



République Tunisienne  
\*\*\*\*  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
\*\*\*\*  
Université de Monastir  
\*\*\*\*  
Institut Supérieur d'Informatique et de Mathématiques de Monastir  
\*\*\*\*



Département de Technologie

N° d'ordre : LF 9



# Projet de Fin d'Etudes

*Présenté en vue de l'obtention du*

Diplôme National de Licence Fondamentale en Sciences  
Informatique

*par*

**Sarra Ben Haddada & Skander Bouraoui**

---

## Conception et Réalisation d'un portail web pour le secteur touristique en Tunisie

---

*Soutenu le 09/07/2020 devant le jury composé de :*

M. Souheyl Mallat  
M. Malek Ben Salem  
Dr. Ramzi Mahmoudi

Président  
Membre  
Encadrant

# Résumé

Dans le cadre de notre projet de fin d'études à l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM) et en vue de l'obtention du Diplôme National de Licence Fondamentale en Science Informatique, nous avons effectué notre stage de fin d'études au sein de la société « GPRO Consulting ».

Ce projet consiste à concevoir et développer un portail web pour le secteur touristique en Tunisie qui favorise la destination communautaire, attire les visiteurs ainsi que des nouveaux prestataires, investisseurs et résidents. Beaucoup d'opportunités seront offertes via l'application..

**Mots clés : React Js, Redux, Node JS, Express JS, PostgreSQL, Knex JS, Power BI, Auth0, UML,...**

# Abstract

As part of our graduation project at the Higher Institute of Computer Science and Mathematics of Monastir (ISIMM) and in order to obtain the Diploma of Fundamental License in Computer Science, we passed our internship of the end of studies within the company "GPRO Consulting".

This project is used to design and develop a web portal for the tourism sector in Tunisia which promote community destination, attract visitors as well as new businesses, investors, and residents. Many opportunities can be found through this application.

**Key words : React Js, Redux, Node JS, Express JS, PostgreSQL, Knex JS, Power BI, Auth0, UML,...**

# Dédicace

Du profond de mon cœur, je dédie ce modeste travail à tous ceux qui sont chers :

A mes parents : **Zouheir & Nabila**

Aucune dédicace ne saurait exprimer mon respect, amour éternel et ma considération pour les sacrifices que vous avez consentis pour mon instruction et mon bien être.

Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

A mes sœurs **Meryam & Molka**, A mon frère **Mehdi**

En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.

A mes amis de toujours :

Khalil Guedria, Hamza Beizig, Slim Ben Turkia, Ahmed Soumer,

Sarra Ben Haddada,

Zayneb Belhadj, Syrine Ben Rayana, Rania Charef,...

&

Toute la famille de IEEE ISIMM Student Branch

Pour toutes les aventures que nous avons vécues ensemble et pour vos encouragements, pour les moments d'échec, de réussite, de joie et de tristesse que nous avons partagés. En souvenir des agréables moments que nous avons passés ensemble. Je vous souhaite une bonne continuation dans vos vies professionnelles.

~~~~~ Skander Bouraoui ~~~~

# Dédicace

Au moment où s'achève le travail, il m'est agréable d'exprimer mes remerciements à tous ceux qui m'ont aidée

## À mes chers Parents

Mes plus profonds remerciements vont à mes parents. Tout au long de mon cursus, ils m'ont soutenue, encouragée et aidée. Ils ont su me donner toutes les chances pour réussir. Qu'ils trouvent, dans la réalisation de ce travail, l'aboutissement de leurs efforts ainsi que l'expression de ma plus affectueuse gratitude

♡

## À mes Sœurs

Ma fierté, ensemble nous avons grandi, nous avons ri, nous nous sommes disputé... mais jamais séparées. Pour votre amour et vos aides. Je vous souhaite le bonheur, la joie et la réussite.

♡

## À mon cher ami Mohamed Boukadida,

Qui n'a cessé de m'encourager et de me soutenir. Je serai reconnaissante pour l'aide qu'il m'a prodiguée et pour ses conseils précieux.

♡

## À mes amis et toutes personnes qui me sont chères,

Nour el Houda, Skander Bouraoui et Amira Jalouli

Avec qui j'ai passé mes meilleurs moments, avec qui j'ai partagé des souvenirs. Amis pour la vie, amis pour toujours. Je vous aime.

♡

## À moi-même,

Pour toutes les nuits blanches...



# Remerciements

Nous tenons à remercier dans un premier temps, toute **l'équipe pédagogique de l'ISIMM** et les intervenants professionnels responsables de la formation sciences de l'informatique, pour avoir assuré la partie théorique de celle-ci.

Nous tenons à exprimer également notre profonde gratitude et nos remerciements les plus sincères à :

## Dr. Ramzi Mahmoudi

Notre encadrant pédagogique pour son minutieux ..., sa patience, ses conseils pertinents et pour l'intérêt qu'il nous a accordé tout au long de ce travail. Acceptant de nous encadrer à ce PFE est une faveur pour nous dont nous sommes très fiers et reconnaissants.

## Mr. Ghazi Atroussi

Notre maître de stage et notre encadreur dans la société GPRO Consulting pour son accueil, ses conseils pratiques et directives, sa confiance, le temps passé ensemble et le partage de son expertise au quotidien.

## Aux membres du jury

Qui ont accepté d'évaluer ce projet de fin d'études en espérant qu'ils trouvent dans ce rapport les qualités de clarté et de motivations qu'ils attendent.

# Tableau des Matières

|                                                     |    |
|-----------------------------------------------------|----|
| Chapitre I : Contexte et objectifs du projet.....   | 1  |
| Introduction.....                                   | 1  |
| 1    Contexte général .....                         | 1  |
| 1.1    Cadre du projet.....                         | 1  |
| 1.2    Présentation de l'entreprise d'accueil ..... | 1  |
| 1.3    Idée maitresse .....                         | 3  |
| 2    Problématique .....                            | 3  |
| 3    Etude de l'existant.....                       | 3  |
| 3.1    Solutions Existantes.....                    | 4  |
| 3.1.1    Solutions en Tunisie.....                  | 4  |
| 3.1.2    Solutions à l'étranger .....               | 7  |
| 3.2    Critique de l'existant.....                  | 8  |
| 4    Solution proposée .....                        | 9  |
| 4.1    Description textuelle .....                  | 9  |
| 4.2    Description visuelle .....                   | 10 |
| 5    Méthodologie de développement .....            | 10 |
| 5.1    Étude de Méthodologie de développement.....  | 11 |
| 5.2    Présentation de quelques méthodologies.....  | 11 |
| 5.2.1    Agile .....                                | 11 |
| 5.2.2    Scrum.....                                 | 11 |
| 5.2.3    Unified Process .....                      | 12 |
| 5.2.3.1    Two Track Unified Process (2TUP).....    | 13 |
| 5.2.3.2    Rational Unified Process (RUP) .....     | 13 |
| 5.3    Étude comparative et évaluation.....         | 14 |
| 5.4    Choix de la méthodologie de travail .....    | 15 |
| 5.4.1    Mise en pratique du processus 2TUP.....    | 15 |
| 5.5    Diagramme de Gantt.....                      | 16 |
| Conclusion.....                                     | 17 |
| Chapitre II : Capture des besoins.....              | 18 |
| Introduction.....                                   | 18 |
| 1    Besoins fonctionnels .....                     | 18 |
| 2    Besoins non fonctionnels .....                 | 20 |
| 3    Besoins techniques .....                       | 21 |
| 3.1    Choix technologique .....                    | 22 |

|                                |                                                           |    |
|--------------------------------|-----------------------------------------------------------|----|
| 3.1.1                          | Front-End .....                                           | 22 |
| 3.1.1.1                        | React JS .....                                            | 22 |
| 3.1.1.2                        | Angular JS .....                                          | 23 |
| 3.1.1.3                        | Vue JS.....                                               | 24 |
| 3.1.1.4                        | Etude comparative et résumé.....                          | 25 |
| 3.1.2                          | Back-End .....                                            | 26 |
| 3.1.2.1                        | Serveur .....                                             | 26 |
| 3.1.2.2                        | Etude comparative entre les Framework Back-End .....      | 26 |
| 3.1.2.3                        | Base de données .....                                     | 26 |
| 3.1.2.4                        | Classement des bases de données .....                     | 28 |
| 3.1.3                          | Choix technologique de notre application.....             | 29 |
| 3.1.3.1                        | Redux.....                                                | 28 |
| 3.1.3.2                        | Knex.js.....                                              | 30 |
| 3.1.3.3                        | Objection.js.....                                         | 30 |
| 3.2                            | Choix du modèle conceptuel .....                          | 32 |
| 3.2.1                          | MVC .....                                                 | 32 |
| 3.2.2                          | MVVM.....                                                 | 33 |
| 3.3                            | Notre choix.....                                          | 34 |
| 3.3.1                          | Pour la couche sécurité du model MVC.....                 | 35 |
| Conclusion.....                |                                                           | 35 |
| Chapitre III : Conception..... |                                                           | 35 |
| Introduction.....              |                                                           | 36 |
| 1                              | Conception générique .....                                | 36 |
| 1.1                            | Le langage UML.....                                       | 36 |
| 2                              | Conception préliminaire des interfaces – Prototypes ..... | 36 |
| 3                              | Conception détaillée .....                                | 38 |
| 3.1                            | Les vues de Kruchten.....                                 | 38 |
| 3.2                            | Identification des cas d'utilisations.....                | 40 |
| 3.2.1                          | Cas d'utilisation général.....                            | 41 |
| 3.2.2                          | Description détaillée des cas d'utilisation.....          | 42 |
| 3.2.2.1                        | Cas d'utilisation associés à « Internaute ».....          | 42 |
| 3.2.2.2                        | Cas d'utilisation associés au « Prestataire ».....        | 43 |
| 3.2.2.3                        | Cas d'utilisation associés à « Administrateur ».....      | 44 |
| 3.2.2.4                        | Cas d'utilisation associés à l'« E-touriste » .....       | 44 |
| 3.3                            | Diagrammes de séquence.....                               | 45 |
| 3.3.1                          | Diagramme de séquence « S'inscrire ».....                 | 46 |
| 3.3.2                          | Diagramme de séquence « Authentification » .....          | 47 |
| 3.3.3                          | Diagramme de séquence « Publier annonces » .....          | 48 |

|                                |                                                                             |    |
|--------------------------------|-----------------------------------------------------------------------------|----|
| 3.3.4                          | Diagramme de séquence « Modifier annonces » .....                           | 49 |
| 3.3.5                          | Diagramme de séquence « Supprimer annonces » .....                          | 50 |
| 3.4                            | Diagrammes d'interaction .....                                              | 50 |
| 3.4.1                          | Diagramme d'interaction « S'authentifier ».....                             | 51 |
| 3.4.2                          | Diagramme d'interaction « Publier une annonce » .....                       | 52 |
| 3.5                            | Diagramme de classe.....                                                    | 55 |
| 3.6                            | Diagramme de composants .....                                               | 57 |
| Conclusion.....                |                                                                             | 57 |
| Chapitre IV : Réalisation..... |                                                                             | 58 |
| Introduction.....              |                                                                             | 58 |
| 1                              | Environnement de développement .....                                        | 58 |
| 1.1                            | Environnement matériel .....                                                | 58 |
| 1.2                            | Environnement logiciel .....                                                | 59 |
| 2                              | Présentation de la solution web.....                                        | 60 |
| 2.1                            | Interface d'accueil associé à un 'Prestataire'.....                         | 61 |
| 2.2                            | Interface de l'espace administrateur pour la gestion des utilisateurs ..... | 61 |
| 2.3                            | Interface d'inscription.....                                                | 62 |
| 2.4                            | Interface d'authentification.....                                           | 63 |
| 2.5                            | Interface d'accueil associé à l'abonné.....                                 | 64 |
| 2.6                            | Interface destination.....                                                  | 65 |
| Conclusion.....                |                                                                             | 65 |
| Bibliographie .....            |                                                                             | 67 |

# Liste des Figures

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Figure I.1 : Logo GPRO Consulting .....                                                  | 2  |
| Figure I.2 : Page d'accueil du site web « Discover Tunisia » <sup>(4)</sup> .....        | 4  |
| Figure I.3 : Page d'accueil du site web « TunisiaTourism » <sup>(5)</sup> .....          | 5  |
| Figure I.4 : Page d'accueil du site web « Destination Tunisie » <sup>(6)</sup> .....     | 6  |
| Figure I.5 : Page d'accueil du site web « Cityzeum Travel » <sup>(7)</sup> .....         | 7  |
| Figure I.6 : Page d'accueil du siteeb « Saudi eVisa » <sup>(8)</sup> .....               | 8  |
| Figure I.7 : Représentation graphique de la solution proposée .....                      | 10 |
| Figure I.8 : Les étapes du processus Agile <sup>(9)</sup> .....                          | 11 |
| Figure I.9 : Les étapes du processus Scrum.....                                          | 12 |
| Figure I.10 : Les étapes du processus Unified Process <sup>(10)</sup> .....              | 13 |
| Figure I.11 : Les étapes du processus 2TUP .....                                         | 15 |
| Figure I.12 : Diagramme de Gantt et planning prévisionnelle <sup>(11)</sup> .....        | 17 |
| Figure II.1 : Les technologies Front-End les plus populaires <sup>(12)</sup> .....       | 22 |
| Figure II.2 : Architecture d'une application utilisant React JS <sup>(13)</sup> .....    | 23 |
| Figure II.3 : Architecture d'une application utilisant Angular JS <sup>(14)</sup> .....  | 24 |
| Figure II.4 : Architecture d'une application utilisant Vue JS.....                       | 25 |
| Figure II.5 : Classement des bases de données <sup>(17)</sup> .....                      | 27 |
| Figure II.6 : Choix technologique de notre application.....                              | 29 |
| Figure II.7 : Cycle React JS - Redux.....                                                | 29 |
| Figure II.8 : Architecture MVC.....                                                      | 32 |
| Figure II.9 : Architecture MVVM.....                                                     | 33 |
| Figure II.10 : Exemple d'utilisation d'OAuth <sup>(18)</sup> .....                       | 34 |
| Figure III.1 : Maquette préliminaire de l'interface d'accueil.....                       | 36 |
| Figure III.2 : Maquette préliminaire de l'interface Inscription et Authentification..... | 37 |
| Figure III.3 : Maquette préliminaire de l'interface « Destination ».....                 | 38 |
| Figure III.4 : Vue 4+1 (Kruchten).....                                                   | 37 |
| Figure III.5 : Diagramme de cas d'utilisation général.....                               | 41 |
| Figure III.6 : Diagramme de cas d'utilisation associés à « Internaute ».....             | 43 |
| Figure III.7 : Diagramme de cas d'utilisation associés au « Prestataire » .....          | 43 |
| Figure III.8 : Diagramme de cas d'utilisation associés à « Administrateur » .....        | 44 |
| Figure III.9 : Diagramme de cas d'utilisation associés à « E-touriste ».....             | 45 |
| Figure III.10 : Diagramme de séquence « S'inscrire » .....                               | 46 |
| Figure III.11 : Diagramme de séquence « Authentification » .....                         | 47 |

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Figure III.12 : Diagramme de séquence « Publier annonces ».....                          | 47 |
| Figure III.13 : Diagramme de séquence « Modifier annonces ».....                         | 48 |
| Figure III.14 : Diagramme de séquence « Supprimer annonces ».....                        | 49 |
| Figure III.15 : Diagramme d'interaction « S'authentifier ».....                          | 50 |
| Figure III.16 : Diagramme d'interaction « Publier une annonce ».....                     | 54 |
| Figure III.17 : Diagramme de classe.....                                                 | 56 |
| Figure III.18 : Diagramme de composants.....                                             | 57 |
| Figure IV.1 : Interface d'accueil associé à l'administrateur.....                        | 59 |
| Figure IV.2 : Interface de l'espace administrateur pour la gestion des utilisateurs..... | 59 |
| Figure IV.3 : Interface d'inscription .....                                              | 60 |
| Figure IV.4 : Interface d'authentification.....                                          | 61 |
| Figure IV.5 : Interface d'accueil associé à l'abonné.....                                | 62 |
| Figure IV.6 : Interface destination.....                                                 | 63 |

# Liste des Tableaux

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Tableau I.1 : Identité de la société .....                                                | 2  |
| Tableau I.2 : Tableau comparatif des applications web existantes.....                     | 9  |
| Tableau I.3 : Tableau comparatif des méthodologies de développement .....                 | 14 |
| Tableau II.1: Besoins non fonctionnels.....                                               | 21 |
| Tableau II.2 : Tableau comparatif entre les différentes technologies Front-End.....       | 25 |
| Tableau II.3 : Tableau comparatif entre les différentes technologies Back-End.....        | 26 |
| Tableau II.4 : Base de données Relationnelles vs NoSQL.....                               | 27 |
| Tableau III.1 : Tableau des cas d'utilisation.....                                        | 41 |
| Tableau III.2 : Description textuelle du diagramme de séquence « S'inscrire ».....        | 45 |
| Tableau III.3 : Description textuelle du diagramme de séquence « Authentification ».....  | 46 |
| Tableau III.4 : Description textuelle du diagramme de séquence « Publier annonces ».....  | 47 |
| Tableau III.5 : Description textuelle du diagramme de séquence « Modifier annonces »..... | 48 |
| Tableau III.6 : Description textuelle du diagramme de séquence « Supprimer annonces »     | 49 |
| Tableau IV.1: Environnement Logiciel.....                                                 | 59 |

# Liste des abréviations

**JEE :** Java Entreprise Edition

**MVC :** Modèle Vue Contrôleur

**JS :** JavaScript

**ERP:** Enterprise Resource Planning

**UML:** Unified Modeling Language

**SPA:** Single Page Application

**HTML:** Hypertext Markup Language

**DOM:** Document Object Model

**CSS:** Cascading Style Sheets

**RXJS:** Reactive Extensions for JavaScript

**PHP:** Hypertext Preprocessor

**SGBDR :** Système de gestion de base de données relationnel

**PSD :** Photoshop Document

**IHM :** Interface Homme Machine

# Introduction Générale

---

Le tourisme est clairement l'un des phénomènes économiques et sociaux qui demeurent le plus remarquable lors du 21<sup>e</sup> siècle. Dans la plupart des pays, le tourisme est désormais l'un des secteurs de l'économie les plus fulgurant et dont la progression est la plus rapide. Le tourisme tunisien reste l'un des secteurs les plus dynamiques de notre économie et la principale source de devises pour le pays. Mais hélas, la Tunisie a beaucoup de mal à rejoindre le giron des décideurs touristiques mondiaux qui, plus que jamais, sont au centre des enjeux et défis technologiques. En conséquence, divers sites Internet ont été créés, dont le but est d'obtenir une meilleure rentabilité afin d'améliorer le secteur du tourisme en particulier après l'apparition du coronavirus. Cette pandémie de covid-19 a touché de plein fouet le secteur touristique et il est impératif de commencer dès à présent à préparer la reprise du tourisme post-pandémie, car tôt ou tard la vie reprendra ses droits, avec un regain d'intérêt pour les voyages et les vacances. En attendant la réduction des pertes causées par ce tsunami viral [\(1\)](#).

C'est dans ce contexte que se déroule notre projet de fin d'étude. En effet, nous avons été chargés du développement d'un portail web pour le secteur du tourisme appelé **We Travel**. Notre application web représente un moyen simplifié pour les e-touristes qui cherchent la facilité de trouver un logement, des voyages organisés en Tunisie, et des restaurants et en prenant les précautions appropriées contre cette épidémie. Le présent rapport présente les différentes étapes de la réalisation de notre projet et s'étale sur six chapitres :

- ❖ Le premier chapitre, intitulé « Contexte et objectifs du projet » présente le contexte général du projet, une étude rapide des solutions existantes ainsi que la présentation de la solution proposée. La méthodologie de travail est également discutée.

- ❖ Le deuxième chapitre, intitulé « Capture des besoins », est consacré à la spécification des besoins fonctionnels et non-fonctionnels. Ainsi que le pilotage du projet par le processus de développement 2TUP.
- ❖ Le troisième chapitre, intitulé « Conception », expose l'architecture générale de notre application web et présente l'environnement de conception.

# CHAPITRE I : CONTEXTE ET OBJECTIFS DU PROJET

## Introduction

Dans ce chapitre introductif, la première partie sera destinée à la présentation de l'organisme d'accueil « GPRO Consulting » en mentionnant leurs spécialités ainsi que leurs qualités. Quant à la deuxième partie, elle sera consacrée à la présentation de l'idée générale de notre projet et la problématique qui a poussé l'organisme à réaliser cette application. Par la suite, nous abordons l'étude des solutions existantes avec leurs critiques et nous introduisons la solution proposée. Enfin, nous terminons ce chapitre par une analyse des différentes méthodologies de développement et nous faisons une comparaison pour choisir la méthode la plus adéquate pour la réalisation de notre application web.

## 1 Contexte général

Cette partie présente le cadre général du projet ainsi que ses principaux objectifs.

### 1.1 Cadre du projet

Le présent travail s'inscrit dans le cadre du projet de fin d'étude en vue de l'obtention du Diplôme de Licence Fondamentale en Informatique délivré par l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM). Notre stage s'est déroulé au sein de la société " GPRO Consulting " pour une durée de trois mois du 01 mars au 31 mai 2020.

### 1.2 Présentation de l'entreprise d'accueil

GPRO Consulting, voir **Figure I.1**, est basée à Sousse et comporte une équipe dynamique, des consultants et un large réseau de partenaires permettant d'offrir un service informatique global dans différents domaines en utilisant la majorité des technologies connues du marché. Elle a pour métier un service informatique de qualité auprès des PME/PMI, à très forte valeur ajoutée misant sur l'innovation et l'expertise. La société qui s'engage à garantir la satisfaction de ses clients.

| Raison sociale | GPRO Consulting                                                                    |
|----------------|------------------------------------------------------------------------------------|
| Nom du gérant  | Ghazi Atroussi                                                                     |
| Adresse        | Avenue Mohamed Attia Khzema<br>Ouest – Sousse                                      |
| Téléphone      | 58 100 435 / 73 276 175                                                            |
| Email          | <a href="mailto:contact@gproconsulting.com">contact@gproconsulting.com</a>         |
| Site web       | <a href="http://www.gproconsulting.com">www.gproconsulting.com</a>                 |
| Compétences    | Java JEE - Spring MVC Framework -<br>AngularJS -Intégration ERP-<br>Android et IOS |

Tableau I.1 : Identité de la société

L'entreprise fournit des solutions dans des domaines diversifiés liés aux :

- ❖ Système ERP
- ❖ Solutions logicielles de gestion industrielle et manufacturière
- ❖ Assistance technique dans les différents processus industriels
- ❖ Etude et consulting dans la mise en place des systèmes d'information
- ❖ Expertise en termes de préparation des certifications
- ❖ Infogérance
- ❖ Développement spécifique adapté au besoin des clients
- ❖ Outsourcing pour le marché européen notamment le marché Français



Figure I.1 : Logo GPRO Consulting

### 1.3 Idée maitresse

Le tourisme est une activité importante dans le monde actuel <sup>(2)</sup>. C'est un domaine en constante expansion qui couvre une grande partie des activités du secteur des services comme l'hôtellerie, la restauration et les agences de voyage.

Dans ce cadre, nous avons été chargés de concevoir et développer un site internet présentant un large choix d'hébergements, des voyages en Tunisie bien illustrés et documentés, des restaurants, des activités et qui permet d'améliorer le processus de recherche aux utilisateurs.

Ainsi pour découvrir des centaines d'offres pour réserver des vacances en ligne ou demander des devis.

## 2 Problématique

Le tourisme est le secteur où la révolution du web est la plus fulgurante. Certains sites dédiés au tourisme coexistent sur la toile : tous les opérateurs, tour-opérateurs, compagnies aériennes, groupes hôtelleries mais aussi les offres touristiques et les sites municipaux.

Les E-touristes sont de vrais spécialistes des outils internet, ils ont la possibilité via des sites internet existants de louer une chambre d'hôtel, de la visiter virtuellement et de réserver dans des restaurants etc...

Cependant selon les recherches, on constate que les sites ne sont pas trop satisfaisants et suffisants pour les E-touristes en particulier pour le marché tunisien qui a reçu plus de 9.429.049 visiteurs en 2019 <sup>(3)</sup> :

- ❖ L'absence d'un forum pour donner les avis, échanger des idées ou faire des commentaires
- ❖ Le temps de réponse des sites est très long
- ❖ Le manque d'espace pour les clients pour s'inscrire et recevoir des notifications d'activités sera fait, des promotions sur les hôtels etc...
- ❖ Manque de statistiques sur les endroits les plus visités et les plus populaires de chaque ville

## 3 Etude de l'existant

L'étude de l'existant constitue une étape préliminaire pour la réalisation d'un projet. Dans cette section, nous menons une analyse détaillée de l'existant, suivie d'une critique et une étude comparative des applications web existent déjà afin de dégager les points faibles et les points forts et les besoins des utilisateurs pour les prendre en considération lors de la

conception et la réalisation. Nous proposons alors quelques solutions en Tunisie et à l'étranger qui offrent le même concept que notre portail web.

### 3.1 Solutions Existantes

#### 3.1.1 Solutions en Tunisie

- **Application 1:** Discover Tunisia

Lancé récemment par l'Office National du Tourisme Tunisien, Discover Tunisia, est un portail électronique dédié à la promotion des destinations touristique en Tunisie, voir **Figure I.2**. Ce portail vous fait découvrir les différents détails sur les prestations touristiques proposées par la Tunisie. Il fait également la sélection des modes d'hébergement et propose les meilleurs hôtels, clubs, maisons d'hôtes, gîtes ruraux et hôtels de charme.

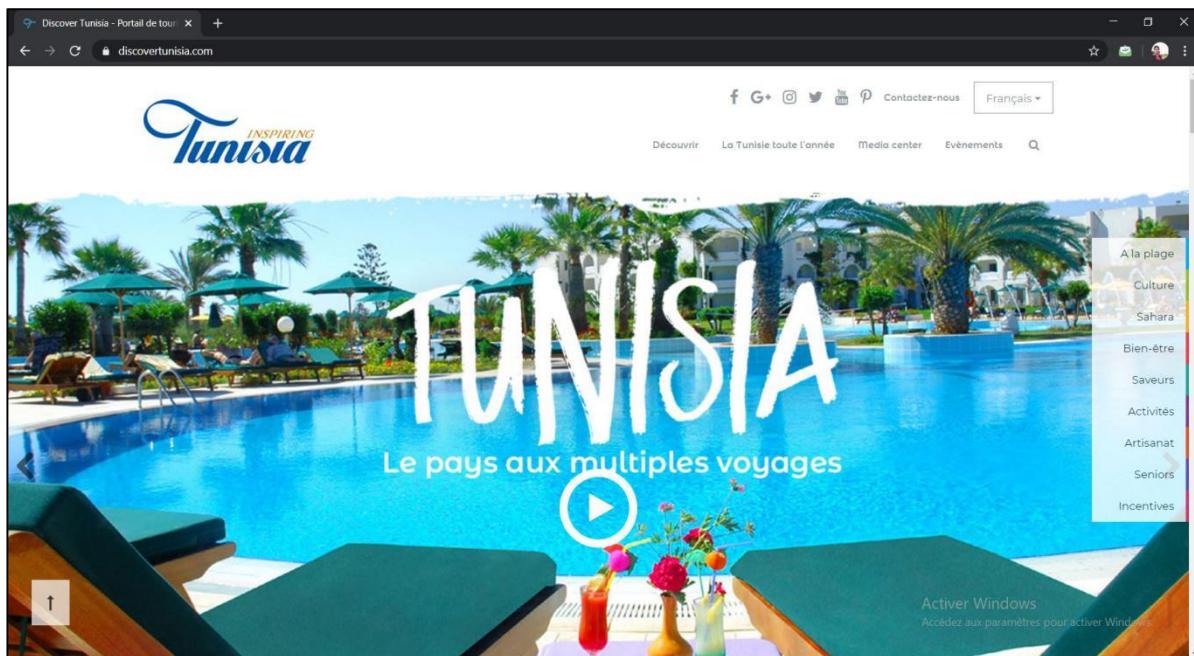


Figure I.2 : Page d'accueil du site web « Discover Tunisia »<sup>(4)</sup>

Les points forts :

- ✓ Disponible en six langues dont le chinois et l'allemand
- ✓ Intègre une carte interactive

Les points faibles :

- ✗ Le portail n'offre pas la possibilité de s'inscrire
- ✗ Ne dispose pas d'un outil de réservation en ligne

- **Application 2 :** TunisiaTourism.info

TunisiaTourism.info, **voir Figure I.3**, est un portail dédié à la promotion du tourisme en Tunisie. Ce nouveau site est une initiative de la société d'édition et de communication MCM (Media, Communication, Marketing), société d'édition spécialisée dans le tourisme, en partenariat avec la compagnie aérienne Tunisair et le ministère de tourisme ainsi que des professionnels du tourisme tunisien. Il s'agit d'un portail destiné à promouvoir le tourisme tunisien auprès des touristes locaux et étrangers .

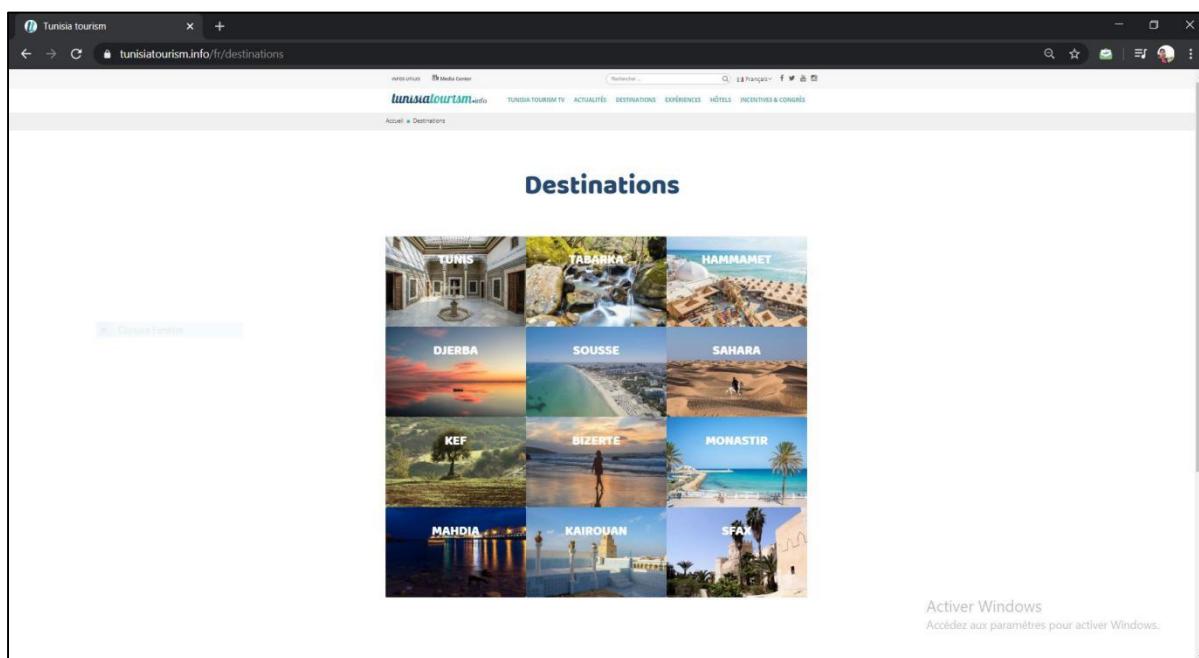


Figure I.3 : Page d'accueil du site web « TunisiaTourism »<sup>(5)</sup>

Les points forts :

- ✓ Propose la première chaîne télévisée numérique spécialisée en tourisme Tunisien
- ✓ Conçu en quatre langues
- ✓ Intègre une carte interactive

Les points faibles :

- ✗ Le portail ne dispose pas d'un espace client.
- ✗ L'indisponibilité d'un système de notification

- **Application 3 :** Destinationtunisie.info

Destinationtunisie.info, **voir Figure I.4**, est un site web développé par une équipe de journalistes professionnels spécialisés notamment dans le tourisme et le transport aérien. Destinationtunisie.info a pour objectif d'informer les professionnels du secteur mais aussi le grand-public des événements qui font le tourisme tunisien au quotidien, y compris celui des compagnies aériennes, du secteur de la culture .

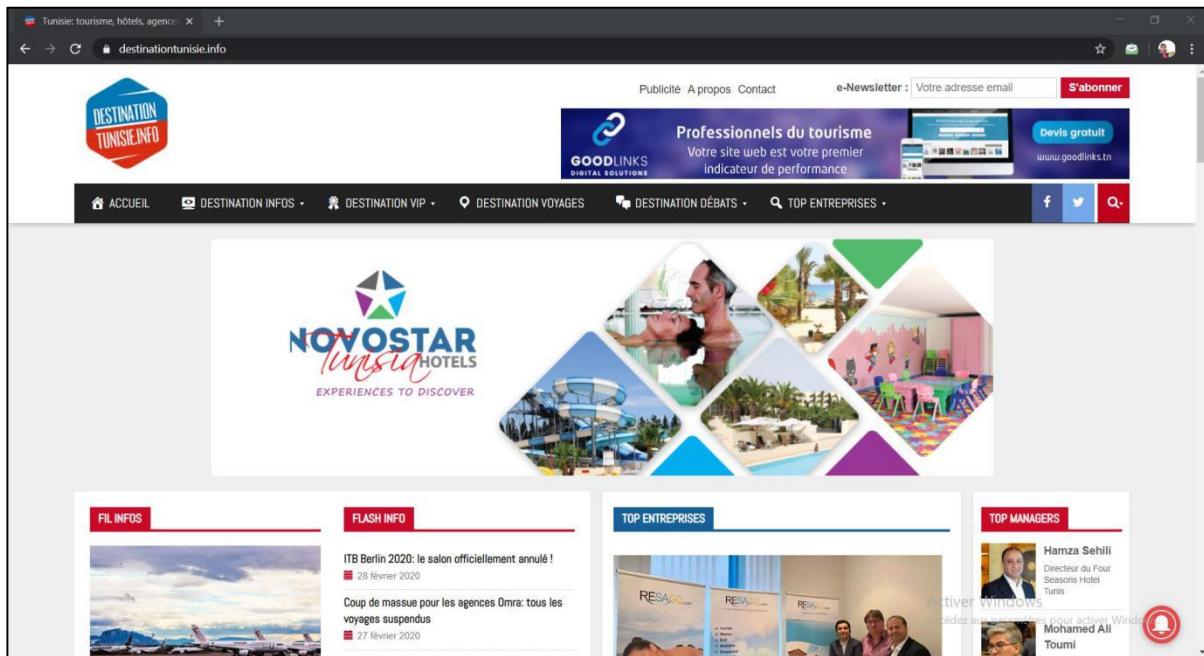


Figure I.4 : Page d'accueil du site web « Destination Tunisie »<sup>(6)</sup>

Les points forts :

- ✓ Propose les meilleures destinations (que ce soit Hôtels, restaurants,...)

Les points faibles :

- ✗ Absence d'un espace client
- ✗ Une ergonomie médiocre incapable de s'adapter au niveau de l'utilisateur

- **Application 4 :** Cityzeum Travel

Cityzeum Travel, voir **Figure I.5**, est une start-up cofondée par des voyageurs et experts du marketing digital .

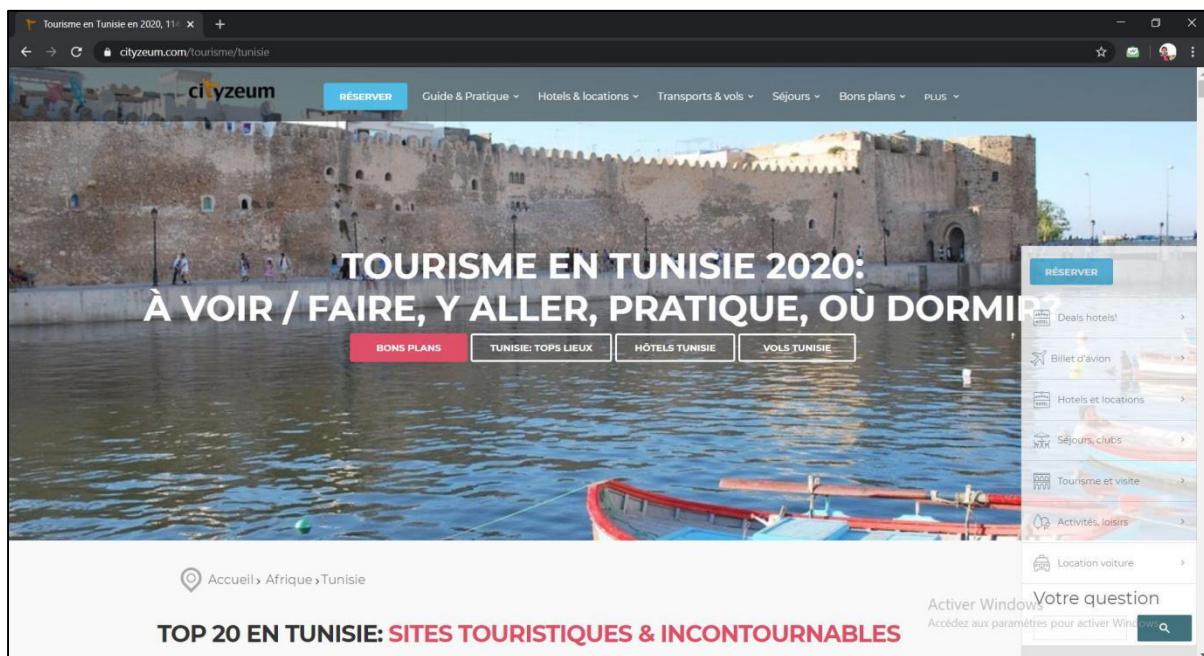


Figure I.5 : Page d'accueil du site web « Cityzeum Travel »<sup>(7)</sup>

Les points forts :

- ✓ Des contenus numériques plus riches
- ✓ Des services touristiques qui dépassent le cadre du guide de voyage numérique
- ✓ La géolocalisation
- ✓ Existence d'un système de réservation

Les points faibles :

- ✗ Ce portail n'offre pas la possibilité de s'inscrire
- ✗ L'indisponibilité d'un système de notification

### 3.1.2 Solutions à l'étranger

- **Application 1 :** Saudi eVisa

La page officielle du tourisme de l'Arabie Saoudite, voir **Figure I.6**. Il fournit aux internautes sur un même portail l'ensemble des informations dont ils ont besoin.

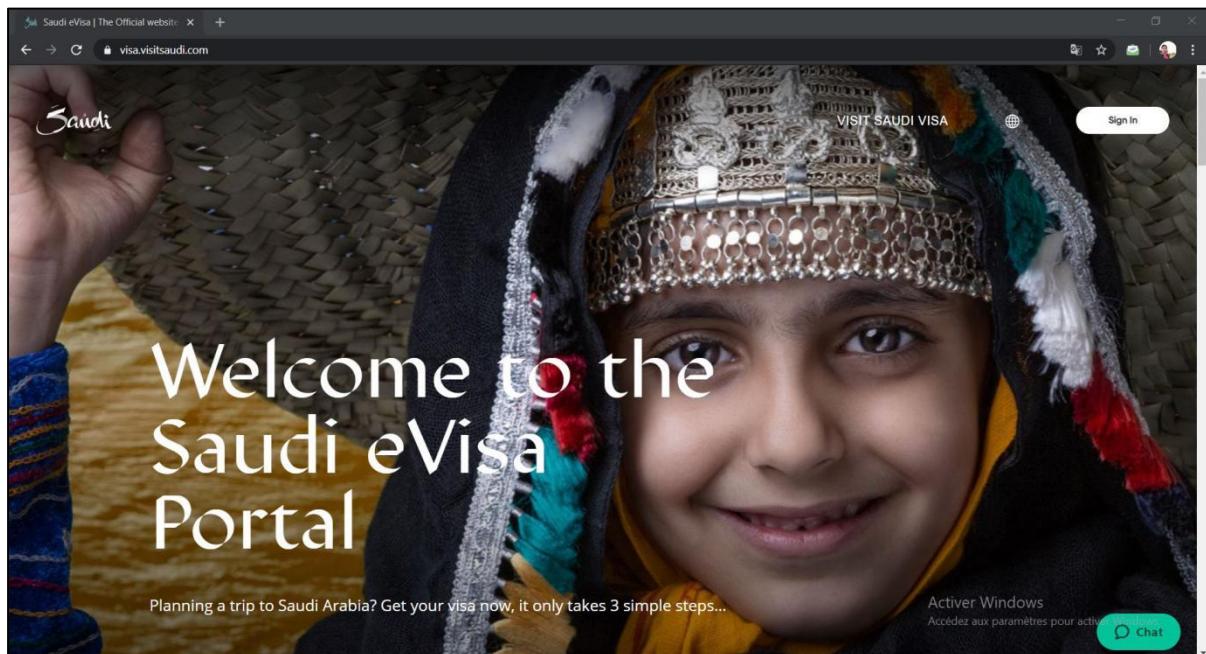


Figure I.6 : Page d'accueil du site web « Saudi eVisa »<sup>(8)</sup>

Les points forts :

- ✓ Très dynamique
- ✓ Innovant
- ✓ Propose les processus de demande et les conditions d'éligibilité pour obtenir le Visa de l'Arabie Saoudite

Les points faibles :

- ✗ Fournit uniquement pour la Royaume d'Arabie Saoudite

### 3.2 Critique de l'existant

Nous pouvons classer le résultat de l'analyse des applications web existantes mentionnées précédemment selon huit critères [Cx] pris en considération dans le processus d'évaluation de ses applications :

**[C1] Interface utilisateur et Attirance :** La qualité du contenu et l'exhaustivité. L'application est-elle facile à utiliser et est-elle esthétiquement attrayante ?

**[C2] Annuaire des prestataires de domaine :** Proposer des renseignements pour chaque prestataire.

**[C3] Espace Clients :** La possibilité de s'inscrire.

**[C4] Existence d'un Tableau de Bord :** Permettant d'analyser les données du portail et fournir des statistiques (Les lieux les plus visités,...)

**[C5] Forum de discussion :** Donner des avis, Echanger des idées, Commenter...

**[C6] Géolocalisation :** Disponibilité d'une carte de guidage.

**[C7] Notifications :** La disponibilité d'un système de notifications.

**[C8] Temps de réponse :** La rapidité de chargement des informations.

|                         | [C1]        | [C2] | [C3] | [C4] | [C5] | [C6] | [C7] | [C8]        |
|-------------------------|-------------|------|------|------|------|------|------|-------------|
| Discover Tunisia        | Moyenne<br> | Oui  | Non  | Non  | Non  | Oui  | Non  | Rapide<br>  |
| TunisiaTourism.info     | Bien<br>    | Oui  | Non  | Non  | Non  | Oui  | Non  | Rapide<br>  |
| Destinationtunisie.info | Faible<br>  | Non  | Non  | Non  | Non  | Oui  | Non  | Lent<br>    |
| Cityzeum Travel         | Moyenne<br> | Oui  | Non  | Non  | Non  | Oui  | Non  | Moyenne<br> |
| Saudi eVisa             | Bien<br>    | Oui  | Oui  | Non  | Non  | Oui  | Non  | Rapide<br>  |

Tableau I.2 : Tableau comparatif des applications web existantes

## 4 Solution proposée

### 4.1 Description textuelle

Tenant compte des critiques de l'existant et des résultats précédemment mentionnés, il s'avère qu'aucune solution existante ne répond convenablement à notre problématique. Ainsi, nous sommes amenés à proposer une solution digne de confiance et qui pourra répondre à nos besoins, nos objectifs. La nouvelle solution doit pallier les lacunes constatées au niveau des solutions développées. Dans ce contexte, nous avons décidé de concevoir et réaliser notre propre portail web qui soutient le système d'information du secteur touristique assurant les fonctionnalités suivantes :

1. Un annuaire des prestataires de domaine
2. Un espace client
3. Forum
4. Evaluation des prestataires par les clients
5. Système de notifications
6. Système de réservation
7. Un tableau de bord permettant l'analyse des données

## 4.2 Description visuelle

Le pictogramme ci-dessous (**Figure I.7**) illustre bien la solution proposée :

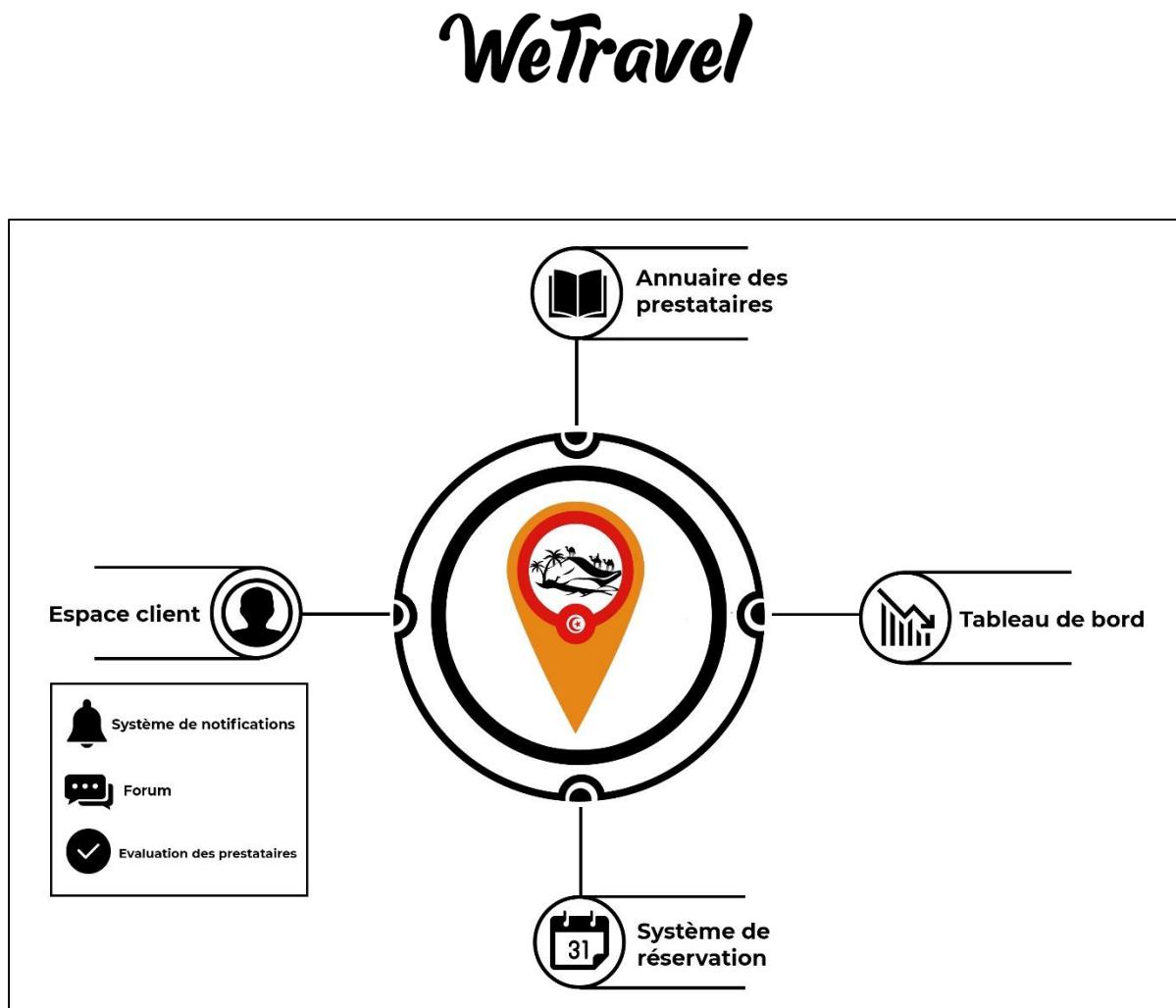


Figure I.7 : Représentation graphique de la solution proposée

## 5 Méthodologie de développement

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. C'est le fait de modéliser un système avant sa réalisation pour bien comprendre son fonctionnement et assurer sa cohérence. Un modèle est ainsi un facteur de réduction des coûts et des délais. Il est donc indispensable pour assurer un bon niveau de qualité de produit dont la maintenance est efficace.

## 5.1 Étude de Méthodologie de développement

Devant le nombre de méthodes disponible, le choix parmi elles devient difficile. Nous étudions à ce propos quelques méthodes de développement objet. Suite à cette étude, nous choisissons la méthodologie la plus correspondante.

## 5.2 Présentation de quelques méthodologies

### 5.2.1 Agile

Agile est une méthodologie qui anticipe le besoin de flexibilité et applique un niveau de pragmatisme dans la livraison du produit fini. Agile nécessite un changement de culture dans de nombreuses entreprises, il se concentre sur la livraison propre de pièces individuelles ou de parties du logiciel et non sur l'ensemble de l'application, voir **Figure I.8**.

Les méthodes agiles prônent 4 valeurs fondamentales :

- ❖ Individus et Interactions sur les processus et les outils
- ❖ Logiciel de travail sur une documentation complète
- ❖ Collaboration des clients à la négociation de contrats
- ❖ Répondre au changement au sujet d'un plan

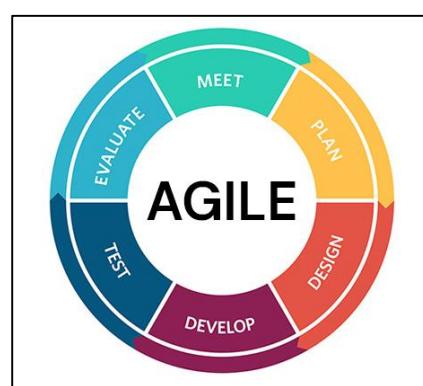


Figure I.8 : Les étapes du processus Agile <sup>(9)</sup>

### 5.2.2 Scrum

Dérivé d'une méthode de gestion de projets agile, la méthode SCRUM définit un cadre de travail permettant la réalisation des projets complexes. Initialement prévu pour le développement de projets type « Software », cette méthode peut être appliquée à tout type de projet, du plus simple au plus

innovant, et d'une manière très simple. Cette méthodologie permet de s'adapter rapidement aux changements d'un client à fréquence régulière. Chaque fin d'itération, l'équipe et le client réévaluent les spécifications du logiciel.

Le cycle de vie d'un projet Scrum, exprimé dans la **Figure I.9**, est divisé en trois parties :

**Phase d'initialisation** : phase linéaire aux processus explicitement connus et dont les inputs et les outputs sont bien déterminés.

**'Sprint' de développement** : est un processus empirique.

**Phase de clôture** : est aussi une phase linéaire dont les processus sont clairement définis.

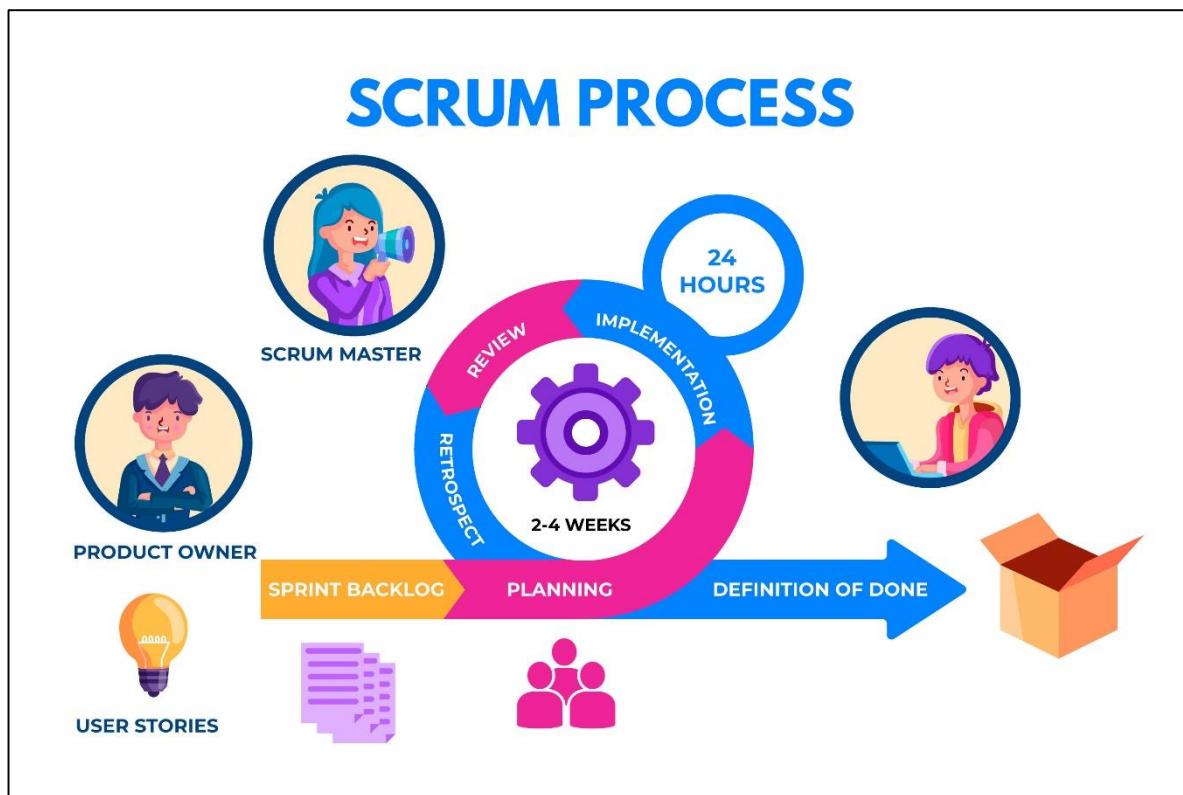


Figure I.9 : Les étapes du processus Scrum

### 5.2.3 Unified Process

Unified Process, voir **Figure I.10**, est un processus de développement centré sur l'architecture, basé sur des cas d'utilisation, itératif et incrémental qui utilise le langage de modélisation uniifié (UML). Unified Process peut être appliqué à différents systèmes logiciels avec différents niveaux de complexité technique et de gestion dans différents domaines et cultures organisationnelles.

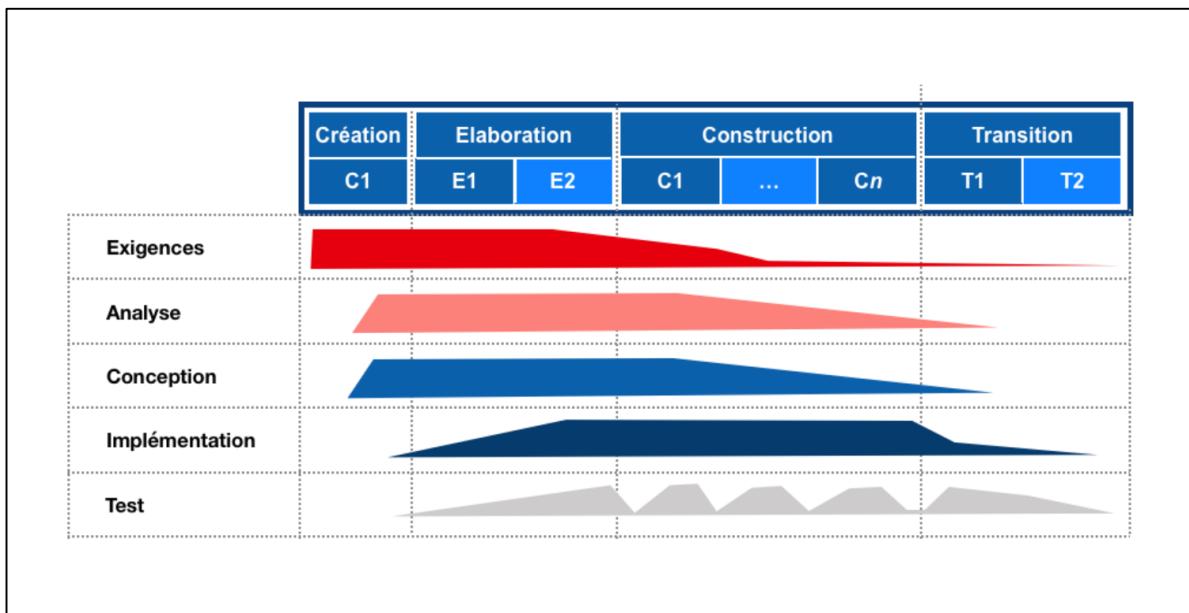


Figure I.10 : Les étapes du processus Unified Process <sup>(10)</sup>

### 5.2.3.1 Two Track Unified Process (2TUP)

2 Tracks Unified Process (2TUP) est un processus uniifié. Le principe de 2TUP est que toute évolution imposée à un logiciel peut être décomposée et traitée en parallèle, selon un axe fonctionnel et un axe technique. Et la réalisation du logiciel consiste à fusionner les résultats de ces deux branches du processus.

### 5.2.3.2 Rational Unified Process (RUP)

C'est un processus de développement de logiciel de Rational, une division d'IBM. Il divise le processus de développement en quatre phases :

- ❖ **Inception** : l'idée de ce projet est déclarée. L'équipe de développement détermine si le projet mérite d'être poursuivi et quelles ressources seront nécessaires.
- ❖ **Elaboration** : l'architecture du projet et les ressources nécessaires sont encore évaluées. Les développeurs considèrent les applications possibles du logiciel et les coûts associés au développement.
- ❖ **Construction** : le projet est développé et complété. Le logiciel est conçu, écrit et testé.
- ❖ **Transition** : le logiciel est diffusé au public. Les derniers ajustements ou mise à jour sont effectués en fonction des commentaires finaux des utilisateurs.

### 5.3 Étude comparative et évaluation

| Méthodologies          | Avantages                                                                                                                                                                                                                                                                                         | Inconvénients                                                                                                                                                                                                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Agile</b>           | Agile offre de multiples opportunités pour l'engagement des parties prenantes et des équipes. En impliquant le client dans chaque étape du projet, il y a un haut degré de collaboration entre le client et l'équipe du projet, ce qui permet à l'équipe de mieux comprendre la vision du client. | - Il a moins bien réussi à intégrer les tests et les opérations à ce mélange.<br>- Manque d'emphase sur la technologie, ce qui peut rendre difficile la vente du concept aux cadres supérieurs qui ne comprennent pas le rôle que la culture joue dans le développement de logiciels. |
| <b>Scrum</b>           | - Le client est au cœur du Projet.<br>- Esprit d'équipe.<br>- Communication lâchée.<br>- Simplicité, efficacité et qualité.<br>- Flexibilité au changement.<br>- Avancement basé sur le concret.                                                                                                  | - Manque de pratique pour obtenir le 'DONE' dans le contexte logiciel.                                                                                                                                                                                                                |
| <b>Unified Process</b> | - Itératif et incrémental.<br>- Focalisé sur les risques.<br>- Des approches éprouvées pour développer et maintenir des logiciels de qualité.<br>- Meilleur niveau de portabilité                                                                                                                 | - Nécessite des experts<br>- Coûteux à personnaliser<br>- Très axé processus                                                                                                                                                                                                          |
| <b>RUP</b>             | - Traçabilité à partir des cas d'utilisation jusqu'au déploiement.<br>- Approche basée sur l'architecture.<br>- Gestion des risques dans les Projets.<br>- Cadre propice à la réutilisation.                                                                                                      | - Coût de personnalisation souvent élevé.<br>- Processus très axé.<br>- Vision non évidente ni immédiate.                                                                                                                                                                             |
| <b>2TUP</b>            | - Convenable pour tout type de projet.<br>- Permet de répondre aux besoins des utilisateurs rapidement.<br>- Canalise et modélise toutes les étapes du développement d'un logiciel.<br>- Produit un modèle focalisé sur le métier des utilisateurs finaux.                                        | Pas de documents types.                                                                                                                                                                                                                                                               |

Tableau I.3 : Tableau comparatif des méthodologies de développement

## 5.4 Choix de la méthodologie de travail

Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final. C'est pour cela qu'on a besoin d'un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels tout en commençant par une étude préliminaire. Ainsi, suite à une étude comparative, notre choix s'est alors porté vers la méthode 2TUP vu qu'elle est caractérisée par une approche nouvelle et originale et qu'elle respecte le cadre de notre projet.

### 5.4.1 Mise en pratique du processus 2TUP

2TUP est un processus uniifié (c'est-à-dire construit sur UML, itératif, centré sur l'architecture conduit par les cas d'utilisation). **La Figure I.11** suivante montre bien les étapes du processus 2TUP.

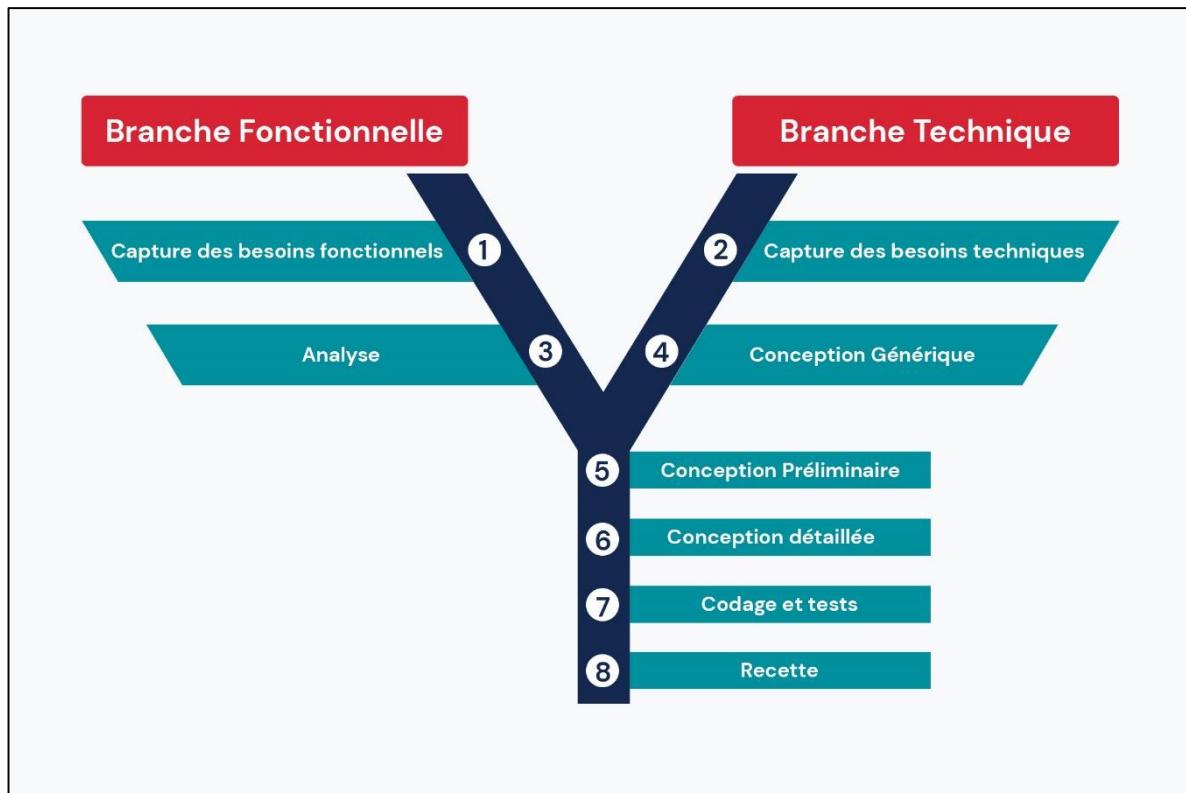


Figure I.11 : Les étapes du processus 2TUP

Le principe fondateur du 2TUP est que toute évolution imposée à un logiciel peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique.

La réalisation du logiciel consiste à fusionner les résultats de ces deux branches du processus. Selon le processus 2TUP que nous suivons, nous passons obligatoirement par les phases suivantes à la suite de notre travail :

**1. L'étude préliminaire :** qui contient une description du service et les cas d'utilisation principaux, c'est une première version de la spécification générale.

**2. La capture des besoins fonctionnels :** qui définit le quoi faire à travers une spécification générale qui décrit le service à développer d'un point de vue fonctionnel et une spécification détaillée qui précise les traitements qui concernent chaque scénario des cas d'utilisation présent en spécification générale tout en respectant les contraintes fonctionnelles et non fonctionnelles.

**3. L'analyse :** où on effectue simultanément l'étude des données et l'étude des traitements à effectuer.

**4. La capture des besoins techniques :** qui permettent de satisfaire les contraintes techniques présentes dans le cahier des charges et donc répondre aux attentes de client.

**5. La conception générique :** qui définit le comment faire.

**6. La conception détaillée :** qui précise l'implémentation technique de l'application. Elle consiste en la fusion de la spécification détaillée et de la conception générique, pour déterminer comment faire le quoi faire dans le détail.

On y trouve le schéma de base de données, les diagrammes de classes et les diagrammes de séquence supplémentaires qui détaillent les interactions entre les composants du logiciel.

**7. Le Codage et les Tests :** qui décrivent comment est l'application est réalisée et installée sur un environnement d'exécution et comment la stratégie de validation est effectuée.

## 5.5 Diagramme de Gantt

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en tête représente les unités de temps les plus adaptées

au projet (jours, semaines, mois etc.). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentant la date de début, la durée et la date de fin. La **Figure I.12** ci-dessous permet donc de visualiser d'un seul coup d'œil :

- Les différentes tâches à envisager
- La date de début et la date de fin de chaque tâche
- La durée escomptée de chaque tâche

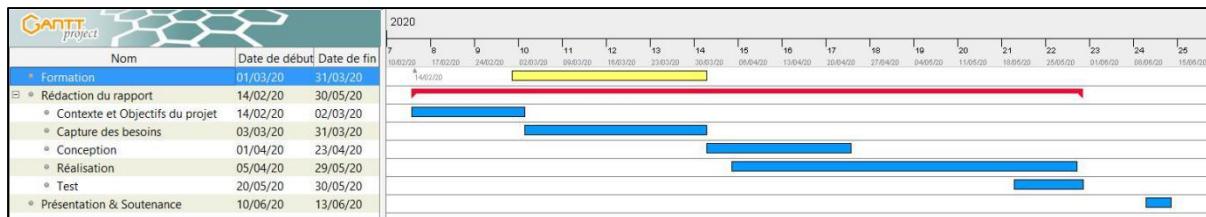


Figure I.12 : Diagramme de Gantt et planning prévisionnelle <sup>(11)</sup>

## Conclusion

Dans ce premier chapitre, nous avons fait une étude de l'existant nous permettant de bien entamer notre projet avec une illustration de la solution souhaitée. Nous avons comparé les différentes méthodes de développement afin de clôturer ce chapitre par une présentation de la méthodologie 2TUP que nous allons adopter pour le développement de notre projet. Le chapitre suivant sera consacré à la capture des besoins.

# CHAPITRE II : CAPTURE DES BESOINS

## Introduction

Le choix et la mise en pratique de la méthodologie 2TUP nous mène au présent chapitre qui sera consacré à l'identification de toutes les fonctionnalités de notre futur système et ceci en recensant les besoins fonctionnels et appréhendant la liste des exigences traduites par les besoins non fonctionnels. Les besoins techniques doivent également être recensés pour garantir une description sans ambiguïté du portail à développer.

### 1 Besoins fonctionnels

Tout système interactif, doit assurer et faciliter l'interaction entre les utilisateurs « acteurs » et le « système ». Un acteur représente le rôle d'une entité externe exploitant le système à travers ses différentes interfaces. Pour notre système, nous avons identifié les acteurs suivants :

- **Internaute :** C'est un visiteur de site, il peut consulter les annonces sur la page d'accueil, comme il peut la rechercher. Il a aussi le droit de créer un profil et devenir ainsi un abonné au site.
- **E-Touriste (Abonné) :** C'est un internaute qui possède un profil dans le site. Il peut commenter, réserver et suggérer des annonces.
- **Administrateur :** C'est un abonné au site qui peut accéder à l'espace d'administration du site. Il peut traiter les réclamations des abonnés, les bloquer et vérifier les annonces des prestations avant de les publier.
- **Prestataire :** C'est un internaute qui possède un profil dans le site. Il peut publier, modifier et supprimer des annonces.

Dans la suite, nous allons exposer les besoins fonctionnels de notre application en étroite relation avec les acteurs précédemment mentionnés.

Nous rappelons que ces besoins doivent répondre aux exigences du futur système.

L'internaute (Visiteur) détient le droit de :

- **Consulter le site** : L'internaute peut consulter les offres de plusieurs activités comme l'hôtellerie, la restauration et les agences de voyage,....

L'abonné (E-touriste) détient le droit de :

- **Commenter un service** : L'abonné peut donner son avis sur les services publiés dans un Forum.
- **Consulter les notifications** : L'abonné est notifié à propos des nouvelles suggestions ou commentaires sur ses annonces.
- **Gérer les Favoris** : L'abonné peut enregistrer des services qui l'intéressent dans ses favoris ou les supprimer.
- **Réserver** : L'abonné peut réserver dans un hôtel ou dans des restaurants, musées, événements et agences de voyage.
- **Evaluer un service** : Les E-touristes peuvent noter les services existants afin de les évaluer.
- **Envoyer une réclamation** : L'abonné peut envoyer une réclamation à l'administrateur via un formulaire dans lequel il explique sa demande.

L'administrateur détient le droit de :

- **Gérer les annonces publiées** : L'administrateur vérifie toutes les annonces publiées. Il peut soit les valider, soit les supprimer.
- **Gérer les comptes des abonnés** : L'administrateur a un accès à la liste des abonnés du site. Il a le droit de les supprimer ou les bloquer.
- **Traiter les réclamations** : L'administrateur consulte les réclamations des abonnés. Il peut supprimer l'annonce ou bien bloquer l'abonné qui a lancé la signalisation.
- **Valider les comptes des prestataires** : L'administrateur doit valider les comptes des prestataires.

Le prestataire détient le droit de :

- **Gérer les annonces** : Le prestataire peut consulter, publier, modifier ou supprimer une annonce.
- **Gérer boîte messagerie** : Le prestataire peut échanger des messages avec d'autres prestataires.

Il est important de signaler qu'il existe de nombreux besoins fonctionnels communs entre les acteurs. Dans ce qui suit, nous allons les détailler :

Les fonctionnalités communes entre le « Internaute » et « E-touriste » :

- **Rechercher un service** : Les 2 acteurs peuvent rechercher des services selon ses choix.

Les fonctionnalités communes entre le « Internaute » et le « Prestataire » :

- **S'inscrire** : Ces 2 acteurs peuvent créer un profil dans le site pour devenir un abonné.

Les fonctionnalités communes entre le « E-touriste », « Administrateur » et le « Prestataire » :

- **S'authentifier** : Ils doivent saisir leur login et leur mot de passe pour accéder chacun à son espace du site.

## 2 Besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les conditions requises permettant d'assurer le bon fonctionnement du système et d'améliorer la qualité des services pour l'utilisateur. Pour notre site web, on a défini les besoins non fonctionnels suivants :

| Besoins non fonctionnels                                                                                                 | Description                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <b>Confidentialité</b>                 | L'authentification est obligatoire afin de bénéficier des services disponibles.                                                                                                                                                                                                                                                     |
|  <b>Ergonomie</b>                       | Tous les standards d'ergonomie doivent être respectés dont la convivialité et la compréhensibilité des interfaces graphiques. De plus, le choix des couleurs, la densité et l'organisation des éléments sur l'écran et aussi l'utilisation des messages informatifs et des messages d'erreurs bien formées et bien lisibles.        |
|  <b>Disponibilité et Maintenabilité</b> | Le site web doit être accessible à tout moment et devra être extensible, c'est-à-dire facilement maintenable et s'adapte aux nouvelles exigences en cas de modification ou d'ajout d'une fonctionnalité.                                                                                                                            |
|  <b>Performance</b>                   | Nous souhaitons réaliser une application web mono page (en anglais Single Page Application ou SPA) qui optimise le temps de chargement de l'application et évite le rechargement de page à chaque action demandée par l'utilisateur, il en résulte une expérience utilisateur fluide en ce qui concerne la navigation dans le site. |
|  <b>Sécurité</b>                      | Les contraintes sécuritaires à prendre en compte lors de la réalisation de notre application sont l'attribution des rôles à chaque groupe d'utilisateurs pour s'assurer que l'accès aux différents espaces est protégé.                                                                                                             |
|  <b>Simplicité</b>                    | Notre application doit être simple à utiliser et ne nécessite pas des connaissances poussées puisqu'elle s'adresse à un public de différent âge incluant des non informaticiens.                                                                                                                                                    |

Tableau II.1 : Besoins non fonctionnels

### 3 Besoins techniques

Nous nous intéressons ici à la branche droite du cycle en Y « Capture des besoins techniques » qui capitalise le savoir-faire technique. Afin de bien expliquer nos choix technologiques, nous avons fait recours à une étude

comparative entre les différentes technologies qui peuvent être utilisés durant notre projet.

### 3.1 Choix technologique

Dans la partie suivante nous présentons les Framework Front-End et Back-End que nous allons utiliser pour le développement de notre site web.

#### 3.1.1 Front-End

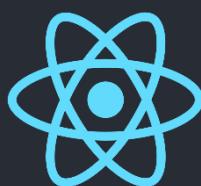
Le nombre des technologies de développement (coté client) est en croissance continue. Parmi les langages les plus populaires nous citons (**Voir Figure II.1**) :

- ❖ React JS
- ❖ Angular JS
- ❖ Vue JS



Figure II.1 : Les technologies Front-End les plus populaires <sup>(12)</sup>

##### 3.1.1.1 React JS



React JS est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web ‘monopage’ (SPA), via des composants dépendant d'un état et générant une page HTML. La **Figure II.2** illustre bien l'architecture d'une application utilisant React JS et Redux.

Avantages :

- ✓ DOM virtuel
- ✓ Haut niveau de flexibilité et de réactivité
- ✓ Facile à apprendre
- ✓ Librairie JavaScript 100% open source

Inconvénients :

- ✗ Long à maîtriser
- ✗ Nécessite une connaissance approfondie
- ✗ Manque de documentation officielle
- ✗ React est ‘unopinionated’, ce qui signifie que les développeurs ont parfois trop de choix.
- ✗

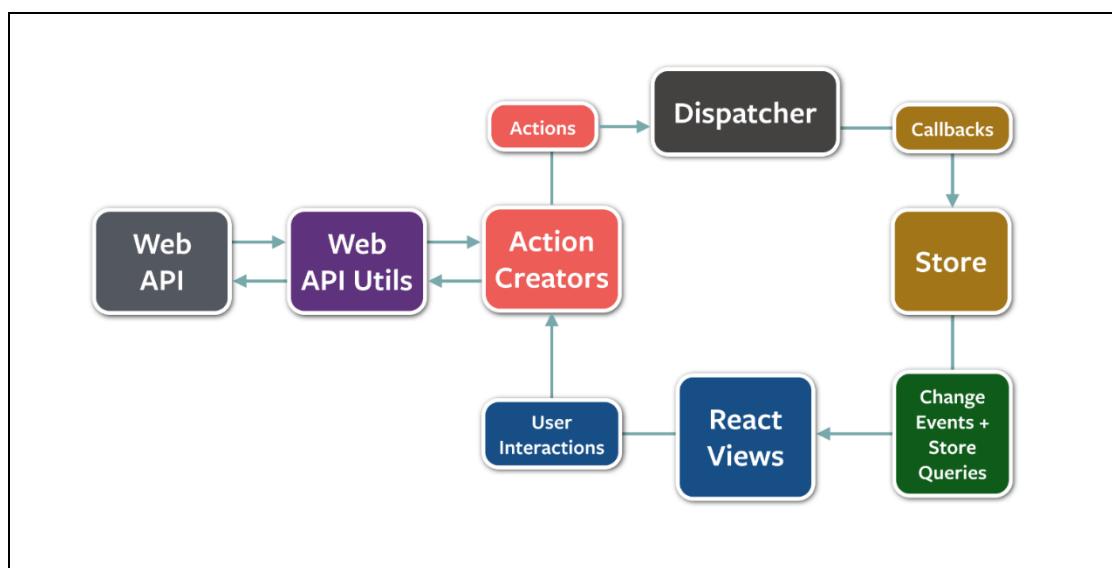


Figure II.2 : Architecture d'une application utilisant React JS <sup>(13)</sup>

### 3.1.1.2 Angular JS



Angular JS est un Framework JavaScript open source développé par Google en 2016, il permet de créer des applications web ‘monopage’ (SPA). Angular est basé sur le langage Type Script et repose aussi sur les langages HTML et CSS. Son architecture, illustrée par la **Figure II.3**, permet de structurer le code et séparer la vue (Interface) des modèles.

Avantages :

- ✓ Documentation détaillée
- ✓ Architecture MVVM (Model-View-ViewModel)
- ✓ Liaison des données bidirectionnelle qui minimise les risques d'erreurs possibles
- ✓ Nouvelles fonctionnalités améliorées telles que RX-JS, compilation plus rapide.

Inconvénients :

- ✗ Les problèmes de migration qui peuvent apparaître lors du passage d'une ancienne version à une autre plus récente
- ✗ Une syntaxe complexe hérité de la première version d'Angular

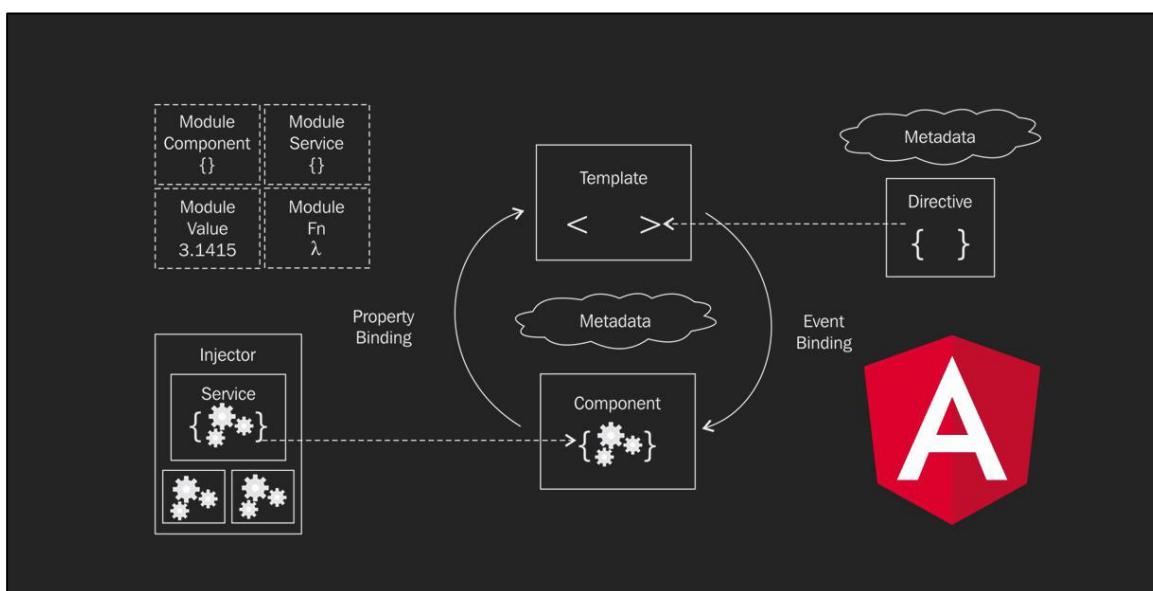


Figure II.3 : Architecture d'une application utilisant Angular JS<sup>(14)</sup>

### 3.1.1.3 Vue JS



Vue JS est un Framework JavaScript. Il a été créé par Evan You en 2014, développeur ayant travaillé chez Google. Il détient pratiquement les mêmes fonctionnalités que ces principaux concurrents React et Angular. La **Figure II.4** illustre bien l'architecture d'une application utilisant Vue JS.

Avantages :

- ✓ HTML optimisé
- ✓ Adaptabilité
- ✓ Intégration impressionnante à grande échelle

Inconvénients :

- ✗ Manque de ressources
- ✗ Il peut y avoir des problèmes d'intégration dans les projets de grande envergure.

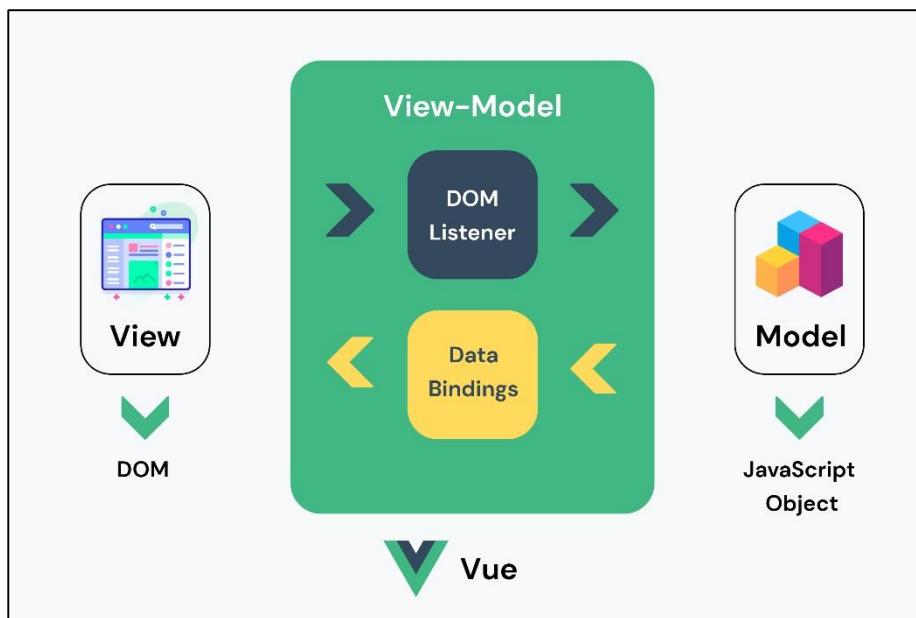


Figure II.4 : Architecture d'une application utilisant Vue JS

### 3.1.1.4 Etude comparative et résumé

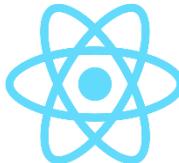
|                                       |  Angular JS |  React JS |  Vue JS |
|---------------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| <b>Performance</b>                    | Moyen                                                                                          | Haute                                                                                         | Haute                                                                                        |
| <b>Scalabilité</b>                    | Haute                                                                                          | Haute                                                                                         | Faible                                                                                       |
| <b>Apprentissage</b>                  | Difficile                                                                                      | Moyen                                                                                         | Faible                                                                                       |
| <b>Disponibilité des développeurs</b> | Haute                                                                                          | Haute                                                                                         | Faible                                                                                       |
| <b>Communauté des développeurs</b>    | Grande                                                                                         | Très grande                                                                                   | Petite                                                                                       |
| <b>Acceptation et confiance</b>       | Haute                                                                                          | Haute                                                                                         | Faible                                                                                       |

Tableau II.2 : Tableau comparatif entre les différentes technologies Front-End<sup>(15)</sup>

### 3.1.2 Back-End

Le Back-End, qui est invisible pour le client, représente une grande partie du développement d'un projet web. On peut décomposer le Back-End en deux parties essentielles :

- Serveur
- Base de données

#### 3.1.2.1 Serveur

Bien qu'il existe une multitude de langages de programmation et de plateformes côté serveur, on va citer quelques exemples telles que :

- ❖ **Python** : est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.
- ❖ **Java** : est un langage de programmation orientée objet créé par James Gosling et Patrick Naughton.
- ❖ **Node JS** : est une plateforme logicielle libre en JavaScript orientée vers les applications réseau hautement concurrentes qui doivent pouvoir monter en charge.
- ❖ **C#** : est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 et destiné principalement au plateforme Microsoft.NET.
- ❖ **PHP** : Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages web dynamiques via un serveur HTTP.

#### 3.1.2.2 Etude comparative entre les Framework Back-End

|                       | <b>Python</b> | <b>Java</b> | <b>Node JS</b> | <b>C#</b> | <b>PHP</b> |
|-----------------------|---------------|-------------|----------------|-----------|------------|
| <b>Population [%]</b> | 21.4%         | 41.6%       | 13.3%          | 26.8%     | 31.5%      |
| <b>Framework</b>      | Django        | Spring      | Express JS     | ASP.NET   | Laravel    |

Tableau II.3 : Tableau comparatif entre les Framework Back-End <sup>(16)</sup>

#### 3.1.2.3 Base de données

Une base de données permet de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une

activité. Lorsqu'il s'agit de choisir une base de données, la décision la plus importante est de choisir une structure de données relationnelle (SQL) ou non relationnelle (NoSQL). Le tableau ci-dessous dresse une comparaison entre ces deux structures :

|                    | <b>BD Relationnelle</b>                             | <b>BD NoSQL</b>                                 |
|--------------------|-----------------------------------------------------|-------------------------------------------------|
| <b>Type</b>        | Relationnelle                                       | Non relationnelle                               |
| <b>Données</b>     | Structurées, stockées dans des tables               | Non structurées stockées dans des fichiers JSON |
| <b>Schéma</b>      | Prédéfini, rigide                                   | Non prédéfini, flexible                         |
| <b>Évolutivité</b> | Verticale (augmenter la charge sur un seul serveur) | Horizontale (partage ou l'ajout de serveurs)    |
| <b>Transaction</b> | ACID                                                | Théorème CAP                                    |

Tableau II.4 : Bases de données Relationnelles vs NoSQL

- Selon cette comparaison, l'avantage principal des systèmes SQL est la puissance pour stocker et manipuler des données très structurées. Il permet de gérer des BDD respectant les 4 critères ACID : Atomicité, Cohérence, Isolation et Durabilité, et garantit donc la sécurité des transactions. Ainsi Les BDD relationnelles sont scalables. L'augmentation du volume de données stocké n'a pas d'incidence sur les données existantes et l'organisation de la base.

- **Choix de Postgresql**

Ce système de gestion de base de données partage la popularité de MySQL. Il s'agit d'un SGBD relationnel orienté objet dans lequel les objets définis par l'utilisateur et l'approche table sont combinés pour créer des structures de données plus complexes. En plus de cela, PostgreSQL a beaucoup de similitudes avec MySQL. Il vise à renforcer les normes de conformité et d'extensibilité. Par conséquent, il peut traiter n'importe quelle charge de travail, à la fois pour des produits mono-machine et des applications complexes. Détenue et développée par PostgreSQL Global Development Group, il reste un complètement open-source.

Parmi les avantages de ce système :

- **Évolutif.** L'évolutivité verticale est une caractéristique de PostgreSQL, contrairement au SGBD MySQL. Étant donné que presque toutes les solutions logicielles personnalisées ont tendance à croître, ce qui entraîne une extension de la base de données, cette option prend très bien en charge la croissance et le développement des entreprises.

- Prise en charge des types de données personnalisés.** PostgreSQL supporte de manière native un grand nombre de types de données par défaut, tels que JSON, XML, H-Store, etc. PostgreSQL en tire parti, car elle est l'une des rares bases de données relationnelles offrant une prise en charge solide des fonctionnalités NoSQL. De plus, il permet aux utilisateurs de définir leurs propres types de données. Étant donné que votre modèle commercial de logiciel peut nécessiter différents types de bases de données tout au long de son existence pour améliorer les performances ou l'exhaustivité des applications, cette option apporte une flexibilité accrue à la table de données.
- Open-source et communautaire.** Postgres est complètement open-source et soutenu par sa communauté, ce qui le renforce en tant qu'écosystème complet. En outre, les développeurs peuvent toujours compter sur une assistance communautaire rapide et gratuite.

### 3.1.2.4 Classement des bases de données

La **Figure II.5** présente le classement de PostgreSQL par rapport d'autres BDD présentée par le site DB-Engines.com.

| Rank     |          |          | DBMS                 | Database Model             | Score    |          |          |
|----------|----------|----------|----------------------|----------------------------|----------|----------|----------|
| Dec 2019 | Nov 2019 | Dec 2018 |                      |                            | Dec 2019 | Nov 2019 | Dec 2018 |
| 1.       | 1.       | 1.       | Oracle               | Relational, Multi-model    | 1346.39  | +10.33   | +63.17   |
| 2.       | 2.       | 2.       | MySQL                | Relational, Multi-model    | 1275.67  | +9.38    | +114.42  |
| 3.       | 3.       | 3.       | Microsoft SQL Server | Relational, Multi-model    | 1096.20  | +14.29   | +55.86   |
| 4.       | 4.       | 4.       | PostgreSQL           | Relational, Multi-model    | 503.37   | +12.30   | +42.74   |
| 5.       | 5.       | 5.       | MongoDB              | Document, Multi-model      | 421.12   | +7.94    | +42.50   |
| 6.       | 6.       | 6.       | IBM Db2              | Relational, Multi-model    | 171.35   | -1.25    | -9.40    |
| 7.       | 7.       | 8.       | Elasticsearch        | Search engine, Multi-model | 150.25   | +1.85    | +5.55    |
| 8.       | 8.       | 7.       | Redis                | Key-value, Multi-model     | 146.23   | +1.00    | -0.59    |
| 9.       | 9.       | 9.       | Microsoft Access     | Relational                 | 129.47   | -0.60    | -10.04   |
| 10.      | 10.      | 11.      | Cassandra            | Wide column                | 120.71   | -2.52    | -1.10    |
| 11.      | 11.      | 10.      | SQLite               | Relational                 | 120.36   | -0.66    | -2.65    |
| 12.      | 12.      | 12.      | Splunk               | Search engine              | 90.53    | +1.46    | +8.34    |
| 13.      | 13.      | 14.      | MariaDB              | Relational, Multi-model    | 86.79    | +1.22    | +9.53    |
| 14.      | 14.      | 15.      | Hive                 | Relational                 | 86.05    | +1.83    | +18.67   |
| 15.      | 15.      | 13.      | Teradata             | Relational, Multi-model    | 78.49    | -1.86    | -0.67    |
| 16.      | 16.      | 21.      | Amazon DynamoDB      | Multi-model                | 61.63    | +0.26    | +7.33    |
| 17.      | 17.      | 16.      | Solr                 | Search engine              | 57.22    | -0.56    | -4.13    |
| 18.      | 19.      | 20.      | SAP Adaptive Server  | Relational                 | 55.55    | +0.25    | -0.27    |
| 19.      | 18.      | 18.      | FileMaker            | Relational                 | 55.14    | -0.60    | -1.51    |
| 20.      | 20.      | 19.      | SAP HANA             | Relational, Multi-model    | 54.17    | -0.94    | -2.14    |

Figure II.5 : Classement des bases de données <sup>(17)</sup>

### 3.1.3 Choix technologique de notre application

On va développer notre portail web avec la pile React JS, Redux, NodeJS, Express JS, PostgreSQL et Knex JS qui est une pile logicielle gratuite avec un code source libre. L'un des principaux avantages de cette pile est qu'elle possède une approche efficace et moderne de développement web. La **Figure II.6** montre cette interaction entre ses différentes technologies.

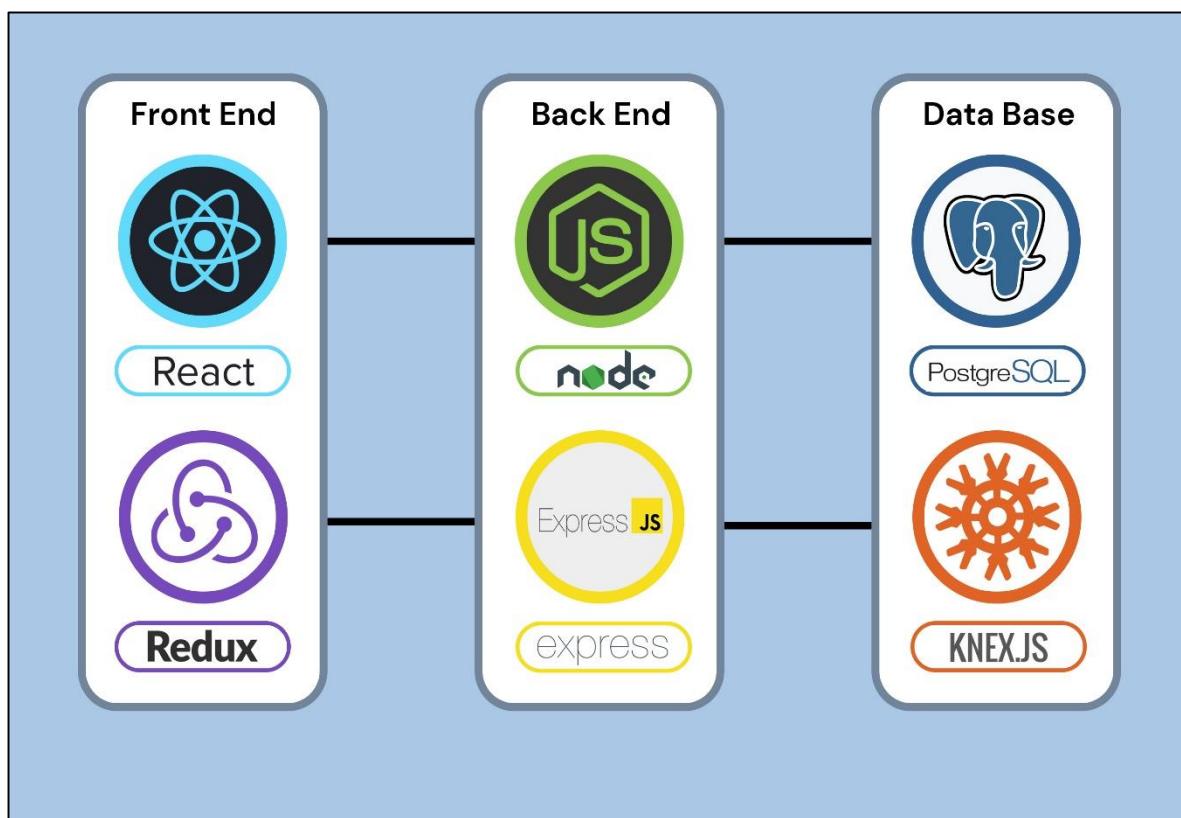


Figure II.6 : Choix technologique de notre application

#### 3.1.3.1 Redux

Redux se présente comme *un système de centralisation des données et des actions*. React est particulièrement convaincant lorsqu'on commence mais on se retrouve rapidement confronté à certains problèmes de conceptions lorsque nos applications deviennent un peu plus élaborées qu'une simple liste. Avec Redux, on ajoute la *notion de store*. Un *store* est global, il contient toutes les informations de tous les composants qui y sont rattachés. Lorsqu'un composant **A** lance une action et entraîne une modification du *store* (dit aussi “*state global*”), le composant **B** aura également les nouvelles données du composant **A**.

Chaque composant peut mettre en place des comportements différents en fonction de l'état des autres.

L'architecture de REDUX est composée comme suit :

- **VIEW PROVIDER** : Représente un View Framework pour s'inscrire avec STORE.
- **ACTION** : un objet pur créé pour stocker les informations relatives à l'événement d'un utilisateur. Il inclut les informations telles que : le type d'action, l'heure de l'événement, l'emplacement de l'événement, ses coordonnées et quel state qu'il vise à modifier.
- **STORE** : Gère l'état de l'application avec la fonction dispatch (action).
- **MIDDLEWARE** : (Logiciel intermédiaire) Fournit un moyen d'interagir avec les objets Action envoyés à STORE avant leur envoi à REDUCER.
- **REDUCER** : (Modificateur) Une fonction pure pour renvoyer un nouvel état à partir de l'état initial. Il ne modifie pas l'état de l'application. Au lieu de cela, il créera une copie de l'état d'origine et le modifiera pour obtenir un nouvel état.

La **Figure II.7** explique bien le cycle entre React JS et Redux.

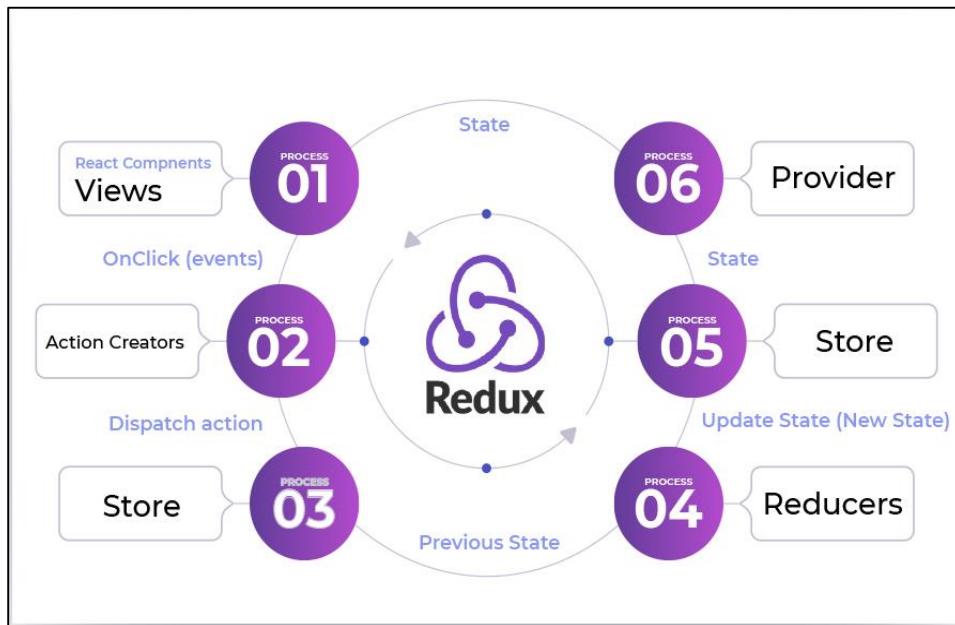


Figure II.7 : Cycle React JS - Redux

### 3.1.3.2 Knex.js

Knex.js est un SQL builder, il permet d'écrire des requêtes avec un syntaxe d'objet. Knex.js fonctionne parfaitement avec Postgres, même les champs spéciaux comme le JSONb sont supportés. Les transactions sont aussi supportées nativement. Son système de migrations et de seed est un allié précieux et permet un travail en équipe simplifié.

- **Migrations :** L'un des grands avantages de Knex est l'outil de migration qui peut nous faciliter la gestion des tables DB. Il crée des fichiers séquentiels avec des horodatages, il gère également les modifications de table, en ajoutant / supprimant des colonnes sans toucher au fichier de migration d'origine.
- **Seed :** Semblable aux migrations, le module knex nous permet de créer des scripts pour insérer des données initiales dans nos tables appelées "seed files". Si nous avons des relations sur nos tables, "the seeding " doit être dans un ordre spécifique afin que nous puissions compter sur des données qui peuvent déjà existées. Par exemple, nous devons d'abord «seed» la table d'établissement, car notre table de statistiques doit valider une clé étrangère avec l'ID d'établissement qui existe déjà.

### 3.1.3.3 Objection.js

Objection.js est un ORM pour Node.js qui vise à rendre l'utilisation de la pleine puissance de SQL et du moteur de base de données sous-jacent aussi simple que possible. Objection.js est construit sur un générateur de requêtes SQL qui est knex. Toutes les bases de données prises en charge par knex sont prises en charge par objection.js. Il nous donne :

- Une manière déclarative simple de définir des modèles et des relations entre eux
- Un moyen simple de récupérer, insérer, mettre à jour et supprimer des objets en utilisant toute la puissance des mécanismes SQL.
- Transactions faciles à utiliser
- Prise en charge officielle de TypeScript
- Validation facultative du schéma JSON
- Une façon de stocker des documents complexes sous forme de lignes uniques

### 3.2 Choix du modèle conceptuel

Avant de développer notre application, il est indispensable de choisir un modèle de conception (pattern design). Parmi les patrons les plus connues, nous mentionnons les patrons Modèle-Vue-Contrôleur (MVC), Modèle-Vue-Vue Modèle (MVVM). Mais, quelle est la différence entre eux ?

#### 3.2.1 MVC

MVC est un modèle de conception de logiciel. Il pousse une bifurcation de préoccupations, ce qui signifie que le modèle et la logique du contrôleur sont séparés de l'interface utilisateur (vue). Par conséquent, la maintenance et les tests de l'application deviennent simples et faciles. Le modèle de conception MVC divise une application en trois aspects principaux : Modèle, Vue et Contrôleur, comme indiqué dans la **Figure II.8** ci-dessous :

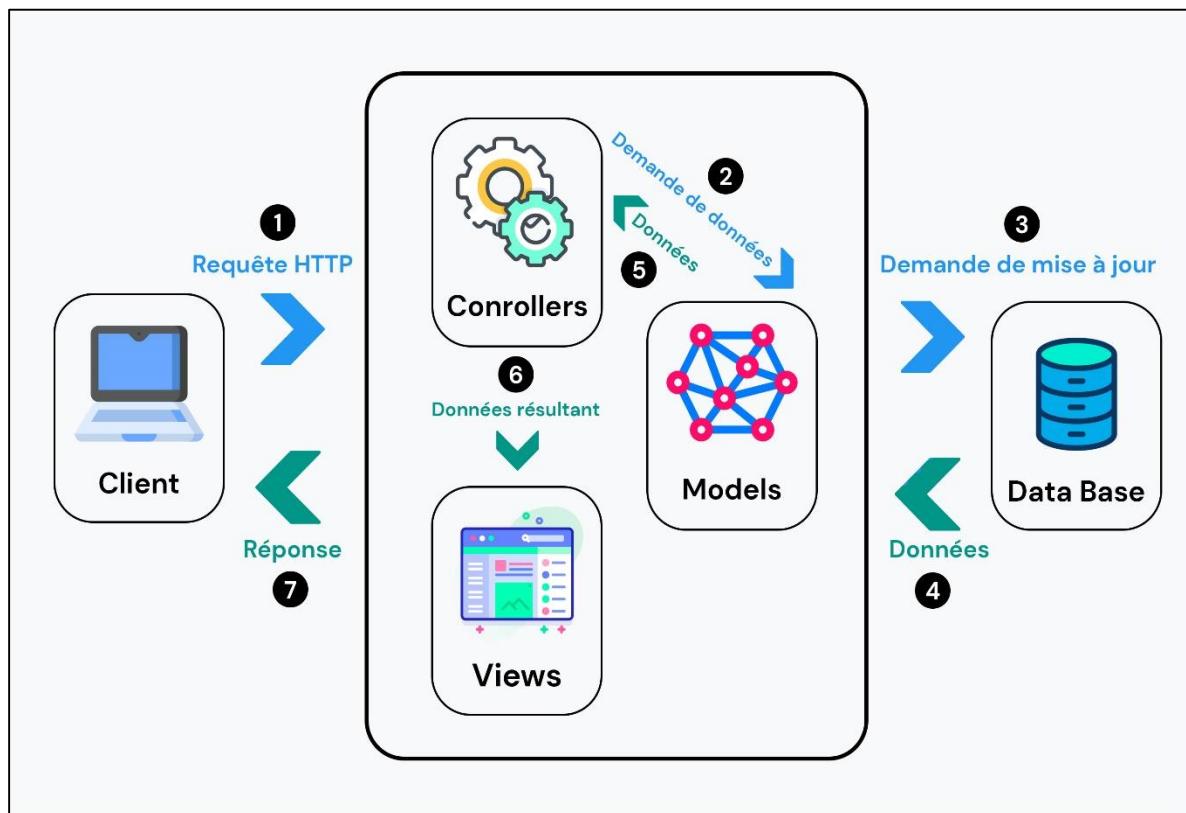


Figure II.8 : Architecture MVC

- **Model** : Le modèle représente une collection de classes qui explique la logique métier, à savoir le modèle métier et le modèle de données (opérations d'accès aux données). Il définit également les règles métier pour les moyens de données comme la façon dont les données peuvent être modifiées et manipulées.
- **View** : La vue représente les composants de l'interface utilisateur tels que CSS, jQuery, HTML, etc (dans le web). Elle affiche les données reçues du contrôleur comme résultat.
- **Controller** : La responsabilité du contrôleur est de traiter les demandes entrantes. Il obtient l'entrée des utilisateurs via la vue, puis traite les données de l'utilisateur via le modèle, en renvoyant les résultats à View. Il agit comme un médiateur entre la vue et le modèle.
  - ➔ En séparant le code d'une application web de cette manière, l'accès aux données est totalement séparé du reste de l'application. Sans dépendance à l'interface utilisateur ou à la logique des actions de l'utilisateur, il est bien plus simple d'écrire des tests automatisés. De plus, la duplication de code sera considérablement amoindrie.

Mais qu'en est-il de l'adaptation et de l'évolution de notre application ? Si le modèle de données est amené à changer, nous devons obligatoirement modifier l'ensemble de nos entités.

### 3.2.2 MVVM

Le modèle MVVM prend en charge la liaison de données bidirectionnelle entre View et ViewModel. Cela permet la propagation automatique des modifications, à l'intérieur de l'état de ViewModel à la vue. Généralement, ViewModel utilise le modèle d'observateur pour informer les modifications apportées au modèle View-Model au modèle. La **Figure II.9** suivante illustre la structure générique d'une architecture MVVM.

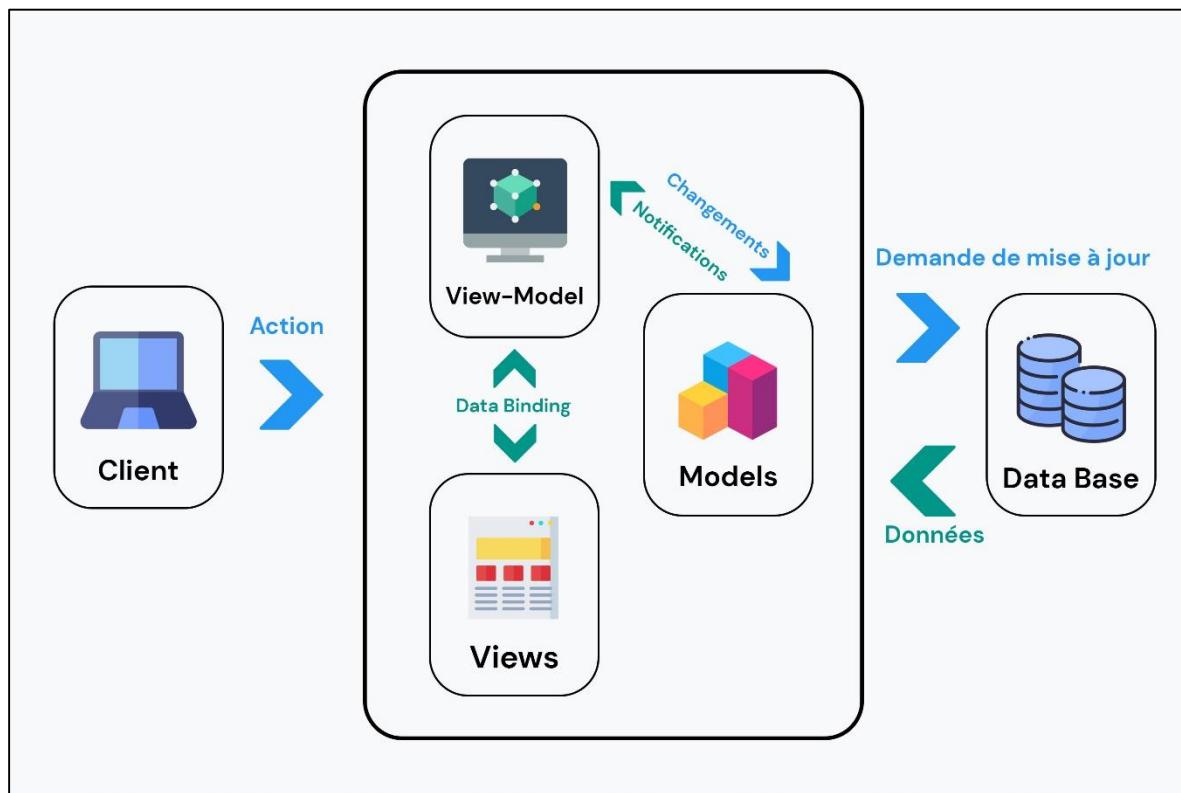


Figure II.9 : Architecture MVVM

- **Model** : Le modèle représente une collection de classes qui explique le modèle métier et le modèle de données. Il définit également les règles métier.
- **Vue** : La vue représente les composants de l'interface utilisateur tels que CSS, jQuery, HTML, etc...
- **ViewModel** : Le modèle de vue est responsable de l'affichage des méthodes, commandes et autres fonctions qui aident à maintenir l'état de la vue, à manipuler le modèle à la suite d'actions sur la vue et à déclencher les événements dans la vue elle-même.

### 3.3 Notre choix

D'après cette étude, nous avons choisi de travailler avec l'architecture MVC, car elle permet de bien séparer la partie logique de la partie présentation et convient parfaitement à la nature et dimension de notre projet. En effet, vu cette séparation entre les parties, il est très facile

d'ajouter et de modifier le code d'une section sans que le reste ne soit toucher. C'est un pattern qui se prête très bien au développement des petites et moyennes applications.

### 3.3.1 Pour la couche sécurité du model MVC

- **OAuth** : abréviation de « Open Authorization », est un protocole standard ouvert permettant une autorisation API sécurisée. Spécifique au domaine de la programmation, le terme API désigne dans ce contexte une interface agissant comme un transmetteur de données entre les différentes applications, interfaces utilisateur ou pages Web. Afin d'éviter tout risque d'interception de ces données par des tiers et leur utilisation abusive, une autorisation des transferts de données effectués entre ces API est par conséquent nécessaire, voir **Figure II.10.**

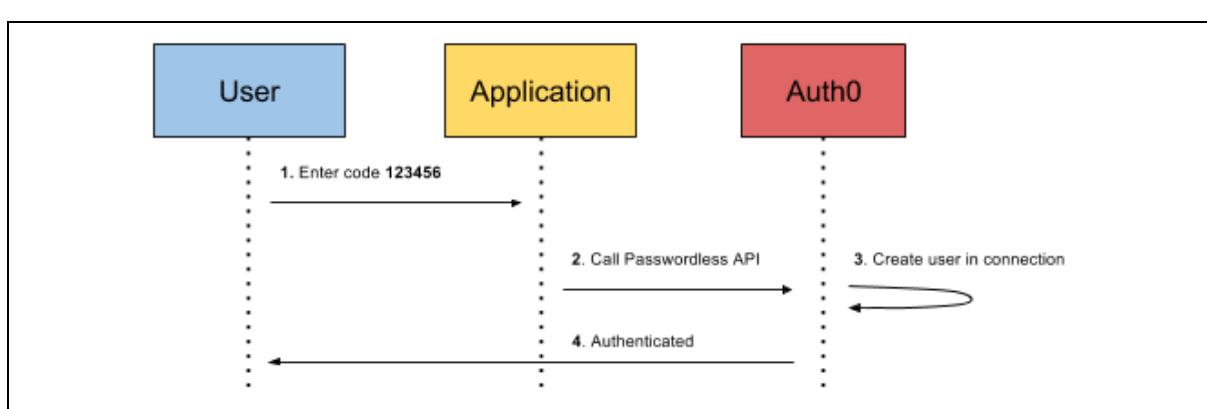


Figure II.10 : Exemple d'utilisation d'OAuth <sup>(18)</sup>

## Conclusion

Durant ce chapitre, nous avons non seulement exposé la mise en pratique du processus 2TUP, mais nous avons également capturé les différents besoins de notre projet en les classant en trois catégories : fonctionnelles ; non fonctionnelles et techniques. Après l'analyse des exigences et le choix des technologies à utiliser, il est le temps de passer à la conception de notre application dans le chapitre suivant.

# CHAPITRE III : CONCEPTION

## Introduction

Une fois la phase de spécification des besoins est achevée, nous entamons au niveau de ce chapitre la phase de conception. On présentera dans la première partie la conception générique en appuyant sur le formalisme UML basé sur les diagrammes. Ensuite, la deuxième partie est dédiée à la conception graphique de notre application web. Enfin, la troisième partie sera consacrée à la réalisation des différents diagrammes UML.

## 1 Conception générique

La phase de conception permet de décrire de manière ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

### 1.1 Le langage UML

Pour faciliter notre tâche nous avons fait recours au langage de modélisation unifié « UML » (Unified Modeling Language) qui permet de modéliser un problème de façon standard. Ce langage qui est né de la fusion de plusieurs méthodes existantes auparavant est devenu une référence en termes de modélisation objet :

- ❖ Il permet le gain, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.
- ❖ Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes.
- ❖ Il est formel et normalisé : son caractère polyvalent et sa souplesse en font un langage universel.

## 2 Conception préliminaire des interfaces – Prototypes

Une maquette est un produit jetable donnant aux utilisateurs une vue concrète mais non définitive de la future interface de l'application. Elle est développée rapidement afin d'améliorer la relation développeur-client. Les maquettes de notre application sont les suivantes :

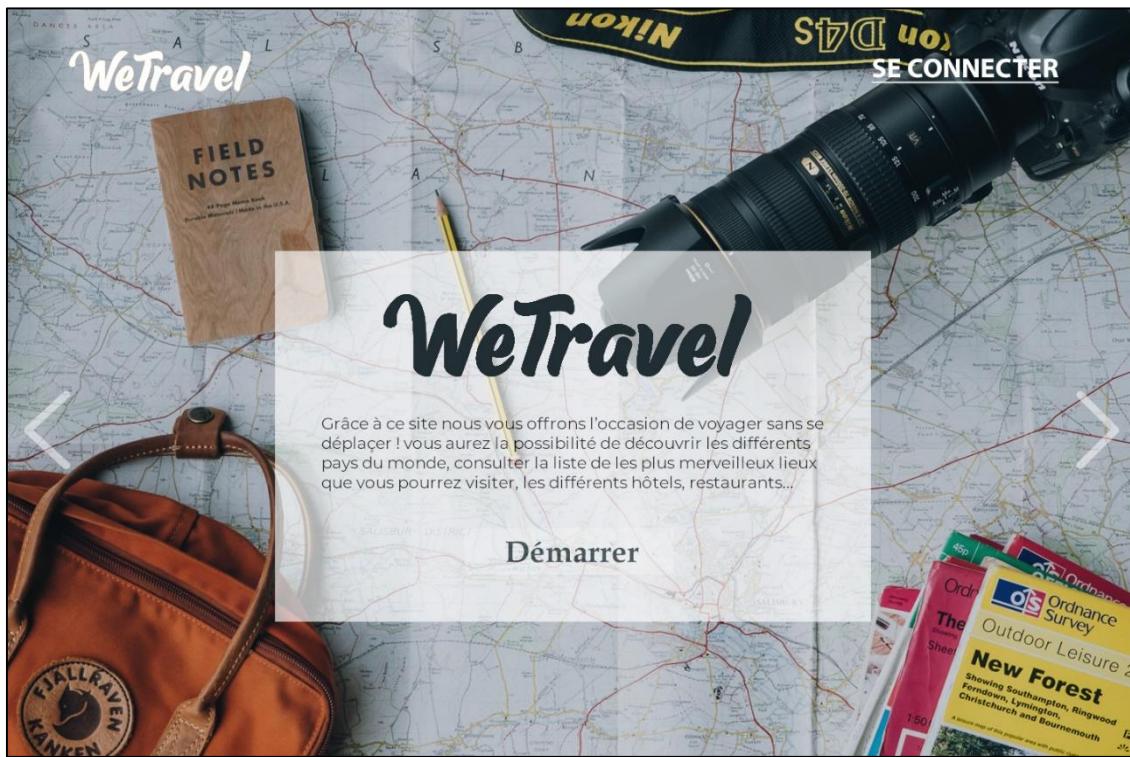


Figure III.1 : Maquette préliminaire de l'interface d'accueil

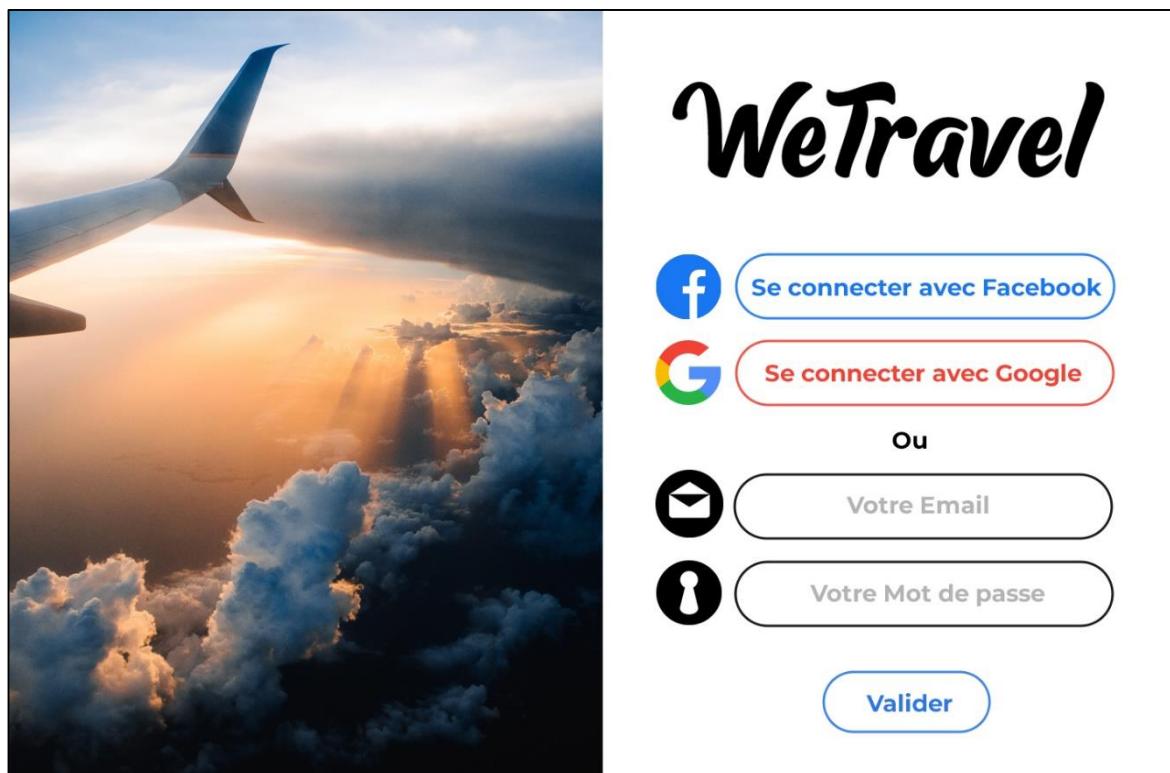


Figure III.2 : Maquette préliminaire de l'interface « Inscription et Authentification »



Figure III.3 : Maquette préliminaire de l'interface « Destination »

### 3 Conception détaillée

#### 3.1 Les vues de Kruchten

Kruchten propose des perspectives indépendantes et complémentaires pour cerner la méthodologie 2TUP qu'on a choisi, pour cela le choix de la démarche de Kruchten nous permettra de passer à la conception. Le langage utilisé sera le langage UML qui propose 13 diagrammes différents qui seront utiliser selon le besoin et répartie selon les vues de Kruchten, voir **Figure III.4**.



Figure III.4 : Vue 4+1 (Kruchten)

- **La vue des cas d'utilisations :** est un guide pour toutes les autres vues : c'est la définition des besoins des clients du système et de l'architecture sur la satisfaction de ces besoins, la colle qui unifie les autres vues, et qui motive les choix, identifie les interfaces et force à se concentrer sur les besoins des utilisateurs. Cette vue présente le :
  - Diagramme de cas d'utilisation
  - Diagramme de package
- **La vue logique :** décrit les aspects statiques et dynamiques d'un système en termes de classes, d'objets, de connexions et de communications. Elle se concentre sur l'abstraction et l'encapsulation. Cette vue présente le :
  - Diagramme de classe
  - Diagramme d'objet
- **La vue de processus :** Cette vue propose la description des interactions entre les différents processus (threads, tâches). Elle exprime la synchronisation et l'allocation des objets. La vue de processus ou dite aussi vue de comportement s'occupe de la vérification du respect des contraintes de fiabilité, d'efficacité et de performances des systèmes multitâches. Cette vue présente le :
  - Diagramme de séquence
  - Diagramme d'activité

- Diagramme de collaboration
- Diagramme d'état de transition
- Diagramme de temps

➤ **La vue des composants :** Cette vue de bas niveau (aussi appelée "vue de réalisation"), permet de visualiser l'organisation des composants (bibliothèque dynamique et statique, code source...) dans l'environnement de développement. Elle permet également de gérer la configuration (auteurs, versions...). Cette vue présente le :

- Diagramme de composants
- Diagramme de structure composite

➤ **La vue de déploiement :** la vue de déploiement est utile pour la maintenance régulière du système pour la tolérance aux fautes et aux pannes. Elle permet aussi de traiter les contraintes de temps de réponse et de bande passantes. Cette vue présente le :

- Diagramme de déploiement

Les 4 diagrammes sélectionnés pour ce projet sont divisés en fonction des vues de Kruchten :

Pour la vue des cas d'utilisateurs : Diagramme de cas d'utilisation

Pour la vue des processus : Diagramme de séquence

Pour la vue Logique : Diagramme de classe

Pour la vue des composants & Vue de déploiement : Diagramme de composants

### 3.2 Identification des cas d'utilisations

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il représente une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use cases). Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs. Il représente toutes les fonctionnalités que le système doit fournir.

### 3.2.1 Cas d'utilisation général

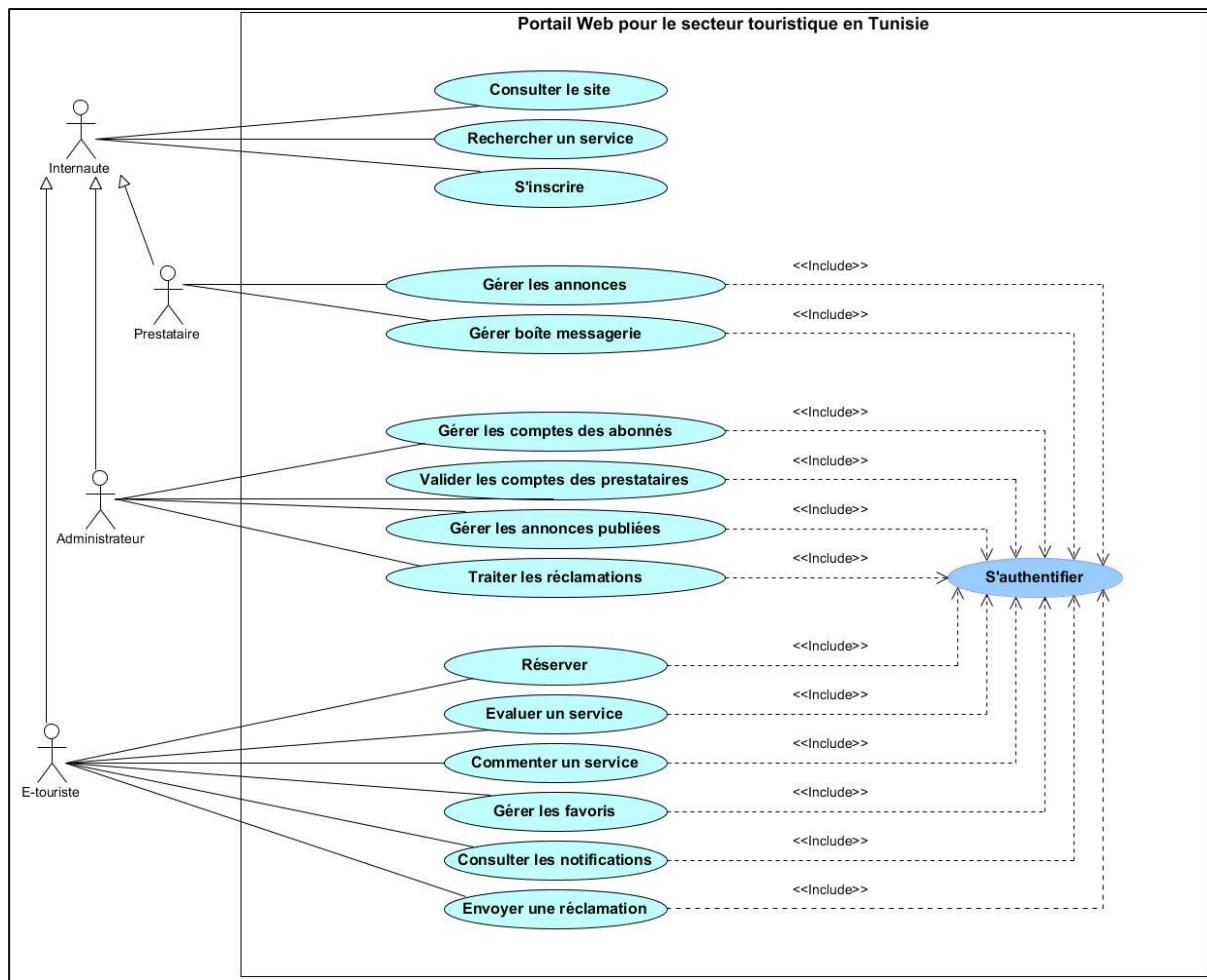


Figure III.5 : Diagramme de cas d'utilisation général

Le **tableau III.1** représente les différents cas d'utilisation, les acteurs impliqués et une description pour chaque cas.

| Cas d'utilisation      | Acteur(s)   | Description                                                                                                                    |
|------------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------|
| Consulter le site      | Internaute  | L'internaute peut consulter les offres de plusieurs activités comme l'hôtellerie, la restauration et les agences de voyage,... |
| Rechercher un service  | Internaute  | Il peut rechercher des services selon ses choix.                                                                               |
| S'inscrire             | Internaute  | Il peut créer un profil dans le site pour devenir un abonné.                                                                   |
| Gérer les annonces     | Prestataire | Le prestataire peut consulter, publier, modifier ou supprimer une annonce.                                                     |
| Gérer boîte messagerie | Prestataire | Le prestataire peut échanger des messages avec d'autres prestataires.                                                          |

|                                      |                |                                                                                                                                            |
|--------------------------------------|----------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Gérer les comptes des abonnés        | Administrateur | L'administrateur a un accès à la liste des abonnés du site. Il a le droit de les supprimer ou les bloquer.                                 |
| Valider les comptes des prestataires | Administrateur | L'administrateur doit valider les comptes des prestataires.                                                                                |
| Gérer les annonces publiées          | Administrateur | L'administrateur vérifie toutes les annonces publiées. Il peut soit les valider, soit les supprimer.                                       |
| Traiter les réclamations             | Administrateur | L'administrateur consulte les réclamations des abonnés. Il peut supprimer l'annonce ou bien bloquer l'abonné qui a lancé la signalisation. |
| Réserver                             | E-touriste     | L'abonné peut réserver dans un hôtel ou dans des restaurants, musées, événements et agences de voyage.                                     |
| Evaluer un service                   | E-touriste     | Les E-touristes peuvent noter les services existants afin de les évaluer.                                                                  |
| Commenter un service                 | E-touriste     | L'abonné peut donner son avis sur les services publiés dans un Forum                                                                       |
| Gérer les favoris                    | E-touriste     | L'abonné peut enregistrer des annonces qui l'intéressent dans ses favoris ou les supprimer.                                                |
| Consulter les notifications          | E-touriste     | L'abonné est notifié à propos des nouvelles suggestions ou commentaires sur ses annonces.                                                  |
| Envoyer une réclamation              | E-touriste     | L'abonné peut envoyer une réclamation à l'administrateur via un formulaire dans lequel il explique sa demande.                             |

Tableau III.1 : Tableau des cas d'utilisation

### 3.2.2 Description détaillée des cas d'utilisation

Dans cette section nous allons présenter une description détaillée de quelques cas d'utilisation.

#### 3.2.2.1 Cas d'utilisation associés à « Internaute »

La **Figure III.6** illustre les fonctionnalités que peuvent être faites par l'internaute lorsqu'il visite le site. Il peut consulter le site, créer un profil pour devenir un abonné et aussi rechercher des annonces.

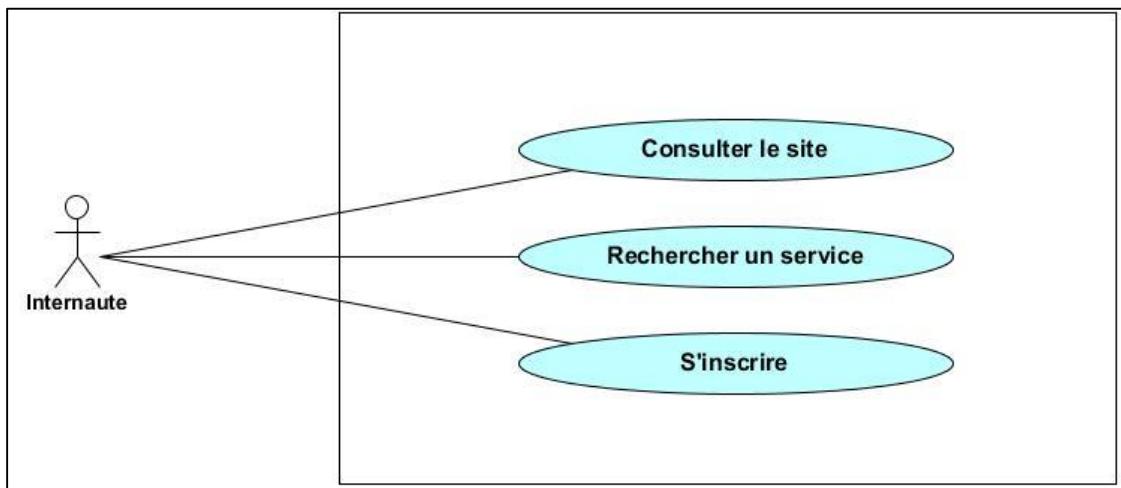


Figure III.6 : Diagramme de Cas d'utilisation associés à « Internaute »

### 3.2.2.2 Cas d'utilisation associés au « Prestataire »

Après l'authentification, le prestataire est autorisé à faire les fonctionnalités illustrées par la **Figure III.7**. Nous citons : Gérer les annonces, les consulter ou encore il peut gérer la boîte messagerie.

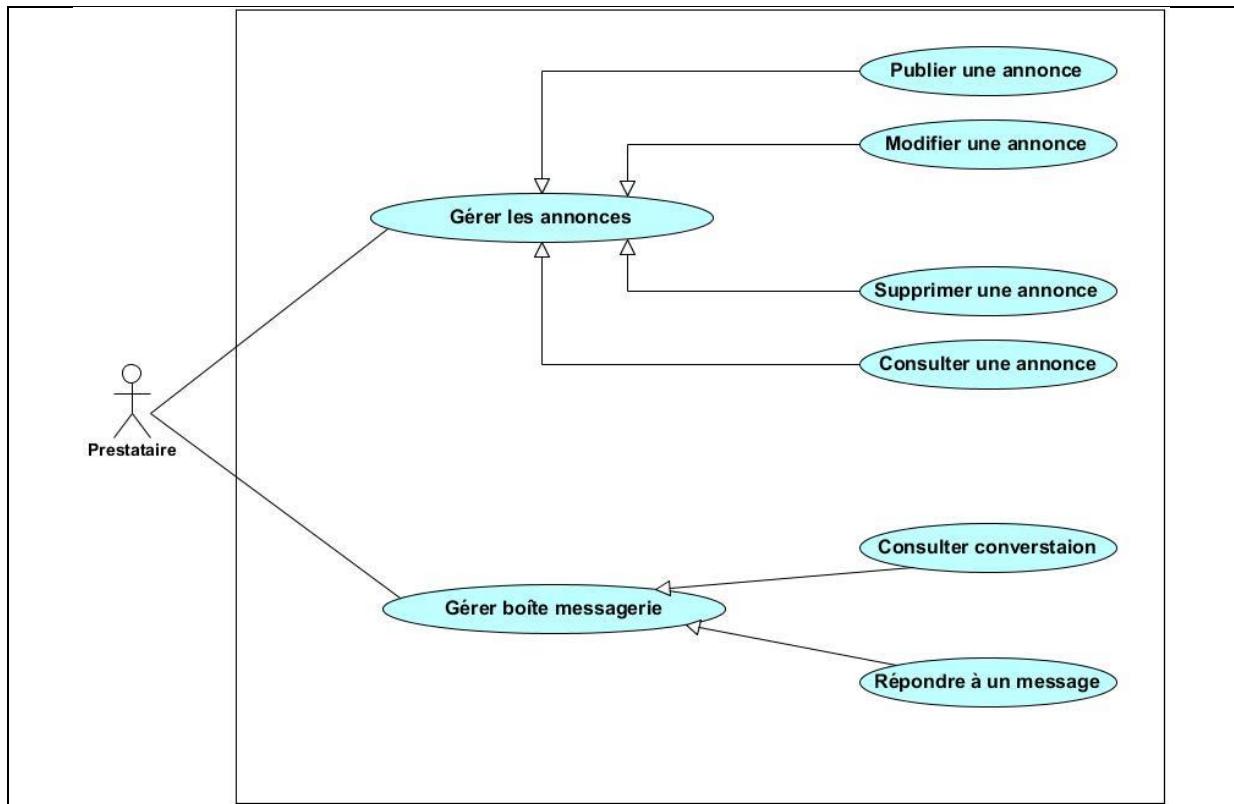


Figure III.7 : Diagramme de cas d'utilisation associés au « Prestataire »

### 3.2.2.3 Cas d'utilisation associés à « Administrateur »

L'administrateur, après l'authentification, il peut assurer les fonctionnalités illustrées par la **Figure III.8**. Il peut gérer la liste des abonnés du site et les annonces publiées. Comme il peut traiter les réclamations des abonnés et valider les comptes des prestataires.

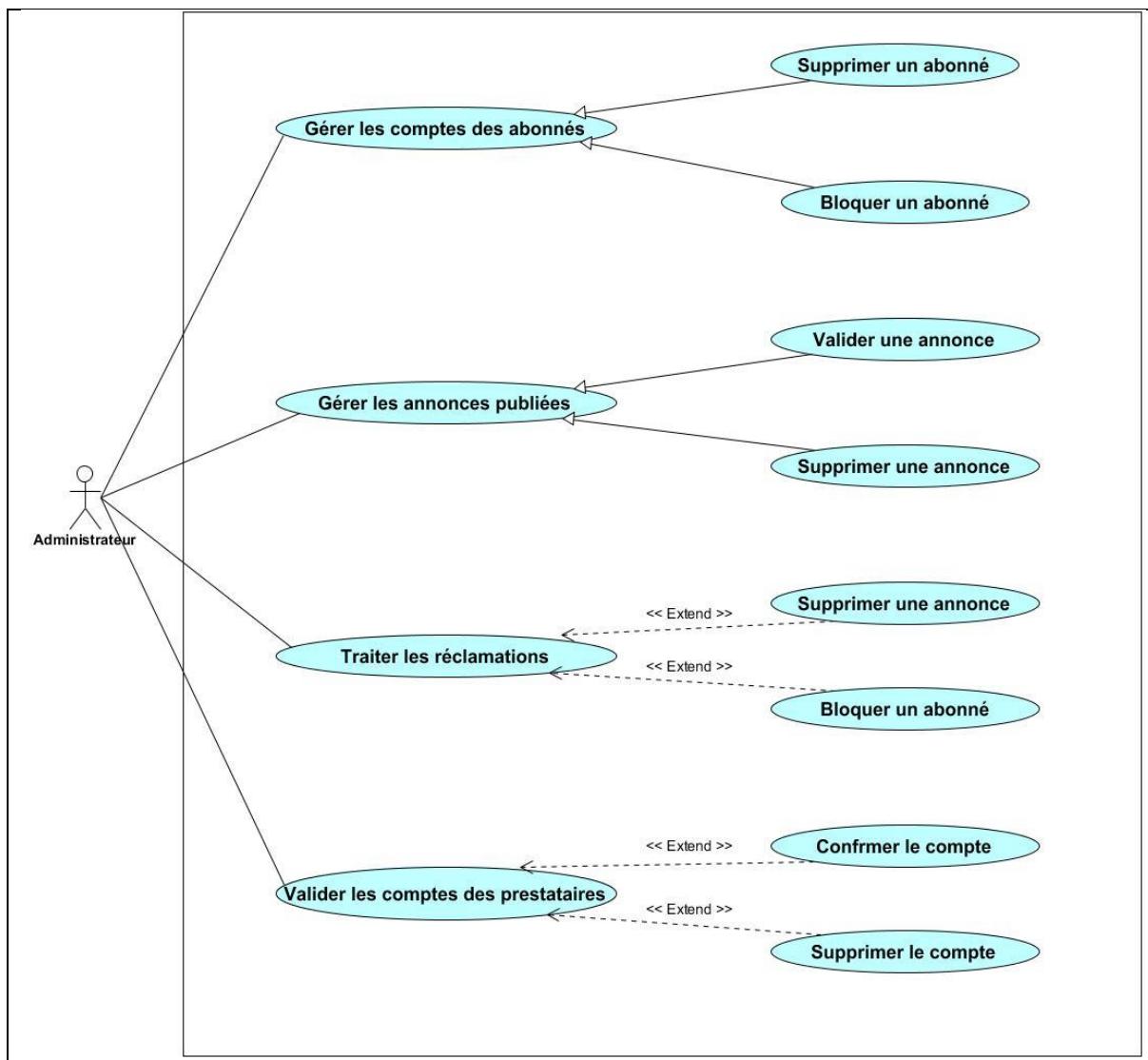


Figure III.8 : Diagramme de cas d'utilisation associés à « Administrateur »

### 3.2.2.4 Cas d'utilisation associés à l'« E-touriste »

La **Figure III.9** illustre les fonctionnalités que peut faire l'abonné après l'authentification. Nous citons : consulter une annonce particulière pour l'ajouter à sa liste des favoris, la supprimer, évaluer un service ou le commenter. Il peut encore envoyer une réclamation à l'administrateur, consulter les notifications et effectuer des réservations.

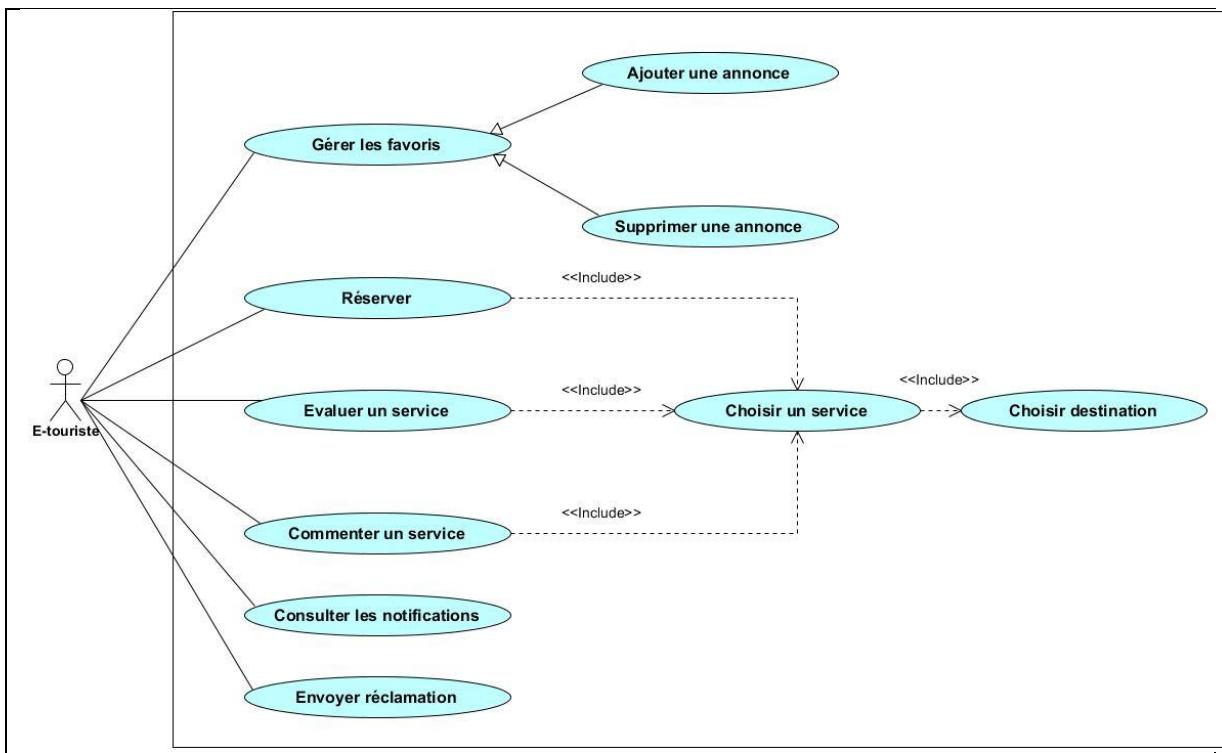


Figure III.9 : Diagramme de Cas d'utilisation associés à l'« E-touriste »

### 3.3 Diagrammes de séquence

Le diagramme de séquence « système » est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. Ce diagramme est utilisé pour représenter certains aspects dynamiques d'un système : dans le contexte d'une opération, d'un système, d'un sous-système, d'un cas d'utilisation (un scénario d'un cas d'utilisation) selon un point de vue temporel. Dans ce qui suit, une représentation quelques scénarios.

### 3.3.1 Diagramme de séquence « S'inscrire »

Le tableau III.2 et la Figure III.10 décrivent le diagramme de séquence « S'inscrire ».

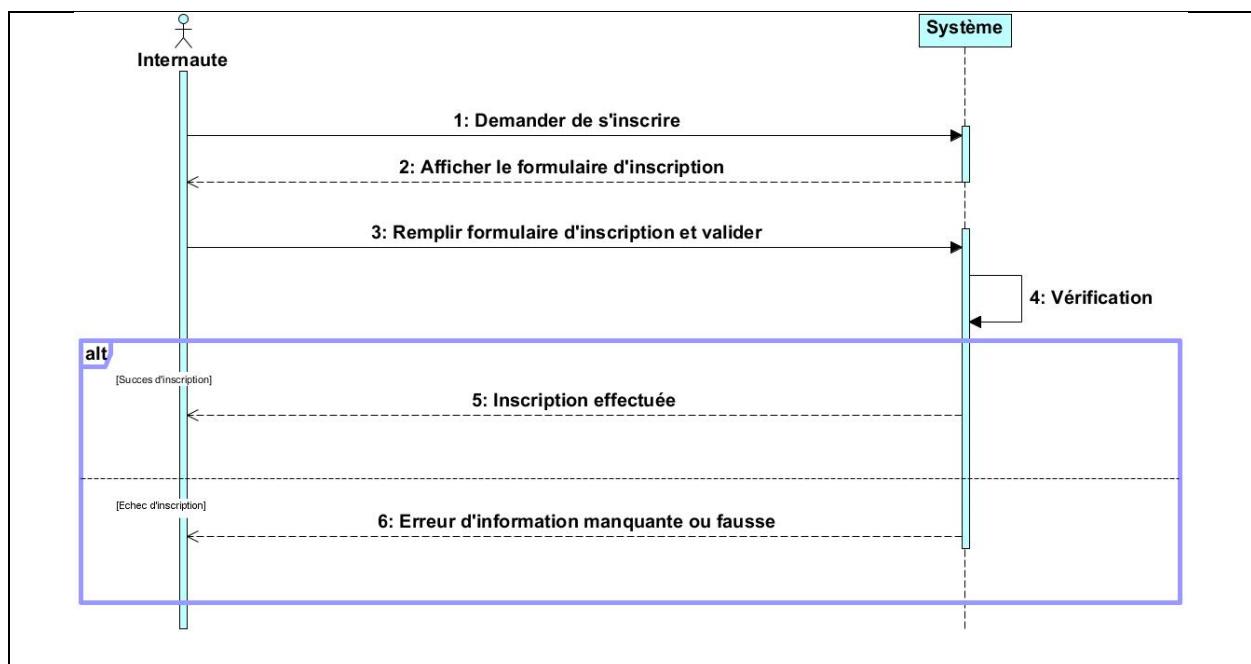


Figure III.10 : Diagramme de séquence « S'inscrire »

|                            |                                                                                                                                                                                                                                                                          |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cas d'utilisation</b>   | S'inscrire                                                                                                                                                                                                                                                               |
| <b>Acteur</b>              | Internaute                                                                                                                                                                                                                                                               |
| <b>Intérêts</b>            | Il permet à l'internaute de créer un profil pour devenir un abonné au site.                                                                                                                                                                                              |
| <b>Précondition</b>        | Accéder à la page d'inscription                                                                                                                                                                                                                                          |
| <b>Scénario nominal</b>    | <ol style="list-style-type: none"> <li>1. L'internaute demande l'inscription au site.</li> <li>2. Le système affiche le formulaire d'inscription.</li> <li>3. L'internaute remplit le formulaire et il valide.</li> <li>4. Le système effectue l'inscription.</li> </ol> |
| <b>Post condition</b>      | Un nouvel abonné est inscrit.                                                                                                                                                                                                                                            |
| <b>Scénario alternatif</b> | <ol style="list-style-type: none"> <li>3.1. Le système indique à l'internaute qu'il y a des champs vides.</li> <li>3.2. L'internaute remplit les champs vides et il valide.</li> </ol> <p>- Reprise de l'étape 4 du scénario nominal.</p>                                |
| <b>Exception</b>           | Utilisateur non inscrit                                                                                                                                                                                                                                                  |

Tableau III.2 : Description textuelle du diagramme de séquence « S'inscrire »

### 3.3.2 Diagramme de séquence « Authentification »

Le **tableau III.3** décrit le diagramme de séquence “Authentification” illustré par la **Figure III.11**.

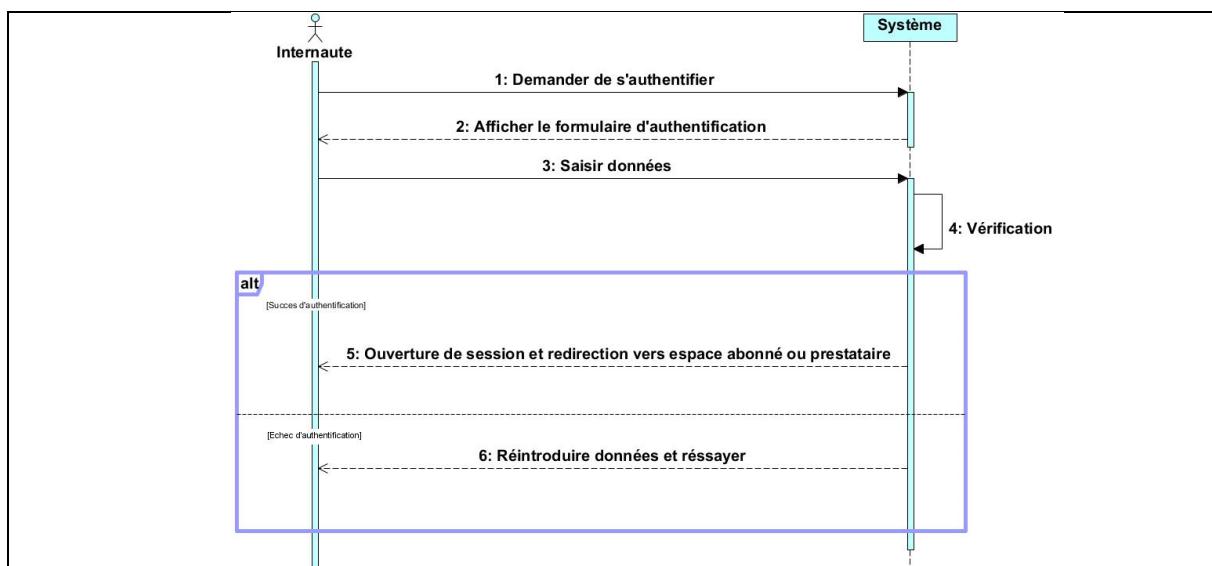


Figure III.11 : Diagramme de séquence « Authentification »

|                            |                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cas d'utilisation</b>   | Authentification                                                                                                                                                                                                                                                                                                                                  |
| <b>Acteur</b>              | Internaute.                                                                                                                                                                                                                                                                                                                                       |
| <b>Intérêts</b>            | Il permet à l'internaute de s'authentifier.                                                                                                                                                                                                                                                                                                       |
| <b>Précondition</b>        | Accéder à la page d'authentification.                                                                                                                                                                                                                                                                                                             |
| <b>Scénario nominal</b>    | <ol style="list-style-type: none"> <li>1. L'internaute demande de s'authentifier.</li> <li>2. Le système affiche le formulaire d'authentification.</li> <li>3. L'internaute saisie les données.</li> <li>4. Le système vérifie les données saisies.</li> <li>5. Ouverture de session et redirection vers espace abonné ou prestataire.</li> </ol> |
| <b>Post condition</b>      | L'internaute est authentifié.                                                                                                                                                                                                                                                                                                                     |
| <b>Scénario alternatif</b> | 4.1 Le système indique que l'utilisateur non autorisé.                                                                                                                                                                                                                                                                                            |
| <b>Exception</b>           | Utilisateur non authentifié.                                                                                                                                                                                                                                                                                                                      |

Tableau III.3 : Description textuelle du diagramme de séquence « Authentification »

### 3.3.3 Diagramme de séquence « Publier annonces »

Le **tableau III.4** décrit le diagramme de séquence “**Publier annonces**” illustré par la **Figure III.12**.

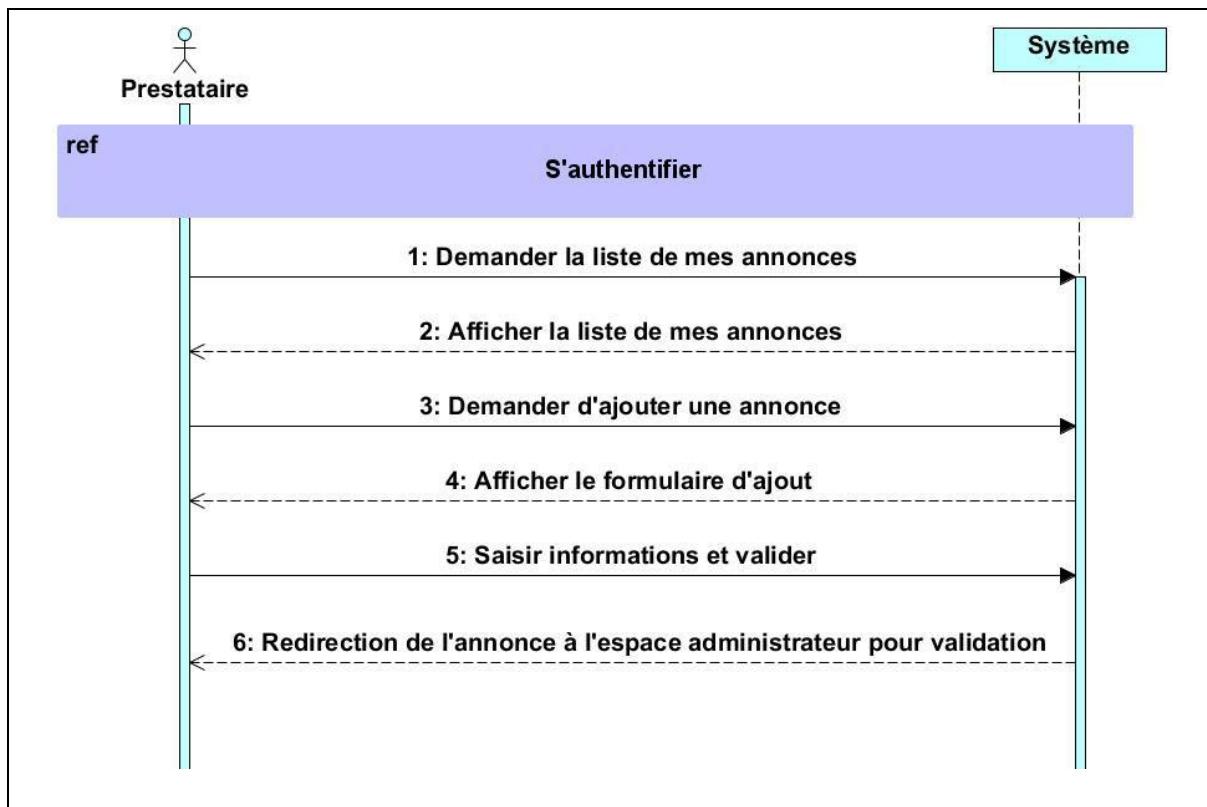


Figure III.12 : Diagramme de séquence « Publier annonces »

|                          |                                                                                                                                                                                                                                                                                                                |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cas d'utilisation</b> | Publier annonces                                                                                                                                                                                                                                                                                               |
| <b>Acteur</b>            | Prestataire                                                                                                                                                                                                                                                                                                    |
| <b>Intérêts</b>          | Il permet au prestataire de publier une annonce.                                                                                                                                                                                                                                                               |
| <b>Précondition</b>      | Prestataire authentifié                                                                                                                                                                                                                                                                                        |
| <b>Scénario nominal</b>  | <ol style="list-style-type: none"> <li>Le prestataire demande la publication d'une annonce.</li> <li>Le système affiche le formulaire de création d'annonce.</li> <li>Le prestataire remplit le formulaire et valide la publication</li> <li>Le système envoie l'annonce à l'espace administrateur.</li> </ol> |
| <b>Post condition</b>    | Nouvelle annonce ajoutée.                                                                                                                                                                                                                                                                                      |
| <b>Exception</b>         | Le système affiche un message d'erreur indiquant que le formulaire de publication d'annonce contient des champs vides.                                                                                                                                                                                         |

Tableau III.4 : Description textuelle du diagramme de séquence « Publier annonces »

### 3.3.4 Diagramme de séquence « Modifier annonces »

Le **tableau III.5** décrit le diagramme de séquence “**Modifier annonces**” illustré par la **Figure III.13**.

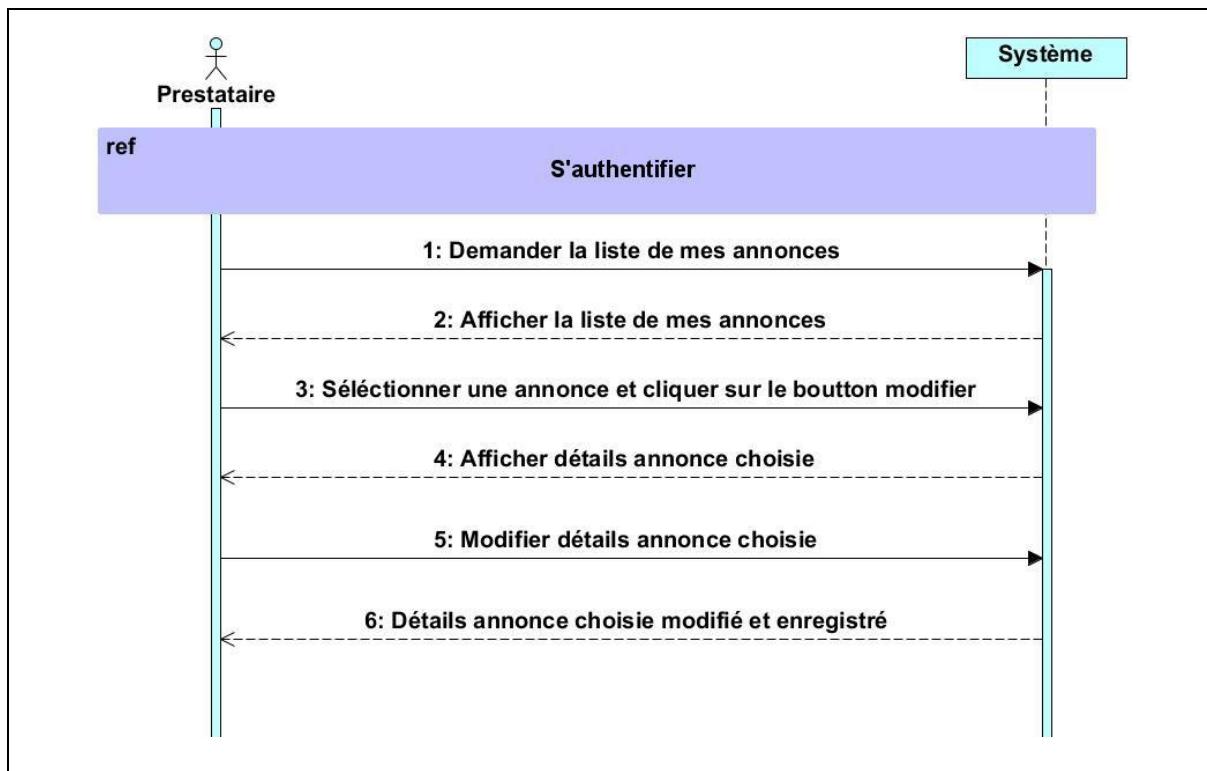


Figure III.13 : Diagramme de séquence « Modifier annonces »

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cas d'utilisation</b> | Modifier annonces                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Acteur</b>            | Prestataire                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Intérêts</b>          | Il permet au prestataire de modifier une annonce.                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Précondition</b>      | Prestataire authentifié                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Scénario nominal</b>  | <ol style="list-style-type: none"> <li>1. Le prestataire demande la liste de ses annonces publiées.</li> <li>2. Le système affiche les annonces publiées du prestataire.</li> <li>3. Le prestataire choisit une annonce et clique sur le bouton modifier.</li> <li>4. Le système affiche les détails de l'annonce.</li> <li>5. Le prestataire modifie l'annonce concernée.</li> <li>6. Le système modifie l'annonce.</li> </ol> |
| <b>Post condition</b>    | La liste des annonces d'abonné est mise à jour.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Exception</b>         | Le système affiche un message d'erreur indiquant que le formulaire de modification d'annonce contient des champs vides.                                                                                                                                                                                                                                                                                                         |

Tableau III.5 : Description textuelle du diagramme de séquence « Modifier annonces »

### 3.3.5 Diagramme de séquence « Supprimer annonces »

Le **tableau III.6** décrit le diagramme de séquence “**Supprimer annonces**” illustré par la **Figure III.14**.



Figure III.14 : Diagramme de séquence « Supprimer annonces »

|                          |                                                                                                                                                                                                                                              |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Cas d'utilisation</b> | Supprimer annonces                                                                                                                                                                                                                           |
| <b>Acteur</b>            | Prestataire                                                                                                                                                                                                                                  |
| <b>Intérêts</b>          | Il permet au prestataire de supprimer une annonce.                                                                                                                                                                                           |
| <b>Précondition</b>      | Prestataire authentifié                                                                                                                                                                                                                      |
| <b>Scénario nominal</b>  | 1. Le prestataire demande la liste de ses annonces publiées.<br>2. Le système affiche les annonces publiées du prestataire.<br>3. Le prestataire choisit une annonce et clique sur le bouton supprimer.<br>4. Le système supprime l'annonce. |
| <b>Post condition</b>    | La liste des annonces d'abonné est mise à jour.                                                                                                                                                                                              |
| <b>Exception</b>         | Le système affiche un message d'erreur indiquant que le formulaire de modification d'annonce contient des champs vides.                                                                                                                      |

Tableau III.6 : Description textuelle du diagramme de séquence « Supprimer annonce

### 3.4 Diagrammes d'interaction

Le diagramme de séquence dynamique, appelé aussi diagramme d'interaction en conception, permet de représenter les interactions d'objets dans le cadre d'un diagramme des cas d'utilisation selon un ordre chronologique. Nous présentons, dans la suite, quelques diagrammes d'interaction.

### 3.4.1 Diagramme d'interaction « S'authentifier »

Avant tout accès au système, l'utilisateur est obligé à s'identifier. Cette identification est importante car elle permet de classifier les utilisateurs et déterminer leurs rôles afin de déterminer les priviléges de chaque utilisateur. L'utilisateur saisit son login et le mot de passe dans un formulaire qui doivent être compatibles avec les données existantes dans la base de données. Si le compte est existé, le système crée un jeton dépendant de son nom utilisateur et rôle Sinon un message d'erreur apparaît, alors l'utilisateur doit réessayer, voir **Figure III.15**

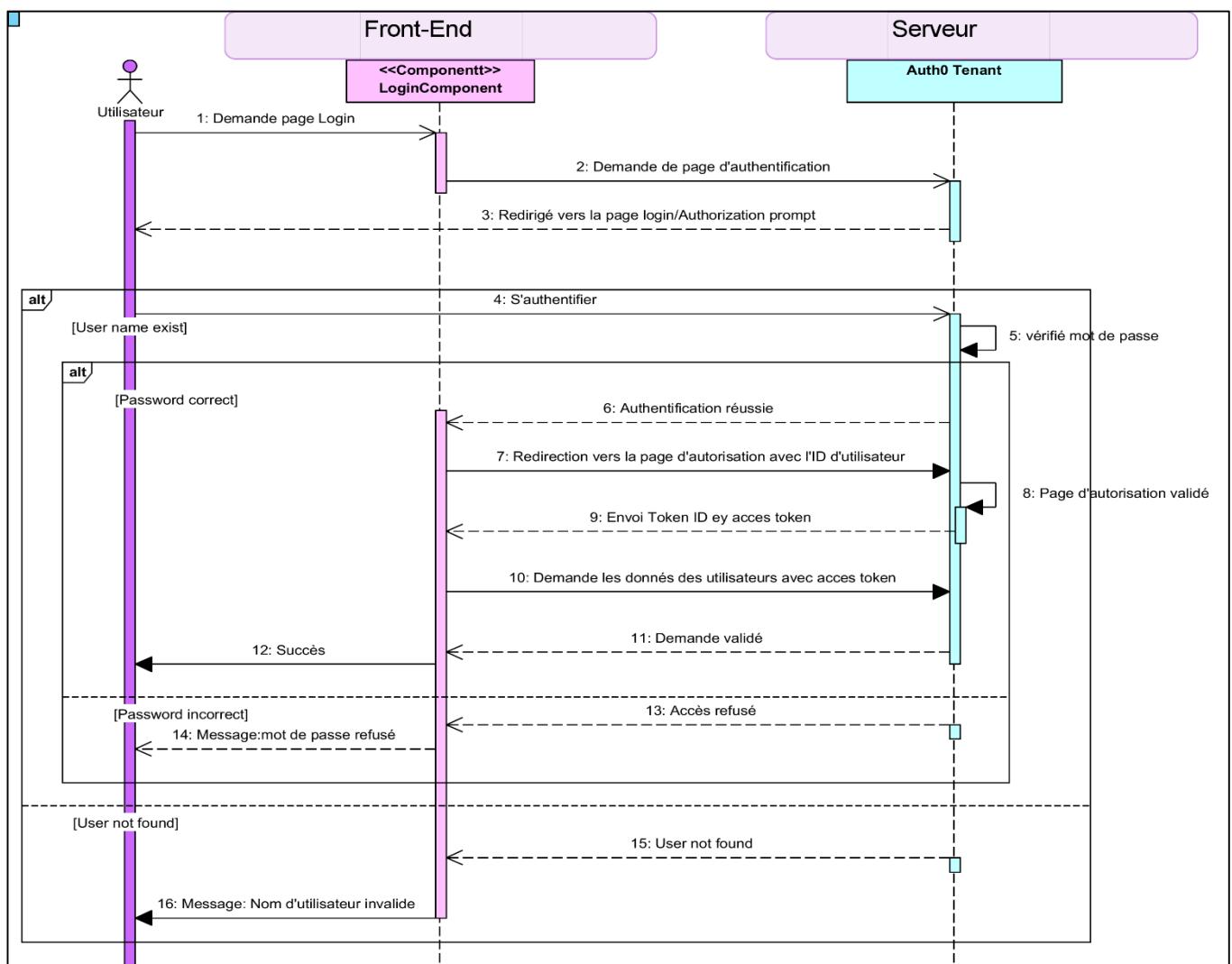
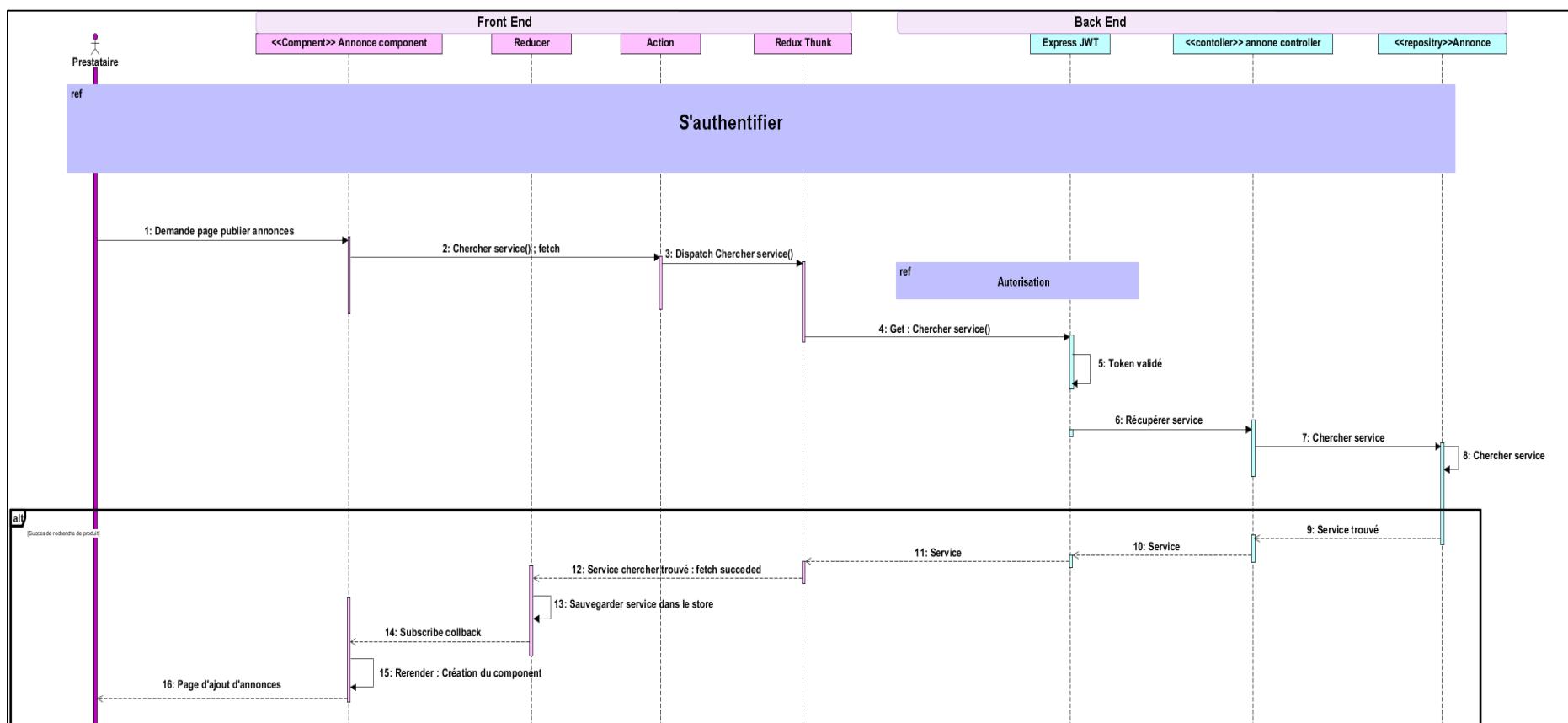
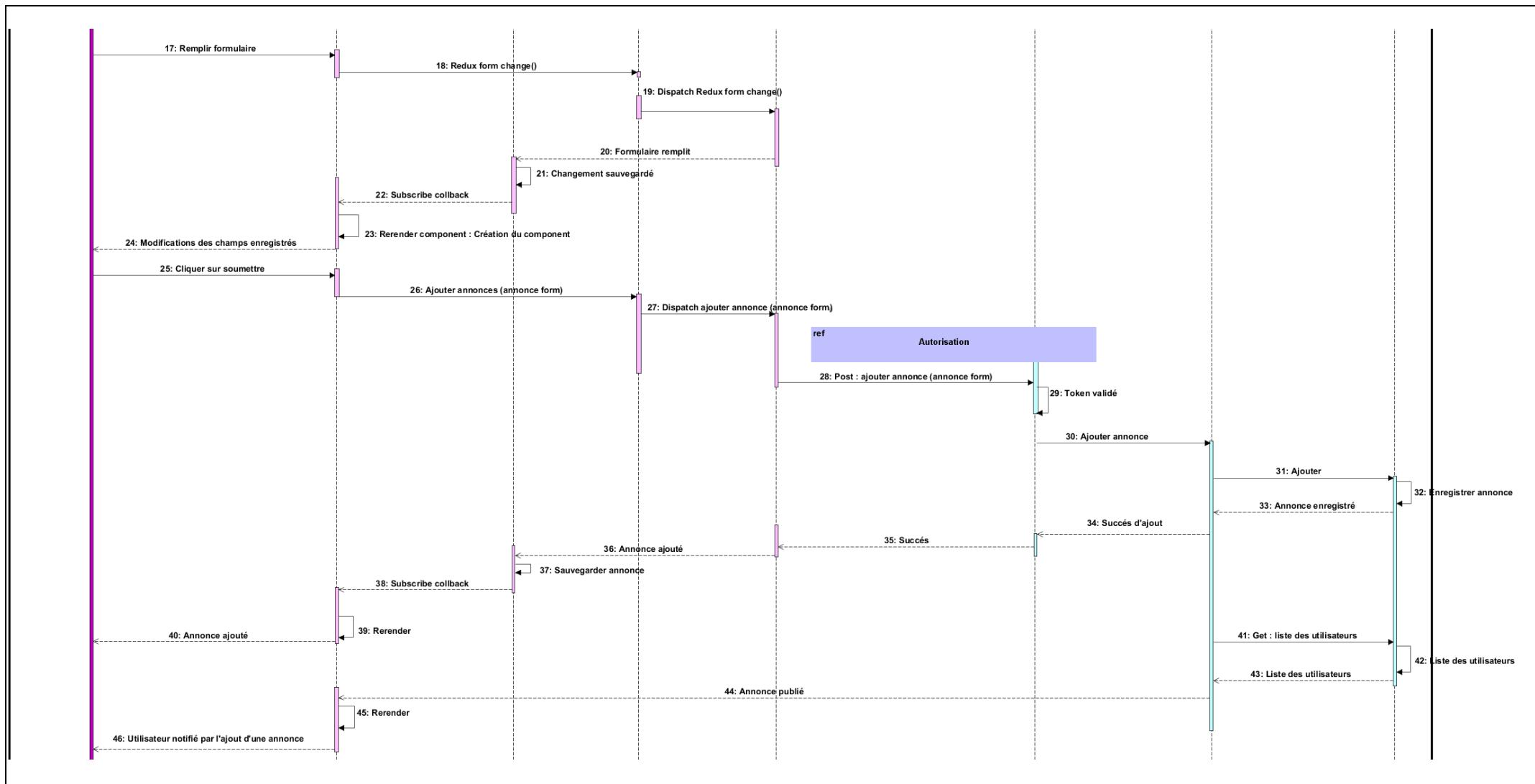


Figure III.15 : Diagramme d'interaction « S'authentifier »

### 3.4.2 Diagramme d'interaction « Publier une annonce »

Le Prestataire interagit avec la vue « Annonce Component ». Il demande la création d'une annonce, il attend jusqu'à ce que la vérification de l'existence de son service est validé, puis il remplit le formulaire affiché et valide. Sa demande est transmise à la partie Back-end qui vérifie l'autorisation à l'aide d'Express JWT. Ensuite, la demande est passée vers le contrôleur « annonceControleur » et puis à la base de données « annonce » qui ajoute l'annonce et renvoie la liste de tous les utilisateurs pour l'informer d'une nouvelle annonce qui a été publiée, voir les **Figures III.16**.





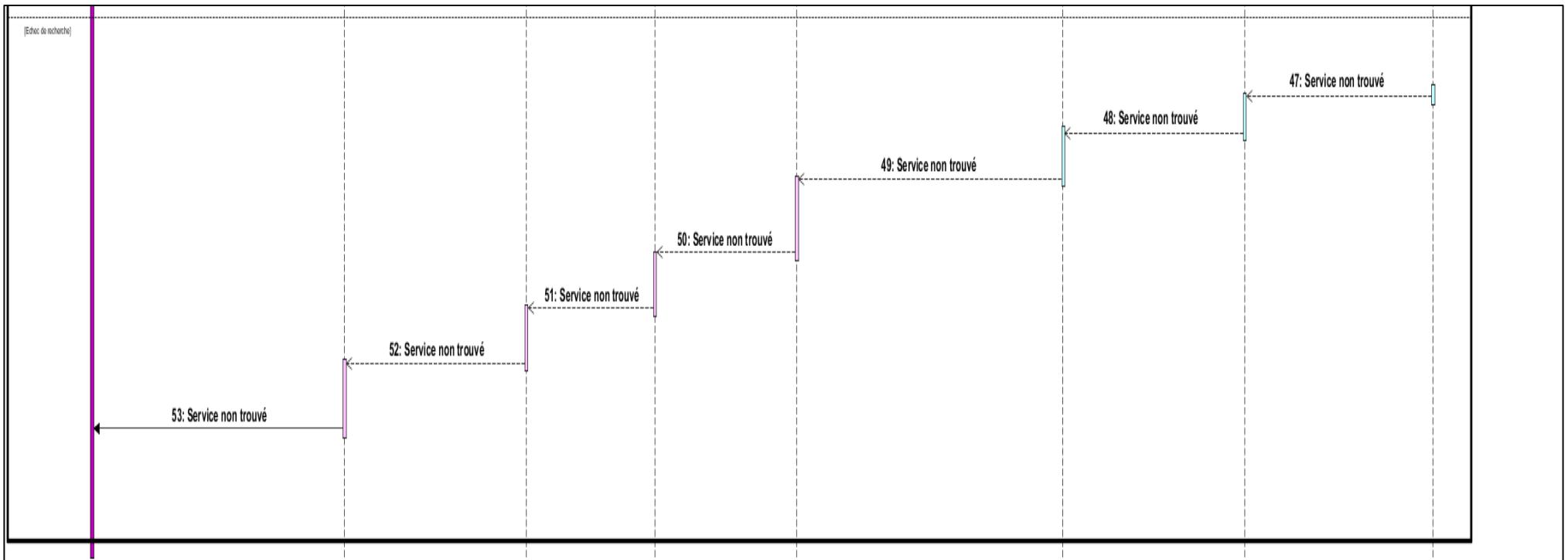


Figure III.16 : Diagramme d'interaction « Publier une annonce »

### 3.5 Diagramme de classe

Un diagramme de classe dans le langage UML est un type de diagramme de structure statique qui décrit la structure d'un système en montrant le système de classes, leurs attributs, les opérations (ou) les méthodes et les relations entre les classes.

La **Figure III.17** ci-dessous illustre bien les diagrammes de classe de notre système selon le modèle MVC :

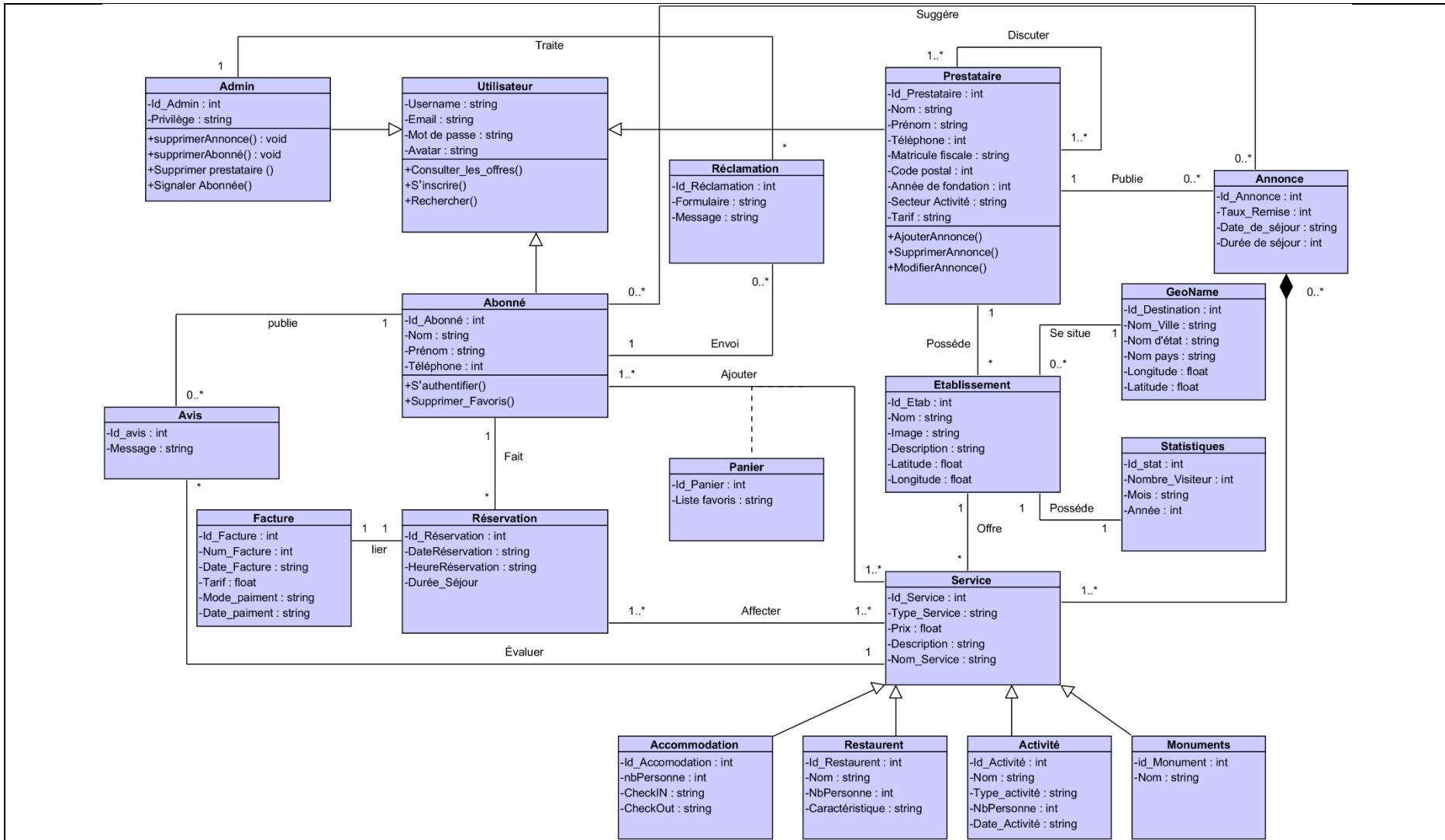


Figure III.17 : Diagramme de classe

### 3.6 Diagramme de composants

Dans le langage UML, un diagramme de composants décrit la manière dont les composants sont câblés ensemble pour former des composants plus importants ou des systèmes logiciels. Ils sont utilisés pour illustrer la structure de systèmes arbitrairement complexes, voir **Figure III.18**.

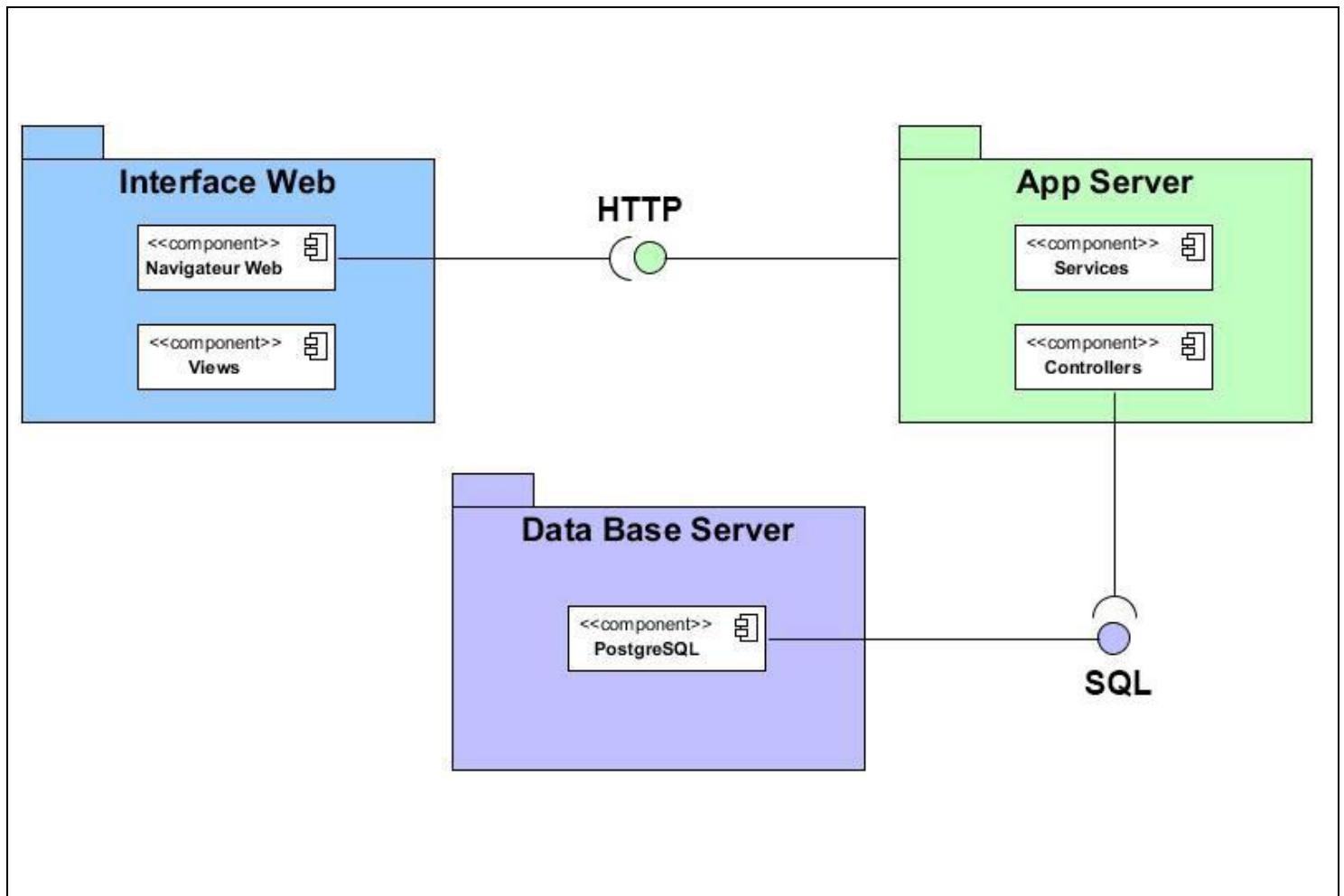


Figure III.18 : Diagramme de composants

### Conclusion

Durant ce chapitre, nous avons présenté à la fois la conception détaillée et la base de données de notre application Web. A ce stade, nous avons terminé une phase très importante qui est la conception.

C'est ainsi qu'on peut maintenant aborder la partie réalisation de l'application. Dans le chapitre suivant, nous allons parler en détails de cette phase.

# CHAPITRE IV : REALISATION

## Introduction

La phase qui suit une conception est éventuellement l'implémentation. Après les trois chapitres précédents, nous arrivons à rassembler nos informations et réaliser notre système, nous commençons ce chapitre par décrire l'environnement de développement matériel et logiciel que nous avons adoptés. Nous présentons ensuite quelques captures d'écrans du site web.

## 1 Environnement de développement

Dans cette partie, nous présentons les différents outils matériels et logiciels nécessaires pour le développement de notre application.

### 1.1 Environnement matériel

Pour la réalisation de ce projet on a disposé des équipements suivants :

- Un premier ordinateur :
  - Marque : ASUS ROG (Republic Of Gamers)
  - Processeur : Intel® Core™ i5-7300HQ CPU @ 2.50GHz X 2
  - Mémoire installée (RAM) : 12.0 Go
  - Type du système : SE 64bits, processeur x64
  - Disque dur : 1 To
  - Système d'exploitation : Windows 10
- Un deuxième ordinateur :
  - Marque : ASUS
  - Processeur : Intel® Core™ i7-6500HQ CPU @ 2.50GHz 2.59GHz
  - Mémoire installée (RAM) : 8.0 Go
  - Disque dur : 1 To
  - Système d'exploitation : Windows 10

## 1.2 Environnement logiciel

Cette section décrit l'environnement logiciel avec lequel nous avons réalisé ce projet.

| <b>Pour la conception</b>                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | Est un logiciel de retouche, de traitement et de dessin assisté par ordinateur édité par Adobe. Il est principalement utilisé pour les traitements des photographies numériques. Photoshop possède son propre format de projet (PSD), qui est plus qu'un simple format de fichier. Le programme accepte également d'importer et d'exporter des fichiers d'image dans les formats les plus courants (GIF, JPEG, TIFF, PNG, ILBM, etc.). |
|  | Est un logiciel de création de diagrammes dans le cadre d'une conception. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en UML.                                                                                                                                                                                                                                                            |
| <b>Pour le développement</b>                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|  | Est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code, open source et gratuit.                                                                                                                                                                                                                                                                                                               |

|                                                                                                     |                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <p>Dans notre projet nous avons suivi un workflow git pour organiser les versions de développement donc nous avons mis en place deux branches « master » et « dev » et nous avons relié la repository avec le serveur web de telle sorte qu'après chaque modification on obtient un mis à jour dans le serveur.</p> |
|  <b>POSTMAN</b>    | <p>Est un éditeur de test, l'une des fonctionnalités les plus puissantes de cet éditeur est sa capacité d'exécuter des tests automatisés sur vos demandes.</p>                                                                                                                                                      |
| <b>Pour la gestion du projet</b>                                                                    |                                                                                                                                                                                                                                                                                                                     |
|                   | <p>Est un outil de gestion de projet en ligne. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches.</p>                                                                                                                                                    |
| <b>Pour les statistiques</b>                                                                        |                                                                                                                                                                                                                                                                                                                     |
|  <b>Power BI</b> | <p>Est une solution de Business Intelligence développée par Microsoft pour permettre aux entreprises d'agréger, d'analyser et de visualiser les données en provenance de sources multiples.</p>                                                                                                                     |

Tableau IV.1 : Environnement logiciel

## 2 Présentation de la solution web

Dans cette partie, nous allons exposer quelques interfaces de notre application web en essayant de présenter les fonctionnalités les plus importantes dans notre travail.

## 2.1 Interface d'accueil associé à l'administrateur

La **Figure IV.1** représente l'interface d'accueil du l'internaute.

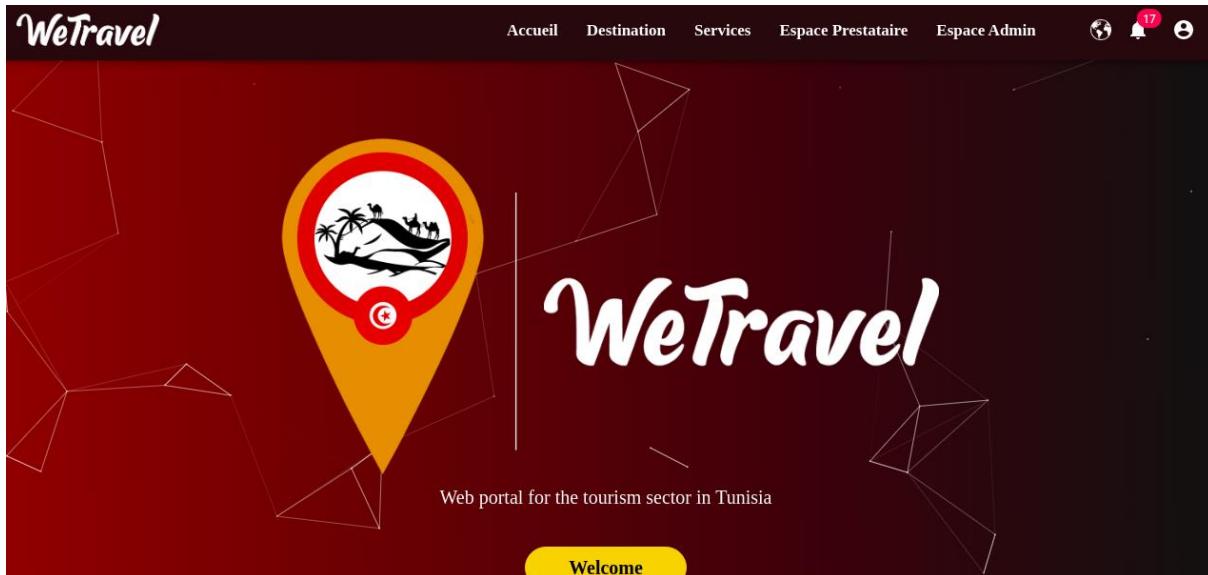


Figure IV.1 : Interface d'accueil associé à l'administrateur

## 2.2 Interface de l'espace administrateur pour la gestion des utilisateurs

À partir de cette interface, **Figure IV.2**, l'administrateur peut consulter la liste des abonnés au site, il peut voir le profil d'un abonné spécifique, le supprimer du site ou le bloquer.

A screenshot of the Auth0 user management interface. The top navigation bar includes the Auth0 logo, search bar, and links for Help &amp; Support, Docs, and Discuss Your Needs. On the left, a sidebar lists various management sections: Getting Started, Activity, Applications, APIs, SSO Integrations, Connections, Universal Login, Users &amp; Roles (which is currently selected), Rules, Hooks, and Multifactor Auth. The main content area is titled "Users" and contains a sub-header: "An easy to use UI to help administrators manage user identities including password resets, creating and provisioning, blocking and deleting users. Learn more ▶". It features a search bar, a "Search by User" dropdown, and a "RESET" button. Below this, a table lists users with columns for Name, Connection, Logins, and Latest Login. Two entries are shown: "Mohamed" connected via "google-oauth2" with 1 login at "3 minutes ago", and "sarra.benabdada@gmail.com" connected via "Username-Password-Authen..." with 1 login at "7 minutes ago".

Figure IV.2 : Interface de l'espace administrateur pour gestion des utilisateurs

### 2.3 Interface d'inscription

Pour devenir abonné, vous devez créer un profil sur le site. L'utilisateur est invité à remplir le formulaire d'inscription, il peut se connecter soit avec son Facebook soit avec son Gmail. Le formulaire génère une erreur dans le cas où le mot de passe ne contient pas les critères nécessaires, voir **Figure IV.3.**

**Welcome**

Log in to dev-is50usc3a to continue to WeTravel.

Email address

Password  (

[Forgot password?](#)

**Continue**

---

OR

Continue with Google

Continue with Facebook

**Welcome**

Sign Up to dev-is50usc3a to continue to WeTravel.

Email address

Password  (

Your password must contain:

- ✓ At least 8 characters
- ✓ At least 3 of the following:
  - ✓ Lower case letters (a-z)
  - ✓ Upper case letters (A-Z)
  - ✓ Numbers (0-9)
  - ✓ Special characters (ex. !@#)

**Continue**

Figure IV.3 : Interface d'inscription

## 2.4 Interface d'authentification

Avant d'accéder au site, l'abonné ou le prestataire doit s'authentifier en saisissant son email et son mot de passe dans le formulaire de connexion,

**Figure IV.4.** Le formulaire génère une erreur dans le cas où les champs obligatoires sont vides ou bien l'utilisateur introduit des données erronées.

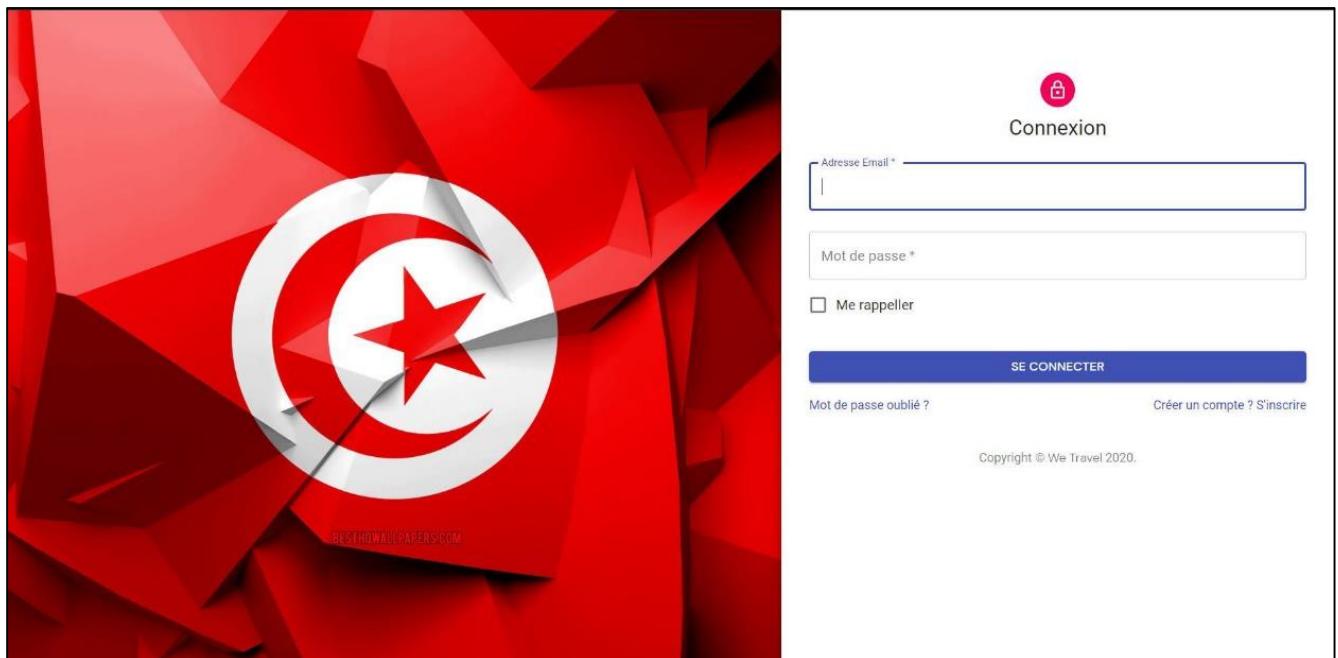


Figure IV.4 : Interface d'authentification

## 2.5 Interface d'accueil associé à l'abonné

La Figure IV.5 : représente la page d'accueil de notre site web « *WeTravel* ».

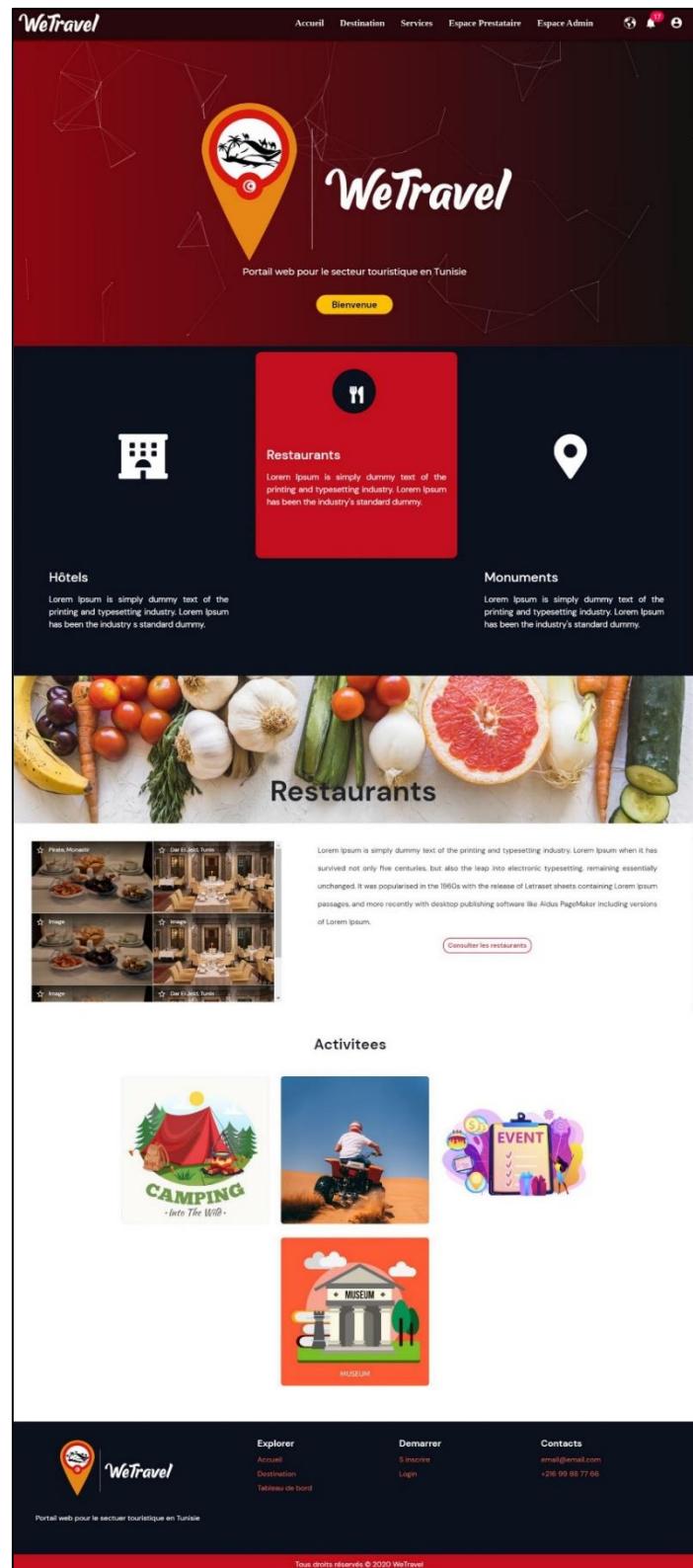


Figure IV.5 : Interface d'accueil associé à l'abonné

## 2.6 Interface destination

La **Figure IV.6** montre les services souhaités que l'abonné puisse choisir.

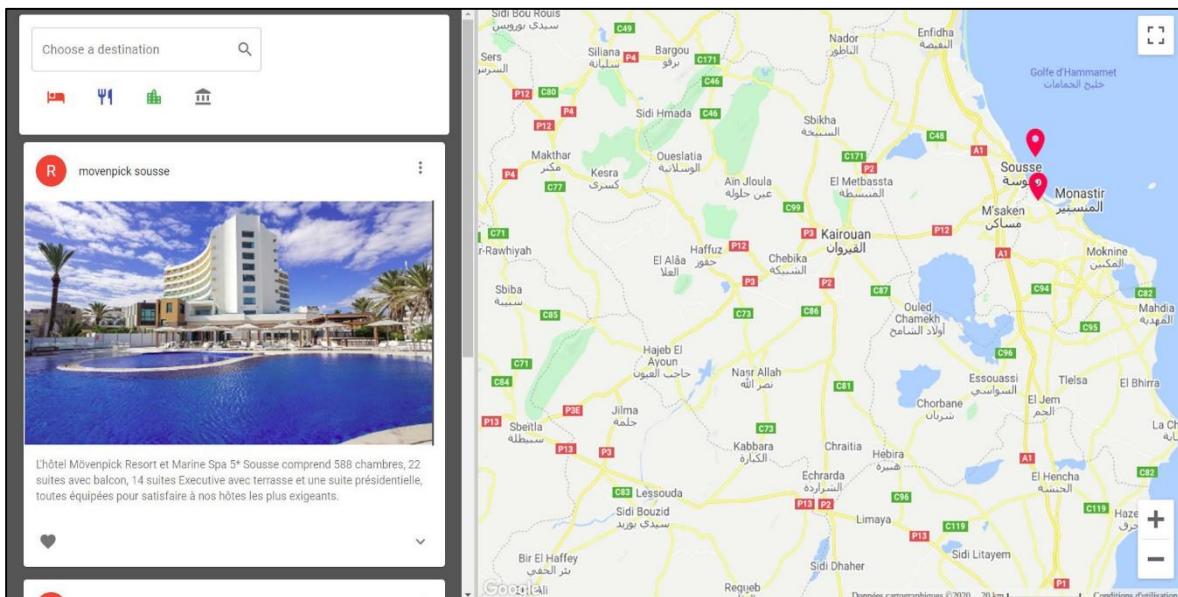


Figure IV.6 : Interface destination

## Conclusion

Afin de bien conclure cet avant dernier chapitre, nous avons présenté les différents éléments composant la réalisation de l'application. Cependant, nous devons mettre en évidence, que la réalisation n'est et ne sera jamais l'étape finale du processus de développement d'un bon logiciel. Il faut continuer à suivre et à superviser l'application à commencer par sa mise en exploitation avec des utilisateurs, dans le but de détecter les éventuels bugs et anomalies et de les rectifier. Ceci permet d'assurer la stabilité et la fiabilité du système.

# Conclusion Générale

---

Le présent rapport est réalisé dans le cadre de notre projet de fin d'études en vue de l'obtention du diplôme de Licence Fondamentale en Sciences de l'Informatique.

Ce travail a pour objectif de concevoir et réaliser un portail web pour le secteur touristique en Tunisie qui favorise la destination communautaire, attire les visiteurs et aide également à attirer de nouveaux prestataires, des investisseurs, des résidents et des opportunités dans la région.

Pour pouvoir compléter notre mission, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de cette application.

Le premier chapitre a été consacré au cadre général du projet. Nous avons commencé par une étude de l'existant où nous avons dégagé les insuffisances des solutions existantes et nous avons mis en évidence la mise en place de la méthodologie de travail 2TUP pour la réalisation de notre application web.

Le deuxième chapitre ayant pour but de spécifier les besoins fonctionnels, non fonctionnels, et les besoins techniques. Cette étape est la partie cœur pour la réussite du projet, et c'est pour cette raison qu'on fait une étude minutieuse pendant une période importante dans la spécification et l'analyse des besoins.

Une fois la phase de spécification des besoins est achevée, nous avons entamé la phase de conception.

Le quatrième chapitre a été dédié à l'aspect implémentation qui contient le choix de l'environnement de travail ainsi que les interfaces réalisés de la partie web.

Ce projet était bénéfique pour nous dans plusieurs sens. Il nous a aidé à s'ouvrir sur le monde professionnel qui implique de la responsabilité et de la discipline ainsi que du travail collaboratif d'équipe. Il a été une occasion pour améliorer nos connaissances dans les environnements de travail et de maîtriser les langages de programmation qui seront certes utiles au niveau professionnel.

# Bibliographie

1. En Tunisie, le redémarrage du tourisme mis à mal par le coronavirus. jeune afrique. [En ligne] 04 2020.  
<https://www.jeuneafrique.com/mag/913961/economie/en-tunisie-le-redemarrage-du-tourisme-mis-a-mal-par-le-coronavirus/>.
2. Tourisme international. Banque Mondiale. [En ligne] [Citation : 01 03 2020.]  
<https://donnees.banquemonde.org/indicateur/ST.INT.ARVL>.
3. Tourisme Tunisie. Lapresse. [En ligne] [Citation : 01 03 2020.]  
<https://lapresse.tn/42982/statistiques-du-secteur-du-tourisme-au-31-decembre-2019-la-culture-de-la-performance/>.
4. Siite officiel du Discover Tunisia. Discover Tunisia. [En ligne] [Citation : 03 03 2020.] <https://www.discovertunisia.com/>.
5. Site officiel du Tunisia Tourism. Tunisia Tourism. [En ligne] [Citation : 03 03 2020.]  
<https://www.discovertunisia.com/>.
6. Site officiel du Destination Tunisie. Destination Tunisie. [En ligne] [Citation : 03 03 2020.] <https://www.destinationtunisie.info/>.
7. Site officiel du Cityzeum. Cityzeum. [En ligne] [Citation : 03 03 2020.]  
<https://www.cityzeum.com/tourisme/tunisie>.
8. Saudi eVisa. Visit Saudi. [En ligne] [Citation : 04 03 2020.]  
<https://visa.visitsaudi.com/>.
9. SCRUM ET LA MÉTHODE AGILE : UNE INTRODUCTION. Wenovio. [En ligne] [Citation : 25 04 2020.] <https://www.wenovio.com/2017/12/scrum-methode-agile-introduction/>.
10. Processus unifié. wikipedia. [En ligne] [Citation : 25 04 2020.]  
[https://fr.wikipedia.org/wiki/Processus\\_unifi%C3%A9](https://fr.wikipedia.org/wiki/Processus_unifi%C3%A9).
11. <https://www.ganttproject.biz/>
12. Chugaev, Alex. « <Https://Kruschecompany.Com/Angular-vs-React-vs-Vue-1/> ». <Https://Kruschecompany.Com/>, 22 May 2020, [http://www.gpro-consulting.com/?fbclid=IwAR0AJqpuhHvxjFWCFOrudK21kRIV8A\\_VwRsrg4i7YCHDMLxOF2Z7qFh3rc](http://www.gpro-consulting.com/?fbclid=IwAR0AJqpuhHvxjFWCFOrudK21kRIV8A_VwRsrg4i7YCHDMLxOF2Z7qFh3rc).
13. HOFF, Nordwin.  
« <Https://Soat.Developpez.Com/Tutoriels/Javascript/Architecture-Flux-Avec-React/> ». <Soat.Developpez.Com>, 18 Jan. 2017.
14. <Https://Www.Youtube.Com/Watch?V=r7kc3CiauDg>. 10 Mar. 2017.

15. Chahine, H. (2018, 06). REACTJS VS ANGULAR VS VUEJS : QUE CHOISIR EN 2018 ?. ambient-it. Récupérée 04, 2020, à partir de <https://www.ambient-it.net/reactjs-vs-angular-vs-vuejs-que-choisir-en-2018/>
16. Gor, Chintan. Django vs Laravel vs Node.Js: Which One You Should Opt For In 2020? 28 Juin, 2018.
17. December DB-Engines Database Ranking: SQLite Competes in Top 10.
18. Passwordless Authentication with SMS.  
<https://auth0.com/docs/connections/passwordless/guides/sms-otp>.