

REPUBLIQUE DU SENEGAL



Un peuple-Un but-Une foi

Ministère de l'Enseignement Supérieur et de la Recherche

Direction Générale de l'Enseignement Supérieur Privé

Institut Supérieur d'Informatique



Département : Génie Informatique

Mémoire de fin de cycle pour l'obtention de Licence professionnel

Option : GÉNIE LOGICIEL

Sujet :

**Conception et réalisation d'un site web de réservation
de voyage pour une agence de voyage.**

Présenté par :

M. Soudaiss ELFAYADINE

Sous la direction de :

M. Ibrahima LO

Spécialité : Ingénierie Logiciel

Année académique : 2024-2025

PUBLIQUE DU SENEGAL



Un peuple-Un but-Une foi

Ministère de l'Enseignement Supérieur et de la Recherche

Direction Générale de l'Enseignement Supérieur Privé

Institut Supérieur d'Informatique



Département : Génie Informatique

Mémoire de fin de cycle pour l'obtention de Licence professionnel

Option : GÉNIE LOGICIEL

Sujet :

Conception et réalisation d'un site web de réservation de voyage pour une agence de voyage

Présenté par :

M. Soudaiss ELFAYADINE

Sous la direction de :

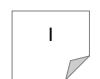
M. Ibrahima LO

Spécialité : Ingénierie Logiciel

Année académique : 2024-2025

Dédicaces

Pour mon père, tu as toujours été mon pilier, ma source de force et de motivation. Tes sacrifices et ton soutien constant m'ont permis d'avancer avec confiance et détermination dans chaque étape de mon parcours. Ce travail est le reflet de tes encouragements et de la foi que tu as toujours eue en moi. Cette modeste dédicace soit l'expression de ma profonde reconnaissance.



Remerciements

Je souhaite exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

Je remercie également l'ensemble des enseignants et encadrants de ma formation pour les connaissances qu'ils m' ont transmises et pour leur accompagnement académique.

Mes pensées vont également à ma famille, qui ma ont soutenus avec patience et encouragements tout au long de cette aventure. Leur soutien inébranlable a été une véritable source de motivation.

Enfin, je souhaite exprimer ma gratitude envers mes amis et collègues, qui ont partagé avec moi cette période exigeante, en m'apportant leur aide, leurs encouragements et des moments de convivialité essentiels à notre équilibre.

Merci à tous pour votre précieuse contribution à la réussite de ce travail.

LISTES DES FIGURES

<i>Figure 1 : Diagramme de cas d'utilisation globale</i>	10
<i>Figure 2: Diagramme de cas d'utilisation Reservation d un voyage</i>	11
<i>Figure 3: Diagramme de cas d'utilisation gestion des offres.....</i>	11
<i>Figure 4: Diagramme de classe</i>	12
<i>Figure 5: Diagramme de sequence Connexion d'un utilitaire.....</i>	13
<i>Figure 6: Diagramme de sequence Faire une reservation</i>	14
<i>Figure 7: page de selection des preferences.....</i>	16
<i>Figure 8: Page de finalisation de réservation</i>	16
<i>Figure 9 : page de connexion</i>	17
<i>Figure 10: page d'inscription.....</i>	18
<i>Figure 11 : page du dashboard Administrateur</i>	18
<i>Figure 12: Repertoire Github</i>	19
<i>Figure 13: Conteneur Docker.....</i>	19
<i>Figure 14: Interface de Sonarqube</i>	20
<i>Figure 15 : Diagramme de deploiement</i>	23

Sigles et acronymes

API : Application Programming Interface

MCD : Modèle Conceptuel de Données

SQL : Structured Query Language

MVC : Modèle-Vue-Contrôleur

CRUD : Create, Read, Update, Delete

UI/UX : User Interface / User Experience

JWT : JSON Web Token

REST : Representational State Transfer

ORM : Object-Relational Mapping

CI/CD : Continuous Integration / Continuous Deployment

PHP : Hypertext Preprocessor

JS : JavaScript

IDE : Integrated Development Environment

DBMS : Database Management System

PSQL : PostgreSQL Command Line Interface

API REST : Application Programming Interface Representational State Transfer

Laravel : Framework PHP pour le développement web

PostgreSQL : Système de gestion de base de données relationnelle

Angular : Framework JavaScript pour le développement front-end

Git : Système de gestion de version

ISI : Institut Supérieur D'Informatique

Table des matières

Dédicaces I

Remerciements II

LISTES DES FIGURES III

Sigles et acronymes IV

Résumé VII

Abstract VIII

INTRODUCTION GÉNÉRALE IX

PREMIÈRE PARTIE : CADRE THÉORIQUE ET MÉTHODOLOGIQUE 1

CHAPITRE 1 : LE CADRE THÉORIQUE 2

 1. Présentation de la structure d'accueil 2

 2. PRÉSENTATION DU SUJET 2

CHAPITRE II : LE CADRE MÉTHODOLOGIQUE 4

 SECTION 1 : MÉTHODE D'INVESTIGATION 4

 SECTION 2 : ÉTUDE COMPARATIVE DES MÉTHODES D'ANALYSE ET DE CONCEPTION 5

DEUXIEME PARTIE : TRAVAUX RÉALISÉS AU COURS DU PROJET 7

CHAPITRE I : ANALYSE 8

 SECTION 1 : PRÉREQUIS THÉORIQUES 8

 SECTION 2 : ANALYSE DU SYSTÈME 9

CHAPITRE II : CONCEPTION 12

 SECTION 1 : DIAGRAMMES DE CLASSE 12

 SECTION 2 : DIAGRAMMES DE SÉQUENCE 13

 SECTION 3 : TRAVAUX APPLICATIFS 14

 SECTION 4 : MAQUETISATION DE LA SOLUTION 16

 SECTION 5 : OUTIL DEVOPS 18

CHAPITRE III: IMPLÉMENTATION 20

 SECTION 1 : ENVIRONNEMENT 20

 SECTION 2 : ÉTUDE DE QUELQUES LANGAGES DE PROGRAMMATION 21

 SECTION 3 : ÉTUDE COMPARATIVE DE QUELQUES SGBD 21

 SECTION 4 : ARCHITECTURE APPLICATIVE 22

SECTION 5 : CHOIX DE LA SOLUTION DE PAIEMENT EN LIGNE	22
CHAPITRE IV : DÉPLOIEMENT	22
SECTION 1 : GESTION DE LA SÉCURITÉ.....	22
SECTION 2 : DIAGRAMME DE DÉPLOIEMENT.....	23
SECTION 3 :ESTIMATION DES COÛTS	23
TROISIEME PARTIE : BILAN.....	25
1-Objectifs atteints.....	26
2 - Objectifs non atteints.....	27
3 - Intérêts personnels.....	27
4- Perspectives d'évolution	27
Conclusion Générale.....	28
Bibliographie i	
Webographie ii	

Résumé

Ce projet vise à concevoir et réaliser un site web moderne pour une agence de voyage, afin de digitaliser la gestion des offres touristiques et de faciliter l'accès aux services pour les clients. Dans un contexte où la majorité des voyageurs utilisent Internet pour organiser leurs déplacements, de nombreuses agences traditionnelles peinent encore à proposer une plateforme en ligne complète, ce qui limite leur visibilité et leur compétitivité.

Pour atteindre cet objectif, une solution web a été développée en utilisant Laravel pour le backend et Angular pour le frontend, avec PostgreSQL comme système de gestion de base de données. L'application permet aux clients de consulter les offres disponibles, de réserver des voyages, de procéder à un paiement en ligne sécurisé et de recevoir des recommandations personnalisées basées sur leurs préférences et comportements.

La réalisation du projet s'est appuyée sur une analyse préalable des besoins fonctionnels et techniques, suivie par une phase de conception structurée, puis de développement et de déploiement. Les résultats attendus sont une interface ergonomique, sécurisée et performante, capable de répondre aux exigences des utilisateurs et d'optimiser les opérations de l'agence. Ce mémoire retrace les étapes de cette réalisation, depuis la conception théorique jusqu'à la mise en ligne de la plateforme.

Abstract

This project aims to design and develop a modern website for a travel agency, with the goal of digitizing the management of tourist offers and making services more accessible to clients. In a context where most travelers rely on the Internet to plan their trips, many traditional agencies still struggle to offer a comprehensive online platform, which limits their visibility and competitiveness.

To achieve this objective, a web solution was developed using Laravel for the backend and Angular for the frontend, with PostgreSQL as the database management system. The application allows clients to browse available offers, book trips, make secure online payments, and receive personalized recommendations based on their preferences and behavior.

The project was carried out following a thorough analysis of functional and technical requirements, followed by a structured design phase, development, and deployment. The expected outcomes include an ergonomic, secure, and high-performance interface that meets user needs and optimizes the agency's operations. This thesis outlines the steps taken throughout the project, from theoretical design to the launch of the platform.

INTRODUCTION GÉNÉRALE

L'industrie du tourisme traverse actuellement une révolution numérique sans précédent. Alors que plus de la majorité des voyageurs utilisent désormais Internet pour planifier leurs déplacements, de nombreuses agences de voyage traditionnelles peinent à suivre cette transformation digitale, particulièrement en Afrique de l'Ouest.

Aujourd'hui, il y a un vrai décalage : d'un côté, les clients veulent pouvoir réserver à toute heure de la journée, et de l'autre, beaucoup d'agences locales continuent à travailler presque uniquement de façon manuelle. Ce retard dans l'usage du numérique les rend moins compétitives face aux grandes plateformes internationales et les empêche de fidéliser une clientèle qui cherche surtout de la rapidité et de l'autonomie.

Le présent mémoire propose une réponse concrète à cette problématique à travers la conception et la réalisation d'une plateforme web complète destinée aux agences de voyage. Cette solution vise à démocratiser l'accès aux outils numériques en offrant un système intégré de gestion des offres touristiques, de réservation en ligne et de paiement sécurisé.

La première partie présente le cadre théorique et méthodologique. Elle expose d'abord l'Institut Supérieur d'Informatique, puis le contexte du projet, la problématique et les objectifs. Elle explique aussi pourquoi nous avons choisi la méthode UML pour analyser et concevoir le système.

La deuxième partie traite de l'analyse et de la conception. Elle montre comment nous avons étudié les besoins du système avec les diagrammes de cas d'utilisation. Ensuite, elle présente la conception avec les diagrammes de classe et de séquence, la création de la base de données et les maquettes de l'interface.

La troisième partie explique la réalisation et le déploiement du site web. Elle détaille les technologies utilisées, l'architecture choisie et la mise en ligne de la plateforme.

Une conclusion générale fait le bilan des objectifs atteints et propose des améliorations pour l'avenir.

PREMIÈRE PARTIE : CADRE THÉORIQUE ET MÉTHODOLOGIQUE

CHAPITRE 1 : LE CADRE THÉORIQUE

1. Présentation de la structure d'accueil

SUNRISE VOYAGE est une agence de voyage fondée en **2005** avec l'objectif d'accompagner les particuliers comme les entreprises dans l'organisation de leurs déplacements. À ses débuts, l'agence se consacrait principalement à la billetterie aérienne et à la réservation d'hôtels. Grâce à son sérieux et à la qualité de son service client, elle a élargi son offre à la gestion de voyages de groupe et le conseil personnalisé aux voyageurs.

Au fil des années, SUNRISE VOYAGE s'est adaptée aux évolutions du secteur touristique. Avec la montée en puissance du numérique et la généralisation des réservations en ligne, l'agence a initié une stratégie de modernisation afin d'intégrer des solutions digitales dans son fonctionnement. Cette transformation vise non seulement à simplifier les procédures pour les clients, mais également à rester compétitive face aux grandes plateformes internationales.

Aujourd'hui, l'agence est structurée autour de plusieurs services essentiels : la **billetterie**, le **tourisme**, la **réservation hôtelière** ainsi que l'**administration et la comptabilité**. Elle met un accent particulier sur l'innovation technologique afin d'améliorer l'expérience des voyageurs. C'est dans ce cadre que s'inscrit mon projet, qui consiste à concevoir une **plateforme de réservation en ligne** intégrant le **paiement électronique sécurisé** et des **recommandations personnalisées**, afin de renforcer la stratégie digitale de SUNRISE TRAVEL et de répondre aux besoins croissants des clients modernes.

2. PRÉSENTATION DU SUJET

2.1.Contexte

Avec la montée en puissance du digital, les agences de voyage doivent s'adapter aux nouveaux usages des clients, de plus en plus autonomes dans l'organisation de leurs séjours. Pourtant, beaucoup d'agences locales fonctionnent encore de manière traditionnelle, sans plateforme en ligne, ce qui limite leur visibilité et leur efficacité.

Ce projet vise à accompagner cette transformation numérique en développant un **site web dynamique** pour une agence de voyage. Il intègre les fonctions essentielles : **présentation des offres, réservation en ligne, paiement sécurisé, et suggestions personnalisées.**

2.2. Problématique

Comment permettre à une agence de voyage de taille modeste de rester compétitive face aux géants du secteur numérique ?

L'absence de solutions digitales limite la réactivité, la gestion client et les ventes. D'où le besoin d'un outil unique permettant de :

- Mettre en valeur les offres de manière attractive ;
- Gérer automatiquement les réservations ;
- Intégrer un système de paiement en ligne ;
- Proposer des recommandations selon les préférences du client.

2.3. Objectifs

Objectif général :

Développer un site web moderne, sécurisé et responsive pour une agence de voyage, capable de gérer efficacement les offres touristiques, les réservations, les paiements en ligne, ainsi que des recommandations personnalisées en fonction des préférences des utilisateurs.

Objectifs spécifiques :

- **Analyse des besoins fonctionnels et techniques** : Identifier les attentes de l'agence et des utilisateurs finaux afin de concevoir une solution adaptée, fiable et évolutive.
- **Conception d'une architecture logicielle adaptée** : Élaborer une structure technique robuste basée sur le framework **Laravel** pour le backend, **Angular** pour le frontend et **PostgreSQL** pour la base de données, garantissant performance et sécurité.
- **Développement d'une interface client intuitive** : Permettre aux utilisateurs de consulter les offres, effectuer des recherches, réserver un voyage et payer en ligne de manière simple et fluide.
- **Mise en place d'une interface administrateur** : Fournir aux responsables de l'agence un tableau de bord pour gérer les offres, les utilisateurs, les réservations et consulter des statistiques en temps réel.

- **Intégration d'un système de paiement en ligne sécurisé :** Utiliser Stripe et une solution locale (comme Wave ou Orange Money) pour offrir plusieurs options de paiement et renforcer la confiance des clients.
- **Implémentation d'un moteur de recommandations personnalisées :** Proposer aux utilisateurs des offres adaptées à leurs préférences ou à leurs réservations passées.
- **Déploiement et mise en ligne de la plateforme :** Héberger le site sur un serveur en ligne pour le rendre accessible à tout moment, sur tout type d'appareil.

2.4. Délimitation du Sujet

Ce mémoire se concentre uniquement sur la **partie technique** de la plateforme web.

Il couvre :

- Le développement du site pour navigateur web (pas d'application mobile) ;
- L'interface client pour les réservations et paiements ;
- L'interface admin pour la gestion de l'agence ;
- L'utilisation des technologies **Angular, Laravel, et PostgreSQL**.

Les aspects internes à l'entreprise (RH, comptabilité...) sont exclus de ce travail.

CHAPITRE II : LE CADRE MÉTHODOLOGIQUE

SECTION 1 : MÉTHODE D'INVESTIGATION

1. Hypothèses

- **H1 :** Une plateforme web de réservation améliore la rapidité et l'accessibilité du service pour les clients.
- **H2 :** L'intégration du paiement en ligne réduit les abandons de réservation.
- **H3 :** Une interface d'administration dédiée facilite la gestion des offres, des paiements et des utilisateurs.

2. Pertinence et Originalité du Sujet

Ce projet répond à un besoin réel de **digitalisation des agences de voyage**, notamment dans un contexte où **les habitudes de consommation évoluent vers l'autonomie en ligne**. Il se distingue par :

- Une **interface client** intuitive ;
 - Une **interface admin** pour gérer les offres, réservations, utilisateurs et paiements ;
 - Un système de **recommandations personnalisées** basé sur les préférences de l'utilisateur ;
 - Et l'intégration d'un **paiement en ligne sécurisé**, encore rare dans les agences locales.
- Ce projet combine **modernité, accessibilité et gain de productivité**.

3. Méthodologie de Recherche

La méthodologie adoptée est **itérative**, en s'inspirant des principes agiles :

- **Analyse des besoins** : entretiens avec des agents de voyage pour identifier les fonctionnalités clés côté admin et client.
- **Modélisation UML** : création de diagrammes pour clarifier l'architecture (cas d'utilisation, séquence, déploiement).
- **Développement technique** :
 - **Frontend (Angular)** : pages client (réservation, suggestions), pages admin (dashboard, gestion offres).
 - **Backend (Laravel)** : API REST pour gérer utilisateurs, offres, paiements.
 - **Base de données (PostgreSQL)** : stockage structuré des données clients, réservations et paiements.
- **Tests** : vérification fonctionnelle et retours utilisateurs sur l'ergonomie.

Cette approche a permis de construire une solution **fonctionnelle, modulable et adaptée aux besoins réels**.

SECTION 2 : ÉTUDE COMPARATIVE DES MÉTHODES D'ANALYSE ET DE CONCEPTION

1. UML (Unified Modeling Language)

UML est un **langage de modélisation orienté objet**, largement utilisé pour concevoir des systèmes logiciels. Il propose plusieurs types de diagrammes pour représenter :

- les **cas d'utilisation** (interactions utilisateurs - système),
- la **structure** (diagrammes de classes),
- le **comportement** (séquence),
- le **déploiement** (où et comment le système tourne).

UML est très adapté aux projets développés avec des langages objets comme Java, PHP ou C#. Il favorise la **modularité**, la **réutilisabilité** et l'**évolutivité**.

2. Merise

Merise est une méthode **plus ancienne**, surtout utilisée dans le monde francophone pour la conception des **bases de données** et des **systèmes d'information**. Elle se base sur trois niveaux d'analyse :

- **Conceptuel** (MLD, MCD),
- **Logique** (structure des données),
- **Physique** (implémentation en base).

Très forte pour le traitement des données, mais moins souple quand il s'agit de représenter l'interaction entre utilisateurs et système, ou de concevoir des interfaces et des composants web.

3. Choix de la méthode

Vu la nature orientée objet du projet (site Angular + API Laravel + base PostgreSQL), le choix de la méthode UML s'impose naturellement.

- UML permet de **modéliser à la fois la structure, le comportement et l'environnement d'exécution** de l'application.
- Elle s'intègre parfaitement à une architecture **MVC** et une approche orientée objet, comme utilisée dans Laravel.
- Les diagrammes UML (cas d'utilisation, classes, séquence, déploiement) facilitent la communication entre les membres de l'équipe, et aident à bien **structurer le code** et la **base de données**.

Conclusion : la méthode UML a été choisie pour sa **souplesse**, sa **complémentarité avec les technologies modernes** utilisées, et sa capacité à **représenter fidèlement les différents aspects du projet**.

DEUXIEME PARTIE : TRAVAUX RÉALISÉS AU COURS DU PROJET

CHAPITRE I : ANALYSE

SECTION 1 : PRÉREQUIS THÉORIQUES

1. Approche fonctionnelle

L'approche fonctionnelle est une méthode d'analyse qui consiste à décomposer le système en fonctions ou traitements à réaliser, sans se préoccuper de la structure des données ou de leur organisation dans le système. Elle est centrée sur **les actions** que le système doit accomplir pour satisfaire les besoins des utilisateurs. Dans cette approche, l'utilisateur est vu comme un acteur déclenchant des fonctions, et chaque fonction est représentée de manière indépendante.

Cette méthode est souvent utilisée avec des outils comme les **organigrammes**, les **diagrammes de flux** ou encore **Merise**. Toutefois, dans les projets modernes, notamment orientés web, cette approche atteint ses limites car elle ne facilite pas la réutilisation des composants ni la maintenance évolutive du système.

2. Approche orientée objet

Contrairement à l'approche fonctionnelle, l'approche orientée objet repose sur l'identification des **entités (ou objets)** du système, chacun regroupant à la fois **des données (attributs)** et **des comportements (méthodes)**. Cette approche permet de modéliser un système de manière plus proche de la réalité et favorise la **modularité**, la **réutilisabilité**, et une meilleure **maintenabilité** du code.

Dans le contexte de développement d'applications modernes utilisant des frameworks comme **Laravel** (backend PHP orienté objet) et **Angular** (frontend structuré en composants), l'approche orientée objet s'avère particulièrement adaptée. Elle permet notamment de créer des modèles de données robustes, de structurer le code métier, et d'assurer une cohérence globale entre les différentes couches du projet.

3. Concepts et formalisme de la méthode choisie

Le formalisme utilisé dans ce projet repose sur la méthode **UML (Unified Modeling Language)**, qui est un langage de modélisation standardisé, basé sur l'approche orientée

objet. UML permet de représenter graphiquement les différents aspects du système à travers plusieurs types de diagrammes, facilitant ainsi la compréhension, la conception et la documentation du projet.

Parmi les principaux diagrammes utilisés :

- **Le diagramme de cas d'utilisation** : pour décrire les interactions entre les acteurs et le système ;
- **Le diagramme de classes** : pour représenter la structure des objets et leurs relations ;
- **Le diagramme de séquence** : pour illustrer les échanges entre les objets dans le temps ;
- **Le diagramme de déploiement** : pour visualiser l'architecture physique du système (serveurs, composants, connexions).

Ce formalisme offre une vue claire et partagée du système, et s'aligne parfaitement avec les technologies employées dans le cadre de ce projet.

SECTION 2 : ANALYSE DU SYSTÈME

1. Recensement et Analyse Théorique

Cette étape consiste à identifier les besoins du système en se basant sur des recherches théoriques, des échanges avec les utilisateurs, ou une analyse de l'existant. Dans le cadre de ce projet, les besoins fonctionnels ont été recensés à partir des usages classiques observés dans une agence de voyage, notamment :

- La consultation des offres de voyage ;
- La réservation d'un voyage ;
- Le paiement en ligne ;
- La réception de suggestions personnalisées ;
- La gestion des offres, des clients et des réservations côté administrateur.

Ces besoins ont été structurés sous forme de cas d'utilisation, afin d'en faciliter la modélisation et la compréhension.

2. Description de la procédure

Le fonctionnement du système peut être résumé en deux grandes procédures principales :

- **Côté client** : un utilisateur accède au site web, consulte les offres disponibles, sélectionne un voyage, procède à la réservation, effectue un paiement sécurisé en ligne, et reçoit éventuellement des suggestions personnalisées selon ses préférences ou son historique.
- **Côté administrateur** : l'admin accède à un tableau de bord sécurisé pour gérer les offres disponibles (ajout, modification, suppression), consulter les réservations effectuées par les clients, et suivre les statistiques générales (nombre de réservations, paiements reçus, destinations populaires...).

3. Diagrammes de cas d'utilisation

Diagramme de cas d'utilisation global

Le diagramme ci-dessous représente de manière globale les principales interactions entre les acteurs du système (utilisateur final, administrateur, visiteur) et les différentes fonctionnalités proposées par la plateforme.

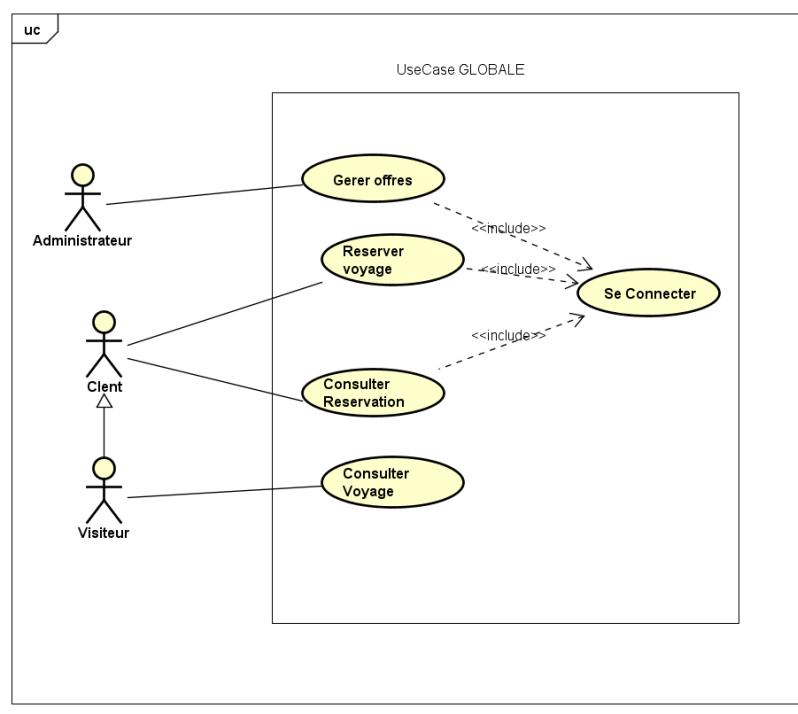


Figure 1 : Diagramme de cas d'utilisation globale

Description du diagramme

Le système comporte **trois types d'acteurs** principaux :

- **Le Visiteur** : utilisateur non authentifié, qui peut uniquement **consulter les offres de voyage** proposées par l'agence.

- **Le Client** : une fois connecté, il a la possibilité de **réserver un voyage, consulter ses réservations**, et interagir plus largement avec le système.
- **L'Administrateur** : responsable de la gestion de la plateforme, il peut **ajouter, modifier ou supprimer des offres de voyage** via la fonctionnalité "Gérer offres".

Cas d'utilisation : Réservation d'un voyage

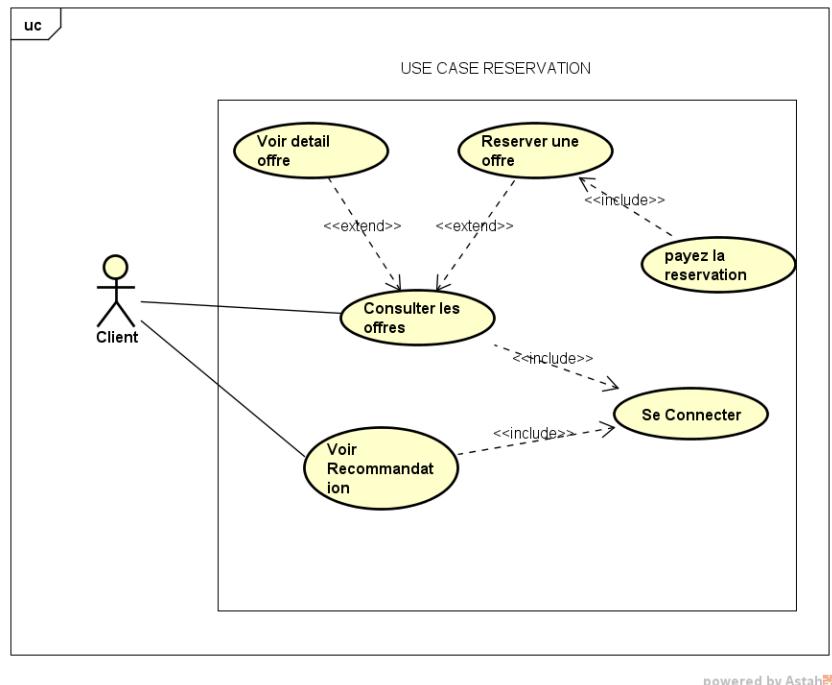


Figure 2: Diagramme de cas d'utilisation Reservation d'un voyage

Cas d'utilisation : Gérer les offres

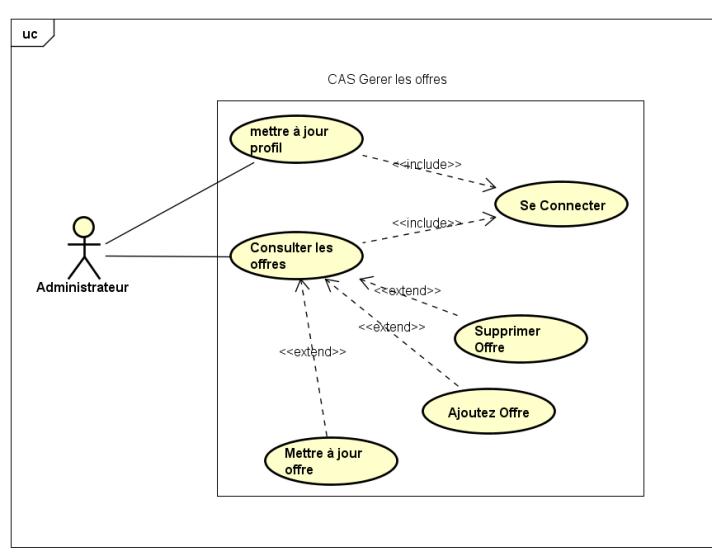
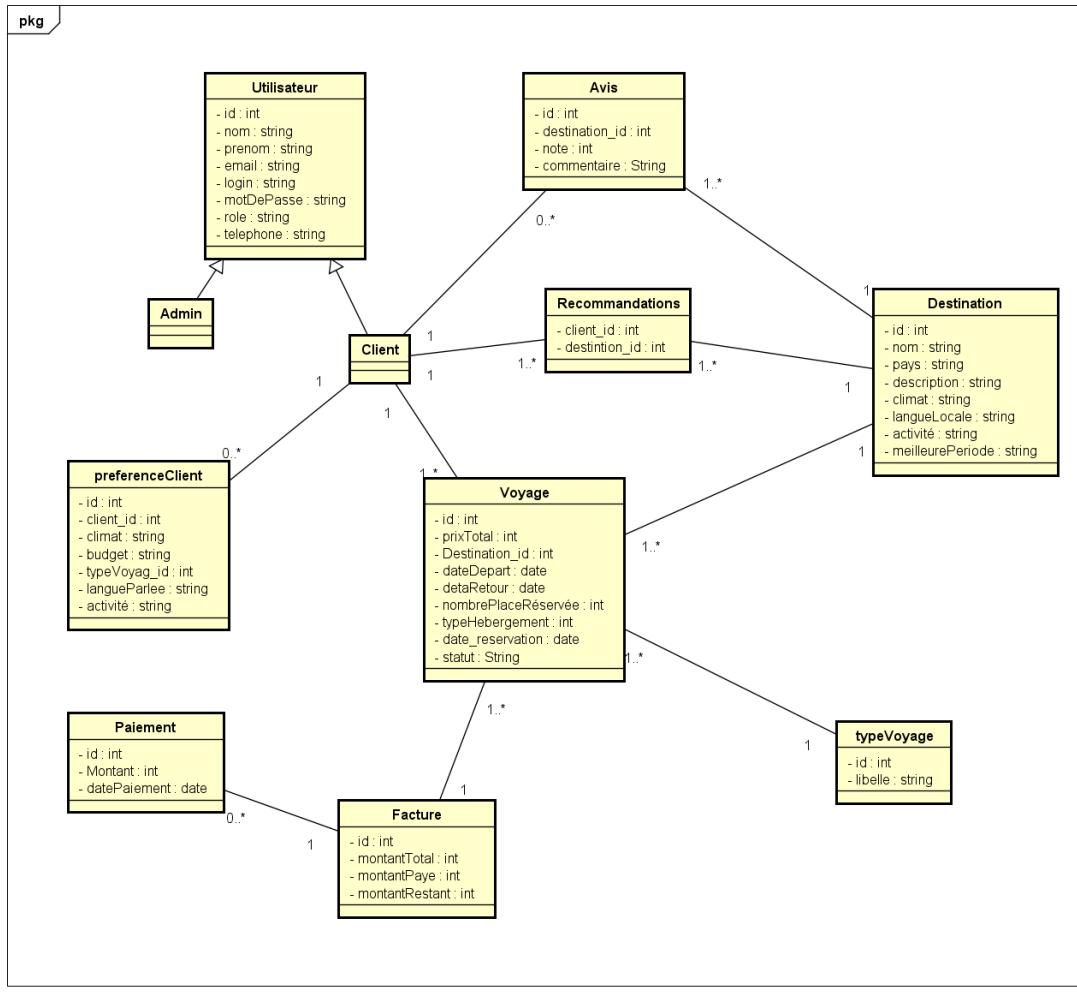


Figure 3: Diagramme de cas d'utilisation gestion des offres

CHAPITRE II : CONCEPTION

SECTION 1 : DIAGRAMMES DE CLASSE



powered by Astah

Figure 4: Diagramme de classe

Le diagramme de classe donne une vue d'ensemble sur la structure statique du système, en mettant en évidence les entités, leurs attributs, méthodes et les relations qui les lient.

Dans ce projet, les classes principales sont : Utilisateur, Client, Administrateur, Offre, Réservation, Paiement, Recommandation. La classe Client hérite d'Utilisateur et chaque réservation est liée à une offre précise.

Ce diagramme permet également de préparer la modélisation de la base de données et les entités du backend Laravel via Eloquent ORM.

SECTION 2 : DIAGRAMMES DE SÉQUENCE

- Présentation des diagrammes de séquence principaux

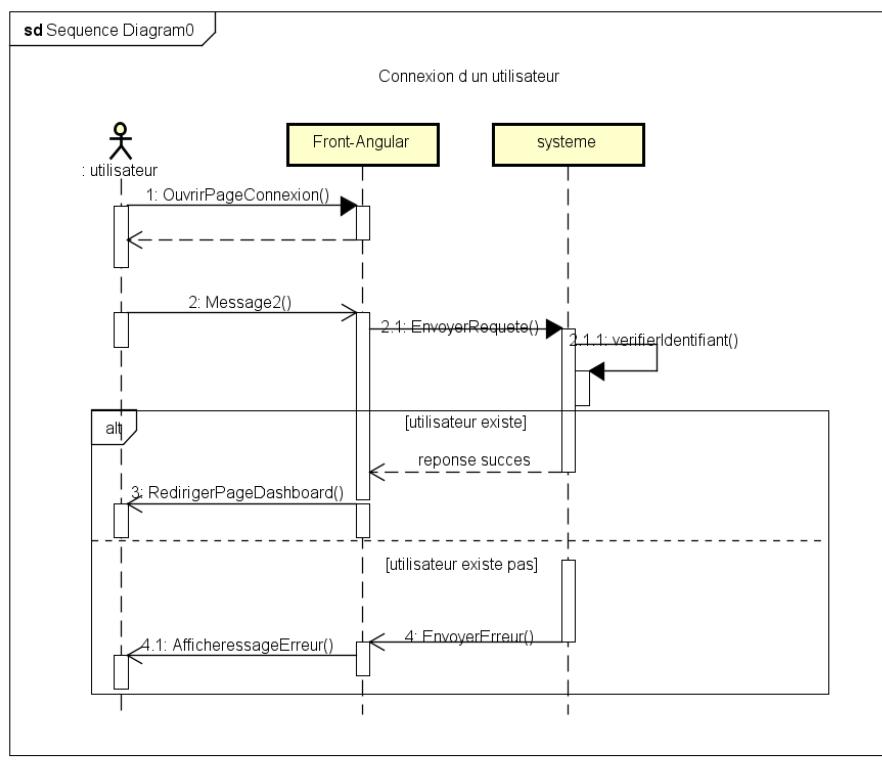


Figure 5: Diagramme de sequence Connexion d'un utilisateur

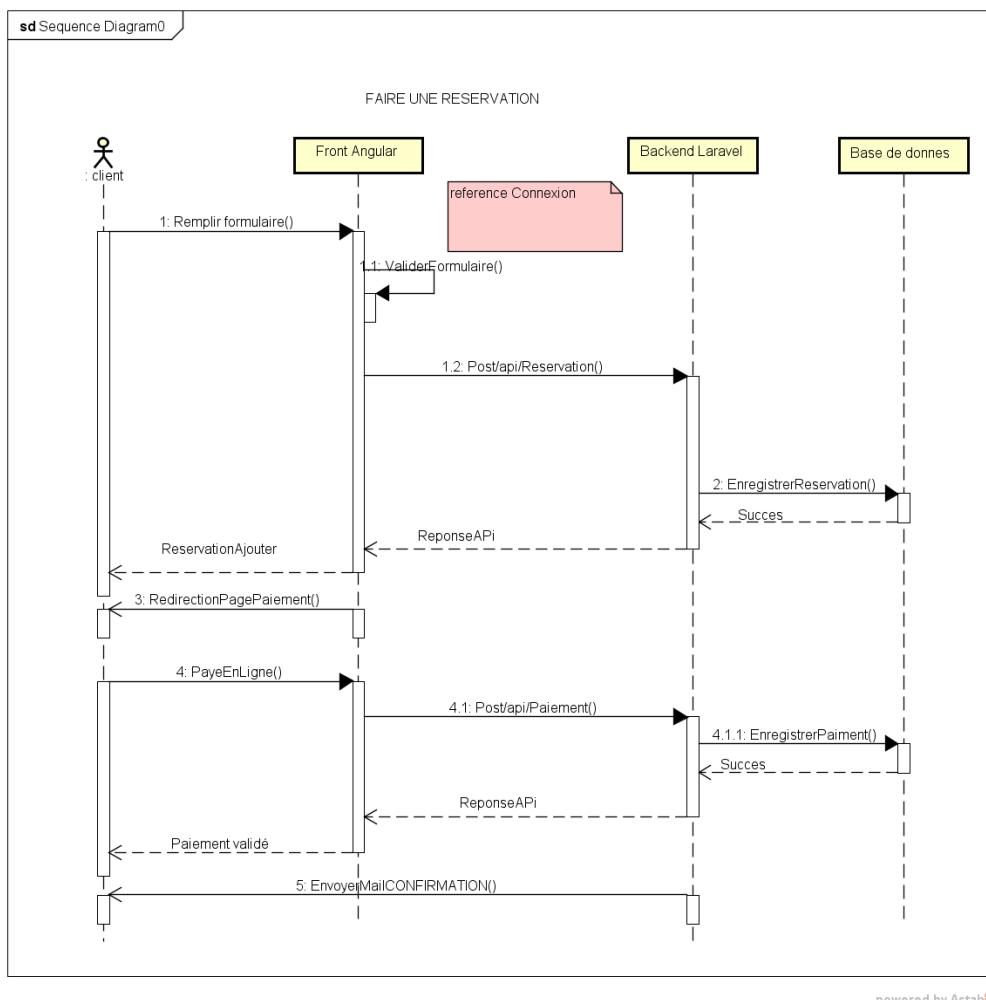


Figure 6: Diagramme de séquence Faire une réservation

Les diagrammes de séquence permettent de représenter l'enchaînement des messages entre les objets et acteurs du système au cours d'un scénario d'usage. Dans ce projet, les principaux cas étudiés concernent :

- Le processus de réservation d'un voyage ;
- L'authentification des utilisateurs ;

Ces diagrammes aident à prévoir la logique métier à implémenter et à structurer les appels entre les différents composants du backend et du frontend.

SECTION 3 : TRAVAUX APPLICATIFS

1)Création des CRUDs des services choisis

Gestion des offres :

- POST /api/offres : Créer une offre

- GET /api/offres : Lister les offres
- PUT /api/offres/{id} : Modifier une offre
- DELETE /api/offres/{id} : Supprimer une offre
- **Gestion des réservations :**
- POST /api/reservations : Créer une réservation
- GET /api/reservations/user/{id} : Réservations d'un client
- PUT /api/reservations/{id} : Modifier le statut d'une réservation
- GET /api/admin/reservations : Voir toutes les réservations

Gestion des clients :

- GET /api/utilisateurs: Liste des utilisateurs
- GET /api/utilisateurs/{id} : Détails d'un utilisateur
- PUT /api/utilisateurs/{id} : Modifier le profil d'un utilisateur
- DELETE /api/utilisateurs/{id} : Désactiver un compte utilisateur

Gestion des paiements :

- POST /api/paiements/stripe : Traiter un paiement via Stripe
- POST /api/paiements/wave : Traiter un paiement via Wave
- GET /api/paiements/history : Historique des paiements
- PUT /api/paiements/{id}/confirm : Confirmer un paiement

2) Crédation de la base de données

Tables créées :

- utilisateurs (id, nom, prénom, email, mot_de_passe, rôle, téléphone)
- voyages (id, description, prix_total, destination_id, date_départ, date_retour, nombre_places, type_hébergement, statut)
- reservations (id, utilisateur_id, voyage_id, date_réservation, statut, montant_total)
- paiements (id, réservation_id, montant, date_paiement, statut)
- préférences (id, utilisateur_id, climat, budget, type_voyage_id, langue_parlée, activité)
- recommandations (id, client_id, destination_id)

SECTION 4 : MAQUETISATION DE LA SOLUTION

Figure 7: page de selection des preferences

➤ **Description de la page sélection de préférence**

Cette interface permet à l'utilisateur de renseigner ses préférences de voyage (climat, zone, langue, budget, durée et activités). Ces informations servent de base au système pour générer des recommandations personnalisées.

Figure 8: Page de finalisation de réservation

➤ **Description de la page de finalisation de réservation**

Cette interface permet à l'utilisateur de renseigner ses préférences de voyage (climat, zone, langue, budget, durée et activités). Ces informations servent de base au système pour générer des recommandations personnalisées.

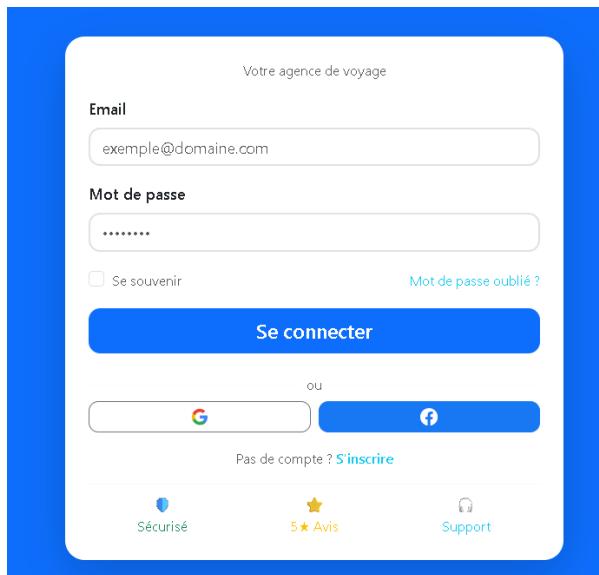


Figure 9 : page de connexion

➤ **Description de la page de connexion**

Cette page permet aux utilisateurs de s'authentifier à l'aide de leurs identifiants (email et mot de passe) afin d'accéder aux différentes fonctionnalités offertes par l'application. Elle constitue un point d'entrée sécurisé vers l'environnement utilisateur.

The screenshot shows a blue-themed sign-up form titled 'Créez votre compte et explorez le monde'. It includes fields for First Name (Soudaiss), Last Name (ELFAYADINE), Email (example@domaine.com), Phone (+221 77 347 56 78), Password, and Confirm Password. There are two checkboxes: 'J'accepte les conditions d'utilisation et la politique de confidentialité' and 'Recevoir les offres spéciales et newsletters'. A large blue button labeled 'Créer mon compte' is at the bottom. Below it, there's a link 'ou s'inscrire avec' followed by Google and Facebook icons. A 'Déjà un compte ? Se connecter' link is also present. At the bottom, there are four icons: 'Paiement sécurisé' (secure payment), 'Offres exclusives' (exclusive offers), 'Programme fidélité' (loyalty program), and 'Support 24/7'.

Figure 10: page d'inscription

➤ **Description de la page d'inscription**

La plateforme d'inscription est conçue pour permettre aux patients de créer un compte et d'accéder aux fonctionnalités de l'application.



Figure 11 : page du dashboard Administrateur

➤ **Description du tableau de bord administrateur**

Cette interface offre une vue d'ensemble sur les statistiques clés du système, telles que le nombre total de réservations, le chiffre d'affaires généré et le nombre de clients . Elle permet à l'administrateur de suivre en temps réel l'activité de la plateforme.

SECTION 5 : OUTIL DEVOPS

L'adoption de pratiques DevOps garantit la qualité et la fiabilité du code tout au long du cycle de développement de la plateforme de réservation. Les outils de versioning, de

conteneurisation et d'analyse de code permettent d'automatiser les processus et de minimiser les risques en production. Cette approche améliore la collaboration et assure un déploiement sécurisé de l'application web.

❖ Git / GitHub

Pour améliorer notre flux de travail, nous avons utilisé Git pour le contrôle de version et GitHub pour la collaboration et la gestion de projets

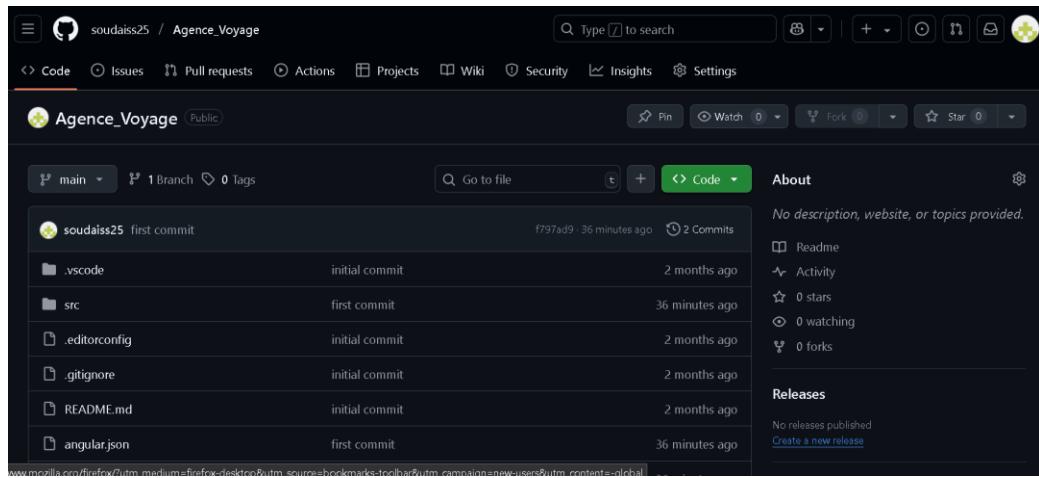


Figure 12: Repository Github

❖ Docker

Une plateforme de conteneurisation qui permet aux développeurs de créer, déployer et exécuter des applications dans des conteneurs légers et portables.

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
nginx	246f09bfffcb	Exited	N/A	8000:80	8 months ago	
sonarqube	38524c8c3441	Exited (130)	N/A	9000:9000	5 months ago	
nexus	0491e41eb6f4	Exited (255)	N/A	8081:8081	1 minute ago	
quirky_black	9eec56a66171	Exited (143)	N/A	4444:4444	5 months ago	
selenium-hub	39596055d26c	Exited (143)	N/A	4444:4444	5 months ago	
sonarqube	cb87c4b9fd14	Exited (130)	N/A	9000:9000	2 minutes ago	

Figure 13: Conteneur Docker

❖ SonarQube

SonarQube est un outil open source d'analyse de code qui automatise l'évaluation de la qualité du code, la détection des bugs, des vulnérabilités et des problèmes de code, tout en fournissant des rapports détaillés pour améliorer la maintenabilité et la sécurité des applications logicielles.

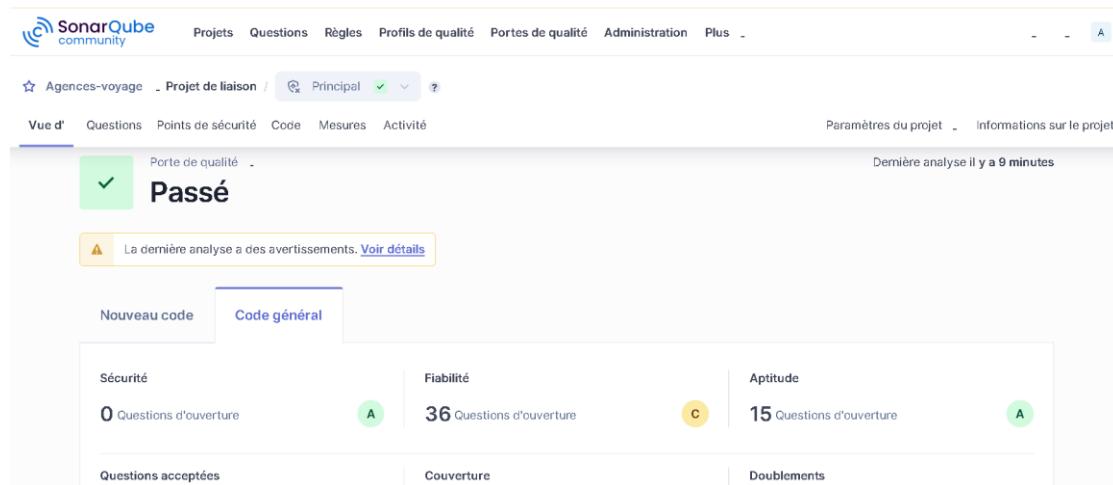


Figure 14: Interface de Sonarqube

CHAPITRE III: IMPLÉMENTATION

SECTION 1 : ENVIRONNEMENT

Le développement a été réalisé en environnement local sur un poste équipé de Windows 10, avec les outils suivants :

- Visual Studio Code pour Angular ;
- PhpStorm pour Laravel ;
- PostgreSQL avec pgAdmin pour la gestion des données ;
- Postman pour le test des API ;
- Git et GitHub pour le versioning.



SECTION 2 : ÉTUDE DE QUELQUES LANGAGES DE PROGRAMMATION

Dans le cadre de ce projet, deux langages principaux ont été utilisés pour le développement de l'application : **TypeScript**, via le framework **Angular**, pour la partie **frontend**, et **PHP**, via **Laravel**, pour la partie **backend**. Cette combinaison permet de bien séparer la gestion de l'interface utilisateur (côté client) de la logique métier (côté serveur).

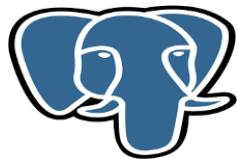
Le **TypeScript** apporte une structure plus rigoureuse au JavaScript traditionnel, ce qui facilite la maintenance et la lisibilité du code, surtout dans des projets de taille moyenne à grande. Grâce à Angular, il est aussi possible d'organiser le code en composants réutilisables, ce qui améliore la modularité de l'application.

Côté serveur, **PHP** est utilisé avec le framework **Laravel**, connu pour sa simplicité et sa rapidité de développement. Laravel offre une architecture claire basée sur le modèle MVC (Modèle-Vue-Contrôleur), ce qui facilite la séparation des différentes couches logiques du projet.

En complément, des langages comme **HTML**, **CSS**, et **JavaScript** sont également utilisés. Le **HTML** permet de structurer les pages, le **CSS** sert à définir leur apparence, tandis que **JavaScript**, parfois utilisé en parallèle avec TypeScript, permet d'ajouter des fonctionnalités dynamiques à l'interface.

SECTION 3 : ÉTUDE COMPARATIVE DE QUELQUES SGBD

Après étude de MySQL, SQLite et PostgreSQL, le choix s'est porté sur PostgreSQL pour ses fonctionnalités avancées, sa stabilité, son intégration facile avec Laravel, et sa compatibilité avec des bases de données relationnelles complexes.



SECTION 4 : ARCHITECTURE APPLICATIVE

L'architecture choisie est basée sur le modèle MVC (Modèle-Vue-Contrôleur), assurant une bonne séparation des responsabilités entre les données, l'interface et la logique métier.

Cette structure améliore la maintenabilité du projet, permet un travail en équipe plus fluide et facilite le test unitaire des composants.

SECTION 5 : CHOIX DE LA SOLUTION DE PAIEMENT EN LIGNE

Pour la gestion des transactions, le service **Stripe** a été retenu en raison de sa fiabilité, de sa compatibilité avec Laravel et Angular, et de son intégration simple et sécurisée. Stripe prend en charge la conformité **PCI DSS**, ce qui permet de traiter les paiements sans stocker de données bancaires sensibles sur la plateforme. Les fonds sont transférés automatiquement sur le compte bancaire de l'agence sous 2 à 7 jours ouvrés, après déduction d'une commission moyenne de **3,4 % + frais fixes** par transaction.

En complément, une **solution de paiement locale** telle que **Wave** a été intégrée pour permettre aux clients de payer directement via leur mobile, une méthode très répandue au Sénégal. Cette double approche combine **modernité, sécurité et accessibilité**, tout en offrant un avantage compétitif face aux agences locales dépourvues de paiement en ligne.

CHAPITRE IV : DÉPLOIEMENT

SECTION 1 : GESTION DE LA SÉCURITÉ

La sécurité de l'application est une priorité. Les mesures prises incluent :

- Authentification par JSON Web Token (JWT) ;
- Validation des entrées côté client et serveur ;
- Hashage des mots de passe avec Bcrypt ;
- Permissions restreintes selon les rôles (admin, client, visiteur).

SECTION 2 : DIAGRAMME DE DÉPLOIEMENT

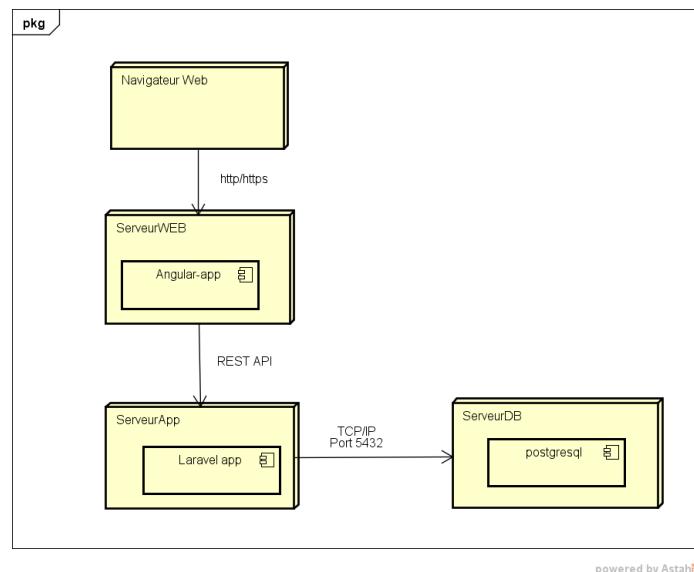


Figure 15 : Diagramme de déploiement

Le diagramme de déploiement présente l'architecture physique du système. On y trouve :

- Le client (navigateur web) ;
- Le serveur d'application (Laravel API REST) ;
- Le serveur frontend (Angular compilé) ;
- Le serveur de base de données PostgreSQL.

SECTION 3 :ESTIMATION DES COÛTS

Le projet a été conçu avec un budget limité, en privilégiant des outils **open source** (Laravel, Angular, PostgreSQL en local) et des environnements gratuits tels que :**Visual Studio Code** et **PHPStorm**. Le développement étant réalisé dans un cadre académique, **aucun coût de main-d'œuvre** n'est comptabilisé. L'objectif est simple : faire fonctionner une solution propre et stable avec le minimum de dépenses.

1. Infrastructure & hébergement

- Services envisagés : Vercel (frontend) et Railway (backend et base de données).
- Hébergement frontend (Vercel) : sera assuré gratuitement.
- Hébergement backend (Railway) : sera pris en charge gratuitement grâce au crédit mensuel inclus (5 \$/mois).
- Nom de domaine : optionnel ; dans un premier temps, les sous-domaines fournis par Vercel (ex. monprojet.vercel.app) et Railway (ex. monapi.up.railway.app) seront utilisés.
- Certificat SSL : fourni automatiquement et sans frais sur les deux plateformes.
- Base de données : PostgreSQL sera déployée directement sur Railway, sans coût supplémentaire dans la limite du quota gratuit.

Ce choix est anticipé afin de permettre un déploiement futur simple et efficace, tout en séparant clairement le frontend et le backend. L'utilisation de PostgreSQL (comme en environnement local) garantira une compatibilité totale et limitera la complexité technique ; pour un coût total **0 FCFA**

2. Solutions de paiement

- Stripe : pas de frais d'installation ; commission **3,4 % + 265 FCFA** par transaction (prélevée uniquement lors des ventes).
- Wave Money : commission **1 %** par transaction ; solution locale adaptée au marché sénégalais.

3. Coûts opérationnels

- Connexion Internet : **15 000 FCFA/mois**, soit **180 000 FCFA/an**.
- Maintenance : assurée par le développeur (suivi post-déploiement académique), **sans coût additionnel** prévu.

Avec un budget annuel limité, ce projet montre qu'il est possible de déployer une solution fonctionnelle et sécurisée en s'appuyant sur des outils gratuits et un hébergement cloud. Le choix de vercel et de railway avec PostgreSQL en production est particulièrement adapté pour un projet de fin d'études : aucun coût d'hébergement et une mise en ligne simple.



Dans ce chapitre, nous revenons sur les objectifs fixés en début de projet, en mettant en lumière ceux atteints avec succès, les compétences acquises au cours de cette expérience, ainsi que les apports du projet pour l'entreprise.

1-Objectifs atteints

Objectif n°1 : Création d'une base de données relationnelle cohérente

La base de données a été conçue selon une structure claire et normalisée, couvrant l'ensemble des entités fonctionnelles de la plateforme (utilisateurs, destinations, voyages, paiements, préférences, etc.). Cette base assure une bonne organisation des données et une facilité d'exploitation.

Objectif n°2 : Développement de l'interface utilisateur

Une interface intuitive et responsive a été mise en place pour permettre aux utilisateurs (clients et administrateurs) de naviguer aisément dans la plateforme. Elle offre une expérience utilisateur fluide et agréable.

Objectif n°3 : Implémentation des modules CRUD

Les opérations de gestion (ajout, modification, suppression, consultation) ont été développées pour plusieurs entités clés telles que les utilisateurs, les destinations, les voyages et les préférences. Ces fonctionnalités assurent la flexibilité et la maintenabilité de l'application.

Objectif n°4 : Personnalisation des recommandations

Grâce aux préférences renseignées par les clients (type de voyage, climat, budget, etc.), un système de recommandations personnalisées a été mis en place, améliorant la pertinence des propositions faites aux utilisateurs.

Objectif n°5 : Intégration d'un système de réservation

Un module de réservation de voyages a été implémenté, permettant aux clients de réserver des séjours, de recevoir des factures et de suivre leurs paiements.

2 - Objectifs non atteints

Tous les objectifs fixés en début de projet ont été atteints. Les modules prévus ont été développés conformément aux besoins fonctionnels identifiés, et la plateforme fonctionne de manière satisfaisante à l'issue du développement.

3 - Intérêts personnels

Ce projet m'a permis de mettre en pratique mes compétences en développement web full-stack, en particulier avec des technologies telles que Laravel, Postgresql et Angular. J'ai également renforcé mes connaissances en modélisation de bases de données, en conception d'IHM respectant les normes UI/UX.

Cette expérience a été enrichissante aussi bien sur le plan technique que professionnel, et m'a permis de mieux comprendre les exigences du développement d'une solution complète en entreprise.

4- Perspectives d'évolution

Bien que le projet ait atteint tous ses objectifs, plusieurs axes d'amélioration et d'évolution peuvent être envisagés pour optimiser la plateforme :

- **Mise en place d'une intelligence artificielle** capable de générer des recommandations encore plus personnalisées en analysant le comportement de navigation et l'historique des voyages des clients.
- **Développement d'une application mobile native** (Android/iOS) pour rendre la solution accessible hors navigateur et améliorer l'expérience utilisateur.
- **Intégration avec d'autres services** : Par exemple, connecter la plateforme à des APIs de météo, de transport ou d'événements pour enrichir le contenu proposé aux utilisateurs.

Ces perspectives permettraient à la plateforme de continuer à évoluer, à s'adapter aux besoins changeants des utilisateurs et à rester compétitive sur le marché.

Conclusion Générale

Au terme de ce travail, nous avons conçu et réalisé une plateforme web complète pour la digitalisation d'une agence de voyage. Cette solution, développée avec Laravel, Angular et PostgreSQL, répond aux besoins identifiés en permettant aux clients de consulter les offres, réserver et payer en ligne, tout en offrant aux administrateurs un outil de gestion efficace. Les objectifs fixés ont été atteints : la plateforme est fonctionnelle, sécurisée et adaptée au contexte local.

Un aspect essentiel de ce projet a été l'utilisation de **diagrammes UML (cas d'utilisation, classes, séquences, déploiement)**. Ces représentations graphiques ont joué un rôle fondamental en facilitant la modélisation et la structuration de la base de données. Elles ont permis d'assurer une conception claire et cohérente, réduisant ainsi les risques d'ambiguïtés lors de l'implémentation.

Ce projet démontre qu'il est possible d'accompagner la transformation numérique des petites entreprises touristiques avec des outils modernes . Les perspectives d'évolution sont l'intégration d'intelligence artificielle pour améliorer les recommandations et le développement d'une application mobile pour étendre la portée de la solution.

Bibliographie

➤ Ouvrages:

UML en action (de l'analyse des besoins à la conception), Pascal Roques et Franck Vallée, 4e édition Eyrolles.

Angular – Développez vos applications web avec le framework JavaScript de Google, Sébastien Ollivier, Éditions ENI, 2^e édition, 2019.

➤ Mémoires:

Mme GUEYE Adji Aby, développement d'un site web de réservation de voyages pour une agence de voyage, ISI, 2023-2024, 43 pages.

M. Ibrahima Amadou Bocoum Conception et réalisation d'une Plateforme de Réservation de Voyage pour une Agence de BOCOUM TRANSPORT, ISI, 2023-2024, 47 pages.

Webographie

<https://laravel.com/docs/11.x/billing> 02/07/2024 09h:15

<https://stripe.com/docs/api> 06/07/2024 11h:30

<https://developer.orange.com/apis> 06/07/2024 14h:45

<https://docs.wave.com/docs> 06/07/2024 16h:20

<https://angular.io/docs> 10/07/2024 10h:35

<https://www.postgresql.org/docs/current/> 18/07/2024 13h:50

<https://docs.render.com/> 14/07/2024 15h:25

<https://getbootstrap.com/docs/5.3/getting-started/introduction/> 16/07/2024 18h:40

<https://laravel.com/docs/11.x/authentication> 22/07/2024 14h:45

<https://stripe.com/docs/payments/accept-a-payment> 06/07/2024 16h:20

<https://github.com/laravel/laravel> 26/07/2024 11h:35

<https://www.w3schools.com/php/> 28/07/2024 13h:50

https://developer.mozilla.org/en-US/docs/Web/API/Payment_Request_API
01/08/2024 10h:25

<https://jwt.io/introduction/> 03/08/2024 12h:40

<https://restfulapi.net/> 07/08/2024 08h:20

<https://docs.github.com/en/actions> 13/08/2024 19h:15

Table des matières

Dédicaces I

Remerciements II

LISTES DES FIGURES III

Sigles et acronymes..... IV

Résumé VII

Abstract VIII

INTRODUCTION GÉNÉRALE IX

PREMIÈRE PARTIE : CADRE THÉORIQUE ET MÉTHODOLOGIQUE 1

CHAPITRE 1 : LE CADRE THÉORIQUE 2

 1. Présentation de la structure d'accueil 2

 2. PRÉSENTATION DU SUJET..... 2

CHAPITRE II : LE CADRE MÉTHODOLOGIQUE..... 4

 SECTION 1 : MÉTHODE D'INVESTIGATION 4

 SECTION 2 : ÉTUDE COMPARATIVE DES MÉTHODES D'ANALYSE ET DE CONCEPTION 5

DEUXIEME PARTIE : TRAVAUX RÉALISÉS AU COURS DU PROJET 7

CHAPITRE I : ANALYSE..... 8

 SECTION 1 : PRÉREQUIS THÉORIQUES..... 8

 SECTION 2 : ANALYSE DU SYSTÈME..... 9

CHAPITRE II : CONCEPTION..... 12

 SECTION 1 : DIAGRAMMES DE CLASSE..... 12

 SECTION 2 : DIAGRAMMES DE SÉQUENCE..... 13

 SECTION 3 : TRAVAUX APPLICATIFS 14

 SECTION 4 : MAQUETISATION DE LA SOLUTION 16

 SECTION 5 : OUTIL DEVOPS 18

CHAPITRE III: IMPLÉMENTATION..... 20

 SECTION 1 : ENVIRONNEMENT 20

 SECTION 2 : ÉTUDE DE QUELQUES LANGAGES DE PROGRAMMATION 21

 SECTION 3 : ÉTUDE COMPARATIVE DE QUELQUES SGBD 21

 SECTION 4 : ARCHITECTURE APPLICATIVE..... 22

SECTION 5 : CHOIX DE LA SOLUTION DE PAIEMENT EN LIGNE	22
CHAPITRE IV : DÉPLOIEMENT	22
SECTION 1 : GESTION DE LA SÉCURITÉ.....	22
SECTION 2 : DIAGRAMME DE DÉPLOIEMENT.....	23
SECTION 3 :ESTIMATION DES COÛTS	23
TROISIEME PARTIE : BILAN.....	25
1-Objectifs atteints.....	26
2 - Objectifs non atteints.....	27
3 - Intérêts personnels.....	27
4- Perspectives d'évolution	27
Conclusion Générale.....	28
Bibliographie i	
Webographie ii	