**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**

**PARKING MANAGEMENT**

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

NHẬN XÉT CỦA GIẢNG VIÊN

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).

Mục lục

[CHƯƠNG 1 TỔNG QUAN ĐỀ TÀI 3](#_Toc532307030)

[1.1. Hiện trạng 3](#_Toc532307031)

[1.2. Giải pháp 3](#_Toc532307032)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 3](#_Toc532307033)

[2.1. Giao thức UART 4](#_Toc532307034)

[2.1.1 Khái niệm 4](#_Toc532307035)

[2.1.2 Các thông số cơ bản 4](#_Toc532307036)

[2.1.3 Hoạt động 4](#_Toc532307037)

[2.2. Chuẩn SPI 4](#_Toc532307038)

[2.2.1 Khái niệm 4](#_Toc532307039)

[2.2.2 Hoạt động 4](#_Toc532307040)

[2.3. Linh kiện được sử dụng trong quá trình thực hiện đồ án. 4](#_Toc532307041)

[2.4. Module NODEMCU ESP8266 4](#_Toc532307042)

[2.5. Module RFID RC522 NFC 13.56 MHZ 4](#_Toc532307043)

[2.6 Mongodb 4](#_Toc532307044)

[2.6.1 Khái niệm 4](#_Toc532307045)

[2.6.2 Ưu điểm 4](#_Toc532307046)

[2.6.3 Áp dụng Mongodb trong đề tài: 4](#_Toc532307047)

[CHƯƠNG 3 THIẾT KẾ VÀ HIỆN THỰC HỆ THỐNG 4](#_Toc532307048)

[3.1. Mô hình thiết kế tổng thể của hệ thống 4](#_Toc532307049)

[3.2. Thiết kế phần cứng 4](#_Toc532307050)

[3.3. Cơ chế hoạt động 4](#_Toc532307051)

[3.4. Quá trình xử lý ở server 4](#_Toc532307052)

[3.5. Quá trình xử lý ở dữ liệu. 4](#_Toc532307053)

[3.6. Quá trình xử lý ở web client. 4](#_Toc532307054)

# 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

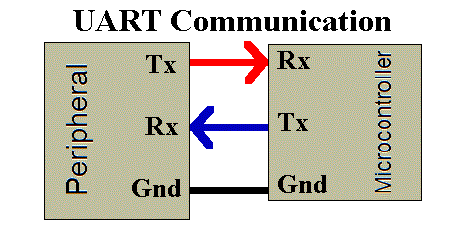
* Xây dựng hệ thống IOT để thông báo cho người dùng biết số lượng chỗ trống của bãi xe và quản lý bãi xe một cách dễ dàng, tiết kiệm chi phí.

# 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.

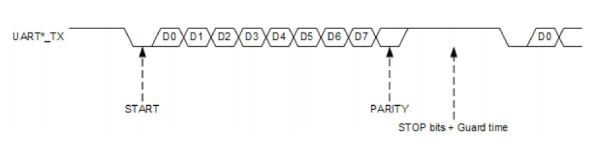


1. Các thông số cơ bản

* 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 giử 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 giử 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 giử đi, sau đó là các bit dữ liệu và kết thúc quá trình truyền là bit STOP bit.

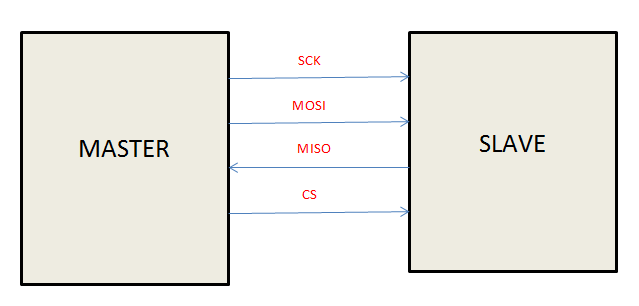


* Khi ở trạng thái chờ mức điện thế ở mức 1 (hight).
* Khi bắt đầu truyền, START bit sẽ chuyển từ 1 xuống 0 để báo hiệu cho bộ nhận là quá trình truyền dữ liệu sắp xảy ra.
* Sau START bit là đến các bit dữ liệ D0-D7( các bit này có thể mức Hight hay Low tùy theo dữ liệu).
* Cuối cùng là STOP bit là bảo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

## Chuẩn SPI

### Khái niệm

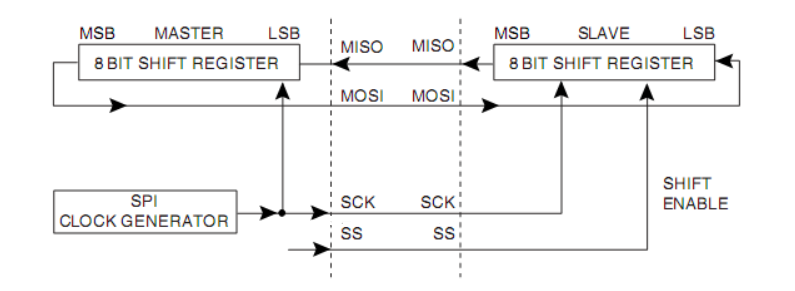
* SPI( Serial Peripheral Bus) được phát triển bởi Motorola. Đây là một chuẩn đồng bộ nối tiếp để truyền dữ liệu ở chế độ song công toàn phần( full-duplex) tức trong cùng một thời điếm có thể xảy ra đòng thời quá trình truyền và nhận. Đôi khi SPI còn được gọi là chuẩn giao tiếp 4 dây(Four-wire).



* SPI là giao tiếp đồng bộ, bất cứ quá trình truyền nòa 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

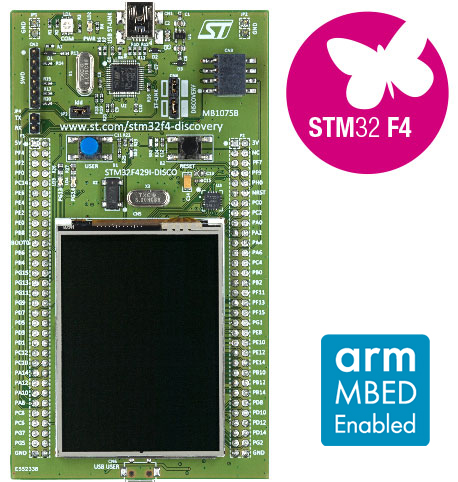
* + Mỗi chip Master hay Slave có thành ghi dữ liệu 8 bits. Cứ mỗi xung nhịp do Master tạo ra trên đường giữ nhịp SCK, một bit trong thanh ghi dữ liệu của Master đực truyền trên đường MOSI, đồng thời một bit trong thnah ghi dữ liệu của chip Slave cũng truyền qua Master trên đường MISO. Do 2 gói dữ liệu trên 2 chip giử 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

1. Mongodb
2. Khái niệm

* 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.

1. Ưu điểm

* Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau.
* Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng. Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
* MongoDB rất dễ mở rộng.
* Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB.
* MongoDB phù hợp cho các ứng dụng realtime.

1. Áp dụng Mongodb trong đề tài:

# CHƯƠNG 3 THIẾT KẾ VÀ HIỆN THỰC HỆ THỐNG

## Mô hình thiết kế tổng thể của hệ thống

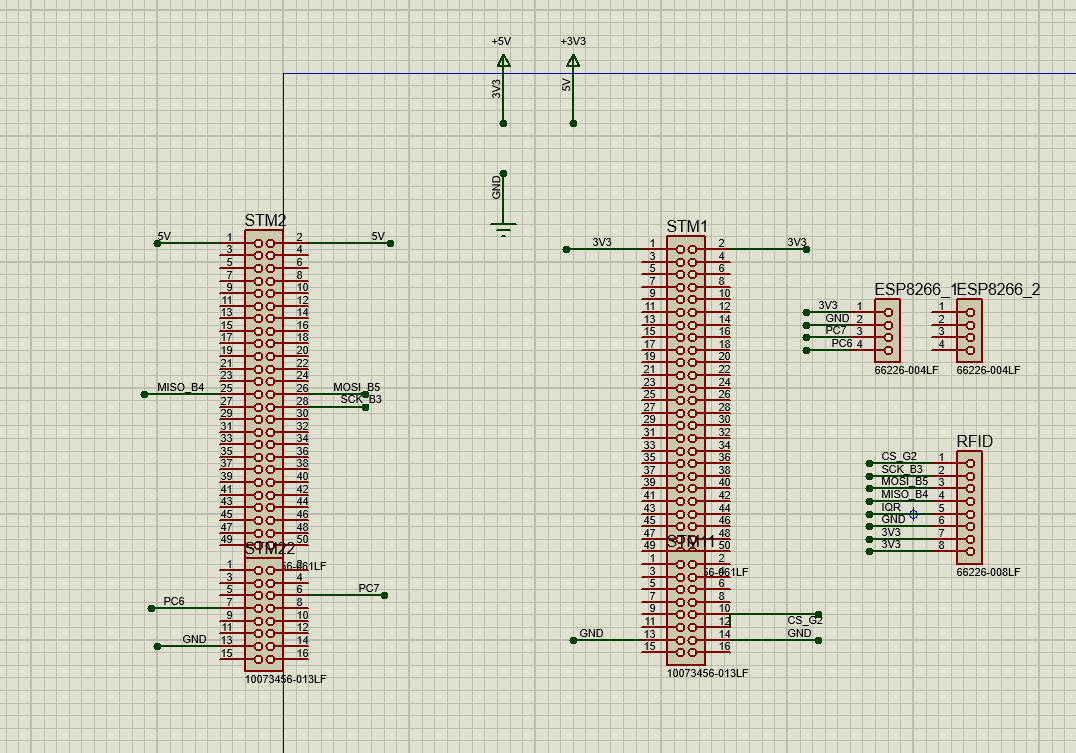
A close up of a logo

Description generated with high confidence

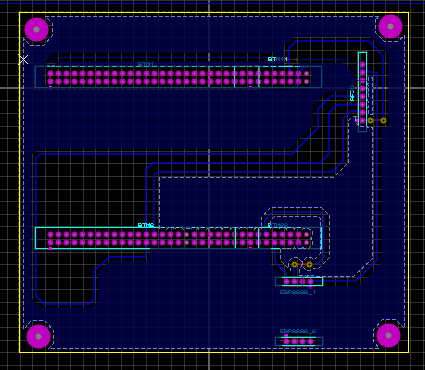
* Hệ thống gồm 4 phần chính: Web client, Server,Database, Node.
* Web client được viết bằng ngôn ngữ HTML, CSS, JavaScript và socketIO. Web client để nhận dữ liệu từ server giử xuống, hiển thị thông báo số lượng chỗ trống xe, địa chỉ, giá thành mỗi lẫn giử của bãi đỗ xe.
* Server được viết bằng ngôn 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 ( Bạn Anh viết).
* 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 giử 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.
* Viết tiếp chức năng của database đây nhá Anh

## Thiết kế phần cứng

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



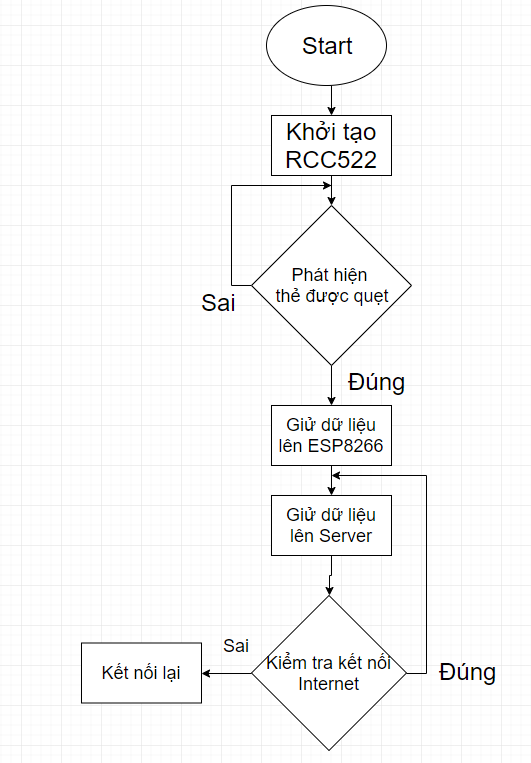
* Thiết kế mạch layout



* 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 giử thông tin lên Server.



## Quá trình xử lý ở server

## Quá trình xử lý ở dữ liệu

1. Database

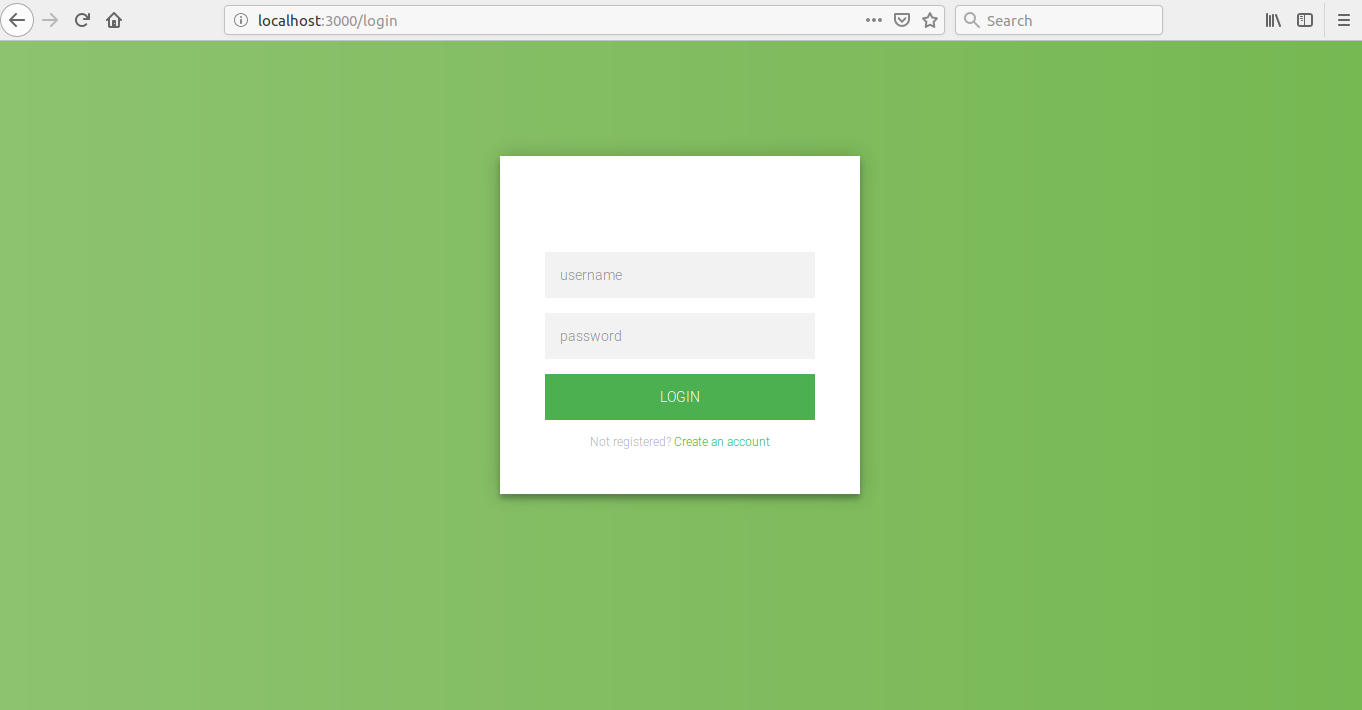
* id: Id thẻ bất kì
* car\_code: Biển số xe để xác minh chủ nhân qua giấy tờ xe
* flag: mục đích để kiểm tra xe có trong database chưa,
* is\_member: member luôn có chỗ,

1. Đối tượng: Guest, Member.
2. Quy trình xử lý Gateway:
3. Quét thẻ từ (chỉ chứa id)
4. Kiểm tra thẻ tồn tại bằng id
   1. Không: thông báo ra màn hình thẻ không tồn tại (ko thuộc nhà xe này)
   2. Có: qua B3
5. Kiểm tra Flag:
   1. If Flag == 1 (xe ra): Flag = 0
      1. Là Member: Flag = 0
      2. Là Guest:
         1. Flag = 0
         2. Xóa thông tin cột car\_code
   2. If Flag == 0 (xe vào):
      1. Là member: Flag = 1
      2. Là Guest:
         1. Hiển thị nhập car\_code
         2. Flag = 1
6. Quy trình xử lý End Devide:
   * + 1. Lấy thông tin chỗ trống cho Guest: Hàm get\_car\_info trả về số chỗ trống

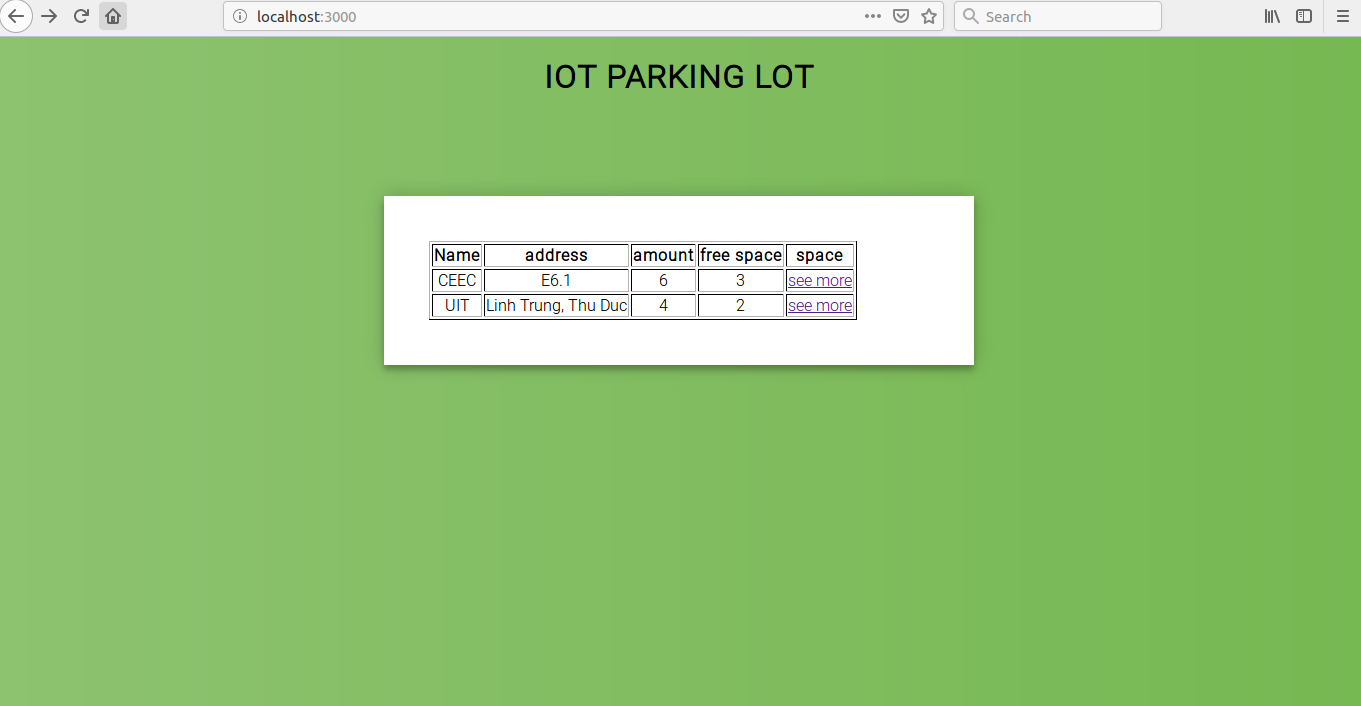
## Quá trình xử lý ở web client

1. Web Client

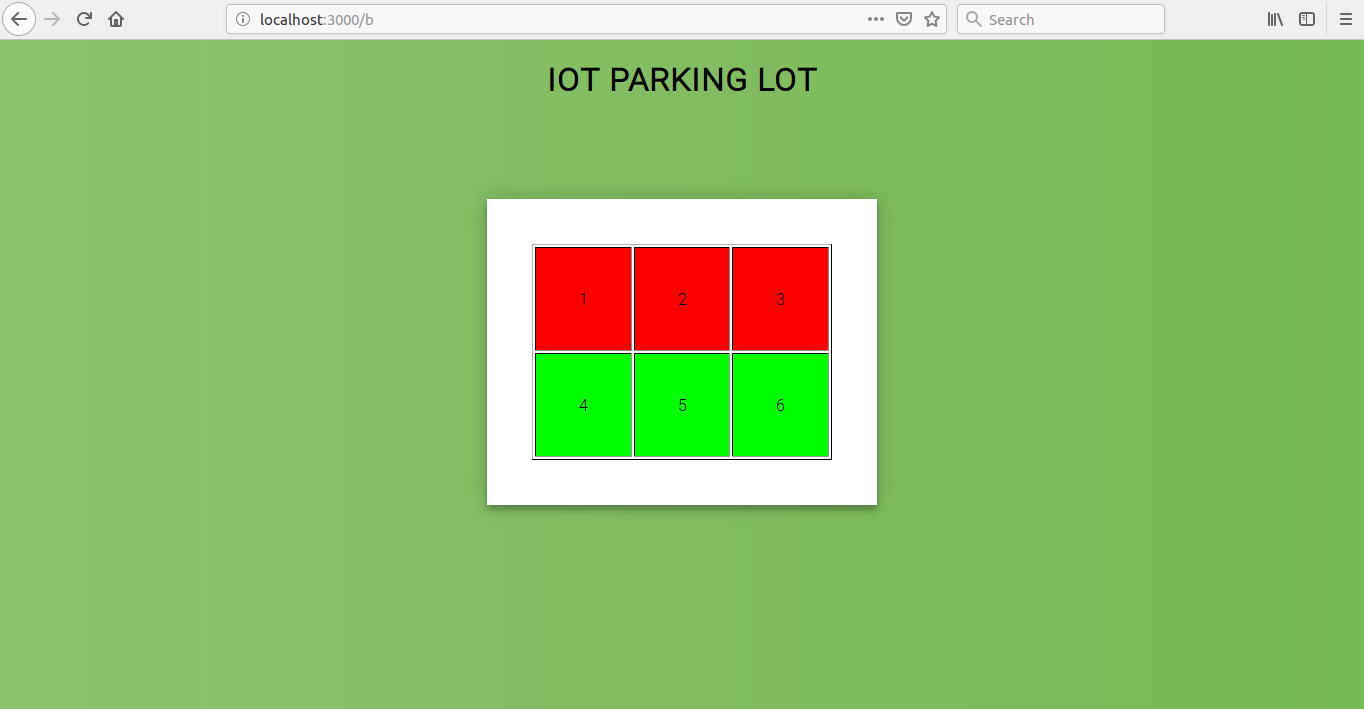
* Người dùng đăng nhập vào account của mình để xem được thông tin bãi đỗ xe hoặc có thể đăng kí thông tin để xem bãi đỗ xe.



* Thông tin các bãi xe, có thể xem chi tiết hơn của một bãi xe khi click see more



* 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 User nhập hết 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.

* Lắp ráp hệ thống.
* 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.