INTERNET OF THINGS

REPORT ASSIGNMENT 2

STUDENT NAME: TRAN QUANG HUY

STUDENT ID: GCD18457

Class: 0602

ASSIGNMENT 2 FRONT SHEET

Qualification	**TEC Level 5 HND Diploma in Computing**		
Unit number and title	**Unit 43: Internet of Things**		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name	Tran Quang Huy	**Student ID**	GCD18457
Class	GCD0602	**Assessor name**	Tran Trong Minh

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	Huy

Grading grid

P5	P6	P7	M5	M6	D2	D3

❒ Summative Feedback:		❒ Resubmission Feedback:
Grade:	Assessor Signature:	Date:
Internal Verifier's Comments:		
Signature & Date:		

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURES

TABLE OF PICTURES

INTRODUCTION

Parking is going through a disruptive shift as technology enables new ways of maximizing utilization of parking spaces, and drivers become more demanding of using apps and other services to find available spaces.

Parking is a generator of revenue for the city, facilitates economic growth and is a factor in improving the quality of life in many cities. Giving drivers access to parking spots in locations near to where they live, work and undertake leisure activities is crucial to ensuring that a cities' economic prosperity thrives.

Many of the issues cities experience with pollution and traffic can be traced directly back to poor management of parking utilization:

- Vehicles circle city streets looking for parking
- New vehicles enter cities even when parking facilities are full
- Vehicles queue waiting to enter and leave car parks

This additional movement creates traffic and increases vehicle miles traveled – which means more pollutants are generated. The indirect effect of this is wasted time and money. Businesses suffer as people cannot reach them, and productivity is affected by people being late for work, school and other appointments. By managing parking demand and supply in a more effective manner, these issues can be reduced or removed entirely.

Part I. The an IoT application using any combination of hardware, software, data, platforms, and services

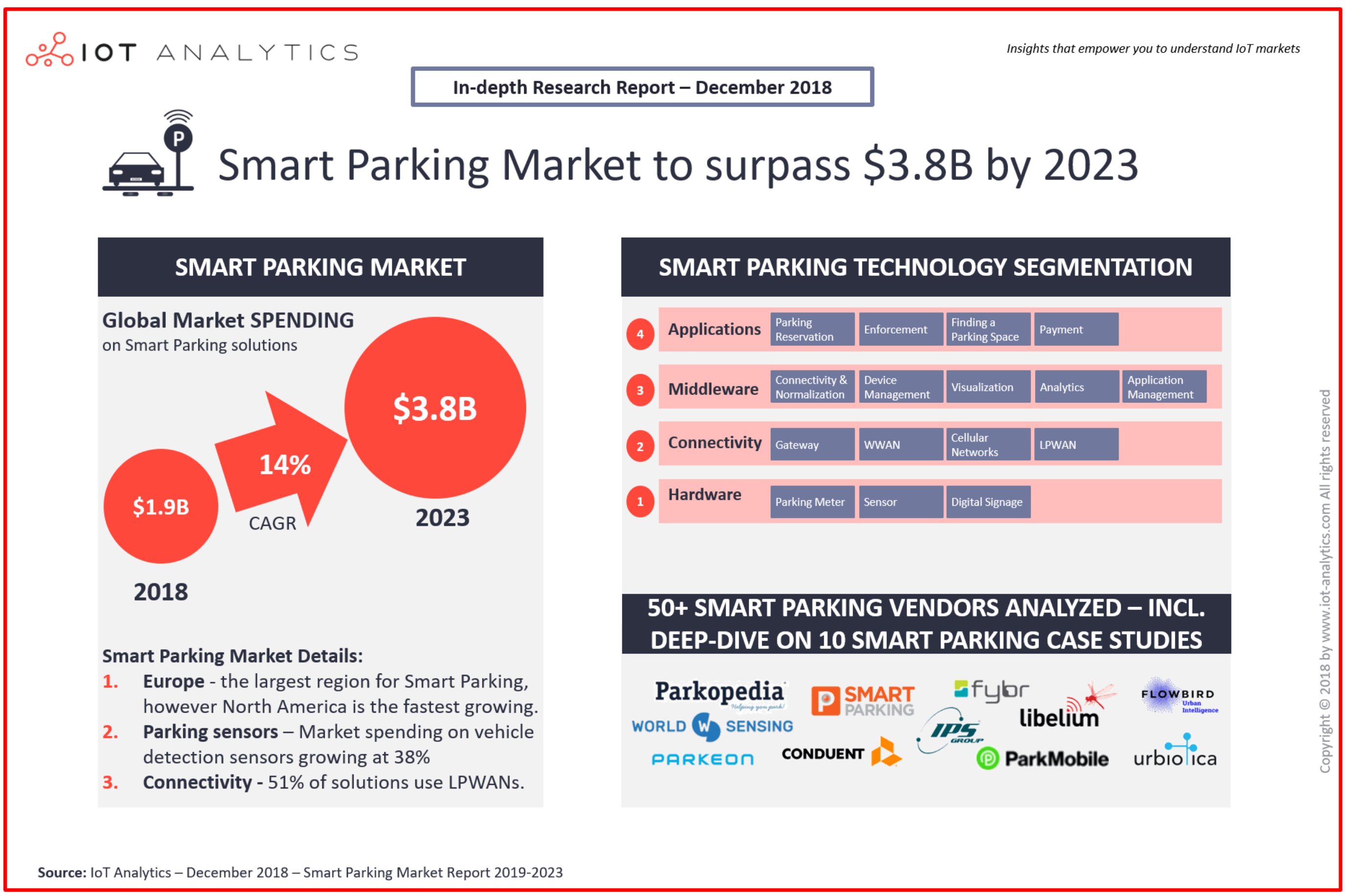
1. The appropriate set of tools to develop the plan into an IoT application

1.1. The problem

Parking problems are not uncommon, especially for big cities. By 2023, market spending for smart parking products and services is expected to grow at a CAGR of 14% and surpass $3.8B according to an IoT Analytics report. The growth of market spending is good news because it will force people to try to find a solution to these traffic problems instead of taking no action.

The problem is quite obvious. But a bigger question is what can technology do to solve it? Is there any way to solve the problem?

With the growing popularity of Smart Cities, there is always a demand for smart solutions for every domain. The IoT has enabled the possibility of Smart Cities with it's over the internet control feature. A person can control the devices installed in his home or office from anywhere in the world by just using a smartphone or any internet-connected devices. There are multiple domains in a smart city and Smart Parking is one of the popular domains in the Smart City.



Picture 1. The Smart Parking Market Report 2019 forecasts $3.8B spending on Smart Parking Solutions by 2023. (iot-analytics., n.d.)

Static traffic is becoming a problem for big cities in Vietnam today. Public car parks and spots cannot be said to meet the parking needs of many people because the number of vehicles exceeds the demand.

Picture 2. Parking in Da Nang city.

In Da Nang city, the key economic sector is tourism. However, according to the statistics in this city, there are already more than **70,000 individual cars** and **tourist cars**, **taxis** and more than **800,000 motorbikes**, but there is still no official parking. scale cars will go into operation in a total of nearly 200 parking lots will build according to the planning of Da Nang city. Consequently, vehicles parked on the road causing traffic congestion and endangering pedestrians.

It may take a lot of time for the driver to locate the parking location. This affects the work and study of people. Sometimes this also affects the security and order of the city and the parking of a car can cause traffic jams.

Picture 3. Traffic jams at Da Nang city.

1.2. The solution

The Smart Parking industry has seen a number of innovations such as Smart Parking Management System, Smart Gate Control, Smart Cameras which can detect types of vehicles, ANPR (Automatic Number Plate Recognition), Smart Payment System, Smart Entry System and many more. Today similar approach will be followed and a smart parking solution will be built which will IR sensor to detect vehicle presence and trigger the gate to open or close automatically. The ESP8266 NodeMCU will be used here as the main controller to control all the peripherals attached to it.

Parking is an area where IoT sensor technology is making a tremendous difference – both for the user experience as well as for parking space management and enforcement. Whether you look at Ho Chi Minh City, Ha Noi or Da Nang city, most major cities especially in Viet Nam, have started to deploy various new Smart Parking Solutions in the last 5 years. Most of these installations now use a combination of vehicle detection sensors, pay-by-app options and in some cases navigation assistance. Our research shows that the penetration rate of these smart parking spaces is still quite low but solutions are popping up in most major cities now – another proof of how pervasive and important IoT technology has become.

Picture 4. GreeParking will solve the problem with a parking car. (xedoisong, n.d.)

With the **GreeParking** system, it is possible to partially solve the above parking problem by applying IoT.

Picture 5. GreeParking logo

The system will notify and show the driver the status of parking on the road where the driver wants to park so that they can make timely decisions to save more time.

Picture 6. Location place to parking using GreeParking

In this IoT Smart Parking System, we will send data to the server for looking up the availability of space for vehicle parking.

1.3. Schematic

As the prototype there are six infrared sensors were connected with the Arduino pins 4 to 9. The infrared sensor VCC pins are connected with the Arduino’s 5v. Grounds are connected with the Arduino’s Ground while the out pins of all the infrared sensors are connected with pin 4 to 9.

The Nodemcu module tx and Rx pins are connected with pin 2 and pin 3 of the Arduino. while the Vin pin of the Nodemcu module is connected with the output of the voltage regulator. This is a regulated 5v power supply based on the lm7805 voltage regulator.

In this system will have 2 Schematic such as main operation (GreeParking) and extend (Gate) for the system:

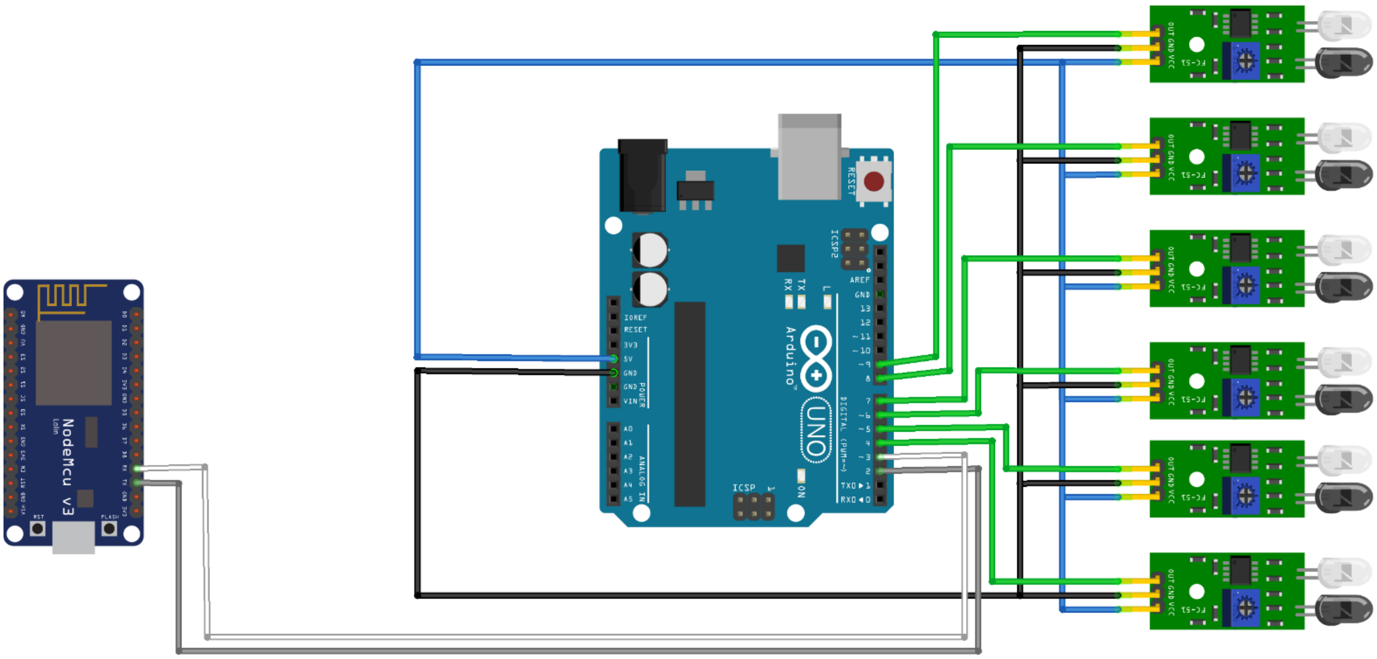


Figure 1. Visualize connection of GreeParking

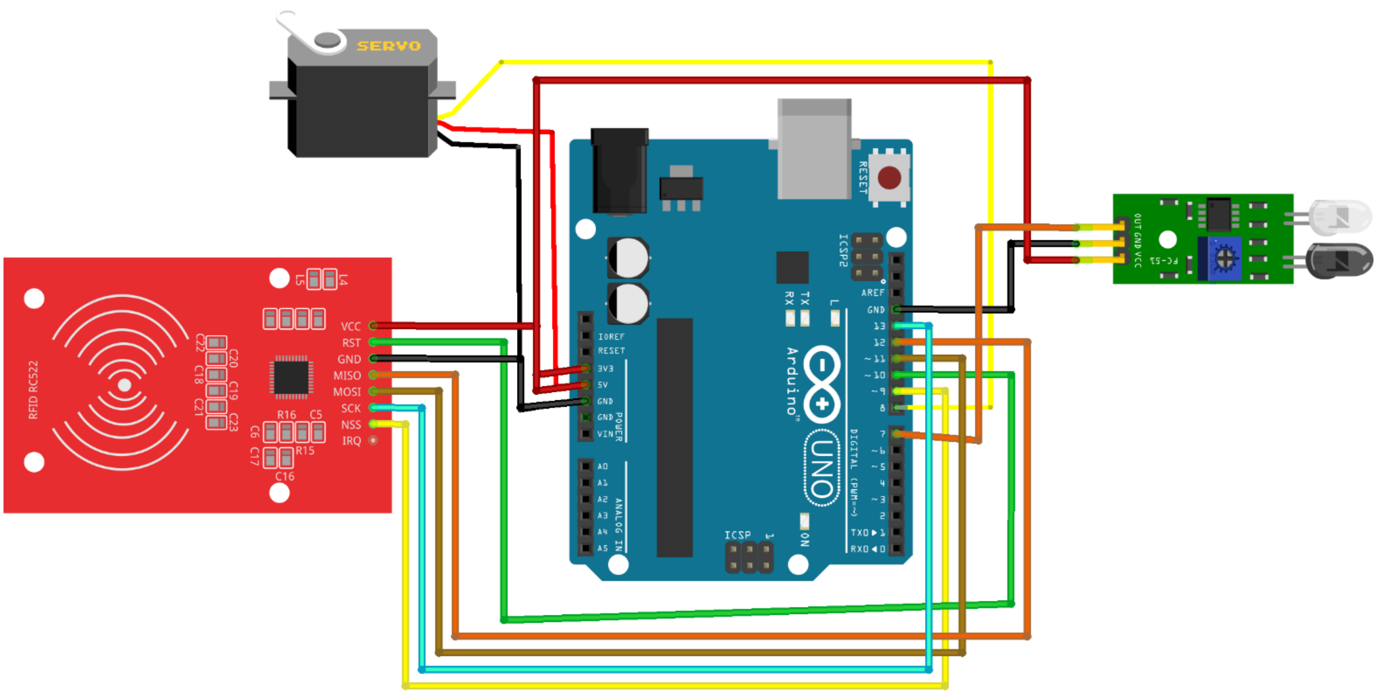


Figure 2. Visualize connection of Gate

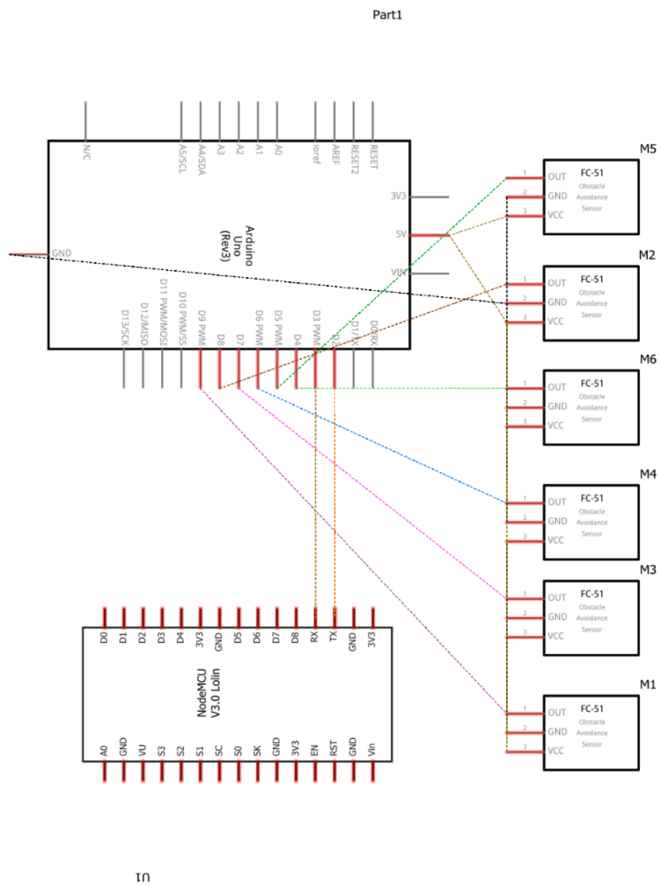


Figure 3. Schematic of GreeParking

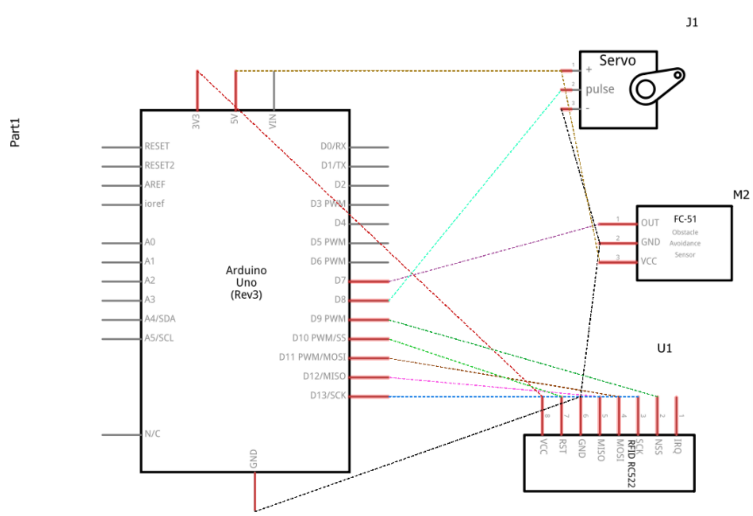


Figure 4. Schematic of Gate

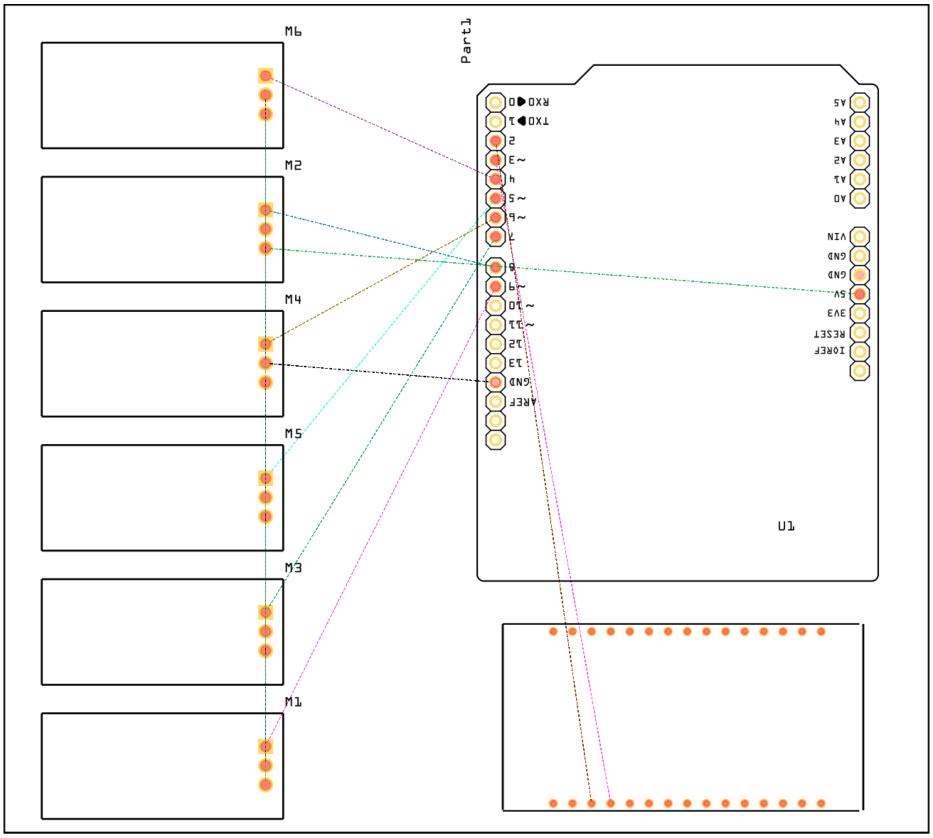


Figure 5. PCB of Greeparking

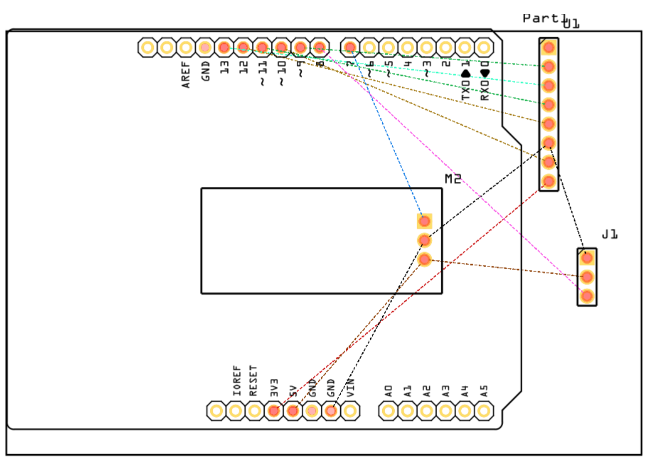


Figure 6. PCB of gate

1.4. IoT Hardware, frameworks, tools

IoT Hardware for prototype:

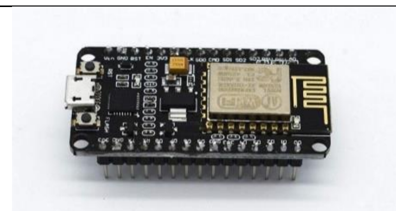
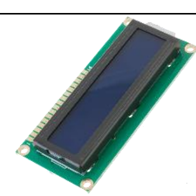
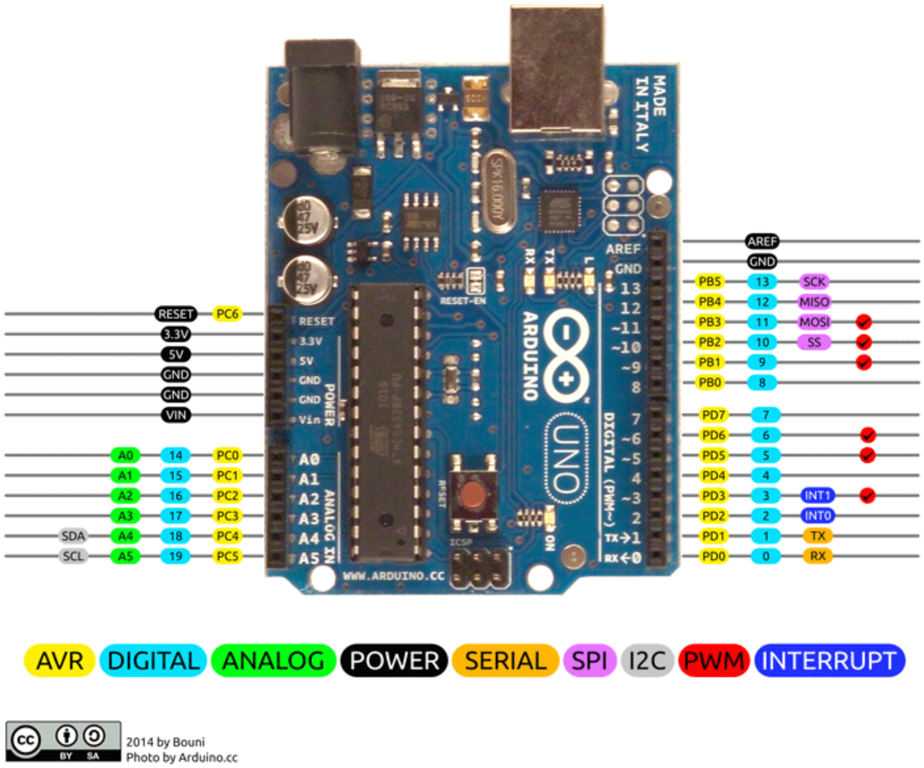
Item	Description
Arduino Uno R3	
Infrared sensor	
Nodemcu esp8266 wifi module	
Servo SG90	
RFID RC522	
LCD I2C 16x2	

Table 1. IoT Hardware

Arduino Uno R3:



Picture 7. Arduino Uno Pin Diagram. (components101, n.d.)

- How to use Arduino Board:

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in Arduino programming. Each pin operates at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50K Ohms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- Serial Pins 0 (Rx) and 1 (Tx): Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- External Interrupt Pins 2 and 3: These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM Pins 3, 5, 6, 9 and 11: These pins provide an 8-bit PWM output by using analogWrite() function.
- SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK): These pins are used for SPI communication.
- In-built LED Pin 13: This pin is connected with a built-in LED when pin 13 is HIGH – LED is on and when pin 13 is LOW, it's off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference() function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- AREF: Used to provide reference voltage for analog inputs with analogReference() function.
- Reset Pin: Making this pin LOW, resets the microcontroller.

- **Pin Description:**

Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by the onboard voltage regulator. The maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output Pins	Digital Pins 0 - 13	It can be used as input or output pins.
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

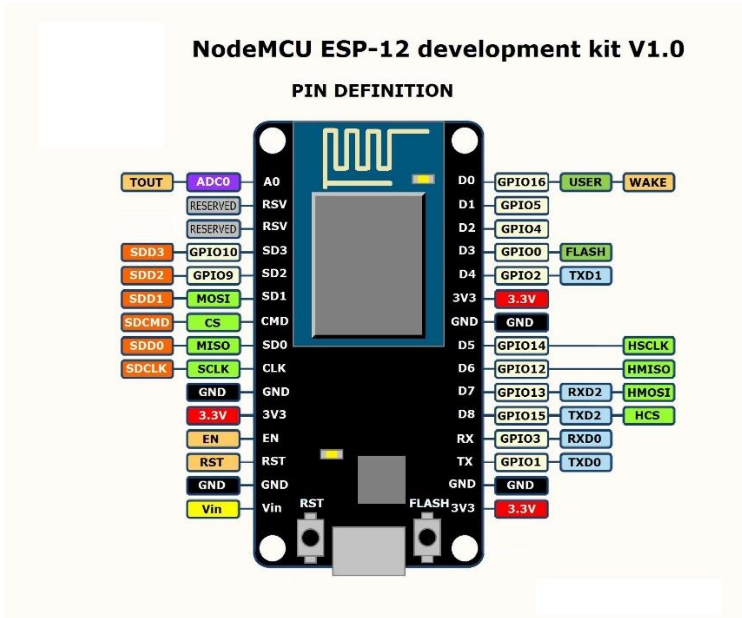
Table 2. Pin description. (components101, n.d.)

- **Arduino Uno Technical Specifications:**

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage	7-12V
Input Voltage Limits	6-20V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)
DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz

Table 3. Arduino Uno Technical Specifications (components101, n.d.)

Nodemcu esp8266 wifi module:



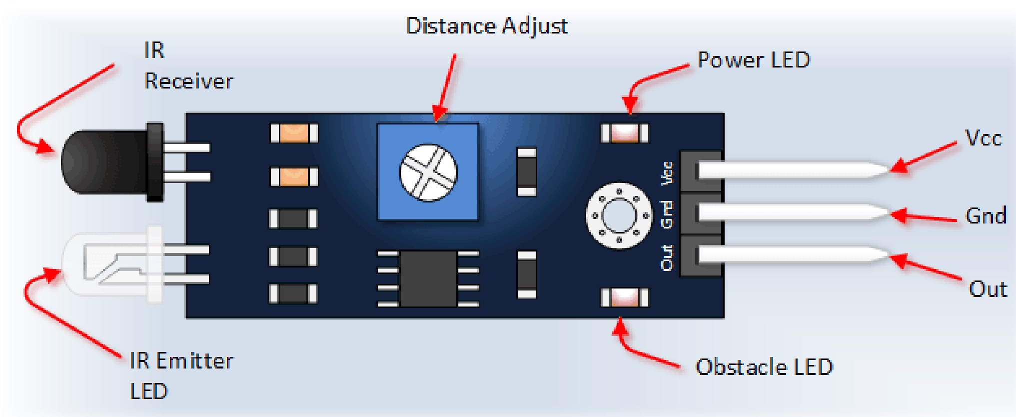
Picture 8. NodeMCU Esp8266 pin diagram. (iotmaker, n.d.)

Categories	Items	Values
Wi-Fi Parameters	Certificates	FCC/CE/ROSH
	Wi-Fi protocols	802.11 b/g/n
	Frequency Range	2.4GHz-2.5GHz (2400M-2483.5M)
Hardware Parameters	Peripheral Bus	UART/HSPI/I2C/I2S/Ir Remote Control
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	16mm*24mm*3mm
	External Interface	N/A
Software Parameters	Wi-Fi mode	station/ softAP/ SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network) / download and write firmware via the host
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Table 4. NodeMCU Parameters. (elecrow, n.d.)

- When performing INPUT and OUTPUT tests on the pins, we obtained the following results:
  - digitalWrite did NOT work with GPIOs 6, 7, 8, 11, and ADC (A0)
  - digitalRead did NOT work with GPIOs 1, 3, 6, 7, 8, 11, and the ADC (A0)
  - analogWrite did NOT work with GPIOs 6, 7, 8, 11, and ADC (A0) (GPIOs 4, 12, 14, 15 have hardware PWM, and the others are by software)
  - analogRead worked only with the ADC (A0)
  - 6, 7, 8, 11 do NOT work for the above four commands

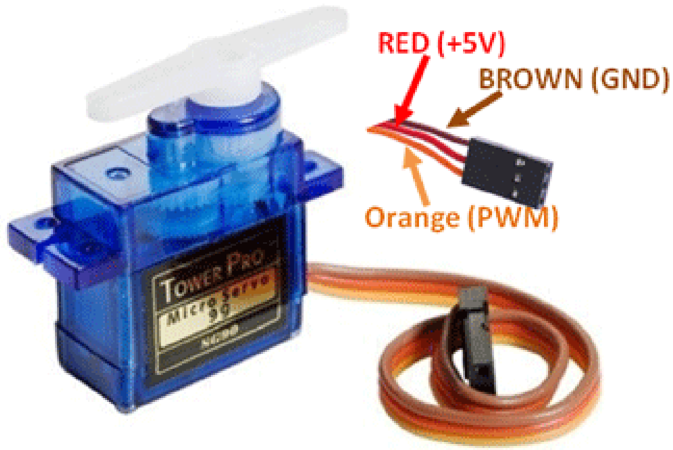
Infrared sensor:



Picture 9. Infrared sensor

- Specifications:
  - The comparator uses LM393, working stably
  - Working voltage: 3.3V - 5V DC.
  - When the power is turned on, the red power indicator lights up.
  - Screw holes 3 mm, easy to fix and install.
  - Size: 3.2cm * 1.4cm
  - The module has been compared the threshold voltage through potentiometer, if used in normal mode, please do not arbitrarily adjust the potentiometer.
- Features:
  - High Reliability
  - Excessive radiant intensity
  - Forward voltage is low
  - Having lead spacing of 2.54mm
  - Maximum wavelength is 940nm
  - Pb free
  - RoHS certified
  - Easy to use with breadboard or perf board
  - Package type is T-1 3/4

Servo SG90:



Picture 10. Servo SG90

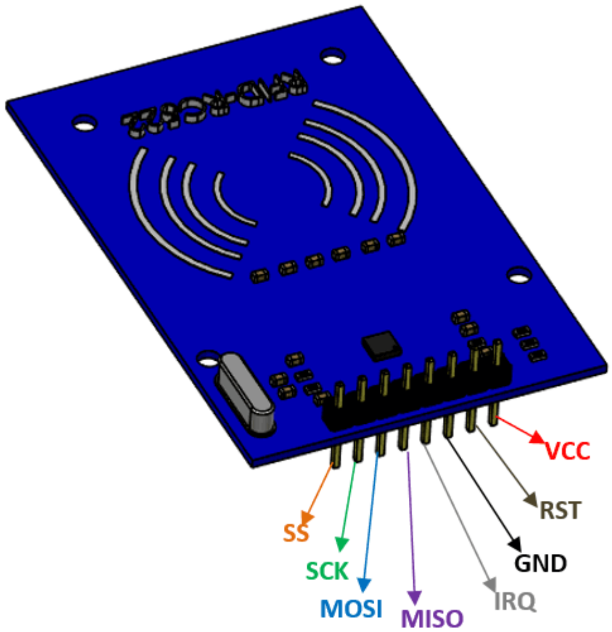
- Wire Configuration:

Wire Number	Wire Colour	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

Table 5. Wire configuration

- Features:
  - Operating Voltage is +5V typically
  - Torque: 2.5kg/cm
  - Operating speed is 0.1s/60°
  - Gear Type: Plastic
  - Rotation: 0°-180°
  - Weight of motor: 9gm
  - Package includes gear horns and screws

RFID RC522:



Picture 11. RFID RC522. (components101, n.d.)

The RC522 is a 13.56MHz RFID module that is based on the MFRC522 controller from NXP semiconductors. The module can support I2C, SPI, and UART and normally is shipped with an RFID card and key fob. It is commonly used in attendance systems and another person/object identification applications.

- RC522 Pin Configuration:

Pin Number	Pin Name	Description
1	VCC	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/Tx	MISO pin when used for SPI communication, acts as SCL for I2c and Tx for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

Table 6. RC522 Pin Configuration

- Features:
  - 13.56MHz RFID module
  - Operating voltage: 2.5V to 3.3V
  - Communication: SPI, I2C protocol, UART
  - Maximum Data Rate: 10Mbps
  - Read Range: 5cm
  - Current Consumption: 13-26mA
  - Power down mode consumption: 10uA (min)

LCD I2C 16x2:

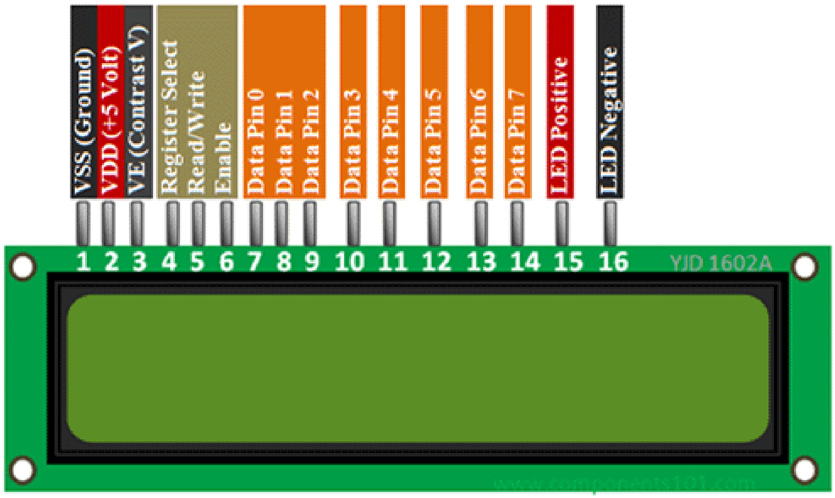


Table 7. 16x2 LCD. (components101, n.d.)

- Pin Configuration:

Pin No:	Pin Name:	Description
1	Vss (Ground)	Ground pin connected to system ground
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V)
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shift between command/data register
5	Read/Write	Used to read or write data. Normally grounded to write data to LCD
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgment

7	Data Pin 0	Data pins 0 to 7 forms an 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.
8	Data Pin 1	
9	Data Pin 2	
10	Data Pin 3	
11	Data Pin 4	
12	Data Pin 5	
13	Data Pin 6	
14	Data Pin 7	
15	LED Positive	Backlight LED pin positive terminal
16	LED Negative	Backlight LED pin negative terminal

- Features:
  - Operating Voltage is 4.7V to 5.3V
  - Current consumption is 1mA without backlight
  - Alphanumeric LCD display module, meaning can display alphabets and numbers
  - Consists of two rows and each row can print 16 characters.
  - Each character is built by a 5×8-pixel box
  - Can work on both 8-bit and 4-bit mode
  - It can also display any custom generated characters
  - Available in Green and Blue Backlight

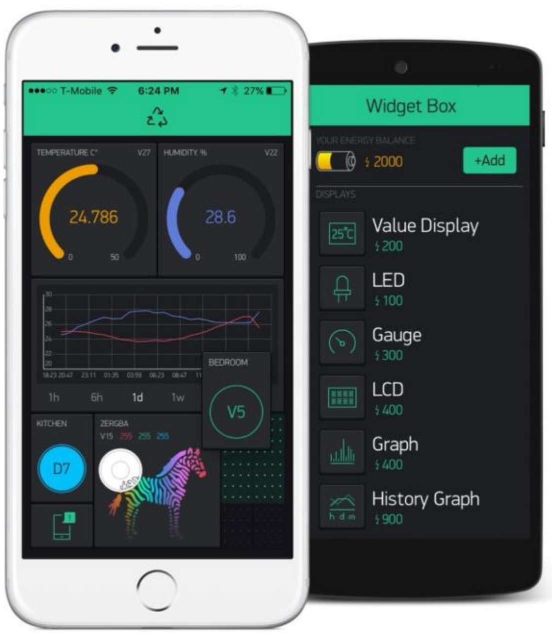
Blynk Platform

Blynk is an Internet-of-Things platform designed to make the development and implementation of smart IoT devices quick and easy. It can be used to read, store, and visualize sensor data and control hardware remotely.

Internet of Things has been all the buzz lately and more and more devices are being talked to the internet every day. With the rise of such amazing technology, the risk of security has also increased substantially. Some of the major concerns in IoT are:

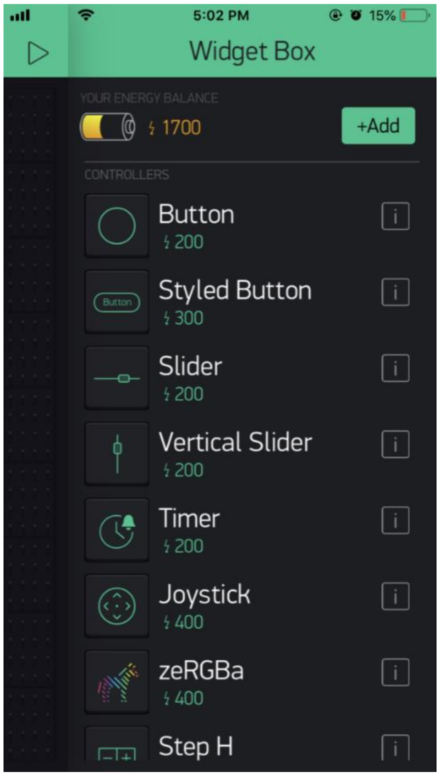
- If IoT devices are sending your data to the internet, the communication needs to be closed and encrypted which cannot be possible without using a dedicated and closed server which is really hard to manage.
- The IoT devices also need to be responsive and again, that is not possible without a server with low latency and high responsiveness.
- In IoT, the platform needs to be compatible with many different types of hardware architecture and devices, so that it doesn't restrict its users with a single type of hardware with limited capabilities.

Keeping in view the problems mentioned above, Blynk is the perfect solution for all these problems. Blynk consists of the following three major components:



Picture 12. Blink application

Blynk App: The mobile app developed by Blynk works as a control panel for visualizing and controlling your hardware. It is available for both Android and iOS. The app offers a very productive interface and various different widgets for different purposes. Blynk works on a currency of its own called energy. New users get 2000 amount of Blynk energy with a free Blynk account and this energy is used to buy and deploy widgets in the projects.



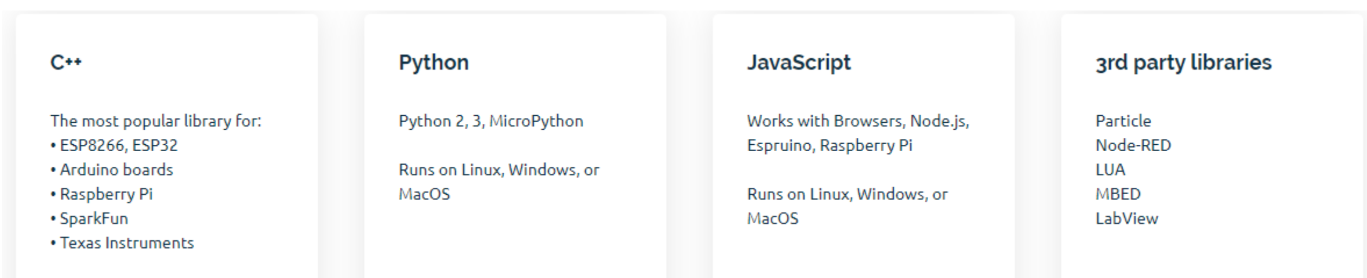
Picture 13. Widget Box on blink app

Blynk Server: The most amazing component of the Blynk Platform which makes it all possible is the Blynk Server. Blynk offers a secure, responsive and centralized cloud service through its server which allows all of this communication between the devices. The Blynk server is also available as open-source so you can literally make your own server and make it even more secure with a little tinkering**.**

Blynk Library: The key feature of Blynk platform which makes it scalable and amazing, is the Blynk Library. The Blynk library makes it possible to connect your hardware and get it up and running in a blink. The support for multiple hardware devices including Arduino, ESP8266 and Raspberry Pi is included in the library and it also makes it possible to connect with hardware through many different ways of communication like Wi-Fi, Bluetooth, BLE, USB and GSM**.**

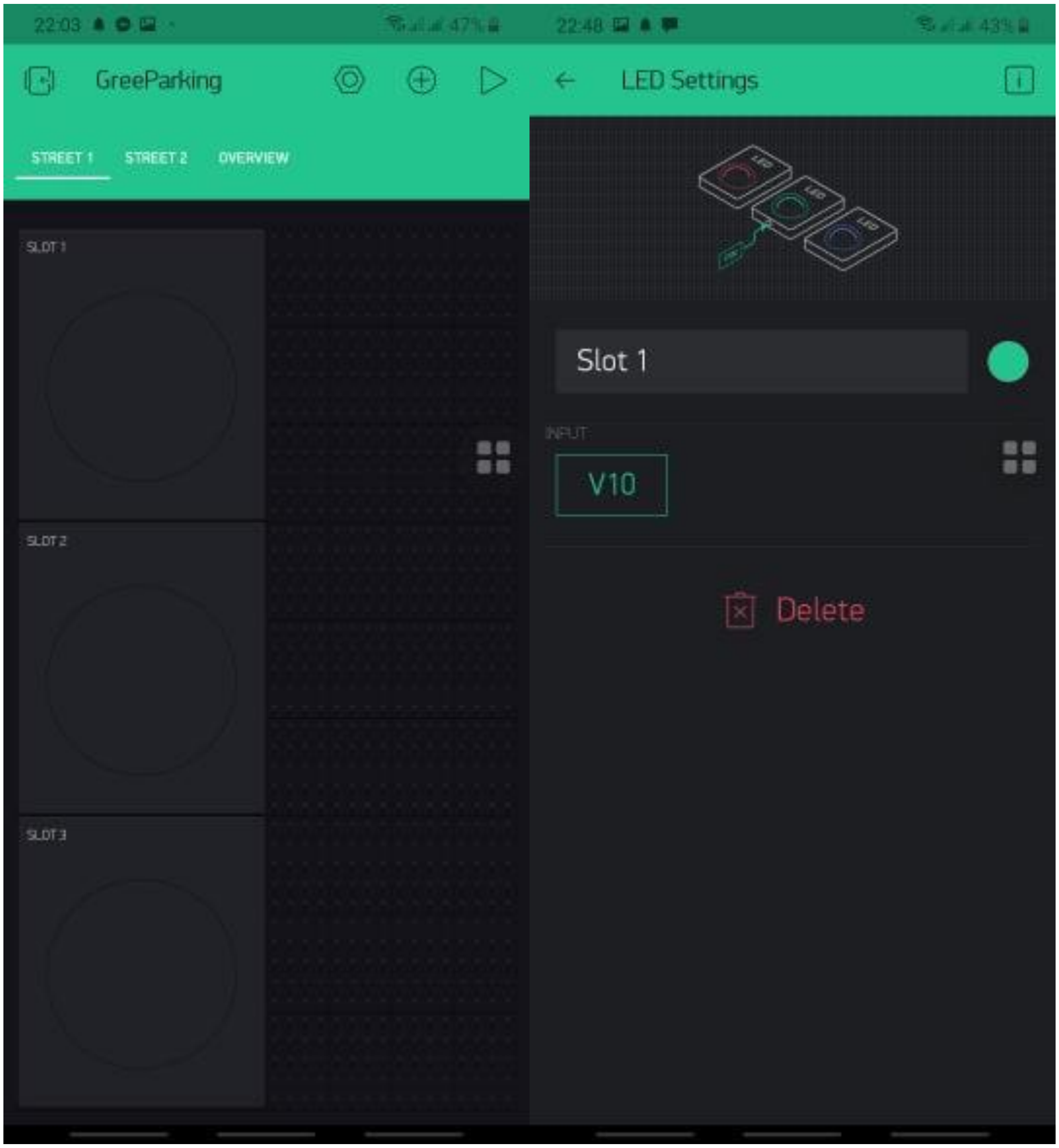
In the parking problem, the Blink Platform was used to solve this because:

- Provides similar API & user interface for all supported devices and hardware
- Connect to the server using:
    - WIFI
    - Bluetooth and BLE
    - Ethernet
    - USB (Serial)
    - GSM
- The interface utilities are easy to use by the provider
- Drag and drop interface directly without writing code
- Easily integrate and add new functions by using the virtual connection ports built into the blynk app
- Track data history
- Communication from device to device using Widgets
- Email, tweet, realtime notifications, etc.
- Blynk HTTP RESTful API: Blynk HTTP RESTful API allows us to easily read and write values to/from Pins in Blynk apps and Hardware (microcontrollers and microcomputers like Arduino, Raspberry Pi, ESP8266, Particle, etc.).Every **PUT** request will update Pin's state both in apps and on the hardware. Every **GET** request will return the current state/value on the given Pin. We also provide simplified API so you can do updates via **GET** requests.
- Supported Hardware: Blynk supports more than 400 boards already, including support for Arduino, Particle, ARM mbed, TI Energia, MicroPython, Node.js, OpenWRT and many Single Board Computers. You can add your own connection types easily.
- Libraries to connect any hardware:



Picture 14. Libraries Blynk to connect any hardware

In this project, the Blynk will show the result about slots parking – What slots is available



Picture 15. Using Blynk for Greeparking

1.5. Planning

1.5.1. Initiating phase

PROJECT CHARTER

1. General Project Information	
Project Name:	GreeParking
Executive Sponsors:	Tran Quang Huy & Head of R&D department
Department Sponsor:	R&D
Impact of the project:	Technology solutions to help parking car in the street

2. Project Team

	Name	Department	Telephone	E-mail
Project Manager:	Trần Quang Huy	R&D	0795541090	tranquanghuy@gmail.com
Team Members:	Huỳnh Thái Hiếu	R&D	0124579852	huynhthaihieu@gmail.com
	Nguyễn Hà Kiều My	R&D	0567489142	nguyenhakiemy@gmail.com
	Dương Minh Phúc	R&D	0564897456	minhphuc@gmail.com
	Lê Hạnh Dung	R&D	0697812313	hanhdung@gmail.com
	Nguyễn Quang Ngọc	Marketing	0798411320	quangngoc@gmail.com

3. Stakeholders
R&D department
HR department
Business department
University of Greenwich (DN Campus)

4. Project Scope Statement
Project Purpose / Business Justification
GreeParking project helps support people in parking cars on the street thereby reducing people’s time and traffic jams.
Objectives (in business terms)
- Reduce the “finding” time about 10-15 mins. - Bring new experiences to people. - Reduce the traffic jam

Deliverables
- Apply IoT into the system - Monitoring by LCD or application on mobile phone
Scope
- Applicable to the main street has a problem about parking. - Technologies: IoT Sensor, Cloud, Rasa core (NLP English, Vietnamese), Java, C language.
Major Known Risks (including significant Assumptions)

Risk	Risk Rating (Hi, Med, Lo)
Slow progress	High
Team member’s attitude & abilities	Medium
Out of budget	High
Technical problems	High

Constraints
- Limit budget for implement project. - Working between departments is difficult.
External Dependencies
There is an agreement between the stakeholder and project team
5. Communication Strategy
- Update progress to team leader every day. - Team leader report to the project manager every week. - Keep track of milestones.
6. Sign-off

	Name	Signature	Date (MM/DD/YYYY)
Executive Sponsor			
Department Sponsor			
Project Manager	Tran Quang Huy	*Huy*	

Table 8. Project Charter

1.5.2. Planning phase

1.5.2.1. Project work breakdown structure (WBS)

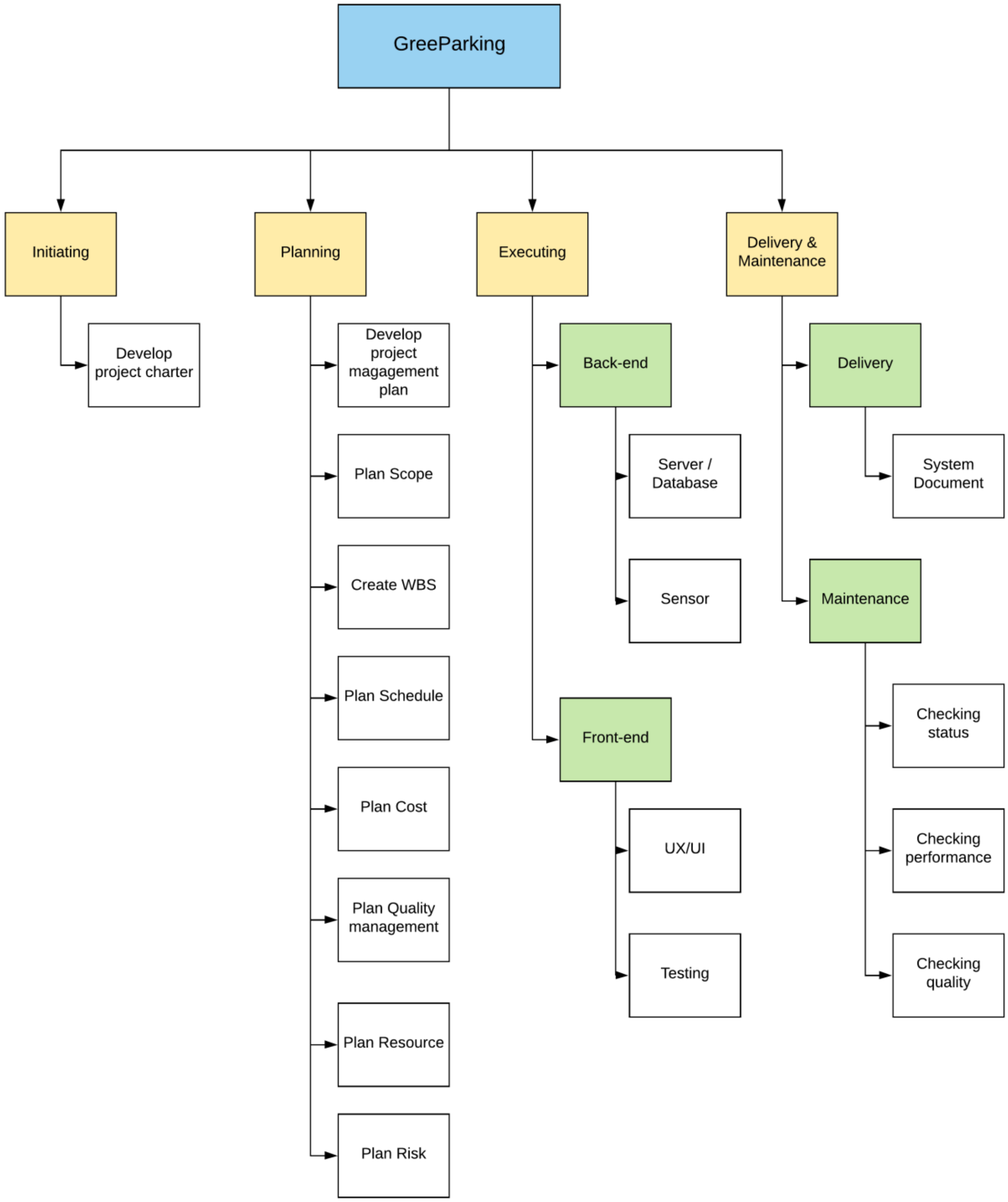


Figure 7. Work Breakdown Structure

1.5.2.2. Project cost

Phase	Process	Members	Work hours	Cost per hour	Total
Initiating					
	Develop project Charter	5 members	24 hours	$5	$600
Planning					
	Develop project management plan	4 members	24 hours	$5	$480
	Plan Scope	3 members	16 hours	$5	$240
	Create WBS	2 members	12 hours	$5	$70
	Plan Schedule	1 member	72 hours	$5	$720
	Plan Cost	2 members	20 hours	$5	$300
	Plan Quality management	3 members	18 hours	$5	$270
	Plan resource	3 members	10 hours	$5	$150
	Plan Risk	3 members	24 hours	$5	$360
Executing					
Back-end	Server	2 members	130 hours	$6	$1.560
	Database	3 members	160 hours	$6	$2.880
	Sensor	2 members	120 hours	$5	$1.200
Front-end	UX/UI	2 members	40 hours	$5	$400
	Testing	3 members	18 hours	$4	$216
Delivery & Maintenance					
Delivery	System Document	4 members	100 hours	$3	$1.200
Maintenance	Checking status	1 member	40 hours	$4	$120
	Checking performance	1 member	40 hours	$4	$120
	Checking quality	2 members	60 hours	$5	$600
Total					**$11.486**

Table 9. Project cost

Equipment for prototype:

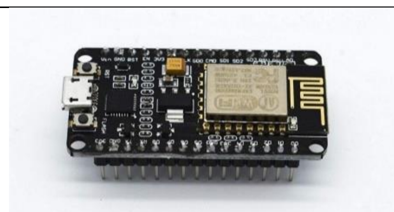
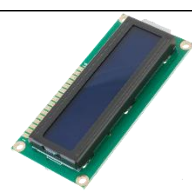
Item	Description	Quantity	Cost	Total
Arduino Uno R3		1	$3.6	$3.6
Infrared sense		7	$0.9	$6.3
Nodemcu esp8266 wifi module		1	$3.8	$3.8
Servo SG90		1	$1.6	$1.6
RFID RC522		1	$1.9	$1.9
LCD I2C 16x2		1	$1.8	$1.8
	Total			**$19**

Table 10. Equipment cost for prototype

1.5.2.3. Project risk

Risk categories	Risk	Responsible by	Risk rating
Technical	Requirements	Project team	Medium
	Technology	Technical team	High
	Interfaces	Developer team	Medium
	Performance	Project team	High
	Quality	Project team	High
External	Customer	Project manager	Medium
	Contract	Project manager	Medium
	Market	Marketing team	High
	Supplier	Project team	Low
Organizational	Project Dependencies	Project team	Medium
	Logistics	Project team	Medium
	Resources	Project manager	Medium
	Budget	Project manager	High
Project management	Planning	Project team	Medium
	Schedule	Project team	Medium
	Estimation	Project manager	High
	Controlling	Project manager	Medium
	Communication	Project team	Low

Table 11. Project risk

1.5.3. Executing phase

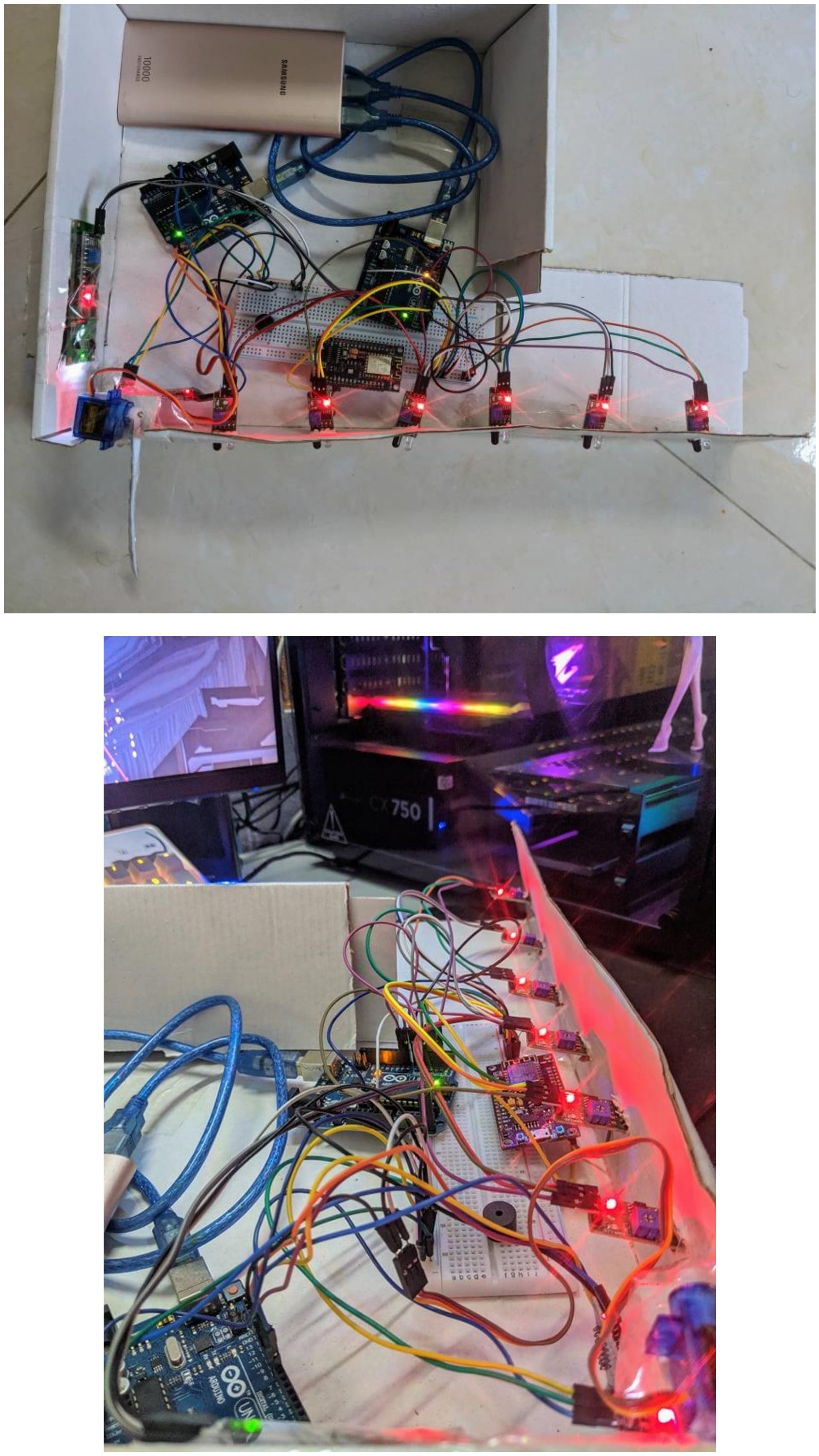
The implementation of the project consists of two main parts: back-end and front-end to create a complete product with the required features of the product.

1.5.4. Delivery and maintenance phase

The last part of the project is to maintain and develop products that operate in the best way to serve customers in the best way.

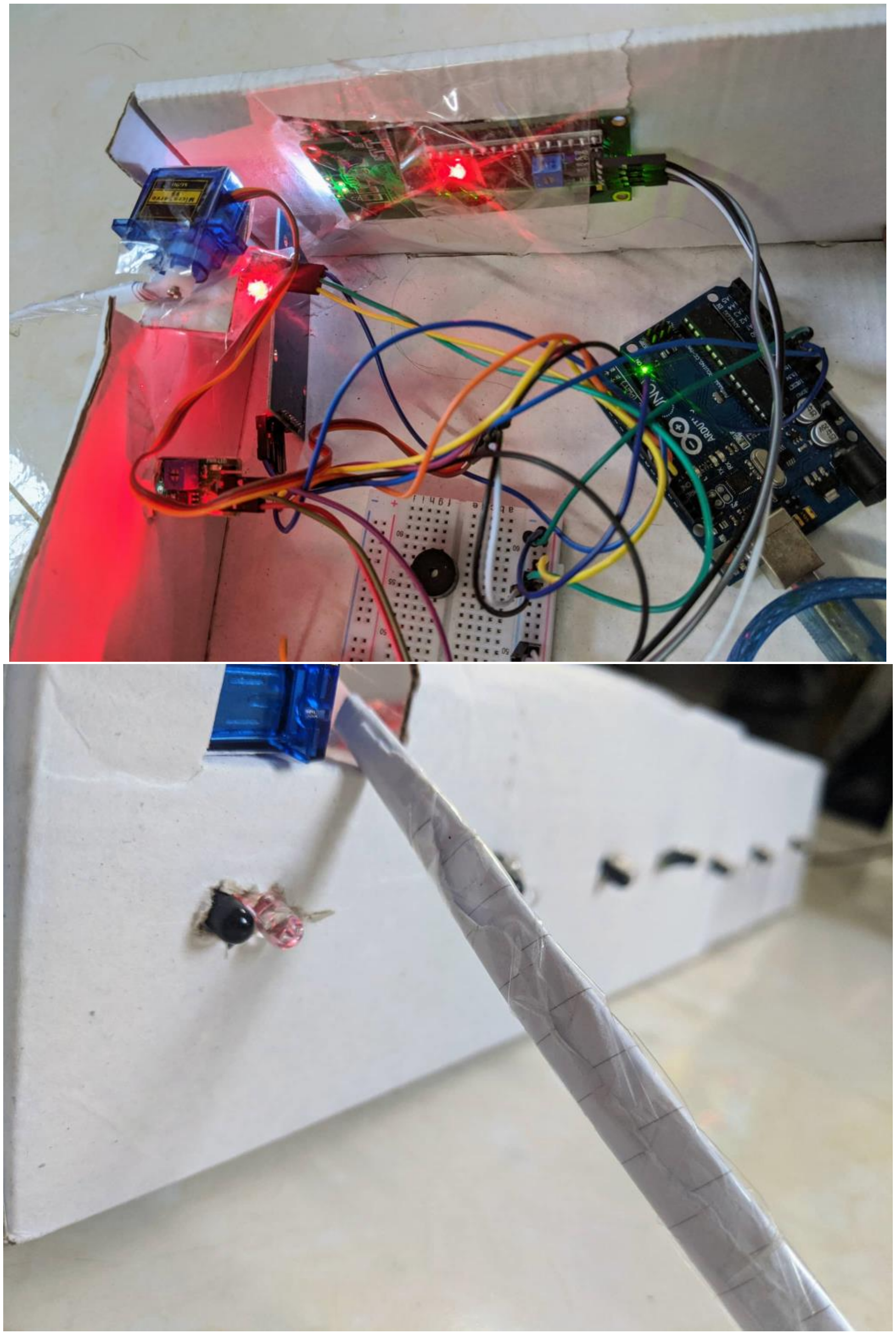
1.6. The actual connection diagrams

- Overview GreeParking (Prototype):



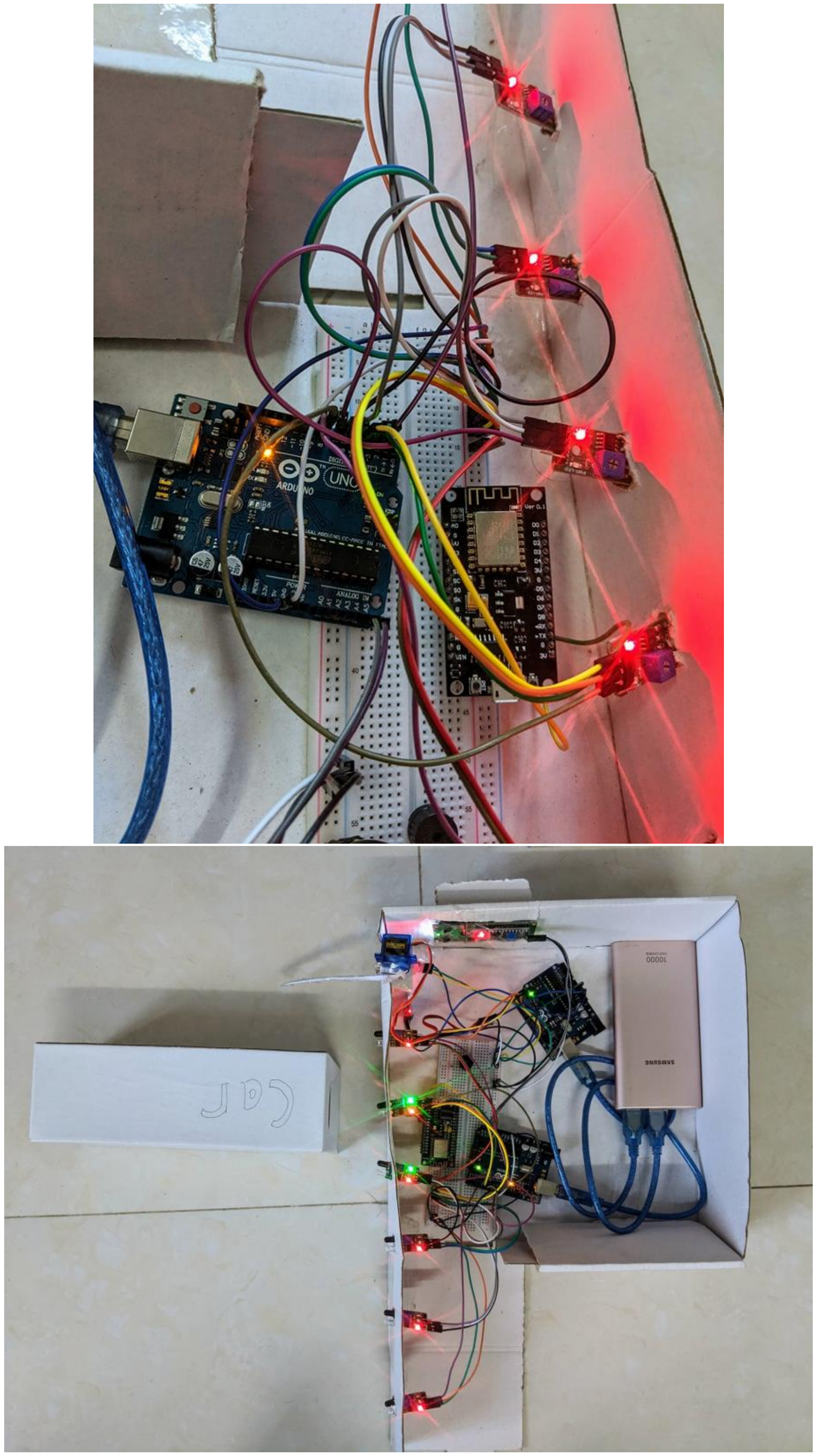
Picture 16. Overview GreeParking (Prototype)

- Gate (Extension):



Picture 17. Gate (Extension)

- Sensor:



Picture 18. Sensor

1.7. Code (Sketch), upload Sketch, and evidence

Code for Arduino UNO R3 (Gate):

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>

#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

Servo myservo;

int parkingIN = 7;
int parkingOUT = 6;

boolean LEDStatus=false;
```

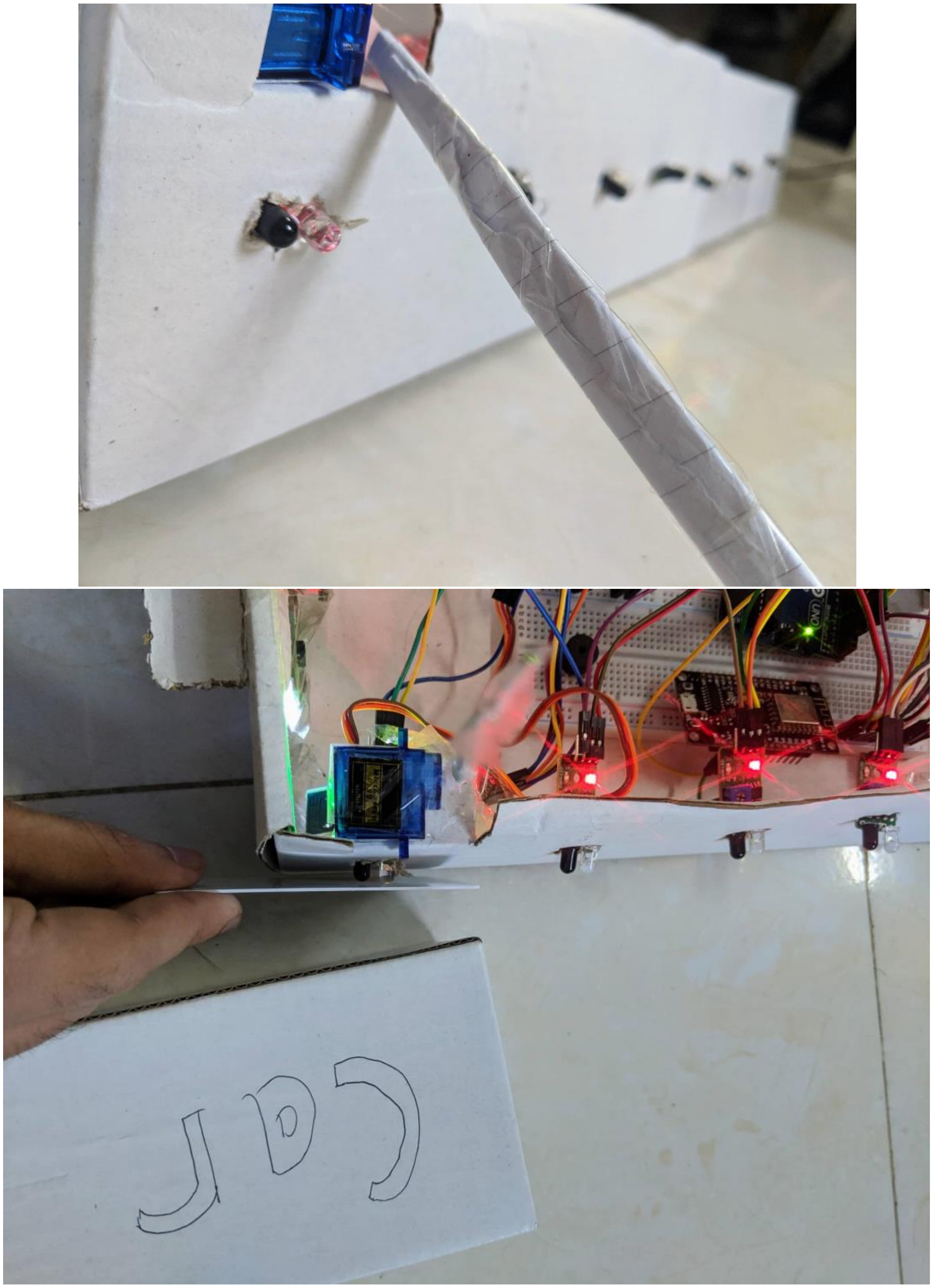
```
void setup()
{
  //Conect servo with pin 8
  myservo.attach(8);

  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522


  Serial.println("Approximate your card to the reader...");
  Serial.println();

  pinMode(parkingIN, INPUT);

  myservo.write(180);
}
```

```
void loop()
{
  // Look for new cards
  if ( !mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( !mfrc522.PICC_ReadCardSerial())
  {
    return;
  }


  //Show UID on serial monitor
  Serial.print("UID tag :");
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }
  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();
  if (digitalRead(parkingIN) == LOW)
  {
    //The UID of the card/cards access
    if (content.substring(1) == "AD D5 07 85")
    {
      Serial.println("Authorized access");
      Serial.println();
      myservo.write(90);
      delay(3000);

      //Sensor IR
      if (digitalRead(parkingIN) == HIGH)
      {
        //Open the gate
        myservo.write(180);
      }
    }

   else
   {
      Serial.println(" Access denied");
      delay(1000);
      myservo.write(180);
    }
  }
```

- **Test Case:**

No.	Action	Inputs	Expected Output	Actual Output	Test Result
AG01	Using correct cards	Using correct card with UID = “AD D5 07 85”	The Serial print = “Authorized access” and the Servo “open”	The Serial print = “Authorized access” and the Servo “open”	Pass
AG02	Using incorrect cards	Using incorrect card with UID = “40 62 5E 7A”	The Serial print = “Access denied” and the Servo “close”	The Serial print = “Access denied” and the Servo “close”	Pass
AG03	Using correct cards after using incorrect cards	Using correct card with UID = “AD D5 07 85” and Using incorrect card with UID = “40 62 5E 7A”	The Serial print = “Authorized access” and the Servo “open”	The Serial print = “Authorized access” and the Servo “open”	Pass
AG04	Using incorrect cards after using correct cards	Using incorrect card with UID = “40 62 5E 7A” and Using correct card with UID = “AD D5 07 85”	The Serial print = “Access denied” and the Servo “close”	The Serial print = “Access denied” and the Servo “close”	Pass
AG05	Check IR sensor off when working	Turn off IR sensor and using the correct card with UID = “AD D5 07 85”	The Serial print = “The system has error” and the Servo “close”	The Serial print = “Access denied” and the Servo “close”	Fail
AG06	Check when the user moves out of range of IR sensor	Active IR sensor and using correct card with UID = “AD D5 07 85” then inactive IR sensor	The Serial print = “Authorized access” and the Servo “open” After that Servo will “close”	The Serial print = “Authorized access” and the Servo “open” After that Servo will “close”	Pass

Table 12. Test Case for Arduino uno R3 (Gate)



Picture 19. How the gate in GreeParking work

Code for Arduino uno R3 (Sensor IR):

```
#include <SoftwareSerial.h>
#include <Wire.h>;
#include <LiquidCrystal_I2C.h>;
LiquidCrystal_I2C lcd(0x3F,16,2);

// Connect nodeMCU with Arduino R3
SoftwareSerial nodemcu(2,3);

// parking slot1 infrared sensor connected with pin number of arduino
int parking1_slot1_ir_s = 4;
int parking1_slot2_ir_s = 5;
int parking1_slot3_ir_s = 6;
int parking2_slot1_ir_s = 7;
int parking2_slot2_ir_s = 8;
int parking2_slot3_ir_s = 9;

//Data for each IR sensor
String sensor1;
String sensor2;
String sensor3;
String sensor4;
String sensor5;
String sensor6;

//Total Parking Car Slot
int countSlot = 6;

// complete data, consisting of sensors values
String cdata ="";
```

```
void setup()
{
  Serial.begin(9600);
  nodemcu.begin(9600);

  pinMode(parking1_slot1_ir_s, INPUT);
  pinMode(parking1_slot2_ir_s, INPUT);
  pinMode(parking1_slot3_ir_s, INPUT);

  pinMode(parking2_slot1_ir_s, INPUT);
  pinMode(parking2_slot2_ir_s, INPUT);
  pinMode(parking2_slot3_ir_s, INPUT);

  lcd.begin();
  lcd.backlight();
  lcd.setCursor(2,0);
  lcd.print("GreeParking");
  lcd.setCursor(3,1);
  lcd.print("System");

}
```

```
void loop()
{
  //Call the function below for each parking slot
  p1slot1();
  p1slot2();
  p1slot3();

  p2slot1();
  p2slot2();
  p2slot3();

  //Show data on LCD
  ShowData();
  countSlot =6;

  // comma will be used a delimeter
  cdata = cdata + sensor1 +"," + sensor2 + ","+ sensor3 +","+ sensor4 + "," +
sensor5 + "," + sensor6 +",";
  Serial.println(cdata);

  //Send data to nodeMCU
  nodemcu.println(cdata);
  delay(6000);

  //resert data in loop()
  cdata = "";

  digitalWrite(parking1_slot1_ir_s, HIGH);
  digitalWrite(parking1_slot2_ir_s, HIGH);
  digitalWrite(parking1_slot3_ir_s, HIGH);

  digitalWrite(parking2_slot1_ir_s, HIGH);
  digitalWrite(parking2_slot2_ir_s, HIGH);
  digitalWrite(parking2_slot3_ir_s, HIGH);
}
```

```
// When NodeMCU send data to Blynk, "255" will turn led on and "0" will turn off

void p1slot1() // parking 1 slot1
{
  if( digitalRead(parking1_slot1_ir_s) == LOW)
  {
    sensor1 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking1_slot1_ir_s) == HIGH)
  {
    sensor1 = "0";
    delay(200);
  }

}

void p1slot2() // parking 1 slot2
{
  if( digitalRead(parking1_slot2_ir_s) == LOW)
  {
    sensor2 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking1_slot2_ir_s) == HIGH)
  {
    sensor2 = "0";
    delay(200);
  }
}


void p1slot3() // parking 1 slot3
{
  if( digitalRead(parking1_slot3_ir_s) == LOW)
  {
    sensor3 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking1_slot3_ir_s) == HIGH)
  {
    sensor3 = "0";
    delay(200);
  }
}
```

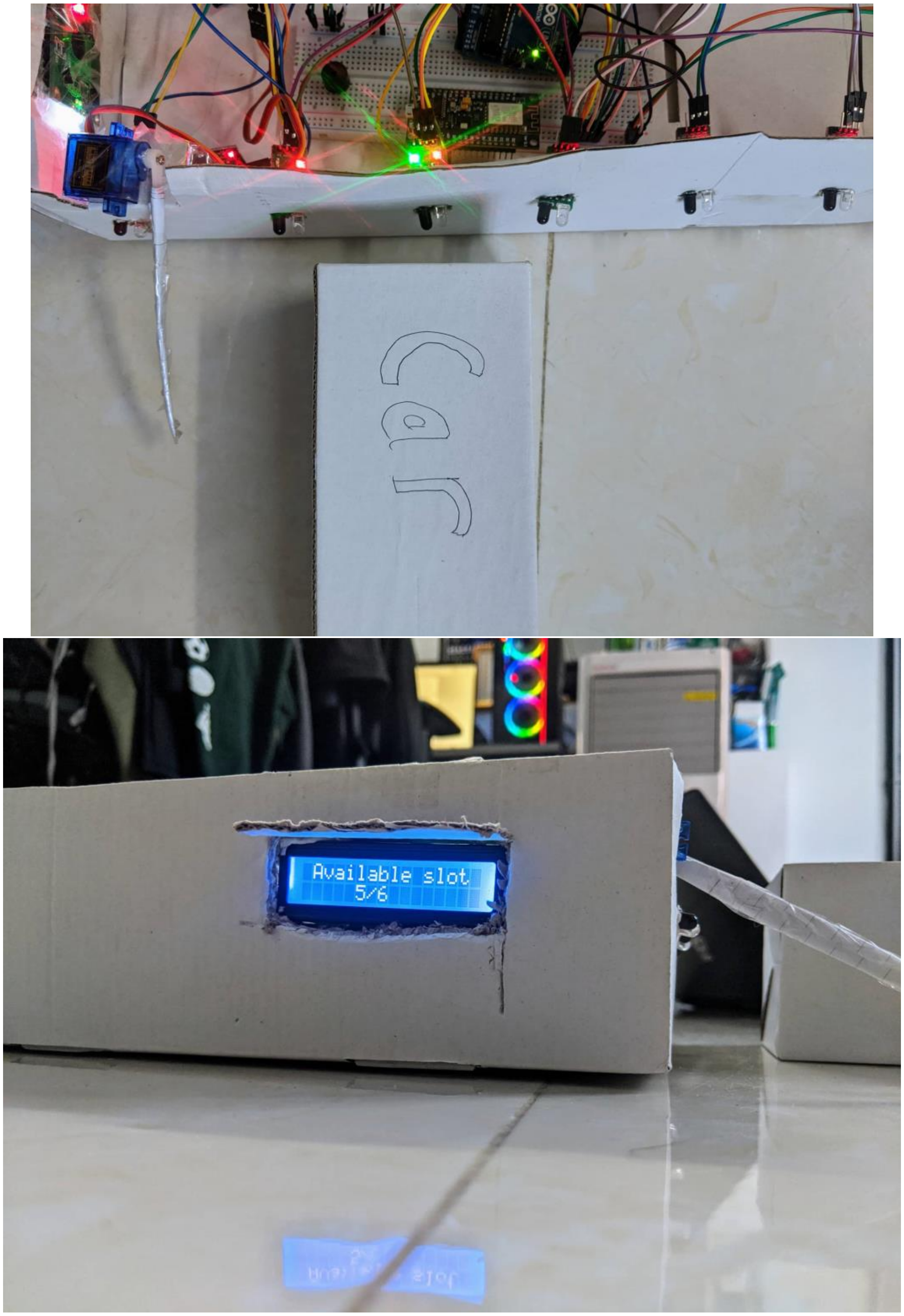
```
// for parking 2

void p2slot1() // parking 1 slot3
{
  if( digitalRead(parking2_slot1_ir_s) == LOW)
  {
    sensor4 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking2_slot1_ir_s) == HIGH)
  {
    sensor4 = "0";
    delay(200);
  }
}


void p2slot2() // parking 1 slot3
{
  if( digitalRead(parking2_slot2_ir_s) == LOW)
  {
    sensor5 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking2_slot2_ir_s) == HIGH)
  {
    sensor5 = "0";
    delay(200);
  }
}


void p2slot3() // parking 1 slot3
{
  if( digitalRead(parking2_slot3_ir_s) == LOW)
  {
    sensor6 = "255";
    countSlot--;
    delay(200);
  }
  if( digitalRead(parking2_slot3_ir_s) == HIGH)
  {
    sensor6 = "0";
    delay(200);
  }
}
```

```
//Show data on LCD
void ShowData()
{
  delay(2000);
  lcd.clear();
  lcd.setCursor(1,0);
  lcd.print("Available slot");

  lcd.setCursor(5,1);
  lcd.print(countSlot);
  lcd.print("/6");

  delay(300);
}
```

- **Test Case:**

No.	Action	Inputs	Expected Output	Actual Output	Test Result
IR01	Show data on LCD with 1/6 car	Active IR sensor slot 2	The LCD shows “Available slots: 5/6”	The LCD shows “Available slots: 5/6”	Pass
IR02	Show data on LCD with 6/6 cars	Active IR sensor slot 1 to slot 6	The LCD shows “Available slots: 0/6”	The LCD shows “Available slots: 5/6”	Pass
IR03	Send data to NodeMCU with 1 car	Active IR sensor slot 2	Cdata = “0,255,0,0,0,0”	Cdata = “0,255,0,0,0,0”	Pass
IR04	Send data to NodeMCU with 6 cars	Active IR sensor slot 1 to slot 6	Cdata = “255,255,255,255,255,255”	Cdata = “255,255,255,255,255,255”	Pass
IR05	Check the LCD working if 1 IR sensor has a problem	Disconnect IR sensor slot 6 and parking car into slot 1	The LCD shows “Available slots: 4/5”	The LCD shows “Available slots: 5/6”	Fail
IR06	Check the system working if 1 IR sensor have problem	Disconnect IR sensor slot 6 and parking car in to slot 1	Cdata = ““255,0,0,0,0,0”	And Cdata = “255,0,0,0,0,0”	Pass

Table 13. Test case for Arduino uno R3 (Sensor IR)



Picture 20. Result of Parking

Code for NodeMCU:

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SoftwareSerial.h>
#include <SimpleTimer.h>

char auth[] = "NCjLxujCirHqooYFtCemy_c8eAyEoTyP";

//WiFi credentials.
char ssid[] = "Greenwich-Student";
char pass[] = "12345678";

SimpleTimer timer;

// complete message from arduino, which consistors of snesors data
String myString;

// received charactors
char rdata;

int led1,led2,led3,led4,led5,led6;
```

```
void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);

  timer.setInterval(1000L,sensorvalue1);
  timer.setInterval(1000L,sensorvalue2);
  timer.setInterval(1000L,sensorvalue3);
  timer.setInterval(1000L,sensorvalue4);
  timer.setInterval(1000L,sensorvalue5);
  timer.setInterval(1000L,sensorvalue6);
}
```

```
void loop()
{
  if (Serial.available() == 0 )
  {
    Blynk.run();
    // Initiates BlynkTimer
    timer.run();
  }

  if (Serial.available() > 0 )
  {
    rdata = Serial.read();
    myString = myString+ rdata;
    // Serial.print(rdata);
    if( rdata == '\n')
    {
      Serial.println(myString);

      String l = getValue(myString, ',', 0);
      String m = getValue(myString, ',', 1);
      String n = getValue(myString, ',', 2);
      String o = getValue(myString, ',', 3);
      String p = getValue(myString, ',', 4);
      String q = getValue(myString, ',', 5);

      // these leds represents the leds used in Blynk application
      led1 = l.toInt();
      led2 = m.toInt();
      led3 = n.toInt();
      led4 = o.toInt();
      led5 = p.toInt();
      led6 = q.toInt();

      myString = "";
    }
  }
}
```

```
// This function sends Arduino's up time every second to Virtual Pin (10).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that define how often to send data to Blynk App.
void sensorvalue1()
{
  int sdata = led1;
  //Send any value at any time.
  Blynk.virtualWrite(V10, sdata);

}
void sensorvalue2()
{
  int sdata = led2;
  //Send any value at any time.
  Blynk.virtualWrite(V11, sdata);

}

void sensorvalue3()
{
  int sdata = led3;
  //Send any value at any time.
  Blynk.virtualWrite(V12, sdata);

}

void sensorvalue4()
{
  int sdata = led4;
  //Send any value at any time.
  Blynk.virtualWrite(V13, sdata);

}

void sensorvalue5()
{
  int sdata = led5;
  //Send any value at any time.
  Blynk.virtualWrite(V14, sdata);
}

void sensorvalue6()
{
  int sdata = led6;
  //Send any value at any time.
  Blynk.virtualWrite(V15, sdata);
}
```

```
//Get data from arduino
String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  int maxIndex = data.length() - 1;

  for (int i = 0; i <= maxIndex && found <= index; i++)
  {
    if (data.charAt(i) == separator || i == maxIndex)
    {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```

Test Case:

No.	Action	Inputs	Expected Output	Actual Output	Test Result
NM01	Check Wi-Fi connection	Turn off Wi-Fi router	The system still working but on an application doesn’t show anything	The system still working but on an application doesn’t show anything	Pass
NM02	Check Wi-Fi connection	Change SSID of Wi-Fi	The system working well	The system doesn’t work	Fail
NM03	Add more sensor	Implement more sensor for project	The system recognizing new sensor and add to the system	The system just keep old sensors	Fail

Table 14. Test case for NodeMCU

2. End-user experiments and examines feedback

2.1. Practical application

Parking problems are not uncommon, especially for big cities. By 2023, market spending for smart parking products and services is expected to grow at a CAGR of 14% and surpass $3.8B according to an IoT Analytics report. The growth of market spending is good news because it will force people to try to find a solution to these traffic problems instead of taking no action.

How smart parking system works:

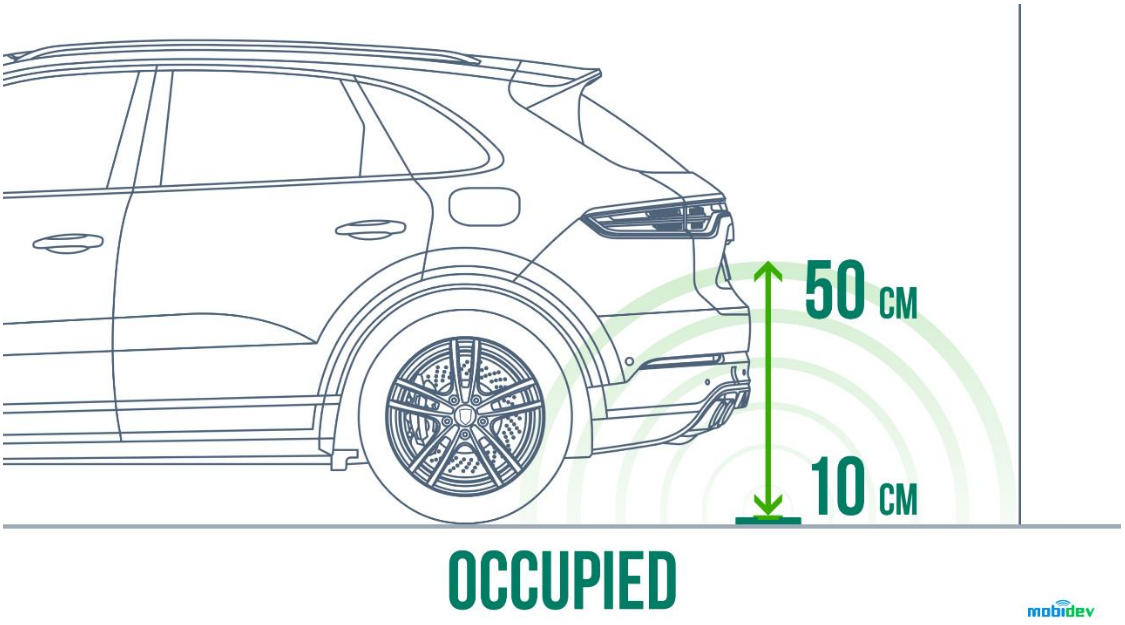
Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place. The user receives a live update about the availability of all parking places and chooses the best one.

In order to investigate technologies behind the smart parking solution, we implemented an internal research project. The main idea was the creation of smart parking using the Internet of Things and ultrasonic sensors, where available parking places could be displayed in an application.

In this case, if the sensor detects nothing up to 50 cm, the status is set to “free” and is shown to the user.

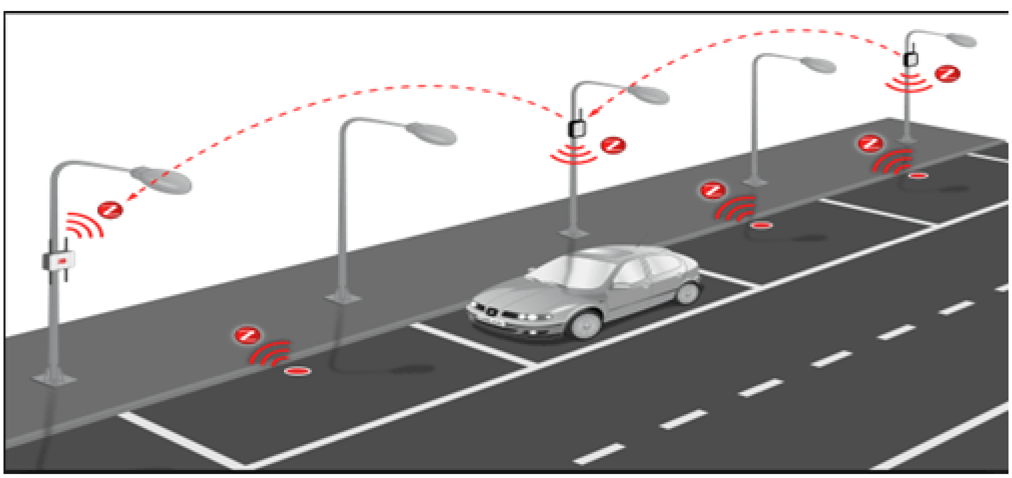
Picture 21. Visualize how GreeParking IR Sensor works with no parking car

The "occupied" range is between 10 and 50 cm.



Picture 22. Visualize how GreeParking IR Sensor works with parking car

When applying this product in practice, The device that has built as part of this project is an application of the Internet of Things (IoT). The electronic device that you build can assist the drivers in selecting a suitable parking spot for them. The data that the system collects will be sent to the cloud and anyone across the globe can access the data to do some analysis.



Picture 23. Demo setup IoT Devices in street. (skyfilabs, n.d.)

2.2. Evidence for apply GreeParking in practice

How GreeParking to work well? it is necessary to have detailed assessments from many stakeholders and many different feedbacks. From those results, can give assessments, improve and overcome the bad points of the product.

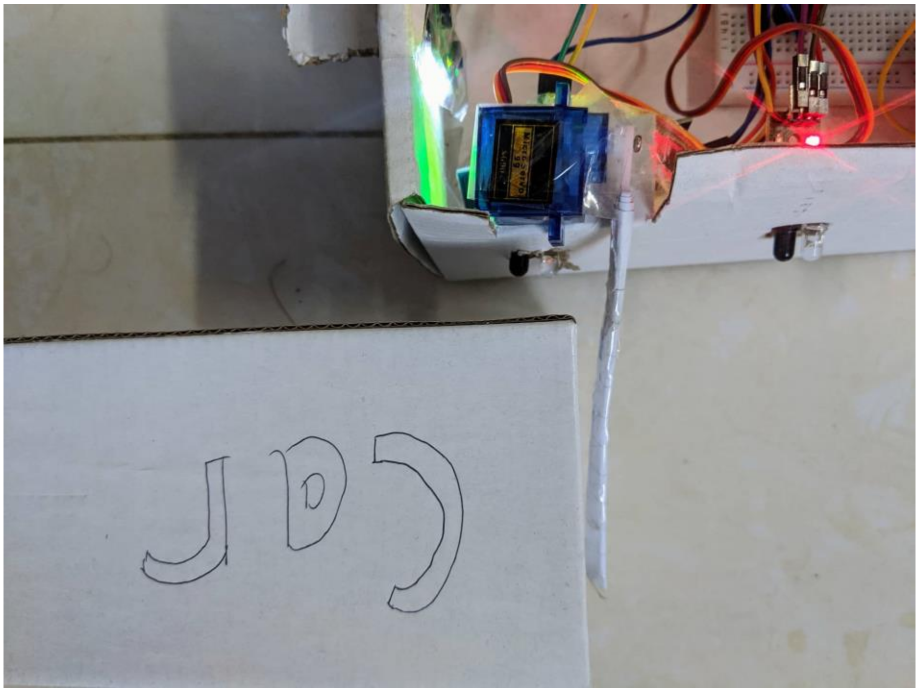
There is a test case for apply GreeParking in practice

No.	Action	Inputs	Expected Output	Actual Output	Test Result
WE1	Check the system works under the rain	Create artificial rain	The sensor still working	The sensor has a problem to recognize car parking	Fail
WE2	Check the system works under the rain with car parking	Create artificial rain and car park in sensor 1	The LCD shows "Available slots: 5/6" And the result on application turn on led slot 1.	The LCD shows "Available slots: 4/6" And the result on application turn on led slot 1 and slot 4	Fail
WE3	Check the Waterproof of system	Put the system into the water	The sensor still working	The sensor is broken	Fail
WE4	Check the system work under Smog	Create artificial smog	The sensor still working	The sensor still working	Pass

Table 15. Test case for using Greeparking in different weather

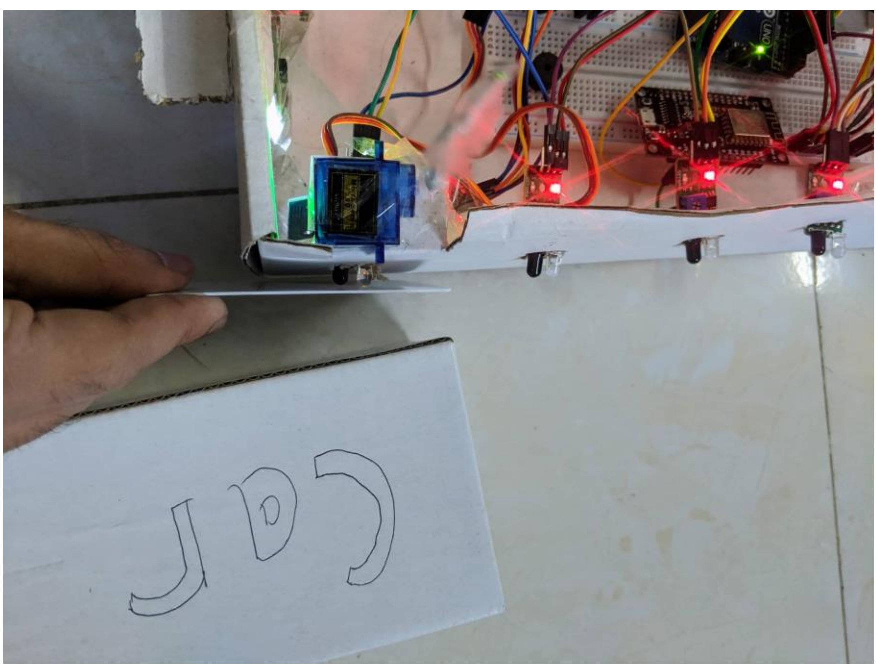
Example GreeParking in practice:

The car wants to park:



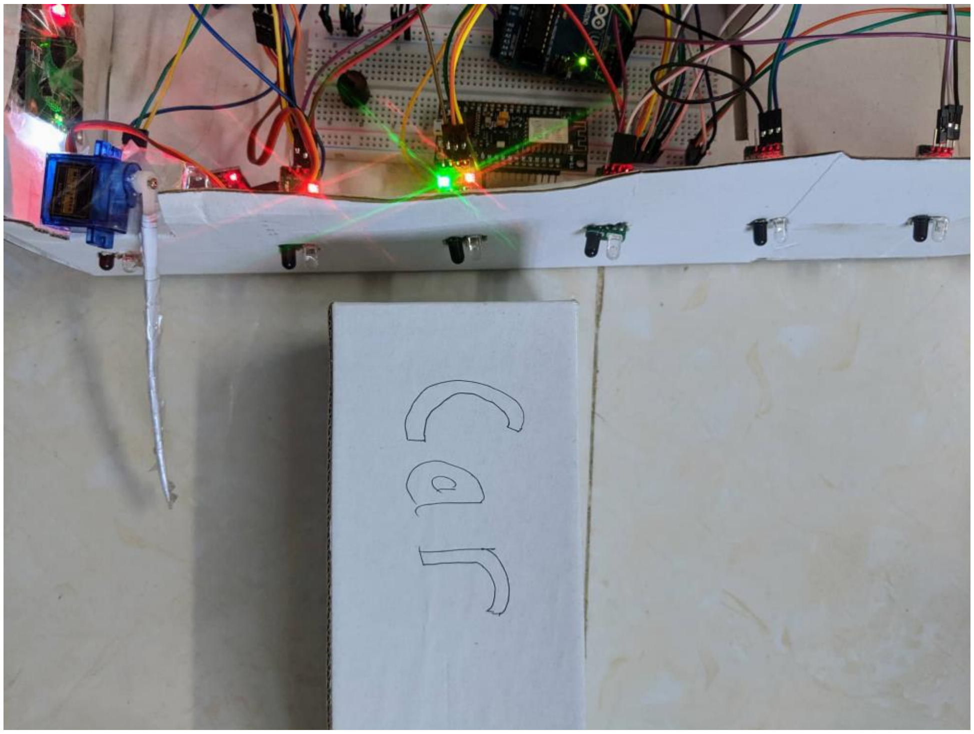
Picture 24. The car wants to park

Using RFID to unlock the security gate:



Picture 25. Using RFID to unlock the security gate

Car parking at slot parking 2:

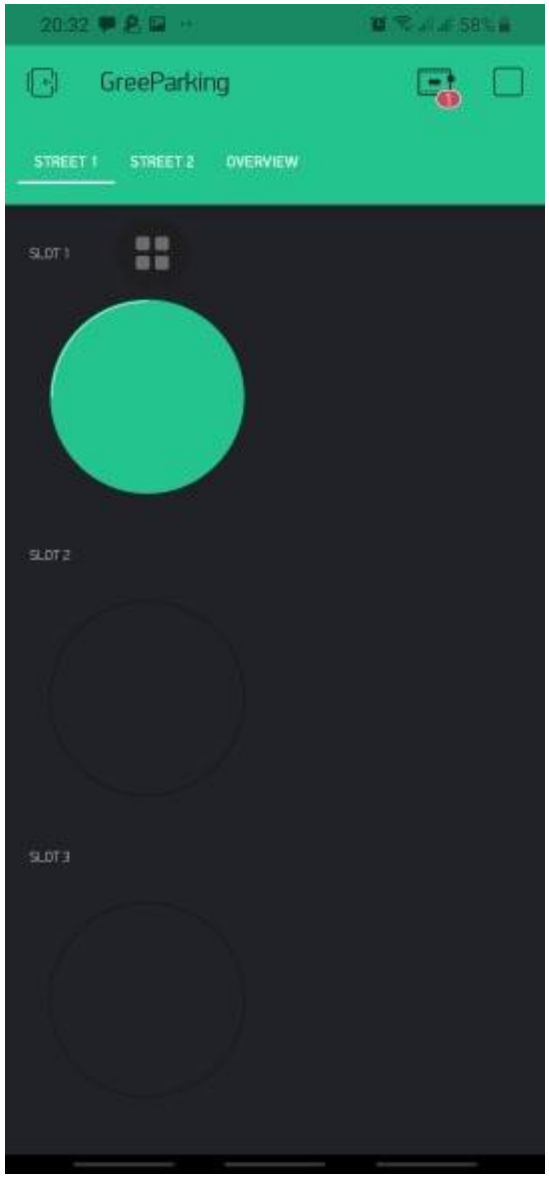


Picture 26. Car parking at slot parking 2

The LCD show the available slot:

Picture 27. The LCD show the available slot

The result when using the application on mobile phone:



Picture 28. The result when using the application on mobile phone

2.3. Point achieved and not achieved of GreeParking

Point achieved:

- The sensors work well in many different weather conditions.
- Solve the problem of parking on some important roads.
- GreeParking system is easy to use.
- Information transmitted from the sensor accurately.
- Easy installation and construction of GreeParking's equipment.

Point not achieved:

- With the harsh weather conditions such as storms, rain, the device has some problems with the sensors.
- Depending on the Wi-Fi network, if there is no Wi-Fi network, the device cannot upload data to the application software.
- The device cannot operate without electricity.

3. Reconcile and evaluate end-user feedback and determine the advantages and disadvantages of chosen IoT techniques.

3.1. The modification

Version	Problem	Solution
Greeparking 1.0	Sensor	Change from Infrared (IR) Sensors to Ultrasonic.
	Application	Change application from Blynk to Greeparking application on Android.
	Networking	Change module Nodemcu esp8266 wifi module to Board SIM800C intoRobot Fox.
Greeparking 2.0	Sensor	Upgrade Waterproof sensor.
	Application	Create application for IOS phone.
	Database	Upload data on Greeparking's cloud.
	Networking	Testing using LoRa to transfer data.
Greeparking 3.0	Energy	Use solar energy

Table 16. Modification of GreeParking through each version

3.2. Evaluate the entire solution, advantages, and disadvantages

The innovative solutions for GreeParking products are based on feedback from users, among them, there is good and bad feedback.

An increasing number of vehicles across the globe are instrumental in intensifying the parking woes. The alarming problem of traffic congestion is rising exponentially on a global scale. The increasing need to counter and address parking-related issues is significant in creating a high demand for efficient parking systems. Smart parking systems provide flexibility and scalability to effectively address the parking issues.

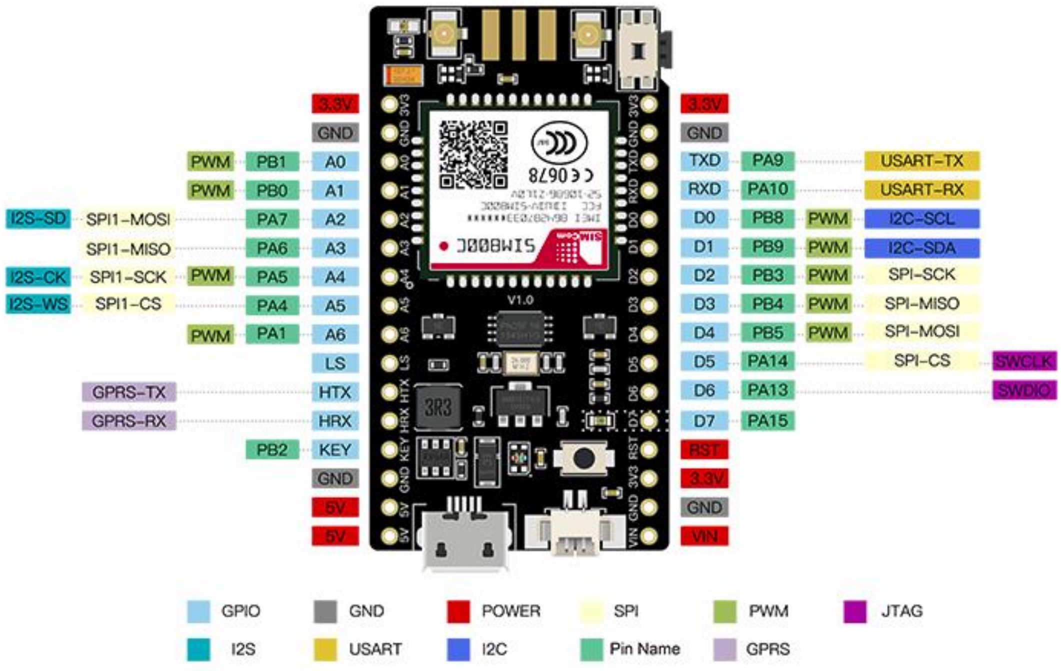
The adoption of smart parking systems involves the use of low-cost sensors and mobile phone-enabled automated payment systems. Such sophisticated features enable users in determining the available parking spaces and reserving the slot in advance reducing the time required to avail parking. The parking system helps in minimizing car emissions by eliminating the need for people to unnecessarily circle the city blocks in search of parking spaces.

Since the system helps in minimizing traffic congestion and pollution levels, governments across the globe are undertaking various initiatives for the large-scale adoption of such parking systems. Favorable administrative schemes are presumed to remarkably contribute to industry expansion in the coming years. The large scale system component is possible only through significant funding initiatives undertaken by the national governments. Nevertheless, the increasing adoption of these systems at airports, corporate campuses, and similar avenues is predicted to fuel the industry prospects.

Plans have been put in place to improve the product, which has overcome most of the issues such as weather, data transfer, energy usage.

3.3. Solutions to improve the solution or to replace in the future

Board: Instead of using NodeMCU and Arduino devices depending on Wi-Fi network conditions, the system can use boards with 3G, 4G, LTE connectivity.



Picture 29. Board SIM800C IntoRobot-Fox. (iotmaker, n.d.)

Sensor: Use IOE-SR05 distance measurement sensor to replace IR sensor to enhance display results

Picture 30. IOE-SR05. (nshopvn, n.d.)

Application: The system is using Blynk to communicate data to users, which does not guarantee the security of information. To improve future products, GreeParking will design a separate application, server, database to meet the above needs.

The system also worked on the concept of UX (user experience) to make a user's stay with the application to be as easy as it could get with full functionalities included.

When designing the applications, the system worked around the following seven concepts:

- Light comes from the sky
- Black and White first
- Doubling the whitespace
- Overlaying text on images
- Making text pop and un-pop
- Simple and uniform font choice
- Stealing like an artist

Waterproof: Solutions to prevent the device from working under rain conditions have not been developed. Therefore, GreeParking will have a waterproof case.

Part II. Evaluate IoT application and detail the problem IoT application solves, the potential impact on people, business, society, and the end-user and the problems it might encounter when integrating into the wider IoT ecosystem

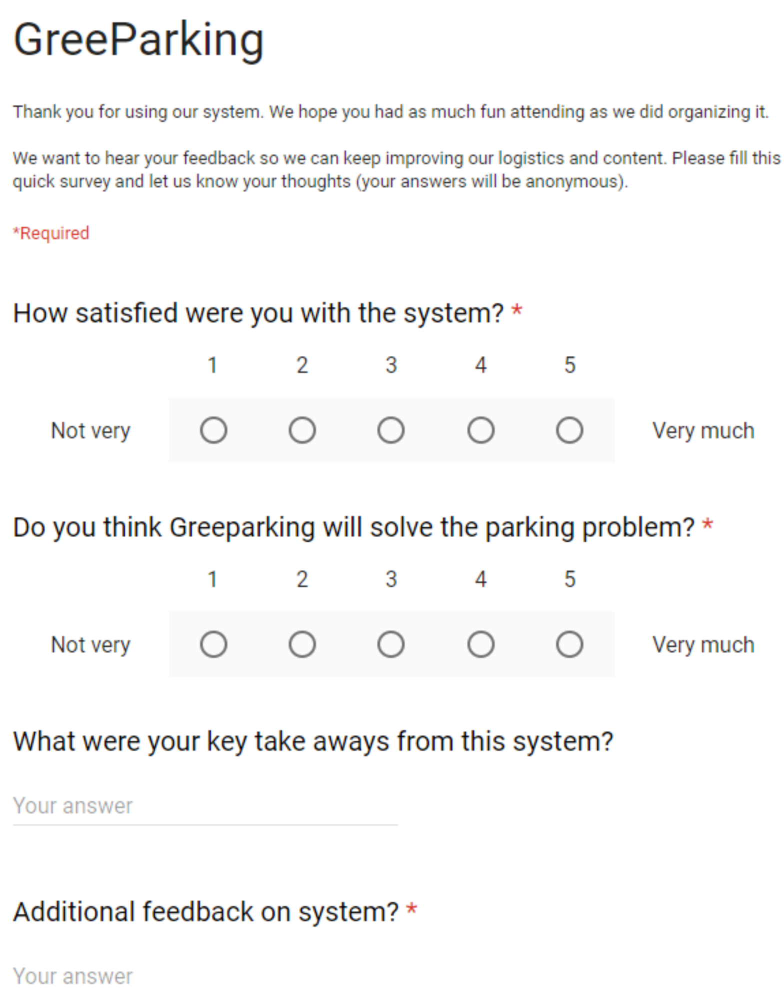
1. Evaluating end-user feedback from your IoT application

1.1. Feedback from users

After applying GreeParking system, the customer experience has responded to many good and bad points of the product.

Experienced participants include car owners, motorcyclists, shop owners along the way, etc.

In order to meet the user's requirement of Greeparking, the survey tools were applied, through interviewing and monitoring customer behavior, the questionnaire was used as the best option to meet the business requirements. Below is a sample survey:



GreeParking

Thank you for using our system. We hope you had as much fun attending as we did organizing it.

We want to hear your feedback so we can keep improving our logistics and content. Please fill this quick survey and let us know your thoughts (your answers will be anonymous).

*Required

How satisfied were you with the system? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

Do you think Greeparking will solve the parking problem? *

	1	2	3	4	5	
Not very	○	○	○	○	○	Very much

What were your key take aways from this system?

Your answer

Additional feedback on system? *

Your answer

Which part did you find most relevant? *

	Not relevant	Relevant	Very relevant	Did not attend
Sensor	○	○	○	○
Server	○	○	○	○
Database	○	○	○	○
Application	○	○	○	○

Any overall feedback for the system?

Your answer

SUBMIT

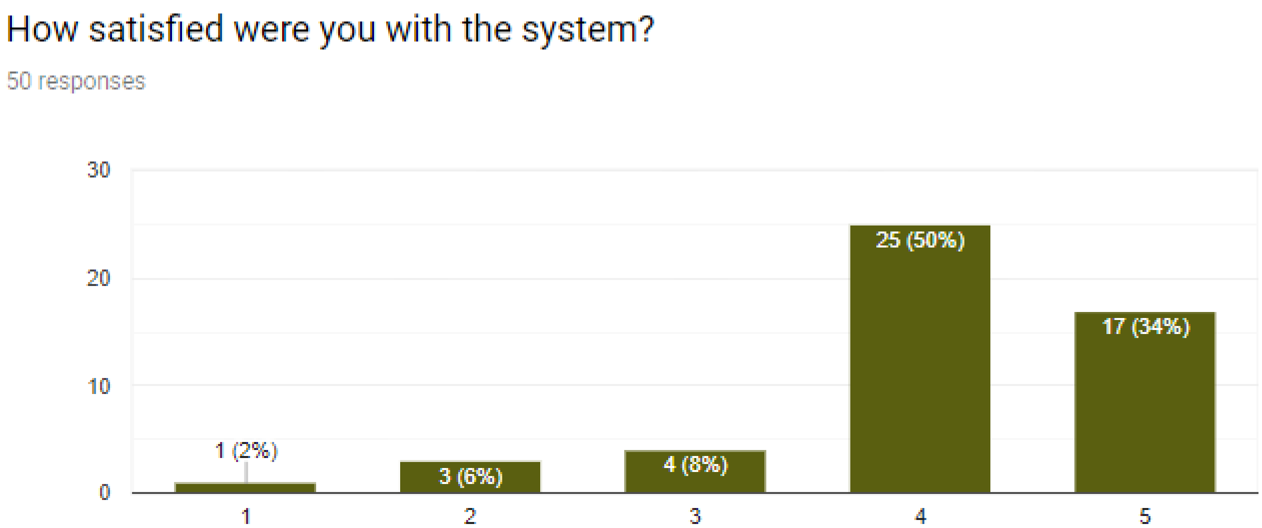
Never submit passwords through Google Forms.

Through the questionnaire sent to 55 users who experienced the GreeParking system, there were 52 users who answered the questionnaire and had 50 questionnaires valid. General statistics on the purpose achieved after using the question system are as follows:

This IoT is a comprehensive system that includes hardware and software. It is important to identify key stakeholders:

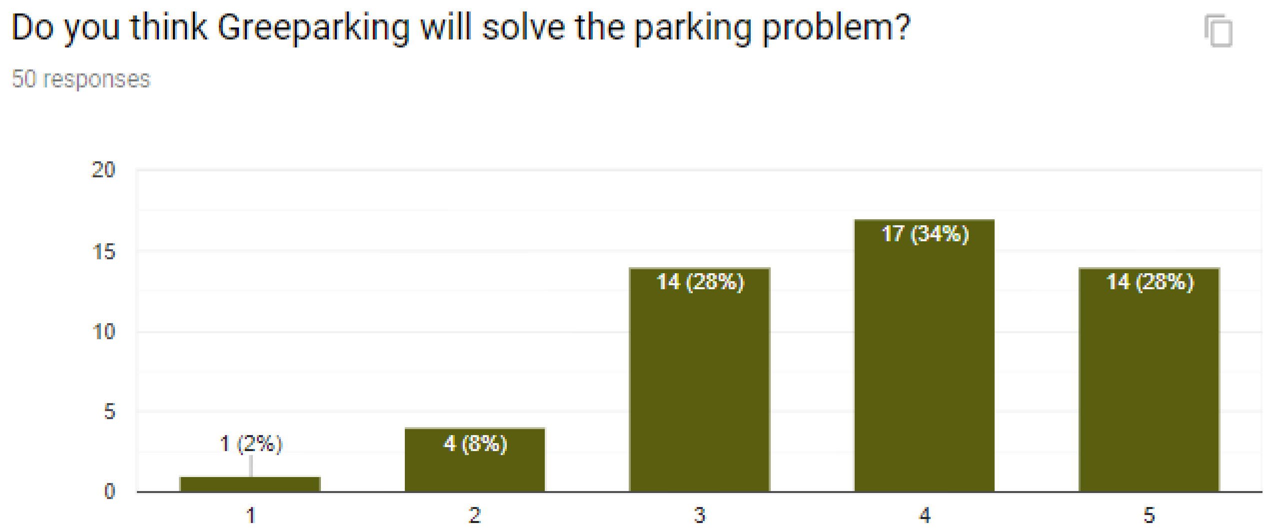
- Customer: Does using the application meet the needs of customers? Is the user happy with the product or has related issues when using?
- Developer: Has there been any improvement in the project implementation, has the hardware and software functions been stabilized? Need to improve or change any hardware or software?
- Related people: Providers of software and hardware services

The result of the survey:



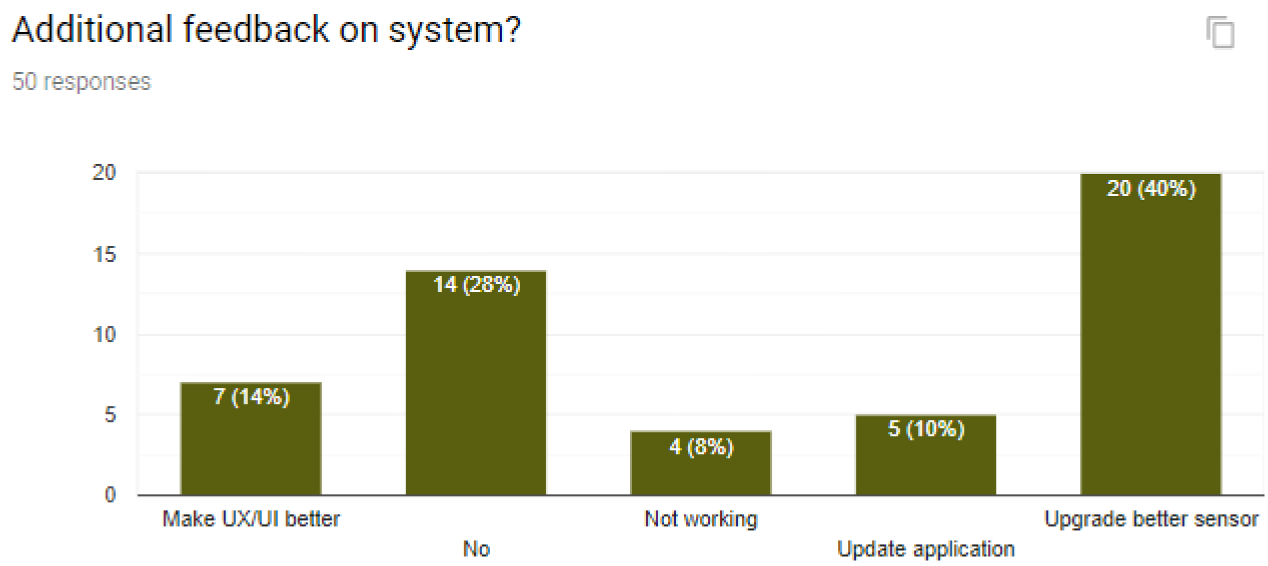
Picture 31. User satisfied result

Most survey participants have a good review when experiencing the GreeParking system.



Picture 32. Result of User's thinking this project will solve the problem

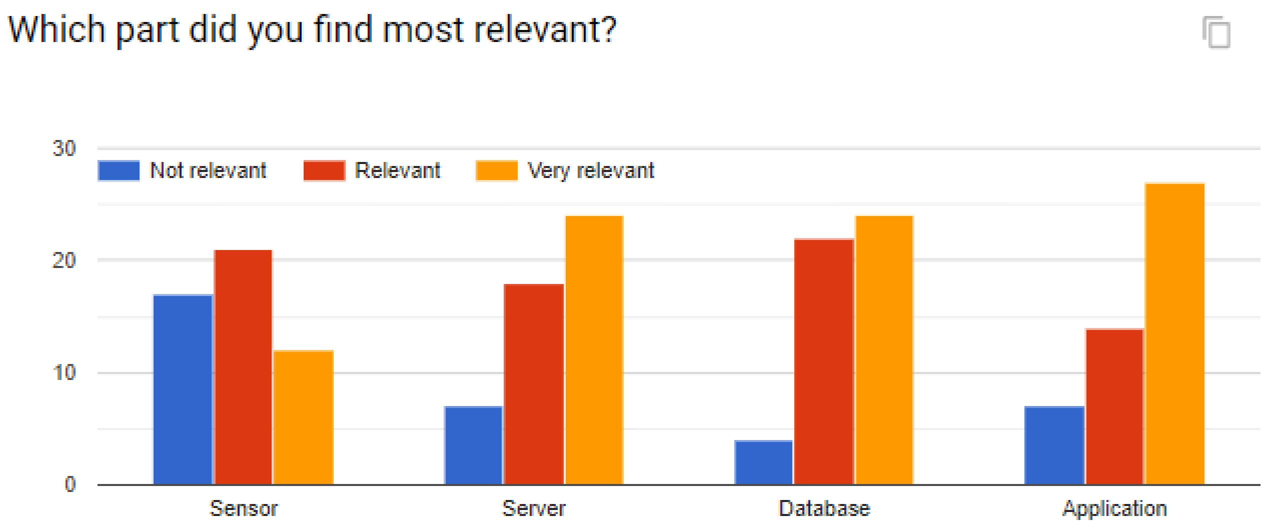
Regarding the problem of parking, users have felt how the system works to solve the problem, but besides that, there are still unsolved issues.



Picture 33. Feedback on system

Based on user feedback, the system needs to upgrade sensors and applications, and improve UX / UI so that users can use it more easily.

- The IR sensor used by the project is not guaranteed to be accurate. Instead of using Infrared (IR) Sensors, the next version of GreeParking will use Ultrasonic.
- Software and UX/UI to operate the system is used through the Blynk platform, so it has not yet provided a complete application to the user. The next version of GreeParking will create an application for Android phones
- The system transmits data via wifi network, because of that in some cases due to internet problems, the system will not work properly. In the next version, new hardware supporting the 3G network will be applied.



Picture 34. Which part did the user find the most relevant

Survey results show that: Server, Database and application are self-compatible with the system. Only the sensor that people think needs to be changed does not match this system.

1.2. IoT application evaluation board

Based on the user's rating (with a scale of 10) for 4 criteria such as Feasibility for practice, Ability to commercialize, Possibility of improvement and problem-solving. Whether the project can grasp feasible development and future development.

Evaluation Criteria	Car owners	Motorcyclists	Shop owners along the street	Other related people
Feasibility for practice	8.8	7.2	8.2	7.9
Ability to commercialize	8.9	8.3	8.0	8.2
Possibility of improvement	9.4	9.1	9.0	9.2
Problem-solving	7.7	7.3	7.9	7.7

Table 17. IoT Application evaluation board

- Feasibility for practice:

Based on the reviews from the GreeParking product experience, the ability to apply the product in practice is very good, meeting many real-life problems such as management, search, and bean related issues Parking.

- Ability to commercialize:

Currently, the car parks in the city do not have devices like GreeParking. So to meet supply and demand, GreeParking will be a solution for parking projects. Besides, to make the city smart, the application of technology like GreeParking is an indispensable part. Many customers target this technology.

- Possibility of improvement:

GreeParking products are products with the ability to improve to serve many aspects of life such as Optimized parking, Reduced traffic, Reduced pollution, Enhanced User Experience, etc. Because of this, IoT technology has the ability to improve very well by changing a few factors such as sensors, servers, etc. The system was able to solve various parking problems.

- Problem-solving:

For GreeParking, the parking problem is still not 100% solved, but with future improvements, GreeParking will solve more than parking issues.

2. Undertake a critical review and compare your final application with the original plan.

2.1. Evaluate the key points of the end-application

- **Feasibility for practice:**

The smart parking industry, on the basis of applications, can be classified into three types; transport transit, government, and commercial. The government application segment accounted for over 62% of the overall market share in 2015 and is predicted to continue its control on the market growth over the next seven years due to government initiatives in developed as well as developing countries.

The benefits of a Smart Parking Solutions are:

  - Optimized parking
  - Reduced traffic
  - Reduced pollution
  - Enhanced User Experience
  - New Revenue Streams
  - Integrated Payments and POS
  - Increased Safety
  - Real-Time Data and Trend Insight
  - Decreased Management Costs
  - Increased Service and Brand Image

- **Ability to commercialize:**

The high growth rate in the registration of new cars worldwide, with major boom from regional economies such as Asia Pacific (APAC), will open the window of opportunities for parking management business. The ongoing and upcoming smart city projects worldwide will create room for intelligent parking management systems. The global parking management industry is expected to grow at a Compound Annual Growth Rate (CAGR) of 11.4% from 2014 to 2019.

The parking management market is estimated to be at $5,025.9 million in 2014. The market is expected to grow in tandem with the growth in vehicle ownership and parking facilities development. Need for smooth traffic flow, business benefits to the parking site operators, and decreasing hardware and connectivity costs are the key drivers for the parking management industry.

The commercial application segment has high growth potential and is projected to gain traction over the forecast period as these places are the major attraction of tourists as well as local citizens. Such systems are increasingly adopted at corporate avenues, shopping malls, theaters, and sports complexes, among others where efficient management of parking is on high priority. High convenience and flexibility along with mobile payment options are projected to impel growth in the commercial segment. Vendors are accentuating on providing convenience to users and are continually deploying new applications with innovative features. The transport transit application segment accounted for over 9% of the overall market share in 2015 and is predicted to grow over the next seven years.

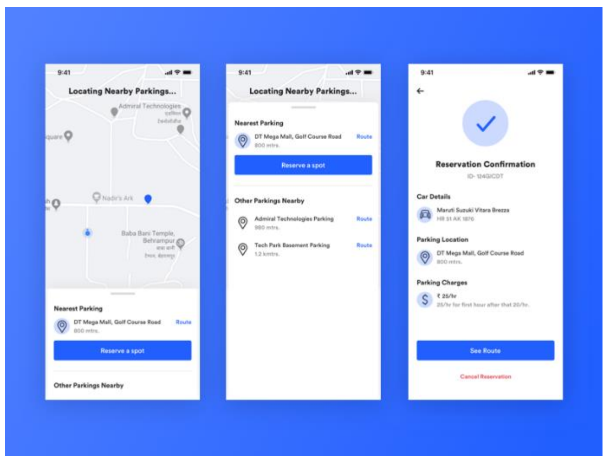
- **Possibility of improvement:**

The different types of services the solution could offer are:

- Parking Monitoring

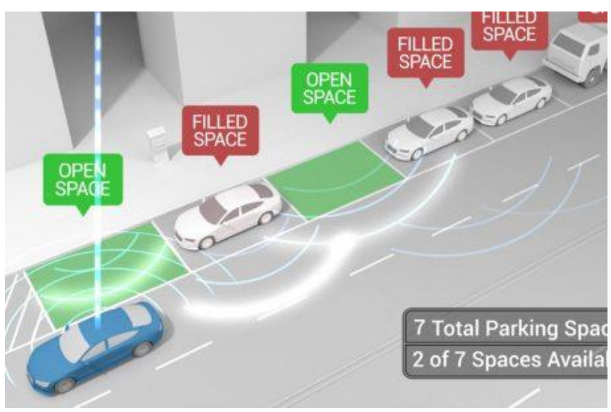
Picture 35. Parking Monitoring

- Parking Reservation



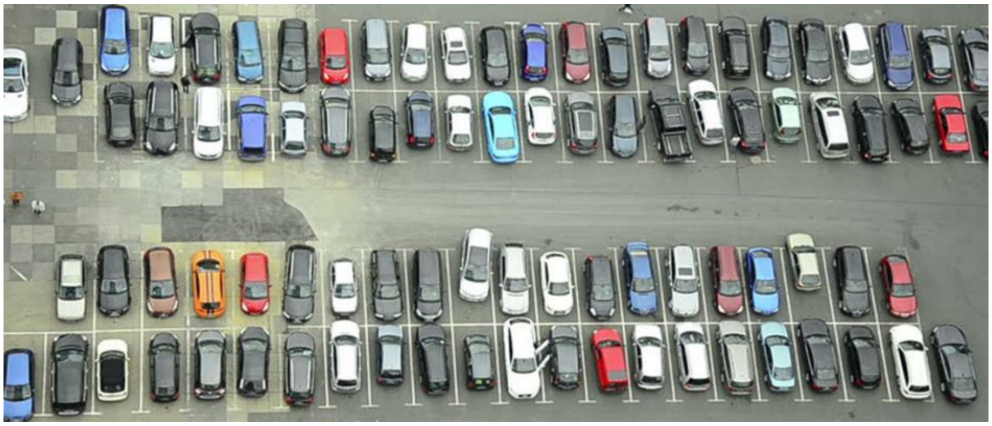
Picture 36. Parking Reservation

- On-Street Parking



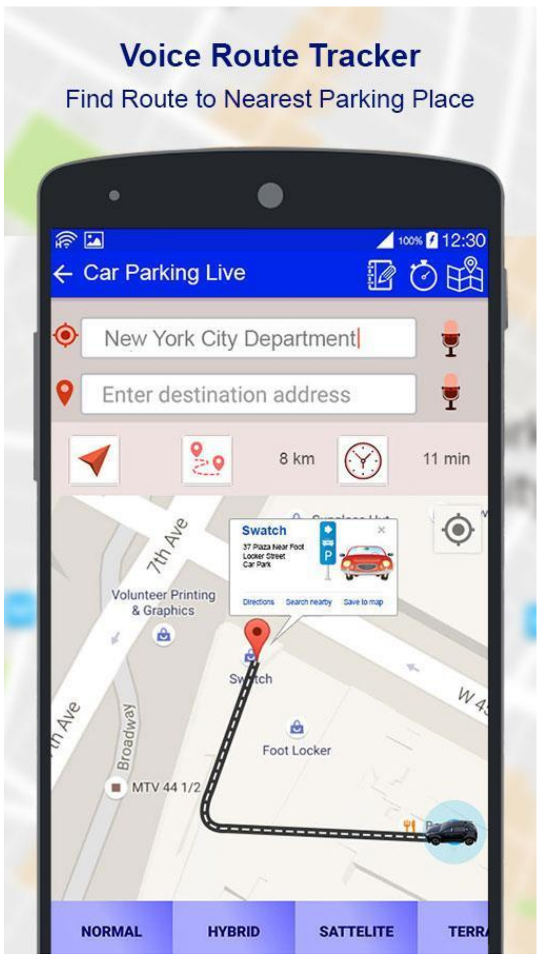
Picture 37. On-Street Parking

- Off-Street Parking



Picture 38. Off-Street Parking

- GPS Parking Directions



Picture 39. GPS Parking Directions

- Payment Methods

Picture 40. Payment methods

- **Problem-solving:**

Smart Parking is a parking strategy that combines technology and human innovation in an effort to use as few resources as possible such as fuel, time and space to achieve faster, easier and denser parking of vehicles for the majority of the time they remain idle.

Smart parking refers to the use of sensing devices to determine occupancy at the space level or at the lot/structure level. Sensing devices can refer to cameras, counting equipment like gates at the entrance of a lot, or sensors that are embedded into the pavement of individual parking spaces, to name a few.

Traffic congestion caused by vehicles is an alarming problem on a global scale and it has been growing exponentially. Car parking problems are a major contributor. Searching for a parking space is a routine activity for many people in cities around the world. This search burns about one million barrels of the world's oil every day.

A new survey from IBM has confirmed what city managers, travelers, and commuters already know from everyday experience: finding a parking space in a big city can be a frustrating and sometimes futile chore. And the problem isn't confined to the U.S. it's an issue around the world.

More than half of drivers among 8,000 commuters in 20 cities worldwide said during the past year they gave up at least once when looking for a parking space, and, one-fourth of them admitted they had argued with someone about a parking spot. New York City, Los Angeles, and Chicago each were represented by 400 commuters in the survey. (infiniteinformationtechnology, n.d.)

2.2. Compare problem in the first GreeParking and current GreeParking

Version	Problem	Solution
Greeparking 1.0	Sensor	Change from Infrared (IR) Sensors to Ultrasonic.
	Application	Change application from Blynk to Greeparking application on Android.
	Networking	Change module Nodemcu esp8266 wifi module to Board SIM800C intoRobot Fox.
Greeparking 2.0	Sensor	Upgrade Waterproof sensor.
	Application	Create application for IOS phone.
	Database	Upload data on Greeparking's cloud.
	Networking	Testing using LoRa to transfer data.
Greeparking 3.0	Energy	Use solar energy

Table 18. Table version of GreeParking

Item	Description
Board	Instead of using NodeMCU and Arduino devices depending on Wi-Fi network conditions, the system can use boards with 3G, 4G, LTE connectivity.
Sensor	Use IOE-SR05 distance measurement sensor to replace IR sensor to enhance display results
Application	The system is using Blynk to communicate data to users, which does not guarantee the security of information. To improve future products, GreeParking will design a separate application, server, database to meet the above needs.
Waterproof	Solutions to prevent the device from working under rain conditions have not been developed. Therefore, GreeParking will have a waterproof case.

Table 19. Table of the item that the system need to do in future

2.3. Plan for the next product cycle

The main features that might be improved are:

- LoRaWAN protocol will increase the autonomous operating time of a smart parking system. Due to LoRa Alliance's specification, the LoRaWAN protocol is optimized for low power consumption, which means that you don't need to change batteries more often than once every 2-5 years. Furthermore, it is designed to scale from a single gateway installation up to large global networks with billions of devices. Thus, it would be a good idea to apply this or any LPWAN protocol for a smart park.
- Data Science and Computer Vision technologies based on video stream are also a perfect match for IoT-based innovations. On the subject of smart parking development, Data Science and Computer Vision might be applied for car number recognition.

This feature allows:

- Automatic gate opening, if the car belongs to a given park
- Navigation of cars to available and suitable parking slots

Sensor integration is also a good upgrade for smart parking systems:

- Ultrasonic sensors: Ultrasonic sensors measure distance by using ultrasonic waves in time between the emission and reception. Advantages: high accuracy of the sensor. The disadvantages: the probability of sensor contamination.
- Electromagnetic field sensors: Electromagnetic field sensor is a small-scale microelectromechanical system device for detecting and measuring magnetic fields. The solution is based on the change of the electromagnetic field as the metal mechanisms approach one another.

Custom mobile and web IoT applications might be developed depending on business needs. The function of a parking slot order may be a killer feature for mobile/web apps. If a parking area is a few kilometers long, an integrated AR navigation may help drivers not to get lost looking for own car.

Cities, parking operators and car park owners should be looking to achieve long term, flexible partnerships with their suppliers. All stakeholders in the value chain should be seeking a mutual relationship with each other that allows for flexibility in parking provision and use of technology to create efficiencies, build better services and maintain customer satisfaction.

3. Critique the overall success of your application.

3.1. How GreeParking solve the problem?

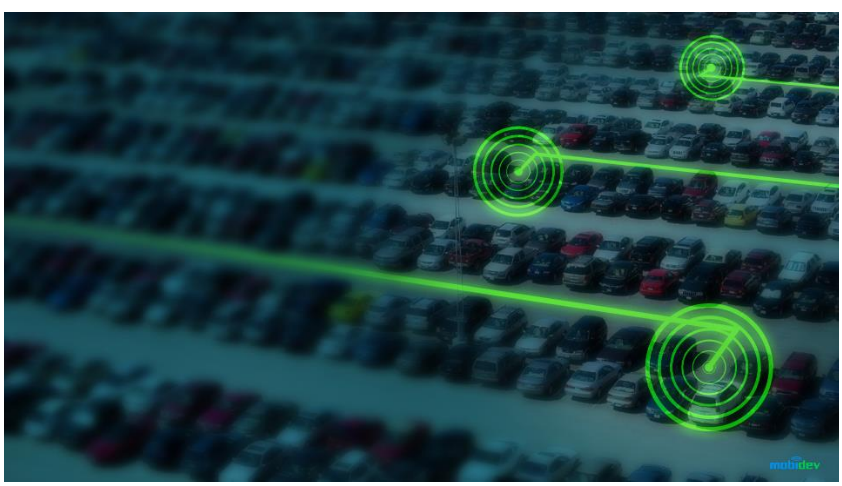
Smart parking development implies an IoT-based system that sends data about free and occupied parking places via web/mobile application. The IoT-device, including sensors and microcontrollers, is located in each parking place. The user receives a live update about the availability of all parking places and chooses the best one.

In order to investigate technologies behind the smart parking solution, we implemented an internal research project. The main idea was the creation of smart parking using the Internet of Things and ultrasonic sensors, where available parking places could be displayed in a web application.

The IoT device consists of an ESP8266 microcontroller and an IR sensor. The sensor periodically measures the distance and transmits this data to the Blynk (microcontroller, which is connected to AWS IoT service via the MQTT protocol).

Picture 41. Setup IR sensor

Periodically, the device sends measurements to the cloud where they are stored in Blynk IoT shadow as a sensor state. A sensor detects a parked car by measuring the distance to the nearest obstacle in our case, to the bottom of the car. The state can be "occupied" if the distance is in the range of 100mm 500mm, "free" if the distance is more than 500mm and "dirty" if the distance is less than 100mm, which means that the sensor may be unclean. All of these values can be easily configured. These values represent both free and occupied parking lots.



Picture 42. Detect car with GreeParking. (mobidev, n.d.)

The goal of this project is to create a user-friendly and adaptable system that can be implemented in large, multi-level parking garages in order to alleviate parking hassles. The ultimate goal is that the ideas and planning demonstrated through this system can then be easily upgraded to an actual parking facility. The purpose of the development of IoT based parking system is to eliminate the unnecessary frustration

drivers experience as they waste priceless minutes circling parking garages looking for a slot. In addition, traffic flow within the area will be better regulated, creating a safe atmosphere for both drivers and other people.

How the system can do?

- To accurately predict and sense slot/vehicle occupancy in real-time
- Guides residents and visitors of the city to available parking
- Simplifies the parking experience and adds value for parking stakeholders, such as drivers
- Help traffic in the city flow more freely leveraging IoT technology
- Plays a major role in creating a better urban environment by reducing the
- emission of carbon dioxide and other pollutants
- It enables better and real-time monitoring and managing of available parking space

Overall benefits include:

- Gives the drivers information about parking spaces available
- Eliminates the hassle of circling a parking garage or area in search of a parking spot
- Helps the traffic flow because of less illegal parking
- Saves resources and time
- Allows drivers to reserve parking slots before arrival
- Creates a way for further advancements and research in the sector

Going to the bigger picture, systems like this can go well beyond avoiding the needless circling of blocks. They can also enable cities to develop fully integrated multi-modal intelligent transportation system that doesn’t rely on cars in the first place.

3.2. The potential impact on people, business, society and the end-user

The GreeParking's designs have been tested both in simulation and practically, which led to design revision because designs on paper and real-world experience are quite different. When starting this project, the project manager has done intensive research about the subject matter and even considered other methods of implementing such as system. Finally, the system should start with components that can be found locally. The system needs to fit into the economic reality that faces in daily life.

Upon designing the system, The analyze's Greeparking from a business point of view which know as SWOT.

SWOT analysis is an acronym for strengths, weaknesses, opportunities, and threats and is a structured planning method that evaluates those four elements of an organization, project or business venture. A SWOT analysis can be carried out for company, product, place, industry, or person. It involves specifying the objectives of the business venture or project and identifying the internal and external factors that are favorable and unfavorable to achieve that objective.

Strengths
There is a greater sense of security due to the fact that patrons do not actually walk to and from their own space.
It is highly feasible for extremely small sites that are unable to accommodate a conventional ramped parking structure.
There is high parking efficiency.
There is no need for driving while looking for available space.
There is a minimal staff requirement if it used by known parkers.
It is possible that the retrieval time is lower than the combined driving/parking/walking time in conventional ramped parking structures

Table 20. SWOT - Strengths

Weaknesses
There is a greater construction cost per space.
Use of redundant systems will result in a greater cost.
It may be a bit confusing for unfamiliar users
It is not recommended for high peak hour volume facilities.
There may be a fear of breakdown
It may require maintenance.

Table 21. SWOT - Weaknesses

Opportunities
The sector is not exploited yet.
The problem domain in question is that can be related to a day to day life.
The project has a small initial investment compared to its final output.
It attracts potential investors.

Table 22. SWOT - Opportunities

Threats
Commercial products
Delivering the project as a fully-fledged commercial product in time.
Trendy factors
The country's economic status

Table 23. SWOT - Threats

Business Models:

The deployment of a smart parking service offers cities and parking operators the opportunity to review their business model and potentially take steps to approach parking and the data it generates in different ways. Smart Parking enables the city, value chain partners, citizens, and other stakeholders to approach parking in a different way, and this could be reflected in the services financial model and thus requirements. Partners such as app providers and vehicle manufacturers offer new ways to monetize parking provision and allow the cost burden to be spread in more ways than today. The data available from a smart parking service opens up new charging models and value-added services, many of which could be made accessible through integration into various subscription and access models. Innovation is driving change in smart cities and smart parking is no exception. The technology will drive new premium parking, vehicle integration, and parking-as-a-service models. Disrupting the current market model should be a consideration of the city of the future, and the city and parking operators must be able to take advantage when the opportunity arises.

The financial model for smart parking solutions can be innovative in its approach to the reasons outlined above. Budget constraints can be addressed through alternative funding models, with grants, asset leasing, and smart investment models all being used to alleviate traditional funding constraints.

As the service grows it is important to use the data generated to create new insights that can provide financial and service remodeling opportunities. For example, data sharing with other departments can mean that congestion charging could be linked to parking availability, meaning new revenue potential and cross-departmental budget allocations can be altered in favor of one or the other. (gsma, n.d.)

Establish:

Cities should move quickly to establish a trusted system with incremental deliverables. It is best not to focus on all aspects of the service at once, but ensure that processes can come online in an incremental fashion so that they can be perfected without impacting on rollout timescales. By phasing the approach, the city can increase confidence in the systems being deployed and ensure that citizens and stakeholders engage with and support the changes.

Scale:

During the phased growth phase, the program may still only be focussed on one site or one service. Once enough confidence has been gained in the system and the required processes have been proven to operate effectively in a live environment, the system can be scaled across multiple sites and open data APIs published to allow developers and other stakeholders to start building innovative solutions. At the point of scale, the city or parking operator can also start to introduce new concepts to drivers and citizens that smart parking allows such as dynamic tariffs or booking of parking spaces. Engagement with partners and citizens should continue, and new opportunities investigated as they arise. Partners such as mobile operators may be able to offer new scope to relationships such as the use of brands and integrated services or apps.

“Starting small” is a good way to get up to speed with new methodologies whilst reducing the risk of mistakes which are difficult to reverse. Building upon this initial step with a series of iterative changes as the project grows will ensure that the full commercial service is more likely to be sustainable and deliver citizen engagement. The core stages of growth for a smart The parking service are outlined below:

- Identify the role of landowners and parking operators
- Research what triggers are needed to change driver behavior
- Compile data on current parking usage
- Understand how other departments can benefit
- Understand learnings from other cities
- Ensure parking cameras, sensors and payments are interoperable
- Test sensor performance and durability
- Sanity check to ensure system can be delivered
- Obtain feedback on service from users
- Understand the system constraints for future modelling
- Undertake data integration with the billing system
- Develop collaboration tools for parking managers
- Launch live, integrated smart parking service to the public
- Integrate parking with other city data
- Educate the public in new parking service
- Form partnerships with parking app providers
- Feedback benefits to stakeholders
- Establish baseline parking parameters
- Put in place clear accountability and governance structures
- Provide a clear line of site between investment and outcome
- Document parking successes and gains

3.3. The problems might it encounter when integrating into the wider IoT ecosystem?

Application with functional and non-functional requirements:

	Requirement
Parking attendant App	The app shall let the parking attendant login successfully
	The app shall allow a parking attendant to enter multiple numbers of occupied slots at a time
	The app shall allow the parking attendant to see the reservation list
	The app shall allow the parking attendant to submit payment of a driver
	The app shall allow the parking attendant to sign out successfully
	The app shall be easy to use
	The app work without crashing
	The app shall return the information it gets instantly
	The app shall quickly send data when submitted
User app	The app should let the first-time user sign up
	The app should allow user to sign in successfully
	The app shall allow user to add plate numbers of car they own
	The app shall allow the user to manage account
	The app shall allow the user to check paid and unpaid payments
	The app shall allow the user to pay for unpaid services
	The app shall allow a user to pick a destination
	The app shall allow user to get map guidance

Table 24 Functional and non-functional requirements.

Implement server:

All the data incoming from different IoT hardware set up will be sent to the central server.

A server is a computer designed to process requests and deliver data to another computer over the internet or a local network. The word "server" is understood by most to mean a web server where web pages can be accessed over the internet

through a client like a web browser. However, there are several kinds of servers and even local ones like file servers that store data within an intranet network.

Although any computer running special software can function as a server, the most typical use of the word references the very large, high-powered machines that function as the pumps pushing and pulling data from the internet.

Most computer networks support one or more servers that handle specialized tasks. As a rule, the larger the network - in terms of clients that connect to it or the amount of data that it moves. The more likely it is that several servers play a role, each dedicated to a specific purpose.

Strictly speaking, the "server" is the software that handles a certain task. However, the powerful hardware that supports this software is also usually called a server because server software coordinating a network of hundreds or thousands of clients requires hardware much more robust than what you'd buy for ordinary consumer use.

Outdated hardware and software:

Since the IoT devices are being used increasingly, the manufacturers of these devices are focusing on building new ones and not paying enough attention to security.

A majority of these devices don't get enough updates, whereas some of them never get a single one. What this means is that these products are secure at the time of purchase but becomes vulnerable to attacks when the hackers find some bugs or security issues.

When these issues are not fixed by releasing regular updates for hardware and software, the devices remain vulnerable to attacks. For every little thing connected to the Internet, regular updates are a must-have. Not having updates can lead to data breach of not only customers but also of the companies that manufacture them.

Malware and ransomware:

The rapid rise in the development of IoT products will make cyberattack permutations unpredictable. Cybercriminals have become advanced today and they lock out the consumers from using their own device.

Predicting and preventing attacks:

Cybercriminals are proactively finding out new techniques for security threats. In such a scenario, there is a need for not only finding the vulnerabilities and fixing them as they occur but also learning to predict and prevent new threats.

The challenge of security seems to be a long-term challenge for the security of connected devices. Modern cloud services make use of threat intelligence for predicting security issues. Other such techniques include AI-powered monitoring and analytics tools. However, it is complex to adapt these techniques in IoT because the connected devices need processing of data instantly.

Data protection and security challenges:

In this interconnected world, the protection of data has become really difficult because it gets transferred between multiple devices within a few seconds. One moment, it is stored in mobile, the next minute it is on the web, and then the cloud.

All this data is transferred or transmitted over the internet, which can lead to data leak. Not all the devices through which data is being transmitted or received are secure. Once the data gets leaked, hackers can sell it to other companies that violate the rights for data privacy and security.

Furthermore, even if the data doesn't get leaked from the consumer side, the service providers might not be compliant with regulations and laws. This can also lead to security incidents. (readwrite, n.d.)

CONCLUSION

Smart parking offers new ways to monetize data generated from drivers and spaces. Mobile operators can discover more meaningful intelligence about a cities' parking usage enabling a greater range of value-added services. This includes anonymized data about where people begin their journeys, not just where they end them, how many people travel together and where people visit after they have parked in certain locations.

Smart parking can be enabled today by Mobile IoT from mobile operators. Mobile operators are strong, low risk, long term partners, well placed to meet all the needs of a smart parking service Secure communications network and management platform, access to open data and integration of payment and billing systems. Mobile operators and Mobile IoT are also future-proofed, as they are based on international standards with a roadmap towards integration with future networks and future smart cities' needs. Mobile IoT also operates in licenced managed spectrum, so it is a robust, scalable choice for all of cities or parking operator needs.

Cities initiating smart parking investigations need to consider their mobile operator as a core partner and work alongside them to scope and implement smart parking. Mobile operators can share their experience of previous deployments, offer economies of scale and understand the intricacies of how to deploy in different environments. All parties in the value chain can benefit from having a mobile operator at the core of a program, as the data generated can be controlled and managed throughout the value chain in a consistent, accessible and secure manner.

Smart parking offers a city or parking operator new ways to engage with the public and creates important secondary benefits including economic growth and reduced traffic and pollution. The business model behind smart parking is maturing now to a point where it is achievable, affordable and beneficial to a city. By ensuring that the cities strategy around communications, data, and financing is robust, cities should move forward with their investigations and procurement of smart parking services.

References

components101. (n.d.). Retrieved from components101: https://components101.com/microcontrollers/arduino-uno

components101. (n.d.). Retrieved from components101: https://components101.com/16x2-lcd-pinout-datasheet

components101. (n.d.). *components101*. Retrieved from components101: https://components101.com/wireless/rc522-rfid-module

elecrow. (n.d.). *elecrow*. Retrieved from elecrow: https://www.elecrow.com/download/ESP-12F.pdf

infiniteinformationtechnology. (n.d.). Retrieved from infiniteinformationtechnology: http://www.infiniteinformationtechnology.com/iot-smart-city-what-is-smart-parking

iot-analytics. (n.d.). Retrieved from iot-analytics.: https://iot-analytics.com/smart-parking-market-report-2019-2023/

iotmaker. (n.d.). Retrieved from iotmaker: https://iotmaker.vn/board-sim800c-intorobot-fox.html

iotmaker. (n.d.). *iotmaker*. Retrieved from iotmaker: https://iotmaker.vn/nodemcu.html

mobidev. (n.d.). Retrieved from mobidev: https://mobidev.biz/blog/iot-based-smart-parking-system

nshopvn. (n.d.). Retrieved from nshopvn: https://nshopvn.com/product/cam-bien-sieu-am-ioe-sr05-ho-tro-ngo-ra-ttl/

readwrite. (n.d.). Retrieved from readwrite: https://readwrite.com/2019/09/05/9-main-security-challenges-for-the-future-of-the-internet-of-things-iot/

skyfilabs. (n.d.). Retrieved from skyfilabs: https://www.skyfilabs.com/project-ideas/automatic-smart-parking-system-using-Iot

xedoisong. (n.d.). Retrieved from xedoisong: https://xedoisong.vn/cong-nghe/can-canh-he-thong-bai-do-xe-tu-dong-dau-tien-tai-viet-nam-25586.html