

# TABLE DES MATIÈRES

Liste des figures .....	7-8
Liste des tableaux .....	9
Liste des abréviations .....	10
Introduction générale.....	11-12
1. Chapitre 1 : Etude de l'existant des systèmes de réservation en ligne .....	13
Introduction .....	13
1.1.    Définition de e-commerce .....	14
1.2.    Définition du terme e-réputation :.....	14
1.3.    Définition des plateformes de réservation :.....	15
1.4    Etude de l'existant :.....	15-16
1.4.1.    Les fonctionnalités de quelques plateformes existantes :.....	17
1.4.1.1.    Plateforme « Airbnb »:.....	17
1.4.1.2.    Plateforme « Nbatou »:.....	17
1.5.    Etude du besoin sur le marché :.....	18
1.5.1.    Google trends : .....	18-19
1.5.2.    Sondage Google form : .....	20-23
Conclusion.....	23
2. Chapitre 2 : Analyse et conception du système « Kriwli » .....	24
Introduction .....	25
2.1.    Spécification des besoins du système.....	25
2.1.1.    Identification des acteurs :.....	25

2.1.2. Besoins fonctionnels : .....	26
2.1.3. Besoins non fonctionnels : .....	26
2.2. Modélisation des besoins fonctionnels :.....	27
2.2.1. Diagramme de cas d'utilisation :.....	27-28
2.2.2. Diagramme de séquence :.....	29
2.2.2.1. Diagramme de séquence « ajouter un hébergement » : .....	29
2.2.2.1.1. Description textuelle : .....	29-30
2.2.2.1.2. Diagramme de séquence système : .....	30-31
2.2.2.2. Diagramme de séquence « Faire une réservation » : .....	32
2.2.2.2.1. Description textuelle : .....	32
2.2.2.2.2. Diagramme de séquence système : .....	33
2.2.3. Diagramme de classe:.....	34
2.2.4. Modèle logique de données :.....	35
Conclusion.....	36
3. Chapitre 3 : Implémentation du système .....	37
Introduction .....	38
3.1. Les technologies utilisées :.....	38
3.1.1. Laravel .....	38
3.1.2. Bootstrap .....	38
3.2. Les outils utilisés :.....	38-39
3.2.1. Google trends : .....	38
3.2.2. Atom .....	39
3.2.3. PhpMyAdmin .....	39
3.2.4. Modelio .....	39

3.2.5.	Wamp .....	39
3.2.6.	MySqlWorkbench .....	39
3.3.	L'architecture de l'application : .....	40
3.4	Maquettes du système : .....	41-43
3.5.	L'implémentation de « Kriwli » :.....	44
3.5.1.	Installation et organisation : .....	45
3.5.2.	Le routes .....	46
3.5.3.	Les migrations .....	46-47
3.5.4.	Les classes model .....	47
3.5.5.	Contrôleur.....	48
3.5.6.	Les vues .....	49
	Conclusion.....	49
	Conclusion générale et perspectives.....	50-51
	Références bibliographiques et webographiques .....	52
	Annexes .....	53-54
	Résumé .....	55-56

# Liste des figures

**Figure 1** : image représentant le taux de recherche de la phrase « location appartement » dans le monde.

**Figure 2** : image représentant le taux de recherche de la phrase « location appartement » en Algérie.

**Figure 3** : image représentant la comparaison entre les différents moyens de recherches utilisés en Algérie pour trouver un appartement en ligne.

**Figure 4** : statistiques reflétant la fréquence à laquelle les voyageurs séjournent.

**Figure 5** : statistiques donnant les destinations favorites des voyageurs algériens.

**Figure 6** : statistiques représentant la méthode de recherche d'hébergement.

**Figure 7** : taux de réservation en ligne réalisé en Algérie.

**Figure 8** : taux de personnes souhaitant faire une réservation en ligne.

**Figure 9** : les critères de sélection d'hébergement.

**Figure 10** : diagramme de cas d'utilisation pour le système de réservation d'hébergement « Kriwli ».

**Figure 11** : Diagramme de séquence pour l'action « ajouter un hébergement ».

**Figure 12** : Diagramme de séquence pour l'action « faire une réservation ».

**Figure 13** : Modélisation des classes de l'application.

**Figure 14** : Base de données de l'application.

**Figure 15** : Page d'accueil partie 1.

**Figure 16** : Page d'accueil qualpartie 2.

**Figure 17** : Page d'accueil partie 3.

**Figure 18** : Page d'enregistrement.

**Figure 19** : Page d'ajout d'hébergement.

**Figure 20** : Page de réservation.

**Figure 21** : Architecture MVC.

**Figure 22** : Dossier d'installation laravel.

**Figure 23** : Création du framework laravel.

**Figure 24** : Figure représentant la page d'accueil de laravel.

**Figure 25** : Figure représentant le fichier route web.php .

**Figure 26 :** Figure représentant la migration de la table utilisateur.

**Figure 27 :** Figure représentant la classe modèle utilisateur.

**Figure 28 :** Figure représentant la contrôleur utilisateur.

**Figure 29 :** Vue connexion d'utilisateur.

## Liste des tableaux

**Tableau 1 :** Caractéristiques de qualité des différentes applications.

**Tableau 2 :** Description textuelle du cas d'utilisation « Ajouter un hébergement ».

**Tableau 3 :** Description textuelle du cas d'utilisation « Faire une réservation ».

## Liste des abréviations

- CRUD = Create, Read, Update, Delete : Créer, Lire, Mettre à jour, Supprimer
- UML = Unified Modeling Language : Langage de modélisation unifié.
- MVC = Modèle-Vue-Contrôleur
- HTTP = Hypertext Transfer Protocol : Protocole de transfert hypertexte.
- ORM = Object-relational mapping : mapping relationnel objet.
- IBE = Internet Booking Engine : Moteur de réservation Internet

# Introduction générale

Aujourd’hui, le besoin digital ne cesse d’augmenter, l’état investit dans le tourisme local et encourage l’initiative algérienne, la nécessité de se précipiter vers une technologie adéquate se voit comme une aubaine pour chaque porteur de projet.

Une des utilisations technologiques les plus répandues à travers le monde qui tend à se répandre dans notre pays est le paiement et la réservation en ligne que nous retrouvons dans la plupart des sites web ainsi qu’applications mobiles.

Le marché algérien des réservations d’hébergement reste difficile à déterminer malgré la présence de certaines plateformes, les réservations se font par des recommandations de voyageurs ayant déjà séjournés dans des lieux similaires, pour répondre à l’avancée technologique, il est important de créer des plateformes web initiant l’avancée.

Afin de traiter le sujet et de répondre aux questionnements émis, un plan de recherche a été établi. Il consiste tout d’abord en un questionnaire posté en ligne à intervalles réguliers sur deux mois. Nous avons également mené des entretiens semi-directifs avec des voyageurs. La recherche a été complétée par de nombreuses lectures sur le thème.

Le système de réservation en ligne est un thème concerné par le projet de loi e-commerce qui s’avère être un outil à faire valoir pour développer le commerce électronique en Algérie.

C’est dans ce contexte que s’intègre notre projet de fin d’étude, qui a pour objectif de mettre en place une plateforme de réservation d’hébergement en ligne.

Ce rapport se définit sur trois chapitres, le premier chapitre contiendra une description des plateformes de réservation en ligne puis exposera l’étude de l’existant ainsi que l’étude du besoin du marché afin de répondre à la réelle demande. L’étude du besoin se réalisera en prenant en considération l’outil google trends utilisé aussi comme baromètre pour définir les tendances du public dans les plateformes de réservation d’hébergement en ligne nationales et internationales.

Au deuxième chapitre nous montrerons l’analyse et la conception du système qui contiendra différents diagrammes ainsi que les spécifications des besoins systèmes et cela avant d’entamer la partie applicative dans le but de bien définir, clarifier les fonctionnalités importantes de l’application. Nous aurons pour aspect de définir des besoins fonctionnels et

non fonctionnels qui sont considérés comme des fonctionnalités du système ou les besoins spécifiant un comportement d'entrée / sortie du système.

Le troisième chapitre portera sur la réalisation de notre application web, et comportera la conception architecturale ainsi que l'environnement de développement, nous détaillerons les différentes étapes de réalisation. Ce chapitre représente la dernière partie de ce rapport.

Nous clôturerons ce rapport par une conclusion générale dans laquelle nous proposerons aussi des perspectives dans le but de rendre notre travail meilleur et encore plus ciblé et segmenté sur une population précise.

# Chapitre 1

Etude de l'existant des  
systèmes de réservation

# 1. Chapitre 1 : Etude de l'existant des systèmes de réservation

## Introduction

Le marché algérien des réservations d'hébergement est complexe et peu structuré, le bouche à oreille reste le moyen le plus utilisé par les voyageurs, il n'existe presque aucun type de réservation en ligne. De cela nous distinguons, donc, que les réservations en ligne ne s'effectuent que dans des sites multinationaux, les réservations d'hébergement se déroulent d'une façon purement ancienne. Et pour répondre au développement technologique, au progrès des plateformes, il est vraiment primordial de créer une plateforme web pour la réservation d'hébergement en ligne.

### 1.1. Définition de e-commerce

Le e-commerce qui intitule commerce électronique se définit comme un groupement de transactions commerciales s'opérant à distance via les réseaux internet sous différents types de terminaux numériques tels que les ordinateurs, tablettes, smartphones et objets connectés.

Ce commerce virtuel se développe d'une façon exponentielle depuis les années 2000 jusqu'à aujourd'hui où l'achat en ligne devient un quotidien incontournable. Le m-commerce est une des composantes du e-commerce qui possède la particularité d'utilisation sur mobile.

Le e-commerce ne recouvre qu'une partie des enjeux du commerce sur Internet car la présence des entreprises sur internet influence également les achats réalisés en points de vente ou par d'autres canaux<sup>1</sup>.

### 1.2. Définition du e-réputation

---

<sup>1</sup> Les autres canaux qui peuvent exister sont représentés par : content marketing, pay per click, email marketing

Généralement l'e-réputation est considérée comme l'image véhiculée et /ou subie par une entreprise ou une marque sur Internet et autres supports numériques. Elle s'applique également à un produit ou service. L'e-réputation est le résultat en termes d'image des contenus produits et diffusés par l'entreprise, mais aussi et surtout celui des contenus produits par les internautes sur les blogs, réseaux sociaux, plateformes d'échanges vidéo, forums et autres espaces communautaires. [1]

### **1.3. Définition des plateformes de réservation**

Une plateforme de réservation sur internet est un site web qui permet aux consommateurs et aux agences de voyages de réserver des vols, des hôtels, des forfaits vacances, des assurances et d'autres services en ligne. Les IBE sont aujourd'hui utilisés par presque toutes les compagnies aériennes pour réduire les dépenses supplémentaires et pour permettre la réservation et le paiement instantané.

### **1.4. Etude de l'existant**

Nous avons pu recenser plusieurs plateformes répondant à un certain besoin parmi ces plateformes Nbatou ainsi que Djumia-House et Lkeria, nous avons défini les critères de qualité pour l'évaluation d'applications web.

Les indicateurs de qualité cités dans ce tableau proviennent du document référencé dans la partie bibliographie [3].

La signification exacte de chaque critère se définit :

- Fonctionnalités : se représente dans l'exactitude, la sécurité et la conformité.
- Fiabilité : maturité, tolérances aux fautes.
- Utilisabilité : compréhensibilité, facilité d'apprentissage.
- Rendement : performance, ressource.
- Maintenabilité : facilité d'analyse, facilité de modification.
- Portabilité : facilité d'adaptation, facilité d'installation, conformité.

Le tableau 1 représente les quatre plateformes étudiées qui permettent de définir les fonctionnalités à prendre en considération, ainsi que l'amélioration pouvant être ajoutée par notre application.

	NBATOU	DJUMIA-HOUSE	LKERIA
FONCTIONNALITÉ	7/10	9/10	7/10
FIABILITÉ	6/10	8/10	7/10
UTILISABILITÉ	5/10	6/10	6/10
MAINTENABILITÉ	4/10	5/10	5/10
RENDEMENT	8/10	4/10	4/10
PORTABILITÉ	9/10	8/10	4/10

*Tableau 1 : Caractéristiques de qualité des différentes applications*

Le système de notation a été réalisé en suivant certaines normes qui existent déjà sur le net, comme par exemple sur la partie sécurité, nous avons utilisé le site de test Sucuri qui est l'un des scanners gratuits le plus célèbre et le plus utilisé. Grâce à ce test, nous pouvons vérifier si le site est infesté de malwares, s'il est blacklisté, s'il est utilisé pour spammer ou s'il n'est pas victime d'un déacement.

## 1.4.1. Les fonctionnalités de quelques plateformes existantes

### 1.4.1.1 Plateforme « Airbnb »

Airbnb est une plateforme référence de réservation en ligne, créée en 2007 par deux américains, le site contient plus 1,5 millions d'annonces.

Les différentes fonctionnalités que nous retrouvons sur la plateforme sont :

- Rechercher hébergement
- Réserver hébergement
- Inscription/ Connexion
- Héberger
- Ajouter une nouvelle annonce
- Supprimer une annonce
- Demander une recommandation
- Payer
- Noter
- Enregistrer appartement
- Ajouter commentaire
- Rechercher dans les commentaires

### 1.4.1.2 Plateforme « Nbatou »

Nbatou est une plateforme de réservation en ligne algérienne, qui offre des solutions d'hébergements depuis 2015.

Les différentes fonctionnalités que nous retrouvons sur la plateforme sont :

- Inscription/ Connexion.
- Louer ma maison.
- Trouver hébergement.
- Calculer prix.
- Réserver hébergement.
- Supprimer réservation.
- Rechercher dans les commentaires.
- Ajouter commentaire.

## 1.5. Etude du besoin sur le marché

L'objectif de notre projet est de concevoir une solution qui doit répondre principalement à un certain besoin précis et bien élaboré et résoudre un problème existant et persistant, pour cela l'étape étude du besoin est nécessaire pour bien définir notre cible. Nous utilisons l'outil « Google Trends » fournisseur de tendance, ainsi qu'un formulaire « Google form » sur un échantillon varié.

### 1.5.1. Utilisation de l'outil « Google trends »

Google Trends permet de connaître les tendances de la thématique étudiée, cela nous aide dans notre projet à déterminer le besoin réel des futurs utilisateurs et nous permet de déterminer les fonctionnalités adéquates au système.

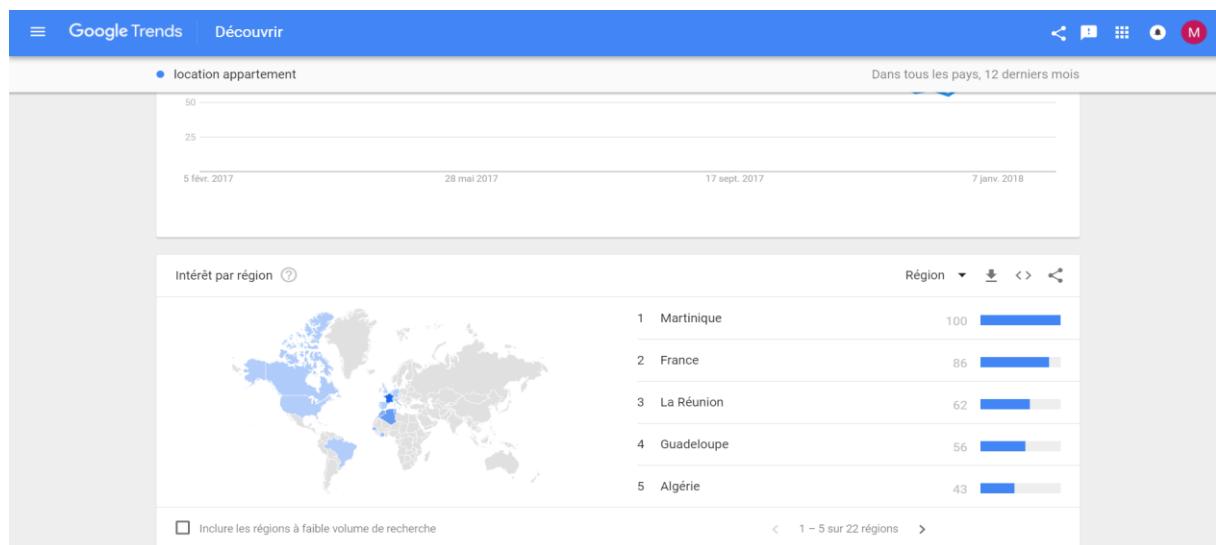
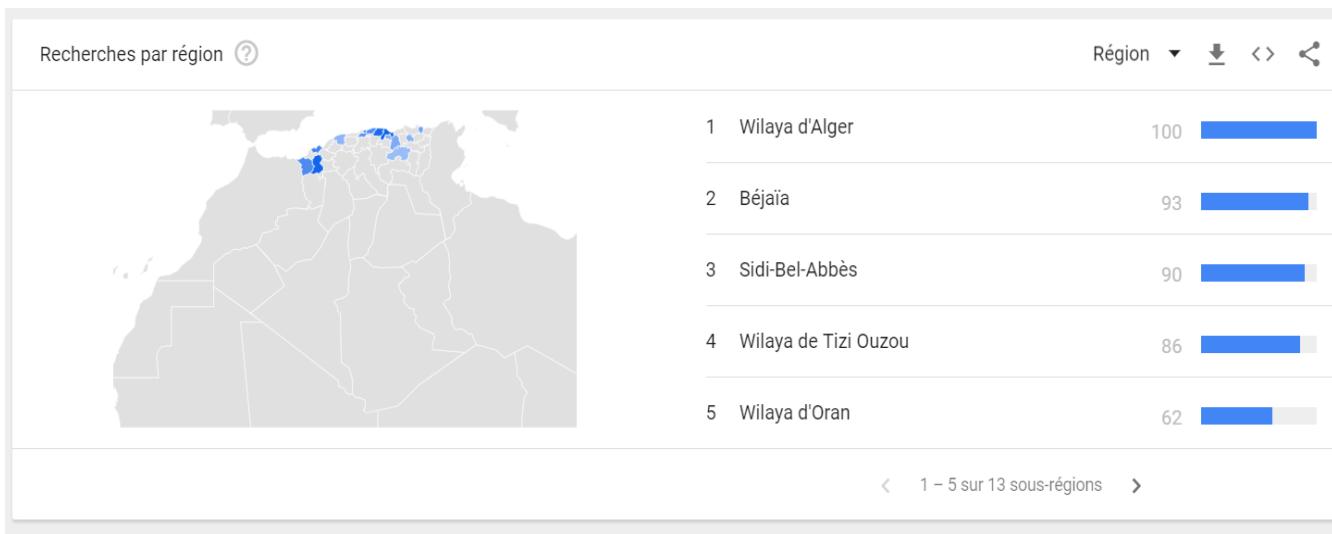


Figure 1 : image représentant le taux de recherche de la phrase « location appartement » dans le monde

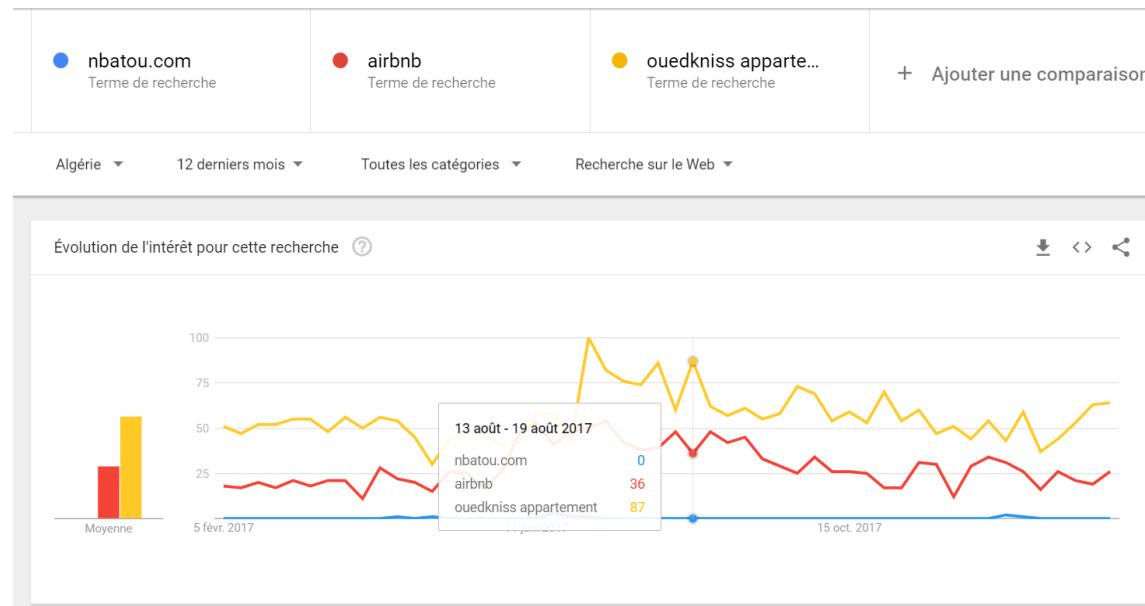
Sur la figure précédente (figure 1), l'Algérie se classe 5 -ème à l'échelle mondial dans la recherche de la phrase « location appartement », ce qui nous permet de reconnaître un besoin dans cette région.



**Figure 2 :** image représentant le taux de recherche de la phrase « location appartement » en Algérie

La figure 2 représente les villes ayant un flux important de recherche d'appartement en location, ce qui montre l'intérêt porté à notre futur système.

La figure suivante (figure 3) montre l'impact des différentes plateformes sur le marché national.



**Figure 3 :** image représentant la comparaison entre les différents moyens de recherches utilisés en Algérie pour trouver un appartement en ligne.

### 1.5.2. Réalisation du sondage « Google form »

Le sondage a été réalisé sur un échantillon de 132 personnes entre la période du 1 er février au 15 mars sur un ensemble de 22 villes algériennes sur des personnes ayant entre 18 et 65 ans. Les questions posées lors de ce sondage nous permettent de déterminer les fonctionnalités nécessaires de notre plateforme réservation d'hébergement en ligne dans ce projet.

Les figures 4 et 5 représentent la fréquence à laquelle les voyageurs séjournent en Algérie, ainsi que les destinations choisies.

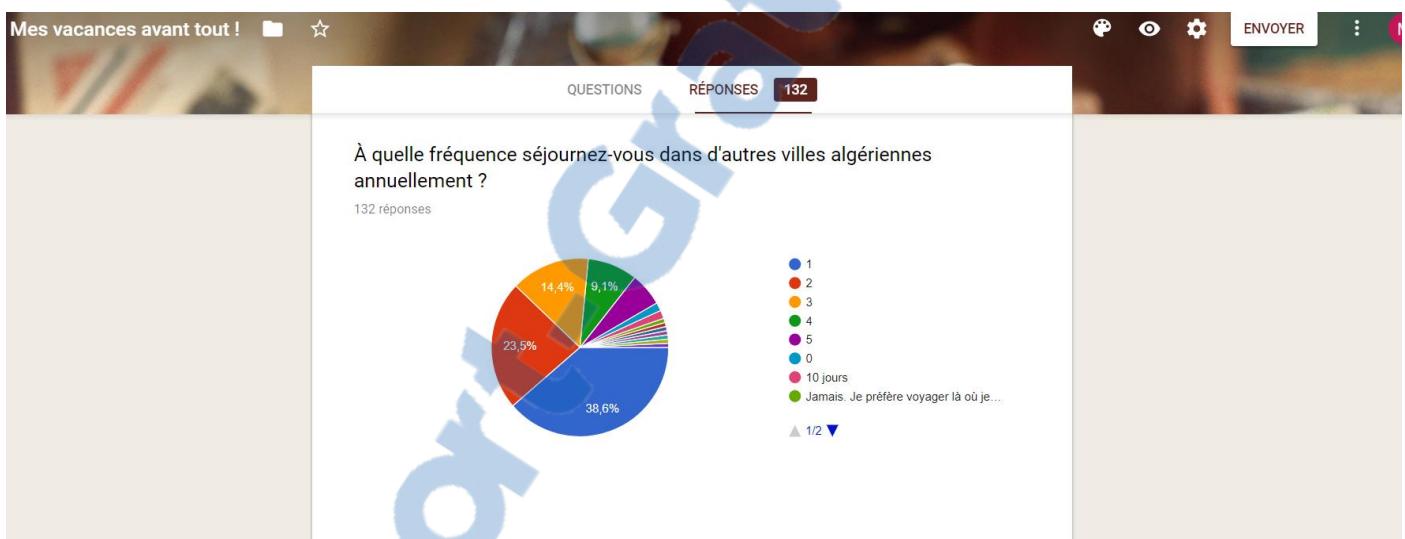


Figure 4 : statistiques reflétant la fréquence à laquelle les voyageurs séjournent

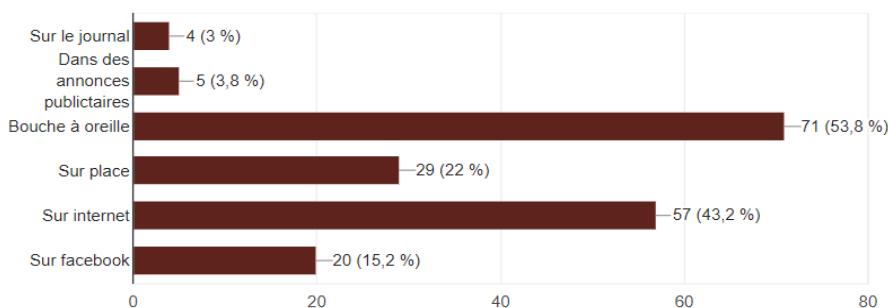


**Figure 5 :** statistiques donnant les destinations favorites des voyageurs algériens.

Comme nous pouvons le constater sur la figure suivante (figure 6), la méthode de recherche d'hébergement favorite reste le bouche à oreille suivi par la recherche sur internet, d'où l'importance accordé à la recommandation ainsi qu'à la rapidité de la réservation sur internet.

#### Quelle est votre méthode de recherche d'hébergement ?

132 réponses

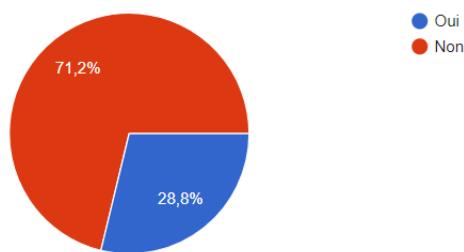


**Figure 6 :** statistiques représentant la méthode de recherche d'hébergement

Sur la figure 7, nous remarquons le taux élevé des personnes ne réalisant pas de réservation en ligne ce qui se traduit par un passage de sensibilisation censé, en revanche nous pouvons comprendre en voyant sur la figure 8, que l'utilisateur est prêt à faire une réservation en ligne.

Aviez-vous déjà fait des réservations en ligne sur des plateformes ?

132 réponses



**Figure 7 : Taux de réservation en ligne réalisé en Algérie**

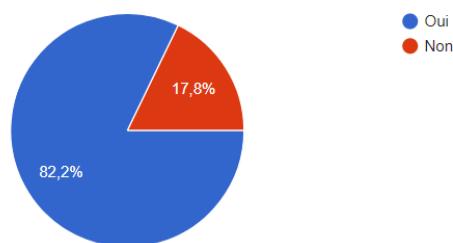
QUESTIONS

RÉPONSES

132

Si non, seriez-vous prêt à réserver en ligne (logement/hôtel)?\*

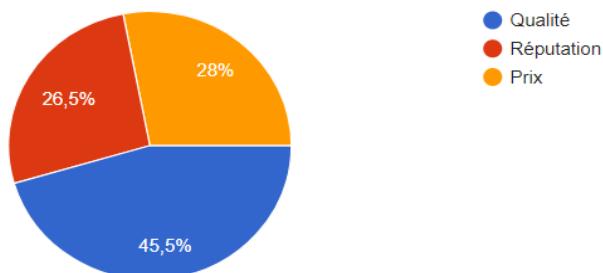
129 réponses



**Figure 8 : taux de personnes souhaitant faire une réservation en ligne**

Lequel des éléments suivants influencerait le plus votre décision si vous deviez chercher à trouver un séjour dans une autre ville

132 réponses



**Figure 9 : les critères de sélection d'hébergement**

## Conclusion

Durant ce chapitre nous avons pu définir les différents besoins existants, nous avons pu aussi donner un contexte général à notre projet de fin d'étude, nous avons établi aussi une étude des systèmes existants qui nous permettent de connaître les fonctionnalités primordiales. Dans le chapitre suivant, nous allons entamer la phase d'analyse et de conception du système.

# Chapitre 2

## Analyse et conception du système Kriwli

## 2. Chapitre 2 : Analyse et conception du système « kriwli »

### Introduction

Avant de commencer à coder la partie applicative, nous nous intéressons à la phase de spécification pour bien définir, clarifier les grandes fonctionnalités de notre application. Ce chapitre consiste à donner une définition précise des besoins fonctionnels et non fonctionnels ainsi que les objectifs visés.

#### 2.1. Spécification des besoins du système

L'analyse de la thématique et des différentes problématiques posées par les outils existants ainsi que la compréhension des besoins utilisateurs a permis de dégager les fonctionnalités qu'offre notre application finale. Les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement seront décrites par la suite comme étant besoins non fonctionnels.

##### 2.1.1. Identification des acteurs

Les acteurs que nous avons réussi à identifier sont :

- **Le loueur :** c'est l'acteur qui a pour rôle principale de réserver les hébergements
- **L'hôte :** c'est l'acteur chargé de rajouter l'hébergement en sa possession.

### **2.1.2. Besoins fonctionnels**

Dans cette section, nous représentons l'ensemble des besoins fonctionnels auxquels devrait répondre notre application web.

Les besoins fonctionnels et les attentes par rapport à notre application dépendent de la nature de l'acteur. Pour cela, nous avons décrit pour chaque acteur les besoins fonctionnels qui lui sont reliés.

Les besoins fonctionnels auxquels notre application doit répondre se résument dans les points suivants :

#### **Le système doit permettre**

- L'authentification des utilisateurs par un login et un mot de passe pour accéder aux différentes fonctionnalités.

#### **Le système doit permettre au loueur de**

- Gérer les réservations.
- Rechercher un hébergement.
- Pré-réservation.
- Ajouter un commentaire.
- Partager l'hébergement.

#### **Le système doit permettre à l'hébergeur de**

- Gérer les hébergements.
- Gérer les dates de disponibilités.
- Gérer les commentaires.

### **2.1.3. Besoins non fonctionnels**

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation comme le langage de programmation, le type du SGBD et du système d'Exploitation [2]

L'ensemble des extensions à réaliser doivent respecter les besoins suivants :

**Ergonomie de l'interface** : L'application doit être facile à utiliser, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

**Fiabilité** : Les informations apportées par l'application doivent être fiables et sûres

**Disponibilité** : L'application doit être disponible à tout instant pour être utilisée par n'importe quel utilisateur.

**Sécurité** : L'application comporte des informations personnelles et sensibles, donc elle doit respecter les règles relatives à la sécurité des systèmes informatiques.

**La performance** : L'application doit être performante c'est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d'une manière optimale.

## 2.2. Modélisation des besoins fonctionnels

La recherche ciblée des besoins fonctionnels, est considérée nécessaire avant d'entamer la conception afin d'obtenir une vue globale sur les exigences de notre application.

### 2.2.1. Diagramme de cas d'utilisation

La connaissance des fonctionnalités à implémenter est primordiale pour établir le diagramme de cas d'utilisation de l'application.

Dans cette section, nous allons détailler le diagramme de cas d'utilisation réalisé.

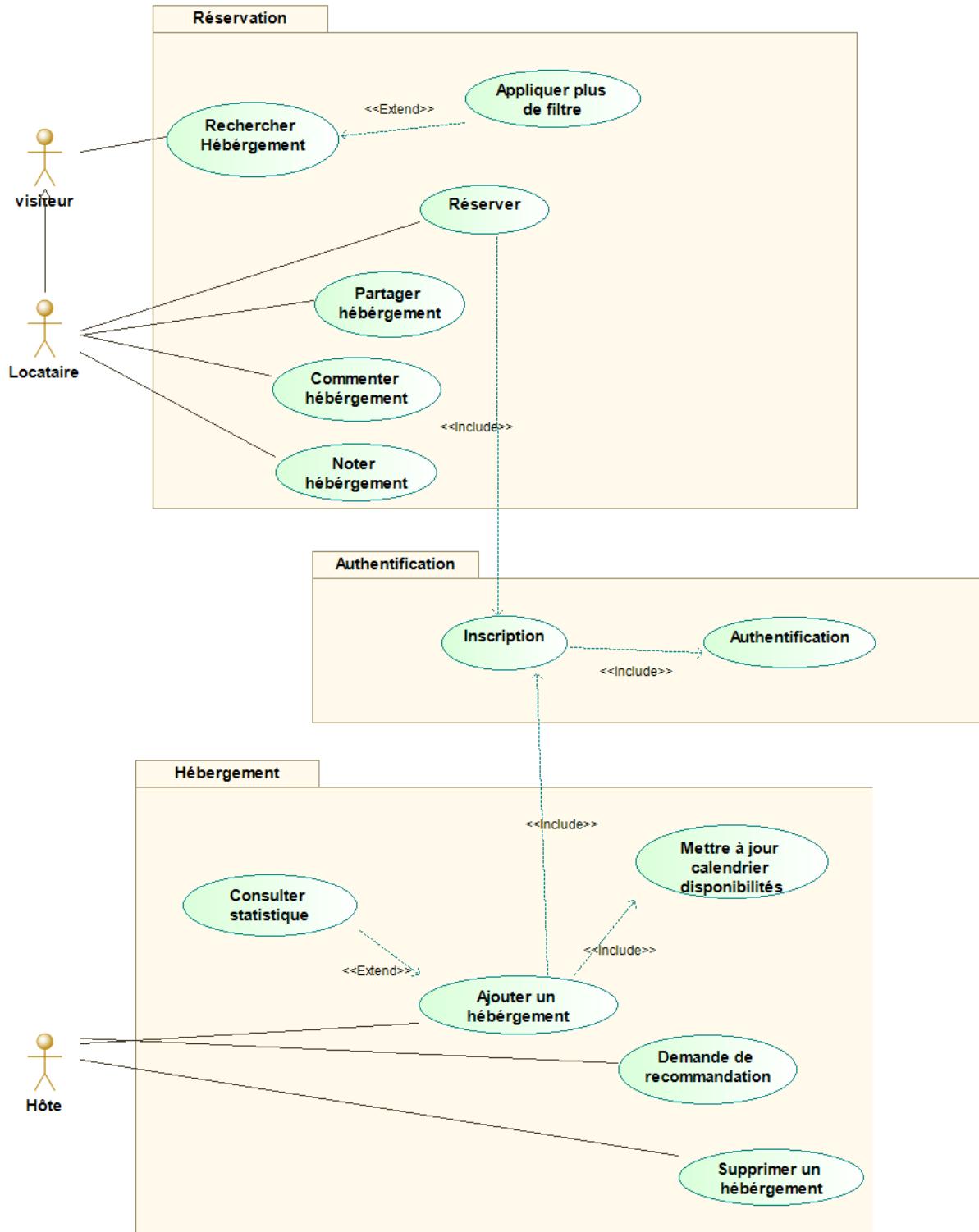


Figure 10 : diagramme de cas d'utilisation pour le système de réservation d'hébergement « Kriwli ».

La figure ci-dessus représente le diagramme de cas d'utilisation principal, à savoir celui concernant la gestion des réservations et consultations. Comme nous pouvons le constater, les deux acteurs principaux de ce diagramme sont le visiteur ainsi que le locataire.

L'application doit donc pouvoir leur permettre de gérer les hébergements et réservations, et pour cela implémenter des fonctionnalités comme la consultation de la liste des hébergements, la recherche ainsi que toutes les fonctionnalités CRUD en général.

Les hôtes sont également des acteurs de ce diagramme, leur rôle ici se résume à l'ajout, la modification et la suppression d'hébergement.

## 2.2.2. Diagramme de séquence

### 2.2.2.1 Diagramme de séquence « Ajouter hébergement »

L'authentification est une tâche primordiale en vue de limiter l'accès et sécuriser notre application, notre module d'authentification est basé sur la confirmation par "Email".

Les différentes interactions sont représentées par le diagramme de séquences de la Figure 11.

#### 2.2.2.1.1 Description textuelle

Le tableau 2 représente la description textuelle du cas d'utilisation « Ajouter un hébergement »

**Scénario 1 : Ajouter hébergement****Objectif :** Ajouter hébergement**Acteurs principaux :** Hôte**Préconditions :** lancement du navigateur**Postconditions :** Affichage de la page d'accueil (succès) ou d'un message d'erreur (échec)**Scénario nominal**

1 : l'hôte saisi l'url du site dans la barre d'adresse du navigateur

2 : le système présente l'interface d'accueil du site

3 : l'hôte sélectionne la partie « Héberger »

4 : le système présente le formulaire d'ajout d'hébergement après la connexion.

**Scénarios alternatifs**

2a : le site est indisponible ; l'url est incorrecte ; le serveur subi un dysfonctionnement

2a1 : le système affiche un message d'erreur

**Tableau 2 : Description textuelle du cas d'utilisation « Ajouter un hébergement »**

### 2.2.2.1.2 Diagramme de séquence système « Ajouter un hébergement »

Cette section concerne le diagramme de séquence modélisant les interactions du cas d'utilisation « Ajouter un hébergement », c'est-à-dire ajouter un hébergement qui peut être réservé avec des renseignements pouvant le décrire correctement.

Le diagramme montre des interactions entre un acteur et le système. L'acteur est celui qui initie la séquence.

À partir de ce diagramme de séquence (figure 11), nous déduisons les opérations suivantes : l'opération se connecter, ajout hébergement, remplir informations, ajouter disponibilités et enfin confirmer l'hébergement.

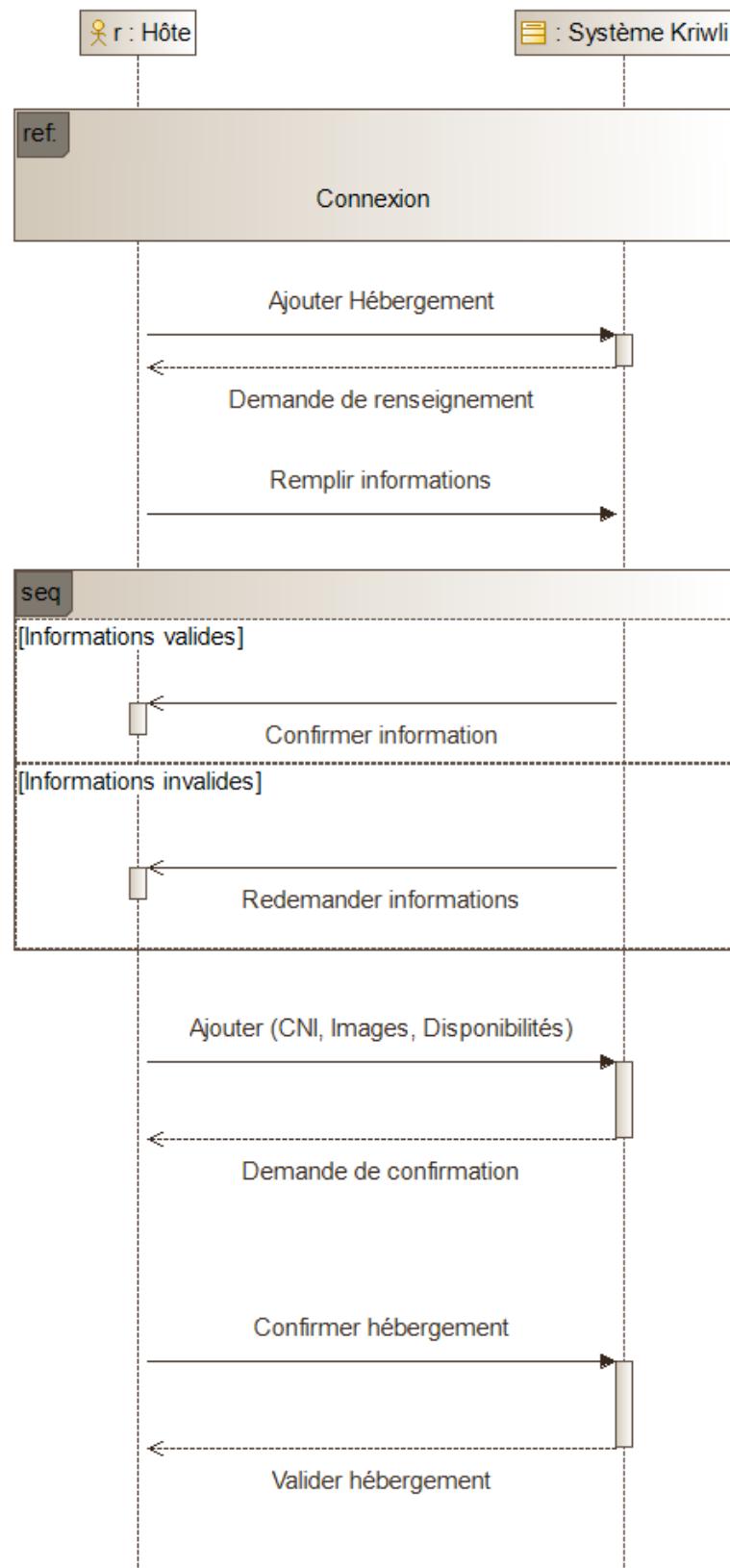


Figure 11 : Diagramme de séquence pour l'action « ajouter un hébergement »

### 2.2.2.2 Diagramme de séquence « Faire une réservation »

La réservation est une importante tâche à savoir celle qui détermine l'objectif de la visite de l'utilisateur.

Les différentes interactions sont représentées par le diagramme de séquences de la Figure 12.

#### 2.2.2.2.1 Description textuelle

Le tableau 3 représente la description textuelle du cas d'utilisation « faire une réservation »

<b>Scénario 2 : Faire une réservation</b>
<b>Objectif :</b> Faire une réservation.
<b>Acteurs principaux :</b> Futur locataire
<b>Préconditions :</b> lancement du navigateur
<b>Postconditions :</b> Affichage de la page d'accueil (succès) ou d'un message d'erreur (échec)
<b>Scénario nominal</b>
1 : Le locataire saisi l'url du site dans la barre d'adresse du navigateur
2 : le système présente l'interface d'accueil du site
3 : Le locataire sélectionne la partie « Réserver »
4 : le système présente la page de confirmation de réservation après la connexion.

*Tableau 3 : Description textuelle du cas d'utilisation « faire une réservation »*

#### 2.2.2.2.2 Diagramme de séquence système « Faire une réservation»

Cette section concerne le diagramme de séquence modélisant les interactions du cas d'utilisation « Faire une réservation », c'est-à-dire obtenir une date de disponibilité d'hébergement et concrétiser une réservation. À partir de ce diagramme de séquence (figure 14), nous déduisons les opérations suivantes : l'opération se connecter, ajout hébergement, recherche hébergement, sélection et enfin réservation.



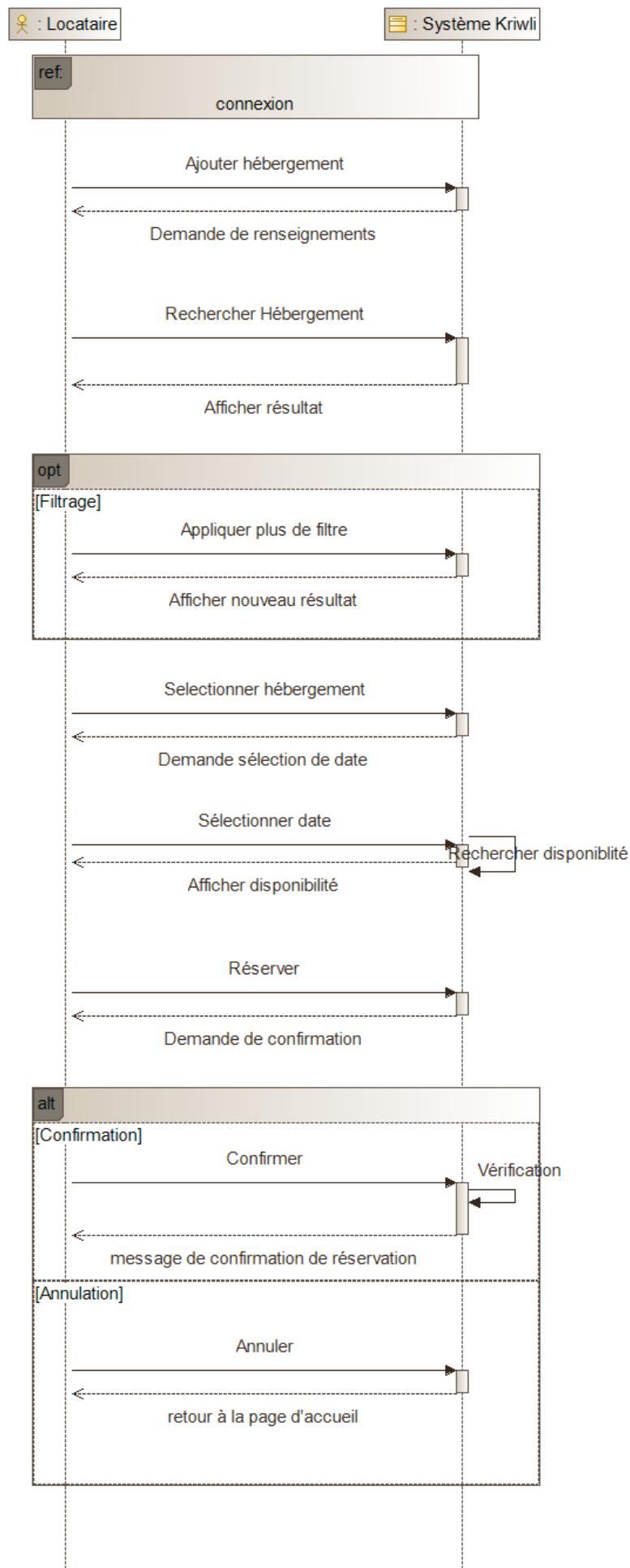


Figure 12 : Diagramme de séquence pour l'action « faire une réservation »

### 2.2.3. Diagramme de classe :

Comme nous pouvons le constater sur la figure 13 l'association est composée des informations suivantes :

- Le diagramme de classe contient 5 classes.
- Un hébergement est consulté par une ou plusieurs personnes, visiteurs/locataires, une fois ce dernier inscrit pourra ajouter un ou plusieurs hébergements.
- Une réservation est réalisée par un ou plusieurs inscrits.

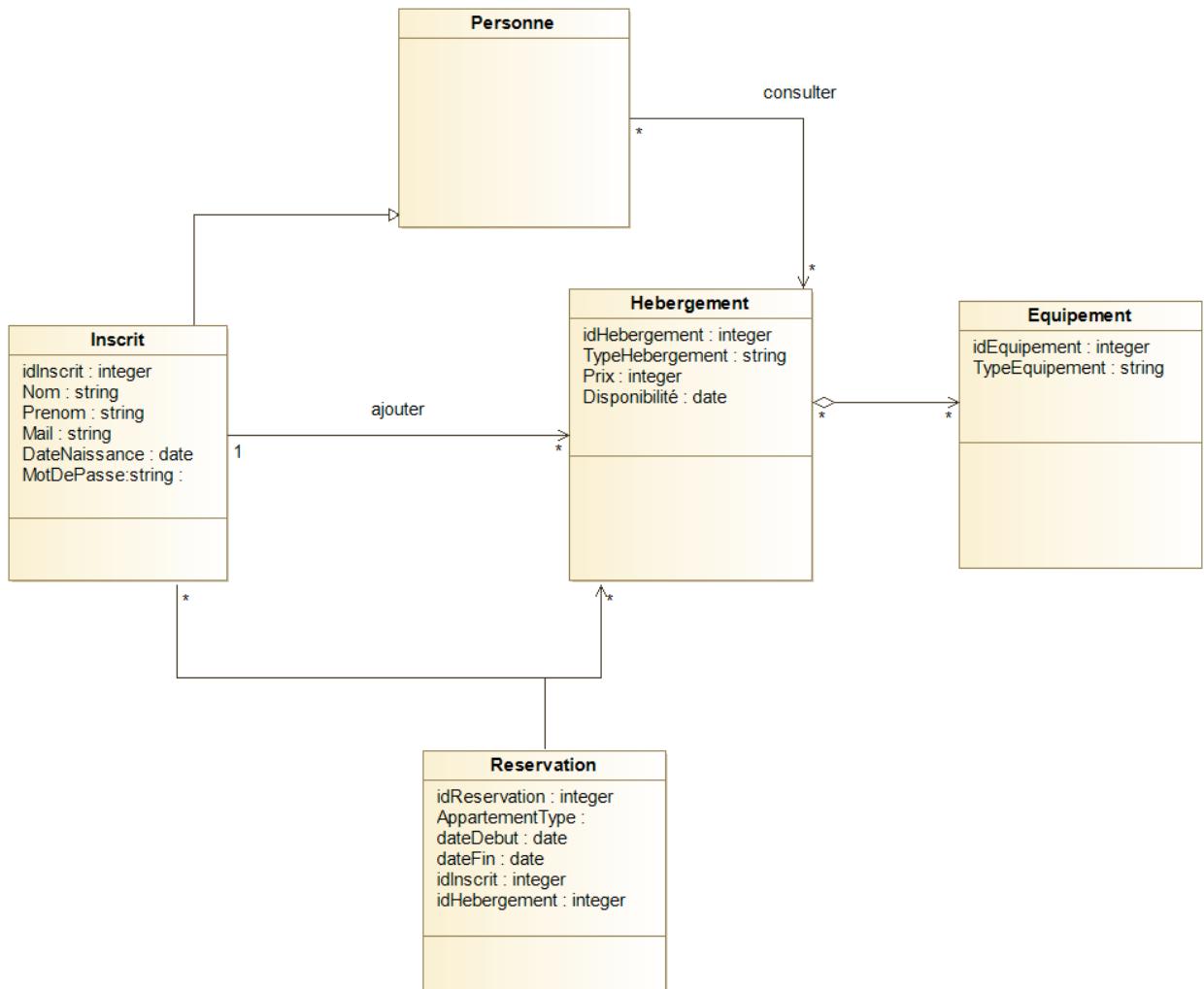


Figure 13 : Modélisation des classes de l'application

## 2.2.4. Modélisation de la base de données de l'application

Nous réalisons la transformation à partir du diagramme de classe pour obtenir le modèle relationnel suivant :

*Inscrit(idInscrit, nom, prenom, dateNaissance, email, motDePasse)*  
*Hébergement(idHébergement, idInscrit\*, idEquipement\*, type, prix, adresse, disponibilités)*  
*Consulter(idConsultation, idPersonne\*, idHébergement\*)*  
*Réservation(idReservation, idInscrit\*, idHébergement\*, dateDebut, dateFin)*  
*Equipement(idEquipement, typeEquipement)*

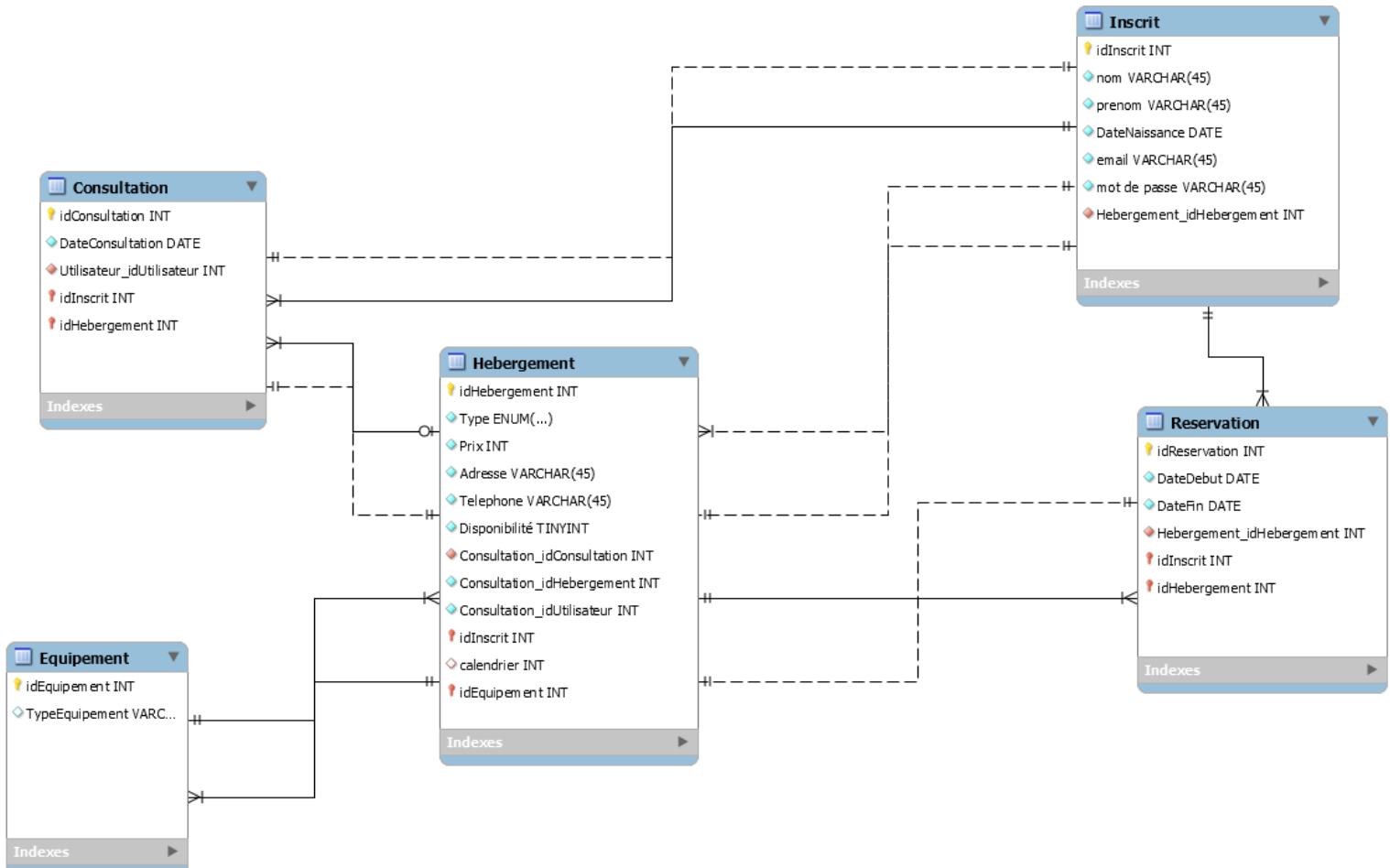


Figure 14 : base de données de l'application

## **Conclusion**

Au cours de ce chapitre, nous avons détailler notre projet, c'est-à-dire les objectifs majeurs à prendre en compte, nous avons aussi réalisé une analyse profonde de la solution adaptée en précisant les différentes fonctionnalités du système.

Dans le chapitre suivant, nous allons entamer la phase de l'implémentation.

# Chapitre 3

## Implémentation du système

### Kriwli

## **3. Chapitre 3 : Implémentation du système**

### **Introduction**

Ce chapitre représente la dernière partie de ce rapport, il traite la phase qui a pour objectif l'implémentation de notre application. Nous débutons, tout d'abord, par la description des technologies utilisées développer notre application. Ensuite nous mentionnons les outils technologiques utilisés. Finalement nous donnons un aperçu sur le travail réalisé en termes de code.

#### **3.1. Les technologies utilisées**

##### **3.1.1. Laravel**

Laravel, initie une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité.

Laravel n'est pas seulement le regroupement de bibliothèques existantes ; c'est aussi un ensemble de nombreux composants originaux et surtout une orchestration de tout cela. [4]

##### **3.1.2. Bootstrap**

Bootstrap est une boîte à outils open source pour le développement avec HTML, CSS et JS, propose un prototypage rapide des idées, permet de créer une application entière, un système de grille réactif, de nombreux composants prédefinis et des plugins puissants construits sur jQuery . [5]

#### **3.2. Les outils utilisés**

### **3.2.1. Google trends**

Google Trends est un outil mis en place par Google pour identifier le nombre de fois où un terme a fait l'objet d'une requête dans son moteur de recherche. Dit autrement, Google Trends est un outil qui permet d'analyser la popularité d'un terme sur le moteur de recherche, dans une période de temps déterminée. [6]

### **3.2.2. Atom**

Atom est un éditeur de texte open source pour OS X, Linux, et Windows développé par GitHub. [7]

### **3.2.3. PhpMyAdmin**

phpMyAdmin est un outil logiciel gratuit écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin supporte un large éventail d'opérations sur MySQL et MariaDB.[8]

### **3.2.4. Modelio**

Modelio est un environnement de modélisation open source UML, il offre une large gamme de fonctionnalités basées sur des standards pour les développeurs de logiciels, les analystes, les concepteurs, les architectes d'entreprise et les architectes système. [9]

### **3.2.5. Wamp**

WampServer est une plate-forme de développement web sous windows pour des applications web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL.[10]

### **3.2.6. Workbench**

MySQL Workbench est un outil visuel unifié pour les architectes de bases de données, MySQL Workbench fournit des outils de modélisation de données, de développement SQL et d'administration complète pour la configuration du serveur [11]

### 3.3. L'architecture de l'application

Le framework laravel se base sur une architecture MVC comme nous indique la figure suivante (figure 21). Le contrôleur est responsable de la logique de contrôle de l'application, il sert à gérer les demandes des utilisateurs et à récupérer des données, en tirant parti des modèles, les modèles servent à interagir avec la base de données et récupérer les informations des objets, les vues pour afficher des pages.

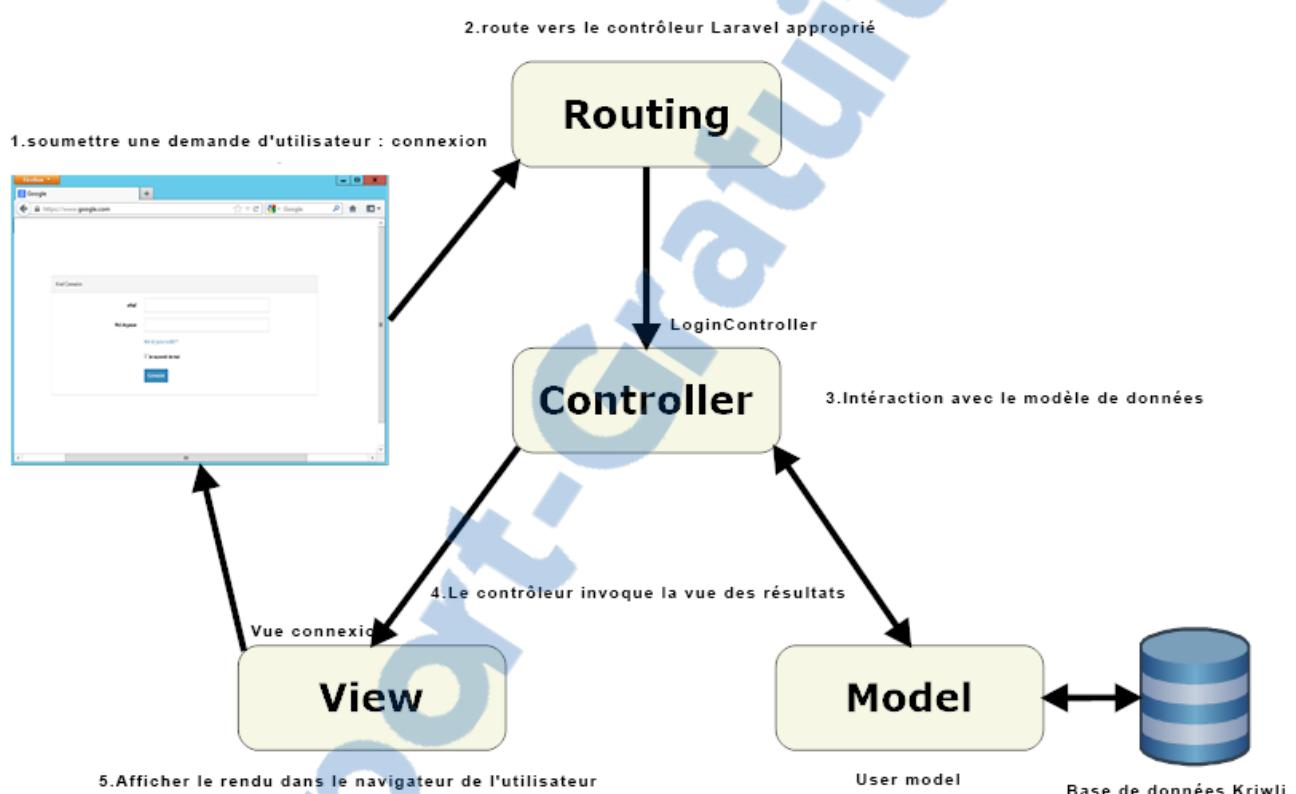
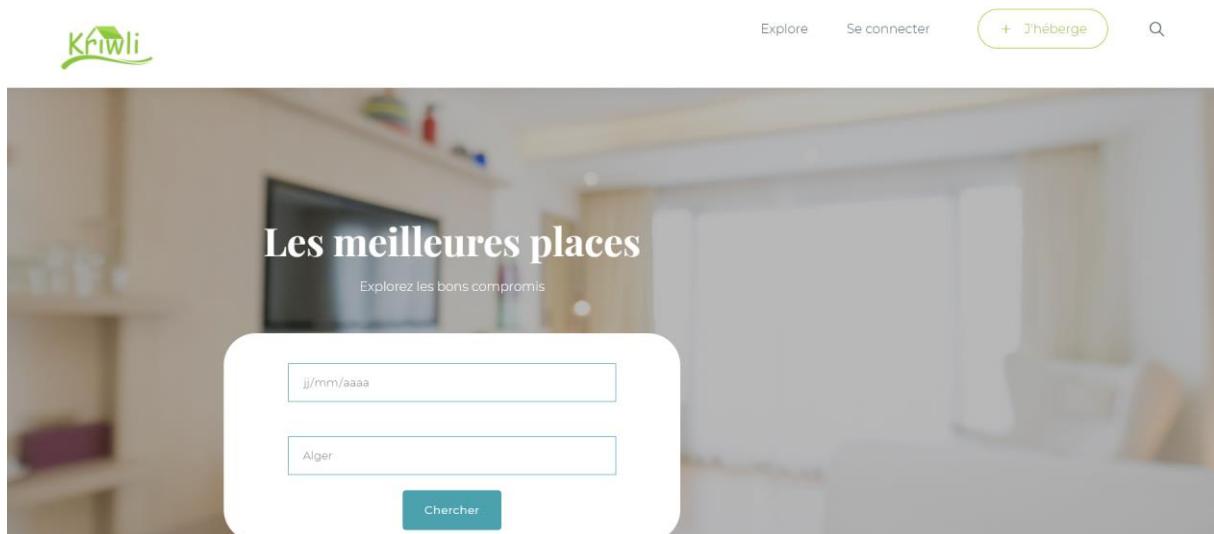


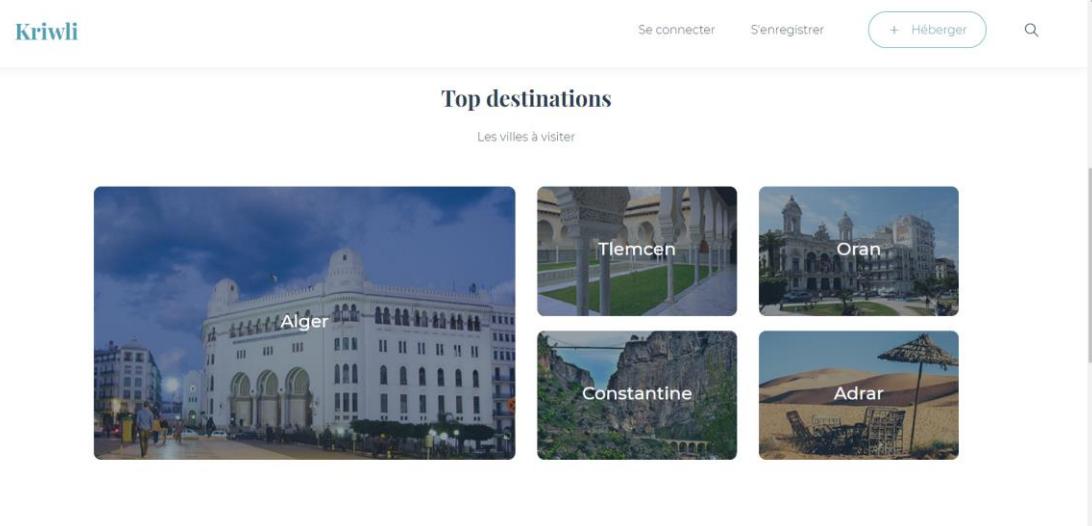
Figure 15 : architecture MVC de l'application Kriwli

### 3.4. Maquettage du système

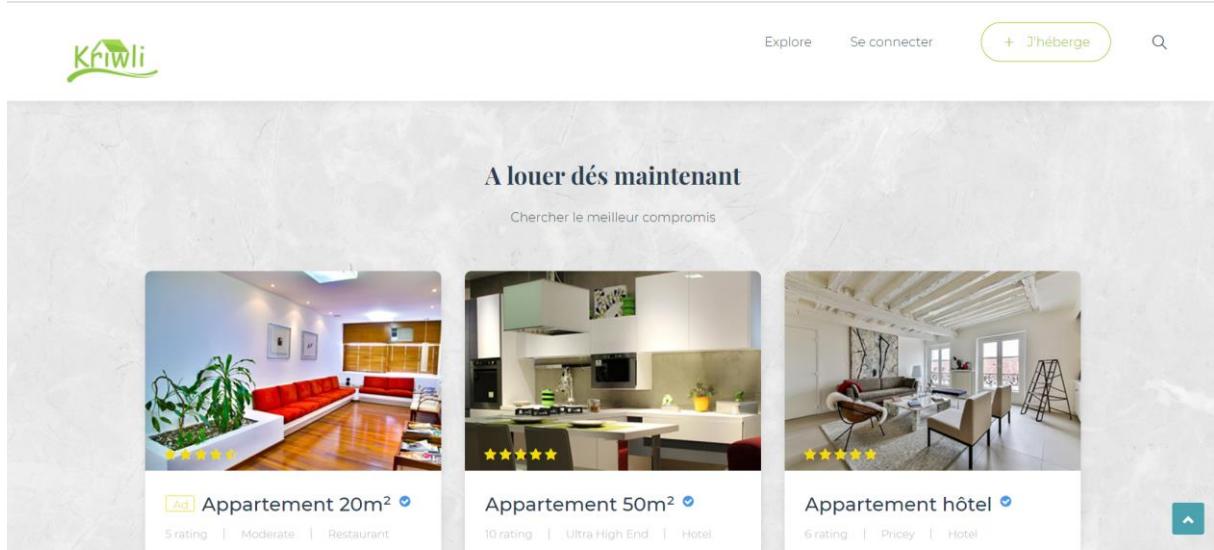
Le maquettage reste une des étapes la plus importantes à définir afin d'avoir un rendu visuel de notre application, nous sommes passés par un maquettage sur papier (Annexe A, Annexe B) avant de pouvoir définir les suivantes (figures 15, 16, 17) Les maquettes suivantes représentent la page d'accueil de notre système qui contient les principales fonctionnalités « rechercher », « réserver », « se connecter ».



**Figure 16 :** Page d'accueil partie 1

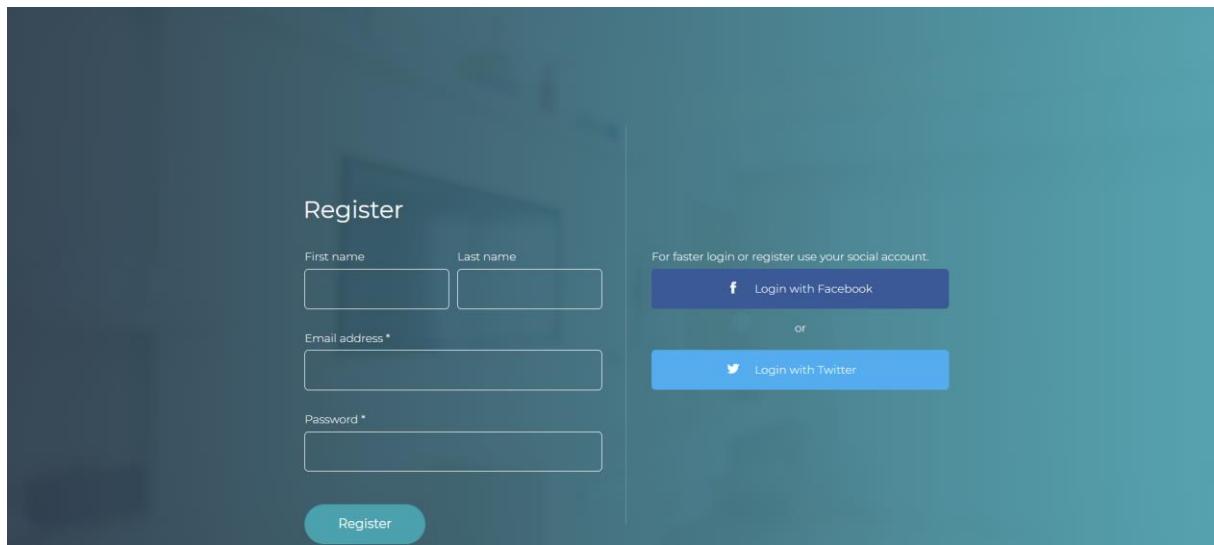


**Figure 17 :** Page d'accueil partie 2



**Figure 18 :** Page d'accueil partie 3

La maquette suivante (figures 18) représente la page de création d'un nouveau compte, c'est-à-dire l'enregistrement des informations des utilisateurs.



**Figure 19 :** Page d'enregistrement

La figure 19 représente la page d'ajout d'un nouvel hébergement, qui doit avant tout passer par la case de connexion.

If you don't have an account you can create one below by entering your email address/username.  
Your account details will be confirmed via email.

[Sign in](#)

---

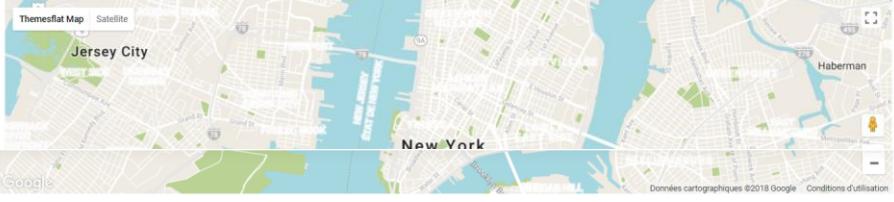
Your email

Listing Name

---

Location (optional)

"New York"



---

Phone

Website

Category

Choose Your Bussines Category

Add Business Hours

Saturday	09:00	22:00	<a href="#">+</a>
Monday	09:00	-	X
Tuesday	09:00	-	X
Wednesday	09:00	-	X
Thursday	09:00	-	X
Friday	09:00	-	X

---

Price Range

Pricey

Price From

Price To

▲

Description

Detail description about your listing

Tags or keywords (Comma separated)

Enter tag or keywords coma seprated ...

---

Your Facebook URL

Your Instagram URL

Your Twitter URL

Your Google Plus URL

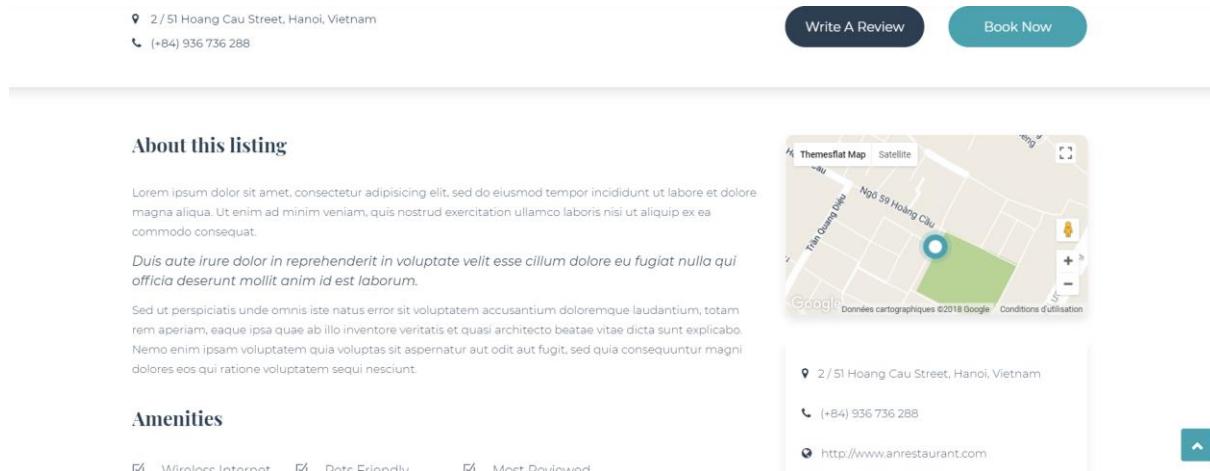
Your Youtube URL

Your linkedin URL

▲

**Figure 20 : Page d'ajout d'hébergement**

Cette figure représente la page de réservation d'un appartement, ainsi que toutes les descriptions qui mettent en avant cet hébergement.



**Figure 21 : Page de réservation**

### 3.5. L'implémentation de « Kriwli » :

#### 3.5.1. Installation et organisation

Avant de commencer l'installation du framework laravel, nous devons installer « composer » qui est un gestionnaire de dépendances libre écrit en php et permet d'installer les bibliothèques, la commande est « composer global require laravel/installer »

Nous nous plaçons dans le répertoire « www » de wamp afin de pouvoir travailler en local.

**Figure 22 :** dossier d'installation de laravel

Pour installer le projet laravel avec des composants de base la commande suivante composer « composer create-project laravel/laravel kriwli --prefer-dist»

**Figure 23 :** création du framework laravel

Le dossier « Kriwli » se créer par la suite contenant les différents dossiers utiles à l'utilisation du framework laravel.

Pour lancer localement notre application laravel, nous utilisons la commande « php artisan serve », nous aurons accès sur notre navigateur web à une page d'accueil laravel qui ressemblera à la suivante.

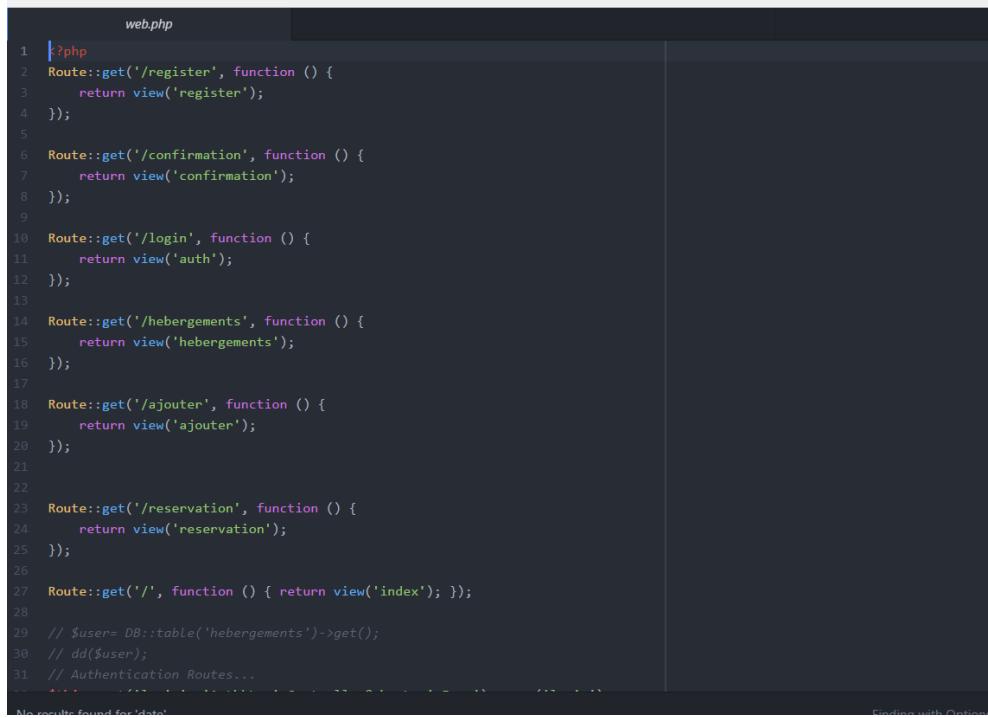


**Figure 24 :** figure représentant la page d'accueil de laravel

### 3.5.2. Les routes

Quand la requête atteint le fichier public/index.php l'application Laravel est créée et configurée et l'environnement est détecté.

Ensuite le fichier web.php est chargé, c'est avec ce fichier que la requête va être analysée et dirigée. Regardons ce qu'on contient notre fichier de routage sur la figure 24.



```
web.php
1  php
2  Route::get('/register', function () {
3      return view('register');
4  });
5
6  Route::get('/confirmation', function () {
7      return view('confirmation');
8  });
9
10 Route::get('/login', function () {
11     return view('auth');
12 });
13
14 Route::get('/hebergements', function () {
15     return view('hebergements');
16 });
17
18 Route::get('/ajouter', function () {
19     return view('ajouter');
20 });
21
22
23 Route::get('/reservation', function () {
24     return view('reservation');
25 });
26
27 Route::get('/', function () { return view('index'); });
28
29 // $user= DB::table('hebergements')-&gt;get();
30 // dd($user);
31 // Authentication Routes...
32
33</pre
```

Figure 25 : image représentant le fichier de route web.php

### 3.5.1. Les migrations

Les migrations représentent la partie base de données de notre application, elle permettent de mettre à jour le schéma des bases de données.

Pour cela nous devons configurer en premier lieu sur le fichier « .env » les champs nécessaires à la connexion à notre base de données créée auparavant.

La création de la migration se fait à l'aide de la commande « php artisan make:migration create\_users\_table ». Nous définissons l'ensemble des champs utilisés dans notre table « utilisateur »

```

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->string('password');
21             $table->rememberToken();
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33
34     }
35 }

```

No results found for 'date'

date

Replace in current buffer

no results

Find

Find All

Replace

Replace All

LF

UTF-8

PHP

0 files

Figure 26 : image représentant la migration de la table utilisateur

### 3.5.2. Classe Model

Cette classe nous permet de donner une représentation sous forme d'objets manipulables à tous les éléments de la base de données et qui permet de simplifier grandement les opérations en utilisant eloquent qui représente un ORM.

La création du model se fait à l'aide de la commande « php artisan make:model Reservation »

```

1 <?php
2 namespace App;
3
4 use Illuminate\Database\Eloquent\Model;
5 use Carbon\Carbon;
6 use Illuminate\Database\Eloquent\SoftDeletes;
7
8 /**
9  * Class Reservation
10 *
11 * @package App
12 * @property integer $idreservation
13 * @property string $datedebut
14 * @property string $datefin
15 * @property string $idinscrit
16 * @property string $typehebergement
17 */
18 class Reservation extends Model
19 {
20     use SoftDeletes;
21
22     protected $fillable = ['idreservation', 'datedebut', 'datefin', 'idinscrit_id', 'typehebergement_id'];
23     protected $hidden = [];
24
25
26     /**
27      * Set attribute to money format
28      * @param $input
29      */
30     public function setIdreservationAttribute($input)
31     {
32
33     }
34 }

```

26 results found for 'date'

date

Replace in current buffer

26 found

Find

Find All

Replace

Replace All

LF

UTF-8

PHP

0 files

Figure 27 : image représentant la classe modèle utilisateur

### 3.5.3. Contrôleur

La tâche d'un contrôleur est de réceptionner une requête et de définir la réponse appropriée, à l'aide de la commande « `php artisan make:controller HebergementController` » nous créons le controller hébergement.

Nous avons les méthodes qui sont chargées de générer la vue, parmi elles la méthode index qui renvoie vers la vue utilisateur.

**Figure 28 :** image représentant le contrôleur utilisateur

### 3.5.4. Les vues :

Le processus de visualisation de pages se fait à travers des vues ainsi que les routes

**Figure 29 :** vue connexion d'utilisateur

## Conclusion

A ce chapitre, notre projet d'études atteint sa fin. Tout au long de ce chapitre, nous avons abordé notre environnement de travail. Par la suite, nous avons expliqué notre architecture d'application afin de présenter finalement les différentes principales parties d'implémentation de notre application réalisée.

# Conclusion générale

Ce projet de fin d'études avait pour ambition d'établir un système de réservation en ligne innovant, en se demandant si le besoin réel existait ou pas et faire face à l'absence des plateformes de réservation d'hébergement en ligne adéquates pour la population locale.

Parmi les personnes sondées, plus de la moitié étaient d'accord avec l'affirmation suivante « seriez-vous prêts à réserver en ligne », tandis qu'une minorité était encore réticente.

Le rapport mentionne toutes les étapes traversées pour arriver au résultat attendu. Il a fallu dans un premier temps recenser les différents besoins existants, nous avons pu aussi donner un contexte général à notre projet et identifier les différentes exigences du futur système.

Nous avons établi ensuite une étude des systèmes existants qui nous permettent de connaître les fonctionnalités primordiales. Par la suite nous avons entamé la phase d'analyse et de conception du système.

Pour conclure le dernier chapitre, notre projet d'étude atteint sa fin. Tout au long de ce chapitre, nous avons abordé notre environnement de travail. Par la suite, nous avons expliqué notre architecture d'application afin de présenter finalement les différentes principales parties d'implémentation de notre application réalisée.

Ce travail nous a été très formateur, puisqu'il a permis de découvrir une nouvelle technologie innovante, et nous a permis également de se confronter à plusieurs contraintes à la fois : contraintes de temps, contraintes d'expérience et de technologie. En outre, ce projet nous a permis d'approfondir nos connaissances dans les bonnes pratiques de l'ingénierie génie logiciel.

## Perspectives

Notre réalisation est encore d'actualité et ne s'arrête pas à ce niveau. En effet plusieurs perspectives s'offrent à ce projet :

Pour rendre l'application plus expansive en termes de base de données, nous proposons d'utiliser MongoDB.

Nous ajouterons aussi un comparateur de prix ainsi que les méthodes de paiement en ligne, e-commerce et m-commerce afin de rendre notre application web plus digital.

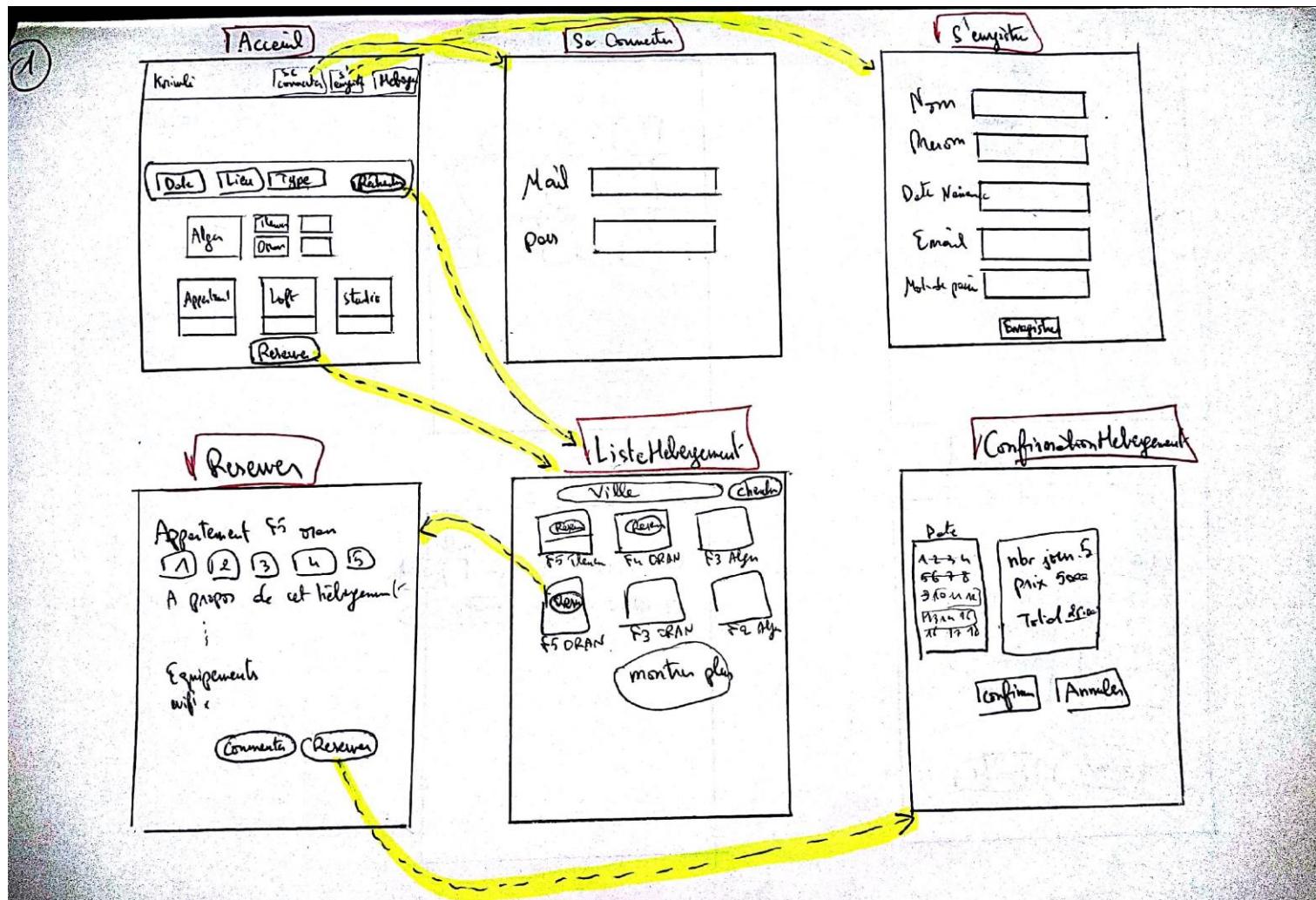
Finalement, vu l'accomplissement de projet, nous souhaitons très fortement qu'il soit le fruit du progrès, de l'évolution et qu'il reste à la hauteur des exigences de la société.

## Références Bibliographiques :

- [1] DEFINITIONS MARKETING, Définition : E-réputation, sur [www.definitionsmarketing.com/definition/e-reputation/](http://www.definitionsmarketing.com/definition/e-reputation/) consulté le 28/05/2018 à 14 h 38
- [2] MEMOIRE ONLINE, Définition des besoins fonctionnels et des besoins non fonctionnels, sur [www.memoireonline.com](http://www.memoireonline.com), consulté le 15/05/2018 à 12 h 14
- [3] Département d'informatique, Pavillon Adrien-Pouliot, Université Laval, Evaluation de la Qualité des Applications Web : Etat de l'Art. [document électronique]. Québec,2001, <https://pdfs.semanticscholar.org/5592/7427ef058c5c792bd69c7fe18a6bb76431a1.pdf> consulté le 15/05/2018 à 12 h 00
- [4] EYROLLES, Découvrez le framework PHP LARAVEL, sur <http://www.eyrolles.com/Chapitres/9782212143980/9782212143980.pdf>, consulté le 10/05/2018 à 11 h 06.
- [5] BOOTSTRAP, sur <https://getbootstrap.com/>, consulté le 10/05/2018 à 11 h 48.
- [6] Journal du net, Google Trends (ex Google Insight) : définition, sur <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203505-google-trends-ex-google-insight-definition/>, consulté le 10/05/2018 à 12 h 22
- [7] Tice éducation, ATOM : un éditeur de texte totalement personnalisable, sur <https://www.tice-education.fr/index.php/tous-les-articles-er-ressources/articles-informatiques/114-informatique/codage/1005-atom-un-editeur-de-texte-totalement-personnalisable>, consulté le 10/05/2018 à 12 h 32
- [8] PhpMyAdmin, Bringing MySQL to the web, sur <https://www.phpmyadmin.net/>, consulté le 10/05/2018 à 13 h 01 .
- [9] Modelio, sur <https://www.modelio.org/>, consulté le 10/05/2018 à 13 h 27.
- [10] Wamp, sur <http://www.wampserver.com/>, consulté le 10/05/2018 à 13 h 29.
- [11] Mysql, sur <https://www.mysql.com/fr/products/workbench/>, consulté le 10/05/2018 à 13 h 30.
- [12] Documentation Laravel, sur <https://laravel.com/docs/5.6>, consulté le 10/02/2018 à 13 h 40.
- [13] O.Guibert, La modélisation en informatique, sur <https://www.labri.fr/perso/guibert/DocumentsEnseignement>, consulté le 11/02/2018 à 14h00
- [14] Découvrez le framework PHP Laravel, <https://openclassrooms.com/courses/dcouvrez-le-framework-php-laravel-1>, consulté le 15/02/2018 à 17 h 40.

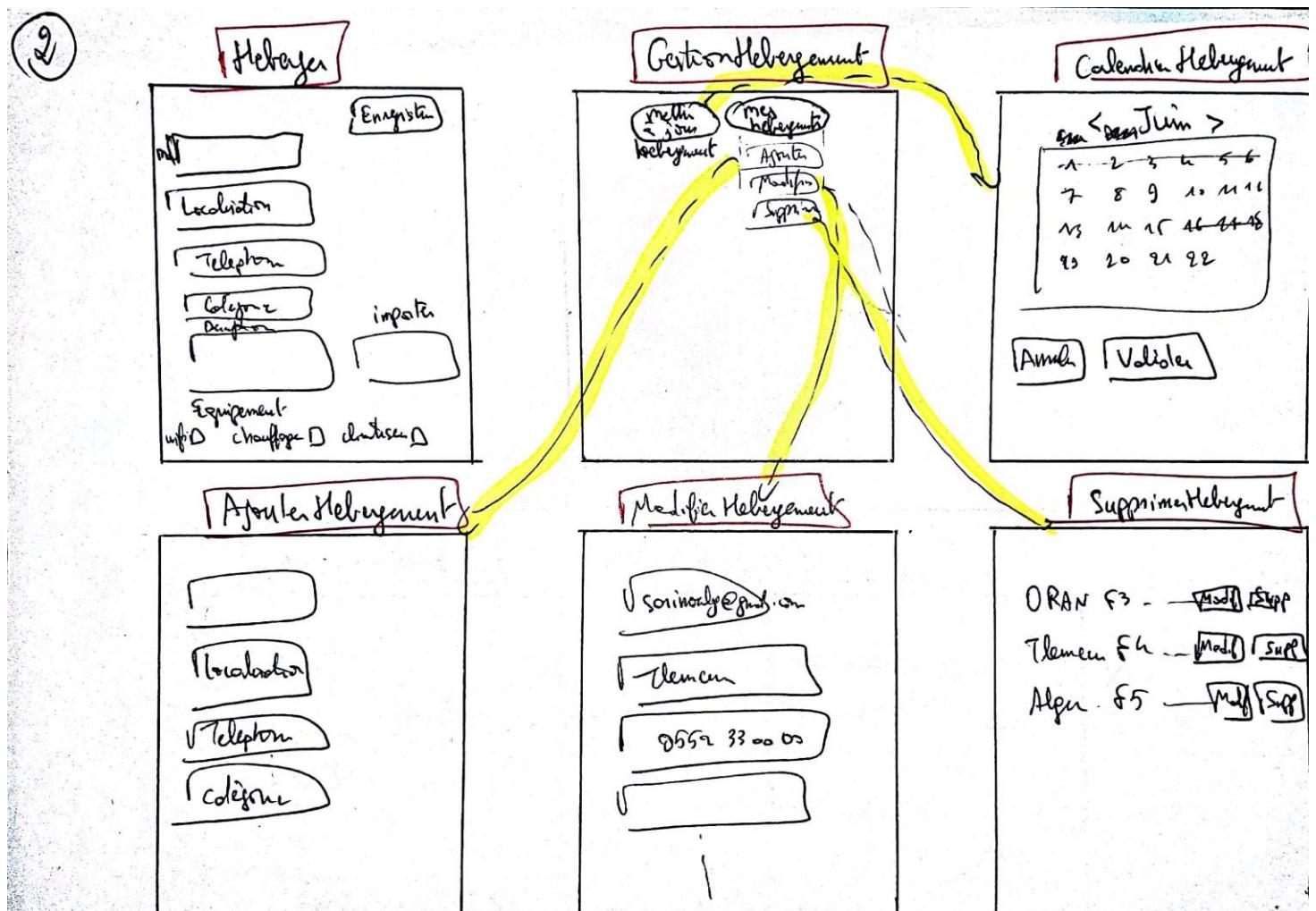
## Annexe A : maquettes de l'application kriwli à l'étape 1 dessinées

Maquettes qui représentent les étapes de connexion et de réservations fait un futur loueur.



## Annexe B : maquettes de l'application kriwli à l'étape 2 dessinées

Annexe représentant la maquette de gestion d'hébergement réalisé par un hébergeur.



# Résumé

Le marché mondial se situe dans une transition digitale qui pousse les fournisseurs à établir des stratégies en ligne ultra concurrentiel, l'Algérie est en voie de développement en termes de numérique, avec l'adoption de la loi e-commerce et de l'apparition de la vente en ligne, les réservations se font en ligne et deviennent une routine pour le citoyen algérien, la réservation d'hébergement se fait à travers certaines plateformes en ligne. Dans notre projet de fin d'étude nous avons conçu et réalisé le système baptisé Kriwli : un système de réservation d'hébergement en ligne.

## Mots clés

**Réservation, hébergement, Algérie, plateformes en ligne**

## ملخص

إن السوق العالمية تمر بمرحلة انتقالية رقمية تدفع الموردين إلى وضع إستراتيجية تنافسية عبر الإنترنت ، وتقوم الجزائر بتطوير القطاع الرقمي ، مع اعتماد قانون التجارة الإلكترونية وظهور المبيعات عبر الإنترنت ، ويتم إجراء الحجوزات عبر الإنترنت وتصبح روتيناً بالنسبة للمواطن الجزائري ، يتم حجز الإقامة من خلال بعض المنصات عبر الإنترنت

## الكلمات الدالة

الحجز ، الاستضافة ، الجزائر ، المنصات الإلكترونية

# **Abstract**

The global market is in a digital transition that pushes suppliers to establish a competitive online strategy, Algeria is developing the digital sector, with the adoption of the e-commerce law and the emergence of the online sales, bookings are made online and become a routine for the Algerian citizen, booking accommodation is through some online platforms.

In our end-of-study project we designed and built the system called Kriwli: an online hosting reservation system.

# **Keywords**

**Booking, hosting, Algeria, online platforms**