

Réf : Ing-GI-2016-04

# Rapport de Projet de Fin d'Études

Pour obtenir le

**Diplôme d'Ingénieur en Génie Informatique**

**Option : Systèmes, Réseaux et Sécurité**

Présenté et soutenu publiquement le 23 juin 2016

Par

**Rabii AOUN**

## Développement d'une application Android pour le pointage des employés avec reconnaissance faciale

### Composition du jury

<b>Madame</b>	Besma FAYECH	<b>Président</b>
<b>Madame</b>	Narjas HACHANI	<b>Rapporteur</b>
<b>Monsieur</b>	Hassen MZOURI	<b>Encadrant Entreprise</b>
<b>Monsieur</b>	Ramzi FARHAT	<b>Encadrant ENSIT</b>
<b>Monsieur</b>	Mohamed ALI KHENISSI	<b>Encadrant ENSIT</b>

**Année universitaire : 2015-2016**

# Dédicaces

## **A mon très cher père Ameer**

A mon premier maître à qui je dois tout, celui qui a guidé mes pas, celui qui m'a transmis la bonté et m'a appris la confiance, le courage et la patience.

Vos sacrifices et vos encouragements ont été pour moi le meilleur gage de réussite, vous avez tout fait pour me garantir un avenir radieux. Vous m'avez inculqué de vraies valeurs dans cette vie. Je vous dois tout et je ferai de mon mieux pour rester un sujet de fierté à vos yeux.

Que ce travail soit un modeste témoignage de mon amour et mon attachement. Que DIEU vous préserve en bonne santé et une longue vie.

Merci d'avoir été toujours là pour moi, je vous aime fort.

## **A ma très chère mère Samia**

Aucune dédicace ne saurait exprimer ma profonde affection et mon immense respect.

Aucun hommage ne pourrait être à la hauteur de tes sacrifices.

Vos encouragements ont été pour moi source de gage de réussite.

Je suis très fière d'avoir réalisé ce que tu as tant espéré et attendu de moi.

Que DIEU te garde et te procure longue vie et santé, je vous aime fort.

## **A mes frères Habib et Achraf**

Votre amitié m'a toujours entouré et aidée.

Que DIEU puisse sauvegarder notre solidarité et notre indéfectible attachement familial.

A tous ceux à qui je pense, qui comptent pour moi et à qui j'ai senti essentiel de leur dédier ce travail.

# **Remerciements**

## **Merci à Dieu avant et après tout.**

C'est avec un grand plaisir que j réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui m'ont aidé à la réalisation de ce travail.

Je remercie très sincèrement les membres du jury pour m'avoir honorée en acceptant d'évaluer mon travail.

Je tiens à adresser une particulière reconnaissance à mon encadreur chez TIT, Mr Hassen Mzouri qui m'a encadré avec patience et pour son effort durant la réalisation de mon projet de fin d'études.

Je tiens aussi à remercier également mon encadrant Mr Ramzi Farhat, Enseignant Chercheur à l'Ecole Nationale Supérieure d'Ingénieurs de Tunis pour sa persévérance, pour ses efforts et ses conseils prodigués dans le suivi de l'avancement du projet.

Je dois aussi beaucoup à mes enseignants à l'Ecole Nationale Supérieure d'Ingénieurs de Tunis pour leurs dévouements et la qualité de la formation qu'ils m'ont offerte tout au long de mon cursus universitaire.

# Table de matière

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre 1 : Présentation du projet.....</b>	<b>4</b>
<b>Introduction .....</b>	<b>4</b>
<b>1. Cadre du projet .....</b>	<b>4</b>
<b>2. Présentation de la société d'accueil.....</b>	<b>4</b>
2.1. Introduction.....	4
2.2. Objectifs de la société .....	5
2.3. Projets achevés.....	5
<b>3. Problématique.....</b>	<b>5</b>
<b>4. Travail demandé.....</b>	<b>6</b>
<b>Conclusion .....</b>	<b>7</b>
<b>Chapitre 2 : Etat de l'art .....</b>	<b>8</b>
<b>Introduction .....</b>	<b>8</b>
<b>1. ERP Odoo 9.....</b>	<b>8</b>
1.1. Définition d'un ERP.....	8
1.2. Introduction à l'ERP Odoo 9 .....	9
1.3. Choix de l'Odoo9 .....	10
<b>2. Communication avec l'ERP Odoo9 .....</b>	<b>10</b>
2.1. Envoie des données avec JSON .....	10
2.2. Envoie des données avec le web service XML-RPC.....	11
<b>3. Détection des visages .....</b>	<b>11</b>
3.1. Méthodes basées sur les connaissances acquises .....	12
3.2. Méthodes basées sur l'apparence.....	12
3.2.1. Définition.....	12
3.2.2. Méthodes de Viola and Jones.....	13
3.2.2.1. Définition .....	13
3.2.2.2. Etape d'apprentissage .....	13
3.2.2.3. Etape de détection .....	14
3.3. Méthodes basées sur les caractéristiques invariantes.....	14
<b>4. Reconnaissance des visages .....</b>	<b>15</b>

4.1.	Méthodes locales pour la reconnaissance des visages .....	16
4.2.	Méthodes globales pour la reconnaissance des visages .....	16
4.3.	Méthodes hybrides pour la reconnaissance des visages .....	17
4.4.	Méthodes classiques pour la reconnaissance des visages .....	17
<b>Conclusion .....</b>		<b>17</b>
<b>Chapitre 3 : Analyse des besoins.....</b>		<b>19</b>
<b>Introduction .....</b>		<b>19</b>
<b>1. Besoins fonctionnels .....</b>		<b>19</b>
1.1.	Gestion des données des employés .....	19
1.2.	Stockage des caractéristiques des visages des employés .....	20
<b>2. Besoins non fonctionnels .....</b>		<b>20</b>
2.1.	Utilisation de l'ERP Odoo .....	20
2.2.	Utilisation d'Android.....	20
2.3.	Sécurité.....	20
2.4.	Simplicité et convivialité de l'interface graphique .....	20
2.5.	Rapidité d'exécution.....	20
<b>3. Diagrammes de cas d'utilisation .....</b>		<b>21</b>
3.1.	Diagramme de cas d'utilisation globale .....	21
3.2.	Diagramme de cas d'utilisations détaillés .....	22
3. 2.1	Diagramme de cas d'utilisation de l'ajout d'un employé.....	22
3. 2.2	Diagramme de cas d'utilisation pour modifier les coordonnées d'un employé et supprimer un employé .....	23
3. 2.3	Diagramme de cas d'utilisation pour le pointage des employés avec reconnaissance faciale .....	24
<b>Conclusion .....</b>		<b>25</b>
<b>Chapitre 4 : Conception.....</b>		<b>26</b>
<b>Introduction .....</b>		<b>26</b>
<b>1. Langage UML .....</b>		<b>26</b>
1.1.	Présentation .....	26
1.2.	Avantages .....	27
<b>2. Conception détaillée .....</b>		<b>27</b>
2.1.	Module d'ajout des employés .....	27
2.1.1.	Ajout du visage de l'employé dans notre base de données.....	28
2.1.2.	Ajout des informations personnelles dans le module ressources humaines de notre ERP Odoo 9 .....	28

2.1.3.	Diagramme de séquence d'ajout d'un employé .....	28
2.2.	Module de détection de visages .....	29
2.2.1	Fonction de classification .....	30
2.2.2	Bloc de décision .....	30
2.3.	Module de reconnaissance de visages .....	31
2.3.1.	Utilisation du module de détection de visage.....	31
2.3.2.	Etape d'identification .....	32
2.4.	Module de pointage des employés .....	32
2.4.1.	Fonctionnement .....	32
2.4.2.	Diagramme de séquence de pointage des employés avec reconnaissance faciale.....	32
2.5.	Diagramme de classes .....	33
<b>Conclusion .....</b>		<b>35</b>
<b>Chapitre 5 : Réalisation .....</b>		<b>36</b>
<b>Introduction .....</b>		<b>36</b>
<b>1. Environnement de Travail.....</b>		<b>36</b>
1.1.	Environnement matériel .....	36
1.2.	Environnement logiciel.....	37
<b>2. Application de pointage avec reconnaissance faciale.....</b>		<b>40</b>
2.1.	Développement du module de pointage .....	41
2.2.	Développement de l'application Android .....	41
<b>3. Chronogramme du projet.....</b>		<b>50</b>
<b>Conclusion .....</b>		<b>51</b>
<b>Conclusion générale .....</b>		<b>52</b>
<b>Références .....</b>		<b>54</b>
<b>Liste des Acronymes.....</b>		<b>56</b>
<b>Annexe 1 .....</b>		<b>57</b>
<b>Annexe 2 .....</b>		<b>61</b>

# Liste de figures

Figure 1: Les différentes applications disponibles dans l'ERP Odoo 9.....	9
Figure 2: Les descripteurs de Harr .....	13
Figure 3: Les deux premiers descripteurs de Haar sélectionnés dans la méthode Viola and Jones. ....	14
Figure 4: Diagramme de cas d'utilisation globale .....	21
Figure 5: Diagramme de cas d'utilisation d'ajout d'un employé.....	23
Figure 6: Diagramme de cas d'utilisation pour modifier les coordonnées d'un employé et supprimer un employé .....	24
Figure 7: Diagramme de cas d'utilisation pour le pointage des employés avec reconnaissance faciale .....	25
Figure 8: Diagramme de séquence d'ajout d'un employé.....	29
Figure 9: Diagramme de séquence de pointage des employés .....	33
Figure 10: Diagramme de classes.....	34
Figure 11: Interface du module pointage des employés.....	41
Figure 12: Première interface de notre application .....	42
Figure 14: Interface d'installation de l'open CV .....	44
Figure 16: Interface d'ajout des informations personnelles de l'employé .....	45
Figure 17: Interface d'ajout d'un employé au niveau du module ressources humaines de l'ERP Odoo9.....	46
Figure 18: Première interface de pointage d'entrée .....	47
Figure 19: Deuxième interface de pointage d'entrée .....	47
Figure 20: Deuxième interface de pointage de sortie.....	48
Figure 21: Interface de pointage au niveau du module pointage des employés dans l'ERP Odoo9.....	49
Figure 22: Interface lorsque le visage n'est pas identifié .....	49
Figure 23: Interface en modifiant quelques aspects extérieurs .....	50
Figure 24: Chronogramme d'activités .....	50
Figure 25: A.1 - Architecture du Système d'exploitation Android .....	58
Figure 26: A.2 - Cycle de vie d'une application Android.....	60

## Introduction générale

La détermination de l'identité d'un individu d'une manière efficace et fiable est un problème critique. En effet, les domaines de la surveillance et de la sécurité sont des domaines de très grandes importances dans notre vie quotidienne. Chaque jour, sans qu'on se rende compte, notre identité est vérifiée par des multiples organisations : lorsqu'on utilise notre carte bancaire, lorsqu'on se connecte à notre compte personnel en ligne et surtout lorsqu'on accède à notre lieu de travail.

Dans ce fait, contrôler l'entrée et la sortie des employés des lieux de travail est un défi important pour les sociétés de nos jours. On parle de pointage. Actuellement, il existe deux méthodes principales pour identifier un individu lors de l'entrée ou de la sortie de son lieu de travail. La première est fondée sur la possession d'un objet tel qu'une pièce d'identité, une clef ou un badge. La deuxième est fondée sur la pointeuse avec l'empreinte digitale. Ces méthodes sont utilisées pour obtenir une sécurité accrue pour l'identification d'un individu dès l'entrée à son lieu de travail. Cependant, ces deux méthodes classiques présentent des inconvénients; en effet, l'utilisation des cartes magnétiques, de clefs et de badges nécessitent qu'on les porte toujours sur soi et n'échappent pas au risque de vol pour falsifier l'identité d'un individu avec une facilité remarquable. De même, avec utilisation de la pointeuse avec l'empreinte digitale, on peut falsifier l'identifié d'un individu en utilisant des dispositifs de copie de l'empreinte digitale. De plus, ces pointeuses sont assez coûteuses. Egaleme nt, la plupart des solutions disponibles sont autonome, ne permettant pas de gérer le pointage de plusieurs locaux éparpillés d'une façon centralisée.

Tous ces inconvénients et difficultés ont donné naissance à l'idée d'utiliser les caractéristiques biométriques [1] du visage comme un moyen pour l'identification d'un individu. En effet, chaque personne possède des caractéristiques uniques et stables présentes sur son visage.

La biométrie du visage [2] apporte une grande simplicité et un confort pour ses utilisateurs. L'avantage de cette technique, qu'on peut l'appeler mot de passe biométrique, est qu'elle ne peut pas être violé ou transmise facilement à une personne. Mais surtout elle ne peut pas être oubliée. On peut prédire qu'il est fort probable que dans le futur proche, le mot de passe biométrique peut remplacer le mot de passe conventionnel surtout dans les domaines qui nécessitent un niveau de sécurité très élevé.



Quotidiennement, le visage humain est certainement la caractéristique biométrique que nous les humains utilisons le plus pour nous identifier. La reconnaissance de visages est devenue par conséquent l'un des systèmes les plus utilisés pour l'identification des individus.

Le but d'un système de reconnaissance de visages est de simuler le système de reconnaissance humain par la machine pour automatiser certaines applications telles que : la télésurveillance, l'accès aux bâtiments sécurisés et surtout l'accès des employés aux lieux de travaux. Le projet présenté dans ce rapport consiste à créer une application Android pour le pointage des employés avec reconnaissance faciale. Ce projet va permettre d'identifier les visages des employés pour effectuer leurs pointages d'entrées et de sorties. Ces horaires d'entrées et de sorties seront visibles dans l'ERP Odoo 9 qu'on utilisera aussi pour gérer les employés de l'entreprise.

Avant d'entamer la réalisation de ce projet, il faut d'abord choisir une méthodologie qu'on doit la respecter et suivre le long de notre projet. Donc, nous avons opté pour la méthodologie en spirale vu que nous allons améliorer notre application à chaque itération. La modèle en spirale est un modèle de cycle de développement logiciel qui contient les différentes étapes du modèle du cycle en V. Avec la réutilisation des versions développées, ce cycle recommence son fonctionnement en produisant un produit plus complet et fiable.

Le présent rapport est structuré en cinq chapitres. Dans le premier chapitre intitulé "Présentation du projet", nous commencerons par la présentation générale de notre projet et de l'organisme accueillant. Après, nous allons définir la problématique ainsi que les objectifs que nous devons atteindre pour la réalisation de notre système de pointage avec reconnaissance faciale.

Dans le deuxième chapitre intitulé "Etat de l'art", nous allons présenter les méthodes existantes permettant de résoudre les différentes problématiques posées dans le cadre de notre projet. C'est au niveau de ce chapitre que nous évaluerons les différentes les méthodes que nous utiliserons pour mettre en œuvre notre application de pointage.

C'est au niveau de ce chapitre que nous allons effectuer une décision concernant les méthodes que nous allons utiliser pour mettre en œuvre notre application de pointage.

Dans le troisième chapitre intitulé "Analyse de besoin", nous allons définir les besoins fonctionnels et non fonctionnels auxquels notre projet doit répondre. Ensuite nous allons analyser la faisabilité organisationnelle et technique du projet.

Dans le quatrième chapitre intitulé "Conception", nous allons présenter la conception générale et détaillée de notre projet.

Dans le dernier chapitre intitulé "Réalisation", nous allons présenter les outils et les méthodes d'implémentations et de développement utilisées qui ont mené à la réalisation de notre solution mobile pour le pointage des employés. A la fin de ce chapitre, nous allons présenter les détails d'implémentation, les tests d'exécution et les résultats aboutis.

# **Chapitre 1 : Présentation du projet**

## **Introduction**

Dans ce premier chapitre introductif, nous allons présenter notre projet intitulé "Développement d'une application Android pour le pointage des employés avec reconnaissance faciale ". Ceci sera fait à travers un aperçu général sur le cadre général du projet et une présentation de la société TIT (Tunisian Information Technology). Nous allons également fournir une description de notre projet en posant la problématique. Il s'agit d'une présentation des problèmes courants relatifs au pointage d'un point de vue technique et technologie. Nous allons également identifier les objectifs principaux auxquels notre projet doit répondre et les vérifier.

### **1. Cadre du projet**

Il s'agit d'un projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique au sein de l'ENSIT : Ecole Nationale Supérieure d'Ingénieurs de Tunis. Ce projet, intitulé " Développement et conception d'une application Android de reconnaissance faciale pour le pointage", a été réalisé à la société TIT (Tunisian Information Technology) durant la période qui s'étalait du 15 février 2016 jusqu'à 15 juin 2016. Il est encadré par Mr Hassan Mzouri directeur général du TIT et par Dr Ramzi Farhat enseignant chercheur à l'ENSIT.

### **2. Présentation de la société d'accueil**

#### **2.1. Introduction**

La Société TIT (Tunisian Information Technology) est une société d'ingénierie logicielle. Elle est spécialisée dans la réalisation de projets mettant en œuvre des technologies web et surtout mobile. Elle a été créée en 2007 par Monsieur Mzoughi Hssan à Tunis en tant que société à responsabilité limitée.

## **2.2. Objectifs de la société**

Les principaux objectifs de la société TIT sont : Le développement et la promotion de nouvelles applications web et mobiles, ainsi que la distribution des ces nouvelles applications pour mieux se rapprocher des clients et satisfaire leurs besoins.

La société TIT est en coordination avec la société française "Universymo", qui est une société spécialisée pour l'enseignement en ligne. Donc, plusieurs des applications et produits distribués par "Universymo" sont développés dans la société TIT.

La société TIT est en train de préparer sa propre campagne mailing pour toucher une cible plus large dans le monde, plus exactement les pays européens comme la France, La Belgique, L'Angleterre, L'Espagne et le Portugal.

## **2.3. Projets achevés**

Les projets réalisés au sein de la société TIT sont des applications mobiles et des applications web.

Parmi les projets réussis et utilisés au pays européens, on peut citer l'application " GPS Winek " : C'est une application mobile utilisable avec toutes les plateformes mobiles pour faire le suivi des employés dans les sociétés.

Les autres applications sont : " Bwing Taxi " et " SFR taxi " qui sont des applications mobiles et web utilisées par les conducteurs de taxi en France.

## **3. Problématique**

Dans cette partie, nous entreprenons une étude critique des solutions de pointages existantes. Les deux meilleures solutions que les entreprises utilisent de nos jours sont : la pointeuse avec l'empreinte digitale et le pointage en utilisant un badge électronique personnalisé.

Les inconvénients de ces solutions classiques sont qu'ils sont peu fiables et très couteuse pour les sociétés.

Commençons par la solution avec le badge électronique personnalisé, cette solution est peu fiable, parce qu'il ya une chance qu'un employé peut pointer à la place d'un autre en utilisant son badge électronique sans que la partie administrative le sache.

Les entreprises qui utilisent cette solution de pointage, doivent avoir des badges électroniques pour chacun de ses employés, ce qui rend cette solution très couteuse pour les entreprises avec un grand nombre d'employés.

L'autre solution de pointage qu'utilise la plupart des sociétés est la pointeuse avec empreinte digitale. Cette solution est couteuse, vu qu'une société doit avoir un minimum de deux ou trois dispositifs de pointage dans son siège.

En effet, si le nombre d'employés est important, l'usage d'une seule pointeuse risque de créer un goulot d'étranglement aux heures d'entrées et de départ. En plus, la solution de pointage avec l'empreinte digitale stocke les données des horaires de pointage de ses employés localement. Ce qui montre des limites lorsqu'on veut avoir une solution distribuée qui utilise une base de données partagées d'empreintes d'employés. C'est le cas en particulier des entreprises ayant plusieurs locaux distribuées géographiquement.

Même si on peut trouver des solutions pour surmonter ce problème, ces solutions restent coûteuses et compliquées à mettre en œuvre. De plus, l'autre inconvénient de cette solution est qu'un employé peut pointer à la place d'un autre en utilisant des dispositifs de copie d'empreintes digitales.

On peut résumer que les deux problèmes majeures de ces solutions existantes est qu'elles sont coûteuses à mettre en œuvre et peu fiables. Donc, l'objectif est de développer une solution de pointage moins coûteuse et plus fiable.

## **4. Travail demandé**

Notre travail consiste à réaliser une application Android pour le pointage des employés avec reconnaissance faciale.

Notre solution de pointage est plus fiable, vu que nous allons utiliser la reconnaissance faciale pour le pointage des employés. Donc, il est très difficile, surtout avec la présence des caméras de surveillance à l'endroit de pointage, de tromper notre système de pointage et de pointer à la place d'un autre employé via une photo représentant son visage par exemple.

Donc, notre système de pointage que nous allons développer permettra aux employés de faire leurs pointages d'entrées et de sorties en utilisant notre application Android pour s'identifier avec reconnaissance faciale.

Ces horaires d'entrées et de sorties des employés seront stockées dans l'ERP Odoo 9 installé dans le serveur de l'entreprise. Plus précisément, ces horaires seront stockés via un module spécifique que nous devons développer et intégrer à l'ERP Odoo 9.

Après, nous devons synchroniser ce module de pointage avec le module Ressources Humaines disponible dans l'ERP. Ce module contiendra toutes les informations personnelles des employés.

Donc, pour la réalisation de notre système de pointage avec reconnaissance faciale, nous avons besoin de : Premièrement, développer un module de pointage des employés et l'intégrer dans notre ERP Odoo 9. Ce module de pointage contiendra les horaires de pointages des employés effectués avec notre application Android, lors de leurs entrées et de sorties.

Ensuite, une fois le module de pointage est développé et installé dans notre ERP, il faut développer une application Android de pointage pour les employés avec reconnaissance faciale.

## **Conclusion**

Ce premier chapitre a abordé le contexte général de notre projet de fin d'études. Le cadre du projet et la présentation de la société TIT ont été suivis d'une description de la problématique de notre projet ainsi que l'introduction à nos objectifs et fonctionnalités principales de notre système de pointage.

Maintenant, avant d'envisager une solution algorithmique à implémenter, il convient d'étudier l'état de l'art en présentant l'ERP Odoo 9 que nous allons utiliser et en étudiant les parties qui concerne la connexion avec l'ERP Odoo9, la détection des visages et la reconnaissance faciale.

## Chapitre 2 : Etat de l'art

### Introduction

Ce projet comporte trois tâches principales. Premièrement, la connexion vers l'ERP Odoo 9. Nous allons utiliser cette connexion pour transférer nos données du Smartphone vers la base de données de l'ERP. Les données sont : les informations personnelles des employés lors de l'inscription et les horaires de pointages.

Deuxièmement, la détection des visages lors de la phase de l'ajout du visage de l'employé.

Troisièmement, la partie de la reconnaissance faciale que nous allons utiliser pour le pointage des employés via notre application Android.

Dans ce deuxième chapitre, nous commençons par introduire l'ERP Odoo9. Ensuite, nous allons citer les différentes méthodes disponibles pour réaliser les différentes tâches de notre projet.

### 1. ERP Odoo 9

#### 1.1. Définition d'un ERP

Un ERP (Entreprise Ressource Planning) [3] est définie comme un groupe de modules qui sont reliés à une base de données unique.

C'est est un progiciel permettant la gestion d'un ensemble des processus opérationnels d'une entreprise.

Il permet d'intégrer plusieurs fonctions de gestion pour bien gérer l'entreprise. Parmi ces fonctions, nous pouvons citer [4] : solution de gestions des employés, solution de gestions des commandes, solutions de gestion des stocks etc.

Autrement dit, un ERP est représenté comme la "colonne vertébrale" d'une entreprise. En effet, il est très difficile de bien gérer une grande entreprise sans l'utilisation d'un ERP.

## 1.2. Introduction à l'ERP Odoo 9

L'ERP Odoo [5], anciennement appelé OpenERP, est un progiciel de gestion des principaux services de l'entreprise adapté à toutes les entreprises et en particulier les PME (Petite et Moyenne Entreprise). Il est très modulable et sans coût de licence, il se caractérise par déploiement rapide à un coût maîtrisé.

L'ERP Odoo9 contient plus de 4000 modules gratuits et intégré au niveau des traitements offerts. Nous pouvons trouver différents modules qui sont toutefois parfaitement compatibles entre eux.

Il est utilisé par plus de 2000000 d'utilisateurs à travers le monde, en grande majorité issus des pays émergents [6].

Parmi les avantages de l'ERP Odoo, le fait qu'il a une communauté de plus de 1400 contributeurs. Il est développé en utilisant le langage Python, qui est un langage de programmation moderne. De plus, il est gratuit et même les versions payantes ont un coût très faible par rapport aux autres ERP.

La figure 1 correspond aux différentes applications classées par catégorie dans l'ERP Odoo 9.

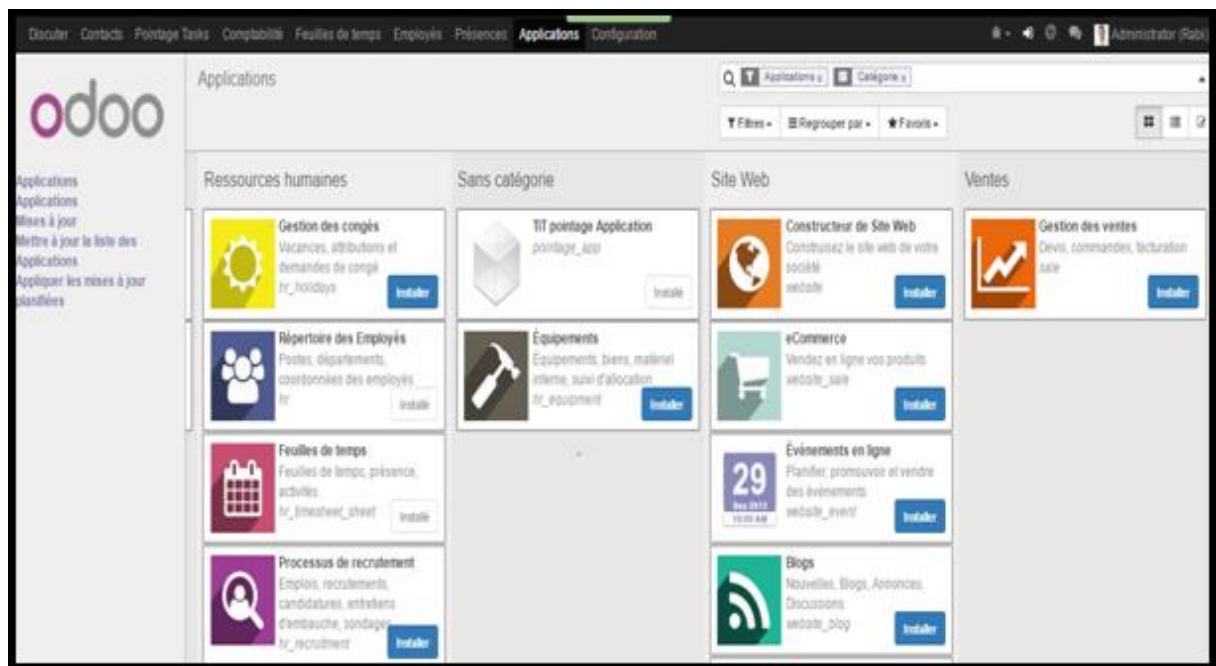


Figure 1: Les différentes applications disponibles dans l'ERP Odoo 9



### **1.3. Choix de l'Odoo9**

Les deux ERP d'Odoo qu'on utilise les plus sont Odoo 9 et Odoo 8. Nous avons opté pour choisir l'Odoo 9 vu les avantages qu'il apporte par rapport à l'Odoo 8.

Premièrement, les modules développés sur l'Odoo 8 fonctionnent parfaitement avec l'Odoo 9. Alors que les modules développés sur Odoo 9 ne fonctionnent pas sur Odoo 8[7]. Deuxièmement, l'Odoo 9 arrive avec des nouvelles fonctionnalités et amélioration qui facilitent notre travail. Enfin, vu que l'ERP Odoo est un ERP Open Source, la plupart des développeurs concentrent leurs travaux sur le développement des modules sur Odoo 9 qu'Odoo 8.

Cela est dû grâce à la migration de plusieurs utilisateurs vers l'Odoo 9.

## **2. Communication avec l'ERP Odoo9**

Il existe plusieurs méthodes pour l'envoi des données d'une application Android vers un ERP installé sur un serveur. Nous sommes appelés à utiliser l'une de ces méthodes pour stocker les informations personnelles des employés qu'on va saisir dans notre Smartphone lors de la phase d'ajout des employés.

Nous allons l'utiliser aussi pour stocker l'horaire de pointage d'entrée et de sortie des employés lors de la phase de pointage avec reconnaissance faciale.

Les méthodes les plus utilisées sont : l'envoi en utilisant le langage JSON et l'envoi avec le web service XMLRPC.

### **2.1. Envoie des données avec JSON**

Le JSON (Java Script Object Notation) est un format d'échange de données permettant la communication entre client et serveur.

C'est un format textuel, ce qui nous donne l'avantage de coder et de décoder facilement les données [8].

Parmi les avantages de l'utilisation du JSON pour l'échange des données, nous avons la vitesse de traitement des données. De plus, nous n'aurons pas besoin de parser les fichiers XML pour extraire les informations.

Enfin, en utilisant le langage JSON, on peut échanger les contenus binaires entre client et serveur avec une représentation textuelle spéciale en utilisant les commandes : `new Buffer(file)` et `toString("base64")`.

## **2.2. Envoie des données avec le web service XML-RPC**

Le XML-RPC (Extensible Markup Language-Remote Procedure Call) est un protocole RPC [9] permettant de faire appel à des fonctions et méthodes utilisables sur un serveur distant et à travers un réseau web.

Ce web service utilise le langage XML pour encoder ses appels et le protocole HTTP comme un mécanisme de transport.

Son fonctionnement est simple : le XML-RPC envoie une requête HTTP à un serveur en mettant en œuvre le protocole. Le client dans notre cas peut être une application web ou mobile ou un logiciel voulant exécuter une méthode ou fonction unique d'un système distant.

En utilisant ce web service, on pourra transmettre plusieurs paramètres comme entrée et en attendant une valeur de retour qui sera renvoyée comme réponse à notre demande.

## **3. Détection des visages**

La détection des visages a une grande importance dans notre projet car elle est la base de la reconnaissance des visages humains dans l'image numérique que l'employé va prendre lors de son pointage d'entrée et de sortie.

La méthode de détection d'un visage sert à reconnaître un visage humain de l'image numérique qu'on va prendre et à le distinguer du reste de l'image.

Ils existent plusieurs méthodes de détection de visages [10] proposées pendant ces dernières années qui sont classées en trois catégories principales :

- Méthodes basées sur la connaissance acquise
- Méthodes basées sur l'apparence
- Méthodes basées sur les caractéristiques invariantes

### **3.1. Méthodes basées sur les connaissances acquises**

Ces méthodes sont basées sur les règles qui représentent les composants principaux des visages humains [9]. Parmi ces règles, nous pouvons citer deux exemples.

- Exemple 1 : Un visage humain dans une image numérique est défini par deux yeux symétriques, une bouche et un nez.
- Exemple 2 : Les relations entre les différents éléments faciaux d'un visage humain sont définies par leurs distances et positions relatives.

Ces méthodes sont efficaces pour nous aider à détecter les visages humains. Mais le problème avec cette solution [11], est qu'il y a une difficulté de traduire les connaissances relatives aux visages humaines en règles bien définies pour les utiliser pour la détection des visages.

Cela va provoquer des erreurs de détection lorsqu'on prend une image et peut affecter négativement la fiabilité de notre système.

### **3.2. Méthodes basées sur l'apparence**

#### **3.2.1. Définition**

L'objectif des méthodes basées sur l'apparence est de déterminer les caractéristiques significatives des visages et des non visages à partir de techniques d'analyse statistique et d'apprentissage organisés par le biais de modèles de distribution. Le but de cette approche est de déterminer si un visage correspond à un visage humain ou non [10].

La classification visage ou non visage est représentée par une variable aléatoire  $x$  (dérivée d'une image ou d'un vecteur caractéristique). La variable  $x$  contiendra le résultat de l'analyse.

Parmi les méthodes basées sur l'apparence, nous pouvons citer : Les réseaux de neurones et l'algorithme de Viola and Jones.

### 3.2.2. Méthodes de Viola and Jones

#### 3.2.2.1. Définition

La méthode de « Viola and Jones » proposée par les deux chercheurs Paul Viola and Michael Jones en 2001 [12] est la méthode la plus utilisée dans le domaine de la reconnaissance faciale pour la détection de visages. L'algorithme « Viola and Jones » est basé sur une étape d'apprentissage supervisée et une étape de détection.

#### 3.2.2.2. Etape d'apprentissage

L'étape de l'apprentissage supervisée nécessite pour son fonctionnement, les descripteurs de Harr (voir figure 2) qui se constituent de deux rectangles adjacents, l'un noir et l'autre blanc. Ces descripteurs sont superposés aux différentes régions de l'image prise à la recherche d'une zone de forte corrélation pour pouvoir détecter la présence éventuelle d'un visage dedans [12].



**Figure 2: Les descripteurs de Harr**

Avec l'utilisation des descripteurs de Harr [13] [14], L'algorithme de Viola and Jones va déterminer la ressemblance entre une zone prise de l'image en question et à ces descripteurs en soustrayant la moyenne des valeurs des pixels contenus dans la région noire à la moyenne des valeurs des pixels contenus dans la région blanche.

Après on peut utiliser la méthode d'AdaBoost [15] pour faire la comparaison entre la zone retenue de l'image après l'utilisation de descripteurs de Harr et le seuil définie par AdaBoost (voir annexe).

Le seuil de la méthode d'Adaboost est la combinaison de plusieurs classificateurs peu performants pour formuler un classificateur plus performant et fort.

La zone de l'image traitée va être retenue si la différence de ressemblance entre celle-ci est les descripteurs de Harr est supérieure à ce seuil de Adaboost.

Le fonctionnement des descripteurs de Harr est le suivant [14] : Pour bien caractériser un visage dans une image numérique, on commence par choisir le premier descripteur .Le choix du premier descripteur est basé sur la propriété que la région des yeux est souvent plus foncée que la région du nez et des joues (voir figure 3).Donc, on commence toujours par la région des yeux dans l'image prise.

Le choix du deuxième descripteur est basé sur le constat que les yeux sont généralement plus foncés que le pont du nez.



**Figure 3: Les deux premiers descripteurs de Haar sélectionnés dans la méthode Viola and Jones.**

### **3.2.2.3. Etape de détection**

La détection du visage consiste à balayer une image à l'aide d'une fenêtre de détection de taille initiale 24px par 24px et de déterminer si un visage y est présent. Lorsque l'image est parcourue entièrement, la taille de la fenêtre est augmentée et le balayage recommence, jusqu'à ce que la fenêtre fasse la taille de l'image.

## **3.3. Méthodes basées sur les caractéristiques invariantes**

Les algorithmes contenus dans cette approche basée sur les caractéristiques invariantes permettent de trouver les caractéristiques structurelles et faciales dans un visage même si l'image est prise dans des différentes conditions de luminosités, angle de vue et différents positions [10].

Ces méthodes sont utilisées pour remédier au problème de changements de la posture de visage. Ils se basent sur des caractéristiques invariables comme la couleur de peau, la texture et les contours [12].

Le problème qu'on peut rencontrer en utilisant ces méthodes est qu'elles ne sont pas efficaces si les images prises sont de faible luminosité, la qualité des images est diminuée ou si certaines parties du visage sont masquées.

Il existe des différentes caractéristiques invariables dans le visage qu'on peut utiliser dans cette approche dont la principale qu'on peut citer est la couleur de la peau.

## **4. Reconnaissance des visages**

Pendant les dernières années, on a développé plusieurs méthodes de reconnaissance de visages.

Ces méthodes développées consistent à reconnaître automatiquement l'identité d'une personne, déjà enregistré dans notre système de base de données, à partir d'une image prise de son visage.

La reconnaissance des visages se divisent en deux catégories [10] :

La reconnaissance des visages à partir des images fixes et la reconnaissance des visages à partir de séquences d'images (à partir d'une vidéo).

On peut constater que la reconnaissance des visages à partir d'une vidéo numérique est plus performante que celle basée sur les images fixes, puisque l'utilisation synchronique des informations temporelles et spatiales aide dans la reconnaissance des visages.

Mais, dans notre projet et pour la partie de la reconnaissance des visages, nous allons utiliser la reconnaissance des visages basée sur les images fixes, puisque c'est le type de données qui est à notre disposition.

La partie de la reconnaissance des visages se compose de deux phases principales nécessaires à son fonctionnement [10] :

- L'identification : Cette phase consiste à déterminer l'identité d'une personne parmi plusieurs identités inscrites dans notre base de données.
- La vérification : Cette phase est utilisée après la phase de l'identification, elle consiste à vérifier que l'identité choisie de l'individu de la phase de l'identification est la bonne.

Enfin, les méthodes de reconnaissance des visages basées sur les images fixes que nous allons utiliser pour faire la reconnaissance des employés lors de leurs pointage se déclinent en quatre grandes catégories: méthodes locales, méthodes globales, méthodes hybrides et les méthodes classiques.

#### **4.1. Méthodes locales pour la reconnaissance des visages**

Ces méthodes locales sont basées sur la notion de la géométrie. Le fonctionnement de ces méthodes locales [10] consiste à faire l'analyse du visage humain avec la description individuelle de ses parties et leurs relations. Elles permettent à appliquer les transformations possibles du visage humain pour les endroits spécifiques de l'image numérique prise pour l'individu. Ces transformations consistent sur les traits du visage et ces points caractérisant comme les coins des yeux, la grandeur de la bouche, le nez, etc.

Ces méthodes locales sont difficiles à mettre en place [10] [17] à cause de la "diversité des visages humaines dans la planète". Mais ils sont plus robustes et donnent un résultat presque parfait pour la reconnaissance des visages humains.

L'avantage des ces méthodes locales pour la reconnaissance des visages pour les individus réside dans la prise en compte de la particularité du visage humain en tant qu'objet à reconnaître, cela est dû à l'exploitation des résultats de la recherche en neuropsychologie et psychologie cognitive sur le système visuel humain une fois enregistré.

L'inconvénient de ces méthodes locales est lorsqu'on prend une image numérique d'un individu qui est flou, ce qui va entraîner un manque de précision dans la phase de "extraction des points" qui résulte à une difficulté lors de la phase de reconnaissance des visages.

Ils existent plusieurs méthodes pour les méthodes locales, la méthode principale est : La méthode de Détecteur de Harris.

#### **4.2. Méthodes globales pour la reconnaissance des visages**

Les méthodes globales sont des méthodes qui utilisent la région entière du visage comme une valeur d'entrée à l'algorithme de reconnaissance du visage [10]. L'image prise par l'individu va être enregistrée dans notre base de données sous format d'une matrice qui contient des valeurs

de pixels du visage pris en compte. Puis elle sera convertie en des vecteurs pour faciliter leur manipulation lors de la phase de l'indentification [17].

L'avantage de ces méthodes globales est qu'elles sont rapides en exécution et à mettre en œuvre.

En revanche, leur inconvénient se présente dont les variations de l'éclairement et l'expression du visage au moment de la prise de l'image de l'individu.

#### **4.3. Méthodes hybrides pour la reconnaissance des visages**

Les méthodes hybrides pour la reconnaissance des visages des individus consistent à faire la fusion de plusieurs méthodes pour rendre la phase de reconnaissance plus robuste et résoudre le problème d'indentification [10].

Ces méthodes hybrides qui utilisent une approche multi-classificatrice pour l'identification des visages est la meilleure solution pour remédier au problème.

L'inconvénient de ces méthodes hybrides est que la fusion de plusieurs méthodes utilisées pour faire la reconnaissance faciale peut ne pas donner un résultat fiable dans certains cas [10].

#### **4.4. Méthodes classiques pour la reconnaissance des visages**

Les méthodes classiques pour la reconnaissance des visages des individus consistent à utiliser l'une des méthodes de la phase de détection de visages et à comparer ses valeurs issues de cette méthode avec les valeurs enregistrées.

La fiabilité de cette méthode de reconnaissance des visages dépend de la fiabilité de la méthode de détection de visages [10].

## **Conclusion**

Dans ce deuxième chapitre Etat de l'art, nous avons présenté l'ERP Odoo que nous allons utiliser dans notre projet ainsi que notre choix pour lequel nous avons choisi la version 9.



Puis, nous avons cité les différentes méthodes que nous pouvons utiliser pour effectuer les tâches principales de notre système.

Pour la connexion avec l'ERP Odoo9, nous avons opté pour l'utilisation du web service xml-rpc vu qu'il est facile à mettre en œuvre et que nous pouvons transmettre plusieurs paramètres comme entrée avec une seule commande.

Pour la détection de visages, nous avons opté pour la méthode de Viola and Jones, l'une des méthodes basées sur l'apparence pour son meilleur taux de détection de visages avec un minimum de temps d'exécution pour le traitement d'un seul visage par image. Cette méthode contient un algorithme de détection d'objet Adaboost qui est devenu une référence mondiale pour la détection des objets en général par ses qualités de rapidité et efficacité.

Enfin, pour la partie de la reconnaissance des visages, nous choisissons la méthode classique, vu que nous avons déjà développé une solution de détection de visage en utilisant l'algorithme de Viola and Jones, et parce que cet algorithme est fiable.

## **Chapitre 3 : Analyse des besoins**

### **Introduction**

Dans ce chapitre, nous allons entamer l'analyse des besoins fonctionnels et non fonctionnels auxquels notre projet doit répondre. Notre analyse sera appuyée sur l'introduction des acteurs et les diagrammes de cas d'utilisations.

### **1. Besoins fonctionnels**

#### **1.1. Gestion des données des employés**

Nous entamons à présent l'étude de nos besoins fonctionnels auxquels notre projet doit répondre. Tout d'abord le premier besoin fonctionnel est celui de gérer les données des employés. Et d'après le cahier de charge cette gestion doit se faire au sein d'un ERP. Il faut que l'ERP intègre un module de gestion de ressources humaines permettant d'ajouter les employés avec toutes leurs informations personnelles.

#### **Gestion de pointage des employés**

Nous avons également besoin d'un module de pointage pour les employés que nous devons l'intégrer dans l'ERP. Ce module doit permettre de sauvegarder les horaires de pointages d'entrées et de sorties des employés accompagné de la date de pointage.

#### **Pointage des employés**

Dans ce projet le pointage se fait via la détection des visages des employés. Ainsi, parmi les besoins fonctionnels nous avons celui relatif aux fonctionnalités de pointage. En effet, il faut que l'employé soit en mesure de pointer son entrée et sa sortie.

### **1.2. Stockage des caractéristiques des visages des employés**

Parmi les besoins fonctionnels nous avons aussi le besoins de stocker les caractéristiques des visages des employés afin de pouvoir appliquer la reconnaissance faciale par la suite. Cette fonctionnalité est réservée à un administrateur en la présence de l'employé.

## **2. Besoins non fonctionnels**

Les besoins non fonctionnels de notre application de pointage des employés avec reconnaissance faciale sont formés essentiellement de contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour son bon fonctionnement.

### **2.1.Utilisation de l'ERP Odoo**

En fait, on nous a demandé d'utiliser l'ERP open source et gratuit Odoo. Ceci doit permettre de diminuer les coûts de la solution de pointage. Mais également ce besoin viens de fait que l'entreprise est déjà familiarisée avec cette solution logicielle.

### **2.2.Utilisation d'Android**

L'application de pointage par reconnaissance faciale doit fonctionner sur un système Andoird. En fait, ceci est justifié par le coût faible des dispositifs mobiles qui supportent ce système par rapport à la concurrence. D'où la nécessité de développer en utilisant le SDK d'Android.

### **2.3.Sécurité**

Les données personnels des employés et l'enregistrement de leurs visages dans les bases de données pour pouvoir pointer ensuite avec reconnaissance faciale, doivent être réservés uniquement pour l'administration.

### **2.4.Simplicité et convivialité de l'interface graphique**

Notre application de pointage doit avoir une interface conviviale et simple pour tout type d'utilisateur. En effet, elle va présenter le premier contact des employés avec l'application.

### **2.5.Rapidité d'exécution**

Le temps d'exécution de notre application lors du pointage avec reconnaissance faciale et l'envoi de données vers l'ERP Odoo ne doit pas dépasser 2 secondes.

### 3. Diagrammes de cas d'utilisation

Dans cette partie, on va utiliser les diagrammes de cas d'utilisations pour voir le comportement fonctionnel de notre application de pointage des employés.

Ces diagrammes vont donner une vision globale sur les interfaces de notre système et décrire les interactions entre les acteurs de notre système et notre système de pointage pour décrire nos besoins fonctionnels.

#### 3.1. Diagramme de cas d'utilisation globale

Dans la figure 4, nous présentons le diagramme de cas d'utilisation globale pour la compréhension du fonctionnement du système.

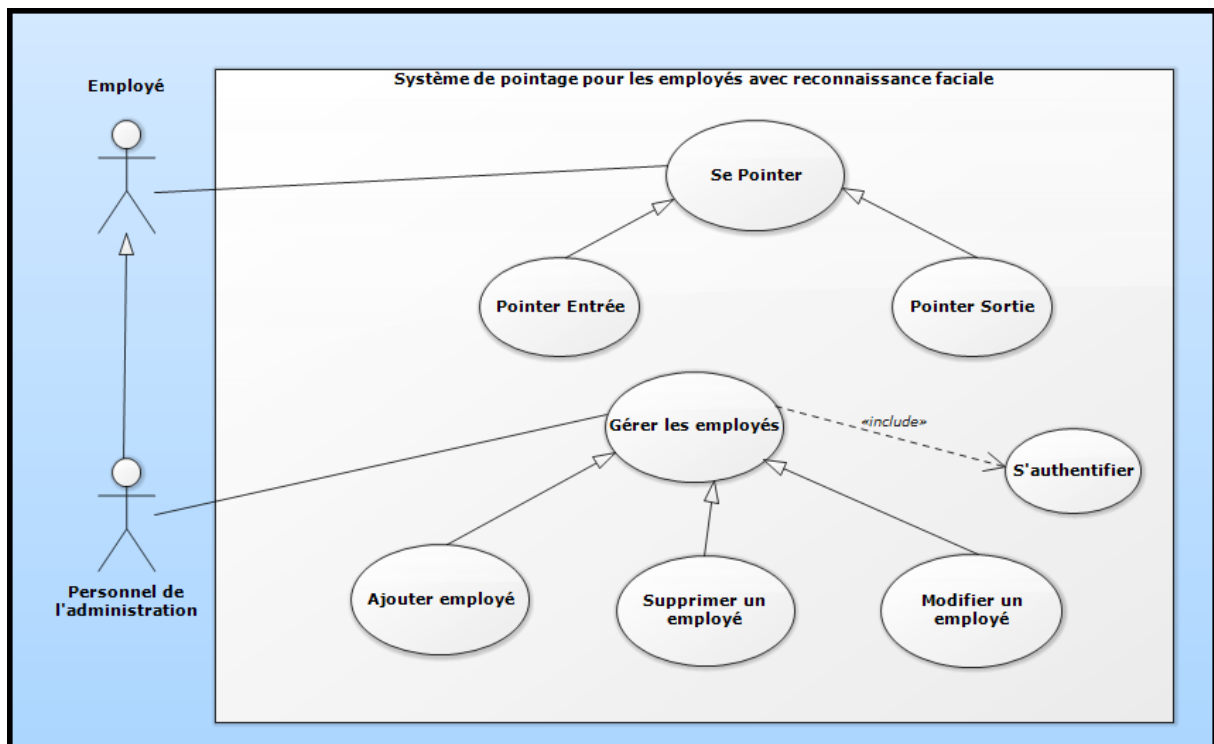


Figure 4: Diagramme de cas d'utilisation globale

### ➤ Identification des acteurs

Les acteurs principaux de notre projet sont : les personnels de l'administration et les employés.

Le personnel de l'administration doit s'authentifier pour gérer les données de ses employés. Il peut ajouter un employé à la société, modifier ces coordonnées ou le supprimer.

Pour les employés et les personnels de l'administration, ils doivent s'identifier avec notre système de reconnaissance faciale pour pointer leurs entrées et leurs sorties dans l'entreprise.

## 3.2. Diagramme de cas d'utilisations détaillés

### 3. 2.1 Diagramme de cas d'utilisation de l'ajout d'un employé

Pour ajouter un employé, une authentification est nécessaire. Après l'authentification, le personnel de l'administration doit effectuer deux tâches (voir figure 5) :

- Ajouter l'image de l'employé en prenant une photo de son visage. Les caractéristiques de la photo seront enregistrées dans la base de données de notre téléphone, pour qu'elles soient utilisées après pendant la phase de reconnaissance faciale.
- Ajouter les informations personnelles concernant l'employé. Ces informations personnelles seront enregistrées dans le module de Ressources humaines de l'ERP Odoo 9 pour être visible pour la partie administrative de l'entreprise.

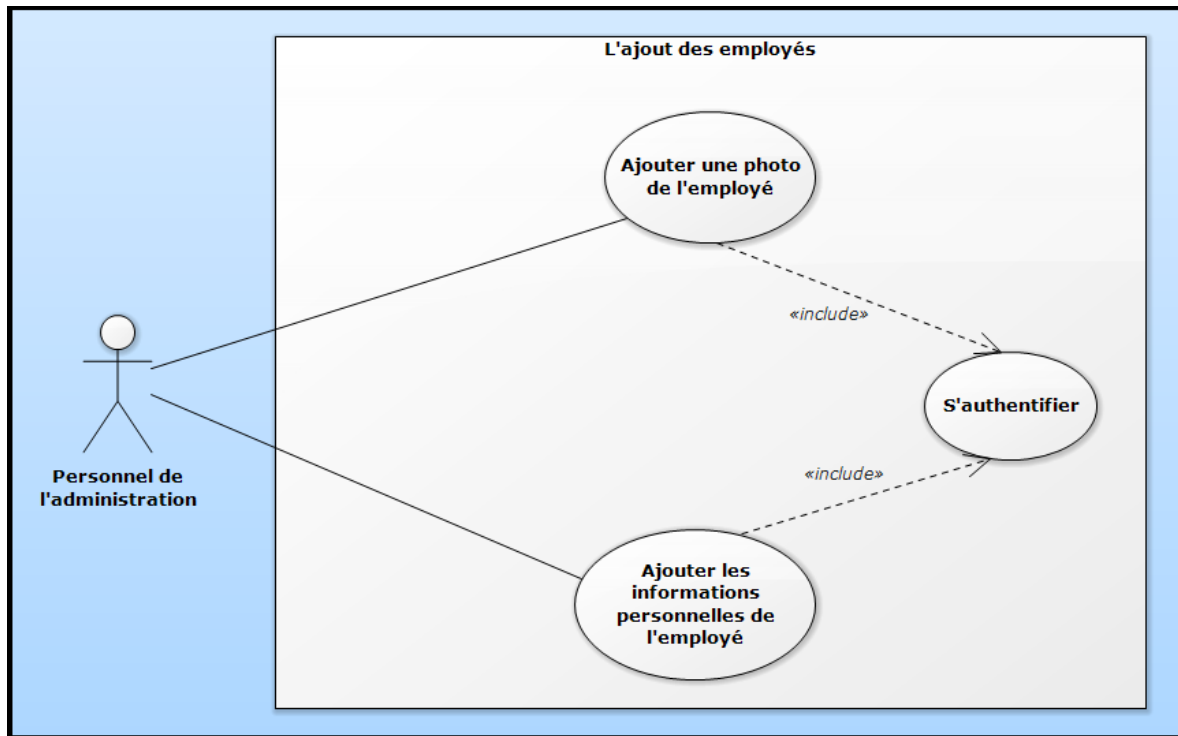


Figure 5: Diagramme de cas d'utilisation d'ajout d'un employé

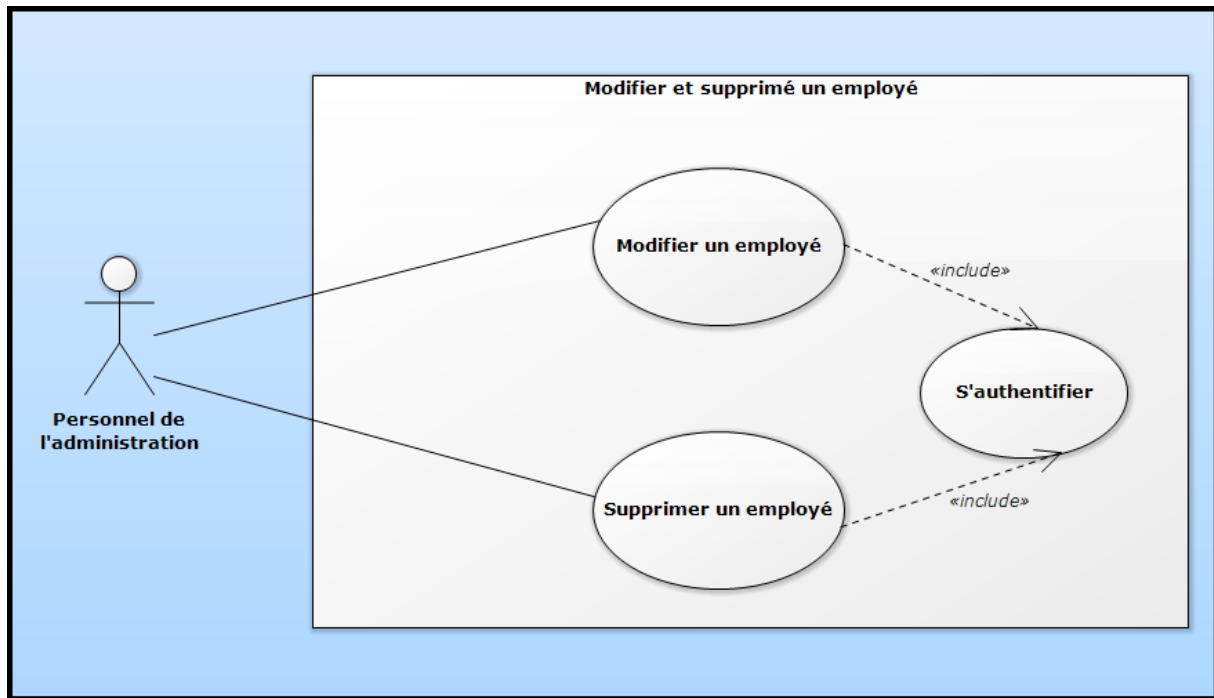
### 3. 2.2 Diagramme de cas d'utilisation pour modifier les coordonnées d'un employé et supprimer un employé

La figure 6 représente le cas d'utilisation pour modifier un employé et le supprimer.

Pour exécuter ces tâches, le personnel de l'administration doit s'authentifier. Ces tâches seront exécutées directement dans le module Ressources Humaines de l'ERP Odoo9.

Les actions sont faciles à exécuter vu la simplicité de l'utilisation de l'ERP Odoo9.

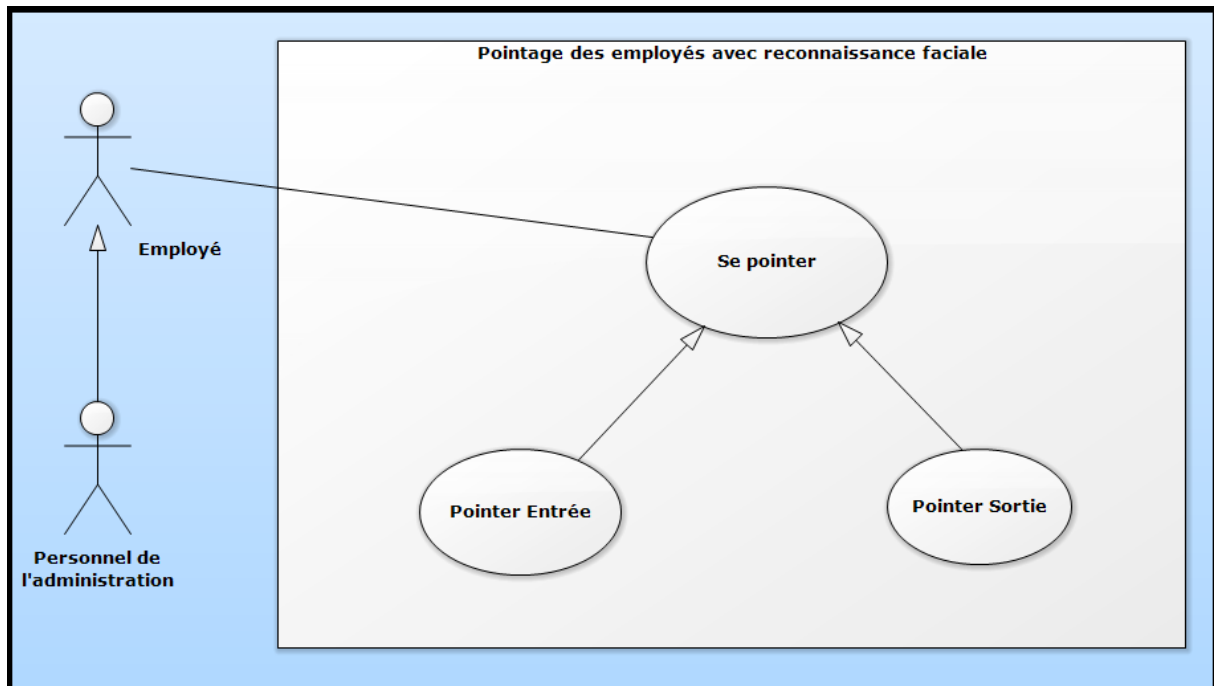
On ne va pas utiliser notre application Android de reconnaissance faciale pour exécuter ces actions, vu que les coordonnées sont directement stockées dans l'ERP lors de la phase de l'ajout des employés.



**Figure 6: Diagramme de cas d'utilisation pour modifier les coordonnées d'un employé et supprimer un employé**

### **3. 2.3 Diagramme de cas d'utilisation pour le pointage des employés avec reconnaissance faciale**

Pour la phase de pointage des employés, l'employé doit s'identifier en utilisant notre application Android de pointage avec reconnaissance faciale. La partie de la reconnaissance faciale permettra aux employés de pointer leurs entrées et sorties (voir figure 7).



**Figure 7: Diagramme de cas d'utilisation pour le pointage des employés avec reconnaissance faciale**

## Conclusion

Ce chapitre nous a permis d'analyser la faisabilité de notre système à travers la détermination de nos fonctionnalités principales. Afin de pouvoir mettre en œuvre notre application de pointage des employés avec reconnaissance faciale, nous devons d'abord présenter notre solution algorithmique pour l'ajout des employés, le fonctionnement de la méthode de détection et de la méthode de reconnaissance de visage que nous avons utilisé et le fonctionnement de la phase de pointage des employés.

Dans le chapitre suivant nous allons présenter notre étude conceptuelle pour parvenir à répondre aux besoins que nous venons d'exprimer.



# Chapitre 4 : Conception

## Introduction

Après avoir tracé les grandes lignes de la phase d'analyse des besoins dans le chapitre précédant, nous allons mettre l'accent maintenant sur la phase de conception de notre système.

Cette phase est très importante dans chaque cycle de vie d'un système informatique car elle va aboutir à la conception et la représentation des diagrammes de séquences et de classe en se basant sur le langage de modélisation UML.

Nous allons se focaliser sur les quatre modules à développer dans le cadre de ce projet, à savoir : Le module de détection de visages, le module d'ajout des employés, le module de reconnaissance de visages et le module de pointages des employés

Dans ce chapitre de conception, nous allons définir ces modules en mettant l'accent sur leurs fonctionnements et en traçons les diagrammes de séquences de chacun module et le digramme de classe.

## 1. Langage UML

### 1.1. Présentation

Nous avons opté pour le choix du langage UML vu qu'il représente beaucoup d'avantages et qu'il nous convient pour la représentation de l'architecture de notre système. En plus, nous avons déjà utilisé ce langage durant nos études d'ingénieurs.

Le langage UML "Unified Modeling Language" ou langage de modélisation unifié, est un langage permettant la modélisation graphique à base de pictogrammes. Il est utilisé pour modéliser une application logicielle dans le cadre de conception orienté objet [19].

Ce langage est l'accomplissement de la fusion de trois de langage de modélisation objet qui sont : Booch, OMT, OOSE. La dernière version est l'UML 2.4.1 lancée en Aout 2012.

On peut appliquer le langage UML utilisé pour d'autres types de systèmes pas nécessairement informatique.

### **1.2. Avantages**

Nous avons choisi le langage UML pour la modélisation des diagrammes de séquences et du diagramme de classe pour notre système de pointage vu qu'il présente plusieurs avantages.

Les avantages du langage UML sont [19] [20] : Premièrement, il est un langage universel et utilisé par la plupart des grandes entreprises. Deuxièmement, il est facile à comprendre et il n'est pas limité au domaine de l'informatique. Enfin, nous utilisons ce langage parce qu'il limite les risques d'erreurs dans notre développement.

## **2. Conception détaillée**

Cette partie représente la conception détaillée de notre projet dans laquelle on va définir le fonctionnement et la conception des quatre principaux modules qui constituent notre application de pointage avec reconnaissance faciale :

- Module de détection de visages
- Module d'ajout des employés
- Module de reconnaissance de visages
- Module de pointages des employés

Pour chaque module, nous allons définir en détails les étapes de son fonctionnement et son diagramme de séquence. Ensuite, nous allons créer le diagramme de classes de notre projet.

### **2.1. Module d'ajout des employés**

Pour réaliser cette tâche d'ajout des employés, c'est à dire pour qu'un employé soit inscrit dans notre entreprise, cela nécessite :

- L'ajout du visage de l'employé dans notre base de données.
- L'ajout de ces informations personnelles dans le module ressources humaines de notre ERP Odoo 9.

### **2.1.1. Ajout du visage de l'employé dans notre base de données**

Dans notre application de pointage, la première chose à faire est de commencer à ajouter nos employés. Ce mécanisme d'ajout des employés, nécessite l'ajout du visage de l'employé en prenant une photo claire de lui avec la caméra frontale de notre Smartphone.

Pour effectuer cette tâche, nous avons besoin du module de détection de visages que nous allons l'expliquer dans la suite de ce chapitre. Ce module va détecter la présence d'un visage humain devant la caméra frontale en appuyant sur le bouton d'ajout de l'employé.

### **2.1.2. Ajout des informations personnelles dans le module ressources humaines de notre ERP Odoo 9**

Après avoir ajouté avec succès le visage d'un employé en lui prenant une photo, nous passons maintenant à ajouter ces informations personnelles tel que : nom et prénom, l'email, numéro de téléphone, poste de travail, département de travail et lieu de travail.

Ces informations seront envoyées vers notre ERP Odoo 9, où est installé notre module ressources humaines qui contiendra tous nos employés inscrits, filtrés par catégorie de département.

Pour cela, nous avons choisi le web service XML-RPC pour effectuer ce transfert de données de notre Smartphone vers notre ERP Odoo 9 via le protocole HTTP.

### **2.1.3. Diagramme de séquence d'ajout d'un employé**

Le fonctionnement du diagramme de séquence de l'ajout d'un employé est le suivant (voir figure 8). Pour ajouter un employé, le personnel de l'administration doit s'authentifier. Après, il commence par ajouter la photo de l'employé qui sera enregistrée dans la base de données de notre application Android. Une fois, la photo est prise, notre application enregistre les valeurs du visage et affiche l'interface d'ajout des informations personnelles de l'employé.

Le personnel de l'administration saisit les informations personnelles de l'employé et valide sa requête. Ces données validées vont être envoyées vers l'ERP Odoo9 et stockées dans le module ressources humaines.

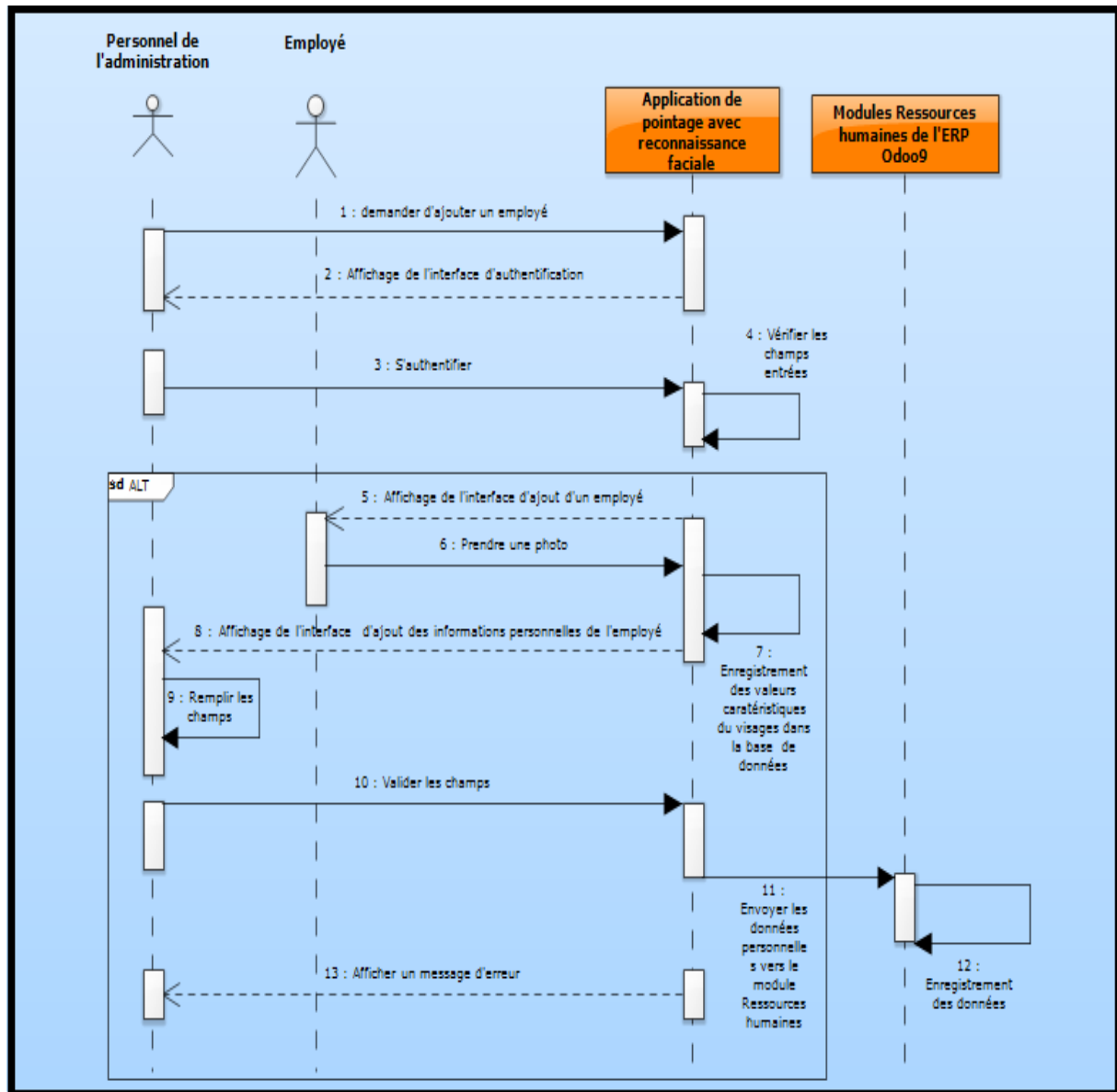


Figure 8: Diagramme de séquence d'ajout d'un employé

## 2.2. Module de détection de visages

Pour réaliser cette phase de détection de visages [10] qui va nous permettre de vérifier la présence d'un visage humain dans une photo prise et de le détecter en enregistrant ses coordonnées, nous avons choisi la méthode de détecteur de visages de "Viola and Jones". Cette méthode est une méthode de détection rapide, fiable et robuste qui bénéficie d'une implémentation dans la bibliothèque Open CV utilisée dans notre application.

La méthode de Viola and Jones [14] est basée sur une étape d'apprentissage et une étape de détection que nous avons défini dans le chapitre 2 de notre projet.

L'étape d'apprentissage que nous avons utilisé va nous permettre de détecter la présence d'un visage dans une photo prise à l'aide des classificateurs.

Cette fonction de classification offerte par l'étape d'apprentissage va nous guider dans la prochaine étape de détection pour réaliser une classification des visages et des non-visages [14].

### **2.2.1 Fonction de classification**

La fonction de classification [10] utilisée pour la détection de visage utilise pour son fonctionnement une cascade de classificateurs qui est basée sur un arbre de décision qui contient plusieurs étapes de création de classificateurs dont le but de détecter les objets en général.

On va utiliser la fonction de classification pour détecter les visages des individus.

Cette Fonction va contenir les descripteurs de Harr comme classificateurs qu'on a définis dans notre deuxième chapitre pour la création des arbres des fonctions de classification.

### **2.2.2 Bloc de décision**

Les paramètres que le bloc de décision va prendre comme entrée en boucle pour son fonctionnement sont : l'image que nous avons pris pour ajouter le visage d'un employé et la fonction de classification défini au-dessous [10]

A chaque étape d'itération, nous allons redimensionner la fenêtre de balayage utilisé pour qu'elle soit compatible avec les descripteurs de Harr utilisés dans la fonction de classification .Le résultat de ce redimensionnement va nous donner une fenêtre de balayage et des descripteurs de Harr appartenant à une même échelle [14].

Cette boucle de bloc de décision va nous garantir la détection de visage dans l'image à toute échelle .Son paramètre est le facteur d'échelle .Sa valeur est entre 1.1 et 1.4.

Donc, notre fenêtre de balayage doit balayer toute l'image pour l'extraction du meilleur facteur d'échelle que nous allons l'utiliser ensuite pour la reconnaissance des visages.

Enfin, pour rendre notre bloc de décision fiable, il faut que nous utilisions un algorithme que nous devons l'exécuter pour chaque itération de fenêtre de balayage utilisé.

### **2.3. Module de reconnaissance de visages**

Pour réaliser la phase de la reconnaissance des visages [10] qui va permettre aux employés de pointer leurs horaires d'entrées et de sortie, nous avons choisi la méthode classique.

Nous pouvons diviser le fonctionnement du module de reconnaissance faciale en deux parties [11]:

Premièrement, nous allons utiliser l'algorithme de "Viola and Jones" que nous avons utilisé lors du module de détection de visages.

Deuxièmement, après le fonctionnement de ce module, nous allons fournir une étape d'identification qui consistera à faire la recherche des paramètres relevés du visage introduit, dans les paramètres des visages des employés existants dans notre base de données.

#### **2.3.1. Utilisation du module de détection de visage**

La première étape pour la reconnaissance faciale est de détecter la présence d'un visage dans la photo numérique que prendra l'employé. Nous avons déjà utilisé l'algorithme de "Viola and Jones " pour le module de détection de visage lors de l'ajout des employés à la société.

Donc, nous allons réutiliser cette méthode pour extraire les valeurs de cette photo que nous utiliserons pour la phase de la reconnaissance faciale, en utilisons les classificateurs de Harr et le seuil de la méthode de AdaBoost [11].

Après, nous allons utiliser ses valeurs dans l'étape suivante d'identification pour identifier un employé.

### **2.3.2. Etape d'identification**

Après avoir extrait les valeurs du visage avec l'algorithme de Viola and Jones. Il s'agit maintenant de comparer les valeurs de ce nouvel visage avec ceux présents dans notre base de données [10] [11].

Si au cours de la recherche, notre système trouve des valeurs identiques de ce nouvel visage parmi ceux présents dans notre base de données, il affichera le nom de l'employé. Sinon, un message s'affiche en indiquant que le visage n'existe pas dans la base de données.

## **2.4. Module de pointage des employés**

### **2.4.1. Fonctionnement**

Pour pointer l'horaire de son entrée et de sa sortie, l'employé doit passer par la phase de la reconnaissance faciale. Si son visage est identifié en tant que employé de cette entreprise, une information qui contiendra l'horaire de son pointage d'entrée et sortie sera envoyée vers notre ERP Odoo 9 et stockée dans le module Pointage des employés.

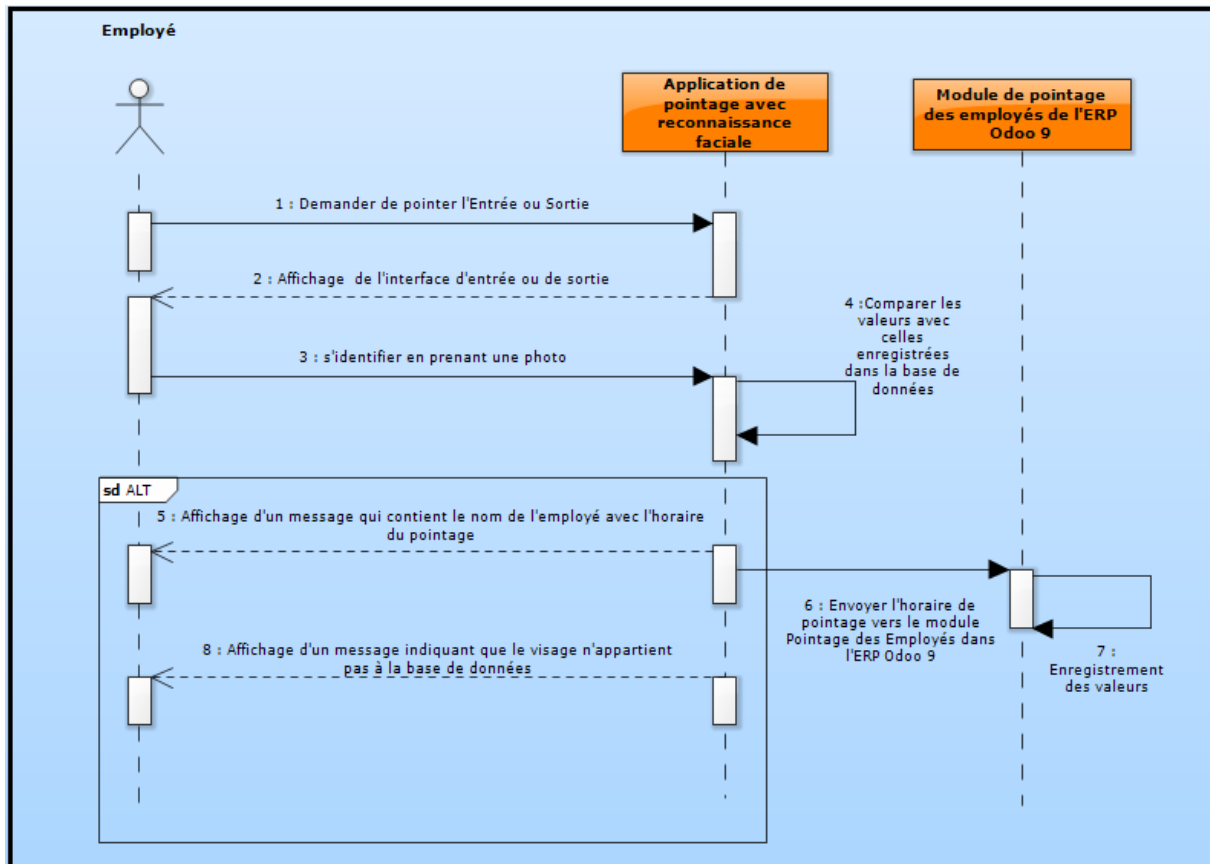
Pour effectuer le pointage, nous avons choisi le web service XML-RPC pour envoyer les données de notre Smartphone vers notre ERP Odoo 9 via le protocole HTTP.

### **2.4.2. Diagramme de séquence de pointage des employés avec reconnaissance faciale**

Le fonctionnement du diagramme de séquence de pointage d'un employé est le suivant (voir figure 9) : Un employé va demander de pointer soit l'entrée, soit sa sortie dans l'entreprise. Notre application Android affichera l'interface de pointage demandé. Après, l'employé va s'identifier en prenant une photo avec le Smartphone. Notre système va comparer les valeurs prises de l'image de cet employé avec les valeurs déjà enregistrées de tous les employés dans sa base de données.

Maintenant, deux cas se présentent : Si notre Système de pointage identifie l'employé, un message indiquant le nom de l'employé va s'afficher avec l'horaire du pointage. Le nom et l'horaire vont être envoyés vers l'ERP Odoo9 et stockés dans le module Pointage des employés.

Si notre système n'identifie pas l'employé, un message sera affiché indiquant que le visage n'appartient pas à la base de données du système.



**Figure 9: Diagramme de séquence de pointage des employés**

## 2.5. Diagramme de classes

Après avoir terminé le traçage des diagrammes de séquences de notre projet, nous passons maintenant à tracer notre diagramme de classe qui est la structure statique de notre système.

Le diagramme de classe va nous permettre d'afficher les objets dans notre système, les relations entre ces derniers, les attributs et les opérations qui caractérisent chaque classe d'objets.

La figure 10 représente le diagramme de classes de notre projet qui contient toutes les classes de notre système ainsi que les différentes relations entre elles.



Les classes Main Activity, Pointer\_Entrée, Pointer\_Sortie, Ajout\_Image, Ajouter\_Coordonnées permettent de gérer les interactions entre le programme et l'utilisateur. Elles offrent les fonctionnalités nécessaires relatives aux fonctionnalités exposées par les interfaces graphiques de notre application.

La classe Main\_Activity est relative à la première interface de notre application. Les classes Pointer\_Entrée et Pointer\_Sortie sont relatives aux interfaces d'entrée et de sorties respectivement. La classe Ajout\_Image est relative à l'interface qui nous permet d'ajouter un visage de l'employé. Et la classe Ajouter\_Coordonnées nous permet d'ajouter les informations personnelles des employés.

La classe Manipuler\_DAO représente la classe qui permet de communiquer avec la base de données qui stocke les données d'authentification. La classe Employé représente la classe qui contient les paramètres et les méthodes utilisées pour la création d'un employé. Et la classe Frametraitements représente les méthodes que nous avons utilisées lors des phases de détection et reconnaissances des visages.

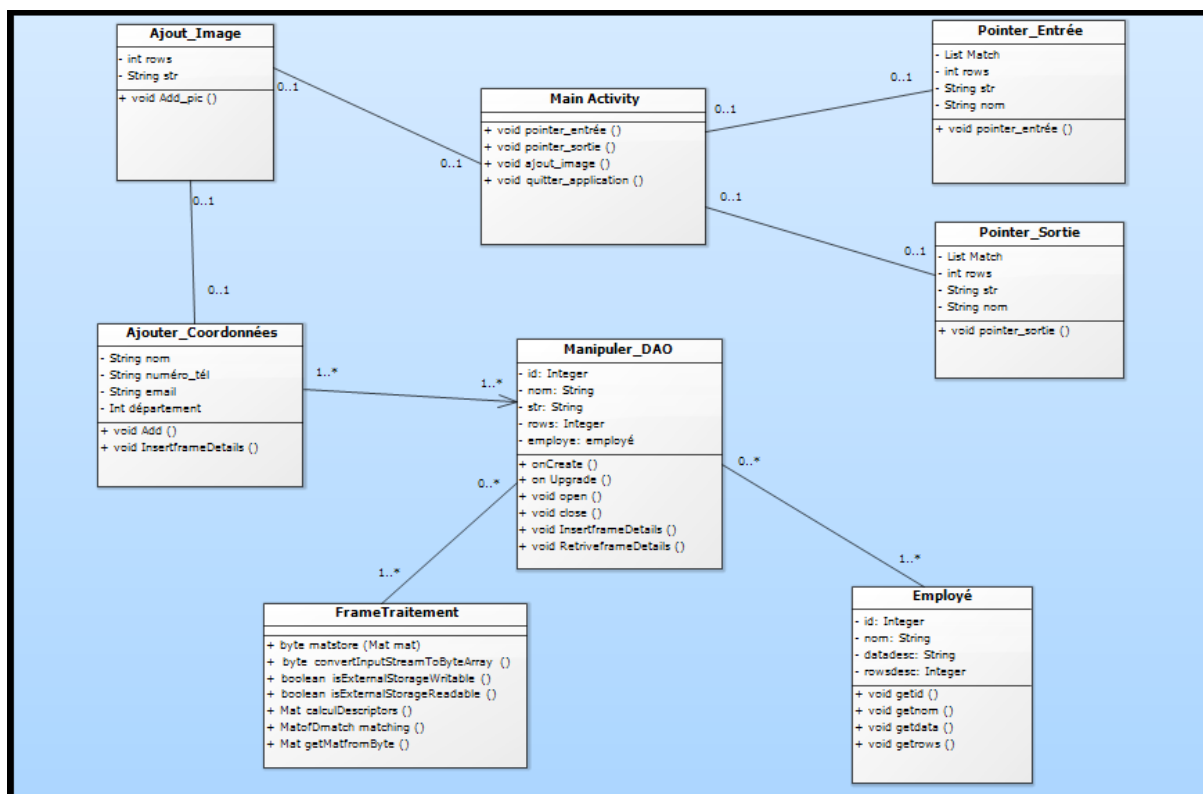


Figure 10: Diagramme de classes

## Conclusion

Dans ce chapitre de conception, nous avons présenté les différents modules et les étapes de conception de notre application de pointage des employés. Nous avons aussi décrit les diagrammes de séquences et le diagramme de classe pour délimiter le cadre de notre travail et pour préparer un terrain favorable pour notre prochaine étape de réalisation.

Dans le chapitre suivant, nous allons nous intéresser à la phase de l'implémentation de notre système de pointage avec reconnaissance faciale en nous basant sur la conception détaillée de ce chapitre.

# Chapitre 5 : Réalisation

## Introduction

Après avoir clôturé les phases d'analyse de besoins et de conception, nous présentons dans ce chapitre les détails de la réalisation de notre application.

Nous commencerons par présenter l'environnement logiciel et matériel dans lesquels notre application Android a été développée en indiquant les technologies et les langages que nous avons utilisés pendant le processus de développement.

Ensuite, nous allons aborder les méthodes que nous avons développées pour l'ajout des employés dans le module ressources humaines de notre ERP Odoo9. Après, nous allons présenter la bibliothèque OpenCV que nous avons utilisé pour les phases de détection et de la reconnaissance des visages. Et comme dernière étape, le pointage des employés avec reconnaissance faciale.

Enfin, nous terminons ce dernier chapitre en effectuant un aperçu sur l'implémentation de notre application et sur les différentes interfaces réalisées qui traduisent le déroulement de notre application de pointage des employés avec reconnaissance faciale.

## 1. Environnement de Travail

Dans cette partie, nous allons présenter l'ensemble des équipements utilisés ainsi que les technologies adoptées qui ont mené à la réalisation de notre projet.

### 1.1. Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé les équipements suivants :

Un ordinateur portable "Asus" avec la configuration suivante :

- Processeur : Intel(R) Core (TM) i7-4710HQ CPU @2.50GHz 2.50GHz

- Mémoire installé (RAM) : 8.00 Go
- Disque Dur : 1000Go
- Système d'exploitation : Windows 8.1 Professionel

Un appareil téléphonique "Samsung J1" ayant les caractéristiques suivantes :

- Processeur : Dual Core, 1200 GHz
- Mémoire Ram : 512 Mo
- Système d'exploitation : Android 4.4.4
- Ecran : 4 ,3 pouces avec 800\*480

## **1.2. Environnement logiciel**

Dans ce paragraphe, nous allons décrire les outils utilisés pour la réalisation de ce projet. Nous avons utilisé la plateforme Android comme plateforme cible en réponse aux besoins non fonctionnels de ce projet. Côté développement, nous avons utilisé Android Studio comme environnement de développement.

Pour établir la communication des données entre notre application mobile et l'ERP Odoo9, nous avons utilisé les bibliothèques xmlrpc client.

Pour les phases de détection et reconnaissance des visages, nous avons utilisé la bibliothèque OpenCV. Nous allons utiliser le API NDK pour compiler les librairies de la bibliothèque de l'OpenCV qui sont développées en langages C/C++ et python.

### **Android Studio**

Pour notre application, nous avons travaillé avec Android Studio 1.5 qui un environnement de développement intégré, libre et universel pour le développement des applications Android. Il est basé sur IntelliJ IDEA.

Pour notre développement avec Android Studio, nous avons utilisé le SDK tools d'une version récente 23.0.0, et notre application cible l'API de niveau 15 et elle est compatible avec les API de niveaux au-delà de celui-ci.

## SDK Android

Le kit de développement (SDK) d'Android [21] contient un ensemble d'outils de développement, accompagné d'un débogueur, des bibliothèques logicielles, une documentation sur son utilisation, des exemples de codes, des tutoriaux et enfin un émulateur utilisé pour tester le fonctionnement des applications développées. On peut l'utiliser sur les plateformes suivantes [22] : Noyau Linux, Mac OS et plateformes Windows XP et ses versions ultérieures.

## Langage Python

Le langage Python est un langage moderne de programmation objet, multi-paradigme et multiplateformes [23]. Ce langage est spécialisé pour la programmation orientée objet et la programmation fonctionnelle.

Le langage python est placé sous une licence libre et fonctionne plusieurs plateformes informatiques, comme Windows, Unix Mac Os, Android, iOS et Java.

Nous avons utilisés le langage python pour développer le module de pointage des employés.

## Bibliothèque OpenCV

La bibliothèque OpenCV (Open Source Computer Vision) [24] est une bibliothèque de traitement d'images et de vision par ordinateur créé par Intel en 1999.

Cette bibliothèque est open source et disponible pour les plateformes suivants : Windows, Linux, Android et MacOS.

Elle comprend des algorithmes et fonctionnalités développer par le Langage C/C++ pour le traitement des images numériques et pour l'analyse de mouvement.

La bibliothèque OpenCV contient les cinq librairies suivantes [25] :

- **Librairie CV** : elle contient les fonctions principales de traitement d'images comme :
  - Détections de mouvement
  - Suivi des Objets

- Conversion des couleurs de l'espace (RGB, HSV)
  - Transformations géométriques (transformée de Fourier, transformée de Hough)
  - Les histogrammes
- **Librairie CVAUX** : Cette librairie a pour fonction de regrouper les fonctions auxiliaires (algorithmes) en assurant le test des fonctions qu'on va utiliser pour le traitement des images comme :
- Suivi des yeux et de la bouche
  - Suivi du modèle 3D
  - Reconnaissance des gestes
  - Utilisation du modèle de Markov 1D et 2D.
- **Librairie CXCORE** : Cette librairie contient les fonctions d'algèbres linéaires comme
- Le calcul des matrices
  - Gestionnaire de mémoire
  - Le dessin de primitives géométriques (rectangles, carrés, lignes, ellipses ....)
- **Librairie HIGHGUI** : Cette librairie contient les fonctions et algorithmes principales pour gérer les entrées-sorties des images et vidéos et les fonctions de création des interfaces graphiques.
- **Librairie ML** (Machine Learning) : Cette librairie contient les fonctions d'apprentissage et de classifications qu'on va utiliser pour la détection et la reconnaissance des visages.

Nous avons utilisé la bibliothèque OpenCV pour les phases de détection de visages et reconnaissance des visages dans notre projet. Grâce à ses librairies, cette bibliothèque nous a permis de déterminer la présence d'un visage lors de la prise d'une photo d'un employé, dans le but de l'enregistrer son visage dans notre base de données.

En utilisant notre système de pointage avec reconnaissance faciale, cet enregistrement permettra de faire la reconnaissance faciale du visage de l'employé pour effectuer son pointage lors de l'entrée et sortie dans sa société [26].

### **Librairie xmlrpc client**

Comme nous l'avons indiqué dans le deuxième chapitre, le xmlrpc est une méthode d'appel de procédure à distance qui utilise le langage xml pour passer ses paramètres via le service HTTP [27].

En utilisant la librairie xmlrpc client, nous allons appeler les méthodes de connexion avec des paramètres sur le serveur où est installé notre ERP Odoo 9 pour stockées et récupérer les données.

### **L'API NDK**

Le NDK "Native Development Kit" est une API du système d'exploitation Android qui est responsable de développer et compiler avec le langage matériel cible [28].

Le NDK est en opposition avec Android SDK qui est une abstraction du code Java, indépendante du matériel.

On va utiliser cette API pour compiler les algorithmes et fonctions des librairies de la bibliothèque OpenCV qu'on a utilisé pour les phases de détection et reconnaissance de visages.

## **2. Application de pointage avec reconnaissance faciale**

La conception des interfaces de l'application "pointage des employés avec reconnaissance faciale" est une étape très importante puisque qu'elle représente l'interaction entre les utilisateurs et notre application. Donc, les interfaces doivent être faciles à comprendre et à utiliser.

Dans cette partie, on va guider les utilisateurs avec des notifications et des messages d'erreurs pour faciliter l'utilisation de notre système de pointage.

## 2.1. Développement du module de pointage

Nous avons développé un module de pointage pour les employés pour stocker les horaires de pointage. Ce module a été développé en langage Python. Nous avons fait la synchronisation de ce module avec le module ressources humaines pour simplifier l'usage de ce module dans l'ERP Odoo9.

Le module de pointage développé contiendra les champs suivants : Le nom de l'employé, l'horaire de pointage d'entrée et de sortie et la date du jour.

Après le développement de ce module, nous l'avons intégré et installé dans l'ERP Odoo9 [29].

La figure 11 représente le module pointage des employés que nous avons développés.



Figure 11: Interface du module pointage des employés

## 2.2. Développement de l'application Android

### Interface utilisateur

Dans la première interface de notre application, on trouve deux boutons de pointage. Le premier est pour pointer l'entrée et le deuxième pour pointer la sortie. Ces deux boutons nous permettront d'effectuer le pointage des employés en utilisant la reconnaissance faciale pour les identifier.

Ensuite, on trouvera une montre digitale et analogique pour afficher le temps actuel de la journée. Enfin, la présence de deux autres boutons à l'extrémité de l'interface responsable de



l'ajout des employés et pour quitter l'application, nécessite une authentification pour les réaliser.

La figure 12 représente la première interface de notre application.

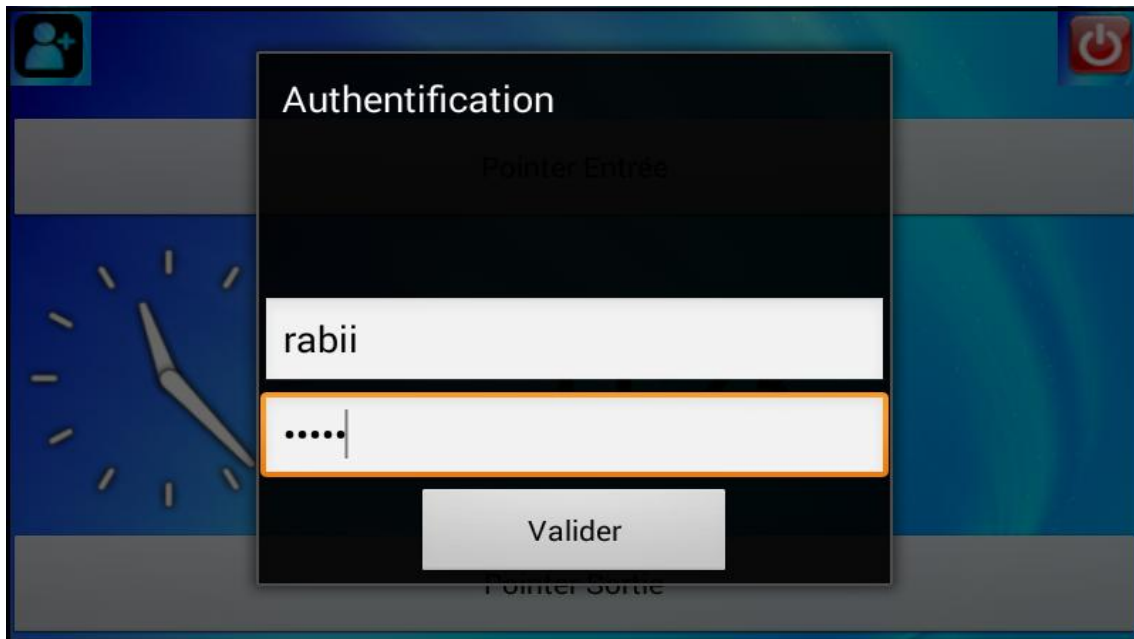


**Figure 12: Première interface de notre application**

### **Interface de l'authentification**

Pour ajouter un employé à notre base de données de la société ou pour quitter notre application de pointage, le personnel de l'administration doit s'authentifier (voir figure 13).

Comme toute application, la sécurité d'accès à ces deux tâches est nécessaire. La figure ci-après représente l'interface à travers laquelle le personnel de l'administration s'identifie. Il saisit son login et son mot de passe pour s'authentifier.



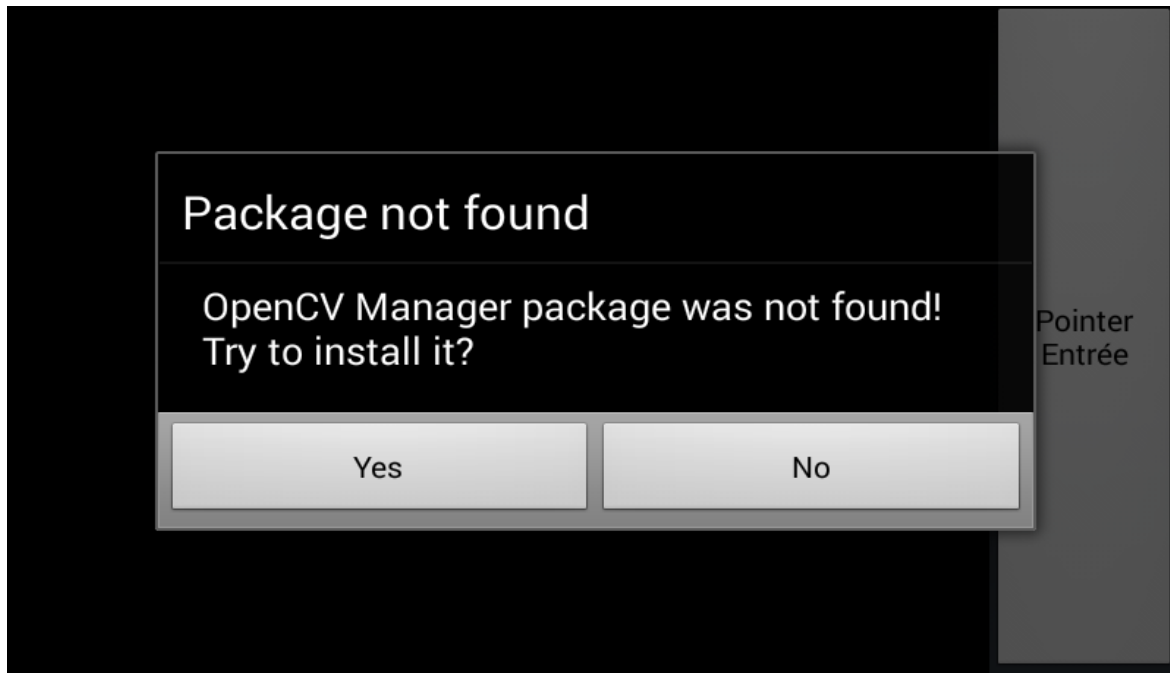
**Figure 13: Interface d'authentification**

### **Interface de l'ajout d'un nouvel employé**

Pour ajouter un employé à notre société, nous devons effectuer deux tâches. Premièrement, on doit ajouter son visage à notre base de données du téléphone pour l'utiliser après lors de son pointage.

Cette tâche nécessite l'installation de la bibliothèque OpenCV dans notre Smartphone pour pouvoir utiliser les librairies de l'OpenCV pour la détection du visage.

Notre application va faire la recherche du package de l'OpenCV dans le Smartphone. S'il n'est pas installé, une petite fenêtre va s'afficher qui nous demande d'installer le package. (Voir figure 14).



**Figure 14: Interface d'installation de l'open CV**

Une fois le package est installé sur notre Smartphone, nous n'avons plus besoin de le réinstaller à chaque ajout d'un nouvel employé.

Maintenant, nous passons à prendre une photo du visage de l'employé. Une fois la photo est prise, on aura un petit message "visage enregistré avec succès " qui nous informe que le visage de l'employé est bien enregistré dans notre base de données (voir figure 15).




**Figure 15: Enregistrement du visage de l'employé**

Deuxièmement, après l'ajout du visage de l'employé, nous passons à ajouter ses informations personnelles qui seront stockées dans le module Ressources Humaines de l'ERP Odoo 9.

La figure 16 représente l'interface qu'on doit utiliser pour ajouter les coordonnées des employés.

Les informations personnelles contiennent le nom, l'email, numéro du téléphone, département dans lequel il travaille et le lieu de travail.

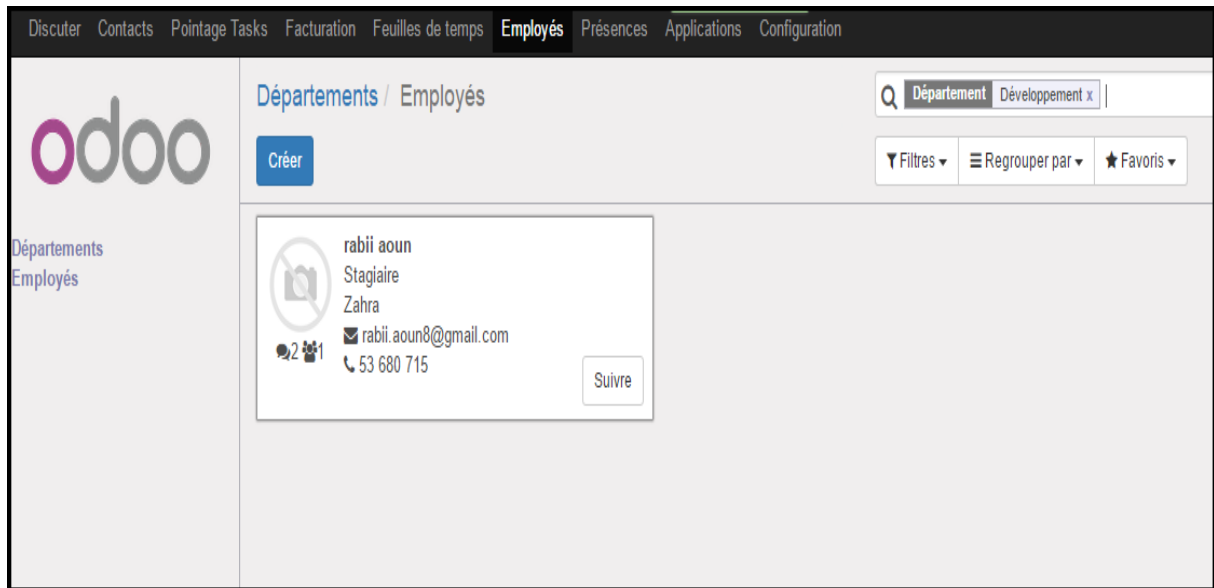


The image shows a web-based form for adding employee personal information. The form has a blue background with white text labels on the left and input fields on the right. The labels are: 'Nom', 'Email', 'Téléphone', 'Département', and 'Lieu du travail'. The input fields contain the following data: 'rabi aoun' for the name, 'rabi.aoun8@gmail.com' for the email, '53680715' for the phone number, 'Stagiare' for the department (with a dropdown arrow), and 'Zahra' for the workplace. A grey button labeled 'Enregistrer' is located to the right of the phone number field. The 'Lieu du travail' field is highlighted with an orange border.

Label	Value
Nom	rabi aoun
Email	rabi.aoun8@gmail.com
Téléphone	53680715
Département	Stagiare
Lieu du travail	Zahra

**Figure 16: Interface d'ajout des informations personnelles de l'employé**

Après l'ajout des informations personnelles de l'employé, on valide notre requête. La figure 17 représente l'employé ajouté, enregistré dans le module Ressources Humaines de l'ERP Odoo9 accompagné des informations que nous avons saisies.



**Figure 17: Interface d'ajout d'un employé au niveau du module ressources humaines de l'ERP Odoo9**

### Interface de pointage pour les employés

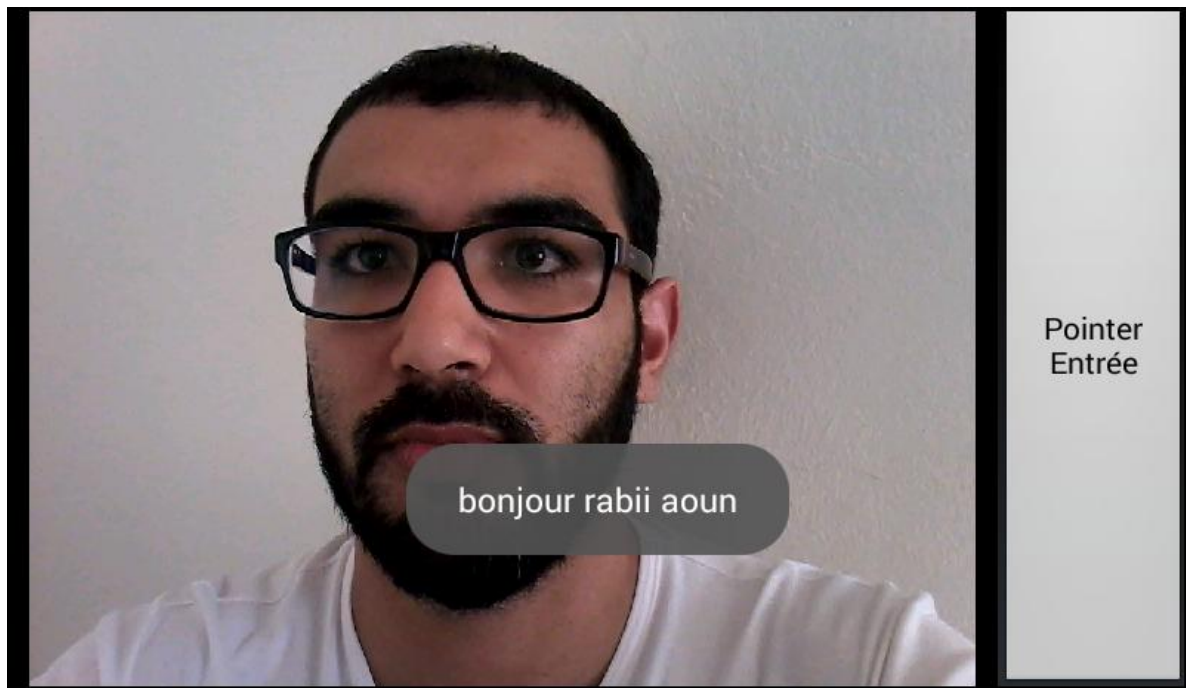
Pour pointer son entrée et sortie à l'entreprise, l'employé doit s'identifier en prenant une photo en utilisant notre système de reconnaissance faciale.

Deux cas se présentent lors du pointage.

Premièrement, si notre système de pointage identifie ce visage en tant que employé déjà enregistré dans notre base de données, on va afficher un message contenant son nom et l'action du pointage accompagné de l'horaire.

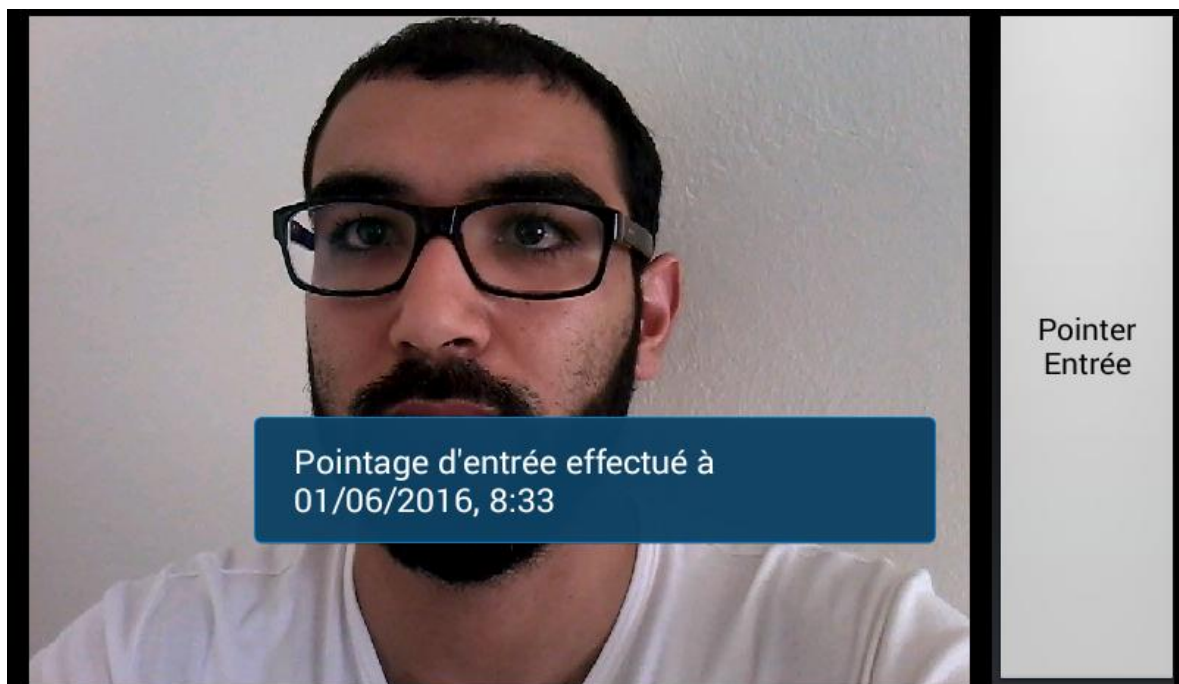
Deuxièmement, si notre système de pointage n'identifie pas le visage de l'employé. Il va afficher un message indiquant que le visage est non identifié.

La figure 18 représente le cas où notre système a identifié l'employé lors de son pointage d'entrée. Un message s'affiche en indiquant le nom de l'employé.



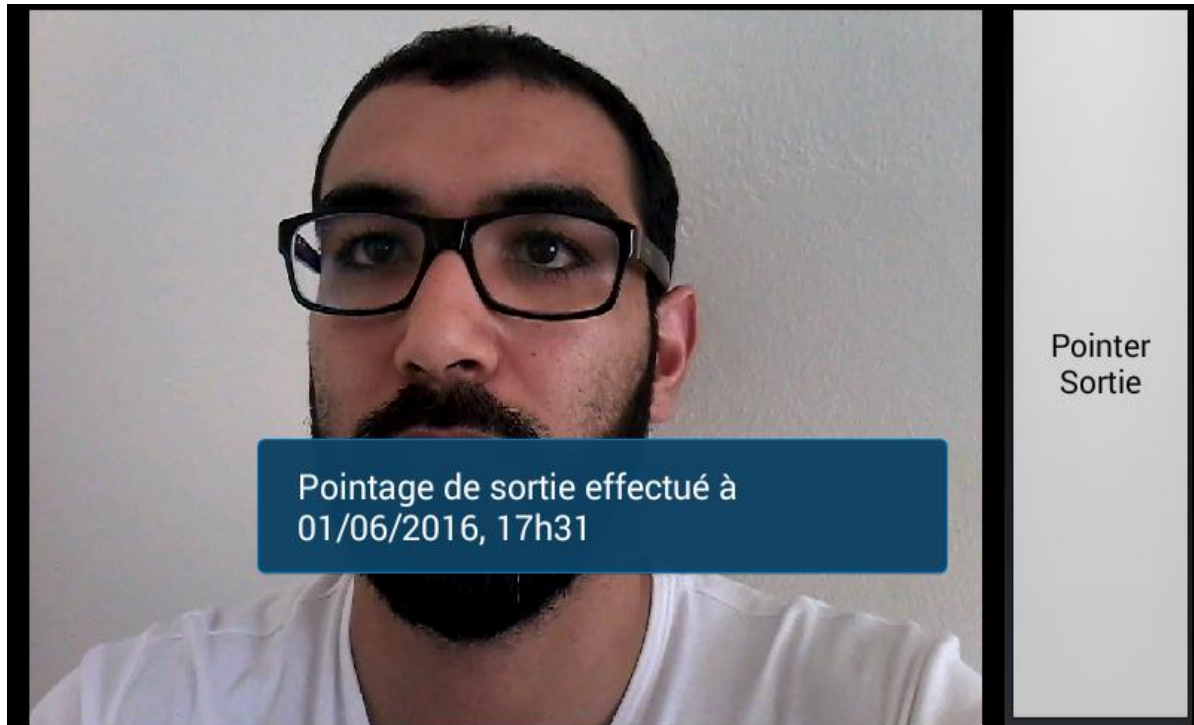
**Figure 18: Première interface de pointage d'entrée**

La figure 19 représente le deuxième message qui va s'afficher lorsque notre système de pointage a identifié l'employé. Le message contient la nature du pointage avec l'horaire.



**Figure 19: Deuxième interface de pointage d'entrée**

Pour le pointage de sortie, la figure 20 représente le deuxième message qui va s'afficher lorsque notre système de pointage a identifié l'employé. Le message contient la nature du pointage avec l'horaire.



**Figure 20: Deuxième interface de pointage de sortie**

La figure 21 représente l'interface du module pointage des employés intégré dans l'ERP Odoo9. Ce module contient le nom de l'employé accompagné de l'horaire de pointage d'entrée et sortie avec la date du jour.



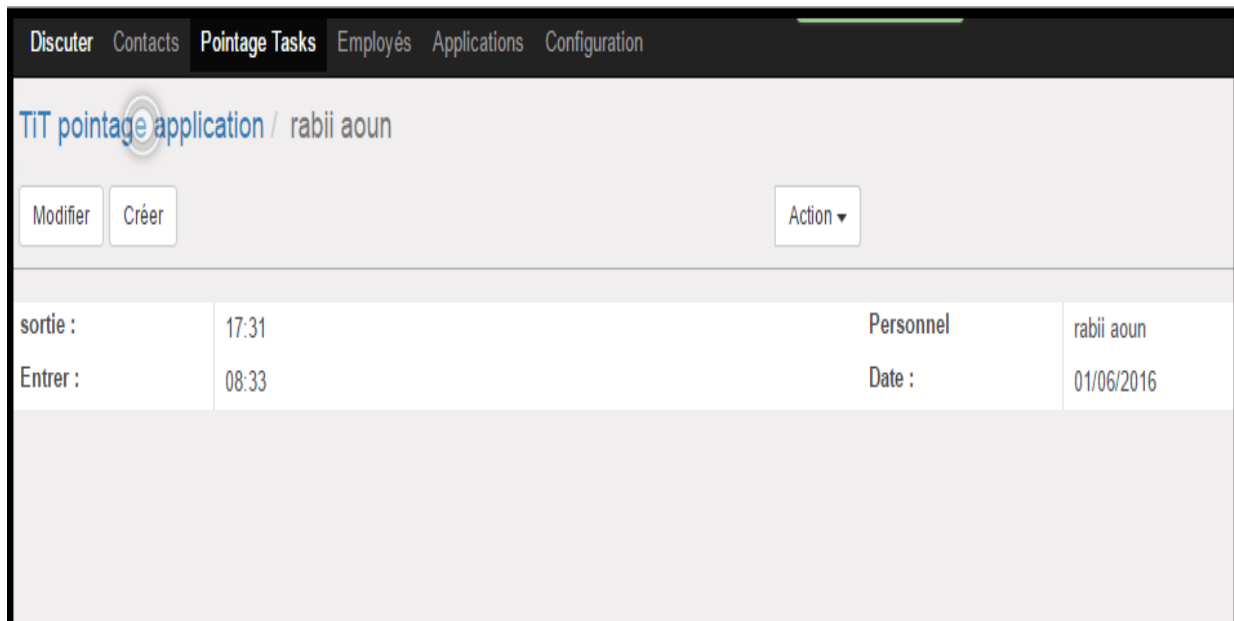


Figure 21: Interface de pointage au niveau du module pointage des employés dans l'ERP Odoo9

Maintenant, Si notre système n'identifie pas le visage de l'employé. Il affichera un message indiquant que le visage introduit n'est pas identifié (voir figure 22).



Figure 22: Interface lorsque le visage n'est pas identifié



La figure 23 nous montre que notre système est fiable. Notre système de pointage identifie le visage de l'employé même lorsqu'il change ses vêtements et modifie quelques aspects extérieurs lors de l'identification.



Figure 23: Interface en modifiant quelques aspects extérieurs

### 3. Chronogramme du projet

Le chronogramme de la figure 24 présente les différentes étapes que nous avons suivies afin de bien réussir notre projet de fin d'études.

	Février				Mars				Avril				Mai				Juin			
Spécification																				
Documentation																				
Conception																				
Implémentation																				
Test																				
Rédaction du rapport																				

Figure 24 : Chronogramme d'activités

## Conclusion

Dans ce dernier chapitre, nous avons présenté le modèle d'implantation de notre système. En premier lieu, nous avons défini et décrit les outils, les librairies, les API que nous avons utilisé. Ensuite, nous avons défini la bibliothèque OpenCV que nous avons utilisé pour réaliser se projet. Enfin, nous avons abordé les aspects fonctionnels de notre système en présentant toutes les interfaces de notre application.

## Conclusion générale

L'identification et la reconnaissance d'un individu reste toujours un problème complexe et difficile à résoudre de manière parfaite. Plusieurs problèmes incombent à cette tâche d'identification et chacun d'eux est non trivial. Pour un système à base de reconnaissance faciale, il existe plusieurs conditions qui affectent sa fiabilité. Cependant, l'étape de détection des visages a une grande influence sur la performance du système surtout dans le module d'identification.

Dans notre projet, nous avons mis en œuvre une approche de détection du visage basée sur un algorithme très fiable qui est l'algorithme de Viola and Jones. Après, en réutilisant la méthode de détection de visages, nous avons pu aborder la phase de reconnaissance faciale qui nous permettra d'identifier les visages des individus enregistrés dans notre base de données.

Vu la fiabilité de ces méthodes, nous les avons utilisées pour fournir un système de pointage avec reconnaissance faciale pour les employés. Notre système utilisera la méthode de détection de visage pour ajouter un nouvel employé dans notre base de données et la méthode de reconnaissance faciale pour l'identifier lors de son pointage. Nous avons utilisé l'ERP Odoo 9 pour l'ajout de nos employés et le stockage des données de pointage issu de notre système.

Ce projet était une occasion pour confronter les difficultés et le stress engendré par les défis de réalisation d'un projet informatique dans un contexte professionnel, avec un cahier de charge exigeant, des contraintes techniques fortes et avec un temps de réalisation limité. C'était également une occasion de plus pour se familiariser avec le travail dans le cadre d'une entreprise professionnelle et exigeante. Et donc c'était une opportunité pour se préparer au marché de l'emploi en améliorant nos capacités de communiquer et d'interagir avec notre environnement.

Techniquement, ce projet nous a permis de maîtriser un nombre conséquent de technologies, de techniques et de savoirs dans plusieurs domaines. En particulier, c'était une occasion pour s'investir dans le développement mobile, le développement en python de modules d'ERP et leur intégration, la manipulation de bibliothèques, d'algorithmes et d'approches relatives à la reconnaissance faciale et la mise en pratique de nos connaissances et compétences acquises durant nos études d'ingénieur en informatique au sein de l'ENSIT.

Ce projet peut connaître des extensions et des améliorations. En particulier, il sera intéressant de proposer des implémentations pour d'autres systèmes d'exploitation mobiles. La piste de la programmation mobile hybride semble intéressante à ce niveau. Egalement, il est intéressant de faire un couplage faible entre l'application mobile et l'ERP. Ainsi, le client ne sera pas obligé d'utiliser un ERP spécifique qui est dans notre cas Odoo 9.

Enfin, ce projet nous a été une excellente expérience à l'échelle personnelle, professionnelle et scientifique. Il représente pour nous une excellente transition du monde académique au monde professionnel.

## Références

- [1] <http://www.biometrie-online.net/> [Consulté le 17/02/2016]
- [2] <http://www.linternaute.com/science/biologie/dossiers/06/0607-biometrie/visage.shtml>  
[Consulté le 17/02/2016]
- [3] <http://www.entreprise-erp.com/articles/definition-erp.html> [Consulté le 20/02/2016]
- [4] <https://www.choisirmonerp.com/ERP-Definition-Perimetre.php> [Consulté le 20/02/2016]
- [5] / <http://www.xcg-consulting.fr/news/2014/10/20/odoo-lerp-adapte-lentreprise/> [Consulté le 20/02/2016]
- [6] [https://www.odoo.com/fr\\_FR/](https://www.odoo.com/fr_FR/) [Consulté le 22/02/2016]
- [7] <https://6it.fr/blog/actualites-et-tendances-1/post/faut-il-passer-a-odoo-9-33> [Consulté le 22/02/2016]
- [8] <http://www.xul.fr/ajax-format-json.html> [Consulté le 29/02/2016]
- [9] <http://www.xml.com/pub/r/1201> [Consulté le 29/02/2016]
- [10] <http://www.biblio.univ-evry.fr/theses/2008/2008EVRY0032.pdf> [Consulté le 10/03/2016]
- [11] <http://thesis.univ-biskra.dz/944/4/Chap%201%20Syst%C3%A8me%20RV%20sept%2012.pdf> [Consulté le 15/03/2016]
- [12] <https://www.uclouvain.be/cps/ucl/doc/commission-gbio/documents/Thesis2.pdf>  
[Consulté le 21/03/2016]
- [13] [https://fr.wikipedia.org/wiki/Caract%C3%A9ristiques\\_pseudo-Haar](https://fr.wikipedia.org/wiki/Caract%C3%A9ristiques_pseudo-Haar) [Consulté le 17/03/2016]
- [14] <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-IJCV-01.pdf> [Consulté le 23/03/2016]

- [15] [https://library.csc.dz/media/thesis/2014-12/20141218110807-OUDJANI\\_memoire.pdf](https://library.csc.dz/media/thesis/2014-12/20141218110807-OUDJANI_memoire.pdf)  
[Consulté le 24/03/2016]
- [16] <http://www.face-rec.org/algorithms/ebgm/wisfelkrue99-facerecognition-jainbook.pdf>  
[Consulté le 28/03/2016]
- [17] [http://www.ummto.dz/IMG/pdf/detection\\_et\\_identification\\_d\\_individus\\_par\\_methode\\_biometrique.pdf](http://www.ummto.dz/IMG/pdf/detection_et_identification_d_individus_par_methode_biometrique.pdf) [Consulté le 4/04/2016]
- [18] [https://fr.wikipedia.org/wiki/Mod%C3%A8le\\_en\\_spirale](https://fr.wikipedia.org/wiki/Mod%C3%A8le_en_spirale) [Consulté le 10/05/2016]
- [19] <http://www.additeam.com/SSII/uml/> [Consulté le 11/05/2016]
- [20] <http://laurent-piechocki.developpez.com/uml/tutoriel/lp/cours/> [Consulté le 11/05/2016]
- [21] [https://fr.wikipedia.org/wiki/Android\\_SDK](https://fr.wikipedia.org/wiki/Android_SDK) [Consulté le 13/05/2016]
- [22] <https://developer.android.com/studio/intro/index.html> [Consulté le 13/05/2016]
- [23] [https://fr.wikipedia.org/wiki/Python\\_\(langage\)](https://fr.wikipedia.org/wiki/Python_(langage)) [Consulté le 13/05/2016]
- [24] <https://fr.wikipedia.org/wiki/OpenCV> [Consulté le 15/03/2016]
- [25] <https://sourceforge.net/projects/opencvlibrary/> [Consulté le 15/03/2016]
- [26] <http://blog.codeonion.com/2015/11/25/creating-a-new-opencv-project-in-android-studio/>  
[Consulté le 16/03/2016]
- [27] <https://docs.python.org/3.1/library/xmlrpc.client.html> [Consulté le 29/02/2016]
- [28] <https://developer.android.com/ndk/index.html?hl=ru> [Consulté le 16/03/2016]
- [29] [https://www.odoo.com/fr\\_FR/forum/aide-1/question/how-do-you-install-a-module-578](https://www.odoo.com/fr_FR/forum/aide-1/question/how-do-you-install-a-module-578)  
[Consulté le 25/02/2016]

## Liste des Acronymes

**API**    Application Programming Interface

**ERP**    Entreprise Ressource Planning

**HTTP**   HyperText Transfert Protocol

**JSON**   JavaScript Object Notation

**NDK**    Native Development Kit

**OpenCV**   Open Computer Vision

**SDK**    Software Development Kit

**UML**    Unified Modeling Language

**XML**    eXtensible Markup Language

**XML-RPC**   eXtensible Markup Language-Remote Procedure Call

# Annexe 1

## Système d'exploitation Android

Android est un système d'exploitation mobile basé sur le noyau Linux. Il a été développé par Google en 2007 spécifiquement pour l'utiliser comme système d'exploitation pour les Smartphone et tablettes. Mais, il s'est diversifié et a touché beaucoup d'autres objets, comme télévisions (Android TV), les Smart Watch (Android Wear) et les Voitures (Android Auto).

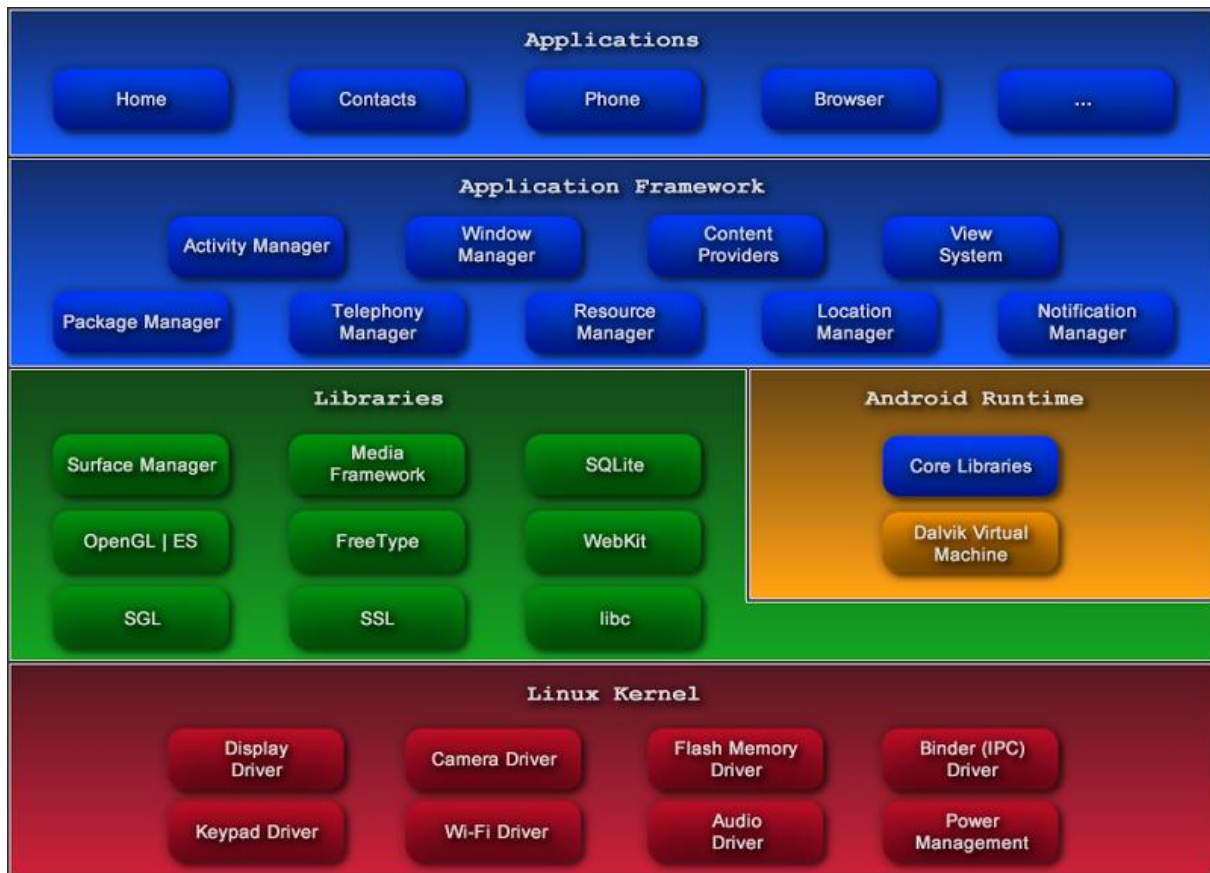
La plateforme Android est entièrement gratuite et sous licence open source .Elle comporte une grande communauté de développeurs qui développent des applications Android dans tous les domaines pour étendre les fonctionnalités de ce système d'exploitation.

### 1. Plate-forme Android

Les différentes couches de la plate-forme Android sont :

- Noyau Linux : il présente la couche de plus bas niveau. Il est basé sur un noyau linux de version 2.6.24, il assure la communication entre la couche logicielle et matérielle conférant ainsi des caractéristiques multitâches.
- Des bibliothèques : Cette couche contient plusieurs bibliothèques écrites avec les langages C/C++. Elle couvre une grande portée comme les bibliothèques graphiques et multimédias.
- Moteur d'exécution Android : Cette couche contient une machine virtuelle Java.
- Applications : Dans cette couche, on trouve toutes les applications d'un téléphone mobile comme le navigateur web, le calendrier, etc.
- Framework Applicatif : Dans cette couche, on trouve les fonctionnalités de la téléphonie ainsi que la gestion de contenu.





**Figure 25: A.1 - Architecture du Système d'exploitation Android**

## 2. Application Android

Chaque application Android est composée d'un ensemble de quatre éléments qui interagissent entre eux.

Ces quatre éléments sont : "Activité ", "Service", "Broadcast Reciver" et "Content Provider".

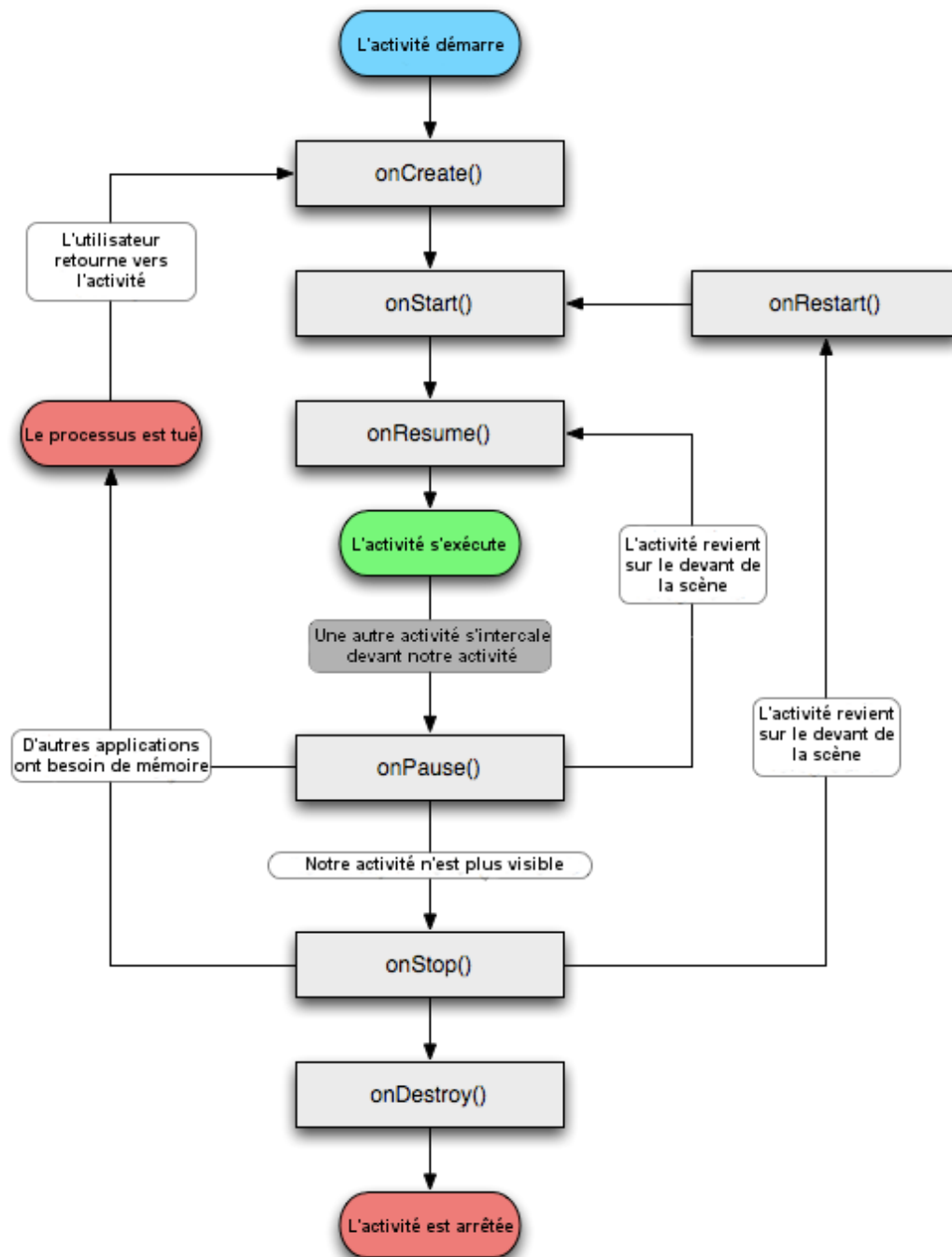
## 3. Cycle de vie d'une application Android

Lors du démarrage de chaque activité, trois méthodes sont appelées automatiquement : "OnCreate", "OnStart", "OnResume".

- **OnCreate** : Cette méthode est appelée lorsqu'on crée une nouvelle activité.
- **OnStart** : Cette méthode est appelé après la méthode OnCreate, mais avant que l'activité soit visible dans l'écran. Une fois la méthode OnStart est terminée
- **OnResume** : Cette méthode est appelée juste après la méthode OnStart si elle est l'activité du premier plan sur l'écran. Maintenant, l'activité est en cours d'exécution et d'interaction avec l'utilisateur. Cette méthode est aussi appelé, si l'activité cède l'avant-plan à une autre activité et que l'activité existe encore.

Lorsqu'on quitte une activité, trois méthodes sont appelées automatiquement : "onPause", "OnStop" et "OnDestroy".

- **OnPause** : Cette méthode est appelée lorsque notre système d'exploitation Android est sur le point d'exécuter une autre activité. Dans ce cas, l'activité n'aura plus accès à l'écran.
- **OnStop** : Cette méthode est appelée lorsque l'activité n'est plus visible, soit parce qu'une activité a pris le premier plan à sa place ou parce que l'activité est détruite.
- **OnDestroy** : Cette méthode donne une dernière chance à l'application de terminer son traitement avant qu'elle ne soit pas détruite.



**Figure 26: A.2 - Cycle de vie d'une application Android**

## **Annexe 2**

### **Méthode d'Adaboost**

#### **1. Définition**

La méthode d'Adaboost est une méthode de boosting, elle est composée par une partie d'intelligence artificielle et d'une partie d'apprentissage automatique.

Elle est introduite par Yoav Freund et Robert Schapire.

Cette méthode permet de combiner plusieurs hypothèses simples pour en créer une hypothèse plus performante.

#### **2. Principe**

La méthode d'Adaboost repose sur une sélection itérative de classifieur faible en fonction d'une distribution des exemples d'apprentissages. Chacun de ces classifieurs est pondéré en fonction de sa difficulté avec le classifieur courant.

## Résumé

Dans le cadre de ce projet, nous avons développé un système de pointage par reconnaissance faciale. Ce système comporte une application Android comme outil de pointage pour les employés. En plus, le système comporte un ERP qui permet à l'administration de suivre l'état de pointage de ses employés. Nous avons utilisé la bibliothèque OpenCV pour gérer les traitements de reconnaissance de visage au niveau de l'application Android. De plus, nous avons utilisé l'ERP Odoo 9 tout en l'enrichissant avec un module ad-hoc pour la synchronisation avec l'application mobile.

**Mots clés :** Système de pointage, reconnaissance faciale, Développement Android, OpenCV.

## Abstract

In this project, we developed a scoring system using facial recognition. This system includes an Android application as a pointing tool for employees. In addition, the system includes an ERP which allows the administration to monitor the status of its employees.

We used the OpenCV library to handle the phase of the face recognition process in the Android application. In addition, we used the ERP Odoo 9 while enriching it with an ad-hoc module for the synchronization with the mobile application.

**Key words :** Scoring system, facial recognition, Android development, OpenCV.

## الملخص

في هذا المشروع قمنا بتطوير نظام التسجيل بإستعمال التعرف على الوجه كأداة. و يشمل هذا النظام إستعمال تطبيق الأندرويد كأداة لنظام التسجيل للموظفين. وبالإضافة إلى ذلك، يتضمن هذا النظام إستعمال مخطط لموارد المؤسسات الذي يسمح للإدارة بمراقبة موظفيها عند تسجيلهم.

استخدمنا مكتبة برمجية مفتوحة للرؤية الحاسوبية لمعالجة الوجه في تطبيق الأندرويد. وبالإضافة إلى ذلك، استخدمنا مخطط موارد المؤسسات Odoo 9 مع إثراء وحدة مخصصة للترامن مع تطبيقات الهاتف المتحرك.

**الكلمات المفاتيح :** نظام التسجيل، التعرف على الوجه، تطبيق الأندرويد، مكتبة برمجية مفتوحة للرؤية الحاسوبية