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

1 Giới thiệu đề tài **2**

2 Cấu trúc của ứng dụng **2**
2.1 Mạng cảm biến . . . 3
2.2 Server . . . 3
2.2.1 Backend . . . 3
2.2.2 Frontend . . . 3
2.3 Ứng dụng di động . . . 4
2.3.1 Android . . . 4

3 Các giao thức sử dụng **6**
3.1 Hypertext Transfer Protocol . . . 6
3.2 Stop and Wait . . . 6

4 Các tính năng của hệ thống **6**

5 Hướng dẫn vận hành **7**
5.1 Cách cài đặt server . . . 7
5.2 Sử dụng các tính năng của website . . . 9
5.3 Sử dụng ứng di động . . . 9
5.3.1 Android . . . 9

6 Kết luận **14**

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.

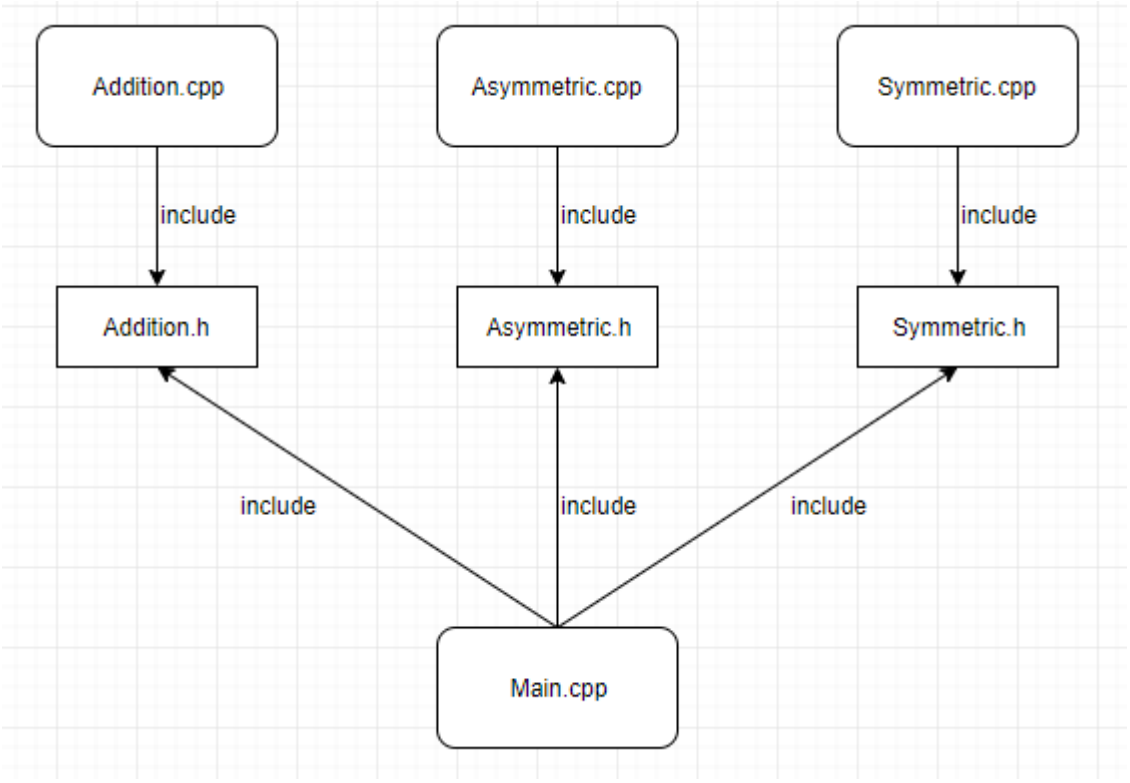
1 Giới thiệu đề tài

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 tập tin sử dụng giải thuật DES.

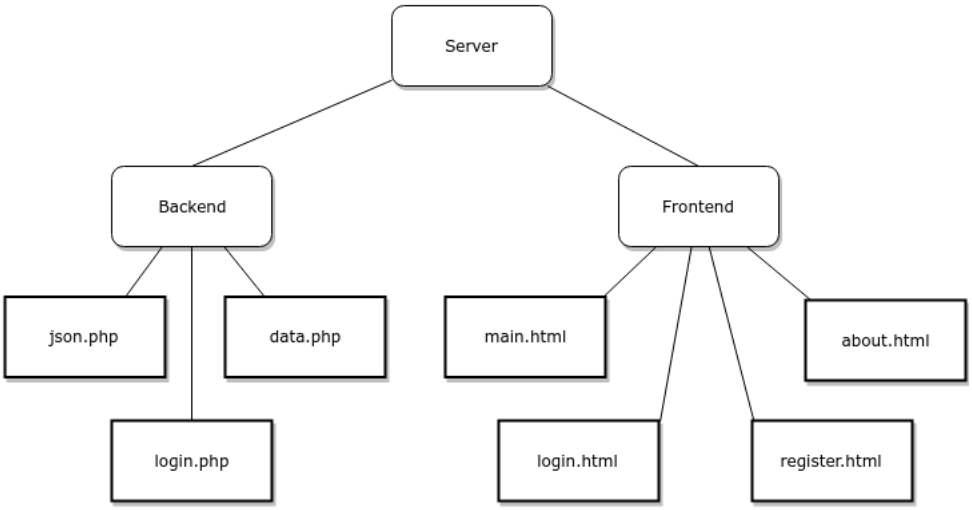
- Asymmetric: Chứa các hàm mã hóa tập tin sử dụng giải thuật RSA.
- Addition: Chứa các hàm mã hóa tập tin sử dụng kỹ thuật giấu tin (Steganography).
- Main: Đọc các đối số và gọi các hàm phù hợp trong ba thư viện trên.

2.1 Mạng cảm biến

...

2.2 Server

Các tệp tin và các trang ở server được tổ chức như sau: Các trang tại server phục vụ cho các



Hình 2: Kiến trúc tệp tin tại server

yêu cầu của đề tài:

2.2.1 Backend

- json.php: trả về dữ liệu dưới dạng JSON. Dữ liệu này được các cảm biến gửi lên và chứa trong cơ sở dữ liệu.
- data.php: nơi các gateway sẽ gửi dữ liệu được tổng hợp từ các sensor. Dữ liệu này sau đó sẽ được đưa vào cơ sở dữ liệu.
- login.php: xử lí các đăng kí và đăng nhập từ người dùng.

2.2.2 Frontend

- main.html: trang chính của người dùng, cung cấp việc lựa chọn các gateway đang có dữ liệu và vẽ biểu đồ theo thời gian.
- about.html: các thông tin về nhóm phát triển và đề tài.

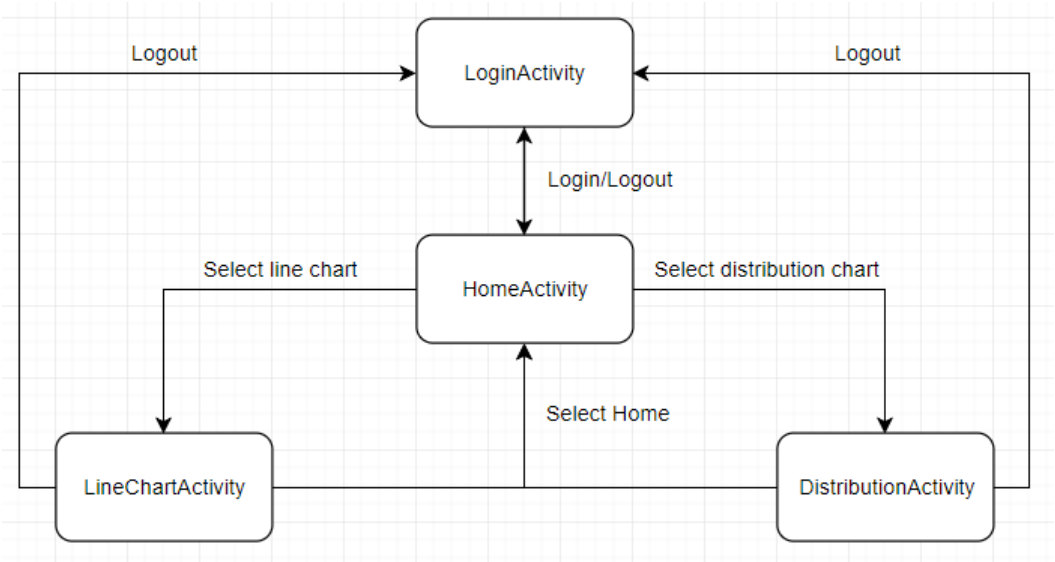
- register.html: trang chưa thông tin đăng kí bởi người dùng.
- login.html: trang đăng nhập để vào ứng dụng.

2.3 Ứng dụng di động

2.3.1 Android

1. **Sơ đồ trạng thái của ứng dụng.**
   Với dứng dụng android, phần mềm được thực thi bằng các activity ứng với mỗi nhiệm vụ khác nhau. Trong bái tập lớn này, nhóm sử dụng các activity cơ bản sau:
   - Đăng nhập (MainActivity hay LoginActivity),
   - Danh sách dữ liệu thô (HomeActivity),
   - Biểu đồ đường (LineChartActivity),
   - Biểu đồ phân bố (DistributionChartActivity).

   Các activity giao tiếp với nhau thông qua intent của android theo sơ đồ hình 3:

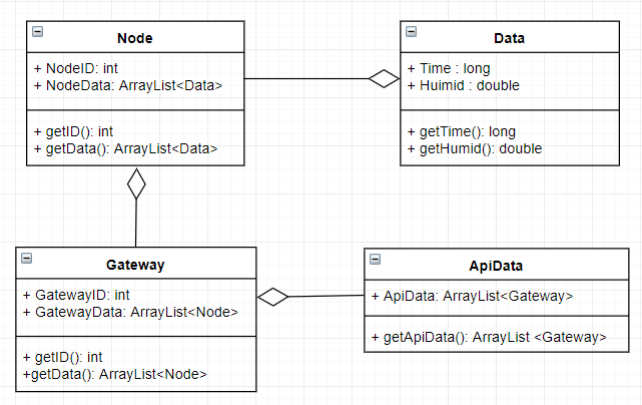


Hình 3: Sơ đồ chuyển trạng thái của các activity

LoginActivity là hoạt động đầu tiên khi bắt dầu sử dụng ứng dụng.
HomeActivity quản lí một danh sách các dữ liệu thô lấy từ server. Danh sách này chỉ trỏ đến một nốt cụ thể (bao gồm một gatewayID và một nodeId) và chứa thông tin trong này ngày (tính từ 0 giờ). Vì dữ liệu có thể quá dài nên nhóm quyết định đặt gới hạn danh sách dài chỉ 20 phần tử.
LineChartActivity: Giống HomeActivity, hoạt động này cũng lấy một mảng các gateway từ server. Tuy nhiên, dữ liệu lấy về được vẽ thành biểu đồ đường thể thấy trực quan sự biển đổi theo thời gian. Và gới hạn của các điểm vẽ biểu đồ cũng là 20 giống HomeActivity.
DistributionChartActivity: Các activity đều lấy dữ liệu từ server về. Với hoạt động

này. Các dữ liệu lấy về sẽ được chia thành 10 nhóm tương ứng với 0-9,10-19,...90-100. Số lượng phần tử trong mỗi nhóm sẽ là căn cứ để tính ra phần trăm đóng góp của môi nhóm trong toàn một dữ iệu lấy về được. Từ đó là vẽ được biểu đồ phân bố giá trị.

2. **Cấu trúc dữ liệu của ứng dụng.**
Cấu trúc dữ liệu của ứng dụng được tổ chức theo cấu trúc của tệp json lấy từ server và được thể hiện thông qua sơ đồ hình 4.



Hình 4: Cấu trúc dữ liệu của ứng dụng

Data là lớp chứa dữ liệu mà một cảm biến gửi cho gateway tại một thời điểm. Vì vậy lớp này gồm 2 thuộc tính là thời gian (time) và độ ẩm (huminity).
Node là đơn vị quản lí dữ liệu của một cảm biến. Nó gồm định danh của cảm biến và một mảng các giá trị mà cảm biến thu nhận được.
Gateway: Một gateway quản lí nhiều cảm biến bởi vậy nó cần chứa một mảng các cảm biến (bao gồm định danh và dữ liệu của chúng). Đồng thời mỗi gateway cũng có định danh riêng của chúng.
ApiData là dữ liệu nơi lưu trữ dữ liệu lấy được tệp json. Nó gôm một mảng các gateway.

3. **Lấy dữ liệu từ tệp json**.
Lấy dữ liệu từ server là một phần quan trọng của ứng dụng. Việc này được quản lí bởi ApiController với cấu trúc hình 5:

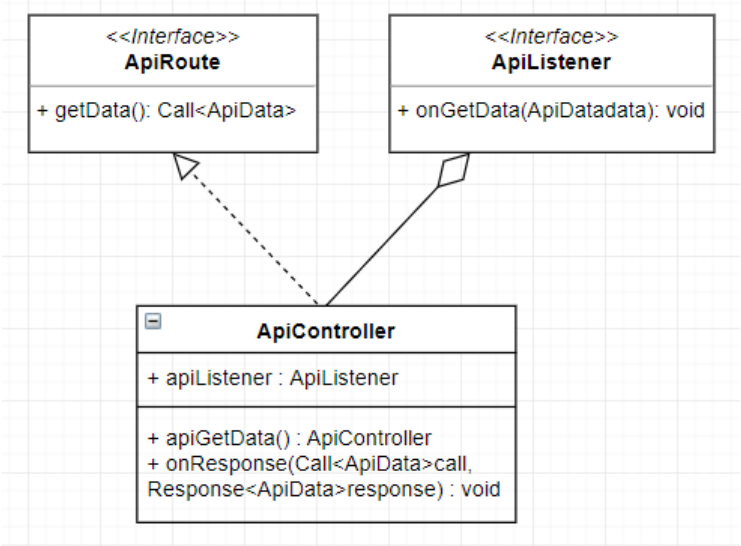
ApiController chứa các phần tử sau:

- Đường dẫn đến địa chỉ chứa tệp json. Ví dụ:

```
public static final String BASE_URL = "http://10.20.83.74/backend/";
```

- Sử dụng thư viện retrofit2 hiện thực phương thức getData() và trong constructure và trả dữ liệu về phương thức onRespone.

ApiRoute bao gồm địa chỉ của tệp json và phương thức getData().
ApiListener chứa phương thức onGetData sử lí dữ liệu lấy được từ json và được hiện thực trong các activity.



Hình 5: Cấu trúc của ApiController

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.

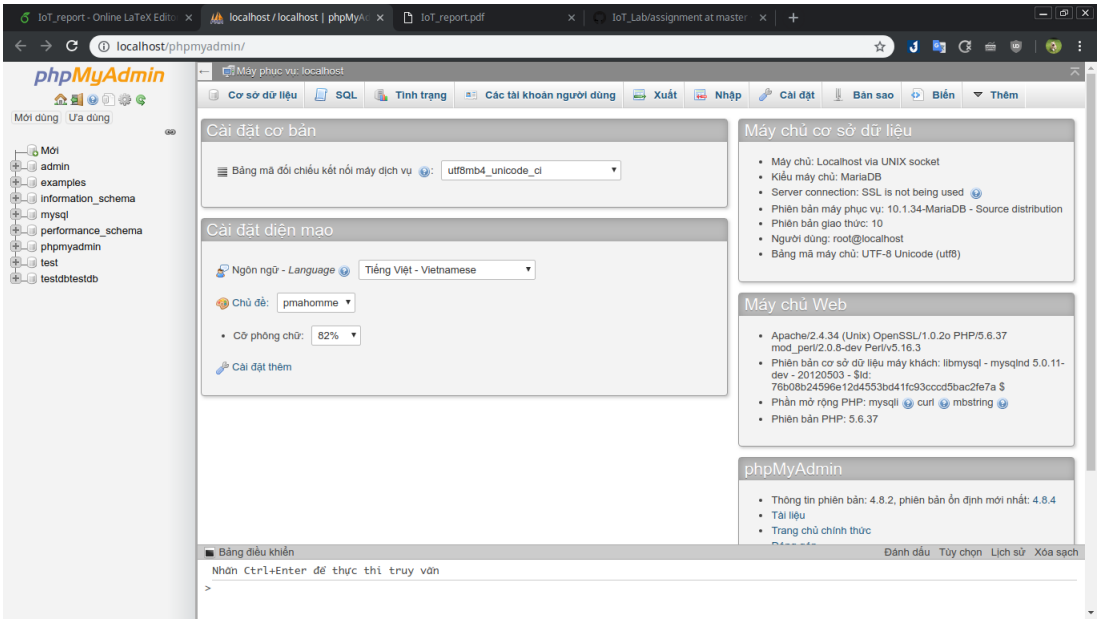
Kế tiếp, để có thể gửi dữ liệu từ client lên server ta cần sửa lại địa chỉ của server (mặc định là 127.0.0.1) thành địa chỉ trong mạng LAN bằng cách sửa file *httpd.conf* tại đường dẫn *etc/httpd.conf*

51 Listen <your local ip address>

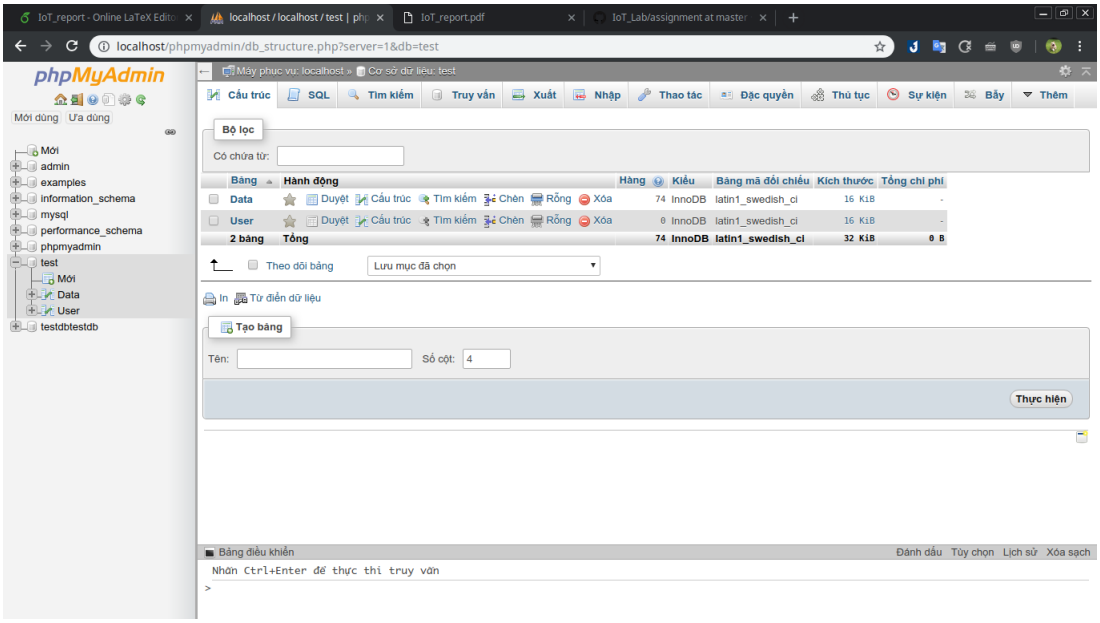
Tải toàn bộ mã nguồn tại: https://github.com/lochoang75/IoT_Lab

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

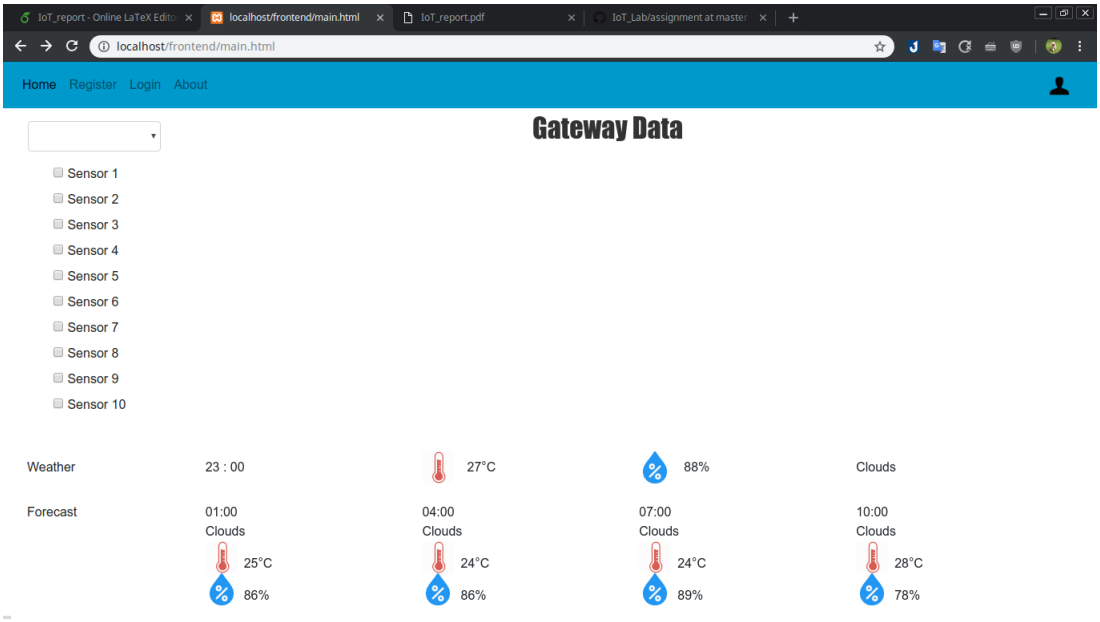
- about.html: http://localhost/frontend/about.html
- main.html: http://localhost/frontend/main.html
- register.html: http://localhost/frontend/register.html
- login.html: http://localhost/frontend/login.html
- json.php: http://localhost/backend/json.php
- data.php: http://localhost/backend/data.php
-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



Hình 6: Giao diện phpmyadmin



Hình 7: Kết quả sau khi nhập thành công database



Hình 8: Giao diện trang chính

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.

5.3 Sử dụng ứng di động

5.3.1 Android

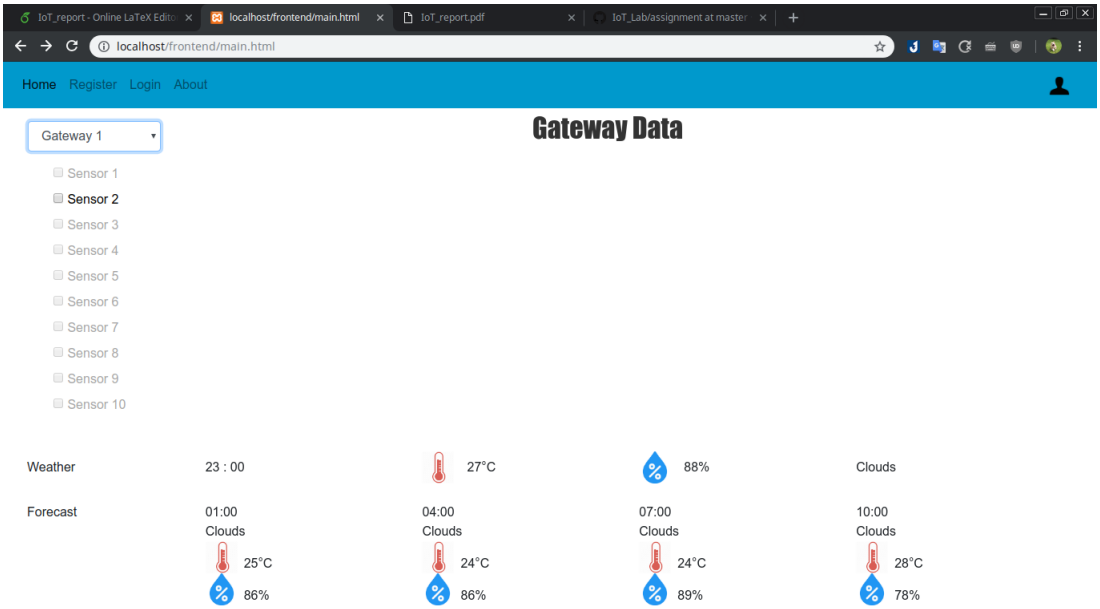
Bước đầu tiên để sử dụng ứng dụng là cài đặt. Sử dụng android studio chạy mã nguồn từ địa chỉ https://github.com/lochoang75/IoT_Lab/tree/master/assignment/iotapp.
Sau khi cài đặt, mở ứng dụng ta thấy màn hình đăng nhập như hình 11

1. **Màn hình đăng nhập**

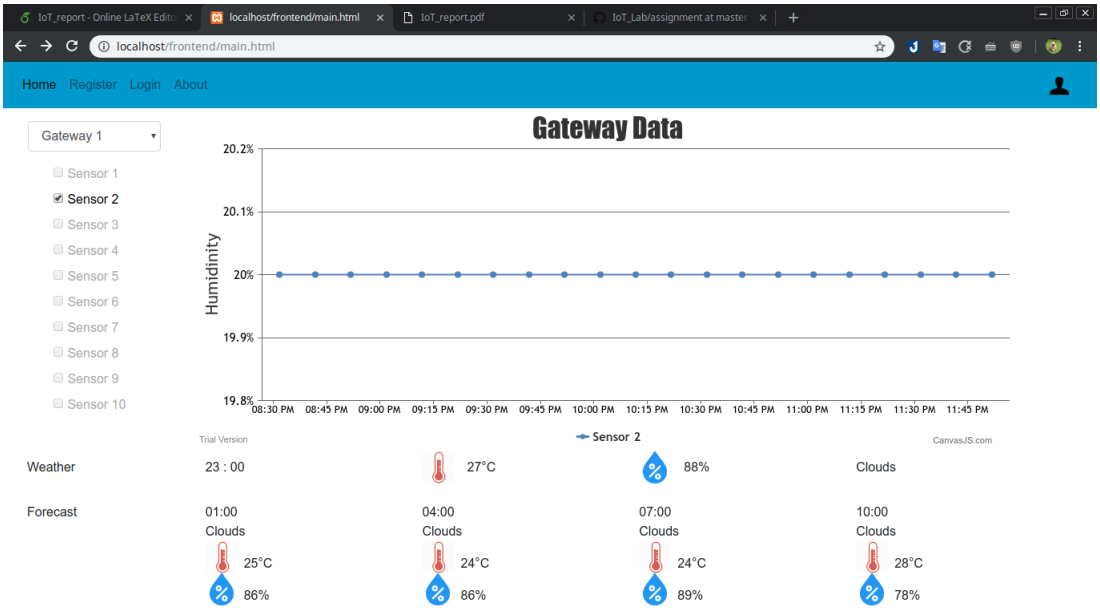
   Chọn đăng nhập chúng ta chuyển đến màn hình chính của ứng dụng như hình 12a.

2. **Màn hình chính**

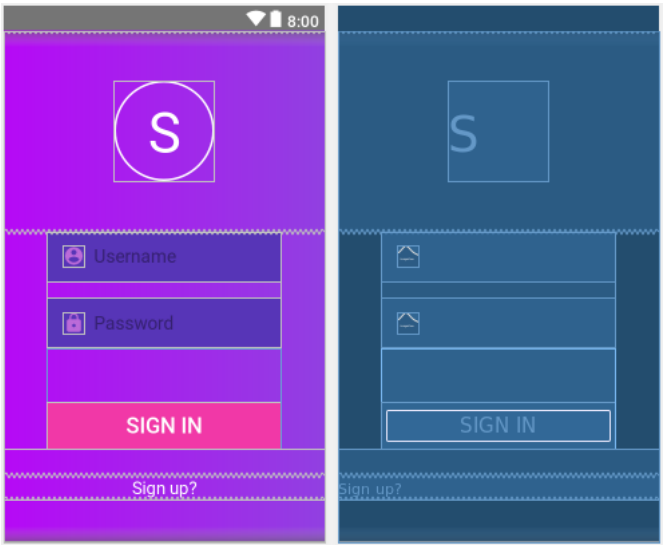
   Tại một thời điểm chỉ có thông tin của một cảm biến được hiển thị, vì thế chúng ta cần chọn định danh cho nó (bao gồm gatewayID và nodeId). Chạm vào "Touch me" để chọn các thông tin tương ứng. Lưu ý: NodeId chỉ được chọn khi gatewayId hợp lệ (đã được chọn).



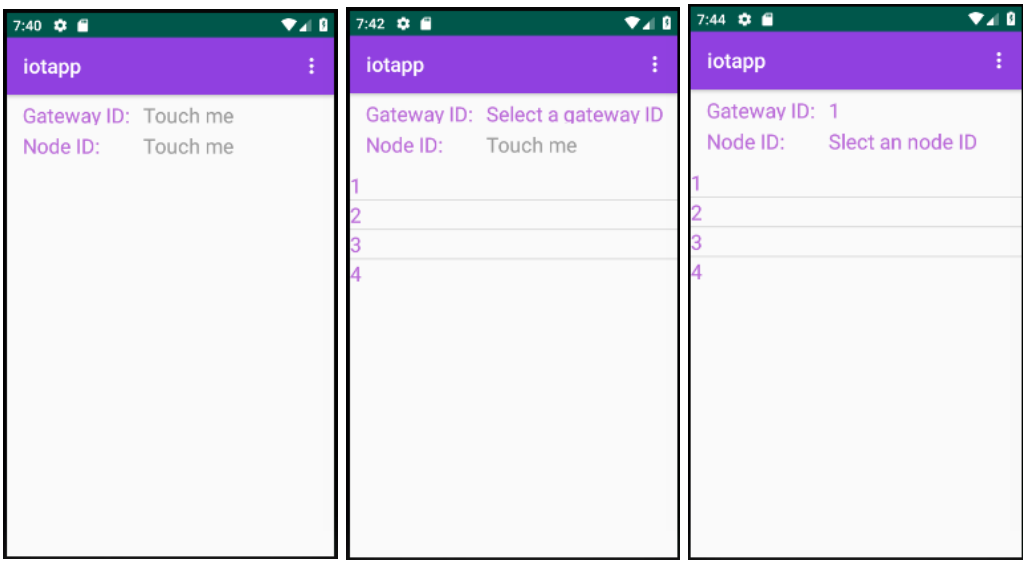
Hình 9: Giao diện sau khi chọn gateway



Hình 10: Giao diện sau khi chọn gateway



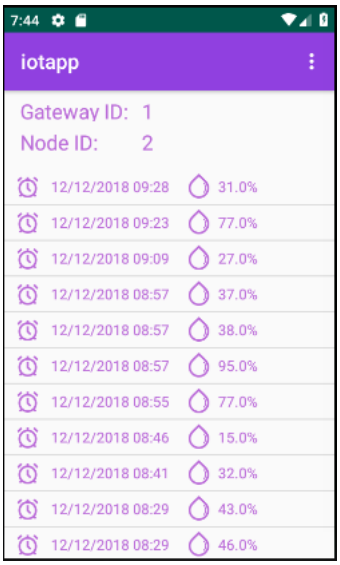
Hình 11: Giao diện sau khi chọn gateway



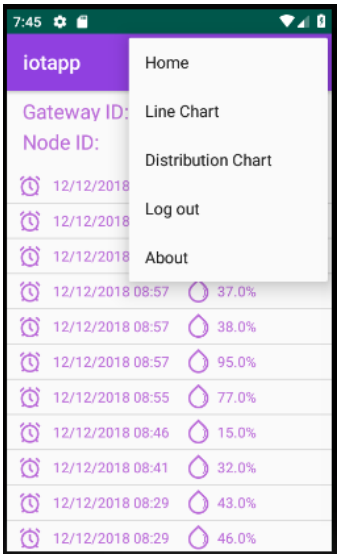
(a) Khởi tạo (b) Chọn gatewayID (c) Chọn nodeID

Hình 12: Màn hình chính

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 13. Để tải lại dữ liệu, người dùng có thể lướt màn hình từ trên xuống.



Hình 13: Danh sách dữ liệu



Hình 14: Menu của ứng dụng

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 14

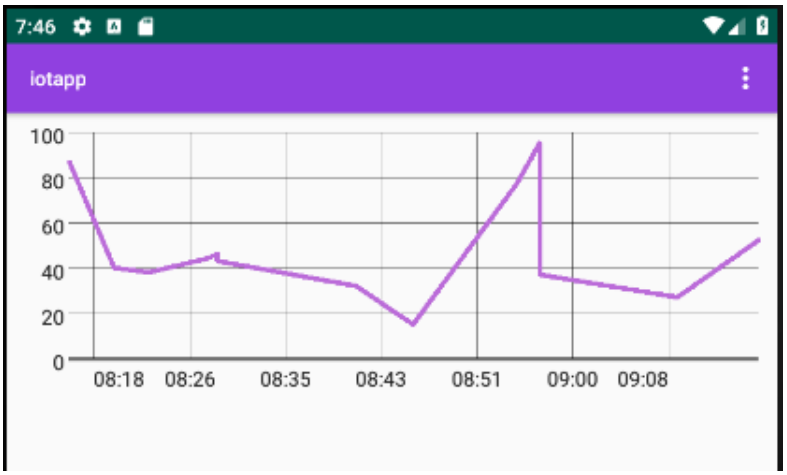
3. **Menu**
   Menu bao gồm các phần sau:

   - **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.

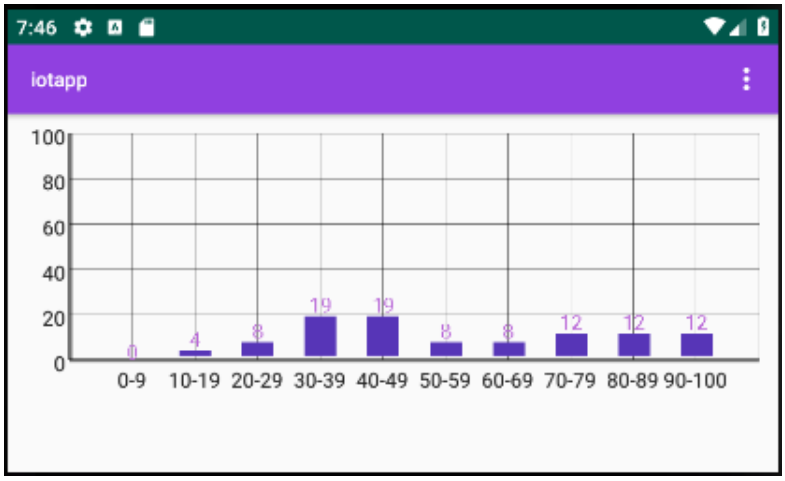
4. **Màn hình biểu đồ đường**

   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**



Hình 15: Biểu đồ đường



Hình 16: 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.

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