**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**CÔNG PHƯƠNG ĐÔNG**

**DƯƠNG VŨ THÁI CƯỜNG**

**KHÓA 11**

**HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN**

**TÌM HIỂU LẬP TRÌNH MẠCH NHÚNG VÀ**

**PHÁT TRIỂN ỨNG DỤNG IoT**

**Năm 2017**

**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**CÔNG PHƯƠNG ĐÔNG**

**DƯƠNG VŨ THÁI CƯỜNG**

**KHÓA 11**

**HỆ ĐÀO TẠO KỸ SƯ DÂN SỰ**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**MÃ SỐ: 5248020103**

**TÌM HIỂU LẬP TRÌNH MẠCH NHÚNG VÀ**

**PHÁT TRIỂN HỆ THỐNG IoT**

**Cán bộ hướng dẫn:** 4/, GV, Ths. Nguyễn Trung Thành

1//, GV, TS. Tạ Minh Thanh

**Năm 2017**

|  |  |
| --- | --- |
| **BỘ QUỐC PHÒNG**  **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA CÔNG NGHỆ THÔNG TIN**  **Phê chuẩn**  Ngày …. tháng …. năm ……  **CHỦ NHIỆM KHOA** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP – TỰ DO – HẠNH PHÚC**  Độ mật: ……………..  Số: …………………. |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: Công Phương Đông Lớp: Tin học 11A Khóa: 11

Ngành: Công nghệ thông tin Chuyên ngành: Hệ thống thông tin

1. Tên đề tài:

Tìm hiểu lập trình nhúng và phát triển hệ thống IoT

2. Các số liệu ban đầu: Không có

3. Nội dung bản thuyết minh:

Bản báo cáo gồm 5 chương. Nội dung các chương như sau

+ Chương I: Khái quát các kiến thức, các công nghệ sử dụng trong hệ thống.

+ Chương II: Mô tả về hệ thống giám sát phương tiện. Chương này nêu ra các nhiệm vụ cơ bản, cơ cấu tổ chức và quy trình làm việc của hệ thống. Ngoài ra, ở chương này còn đưa ra các mẫu biểu sẽ được sử dụng trong hệ thống.

+ Chương III: Chương này tập trung phân tích và thiết kế hệ thống. Từ quy trình làm việc của hệ thống đưa ra các chức năng nghiệp vụ chính của hệ thống. Ngoài ra, ở chương này còn đưa ra dữ liệu nghiệp vụ và giao diện của hệ thống.

+ Chương IV: Triển khai phần cứng. Đưa ra các lệnh được sử dụng và việc kết hợp giữa Arduino và Module Sim 808 để có thể gửi tọa độ lên server.

+ Chương V: Kết luận những thứ đạt được qua đồ án.

4. Số lượng, nội dung các bản vẽ (ghi rõ loại, kích thước và cách thực hiện các bản vẽ) và các sản phẩm cụ thể (nếu có): Không có

5. Cán bộ hướng dẫn

\* Họ và tên: Nguyễn Trung Thành

\* Cấp bậc: Đại úy

\* Chức vụ: Giáo viên

\* Đơn vị: Bộ môn Công nghệ Mạng

\* Kiểu hướng dẫn:

|  |  |
| --- | --- |
| 🗹 Toàn bộ | 🞎 Từng phần |

|  |  |
| --- | --- |
| Ngày giao: …/…/......  **Chủ nhiệm bộ môn** | Ngày hoàn thành: .../…/…...  Hà Nội, ngày … tháng … năm …...  **Cán bộ hướng dẫn**  (Ký, ghi rõ họ tên, học hàm, học vị) |

**Học viên thực hiện**

Đã hoàn thành và nộp đồ án ngày … tháng ... năm …...

(Ký và ghi rõ họ tên)

**BẢNG GIẢI THÍCH CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| STT | Từ viết tắt | Viết đầy đủ |
| 1 | IoT | Internet of things |
| 2 | GPRS | General Packet Radio Service |
| 3 | GPS | Global Positioning System |
| 4 | IP | Internet Protocol |
| 5 | VM | Virtual Machine |

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc482057997)

[Chương 1: TỔNG QUAN 2](#_Toc482057998)

[1. Các mạch nhúng 2](#_Toc482057999)

[1.1. Mạch Arduino 2](#_Toc482058000)

[1.2. Module Sim808 9](#_Toc482058001)

[2. Các công nghệ liên quan 11](#_Toc482058002)

[2.1. Internet of things 11](#_Toc482058003)

[2.2. General Packet Radio Service 13](#_Toc482058004)

[2.3. Global Positioning System 14](#_Toc482058005)

[2.4. Amazon Webservice 17](#_Toc482058006)

[2.5. REST API 17](#_Toc482058007)

[2.6. C++ REST SDK 18](#_Toc482058008)

[2.7. ASP .NET Core 19](#_Toc482058009)

[2.8. Android 21](#_Toc482058010)

[Chương 2: MÔ TẢ HỆ THỐNG 27](#_Toc482058011)

[1. Nhiệm vụ cơ bản 27](#_Toc482058012)

[2. Cơ cấu tổ chức: 27](#_Toc482058013)

[3. Quy trình làm việc của hệ thống 27](#_Toc482058014)

[4. Mẫu biểu 30](#_Toc482058015)

[Chương 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 32](#_Toc482058016)

[1. Phân tích chức năng nghiệp vụ 32](#_Toc482058017)

[1.1. Mô hình hóa chức năng nghiệp vụ 32](#_Toc482058018)

[1.2. Mô hình hóa tiến trình nghiệp vụ: 33](#_Toc482058019)

[1.3. Biểu đồ hoạt động 37](#_Toc482058020)

[1.4. Thiết kế tiến trình hệ thống 42](#_Toc482058021)

[1.5. Sơ đồ use case 43](#_Toc482058022)

[1.6. Sơ đồ tuần tự 44](#_Toc482058023)

[2. Phân tích dữ liệu nghiệp vụ 54](#_Toc482058024)

[2.1. Xác định kiểu dữ liệu ban đầu 54](#_Toc482058025)

[2.2. Mô hình liên kết thực thể 54](#_Toc482058026)

[2.3. Mô hình quan hệ 57](#_Toc482058027)

[2.4. Đặc tả dữ liệu 60](#_Toc482058028)

[3. Thiết kế giao diện 62](#_Toc482058029)

[3.1. Giao diện quản trị 62](#_Toc482058030)

[3.2. Giao diện người dùng 65](#_Toc482058031)

[Chương 4: TRIỂN KHAI PHẦN CỨNG 72](#_Toc482058032)

[1. Triển khai phần cứng 72](#_Toc482058033)

[1.1. Lập trình với Module Sim808 72](#_Toc482058034)

[1.2. Kết hợp Module Sim808 với Arduino UNO 74](#_Toc482058035)

[KẾT LUẬN 75](#_Toc482058036)

[TÀI LIỆU THAM KHẢO 77](#_Toc482058037)

# **LỜI MỞ ĐẦU**

Internet of things hay còn được gọi là vạn vật kết nối, đang là một xu hướng mới đang dẫn đầu trong cuộc cách mạng về khoa học kỹ thuật. Thời đại IoT đang tăng tốc và sẽ thay đổi cách chúng ta làm việc, sinh hoạt và giải trí hằng ngày. Với những thiết bị IoT, chúng ta không cần phải mất công đi ra chỗ công tắc đèn, quạt … chỉ để bật chúng lên, thay vào đó, chúng ta có thể chỉ cần dùng một chiếc điện thoại thông minh và điều khiển mọi vật trong nhà.

Và để thực hiện việc IoT hóa các đồ vật xung quanh chúng ta thì chúng ta cần đến những mạch nhúng để có thể can thiệp vào cách thức vận hành của chúng. Sức mạnh của IoT có bị hạn chế hay không hoàn toàn phụ thuộc vào các mạch nhúng bên trong các thiết bị trong hệ thống IoT. Chính vì thế mà lập trình mạch nhúng và IoT sẽ đi đôi với nhau, không thể tách rời.

Với đề tài này, em muốn mình có thể tạo ra được một hệ thống giúp người dùng có thể giám sát được các phương tiện của mình thông qua các thông tin như: vị trí hiện tại của xe, lịch sử di chuyển của xe.

Với kiến thức còn hạn chế, tuy đã cố gắng nỗ lực hết mình song đề tài của em vẫn không thể tránh khỏi những thiếu sót. Em mong nhận được những đóng góp về thiếu sót trong đề tài của em từ các thầy, cô và các bạn để đề tài có thể được hoàn thiện hơn nữa.

*Em xin chân thành cảm ơn!!!*

# **Chương 1: TỔNG QUAN**

## 1. Các mạch nhúng

Phần này giới thiệu về các mạch sử dụng trong hệ thống bao gồm: mạch Arduino và Module Sim808. Với mỗi mạch đều được giới thiệu về các thông số kỹ thuật cơ bản và cách để chúng ta có thể giao tiếp với chúng.

### 1.1. Mạch Arduino

1.1.1 Lịch sử ra đời

+ Arduino được khởi động vào năm 2005 như là một dự án dành cho sinh viên trại Interaction Design Institute Ivrea (Viện thiết kế tương tác Ivrea) tại Ivrea, Italy. Vào thời điểm đó các sinh viên sử dụng một "BASIC Stamp" (con tem Cơ Bản) có giá khoảng $100, xem như giá dành cho sinh viên. Massimo Banzi, một trong những người sáng lập, giảng dạy tại Ivrea. Cái tên "Arduino" đến từ một quán bar tại Ivrea, nơi một vài nhà sáng lập của dự án này thường xuyên gặp mặt. Bản thân quán bar này có được lấy tên là Arduino, Bá tước của Ivrea, và là vua của Italy từ năm 1002 đến 1014.

+ Lý thuyết phần cứng được đóng góp bởi một sinh viên người Colombia tên là Hernando Barragan. Sau khi nền tảng Wiring hoàn thành, các nhà nghiên cứu đã làm việc với nhau để giúp nó nhẹ hơn, rẻ hơn, và khả dụng đối với cộng đồng mã nguồn mở. Trường này cuối cùng bị đóng cửa, vì vậy các nhà nghiên cứu, một trong số đó là David Cuarlielles, đã phổ biến ý tưởng này.

1.1.2. Các broad chính

+ Phần cứng Arduino gốc được sản xuất bởi công ty Italy tên là Smart Projects. Một vài board dẫn xuất từ Arduino cũng được thiết kế bởi công ty của Mỹ tên là SparkFun Electronics.

+ Sáu phiên bản phần cứng của Arduino cũng đã được sản xuất thương mại tính đến thời điểm hiện tại.

+ Các broad Arduino mẫu:

- Arduino Diecimila in Stoicheia

- Arduino Duemilanove (rev 2009b)

- Arduino UNO

- Arduino Leonardo

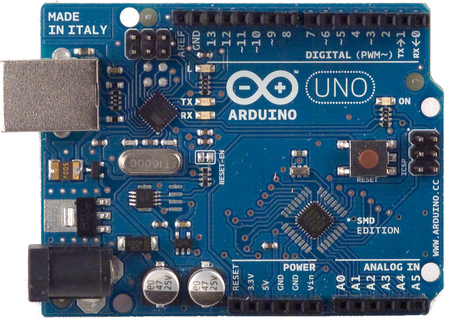
- Arduino Mega

- Arduino MEGA 2560 R3

- Arduino Nano

- Arduino Due (nền tảng ARM)

1.1.3. Arduino Uno R3



*Hình 1.1. Mạch Arduino Uno R3*

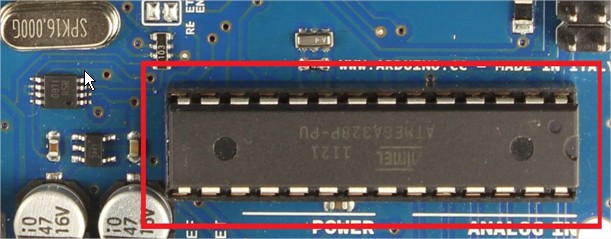
Nhắc tới dòng mạch Arduino dùng để lập trình, cái đầu tiên mà người ta thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (R3).

1.1.3.1. Thông số cơ bản

*Bảng 1.1. Các thông số cơ bản của mạch Arduino Uno R3*

|  |  |
| --- | --- |
| Vi điều khiển | ATmega328 họ 8bit |
| Điện áp hoạt động | 5VDC (Chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16MHz |
| Dòng tiêu thụ | Khoảng 30mA |
| Điện áp vào khuyến dùng | 7 – 12VDC |
| Điện áp vào giới hạn | 6 – 20VDC |
| Số chân Digital (I/O) | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30mA |
| Dòng ra tối đa (5V) | 500mA |
| Dòng ra tối đa (3.3V) | 50mA |
| Bộ nhớ flash | 32KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |

1.1.3.2. Vi điều khiển



*Hình 1.2. Vi điều khiển của mạch Arduino Uno R3*

Vi điều khiển được sử dụng ở đây là ATmega328. Đây là bộ não của Arduino UNO R3, nó có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa …

1.1.3.3. Năng lượng

+ Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lí nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

+ Các chân năng lượng:

- GND: cực âm của nguồn điện cấp cho Arduino UNO.

- 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.

- 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.

- Vin: nối cực dương của nguồn cấp với chân này để cấp nguồn cho Arduino UNO.

- IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này.

- RESET: dùng để reset vi điều khiển.

***\* Một vài lưu ý khi cấp nguồn cho Arduino UNO:***

- Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó phải hết sức cẩn thận kiểm tra cực âm – dương trước khi cấp nguồn cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ khiến nó bị hỏng, không thể dùng được nữa.

- Các chân 3.3V và 5V dùng để cấp nguồn cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng mạch.

- Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng mạch.

- Cấp điện áp trên 13V vào chân RESET trên mạch có thể làm hỏng vi điều khiển ATmega328.

- Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

- Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.

- Cường độ dòng điện qua một chân Digital hoặc Analog bất kỳ của Arduino UNO nếu vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó, nếu không để truyền/nhận dữ liệu, ta phải mắc một điện trở hạn dòng.

1.1.3.4. Bộ nhớ

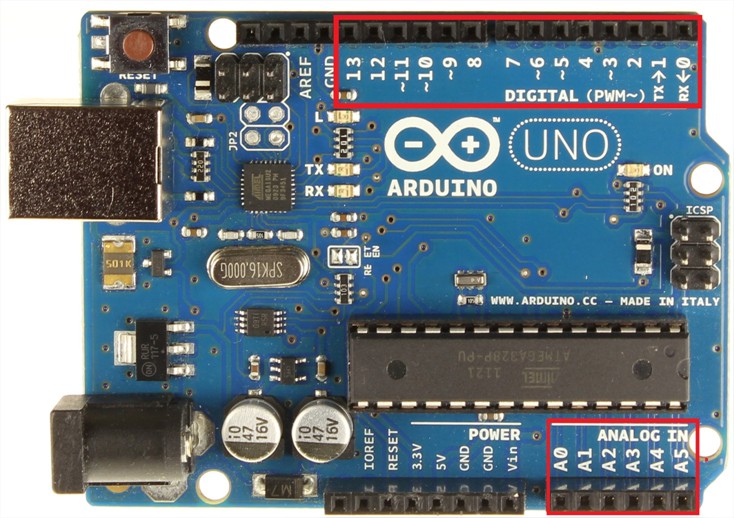
Vi điều khiển Atmega328 tiêu chuẩn cung cấp cho người dùng:

+ 32KB bộ nhớ Flash: nơi lưu trữ những đoạn lệnh. Thường thì sẽ có khoảng vài KB trong số 32KB được dùng cho bootloader.

+ 2KB cho SRAM: giá trị các biến được khai báo khi lập trình sẽ lưu ở đây. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

+ 1KB cho EEPROM: là nơi dùng để đọc/ghi dữ liệu mà không lo bị mất đi khi mất điện.

1.1.3.5. Các cổng vào ra



*Hình 1.3. Các cổng vào ra của mạch Arduino Uno R3*

+ Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có hai mức điện áp là 0v và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up từ được cài đặt ngay trong vi điều khiển ATmega328. Một số chân digital có chức năng đặc biệt như sau:

- 2 chân Serial: 0 (RX) và 1(TX) dùng để gửi (Transmit – TX) và nhận (Receive – RX) dữ liệu TTL Serial. Arduino UNO có thể giao tiếp với các thiết bị khác thông qua hai chân này.

- Chân PWM (gồm các chân: 3, 5, 6, 9, 10, 11): cho phép xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 🠢 28-1 tương ứng với 0V 🠢 5V) bằng hàm analogWrite(). Nói một cách đơn giản, điện áp ở chân này có thể được điều chỉnh từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

- Chân giao tiếp SPI (gồm các chân: 10 – SS, 11 – MOSI, 12 – MISO, 13 – SCK): ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

- LED 13: trên Arduino UNO có một đèn led màu cam (ký hiệu chữ L). Khi bấm nút Reset, đèn này sẽ nhấp nháy. Đèn led này được nối với chân 13. Khi chân này được sử dụng, đèn led sẽ sáng.

+ Arduino UNO có 6 chân analog cung cấp độ phân giải tín hiệu 10bit để đọc giá trị điện áp trong khoảng 0 – 5V. Đặc biệt, Arduino UNO có hai chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

1.1.4. Lập trình cho Arduino

+ Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “ngôn ngữ Arduino”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu.

+ Người dùng chỉ cần định nghĩa 2 hàm để tạo ra một chương trình vòng thực thi (cyclic executive) có thể chạy được:

- **setup():** hàm này chạy mỗi khi khởi động một chương trình, dùng để thiết lập các cài đặt

- **loop():** hàm này được gọi lặp lại cho đến khi tắt nguồn board mạch

+ Một chương trình điển hình cho một bộ vi điều khiển đơn giản chỉ là làm cho một bóng đèn Led sáng/tắt. Trong môi trường Arduino, ta sẽ phải viết một chương trình giống như sau:

#define LED\_PIN 13

void setup () {

pinMode (LED\_PIN, OUTPUT); *// Đặt chân 13 làm đầu ra digital*

}

void loop () {

digitalWrite (LED\_PIN, HIGH); *// Bật LED on*

delay (1000); *// chờ trong 1 giây (1000 mili giây)*

digitalWrite (LED\_PIN, LOW); *// Tắt LED off*

delay (1000); *// chờ trong 1s*

}

### 1.2. Module Sim808

1.2.1. Tổng quát

+ Module Sim 808 là Module GSM/GPS, được xây dựng dựa trên Sim808 của SIMCOM, hỗ trợ GSM/ GPRS với bốn băng tần và công nghệ định vị vệ tinh GPS. Ngoài hai chức năng chính GSM, GPS, Module Sim808 còn hỗ trợ thêm tính năng Bluetooth.

+ Module Sim808 có GPS với độ nhạy cao với 22 kênh theo dõi và 66 kênh tiếp nhận. Bên cạnh đó, nó cũng hỗ trợ công nghệ A-GPS, giúp cho việc định vị được chính xác hơn, ngay cả khi thiết bị ở trong nhà.

+ Đặc tính chung:

- Hoạt động ở 4 băng tần Quad-band 850/900/1800/1900MHz.

- Đã tích hợp module nguồn DC-DC, với nguồn đầu vào 9-24VDC cho đầu ra 4.3VDC cấp cho module sim.

- Hỗ trợ GPRS multi-slot class 12.

- Điều khiển bằng tập lệnh AT qua giao diện UART TTL.

- Tích hợp GPS/CNSS và hỗ trợ A-GPS.

- Dùng được cho cả MCU 3.3V và 5V.

1.2.2. Sơ đồ chân



*Hình 1.4. Mạch Sim808*

+ Vin MCU: Là chân input, dùng để tương thích mức điện áp giao tiếp UART giữa vi điều khiển và module. Nguồn của vi điều khiển là 3.3V hay 5V sẽ cấp tới chân Vin MCU.

+ Vin Sim: Là chân input, chân nguồn cấp cho Module Sim808. Với Module Sim 808 đã tích hợp nguồn DC-DC, chân Vin Sim không cần sử dụng.

+ STA: Là chân output, có thể dùng để đọc trạng thái của Module Sim808, để xem Module đã được khởi động hay chưa? Nếu tín hiệu đọc về từ chân STA ở mức cao thì Module đã được khởi động, còn ở mức thấp thì Module đang ngừng hoạt động.

+ PWK: Là chân input, dùng để bật hoặc tắt Module Sim808. Chân PWK được điều khiển từ mức thấp lên mức cao, với thời gian ở mức cao tối thiểu 1 giây thì Module Sim808 sẽ được bật hoặc tắt.

+ RST: Là chân input, dùng để khởi động lại Module Sim808. Để reset Module Sim808 xuất một xung từ mức thấp lên mức cao tới chân RST, với thời gian ở mức cao tối thiểu 105ms.

+ RXD: Là chân output, được nối với chân RXD của MCU.

+ TXD: Là chân input, được nối với chân TXD của MCU.

+ GND: 0VDC, được nối chung với GND của MCU.

## 2. Các công nghệ liên quan

Phần này giới thiệu các công nghệ được áp dụng trong hệ thống bao gồm: IoT, GPRS, GPS, Amazon Webservice, REST API, C++ REST SDK, ASP .NET Core, Android. Với mỗi công nghệ đều có phần giới thiệu về công nghệ và cách thức chúng tác động đến đề tài này.

### 2.1. Internet of things

Để có thể thiết kế lên một hệ thống áp dụng Internet of Things (IoT) thì việc tìm hiểu xem nó là cái gì, một hệ thống IoT cần những gì là điều rất quan trọng. Chính vì vậy mà ta sẽ đi tìm hiểu IoT trước tiên.

2.1.1. Internet of things là gì?

+ IoT là một liên mạng của các thiết bị vật lý, các loại xe cộ, các tòa nhà, và các đồ vật khác mà ở đó, mỗi vật thể đều được gắn với các thiết bị điện tử, các phần mềm, các cảm biến và khả năng kết nối mạng nhằm mục đích giúp cho các vật thể này có thể thu thập và trao đổi dữ liệu với nhau. Nói đơn giản IoT là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.

+ Một vật thể trong IoT có thể là một con người với một trái tim cấy ghép; một con vật ở trang trại với bộ chip sinh học; một chiếc xe với bộ cảm ứng tích hợp cảnh báo tài xế khi bánh xe xẹp hoặc bất kỳ vật thể tự nhiên hay nhân tạo nào mà có thể gán được một địa chỉ IP và cung cấp khả năng truyền dữ liệu thông qua mạng lưới.

+ Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng. Nếu mọi đối tượng, kể cả con người, được “đánh dấu” để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc kết nối thì có thể thực hiện qua wifi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, hồng ngoại…

2.1.2. Ứng dụng của IoT

+ IoT có ứng dụng rộng vô cùng, có thể kể ra một số thứ sau:

- Quản lý chất thải.

- Quản lý và lập kế hoạch quản lý đô thị.

- Quản lý môi trường.

- Mua sắm thông minh.

- Nhà thông minh.

- Quản lý các thiết bị cá nhân.

- Quản lý, giám sát các phương tiện.

+ Tác động của IOT rất đa dạng, trên các lĩnh vực: quản lý hạ tầng, y tế, xây dựng và tự động hóa, giao thông…. Cụ thể trong lĩnh vực y tế, Thiết bị IoT có thể được sử dụng để cho phép theo dõi sức khỏe từ xa và hệ thống thông báo khẩn cấp. Các thiết bị theo dõi sức khỏe có thể dao động từ huyết áp và nhịp tim màn với các thiết bị tiên tiến có khả năng giám sát cấy ghép đặc biệt, chẳng hạn như máy điều hòa nhịp hoặc trợ thính tiên tiến.

### 2.2. General Packet Radio Service

Một hệ thống IoT thì không thể thiếu được Internet. Không có Internet thì hệ thống không thể hoạt động. Và General Packet Radio Service chính là lựa chọn của chúng tôi dùng cho việc truyền dữ liệu từ mạch nhúng tới server.

2.2.1. General Packet Radio Service là gì?

+ GPRS là một dịch vụ dữ liệu di động dạng gói dành cho những người dùng Hệ thống thông tin di động toàn cầu (GSM) và điện thoại di động IS-136.

+ Nó cung cấp dữ liệu ở tốc độ từ 56 đến 114 kbps.

+ GPRS có thể được dùng cho những dịch vụ như truy cập Giao thức Ứng dụng Không dây (WAP), Dịch vụ tin nhắn ngắn (SMS), Dịch vụ nhắn tin đa phương tiện (MMS), và với các dịch vụ liên lạc Internet như email và truy cập World Wide Web.

+ Các hệ thống di động 2G kết hợp với GPRS thường được gọi là **"2.5G"**, có nghĩa là, một công nghệ trung gian giữa thế hệ điện thoại di động thứ hai (2G) và thứ ba (3G). Nó cung cấp tốc độ truyền tải dữ liệu vừa phải, bằng cách sử dụng các kênh đa truy cập theo phân chia thời gian (TDMA) đang còn trống, ví dụ hệ thống GSM.

2.2.2. Phân loại các thiết bị sử dụng GPRS

Các thiết bị được sử dụng GPRS được chia thành 3 loại:

+ **Loại A:** Có thể kết nối vào dịch vụ GPRS và dịch vụ GSM (thoại, SMS) cùng lúc.

**+ Loại B:** Có thể kết nối vào dịch vụ GPRS và dịch vụ GSM (thoại, SMS) nhưng chỉ dùng một trong hai dịch vụ vào một thời điểm.

**+ Loại C:** Được kết nối với hoặc dịch vụ GPRS hoặc dịch vụ GSM (thoại, SMS). Việc chuyển đổi giữa hai dịch vụ phải thực hiện thủ công.

2.2.3. Lý do lựa chọn GPRS

Có nhiều lựa chọn mạng không dây khác tiêu biểu như sử dụng wifi nhưng chúng tôi chọn GPRS là bởi vì mạng GPRS tuy tốc độ không cao nhưng độ phủ sóng của nó rộng khắp toàn quốc. Hiện tại ở Việt Nam, mạng wifi vẫn chỉ đang dừng lại ở việc lắp đặt trong nhà chứ chưa được phủ sóng rộng khắp toàn quốc như GPRS. Thêm nữa, việc truyền tải dữ liệu giữa mạch nhúng và server không đòi hỏi phải có kết nối mạng tốc độ cao. Chính vì thế mà chúng tôi đã lựa chọn GPRS.

### 2.3. Global Positioning System

Phần này giới thiệu về global positioning system (GPS), cơ chế hoạt động của nó và cũng như vai trò của nó đối với hệ thống IoT này.

2.3.1. Global Positioning System là gì?

+ **GPS** thực chất là một mạng lưới bao gồm 27 vệ tinh quay xung quanh trái đất. Trong số 27 vệ tinh này, 24 vệ tinh đang hoạt động, 3 vệ tinh còn lại đóng vai trò dự phòng trong trường hợp 1 trong số 24 vệ tinh chính bị hư hỏng.

+ Dựa vào cách sắp đặt của các vệ tinh này, khi đứng dưới mặt đất, bạn có thể nhìn được ít nhất là 4 vệ tinh trên bầu trời tại bất kì thời điểm nào.

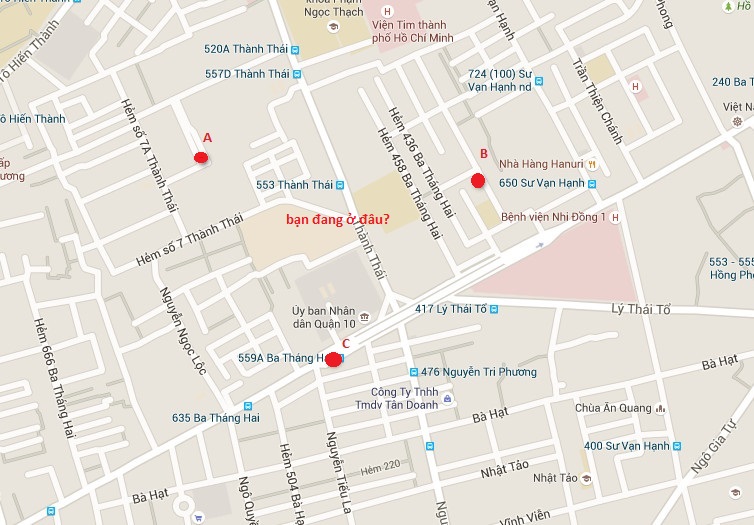
2.3.2. Cơ chế hoạt động của GPS

+ Các vệ tinh **GPS**bay hai vòng trong một ngày theo một quỹ đạo đã được tính toán chính xác và liên tục phát các tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận các tín hiệu này và giải mã bằng các phép tính lượng giác, qua đó sẽ tính toán và hiển thị được vị trí của người dùng.

+ Các đầu thu GPS (các thiết bị smartphone…) thu dữ liệu từ các vệ tinh GPS ở trên bầu trời. Nói một cách đơn giản, mỗi vệ tinh cho bạn biết khoảng cách chính xác từ vị trí của bạn đến vệ tinh đó hoặc một điểm nào đó trên trái đất

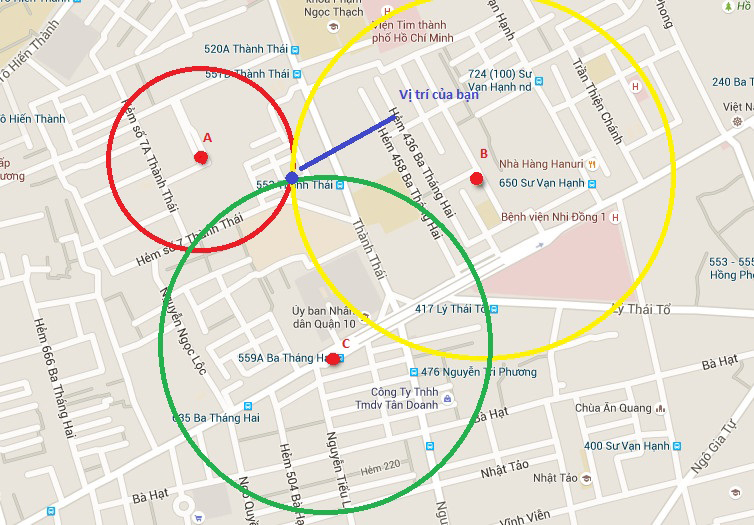
+ Cơ chế hoạt động của GPS rất đơn giản, bạn có thể tưởng tượng như sau:

- Trên bản đồ có 3 điểm cố định A, B, C. Dữ liệu GPScho bạn biết khoảng cách lần lượt từ điểm A, B, C đến nơi bạn đứng là 1km, 3km, 2km.



*Hình 1.5*

- Sau đó bạn vẽ 3 vòng tròn có tâm là A, B, C với bán kính lần lượt là 1km, 3km và 2km.



*Hình 1.6*

- Vị trí giao nhau của ba vòng tròn chính là vị trí của bạn.

+ Thiết bị thu phải nhận được tín hiệu của ít nhất ba vệ tinh để cho ra vị trí hai chiều và để theo dõi được chuyển động của bạn. Khi nhận được tín hiệu của ít nhất bốn vệ tinh, máy sẽ cho ra được vị trí ba chiều. Một khi vị trí của bạn đã tính được thì thiết bị thu có thể tính các thông tin khác, như tốc độ di chuyển, hướng chuyển động, bám sát di chuyển, khoảng hành trình, khoảng cách đích đến và nhiều thứ khác nữa.

+ Để đưa ra vị trí chính xác, rất nhiều thiết bị GPS kết nối tới ít nhất là 4 vệ tinh. Đó là lý do vì sao đôi khi để tìm ra vị trí chính xác của bạn, hệ thống GPS lại mất nhiều thời gian tới vậy. Đó cũng là lý do vì sao đôi khi bạn bị mất sóng GPS: thiết bị di động của bạn có thể đã kết nối tới 1 hoặc 2 vệ tinh, song 2 vệ tinh vẫn là không đủ.

2.3.3. Độ chính xác của GPS

+ Các máy thu GPS ngày nay cực kì chính xác, nhờ vào thiết kế nhiều kênh hoạt động song song của chúng. Các máy thu 12 kênh song song (của Garmin) nhanh chóng khóa vào các quả vệ tinh khi mới bật lên và duy trì kết nối bền vững.

+ Trạng thái của khí quyển và các nguồn gây sai số khác có thể ảnh hưởng tới độ chính xác của máy thu GPS. Các máy thu GPS có độ chính xác trung bình trong vòng 15 mét.

+ Các máy thu mới hơn với khả năng WAAS (*Wide Area Augmentation System*) có thể tăng độ chính xác trung bình tới dưới 3 mét. Không cần thêm thiết bị hay mất phí để có được lợi điểm của WAAS bằng các máy phát hiệu. Để thu được tín hiệu đã sửa lỗi, người dùng phải có máy thu tín hiệu vi sai bao gồm cả ăn-ten để dùng với máy thu GPS của họ.

2.3.4. Ứng dụng của GPS

GPS đã được ứng dụng rất nhiều trong đời sống sản xuất. Có thể kể đến một số ứng dụng như:

+ Ứng dụng trong dân dụng: giám sát quản lý vận tải; theo dõi vị trí, tốc độ, hướng di chuyển…

+ Ứng dụng trong quân sự: bom thông minh JDAM, tên lửa không đối đất, tên lửa tấn công đất liền, tên lửa hành trình, tên lửa đất đối đất…

2.3.5. Vai trò của GPS đối với hệ thống

GPS đóng vai trò cực kỳ quan trọng trong hệ thống giám sát phương tiện. Nếu không có GPS thì hệ thống sẽ không hoạt động được vì lúc đấy, ta sẽ không xác định được vị trí của phương tiện nữa và điều này dẫn đến việc hệ thống sẽ bị lỗi.

2.4. Amazon Webservice

+ Amazon web services là một trong số các dịch vụ của Amazon hoạt động dựa trên nền tảng cloud computing (điện toán đám mây).

+ Tổng quan kiến trúc:

**- Global Infrastructure (cơ sở hạ tầng nằm trên toàn cầu).**

**- Các dịch vụ cơ sở – Lưu trữ: S3 (Simple Storage Service).**

**- Các dịch vụ cơ sở – Database.**

**- Các dịch vụ cơ sở – Mạng: VPC.**

**- Application Services (Các dịch vụ ứng dụng).**

+ Amazon Elastic Compute Cloud (Amazon EC2) là một dịch vụ IaaS (Infrastructure as a Service) trên hạ tầng cơ sở của Amazon. Có nghĩa là bạn có thể tạo hoặc tạm dừng ra một máy chủ ảo (Virtual Machine) với các cấu hình thông số phần cứng có thể thay đổi rất nhanh với giao diện trực quan, đặc biệt là chi phí giá thành khá rẻ với dịch vụ on-demand tức là dùng bao nhiêu tính tiền bấy nhiêu. So với việc bạn tự xây dựng infrastructure, thì việc sử dụng EC2 sẽ rất tiện lợi và tiết kiệm chi phí và nhân sự rất nhiều.

### 2.5. REST API

+ REST là viết tắt của Representational State Tranfer. Giải thích đơn giản, REST là một loạt hướng dẫn và dạng cấu trúc dùng cho việc chuyển đổi dữ liệu. Thông thường, REST hay được dùng cho ứng dụng web, nhưng cũng có thể làm việc được với dữ liệu phần mềm.

+ Trong thực tế, REST đã có ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều.

+ REST là một bộ quy tắc để tạo ra một ứng dụng Web Service, mà nó tuân thủ 4 nguyên tắc:

- Sử dụng các phương thức HTTP một cách rõ ràng. REST đặt ra một quy tắc đòi hỏi lập trình viên xác định rõ ý định của mình thông qua các giao thức của HTTP:

* GET để truy vấn object
* POST để tạo object mới
* PUT để sửa đổi hoặc thay thế một object
* DELETE để loại bỏ một object

Các nguyên tắc ở trên là không bắt buộc. Tuy nhiên, REST đưa ra các nguyên tắc ở trên với mục đích đưa mọi thứ trở lên rõ ràng và dễ hiểu.

- Phi trạng thái

Có nghĩa là nó không lưu giữ thông tin của client. Chẳng hạn bạn vừa yêu cầu xem trang thứ 2 của tài liệu, và bây giờ bạn muốn xem trang tiếp theo thì REST sẽ không lưu trữ rằng trước đó bạn đã xem trang thứ 2.

Như vậy, các thành phần máy chủ phi trạng thái ít phức tạp hơn để thiết kế.

- Đưa ra cấu trúc thư mục giống như URI

REST đưa ra một cấu trúc để người dùng có thể truy cập vào tài nguyên của nó thông qua các URL

- Truyền tải XML, JSON hoặc cả hai

Khi Client gửi một yêu cầu tới web service nó thường được truyền tải dưới dạng XML hoặc JSON và thông thường nhận về với hình thức tương tự.

### 2.6. C++ REST SDK

+ C++ REST SDK là một dự án của Microsoft dùng cho việc giao tiếp giữa client-server dựa trên cloud trong native code sử dụng một thiết kế API C++ bất đồng bộ hiện đại. Project này nhằm giúp các lập trình viên C++ kết nối và tương tác với các dịch vụ.

+ Đặc điểm:

- Hỗ trợ HTTP Client/Server.

- Hỗ trợ kiểu dữ liệu JSON, URI.

- Hỗ trợ Web Socket Client, Oauth.

- PPL Tasks – không đồng bộ dựa trên C++11.

- Chạy được trên Windows desktop, Window Store, Windows Phone, Ubuntu, OS X, iOS và Android.

+ C++ REST SDK giúp việc xây dựng 1 REST API bằng ngôn ngữ C++ trở nên đơn giản, không yêu cầu người sử dụng phải biết rõ về HTTP.

### 2.7. ASP .NET Core

+ Giới thiệu:

- ASP.NET Core là một mã nguồn mở và là nền móng mới cho xây dựng cloud trên internet kết nối các ứng dụng web, IoT và mobile backend. ASP.NET Core có khả năng chạy trên .NET Core hoặc trên .NET framework.

- ASP.NET Core có khả năng phát triển cà chạy trên nhiều môi trường khác nhau như Windows, MacOS và Linux.

- ASP.NET là mã nguồn mở.

+ Tại sao nên sử dụng ASP.NET Core

- ASP.NET đã trở nên rất phổ biến trong việc xây dựng và phát triển các ứng dụng web.

- ASP.NET Core có một số thay đổi kiến trúc làm cho nó gọn nhẹ hơn. ASP.NET Core không còn dựa trên System.Web mà dựa trên tập hợp các granular và các NuGet. Nhờ thế cho phép bạn tối ưu hóa các ứng dụng của bạn nhờ chỉ cần sử dụng các gói NuGet mà bạn cần.

- Ngoài ra, nó bảo mật tốt hơn, cả thiện hiệu suất và giảm chi phí nhờ việc bạn chỉ cần trả cho những gì bạn sử dụng.

+ Đặc điểm

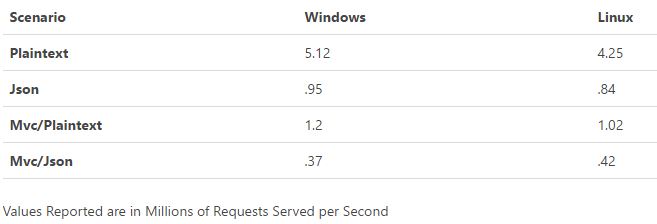
- Hỗ trợ đa nền tảng: Có thể phát triển và chạy ứng dụng web trên nhiều nền tảng khác nhau như Linux, Windows, MacOS.

- Mã nguồn mở

Mã nguồn của ASP.NET Core đã có sẵn trên Github. Bạn có thể tải và thay đổi bất kì đoạn code nào mà bạn chưa ưng ý.

Bạn cũng có thể đóng góp bất kì một cải tiến nào cho ASP.NET Core.

- Hiệu suất được cải thiện: Microsoft giới thiệu máy chủ mới đi kèm với ASP.NET Core là web Kestrel. Đây là máy chủ .NET có hiệu năng tốt nhất tính đến thời điểm hiện tại



*Hình 1.7*

- Được xây dựng trên Dependency Injection: Việc này giúp cho việc xây dựng ứng dụng web ASP.NET Core trở nên đơn giản, rõ ràng và dễ bảo trì

- Hợp nhất Web UI và Web API

Microsoft đã đưa tất cả các framework vào 1 framework duy nhất, nhẹ hơn nhưng có đầy đủ tính năng của MVC và Web API.

Với việc hợp nhất này, bạn không phải cân nhắc định tuyến khác nhau cho một API Controller so với MVC Controller.

- MVC Helper Tag:

Với ASP.NET Core, Microsoft đã giới thiệu tag helper để tạo ra mã phía Client từ .NET và làm cho nó dễ dàng sử dụng hơn để tái sử dụng trong Razor Markup.

Nó được tham chiếu trong đánh dấu phía máy chủ như thế nào là 1 tag HTML. Công cụ Razor sẽ nhận ra và thực thi các đoạn mã .NET có liên quan tương ứng với nó.

### 2.8. Android

Phần này sẽ giới thiệu về hệ điều hành Android, lý do lựa chọn Android.

2.8.1. Android là gì?

Android là một hệ điều hành di động được phát triển bởi Google dựa trên Linux. Android được thiết kế chủ yếu cho các thiết bị cảm ứng như điện thoại thông minh và máy tính bảng.

2.8.2. Nền tảng Android

+ Với khả năng rộng rãi của Android, sẽ rất dễ dàng nhầm lẫn nó với một hệ điều hành máy tính để bàn. Android là một môi trường phân tầng, xây dựng trên nền của hệ điều hành nhân Linux, và nó bao hàm nhiều chức năng phong phú. Hệ thống con giao diện người sử dụng gồm có:

- Cửa sổ.

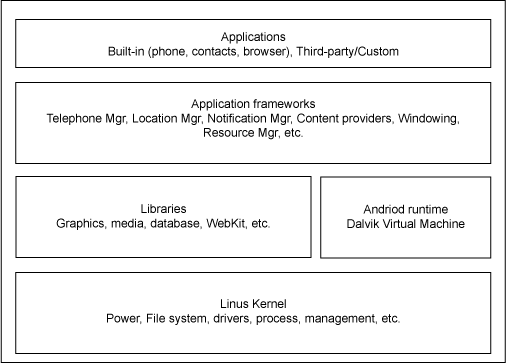
- Khung nhìn.

- Các tiểu trình để hiển thị các phẩn tử phổ biến như các hộp biên soạn, danh sách, danh sách thả xuống.

+ Android gồm một trình duyệt có thể nhúng vào được, được xây dựng dựa trên WebKit, chính là máy (engine) trình duyệt mã nguồn mở cũng đang là động cơ của trình duyệt Safari di động của iPhone.

+ Android khoe ra một mảng dồi dào nhiều tùy chọn kết nối, gồm WiFi, Bluetooth, và dữ liệu không dây qua một kết nối di động (như GPRS, EDGE, và 3G). Một kỹ thuật phổ biến trong các ứng dụng Android là liên kết tới Google Maps để hiển thị một địa chỉ trực tiếp trong một ứng dụng. Việc hỗ trợ cho các dịch vụ dựa trên địa điểm (chẳng hạn như GPS) và các dụng cụ đo gia tốc cũng có sẵn trong chồng phần mềm Android, mặc dù không phải tất cả các thiết bị Android được trang bị với các phần cứng cần thiết. Cũng có cả hỗ trợ máy ảnh (camera) nữa.

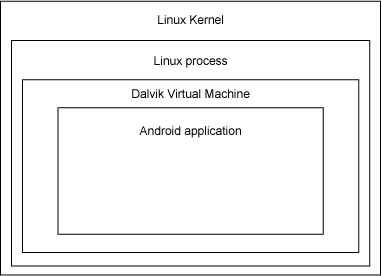
+ Về mặt lịch sử, có hai lĩnh vực mà các ứng dụng di động đã gắng sức theo kịp các đối tác máy tính để bàn là đồ họa/đa phương tiện, và phương thức lưu trữ dữ liệu. Android giải quyết thách thức về đồ họa bằng sự hỗ trợ dựng sẵn cho đồ họa 2-D và 3-D, gồm cả thư viện OpenGL. Gánh nặng lưu trữ dữ liệu được giảm nhẹ vì nền Android có gồm cả cơ sở dữ liệu SQLite mã nguồn mở phổ biến. Hình bên dưới cho thấy một khung nhìn đơn giản hoá về các tầng phần mềm Android.



*Hình 1.8. Các tầng phần mềm Android*

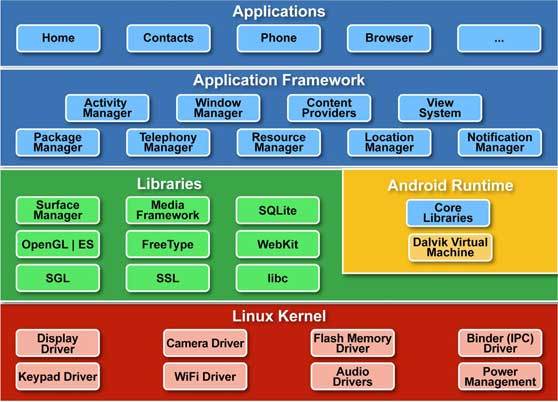
2.8.3. Kiến trúc ứng dụng

+ Như đã đề cập, Android chạy bên trên một hệ điều hành nhân Linux. Các ứng dụng Android được viết bằng ngôn ngữ lập trình Java, và chúng chạy trong một máy ảo (VM). Điều quan trọng là phải lưu ý rằng VM không phải là một máy ảo Java (JVM) như bạn chờ đợi, mà là máy ảo Dalvik (Dalvik Virtual Machine), một công nghệ mã nguồn mở. Mỗi ứng dụng Android chạy trong một cá thể của máy ảo Dalvik, đến lượt mình, cá thể máy ảo này nằm trong một tiến trình do nhân Linux quản lý.



*Hình 1.9. Máy ảo Dalvik*

+ Hệ điều hành Android là một ngăn xếp của các thành phần phần mềm mà có thể đại khái phân chia thành 5 khu vực và 4 lớp chính. Hình dưới đây minh họa sơ đồ cấu trúc Android.



*Hình 1.10. Sơ đồ cấu trúc Android*

- Lớp Linux Kernel trong Android: Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, … Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

- Libraries trong Android: Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.

- Android Libraries: Phần này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn.

- Android Runtime: Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là Dalvik Virtual Machine là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android. Máy ảo Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó. Android Runtime cũng cung cấp bộ thư viện cốt lõi, cho phép các lập trình viên Android sử dụng để viết các ứng dụng Android.

- Application Framework: Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

2.8.4. Tại sao lại là Android?

+ Với một hệ thống IoT, thì giao diện người dùng cũng không kém phần quan trọng. Nó đóng vai trò giúp cho người dùng có thể tương tác được với hệ thống, giúp cho họ có thể điều khiển được mạch nhúng để từ đó điều khiển hệ thống hoạt động theo ý của mình. Và Android ở đây để giúp ta xử lý được vấn đề đó.

+ Lý do tiếp theo mà tôi muốn nhắc tới đó là vấn đề về người dùng. Hiện nay thì không hiếm khi bạn bắt gặp cảnh tượng một người đang đi đường và dùng điện thoại thông minh. Phải nói là điện thoại thông minh đang trở nên phổ cập với mọi người. Và hiện tại đang có hai hệ điều hành dành cho những chiếc điện thoại thông minh là: Android và iOs. Thế nhưng, không phải ai cũng có điều kiện để mua một chiếc iphone nên là số người sử dụng điện thoại sử dụng hệ điều hành Android là đông hơn. Thêm nữa, để viết một ứng dụng chạy trên hệ điều hành iOs thì ta buộc phải có một chiêc Macbook hoặc iMac. Điều này là hoàn toàn khó khăn với một sinh viên như tôi. Chính vì thế mà tôi đã chọn phát triển ứng dụng trên hệ điều hành Android trước.

# **Chương 2: MÔ TẢ HỆ THỐNG**

## 1. Nhiệm vụ cơ bản

Hệ thống được xây dựng nhằm mục đích giúp người dùng có thể giám sát các phương tiện của mình một cách dễ dàng, thuận lợi nhất. Thông qua hệ thống, người dùng có thể xem được vị trí hiện tại của phương tiện của mình, xem lịch sử di chuyển của phương tiện đó.

## 2. Cơ cấu tổ chức:

Hệ thống gồm hai bộ phận chính: bộ phận quản trị và bộ phận hỗ trợ.

+ Bộ phận quản trị: Quản lý thông tin của khách hàng, quản lý thông tin các thiết bị, phương tiện đã đăng ký với hệ thống.

+ Bộ phận hỗ trợ: Kiểm tra thông tin khách hàng, đăng ký dịch vụ cho khách hàng muốn sử dụng dịch vụ.

## 3. Quy trình làm việc của hệ thống

\* Đăng ký tài khoản:

+ Người dùng đến cửa hàng mua thiết bị và đăng ký xe sử dụng thiết bị với quản trị viên. Nhân viên bộ phận hỗ trợtiến hành xác nhận thông tin, nhập thông tin khách hàng, nhập thông tin xe của khách hàng và nạp chương trình cho thiết bị.

+ Thông tin người dùng đăng kí với hệ thống bao gồm: email, username, họ tên, địa chỉ, số điện thoại.

+ Quản trị viên sẽ tiến hành kích hoạt tài khoản sau khi việc đăng kí từ bộ phận hỗ trợ hoàn tất.

\* Đăng kí xe sử dụng thiết bị:

+ Nếu là người dùng mới, sau khi nhân viên bộ phận quản trị xác nhận và kích hoạt tài khoản, nhân viên bộ phận hỗ trợ thực hiện đăng kí thông tin xe của người dung.

+ Thông tin xe cần đăng kí bao gồm: biển số xe, loại xe, hãng xe, mô tả xe.

+ Sau khi đăng kí thông tin xe hoàn tất, bộ phận hỗ trợ sẽ tiến hành nạp chương trình vào thiết bị và giao cho người dùng.

+ Nếu là đã là người dùng của hệ thống, muốn đăng kí thêm xe sử dụng thiết bị, người dùng mang thiết bị ra cửa hàng, nhân viên bộ phận hỗ trợ sẽ tiến hành đăng kí.

\* Đăng ký lại xe sử dụng thiết bị:

+ Người dùng có thể đăng ký lại xe sử dụng một thiết bị giám sát bằng cách mang thiết bị ra cửa hàng và cung cấp đầy đủ thông tin cần thiết cho quản trị viên.

+ Thông tin cần thiết bao gồm: biển số xe, loại xe, hãng xe, mô tả (không bắt buộc). Bộ phận hỗ trợ sẽ tiến hành nhập lại biển số xe và nạp lại chương trình cho thiết bị giám sát.

\* Đăng nhập:

+ Sau khi có tài khoản được cấp bởi quản trị viên, người dùng tiến hành đăng nhập để sử dụng ứng dụng giám sát.

+ Với những người dùng lần đầu đăng nhập vào hệ thống, người dùng sẽ phải tiến hành đổi mật khẩu

+ Sau khi đăng nhập xong, người dùng có thể sử dụng ứng dụng để chọn xe và xem vị trí hiện tại cũng như lịch sử di chuyển của xe đó.

\* Xem vị trí hiện tại của thiết bị:

+ Người dùng chọn xe được giám sát, sau đó ấn nút xem vị trí hiện tại. Lúc này trên bản đồ sẽ hiện ra vị trí hiện tại của phương tiện.

\* Xem lịch sử di chuyển của phương tiện:

+ Sau khi đăng nhập thành công, người dùng có thể xem được lịch sử di chuyển của một phương tiện bất kỳ (phương tiện này phải thuộc trong danh sách các phương tiện được giám sát mà người dùng đã đăng ký với nhà sản xuất) theo ngày.

+ Với mỗi một khoảng thời gian, người dùng có thể xem chi tiết quãng đường mà xe đã đi.

\* Bật/ tắt chức năng ghi lại lịch sử di chuyển của phương tiện:

+ Người dùng có thể bật/tắt chức năng ghi lại lịch sử di chuyển ở trên ứng dụng giám sát phương tiện.

\* Quản lý tài khoản:

+ Sau khi bộ phận hỗ trợ hoàn tất nhập thông tin người dùng. Bộ phận quản trị sẽ kiểm tra lại thông tin và thực hiện kích hoạt tài khoản.

+ Tài khoản đã được đăng kí sẽ được lưu lại trong cơ sở dữ liệu của hệ thống.

+ Việc sửa thông tin tài khoản sẽ được thực hiện bởi bộ phận quản trị.

\* Quản lý xe đã đăng kí:

+ Xe của người dùng được đăng kí sẽ được lưu lại trong hệ thống cùng với mã thiết bị đã được gắn với xe.

+ Thực hiện sửa thông tin xe sẽ do quản trị viên chịu trách nhiệm.

\* Đổi mật khẩu:

+ Người dùng có thể tiến hành đổi mật khẩu khi sử dụng ứng dụng giám sát. Các thông tin cần có: mật khẩu cũ, mật khẩu mới.

## 4. Mẫu biểu

4.1. Mẫu biểu “Đăng ký sử dụng dịch vụ giám sát phương tiện”

**ĐĂNG KÝ DỊCH VỤ GIÁM SÁT PHƯƠNG TIỆN**

Ngày … Tháng … Năm ......

\* \* \*

*Thông tin đăng ký tài khoản:*

Họ và tên:

Địa chỉ:

Số điện thoại:

Email:

Username:

*Thông tin xe đăng kí sử dụng:*

Biển số xe:

Loại xe:

Hãng xe:

Người lập đơn

Kí tên

4.2. Mẫu biểu “Đăng ký xe mới sử dụng thiết bị”

**ĐĂNG KÝ XE MỚI SỬ DỤNG THIẾT BỊ**

Ngày … Tháng … Năm ......

\* \* \*

*Thông tin xe cũ*

Biển sỗ xe:

*Thông tin xe mới đăng kí sử dụng:*

Biển số xe:

Loại xe:

Hãng xe:

Người lập đơn

Kí tên

# **Chương 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG**

## 1. Phân tích chức năng nghiệp vụ

Phần này đưa ra các chức năng nghiệp vụ chính của hệ thống. Từ đó mô hình hóa hệ thống: đưa ra các sơ đồ luồng dữ liệu nghiệp vụ, biểu đồ hoạt động, tiến trình hoạt động, sơ đồ use case và sơ đồ tuần tự.

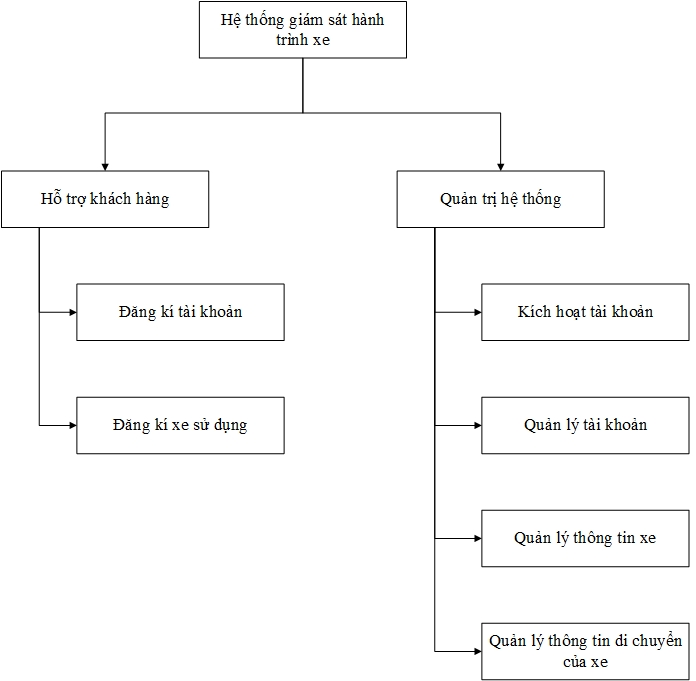
### 1.1. Mô hình hóa chức năng nghiệp vụ

\* Gom nhóm các chức năng chính của hệ thống:

*Bảng 3.1. Bảng gom nhóm các chức năng chính của hệ thống*

|  |  |  |
| --- | --- | --- |
| Đăng ký tài khoản | Hỗ trợ khách hàng | Hệ thống giám sát phương tiện |
| Đăng ký xe sử dụng |
| Kích hoạt tài khoản | Quản trị hệ thống |
| Quản lý tài khoản |
| Quản lý thông tin xe |
| Quản lý thông tin di chuyển của xe |

\* Sơ đồ phân rã chức năng:



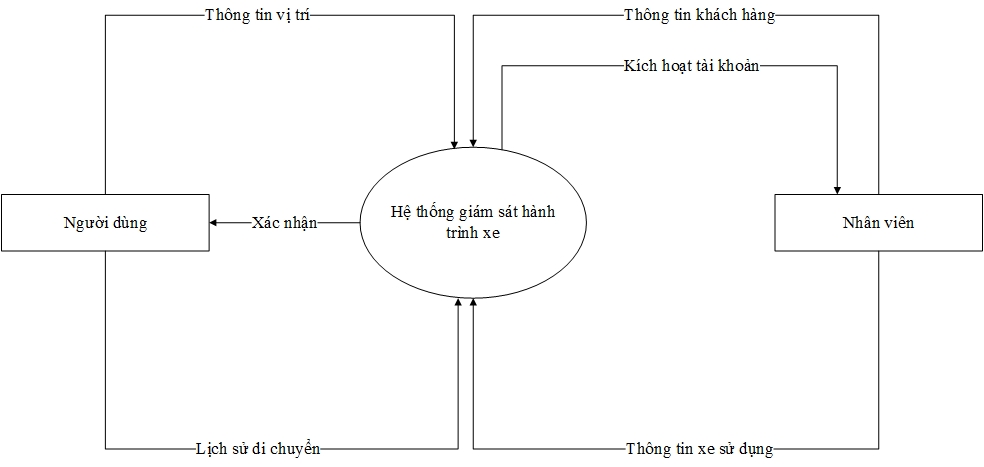
*Hình 3.1. Sơ đồ phân rã chức năng*

### 1.2. Mô hình hóa tiến trình nghiệp vụ:

1.2.1. Ký hiệu sử dụng

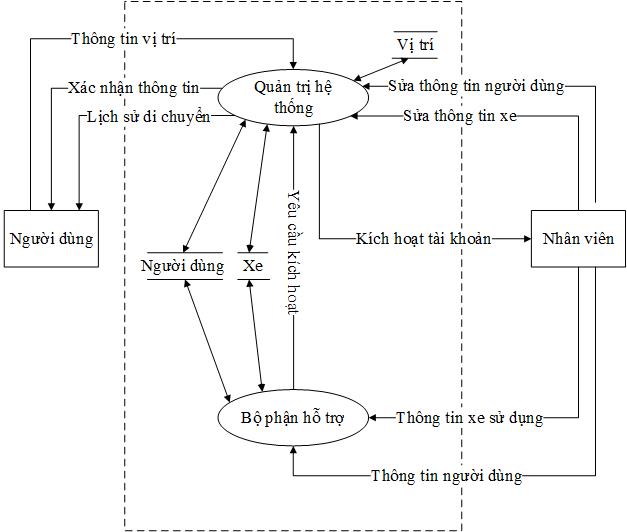
|  |  |
| --- | --- |
|  | Tiến trình |
|  | Tác nhân ngoài |
|  | Luồng dữ liệu |

1.2.2. Sơ đồ luồng dữ liệu mức khung cảnh



*Hình 3.2. Sơ đồ luồng dữ liệu mức khung cảnh*

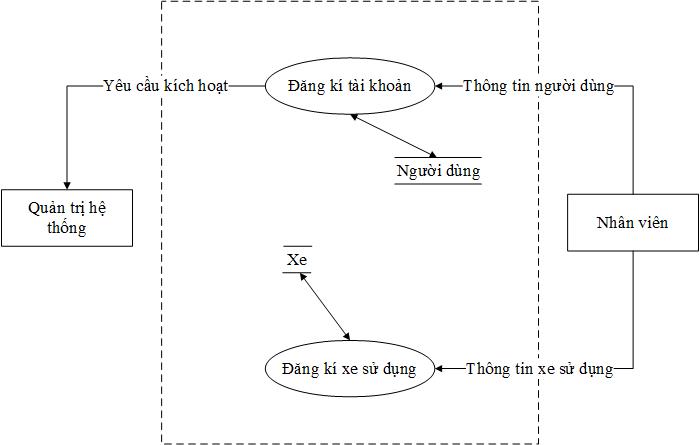
1.2.3. Sơ đồ luồng dữ liệu mức đỉnh



*Hình 3.3. Sơ đồ luồng dữ liệu mức đỉnh*

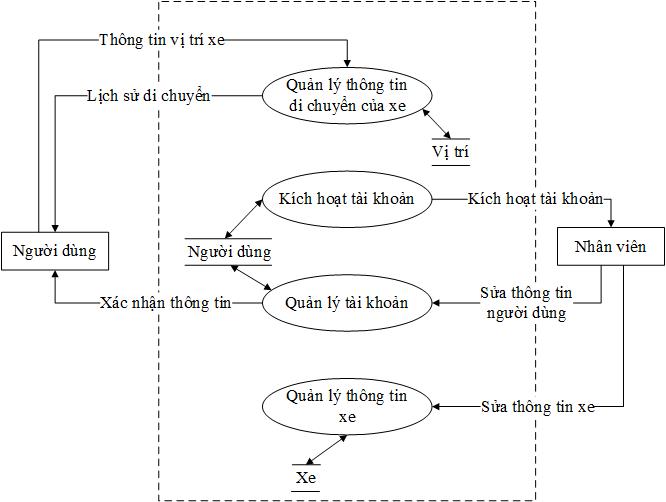
1.2.4. Sơ đồ luồng dữ liệu mức 2

1.2.4.1. “Hỗ trợ khách hàng”



*Hình 3.4. Sơ đồ luồng dữ liệu “Hỗ trợ khách hàng”*

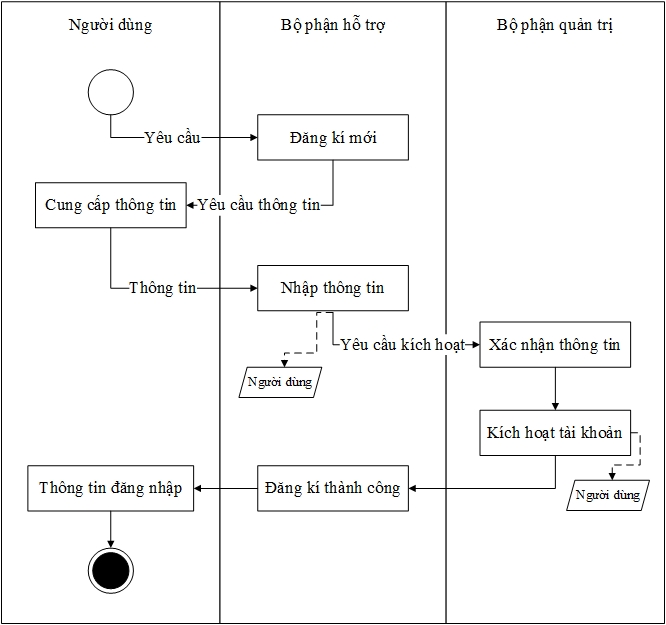
1.2.4.2. “Quản trị hệ thống”



*Hình 3.5. Sơ đồ luồng dữ liệu “Quản trị hệ thống”*

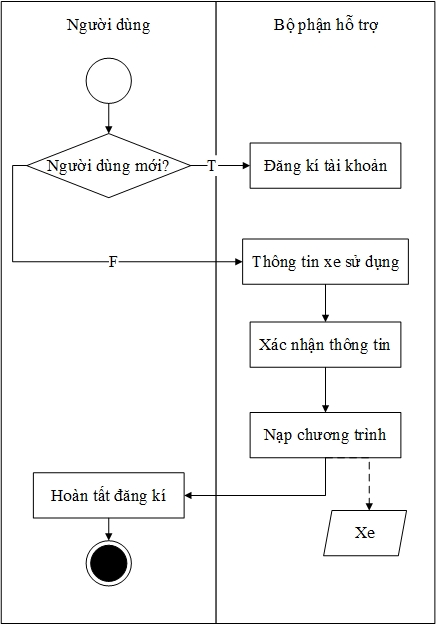
### 1.3. Biểu đồ hoạt động

1.3.1. “Đăng ký tài khoản”



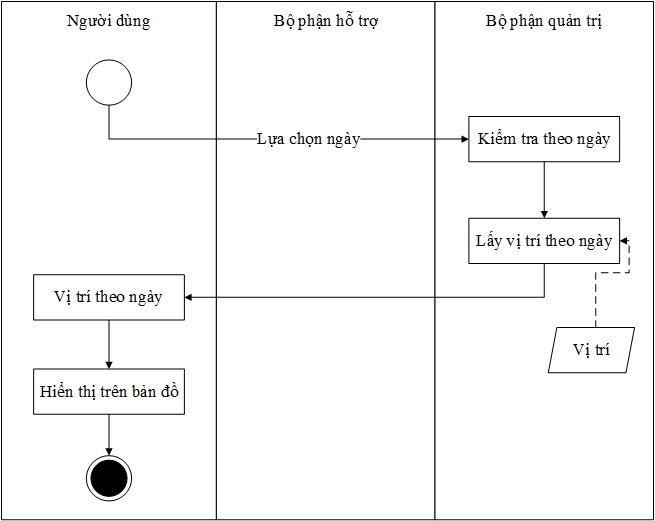
*Hình 3.6. Biểu đồ hoạt động “Đăng ký tài khoản”*

1.3.2. “Đăng ký xe”



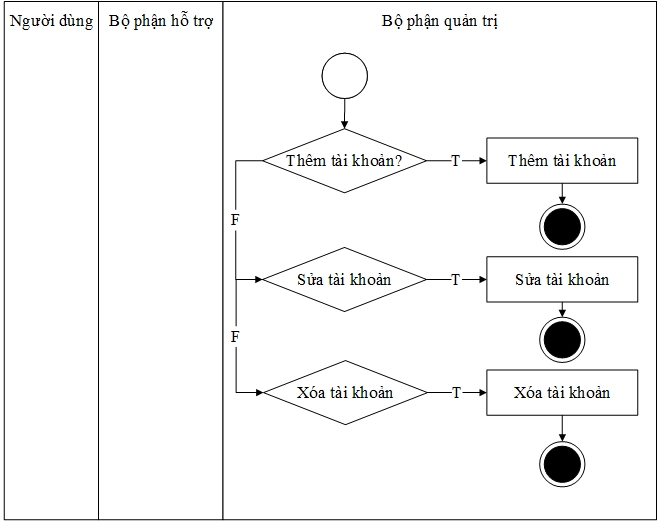
*Hình 3.7. Biểu đồ hoạt động “Đăng ký xe”*

1.3.3. “Xem lịch sử di chuyển”



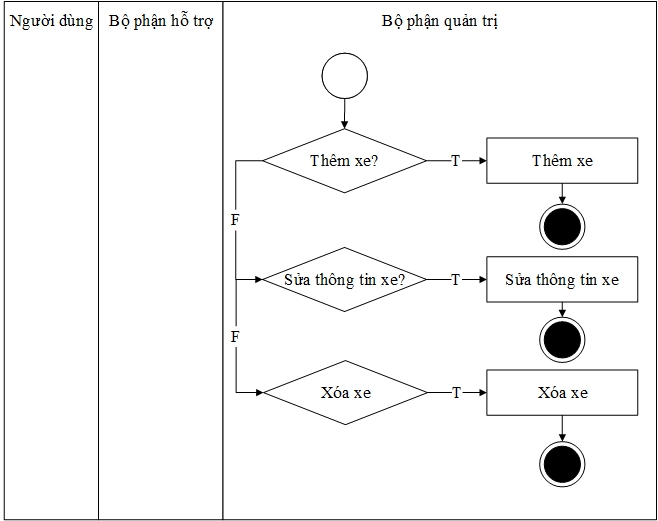
*Hình 3.8. Biểu đồ hoạt động “Xem lịch sử di chuyển”*

1.3.4. “Quản lý tài khoản”



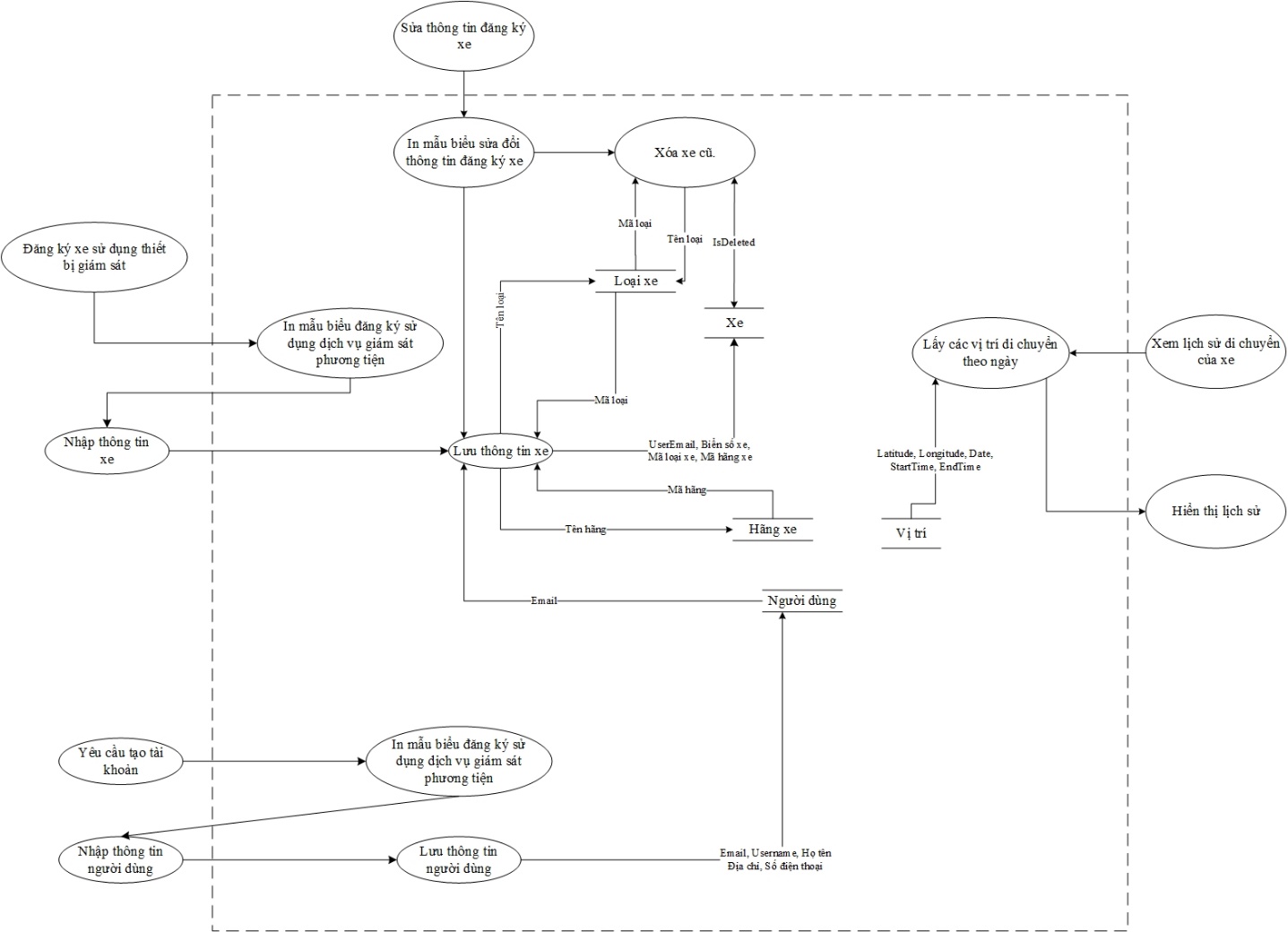
*Hình 3.9. Biểu đồ hoạt động “Quản lý tài khoản”*

1.3.5. “Quản lý xe”



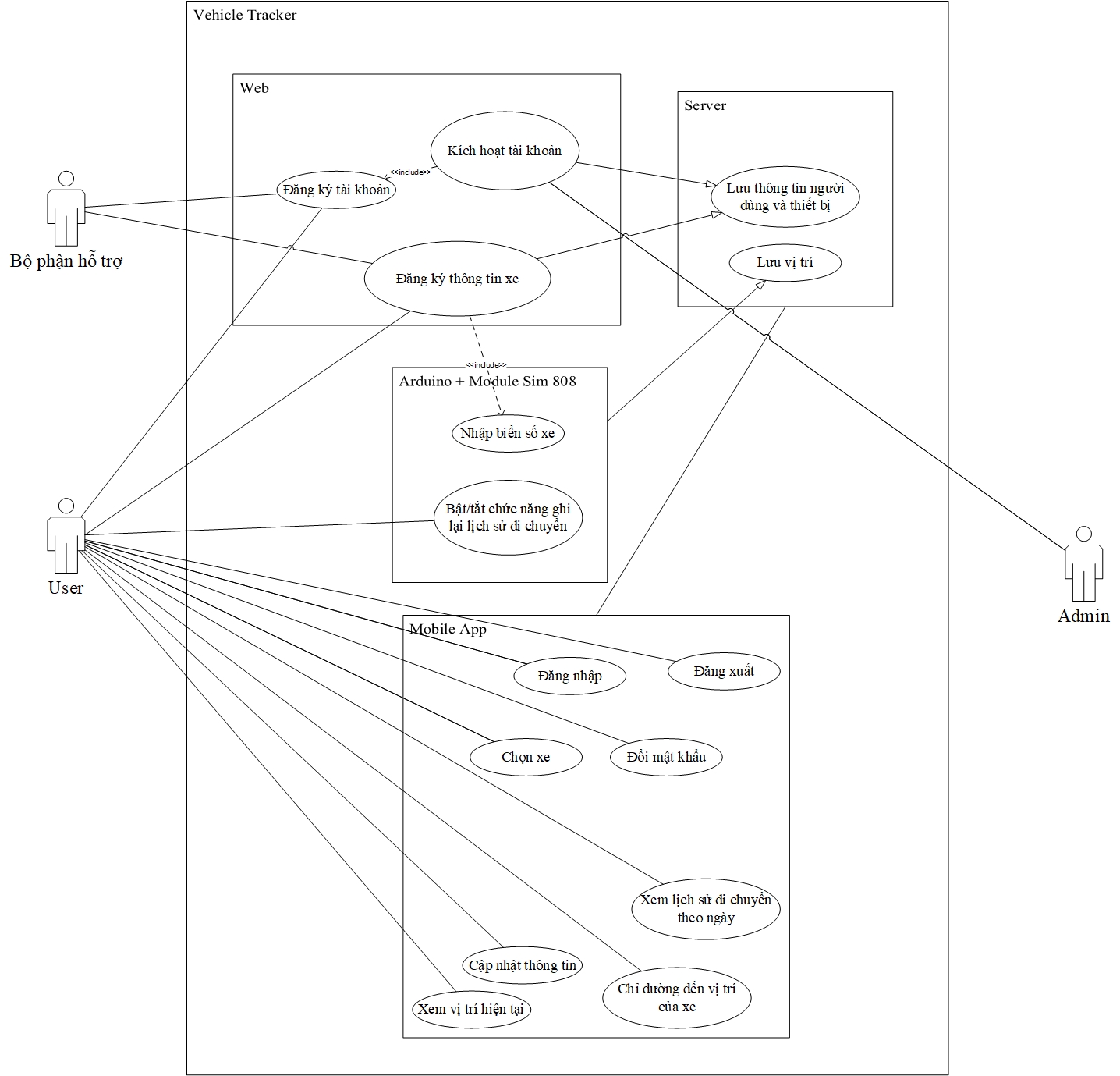
*Hình 3.10. Biểu đồ hoạt động “Quản lý xe”*

### 1.4. Thiết kế tiến trình hệ thống



*Hình 3.11. Tiến trình hệ thống*

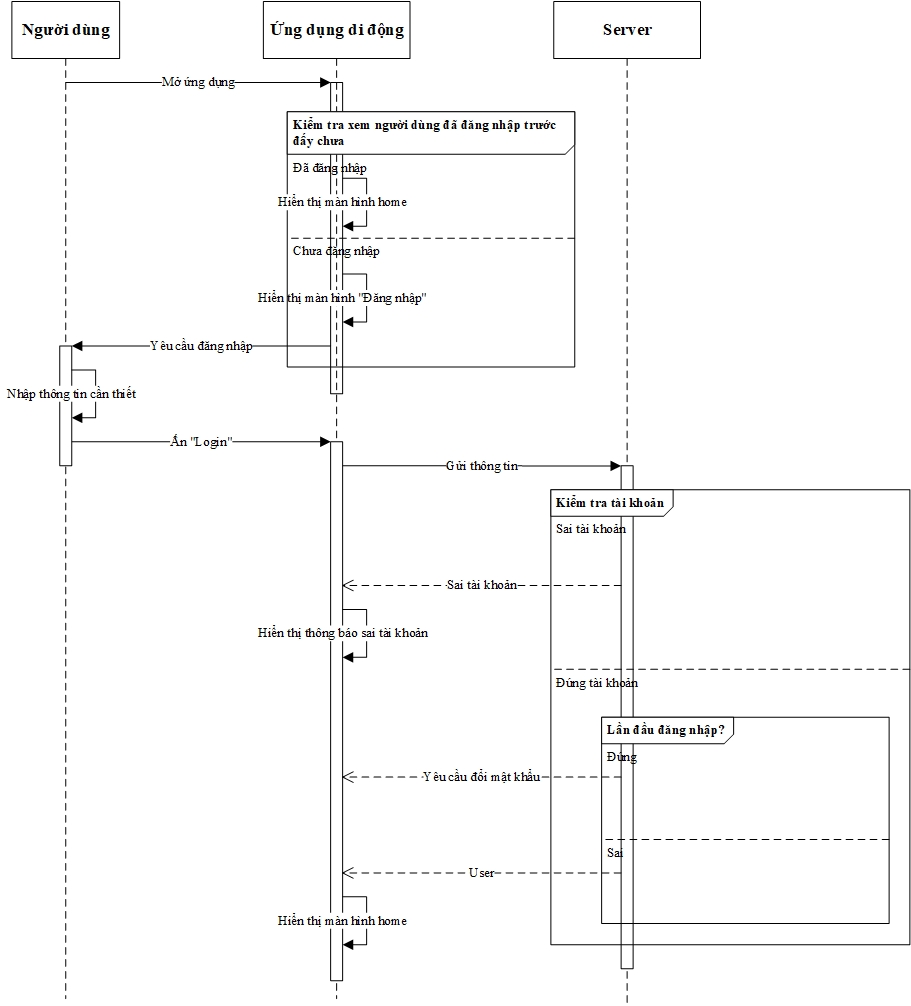
### 1.5. Sơ đồ use case



*Hình 3.12. Sơ đồ use case*

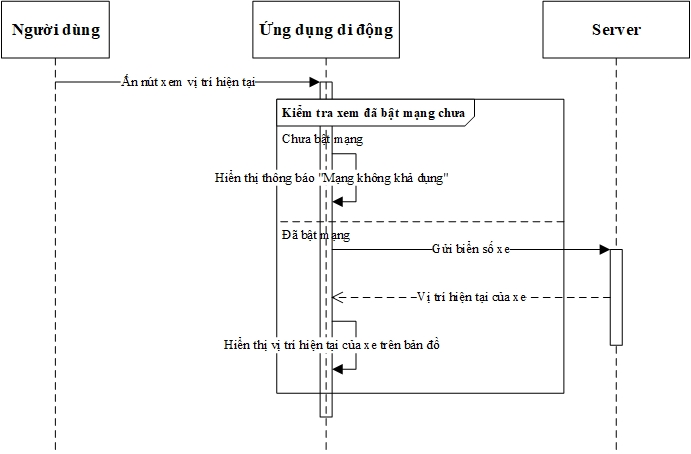
### 1.6. Sơ đồ tuần tự

1.6.1. “Đăng nhập”



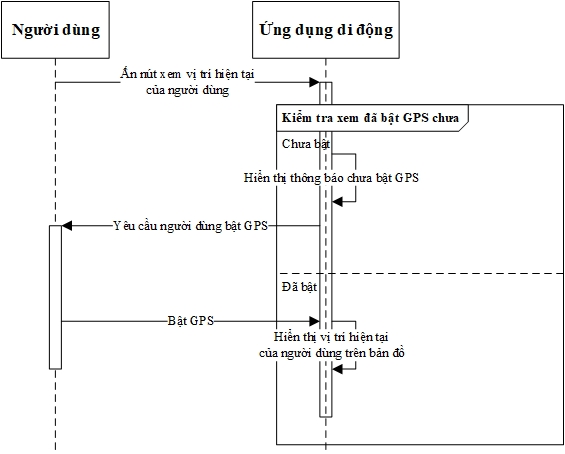
*Hình 3.13. Sơ đồ tuần tự “Đăng nhập”*

1.6.2. “Xem vị trí hiện tại của xe”



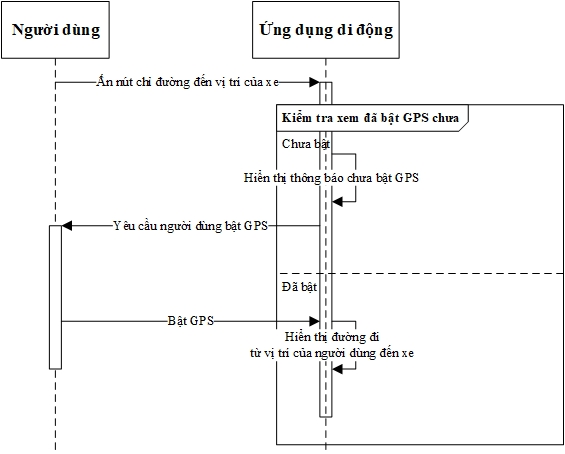
*Hình 3.14. Sơ đồ tuần tự “Xem vị trí hiện tại của xe”*

1.6.3. “Xem vị trí hiện tại của người dùng”



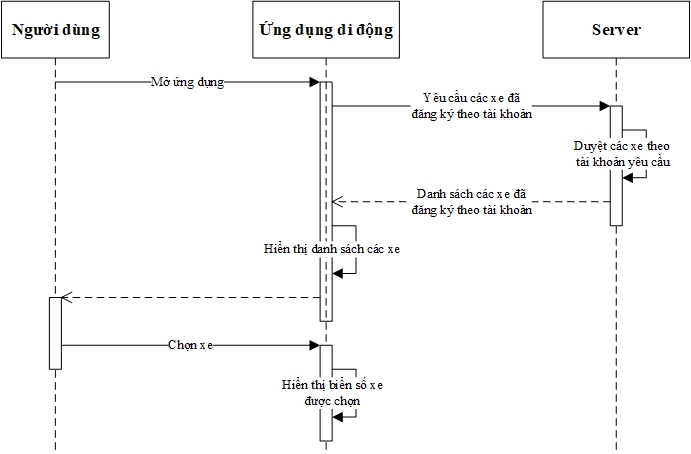
*Hình 3.15 Sơ đồ tuần tự “Xem vị trí hiện tại của người dùng”*

1.6.4. “Chỉ đường đến vị trí của xe”



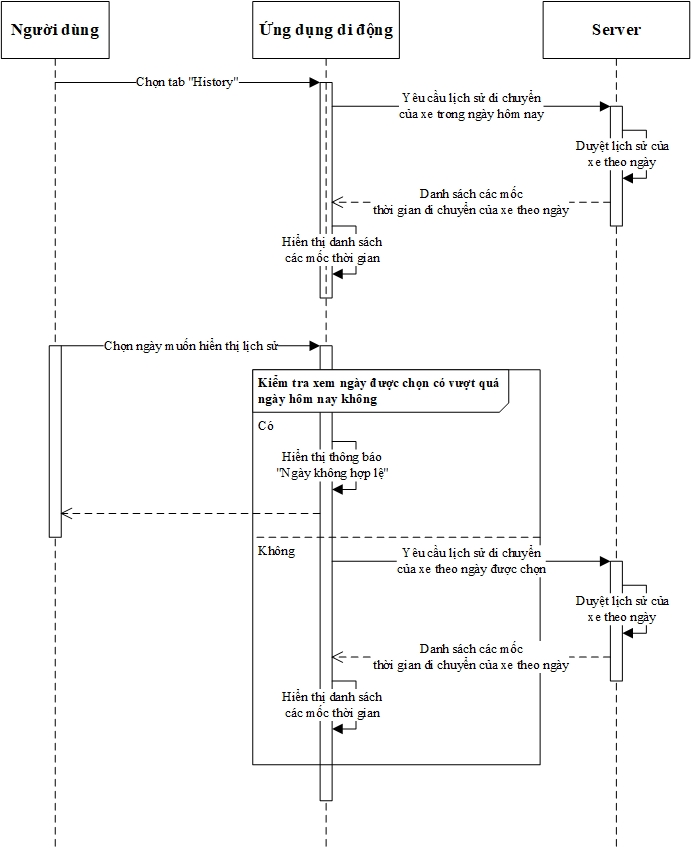
*Hình 3.16 Sơ đồ tuần tự “Xem vị trí hiện tại của người dùng”*

1.6.5. “Chọn xe”



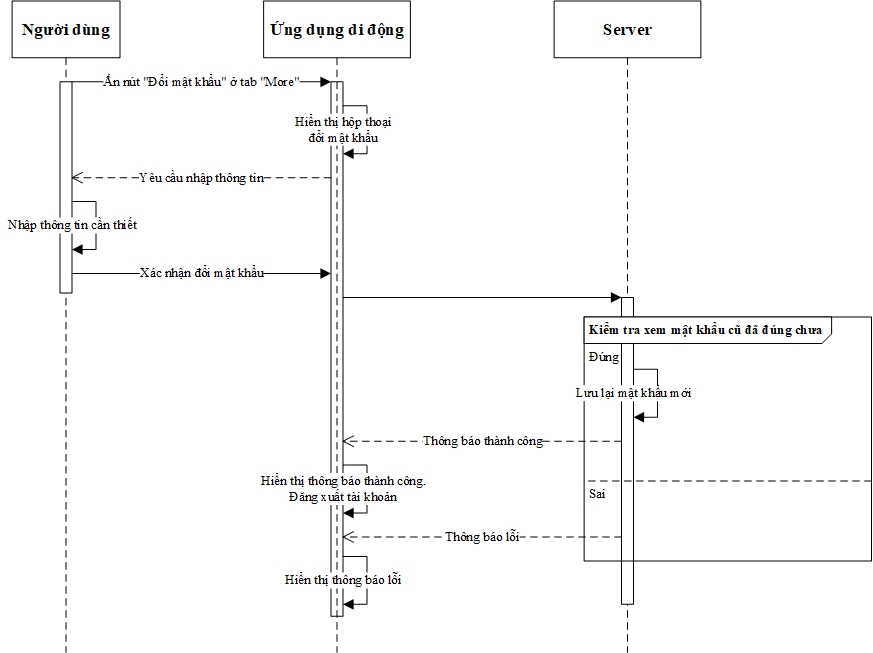
*Hình 3.17 Sơ đồ tuần tự “Chọn xe”*

1.6.6. “Xem lịch sử di chuyển theo ngày”

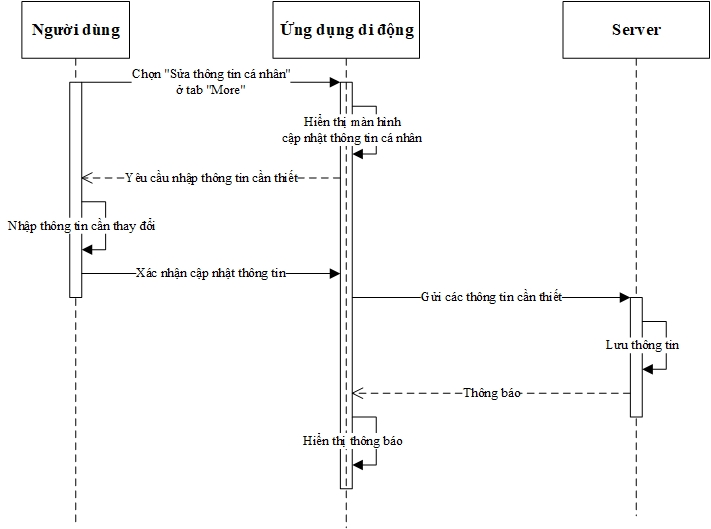


*Hình 3.18. Sơ đồ tuần tự “Xem lịch sử di chuyển theo ngày”*

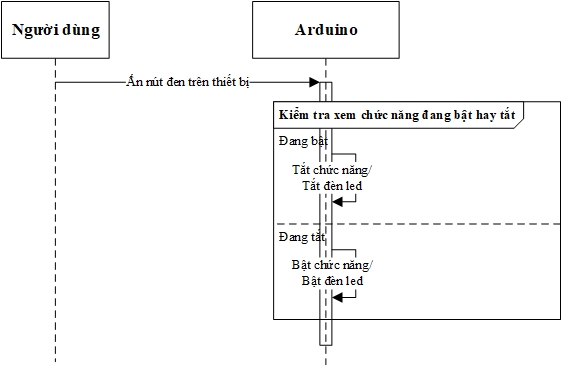
1.6.7. “Đổi mật khẩu”



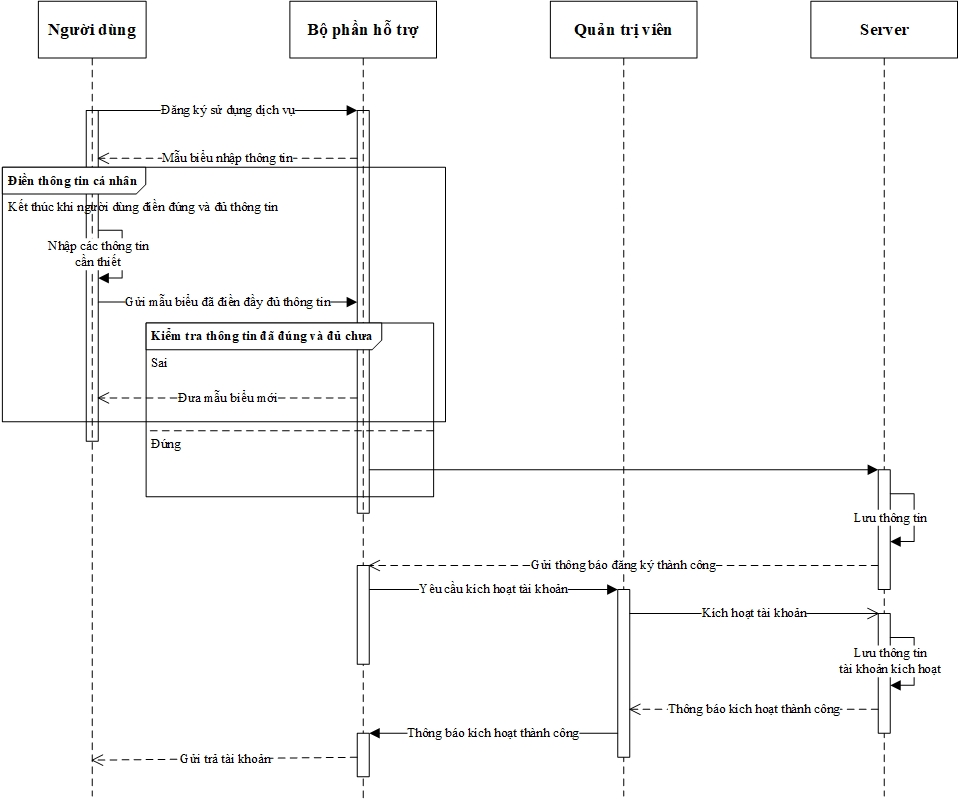
1.6.8. “Cập nhật thông tin cá nhân”



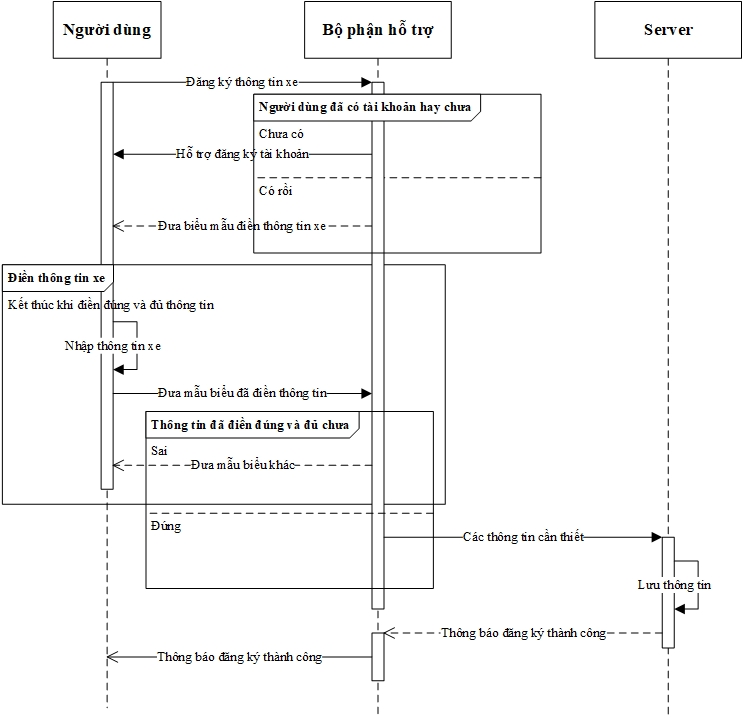
1.6.9. “Bật/tắt chức năng ghi lại lịch sử di chuyển”



1.6.10. “Đăng ký tài khoản”



1.6.11. “Đăng ký xe”



## 2. Phân tích dữ liệu nghiệp vụ

Phần này tập trung chủ yếu vào việc phân tích dữ liệu nghiệp vụ của hệ thống, để từ đó đưa ra được cơ sở dữ liệu cuối cùng cho hệ thống.

### 2.1. Xác định kiểu dữ liệu ban đầu

Từ mô tả hệ thống, ta xác định được những thực thể sau:

NGƯỜI DÙNG (Email, Tên đăng nhập, Mật khẩu, Họ tên, Địa chỉ nhà, Số điện thoại, Vai trò).

PHƯƠNG TIỆN (Biển số xe, Tên người sở hữu, Hãng xe, Loại xe, Mô tả, Vĩ độ, Kinh độ, Ngày, Thời gian bắt đầu, Thời gian kết thúc).

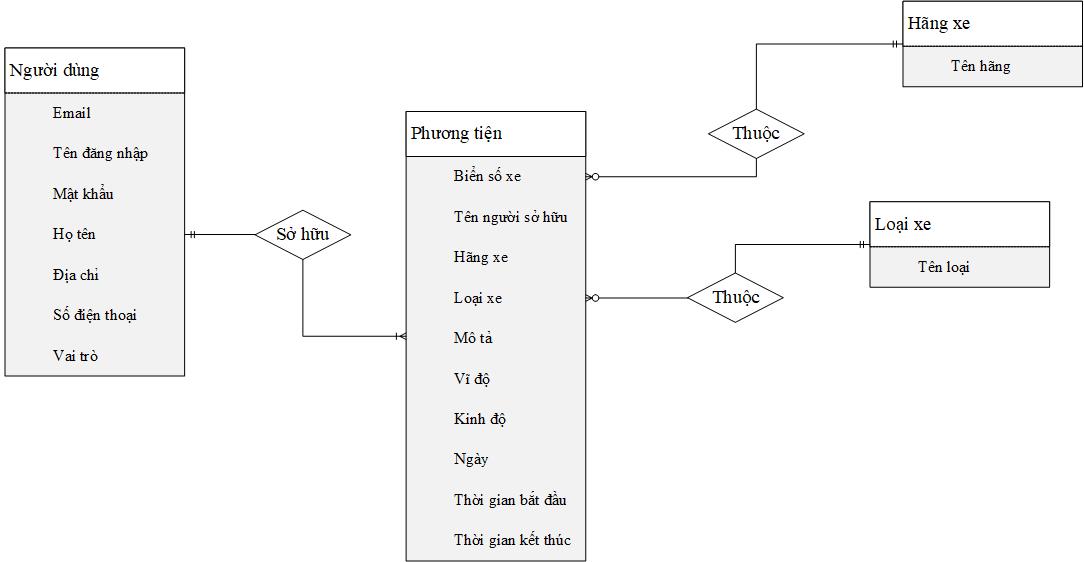
HÃNG XE (Tên hãng xe).

LOẠI XE (Tên loại xe).

VỊ TRÍ (Vĩ độ, Kinh độ, Biển số xe, Ngày, Thời gian bắt đầu, Thời gian kết thúc).

### 2.2. Mô hình liên kết thực thể

2.2.1. Mô hình liên kết thực thể mở rộng

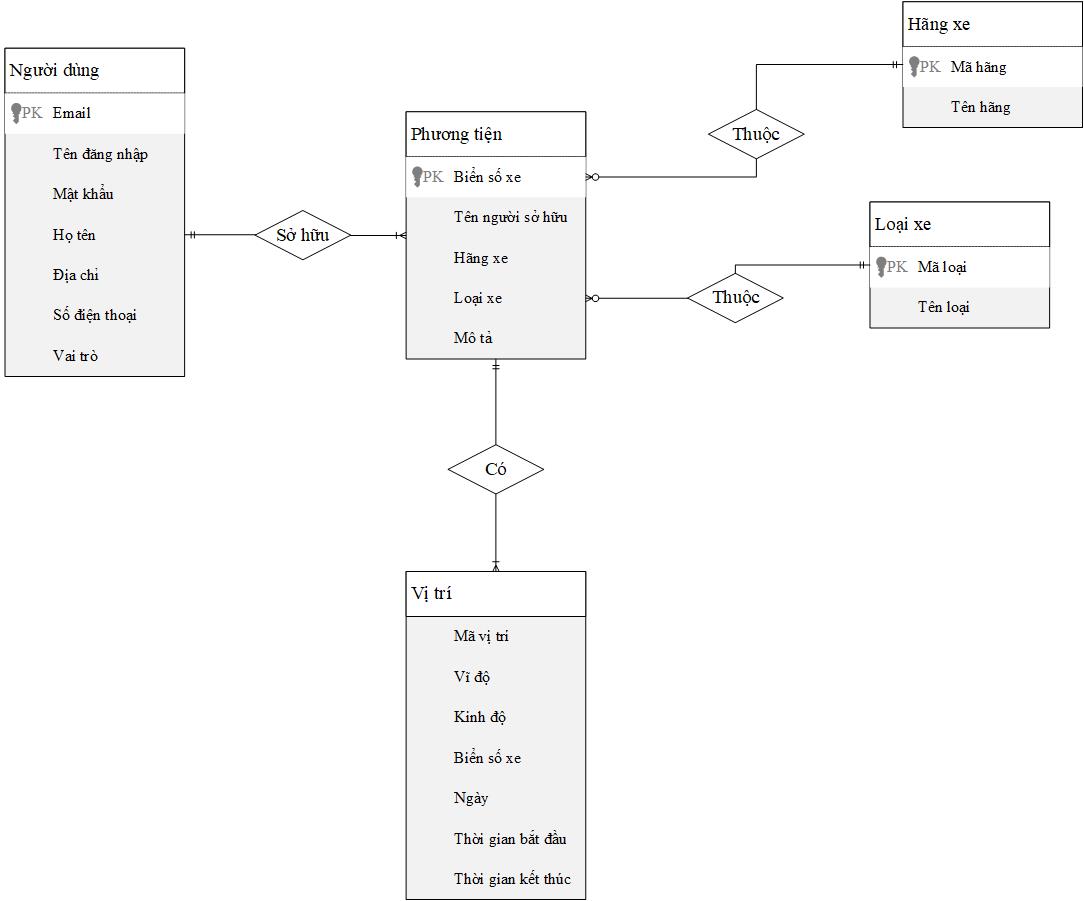


2.2.2. Mô hình liên kết thực thể kinh điển

Từ mô hình liên kết thực thể mở rộng, ta chuyển sang mô hình liên kết thực thể kinh điển như sau:

+ Các bảng có thuộc tính đa trị sẽ được tách thành các bảng riêng.

+ Thêm các thuộc tính cũng như xác định khóa chính cho các bảng.

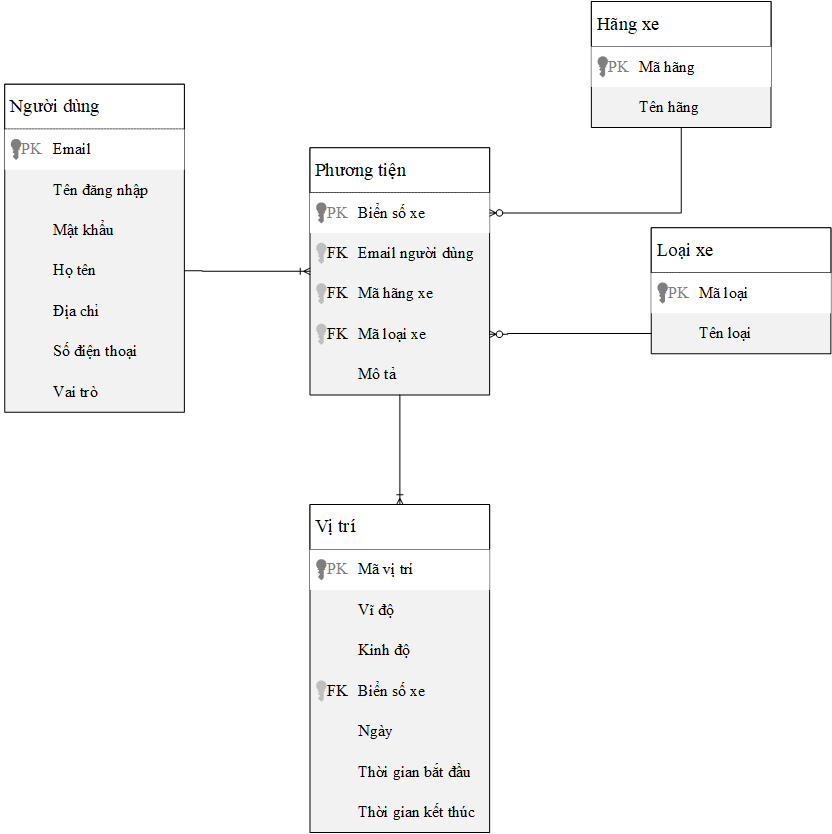


2.2.3. Mô hình liên kết thực thể hạn chế

Từ mô hình liên kết thực thể kinh điển, ta chuyển sang mô hình liên kết thực thể hạn chế như sau:

+ Thêm các thuộc tính khóa ngoại vào các bảng.

+ Bổ sung đầy đủ khóa chính cho các bảng.



### 2.3. Mô hình quan hệ

Từ mô hình liên kết thực thể hạn chế, ta chuyển sang mô hình quan hệ như sau:

+ Chuẩn hóa lại tên gọi:

- Thực thể “Người dùng”:

|  |  |  |
| --- | --- | --- |
| Người dùng 🠢 User | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Email | Email |
| 2 | Tên đăng nhập | Username |
| 3 | Password | Password |
| 4 | Họ tên | Name |
| 5 | Địa chỉ | Address |
| 6 | Số điện thoại | PhoneNumber |
| 7 | Vai trò | Role |

- Thực thể “Phương tiện”:

|  |  |  |
| --- | --- | --- |
| Phương tiện 🠢 Vehicle | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Biển số xe | NumberPlate |
| 2 | Email người dùng | UserEmail |
| 3 | Mã hãng xe | BrandId |
| 4 | Mã loại xe | TypeId |
| 5 | Mô tả | Description |

- Thực thể “Hãng xe”:

|  |  |  |
| --- | --- | --- |
| Hãng xe 🠢 Brand | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã hãng | Id |
| 2 | Tên hãng | Name |

- Thực thể “Loại xe”:

|  |  |  |
| --- | --- | --- |
| Loại xe 🠢 Type | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã loại | Id |
| 2 | Tên loại | Name |

- Thực thể “Vị trí”:

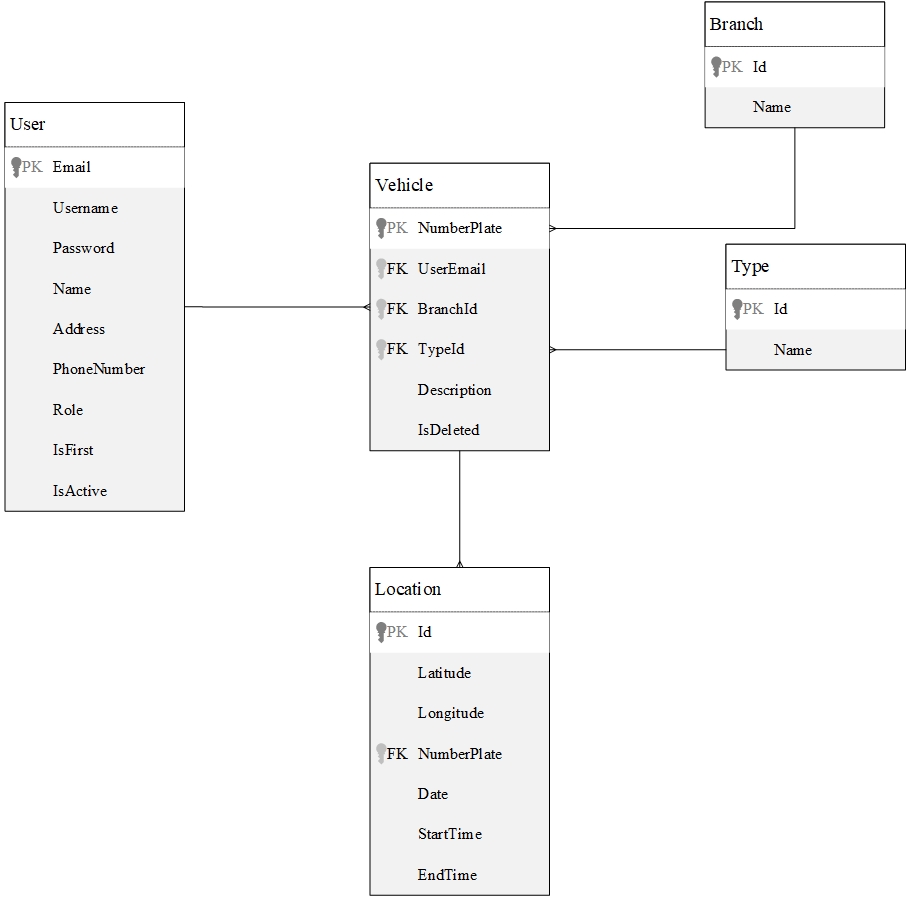
|  |  |  |
| --- | --- | --- |
| Vị trí 🠢 Location | | |
| **STT** | **Tên trường cũ** | **Tên trường mới** |
| 1 | Mã vị trí | Id |
| 2 | Vĩ độ | Latitude |
| 3 | Kinh độ | Longitude |
| 4 | Biển số xe | NumberPlate |
| 5 | Ngày | Date |
| 6 | Thời gian bắt đầu | StartTime |
| 7 | Thời gian kết thúc | EndTime |

+ Thêm các trường để đảm bảo tính an toàn cho hệ thống. Cụ thể:

- Bảng “User” thêm trường “IsFirst” để xác định xem có phải lần đầu người dùng đăng nhập hay không; thêm trường “IsActive” để xác định xem người dùng này đã được kích hoạt hay chưa.

- Bảng “Vehicle” thêm trường “IsDeleted” để xác định xem phương tiện đó đã bị loại bỏ khỏi hệ thống hay chưa.

*Mô hình quan hệ:*



### 2.4. Đặc tả dữ liệu

+ Bảng “User”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Email | varchar(80) | x |  | Email đăng ký tài khoản |
| 2 | Username | varchar(20) |  |  | Tên đăng nhập hiển thị |
| 3 | Password | varchar(20) |  |  | Mật khẩu đăng nhập |
| 4 | Name | nvarchar(100) |  |  | Tên thật |
| 5 | Address | nvarchar(100) |  |  | Địa chỉ nhà |
| 6 | PhoneNumber | varchar(20) |  |  | Số điện thoại của người dùng |
| 7 | Role | int |  |  | Vai trò của người dùng:  1 – Quản trị viên  2 – Người dùng |
| 8 | IsFirst | int |  |  | Có phải lần đầu đăng nhập không  0 – Không phải  1 – Có |
| 9 | IsActive | int |  |  | Tài khoản đã được kích hoạt hay chưa  0 – Chưa được  1 – Đã kích hoạt |

+ Bảng “Branch”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Name | nvarchar(100) |  |  | Tên hãng xe |

+ Bảng “Type”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Name | nvarchar(100) |  |  | Tên loại xe |

+ Bảng “Vehicle”:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | NumberPlate | varchar(20) | x |  | Biển số xe đăng ký với thiết bị |
| 2 | UserEmail | varchar(80) |  | x | Email của người dùng |
| 3 | BranchId | int |  | x | Mã hãng xe |
| 4 | TypeId | int |  | x | Mã loại xe |
| 5 | Description | nvarchar(300) |  |  | Mô tả về thiết bị |
| 6 | IsDeleted | int |  |  | Đã bị xóa hay chưa  0 – Chưa xóa  1 – Đã xóa |

+ Bảng “Location”:

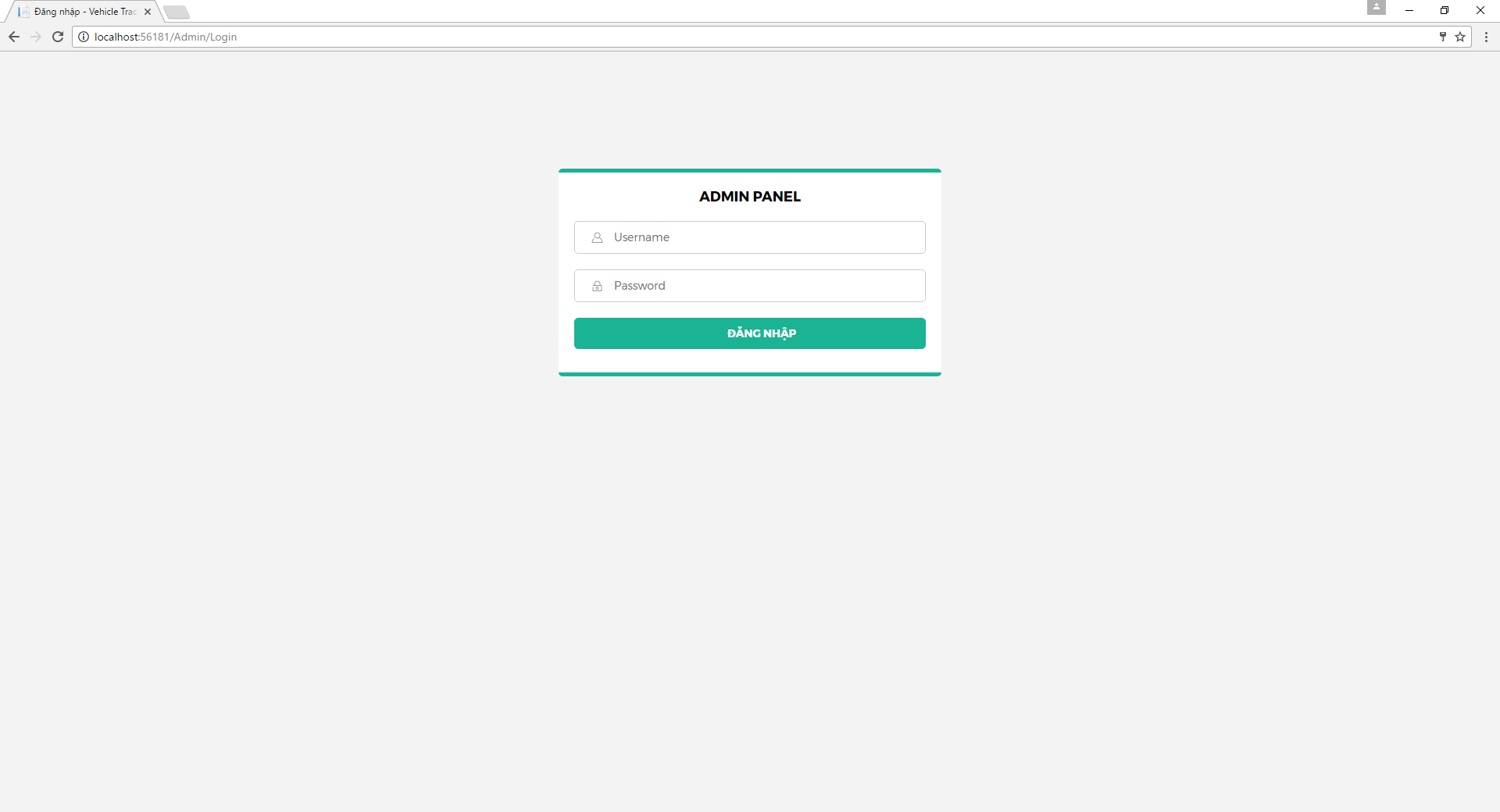
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường** | **Kiểu dữ liệu** | **Khóa chính** | **Khóa ngoại** | **Mô tả** |
| 1 | Id | int | x |  | Id |
| 2 | Latitude | varchar(20) |  |  | Vĩ độ |
| 3 | Longitude | varchar(20) |  |  | Kinh độ |
| 4 | NumberPlate | varchar(20) |  | x | Biển số xe |
| 5 | Date | varchar(20) |  |  | Ngày |
| 6 | StartTime | varchar(20) |  |  | Thời gian bắt đầu ghi lịch sử |
| 7 | EndTime | varchar(20) |  |  | Thời gian kết thúc ghi lịch sử |

## 3. Thiết kế giao diện

### 3.1. Giao diện quản trị

3.1.1. Màn hình “Đăng nhập”

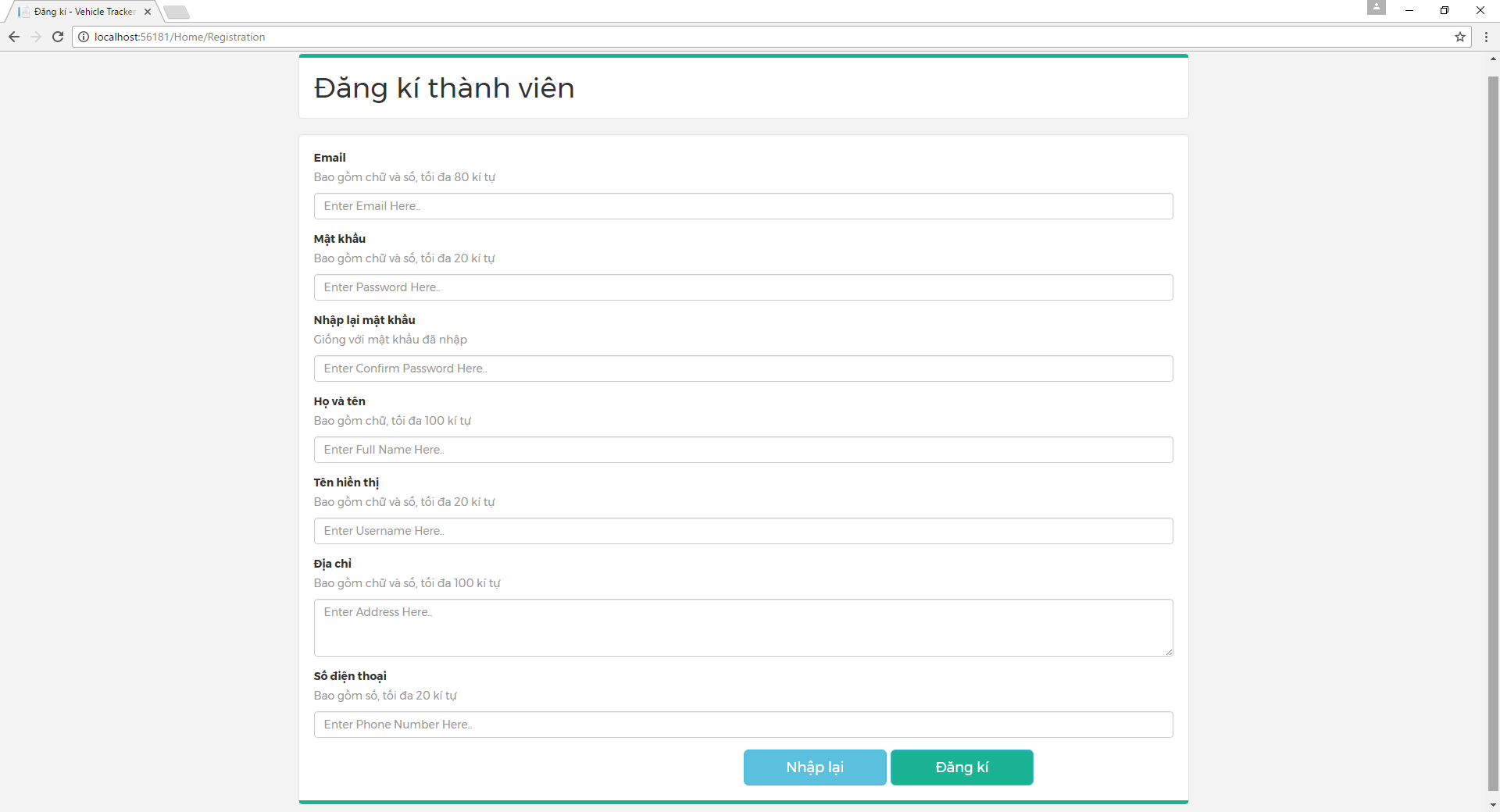
Màn hình đăng nhập vào trang Admin. Nếu là Admin, nhập thông tin User và Password vào ô tương ứng, ấn đắng nhập. Nếu thành công, sẽ đi vào trang quản trị. Nếu không, sẽ hiện ra thông báo yêu cầu đăng nhập lại



*Màn hình đăng nhập Admin*

3.1.2. Màn hình “Đăng ký thành viên”

Màn hình đăng kí thành viên. Bộ phận hỗ trợ sẽ nhập thông tin theo mẫu biểu đăng kí mà người dùng đã điển. Nếu thông tin không đúng kiểu dữ liệu sẽ hiện thông báo và yêu cầu nhập lại. Sau khi các thông tin chính xác, ấn đăng kí.



*Màn hình đăng kí thành viên*

3.1.3. Màn hình “Quản lý thông tin người dùng”

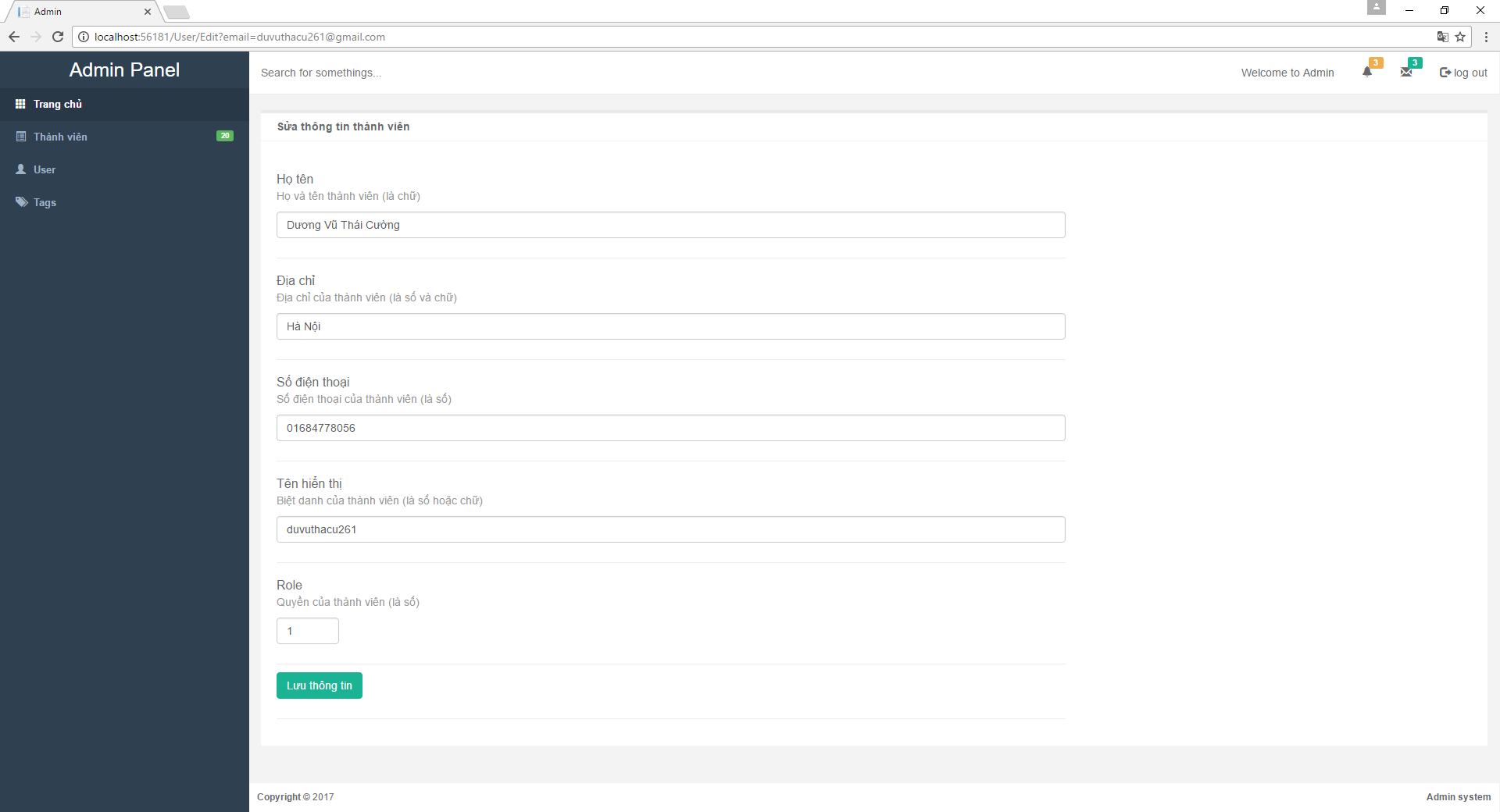
Màn hình quản lý thông tin người dùng trong Admin Panel. Admin có thể sửa, xem và xóa thông tin người dùng nếu cần thiết



*Màn hình quản lý thông tin người dùng*

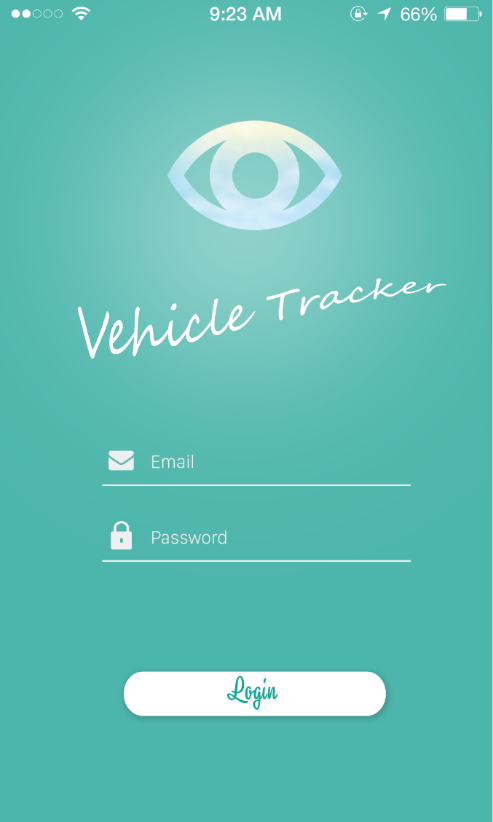
3.1.4. Màn hình “Sửa thông tin người dùng”

Màn hình sửa thông tin người dùng trong Admin Panel



### 3.2. Giao diện người dùng

3.2.1. Màn hình “Đăng nhập”



*Màn hình “Đăng nhập”*

+ Đây là màn hình đăng nhập của ứng dụng. Khi mới sử dụng ứng dụng, người dùng sẽ phải đăng nhập thông qua màn hình này. Các thông tin cần nhập là: username và mật khẩu của người dùng.

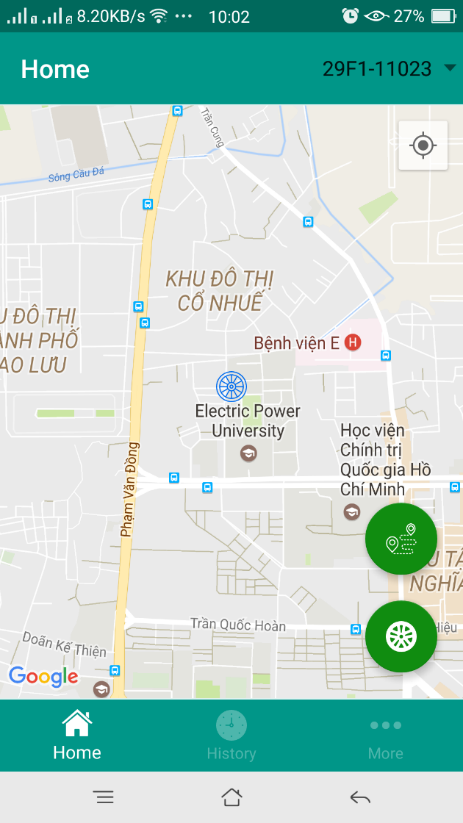
3.2.2. Màn hình “Home”

+ Đây là màn hình hiển thị vị trí của người dùng và của phương tiện mà người dùng đang chọn. Mặc định ứng dụng sẽ hiển thị vị trí hiện tại của người dùng.



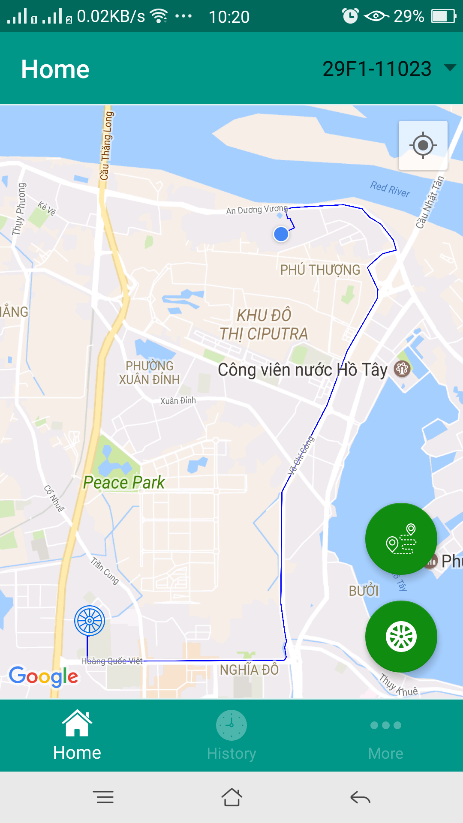
*Màn hình “Home”*

+ Người dùng có thể xem vị trí của xe bằng cách ấn nút tròn có hình bánh xe.



*Màn hình hiển thị vị trí của xe*

+ Ấn nút phía trên nút hiển thị vị trí xe để hiện đường đi đến xe của mình.



*Màn hình chỉ đường đến vị trí hiện tại của xe*

+ Chọn xe để giám sát bằng cách ấn vào mũi tên ở trên toolbar



*Màn hình chọn xe để giám sát*

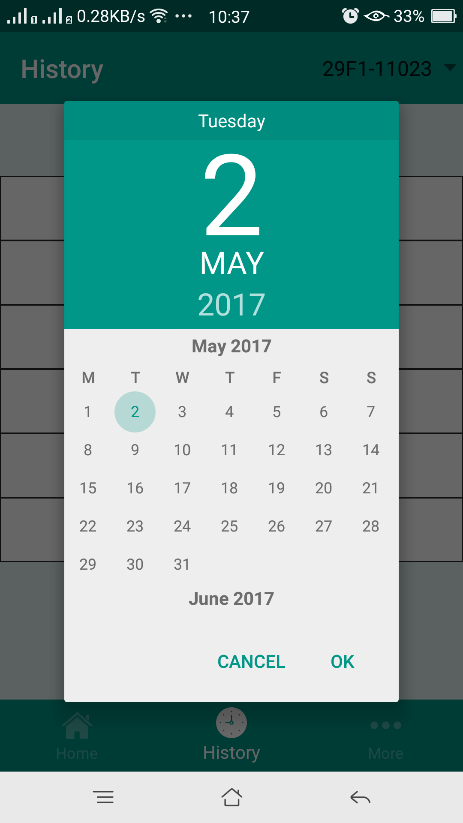
3.2.3. Màn hình “Lịch sử”

+ Đây là màn hình hiển thị lịch sử di chuyển của phương tiện được chọn theo ngày.



*Màn hình hiển thị lịch sử di chuyển*

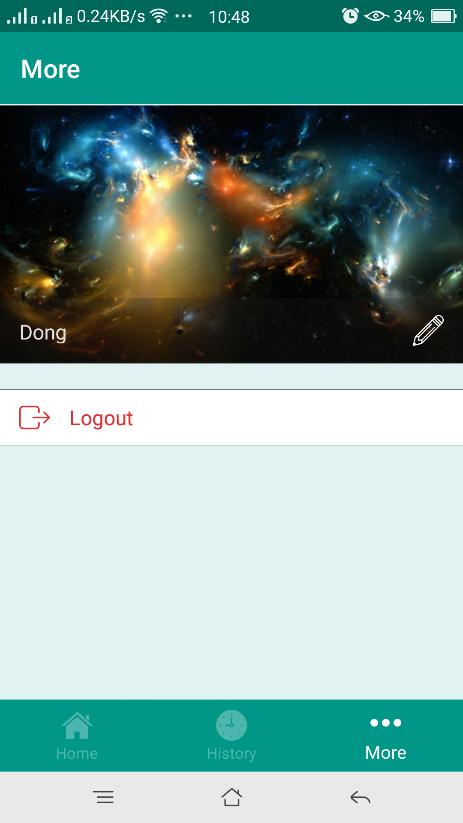
+ Người dùng có thể chọn ngày để xem lịch sử bằng cách ấn vào nút hiển thị ngày



*Màn hình chọn ngày*

3.2.4. Màn hình “More”

+ Ở màn hình này, người dùng có thể cập nhật lại thông tin cá nhân, đăng xuất tài khoản.



*Màn hình “More”*

+ Người dùng ấn “Logout” để đăng xuất.

# **Chương 4: TRIỂN KHAI PHẦN CỨNG**

## 1. Triển khai phần cứng

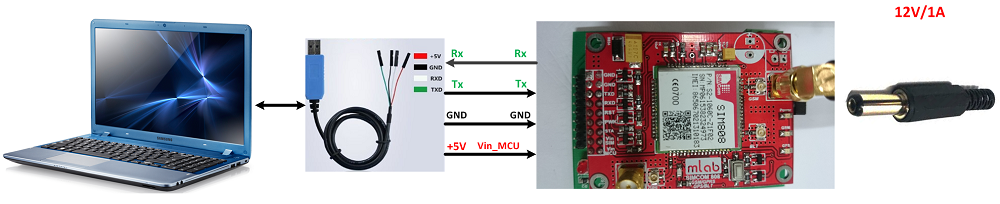
### 1.1. Lập trình với Module Sim808

\* Yêu cầu:

+ 1 nguồn 12V – 2A.

+ 1 dây USB to TTL.

\* Cài đặt như sau



\* Lập trình

+ Các lệnh dùng cho ứng dụng HTTP

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT+HTTPINIT | Khởi tạo dịch vụ HTTP | OK – Khởi tạo thành công.  ERROR – Khởi tạo thất bại. |
| AT+HTTPTERM | Hủy dịch vụ HTTP | OK – Hủy thành công.  ERROR – Hủy thất bại |
| AT+HTTPPARA=  <HTTPParamTag>,  <HTTPParamValue> | Thiết lập giá trị cho các tham số HTTP  \* HTTPParamTag:  + “URL”: (Tham số HTTP bắt buộc) | OK – Thiết lập thành công.  ERROR – Thiết lập thất bại. |
| AT+HTTPACTION=  <Method> | Thiết lập phương thức HTTP  \* Method:  + 0: GET  + 1: POST  + 2: HEAD | OK – Thành công.  ERROR – Thất bại. |
| AT+HTTPREAD | Đọc dữ liệu server trả về | + Đọc thành công  HTTPREAD:<data\_len>  <data>  OK  + Đọc thất bại: ERROR |

+ Các lệnh chung

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT | Kiểm tra phản hồi của Module Sim808 | OK => Module hoạt động bình thường |

+ Lệnh điều khiển GPS

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT+CGNSPWR=[x] | Điều khiển bật tắt GPS  x = 1: bật  x = 0: tắt |  |
| AT+CGNSINF | Yêu cầu thông báo thông tin GPS |  |
| AT+CGNSURC=[x] | Cài đặt chu kỳ báo cáo thông tin GPS  x là thời gian theo mili giây. |  |

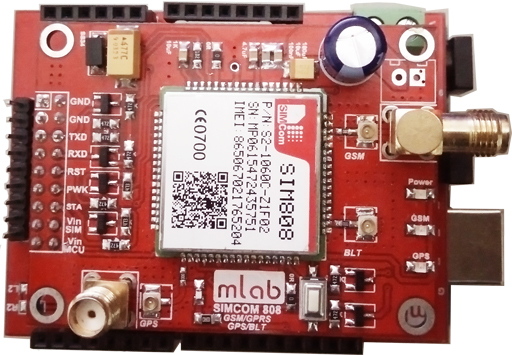
+ Lệnh điều khiển GPRS

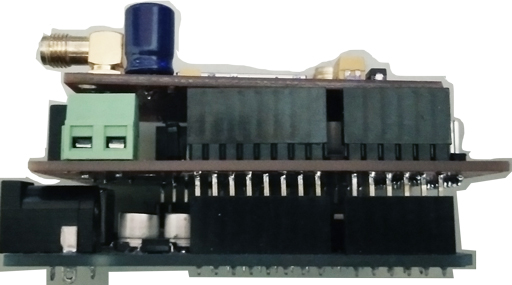
|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Giá trị trả về** |
| AT+CGATT? | Xem trạng thái đã được gắn GPRS hay chưa | + CGATT: <state>  OK  <state>: 0 – chưa gắn  1 – đã gắn |
| AT+CGACT=<state>,<cid> | Kích hoạt hoặc hủy bỏ PDP Context | OK  <state>: 0 – hủy bỏ  1 – kích hoạt  <cid>: PDP context id |

### 1.2. Kết hợp Module Sim808 với Arduino UNO

\* Yêu cầu: 1 nguồn 12V – 2A

\* Cài đặt:





# **KẾT LUẬN**

1. Kết quả đạt được

+ Đã phát triển được một hệ thống IoT: hệ thống giám sát phương tiện.

+ Bước đầu làm quen với lập trình nhúng.

+ Hoàn thiện được các chức năng cơ bản của hệ thống: xem vị trí hiện tại của phương tiện và xem lịch sử di chuyển của phương tiện.

2. Hướng phát triển

Định hướng hệ thống giám sát phương tiện trong những phiên bản tiếp theo

+ Người dùng có thể xem được những xe xung quanh mình (các xe phải cùng sử dụng chung hệ thống).

+ Phần hiển thị lịch sử di chuyển sẽ thể hiện được tốc độ di chuyển của phương tiện.

+ Xây dựng ứng dụng chạy trên hệ điều hành iOs.

# **TÀI LIỆU THAM KHẢO**

[1]. <https://www.arduino.cc/en/Guide/HomePage>

[2]. <http://www.webopedia.com/TERM/I/internet_of_things.html>

[3]. <https://vi.wikipedia.org/wiki/M%E1%BA%A1ng_l%C6%B0%E1%BB%9Bi_v%E1%BA%A1n_v%E1%BA%ADt_k%E1%BA%BFt_n%E1%BB%91i_Internet>