République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara – Boumerdès
Faculté des Sciences
Département d'Informatique

Filière : Informatique
Spécialité : Système d'information

Mémoire de fin de Cycle en vue de l'obtention du
Diplôme de Licence

Thème

Gestion de pointage avec reconnaissance faciale

Présenté par :

- *AMEZIANE Katia*
- *HARRACHE Lilia*
- *FIAK Khadidja*

Stage pratique réalisé au : ***Sonatrach***

Soutenu le / /2026 Devant le jury composé de

Mr. MOKRANI Hocine	:	Encadrant
Mme. SAADI Khaoula	:	Promoteur (thème externe)
Mme. AHMED NACER Amina	:	Examinateur

Table des matières

1 Entreprise d'accueil et Problématique **6**
1.1 Présentation de l'entreprise SONATRACH 7
1.1.1 Les missions de SONATRACH 7
1.1.2 Organigramme de "SONATRACH" 8
1.2 Présentation de la division d'exploration 8
1.3 Problématique 9
1.4 Objectifs du travail 10

2 Analyse des besoins et conception du système **11**
2.1 Modélisation UML 12
2.2 Méthode de conception 12
2.3 Analyse des besoins 12
2.3.1 Besoins fonctionnels 12
2.3.2 Besoins non fonctionnels 13
2.3.3 Identification des acteurs 13
2.3.4 Diagramme de cas d'utilisation et table descriptive 13
2.4 Conception du système 16
2.4.1 Diagrammes de séquence et tables descriptives 16
2.4.2 Diagramme de classes 28
2.4.3 Règles de gestion 28
2.4.4 Dictionnaire de données 29
2.4.5 Diagramme de classes général 30
2.4.6 Règles de transformation 30
2.4.7 Modèle Logique de Données (MLD) : 31

3 Architecture et implémentation du système **32**
3.1 Architecture générale de l'application 33
3.1.1 Modèle MVC 33
3.1.2 Diagramme de déploiement 34
3.2 Environnement de travail et outils utilisés 34
3.2.1 Matériel informatique (hardware) 34
3.2.2 Environnement logiciel (software) 35
3.3 Les interfaces de l'application web 37

Table des figures

1.1 Logo de SONATRACH [1] . . . 7
1.2 Organigramme de SONATRACH [2] . . . 8
1.3 Organigramme de la division Exploration [1] . . . 9

2.1 Diagramme de cas d'utilisation général [6] . . . 14
2.2 Diagramme de séquence du cas d'utilisation « S'authentifier » [5] . . . 16
2.3 Diagramme de séquence du cas d'utilisation « Scanner un visage » [5] . . . 18
2.4 Diagramme de séquence du cas d'utilisation « Gérer un visiteur » [5] . . . 19
2.5 Diagramme de séquence du cas d'utilisation « Consulter le pointage » [5] . . . 20
2.6 Diagramme de séquence du cas d'utilisation « Importer une justification » [5] . . . 21
2.7 Diagramme de séquence du cas d'utilisation « Consulter les pointages des employés » [5] . . . 22
2.8 Diagramme de séquence du cas d'utilisation « Exporter les données » [5] . . . 23
2.9 Diagramme de séquence du cas d'utilisation « Traiter les justifications » [5] . . . 24
2.10 Diagramme de séquence du cas d'utilisation « Ajouter un employé » [5] . . . 25
2.11 Diagramme de séquence du cas d'utilisation « Modifier un employé » [5] . . . 26
2.12 Diagramme de séquence du cas d'utilisation « Supprimer un employé »[5] . . . 27
2.13 Diagramme de classes général [6] . . . 30

3.1 Schéma de l'architecture Modèle-Vue-Contrôleur (MVC) . . . 33
3.2 Diagramme de déploiement . . . 34
3.3 Interface d'Accueil . . . 37
3.4 L'interface dédiée à l'authentification . . . 37
3.5 Interface de l'espace agent . . . 37
3.6 Interface de l'espace RH . . . 38
3.7 Interface de l'espace Employe . . . 38

Liste des tableaux

2.1 Table descriptive des cas d'utilisation général 16
2.2 Table descriptive du cas d'utilisation « S'authentifier » 17
2.3 Table descriptive du cas d'utilisation « Scanner le visage » 19
2.4 Table descriptive du cas d'utilisation«Gérer un visiteur» 20
2.6 Table descriptive du cas d'utilisation « Importer une justification » 22
2.7 Table descriptive du cas d'utilisation « Consulter les pointages des employés » . 23
2.9 Table descriptive du cas d'utilisation « Traiter une justification » 25
2.10 Table descriptive du cas d'utilisation « Ajouter un employé » 25
2.11 Table descriptive du cas d'utilisation « Modifier un employé » 26
2.12 Table descriptive du cas d'utilisation « Supprimer un employé » 27
2.13 Dictionnaire de données 29

3.1 Caractéristiques des machines utilisées 35
3.2 Comparaison des bibliothèques de reconnaissance faciale 36

Introduction générale

Dans un contexte professionnel en constante évolution, les entreprises cherchent à améliorer leur efficacité et à moderniser leurs processus pour rester compétitives. **SONATRACH**, pilier de l'industrie énergétique algérienne, s'inscrit pleinement dans cette dynamique de transformation digitale et d'innovation.

Le suivi des présences et la gestion du personnel constituent des enjeux cruciaux pour la performance des entreprises. Le système actuel de gestion des ressources humaines, **ReSHum**, bien qu'ayant servi l'entreprise pendant de nombreuses années, présente aujourd'hui des limitations importantes : mises à jour fastidieuses, vulnérabilité des données et suivi manuel des présences. Ces contraintes freinent l'efficacité du service RH et limitent sa contribution optimale à la performance globale de **S**ONATRACH.

C'est dans ce cadre que s'inscrit notre projet de fin de cycle, qui consiste en la conception et la réalisation d'un **système de pointage intelligent basé sur la reconnaissance faciale**. Ce système permet d'automatiser le suivi des présences, de sécuriser l'accès aux locaux et d'assurer une gestion fiable et rapide des données du personnel. Il offre ainsi une solution complète, moderne et adaptée aux besoins actuels de l'entreprise.

Ce mémoire est structuré pour présenter successivement le contexte théorique, la conception technique du système, son implémentation et l'évaluation des résultats obtenus, démontrant l'apport concret de cette solution pour la gestion des ressources humaines et le contrôle des présences chez **S**ONATRACH.

Chapitre 1

Entreprise d'accueil et Problématique

Introduction

La réalisation de tout projet au sein d'une entreprise nécessite une compréhension approfondie de son environnement organisationnel et fonctionnel. Cette étape permet d'identifier les besoins réels et de définir les objectifs du projet.

Dans le cadre de notre Projet de Fin de Cycle, nous avons effectué notre stage au sein de SONATRACH. Ce chapitre est consacré à la présentation de l'entreprise ainsi qu'à l'étude de l'existant, afin d'analyser le fonctionnement actuel du système et d'identifier les besoins ayant motivé la conception de la solution proposée

1.1 Présentation de l'entreprise SONATRACH

SONATRACH, « Société Nationale pour la Recherche, la Production, le Transport, la Transformation et la Commercialisation des Hydrocarbures », est la compagnie nationale des hydrocarbures d'Algérie. Fondée le **31 décembre 1963**, peu après l'indépendance nationale, son but principal est la gestion, l'exploitation et la valorisation des ressources pétrolières et gazières du pays [1].Afin d'illustrer l'identité visuelle de l'entreprise, la figure suivante présente le logo officiel de SONATRACH.



FIGURE 1.1 – Logo de SONATRACH [1]

Dans ses premières années, SONATRACH a progressivement renforcé le contrôle national sur les ressources énergétiques, qui étaient auparavant gérées en grande partie par des entreprises étrangères. Cette évolution a abouti à la nationalisation complète du secteur pétrolier en **février 1971** [1], donnant ainsi à l'Algérie un contrôle total sur ses ressources énergétiques. Depuis lors, SONATRACH s'est imposée comme l'un des acteurs majeurs du secteur énergétique en Afrique et dans le monde arabe [1].

1.1.1 Les missions de SONATRACH

Les missions de SONATRACH s'inscrivent dans une stratégie globale de développement et de valorisation des ressources énergétiques nationales. Elles se déclinent comme suit :

- **Exploration :** recherche, prospection et évaluation de nouveaux gisements d'hydrocarbures [1].
- **Production :** développement, exploitation et gestion des gisements de pétrole et de gaz [1].
- **Transport :** gestion et exploitation du réseau de transport par canalisations (oléoducs et gazoducs) [1].
- **Transformation :** traitement et liquéfaction du gaz naturel ainsi que la valorisation des hydrocarbures [1].

— **Commercialisation :** distribution et vente des hydrocarbures et de leurs dérivés sur les marchés national et international [1].

Grâce à l'ensemble de ces activités intégrées, SONATRACH joue un rôle stratégique majeur dans l'économie algérienne et constitue un pilier essentiel du secteur énergétique national.

1.1.2 Organigramme de "SONATRACH"

L'organisation de SONATRACH repose sur une structure hiérarchique claire, répartissant les responsabilités entre les différentes divisions opérationnelles et fonctionnelles. Au sommet se trouve le **Président Directeur Général**, entouré de comités et de directions centrales. Cette structure se décline ensuite en plusieurs activités stratégiques, parmi lesquelles on trouve l'Activité Exploration-Production, l'Activité Transport par Canalisations, l'Activité Liquéfaction et Séparation, l'Activité Raffinage et Pétrochimie, et l'Activité Commercialisation.La figure ci-dessous illustre cette organisation hiérarchique

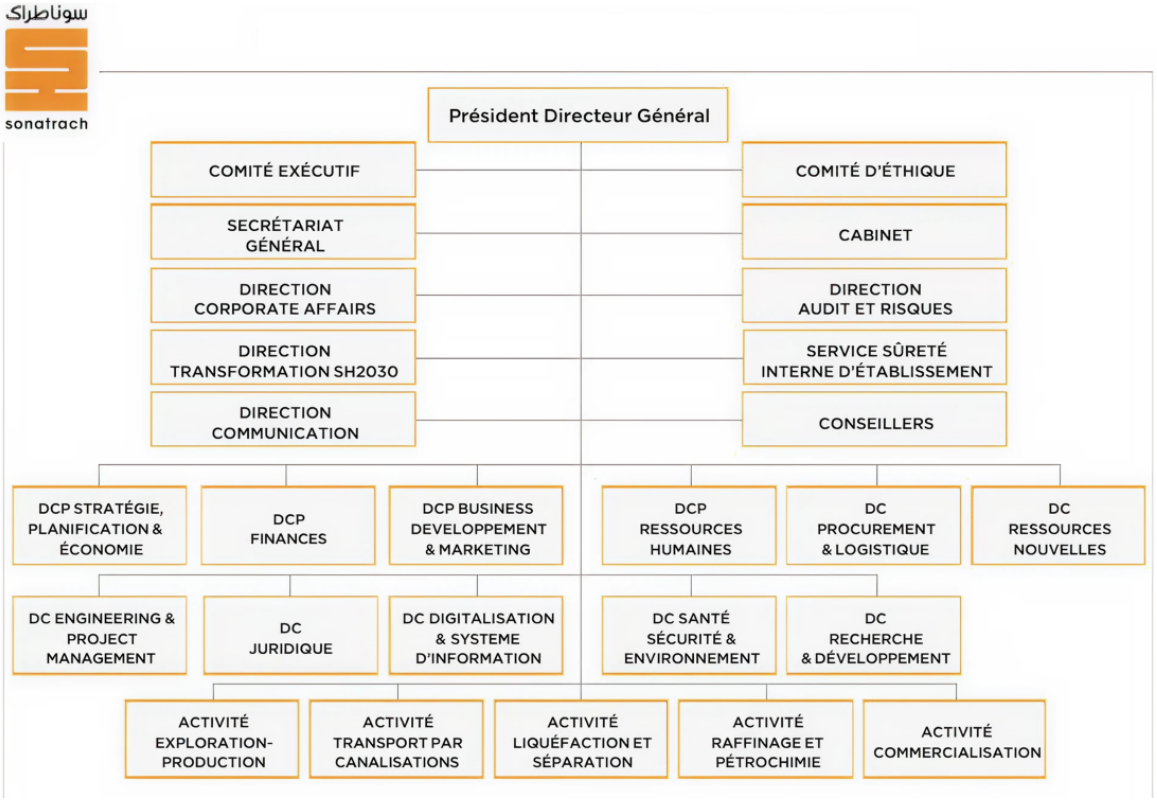


FIGURE 1.2 – Organigramme de SONATRACH [2]

C'est au sein de l'**Activité Exploration-Production** que s'inscrit la **Division Exploration**, là où notre stage pratique a été réalisé. Cette division joue un rôle stratégique majeur dans la découverte et l'évaluation des ressources énergétiques nationales.

1.2 Présentation de la division d'exploration

La division Exploration de SONATRACH est chargée de la recherche et de l'évaluation des gisements d'hydrocarbures. Elle mène des études géologiques et supervise les activités d'exploration afin d'assurer le renouvellement des ressources énergétiques nationales.

Cette division regroupe plusieurs directions et départements spécialisés (planification, finances, ressources humaines et logistique). Notre stage pratique a été réalisé au sein du Département Technologie de l'Information, rattaché à la Direction Data Management et représenté

en vert dans l'organigramme ci-dessous. Ce département nous a permis d'identifier les besoins liés à la gestion du personnel et au suivi des présences

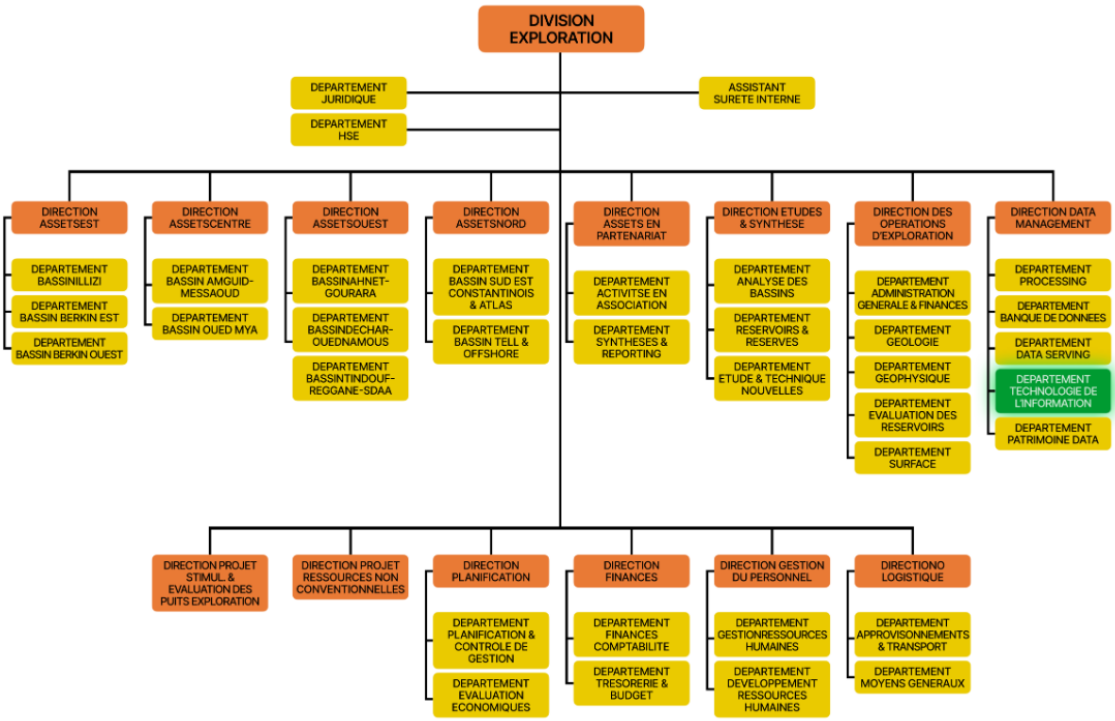


FIGURE 1.3 – Organigramme de la division Exploration [1]

1.3 Problématique

Sonatrach, entreprise nationale opérant dans le secteur des hydrocarbures en Algérie, a connu une croissance importante de ses effectifs, notamment au sein de sa Division Exploration et Production. Cette évolution a engendré une complexité accrue dans la gestion des ressources humaines, rendant les tâches administratives plus exigeantes.

Pour répondre à ces besoins, un logiciel desktop de gestion du personnel, *ReSHum*, a été mis en place. Ce système permet de réaliser plusieurs opérations administratives liées à la gestion des employés. Cependant, malgré son utilité, il présente aujourd'hui certaines limites face aux exigences actuelles et aux standards des solutions modernes.

Parmi les principales insuffisances du logiciel, on peut citer :

- **Mise à jour complexe :** les mises à jour nécessitent des interventions techniques sur chaque poste.
- **Risque de perte de données :** en cas de panne du système ou d'une mauvaise gestion des sauvegardes.
- **Accès limité :** les employés ne peuvent pas consulter directement l'historique de leurs pointages, ce qui les oblige à se déplacer au service des ressources humaines pour obtenir ces informations.
- **Interface peu ergonomique :** l'interface du logiciel est ancienne et peu intuitive, ce qui complique son utilisation quotidienne.
- **Sécurité insuffisante :** absence de mécanismes de sécurité avancés et centralisés.

— **Traitement partiellement manuel :** certaines tâches, comme le suivi des absences, manquent d'automatisation.

Avec l'évolution rapide des technologies web, les entreprises s'orientent vers des solutions plus accessibles, centralisées et sécurisées. Dans ce contexte, il devient nécessaire pour Sonatrach de moderniser son système de gestion des ressources humaines afin d'améliorer son efficacité et de mieux répondre aux besoins actuels et futurs.

1.4 Objectifs du travail

L'objectif principal de ce projet est de concevoir et de développer une plateforme web moderne permettant d'améliorer la gestion du personnel et de dépasser les limites du système actuellement utilisé par la Sonatrach.

Plus spécifiquement, ce travail vise à :

— Développer une plateforme web facilitant la gestion des informations du personnel dans un seul système.
— Mettre en place un système de pointage basé sur la reconnaissance faciale afin d'automatiser l'enregistrement des présences de manière rapide, fiable et sécurisée.
— Faciliter le suivi des absences et des retards grâce à une gestion automatisée et mieux organisée.
— Permettre aux employés de consulter l'historique de leurs pointages directement depuis la plateforme.
— Renforcer la sécurité et la fiabilité des données à travers des mécanismes d'authentification, de contrôle d'accès et de centralisation des informations.
— Offrir une interface moderne et intuitive simplifiant l'utilisation du système pour les différents utilisateurs.
— Faciliter la communication entre le service des ressources humaines et les employés concernant les absences et les demandes de congé, afin de réduire les déplacements et de gagner du temps.

Conclusion

Ce premier chapitre nous a permis de présenter le contexte de notre stage au sein de SONATRACH, d'identifier les limites du système existant et de définir les objectifs de notre projet. Ces éléments constituent la base sur laquelle repose la conception du système, qui sera détaillée dans le chapitre suivant.

Chapitre 2

Analyse des besoins et conception du système

Introduction

Avant de commencer le développement de notre système de pointage par reconnaissance faciale, il est important de bien comprendre ce que le système doit faire et comment il doit fonctionner.

Dans ce chapitre, nous allons d'abord présenter la méthode de conception que nous avons utilisée, qui est **RUP [3]**. Ensuite, nous allons identifier les besoins du système, c'est-à-dire ce que le système doit offrir comme fonctionnalités aux utilisateurs. Enfin, nous allons modéliser le système à l'aide des diagrammes **UML** pour donner une vision claire et complète de son fonctionnement.

2.1 Modélisation UML

UML (Unified Modeling Language) est un langage de modélisation graphique standard utilisé pour représenter la structure et le comportement d'un système logiciel **[4]**. Il permet de visualiser, spécifier et documenter les différents aspects d'un système de manière claire et compréhensible.

Dans le cadre de notre projet, nous avons utilisé UML comme approche de modélisation afin de représenter les acteurs du système, leurs interactions et les principaux scénarios d'utilisation.

2.2 Méthode de conception

Pour mener à bien la conception de notre système, nous avons choisi la méthode **RUP** (Rational Unified Process), qui est une approche itérative et incrémentale du développement logiciel. Cette méthode organise le développement en plusieurs phases bien définies et s'appuie sur les diagrammes UML pour la modélisation du système.

RUP se base sur quatre phases principales :

— **Inception :** définition des objectifs et du périmètre du projet.
— **Élaboration :** analyse des besoins et conception de l'architecture du système.
— **Construction :** développement et implémentation du système.
— **Transition :** déploiement et mise en production du système.

Nous avons opté pour cette méthode car elle est particulièrement adaptée aux projets nécessitant une modélisation rigoureuse et une organisation structurée du développement.

2.3 Analyse des besoins

Avant de concevoir notre système, il est essentiel d'identifier et d'analyser les besoins des utilisateurs. Cette étape permet de définir clairement ce que le système doit faire et les contraintes qu'il doit respecter. On distingue deux types de besoins : les besoins fonctionnels et les besoins non fonctionnels.

2.3.1 Besoins fonctionnels

Les besoins fonctionnels décrivent les fonctionnalités que le système doit offrir à ses utilisateurs. Dans notre projet, les besoins fonctionnels sont les suivants :

— **Agent de sécurité :**
  — S'authentifier au système.
  — Lancer la reconnaissance faciale via la webcam.
  — Consulter les logs de pointage des employés.

  - Enregistrer un visiteur non reconnu via un formulaire.
  - Imprimer le badge visiteur.
- **Employé :**
  - S'authentifier au système.
  - Consulter ses pointages (arrivée et sortie).
  - Soumettre un justificatif d'absence.
- **Responsable RH :**
  - S'authentifier au système.
  - Consulter la liste des employés.
  - Ajouter, modifier et supprimer un employé.
  - Consulter les heures d'arrivée et de sortie des employés.
  - Traiter les justificatifs d'absence (accepter ou refuser).

2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels définissent les contraintes et les critères de qualité que le système doit respecter. Dans notre projet, les besoins non fonctionnels sont les suivants :

- **Performance :** le système doit reconnaître un visage et enregistrer le pointage en temps réel, sans délai perceptible.
- **Sécurité :** l'accès au système est protégé par une authentification par email et mot de passe, avec une redirection automatique selon le rôle de l'utilisateur.
- **Fiabilité :** le système doit fonctionner de manière stable et continue, en garantissant l'exactitude des données de pointage enregistrées.
- **Ergonomie :** l'interface doit être simple, intuitive et facile à utiliser pour tous les types d'utilisateurs.
- **Compatibilité :** le système doit être accessible depuis différents navigateurs web et systèmes d'exploitation.
- **Maintenabilité :** le code doit être bien structuré et documenté pour faciliter les futures modifications et améliorations.

2.3.3 Identification des acteurs

Un acteur représente toute entité externe qui interagit avec le système. Dans notre système de pointage par reconnaissance faciale, les acteurs identifiés sont les suivants :

- **Agent de sécurité :** il s'authentifie au système, lance la reconnaissance faciale et prend en charge l'enregistrement des visiteurs en cas de non-reconnaissance.
- **Employé :** il accède à son espace personnel après authentification afin de consulter ses informations de pointage, ses absences et de soumettre des justifications.
- **Responsable RH :** il s'authentifie au système, puis accède au tableau de bord pour consulter les données de pointage des employés, traiter les justificatifs d'absence et gérer les informations du personnel.

2.3.4 Diagramme de cas d'utilisation et table descriptive

a. Diagramme de cas d'utilisation général

Le diagramme ci-dessous représente l'ensemble des cas d'utilisation de notre système de pointage par reconnaissance faciale, pour les trois acteurs identifiés : l'Agent de sécurité, l'Employé et le Responsable RH.

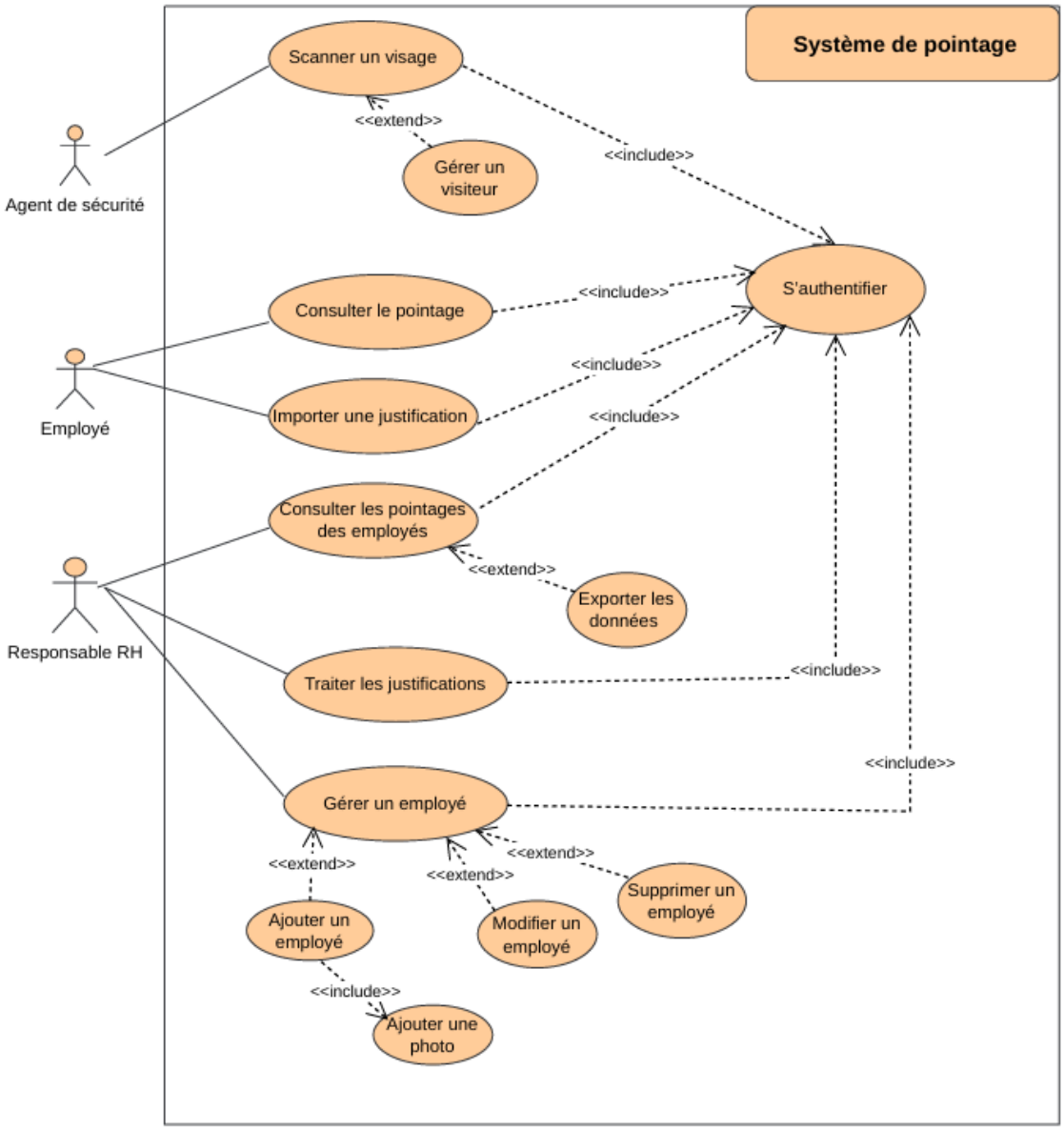


FIGURE 2.1 – Diagramme de cas d'utilisation général [6]

b. Table descriptive

NOM	ACTEUR	DESCRIPTION	PRÉCONDITION	RÉSULTAT
S'Authentifier	Agent / Employé / Responsable RH	L'utilisateur saisit ses identifiants (Email/mot de passe). Le système vérifie les identifiants et redirige vers la page correspondant à son rôle.	L'utilisateur possède un compte actif dans le système.	Session ouverte, redirection vers la page du rôle (Agent, Employé ou RH).
Scanner un visage	Agent de sécurité	L'agent active la webcam et lance la reconnaissance faciale.	Agent authentifié, webcam disponible.	Employé reconnu → accès accordé et présence enregistrée. Sinon → alerte déclenchée.

NOM	ACTEUR	DESCRIPTION	PRÉCONDITION	RÉSULTAT
Gérer un visiteur	Agent de sécurité	Lorsque le visage n'est pas reconnu, l'agent confirme qu'il s'agit d'un visiteur, remplit un formulaire (nom, pièce d'identité, motif) et lui accorde l'accès.	Visage non reconnu par le système, alerte déclenchée.	Visiteur enregistré dans le journal et accès accordé.
Consulter le pointage	Employé	L'employé consulte son historique de présence, ses absences et ses retards depuis son espace personnel.	Employé authentifié.	Tableau de bord personnel affiché avec les données de présence.
Importer une justification	Employé	L'employé soumet un document justifiant une absence (certificat médical, autorisation de congé, etc.). Le fichier est envoyé au RH pour validation.	Employé authentifié, absence enregistrée dans le système.	Justification enregistrée, notification envoyée au RH.
Ajouter un employé	RH	Le RH crée un nouveau profil employé : informations personnelles, poste, département, photo et données faciales biométriques enregistrées dans la base.	Utilisateur RH authentifié.	Nouvel employé ajouté, profil actif et reconnaissable par le système.
Modifier un employé	RH	Le RH met à jour les informations d'un employé existant (poste, département, coordonnées). Peut inclure la mise à jour de la photo faciale.	Employé existant dans la base de données.	Profil employé mis à jour dans le système.
Supprimer un employé	RH	Le RH supprime le profil d'un employé qui a quitté l'entreprise. Les données biométriques associées sont également supprimées.	Profil employé existant, RH authentifié.	Profil et données faciales supprimés, employé inactif.
Traiter les justifications	RH	Le RH consulte les justifications soumises par les employés et les approuve ou les rejette.	Justification soumise par un employé, RH authentifié.	Justification approuvée ou rejetée, statut d'absence mis à jour.

NOM	ACTEUR	DESCRIPTION	PRÉCONDITION	RÉSULTAT
Consulter les pointages des employés	RH	Le RH visualise les horaires d'arrivée/départ, retards et absences de tous les employés pour une période donnée.	RH authentifié, données de présence disponibles.	Tableau des pointages affiché avec filtres par date et employé.
Exporter les données	RH	Le RH exporte le rapport de pointages au format Excel ou PDF pour archivage ou transmission à la direction.	Rapport de pointages disponible.	Fichier Excel ou PDF généré et téléchargé sur le poste du RH.

TABLE 2.1: Table descriptive des cas d'utilisation général

2.4 Conception du système

La conception du système permet de définir l'architecture interne et le fonctionnement détaillé de notre application. Pour cela, nous utilisons deux types de diagrammes UML : les diagrammes de séquence et le diagramme de classes.

2.4.1 Diagrammes de séquence et tables descriptives

a1. Diagramme de séquence du cas d'utilisation « S'authentifier »

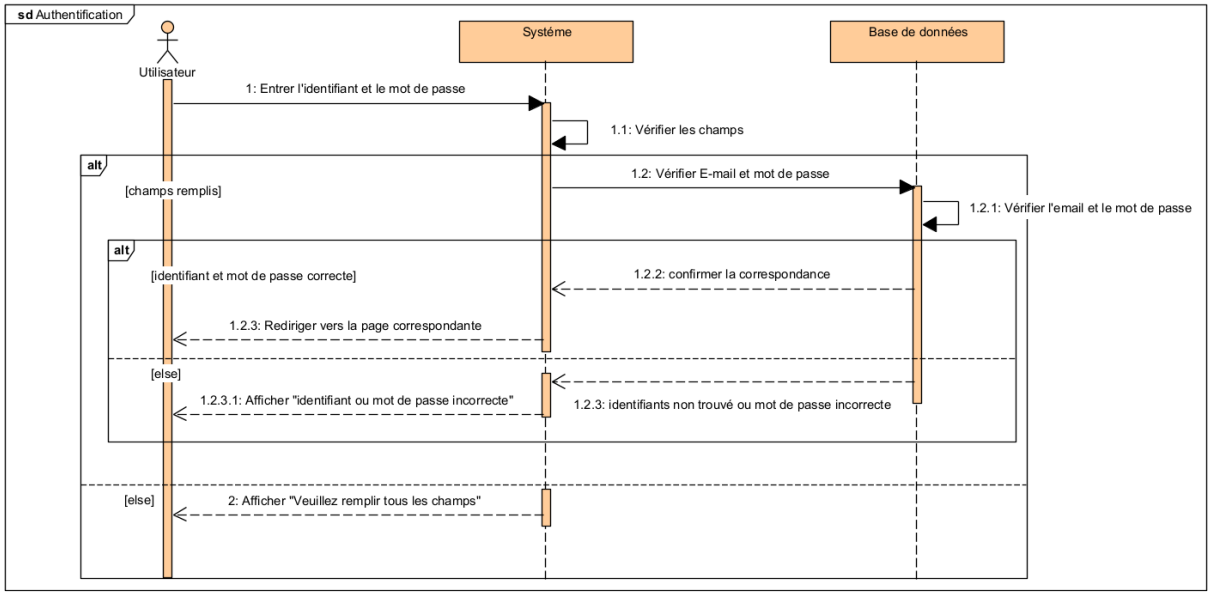


FIGURE 2.2 – Diagramme de séquence du cas d'utilisation « S'authentifier » [5]

a2. Table descriptive du cas d'utilisation « S'authentifier »

Acteur	Utilisateur
Nom du cas d'utilisation	Authentification
Objectif	Permettre à l'utilisateur d'accéder au système en saisissant ses identifiants (email et mot de passe) afin d'être redirigé vers l'interface correspondante à son rôle.
Pré-condition	L'utilisateur possède un email et un mot de passe valides.
Scénario nominal	1. Entrer l'identifiant (email) et le mot de passe. 1.1 Le système vérifie que tous les champs sont remplis.
Scénario alternatif	**[Champs remplis & identifiants corrects] :** 1.2 Le système envoie les identifiants à la base de données. 1.2.1 La base de données vérifie l'email et le mot de passe. 1.2.2 La base de données confirme la correspondance. 1.2.3 Le système redirige l'utilisateur vers son espace. **[Identifiants incorrects] :** 1.2.3 La base de données retourne une erreur. 1.2.3.1 Le système affiche : "Identifiant ou mot de passe incorrect". **[Champs non remplis] :** 2. Le système affiche : "Veuillez remplir tous les champs".
Post-condition	• **[Succès] :** L'utilisateur est authentifié et redirigé vers son espace. • **[Échec identifiants] :** L'accès est refusé et un message d'erreur est affiché. • **[Champs vides] :** Le formulaire reste affiché avec un message d'avertissement.

TABLE 2.2 – Table descriptive du cas d'utilisation « S'authentifier »

b1. Diagramme de séquence du cas d'utilisation « Scanner un visage »

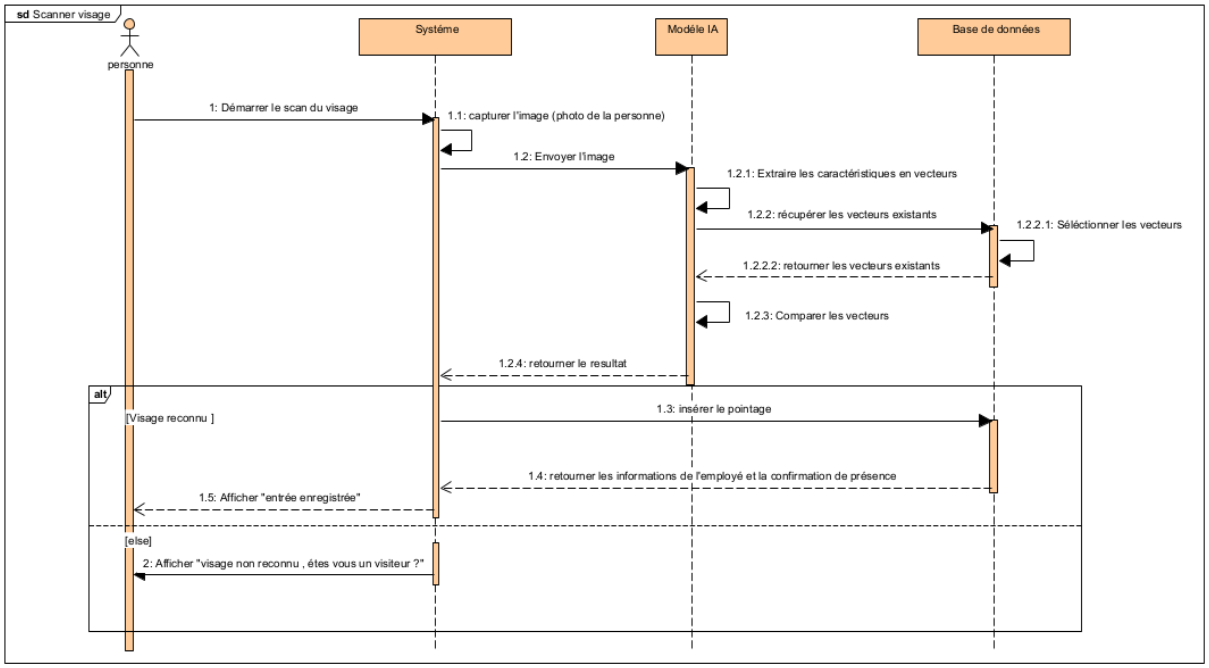


FIGURE 2.3 – Diagramme de séquence du cas d'utilisation « Scanner un visage » [5]

b2. Table descriptive du cas d'utilisation « Scanner le visage »

Acteur	Personne (Employé / Visiteur)
Nom du cas d'utilisation	Scanner le visage
Objectif	Permettre à une personne d'être identifiée par reconnaissance faciale afin d'enregistrer automatiquement son pointage ou de détecter qu'elle est un visiteur.
Pré-condition	- Le système est opérationnel et la caméra est active.
Scénario nominal	1. démarrer le scan du visage. 1.1. Le système capture l'image (photo de la personne). 1.2. Le système envoie l'image au modèle IA. 1.2.1. Le modèle IA extrait les caractéristiques faciales sous forme de vecteurs. 1.2.2. Le modèle IA récupère les vecteurs existants depuis la base de données. 1.2.2.1. La base de données sélectionne les vecteurs correspondants. 1.2.2.2. La base de données retourne les vecteurs existants au modèle IA. 1.2.3. Le modèle IA compare les vecteurs extraits avec les vecteurs existants. 1.2.4. Le modèle IA retourne le résultat de la comparaison au système.
Scénario alternatif	**[Visage reconnu]** : 1.3. Le système insère le pointage dans la base de données. 1.4. La base de données retourne les informations de l'employé et la confirmation de présence. 1.5. Le système affiche "Entrée enregistrée" à la personne. **[else Visage non reconnu]** : 2. Le système affiche "Visage non reconnu, êtes-vous un visiteur ?" à la personne.

Post-condition	**[Succès]** : le pointage de l'employé est enregistré dans la base de données et une confirmation est affichée. **[Échec]** : le visage n'est pas reconnu ; un message d'alerte est affiché invitant la personne à s'identifier comme visiteur.

TABLE 2.3: Table descriptive du cas d'utilisation « Scanner le visage »

c1. Diagramme de séquence du cas d'utilisation « Gérer un visiteur »

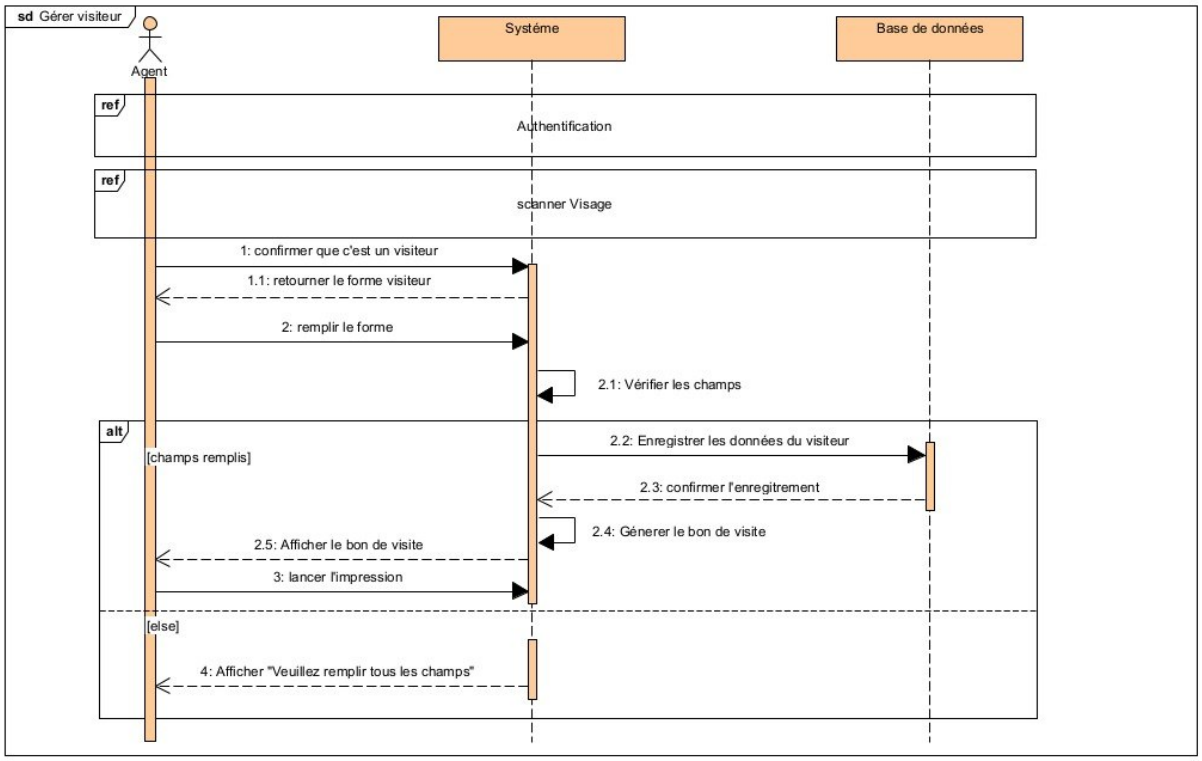


FIGURE 2.4 – Diagramme de séquence du cas d'utilisation « Gérer un visiteur » [5]

c2. Table descriptive du cas d'utilisation « Gérer visiteur »

Acteur	Agent
Nom du cas d'utilisation	Gérer visiteur
Objectif	Permettre à l'agent d'enregistrer un visiteur et de générer automatiquement un bon de visite.
Pré-condition	- L'agent est authentifié dans le système. - Le scan du visage a été effectué et le visage n'a pas été reconnu.
Scénario nominal	1. confirmer que c'est un visiteur. 1.1. Le système retourne le formulaire visiteur. 2. remplir le formulaire. 2.1. Le système vérifie les champs.

Scénario alternatif	**[Champs remplis] :** 2.2. Le système enregistre les données du visiteur dans la base de données. 2.3. La base de données confirme l'enregistrement au système. 2.4. Le système génère le bon de visite. 2.5. Le système affiche le bon de visite à l'agent. 3. lancer l'impression du bon de visite **[else - Champs non remplis] :** 4. Le système affiche « Veuillez remplir tous les champs ».
Post-condition	**[Succes] :** le visiteur est enregistré dans la base de données et un bon de visite est généré et imprimé. **[Echec] :** aucun enregistrement n'est effectué ; un message d'erreur est affiché à l'agent.

TABLE 2.4: Table descriptive du cas d'utilisation«Gérer un visiteur»

d1. Diagramme de séquence du cas d'utilisation « Consulter le pointage »

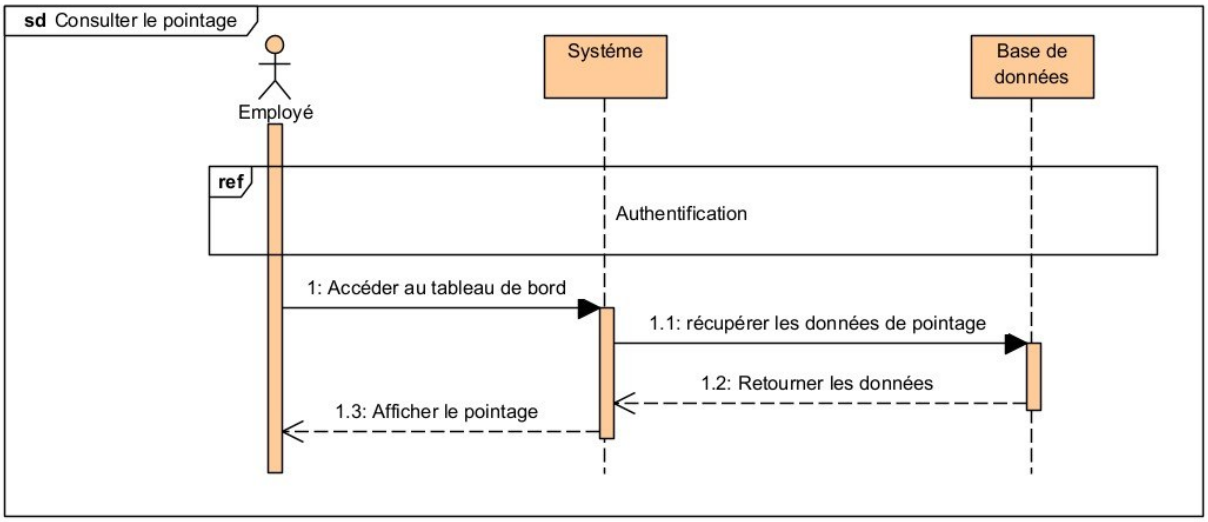


FIGURE 2.5 – Diagramme de séquence du cas d'utilisation « Consulter le pointage » [5]

d2. Table descriptive du cas d'utilisation « Consulter le pointage »

Acteur	Employé
Nom du cas d'utilisation	Consulter le pointage
Objectif	Permettre à l'employé de consulter son historique de pointage (arrivées et départs) depuis son tableau de bord.
Pré-condition	L'employé est authentifié dans le système.
Scénario nominal	1. accéder au tableau de bord. 1.1. Le système envoie récupérer les données de pointage. 1.2. La base de données retourne les données de pointage. 1.3. Le système affiche les données de pointage à l'employé.
Scénario alternatif	Aucun scénario alternatif
Post-condition	Les données de pointage de l'employé sont affichées correctement dans le tableau de bord.

e1. Diagramme de séquence du cas d'utilisation « Importer une justification »

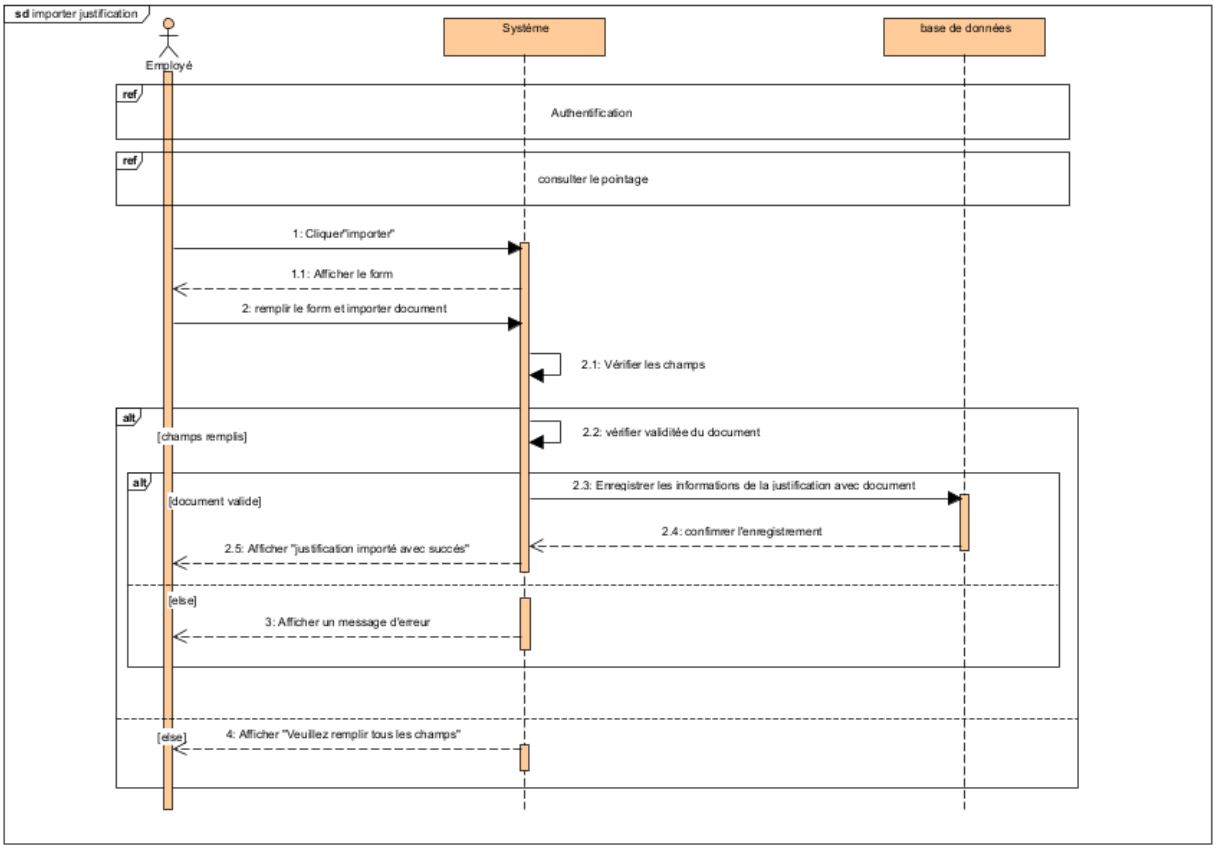


FIGURE 2.6 – Diagramme de séquence du cas d'utilisation « Importer une justification » [5]

e2. Table descriptive du cas d'utilisation « Importer une justification »

Acteur	Employe
Cas d'utilisation	Importer une justification
Objectif	Permettre a l'employé d'importer une justification associée a une absence, en remplissant un formulaire et en joignant un document valide.
Precondition	1. L'employé est authentifié dans le système.
Scenario nominal	1. Cliquer sur "importer". 1.1. Le système affiche le formulaire d'importation. 2. remplir le formulaire et importer le document. 2.1. Le système verifier les champs du formulaire. 2.2. Le système verifier la validité du document. 2.3. Le système enregistre les informations de la justification avec le document dans la base de données. 2.4. La base de données confirme l'enregistrement. 2.5. Le système affiche le message "Justification importée avec succès".

Scenario alternatif	**[Champs remplis] :** 2.2. Le système verifier la validité du document. **[document valide] :** 2.3. Le système enregistre les informations de la justification avec le document dans la base de données. 2.4. La base de données confirme l'enregistrement. 2.5. Le système affiche le message "Justification importée avec succès". **[else - document non valide] :** 3. Le système afficher lr message d'erreur. **[else - Champs non remplis] :** 4. Le système affiche "Veuillez remplir tous les champs".
Post-condition	**[Succès] :** La justification est enregistrée dans la base de données et le message de confirmation est affiché. **[Echec] (document invalide) :** Aucun enregistrement effectué, message d'erreur affiché a l'employé. **[Echec] (champs manquants) :** Aucun enregistrement effectué, message de validation affiché a l'employé.

TABLE 2.6: Table descriptive du cas d'utilisation « Importer une justification »

f1. Diagramme de séquence du cas d'utilisation « Consulter les pointages des employés »

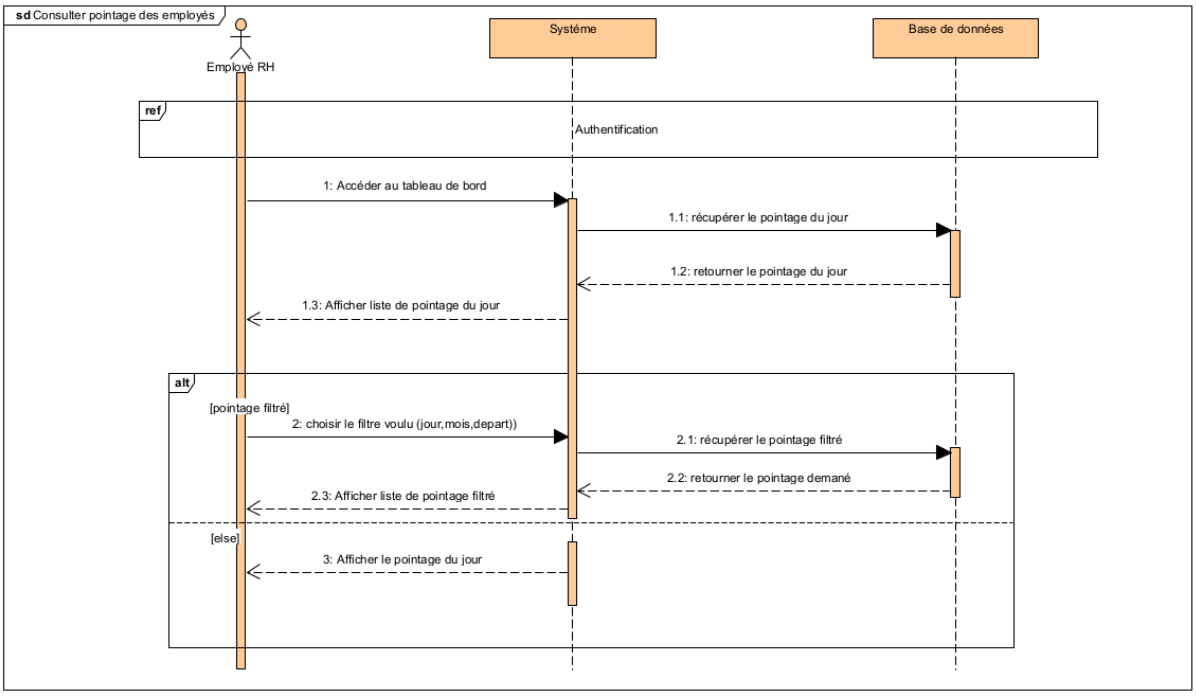


FIGURE 2.7 – Diagramme de séquence du cas d'utilisation « Consulter les pointages des employés » [5]

f2. Table descriptive du cas d'utilisation « Consulter les pointages des employés »

Acteur	Employe RH
Cas d'utilisation	Consulter le pointage des employes
Objectif	Permettre a employé RH de consulter les pointages du jour par defaut, et d'accéder a l'historique via des filtres (par mois ou par jour ou dep).

Precondition	1. L'employè RH est authentifié dans le système
Scenario nominal	1. acceder au tableau de bord. 1.1. Le système récupère le pointage du jour depuis la base de données. 1.2. La base de données retourne le pointage du jour. 1.3. Le système affiche les pointages du jour.
Scenario alternatif	**[Pointage filtrée] :** 2. choisir le filtre voulu (par jour ou par mois,dep). 2.1. Le système récupère le pointage filtre depuis la base de données. 2.2. La base de données retourne le pointage demandé 2.3. Le système affiche la liste de pointage filtrée.
Post-condition	**[Succès] :** Les pointages (du jour ou filtres) sont affiches correctement a l'employé RH. **[Echec] :** Aucun pointage trouve pour le filtre sélectionné : message indiquant l'absence de données.

TABLE 2.7: Table descriptive du cas d'utilisation « Consulter les pointages des employés »

g1. Diagramme de séquence du cas d'utilisation « Exporter les données »

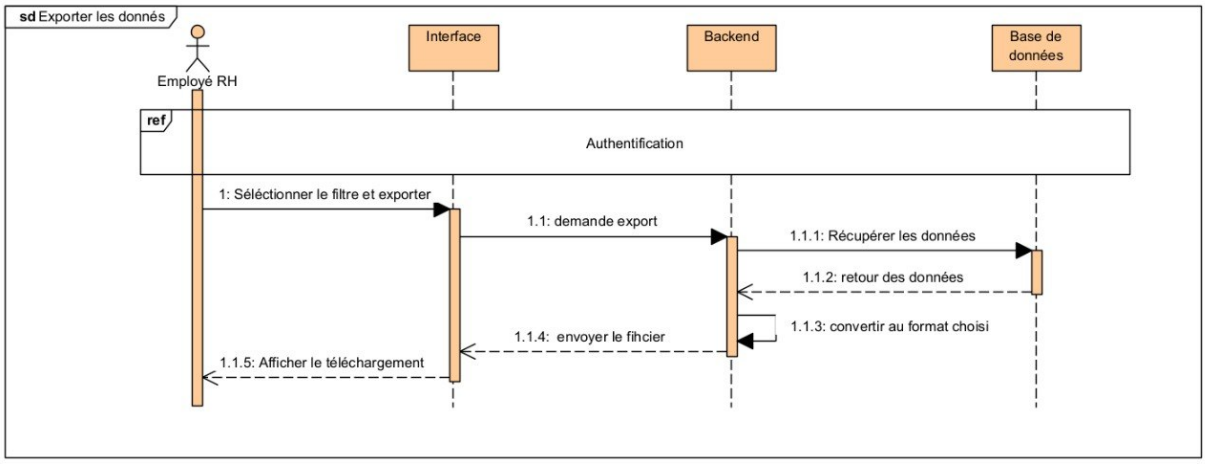


FIGURE 2.8 – Diagramme de séquence du cas d'utilisation « Exporter les données » [5]

g2. Table descriptive du cas d'utilisation « Exporter les données »

Acteur	Employé RH
Nom du cas d'utilisation	Exporter les données
Objectif	Permettre à l'employé RH de sélectionner un filtre et d'exporter les données de pointage dans un format choisi.
Pré-condition	- L'employé RH est authentifié dans le système
Scénario nominal	1. sélectionner le filtre et exporter 1.1 L'interface demande l'export au backend. 1.1.1 Le backend récupère les données depuis la base de données. 1.1.2 La base de données retourne les données 1.1.3 Le backend convertit les données au format choisi 1.1.4 Le backend envoie le fichier généré à l'interface. 1.1.5 L'interface affiche le téléchargement
Scénario alternatif	Aucun scénario alternatif
Post-condition	**[Succès] :** Le fichier est généré et le téléchargement est lancé côté navigateur.

h1. Diagramme de séquence du cas d'utilisation « Traiter les justifications »

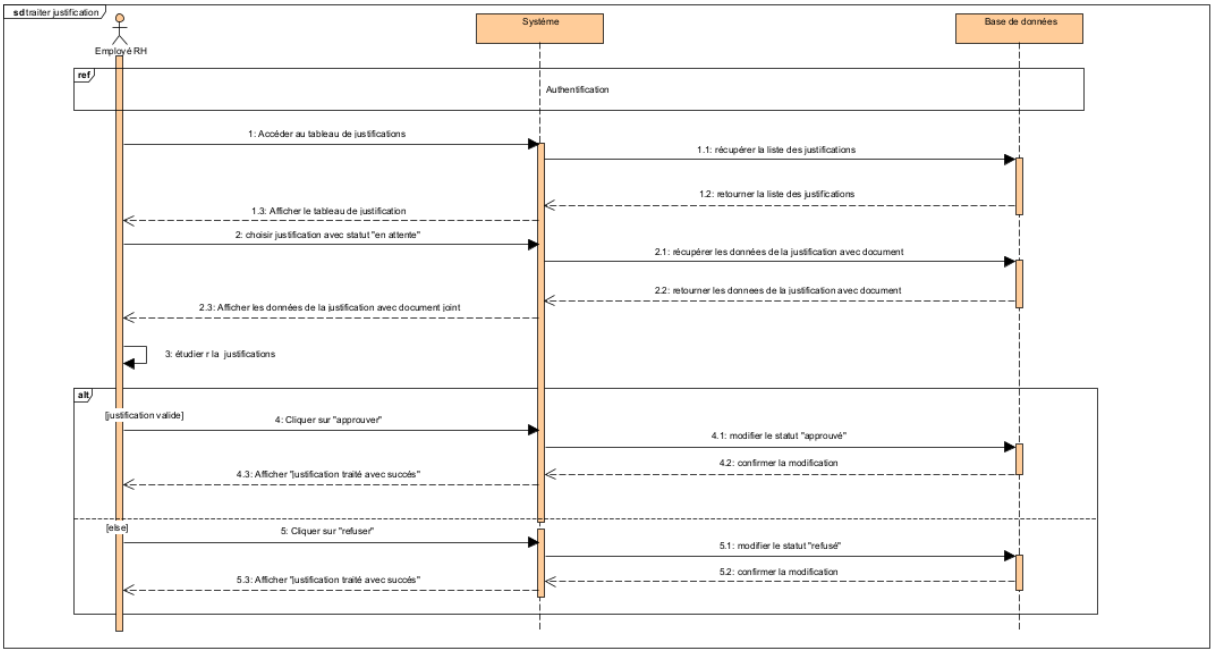


FIGURE 2.9 – Diagramme de séquence du cas d'utilisation « Traiter les justifications » [5]

h2. Table descriptive du cas d'utilisation « Traiter une justification »

Acteur	Employé RH
Nom du cas d'utilisation	Traiter une justification
Objectif	Permettre à l'employé RH de consulter, étudier et traiter les justifications soumises par les employés en les approuvant ou en les refusant.
Pré-condition	- L'employé RH est authentifié dans le système. - Des justifications avec le statut « en attente » existent dans la base de données.
Scénario nominal	1. accéder au tableau de justifications. 1.1. Le système récupère la liste des justifications depuis la base de données. 1.2. La base de données retourne la liste des justifications. 1.3. Le système affiche le tableau de justifications. 2. choisir justification avec le statut « en attente ». 2.1. Le système récupère les données de la justification avec le document. 2.2. La base de données retourne les données de la justification avec le document. 2.3. Le système affiche les données de la justification avec le document joint. 3. étudier la justification.
Scénario alternatif	**[Justification valide] :** 4. cliquer sur « approuver ». 4.1. Le système modifie le statut en « approuvé » dans la base de données. 4.2. La base de données confirme la modification. 4.3. Le système affiche « Justification traitée avec succès ». **[else] Justification non valide :** 5. L'employé RH clique sur « refuser ». 5.1. Le système modifie le statut en « refusé » dans la base de données. 5.2. La base de données confirme la modification. 5.3. Le système affiche « Justification traitée avec succès ».
Post-condition	**[Succès - Approuvé] :** le statut de la justification est mis à jour en « approuvé » dans la base de données.

	[Succès - Refusé] : le statut de la justification est mis à jour en « refusé » dans la base de données.

TABLE 2.9: Table descriptive du cas d'utilisation « Traiter une justification »

i1. Diagramme de séquence du cas d'utilisation « Gérer un employé »

i2. Diagramme de séquence du cas d'utilisation « Ajouter un employé »

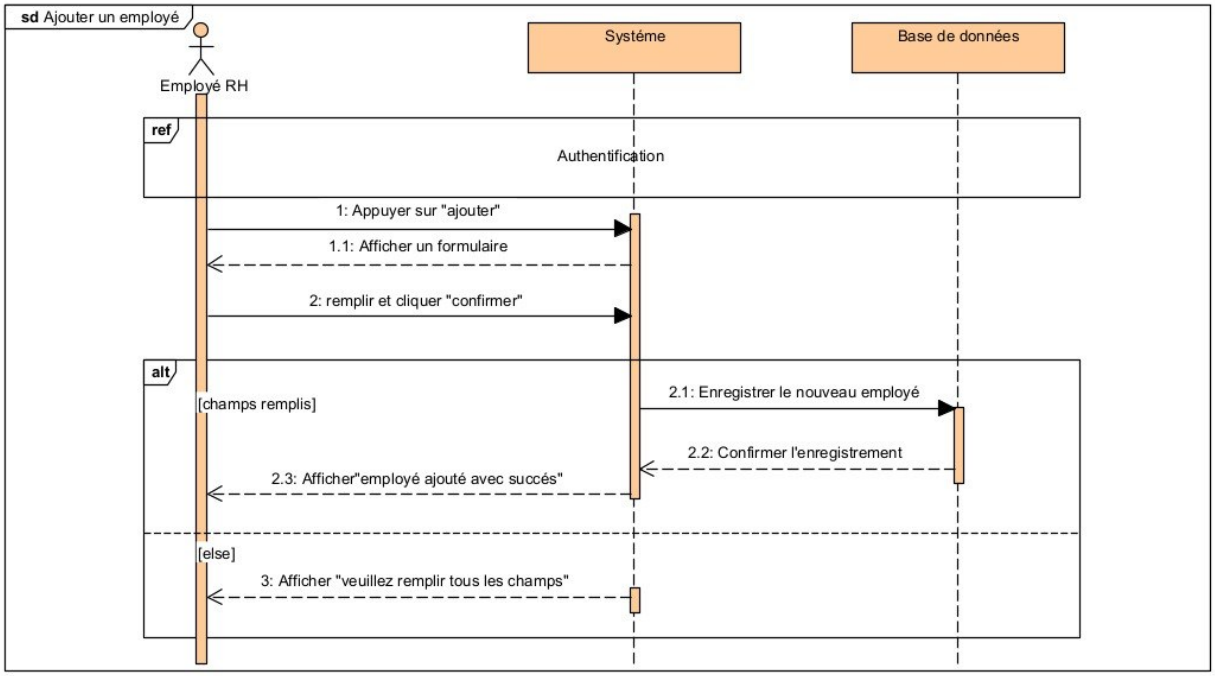


FIGURE 2.10 – Diagramme de séquence du cas d'utilisation « Ajouter un employé » [5]

i3. Table descriptive du cas d'utilisation « Ajouter un employé »

Acteur	Employé RH
Nom du cas d'utilisation	Ajouter un employé
Objectif	Permettre à l'employé RH d'ajouter un nouvel employé dans le système
Pré-condition	L'employé RH est authentifié .
Scénario nominal	1. appuyer sur "Ajouter". 1.1. Le système affiche un formulaire d'ajout. 2. remplir et cliquer sur "Confirmer".
Scénario alternatif	**[champs remplis] :** 2.1. Le système enregistre le nouveau employé dans la base de données. 2.2. La base de données confirme l'enregistrement 2.3. Le système affiche le message "Employé ajouté avec succès". **[else — Champs non remplis] :** 3. le système affiche le message "Veuillez remplir tous les champs".
Post-condition	Le nouveau employé est enregistré dans la base de données et apparaît dans la liste des employés.

TABLE 2.10: Table descriptive du cas d'utilisation « Ajouter un employé »

i4. Diagramme de séquence du cas d'utilisation « Modifier un employé »

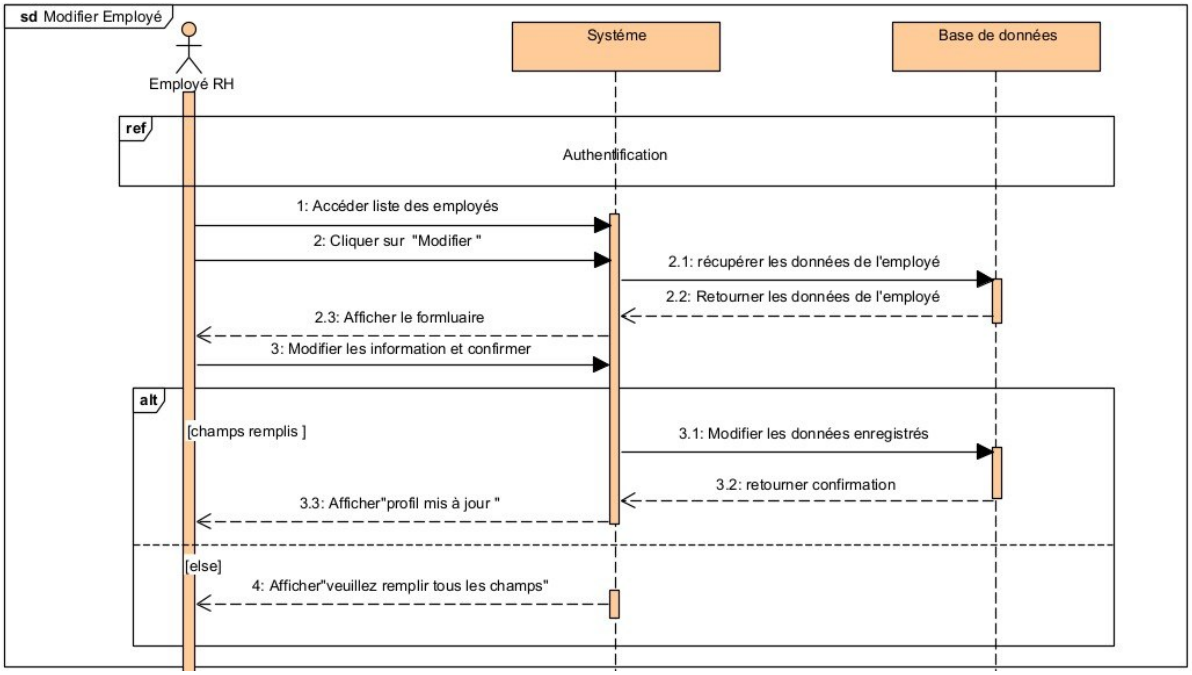


FIGURE 2.11 – Diagramme de séquence du cas d'utilisation « Modifier un employé » [5]

i5. Table descriptive du cas d'utilisation « Modifier un employé »

Acteur	Employé RH
Nom du cas d'utilisation	Modifier employé
Objectif	Permettre à l'employé RH de modifier les informations d'un employé existant en accédant à son profil depuis la liste des employés et en confirmant les modifications.
Pré-condition	L'employé RH est authentifié dans le système
Scénario nominal	1. accéder à la liste des employés. 2. cliquer sur "Modifier". 2.1. Le système récupère les données de l'employé depuis la base de données. 2.2. La base de données retourne les données de l'employé. 2.3. Le système affiche le formulaire pré-rempli avec les données de l'employé. 3. modifier les informations et clique sur "Confirmer".
Scénario alternatif	**[champs remplis] :** 3.1. Le système envoie les données modifiées à la base de données. 3.2. La base de données retourne la confirmation. 3.3. Le système affiche le message "Profil mis à jour" **[else– Champs non remplis] :** 4. le système affiche le message "Veuillez remplir tous les champs".
Post-condition	**[Succès]** : Les informations de l'employé sont mises à jour dans la base de données et le message "Profil mis à jour" est affiché. **[Echec]** : Aucune modification n'est effectuée si les champs sont incomplets.

TABLE 2.11: Table descriptive du cas d'utilisation « Modifier un employé »

i6. Diagramme de séquence du cas d'utilisation « Supprimer un employé »

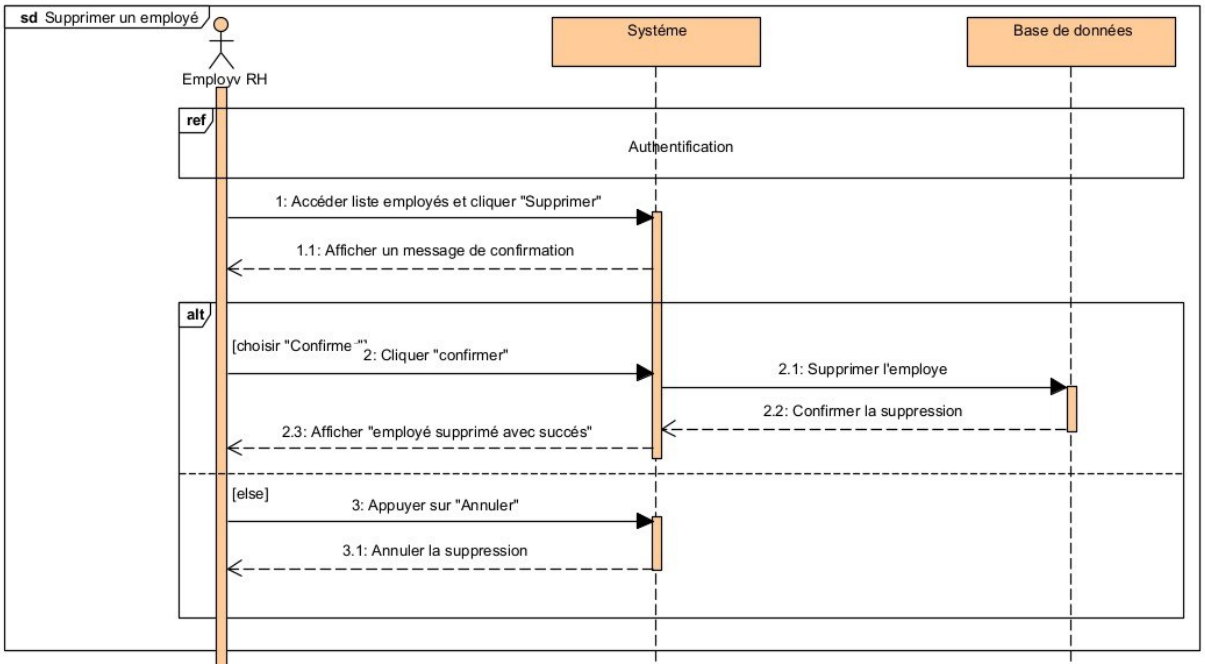


FIGURE 2.12 – Diagramme de séquence du cas d'utilisation « Supprimer un employé »[5]

i7. Table descriptive du cas d'utilisation « Supprimer un employé »

Acteur	Employé RH
Nom du cas d'utilisation	Supprimer employé
Objectif	Permettre à l'employé RH de supprimer un employé existant.
Pré-condition	L'employé RH est authentifié .
Scénario nominal	1. Accéder la liste des employés et cliquer sur "Supprimer". 1.1. Le système affiche un message de confirmation. **[choisir "Confirmer"]** : 2. L'employé RH clique sur "Confirmer". 2.1. Le système envoie une requête de suppression à la base de données. 2.2. La base de données confirme la suppression. 2.3. Le système affiche "Employé supprimé avec succès".
Scénario alternatif	**[choisir "Confirmer"]** : 2. cliquer sur "Confirmer". 2.1. Le système envoie supprimer l'employées à la base de données. 2.2. La base de données confirme la suppression. 2.3. Le système affiche "Employé supprimé avec succès". **[else "Annulation"]** : 3. appuyer sur "Annuler". 3.1. Le système demande annuler la suppression.
Post-condition	L'employé sélectionné est supprimé de la base de données. Aucune modification n'est effectuée en cas d'annulation.

TABLE 2.12: Table descriptive du cas d'utilisation « Supprimer un employé »

2.4.2 Diagramme de classes

2.4.3 Règles de gestion

Les règles de gestion décrivent les contraintes fonctionnelles et les comportements attendus du système intelligent de pointage par reconnaissance faciale.

- Tout utilisateur du système doit disposer d'un compte unique identifié par une adresse email et un mot de passe.
- Un compte peut être désactivé par le personnel RH. Un compte inactif ne peut plus accéder au système.
- En cas de tentatives de connexion échouées répétées, le système doit bloquer temporairement l'accès au compte concerné.
- Chaque utilisateur est automatiquement redirigé vers l'interface correspondant à son rôle après authentification.
- Seul le personnel RH est habilité à ajouter, modifier et supprimer un employé dans le système.
- Chaque employé est identifié de manière unique par un matricule attribué par le service RH.
- L'ajout d'un employé implique obligatoirement la saisie du nom, prénom, département, poste et date d'embauche.
- Toute suppression d'un employé doit faire l'objet d'une confirmation explicite avant exécution. L'opération est irréversible.
- Un employé supprimé ne doit plus apparaître dans la liste active, mais ses données historiques de pointage doivent être conservées.
- Un pointage en retard est automatiquement marqué selon l'heure de référence définie par l'entreprise.
- Un justificatif d'absence doit obligatoirement comporter une date de début, une date de fin et un motif.
- Seul le personnel RH est habilité à accepter ou refuser un justificatif d'absence.
- Un employé ne peut pas soumettre deux justificatifs couvrant la même période.
- Les absences non justifiées dans un délai défini sont automatiquement considérées comme injustifiées.
- L'enregistrement d'un visiteur nécessite obligatoirement le nom, prénom, le motif de visite et le nom de la personne visitée.
- Un badge visiteur à numéro unique est généré automatiquement lors de l'enregistrement et reste valide durant toute la visite
- Un visiteur ne peut pas accéder aux ressources internes du système d'information de l'entreprise.
- L'heure d'arrivée et l'heure de départ du visiteur doivent être enregistrées.

Les points suivants décrivent le fonctionnement du système ainsi que les différentes contraintes liées au pointage, aux absences et à la gestion des utilisateurs :

- Le pointage est effectué exclusivement via la webcam de la borne de pointage opérée par l'agent.
- Chaque employé ne peut avoir qu'un seul pointage d'arrivée par jour. Un second passage dans la même journée est enregistré comme heure de sortie.
- En cas de non-reconnaissance faciale, le système affiche une alerte et propose l'enregistrement de la personne en tant que visiteur.
- Un employé peut consulter ses propres informations personnelles ainsi que son historique de pointage sans possibilité de modification.

— Un employé peut soumettre un justificatif d'absence depuis son espace personnel.
— Tout justificatif soumis reste en attente jusqu'à validation ou refus par le personnel RH.
— L'employé peut consulter le statut de son justificatif depuis son espace personnel : en attente, accepté ou refusé.
— Le personnel RH peut consulter l'historique des pointages de tous les employés sur une période donnée.
— Le personnel RH peut exporter les données de pointage pour des besoins de reporting.
— Les données des visiteurs sont conservées dans le système à des fins de traçabilité et de sécurité.

2.4.4 Dictionnaire de données

Le tableau suivant présente l'ensemble des entités, attributs, types et descriptions du système de pointage.

N°	**Classe**	**Code**	**Description**	**Type**
01	PERSONNE	id	Identifiant unique de la personne	INT
		nom	Nom de famille	String
		prenom	Prénom de la personne	String
		email	Adresse email unique	String
		dateNaissance	Date de naissance	Date
		numeroCarte	Numéro de carte d'identité	String
		lieuNaissance	Lieu de naissance	String
02	EMPLOYE	role	Rôle de l'employé	String
		estActif	Statut actif ou désactivé	boolean
		post	Poste occupé par l'employé	String
03	COMPTE	id	Identifiant unique du compte	INT
		email	Email de connexion	String
		passwordHash	Mot de passe haché	String
		token	Token de session	String
04	POINTAGE	id	Identifiant unique du pointage	INT
		date	Date du pointage	Date
		heureEntree	Heure d'arrivée	Time
		heureSortie	Heure de départ	Time
		statut	Statut : présent, retard, absent	String
05	JUSTIFICATION	id	Identifiant unique du justificatif	INT
		dateAbsence	Date de l'absence	Date
		motif	Motif de l'absence	String
		statut	Statut du justificatif	String
		fichier	Chemin vers le fichier joint	String
06	Date	valeur	Valeur de la date	Date
07	PHOTO	id	Identifiant unique de la photo	INT
		path	Chemin vers l'image	String
		embedding	Vecteur de reconnaissance faciale	float[]
08	VISITEUR	badgeNumero	Numéro de badge du visiteur	String
09	VISITE	id	Identifiant unique de la visite	INT
		dateVisite	Date de la visite	Date
		motifVisite	Motif de la visite	String
		departementVisite	Département visité	String
		heureArrivee	Heure d'arrivée du visiteur	DateTime
		heureDepart	Heure de départ du visiteur	DateTime
10	DEPARTEMENT	id	Identifiant unique du département	INT
		nom	Nom du département	String

TABLE 2.13: Dictionnaire de données

2.4.5 Diagramme de classes général

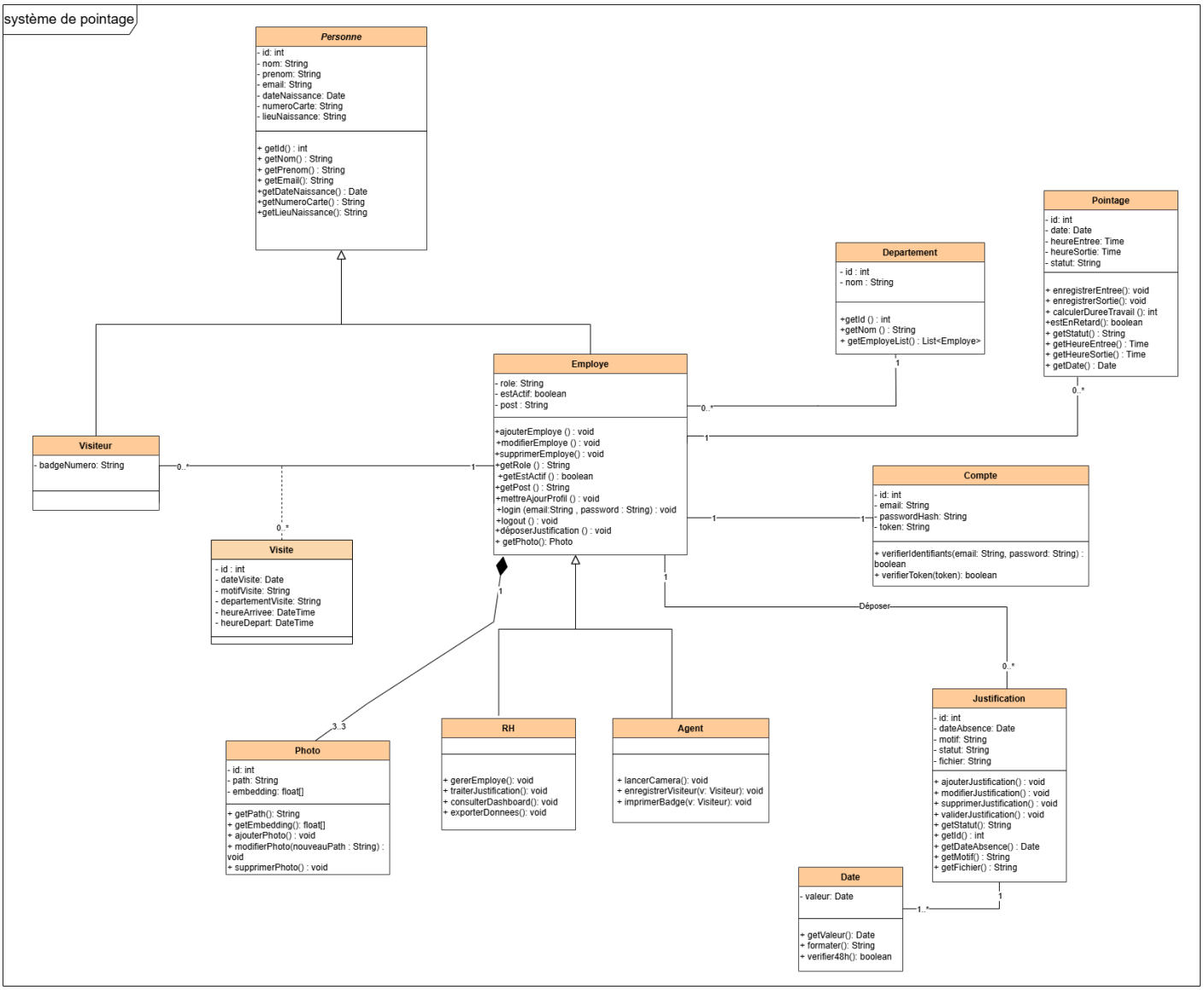


FIGURE 2.13 – Diagramme de classes général [6]

2.4.6 Règles de transformation

Les règles suivantes décrivent le passage du diagramme de classes UML vers le Modèle Logique de Données (MLD) relationnel.

— Chaque classe concrète du diagramme (PERSONNE, EMPLOYE, COMPTE, POINTAGE, JUSTIFICATION, PHOTO, VISITEUR, VISITE, DEPARTEMENT) est transformée en une table relationnelle. Les attributs de la classe deviennent les colonnes de la table.

— La classe mère PERSONNE est conservée comme table. Les sous-classes EMPLOYE et VISITEUR ont chacune leur propre table avec une clé étrangère vers PERSONNE. Les attributs spécifiques restent dans chaque table fille.

— Les sous-classes RH et AGENT n'ont pas d'attributs propres dans le diagramme. Elles sont absorbées dans la table EMPLOYE via l'attribut role. Aucune table séparée n'est créée pour RH et AGENT.

— Un employé possède un seul compte et un compte appartient à un seul employé. La clé étrangère compte_id est placée dans la table EMPLOYE avec une contrainte UNIQUE.

— Un employé peut avoir plusieurs pointages, mais chaque pointage appartient à un seul employé. La clé étrangère employe_id est placée dans la table POINTAGE (côté N).

— Un employé peut soumettre plusieurs justificatifs. Chaque justificatif est lié à un seul employé. La clé étrangère employe_id est placée dans la table JUSTIFICATION.

— Un employé peut avoir plusieurs photos pour la reconnaissance faciale (multiplicité 1..*). La clé étrangère employe_id est placée dans la table PHOTO.

— Un département contient plusieurs employés, mais chaque employé appartient à un seul département. La clé étrangère departement_id est placée dans la table EMPLOYE.

— Un visiteur peut effectuer plusieurs visites (multiplicité 0..*). Chaque visite est liée à un seul visiteur. La clé étrangère visiteur_id est placée dans la table VISITE.

— La relation entre VISITE et EMPLOYE (multiplicité 3..3 côté EMPLOYE) indique que chaque visite est liée à exactement un employé visité. La clé étrangère employe_id est placée dans la table VISITE.

— La classe Date est une classe utilitaire sans entité propre. Son attribut valeur est absorbé sous forme de colonnes de type DATE ou DATETIME dans les tables concernées (POINTAGE, JUSTIFICATION, VISITE). La méthode verifier48h() est gérée par la logique applicative.

— La relation « Dépose » entre JUSTIFICATION et la classe Date ne génère pas de table d'association. L'attribut valeur de la classe Date est absorbé sous forme de colonne dateAbsence de type DATE dans la table JUSTIFICATION.

2.4.7 Modèle Logique de Données (MLD) :

Convention : identifiant primaire souligné, clé étrangère précédée de # et en italique.

— **PERSONNE**(id, nom, prenom, email, dateNaissance, numeroCarte, lieuNaissance)
— **EMPLOYE**(id, #*personne_id*, #*compte_id*, #*departement_id*, role, estActif, post)
— **VISITE**(id, #*visiteur_id*, #*employe_id*, dateVisite, motifVisite, departementVisite, heureArrivee, heureDepart)
— **VISITEUR**(id, #*personne_id*, badgeNumero)
— **DEPARTEMENT**(id, nom)
— **COMPTE**(id, email, passwordHash, token)
— **POINTAGE**(id, #*employe_id*, date, heureEntree, heureSortie, statut)
— **JUSTIFICATION**(id, #*employe_id*, dateAbsence, motif, statut, fichier)
— **PHOTO**(id, #*employe_id*, path, embedding)

Conclusion

En conclusion, ce chapitre nous a permis de définir les besoins fonctionnels du système de pointage par reconnaissance faciale et de présenter la méthode de conception RUP adoptée pour le développement. À travers les différents diagrammes UML, notamment les diagrammes de cas d'utilisation, de séquence et de classes, nous avons pu modéliser le fonctionnement global du système ainsi que les interactions entre ses différents composants. Cette étape de conception constitue une base essentielle pour assurer un développement clair, structuré et efficace du système.

Chapitre 3

Architecture et implémentation du système

Introduction

Dans cette partie, nous présenterons les choix techniques et architecturaux adoptés afin d'assurer la performance, la sécurité et la maintenabilité du système. Nous décrirons également l'architecture générale de la solution ainsi que l'organisation des différentes couches de l'application.

Ensuite, nous exposerons les outils matériels (hardware) et logiciels (software) utilisés durant le développement. Nous présenterons également l'architecture adoptée pour la structuration de l'application, basée sur le modèle **MVC (Modèle–Vue–Contrôleur)**.

Enfin, quelques captures d'écran des principales interfaces seront présentées afin d'illustrer le fonctionnement de l'application et les résultats obtenus.

3.1 Architecture générale de l'application

3.1.1 Modèle MVC

Dans notre application, nous avons adopté l'architecture MVC (Model–View–Controller). Il s'agit d'un modèle de conception qui permet de structurer une application en trois parties distinctes afin de séparer les données, l'interface utilisateur et la logique de contrôle.

Cette approche permet de mieux organiser le code, de faciliter la maintenance et de rendre l'application plus évolutive.

Le **Modèle (Model)** gère les données ainsi que la logique métier. Il s'occupe aussi des interactions avec la base de données et du traitement des informations.

La **Vue (View)** correspond à l'interface affichée à l'utilisateur. Elle permet de présenter les données et de gérer l'interaction avec l'utilisateur.

Le **Contrôleur (Controller)** joue le rôle d'intermédiaire entre le modèle et la vue. Il reçoit les actions de l'utilisateur, traite les données via le modèle, puis renvoie le résultat à l'interface.

Dans notre projet, cette organisation facilite les échanges entre le frontend, le backend et la base de données, tout en gardant une structure claire et facile à faire évoluer.

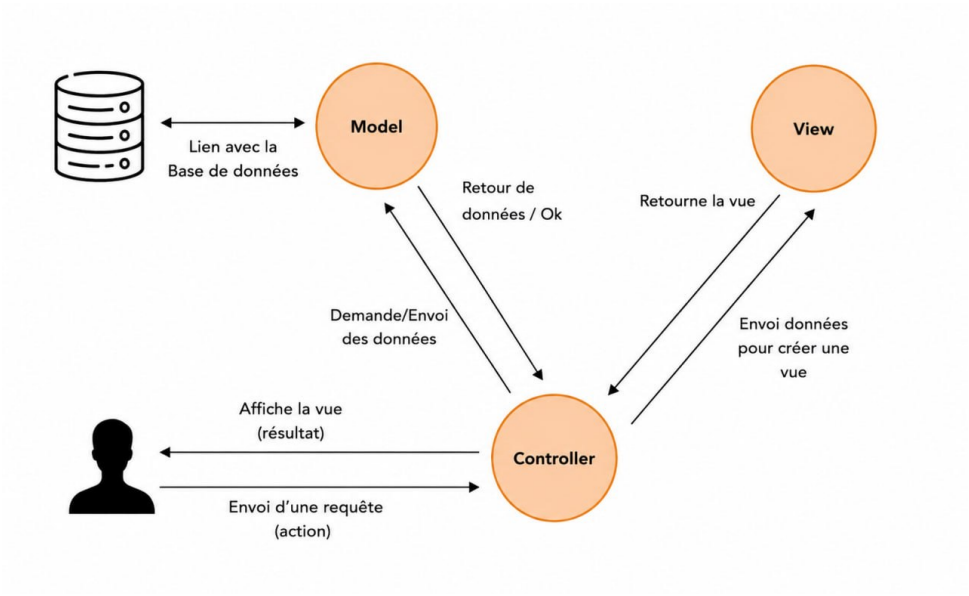


FIGURE 3.1 – Schéma de l'architecture Modèle-Vue-Contrôleur (MVC)

3.1.2 Diagramme de déploiement

Le diagramme de déploiement ci-dessous illustre l'architecture physique du système de pointage par reconnaissance faciale. Il met en évidence trois nœuds principaux : le poste utilisateur exécutant l'interface React.js via un navigateur web, le serveur local hébergeant le backend Flask et le module de reconnaissance faciale InsightFace, ainsi que le serveur de base de données PostgreSQL.

Les échanges entre le frontend et le backend s'effectuent via des requêtes HTTP/REST au format JSON, tandis que la communication avec la base de données est assurée via SQL sur TCP/IP. Le module InsightFace est intégré au backend afin d'assurer le traitement de la reconnaissance faciale en temps réel

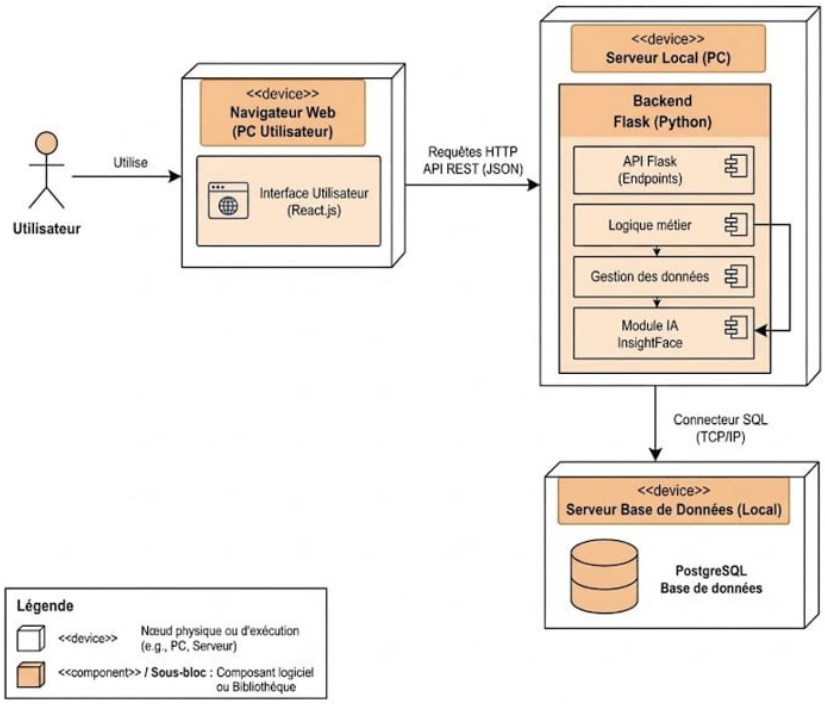


FIGURE 3.2 – Diagramme de déploiement

3.2 Environnement de travail et outils utilisés

3.2.1 Matériel informatique (hardware)

Le tableau suivant présente les caractéristiques des équipements physiques que nous avons utilisés pour réaliser ce projet

Caractéristi machine	Nom de machine	Processeur	Mémoire RAM	Type de Système	Système d'exploitation	utilisé comme
Machine 01	DELL	Intel Core i5-8265U @ 1.60GHz (1.80 GHz)	16,0 Go	64-bit	Windows 11	SERVEUR
Machine 02	HP	Intel Core i7-8550U @ 1.80GHz (1.99 GHz)	8,00 Go	64-bit	Windows 11	Client

TABLE 3.1 – Caractéristiques des machines utilisées

3.2.2 Environnement logiciel (software)

Cette partie présente les différents logiciels et technologies utilisés dans la réalisation du projet.

Visual Studio Code : Cet éditeur de code source extensible développé par Microsoft a servi d'environnement de développement principal (IDE) pour centraliser l'écriture et la gestion de tout notre code frontend et backend. Nous l'avons choisi pour sa légèreté, son interface intuitive et son riche écosystème d'extensions qui optimise le flux de travail.



React.js : Cette bibliothèque JavaScript open-source développée par Meta est basée sur une architecture de composants réutilisables et a été utilisée pour concevoir l'intégralité de l'interface utilisateur (frontend). Nous avons choisi React pour sa grande réactivité grâce au DOM virtuel, ce qui garantit une application fluide et facilement évolutive [11].



Tailwind CSS : Ce framework CSS moderne basé sur des classes utilitaires a été intégré à React.js pour réaliser la mise en page, le design global et le responsive de notre application. Nous l'avons sélectionné car il permet un développement visuel extrêmement rapide sans avoir à rédiger de longues feuilles de style personnalisées [14].



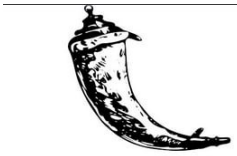
PostgreSQL : Ce système de gestion de base de données relationnelle open-source et robuste a été déployé pour assurer le stockage persistant et sécurisé de toutes les données (utilisateurs, employés, historiques). Nous l'avons choisi pour sa fiabilité exemplaire, sa conformité ACID et sa parfaite gestion des requêtes complexes et des volumes de données importants [13].



Python : Python est un langage de programmation de haut niveau, interprété et reconnu pour sa syntaxe claire, lisible et épurée. Nous l'avons choisi comme langage principal de notre backend car il s'impose comme la référence absolue en intelligence artificielle, permettant d'exploiter nativement les bibliothèques avancées de vision par ordinateur indispensables à notre système [15].



Python Flask : Ce micro-framework web léger et minimaliste en Python a été utilisé pour propulser le backend de l'application, gérant la logique métier, la base de données et l'API REST. Le choix de Flask repose sur sa flexibilité (architecture en usine) et sa capacité à s'intégrer nativement et efficacement avec des scripts d'intelligence artificielle [12].



Postman : Cette plateforme spécialisée dans le développement et le test d'API a été utilisée intensivement pour simuler des requêtes HTTP et valider le bon fonctionnement de nos routes Flask avant leur connexion au frontend. Il a été choisi car il accélère grandement la phase de débogage et permet de tester le backend de manière totalement isolée [16].



InsightFace : Après une étude comparative avec FaceNet et DeepFace (voir tableau ci-dessous), le framework de vision par ordinateur InsightFace a été retenu et intégré à Flask pour la détection et l'extraction des caractéristiques faciales. Ce choix est motivé par sa précision de pointe et sa vitesse de traitement optimale, indispensables pour une reconnaissance efficace sur un flux vidéo en temps réel [10].



Critère	**FaceNet**	**DeepFace** [9]	**InsightFace**
Développé par	Google	Facebook / MIT	Jia Deng et al.
Année	2015	2021	2021
Architecture	Inception ResNet	VGG-16 / ResNet / ArcFace	ArcFace / RetinaFace
Détection de visage	Externe (MTCNN)	Intégrée	Intégrée (RetinaFace)
Précision (LFW)	∼99.63%	∼99.38%	∼99.83%
Dimension embeddings	128D	128 – 4096D	512D
Analyse d'attributs	Non	Partielle	Complète
Performance CPU	Moyenne	Lente	Rapide
Intégration Python	Moyenne	Facile	Très facile
Multi-visages	Non natif	Oui	Oui (natif)
Licence	Apache 2.0	MIT	MIT

TABLE 3.2 – Comparaison des bibliothèques de reconnaissance faciale

3.3 Les interfaces de l'application web

Dans cette partie, nous présentons quelques captures d'écran de notre application web afin d'illustrer son interface et ses principales fonctionnalités..

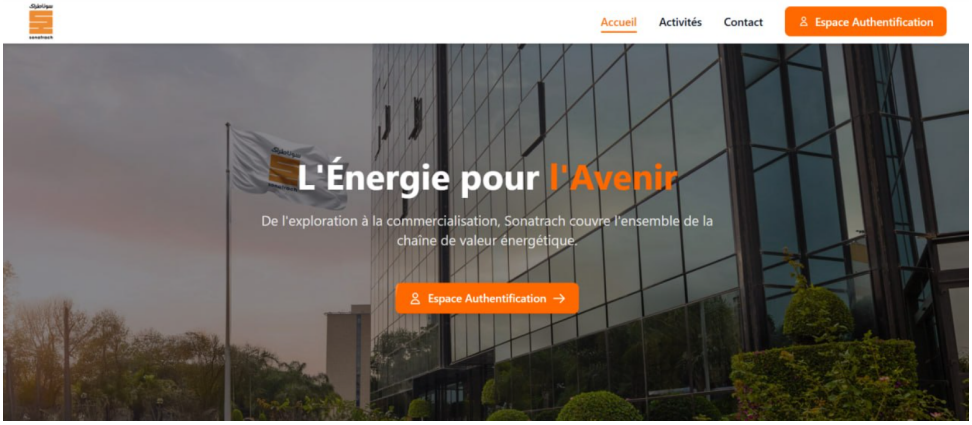


FIGURE 3.3 – Interface d'Accueil

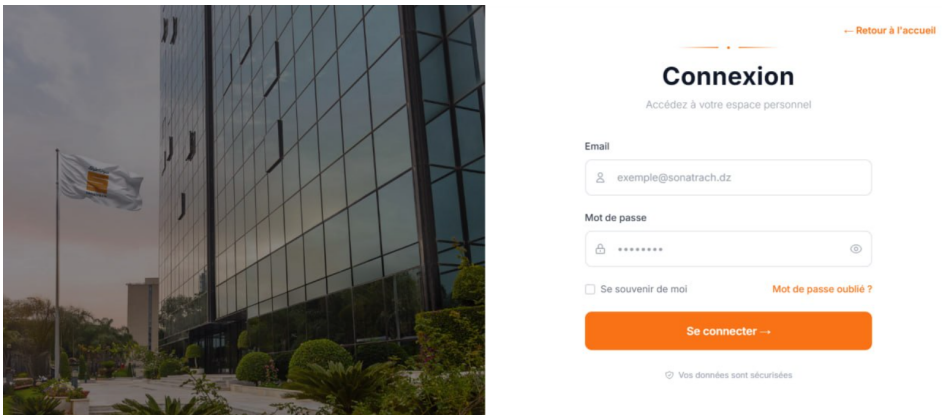


FIGURE 3.4 – L'interface dédiée à l'authentification

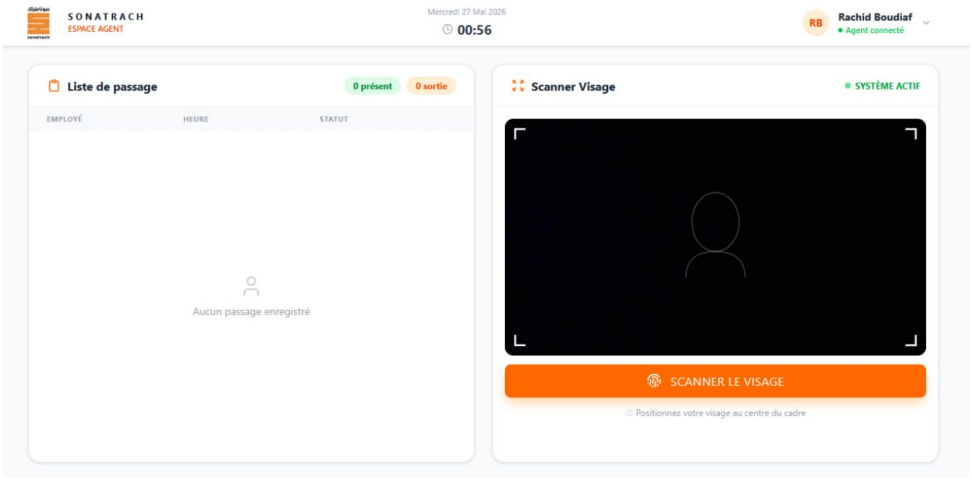


FIGURE 3.5 – Interface de l'espace agent

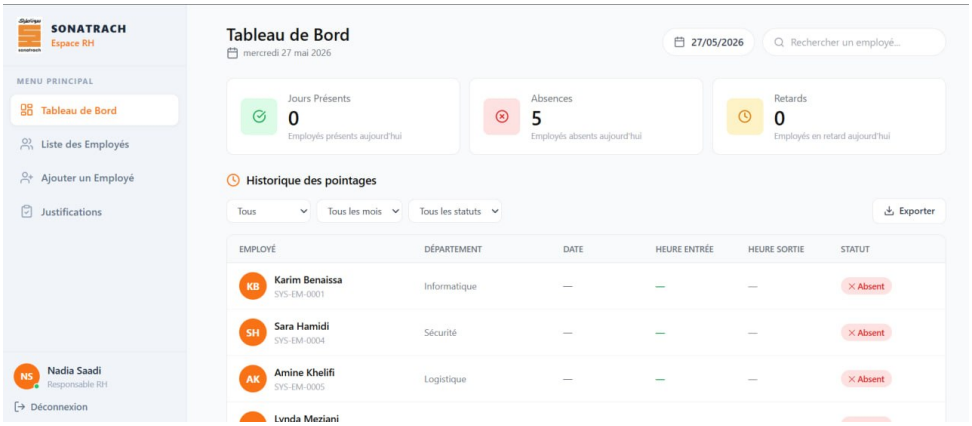


FIGURE 3.6 – Interface de l'espace RH

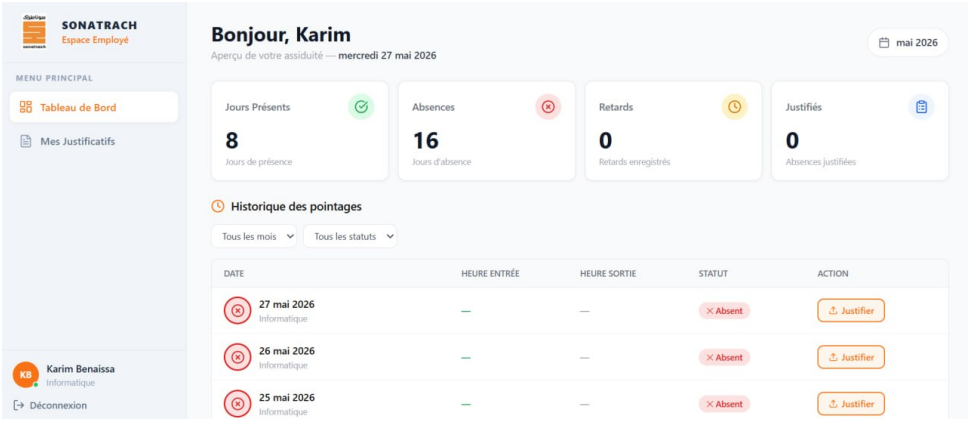


FIGURE 3.7 – Interface de l'espace Employe

Conclusion

En conclusion, ce chapitre a été consacré à la phase de réalisation de l'application à travers la présentation des choix techniques et des architectures adoptées. Nous avons décrit l'environnement matériel et logiciel utilisé durant le développement ainsi que l'architecture MVC choisie pour assurer une meilleure organisation et séparation des composants de l'application. Ce chapitre a également permis de présenter le modèle d'intelligence artificielle intégré au système ainsi que son rôle dans l'amélioration des fonctionnalités proposées. Enfin, les différentes interfaces réalisées ont illustré concrètement le fonctionnement de l'application et les résultats obtenus

Conclusion générale

Ce mémoire a présenté la conception et l'implémentation d'un système intelligent de pointage par reconnaissance faciale, réalisé dans le cadre d'un stage pratique au sein de la Division Exploration de SONATRACH. Ce projet répond à un besoin concret de l'entreprise en matière de suivi des présences, en proposant une solution moderne, fiable et sécurisée, capable d'automatiser l'enregistrement des pointages et de centraliser la gestion du personnel.

Tout au long de ce travail, nous avons suivi une démarche structurée basée sur la méthode RUP, en passant par l'analyse des besoins, la modélisation UML, la conception de la base de données et l'implémentation d'une application web complète. Le système intègre un module de reconnaissance faciale en temps réel, retenu pour sa précision et ses performances, au service d'une gestion du personnel plus efficace.

L'application développée permet aujourd'hui d'assurer les fonctionnalités essentielles du système : l'authentification sécurisée avec redirection selon le rôle, le pointage automatique par reconnaissance faciale, la gestion des visiteurs avec génération de badge, la soumission et le traitement des justificatifs d'absence, ainsi que la gestion complète des employés par le personnel RH. Ces résultats se traduisent par un gain de temps significatif, une réduction des erreurs de saisie et un renforcement de la sécurité grâce à l'identification biométrique des employés.

Bien que les résultats obtenus soient satisfaisants, certaines limites subsistent dans le système actuel.

Le module de reconnaissance faciale a été testé dans un environnement contrôlé, et son déploiement en conditions réelles avec les employés de SONATRACH constitue une étape importante pour une utilisation effective.

Sur le plan fonctionnel, l'espace employé peut être amélioré par l'ajout d'un module de demande de congé, permettant une gestion plus complète des ressources humaines.

Concernant le déploiement, l'application est conçue pour fonctionner uniquement sur le réseau interne de l'entreprise, garantissant ainsi la sécurité et la confidentialité des données. Une extension à l'ensemble des sites de la Division Exploration permettrait d'exploiter pleinement son potentiel.

Ce projet constitue une base solide et évolutive pour une gestion moderne et sécurisée des ressources humaines, et met en évidence l'apport concret de l'intelligence artificielle dans l'amélioration des processus administratifs au sein de SONATRACH.

Bibliographie

[1] Site officiel SONATRACH. [En ligne]. Disponible sur : https ://www.sonatrach.com

[2] Site officiel SONATRACH, Organigramme général. [En ligne]. Disponible sur : https ://www.sonatrach.com

[3] P. Kruchten, *The Rational Unified Process : An Introduction*, 3ème édition, Addison-Wesley, 2003.

[4] G. Booch, J. Rumbaugh & I. Jacobson, *Le guide de l'utilisateur UML*, 2ème édition, Eyrolles, 2005.

[5] Visual Paradigm, *Documentation officielle Visual Paradigm — Outil de modélisation UML*. [En ligne]. Disponible sur : https ://www.visual-paradigm.com

[6] draw.io (diagrams.net), *Outil de création de diagrammes en ligne*. [En ligne]. Disponible sur : https ://www.drawio.com

[7] J. Deng, J. Guo, N. Xue & S. Zafeiriou, "ArcFace : Additive Angular Margin Loss for Deep Face Recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.

[8] F. Schroff, D. Kalenichenko & J. Philbin, "FaceNet : A Unified Embedding for Face Recognition and Clustering", in *Proceedings of CVPR*, 2015, pp. 815–823.

[9] S. Serengil & A. Ozpinar, "LightFace : A Hybrid Deep Face Recognition Framework", in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2020.

[10] InsightFace, *Documentation officielle InsightFace*. [En ligne]. Disponible sur : https ://www.insightface.ai

[11] React.js, *Documentation officielle React*. [En ligne]. Disponible sur : https ://www.react.dev

[12] Flask, *Documentation officielle Flask*. [En ligne]. Disponible sur : https ://www.flask.palletsprojects.com

[13] PostgreSQL, *Documentation officielle PostgreSQL*. [En ligne]. Disponible sur : https ://www.postgresql.org/docs

[14] Tailwind CSS, *Documentation officielle Tailwind CSS*. [En ligne]. Disponible sur : https ://www.tailwindcss.com/docs

[15] Python Software Foundation, *Documentation officielle Python 3*. [En ligne]. Disponible sur : https ://www.docs.python.org/3

[16] Postman, *Documentation officielle Postman*. [En ligne]. Disponible sur : https ://www.learning.postman.com