**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



**BÁO CÁO ĐỒ ÁN HỆ THỐNG NHÚNG**

**Đề tài**

**CAR PARKING**

*Giảng viên hướng dẫn:* **Lê Trung Quân**

*Sinh viên thực hiện:* **NHÓM 02**

1. Dư Công Hoàng Anh 15520016
2. Nguyễn Chí Bảo 15520039
3. Trần Nam Bàng 15520032
4. Nguyễn Trung Sĩ 15520726

Mục lục

[LỜI CẢM ƠN 3](#_Toc532417414)

[CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI 4](#_Toc532417415)

[1.1. Hiện trạng 4](#_Toc532417416)

[1.2. Giải pháp 4](#_Toc532417417)

[1.3. Mục đích đối tượng và phạm vi nghiên cứu 4](#_Toc532417418)

[1.3.1. Mục đích nghiên cứu 4](#_Toc532417419)

[1.3.2. Đối tượng nghiên cứu 4](#_Toc532417420)

[1.3.3. Phạm vi nghiên cứu 4](#_Toc532417421)

[1.4. Phương pháp thực hiện 4](#_Toc532417422)

[1.5. Thuận lợi và khó khăn 5](#_Toc532417423)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 6](#_Toc532417424)

[2.1. Giao thức UART 6](#_Toc532417425)

[2.1.1 Khái niệm 6](#_Toc532417426)

[2.1.2 Các thông số cơ bản 6](#_Toc532417427)

[2.1.3 Hoạt động 6](#_Toc532417428)

[2.2. Chuẩn SPI 7](#_Toc532417429)

[2.2.1 Khái niệm 7](#_Toc532417430)

[2.2.2 Hoạt động 8](#_Toc532417431)

[2.3. Linh kiện được sử dụng trong quá trình thực hiện đồ án. 9](#_Toc532417432)

[2.4. Module NODEMCU ESP8266 10](#_Toc532417433)

[2.5. Module RFID RC522 NFC 13.56 MHZ 10](#_Toc532417434)

[CHƯƠNG 3 THIẾT KẾ VÀ HIỆN THỰC HỆ THỐNG 11](#_Toc532417435)

[3.1. Mô hình thiết kế tổng thể của hệ thống 11](#_Toc532417436)

[3.2. Thiết kế phần cứng 12](#_Toc532417437)

[3.3. Cơ chế hoạt động 14](#_Toc532417438)

[3.4. Quá trình xử lý ở server 16](#_Toc532417439)

[3.5. Quá trình xử lý ở dữ liệu 16](#_Toc532417440)

[3.6. Quá trình xử lý ở web client 17](#_Toc532417441)

[KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ 19](#_Toc532417442)

[TÀI LIỆU THAM KHẢO 19](#_Toc532417443)

# LỜI CẢM ƠN

Nhóm chúng em xin cảm ơn đến Khoa Mạng máy tính và Truyền thông và các thầy (cô) đã tạo điều kiện cho chúng em làm đồ án môn Hệ thống nhúng (học kì I năm học 2018-2019)

Đặc biệt chúng em gửi lời cảm ơn tới thầy Lê Trung Quân và thầy Đặng Lê Bảo Chương đã tận tình giúp đỡ và hỗ trợ chúng em trong các giai đoạn thực hiện đề tài đồ án. Nhờ vậy, chúng em cũng tích lũy được vốn kiến thức và kinh nghiệm hơn trong việc xây dựng và làm việc nhóm hiệu quả.

Chúng em cũng không lường hết sai sót và độ hoàn thiện đề tài vì thế rất mong quý thầy (cô) góp ý để chúng em hoàn thành tốt hơn.

Chúng em xin chân thành cảm ơn và gửi lời chúc sức khỏe tới các thầy (cô).

# CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI

## Hiện trạng

* Hiện nay trong các khu vực trung tâm thành phố, các khu vực đông dân cư, việc tìm kiếm một bãi đỗ xe khi chúng ta đi vào khu vực đó là rất khó khăn và mất thời gian. Điều đó dẫn đến tình trạng đậu xe ở lòng đường vỉa hè diễn ra phức tạp.

## Giải pháp

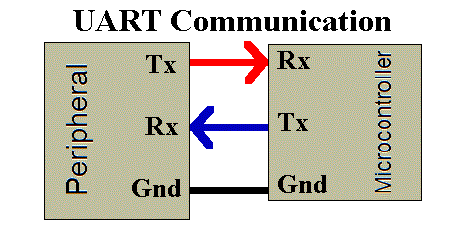
* Xây dựng hệ thống IOT để thông báo cho người dùng biết số lượng chỗ trống của bãi xe và quản lý bãi xe một cách dễ dàng, tiết kiệm chi phí.
  1. Mục đích đối tượng và phạm vi nghiên cứu
     1. Mục đích nghiên cứu
* Đồ án tập trung nghiên cứu, phát triển hệ thống thông báo số lượng chỗ trống, địa chỉ của các bãi đỗ xe, kết nối các bãi đỗ xe.
* Đồ án sử dụng module RCC522 để đọc thẻ, kit stm32f429 dùng để điều khiển trung tâm các module hỗ trợ khác như ESP8266… để tạo một hệ thống hoàn chỉnh. Cụ thể đồ án thực hiện:
* Tìm hiểu cơ chế hoạt động của RCC522.
* Tìm hiểu giao tiếp giữa các thành phần trong hệ thống như kit stm32f429, module RCC522, ESP8266.
* Xây dựng server, database, web client để tạo thành một trang web để hiển thị thông tin số lượng chỗ trống, địa chỉ bãi giữ xe.
  + 1. Đối tượng nghiên cứu
* Các thiết bị lắp ráp hệ thống quẹt thẻ xe, và ứng dụng web quản lý bãi đỗ xe và một vài tác vụ liên quan.
  + 1. Phạm vi nghiên cứu
* Trong giới hạn đồ án, nhóm giới hạn nghiên cứu quản lý 2 bãi đỗ xe, số lượng chỗ để xe mỗi bãi là 9 và 4 chỗ. Ngoài ra việc mở rộng rất dễ dàng.
  1. Phương pháp thực hiện
* Nhóm đã tìm và nghiên cứu các tài liệu liên quan đến hệ thống để rút ra những thông tin cần thiết để phục vụ cho đồ án. Sau đó nhóm thực hiện tầng thành phần của hệ thống. Cuối cùng kết nối các thành phần của hệ thống và giải quyết các vấn đề phát sinh để hoàn thiện hệ thống.
* Tìm hiểu tài liệu, các dự án đã và đang được phát triển trên module RCC522 và datasheet của module.
* Tìm hiểu về ESP8266 để lựa chọn phiên bản và nền tảng phù hợp. Sau đó tìm hiểu socket.IO để ứng dụng vào đồ án.
* Tìm hiểu về lập trình web qua các bài viết hướng dẫn trên mạng sử dụng HTML, JavaScipt để xây dựng giao diện web đơn giản.
* Tìm hiểu về socket.IO để kết hợp với NodeJs ở backend và ESP8266.
* Giải quyết các vấn đề chính xác của module, độ ổn định.
  1. Thuận lợi và khó khăn
     1. Thuận lợi
* Trong quá trình làm đề tài nhóm đã nhận được sự giúp đỡ của bạn bè, thầy cô.
* Ngoài ra sự phát triển của internet đã đem lại cho nhóm nhiều tài liệu bổ ích và rất thiết thực.
* Sự hợp tác tốt giữa các thành viên trong nhóm là một phần đóng góp thành công cho đồ án.
  + 1. Khó khăn
* Kiến thức và kinh nghiệm trong lập trình web và server còn hạn chế nên trong việc phát triển web về giao diện cũng như bảo mật còn gặp nhiều khó khăn.

# CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

## Giao thức UART

1. Khái niệm

* UART(Universal Asynchronous Receiver – Transmitter) là bộ truyền nhận nối tiếp không đồng bộ.

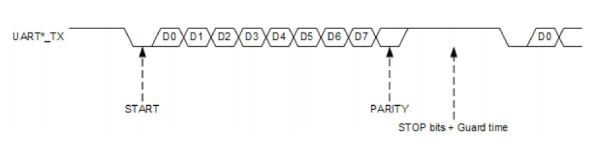


1. Các thông số cơ bản

* Baund rate: khoảng thời gian dành cho 1 bit được truyền. Phải được cài đặt giống nhau ở giữ và nhận.
* Frame: Khung truyền quy định về số bit trong mỗi lần truyền.
* Start bit: Là bit đầu tiên được truyền trong 1 Frame. Báo cho thiết bị nhận có một gói dữ liệu sắp được truyền đến.
* Data: Dữ liệu cần truyền. Bit có trọng số nhỏ nhất LSB được truyền trước sau đó đến bit MSB.
* Parity bit: Bit kiểm tra dữ liệu.
* Stop bit: là bit báo cho thiết bị rằng các data đã được gửi xong. Thiết bị nhận sẽ tiến hành kiểm tra khung truyền nhằm đảm bảo tính đúng đắn của dữ liệu.
* Stop bit: Là bit báo cho thiết bị rằng các data đã được gửi xong. Thiết bị nhận sẽ tiến hành kiểm tra khung truyền nhằm đảm bảo tín đúng đắn của dữ liệu.

1. Hoạt động

* Để bắt đầu cho việc truyền dữ liệu bằng UART, một START bit được gửi đi, sau đó là các bit dữ liệu và kết thúc quá trình truyền là bit STOP bit.

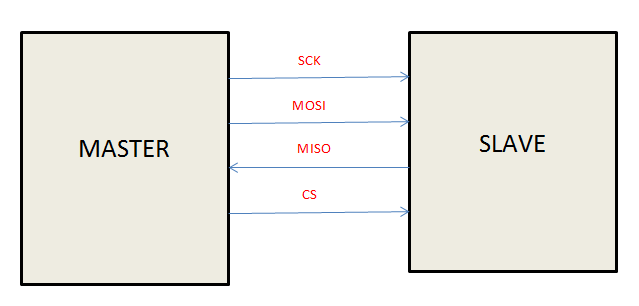


* Khi ở trạng thái chờ mức điện thế ở mức 1 (hight).
* Khi bắt đầu truyền, START bit sẽ chuyển từ 1 xuống 0 để báo hiệu cho bộ nhận là quá trình truyền dữ liệu sắp xảy ra.
* Sau START bit là đến các bit dữ liệ D0-D7( các bit này có thể mức Hight hay Low tùy theo dữ liệu).
* Cuối cùng là STOP bit là bảo cho thiết bị rằng các bit đã được gửi xong. Thiết bị nhận sẽ tiến hành kiểm tra khung truyền nhằm đảm bảo tính đúng đắn của dữ liệu.

## Chuẩn SPI

### Khái niệm

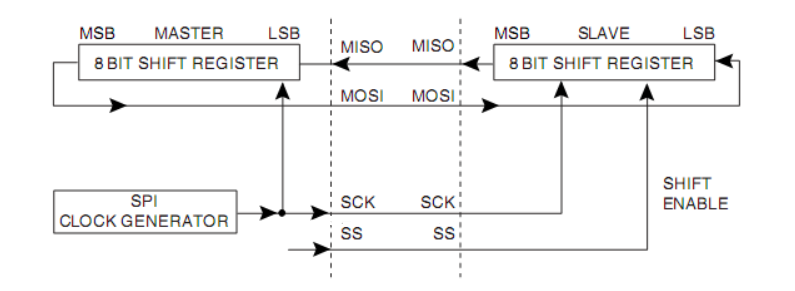
* SPI( Serial Peripheral Bus) được phát triển bởi Motorola. Đây là một chuẩn đồng bộ nối tiếp để truyền dữ liệu ở chế độ song công toàn phần( full-duplex) tức trong cùng một thời điếm có thể xảy ra đòng thời quá trình truyền và nhận. Đôi khi SPI còn được gọi là chuẩn giao tiếp 4 dây(Four-wire).



* SPI là giao tiếp đồng bộ, bất cứ quá trình truyền nào cũng được đồng bộ hóa với tín hiệu clock chung. Tín hiệu sinh ra bởi master.
* Trong giao diện SPI có bốn tín hiệu số:
* MOSI hay SI – cổng ra bên Master ( Master Out Slave In). Đây là chân dành cho việc truyền tín hiệu từ Master tới Slave.
* MISO hay SO – Cổng ra bên Slave ( Master In Slauve Out). Đây là chân dành cho việc truyền tín hiệu từ Slave tới Master.
* SCLK hay SCK là tín hiệu clock đồng bộ(Serial Clock). Xung nhịp chỉ được tạo bởi Master.
* CS hay SS là tín hiệu chọn vi mạch (Chip Select hoặc Slave Select). SS sẽ ở mức thấp khi không làm việc. Nếu Master kéo SS xuống cao thì sẽ xảy ra quá trình giao tiếp.

### Hoạt động

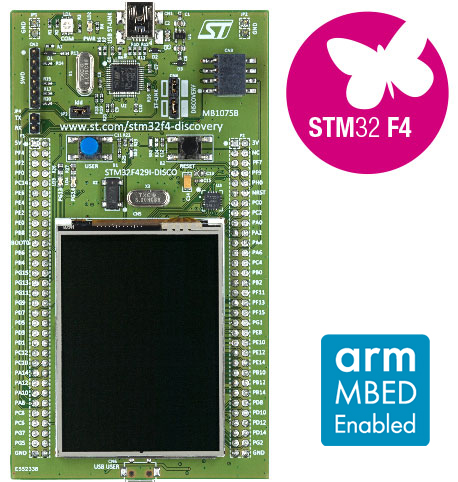
* + Mỗi chip Master hay Slave có thành ghi dữ liệu 8 bits. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, một bit trong thanh ghi dữ liệu của Master đực truyền trên đường MOSI, đồng thời một bit trong thnah ghi dữ liệu của chip Slave cũng truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip gửi qua lại đồng thời nên quá trình truyền dư liệu này được gọi là “song công”.



* + Cực của xung giữ nhip, phase và chế độ hoạt động: cực của xung giữ nhip(Clock Polariy) được gọi tắt là CPOL là khái niện dùng chỉ trạng thái của chân SCK ở trạng thái nghỉ. Ở trạng thái nghỉ(Idle), chân SCK có thể dữ mức cao CPOL=1 hoặc thấp CPOL=0. Phase(CPHA) dùng để chỉ cách mà dữ liệu được lấy mẫu theo xung nhịp. Dữ liệu có thể được lấu mẫu cạnh lên của SCK (CPHA=0) hoặc cạnh xuống. Sự kết hợp của SPOL và CPHA làm nên 4 chế đọ hoạt động của SPI.

## Linh kiện được sử dụng trong quá trình thực hiện đồ án.

* Kit STM32f429ZITX là vòng vi điều khiển 32 bit của hãng ST Microelectronic, dựa trên lõi ARM Cortex-M4.



* Thông số kỹ thuật.
* Kiến trúc ARM Cortex M4 32 bits với tốc độ tối đa 168 MHZ.
* Bộ nhớ: 2 MB of Flash memory, 256 Kbytes of RAM.
* Chuẩn giao tiếp: I2C, SPI, UART/USART.
* Ngoại vi: ADC, DMA, PWM, WDT, RTC.
* Điện áp hoạt động: 2.4V ~ 3.6V.

## Module NODEMCU ESP8266

* Module wifi được tích hợp nhân xử lý SoC ESP8266 sử dụng trong các sản phẩm IOT và ứng dụng truyền nhận dữ liệu qua wifi.
* Thông số kỹ thuật.
* Tốc độ xử lý: 80 MHZ.
* WiFi: 2.4 GHz hỗ trợ chuẩn 802.11 b/g/n
* Điện áp hoạt động: 3.3V
* Điện áp vào: 5V thông qua cổng USB
* Số chân I/O: 11 (tất cả các chân I/O đều có Interrupt/PWM/I2C/One-wire, trừ chân D0)
* Số chân Analog Input: 1 (điện áp vào tối đa 3.3V)
* Bộ nhớ Flash: 4MB
* Giao tiếp: Cable Micro USB.
* Hỗ trợ bảo mật: WPA/WPA2
* Tích hợp giao thức TCP/IP
* Lập trình trên các ngôn ngữ: C/C++, Micropython, NodeMCU – Lua

## Module RFID RC522 NFC 13.56 MHZ

* Module RFID RC522 sử dụng IC MFRC522 của Phillip dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56mhz, với thiết kế nhỏ gọn, module này là sự lựa cho các ứng dụng về ghi đọc thẻ RFID.
* Thông số kỹ thuật:
* Nguồn: 3.3VDC, 13 - 26mA
* Dòng ở chế độ chờ: 1013mA
* Dòng ở chế độ nghỉ: <80uA
* Tần số sóng mang: 13.56MHz
* Khoảng cách hoạt động: 0～60mm（mifare1 card）
* Giao tiếp: SPI
* Tốc độ truyền dữ liệu: tối đa 10Mbit/s
* Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire
* Kích thước: 40mm × 60mm

# CHƯƠNG 3 THIẾT KẾ VÀ HIỆN THỰC HỆ THỐNG

## Mô hình thiết kế tổng thể của hệ thống

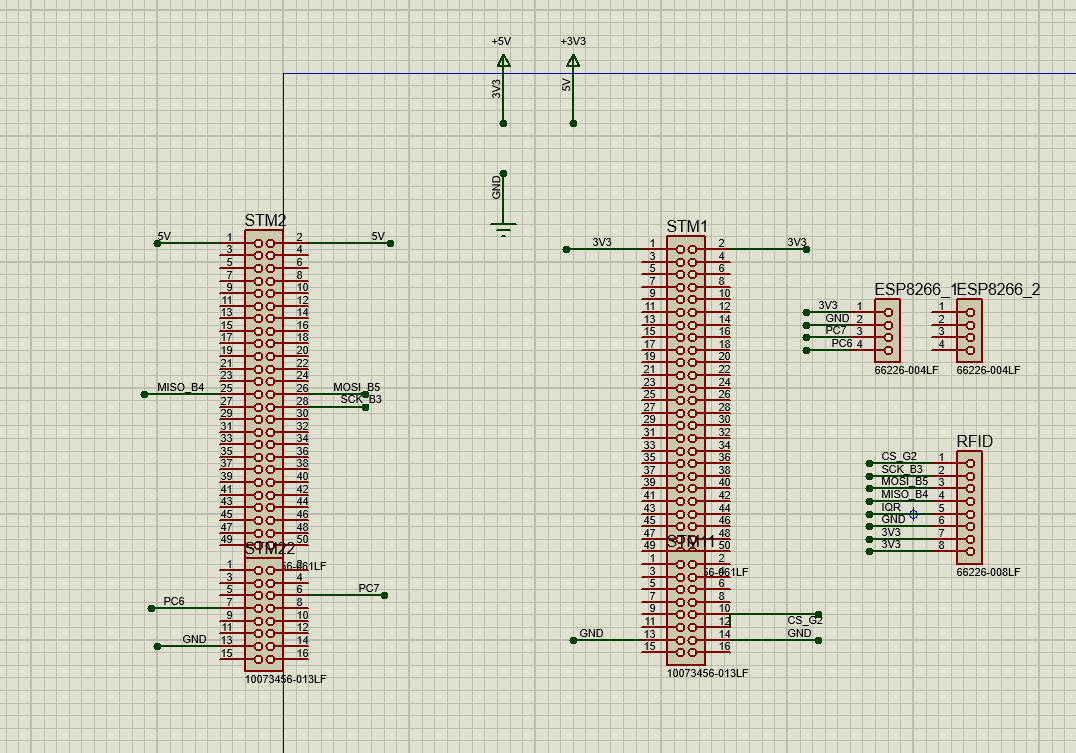
A close up of a logo

Description generated with high confidence

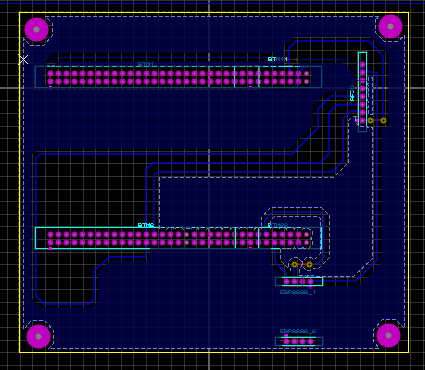
* Hệ thống gồm 4 phần chính: Web client, Server,Database, Node.
* Web client được viết bằng ngôn ngữ HTML, CSS, JavaScript và socketIO. Web client để nhận dữ liệu từ server gửi xuống, hiển thị thông báo số lượng chỗ trống xe, địa chỉ, giá thành mỗi lẫn gửi của bãi đỗ xe.
* Server được viết bằng ngôn ngữ nodejs với framework express và SocketIO. Dùng để nhận dữ liệu từ Node gửi lên, xử lý cái yêu cầu từ Webclient, giao tiếp dữ liệu với database.
* Database sử dụng mongdb: MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở thuộc học NoSQL. Nó được thiết kế theo kiểu hướng đối tượng, các bảng trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ trên bảng không cần tuân theo một cấu trúc nhất định nào cả (điều này rất thích hợp để làm big data). MongoDB lưu trữ dữ liệu theo hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON nên truy vấn sẽ rất nhanh
* Node gồm RFID RCC 522, Kit STM32F429 và Node MCU ESP 8266. STM32F429 thu thập dữ liệu từ đầu đọc RCC 522 thông qua giao thức SPI, sau đó dữ liệu được chuyển qua ESP 8266 bằng giao thức UART. ESP 8266 có nhiệm vụ gửi dữ liệu lên server bằng SocketIO.
* Chức năng chính của hệ thống.
* Thông báo số lượng chỗ trống, giá thành, địa chỉ của bãi đỗ xe.
* Database
* Lưu trữ dữ liệu các bảng gồm: User (id, car\_code, parking\_flag, is\_member, name\_parking\_lot) ; Parking Lot (name, address, amount); Admin (user, pass, email, seasionKey)

## Thiết kế phần cứng

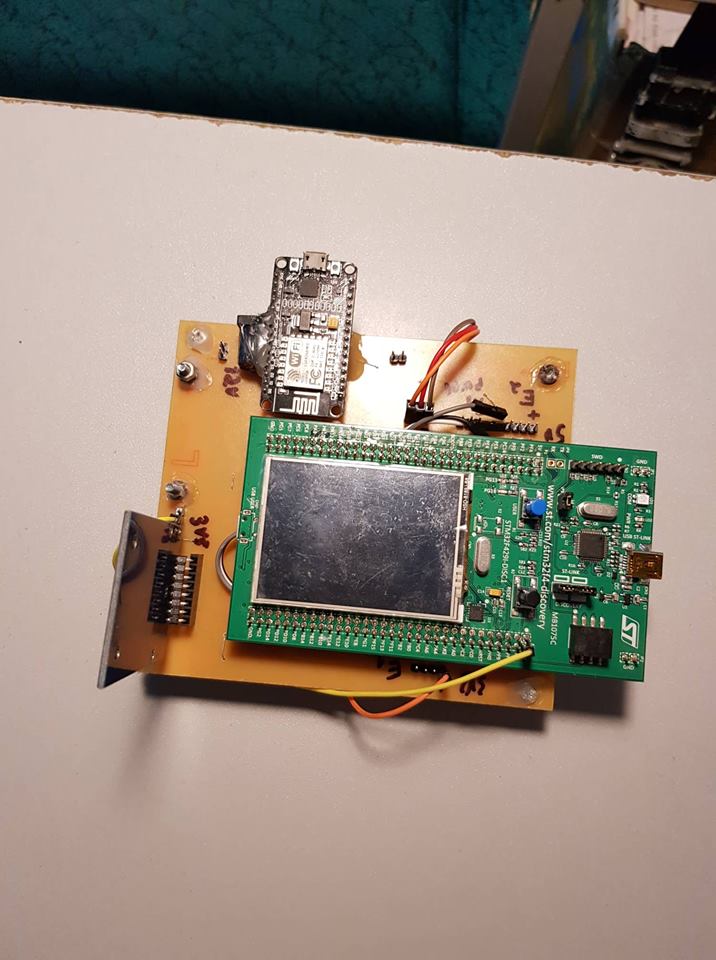
* Mạch bao gồm các thành phần chính:
* Kit STM32F429.
* RFID RCC 522.
* Node MCU ESP8266.
* Thiết kế mạch.



* Thiết kế mạch layout

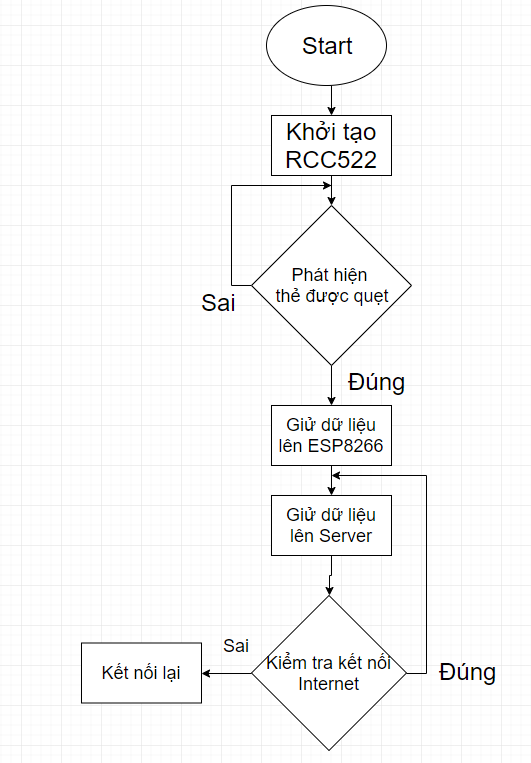


* Gia công PCB hoàn chỉnh.



## Cơ chế hoạt động

* Quá trình đọc thẻ và gửi dữ liệu lên Server
* Khi người dùng quẹt thẻ, hệ thống sẽ đọc thẻ và đồng thời gửi thông tin lên Server.



## Quá trình xử lý ở server

* Hiển thị dữ liệu trên web client: từ server lấy dữ liệu từ database và tạo một sự kiện để clinet nhận được và show lên web.Nhận và xử lý dữ liệu từ node MCU: node MCU sẽ tạo một sự kiện truyền dữ liệu với định dạng json khi thẻ được quẹt, server sẽ nhận sự kiện đó và xử lỹ thông tin sau đó lưu vào database. có 2 sự kiện sign\_member và parked
* sign\_member: đơn giản là lưu thông tin và biển số xe vào database
* parked : kiểm tra sự kiện này là gửi xe hay lấy xe ra, là guest hay là member từ đó có hướng xử lý khác nhau
* Quản lý bãi giữ xe: phát sinh phần này là do xử lý sự kiện guest gửi xe phải nhập vào biển số xe, quản lý bãi xe phải đăng nhập với quyên admin đễ dẫn đén trang nhập biển số xe, khi 1 guest gửi xe thì người quản lý chỉ cần nhập vào biển số trên màn hình là được.

## Quá trình xử lý ở dữ liệu

1. Database

* Bảng user
* id: Id thẻ được gửi từ gateway lên.
* car\_code: dãy chữ, số cho biển số xe được nhập vào từ "form nhập trên giao diện” để xác minh chủ nhân qua giấy tờ xe.
* flag: trả về 1 hoặc 0, mục đích để kiểm tra xe có trong database chưa.
* is\_member: là thành viên nếu giá trị trả về 1, ngược lại là khách.
* Bảng parking\_lot
* name: tên của nhà xe
* address: địa chỉ của nhà xe
* amount: số lượng chỗ còn trống
* Bảng admin
* username: tên admin
* pass: mật khẩu
* email: mail của admin
* seasonKey: Khóa đối xứng, công nghệ mã hóa cho phép mỗi cuộc trao đổi là “1 key khác”

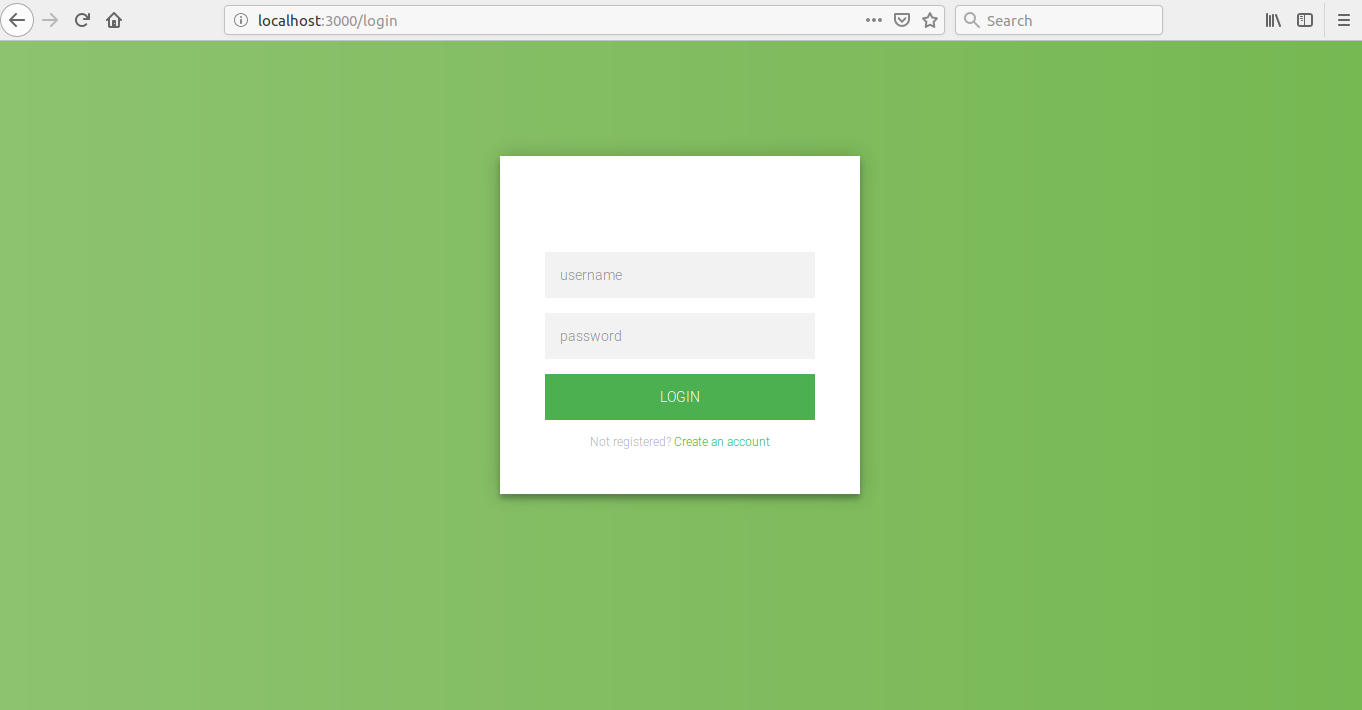
1. Đối tượng: Khách, Thành viên.
2. Dữ liệu cho Gateway:
3. Quét thẻ từ chứa dữ liệu id;
4. Kiểm tra thẻ tồn tại bằng giá trị của id
   1. Không tồn tại: thông báo ra màn hình thẻ không tồn tại (ko thuộc nhà xe này)
   2. Tồn tại: qua bước kiểm tra flag
5. Kiểm tra flag:
   1. Nếu flag = 1 (tức là xe ra): Thay đổi giá trị của flag = 0
      1. Truy xuất thông tin is\_member =1 (là thành viên): Thay đổi giá trị của flag = 0
      2. Truy xuất thông tin is\_member =0 (là khách):
         * Thay đổi giá trị của flag = 0
         * Xóa thông tin của car\_code
   2. Nếu flag = 0 (xe vào):
      1. Tương tự là member: Thay đổi giá trị của flag = 1
      2. Tương tự là Guest:
         * Thông tin biển số xe được lưu vào car\_code
         * Thay đổi giá trị của flag = 1
6. Dữ liệu cho web client:
   * + - Lấy thông tin chỗ trống (tính toán chỗ trống từ giá trị amount), địa chỉ nhà xe (address) từ bảng parking\_lot
7. Dữ liệu cho admin:

* Lưu thông tin các tài khoản được phép quyền Admin truy cập vào tinh chỉnh dữ liệu database.

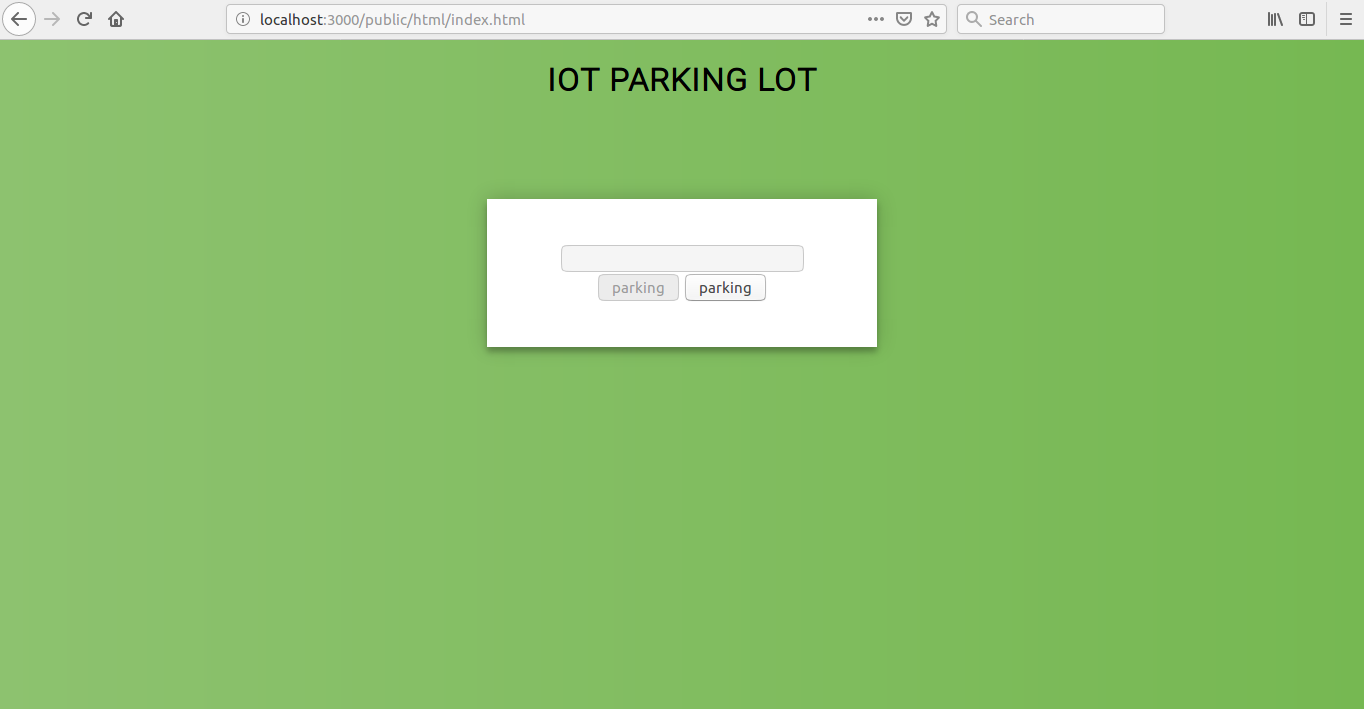
## Quá trình xử lý ở web client

1. Web Client

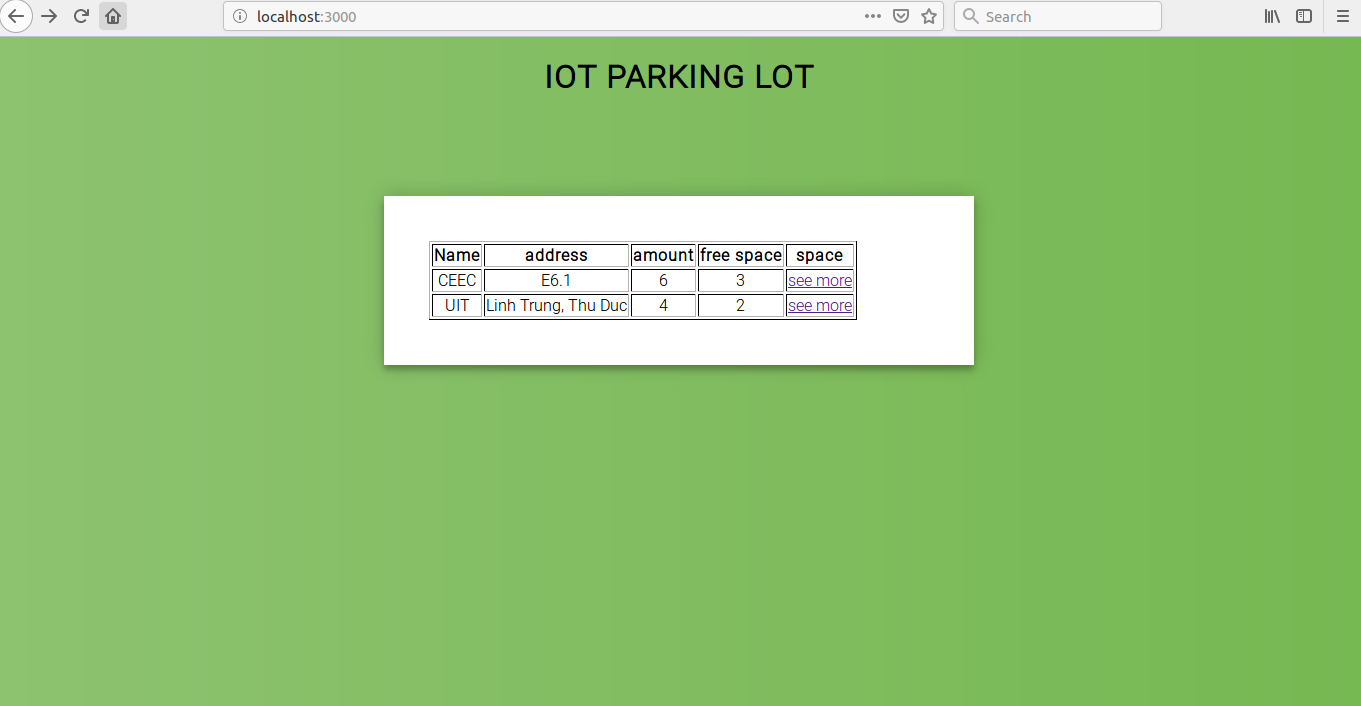
* Người quản lí đăng nhập để xem thông tin bãi đỗ xe.



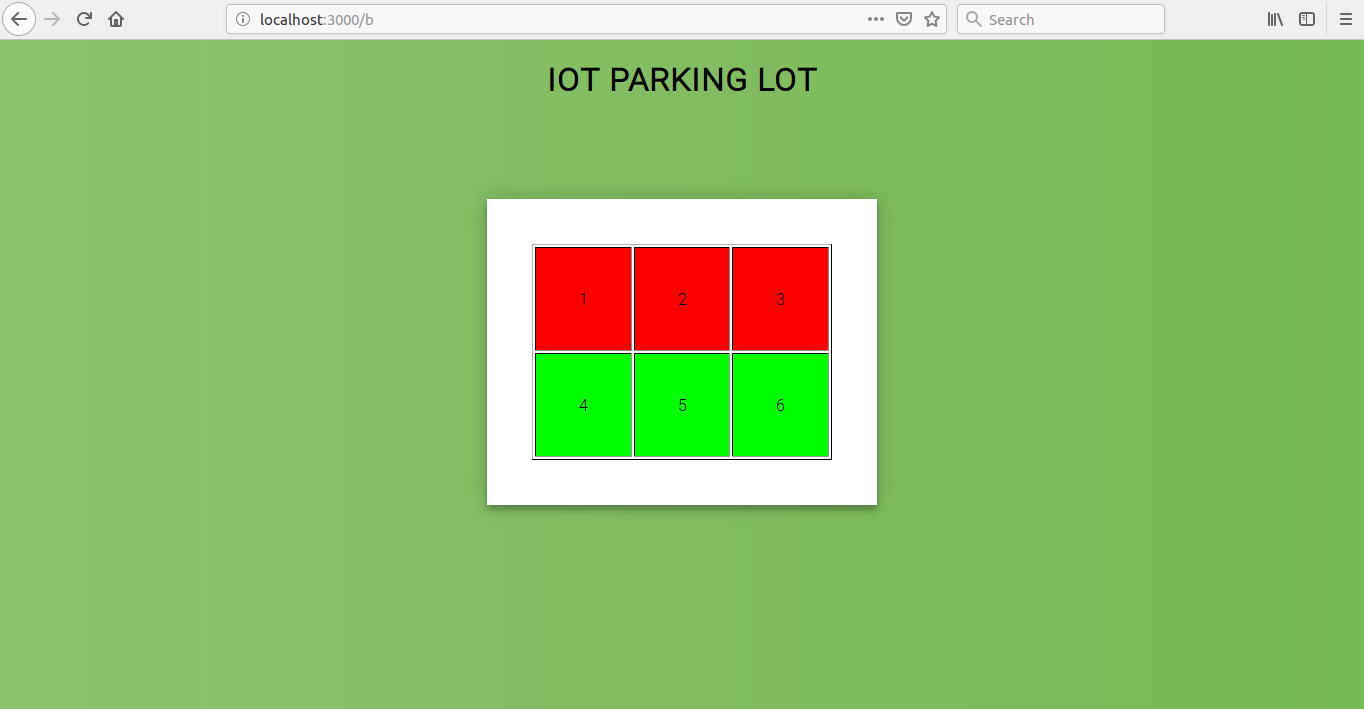
* Người quản lí nhập thông tin số xe



* Thông tin chung các bãi xe và người dùng có thể xem chi tiết hơn của một bãi xe



* Chi tiết từng bãi xe: màu đỏ là vị trí đã có xe đỗ, màu xanh thì chỗ vẫn còn trống



1. Xử lí ở Web Client

* Khi có kết nối được tạo giữa Socket Client và Socket Server thì ở Client sẽ gửi thông tin yêu cầu thông qua biến message để Server thực hiện đúng công việc User yêu cầu (đăng nhập/đăng kí, xem thông tin bãi xe, xem thông tin chi tiết bãi xe)
* Đăng kí/Đăng nhập: Khi có yêu cầu đăng nhập và người quản lí nhập đúng thông tin (đã qua bước kiểm tra về thông tin User nhập vào thỏa các điều kiện của username hay password, định dạng mail) thì Client sẽ gửi thông tin username và password đến Server để kiểm tra thông tin user. Khi có yêu cầu đăng kí thì Client sẽ gửi thông tin username và password đến Server để lưu lại thông tin user.
* Xem thông tin bãi xe: Client gửi yêu cầu lấy thông tin các bãi xe đến Server và đợi các giá trị mà Server trả về sau đó in thông tin ra cho User biết.
* Xem thông tin chi tiết từng bãi xe: Khi click vào see more thì client gửi yêu cầu lấy thông tin của bãi xe muốn xem (mỗi bãi xe thì có mã riêng) và đợi Server gửi về thông tin về nếu như chỗ nào đã có xe đỗ rồi thì sẽ chuyển background thành màu đỏ.

# KẾT QUẢ THỬ NGHIỆM VÀ ĐÁNH GIÁ

* Kết quả
* Xây dựng được hệ thống IOT để quản lý bãi đỗ xe, với chức năng quả lý, thông báo cho người dùng biết số lượng chỗ trống, địa chỉ, giá mỗi lần gửi xe trong các bãi đỗ xe.
* Nhận xét và đánh giá
* Hệ thống tốn ít chi phí hơn so với các hệ thống quản lỹ bãi xe thông thường, tích hợp được hệ thống IOT vào quản lý. Hệ thống mang tính realtime.
* Hướng phát triển
* Tích hợp google map vào hệ thống để người dùng tìm được bãi đỗ xe gần nhất.

# TÀI LIỆU THAM KHẢO

<https://en.wikipedia.org/wiki/MongoDB>

<http://arduino.vn/>

<https://stm32f4-discovery.net/>