

## Third Deliverable: High Level Design of the proposed solution

Isra Mariem Thabti , Hiba Abdelli BA/IT

During the process of enhancing Security with Facial Recognition Technology , our focus was on integrating “a facial recognition attendance system” into our security infrastructure. We dedicated our efforts to creating both a desktop application and a web-based application, each tailored with distinct features and components. Nevertheless, our primary objective remains centered around further advancing facial authentication technology.

**Here's a concise guide to enhance your comprehension of design and functionalities: (Please click the link to access the two videos we've attached.)**

[Simple Video Guide to Understanding the Design and Functionalities](#)

### **Desktop Application :**

#### **System Architecture:**

Client-Server architecture where the client is the application interface and the server performs the facial recognition tasks.

Components: Client Application, Server Application .

#### **Components:**

##### **Client Application:**

Responsible for capturing images from the webcam, displaying the webcam feed, and handling user interactions (login, register new user).

- User Interface: Provides buttons for login and registering a new user, webcam feed display for capturing images.
- Image Capture Module: Captures images from the webcam.
- User Interaction Module: Handles user interactions and triggers actions like login and register new users.

##### **Server Application:**

Responsible for processing images, detecting faces, computing face encodings, storing/retrieving face encodings, and performing face recognition.

- **Facial Detection Module:** Detects faces in captured images.

- **Facial Encoding Module:** Computes face encodings for detected faces.
- **Database Module:** Stores and retrieves face encodings along with user identities.

## **Working Flows:**

### **Login Flow:**

- The User clicks the login button.
- The Client captures an image from the webcam.
- The Client sends the captured image to the server.
- The Server detects faces in the image, computes face encodings.
- The Server compares computed face encodings with stored face encodings.
- If a match is found, the user is logged in; otherwise, an error message is displayed.

### **Register New User Flow:**

- The User clicks the register new user button.
- The Client captures an image from the webcam.
- The Client sends the captured image to the server.
- The Server detects faces in the image, computes face encodings.
- The Server stores the computed face encoding along with the user identity.
- A Confirmation message is displayed to the user.

## **Roles:**

1. **User:** Interacts with the client application to perform actions like login and register new user.
2. **Client Application:** Captures images, handles user interactions, and communicates with the server.
3. **Server Application:** Processes images, performs facial detection and recognition, stores/retrieves face encodings, and communicates with the client.

## **Functions and Related Steps:**

1. **Login Function:**
  - Capture image from the webcam.
  - Send the image to the server.
  - The Server processes the image: detects faces, computes face encodings.
  - The Server compares face encodings with stored encodings.
  - The Server sends the result (success or failure) to the client.
  - The Client displays a success message or an error message.
2. **Register New User Function:**
  - Capture image from the webcam.
  - Send the image to the server.

- The Server processes the image: detects faces, computes face encodings.
- The Server stores the computed face encoding along with the user identity.
- The Server sends a confirmation message to the client.
- The Client displays a success message.

### **Data Exchanged:**

1. **Image Data:**
  - Captured images are exchanged between the client and server for processing.
2. **Face Encodings:**
  - Face encodings (numerical codes representing facial features) are computed by the server and stored in the database.
  - During login, face encodings of captured images are compared with stored encodings.
3. **User Identity:**
  - User identity (username or unique identifier) is associated with the stored face encodings in the database.
4. **Messages:**
  - Success or error messages are exchanged between the client and server to inform the user about the login or registration outcome.

### **Web Application:**

#### **Components:**

- **Flask Web Application:** Acts as the main interface for users to interact with the attendance system.
- **OpenCV (cv2) :** Used for capturing video from the webcam and performing face detection and recognition.
- **Machine Learning Model (KNN):** Trained on facial images to recognize faces.
- **HTML Templates:** Used for rendering the web pages with attendance information and user interface elements.
- **CSV Files:** Store attendance records in a structured format.
- **Directories:** Used for organizing user images and other resources.

#### **Roles:**

- **Users:** Interact with the web application to perform tasks such as taking attendance, adding new users, and viewing attendance records.
- **System:** Handles the processing of webcam input, face detection and recognition, attendance management, and user interface rendering.

#### **Data Flow:**

1. **Webcam Input:** The system captures video frames from the webcam.
2. **Face Detection:** OpenCV's face detection algorithm locates faces in the video frames.
3. **Face Recognition:** The detected faces are passed to the trained machine learning model for recognition.
4. **Attendance Management:** Upon successful recognition, the system logs the attendance of the recognized person.
5. **Web Interface:** The attendance records are displayed on the web interface for users to view.

## **Working Flows:**

- **Adding New User:**
  - User provides a name and ID for the new user.
  - The system captures multiple images of the new user's face using the webcam.
  - The captured images are saved in the appropriate directory.
  - The machine learning model is trained using the new data.
- **Taking Attendance:**
  - User initiates the attendance process through the web interface.
  - The system captures video frames from the webcam.
  - Faces are detected in the frames, and recognition is performed using the trained model.
  - If a recognized face is identified, their attendance is logged.
- **Viewing Attendance Records:**
  - Users can access the web interface to view attendance records.
  - The system retrieves attendance data from CSV files.
  - HTML templates are used to render the attendance records on the web page.

## **Exchanged Messages/Data:**

### **Input:**

- User input via web interface (e.g., adding new user details).
- Video frames captured from the webcam.

### **Output:**

- Attendance records displayed on the web interface.

### **Data:**

- Images of registered users' faces stored in directories.
- Attendance records stored in CSV files.
- Trained machine learning model for face recognition.

