

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MEMOIRE DE MASTER PROFESSIONNEL

En
Informatique
Option
Génie Logiciel

Thème

Conception et réalisation d'une application
Web de location de vacances, cabanes,
maison de plages et logements.

Présenter par : Mr HAMMOUM Yanis.

Mr KHOULALENE Samy.

soutenu le 12 octobre 2021 devant le jury composé de :

President	Dr M. MOKTEFI	Maître de conf.	U. A/Mira Bejaia.
Rapporteur	Pr K. AMROUN	Professeur	U. A/Mira Bejaia.
Co-rapporteur	Dr N. RAGAB	Doctorant	U. A/Mira Bejaia.
Examineur	Dr K. SELLAMI	Maître de conf.	U. A/Mira Bejaia.

Bejaia, oct 2021.

** Remerciements **

Nous tenons tout d'abord à remercier **DIEU** le tout puissant qui nous a donné la force de réaliser ce projet.

Nous ne pouvons pas oublier d'adresser notre gratitude à nos chers parents qui, tout au long de ce travail nous ont soutenues avec leurs précieux encouragements et patiences.

Nous remerciant également Notre encadreur Mr AMROUN Kamal et co-encadreur Mr ELSAKAAN Nadim dit RAGAB qui nous ont orienté grâce à leur précieux conseils, critiques constructives et leur orientations qui nous ont été très bénéfique tout au long de ce travail. Nous réservons un vif remerciement à tous ceux qui ont contribué de près ou de loin à l'aboutissement de notre travail.

Aussi, nous remercions également Mr Mocketfi Mohand le président de jury ainsi que Mr SELLAMI Khaled qui ont accepté d'examiner ce modeste travail.

Que tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail trouvent l'expression de nos remerciements les plus chaleureux.

※ *Dédicaces* ※

*À mon exemple éternel, mon soutien moral, celui qui s'est toujours sacrifié pour me voir réussir, mon père, que **ALLAH** t'accueille dans son vaste paradis.*

*A la lumière de mes jours, source de mes efforts, ma mère, que **ALLAH** te donne une bonne santé et une longue vie.*

À mes chères sœurs, je dédie ce travail dont le grand plaisir leur revient avant tout pour leurs conseils et leurs encouragements.

À mon partenaire Samy et toute la famille KHOULALENE.

Aux personnes qui m'ont toujours aidé et encouragé, qui ont toujours été à mes côtés, et qui m'ont accompagné durant mes études supérieures.

*Mr. **HAMMOUM Yanis***

※ *Dédicaces* ※

*À mon exemple éternel, mon soutien moral, celui qui s'est toujours sacrifié pour me voir réussir, mon père, que **ALLAH** t'accueille dans son vaste paradis.*

*A la lumière de mes jours, source de mes efforts, ma mère, que **ALLAH** te donne une bonne santé et une longue vie.*

À mes chères sœurs, je dédie ce travail dont le grand plaisir leur revient avant tout pour leurs conseils et leurs encouragements.

À mon partenaire Yanis et toute la famille HAMMOUM.

Aux personnes qui m'ont toujours aidé et encouragé, qui ont toujours été à mes côtés, et qui m'ont accompagné durant mes études supérieures.

*Mr. **KHOULALENE** **Samy***

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	v
Notations et symboles	vi
Introduction générale	1
1 Généralités et définitions	3
1.1 Définition de <i>E</i> -commerce	3
1.2 Application de réservation en ligne	4
1.2.1 Définition d'une application de réservation	4
1.2.2 Pourquoi une application de réservation ?	4
1.2.3 Description de l'application de réservation	5
1.3 Etude de l'existant	5
1.4 Présentation du projet	7
1.5 Problématique	7
1.6 Objectifs du projet	8
1.7 Conclusion	8
2 Conception et modélisation	9
2.1 Introduction	9
2.2 Langage de modélisation	9

2.2.1	MERISE	9
2.2.2	UML	10
2.2.3	Choix du langage de modélisation	10
2.2.4	Diagrammes d'UML	10
2.3	Méthodes de développement logiciel	13
2.3.1	Méthode Agile	13
2.3.2	Processus UP	14
2.3.3	Méthode XP	14
2.4	Modélisation de notre système	14
2.4.1	Spécification des besoins	15
2.4.2	Analyse des besoins	24
2.5	Conclusion	44
3	Implémentation	45
3.1	Introduction	45
3.2	Outils de développement	45
3.2.1	Environnement logiciel	45
3.2.2	Environnement de programmation	52
3.3	Présentation de l'application	60
3.3.1	Interface d'accueil	61
3.3.2	Interface correspondant à la création d'un compte	62
3.3.3	Interface d'authentification	63
3.3.4	Interface correspondant à l'ajout d'un hébergement	64
3.3.5	Interface correspondant à la page d'hébergement	65
3.3.6	Interface correspondant aux détails sur un hébergement	66
3.3.7	Interface correspondant à la réservation d'un hébergement	67
3.4	Conclusion	68
	Conclusion et perspectives	69
	Bibliographie	70

Table des figures

2.1	Diagramme d'UML.	11
2.2	Diagramme de cas d'utilisation.	18
2.3	Diagramme de séquence «Créer compte»	25
2.4	Diagramme de séquence «Recherche hébergement»	27
2.5	Diagramme de séquence «Explorer les environnements»	29
2.6	Diagramme de séquence «S'authentifier»	31
2.7	Diagramme de séquence «Réserver hébergement»	33
2.8	Diagramme de séquence «Payer»	35
2.9	Diagramme de séquence «Publier hébergement»	37
2.10	Diagramme de séquence «Gérer compte utilisateur»	39
2.11	Diagramme de classes	41
3.1	Logo visual studio	46
3.2	Logo npm	47
3.3	Logo Git Bash	48
3.4	Logo Mailchimp	48
3.5	Logo Postman	49
3.6	Logo Hyper	50
3.7	Logo Robo 3T	50
3.8	Logo StarUML	51
3.9	Logo Heroku	52
3.10	Logo HTML5	53
3.11	Logo CSS3	54
3.12	Logo JavaScript	55

3.13 Logo Bootstrap	55
3.14 Logo jQuery	56
3.15 Logo Node.js	57
3.16 Logo Express.js	57
3.17 Logo MongoDB	58
3.18 Logo Passport	59
3.19 Logo EJS	59
3.20 Interface d'accueil	61
3.21 Interface pour créer un compte	62
3.22 Interface d'authentification	63
3.23 Interface pour ajouter un hébergement	64
3.24 Interface pour les hébergements	65
3.25 Interface correspondant aux détails sur un hébergement	66
3.26 Interface pour réserver un hébergement	67

Liste des tableaux

2.1	Besoin fonctionnel de chaque acteur	16
2.2	Cas d'utilisation «Créer compte»	19
2.3	Cas d'utilisation «Recherche hébergement»	20
2.4	Cas d'utilisation «Explorer les environnements»	20
2.5	Cas d'utilisation «S'authentifier»	21
2.6	Cas d'utilisation «Résérvier hébergement»	22
2.7	Cas d'utilisation «Payer»	23
2.8	Cas d'utilisation «Publier hébergement»	23
2.9	Cas d'utilisation «Gérer compte utilisateur»	24
2.10	Dictionnaire de données de l'application	43

Notations et symboles

API	A pplication P rogramming I nterface.
CSS	C ascading S tyle S heets.
EJS	E mbded J ava S cript.
Git Bash	B ourne A gain S hell.
HTML	H yper T ext M arkup L anguage.
IEEE	I nstitute of E lectrical and E lectronics E ngineers.
JS	J ava S cript.
JSON	J ava S cript O bjet N otation.
MAC OS	M ACintosh O perating S ystem.
MERISE	M éthode E tude R éalisation I nformatique S ystèmes E ntreprise.
NPM	N ode P ackage M anager.
OMG	O bjet M anagement G roup.
PHP	H ypertext P reprocessor.
SMS	S hort M essage S ervice.
SSPL	S erver S ide P ublic L icense.
UML	U nified M odeling L anguage.
UP	U nified P rocess.
W3C	W orld W ide W eb C onsortium.
XML	e Xtensible M arkup L anguage.
XP	E Xtreme P rogramming.

Introduction générale

Depuis son apparition, l'informatique est devenue accessible et surtout indispensable pour toutes les entreprises avides de hautes technologies pour assurer la bonne gestion des services et par la même occasion sa survie dans son domaine d'activité. Les principaux objectifs de l'informatisation sont avant tout le gain de temps ainsi que la sauvegarde permanente des données.

Ces deux dernières années, la pandémie du coronavirus s'est répandue à travers le monde entier ce qui a affecté la vie des gens en ce qui concerne non seulement leur vie quotidienne mais également leur vacances. De plus, les voyages internationaux sont difficiles à faire ce qui a orienté les algériens vers des voyages nationaux en visitant des villes algériennes. Cependant, il n'existe pas beaucoup de services de réservation en ligne en Algérie qui permettra aux utilisateurs de réserver en avance et d'éviter d'avoir de mauvaises surprises. De ce fait, une augmentation du besoin digital est nécessaire.

L'objectif de ce mémoire consiste à traiter les défis susmentionnés : l'inexistence de service de réservation d'hébergement à Béjaia et l'orientation supervisée des voyageurs en proposant une application web de réservation d'hébergement pour la ville de Béjaia capables d'améliorer la location de maisons privées partout à Béjaia mais aussi permettre aux propriétaires des maisons de vacances de lister leur espaces de vacances

Ce mémoire est réparti en trois principaux chapitres décrivant les différentes étapes à établir pour la réalisation de notre projet comme décrits ci-dessous :

Le premier chapitre « Généralités et définitions » présente quelques concepts de bases en relation avec notre projet, l'étude de l'existant ainsi que la spécification des besoins de l'application à développer.

Le second chapitre « Conception et modélisation » vise à définir une feuille de route du projet contenant l'ensemble des différents diagrammes (cas d'utilisation, séquence, etc.) participant à la modélisation de notre application.

Le troisième chapitre « Implémentation » concerne la réalisation du projet en présentant les différents outils utilisés ainsi que quelques interfaces correspondant à notre application.

Enfin, nous concluons ce travail et présentons quelques perspectives.

Généralités et définitions

Bejaia autrefois appeler 'Bougie' est une région algérienne située au bord de la mer méditerranée, elle dispose d'une importante position géographique ce qui la rend attractives aux gens qui aiment y passer leur vacances. Cependant, avec la pandémie du coronavirus le besoin d'une réservation en ligne est nécessaire.

Pour cela une méthode de travail est nécessaire pour répondre aux exigences des voyageurs, gérer l'ensemble des activités et remédier aux incidents rencontrés. Afin de réaliser cela, un mécanisme dans le cadre d'une structure optimale et performante est indispensable.

Dans ce chapitre, nous présentons une vue sur les différentes applications existantes en se focalisant sur les anomalies de ces dernières pour ainsi présenter les objectifs de notre travail.

1.1 Définition de *E*-commerce

La technologie de l'information a transformé la manière dont les gens travaillent. le commerce électronique a déclenché une autre révolution, qui change la manière dont les entreprises achètent et vendent des produits et des services.

Le commerce électronique fait référence à l'échange sans papier d'informations commerciales à l'aide de l'échange de données électroniques, du courrier électronique et des billets électroniques. Le commerce électronique automatise non seulement les processus manuels et les transactions papier, mais aide également les organisations

à passer à un environnement entièrement électronique et à modifier leur mode de fonctionnement.[1]

1.2 Application de réservation en ligne

1.2.1 Définition d'une application de réservation

Il s'agit d'une solution logicielle permettant aux voyageurs potentiels de réserver et de payer leur réservations en utilisant un site Web et d'autres canaux, le tout en un seul endroit. En outre, un système de réservation en ligne s'appuie sur une technologie intelligente afin d'éliminer les risques associés à la saisie manuelle et à l'erreur humaine comme par exemple les doubles réservations.[2]

1.2.2 Pourquoi une application de réservation ?

L'utilisation d'une application de réservation permet de [3] :

- ***garder les entreprises 24/7 ouvertes*** : en effet, cela permet aux voyageurs d'en savoir plus sur votre entreprise sans devoir attendre les heures d'ouvertures pour réserver leur places.
- ***améliorer la gestion de ressources*** : en automatisant les opérations commerciales grâce à l'élimination du besoin de saisie manuelle des données, la gestion des réservations, la surveillance de l'inventaire, la création des rapports et le traitement des paiements.
- ***renforcer la sécurité des données*** : en utilisant le cloud pour la sauvgarde des données aux lieu des ordinateurs de bureau susceptible à des pannes.
- ***créer une meilleure expérience client*** : la technologie de réservation accélère tout en quelques étapes seulement, les clients peuvent réserver, payer et signer les documents requis tout en obtenant une confirmation dans leur boîte de réception immédiatement.
- ***rester organiser*** : tout est au même endroit. Par une simple connexion, une

ventilation des réservations quotidiennes, quels clients doivent encore payer et signer, quel inventaire est disponible, etc.

1.2.3 Description de l'application de réservation

Les applications de réservation offrent aux consommateurs des services de recherche et de comparaison performants et constituent des vitrines très pratiques afin de réaliser un choix optimal. Parmi leur caractéristiques on peut trouver :

- Basé sur le cloud.
- Système d'inventaire.
- Réservation en temps réel.
- Interface conviviale.
- Soutien à la clientèle.
- Automatisation des SMS et des e-mails.
- Intégration avec des logiciels tiers (par exemple : un systèmes de point de vente, portails d'évaluation, etc.).
- Gestion des canaux de distribution.
- Gestion des ressources.

1.3 Etude de l'existant

Dans cette partie, nous allons présenté quelques sites pertinent qui permettent la réservation en ligne. on peut distingué :

i) Les sites de références mondiale

- **Booking.com** est un site néerlandais qui propose des hébergements dans différents types de propriétés. Il est mis en 43 autres langues. Il a commencé comme une petite start-up à Enschede en 1996, financée par l'université de Twente. Le site est désormais basé à Amsterdam depuis 2005. En 2014, la société est présente dans 59 pays. Elle propose de l'hébergement dans plus de

540 000 propriétés dans 202 pays. Ce site ne facture pas de frais de réservation à ses clients.[4]

- **Airbnb** anciennement appelé Airbed and breakfast c'est à dire « matelas gonflable et petit déjeuner » en français) est un service de plateforme communautaire payant de location de logements de particuliers, d'entreprises hôtelières (chambres d'hôte et d'hôtel), et d'investisseurs en immobiliers locatifs para-hôtelières fondée en 2008 par les Américains Brian Chesky, Joe Gebbia et Nathan Blecharczyk. Le site Internet contient en 2015 plus de 1,5 million d'annonces dans 34 000 villes et 192 pays et établie à San Francisco, la société est détenue et exploitée par Airbnb Inc.[5]

Les différentes fonctionnalités que nous pouvons retrouver :

- 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.

i) Les sites de références algérien

- **lkeria** est un site de réservation en ligne d'hébergement qui possède beaucoup de problèmes parmi eux :
 - Réservation par téléphone qui consomme beaucoup de temps.
 - Une faible ergonomie.
 - Absence d'option de paiement en ligne.
- **Ouedkniss** est un site web d'annonces parmi les sites les plus populaires en Algérie qui possède plusieurs type de catégories d'annonces, Parmi eux : la location d'hébergement. Néanmoins, il a plusieurs inconvénients, on peut citer

le manque de détails sur les annonces comme les prix et les images et l'absence de Paiements en ligne avec une faible ergonomie. De plus, il ne permet pas la finalisation d'une commande.

1.4 Présentation du projet

Béjaia est un lieu magnifique que ce soit pour les algériens qui choisissent d'y passer leurs vacances ou pour les étrangers qui viennent séjourner. De plus, avec la pandémie du coronavirus le besoin digital ne cesse d'augmenter dans le monde entier. Par conséquent, Réaliser une application en ligne à double face basée sur la communauté qui facilite le processus de réservation d'espaces de vacances pour les voyageurs à Béjaia est une tâche primordiale. D'une part, parce que cela permet aux propriétaires des maisons de vacances de lister leur espaces de vacances mais aussi de gagner de l'argent grâce à cette location. D'autre part, il offre aux voyageurs un accès facile à la location de maisons privées partout à Béjaia mais aussi les profils personnels ainsi qu'un système de notation et d'évaluation fournissent des informations sur l'hôte et de manière réciproque, les hôtes peuvent choisir eux-mêmes à qui louer leur espaces.

1.5 Problématique

Les réservations des maisons de vacances se font par des recommandations de voyageurs ayant déjà séjournés dans des lieux similaires. Le marché algérien des réservations d'hébergement reste difficile à déterminer. à l'heure actuelle, les utilisateurs se contentent généralement des réseaux sociaux pour réserver leurs hébergement. Cependant, il n'existe pas de mécanisme fiable pour assurer une bonne réservation d'ébergement propre à Béjaia. Il est également difficile pour un utilisateur de négocier sur les détails concernant la date et les prix des réservations.

1.6 Objectifs du projet

L'idée générale de ce projet consiste à réaliser une solution informatique qui doit assurer une bonne gestion de réservation. En d'autres termes, concevoir une application de réservation d'hébergement en ligne qui pourra de manière concrète permettre à un utilisateur d'exprimer une demande de réservation en ligne qui sera :

- facile à manipuler.
- protège les informations personnelles des utilisateurs.
- réduire le nombre d'appels téléphoniques reçus
- bénéficie l'économie algérienne.
- développer le commerce électronique en Algérie.
- permet aux propriétaires de lister leur espace de vacances et de gagner de l'argent.
- faire une réservation complète sans aucun problème.
- aider le voyageur à faire le choix idéal pour son logement.

1.7 Conclusion

Au cours de ce chapitre, nous avons réalisé une étude sur les différentes applications d'hébergements existantes en lignes. Cela nous a permis d'identifier les lacunes et faiblesses dont souffrent quelques-unes d'elles pour ainsi, établir un bilan sur ses besoins vis-à-vis des problèmes actuels en termes de gestion de réservation en ligne.

Dans ce qui suit, nous allons concevoir une application de réservation en ligne pour résoudre les problèmes précités, à travers divers outils et méthodes.

Conception et modélisation

2.1 Introduction

Nous allons établir la conception de notre application, une étape très importante pour la réussite d'un projet informatique. La description des éléments du système, représentation des besoins des utilisateurs afin d'avoir une bonne réflexion avant de passer à la réalisation s'avère obligatoire. Cette partie du projet se résume à la présentation du langage de modélisation et à la présentation des diagrammes (diagrammes de séquence, diagrammes des cas d'utilisation et le diagramme de classes).

Avant de présenter notre conception, nous allons parler sur les différents langages et méthodes de modélisation existantes.

2.2 Langage de modélisation

Il existe deux grandes familles de langages de modélisation d'application informatique, MERISE et UML.

2.2.1 MERISE

MERISE est une méthode d'analyse et de conception des systèmes d'information. Son but est d'arriver à concevoir un système d'information pour les entreprises. Elle est basée sur la séparation des données et des traitements à effectuer en plusieurs

modèles conceptuels et physiques. La méthode MERISE date de 1978-1979, et suite à une consultation nationale lancée en 1977 par le ministère de l'industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information [6].

2.2.2 UML

UML est un langage formel normalisé et un support de communication performant qui permet grâce à sa représentation graphique de concevoir des solutions qui facilitent la comparaison. Son caractère polyvalent et sa souplesse ont en font un langage de modélisation universel. Selon l'organisme Object Management Group (OMG). UML est un langage visuel dédié à la spécification, la construction et la documentation d'un système. OMG est actuellement responsable de documenter et de faire évaluer UML. L'acronyme UML signifie Unified Modeling Language, soit « Langage unifié de modélisation », c'est-à-dire que UML intègre dans un même langage graphique des formalismes et des notations empruntés à des méthodes de développement des systèmes qui ont fait leur preuve dont notamment les méthodes dites orientées objet [7].

2.2.3 Choix du langage de modélisation

Notre choix s'est orienté vers le modèle UML comme méthode de développement, parce qu'avant tout elle nous permet d'obtenir une modélisation de très haut niveau indépendante des langages et des environnements de développement, mais aussi c'est l'une des meilleures méthodes qui sont utilisées actuellement [7].

2.2.4 Diagrammes d'UML

UML dispose d'un ensemble de diagrammes répertoriés selon deux grandes catégories comme décrit dans la figure 2.1[8].

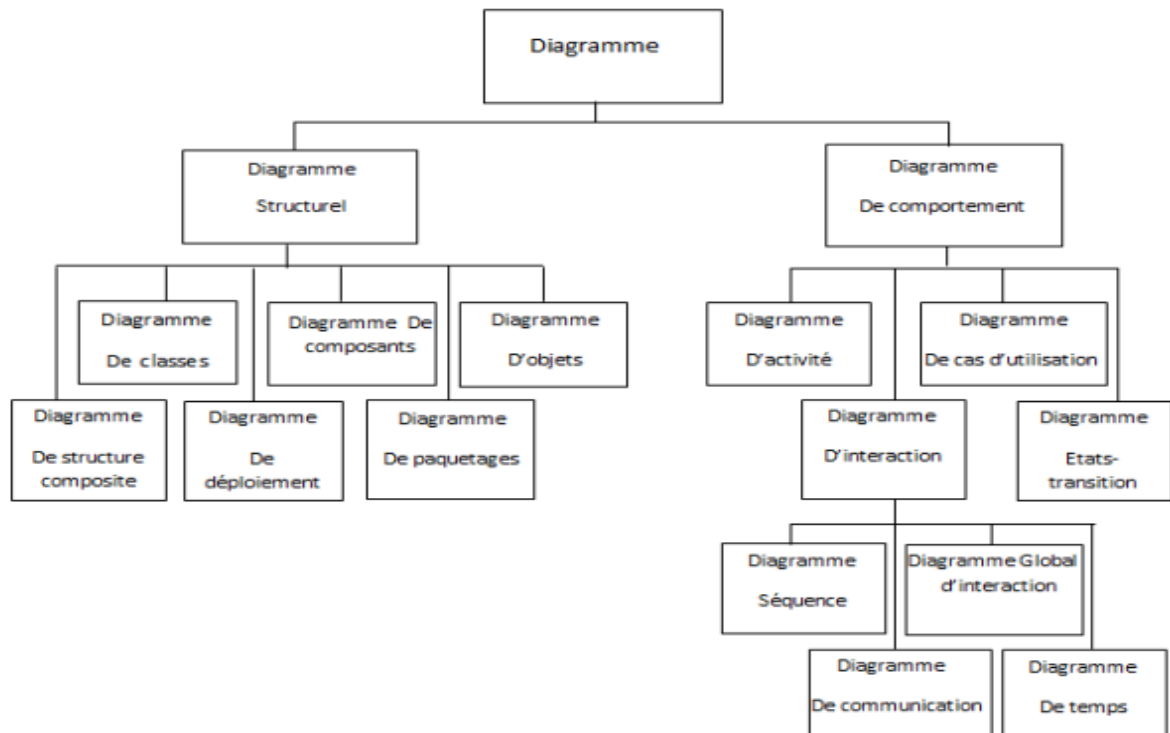


FIGURE 2.1 – Diagramme d'UML.

2.2.4.1 Diagramme de comportement

Il permet de visualiser, spécifier et construire les aspects dynamiques d'un système. Les schémas de comportements sont classés comme suit [9]

2.2.4.1.1 Diagramme de cas d'utilisation il décrit une suite d'actions d'un système. L'objectif est de représenter les besoins des utilisateurs par rapport au système.

2.2.4.1.2 Diagramme d'activité il permet de mettre l'accent sur les traitements. Ainsi que de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

2.2.4.1.3 Diagramme de séquence l'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges.

2.2.4.1.4 Diagramme d'interaction il permet de représenter graphiquement un ensemble d'objets de classes différentes collaborant dans le cadre d'un scénario d'exécution du système. Dans ce genre de diagrammes, les objets communiquent en envoyant des messages qui invoquent des opérations sur les objets récepteurs. Le diagramme d'interaction implique l'utilisation de l'un des diagrammes suivants

i) Diagramme de séquence : l'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges.

ii) Diagramme de communication : il est souvent utilisé pour illustrer un cas d'utilisation ou pour décrire une opération. Le diagramme de communication aide à valider les associations du diagramme de classe.

iii) Diagramme d'interaction globale : Il permet de représenter une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activité.

iv) Diagramme de temps : il permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

2.2.4.2 Diagramme structurel

Il permet de décrire les aspects statiques du système. Il comporte les diagrammes suivants [9]

2.2.4.2.1 Diagramme de structure composite : permet de décrire des collaborations d'instances (de classes, de composants, etc.) constituant des fonctions particulières du système à développer.

2.2.4.2.2 Diagramme de classe : représente les classes intervenant dans le système, chaque classe se décrit par des données dite attributs et des traitements dont elle est responsable vis-à-vis d'elle-même et des autres classe.

2.2.4.2.3 Diagramme de paquetage : un paquetage regroupe des éléments de la modélisation appelés aussi membres, portant sur un sous-ensemble du système. Le découpage en paquetage doit traduire un découpage logique du système à construire qui corresponde à des espaces de nommage homogènes.

2.2.4.2.4 Diagramme de déploiement : le diagramme de déploiement permet de représenter l'architecture physique supportant du système. Cette architecture comprend des nœuds qui correspondent aux supports physiques (serveurs, routeurs, etc.) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables) sur ces nœuds.

2.2.4.2.5 Diagramme de composant : permet de représenter les composants logiciels d'un système ainsi que les liens existant entre ces composants.

2.2.4.2.6 Diagramme d'objet : le diagramme d'objets permet la représentation d'instances des classes et des liens entre instances.

2.3 Méthodes de développement logiciel

Il existe de nombreuses méthodes de développement logiciel parmi elles on a cité [10] :

2.3.1 Méthode Agile

Les méthodes agiles sont des approches itératives et incrémentales qui visent à réduire le cycle de vie du logiciel en développant une version minimale, puis en

intégrant les fonctionnalités par les itérations basées sur une écoute du client et des tests tout au long du cycle de développement.

2.3.2 Processus UP

Le processus unifié est un processus de développement logiciel qui est :

- Itératif et incrémental : il se base sur la répétitions des traitements du processus de développement après validation d'une première version.
- Piloté par les cas d'utilisation d'UML : le but d'un système d'information est de collecter le maximum de besoins possibles des utilisateurs, et le cas d'utilisation est le moyen qui illustre ça.
- Centré sur l'architecture logicielle : Kruchten propose différentes perspectives, indépendantes et complémentaires, qui permettent de définir un modèle d'architecture (publication IEEE, 1995).

Donc l'objectif d'un processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques.

2.3.3 Méthode XP

EXtreme Programming est une méthode de développement d'agile, orientée objet et dans les ressources sont régulièrement actualisées. C'est une méthode de management qui permet d'accélérer la réalisation des projets.

2.4 Modélisation de notre système

Lors de cette partie nous allons nous concentrer sur la conception de notre application web, en indiquant les acteurs ainsi que leurs rôles et les différents diagrammes que nous jugeons importants.

2.4.1 Spécification des besoins

La spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle permet d'identifier les acteurs réactifs du système ainsi que l'ensemble des actions avec lesquelles ils interviennent afin de donner un résultat optimal et satisfaisant au client.

2.4.1.1 Identification des acteurs du système

Un acteur est un rôle incarné par une entité donnée (utilisateur humain, dispositif matériel ou logiciel) qui interagit directement avec le système étudié. Le tableau suivant nous décrit les différents acteurs existants dans notre application mais également le rôle qu'il joue dans cette dernière [11].

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages éventuellement porteurs de données.

Les acteurs candidats sont systématiquement :

- les utilisateurs humains directs : identifiez tous les profils possibles, sans oublier l'administrateur, l'opérateur de maintenance, etc.
- les autres systèmes connexes qui interagissent aussi directement avec le système.

Il existe 3 principaux acteurs dans notre application qui sont :

1- Visiteur : il s'agit de la personne qui visite seulement l'application sans avoir la possibilité de louer ou de publier des hébergements.

2- Utilisateur : il s'agit de la personne qui a la possibilité de visiter l'application mais aussi de réserver et publier un hébergement.

3- Administrateur : qui est en charge de la maintenance de l'application en supprimant, ajoutant ou modifiant les fonctionnalités de l'application, et également de contrôler et surveiller les publications des utilisateurs.

2.4.1.2 Identification des besoins du système

L'identification des besoins consiste à recueillir des besoins en utilisant le concept de cas d'utilisation proposé par UML où toute classe candidate du modèle statique d'analyse est identifiée.

2.4.1.2.1 Besoins fonctionnels Un système ne devient opérationnel que s'il satisfait les besoins fonctionnels (besoins métiers). Principalement, notre application doit couvrir les besoin fonctionnels représentés dans le tableau 2.1 associé à leurs acteurs.

Acteur	Rôle
Administrateur	Vérifier les publications.
	Supprimer les publications.
	Maintenance de l'application.
	Modifier fonctionnalité.
	Supprimer fonctionnalité.
	Ajouter fonctionnalité.
Utilisateur (Administrateur)	S'authentifier.
	Réserver hébergement.
	Supprimer réservation.
	Payer.
	Modifier réservation.
	Gérer compte utilisateur.
	Publier hébergement.
	Supprimer hébergement.
	Modifier hébergement.
Visiteur (administrateur, utilisateur)	Créer un compte.
	Recherche hébergement.
	Explorée les environnements touristiques.

TABLEAU 2.1 – Besoin fonctionnel de chaque acteur

2.4.1.2.2 Besoins non fonctionnels Les besoins non fonctionnels sont des exigences qui portent sur le comportement du système, mais permettent d'identifier

ses exigences internes et externes qui garantissent sa performance et le respect des exigences de l'utilisateur.

- *Performance* : Notre application doit assurer un temps de réponse minimum tout en répondant aux besoins du manipulateur.
- *La simplicité* : chaque utilisateur pourra utiliser cette application d'une manière facile et claire.
- *L'ergonomie de l'interface* : les interfaces doivent être simples et conviviales.
- *La modularité de l'application* : avoir un code simple, facile à maintenir et à comprendre en cas de besoin.
- *Sécurité* : utilisation de l'application en toute sécurité sans avoir peur.
- *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.

2.4.1.3 Expression des besoins fonctionnels

Afin de clarifier l'enchaînement et le déroulement des fonctionnalités de notre application, une description est nécessaire. Les tableaux ci-dessous décrivent la chronologie dans chaque cas d'utilisation.

2.4.1.3.1 Identification des cas d'utilisations La figure 2.2 représente le diagramme de cas d'utilisation.

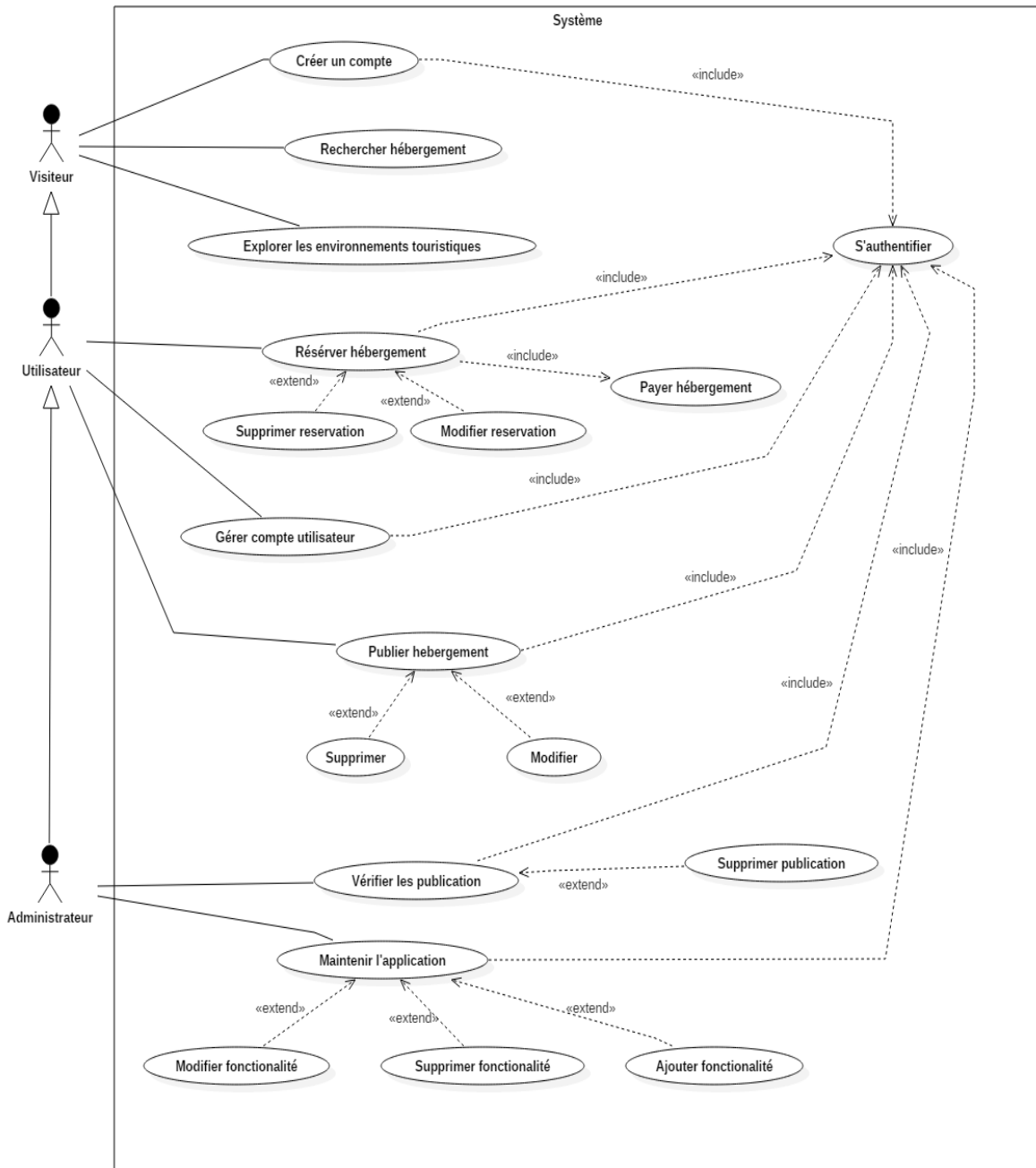


FIGURE 2.2 – Diagramme de cas d'utilisation.

i) Cas d'utilisation «Créer compte» :

Sommaire d'identification	
Titre	Créer compte.
Résumé	Créer compte pour pouvoir publier et louer un hébergement.
Acteur	Visiteur
Description des scénarios	
Pré-conditions	Le visiteur ne contient pas un compte utilisateur
Scénario nominal	L'utilisateur accède à l'application.
	Le système affiche l'accueil
	Le visiteur clique sur créer compte.
	Le système affiche un formulaire.
	Le visiteur remplit le formulaire.
	Le visiteur clique sur terminé.
	Le système envoie un email de validation.
Enchaînement d'erreurs	Email existe déjà.
	Mot de passe de confirmation n'est pas compatible.
	Le mot de passe n'est pas conforme avec la norme de l'application.
Post-conditions	Le Visiteur peut maintenant publier et louer un hébergement.

TABLEAU 2.2 – Cas d'utilisation «Créer compte»

ii) Cas d'utilisation «Recherche hébergement» :

Sommaire d'identification	
Titre	Recherche hébergement.
Résumé	Tout les acteurs peuvent faire une recherche d'hébergement
Acteur	Visiteur, utilisateur, administrateur
Description des scénarios	
Pré-conditions	Application accessible.
Scénario nominal	L'utilisateur accède à l'application.
	Le système affiche l'accueil.
	Le visiteur clique sur recherche d'un loyer.
Enchaînement d'erreurs	L'hébergement Recherché non trouver.
Post-conditions	Le visiteur obtient le résultat de sa recherche.

TABLEAU 2.3 – Cas d'utilisation «Recherche hébergement»

iii) Cas d'utilisation «Explorer les environnements» :

Sommaire d'identification	
Titre	Explorer les environnements.
Résumé	Tout les acteurs peuvent explorer les environnements touristiques
Acteur	Visiteur, utilisateur, administrateur
Description des scénarios	
Pré-conditions	Application accessible.
Scénario nominal	L'utilisateur accède à l'application.
	Le système affiche l'accueil.
	Le visiteur clique sur un environnement touristique parmi d'autres.
Enchaînement d'erreurs	
Post-conditions	le système affiche les hébergements de l'environnement choisie.

TABLEAU 2.4 – Cas d'utilisation «Explorer les environnements»

iv) Cas d'utilisation «S'authentifier» :

Sommaire d'identification	
Titre	S'authentifier.
Résumé	L'utilisateur introduit son login et mot de passe pour accéder à l'application.
Acteur	Visiteur, utilisateur, administrateur
Description des scénarios	
Pré-conditions	Application Web lancée.
Scénario nominal	L'utilisateur demande l'accès au système.
	Le système affiche le formulaire d'authentification.
	L'utilisateur saisit son login et mot de passe.
	Le système vérifie l'existence du compte.
	Si l'utilisateur est identifié, le système affiche l'interface de l'application.
Enchaînement d'erreurs	Email ou mot passe incorrecte.
	Le système affiche un message d'erreur.
	Le système ré-affiche le formulaire d'authentification.
Post-conditions	L'utilisateur peut louer ou publier un hébergement.

TABLEAU 2.5 – Cas d'utilisation «S'authentifier»

v) *Cas d'utilisation «Résérvé hébergement» :*

Sommaire d'identification	
Titre	Résérvé hébergement.
Résumé	L'utilisateur réserve l'hébergement de sont choix.
Acteur	Utilisateur, administrateur
Description des scénarios	
Pré-conditions	Utilisateur authentifié.
Scénario nominal	L'utilisateur accède à l'application.
	Le système affiche l'interface d'accueil.
	L'utilisateur s'authentifie.
	L'utilisateur fait une recherche d'hébergement.
	L'utilisateur choisie un hébergement.
	L'utilisateur saisie la data de séjour.
	L'utilisateur clique sur le bouton réserver.
	Le système affiche un formulaire à remplir.
L'utilisateur remplit le formulaire et le valide.	
Enchaînement d'erreurs	Hébergement non disponible à la date de séjour demendée.
Post-conditions	L'utilisateur paye et valide sa réservation.

TABLEAU 2.6 – Cas d'utilisation «Résérvé hébergement»

vi) *Cas d'utilisation «Payer» :*

Sommaire d'identification	
Titre	Payer.
Résumé	L'utilisateur paye son hébergement.
Acteur	Utilisateur, administrateur
Description des scénarios	
Pré-conditions	Authentification. L'utilisateur remplit le formulaire de réservation.
Scénario nominal	Le système affiche les modes de paiements disponibles.
	L'utilisateur choisit l'un des modes de paiement.
	L'utilisateur remplit le formulaire de paiement.
	L'utilisateur valide sa requête.
Enchaînement d'erreurs	Cordonnées de la carte invalide.
Post-conditions	L'utilisateur réserve l'hébergement.

TABLEAU 2.7 – Cas d'utilisation «Payer»

vii) Cas d'utilisation «Publier hébergement» :

Sommaire d'identification	
Titre	Publier hébergement.
Résumé	Permet à l'utilisateur de publier un hébergement.
Acteur	Utilisateur, administrateur
Description des scénarios	
Pré-conditions	Utilisateur authentifié.
Scénario nominal	L'utilisateur clique sur ajouter un hébergement.
	Le système affiche le formulaire à remplir.
	L'utilisateur remplit le formulaire.
	L'utilisateur clique sur ajouter.
Enchaînement d'erreurs	Formulaire de publication d'hébergement n'est pas rempli correctement.
Post-conditions	L'hébergement publié avec succès.

TABLEAU 2.8 – Cas d'utilisation «Publier hébergement»

viii) Cas d'utilisation «Gérer compte utilisateur» :

Sommaire d'identification	
Titre	Gérer compte utilisateur.
Résumé	Permet à l'utilisateur de modifier ses informations et observer son historique.
Acteur	Utilisateur, administrateur
Description des scénarios	
Pré-conditions	Utilisateur authentifié.
Scénario nominal	L'utilisateur clique sur gérer mon compte.
	L'utilisateur peut voir et modifier ses informations personnels.
Enchaînement d'erreurs	
Post-conditions	

TABLEAU 2.9 – Cas d'utilisation «Gérer compte utilisateur»

2.4.2 Analyse des besoins

Dans ce qui suit, nous allons citer les différents diagrammes utilisés dans ce projet à savoir les diagrammes de séquences et diagramme de classes ainsi que le modèle relationnel et le dictionnaire de données utilisés.

2.4.2.1 Diagrammes de séquences

Cette partie présente les différents diagrammes de séquences.

2.4.2.1.1 Diagramme de séquence «Créer compte» Le diagramme de séquence dans la figure 2.3 représente le scénario de création d'un compte de la part du visiteur. Le visiteur clique sur créer compte, un formulaire s'affiche. Après, il remplit le formulaire et le valide. Le système va ensuite vérifier si les champs ont été bien remplie (ou bien saisis). Si le visiteur oublie de remplir un champ ou qu'il est mal introduit, le système lui demande de re-saisir le champ, Sinon le compte va être créer avec succès et un email de confirmation va être envoyer.

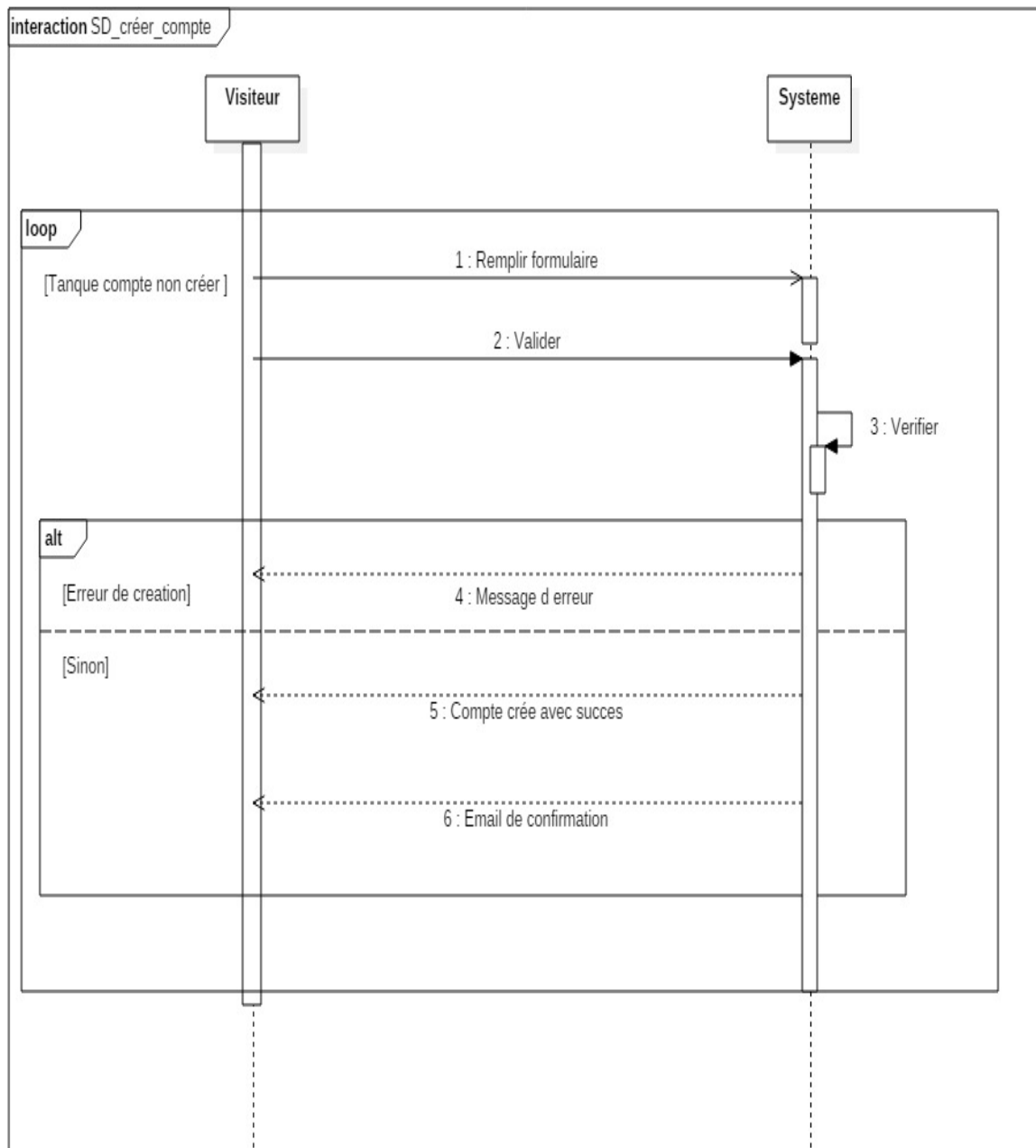


FIGURE 2.3 – Diagramme de séquence «Créer compte»

2.4.2.1.2 Diagramme de séquence «Recherche hébergement» Ce diagramme de séquence représente le scénario de recherche d'un hébergement par un visiteur. Le visiteur clique sur le champ recherche dans l'accueil pour rechercher un hébergement, ensuite le visiteur doit saisir les informations demandées (le lieu, la date et le nombre de personnes) et réalise sa recherche. Après, le système vérifie la disponibilité de l'hébergement recherché, s'il existe il affiche tous les hébergement correspondant à sa recherche, sinon le système n'affiche aucun résultat comme illustré dans la figure 2.4.

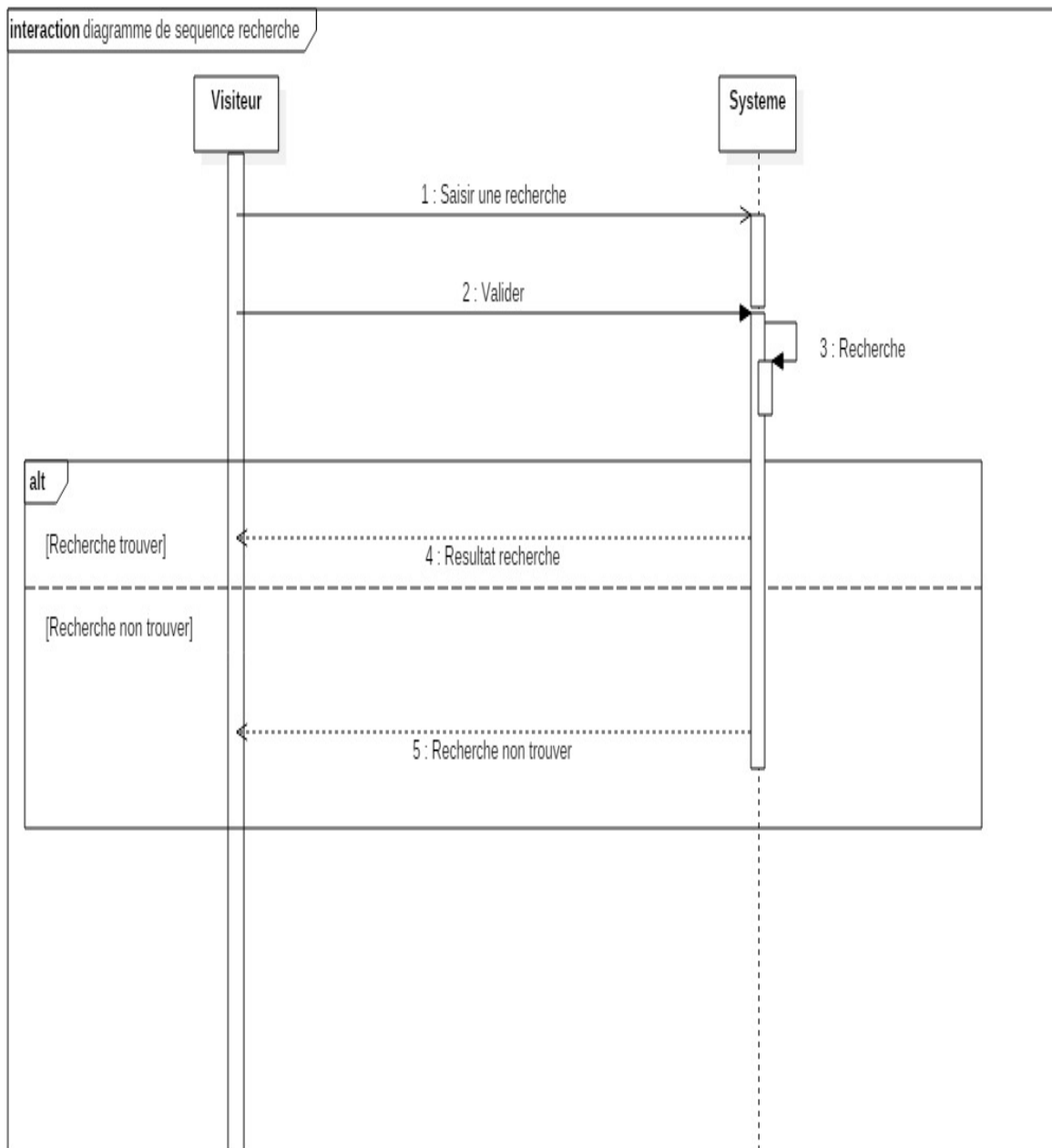


FIGURE 2.4 – Diagramme de séquence «Recherche hébergement»

2.4.2.1.3 Diagramme de séquence «Explorer les environnements» la figure 2.5 représente le diagramme de séquence du scénario d'exploration des environnements touristiques de la part de visiteur. Le visiteur explore les environnements touristiques qui sont dans la page d'accueil de l'application et qui contient les plus importants environnements touristiques. Après, le visiteur choisie un environnement touristique, le système affiche une page qui contient tout les hébergements qui existe dans l'application qui sont inclut dans l'environnement choisie.

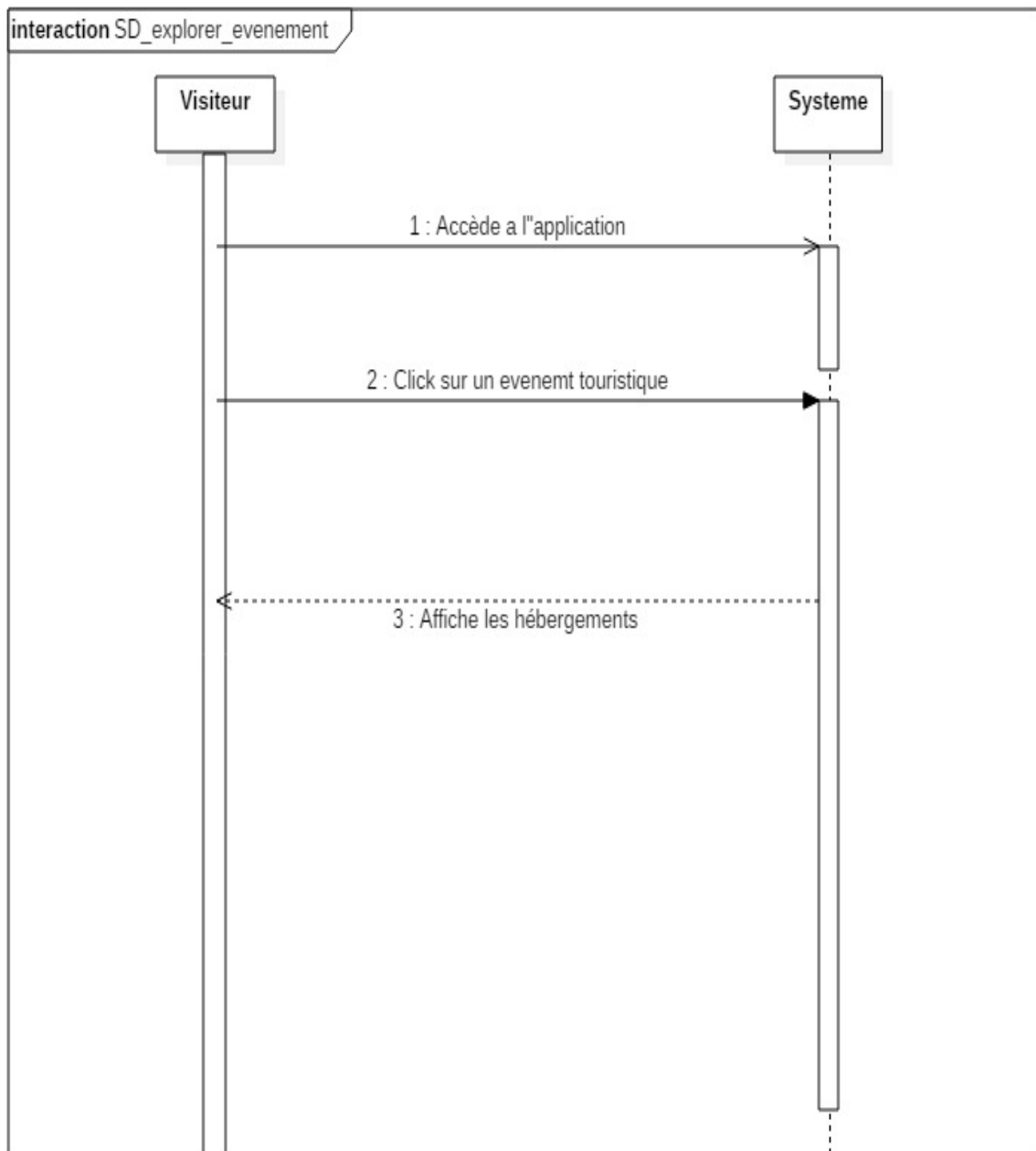


FIGURE 2.5 – Diagramme de séquence «Explorer les environnements»

2.4.2.1.4 Diagramme de séquence «S’authentifier» Le diagramme de séquence dans la figure 2.6 représenté le scénario d’authentification d’un utilisateur de l’application Web. Afin d’accéder aux fonctionnalités de notre application (comme ajouter ou réserver un hébergement), tous les acteurs (administrateur, utilisateur) doivent s’authentifier. L’utilisateur doit tout d’abord remplir le formulaire afin de s’authentifier en introduisant son email et son mot de passe puis valider sa requête. Le système va vérifier ensuite si les deux champs ont été bien saisis et bien conformes aux formats de données. Si l’utilisateur oublie un champ ou qu’il est mal introduit, le système lui demande de re-saisir sinon S’il n’y a pas d’erreur l’acteur va être authentifié.

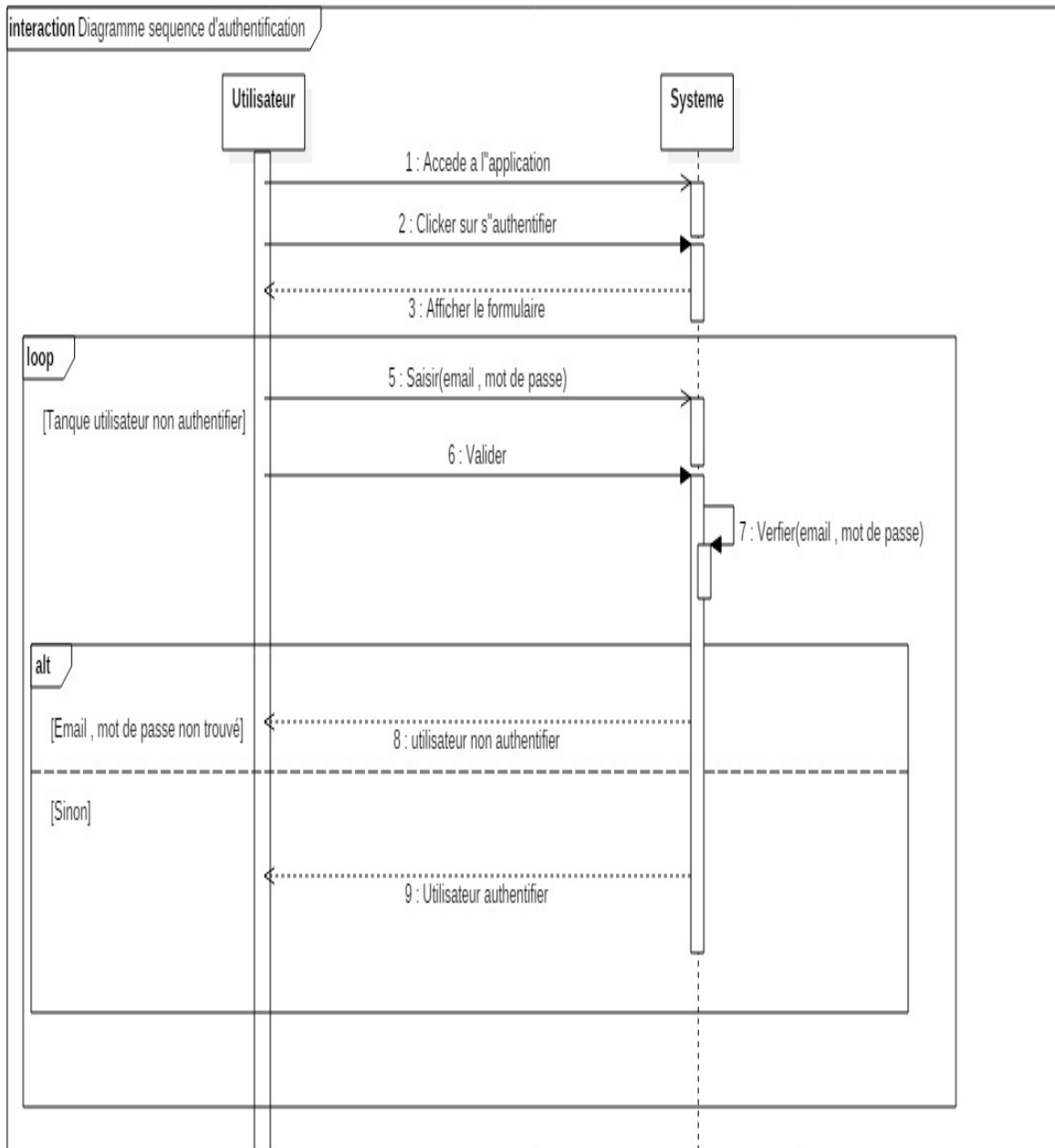


FIGURE 2.6 – Diagramme de séquence «S’authentifier»

2.4.2.1.5 Diagramme de séquence «Réserver hébergement» Le diagramme de séquence dans la figure 2.7 représente le scénario de réservation d'hébergement de la part de l'utilisateur. Après authentification, l'utilisateur choisie un hébergement qui souhaite réserver. Un formulaire à remplir est afficher. Après le remplissage du formulaire et sa validation, si les champs sont valides un message d'acceptation va apparaitre, sinon un message d'erreur apparaitra et le champ erroné va être indiquer. L'utilisateur clique sur payer pour payer sont hébergement ensuite valide sa requête et termine où un message d'acceptation final sera affiché.

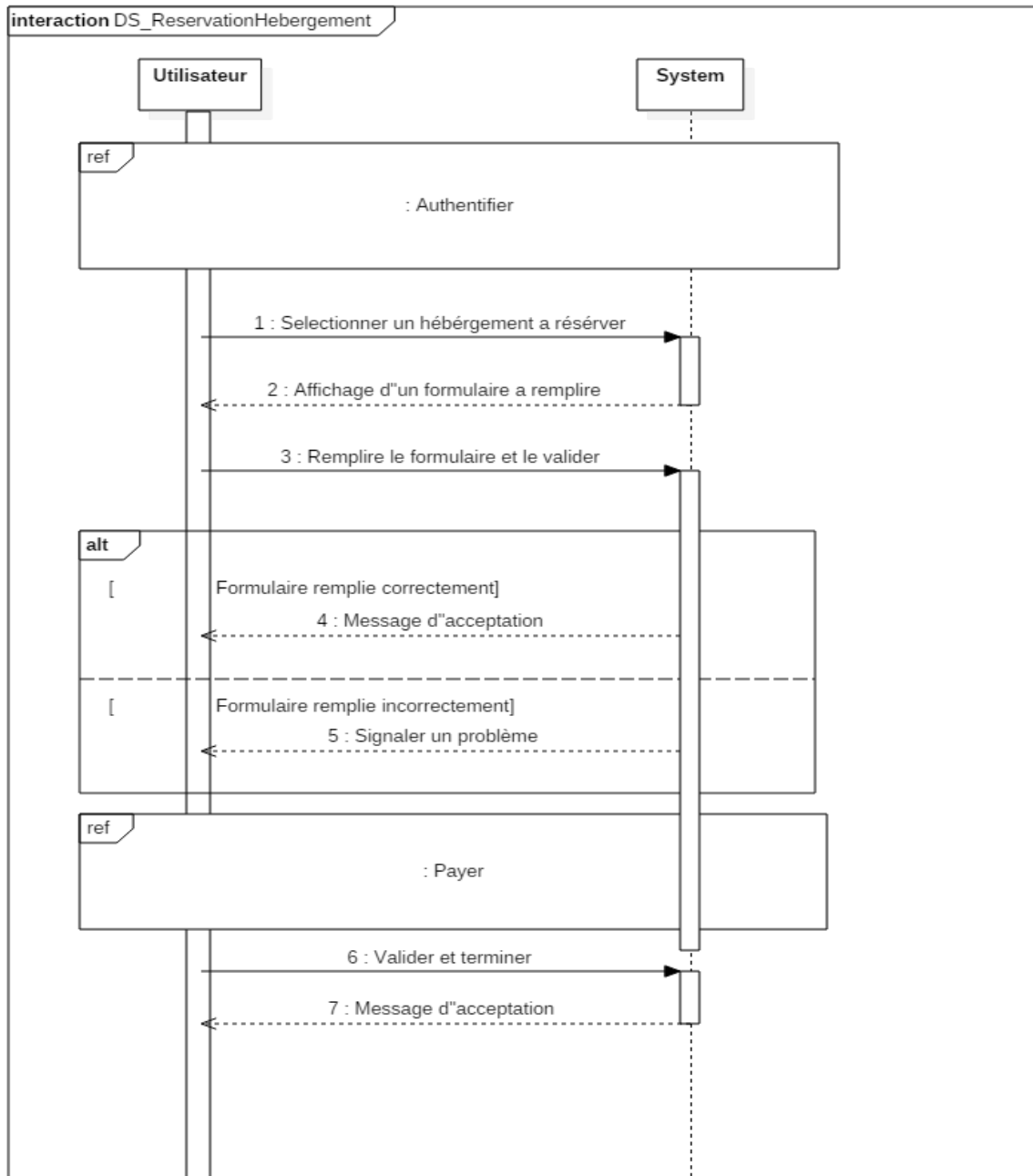


FIGURE 2.7 – Diagramme de séquence «Réserver hébergement»

2.4.2.1.6 Diagramme de séquence «Payer» Le diagramme de séquence dans la figure 2.8 représente le scénario de paiement d'hébergement de la part de l'utilisateur. Après authentification de l'utilisateur et le remplissage du formulaire de réservation, l'utilisateur doit payer sa réservation. En cliquant sur payer, un formulaire de paiement va être affiché où l'utilisateur remplit ce dernier et clique sur valider. Dans le cas où ces informations sont cohérentes et valides, un message d'acceptation de paiement apparaîtra, sinon un message d'erreur sera affiché.

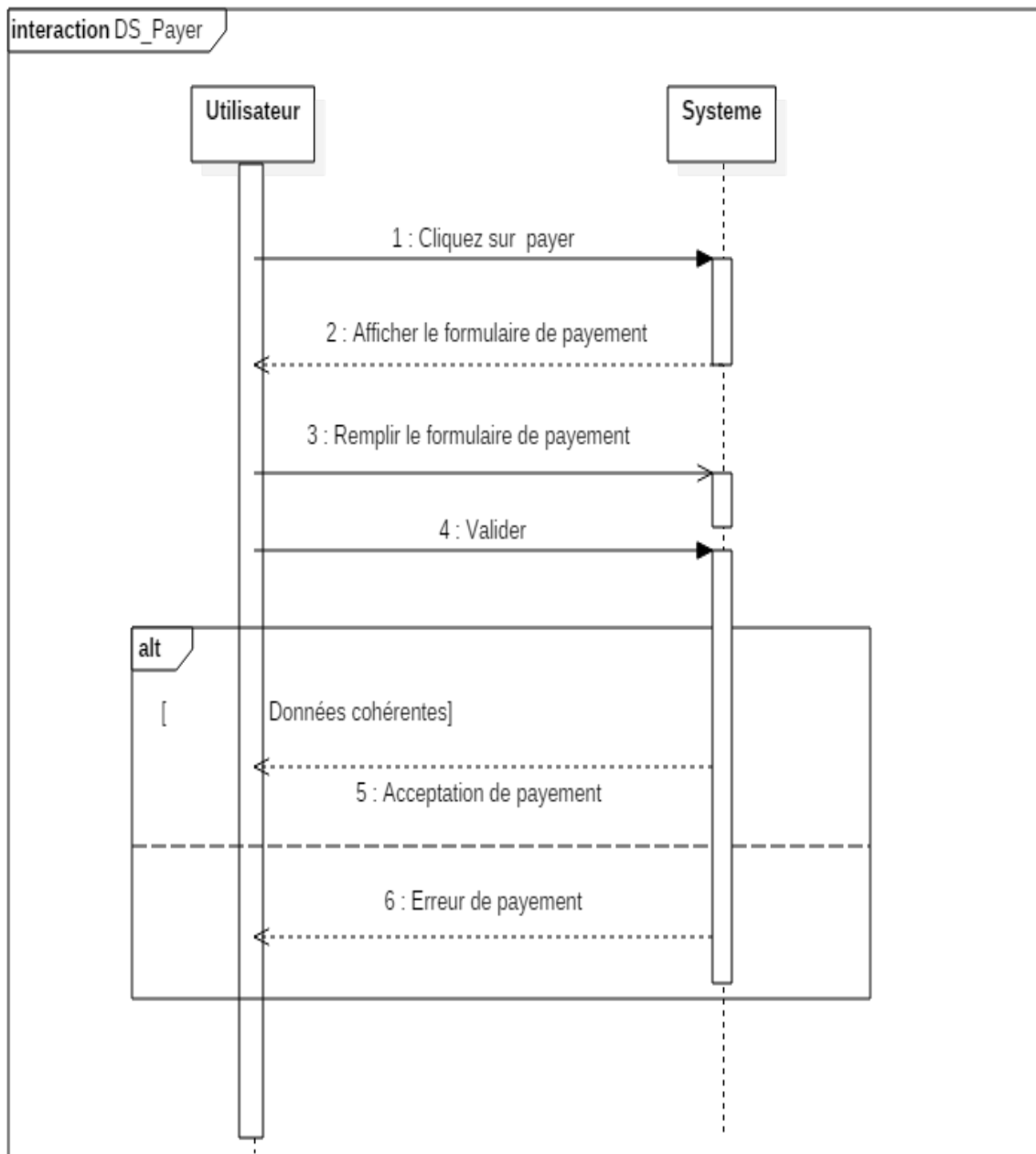


FIGURE 2.8 – Diagramme de séquence «Payer»

2.4.2.1.7 Diagramme de séquence «Publier hébergement» Ce diagramme de séquence représente le scénario d'ajout d'hébergement de la part de l'utilisateur. Après y être connecté, l'utilisateur clique sur ajouter hébergement et le système affiche un formulaire où l'utilisateur remplit ce dernier et clique sur terminer. Ensuite, le système vérifie la conformité de tous les champs tel que s'ils ne sont pas sous un format valide, le système affiche un message d'erreur et redemandera de ré-introduire les données.

Dans le cas où les champs sont valides, le système vérifie leurs cohérences (disponibilité, absence de chevauchements, etc) et mets à jour la page en ajoutant l'hébergement à la page des publications d'hébergements.

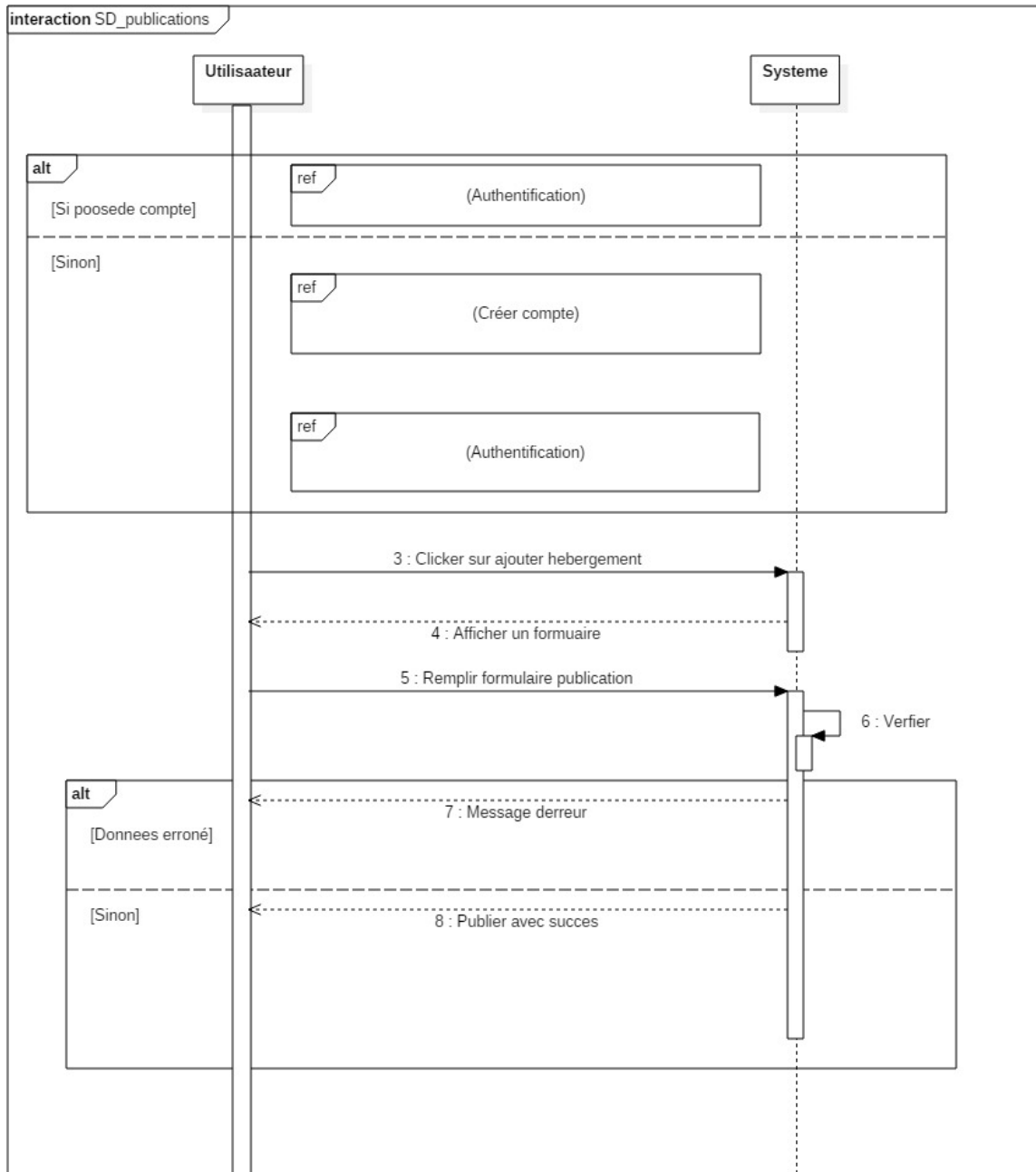


FIGURE 2.9 – Diagramme de séquence «Publier hébergement»

2.4.2.1.8 Diagramme de séquence «Gérer compte utilisateur» Ce diagramme de séquence dans la figure 2.10 représente le scénario de gestion d'un compte de la part de l'utilisateur. Après y être connecté, l'utilisateur clique sur gérer. Le système affiche une table qui contient les paramètres reliés à la gestion du compte utilisateur. Ensuite, l'utilisateur peut modifier ses informations personnelles (numéro de téléphone, photo de profil, etc.) comme il peut aussi consulter l'historique des locations et des transactions ainsi que le nombre de consultations de son hébergement s'il en a un.

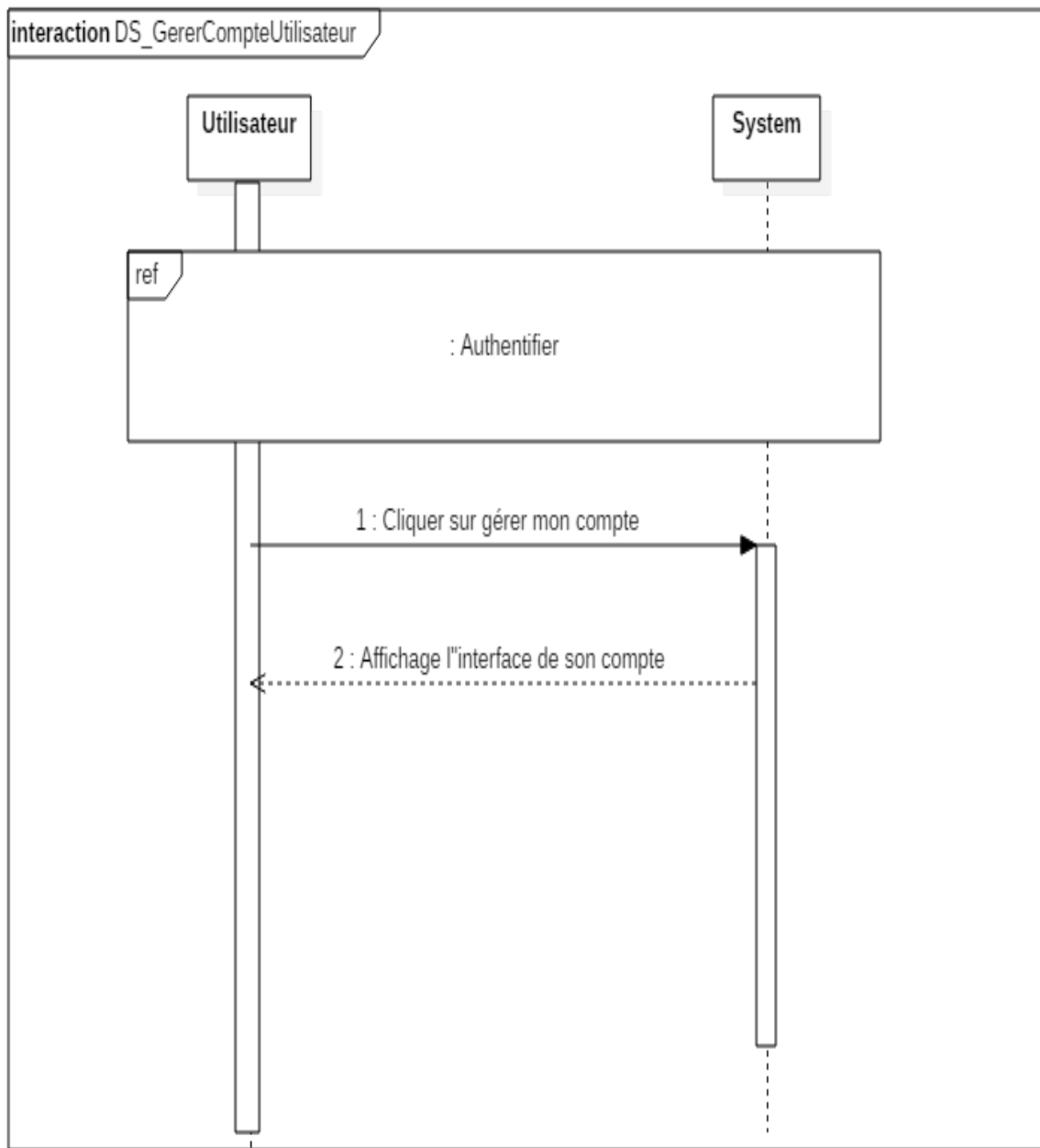


FIGURE 2.10 – Diagramme de séquence «Gérer compte utilisateur»

2.4.2.2 Diagramme de classes

Le diagramme de classes identifie les différentes entités du système caractérisé par des attributs ainsi que les associations entre elles comme le montre la figure 2.11.

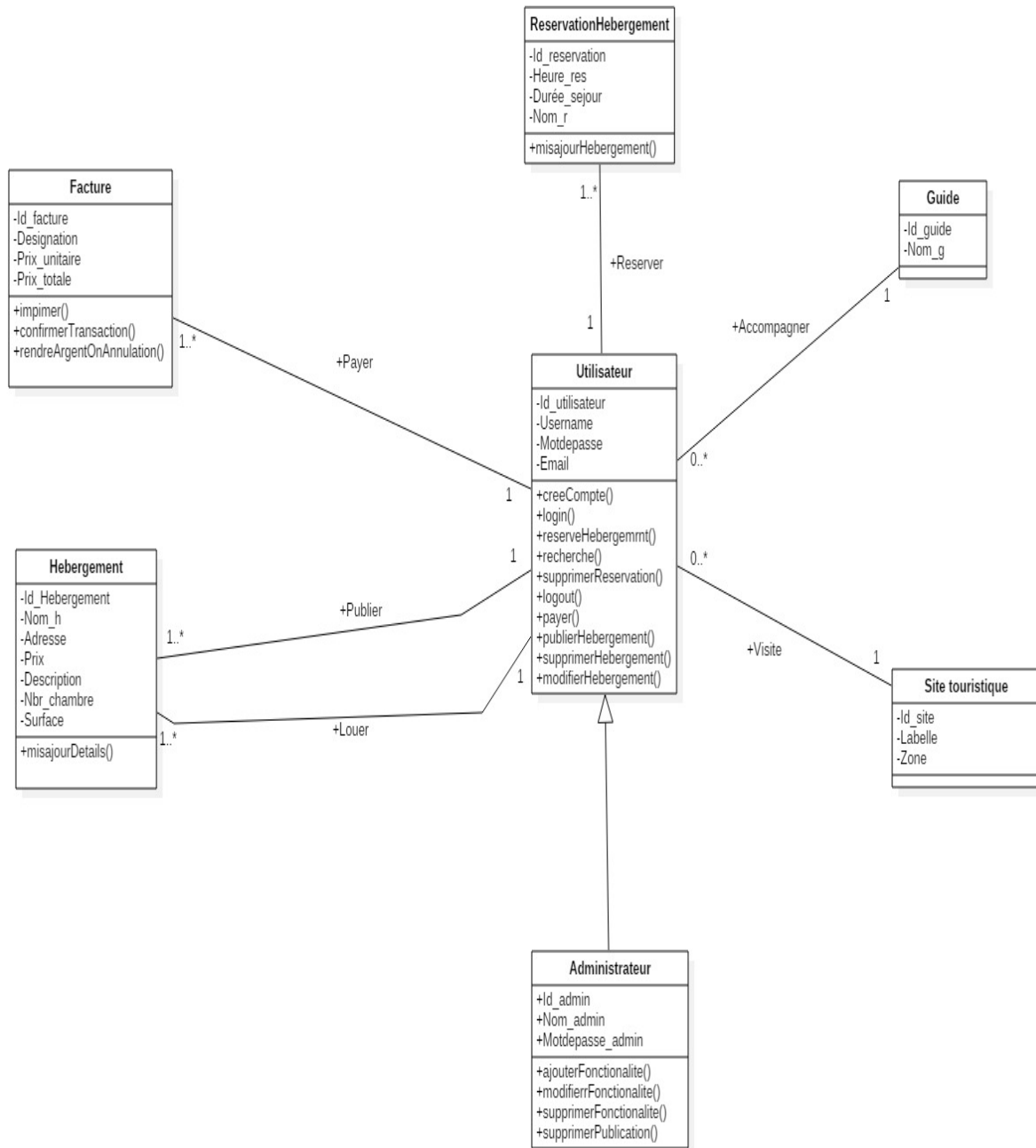


FIGURE 2.11 – Diagramme de classes

2.4.2.3 Modèle relationnel

En appliquant les règles de passages sur le diagramme de classe, nous obtiendrons le modèle relationnel de données correspondant à notre bases de données comme suit :

Utilisateur (id_utilisateur, Username, Motdepasse, Email, #Id_site, #Id_guide).

Administrateur (Id_admin, Nom_admin, Motdepasse_admin, #id_utilisateur, #Username, #Motdepasse, #Email, #Id_site, #Id_guide).

Hébergement (Id_hebergement, Nom_h, Adresse, Prix, Description, Nbre_chambre, Surface, #Id_utilisateur).

ReservationHébergement (Id_res, Heure_res, Duree_sejour, Nom_r, #id_utilisateur).

Facture (Id_facture, Designation, Prix_unitaire, Prix_total, #Id_utilisateur).

Sitetouristique (Id_site, Labelle, Zone).

Guide(Id_guide, Nom_g).

2.4.2.4 Dictionnaire des données

Le dictionnaire des données représenté dans le tableau 2.10 est un document qui regroupe toutes les données recensées, spécifiées et recueillies, que nous aurons à conserver dans notre bases de données (et qui figureront dans le diagramme de classes).

Nom de la classe	Codification	Désignation	Type	Remarque
Utilisateur	Id_utilisateur	identifiant de l'utilisateur	AN	au moins 8 caractères respecter le format : string@string.string
	Username	le nom de l'utilisateur	AN	
	Motdepasse	mot de passe de l'utilisateur	AN	
	Email	l'email de l'utilisateur	AN	
Administrateur	Id_admin	identifiant de l'administrateur	AN	au moins 8 caractères
	Nom_admin	le nom de l'administrateur	AN	
	Motdepasse_admin	mot de passe de l'administrateur	AN	
Reservation-Hebergement	Id_reservation	identifiant de la réservation	AN	
	Heure_res	l'heure de la réservation	Time	
	Duree_sejour	la durée de séjour	N	
	Nom_r	nom de la réservation	AN	
Guide	Id_guide	identifiant du guide	AN	
	Nom_g	le nom du guide	AN	
SiteTouristique	Id_site	identifiant du site touristique	AN	
	Labelle	labelle du site touristique	A	
	Zone	la zone ou se situe le site touristique	A	
Facture	Id_facture	identifiant de la facture	AN	
	Désignation	le nom de la facture	AN	
	Prix_unitaire	le prix d'une journée	N	
	Prix_total	le prix total de la réservation	N	
Hébergement	Id_hebergement	identifiant de l'hébergement	AN	donnez nous des images de taille 870x500
	Nom	le nom de l'hébergement	AN	
	Adresse	l'adresse de l'hébergement	AN	
	Prix	le prix l'hébergement	N	
	Description	description de l'hébergement	AN	
	Nbr_chambre	nombre de chambre de l'hébergement	N	
	Surface	la surface de l'hébergement	AN	
Image	image d'hébergement	PX		

TABLEAU 2.10 – Dictionnaire de données de l'application

Types de données du dictionnaire :

- Alphabétique(A) : la donnée est uniquement composée de caractères alphabétiques.
- Numérique(N) : la donnée est composée uniquement de nombres (entiers ou réels).
- Alphanumérique(AN) : La donnée peut être composée à la fois de caractères alphabétiques et numériques.
- Time : la donnée est une heure.
- PX : l'image est en pixel.

2.5 Conclusion

Durant ce chapitre, nous avons présenté quelques langages de modélisation de conception (MERISE et UML) pour ainsi penché sur l'une d'elle qui se trouve être l'UML. Puis, nous avons établis une modélisation des besoins du système grâce aux différents diagrammes (diagramme de séquence, diagramme de classes, diagramme de cas d'utilisation, etc.). Cette modélisation nous sera utile dans la partie implémentation.

Implémentation

3.1 Introduction

Après avoir abordé la partie conception approprié à notre projet nous allons, dans ce chapitre décrire la phase de réalisation de notre application. Ceci en spécifiant les outils, les langages et l'environnement de développement, puis une présentation des différentes interfaces de notre application.

3.2 Outils de développement

Dans cette partie, nous présentons les outils de développements que nous avons utilisés tous au long de notre travail.

3.2.1 Environnement logiciel

3.2.1.1 Visual studio

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (tels que C++, C#, Java,

Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity) [12]. La figure 3.1 présente le logo de visual studio.

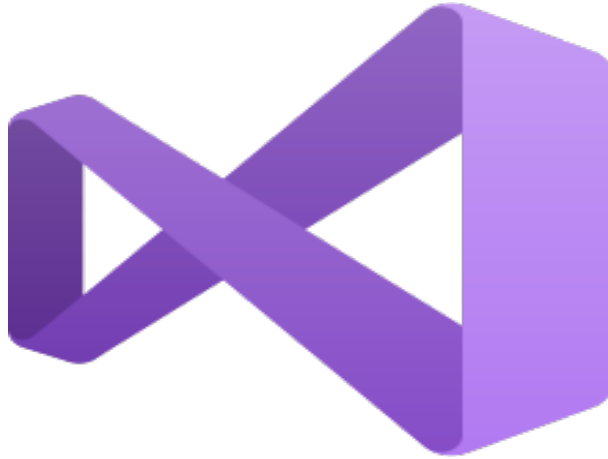


FIGURE 3.1 – Logo visual studio

3.2.1.2 npm

npm est le gestionnaire de paquets officiel de Node.js. Depuis la version 0.6.3 de Node.js, npm fait partie de l'environnement et est donc automatiquement installé par défaut. npm fonctionne avec un terminal et gère les dépendances pour une application. Il permet également d'installer des applications Node.js disponibles sur le dépôt npm [13]. La figure 3.2 présente le logo de npm.

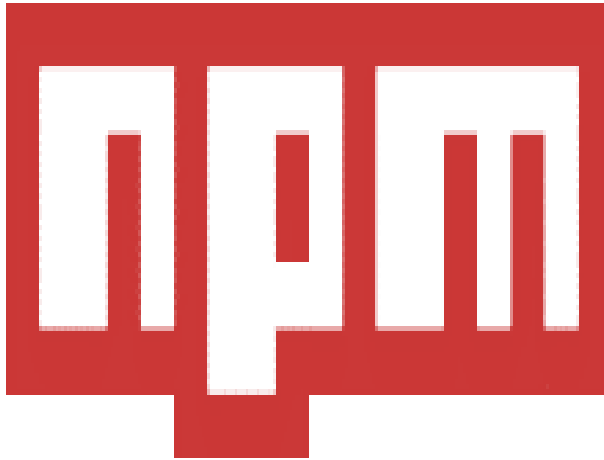


FIGURE 3.2 – Logo npm

3.2.1.3 Git Bash

Git Bash est une application pour les environnements Microsoft Windows qui fournit une couche d'émulation pour une expérience de ligne de commande Git. Bash est l'acronyme de Bourne Again Shell. Un shell est une application de terminal utilisée pour faire l'interface avec un système d'exploitation par le biais de commandes écrites. Bash est un shell par défaut populaire sous Linux et MacOS. Git Bash est un package qui installe Bash, quelques utilitaires Bash courants et Git sur un système d'exploitation Windows [14]. La figure 3.3 présente le logo de Git Bash.



FIGURE 3.3 – Logo Git Bash

3.2.1.4 Mailchimp

Mailchimp est une plateforme d'automatisation de marketing qui vous aide à partager des campagnes publicitaires et des campagnes d'e-mail avec des clients et d'autres parties intéressées. Notre stratégie de marketing par e-mails repose sur des pratiques de gestion des listes saines, des campagnes conçues de manière ingénieuse et d'une analyse profonde des données [15]. La figure 3.4 présente le logo de Mailchimp.



FIGURE 3.4 – Logo Mailchimp

3.2.1.5 Postman

Postman est un logiciel qui se focalise sur les tests des API. Il est devenu très populaire pour tester les Micro services, notamment grâce à sa simplicité et ses fonctionnalités très spécialisées [16]. La figure 3.5 présente le logo de Postman.



FIGURE 3.5 – Logo Postman

3.2.1.6 Hyper

Hyper est un terminal basé sur les technologies Web, il se base sur JavaScript, HTML et CSS, offrant une expérience magnifique et extensible aux utilisateurs d'interfaces en ligne de commande [17]. La figure 3.6 présente le logo de Hyper.



FIGURE 3.6 – Logo Hyper

3.2.1.7 Robo 3T

Robo 3T (anciennement Robomongo) est une interface graphique utilisateur simple gratuite pour MongoDB. Elle a pour principale fonction l'exécution de requêtes, la création d'index et l'affichage de documents. [18]. La figure 3.7 présente le logo de Robo 3T.

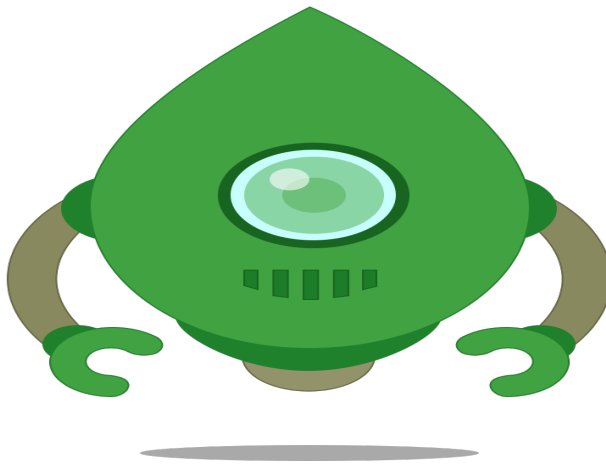


FIGURE 3.7 – Logo Robo 3T

3.2.1.8 StarUML

StarUML est un logiciel de modélisation UML, qui a été cédé comme open source par son éditeur, à la fin de son exploitation commerciale sous une licence modifiée de GNU GPL. Aujourd'hui la version StarUML V3 n'existe qu'en licence propriétaire. StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0. Il est écrit en Delphi1, et dépend de composants Delphi propriétaires (non open-source) [19]. La figure 3.8 présente le logo de StarUML.



FIGURE 3.8 – Logo StarUML

3.2.1.9 Heroku

Heroku est une plate-forme cloud en tant que service (PaaS) basée sur des conteneurs. Les développeurs utilisent Heroku pour déployer, gérer et faire évoluer des applications modernes. Cette plate-forme est élégante, flexible et facile à utiliser, offrant aux développeurs le chemin le plus simple pour mettre leurs applications sur le marché. Heroku est entièrement géré, ce qui donne aux développeurs la liberté de se concentrer sur leur produit principal sans être distraits par la maintenance des serveurs, du matériel ou de l'infrastructure. L'expérience Heroku fournit des services, des outils, des flux de travail et une assistance polyglotte, tous conçus pour améliorer la productivité des développeurs [30]. La figure 3.9 présente le logo de Heroku.



FIGURE 3.9 – Logo Heroku

3.2.2 Environnement de programmation

3.2.2.1 HTML5

Html5 (Hyper Text Markup Langage 5) est la dernière version en date du langage de développement web HTML. Il comprend de nouvelles balises et de nouveaux attributs pour les pages web et ouvre surtout de nouvelles possibilités de développements pour les applications web mobiles ainsi que l'utilisation et l'échange avec des fonctionnalités propres aux smart phones et Tablettes numérique (carnet d'adresse, appareil, etc.). Avant celui-ci seules des applications mobiles natives pouvaient l'utiliser [20]. La figure 3.10 présente le logo de HTML5.



FIGURE 3.10 – Logo HTML5

3.2.2.2 CSS3

Css3 (Cascading style Sheets 3) est un langage de feuille de style utilisé afin de décrire la présentation d'un document écrit en HTML ou en XML, Il décrit la façon dont les éléments doivent être affichés à l'écran. Ce dernier été parmi les langages principaux du web et a été standardisé par le W3C. Ce standard évolue sous forme de niveaux, CSS1 est désormais considéré comme obsolète, CSS2.1 correspond à la recommandation et CSS3, qui est découpé en modules plus petits est en voie de standardisation [21]. La figure 3.11 présente le logo de CSS3.



FIGURE 3.11 – Logo CSS3

3.2.2.3 JavaScript

JavaScript est un langage de script multi plateforme orienté objet utilisé pour rendre les pages Web interactives (par exemple, avoir des animations complexes, des boutons cliquables, des menus contextuels, etc.). Il existe également des versions côté serveur plus avancées de JavaScript telles que Node.js, qui vous permettent d'ajouter plus de fonctionnalités à un site Web que de télécharger des fichiers (comme la collaboration en temps réel entre plusieurs ordinateurs). Dans un environnement hôte (par exemple, un navigateur Web), JavaScript peut être connecté aux objets de son environnement pour fournir un contrôle programmatique sur eux. JavaScript contient une bibliothèque standard d'objets, tels que Array, Date, et Math, et un ensemble central d'éléments de langage tels que des opérateurs, des structures de contrôle et des instructions [22]. La figure 3.12 présente le logo de JavaScript.



FIGURE 3.12 – Logo JavaScript

3.2.2.4 Bootstrap

Bootstrap est une bibliothèque d'outils HTML, CSS et JavaScript destinés à aider les développeurs de site et d'applications web [23]. La figure 3.13 présente le logo de Bootstrap.



FIGURE 3.13 – Logo Bootstrap

3.2.2.5 jQuery

jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Cela simplifie considérablement la navigation et la manipulation de documents HTML, la gestion des événements, l'animation et Ajax avec une API facile à utiliser qui fonctionne sur une multitude de navigateurs. Avec une combinaison de polyvalence et d'extensibilité, jQuery a changé la façon dont des millions de personnes écrivent JavaScript [24]. La figure 3.14 présente le logo de jQuery.



FIGURE 3.14 – Logo jQuery

3.2.2.6 Node.js

En tant qu'environnement d'exécution JavaScript asynchrone piloté par des événements, Node.js est conçu pour créer des applications réseau évolutives. Node.js est similaire dans sa conception et influencé par des systèmes tels que Ruby's Event Machine et Python's Twisted. Node.js pousse le modèle d'événement un peu plus loin. Il présente une boucle d'événement comme une construction d'exécution au lieu d'une bibliothèque [25]. La figure 3.15 présente le logo de Node.js.



FIGURE 3.15 – Logo Node.js

3.2.2.7 Express.js

Express.js est un framework pour construire des applications web basées sur Node.js. C'est de fait le framework standard pour le développement de serveur en Node.js. L'auteur original, TJ Holowaychuk, le décrit comme un serveur inspiré de Sinatra dans le sens qu'il est relativement minimaliste tout en permettant d'étendre ses fonctionnalités via des plugins [26]. La figure 3.16 présente le logo de Express.js.



FIGURE 3.16 – Logo Express.js

3.2.2.8 MongoDB

MongoDB est un programme de base de données multi plateforme orienté document disponible à la source. Classé comme un programme de base de données NoSQL, MongoDB utilise des documents de type JSON avec des schémas facultatifs. MongoDB est développé par MongoDB Inc. et sous licence SSPL (Server Side Public License) [27]. La figure 3.17 présente le logo de MongoDB.



FIGURE 3.17 – Logo MongoDB

3.2.2.9 Passport

Passport est un middleware d'authentification pour Node.js. Extrêmement flexible et modulaire, Passport peut être déposé discrètement dans n'importe quelle application Web basée sur Express. Un ensemble complet de stratégies d'authentification de support à l'aide d'un nom d'utilisateur et mot de passe, Facebook, Twitter) [28]. La figure 3.18 présente le logo de Passport.



FIGURE 3.18 – Logo Passport

3.2.2.10 EJS

EJS est un langage de modèles simple qui vous permet de générer un balisage HTML avec du JavaScript simple. Aucune religiosité sur la façon d'organiser les choses. Pas de réinvention de l'itération et du flux de contrôle. C'est tout simplement du JavaScript [29]. La figure 3.19 présente le logo de EJS.



FIGURE 3.19 – Logo EJS

3.3 Présentation de l'application

Dans cette partie, nous présentons les différentes interfaces de l'application :

3.3.1 Interface d'accueil

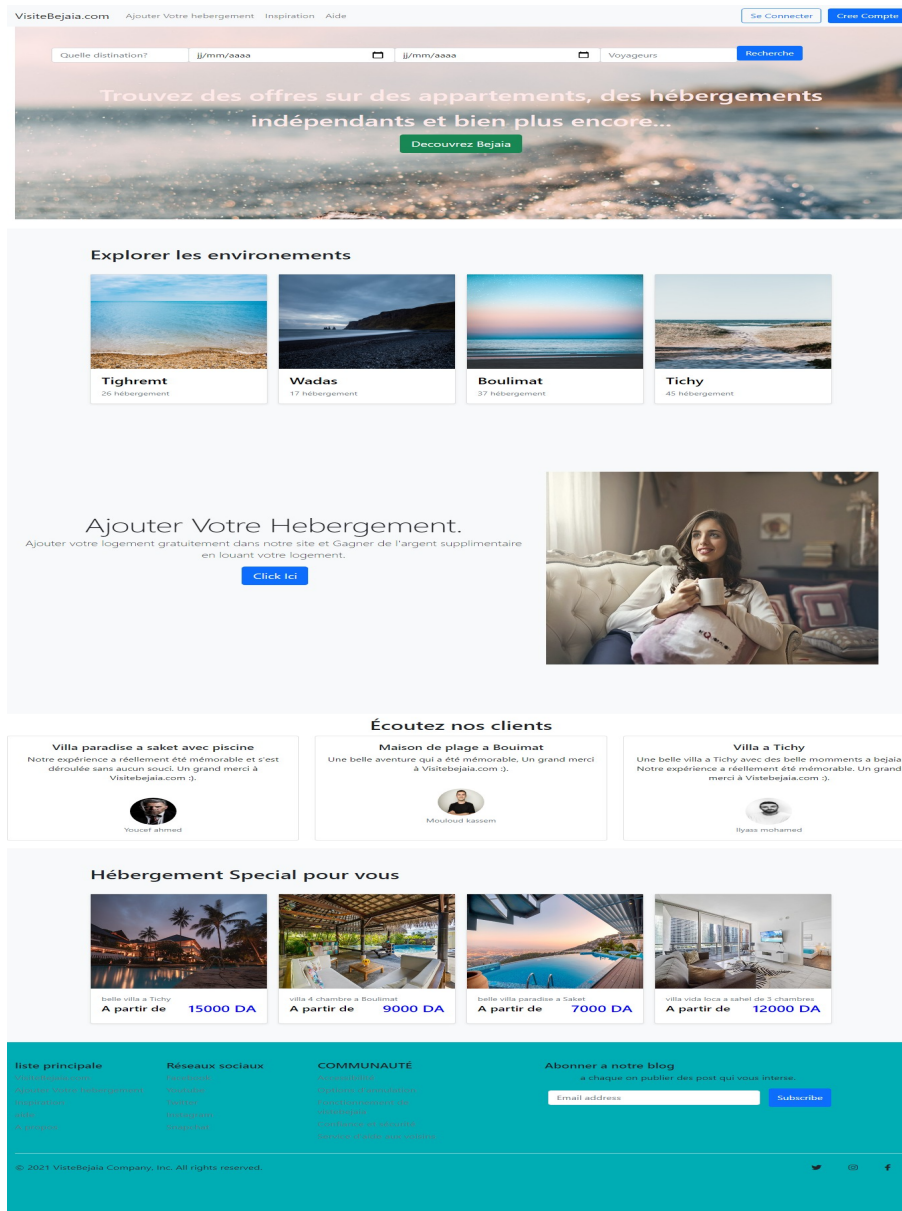
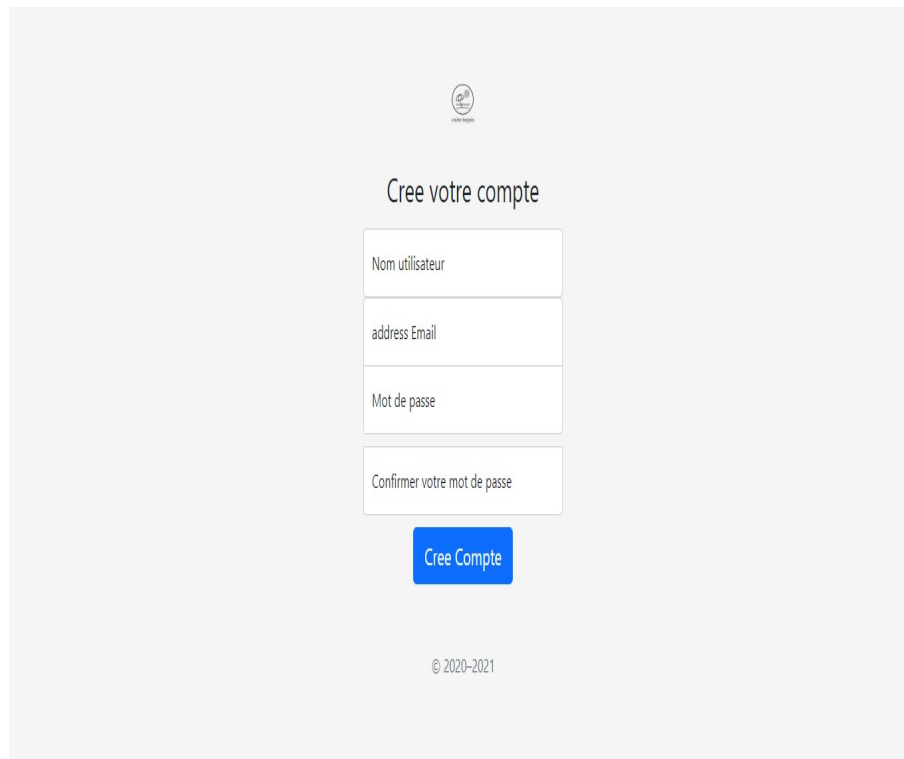


FIGURE 3.20 – Interface d'accueil

3.3.2 Interface correspondant à la création d'un compte



The image shows a user registration interface. At the top center, there is a small circular logo with the letters 'C' and 'P' inside. Below the logo, the title 'Cree votre compte' is displayed. The form consists of four input fields stacked vertically: 'Nom utilisateur', 'address Email', 'Mot de passe', and 'Confirmer votre mot de passe'. Below these fields is a blue button labeled 'Cree Compte'. At the bottom center, there is a copyright notice: '© 2020-2021'.

FIGURE 3.21 – Interface pour créer un compte

3.3.3 Interface d'authentification

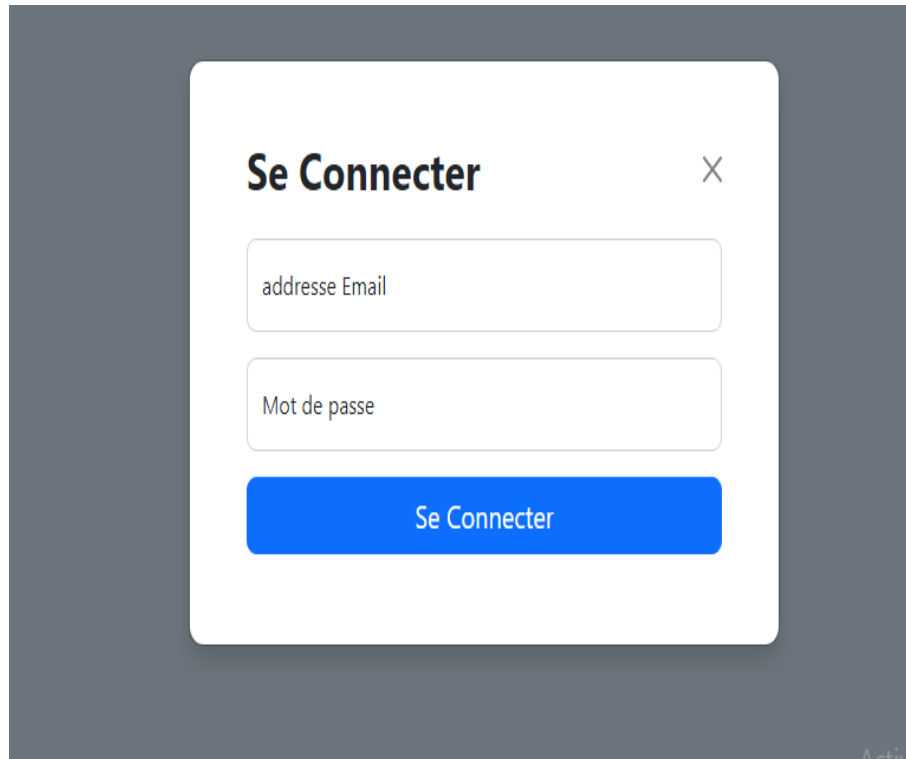


FIGURE 3.22 – Interface d'authentification

3.3.4 Interface correspondant à l’ajout d’un hébergement

The image shows a web form titled "Ajouter hébergement" on the website "VisiteBejaia.com". The form is contained within a light blue rounded rectangle and includes the following fields and elements:

- Nom d'hébergement :** A text input field.
- Type d'hébergement :** A dropdown menu with the text "Choisissez le type de votre hébergement".
- Nombre de chambre :** A text input field.
- Equipements :** A dropdown menu with the text "equipements".
- Surface par m2:** A text input field.
- Adresse:** A text input field.
- Capacité de voyageurs :** A text input field.
- Images :** A file selection area with a button labeled "Sélect. fichiers" and the text "Aucun fichier choisi".
- Prix :** A text input field.
- Description :** A large text area for entering details.
- Buttons:** Two blue buttons at the bottom of the form labeled "valider" and "annuler".

Below the form, the website's footer is visible on a teal background. It contains four main sections:

- liste principale:** Links to "VisiteBejaia.com", "Ajouter votre hébergement", "Inspiration", "Aide", and "À propos".
- Réseaux sociaux:** Links to "Facebook", "Youtube", "Twitter", "Instagram", and "Snapchat".
- COMMUNAUTÉ:** Links to "Accessibilité", "Options d'accessibilité", "Fonctionnement de notre blog", "Confiance et sécurité", and "Service à la clientèle".
- Abonner a notre blog:** A section with the text "à chaque on publier des post qui vous interese." and a "Subscribe" button next to an "Email address" input field.

At the bottom of the footer, there is a copyright notice: "© 2021 VisiteBejaia Company, Inc. All rights reserved." and social media icons for Twitter, Instagram, and Facebook.

FIGURE 3.23 – Interface pour ajouter un hébergement

3.3.5 Interface correspondant à la page d’hébergement

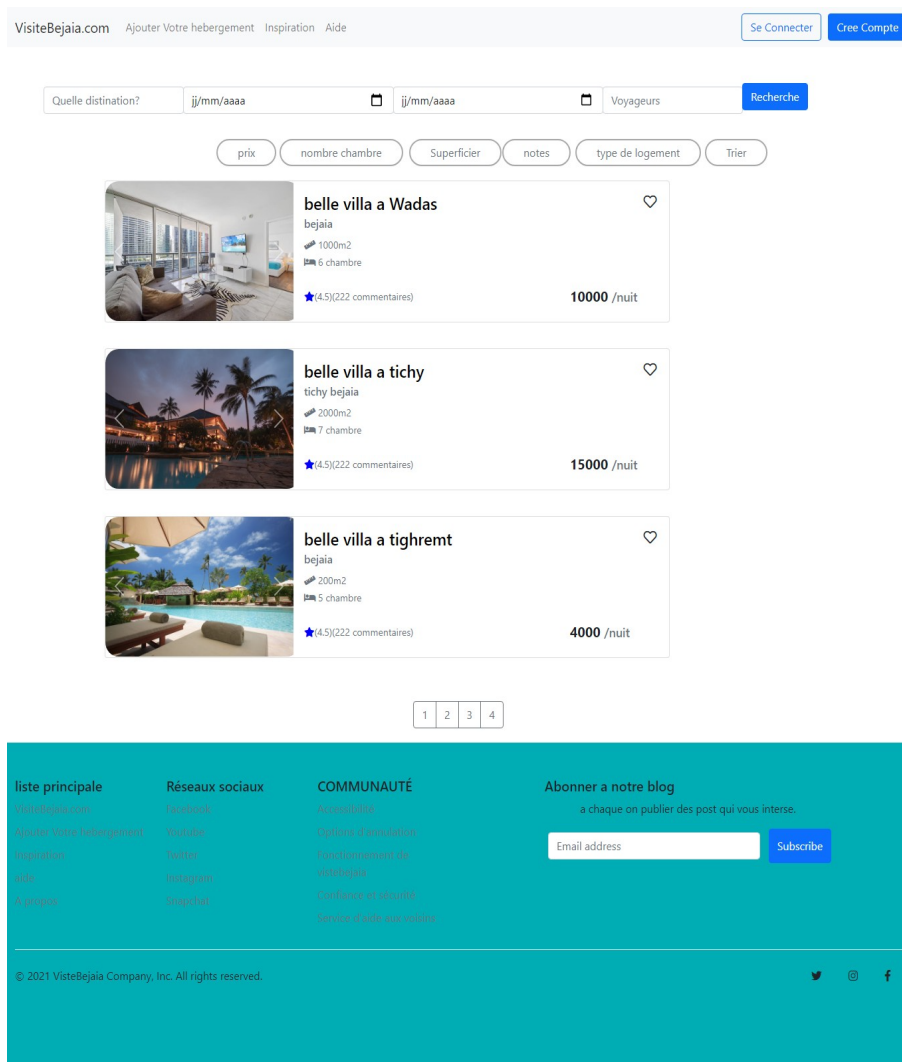


FIGURE 3.24 – Interface pour les hébergements

3.3.6 Interface correspondant aux détails sur un hébergement

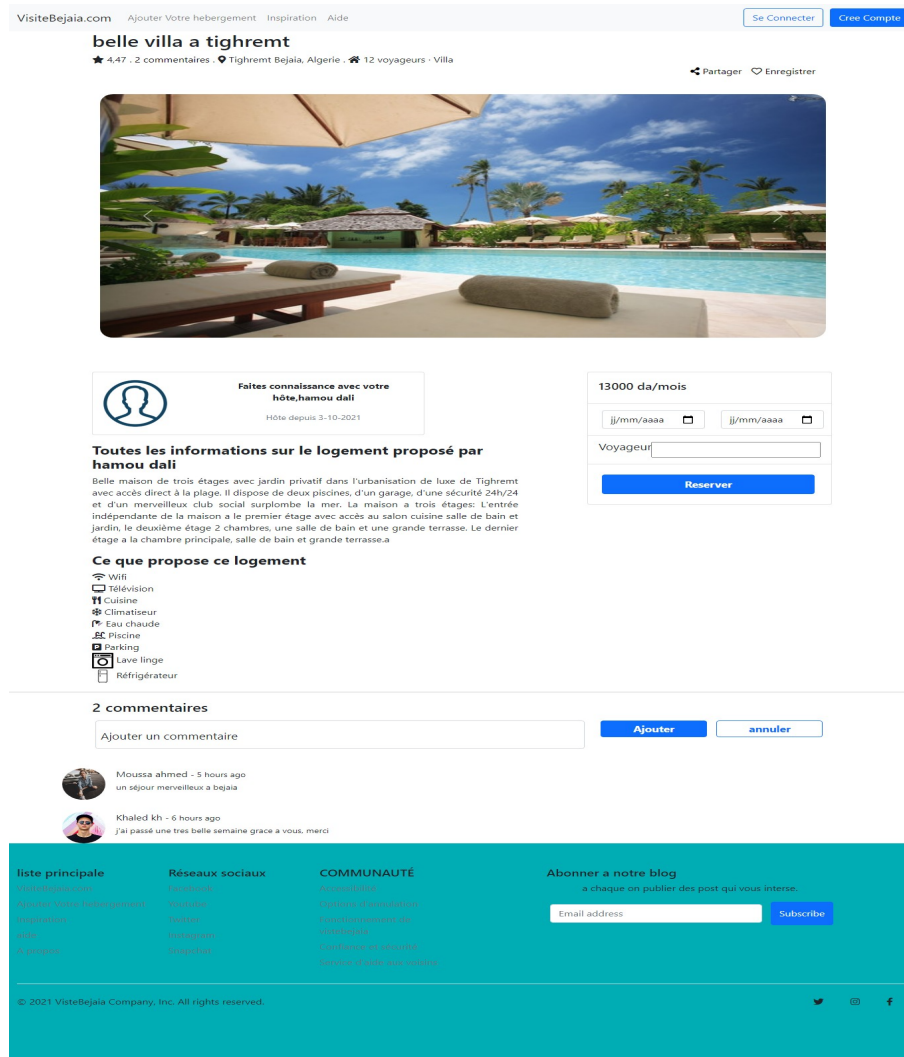


FIGURE 3.25 – Interface correspondant aux détails sur un hébergement

3.3.7 Interface correspondant à la réservation d'un hébergement

Votre reservation

Date d'arriver

Date de depart

Nombre de voyageurs

Payment

Credit card
 Debit card
 PayPal

Name on card

Credit card number


Full name as displayed on card

Expiration

CVV

Terminer votre reservation

Votre carte



Villa piscine a Boulimat

120m² 2 chambre

★(4.5)(222 commentaires)

3000*7 nuits	21000 DA
Frais de nettoyage	2000 DA
Total (DA)	23000 DA

Activer Windows
 Accédez aux paramètres pour activer Windows.

FIGURE 3.26 – Interface pour réserver un hébergement

3.4 Conclusion

L'implémentation et le codage restent la partie la plus importante dans le processus de création d'une application web. Lors de ce chapitre nous avons décrit brièvement les outils et les environnements de développement qu'on a choisi ainsi que quelques interfaces qu'on a jugé pertinentes.

Conclusion et perspectives

L'informatique nous montre son utilité dans la vie quotidienne du citoyen et des entreprises. C'est dans cette optique que rentre notre projet de fin de cycle. Nous avons réaliser une application web qui permettra au utilisateurs de réserver des hébergements sans avoir à se déplacer pour profiter pleinement de leur voyage.

L'idée générale du projet consiste à concevoir un outil applicatif qui pourra de façon concrète permettre à un utilisateur de réserver un hébergement en ligne à Béjaia, en choisissant parmi les offres proposées par l'application l'endroit où il souhaite passer ses vacances. L'administrateur utilisera le même système pour gérer et maintenir l'application d'une part et d'autre part effectuer les différentes tâches liées à l'application tel qu'une réservation, suppression ou la modification d'un hébergement. Pour la réalisation de ce projet, le langage de modélisation UML a été utilisé pour réaliser d'étape de conception. Les différents diagrammes utilisés ont permis de distinguer les rôles des différents acteurs concernés par l'application, leurs interactions avec le système, mais aussi de concevoir notre application de manière simple et efficace, ce qui nous a permis d'avoir une approche complète sur le développement logiciel.

La finalisation des différents services envisagés dans notre application a apporté de nouvelles ressources surtout dernièrement avec la pandémie du covid 19. Néanmoins, de nouvelles améliorations peuvent être élaboré tel qu'un service de messagerie pour les utilisateurs souhaitant se renseigner sur un hébergement spécifique.

Bibliographie

- [1] K. Kamlesh and N. Debjani, *E-commerce*, Tata McGraw-Hill Education, 2005
- [2] K. Steeves What Exactly Is an Online Booking System ? [en ligne], (*page consultée le 24/08/2021*), <https://www.checkfront.com/blog/what-is-an-online-booking-system>
- [3] A fresh look at online booking systems [en ligne], (*page consultée le 24/08/2021*), <https://www.hotelmize.com/blog/what-is-an-online-booking-system-and-how-do-i-choose-one/>
- [4] Booking.com [en ligne], (*page consultée le 24/08/2021*), <https://fr.wikipedia.org/wiki/Booking.com>
- [5] Airbnb [en ligne], (*page consultée le 24/08/2021*), <https://fr.wikipedia.org/wiki/Airbnb>
- [6] C. Morley and J. Hugues and B. Leblanc, *Description d'UML pour l'analyse d'un système d'information*, DUNOD, *UML2 pour l'analyse d'un système d'information*, Paris, 2000
- [7] G. Roy, *Conception de bases de données avec UML*, PUQ, 1ère édition, 2007
- [8] J. L. Mbenga, *Mise en œuvre d'un prototypage alpha mercuriale de gros œuvre dans la construction mémoire de licence informatique*, Informatique et Télécommunications, Université pédagogique nationale de Kinshasa, Congo, 2012

- [9] J. Gabay and D. Gabay, *Les diagrammes comportementaux*, DUNOD, *UML2 Analyse et conception*, Paris, 2008, p. 61
- [10] P. Roques, *Conception objet détaillé*, EYROLLES, *Les cahiers du programmeur UML2 modélisé une application web*, Paris, 2008
- [11] P. Roques and F. Vallee, *UML 2 en action : de l'analyse des besoins à la conception*, Eyrolles, paris, 2007
- [12] Documentation for Visual Studio Code [en ligne], (*page consultée le 07/06/2021*), <https://code.visualstudio.com/docs>
- [13] npm [en ligne], (*page consultée le 08/06/2021*), <https://fr.wikipedia.org/wiki/Npm>
- [14] Git Bash - Atlassian [en ligne], (*page consultée le 08/06/2021*), <https://www.atlassian.com/fr/git/tutorials/git-bash>
- [15] Prise en main de Mailchimp [en ligne], (*page consultée le 08/06/2021*), <https://mailchimp.com/fr/help/getting-started-with-mailchimp/>
- [16] Openclassroom : testez votre API grâce à Postman [en ligne], (*page consultée le 08/06/2021*), <https://openclassrooms.com/fr/courses/4668056-construisez-des-microservices/5123020-testez-votre-api-grace-a-postman>
- [17] Best Packages and Themes for Hyper : The Terminal Built on Web Technologies [en ligne], (*page consultée le 04/10/2021*), <https://scotch.io/tutorials/best-packages-and-themes-for-hyper-the-terminal-built-on-web-technologies>
- [18] Connexion avec Robo3T [en ligne], (*page consultée le 04/10/2021*), <https://cloud.ibm.com/docs/ComposeForMongoDB?topic=ComposeForMongoDB-connexion-avec-robo3t&locale>
- [19] StarUML [en ligne], (*page consultée le 04/10/2021*), <https://fr.wikipedia.org/wiki/StarUML>

- [20] Technologies web pour développeurs [en ligne], (*page consultée le 07/06/2021*),
<https://developer.mozilla.org/fr/docs/Web/HTML>
- [21] CSS : qu'est-ce que c'est? [en ligne], (*page consultée le 08/06/2021*), ,
<https://www.futura-sciences.com/tech/definitions/internet-css-4050/>
- [22] Web technology for developers [en ligne], (*page consultée le 07/08/2021*),
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- [23] Build fast, responsive sites with Bootstrap [en ligne], (*page consultée le*,
<https://getbootstrap.com/>
- [24] What is jQuery ? [en ligne], (*page consultée le 08/06/2021*), <https://jquery.com/>
- [25] About Node.js [en ligne], (*page consultée le 07/06/2021*),
<https://nodejs.org/en/about/>
- [26] Express.js [en ligne], (*page consultée le 07/06/2021*),
<https://fr.wikipedia.org/wiki/Express.js>
- [27] MongoDB [en ligne], (*page consultée le 07/06/2021*),
<https://en.wikipedia.org/wiki/MongoDB>
- [28] Passport [en ligne], (*page consultée le 04/06/2021*), <http://www.passportjs.org/>
- [29] What is EJS? [en ligne], (*page consultée le 04/06/2021*), <https://ejs.co>
- [30] What is Heroku? [en ligne], (*page consultée le 05/10/2021*),
<https://www.heroku.com/about>

RÉSUMÉ

Avec la prolifération de la pandémie du covid 19 ces deux dernière années, le voyage international est devenu difficile en raison des protocoles sanitaires à respecter. de ce fait, les gens se sont tournés vers des voyages nationaux et visiter diffeéretes wilaya de leur choix. En raison du manque d'application web reliée aux réservation d'hébergement en algerie de manière générale mais surtout à Béjaia de manière spécifique, nous avons proposé une application web d'hébergement en ligne de location de vacances, cabanes, maison de plages et logements. Cette application vise à faciliter aux utilisateurs d'effectuer leur réservations en avance. Cela laisse clairement plus de temps, de chercher, bien choisir etc. Mais aussi d'obtenir de meilleurs prix en comparant entre les offres proposées.

Mots clés : application web, UML, MongoDB, HTML5, CSS3, jQuery, Javascript, Node.js.

ABSTRACT

With the proliferation of the covid 19 pandemic over the past two years, international trips has become difficult. Therefore, people have turned to national trips in order to visit different town of their choices. Due to the lack of web application related to reservations in Algeria generally and in Béjaia especially. In this thesis, we have proposed an online web application for vacation cabins, beach houses and lodging. This application aims at providing an easier way for users wishing to spend their holidays in bejaia to make their reservations in advance. Furthermore, this application clearly allows reducing more time search, choose well but also obtaining better reservation prices by comparing between the present offers.

Key words : web application, UML, MongoDB, HTML5, CSS3, jQuery, Javascript, Node.js.