المعهد العالي للدراسات التكنولوجيــة في المواصلات بتونس
Institut Supérieur des Etudes Technologiques en Communications de Tunis

Department: STIC

Reference :

Applied License in Information and Communication Sciences and Technologies

Option:

End of Studies Project

Face recognition system

Realized by: Eya CHAIBNI
& Montaha REBHI

Class: RST B

Superviser(s): Afef GUEIDI
Rabii MALLEK

Host Company: Telcotec

Academic year: 2022-2023

Dedications

“

I dedicate this important moment in my life to:

My dear mother and father, who gave me their all and sacrificed everything for me to reach this point. Nothing I ever do can repay the debt I owe them. May this work rise up to their expectations, for it is only the beginning.

My brothers, my lifelong companions and trusted allies, whose presence has given me the confidence to overcome challenges and reach new heights.

My teachers, from first grade until this point I am forever grateful for every letter they taught me.

Every single one of my friends, who have always encouraged me, and to whom I wish even greater success.

To all those I love, those that provided nothing but unconditional love, affection and encouragement; those that stuck by my side during the toughest of times.

”

Eya

Dedications

“

I dedicate this important moment in my life to:

My dear grand father ,your passing has left a void in my heart, but the impact you made on my education and personal growth will forever remain.

My dear father, thank you for tireless sacrifices and endless encouragement. Your belief in me has been a constant motivation to strive for excellence.

My dear mother, your love, strength, and nurturing nature have shaped me into the person I am today. Thank you for always being there to encourage me.

My beloved sister and brother, you have been my closest allies, confidants, and sources of inspiration.

My teachers, from first grade until this point I extend my deepest appreciation for your guidance and wisdom .

My friends, I wish you continued success and express my heartfelt gratitude for your unconditional love and support.

To all those I love, those that provided nothing but unconditional love and encouragement; those that stuck by my side during the toughest of times.

”

Montaha

Acknowledgments

It is with sincere pleasure that we reserve these lines to express our thanks to the people who helped us finalize this project during a significant period in our academic career.

To begin with, we would like to thank the company for providing us with such an informative experience. We want to thank the founder and CTO of TELCOTEC **Mr. Taieb MASMOUDI** as well as our direct supervisor **Mr. Rabii MALLEK** for his assistance throughout the work as well as the confidence he has given us.

Her critical eye was very valuable in structuring the work and improving the quality of the different sections. We express our deepest thanks to our supervisor **Mrs. Afef GUEIDI** for her continuous assistance throughout the project. Her motivation, encouragement, and experience helped us overcome any problem. As she gave us the techniques to be well organized during a long project period.

May all the members of the jury find here the expression of our gratitude for having accepted to evaluate our work.

Finally, I would like to express my sincerest thanks and gratitude towards the entire academic and administrative staff of ISET'Com as well as my colleagues and alumni of ISET'Com for their assistance and support.

Contents

General Introduction **1**

1 Preliminary Study **2**
1.1 Host Company: TELCOTEC 3
1.1.1 General Presentation 3
1.1.2 Functional Organization 3
1.1.3 TELCOTEC's services 4
1.2 Project Context 4
1.2.1 Problematic 4
1.2.2 Study of existing systems 5
1.2.3 Critique of the existing systems 5
1.2.4 The proposed solution 6
1.2.5 Workflow of the proposed solution 6
1.3 Project planning 7
1.4 The work methodology 8
1.4.1 The traditional method 8
1.4.2 The Agile method 8
1.4.3 Comparison of the traditional and Agile Approaches 9
1.4.4 Adopted methodology 9
1.4.5 Kanban Principles 10

2 State of the Art **11**
2.1 Face Recognition 12
2.2 Facial Recognition Steps 12
2.2.1 Face Detection 13
2.2.2 Image Processing 14
2.2.3 Extraction of characteristics 14
2.2.4 CNN pre-trained model 16
2.3 Recognition or verification 18
2.3.1 Euclidean distance 18
2.3.2 Cosine similarity 19

3 Requirements Analysis and Design **20**
3.1 Requirements analysis 21
3.1.1 Identifying actors 21
3.1.2 Functional requirements 21
3.1.3 Non-functional requirements 22
3.2 Prototype Interfaces 22

3.3 Design . . . 22
3.3.1 Modeling language . . . 22
3.3.2 Use case diagram . . . 23
3.3.3 Global use case's refinement . . . 24
3.3.4 Class diagram . . . 32
3.3.5 Activity diagram . . . 34

4 The Proposed Face Recognition System . . . 36
4.1 Work Environment . . . 37
4.1.1 Hardware Environment . . . 37
4.1.2 Software Environment . . . 37
4.2 Face Recognition . . . 38
4.2.1 Face Detection . . . 38
4.2.2 Characteristics Extraction . . . 39
4.2.3 Face Matching . . . 39
4.3 Proposed Approach . . . 41
4.4 The mobile and web applications . . . 43
4.4.1 Logical Architecture . . . 43
4.4.2 Frontend implementation . . . 44
4.4.3 Backend implementation . . . 49
4.5 System integration . . . 52
4.5.1 Saving image data in MongoDB . . . 52
4.5.2 Displaying data on the mobile application . . . 52

General Conclusion and Perspectives . . . 54

Annexes . . . 57

List of Figures

1.1 The company's organization. . . . 3
1.2 Workflow of the solution. . . . 7
1.3 Gantt chart 8
1.4 Classical approach VS Agile approach 9

2.1 Facial Recognition Steps Pipeline 13
2.2 Deep Neural Network Architecture 15
2.3 Architecture of a Convolutional Neural Network 16
2.4 FaceNet Model structure 17
2.5 Measurements generated by FaceNet for a face 17
2.6 The Triplet Loss[17] 18

3.1 Global Use Case Diagram 23
3.2 Log In Use Case Diagram 25
3.3 Log In Sequence Diagram. . . . 26
3.4 Admin Authentication Use Case Diagram 27
3.5 Admin Authentication Sequence Diagram 28
3.6 Face Recognition Use Case Diagram 29
3.7 Face Recognition Sequence Diagram. . . . 31
3.8 Class Diagram. . . . 33
3.9 Activity Diagram. . . . 34

4.1 Face Recognition frontal Test 40
4.2 Face Recognition non frontal Test 40
4.3 Frontal test with a mask on 41
4.4 The proposed face recognition system architecture 42
4.5 The MVC architecture 43
4.6 Explanatory map for the interfaces 45
4.7 Explanatory map for the interfaces 46
4.8 Login components 47
4.9 Home page components 47
4.10 Update User Interface 48
4.11 Update User Interface 48
4.12 Thunder client console: API 'POST' request (register) 50
4.13 Thunder client console : API 'POST' request (login) 50
4.14 Thunder client console : API 'GET' request 51
4.15 MongoDB console (1) 51
4.16 MongoDB console (2) 52

List of Tables

1.1 The Existing systems 5
1.2 Comparison of the traditional and Agile Approaches 9

2.1 Face detectors. 14

3.1 Global use case description 24
3.2 Log In Use Case description 25
3.3 Admin Authentication Use Case Description 27
3.4 Face Recognition Use Case Description 30

4.1 Characteristics of the first machine 37
4.2 Characteristics of the second machine 37
4.3 Software Environment 38
4.4 Comparison between detectors 38

List of acronyms

AI	*Artificial Intelligence*
API	*Application Programming Interface*
CNN	*Convolutional Neural Network*
CRUD	*Create, Read, Update and Delete*
DNN	*Deep Neural Network*
HTTP	*Hypertext Transfer Protocol*
ICT	*Information Communication Technologies*
IT	*Information Technology*
JSON	*JavaScript Object Notation*
LBP	*Local Binary Pattern*
MTCNN	*MultiTask Cascaded Convolutional Neural Network*
MVC	*Model View Controller*
NICT	*New Information and Communication Technologies*
RD	*Reasearch and Development*
UML	*Unified Modeling Language*

General Introduction

Keeping order and ensuring public safety is paramount in law enforcement. Law enforcement agencies now have powerful tools to enhance their efficiency and effectiveness thanks to technological advancements. One such cutting-edge innovation is a face recognition system tailored specifically for policemen and guards.

A face recognition system, integrated into a mobile application, simplifies law enforcement duties. By combining artificial intelligence and advanced algorithms, this system equips officers with a powerful tool to quickly and accurately identify suspects, wanted criminals, and missing persons. In this way, they can prevent and help solve crimes easily and proactively respond to potential threats and incidents.

This mobile application serves as a comprehensive platform, combining the capabilities of a robust face recognition system with the convenience and mobility of a handheld device. This helps policemen access current information. Integrating these tools allows law enforcement personnel to make informed decisions faster and more effectively, resulting in safer communities.

This report will be structured into four main chapters:

Firstly, we will start with the **Preliminary Study** of the project. This is where we present our host company, along with the project context and the proposed solution.

In the second chapter **State of the Art** we will discuss various theoretical concepts, introduce the selected terms for the project and explain how they interact and benefit the final goal.

The third chapter **Requirements Analysis and Design** will contain the conception of our project.

The fourth and final chapter **The Proposed Face Recognition System** focuses on implementing a face recognition system while highlighting its integration with a mobile application.

In the end, we close with a general conclusion that summarizes the work we've done and presents several perspectives.

Chapter 1

Preliminary Study

Introduction

During this chapter, we will introduce the host company and its various activities. In a later section, we will discuss the general context of the project, the problematic, and the proposed solution. Finally, we will propose the most convenient work methodology to complete our project.

1.1 Host Company: TELCOTEC

In this section of the report, information are provided regarding the company's general appearance, organizational structure and services.

1.1.1 General Presentation

Telcotec was founded in January 2015 to offer network engineering and New Information and Communication Technologies (NICT) consulting, studies, and project management services. Its strategic location, Technopark Elgazala, provides a perfect environment for growth in Information Communication Technologies (ICTs)[**1**].

1.1.2 Functional Organization

The figure(1.1) below presents the organization chart of the company TELCOTEC by hierarchical level between the various services and personnel. Our collaboration was primarily focused on the IT department.

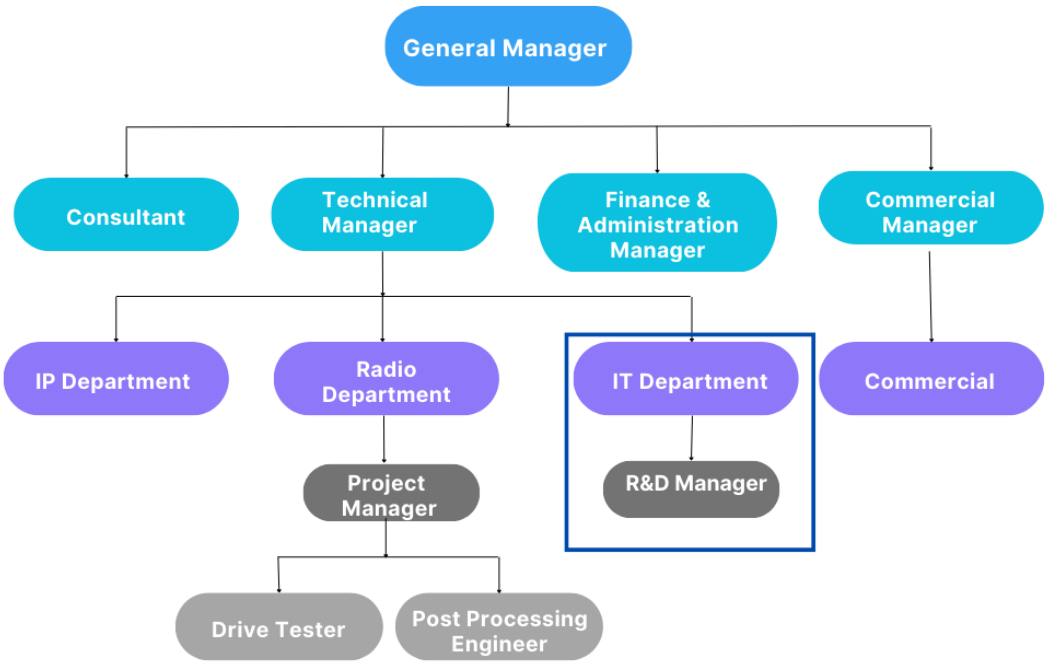


Figure 1.1: The company's organization.

1.1.3 TELCOTEC's services

TELCOTEC provides operators, regulators, and manufacturers of telecommunications equipment with high-quality engineering, consulting, and training services in the technical and management fields of telecommunications.

Among the services offered by TELCOTEC, we can mention:

- Audit of the networks: TELCOTEC helps to know with precision the strong and weak points of the infrastructures and the networks in terms of architectures, parameter setting, consistency of data, and configuration.
- Network optimization: Based on strategic decisions, it defines the values of the physical and logical factors of the infrastructures allowing it to reach the objectives and ambitions of quality of services at the lowest cost.
- Network architecture design: It determines the architectural decisions best suited to the situation and environment.
- Writing technical specifications and evaluating proposals: It is used to write the technical specifications that set out in precise terms the needs and requirements for an effective competitive bidding process in the short term and over time. It also helps to evaluate the recommendations to make the most appropriate choice according to the expectations and the context.
- Strategy consulting: It helps to find the appropriate alternatives and make the best decisions to exploit the benefits in the best way.

1.2 Project Context

In this part, we will talk about the problematic with a brief critique of existing systems, and suggest our solution.

1.2.1 Problematic

In various public places, law enforcement officers and agents are the first lines of defense for public safety and order. They are also responsible for preventing and responding to incidents. However, law enforcement officers and agents encounter significant obstacles to maintaining security and order in public places. These challenges include limited resources, rapidly evolving threats, and crowd management difficulties. To overcome these obstacles, it is important to empower officers and agents with the necessary tools and support. We can enhance their ability to ensure citizen safety and maintain order in various public spaces.

To further understand and address these challenges, it is necessary to analyze existing systems that seek to overcome them. In this study, different systems and technologies will be compared to assess their effectiveness in managing law enforcement challenges.

1.2.2 Study of existing systems

In this section, different systems and technologies will be compared to measure their effectiveness in managing law enforcement challenges, as shown in Table(1.1).

Table 1.1: The Existing systems

System	Description	Technology Used	Limits
Predictive Policing Systems [2]	This system predicts and helps prevent potential future crimes by : -Identifying high-risk places and times . -Analyzing historical crime data	- Machine learning algorithms -Data analytics	- Data Bias - Prediction Accuracy
Communication Systems[**3**]	This system ensure reliable and efficient communication for coordination and information sharing among law enforcement personnel	- Two-way radios - Mobile data terminals	- Limited Data Capacity -Coverage Issues -Security Challenges
Surveillance Systems[**4**]	A fixed camera identifies and tracks objects and faces . When a human face is detected , it records the sequence of that occurence	-Object detection with Haar cascades -Feature extraction with LBP algorithm	- Fixed camera - Manual review - Response Time .

1.2.3 Critique of the existing systems

After studying these different systems' characteristics and limitations, it is crucial to focus on accuracy, data capacity, coverage and efficiency. By addressing these aspects, our proposed solution can be optimized for improved performance and overcome potential challenges.

1.2.4 The proposed solution

In response to the specific needs of the law enforcement sector, our solution is to develop a highly efficient and accurate face recognition system, utilizing AI technology. This system will be integrated with body cameras worn by police officers or security guards, enabling real-time detection and matching of individuals.

When a suspect is identified, the system will promptly send an alert to a designated mobile application used by law enforcement personnel.This alert will provide detailed information, if available, about the matched suspect.

By leveraging this advanced technology, our solution aims to enhance the efficiency and effectiveness of law enforcement operations. It aids in the identification and apprehension of potential threats.

1.2.5 Workflow of the proposed solution

1. The Camera: The system starts with a body camera worn by a police officer or security guard. The camera captures live video feeds and images of individuals encountered during their duties.

2. Face Detection: The captured video feed is processed using AI-based face detection algorithms. These algorithms analyze video frames and identify faces.

3. Face Recognition: Once faces are detected, the system applies AI-powered face recognition algorithms to compare the captured faces with a pre-existing database of suspects. The face recognition algorithm computes a similarity score or matches the probability for each detected face.

4. Suspect Match: If a face in the captured footage matches a suspect in the database above a defined threshold, it is considered a positive match. This information triggers further actions in the system.

5. Alert Generation: The system generates an alert in real-time, containing relevant details about the identified suspect. This alert is sent as a notification to a designated mobile application used by law enforcement personnel.

6. Mobile Application: Law enforcement personnel receive the alert on their mobile application, providing them with detailed information about the suspect, such as name, photograph, criminal record, and any other relevant data available.

7. Database Management: The solution also includes a back-end system for managing the suspect database. This system allows authorized personnel (admin) to update and maintain the database, ensuring it remains up-to-date with accurate information.

The following figure(1.2) represents the workflow of our proposed solution.

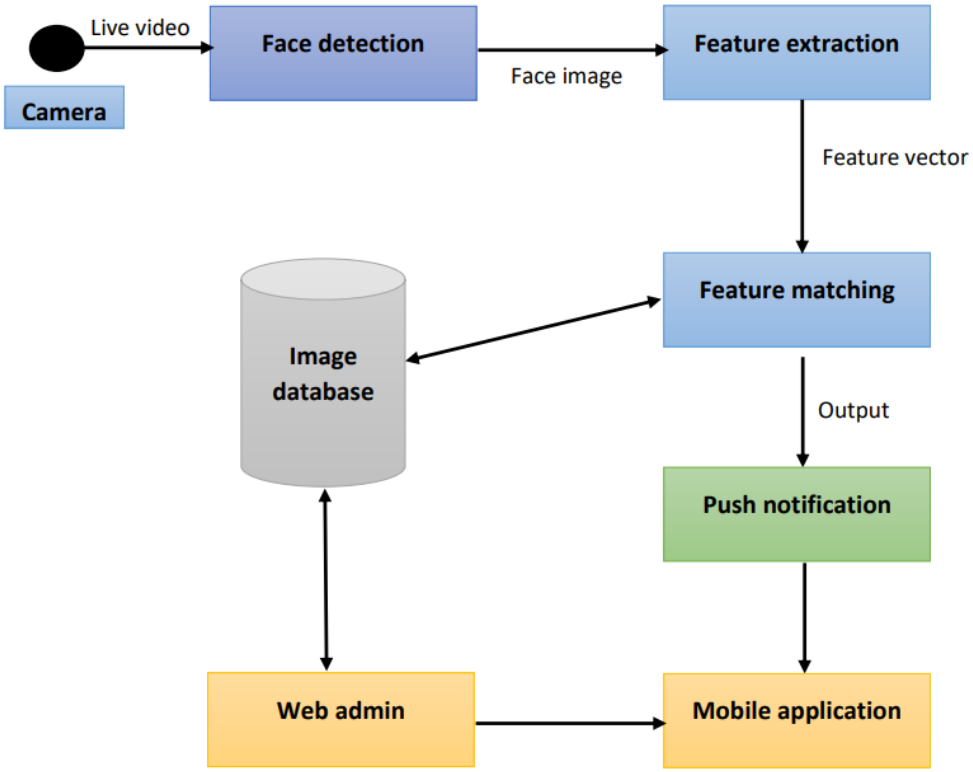


Figure 1.2: Workflow of the solution.

1.3 Project planning

The Gantt chart, commonly used in project management, is one of the most effective tools for visually representing the progress of the various tasks that a project involves.

The left-hand column of the diagram lists all the tasks to be carried out, while the header line represents the most appropriate time units for the project (days, weeks, months, etc.). Each task is marked by a horizontal bar, whose position and length indicate the start date, the duration, and the end date.

This schedule can be presented, in collaboration with the company's manager, in a GANTT chart as shown in the diagram in Figure(1.3) .

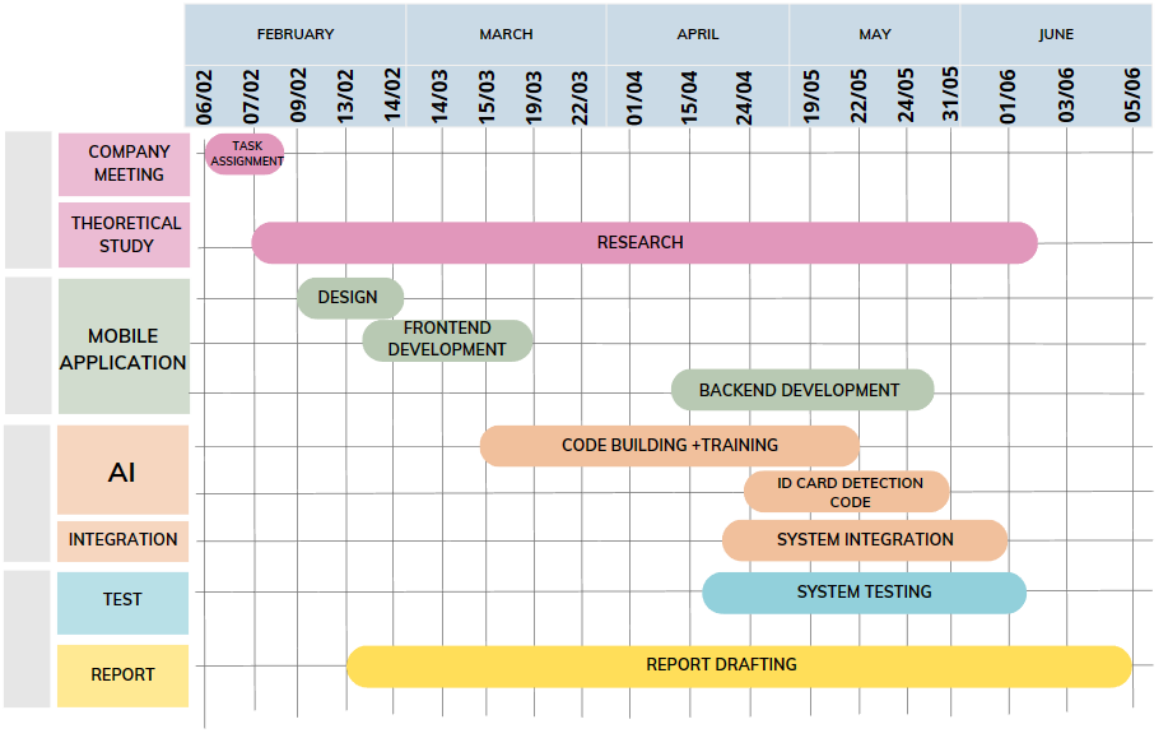


Figure 1.3: Gantt chart

1.4 The work methodology

As a good practice, it is strongly advised to select a method for structuring the project development phases to achieve optimal effectiveness. We will then compare the classical technique and the agile approach to determine which methodology is best for our project.

1.4.1 The traditional method

The Waterfall methodology ,also known as the Waterfall model , is a sequential classical development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with each phase completely wrapping up before the next phase begins[**5**].

1.4.2 The Agile method

Agile methodology is a project management framework that breaks projects down into several dynamic phases, commonly known as sprints.

The Agile framework is an iterative methodology. After every sprint, teams reflect and look back to see if there was anything that could be improved so they can adjust their strategy for the next sprint[**6**].

1.4.3 Comparison of the traditional and Agile Approaches

The following table compares the classical approach with the agile approach.

Table 1.2: Comparison of the traditional and Agile Approaches

	Classical approach	Agile approach
Strenghts	Project planning is predictive and relatively detailed from the start, allowing developers to detect potential hazards in advance	- The ability to adapt to changing needs of our customers needs. - Future users will be able to see the results of the parameterization continuously. - Good communication and responsibility of the team.
Weaknesses	Quality control at the end of the development cycle (tunnel effect). The customer discovers the finished product at the end of the project.	The lack of planning can lead to unexpected delays compared to the conventional method.

The figure(1.4) illustrates the difference between the two approaches.

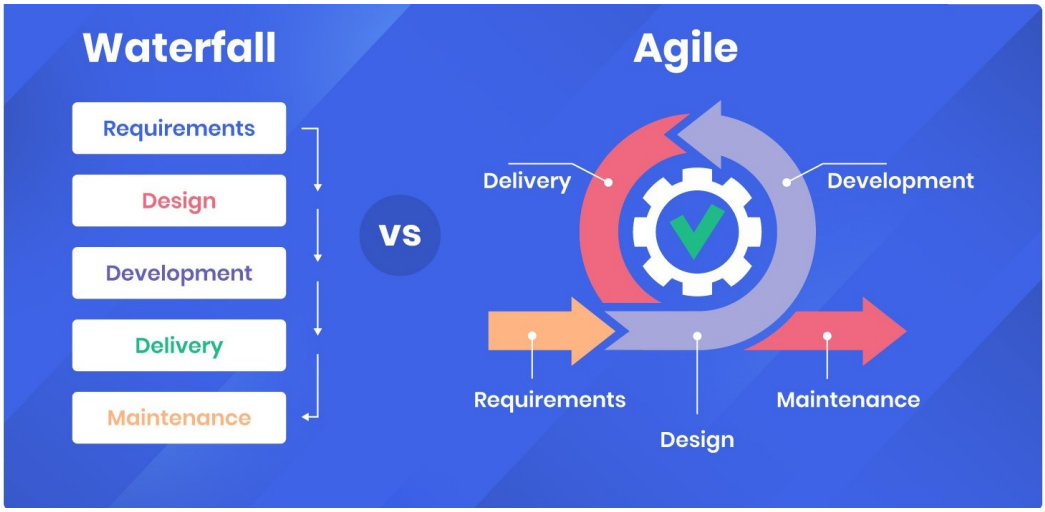


Figure 1.4: Classical approach VS Agile approach

1.4.4 Adopted methodology

In our project we adopted the Agile Kanban Method, and used Trello as a working space.(Annex D)

The Japanese word “Kanban”, meaning “visual board” or a “sign”, has been used in the sense of a process definition since the 1950s.

Kanban is a popular Lean workflow management method for defining, managing, and improving services that deliver knowledge work. It helps you visualize work, maximize efficiency, and improve continuously. Work is represented on Kanban boards, allowing you to optimize work delivery across multiple teams and handle even the most complex projects in a single environment [**7**].

1.4.5 Kanban Principles

- Start by doing what you know how to do and with what you already have: the Kanban method uses the processes already in place and encourages the improvement of the processes already in place later.
- Agree to apply incremental changes: the team must agree to improve the system in place gradually, this must be done gradually.
- Respect the current process, roles, responsibilities, and titles: to avoid rushing teams, it will be necessary to respect the roles, responsibilities, and titles of each.
- Leadership at all levels: whether employees or senior managers, all actors in the production chain who wish to implement continuous improvement must be encouraged [**8**].

Conclusion

In this first chapter, we presented the host organization and its services. We then defined the problem and analyzed the existing solutions and our proposed system. In the end, we specified the ”Kanban” methodology adopted throughout the project.

In the second chapter, we will have a more in-depth look at the different concepts involved in the project.

Chapter 2

State of the Art

Introduction

In this chapter, we will explain general notions of facial recognition, how it works, and its different steps, as well as artificial intelligence technology and its application in facial recognition.

2.1 Face Recognition

Facial recognition is a biometric process just like the recognition of fingerprints, iris, or voice. So it's identifying a person. The facial recognition system is a software application designed to automatically recognize a person's face.

Using algorithms, this application analyzes all facial features such as eye spacing, nose edges, corners of lips, ears, and chin, from an image of her face that can come from both a photo and a video[9] .

Several methods of face recognition have been proposed during the last 30 years, following two main axes: recognition from still images and recognition from a sequence of images (video).

Face recognition based on video is preferable to that based on still images, since the simultaneous use of temporal and spatial information helps in recognition.

In our project, we focused on video-based recognition, since it is more in demand and more relevant[10].

2.2 Facial Recognition Steps

A facial recognition system consists essentially of the following steps:

- Face detection;
- Face Processing;
- Extraction of characteristics;
- Recognition or verification.

These steps are explained through the following Pipeline

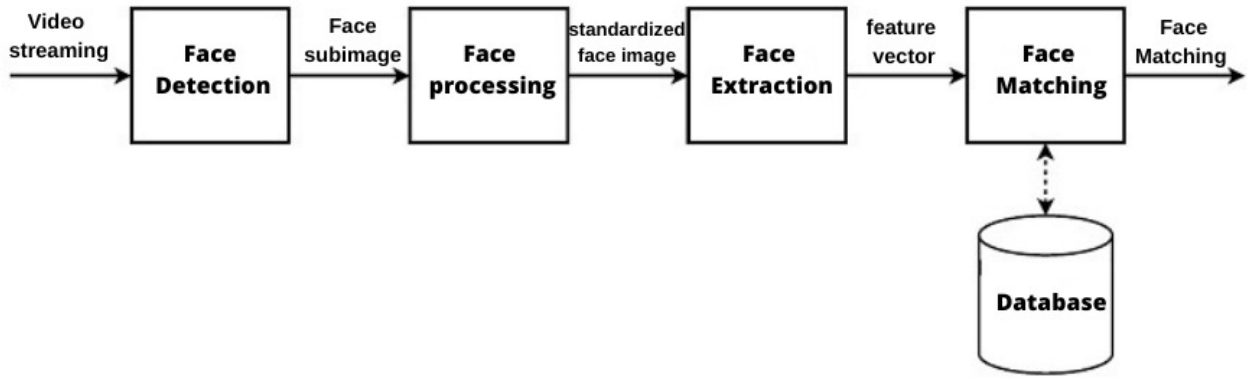


Figure 2.1: Facial Recognition Steps Pipeline

2.2.1 Face Detection

Face detection consists of estimating the face boundary box in an image given. The goal is to extract the face so that it can be better used in recognition.

To make the face recognition system more robust and easy to design, some systems include a face alignment.

The detection of faces must face several difficulties to know :

- The variation of the pose: The ideal condition for the detection of faces is that the image is frontal. Except that this is very unlikely in uncontrolled general conditions.
- Occlusion of elements: The presence of elements such as beards, glasses, or hats introduces a high variability. Faces can also be partially covered by objects or other faces.
- Facial expressions: Facial features also vary greatly due to different facial gestures.
- Imaging conditions: Different cameras and dark conditions, such as lighting conditions and camera type affect the quality of an image, thus affecting the appearance of a face.

If there are multiple faces in an image, they must all be detected.The stage of detection is very essential in the process of facial recognition.

Several detection methods could be used, in the table below we present three of these methods.

Table 2.1: Face detectors.

Detector	Description
CNN	Davis King, the creator of DLIB, formed a CNN face detector based on his work on maximum margin object detection [11]. The method is a very precise.
DNN	It comes with a network of convolutive neurons (CNN) pre-trained for face detection. This new model improves face detection performance compared to traditional models, such as Haar Cascade.
MTCNN	It is a modern face detection tool, using a 3-stage neural network detector. MTCNN is very precise and robust. It correctly detects faces even with large sizes, lighting and rotations. But it is slower compared to some detectors like DNN .

2.2.2 Image Processing

After the face detection step, the detected face regions are extracted from the input image using the bounding boxes or facial landmarks. These face regions are then pre-processed before being passed to the face recognition model for further analysis. The preprocessing steps include:

- Resizing the face regions to a fixed size required by the face recognition model, such as 160x160 pixels.

- Converting the color format of the face regions from BGR (Blue-Green-Red) to RGB (Red-Green-Blue) format, as FaceNet typically expects RGB images.

2.2.3 Extraction of characteristics

After the detection stage comes the extraction of the characteristics which is the basic and most important initialization stage for facial recognition consisting of the extraction of the components of the face.

There are different methods for extracting the various combinations of characteristics which theoretically cannot be identical for two persons except identical twins.

We can divide extraction methods into Shallow and deep learning approach .

- **The Deep Learning Approach**

It is a Machine Learning class that uses multiple layers for the progressive extraction of the highest level characteristics from a raw input. In fact, the word "deep" in deep learning refers to the number of layers hidden from a network of neurons through which data is transformed.
Deep learning models are able to extract better features than shallow models, which means that the more layers there are, the more they help to learn the features effectively.

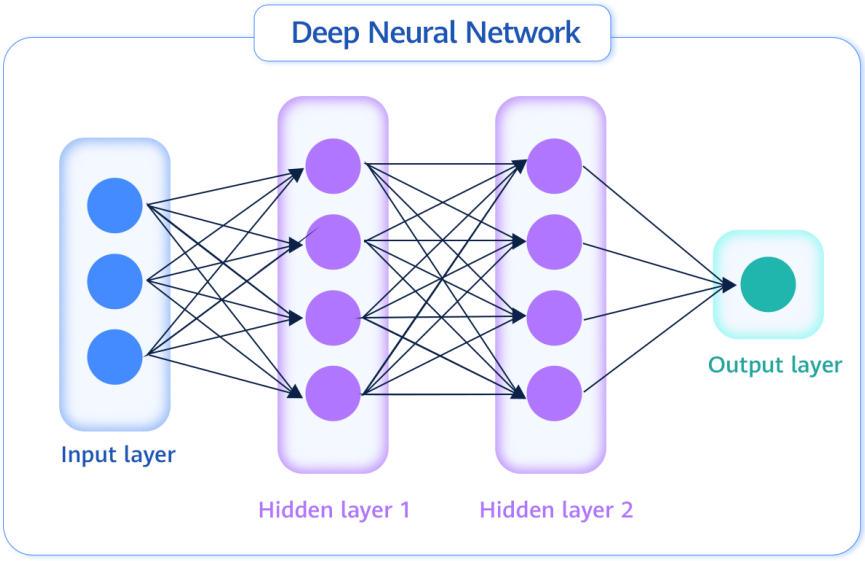


Figure 2.2: Deep Neural Network Architecture

In the 1990s, Yann Le Cun, considered one of the inventors of deep learning developed the technique of convolutive networks for image recognition[12].

- **Convolutional neural networks**

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data[13].
The architecture of CNN is illustrated in Figure (2.3), comprises two components:

- Feature extraction
- Classification

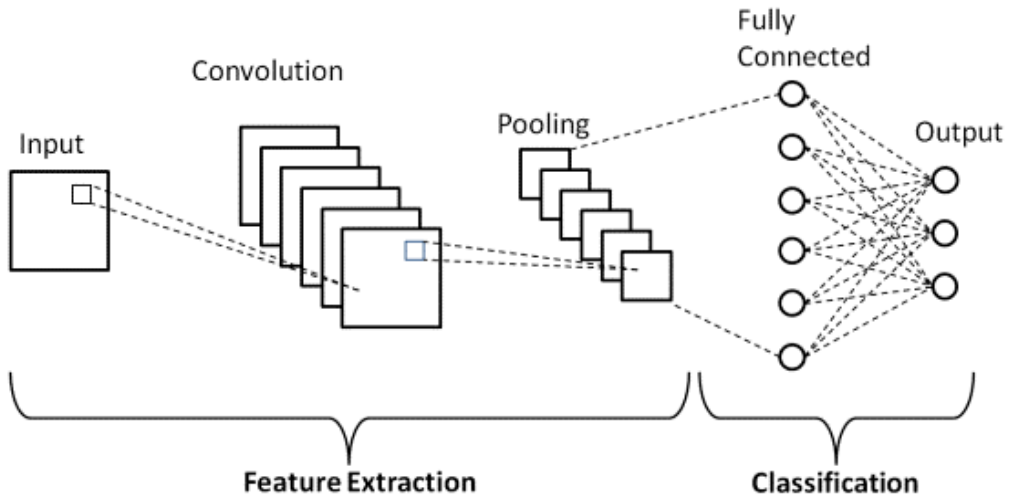


Figure 2.3: Architecture of a Convolutional Neural Network .

- Feature extraction: refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set[14].
  In CNN, convolution layers extracts important patterns from data, while pooling layers reduces its size, capturing key information. Together, they enable the network to learn and recognize meaningful features for tasks like image classification[15].
- Classification: is the task of assigning a label or class to an input image based on the extracted features.

2.2.4 CNN pre-trained model

FaceNet is a cutting-edge neural network for face recognition, verification and grouping. It is a 22-layer deep neural network, published by Google researchers Schroff et al[16].

This model is trained on an extensive dataset and is quite good and widely used in face recognition.

FaceNet's idea is to generate 128 measurements for each face. The training process works by examining 3 images at a time . Two pictures of a known person and the third for a totally different person.

These measurements are processed by the triplet loss function that reassures vectors for the same identity to be a smaller distance. On the other hand, vectors for different identities are on the more considerable distance.

The following figure (2.4) describes the structure of FaceNet model.

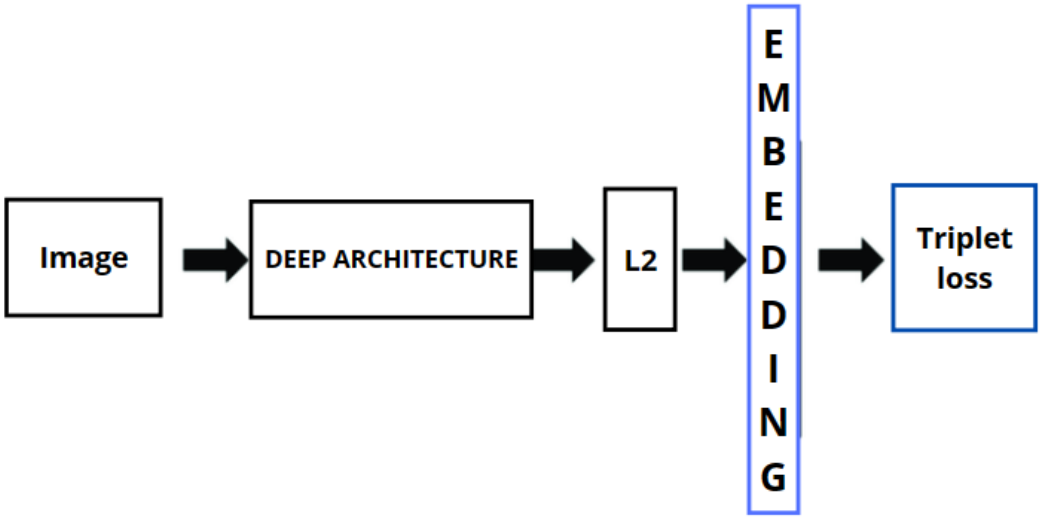


Figure 2.4: FaceNet Model structure

FaceNet takes a picture of an incoming face and takes out a vector of 128 numbers which represent the most important characteristics of a face. This vector is called embeddings contains all the important information in an image is embedded in that vector as shown in the figure (2.5).

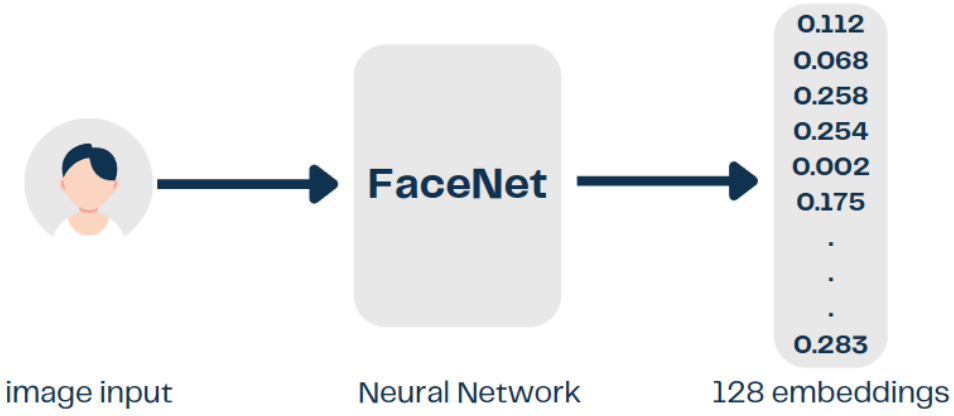


Figure 2.5: Measurements generated by FaceNet for a face

The algorithm then examines the measurements it generates for each of these three images. It then slightly modifies the neural network to ensure that the measurements it generates for images 1 and 2 are slightly closer and that the measurements for images 2 and 3 are slightly further apart using the Triplet Loss method.

The Triplet Loss minimizes the margin between an anchor and a positive, both of which have the same identity, and maximizes the margin between the anchor and a negative of a different identity.
The figure (2.6) explains the way Triplet Loss functions.

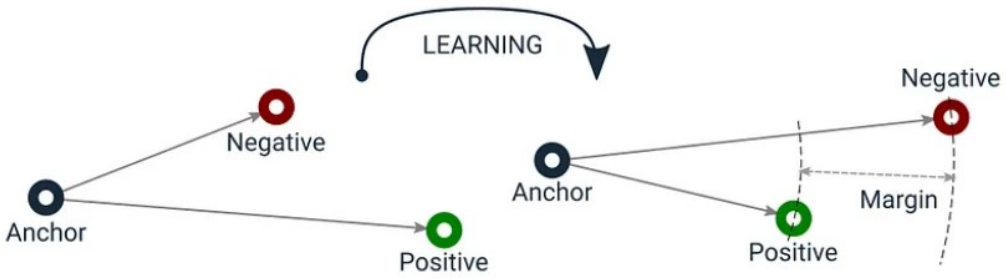


Figure 2.6: The Triplet Loss[17]

After repeating this step millions of times, the neural network learns to reliably generate 128 measurements for each person. Ten different photos of the same person should give about the same measurements.

2.3 Recognition or verification

After choosing the architecture, the extraction of deep features can be performed for test data , face identification and verification operations can be performed. The matching of faces can be done using the Euclidean distance or the Similarity in cosine.

2.3.1 Euclidean distance

The Euclidean distance can be used to calculate the distance between any two points in two-dimensional space, and also to measure the absolute distance between points in N-dimensional space. For face recognition, smaller values indicate more similar faces. The image to be detected is processed by a function to obtain a 128-dimensional face feature vector, this constitutes the condition for face similarity calculation under Euclidean distance.[18]

Suppose the face feature of the image to be detected A(x1..x2.....x128) training sample face features B(y1..y2..y128)the Euclidean distance calculation formula is as follows:

$$AB = \sqrt{(x1 - y1)^2 + (x2 - y2)^2 \quad +....+....(x128 - y128)^2}$$

2.3.2 Cosine similarity

It calculates similarity by measuring the cosine of angle between two vectors in their respective vector spaces[19]. A is the detected image vector and B is the feature vector.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Conclusion

In this chapter, we explored facial recognition, its processes as well as the artificial intelligence role. The following chapter is about the requirements analysis and design of our proposed system.

Chapter 3

Requirements Analysis and Design

Introduction

After placing the project in its theoretical framework, we analyze in this chapter the design of our project. The first part of this chapter will focus on functional and non-functional requirements. The second part will be dedicated to the presentation of the system through dedicated diagrams.

3.1 Requirements analysis

This section presents the functional and non-functional requirements identified for understanding the system's interest.

3.1.1 Identifying actors

An actor is a person, software, or hardware that interacts with the system. The actors identified for our platform are:

Administrator: This is the person responsible for managing the admin interface, and the addition or deletion of users.

User: the one who uses the mobile application to identify suspected people under the administrator's control.

3.1.2 Functional requirements

Functional requirements are the tasks that the application must perform to meet the client's requirements and align with the user's needs. Therefore, our facial recognition system must meet the following functional requirements:

- Authentication: This use case is critical for the application as most system tasks require an authentication procedure. This means that every actor needs to authenticate with the system to access their space and perform essential application operations.
- Notification: the agent should be able to check the received alert notifications and have access to the current police stations according to his location.
- Face detection: Recording data which is the detected faces to retrieve them later.
- Face recognition: recognize the detected faces.
- Speed: The response and processing time of the application must be minimal to facilitate the reservation process for the increasing number of customers.

3.1.3 Non-functional requirements

It is the constraints that evaluate a system's service and define the level of customer satisfaction. The main non-functional needs of our solution can be summarized in the following points:

- Usability: The application's interfaces must be ergonomic, simple, and clear. Their manipulation should be easy to adapt well to users.
- Efficiency: Our application's efficiency should allow tasks to be accomplished with a minimum of manipulation. This must be guaranteed for our product to integrate into the market easily.
- Accessibility: The application must be accessible to multiple users at the same time.
- Reliability: The application must ensure proper functioning without errors.
- Performance: Optimization of import and export times for data and processing of results: improve the speed and efficiency of importing and exporting data, as well as processing the results to make these operations faster and more effective.
- Scalability: It must be possible to extend the system.

3.2 Prototype Interfaces

A prototype is an original design with all the technical qualities and characteristics of a new product. In the field of application development, a technique has appeared interesting: it is prototyping.

This technique consists in preparing some graphical interfaces of the application by using a tool of a conception of prototypes to validate the choice of the customer that will allow us to improve and optimize before the realization to avoid being blocked in choices that would be difficult to modify once the application is created.

We used the Figma tool to design the required interfaces . (Annexe B)

3.3 Design

After analyzing the functional and non-functional needs of the project and defining the stakeholders who will interact with the application, we develop the various diagrams necessary for the system's design.

3.3.1 Modeling language

Unified Modeling Language (UML): is a graphical modeling language based on pictograms that is designed as a standardized method of visualization in software development and object-oriented design.For creating our diagrams, we used PlantUML[20].

3.3.2 Use case diagram

After defining the actors of our project, we will present the use case diagram. This diagram describes the interaction between the actors and the system.

Figure (3.1) shows the use case diagram of our project. It describes the behavior of the general system.

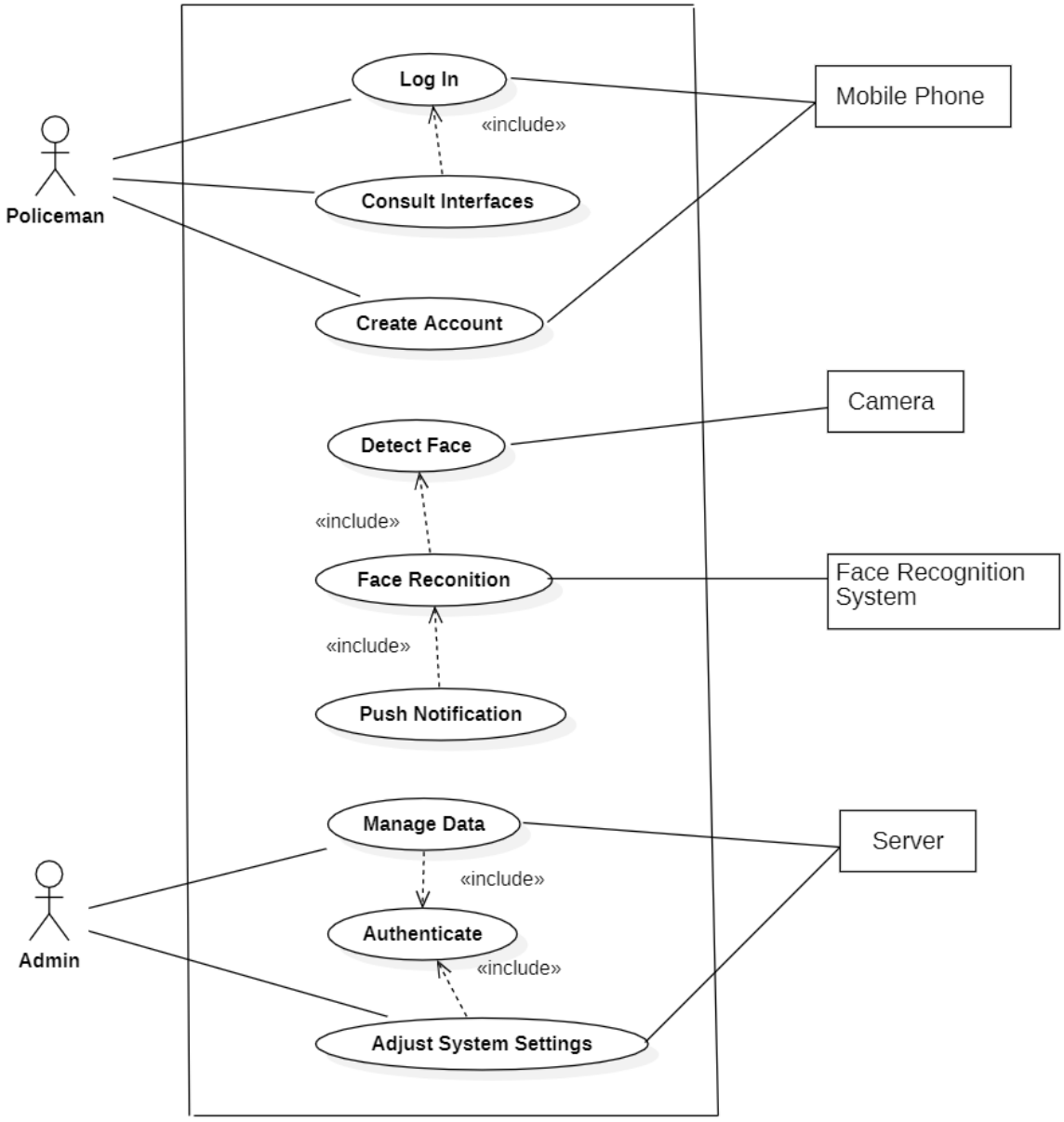


Figure 3.1: Global Use Case Diagram

The table below illustrates the global use case diagram description.

Table 3.1: Global use case description

Use Case	Details
Create an account	This use case allows users to create an account.
Log in	This use case allows users to log in.
Consult interfaces	This use case allows users to access the mobile application's various functionalities and interfaces. .
Detect face image	This use case allows to detect faces in real-time through a body camera .
Face recognition	This use case allows to: • Compare the detected face to the database images. • Recognize the required image of the suspected person.
Push Notification	This use case allows to: • Push notification if the similarity between the detected face and the database images is confirmed (a suspected person is detected). • Receive real-time notifications of identified individuals.
Admin authentication	This use case allows the administrator of the company to log into the system and access its various functionalities.
Manage data	This use case allows to: • Create data. • Update data. • Delete data.
Adjust system settings	This use case allows to: • Manage users. • update settings.

3.3.3 Global use case's refinement

To deepen our study of the overall functionalities of the system we must describe the most important use cases to understand the tasks and requirements.

- **Login**: It is a use case that allows the user, already registered in the database to log in.

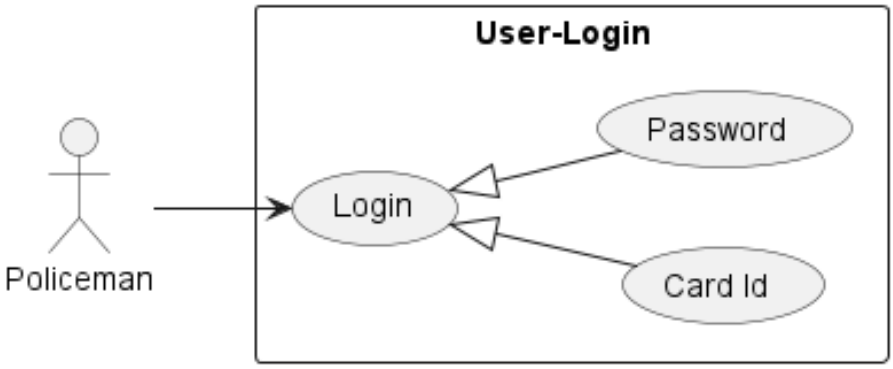


Figure 3.2: Log In Use Case Diagram

The following table 3.2 shows a description of this use case and the steps necessary for authentication.

Table 3.2: Log In Use Case description

Title	Log In management
Actor	The agent that will use the mobile application
Description	The actor identifies himself with his face id or password, and the database system verifies his membership in the list of users.
Pre-conditions	The actor is already on the user list.
Post-conditions	The user is connected to the Internet to ensure connection to the application.
Nominal scenario	• The user connects by analyzing his face id or password. • The database MongoDB checks the data entered by the user. • The system then grants access authorization. • By clicking on the login button, the user will be directed to the home page.
Alternative scenario	• The user enters the login and password. • The system verifies the existence of the user. • The system displays an error message. • The system asks to try authentication again. • The user tries again.

By starting the application, the first scenario that takes place on the user side is LogIn . Indeed, the user must enter his connection parameters. The system ensures the availability of the login and password, to then give access to the functionalities of the application.

If the data entered is invalid or a field is empty, an error occurs and the user will be invited to verify their data and try again.

It is explained by the following sequence diagram:

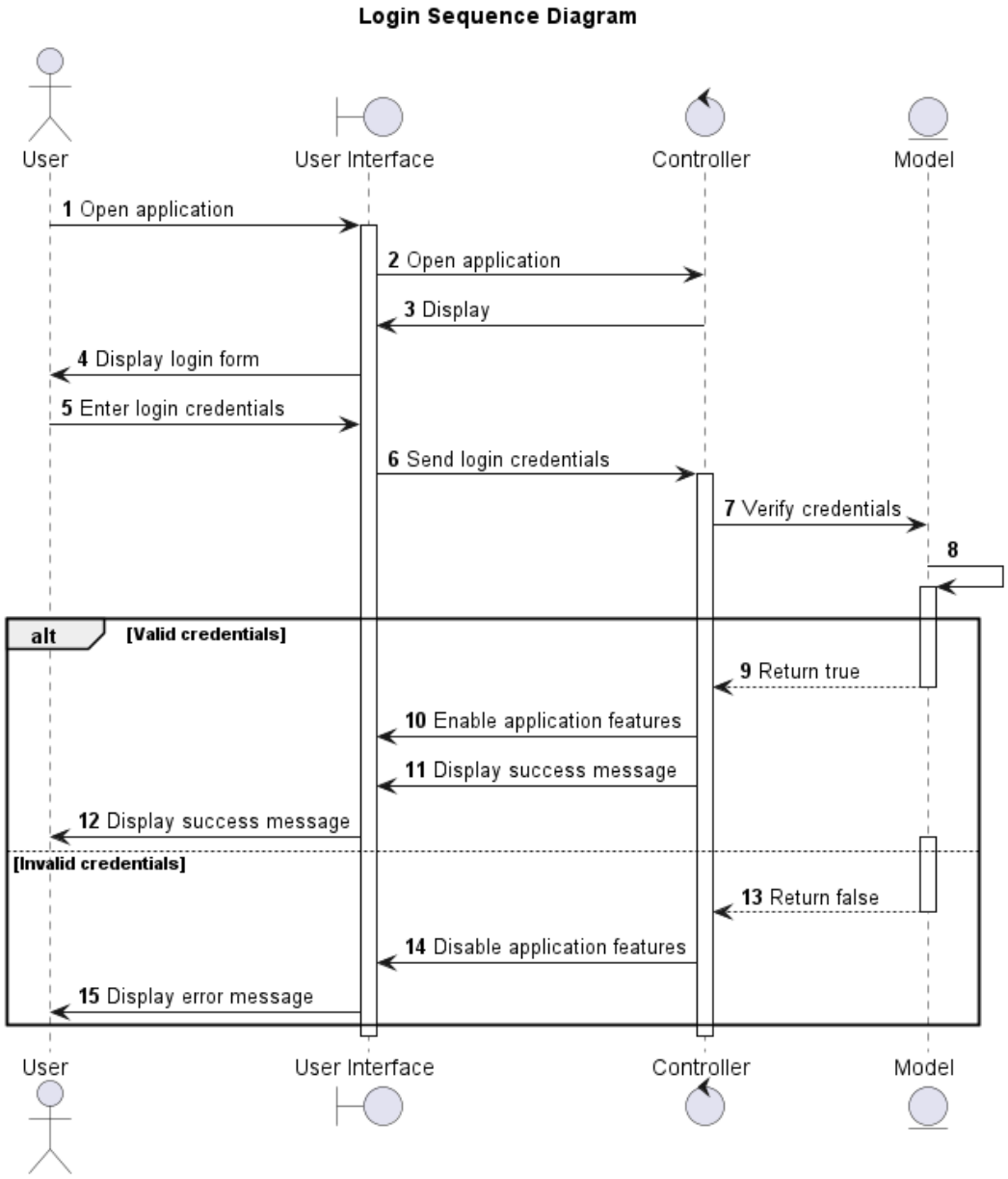


Figure 3.3: Log In Sequence Diagram.

This sequence diagram has four components. User, view, controller and model. the user is the policeman that opens PoliSee application and initiates a sequence. The view is the Login interface that displays a login form and success/failure messages. Controllers are responsible for processing user input and sending requests to models to validate credentials. The model is the data layer that stores and retrieves user credentials.

- **Admin Authentication:** "Authentication" is a use case that allows the admin already registered in the database to log in.

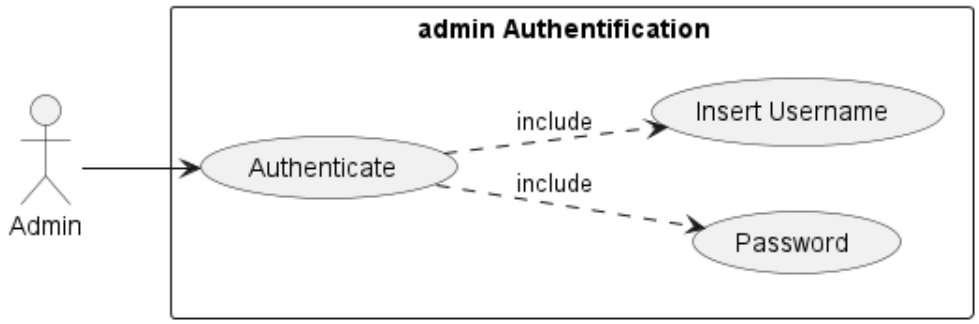


Figure 3.4: Admin Authentication Use Case Diagram

The following table shows a description of this use case. Through this table, the steps necessary for authentication, as well as the constraints to be met, are described.

Table 3.3: Admin Authentication Use Case Description

Title	Authentication management
Actor	The admin that will use the web management interface.
Description	The admin identifies himself with his password and Username, and the database system verifies his membership in the list of admins.
Pre-conditions	The admin is already on the admin list.
Post-conditions	The admin is connected to the Internet to ensure connection to the application.
Nominal scenario	• The admin connects by analyzing his password and Username. • The database MongoDB checks the data entered by the admin. • The system then grants access authorization. • By clicking on the login button, the admin will be directed to the system management interface and manage data or adjust system settings.
Alternative scenario	• The admin enters the Username and password. • The system verifies the existence of the admin. • The system displays an error message. • The system asks to try authentication again. • The admin tries again.

The scenario that takes place on the admin side is Authentication. Indeed, the admin must enter his connection parameters. The system ensures the availability of the login and password, to then give access to the functionalities of the system management interface.

If the data entered is invalid or a field is empty, an error occurs and the admin will be invited to verify their data and try again. It is explained by the following sequence diagram in figure (3.5):

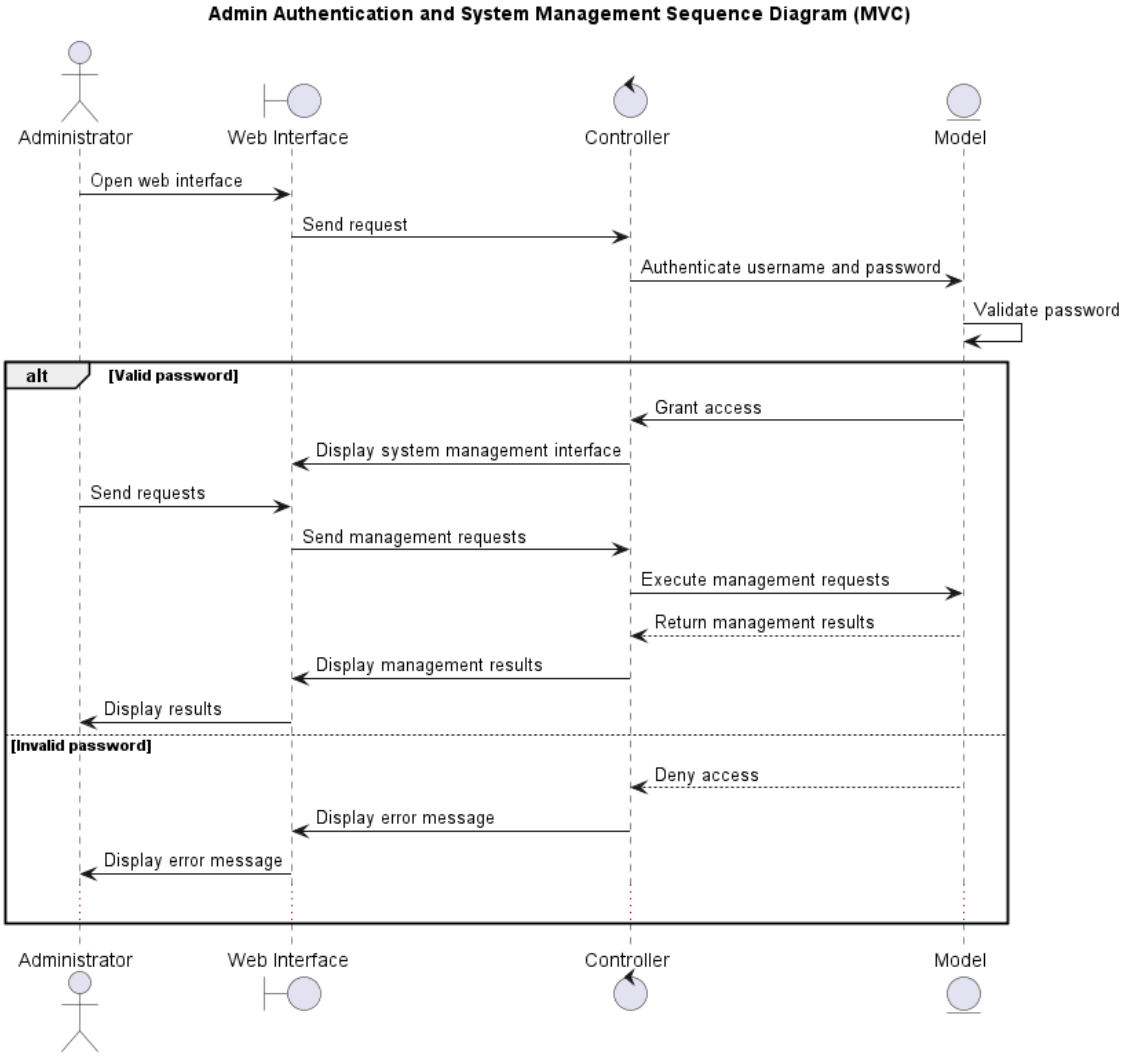


Figure 3.5: Admin Authentication Sequence Diagram

In this sequence diagram, the View component has been replaced with a Web Interface Border component. The administrator continues to open the web interface to initiate use cases. The web interface sends requests to the controller, which communicates with the model to authenticate the username and password.

- **Face Recognition:** This use case describes the process of detecting a face using the camera and displaying the face recognition result on the mobile application.

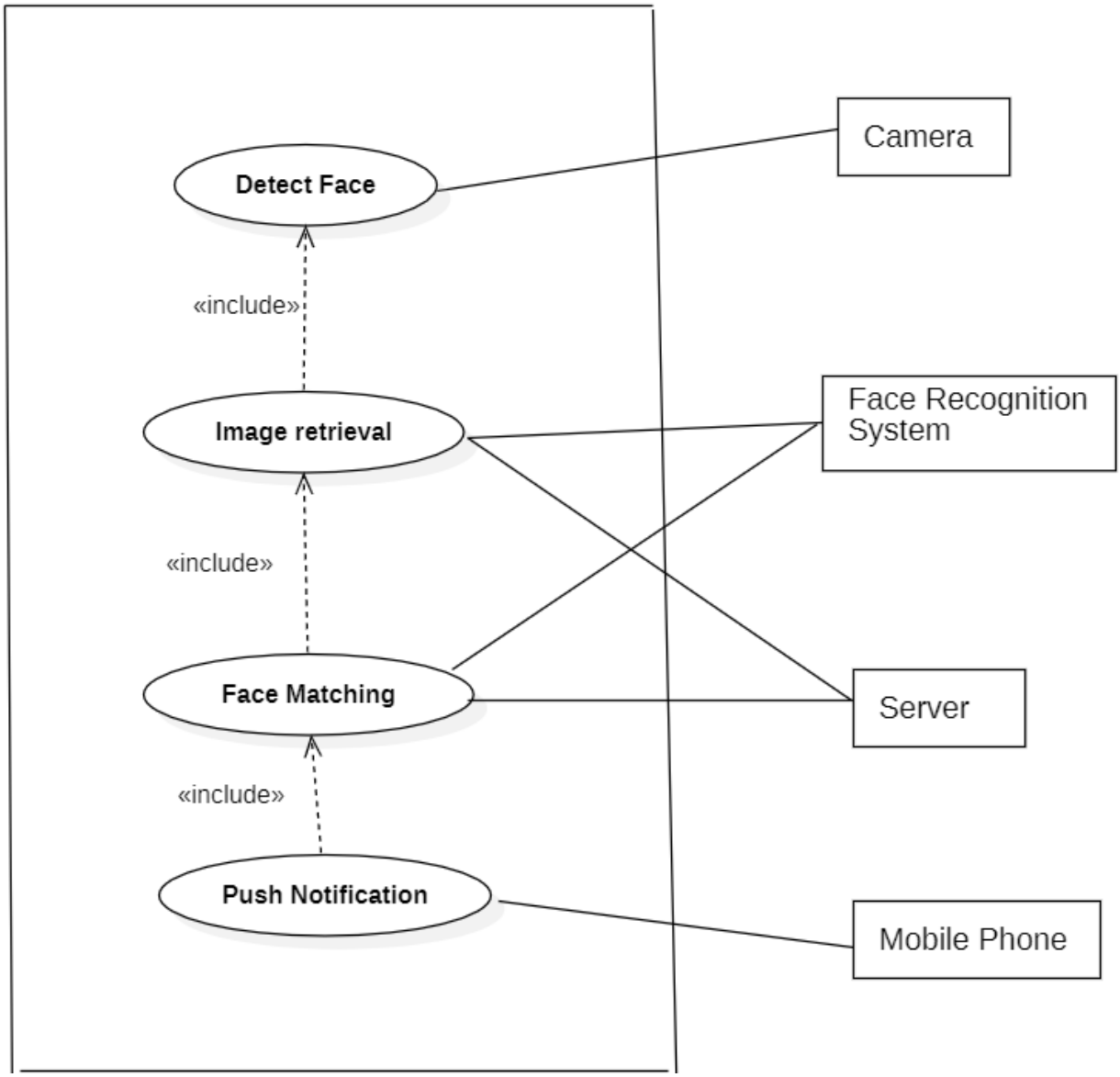


Figure 3.6: Face Recognition Use Case Diagram

The following table shows a description of this use case. Through this table, the steps necessary for Face Recognition, as well as the constraints to be met, are described

Table 3.4: Face Recognition Use Case Description

Title	Face Recognition
Actor	The camera that will detect the person's face. The face recognition system will process the face recognition. The database server contains the wanted persons' pictures. The device mobile phone
Description	The camera detects a face and the process of face recognition starts.
Pre-conditions	The camera is already working on.
Post-conditions	The user is connected to the Internet to ensure connection to the application
Nominal scenario	• The camera starts the video streaming and captures face. • The capture is processed to detect a face. • If the detected face is recognized the result will be displayed on the mobile application through a notification. • By clicking on the notification the picture of the detected person and its ID will be displayed.
Alternative scenario	• The camera is working. • The video streaming starts. • The frames are captured. • If the face detected is not recognized an alert displays on the mobile application indicating that the person is unwanted.

The scenario that takes place on the face recognition system side is face detection. Indeed, the user must be online. The system ensures the matching between the detected face and the database pictures, to then display the result of the detected person wanted. If the detected face is unknown then an alert will be displayed indicating that this person is unwanted.
It is explained by the following sequence diagram in figure (3.7)

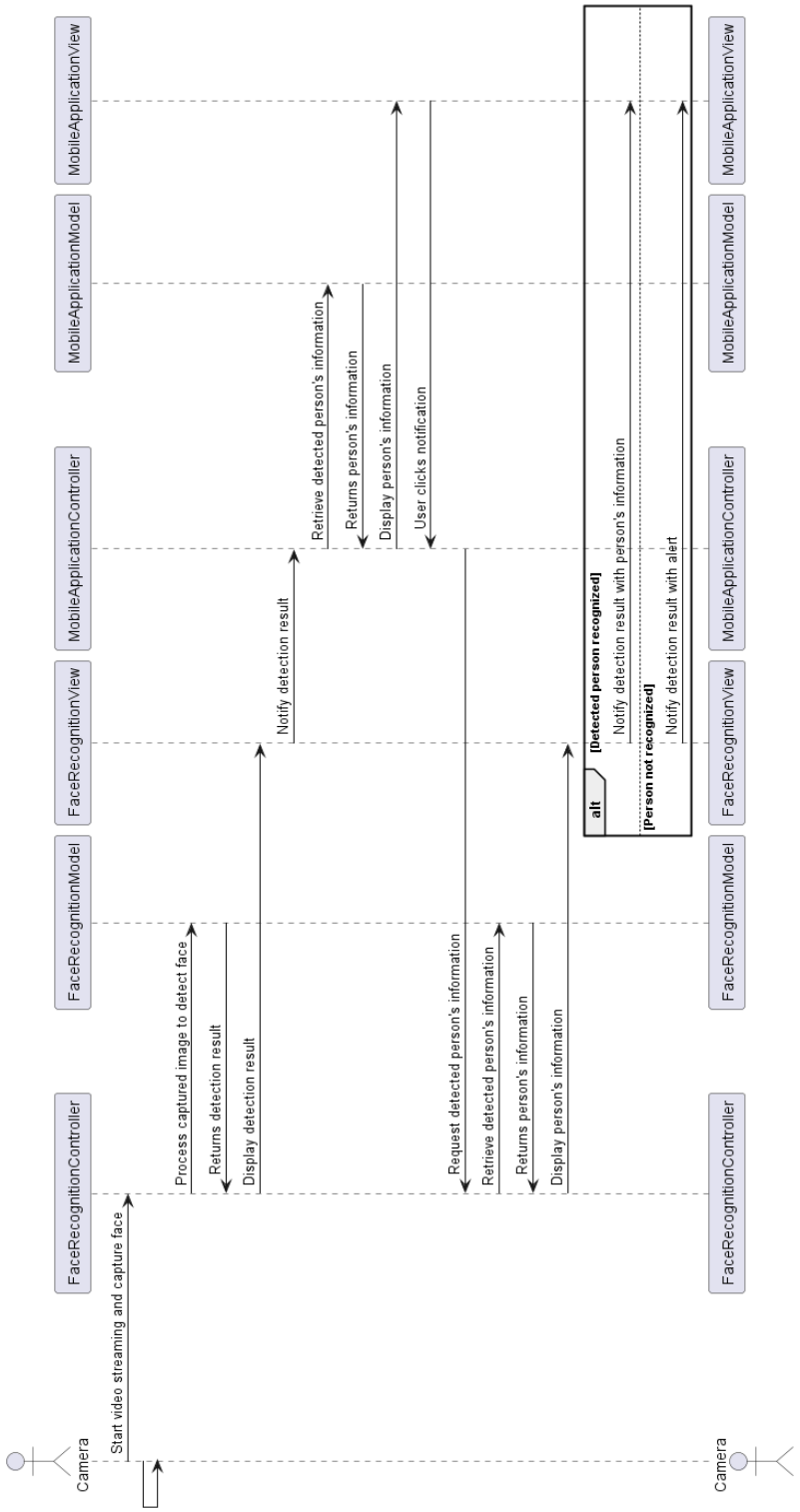


Figure 3.7: Face Recognition Sequence Diagram.

In this diagram, FaceRecognitionController communicates with FaceRecognitionModel to process captured images and recognize faces. Recognition results are displayed in FaceRecognitionView. MobileApplicationController is notified of recognition results and receives information about the recognized person from MobileApplicationModel. The information is then displayed by her MobileApplicationView. If a person is recognized, the notification will contain that person's information.

Finally, when the user clicks on the notification, the MobileApplicationController requests the recognized person's information from her FaceRecognitionController and displays it in the MobileApplicationView.

3.3.4 Class diagram

The class diagram is regarded as the most important diagram in object-oriented modeling, it is the only mandatory diagram in such modeling. It allows to define the internal structure of the system, unlike the use case diagram which shows the system from the actor's point of view.
The figure (3.8) represents the static view of our application shown by the following class schematic:

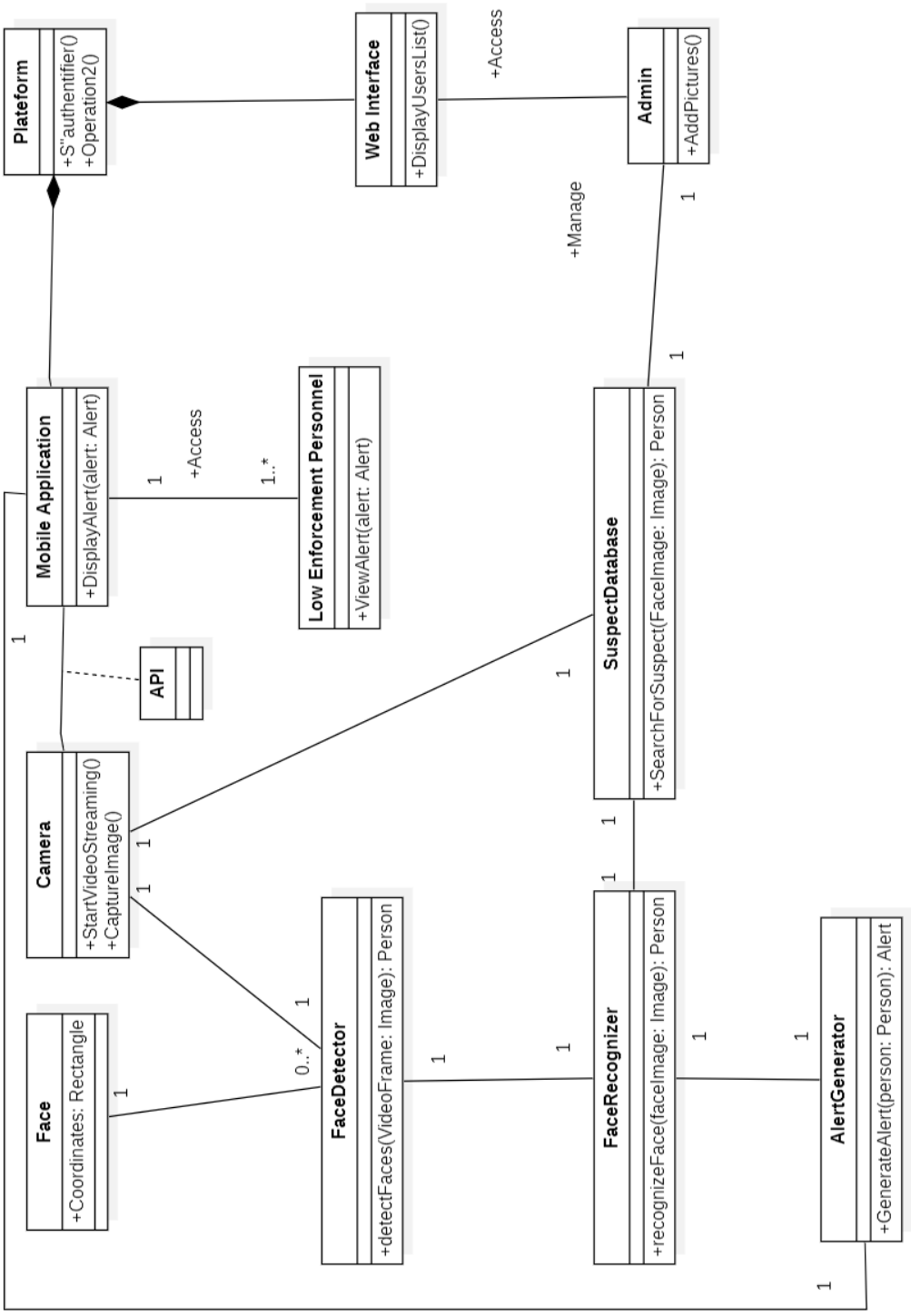


Figure 3.8: Class Diagram.

3.3.5 Activity diagram

Activity diagrams allow us to focus on the processes. They are therefore especially suitable for modeling control flows and data flows.

They provide a graphical representation of the behavior of a method or the progression of a use case. To that effect, the activity chart was created to better understand the system briefly.

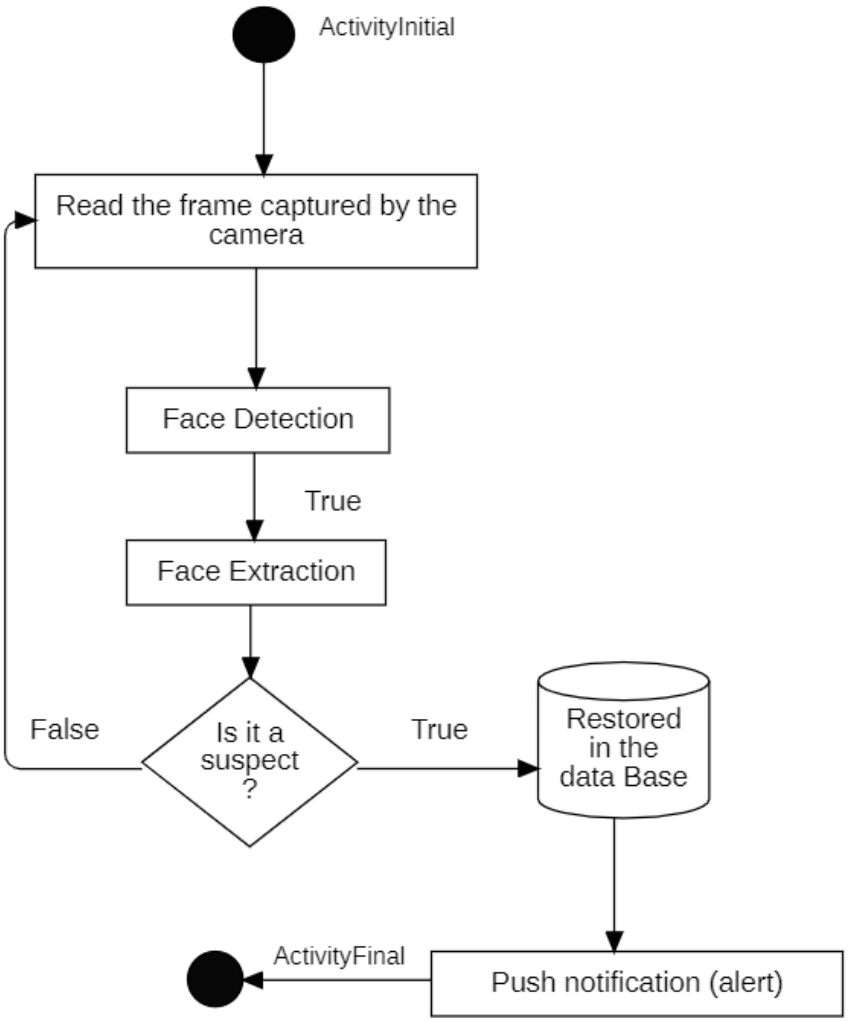


Figure 3.9: Activity Diagram.

The camera detects the existence of a face and captures the frame from the video streaming. The face recognition system reads the captured frame and detects the face.

There are two scenarios:

- If that face is recognized as a suspected person then a notification will be displayed on the mobile application by clicking on this alert the person's picture will be displayed with its ID after being restored in the database.

- Else if the detected face is not labeled as a suspected person the mobile application will not receive anything.

Conclusion

This section enabled us to detail the project specifications, identify the use cases and develop the corresponding diagrams. At this stage, we are ready to move on to developing the artificial intelligence face recognition code , mobile and web applications in the next chapter.

Chapter 4

The Proposed Face Recognition System

Introduction

This chapter presents firstly the tools we used during our internship. The second part includes our comparative study of different face detectors and characteristic extraction models.

It also includes our proposed approach to facial recognition system implementation. Our final section discusses our association in developing our mobile application.

4.1 Work Environment

This section presents an overview of the hardware and software environments.

4.1.1 Hardware Environment

The development of the application is carried out through two laptops having the following characteristics:

Table 4.1: Characteristics of the first machine

Manufacturer	Lenovo Thinkpad
processor	Ryzen 7
Installed Memory (RAM)	16Go
Operating System	Windows 11

Table 4.2: Characteristics of the second machine

Manufacturer	Acer aspire3
processor	Intel Core i5
Installed Memory (RAM)	8Go
Operating System	Windows 10

4.1.2 Software Environment

In order to build a face recognition mobile application , we will have to choose between several technologies. It depends on several factors, namely the type of project, its needs and specifications.

The choice of the technology to be used is very important because it impacts the duration of realization, the safety and the possible functionalities of the project.

Throughout the development phase, we used the following software environment.(Annexe A)

Table 4.3: Software Environment

	Tools
Face Recognition	• Python • OpenCV • Numpy • Dlib • Face Recognition Package
Application Development	• Flutter • Dart • Angular • Node.Js
Data Base	• MongoDB

4.2 Face Recognition

This section describes the key steps in face recognition.

4.2.1 Face Detection

To choose the suitable face detector we made a comparative study between the most used face detectors DNN, CNN and MTCNN. The comparison was made with a database of 3600 images that we built using images of different types (frontal and non frontal images, images with glasses, etc). The following table compares the face detectors for the criteria listed below.

- **Detection accuracy and Frame per second measurement**

Table 4.4: Comparison between detectors

Model	FPS	FPS Average	Accuracy
DNN	466.50	2.88	97.06%
CNN	117.31	0.62	90.74%
MTCNN	324.46	2.83	95.65%

The DNN detection method had the best results in the Accuracy/FPS ratio, since under the different conditions it had the best accuracy, and an acceptable FPS overall.(Annexe C)

For each image, the dnn model for face detection performs the following steps:

• Processes the input image and detects the regions of the image that potentially contain faces.

• If a face is detected,The model outputs bounding boxes that indicate the location of detected faces in the image.

4.2.2 Characteristics Extraction

For our system, we used facenet model .It is an algorithm based on a deep convolutional neural network (CNN), which can be used for face recognition, verification, and clustering. FaceNet processes face regions and generate high-dimensional feature vectors that represent each face's unique characteristics. It starts by mapping face images into a Euclidean space so that the distance between images corresponds to similarity .

A major advantage is that the model is extremely lightweight, representing each face using only 128 bytes of data.

For each database image, the following steps are applied:

• The FaceNet model is used to obtain the face encoding of the face .

• The face encoding and the corresponding name are added to the suspect-face-encoding and suspect-face-names lists, respectively.

For each detected face the same steps are applied but the detected face encoding is saved in the Detected-face-encoding list.

4.2.3 Face Matching

The detected face encoding is compared with the suspect face encodings using the Euclidean distance metric. The distances are calculated between the encoding of the detected face and all the suspect face encodings.

The minimum distance index is determined to find the closest match. If the minimum distance is below a threshold (0.6 in this case), the recognized face is assigned the corresponding name from the suspect-face-names list. Otherwise, it is labeled as "Unknown".

After generating our face recognition code we have tried to test its performance .

In the first test, the face recognition system detected two faces. The first face was detected as ”suspect” as it is labled ”suspect” in the database but the second face was not defined in the database so the recognition result is ”unknown”.

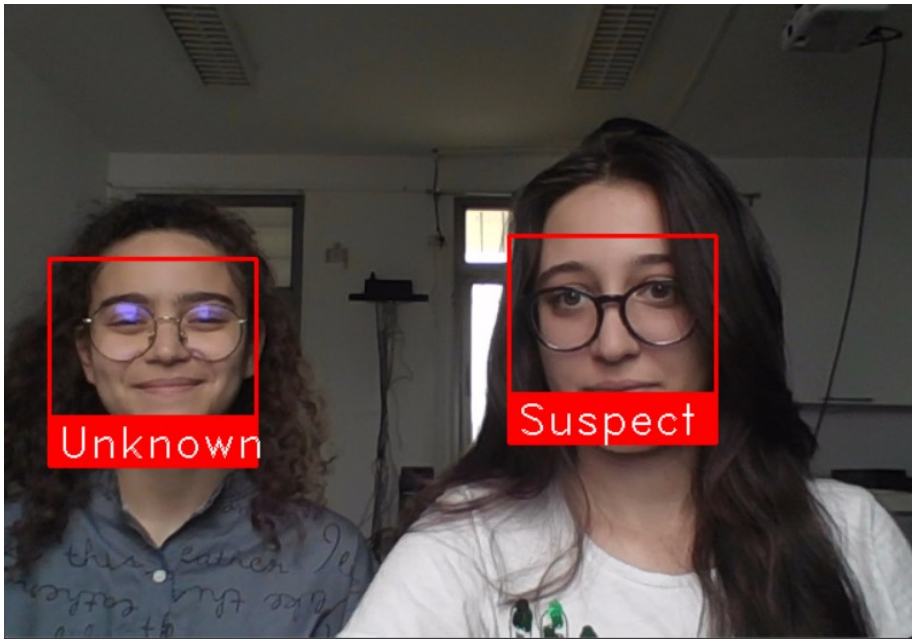


Figure 4.1: Face Recognition frontal Test

The second test is non-frontal to ensure that our proposed face recognition system recognizes faces from a different side as shown in the figure (4.2).

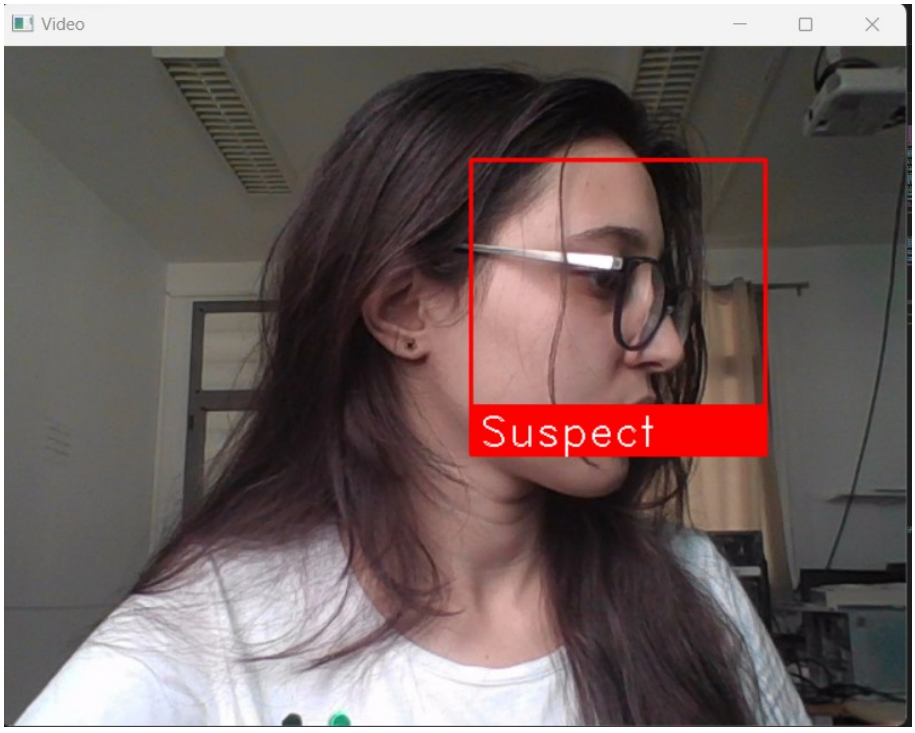


Figure 4.2: Face Recognition non frontal Test

The last test is frontal with a mask on to ensure that our proposed face recognition system is able to detect and recognize masked faces as shown in the figure (4.3).

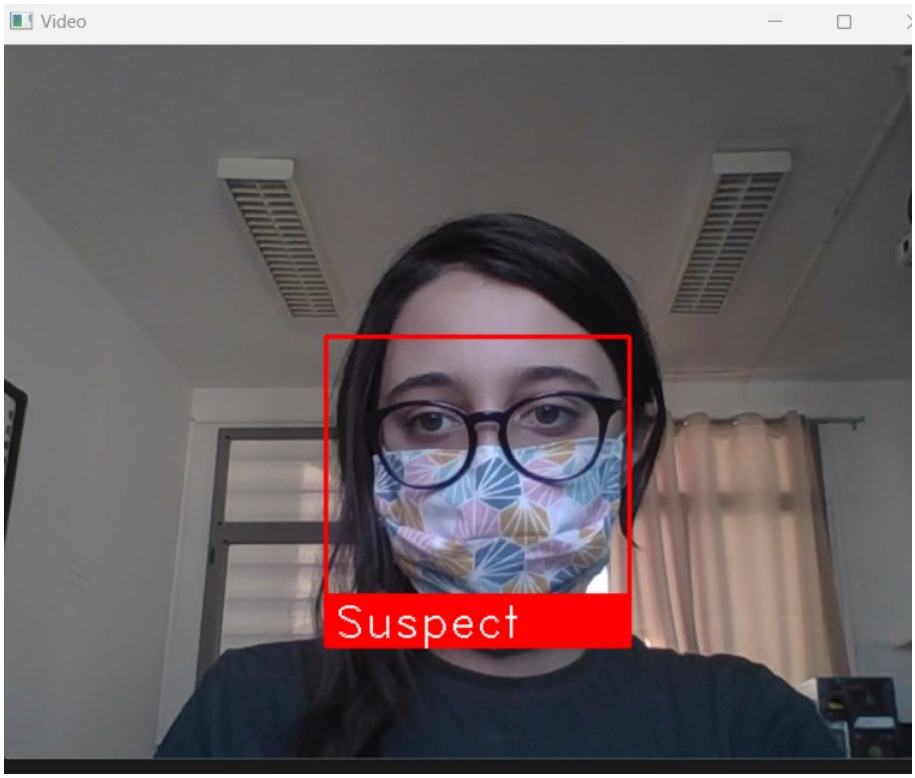


Figure 4.3: Frontal test with a mask on

4.3 Proposed Approach

Based on the above, we selected the best performing methods for the system we propose:

- Face detection using the openCV DNN face detector.
- Extraction of characteristics using the FaceNet.
- Compare faces using the Euclidean distance similarity.

We can summarize our facial recognition system on the following diagram.

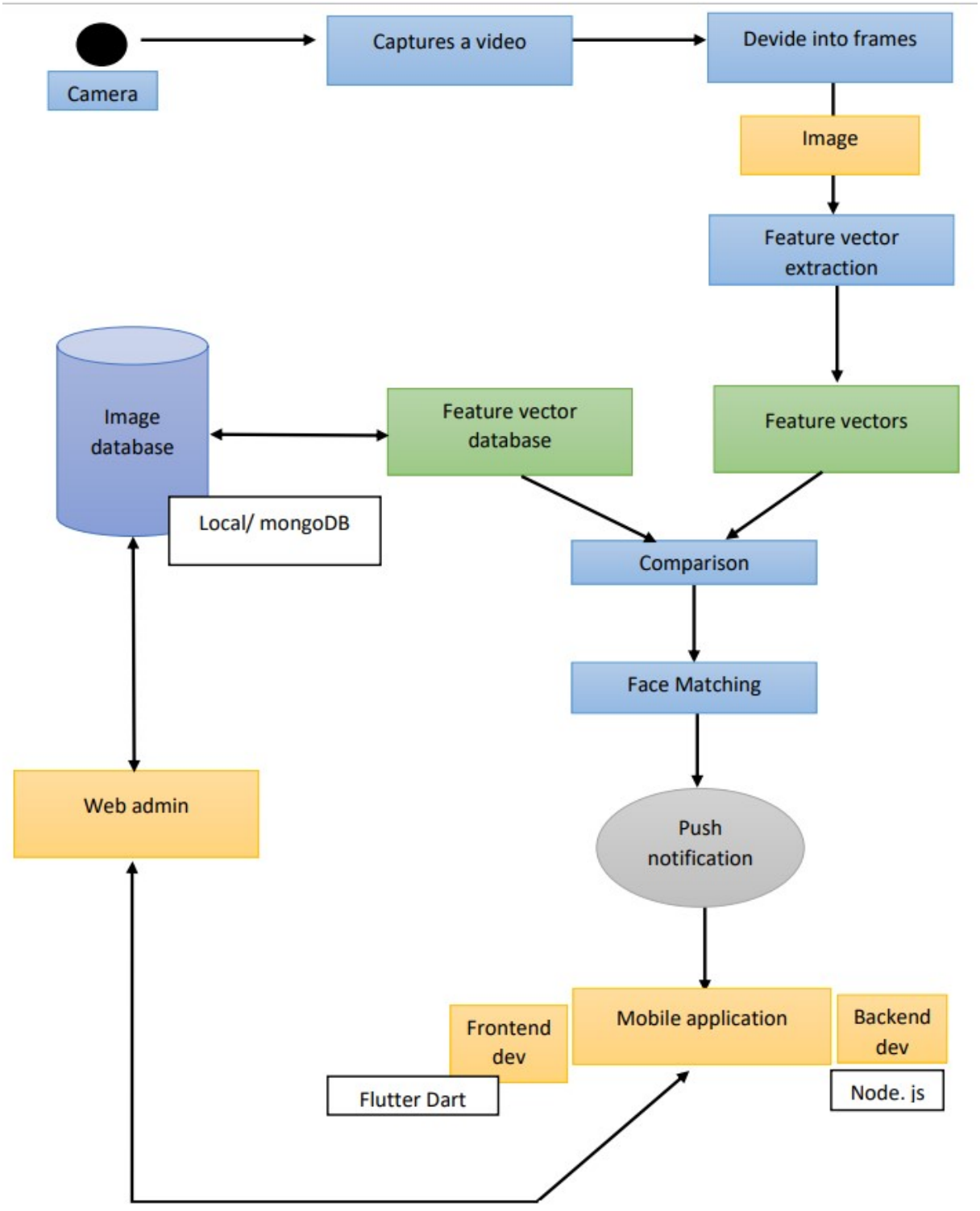


Figure 4.4: The proposed face recognition system architecture

The system checks the existence of a suspected person captured through the cam by comparing the detected face to the data entered in the database. the Policeman automatically receive an alert displayed on the mobile application.

4.4 The mobile and web applications

After presenting the proposed system , we will move on to the implementation of the frontend part using the MVC (Model, View and Controller) architecture.

4.4.1 Logical Architecture

This architecture demonstrates a highly powerful concept that is used in the development of applications. Its primary goal is to separate data (model), display (view) and actions (controller). This architecture offers better performance, as well as easy maintenance. The components of MVC are:[21]

- M (Model): The model is responsible for maintaining the state of the application.
- V (View): The view is responsible for generating a user interface, normally based on model data.
- C (Controller): Controllers receive events from the outside world (normally user input), interact with the model, and display an appropriate view for the user.

The following figure shows the design pattern of the MVC architecture:

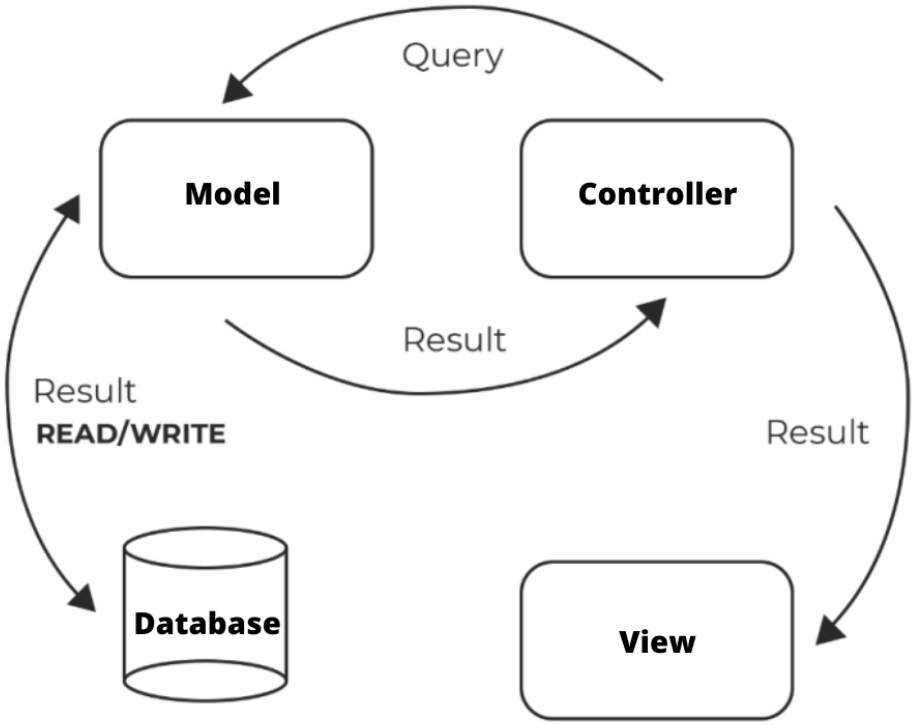


Figure 4.5: The MVC architecture

4.4.2 Frontend implementation

- **Mobile application design**

  As a result of studying and criticizing the existing system, we could then propose an effective solution. Indeed, our solution aims to facilitate mobile application access and manipulation.

  The user accesses the application and consults the various interfaces including profile, Nearby, home and settings (menu side). On the main interface which is the home page, the policeman can check alerts if a suspected person is detected.

  The first step was to design a prototype on Figma explaining how the application would look and function.

  Here are the different interfaces for better understanding:

  Sign-In interface: Users enter their login and password here to access the application.

  Home Page interface: This is the main interface where policemen can consult interfaces and alerts.

  Nearby interface: Policemen are able to Check the nearby police stations according to their current location.

  Profile interface: This is where the user's personal information will be shown.

  Contact interface: In this interface, the user can contact us.

  Language interface: This interface contains multiple language choices for the user.

  About us interface: This interface provides information about our services.

  Privacy Policy interface: This interface explains the type of data we collect and how we use it.

The mobile application is represented by an explanatory map to better explain its interfaces as shown in figure (4.6).

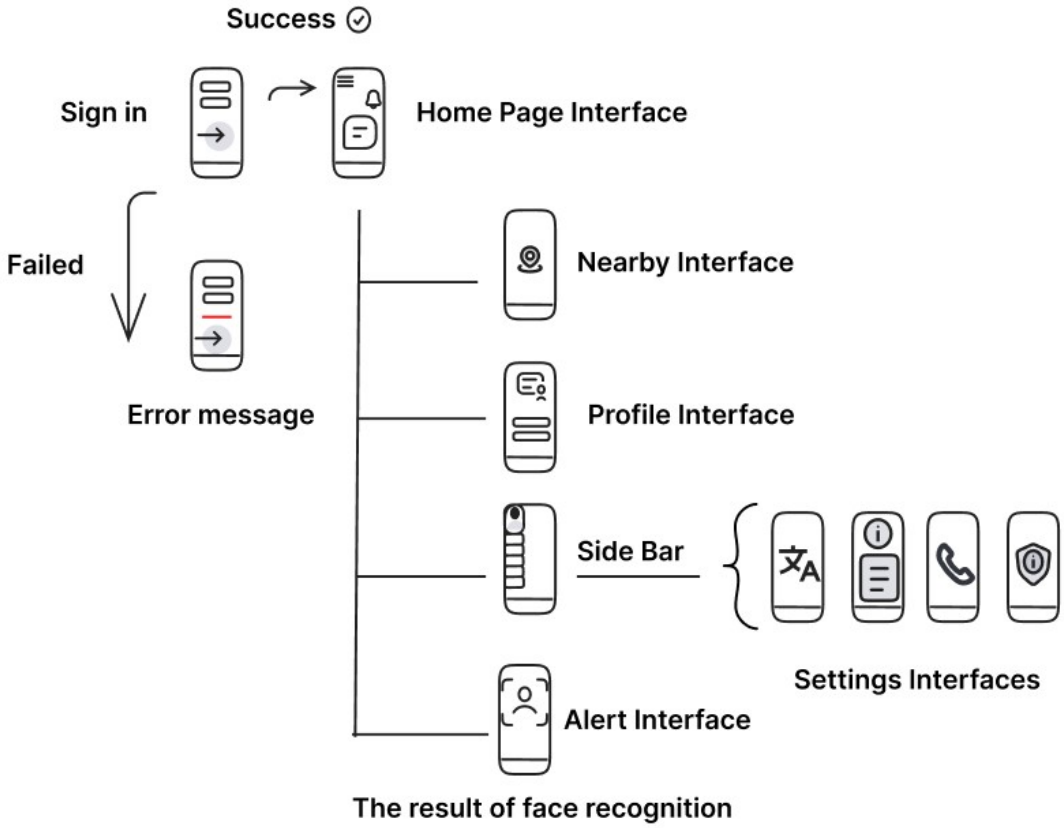


Figure 4.6: Explanatory map for the interfaces

- **Web application design**

  As a result of studying and criticizing the existing system, we could then propose an effective solution. Indeed, our solution aims to facilitate mobile application access and manipulation.

  The Admin accesses the web application and consults the various interfaces including Add User, Table List, Home and Add Data (Suspected persons' Images). The first step was to design a prototype on Figma explaining how the application would look and function.

  Here are the different interfaces for better understanding:

  Sign-In interface: The admin enter its User NName and password here to access the application.

  Update users interface: This is where the admin can manage and update policemen personnel information.

  Table List interface: This interface contains the Table list of policemen using the mobile application .

  Add Data interface: This is where the admin can add Suspected people's images to the data base.

The web application is represented by an explanatory map to better explain its interfaces as shown in figure (4.7).

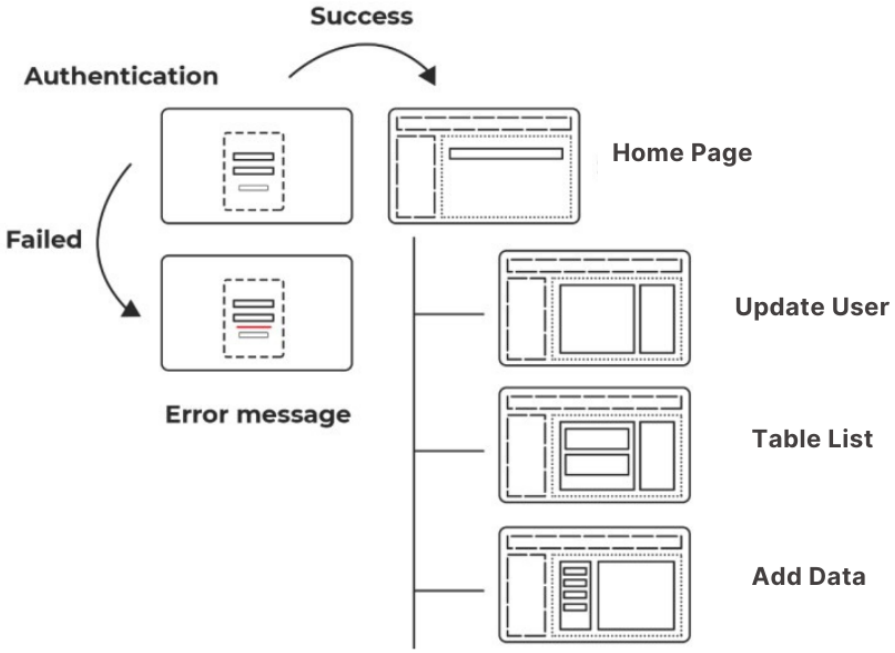


Figure 4.7: Explanatory map for the interfaces

- **Frontend implementation process**

  Our main goal in this part will the description of the implementation process of the frontend section.

  - Mobile application

  After designing a prototype on Figma explaining how the mobile application works, we divided the design into components in our dart code where we worked on each component as a separate part.

  • To access the application, the low enforcement personnel needs an account. To ensure this functionality, we have created a user Sign Up component.

  • The policeman needs to authenticate to access the PoliSee application. To ensure this functionality, we have created a Login component.
  He can access the application by scanning his card service. The login is a succes if the id scanned matches his registered id in the data base. Otherwise , he can choose to login with his password as shown in the figure (4.8).

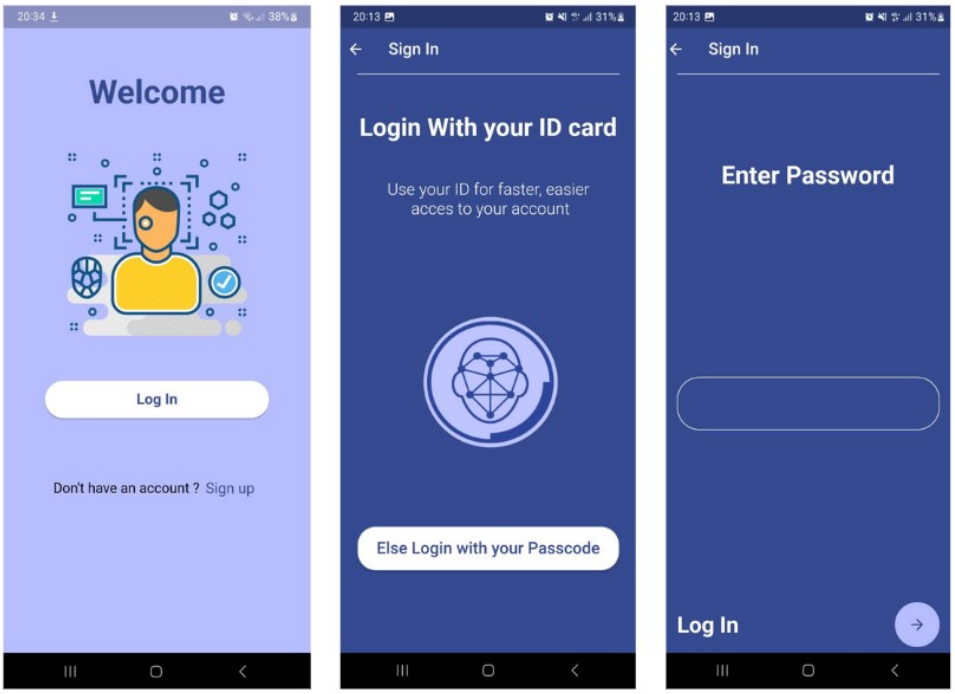


Figure 4.8: Login components

• The policeman, must verify all alerts sent by the face recognition system. To ensure this functionality, we have created a component for displaying alerts with the associated recognized face.

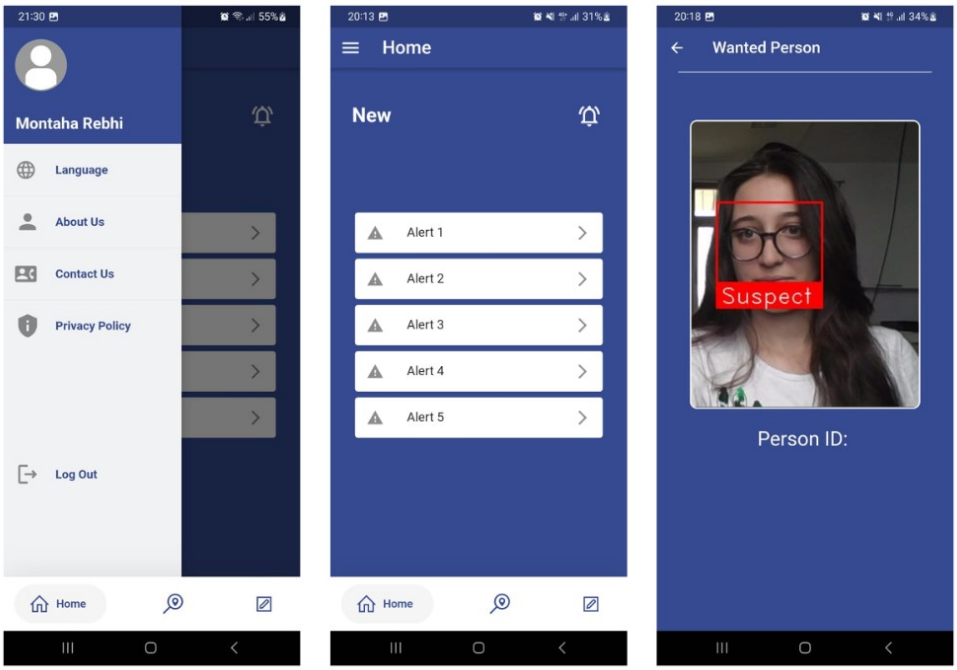


Figure 4.9: Home page components

- Web Application

• The web admin can check policemen list . To ensure this functionality, we have created the ”Table List” interface.

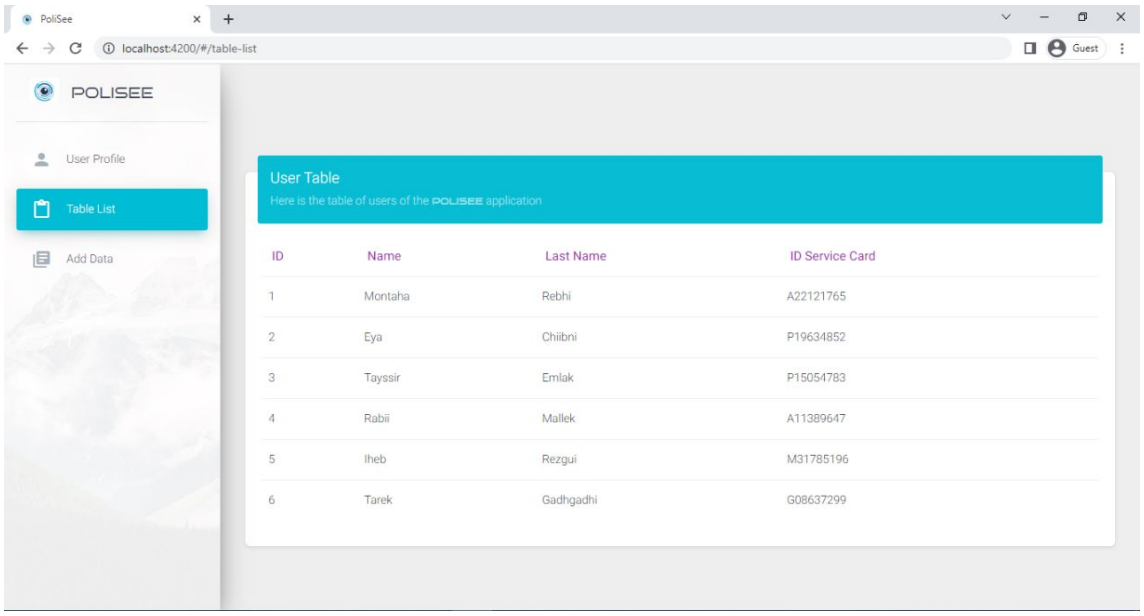


Figure 4.10: Update User Interface

• The admin must add,when needed,suspected people’s images the the data base of our face recognition system. To ensure this functionality, we have created the ”Add Data” interface .

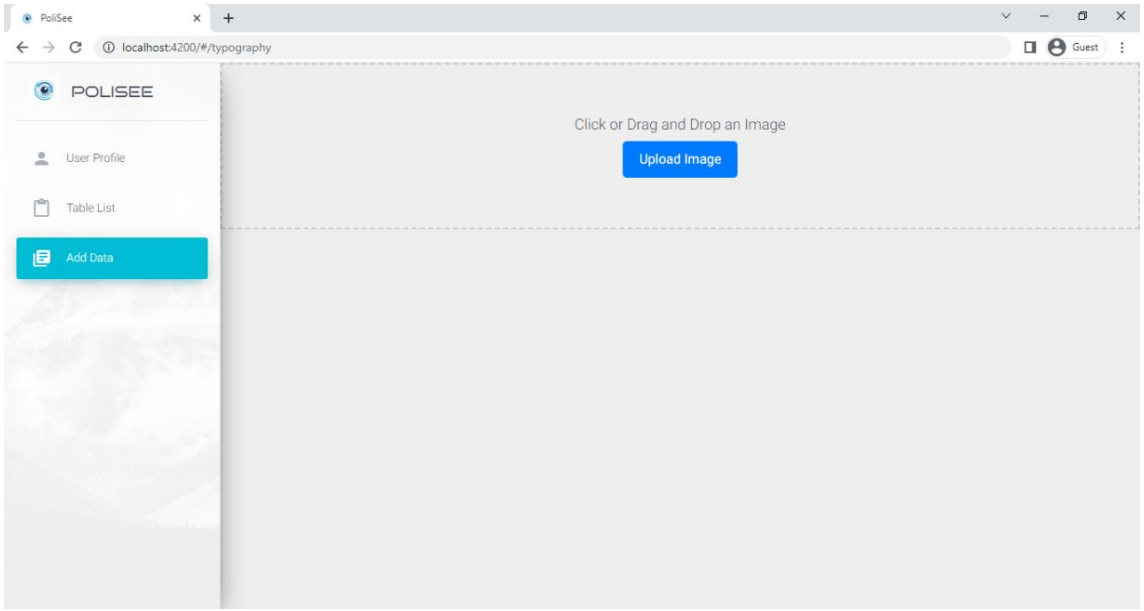


Figure 4.11: Update User Interface

4.4.3 Backend implementation

The main objective of this part will be the development of the backend application which manages the "behind the scenes" functionality of our mobile application while interacting with a defined database.

- We started by arranging the initial environment configuration so that the Node.Js framework and MongoDB work perfectly together.
- Next, to build a flexible and powerful API, we used Express.js and Mongoose in Node.js.
- Express.js handled HTTP requests, while Mongoose facilitated MongoDB integration, enabling efficient CRUD operations.
- Our Node.js backend project has many applications. Each application is a set of code files based on the MVC pattern and interacts with the MongoDB database through the Model layer. A model class represents a collection in our database.
- The next step in our Node.js and MongoDB implementation was defining route handlers.
- Node.js route handlers handle incoming HTTP requests and return appropriate responses. We then map these route handlers to specific URLs, ensuring that each request is directed to the corresponding handler function.
- We have developed our Node.js backend application integrated with MongoDB that offers REST APIs for POST / GET / PUT user entities. These APIs enable easy interaction with the application, allowing for CRUD operations on the data.

The following screenshot captures the ThunderClient interface illustrating the POST method employed for submitting registration data:

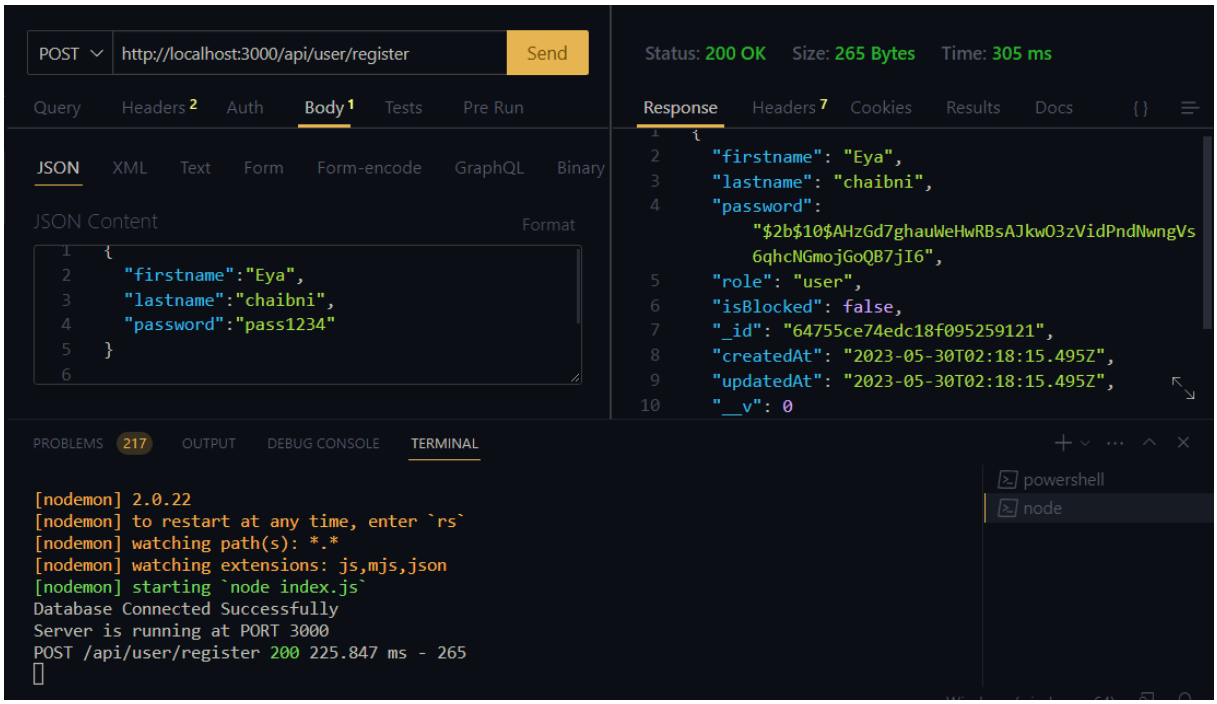


Figure 4.12: Thunder client console: API 'POST' request (register)

The figure 4.13 represent the ThunderClient interface showcasing the POST method utilized for user login.

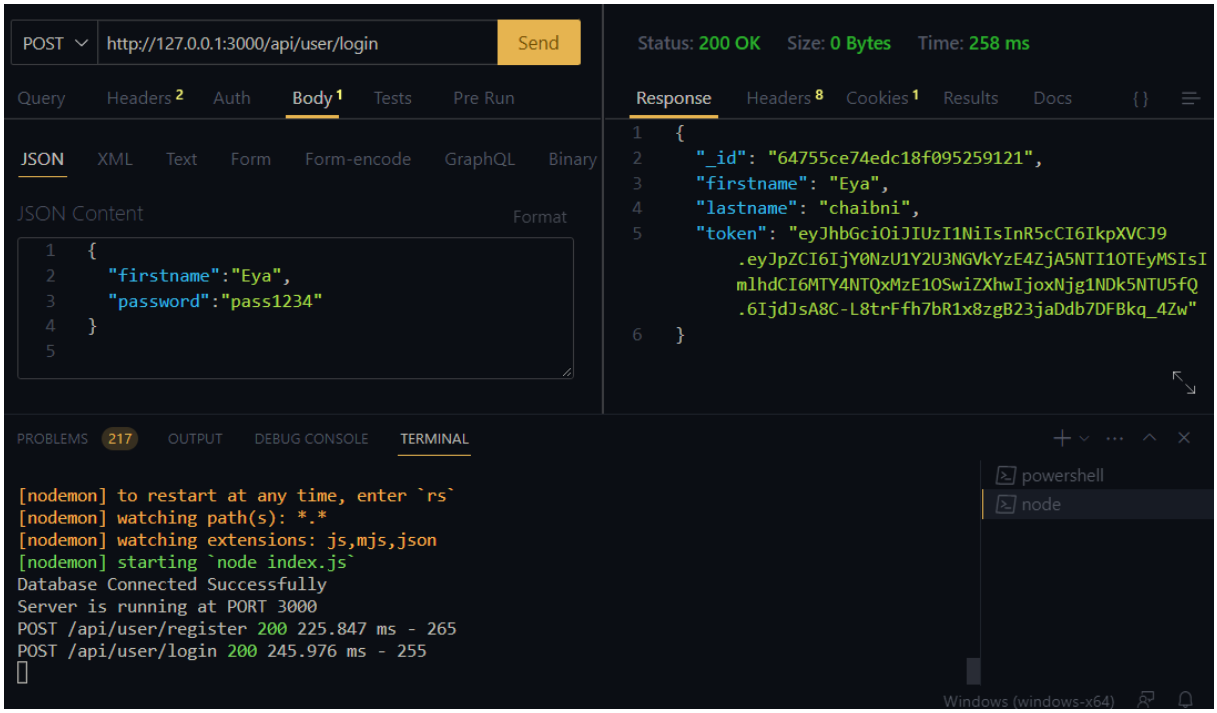


Figure 4.13: Thunder client console : API 'POST' request (login)

The screenshot displays the ThunderClient console, demonstrating an API 'GET' request for retrieving user information.

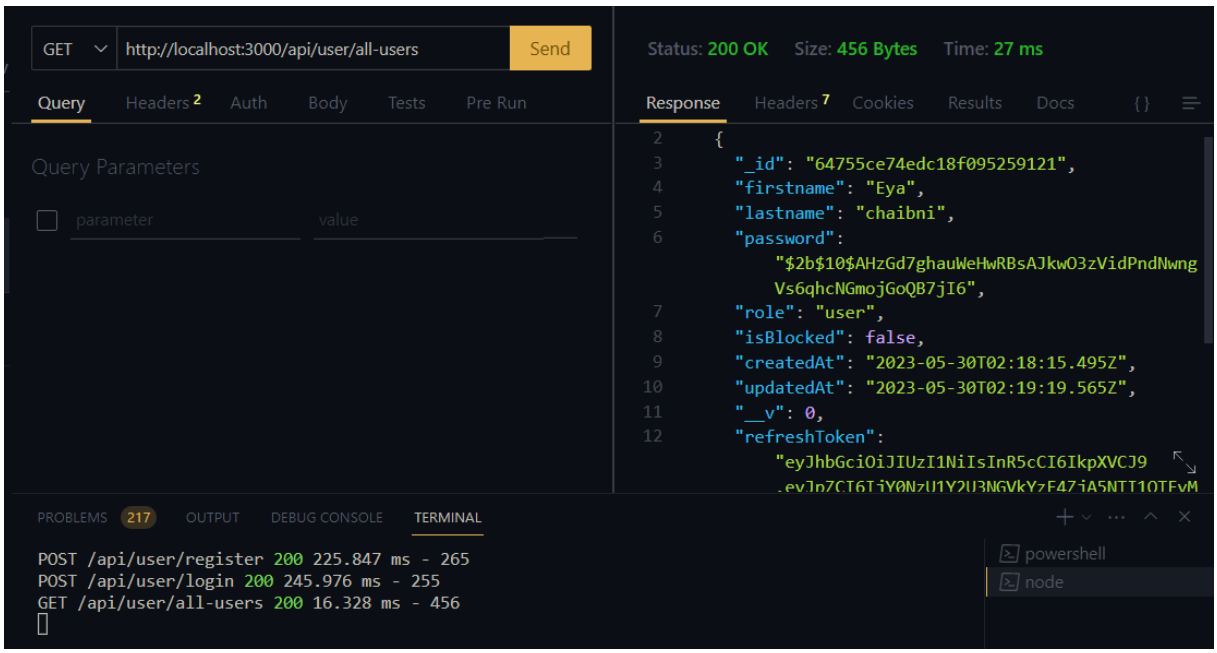


Figure 4.14: Thunder client console : API 'GET' request

The following screenshot shows the MongoDB console, revealing the database interface and data representation within the database.

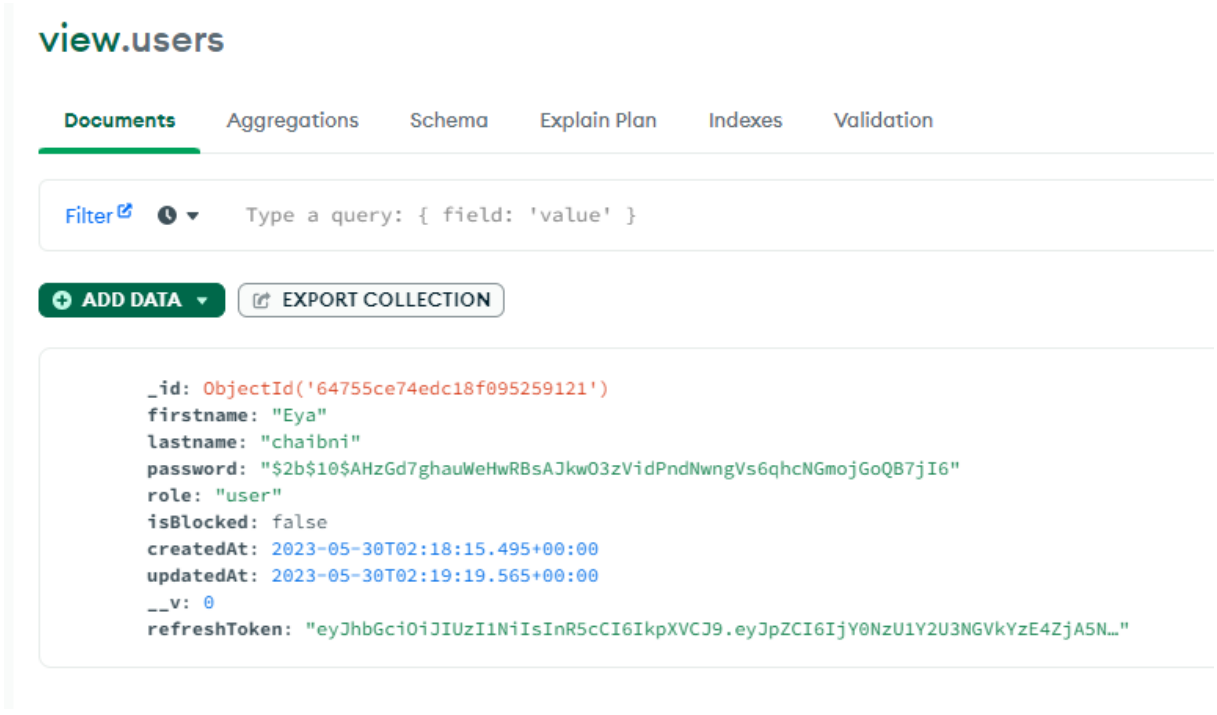


Figure 4.15: MongoDB console (1)

4.5 System integration

After matching the faces, the system pushes a notification to the mobile application. In this section, we will see how different parts are integrated to achieve that objective.

4.5.1 Saving image data in MongoDB

This step involves:

- Establishing a connection to the MongoDB server using the “PyMongo” library.
- Select the appropriate database within MongoDB where you want to store the face images. (Annexe A)
- In the next step, we will use the ”Gridfs” library for storing the images into the MongoDB database.(Annexe A)

After executing our code our image is stored in ’fs.files’ folder with an ObjectID where you can see all the detail related to the image like -id, filename, length and uploadDate.As shown in the figure 4.16

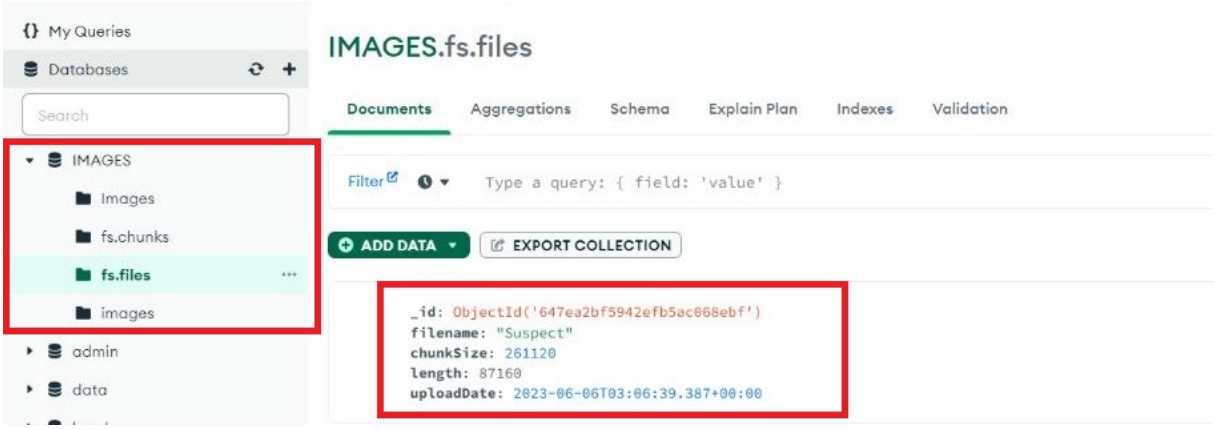


Figure 4.16: MongoDB console (2)

4.5.2 Displaying data on the mobile application

Next, we need to display the image stored in MongoDB on the mobile application.

This step involves:

- First, we establish a connection between the Node.js application and the MongoDB database.
- Then we set up an API endpoint in the Node.js application to handle the notification button click event from the front end. This endpoint should receive the necessary data to identify the specific image to retrieve.

- After converting the image data depends on how it is stored in MongoDB. We send the image data in the response.
- Next in the front-end code, we set up an event listener for the notification button.
- When the button is clicked, the front-end sends an HTTP request to the Node.js backend, which retrieves the image data from MongoDB and sends it back as an HTTP response to be displayed on the front end.

Conclusion

This chapter presents the last part of our report, during which we underlined the proposed approach as well as the different stages adopted during our project realization.

General Conclusion and Perspectives

During this end-of-studies internship, we developed a face recognition system for law enforcement agencies and personnel. This system is mainly composed of two key elements: a face recognition system and a mobile application. Its goal is to help law enforcement personnel quickly and accurately identify criminals and suspects to prevent and solve crimes. This will ensure public safety.

The project provided an opportunity to utilize both AI and software developement.

Through the AI aspect of this project, we gained insights into improving face recognition systems. We learned how AI neural networks enhance accuracy and efficiency while using DNN pre-trained model for face detection and facenet model for face recognition. As a result, policmen can effectively and quickly identify suspects and wanted criminals to enhance public safety.

In the mobile application development aspect of this project, we gained valuable knowledge and skills in creating an optimal user experience for law enforcement personnel using the combined power of Flutter Dart, and Node.js integration to develop a comprehensive mobile application. This integration allowed us to provide them with a comprehensive platform that combines the power of the face recognition system with the convenience of a handheld device.

This work has achieved its objectives, but, like all work, it cannot claim to be perfect, which brings us to the prospects for improving our solution.

We can propose perspectives like working with larger data sets or expanding the database to improve face recognition system accuracy. Also focusing on integrating the system with body cameras to enable recognition at longer distances would be beneficial.

Webography

[1] Telcotec, "Telcotec". Website, [Online]. Available: https://telcotec.tn/ (last consultation: 13/02/2023).

[2] Brennan Center for Justice, "Predictive Policing Explained". Research report, [Online]. Available: https://www.brennancenter.org/our-work/research-reports/predictive-policing-explained (last consultation: 05/04/2023).

[3] Chicago Communications, "What Police Communication Technology Do Officers Need?". Blog post, [Online]. Available: https://www.chicomm.com/blog/ (last consultation: 05/04/2023)

[4] Springer, "Chapter 93 - Title of the Chapter". In: Book Title, Publisher, [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-10-0557-2_93/ (last consultation: 05/04/2023).

[5] Adobe Business Blog, "Waterfall". Blog post, [Online]. Available: https://business.adobe.com/blog/basics/waterfall (last consultation: 06/03/2023).

[6] Asana, "Agile Methodology". Resources, [Online]. Available: https://asana.com/resources/agile-methodology (last consultation: 06/03/2023)

[7] Kanbanize, "What is Kanban?". Kanban Resources, [Online]. Available: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban (last consultation: 07/03/2023).

[8] Project Management Academy, "4 Important Kanban Principles to Remember When Leading an Agile Team". Blog post, [Online]. Available:

https://projectmanagementacademy.net/resources/blog/

(last consultation: 07/03/2023).

[9]Amazon Web Services, "Facial Recognition". What Is, [Online]. Available: https://aws.amazon.com/what-is/facial-recognition/ (last consultation: 22/02/2023).

[10] Nefian, A., Hayes, M. H. (2003). "Face detection and recognition using hidden Markov models". In: Proceedings Ninth IEEE International Conference on Computer Vision, 2003. ICCV 2003, 1, pp. 362-369, [Online]. Available: https://www.semanticscholar.org/ (last consultation: 22/02/2023).

[11] ResearchGate, "A distance weighted linear regression classifier based on optimized distance calculating approach for face recognition". [Online].

Available: https://www.researchgate.net/publication/335003517_A_distance_weighted_linear_regression_classifier_based_on_optimized_distance_calculating_

approach_for_face_recognition (last consultation:10/04/2023)

[12] Le Monde, "Yann LeCun : l'intelligence en réseaux". Sciences article, [Online].

Available: https://www.lemonde.fr/sciences/article (last consultation: 17/04/2023).

[13] Towards Data Science, "Convolutional Neural Networks Explained". [Online]. Available: https://towardsdatascience.com/ (last consultation: 17/04/2023)

[14] MathWorks, "Feature Extraction". [Online]. Available: https://www.mathworks.com/discovery/feature-extraction.html#:~:text=Feature%20extraction%20refers%20to%20the,directly%20to%20the%20raw%20data (last consultation: 14/04/2023)

[15] DataGen, "Image Classification Using CNN". [Online]. Available: https://datagen.tech/guides/image-classification/image-classification-using-cnn/#:~:text=Image%20classification%20is%20the%20task,label%20of%20new%2C%20unseen%20images (last consultation: 14/04/2023)

[16] arXiv, "Very Deep Convolutional Networks for Large-Scale Image Recognition". [Online]. Available: https://arxiv.org/abs/1503.03832 (last consultation: 06/04/2023).

[17]Towards Data Science, "Triplet Loss: Advanced Intro". [Online]. Available: https://towardsdatascience.com/triplet-loss-advanced-intro-49a07b7d8905 (last consultation: 07/04/2023).

[18] ScienceDirect, "Cosine Similarity". [Online]. Available: https://www.sciencedirect.com/topics/computer-science/cosine-similarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis (last consultation: 07/04/2023).

[19] Foliant Documentation, "PlantUML Preprocessor". [Online]. Available: https://foliant-docs.github.io/docs/preprocessors/plantuml/ (last consultation: 19/05/2023).

[20] Vitalflux, "Machine Learning Programming Languages List". [Online]. Available: https://vitalflux.com/machine-learning-programming-languages-list/#:~:text=Python%20is%20a%20programming%20language,variety%20of%20libraries%20and%20tools (last consultation: 19/05/2023).

[21] NumPy, "Absolute Beginners Tutorial". [Online]. Available: https://numpy.org/doc/stable/user/absolute_beginners.html (last consultation: 07/06/2023).

[22] dlib, [Online]. Available: http://dlib.net/ (last consultation: 07/06/2023)

[23] OpenCV, "About OpenCV". [Online]. Available: https://opencv.org/about/ (last consultation: 07/06/2023).

[24] GitHub, "ageitgey/face-recognition". [Online]. Available: https://github.com/ageitgey/face-recognition (last consultation: 07/06/2023)

[25] Flutter, "Flutter". [Online]. Available: https://flutter.dev/ (last consultation: 07/06/2023).

[26] Node.js, "Learn Node.js". [Online]. Available: https://nodejs.dev/en/learn/ (last consultation: 07/06/2023)

[27]Visual Studio Code, "Visual Studio Code Documentation". [Online]. Available: https://code.visualstudio.com/docs (last consultation: 07/06/2023).

Annexes

Annexe A: Definitions

1. Python

Is a programming language that comes with a number of very attractive machine-learning libraries as well as a set of powerful packages for analytical needs. This encouraged us to choose Python over other programming languages [20].

2.OpenCV

OpenCV (Open Source Computer Vision Library) it has interfaces for Python and Java as well as C++. It is a powerful library for computer vision and image processing tasks. It provides a range of functions and algorithms to manipulate and analyze images and videos[21].

3. Numpy

NumPy (Numerical Python) is an open source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. The NumPy library contains multidimensional array and matrix data structures[22].

4. Dlib

Dlib is a modern C++ toolkit containing learning algorithms and tools to create complex C++ software to solve real-world problems. It is used in both industry and academia in a wide range of fields, including robotics, embedded devices, mobile phones, and large high-performance computing environments[23].

5. Face-recognition

The library face recognition is based on deep learning, it supports single-shot learning which means it needs a single picture to train itself to detect a person and promises accuracy greater than 96% using a single training image[24].

6. Flutter Dart

Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase[25].

7. Node.Js

Node.js is an open-source and cross-platform JavaScript runtime environment used for the back end development. It runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant[26].

8. Angular

Angular is a development platform for building mobile and desktop web applicationsusing

TypeScript/JavaScript and other languages[27].

9. VsCode

Visual Studio Code is a lightweight but powerful source code editor . It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages and runtimes (such as C++, Java, Python, PHP, Go, .NET).

10. Anaconda3

Anaconda is a Python programming language distribution for scientific computing that promises to make package management and deployment easier.

11. PlantUML

PlantUML is a tool to generate diagrams from plain text. This preprocessor finds Plant UML diagrams definitions in the source and converts them into images on the fly during project build.

12. Figma

Figma is a combination of the words “figure” and “magnify.” It’s an easy-to-use vector based design tool that allows users to create high-fidelity prototypes quickly and efficiently.

13. MongoDB

MongoDB Atlas is a multi-cloud database service by the same people that build MongoDB. Atlas simplifies deploying and managing your databases while offering the versatility you need to build resilient and performant global applications on the cloud providers of your choice.

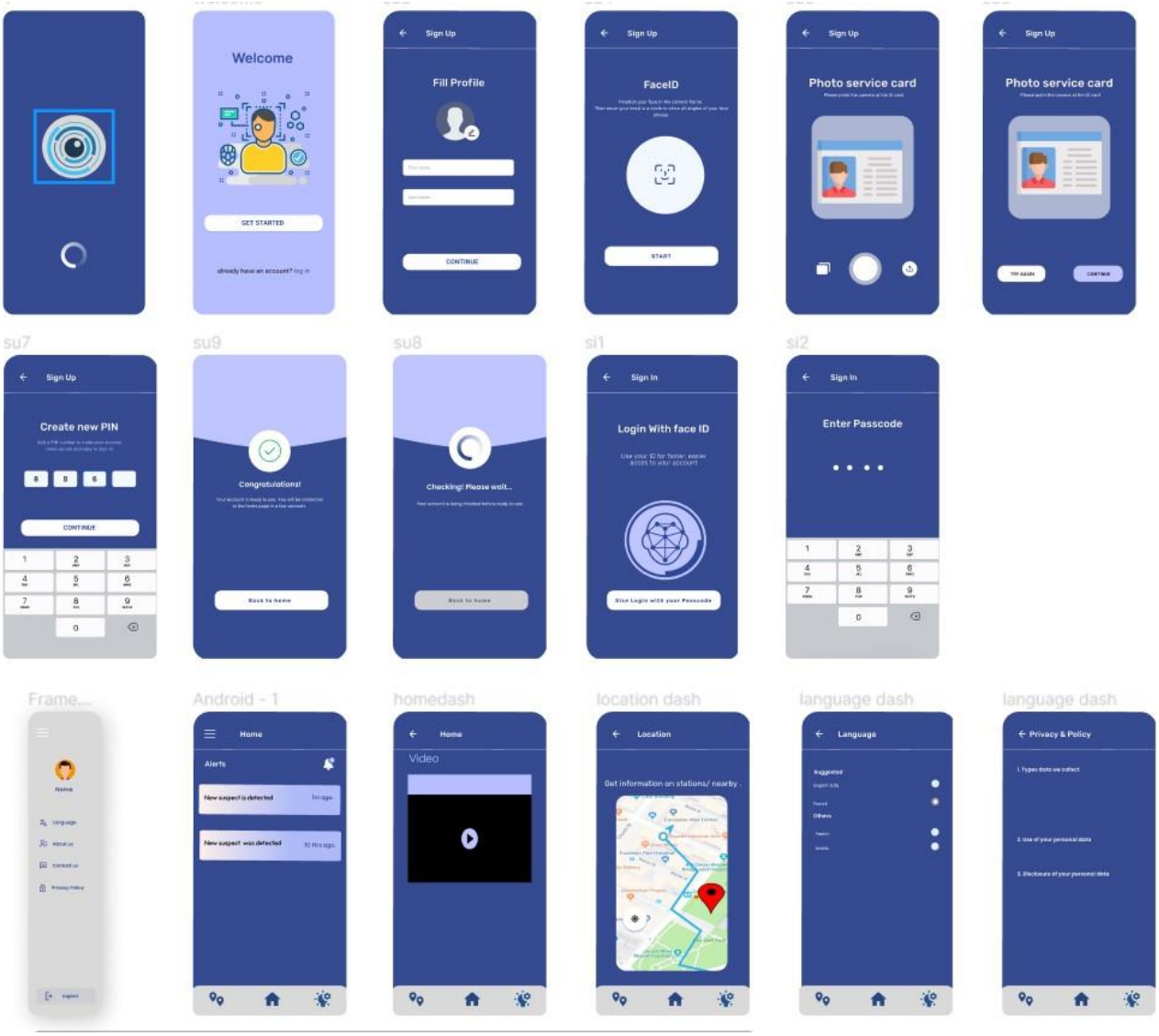
14. PyMongo

It is a Python distribution containing tools for working with MongoDB, and is the recommended way to work with MongoDB from Python.

15. GridFS

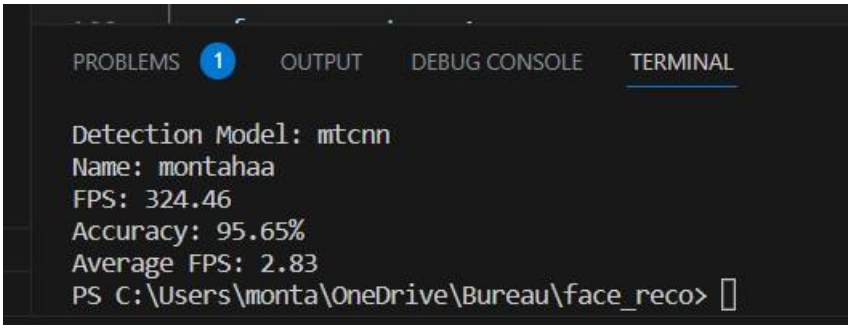
It is a specification for storing and retrieving files that exceed the BSON-document size limit of 16 MB.

Annexe B: Mobile Application Design



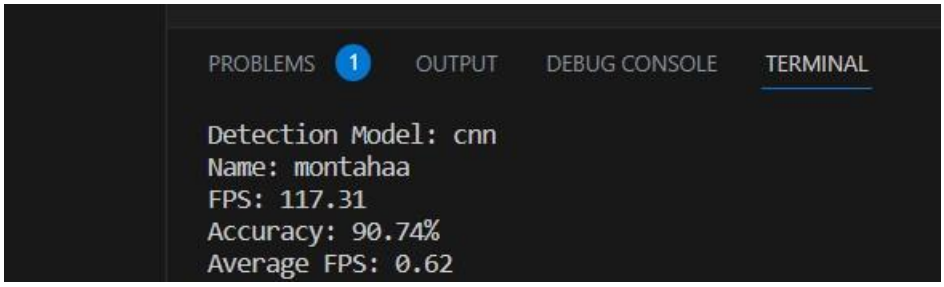
Annexe C: Face detectors comparison

- MTCNN

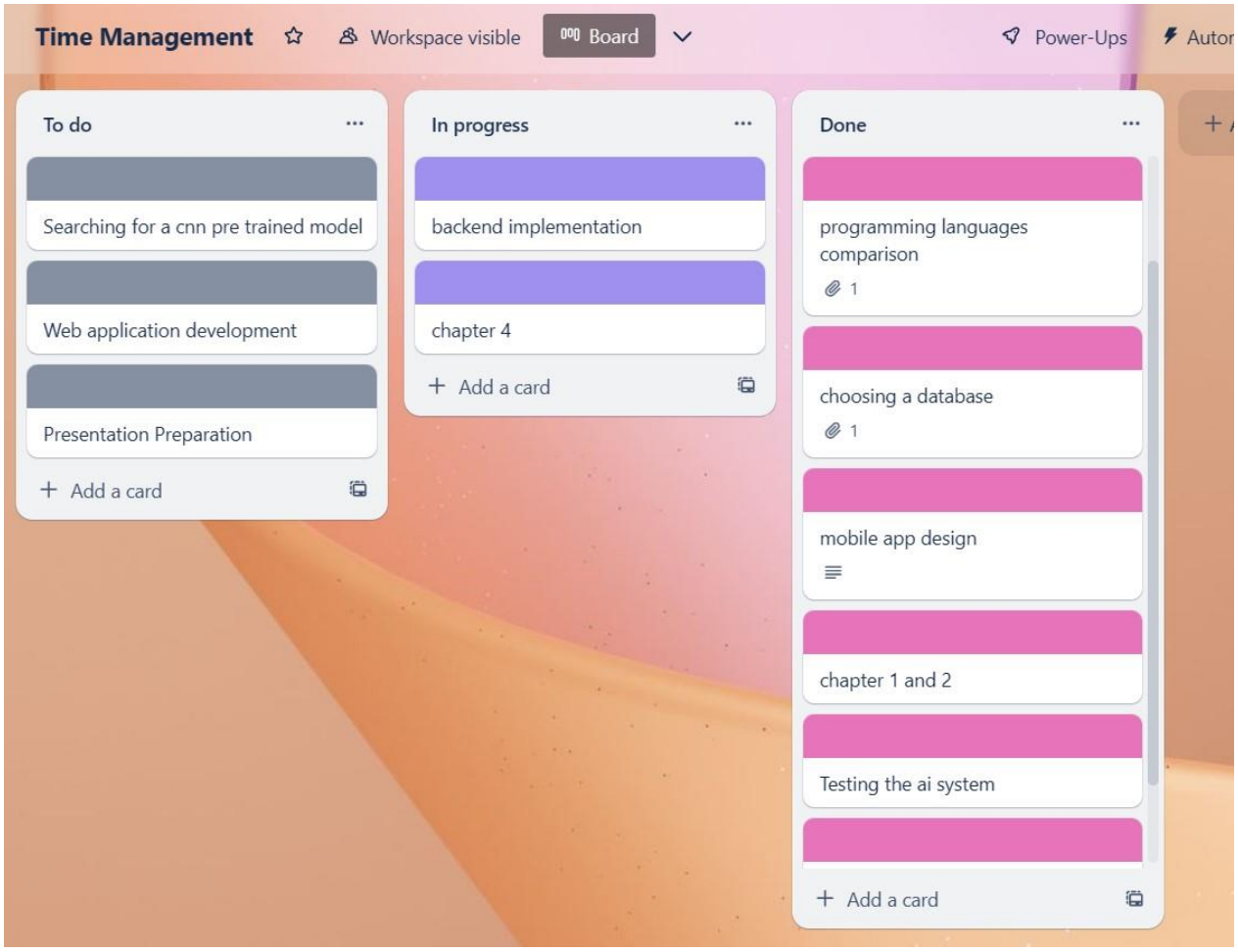


- DNN

- CNN



Annexe D: Trello interface



Résumé

Ce rapport fait partie du projet de fin d'études réalisé au sein de Telcotec pour l'obtention du diplôme de licence appliquée en sciences et technologies de l'information et de la communication. L'objectif de notre projet est de développer un système de reconnaissance faciale basé sur l'apprentissage profond pour comparer les caractéristiques faciales en temps réel avec une base de données enregistrée. Le système analyse les images capturées pour fournir une correspondance et affiche les informations pertinentes sur une application mobile.

Mots clés : Reconnaissance faciale, Apprentissage profond, Caractéristiques faciales, Application mobile

Abstract

This report is part of the End of Study Project carried out within Telcotec to obtain the Applied Bachelor in Information and Communication Sciences and Technologies Diploma. The purpose of our project is to develop a face recognition system based on deep learning to compare real-time facial features with a registered database. The system analyzes captured images to provide a match and displays relevant information on a mobile application.

Keywords: Face recognition, Deep-learning, Facial features, Mobile application.