Table des matières

[Table des figures iv](#_Toc140411232)

[Liste des tableaux vi](#_Toc140411233)

[Introduction générale 1](#_Toc140411234)

[CHAPITRE 1 : La reconnaissance faciale 2](#_Toc140411235)

[1.2 La reconnaissance faciale 2](#_Toc140411236)

[1.3 Domaines de la Reconnaissance Faciale 2](#_Toc140411237)

[1.4 Historique 3](#_Toc140411238)

[1.5 Les techniques de détection et de reconnaissance faciale 4](#_Toc140411239)

[1.6 Problématique 5](#_Toc140411240)

[1.7 Conclusion 7](#_Toc140411241)

[CHAPITRE 2 : lancement du projet 8](#_Toc140411242)

[I. CAHIER DE CHARGE DU PROJET 8](#_Toc140411243)

[1. Besoins fonctionnels 8](#_Toc140411244)

[2. Besoins non fonctionnels 8](#_Toc140411245)

[4. Diagramme de contexte 9](#_Toc140411246)

[II. METHODE DE DEVELOPPEMENT 10](#_Toc140411247)

[1. Méthode adoptée 10](#_Toc140411248)

[2. Constitution de l’´équipe 10](#_Toc140411249)

[3. Etude comparative 10](#_Toc140411250)

[4. Architecture globale du logiciel 11](#_Toc140411251)

[5. Product Backlog de Delta-face 12](#_Toc140411252)

[6. Diagramme des cas d’utilisation global 13](#_Toc140411253)

[7. Branche Technique 16](#_Toc140411254)

[A. Architecture 16](#_Toc140411255)

[B. Diagramme de séquence 17](#_Toc140411256)

[1. Notion de scenario 17](#_Toc140411257)

[2-Enumération des diagrammes de séquences 18](#_Toc140411258)

[a) Digramme de séquence détaille du cas « s’authentifier » 18](#_Toc140411259)

[a) Diagramme Séquence <capture photo >> pour autorisation en salle 19](#_Toc140411260)

[b) Diagramme d’etat de la <<camera>> 20](#_Toc140411261)

[c) Diagramme d’activité <<d’authentification>> pour administrateur 21](#_Toc140411262)

[d) Diagramme d’activité <<d’authentification>> pour chef-salle 22](#_Toc140411263)

[e) Diagramme d’activité de la reconnaissance 23](#_Toc140411264)

[f) Diagramme de classes du systeme 24](#_Toc140411265)

[8. Réalisation 25](#_Toc140411266)

[a) Mise en place des Sprint 26](#_Toc140411267)

[b) Séance du Sprint planning 26](#_Toc140411268)

[c) Séance du Sprint planning 1 26](#_Toc140411269)

[4. Sprint backlog 1 estimé 27](#_Toc140411270)

[5. Demarrage du sprint 28](#_Toc140411271)

[6. Apprentissage de la Camera 28](#_Toc140411272)

[9. Codage 29](#_Toc140411273)

# Introduction générale

La reconnaissance de visages est une technologie biométrique en vogue, elle est très utilisée dans les applications d’authentification, de contrôle d’accès et de vidéo de surveillance, on trouve plusieurs méthodes globales, locales et hybrides de reconnaissance de visages.

Ces derniers temps et avec l'avancement de la technologie, les mots de passe et les clés qu'on utilise dans tous les domaines de sécurité et de contrôle d’accès sont devenus facilement falsifiables et franchissables. C’est pour ça, que les chercheurs de différents domaines ont orientés leurs travaux sur des clés et des mots de passe impossible à falsifier, surs et surtout efficaces. Et c’est ainsi que fut inventée la biométrie qui est devenue à la mode dans les domaines qui requiert un niveau élevé de sécurité et de contrôle.

Et parmi toutes les technologies biométriques qui existent, la reconnaissance faciale est l'une des technologies les plus utilisées et les plus adaptées car elle nous permet d'exploiter de nombreuses informations relatives à une personne.

Malgré l'énorme progression en informatique surtout dans le domaine du traitement d’images on trouve toujours des difficultés concernant les systèmes appliquant les techniques de reconnaissance faciale, c'est pour cela qu’elle a engendré un grand nombre de travaux de recherche au cours des 50 dernières années et ceci dans différents domaines tels que l’identification de modèles, réseaux de neurones, vision par ordinateur, infographie …

Dans ce PFE, on s’est justement intéressé à cette thématique fort utile et très demandée dans le domaine de la sécurité des réseaux, la sécurité nationale et même internationale. On s’est proposé d’implémenter une application de reconnaissance faciale utilisant la technique la plus performante de l’état de l’art.

# CHAPITRE 1 : La reconnaissance faciale

**1.1 Introduction**

Durant la vie quotidienne chacun de [nous identifie](http://www.rapport-gratuit.com/) tout au long de la journée différents visages. Ainsi lorsque nous rencontrons une personne, notre cerveau va chercher dans notre mémoire et vérifie si cette personne est répertoriée ou non, c'est une tâche aisée pour les humains. En est-il de même pour une machine ?

Dans ce premier chapitre, nous allons justement présenter les grandes lignes de notre travail, nous allons ainsi expliquer des notions générales sur la reconnaissance faciale, le fonctionnement d’un système de reconnaissance faciale, On va aussi aborder quelques techniques utilisées, des exemples et les domaines d’application.

.

## 1.2 La reconnaissance faciale

La reconnaissance faciale est un procédé biométrique au même titre que la reconnaissance d’empreintes digitales, d’iris, ou vocale. Cela consiste donc à déterminer l’identité d’une personne. Le système de reconnaissance faciale est une application logicielle visant à reconnaître une personne grâce à son visage de manière automatique.

A l’aide d’algorithmes, cette application analyse toutes les caractéristiques faciales telles que l’écartement des yeux, des arêtes du nez, des commissures des lèvres, des oreilles, du menton, à partir d’une image de son visage qui peut provenir à la fois d’une photo ou d’une vidéo. Donc de nombreux formats d’image peuvent être utilisés qu’ils soient statiques (JPG, bmp, png, GIF …) ou en mouvement (MPEG4 Part 2) [1]

Plusieurs méthodes de reconnaissance de visages ont été proposées durant ces 30 dernières années, suivant deux grands axes : la reconnaissance à partir d’images fixes et la reconnaissance à partir de séquence d’images (vidéo). La reconnaissance de visages basée sur la vidéo est préférable à celle basée sur des images fixes, puisque l’utilisation simultanée des informations temporelles et spatiales aide dans la reconnaissance.

Dans notre projet, on s’est focalisé sur la reconnaissance basée sur la vidéo, puisqu’elle est plus demandée et plus pertinente [2].

## 1.3 Domaines de la Reconnaissance Faciale

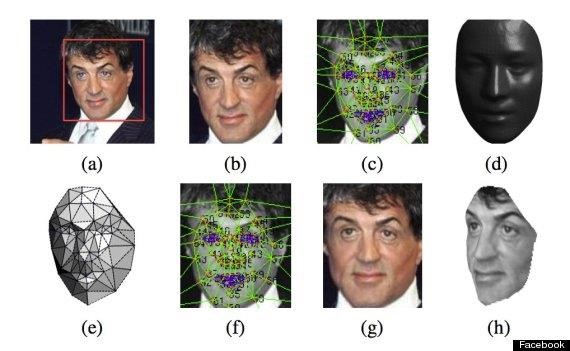
Aujourd’hui la reconnaissance faciale est utilisée principalement pour des raisons sécuritaires. Elle peut être utilisée à des fins très diverses. Par exemple, l'authentification, le contrôle d’accès (autorisation) et la vidéo de surveillance.

Un bon exemple de l'usage des applications d’identification, est le nouveau tunnel qui sera installé d'ici la fin de l'été, situé à Dubaï premier de ce type dans le monde. Il s’agit d’un système biométrique qui permet aux passagers d’être identifiés en traversant un tunnel dans le but d’augmenter l’efficacité des points de contrôle de sécurité. Ils n’ont même pas besoin de montrer leur passeport. L’outil fonctionne grâce à la reconnaissance de l’iris et du visage. La procédure dure environ 15 secondes. [3]

La reconnaissance faciale est aussi utilisée dans les Applications militaires. Un bon exemple de ce domaine est l'utilisation des lunettes de style « Robocop » munies d’une petite caméra d’une portée de 12 milles (19,3 km) par la marine américaine, la caméra peut aussi faire partie de l'optique d'un soldat sur son arme. Grâce à cet équipement, les soldats peuvent identifier des ennemis en quelques secondes sur le terrain, et cela sans réseau à large bande. [4]

En revanche, on distingue un autre domaine d’application de ces systèmes qui est l'assistance à l'utilisateur. Les systèmes de reconnaissance faciale sont de plus en plus présents au quotidien. Ils sont par exemple utilisés sur les réseaux sociaux sur internet pour identifier quelqu'un sur une photo, sur les Smartphones pour les déverrouiller...

Facebook a la plus grande collection et exposition de photos au monde et a maintenant une chance d'unifier tous les visages. Oui, même les personnes qui ne sont pas identifiées. Facebook a annoncé qu'il a développé un programme appelé DeepFace, qui selon les chercheurs, peut déterminer si deux visages appartiennent à la même personne avec une précision de 98%. Selon Facebook, dans 97,53% des cas, les photos à différents âges de même personnes ont la même réponse au test, comme le prouve la procédure de DeepFace pour la reconnaissance du célèbre acteur Sylvester Stallone. [5]



***Figure 1 La procédure de DeepFace pour la reconnaissance***

La nouveauté dans la reconnaissance faciale arrive grâce au développement de nouvelles caméras de type 3D. Ces caméras obtiennent de meilleurs résultats que les caméras classiques, parce qu’elles acquièrent une image tridimensionnelle de chaque visage (perspectives)pour identifier une personne lorsqu’elle passe par le portail d’authentification.

## 1.4 Historique

La reconnaissance faciale est une technique biométrique relativement récente. Si l’empreinte digitale est la technique biométrique la plus ancienne inventée en 1903 pour rechercher les criminels, la reconnaissance des visages a été développé par "Benton et Van Allen" en 1968 pour évaluer la capacité d'identification des visages non familiers. Il ne s'agit pas d'un test de reconnaissance ménisque de visages familiers ou non familiers, mais d'une épreuve consistant à apparier des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale [6].

L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle depuis le milieu des années 90, avec l’utilisation efficace de nouvelles technologies, notamment l’ordinateur et sa capacité de traitement d'images. L’utilisation de ces techniques existe depuis qu’une machine est capable de comprendre ce qu’elle « voit » lorsqu’on la connecte à une ou plusieurs caméras, c’est à dire que les premiers essais datent du début des années 70 (Benton et Van Allen en 1968), et sont basés sur des méthodes à bases d’heuristiques, basés sur des attributs faciaux mesurables comme l’écartement des yeux, des sourcils, des lèvres, la position du menton, la forme, etc. Ces méthodes sont très peu robustes, car elles font de nombreuses suppositions en se plaçant dans des cas très simples (visage de face, bonnes conditions d’illuminations, etc. L’une des premières tentatives de reconnaissance de visage est faite par Takeo Kanade en 1973 lors de sa thèse de doctorat à l’Université de Kyoto. [7][8]

## 1.5 Les techniques de détection et de reconnaissance faciale

Il existe une multitude de techniques consacrées à la détection du visage. Mais pour la plupart il est d’intérêt que ces techniques se basent sur des éléments du visage qui sont le moins susceptibles aux changements :

Les grands traits supérieurs des orbites, les secteurs entourant les pommettes, les côtés de la bouche et d’autres caractéristiques similaires de façon à ignorer les changements comme la coupe de cheveux.

Les méthodes de base les plus utilisées sont :

* ***Le traitement automatique du visage***

C’est une technologie rudimentaire, elle caractérise les visages par des distances et des proportions entre des points particuliers comme les deux yeux, le nez, les coins de la bouche. Aussi éprouvé que les autres technologies, le traitement automatique du visage est le plus efficace dans des situations de capture d’image avec peu d’éclairage [9].

* ***Eigenface***

Les « Eigenfaces » furent le premier type de caractérisation utilisé avec succès dans des traitements faciaux tels que la détection et la reconnaissance du visage [10]

Elle utilise une représentation des éléments caractéristiques d’une image de visage à partir d’images modèles en niveau de gris. Des variantes d’Eigenface sont fréquemment utilisées comme base pour d’autres méthodes de reconnaissance [11].

* ***L'analyse des points particuliers***

Elle est la technique d’identification faciale la plus utilisée. Cette dernière se rapproche de Eigenface, mais elle est capable de s’adapter à des changements d’aspect facial (sourire, froncement des sourcils, ...). Les ingénieurs numériques l'utilisent souvent [12].

* ***LDA (Linear discriminant analysis) fisher***

Elle fait partie des techniques d’analyse discriminante prédictive. Il s’agit d’expliquer et de prédire l’appartenance d’un individu à une classe (groupe) prédéfinie à partir de ses caractéristiques mesurées à l’aide de variables prédictives [13].

* ***Filtre de Haar***

La détection du visage dans ce filtre est réalisée par un filtre multi-échelles de Haar. Les propriétés d'un visage sont décrites dans un fichier XML. Elles ne sont pas choisies au hasard et reposent sur un échantillon de quelques centaines d’images tests [14]. Nous reviendrons plus en détails sur cette méthode.

* ***Méthode LBP* (*Local Binary Patterns)***

Les LBP (Modèles binaires locaux) ont également été utilisés pour la détection du visage ou ce dernier est subdivisé en sous–régions carrées de taille égale sur lesquelles sont calculées les caractéristiques LBP. Les vecteurs obtenus sont ensuite concaténés pour obtenir le vecteur de caractéristiques final.

## 1.6 Problématique

Le problème de la reconnaissance des visages peut être formulé comme suit :

Étant donnée une ou plusieurs images d’un visage, la tâche est de trouver ou de vérifier l’identité d’une personne par comparaison de son visage à l’ensemble des images de visages stockées dans une base de données. En général, un système de biométrie faciale est constitué de deux modules : un module de segmentation de visages (détection ou localisation de visage), et un module de reconnaissance qui se déroule en trois étapes : normalisation ou prétraitement, extraction de caractéristiques faciales, classification et reconnaissance

La reconnaissance faciale pose de nombreux défis car les visages sont des objets déformable 3D.

Nous nous limitons dans ce travail à une reconnaissance à partir d’une image 2D de visage en environnements. De tels systèmes doivent pouvoir s’aranchir des problèmes suivants :

* **variations des poses**: Les changements d’orientation et les changements de l’angle d’inclination du visage engendrent de nombreuses médications d’apparence dans les images collectées. Provoquant des déformations qui font varier la forme globale du visage.



***Figure 2 Changement d’angle ( poses ).***

* **Changer l'éclairage:** L'intensité et la direction de l'éclairage pendant la prise de vue affectent grandement l'apparence du visage dans l'image. Donc il faut éviter surtout de collecter des vues à différents moments, à l'intérieur et à l'extérieur. Étant donné une forme particulière du visage humain, ces changements dans l'éclairage peuvent révéler des ombres qui mettent en évidence ou cachent certaines caractéristiques du visage.



***Figure 3 Changement d’éclairage***

* **Changement des expressions faciales:** Le visage est un élément non-typique. Le sentiment qui exprime le visage, en plus de la transformation résultant de la parole, se traduit par un changement significatif perceptible et le nombre de structures possibles ne peut être étendu de façon réaliste.



***Figure 4 Changement des expressions faciales***

* **Influence des occultations :** Un visage peut être partiellement masqué par des objets ou par le port d’accessoires tels que les lunettes, un chapeau, une écharpe. Les occultations peuvent être intentionnelles ou non. Dans le contexte de la reconnaissance, il peut s’agir d’une volonté délibérée d’empêcher la reconnaissance. Il est clair qu'elle sera d’autant plus dicile que peu d’éléments discriminants seront simultanément visibles

D'autre part, relever des défis spécifiques à la reconnaissance:

* Images prises à des années d'intervalle: l'apparence du visage peut changer de manière significative au fil des ans, conduisant à des fausses erreurs de rejet qui peuvent provoquer le refus d'accès des utilisateurs légitimes.
* **Identités similaires**: les systèmes de reconnaissance biométrique sont suspectés d'erreurs d'acceptation fictive lorsqu'un imposteur ayant une identité similaire peut être accepté en tant qu'utilisateur légitime.

Dans certains cas, les données d'entraînement utilisées pour les systèmes de détection et de reconnaissance de visage sont des images de visages frontaux, mais évidemment, certaines difficultés apparaissent lorsqu'on essaie d'analyser des visages avec des angles de pose différents. Par conséquent, certains systèmes de traitement introduisent des images ayant subi une rotation dans leur ensemble de données de formation et d'autres tentent d'identifier les caractéristiques particulières du visage comme les yeux, la bouche et le nez, afin d'améliorer les ratios positifs.

## 1.7 Conclusion

Dans ce chapitre nous avons présenté les grandes lignes de notre travail, comme les notions de base et les techniques utilisées (état de l’art). Nous avons ainsi mis en valeur les points critiques à soigner et dégager les parties à aborder lors des étapes suivantes de la réalisation de notre projet. Dans le chapitre suivant, on abordera l’analyse de notre logiciel pour spécifier les étapes du travail et en dégager les principales fonctionnalités de l’application.

# CHAPITRE 2 : lancement du projet

1. CAHIER DE CHARGE DU PROJET
2. Besoins fonctionnels

L’application delta-face doit être capable d’apprendre des visages inconnus par le biais d’une caméra et de les stocker dans un dossier pour pouvoir un usage d’entrainement à la reconnaissance des étudiants à tout moment. A la reconnaissance d’un visage d’un étudiant doit tout de suite l’associer à son identité personnelle. Elle doit aussi marquer présent dans la liste de présence. L’application delta-face sera sollicitée par un administrateur et un chef de salle.

**Concernant l’administration** :

* S’authentifier: L’administrateur peut accéder à son interface avec une adresse mail et mot de passe spécifique pour lui.
* Ajouter images des étudiants: l’administrateur doit capturer trois images (face, droite, gauche) pour chaque étudiant pour faciliter a la camera de bien connaıtre les étudiants.
* Gérer statistiques: L’administrateur aura la possibilité de : Visualiser des pourcentages et les courbe de perte et de gain de reconnaissance sur l’application delta-faces.

**Concernant le chef salle** :

* S’authentifier: le **chef salle** peut accéder a son interface avec une adresse mail et mot de passe spécifique pour lui.
* Capturer les visages des étudiants.
* Générer la liste de présences des étudiants ayants composé dans sa salle.

1. Besoins non fonctionnels

Les besoins non fonctionnels d´écrivent les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement d’où l’application doit présenter une interface agréable et facile à utiliser. Le système doit répondre aux exigences suivantes :

**L’ergonomie des interfaces** : L’application doit être compatible sur différents systèmes d’exploitation (LINUX, Windows...) pour être mieux adaptée aux utilisateurs vises.

**La sécurité** :

* il est n´nécessaire de s’authentifier avant de faire n’importe quelles taches.
* Tous les mots de passe doivent être cryptes dans la base.

**Audit** : La vérification des champs est importante lors de la saisie pour éviter de saisir des caractères incorrects.

**La fiabilité et la facilite à administrer** : Il faut que le système conçu soit facile à administrer et capable de fonctionner sans erreur pour améliorer l’interaction entre l’utilisateur et la machine.

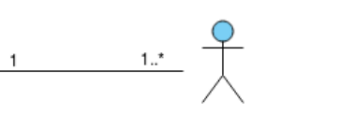
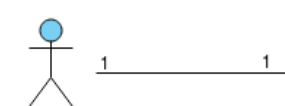
1. Identification des acteurs

**Un acteur** est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une ou plusieurs fonctions concernant les cas d’utilisations. Pour l’application delta-face nous les avons évoqué dans ce qui précède mais nous donnons de plus amples détails dans le tableau les principaux acteurs de cette application sont réalisé dans le tableau 1.1:

|  |  |
| --- | --- |
| Acteur | Role |
| Administrateur | un admin a pour rôle d’entraıner la camera (entrer trois images pour chaque étudiant selon trois positions : de face, de gauche et de droit), gérer les données et gérer les statistiques. Dépendant du nombre d’étudiants, ici il s’agit de nous. |
| Chef-salle | Un chef-salle a pour rôle de vérifier si l’étudiant qui se pointe est vraiment de ce niveau et appartient réellement a sa salle de composition, donc pour cela il doit prendre une image de chacun des étudiant pointer et vérifier s’ils sont autorisés ou non à composer dans sa salle. |

1. Diagramme de contexte

La figure 1.2 présente le diagramme de contexte qui montre les acteurs interagissant avec le système ”Delta-face”. Delta-face est un système de pointage base sur la reconnaissance faciale, qui permet à un chef-de-salle de gérer la liste des étudiant et de consulter leur état de présence.



1. METHODE DE DEVELOPPEMENT
2. Méthode adoptée

La majeure hantise de toute équipe de développement est de ne pas terminer le projet dans les dates prévues à cause d’une mauvaise spécification ou d’un changement brusque des besoins cela conduit à un environnement de travail stressant et à la peur de dépassement de capacités et des délais de livraison. Le recours à l’Agile permet de surpasser ces types de problèmes. Le cadre de processus Scrum est connu pour ses nombreux vertus dont le numéro un est la production de logiciels de qualité s’ajoute la maıtrise de l’´évolution des besoins du marché et de l’utilisateur Nous ajoutons que Sacrum est :

* Facile à comprendre.
* Donne une souplesse dans le d´enveloppement d’un projet.
* Mené vers un produit s’adapte le plus aux exigences du client et des utilisateurs.
* Permet par ses rôles et ses évènements une coopération étroite entre clients et équipes de développement.

1. Constitution de l’´équipe

Compose de trois principaux rôles qui sont :

* **Product Owner (PO):** C’est le représentant du client, c’est lui qui connait parfaitement et sait définir les fonctionnalités du produit, c’est lui qui les priorise. C’est à lui d’accepter ou de rejeter les taches réalisées. Il est le maıtre de la réunion de planification de Sprint. Dans le cas de réalisation de mon projet, le PO est BANOLOG.
* **Scrum master (SM)**: FOKOU LAURES chargée de prendre le rôle du Scrum master puisqu’il assure le suivie de processus, assure un travail complet et productif, entraıner l’équipe de développement dans des environnements organisationnels dans lesquels Scrum n’est pas encore adopte et compris et les aider à créer un produit de grande valeur ajoutée.
* **Equipe de développement (DevTeam**): Nous, FOTSO YVANOL, TANKOU GHISLAIN, représentons l’équipe de développement puisque nous réalisons toutes les fonctionnalités. Nous sommes tenus de présenter chaque daily meeting les résultats de notre travail. Nous maintenons a jour les spécifications détaillées de notre application et nous sommes entièrement engages et responsable de la réalisation des User Stories du Product Backlog.

1. Etude comparative

Notre système DELTA-FACE s’apparente à un système de pointage. Un système de pointage est un outil simple qui permet de mesurer les heures de travail des salariés, cet outil a du évoluer au cours du temps. Il existe des différents types de systèmes de pointage tels que: la feuille de pointage, la pointeuse par les empreintes digitales, la pointeuse mécanique, la pointeuse mobile, la badgeuse...

* **Feuille de pointage**: Comme tout système, ce système permet d’ajuster les heures de travail avec précision de manière que chaque salarie reçoit le salaire qu’il mérite. Ce système n’est pas payant, il est peu couteux, mais présente de nombreux inconvénients qui font que les entreprises évitent de les utiliser, surtout les grandes entreprises.
* **Système de reconnaissance par les empreintes digitales**: Ce système est un peu plus innovant que la feuille de pointage, fiable, performant et facilite le marquage des absences des salaries. Toutefois il présent un gros inconvénient qui est le manque d’hygiène, les microbes des doigts sales se dispersent sur le lecteur.
* **Badgeuse**: Pour justifier les horaires du personnel, l’employeur doit se mettre en place devant une badgeuse. Une badgeuse permet de mieux gérer le temps de travail d’un employé, elle peut même ouvrir la porte mais le jour ou un employé oublie ou perd son badge il ne peut pas accéder. Récupérer un autre badge peut causer une perte de temps. De même qu’un employé peut facilement badger pour un autre.

Afin de contourner les inconvénients de ces types de systèmes de pointage, nous proposons de réaliser un système base sur la reconnaissance faciale, cette technique est peu couteuse, performante, conviviale.



Figure 1.1: Exemples de systèmes de pointage

1. Architecture globale du logiciel

Dans Scrum il formellement déconseille pour la DevTeam de consacrer tout un sprint a étayer les infirmes détails de l’architecture d’une application. Celle-ci se construira au fur et à mesure de l’intégration des incréments des différents sprints. Toutefois le Product Owner de part sa maıtrise de son produit, dispose d’une vision claire et globale de son architecture physique et logique. C’est architecture globale que nous expliquons dans ce qui suit. Dans Delta-face, le chef-salle et l’administrateur peuvent réagir avec le système. Dans le premier niveau, l’administrateur utilise un ordinateur pour accéder a ses interfaces ainsi que le chef salle. Au niveau deux le serveur reçoit ces informations et communique avec le serveur de base de donnée pour authentifier l’administrateur et le chef -salle. Le serveur souvent a besoin de prendre ou de mettre à jour quelques données et il doit interroger la base. Dans le dernier niveau, la base de données va stocker les données et les fournir au serveur sur demande.



1. Product Backlog de Delta-face

Un Product Backlog contient une liste des éléments fonctionnels nécessaires pour l’accomplissement du produit. La DevTeam interviendra avec ses compétences complémentaires, par sa créativité, par son engagement et implication, par sa solidarité et par son grand sens de responsabilité a traduire le Product Backlog en un produit compétitif a placer sur notre établissement. Le Product Backlog de Delta-face est dresse dans le tableau 1.2. Il contient les champs suivants :

**ID** : C’est le nombre unique pour chaque user story.

**User stories** : Ce sont les phrases qui d´écrivent les fonctionnalités souhaitées par le client.

**Priorité** : C’est la priorité du développement de chaque histoire utilisateurs. J’ai pu établir une étude afin de répartir les histoires par priorité selon les normes suivantes:

• La complexité

* La priorité :
* – Elevee : doit être fait.
* Moyenne : devrait être fait dans la mesure du possible.
* Faible : pourrait être fait dans la mesure où cela n’a pas d’impact sur les autres taches.

|  |  |  |
| --- | --- | --- |
| Id | User stories | Priorite |
| 1 | En tant qu’administrateur, je veux pouvoir a tout moment, consulter les données | elevee |
| 2 | En tant qu’administrateur, je peux entraıner mes cameras sur de nouveaux étudiant en utilisant le réseau de neurone | elevee |
| 3 | En tant qu’administrateur, je peux entraıner mes cameras sur de nouveaux étudiant en utilisant la méthode de calcul par distance Euclidienne | elevee |
| 4 | En tant qu’administrateur, je veux pouvoir supprimer des étudiants | elevee |
| 5 | En tant que chef de salle, je veux pouvoir consulter historiques d’absences des étudiant | elevee |
| 6 | En tant que chef de salle, je veux pouvoir générer la liste de présence des étudiants | elevee |
| 7 | En tant que chef de salle, je veux pouvoir lancer et arrêter la camera | moyenne |
| 8 | En tant que chef de salle, je veux pouvoir consulter la liste des présence uniquement et la liste de présence et absence de mes étudiants | moyenne |

Table 1.2: Product Backlog

1. Diagramme des cas d’utilisation global

Le but de cette section est de d´écrire le comportement attendu de l’application. Pour ce faire, nous nous basons sur le diagramme de cas d’utilisation globale du langage de modélisation UML pour l’administrateur ainsi que le chef-salle pour modéliser les fonctionnalités de l’application. La figure 1.3 nous montrons ce diagramme qui présente les différentes activités réalisées par l’administrateur et le chef-salle afin de gérer et contrôler l’application en plus des différentes activités offertes pour un chef-salle.

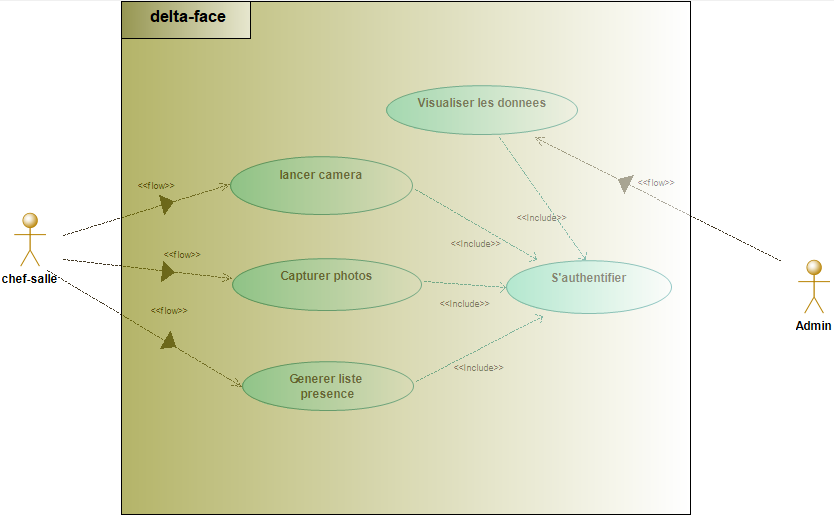


Figure : 1.3 Diagramme de cas d’utilisation globale du système

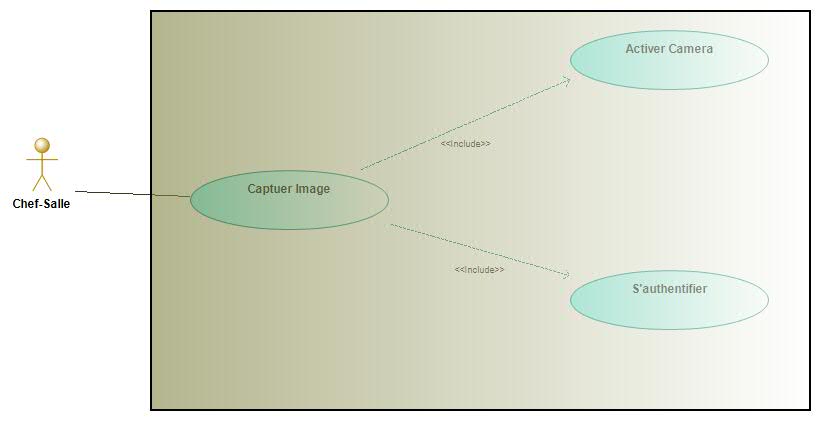


Figure : 1.3 Diagramme de cas d’utilisation capturer une image

* Description des cas d’utilisation
* Lancer la camera
* Nom : Lancer Camera
* Acteur : Chef-salle
* Type : Primaire
* Évènement déclencheur : Le chef salle veut capturer un étudiant pour vérifier s’il fait partie de la salle.
* Description : le chef -salle appui sur le bouton onCamera de la menu bar pour lancer la camera de l’ordinateur ou la webcam
* Dépendance : les étudiants doivent figurer dans le système.
* Précondition : le chef-salle doit au préalable s’authentifier.
* Exception : Aucune.
* Règle de terminaison : le chef-salle appuis la touche « **c**«  du clavier pour capturer ou appui sur le bouton OffCamera de l’application.
* Générer la liste de présence :
* Nom : Générer la liste de présence ;
* Acteur : chef-salle ;
* Type : primaire ;
* Évènement déclencheur : Le chef salle veut obtenir la liste des étudiants qui ont répondu présent à l’examen.
* Description : le chef -salle appui sur le bouton générer la liste de présence situer dans la menu bar de l’application
* Dépendance : Au moins un étudiant doit exister dans le système.
* Précondition : Le chef-salle doit s’authentifier.
* Exception : aucune.
* Regel te terminaison : fermeture du fichier contenant la liste de présence/absence des étudiant.
* S’authentifier
* Nom : s’authentifier ;
* Acteur : chef-salle ;
* Type : primaire ;
* Évènement déclencheur : Le chef salle veut accéder à son interface.
* Description : le chef -salle démarre l’application et accède a l’interface login ou il doit renseigner ses paramètres.
* Dépendance : Ses paramètres doivent figurer dans le système / dans la base des chef-salle.
* Précondition : aucune.
* Exception : paramètres/information incorrecte.
* Regel te terminaison : paramètre /information correcte.

1. Branche Technique

Nous entamerons ce chapitre par la présentation de l’architecture de notre système. Ensuite, nous passerons à la deuxième partie la présentation du diagramme de déploiement et des diagrammes de séquences détaillés et diagramme d’état, et diagramme d’activité et en fin le diagramme de classe.

1. Architecture

Opérer une connexion directe à la base de données depuis le téléphone Android n’est pas conseillé d’un point de vue architecture logicielle. Il est en effet préférable de passer par un middleware. Cette couche serveur intermédiaire sera la seule habilitée à se connecter à la base de données, ce qui est plus sécurisé. Donc La méthode la plus répandue pour se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec Django, donc le plus évident est d'écrire des scripts Python/Django pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP. J’ai codé les données dans le format JSON, afin de communiquer les données entre Django et Android, en exploitant les options faciles à utiliser construit dans les fonctions JSON dans les deux langages.

En fait, si on parle de l'architecture 3-tiers de point de vue technologie, le chef-salle est la plateforme Android, le serveur web est le Django et le serveur de bases de données est le MySQL.

En fait notre projet est divisé en trois parties à savoir l’écriture des scripts Django, et le développement de l’application Android, puis en fin le développement d’une application Desktop (python => Tkinter).

La partie serveur est composée de deux serveurs distants : le serveur web et le serveur de base données. Le serveur Web utilisé est le serveur Django, il est le serveur le plus répandu sur internet pour déployer les scripts python. Le serveur de bases de données utilisé est le serveur MySQL

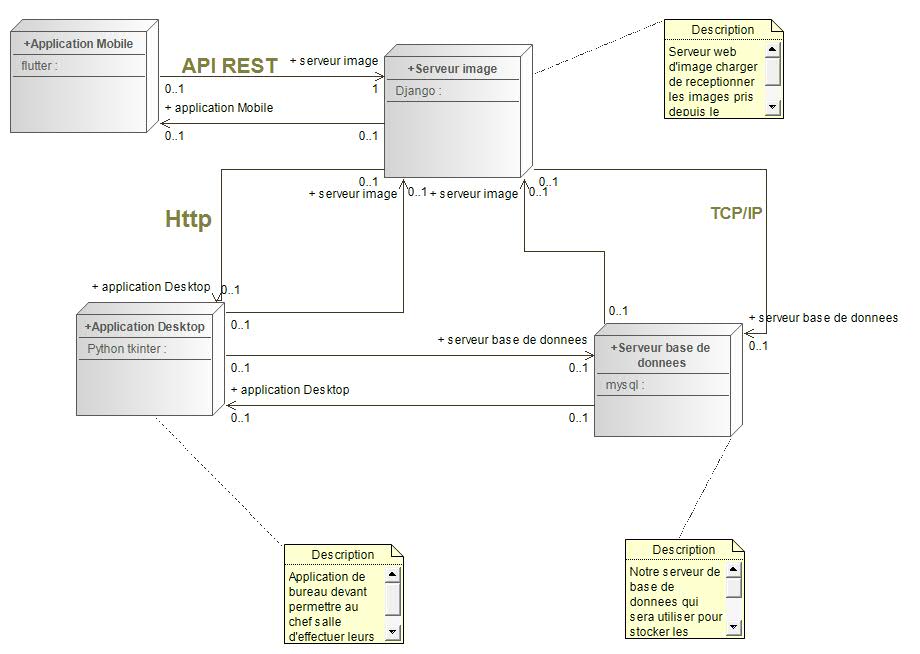


Figure 1.4 : Diagramme de déploiement.

1. Diagramme de séquence
2. Notion de scenario

Un scenario est une trame narrative qui décrit un ensemble d’interactions possibles entre des utilisateurs (acteurs) et des systèmes. Il décrit dont une suite d’étapes et d’interaction entre acteurs et système. Afin de mieux appréhender le concept, il convient de décrire les scenarios spécifiques aux besoins fonctionnels des utilisateurs à l’aide des diagrammes de séquences.

Nous pouvons distinguer 05 scenarios importants dont:

* Authentification, qui concerne les acteurs intervenant dans le système
* Visualiser les donnees
* Lancer la Camera
* Générer la liste de présence
* Capturer les photos

Le diagramme de séquence permet de représenter la succession chronologique des opérations réalisées par un acteur. Il indique les objets qui seront manipulés par l'acteur et les opérations. Il représente les messages échangés entre les objets dans un ordre chronologique. Il donne une notion temporelle aux messages. Nous l’utilisons au niveau de la branche fonctionnelle de notre processus pour représenter l'opération de reconnaissance d’un étudiant, d’authentification des chef-salle.

### 2-Enumération des diagrammes de séquences

1. Digramme de séquence détaille du cas « s’authentifier »

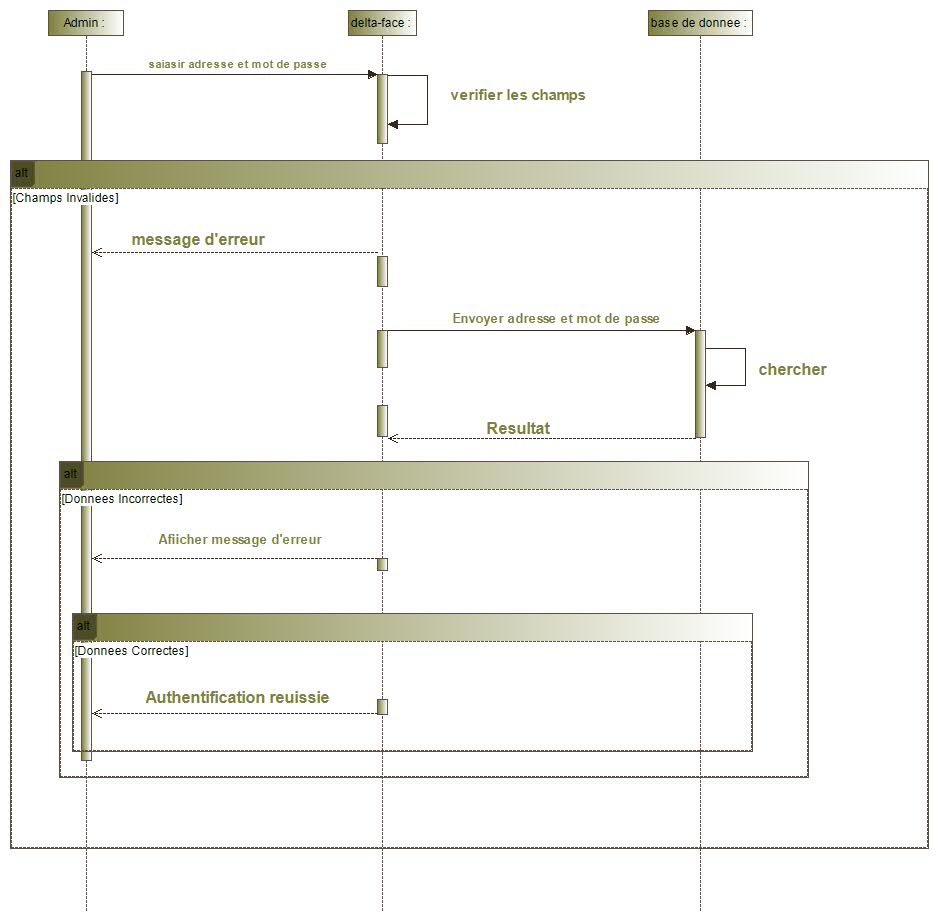


Figure :1.6 diagramme authentification <<admin>>.

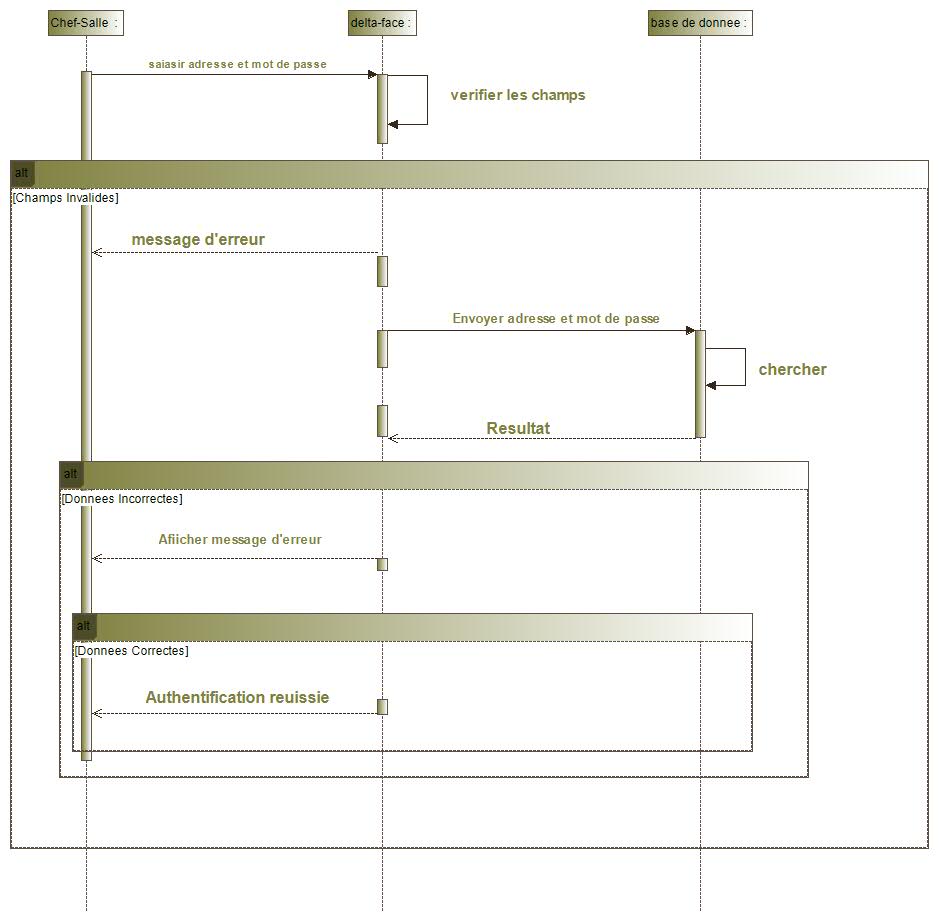
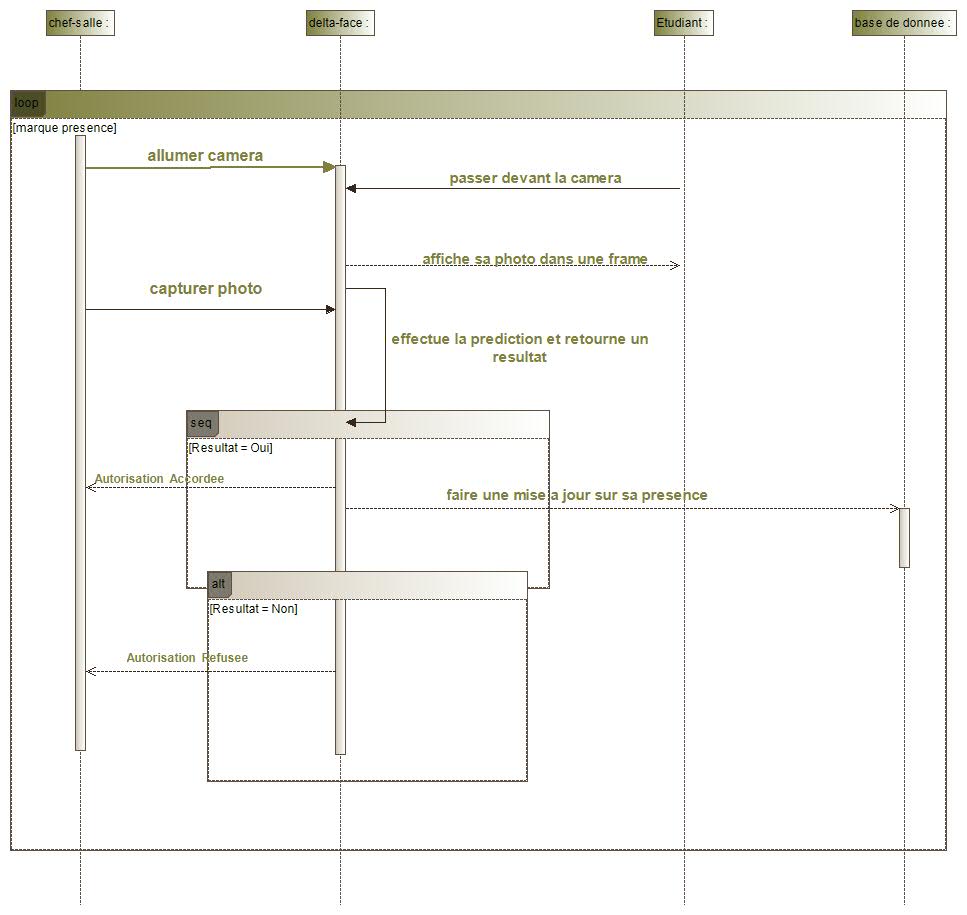
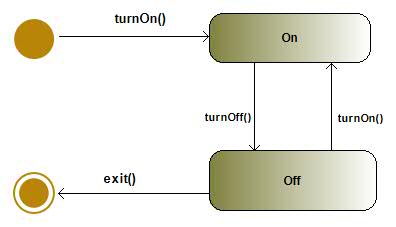


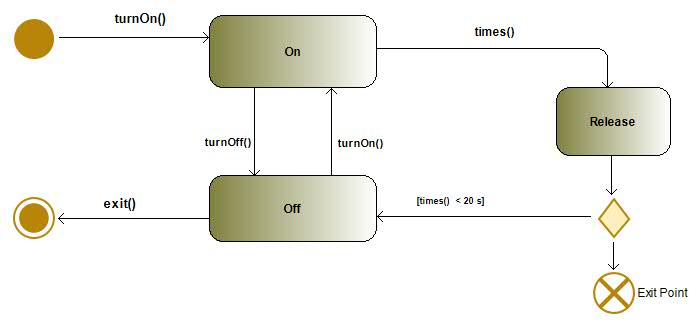
Figure :1.7 diagramme authentification « chef-salle «

1. Diagramme Séquence <capture photo >> pour autorisation en salle

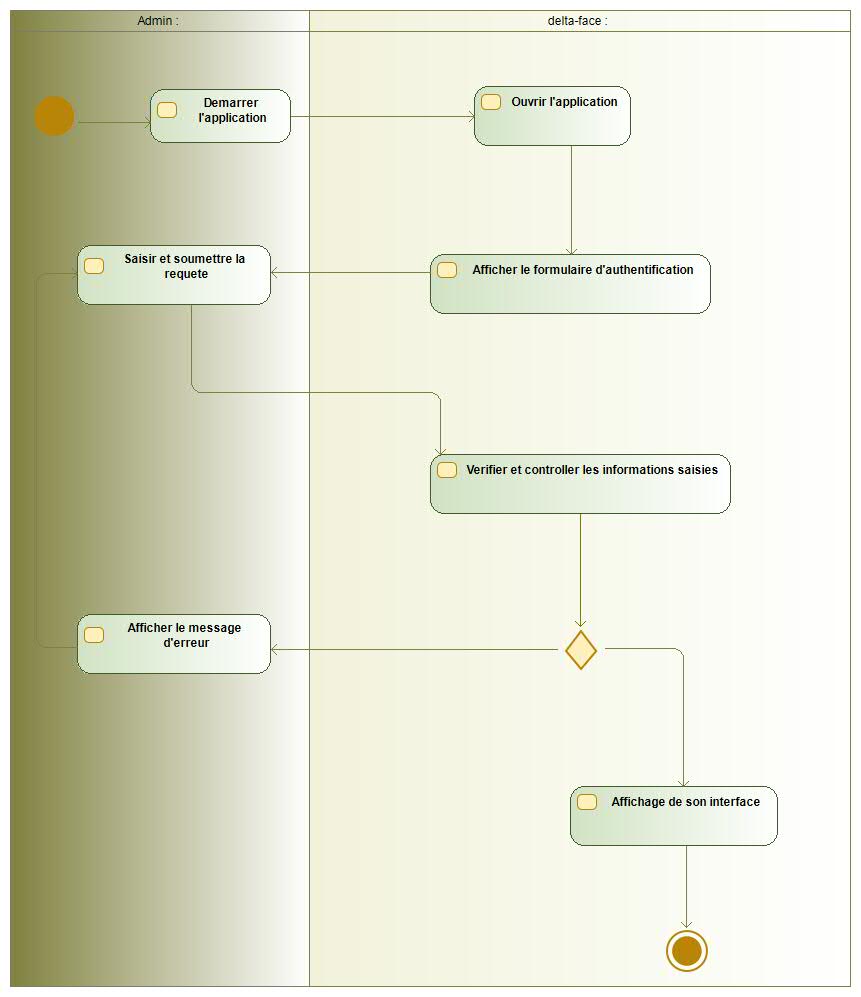


1. Diagramme d’etat de la <<camera>>

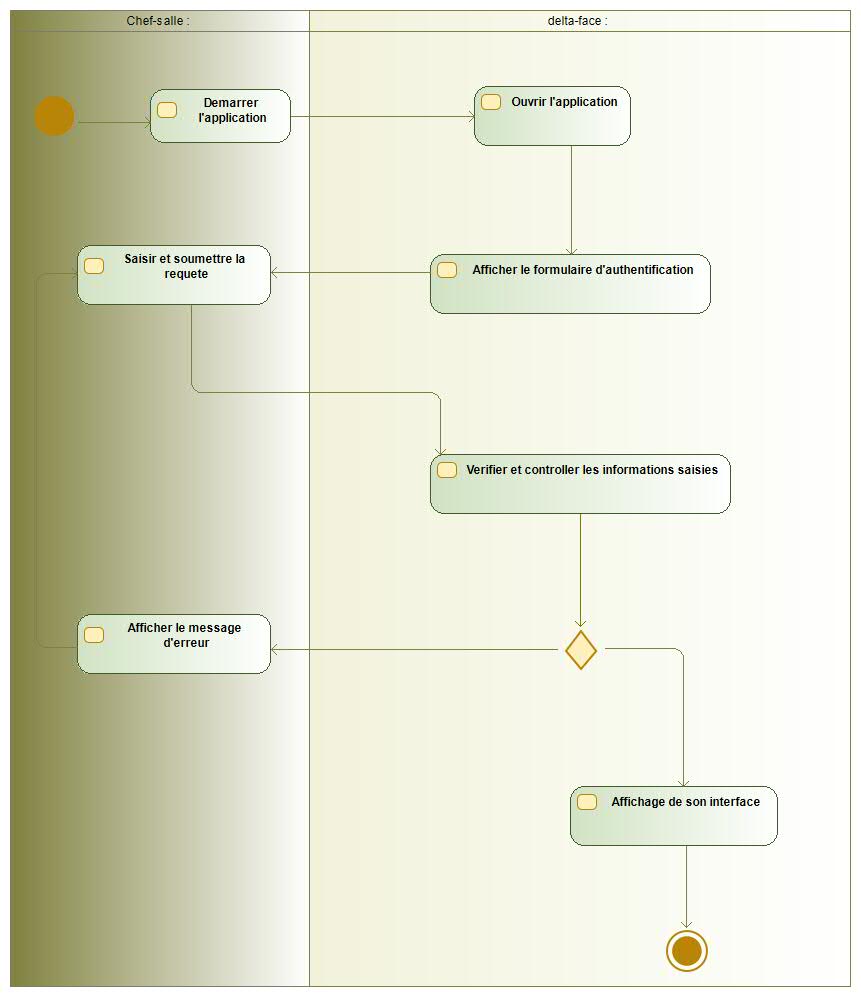




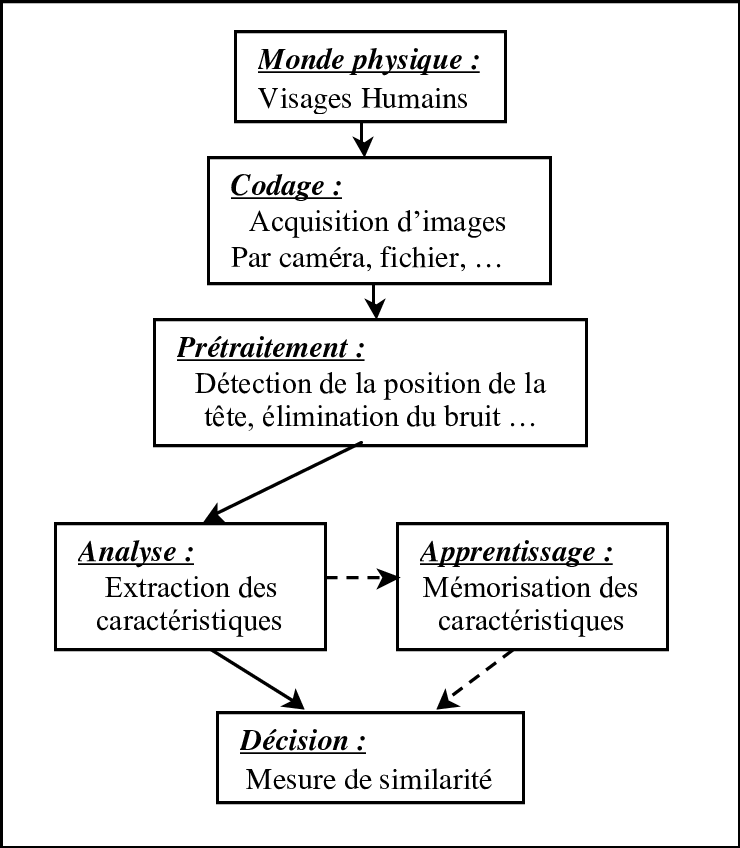
1. Diagramme d’activité <<d’authentification>> pour administrateur



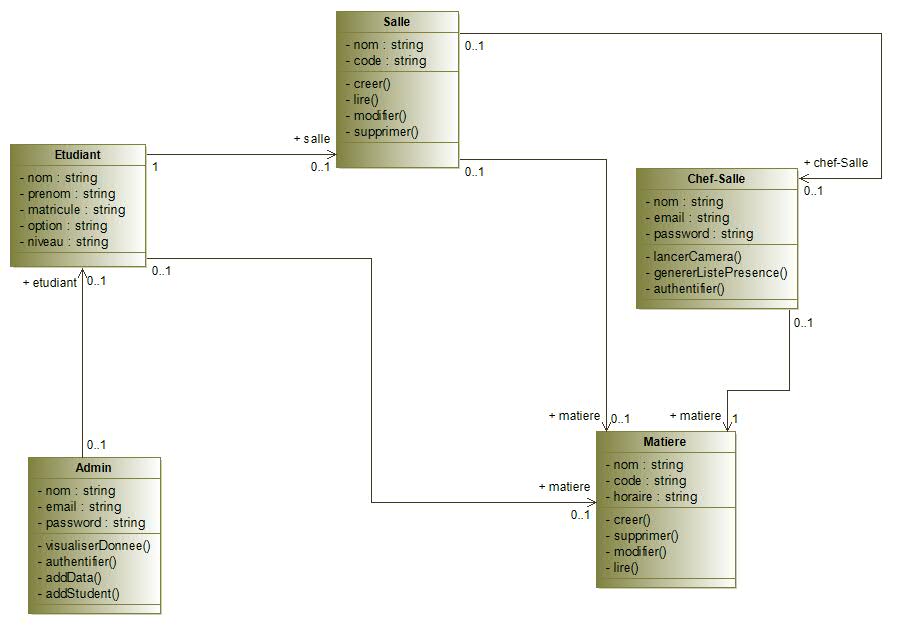
1. Diagramme d’activité <<d’authentification>> pour chef-salle



1. Diagramme d’activité de la reconnaissance



1. Diagramme de classes du systeme



1. Réalisation

* **XAMP**: est un acronyme pour "cross-Platform, Apache, MySQL, PHP, and Perl" (plateforme croisée, Apache, MySQL, PHP et Perl). Il contient tous les composants nécessaires pour développer des applications web, y compris un serveur web (Apache), une base de données (MySQL) et un langage de script (PHP).
* **MYSQL**: MySQL est un système de gestion de base de données relationnelle (SGBDR) open source, développé par Oracle Corporation
* **PYTHON**: Python est un langage de programmation interprété, multiplateforme, orienté objet et de haut niveau. Il a été créé à la fin des années 1980 par Guido van Rossum et est actuellement maintenu par la Python Software Foundation. Python est utilisé dans de nombreux domaines, tels que l'analyse de données, l'apprentissage automatique, l'intelligence artificielle, la science des données, l'automatisation, le développement de jeux, le développement web, etc.
* **DJANGO** : Django est un framework web open source en Python. Il a été créé en 2003 par Adrian Holovaty et Simon Willison pour répondre aux besoins de développement rapide d'applications web complexes et évolutives. Django suit le modèle de conception MVC (Modèle-Vue-Contrôleur) et offre une grande variété de fonctionnalités prêtes à l'emploi pour simplifier le développement d'applications web, telles que la gestion de l'authentification, la gestion des formulaires, la gestion des modèles de base de données et la génération automatique de code HTML.
* **FLUTTER**: Flutter est un framework open source développé par Google pour la création d'applications mobiles pour les systèmes d'exploitation Android et iOS.
* **POSTMAN** : est un outil de développement et de test d'API (Application Programming Interface) qui permet aux développeurs de créer, tester, documenter et partager des API.
* **MODELIO** : Modelio est un outil de modélisation open source pour les systèmes d'information et les applications d'entreprise. Modelio offre une interface utilisateur graphique conviviale et intuitive pour la modélisation, la conception et la documentation des systèmes d'information et des applications d'entreprise. Il permet aux utilisateurs de créer des modèles de différentes perspectives, d'ajouter des annotations et de générer automatiquement des documents de spécification et des rapports de modélisation.
* **NUNPY** : NumPy est une bibliothèque open source pour le langage de programmation Python, qui est largement utilisée pour le calcul scientifique et numérique. NumPy offre des tableaux multidimensionnels (appelés "ndarrays") et des fonctions mathématiques pour effectuer des opérations sur ces tableaux, telles que les opérations arithmétiques, les fonctions trigonométriques, les fonctions statistiques, etc.
* **TKINTER** : Tkinter est une bibliothèque graphique pour le langage de programmation Python. Tkinter offre une variété de widgets graphiques préconfigurés tels que des boutons, des boîtes de dialogue, des champs de texte et des menus déroulants, ce qui facilite la création d'interfaces utilisateur graphiques pour les programmes Python.
* **TENSORFLOW** : TensorFlow est une bibliothèque open source pour l'apprentissage automatique et l'intelligence artificielle. TensorFlow offre une grande variété d'outils et de bibliothèques pour le développement d'applications d'apprentissage automatique, y compris des outils pour le traitement des données, le développement de modèles, l'entraînement de modèles, l'inférence de modèles et le déploiement de modèles.
* **KERAS** : est une bibliothèque open source pour l'apprentissage en profondeur (deep learning) qui permet aux développeurs de créer et d'entraîner facilement des réseaux de neurones artificiels

1. Mise en place des Sprint

Un Sprint est ”le cœur de Scrum ” . Il dure généralement entre 2 et 4 semaines, c’est une suite de taches que l’équipe Sacrum doit bien effectuer pour un objectif préalablement fixe. Dans ce chapitre nous allons présenter la réalisation du premier Sprint qui se chargera de la réalisation des objectifs des éléments du Product Back log les plus élevés, pouvant se tenir dans un sprint, et respectant une capacité de travail de DevTeam soutenue. Rappelons que la DevTeam ce n’est que moi-même. L’organisation du travail se fait sur trois phases principales qui sont l’analyse, la conception et la réalisation.

1. Séance du Sprint planning

Chaque Sprint aura des buts aussi détermines définis auparavant par l’équipe, ce Sprint est entièrement dédie à des technical stories qui ont comme objectif de montrer les importantes étapes à réaliser pour développer la première partie de notre projet qui est la reconnaissance avec apprentissage sur les captures des caméras. Les technical stories ce sont des histoires utilisateurs qui n’apportent aucune valeur perçue a l’utilisateur final, mais offre une architecture importante nécessaire pour offrir une valeur future aux utilisateurs.

1. Séance du Sprint planning 1

Suite a une réunion de planification du Sprint suivie par le product owner qui dure six heures. Nous avons pris la décision suivante pour réussir à effectuer l’apprentissage de la camera:

* **Opencv** pour capturer les images à partir des caméras.
* **Dlib face detecto**r pour détecter les visages dans une image.
* **Dlib face recognition** pour extraire le descripteur d’un visage.
* **Distance Euclidienne** pour trouver le visage le plus proche du visage cible (premier essai)
* **Numpy**
* **Pandas**
* **Keras**
* **Tensorflow**

Sprint backlog 1 estimé

Le Sprint Backlog est un outil qui facilite la répartition des taches. Donc pour atteindre l’objectif de ce Sprint, il faut réaliser les taches présentées dans ce Sprint Backlog sans défaut, ce tableau 2.1 montre les importantes taches que nous devons les réaliser :

|  |  |  |
| --- | --- | --- |
| Id | User stories | Complexity |
| 1 | En tant qu’un d´enveloppeur je dois savoir l’utilisation de Opencv pour capturer par les caméras. | Elevee |
| 2 | En tant qu’un développeur je dois savoir l’utilisation de dlib face detector pour la détection des visages. | Elevee |
| 3 | En tant qu’un développeur je dois savoir l’utilisation du dlib face recognition pour extraire les descripteurs de visages. | Elevee |
| 4 | En tant qu’un développeur je dois maıtriser le calcul de la distance Euclidienne. | Elevee |
| 5 | En tant qu’un développeur je dois maıtriser l’utilisation de la librairie pandas | Elevee |
| 6 | En tant qu’un développeur je dois maıtriser l’utilisation de la librairie Numpy | Elevee |
| 7 | En tant qu’un développeur je dois maıtriser l’utilisation de la librairie Tensorflow | Elevee |

1. Demarrage du sprint

Plusieurs outils que l’on peut l’utiliser pour établir la reconnaissance faciale telle que l’utilisation de Tensorflow, un framework qui contient des algorithmes qu’on peut utiliser pour faire l’apprentissage des caméras et le training s u r l es images. Tensorflow génère plusieurs calculs et nécessite des capacités de traitement très élevés, d’autant plus qu’il faut investir beaucoup de temps pour se l’approprier. Notre machine de 8Go et notre durée de stage de 4 mois s’apprêtent adéquats pour l’utilisation de cette excellente technologie de reconnaissance faciale.

1. Apprentissage de la Camera

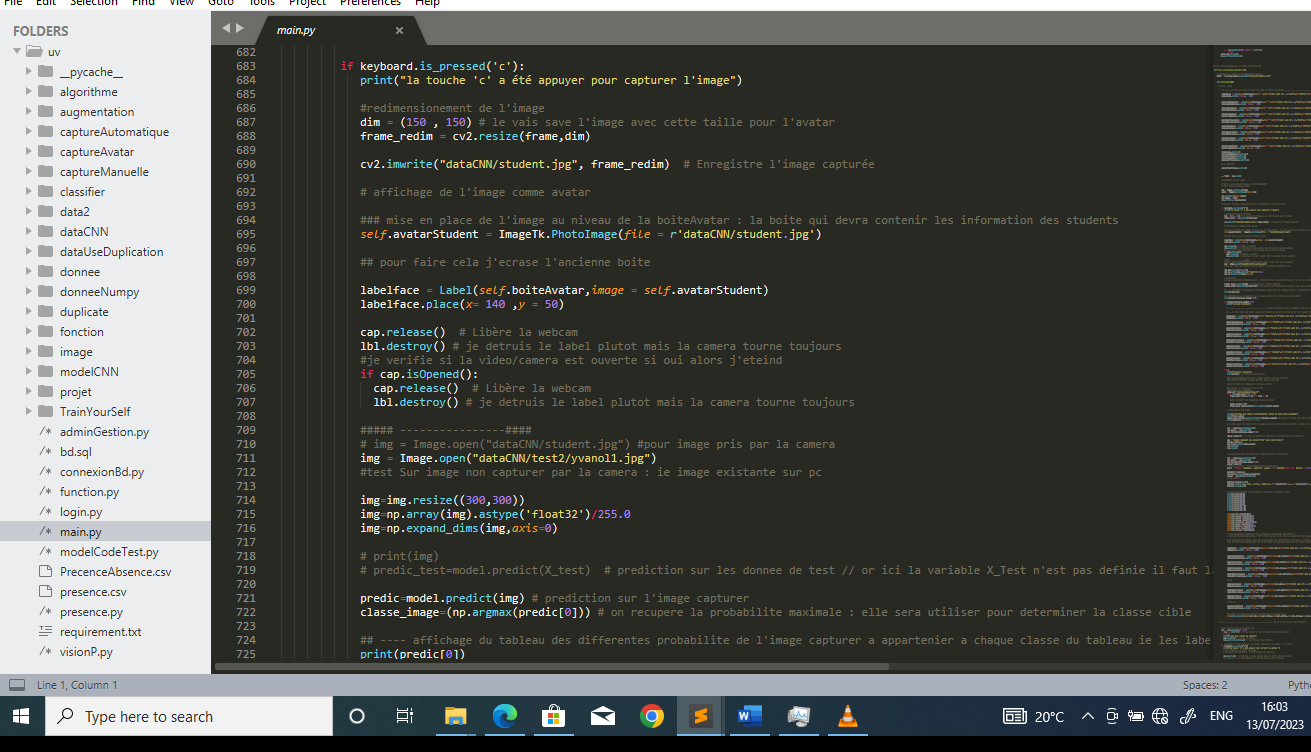
**Capture d’image**: Pour qu’on puisse ouvrir notre camera et capturer des images il faut utiliser Open Cv (Open computer Vision). OpenCV est une bibliotheque d’executable qui facilite la reconnaissance d’images en mouvement. Elle fournit une interface permettant a Python de capturer des flux video, et egalement pour extraire les images afin que nous puissions les traiter plus tard.

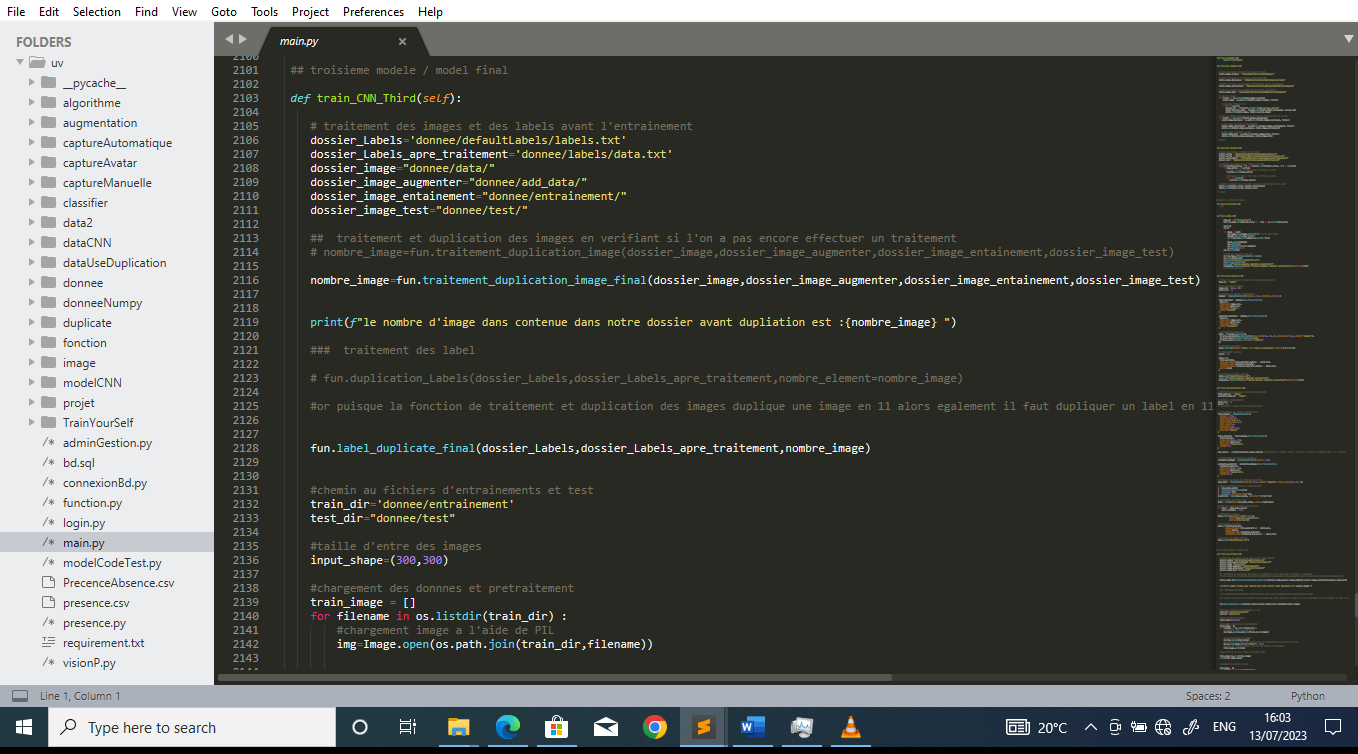


**Figure :1.9 Logo open CV**

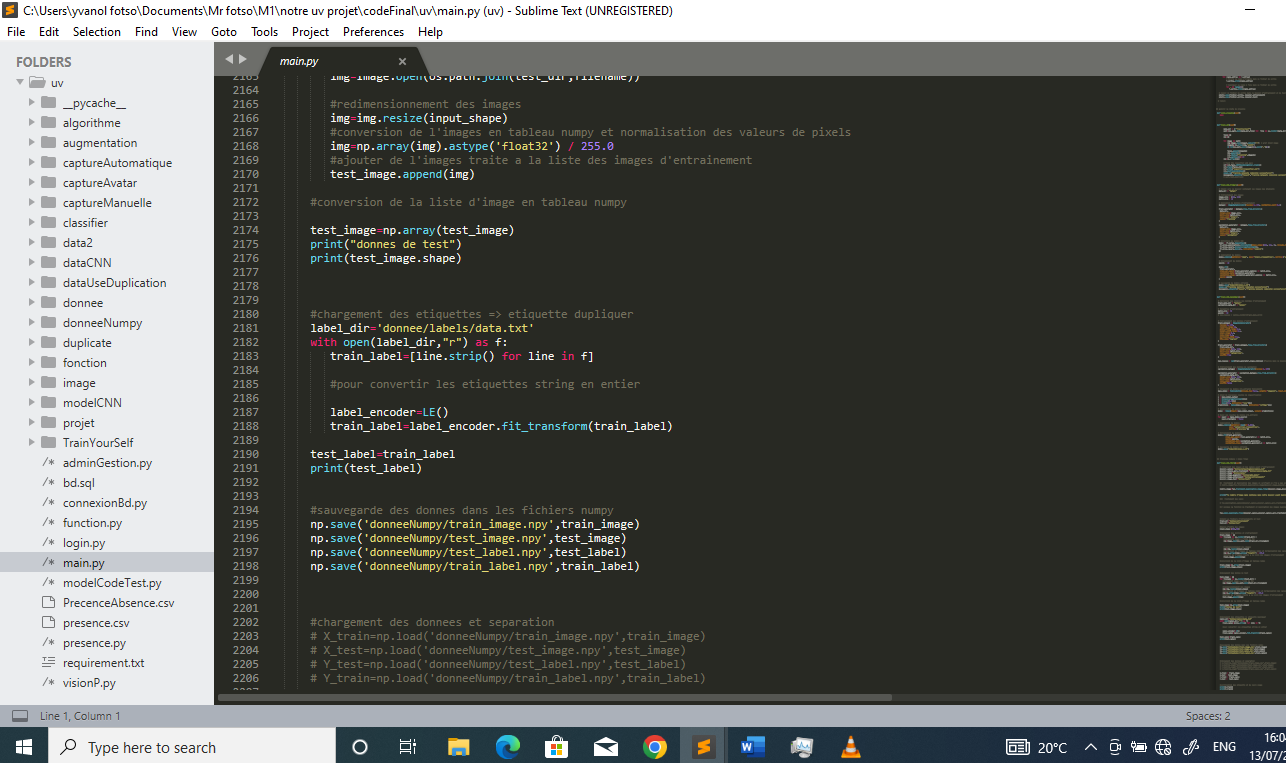
1. Codage

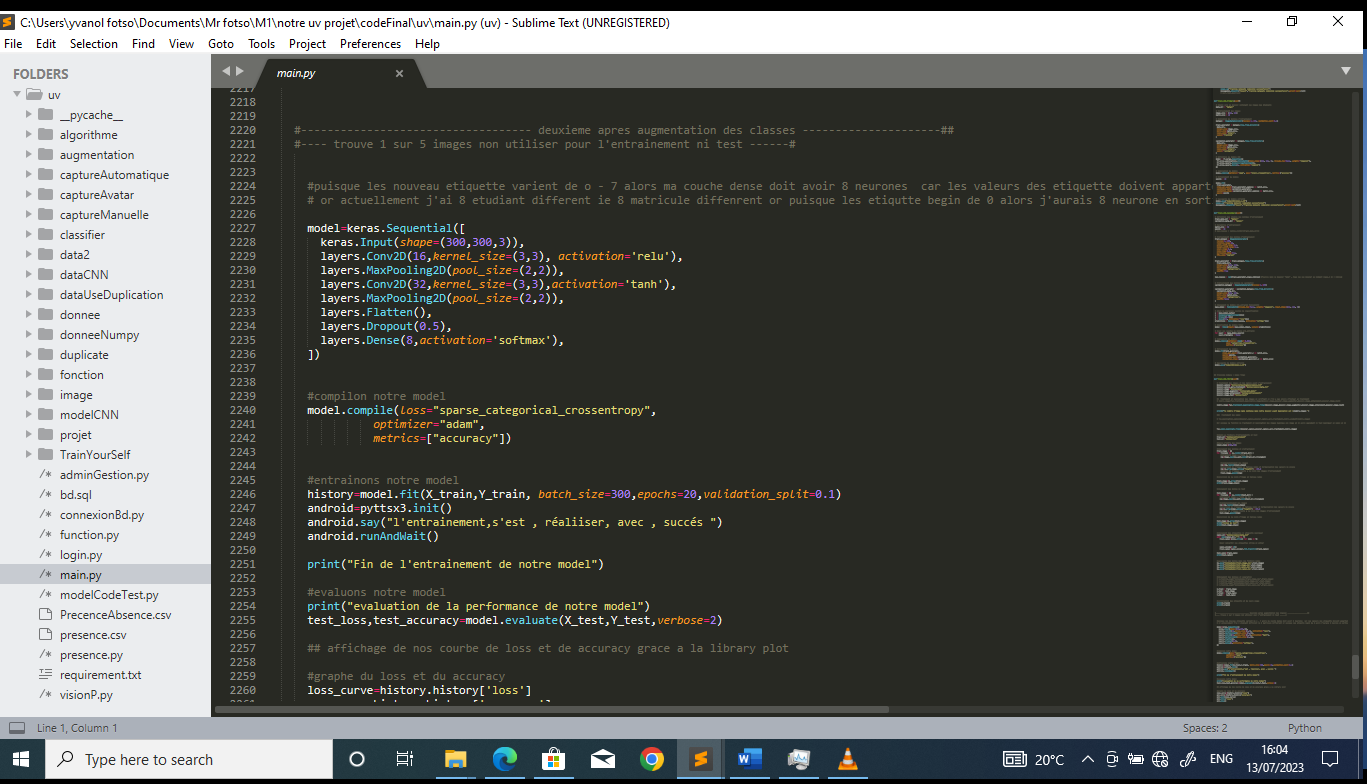
Application desktop

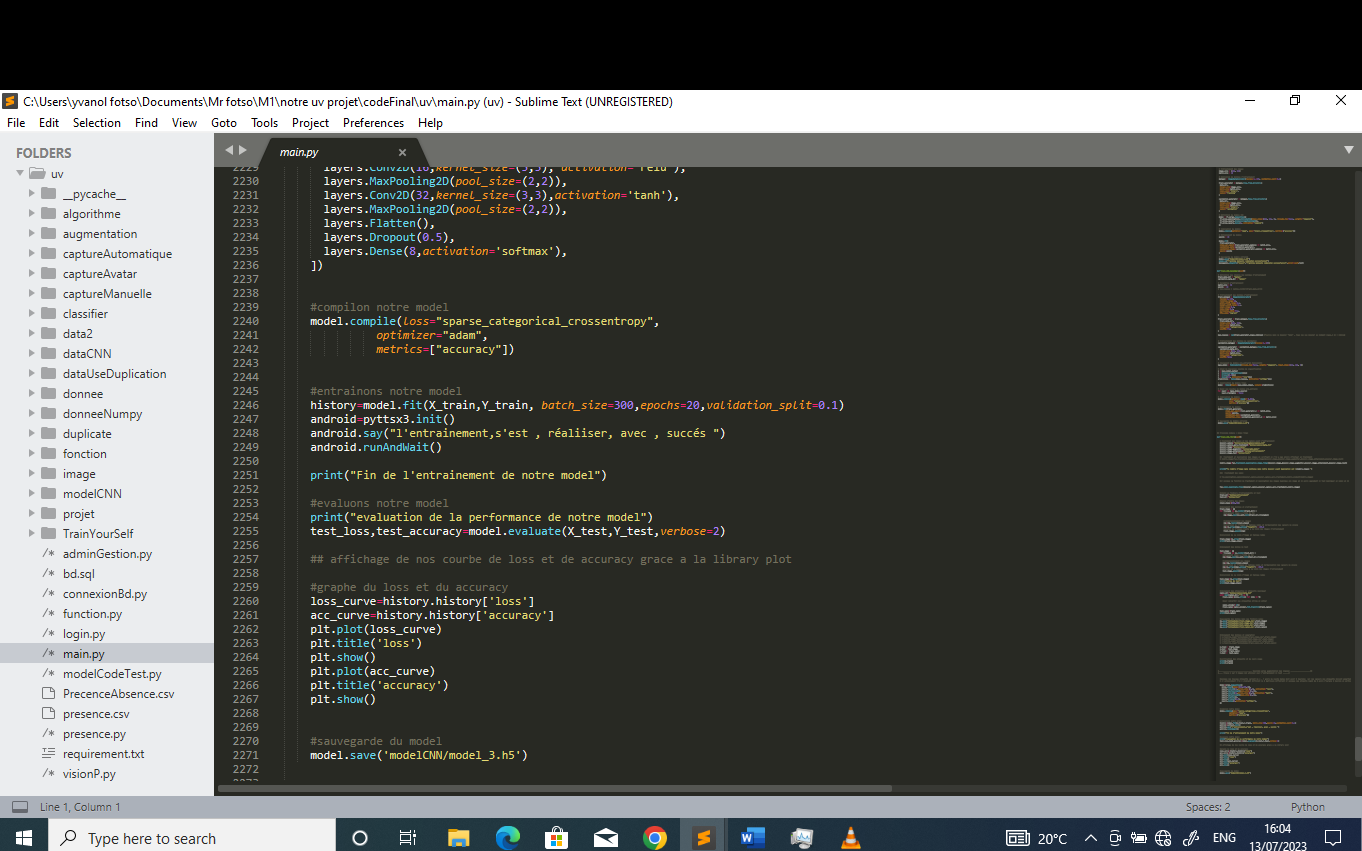




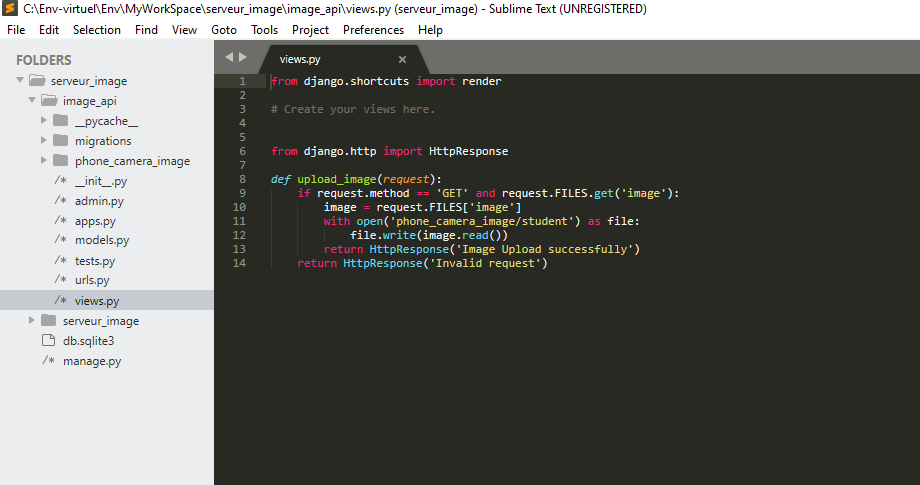


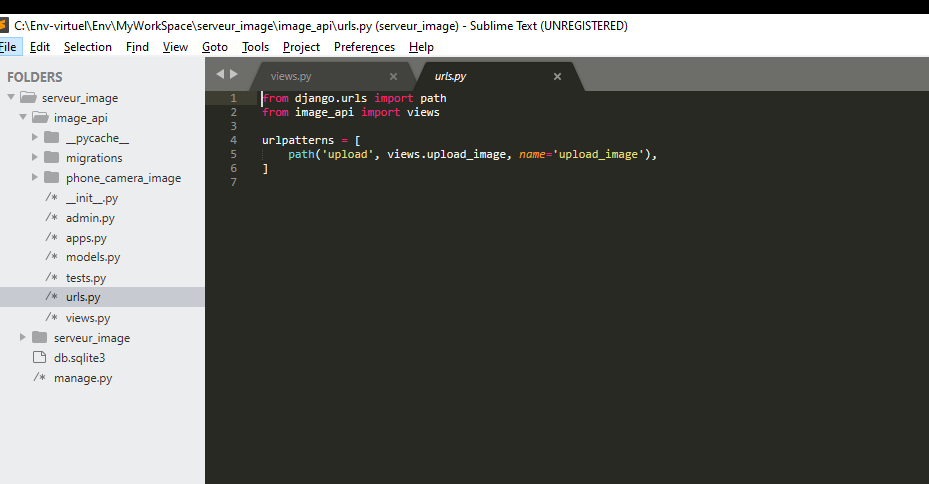




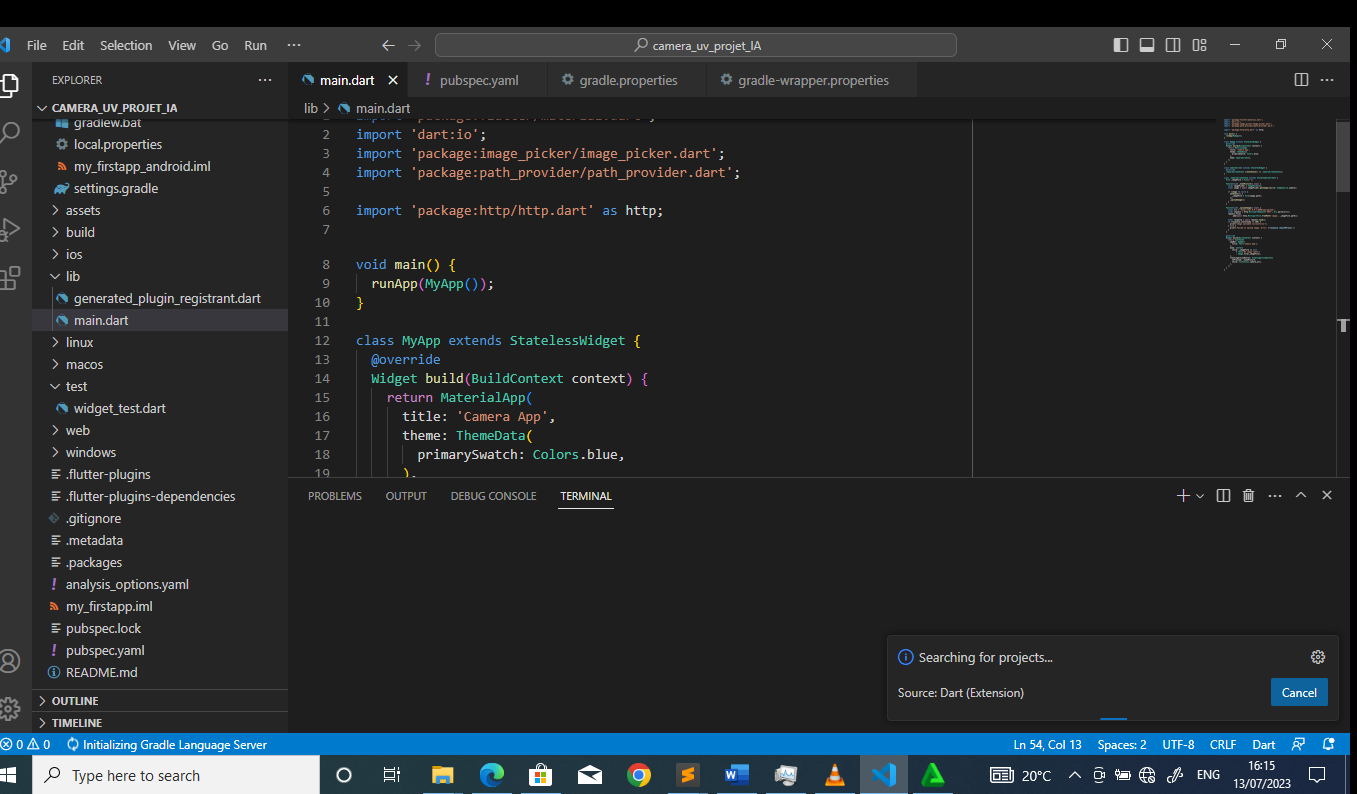


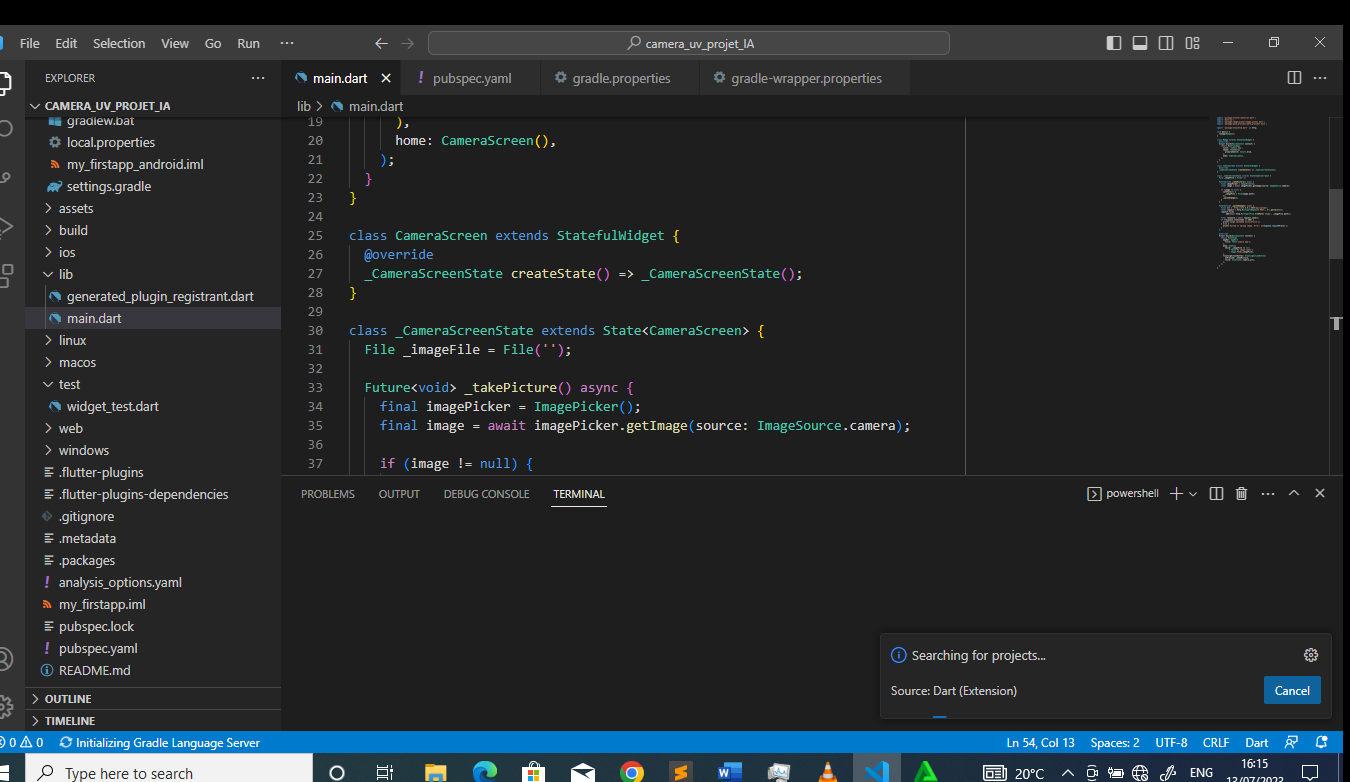
Serveur d’image

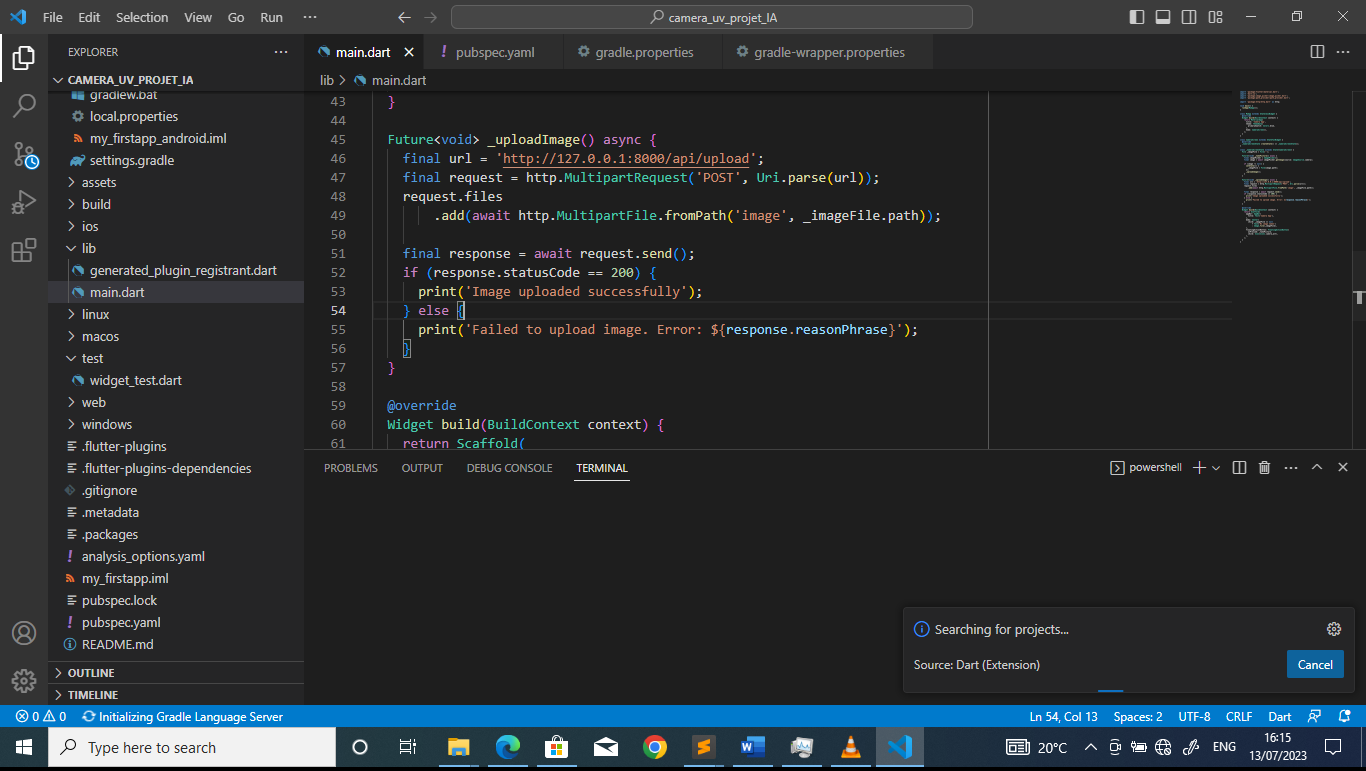




Code application mobile

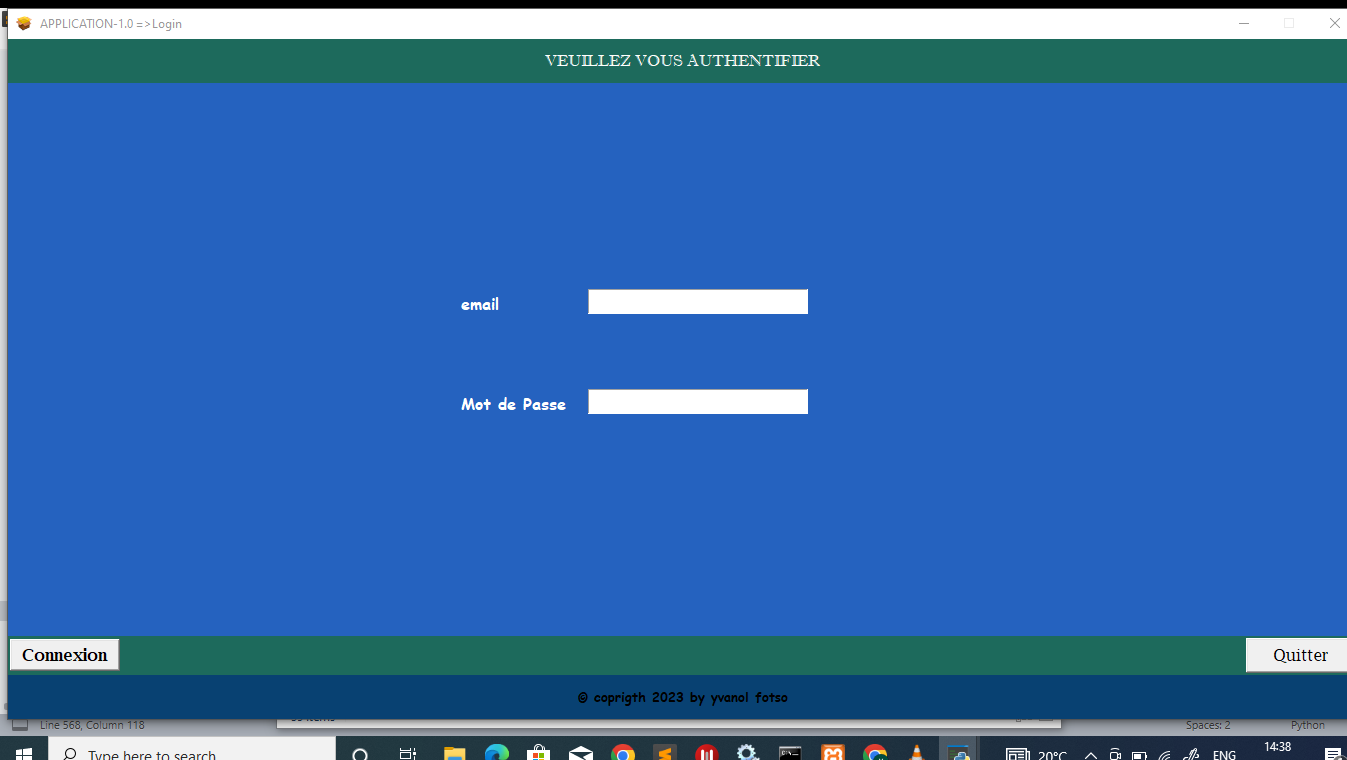




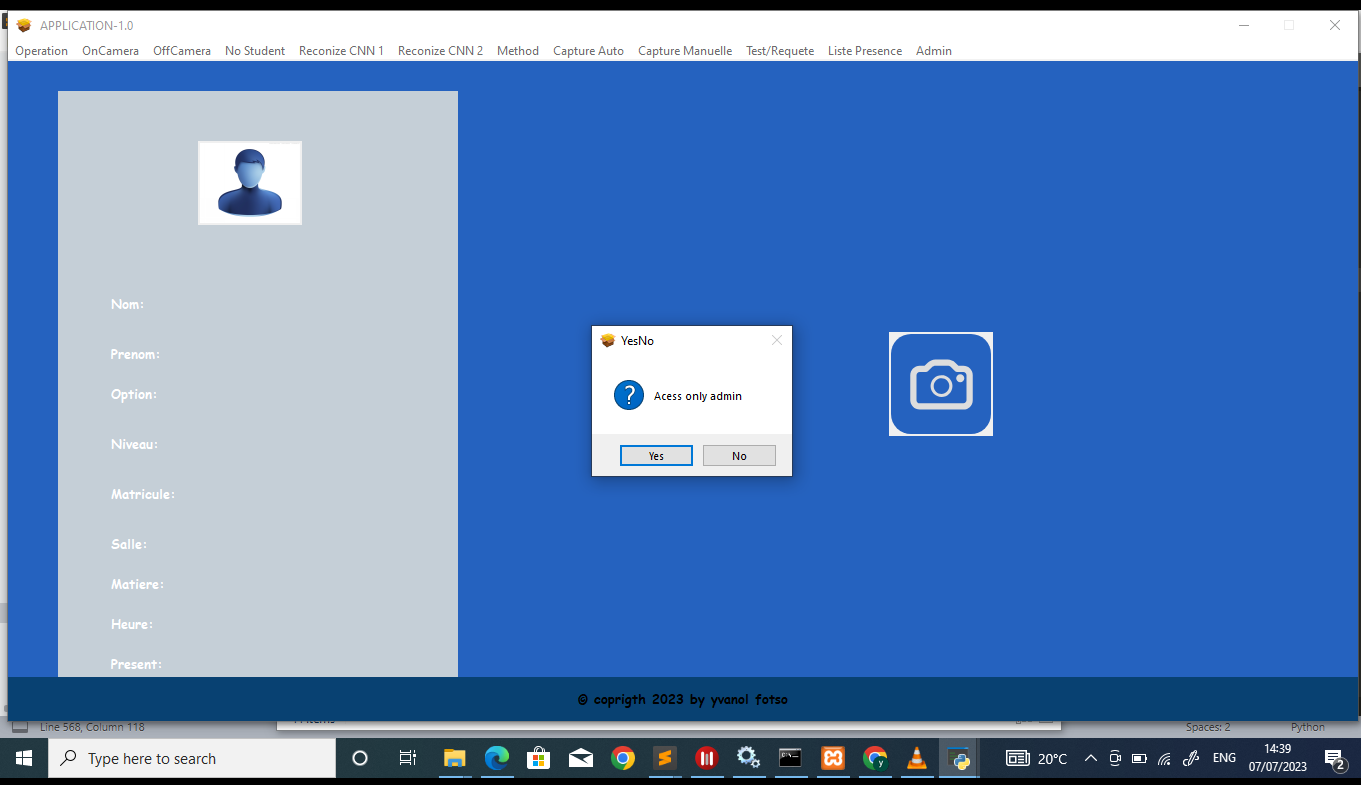


1. Nos interface

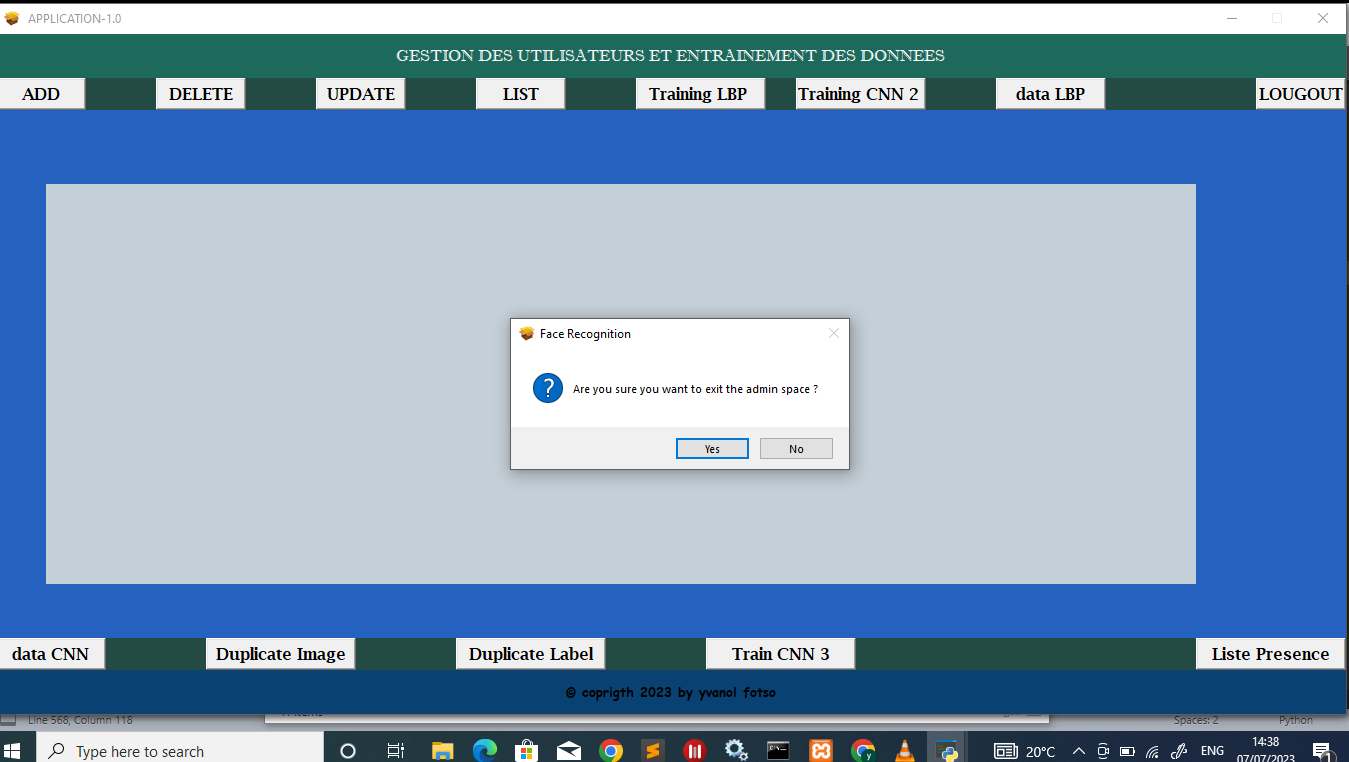
<<Login admin et login chef-salle>>



1. Interfaces chef- salle



1. Interface admin



1. Test

