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

Mật mã và an ninh mạng

Đề tài

Ứng ụng mã hóa dữ liệu

GVHD: Nguyễn Trần Hữu Nguyên
SV: Nguyễn Trần Lê Minh - 1511003
Lê Duy Thanh - 1512990
Nguyễn Xuân Nam - 1512098

TP. HỒ CHÍ MINH, THÁNG 03/2019

Mục lục

Mã hóa là phương pháp bảo vệ dữ liệu cá nhân nhạy cảm trên máy tính của bạn. Việc mã hóa còn ngăn chặn bất cứ ai đọc dữ liệu của bạn khi bạn gửi thông tin qua mạng hay đồng bộ lên máy chủ, cloud,...

Trong bài tập lớn này, nhóm sẽ thực hiện một số giải thuật mã hóa dễ cho các tập tin trong máy được an toàn.

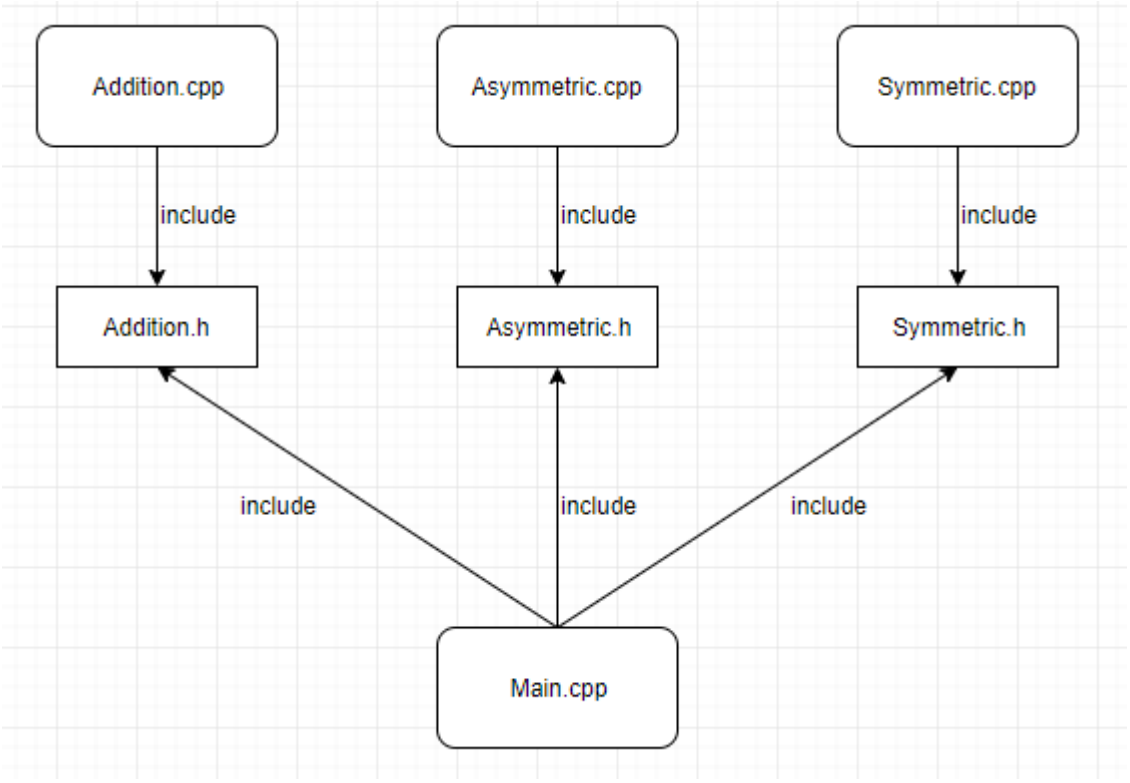
1 Giới thiệu đề tài

Nhóm đã hiện thực ba giải thuật mã hóa phổ biến đó là RSA, DES và Steganography.

Với RSA và DES, nhóm đọc dữ liệu cần mã hóa (plaint text) từ tệp văn bản (*.txt file - text file). Đối với Steganography, nhóm đọc dữ liệu từ text file trộn vào một ảnh mang. Để chứng minh plaint text trùng với dữ liệu được giải mã, nhóm sử dụng hàm băm để đối chiếu hai tệp.

2 Cấu trúc của ứng dụng

Cấu trúc của ứng dụng được miêu tả trong hình 1: Chương trình gồm 4 mô đun chính là:



Hình 1: Kiến trúc chính của hệ thống

- Symmetric: Chứa các hàm mã hóa đối xứng tập tin sử dụng giải thuật DES.

- Asymmetric: Chứa các hàm mã hóa bất đối xứng tập tin sử dụng giải thuật RSA.
- Addition: Chứa các hàm mã hóa tập tin sử dụng kỹ thuật giấu tin (Steganography).
- Main: Đọc các đối số và gọi các hàm phù hợp trong ba thư viện trên.

2.1 Symmetric

Giải thuật mã hóa đối xứng nhóm sử dụng là DES với ECB mode. Trong tệp "Symmetric.h":

```
char *stringFromFile(char filename[], const char type[]);
```

Trả về chuỗi giá trị trong tệp filename.
type: chế độ đọc tệp ("r", "rb", ...).
Lưu ý: Cần giải phóng vùng nhớ cho con trỏ trả về do có sử dụng malloc trong hàm này.

```
unsigned char *DES_encrypt(EVP_CIPHER_CTX *en, unsigned char *plaintext, int *plain_l
```

Mã hó plaintext với độ dài plain_len theo giải thuật được quy định trong (*en). Trả về giá trị của chuỗi mã hóa. Lưu ý: Cần giải phóng vùng nhớ cho con trỏ trả về do có sử dụng malloc trong hàm này.

```
char *DES_decrypt(EVP_CIPHER_CTX *de, unsigned char *ciphertext, int *cipher_len);
```

Ngược với hàm trên, hàm này giải mã ciphertext với độ dài cipher_len.
Giá trị của (*de) quy định giải thuật sử dụng để giải mã.
Giá trị trả về là plaintext.
Lưu ý: Cần giải phóng vùng nhớ cho con trỏ trả về do có sử dụng malloc trong hàm này.

2.2 Asymmetric

Giải thuật mã hóa bất đối xứng nhóm chọn là RSA.Trong tệp "Asymmetric.h":

```
RSA * create_RSA(RSA *keypair, int pem_type, char *file_name);
```

Được sử dụng để tạo ra cặp key (private và public key) cho việc mã hóa và giải mã.
Key được lưu trong tệp file_name.

```
int public_encrypt(int flen, unsigned char* from, unsigned char *to, RSA* key, int pa
```

Sử dụng public key để mã hóa chuỗi (*from) với độ dài flen và lưu chuỗi mã hóa vào (*to). Trả về độ dài của chuỗi mã hóa.

```
int private_decrypt(int flen, unsigned char* from, unsigned char *to, RSA* key, int p
```

Sử dụng private key để giải mã chuỗi (*from) với độ dài flen và lưu vào chuỗi (*to).
Trả về giá trị độ dài của plaintext.

2.3 Addition

Giải thuật mã hóa ngoài bài giảng nhóm sử dụng là kỹ thuật che giấu tập tin. Trong tệp "Addition.h":

```
void steganography_encode(char* inputfile, Mat image, char*outputfile);
```

Đọc tệp inputfile dưới dạng bit.
Sử dụng phép tính "and" bit cho mỗi bit trong inputfile ứng với giá trị bit cuối trong mỗi kênh màu, pixels của ảnh image. Từ đó thu được một ảnh có sai khác rất nhỏ so với ảnh gốc.
Ghi ảnh kết quả vào tệp outputfile.

```
void steganography_decode(Mat image, char*outputfile);
```

Từ ảnh image, rút ra thông điệp và lưu nó vào outputfile.

3 Các giao thức sử dụng

3.1 Hypertext Transfer Protocol

Hypertext Transfer Protocol(HTTP)[1] là một giao thức dùng trong việc truyền các dữ liệu trên các web. Là một phương thức sử dụng mô hình client-server. Server sẽ lắng nghe ở cổng 80 và trả lời các yêu cầu từ client. Dữ liệu trả về từ server bao gồm các mã trạng thái (status code) để cho biết tình trạng truy cập của người dùng. Các mã trạng thái thường gặp như 200 -thành công, 301- chuyển trang, 404- trang không tìm thấy....

HTTP cung cấp nhiều phương thức cho người sử dụng nhưng có 2 phương thức được sử dụng chính trong bài tập lớn của nhóm:

- POST: đây là phương thức dùng để gửi dữ liệu từ gateway lên server. POST cho phép gửi kèm dữ liệu từ client tới server trong gói tin.
- GET: đây là phương thức dùng để lấy dữ liệu từ server mà trong bài tập lớn này là dữ liệu JSON từ server.

3.2 Stop and Wait

...

4 Các tính năng của hệ thống

Hệ thống đáp ứng được các yêu cầu cơ bản của một hệ thống theo dõi độ ẩm đất bao gồm:

- Thu thập dữ liệu về độ ẩm đất từ các cảm biến.
- Vẽ biểu đồ cho thấy sự biến động về độ ẩm trong ngày.
- Có khả năng kiểm tra các thông báo về nhiệt độ và lượng mưa của khu vực bằng các opensource API[2].
- Hệ thống có sự quản lí phân cấp và có khả năng mở rộng.

5 Hướng dẫn vận hành

5.1 Cách cài đặt server

Để có thể cài đặt và sử dụng localhost, chúng ta cần cài đặt một phần mềm có khả năng lắng nghe cổng 80 của máy tính. Ở đây nhóm sử dụng phần mềm xampp là một phần mềm miễn phí bao gồm :

- Apache: cho phép lắng nghe cổng 80 để tiếp nhận các yêu cầu.
- Mysql: đây là cơ sở dữ liệu có cấu trúc miễn phí.
- PHP: là một ngôn ngữ rất phổ biến được sử dụng ở server.

Phần mềm có thể tải về tại đường dẫn: https://www.apachefriends.org/index.html.

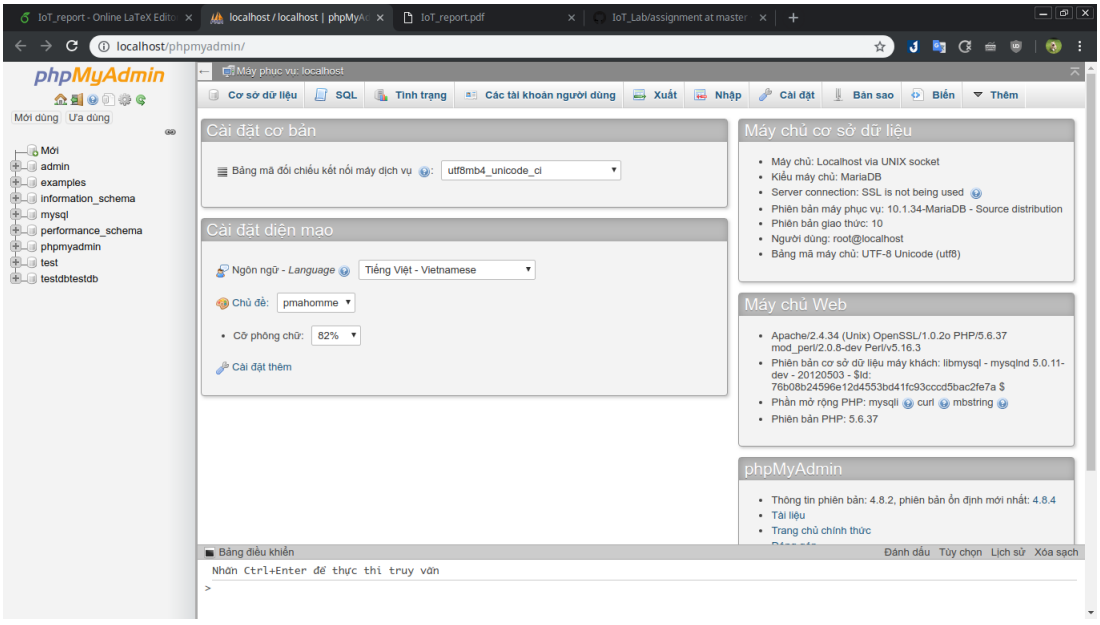
Sau khi cài đặt xampp, chúng ta có thể sử dụng mạng wifi hoặc tự phát wifi.

Kế tiếp, để có thể gửi dữ liệu từ client lên server ta cần sửa lại địa chỉ của server (mặc định là 127.0.0.1) thành địa chỉ trong mạng LAN bằng cách sửa file *httpd.conf* tại đường dẫn *etc/httpd.conf*

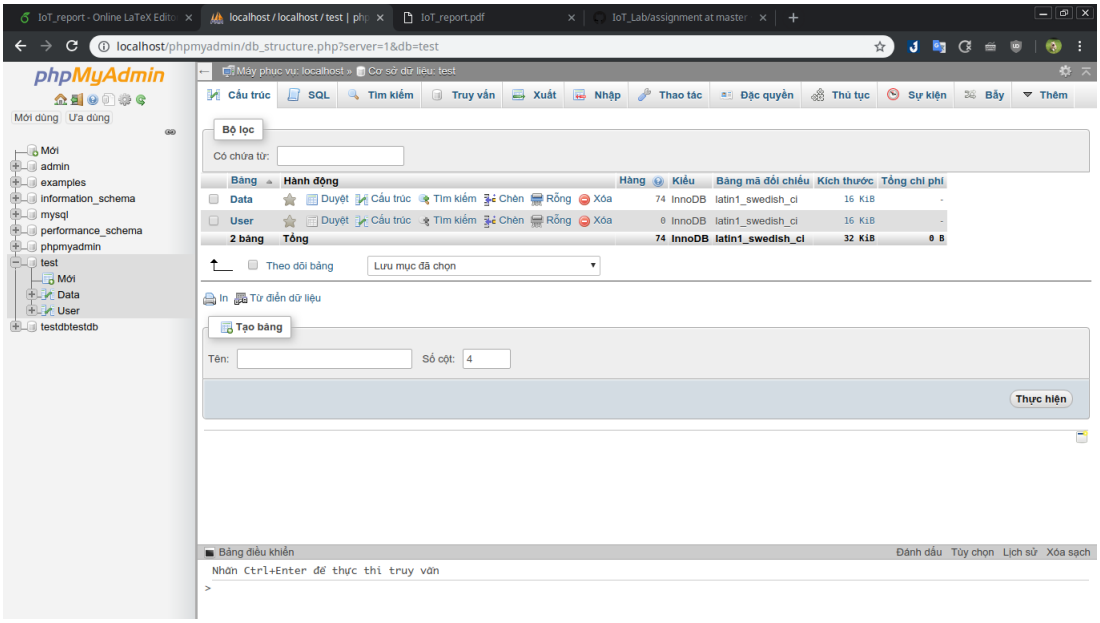
51 Listen <your local ip address>

Tải toàn bộ mã nguồn tại: https://github.com/lochoang75/IoT_Lab

Khởi chạy xampp và vào đường dẫn http://localhost/phpmyadmin để nhập cơ sở dữ liệu.



Hình 2: Giao diện phpmyadmin



Hình 3: Kết quả sau khi nhập thành công database

Tạo một bảng có tên test và chọn vào **nhập(import)**, chọn file *test.sql* để chèn cơ sở dữ liệu bao gồm 2 bảng như hình bên dưới.

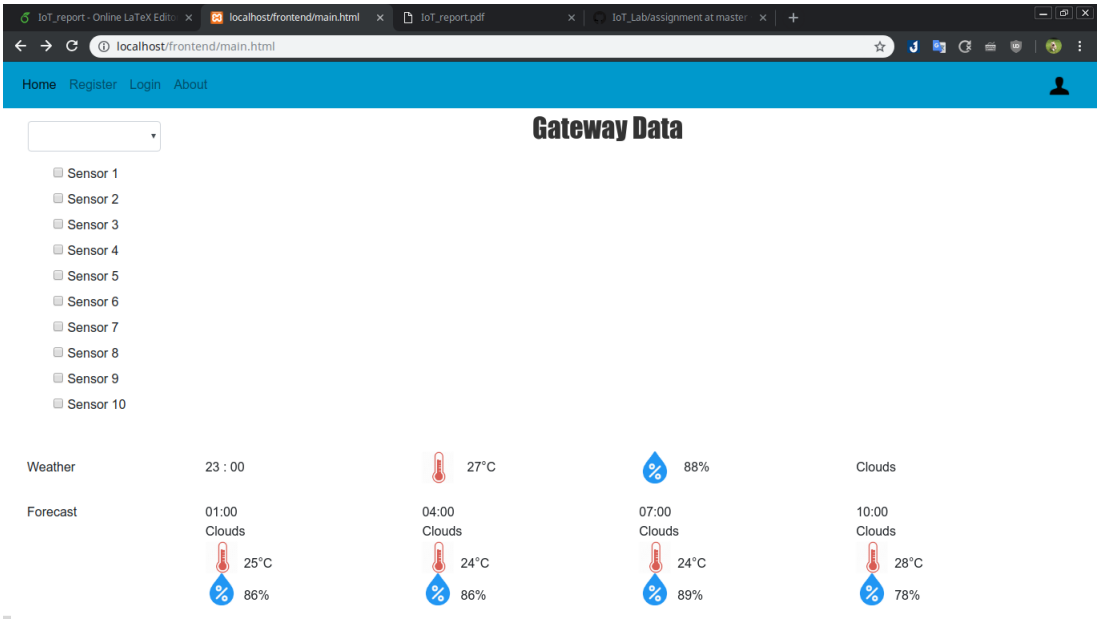
Cuối cùng chuyển thư mục backend và frontend vào thư mục htdocs(thư mục root mặc định của localhost). Nếu thành công các đường dẫn cho các file ở server sẽ là:

- about.html: `http://localhost/frontend/about.html`
- main.html: `http://localhost/frontend/main.html`
- register.html: `http://localhost/frontend/register.html`
- login.html: `http://localhost/frontend/login.html`
- json.php: `http://localhost/backend/json.php`
- data.php: `http://localhost/backend/data.php`
- login.php: `http://localhost/backend/login.php`

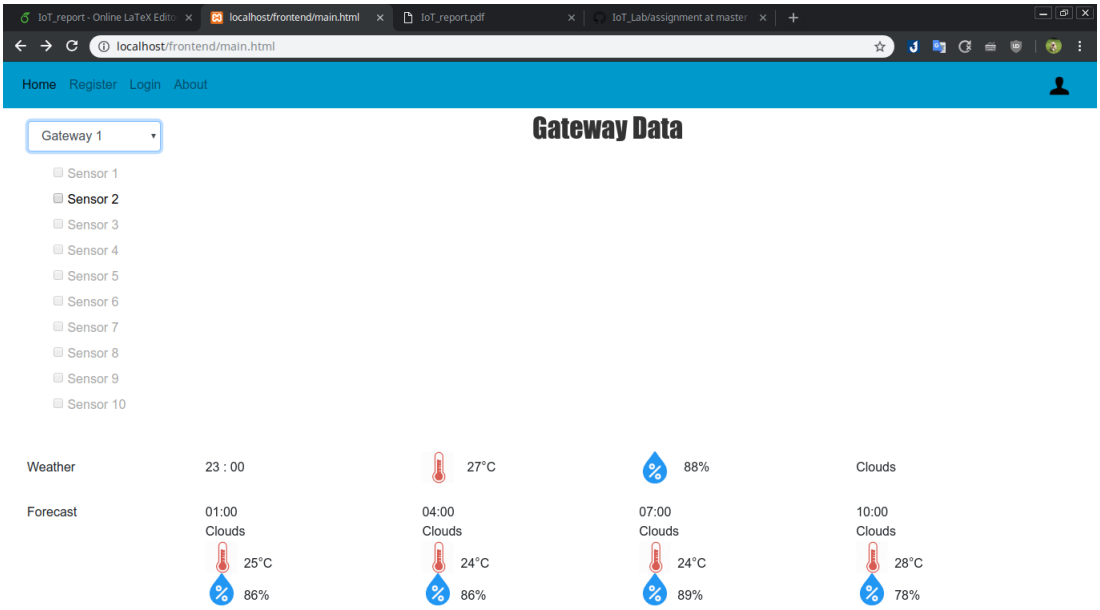
5.2 Sử dụng các tính năng của website

Sau khi cài đặt xong server, trang chính sẽ như hình sau:

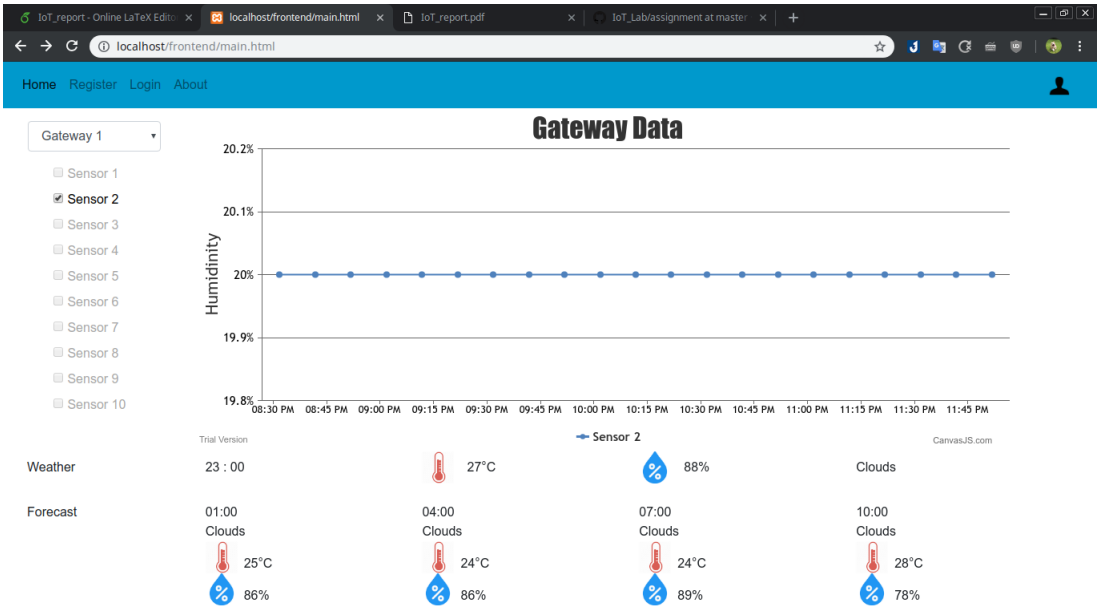
Để hiển thị được biểu đồ, người dùng cần chọn gateway cần hiển thị. Các cảm biến có dữ liệu sẽ cho phép chọn: Cuối cùng chọn một hoặc một vài cảm biến và đồ thị của cảm biến đó sẽ được vẽ ra. Phần dưới của website là các thông tin dự báo về nhiệt độ và độ ẩm theo từng thời điểm trong ngày và được cập nhật mỗi giờ.



Hình 4: Giao diện trang chính



Hình 5: Giao diện sau khi chọn gateway



Hình 6: Giao diện sau khi chọn gateway

5.3 Sử dụng ứng di động

5.3.1 Android

Bước đầu tiên để sử dụng ứng dụng là cài đặt. Sử dụng android studio chạy mã nguồn từ địa chỉ https://github.com/lochoang75/IoT_Lab/tree/master/assignment/iotapp.
Sau khi cài đặt, mở ứng dụng ta thấy màn hình đăng nhập như hình 7

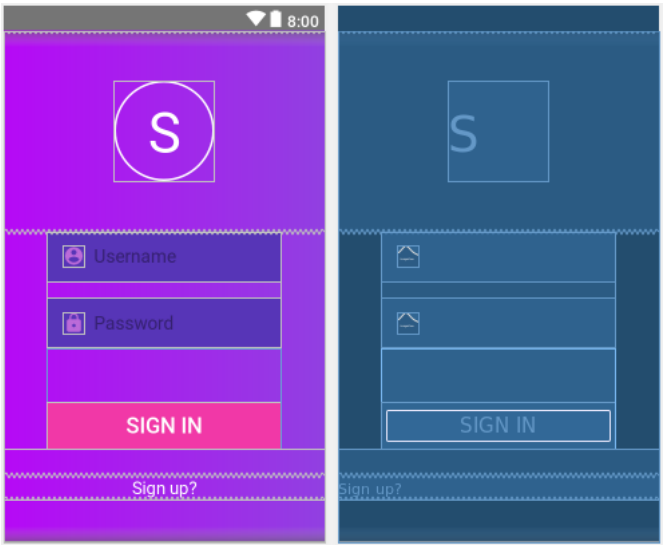
1. **Màn hình đăng nhập**

   Chọn đăng nhập chúng ta chuyển đến màn hình chính của ứng dụng như hình 8a.

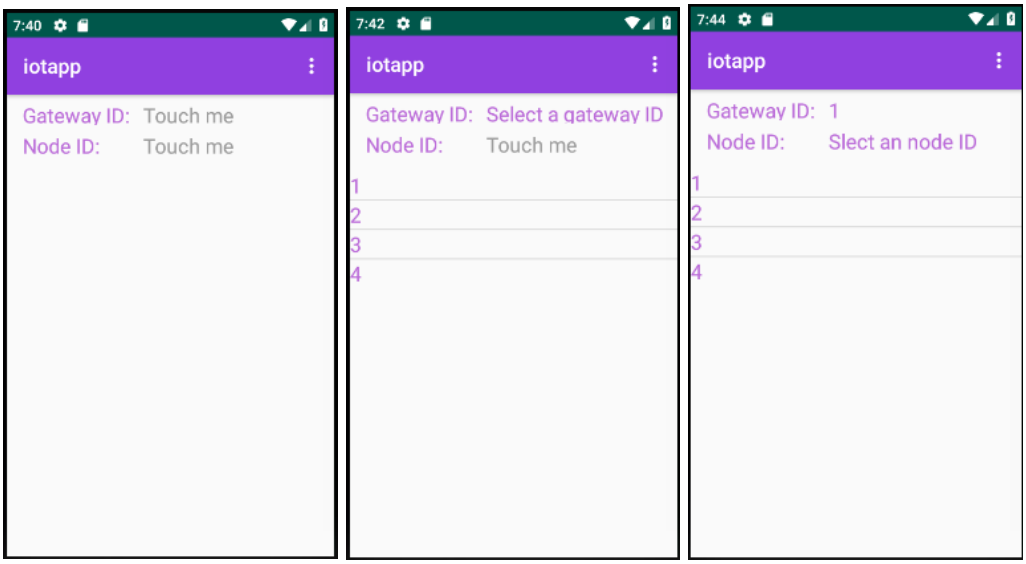
2. **Màn hình chính**

   Tại một thời điểm chỉ có thông tin của một cảm biến được hiển thị, vì thế chúng ta cần chọn định danh cho nó (bao gồm gatewayID và nodeId). Chạm vào "Touch me" để chọn các thông tin tương ứng. Lưu ý: NodeId chỉ được chọn khi gatewayId hợp lệ (đã được chọn).
   Sau khi chọn xong định danh, trên màn hình sẽ hiển thị một danh sách thông tin thu thập được từ cảm biến như hình 9. Để tải lại dữ liệu, người dùng có thể lướt màn hình từ trên xuống.

   Trên góc trên bên phải màn hình có nút menu. Chọn menu để chuyển tới các màn hình khác. Khi chọn menu, chúng ta thấy xuất hiện hình 10

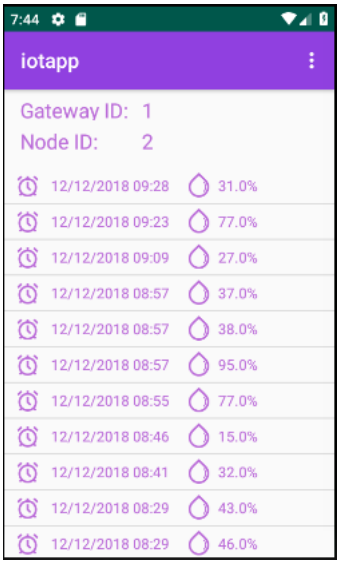


Hình 7: Giao diện sau khi chọn gateway

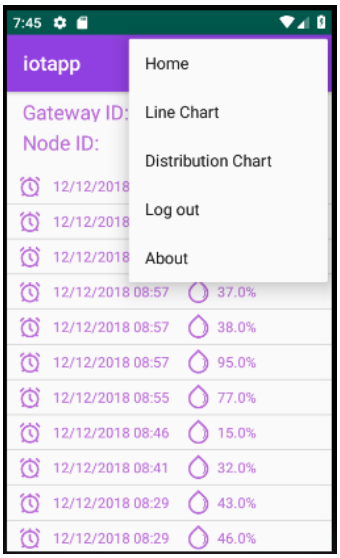


(a) Khởi tạo (b) Chọn gatewayID (c) Chọn nodeID

Hình 8: Màn hình chính



Hình 9: Danh sách dữ liệu



Hình 10: Menu của ứng dụng

3. **Menu**
   Menu bao gồm các phần sau:

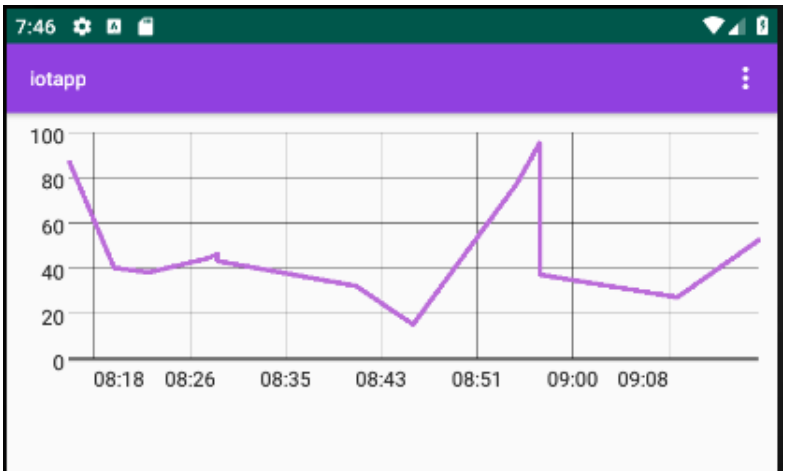
   - **Home** Chuyển đến màn hình chính.
   - **Line Chart** Chuyển đến màn hình biểu đồ đường (xem mục 4)
   - **Distribution Chart** Chuyển đến màn hình biểu đồ phân b.(xem mục 5)
   - **Logout** Chuyển đến màn hình đăng nhập.

4. **Màn hình biểu đồ đường**

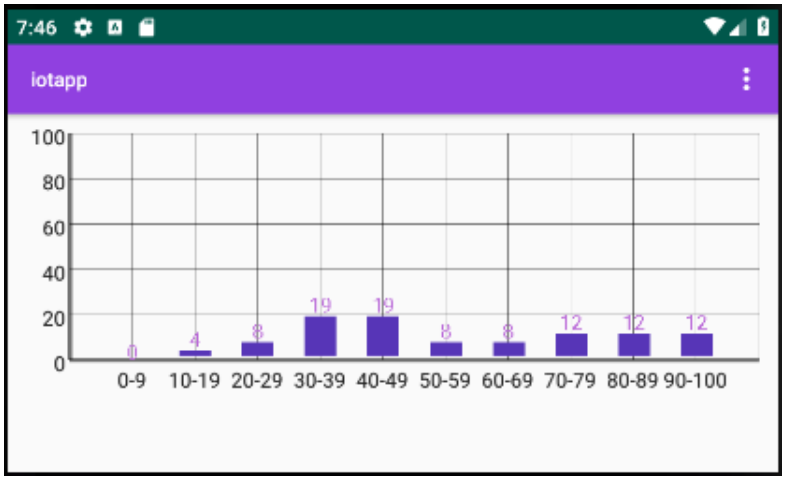
   Dữ liệu từ màn hình chính sẽ được vẽ thành biểu đồ tại đây. Lưu ý: chỉ có thể chuyển đế màn hình này khi định danh của cảm cảm biến đã được chọn.
   Trục X là trục thời gian từ 0 giờ đến 24 giờ.
   Trục Y là trục giá trị độ ẩm từ 0 đến 100 phần trăm.

5. **Màn hình biểu đồ phân bố**

   Dữ liệu từ tệp json sẽ được tính ra phần trăm đóng góp của các nhóm và được biểu thị trên biểu đồ.
   Trục X là trục của các nhóm tương ứng mới giá trị độ ẩm của mỗi data nhận được.
   Trục Y là giá trị phần trăm đóng góp của mỗi nhóm mà mỗi nhóm đóng góp.



Hình 11: Biểu đồ đường



Hình 12: Biểu đồ phân bố

6 Kết luận

Kết luận về các nội dung và kết quả thực hiện được.

Tài liệu

[1] HTTP Definnition, https://techterms.com/definition/http

[2] https://openweathermap.org/api