Université Paris 13 Villetaneuse

Internet des objets -IOT

Développement d'un objet connecté smart door

Présenté par :
MTIBAA Ameni
OBROCHTA André
RAHERINIAINA Lucky
KADRI Adlane

Enseignants :
Aomar OSMANI
Hamid MASSINISSA

Contents

1 Introduction

Notre projet intitulé « Smart Door » s'inscrit dans le cadre du module Internet des objets connectés (IoT).

Avant d'arriver à ce que l'on appelle "clef connectés" actuel, l'être humain aux cours du temps a chercher par diverses techniques à sécuriser son bien (maison, argent, voiture) et à restreindre l'accès Par diverses techniques telles que le verrouillage mécanique. Les avancées scientifique ont permis de concevoir des systèmes à verrouillage électrique.

À l'air de l'internet et des données actuel, des nouvelle solution sont présent dont 'les clefs connecter'.

Pour débuter ce projet, nous devons nous appuyer sur un document essentiel à l'élaboration de ce travail : le cahier des charges. Ce dernier sera notre guide pour définir notre projet tant sur les objectifs à atteindre et comment y parvenir.

Par la suite, nous allons résumer l'état de l'art de notre produit, cette étude a pour but de chercher toutes les informations existantes concernant un domaine et à en faire une synthèse afin d'avoir un suivit des évolutions techniques, d'identifier les meilleures pratiques afin d'augmenter la qualité des produits tout en diminuant les coûts de production et anticiper sur la concurrence.

Nous parlerons de la conception par la suite, c'est-à-dire les différentes étapes qui ont permis d'aboutir à un système fonctionnel.

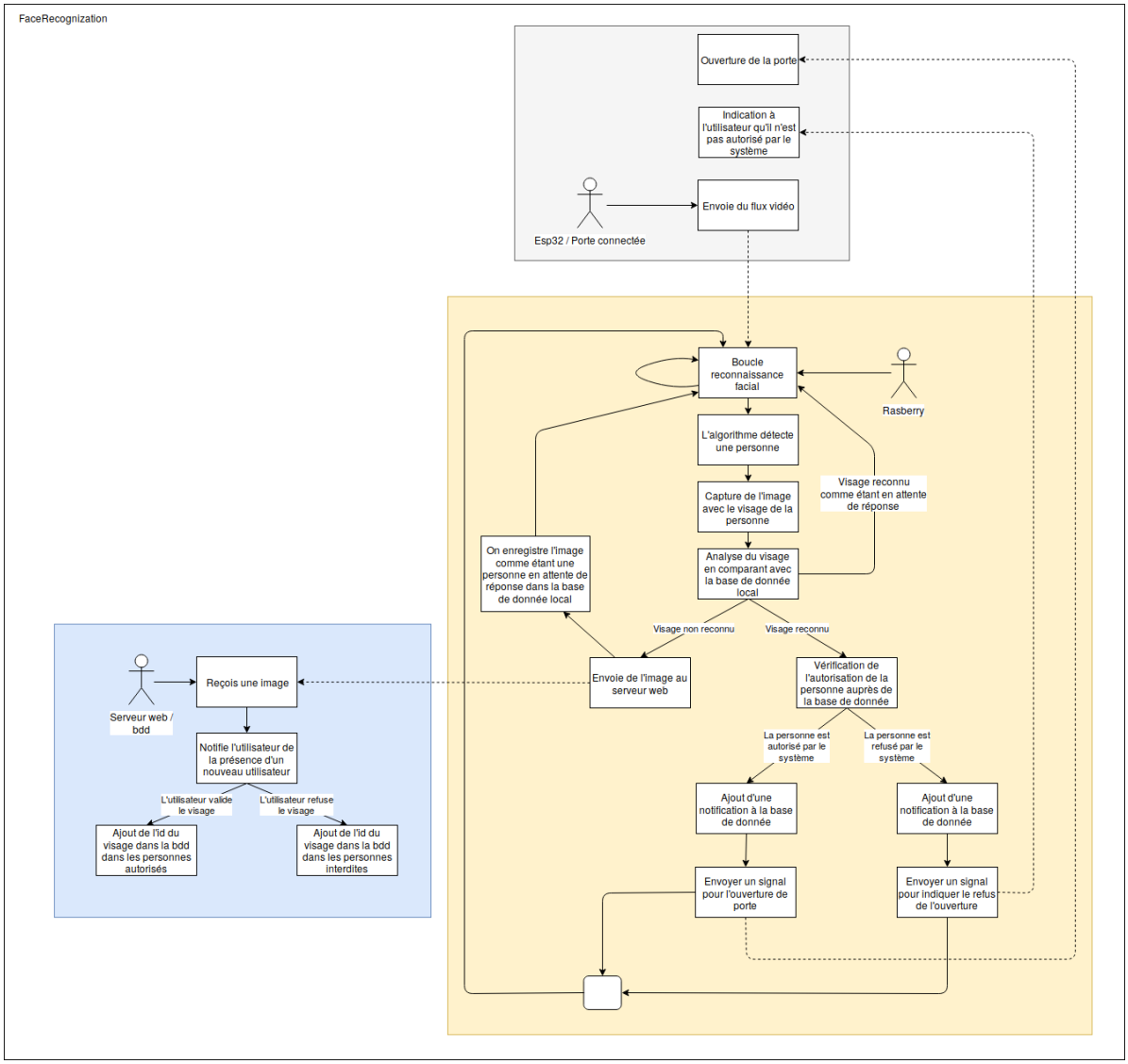
Et enfin, chaque membre du groupe apportera une conclusion personnelle sur son ressentie du module, ce que cela a pu apporter et des élément améliorable.

2 Cahier des Charges

Cette partie est une synthèse en quelques pages de l'analyse réalisée dans le cahier des charges.

- Concernant l'analyse des besoin du client, nous pouvons dire que le but de ce projet est de réaliser une clé connectée afin d'ouvrir et fermer une porte à l'aide de la reconnaissance faciale. En effet, cette solution peut présenter plusieurs avantages pour le client :
  - Une personne souhaitant mettre sa maison ou son appartement en location peut donner l'accès à son bien uniquement pendant une période défini.
  - Le propriétaire n'a plus à se soucier des clés pour rentrer chez lui, de plus il peut très bien laisser la porte à une personne à distance.
- Les objectifs de SmartDoor :
  - Le système doit ouvrir la porte à l'utilisateur lorsque celui-ci est reconnu, ceci est indiqué avec un led bleu.
  - Si un inconnu tente d'ouvrir la porte, l'inconnu verra le led rouge s'illuminer et une notification sera envoyé au propriétaire.
  - Le propriétaire pourra ajouter des personnes dans la liste des personnes autorisées.

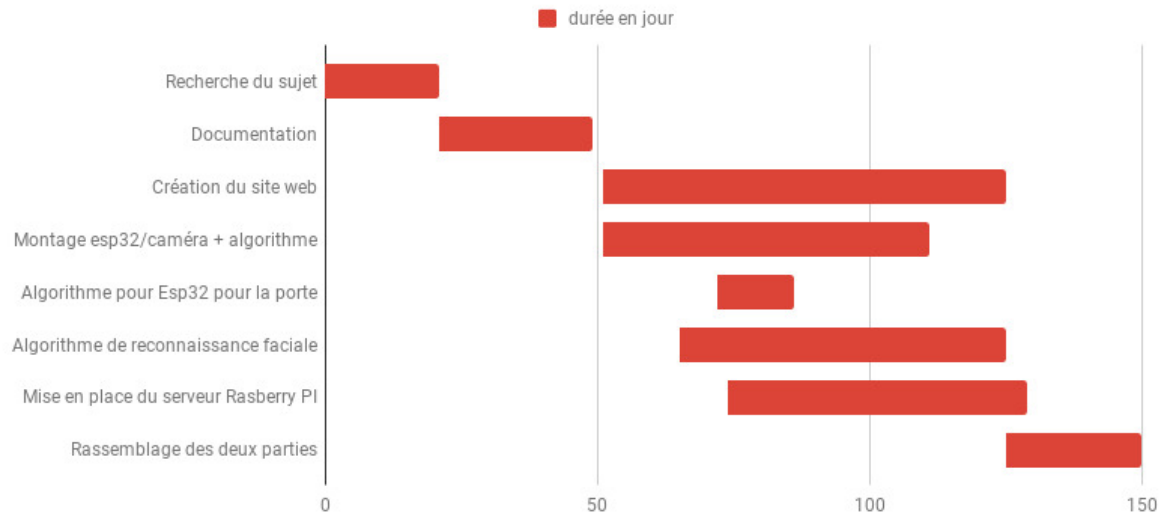
2.1 Diagramme des Cas d'utilisation



2.2 Les limites de SmartDoor

- La reconnaissance faciale en 2D reste une technologie très faillible et présente des limites. Il est possible de tromper l'algorithme avec une simple photo voir même une portrait peinture...
- De plus, le taux de probabilité de reconnaissance n'est pas de 100%, il se peut que la reconnaissance faciale détecte un faux positif, les faux positifs sont particulièrement présents lorsque le visage est très éloigné de l'objectif ou que la photo est de mauvaise qualité.

2.3 Diagramme de Gantt



Le changement majeur qui a été ajouté par rapport à ce qui a été mentionné dans le cahier des charges initial est l'ajout du Rasberry Pi model B+ comme matériel en plus.

Nous avons fait ce choix, car nous avons jugé pour le professionnalisme du projet qu'il était intéressant de rendre accessible à un plus grand public l'application.

Le Rasberry nous a paru être une solution évidente par son bas prix par rapport a un vrai serveur, sa puissance, car dans le projet nous avons besoin de puissance pour faire de la reconnaissance, sa basse consommation en électricité, par une forte communauté en IoT qui utilise ce matériel, il est très facile de trouver de la documentation.

De plus par son ergonomie en effet, il est très facilement transportable.

3 Etat de l'art

Cette partie ne peut remplacer l'étude réalisée dans ce document "état de l'art", c'est pour cela que nous en faisons une synthèse. Ainsi, nous mettrons plus de valeur au produit ayant une réelle similarité avec celui que nous souhaitons concevoir.

Dans l'étude de l'état de l'art nous avons trouvé diverses techniques aux cours du temps qui a permis à l'homme de sécuriser son bien.

Cela a commencé il y a 4500 ans avec l'apparition des premières serrures mécanique, les techniques se sont améliorées aux cours des années jusqu'au codage mécanique qu'on retrouve dans les divers coffres-forts de notre temps.

Avec l'évolution fulgurante des découvertes scientifique dans le domaine physique, des techniques de verrouillage électrique ont vu leur jour. Nous avons comme exemples les digicodes, l'accès par carte électromagnétique ou encore les verrouillages centraliser qu'on retrouve dans l'automobile. Avec son application Maison, Apple HomeKit est depuis

septembre 2016, Apple souhaite être présent sur le marché de l'IOT (internet des objets)e et plus particulièrement sur les objets connectés liés au logement.Ainsi plusieurs produit comme la serrure connectée Danalock V3, conçu pour fonctionner avec HomeKit d'Apple. Il permet de contrôler l'accès au domicile facilement et permet de verrouiller une porte de manière sécurisée.

À l'air d'internet et de l'apprentissage artificiel divers produit et technique ont vu leur jour.

Dont le Ring qui appartient à Amazon ou encore l'August Doorbel Cam Pro (August Home Inc) qui sont des sociétés spécialisées dans les produites domotiques. Ils ont mis en place des produits qui permettent d'enregistrer des visages et d'en faire un historique, alerter quand une personne est détectée, possibilité d'accepter ou refuser une personne.

En dernière partie, nous avons parlé de divers technologie de reconnaissance biométrique cela inclus la reconnaissance d'empreinte, la reconnaissance vocale, la reconnaissance oculaire et enfin la reconnaissance faciale.

Celui qui nous intéresse étant la reconnaissance faciale, nous avons découvert qu'il s'agit d'une application logicielle visant à reconnaître une personne grâce à son visage de manière automatique. Ces systèmes sont généralement utilisés à des fins de sécurité pour déverrouiller ordinateur/mobile/console, mais aussi en domotique.
Nous nous somme donc intéressée au leader sur la marche qui développe ce type de logiciel, nous pouvons citer :

- **GaussianFace :** Algorithme chinois considérer plus performant que l'être humain.
- **DeepFace :** Algorithme créé et utilisé par FaceBook.
- **FaceNet :** Algorithme créé par Google avec une version open source.
- **OpenCV :** Librairie de reconnaissance faciale open source et nécessitant une petite unité de calcul.

- **Netatmo :** Start-up française spécialisée dans les objets connectés. En 2015, réalise une caméra, Welcome, qui permet de reconnaître les visages en cas d'intrusion dans une maison, utilisable sur Apple HomeKit et l'assistant google,.

Nous pouvons donc tester l'utilisation d'OpenCV, car il nous semble être la plus simple pour notre projet, car elle est totalement libre.

4 Promotion du produit

Vous voulez un gadget qui vous permet de voir qui est devant votre porte sans être à la maison ? Vous voulez savoir qui y entre et sort et à quelle heure ? Vous avez un invité devant la porte et vous conduisez.
Comme un fou pour lui ouvrir la porte et ne pas le laisser dehors longtemps ? Vous voulez que votre porte vous reconnaisse quand vous oubliez vos clés ? Ou vous voulez qu'elle vous informe quand il y a des inconnus devant votre porte ?
Vous vous dîtes sûrement que cela va vous coûter une petite fortune. Et si on vous dit que cela est possible pour moins de 50 euros ? Oui, vous ne rêvez pas ! C'est "smart door" une caméra que vous placez devant votre porte et une application qui vous permet de superviser en temps réel le trafic devant votre porte.

5 Fonctionnement

Pour réaliser notre projet, nous avons réparti le travail en deux équipes :
- **KADRI** et **MTIBAA** pour le développement du système de reconnaissance faciale.
- **OBROCHTA** et **RAHERINIAINA** pour le développement de l'application web.

5.1 Développement du système de reconnaissancefaciale

5.1.1 Streaming et capture d'images

Dans cette partie, nous commencerons par décrire les différents outils/Framework qu'on a utilisés. Nous détaillons, ensuite, comment nous avons relié les différents composants et nous expliquons à la fin un peu l'aspect technique.

1. **Composants** :
   Pour que notre code fonctionne, vous avez besoin des composants suivants:

   - Module caméra
   - Module ESP32
   - Ordinateur avec ESP-IDF

   (a) Caméra:
   Pour la caméra, nous avons opté pour l'OV2640 qui est à la base un module Arduino. Malheureusement, la caméra que nous avons achetée d'Amazon était d'un autre constructeur(MARLIN P. JONES & ASSOC., INC) et ne possédait pas une bonne documentation. Ainsi, quand nous avons voulu essayer les exemples du web. Nous avons rendu compte qu'il y avait seulement 17 Pin utilisable (Le 18e est NC : not connected).
   Les pin disponible sur notre caméra sont les suivant :

   i. Vcc : source de courant
   ii. GND : sol
   iii. VSYNC : Synchronisation verticale , indique le frame actif

7

iv. SCL (= SIOC) : Horloge SCCB
v. SDA (=SIOD) : Données SCCB
vi. HREFF : Référence horizontale
vii. D0 - D7 : Pixel Data Bit 0-7
viii. DCLK (=PCLK) : Horloge de pixel
ix. PWDN : Mise hors tension de la caméra
x. NC : pas connecté

Figure : comparaison entre le module de Arduino à gauche (20 Pin) et notre module à droite (18 Pin)
Heureusement, selon la documentation du constructeur, notre caméra supporte : le format de sortie RGB, différentes tailles de frame (UXGA, SXGA, VGA, QVGA ...) , l'interface I2C et la compression d'image pour la sortie d'image JPEG. Cela semblait bien pour nos besoins.

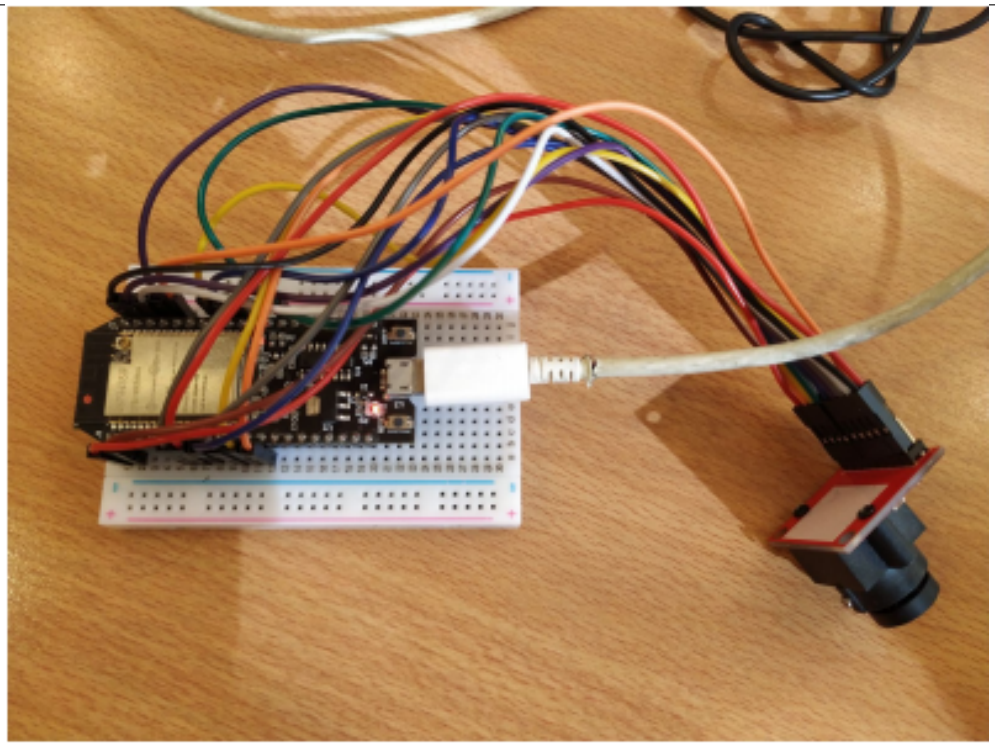
(b) ESP32 :
Après une étude, nous avons conclu que le module ESP32 WROVER est le mieux adapté pour notre projet vu qu'il contient une mémoire PSRAM. (une RAM dynamique avec un circuit de rafraîchissement et de contrôle d'adresse intégré qui lui permet de se comporter de la même manière que la RAM statique).

(c) ESP-IDF :
Au début, nous avons commencé à coder sur Arduino IDE vu l'ergonomie de son interface. Mais nous avons eu besoin des bibliothèques qui ne sont disponible que sur ESP IDF (Comme « camera.h » et « http-server.h »)

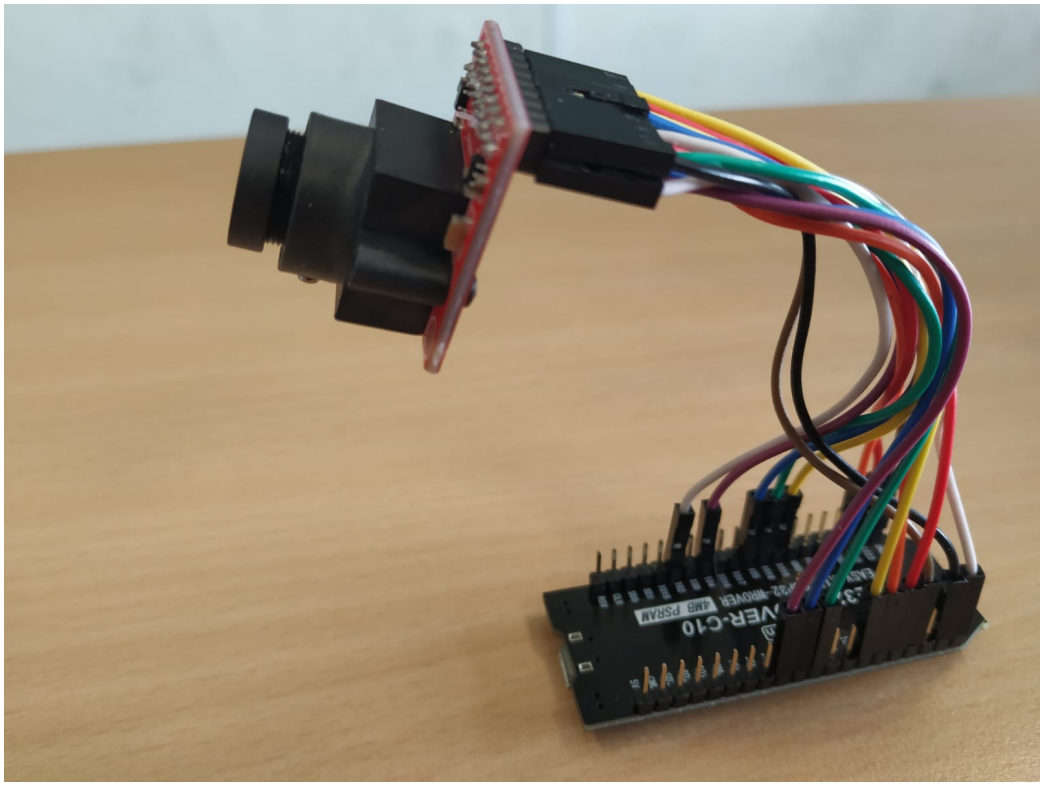
2. **Câblage** :
Le tableau ci-dessous montre les PIN utilisées dans notre projet pour connecter l'ESP32 et la caméra :

Camera Pin	ESP32 WROVER pin
SCL	IO27
SDA	IO26
VSYNC	IO25
HREF	IO23
PCLK	IO22
D0	IO4
D1	IO5
D2	IO18
D3	IO19
D4	SENSOR-VP
D5	SENSOR-VN
D6	IO34
D7	IO35
PWDN	GND
3V3	3V3
RESET	IO2
GND	GND
NC	

Attention ! Relier l'ESP et la caméra avec des longs câbles ne va pas fonctionner :

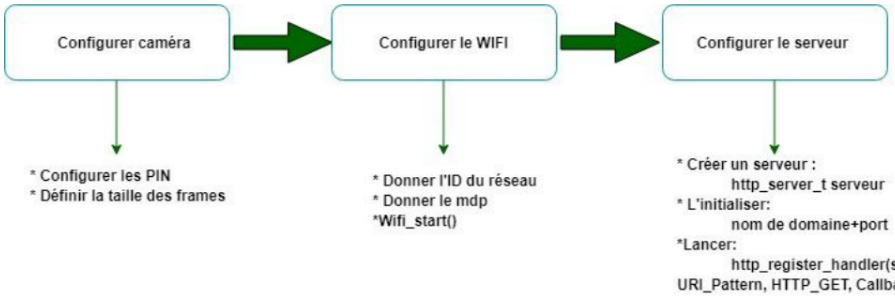


Le câblage qui nous a paru le mieux adapté est le suivant :

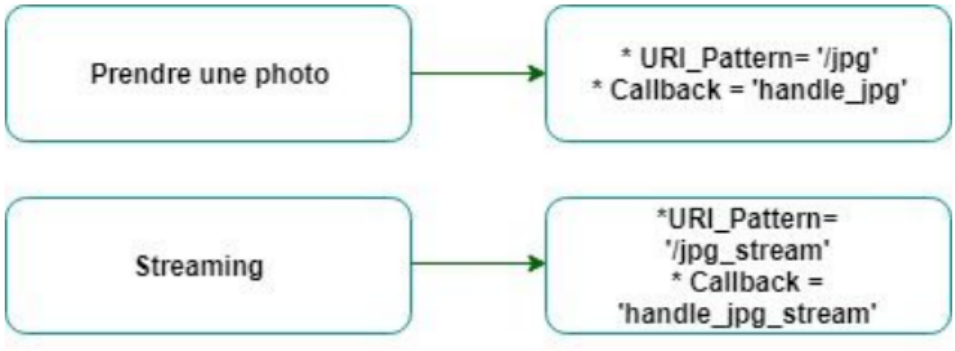


3. **Et le code** ?
   Au début, pour faire fonctionner la caméra, nous avons utilisé l'exemple "camera_web_server" de ESP WHO. Après, nous nous somme inspirés de plusieurs

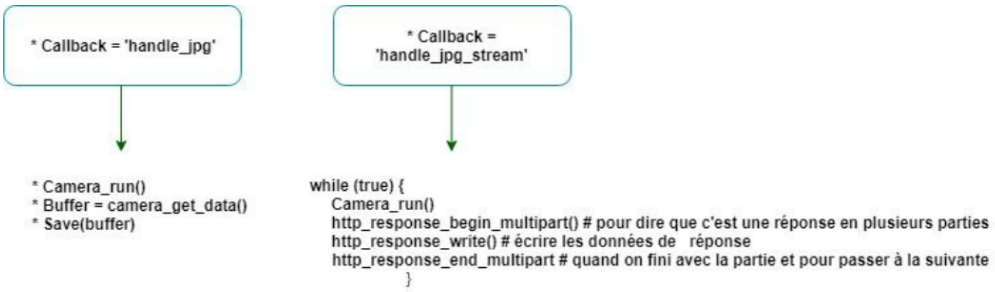
dépôt GitHub pour atteindre notre objectif. Voilà alors les différentes étapes à suivre :



Dans la configuration du serveur, les deux paramètres "URI_Pattern" et "Callback" dépendent de ce qu'on veut que notre caméra fasse : renvoyer des images ou renvoyer des vidéos :



Le code de la fonction de Callback est comme suit :



5.1.2 Méthodes de reconnaissance faciale

Il existe deux méthodes principales dans la reconnaissance des visages, chacune de ces deux méthodes se base sur les trois étapes :

- **Détection de visage :** pour repérer un visage sur une image (c.-à-d. vérifier si l'image contient un visage d'une personne ou pas)

- **Analyse du visage :** cette analyse donne un résultat numérique qui va être utilisé par la suite pour la reconnaissance.

- **Reconnaissance du visage :** en utilisant le résultat numérique obtenu après l'analyse du visage pour le comparer avec les visages stockés dans une base de données

Il est donc nécessaire, avant toute tentative d'identification d'un utilisateur, de constituer une base de données contenant pour chaque utilisateur une ou plusieurs captures d'images. Plusieurs recherches dans ce domaine continuaient, de nombreux algorithmes différents ont été développés en mentionnant l'algorithme PCA (Analyse des composants principaux en utilisant les espaces propres "eigenfaces") et l'algorithme d'Analyse discriminante linéaire.

Les deux méthodes principales de reconnaissance faciales :

- **Reconnaissance 2D :** C'est une méthode classique consiste à reconnaitre un utilisateur à partir d'une photo de lui, prise par une caméra "comme notre cas" ou simplement être déjà enregistré dans un dossier "le dossier en cours sur le schéma en-dessus". La reconnaissance est ensuite effectuée par deux catégories d'algorithmes, ces algorithmes pouvant s'appuyer sur différents éléments, tels que la forme des éléments faciaux, tels que les yeux et leur espacement, la bouche...
  Les deux catégories d'algorithme maintienne précédemment sont les suivants :

  – La première catégorie crée une image géométrique de l'utilisateur en fonction de différents paramètres (taille des éléments de visage, forme et distance les séparant). Les paramètres récupérés sont ensuite codés et comparés à ceux présents dans la base de données.

  – La deuxième catégorie code l'image numériquement, à l'aide d'algorithmes de Fourier "un algorithme de calcul de la transformation de Fourier discrète", d'utilisations d'espaces propres pour créer des vecteurs de pondération, ou en établissant la moyenne sur certaines zones de l'image.

- **Reconnaissance 3D :** C'est une amélioration de la méthode de reconnaissance 2D, l'idée est de crée un modèle 3D en utilisant plusieurs photos prises successivement ou une vidéo pour avoir différents points de vue de la personne à reconnaître afin de créer le modèle 3D.

5.1.3 Détection des visages

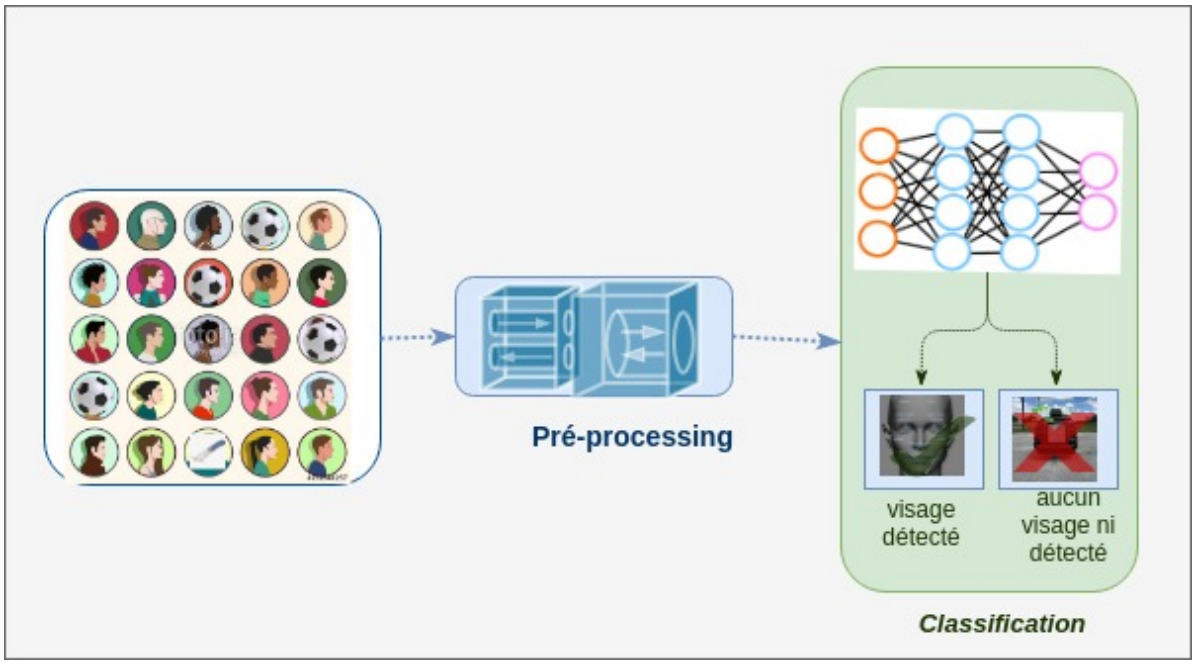
La détection des visages implique la séparation des fenêtres d'image en deux classes, une face contenant des visages (masquant l'arrière-plan "encombrement"). Ce traitement est indispensable avant la phase de reconnaissance. En effet, le processus de reconnaissance de visages ne pourra jamais devenir intégralement automatique s'il n'a pas été précédé par une étape de détection efficace. Le traitement consiste à rechercher dans une image la position des visages et de les Extraire sous la forme d'un ensemble d'image dans le but de faciliter leur traitement ultérieur "ces images vont d'être stockés dans un

dossier pour les utiliser pour la classification des personnes". Le problème est encore compliqué par les variations de pose "vue de profil ou de face", les différences de conditions d'éclairage, de qualités d'image, de rotation du visage et d'illuminations (ce dernier type de difficulté pouvant être surmonté par un prétraitement de normalisation et compensation de l'illumination). La tâche de détection de visages peut être décomposée en deux phase principales.

La première est une phase de **classification** qui prend une image en entrée et génère une valeur binaire de oui ou non, indiquant si des visages sont présents dans l'image (dans notre algorithme si le résultat est non, l'image va être supprimé car il n'y a pas de personne sur l'image).

La deuxième est la phase de **localisation** des visages qui a pour objectif de prendre une image en entrée et de localiser le ou les visages au sein de cette image sous forme de cadre de sélection avec (x, y, largeur, hauteur) "en sauvegardant les visages dans un répertoire pour les traiter par la suite".

Un détecteur de visage idéal serait donc en mesure de détecter la présence de tout visage dans n'importe quel ensemble de conditions d'éclairage, sur n'importe quel fond.

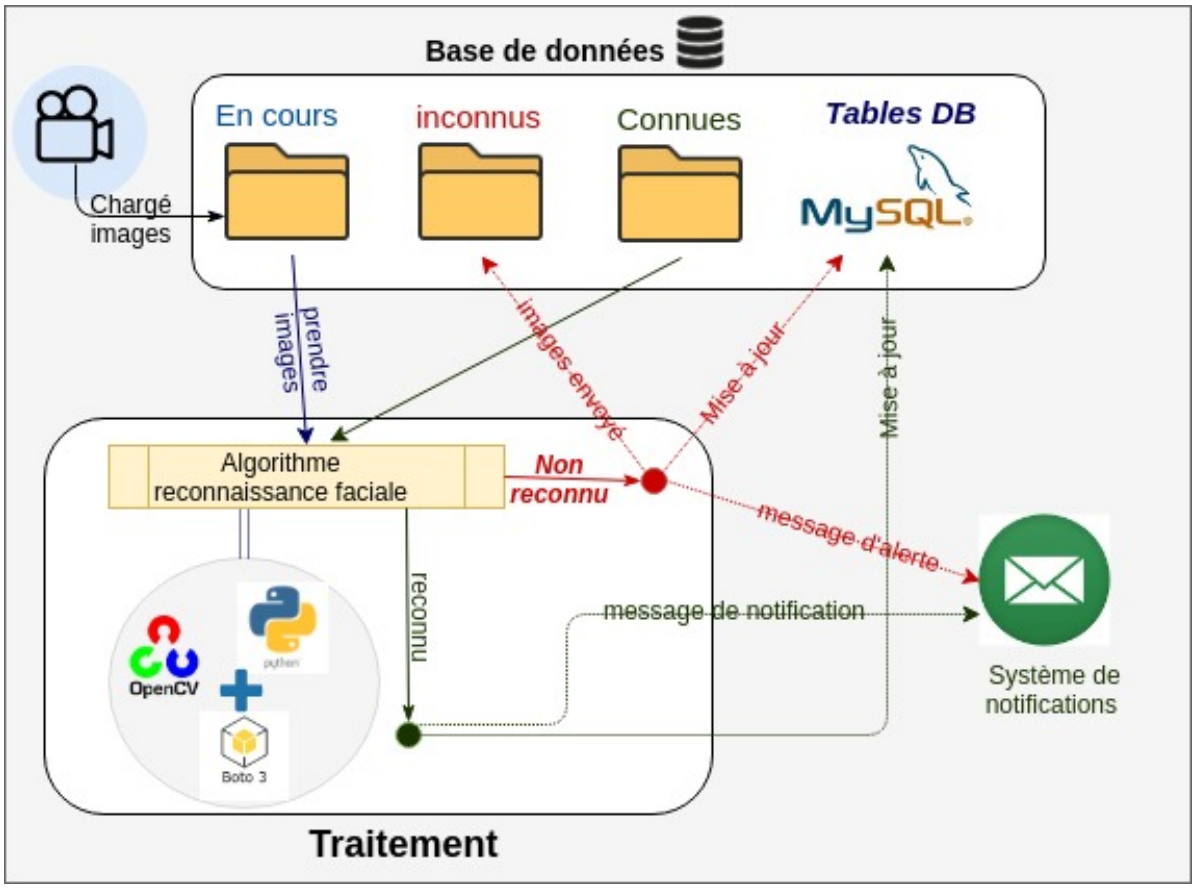


5.1.4 Reconnaissance Faciale

La reconnaissance faciale consiste à identifier un objet déjà détecté en tant que visage connu ou inconnu. Souvent, le problème de la reconnaissance des visages est confondu avec celui de la détection des visages, c'est-à-dire qu'avant de faire la reconnaissance de visage, on doit d'abord vérifier si l'objet détecté est un visage pour décider à la fin si c'est une personne connue ou inconnu en utilisant une base de données de faces afin de valider cette face d'entrée.

Cette partie est consacrée à l'implémentation du système de reconnaissance faciale. En premier temps, nous allons présenter un schéma qui englobe tout le système, par la suite, nous présentons les outils de développement, les différents aspects techniques liés

à l'implémentation de l'application de reconnaissance de visages.



Ce système doit être conçu pour pouvoir être utilisé pendant la détection des personnes autorisé devant la caméra de notre clé connectée. Pour conduire ce système nous avons décidé d'utiliser le langage de programmation python, et le choix des bibliothèques OPENCV, boto3 et face recognition pour python étaient inévitables du fait de l'absence de système capable de détecter et de reconnaître des personnes de manière fiable. Pour le système de notifications nous avons utilisé un module de python pour l'envoie des emails qui s'appelle SMTP.

5.1.5 Reconnaissance Faciale en utilisant OpenCV

OpenCV (Open Source Computer Vision) est une bibliothèque de fonctions de programmation pour la vision par ordinateur en temps réel. La partie du projet consacrée à la détection des visages a été réalisée à l'aide d'une bibliothèque OpenCV pour python. La raison en était que la plupart des API Face ne permettaient la détection que sur les images, alors que le projet devait faire en sorte que la détection des visages soit effectuée sur une séquence vidéo en direct afin d'accélérer le processus de vérification de la présence des personnes devant la caméra.

Au début ça nous apparait que cette bibliothèque est suffisamment souple pour notre projet, vu qu'elle permet de détecter les visages en temps réel et les surligner en traçant

un rectangle autour des visages des personnes qui sont devant la caméra, mais les classificateurs que nous avons pu trouver disponibles ne puissent pas toujours identifier les bonnes personnes "on aura des fausses décision pour la reconnaissance faciale" alors nous avons dû chercher une autre alternatives qui est la bibliothèque BOTO3 de AMAZON AWS.

La figure suivante montre le script de détection des visages en utilisant opencv.

```
import requests as rq
import numpy as np
import cv2 as cv

url = "http://10.100.20.250:8080/shot.jpg"
detector = cv.CascadeClassifier('haarcascade_frontalface_default.xml')
while True:
        pic_resp = rq.get(url)
        pic_array = np.array(bytearray(pic_resp.content),dtype=np.uint8)
        img = cv.imdecode(pic_array, -1)
#define the screen resulation
        img_scaled = cv.resize(img,None,fx= 0.4,fy= 0.4)
        gray = cv.cvtColor(img_scaled, cv.COLOR_BGR2GRAY)
        faces = detector.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)

        for (x,y,z,w) in faces:
                print(x,y,z,w)
                pic_gray = gray [y:y+w , x:x+z]
                pic_color = img_scaled[y:y+w, x:x+z]
                colorRectangle = (141, 250, 205)
                cv.rectangle(img_scaled, (x,y) , (x+z,y+w) ,colorRectangle, 3)
        cv.imshow("camera",img_scaled)
        if cv.waitKey(100) & 0xFF == ord('x'):
                break
```

La figure suivante montre la partie de l'apprentissage pour faire la reconnaissance faciale en utilisant opencv.

```
model = cv.face.LBPHFaceRecognizer_create()

for root, dirs, files in os.walk(dirContent):
        for file in files:
                if file.endswith("png") or file.endswith("jpg") or file.endswith("jpeg"):
                        #get the path of each picture
                        imgPath=os.path.join(root,file)
                        #get to name of the person (diroctory) of each picture
                        dirName= os.path.basename(os.path.dirname(imgPath))
                        lowerDirName = os.path.basename(os.path.dirname(imgPath)).replace(" ","-").lower()  #
                        """
                                treat data, use the path  to get the image, after reading  the image  we use
                                the numpy lib to change or read this image as an array to start the trainner,
                                because we have  to use  informations of  images as numbers to learn about it
                        """
                        pillowImage = Image.open(imgPath).convert("L") #covert to grayScale  before (mode = RGB) now (mode
                        imgTOarray = np.array(pillowImage, "uint8")
                        faces = detector.detectMultiScale(imgTOarray, scaleFactor=1.5, minNeighbors=5)
                        if not lowerDirName in label_ids :
                                label_ids[lowerDirName]= cpt
                                cpt =cpt+1
                        id_ = label_ids[lowerDirName]
                        for(x,y,z,w) in faces:
                                #print(x,y,z,w)
                                faceData = imgTOarray[y:y+w ,x:x+z]
                                faceLabel = id_
                                xTraining.append(faceData)
                                yTraining.append(faceLabel)
with open("labels.pickle",'wb') as f :
        pickle.dump(label_ids, f)
model.train(xTraining,np.array(yTraining))
model.save("trainner.yml")
```

5.1.6 Reconnaissance Faciale en utilisant Boto3

Boto3 est le kit de développement logiciel Amazon Web Services (AWS) pour Python. Il permet aux développeurs Python de créer, configurer et gérer des services AWS, tels que EC2 et S3.

Boto3 fournit une API facile à utiliser, orientée objet, ainsi qu'un accès de bas niveau aux services AWS. Il fournit des fonctions pour le traitement des images "la classification, comparaison."

Après avoir utilisé boto3, nous avons pu avoir un système qui peut identifier les utilisateurs d'une manière fiables, mais nous avons constaté à la fin que c'est payé "nous avions juste une période d'essai pour l'utilisation des services AWS".

Après quelque recherche et documentations, nous avons trouvé un package de python gratuit qui s'appelle face_recognition pour le traitement d'images, nous allons expliquer par la suite l'implémentation de notre système en utilisant cette bibliothèque..
Le script suivant montre l'utilisation du services AWS boto3 pour la comparaison des images

```python
import boto3

#The two faces we're gonna compare
inputFace = ""
targetFace = ""
#config the region -in our account id use us-east-2
boto3.setup_default_session(region_name='us-east-2')
#use The client rekognition for face comparison
client = boto3.client('',aws_access_key_id='',aws_secret_access_key= '')
#picture to base64, because we gonna use local files
with open(inputFace, "rb") as image_file:
     input_Bytes = image_file.read()
with open(targetFace, "rb") as image_file:
     target_Bytes = image_file.read()
#lunch the function of face comparison
AWSresponse = client.compare_faces(
        SourceImage={'Bytes': input_Bytes},
        TargetImage={'Bytes': target_Bytes})
faceMatches_ =AWSresponse['FaceMatches']
faceUnMatches_ = AWSresponse['UnmatchedFaces']
#display result of comparison between the two faces
if len(faceUnMatches_)==0:
        faceInformations = faceMatches_
        for information in faceInformations:
                faceSimilarity = information
                confidence = information['Face']
                print("Recognized with {}""%"" confidence ".format(confidence['Confidence']))
else :
        faceInformations = faceUnMatches_
        for information in faceInformations:
                confidence = information['Confidence']
                print("Non recognized with {}""%"" confidence ".format(confidence))

```

5.1.7 Reconnaissance Faciale en utilisant face_recognition

Face_recogniton est une bibliothèque python, il est utilisé pour reconnaître et manipuler des visages à partir de Python ou de la ligne de commande avec d'une manière très simple et efficace.

L'idée dans cette partie est d'utiliser notre caméra pour récupérer une vidéo au temps réel (ensemble des images) et à chaque fois on récupérer cet ensemble des images pour le comparer avec celles de notre base de données "cette bibliothèque permis de faire une comparaison entre deux bases de données", après on récupère les personnes reconnus.

Le commande qui permit de comparer deux bases de données est le suivant :
face_recognition ./dataBase_i_know/ ./database_iwant_to_known/

Les scripts suivant de notre système en utilisant cette bibliothèque

Détection des visages :

```
                    # ***   Detection  ***
#The current Path
currentPath = os.path.dirname(os.path.abspath(__file__))
#comming dir contains pictures taken for people who're in front of the camera
dirPath = os.path.join(currentPath, "comming")
dirPathSavingFaces = os.path.join(currentPath, "toTreat")
filelist = [ f for f in os.listdir(dirPathSavingFaces) if f.endswith("png") or f.endswith("jpg") or f.endswith("jpeg")  ]
detected = False
for f in filelist:
    os.remove(os.path.join(dirPathSavingFaces, f))
for root, dirs, files in os.walk(dirPath):
        for file in files:
                if file.endswith("png") or file.endswith("jpg") or file.endswith("jpeg"):
                        #print(os.path.join(root,file))
                        imgPath =os.path.join(root,file)
                        image = face_recognition.load_image_file(imgPath)
                        face_locations = face_recognition.face_locations(image)
                        if len(face_locations)!=0 and detected == False:
                                detected = True
                        for face_location in face_locations:
                            # Print the location of each face in this image
                            top, right, bottom, left = face_location
                            getFaceLocation = image[top-20:bottom+20, left-20:right+20]
                            facePILL = Image.fromarray(getFaceLocation)
                            index = random.randint(1,19061995)
                            rangeLetters='ADLANEKADRI19061995'
                            pictSavedNames = ''.join((random.choice(rangeLetters) for cpt in range(len(str(index))))+ "." +'
                            saveRoot = os.path.join(currentPath, "toTreat")
                            savePath =os.path.join(saveRoot,pictSavedNames)
                            facePILL.save(savePath)
```

Reconnaissance de visage:

```
if detected :
        #The current Path
        currentPath = os.path.dirname(os.path.abspath(__file__))
        #connu dir contains pictures taken for known people
        dirPathConnu = os.path.join(currentPath, "connu")
        dataBase = []
        names= []
        for root, dirs, files in os.walk(dirPathConnu):
                for file in files:
                        if file.endswith("png") or file.endswith("jpg") or file.endswith("jpeg"):
                                imgPath =os.path.join(root,file)
                                imageKnown = face_recognition.load_image_file(imgPath)
                                #print(len(face_recognition.face_encodings(imageKnown)))
                                fileEncoded = face_recognition.face_encodings(imageKnown)[0]
                                dataBase.append(fileEncoded)
                                names.append(os.path.splitext(file)[0])
        dirPath = os.path.join(currentPath, "toTreat")
        dataBase2 = []
        for root, dirs, files in os.walk(dirPath):
                for file in files:
                        if file.endswith("png") or file.endswith("jpg") or file.endswith("jpeg"):
                                imgPath =os.path.join(root,file)
                                imageKnown = face_recognition.load_image_file(imgPath)
                                fileEncoded = face_recognition.face_encodings(imageKnown)[0]
                                dataBase2.append(fileEncoded)
        recognized= []
        for pictEncoded in dataBase2:
                response = face_recognition.compare_faces(dataBase, pictEncoded)
                listResult= pd.Series(response)
                #print(s[s].index.values)
                for val in listResult[listResult].index.values :
                        if names[val] not in recognized :
                                recognized.append(names[val])
```

Décision du système:

```
                        # Decision of recognition
        print(recognized)
        if(len(recognized)==0):
                #copy face to other path 'unknown'
                currentPath = os.path.dirname(os.path.abspath(__file__))
                dirPath = os.path.join(currentPath, "toTreat")
                dirPathSaving = os.path.join(currentPath, "inconnu")
                filelist = [ f for f in os.listdir(dirPath) if f.endswith("png") or f.endswith("jpg") or f.endswith("j
                for f in filelist :
                        source =os.path.join(dirPath,f)
                        img = Image.open(source)
                        imG = img.resize((width, height), Image.ANTIALIAS)
                        rangeLetters='UNKNOWNunknownadlanekadriADLANEKADRI19061995'
                        newNames = ''.join((random.choice(rangeLetters) for cpt in range(len(str(index)))))+ "." +'jpg
                        imG.save(os.path.join(dirPathSaving,newNames))
                        # adjust width and height to your needs
                # insert in database
                insert(0,"unknown")
                valeur = 0
                subject = "WARNING" msg = "Attention! \n il y a un inconnu devant la porte"
        else:
                subject = "IFORMATION"
                valeur =1
                a = "these person have returned: \n"
                b = "this person has returned: \n"
                msg = "We could detect that %s "%(a if len(recognized)>1 else b)
                for personName in recognized:
                        insert(1,personName)
                        msg = msg + "- " + personName + "\n"
        sms.sendingTEXT(subject, msg)
        updateDoor(valeur)
else :
        print("any person detected")
```

5.1.8 Système de notification

Nous avons utilisé smtplib, ce module définit un objet de session client SMTP pouvant être utilisé pour envoyer des messages à tout ordinateur Internet doté d'un démon écouteur SMTP ou ESMTP. nous avons utilisé ce module de cette maniére :

```python
import smtplib
from getpass import getpass
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from email.mime.base import MIMEBase

def sendingTEXT(subject, msg):
    try:
        server = smtplib.SMTP('smtp.gmail.com:587')
        server.ehlo()
        server.starttls()
        EmailSender= "smartkey19061995@gmail.com"
        Password="sma8888"
        EmailReceiver="adlan68@live.fr"
        server.login(EmailSender, Password)
        message = 'Subject: {}\n\n{}'.format(subject, msg)
        server.sendmail(EmailSender, EmailReceiver, message)
        server.quit()
        print("great: Email sent!")
    except:
        print("failed :/")


```

5.2 développement de l'application web

Dans cette partie, nous allons vous expliquer comment nous avons pu mettre en place, l'application web côté client en libre accès avec l'url: http://luckyraheriniaina.ddns.net/smartdoor/ L'application permet au client de:

- Valider une personne reconnu par le système
- Refuser une personne
- Ajouter des nouvelles personnes valide par le système
- Gérer ces informations personnels
- Se connecter / Se déconnecter

5.2.1 Etape 1: Mise en place d'un serveur web

Pour créer un serveur web de type LAMP, nous nous sommes muni d'un rasberry pi 3 model b+.

Nous avons configuré celui-ci de sorte qu'il s'exécute avec la dernière image "Raspbian" existant avec l'interface graphique.

Nous avons ensuite installé Apache 2 qui permet à la machine d'analyser les requêtes d'un utilisateur sous forme http, et de retourner le fichier correspondant à la requête dans le cas inverse, une erreur si le fichier n'est pas trouvé, ou la requête mal formulée.

Notre site étant dynamique, l'outil MySQL est essentiel pour la gestion de la base donnée. Nous avons installé PhpMyAdmin pour administrer cette base de données.

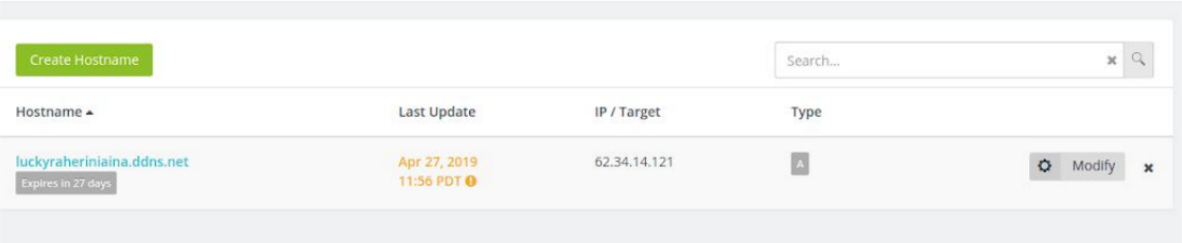
Et enfin le langage PHP, le langage qui permet de mettre en place des sites dynamiques.

5.3 Etape 2: Création d'un nom de domaine

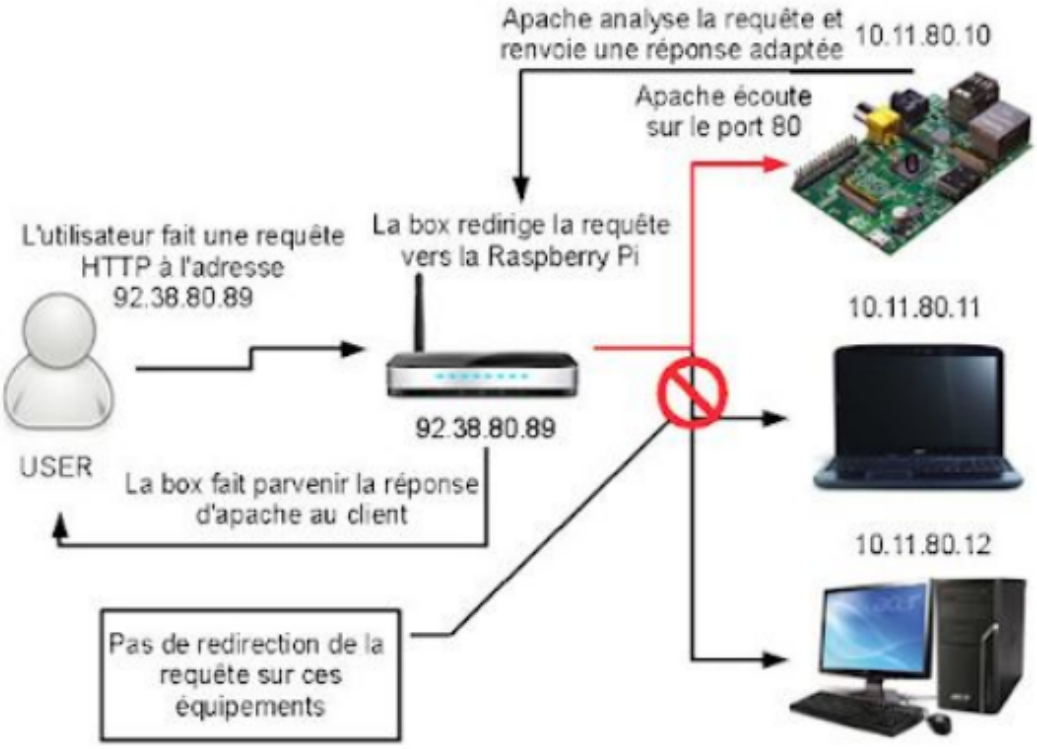
Pour la création d'un nom de domaine, nous nous somme rendue sur le site www.noip.com qui propose la création au choix de nom domaine non-payant.

Ainsi nous avons choisis luckyraheriniaina.ddns.net.

Quand le nom de domaine est libre et valider par le site, noip associe notre adresse IP au nom de domaine choisis, comme ci-dessous.



5.3.1 Etape 3: Configuration de la box



Il ne reste plus qu'à se rendre dans la configuration de le box à la maison, suivant l'opérateur l'adresse peut varier (192.168.0.0 / 192.168.0.1 / 192.168.0.255).

Nous allons configurer la redirection de port dans la rubrique NAT, il faut l'activer si cela est désactivé.

Ensuite, nous créons le port 80 (HTML) en protocole TCP, qu'on va rediriger vers le Rasberry qui aura une adresse local qui lui a été associe dès lors que nous l'avons branché sur le réseau local, comme illustre sur l'image dessus/dessous

Il nous reste plus qu'à nous rendre dans DynDNS, qui va nous permettre de donner un nom à notre adresse IP, cela évite de saisir l'ip en entier a chaque fois que nous souhaitons accéder au site.

Pour se faire nous activons DynDNS, si cela n'est pas fait.

Ensuite, nous reprenons le nom de domaine que nous avons créé sur le site noip, et nous Saisissons les informations comme sur l'image ci-dessous.

Ainsi nous venons de mettre en ligne notre serveur web Raspbian. Nous pouvons maintenant procéder au développement du site.

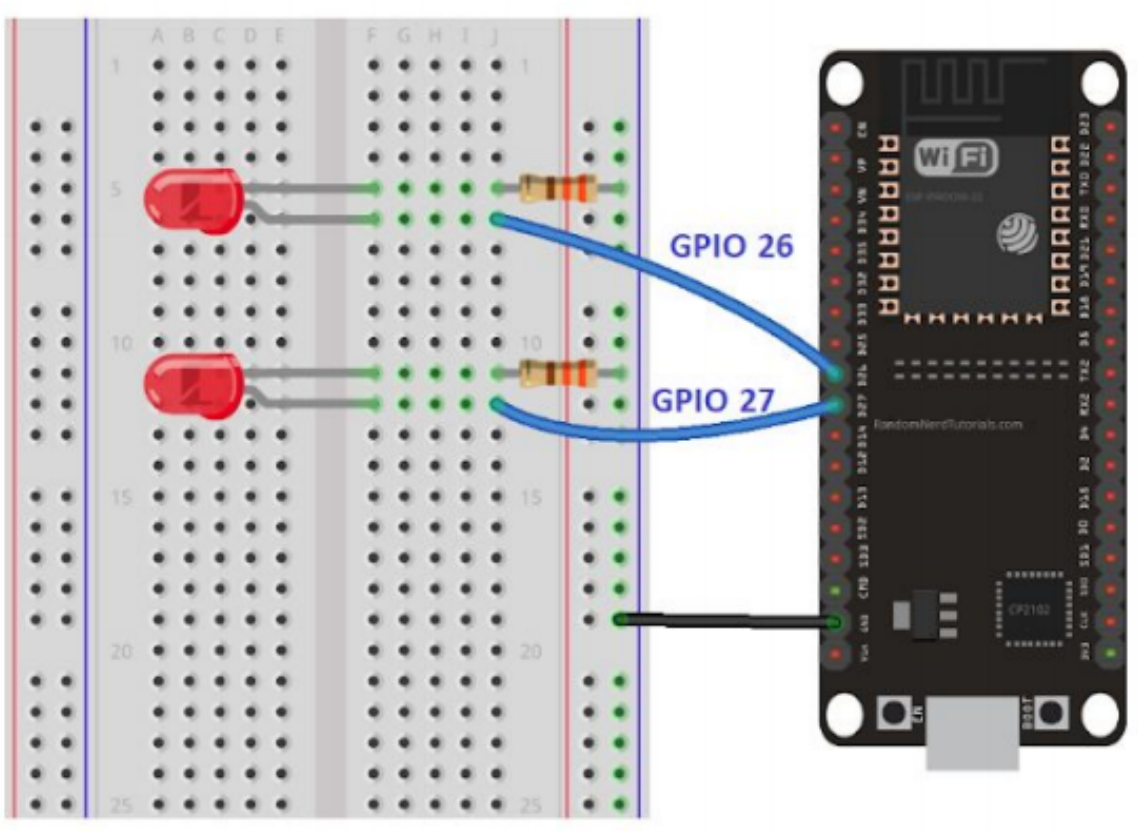
22

5.3.2 Etape 4: Développement du site

Dans cette mini partie, nous allons montrer en détail les différents fonctionnalités de l'application web accessible sur les smartphone, ordinateur et tablette.

Nous avons utiliser les framework bootstrap version 4 pour rendre le site responsive et ...

5.3.3 Etape 5: Communication entre ESP32 - Rasberry Pi



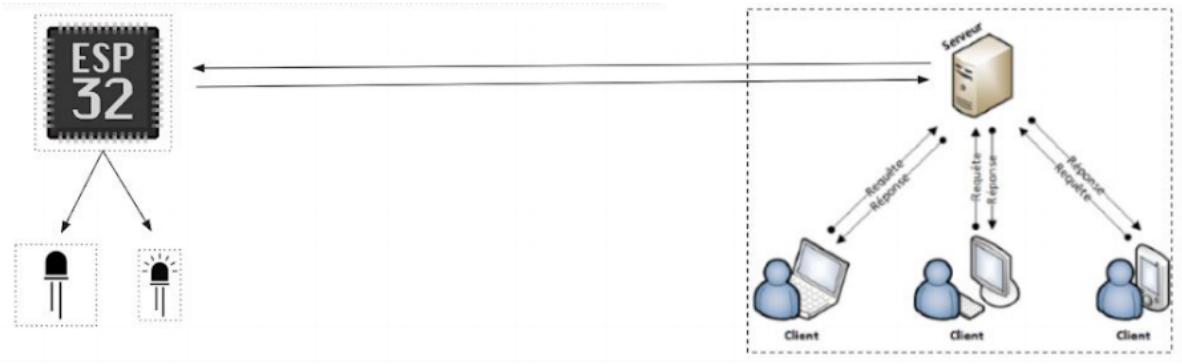
Dans cette partie nous allons vous expliquer comment nous avons fait pour gérer les Leds a distances depuis la base de donnée.

Dans un premier temps nous avons fait le câblage comme dans le schéma ci-dessus, nous avons utiliser en plus de l'ESP32, trois câbles mâle-mâle, deux resistant et deux leds, rouge et bleu.

Pour le développement nous avons utiliser l'IDE Arduino. Les tests ont d'abord été

réalisée en local avant d'être copiés sur le raspberry.

Pour le Rasberry nous avons procédé à l'ouverture du port 3306 dans la box, afin de permettre l'accès distant à un serveur Mysql



Le script interroge en continu la table "open_door" toutes les secondes et d"s qu'un visage est reconnu, un des deux leds reste allumé pendant 6 secondes, suivant les circonstance nous avons les résultats suivant.

	Valeurs	Résultats
Aucun visage détecté	0	Rien
Une personne autorisée est détectée	1	Lumière bleu s'allume
Une personne personne non autorisée est détecté	2	Lumière rouge s'allume

6 Résumé des tâches réalisées pour chaque membre

- MTIBAA Ameni:

  – Résumé:

    Dans ma partie, j'étais amenée à faire fonctionner l'ESP32 WROVER avec la caméra OV2640. Je devrais intégrer dans l'application d'André un streaming de vidéo et fournir à Adlane des photos prise avec la caméra pour qu'il puisse faire la détection et reconnaissance de visage . Pour le rapport, j'ai décrit ma partie ainsi que la promotion du produit

- Difficultés:
  Malgré l'aspect simple de ma tâche, elle cache derrière des différentes difficultés :

  * C'est ma première expérience avec les microcontrôleurs et l'embarqué en générale.
  * Depuis ma L1, je n'ai pas programmé en C.
  * j'étais amenée à travailler avec ESP WROVER et non pas ESP WROOM puisque c'est la mieux adaptée avec la caméra mais malheureusement elle posséde beaucoup moins de documentation.
  * j'ai acheté la caméra (ov2640) de Amazon en croyant que c'est la même caméra utilisée dans les tutos puisqu'elle porte le même nom. À ma grande surprise, le nom des PIN de la caméra acheté était différent de la vrai OV2640 donc j'ai passé du temps pour trouver la documentation de cette caméra. Ainsi, il s'est avéré qu'il manque un PIN très important (Celui de l'horloge : XCLK) qui est utilisé dans presque tous les exemples d'internet. Donc j'ai passé énormément du temps à essayer de trouver une autre solution pour la faire marcher (avec I2S Pour la séparation des données et des horloges) . Quand ça n'a pas marché, j'ai acheté une autre caméra et des fils.
    Finalement, c'était un problème de fils.

- Ce que j'ai appris :
  Que rien n'est impossible (fonctionner la caméra, or qu'il y a des gens qui ont eu le même problème et qui n'ont pas pu la faire fonctionner)

- Appréciation du cours :
  Le cours est vraiment intéressant, mais je pense qu'il doit se faire sur deux semestres au lieu d'un seul. Il nécessite énormément de temps (personnellement, j'ai dû sacrifier mes week-end pour pouvoir y travailler).

• KADRI Adlane:

- résumé:
  durant ce projet, j'étais amené à développer un système de reconnaissance faciale, proposer une architecture de travail pour faciliter a la fin le regroupement de toutes les taches.
  J'ai fait la liaison entre toute les taches (l'application web fait par andré et le streaming de vidéo fait par ameni), et pour le rapport, j'ai rédigé toute la partie du système de reconnaissance faciale ainsi l'organisation du rapport en latex.

- Difficultés:
  Les difficultés que j'ai rencontrées durant ce projet, c'étais comment choisir les technologies de développement.
  Et vu que ma partie était presque "backend" le développement du système alors la plupart des problèmes étaient techniques (le traitement des images

surtout comme notre caméra donne des images flou, j'ai eu des problèmes pour l'encodage avant la reconnaissance).
Aussi, j'ai eu un problème à la fin pour relier tout les taches pour avoir un seul système qui fonctionne d'une manière fiable.

- RAHERINIAINA Lucky:

  - Tâche d'effectuer:
    Dans la répartition des tâches, l'équipe s'est divisée en deux sous-groupes, André et moi étions responsable du développement de l'application web, Ameni et Adlane de la reconnaissance faciale.
    Dans cette partie, je me suis occupé de l'achat du rasberry pi 3 model b+, après l'avoir configuré, j'ai mis en place un serveur LAMP (Linux, Apache2, MySQL, PHP).
    Comme nous somme limitée au niveau budget, je me suis rendu sur le site noip.com afin de créer un nom de domaine gratuitement.
    J'ai par la suite configurer ma box afin de rediriger les requetes html vers le rasberry, en plus de cela j'ai configurer DynDNS afin d'associer le nom de domaine a mon adresse IP.
    Les ports FTP, SSH, MySQL on également été redirigé vers le Rasberry pour permettre à l'équipe de communiquer avec la Rasberry à distance.
    J'ai collaboré avec André sur le développement du site et développement sur Arduino du script qui permet de gérer les Leds qui s'active lorsqu'une personne est reconnu.
    Sur le site j'ai développé la page addusers, qui permet d'ajouter de nouvelle personne reconnu par le système.
    J'ai contribué dans la rédaction du cahier des charges, de l'étude de l'états de l'art, dans la réalisation de la vidéo et du rapport final.

  - Difficultés:
    J'ai eu des problèmes lorsque j'ai voulu brancher l'ESP32 sur le logiciel Arduino, le port n'était pas accessible.
    Pour régler le problème j'ai changer les privilège du port en question dans le fichier système de linux.
    J'ai également eu des problèmes de bibliothèque qui manquait pour le développement Arduino avec l'ESP32, pour fair face a ce probleme j'ai tout réinstaller.
    Une des difficultés majeurs qui nous a retardé a été du côté organisationnel, en effet il n'a pas toujours été évident de se retrouver tous ensemble pour travailler le projet, certain personne était en stage, d'autre avait un travail à côté en plus des autres projets, or le projet nécessite énormément de temps de documentation, de réflexion et d'échange.

  - Avie personnelle sur le module:
    Ce projet a été d'une très grande richesse tant dans le plan personnel que

professionnel.
Dans un premier temps, cela m'a permis d'apprendre à utiliser des outils de gestion de projet tel que GitHub, à mettre en place un serveur web.
J'ai pu mettre en avant les acquis aux cours des années antérieures.
J'ai également une meilleure idée de ce qu'est l'IoT dans la vie courante, des étapes à effectuer lors de l'élaboration d'un produit tel que la rédaction du cahier de charges, l'état de l'art et de la partie conception.
Je souhaite remercier Monsieur Osmani de nous avoir transmis de nouvelle connaissance, de nous avoir fait partager son expérience dans le domaine de l'auto-entreprenariat et de nous avoir sensibilisés au potentiel que peut nous apporter l'IoT dans la marche avenir.
Je remercie également Monsieur Hamidi, pour nous avoir encadré tout au long du semestre notamment pour les problèmes d'ordre technique.

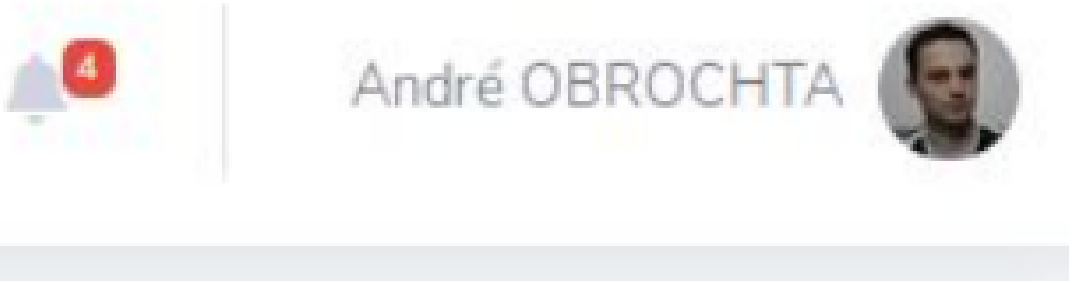
- OBROCHTA Andre:

  - Réalisation du site internet:
    Une des parties que j'ai réalisé fut le site internet permettant d'administrer SmartDoor, pour cela avec l'aide de Lucky nous avons utilisé les outils suivants:

    * Bootstrap
    * Html
    * Php
    * SB Admin 2 (une template afin de gagner du temps dans la conception du site internet)
    * MariaDB

    La fonction principal du site est de pouvoir voir, modifier, ajouter et accepter des personnes autorisés par le système.
    Il a fallu dans un premier temps concevoir une base de donnée, j'ai donc dû analyser tous les besoins du futur site.
    Une fois la base de donnée créé, je me suis occupé de la page de connexion et une zone d'administration, tout cela relié à la base de donnée. Il a fallu à l'aide de PHP interdire la zone d'administration aux utilisateurs non connectés.
    Je me suis occupé de faire l'interaction pour accepter ou refuser des personnes. Pour réaliser cela, le système est assez simple, PHP cherche les images dans les dossiers et chaque dossier correspond à la situation de la personne, on a par exemple un dossier pour les visages acceptés, lorsque l'utilisateur veut voir les utilisateurs acceptés, PHP va simplement afficher toutes les images qui se trouve dans le dossier. Si l'utilisateur change d'avis et souhaite refuser un visage accepté, PHP va simplement déplacé l'image vers le dossier personne refusé.
    Pour des raisons de sécurité et des raisons pratiques, il m'a semblé utile d'ajouter des logs pour sauvegarder chaque interaction avec le site web afin

que par exemple voir s'il y a eu intrusion dans le système ou voir quel administrateur a accepté telle personne etc..
À chaque fois que des logs sont insérés dans la base de données, l'utilisateur reçoit une notification.



– Réalisation de l'algorithme pour l'esp32 afin d'effectuer l'ouverture de porte :
Le but de cet algorithme est d'ouvrir la porte lorsque la reconnaissance faciale détecte un visage qui est accepté par le système. Lorsqu'un visage est accepté, le programme va modifier une valeur dans la base de donnée à la table 'open_door', notre esp32 lui va lire en continue la valeur à cette table, si celle-ci est égale à 0, cela signifie qu'aucun visage est reconnu, si par contre la valeur est à 1, un visage est reconnu et 2 signifie que le visage est reconnu mais pas accepter par le système, on ouvre donc la porte et on attend 6 secondes, le temps que la personne ouvre la porte, avant de la fermer puis on recommence la boucle. Pour nous simplifier la tâche, nous avons choisi de représenter l'ouverture et fermeture de porte par l'allumage d'une led rouge et bleu.

– Difficultés:
Malgré les différentes étapes de conceptions réalisés, il y a toujours des différences entre ce que nous pensons être là bonne solution et celle qui au final semble fonctionner. Cela est sûrement dû à notre manque d'expérience à réaliser un projet de cette manière là. Ceci a pour effet de devoir constamment adapter notre projet avec l'autre équipe, et cela a posé des problèmes car il a fallu changer plus d'une fois notre conception et cela c'est aussi traduit par une perte de temps non négligeable car à chaque changement de conception, il a fallu modifier voir refaire le code.
Il a été difficile de connecter l'esp32 à une base de donnée, en effet il a fallu rajouter une bibliothèque qui nous semblait au départ ne pas bien fonctionné. En effet, nous pensions que peut être la bibliothèque était obsolète avec les nouvelles versions de MariaDB.

Après un certain temps nous nous sommes rendu compte que cela venait du fait que notre base de donnée n'était pas distante et donc inaccessible hors

réseau local.

Une autre difficulté rencontrée fut le fait que nous avions seulement un exemplaire des objets utilisés. Cela m'a été problématique puisqu'il est par conséquent impossible de travailler chez soi en s'assurant que notre programme fonctionne comme souhaité avec les différents objets.

Et cela a posé aussi un problème lorsqu'on travaillait ensemble car pour tester l'objet en question, il fallait attendre que la personne qui utilise l'objet finisse.
Pour faire de la reconnaissance faciale, le langage qui est de loin le plus utilisé est le python, n'ayant jamais fais de python auparavant, il a fallu que j'apprenne à utiliser ce langage.
Ceci fut la même chose avec Github car il s'agissait de la première fois que j'utilisais cet outil pour travailler en groupe.

7 Conclusion générale

Notre projet nommé "smart door" est, comme son nom l'indique, un système permettant la détection d'une personne devant une porte, celle-ci s'ouvre une fois que le visage est reconnu et accepté par le système. Ceci est relié à une application web/mobile qui permet de voir en temps réel le trafic ainsi que les gens qui sont reconnus et ceux qui ne sont pas reconnus.
Ce projet a relié plusieurs domaines informatique : l'embarqué, la machine learning et le web.
L'un des enjeux était de relier les différents environnements de travail ensemble. Ceci a été possible grâce à l'utilisation de Github. Ce projet nous a permis de comprendre l'importance d'utiliser un git pour un travail en groupe et comprenons mieux maintenant pourquoi elle est autant demandée au sein des entreprises.
Ce projet a été aussi bénéfique, car chacun d'entre nous a développé des compétences spécifiques à la partie qui était à sa charge.
Nous avons réussi à rassembler les parties et les intégrer et donc achever les objectifs qu'on a fixés au début : développer un système de reconnaissance faciale du bout en bout, de la prise d'image et la reconnaissance à la gestion et supervision.

29