**Projet : Système de Pointage Intelligent par**  **Reconnaissance Faciale**

1. **Description Générale de Projet :**

L'idée Générale de projet est de faciliter l’opération de pointage et gestion des absences donc on va développer un système intelligent de pointage par reconnaissance faciale à l’aide de carte Raspberry pie et une application mobile pour les employés et une application web pour les admins (HR Manager).

1. **Etudes de l’existant :**
2. **Description de l’existant**

Le pointage est une tache obligatoire dans toutes les sociétés mais la plupart de temps elle s’effectuée que ce soit avec papier ou à l’aide d’un logiciel tableur “Excel”

1. **Critiques de l’existant**

À la vue de situation sanitaire et à la propagation du virus “COVID 19”, le touchées aux objets et le contact des personnes est interdit donc on essaye de développer une solution plus sécurisée.la solution existante n’est plus parfaite citons par exemple :

* Le pointage et gestion des absences sont effectuées d’une manière primitive et manuelle
* Le distancement n’est pas garanti et donc la chance de l’infection par le virus est augmentée
* La manque de rapidité et fiabilité (un autre employer peut pointer à votre place au cas de manque de surveillance .

1. **Solution proposée**

La solution proposée consiste à construire un système de pointage intelligent par reconnaissance faciale pour assurer le distancement et une application mobile pour les employés pour faciliter et automatiser la tâche.

1. **Spécification des besoins**

La réussite de tout projet dépend de la qualité de son départ. De ce fait, l’étape de spécification constitue la base de départ de notre travail. Pour assurer aux objectifs de système, il est essentiel de parvenir à une vue claire des différents besoins escomptés de notre projet.

1. **Besoins fonctionnels**

Un besoin fonctionnel est un besoin spécifiant une action qu’un système doit être capable d’effectuer sans aucune contrainte physique : besoins spécifiant un comportement d’entrée/sortie d’un système.

Notre système doit offrir les fonctionnalités suivantes :

* + Permettre aux employées de pointer deux fois par jour par reconnaissance faciale.
  + Réclamer au Admin ou au HR Manager en cas d’erreur ou faute.
  + Permettre aux admins de créer ou gérer les employer, créer un modèle de visage, créer un compte, modifier, supprimer ...
  + Enregistrer les pointages et les absents dans la base de données.

1. **Besoins non fonctionnels**

Un besoin non fonctionnel est un besoin spécifiant les propriétés que le système doit offrir, autres que les détails des fonctionnalités, telles que les contraintes liées à l’environnement.

Pour notre système, nous citons les besoins non fonctionnels suivants :

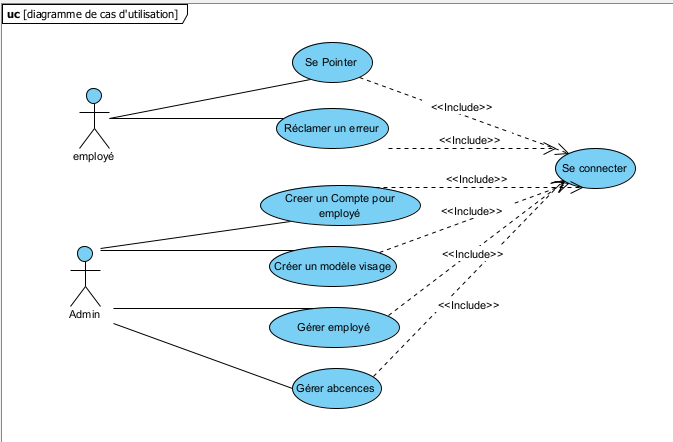
* + Ergonomie et autonomie : cette application doit être facile à utiliser, la tâche est automatisée.
  + Fiabilité : cette application doit répondre à toute intervention de l'utilisateur sans défaillance, avec un temps de réponse rapide.
  + Sécurité : l’application offre à l’utilisateur la possibilité de créer un code de sécurité, l’accès sera ainsi par authentification.

1. **Modélisation des besoins fonctionnels**
2. **Identification des acteurs**

Un acteur représente l’abstraction d’un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système. Notre système a deux acteurs qui sont :

* Employée
* Admin ou HR Manager

1. **Diagramme de cas d’utilisation**



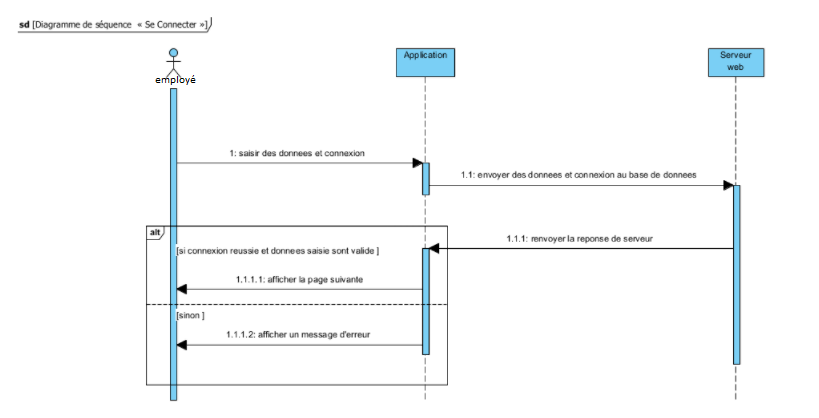
# **Conception :**

1. **Diagrammes de séquences :**

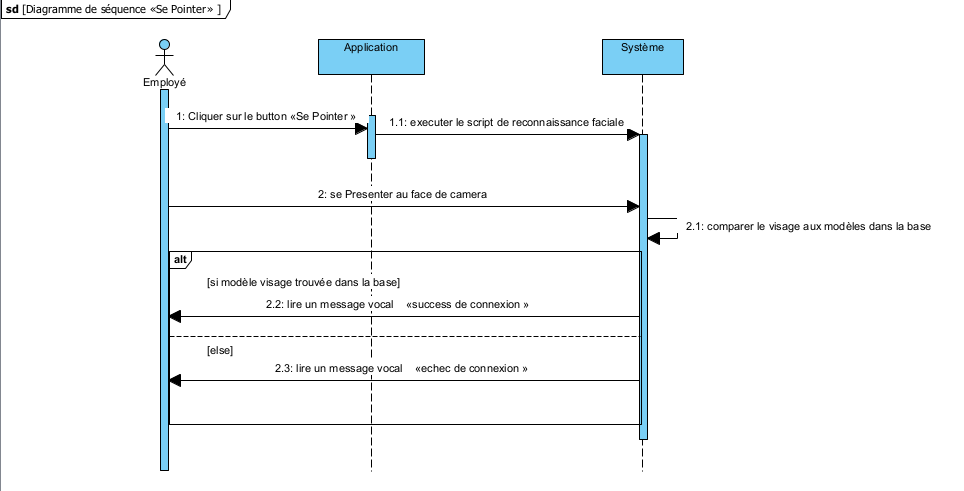
Les diagrammes de séquence sont la représentation graphique des interactions entre acteurs et le système selon un ordre chronologique dans la formulation UML. [11]

Le but est de décrire comment se déroulent les actions entre les acteurs ou objet.

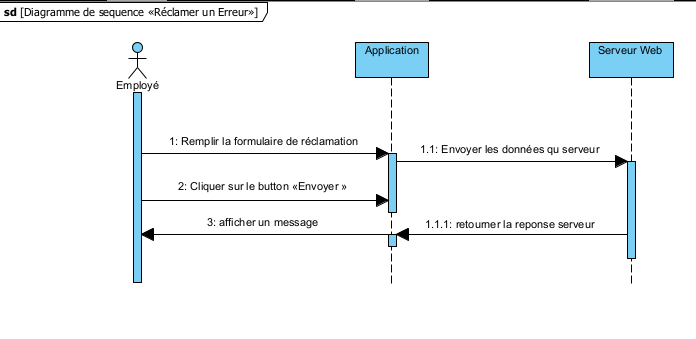
1. **Diagramme de séquence « Se Connecter »**



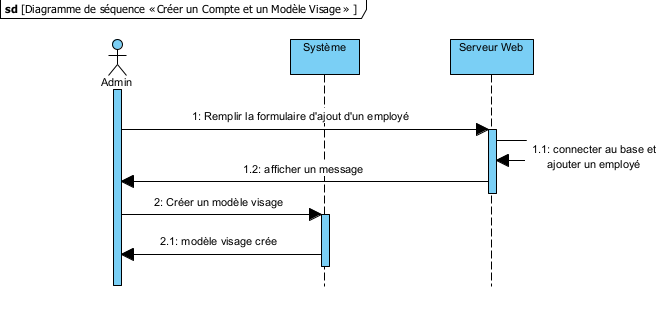
1. **Diagramme de séquence « Se Pointer »**



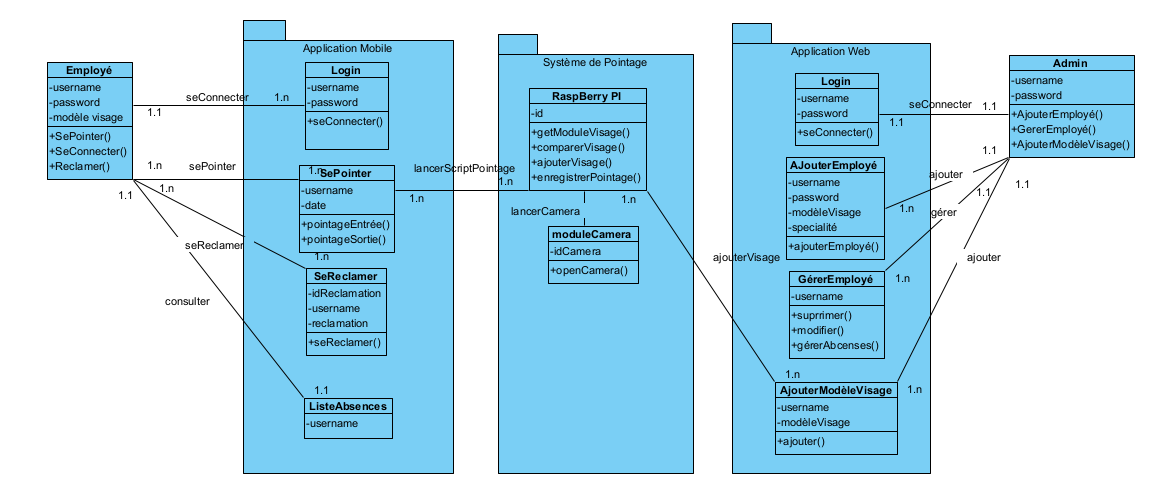
1. **Diagramme de séquence « Réclamer un Erreur »**



1. **Diagramme de séquence « Créer un Compte et un Modèle Visage »**



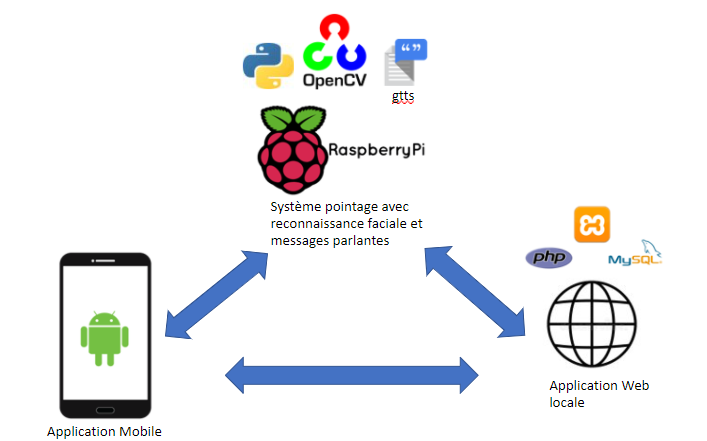
1. **Diagramme de classes :**



1. **Architecture Technique :**

Dans ce projet on trouve généralement Trois parties principales :

* L’application mobile
* Le serveur web (local)
* Le Système de pointage avec reconnaissance faciale (Raspberry Pi + module camera)



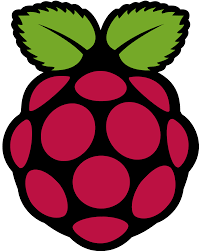
1. **Environnement matériel**

Au cours de notre projet, nous avons utilisé comme environnement matériel les éléments suivants, avec leurs spécifications techniques :

* Laptop Dell Inspiron 3543 :
* Processeur : Intel(R) Core i5-5200U 2.2 GHz
* Disque dur : 128Go SSD +1TB HDD
* Système d’exploitation Windows 11
* Mémoire installe (RAM) : 8 Go
* Carte Raspberry Pi
* Module Camera pour Raspberry Pi(on a utilisé une webcam pour ce cas ).
* Smartphone Android

1. **Environnement** **logiciel**

Pour la réalisation de cette application j’ai utilisé les logiciels et les Frameworks suivants :

* Description: File:Visual Studio Code 1.35 icon.svg - Wikimedia Commons **Visual Code :** pour développer les pages HTML et PHP et encore pour le script CSS et JavaScript.
* Logo Xampp PNG transparents - StickPNG **XAMPP :** est un logiciel sans licences GNU permettant d’émuler un Apache sous Windows. Il permet d’utiliser des fichiers PHP et d’installer des bases de données MYSQL local sur un ordinateur.
* MySQL — Wikipédia **Le SGBD MySQL**
*  **Un logiciel Visual Paradigm :** est un logiciel de création de diagramme dans le cadre d’une programmation. Tout en un, il possède plusieurs options permettant une large possibilité de modélisation en UML. Pour la création des diagrammes.
*  **Raspberry Pi OS** : Raspberry Pi OS est un système d'exploitation basé sur Debian pour Raspberry Pi. Depuis 2015, il est officiellement fourni par la Fondation Raspberry Pi
*  Python : le langage de programmation célèbre
*  Librairie OpenCV (pour Open Computer Vision) : est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel.
*  **gTTS (Google Text-to-Speech) :** est une bibliothèque Python pour s'interfacer avec l'API text-to-speech de Google Translate pour transformer les messages écrits en des messages vocale.
*  Android Studio est l'environnement de développement intégré officiel pour le système d'exploitation Android de Google, construit sur le logiciel IntelliJ IDEA de JetBrains et conçu spécifiquement pour le développement Android
* **JSch** est une implémentation Java pure de SSH2. JSch vous permet de vous connecter à un serveur ssh (Raspberry pi pour notre cas) et d’exécuter un script à distance.

1. **Réalisation :**
   * + 1. **Application Android** :

L’application réalisée est composée d’un ensemble d’interfaces permettant de connecter au compte utilisateur si vous oubliez vos informations de connexion vous pouvez contacter l’admin en cliquant sur le bouton « mot de passe oubliée » une interface réclamation s’affiche avec une formulaire préinscrit .en s’authentifiant un interface pointage apparait qui contient trois bouton pointage entrée, pointage sortie et réclamation ,le bouton réclamation affiche la page réclamation ,un petit bouton paramètre permet d’afficher l’interface de configuration pour changer l’adresse IP web ou raspberry pi

Dans ce qui suit, nous présenterons les interfaces de l’application réalisée en essayant de décrire les différents objets interactifs mis à la disposition de l’utilisateur.

* 1. **interface « page de démarrage »**

Lors de lancement de notre application, une interface écran de démarrage (Splash Screen) apparaît pendant 3 secondes mentionnant le nom de l’application comme le montre la figure suivante (figure 1).

* 1. **Interface de connexion**

une interface de connexion s’affiche lorsque l’utilisateur lance l’application. Pour s’authentifier il doit saisir ses informations de connexion fournis par la société et clique sur le bouton « Connecter » comme le montre la figure suivante. (Figure 2)

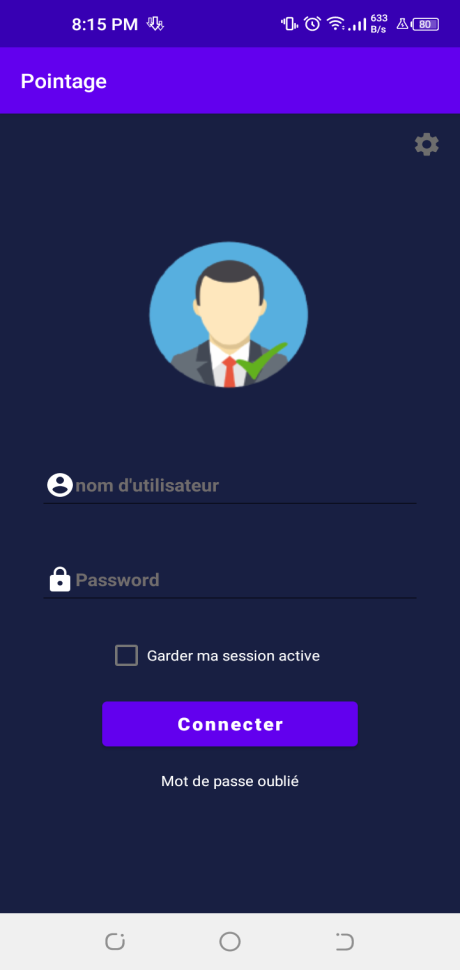


Figure 1 écran de démarrage

Figure 2 page de connexion

* 1. Interface Pointage :

C’est l’interface principale de notre application ou l’employé peut lancer le processus de pointage par reconnaissance faciale par le bouton pointage entrée ou sortie (figure 3) il peut aussi se réclamer en cas de problème de pointage pour informer l’administrateur par un email pour corriger ce soucis (figure 4) ou bien accéder au paramètre de l’application (figure 5).

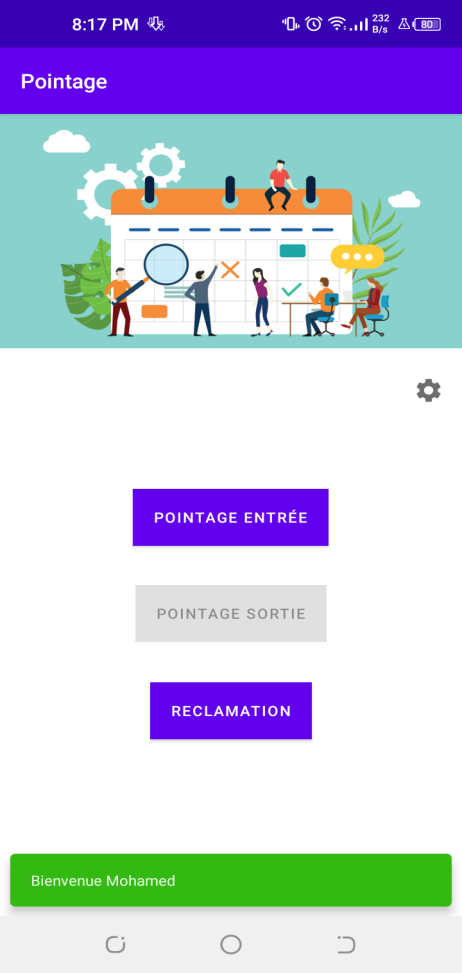
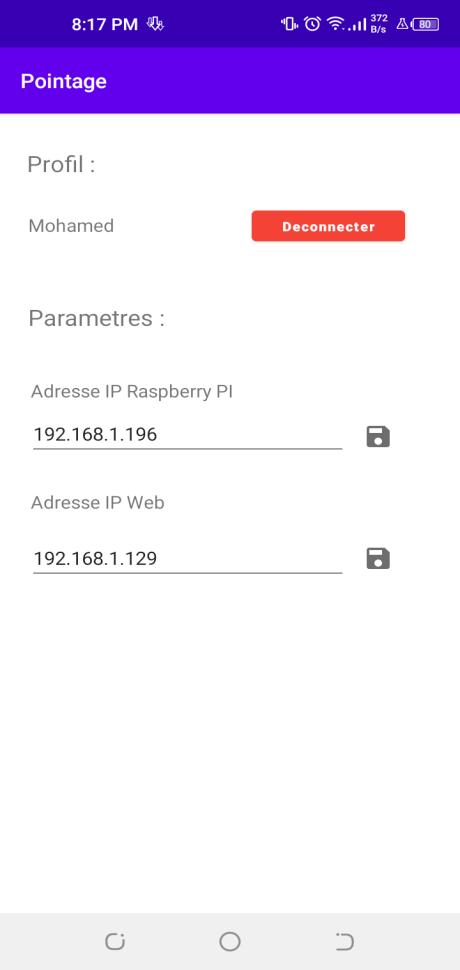


Figure 5 interface paramètre

Figure 4 interface reclamation

Figure interface pointage

**2. Système de pointage par reconnaissance faciale (Raspberry Pi + Camera) :**

le système de pointage est réalisé à l’aide de la carte raspberry pi ou on n’a installé open cv la bibliothèque graphique célèbre pour traitement d’image en temps réel et gTTS pour convertir nos messages écrites en des speech et requests pour les requêtes http

notre code se repartie en deux fichiers

* **créer\_modele.py**: code python pour demander d’employé ses informations (cin ,nom, mot de passe …) et lancer le processus de création de modelé facial et enfin créer un compte en enregistrant ses informations dans la base de données .le résultat d’exécution de cette commande est montrée dans la figure 6.
* **pointage.py :** code python pour la reconnaissance faciale en premier lieu il va créer les fichiers sons des messages et par la suite commencer la reconnaissance faciale si il reconnait l’employé il lit un message de bienvenue en indiquant son nom sinon il lit le message d’erreur , si le résultat d’opération est positif il va exécuter une requête web pour se pointer.

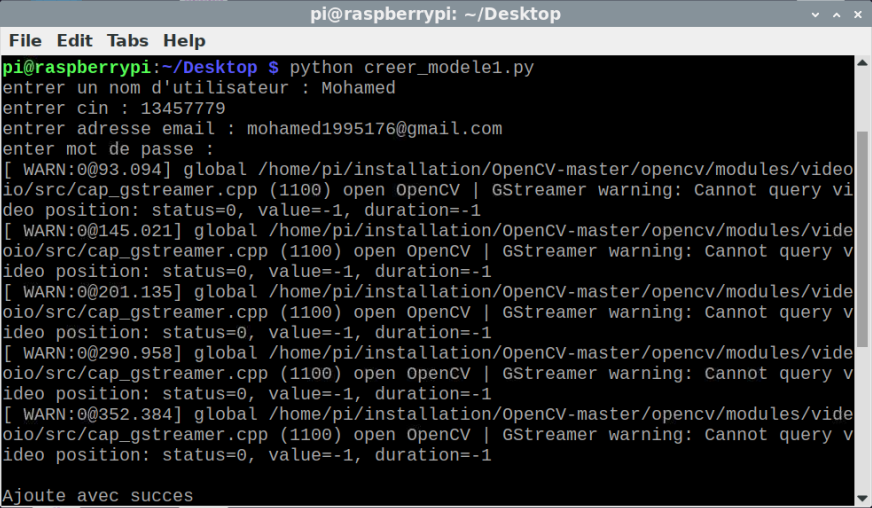
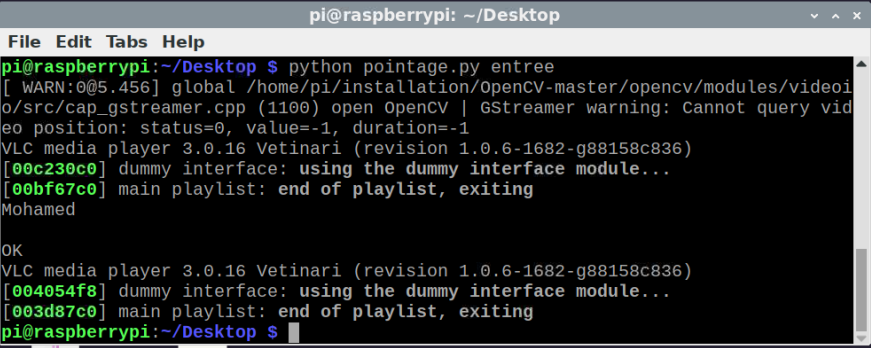
****

Figure 7 pointage.py

Figure 6 creer\_modele.py