Projet de Fin d’Etudes

Licence Sciences et Techniques Génie Informatique

Département Informatique

Développement d’une application web de mise en relation entre recruteurs et chercheurs d’emploi

Lieu de stage : Ecole Polytechnique des Génies



|  |  |
| --- | --- |
| Réalisé par : | Encadré par : |
| **EL MAHDI MAJDI****ABDERRAHIME ELKOURCHI** | **Pr. Abdelali Boushaba****Mr. Alae Edine Lazrak** |

Soutenu le 20/06/2025 devant le jury composé de :

# **Pr. Rachid Ben Abbou**

# **Pr. Khalid Zenkouar**

# **Pr. Abdelali Boushaba**

Remerciements

Au terme de ce projet de fin d’études, nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Avant tout, nos remerciements vont à Dieu Le Tout-Puissant, pour nous avoir donné la force, la patience et la persévérance nécessaires pour mener à bien ce projet.

Nous remercions chaleureusement l’équipe pédagogique de la Faculté des Sciences et Techniques de Fès, pour la qualité de la formation qu’elle nous a dispensée durant ces trois années et pour l’encadrement dont nous avons bénéficié tout au long de notre parcours universitaire.

Nous exprimons notre reconnaissance à Monsieur Abdelali BOUSHABA, notre encadrant, pour son suivi rigoureux, ses conseils pertinents et sa disponibilité précieuse durant toutes les étapes de ce travail.

Nous adressons également nos remerciements aux membres du jury pour l’attention qu’ils porteront à l’évaluation de notre projet.

Nos remerciements les plus sincères vont aussi à Monsieur Lazrak Alae Eddine, directeur de L'Ecole Polytechnique des Génies, pour nous avoir accueillis au sein de son entreprise et pour l’environnement professionnel stimulant qu’il nous a offert. Grâce à cette opportunité, nous avons pu enrichir nos compétences et mettre en pratique nos acquis académiques dans un cadre concret.

Enfin, nous tenons à remercier nos familles, nos collègues, et toutes les personnes qui nous ont soutenus moralement ou techniquement durant cette aventure. Leur soutien a été un véritable moteur pour surmonter les différents défis rencontrés.

Résumé

Notre projet de fin d’études consiste en la création d’une application web nommée **"Job Souk"**, dédiée à la mise en relation entre les chercheurs d’emploi et les recruteurs. L’objectif principal de cette plateforme est de centraliser, automatiser et simplifier le processus de recrutement au Maroc, en répondant aux besoins spécifiques du marché local.

Pour atteindre cet objectif, nous avons d’abord étudié les pratiques actuelles de recrutement et identifié leurs limites. Nous avons ensuite conçu une solution ergonomique et accessible, intégrant des fonctionnalités essentielles telles que la gestion des profils, la recherche d’offres, le suivi des candidatures, la messagerie interne et l’organisation des entretiens. Le développement s’est appuyé sur des technologies web modernes comme HTML, CSS, JavaScript, PHP et MySQL.

Ce rapport présente l’ensemble des étapes de conception, de développement et d’implémentation de ce projet.

Abstract

Our end-of-study project consists of the development of a web application called **"Job Souk"**, designed to connect job seekers with recruiters. The main objective of this platform is to centralize, automate, and simplify the recruitment process in Morocco, while addressing the specific needs of the local market.

To achieve this goal, we began by analysing the current recruitment practices and identifying their limitations. We then designed an ergonomic and user-friendly solution, incorporating key features such as profile management, job search, application tracking, integrated messaging, and interview scheduling. The development was carried out using modern web technologies such as HTML, CSS, JavaScript, PHP, and MySQL.

This report details all the stages of the design, development, and implementation of this project.

Sommaire

[Chapitre 1 : Contexte général du projet 8](#_Toc199972419)

[**1.** **Présentation de lieu de stage** 9](#_Toc199972420)

[**2.** **Étude de l’existant** 9](#_Toc199972422)

[**2.1.** **Description de l’existant** 9](#_Toc199972423)

[**2.2.** **Critique de l’existant** 10](#_Toc199972424)

[**3.** **Solution proposée** 10](#_Toc199972425)

[**4.** **Cahier des charges** 12](#_Toc199972427)

[**4.1.** **Besoins fonctionnels** 12](#_Toc199972429)

[**4.1.1.** **Gestion des candidats** 12](#_Toc199972431)

[**4.1.2.** **Gestion des recruteurs** 12](#_Toc199972432)

[**4.1.3.** **Gestion des candidats et recruteurs** 13](#_Toc199972433)

[**4.1.4.** **Messagerie** 13](#_Toc199972440)

[**4.1.5.** **Notifications** 13](#_Toc199972441)

[**4.2.** **Besoins techniques** 13](#_Toc199972442)

[Chapitre 2 : Analyse et conception 14](#_Toc199972443)

[**1.** **Analyse et Conception** 15](#_Toc199972444)

[**1.1.** **Langage de modélisation** 15](#_Toc199972445)

[**1.2.** **Analyse des besoins** 15](#_Toc199972446)

[**1.2.1.** **Les acteurs du système** 15](#_Toc199972447)

[**a)** **Candidat** 15](#_Toc199972449)

[**b)** **Recruteur (Entreprise)** 16](#_Toc199972450)

[**c)** **Administrateur** 16](#_Toc199972451)

[**1.2.2.** **Diagrammes des cas d’utilisation** 16](#_Toc199972452)

[**a)** **Diagramme des cas d'utilisation Candidat** 17](#_Toc199972454)

[**b)** **Diagramme des cas d'utilisation Entreprise** 17](#_Toc199972456)

[**c)** **Diagramme des cas d'utilisation Administrateur** 18](#_Toc199972459)

[**1.2.3.** **Présentation des cas d’utilisation et des diagrammes associés (activités ou séquences)** 20](#_Toc199972462)

[**a)** **Diagrammes d’activité** 20](#_Toc199972466)

[**b)** **Diagrammes séquence** 22](#_Toc199972468)

[**2.** **Modèle statique** 24](#_Toc199972470)

[**2.1.** **Diagramme de classes** 24](#_Toc199972471)

[**2.2.** **Schéma de la base de données** 27](#_Toc199972476)

[Chapitre 3 : Interfaces de l’application réalisée 28](#_Toc199972477)

[**1.** **L'environnement de développement** 29](#_Toc199972478)

[**1.1.** **Les langages et technologies web** 29](#_Toc199972480)

[**2.** **Les interfaces graphiques de l’application** 32](#_Toc199972482)

[Références 32](#_Toc199972483)

Liste des figures

[**Figure 1: Diagramme des cas d'utilisation du Candidat.** 15](#_Toc199839206)

[**Figure 2: Diagramme des cas d'utilisation de l’Entreprise.** 16](#_Toc199839207)

[**Figure 3: Diagramme des cas d'utilisation de l’Administrateur.** 17](#_Toc199839208)

[**Figure 4:Diagrammes d’activité : Publier offre d’emploi** 19](#_Toc199839209)

[**Figure 5: Diagrammes de séquence : Planifier un entretien** 20](#_Toc199839210)

[**Figure 6: Diagramme de classes** 22](#_Toc199839211)

[**Figure 7:Logo HTML** 25](#_Toc199839212)

[**Figure 8:Logo CSS** 25](#_Toc199839213)

[**Figure 9: Logo Javascript** 26](#_Toc199839214)

[**Figure 10:Logo Bootstrap** 26](#_Toc199839215)

[**Figure 11:Logo Laravel** 26](#_Toc199839216)

[**Figure 12:Logo XAMPP** 27](#_Toc199839217)

[**Figure 13:Logo phpMyAdmin** 27](#_Toc199839218)

[**Figure 14:Logo Enterprise Architect** 27](#_Toc199839219)

[**Figure 15:Logo Git** 28](#_Toc199839220)

[**Figure 16:Logo GitHub** 28](#_Toc199839221)

Liste des tableaux

[Table 1: Acteurs du système et leurs rôles 8](#_Toc199725256)

[Table 2:Description détaillée du cas d’utilisation Publier offre d’emploi 17](#_Toc199725257)

[Table 3:Description détaillée du cas d’utilisation Planifier un entretien 19](#_Toc199725258)

Liste des acronymes

|  |  |
| --- | --- |
| **Acronyme** | **Désignation** |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| PHP | Hypertext Preprocessor |
| UML | Unified Modeling Language |
| EPG | Ecole Polytechnique des Génies |

Introduction

Dans un monde professionnel en constante évolution, où la rapidité et l’efficacité sont devenues des impératifs, la recherche d’emploi et le recrutement restent souvent des processus longs, désorganisés et peu adaptés aux réalités locales. Malgré l’essor des plateformes de recrutement, de nombreux chercheurs d’emploi et recruteurs marocains rencontrent encore des difficultés à établir un lien direct, fluide et personnalisé.

En tant qu’étudiants en Génie Informatique, nous avons eu l’opportunité de réaliser notre stage de fin d’études au sein de l’École Polytechnique des Génies, un établissement dynamique engagé dans la formation pratique et le développement de solutions numériques concrètes. Ce cadre propice à l’innovation nous a permis de concevoir et de développer ensemble une application web nommée “Job Souk”, destinée à faciliter la mise en relation entre recruteurs et chercheurs d’emploi, tout en automatisant les principales étapes du processus de recrutement.

Ce projet est né du besoin de centraliser les informations, d’optimiser le suivi des candidatures et d’améliorer la communication entre les deux parties, grâce à une messagerie intégrée et une interface intuitive. Il s’inscrit dans une volonté de proposer une solution simple, efficace et spécifiquement adaptée au contexte marocain.

Ce rapport retrace les principales étapes de la conception, du développement et de la mise en œuvre de notre projet, selon la structure suivante :

* Le premier chapitre introduit le lieu de stage, analyse l’existant, identifie la problématique, et présente les solutions proposées à travers un cahier des charges détaillé.
* Le deuxième chapitre est consacré à l’analyse fonctionnelle et à la conception du système. Il présente les acteurs, leurs rôles, ainsi que les différents modèles UML utilisés.
* Le troisième chapitre décrit l’environnement de développement, les technologies employées, ainsi que les interfaces principales de l’application réalisée.

À travers ce projet, nous avons pu mettre en pratique les compétences acquises durant notre formation, tout en répondant à un besoin concret du marché de l’emploi marocain.

# Chapitre 1 : Contexte général du projet

1. **Présentation de lieu de stage**

Basée à Fès, l’École Polytechnique des Génies (EPG) est un établissement d’enseignement supérieur privé, reconnu pour la qualité de ses formations dans les domaines de l’informatique, des nouvelles technologies et de l’ingénierie. Elle offre des cursus variés, allant du niveau qualification jusqu’au master professionnel, accessibles en présentiel ou à distance.

**Domaines de Compétence :**

* **Développement Informatique** : Enseignement des langages modernes (C/C++, Python, Java) et des technologies web et mobiles (Django, Laravel, Android, iOS).
* **Systèmes Embarqués & Objets Connectés** : Formation pratique sur les microcontrôleurs, cartes Arduino, capteurs et systèmes intelligents.
* **Modélisation 3D & Conception Assistée par Ordinateur** : Maîtrise de logiciels comme SolidWorks pour le design industriel et la conception mécanique.
* **Certifications Internationales** : Préparation aux certifications professionnelles telles que Microsoft, Cisco, Oracle et ISO.
* **Langues & Développement Personnel** : Mise à niveau en anglais technique, communication professionnelle et gestion de projet.

En plus de son volet pédagogique, l’EPG dispose d’un service professionnel dédié au développement web et mobile, ainsi qu’au référencement web. Ce département conçoit des sites internet et des applications mobiles sur mesure, tout en assurant l’optimisation de leur visibilité en ligne (SEO). Il offre ainsi aux étudiants l’opportunité de participer à des projets concrets, alignés avec les besoins réels du marché et les pratiques actuelles du secteur numérique.

Avec son approche pédagogique orientée vers la pratique, ses équipements modernes et son encadrement qualifié, l’EPG représente un lieu de stage idéal pour les étudiants souhaitant acquérir des compétences concrètes dans un environnement dynamique et professionnalisant. [1]

1. **Étude de l’existant**
   1. **Description de l’existant**

Actuellement, les processus de recrutement sont majoritairement réalisés via des méthodes traditionnelles ou des plateformes génériques qui ne répondent pas nécessairement aux besoins spécifiques des entreprises et des candidats. Ces méthodes incluent :

* **L’envoi de CV par e-mail** : Les candidats transmettent leurs candidatures à de nombreuses entreprises sans véritable suivi, obligeant les recruteurs à effectuer un tri manuel des dossiers reçus.
* **Les annonces sur des groupes ou forums** : Certains recrutements se font à travers des publications sur des groupes Facebook, forums ou réseaux sociaux, ce qui manque de professionnalisme et de structuration.
* **Utilisation de plateformes généralistes** : Des plateformes comme Indeed, LinkedIn ou Emploi.ma sont souvent utilisées, mais elles sont parfois trop complexes, inadaptées au marché local, ou payantes pour les entreprises souhaitant publier plusieurs offres.

Sur le plan technique, les entreprises locales, notamment les PME, ne disposent pas toujours d’outils numériques efficaces pour gérer tout le processus de recrutement (de la publication de l’offre à l’entretien), et les chercheurs d’emploi n'ont pas accès à un système leur permettant de suivre l’état de leurs candidatures ou de valoriser facilement leurs compétences.

* 1. **Critique de l’existant**

L’analyse de la situation actuelle fait apparaître plusieurs **limitations majeures** :

* **Manque de centralisation** : Les informations concernant les offres d’emploi, les candidats et les processus de recrutement sont dispersées sur différents supports (e-mails, fichiers, réseaux sociaux...).
* **Risque de perte de données** : Les candidatures peuvent se perdre dans les boîtes mail, sans système de sauvegarde structuré.
* **Absence de suivi** : Les candidats ne peuvent pas suivre l’évolution de leurs candidatures (vue, acceptée, refusée, entretien planifié...).
* **Gain de temps non optimal** : Les recruteurs passent beaucoup de temps à trier, lire, filtrer les CV et organiser manuellement les entretiens.
* **Manque d’interactivité** : Il n’existe généralement pas de système de messagerie directe ou de planification automatisée des entretiens entre les candidats et les recruteurs.
* **Aucune personnalisation locale** : Les plateformes généralistes ne prennent pas toujours en compte les réalités et besoins du marché d’emploi local, en termes de langue, secteur, ou types de contrats proposés.

1. **Solution proposée**

Face aux limites identifiées dans l’existant, et sous l’encadrement de l’EPG, nous a proposé de développer une **application web** nommée **“Job Souk”** dont l’objectif principal est de **faciliter la mise en relation entre les recruteurs et les chercheurs d’emploi**, tout en **centralisant et automatisant** le processus de recrutement.

* **Objectifs de la solution :**

L’application “Job Souk” vise à :

* **Centraliser les informations** (offres, candidatures, profils) dans une base de données unique, évitant ainsi la dispersion des données.
* **Simplifier le processus de recrutement** grâce à une interface intuitive et des outils adaptés pour les entreprises comme pour les candidats.
* **Permettre un suivi en temps réel** de l’état des candidatures (envoyée, acceptée, en entretien, rejetée…).
* **Offrir un système de messagerie intégrée** entre recruteurs et candidats, facilitant la communication directe.
* **Automatiser certaines tâches** comme la planification des entretiens ou la validation des comptes par les administrateurs.
* **S’adapter au contexte local**, notamment en termes de langue, de types de contrats, et d’usage simplifié.
* **Fonctionnalités principales :**

Le tableau ci-dessous présente les différents acteurs du système et leurs cas d'utilisation :

|  |  |
| --- | --- |
| Acteur | Les Cas d’utilisation |
| Candidats | * Création et gestion de leur profil personnel (CV , compétences, expériences, etc.). * Consultation et recherche des offres d’emploi. * Postulation en un clic à une offre. * Suivi des candidatures envoyées. * Accès à un système de messagerie avec les recruteurs. |
| Recruteurs | * Création et gestion du profile entreprise. * Publication et gestion des offres d’emploi. * Consultation des candidatures reçues par offre. * Sélection des profils, envoi d’invitations à des entretiens. * Communication avec les candidats via un espace de messagerie. |
| Administrateur | * Gestion des comptes entreprises et candidats (validation, désactivation, suppression). * Suivi des offres publiées (masquer, supprimer). * Traitement des signalements effectués par les utilisateurs * Gestion et mise à jour des annonces sur la plateforme. * Organisation et gestion des catégories d’offres. |

Table 1: Acteurs du système et leurs rôles

* **Avantages de la solution proposée**
* **Adaptation au contexte marocain** : la plateforme prend en compte les spécificités locales (langue, contrats types, attentes des recruteurs marocains), offrant ainsi une meilleure accessibilité aux utilisateurs nationaux.
* **Centralisation des données** : toutes les informations relatives aux offres, aux candidats et aux entreprises sont regroupées dans une base unique, assurant une meilleure cohérence et gestion des données.
* **Digitalisation du processus de recrutement** : la solution automatise plusieurs tâches manuelles (postulation, validation de comptes, planification d’entretiens), réduisant les délais et les erreurs.
* **Amélioration de la visibilité des profils** : les chercheurs d’emploi, notamment les jeunes diplômés, bénéficient d’un espace dédié pour présenter leur profil de manière professionnelle et structurée.
* **Communication facilitée** : le système de messagerie interne permet un échange rapide et fluide entre recruteurs et candidats, évitant le recours à des canaux externes dispersés.
* **Suivi transparent des candidatures** : chaque candidat peut visualiser l’état d’avancement de ses demandes (envoyée, acceptée, en entretien, rejetée...), ce qui améliore l’expérience utilisateur.
* **Soutien à l’emploi national** : en facilitant l’accès aux offres locales et en simplifiant la relation recruteur/candidat, “Job Souk” participe à la dynamisation du marché de l’emploi au Maroc.

1. **Cahier des charges**

Le cahier des charges de l’application **Job Souk** définit les besoins fonctionnels et techniques nécessaires à la conception et à la mise en œuvre du système.

* 1. **Besoins fonctionnels**

L’application doit permettre aux **différents utilisateurs** (candidats, recruteurs, administrateur) d’interagir avec le système selon leurs droits d’accès respectifs.

* + 1. **Gestion des candidats**

Le système doit permettre au candidat de :

* Créer un compte candidat.
* Se connecter et se déconnecter.
* Compléter/modifier son profil (CV, compétences, expériences, formations...).
* Consulter la liste des offres d’emploi disponibles.
* Postuler à une ou plusieurs offres.
* Suivre l’état de ses candidatures.
* Consulter les messages reçus et envoyer des messages aux recruteurs.
  + 1. **Gestion des recruteurs**

Le système doit permettre au recruteur de :

* Créer un compte entreprise.
* Se connecter et se déconnecter.
* Compléter/modifier le profil de l’entreprise.
* Publier de nouvelles offres d’emploi.
* Modifier ou supprimer une offre.
* Consulter les candidatures reçues pour une offre donnée.
* Sélectionner/filtrer des profils intéressants.
* Envoyer des messages aux candidats.
* Planifier des entretiens.
  + 1. **Gestion des candidats et recruteurs (Admin)**

Le système doit permettre à l’administrateur de :

* Se connecter à l’espace d’administration sécurisé.
* Gérer les comptes des recruteurs et des candidats (validation, désactivation et suppression).
* Modérer les offres d’emploi publiées (validation, masquage et suppression).
* Gérer les annonces publiées sur la plateforme.
* Administrer les catégories et sous-catégories d’offres.
* Traiter les signalements effectués par les utilisateurs.
  + 1. **Messagerie**

Le système doit inclure une messagerie interne permettant aux recruteurs et candidats d’échanger des messages en temps réel.

* + 1. **Notifications**

Le système doit notifier les utilisateurs :

* Lorsqu’un message est reçu.
* Lorsqu’une candidature est acceptée ou rejetée.
* Lorsqu’un entretien est planifié.
  1. **Besoins techniques**

Technologies utilisées pour le développement de l'application:

* **Frontend**: HTML, CSS, JavaScript, Bootstrap
* **Backend** : Laravel
* **Base de données**: MySQL
* **Environnement de développement** : XAMPP, Visual Studio Code, git/GitHub
* **Modèle MVC** : utilisé pour l’organisation du code côté serveur

Chapitre 2 : Analyse et conception

1. **Analyse et Conception**
   1. **Langage de modélisation**

Pour modéliser le système **Job Souk**, nous avons utilisé le **langage UML** [2] **.**Ce langage de modélisation est largement utilisé dans le domaine du développement logiciel pour représenter graphiquement :

* Les **acteurs** et leurs interactions avec le système (via les **cas d’utilisation**),
* Les **activités et les processus métier** (via les **diagrammes d’activités**),
* Les **objets et leurs relations** (via les **diagrammes de classes**),
* Les **échanges dynamiques** entre objets (via les **diagrammes de séquence**).

L'utilisation d’UML facilite la **compréhension, la conception et la communication** autour de la solution proposée, notamment en équipe ou avec un encadrant technique.

* 1. **Analyse des besoins**
     1. **Les acteurs du système**

Dans l’application Job Souk, trois types d’acteurs principaux interagissent avec le système. Chacun possède des droits d’accès et fonctionnalités spécifiques selon son rôle.

* + - 1. **Candidat**
* **Définition**

Le candidat est un utilisateur à la recherche d’un emploi. il s’inscrit sur la plateforme pour créer un profil, consulter les offres d’emploi disponibles et y postuler. il peut aussi échanger avec les recruteurs via un système de messagerie.

* **Fonctionnalités accessibles**
* Créer un compte candidat.
* Se connecter / Se déconnecter.
* Compléter ou mettre à jour son profil (CV, photo, expériences, formations, compétences, etc.).
* Rechercher des offres d’emploi (par mot-clé, lieu, type de contrat…).
* Consulter les détails d’une offre.
* Postuler à une offre.
* Suivre l’état de ses candidatures.
* Recevoir des notifications concernant ses candidatures.
* Échanger des messages avec les recruteurs via une messagerie intégrée.
  + - 1. **Recruteur (Entreprise)**
* **Définition**

Le recruteur est un représentant d’entreprise ou une entreprise elle-même, inscrite sur la plateforme afin de publier des offres d’emploi, recevoir des candidatures et entrer en contact avec les candidats.

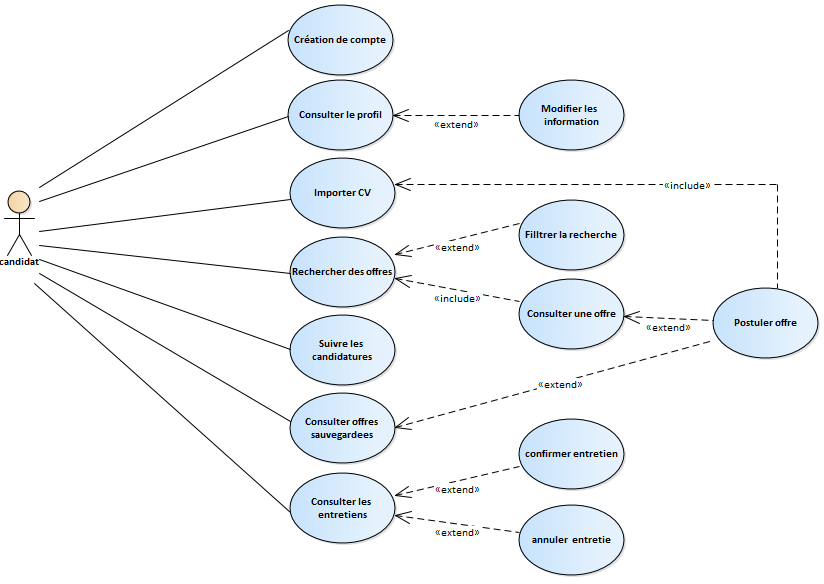
* **Fonctionnalités accessibles**
* Créer un compte entreprise.
* Se connecter / Se déconnecter.
* Compléter/modifier le profil de l’entreprise (logo, description, adresse, secteur d’activité...).
* Publier une nouvelle offre d’emploi.
* Modifier ou supprimer une offre existante.
* Consulter la liste des candidatures reçues pour chaque offre.
* Télécharger les CV des candidats.
* Sélectionner un candidat (accepter ou refuser une candidature).
* Envoyer des messages aux candidats.
* Planifier des entretiens.
  + - 1. **Administrateur**
* **Définition**

L’administrateur est l’utilisateur en charge de la supervision et de la bonne gestion de la plateforme. Il possède un accès complet au système, permettant la gestion des utilisateurs, la modération des contenus, et l’accès aux statistiques.

* **Fonctionnalités accessibles**
* Se connecter à l’espace d’administration.
* Valider ou refuser les inscriptions des recruteurs.
* Supprimer ou désactiver des comptes (candidats ou recruteurs).
* Masquer ou supprimer des offres d’emploi.
* Accéder aux statistiques de la plateforme (nombre de candidats, offres, candidatures, entreprises inscrites…).
* Intervenir manuellement sur le contenu en cas de problème ou de non-respect des règles d’utilisation.
  + 1. **Diagrammes des cas d’utilisation**

Le diagramme des cas d'utilisation est un outil clé en ingénierie logicielle qui illustre les interactions entre les utilisateurs et le système. Il montre les différentes fonctionnalités de l'application et comment les utilisateurs les utilisent. Les éléments principaux sont les acteurs, les cas d'utilisation et leurs interactions. Ce diagramme aide à clarifier les besoins fonctionnels du système pour tous les participants au projet.

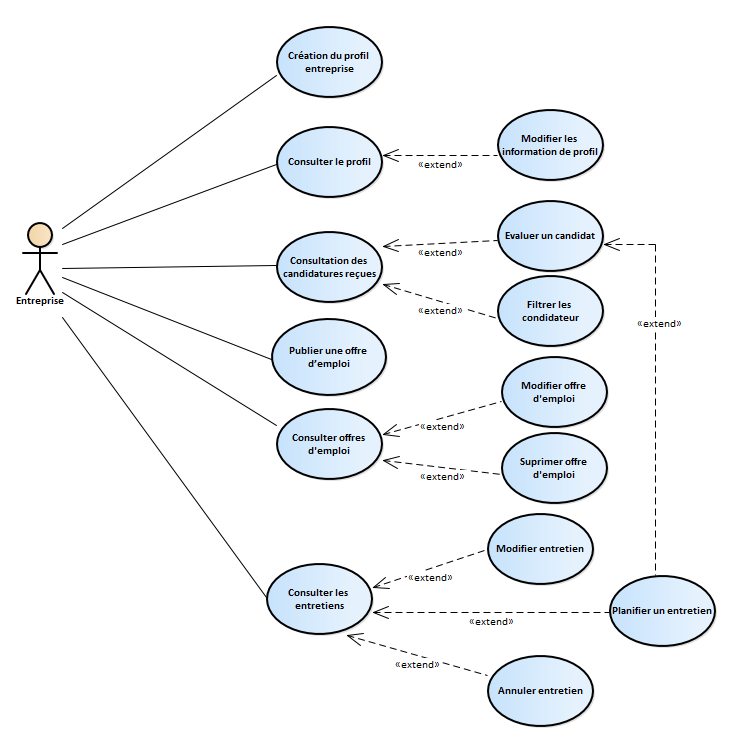
* + - 1. **Diagramme des cas d'utilisation Candidat**

Le diagramme des cas d'utilisation du Candidat montre les différentes interactions qu’un utilisateur en recherche d’emploi peut avoir avec le système Job Souk. Le candidat peut créer un compte, gérer son profil, importer son CV, rechercher des offres, postuler, suivre ses candidatures, consulter ses offres sauvegardées et gérer ses entretiens. Ce diagramme met en évidence les besoins fonctionnels de l'utilisateur côté demandeur d’emploi.

**Figure 1: Diagramme des cas d'utilisation du Candidat**

* + - 1. **Diagramme des cas d'utilisation Entreprise**

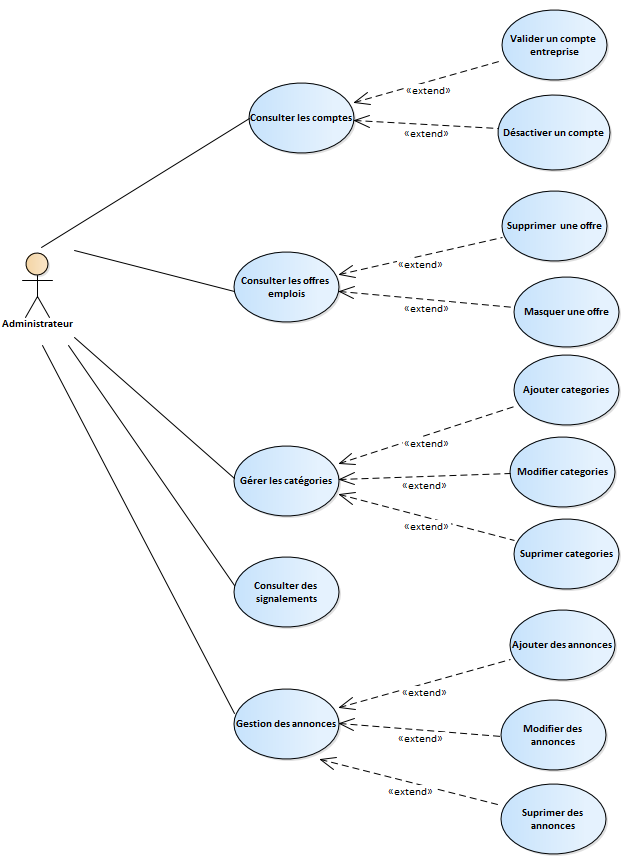
Le diagramme des cas d'utilisation de l’Entreprise illustre les fonctionnalités accessibles à un recruteur. Une entreprise peut créer et modifier son profil, publier des offres d’emploi, consulter les candidatures reçues, évaluer des candidats, planifier ou modifier des entretiens, ainsi que gérer ses offres. Ce diagramme représente les principales opérations permettant à l’entreprise de gérer efficacement le processus de recrutement.

****

**Figure 2: Diagramme des cas d'utilisation de l’Entreprise.**

* + - 1. **Diagramme des cas d'utilisation Administrateur**

Le diagramme des cas d'utilisation de l’Administrateur décrit les actions de gestion et de modération sur la plateforme. L’administrateur peut consulter les comptes, valider ou désactiver ceux des entreprises, gérer les offres d’emploi (masquer ou supprimer), gérer les catégories, les annonces et consulter les signalements. Ces cas d’utilisation assurent le bon fonctionnement et le contrôle global du système.

****

**Figure 3: Diagramme des cas d'utilisation de l’Administrateur.**

* + 1. **Présentation des cas d’utilisation et des diagrammes associés (activités ou séquences)**

Dans cette section, nous présentons en détail les principaux cas d'utilisation de notre application web, en expliquant leur fonctionnement. Chaque cas d'utilisation correspond à une interaction précise entre un utilisateur et le système, en intégrant les étapes à suivre, les conditions préalables ainsi que les résultats attendus. Cette analyse permet de mieux comprendre comment le système satisfait les besoins des utilisateurs et garantit la bonne implémentation des fonctionnalités.

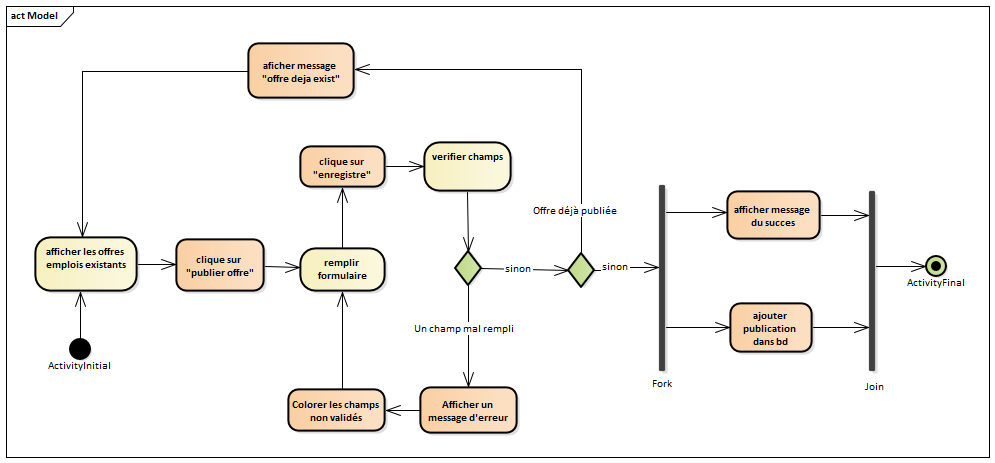
Pour chaque cas d'utilisation, nous proposons une description claire et organisée, incluant les acteurs concernés, le scénario principal, les scénarios alternatifs ainsi que les éventuelles extensions.

Les diagrammes de séquence mettent en évidence l’enchaînement temporel des interactions entre les acteurs et le système, tandis que les diagrammes d’activité retracent les flux de contrôle et les différentes étapes du processus. Ces représentations UML facilitent la compréhension des processus complexes et des interactions détaillées au sein du système.

* + - 1. **Diagrammes d’activité**
* **Publier offre d’emploi**

|  |  |
| --- | --- |
| Champ | Description |
| Nom du Cas d’Utilisation | Publier offre d’emploi |
| Acteurs | Entreprise |
| Description | Ce cas d’utilisation permet à une entreprise de créer et de soumettre une nouvelle offre d’emploi sur la plateforme. Il inclut la saisie des détails de l’offre, la gestion des informations associées et la publication de l’annonce pour qu’elle soit visible par les candidats. |
| Préconditions | L'entreprise doit avoir un compte actif et validé. |
| Postconditions | -Une nouvelle offre d'emploi est créée et enregistrée dans le système.  -L'offre d'emploi est visible par les candidats et les administrateurs.  - L'entreprise peut consulter, modifier ou supprimer l'offre qu'elle a publiée. |
| Scénario Principal | 1. L’entreprise se connecte à son   espace recruteur.   1. L’entreprise clique sur "Créer une   offre d’emploi".   1. Le système affiche un formulaire   avec :   * Champs obligatoires : Titre,   description, localisation, type de contrat, secteur, compétences requises.   * Champs optionnels : Salaire   estimé, télétravail possible, avantages.   * Options de publication : Date   de clôture, visibilité (immédiate ou planifiée).   1. L’entreprise remplit le formulaire et clique sur "Soumettre". 2. Le système vérifie les données   (champs obligatoires complétés).   1. Le système enregistre l’offre avec le   statut "Publiée".   1. Le système :  * Affiche un message de confirmation. * Rend l’offre visible aux candidats. * Notifie l’entreprise par notification. |
| Scénarios Alternatifs | **A1. Données manquantes ou invalides**  -Le système détecte des erreurs (ex : titre vide, date de clôture passée). -Le système :   * Affiche les erreurs en surbrillance. * Bloque la soumission jusqu’à   correction.  **A2. Modification après publication**  -L’entreprise modifie une offre publiée :   * Si les modifications sont mineures   (ex : correction de faute), l’offre reste en ligne.   * Si les modifications sont majeures   (ex : changement de contrat), le système peut exiger une revalidation par un admin (selon règles métier).  **A3. Rejet par l’administrateur**  -Un admin rejette l’offre (ex : contenu inapproprié). -Le système :   * Passe le statut à "Rejetée". * Notifie l’entreprise du rejet. |
| Notes | -Intégrations possibles : |

Table 2:Description détaillée du cas d’utilisation Publier offre d’emploi

****La figure ci-dessous illustre le diagramme de séquence pour le cas d'utilisation " Publier offre d’emploi " :

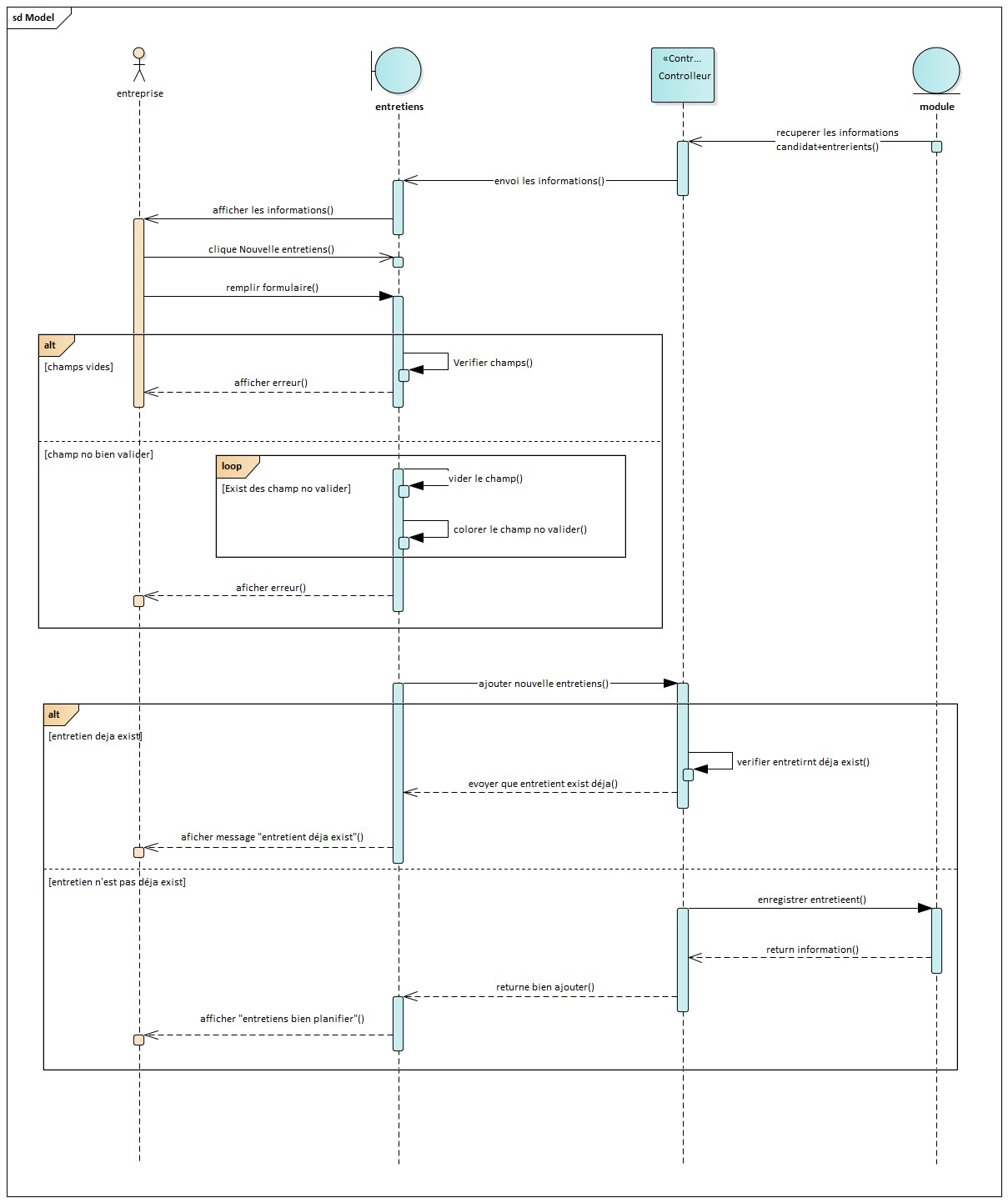
**Figure 4:Diagrammes** **d’activité : Publier offre d’emploi**

* + - 1. **Diagrammes séquence**
* **Planifier un entretien**

|  |  |
| --- | --- |
| Champ | Description |
| Nom du Cas d’Utilisation | Planifier un entretien |
| Acteurs | Entreprise |
| Description | Ce cas d’utilisation permet à une entreprise de proposer des dates et heures d'entretien à un candidat pour une offre d'emploi spécifique, et de gérer le processus de planification jusqu'à sa confirmation. |
| Préconditions | -L'entreprise doit avoir consulté les candidatures (pour une offre donnée) et sélectionné un ou plusieurs candidats à interviewer.  -Le candidat concerné par l'entretien doit exister dans le système et être associé à une candidature.  -L'entreprise doit avoir accès à l'offre d'emploi concernée. |
| Postconditions | - Une proposition d'entretien est envoyée au candidat.  -L'entretien est enregistré dans le système avec le statut.  -Le système peut déclencher des notifications (notification in-app) à l'entreprise et au candidat concernant la proposition d'entretien. |
| Scénario Principal | 1. L’entreprise sélectionne un candidat dans la liste   des candidatures pour une offre.   1. Le système affiche les détails du candidat. 2. L’entreprise clique sur "Planifier un entretien". 3. Le système propose un formulaire pour saisir les   détails de l’entretien (date, heure, durée, format [présentiel/à distance]).   1. L’entreprise valide la proposition. 2. Le système :   -Envoie une notification au candidat avec les détails et un lien pour accepter/refuser/modifier.  -Enregistre l’entretien avec le statut "Proposé".   1. Le candidat accepte la proposition. 2. Le système :   -Met à jour le statut de l’entretien en "Confirmé".  -Notifie l’entreprise et le candidat.  -Ajoute l’entretien aux calendriers respectifs. |
| Scénarios Alternatifs | -A1. Le candidat propose une autre date/heure :  -Le candidat refuse la proposition et suggère une nouvelle plage horaire. -Le système notifie l’entreprise et lui demande de confirmer ou refuser la nouvelle proposition. -Si l’entreprise accepte :   * Le statut passe à **"Confirmé"**. * Les notifications sont envoyées.   -A2. L’entreprise annule l’entretien avant confirmation:  -L’entreprise annule l’entretien depuis l’interface. -Le système :   * Met à jour le statut en **"Annulé"**. * Notifie le candidat avec la raison.   -A3. Le candidat ne répond pas : Si le candidat ne répond pas sous X jours (délai configurable), le système :   * Envoie un rappel automatique. * Passe éventuellement le statut en "Expiré". |
| Notes | -Intégrations possibles :   * Synchronisation avec des outils de calendrier   (Google Calendar, Outlook).   * Envoi de notifications SMS. |

Table 3:Description détaillée du cas d’utilisation Planifier un entretien

La figure ci-dessous illustre le diagramme de séquence pour le cas d'utilisation "Planifier un entretien" :

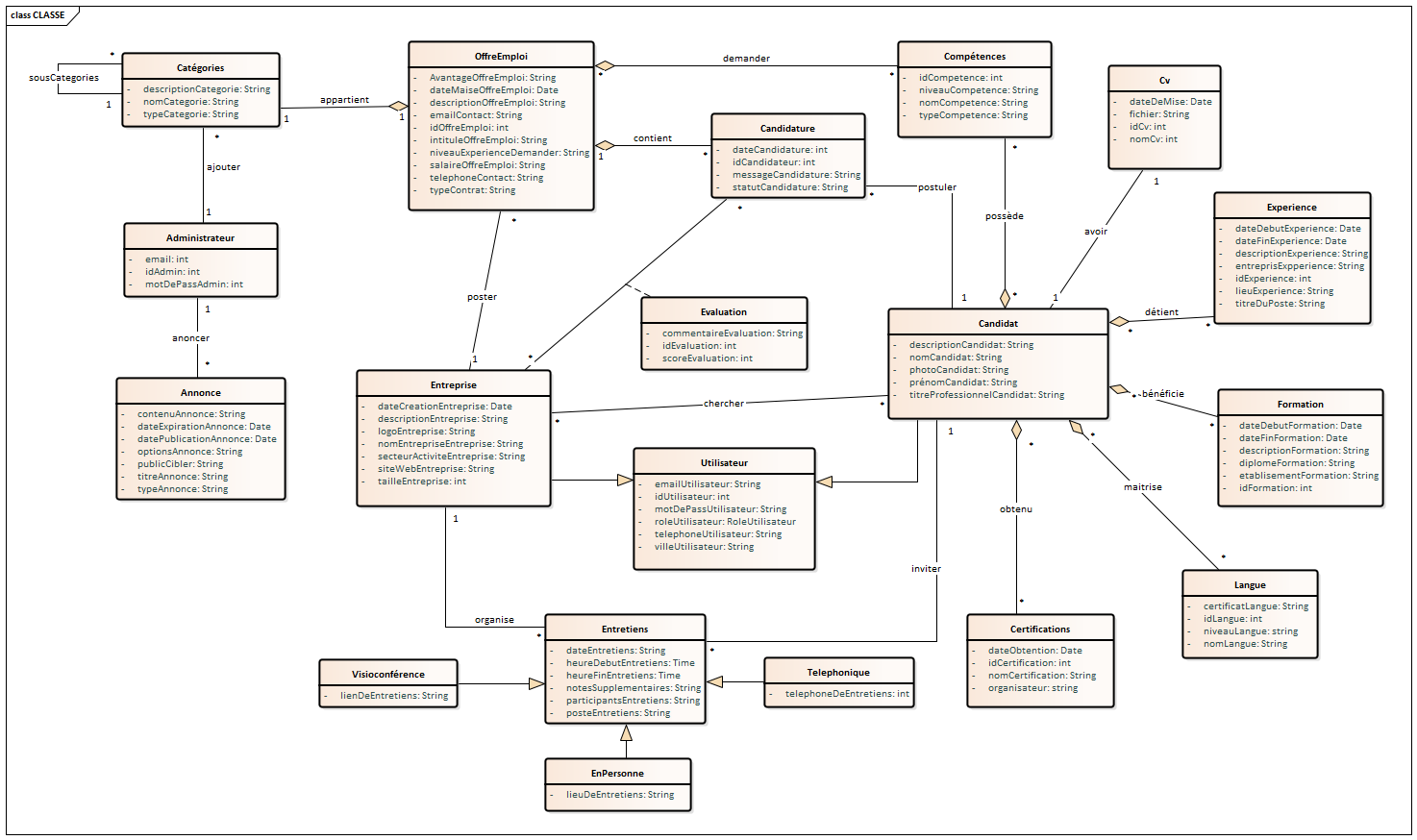
**Figure 5: Diagrammes de séquence : Planifier un entretien**

1. **Modèle statique**
   1. **Diagramme de classes**

Le diagramme de classes représente la structure statique du système Job Souk en modélisant les entités principales, leurs attributs et les relations entre elles.

Le système distingue deux types d’utilisateurs, les candidats et les recruteurs, qui héritent d’une classe générale Utilisateur pour factoriser les informations communes. L’administrateur, quant à lui, est modélisé séparément, car il possède des responsabilités spécifiques de gestion de la plateforme.

Le modèle inclut également les classes OffreEmploi, Candidature, Entretien, Compétence et Message, qui reflètent les fonctionnalités clés du système. Ce diagramme facilite la conception du système en offrant une vue globale claire et cohérente.

****

**Figure 6: Diagramme de classes**

* 1. **Schéma de la base de données**

Chapitre 3 : Interfaces de l’application réalisée

1. **L'environnement de développement**

Ce chapitre détaille les outils et technologies que nous avons utilisés pour la conception, l'implémentation, les tests et la gestion de notre application web. Nous avons sélectionné ces éléments pour optimiser notre cycle de développement logiciel, en tirant parti de leurs caractéristiques spécifiques pour assurer le succès du projet.

* 1. **Les langages et technologies web**



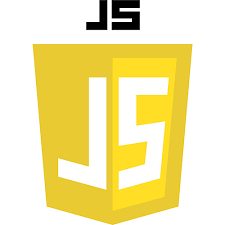
**Figure 7:Logo HTML**

**HTML (HyperText Markup Language)** [3] est le langage de balisage standard utilisé pour créer et structurer des pages web. Dans notre projet, HTML a constitué la base de la conception de l’interface utilisateur. Il nous a permis de définir la structure du contenu à travers des éléments tels que les titres, les paragraphes, les images, les formulaires et les liens. Grâce à HTML, nous avons pu organiser les informations de manière claire et logique, facilitant ainsi la navigation et l’interaction des utilisateurs avec notre application.



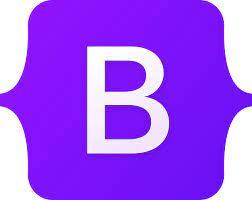
**Figure 8:Logo CSS**

**CSS (Cascading Style Sheets)** [4] est un langage utilisé pour définir l’apparence des documents HTML. Dans notre projet, CSS nous a permis de styliser les éléments HTML afin d’offrir une interface esthétique, cohérente et responsive. Nous avons utilisé des règles CSS pour contrôler les couleurs, les marges, les alignements, les tailles de police, et la disposition des blocs, améliorant ainsi l’expérience visuelle des utilisateurs.

****

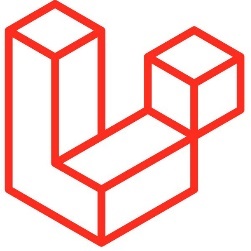
**Figure 9: Logo Javascript**

**JavaScript** [5] est un langage de programmation orienté objet, principalement utilisé pour rendre les pages web interactives. Dans notre projet, JavaScript a été utilisé pour gérer les événements utilisateurs, valider les formulaires côté client, et mettre à jour dynamiquement certaines parties de l’interface sans recharger la page. Cela a apporté une meilleure fluidité et réactivité à notre application.



**Figure 10:Logo Bootstrap**

**Bootstrap** [6] est un framework CSS open-source développé par Twitter, conçu pour faciliter le développement d’interfaces web responsive et modernes. Dans notre projet, Bootstrap a été utilisé pour accélérer la mise en forme des pages grâce à ses composants prêts à l’emploi (boutons, formulaires, cartes, barres de navigation, etc.) et à son système de grille. Il nous a permis de créer une interface cohérente et adaptative sur différents appareils, tout en réduisant le temps de développement lié au design.



**Figure 11:Logo Laravel**

**Laravel** [7] est un framework PHP moderne qui suit l’architecture MVC (Modèle-Vue-Contrôleur). Dans notre projet, Laravel a servi à développer la partie backend de l’application de manière structurée et sécurisée. Il nous a offert de nombreuses fonctionnalités prêtes à l’emploi, comme la gestion des routes, l’authentification, l’ORM Eloquent, et le système de migration de la base de données, rendant le développement plus rapide et fiable.



**Figure 12:Logo XAMPP**

**XAMPP** [8] est un environnement de développement web local qui regroupe Apache, MySQL (ou MariaDB), PHP et Perl. Nous avons utilisé XAMPP pour héberger localement notre application web et exécuter nos scripts PHP pendant le développement. Il nous a offert un serveur complet, simple à installer et à configurer, ce qui a facilité les phases de test et de débogage sans avoir besoin d’un serveur distant.



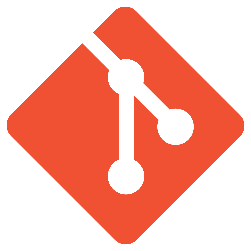
**Figure 13:Logo phpMyAdmin**

**PhpMyAdmin** [9] est une application web libre qui permet de gérer facilement des bases de données MySQL via une interface graphique. Nous l’avons utilisée pour concevoir, visualiser et modifier notre base de données. Elle nous a facilité la gestion des tables, des relations, des requêtes SQL, et des utilisateurs, tout en nous faisant gagner du temps dans le développement backend.



**Figure 14:Logo Enterprise Architect**

**Enterprise Architect**  [10] est un outil de modélisation UML développé par Sparx Systems. Nous l’avons utilisé pour concevoir les modèles nécessaires à la phase de conception, notamment les diagrammes de cas d’utilisation, de classes, de séquence et de base de données. Ces schémas ont facilité la compréhension de l’architecture de notre application et assuré une bonne communication entre les membres de l’équipe.



**Figure 15:Logo Git**

**Git** [11] est un système de gestion de versions distribué, qui permet de suivre l’évolution d’un projet logiciel, de gérer les branches de développement et de collaborer efficacement. Nous avons utilisé Git pour enregistrer chaque modification du code source, expérimenter de nouvelles fonctionnalités sans perturber le projet principal, et maintenir un historique clair des contributions.



**Figure 16:Logo GitHub**

**GitHub** [12] est une plateforme de développement collaboratif qui repose sur Git. Elle nous a permis d’héberger notre projet à distance, de collaborer en équipe, de partager le code, d’utiliser un système de suivi des problèmes (issues), et de gérer les versions. GitHub a joué un rôle important dans l’organisation du travail en groupe et la centralisation des ressources.

1. **Les interfaces graphiques de l’application**

# Références

|  |  |
| --- | --- |
| [1] | «L'Ecole Polytechnique des Génie,» 1 juin 2025. [En ligne]. Available: https://epg.ma/. |
| [2] | . B. Grady, R. James et J. Ivar, The Unified Modeling Language User Guide, 2005. |
| [3] | «HTML (HyperText Markup Language) | MDN,» 2 6 2025. [En ligne]. Available: https://developer.mozilla.org/fr/docs/Web/HTML. |
| [4] | «CSS : Feuilles de style en cascade | MDN,» 2 6 2025. [En ligne]. Available: https://developer.mozilla.org/fr/docs/Web/CSS. |
| [5] | «JavaScript | MDN,» 2 6 2025. [En ligne]. Available: https://developer.mozilla.org/fr/docs/Web/JavaScript. |
| [6] | «Bootstrap · The most popular HTML, CSS, and JS library in the world.,» 2 6 2025. [En ligne]. Available: https://getbootstrap.com/. |
| [7] | «Laravel - The PHP Framework For Web Artisans,» 2 6 2025. [En ligne]. Available: https://laravel.com/. |
| [8] | «XAMPP - Wikipedia,» 2 6 2025. [En ligne]. Available: https://en.wikipedia.org/wiki/XAMPP. |
| [9] | «phpMyAdmin,» 2 6 2025. [En ligne]. Available: https://www.phpmyadmin.net/. |
| [10] | «Enterprise Architect - Wikipedia,» 2 6 2025. [En ligne]. Available: https://en.wikipedia.org/wiki/Enterprise\_Architect\_(software). |
| [11] | «Git,» 2 6 2025. [En ligne]. Available: https://git-scm.com/. |
| [12] | «GitHub,» 2 6 2025. [En ligne]. Available: https://github.com/. |