****



IIR5

INGÉNIERIE INFORMATIQUE&RÉSEAUX

**ANNÉE UNIVERSITAIRE**

**2021 / 2022**

**OPTION**

MÉTHODES INFORMATIQUES APPLIQUÉES À LA GESTION DES ENTREPRISES (M.I.A.G.E)

# PROJET DE STAGE D’ÉTÉ THÈME

**Conception et réalisation d’une application web pour la gestion locative**

RÉALISÉ PAR HARMOUCH YASSIN

ENCADRÉ PAR Mr. AATILA MUSTAPHA

SOUTENU LE 25 décembre 2021

# 

# Dédicace

Au nom de Dieu, le plus Clément, le plus Miséricordieux

Nous dédions ce projet en témoignage de notre gratitude et de tous nos respects à :

Nos parents pour toute l’affection, tous les sacrifices, les prières et les encouragements qu’ils n’ont pas cessé de nous prodiguer tout au long de nos études.

A nos familles, nos amis, et tous nos proches qui n’ont pas cessés de nous souhaiter bonheur et réussite.

A tous ceux qui nous ont soutenus tout au long de ce projet.

A notre encadrant Pr Mustapha Aatila, A nos enseignants.

# Remerciements

Nous voulons exprimer par ces quelques lignes de remerciements notre gratitude envers tous ceux en qui, par leurs présences, leurs soutiens, leurs disponibilités et leurs conseils, nous avons trouvé courage afin d’accomplir ce projet.

Nous tiendrons à remercier également nous familles de leurs soutiens et l’encourage pendant cette période qui a été difficile sur nous.

Nous commençons par remercier, toute l’équipe pédagogique de l’université marocaine des sciences de l’ingénierie de Marrakech et les intervenants professionnels responsables de la formation en Génie Informatique, pour avoir assuré la partie théorique de celle-ci.

Nous tenons à remercier, également, Monsieur Mustapha Aatila qui nous a fait l’honneur d’être notre encadrant. Nous le remercions profondément pour son encouragement continu et aussi d’être toujours disponible pour nous écouter, nous aider et nous guider à retrouver le bon chemin par sa sagesse et ses précieux conseils.

# Résumé

**Ce rapport fût le fruit de notre expérience pendant deux mois de développement d’un site web avec le Framework Laravel et la technologie Ajax.**

**Ce rapport décrit le projet que nous avons traité : « Création d’une application de gestion locative permettant de générer des réservations qu’on pourra visualiser et gérer à partir du web».**

**Notre projet s’agissait d’offrir deux espaces différents, un espace privé a l’admin pour qu’ils puissent gérer les différents aspects de sa villa avec une confidentialité totale.**

**Notre mission était de réaliser une application web fiable et évolutive pour la gestion locative : cas d’une villa.**

**Cette application permettra aux autres utilisateurs de communiquer et de partager des informations librement et avoir un environnement interactif entre ses utilisateurs.**

**L’objectif était de réaliser la conception et l’étude complète du sujet donné, de proposer une solution technique appropriée et d’assurer l’intégrité des données, la non-redondance de l’information, ainsi que la réduction des temps de traitements.**

**Le présent travail représente une solution qui va répondre au besoin de facilite les taches pour l’admin d’après un environnement de travail ordonner, organiser, sécuriser et accessible par les différents membres de notre site web.**

Tableau I la liste des acronymes

|  |  |
| --- | --- |
| Acronyme | Définition |
| BDD | **Base de données** |
| CRUD | **Create read update delete** |
| CSS3 | **Cascading Style Sheets 3** |
| DOM | **Document Object Model** |
| EMSI | **Université Marocaine des sciences de l’ingénierie** |
| JSON | **JavaScript Object Notation** |
| JS | **JavaScript** |
| HTML5 | **Hypertext Markup Language5** |
| MIT | **Massachusetts Institute of Technology** |
| MVC | **Modèle-Vue-Contrôleur** |
| PFE | **Projet de fin d’étude** |
| PHP | **Hypertext Preprocessor** |
| SQL | **Structured Query Language** |
| SGBD and Xml | **Système de gestion de base de données** |
| UML | **Unified Modelling Language** |
| WAMP | **Windows , Apache , MySQL, PHP** |
| Ajax | **Asynchronous JavaScript** |
| XML | **Extensible Markup Language** |

# Table des matières

Table des matières

[Dédicace 1](#_Toc87812525)

[Remerciements 3](#_Toc87812526)

[Résumé 4](#_Toc87812527)

[1 Table des matières 6](#_Toc87812528)

[Introduction générale 8](#_Toc87812529)

[2 CHAPITRE I 9](#_Toc87812530)

[2.1.1 Introduction : 10](#_Toc87812531)

[2.1.2 Objectif : 10](#_Toc87812532)

[2.1.3 Problématique : 10](#_Toc87812533)

[2.1.4 Solution : 10](#_Toc87812534)

[3 CHAPITRE II 11](#_Toc87812535)

[3.1 Introduction 12](#_Toc87812536)

[3.2 Cycle de vie de développement de projet 12](#_Toc87812537)

[3.3 Présentation du langage UML : 13](#_Toc87812538)

[3.4 Intérêt de la modélisation : 13](#_Toc87812539)

[3.5 Les avantages d’UML : 13](#_Toc87812540)

[4 Analyse Fonctionnelle : 15](#_Toc87812541)

[4.1 Définition : 15](#_Toc87812542)

[4.2 Règle de gestion 15](#_Toc87812543)

[4.2.1 Mode Réservations : 15](#_Toc87812544)

[4.2.2 Mode Clients : 15](#_Toc87812545)

[4.2.3 Mode Agence de location: 15](#_Toc87812546)

[4.2.4 Mode Facture : 15](#_Toc87812547)

[4.2.5 Mode Statistiques : 15](#_Toc87812548)

[4.2.6 Mode Villa : 15](#_Toc87812549)

[4.3 Les acteurs et leurs rôles : 15](#_Toc87812550)

[4.3.1 Admin ( propriétaire du bien ): 16](#_Toc87812551)

[4.3.2 Agence de location (Courtiers ): 16](#_Toc87812552)

[4.4 Les Scénarios : 16](#_Toc87812553)

[4.4.1 Définition : 16](#_Toc87812554)

[4.4.2 Acteur : Admin 16](#_Toc87812555)

[4.4.3 Acteur : Agence de location (Courtier): 18](#_Toc87812556)

[5 Les Diagrammes : 19](#_Toc87812557)

[Diagramme de cas d’utilisation : 19](#_Toc87812558)

[5.1.1 Définition : 19](#_Toc87812559)

[5.2 Diagramme de classe : 20](#_Toc87812560)

[5.2.1 Définition : 20](#_Toc87812561)

[5.3 Les diagrammes de séquences : 21](#_Toc87812562)

[5.3.1 Définition : 21](#_Toc87812563)

[5.3.2 Création d’un bien : 21](#_Toc87812564)

[5.3.3 Ajouter Réservation: 22](#_Toc87812565)

[5.3.4 Imprimer facture: 23](#_Toc87812566)

[5.4 Diagrammes d’états : 24](#_Toc87812567)

[5.4.1 Diagramme d’état de la gestion de réservation 24](#_Toc87812568)

[5.4.2 Diagramme d’état de la gestion de bien 24](#_Toc87812569)

[5.5 Conclusion 25](#_Toc87812570)

[6 CHAPITRE 3 26](#_Toc87812571)

[7 Choix technologique : 27](#_Toc87812572)

[7.1 Réalisation et mise en œuvre : 27](#_Toc87812573)

[Langages utilisés 27](#_Toc87812574)

[7.1.1 HTML5 27](#_Toc87812575)

[7.1.2 CSS3 : 27](#_Toc87812576)

[7.1.3 PHP 27](#_Toc87812577)

[7.1.4 JavaScript 27](#_Toc87812578)

[7.1.5 Ajax 27](#_Toc87812579)

[7.2 Logiciel utilisés : 28](#_Toc87812580)

[7.2.1 PowerAMC : 28](#_Toc87812581)

[7.2.2 larogon : 28](#_Toc87812582)

[7.2.1 Heidi SQL : 28](#_Toc87812583)

[7.2.2 visual studio code : 28](#_Toc87812584)

[7.3 Le patron de la conception : 28](#_Toc87812585)

[7.4 Framework utilisé : 30](#_Toc87812586)

[7.4.1 Laravel7 30](#_Toc87812587)

[7.5 L’architecture des dossiers du projet 30](#_Toc87812588)

[7.5.1 Les principaux dossiers du projet : 30](#_Toc87812589)

[CHAPITRE 4 34](#_Toc87812590)

[8 Phases de tests et validation : 35](#_Toc87812591)

[9 Les interfaces graphiques : 36](#_Toc87812592)

[9.1 Introduction : 36](#_Toc87812593)

[9.2 Interfaces graphiques : 36](#_Toc87812594)

[9.2.1 Page d’accueil 36](#_Toc87812595)

[9.2.2 Ecran de la page d’accueil d’admin: 39](#_Toc87812596)

[9.2.3 Partie administrateur 40](#_Toc87812597)

[9.2.4 Interface de la gestion d’un utilisateur. 41](#_Toc87812598)

[9.2.5 Partie Agence de location 48](#_Toc87812599)

[10 Conclusion générale : 52](#_Toc87812600)

**LISTE DES FIGURES**

|  |
| --- |
| **Figure 1:Cycle de vie de développement d’une application**  **Figure 2:Diagrammes de cas d’utilisations de la gestion locative** |
| **Figure3:Diagrammes de classe de la gestion locative**  **Figure 4:Le diagramme de séquence de la création d’un bien**  **Figure 5:Le diagramme de séquence d’ajouter une réservation**  **Figure 6:Le diagramme de séquence d’imprimer une facture**  **Figure 7: Diagramme d'état de la gestion de réservation**  **Figure 8:Diagramme d'état du compte de la gestion de bien**  **Figure9:Schéma de la notion MVC**  **Figure10:Ecran du dossier « Controller »**  **Figure11:Ecran des modèles utilisés**  **Figure 12:Ecran du dossier « Migrations »**  **Figure 13:Ecran du dossier « Views »**  **Figure 14:Ecran du dossier « Public »**  **Figure 15: Ecran de la page « Accueil »**  **Figure 16:Ecran de la page d’accueil (partie2)**  **Figure 17: Ecran de la page d’accueil (partie3)**  **Figure 18: Ecran de la page d’accueil (partie4)**  **Figure 19: Ecran de la page d’accueil (partie5)**  **Figure 20:Ecran de la page de se connecter**  **Figure 21:Ecran de la page «home» d’admin**  **Figure 22:Ecran de la page se connecter pour l’admin**  **Figure 23:Ecran de visualisation les comptes des utilisateurs**  **Figure 24:Ecran de l’interface employées.**  **Figure 25:Ecran d’ajouter les employées**  **Figure 26:Ecran de confirmation de la suppression.**  **Figure 27:Ecran de la page de modification d’un employé.**  **Figure 28:Ecran de gérer les offres**  **Figure 29:Ecran de la gestion de départements.**  **Figure 30:Ecran de gérer les régions.**  **Figure 31:Ecran de l’interface facture.**  **Figure 32:Ecran de facture.**  **Figure 33:Ecran de visualiser les statistiques**  **Figure 34:Ecran de l’interface de se connecter.**  **Figure 35:Ecran de l’interface agence.**  **Figure 36:Ecran de l'impression de facture.**  **Figure 37:Ecran de gérer les réservations.**  **Figure 38:Ecran de visualiser les statistiques.** |

# Introduction générale

*Dans le cadre de notre formation en Master, au sein du département génie informatique de l’école marocaine des sciences de l’ingénierie de Marrakech. Nous avons été amenés à réaliser un projet de fin d’études qui a pour thème « Développement d’une application web pour la gestion d’une villa» afin de concrétiser nos acquis durant notre formation d’une part. D’autre part, c’est afin de nous initier à la vie professionnelle et pratique qui diffère de la vie académique dont nous avons eu l’habitude.*

*Pour cela, nous avons décidé de développer une application de gestion locative permettant de générer des réservations qu'on pourra visualiser et gérer à partir du web.*

*L'application devra être utile au propriétaire des villas qui pourra ajouter, modifier et supprimer certaines informations. II devra pouvoir effectuer les contrats de location et les Imprimer.*

*L'application sera développée en langage PHP en utilisons le Framework laravel et sera reliée à une base de données MySQL.*

*Le présent rapport Le présent rapport est composé de quatre chapitres :*

* ***Le premier chapitre est consacré à la présentation du projet ainsi le planning prévisionnel.***
* ***Le second chapitre fera l’objet de l’analyse et l’étude conceptuelle de notre site web.***
* ***Le troisième chapitre se consacre à l’étude technique du projet***
* ***Le dernier chapitre est consacré à la réalisation de notre application.***

*Enfin, nous clôturons ce mémoire par une conclusion générale et perspective.*

**PRESENTATION DE L’ORGANISME D’ACCEUIL**

 **Présentation générale**

**GreenCom** est un centre de formation et de conseil spécialisé en développement informatique, il propose des formations de qualité destinées aux développeurs informatique d’un niveau débutant, intermédiaire ou professionnel.

**Fiche d’identité de la société**

* **Raison sociale : GreenCom**
* **Capital social de :** N/A
* **Fondée en :** 2011
* **Siège social :** Marrakech.
* **Adresse principale : Appartement 1 étage 1,Imm Machaallah,Abouab Marrakech، 40000**
* **Portables :** 0707712871
* **Registre de commerce :** 318517
* **Patente :** 406696719
* **Nombre de salariés :** 11 – 20 employés
* **Site web :** www. Greencom-agency.com

***Présentation du sujet***

*Ce chapitre présentera l’idée générale de notre projet sous-titre «* *Développement d’une application web pour la gestion locative : cas d’une villa».*

### Introduction :

L’admin cherche à améliorer la gestion et la communication avec les agences de location (les courtiers), en mettant en place un nouveau système d’information pour les agences de location (les courtiers). Ce dernier permettra de réserver les villas, modifier une réservation, supprimer une réservation ,la gestion des clients, imprimer des factures , visualiser les statistiques .

### Objectif :

Concevoir et réaliser une application web pour informatiser des tâches manuelles comme la gestion des clients, la gestion des biens , la gestion des réservations, consulter les statistiques.

### Problématique :

Actuellement la gestion des biens est gérée de manière manuelle le système actuel est ensemble de papiers et dossiers et fichiers Excel, ce qui engendre un certain nombre de problèmes tels que :

1. Volume important des informations traitées manuellement

2. Possibilité d’erreur dans le remplissage des différents documents et registres.

3. Perte de temps lors de recherche des informations dans les registres.

### Solution :

Informatisation du système d’information afin, d’assurer l’accès instantané aux données et une sécurisation de ces dernières, ce qui simplifie le travail administratif. Le site-web va permet aux admins et l’agence de location d’interagir avec la base de données en fonction de leurs besoins.

Travail demandé Créer une application web, fournissant 2 espaces principaux (acteurs):

* Administrateur

• la gestion des utilisateurs.

• la gestion des clients.

• la gestion des réservations.

• Imprimer les factures.

• Visualiser les statistiques.

* Agence de location (Courtiers)

• Un espace courtier qui a les fonctionnalités suivantes :

• Ajouter, Modifier ,Supprimer une réservation.

• Visualiser les statistiques.

• Imprimer les facteurs.

# CHAPITRE II

***Analyse et conception***

*Ce chapitre présentera le sujet, l’analyse et les conceptions détaillées avec les différents diagrammes en utilisant la modélisation UML. Il détaillera quelques user stories de l’application.*

## Introduction

Penser avant d’agir, faire des plans avant de construire, concevoir d’abord, développer ensuite c’est la démarche qui doit être suivre lors du développement d’une application et pour réussir n’importe quel projet. En effet, La conception d’un système informatique est une étape très importante qui va influencer la qualité et la fiabilité de toute application. D’abord, nous allons commencer ce chapitre par l’explication du modèle de cycle de vie de projet qu’on a choisi. Ensuite nous allons passer à la partie de conception détaillée où nous présentons l’architecture globale de l’application. Enfin nous citons les différents diagrammes de cas d’utilisation, de séquences et le diagramme de classes.

## Cycle de vie de développement de projet

Le cycle de vie d’un logiciel (en anglais software life cycle), désigne toutes les étapes du développement d’un logiciel, de sa création à sa disparition. L’objectif d’un tel découpage est de permettre de définir des bornes intermédiaires permettant la confirmation du développement logiciel, c’est-à-dire la coïncidence du logiciel avec les besoins exposés, et la vérification du processus de développement, c’est-à-dire la correspondance des méthodes mises en œuvre. Pour fournir une meilleure réalisation, nous avons présenté le modèle en cascade (a été mis au point dès 1996). Il est décrit par la figure suivant :

**Figure 1:Cycle de vie de développement d’une application**

* Dans ce modèle le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s’ils sont jugés satisfaisants.

## Présentation du langage UML :

UML (en anglais Unified Modeling Language, « langage de modélisation unifié ») est un langage graphique de modélisation des données et des traitements. C’est une formalisation non- propriétaire de la modélisation objet utilisée en génie logiciel. UML spécifie plusieurs objectifs qui font un outil exact de communication :

* ***Comprendre et décrire les besoins.***
* ***Spécifier un système.***
* ***Établir l’architecture logicielle.***

## Intérêt de la modélisation :

L’utilisation de la modélisation conceptuelle dans le développement des systèmes d’information permet de prendre en compte les besoins des applications d’une façon plus adéquats et de présenter d’une manière abstraite certains aspects des systèmes physiques et humains.

## Les avantages d’UML :

* UML est un langage formel et standardisé.
* Gain de précision.
* Motivation à l’utilisation d’outils.
* Gagne de stabilité et de fixité.
* UML est un support de communication adéquat et compétitif.
* Il éclaire et facilite la compréhension de représentation abstraite complexe.
* Son caractère plurivalent et sa souplesse en font un langage universel.
* UML a pour objectif de spécifier, édifier et documenter les systèmes à base de logiciel.
* UML n’est pas une méthode mais une notation qui laisse la liberté de la conception.
* UML est un langage qui permet de modéliser tous les types de systèmes informatiques mais, qui nécessite toutefois une méthodologie de conception.

* UML normalises les concepts objet, sa notion graphique permet d’exprimer une solution objet, ce qui simplifie la comparaison et l’appréciation des solutions.
* UML cadre l’analyse objet, il permet non seulement de représenter les concepts objets, mais il sous-entend une démarche d’analyse qui permet de reproduire une solution objet de manière itérative, grâce aux diagrammes, qui supportent l’abstraction.
* Un diagramme UML est une représentation graphique, et à chaque vue correspondent des diagrammes qui sont répartis selon leurs aspects statiques ou dynamiques :

1. Statique :

* Cas d’utilisation
* Classes
* Composants
* Objets
* Déploiement

1. Dynamique (comportementaux) :

* Séquences
* Activité
* État-transition
* Collaboration

1. Fonctionnel :

* Cas d’utilisation
* Collaboration
* Ces diagrammes ne sont pas nécessairement tous produits à l’occasion d’une modélisation. Les plus utilisables sont les diagrammes d’activités, de classes, de cas d’utilisation, d’objets, d’états transitions et de séquence. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d’étude à qu’ils permettent de formaliser les contraintes de la réalisation et les solutions.

Voici une présentation rapide des différents diagrammes UML qui vont être utilisés tout au long du projet

* Le diagramme des cas d’utilisation : Représente la structure des fonctionnalités nécessaires aux utilisateurs des systèmes.
* Le diagramme de classe : L’un des diagrammes les plus importants dans un développement orienté objet. Sur la branche fonctionnelle, ce diagramme est prévu pour développer la structure des entités manipulées par les utilisateurs.

En conception, le diagramme de classes représenté la structure d’un code orienté objet.

* Le diagramme de séquence : Représente les échanges de messages entre objets, dans le cadre d’un fonctionnement particulier du système.
* Le diagramme d’état : Les diagrammes d’états montrent les différents états d’un objet ainsi que les transitions entre ces états. Ils peuvent également représenter un protocole

# Analyse Fonctionnelle :

## Définition :

**L'analyse fonctionnelle** permet de traduire un besoin client de façon très détaillée et très structurée sans décrire un moyen mais en restant axée sur les caractéristiques du résultat à obtenir.

## Règle de gestion

La gestion d’une villa offrira les services suivants :

### Mode Réservations :

* L’admin peut créer et modifier une réservation.
* Un courtier peut consulter et modifier une réservation.
* L’admin peut supprimer une réservation.

### Mode Clients :

* L’admin peut ajouter les clients.
* L’admin peut modifier les clients.
* L’admin peut supprimer les clients.

### Mode Agence de location:

* L’admin peut ajouter les agences de location.
* L’admin peut modifier les agences de location.
* L’admin peut supprimer les agences de location.

### Mode Facture :

* L’admin peut imprimer des factures.
* Les courtiers peuvent imprimer des factures.

### Mode Statistiques :

* L’admin peut visualiser les statistiques.
* Les courtiers peuvent visualiser les statistiques.

### Mode Villa :

* L’admin peut ajouter des villas.
* L’admin peut modifier des villas.
* L’admin peut supprimer des villas.

## Les acteurs et leurs rôles :

### Admin ( propriétaire du bien ):

* Créer les comptes des courtiers.
* Ajouter les clients.
* Modifier les clients.
* Supprimer les clients.
* Ajouter les réservations.
* Modifier et supprimer les réservations.
* Visualiser les statistiques.
* Imprimer les factures.
* Ajouter, Modifier, Supprimer les villas.

### Agence de location (Courtiers ):

* Ajouter les réservations.
* Modifier et supprimer les réservations.
* Visualiser les statistiques.
* Imprimer les factures.

.

## Les Scénarios :

### Définition :

Les scénarios mettent en évidence les interactions entre les actions de l’utilisateur et le système (le logiciel). Cela peut se faire par une liste numérotée d’actions ou sous forme de tableau qui démontre clairement ce qui est réalisé par l’utilisateur et ce qui est du ressort du système.

### Acteur : Admin

#### Ajouter les clients :

1. Choisir l’option ‘Clients’
2. Cliquer sur « Ajouter »
3. Remplir le formulaire
4. Cliquer sur « Enregistrer »

#### Modifier les clients :

Choisir l’option ‘Clients’

Cliquer sur « Edit »

Modifier le formulaire

Cliquer sur « Enregistrer »

#### Supprimer les Clients :

Choisir l’option ‘Clients’

Cliquer sur « Supprimer »

Confirmer la suppression en cliquant sur « Yes,delete it »

#### Modifier les users :

Choisir l’option ‘users’

Cliquer sur « Edit »

Modifier le formulaire

Cliquer sur « Enregistrer »

#### Imprimer les factures :

1. Choisir l’option ’Imprimer’ pour imprimer la facture
2. Cliquer sur « Imprimer ».
3. Confirmer l’impression .

#### Ajouter les users :

Choisir l’option « users »

Cliquer sur « ajouter»

Remplir le formulaire.

Cliquer sur « Enregistrer »

#### Supprimer les users :

Sélectionner l’option ‘users’

Cliquer sur « Supprimer »

Confirmer la suppression en cliquant sur « Yes,delete it »

#### Visualiser les statiques :

1. Sélectionner l’option ‘statics’
2. Choisir le diagramme.

#### Ajouter une réservation :

1. Choisir l’option ‘Réservations’
2. Cliquer sur « Ajouter une réservations»
3. Remplir le formulaire
4. Cliquer sur « Enregistrer »

#### Modifier une réservation :

Choisir l’option ‘Réservations’

Cliquer sur « Edit »

Modifier le formulaire

Cliquer sur « Enregistrer »

#### Supprimer une réservation :

Choisir l’option ‘Réservations’

Cliquer sur « Supprimer »

Confirmer la suppression en cliquant sur « Yes,delete it »

#### Ajouter les bien :

1. Choisir l’option ‘Biens’
2. Cliquer sur « Ajouter »
3. Remplir le formulaire
4. Cliquer sur « Enregistrer »

#### Modifier les bien :

Choisir l’option ‘Biens’

Cliquer sur « Edit »

Modifier le formulaire

Cliquer sur « Enregistrer »

#### Supprimer les bien :

Choisir l’option ‘biens’

Cliquer sur « Supprimer »

Confirmer la suppression en cliquant sur « Yes,delete it »

### Acteur : Agence de location (Courtier):

#### Imprimer les factures :

1. Choisir l’option ’Imprimer’ pour imprimer la facture
2. Cliquer sur « Imprimer »
3. Confirmer l’impression

#### Visualiser les statiques :

1. Sélectionner l’option ‘statics’
2. Choisir le diagramme.

#### Ajouter une réservation :

1. Choisir l’option ‘Réservations’
2. Cliquer sur « Ajouter une réservations»
3. Remplir le formulaire
4. Cliquer sur « Enregistrer »

#### Modifier une réservation :

1. Choisir l’option ‘Réservations’
2. Cliquer sur « Edit »
3. Modifier le formulaire
4. Cliquer sur « Enregistrer »

#### Supprimer une réservation :

1. Choisir l’option ‘Réservations’
2. Cliquer sur « Supprimer »
3. Confirmer la suppression en cliquant sur « Yes,delete it »

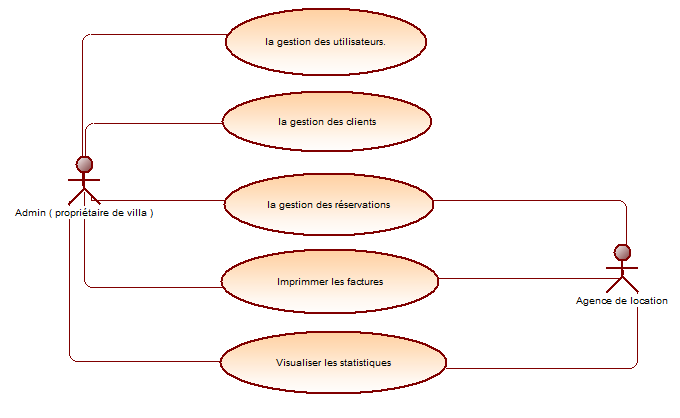
# Les Diagrammes :

## Diagramme de cas d’utilisation :

### Définition :

Le diagramme de cas d’utilisation permet de déterminer les possibilités d'interférence entre le système et les acteurs, c'est-à-dire déterminer toutes les fonctionnalités que doit fournir le système. Il permet aussi de délimiter ce dernier.

* Chaque usage effectué par les acteurs est représenté par un cas d'utilisation.
* Chaque cas d'utilisation symbolise une fonctionnalité qui leur est offerte afin d’engendrer le résultat attendu.
* Le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur.

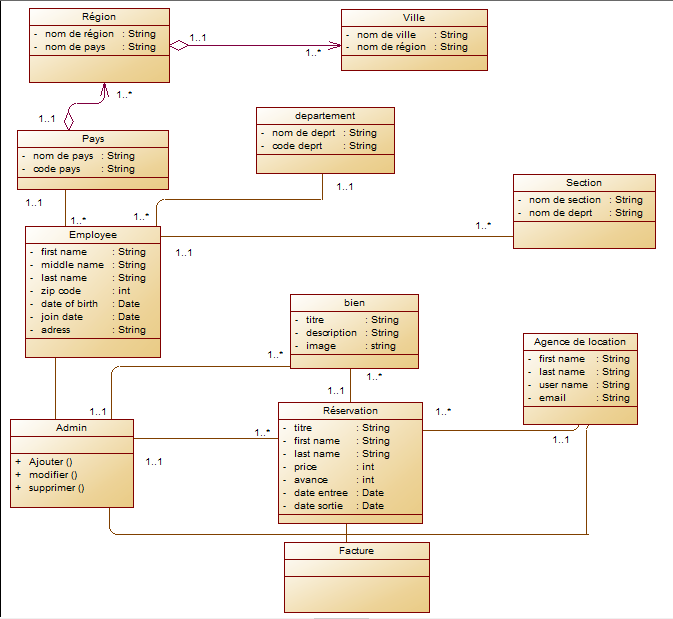


|  |
| --- |
| **Figure 2:Diagrammes de cas d’utilisations de la gestion locative** |

## Diagramme de classe :

### Définition :

Diagramme de classe : Un diagramme de classe est un diagramme UML qui contient des classes, des interfaces, des packages et leurs relations, et qui fournit une vue logique de tout ou partie d’un système informatique. On construit un diagramme de classes pour simplifier l’interaction des objets d’un système qu’on est en train de modéliser. Ces diagrammes expriment la structure statique d’un système en termes de classes et de relations entre eux. Une classe décrit un ensemble d’objets et une association décrit un ensemble de liens. Un diagramme de classe n’exprime rien de spécifique concernant les liens d’un objet particulier, mais il décrit, le lien potentiel entre un objet et d’autres objets



**Figure3:**Diagrammes de classe de la gestion locative

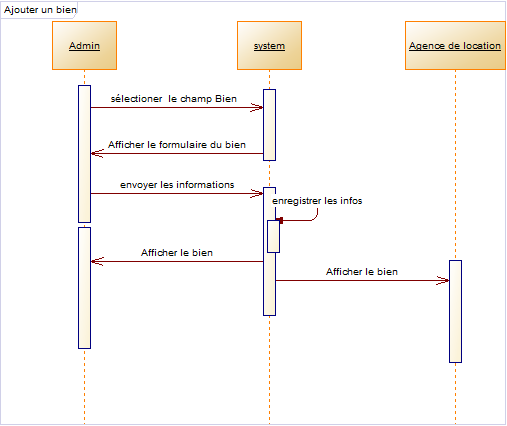
## Les diagrammes de séquences :

### Définition :

**Le diagramme de séquence**, décrit les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. Un diagramme de séquence montre une interférence présentée en séquence dans le temps. En particulier, il montre aussi les objets qui participent à l'interaction par leur "ligne de vie" et les messages qu'ils échangent présentés en séquence dans le temps. Voici quelques notions de base du diagramme :

* Scénario : une liste d'actions qui décrivent une interaction entre un acteur et le système.
* Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.
* Message : Un message définit une communication particulière entre des lignes de vie (objets ou acteurs)

### Création d’un bien :



**Figure 4:**Le diagramme de séquence de la création d’un bien

### Ajouter Réservation:

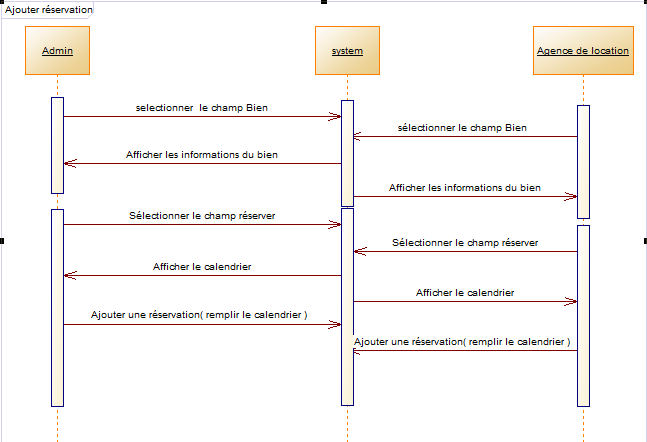
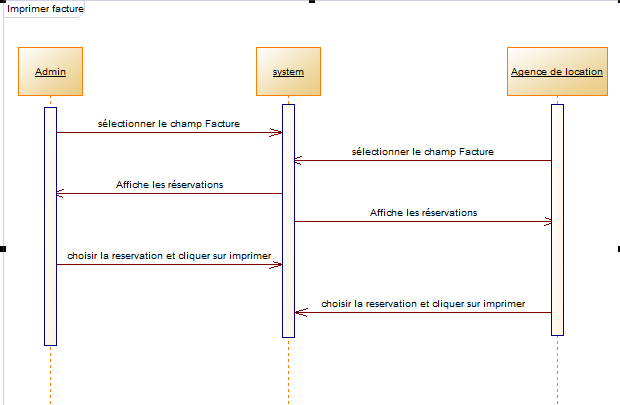


Figure 5:Le diagramme de séquence d’ajouter une réservation

### Imprimer facture:



**Figure 6:**Le diagramme de séquence d’imprimer une facture

## Diagrammes d’états :

### Diagramme d’état de la gestion de réservation

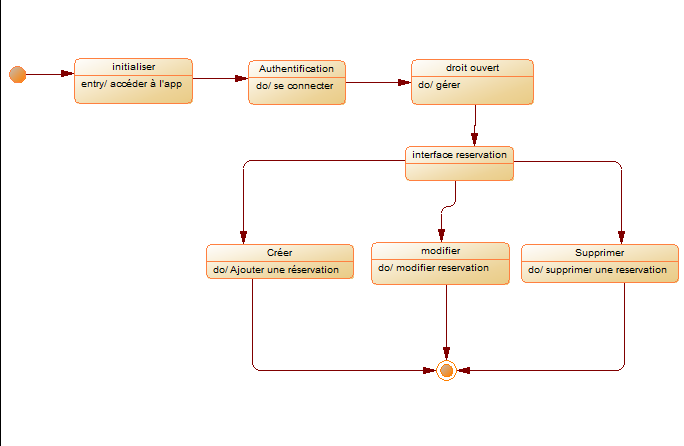
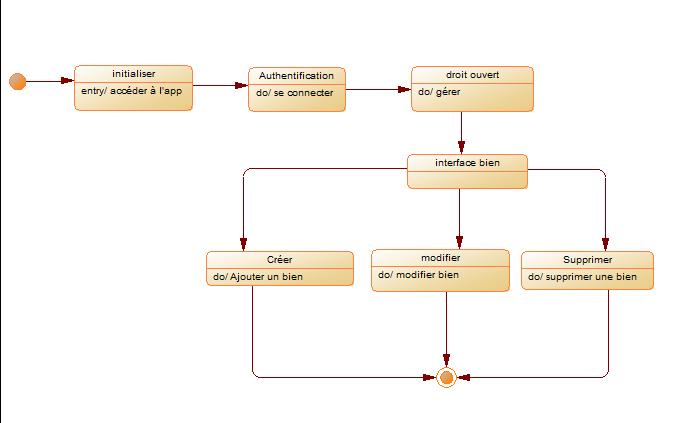


Figure 7: Diagramme d'état de la gestion de réservation

### Diagramme d’état de la gestion de bien



**Figure 8:Diagramme d'état du compte de la gestion de bien**

## Conclusion

Ce chapitre a été consacré à la modélisation UML du système.

Nous avons présenté différents diagrammes : les diagrammes de cas d’utilisation, de classe, et de séquences afin de spécifier de façon détaillée les aspects fonctionnels, dynamiques et statiques du système. Dans le chapitre suivant, nous présenterons la réalisation et la mise à l’essai de ce système.

***Choix technologique, Architecture logicielle***

*Ce chapitre détaillera les listes des modules de test, l’architecture technique et applicative de notre application.*

# Choix technologique :

## Réalisation et mise en œuvre :

Dans cette partie on présentera les différents outils utilisés du langages, Framework, logiciel et la plateforme de développement web qui permettront d’atteindre les objectifs et donc réaliser ce projet et d’assurer le bon développement de l’application Web

Langages utilisés :

### img HTML5

HTML5 est la dernière évolution des standards qui définissent HTML. Il s'agit de la nouvelle version du langage HTML, avec de nouveaux éléments, attributs et comportements, mais aussi un ensemble plus large de technologies qui permettent des sites web plus variés et puissants, et des applications web.

Conçu pour être utilisable par tous les développeurs de l'Open Web, cette page de référence fait des liens vers de nombreuses ressources sur les technologies HTML5, classés dans différents groupes d'après leur fonction.

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.png CSS3 :

CSS (feuilles de styles) est un langage qui permet de gérer la présentation d’une page Web. Les styles permettent de définir des règles appliquées à un ou plusieurs documents HTML. Ces règles portent sur le positionnement des éléments, l’alignement, les polices de caractères, les couleurs, les marges et espacements, les bordures, les images de fonds, etc.

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.png PHP

PHP est un langage de programmation libre5, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP4, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.

PHP a permis de créer un grand nombre de sites web célèbres. Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

### **C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.png** JavaScript

JavaScript est un langage de scripts qui incorporé aux balises Html, permet d'améliorer la présentation et l'interactivité des pages Web.

JavaScript est donc une extension du code Html des pages Web. Les scripts, qui s'ajoutent ici aux balises Html, peuvent en quelque sorte être comparés aux macros d'un traitement de texte.

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.pngAjax

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 combine JavaScript, les CSS, JSON, XML, le DOM et le XML, http, Request afin d'améliorer maniabilité et confort d'utilisation des applications internet riches

## Logiciel utilisés :

### downloadPowerAMC :

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées, de réaliser tous les types de modèles informatiques et d’améliorer la modélisation, les processus, le coût et la production d’application.

Plate-forme de développement web utilisés :

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.jpglarogon :

**Laragon** est un environnement de développement web rapide, flexible, intuitif, productif et puissant qui d'adresse à tous. **Laragon** vous offre tout ce dont vous avez besoin pour créer des applications web modernes.

**Laragon** est très léger et restera aussi maigre que possible. Le binaire de base lui-même est inférieur à 2 Mo et utilise moins de 4 Mo de RAM lors de l'exécution.

**Laragon** est très facile à utiliser:  
pendant presque tout le temps, il fonctionnera immédiatement sans que vous n'ayez besoin de toucher à des fichiers de configuration.

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.pngHeidi SQL :

**Heidisql**: est un outil d’administration de base de données possédant un éditeur SQL et un constructeur de requête. Il a été développé et optimisé pour être utilisé avec le **SGBD** relationnel **MYSQL.** En effet, grâce à son interface simple d’utilisation de type Windows l’application vous permet d’administration et de naviguer au sein de vos bases de données, mais également de générer des exports SQL, de synchroniser les éléments entre 2 bases, d’importer des fichier Texte, etc.

### C:\Users\ETTAGDOUINI\Pictures\download.png visual studio code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows,

Linux et macOS2.

Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires.

Le code source de Visual Studio Code provient du projet logiciel libre et open source VSCode de

Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation.

Dans le Stack Overflow 2019 Developer Survey, Visual Studio Code a été classé comme l'outil d'environnement de développement le plus populaire, avec 50,7% des 87317 répondants déclarant l'utiliser.

## Le patron de la conception :

« MVC » c’est un design pattern (patron de la conception), c’est-à-dire un concept d’architecture logicielle pour son application, il permet d’avoir un code plus structuré, plus évolutif, plus maintenable, permettant de profiter de plusieurs mécanismes, d’avoir la persistance des données.

Le ‘’Modèle ‘’ est la représentation interne des données, il permet comme son nom l’indique de modéliser les données que l’on va manipuler dans l’application.

Le modèle représente les véritables données avec toutes les informations qu’elles véhiculent.

La ‘’ Vue ‘’ quant à elle est la représentation visuelle de ces données à l’écran.

Le ‘’Contrôleur ‘’enfin sert à faire l’interface entre le modèle et la vue.

En effet, puisque le modèle et la vue sont sensés être au maximum indépendants, le contrôleur sert à faire le lien pour faire pour faire communiquer l'un (M) avec l’autre (V).

Ci-dessous **le mode de fonctionnement du MVC** :

Mysql

View

Module

Controller

Dispatcher

Browser

Web Server

Figure9:Schéma de la notion MVC

## Framework utilisé :

### C:\Users\ETTAGDOUINI\AppData\Local\Microsoft\Windows\INetCache\Content.Word\download.jpg Laravel7

**Laravel** est un Framework de développement de type MVC et un web open source écrit en PHP2 respectant le principe modèle-vue-contrôleur et entièrement développé en programmation orientée objet. Laravel est distribué sous licence MIT, avec ses sources hébergées sur Git Hub.

Laravel fournit des fonctionnalités en termes de routage de requête, de mapping objet-relationnel (un système baptisé Eloquent implémentant Active Record), d'authentification, de vue (avec Blade), de migration de base de données, de gestion des exceptions et de test unitaire2.

## L’architecture des dossiers du projet

### Les principaux dossiers du projet :

* Le dossier « app » contient le code source PHP (fonctions et classes).
* Le dossier « resources/views » contient les fichiers HTML.
* Le dossier « public » contiendra les images, CSS et les scripts JavaScript.
* Le fichier « routes/web.php » définira quelle action est associée à quelle URL (« /contact » vers cette fonction PHP par exemple).
* Le dossier config et le fichier « .env » qui contiendront vos configurations (mots de passe, nom de domaine, type de base de données…).
* Le fichier artisan que l’on a déjà exécuté avec ‘’php artisan serve’’ nous permettra de faire un bon nombre de choses en ligne de commande.

#### Les contrôleurs :

Une image contenant texte, capture d’écran, écran

Description générée automatiquement

Figure10:Ecran du dossier « Controller »

#### Les modèles :

Une image contenant texte

Description générée automatiquement

Figure11:Ecran des modèles utilisés

#### Les migrations :

Une image contenant texte

Description générée automatiquement

Figure 12:Ecran du dossier « Migrations »

#### Les vues :

Une image contenant texte

Description générée automatiquement

Figure 13:Ecran du dossier « Views »

#### Public :

Une image contenant texte

Description générée automatiquement

Figure 14:Ecran du dossier « Public »

# CHAPITRE 4

***Réalisation du projet***

*Ce chapitre présentera les interfaces graphiques de notre application qui fait la gestion d’un bien exemple d’une villa*

# Phases de tests et validation :

En informatique un test désigne une procédure de vérification partielle d’un système informatique. Le but est de trouver un nombre maximum de comportements problématiques de l’application, car il est impossible de prouver qu’une application fonctionne bien dans tous les cas. Plus on trouve d’erreurs plus il y’a de chances d’avantage d’erreurs dans l’application visé. Les tests de vérification ou de validation visent à s’assurer que ce système réagit de la façon prévue par ses concepteurs (spécifications) ou est conforme aux attentes du client (Besoins). Dans cette partie nous allons présenter une liste des modules de test, scénario de test et le résultat obtenu à partir du tableau suivant :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tableau II: les résultats des modules de test   |  |  |  | | --- | --- | --- | | Module | Scénarios de test | Résultat obtenue | | **Connection à l'application** | **Entrer un login et un mot de passe correct** | **Exécution correcte** | | **Entrer un login et un mot de passe incorrect** | **Exécution correcte** | | **Connection à la base des données** | **Sélectionner des données de la base des données** | **Exécution correcte** | | **Gestion des données** | **Insertion des données** | **Exécution correcte** | | **Modification des données** | **Exécution correcte** | | **Suppression des données** | **Exécution correcte** | | **Consultation des données** | **Exécution correcte** | | **Gestion des interfaces** | **Bon fonctionnement(navigation, formulaire etc….)** | **Exécution correcte** | | **Gestion des exceptions** | **Insertion des données existantes** | **Exécution correcte** | | **Transfert correcte des informations entre les interfaces** | **Exécution correcte** | | **Mise à jour de BD à chaque modification** | **Exécution correcte** | | **Insertion au BD** | **Exécution correcte** | |

# Les interfaces graphiques :

## Introduction :

L’interface graphique est une partie très importante pour la réalisation d’une application web convenable et conviviale car elle permet de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application offrant un certain plaisir à l’utilisateur lors de sa navigation. Ainsi ce critère peut faire la différence entre une application et une autre bien qu’elles aient les mêmes fonctionnalités. Dans la conception des interfaces de notre application, nous avons respecté un ensemble des choix ergonomiques comme la lisibilité, la compréhensibilité, etc.

## Interfaces graphiques :

### Page d’accueil

* 1er partie

Cette partie de l'interface accueil permet aux utilisateurs de se connecter

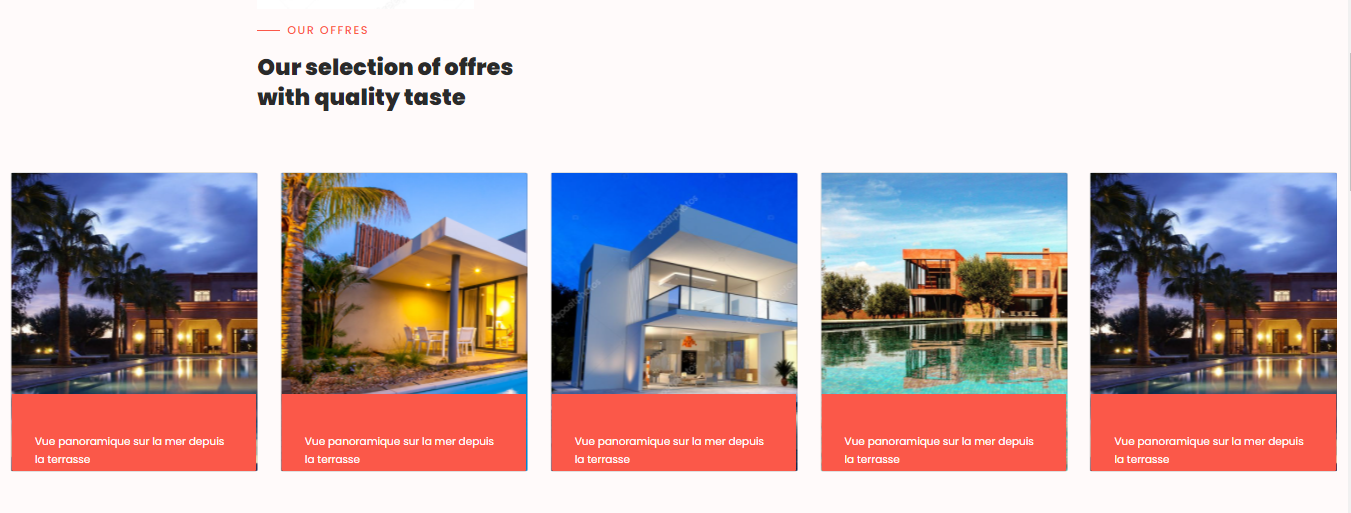
**Figure 15: Ecran de la page « Accueil »**

Une image contenant texte, extérieur, signe, capture d’écran

Description générée automatiquement

* 2ème partie :

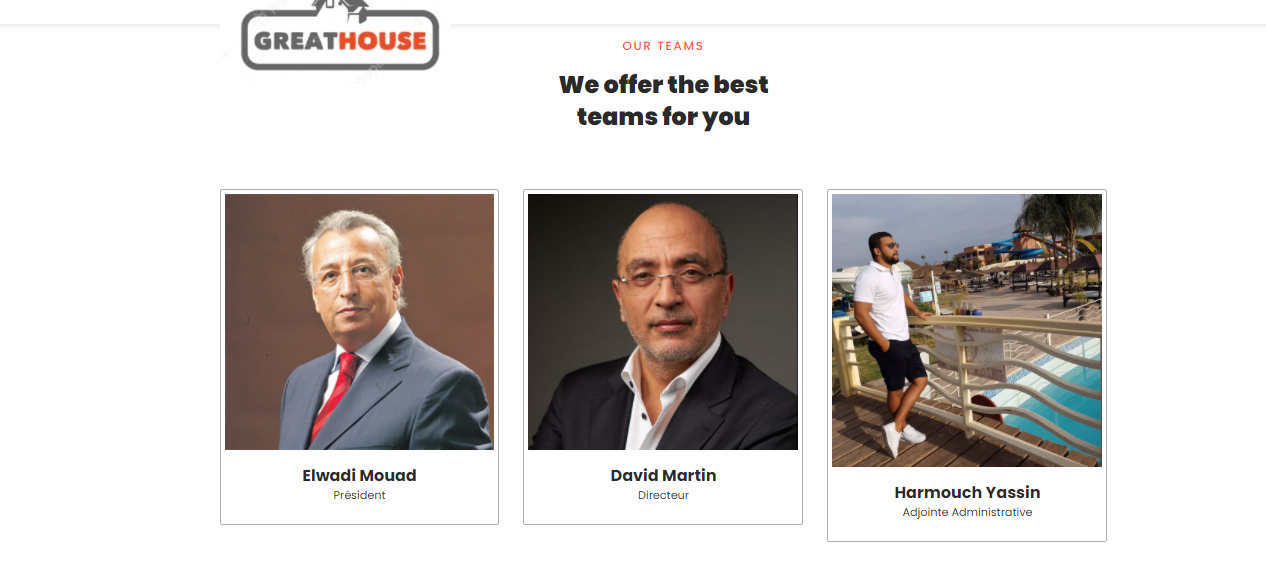
Cette partie de l'interface accueil contient les offres



**Figure 16:Ecran de la page d’accueil (partie2)**

* 3ème partie :

Cette partie de l'interface accueil contient l’équipe.

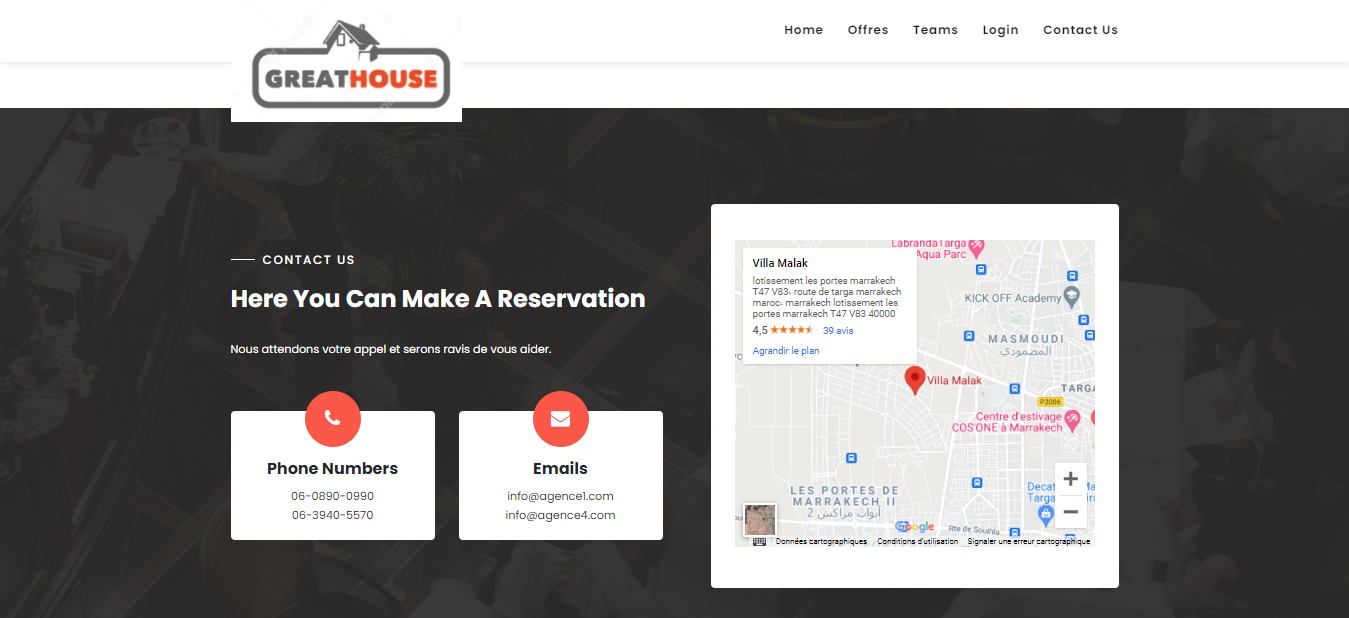


**Figure 17: Ecran de la page d’accueil (partie3)**

* 4ème partie :

Cette partie contient les contacts administratifs et l'emplacement du bien.

**Figure 18: Ecran de la page d’accueil (partie4)**



* 5eme partie :

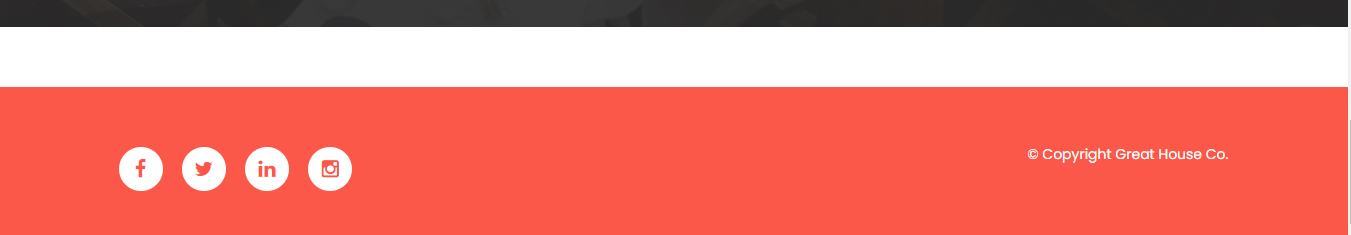
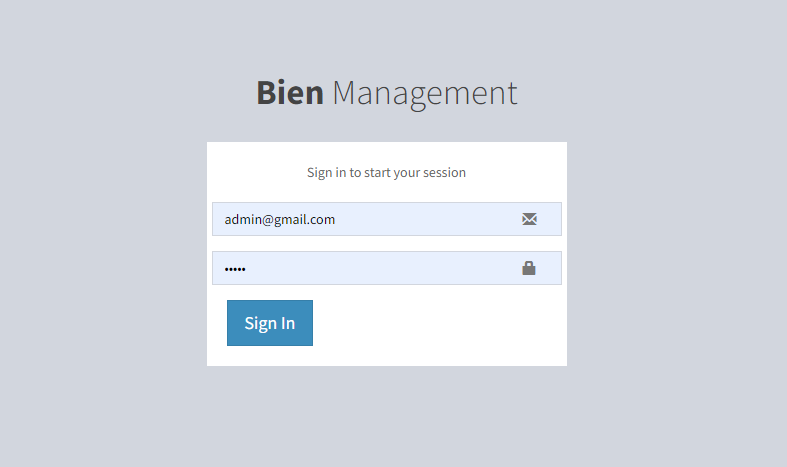


Figure 19: **Ecran de la page d’accueil (partie5)**

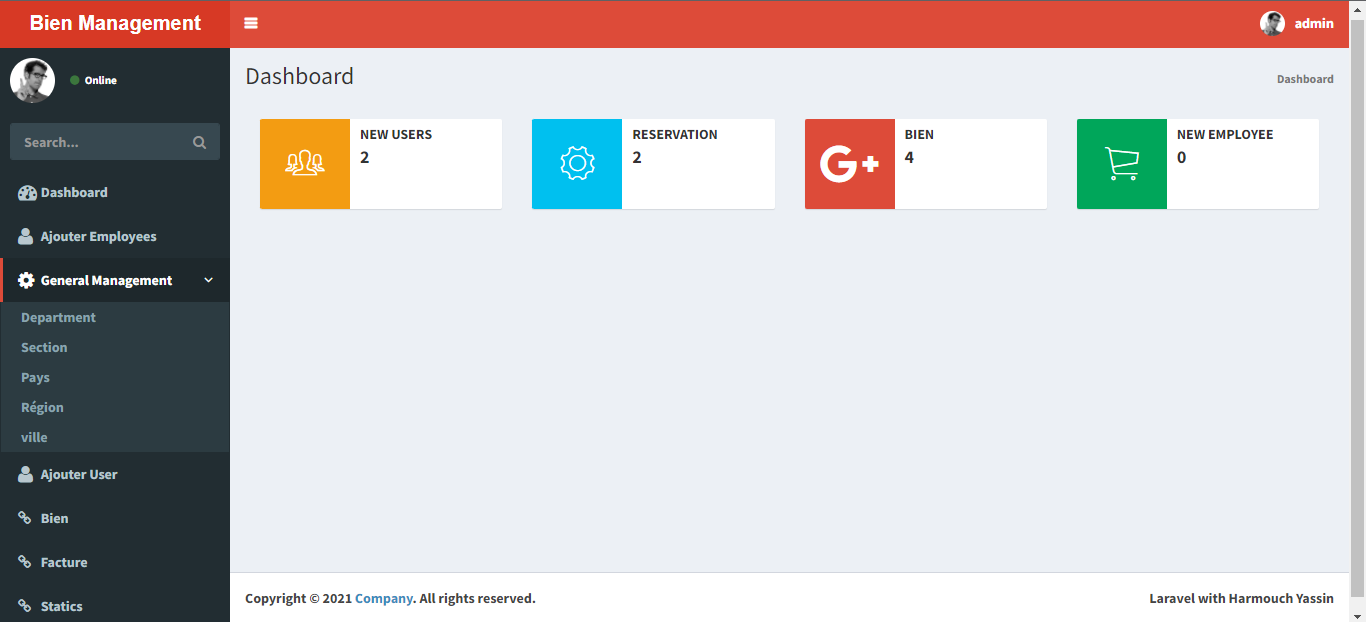
#### Espace de se connecter



**Figure 20:Ecran de la page de se connecter**

### Ecran de la page d’accueil d’admin:

Après le remplissage du formulaire va diriger vers la page home d’admin



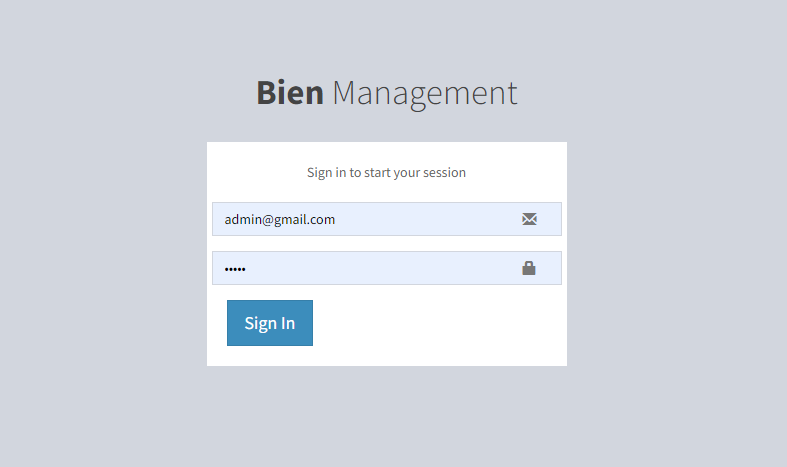
**Figure 21:Ecran de la page «home» d’admin**

### Partie administrateur

La partie administrateur permet à celui-ci :

* Gérer les comptes des utilisateurs.
* Gérer les biens
* Gérer les employées.
* Imprimer les factures.
* Gérer les réservations

#### Interfaces se connecter



**Figure 22:Ecran de la page se connecter pour l’admin**

### Interface de la gestion d’un utilisateur.

#### Visualiser les comptes des utilisateurs

Premièrement vous cliquer sur « Ajouter users»

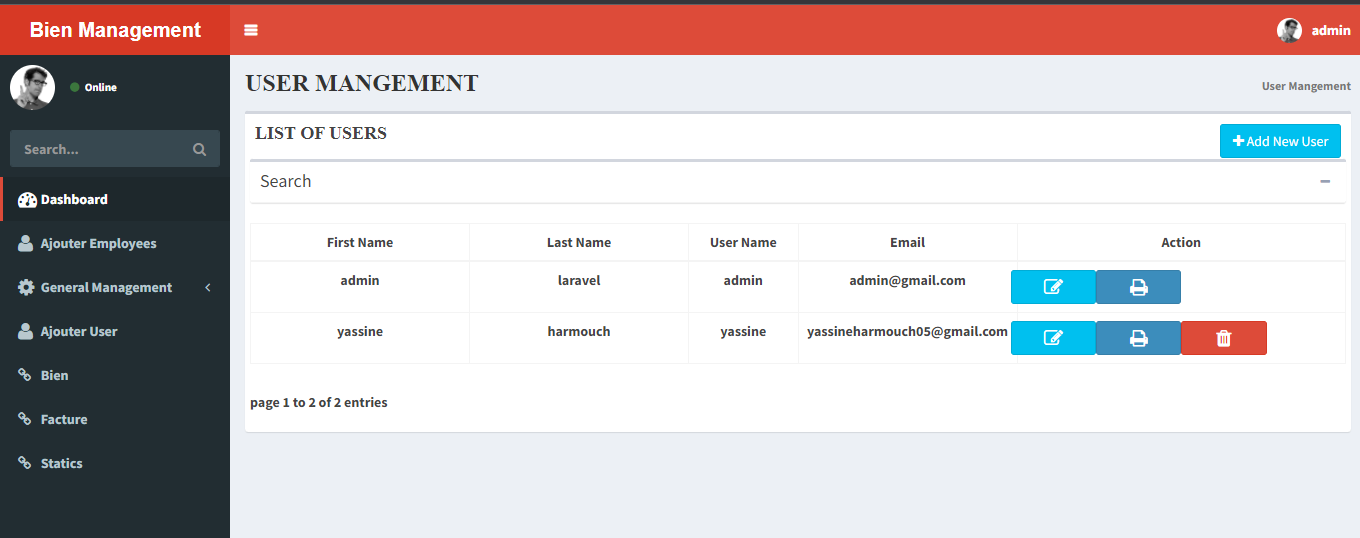


Figure 23:Ecran de visualisation les comptes des utilisateurs

#### Interface employée

Cette interface permet à l’admin de gérer ses employées.

Une image contenant texte

Description générée automatiquement

Figure 24:Ecran de l’interface employées.

#### Ajouter les employées.

Cette interface permet à l’admin d’ajouter les employées.

**Une image contenant texte

Description générée automatiquement**

Figure 25:Ecran d’ajouter les employées

#### Supprimer les employées.

Cette interface permet à l’admin de supprimer les employées

Une image contenant texte

Description générée automatiquement

Figure 26:Ecran de confirmation de la suppression.

#### Interface modifier employées.

Cette interface permet à l’admin de modifier les employées.

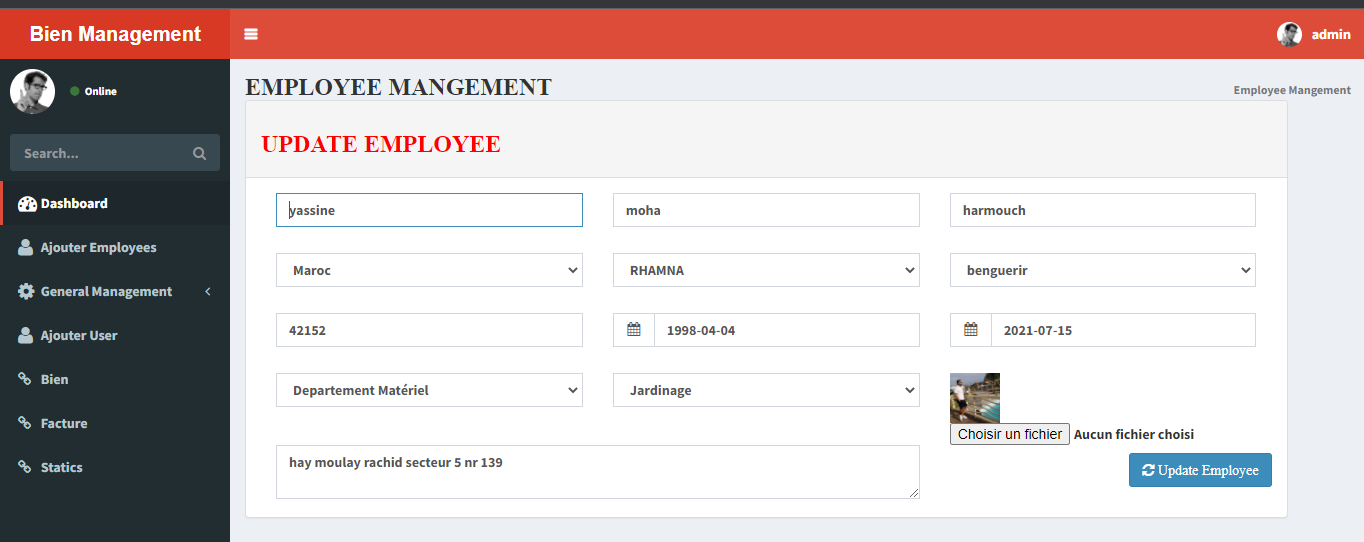
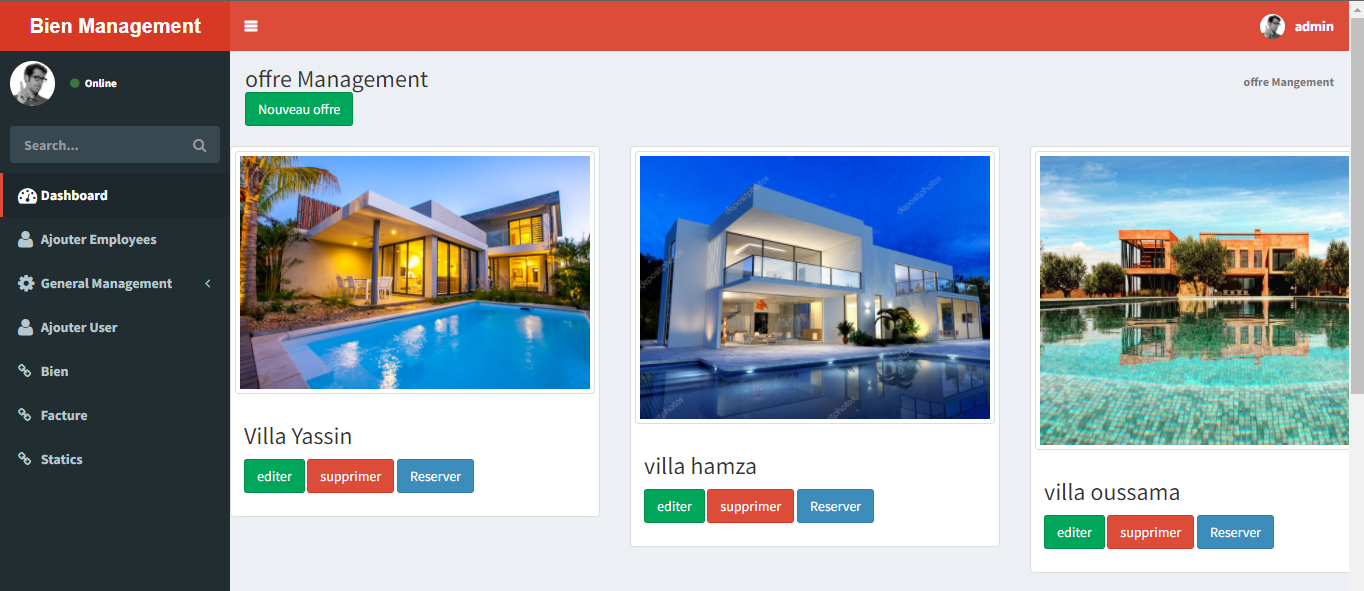


Figure 27:Ecran de la page de modification d’un employé.

#### Interface d’offre.

Cette interface permet à l’admin de gérer les offres.

Figure 28:Ecran de gérer les offres



#### Interface de gérer les départements.

Cette interface permet à l’admin de gérer les départements.

Une image contenant texte

Description générée automatiquement

Figure 29:Ecran de la gestion de départements.

#### Interface de gérer les pays.

cette interface permet à l’admin de gérer les pays.

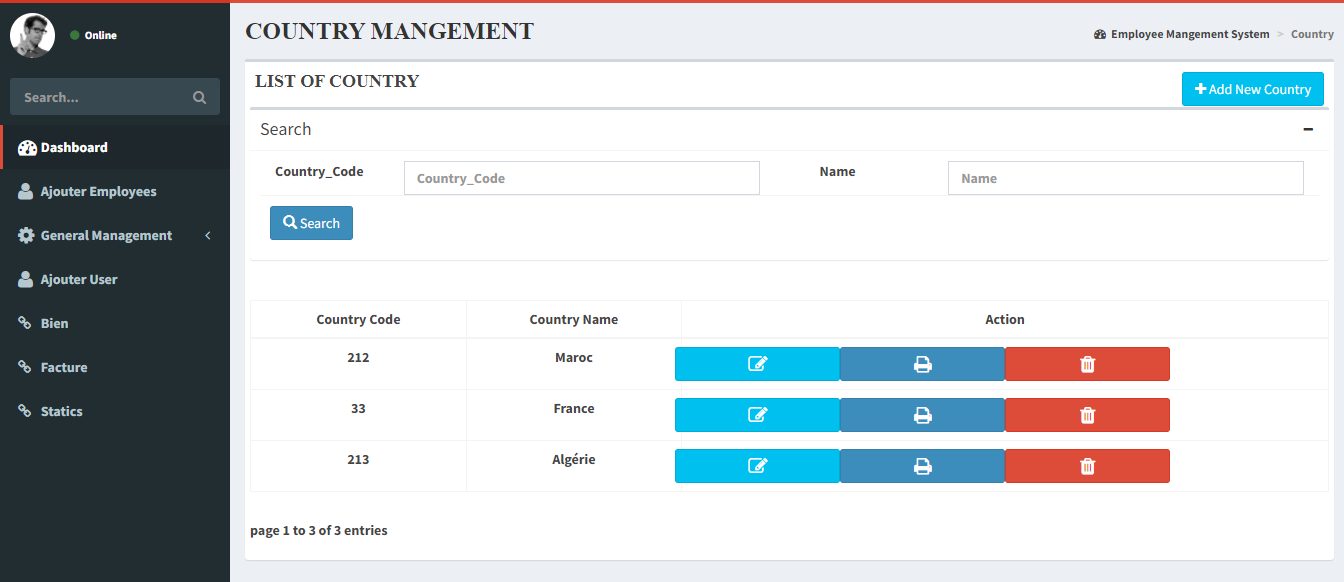
.

Figure 30:Ecran de gérer les pays

#### Interface de gérer les régions.

Cette interface permet a l’admin de gérer les régions.

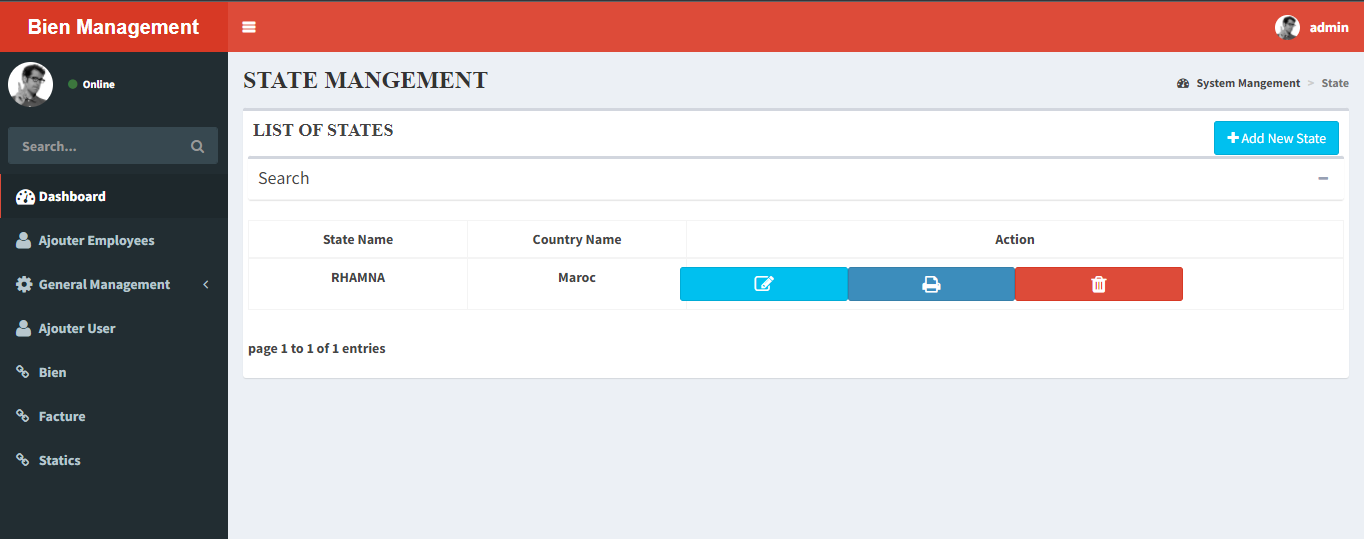
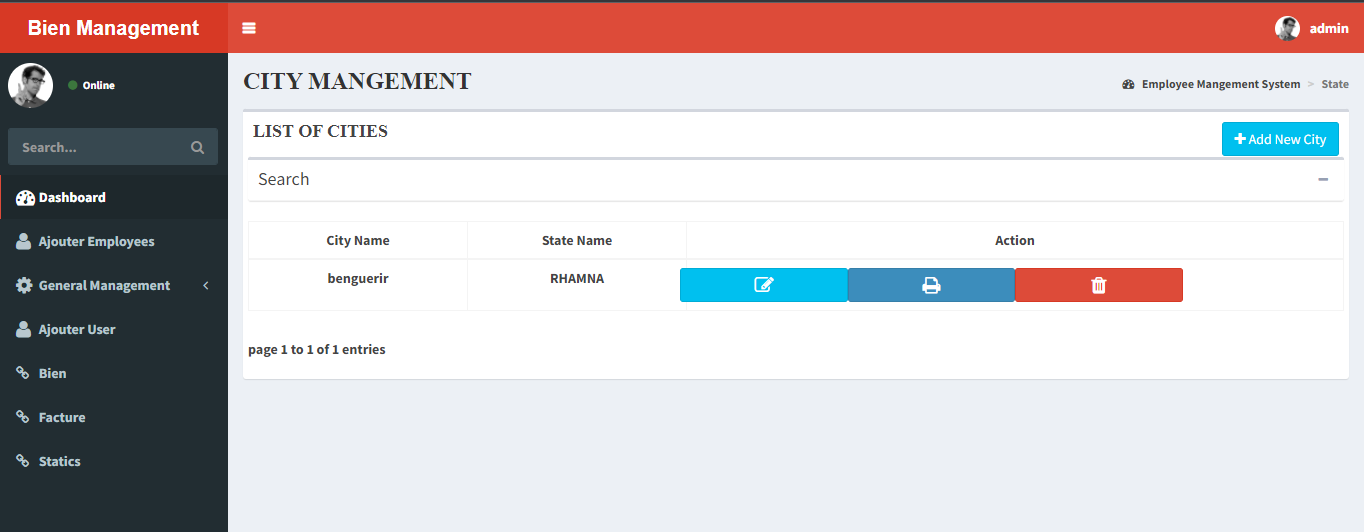


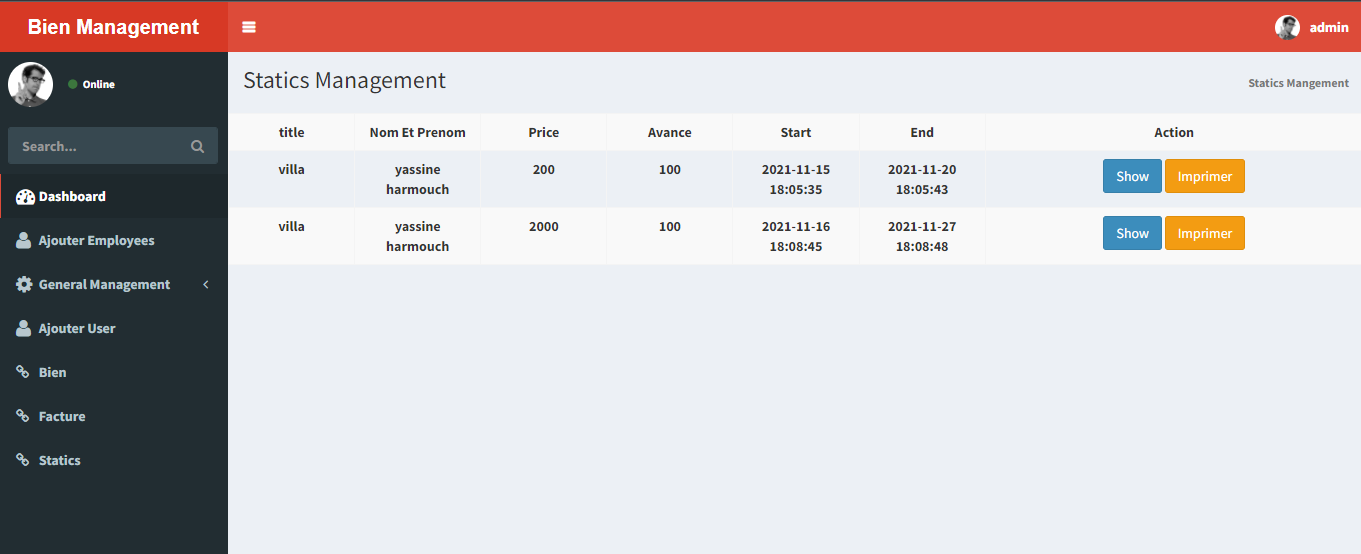
Figure 31:Ecran de gérer les régions.

#### Interface de gérer les villes



***Figure 32:Ecran de la gestion des villes.***

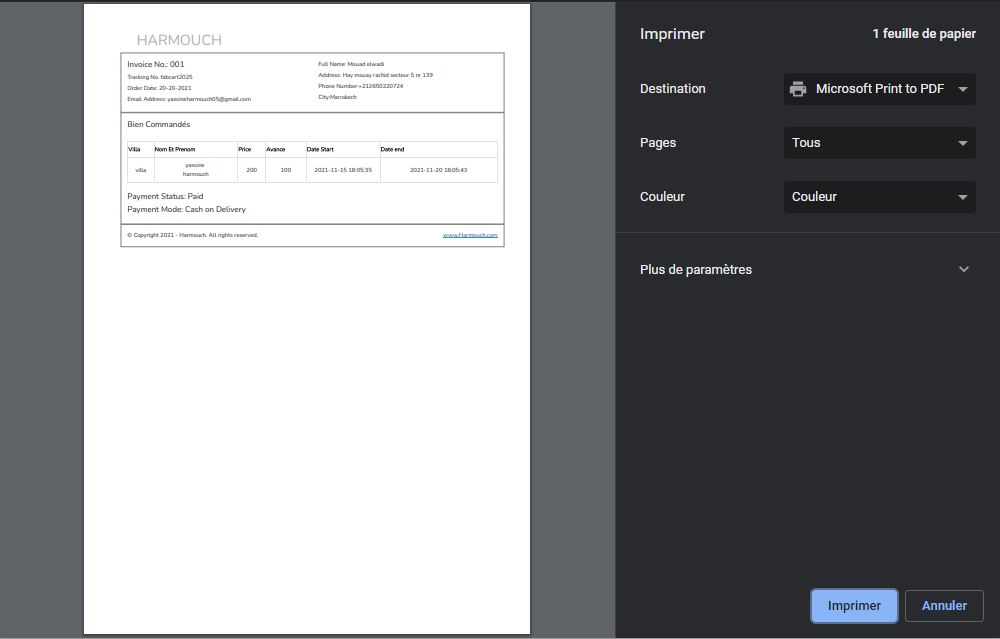
#### Interface facture.



**Figure 33:Ecran de l’interface facture.**

#### Interface pour imprimer facture

Cette interface permet à l’admin d’imprimer facture.

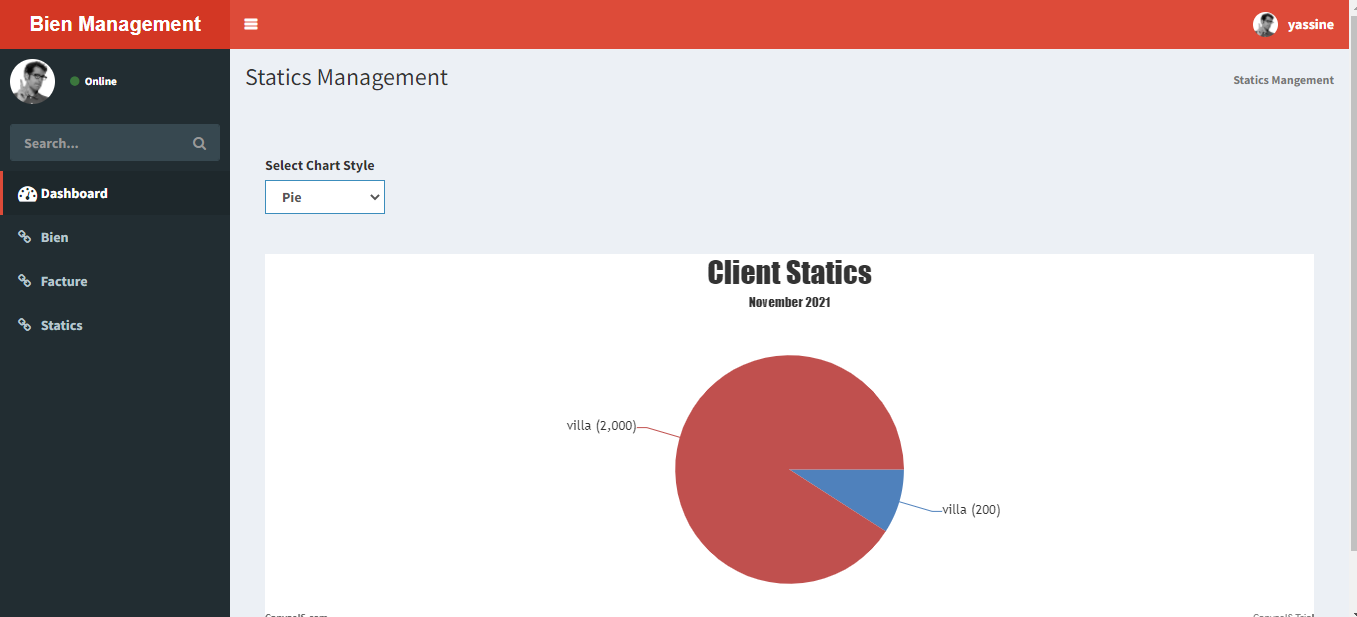


**Figure 34:Ecran de facture.**

#### Interface de visualiser les statistiques.

Cette interface permet à l’admin de visualiser les statistiques.





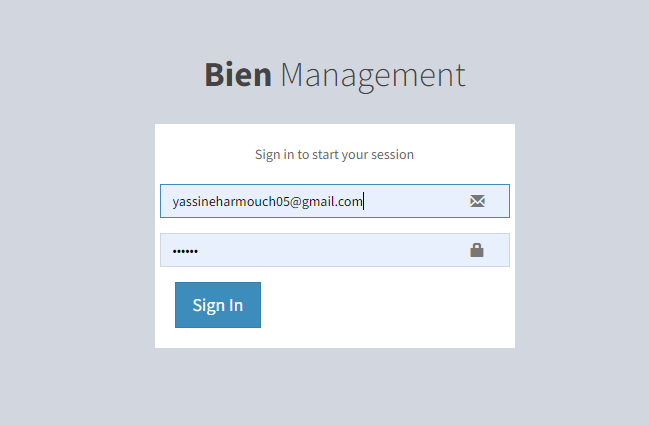
**Figure 35:Ecran de visualiser les statistiques**

### Partie Agence de location

La partie Agence de location permet à celui-ci :

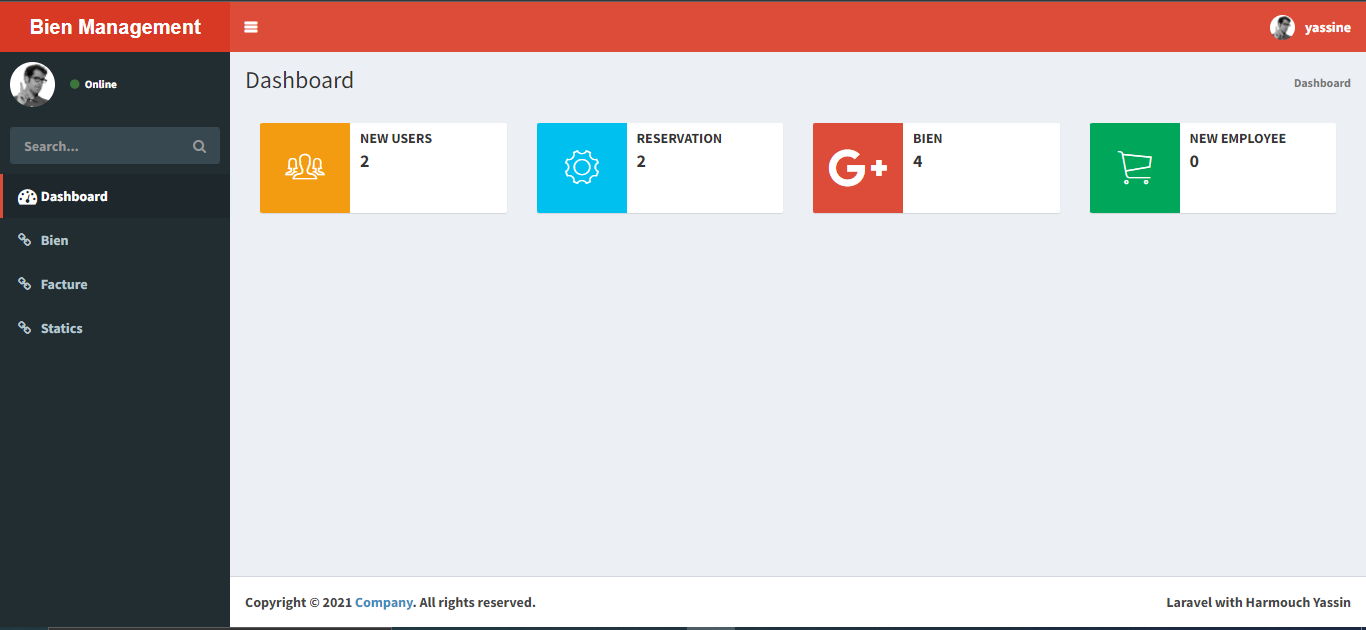
* Visualiser les statistiques.
* Gérer les réservations.
* Imprimer les factures.

#### Interfaces de se connecter



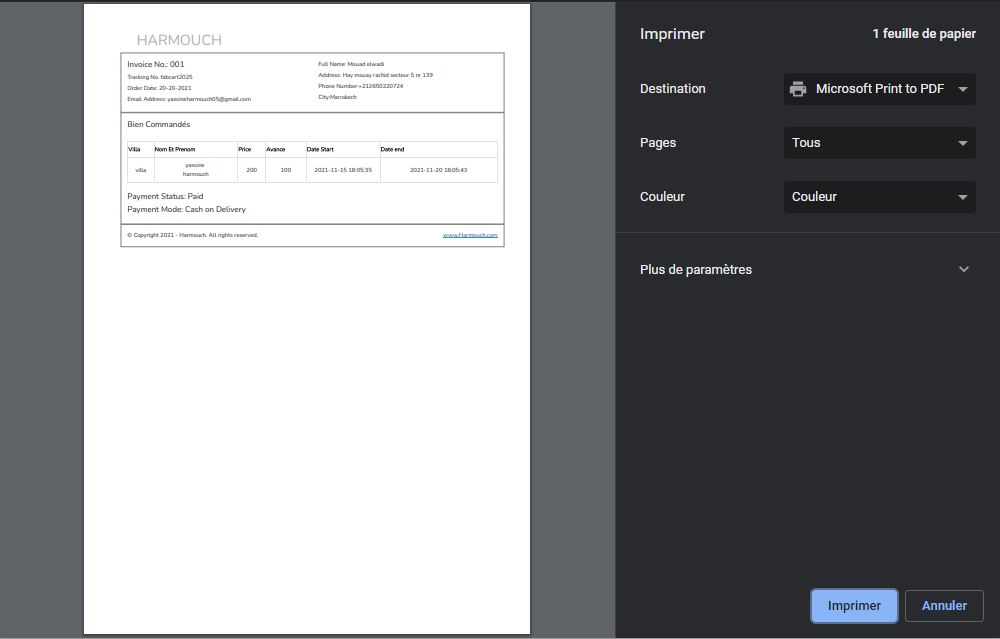
**Figure 36:Ecran de l’interface de se connecter.**

#### Interface d’agence de location.



**Figure 37:Ecran de l’interface agence.**

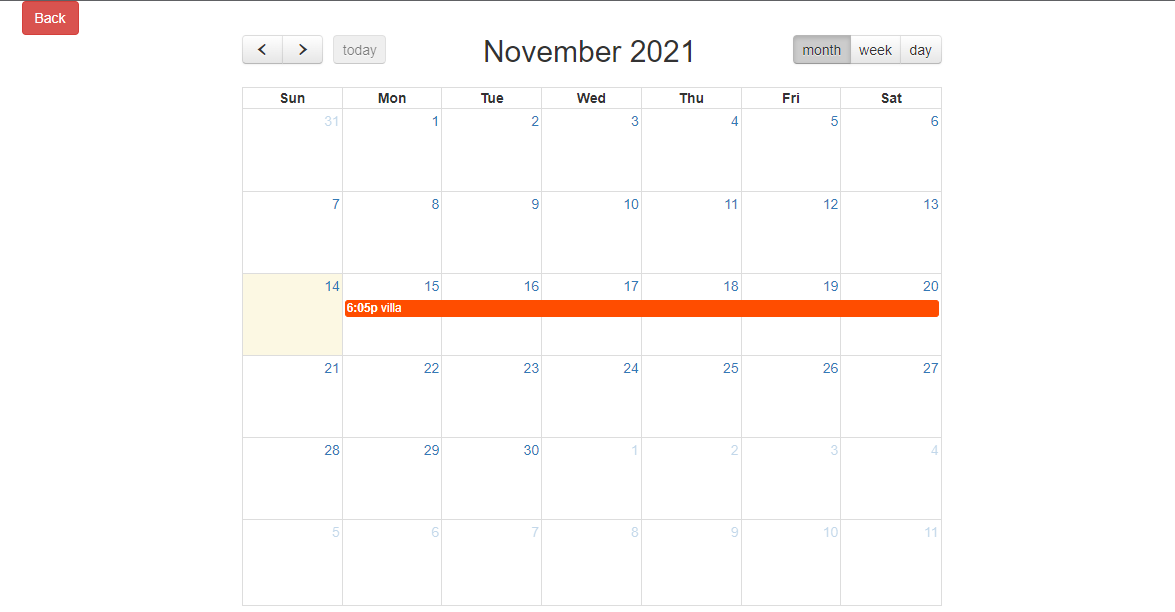
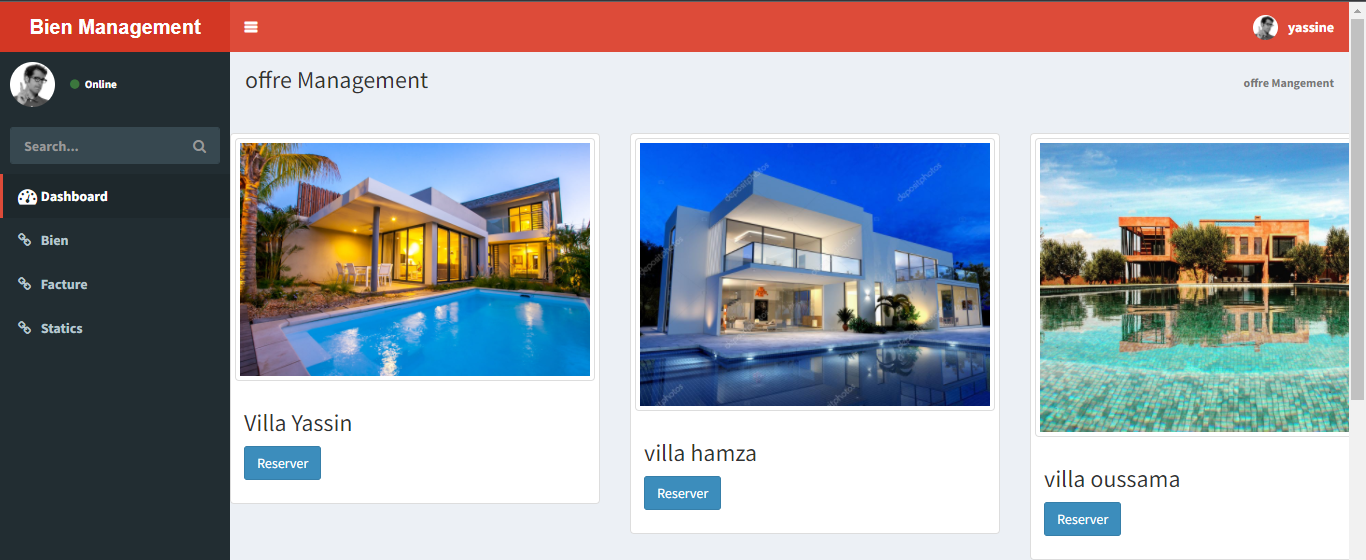
#### Interface d’impression de facture



**Figure 37:Ecran de l'impression de facture.**

#### Interfaces de gérer les réservations.

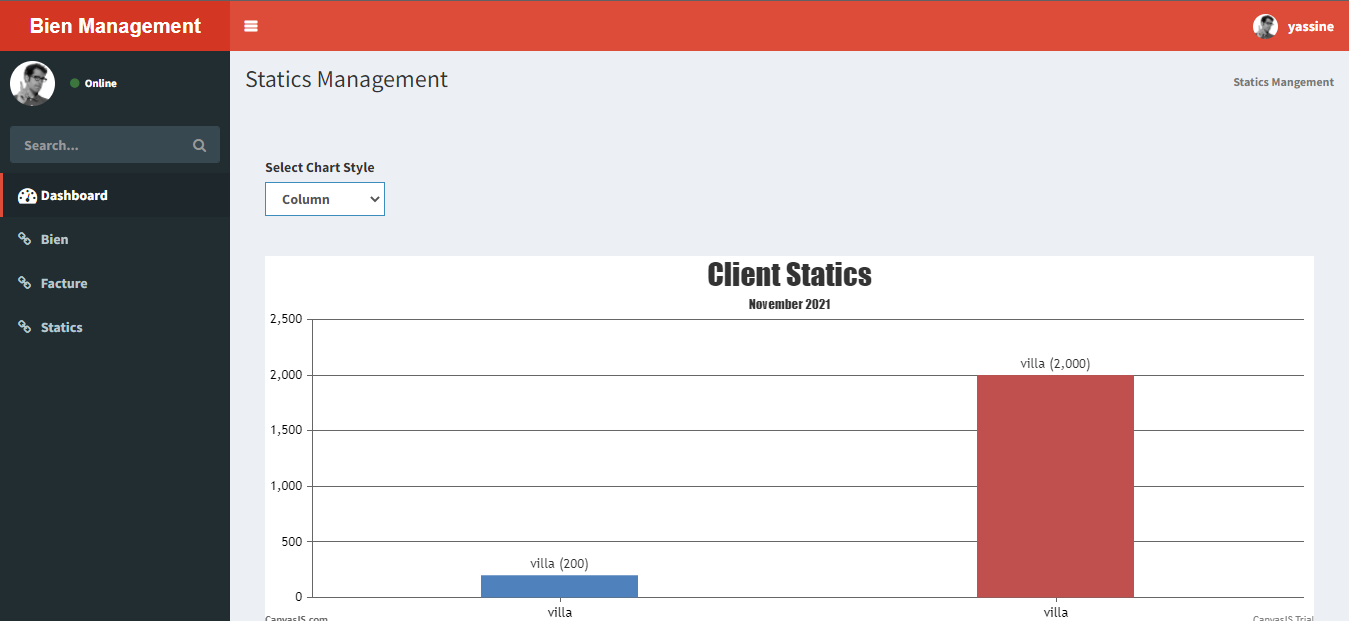
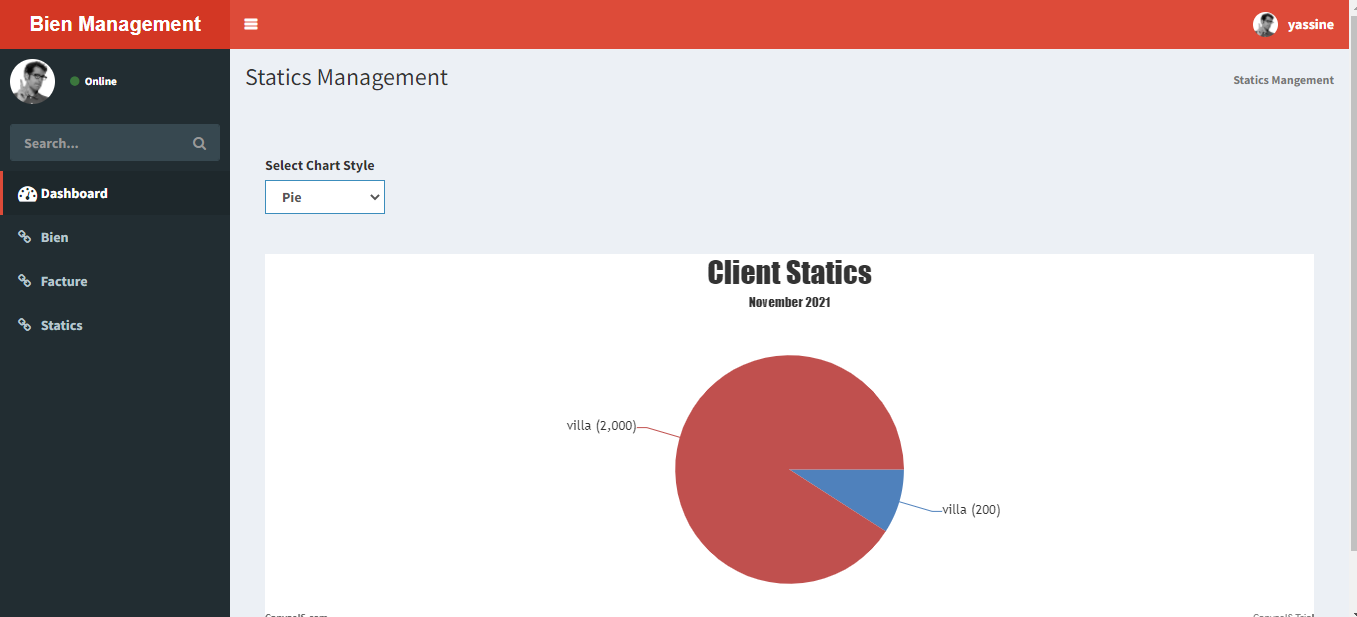
L’interface permet à l’agence de faire des réservations.



**Figure 38:Ecran de gérer les réservations.**

#### Interfaces pour visualiser les statistiques.

Cette interface permet à l’agence de visualiser les statistiques.



**Figure 39:Ecran de visualiser les statistiques.**

# Conclusion générale :

L’objectif de notre projet et de développer une application de gestion locative permettant de générer des réservations qu'on pourra visualiser et gérer à partir du web .

Le point de départ de la réalisation de ce projet était une récolte d’informations nécessaires pour

dresser un état de l’existant, présenter un aperçu sur la problématique

.

Par la suite, nous nous sommes intéressés à l’analyse et la spécification des besoins qui nous a

permis de distinguer les différents acteurs interagissant avec l’application visée.

L’objectif de la partie suivante était la conception détaillée, dans laquelle nous avons fixé la

structure globale de l’application. Le dernier volet de notre projet était la partie réalisation qui

a été consacrée à la présentation des outils du travail et les interfaces les plus significatives de

notre application.

L’apport de ce travail a été d’une importance très considérable. En effet, il nous a permis : de

suivre une méthodologie de travail bien étudiée, d’approfondir nos connaissances dans le

monde de développement des applications et de bien nous exercer sur le Framework "LARAVEL".

Un des points importants que nous avons constaté durant nos travaux, est l’importance du travail

de groupe , comment compter sur soi pour résoudre les problèmes lorsqu’ils se présentent, comment

être méticuleux dans notre travail, comment être attentifs aux indications de nos supérieurs, comment

être bien organisés pour accomplir dans les meilleurs délais et meilleures conditions les tâches qui

nous sont confiées.

En perspectives cette application pourrait être améliorée et enrichie par quelques modules ou interfaces

pour mieux l’adopter aux besoins de l’utiliser .