Dédicace

Avec une émotion profonde et sincère, je dédie ce travail :  
À ma mère et à mon père, sources inépuisables de courage, de confiance et d'amour. Vos sacrifices et votre soutien indéfectible m’ont permis de rêver et de réaliser mes ambitions. Vous êtes les véritables architectes de ma réussite.  
À mes professeurs, guides éclairés sur le chemin de la connaissance. Vos enseignements, votre patience et vos encouragements m’ont accompagné à chaque étape de ce projet. Votre dévouement à mon éducation est une lumière qui m’a guidé.  
À mes amis et camarades, compagnons de route et de vie. Votre présence, vos rires, vos conseils et votre solidarité ont rendu cette aventure inoubliable et enrichissante.  
À toutes les personnes, proches ou lointaines, qui ont contribué à ma progression. Vos gestes de gentillesse, vos mots de soutien et vos actions ont été des ponts vers de nouveaux horizons.  
À tous ceux qui m’ont permis de croire en moi-même et en mes capacités, j’exprime ma gratitude la plus profonde et sincère.

Remerciements

Au terme de mon stage effectué au sein de **SOGELEC Tunisie**, j’adresse mes vifs remerciements à **M. Med Faouzi Ben Mefteh**, Président Directeur Général, ainsi qu’à **M. Habib Hannachi**, Directeur Technique, de m’avoir permis d’accomplir ce projet dans leur honorable unité.

Je tiens à exprimer ma profonde gratitude à mon encadrant professionnel, **M. Anis Jmal**, pour l’intérêt porté à mon travail, pour son accompagnement continu et ses précieux conseils qui ont grandement enrichi mon expérience.

Je remercie également mon encadrante académique, **Mme** , enseignante à l’**ISI Kef**, pour sa disponibilité, son soutien constant et ses orientations méthodologiques qui m’ont guidé tout au long de la réalisation de ce projet.

Mes remerciements s’adressent aussi aux membres du jury pour l’honneur qu’ils me font en acceptant d’évaluer mon travail et pour le temps consacré à l’examen de ce mémoire.

Je rends hommage à mes parents qui n’ont cessé de m’encourager et de m’orienter tout au long de mon parcours.

Enfin, j’exprime ma profonde reconnaissance à l’ensemble de mes enseignants de l’**ISI Kef** pour les connaissances transmises et leur accompagnement durant ma formation

## Introduction générale

Dans le cadre de ma formation à l’Institut Supérieur de l’Informatique du Kef (**ISI Kef**), j’ai été amené à réaliser un **Projet de Fin d’Année (PFA)**. Ce projet représente une opportunité d’approfondir mes connaissances théoriques et pratiques et de les mettre en application dans un contexte professionnel réel.

J’ai ainsi choisi d’effectuer mon projet au sein de la société **SOGELEC Tunisie**, spécialisée dans les solutions électriques et de maintenance. Le projet porte sur le **développement d’une plateforme web de gestion des interventions techniques et du suivi de maintenance**, un outil destiné à optimiser la communication entre clients, techniciens et administrateurs, tout en assurant une meilleure traçabilité des interventions.

Ce projet s’inscrit dans la continuité des besoins réels de l’entreprise, en mettant en œuvre les technologies web modernes (Laravel, MySQL, et Bootstrap) pour concevoir une application robuste, efficace et adaptée aux besoins métiers.

Le présent rapport est structuré en quatre chapitres :

1. Dans le premier chapitre, je présenterai le **cadre général du projet** et les objectifs attendus.
2. Dans le deuxième chapitre, je développerai une **étude théorique** relative aux concepts de la gestion des interventions et des solutions logicielles similaires.
3. Le troisième chapitre sera consacré à l’**analyse et au dimensionnement du système**, en définissant les besoins fonctionnels et techniques.
4. Enfin, dans le quatrième chapitre, je détaillerai la **conception et la réalisation de la plateforme web**, avant de conclure par une conclusion générale mettant en évidence les apports et perspectives de ce travail.

# Chapitre I : L’entreprise D’accueil

## Introduction

Ce chapitre est une présentation générale de l’entreprise d’accueil, il présente les différentes activités et domaines que le bureau d’étude les prend en charge.

## Présentation de l’entreprise

## Historique

SOGELEC est une société d'installation de Matériels électrique en Tunisie, Leader de son secteur depuis sa création en 1972 par Monsieur Noureddine BEN MEFTAH avec un Capital de 1.800.000 TND.SOGELEC est active dans le secteur d'installation Matérielles Électrique en Tunisie et au Maghreb.



## Le statut juridique

Date de constitution : 1976

Siège Social : Rue El Cham, Tunis 1002.

Forme juridique : Société Anonyme

Direction : Mr. Faouzi Ben Meftah (Directeur général) Nombre d’agence : 1

Capital : 1.8 million de dinars tunisiens

Téléphone : 71 831 381

Site web : [http://sogelec.net](http://sogelec.net/)

Matricule fiscal : DE-480354

## Organigramme Sogelec

Une image contenant texte, capture d’écran, cercle

Description générée automatiquement

Figure 1 - Organigramme Sogelec

## Mission de SOGELEC

Être à la pointe dans le secteur d'installation Électrique en Tunisie pour mériter notre position et garantir la satisfaction de nos clients.

Être à jour dans les normes matérielles Électriques et les nouveautés du secteur pour conseiller le client dans son projet d'installation Électrique grâce à notre Bureau d'études et d'engineering en Installations Électriques et pour travailler en synergie avec notre client

## SOGELEC & Environnement

* Un modèle de croissance économique, rentable.
* Un engagement dans la lutte contre le changement climatique.
* Le développement personnel de ses collaborateurs.
* Des actions =concrètes vis-à-vis de la société par :
* Dialogue constant cultivé avec toutes les parties prenantes
* Mise en place d’outils d’information et d’évaluation performants.
* Cinq engagements forts en matière de responsabilité d’entreprise :
* Proposition de solutions électriques Performantes.
* Respectant la Préservation de l'Environnement
* Mise en œuvre d’une croissance économique durable
* Favoriser le développement personnel des salariés
* Actions sociétales auprès des communautés locales

## Cadre générale du projet

## Présentation du projet

L’hôtel est constitué de **(10) niveaux** (Sous-sol+RDC+Mezzanine+6Etages hébergements+1Rooftop) dont les fonctions sont reparties comme suit :

* Sous-sol : principalement dédié à une cuisine, une buanderie, des locaux techniques et le stationnement des voitures,
* Rez de chaussée : Restaurant, cuisine, la réception, des salles de réunions, poste transfo, infirmerie…
* Mezzanine : Bureaux, salle de surveillance, des locaux techniques, locaux du personnel
* Etage 1 : Hébergements : 30 chambres
* Etage 2 : Hébergements : 30 chambres
* Etage 3 : Hébergements : 28 chambres
* Etage 4 : Hébergements : 28 chambres
* Etage 5 : Hébergements : 28 chambres
* Etage 6 : Hébergements : 26 chambres
* Etage 7 (Rooftop) : restaurant, terrasse, salle de sport, piscine, cuisine, locaux techniques

## Les étapes de réalisation du projet

Les étapes de réalisation d’un projet sont présentées par l’organigramme ci-dessous.

Maitre d’ouvrage

Consultation

Architecte

Plan

Bureau d’étude

Dossier d’appel d’offre

Approbation

Avant- projets détaillés

Approbation

Avant- projet sommaire

Bureau de contrôle

Approbation

Entreprise

Approbation

Figure 2 – schéma de réalisation du projet

* **Le maitre d’œuvre :** C’est le responsable de réalisation de l’étude architecturale ainsi qu’il est compétent à l’intervenir à tous les niveaux d’un projet de conception à la réalisation. Après avoir planifier le projet avec l'architecte, la première étape est de consulter le projet au bureau d’étude.
* **Le bureau d’étude :** Le bureau d'étude assure la réalisation des études de l’électricité, courant fort, courant faible, sécurité incendie, vidéosurveillance, alarme intrusion, contrôle d’accès, dossier protection civile, ainsi qu’électrification et éclairage public, Les interventions commencent par la conception et l’identification du projet jusqu'au lancement des appels d’offres pour la réalisation des travaux.

Les différentes étapes de réalisation des études sont :

* Les études de faisabilité.
* Les études d’avant-projet sommaire.
* Les études d’avant-projet détaillé.
* L’élaboration du cahier des charges technique (CCTP) en rapport avec les besoins du client et les normes en vigueurs.
* Dossier de consultation des entreprises.
* L’étude comparative des offres techniques et financières des entreprises.
* **Bureau de contrôle :** Le bureau de contrôle a pour mission d'assister les maitres d'ouvrage, il vérifie le respect des normes de construction et des normes électrique en phase d’étude et encours d’exécution du projet.
* **L’entreprise :** L’entreprise s’engage à mettre en œuvre ses ressources humaines, matérielles et financières afin d’offrir des services d’entrepreneur électricité de qualité.

## Eclairage

Tout l'éclairage du projet sera de type LED :

* Les chambres seront éclairées par des spots et ruban LED,
* La salle de sport sera éclairée par des spots et ruban LED,
* La piscine sera éclairée par des spots LED, des rubans LED et des appliques murales,
* Les sièges seront éclairés par lustres, des suspensions, des spots et des rubans LED, Le hall d'entrée principale sera éclairé par des lustres spots LED et Les cuisines seront éclairées par des panels LED étanches.

Une image contenant intérieur, décoration d’intérieur, mur, propriété

Description générée automatiquement

Figure 3 - Eclairage salon et salle de réunion

Une image contenant bâtiment, propriété, mur, intérieur

Description générée automatiquement

Figure 4 Eclairage restaurant et piscine

Une image contenant intérieur, mur, décoration d’intérieur, meubles

Description générée automatiquement

Figure 5 – Eclairage salle de sport et chambre

## Conclusion

Ce chapitre est un aperçue globale sur les activités et les types d’études du Sogelec. Pour le chapitre suivant on va étudier d’une manière générale les installations électriques, câblage, dimensionnement et choix des composants

# Chapitre II : Analyse et Spécifications des besoins

## Introduction

Dans ce chapitre, nous allons analyser et définir les besoins du projet afin d’assurer une conception claire et précise. L’objectif est de modéliser les fonctionnalités attendues, d’identifier les différents acteurs du système et de spécifier leurs interactions à travers des cas d’utilisation.

Notre plateforme vise à simplifier et automatiser la gestion des interventions techniques et des tickets de maintenance. Elle doit offrir aux clients un moyen simple de soumettre leurs demandes, aux techniciens une interface claire pour planifier et exécuter leurs tâches, et aux administrateurs une vision globale pour superviser l’ensemble des activités.

## Spécification des besoins :

Le système doit offrir les fonctionnalités suivantes :

1. **Gestion des utilisateurs :**
   * Création, modification et suppression des comptes (administrateur).
   * Attribution de rôles (client, technicien, administrateur).
2. **Gestion des tickets de maintenance :**
   * Création d’un ticket par le client (description du problème, priorité, date, etc.).
   * Consultation de l’historique des tickets soumis.
   * Suivi en temps réel de l’état d’avancement d’un ticket (en attente, en cours, résolu).
3. **Gestion des interventions techniques :**
   * Attribution des tickets aux techniciens (par l’administrateur).
   * Planification et mise à jour des interventions (technicien).
   * Ajout de rapports d’intervention (technicien).
4. **Tableaux de bord et supervision :**
   * Vue d’ensemble des tickets et interventions (administrateur).
   * Génération de statistiques (tickets ouverts/fermés, interventions par technicien, délais moyens).
5. **Notifications et communication :**
   * Notification automatique par email lors de la création ou mise à jour d’un ticket.
   * Alertes pour informer les techniciens de nouvelles interventions.
   * Accusé de réception pour le client après la création d’un ticket.

### Identification des acteurs

* + - **Administrateur** : supervise le système, gère les utilisateurs (techniciens/clients), attribue les interventions, et contrôle l’avancement des tickets
    - **Technicien** : reçoit les interventions qui lui sont assignées, les exécute, met à jour leur statut et peut fournir un rapport.
    - **Client** : crée des tickets pour signaler un problème, suit l’état de ses demandes et consulte les interventions réalisées.

### Besoins fonctionnels :

### Besoins non fonctionnels :

Le système doit permettre:

1. **Gestion des utilisateurs** : création, mise à jour et suppression des comptes (admin).
2. **Soumission et suivi des tickets** (client).
3. **Planification et gestion des interventions** (admin/technicien).
4. **Mise à jour du statut des interventions** (technicien).
5. **Notifications automatiques** par email et tableau de bord (tous acteurs).
6. **Tableaux de bord dynamiques** avec statistiques (admin).
   * + **Accessibilité** : plateforme responsive (desktop & mobile).
     + **Performance** : temps de réponse rapide (< 3s).
     + **Sécurité** : authentification sécurisée (Laravel Breeze), gestion des rôles et permissions.
     + **Fiabilité** : disponibilité minimale de 99% en production.
     + **Scalabilité** : possibilité d’ajouter de nouveaux modules (chat en temps réel, paiement, etc.).

## Diagramme de cas d’utilisation :

Les diagrammes des cas d’utilisation constituent un élément fondamental dans la modélisation UML, permettant de représenter de manière claire et synthétique les différentes fonctionnalités offertes par le système du point de vue des utilisateurs ou des acteurs externes.

Ces diagrammes illustrent les interactions entre les acteurs et le système en mettant en évidence les objectifs ou services que chaque acteur peut réaliser. Ils servent à décrire les besoins fonctionnels du système en identifiant les cas d’utilisation principaux, ce qui facilite la compréhension des attentes des utilisateurs et la définition des limites du système.

En représentant graphiquement les différents cas d’utilisation, ainsi que leurs relations avec les acteurs concernés, ces diagrammes aident à structurer et organiser les exigences fonctionnelles. Ils fournissent également une base solide pour la conception détaillée, la validation et la communication entre les membres de l’équipe projet.

### Diagramme de cas d’utilisation global : A diagram of a diagram AI-generated content may be incorrect.

Figure 6 - Diagramme des cas d'utilisation global

### Description textuelle des cas d’utilisation :

|  |  |  |
| --- | --- | --- |
| **Cas d’utilistaion** | Acteur | Description |
| Gérer les utilisateurs | Administrateur | L’administrateur peut créer, modifier ou supprimer des comptes (clients, techniciens). |
| Attribuer une intervention | Administrateur | |  | | --- | | L’administrateur affecte un ticket ou une intervention à un technicien. |  |  | | --- | |  | |
| Consulter le tableau de bord | |  | | --- | | Administrateur |  |  | | --- | |  | | |  | | --- | | L’administrateur accède à une vue globale (tickets, interventions, techniciens). |  |  | | --- | |  | |
| Consulter les interventions | Technicien | |  | | --- | | Le technicien accède à la liste des interventions qui lui sont assignées. |  |  | | --- | |  | |
| |  | | --- | | **Mettre à jour une intervention** |  |  | | --- | |  | | Technicien | |  | | --- | | Le technicien change le statut d’une intervention (nouvelle, en cours, terminée). |  |  | | --- | |  | |
| |  | | --- | | **Ajouter un rapport** |  |  | | --- | |  | | Technicien | |  | | --- | | Le technicien rédige un compte rendu après avoir terminé une intervention. |  |  | | --- | |  | |
| |  | | --- | | **Créer un ticket** |  |  | | --- | |  | | Client | |  | | --- | | Le client soumet une demande d’intervention via le formulaire dédié. |  |  | | --- | |  | |
| |  | | --- | | **Suivre l’état d’un ticket** |  |  | | --- | |  | | Client | |  | | --- | | Le client peut consulter en temps réel l’avancement de ses demandes. |  |  | | --- | |  | |
| |  | | --- | | **Recevoir des notifications** |  |  | | --- | |  | | |  | | --- | | Tous acteurs |  |  | | --- | |  | | Chaque acteur reçoit une notification (par email ou tableau de bord) selon ses actions. |

Tableau 1 – Diagramme des cas d’utilisation

## Diagramme de cas d’utilisation raffiné :

### Diagramme de cas d’utilisation général de l’utilisateur : A diagram of a diagram AI-generated content may be incorrect.

Figure 7 - Diagramme de cas d'utilisation général de l'utilisateur

### Diagramme de cas d’utilisation « s’authentifier » :

Figure 8 - Diagramme de cas d'utilisation « s'authentifer »

Description de diagramme de cas d’utilisation d’authentification :

|  |  |
| --- | --- |
| Titre | S’authentifier |
| Acteur | Administrateur / Utilisateur |
| Résumé (Parties Prenantes et Intérêts) | L’administrateur ou l’utilisateur accède à l’interface d’authentification, saisit son email et mot de passe. Le système vérifie les informations saisies. Si les données sont correctes, l’administrateur ou l’utilisateur accède à son espace spécifique (administration ou utilisateur). Sinon, un message d’erreur est affiché. |
| Pré-condition(s) | * Le système est opérationnel. * Les paramètres (email et mot de passe) sont saisis par l’administrateur ou l’utilisateur. |
| Post-condition(s) | * Authentification valide. * Message d’erreur en cas de données invalides. |
| Scénario nominal | 1. L’administrateur ou l’utilisateur accède à l’interface d’authentification. 2. L’administrateur ou l’utilisateur saisit son email etmot de passe. 3. Le système vérifie les données saisies. |
|  | 4. Si les données sont correctes, l’administrateur accède à l’espace administrateur et l’utilisateur accède à son espace utilisateur. |
| Scénario alternatif | 1. L’administrateur ou l’utilisateur accède à l’interface d’authentification. 2. L’administrateur ou l’utilisateur saisit son email etmot de passe. 3. Le système vérifie les données saisies. 4. Si les informations sont incorrectes, un messaged’erreur est affiché. |

Tableau 2 - Description de diagramme de cas d’utilisation d’authentification

### Diagramme de cas d’utilisation « s’inscrire » :

Figure 9 - Diagramme de cas d’utilisation « s’inscrire »

Description de diagramme de cas d’utilisation de « s’inscrire » :

| **Titre** | **S’inscrire** |
| --- | --- |
| **Acteur** | Utilisateur |
| **Résumé (Parties Prenantes et Intérêts)** | Un utilisateur accède à l’interface d’inscription pour créer un nouveau compte. Il **remplit le formulaire** avec les informations suivantes : nom complet, email, mot de passe, confirmation du mot de passe. Le système **vérifie les données** saisies, crée un nouveau compte si elles sont valides, puis affiche un message de confirmation. En cas d’erreur (email déjà utilisé, données manquantes, mot de passe incorrect, etc.), un message approprié est affiché. |
| **Pré-condition(s)** | Le système est opérationnel. L’utilisateur n’est pas encore inscrit. |
| **Post-condition(s)** | Nouveau compte utilisateur créé si les données sont valides. Message d’erreur affiché si les données sont invalides ou l’email existe déjà. |
| **Scénario nominal** | 1. L’utilisateur accède à l’interface d’inscription. L’utilisateur **remplit le formulaire** avec les informations requises. 2. 3. Le système **vérifie les données saisies**. 3. Si les informations sont valides, le compte est créé. 4. 5. Le système **affiche un message de confirmation** ou redirige l’utilisateur vers son espace personnel. |
| **Scénario alternatif** | 1. L’utilisateur saisit un email déjà utilisé.  2. Le système détecte l’erreur et affiche un message indiquant que l’email est déjà associé à un compte.  3. L’utilisateur saisit des données invalides (champ manquant, mot de passe trop court, etc.).  4. Le système **affiche un message d’erreur correspondant**. |

Tableau 3 – Description de diagramme de cas s’utilisation « s’inscrire »

### Diagramme de cas d’utilisation « Consulter Profil" :

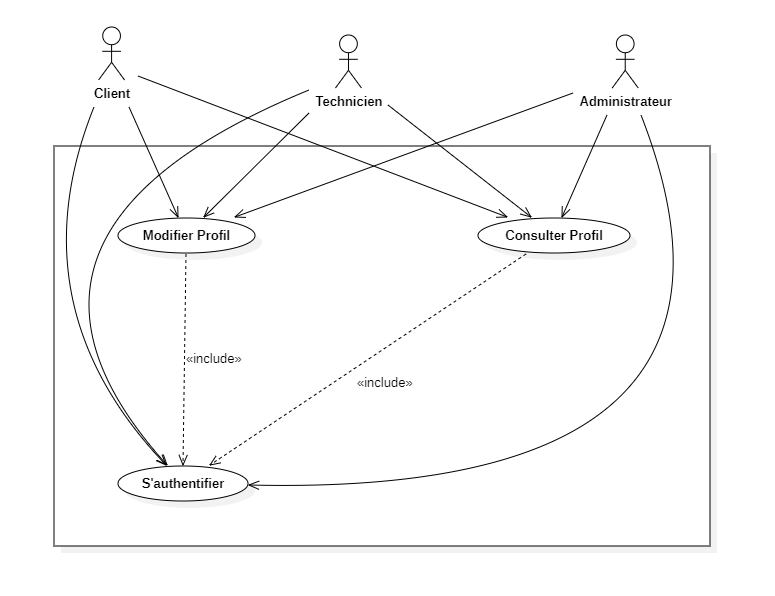


Figure 10 – Diagramme de cas d’utilisation « Consulter le profil »

Description de diagramme de cas d’utilisation de « Consulter le profil» :

| **Titre** | **Consulter le profil** |
| --- | --- |
| Acteur | Client, Technicien, Administrateur |
| Résumé (Parties Prenantes et Intérêts) | L’utilisateur accède à son profil après authentification pour visualiser ses informations personnelles et son rôle dans la plateforme. Il peut également modifier certaines données si nécessaire. |
| Pré-condition(s) | L’utilisateur est authentifié. |
| Post-condition(s) | Les informations du profil sont affichées correctement. Les modifications éventuelles sont sauvegardées. |
| Scénario nominal | 1. L’utilisateur se connecte.2. Il accède à la page “Profil”.3. Le système affiche les informations personnelles.4. L’utilisateur peut modifier certaines données.5. Les modifications sont enregistrées. |
| Scénario alternatif | 1. L’utilisateur tente d’accéder sans authentification.2. Le système demande de se connecter. |

Tableau 11 - Description de diagramme de cas s’utilisation « Consulter Profile »

### Diagramme de cas d’utilisation « Créer Ticket » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 11 - Diagramme de cas d’utilisation « Créer Ticket » :

Description de diagramme de cas d’utilisation de « Créer Ticket » :

| **Titre** | **Créer Ticket** |
| --- | --- |
| Acteur | Client |
| Résumé (Parties Prenantes et Intérêts) | Le client crée une demande d’intervention (ticket) sur la plateforme pour signaler un problème ou demander un service. Le système vérifie les données et enregistre le ticket pour traitement par un technicien. |
| Pré-condition(s) | Le client est authentifié. |
| Post-condition(s) | Un nouveau ticket est créé dans la base de données et assigné éventuellement à un technicien. |
| Scénario nominal | 1. Le client se connecte.2. Il accède à la page “Créer Ticket”.3. Il saisit les informations du ticket (titre, description, urgence, etc.).4. Le système enregistre le ticket.5. Le ticket est disponible pour consultation et suivi. |
| Scénario alternatif | 1. Le client tente de créer un ticket sans authentification.2. Le système demande de se connecter. |

Tableau 4 - Description de diagramme de cas s’utilisation « Créer Ticket »

### Diagramme de cas d’utilisation « Suivre Ticket » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 12 - Diagramme de cas d’utilisation « Suivre Ticket » :

Description de diagramme de cas d’utilisation de « Créer Ticket » :

| **Titre** | **Suivre Ticket** |
| --- | --- |
| Acteur | Client, Technicien, Administrateur |
| Résumé (Parties Prenantes et Intérêts) | Permet à un client de suivre l’état de ses tickets en temps réel. Le technicien et l’administrateur peuvent consulter le ticket et mettre à jour son statut. |
| Pré-condition(s) | L’utilisateur est authentifié. |
| Post-condition(s) | Le ticket est consulté avec son statut actuel. Les mises à jour sont visibles par tous les acteurs concernés. |
| Scénario nominal | 1. L’utilisateur se connecte.2. Il accède à la liste de ses tickets.3. Il sélectionne un ticket pour consulter les détails et le statut.4. Les informations sont affichées en temps réel. |
| Scénario alternatif | 1. L’utilisateur tente d’accéder aux tickets sans authentification.2. Le système redirige vers la page de connexion. |

Tableau 5 - Description de diagramme de cas d’utilisation de « Créer Ticket »

### Diagramme de cas d’utilisation « Consulter Interventions » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 13 - Diagramme de cas d'utilisation « Consulter Intervetion »

Description de diagramme de cas d’utilisation de « Créer Ticket » :

| **Titre** | **Consulter Interventions** |
| --- | --- |
| Acteur | Client, Technicien, Administrateur |
| Résumé (Parties Prenantes et Intérêts) | Permet au client de suivre les interventions associées à ses tickets. Les techniciens peuvent consulter leurs interventions assignées et les administrateurs ont une vue complète pour le suivi et la planification. |
| Pré-condition(s) | L’utilisateur est authentifié. |
| Post-condition(s) | La liste des interventions est affichée avec leur statut et détails. |
| Scénario nominal | 1. L’utilisateur se connecte.2. Il accède à la liste des interventions.3. Il sélectionne une intervention pour voir les détails.4. Les informations sont affichées (date, technicien, ticket lié, statut). |
| Scénario alternatif | 1. L’utilisateur n’est pas connecté → redirection vers la page de connexion.2. Si aucune intervention n’existe, un message indique « Aucune intervention disponible ». |

Tableau 6 - Description de diagramme de cas d’utilisation de « Consulter Interventions » :

### Diagramme de cas d’utilisation « Planning Intervention » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 14 - Diagramme de cas d’utilisation « Planning Intervention »

Description de diagramme de cas d’utilisation « Planning Interventions :

| **Titre** | **Planning Interventions** |
| --- | --- |
| Acteur | Technicien, Administrateur |
| Résumé (Parties Prenantes et Intérêts) | Permet au technicien de connaître son planning d’interventions. L’administrateur peut organiser, modifier et notifier les interventions. |
| Pré-condition(s) | L’utilisateur est authentifié. |
| Post-condition(s) | Le planning est affiché et mis à jour en temps réel. Notifications envoyées si nécessaire. |
| Scénario nominal | 1. L’utilisateur se connecte.2. Il accède au planning.3. Il consulte/modifie les interventions planifiées.4. Les modifications sont enregistrées et les techniciens sont notifiés. |
| Scénario alternatif | 1. L’utilisateur non authentifié → redirection vers la page de connexion.2. Aucun planning disponible → message « Planning vide ». |

Tableau 7 - Description de diagramme de cas d’utilisation « Planning Interventions»

### Diagramme de cas d’utilisation « Gérer Utilisateurs » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 15 – Diagramme de cas d’utilisation « Gérer Utilisateurs »

Description de diagramme de cas d’utilisation « Gérer Utilisateurs» :

| **Titre** | **Gérer Utilisateurs** |
| --- | --- |
| Acteur | Administrateur |
| Résumé (Parties Prenantes et Intérêts) | L’administrateur peut créer, modifier, supprimer et consulter les utilisateurs afin de gérer les comptes de manière efficace et sécurisée. |
| Pré-condition(s) | L’administrateur est authentifié. |
| Post-condition(s) | Les utilisateurs sont ajoutés, modifiés ou supprimés selon les actions. Les données sont mises à jour dans la base. |
| Scénario nominal | 1. L’administrateur se connecte.2. Il consulte la liste des utilisateurs.3. Il ajoute/modifie/supprime un utilisateur.4. Le système confirme l’opération. |
| Scénario alternatif | 1. Tentative d’ajout d’un email déjà utilisé → message d’erreur.2. Tentative de suppression d’un utilisateur inexistant → message d’erreur. |

Tableau 8 - Description de diagramme de cas d'utilisation « Gérer Utilisateurs »

### Diagramme de cas d’utilisation « Gérer Tickets/Interventions » :

A diagram of a diagram

AI-generated content may be incorrect.

Figure 16 - Diagramme de cas d'utilisation « Gérer Tickets/Interventions »

Description de diagramme de cas d'utilisation « Gérer Tickets/Interventions »

| **Titre** | **Gérer Tickets/Interventions** |
| --- | --- |
| Acteurs | Administrateur, Technicien, Client |
| Résumé (Parties Prenantes et Intérêts) | Permet la gestion complète du cycle de vie d’un ticket et des interventions associées, pour assurer un suivi efficace et transparent. |
| Pré-condition(s) | L’acteur est authentifié. |
| Post-condition(s) | Tickets et interventions sont créés, modifiés, assignés ou clôturés correctement. |
| Scénario nominal | 1. Le client crée un ticket.2. L’administrateur l’affecte à un technicien.3. Le technicien planifie et exécute l’intervention.4. Le ticket est clôturé.5. Tous les acteurs peuvent consulter l’état du ticket. |
| Scénario alternatif | 1. Le ticket contient des informations invalides → message d’erreur.2. Tentative de clore un ticket non résolu → notification d’erreur. |

Tableau 9 - Description de diagramme de cas d'utilisation « Gérer Tickets/Interventions »

### Diagramme de cas d’utilisation « Consulter Tableau de bord » :

A diagram of a network

AI-generated content may be incorrect.

Figure 17 - Diagramme de cas d'utilisation « Consulter Tableau de bord »

Description de diagramme de cas d'utilisation « Consulter Tableau de bord »

| **Titre** | **Consulter Tableau de bord** |
| --- | --- |
| Acteurs | Administrateur, Technicien |
| Résumé (Parties Prenantes et Intérêts) | Permet de visualiser l’état global des tickets et interventions pour une meilleure organisation et suivi. |
| Pré-condition(s) | L’acteur est authentifié. |
| Post-condition(s) | Les statistiques et plannings sont consultés correctement. |
| Scénario nominal | 1. L’acteur se connecte.2. L’acteur accède au tableau de bord.3. Les statistiques et le planning sont affichés. |
| Scénario alternatif | 1. Tentative d’accès sans authentification → accès refusé.2. Erreur d’affichage des statistiques → message d’erreur. |

Tableau 10 - Description de diagramme de cas d'utilisation « Consulter Tableau de bord »  
A diagram of a diagram

AI-generated content may be incorrect.

Figure 18 - Diagramme de cas d’utilisation « Consulter le profil»

## V. Conclusion

Ce chapitre a permis de détailler les fonctionnalités principales de la plateforme de gestion SAV en identifiant les acteurs clés : **Client, Technicien et Administrateur**, ainsi que leurs interactions avec le système.

Les différents diagrammes de cas d’utilisation ont clarifié :

* les opérations disponibles pour chaque acteur (authentification, création et suivi de tickets, gestion des interventions, consultation du tableau de bord…),
* les relations entre les fonctionnalités (par exemple, chaque opération inclut l’authentification),
* et les scénarios nominaux et alternatifs pour chaque cas d’usage.

Ces diagrammes fournissent une **base solide pour le développement**, garantissant que le backend et le frontend soient alignés avec les besoins fonctionnels et les exigences métier du projet. Ce chapitre constitue ainsi une **étape fondamentale pour la mise en œuvre efficace de la plateforme**, facilitant la planification des développements et l’assurance qualité.

# Chapitre III : Conception

## introduction :

Après l’analyse des besoins fonctionnels et la détermination des spécifications, ce chapitre présente la conception du projet à travers une série de diagrammes UML.

Nous détaillerons les diagrammes essentiels à la conception, notamment les diagrammes de classes et de séquences.

Ces diagrammes permettent de représenter clairement les interactions entre les acteurs et le système ainsi que la structure statique des données, offrant ainsi une vision complète du fonctionnement attendu.

## Les diagrammes de séquences :

Les diagrammes de séquences sont un outil essentiel de la modélisation UML, permettant de représenter les interactions dynamiques entre les différents acteurs (utilisateurs, systèmes externes) et le système étudié. Ils illustrent de manière chronologique les échanges de messages, d’informations ou d’appels de méthodes entre les objets ou composants du système lors d’un scénario spécifique.

Grâce à cette représentation détaillée des échanges, les diagrammes de séquences facilitent la compréhension du déroulement des processus métiers, des flux de communication, ainsi que des interactions nécessaires pour réaliser une fonctionnalité donnée. Ils permettent également d’identifier clairement les responsabilités de chaque acteur et les étapes successives dans le traitement d’une opération.

### Diagramme de séquence d’inscription : A diagram of a web AI-generated content may be incorrect.

**Scénario nominal :**

1. L’utilisateur accède à la page d’inscription via l’interface web.
2. Le système affiche un formulaire d’inscription.
3. L’utilisateur saisit ses informations (nom, email, mot de passe).
4. L’interface envoie les données d’inscription au système.
5. Le système vérifie dans la base de données si l’email existe déjà.
6. Si l’email est libre, le système enregistre le nouvel utilisateur.
7. La base de données confirme la création du compte.
8. Le système envoie une confirmation d’inscription.
9. L’interface affiche un message de succès à l’utilisateur (« Compte créé avec succès »).

**Scénario alternatif :**

1. L’utilisateur saisit un email déjà utilisé.
2. Le système détecte que l’email existe dans la base de données.
3. Le système envoie un message d’erreur à l’interface.
4. L’interface affiche à l’utilisateur que l’email est déjà enregistré.

### Diagramme de séquence d’authentification :

**Scénario nominal**

1. L’utilisateur accède à la page de connexion.
2. Il saisit son email et son mot de passe.
3. Le système vérifie les informations dans la base de données.
4. Si les informations sont correctes, l’utilisateur est authentifié.
5. L’utilisateur est redirigé vers son tableau de bord.

**Scénario alternatif**

* Si l’utilisateur saisit un email ou mot de passe incorrect, un message d’erreur est affiché.
* L’utilisateur peut ressaisir ses informations.

### Diagramme de séquence « Créer Ticket » :

**Scénario nominal**

1. Le client accède à la page "Créer Ticket".
2. Il saisit le titre, la description et la catégorie du ticket.
3. Le système enregistre le ticket dans la base de données.
4. Le client reçoit une confirmation que le ticket a été créé.

**Scénario alternatif**

* Si certaines informations sont manquantes ou invalides, le système affiche un message d’erreur.
* Le client peut corriger les champs et soumettre à nouveau.

### Diagramme de séquence « Suivre Ticket » : A diagram of a system AI-generated content may be incorrect.

**Scénario nominal**

1. Le client accède à sa section "Mes Tickets".
2. Le système récupère tous les tickets associés au client depuis la base de données.
3. La liste des tickets, avec leur statut, est affichée au client.

**Scénario alternatif**

* Si aucun ticket n’existe, le système affiche un message indiquant qu’aucun ticket n’est disponible.
* Si une erreur de connexion à la base de données survient, un message d’erreur est affiché.

### Diagramme de séquence « Consulter Interventions » :

**Scénario nominal :**

1. Le Technicien accède à l’interface “Consultation des interventions”.
2. Il clique pour afficher la liste des interventions assignées.
3. Le système interroge la base de données et récupère les interventions.
4. La liste est affichée dans l’interface du Technicien.

**Scénario alternatif :**

* Si aucune intervention n’est assignée, le système affiche “Aucune intervention disponible”.

### Diagramme de séquence « Planning Interventions » :

1. L’Administrateur accède à l’interface “Planning des interventions”.
2. Il consulte toutes les interventions planifiées et peut vérifier la disponibilité des Techniciens.
3. Le Technicien consulte uniquement ses interventions assignées dans le planning.
4. Le système récupère les données depuis la base et affiche le planning correspondant.

**Scénario alternatif :**

* Si aucune intervention n’est planifiée pour le Technicien ou l’Administrateur, le système affiche “Aucune intervention disponible”.

### Diagramme de séquence « Gérer Utilisateurs » :

**Scénario nominal :**

1. L’Administrateur accède à l’interface “Gestion des utilisateurs”.
2. Il consulte la liste complète des utilisateurs.
3. Il choisit d’ajouter, modifier ou supprimer un utilisateur.
4. Le système met à jour la base de données et confirme l’action.

**Scénario alternatif :**

* Si les informations saisies sont invalides ou manquantes, le système affiche un message d’erreur.
* Si l’utilisateur n’existe pas lors de la modification ou suppression, le système affiche “Utilisateur introuvable”.

### Diagramme de séquence « Gérer Tickets/Interventions » :

**Scénario nominal :**

1. L’Administrateur accède à l’interface “Gestion des tickets/interventions”.
2. Il consulte tous les tickets et interventions existants.
3. Il peut créer, modifier ou supprimer un ticket/intervention.
4. Le système met à jour la base de données et confirme l’action.
5. Le Technicien peut consulter ses tickets/interventions assignés.

**Scénario alternatif :**

* Si les informations saisies sont invalides, le système affiche un message d’erreur.
* Si le ticket ou l’intervention n’existe pas lors de la modification ou suppression, le système affiche “Ticket/Intervention introuvable”.

### Diagramme de séquence « Consulter Tableau de bord » : A diagram of a system AI-generated content may be incorrect.

**Scénario nominal :**

1. L’Administrateur se connecte et accède au tableau de bord.
2. Le système récupère et affiche les statistiques clés et informations importantes.
3. Le Technicien se connecte et consulte son tableau de bord personnel avec ses tickets et interventions assignés.

**Scénario alternatif :**

* Si les données ne sont pas disponibles, le système affiche un message d’erreur ou une page vide.
* Si l’utilisateur n’est pas authentifié, le système redirige vers la page de connexion.

## Diagramme de classe :

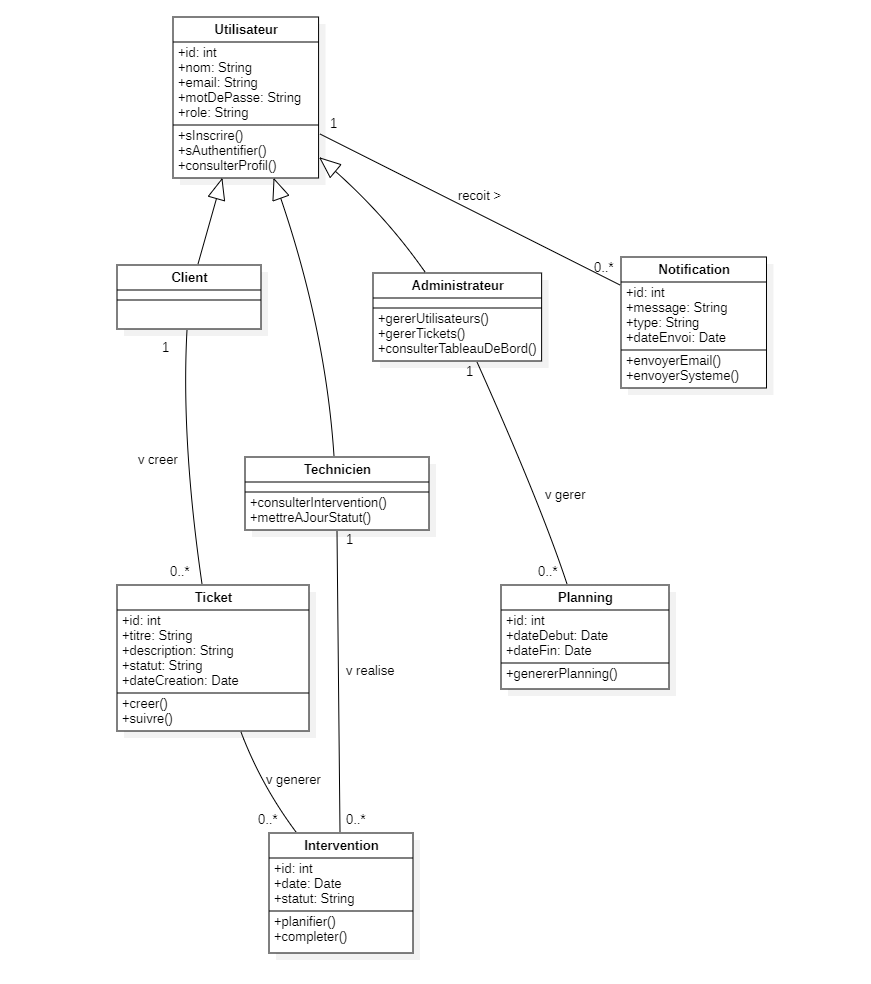


Figure 19 - Diagramme de classe

Ce diagramme de classe représente la structure statique du système. Il met en évidence les entités principales, leurs attributs, leurs méthodes et les relations entre elles.

## Conclusion :

La phase de conception a joué un rôle fondamental dans la structuration et la modélisation de notre application. À travers les différents diagrammes de séquence, nous avons pu visualiser les interactions entre les utilisateurs et le système pour chaque fonctionnalité clé, tandis que le diagramme de classe a permis de représenter la structure statique de l’application de manière claire et cohérente. Cette étape a facilité la transition vers le développement en posant les bases solides nécessaires à une implémentation efficace, tout en assurant une meilleure compréhension du fonctionnement global de la plateforme

# Chapitre IV: Réalisation

## Introduction :

À la suite de la phase d’analyse et de conception, cette étape de réalisation consiste à transformer les spécifications fonctionnelles en une application web opérationnelle. Dans ce chapitre, nous présentons l’architecture technique adoptée, l’environnement de développement, les technologies choisies, ainsi que l’organisation logique des composants de notre plateforme de gestion des interventions et de la maintenance.

## Architecture web de l’application

Nous avons opté pour une architecture **Laravel + React (full stack)**, qui repose sur un backend robuste et une interface frontend moderne. Ce choix garantit la cohérence entre la logique métier et la présentation, tout en assurant une bonne performance et une facilité de maintenance.

### Architecture technique (Larvel + React)

* **Client (Frontend)** : développé avec **React.js**, il constitue l’interface utilisateur. Il interagit avec le backend via des API REST sécurisées.
* **Serveur (Backend)** : réalisé avec **Laravel (PHP)**, il gère la logique métier, l’authentification, la gestion des rôles (Admin, Technicien, Client) et la communication avec la base de données.
* **Base de données** : **MySQL**, une base relationnelle adaptée pour stocker les informations relatives aux utilisateurs, tickets, interventions, plannings et notifications.
* Cette architecture permet une séparation claire des responsabilités (présentation, logique métier, données) et assure la scalabilité de l’application.

## Environnement de développement

| **Composant** | **Technologie / Outil utilisé** |
| --- | --- |
| Backend | Laravel |
| Frontend | React.js, Vite, Tailwind CSS |
| Base de données | MySQL |
| Tests API | Postman, phpMyAdmin |
| IDE | Visual Studio Code |
| Gestion de version | Git + GitHub |
| Dépendances | npm |

## Technologies utilisées

Ce projet repose sur un ensemble de technologies web modernes permettant de développer une application complète, performante, sécurisée et ergonomique. Voici une description détaillée de chacune des technologies et outils utilisés dans le cadre du développement.

React.js [1] est une bibliothèque JavaScript développée par Facebook, utilisée pour la création d’interfaces utilisateur dynamiques. Elle repose sur un système de composants réutilisables et sur un DOM virtuel permettant des mises à jour efficaces de l’interface. Grâce à une approche déclarative, React simplifie la gestion des états de l’application, en facilitant la construction d’applications web en "Single Page Application" (SPA). Cette technologie permet de créer des interfaces modulaires, maintenables et hautement réactives, tout en garantissant de bonnes performances côté client.

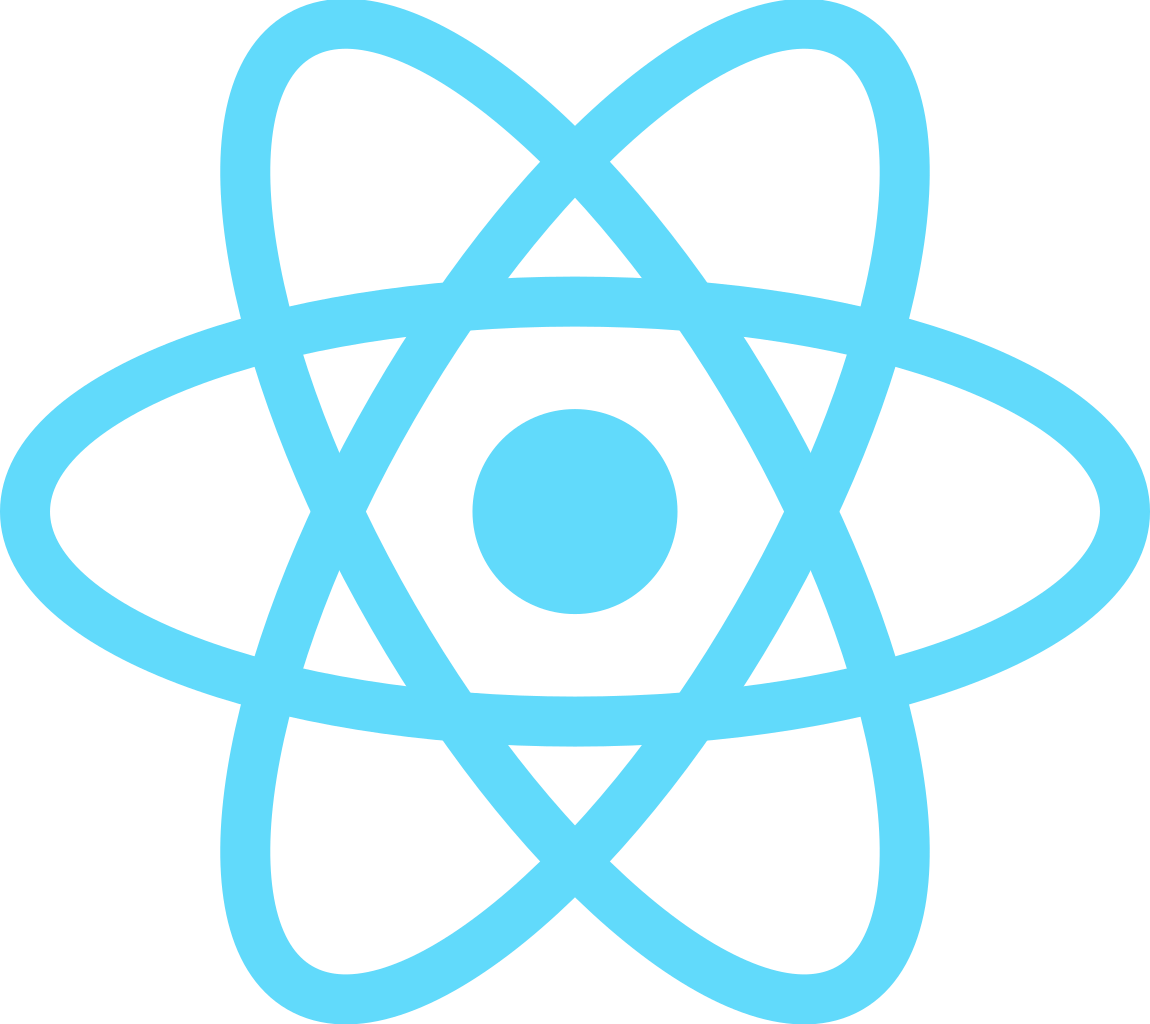


Figure 20 - Logo de React.js

**Laravel [1]** est un framework PHP backend puissant et élégant, conçu pour simplifier le développement des applications web. Il offre une gestion efficace des routes, de la sécurité, des migrations et de l’authentification. Grâce à son architecture MVC et son système de paquets (Composer), Laravel permet de créer rapidement des fonctionnalités complexes comme la gestion des tickets, le suivi des interventions et la planification.

A red line art of a letter u

AI-generated content may be incorrect.

Figure 21 - Logo de Laravel

**MySQL [2]** est un système de gestion de base de données relationnelle open source largement utilisé. Dans ce projet, MySQL est utilisé pour stocker et organiser toutes les données liées aux utilisateurs, tickets, interventions et plannings. Sa structure relationnelle et sa robustesse garantissent une gestion sécurisée et performante des informations critiques du système.

A blue dolphin with yellow text

AI-generated content may be incorrect.

**Bootstrap [3]** est un framework CSS populaire, utilisé pour concevoir rapidement des interfaces web responsives et ergonomiques. Il propose un système de grille flexible et une collection de composants préconçus (boutons, formulaires, cartes, modals, etc.), ce qui permet de gagner du temps tout en assurant une cohérence visuelle dans l’application.



Figure 22 - Logo de bootstrap

JWT (JSON Web Tokens) [8] est une méthode d’authentification basée sur des jetons signés. Elle permet à l’utilisateur de s’authentifier une fois, puis d’accéder aux routes protégées sans avoir à se reconnecter. C’est un mécanisme léger, sécurisé et stateless, largement utilisé dans les architectures REST. Un token est généré au moment de la connexion, signé avec une clé secrète, et envoyé dans l’en-tête des requêtes suivantes pour identifier l’utilisateur.

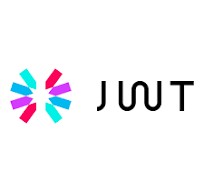


Figure 23 - Logo de JWT

Postman [6] est un outil de test d’API utilisé pour envoyer des requêtes HTTP, analyser les réponses, automatiser les tests et documenter les endpoints. Il a été essentiel pour valider les routes de l’API Express lors du développement du backend. Grâce à son

interface graphique conviviale, Postman permet de simuler des requêtes GET, POST, PUT et DELETE avec des paramètres personnalisés et des en-têtes d’authentification.

A white icon of a person in a helmet

AI-generated content may be incorrect.

Figure 24 - Logo de Postman

Visual Studio Code [9] est un éditeur de code multiplateforme développé par Microsoft. Léger, puissant et extensible, il prend en charge de nombreux langages de programmation et dispose d’une vaste bibliothèque d’extensions. Il propose des fonctionnalités avancées telles que la coloration syntaxique, l’autocomplétion intelligente, la gestion de projet avec Git intégré, ainsi que des outils de débogage. VS Code est l’un des éditeurs les plus populaires parmi les développeurs web.

A blue logo with a black background

AI-generated content may be incorrect.

Figure 25 - Logo de Visual Studio Code

## Architecture logique : le modèle MVC

L’architecture logicielle adoptée repose sur le modèle MVC (Modèle–Vue–Contrôleur), une architecture largement utilisée dans le développement d’applications logicielles, notamment les applications web. Ce modèle vise à séparer les différentes responsabilités du système afin d’améliorer la modularité, la lisibilité et la maintenabilité du code.

— Modèle (Model) : Il représente la couche de gestion des données et de la logique métier. Il est responsable de l’accès aux données, de leur traitement ainsi que de leur validation. Il agit comme une interface entre la base de données et les autres composants de l’application.

— Vue (View) : Elle constitue la couche de présentation. Son rôle est d’afficher les informations à l’utilisateur, en se basant sur les données fournies par le modèle. Elle est également chargée de la gestion de l’interface utilisateur et de l’interaction avec celui-ci.

— Contrôleur (Controller) : Il sert d’intermédiaire entre le modèle et la vue. Il reçoit les entrées utilisateur, les traite, applique la logique métier et met à jour le modèle et/ou la vue en conséquence. Il permet ainsi de coordonner le fonctionnement global de l’application.

Ce découpage clair en trois couches distinctes facilite le développement collaboratif, permet une réutilisation du code et rend l’application plus évolutive en cas de modifications futures.

## Les interfaces de l’application

Dans cette section, nous présentons les principales interfaces de notre plateforme e-commerce à travers des captures d’écran illustrant l’expérience utilisateur et les fonctionnalités clés.