

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BÁO CÁO ĐỒ ÁN ĐA NGHÀNH

HỆ THỐNG IOT XÁC THỰC KHUÔN MẶT ĐIỂM DANH VÀ QUẢN LÝ NHÂN VIÊN

Giảng viên hướng dẫn: Nguyễn Thị Ái Thảo

Mã số nhóm: L01_HTTT_8

Mã môn: CO3111

STT	Họ tên SV	MSSV	Tên lớp	Ngành
1	Bùi Quang Bằng	2012681	MT20KT06	Kỹ thuật máy tính
2	Tạ Đình Tiến	2110583	MT21KHTN	Khoa học máy tính
3	Mai Hoàng Danh	2110896	MT21KH01	Khoa học máy tính
4	Cao Đức Dương	2110971	MT21KH02	Khoa học máy tính
5	Trương Đức Dũng	2113080	MT21KH10	Khoa học máy tính
6	Phạm Bá Hoàng	2113414	MT21KH10	Khoa học máy tính

Tp. Hồ Chí Minh, Tháng 5/2024



Mục lục

1 Phân công	3
2 Giới thiệu đề tài	3
3 Các yêu cầu	4
3.1 Yêu cầu chức năng	4
3.1.1 Hệ thống IoT	4
3.1.2 Ứng dụng quản lý	5
3.2 Yêu cầu phi chức năng	5
3.2.1 Hệ thống IoT	5
3.2.2 Ứng dụng quản lý	5
4 Thiết bị	6
4.1 MicroBit V2	6
4.2 Các thiết bị ngoại vi	7
5 IOT Gateway và Hệ thống thiết bị	8
5.1 IOT Gateway	8
5.1.1 Kết nối tới hệ thống thiết bị	9
5.1.2 Kết nối tới IOT Server	10
5.1.3 Hệ thống thiết bị	10
6 Use case	10
6.1 Xác thực khuôn mặt và điều khiển cửa	10
6.2 Theo dõi check-in và check-outout	11
6.3 Thông báo và nhắc nhở nhân viên	12
6.4 Xem thống kê và báo cáo	13
6.5 Tài khoản	14
6.6 Giao tiếp và phản hồi	16
7 Mockup design	18
7.1 Login Screen	19
7.2 Home Screen	20
7.3 Control Panel Screen	22
7.4 Profile Screen	23
7.5 Attendance Track Screen	24
7.6 My Attendance Track Screen	25
7.7 Dashboard Screen	26
7.8 Employee Registration Screen	27
7.9 Create Face Model Screen	28
7.10 Feedback/Report Screen	29
7.11 OTP Request Screen	30
7.12 Logout	31
7.13 Face Recognition UI	32
8 Architecture	32
8.1 Backend System	33
8.2 IoT Devices	33
8.3 Frontend	34
8.4 Workflow	34
9 Class diagram	35



10 Database	37
10.1 Thiết kế EERD	37
10.2 Thiết kế logic - Ánh xạ lược đồ thực thể, mối liên kết mở rộng	37
10.2.1 Ánh xạ dạng bảng	37
10.2.2 Ánh xạ dạng viết	37
11 Technology	38
11.1 React Native	38
11.2 Expo Go	39
11.3 ExpressJs	39
11.4 Face-api.js	39
12 Kết luận	40
Tài liệu tham khảo	40

1 Phân công

Họ và tên	MSSV	Công việc	Mức độ
Bùi Quang Bằng	2012681	1. Thiết kế hệ thống tổng quát 2. Lập trình IOT Gateway để kết nối hệ thống thiết bị và Server 3. Lập trình hệ thống thiết bị phần cứng 4. Thiết kế IOT server Adafruit	100%
Tạ Dinh Tiến	2110583	1. Hiện thực backend cho ứng dụng quản lý. 2. Tìm hiểu thư viện speakeasy, jwt hỗ trợ việc xác thực OTP và xác thực người dùng.	100%
Mai Hoàng Danh	2110896	1. Thiết kế mockup cho ứng dụng. 2. Hiện thực frontend cho ứng dụng.	100%
Cao Đức Dương	2110971	1. Thiết kế database. 2. Hiện thực procedure trong database hỗ trợ backend API 3. Thiết kế class diagram.	100%
Trương Đức Dũng	2113080	1. Thiết kế mockup cho ứng dụng 2. Hiện thực frontend cho ứng dụng	100%
Phạm Bá Hoàng	2113414	1. Hiện thực backend cho hệ thống IoT và ứng dụng quản lý. 2. Tìm hiểu, áp dụng thư viện face-api.js và hiện thực để xác thực khuôn mặt.	100%

2 Giới thiệu đề tài

Trong bối cảnh hiện nay, an ninh và quản lý nhân sự hiệu quả là những yếu tố then chốt đối với sự phát triển bền vững của các công ty. Việc kiểm soát ra vào và điểm danh nhân viên bằng phương pháp thủ công không chỉ tốn kém thời gian mà còn không đảm bảo độ chính xác và an toàn cao. Những bất cập này đã thúc đẩy nhu cầu tìm kiếm một giải pháp hiện đại và hiệu quả hơn.

Vì vậy, đề tài "Hệ thống xác thực khuôn mặt cho công ty" được lựa chọn nhằm giải quyết các vấn đề nêu trên.

Hệ thống này không chỉ giúp tự động hóa quy trình điểm danh và kiểm soát ra vào, mà còn nâng cao mức độ bảo mật bằng cách xác thực danh tính nhân viên qua công nghệ nhận diện khuôn mặt. Ngoài ra, một ứng dụng di động dành cho cả nhân viên và quản lý sẽ được phát triển để tận dụng dữ liệu từ hệ thống xác thực, giúp theo dõi thời gian làm việc của nhân viên, gửi thông báo check-in và check-out, thống kê thời gian có mặt, ngày nghỉ, và báo vắng. Việc áp dụng hệ thống này không chỉ nâng cao hiệu quả quản lý nhân sự mà còn góp phần xây dựng một môi trường làm việc an toàn và hiện đại.



Hình 1: Hệ thống xác thực khuôn mặt cho công ty

3 Các yêu cầu

3.1 Yêu cầu chức năng

3.1.1 Hệ thống IoT

- Hệ thống lấy input từ camera ngoài cổng công ty để nhận diện khuôn mặt nhân viên.
- Hệ thống lấy input thông qua các cảm biến nhiệt độ, độ ẩm cập nhật thông tin lên màn hình, gửi và nhận thông qua broker Adafruit IO thông qua MQTT protocol.
- Các bước xác thực khuôn mặt:

- B1: Người dùng tới máy để điểm danh.
- B2: Đưa khuôn mặt vào màn hình và xác thực.

Hệ thống sẽ kiểm tra với mẫu khuôn mặt đã được train thông qua model có sẵn trong cơ sở dữ liệu:

- * Tìm kiếm khuôn mặt thành công: Hệ thống phải hiển thị thông tin nhân viên khớp với hệ thống để người quét đối chiếu (Hệ thống phải có một màn hình nhỏ hiển thị tên/MSNV).
- * Không tìm thấy mẫu khuôn mặt phù hợp: Hệ thống hiển thị "No one".

Màn hình hệ thống sẽ bao gồm 2 nút YES/NO:

- * Trong trường hợp xác thực đúng với khuôn mặt của nhân viên, chọn YES để xác thực và cập nhật trạng thái điểm danh.
- * Trong trường hợp xác thực sai khuôn mặt của người dùng, chọn NO để hệ thống reset và xác thực lại.
- * Trong trường hợp vẫn xác thực sai, nhân viên có thể chọn hình thức xác thực qua OTP để điểm danh thông qua app.

3.1.2 Ứng dụng quản lý

- Xác thực để đăng nhập.
- Ứng dụng mobile cho phép quản lý điểm danh nhân viên.
- Thông báo thời điểm check-in và check-out cho nhân viên. Dữ liệu gửi tới nhân viên có dạng:
 - Họ tên.
 - Thời gian.
 - Vị trí.
 - Giá trị khi điểm danh (check in, check out, sớm, muộn,...)
 - Chức vụ.
- Hệ thống được thiết lập thời gian check-in check-out, dựa vào người quét bên trong hay bên ngoài để xác thực.
- Chức năng thống kê thời gian làm việc, thời gian check-in trung bình, số ngày làm và số ngày nghỉ.
- Thông báo nhắc nhở cho nhân viên khi đến muộn.
- Nhân viên có thể cập nhật thông tin cơ bản cho tài khoản của mình (avatar, mật khẩu,...).
- Chức năng cho nhân viên gửi form, xem form cho quản lý khi có vấn đề bất cập.
- Nhân viên có thể chọn xác thực bằng OTP khi hệ thống bị lỗi hoặc xác thực sai.
- Quản lý có thể đăng ký tài khoản cho nhân viên, cập nhật hình ảnh của nhân viên.
- Quản lý nhận form và trả lời form cho nhân viên.
- Quản lý có thể giám sát các dữ liệu check in, check out từ thiết bị IoT.

3.2 Yêu cầu phi chức năng

3.2.1 Hệ thống IoT

- Chạy 24/7 với thời gian ngừng hoạt động theo lịch trình tối thiểu và tốt nhất là không có thời gian ngừng hoạt động đột xuất.
- Kết nối ổn định, dữ liệu nhận được chính xác, sai lệch thấp.
- Thời gian trễ so với thời gian thực thấp.

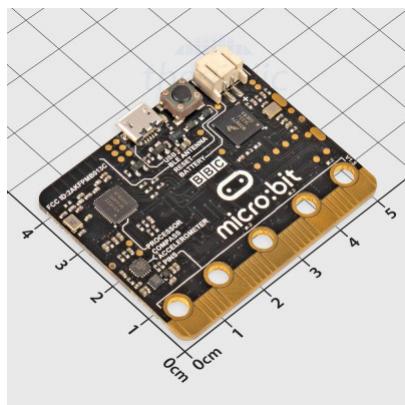
3.2.2 Ứng dụng quản lý

- Bảo mật cao cho dữ liệu nhân viên.
- Tính ổn định và đáng tin cậy của hệ thống.
- Tính năng tương thích trên nhiều thiết bị di động và hệ điều hành.
- Giao diện thân thiện và dễ sử dụng.
- Hiệu suất cao để đáp ứng nhu cầu thời gian thực.

4 Thiết bị

4.1 MicroBit V2

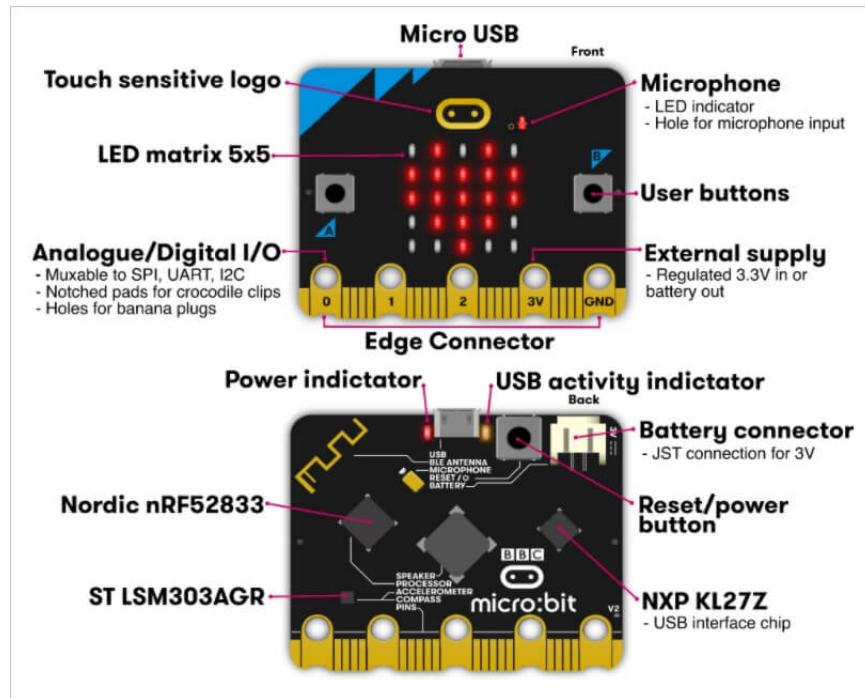
Kit Phát Triển BBC MicroBit là một hệ thống nhúng dựa trên ARM phần cứng mã nguồn mở được thiết kế bởi BBC để sử dụng trong giáo dục máy tính. BBC micro bit là một máy vi tính cầm tay, có thể lập trình được, có thể được sử dụng cho tất cả các loại sáng tạo thú vị, từ robot đến nhạc cụ - khả năng là vô tận.



Thông số kỹ thuật

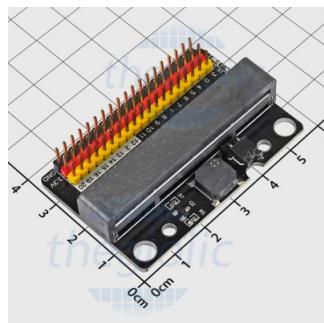
- CPU M0 Cortex ARM 32-bit
- Bo mạch lập trình bao gồm 25 đèn LED đỏ
- Cổng Micro USB, dùng để cấp nguồn cho board
- Ngõ ra (I/O) để kết nối với ngoại vi
- Cho phép kết nối Bluetooth
- Điện áp ngõ ra: 3VDC
- Một cổng 20-pin pitch 1.27mm
- Hai nút có thể hoạt động như một bộ điều khiển trò chơi
- Một cổng kết nối pin AAA
- Một giao tiếp kẽ
- Một la bàn

Cấu tạo microbit

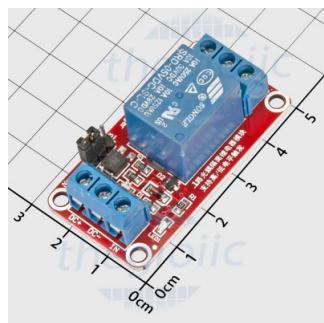


4.2 Các thiết bị ngoại vi

Các thiết bị ngoại vi sử dụng trong đồ án



Hình 2: Mạch mở rộng cho MicroBit



Hình 3: Module relay



Hình 4: Cảm biến hồng ngoại



Hình 5: Cảm biến nhiệt độ độ ẩm

5 IOT Gateway và Hệ thống thiết bị

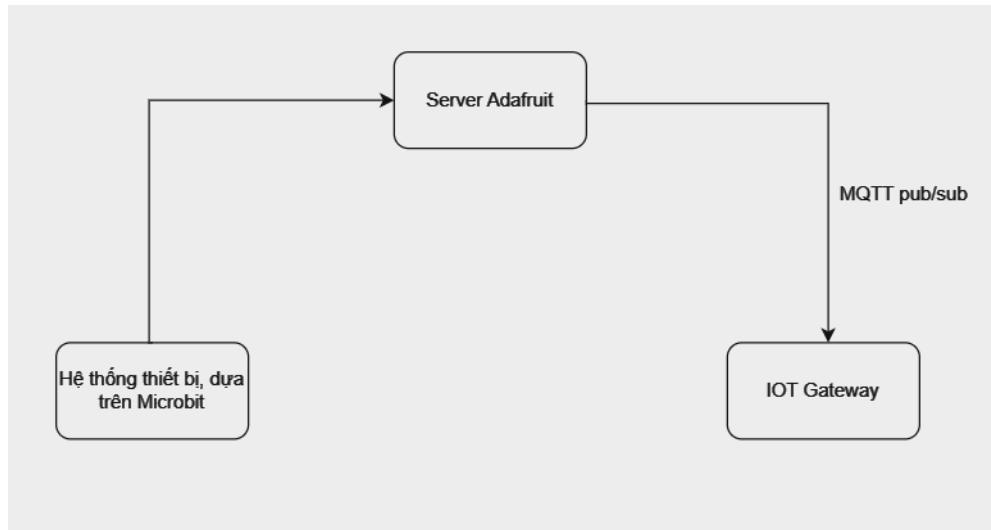
5.1 IOT Gateway

IOT Gateway là một thiết bị trung gian kết nối các thiết bị IoT (Internet of Things) với mạng internet. Nó hoạt động như một cầu nối, cho phép các thiết bị không có khả năng kết nối trực tiếp với internet có thể truyền và nhận dữ liệu qua mạng internet. Các chức năng chính của IoT Gateway bao gồm:

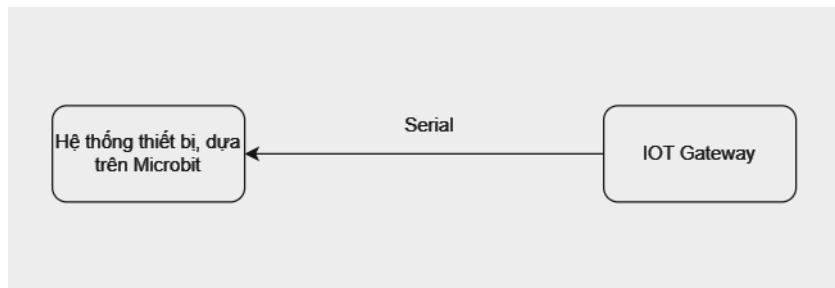
- **Kết nối và giao tiếp:** Kết nối các thiết bị IoT với nhau và với mạng internet thông qua các giao thức và công nghệ kết nối như WiFi, Bluetooth, Zigbee, Z-Wave, và nhiều loại khác
- **Lưu trữ và xử lý dữ liệu:** Lưu trữ dữ liệu tạm thời trước khi truyền lên đám mây hoặc máy chủ, và có thể xử lý dữ liệu để giảm tải cho các hệ thống trung tâm
- **Bảo mật:** Đảm bảo an toàn cho dữ liệu truyền qua bằng cách mã hóa và thực hiện các biện pháp bảo mật khác

IOT Gateway thường được sử dụng trong các hệ thống IoT để tối ưu hóa hiệu suất, bảo mật và khả năng mở rộng của mạng IoT.

5.1.1 Kết nối tới hệ thống thiết bị



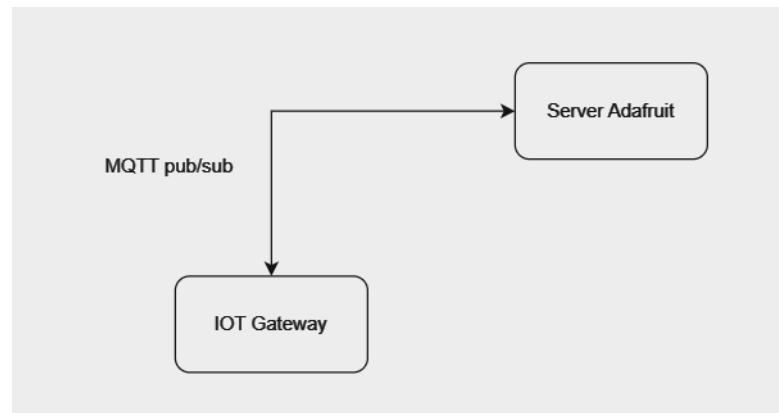
Hình 6: Hệ thống kết nối không sử dụng Serial



Hình 7: Hệ thống kết nối sử dụng Serial (được sử dụng trong đồ án)

IOT Gateway sẽ kết nối với hệ thống thiết bị điều khiển cửa thông qua kết nối Serial UART. Cách kết nối này sẽ đảm bảo tốc độ xử lý nhanh và chính xác vì là kết nối ngắn và truyền tín hiệu qua dây dẫn hơn là kiến trúc bộ xử lý trung tâm kết nối đến IOT Server vì kết nối này là kết nối qua Internet, sẽ có rất nhiều yếu tố ảnh hưởng đến tốc độ giao tiếp giữa 2 host.

5.1.2 Kết nối tới IOT Server



Hình 8: Kết nối MQTT thông qua mạng Internet

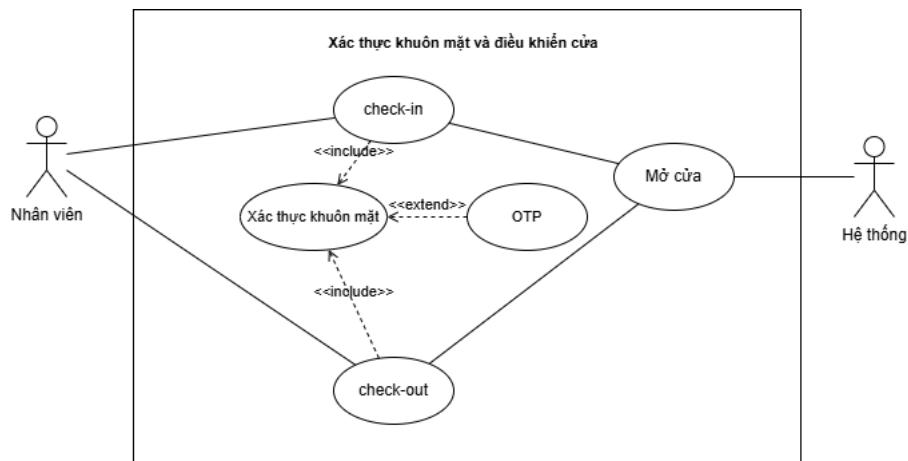
5.1.3 Hệ thống thiết bị

Hệ thống sẽ được để chế độ mặc định là cửa đóng. Khi có tín hiệu checkin hoặc checkout của hệ thống và tín hiệu hồng ngoại cho thấy có người ở khu vực checkin hoặc checkout thông qua cảm biến hồng ngoại, thì cửa sẽ mở cho nhân viên qua và đóng lại sau 5 giây. Khi mở ra hay đóng lại thì hệ thống sẽ tự động gửi lại tín hiệu cho IOT Gateway để publish lên IOT Server để kiểm soát và theo dõi từ xa.

Ngoài ra, khu vực checkin và checkout luôn được cập nhật nhiệt độ và độ ẩm hiện tại từ cảm biến DHT22. Hệ thống lấy dữ liệu từ IOT Server được cảm biến DHT22 publish dữ liệu thông qua IOT Gateway.

6 Use case

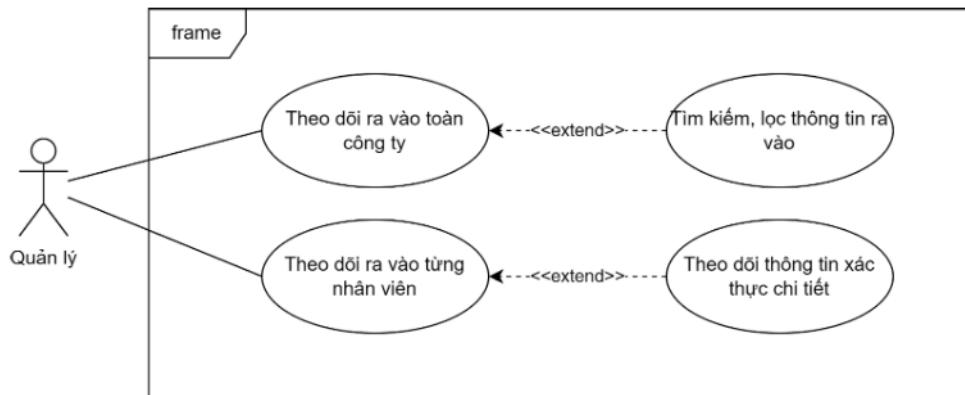
6.1 Xác thực khuôn mặt và điều khiển cửa



Bảng 1: Use case «Xác thực khuôn mặt và điều khiển cửa»

Use case	Descriptions
ID	UC - 01
Tên	Xác thực khuôn mặt và điều khiển cửa
Mô tả	Mở cửa cho người dùng bằng cách xác thực khuôn mặt
Actor	Nhân viên
Điều kiện tiên quyết	Nhân viên tiếp cận hệ thống điểm danh và hệ thống lấy dữ liệu từ camera.
Điều kiện sau	Cửa sẽ được mở cho người dùng
Dòng sự kiện chính	Hệ thống nhận diện khuôn mặt và kiểm tra danh sách nhân viên. Bước 1: Người dùng tiếp cận hệ thống. Camera sẽ nhận diện khuôn mặt, xác thực dựa trên cơ sở dữ liệu được lưu trong hệ thống. Bước 2: Hệ thống sẽ tồn một khoảng thời gian nhất định để phản hồi. Nếu nhân viên có dữ liệu hình ảnh trong cơ sở dữ liệu. Cửa sẽ tự động mở và điểm danh cho nhân viên.
Dòng sự kiện thay thế	Nếu không thành công, hệ thống thông báo lỗi và cung cấp phương tiện xác thực thay thế (OTP).

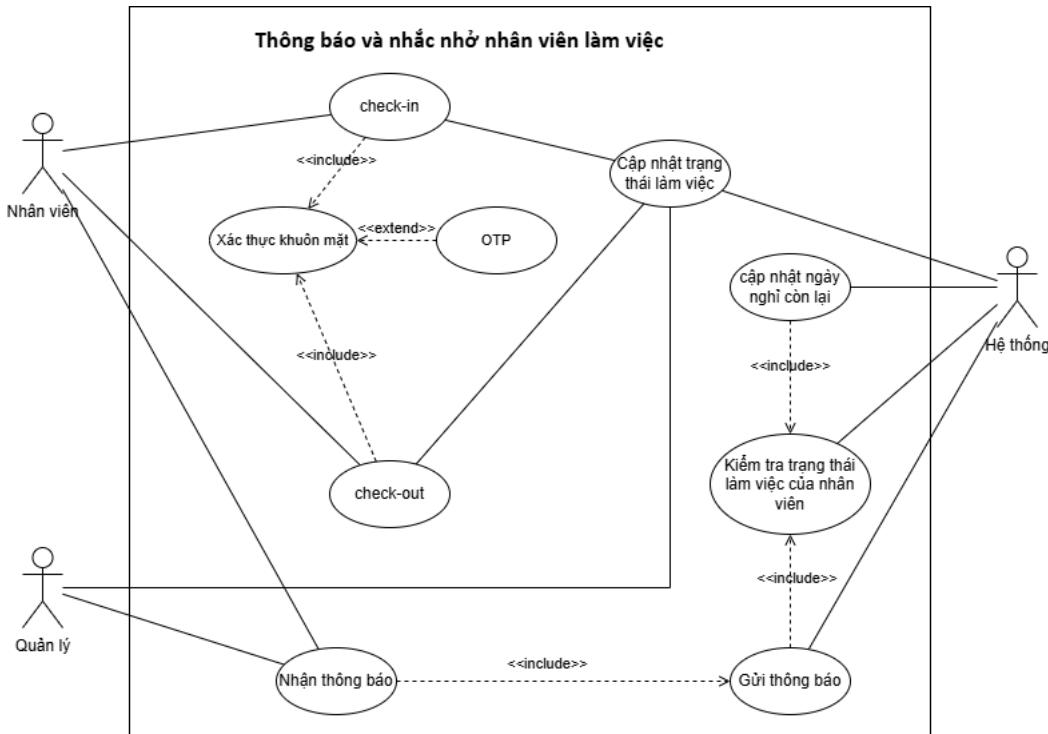
6.2 Theo dõi check-in và check-out



Bảng 2: Use case «Theo dõi check-in và check-out»

Use case	Descriptions
ID	UC - 02
Tên	Theo dõi thông tin check-in và check-out
Mô tả	Quản lý theo dõi thời điểm check-in và check-out của các nhân viên trong ngày
Actor	Quản lý
Điều kiện tiên quyết	1. Hệ thống hoạt động ổn định. 2. Quản lý đã đăng nhập.
Điều kiện sau	Quản lý theo dõi thành công thời gian ra vào
Dòng sự kiện chính	<ul style="list-style-type: none"> 1. Quản lý ấn chọn “Thông tin ra vào” 2. Hệ thống hiển thị danh sách nhân viên cùng với thời điểm ra/vào công ty 3. Quản lý ấn chọn vào tên một nhân viên để đi đến trang thông tin ra vào chi tiết của nhân viên đó 4. Hệ thống hiển thị danh sách các mốc thời gian ra/vào công ty trong ngày của nhân viên và các thông tin khác (thông tin ra vào của các ngày trước đó, số ngày làm/vắng trong tháng,...) 5. Quản lý ấn chọn vào từng mốc thời gian ra vào cụ thể để theo dõi thông tin xác thực chi tiết (hình ảnh được ghi nhận để xác thực, loại xác thực: diêm danh hay ra vào giữa giờ,...)
Dòng sự kiện thay thế	<ul style="list-style-type: none"> Alternative 1: Tại bước 2: Quản lý có thể ấn chọn thanh tìm kiếm để tra cứu thông tin ra vào của một nhân viên cụ thể (bằng mã NV hoặc họ tên) Alternative 2: Tại bước 2: Quản lý có thể ấn chọn và sử dụng các nút sort, filter để dễ dàng theo dõi thông tin ra vào tại các khoảng thời gian cụ thể
Ngoại lệ	Nếu hệ thống xảy ra lỗi trong quá trình trên, hệ thống sẽ hiển thị thông báo lỗi và điều hướng về giao diện trang chủ.

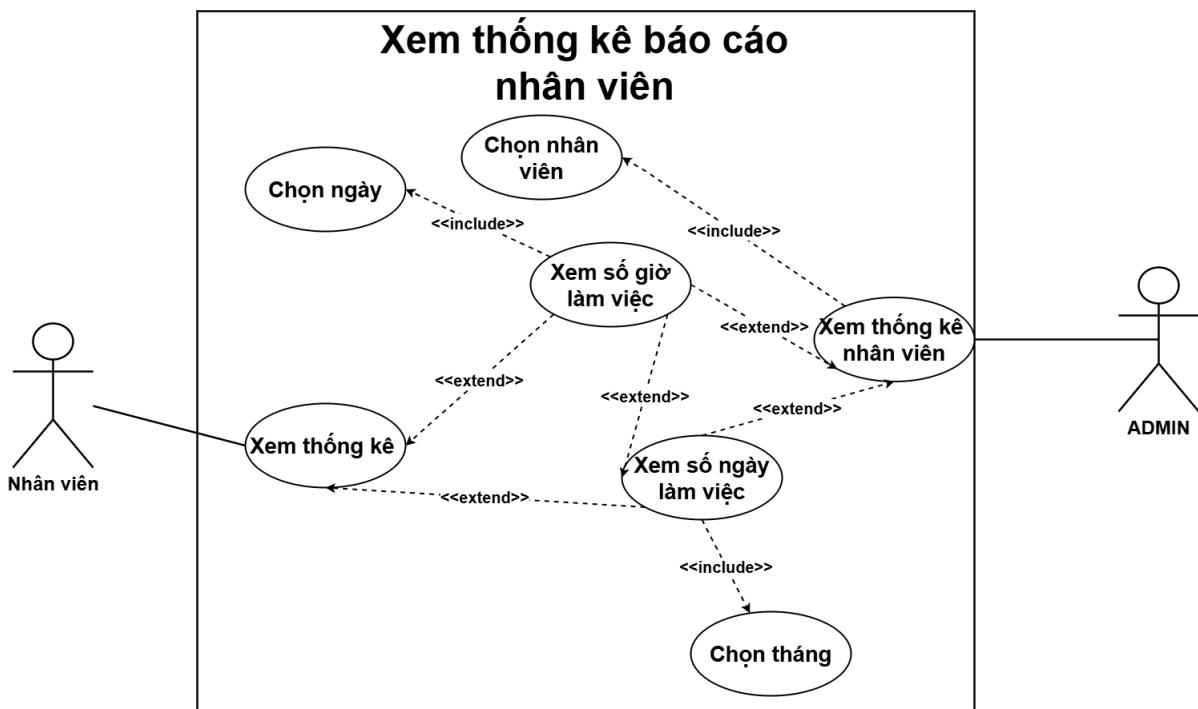
6.3 Thông báo và nhắc nhở nhân viên



Bảng 3: Use case «Thông báo và nhắc nhở nhân viên»

Use case	Descriptions
ID	UC - 03
Tên	Hệ thống thông báo và nhắc nhở thời gian làm việc
Mô tả	Hệ thống cung cấp thông báo và nhắc nhở cho nhân viên trong quá trình check-in và check-out tại nơi làm việc. Các thông báo và nhắc nhở này được thiết kế để giúp nhân viên duy trì một lịch trình làm việc hiệu quả và tuân thủ các quy định nội bộ của công ty về thời gian làm việc và nghỉ giữa giờ.
Actor	Nhân viên, quản lý, hệ thống
Điều kiện tiên quyết	Hệ thống đang hoạt động, thông tin của nhân viên đã được đăng ký trong hệ thống và có quyền truy cập vào nơi làm việc.
Điều kiện sau	Nhân viên nhận được thông báo và nhắc nhở phù hợp dựa trên thời gian check-in, check-out hoặc trong khoảng thời gian nghỉ của họ.
Dòng sự kiện chính	<ol style="list-style-type: none"> Hệ thống gửi thông báo nhắc nhở tới nhân viên trước 15 phút trong trường hợp nhân viên chưa check in lần đầu trong các ca. Hệ thống gửi thông báo check in/ check out thành công khi nhân viên hoàn thành xác thực khuôn mặt và ấn xác nhận (YES) hoặc sử dụng OTP. Khi nhân viên check in muộn, hệ thống xác nhận, tính toán thời gian muộn trong ca làm việc vào lúc điểm danh và gửi thông tin tới nhân viên thông qua ứng dụng. Trước thời gian kết thúc ca 15 phút, hệ thống gửi thông báo tới nhân viên, nhắc nhở thời gian sắp tan ca. Khi kết thúc ngày, hệ thống cập nhật thông tin check-in/check-out gần nhất và cuối cùng để xác định xem ngày đó đã đi làm, trạng thái làm việc và lần cuối check out là bao nhiêu để tính toán thời gian, checkout sớm hoặc đúng giờ.
Ngoại lệ	Nếu có vấn đề kỹ thuật với hệ thống, thông báo có thể không được gửi đi như mong đợi.

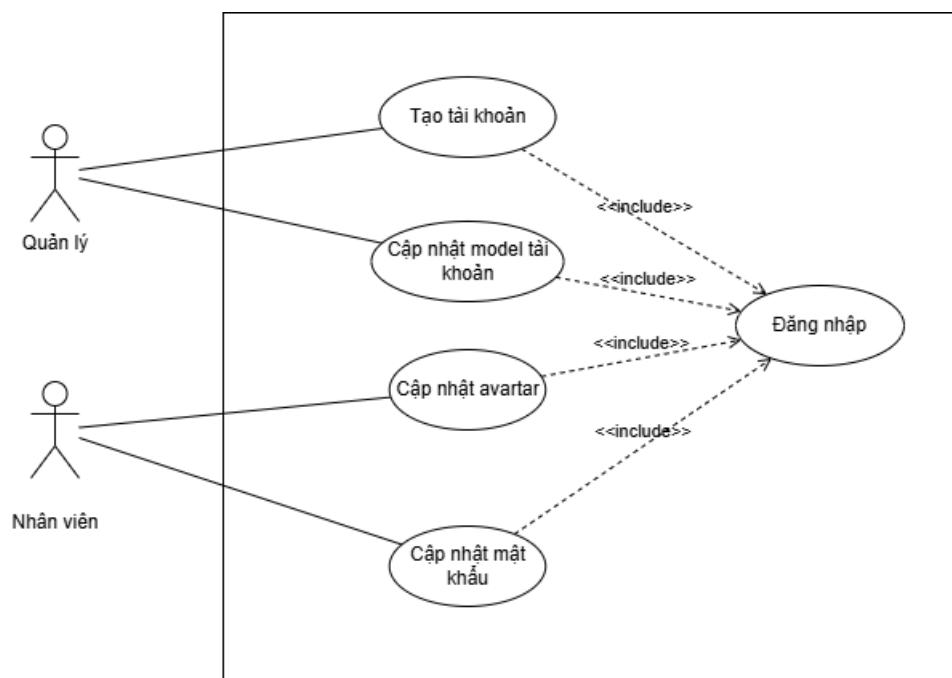
6.4 Xem thống kê và báo cáo



Bảng 4: Use case «Xem thống kê và báo cáo»

Use case	Descriptions
ID	UC - 04
Tên	Xem thống kê báo cáo nhân viên
Mô tả	Xem thống kê báo cáo nhân viên
Actor	Quản lý
Điều kiện tiên quyết	Hệ thống đang hoạt động Quản lý đã đăng nhập
Điều kiện sau	Quản lý xem được báo cáo
Dòng sự kiện chính	1.Quản lý bấm nút xem thống kê báo cáo nhân viên 2.Diều hướng tới hệ thống quản lý nhân viên 3.Tìm kiếm nhân viên cần xem thống kê báo cáo 4.Diều hướng tới thống kê báo cáo của nhân viên được chọn, hiển thị các tùy chọn chức năng: xem thời gian làm việc, xem số ngày làm việc trong tháng, xem ngày nghỉ.
Ngoại lệ	3.a Nhân viên cần tìm không tồn tại thì hiển thị tìm kiếm không thành công 3.a.1 Quản lý chọn lệnh tìm kiếm lại 3.a.2 Quản lý chọn về menu

6.5 Tài khoản





Bảng 5: Use case «Tạo tài khoản»

Use case	Descriptions
ID	UC - 05 - 1
Tên	Tạo tài khoản
Mô tả	Tạo tài khoản với thành viên trong hệ thống chưa có tài khoản
Actor	Quản lý
Điều kiện tiên quyết	Hệ thống đang hoạt động. Quản lý đã đăng nhập.
Điều kiện sau	Tạo tài khoản thành công
Dòng sự kiện chính	1. Quản lý bấm nút tạo tài khoản 2. Quản lý điền thông tin : Username, mật khẩu, ID của thành viên trong hệ thống. 3. Hệ thống kiểm tra tài khoản đã tồn tại hay chưa. 4. Hệ thống thêm tài khoản vào cơ sở dữ liệu. 5. Hệ thống hiện tạo mã thành công
Dòng sự kiện thay thế	3.a. Tài khoản đã tồn tại. .3.a.1. Trở về bước 2.

Bảng 6: Use case «Cập nhật model tài khoản»

Use case	Descriptions
ID	UC - 05 - 2
Tên	Cập nhật model tài khoản
Mô tả	Quản lý cập nhật hình ảnh phục vụ cho việc train model xác thực khuôn mặt
Actor	Quản lý
Điều kiện tiên quyết	Hệ thống đang hoạt động. Quản lý đã đăng nhập.
Điều kiện sau	cập nhật hình ảnh thành công
Dòng sự kiện chính	1. Quản lý bấm nút cập nhật hình ảnh 2. Quản lý chọn 2 hình thức: Thêm ảnh đã có sẵn hoặc chụp ảnh. 3. Hệ thống kiểm tra ảnh 4. Hệ thống thêm ảnh vào cơ sở dữ liệu.
Dòng sự kiện thay thế	3.a. Ảnh không phù hợp với cấu trúc của hệ thống. .3.a.1. Trở về bước 2.

Bảng 7: Use case «Đăng nhập»

Use case	Descriptions
ID	UC - 05 - 4
Tên	Đăng nhập
Mô tả	Người dùng đăng nhập
Actor	Người dùng
Điều kiện tiên quyết	Hệ thống đang hoạt động. Người dùng đã có tài khoản. Kết nối mạng.
Điều kiện sau	Đăng nhập thành công
Dòng sự kiện chính	1.Người dùng nhập SDT, mật khẩu 2.Xác nhận đăng nhập 3.Hệ thống kiểm tra đăng nhập 4.Phản hồi đăng nhập thành công 5.Chuyển đến giao diện mặc định của ứng dụng
Dòng sự kiện ngoại lệ	4.a.Phản hồi đăng nhập thất bại, quay lại bước 1.

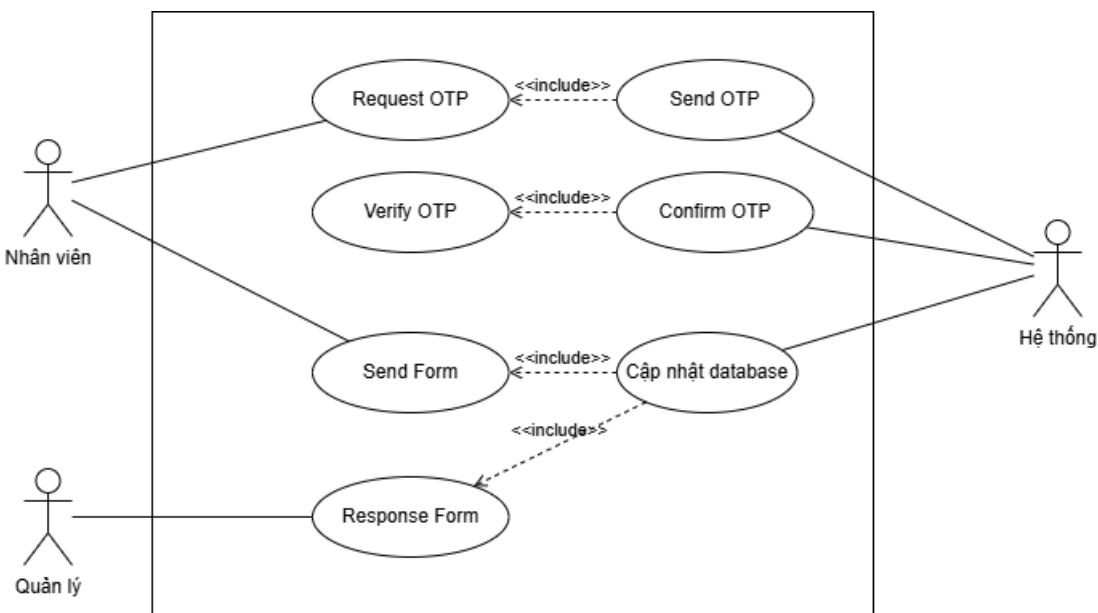
Bảng 8: Use case «Cập nhật avatar»

Use case	Descriptions
ID	UC - 05 - 5
Tên	Cập nhật avatar
Mô tả	Người dùng cập nhật avatar cho tài khoản của mình
Actor	Người dùng
Điều kiện tiên quyết	Hệ thống đang hoạt động. Người dùng đã đăng nhập.
Điều kiện sau	Cập nhật avatar thành công
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng bấm chọn cập nhật avatar. 2. Người dùng chọn ảnh và thêm vào. 3. Hệ thống kiểm tra ảnh 4. Hệ thống thêm ảnh vào cơ sở dữ liệu.
Dòng sự kiện thay thế	<ol style="list-style-type: none"> 3.a. Ảnh không phù hợp với cấu trúc của hệ thống. 3.a.1. Trở về bước 2.

Bảng 9: Use case «Cập nhật mật khẩu»

Use case	Descriptions
ID	UC - 05 - 6
Tên	Cập nhật mật khẩu
Mô tả	Người dùng cập nhật lại mật khẩu của mình
Actor	Người dùng
Điều kiện tiên quyết	Hệ thống đang hoạt động. Người dùng đã đăng nhập.
Điều kiện sau	Cập nhật mật khẩu thành công
Dòng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng bấm chọn đổi mật khẩu. 2. Người dùng điền mật khẩu. 3. Hệ thống thay đổi mật khẩu trong cơ sở dữ liệu.

6.6 Giao tiếp và phản hồi





Bảng 10: Use case «OTP»

Use case	Descriptions
ID	UC - 06 - 01
Tên	Yêu cầu mã OTP check in/check out
Mô tả	Tạo mã OTP cho nhân viên để mở cửa trong trường hợp hệ thống gặp vấn đề
Actor	Nhân viên, quản lý, hệ thống
Điều kiện tiên quyết	Hệ thống đang hoạt động Người dùng đang trong phiên đăng nhập
Điều kiện sau	Mã OTP được hệ thống tạo ra và sẵn sàng được gửi về cho User
Dòng sự kiện chính	1. User nhấn nút “Yêu cầu cấp OTP” (checkin/checkout) 2. User nhập lí do cần cấp OTP. 3. API yêu cầu tạo OTP được gửi đến hệ thống. 4. Hệ thống tạo mã OTP để gửi về cho User đã yêu cầu, OTP có hiệu lực trong vòng 1 phút (60 giây). 5. Người dùng nhập OTP. 6. Hệ thống xác nhận OTP, mở cửa cho người dùng.
Dòng sự kiện thay thế	6.a. OTP sai trở về bước 1.

Bảng 11: Use case «Tạo đơn»

Use case	Descriptions
ID	UC - 06 - 02
Tên	Tạo đơn (form)
Mô tả	Người dùng tạo đơn. Đơn được ghi đầy đủ thông tin nhân viên, thời gian và lý do viết đơn.
Actor	Nhân viên, hệ thống
Điều kiện tiên quyết	Hệ thống đang hoạt động Người dùng đang trong phiên đăng nhập
Điều kiện sau	Người dùng gửi đơn về cho hệ thống để quản lý xem xét và xác nhận.
Dòng sự kiện chính	1. Người dùng nhấn nút “Tạo đơn” 2. Một đơn trống được hiện ra với các mục thông tin để cho người dùng điền vào 3. Người dùng điền thông tin vào đơn và nhấn nút gửi 4. Đơn sẽ được lưu vào database và gửi về cho quản lý để xem xét và xác nhận
Dòng sự kiện thay thế	3.a. Người dùng không điền thông tin hoặc điền thông tin sai lệch 3.a.1. Hệ thống sẽ đưa ra thông báo điền lại thông tin



Bảng 12: Use case «Phản hồi đơn»

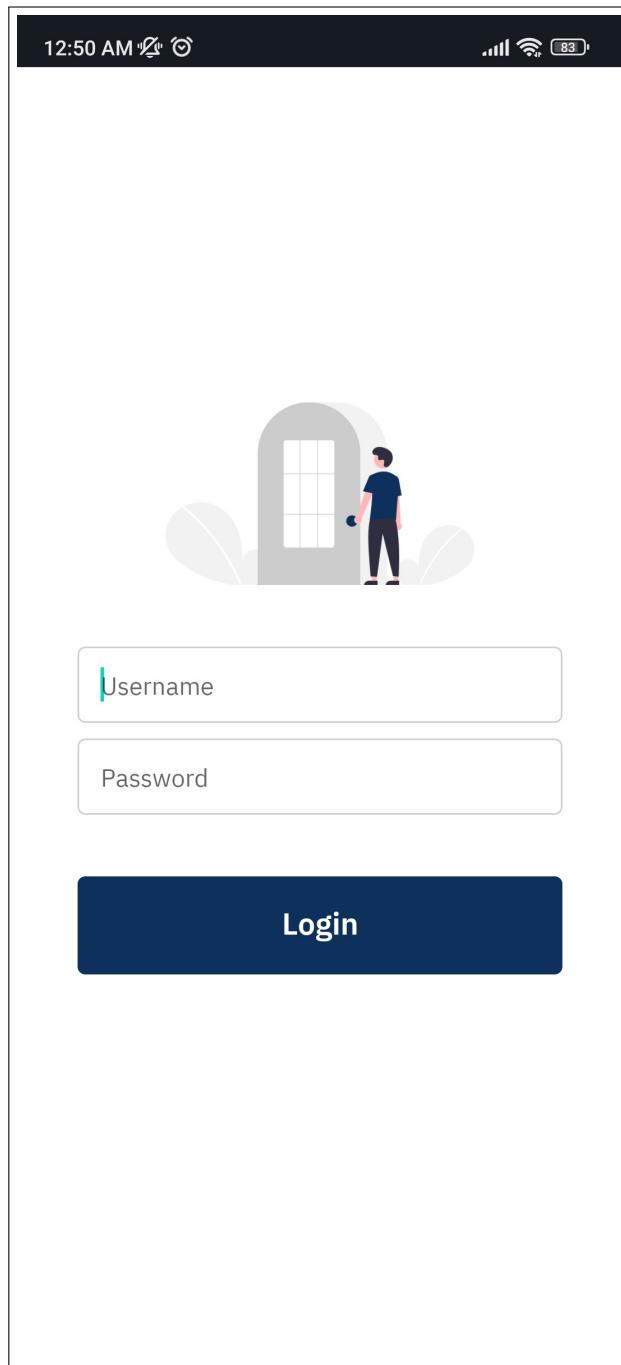
Use case	Descriptions
ID	UC - 06 - 03
Tên	Phản hồi đơn
Mô tả	Quản lý nhận được các đơn của nhân viên từ hệ thống. Quản lý có thể xác nhận hoặc từ chối các đơn này và xác nhận đã nhận được đơn.
Actor	Quản lý
Điều kiện tiên quyết	Hệ thống đang hoạt động Quản lý đang trong phiên đăng nhập
Điều kiện sau	Quản lý xác nhận đơn của nhân viên, có thể cho phép hoặc từ chối với yêu cầu này.
Dòng sự kiện chính	1. Quản lý nhận được đơn yêu cầu của nhân viên từ hệ thống. 2. Quản lý xem xét yêu cầu và chọn “Đồng ý” hoặc “Từ chối” 3. Quản lý xác nhận đã xem đơn yêu cầu.

7 Mockup design

Lưu ý: Một số hình ảnh dưới đây không còn phản ánh thiết kế hiện tại của ứng dụng và chỉ được lưu giữ cho mục đích lưu trữ và so sánh.



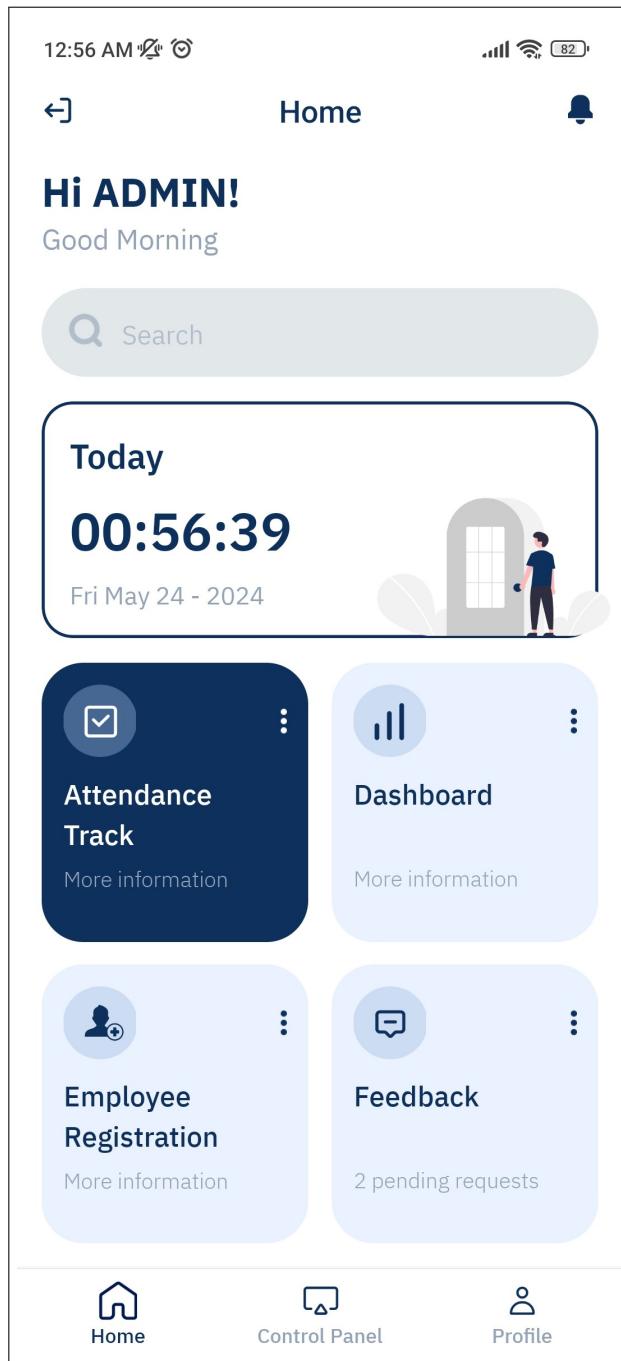
7.1 Login Screen



Hình 9: Login Screen

Tại trang đăng nhập, người dùng nhập tài khoản và mật khẩu được cấp để truy cập vào ứng dụng

7.2 Home Screen



Hình 10: Admin Home Screen

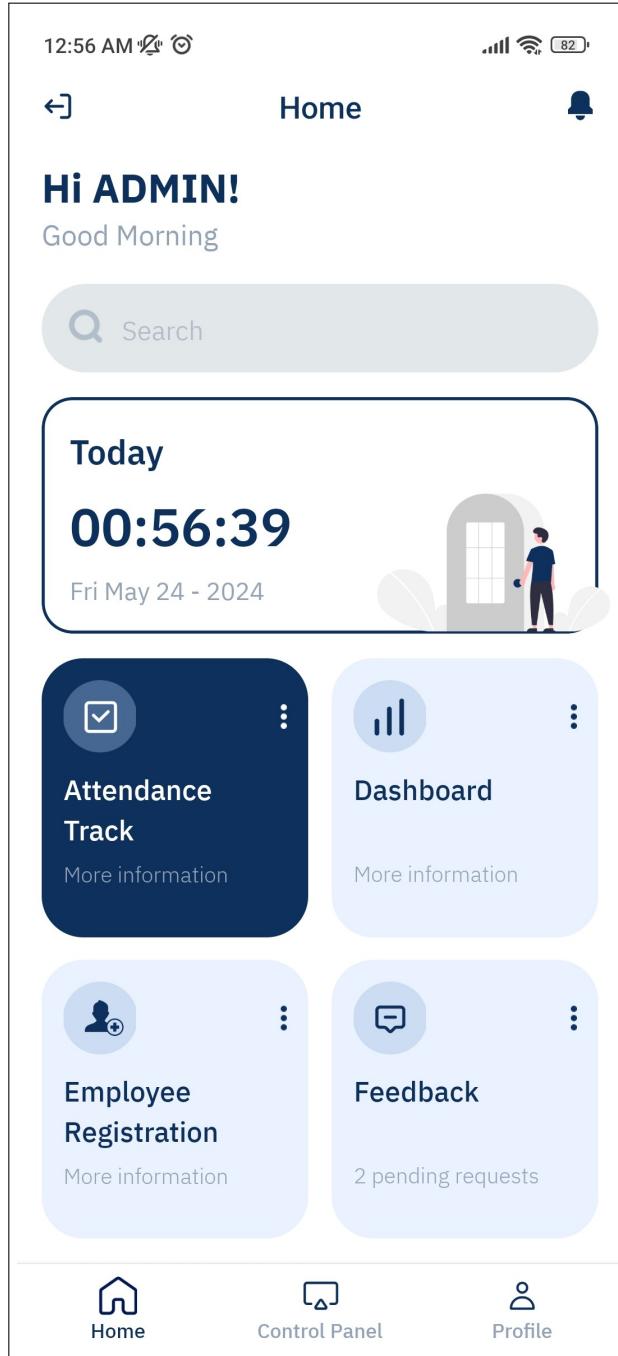
Nhóm thiết kế một màn hình chính cho Admin (tài khoản dành cho người quản lý) hiển thị các thông tin ngày giờ cơ bản. Góc phía trên cùng bên trái là nút đăng xuất, góc phía trên cùng bên phải là nút hiển thị thông báo.

Màn hình chính của Admin sẽ bao gồm 4 nút thẻ hiển thị 4 nhóm chức năng chính của Admin bao gồm:

- Attendance Track: Theo dõi thông tin ra vào của toàn công ty
- Dashboard: Tạo thống kê dữ liệu

- Employee Registration: Đăng ký tài khoản cho nhân viên mới
- Feedback: Nhận phản hồi từ nhân viên

Thanh Tab phía bên dưới điều hướng giữa 3 giao diện Home, Control Panel và Profile



Hình 11: User Home Screen

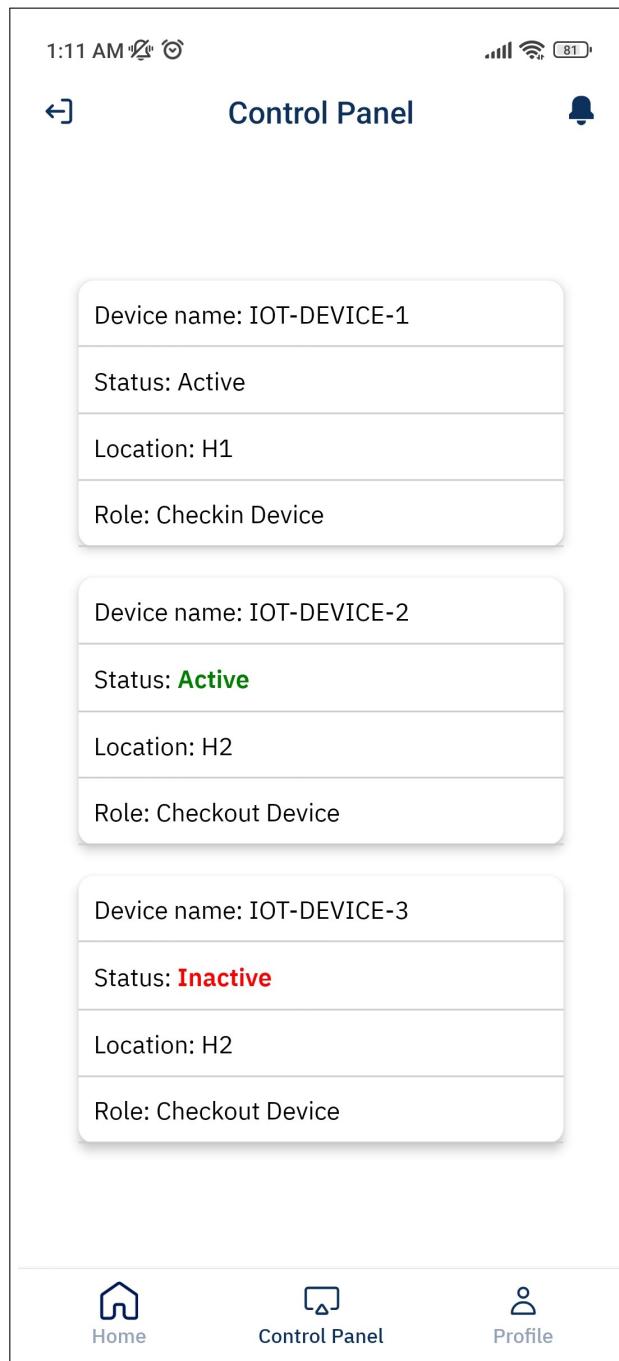
Màn hình chính cho User (nhân viên) cũng được thiết kế tương tự. Trong đó, 4 nhóm chức năng chính của User bao gồm:

- Attendance Track: Theo dõi thông tin ra vào của bản thân
- Dashboard: Tạo thống kê dữ liệu

- OTP Request: Yêu cầu mã OTP trong trường hợp khẩn cấp
- Report: Gửi phản hồi tới quản lý

Thanh Tab phía bên dưới điều hướng giữa 2 giao diện Home và Profile

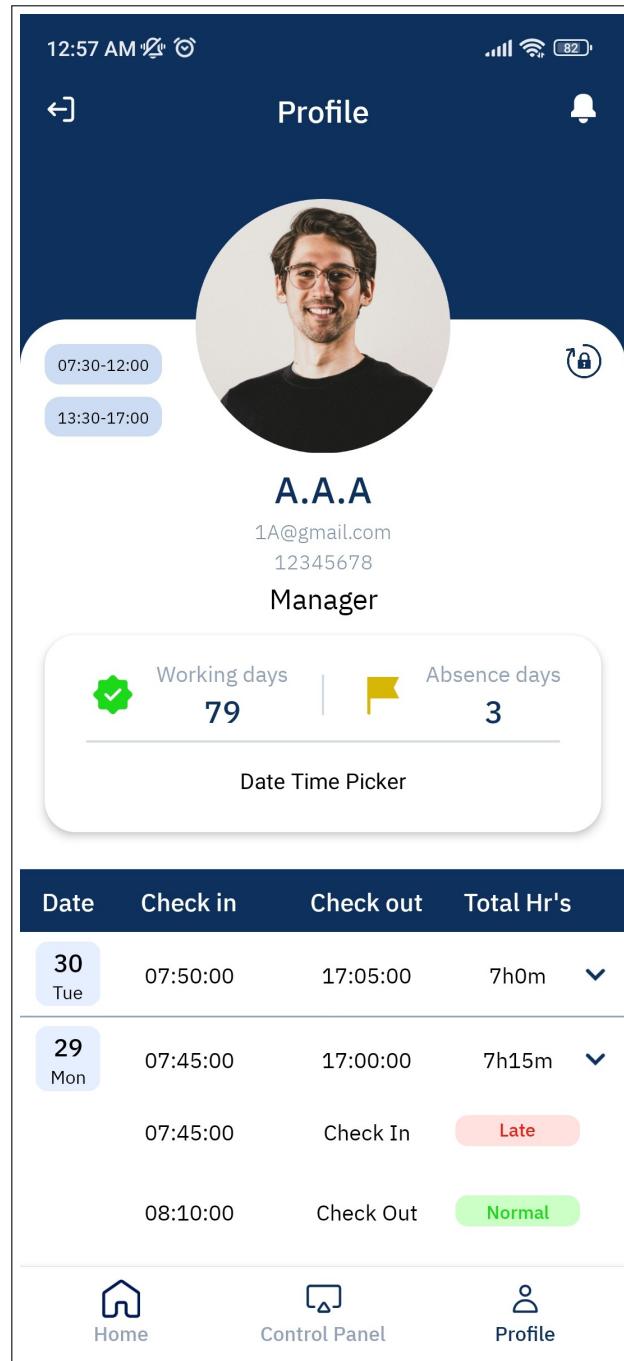
7.3 Control Panel Screen



Hình 12: Control Panel Screen

Màn hình trên chứa thông tin của các thiết bị IoT được đặt trong công ty như: tên thiết bị, trạng thái, địa điểm, chức năng

7.4 Profile Screen

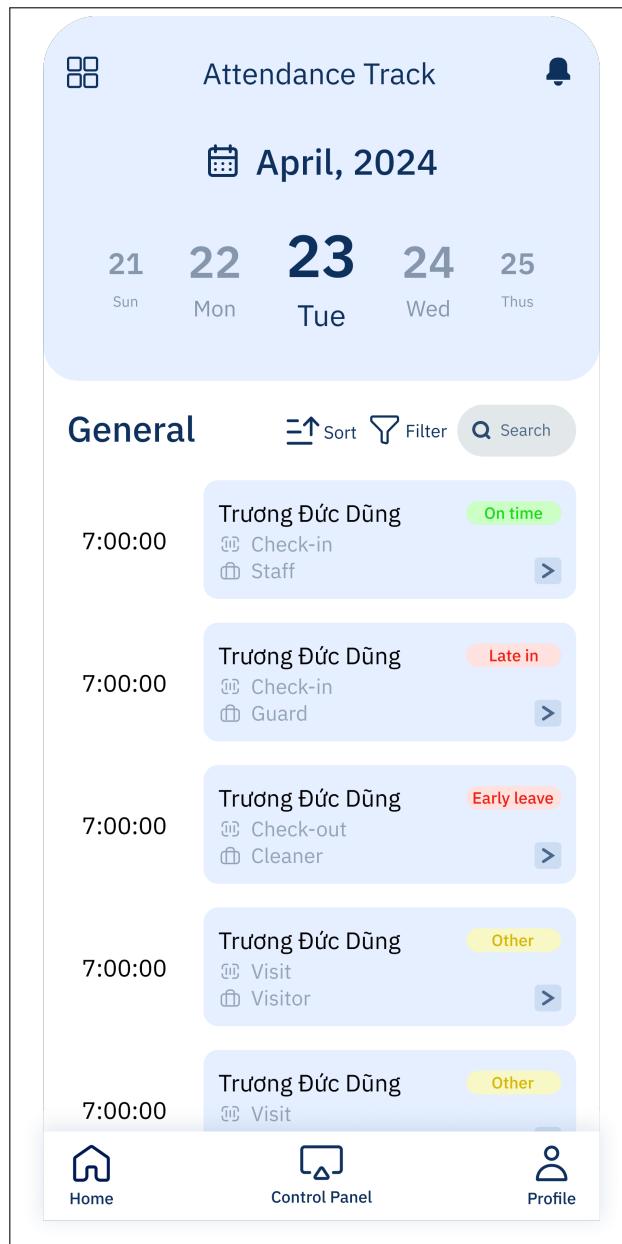


Hình 13: Profile Screen

Màn hình trên chứa thông tin cá nhân của nhân viên như: họ tên, email, số điện thoại, chức vụ, số ngày làm việc, số ngày vắng, ca làm việc, avatar.

Ngoài ra, phía nửa dưới giao diện là thông tin ra vào cùng với số giờ làm việc của nhân viên đó trong tháng.

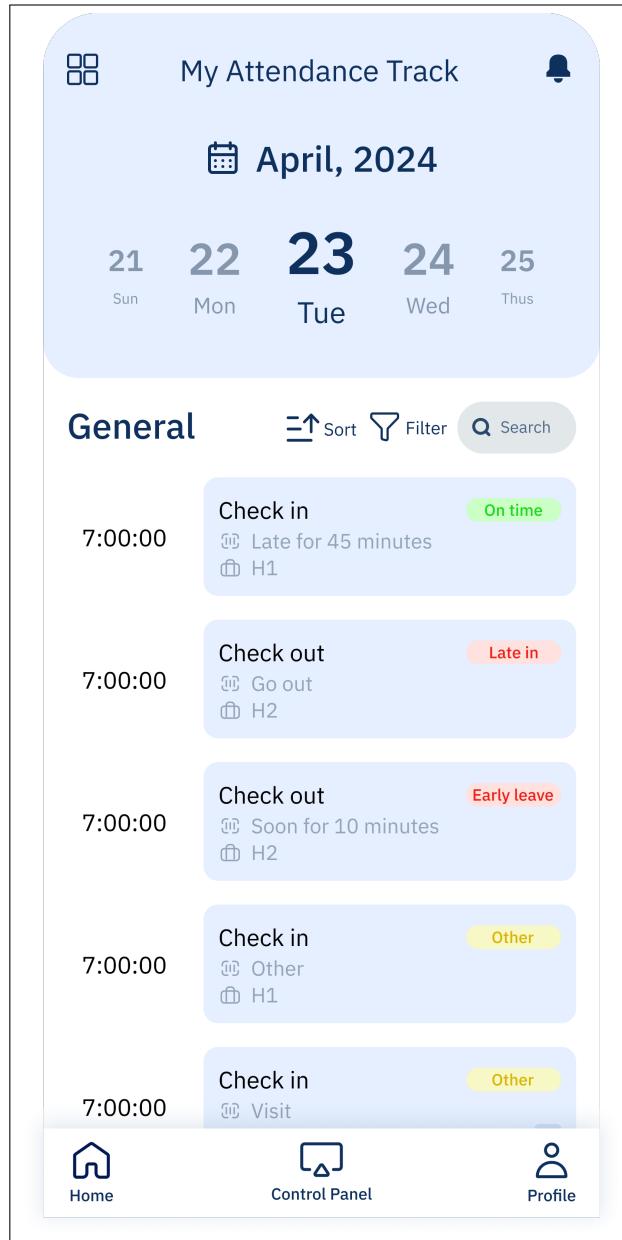
7.5 Attendance Track Screen



Hình 14: Attendance Track Screen

Màn hình trên (dành cho quản lý) thể hiện thông tin ra vào của tất cả nhân viên trong công ty với giao diện chọn ngày để theo dõi. Thông tin ra vào được cung cấp bao gồm: thời gian quét, tên nhân viên, hành động, chức vụ, trạng thái,... cùng với các nút chức năng như Sort, Filter, Search.

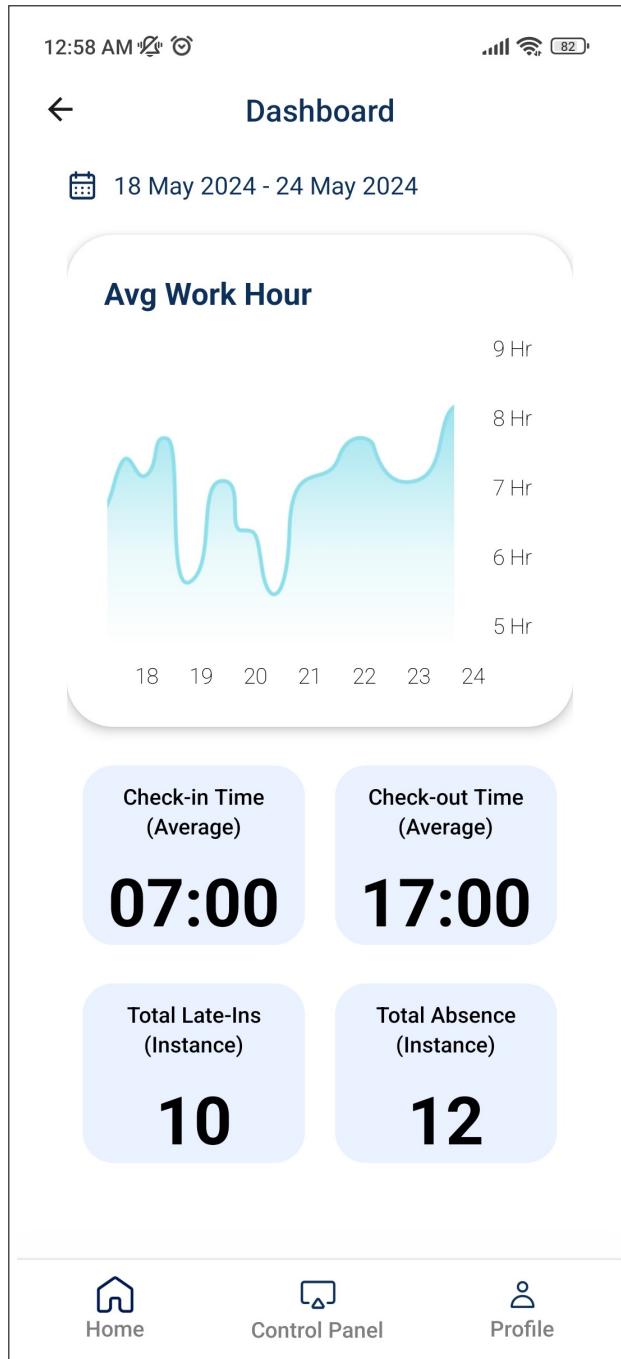
7.6 My Attendance Track Screen



Hình 15: My Attendance Track Screen

Màn hình trên (dành cho nhân viên) thể hiện thông tin ra vào của bản thân với giao diện chọn ngày để theo dõi. Thông tin ra vào được cung cấp bao gồm: thời gian quét, hành động, thông tin thêm, địa điểm, trạng thái,... cùng với các nút chức năng như Sort, Filter, Search.

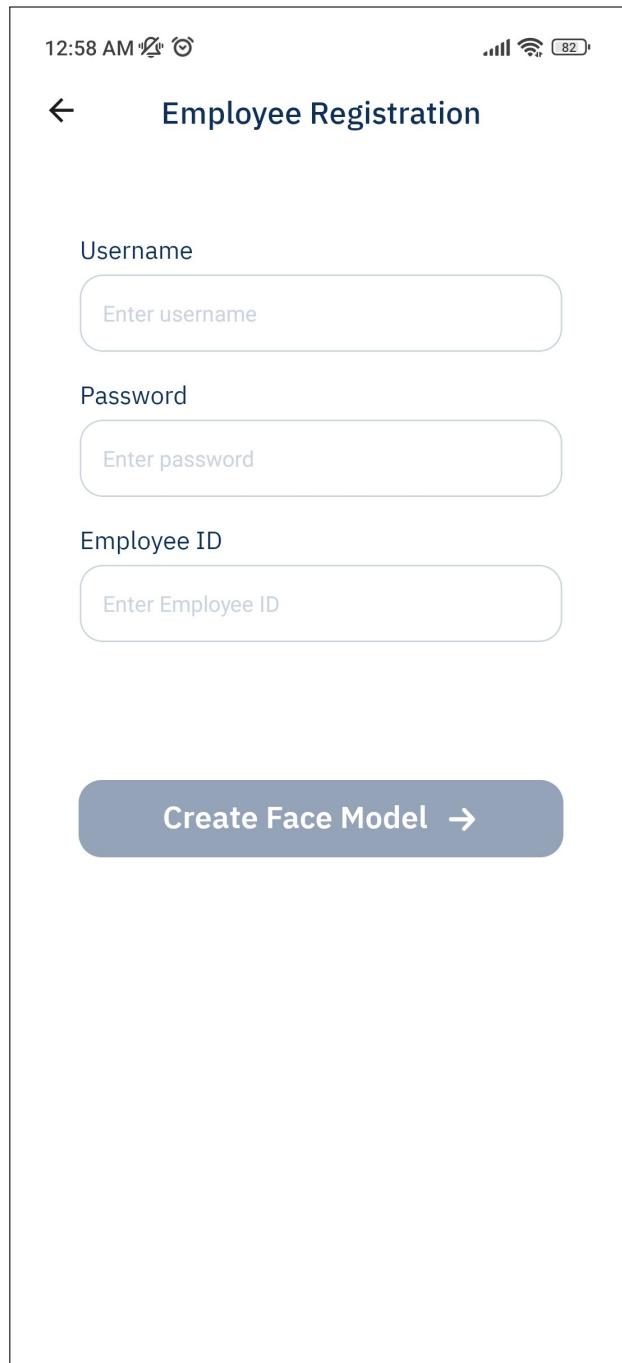
7.7 Dashboard Screen



Hình 16: Dashboard Screen

Màn hình trên thể hiện thống kê dữ liệu cho ứng dụng với phần chọn mốc thời gian thống kê, trung bình số giờ làm việc của toàn công ty và các nhóm dữ liệu khác.

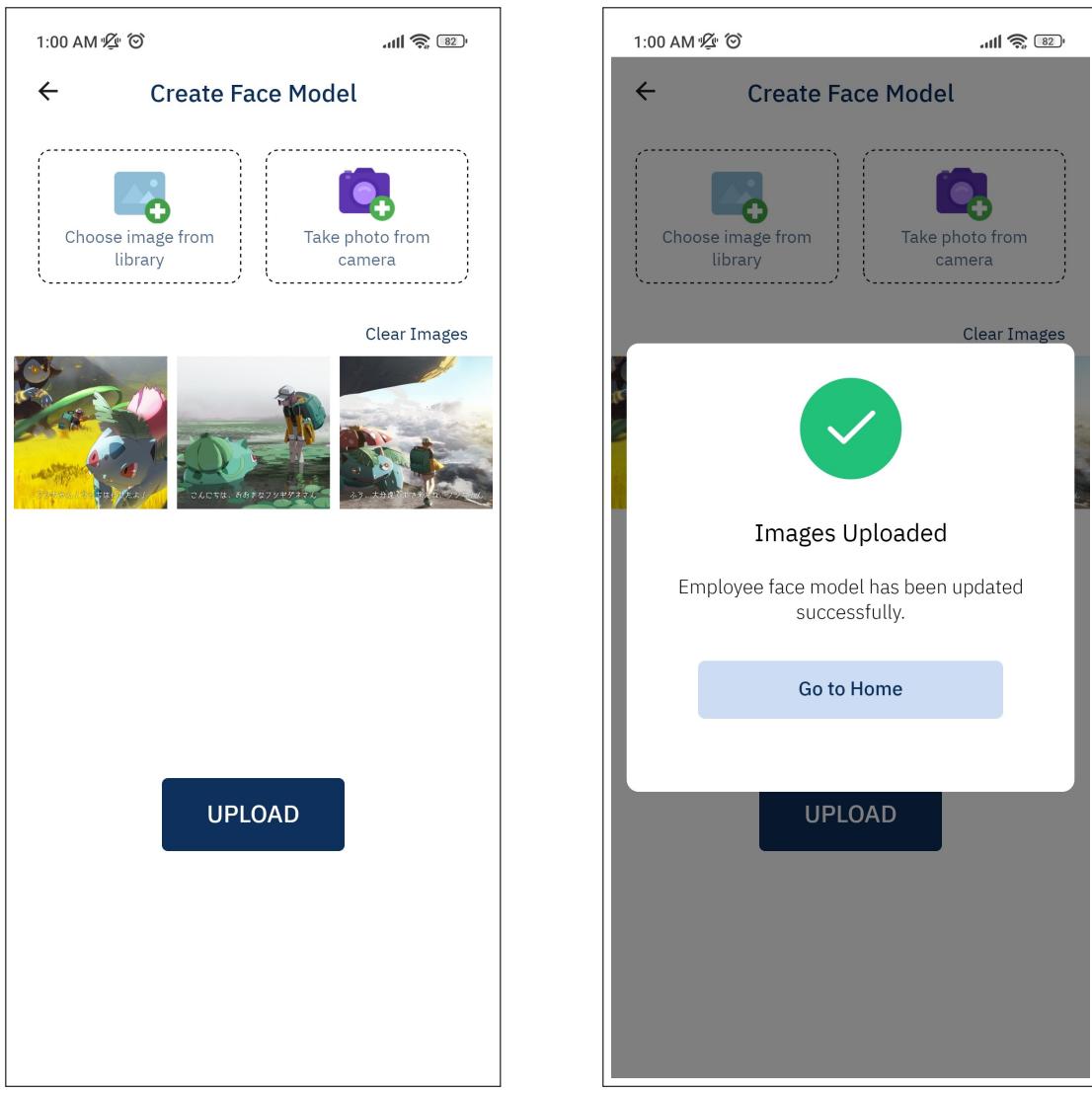
7.8 Employee Registration Screen



Hình 17: Employee Registration Screen

Màn hình cho quản lý tạo tài khoản cho nhân viên mới, nhập vào tên tài khoản, mật khẩu cùng với ID của nhân viên. Nút Create Face Model điều hướng tới bước tiếp theo.

7.9 Create Face Model Screen



(a) Create Face Model Screen

(b) Create Success Modal

Màn hình gồm hai nút cho phép người dùng tải ảnh trực tiếp từ bộ nhớ hoặc chụp ảnh mới. Danh sách ảnh được hiển thị bên dưới. Nhấn chọn nút Upload để xác nhận cập nhật mẫu khuôn mặt. Thông báo cập nhật thành công hiện ra như hình (b).



7.10 Feedback/Report Screen

Request

Request Submit

Mai Hoàng Danh

danh.maikhmt23@hcmut.edu.vn

Staff

Description

🕒 Date and Time

🕒 Date of request

>Type

Type of request

💬 Notes

Add some notes...

CancelSubmit

HomeControl PanelProfile

Feedback

Request

Mai Hoàng Danh

danh.maikhmt23@hcmut.edu.vn

Staff

Description

🕒 Date and Time

16:05:14, 24 April 2024

>Type

Early leave Request

💬 Notes

Cho em về sớm đón con seep oi

REJECTAPPROVE

HomeControl PanelProfile

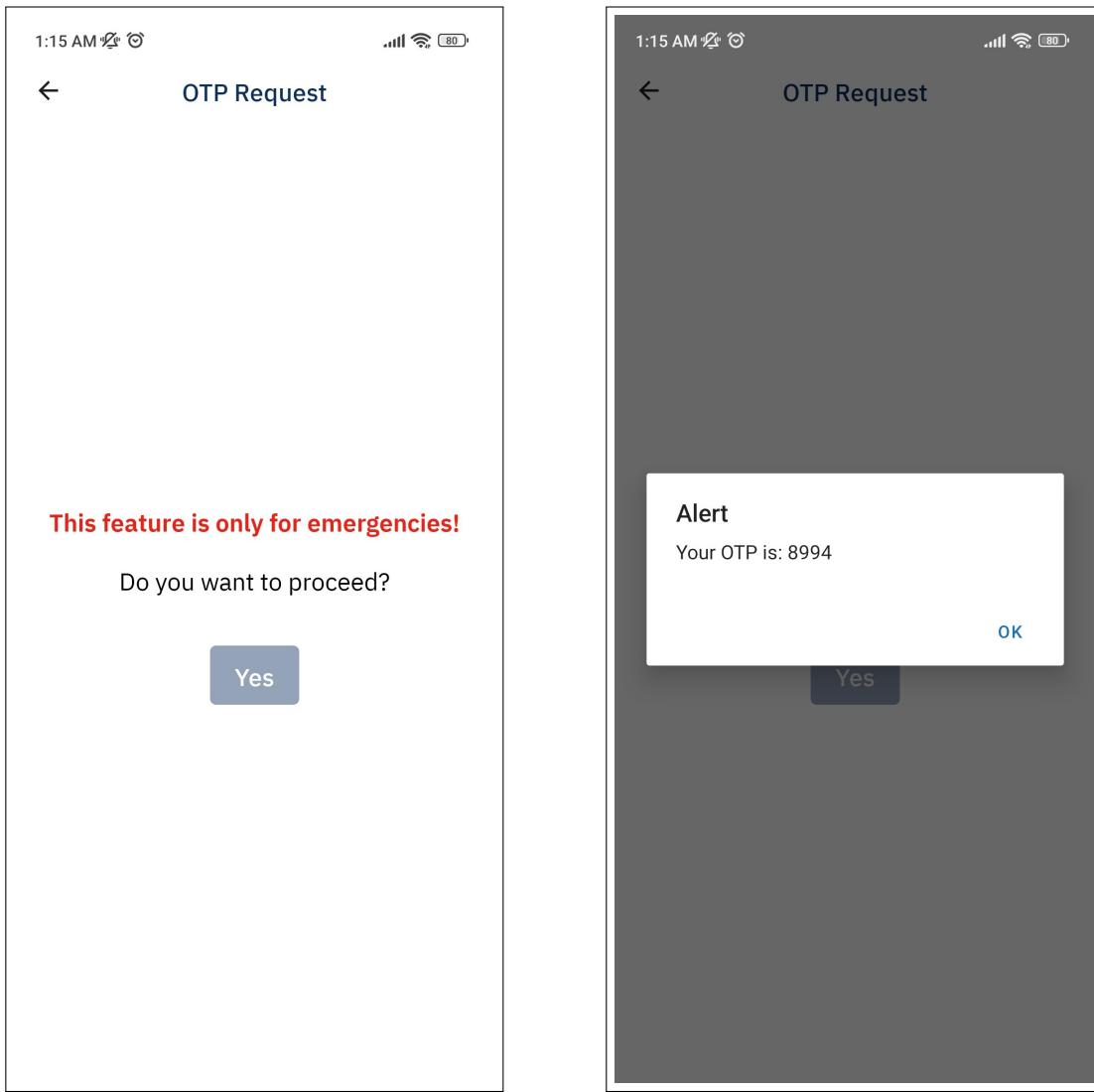
(a) Report Screen

(b) Feedback Screen

Hai màn hình trên cho phép nhân viên gửi phản hồi cho người quản lý với các thông tin như ngày tháng, loại phản hồi, nội dung... Các phản hồi này có thể được chấp nhận hoặc từ chối để xử lý.



7.11 OTP Request Screen



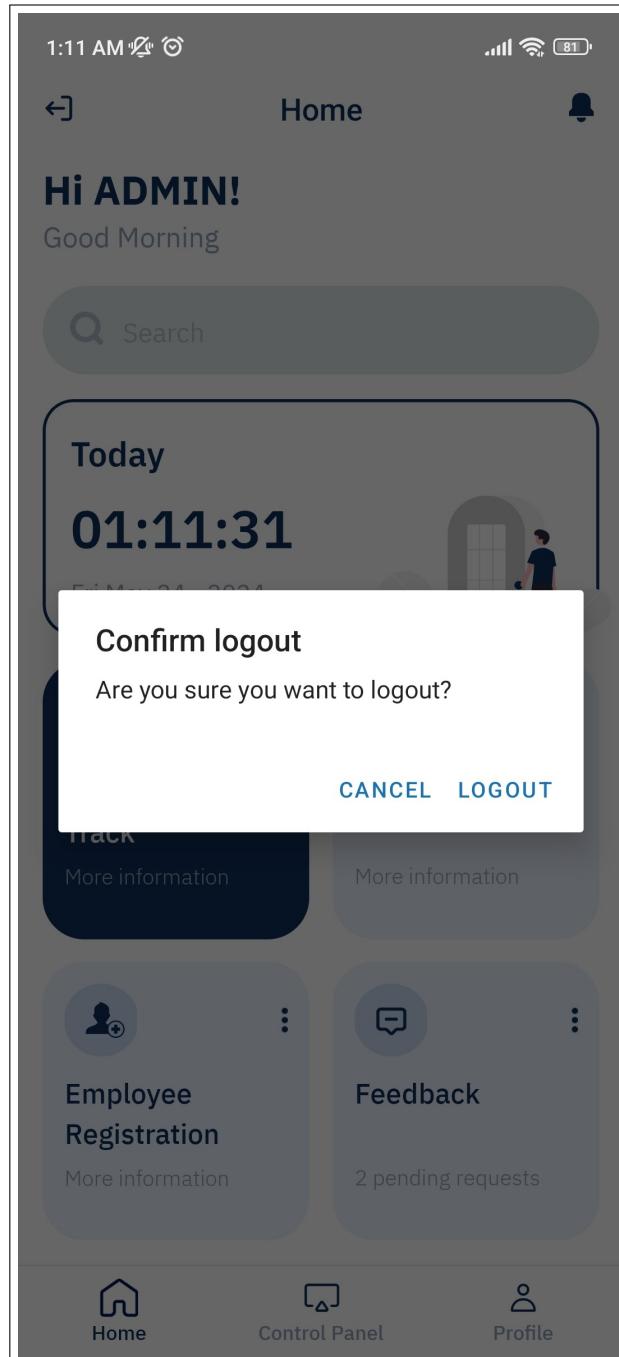
(a) OTP Request Screen

(b) OTP Accept Screen

Màn hình thể hiện chức năng yêu cầu mã OTP trong các trường hợp khẩn cấp như hệ thống nhận diện khuôn mặt gấp vấn đề, người quản lý không nhận được phản hồi,...

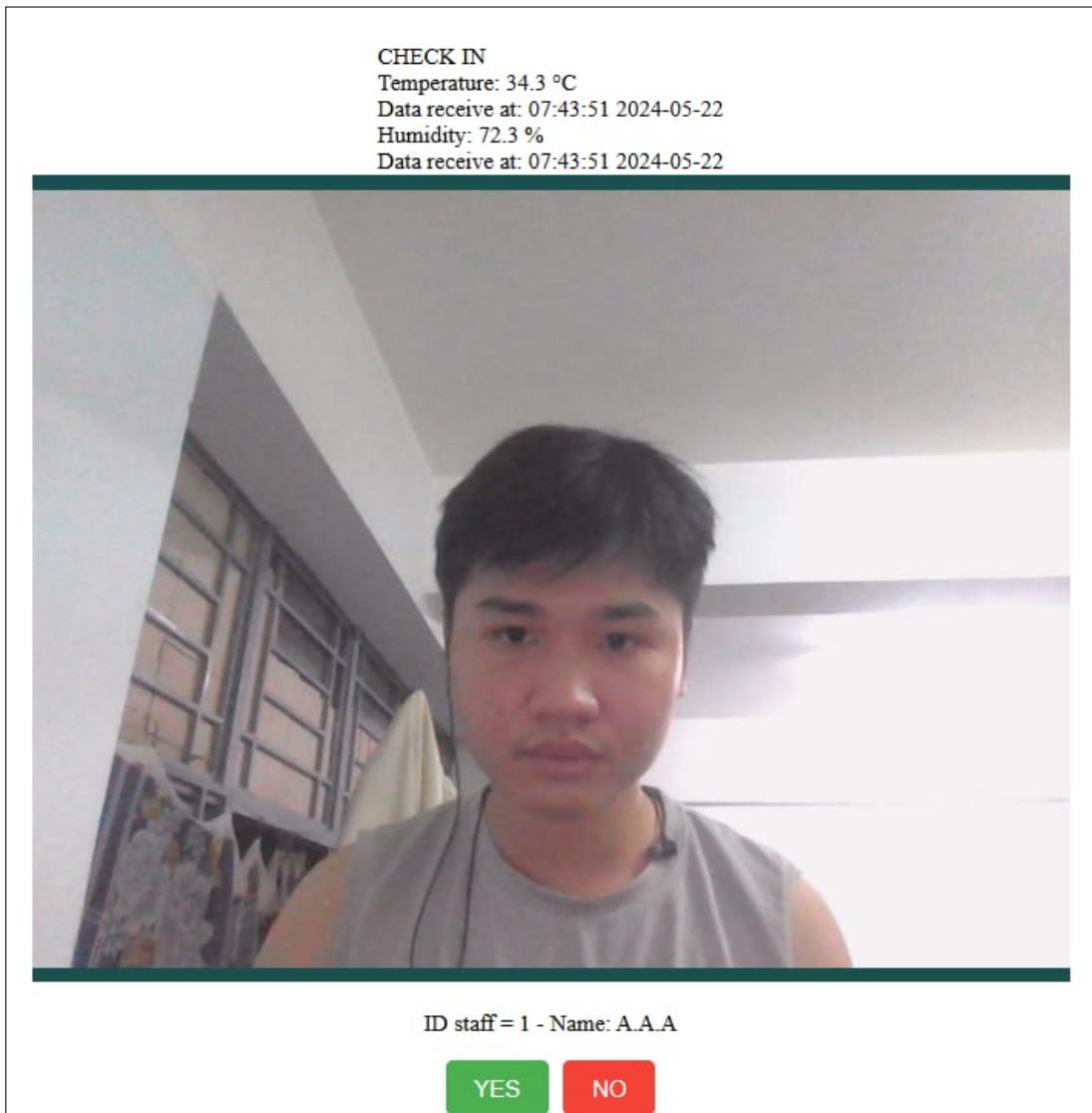


7.12 Logout



Hình 21: Logout Modal

7.13 Face Recognition UI



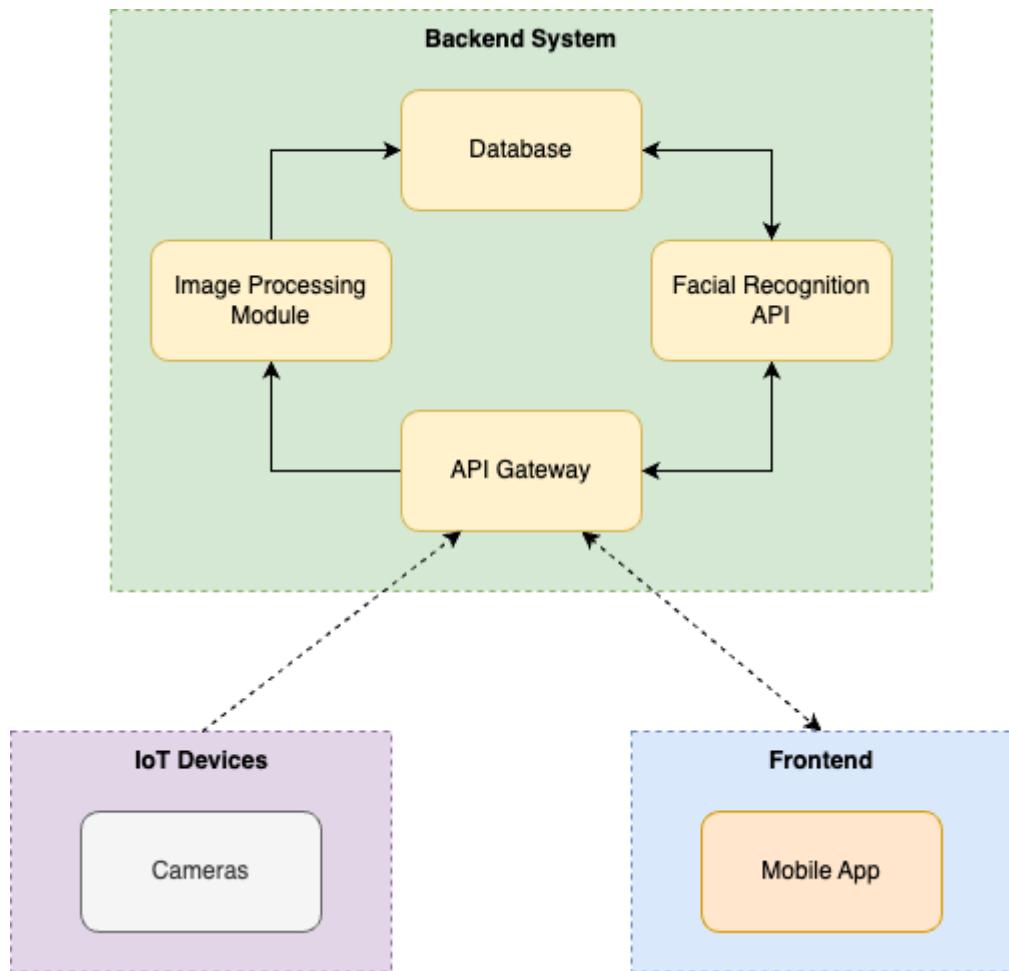
Hình 22: Face Recognition UI

Đây là màn hình mô phỏng hệ thống nhận diện khuôn mặt với các thông tin lấy từ thiết bị IoT như: nhiệt độ, độ ẩm,...

Thông tin nhân viên được hệ thống nhận diện sẽ hiển thị và được xác nhận bởi người quét thông qua 2 nút Yes và No.

8 Architecture

Sơ đồ kiến trúc bao gồm việc đại diện các thành phần khác nhau của hệ thống và tương tác của chúng. Dưới đây là về kiến trúc của ứng dụng sử dụng cơ sở dữ liệu để lưu trữ hình ảnh khuôn mặt, API từ thư viện trực tuyến để nhận diện, thiết bị IoT với camera, và ứng dụng di động:



Hình 23: Kiến trúc của ứng dụng

8.1 Backend System

Gồm các module riêng với các chức năng bổ trợ cho nhau:

- **API Gateway:** Xử lý dữ liệu hình ảnh nhận từ thiết bị IoT và các yêu cầu từ ứng dụng di động.
- **Image Processing Module:** Nhận hình ảnh từ API Gateway, xử lý (ví dụ: thay đổi kích thước, định dạng) và lưu trữ chúng trong database.
- **Database:** Lưu trữ hình ảnh khuôn mặt đã xử lý và dữ liệu liên quan (như thời gian, ID thiết bị, v.v.).
- **Facial Recognition API:** Khi cần nhận diện khuôn mặt, dịch vụ này tương tác với mô hình nhận diện khuôn mặt từ thư viện trực tuyến để phân tích hình ảnh khuôn mặt và trả về kết quả.

8.2 IoT Devices

Bao gồm các thiết bị camera có khả năng nhận diện và chụp hình khuôn mặt. Các thiết bị này có phần mềm nhúng để xử lý hình ảnh và gửi chúng về backend để xử lý và lưu trữ.



8.3 Frontend

Là ứng dụng giao tiếp với backend để lấy kết quả nhận diện và thông báo. Có các tính năng như:

- Cho phép người dùng đăng ký, đăng nhập, và quản lý hồ sơ của mình.
- Giao tiếp với API Gateway để truy cập kết quả nhận diện khuôn mặt và nhận thông báo.
- Hiển thị kết quả nhận diện và cảnh báo cho người dùng.

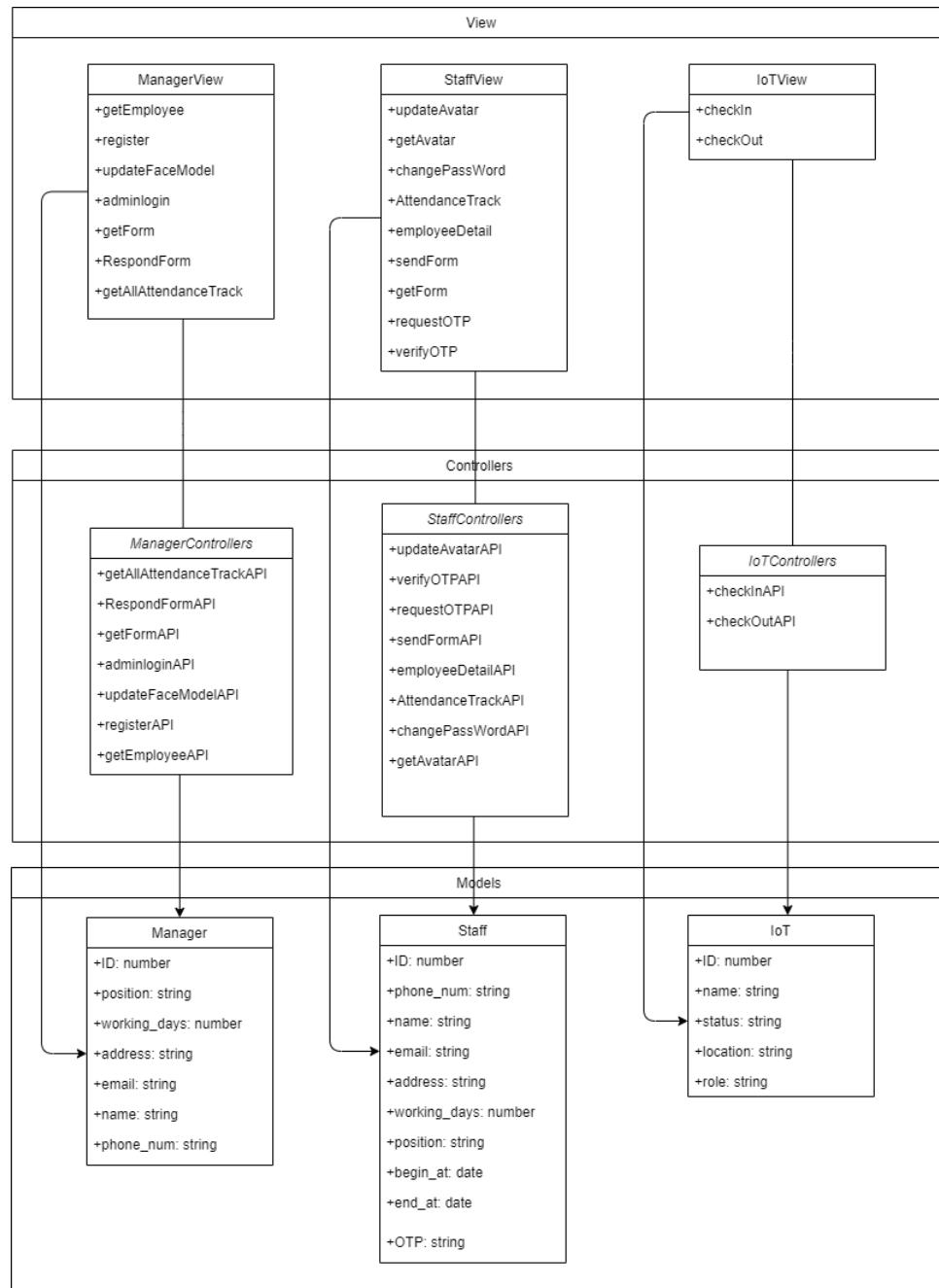
8.4 Workflow

Quy trình hoạt động của hệ thống được diễn ra như sau:

1. Thiết bị IoT chụp hình ảnh khuôn mặt và gửi chúng về backend qua API Gateway.
2. API Gateway chuyển tiếp hình ảnh đến module xử lý hình ảnh.
3. Module Xử lý hình ảnh xử lý và lưu trữ hình ảnh trong Database.
4. Khi cần nhận diện khuôn mặt, module Xử lý Hình ảnh tương tác với Facial Recognition API.
5. Facial Recognition API giao tiếp với mô hình nhận diện khuôn mặt từ thư viện trực tuyến và trả về kết quả.
6. Ứng dụng Di động lấy kết quả nhận diện và thông báo qua API Gateway và hiển thị chúng cho người dùng.

Kiến trúc này đảm bảo xử lý và quản lý hiệu quả hình ảnh khuôn mặt, sử dụng các dịch vụ đám mây và mô hình nhận diện khuôn mặt trực tuyến để cung cấp trải nghiệm liền mạch cho người dùng.

9 Class diagram



Hình 24: Class Diagram

Models là phần của ứng dụng chịu trách nhiệm cho việc quản lý và xử lý dữ liệu. Chúng đại diện cho cấu trúc dữ liệu và logic của ứng dụng, thực hiện các thao tác như truy xuất, cập nhật và xử lý dữ liệu từ cơ sở dữ liệu hoặc các nguồn dữ liệu khác. Models cung cấp một cách tiếp cận tổ chức và làm việc với dữ liệu một cách cấu trúc và hợp lý, đồng thời giữ cho dữ liệu được duy trì và truy cập một cách an toàn và hiệu quả.

View là thành phần chịu trách nhiệm hiển thị dữ liệu và giao diện người dùng tương ứng với các yêu cầu được xử lý bởi Controllers. Nó giúp hiển thị thông tin một cách hợp lý và trực quan cho người dùng, đồng thời tạo ra giao diện tương tác để tương tác với hệ



thống. View là cầu nối giữa dữ liệu và người dùng, mang lại trải nghiệm người dùng cuối cùng.

Controllers là lớp chịu trách nhiệm tiếp nhận và xử lý các yêu cầu HTTP từ người dùng cũng như xử lý dữ liệu của người dùng, sau đó trả về phản hồi phù hợp. Chúng hoạt động như một giao diện trung gian giữa người dùng và hệ thống backend, đảm bảo rằng các yêu cầu được truyền đạt và xử lý chính xác.

- Manager Controller

- getAllAttendanceTrackAPI: Nhận toàn bộ dữ liệu tracking do hệ thống IoT thu thập được khi có người checkin, checkout.
- getFormAPI: Xem tất cả các form đã được người dùng tạo ra.
- respondFormAPI: Xử lý yêu cầu trả lời form của người dùng.
- updateFaceModelAPI: Cập nhật dữ liệu ảnh cho tài khoản của nhân viên để train model.
- registerAPI: Xử lý đăng ký tài khoản cho nhân viên chưa có tài khoản trên hệ thống.
- getEmployeeAPI: Lấy tất cả thông tin nhân viên của công ty có sẵn trên hệ thống.

- Staff Controller

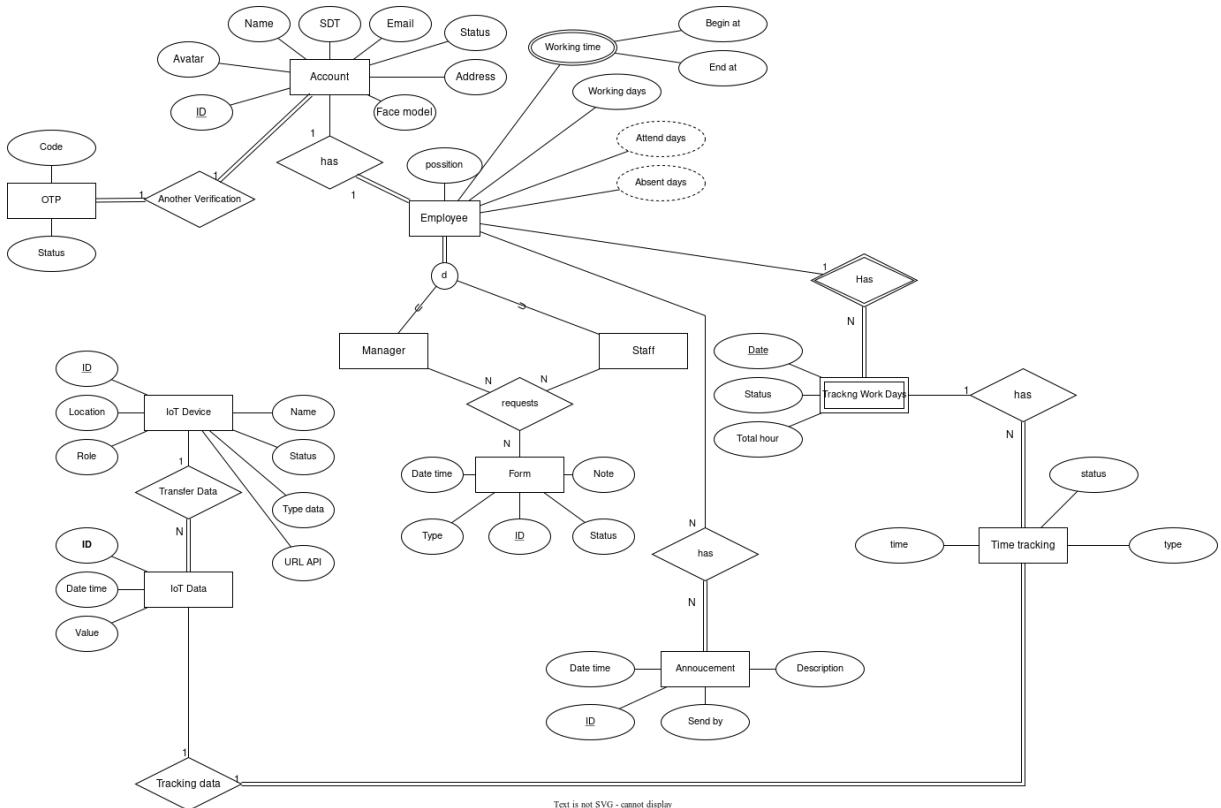
- updateAvatarAPI: Xử lý yêu cầu cập nhật ảnh đại diện của người dùng.
- requestOTPAPI: Yêu cầu nhận OTP từ hệ thống để checkin/checkout.
- verifyOTPAPI: Yêu cầu xác thực OTP để mở cửa checkin/checkout.
- sendFormAPI: Tạo form và gửi form lên hệ thống.
- employeeDetailAPI: Lấy tất cả dữ liệu của tài khoản người dùng.
- attendanceTrackAPI: Lấy dữ liệu được tracking bởi thiết bị IoT khi check-in/checkout.
- changePasswordAPI: Yêu cầu xử lý cập nhật mật khẩu của tài khoản.
- getAvatarAPI: Xử lý yêu cầu nhận avatar của tài khoản người dùng.

- IoT Controller

- checkInAPI: Xử lý yêu cầu check in trên màn hình của hệ thống để xác thực và mở cửa cho người dùng.
- checkOutAPI: Xử lý yêu cầu check out trên màn hình của hệ thống để xác thực và mở cửa cho người dùng.

10 Database

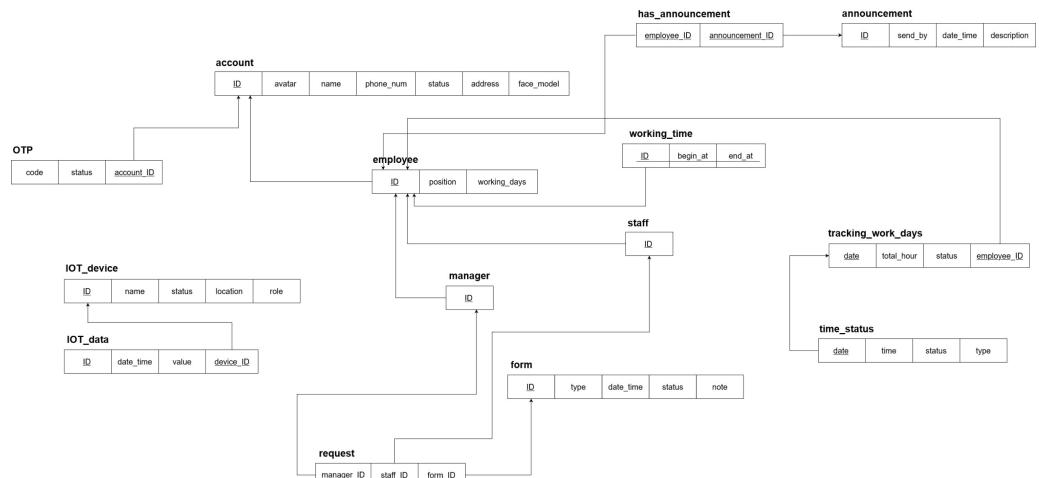
10.1 Thiết kế EERD



Hình 25: Enhanced Entity-Relationship Diagram

10.2 Thiết kế logic - Ánh xạ lược đồ thực thể, mối liên kết mở rộng

10.2.1 Ánh xạ dạng bảng



10.2.2 Ánh xạ dạng viết

- **ACCOUNT** (ID, account_name, password, avatar, status, face_model)



- **EMPLOYEE** (ID, position, working_days, address, email, name, phone_num)
 - **MANAGER** (ID)
Foreign Key: ID to employee.ID
 - **STAFF** (ID)
Foreign Key: ID to employee.ID
 - **WORKING _ TIME** (ID, begin_at, end_at)
Foreign Key: ID to employee.ID
- **FORM** (ID, type, date_time_send, status, note, date_time_respond, response)
 - **REQUEST** (form_ID, manager_ID, staff_ID)
Foreign Key: form_ID to form.ID
Foreign Key: manager_ID to manager.ID
Foreign Key: staff_ID to staff.ID
- **ANNOUNCEMENT** (ID, send_by, date_time, description)
 - **HAS _ ANNOUNCEMENT** (announcement_ID, employee_ID)
Foreign Key: announcement_ID to announcement.ID
Foreign Key: employee_ID to employee.ID
- **OTP** (code, status, account_ID)
Foreign Key: account_ID to account.ID
- **IOT _ DEVICE** (ID, name, status, location, role)
- **TRACKING _ WORK _ DAYS** (date, employee_ID, status)
Foreign Key: employee_ID to employee.ID
- **IOT _ DATA** (ID, date, time, status, value, type, device_ID, employee_ID)
Foreign Key: device_ID to iot_device.ID
Foreign Key: employee_ID to tracking_work_days.ID

11 Technology

11.1 React Native

React Native là một framework mã nguồn mở do Facebook phát triển, cho phép lập trình viên xây dựng ứng dụng di động cho cả iOS và Android bằng cách sử dụng JavaScript và React. Được giới thiệu lần đầu vào năm 2015, React Native đã nhanh chóng trở nên phổ biến nhờ khả năng tiết kiệm thời gian và chi phí phát triển thông qua việc chia sẻ mã nguồn giữa hai nền tảng.

Một số đặc điểm nổi bật của React Native:

- **Cross-platform:** Chỉ cần viết một lần, ứng dụng có thể chạy trên cả iOS và Android.
- **Hiệu suất cao:** Sử dụng các thành phần gốc (native components) của từng nền tảng để đảm bảo hiệu suất tốt nhất.
- **Hot Reloading:** Cho phép lập trình viên xem ngay lập tức các thay đổi trong mã nguồn mà không cần phải build lại ứng dụng.
- **Cộng đồng mạnh mẽ:** Có một cộng đồng lớn hỗ trợ và nhiều thư viện, module có sẵn để sử dụng.

11.2 Expo Go

Expo Go là một ứng dụng di động miễn phí được phát triển bởi nhóm Expo, cho phép các nhà phát triển kiểm tra và chạy các ứng dụng React Native của họ trên thiết bị di động một cách nhanh chóng mà không cần phải qua bước build phức tạp. Expo Go giúp giảm bớt thời gian và công sức khi phát triển và kiểm thử ứng dụng di động.

Một số đặc điểm nổi bật của Expo Go:

- **Dễ sử dụng:** Chỉ cần cài đặt ứng dụng Expo Go trên thiết bị di động, sau đó quét mã QR để chạy ứng dụng React Native ngay lập tức.
- **Cross-platform:** Hỗ trợ cả iOS và Android, cho phép kiểm thử ứng dụng trên nhiều thiết bị và hệ điều hành khác nhau.
- **Không cần cấu hình phức tạp:** Loại bỏ nhu cầu cài đặt các công cụ phát triển phức tạp hoặc thiết lập môi trường.
- **Live Reloading:** Hỗ trợ tự động tải lại ứng dụng sau khi có thay đổi trong mã nguồn, giúp quá trình phát triển nhanh chóng và hiệu quả hơn.

11.3 ExpressJs

Express.js là một framework web tối giản và linh hoạt cho Node.js, được thiết kế để xây dựng các ứng dụng web và API nhanh chóng và dễ dàng. Được phát triển bởi TJ Holowaychuk và ra mắt lần đầu vào năm 2010, Express.js đã trở thành một trong những framework phổ biến nhất trong cộng đồng Node.js.

Một số đặc điểm nổi bật của Express.js:

- **Tối giản và linh hoạt:** Cung cấp một bộ công cụ cơ bản để xây dựng các ứng dụng web, cho phép lập trình viên tự do tùy chỉnh và mở rộng theo nhu cầu.
- **Middleware:** Hỗ trợ hệ thống middleware mạnh mẽ, giúp quản lý các yêu cầu HTTP, routing, và các tính năng khác một cách dễ dàng.
- **Hiệu suất cao:** Được thiết kế để xử lý nhiều yêu cầu đồng thời với tốc độ cao, tận dụng tối đa hiệu suất của Node.js.
- **Cộng đồng mạnh mẽ:** Có một cộng đồng rộng lớn và nhiều tài nguyên, module, và plugin có sẵn để sử dụng.

11.4 Face-api.js

Face-api.js là một Javascript API dùng để nhận diện gương mặt trên Browser cũng như NodeJS sử dụng Tensorflow.js.

Một số tính năng chính:

- Phát hiện khuôn mặt và nhận diện khuôn mặt
- Xác thực gương mặt
- Nhận diện cảm xúc của gương mặt
- Dự đoán tuổi và nhận diện giới tính



12 Kết luận

Trong bài tập lớn này, dưới sự hướng dẫn của cô, nhóm đã được những nhiệm vụ sau:

- Xây dựng app điểm danh nhận diện khuôn mặt kết hợp với hệ thống IOT, cụ thể là công ty thông minh
- Biết sử dụng thêm các framework và các công nghệ như face-api, express,...
- Phát triển kỹ năng làm việc nhóm và sắp xếp thời gian hợp lý

Một số hướng phát triển thêm sau này của nhóm:

- Nghiên cứu phát triển thêm để tăng độ chính xác nhận diện khuôn mặt
- Hỗ trợ đa ngôn ngữ
- Phát triển tích hợp hệ thống đến với nhiều môi trường hơn như trường học,...

Tài liệu

- [1] Web Dev Simplified, *Build Real Time Face Detection With JavaScript*, doi:<https://github.com/WebDevSimplified/Face-Detection-JavaScript>
- [2] Web Dev Simplified, *Easy Face Recognition Tutorial With JavaScript*, doi:<https://github.com/WebDevSimplified/Face-Recognition-JavaScript>