Sommaire

[**Chapitre 1** 2](#_Toc454307690)

[Analyse et Conception 3](#_Toc454307691)

[I) Introduction 3](#_Toc454307692)

[II) Diagramme UML 3](#_Toc454307694)

[Diagramme de cas d’utilisation: 5](#_Toc454307696)

[Diagramme de Classe: 5](#_Toc454307696)

IV)[Conclusion 8](#_Toc454307697)

[**Chapitre 2** 9](#_Toc454307698)

[Techniques et outils utilisés 9](#_Toc454307699)

[1) Langages Utilisés : 10](#_Toc454307700)

[MYSQL : 10](#_Toc454307701)

[Python 10](#_Toc454307702)

[3) Espaces de travail 10](#_Toc454307705)

[PowerAMC 10](#_Toc454307706)

[4) Conclusion : 11](#_Toc454307709)

[**Chapitre 3** 12](#_Toc454307710)

[Réalisation 12](#_Toc454307711)

[I) Application Desktop. 13](#_Toc454307712)

[1)Interface de l’application: 13](#_Toc454307713)

[**Conclusion Générale** 16](#_Toc454307732)

**Introduction Générale.**

# Ce projet vise à développer une application de gestion des employés qui inclut un système de reconnaissance faciale pour le suivi des heures de travail et des fonctionnalités pour la gestion des données des employés. L'application permet aux gestionnaires de suivre les performances, de générer des rapports, et de gérer les tâches et les horaires de travail. L'utilisation du langage Python et des algorithmes de reconnaissance faciale sont au cœur de cette solution. Le projet met également l'accent sur la sécurité des données et la confidentialité des employés.

# Chapitre 1

# Analyse et Conception

* Diagrammes Utilisés

Ce chapitre définit l’architecture de base de la conception de notre système. Il inclut aussi les diagrammes utilisés tels que le diagramme de classe, diagramme de cas d’utilisation.

# I) Introduction

Comme n’importe quel type de projet, un projet informatique nécessite une phase d’analyse, suivi d’une étape de conception.

Langage (UML), est un [langage](http://fr.wikipedia.org/wiki/Langage) de modélisation graphique à base de [pictogrammes](http://fr.wikipedia.org/wiki/Pictogramme) conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en [développement logiciel](http://fr.wikipedia.org/wiki/D%C3%A9veloppement_logiciel) et en [conception orientée objet](http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet).

### 

# II) Diagramme UML

# Diagramme de Cas d’utilisation :

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils ont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

Une image contenant diagramme, texte, capture d’écran, ligne

Description générée automatiquement

Une image contenant diagramme, cercle, capture d’écran, ligne

Description générée automatiquement

Figure 1: Diagramme de Cas d’utilisation :

**Acteurs Employé :** Représente les utilisateurs de l'application, qui interagissent avec le système pour réaliser diverses actions liées à leurs informations personnelles et horaires de travail.

**Acteur Gestionnaire :** Utilisateur principal de l'application, responsable de la gestion des informations et des horaires des employés.

# Le gestionnaire est lié au cas d'utilisation « Gérer les informations des employés », qui se décompose en deux cas d'utilisation par généralisation : « Afficher les informations personnelles » et « Gérer les horaires de travail ». De plus, le cas d'utilisation « Gérer les employés » est étendu pour inclure les fonctionnalités « Ajouter un employé » et « Supprimer un employé » par généralisation.

L'acteur **Employé** est associé au cas d'utilisation **Reconnaissance faciale**, un processus clé permettant de suivre la présence des employés de manière automatisée et précise. Ce cas d'utilisation se subdivise en deux cas distincts :

1. **Pointer arrivée** : Ce cas d'utilisation permet à l'employé d'enregistrer son heure d'arrivée au travail via la reconnaissance faciale. Lorsqu'un employé arrive sur le lieu de travail, le système reconnaît son visage et enregistre automatiquement son heure d'entrée, assurant ainsi un suivi précis et sans intervention manuelle.
2. **Pointer départ** : Ce cas d'utilisation permet à l'employé d'enregistrer son heure de départ via la reconnaissance faciale. En fin de journée ou lors de la fin de son service, l'employé passe par le système de reconnaissance faciale pour marquer son départ. Le système enregistre alors l'heure de sortie, garantissant un suivi exact des heures travaillées.

# Diagramme de Classe :

Le diagramme de classes est un schéma utilisé en [génie logiciel](http://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel) pour présenter les [classes](http://fr.wikipedia.org/wiki/Classe_%28informatique%29) et les [interfaces](http://fr.wikipedia.org/wiki/Interface_%28informatique%29) des systèmes ainsi que les différentes relations entre celles-ci. Ce [diagramme](http://fr.wikipedia.org/wiki/Diagramme) fait partie de la partie [statique](http://fr.wikipedia.org/wiki/Statique) d'[UML](http://fr.wikipedia.org/wiki/Unified_modeling_language) car il fait abstraction des aspects temporels et [dynamiques](http://fr.wikipedia.org/wiki/Dynamique).

Une image contenant texte, nombre, capture d’écran, thermomètre

Description générée automatiquement

Figure 2 : Diagramme de cas d'utilisation

|  |
| --- |
| **Classe Employé :**  **Attributs :**  idEmp : Identifiant unique de l'employé (int)  nom : Nom de l'employé (String)  prenom : Prénom de l'employé (String)  dateDebut : Date de début de l'emploi (Date)  dateFin : Date de fin de l'emploi (Date)  poste : Poste de l'employé (String)  departement : Département de l'employé (String)  photo : Photo de l'employé (Image)  face : Données de reconnaissance faciale de l'employé (Binary)  **Méthodes :**  ajouterEmploye() : Méthode pour ajouter un employé.  supprimerEmploye() : Méthode pour supprimer un employé.  **Classe Horaire :**  **Attributs :**  dateArrivee : Date et heure d'arrivée (DateTime)  dateDepart : Date et heure de départ (DateTime)  **Méthodes :**  pointerArrivee() : Méthode pour enregistrer l'heure d'arrivée.  pointerDepart() : Méthode pour enregistrer l'heure de départ.  calculerTempsDeTravail() : Méthode pour calculer le temps de travail total. |

# Conclusion

Nous avons présenté dans ce chapitre les différents diagrammes élaborés dans la conception du projet et qui nous ont aidés à avoir une vue globale sur tout le projet, ce qui nous facilite l’étape suivante, la réalisation.

# 

# Chapitre 2

# Techniques et outils utilisés

* Les langages utilisés
* Les espaces de travail

# 1) Langages Utilisés :

# 1.1) MYSQL :

MySQL études meilleurs SGBDR utilisé pour développer des applications logicielles basées sur le Web. C’est un système de gestion de base de données la plus populaire Open Source relationnelle SQL. (Structured Query Language) gratuit et très simple d’emploi.

# 1.2) Python

Python est un langage de programmation polyvalent, populaire pour le développement d'applications de bureau grâce à sa simplicité et à sa large gamme de bibliothèques. Il offre une syntaxe claire et concise, une grande communauté de développeurs, et des frameworks tels que PyQt et Tkinter pour créer des interfaces utilisateur graphiques (GUI) interactives. Python est largement utilisé dans le développement d'applications de bureau en raison de sa facilité d'apprentissage et de sa capacité à fonctionner sur plusieurs plateformes, en faisant un choix idéal pour les projets de développement rapide d'applications (RAD).

# 3) Espaces de travail

# 3.1) PowerAMC

Est un logiciel de conception créé par la société SDP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il permet alors de réaliser tous les types de modèles informatiques. Il reste un des seuls qui permet de travailler avec la méthode Merise. Selon Riff News, cela permet d'améliorer la modélisation, les processus, le coût et la production d'applications**.**

### 

# 4) Conclusion :

Ce chapitre décrit les différentes technologies qu’on a utilisées, lors de l’élaboration de notre projet, ainsi que les logiciels qui nous ont permis de réaliser à bien notre site.

# 

# Chapitre 3

# Réalisation

# I) Application Desktop.

# 1) Interface de L’application :

# Une image contenant texte, capture d’écran, affichage, logiciel Description générée automatiquement

# L'interface de gestion des employés de l'application « FaceSmart » est conçue pour offrir une expérience utilisateur simple et intuitive pour les gestionnaires. Elle se divise en plusieurs sections fonctionnelles permettant d'ajouter, supprimer et gérer les informations des employés, ainsi que d'utiliser la reconnaissance faciale pour enregistrer les arrivées et départs des employés.

## Espace ajout employé :

**Le formulaire d'ajout d'employé** : Ce formulaire est visible lorsque l'utilisateur clique sur le bouton "Ajouter un employé ». Il permet de saisir les informations relatives à un nouvel employé, telles que son nom, son prénom, son département, sa date de début, sa date de fin et son poste ainsi que la caméra s’ouvre et elle prend une capture de visage et il stock l’image en deux manière (blop et sous en forme de matrice.

## Espace liste employé :

**La liste des employés** : Cette liste est visible lorsque l'utilisateur clique sur le bouton "Liste des employés". Elle affiche tous les employés présents dans la base de données, avec leurs informations respectives.

## Espace de travail :

**Le bouton "Supprimer employé"** : Ce bouton permet de supprimer un employé de la base de données à l’aide d’employé choisi.

**Le bouton "Démarrer la reconnaissance faciale"** : Ce bouton permet d'activer la reconnaissance faciale pour distinguer chaque employé d’un autre et afficher le nom correspond d’employé sur son visage.

**Le bouton "Pointer arrivée"** : Ce bouton permet de pointer l'arrivée d'un employé à l’aide de son visage, il prend en compte le visage présent puis il cherche le dans la base donner si il est similaire puis il point l’arriver .

**Le bouton "Pointer départ"** : Ce bouton permet de pointer le départ d'un employé.

# IV) Conclusion

En résumé, l'interface de gestion des employés est conçue pour être simple et intuitive à utiliser. Elle permet aux utilisateurs d'ajouter et de supprimer des employés, ainsi que de pointer leur arrivée et leur départ.

# 

# Conclusion Générale

Les étapes que nous avons suivies pour atteindre nos objectifs consiste à faire une étude préalable qui nous a permis de collecter les informations, après nous avons modélisé toutes les fonctionnalités nécessaires pour réaliser notre application. En se basant sur UML, nous avons modélisé les diagrammes (classe, et cas d’utilisation). Après on a passé a la phase de codage.

En conclusion, notre projet de reconnaissance faciale se distingue par sa conception robuste, ses fonctionnalités avancées et son engagement envers la sécurité et la confidentialité. Il offre une solution fiable et évolutive pour répondre aux besoins critiques d'identification et de contrôle d'accès dans divers secteurs.

Réalisé par :

Ghayt El Idrissi Dafali

Reda Bouimakliouine