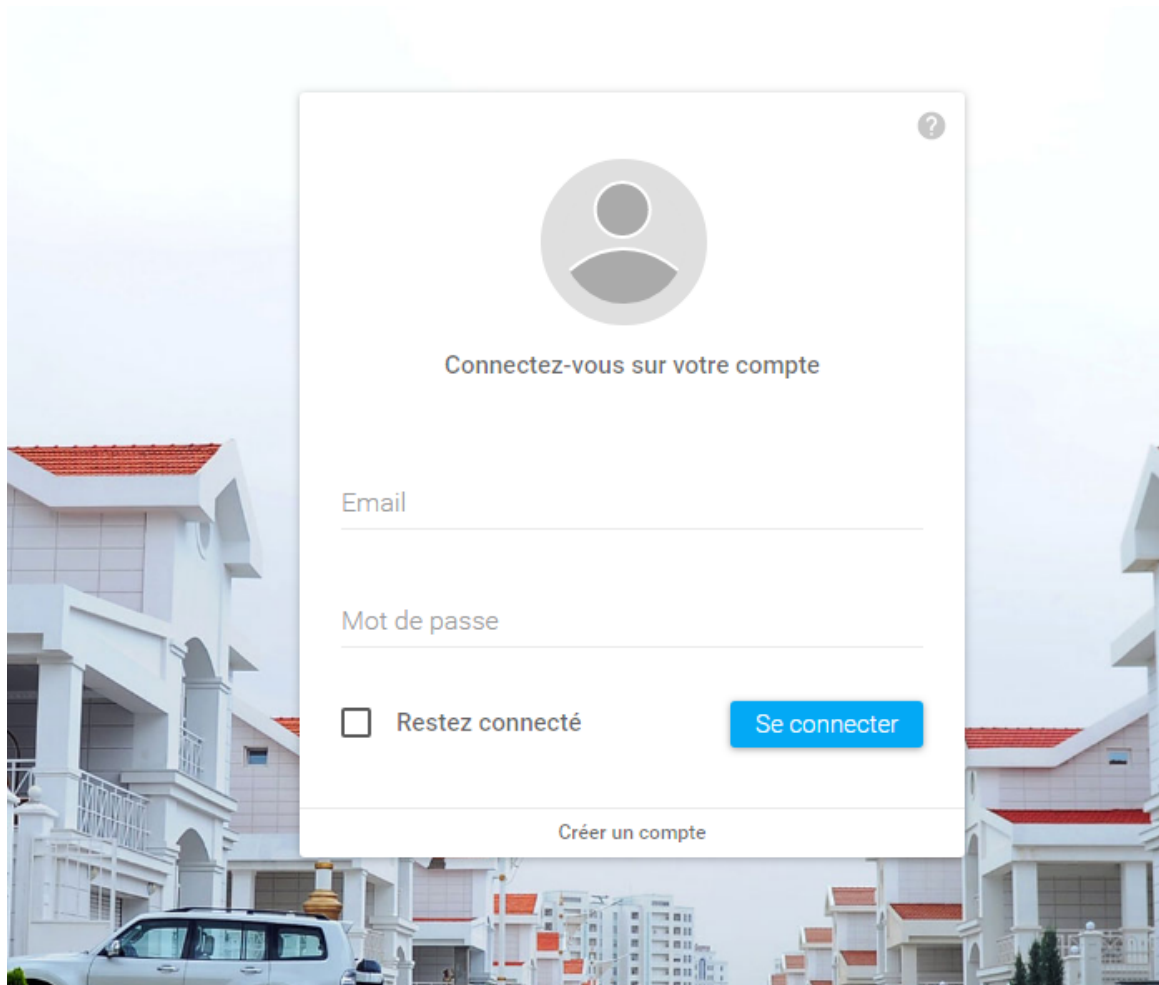


FIGURE 3.16 – Interface connexion

- **Contact**

The screenshot shows a contact page for a company named 'Hire & sale'. The page has a header with a logo and navigation links: Accueil, Services, Devenez Hôte, Inscription, Connexion Hôte, Connexion Client, and Contact. The main heading is 'Contactez-nous'. Below this, there are three columns of contact information: Address (Ecole Nationale Supérieure d'informatique et d'analyse des systèmes), Phone (+212 6 55 68 12), and Email (info@hireSale.com). A message prompt asks if the user has questions and encourages them to send a message. Below the prompt is a form with fields for 'Votre Nom', 'Votre Email', 'Sujet', and 'Message', followed by an 'Envoyer message' button. The footer contains three sections: 'Uptown' with social media links, 'À propos' with links to 'Notre Histoire' and 'Nos Carrières', and 'Avez-vous des questions?' with contact details for a San Francisco office.

Hire & sale

Accueil Services Devenez Hôte Inscription Connexion Hôte Connexion Client **Contact**

Contactez-nous

Address:
Ecole Nationale Supérieure
d'informatique et d'analyse des
systèmes

Phone:
+ 212 6 55 68 12

Email:
info@hireSale.com

Si vous avez des questions
N'hésitez pas à nous envoyer un message

Votre Nom

Votre Email

Sujet

Message

Envoyer message

Uptown
Réservez des hébergements uniques.

À propos
→ Notre Histoire
→ Nos Carrières

Avez-vous des questions?
203 Fake St. Mountain View, San Francisco,
California, USA
+2 392 3929 210
info@yourdomain.com

FIGURE 3.17 – Interface contact

3.6.2 Espace client

- Menu client

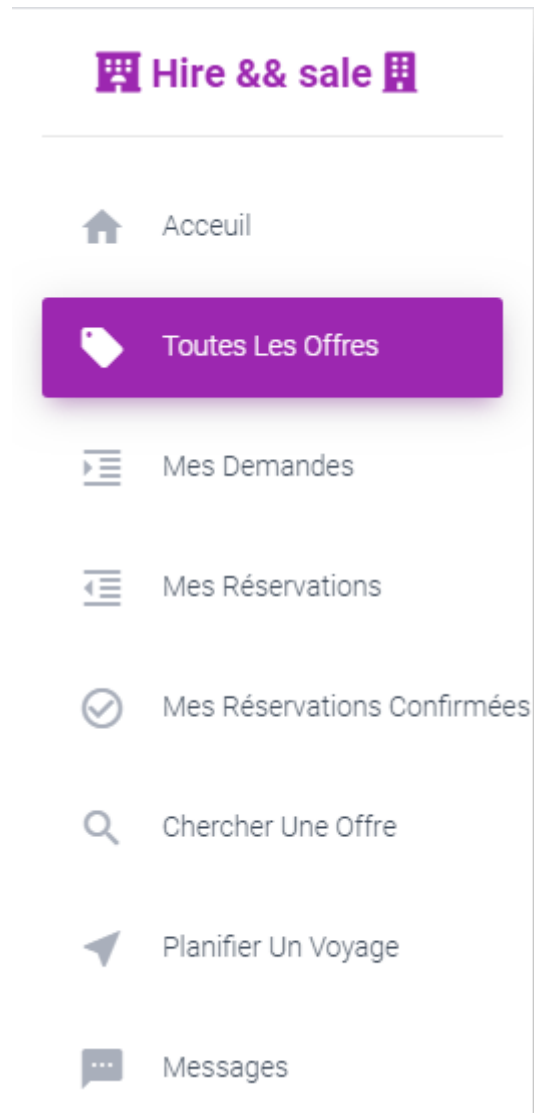


FIGURE 3.18 – Interface "Menu client"

- Toutes les offres

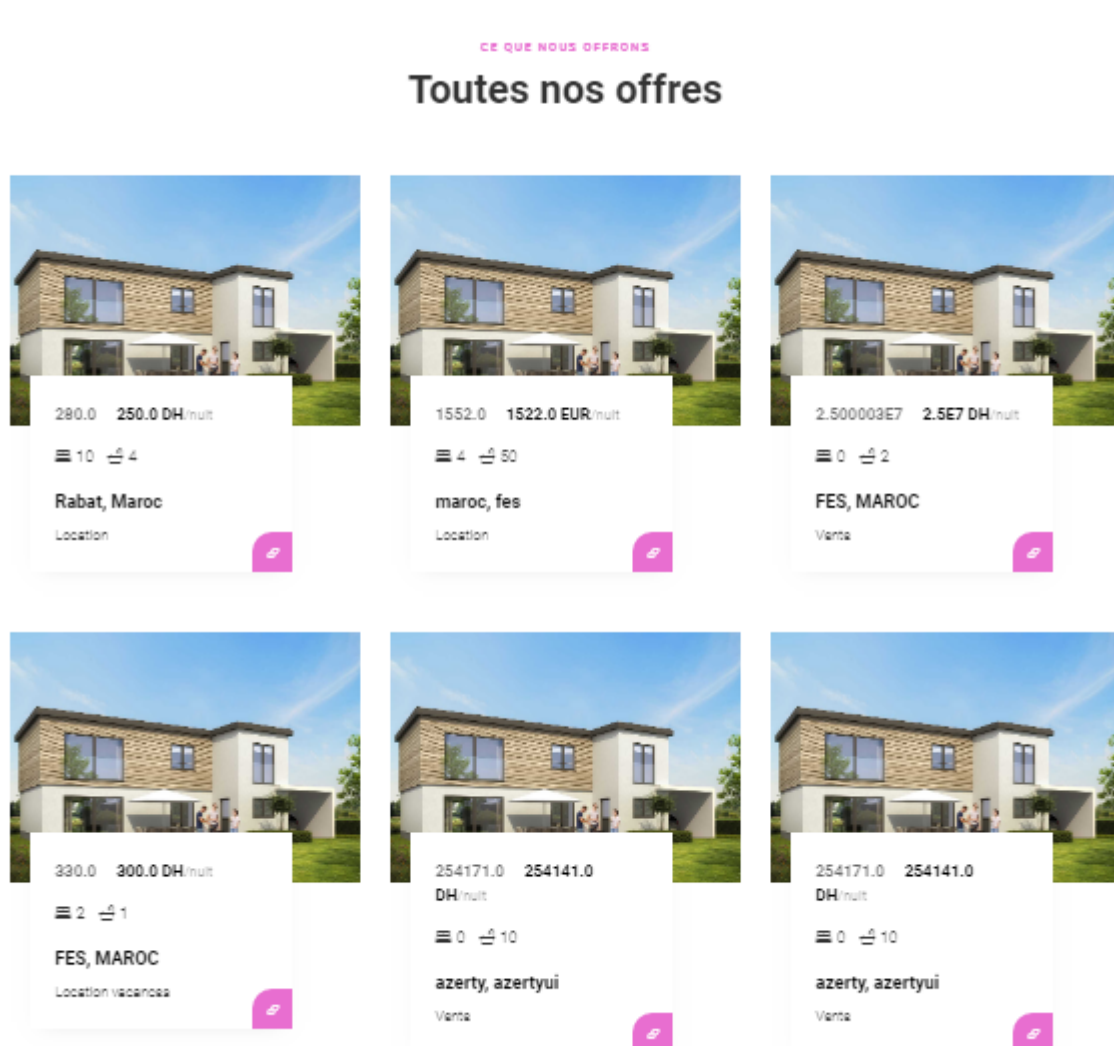


FIGURE 3.19 – Interface "Liste des tous les offres"

- Listes des réservations

MES RÉSERVATIONS						
Demandes Achat / Location traitées						
Demandes de location						
Liste des demandes de location que vous avez effectuées						
N° Demande	Date de la demande	Date Début	Date fin	Statut	Offre	Action
9	2020-02-05	2020-11-11	2020-12-21	ACCEPTÉE		CONFIRMÉE
10	2020-02-05	2020-11-11	2020-12-21	NON ACCEPTÉE		
11	2020-02-05	2015-02-11	2020-02-20	ACCEPTÉE		CONFIRMÉE
12	2020-02-05	2019-10-10	2020-10-10	NON ACCEPTÉE		
13	2020-02-05	2019-10-10	2020-10-10	ACCEPTÉE		CONFIRMÉE
53	2020-02-07	2020-02-02	2020-03-15	ACCEPTÉE		CONFIRMATION
Demandes d'achat						
Liste des demandes d'achat que vous avez effectuées						
N° Demande	Date de la demande	Statut	Offre	Action		
8	2020-01-29	ACCEPTÉE		CONFIRMÉE		
20	2020-02-02	ACCEPTÉE		CONFIRMÉE		
21	2020-02-02	NON ACCEPTÉE				
22	2020-02-05	ACCEPTÉE		CONFIRMÉE		
23	2020-02-05	ACCEPTÉE		CONFIRMÉE		
51	2020-02-07	ACCEPTÉE		CONFIRMATION		

FIGURE 3.20 – Interface "Liste des réservations"

- Liste des réservations confirmées

MES RÉSERVATIONS

Demandes Achat / Location traitées

Demandes de location

Liste des demandes de location que vous avez effectuées

N° Demande	Date de la demande	Date Début	Date fin	Statut	Offre	Action
9	2020-02-05	2020-11-11	2020-12-21	ACCEPTÉE		CONFIRMÉE
11	2020-02-05	2015-02-11	2020-02-20	ACCEPTÉE		CONFIRMÉE
13	2020-02-05	2019-10-10	2020-10-10	ACCEPTÉE		CONFIRMÉE

Demandes d'achat

Liste des demandes d'achat que vous avez effectuées

N° Demande	Date de la demande	Statut	Offre	Action
8	2020-01-29	ACCEPTÉE		CONFIRMÉE
20	2020-02-02	ACCEPTÉE		CONFIRMÉE
22	2020-02-05	ACCEPTÉE		CONFIRMÉE
23	2020-02-05	ACCEPTÉE		CONFIRMÉE

FIGURE 3.21 – Interface "Liste des réservations confirmées"

- Chercher une offre

Chercher l'offre qui vous plaît

Recherche multicritères et critères spécifiques

☐ Ville ☐ Pays ☐ Nombre chambre

☐ Location ☐ Vente

Search ...

CHERCHER

Recherche par date de disponibilité

Date début : jj/mm/aaaa Date fin : jj/mm/aaaa

CHERCHER

FIGURE 3.22 – Interface pour chercher une offre

3.6.3 Espace vendeur

- Menu Vendeur

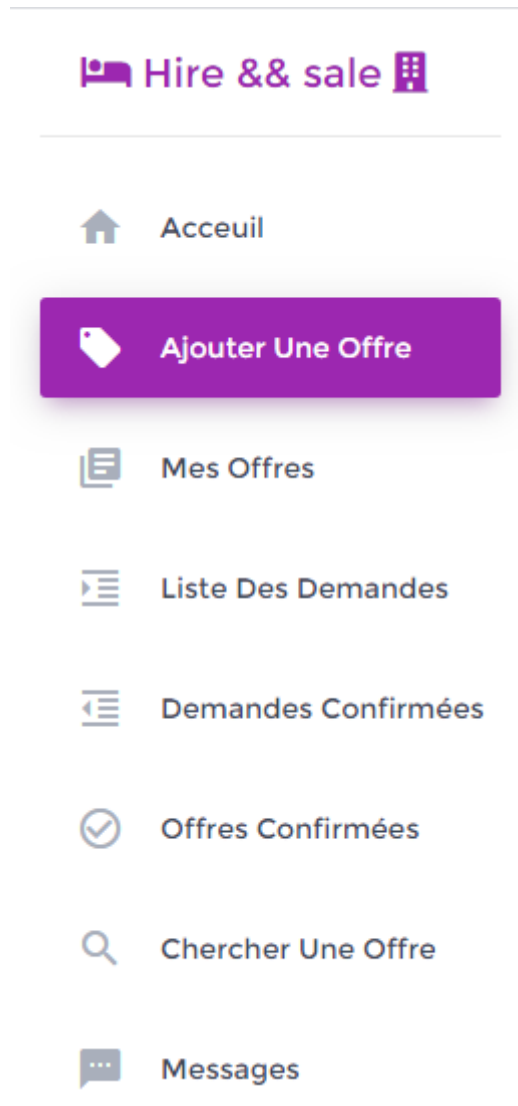


FIGURE 3.23 – Menu vendeur

- Créer une nouvelle offre

1 CATEGORIE 2 EMPLACEMENT 3 INFORMATIONS 4 DESCRIPTION

CATEGORIE DE L'IMMOBILIER

Choisissez la catégorie de votre immobilier

Transaction * :

☒ Vente ☐ Location ☐ Location vacances

Catégorie * :

☒ Appartement ☐ Maison ☐ Villas & maisons de luxe

☐ Riad

Next

FIGURE 3.24 – Interface pour créer une offre

- Liste des offres créées
































<div>Liste des offres</div> <div>la liste de tous vos offres effectuées</div>								
Catégorie	Vente/Location	Pays	Ville	Adresse	Date Début	Date Fin	Etat	Action
chambreHauberge	Location	maroc	FES	ista hay adarissa fes	2010-10-10	2020-02-10	CONFIRMÉE	
chambreHote	Vente	2010	fes	145	2010-10-10	2020-12-11	CONFIRMÉE	
chambreHote	Location	Maroc	Rabat	Ista hay adarissa fes	2020-01-30	2020-03-20	EN ATTENTE	  
Chambres	Location	jijijij	iiiiii	iiiiii	2010-10-10	2018-10-10	EN ATTENTE	  
Maison	Location	fes	maroc	zertyu	2020-10-10	2020-10-30	EN ATTENTE	  
Chambres	Location	maroc	fes	bab jyaaf fes	2010-10-10	2020-02-20	CONFIRMÉE	
Maison	Vente	azertyui	azerty	azertyuio	----	----	EN ATTENTE	  
Maison	Vente	azertyui	azerty	azertyuio	----	----	CONFIRMÉE	
Chambres	Location	azertyu	azertyu	azertyuio	2019-10-10	2020-10-10	EN ATTENTE	  
Maison	Location	Maroc	Rabat	Agdal Rabat	2020-10-10	2020-12-31	EN ATTENTE	  
Maison	Location	Maroc	Rabat	Agdal Rabat	2020-10-10	2020-12-31	EN ATTENTE	  
Maison	Location	Maroc	Rabat	Agdal Rabat	2020-10-10	2020-12-31	EN ATTENTE	  
Maison	Location	MAROC	FES	adarisa	2020-10-10	2020-12-31	EN ATTENTE	  

FIGURE 3.25 – Interface "Liste des offres créées"

- Liste des demandes

Demande de Location							
Liste des demandes de location							
N° Demande	Date de la demande	Date Début	Nombre Nuits	Date fin	Statut	Client	Action
3	2020-01-28	2020-03-05	15	2020-03-20	EN ATTENTE		
6	2020-02-02	2015-11-11	1000	2018-08-07	EN ATTENTE		
7	2020-02-02	2011-11-11	3013	2020-02-10	EN ATTENTE		
8	2020-02-02	2011-11-11	3013	2020-02-10	EN ATTENTE		
11	2020-02-05	2015-02-11	1835	2020-02-20	ACCEPTÉE		
12	2020-02-05	2019-10-10	366	2020-10-10	NON ACCEPTÉE		
13	2020-02-05	2019-10-10	366	2020-10-10	ACCEPTÉE		
53	2020-02-07	2020-02-02	42	2020-03-15	ACCEPTÉE		

Demande d'Achat				
Liste des demandes d'Achat				
N° Demande	Date de la demande	Statut	Client	Action
8	2020-01-29	ACCEPTÉE		
10	2020-01-29	EN ATTENTE		
13	2020-01-29	EN ATTENTE		
16	2020-02-02	EN ATTENTE		
18	2020-02-02	EN ATTENTE		
19	2020-02-02	EN ATTENTE		
20	2020-02-02	ACCEPTÉE		
21	2020-02-02	NON ACCEPTÉE		
23	2020-02-05	ACCEPTÉE		
51	2020-02-07	ACCEPTÉE		

FIGURE 3.26 – Interface "Liste des demandes"

3.6.4 Espace administrateur

- Menu Administrateur

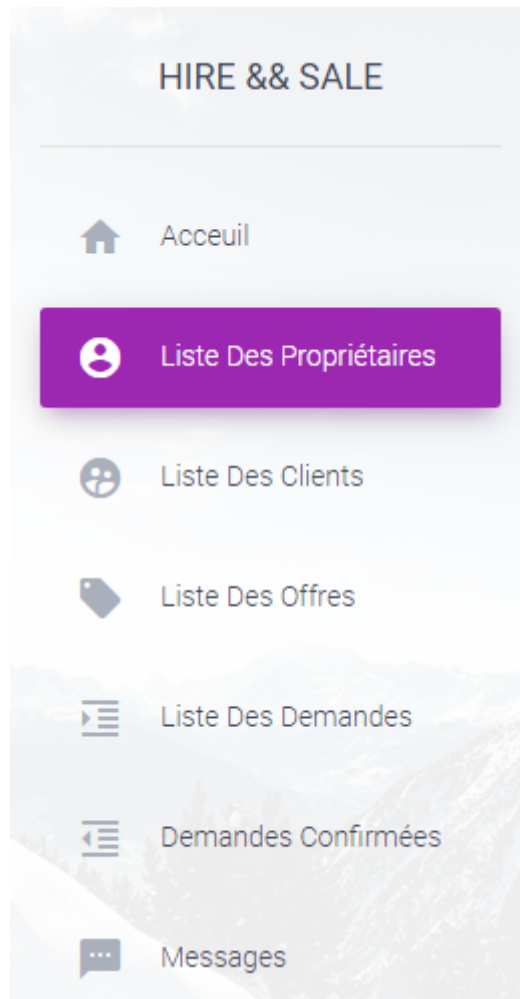


FIGURE 3.27 – Menu administrateur

- Liste des clients





































Clients						
la liste de tous les clients						
Nom	Prénom	Date de naissance	Sexe	Pays	Téléphone	Action
CHADDAD	Zineb	20/07/1995		italie	0622145441	 
Fadil	Zakaria	16/11/1989	Homme	Maroc	0615478954	 
CHADDAD	Zineb	20/07/1995		italie	0622145441	 
AMRANE	Taha	22/07/1988		italie	0655417889	 
AMJAD	Ayoub	10/06/2001		canada	0621214747	 
BAHSSINE	Mohammed	05/09/2000		italie	0622145441	 
CHADDAD	Zineb	20/07/1995		italie	0622145441	 
AMRANE	Taha	22/07/1988		italie	0655417889	 
AMJAD	Ayoub	10/06/2001		canada	0621214747	 
BAHSSINE	Mohammed	05/09/2000		italie	0622145441	 
BAHSSINE	Youssra	17/07/1996		italie	0674896523	 
IBRAHIMI	Abdellah	20/05/1969		cor�e du sud	0652521452	 
BAHI	Btissam	17/05/1986		canada	0655417889	 
ALAOUI	Asmae	17/05/1986		italie	0655417889	 
CHADDAD	Hamza	22/07/1988		italie	0621214747	 
GHAZLANE	Badr	05/09/2000		france	0655417889	 
BAHI	Ayoub	22/05/1964		italie	0622145441	 
Client	Ilham	14/08/1997	Femme	maroc	065781247	 

FIGURE 3.28 – Interface "Liste des clients"

- Liste des vendeurs





















Propriétaires						
la liste de tous les propriétaires						
Nom	Prénom	Date de naissance	Sexe	Adresse	Téléphone	Action
ghazlane	mohammed	2020-12-31	Femme	azertyuiop	0655681244	 
ghazlane	mohammed	2019-12-30	Femme	Ista	0655681244	 
ghazlane	mohammed	2019-12-30	Femme	Ista	0655681244	 
taha	mohammed	2010-10-10	Homme	azertyui	azer	 
ghafa2il	mohammed	2010-10-10	Homme	azertyuiop	0655681244	 
mohammed	mohammed	2020-12-31	Femme	azertyuio	azertyu	 
mohammed	mohammed	2010-10-10	Homme	azertyuio	azerty	 
mohammed	mohammed	2020-12-31	Femme	azerty	azerty	 
MOHAMMED	GHAZLANE	1997-11-03	Homme	azertyuio	zerty	 
Ghazlane	Mohammed	1997-11-03	Homme	Ista hay adarissa Fes	0655681244	 

FIGURE 3.29 – Interface "Liste des propriétaires"

Conclusion

Au cours de la réalisation du projet, nous avons eu l'opportunité de mettre en œuvre différentes connaissances acquises, d'améliorer nos connaissances en JavaEE et d'acquérir de nouveaux outils de développement tels que l'architecture MVC, JAVA (orienté objet), HTML, CSS et JavaScript.

Notre travail s'est fixé comme objectifs de satisfaire le maximum des besoins de l'application et faciliter les tâches aux clients, propriétaires et administrateur.

Les difficultés majeures qu'on a rencontrées résident essentiellement dans la nouveauté des technologies avec lesquelles on a travaillé et la contrainte du temps pour les maîtriser. Le résultat est donc un site web qui va faciliter la gestion de location et vente des hébergements.

Comme perspectives on peut envisager par la suite à ajouter de nouvelles fonctionnalités comme l'amélioration du rendu visuel, l'amélioration de la sécurité de l'application et l'ajout de la gestion de paiement.

Bibliographie

- [1] UML, diagramme de cas d'utilisation, [En ligne], Disponible sur : <http://www.uml-sysml.org/>
- [2] W3, Site officiel de W3, [En ligne], Disponible sur : <http://www.w3.org/TR/soap/>
- [3] Eclipse, Site officiel d'Eclipse, [En ligne], Disponible sur : <http://www.eclipse.org/>
- [4] BootStrap, Site officiel , [En ligne], Disponible sur : <https://getbootstrap.com/>
- [5] EnterpriseArchitects, Site officiel , [En ligne], Disponible sur : <http://enterprisearchitects.com/>
- [6] Apache Tomcat, Site officiel , [En ligne], Disponible sur : <http://tomcat.apache.org/>
- [7] developpez, site officiel, [en ligne] disponible sur <https://www.developpez.net>
- [8] lebigdata, site officiel, [en ligne], disponible sur <https://www.lebigdata.fr>
- [9] openclassrooms, site officiel, [en ligne], disponible sur <http://openclassrooms.com/>
- [10] infowebmaster, site officiel, [en ligne], disponible sur <http://www.infowebmaster.fr/>
- [11] stackoverflow, site officiel, [en ligne], disponible sur <https://stackoverflow.com/>