**LỜI MỞ ĐẦU**

*Công nghệ thông tin là một trong những ngành phát triển vượt bậc trong những*

*năm gần đây. Ngày nay với sự phát triển nhanh chóng của xã hội thì công nghệ thông*

*tin được ứng dụng rộng rãi ở hầu hết tất cả các lĩnh vực và ngày càng đóng vai trò*

*quan trọng, trở thành một phần thiết yếu trong đời sống hằng ngày. Công nghệ thông*

*tin là một ngành đòi hỏi ở người học một nền tảng kiến thức vững chắc, sự tư duy*

*logic cao, hiểu biết sâu rộng trên nhiều lĩnh vực. Với chúng em hiện đang là những*

*sinh viên công nghệ thông tin cần phải có sự đầu tư, không ngừng học hỏi để nâng cao*

*kiến thức. Do đó để củng cố lại kiến thức đã học, đề tài mà em chọn để thực hiện đồ*

*án LTHT và VDK là “Hệ thống chấm công và mở khóa dùng nhận dạng vân tay”.*

*Thông qua quá trình thực hiện đồ án, chúng em đã nắm bắt được những kỹ thuật*

*quan trọng của việc xây dựng code, xử lí mạch, lắp ráp và hoàn thiện sản phẩm. Chúng*

*em xin chân thành cảm ơn thầy Ninh Khánh Duy đã bổ sung ý tưởng và tận tình giúp đỡ*

*chúng em thực hiện đồ án này.*

*Trong quá trình thực hiện, không tránh khỏi sai sót. Chúng em rất mong nhận được sự góp ý từ phía thầy cô để bài làm của chúng em được hoàn thiện hơn.*

*Đà Nẵng, tháng 5 năm 2019*

**TÓM TẮT ĐỒ ÁN**

“Hệ thống chấm công và mở khóa dùng nhận dạng vân tay” nhằm giải quyết vấn đề thực tế hiện nay của các công ty đó là: cần 1 hệ thống chỉ cho phép nhân viên trong công ty mở cửa, đảm bảo tính an toàn của môi trường làm việc chuyên nghiệp, bên cạnh đó kết hợp chức năng chấm công để dễ dàng quản lý nhân viên. Phương án giải quyết là sử dụng bộ cảm biến vân tay kết hợp với hệ thống quản lý cơ sở dữ liệu firebase và chấm công thông qua website. Sau khi nghiên cứu và thực hiện, nhóm đã hoàn thành được một hệ thống với các chức năng cơ bản: mở cửa bằng vân tay, thêm vân tay mới, xóa vân tay.

**I. Giới thiệu đề tài**

**Tên đề tài: Hệ thống chấm công và mở khóa dùng nhận dạng vân tay**

**1. Đặt vấn đề**

Hệ thống mở khóa kết hợp chấm công dùng nhận dạng vân tay là một hệ thống giúp cho công tác chấm công trong các công ty được thực hiện một cách nhanh chóng, chính xác và tiết kiệm được thời gian. Việc chấm công bằng vân tay đã giúp rất nhiều doanh nghiệp giải quyết được vấn đề: thống kê chính xác tình trạng đi muộn về sớm, đảm bảo tuyệt đối, tiện lợi hơn cách tính công truyền thống. Ngoài ra việc sử dụng nó tạo thói quen nguyên tắc tuân thủ thời gian làm việc của công ty với ý thức làm việc chuyên nghiệp. Với việc vân tay của mỗi người là duy nhất, hệ thống còn đảm bảo về tính chính xác của các lần chấm công.

Cụ thể, mỗi ngày đi làm, nhân viên chỉ cần đặt tay vào phần cảm biến để hệ thống nhận dạng, hệ thống sẽ tự động chấm công và lưu kết quả vào cơ sở dữ liệu tương ứng đồng thời mở cửa. Dữ liệu này sẽ được thu thập lại phục vụ cho việc quản lí của công ty.

**2. Kết quả sản phẩm hiện có**

Hiện nay trên thị trường có rất nhiều loại máy chấm công bằng vân tay với giá cả, chất lượng và mẫu mã đa dạng. Các loại máy trên đều đi kèm với hệ thống website để quản lý về thời gian đi làm của nhân viên. Hầu hết các loại máy đó đều có thể sử dụng với công ty có số lượng nhân viên lớn, khoảng vài nghìn người.

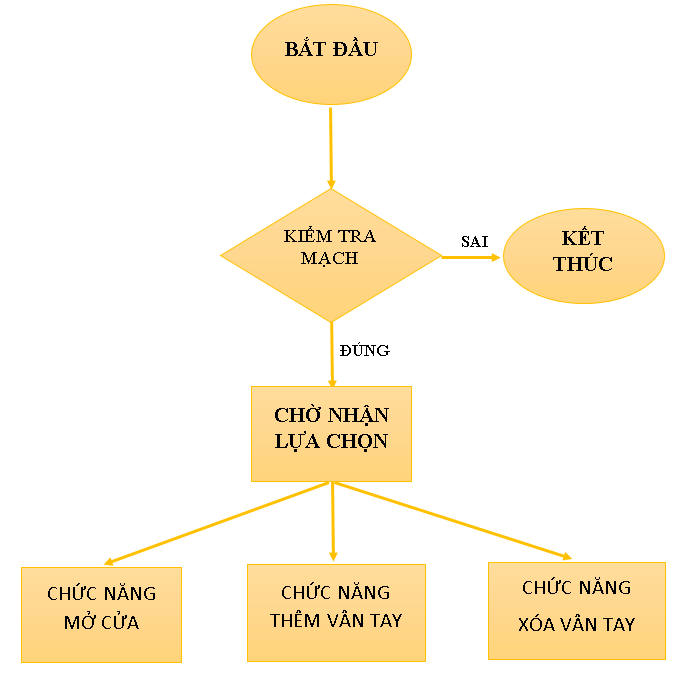
**3. Đề xuất giải pháp**

Trong phạm vi đồ án môn học này, chúng em chỉ thiết kế hệ thống sử dụng các linh kiện có giá cả vừa phải, tính ổn định và chính xác tương đối. Tuy nhiên vẫn đảm bảo được các chức năng cơ bản có thể áp dụng cho các cửa hàng vừa và nhỏ với số lượng nhân viên khoảng vài chục người. Hiện nay trên thị trường đa số các loại thiết bị này chỉ có nhiệm vụ chấm công. Qua quá trình tìm hiểu nhóm chúng em quyết định kết hợp chức năng mở khóa để tăng tính ứng dụng cho sản phẩm trên thực tế.

**II. Phân tích đề tài**

**1. Chức năng của hệ thống**

Sau khi hoàn thành đồ án nhóm chúng em đã tạo ra được một hệ thống thực hiện chấm công và mở khóa dùng nhận dạng vân tay với các chức năng: quét vân tay để mở cửa đồng thời lưu lịch sử phục vụ cho việc chấm công nếu vân tay đã được đăng ký trong hệ thống; thêm vân tay mới vào hệ thống sau đó cập nhật thông tin cá nhân; xóa vân tay.

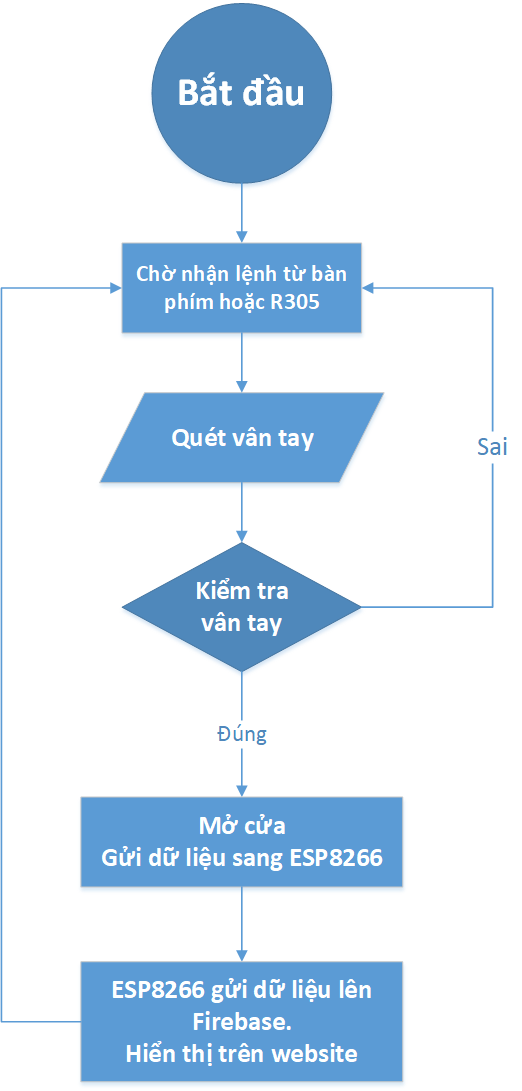
Sơ đồ khối giải thuật tổng quát như sau:

Hình 1. Sơ đồ khối giải thuật tổng quát

**1.1. Chức năng mở cửa**

Muốn mở cửa công ty nhân viên phải đảm bảo đã đăng ký vân tay trên hệ thống trước đó. Khi đi làm nhân viên sẽ đặt tay lên cảm biến, cảm biến sẽ tiến hành quét hình ảnh vân tay sau đó so sánh với dữ liệu trong bộ nhớ. Nếu có vân tay trùng khớp thì màn hình hiển thị thông báo mở cửa và loa báo hiệu đồng thời mở khóa điện. Cùng lúc đó ID của nhân viên sẽ được gửi từ cảm biến vân tay lên cơ sở dữ liệu thông qua module wifi. Bên cạnh đó module wifi sẽ lấy dữ liệu thời gian ngay lúc mở cửa để lưu vào lịch sử.

Sơ đồ khối giải thuật như sau:

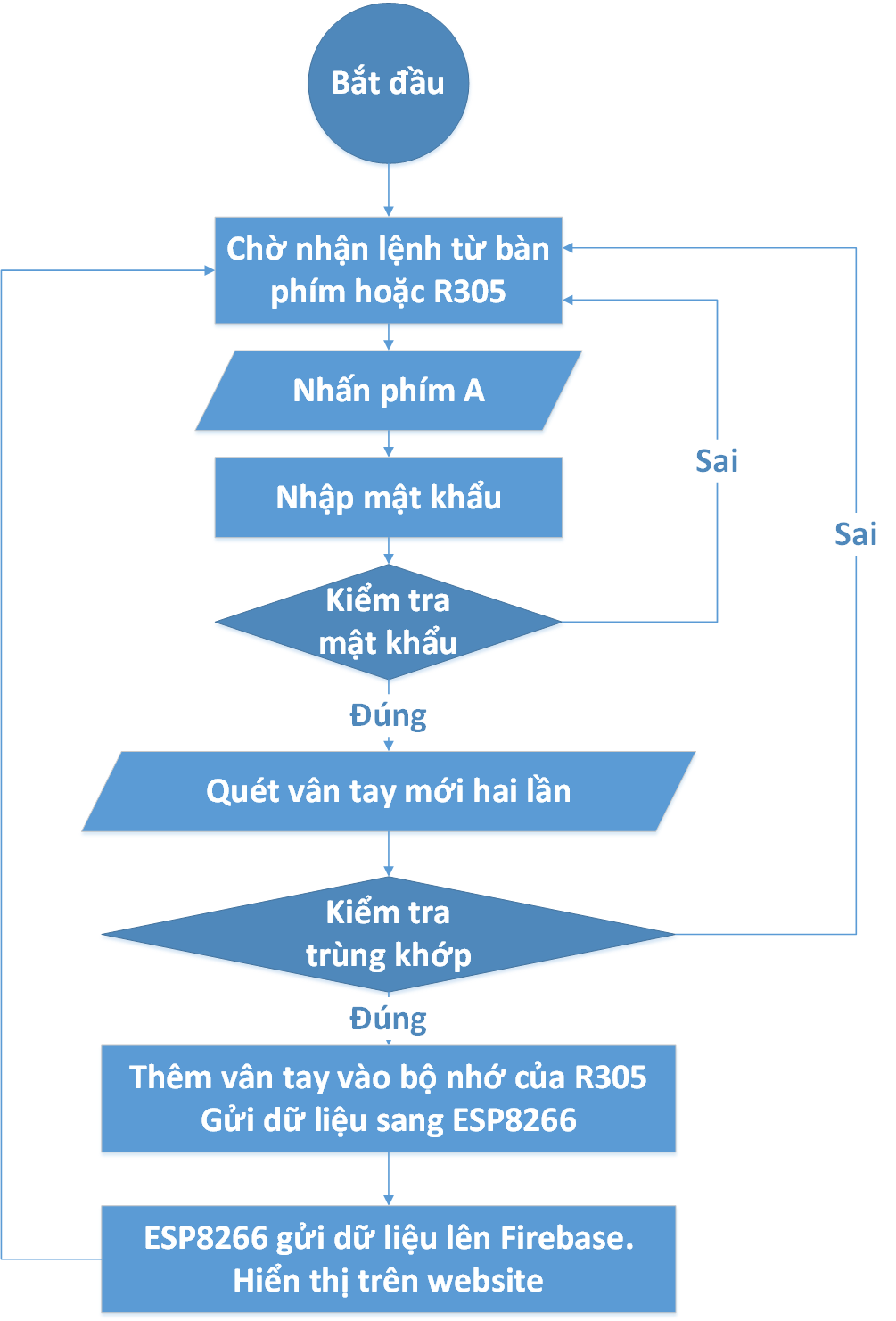


Hình 2. Sơ đồ khối giải thuật chức năng mở cửa

**1.2. Chức năng thêm**

Trong trường hợp nhân viên mới vào công ty thì cần phải đăng ký vân tay mới vào hệ thống. Khi đó nhân viên ấn phím “A” trên bàn phím rồi nhập mật khẩu mà người quản lý cung cấp để thêm. Hệ thống sẽ yêu cầu xác nhận vân tay hai lần để đảm bảo tính chính xác. Sau đó nhân viên sẽ được cấp một ID tự động, quản lý sẽ lên website nhập thông tin cá nhân cho nhân viên đó dựa vào ID. Vậy là đã hoàn thành việc đăng ký vân tay mới, nhân viên sẽ mở cửa thành công.

Sơ đồ khối giải thuật như sau:



Hình 3. Sơ đồ khối giải thuật chức năng thêm

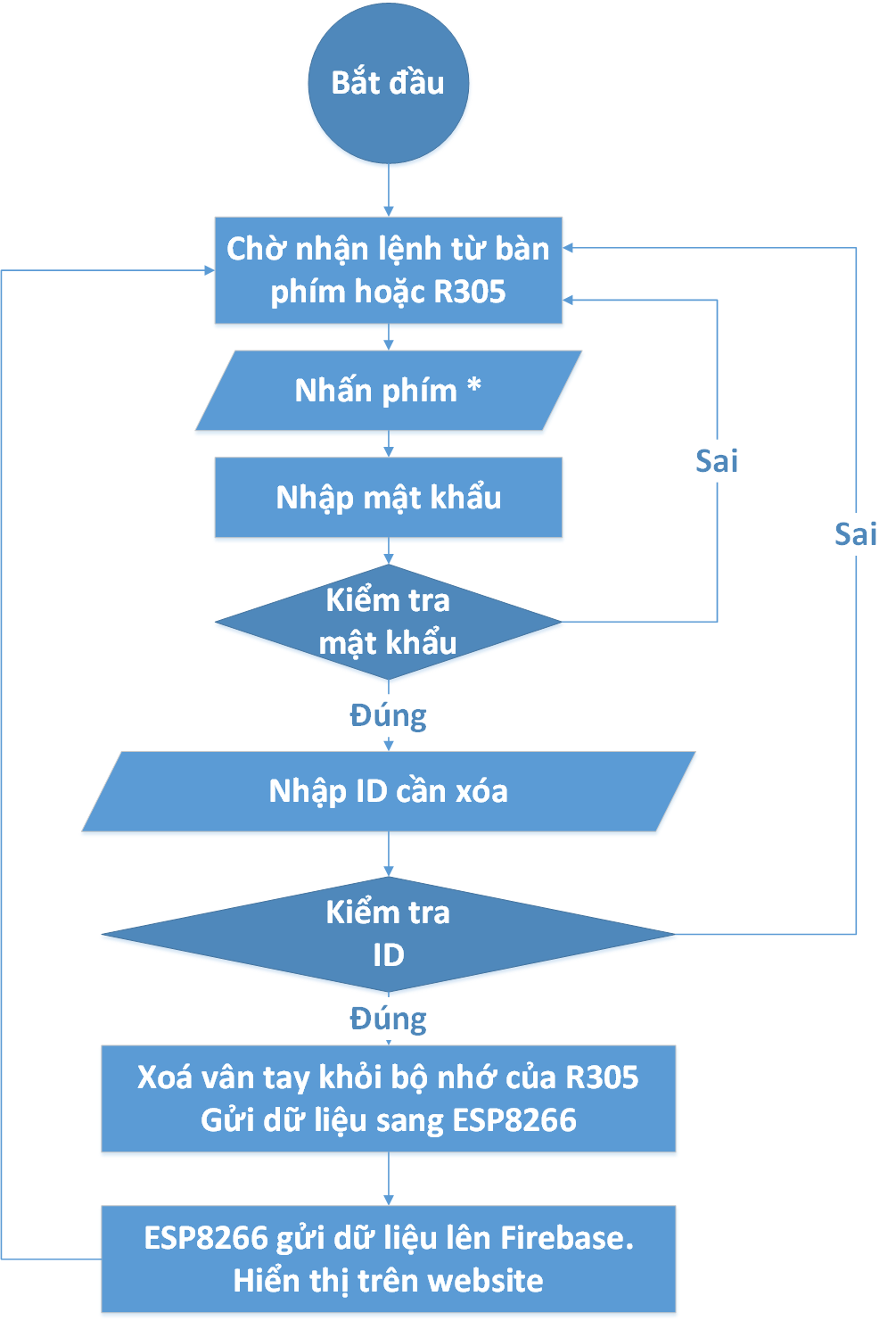
**1.3. Chức năng xóa**

Đối với chức năng xóa, có 2 lựa chọn:

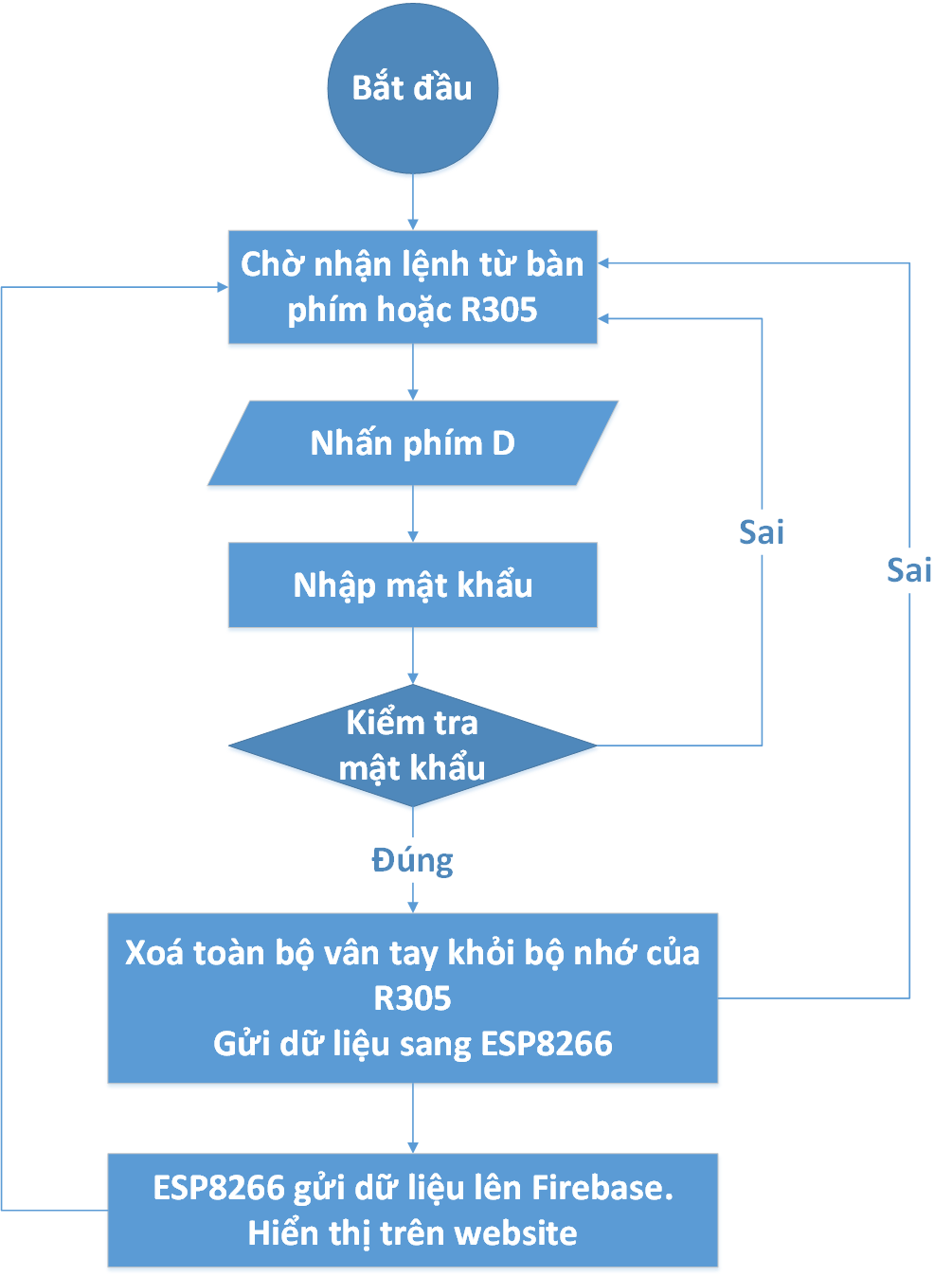
- Xóa một vân tay: muốn xóa một vân tay, quản lý sẽ thực hiện ấn phím “\*” trên bàn phím sau đó hệ thống sẽ yêu cầu nhập mật khẩu (chỉ có người quản lý mới biết mật khẩu). Nếu mật khẩu đúng thì thực hiện xóa bằng cách nhập ID muốn xóa.

- Xóa tất cả vân tay: muốn xóa tất cả vân tay, quản lý thực hiện ấn phím “D” trên bàn phím sau đó nhập mật khẩu, nếu mật khẩu đúng thì việc xóa tất cả vân tay đã thành công.

Sơ đồ khối giải thuật như sau:



Hình 4. Sơ đồ khối giải thuật chức năng xóa một vân tay



Hình 5. Sơ đồ khối giải thuật chức năng xóa tất cả vân tay

**2. Các linh kiện sử dụng**

Từ những phân tích và mô hình tổng quan ở phần trước, ta tiến hành lựa chọn các linh kiện phần cứng sau:

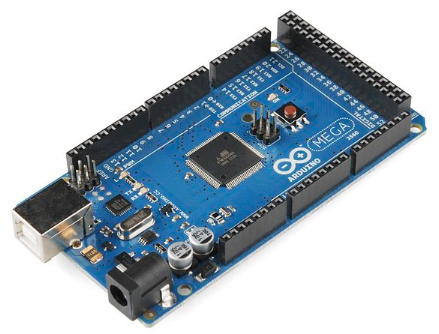
**2.1. Board mạch Arduino Mega 2560**

Board Arduino Mega 2560 là một phần quan trọng của hệ thống, nó sẽ điều chỉnh mọi hoạt động của hệ thống bao gồm giao tiếp với cảm biến vân tay thông qua thư viện Adafruit\_FingerPrint - thư viện này cung cấp các giao tiếp cho phép bạn kiểm tra vân tay, đăng kí lưu trữ vân tay, xóa vân tay. Ngoài ra board Arduino Mega2560 còn giao tiếp với module wifi thông qua thư viện SoftwareSerial để gởi dữ liệu ID vân tay, kích hoạt loa, ghi nhận dữ liệu từ bàn phím, đưa kết quả đã nhận ra màn hình, điều khiển khóa điện thông qua các hàm tự viết.

**Thông số kỹ thuật:**

Bảng 1. Thông số kỹ thuật của Arduino Mega 2560

|  |  |
| --- | --- |
| Chip xử lý | ATmega2560 |
| Điện áp hoạt động | 5V |
| Điện áp vào (đề nghị) | 7V-15V |
| Điện áp vào (giới hạn) | 6V-20V |
| Cường độ dòng điện trên mỗi 3.3V pin | 50 mA |
| Cường độ dòng điện trên mỗi I/O pin | 20 mA |
| Flash Memory | 256 KB |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Clock Speed | 16 MHz |



Hình 6. Arduino Mega 2560

**2.2. Cảm biến vân tay R305**

Nguyên lý hoạt động của module cảm biến vân tay cơ bản gồm 2 phần:

- Lấy dữ liệu hình ảnh của vân tay: Khi lấy dữ liệu, module sẽ lấy dữ liệu hình ảnh vân tay 2 lần thông qua cảm biến quang học và xử lý 2 hình ảnh này để tạo ra một mẫu.

- Kết hợp các dữ liệu để tạo ra mẫu vân tay: Quá trình này xảy ra sau khi đã có hình ảnh vân tay. Đây là quá trình kết hợp 2 mẫu hình ảnh vân tay để tạo ra một mẫu. Hệ thống sẽ xử lý để lưu trữ vào thư viện vân tay của module.

Sau khi người dùng quét vân tay, cảm biến sẽ thực hiện giao tiếp với Arduino thông qua các thư viện đã nói ở phần trên và thực hiện tác vụ tương ứng.

**Thông số kỹ thuật:**

Bảng 2. Thông số kỹ thuật của R305

|  |  |
| --- | --- |
| Nguồn cấp | 3.6 - 6VDC |
| Giao tiếp | TTL-UART hoặc USB 1.1 |
| Dòng điện hoạt động | 100 - 150mA |
| Chế độ nhận dạng | 1:1 hoặc 1:N |
| Baudrate | 9600xN bps (mặc định N=6 tức 9600x6 = 57600bps) |
| Thời gian nhận dạng trung bình | < 0.8s (1:880) |

Hình 7. Cảm biến vân tay R305

**2.3. ESP8266**

ESP 8266 đóng vai trò đưa dữ liệu lên cơ sở dữ liệu để quản lý chúng. Module wifi này sẽ kết nối với mạng wifi (có tên và mật khẩu do chúng ta cung cấp) thông qua thư viện ESP8266WiFi. Ngoài ra nó sẽ nhận dữ liệu là ID của vân tay từ Arduino gửi qua (thông qua thư viện SoftwareSerial) và gửi ID đó lên cơ sở dữ liệu Firebase (thông qua thư viện FirebaseArduino). Vấn đề quan trọng trong việc chấm công là phải lấy chính xác thời gian thực khi nhân viên quét vân tay mở cửa, vì vậy, ta dùng thư viện NTPTimeESP để lấy thời gian thực từ module wifi và gửi lên cơ sở dữ liệu.

**Thông số kỹ thuật:**

Bảng 3. Thông số kỹ thuật của ESP8266

|  |  |
| --- | --- |
| WiFi | 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n |
| Điện áp hoạt động | 3.3V |
| Điện áp vào | 5V thông qua cổng USB |
| Số chân I/O | 11 |
| Số chân Analog Input | 1 (điện áp vào tối đa 3.3V) |
| Bộ nhớ Flash | 4MB |
| Giao tiếp | Cable Micro USB ( tương đương cáp sạc điện thoại ) |

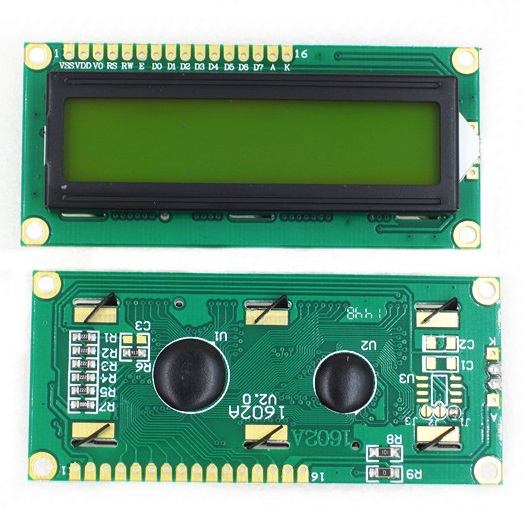
****

Hình 8. ESP8266

**2.4. Màn hình LCD 1602**

Màn hình LCD giao tiếp với Arduino (thông qua thư viện LyquidCrystal). Arduino điều khiển LCD dựa vào hàm xác định vị trí muốn hiển thị thông báo với 2 tham số là vị trí hàng (2 hàng) và cột (16 cột), LCD sẽ nhận lệnh và hiển thị các thông báo lên màn hình một cách chính xác về vị trí và nội dung ứng với các tác vụ tương ứng.

**Thông số kỹ thuật:**

Bảng 4. Thông số kỹ thuật của LCD 1602

|  |  |
| --- | --- |
| Kích thước | 80x36mm |
| Điện áp hoạt động | 3.3VDC |
| Chế độ hiển thị | STN, blub |
| Hiển thị | 16 ký tự x 2 dòng |
| Dữ liệu đầu vào | 4-bit hoặc 8-bit giao diện có sẵn |
| Hiển thị Font | 5 x 8 Dots |
| Nguồn cung cấp | (5V ± 10%) |

Hình 9. Màn hình LCD 1602

**2.5. Keypad 4x4**

Keypad 4x4 có đến 16 nút nhấn, nghĩa là nếu làm một cách thông thường (dùng chân digital) thì chúng ta phải cần đến 16 chân Arduino để đọc. Nhưng với bàn phím này, chúng ta chỉ cần dùng 8 chân (4 chân hàng ngang (row) và 4 chân cột dọc (column)).

Để kiểm tra một nút có được nhấn hay không, ta sẽ sử dụng hàm getKey() trong thư viện KeyPad. Thuật toán được sử dụng là thuật toán quét phím được mô tả như sau:

Với mỗi hàng (R1 đến R4), chọn ra hàng Ri, cấp cực âm (0v) cho hàng Ri:

- Nếu điện áp ở chân Cj bất kì là dương => chưa nhấn

- Nếu điện áp ở chân Cj bất kì là âm => đang nhấn

Sau khi bấm, dữ liệu về phím đó sẽ được gởi đến Arudino, Arduino sẽ xử lý và thực hiện tác vụ tương ứng với phím được bấm.

**Thông số kỹ thuật:**

Bảng 5. Thông số kỹ thuật của Keypad 4x4

|  |  |
| --- | --- |
| Độ dài cáp | 88mm |
| Nhiệt độ hoạt động | 0 ~ 70 |
| Kích thước bàn phím | 77 x 69 mm |
| Đầu nối ra 8 chân | |

Hình 10. Keypad 4x4

**2.6. Còi Buzzer**

Còi được kết nối tới Arduido và có nhiệm vụ phát âm thanh thông báo khi việc kiểm tra vân tay đúng hoặc sai. Arduino điều khiển hoạt động của loa theo hàm tự viết, có thể thay đổi giai điệu theo nhu cầu.

**Thông số kỹ thuật:**

Bảng 6. Thông số kỹ thuật của còi Buzzer

|  |  |
| --- | --- |
| Điện áp hoạt động | 3 ~ 24VDC |
| Dòng điện tiêu thụ | 30mA 12V, 40mA 24V |
| Biên độ âm thanh | lên đến 100dB |
| Kích thước | 27 x 15mm |
| Khối lượng | 10g |

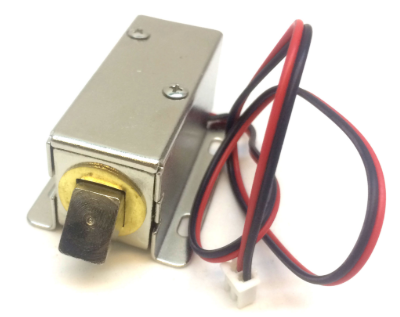
Hình 11. Còi Buzzer

**2.7. Khóa điện và transistor**

Sử dụng khóa điện để mô phỏng cho việc đóng mở cửa. Khi vân tay đúng set chân LOCK\_PIN (chân số 4 của Arduino) là HIGH để mở cửa. Cửa được mở sau 2 giây thì set chân LOCK\_PIN (chân số 4 của Arduino) là LOW để đóng cửa.

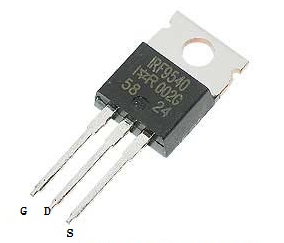
Khóa được nối với nguồn, tuy nhiên do khóa điện hoạt động trong khoảng 12V nên ta cần một con transistor để khuếch đại dòng điện đủ cho khóa điện hoạt động.

**Thông số kỹ thuật:**

Bảng 7. Thông số kỹ thuật của khóa điện

|  |  |
| --- | --- |
| Vật liệu | Thép không gỉ |
| Nguồn điện | 12V DC |
| Dòng điện làm việc | 0.8A |
| Công suất | 9.6W |
| Yêu cầu nguồn cấp | 12VDC/1A |

Hình 12. Khóa điện

Bảng 8. Cách nối các chân của transistor

|  |  |
| --- | --- |
| Transistor |  |
| Chân G | Chân 4 của Arduino Mega 2560 |
| Chân D | Dây âm của khóa điện |
| Chân S | GND |

Hình 13. Transistor

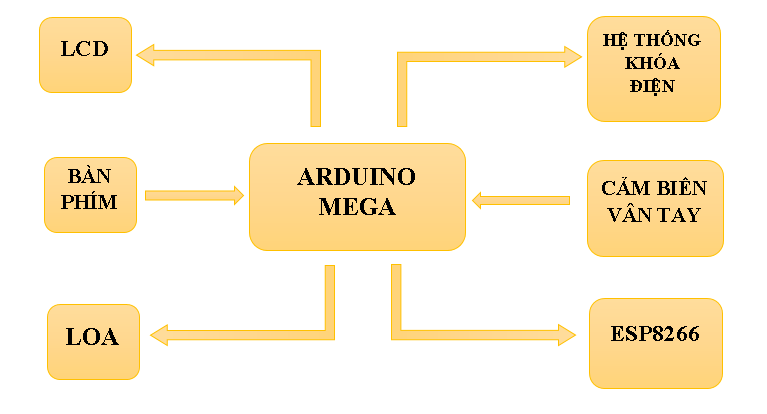
**3. Bảng kê chi phí linh kiện**

Bảng 9. Chi phí linh kiện

|  |  |
| --- | --- |
| Tên linh kiện | Giá thành |
| Arduino Mega 2560 | 180.000 |
| Cảm biến vân tay R305 | 590.000 |
| ESP8266 | 110.000 |
| Màn hình LCD 1602 | 30.000 |
| Keypad 4x4 | 18.000 |
| Còi Buzzer | 12.000 |
| Khóa điện | 110.000 |
| Transistor | 3.000 |
| Breadboard | 18.000 |
| Xốp mô hình | 40.000 |
| Dây nối | 20.000 |
|  | **Tổng cộng: 1.131.000** |

**III. Giải pháp triển khai**

**1. Kết nối giữa các module và nguyên tắc hoạt động**

****Các module chính của hệ thống được kết nối như hình dưới đây:

Hình 14. Sơ đồ ghép nối các module

* Hệ thống gồm 3 phần chính:
* Arduino Mega : kết nối và quản lí các module khác với nhau.
* ESP8266 NodeMCU : đọc dữ liệu gửi từ mega, kết nối wifi và gửi dữ liệu lên cơ sở dữ liệu Firebase.
* Các module khác:
  + Thiết bị đầu vào:
* R305: Quét và nhận hình ảnh vân tay (nối với pin 12,13 của mega)
* Bàn phím: Nhận các kí tự số và chữ cái (0…9, A, B, C, D, \*, #) (Các pin theo hàng ngang và dọc là {46, 47, 48, 49} và {50, 51, 52, 53} )
  + Thiết bị đầu ra:
* LCD: Hiển thị các yêu cầu và kết quả (pin 41, 39, 37, 35, 33, 31)
* Loa: Phát âm thanh thông báo (đúng hoặc sai) (pin 5)
* Khóa điện: đọc và mở cửa (pin 4)
* Một số quy định gửi dữ liệu giữa Arduino Mega và ESP8266 NodeMCU:
* Arduino giao tiếp với một một ESP8266 NodeMCU thông qua cổng Serial ảo được tạo từ thư viện Software Serial. Ở đây cổng ảo của Arduino là SoftwareSerial Arduino(11, 10); và của ESP8266 là SoftwareSerial NodeMCU(D2,D3);
* Mỗi lần gửi dữ liệu, Arduino sẽ gửi hai biến là type (loại hoạt động) và ID (mã vân tay). Các biến được gửi ngăn cách nhau bởi kí tự xuống hàng. Các quy định nhận gửi để xử lí dữ liệu mà người code tự quy định như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Hoạt động | Type | ID |
| 1 | Thêm vân tay mới | 1 | ID lấy từ fingerprint |
| 2 | Xóa một vân tay | 2 | ID nhập từ bàn phím |
| 3 | Kiểm tra vân tay và mở cửa | 3 | ID lấy từ fingerprint |
| 4 | Xóa toàn bộ vân tay | 4 | ID cao nhất hiện có trong fingerprint |

* Với các nguyên lí hoạt động của từng module đã nêu ở mục II.2, chúng được Arduino Mega kết nối và quản lí như sau:
* Khi nạp nguồn cho hệ thống, Arduino Mega sẽ kiểm tra mạch nối giữa Arduino Mega và module cảm biến nhận dạng vân tay R305 (4 chân VCC, GND, TD, RD).
* Nếu cắm sai thì màn hình LCD sẽ hiển thị “Wrong Cicruit!”.
* Nếu cắm đúng thì màn hình LCD sẽ hiển thị “Correct Cicruit!”.
* Khi hệ thống đã sẵn sàng, màn hình LCD hiển thị “Smart Lock” là tên của hệ thống và “Waiting…” chờ nhận lệnh từ bàn phím hoặc chờ nhận mã quét vân tay từ R305.

**TH1: Kiểm tra vân tay**

Khi Arduino Mega không nhận lệnh nào từ bàn phím và vân tay được đưa vào cảm biến:

* Module cảm biến R305 sẽ trả hình ảnh vân tay mà nó quét được về một mã ID cụ thể nếu hình ảnh đó được tìm thấy bằng lệnh finger.fingerID , ngược lại trả về -1.
* Arduino Mega lấy mã ID đó bằng hàm getFingerprintIDez() để xử lí:

+ Nếu ID = -1 :

* Chuông báo âm sai bell(melody\_notFound, noteDurations\_notFound);
* LCD hiện thị “Not Found” bằng hàm Display(String msg, int pos)
* Gửi dữ liệu qua ESP gồm: type=3 và ID=0. Sử dụng hàm Arduino.print(ID)

+ Nếu ID khác -1:

* Chuông báo âm đúng bell(melody, noteDurations);
* LCD hiện thị “Hello #ID” và “Openning…”
* Mở rồi đóng khóa điện digitalWrite(LOCK\_PIN, HIGH); digitalWrite(LOCK\_PIN, LOW);
* Gửi dữ liệu qua ESP gồm: type=3 và ID.

**TH2: Thêm vân tay mới**

Khi Arduino Mega nhận lệnh A nào từ bàn phím: (hàm nhận keypad.getKey(); của thư viện "Keypad.h" )

* Hệ thống yêu cầu nhập Password (password : 0000)

+ Nếu sai password:

* LCD hiển thị “Wrong Password!”
* Kết thúc và quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

+ Nếu đúng password:

* LCD hiển thị “Correct Password!”
* Đến bước tiếp theo
* Yêu cầu quét và lấy vân tay hai lần, dùng hàm getFingerprintEnroll()

+ Nếu 2 lần lấy vân tay trùng khớp nhau:

* LCD hiển thị “Match!”.
* Gửi dữ liệu qua ESP gồm: type=1 và ID vân tay mới thêm vào (ID tăng dần sau mỗi lần thêm)
* Quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

+ Nếu không thành công:

* LCD hiển thị “Not Match!”
* Quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

**TH3: Xóa một vân tay**

Khi Arduino Mega nhận lệnh \* nào từ bàn phím: (hàm nhận keypad.getKey(); của thư viện "Keypad.h" )

* Hệ thống yêu cầu nhập Password (password : 0000)

+ Nếu sai password:

* LCD hiển thị “Wrong Password!”
* Kết thúc và quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

+ Nếu đúng password:

* LCD hiển thị “Correct Password!”
* Đến bước tiếp theo
* Yêu cầu nhập mã ID muốn xóa (kết thúc bằng kí tự #, ví dụ 2#, 17#)

+ Nếu ID có tồn tại trong bộ nhớ của module R305

* Mega xóa vân tay có ID đã nhập trong bộ nhớ của R305 bằng hàm finger.deleteModel(id);
* Gửi dữ liệu qua ESP gồm: type=2 và ID vân tay vừa xóa
* Quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

+ Nếu ID không tồn tại

* LCD hiển thị “ID not indefined!”
* Quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

**TH4: Xóa tất cả vân tay**

Khi Arduino Mega nhận lệnh D nào từ bàn phím: (hàm nhận keypad.getKey(); của thư viện "Keypad.h" )

* Hệ thống yêu cầu nhập Password (password : 0000)

+ Nếu sai password:

* LCD hiển thị “Wrong Password!”
* Kết thúc và quay lại màn hình chờ lệnh tiếp theo “Waiting … ”

+ Nếu đúng password:

* LCD hiển thị “Correct Password!”
* Mega sẽ xóa tất cả bộ nhớ của R305 bằng h àm finger.emptyDatabase();
* Gửi dữ liệu qua ESP gồm: type=4 và ID cao nhất có trong bộ nhớ của R305
* Ngoài ra, ta có thể nhấn phím “C” để kiểm tra số vân tay hiện có và ID cao nhất..

**2. Module ESP8266 và cơ sở dữ liệu Firebase**

Module ESP8266 sẽ nhận hai giá trị id và type từ Arduino Mega bằng đoạn mã:





Trong đó, NodeMCU là cổng Serial của ESP khi kết nối với Arduino:

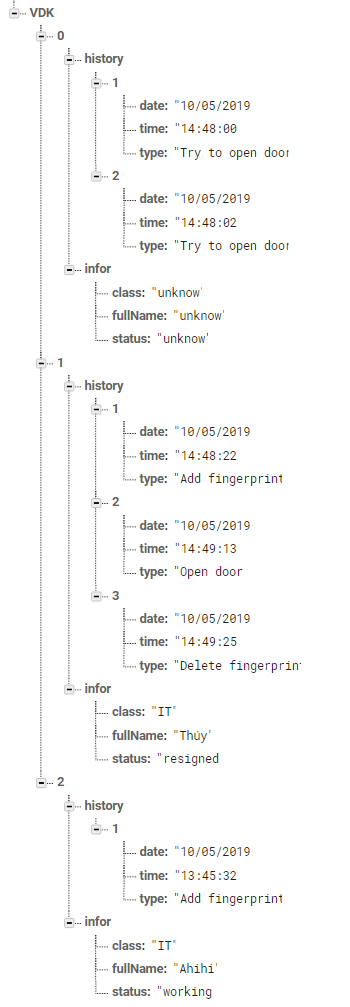
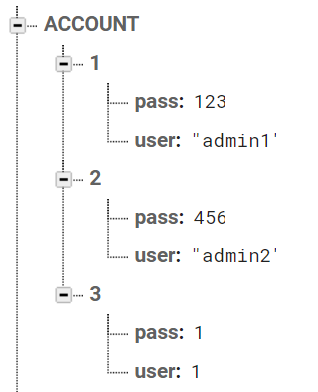


* id:
* id = 0: có người lạ xâm nhập vào hệ thống, đặt dữ liệu cho các trường Name, Department, Status là “unknow”; đặt thời gian cho các trường Date, Time; đặt dữ liệu cho trường Type là “Try to open door”.
* type:
* type = 1: tạo 1 id mới, đặt dữ liệu cho các trường Name, Department là “X” để quản lý đăng nhập vào website đặt lại dữ liệu theo thông tin mà nhân viên cung cấp; đặt dữ liệu cho trường Status và Type lần lượt là “working”, “Add fingerprint”; đặt Date, Time.
* type = 2: xóa 1 id, đặt lại dữ liệu cho trường Status là “resigned”; cho trường Type là “Delete fingerprint”; đặt Date, Time.
* type = 4: xóa tất cả id, duyệt qua tất cả các id có trong hệ thống và thực hiện giống như khi type = 2.
* type = 3 và id != 0: đã có id trong hệ thống, thực hiện mở cửa, lúc này đặt dữ liệu cho trường Type là “Open door”; đặt Date, Time để lưu lịch sử.

Việc đặt dữ liệu cho một trường nào đó trong Firebase được thực hiện bằng đoạn mã sau:

Trong đó, x là đường dẫn đến trường muốn đặt dữ liệu; VDK, history, type là các node trong Firebase.

Cơ sở dữ liệu được tạo trên Firebase có dạng json, toàn bộ dữ liệu là một json tree lớn cùng với nhiều điểm node.



Hình 15. Các node trong Firebase

Trong node ACCOUNT có các node pass và user cung cấp mật khẩu và tên đăng nhập để đăng nhập vào website thực hiện chức năng thêm nhân viên.

Trong node VDK có node đại diện cho id từng vân tay, trong đó sẽ có các node history lưu lại lịch sử các hoạt động được thực hiện trên hệ thống và node infor chứa thông tin của nhân viên.

Mỗi lần ta muốn truy xuất tới thông tin nào trong json tree, ta cần đi theo một đường dẫn chính xác từ node lớn nhất tới thông tin muốn truy xuất.

1. **Website quản lý:**

Website chia làm ba chức năng chính:

* Hiển thị dữ liệu
* Tìm kiếm dữ liệu
* Thêm dữ liệu
  1. *Hiển thị dữ liệu*
* Website có hai dạng table để hiện thị:
* Table “Lịch sử hoạt động” (History) bao gồm các trường:
* STT (Số thứ tự)
* ID, Name, Department (Mã số, Tên và Bộ phận làm việc của nhân viên thực hiện)
* Date, Time (Ngày tháng, thời gian hoạt động)
* Type (Loại hoạt động: thêm vân tay, mở cửa, xóa vân tay, người lạ xâm nhập)
* Table “Danh sách nhân viên kèm chấm công” (Employee) bao gồm các trường:
* ID, Name, Department (Mã số, Tên và Bộ phận làm việc của nhân viên)
* Số lượng ngày đi làm trong khoảng thời gian chấm công: chấm theo ngày, tháng, hoặc chấm hết tất cả
* Status (Trạng thái hiện tại của nhân viên: đang làm việc, nghỉ việc, người lạ)
* Để kết nối với Firebase, phải chỉ định URL Cơ sở dữ liệu thời gian thực khi khởi tạo SDK JavaScript bằng đoạn mã sau:

var config = {  
    apiKey: " ***AIzaSyDeHy73vDX957nCHNFQwPm2HnjpgIObA\_c***",  
    authDomain: " ***hanghang-2dc20.firebaseapp.com***",  
    databaseURL: " **https://hanghang-2dc20.firebaseio.com**",  
    projectId: "**hanghang-2dc20**",

storageBucket: " ***hanghang-2dc20.appspot.com***",

messagingSenderId: "**120693200632**"  
  };  
  firebase.initializeApp(config);

* Để đọc hoặc ghi dữ liệu từ cơ sở dữ liệu, bạn cần có một thể hiện của firebase.database.Reference:

// Get a reference to the database service

var database = firebase.database();

var database

Để đọc dữ liệu tại một đường dẫn của Firebase và lắng nghe các thay đổi, ta sử dụng phương thức on() của firebase.database.Reference để quan sát các sự kiện. Tuy nhiên trong website này, khi khởi tạo một thành phần UI, nó sẽ được giữ nguyên cho đến khi người dùng gọi nó lại lần nữa. Vì vậy ta hiển thị dữ liệu mà không cần lắng nghe các thay đổi. Ta sử dụng once() , nó kích hoạt một lần và sau đó không kích hoạt lại. Ta sử dụng sự kiện value để đọc nội dung dữ liệu tại một đường dẫn Firebase nhất định. Bạn có thể lấy lại dữ liệu bằng phương thức val().

var refVDK = firebase.database().ref('VDK/');  
refVDK.once('value', function(item\_list) {

item\_list.forEach(function(item){

console.log(item.val());

}  
});

Sau khi lấy được dữ liệu, ta cho hiển thị theo dạng Table.

Hình 16: Demo chức năng hiển thị tất cả lịch sử

* 1. *Tìm kiếm dữ liệu*

Tương tự như hiển thị website sẽ có hai dạng tìm kiếm theo hai loại table. Chức năng tìm kiếm cũng dựa trên các trường của từng table đã nêu ở trên. Ta sử dụng lại hàm hiển thị dữ liệu và có kèm theo một số điều kiện if else để trả về kết quả tìm kiếm. Có một số lưu ý cụ thể như sau:

* Nếu tất các cái trường đều trống hoặc không tìm được dữ liệu phù hợp, trả về kết quả “NO RESULT”
* Nếu có nhiều trường cùng được điền, kết quả tìm kiếm sẽ trả về dữ liệu mà thỏa mãn cùng lúc tất cả các trường đó.
* Trường Status chỉ dùng khi tìm kiếm Danh sách nhân viên và chấm công (Employee)
* Trường Type Activities chỉ dùng khi tìm kiếm lịch sử hoạt động (History)

Hình 17: Demo chức năng tìm kiếm( tìm nhân viên có Department là IT)

* 1. *Thêm dữ liệu*

Chức năng này được dùng khi muốn thêm một nhân viên mới vào hệ thống. Để sử dụng chức năng này, ta cần có tài khoản:

* Đầu tiên thêm vân tay tại module cảm biến vân tay R305
* Qua xử lí của phần cứng, mã số ID và hoạt động thêm vân tay của nhân viên mới này sẽ bổ sung trên Firebase và Website, nhưng lúc này chưa có thông tin về tên họ, bộ phận làm việc.
* Ta đăng nhập rồi cập nhập thông tin của nhân viên lên hệ thống.
* Nếu không điền đầy đủ các thông tin, hoặc điền sai ID, thao tác này sẽ không được thực hiện.

Thao tác update dữ liệu của Firebase dùng phương thức set() như sau:

firebase.database().ref('VDK/' + userId + '/infor').set({  
    fulName: keyName,  
    class: keyClass   
}



Hình 18: Demo chức năng thêm thông tin nhân viên

**IV. Kết quả đạt được**

Sau quá trình thực hiện đồ án, nhóm chúng em đã tạo ra được một hệ thống hoàn chỉnh, hoàn thành các chức năng đặt ra ban đầu.

* **Chức năng mở cửa**

Nhóm đã hoàn thành tốt chức năng này, độ chính xác, an toàn cao khi mà vân tay của mỗi người là duy nhất thì chỉ có các nhân viên của công ty đã đăng ký vân tay vào hệ thống mới có thể mở cửa được, loại bỏ được việc những người có ý đồ xấu lẻn vào công ty. Thêm vào đó việc mở cửa sẽ được hệ thống lưu vào lịch sử thông qua website để thực hiện chấm công. Nếu một ngày nhân viên mở cửa nhiều lần thì chỉ được tính là một lần chấm công.

* **Chức năng thêm vân tay**

Hoàn thành việc thêm một vân tay mới khi một nhân viên mới vào công ty làm việc

để nhân viên đó có thể thực hiện mở cửa đồng thời được chấm công.

* **Chức năng xóa vân tay**

Hoàn thành việc xóa một vân tay khi nhân viên nghỉ việc tránh cho việc nhân viên đó quay lại công ty với mục đích xấu; xóa toàn bộ vân tay nếu muốn reset lại hệ thống.

* **Website**

Hoàn thành website với các chức năng:

- Xem danh sách nhân viên kèm chấm công.

- Xem lịch sử hoạt động.

- Tìm kiếm nhân viên theo ID, tên, bộ phận, trạng thái.

- Tìm kiếm lịch sử hoạt động theo ID, tên, bộ phận, ngày, tháng, loại hoạt động.

- Chức năng đăng nhập đối với admin: Khi đăng nhập thành công sẽ có thêm chức năng thêm nhân viên. Khi nhân viên thực hiện thêm vân tay trên hệ thống, lúc này sẽ có một ID tương ứng với vân tay đó và quản lý sẽ dựa vào ID này để nhập thông tin cá nhân cho nhân viên.

**V. Đánh giá và kết luận**

1.Đánh giá hệ thống:

Hệ thống hoạt động ổn định, đáp ứng yêu cầu đề ra ban đầu.

- Cảm biến vân tay hoạt động tốt, khả năng nhận dạng vân tay nhanh (chưa tới 0.8s).

- Thời gian xử lý các tác vụ nhanh chóng và độ chính xác cao.

- Mô hình hệ thống đẹp, gọn gàng.

- Giao diện trực quan, cung cấp đầy đủ thông tin tới người dùng.

- Có hệ thống quản lý bằng website, giúp quản lý thực hiện chấm công nhanh chóng và dễ dàng truy cập mọi lúc mọi nơi.

- Có tính bảo mật khi có sự phân quyền cho website.

2.Kết luận:

Hệ thống sau khi hoàn thành đã thực hiện được các chức năng cơ bản. Tuy nhiên, tính ổn định của hệ thống chưa đảm bảo đáp ứng được yêu cầu khi đưa vào áp dụng thực tế ở các cửa hàng, đồng thời, số ID vân tay lưu được còn ít, không phù hợp ở các cơ quan có số lượng nhân viên lớn. Trong phạm vi đồ án môn học, nhóm em chỉ thực hiện bằng các linh kiện có qui mô chức năng nhỏ, vì vậy, để khắc phục các vấn đề trên, ta cần thay thế bằng linh kiện tốt hơn.

Qua đồ án này nhóm chúng em đã có thêm nhiều kỹ năng trong quá trình làm việc nhóm, được thực hành trực tiếp với Arduino Mega 2560 và các linh kiện được sử dụng, qua đó trau dồi thêm kiến thức về hai học phần lập trình hệ thống và học phần vi điều khiển đã được học.

**TÀI LIỆU THAM KHẢO**

**-** <http://arduino.vn/bai-viet/1172-lap-trinh-esp8266-bang-arduino-ide>

- <https://hocarm.org/esp8266-va-firebase/>

- <https://firebase.google.com/docs/database/web/read-and-write>

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Nội dung công việc | SV thực hiện | Ghi chú |
| 1 | * Thảo luận ý tưởng | Cả nhóm |  |
| 2 | * Thiết kế đề xuất đồ án | Cả nhóm |  |
| 3 | * Tìm hiểu các loại module cần thiết và giá thành hợp lí:   + Arduino Mega, cảm biến vân tay  + ESP 8266  + LCD, loa, bàn phím, khóa điện | Thúy  Duyên  Hằng |  |
| 4 | - Mua linh kiện | Cả nhóm |  |
| **5** | * Lắp ráp các phần của hệ thống | Cả nhóm |  |
| **6** | -Lập trình các chức năng của hệ thống | Thúy | Mở cửa, thêm vân tay, xóa vân tay |
| -Chuyển dữ liệu từ Arduino qua ESP8266 |
| **7** | -Gởi dữ liệu từ ESP8266 lên cơ sở dữ liệu | Duyên | Sử dụng cơ sở dữ liệu Firebase |
| -Xây dựng cơ sở dữ liệu |
| **8** | * Thiết kế website | Hằng | Các chức năng xem lịch sử, xem nhân viên, chấm công, thêm thông tin nhân viên |
| * Đưa web lên hosting |
| **9** | * Kiểm tra các chức năng và chỉnh sửa lỗi | Cả nhóm |  |
| **10** | * Hoàn thiện hệ thống | Cả nhóm |  |
| **11** | * Làm báo cáo đồ án | Cả nhóm |  |