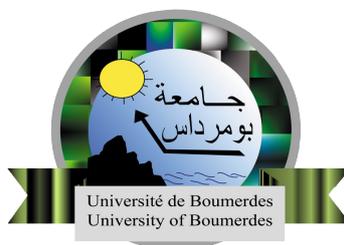


RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITÉ M'HAMED BOUGARA BOUMERDES



FACULTÉ DES SCIENCES DE L'INGÉNIEUR (FSI)  
FILÈRE : GÉNIE ÉLECTRIQUE  
DÉPARTEMENT : INGÉNIERIE DES SYSTÈMES ELECTRIQUES

EN VUE DE L'OBTENTION DU DIPLÔME DE MASTER  
SPÉCIALITÉ : RÉSEAUX ET TÉLÉCOMMUNICATIONS.

---

## Conception et Réalisation d'une Application Android qui transforme une pièce d'identité en format exploitable

---

SOUMIA SACI

*Mémoire présenté pour l'obtention du diplôme Master en  
Réseaux et télécommunications*

*Soutenue le 26/06/2018 en présence de :*

Dr.	MOHAMED AMINE RIAHLA	PRÉSIDENT	UMBB
Dr.	YASSINE MERAIHI	EXAMINATEUR	UMBB
Mme.	SAMIRA MECHID	PROMOTRICE	UMBB

*Année universitaire 2017-2018*

# Résumé

Ce mémoire présente une conception et réalisation d'une application mobile Android qui permet de prendre une photo d'une carte d'identité, un permis de conduire ou un passeport algérien et récupérer les informations fournies par la pièce d'identité telles que le nom et le prénom . . .et les rendre dans un format modifiable et exploitable afin d'accélérer le traitement des dossiers ,le système de pointage dans les sociétés. Et dans plusieurs contextes. L'application sera réalisée avec le langage de programmation Java et Android en explorant les techniques des « OCR » et « détection de visage » existants et choisira la meilleure en terme de rapidité et conformité des données issues.

**Mots clés :** Android, OCR, Détection de visage.

# Abstract

This thesis presents a design and implementation of an Android mobile application that allows to take a picture of an Algerian identity card, a driver's license or passport and retrieve the information provided by the identity document such as the name, first name and the date of birth . . .and convert them into an editable and exploitable format to speed up the processing of files, sign-in system in societies. And in several others contexts.

The application will be realized with the programming language Java and Android by exploring the techniques of « OCR » and « face detection » existing and choose the best in terms of speed and compliance of the data.

**Key words :** Android, OCR, Face detection

# Remerciement

*Je remercie Allah le plus puissant de m'avoir accordé des connaissances de la science et de m'avoir donné le courage et la patience pour réaliser ce travail.*

*C'est avec vif respect et très grand plaisir que nous présentons nos sincères remerciements à :*

*Notre chère promotrice Mme. Samira MECHID pour ses précieux conseils.*

*Tous les enseignants de la structure RÉSEAUX ET TÉLÉCOMMUNICATIONS*

*Les membres des jurys pour l'effort qu'ils feront dans le but d'examiner ce modeste travail*

*Enfin, je remercie tous ceux qui ont contribué à l'élaboration de ce travail de près ou de loin et qui méritent d'y trouver leurs noms.*

# Dédicace

*Je dédie ce modeste travail :*

*À mon père*

*À ma mère*

*À la mémoire de mon frère*

*À mes frères*

*À ma soeur*

*À mes amies*

*À tous.*

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>iv</b>
<b>Liste des tableaux</b>	<b>v</b>
<b>Introduction Générale</b>	<b>2</b>
<b>1 OCR</b>	<b>5</b>
1 Introduction . . . . .	5
2 Définition . . . . .	5
3 Techniques de systèmes de reconnaissance optique de caractères . . .	6
3.1 Acquisition . . . . .	7
3.2 Segmentation d'emplacement . . . . .	7
3.3 Pré-traitement . . . . .	8
3.4 Segmentation . . . . .	9
3.5 Représentation . . . . .	11
3.6 L'extraction de caractéristiques . . . . .	11
3.7 Entraînement et reconnaissance . . . . .	12
3.8 Post-traitement . . . . .	12
4 Les types des erreurs . . . . .	13
4.1 Les erreurs de segmentation . . . . .	13
4.2 Les erreurs de reconnaissance de caractères . . . . .	14
5 Les application de l'OCR . . . . .	15
5.1 Imagerie de facture . . . . .	15
5.2 Industrie juridique . . . . .	15
5.3 Bancaire . . . . .	16
5.4 Captcha . . . . .	16
5.5 Dépôts institutionnels et bibliothèques numériques . . . . .	16
5.6 Reconnaissance musicale optique . . . . .	17
6 Conclusion . . . . .	17
<b>2 Détection du Visage</b>	<b>18</b>
1 Introduction . . . . .	18

2	Définition . . . . .	18
3	Le principe de la détection de visages . . . . .	18
4	Méthodes de détection de visages . . . . .	19
4.1	Approches et Méthodes basées sur les caractéristiques du visage	20
4.1.1	Approches basées sur la géométrie de visages . . . . .	20
4.1.2	Approches basées sur la couleur de la peau . . . . .	20
4.1.3	Approches basées sur la connaissance généralisée . . . . .	21
4.2	Les Méthodes Globales . . . . .	21
4.2.1	Approche PCA ou Les Visages Propres . . . . .	21
4.2.2	Approches Probabilistes (Statistiques) . . . . .	21
4.2.3	Approches basées sur les réseaux de neurones . . . . .	21
4.3	Les Méthodes Hybrides . . . . .	22
5	Applications . . . . .	22
5.1	Capture de mouvement du visage . . . . .	22
5.2	La reconnaissance faciale . . . . .	22
5.3	La photographie . . . . .	22
5.4	Commercialisation . . . . .	22
6	Conclusion . . . . .	23
<b>3</b>	<b>Conception et analyse</b>	<b>25</b>
1	Introduction . . . . .	25
2	La méthodologie adoptée . . . . .	25
2.1	Le processus unifié . . . . .	25
2.1.1	Processus . . . . .	25
3	Le langage de modélisation . . . . .	26
4	Spécification des besoins . . . . .	27
4.1	Les besoins fonctionnels . . . . .	27
4.1.1	Identification des acteurs . . . . .	27
4.1.2	Identification des cas cas d'utilisateur . . . . .	27
4.1.3	Le diagramme de cas d'utilisations . . . . .	28
4.1.4	Description des cas d'utilisation . . . . .	29
4.2	Les besoins non fonctionnels . . . . .	31
4.2.1	L'ergonomie des interfaces . . . . .	32
4.2.2	Sécurité . . . . .	32
4.2.3	Efficacité . . . . .	32
4.2.4	Maintenabilité . . . . .	32
5	Diagrammes de séquence . . . . .	32
6	Diagramme d'activités . . . . .	37
7	Diagramme de classes . . . . .	38
8	Passage au modèle objet relationnel . . . . .	38
9	Conclusion . . . . .	39

<b>4</b>	<b>implémentation</b>	<b>41</b>
1	Introduction . . . . .	41
2	Environnement de travail . . . . .	41
2.1	Environnement matériel . . . . .	41
2.2	Environnement logiciel . . . . .	42
2.2.1	Choix de langage . . . . .	42
2.2.2	Choix de l’outil de développement . . . . .	42
2.2.3	Choix de la bibliothèques . . . . .	43
2.2.3.1	SQLite . . . . .	43
2.2.3.2	OpenCV . . . . .	43
3	Langage de programmation . . . . .	43
3.1	JAVA . . . . .	43
3.2	XML . . . . .	43
3.3	C++ . . . . .	44
4	Diagramme de déploiement . . . . .	44
5	Présentation de l’application . . . . .	45
6	Conclusion . . . . .	50
	<b>Conclusion Générale</b>	<b>52</b>
	<b>Bibliographie</b>	<b>53</b>
	<b>Annexes</b>	<b>57</b>
	<b>Android</b>	<b>57</b>
	Les système OS mobile . . . . .	57
	Définition . . . . .	57
	Les principaux systèmes d’exploitation mobiles . . . . .	57
	Android de Google . . . . .	57
	IOS de Appel . . . . .	58
	Windows Phone de Microsoft . . . . .	59
	BlackBerry de RIM . . . . .	59
	Pourquoi Android? . . . . .	60
	Présentation de la plate-fomre Android . . . . .	61
	Historique . . . . .	62
	Fonctionnalités d’Android . . . . .	63
	Les versions d’Android : . . . . .	63
	Architecture Android . . . . .	63
	Le noyau Linux . . . . .	64
	Couche d’abstraction matérielle . . . . .	64
	Android Runtime . . . . .	66
	Bibliothèques C / C ++ natives . . . . .	66
	Java API Framework . . . . .	66

# Table des figures

1.1	Les différentes zones de reconnaissance de caractères. . . . .	6
1.2	Les composants d'un système OCR. . . . .	8
1.3	Normalisation et lissage d'un symbole. . . . .	9
1.4	application du RLSA sur une image. . . . .	10
1.5	Fusion horizontale des régions. . . . .	14
2.1	Modèle géométrique du visage. . . . .	20
3.1	La méthode de conception adoptée . . . . .	26
3.2	Diagramme de cas d'utilisation globale. . . . .	28
3.3	Diagramme de séquence «ajouter une carte d'identité» . . . . .	33
3.4	Diagramme de séquence «supprimer une carte d'identité» . . . . .	34
3.5	Diagramme de séquence «modifier une carte d'identité» . . . . .	35
3.6	Diagramme de séquence «partager une carte d'identité» . . . . .	36
3.7	Diagramme d'activités . . . . .	37
3.8	Diagramme de classe . . . . .	38
4.1	Caractéristiques sur l'ordinateur et SE utilisé . . . . .	41
4.2	Android Studio . . . . .	42
4.3	Le diagramme de déploiement . . . . .	44
4.4	Fenêtre du lancement d'application . . . . .	45
4.5	La fenêtre principale . . . . .	46
4.6	La fenêtre formulaire . . . . .	47
4.7	fenêtre pour saisir les modifications . . . . .	48
4.8	Boîte de dialogue pour confirmer la suppression . . . . .	49
4.9	Liste des application pour partager la carte. . . . .	50
1	Part de marché mondiale des OS mobiles. . . . .	58
2	Part de marché algérien des OS mobiles. . . . .	59
3	Logo Android. . . . .	61

4 L'architecture Android. . . . . 65

# Liste des tableaux

1 Version d'android . . . . . 64

# Introduction Générale

# Introduction Générale

## Introduction à la problématique

La gestion des dossiers est l'une des procédures les plus exigeantes en terme de temps et de mains-d'œuvre ...les collaborateurs passent trop de temps à saisir les données manuellement puis les transformer en format numérique.

Le progrès technologique et la sophistication des appareils photo et des gadgets permet nous de trouver une solution de numérisation et de reconnaissance de ces documents papier, en addition de stocker les données extraits automatiquement dans la base de données.

Cette solution n'est pas limitée à ce problème, on peut l'exploiter dans des divers domaines, on peut l'utiliser dans le service commercial pour faciliter le saisie de données des clients, dans les hôpitaux aussi pour saisir les données des patients.

pour cela nous avons proposé une application Android pour scanner les papiers d'identité et remplir un formulaire et les enregistrer automatiquement afin de les exploiter ultérieurement, ainsi pour aider les employés à gérer leur travail facilement.

## Objectif du projet :

Le principale objectif de notre travail est d'offrir aux employées un outil développée, facile à a utilisée et à la porté de tout le monde pour scanner les papiers d'identité afin d'acceleré les prosédure administrative. Afin d'optimiser le temps, le facteur principal dans ces processus.

## Organisation du mémoire

Le mémoire est organisé en trois parties. La première, incluant les deux chapitres suivant cette introduction port l'état de l'art sur la technologie OCR et la détection du visage, lesquelles vont nous aider à mettre en œuvre mon projet.

La deuxième partie présente la solution proposée : une conception et analyse de l'application en utilisant une méthodologie de conception et un langage de modélisation.

La troisième partie contient l'implémentation de la solution proposée .

Voici le contenu plus détaillé des chapitres présentés :

- Le premier chapitre introduit la technologie OCR, en détaillant la technique utilisée pour la mise en place de cette technologie, ainsi ses diverses domaines d'exploitations.
- Le deuxième chapitre décrit la détection de visage, les techniques de détection et les applications de cette technologie.
- Le troisième chapitre contient l'analyse et la conception détaillée en présentant les diagrammes de cas d'utilisation, de séquence et d'activité.
- Le quatrième chapitre est consacré à la partie réalisation et implémentation de l'application, d'où nous présenterons les différents outils de développement utilisés ainsi qu'une vue finale de l'application par des captures d'écran pour bien l'illustrer.
- Enfin nous terminons le mémoire par une conclusion générale.
- L'Annexe 1 présente le système d'exploitation mobile Android.

**PARTIE I**  
**Etat de l'art**

# Chapitre 1

## OCR

### 1 Introduction

La réalisations des fonctions humaines par la machine, comme la lecture, est un rêve ancien. Cependant, au cours des sept dernières décennies, la lecture automatique est passée d'un rêve à la réalité grâce au développement de systèmes de reconnaissance optique de caractères (OCR) sophistiqués et robustes. La technologie OCR nous permet de convertir différents types de documents tels que des documents papier numérisés, des fichiers pdf ou des images capturées par un appareil photo numérique en données modifiables et interrogeables.

Les systèmes OCR sont devenus l'une des applications les plus réussies de la technologie dans les domaines de la reconnaissance de formes et de l'intelligence artificielle. Bien que de nombreux systèmes commerciaux d'OCR existent pour une grande variété d'applications, les machines disponibles ne sont toujours pas capables de concurrencer les capacités de lecture humaine avec les niveaux de précision souhaités [1].

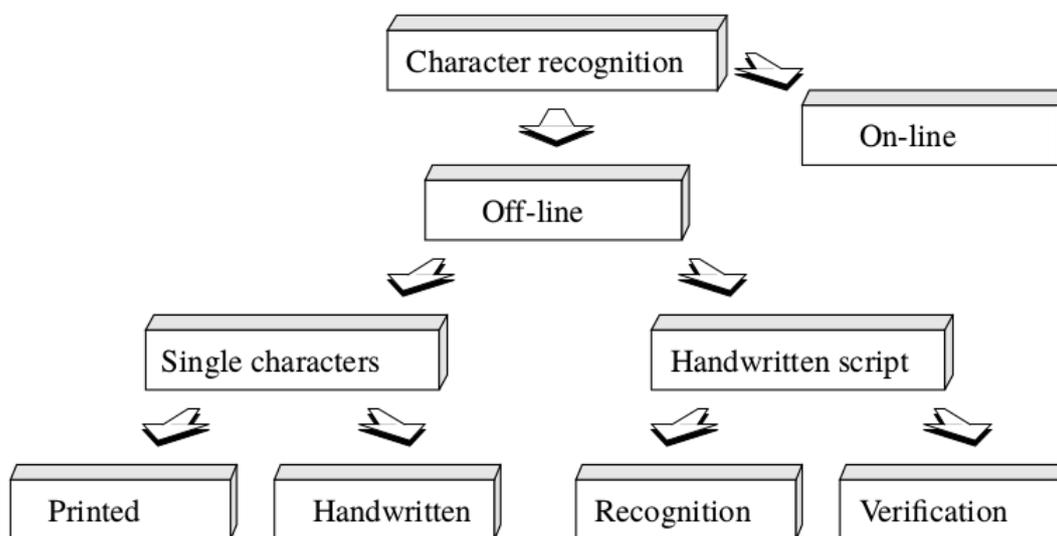
Dans ce chapitre nous avons donc pour but de présenter l'OCR et les différents zone de reconnaissance ainssi que la technique d'OCR , on met l'accent aussi sur les erreurs communes d'ocr et nous terminons par les domaines d'application.

### 2 Définition

La reconnaissance optique de caractères ou OCR (en anglais : Optical Character Recognition) est une technologie qui permet de convertir différents types de documents tels que les documents papiers scannés, les fichiers PDF ou les photos numériques vers des formats modifiables et exploitables [2], Sur le plan méthodologique, l'OCR propose des approches différentes suivant le mode d'écriture : Manuscrit ou imprimé. Deux domaines distincts sont considérés, il s'agit de la reconnaissance statique, dite encore « hors-ligne », qui travaille sur un instantané d'encre numérique

(sur une image) et la reconnaissance dynamique « en-ligne » où les symboles sont reconnus au fur et à mesure qu'ils sont écrits à la main.

OCR est requis lorsque l'information est lisible à la fois pour les humains et les machines. ses systèmes ont sculpté une place de niche dans la reconnaissance des formes [1]. Leur unicité réside dans le fait qu'il n'exige pas de contrôle du processus qui produit l'information. OCR traite le problème de la reconnaissance des caractères traités optiquement. La reconnaissance optique est effectuée hors ligne après que l'écriture ou l'impression a été achevée tandis que la reconnaissance en ligne est réalisée lorsque l'ordinateur reconnaît les caractères au fur et à mesure qu'ils sont dessinés. Les caractères imprimés et manuscrits peuvent être reconnus, mais les performances dépendent directement de la qualité des documents d'entrée. Plus l'entrée est contrainte, meilleure est la performance du système OCR. Mais quand il s'agit de la performance de l'écriture totalement libre des machines OCR est encore discutable. La figure 1.1 montre la représentation schématique de différentes zones de reconnaissance de caractères.



Source : Line EIKVIL. Ocr, optical character recognition

FIGURE 1.1 – Les différentes zones de reconnaissance de caractères.

### 3 Techniques de systèmes de reconnaissance optique de caractères

Le concept principal dans la reconnaissance automatique des motifs est d'abord d'enseigner à la machine la classe de motifs qui peut apparaître et à quoi ils ressemblent [3]. Dans l'OCR les modèles sont des lettres, des chiffres et des symboles

spéciaux comme des virgules, des points d'interrogation ainsi que des caractères différents. L'enseignement de la machine est effectué en montrant à la machine des exemples de caractères de toutes les classes différentes. Sur la base de ces exemples, la machine construit un prototype ou une description de chaque classe de caractères. Pendant la reconnaissance, les caractères inconnus sont comparés aux ces descriptions précédemment obtenues et affectés à la classe qui donne la meilleure correspondance.

La reconnaissance optique de caractères implique la traduction du texte dans l'image en codes de caractères modifiables tel que l'ASCII<sup>1</sup> [4]. Les systèmes OCR proposés par les différents chercheurs sont composés d'un ensemble de modules. L'architecture du système varie d'un système à un autre en fonction des besoins.

Le système montré dans la figure 2.1 peut être une généralisation de tous les systèmes proposés. La première étape consiste à numériser un document analogique en utilisant un scanner optique ou une caméra dans notre cas. Lorsque des régions contenant du texte sont localisées, chaque symbole est extrait par un processus de segmentation. Les symboles extraits sont prétraités, ce qui élimine le bruit pour faciliter l'extraction des caractéristiques. L'identité de chaque symbole est trouvée en comparant les caractéristiques extraites avec les descriptions des classes de symboles obtenues à travers une phase d'apprentissage précédente. Enfin, des informations contextuelles sont utilisées pour reconstruire les mots et les nombres du texte original.

### 3.1 Acquisition

Le premier élément de l'OCR est l'acquisition. Grâce à ce processus, l'image numérique du document original est capturée et numérisée. Le résultat de cette phase représente l'entrée de système. Il dépend de deux paramètres importants Résolution et Niveau d'éclairage. L'acquisition est assez simple mais très importante car elle influence sérieusement les étapes suivantes. à ce niveau, nous utilisons un code android pour capturer et enregistrer l'image.

### 3.2 Segmentation d'emplacement

Le composant suivant de l'OCR est la segmentation de l'emplacement. La segmentation détermine les constituants d'une image. Il est nécessaire de localiser les régions de document qui ont des données imprimées et qui se distinguent des figures et des graphiques. La segmentation est l'isolation des caractères ou des mots. La plupart des algorithmes OCR segmentent les mots en caractères isolés qui sont

---

1. American Standard Code for Information Interchange en français Code américain normalisé pour l'échange d'information

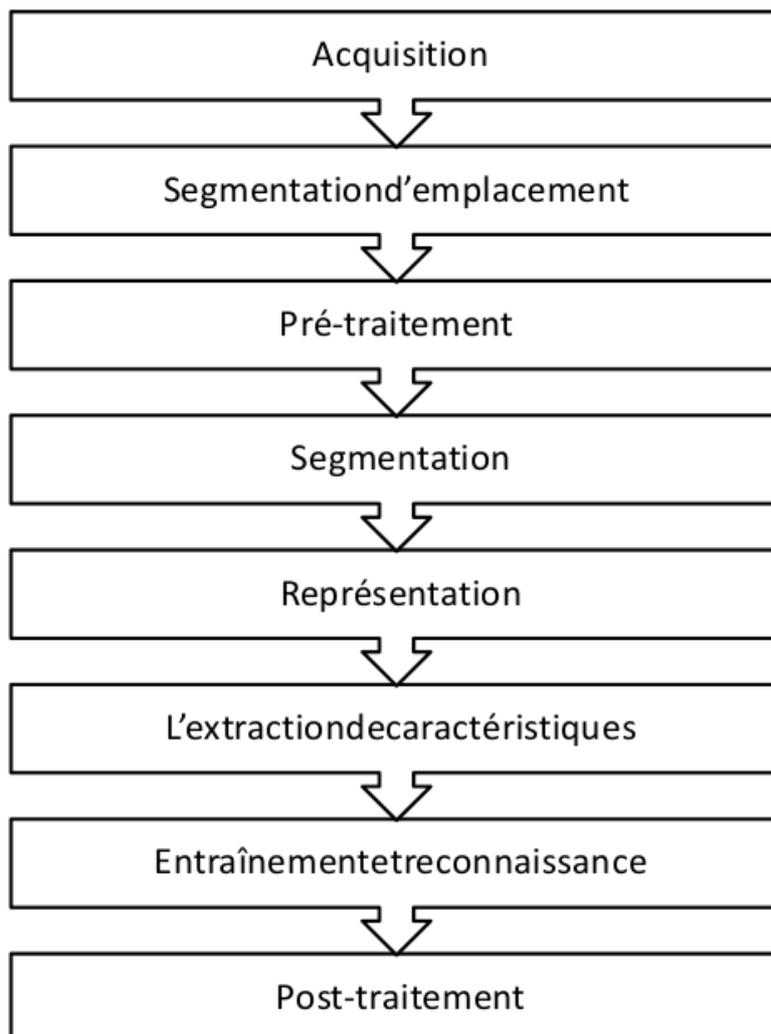


FIGURE 1.2 – Les composants d'un système OCR.

reconnus individuellement. Habituellement, la segmentation est effectuée en isolant chaque composant connecté. Cette technique est facile à mettre en œuvre, mais des problèmes surviennent si les caractères touchent ou si ils sont fragmentés et se composent de plusieurs parties [5].

### 3.3 Pré-traitement

Le troisième composant de l'OCR est le pré-traitement. L'image résultant du processus d'acquisition peut contenir une certaine quantité de bruit, les caractères peuvent être tachés ou cassés. Certains de ces défauts, qui peuvent causer ultérieurement de faibles taux de reconnaissance, peuvent être éliminés en utilisant un pré-processeur pour lisser les caractères numérisés.

Le lissage implique à la fois le remplissage et l'amincissement. Le remplissage élimine les petites coupures, les espaces et les trous dans les caractères numérisés, tandis que l'amincissement réduit la largeur de la ligne. Les techniques les plus courantes pour lisser, déplace une fenêtre à travers l'image binaire du personnage, appliquant certaines règles au contenu de la fenêtre [2].

En plus du lissage, le pré-traitement inclut généralement la normalisation. La normalisation est appliquée pour obtenir des caractères de taille, d'inclinaison et de rotation uniformes. Pour pouvoir corriger la rotation, l'angle de rotation doit être trouvé. Pour les pages tournées et les lignes de texte, des variantes de Hough transform sont couramment utilisées pour détecter l'asymétrie. Cependant, il n'est pas possible de trouver l'angle de rotation d'un seul symbole tant que le symbole n'a pas été reconnu. Par conséquent, le pré-traitement entraîne la symétrie et l'alignement des caractères dans une image numérisée [6].



FIGURE 1.3 – Normalisation et lissage d'un symbole.

### 3.4 Segmentation

L'étape de pré-traitement produit une image de caractère propre dans le sens où une quantité suffisante d'informations de forme, une compression élevée et un faible bruit sur une image normalisée sont obtenus. Le composant OCR suivant est la segmentation. Bien que l'OCR fonctionne sur des textes relativement longs, la reconnaissance s'effectue non pas de manière globale mais caractère par caractère. Le but de la segmentation est de trouver dans une image où est situé le texte et isoler chacun de ses caractères. Cette étape peut être divisée en deux phases. D'abord la recherche de l'emplacement des lignes de texte puis dans un second temps la recherche de l'emplacement de chacun des caractères dans cette ligne.

La segmentation peut être effectuée par le RLSA<sup>2</sup>. Cet algorithme consiste à noircir les pixels blancs compris entre deux pixels noirs si le nombre de ces pixels blancs est inférieur à  $n$ . Concrètement cela revient à noircir les zones ou les points

---

2. Run Length Smoothing Algorithm

noirs sont proches. L'algorithme est effectué verticalement et horizontalement. Le résultat de la segmentation est l'intersection des zones noires obtenues par les deux répétitions de l'algorithme. En faisant varier  $n$ , on pourra segmenter les caractères, les mots, les lignes ou les paragraphes [7].

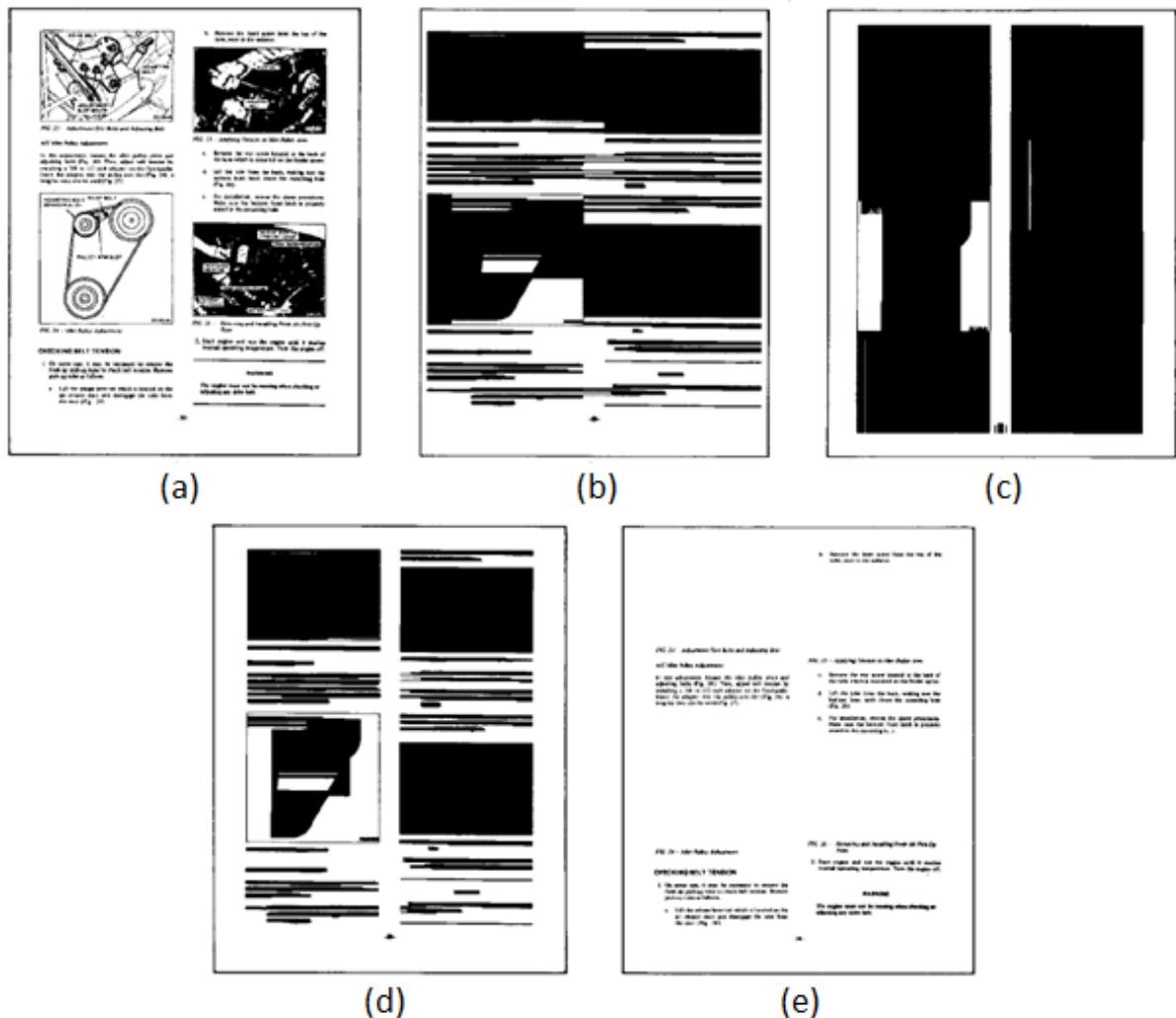


FIGURE 1.4 – application du RLSA sur une image.

L'image (a) correspond au document initial.

L'image (b) est le résultat du RLSA effectué horizontalement.

L'image (c) est le résultat du RLSA effectué verticalement

L'image (d) correspond à l'intersection des zones noires des images (b) et (c).

L'image (e) indique les blocs considérés comme étant du texte.

La segmentation permet d'isoler les éléments textuels, mots et caractères, pour la reconnaissance. Elle se base sur des mesures de plages blanches (interlignes et inter caractères) pour faire la séparation. La multiplicité des polices et la variation

des justifications empêchent de stabiliser les seuils de séparation, conduisant à la génération de blancs inexistantes ou au contraire à l'ignorance de blancs séparateurs de mots. Ce type d'erreur est très fréquent, d'après une récente étude réalisée par Nagy et al[8].

### 3.5 Représentation

Le cinquième composant de l'OCR est la représentation. La représentation de l'image joue l'un des rôles les plus importants dans tout système de reconnaissance. Dans le cas le plus simple, les images en niveaux de gris ou binaires sont transmises à un système de reconnaissance. Cependant, dans la plupart des systèmes de reconnaissance, afin d'éviter une complexité supplémentaire et d'augmenter la précision des algorithmes, une représentation plus compacte et caractéristique est requise. A cet effet, un ensemble de caractéristiques est extrait pour chaque classe qui aide à le distinguer des autres classes tout en restant invariant par rapport aux différences caractéristiques au sein de la classe. Les méthodes de représentation des images de caractères sont généralement classées en trois groupes principaux : transformation globale et expansion en série, représentation statistique et la représentation géométrique et topologique [9].

### 3.6 L'extraction de caractéristiques

Le sixième composant OCR est l'extraction de caractéristiques. L'objectif de l'extraction de caractéristiques est de capturer les caractéristiques essentielles des symboles. L'extraction de caractéristiques est acceptée comme l'un des problèmes les plus difficiles de la reconnaissance de formes. La manière la plus directe de décrire le caractère est par l'image raster réelle. Une autre approche consiste à extraire certaines caractéristiques qui caractérisent les symboles mais qui laissent des attributs sans importance. Les techniques d'extraction de ces caractéristiques sont divisées en trois groupes. (a) la distribution des points (b) les transformations et les expansions en série et (c) l'analyse structurelle. Les différents groupes de caractéristiques sont évalués en fonction de leur sensibilité au bruit, de leur déformation, de leur facilité de mise en œuvre et de leur utilisation. Les critères utilisés dans cette évaluation sont : (a) robustesse en termes de bruit, distorsions, variation de style, translation et rotation et (b) utilisation pratique en termes de vitesse de reconnaissance, de complexité de mise en œuvre et d'indépendance. Certaines des techniques d'extraction de caractéristiques couramment utilisées sont l'appariement et la corrélation de modèles, les transformations, la distribution de points et l'analyse structurelle. Pour les lecteurs intéressés, d'autres détails sont disponibles dans [5].

Une autre tâche importante associée à l'extraction de caractéristiques est la classification. La classification est le processus qui consiste à identifier chaque caractère et à lui attribuer une classe de caractères correcte. Les deux principales catégories

d'approches de classification pour l'OCR sont la théorie de la décision et les méthodes structurelles. Dans la reconnaissance de la décision théorique, la description du caractère est représentée numériquement dans le vecteur de caractéristiques. Il peut également y avoir des caractéristiques de modèle dérivées de la structure physique du caractère qui ne sont pas aussi facilement quantifiées. Ici, la relation entre les caractéristiques peut être importante pour décider de l'appartenance à une classe. Par exemple, si nous savons qu'un caractère se compose d'un trait vertical et d'un trait horizontal, il peut être «L» ou «T». La relation entre deux traits est nécessaire pour distinguer les caractères. Les principales approches de la reconnaissance de la théorie de la décision sont les classificateurs de distances minimales, les classificateurs statistiques et les réseaux de neurones. Dans la reconnaissance structurelle, les méthodes syntaxiques sont les approches les plus répandues [9][5]

### 3.7 Entraînement et reconnaissance

Le septième composant OCR est l'entraînement et la reconnaissance. Les systèmes OCR utilisent largement les méthodologies de reconnaissance de formes qui affectent un échantillon inconnu dans une classe prédéfinie. Les OCR sont étudiées dans quatre approches générales de la reconnaissance des formes, comme suggéré dans [9] : Correspondance des modèle, les techniques statistiques, les techniques structurelles et les RNA (réseaux de neurones artificiels). Ces approches ne sont pas nécessairement indépendantes ni disjointes les unes des autres. Parfois une technique de l'OCR dans une approche peut également être considérée comme faisant partie d'autres approches. Dans toutes les approches ci-dessus, les techniques de l'OCR utilisent des stratégies holistiques ou analytiques pour les étapes de l'entraînement et de reconnaissance. La stratégie holistique utilise des approches descendantes pour reconnaître le caractère complet en éliminant le problème de segmentation. Le prix de cette économie de calcul est de limiter le problème de l'OCR à un vocabulaire limité. En outre, en raison de la complexité introduite par la représentation d'un seul caractère ou d'un seul trait, la précision de reconnaissance est diminuée. D'autre part, les stratégies analytiques utilisent une approche ascendante à partir du niveau de l'AVC ou du personnage et allant vers la production d'un texte significatif. Les algorithmes de segmentation explicites ou implicites sont requis pour cette stratégie, non seulement en ajoutant une complexité supplémentaire au problème, mais également en introduisant une erreur de segmentation dans le système. Cependant, avec la coopération de l'étape de segmentation, le problème est réduit à la reconnaissance de simples caractères isolés ou traits, qui peuvent être manipulés pour un vocabulaire illimité avec des taux de reconnaissance élevés [10].

### 3.8 Post-traitement

Le huitième composant de l'OCR est le post-traitement. Certaines des activités de post-traitement les plus couramment utilisées comprennent le groupement

et la détection et la correction d'erreurs. Dans le regroupement des symboles dans le texte sont associés à des chaînes. Le résultat de la reconnaissance de symbole simple dans le texte est un ensemble de symboles individuels. Cependant, ces symboles ne contiennent généralement pas suffisamment d'informations. Ces symboles individuels sont associés les uns aux autres en composant des mots et des nombres. Le regroupement des symboles en chaînes est basé sur l'emplacement des symboles dans le document. Les symboles qui sont suffisamment proches sont regroupés. Pour les polices avec un processus de groupement de pas fixe, il est facile de connaître la position de chaque caractère. Pour les caractères composés, la distance entre les caractères est variable. La distance entre les mots est significativement plus grande que la distance entre les caractères et le regroupement est donc possible. Les problèmes se produisent pour les caractères manuscrits lorsque le texte est faussé. Jusqu'à ce que le regroupement de chaque caractère soit traité séparément, le contexte dans lequel chaque caractère apparaît n'a pas été exploité. Cependant, dans les problèmes avancés de reconnaissance de texte optique, un système constitué uniquement d'une reconnaissance de caractère unique n'est pas suffisant. Même les meilleurs systèmes de reconnaissance ne donneront pas une identification correcte à 100% de tous les caractères [9][8][3].

Le post-traitement est effectué quand le processus de reconnaissance aboutit à la génération d'une liste de lettres ou de mots possibles, éventuellement classés par ordre décroissant de vraisemblance. Le but principal est d'améliorer le taux de reconnaissance en faisant des corrections orthographiques ou morphologiques à l'aide de dictionnaires de digrammes, tri-grammes ou n-grammes. Quand il s'agit de la reconnaissance de phrases entières, on fait intervenir des contraintes de niveaux successifs : Lexical, syntaxique ou sémantique [2].

## 4 Les types des erreurs

L'intelligence artificielle ne peut pas rivaliser avec l'intelligence humaine, c'est pourquoi OCR peut faire des erreurs différentes pendant le processus de reconnaissance, tel que les erreurs de segmentation et les erreurs de reconnaissance de caractères.

### 4.1 Les erreurs de segmentation

La segmentation du document conduit à la décomposition du document en unités structurelles telles que des régions textuelles ou des graphiques. Une mauvaise application de la méthode de segmentation conduit à des erreurs [11]. Ces erreurs sont :

- Fusion horizontale de régions textuelles : cette erreur conduit à la fusion de lignes adjacentes appartenant à des colonnes différentes. Elle influe sur l'ordre de lecture. comme le montre la figure 1.5 où l'ordre normal : 1, 2, 3, 4 est transformé en 1, 3, 2, 4.

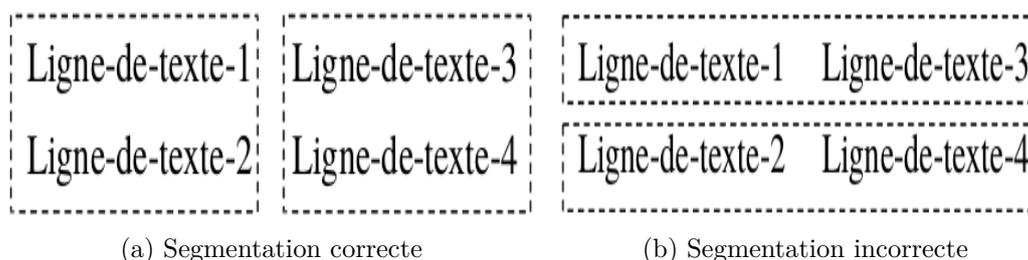


FIGURE 1.5 – Fusion horizontale des régions.

- Fusion verticale de régions textuelles : Cette erreur conduit à regrouper deux paragraphes. Elle n'altère pas l'ordre de lecture mais sa correction est nécessaire pour la classification du texte, par ex. le titre ou des notes de bas de page collés au texte.

- Scission horizontale de régions textuelles : Cette erreur conduit à un faux ordre de lecture.

- Scission verticale de régions textuelles : Ce cas est similaire au précédent ; il ne cause pas d'erreur grave car l'ordre de lecture n'est pas changé.

- Région non détectée : cette erreur indique la non détection d'une région de texte, assimilée sans doute à un graphique ou à du bruit.

- Graphique/bruit confondu à du texte : Ceci indique que l'OCR a dû interpréter un graphique ou du bruit comme du texte. Cette erreur conduit à des séquences perturbées de caractères dans le texte. Cette erreur concerne également les formules mathématiques.

- Fusion horizontale avec graphique/bruit : Ceci conduit, comme dans le cas précédent, à l'insertion de séquences erronées de caractères dans le texte.

- Fusion verticale avec graphique/bruit : Ce cas est identique au cas précédent, sauf qu'il se produit verticalement [12].

## 4.2 Les erreurs de reconnaissance de caractères

Un OCR peut faire quatre types d'erreur sur la reconnaissance des caractères :

- Une confusion, en remplaçant un caractère par un autre, si les caractères sont morphologiquement proches (par ex. « o,0 », « c,( », « n,h »,« s,5»).
- Une suppression, en ignorant un caractère, considéré comme un bruit de l'image.
- Un rejet, en refusant un caractère soit parce qu'il n'est pas connu par le système, soit parce que le système n'est pas sûr de sa reconnaissance ; dans ce cas, le système propose un caractère spécial, en général le car celui-ci apparaît rarement dans les documents papier.
- Un ajout, en dédoublant un caractère par deux autres dont la morphologie de leurs formes accolées peut être proche du caractère (par ex. « m, rn», « d,cl »,« w,vv ») [12].

## 5 Les application de l'OCR

La reconnaissance optique des caractères a été appliquée à un certain nombre d'applications. Certains d'entre eux ont été expliqués ci-dessous.

### 5.1 Imagerie de facture

La Correction Cordial gratuitement sur 1000 caractères L'imagerie des factures est largement utilisée dans de nombreuses applications commerciales pour suivre les dossiers financiers et éviter l'accumulation de paiements accumulés. Dans les agences gouvernementales et les organisations indépendantes, OCR simplifie la collecte et l'analyse des données, entre autres processus. À mesure que la technologie continue à se développer, de plus en plus d'applications sont trouvées pour la technologie OCR, y compris l'utilisation accrue de la reconnaissance de l'écriture manuscrite. De plus, d'autres technologies liées à la ROC, tel que la reconnaissance de codes à barres, sont utilisées quotidiennement dans le commerce de détail et dans d'autres industries [13].

### 5.2 Industrie juridique

L'industrie juridique est également l'un des bénéficiaires de la technologie OCR. OCR est utilisé pour numériser des documents, et directement entrer dans la base de données informatique. Les professionnels du droit peuvent également rechercher des documents requis à partir d'énormes bases de données en tapant simplement quelques mots-clés [13].

### 5.3 Bancaire

Une autre application importante de l'OCR est dans le secteur bancaire, où il est utilisé pour traiter les contrôles sans intervention humaine. Une vérification peut être insérée dans une machine où le système scanne le montant à émettre et le bon montant d'argent est transféré. Cette technologie a presque été perfectionnée pour les chèques imprimés et elle est assez précise pour les chèques manuscrits, tout en réduisant le temps d'attente dans les banques[13].

### 5.4 Captcha

Un CAPTCHA est un programme qui peut générer et classer des tests que l'humain peut réussir mais que les programmeurs actuels ne peuvent pas faire. Le piratage est une menace sérieuse pour l'utilisation d'Internet. Aujourd'hui, la plupart des activités humaines comme les transactions économiques, l'admission à l'éducation, les inscriptions, les réservations de voyages, etc. sont effectuées via Internet et tout cela nécessite un mot de passe mal utilisé par les pirates informatiques. Ils créent des programmes pour imiter les attaques de dictionnaire et les fausses inscriptions automatiques qui mènent au gaspillage de mémoire et aux ressources du site Web. Attaque de dictionnaire est une attaque contre les systèmes authentifiés par mot de passe où un pirate écrit un programme à essayer à plusieurs reprises différents mots de passe comme dans un dictionnaire des mots de passe les plus courants. Dans CAPTCHA, une image composée de séries de lettres numériques est générée, masquée par les techniques de distorsion d'image, la taille et la variation des polices, les arrière-plans gênants, les segments aléatoires, les hautes lumières et le bruit de l'image. Ce système peut être utilisé pour supprimer ce bruit et segmenter l'image pour rendre l'image traitable pour les systèmes OCR<sup>3</sup> [13].

### 5.5 Dépôts institutionnels et bibliothèques numériques

Les dépôts institutionnels sont des collections numériques des résultats créés dans une université ou un établissement de recherche. C'est un site en ligne de données intellectuelles d'une institution, en particulier un institut de recherche où il est collecté, conservé et diffusé. Il contribue à ouvrir les résultats d'une institution et à lui donner une visibilité et un impact plus important au niveau mondial. Favorise et encourage les approches interdisciplinaires de la recherche et facilite le développement et la partage de matériel et d'outils de recherche numérique. Il s'agit essentiellement d'une collection d'articles de revues à comité de lecture, de comptes rendus de conférences, de données de recherche, de monographies, de livres, de thèses et de dissertations et de présentations. Leur premier rôle est de fournir la littérature Open Access. La mise en œuvre pratique de ceci comprend la mise en place d'un système qui consiste en un scanner qui balaye les documents [13].

---

3. Optical Character Recognition

## 5.6 Reconnaissance musicale optique

Système d'apprentissage automatisé extraire des informations à partir d'images et fait partie des principales recherches. La reconnaissance optique musicale (OMR) née en 1950 est un domaine développé et visait initialement à reconnaître des feuilles imprimées qui peuvent être éditées en forme jouable à l'aide de méthodes électroniques et électrochimiques. Un système OMR a de nombreuses applications comme le traitement de différentes classes de musique, la numérisation à grande échelle de données musicales et peut également être utilisé pour la diversité dans la notation musicale. L'amélioration de l'image et la segmentation constituent l'étape de base, d'où l'accent mis sur le document [13].

## 6 Conclusion

Dans ce chapitre, nous avons présenté certains concepts généraux liés à la reconnaissance optique des caractères. Ensuite nous avons énuméré les principaux problèmes rencontrés par l'OCR. Nous avons aussi abordé les différentes étapes intervenant dans la conception d'un système de reconnaissance de caractères. Le chapitre suivant est consacré à une autre technique qui est la détection de visage.

# Chapitre 2

## Détection du Visage

### 1 Introduction

Le problème de la détection de visages a été traité par plusieurs méthodes et techniques différentes. Dans ce chapitre, on essaye d'expliquer le principe de la détection de visages et les enjeux à prendre en considération. Après, on fait un survol des différentes approches et méthodes de détection, et à la fin on présente quelque application de cette vision.

### 2 Définition

La détection de visage est une technologie informatique utilisée dans une variété d'applications qui identifie les visages humains dans les images numériques [14]. La détection de visages fait également référence au processus psychologique par lequel les humains localisent et assistent les visages dans une scène visuelle [15].

La détection de visages peut être considérée comme un cas spécifique de détection de classe d'objets. Dans la détection de classe d'objets, la tâche consiste à trouver les emplacements et les tailles de tous les objets d'une image appartenant à une classe donnée. Les exemples incluent les torsos supérieurs, les piétons et les voitures [16].

### 3 Le principe de la détection de visages

Le concept de base de la détection de visages serait celui des K-PPV (K-Plus Proches Voisins), qui consiste à parcourir l'image avec une fenêtre, puis à comparer chaque «Petite image» extraite avec une série de visages types et de définir un «Visage» comme étant tout résultat dont la distance à l'une des images de la base soit suffisamment faible ( $\text{Erreur} < \text{seuil}$ ).

Partons de ce constat, il est nécessaire que la détection de la présence ou non d'un visage dans une image soit basée sur des éléments stables et relativement descriptifs du visage humain et qui permettent ensuite de le reconnaître. Parmi ces éléments,

on peut citer la forme du visage, la couleur de la peau, le contour des yeux, la forme du nez ou de la bouche..

Mais en considérant des tailles, des orientations, des rotations et des éclairages différents il faudrait pouvoir comparer chaque « imagerie » extraite à des centaines de références ! Si l'on rajoute les expressions faciales (sourires, grimaces,...). La détection de visages devient un problème difficile à traiter et c'est pour plusieurs raisons :

- Bien que la plupart des visages soient structurellement semblables avec des caractères morphologiques communs (yeux, bouche, nez,...) placés selon une certaine configuration spatiale, il existe de grandes différences entre deux visages (forme du nez, couleur des yeux, couleur de peau,...).
- Certains caractères morphologiques peuvent être présents ou non selon les visages comme par exemple la moustache, la barbe, ...
- Certains caractères extérieurs peuvent déformer des caractères morphologiques comme par exemple le bronzage modifiant partiellement la couleur de la peau, l'âge peut modifier les rides du visage ou la couleur des cheveux, un éventuel accident peut laisser ses traces sur le visage, les lunettes, ...
- Un visage peut avoir des orientations et des dimensions très différentes. Il s'y ajoute les conditions d'éclairage et la position dans l'image où certaines zones du visage peuvent être cachées soit par un objet soit par un autre visage.
- Les visages sont avant tout des structures 3D dans un espace 3D, de nombreux paramètres s'ajoutent encore au problème original : des contraintes de luminosité (dues soit à la position de la tête, soit au type d'éclairage choisi), de couleur, d'ombres et de rotations éventuelles de la tête, ...

## 4 Méthodes de détection de visages

Il existe de nombreuses méthodes de détection de visages et qui se diffèrent, aussi bien par les approches qu'elles emploient, que par les techniques d'apprentissage qu'elles utilisent. Yang et Al. classifient ces techniques en quatre classes :[17, 18]

- Techniques descriptives basées sur la connaissance.
- Techniques basées sur l'extraction de paramètres caractéristiques invariants.
- Techniques basées sur la superposition de caractéristiques.
- Techniques basées sur l'apparence.

Les techniques basées sur l'apparence utilisent l'analyse statistique et l'apprentissage automatique pour construire des machines capables de séparer les visages des non-visages. Les réseaux de neurones, les machines à vecteurs de support (SVM), les classificateurs Bayesiens, les modèles de Markov cachés (HMM) sont parmi les méthodes d'apprentissage automatique les plus souvent utilisées.

Ces méthodes peuvent être divisées en deux catégories : Les méthodes géométriques et les méthodes globales. La performance de ces méthodes dépend de la précision avec laquelle les informations utiles du visage sont extraites.

## 4.1 Approches et Méthodes basées sur les caractéristiques du visage

On les appelle aussi les méthodes à traire, à caractéristiques locales, ou analytiques. L'analyse du visage humain est donnée par la description individuelle de ses parties, leurs positions et de leurs relations. Ce modèle correspond à la manière avec laquelle l'être humain perçoit le visage, c'est-à-dire, à nos notions de traits de visage et ses parties comme les yeux, le nez et la bouche, ce qui permet de conclure la présence ou non du visage dans l'image à analyser [19].

### 4.1.1 Approches basées sur la géométrie de visages

La plupart des approches se sont concentrées sur l'extraction des traits du visage à partir d'une image et sur la définition d'un modèle adéquat pour représenter ce visage. Un certain nombre de stratégies ont modélisé et classé les visages sur la base de distances normalisées et angles entre points caractéristiques : Les yeux, les sourcils, la bouche, le nez,... mais peuvent être d'un niveau de détail beaucoup plus fin.

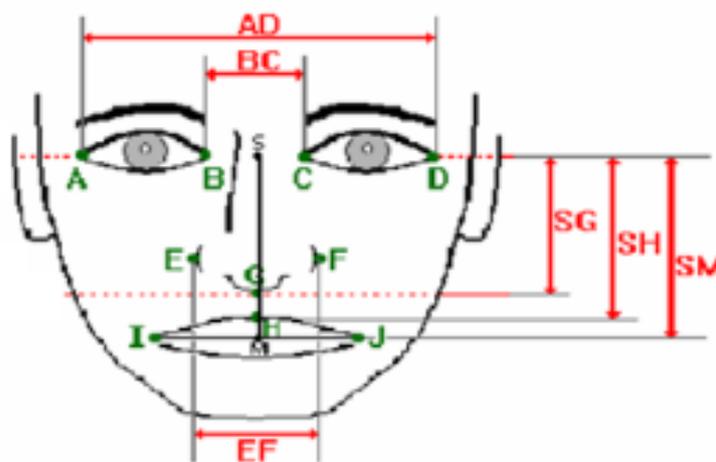


FIGURE 2.1 – Modèle géométrique du visage.

### 4.1.2 Approches basées sur la couleur de la peau

Dans cette approche, la couleur de peau humaine a été employée comme un dispositif efficace pour la détection de visage, et les applications liées. Bien que la couleur de peau diffère d'un individu à un autre, plusieurs études ont prouvé que la différence principale existe dans l'intensité plutôt que la chrominance. Plusieurs espaces de couleur ont été employés pour marquer des pixels de peau. Quoique l'information de couleur semble être un outil efficace pour identifier des secteurs faciaux, les modèles de couleur de peau peuvent échouer quand le spectre

(la température corrélée de couleur) de la source lumineuse change de manière significative. En outre, les caractéristiques du dispositif d'acquisition (équilibre spécifiquement blanc) effectueront également la transformation de couleur entre l'environnement et l'image.

### 4.1.3 Approches basées sur la connaissance généralisée

Dans cette approche, les algorithmes développés sont basés sur des heuristiques au sujet de l'aspect de visages. Bien qu'il soit simple de créer une heuristique pour décrire le visage humain, la difficulté principale est dans la traduction de ces heuristiques dans des règles de classification d'une manière efficace.

## 4.2 Les Méthodes Globales

Cette classe regroupe les méthodes qui mettent en valeur les propriétés globales du visage. Le visage est traité comme un tout. Dans ces méthodes on génère une base d'exemples à partir de laquelle un classificateur va apprendre ce qu'est un visage (apprentissage). Ces systèmes sont très performants, mais très lents en phase d'apprentissage donc lourds à mettre en oeuvre [19]. Parmi les approches les plus importantes réunies au sein de cette classe on trouve :

### 4.2.1 Approche PCA ou Les Visages Propres

L'approche PCA<sup>4</sup> (Principal Components Analysis) est une manière intuitive et appropriée de construire un sous-espace pour représenter une classe d'objet dans beaucoup de cas. Cependant, pour modéliser la variété dans des images de visages, PCA n'est pas nécessairement optimal. L'espace de visage pourrait mieux être représenté en le divisant en sous-classes. La plupart des méthodes qui ont été proposées sont basées sur un certain mélange de Gaussiens multidimensionnel.

### 4.2.2 Approches Probabilistes (Statistiques)

Ces approches reposent essentiellement sur la théorie de décision pour résoudre les problèmes de classement et de classification, et c'est pour ça qu'ils utilisent généralement la classification fondée sur le théorème de Bayes. Ces approches posent le problème de la complexité de calcul qui reste très élevée.

### 4.2.3 Approches basées sur les réseaux de neurones

Cette approche repose essentiellement sur la notion d'apprentissage qui est depuis de nombreuses années au coeur de la recherche en intelligence artificielle. Puisque la détection de visages peut être comprise comme problème d'identification de modèle de deux classes (visage ou non-visage), plusieurs méthodes utilisant les réseaux de neurones ont été présentées pour la solution.

---

4. Principal component analysis en français : Analyse des composants principaux

### 4.3 Les Méthodes Hybrides

La plupart des techniques courantes dans la détection de visages sont basées sur l'hybridation de deux ou plusieurs méthodes déjà expérimentées.

## 5 Applications

### 5.1 Capture de mouvement du visage

La capture de mouvements faciaux est le processus de conversion électronique des mouvements du visage d'une personne en une base de données numérique à l'aide de caméras ou de scanners laser. Cette base de données peut ensuite être utilisée pour produire une animation informatique CG (infographie) pour des films, des jeux ou des avatars en temps réel. Parce que le mouvement des caractères CG est dérivé des mouvements de personnes réelles, il en résulte une animation de personnages informatiques plus réaliste et nuancée que si l'animation était créée manuellement[20].

### 5.2 La reconnaissance faciale

Un système de reconnaissance faciale est une technologie capable d'identifier ou de vérifier une personne à partir d'une image numérique ou d'une image vidéo à partir d'une source vidéo. Il existe plusieurs méthodes dans lesquelles les systèmes de reconnaissance faciale fonctionnent, mais en général, ils fonctionnent en comparant les caractéristiques faciales sélectionnées à partir d'une image donnée avec des visages dans une base de données.

La détection de visage est utilisée en biométrie, souvent dans le cadre (ou avec) d'un système de reconnaissance faciale. Il est également utilisé dans la vidéosurveillance, l'interface homme-machine et la gestion de bases de données d'images [21].

### 5.3 La photographie

Certains appareils photo numériques récents utilisent la détection de visage pour l'autofocus. La détection de visages est également utile pour sélectionner des zones d'intérêt dans les diaporamas photo utilisant un effet Ken Burns à l'échelle et à l'échelle. Les appareils modernes utilisent également la détection de sourire pour prendre une photo à un moment approprié.

### 5.4 Commercialisation

La détection des visages suscite l'intérêt des spécialistes du marketing. Une webcam peut être intégrée dans un téléviseur et détecter tout visage qui passe. Le

Le système calcule ensuite la race, le sexe et la tranche d'âge du visage. Une fois les informations collectées, une série de publicités spécifiques à la race, au sexe ou à l'âge détecté peut être lue. Un exemple d'un tel système est OptimEyes et est intégré dans le système de signalisation numérique Amscreen [22].

## 6 Conclusion

Dans ce chapitre, nous avons vu le principe de la détection de visages humains dans une image, ensuite on a présenté les méthodes de détection et à la fin, on a cité quelques applications de la détection de visage. Comme celui-ci nous avons arrivé à la fin de l'état de l'art et nous allons entamer la conception dans le chapitre suivant.

**PARTIE II**  
**Conception et Analyse**

# Chapitre 3

## Conception et analyse

### 1 Introduction

Ce chapitre présente une mise en perspective de la modélisation informatique orienté objet de notre projet au cours de laquelle différents travaux d'analyse. Pour ce faire, la méthodologie informatique UP ou Unified Process (en français processus unifié) combinée au langage de modélisation UML<sup>5</sup> ont été utilisés.

### 2 La méthodologie adopté

Il existe plusieurs méthodes de développement logiciel construites sur UML comme la méthode : UP, RUP, TTUP, UP agile, XP, 2TUP .....

Parmi ses méthodes notre choix est basé sur la méthode UP (Unified Process). Car elle est une méthodologie qui permet de prendre en compte plusieurs aspects différents et pertinents pour la construction de modèles au cours d'itérations successives [23].

#### 2.1 Le processus unifié

Le UP ou Unified Process est un processus de conception/développement de logiciel. Il apporte une approche disciplinée pour assigner des tâches et des responsabilités dans une organisation de développement. Son but est d'assurer la production de logiciels de qualité qui rencontrent les besoins de ses utilisateurs [24].

##### 2.1.1 Processus

Un processus décrit qui fait quoi, comment et quand. Le UP définit ainsi quatre éléments primaires de modélisation :

- Le membre est le qui : Diagramme de cas d'utilisation.
- L'activité est le comment : Diagramme d'activités.
- L'artefact est le quoi : Diagramme de séquence.

---

5. Unified Modeling Language en français Langage de modélisation unifié

## Objectifs

- Construire des modèles de systèmes.
- Organiser le travail.
- Gérer le cycle de vie de A à Z.
- Gérer le risque.
- Obtenir de manière répétitive des produits de qualité constante.

La méthode adoptée dans ce chapitre incite à suivre la démarche présentée par le schéma dans la figure 3.1 , cette manière de marcher utilise quatre diagrammes UML : un diagramme fonctionnel ( diagramme de cas d'utilisation), deux diagrammes dynamiques ( diagramme d'activités et diagramme de séquence) et un diagramme statique (diagramme de classes).

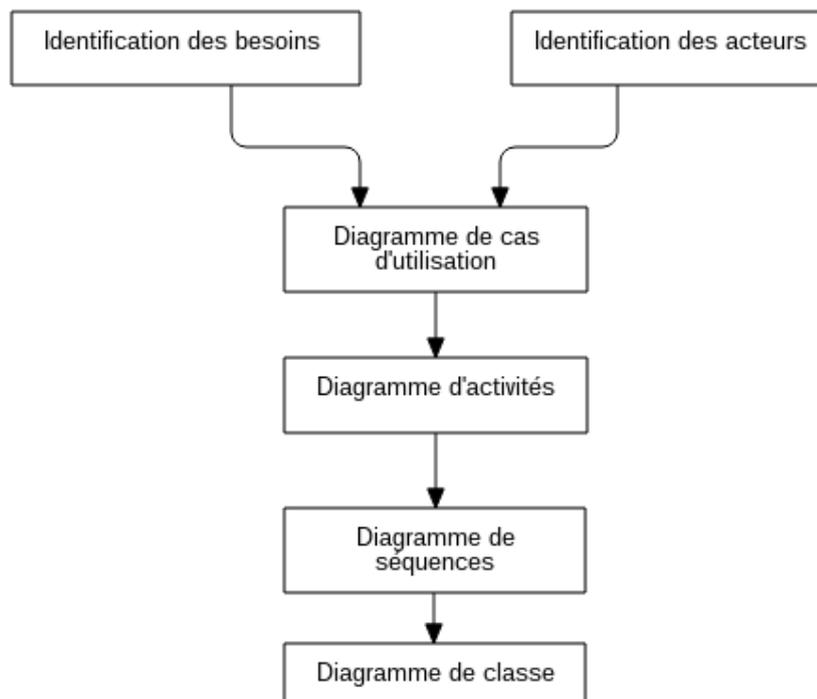


FIGURE 3.1 – La méthode de conception adoptée

## 3 Le langage de modélisation

Pour programmer une application, il ne convient pas de mettre l'accent que sur l'écriture du code. Il faut d'abord organiser ses idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. C'est cette

démarche antérieure à l'écriture que l'on appelle modélisation, son produit est un modèle.

Cette modélisation nécessite l'utilisation d'un langage permettant la description du système logiciel ainsi que sa compréhension par ses futurs utilisateurs. Pour ce faire, nous choisissons UML comme langage de modélisation de notre système, car il comble une lacune importante des technologies objets. Il permet d'exprimer et d'élaborer des modèles objets, indépendamment de tout langage de programmation. De plus, grâce à sa notation graphique, il permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évolution de solutions. Ainsi que, l'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

## 4 Spécification des besoins

Avant de commencer le travail d'analyse et de conception, il est indispensable de savoir exactement ce que l'on va construire, c'est le but de la spécification des besoins. On va donc définir les besoins fonctionnels et non fonctionnels de l'application [25].

### 4.1 Les besoins fonctionnels

Afin de garantir la réussite et l'efficacité de notre projet, il faut définir avec précision la bordure de l'application à développer. Ceci signifie l'énumération des divers services que notre système est supposé offrir aux différents utilisateurs.

#### 4.1.1 Identification des acteurs

Un acteurs (utilisateur, dispositif matériel ou autre système) est une entité externe au système en interaction avec ce dernier. on a que un seul acteur qui est l'utilisateur de l'application.

#### 4.1.2 Identification des cas cas d'utilisateur

Les cas d'utilisation permettent d'exprimer les besoins des utilisateurs d'un système. Ils mettent en évidence les relations fonctionnelles entre les acteurs et le système étudié.

#### ● Les besoins de l'utilisateur :

1. Scanner une nouvelle carte d'identité.
2. Scanner un nouveau permis de conduire.
3. Scanner un nouveau passeport.
4. Supprimer une carte d'identité.

5. Supprimer un permis de conduire.
6. Supprimer un passeport.
7. Modifier une carte d'identité.
8. Modifier un permis de conduire.
9. Modifier un passeport.
10. Partager les information d'une carte.

#### 4.1.3 Le diagramme de cas d'utilisations

Il s'agit d'un graphe d'acteurs, un ensemble de cas d'utilisation a l'intérieur de la frontière du système, d'associations de communication (participation) entre acteurs et cas d'utilisation, et de généralisations parmi les cas d'utilisation [26].

Le diagramme dans la figure 3.2 illustre bien le fonctionnement de l'application, il représente le diagramme global des cas d'utilisation.

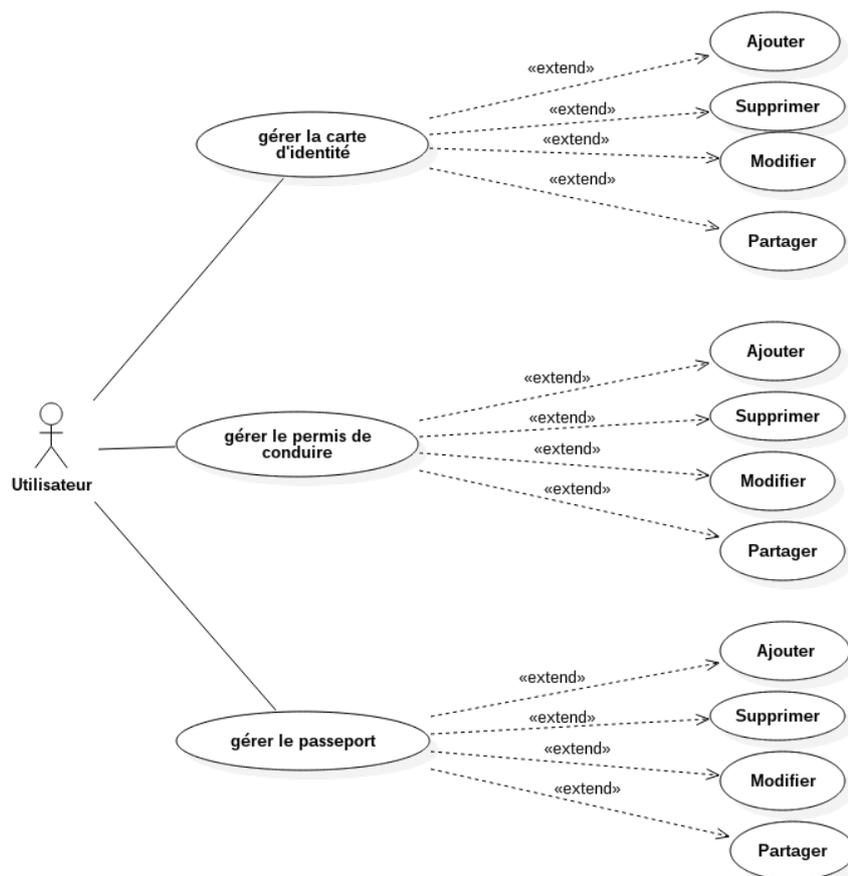


FIGURE 3.2 – Diagramme de cas d'utilisation globale.

#### 4.1.4 Description des cas d'utilisation

pour détailler la dynamique de cas d'utilisation la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système [27]. On peut dire alors qu'un cas d'utilisation est une collection de scénarios de succès ou d'échec qui décrit la façon dont un acteur utilise le système pour atteindre un objectif. nous allons ainsi distinguer :

- Le scénario «nominal» : Celui qui satisfait les objectifs des acteurs par le chemin le plus direct de succès.
- Les «alternatives» : Qui comprennent tous les autres scénarios de succès (fin nominale) ou d'échec (erreur).

##### • Cas d'utilisation «ajouter une carte d'identité»

**But :** Ce cas est utilisé pour scanner une nouvelle carte d'identité et l'ajouter à la base de données.

##### **Scénario nominal :**

1. L'utilisateur ouvre l'application.
2. L'application récupère toutes les informations dans la base de données afin de les afficher après.
3. La base de données effectue tout le traitement et envoie les résultats au système.
4. L'application s'ouvre, en affichant la page d'accueil à l'écran.
5. L'utilisateur clique sur ajouter une nouvelle carte .
6. Le système envoie la liste des cartes que l'utilisateur peut scanner.
7. L'utilisateur choisit la carte d'identité.
8. Le système appelle la caméra pour prendre une photo de la carte et récupère la photo.
9. Le système effectue des traitements sur l'image afin d'extraire les données.
10. Le système envoie les données extraites à la base de données.
11. Le système informe l'utilisateur qu'une nouvelle carte d'identité a été ajoutée avec succès.

##### **Alternatives :**

**9a** Le système n'a rien trouvé dans l'image

**9a 1.** Le système informe l'utilisateur.

**● Cas d'utilisation «supprimer une carte d'identité»**

**But :** Ce cas est utilisé pour supprimer une carte d'identité de la base de données.

**Scénario nominal :**

1. L'utilisateur ouvre l'application.
2. L'application récupère toutes les informations dans la base de données afin de les afficher après.
3. La base de données effectue tout le traitement et envoie les résultats au système.
4. L'application s'ouvre, en affichant la page d'accueil à l'écran.
5. L'utilisateur clique sur une des cartes dans la liste des cartes déjà scannées.
6. le système ouvre cette carte avec plus de détails et des nouvelles opérations que l'utilisateur peut les exécuter.
7. L'utilisateur choisit supprimer.
8. Le système envoie un pop-up pour confirmer la suppression.
9. L'utilisateur clique sur supprimer pour continuer la tâche.
10. Le système envoie à la base de données une requête pour supprimer la carte choisie.
11. Le système informe l'utilisateur que la carte a été supprimée.

**Alternatives :**

**9a** L'utilisateur clique sur annuler .

**9a 1.** Le système ferme la fenêtre contextuelle et retourne à la page précédente.

**● Cas d'utilisation «modifier une carte d'identité»**

**But :** Ce cas est utilisé pour modifier une carte d'identité qui existe déjà dans la base de données.

**Scénario nominal :**

1. L'utilisateur ouvre l'application.
2. L'application récupère toutes les informations dans la base de données afin de les afficher après.
3. La base de données effectue tout le traitement et envoie les résultats au système.

4. L'application s'ouvre, en affichant la page d'accueil à l'écran.
5. L'utilisateur sur une des cartes dans la liste des cartes déjà scannées.
6. Le système ouvre cette cartes avec plus de détails et des nouvelle opérations que l'utilisateur peut les exécuter.
7. L'utilisateur choisit modifier.
8. Le système envoie une autre page ou l'utilisateur peut insérer les modification.
9. L'utilisateur effectue les modifications puis clique sur valider.
10. Le système envoie à la base de données Les modification.
11. Le système informe l'utilisateur que la carte a été modifié.

#### ● Cas d'utilisation «partager une carte d'identité»

**But :** Ce cas est utilisé pour partager une carte d'identité qui existe déjà dans la base de données.

##### **Scénario nominal :**

1. L'utilisateur ouvre l'application.
2. L'application récupère toutes les informations dans la base de données afin de les afficher après.
3. La base de données effectue tout le traitement et envoie les résultats au système.
4. L'application s'ouvre, en affichant la page d'accueil à l'écran.
5. L'utilisateur sur une des cartes dans la liste des cartes déjà scanner
6. Le système ouvre cette cartes avec plus de détails et des nouvelle opérations que l'utilisateur peut les exécuter.
7. L'utilisateur choisit partager.
8. Le système envoie une liste des application de communication pour choisir une.
9. L'utilisateur choisit une parmi les applications affichées et envoie le message au destinataire.

**Remarque :** Pour effectuer les opérations sur les permis de conduire ou le passeport on peut suivre le même scénario que la carte d'identité.

## 4.2 Les besoins non fonctionnels

Les besoins non fonctionnels représentent l'ensemble des restrictions ou des contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement. Dans notre cas il s'agit de :

### 4.2.1 L'ergonomie des interfaces

L'application doit présenter une interface conviviale qui tient compte de la taille de l'écran de l'appareil mobile auquel elle est destinée. La manipulation de l'interface ne doit pas nécessiter des connaissances poussées en informatique, elle doit être simple et claire afin de s'adapter aux connaissances de son utilisateur.

### 4.2.2 Sécurité

L'application doit garantir aux utilisateurs que toutes les données sont sécurisées et qu'elles ne seront pas voler pas ou malutiliser.

### 4.2.3 Efficacité

L'application doit être fonctionnelle indépendamment de toutes circonstances peuvent entourer l'utilisation.

### 4.2.4 Maintenabilité

Le code de l'application doit être lisible et compréhensible pour faciliter son évolution et sa mise à jour.

## 5 Diagrammes de séquence

Les diagrammes de séquence et les diagrammes de collaboration présentés au point suivant sont appelés de manière générique diagrammes d'interactions dans UML. En effet, ils décrivent les interactions entre les objets que contient un système ou un processus. Ils permettent également de montrer les interactions d'un système avec son environnement. Les diagrammes de séquence permettent de modéliser un système de manière dynamique et ils s'attachent principalement à montrer la circulation et l'ordre chronologique des messages, autrement dit ils décrivent la circulation de l'information. Ils sont directement basés sur les scénarios définis avec les cas d'utilisation. Les diagrammes de collaboration quant à eux montrent ces interactions sous un angle structurel plutôt que temporel [28] .

Quand les cas d'utilisation se ressemblent leurs diagrammes de séquences peuvent se ressembler à leur tour, on se contentera alors de présenter uniquement les diagrammes de séquences d'un seul type des cartes. Les diagramme retenus sont donc :

- Diagramme de séquence «ajouter une carte d'identité» La figure 3.3
- Diagramme de séquence «supprimer une carte d'identité» La figure 3.4
- Diagramme de séquence «modifier une carte d'identité» La figure 3.5
- Diagramme de séquence «partager une carte d'identité» La figure 3.6

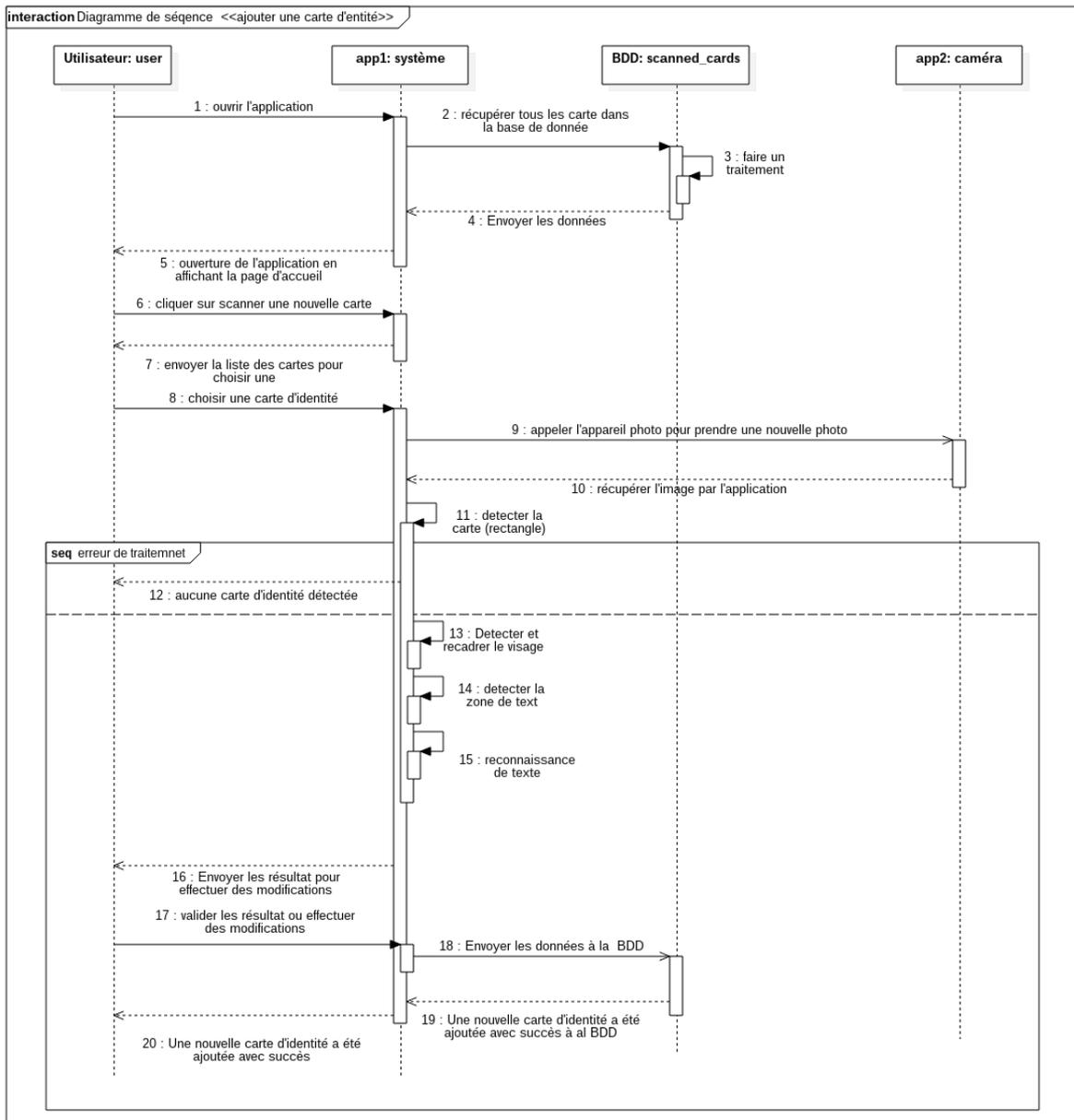


FIGURE 3.3 – Diagramme de séquence «ajouter une carte d'identité»

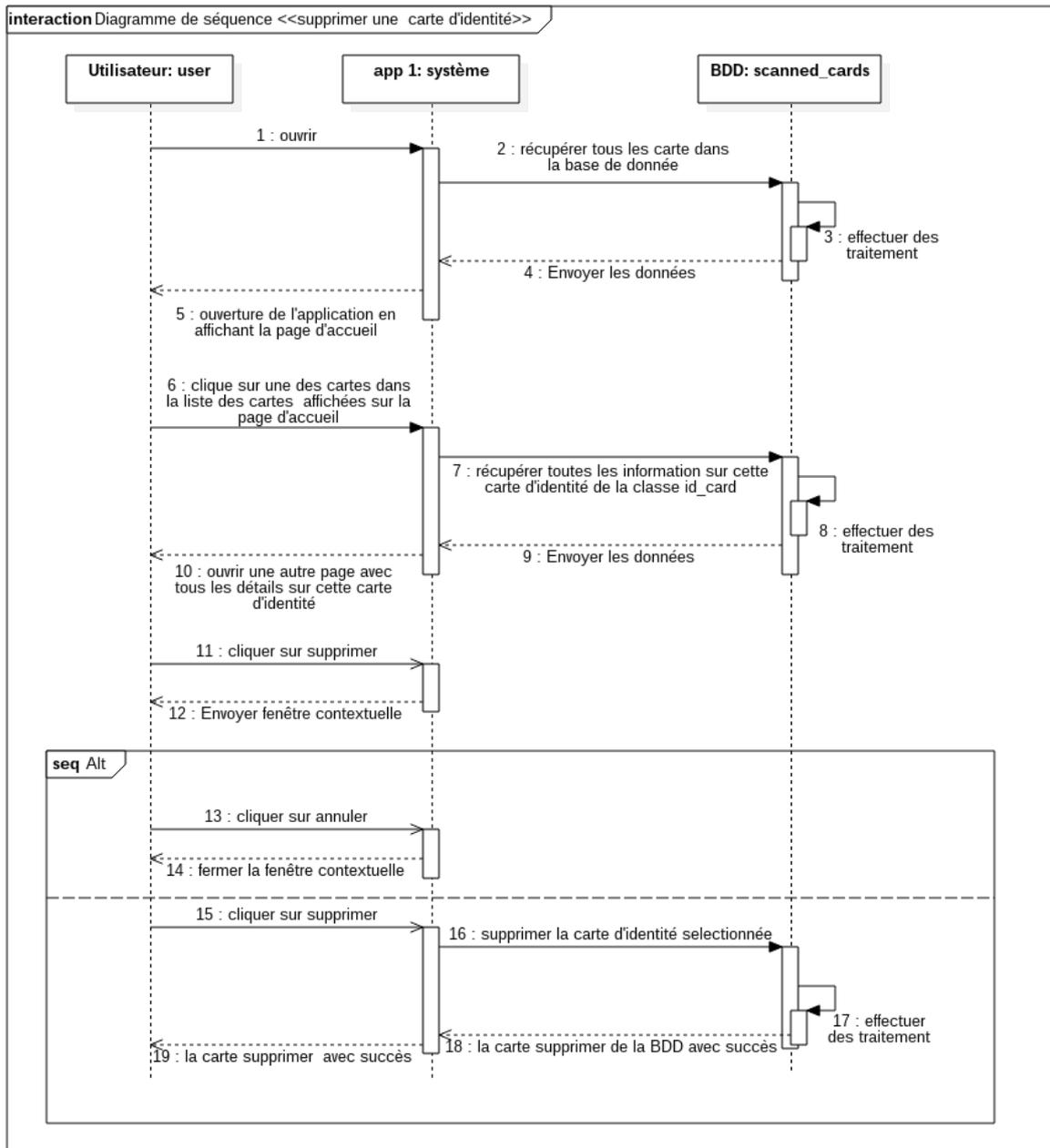


FIGURE 3.4 – Diagramme de séquence «supprimer une carte d'identité»

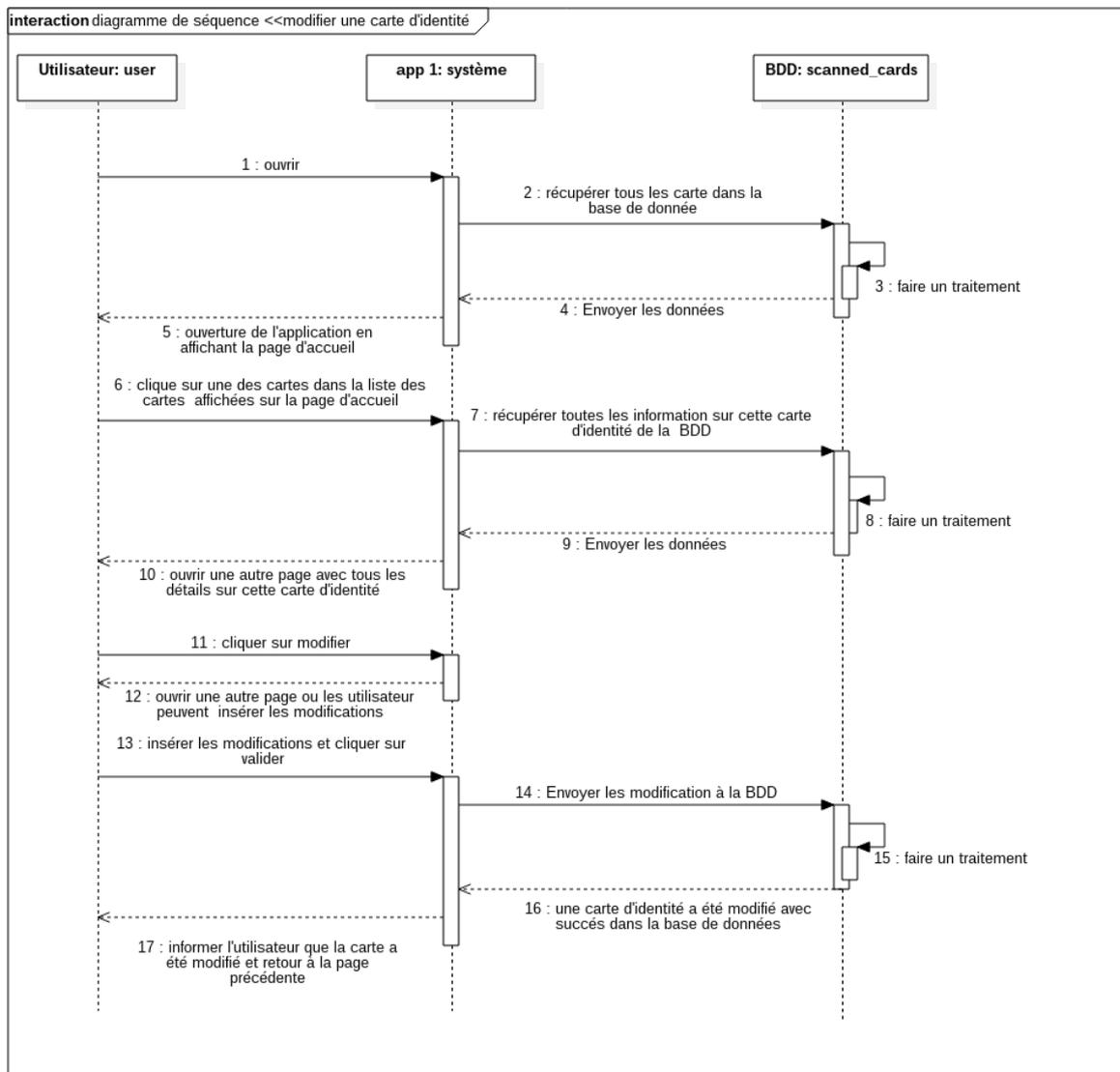


FIGURE 3.5 – Diagramme de séquence «modifier une carte d'identité»

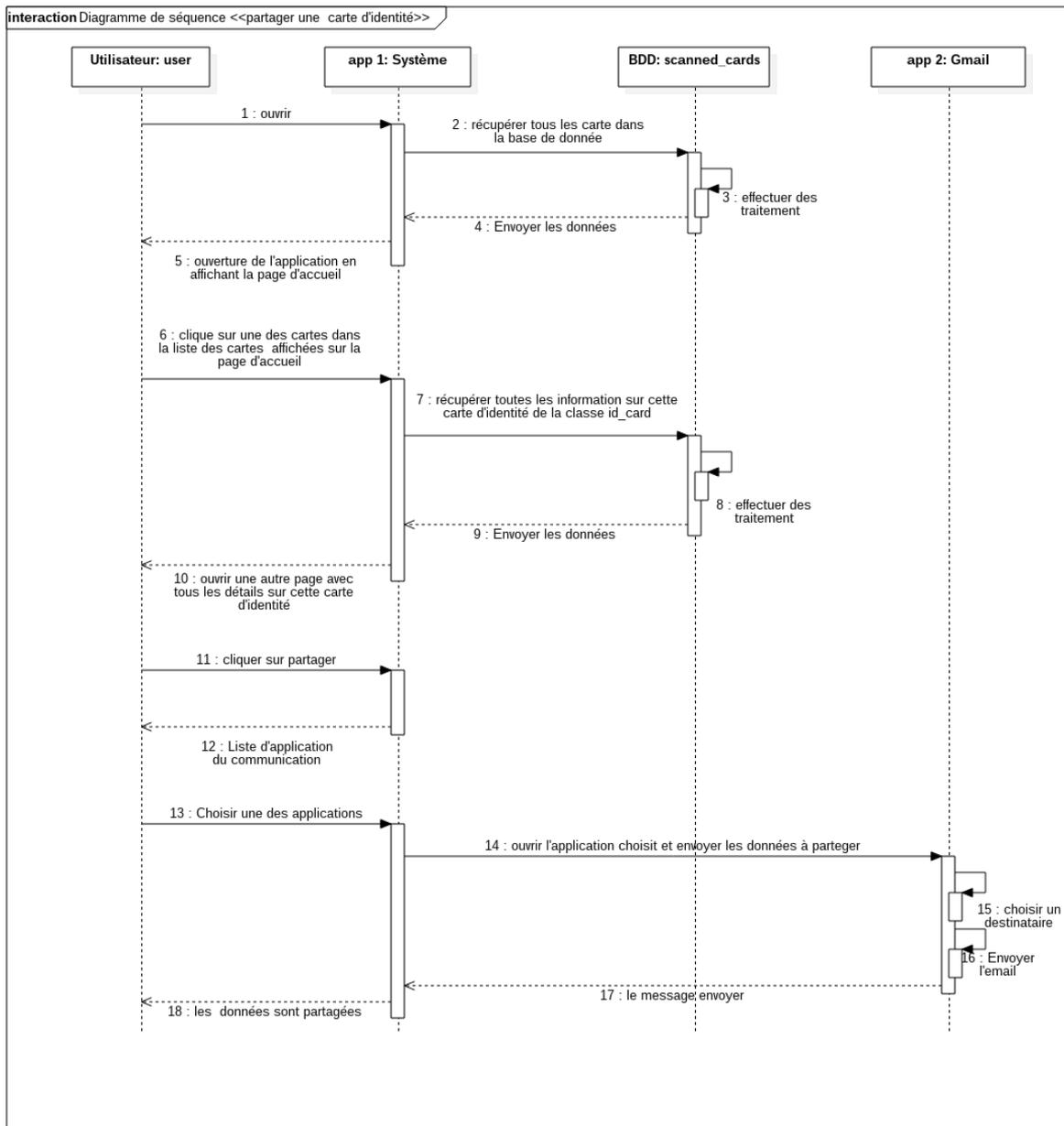


FIGURE 3.6 – Diagramme de séquence «partager une carte d'identité»

## 6 Diagramme d'activités

Les diagrammes d'activités montrent les flux de contrôle et les flux d'information qui circulent au sein d'un système ou d'un sous-système. Ils permettent de représenter le séquençage exact des activités et de définir des conditions d'exécution [28]. Les diagrammes d'activités sont indispensables à la modélisation de processus, car ils décrivent précisément comment les processus ou sous-processus sont réalisés. La figure 3.7 présente le diagramme d'activités de cas d'utilisations « ajouter une carte ».

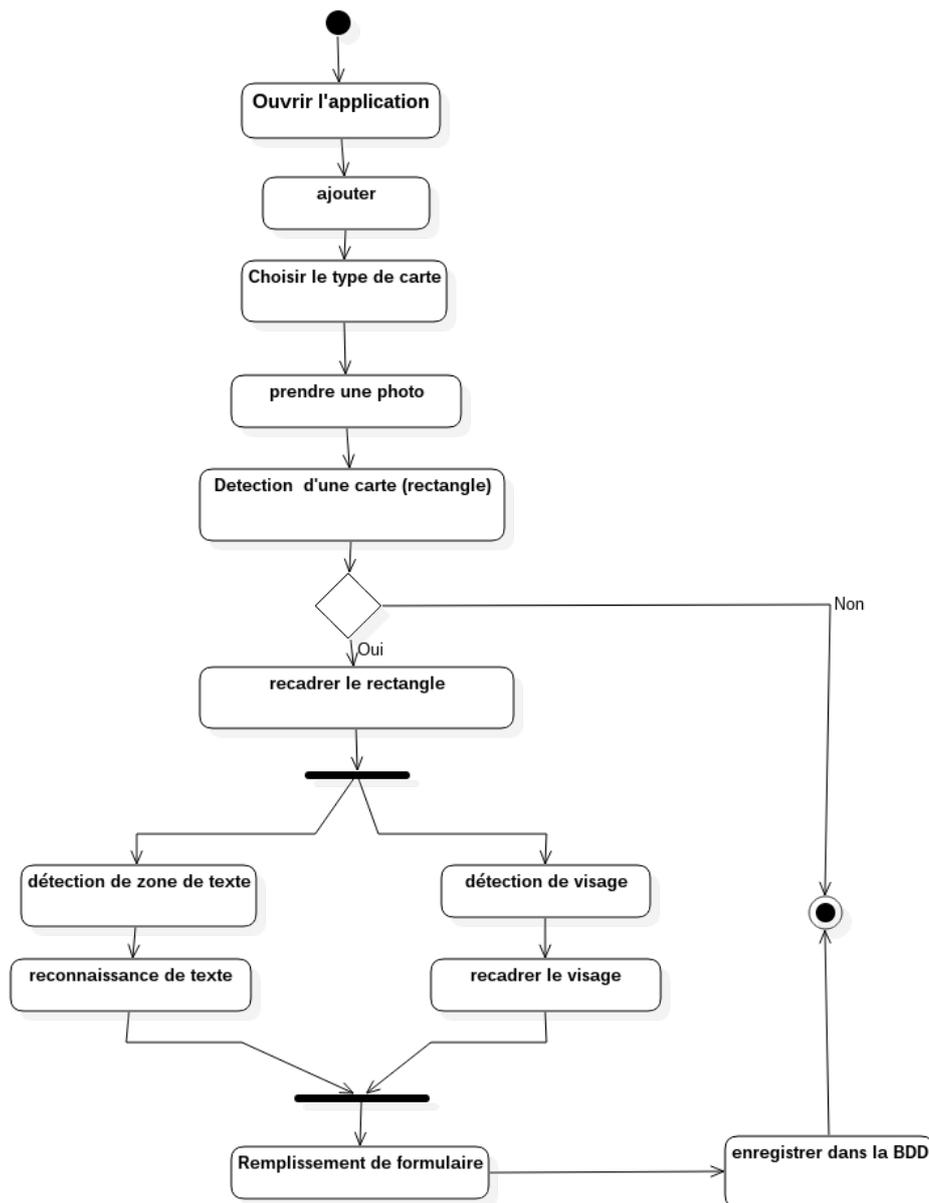


FIGURE 3.7 – Diagramme d'activités

## 7 Diagramme de classes

Le diagramme de classes la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser le cas d'utilisation. La figure 3.8 montre le diagramme de classe de notre application.

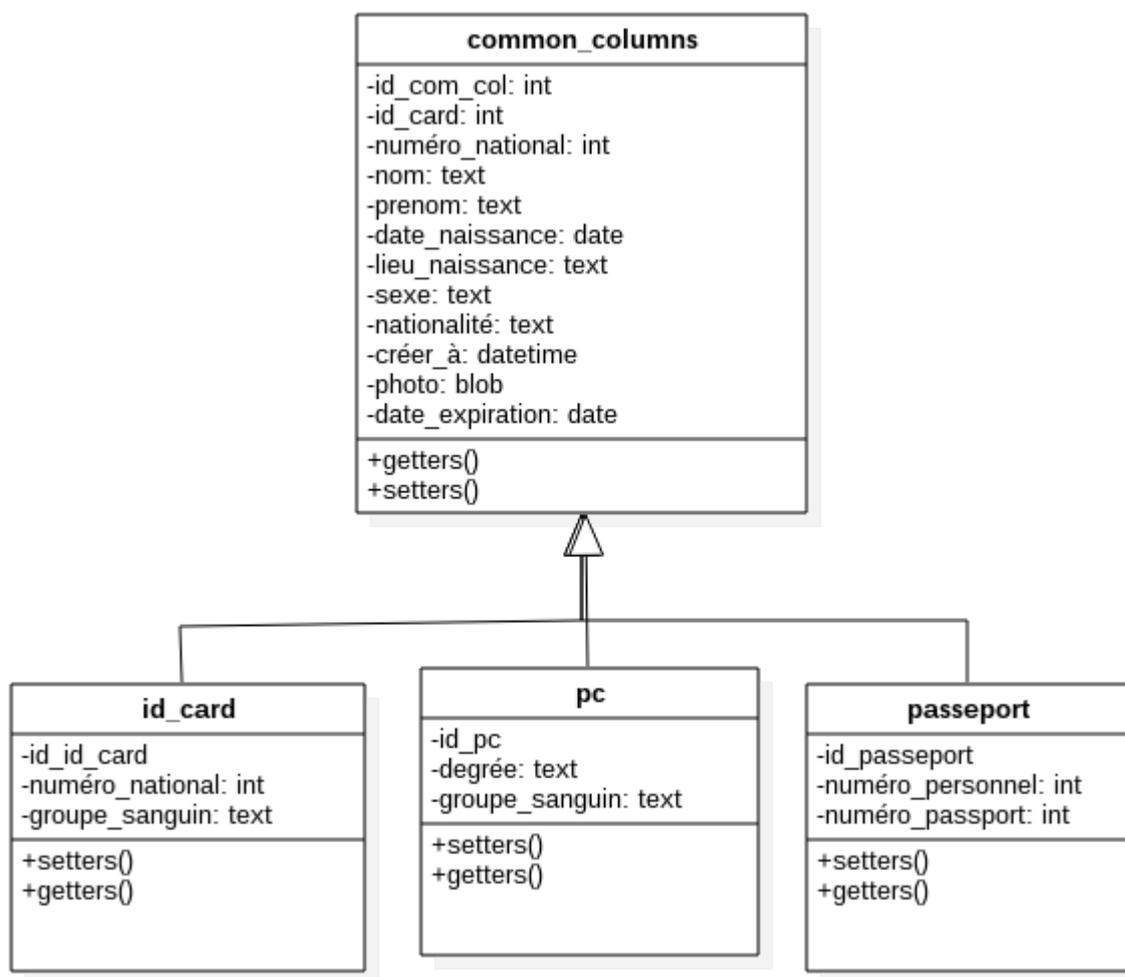


FIGURE 3.8 – Diagramme de classe

## 8 Passage au modèle objet relationnel

On a dans ce qui suit identifier les tables qui constituent la base de données. Pour ce faire on doit effectuer un passage du modèle objet au modèle relationnel :

- Représenter des classes d'objets en tables : Chaque classe est représentée par une table.

- Représenter des associations :
  - Chaque association plusieurs à plusieurs est représentée par une table distincte.
  - Une association un à plusieurs est enfouie comme clé étrangère.

Pour distinguer les clés primaires et les clés étrangères , on procède comme suit :

- Les clés primaires sont présentées avec caractères gras soulignés.
- Une étoile est posée à côté des clés étrangères.

Le modèle relationnel de mon système est :

- common\_columns(id\_com\_col, id\_card, numéro\_national, nom, prenom, date\_naissance, lieu\_naissance, sexe, nationalité, créer\_à, photo, date\_expiration).
- id\_card(id\_id\_card, groupe\_sanguin, id\_com\_col\*).
- passeport(id\_passeport, numéro\_personnel, numéro\_passport, id\_com\_col\*).
- pc(id\_pc, groupe\_sanguin, degré, id\_com\_col\*).

## 9 Conclusion

Dans ce chapitre on a présenté le langage de modélisation, après on a déterminé les besoins fonctionnels et non fonctionnels de l'acteurs de notre système. Une analyse plus détaillée est obtenue grâce aux diagrammes de cas d'utilisation et de séquence. une fois la conception terminée , on est à présent capable de passer à l'implémentation, objet de chapitre suivant.

**PARTIE III**  
**Implémentation**

# Chapitre 4

## implémentation

### 1 Introduction

Ce chapitre constitue le coeur et l'âme du processus de la réalisation de l'application, c'est-à-dire citer les outils, langages de programmation et environnement de développement utilisé. Il est consacré à la présentation des outils utilisés pour réaliser le projet, et la représentation de l'application finale avec des illustrations en capteurs d'écran.

### 2 Environnement de travail

Pour la réalisation de travail, nous avons eu recours aux environnements matériel et logiciel :

#### 2.1 Environnement matériel

Pour le devloppement de cette application on a utilisé un ordinateur portable TOSHIBA présenté dans la figure 4.1

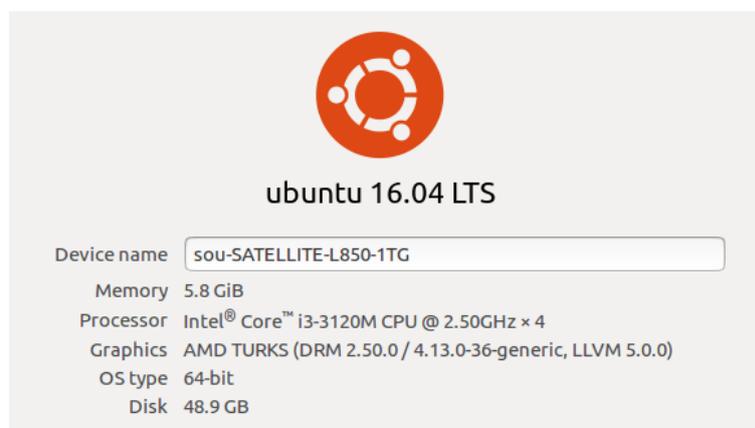


FIGURE 4.1 – Caractéristiques sur l'ordinateur et SE utilisé

On a utilisé aussi deux téléphone portable ses caractiristique sont dféctré au suivant

- SAMSUNG Galaxy S5 SM-G900T
  - RAM : 2 Go
  - Android : 6.0.1
  - API :23
- SAMSUNG Galaxy J5 SM-J500H
  - RAM : 1.5 Go
  - Android : 5.1.1
  - API :22

## 2.2 Environnement logiciel

Dans cette partie on y présentera les choix des technologies et outils adoptés pour le développement du système :

### 2.2.1 Choix de langage

Tandis qu'un nouveau langage de programmation a vraiment attiré l'attention de la communauté Android est Kotlin, mais On a choisi JAVA car il est conseillé pour les débutants en programmation Android. On a utilisé aussi d'autres langages de programmation comme le c++ et le XML pour les interfaces d'utilisateur.

### 2.2.2 Choix de l'outil de développement

Le choix d'Android studio est effectué grâce à sa facilité et simplicité. Android Studio est l'environnement de développement intégré officiel du système d'exploitation Android de Google, basé sur le logiciel IntelliJ IDEA de JetBrains et conçu spécifiquement pour le développement Android.



FIGURE 4.2 – Android Studio

### 2.2.3 Choix de la bibliothèques

**2.2.3.1 SQLite** Pour la base de données, étant donné qu'on travaillera sur un dispositif local, Android propose d'utiliser le moteur SQL nommé SQLite.

SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnel accessible par le langage SQL. Contrairement aux serveurs de base de données traditionnels, comme MySQL ou PostgreSQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

**2.2.3.2 OpenCV** Après avoir établi une étude des concepts théoriques et rassembler le maximum d'informations sur les OCR et la détection de visage ,on a testé deux bibliothèques (OpenCv et TensorFlow) mais mon choix s'est fixé sur la librairie Open Cv. puisque il est plus facile à adapter avec Android studio par rapport à TensorFlow, ainsi que ses résultats sont bons en terme d'efficacité et rapidité.

**Qu'est-ce qu'Open CV ?** Open CV est une bibliothèque C développée par Intel, multiplateforme (Windows, Linux, Mac OS), gratuite (sous licence BSD) et open-source. Son but est de fournir des fonctions pour le traitement d'images en temps réel :

- Utilisation de webcams
- Détection d'objets
- Reconnaissance de mouvements
- Application de filtres

## 3 Langage de programmation

### 3.1 JAVA

Java est un langage de programmation largement utilisé, conçu expressément dans l'environnement distribué d'Internet. C'est le langage de programmation le plus populaire.

### 3.2 XML

Il représente le langage de balisage extensible et est une spécification recommandée par le W3C en tant que langage de balisage général. Cela signifie que, contrairement aux autres langages de balisage, XML n'est pas prédéfini, vous devez donc définir vos propres balises. L'objectif principal de la langue est le partage de données entre différents systèmes, tels qu'Internet.Android fournit un vocabulaire XML simple qui correspond aux classes et sous-classes View, telles que celles des widgets et des layouts.

### 3.3 C++

Le C++ est un langage de programmation de bas niveau. Il est l'évolution du langage C, un langage plus simple et surtout beaucoup plus ancien. Le C++ est utilisé pour écrire des programmes bas niveau qui interagissent directement avec le système.

## 4 Diagramme de déploiement

Le diagramme de déploiement fait parti des diagrammes structuraux (statique), il représente :

- la disposition physique des ressources matérielles qui constituent le système et montre la répartition des composants (élément logiciels) sur ces matériels.
- La nature des connexions de communication entre les différentes ressources matérielles.

On a utilisé ce diagramme de déploiement pour présenté l'architecteur logicielle et matérielle du système. Le diagramme est montré dans la figure 4.3 :

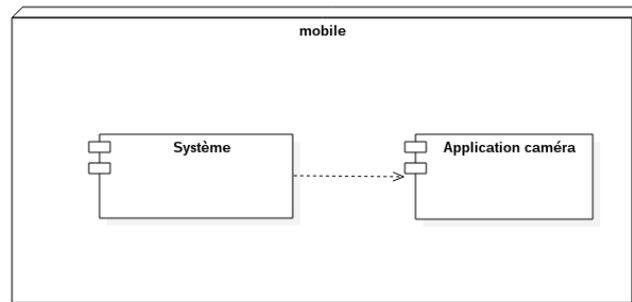


FIGURE 4.3 – Le diagramme de déploiement

## 5 Présentation de l'application

pour lancer l'application, clique sur « DZidScanner » 4.4.

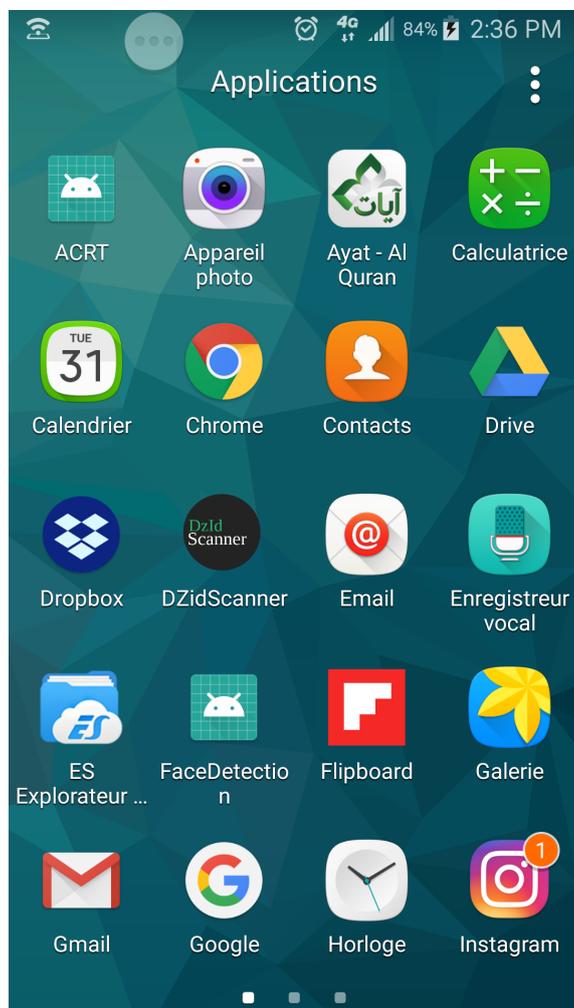
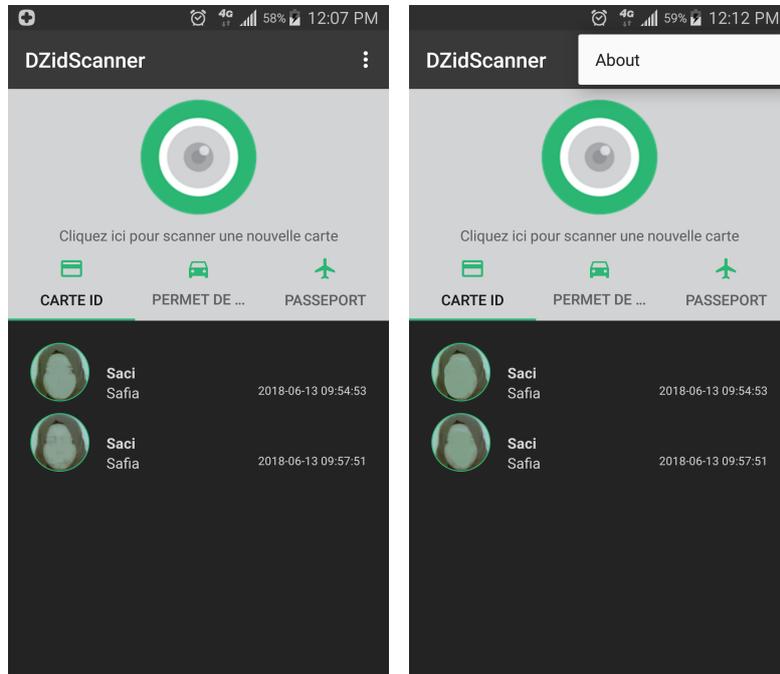


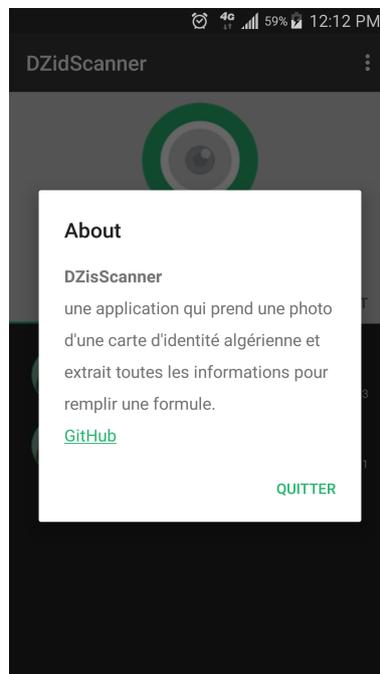
FIGURE 4.4 – Fenêtre du lancement d'application

Après l'ouverture de l'application la première page à afficher est la fenêtre principale. Cliquez sur ajouter pour scanner une nouvelle carte, après la caméra s'ouvre.



(a) l'activité main

(b) Le menu de l'activité main



(c) la biote de dialog about

FIGURE 4.5 – La fenêtre principale

Cliquez sur l'une des carte, une formulaire contient plus d'information s'affichera, la fenêtre formulaire est présentée dans la figure 4.6a.

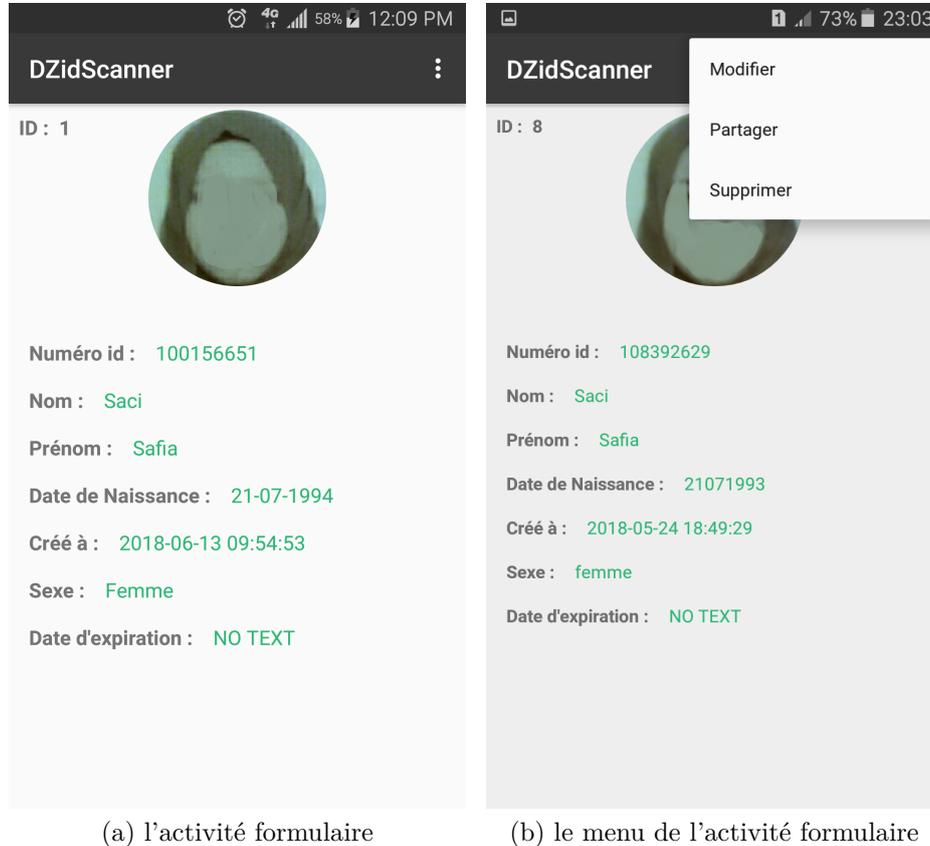
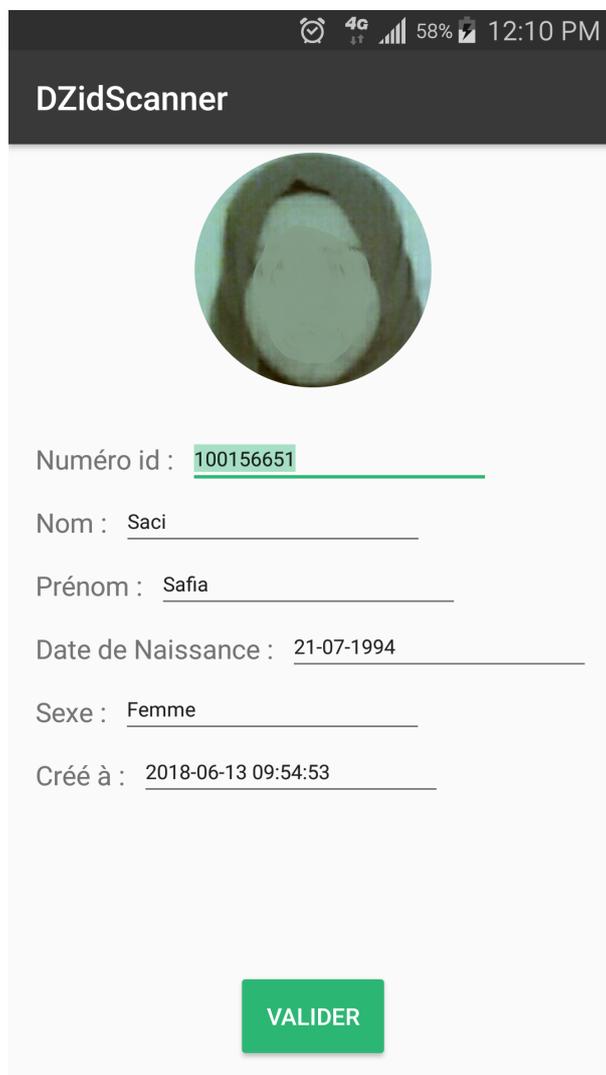


FIGURE 4.6 – La fenêtre formulaire

Si vous voulez supprimer, modifier ou partager une carte cliquez sur le menu et une liste de choix apparaîtra comme il est affiché dans la figure 4.6b

Si vous avez cliqué sur « modifier » la fenêtre 4.7 se réouvrira.



The screenshot shows the DZidScanner application interface. At the top, there is a status bar with icons for alarm, 4G network, signal strength, 58% battery, and the time 12:10 PM. Below the status bar is a dark header with the text "DZidScanner". The main content area features a circular profile picture of a person. Below the picture, there are several form fields with labels and values:

- Numéro id : 100156651
- Nom : Saci
- Prénom : Safia
- Date de Naissance : 21-07-1994
- Sexe : Femme
- Créé à : 2018-06-13 09:54:53

At the bottom center of the form is a green button labeled "VALIDER".

FIGURE 4.7 – fenêtre pour saisir les modifications

Après la terminaison des modifications cliquez sur valider et la fenêtre 4.6a se réouvrira.

Si vous avez cliqué sur « supprimer » une boîte de dialogue s'affichera pour confirmer la suppression comme il est indiqué dans la figure 4.8.

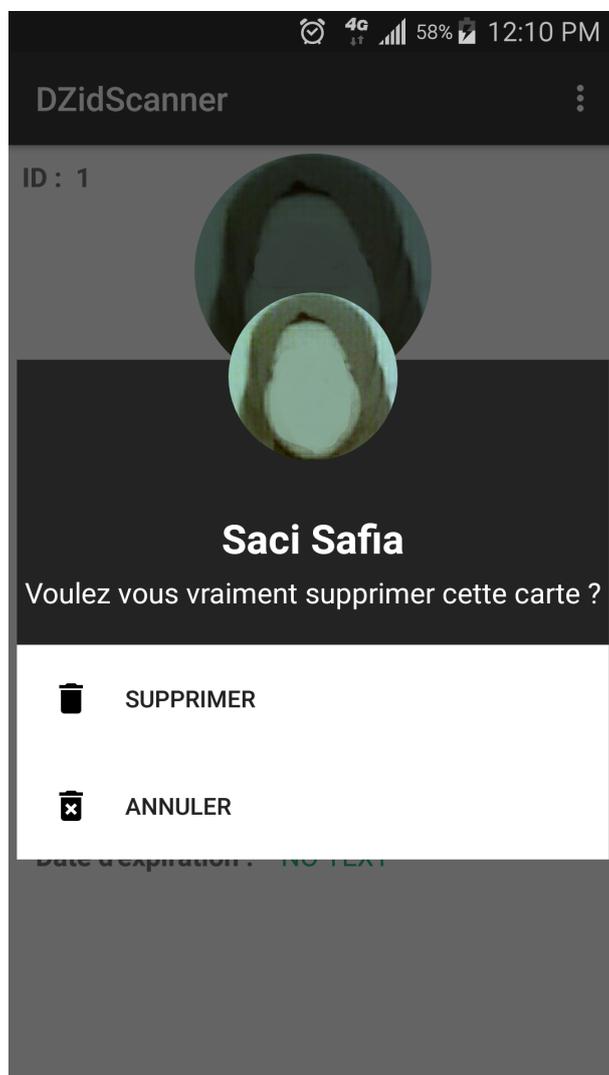


FIGURE 4.8 – Boîte de dialogue pour confirmer la suppression

Si vous avez cliqué sur « Supprimer » la carte sera supprimée et la fenêtre 4.5a s'ouvrira. Sinon si vous avez cliqué sur « Annuler » la boîte de dialogue sera fermée et la fenêtre 4.6a s'ouvrira.

Si vous avez cliqué sur « partager » une liste des applications de communication s'affichera pour choisir une, comme il est indiqué dans la figure 4.9.

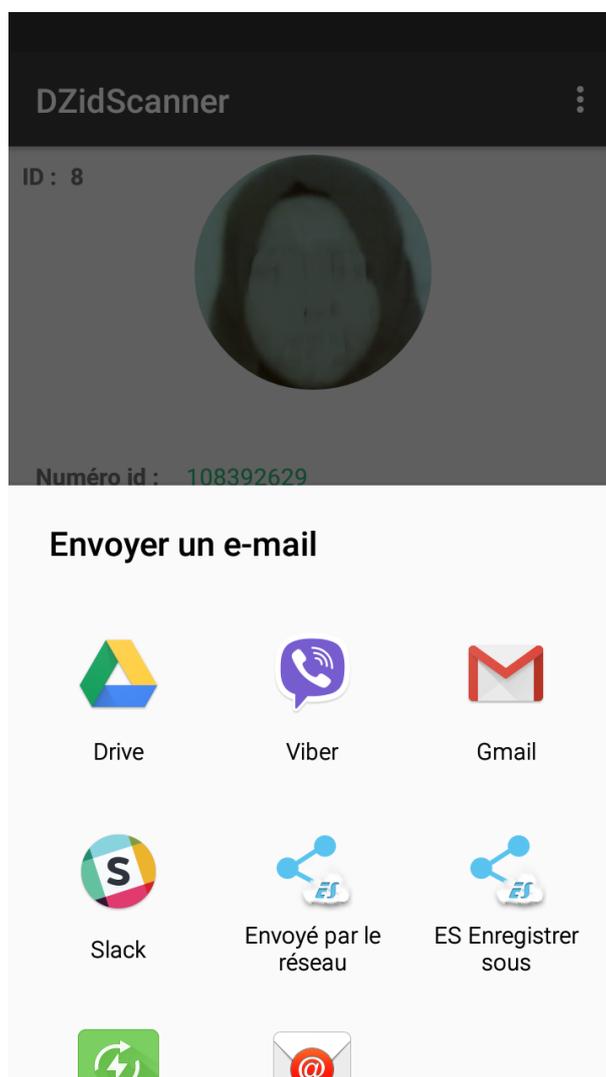


FIGURE 4.9 – Liste des application pour partager la carte.

## 6 Conclusion

Ce chapitre a été consacré à l'implémentation de l'application après avoir présenté l'environnement et les outils de développement, les captures d'écran accompagnées des explications illustrant son fonctionnement.

# Conclusion Générale

# Conclusion Générale

Le présent mémoire est le résultat de notre travail que nous avons effectué dans le cadre de la réalisation de notre projet de fin d'études du Master en Réseaux et Télécommunication. Lors de ce travail nous avons pu mettre en pratique notre connaissances théoriques acquises durant notre parcours universitaire, de plus, nous avons les enrichi par des nouvelles reconnaissances telles que l'OCR, Andoid et un nouveau langage de programmation java.

L'objectif de notre travail a consisté principalement à concevoir et à réaliser une application Android qui permet de prendre une photo d'une carte d'identité, un permis de conduire ou un passeport algérien et récupérer les informations fournies par la pièce d'identité et les rendre dans un format modifiable et exploitable afin d'accélérer le traitement des dossiers ,le pointage dans les sociétés et dans plusieurs contextes.

Nous avons, tout d'abord , entamé notre étude par la spécification des besoins qui est une étape cruciale et nécessaire pour mieux assimiler les techniques utilisées dans la réalisation de ce projet, puis par la définition des principaux intervenants et l'identification des besoins. Ensuite nous avons procédé à l'analyse qui a permis la conception d'une architecture de base stable, qui est le point de départ à la conception dans laquelle nous avons utilisé UML comme langage de modélisation. Enfin, l'implémentation, qui nous a permis de développer notre application en tenant compte de l'environnement de travail logiciel et matériel.

Notre application n'a pas atteint son état final, elle ne scanne que la carte d'identité pour l'instant alors comme perspective :

On estime de scanner les autres pièces d'identité telle que le passeport et le permis de conduire.

# Bibliographie

- [1] Sukhpreet Singh. Optical character recognition techniques : a survey. *Journal of emerging Trends in Computing and information Sciences*, 4(6) :545–550, 2013.
- [2] Line EIKVIL. Ocr, optical character recognition. *Norsk Regnesentral*, December 1993.
- [3] M. Cheriet, N. Kharma, C.L. Liu, and C. Suen. *Character Recognition Systems : A Guide for Students and Practitioners*. Wiley, 2007.
- [4] Fadoua ATAA ALLAH Khadija EL GAJOUJ. Vers un système de reconnaissance optique des caractères dans des documents multilingues : Français-amazighe. *International workshop IHMIM*, Mai 2014.
- [5] Andrew Young. *Handbook of Pattern Recognition and Image Processing*. Academic Press, 1986.
- [6] Preeti Verma Abhishek Verma, Suket Arora. Ocr-optical character recognition. In *7th International Conference on Recent Innovation in Science*, pages 230,240, September 2016.
- [7] Quentin Savary. Fonctionnement de l’ocr. 1ère partie la segmentation, 2016. [www.whatsupdata.fr/content/fonctionnement-de-locr-2%C3%A8me-partie-la-reconnaissance](http://www.whatsupdata.fr/content/fonctionnement-de-locr-2%C3%A8me-partie-la-reconnaissance), consulter le 2018-05-01.
- [8] Stephen V. Rice George Nagy, Thomas A. Nartker. Optical character recognition : An illustrated guide to the frontier. *Kluwer Academic Publishers*, 1999.
- [9] A Chaudhuri. Some experiments on optical character recognition systems for different languages using soft computing techniques. Technical report, Birla Institute of Technology Mesra, 2010.
- [10] A. Chaudhuri, K. Mandaviya, P. Badelia, and S.K. Ghosh. *Optical Character Recognition Systems for Different Languages with Soft Computing*. Studies in Fuzziness and Soft Computing. Springer International Publishing, 2017.
- [11] Stefan Agne, Markus Rogger, and Jörg Rohrschneider. Benchmarking of document page segmentation. In *Document Recognition and Retrieval VII*, volume 3967, pages 165–172. International Society for Optics and Photonics, 1999.
- [12] Abdel Belaïd and Hubert Cecotti. Reconnaissance de caractères : évaluation des performances, 2006.

- 
- [13] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of ocr applications. *International Journal of Machine Learning and Computing*, 2(3) :314, 2012.
- [14] Robert Frischholz. Face detection. *Online Resources*. (<http://www.facedetection.com/>), 2005.
- [15] Michael B Lewis and Hadyn D Ellis. How we detect a face : A survey of psychological evidence. *International Journal of Imaging Systems and Technology*, 13(1) :3–7, 2003.
- [16] Bastian Leibe, Aleš Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International journal of computer vision*, 77(1-3) :259–289, 2008.
- [17] W Karam, C Mokbel, H Greige, B Pesquet-Popescu, and G Chollet. Un système de détection de visage et d'extraction de paramètres basé sur les svm et des contraintes. CORESA, 2004.
- [18] Gerasimos Potamianos, Chalapathy Neti, Juergen Luetttin, and Iain Matthews. Audio-visual automatic speech recognition : An overview. *Issues in visual and audio-visual speech processing*, 22 :23, 2004.
- [19] Moad Benkiniouar and Mohamed Benmohamed. Méthodes d'identification et de reconnaissance de visages en temps réel basées sur adaboost. *Journée d'informatique graphique Biskra*, 2005.
- [20] Alberto Menache. *Understanding motion capture for computer animation and video games*. Morgan kaufmann, 2000.
- [21] Anil K Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1) :4–20, 2004.
- [22] Gurvinder Singh Shergill, Olaf Diegel, Abdolhossein Sarrafzadeh, and Aruna Shekar. Computerized sales assistants : The application of computer technology to measure consumer interest-a conceptual framework. *Journal of Electronic Commerce Research*, 9(2) :176, 2008.
- [23] Yves Wautelet, Laurent Louvigny, and Manuel Kolp. Le unified process comme méthodologie de gestion de projet informatique. éléments d'application en milieu sidérurgique. Technical report, Working Paper IAG 109/04, Université Catholique de Louvain, 2004.
- [24] Pierre-Alain Muller and Nathalie Gaertner. *Modélisation objet avec UML*, volume 514. Eyrolles Paris, 2000.
- [25] Pascal Roques and Franck Vallée. *UML 2 en action : de l'analyse des besoins à la conception*. Editions Eyrolles, 2011.

- [26] VALÉRY Bévo, Ghislain Lévesque, and Alain Abran. Application de la méthode ffp a partir d'une spécification selon la notation uml : Compte rendu des premiers essais d'application et questions. In *9th International Workshop Software Measurement, Lac Supérieur, Canada*, 1999.
- [27] Pascal Roques. *UML 2.5 par la pratique : Etudes de cas et exercices corrigés*. Editions Eyrolles, 2018.
- [28] Olivier Glassey and Jean-Loup Chappelet. *Comparaison de trois techniques de modélisation de processus : ADONIS, OSSAD et UML*. IDHEAP, Institut de hautes études en administration publique, 2002.
- [29] Les systèmes d'exploitation des smartphones, 2018. [www.monpetitmobile.com/choisir-mobile/systemes-exploitation-smartphones](http://www.monpetitmobile.com/choisir-mobile/systemes-exploitation-smartphones), consulter le 03-05-2018.
- [30] BOURAOUI Mohamed. *Les Systèmes d'exploitation mobile*. ISET, Sep 2016.
- [31] Site officiel de android, 2018. [www.android.com](http://www.android.com), consulter le 03-05-2018.
- [32] Rick Rogers, John Lombardo, Zigurd Mednieks, and Blake Meike. *Android application development : Programming with the Google SDK*. O'Reilly Media, Inc., 2009.
- [33] Site officiel de open handset alliance, 2018. [www.openhandsetalliance.com](http://www.openhandsetalliance.com), consulter le 09-03-2018.
- [34] Android central - android's early days, 2018. [www.androidcentral.com/androids-early-days](http://www.androidcentral.com/androids-early-days), consulter le 05-03-2018.
- [35] Itpro - google claims 65,000 android phones shipping daily, mai 2012. [www.itpro.co.uk/623329/google-claims-65000-android-phones-shipping-daily](http://www.itpro.co.uk/623329/google-claims-65000-android-phones-shipping-daily), consulter le 05-03-2018.
- [36] Idc - apple beats microsoft at their own game while amazon primes the low end of the tablet market, according to idc, Avril 2016. [www.idc.com/getdoc.jsp?containerId=prUS41218816](http://www.idc.com/getdoc.jsp?containerId=prUS41218816), dernier accès le 05-03-2018.
- [37] The app maker blog - how long does it take for your app to be approved ?, Août 2014. <https://www.appmakr.com/blog/how-long-app-approved/>, consulter le 05-03-2018.
- [38] Facebook newsroom - introducing home, Avril 2013. <http://newsroom.fb.com/news/2013/04/introducing-home/>, consulter le 05-03-2018.
- [39] Platform architecture, 2018. [www.developer.android.com](http://www.developer.android.com), consulter le 09-03-2018.

# Annexes

# Android

## Les système OS mobile

### Définition

Les systèmes d'exploitation mobiles (OS) peuvent être définis comme les logiciels permettant à un smartphone ou un téléphone mobile basique de fonctionner. Ils permettent de ce fait aux utilisateurs de pouvoir passer un appel téléphonique, naviguer sur leurs téléphones parmi toutes les rubriques, télécharger des applications ou encore paramétrer et personnaliser leurs smartphones [29].

En d'autres termes un OS mobile c'est l'ensemble des programmes responsables de la gestion des opérations, du contrôle, de la coordination, de l'utilisation du matériel et de partage des ressources d'un dispositif entre divers programmes tournant sur ce dispositif [30].

### Les principaux systèmes d'exploitation mobiles

On distingue une vingtaine de système d'exploitation sur le marché, certains étant développés directement par et pour les fabricants de téléphones eux-mêmes, d'autres formant des conglomérats pour mutualiser les coûts et effectuer des synergies, et d'autres souhaitant uniquement se focaliser sur le développement de systèmes d'exploitation[29]. Voici les systèmes d'exploitation les plus populaires :

#### Android de Google

Depuis maintenant de nombreuses années, Android, développé par les ingénieurs de Google, s'impose comme l'OS de référence du marché, loin devant ses concurrents. Il atteint ainsi en 2014 le milliard de smartphones équipés, quand iOS à la seconde place ne dépasse pas les 180 millions. L'avantage de ce système d'exploitation se résume dans le fait qu'il est mis à disposition de tous les fabricants de téléphones, à condition qu'ils respectent certaines conditions liés à la mise en avant de services Google. Pour le reste, l'OS est personnalisable à souhait, par les consommateurs comme par les fabricants. D'après les statistique de StatCounter<sup>6</sup> Android est le

---

6. Statcounter est un outil d'analyse du trafic Web.

premier OS mobile dans le monde ainsi que dans le marché algérien, comme le démontre les graphique dans les figures 1 2 , de nos jours, la majorité des téléphones vendus dans le monde et l'Algérie tournent sous Android. On peut ainsi citer de très nombreuses marques de smartphones, parmi lesquelles Samsung, Acer, HTC, LG, Sony ou encore Huawei [29].

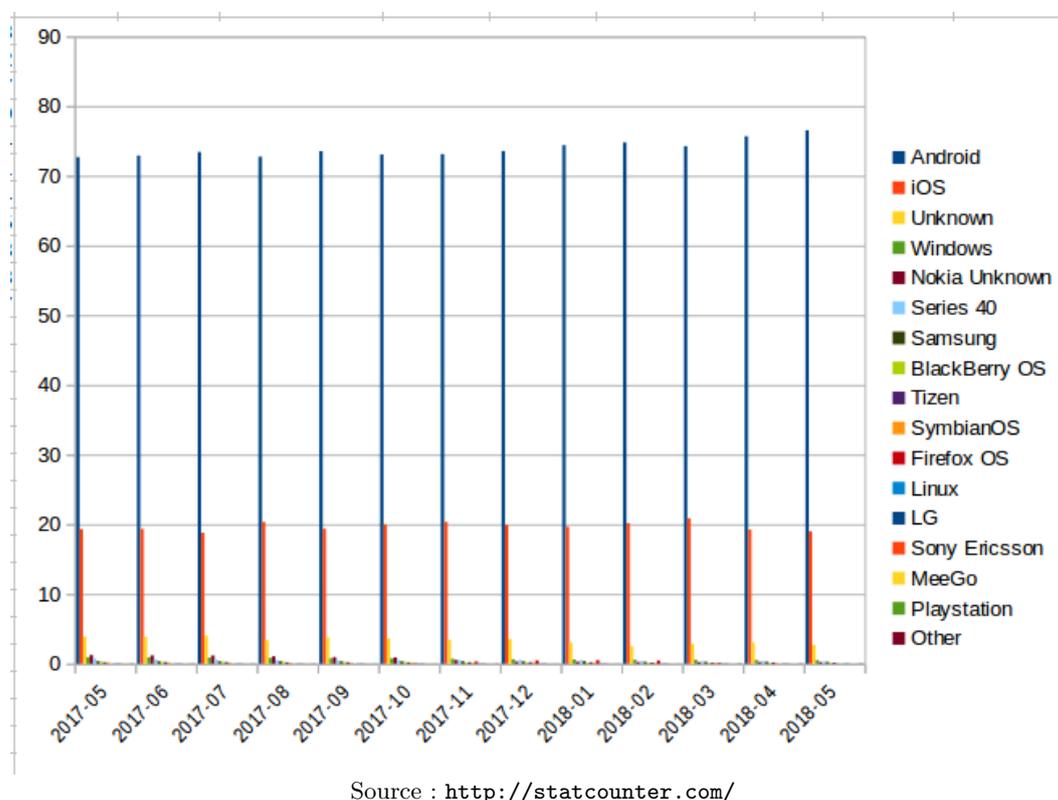
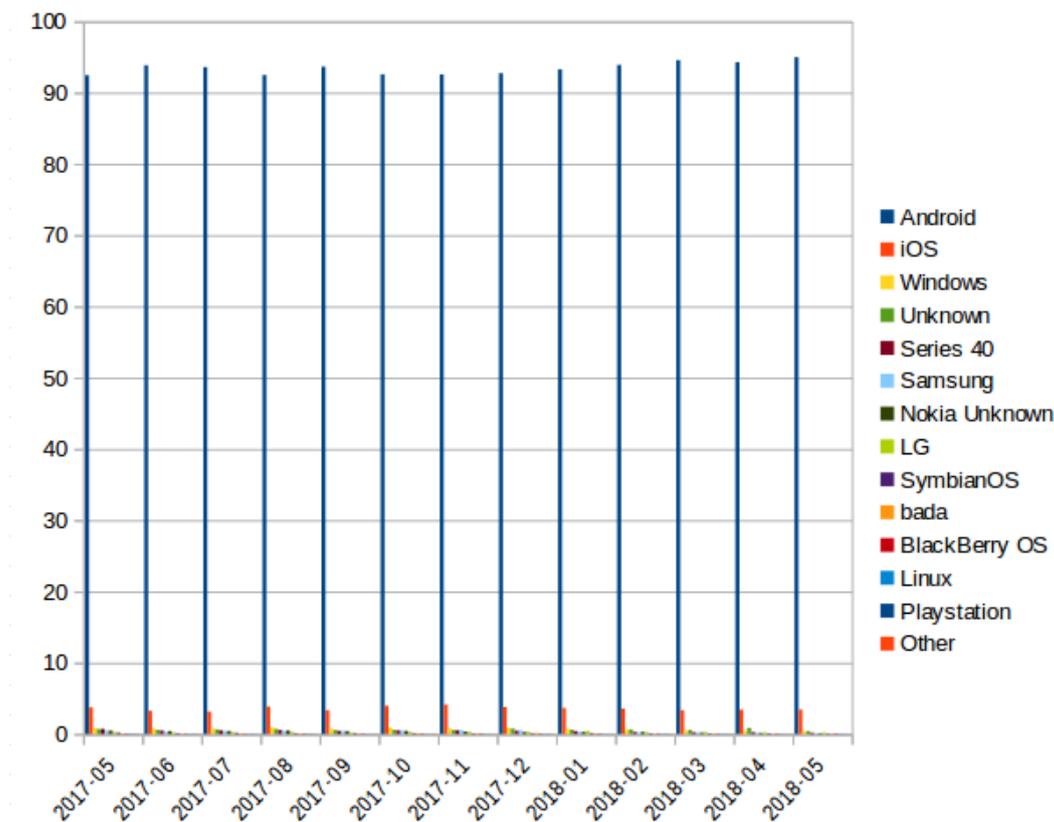


FIGURE 1 – Part de marché mondiale des OS mobiles.

## IOS de Appel

Avec Android, iOS de l'entreprise américaine Apple concentre une très large majorité des smartphones de la planète, les autres que nous vous présentons plus bas étant loin de les égaler. L'iOS d'Apple est connu et présenté par la firme de Cupertino comme étant le système d'exploitation proposant la meilleure expérience utilisateur. Avec son flat design apparu à partir de l'iOS 7, Apple propose son logiciel uniquement sur ses propres produits. Il faut de ce fait passer par l'Apple Store pour installer sur son iPhone ou iPad les applications qui vous plaisent, parmi les centaines de milliers existantes. L'écosystème d'Apple rendant tous les produits de la marque compatibles entre eux, permet une ergonomie inégalée à ce jour, positionnant Apple comme fabricant éminent, et comme second plus gros système d'exploitation au monde en volume [29].



Source : <http://statcounter.com/>

FIGURE 2 – Part de marché algérien des OS mobiles.

### Windows Phone de Microsoft

Smartphones tournant sous Windows Phone restent à ce jour limités face à Android, bien que toutefois présents en nombre. Microsoft, qui a pris le virage des smartphones en retard, souffre de son manque d'adoption par les constructeurs et consommateurs. Basé sur le design de Windows 8 pour PC, il n'est pas du goût de tout le monde. Il reste cependant aimé par les utilisateurs pour la possibilité de personnaliser de manière poussée la page d'accueil et sa simplicité de gestion des applis. Face aux géants que sont iOS et Android, il a toutefois de la peine à intéresser suffisamment les développeurs d'applications, certains d'entre eux préférant le délaissier au profit des OS beaucoup plus fréquentés, au grand déplaisir des utilisateurs [29].

### BlackBerry de RIM

Ancienne star de la téléphonie, prisé par le monde professionnel et particulièrement des cadres d'entreprise, l'OS Blackberry de RIM<sup>7</sup> a perdu de ses parts de marché depuis la dernière décennie. Il n'empêche que les smartphones de la marque,

7. **Research In Motion** a Canadian multinational company specializing in enterprise software and the Internet of things.

avec leur design complètement revus, intéressent maintenant une partie ciblée des consommateurs. A l'attention des personnes avides de claviers physiques, Blackberry OS est tourné avant tout vers la lecture des e-mails et SMS.

Fait identifiable, les applications Android fonctionnent maintenant sous Blackberry OS, rendant plus attrayant le système d'exploitation de RIM, en proposant un très large catalogue d'applications compatibles innovantes[29].

## Pourquoi Android ?

*“La plate-forme ‘Android’ permet aux fabricants d’appareils d’innover et d’être plus compétitifs. Les développeurs d’applications peuvent toucher une audience plus vaste et consolider leur activité, tandis que les consommateurs disposent d’un choix d’appareils sans précédent, à des prix plus bas que jamais.”*

-Hiroshi Lockheimer, Vice-Président senior, Android, Chrome OS et Play

**accès plus facile à davantage de technologies dans plus de pays** Le code source d'Android est accessible librement à quiconque souhaite le télécharger, le personnaliser et le distribuer. Cela permet aux fabricants de créer des appareils mobiles à moindre coût, et les utilisateurs du monde entier ont ainsi accès à une technologie mobile jusqu'ici hors de portée[31].

**La liberté de choisir :** Le modèle Open Source d'Android stimule l'innovation en offrant aux fabricants d'appareils la liberté de personnaliser leurs téléphones et le système d'exploitation Android, et de préinstaller les applications de leur choix. Ainsi, les consommateurs disposent d'un plus grand choix en matière d'appareils et d'applications [31].

**Un moteur de réussite économique :** D'ici 2020, l'écosystème mobile devrait compter environ 15 millions d'employés à travers le monde. Android dynamise cette évolution en aidant les développeurs et les fabricants à créer encore plus d'emplois [31].

**Favoriser l'accès universel à l'information** Android permet de produire des appareils à prix abordables pour les marchés émergents et de donner accès à des milliers d'applications liées à l'éducation et à la santé. Il aide plus de personnes à accéder à l'information et à saisir des opportunités que n'importe quelle autre plate-forme mobile [31].

## Présentation de la plate-fomre Android

Android c'est une plateforme complète pour appareil mobile[32], lancé par Google et l'OHA<sup>8</sup> qui compose d'opérateurs mobiles, de fabricants de téléphones et d'éditeurs logiciels [33]. La plateforme Android est un Système d'exploitation basée sur le noyan linux, entièrement gratuit, sous licence open source. Elle est composée d'un système d'exploitation, de librairies,et d'un ensemble d'applications. Rappelons les



FIGURE 3 – Logo Android.

points clés d'Android en tant que plate-forme :

- Elle est innovante car toutes les dernières technologies de téléphonie y sont Intégrées : écran tactile, accéléromètre, GPS, appareil photo numérique etc.
- elle est accessible car en tant que développeur vous n'avez pas à acheter de matériel spécifique (si vous voulez aller plus loin que l'utilisation d'un émulateur, un téléphone Android pour effectuer vos tests vous sera toutefois nécessaire), ni à connaître un langage peu utilisé ou spécifique : le développement sur la plateforme Android est en effet réalisé en langage Java, un des langages de programmation les plus répandus.
- elle est ouverte parce que la plate-forme Android est fournie sous licence open source, permettant à tous les développeurs – et constructeurs – de consulter les sources et d'effectuer les modifications qu'ils souhaitent.

---

8. Open Handset Alliance : C'est un consortium de 84 entreprises telles que Google, Samsung, AKM, synaptics, KDDI, Garmin, Teleca, Ebay, Intel...etc. Il a été créé le 5 novembre 2007, dirigé par Google. Il s'engage à faire progresser les normes ouvertes, à fournir des services et à déployer des combinés à l'aide de la plateforme Android.

## Historique

### **2006 Les fabricants de smartphones n'ont qu'un choix limité**

Les fabricants qui souhaitent créer un smartphone n'ont que deux options coûteuses : Acheter la licence d'un système d'exploitation ou développer leur propre OS [31].

### **2007 Android s'établit en tant que plate-forme Open Source**

Google contribue à la formation de l'Open Handset Alliance et établit Android comme une plate-forme Open Source que quiconque peut télécharger, modifier et installer gratuitement, sur n'importe quel appareil mobile [31].

### **2008 Lancement d'Android Cupcake**

La mise à jour Cupcake est appliquée à Android. Elle permet à des fabricants d'appareils, tels que HTC et Samsung, et à des opérateurs tels que T-Mobile et Deutsche Telekom de commencer à personnaliser l'interface de leurs téléphones [34].

### **2009 D'autres plates-formes basées sur Android émergent**

Lancement de CyanogenMod : Le premier système d'exploitation alternatif créé à partir du code source gratuit d'Android. Les plates-formes fermées ne se différencient pas encore de façon flagrante [31].

### **2010 Android permet de diversifier le marché du smartphone**

Au cours d'une réunion avec ses actionnaires, Google annonce qu'Android est utilisé sur 34 types d'appareils mobiles à travers 49 pays, offrant ainsi aux utilisateurs de smartphones un choix d'appareils plus vaste que jamais [35].

### **2011 Android arrive sur les tablettes**

Lancement de la version Android 3.0 Honeycomb, dont le design est adapté aux tablettes. Peu de temps après, Amazon lance la tablette Kindle Fire, qui fonctionne sous le système d'exploitation Fire, basé sur Android. Au cours du premier trimestre 2016, Amazon devient le troisième plus grand fournisseur de tablettes au monde [36].

### **2012 Lancement de Google Play**

Android Market a changé de nom et s'appelle désormais Google Play. Tout comme Android Market, Google Play permet aux développeurs de distribuer leurs applications en quelques heures, contrairement aux plates-formes fermées qui peuvent mettre plusieurs jours à approuver une seule application [37].

### **2013 Facebook crée sa propre expérience Android**

Facebook lance Facebook Home, un lanceur d'applications personnalisé pour Android, basé sur les applications Facebook. Avec cet outil, le flux d'actualités Facebook devient l'écran d'accueil sur certains appareils Android [38].

**2014 Android se développe au-delà des appareils mobiles**

Audi, GM, Google, Honda, Hyundai et NVIDIA présentent l'Open Automotive Alliance (OAA), qui a pour objectif d'intégrer la plate-forme Android aux automobiles. Plus de 45 grandes marques automobiles sont aujourd'hui membres de ce groupe [31].

**2015 Les smartphones deviennent plus accessibles que jamais**

Grâce à l'innovation de certains fabricants, de plus en plus d'appareils Android à moins de 50 \$ sont disponibles. Il n'existe toujours pas de smartphones aussi abordables sur les plates-formes fermées.

**2016 Une décennie d'innovations pour un choix toujours plus vaste**

À ce jour, près de 1 300 marques ont développé plus de 24 000 appareils Android. Entre 2015 et 2016, les fabricants ont lancé plus de 600 appareils. Il existe aujourd'hui des dizaines de plates-formes mondiales permettant de télécharger des applications Android, et bien d'autres encore au niveau local.

## Fonctionnalités d'Android

Android dispose d'un grand nombre de fonctionnalités qui facilitent la communication avec l'appareil et rendent son utilisation agréable et facile, on rappelle les plus importants [30] :

- Framework d'application
- Machine virtuelle Dalvik
- Navigateur web intégré
- API graphique 2D et 3D
- Codecs audio et vidéo
- WiFi, EDGE, 3G, LTE, Bluetooth...
- Camera, GPS.

## Les versions d'Android :

L'historique des versions d'Android a débuté avec la sortie de la version 1.0 en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités, le tableau 1 résume quelques versions de l'android :

## Architecture Android

Android est une pile logicielle open-source basée sur Linux, créée pour un large éventail d'appareils et de facteurs de forme. Le diagramme dans la figure 4 montre

Nom	Version	API	Date de sortie
Sans Nom	1.0	1	23 septembre 2008
Petit Four	1.1	2	9 février 2009
Cupcake	1.5	3	27 avril 2009
Donut	1.6	4	15 septembre 2009
Eclair	2.0 – 2.1	5-7	26 octobre 2009
Froyo	2.2 – 2.2.3	8	20 mai 2010
Gingerbread	2.3 – 2.3.7	9-10	6 décembre 2010
Honeycomb	3.0 – 3.2.6	11 – 13	22 février 2011
Ice Cream Sandwich	4.0 – 4.0.4	14 – 15	18 octobre 2011
Jelly Bean	4.1 – 4.3.1	16 – 18	9 juillet 2012
KitKat	4.4 – 4.4.4	19 – 20	31 octobre 2013
Lollipop	5.0 – 5.1.1	21 – 22	12 novembre 2014
Marshmallow	6.0 – 6.0.1	23	05 octobre 2015
Nougat	7.0 – 7.1.2	24 – 25	22 août 2016
Oreo	8.0 – 8.1	26 – 27	21 août 2017
Android P	9	28	

TABLE 1 – Version d’android

les principaux composants de la plateforme Android [39].

## Le noyau Linux

La base de la plate-forme Android est le noyau Linux. Par exemple, Android Runtime (ART) s’appuie sur le noyau Linux pour les fonctionnalités sous-jacentes telles que le threading et la gestion de la mémoire de bas niveau.

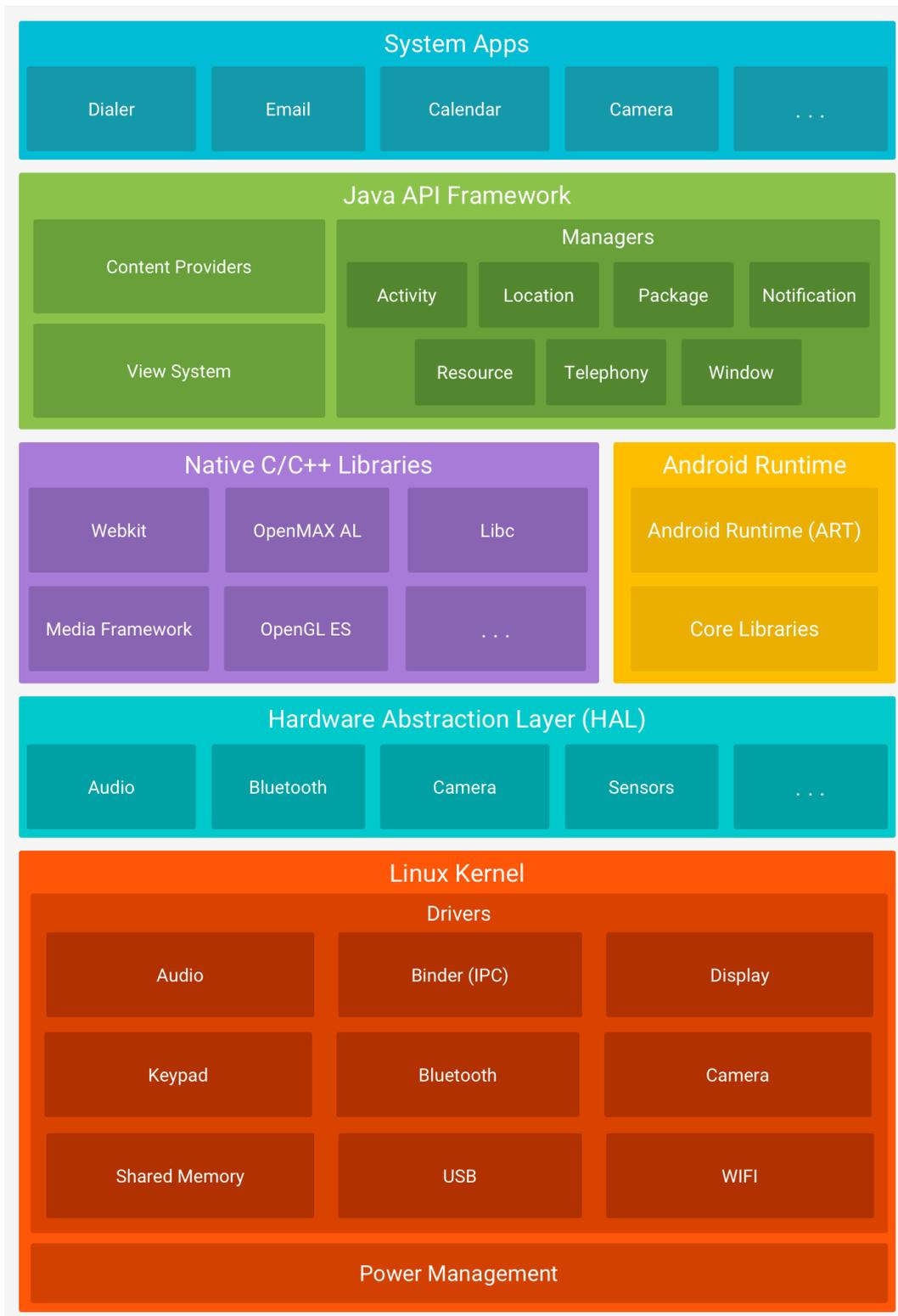
L’utilisation d’un noyau Linux permet à Android de tirer parti des principales fonctions de sécurité et permet aux fabricants de périphériques de développer des pilotes matériels pour un noyau bien connu.

## Couche d’abstraction matérielle

La couche d’abstraction matérielle (HAL<sup>9</sup>) fournit des interfaces standard qui exposent les capacités matérielles du périphérique à l’infrastructure d’API<sup>10</sup> Java de niveau supérieur. Le HAL se compose de plusieurs modules de bibliothèque, chacun d’entre eux mettant en œuvre une interface pour un type spécifique de composant matériel, tel que le module caméra ou bluetooth. Lorsqu’une API de structure effectue un appel pour accéder au matériel de l’appareil, le système Android charge le module de bibliothèque pour ce composant matériel.

9. Hardware Abstraction Layer : en français couche d’abstraction matérielle

10. Application programming interface : en français Interface de programmation d’applications



Source : <http://android.com/>

FIGURE 4 – L’architecture Android.

## Android Runtime

Pour les appareils exécutant Android version 5.0 (niveau d'API 21) ou supérieur, chaque application s'exécute dans son propre processus et avec sa propre instance de l'Android Runtime (ART). ART est conçu pour exécuter plusieurs machines virtuelles sur des périphériques à faible mémoire en exécutant des fichiers DEX, un format de bytecode spécialement conçu pour Android, optimisé pour un encombrement mémoire minimal. Construire des chaînes d'outils, telles que Jack<sup>11</sup>, compiler des sources Java dans le bytecode DEX, qui peut fonctionner sur la plate-forme Android.

Certaines des principales caractéristiques de l'ART sont les suivants :

- Compilation AOT (ahead-of-time) et JIT (just-in-time).
- Garbage optimisé (GC).
- Meilleure prise en charge du débogage, notamment un profileur d'échantillonnage dédié, des exceptions de diagnostic détaillées et des rapports d'erreur, ainsi que la possibilité de définir des points de surveillance pour surveiller des champs spécifiques.

Avant Android version 5.0 (niveau API 21), Dalvik était le moteur d'exécution Android. Si votre application fonctionne bien sur ART, cela devrait fonctionner également avec Dalvik, mais l'inverse n'est peut-être pas vrai.

Android inclut également un ensemble de bibliothèques d'exécution principales qui fournissent la plupart des fonctionnalités du langage de programmation Java, y compris certaines fonctionnalités de langage Java 8, que la structure de l'API Java utilise.

## Bibliothèques C / C ++ natives

De nombreux composants et services système Android de base, tels que ART et HAL, sont construits à partir de code natif qui nécessite des bibliothèques natives écrites en C et C ++. La plate-forme Android fournit des API de structure Java pour exposer les fonctionnalités de certaines de ces bibliothèques natives aux applications. Par exemple, vous pouvez accéder à OpenGL ES via l'API Java OpenGL du framework Android pour ajouter du support pour dessiner et manipuler des graphiques 2D et 3D dans votre application.

Si vous développez une application nécessitant du code C ou C ++, vous pouvez utiliser Android NDK pour accéder à certaines de ces bibliothèques de plates-formes natives directement à partir de votre code natif.

## Java API Framework

L'ensemble complet des fonctionnalités du système d'exploitation Android est disponible pour vous à travers des API écrites en langage Java. Ces APIs constituent

---

11. Jack est une chaîne d'outils Android qui compile la source Java en Android bytecode.

les éléments de base dont vous avez besoin pour créer des applications Android en simplifiant la réutilisation du noyau, composants et services du système modulaire, qui comprennent les éléments suivants :

- Un système de visualisation riche et extensible que vous pouvez utiliser pour créer l'interface utilisateur d'une application, y compris des listes, des grilles, des zones de texte, des boutons et même un navigateur Web intégrable.
- Un gestionnaire de ressources, fournissant un accès à des ressources non-code telles que des chaînes de caractères localisées, des graphiques et des fichiers de mise en page.
- Un gestionnaire de notifications qui permet à toutes les applications d'afficher des alertes personnalisées dans la barre d'état.
- Un gestionnaire d'activité qui gère le cycle de vie des applications et fournit une pile de navigation commune.
- Les pourvoyeurs de contenu qui permettent aux applications d'accéder aux données d'autres applications, telles que l'application Contacts, ou de partager leurs propres données.

Les développeurs ont un accès complet aux mêmes API de structure que les applications système Android.

## Les Application

Android est livré avec un ensemble d'applications de base pour le courrier électronique, la messagerie SMS, les calendriers, la navigation sur Internet, les contacts, et plus encore. Les applications incluses avec la plateforme n'ont aucun statut particulier parmi les applications que l'utilisateur choisit d'installer. Ainsi, une application tierce peut devenir le navigateur Web par défaut, le service de messagerie SMS ou même le clavier par défaut de l'utilisateur (certaines exceptions s'appliquent, telles que l'application Paramètres du système).

Les applications système fonctionnent à la fois comme des applications pour les utilisateurs et pour fournir des fonctionnalités clés auxquelles les développeurs peuvent accéder depuis leur propre application. Par exemple, si votre application souhaite envoyer un message SMS, vous n'avez pas besoin de créer cette fonctionnalité vous-même. Vous pouvez à la place appeler n'importe quelle application SMS déjà installée pour envoyer un message au destinataire que vous spécifiez.