



Université Tunis El Manar
École Nationale d'Ingénieurs
de Tunis



Département des Technologies de l'Information et de la
Communication

Rapport de projet de fin d'année II

Implémentation de la Reconnaissance Faciale dans un Environnement de Stockage Cloud Sécurisé

Réalisé par

Mohamed Aziz KTATA

Sami MASMOUDI

Classe

2ème année Informatique

Supervisé par : **Dr. Hamza HAMMAMI**

2023 - 2024

Résumé

Ce rapport expose la conception et le développement d'une plateforme de sauvegarde de fichiers, fruit de notre projet de fin d'année à l'École Nationale d'Ingénieurs de Tunis (ENIT). Notre solution offre une réponse sûre et pratique pour le stockage des données numériques dans le cloud. Intégrant des fonctionnalités avancées telles que le chiffrement des données et la reconnaissance faciale, notre plateforme garantit la confidentialité et la sécurité des informations des utilisateurs. L'objectif principal de ce projet est de répondre aux exigences croissantes en matière de stockage et de protection des données, tout en préservant l'intégrité et l'accessibilité des données utilisateur. Ce rapport présente une vue d'ensemble détaillée du processus de conception, de développement et d'implémentation de notre plateforme, ainsi que les résultats obtenus et les perspectives futures.

Mots clés : Stockage numérique, Cloud, Reconnaissance Faciale, Réseau de neurones, Deep Learning, Chiffrement de données

Abstract

Dans le cadre de notre projet de fin d'année à l'École Nationale d'Ingénieurs de Tunis (ENIT), nous avons développé une plateforme de sauvegarde de fichiers offrant une solution sécurisée et accessible pour stocker les données numériques dans le cloud. Cette plateforme intègre des fonctionnalités avancées telles que le cryptage des données pour assurer la confidentialité des informations et la reconnaissance faciale pour une authentification sécurisée lors de l'accès aux fichiers. Notre objectif principal est de garantir la disponibilité, l'intégrité et la sécurité des données des utilisateurs, répondant ainsi aux besoins croissants en matière de stockage et de protection des données dans un environnement numérique en constante évolution.

Remerciements

Nous souhaitons tout d’abord exprimer notre profonde gratitude à toute l’équipe pédagogique de l’École Nationale d’Ingénieurs de Tunis (ENIT) pour nous avoir donné l’opportunité de réaliser ce projet.

Nous tenons également à remercier sincèrement Dr. Hamza Hammami pour son encadrement attentif et ses précieux conseils tout au long de notre projet. Sa guidance nous a été d’une aide inestimable dans notre apprentissage et notre progression académique.

Nous souhaitons également adresser nos remerciements à toutes les personnes qui ont contribué à la réussite de notre projet, que ce soit par leur soutien, leurs conseils ou leur expertise. Leur engagement a été essentiel pour le développement de nos compétences et la concrétisation de ce travail.

Nous sommes profondément reconnaissants envers chacune de ces personnes pour leur contribution à cette expérience enrichissante et pour l’opportunité qu’elles nous ont offerte.

Table des matières

Remerciements	iv
Introduction Générale	1
1 Cadre Général du Projet	2
1 Introduction	3
2 Cadre du Projet	3
3 Étude et critique de l'existant	3
3.1 Étude l'existant	4
3.2 Critique de l'existant	5
4 Solution proposée	6
4.1 Diagramme de contexte	6
5 Conclusion	7
2 Étude Théorique	8

1	Introduction	9
2	La reconnaissance faciale	9
2.1	Les étapes de reconnaissance faciale	9
2.2	Conclusion	15
3	Le chiffrement	16
3.1	Les types de chiffrement	16
3.1.1	Chiffrement à clés symétriques	16
3.1.2	Chiffrement à clés asymétrique	17
3.2	Algorithmes de chiffrement courants	18
3.3	Avantages du chiffrement	19
3.4	Inconvénients du chiffrement	19
3.5	Conclusion	20
4	Conclusion	20
3	Étude Conceptuelle	21
1	Introduction	22
2	Spécification des besoins	22
2.1	Besoins fonctionnels	22
2.1.1	Pour l'administrateur :	22
2.1.2	Pour l'utilisateur :	22
2.2	Les besoins non fonctionnels	23
3	Les diagrammes des cas d'utilisation	24
3.1	Le diagramme de cas d'utilisation Général	24
3.2	Les diagrammes de cas d'utilisation spécifiques	24
3.3	Les diagrammes de cas d'utilisation relatifs à l'administrateur	25
3.3.1	Description textuelle du cas d'utilisation Login	26

3.3.2	Description textuelle du cas d'utilisation "Manage Users"	27
3.3.3	Description textuelle du cas d'utilisation "Manage Settings"	28
3.4	Diagramme de cas d'utilisation relatif à l'utilisateur	29
3.4.1	Description textuelle du cas d'utilisation "Login"	29
4	Conception	31
4.1	Conception de la vue statique	32
4.1.1	Diagramme de classes	32
4.2	Conception de la vue dynamique	35
4.2.1	Diagrammes de séquences relatif à l'Administrateur	35
4.2.2	Diagrammes de séquences relatifs à l'utilisateur	39
5	Conclusion	45
4	Réalisation et tests	46
1	Introduction	47
2	Environnement du travail	47
2.1	Environnement matériel	47
2.2	Environnement logiciel	48
2.2.1	PaceStar UML	48
2.2.2	Jupyter Notebook	48
2.2.3	Python	49
2.2.4	Tensorflow	49
2.2.5	Keras	50
2.2.6	OpenCV	50
2.2.7	Flask	51
2.2.8	SQLite	52
2.2.9	React	53

3	Implémentation	53
3.1	Partie web	54
3.1.1	Expérience utilisateur	54
3.1.2	Expérience administrateur	56
4	Détection et vérification des utilisateurs	57
4.1	Phase de pré-traitement	57
4.2	Phase de vérification	57
5	Conclusion	58
	Conclusion générale et perspectives	59
	Bibliographie	60

Table des figures

1.1	Logo de l'application Dropbox	4
1.2	Logo de l'application Google Drive	5
1.3	Logo de l'application OneDrive	5
1.4	Diagramme de contexte statique	6
2.1	Normalisation de la taille de l'image.	10
2.2	Correction de la rotation et de l'inclinaison de l'image.	10
2.3	Les traits clés du visage humain.	11
2.4	Les différents étapes de MTCNN	12
2.5	Extraction de caractéristiques faciales	13
2.6	Architecture RESNET50, formée sur VGGFace2 pour extraction des traits clés du visage humain	14
2.7	Illustration simplifiée de l'architecture d'un réseau neuronal siamois.	15
2.8	Schéma du chiffrement symétrique.	17

2.9	Schéma du chiffrement asymétrique.	17
3.1	Diagrammes de cas d'utilisation Général	24
3.2	Diagrammes de cas d'utilisation relatif à l'administrateur	25
3.3	Le diagramme de cas d'utilisation relatif à l'utilisateur	29
3.4	Diagramme de classes	33
3.5	diagramme de séquence relatif au scénario "Login - Admin"	35
3.6	Diagramme de séquence relatif au scénario "Add User"	36
3.7	diagramme de séquence relatif au scénario "Manage User"	37
3.8	diagramme de séquence relatif au scénario "Manage system settings"	38
3.9	diagramme de séquence relatif au scénario "Signup"	39
3.10	Diagramme de séquence relatif au scénario "Login - User"	40
3.11	Diagramme de séquence relatif au scénario "Configure account settings"	41
3.12	Diagramme de séquence relatif au scénario "Upload File"	42
3.13	Diagramme de séquence relatif au scénario "Download"	43
3.14	Diagramme de séquence relatif au scénario "View File"	43
3.15	Diagramme de séquence relatif au scénario "Delete File"	44
4.1	Logo de PaceStar UML	48
4.2	Logo de Jupyter Notebook	49
4.3	Logo de Python	49
4.4	Logo de TensorFlow	50
4.5	Logo de Keras	50
4.6	Logo d'OpenCV	51
4.7	Logo de Flask	52
4.8	Logo SQLite	52
4.9	Logo de la bibliothèque React	53

4.10 Interface d'authentification	54
4.11 Interface d'authentification	55
4.12 Interface principale de la plateforme	55
4.13 Interface d'ajout des fichiers	56
4.14 Interface du tableau de bord administratif	56
4.15 Le déroulement du processus de la reconnaissance faciale	58

Liste des tableaux

3.1	Description textuelle du cas d'utilisation "Login"	26
3.2	Description textuelle du cas d'utilisation "Manage Users"	27
3.3	Description textuelle du cas d'utilisation "Manage Settings"	28
3.4	Description textuelle du cas d'utilisation "Login"	29
3.5	Description textuelle du cas d'utilisation "Manage Files"	30
3.6	Description textuelle du cas d'utilisation "Configure Settings"	31
3.7	Les différentes catégories de classes qui composent notre application	34

Introduction Générale

Le développement rapide des nouvelles technologies d'information a engendré une croissance exponentielle du volume de données. Face à cette augmentation, le stockage sur une plateforme cloud présente de nombreux avantages. Il offre un accès centralisé aux données, évitant ainsi les contraintes liées aux supports physiques de stockage, tout en permettant de réduire les coûts et de garantir la sécurité des données. De plus, il facilite l'accessibilité multi-plateforme, offrant ainsi une expérience utilisateur optimale.

Dans cette perspective, notre projet de fin d'année vise à créer une plateforme web de stockage de données sécurisée avec des fonctionnalités telles que le cryptage et la reconnaissance faciale. Notre objectif est de permettre aux utilisateurs de sauvegarder leurs données et d'y accéder, tout en offrant une expérience utilisateur unique et innovante.

Ce rapport est divisé en 4 chapitres :

- Chapitre 1 : "Cadre Général" : énoncer les objectifs visés. Il offre une vision globale du projet et de ses enjeux.
- Chapitre 2 : "Étude Théorique" : Exploration des principes fondamentaux dans les domaines de la reconnaissance faciale et le cryptage.
- Chapitre 3 : "Étude Conceptuelle" : Présentation d'une analyse conceptuelle du projet, notamment à travers les outils de modélisation.
- Chapitre 4 : "Réalisation et tests" : Mise en œuvre concrète de notre solution

Cadre Général du Projet

Sommaire

1	Introduction	3
2	Cadre du Projet	3
3	Étude et critique de l'existant	3
3.1	Étude l'existant	4
3.2	Critique de l'existant	5
4	Solution proposée	6
4.1	Diagramme de contexte	6
5	Conclusion	7

1 Introduction

Dans un paysage numérique en constante évolution, la sécurisation des échanges de données est devenue une préoccupation majeure. Notre projet se positionne dans ce contexte en proposant une plateforme de sauvegarde de fichiers sécurisée, dotée de fonctionnalités de cryptage avancées et de reconnaissance faciale. Cette initiative émerge en réponse à la nécessité croissante de protéger la confidentialité des données échangées en ligne.

Le domaine du projet englobe les aspects de sécurité des données et de technologies biométriques. Les objectifs de notre application sont clairs : garantir la confidentialité des échanges de fichiers, et renforcer l'authentification des utilisateurs. Ce rapport examinera de manière concise la conception, le développement et les perspectives de notre plateforme.

2 Cadre du Projet

Dans le cadre de notre projet de fin d'année, à l'École Nationale d'Ingénieurs de Tunis (ENIT), nous avons développé une plateforme de sauvegarde de fichiers. Cette initiative vise à fournir aux utilisateurs une solution robuste et sécurisée pour la gestion et la sauvegarde de leurs données numériques.

La plateforme permet aux utilisateurs de stocker leurs fichiers de manière sécurisée dans le cloud, avec la possibilité d'accéder à leurs données à tout moment et depuis n'importe quel appareil connecté à internet.

L'objectif principal de cette plateforme est de garantir la disponibilité et l'intégrité des données des utilisateurs. En fournissant une solution complète de sauvegarde et de gestion des fichiers, notre projet vise à répondre aux besoins croissants en matière de stockage et de sécurité des données dans un environnement numérique en constante évolution.

3 Étude et critique de l'existant

Cette section commence par une étude sur les applications existantes, en énumérant les critiques et en présentant ensuite la solution proposée.

3.1 Étude l'existant

Une étude de l'existant est une étape cruciale lors de la mise en place d'un projet, car elle permet d'évaluer les solutions déjà disponibles sur le marché, leurs avantages et leurs inconvénients, afin de définir clairement le positionnement et la valeur ajoutée du nouveau projet.

Voici une liste de quelques-unes des applications de stockage en ligne :

- **Dropbox** : Dropbox est l'un des premiers services de stockage cloud grand public et est largement utilisé pour sa simplicité d'utilisation et sa compatibilité multi-plateforme. Il offre une gamme de fonctionnalités telles que la synchronisation de fichiers entre plusieurs appareils, la collaboration en temps réel sur des documents, et la sauvegarde automatique des photos depuis les appareils mobiles[1]. La figure 1.1 illustre le logo de l'application Dropbox.



FIGURE 1.1 – Logo de l'application Dropbox

- **Google Drive** : Google Drive fait partie de l'écosystème Google et est étroitement intégré à d'autres services tels que Gmail et Google Docs. Il offre un stockage cloud gratuit avec un espace de stockage initial généreux pour les utilisateurs de comptes Google. Google Drive permet également la collaboration en temps réel sur des documents, des feuilles de calcul et des présentations, et offre une intégration transparente avec les autres produits Google[2]. La figure 1.2 montre le logo de l'application Google Drive.



FIGURE 1.2 – Logo de l'application Google Drive

- **OneDrive** : OneDrive offre des fonctionnalités de sécurité similaires à celles de Dropbox et Google Drive, telles que le cryptage des données en transit et au repos, ainsi que des options d'authentification à deux facteurs (2FA) pour protéger les comptes utilisateur. Étant intégré à l'écosystème Microsoft, OneDrive bénéficie également des normes de sécurité strictes de Microsoft pour protéger les données stockées[3]. La figure 1.3 présente le logo de l'application OneDrive.



FIGURE 1.3 – Logo de l'application OneDrive

3.2 Critique de l'existant

Bien que les services de stockage des fichiers dans le cloud, citons les exemples de Google Drive, Dropbox et One Drive, offrent des mesures de sécurité assez robustes telles que le chiffrement des données et l'authentification à deux facteurs, l'absence d'accès biométrique à ces plateformes demeure une lacune à ne pas négliger. L'absence de telle fonctionnalité facultative compromet la sécurité des comptes utilisateur, car les méthodes traditionnelles d'authentification peuvent être contournées, notamment par le biais de piratage de mots de passe ou de vol d'identifiants.

4 Solution proposée

Notre projet se concentre sur le développement d'une plateforme de sauvegarde de fichiers robuste et sécurisée, offrant une solution complète pour la gestion et la protection des données numériques.

Étant donné que la sécurité de la plateforme est notre premier souci, notre solution se distingue des solutions existantes par l'intégration d'un système d'authentification avancée basée sur la reconnaissance faciale. Cette technologie renforce considérablement la sécurité de la plateforme en s'assurant que seules les personnes autorisées peuvent accéder à leur compte. Ainsi, même en cas de compromission potentielle des identifiants de connexion, l'accès aux données sensibles reste protégé, car il nécessite une vérification biométrique supplémentaire.

Notre objectif est de fournir une solution innovante pour la sauvegarde et la gestion des fichiers, permettant aux utilisateurs de stocker leurs données en toute tranquillité d'esprit et d'y accéder facilement à tout moment, en garantissant un système d'authentification robuste et sécurisé.

4.1 Diagramme de contexte

Introduction sur le Diagramme de Contexte :

Le diagramme de contexte est une représentation visuelle simplifiée des interactions entre les principaux composants d'un système. Il offre une vue d'ensemble claire des entités impliquées et de la manière dont elles interagissent les unes avec les autres pour réaliser les fonctionnalités du système. Ce diagramme est illustré dans la figure 1.4 :

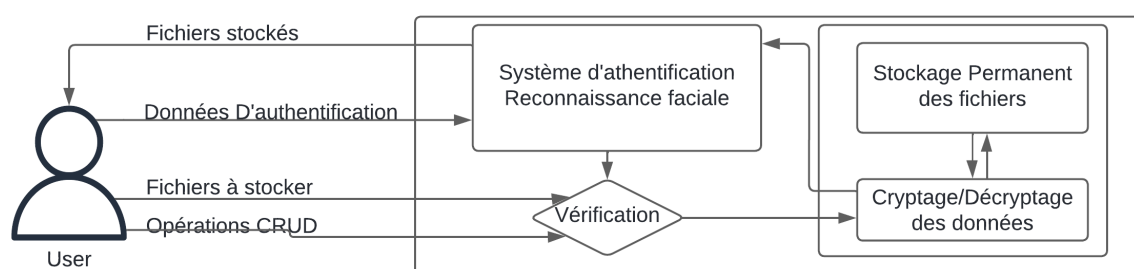


FIGURE 1.4 – Diagramme de contexte statique

Description du Diagramme de Contexte : La plateforme comprend quatre composants principaux : - "User" : représentant l'utilisateur qui interagit avec le système. - "Système d'Authentification" : vérifiant les données d'authentification fournies et l'identité biométrique l'utilisateur. - "Système de Cryptage/Decryptage" : responsable du cryptage et du décryptage des fichiers. - "Stockage Permanent" : représentant le stockage des fichiers.

Les méthodes associées à la plateforme permettent d'effectuer des opérations CRUD (Create, Read, Update, Delete) sur les fichiers, notamment : - Authentification de l'utilisateur. - Stockage d'un document sur le drive après cryptage. - Récupération d'un document du drive et décryptage. - Suppression d'un document du drive.

5 Conclusion

Ce chapitre a fourni une vue d'ensemble du projet, en présentant le contexte, la problématique et l'objectif de cet ouvrage. Nous avons ensuite effectué une étude de l'existant en décrivant les applications concurrentes et leurs limites. Nous avons également présenté notre solution, à l'aide du diagramme de contexte pour décrire les interactions entre les acteurs et le système. Le chapitre suivant examine en détail les algorithmes et les outils utilisés dans le développement du système de reconnaissance faciale, ainsi que les techniques de chiffrement des données employées dans le système.

Étude Théorique

Sommaire

1	Introduction	9
2	La reconnaissance faciale	9
2.1	Les étapes de reconnaissance faciale	9
2.2	Conclusion	15
3	Le chiffrement	16
3.1	Les types de chiffrement	16
3.2	Algorithmes de chiffrement courants	18
3.3	Avantages du chiffrement	19
3.4	Inconvénients du chiffrement	19
3.5	Conclusion	20
4	Conclusion	20

1 Introduction

Dans ce chapitre, nous examinerons les techniques de reconnaissance faciale, utilisées pour identifier une personne à partir de ses caractéristiques faciales. Nous aborderons également le chiffrement des données, une méthode essentielle pour sécuriser les informations stockées sur notre plateforme. Le chiffrement transforme les données en un format illisible, offrant une protection contre les accès non autorisés. L'objectif est de fournir une compréhension approfondie de ces deux techniques et de leur importance dans la sécurité des données sur notre plateforme.

2 La reconnaissance faciale

La reconnaissance faciale est une technologie en constante évolution qui permet l'identification d'une personne par ses caractéristiques faciales. Cette technique de reconnaissance biométrique est utilisée dans divers domaines, tels que la sécurité, la surveillance, la publicité et la recherche médicale. Dans cette section, nous examinerons les différentes étapes et méthodes de reconnaissance faciale. Nous aborderons également les techniques et les algorithmes utilisés pour mettre en oeuvre cette technologie.

2.1 Les étapes de reconnaissance faciale

Les étapes clés de la reconnaissance faciale sont la détection de visages, l'extraction de caractéristiques faciales et la reconnaissance des visages. Dans cette réponse, nous allons examiner chacune de ces étapes plus en détail.

1. **Capture et pré-traitement des images** : La première étape de la reconnaissance faciale consiste à prendre plusieurs photos du sujet mis en question. Dans notre cas, il s'agit de capture d'images de visage. Il est recommandé que le sujet tourne son visage afin de capturer tous les angles possibles et les différentes expressions faciales. Afin de rendre la reconnaissance des visages plus efficace, le système de détection des visages doit surmonter plusieurs difficultés :

- Les conditions d'imagerie : Les différences au niveau de la qualité des images capturées et les conditions d'obscurité posent également un défi au système de détection à surmonter, puisque les conditions d'éclairage ainsi que le type de la caméra affectent le rendu d'une image, et affecte donc l'apparence d'un visage.

- L'obstruction d'éléments : La présence des barbes, des lunettes ou des chapeaux

introduit une forte variabilité. Le système doit être invariant à ces changements.

- Les expressions faciales : Les traits du visage peuvent varier en raison des différents gestes du visage.

Pour adresser ces contraintes, les images enregistrées sont soumises à un processus de prétraitement ; normalisation de la taille des images, égalisation de l'histogramme, correction de la rotation et de l'inclinaison, suppression des artefacts et bruit à l'aide du filtrage de Gauss tout en préservant les contours et les détails de l'image. Ces techniques servent à mieux détecter les visages et améliorer de manière significative la précision de la vérification de l'utilisateur concerné [4].

La figure 2.1 montre un exemple d'une technique de pré-traitement : Normalisation de la taille de l'image.

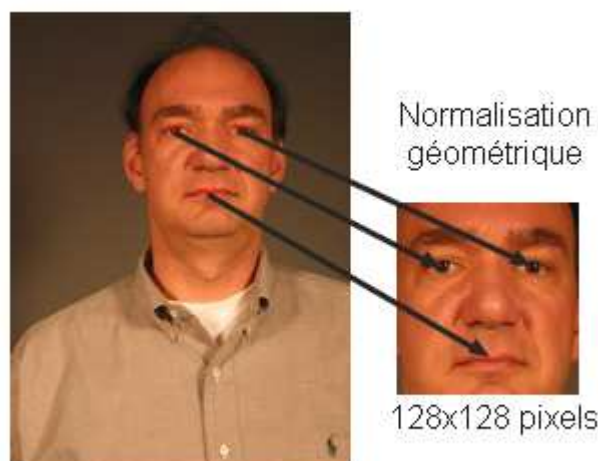


FIGURE 2.1 – Normalisation de la taille de l'image.

La figure 2.2 montre un exemple d'une technique de pré-traitement : correction de la rotation et de l'inclinaison de l'image [5].

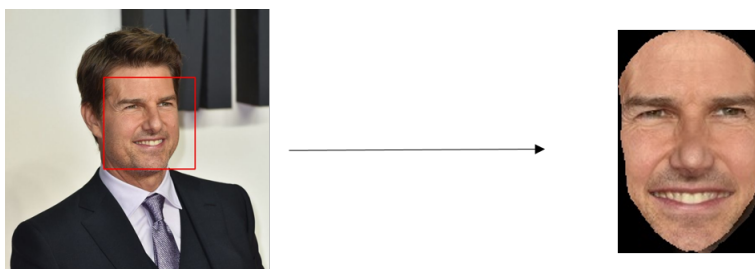


FIGURE 2.2 – Correction de la rotation et de l'inclinaison de l'image.

2. **Détection de visages** : La deuxième étape de la reconnaissance faciale consiste à détecter les visages dans une image. Cette étape est réalisée grâce à diverses approches, notamment l'analyse des caractéristiques locales comme celles utilisées dans les détecteurs Histogramme des Gradients (HOG) et les cascades de Haar, la reconnaissance de formes, et l'apprentissage en profondeur (Deep Learning-Based Face Detection). Ces algorithmes sont conçus pour repérer les contours et les caractéristiques clés du visage, comme les yeux, le nez et la bouche [6].

La figure 2.3 montre les caractéristiques clés du visage humain.

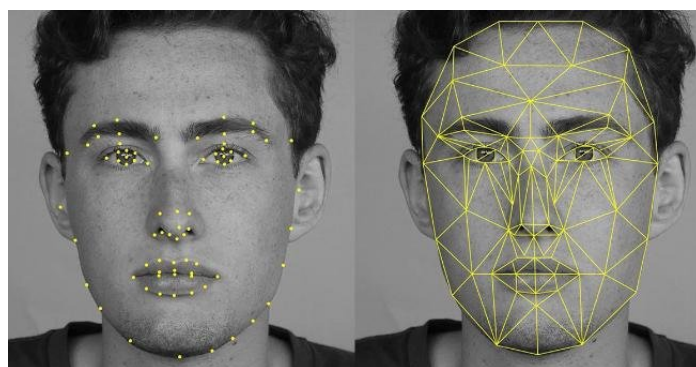


FIGURE 2.3 – Les traits clés du visage humain.

Pour assurer la meilleure détection possible de visages humains, nous avons choisi de nous concentrer exclusivement sur les méthodes basées sur l'apprentissage en profondeur. Les réseaux de neurones convolutifs (CNN), type d'algorithme de deep learning souvent appliqué, pour analyser et apprendre des caractéristiques visuelles, sont au cœur de notre approche. Pour la détection des visages, nous avons utilisé l'algorithme MTCNN (Multi-Task Cascaded Convolutional Neural Network).

- **MultiTask Cascaded Convolutional Neural Network** : est une approche contemporaine pour détecter les visages, utilisant un détecteur à trois niveaux basé sur des réseaux neuronaux convolutifs (CNN). Initialement, l'image est redimensionnée de manière itérative pour repérer des visages de différentes tailles. Ensuite, le P-Net (Proposal Net) analyse ces images, effectuant une première détection avec un seuil de détection relativement bas., et c'est pourquoi il détecte les faux positifs, même après NMS (NonMaximum Suppression). Les zones candidates, incluant les faux positifs, sont dirigées vers le deuxième réseau, appelé R-Net (Refinement Network), qui affine les détections (également en utilisant NMS) pour obtenir des boîtes englobantes plus précises. Ensuite, l'étape finale, l'O-Net (Output Network), réalise le raffinement final des boîtes englobantes.

Cette approche garantit non seulement la détection des visages, mais également une grande précision dans le placement des cadres de délimitation. [7].

La figure 2.4 montre les différents étapes de MTCNN

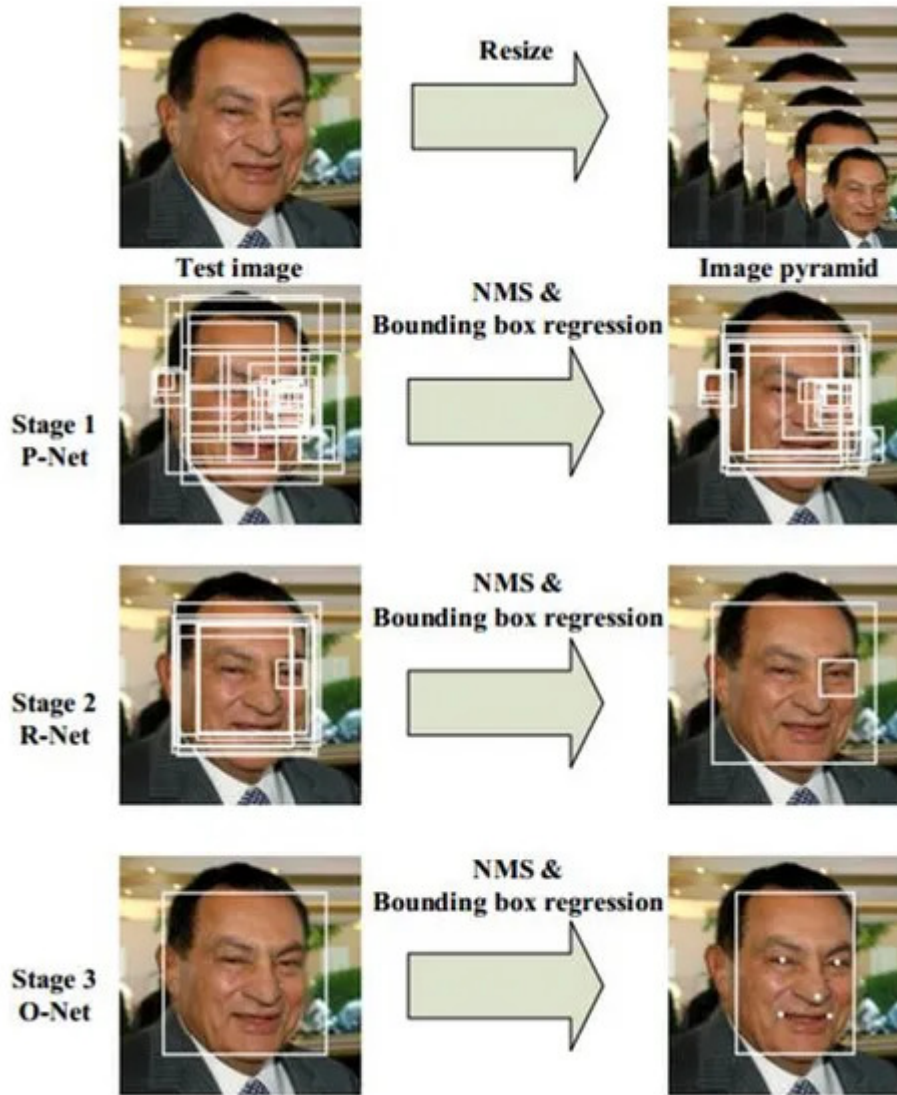


FIGURE 2.4 – Les différents étapes de MTCNN

MTCNN est très robuste. Il détecte correctement les visages même avec des tailles et des rotations importantes. Mais, il est plus lent par rapport aux autres détecteurs comme Haar-cascade, HOG et DNN. Un autre inconvénient se présente ici, c'est qu'il détecte les faux négatifs, problème auquel nous pouvons remédier en ajustant le seuil.

3. **Extraction de caractéristiques faciales** : La troisième étape de la reconnaissance faciale consiste à extraire les caractéristiques faciales des visages détectés telles que la forme du nez et des lèvres, la distance entre les yeux, la position et l'inclinaison des sourcils, etc. Celles-ci sont ensuite utilisées pour former un vecteur de caractéristiques unique pour chaque visage [8]. La figure 2.5 illustre une visualisation du processus d'extraction de caractéristiques faciales.



FIGURE 2.5 – Extraction de caractéristiques faciales

Grâce à l'abondance de données massives et à la disponibilité de réseaux spécialisés pour l'apprentissage profond, l'emploi des architectures CNN, telles que AlexNet, VGGNet, FaceNet, ResNet et VGGFace, est devenue particulièrement intéressante et appliquée de plus en plus. Parmi les architectures CNN citées auparavant, Dans ce projet, nous avons choisi d'utiliser l'architecture VGGFace pour l'extraction des caractéristiques faciales.

- VGGFace : décrit dans l'article de Omkar Parkhi intitulé "Deep Face Recognition" (2015), a introduit un ensemble de données massif spécialement conçu pour l'entraînement des systèmes de reconnaissance faciale basés sur les réseaux neuronaux. Ce modèle génère des caractéristiques généralisées à partir de visages, il utilise la fonction "triplet loss". VGGFace2, dans l'article "VGGFace2 : A dataset for recognizing faces across pose and age" publié en 2017[9], est un ensemble de données faciales complet composé de 3,31 millions d'images réparties sur 9131 sujets, illustrant des variations diverses en termes de pose, d'âge, d'éclairage, d'ethnicité et de profession. Des modèles sont entraînés sur ce dataset, en particulier les architectures ResNet-50 et SENet, et ont atteint une performance de pointe [10].

La figure 2.6 présente une illustration de l'architecture RESNET50, formée sur VGGFace pour extraire les caractéristiques clés du visage humain.

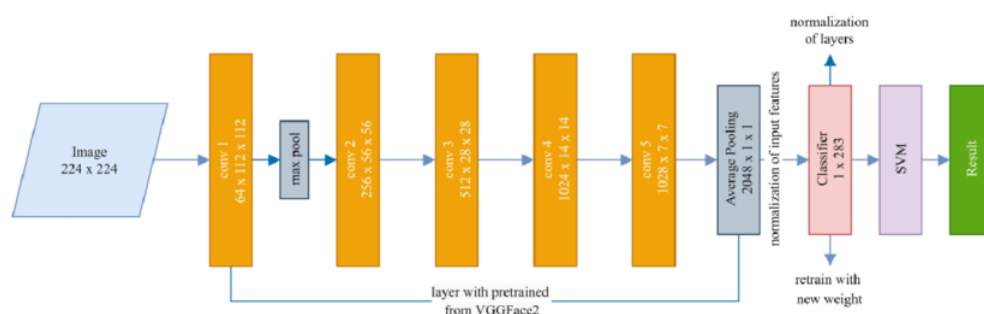


FIGURE 2.6 – Architecture RESNET50, formée sur VGGFace2 pour extraction des traits clés du visage humain

4. **Classification des visages** : La quatrième étape de la reconnaissance faciale consiste à vérifier ou identifier les visages en comparant les vecteurs de caractéristiques extraits avec ceux stockés dans la base de données. Concernant le choix d'architecture d'apprentissage automatique qui va effectuer la classification des images, dans ce cas, il est recommandé d'employer les architectures basées sur des fonctions de calcul de similarité, qui servent à déterminer si les images des visages correspondent, à celles déjà enregistrées. La base de données pourra elle-même être enrichie en temps réel, au fur et à mesure de l'inscription de nouveaux utilisateurs et donc, la détection de nouveaux visages. La solution adéquate se base sur une architecture siamoise.

- Les réseaux neuronaux siamois sont conçus pour extraire une mesure de similarité à partir de deux entrées distinctes qui possèdent une relation abstraite de similarité. Les premières recherches utilisant ces architectures se réfèrent à l'étude de Bromley et al. (1994), qui traitait de la vérification de signatures. Le document de recherche "Siamese Neural Networks for One-shot Image Recognition" réalisé par Gregory Koch, Richard Zemel et Ruslan Salakhutdinov et publié en 2015[11] explore l'emploi d'une telle architecture dans le contexte de classification d'images, basé sur le concept de "One-Shot Learning".

L'architecture siamoise fait intervenir deux réseaux de neurones identiques, partageant les mêmes paramètres (poids), qui prennent deux entrées indépendantes et qui se rejoignent à la fin grâce à une fonction de pénalité. Cette fonction se base sur une métrique calculée à partir des représentations de plus haut-niveau des deux réseaux. Chaque visage détecté est encodé par le réseau, en un vecteur. Celui-ci est comparé à une série d'empreintes faciales connues. Le réseau siamois fournit une valeur scalaire, intitulée score de similarité directement déduite de la distance séparant les représentations vectorielles des images fournies en entrée. Cette mesure est évaluée en fonction d'un seuil, au-delà duquel les empreintes

sont considérées comme identiques, permettant ainsi de vérifier l'identité de l'individu concerné [12].

La figure 2.7 présente une illustration simplifiée d'un réseau siamois à deux couches cachées simples pour la classification binaire avec une prédiction logistique p . La structure du réseau est répliquée dans les sections supérieure et inférieure pour former des réseaux jumeaux, avec des matrices de poids partagées à chaque couche.

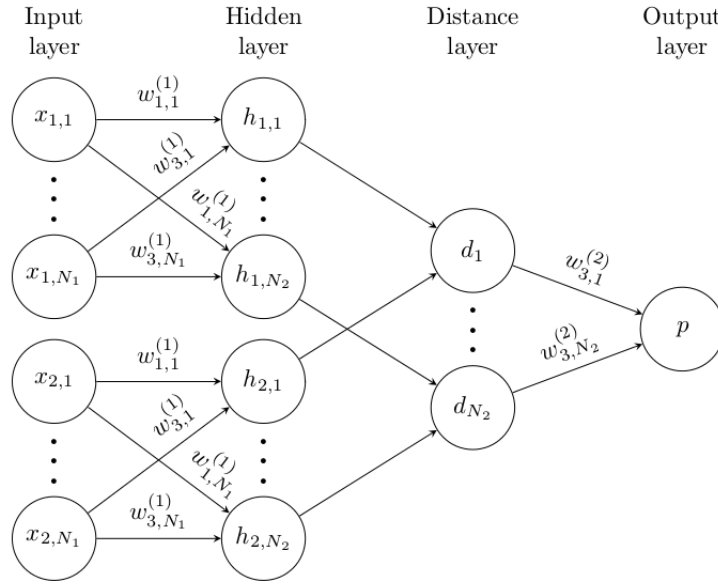


FIGURE 2.7 – Illustration simplifiée de l'architecture d'un réseau neuronal siamois.

Il convient de noter que la qualité des images joue un rôle important dans la précision de la reconnaissance faciale. Les images floues ou mal éclairées peuvent rendre la détection de visages difficile, tandis que les images de haute qualité permettent une reconnaissance plus précise. C'est pourquoi que la phase de prétraitement des images est cruciale afin d'assurer une bonne performance du réseau siamois.

2.2 Conclusion

La reconnaissance faciale est une technologie en plein essor, employée dans diverses applications, telles que la sécurité, la surveillance et la vérification d'identité. Cependant, il est important de comprendre les avantages et les limites de cette technologie afin de l'utiliser de manière responsable et éthique. Les différentes étapes et méthodes de reconnaissance

faciale, ainsi que les techniques et algorithmes utilisés, ont été examinés en détail dans ce chapitre, ce qui nous a permis d'acquérir une compréhension solide de cette technologie. Grâce à cette base de connaissances solide, nous avons pris la décision de s'orienter vers la conception d'un model d'apprentissage profond pour l'authentification des utilisateurs. Notre solution garantit un niveau de sécurité robuste, assurant la protection des données confidentielles.

3 Le chiffrement

Le chiffrement est l'un des éléments fondamentaux de la cryptographie. Ceci consiste à protéger les informations contre le vol à l'aide de modèles mathématiques. Ces algorithmes convertissent un texte brut lisible par les humains en texte incompréhensible, dans d'autres termes, en texte chiffré. Seules les personnes autorisées, peuvent le déchiffrer, à l'aide d'une clé de déchiffrement. Les méthodes de chiffrement sécurisées utilisent un nombre si important de clés cryptographiques qu'il est pratiquement impossible pour une personne non autorisée de deviner la bonne combinaison, ni utiliser un ordinateur pour essayer chaque combinaison potentielle (intitulée attaque par force brute). Ce processus peut être simple ou complexe, selon les exigences demandées. Les techniques de chiffrement moderne sont beaucoup plus sophistiquées que jamais, car elles utilisent des chaînes de centaines, voire de milliers de caractères générés par ordinateur comme clés de déchiffrement.

3.1 Les types de chiffrement

Les deux types d'algorithmes de chiffrement les plus utilisés sont symétriques et asymétriques.

3.1.1 Chiffrement à clés symétriques

Le chiffrement symétrique, connu aussi sous le nom d'algorithme de clé privée ou partagée, emploie la même clé pour chiffrer et déchiffrer le fichier en question. Les chiffrements de clés symétriques sont moins coûteux, en ressources de calcul, à générer. Ils ne consomment pas autant de puissance de calcul pour le chiffrement et le déchiffrement, par rapport aux autres techniques de chiffrement, ce qui réduit significativement le délai de décodage des données [13]. La figure 2.8 présente un schéma du chiffrement symétrique : la même clé est employée pour le chiffrement et le déchiffrement.

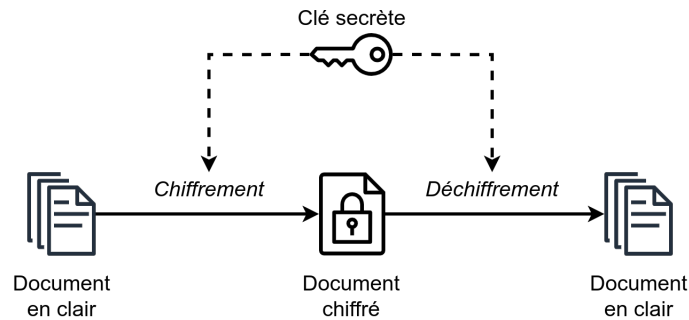


FIGURE 2.8 – Schéma du chiffrement symétrique.

L'inconvénient majeur pour ce type de chiffrement c'est que si une personne non autorisée possède la clé, elle pourra avoir accès à tous les messages et données envoyés entre les parties et découvrir leur contenu. Ainsi, le partage de la clé doit être lui-même chiffré avec une autre clé cryptographique, ce qui engendre un cycle de dépendance.

3.1.2 Chiffrement à clés asymétrique

Le chiffrement asymétrique, aussi connu sous le nom de cryptographie à clé publique, repose sur l'idée de l'utilisation de deux clés distinctes pour chiffrer et déchiffrer les données. L'une s'agit d'une clé publique, partagée entre toutes les parties concernées, destinée pour le chiffrement. Toute personne disposant de cette clé peut envoyer un message chiffré, mais seuls les détenteurs de la deuxième clé privée peuvent déchiffrer le message, donc lire le contenu du message envoyé par l'autre partie [14]. La figure 2.9 présente schéma de chiffrement asymétrique implique l'utilisation de deux clés distinctes : une pour le chiffrement et une autre pour le déchiffrement.

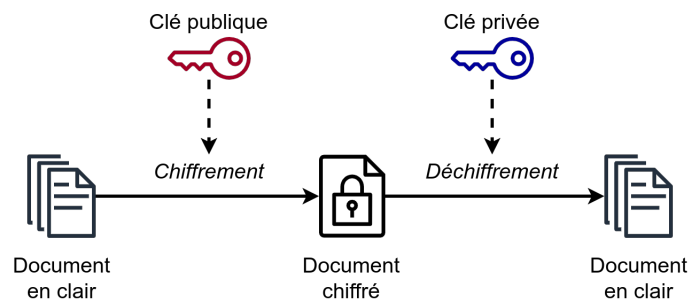


FIGURE 2.9 – Schéma du chiffrement asymétrique.

Le chiffrement asymétrique est désormais plus coûteux à générer et nécessite une puissance de calcul assez importante pour le déchiffrement, car la clé de chiffrement publique est souvent volumineuse (entre 1024 et 2048 bits). C'est pour cette raison que le chiffrement asymétrique n'est pas généralement adapté aux données volumineuses.

3.2 Algorithmes de chiffrement courants

Plusieurs types d'algorithmes de chiffrement existent pour différentes circonstances. Les algorithmes les plus courants sont :

1. Data Encryption Standard (DES) : La DES est une norme de chiffrement développée dans les années 1970. Sa taille n'était que de 56 bits, ceci la rend obsolète dans l'écosystème actuel. Elle a eu une influence majeure sur l'évolution de la cryptographie. En se basant sur cette norme, les cryptographes ont menés des recherches plus approfondies dans le but d'élaborer des systèmes de chiffrement plus sophistiqués.
2. Triple DES : La prochaine évolution de DES a pris le bloc de chiffrement de DES et l'a appliqué trois fois à chaque bloc de données qu'il a chiffré en le chiffrant, en le déchiffrant, puis en le chiffrant à nouveau. L'approche a considérablement accru la longueur de la clé, rendant ainsi la tâche de déchiffrement par une attaque par force brute beaucoup plus ardue. Néanmoins, 3DES est désormais jugé non sécurisé et a finalement été abandonné.
3. Advanced Encryption Standard (AES) est actuellement le système de chiffrement le plus largement utilisé. Il a été conçu sur la base d'un concept appelé "réseau de substitution". AES est un algorithme de chiffrement par bloc de 128 bits, pouvant être utilisé avec des clés de 128, 192 ou 256 bits.
4. Twofish : est un algorithme conçu pour être utilisé à la fois dans des environnements matériels et logiciels. Il est reconnu par les experts comme étant le chiffrement symétrique le plus rapide. Il peut supporter des clés allant jusqu'à 256 bits.

Bien que l'utilisation de Twofish soit gratuite, elle ne relève ni de la protection par brevet ni du modèle Open Source. Malgré cela, elle est intégrée dans des applications de chiffrement populaires telles que PGP (Pretty Good Privacy).

5. RSA : l'un des premiers systèmes de chiffrement asymétrique. Il doit son nom aux trois chercheurs du MIT qui l'ont décrit en 1977 : Rivest, Shamir et Adleman. Contrairement au chiffrement symétrique, RSA utilise deux clés distinctes : une clé publique et une clé privée. La clé publique peut être utilisée par quiconque pour chiffrer des données, mais seule la personne possédant la clé privée correspondante peut les déchiffrer. Les clés RSA ont tendance à être longues, habituellement de 2048 ou 4096 bits, ce qui peut les rendre coûteuses et lentes à manipuler. Elles

sont souvent utilisées pour chiffrer des informations sensibles ou pour sécuriser les échanges de clés de chiffrement symétrique.

3.3 Avantages du chiffrement

Les techniques de chiffrement présentent myriade d'avantages pour la protection des données sensibles des utilisateurs, parmi ceux-ci, on cite :

1. Protection des données : Les données circulent continuellement, que ce soit des messages entre utilisateurs ou des transactions financières. Le chiffrement, combiné à d'autres mesures de sécurité telles que l'authentification, permet de protéger les données lors de leur transfert entre différents appareils ou serveurs.
2. Garantie de l'intégrité des données : En plus d'empêcher l'accès des personnes non autorisées au texte brut des données, le chiffrement garantit que les données sont protégées contre toute utilisation malveillante, que ce soit pour commettre des fraudes, des extorsions, ou altérer des documents confidentiels.
3. Protection des transformations numériques : Avec l'essor de l'utilisation du stockage dans le cloud par les organisations et les individus, le chiffrement joue un rôle essentiel dans la protection de ces données, qu'elles soient en transit dans le cloud, au repos sur le serveur, ou en cours de traitement.
4. Réglementation : Un paysage réglementaire en constante évolution impose aux entreprises de renforcer leurs mesures de cryptage pour protéger les données sensibles. Cette exigence s'applique particulièrement aux domaines de la santé, des transactions par carte de paiement et au respect du RGPD.

Ces réglementations, telles que la norme PCI DSS et le RGPD, établissent des normes rigoureuses pour garantir la confidentialité et l'intégrité des informations sensibles. Le cryptage s'impose comme un élément crucial de cette stratégie de protection

3.4 Inconvénients du chiffrement

Malgré ses nombreux bienfaits, le chiffrement présente également des inconvénients qu'il faut prendre en considération.

1. Rançongiciels :

Bien que le chiffrement doit être uniquement utilisé pour sécuriser les données, il peut parfois être détourné par des acteurs malveillants pour mener des attaques

de ransomware. En cas d'attaque sur les données d'une organisation, les acteurs peuvent les chiffrer et les rendant inaccessibles jusqu'à ce qu'une rançon soit versée pour les récupérer.

2. Gestion des clés : Le chiffrement devient moins efficace si les clés de chiffrement et de déchiffrement ne sont pas sécurisées. Les cybercriminels ciblent souvent l'acquisition des clés de chiffrement d'une organisation. La perte de clés de chiffrement (par exemple, lors d'une catastrophe naturelle qui compromet les serveurs) peut rendre les données importantes inaccessibles pour les organisations. C'est pourquoi les organisations recourent à un système de gestion de clés sécurisé pour gérer et protéger leurs clés d'accès.

3.5 Conclusion

La cryptographie joue un rôle essentiel dans la préservation des données sensibles. Bien que les avantages du chiffrement soient nombreux, notamment la protection de tout type de données sur n'importe quel appareil, il est important de prendre en considération les limites des technologies du chiffrement. Dans le cas de notre projet, choisir les bonnes pratiques du chiffrement est essentiel pour instaurer la confiance des utilisateurs dans le produit que nous développons et garantir la sécurité de leurs données confidentielles.

4 Conclusion

À travers ce chapitre, nous avons exploré deux piliers fondamentaux de la sécurité des données : la reconnaissance faciale et le chiffrement. La reconnaissance faciale offre un potentiel immense dans le domaine de vérification d'identité, tandis que le chiffrement garantit l'intégrité des informations sensibles. En combinant ces deux technologies, nous sommes en train de concevoir une solution robuste et sécurisée, conforme aux normes de sécurité les plus élevées, tout en respectant la vie privée des utilisateurs, dans le monde du stockage en ligne.

Dans le chapitre suivant, nous passerons en revue les aspects conceptuels de notre projet en utilisant le langage de modélisation unifié UML, qui nous permettra de présenter une variété de diagrammes pour modéliser la structure statique et dynamique de notre projet.

Étude Conceptuelle

Sommaire

1	Introduction	22
2	Spécification des besoins	22
2.1	Besoins fonctionnels	22
2.2	Les besoins non fonctionnels	23
3	Les diagrammes des cas d'utilisation	24
3.1	Le diagramme de cas d'utilisation Général	24
3.2	Les diagrammes de cas d'utilisation spécifiques	24
3.3	Les diagrammes de cas d'utilisation relatifs à l'administrateur	25
3.4	Diagramme de cas d'utilisation relatif à l'utilisateur	29
4	Conception	31
4.1	Conception de la vue statique	32
4.2	Conception de la vue dynamique	35
5	Conclusion	45

1 Introduction

Dans ce chapitre, nous allons identifier les besoins fonctionnels du système en et définir les besoins non fonctionnels, tels que les performances, la sécurité et la fiabilité. Ensuite, nous présenterons une conception détaillée du système à l'aide de diagrammes statiques et dynamiques. Pour cela, nous utiliserons le langage de modélisation UML pour visualiser les différentes parties du système et les relations entre elles.

2 Spécification des besoins

2.1 Besoins fonctionnels

La spécification des besoins fonctionnels est une étape essentielle dans tout projet de développement de logiciels. Pour notre projet, nous avons identifié deux acteurs clés qui interagissent avec notre système : l'administrateur et l'utilisateur.

2.1.1 Pour l'administrateur :

- Consultation des données : L'administrateur doit pouvoir consulter les informations relatives aux utilisateurs ainsi que les données biométriques associées.
- Modification des informations des profils : Il doit être en mesure de mettre à jour les informations des profils des utilisateurs, notamment en cas de changement de statut ou d'identification de nouvelles informations.
- Gestion des données stockées : L'administrateur est responsable de la gestion des données stockées sur la plateforme, y compris la suppression des données obsolètes ou non pertinentes.

2.1.2 Pour l'utilisateur :

- Authentification : L'utilisateur doit pouvoir s'authentifier de manière sécurisée sur la plateforme, en utilisant des méthodes telles que la reconnaissance faciale
- Accès au dossier de sauvegarde : Une fois authentifié, l'utilisateur doit pouvoir accéder à son dossier de sauvegarde personnel, où il peut stocker et gérer ses fichiers
- Gestion du compte : L'utilisateur doit avoir la possibilité de modifier ses informations de profil, ses préférences de sécurité et les paramètres de sauvegarde des

fichiers.

2.2 Les besoins non fonctionnels

Dans un projet de reconnaissance faciale, les besoins non fonctionnels sont des exigences qui ne sont pas directement liées aux fonctionnalités de base du système, mais qui sont essentielles pour garantir la performance, la sécurité et la fiabilité du système.

- Les performances sont un besoin clé, car la reconnaissance faciale implique le traitement rapide de grandes quantités de données en temps réel. Le système doit être en mesure de détecter et d'identifier les visages rapidement et avec une grande précision.
- La sécurité est également une préoccupation majeure, car les données biométriques et les documents sauvegardés nécessitent une protection rigoureuse contre les menaces de sécurité.
- La fiabilité est un autre besoin important, car le système doit être fiable et stable pour garantir une utilisation continue.

Pour répondre à ces besoins non fonctionnels, le projet de reconnaissance faciale doit être conçu avec des mesures de sécurité et de sauvegarde robustes, une capacité de traitement rapide et une maintenance régulière pour assurer une utilisation fiable et durable.

3 Les diagrammes des cas d'utilisation

3.1 Le diagramme de cas d'utilisation Général

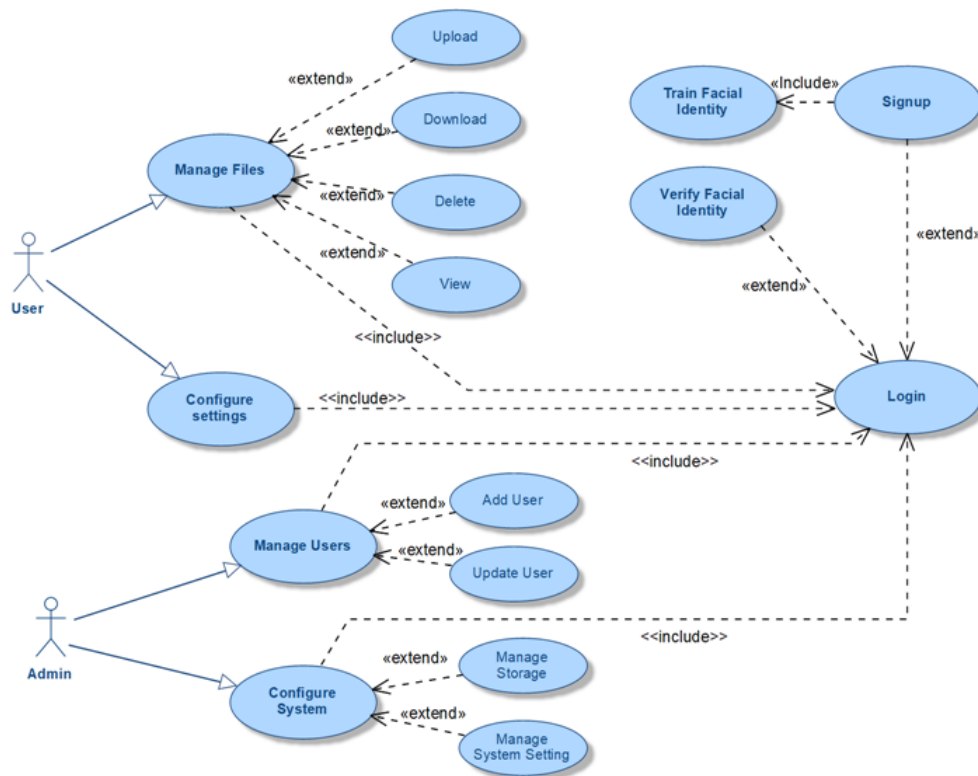


FIGURE 3.1 – Diagrammes de cas d'utilisation Général

3.2 Les diagrammes de cas d'utilisation spécifiques

Dans cette section, nous allons décrire les diagrammes de cas d'utilisation relatifs à chaque utilisateur. Nous allons commencer par examiner le diagramme de cas d'utilisation Général.

3.3 Les diagrammes de cas d'utilisation relatifs à l'administrateur

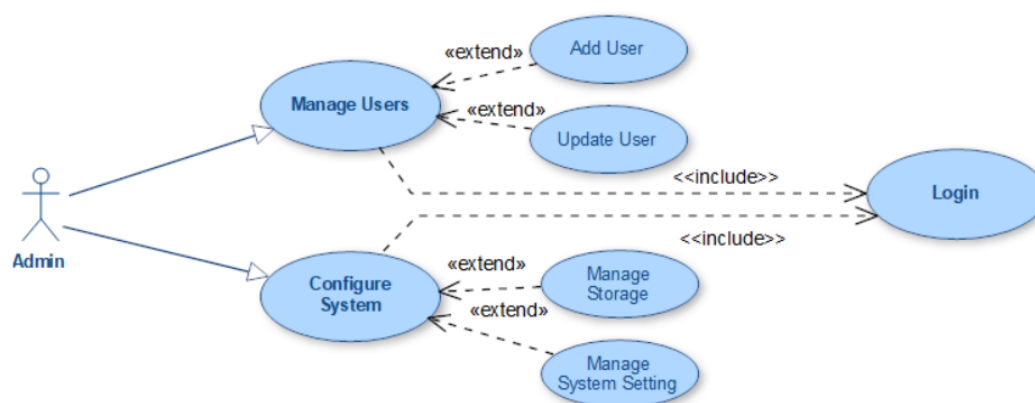


FIGURE 3.2 – Diagrammes de cas d'utilisation relatif à l'administrateur

Le rôle de l'administrateur est de consulter et modifier les données des utilisateurs et leurs informations biométriques associées. De plus, il est chargé de gérer les données stockées sur la plateforme, en assurant une suppression des données obsolètes ou non pertinentes.

3.3.1 Description textuelle du cas d'utilisation Login

TABLE 3.1 – Description textuelle du cas d'utilisation "Login"

Titre	Login
Résumé	Se connecter à son compte dans la plateforme
Acteur	Administrateur
Pré-condition	L'administrateur doit avoir un compte Admin vérifié
Post-condition	L'administrateur se connecte avec succès.
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur se connecte à son compte en utilisant ses identifiants.2. L'administrateur est dirigé vers une interface de vérification d'identité faciale.3. L'administrateur est dirigé vers la page d'accueil "Dashboard" .
Scénario alternatif	Informations erronées Echec de la vérification faciale

3.3.2 Description textuelle du cas d'utilisation "Manage Users"

TABLE 3.2 – Description textuelle du cas d'utilisation "Manage Users"

Titre	Gérer les utilisateurs
Résumé	L'exécution du cas d'utilisation permet à l'administrateur de modifier les informations et les permissions des utilisateurs existants ou ajouter des nouveaux utilisateurs.
Acteur	Administrateur
Pré-condition	L'administrateur doit s'authentifier pour pouvoir exécuter ce cas d'utilisation.
Post-condition	Les utilisateurs ont été modifiées avec succès.
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur se connecte à son compte en utilisant ses identifiants.2. Il doit cliquer sur le bouton "Manage Users".3. Le système renvoie toutes les informations des utilisateurs.4. L'administrateur appuie sur le bouton "Confirmer".5. Le système met à jour la base de données avec les nouvelles informations modifiées.
Scénario alternatif	<p>A1. La connexion n'a pas pu être établie. A2. Il manque un champ obligatoire. A3. L'authentification a échoué.</p>

3.3.3 Description textuelle du cas d'utilisation "Manage Settings"

TABLE 3.3 – Description textuelle du cas d'utilisation "Manage Settings"

Titre	Manage settings
Résumé	L'exécution du cas d'utilisation permet à l'administrateur de manipuler les différents paramètres de l'application.
Acteur	Administrateur
Pré-condition	L'administrateur doit s'authentifier pour pouvoir exécuter ce cas d'utilisation.
Post-condition	Les paramètres ont été modifiés avec succès.
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur se connecte à son compte en utilisant ses identifiants.2. Il doit cliquer sur le bouton "Manage Settings".3. Le système renvoie toutes les informations des utilisateurs.4. L'administrateur appuie sur le bouton "Confirmer".5. Le système met à jour la base de données avec les nouvelles informations modifiées.
Scénario alternatif	<ol style="list-style-type: none">A1. La connexion n'a pas pu être établie.A2. Il manque un champ obligatoire.A3. L'authentification a échoué.

Cette section nous a offert une vue approfondie des actions possibles pour l'administrateur, allant de la gestion des utilisateurs à la configuration des paramètres, essentielles pour garantir le bon fonctionnement et la sécurité de la plateforme. Dans les prochains paragraphes, nous aborderons les diagrammes des cas d'utilisation de l'utilisateur.

3.4 Diagramme de cas d'utilisation relatif à l'utilisateur

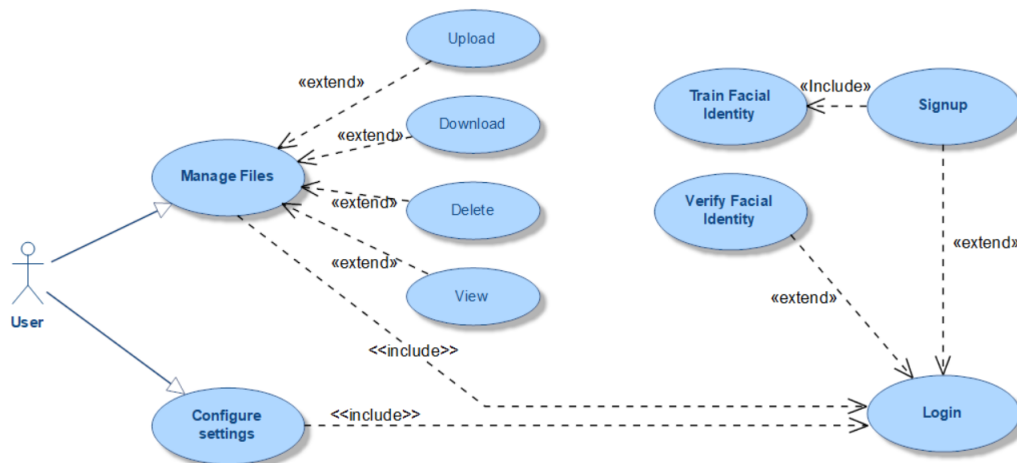


FIGURE 3.3 – Le diagramme de cas d'utilisation relatif à l'utilisateur

3.4.1 Description textuelle du cas d'utilisation "Login"

TABLE 3.4 – Description textuelle du cas d'utilisation "Login"

Titre	Login
Résumé	Se connecter à son compte dans la plateforme
Acteur	Utilisateur
Pré-condition	L'utilisateur doit créer un compte "Sign-Up" et enregistrer son identité faciale "Train Facial Identity"
Post-condition	L'utilisateur se connecte avec succès.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur se connecte à son compte en utilisant ses identifiants. 2. L'utilisateur est dirigé vers une interface de vérification d'identité faciale. 3. L'utilisateur est dirigé vers la page d'accueil dans laquelle il peut consulter et modifier ses fichiers.
Scénario alternatif	Informations erronées Echec de la vérification faciale

TABLE 3.5 – Description textuelle du cas d'utilisation "Manage Files"

Titre	Manage Files
Résumé	L'exécution du cas d'utilisation permet à l'utilisateur, de Télécharger, Supprimer et Voir ses fichiers.
Acteur	Utilisateur
Pré-condition	L'utilisateur doit s'authentifier pour pouvoir exécuter ce cas d'utilisation.
Post-condition	Fichiers Modifiés avec succès
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur se connecte à son compte en utilisant ses identifiants.2. Il doit cliquer sur le bouton Upload ou view ou Download ou Delete.3. Le système Demande à l'utilisateur de vérifier son identité faciale si cette fonction est activée.4. Le système accepte ou refuse la modification effectué par l'utilisateur
Scénario alternatif	A1. La connexion n'a pas pu être établie. A2. L'authentification a échoué. A3. Fichier trop volumineux

TABLE 3.6 – Description textuelle du cas d'utilisation "Configure Settings"

Titre	Configure Settings
Résumé	L'exécution du cas d'utilisation permet à l'administrateur de changer les paramètres liés à son compte.
Acteur	Utilisateur
Pré-condition	L'utilisateur doit s'authentifier pour pouvoir exécuter ce cas d'utilisation.
Post-condition	Les personnes recherchées ont été supprimées avec succès.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur se connecte à son compte en utilisant ses identifiants. 2. Il doit cliquer sur le bouton "Configure Settings". 3. Le système renvoie la liste des paramètres de comptes à modifier. 4. L'utilisateur choisit les options à modifier. 5. Ensuite, en cliquant sur le bouton "Save" et en confirmant l'action, l'utilisateur doit vérifier son identité faciale.
Scénario alternatif	<p>A1. La connexion n'a pas pu être établie.</p> <p>A2. L'authentification a échoué.</p>

4 Conception

Dans cette section, nous abordons l'étape la plus cruciale de l'élaboration de ce travail, à savoir la conception. Nous présentons une conception détaillée à travers la description des différents diagrammes de séquences et du diagramme de classes, qui sont en lien avec les cas d'utilisation exposés dans la section précédente du chapitre.

Nous allons maintenant entamer la conception détaillée de notre application en abordant deux aspects fondamentaux : la vue statique et la vue dynamique. La conception de la vue statique met l'accent sur la représentation des structures statiques du système, telles que les classes et les relations entre elles. Quant à la conception de la vue dynamique, elle concerne les aspects temporels et les interactions entre les différents éléments du système.

4.1 Conception de la vue statique

La conception de la vue statique d'un système consiste à identifier les différents éléments qui composent ce système et les relations qui existent entre eux. Cette vue peut être représentée à travers un diagramme de classes, qui est l'un des diagrammes les plus utilisés en UML. Ce diagramme permet de modéliser les classes, les attributs, les méthodes et les relations entre les classes. Il représente également les différentes couches du système, ainsi que les associations et les agrégations entre les classes. La conception de la vue statique est essentielle pour la compréhension du système, car elle permet de visualiser l'organisation de ses composants et leur relation les uns avec les autres. Elle constitue également une étape importante dans la conception d'un système, car elle permet de définir les fondations sur lesquelles la vue dynamique du système sera construite.

4.1.1 Diagramme de classes

Le diagramme de classes permet de visualiser la structure statique du système, en représentant les différentes classes et les relations qui existent entre elles. Dans la figure 3.4, nous présentons les différentes classes impliquées dans le fonctionnement de notre application.

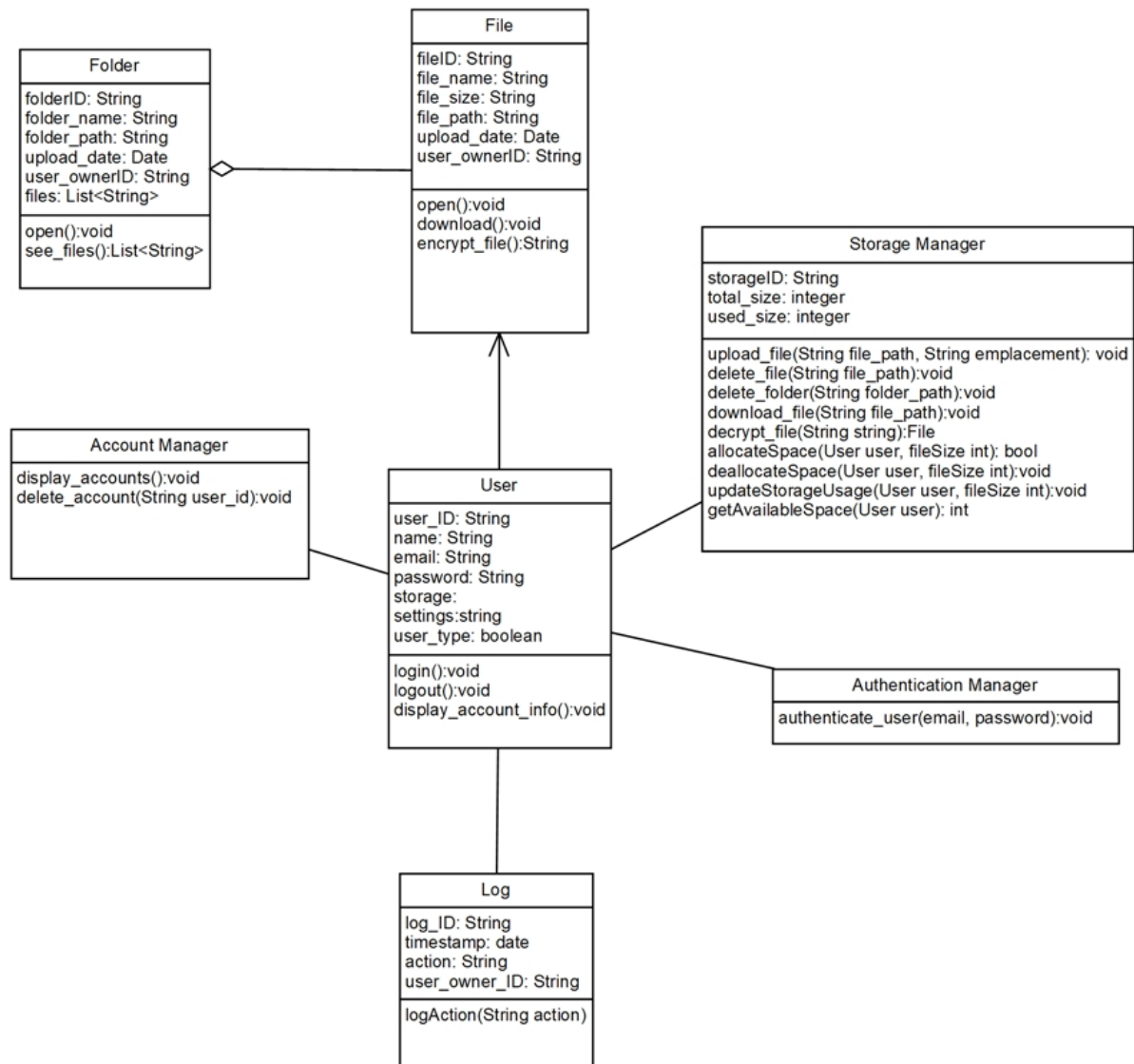


FIGURE 3.4 – Diagramme de classes

TABLE 3.7 – Les différentes catégories de classes qui composent notre application

Classe	Description
Utilisateur	<p>Un utilisateur est identifié par les attributs suivants : un identifiant unique, nom, prénom, email, mot de passe, ainsi que le type de l'utilisateur ; est ce qu'il s'agit d'un utilisateur régulier ou un administrateur.</p> <p>Si l'utilisateur s'agit d'un administrateur, les méthodes auxquelles il a accès sont : "S'authentifier", "Gérer son profil", "Consulter les comptes des utilisateurs", "Supprimer un compte".</p> <p>Sinon, un utilisateur normal a désormais accès à : "S'authentifier", "Gérer son profil" (pour modifier ses propres informations), "Consulter, ajouter, télécharger ou supprimer des fichiers".</p>
Fichier	<p>Un fichier est identifié par les attributs suivants : un identifiant unique, nom, volume, date d'ajout, chemin d'accès et identifiant du propriétaire.</p> <p>Les méthodes possibles sont : "Consulter", "Télécharger" ou "Chiffrer".</p>
Dossier	<p>Un dossier est identifié par les attributs suivants : un identifiant unique, nom, volume, date de création, chemin d'accès et identifiant du propriétaire.</p> <p>Les méthodes possibles sont : "Ouvrir dossier" et "Consulter le contenu du dossier".</p>
Gestionnaire de stockage	<p>Un gestionnaire de stockage est identifié par les attributs suivants : un identifiant unique, nom, volume libre et volume occupé.</p> <p>Les méthodes possibles sont : "Uploader fichier", "Supprimer fichier", "Supprimer dossier", "Télécharger fichier", "Déchiffrer fichier", "Gérer espace mémoire" et "Consulter espace libre".</p>
Gestionnaire d'authentification	<p>Les méthode(s) possible(s) pour le gestionnaire d'authentification est (sont) : "Authentification utilisateur".</p>
Gestionnaire de comptes	<p>Les méthode(s) possible(s) pour le gestionnaire de comptes est (sont) : "Consulter informations comptes" et "Supprimer compte".</p>
Historique	<p>la classe "Historique" est identifiée par les attributs suivants : date, description, et l'originale de l'action.</p> <p>Les méthodes auxquelles on peut accéder sont : "Consulter l'historique" et "Supprimer l'historique".</p>

4.2 Conception de la vue dynamique

La conception de la vue dynamique UML permet de visualiser les interactions entre les objets du système lors de l'exécution des différentes fonctionnalités. Cette vue est représentée sous forme de diagrammes d'interaction, tels que les diagrammes de séquence. Le diagramme de séquence illustre les séquences d'échanges de messages entre les objets et montre l'ordre dans lequel ces échanges ont lieu.

4.2.1 Diagrammes de séquences relatif à l'Administrateur

Nous décrivons dans ce qui suit les différents diagrammes de séquence de quelques cas d'utilisation. Le diagramme de séquence est un type de diagramme de comportement UML qui montre l'interaction entre différents objets d'un système selon une séquence d'actions. Il permet de représenter l'ordre des messages échangés entre les différents objets, ainsi que les événements qui déclenchent ces messages. Les diagrammes de séquence sont souvent utilisés pour modéliser les scénarios d'utilisation d'un système et pour spécifier les contraintes temporelles associées à ces scénarios.

La figure 3.5 présente le diagramme de séquence relatif au scénario Administrateur "S'authentifier".

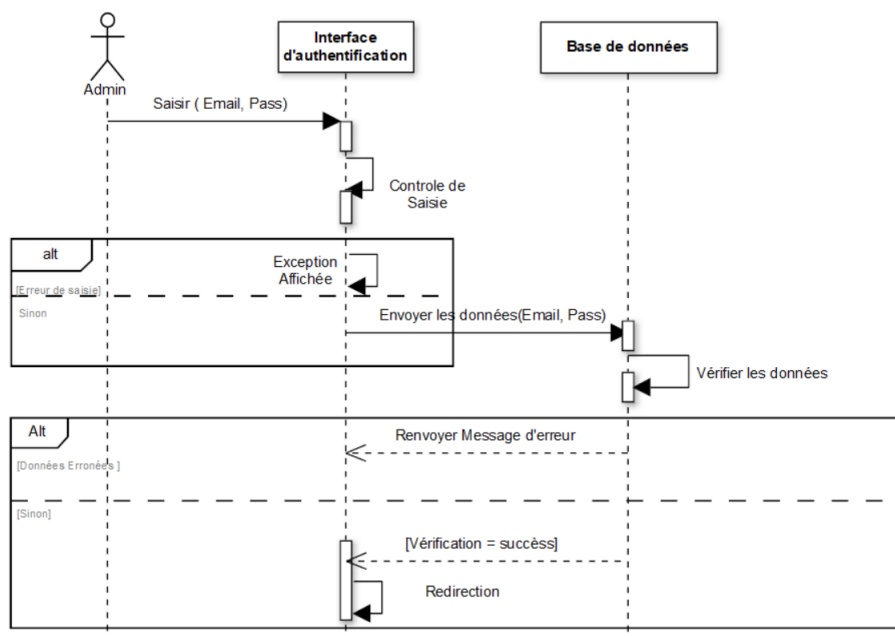


FIGURE 3.5 – diagramme de séquence relatif au scénario "Login - Admin"

La figure 3.6 illustre le diagramme de séquence relatif au scénario "Ajouter Utilisateur".

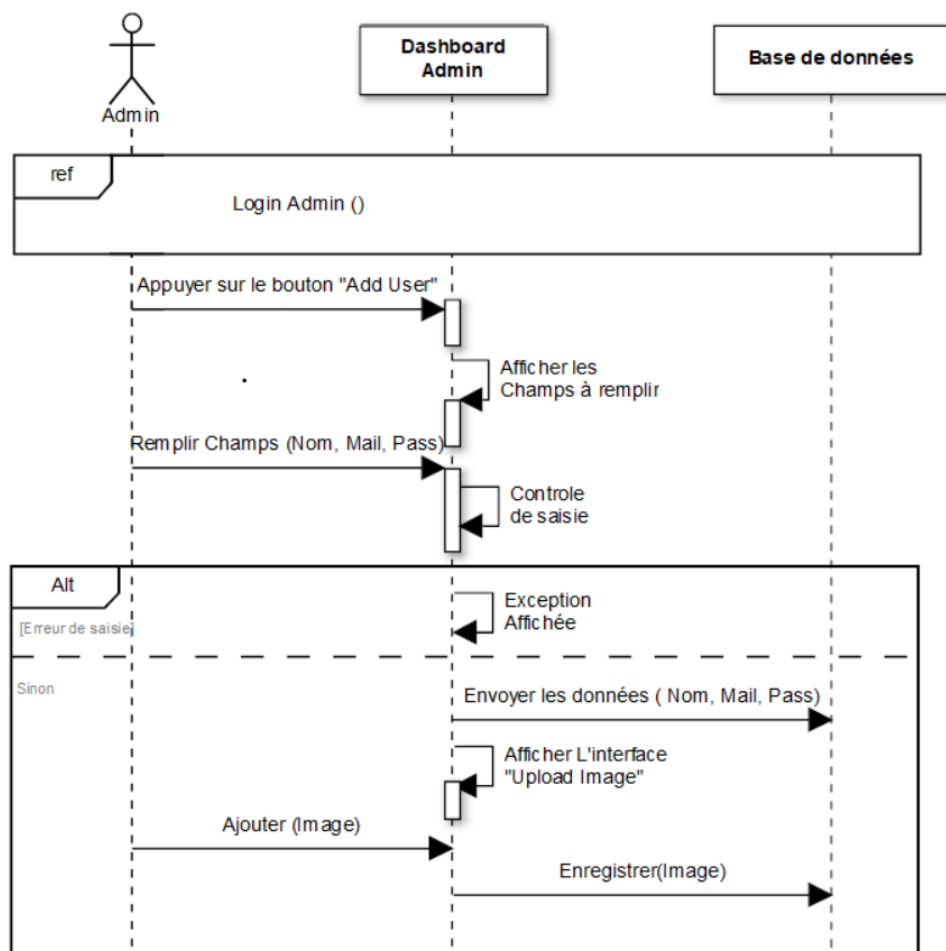


FIGURE 3.6 – Diagramme de séquence relatif au scénario "Add User"

La figure 3.7 montre le diagramme de séquence relatif au scénario "Gérer utilisateur".

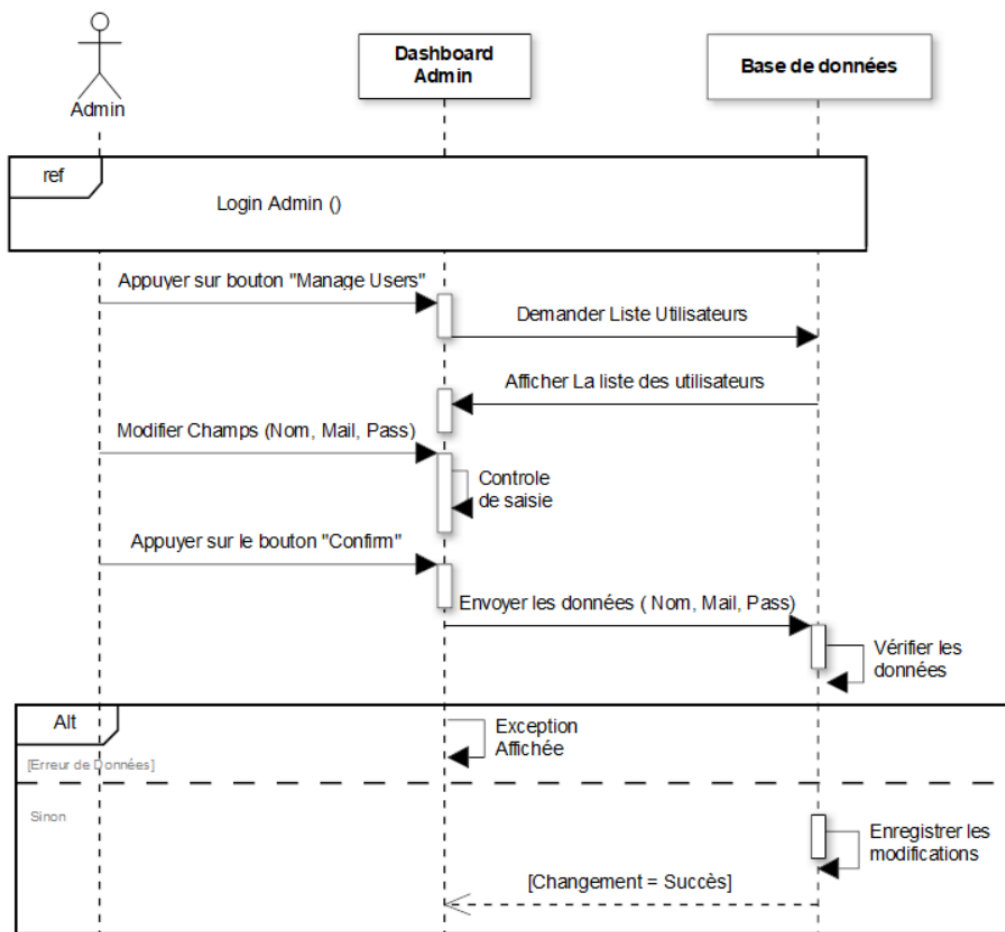


FIGURE 3.7 – diagramme de séquence relatif au scénario "Manage User"

La figure 3.8 montre le diagramme de séquence relatif au scénario "Gérer paramètres système".

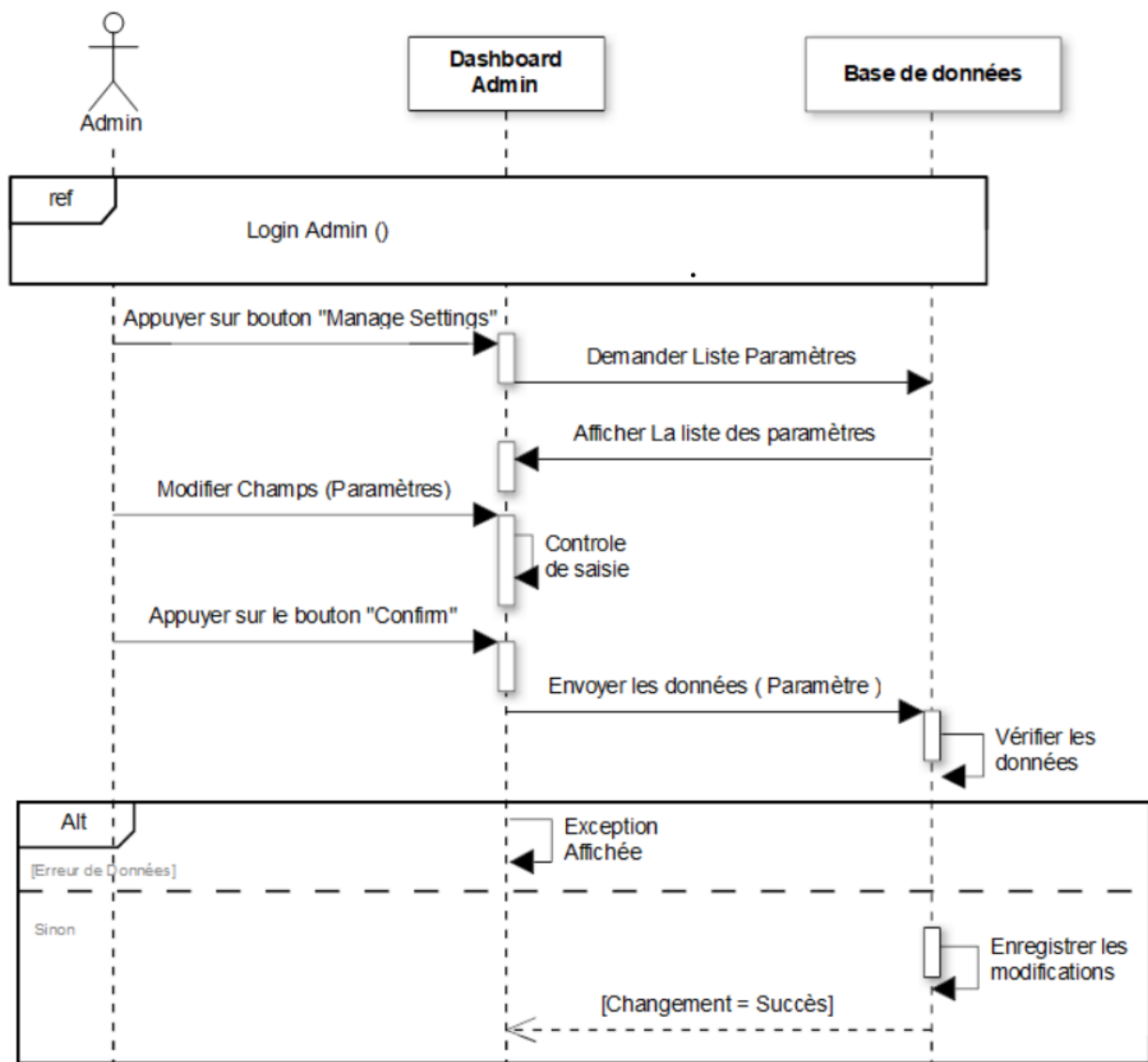


FIGURE 3.8 – diagramme de séquence relatif au scénario "Manage system settings"

4.2.2 Diagrammes de séquences relatifs à l'utilisateur

La figure 3.9 illustre le diagramme de séquence relatif au scénario "S'inscrire"

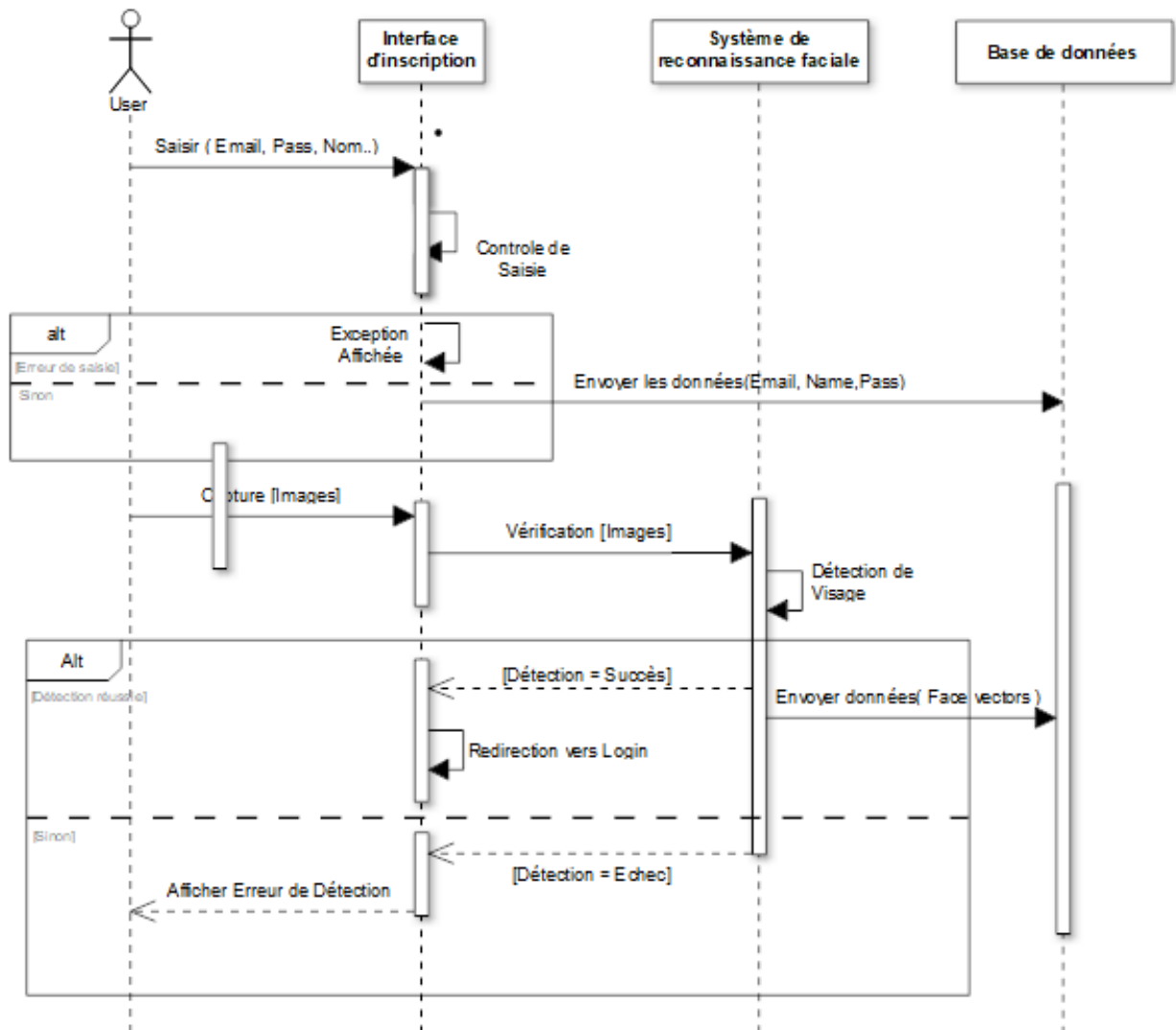


FIGURE 3.9 – diagramme de séquence relatif au scénario "Signup"

La figure 3.10 illustre le diagramme de séquence relatif au scénario "S'authentifier"

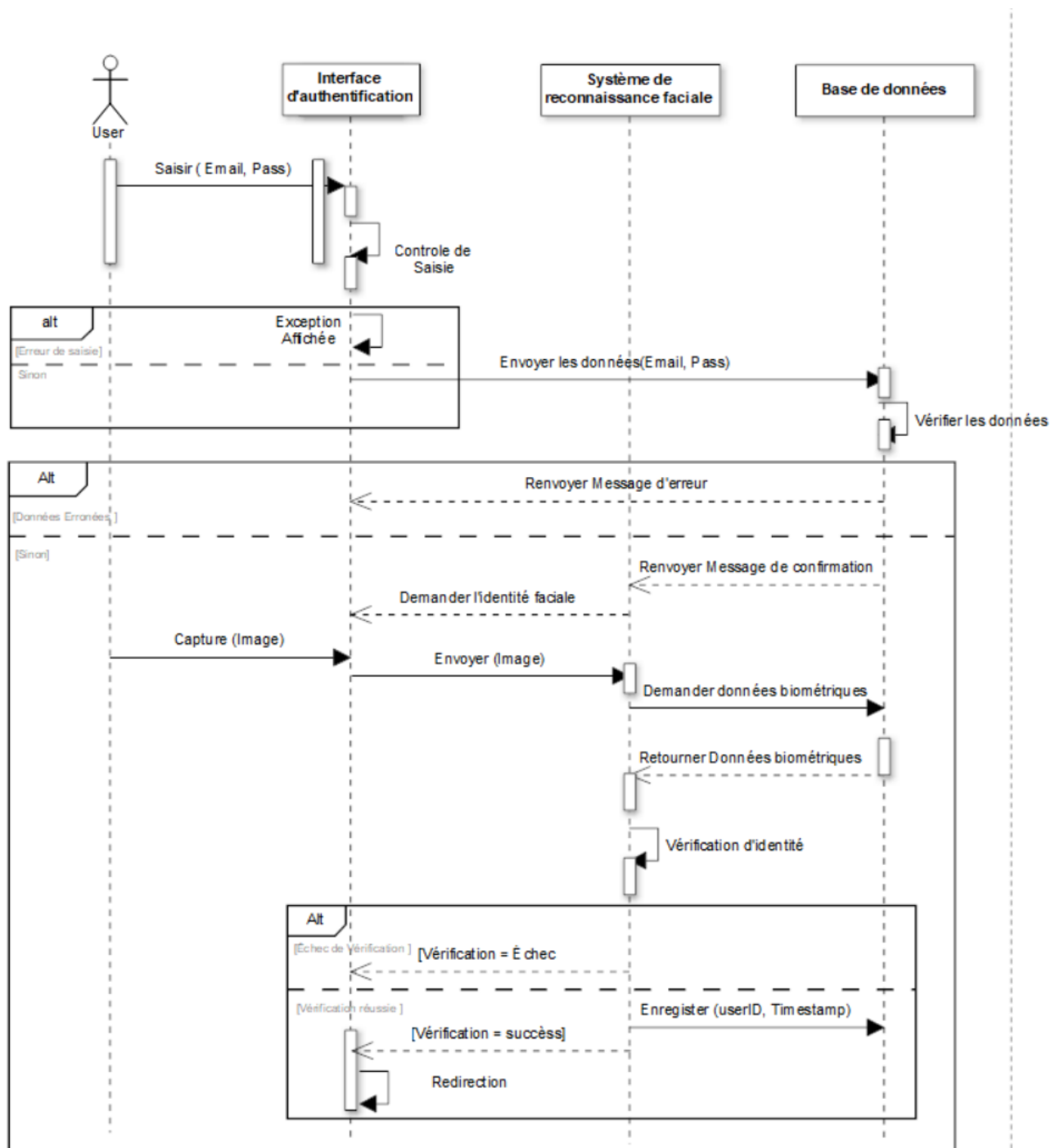


FIGURE 3.10 – Diagramme de séquence relatif au scénario "Login - User"

La figure 3.11 illustre le diagramme de séquence relatif au scénario "Gérer les paramètres du compte"

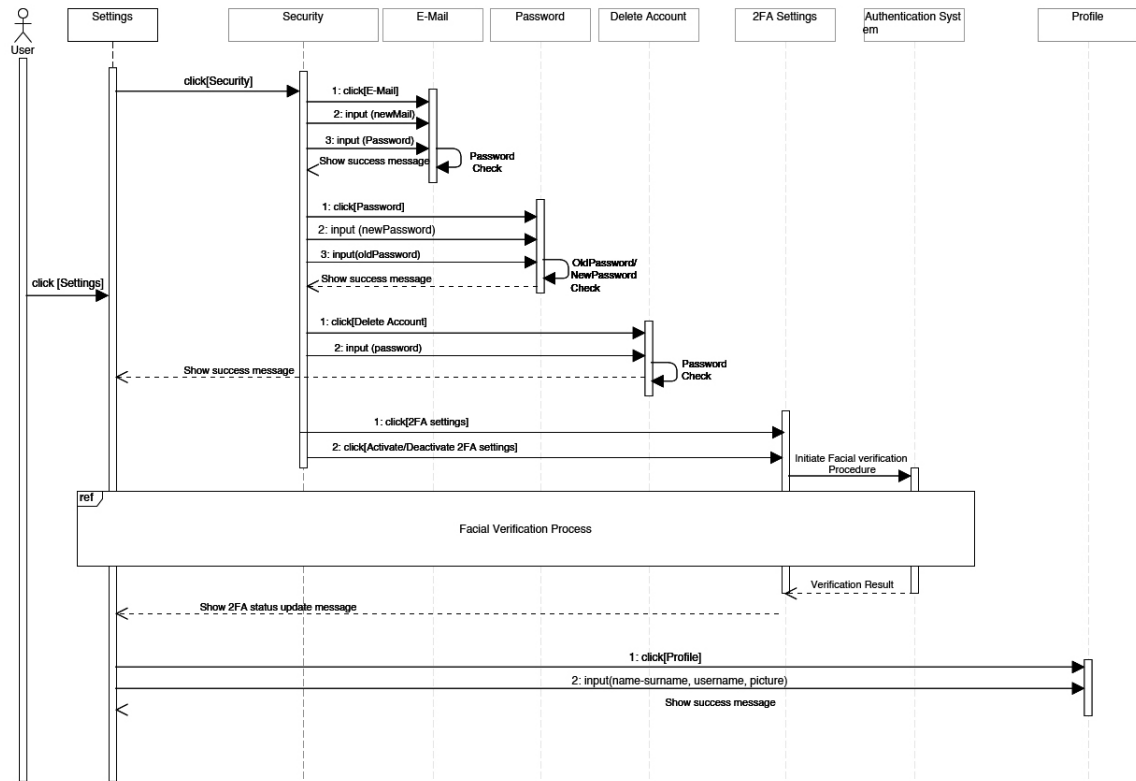


FIGURE 3.11 – Diagramme de séquence relatif au scénario "Configure account settings"

La figure 3.12 illustre le diagramme de séquence relatif au scénario "Upload Fichier"

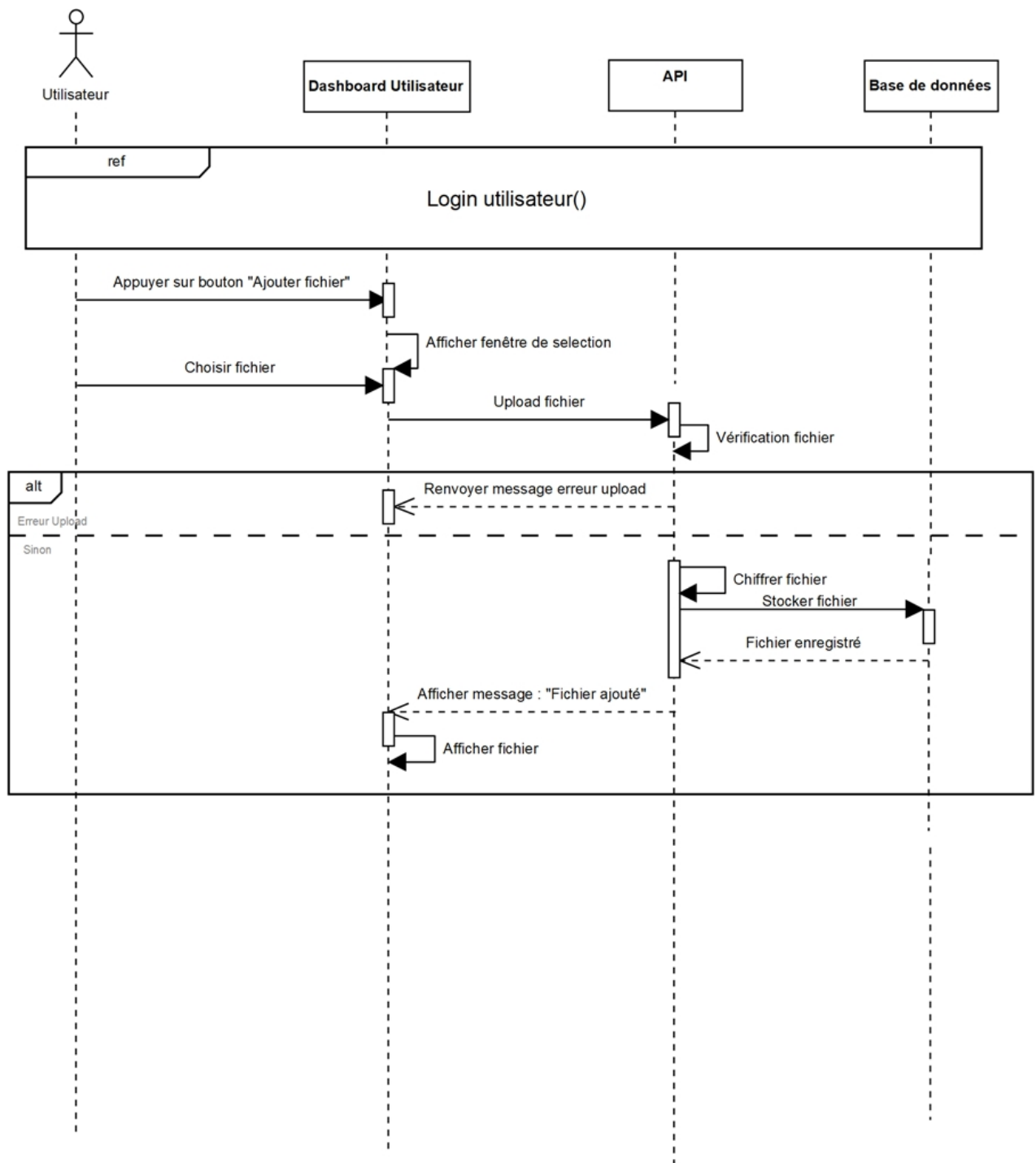


FIGURE 3.12 – Diagramme de séquence relatif au scénario "Upload File"

La figure 3.13 illustre le diagramme de séquence relatif au scénario "Télécharger Fichier"

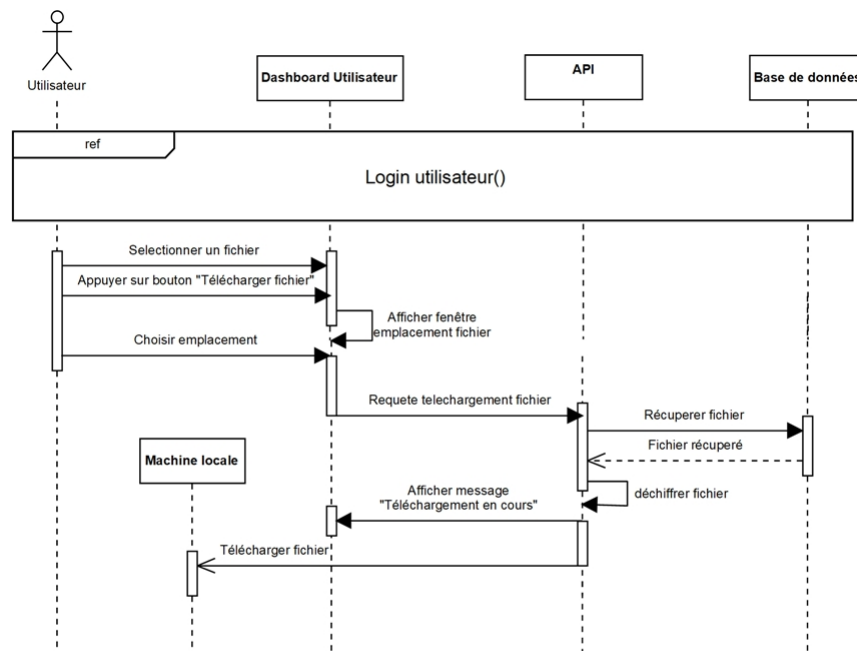


FIGURE 3.13 – Diagramme de séquence relatif au scénario "Download"

La figure 3.14 illustre le diagramme de séquence relatif au scénario "Consulter Fichier"

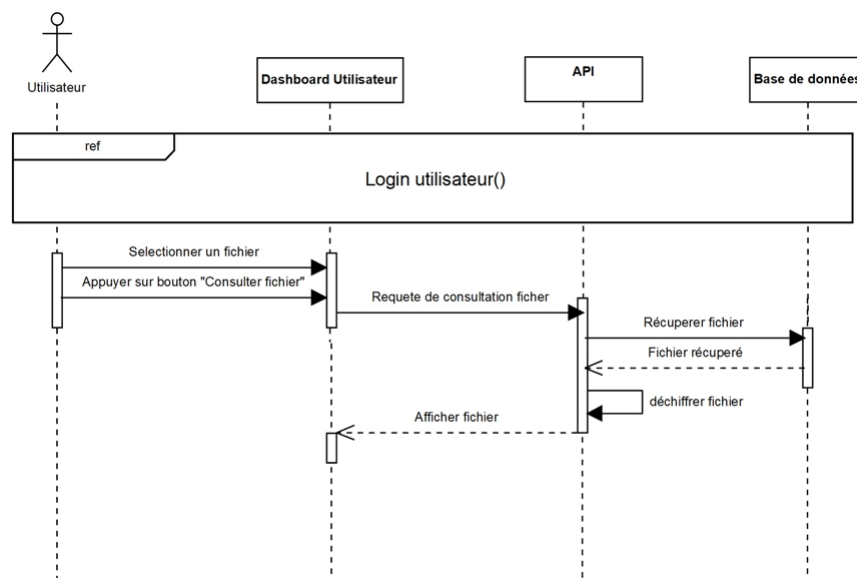


FIGURE 3.14 – Diagramme de séquence relatif au scénario "View File"

La figure 3.15 illustre le diagramme de séquence relatif au scénario "Supprimer Fichier"

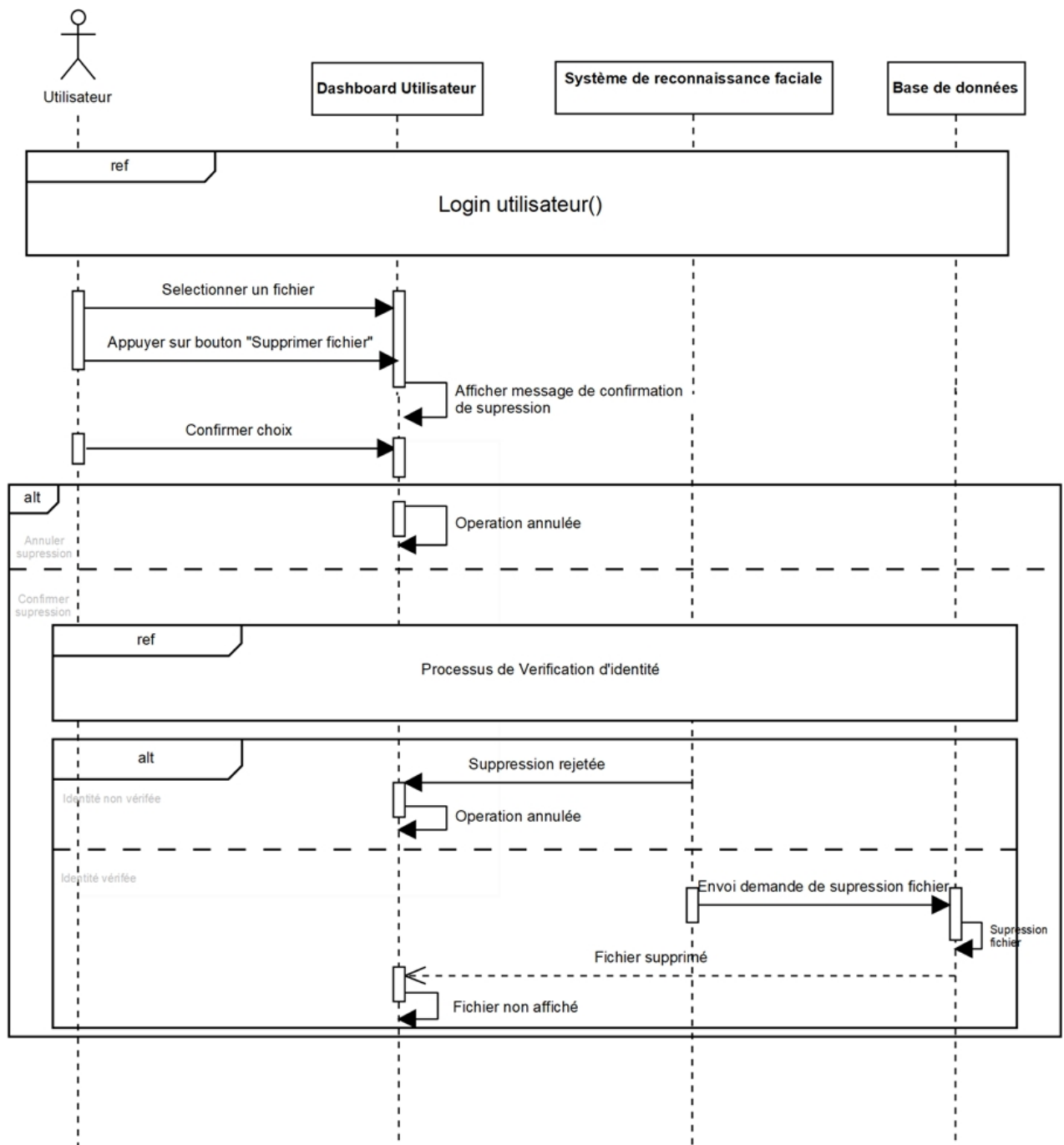


FIGURE 3.15 – Diagramme de séquence relatif au scénario "Delete File"

5 Conclusion

Dans ce chapitre, nous avons présenté le langage de modélisation UML avec ses caractéristiques et ses avantages, ainsi que les besoins fonctionnels et non fonctionnels de notre application. Nous avons accompagné ces besoins par des diagrammes de cas d'utilisation et des descriptions textuelles pour clarifier les fonctionnalités de notre application. Dans la deuxième partie, nous avons décrit en détail la conception de notre application. Nous avons présenté la conception de la vue statique et dynamique en utilisant des diagrammes de classes et de séquence. Cette partie a pour but de fournir une vue globale de l'architecture de notre application. Dans le prochain chapitre intitulé "Réalisation et tests", nous allons présenter les outils que nous avons utilisés pour réaliser notre application et nous fournirons des captures d'écran de notre application en action pour illustrer son fonctionnement.

Réalisation et tests

Sommaire

1	Introduction	47
2	Environnement du travail	47
2.1	Environnement matériel	47
2.2	Environnement logiciel	48
3	Implémentation	53
3.1	Partie web	54
4	Détection et vérification des utilisateurs	57
4.1	Phase de pré-traitement	57
4.2	Phase de vérification	57
5	Conclusion	58

1 Introduction

La phase de réalisation est une étape importante dans le processus de développement logiciel. Cette phase consiste à traduire le modèle UML en code source et à tester le système pour s'assurer qu'il fonctionne correctement.

La première étape de la phase de réalisation consiste à traduire le modèle UML en code source. Cela implique souvent l'utilisation d'outils de génération de code UML qui permettent de créer automatiquement le code à partir du modèle. Ces outils génèrent souvent un squelette de code qui peut être complété manuellement.

La phase de test est une étape cruciale dans le processus de développement logiciel. Elle consiste à s'assurer que le système fonctionne correctement et répond aux exigences spécifiées. Les tests peuvent être effectués à différents niveaux, allant de la validation des exigences à la vérification de la conformité du code avec le modèle UML.

Dans ce chapitre, nous commençons par présenter l'environnement matériel et logiciel dans lequel les travaux ont été réalisés. Ensuite, nous décrivons en détail les travaux achevés à travers des captures d'écran illustratives.

2 Environnement du travail

Cette partie comprend une description détaillée des outils et des logiciels que nous avons sélectionnés en fonction de nos besoins, accompagnée d'une explication justifiant nos choix.

2.1 Environnement matériel

Pour pouvoir développer l'application, nous avons utilisé un laptop Asus doté des caractéristiques suivantes :

- Processeur : Intel Core i5 9ème génération
- RAM : 20.00 GO.
- Disque dur : 1 TO.
- Système d'exploitation : Windows 10 famille.
- Carte graphique : Nvidia Geforce 1650

2.2 Environnement logiciel

Afin de garantir la réalisation et le bon fonctionnement de notre projet, nous avons utilisé les logiciels suivants :

2.2.1 PaceStar UML

PaceStar UML est un outil de modélisation UML (Unified Modeling Language) qui permet aux développeurs de concevoir et de visualiser des systèmes logiciels de manière graphique. Il offre une interface conviviale et des fonctionnalités puissantes pour la création de diagrammes UML, y compris les diagrammes de classes, de séquence, d'activité, de composants, de déploiement, etc. PaceStar UML est un outil puissant et polyvalent qui aide les développeurs à concevoir, visualiser et documenter efficacement les systèmes logiciels en utilisant le langage de modélisation UML [15]. La figure 4.1 illustre le logo de PaceStar UML.



FIGURE 4.1 – Logo de PaceStar UML

2.2.2 Jupyter Notebook

Jupyter Notebook est une application web open-source qui permet de créer et de partager des documents interactifs contenant du code, des visualisations et du texte explicatif. Il est largement utilisé dans les domaines de la science des données, de l'apprentissage automatique, de la recherche académique et de l'éducation pour créer des rapports, des tutoriels et des présentations interactives. Les notebooks Jupyter sont facilement partageables et collaboratifs. Les utilisateurs peuvent partager leurs notebooks avec d'autres en les exportant dans différents formats, tels que HTML, PDF ou diaporama, ou en les publiant sur des plateformes en ligne telles que GitHub, JupyterHub ou Jupyter Notebook Viewer. Cela favorise la collaboration, la reproductibilité et la transparence dans le travail scientifique et analytique [16]. La figure 4.2 illustre le logo de Jupyter Notebook.



FIGURE 4.2 – Logo de Jupyter Notebook

2.2.3 Python

Python est un langage de programmation populaire et très utilisé dans différents domaines tels que la science des données, l'intelligence artificielle, le développement web et le scripting système. Il est apprécié pour sa syntaxe simple et facile à comprendre, ce qui en fait un choix idéal pour les débutants en programmation. Python dispose également d'une grande communauté de développeurs actifs qui contribuent constamment à son développement et à son amélioration. En outre, Python est open source, ce qui signifie qu'il est disponible gratuitement et que tout le monde peut y contribuer. En somme, Python est un langage de programmation flexible, puissant et largement utilisé qui offre de nombreuses possibilités pour les développeurs de tous niveaux [17]. La figure 4.3 illustre le logo de Python.



FIGURE 4.3 – Logo de Python

2.2.4 Tensorflow

TensorFlow est une bibliothèque open-source développée par Google, utilisée pour créer et déployer des modèles d'apprentissage automatique et en profondeur. En raison de sa flexibilité et de ses performances élevées, TensorFlow est devenu l'une des bibliothèques les plus populaires pour l'apprentissage automatique. Que ce soit pour la classification

d'images, la génération de texte ou la prédiction de séries temporelles, TensorFlow offre des outils puissants pour répondre à une grande variété de besoins en matière d'IA [18]. La figure 4.4 illustre le logo de TensorFlow.



FIGURE 4.4 – Logo de TensorFlow

2.2.5 Keras

Keras est une bibliothèque haut niveau pour Python qui simplifie également la création et la formation de modèles d'apprentissage automatique et d'apprentissage en profondeur. Avec son interface conviviale, Keras permet aux développeurs de se concentrer sur la conception de modèles sans se soucier des détails de bas niveau. En tant que couche d'abstraction au-dessus de bibliothèques sous-jacentes telles que TensorFlow et Theano, Keras facilite la création de modèles complexes tout en offrant une grande flexibilité pour les utilisateurs avancés [19]. La figure 4.5 illustre le logo de Keras.



FIGURE 4.5 – Logo de Keras

2.2.6 OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque open-source de traitement d'images et de vision par ordinateur, initialement développée par Intel en 1999. Elle

est conçue pour aider les développeurs à créer des applications de traitement d'images et de vision par ordinateur efficaces et rapides. OpenCV est écrite en C++ mais dispose d'interfaces pour différents langages de programmation tels que Python et Java. Elle est utilisée pour diverses applications telles que la détection de visages, la reconnaissance de formes, la surveillance, la réalité augmentée, etc. OpenCV fournit des fonctions pour lire, écrire, afficher et traiter des images et des vidéos en temps réel. Elle prend également en charge différents formats d'image et de vidéo, ce qui en fait une bibliothèque de traitement d'images très flexible pour les développeurs [20]. La figure 4.6 présente le logo d'OpenCV.



FIGURE 4.6 – Logo d'OpenCV

2.2.7 Flask

Flask est un framework web léger et flexible pour Python. Il est conçu pour créer des applications web rapidement et facilement en utilisant des concepts simples et une structure modulaire. Flask suit le paradigme de la programmation orientée routes, où les différentes fonctionnalités de l'application sont associées à des URL spécifiques, appelées routes. Les développeurs définissent des fonctions appelées "vues" qui sont exécutées lorsque l'application reçoit une requête HTTP sur une route particulière. Ces vues peuvent générer des réponses dynamiques à partir des données enregistrées dans la base de données de l'application, ou retourner des templates HTML pour afficher des pages web. Flask est un framework web minimaliste et puissant qui simplifie le processus de développement d'applications web en fournissant une structure de base et une grande flexibilité. Il est idéal pour les petites et moyennes applications web, ainsi que pour les projets qui nécessitent une approche modulaire et légère [21]. La figure 4.7 illustre le logo de Flask.



FIGURE 4.7 – Logo de Flask

2.2.8 SQLite

SQLite est un système de gestion de base de données relationnelle qui utilise SQL (Structured Query Language) pour gérer et manipuler des données stockées localement dans des fichiers de base de données. Il est largement utilisé dans le développement de logiciels, les applications mobiles et les applications web qui nécessitent un stockage local de données.

Dans SQLite, vous pouvez utiliser des requêtes SQL pour effectuer des opérations telles que la récupération, l'ajout, la modification ou la suppression de données dans la base de données. Les développeurs peuvent utiliser des commandes telles que SELECT, INSERT, UPDATE, DELETE pour interagir avec les données stockées dans la base de données SQLite. Les avantages de SQLite résident dans sa simplicité, sa légèreté et sa facilité d'intégration dans divers projets. Étant une bibliothèque C embarquée, SQLite ne nécessite aucune configuration de serveur, ce qui en fait une option idéale pour les applications nécessitant un stockage de données local sans l'overhead d'un serveur de base de données externe [22]. La figure 4.8 illustre le logo de SQLite.



FIGURE 4.8 – Logo SQLite

2.2.9 React

React est une bibliothèque JavaScript open source développée par Facebook, utilisée pour la construction d'interfaces utilisateur (UI) interactives et dynamiques. Conçu pour simplifier le développement des applications web complexes, React repose sur un modèle de programmation déclaratif qui permet aux développeurs de créer des composants réutilisables encapsulant le code HTML, JavaScript et CSS associé à une partie de l'interface utilisateur. Ce modèle favorise la modularité, la maintenabilité et la réutilisabilité du code, ce qui facilite la construction d'applications évolutives et robustes. React est souvent utilisé en combinaison avec d'autres bibliothèques et frameworks, tels que Redux pour la gestion de l'état de l'application et React Router pour la gestion de la navigation. Cette combinaison de technologies permet aux développeurs de construire des applications web robustes et performantes, tout en bénéficiant des avantages de la modularité, de la réutilisabilité et de la performance offerts par React [21]. La figure 4.9 illustre le logo de React.

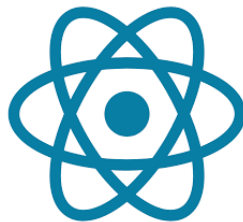


FIGURE 4.9 – Logo de la bibliothèque React

Après avoir présenté l'environnement de travail dans la première section, la deuxième section est concentrée sur les détails de la mise en oeuvre de la solution et de l'implémentation des concepts discutés.

3 Implémentation

Cette application compte deux parties de réalisation.

- Partie web : Destinée à présenter le déroulement de l'expérience utilisateur et administrateur lors de leur interaction avec notre plateforme.
- Partie détection et reconnaissance : Focalisée sur le processus de la détection et la vérification des utilisateurs grâce à notre model intelligent de reconnaissance faciale implémentée dans l'application web.

3.1 Partie web

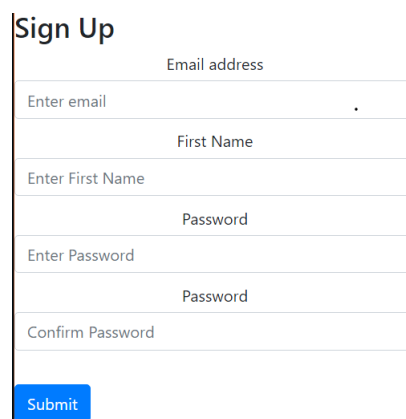
L'interface graphique est une partie essentielle de la création d'une application web conviviale offrant une expérience utilisateur agréable lors de la navigation. Elle peut faire toute la différence entre deux applications ayant les mêmes fonctionnalités. Dans cette partie, nous présenterons l'expérience en tant que utilisateurs et administrateur lors de leur interaction avec notre plateforme. Nous inclurons également des captures d'écran des points clés de cette partie pour une meilleure compréhension.

3.1.1 Expérience utilisateur

1. Interface d'authentification :

Dans la page d'authentification, l'utilisateur a le choix de s'inscrire ou de se connecter s'il possède déjà un compte. Le processus d'inscription est simple et intuitif, nécessitant de l'utilisateur des informations de base telles que son nom, son adresse e-mail et un mot de passe sécurisé. Une fois ces informations fournies, l'utilisateur est invité à capturer des photos de son visage selon les instructions du système de reconnaissance faciale. Ces images sont essentielles pour renforcer la sécurité de son compte. Une fois les données biométriques sont enregistrées avec succès, l'utilisateur est redirigé vers la page de connexion, le portail final à franchir pour accéder à la plateforme et bénéficier de ses fonctionnalités.

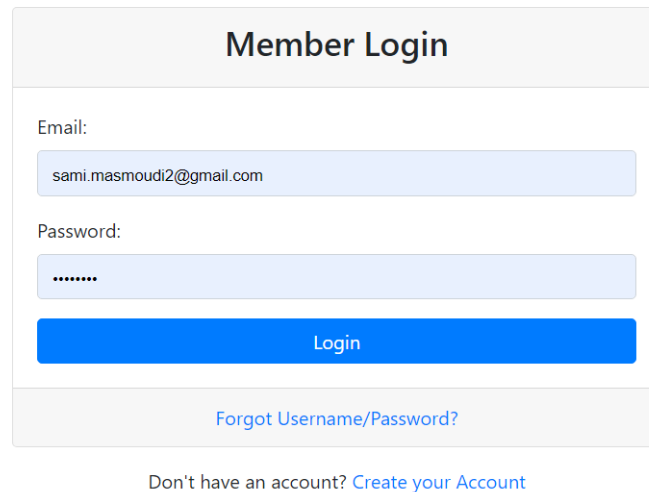
La figure 4.10 illustre l'interface d'inscription.



The image shows a web form titled "Sign Up". It contains five input fields: "Email address" with a placeholder "Enter email", "First Name" with a placeholder "Enter First Name", "Password" with a placeholder "Enter Password", another "Password" field with a placeholder "Confirm Password", and a blue "Submit" button at the bottom.

FIGURE 4.10 – Interface d'authentification

La figure 4.11 illustre l'interface de connexion.



The image shows a 'Member Login' form. It has a title 'Member Login' at the top. Below it, there are two input fields: 'Email:' with the value 'sami.masmoudi2@gmail.com' and 'Password:' with masked characters '*****'. A blue 'Login' button is below the password field. At the bottom, there is a link 'Forgot Username/Password?'. Below the form, there is a text 'Don't have an account?' followed by a link 'Create your Account'.

FIGURE 4.11 – Interface d'authentification

2. Interface principale : Notre application offre une interface "user-friendly" qui permet à l'utilisateur de stocker et organiser ses fichiers en ligne de manière sécurisée et pratique. Une fois qu'il s'est authentifié, l'utilisateur est désormais capable de consulter, ajouter, télécharger ou supprimer des fichiers. La suppression des fichiers est irréversible, c'est pourquoi que ça nécessite la vérification faciale pour confirmer le choix de l'internaute.

La figure 4.12 illustre l'interface principale de la plateforme.

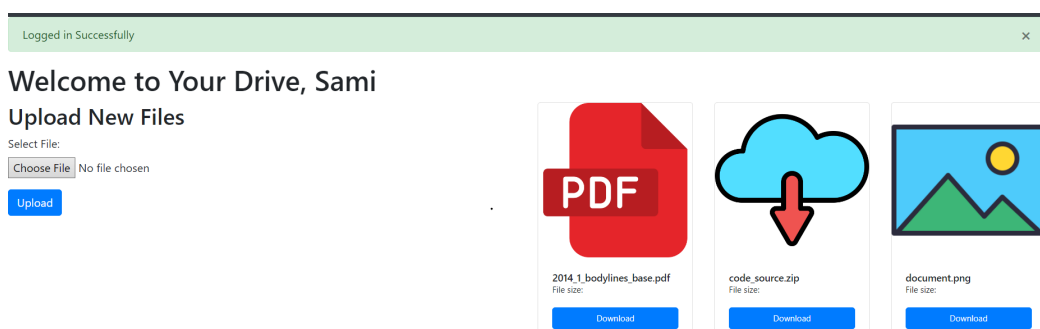


FIGURE 4.12 – Interface principale de la plateforme

La figure 4.13 présente l'exécution du cas d'utilisation ajout des fichiers

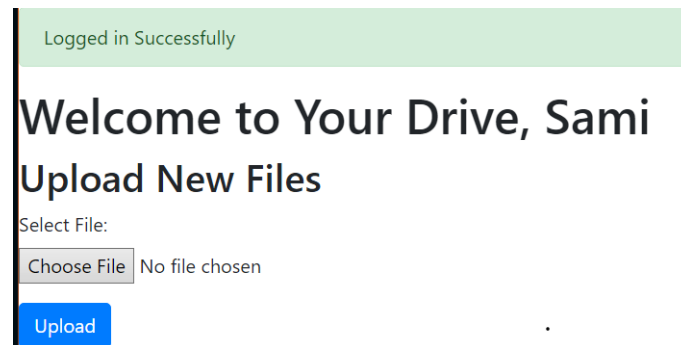


FIGURE 4.13 – Interface d'ajout des fichiers

3.1.2 Expérience adminstrateur

1. Processus de connexion : Dans la page d'authentification, l'administrateur est invité à capturer des photos de son visage selon les instructions du système de reconnaissance faciale pour accéder au la tableau de bord administratif.
2. Interface du tableau de bord administratif : Notre application offre une interface détaillée permettant à l'administrateur de modifier les paramètres globales de l'application, et consulter les informations des utilisateurs inscrits dans la plateforme. L'administrateur peut également gérer les réglages de stockage pour chaque utilisateur et l'informer de ce changement dans les termes d'utilisation du service.

La figure 4.14 illustre l'interface du tableau de bord adminstratif.

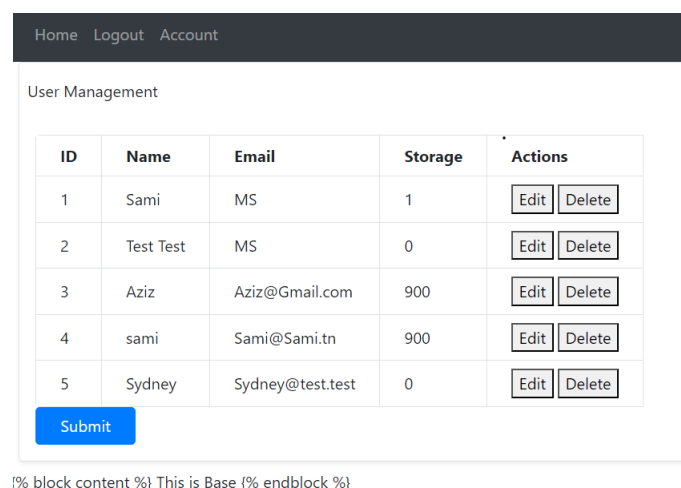


FIGURE 4.14 – Interface du tableau de bord adminstratif.

4 Détection et vérification des utilisateurs

La détection et vérification de l'identité de l'utilisateur au moment de connexion reposent principalement sur deux phases :

4.1 Phase de pré-traitement

La phase de pré-traitement des images est une étape essentielle dans le processus de détection et vérification de l'identité de l'utilisateur. Avant de comparer les images faciales capturées au moment de connexion, l'application prétraite les images capturées par la webcam et les images enregistrées de l'utilisateur concerné. Cela comprend le redimensionnement des images pour qu'elles aient la même taille et la mise à l'échelle des valeurs des pixels pour qu'elles soient comprises entre 0 et 1.

4.2 Phase de vérification

Une fois la phase de pré-traitement est terminée, un modèle de réseau neuronal spécifiquement conçu pour la reconnaissance faciale est chargé. Ce modèle a été entraîné sur un grand ensemble d'images pour apprendre à extraire des caractéristiques discriminantes des visages. Ensuite, le module de vérification calcule une métrique de similarité entre les vecteurs de caractéristiques des images capturées et ceux des images enregistrées qui sert à définir à quel point les images doivent être similaires pour être considérées comme correspondantes. Si la métrique de similarité dépasse le seuil, l'utilisateur est considéré comme vérifié, sinon, il est rejeté. Finalement, l'interface web affiche un message indiquant si l'utilisateur est vérifié avec succès. Si c'est le cas, il peut être autorisé à accéder au système, sinon, l'accès à la plateforme est refusé et il est invité à réessayer.

La figure 4.18 illustre le déroulement du processus de la reconnaissance faciale.

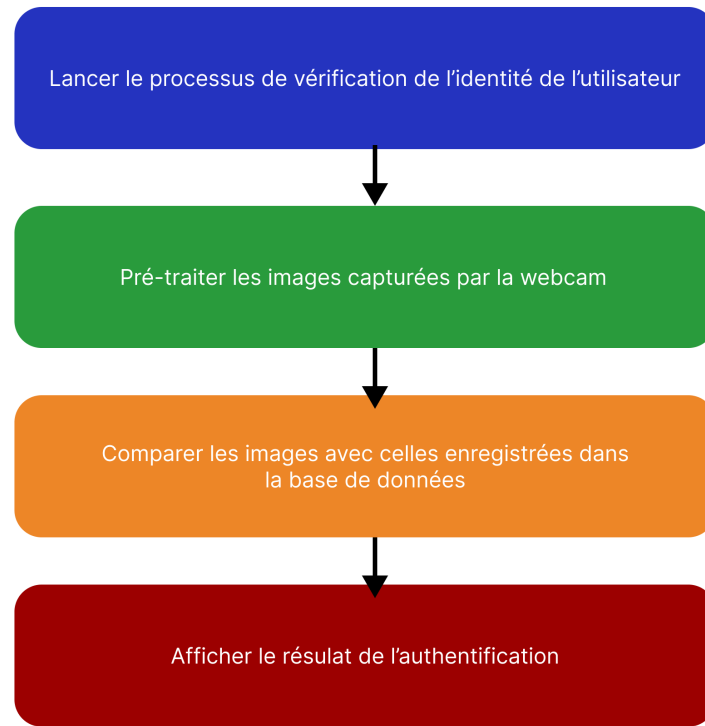


FIGURE 4.15 – Le déroulement du processus de la reconnaissance faciale

5 Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel ainsi que le déroulement de la partie web de point de vue de l'administrateur et de l'utilisateur. Avec ce chapitre, nous concluons la phase d'implémentation de notre solution.

Conclusion générale et perspectives

Au fil de notre rapport, nous avons exploré en détail la conception, l'implémentation de notre plateforme. Notre objectif principal est de garantir la confidentialité du stockage des fichiers tout en renforçant l'authentification des utilisateurs. Nous offrons aux utilisateurs une solution fiable et efficace pour la gestion sécurisée de leurs données numériques.

En conclusion, notre plateforme, menée d'un système de vérification d'identité assez robuste, représente une contribution significative à l'évolution du secteur de stockage des fichiers en ligne, en offrant une solution radicale face aux défis croissants de la préservation de la vie privée et la sécurité des données. Quant aux perspectives, il serait intéressant de poursuivre le développement de cette application en y ajoutant des fonctionnalités telles que le partage sécurisé de fichiers entre utilisateurs. De plus, nous chercherons à améliorer davantage la performance du modèle de reconnaissance faciale en le rendant invariant à toutes les circonstances, telles que la faible luminosité, les images de qualité basse ou les changements au niveau des traits de visage des utilisateurs. Nous visons aussi à optimiser ce modèle pour raccourcir les délais d'analyse et de traitement des images. Par ailleurs, l'intégration d'un système de gestion de fichiers permettra de faciliter l'organisation des fichiers par l'utilisateur et assurera un accès rapide à ces derniers. Enfin, l'extension future de notre projet pourrait inclure l'implémentation de la technologie blockchain pour renforcer encore davantage la sécurité et la traçabilité des échanges de fichiers sur notre plateforme. Cette idée de projet est extrêmement extensible, offrant ainsi un potentiel considérable pour répondre aux besoins en constante évolution de nos utilisateurs.

Bibliographie

- [1] me.pcmag.com/en/file-sync-backup/19066/dropbox,
- [2] apps.apple.com/us/app/google-drive/id507874739,
- [3] www.tomsguide.com/reviews/microsoft-onedrive-review,
- [4] researchgate.net/figure/Illustration-des-differentes-etapes-du-pretraitement-Extraction-des-caracteristiques_fig1_238728746,
- [5] medium.com/@appstud/lab-appstud-reconnaissance-de-visages-grace-au-machine-learning-85a612b368db,
- [6] www.researchgate.net/figure/Key-Features-from-a-human-face_fig2_337306950,
- [7] R. Kaur et E. Himanshi, «Face recognition using Principal Component Analysis», Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015, p. 585-589, juill. 2015, doi : 10.1109/IADCC.2015.7154774.
- [8] www.dreamstime.com/woman-face-recognition-biometric-verification-woman-face-recognition-biometric-verification-concept-image107714380,
- [9] "VGGFace2 : A dataset for recognising faces across pose and age" Paper by Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, Andrew Zisserman, submitted on 23 Oct 2017,
- [10] www.researchgate.net/figure/ResNet-50-Model-where-the-input-feature-classifier-layer-and-its-layers-are-normalized_fig2_350711158/download?_ip=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6Il9kaXJlY3QiLCJwYWdlIjoilX2RpcmVjdCJ9fQ,
- [11] "Siamese Neural Networks for One-shot Image Recognition" Paper by Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, Department of Computer Science, University of To-

ronto. Toronto, Ontario, Canada,

- [12] "Siamese Neural Networks for One-shot Image Recognition" Paper by Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, Department of Computer Science, University of Toronto. Toronto, Ontario, Canada,
- [13] fr.wikipedia.org/wiki/Cryptographie_sym%C3%A9trique,
- [14] fr.wikipedia.org/wiki/Cryptographie_asym%C3%A9trique,
- [15] www.pacestar.com/uml/ud60ug.pdf,
- [16] en.wikipedia.org/wiki/Project_jupyter,
- [17] en.wikipedia.org/wiki/Python%28programming_language%29,
- [18] en.wikipedia.org/wiki/TensorFlow,
- [19] en.wikipedia.org/wiki/Keras,
- [20] en.wikipedia.org/wiki/OpenCV,
- [21] [en.wikipedia.org/wiki/Flask_{\(web_f\)ramework}](http://en.wikipedia.org/wiki/Flask(web_framework)),
- [22] [en.m.wikipedia.org/wiki/File :SQLite370.svg](http://en.m.wikipedia.org/wiki/File:SQLite370.svg),
- [23] [en.wikipedia.org/wiki/React_{\(JavaScript_l\)ibrary}](http://en.wikipedia.org/wiki/React(JavaScript_library))

