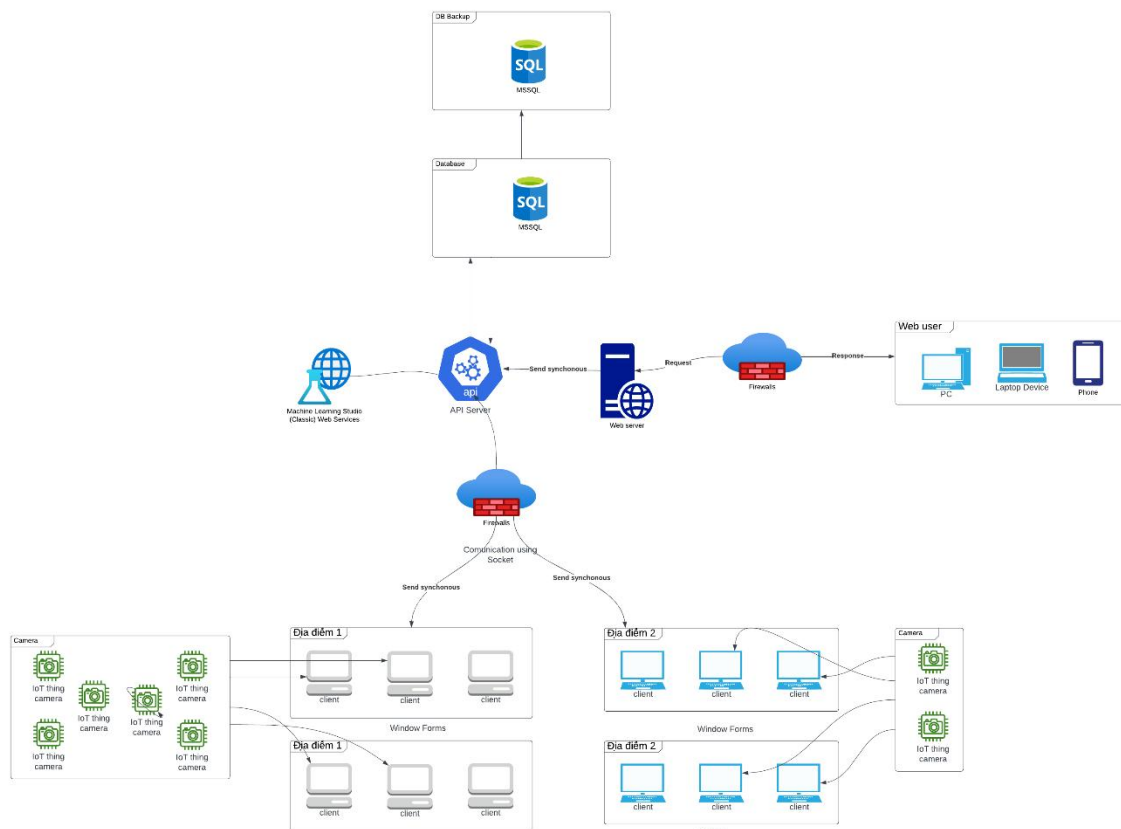


EXERCISE 2

Yêu cầu: Thiết kế một hệ thống vận hành hiệu quả và an toàn cho 200 edge devices (chạy hệ điều hành ubuntu) được đặt tại 200 địa điểm khác nhau. Hệ thống cần đáp ứng các yêu cầu sau:

- Tự động update, revert code cho các edge device một cách tự động và hiệu quả.
- Giám sát, quản lý trạng thái, hoạt động, tài nguyên cho các Edge Device theo thời gian thực.
- Khắc phục sự cố và bảo trì các edge device từ xa.
- Đảm bảo an ninh mạng cho hệ thống và dữ liệu.
- Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu tăng trưởng số lượng edge device trong tương lai.
- Hệ thống phải dễ sử dụng và quản lý.
- Hệ thống phải hoạt động ổn định và an toàn.

1. Design work flow



Hình 1: thiết kế hệ thống

2. Cách thức hoạt động của hệ thống

Hệ thống này sẽ gồm 4 phần cốt lõi đó là BE, FE server, database và client là các edge devices (Jetson), để đáp ứng các yêu cầu trên ta cần thiết kế và xây dựng kiến trúc như sau.

BE:

- Xây dựng socket server để các client có thể kết nối GET và POST dữ liệu
- Lấy thông tin trạng thái, hoạt động, tài nguyên của các edge device qua kênh socket.
- Viết API để đồng bộ các cấu hình cho các edge device khác nhau. Ví dụ ta có chương trình nhận diện khuôn mặt trên Jetson, ta muốn cấu hình thời gian nào mới bật nhận diện lên, ta sẽ thực hiện điều này trên website FE sẽ gọi API từ đó API server gửi data qua socket để nhận config mới. Điều này sẽ thuận tiện cho ta custom nhiều thứ trên edge device 1 cách từ xa. Trong trường hợp 1 vài thiết bị offline thì khi hoạt động lại nó sẽ tự gọi API để kiểm tra thông tin mới nhất và đồng bộ về máy.
- Xây dựng API để quản lý thông tin device, sẽ lưu 1 số thông tin quan trọng như IP, SSH, trạng thái, tài nguyên, vị trí thiết bị, ... Để khi có sự cố về phần mềm ta có thể teamview hay ssh và khắc phục kịp thời. Về vấn đề phần cứng ta biết được vị trí thiết bị đã lưu ta có thể đưa ra phương án đối ứng kịp thời.
- Sử dụng Fast API, API .NET hoặc Nodejs để xây dựng API

FE:

- Sử dụng các framework như React hoặc Angular để xây dựng giao diện người dùng để sử dụng và hiệu quả.
- Thiết kế các trang web để quản lý và theo dõi trạng thái của các thiết bị edge.
- Thiết kế giao diện có thể sử dụng trên nhiều nền tảng khác nhau, bao gồm máy tính, điện thoại di động và máy tính bảng.

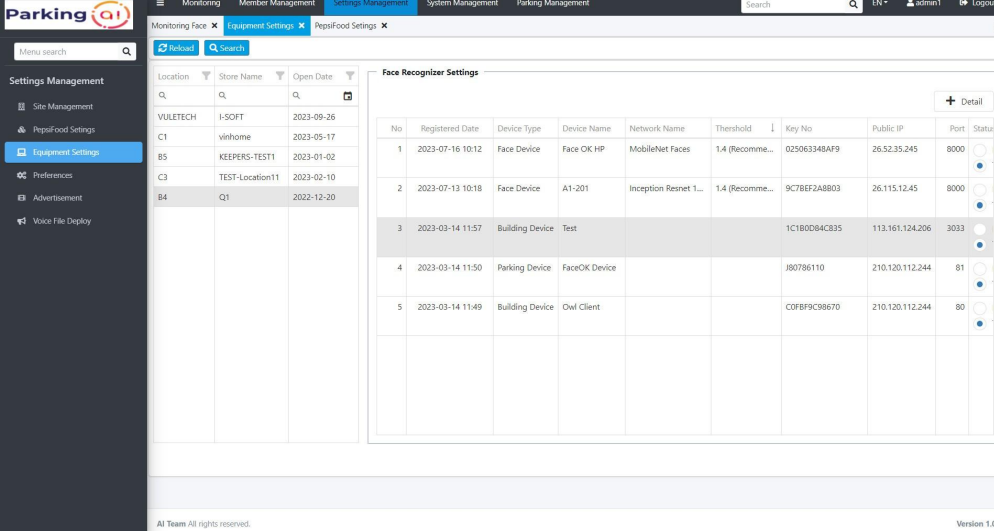
Database:

Bảng này đại diện cho chi nhánh triển khai device	Chứa thông tin của các thiết bị	Dùng để lưu các config cho device	Lưu thông tin người quản lý chi nhánh
 <p>tblStoreMaster</p> <ul style="list-style-type: none">StoreNoLocationStoreNameBitNumberZipCodeAddress1Address2BitPhoneNumberOpenDateRegistDateStatusCapacityMonitoringStartTimeMonitoringEndTimeMinorAgeOnClientHomeSoundClientHomeSoundMemoSiteType	 <p>tblStoreDevice</p> <ul style="list-style-type: none">StoreDeviceNoStoreNoDeviceNameDeviceTypeDeviceKeyNoDevicePublicIPDeviceUsePortDeviceStatusIDPathRegistUseridRegistDateDeviceKeyUseDeviceKeyNoModel	 <p>tblStoreEnvironmentSetting</p> <ul style="list-style-type: none">EnvironmentSettingNoStoreNoCertifCriteriaPhoneInputSimilarityRateApprovalAuthAfterCompletedAuthAfterCardIdDidUseScannerUseCamera	 <p>tblUserMgtStore</p> <ul style="list-style-type: none">UserNoStoreNoMemoRegistDate

Client (Jetson):

- Containerization: sử dụng Docker để container hóa các ứng dụng và Kubernetes để quản lý các containers. Điều này giúp việc triển khai, cập nhật và quản lý ứng dụng trở nên dễ dàng và linh hoạt hơn, đồng thời tăng cường khả năng mở rộng và phục hồi từ sự cố.
- Jtop: ta dùng thư viện này để lấy ra các thông tin tài nguyên như CPU, GPU và nhiệt độ của edge devices (Jetson)
- Sử dụng git: hỗ trợ việc triển khai code mới và revert code một cách linh hoạt và tự động. Ví dụ khi ta có bất kỳ thay đổi về code, API sẽ gửi socket cần update chương trình, sau đó thiết bị sẽ tự chạy script được config sẵn để update code từ git, build và chạy chương trình lên

Khi triển khai trang web ta sẽ dễ dàng quản lý khi số lượng các device tăng lên, ta thiết kế database phù hợp để quản lý chặt chẽ. Lấy hình ảnh bên dưới là ví dụ, ta sẽ có trang quản lý quản các chi nhánh triển khai, ở mỗi chi nhánh sẽ có nhiều device và ta cần assign 1 người quản lý thiết bị ở chi nhánh, khi thấy bất kỳ sự cố quản lý chi nhánh sẽ liên lạc với team dev để kịp thời xử lý hoặc người quản lý là người developer có thể tự xử lý lỗi. Mỗi người sẽ quản lý quy mô nhỏ để dễ dàng quản lý thiết bị khi số lượng device tăng.



No	Registered Date	Device Type	Device Name	Network Name	Threshold	Key No	Public IP	Port	Status
1	2023-07-16 10:12	Face Device	Face OK HP	MobileNet Faces	1.4 (Recomm...	025063348AF9	26.52.35.245	8000	<input type="radio"/>
2	2023-07-13 10:18	Face Device	A1-201	Inception Resnet 1...	1.4 (Recomm...	9C78EF2A8B03	26.115.12.45	8000	<input type="radio"/>
3	2023-03-14 11:57	Building Device	Test			1C180D84C835	113.161.124.206	3033	<input type="radio"/>
4	2023-03-14 11:50	Parking Device	FaceOK Device			J80786110	210.120.112.244	81	<input type="radio"/>
5	2023-03-14 11:49	Building Device	Owl Client			C0F8F9C98670	210.120.112.244	80	<input type="radio"/>

3. An Ninh Mạng

Tường lửa và IDS/IPS: Thiết lập tường lửa trên mỗi edge device và sử dụng Hệ thống phát hiện và ngăn chặn xâm nhập (IDS/IPS) để phát hiện và ngăn chặn các mối đe dọa.

VPN và Mã hóa: Sử dụng VPN để tạo kênh truyền dữ liệu an toàn giữa trung tâm điều khiển và các edge devices. Mã hóa dữ liệu cả trong quá trình truyền tải và lưu trữ. Dự vào hình 1, để dữ liệu có thể đi ra bên ngoài cần phải đi qua tường lửa

Reference

Video

[1]

https://drive.google.com/drive/folders/1SEkYidJIFY4unfC0FRbSUWm0OnC1AGwK?usp=drive_link

Document

[2] [https://docs.google.com/document/d/11Al0lqAFaj-](https://docs.google.com/document/d/11Al0lqAFaj-SYAFI03qNQnONooecyaOh/edit?usp=sharing&oid=112511733103624990964&rtpof=true&sd=true)

[SYAFI03qNQnONooecyaOh/edit?usp=sharing&oid=112511733103624990964&rtpof=true&sd=true](https://docs.google.com/document/d/11Al0lqAFaj-SYAFI03qNQnONooecyaOh/edit?usp=sharing&oid=112511733103624990964&rtpof=true&sd=true)