**096407TÓM TẮT ĐỒ ÁN**

*Trong khuôn khổ của đồ án này, với ý tưởng nhằm đơn giản hóa nhưng đồng thời lại tăng tính bảo mật, thông minh thay vì dùng những chiếc khóa thông thường với hình ảnh ổ khóa và chìa khóa đơn giản chúng em đã có ý tưởng làm ra “****chìa khóa thông minh”****. Chìa khóa thông minh là sản phẩm sử dụng kết hợp cảm biến vân tay và mật khẩu được nhập từ keypad, thông qua thuật toán xử lý dữ liệu người dùng được lưu trữ trên hệ thống và webserver từ đó cho phép người dùng mở cánh cửa của nhà mình, công sở, văn phòng hay những nơi cần bảo mật cao khác. Hơn nữa, dữ liệu sau đó cũng được lưu trữ và dễ dàng trích xuất. Tuy thu được các kết quả mong đợi nhưng vẫn còn một số hạn chế nhất định, sản phẩm này nếu được phát triển hơn nữa sẽ giúp cho việc bảo mật của các căn hộ, tòa nhà, công ty, công sở, … được nâng cao hơn rất nhiều lần.*

***Lời cảm ơn****: Trong quá trình thực hiện đề tài này, chúng em xin gửi lời cảm ơn chân thành đến thầy* ***Phạm Công Thắng*** *đã nhiệt tình giúp đỡ chúng em hoàn thành đề tài này. Mặc dù đã cố gắng nghiên cứu nhưng do kinh nghiệm, thời gian và kiến thức còn nhiều hạn chế nên không thể tránh khỏi những sai sót nhất định trong quá trình làm đồ án này. Rất mong nhận được sự góp ý của quý thầy cô để chúng em ngày càng hoàn thiện hơn.*

*Chúng em xin chân thành cảm ơn.*

MỤC LỤC

[**I.** **Giới thiệu đề tài** 3](#_Toc44084945)

[**1.** **Giới thiệu hiện trạng** 3](#_Toc44084946)

[**2.** **Yêu cầu đặt ra** 3](#_Toc44084947)

[**3.** **Giải pháp đề xuất** 3](#_Toc44084948)

[**4.** **Bảng phân công nhiệm vụ** 4](#_Toc44084949)

[**II.** **Giải pháp** 4](#_Toc44084950)

[**1.** **Chức năng chính hệ thống** 4](#_Toc44084951)

[**2.** **Cơ sở lý thuyết nhận dạng vân tay** 5](#_Toc44084952)

[a. Giới thiệu 5](#_Toc44084953)

[b. Nâng cao chất lượng ảnh vân tay 7](#_Toc44084954)

[c. Nhị phân hóa ảnh vân tay 7](#_Toc44084955)

[d. Làm mỏng vân tay 8](#_Toc44084956)

[e. Trích xuất các điểm Minutiae 9](#_Toc44084957)

[f. Loại bỏ các điểm Minutiae giả 9](#_Toc44084958)

[g. So sánh vân tay 9](#_Toc44084959)

[**3.** **Sơ đồ khối hệ thống** 10](#_Toc44084960)

[**4.** **Mô tả phần mềm** 13](#_Toc44084961)

[**5.** **Mô tả phần cứng** 13](#_Toc44084962)

[a. Mô tả chi tiết các linh kiện 13](#_Toc44084963)

[b. Bảng thống kê chi phí cho toàn bộ sản phẩm 18](#_Toc44084964)

[**III.** **Kết quả đạt được** 18](#_Toc44084965)

[**IV.** **Kết luận** 23](#_Toc44084966)

[**V.** **Tài liệu tham khảo** 24](#_Toc44084967)

1. **Giới thiệu đề tài**
2. **Giới thiệu hiện trạng**

Trên thế giới đã có rất nhiều hệ thống, thiết bị áp dụng theo mô hình “chìa khóa thông minh” dựa vào nhân trắc học như mống mắt, vân tay, hình dạng khuôn mặt, ... là các đặc điểm riêng biệt của mỗi con người cho nên tính bảo mật mang lại rất cao. Tại Việt Nam, để bắt kịp thế giới, những năm gần đây đã có các công ty, tập đoàn đã đưa mô hình này vào các thiết bị công nghệ như smartphone, khóa cửa, …

Nhận thấy đây là vấn đề còn nhiều tiềm năng trong tương lai sắp tới cũng như nhu cầu bảo mật trong cuộc sống ngày một hiện đại là điều tất yếu, nhằm đảm bảo tính bảo mật đó lại có tính thông minh, áp dụng công nghệ mới, chúng em đã nghiên cứu mô hình khóa cửa áp dụng công nghệ nhận diện vân tay và mật khẩu kết hợp thông qua hệ thống website xử lý dữ liệu tạo ra một khả năng bảo mật và thông minh hơn, tiện lợi hơn so với các loại khóa cửa truyền thống. Từ đó góp phần vào việc thúc đẩy nâng cao chất lượng cuộc sống, ổn định cho xã hội.

1. **Yêu cầu đặt ra**

* Cần có một hệ thống phải hoặc động chính xác tuyệt đối và ổn định, tính bảo mật hàng đầu.
* Đảm bảo hệ thống có thể hoặc động ở một chế độ hoặc tích hợp đồng thời nhiều chế độ làm việc khác nhau (thường là hai) cho người dùng dễ dàng lựa chọn sử dụng.
* Đảm bảo quản lý được thông tin người ra vào thông qua website xử lý dữ liệu được kết nối và trích xuất ra file khi cần thiết.
* Tiết kiệm chi phí xây dựng hệ thống.

1. **Giải pháp đề xuất**

* Sản phẩm chúng em hướng đến có hai chế độ: Một là dùng mật khẩu, hai là dùng vân tay đã được nạp sẵn trên hệ thống, người cần mở cửa phải chọn một trong hai chế độ để có thể mở cửa và đi vào bên trong.
* Người quản trị hệ thống có thể khởi tạo vân tay để cấp quyền ra vào cửa hoặc xóa vân tay để bãi bỏ quyền ra vào cửa của một người.
* Hệ thống ghi lại lịch sử ra vào của mỗi người cũng như là những người cố gắng mở cửa bằng vân tay nhưng không được. Hệ thống cho phép tìm kiếm thông tin cụ thể liên quan đến mỗi người dùng, và trích xuất lịch sử mở cửa ra vào thành file Excel khi cần thiết.

1. **Bảng phân công nhiệm vụ**

Nhiệm vụ của các thành viên thực hiện được phân công như sau (Bảng 1):

|  |  |  |
| --- | --- | --- |
| STT | Bảng phân công nhiệm vụ | |
| 1 | Trần Văn Sĩ (nhóm trưởng) | Theo dõi và quản lý tiến độ. Hỗ trợ lấy ảnh vân tay, nghiên cứu và thực hiện xây dựng website quản lý cho admin và xử lý ảnh trên server |
| 2 | Phan Văn Nhân | Nghiên cứu xử lý ảnh và nhận biết vân tay |
| 3 | Nguyễn Thiện Phúc | Thiết kế, thực hiện mô hình đồ án và mở khóa bằng mật khẩu |
| 4 | Nguyễn Văn Huy Tưởng | Hỗ trợ ý tưởng và thực hiện, thiết kế mô hình, tổng kết ý kiến và chỉnh sửa. Viết báo cáo đồ án |

Bảng 1: Bảng phân công nhiệm vụ của các thành viên

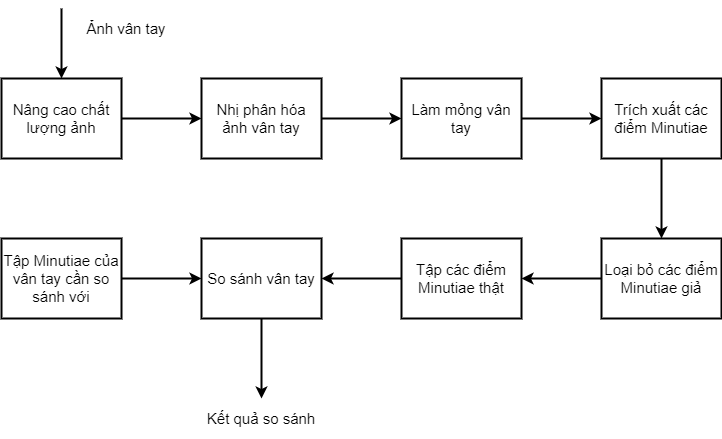
1. **Giải pháp**
2. **Chức năng chính hệ thống**

Khi người dùng đến trước cửa, hệ thống cung cấp hai chế độ để mở cửa, một là dùng mật khẩu, hai là dùng dấu vân tay. Người dùng chọn một trong hai chế độ để thực hiện việc mở cửa. Khi người dùng chọn chế độ dùng mật khẩu, người đó phải thông qua keypad nhập mật khẩu vào, hệ thống tiếp nhận mật khẩu do người dùng nhập vào và đối chiếu với mật khẩu mặc định cài đặt sẵn từ trước, nếu đúng thì cửa được mở, sai thì không. Nếu chọn chế độ dùng dấu vân tay, người dùng ấn vân tay của mình đã được cấp quyền truy nhập từ trước, hệ thống tiếp nhận ảnh dấu vân tay của người đó đối chiếu với dữ liệu các dấu vân tay được cấp quyền truy nhập cửa có ở trên server, nếu trùng khớp thì cửa mở, sai thì không.

Sau mỗi lần mở khóa bằng vân tay, thông tin được lưu trữ trên server, tính toán số lần mở cửa, người đã tiến hành mở cửa để trích xuất thông tin lúc cần thiết.

1. **Cơ sở lý thuyết nhận dạng vân tay**

Hiện nay nhận dạng vân tay đã và đang được sử dụng rộng rãi trong nhiều lĩnh vực của xã hội. Rất nhiều thuật toán đã được nghiên cứu và triển khai, trong đó thuật toán được nêu dưới đây (Hình 1) là thuật toán được sử dụng phổ biến nhất và cũng là thuật toán được sử dụng trong đồ án này.



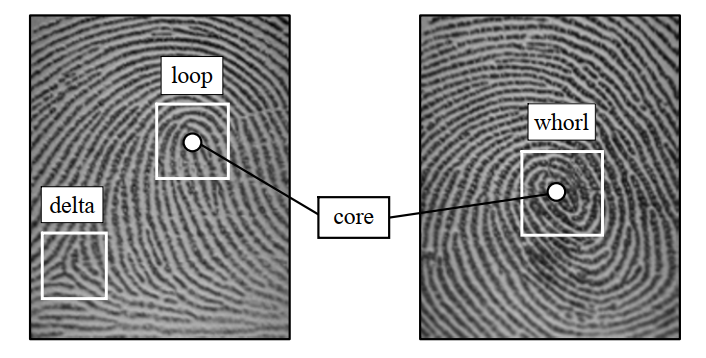
Hình 1: Sơ đồ thuật toán nhận dạng vân tay

1. **Giới thiệu**

Cơ sở của nhận dạng vân tay là những điểm đặc trưng trong cấu tạo của vân tay và vân tay của mỗi người là duy nhất.

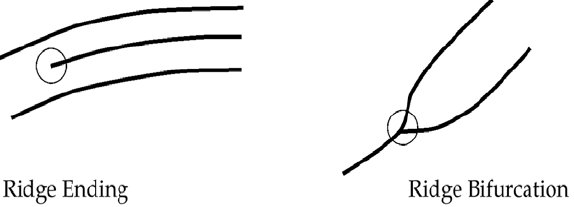
Vân tay là cấu trúc gồm các vân lồi (ridge) và vân lõm (valley) xen kẽ lẫn nhau. Trên các ảnh vân tay tồn tại các điểm đặc trưng và được phân thành hai loại:

* Singularity: là vùng trên vân tay có cấu trúc khác thường so với những vùng khác. Singularity được chia thành ba loại hình: Loop, Delta và Whorl (Hình 2).



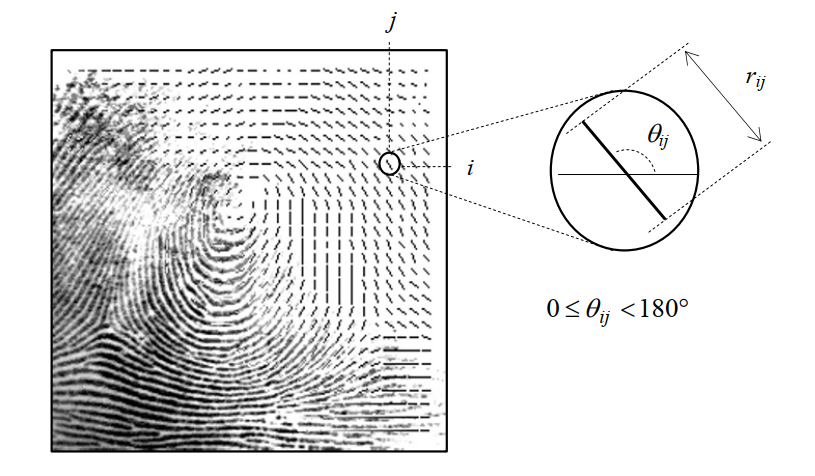
Hình 2: Các loại điểm Singularity

* Minutiae: là những điểm trên các vân lồi mà tại đó cấu trúc của vân lồi bị thay đổi. Hai loại điểm Minutiae quan trọng, thường được sử dụng cho nhận dạng vân tay đó là: Ridge Ending (điểm mà đường vân kết thúc) và Bifurcation (điểm rẽ nhánh) (Hình 3).



Hình 3: Hai loại điểm Minutiae

Trường định hướng: Ảnh vân tay là ảnh định hướng, các đường vân là các đường công theo các hướng xác định. Góc định hướng θxy hợp bởi phương của một điểm (x, y) trên đường vân với phương ngang được gọi là hướng của điểm đó, nằm trong đoạn [0o, 180o] [1]. Tập hợp các hướng của các điểm trên ảnh vân tay được gọi là trường định hướng của ảnh vân tay đó (Hình 4).



Hình 4: Ví dụ về hình ảnh của trường định hướng

1. **Nâng cao chất lượng ảnh vân tay**

Có rất nhiều phương pháp để nâng cao chất lượng ảnh vân tay, trong đó phương pháp sử dụng bộ lọc Gabor, được Hong, Wan và Jain (1998) đưa ra, đã đạt hiệu quả rất tốt [2, 3].

Bộ lọc Gabor có dạng:

,

Trong đó: + θ là hướng của bộ lọc.

+ f là tần số của sóng phẳng hình sin.

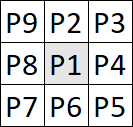
+ là các độ lệch chuẩn.

1. **Nhị phân hóa ảnh vân tay**

Sử dụng phương pháp ngưỡng toàn cục. Được thực hiện bằng cách chọn một ngưỡng toàn cục t, thiết lập các điểm ảnh có độ xám nhỏ hơn t về 0 và các điểm ảnh còn lại về 1.

1. **Làm mỏng vân tay**

Sử dụng thuật toán Zhang – Suen. Thuật toán thực thi trên điểm ảnh có giá trị bằng 1 cùng với 8 điểm ảnh lân cận được đánh số (Hình 5).



Hình 5: Đánh số các điểm lân cận

Định nghĩa hai giá trị được sử dụng trong thuật toán:

A(P1) = Số lần giá trị chuyển từ 0 sang 1 trong chuỗi P2 🡪 P3 🡪 P4 🡪 P5 🡪P6 🡪 P7 🡪 P8 🡪 P9 🡪 P2.

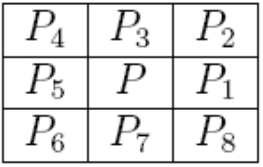
B(P1) = số điểm ảnh lân cận P1 có giá trị bằng 1.

Thuật toán:

* Bước 1: Thuật toán sẽ duyệt qua tất cả các điểm ảnh có giá trị bằng 1, nếu thỏa mãn các điều kiện sau thì điểm ảnh đấy sẽ được thiết lập về 0:
  + 2 <= B(P1) <= 6.
  + A(P1) = 1.
  + Ít nhất một trong các điểm ảnh P2, P4, P6 có giá trị là 0.
  + Ít nhất một trong các điểm ảnh P4, P6, P8 có giá trị là 0.
* Bước 2: Thuật toán sẽ duyệt qua tất cả các điểm ảnh có giá trị bằng 1, nếu thỏa mãn các điều kiện sau thì điểm ảnh đấy sẽ được thiết lập về 0:
  + 2 <= B(P1) <= 6.
  + A(P1) = 1.
  + Ít nhất một trong các điểm ảnh P2, P4, P8 có giá trị là 0.
  + Ít nhất một trong các điểm ảnh P2, P6, P8 có giá trị là 0.
* Bước 3: Lặp lại 2 bước trên cho đến khi không có điểm ảnh nào bị thay đổi.

1. **Trích xuất các điểm Minutiae**

Sử dụng phương pháp Crossing Number (CN) đối với các điểm pixel có giá trị bằng 1 cùng với 8 điểm ảnh lân cận, được đánh số (Hình 6).



Hình 6: Đánh số các điểm lân cận

Giá trị CN được tính bởi công thức: CN = 0.5

Với giá trị CN tương ứng ta có thể xác định các điểm Minutiae (Ridge ending và Bifurcation) (Bảng 2) [4]:

|  |  |
| --- | --- |
| CN | Loại |
| 0  1  2  3  4 | Điểm cô lập (Isolated point)  Điểm kết thúc (Ridge ending point)  Điểm liên tục (Continuing ridge point)  Điểm rẽ nhánh (Bifurcation point)  Điểm giao nhau (Crossing point) |

Bảng 2: Phân loại điểm Minutiae dựa vào Crossing Number

1. **Loại bỏ các điểm Minutiae giả**

* Loại bỏ các điểm gần với đường bao của ảnh vân tay (trong vòng 15 pixels).
* Loại bỏ các điểm Minutiae quá gần nhau (khoảng cách giữa 2 điểm Minutiae < D).

Với D là khoảng cách trung bình giữa các vân lồi (Ridge) [5].

1. **So sánh vân tay**

Gọi I và T lần lượt là tập Minutiae của ảnh vân tay đầu vào và ảnh vân tay cần so sánh.

**T** = {m1, m2, …, mm} với mi = {xi, yi, θi, ti}, i = 1…m

**I** = {m’1, m’2, …, m’n} với m’j = {x’j, y’j, θ’j, t’j}, j = 1…n

m và n là số điểm Minutiae của **T** và **I**.

Với (xi, yi) là tọa độ của điểm Minutiae, θi là hướng cục bộ và ti chỉ loại (1: Ridge ending, 3: Bifurcation).

Trước khi so sánh, cần tiến hành căn chỉnh ảnh vân tay để tăng độ chính xác của quá trình so sánh ảnh vân tay. Sử dụng điểm core trên vân tay làm tâm để căn chỉnh hai ảnh vân tay.

Theo định nghĩa của Henry (1900) điểm *core* là điểm nằm ở phía Bắc nhất của vân lồi (Ridge) trong cùng [1].

Sử dụng phương pháp xác định điểm *core* của Bahgat, Khalil, Kader và Mashali đưa ra [6], cũng như phương pháp của Cao, Sun, Mao và Mei trong trường hợp đặc biệt [7].

Một điểm m’j trong **I** và một điểm mi trong **T** được xem là khớp với nhau nếu hai điểm đó là cùng loại (cùng là Ridge ending hoặc cùng là Bifurcation), khoảng cách (*sd*) giữa hai điểm nhỏ hơn ngưỡng r0 và sự khác nhau về hướng (*dd*) nhỏ hơn ngưỡng θ0 [1].

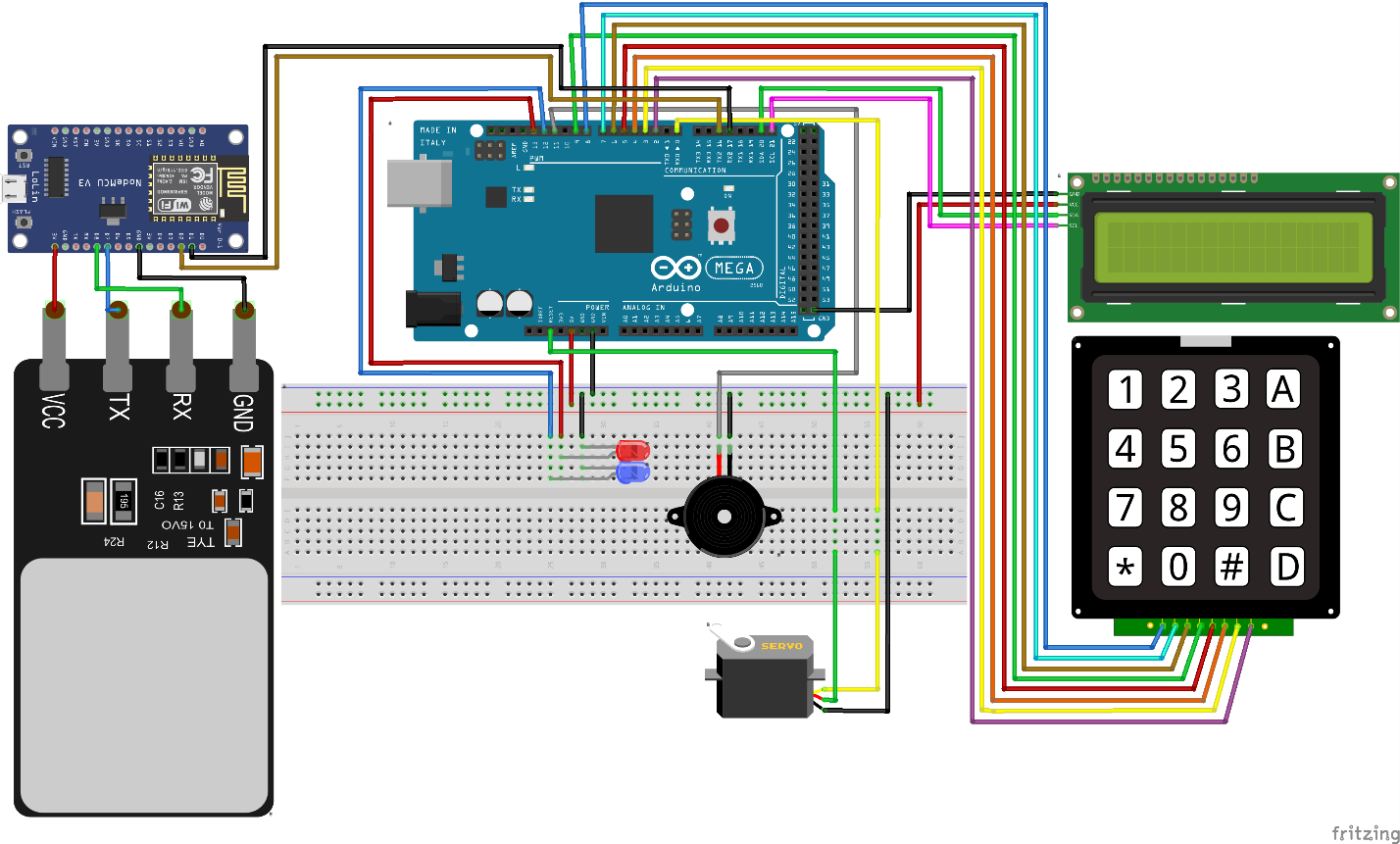
*sd*(m’j, mi) = , và

*dd*(m’j, mi) = *min*(||, )

Gọi k là số cặp điểm khớp với nhau. Tỉ lệ giống nhau được tính bởi công thức:

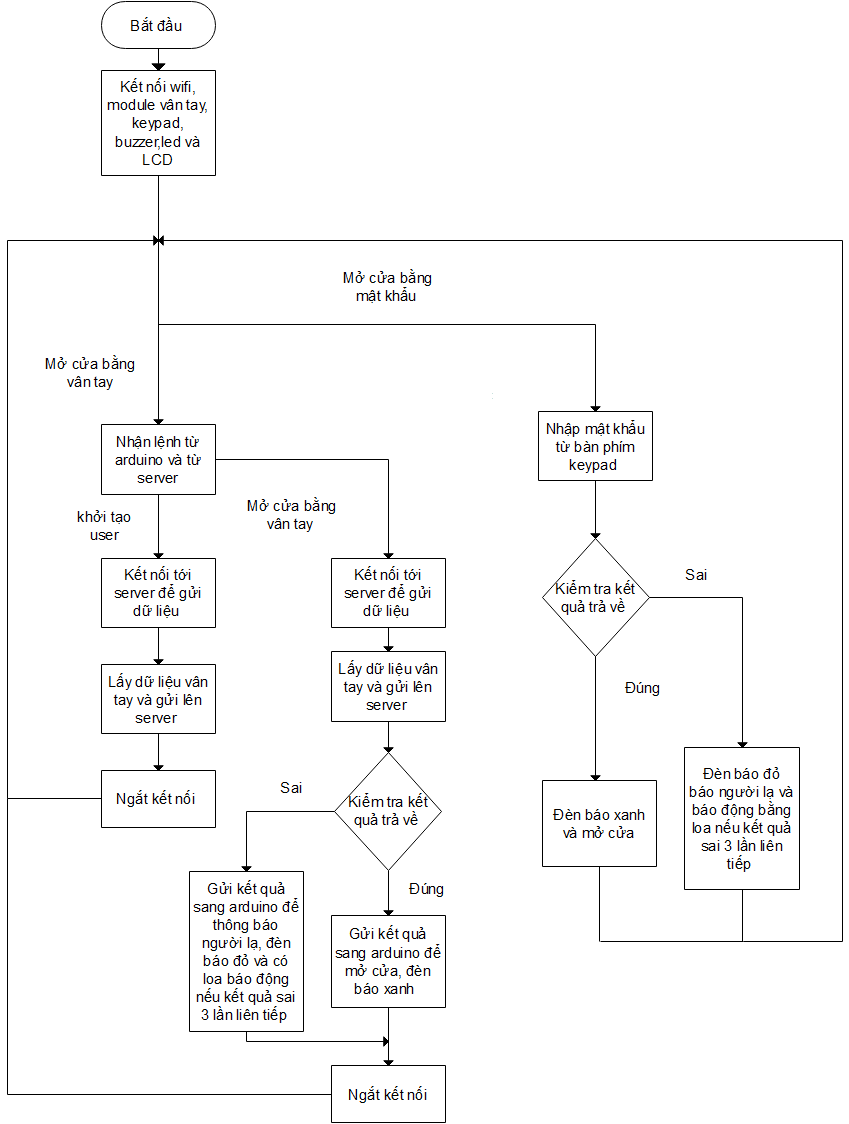
1. **Sơ đồ khối hệ thống**

* Sơ đồ kết nối các linh kiện: Các linh kiện được mắc mạch và kết nối như sau (Hình 7).



Hình 7: Sơ đồ kết nối các linh kiện phần cứng

* Sơ đồ khối giải thuật: Thể hiện cách thức hoạt động của hệ thống (Hình 8).



Hình 8: Sơ đồ khối giải thuật

1. **Mô tả phần mềm**

* Sử dụng web server lập trình bằng ngôn ngữ PHP để tương tác với database, thực hiện việc quản lý các thông tin từ xa như lấy dữ liệu, xóa, cập nhật, so sánh, lưu dữ liệu bao gồm các vân tay người dùng, thống kê thông tin truy nhập yêu cầu mở cửa bằng vân tay.
* ESP8266 Node MCU sẽ đảm nhận việc gửi dữ liệu vân tay của người dùng lên server để xử lý khi nhận được yêu cầu là khởi tạo người dùng hoặc nhận được yêu cầu mở cửa bằng vân tay từ phía Arduino, nhận kết quả trả về từ server để yêu cầu Arduino mở cửa hay là không.
* Sử dụng Arduino IDE (là chính), PhpStorm, Spider để hỗ trợ lập trình tạo nên chương trình sản phẩm.

1. **Mô tả phần cứng**
2. **Mô tả chi tiết các linh kiện**

* Esp8266 Node MCU [8]:
  + Là thiết bị được sử dụng trong các dự án IoT và điều khiển thiết bị khác. Mạch nhỏ gọn, sử dụng giao tiếp UART, được tích hợp wifi 2,4 GHz, giao thức TCP/IP có thể dùng cho lập trình.
  + Cấu tạo gồm 30 chân, trong đó có các chân GPIO vừa có thể làm chân analog vừa có thể làm chân digital, có 1 cổng USB nạp mã code.
  + Tham số kĩ thuật:

- 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

- Bộ nhớ Flash: 4MB

- Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)

* + Muốn việc truyền nhận thông tin được diễn ra, người dùng cần phải kết nối Esp với mạng wifi bằng cách thiết lập cho Esp biết tên và mật khẩu mạng wifi, sau đó mới có thể tiến hành các thiết lập lệnh tiếp theo.
  + Code kết nối

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

AS608 Fingerprint Senso [9]:

* + Là thiết bị sử dụng giao tiếp UART TLL hoặc USB để kết nối với vi điều khiển hoặc kết nối trực tiếp với máy tính (thông qua mạch chuyển USB-UART hoặc giao tiếp USB)
  + Cấu tạo gồm có 8 chân, tuy nhiên chỉ cần 4 chân để kết nối với module Arduino thông qua giao tiếp UART hoặc USB.
  + Tham số kĩ thuật:

- Điện áp nguồn: 3.6 – 6V DC

- Dòng điện hoạt động: 120mA – cực đại: 150mA

- Thời gian chụp ảnh vân tay: dưới 0.1s

- Dung lượng: 256 byte/mẫu, mẫu sẵn: 512 byte, lưu trữ tối đa 162 mẫu vân tay

- Tốc độ truyền (baud): 9600, 19200, 28800, 38400, 57600 (mặc định).

* Code kết nối

finger.begin(57600);

if (finger.verifyPassword())

Serial.println("Found fingerprint sensor!");

else

Serial.println("Did not find fingerprint sensor");

* LCD 16x2 I2C [10]:
  + Dùng để hiển thị các thông tin do người dùng thiết lập. Màn hình text LCD xanh có khả năng hiển thị 2 dòng với mỗi dòng 16 kí tự.
  + Cấu tạo LCD gồm 16 chân, tuy nhiên thay vì kết nối nhiều chân với LCD thì ta có thể kết nối thông qua module I2C chỉ dùng 4 chân để việc kết nối được gọn hơn. Module I2C hỗ trợ các loại LCD sử dụng driver HD44780 (LCD 16x2, 20x4, …) và tương thích với hầu hết các vi điều khiển hiện nay.
  + Tham số kĩ thuật:

- Điện áp hoạt động: 2.5 - 6V DC

- Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).

* Code kết nối: Thêm thư viện điều khiển I2C

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x3F, 16, 2);

Void Setup() {

lcd.init();

lcd.backlight();

lcd.setCursor(5, 0);

lcd.print("Welcome");

}

* Servo [11]:
  + Là loại động cơ kích thước nhỏ, được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng. Động cơ có tốc độ phản ứng nhanh, có thể dễ dàng điểu khiển góc quay bằng phương pháp điều độ rộng xung PWM.
  + Servo chỉ cần 3 chân để có thể giao tiếp với module Arduino
  + Tham số kĩ thuật:

- Bị giới hạn trong phạm vi 180, 270 hoặc 90 độ.

- Tín hiệu PMW có chu kỳ 20 ms, tần số 50Hz

- Điện áp nguồn: 5V

* + Code kết nối:

Thêm thư viện để có thể giao tiếp với Servo

#include <Servo.h>

Servo servo;

int Position = 0;

Hai hàm dùng để đóng/mở cửa:

void ServoOpen() {

for (Position = 0; Position <= 120; Position += 5) {

servo.write(Position);

delay(15);

}

}

void ServoClose() {

for (Position = 120; Position >= 0; Position -= 5) {

servo.write(Position);

delay(15);

}

}

* Matrix keypad 4x4:
  + Là thiết bị có các phím bấm từ 0-9, từ A-D, # và \*, sắp xếp theo 4 hàng và 4 cột. Cho phép người dùng có thể thao tác bấm trực tiếp lên các phím bấm.
  + Cấu tạo gồm có 8 chân giao tiếp với Arduino module.
  + Tham số kĩ thuật [12]:

- Độ dài cáp: 88mm

- Nhiệt độ hoạt động: 0 ~ 70°C

- Kích thước bàn phím 77 x 69 mm

* + Code kết nối: Thêm thư viện để có thể giao tiếp với Keypad

#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

byte rowPins[ROWS] = {8, 7, 6, 9};

byte colPins[COLS] = {5, 4, 3, 2};

Keypad myKeypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

* Module Arduino Mega 2560 [13]:
* Là một vi điều khiển hoạt động dựa trên chip Atmega2560, cho phép nạp mã lập trình, kết nối, điều khiển các phần cứng mô hình.
* Cấu tạo gồm 54 chân digital, trong đó 15 chân cố thể sử dụng như những chân PMW là từ chân số 2 đến 13 và chân 44, 45, 46, 6 chân ngắt ngoài, 16 chân vào analog (A0 – A15), 4 cổng Serial giao tiếp phần cứng.
* Tham số kĩ thuật:
  + - Điện áp hoạt động: 5V, điện áp vào đề nghị: 7 – 15V giới hạn 6 - 20V.
    - Cường độ dòng điện trên mỗi 3.3V pin là 50mA, trên mỗi I/O pin là 20mA
    - Flash Memory: 256 KB, SRAM: 8KB, EEPROM 4KB
    - Clock Speed: 16 MHz
* Buzzer
* Còn gọi là loa mini hay còi báo, là thiết bị tạo ra tiếng còi hoặc tiếng bíp, đòi hỏi phải có các bộ điều khiển dao động (hoặc vi điều khiển) để điều khiển nó.
* Cấu tạo gồm 2 chân bao gồm một chân GND và một chân tín hiệu
* Tham số kĩ thuật:
* hoạt động ở điện áp 3 – 5V DC, khoảng cách chân 7,6mm
* Tần số 2300 Hz
* Nhiệt độ hoạt động -20 đến 65°C
* Code kết nối:

Khai báo chân tín hiệu:

int buzzer = 11;

Thiết lập bật/tắt tiếng kêu của Buzzer

tone(buzzer,450);

noTone(buzzer);

1. **Bảng thống kê chi phí cho toàn bộ sản phẩm**

Tất cả các chi phí phần cứng được tổng hợp thông qua Bảng 3 như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên linh kiện | Số lượng | Giá thành (VNĐ) |
| 1 | Arduino Mega 2560 | 1 | 185.000 |
| 2 | Esp8268 Node MCU | 1 | 85.000 |
| 3 | As608 fingerprint Sensor | 1 | 300.000 |
| 4 | Servo | 1 | 25.000 |
| 5 | Màn hình LCD + I2C | 1 | 40.000 |
| 6 | Matrix keypad 4x4 | 1 | 16.000 |
| 7 | Breadboard | 1 | 20.000 |
| 8 | LED | 2 | 2.000 |
| 9 | Buzzer | 1 | 3.000 |
| 10 | Dây dẫn | 40 | 20.000 |
| 11 | Hộp nhựa | 1 | 25.000 |
| Tổng cộng: | | | 721.000 |

Bảng 3: Bảng thống kê chi phí cho toàn bộ các linh kiện sản phẩm

1. **Kết quả đạt được**

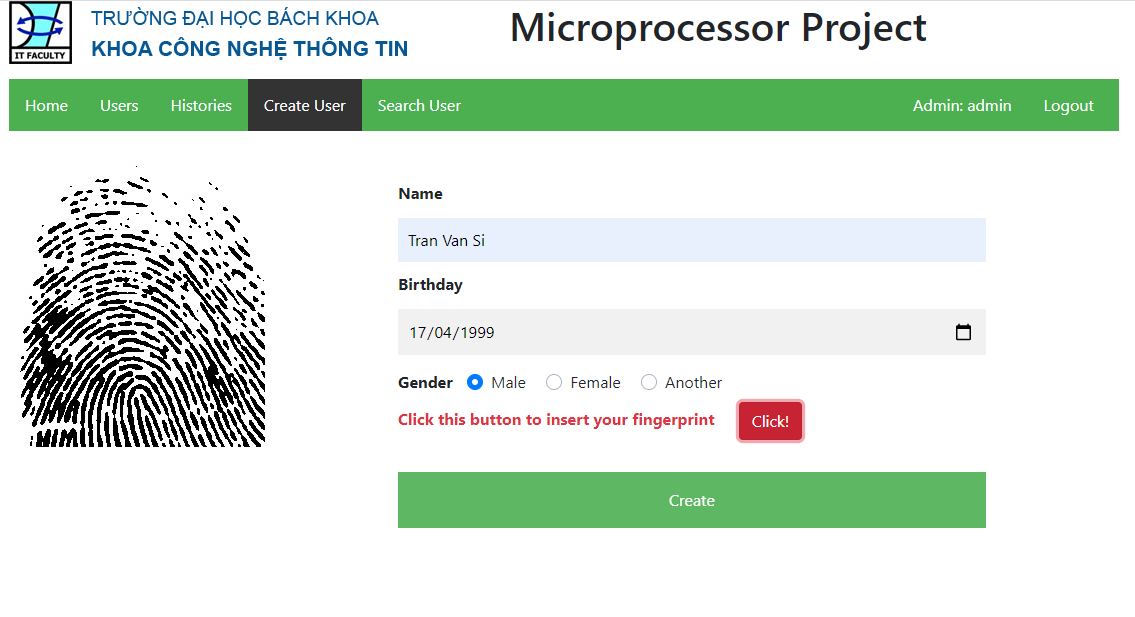
Quá trình lấy mẫu vân tay được áp dụng cho cả 4 thành viên trong nhóm, chủ yếu là ngón trỏ và ngón cái trên bàn tay. Nhìn chung chất lượng vân tay trích xuất được là rất rõ ràng đáp ứng được các giải pháp đưa ra, sau đây là một số kết quả làm việc của hệ thống nói chung:

* Giao diện chính của webserver (Hình 9):



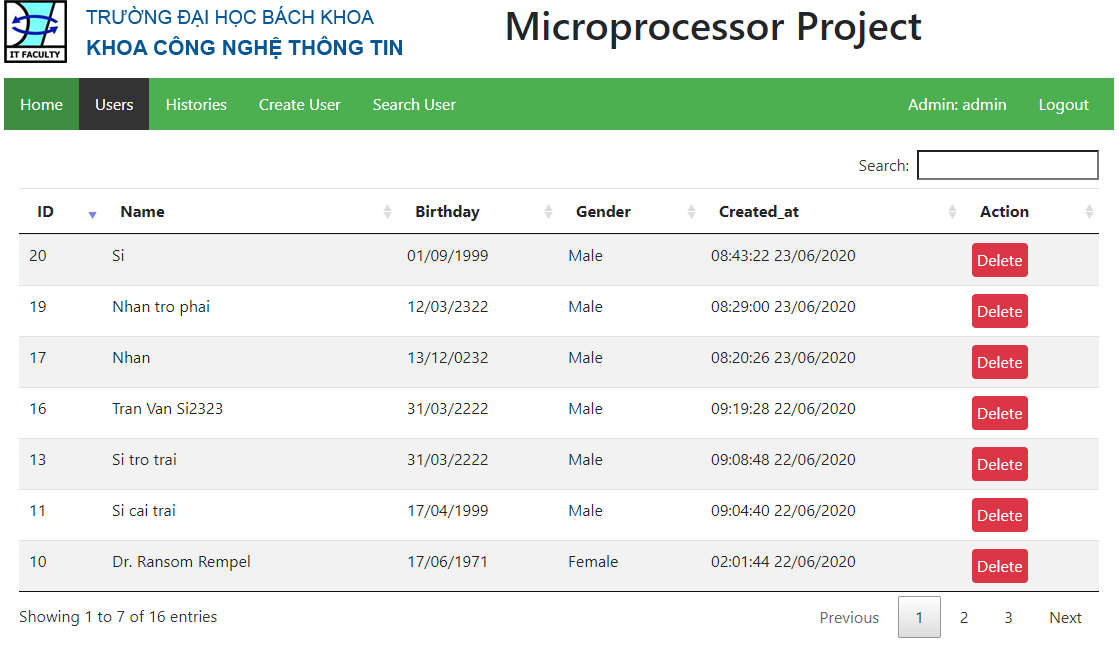
Hình 9: Trang chủ của webserver

* Sau khi người dùng được khởi tạo vân tay hệ thống sẽ lưu lại trên server (Hình 10).



Hình 10: Khởi tạo vân tay cho người dùng

* Thông tin người dùng sau khi khởi tạo vân tay xong sẽ được lưu trong danh sách người dùng (Hình 11).



Hình 11: Danh sách các người dùng đã được khởi tạo vân tay

* Hệ thống dùng cả hai chế độ làm việc, người dùng có thể lựa chọn để ra vào cửa. Nếu mở cửa 3 lần không thành công, hệ thống sẽ bật loa báo động kêu. Trạng thái bắt đầu của hệ thống (Hình 12), cửa mở thành công (Hình 13), hệ thống báo sai khi mở không thành công (Hình 14) được mô tả như sau:



Hình 12: Trạng thái ban đầu

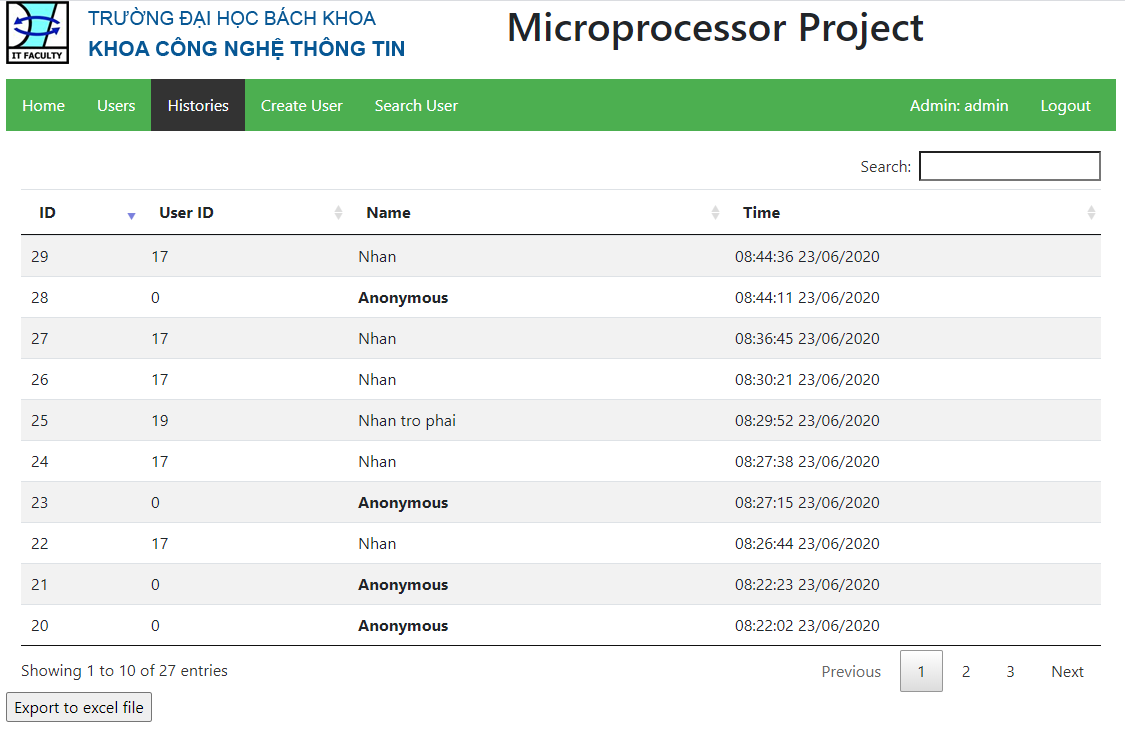


Hình 13: Cửa được mở khi người dùng nhập mật khẩu hay nhận diện vân tay thành công



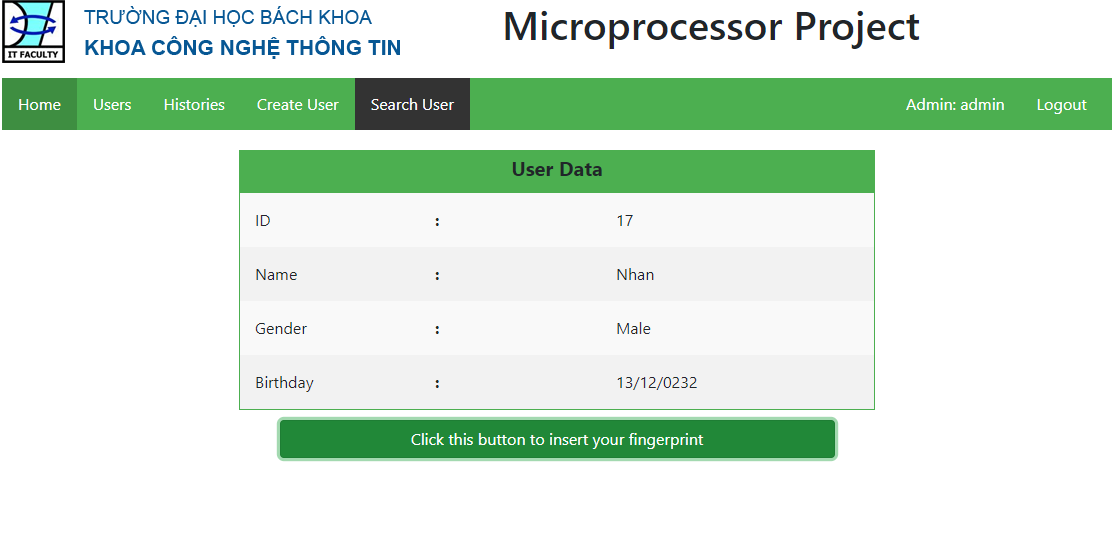
Hình 14: Hệ thống báo sai, khi mở cửa không thành công

* Lịch sử các lần mở khóa bằng nhận diện vân tay của các người dùng được lưu lại trên server (Hình 15) và có thể trích xuất thành file Excel khi cần thiết.



Hình 15: Danh sách lịch sử các người dùng đã mở cửa bao gồm cả người lạ, hoặc những lần chưa nhận diện được

* Hệ thống còn cho phép tìm kiếm thông tin của một người dùng đã lưu lại từ trước đó trên hệ thống (Hình 16).



Hình 16: Tìm kiếm thông tin của một người dùng

Sau khi thực hiện và thu được các kết quả như trên, chúng em có một vài nhận xét thu được như sau:

* Kích thước của một ảnh vân tay là 256x286, giải giá trị từ 0-255, là ảnh xám.
* Tính năng mở khóa bằng nhận diện vân tay: Ảnh được lấy trực tiếp từ 4 thành viên, mỗi người hai ngón là ngón cái và ngón trỏ tạo thành 8 trường hợp. Ứng với mỗi trường hợp lấy 10 mẫu, chọn 1 mẫu làm cơ sở, 9 mẫu còn lại dùng để so sánh sự so khớp nên ta có số lần so khớp là 9\*8 = 72 lần:
* Số lần thử mở khóa: 72.
* Số lần nhận đúng vân tay: 63.
* Số lần không nhận được vân tay: 8.
* Số lần nhận sai vân tay: 1.
* Tỉ lệ nhận đúng: 87.5%.
* Tỉ lệ nhận sai: 1.39%.
* Tỉ lệ không nhận được vân tay: 11.11%
* Chúng em đã kiểm thử trong điều kiện ngoài trời và trong nhà, nhận thấy điều kiện hoạt động lý tưởng của hệ thống là từ 15-45°C, với điều kiện hoạt động tốt hơn ở trong không gian của các công trình so với ở ngoài trời, ánh sáng vừa phải không quá gắt, không nên để trực tiếp với ánh sáng mặt trời vì có thể làm sai lệch dấu vân tay do chênh lệnh ánh sáng trong và ngoài thiết bị cảm biến vân tay.
* Hệ thống hoạt động khá chính xác và ổn định, thời gian cho hệ thống mở cửa bằng mật khẩu là khoảng nhỏ hơn 1 giây, bằng vân tay là khoảng 10 giây. Hệ thống vẫn cần khắc phục hoàn thiện hơn ở mặt thời gian xử lý và độ chính xác của cảm biến trong việc nhận và so sánh ảnh vân tay.

1. **Kết luận**

Hệ thống “Chìa khóa thông minh” là một giải pháp mang tính tương lai, được ứng dụng rộng rãi trong đời sống vì tình ứng dụng cao của nó.

Hệ thống đáp ứng và giải quyết được các yêu cầu và giải pháp mà nhóm đã xây dựng, thu được các kết quả như mong muốn. Kết quả mang lại khá chính xác và ổn định, tuy nhiên nếu muốn phát triển hơn nữa thì hệ thống này cần cải tiến nhiều hơn như giảm thời gian xử lý, tăng tính chính xác hơn nữa trong việc nhận và so sánh các mẫu vân tay, …. Nếu có thêm nhiều thời gian và kinh phí xây dựng chúng em sẽ có thể hoàn thiện hơn những hạn chế mà hệ thống này đang mắc phải, tạo ra một hệ thống hoàn thiện nhất, góp một phần nhỏ cho đời sống xã hội ổn định và phát triển hơn.

1. **Tài liệu tham khảo**

[1]. Maltoni, Maio, K. Jain and Prabhakar, “Handbook of Fingerprint Recognition”, Second Edition.

[2]. Lin Hong, Yifei Wan and Anil Jain, “Fingerprint Image Enhancement: Algorithm and Performance Evaluation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 8 (1998).

[3]. Utkarsh-Deshmukh, Fingerprint Enhancement Python, URL: <https://github.com/Utkarsh-Deshmukh/Fingerprint-Enhancement-Python>. [Date accessed: 01/05/2020]

[4]. R. Bansal, P. Sehgal and P. Bedi, “Minutiae Extraction from Fingerprint Image – a Review”. IJCSI, Vol.8, Issue 5, No 3, September 2011.

[5]. M. James Stephen, Prasad Reddy P.V.G.D, Prof. K.R.Sudha, “Removal of False Minutiae with Modified Fuzzy Rules”.

[6]. G.A. Bahgat, A.H. Khalil, N.S. Abdel Kader, S. Mashali, “Fast and accurate algorithm for core point detection in fingerprint images”.

[7]. Guo Cao, Quan-Sen Sun, Zhihong Mao and Yuan Mei, “Detection of Core Points in Fingerprint Images based on Edge Map”, 2009 International Conference on Electronic Computer Technology.

[8]. URL: <https://lophocvui.com/iot-internet-of-things/smart-home/openhab/tong-quan-ve-esp8266/>. [Ngày truy cập: 20/05/2020]

[9]. Lady Ada, “Adafruit Optical Fingerprint Sensor”, Adafruit Industries, Oct 28th 2019.

[10]. URL: <https://arduinokit.vn/giao-tiep-i2c-lcd-arduino>. [Ngày truy cập: 20/05/2020]

[11]. URL: <https://arduinokit.vn/dong-co-servo-la-gi>. [Ngày truy cập: 20/05/2020]

[12]. URL: <https://tae.vn/phim-keypad-matrix-4x4-nhua>. [Ngày truy cập: 20/05/2020]

[13]. URL: <https://hshop.vn/products/arduino-mega-2560r3>. [Ngày truy cập: 20/05/2020]