**Nhóm 3 CNPM 5  
  
I. Yêu cầu phần mềm**

**1. Yêu cầu chức năng**

**Quản lý người dùng**

* Đăng ký và đăng nhập: Cho phép người dùng tạo tài khoản và đăng nhập vào hệ thống.
* Phân quyền người dùng: Phân chia thành các nhóm quyền như Admin, Người dùng thông thường.
* Quản lý thông tin cá nhân: Cho phép người dùng cập nhật thông tin cá nhân.

**Chức năng nhận diện khuôn mặt**

* Tải ảnh lên: Hệ thống nhận diện các đối tượng khuôn mặt trong ảnh tải lên từ thiết bị.
* Chụp ảnh từ webcam: Cho phép chụp ảnh qua webcam và nhận diện các khuôn mặt trong ảnh.
* Video trực tiếp: Nhận diện khuôn mặt qua webcam theo thời gian thực, hiển thị kết quả ngay lập tức.
* Xử lý đa khuôn mặt: Có khả năng nhận diện nhiều khuôn mặt cùng lúc trong một ảnh/video.

**Quản lý dữ liệu**

* Lưu trữ lịch sử: Ghi lại thời gian, vị trí và kết quả nhận diện.
* Xuất báo cáo: Tạo báo cáo thống kê về các lần nhận diện theo thời gian hoặc người dùng.
* Tìm kiếm: Cho phép tìm kiếm trong dữ liệu lịch sử theo nhiều tiêu chí khác nhau.

**2. Yêu cầu phi chức năng**

**Hiệu suất**

* Thời gian phản hồi: Xử lý và phản hồi kết quả nhận diện trong vòng 2 giây cho ảnh tĩnh.
* Độ chính xác: Đạt tỷ lệ nhận diện chính xác tối thiểu 95% trong điều kiện ánh sáng bình thường.
* Khả năng xử lý: Hỗ trợ đồng thời nhiều người dùng, tối thiểu 50 người cùng lúc.

**Bảo mật**

* Mã hóa dữ liệu: Bảo vệ thông tin khuôn mặt được lưu trữ bằng mã hóa.
* Kiểm soát truy cập: Chỉ người dùng có quyền mới được truy cập vào dữ liệu nhạy cảm.
* Tuân thủ quy định: Đảm bảo việc thu thập và sử dụng dữ liệu tuân thủ các quy định về bảo vệ dữ liệu cá nhân.

**Khả năng sử dụng**

* Giao diện thân thiện: Thiết kế trực quan, dễ sử dụng cho cả người dùng mới.
* Hỗ trợ đa thiết bị: Tương thích với máy tính, điện thoại, tablet.
* Đa ngôn ngữ: Hỗ trợ tiếng Việt và tiếng Anh.

**Độ tin cậy**

* Sẵn sàng: Hệ thống hoạt động liên tục 24/7 với thời gian ngừng hoạt động không quá 0.1%.
* Khả năng phục hồi: Tự động sao lưu dữ liệu và khôi phục khi cần thiết.

**3. Giao diện người dùng**

**Trang chủ**

* Banner chính: Hiển thị giới thiệu tổng quan về hệ thống nhận diện khuôn mặt.
* Khu vực truy cập nhanh: Các nút dẫn đến chức năng Upload ảnh, Chụp ảnh, Video trực tiếp.
* Thông tin hướng dẫn: Hướng dẫn ngắn gọn về cách sử dụng các tính năng chính.

**Trang Giới thiệu**

* Thông tin chi tiết về hệ thống: Giải thích công nghệ sử dụng (Flask, OpenCV).
* Mục đích và ứng dụng: Mô tả các ứng dụng tiềm năng trong học tập, nghiên cứu và phát triển.
* Câu hỏi thường gặp: Giải đáp các thắc mắc phổ biến của người dùng.

**Trang Tính năng**

* Upload ảnh: Giao diện kéo thả hoặc chọn ảnh từ thiết bị, hiển thị kết quả nhận diện.
* Chụp ảnh: Kết nối với webcam để chụp và nhận diện đối tượng trong ảnh[[2]](#fn2).
* Video trực tiếp: Giao diện hiển thị video từ webcam và kết quả nhận diện theo thời gian thực.

**Trang Thư viện**

* Danh sách các ảnh đã xử lý: Hiển thị dạng lưới với thông tin thời gian và kết quả.
* Bộ lọc: Cho phép lọc theo thời gian, loại nhận diện.
* Chức năng chia sẻ: Cho phép chia sẻ kết quả qua email hoặc mạng xã hội.

**Trang Lịch sử**

* Bảng dữ liệu: Hiển thị lịch sử các lần sử dụng hệ thống.
* Biểu đồ thống kê: Trực quan hóa dữ liệu sử dụng theo thời gian.
* Chức năng xuất báo cáo: Cho phép xuất dữ liệu dưới dạng CSV, PDF.

**4. Đề xuất tài nguyên sử dụng**

**Ngôn ngữ lập trình & Frameworks**

* Python: Ngôn ngữ chính cho backend và xử lý hình ảnh.
* Flask: Framework web để xây dựng backend và API.
* HTML/CSS/JavaScript: Xây dựng giao diện người dùng.
* Bootstrap: Framework CSS cho thiết kế responsive.

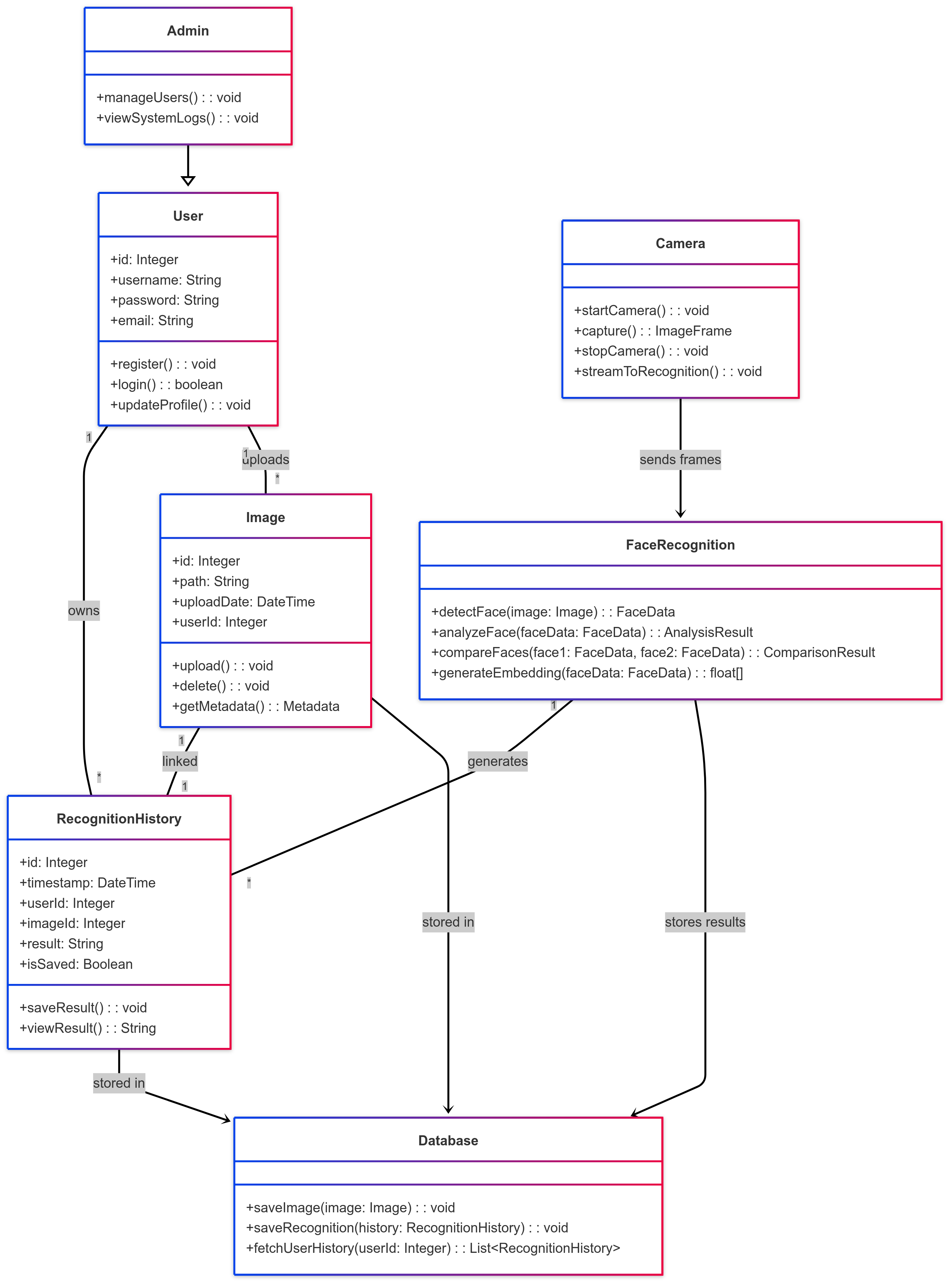
**Thư viện xử lý ảnh và AI**

* OpenCV: Thư viện xử lý hình ảnh và video.
* Dlib: Thư viện hỗ trợ nhận diện khuôn mặt chính xác[]](#fn5).
* TensorFlow/PyTorch: Các framework deep learning cho mô hình nhận diện nâng cao.
* FaceNet/OpenFace: Mô hình pre-trained cho nhận diện khuôn mặt[[6]](#fn6).

**Cơ sở dữ liệu**

* SQLite: Lưu trữ dữ liệu người dùng và lịch sử nhận diện cho hệ thống quy mô nhỏ[[5]](#fn5).
* MySQL/PostgreSQL: Cho hệ thống quy mô lớn hơn với nhiều người dùng.
* MongoDB: Lưu trữ dữ liệu phi cấu trúc như hình ảnh khuôn mặt.

**Yêu cầu phần cứng**

* Máy chủ: Bộ xử lý x86-64 (64-bit) tương thích, tốc độ tối thiểu 2GHz.
* RAM: Tối thiểu 512MB RAM trống cho ứng dụng[[7]](#fn7).
* Thiết bị đầu vào: Camera hoặc webcam hỗ trợ DirectShow (Windows) hoặc GStreamer (macOS/Linux).  
  **5. Mô hình hệ thống sử dụng UML**

**II. Thiết kế kiến trúc phần mềm**

**1. Tài nguyên sử dụng**

**Công nghệ Backend**

* Python 3.: Ngôn ngữ lập trình chính.
* Flask : Microframework để xây dựng RESTful API và ứng dụng web.
* WSGI Server: Gunicorn (Linux) hoặc Waitress (Windows) để triển khai ứng dụng Flask.
* JWT (JSON Web Tokens): Xác thực và ủy quyền người dùng.

**Công nghệ Frontend**

* HTML: Xây dựng giao diện người dùng.
* JavaScript : Logic phía client.
* Bootstrap : Framework CSS cho thiết kế responsive.
* AJAX: Giao tiếp không đồng bộ với backend.

**Thư viện xử lý hình ảnh và AI**

* OpenCV: Xử lý hình ảnh và video, phát hiện khuôn mặt[[1]](#fn1)[[5]](#fn5).
* Dlib : Định vị các điểm đặc trưng (facial landmarks) trên khuôn mặt.
* Face\_recognition: Thư viện Python wrapper cho Dlib, đơn giản hóa việc nhận diện.
* NumPy: Xử lý mảng và ma trận cho tính toán số học.
* SciPy: Xử lý tín hiệu và hình ảnh nâng cao.

**Cơ sở dữ liệu**

* SQLite: Cơ sở dữ liệu nhẹ cho hệ thống quy mô nhỏ[[5]](#fn5).
* SQLAlchemy: ORM (Object Relational Mapper) để tương tác với cơ sở dữ liệu.
* Alembic: Quản lý phiên bản cơ sở dữ liệu.

**Công cụ phát triển**

* Git: Quản lý mã nguồn.
* Visual Studio Code/PyCharm: IDE phát triển.
* Docker: Đóng gói và triển khai ứng dụng.
* Postman: Kiểm thử API.

**2. Các thành phần chính**

**Frontend (Giao diện người dùng)**

* **Trang chủ**: Giới thiệu hệ thống và truy cập nhanh các chức năng chính.
* **Module Upload Ảnh**: Giao diện kéo thả và tải lên hình ảnh.
* **Module Webcam**: Kết nối và điều khiển webcam, chụp ảnh.
* **Module Video trực tiếp**: Xử lý video từ webcam theo thời gian thực.
* **Module Lịch sử & Báo cáo**: Hiển thị và quản lý lịch sử nhận diện.
* **Module Quản lý tài khoản**: Quản lý thông tin cá nhân.

**Backend (Máy chủ xử lý)**

* **API Controller**: Xử lý các request từ frontend.
* **Authentication Service**: Quản lý đăng nhập, đăng ký và xác thực.
* **Image Processing Service**: Xử lý ảnh trước khi nhận diện.
* **Face Detection Service**: Tìm vị trí khuôn mặt trong ảnh/video.
* **Face Recognition Service**: Nhận diện và so sánh khuôn mặt.
* **History Service**: Quản lý lịch sử nhận diện.
* **Database Service**: Tương tác với cơ sở dữ liệu.

**Module nhận diện khuôn mặt**

* **Face Detection**: Phát hiện vị trí khuôn mặt trong ảnh/video .
* **Face Alignment**: Căn chỉnh khuôn mặt để chuẩn hóa góc nhìn.
* **Feature Extraction**: Trích xuất đặc trưng từ khuôn mặt .
* **Comparison Engine**: So sánh đặc trưng với dữ liệu có sẵn để nhận diện.

**Cơ sở dữ liệu**

* **User DB**: Lưu trữ thông tin người dùng.
* **Face DB**: Lưu trữ đặc trưng khuôn mặt (embedding vectors).
* **Image DB**: Lưu trữ metadata của ảnh (không lưu ảnh gốc nếu không cần thiết).
* **History DB**: Lưu trữ lịch sử nhận diện và kết quả.

**3. Các thành phần giao tiếp với nhau**

**Luồng tương tác Upload ảnh**

1. **Người dùng → Frontend**: Người dùng tải ảnh lên thông qua giao diện web.
2. **Frontend → Backend API**: Frontend gửi ảnh và yêu cầu nhận diện đến API.
3. **Backend API → Image Processing Service**: API chuyển ảnh đến dịch vụ xử lý ảnh để tiền xử lý.
4. **Image Processing Service → Face Detection Service**: Ảnh đã xử lý được chuyển đến dịch vụ phát hiện khuôn mặt.
5. **Face Detection Service → Face Recognition Service**: Khuôn mặt đã phát hiện được chuyển đến dịch vụ nhận diện.
6. **Face Recognition Service → Database Service**: Kết quả nhận diện được lưu vào cơ sở dữ liệu.
7. **Backend API → Frontend**: Kết quả nhận diện được trả về cho giao diện người dùng.
8. **Frontend → Người dùng**: Hiển thị kết quả nhận diện cho người dùng.

**Luồng tương tác Video trực tiếp**

1. **Người dùng → Frontend**: Người dùng yêu cầu kích hoạt webcam.
2. **Frontend → Camera API**: Frontend kích hoạt webcam thông qua Camera API của trình duyệt.
3. **Camera API → Frontend**: Luồng video được truyền về frontend.
4. **Frontend → Backend API**: Frontend gửi từng frame video đến backend API.
5. **Backend API → Face Detection & Recognition**: Các frame được xử lý để phát hiện và nhận diện khuôn mặt.
6. **Backend API → Frontend**: Kết quả nhận diện được trả về liên tục.
7. **Frontend → Người dùng**: Hiển thị kết quả nhận diện trực tiếp trên giao diện.

**Luồng xác thực người dùng**

1. **Người dùng → Frontend**: Người dùng nhập thông tin đăng nhập.
2. **Frontend → Authentication Service**: Gửi thông tin đăng nhập để xác thực.
3. **Authentication Service → Database Service**: Kiểm tra thông tin đăng nhập trong cơ sở dữ liệu.
4. **Authentication Service → Frontend**: Trả về token xác thực nếu đăng nhập thành công.
5. **Frontend → Backend API**: Gửi token xác thực kèm theo mỗi request.

**4. Công dụng**

**Ứng dụng trong an ninh và giám sát**

* **Kiểm soát ra vào**: Xác thực danh tính người ra vào các khu vực hạn chế.
* **Giám sát an ninh**: Phát hiện và nhận diện người trong các hệ thống camera an ninh.
* **Phát hiện xâm nhập**: Cảnh báo khi phát hiện người lạ trong khu vực giám sát.

**Ứng dụng trong giáo dục và nghiên cứu**

* **Điểm danh tự động**: Tự động ghi nhận sự hiện diện của học sinh/sinh viên trong lớp học.
* **Phân tích hành vi**: Nghiên cứu biểu hiện khuôn mặt trong các bối cảnh giáo dục.
* **Nền tảng học tập**: Cung cấp công cụ thực hành cho sinh viên ngành CNTT, AI.

**Ứng dụng trong doanh nghiệp**

* **Chấm công tự động**: Ghi nhận thời gian đến/rời nơi làm việc của nhân viên.
* **Xác thực khách hàng**: Nhận diện khách hàng VIP hoặc người cần được phục vụ đặc biệt.
* **Tương tác khách hàng**: Cá nhân hóa trải nghiệm dựa trên nhận diện khuôn mặt.

**Ứng dụng trong sức khỏe và y tế**

* **Theo dõi bệnh nhân**: Giám sát bệnh nhân không cần thiết bị gắn trên người.
* **Phân tích biểu hiện**: Hỗ trợ chẩn đoán các vấn đề liên quan đến biểu hiện khuôn mặt.
* **Kiểm soát truy cập**: Hạn chế truy cập vào các khu vực y tế nhạy cảm.

**5. Sơ đồ kiến trúc dùng UML**

**Component Diagram**

* **Thành phần chính**:
  + Web Client: Giao diện người dùng.
  + API Gateway: Điểm vào chính cho tất cả các request.
  + Authentication Service: Xử lý xác thực và ủy quyền.
  + Image Processing Service: Xử lý và chuẩn bị hình ảnh.
  + Face Detection Service: Phát hiện khuôn mặt trong hình ảnh.
  + Face Recognition Service: Nhận diện khuôn mặt đã phát hiện.
  + Database Service: Quản lý dữ liệu.
* **Mối quan hệ**:
  + Web Client → API Gateway: Gửi request và nhận response.
  + API Gateway → Authentication Service: Xác thực request.
  + API Gateway → Image Processing/Face Detection/Face Recognition: Chuyển tiếp request đã xác thực.
  + Các service → Database Service: Truy xuất và lưu trữ dữ liệu.

**Deployment Diagram**

* **Các node**:
  + Client Device: Máy tính/điện thoại người dùng.
  + Web Server: Cung cấp trang web và API.
  + Application Server: Xử lý logic nghiệp vụ và nhận diện khuôn mặt.
  + Database Server: Lưu trữ dữ liệu.
* **Các artifact**:
  + Web UI: Giao diện người dùng (HTML, CSS, JS).
  + API Service: RESTful API phục vụ frontend.
  + Face Recognition Engine: Module xử lý nhận diện khuôn mặt.
  + Database: Cơ sở dữ liệu lưu trữ thông tin.
* **Mối quan hệ**:
  + Client Device → Web Server: Kết nối HTTP/HTTPS.
  + Web Server → Application Server: Kết nối TCP/IP.
  + Application Server → Database Server: Kết nối SQL.

**State Diagram**

* **Các trạng thái của quy trình nhận diện**:
  + Idle: Hệ thống chờ đầu vào.
  + Processing Image: Đang xử lý hình ảnh đầu vào.
  + Detecting Faces: Đang phát hiện khuôn mặt trong hình ảnh.
  + Extracting Features: Đang trích xuất đặc trưng từ khuôn mặt.
  + Comparing: Đang so sánh với cơ sở dữ liệu.
  + Displaying Results: Đang hiển thị kết quả.
  + Error: Xảy ra lỗi trong quá trình xử lý.
* **Các sự kiện chuyển trạng thái**:
  + Idle → Processing Image: Người dùng tải ảnh lên/kích hoạt webcam.
  + Processing Image → Detecting Faces: Ảnh đã được tiền xử lý.
  + Detecting Faces → Extracting Features: Khuôn mặt đã được phát hiện.
  + Extracting Features → Comparing: Đặc trưng đã được trích xuất.
  + Comparing → Displaying Results: So sánh hoàn tất.
  + Bất kỳ trạng thái → Error: Xảy ra lỗi.
  + Error/Displaying Results → Idle: Reset hệ thống.