A Dynamic Parking spot Allocation Based on web Application using RGB LED modules of Arduino

Jiwoo Park
Department of Computer Science
Chungnam National University
Daejeon, Republic of Korea
201702018@o.cnu.ac.kr

Jungkyun Woo
Department of Computer Science
Chungnam National University
Daejeon, Republic of Korea
201702042@o.cnu.ac.kr

Byeonggyu Kim
Korean Linguistic and Literature
Chungnam National University
Daejeon, Republic of Korea
201600068@o.cnu.ac.kr

Andy Lin
Information Technology
Purdue University
West Lafayette, United States
Andylin321123@gmail.com

Hazza Alkalbani
Cybersecurity
Purdue University
West Lafayette, United States
halkalba@purdue.edu

***Abstract*—Parking has been a problem in crowded city as it causes individuals to lose time due to unnecessary searching of parking spots. Finding a parking spot has been a problem for visitors due to limited parking spots that are only offered to faculties/student. Therefore, unlike previous studies focusing on humans to make parking easier, this paper can utilize a limited space by using dynamic parking spot allot which is a smart parking system manipulated by web Application. The system also includes Node-Red and RGB LED that can help users to check availability of empty parking spots. The primary objective of the research is to make parking lots more effective, reliable, and flexible for drivers. This case study will provide a solution that will allow better time management and space scarcity for users.**

***Index Terms*—Smart parking system, Web Application, Node-Red, RGB lighting, Dynamic allot**

I. Introduction

According to the CECI (*Computer and Enterprise Investigations Conf.*), US Number of Registered Vehicles data remains in active status [21]. It keeps increasing every year from 250 million to 280 million units between 2011 to 2019 [21]. Parking problems are correlated with the number of vehicles, as there is no perpetual movement of private vehicles [20]. As a result, overflowing vehicles require parking space and have to park most of the time. Parking spaces are limited in comparison to the increased number of vehicles, which causes two problems. According to Shoup, D. *et al.*, almost 30% of traffic are caused by people searching for parking spaces [23]. The problem in effect is in terms of scarcity (few seats compared to demand) or mismanagement (inefficient usage of available facilities) [8]. Prior to the smart systems, the parking administrator had to deal with each ticket one by one. However, with the smart system, these problems are no longer present. In recent times, not only checking a seats availability in real-time but also an unmanned ticket system contributes to eliminating existing wastes.

In many research studies, a smart parking system has been implemented to solve parking problems caused by overflowing vehicles. It is verified that the smart parking industry is expected to grow at a Compound Annual Growth Rate (CAGR) of 11.4% from 2014 to 2019 [7]. A smart parking system could save 220,000 gallons of fuel by 2030, as well as approximately 300,000 gallons by 2050, if successfully implemented in research [7]. Among smart parking systems, IoT (Internet of Things) is one of the most actively investigated technologies. An IoT system means that the network of physical objects 'things' communicates with each other through TCP/IP (Transmission Control Protocol/Internet Protocol) such as Bluetooth, WiFi, *etc*. [2]. With the help of instruments such as laptops, cellphones, and tablets, people can track things using the data that is shared by these physical objects of things. As the data will be updated immediately, users can take action if anything unexpected were to be detected. Implementing the real-time interchangeability, hence, helps users to prevent unexpected problems. Conducting the IoT field is inevitable to solve the problems of rapidly expanding automobiles which will enchant productivity and reliability.

While many smart parking systems have been developed, they have not yet been utilized in a parking space as they just indicate on how to park in vacant parking spots. In contrast to existing methods, this paper examines allocated parking spots that are available but dependent on the driver's status. The allocation of space will inevitably bring disproportion. If one of the facilities holds an event, the parking lot (e.g. for visitors) will be more crowded than usual. The visitor will be confused if there aren't any vacant seats even though there are visitors' spots available which is one of the reasons on why a dynamic system is needed. The dynamic system corrects the disproportion by adjusting the allocated parking slots based on RGB LED colors as each allocated parking space has a different RGB color. In order to accommodate the growing demand, unused spots which are not suitable for drivers will

be made usable. In addition, an LCD display and web page can display the availability of parking spaces in real-time, so that car owners can park right away without having to wait.

II. Related Work

Developing flexible allocation parking spots requires various equipment. This section compares and analyzes other papers to know how to work the best way to solve the parking shortage. Most smart systems make use of IoT, however, different devices are needed for different purposes. The IoT equipment is grouped into four major parts: mainboard, modules, server, and platform.

A. Mainboard

The most powerful mainboard of IoT systems is Arduino and Raspberry Pi. Those devices are the mainboard that can control other modules. As each device has pros and cons, they shall be used according to the technology they conduct. Arduino is used if the technology is simple such as classifying vehicles that can park [4] or notifying vacancy spots [11]. Those of the project do not use complicated operations yet simple control of the external device. On the other hand, there is a technology that analyzes the nearest spot with an algorithm [3] and helps parking security by recognizing the face [6]. What these technologies have in common is that they choose Raspberry Pi to use complicated calculations. To sum up, Arduino and Raspberry Pi have a balance between cost and performance. Arduino is cheaper but has low performance. On the contrary, the stronger Raspberry Pi has, the more expensive it is. Table. I clearly indicates their spec. This project needs a simple function so that Arduino will be used.

B. Modules

For operating necessary functions, adequate modules have to be followed. In other research, there are two main systems with an unmanned system or searching vacant spot systems. Unmanned system supervises entering and exiting vehicles and it will manage the data on those of vehicles using RFID (Radio-frequency identification) module [6] [18]. The other way, searching for a vacancy, verifies the empty parking spot using IR (Infrared) sensor or UR (Ultrasonic) sensor and marks with LED light [3] [16]. However, both systems use LCD display modules to show real-time parking lot status. The RFID module is excluded in this paper as it does not manage entering and exiting vehicles. Additionally, an IR sensor is adopted Between the IR sensor and UR sensor that notifies parking spot status. IR sensor can be connected to more devices as there are fewer cables to connect to another device. 16*2 LCD display module also will be used to display real-time information.

TABLE I
Spec differences between Arduino Uno R3 and Raspberry Pi 2 B

	Aruduino Uno R3	Raspberry Pi 2 B
Processor	AVR ATMEGA328p	Broadcom BCM-2836
Clock Speed	16 MHz	900 MHz
Register Width	8-bit	32-bit
RAM	2KB	1GB
GPIO Pins	32	40
I/O Current MAX	40mA	5-10mA
Power	7-12V 175mW	5V 600mW
Operating System	None	Linux & Others

C. Server

Due to the IoT characteristics, a server that shares real-time data is needed. The mainly used servers are Firebase, Node-Red, Thingsboard, and Cloud. Among them, Node-Red which is not only easily and organically wired to Arduino but also well compatible with devices will be used as a server.

D. Platform

Whatever technology is made the results eventually depend on how useful users feel this technology is. Users mainly access the technology using web or App, or both. For example, people use an App to register their vehicles and web is used for notifying seats availability from a parking lot map [5]. There are two reasons why we adopt only the web platform instead of the App. One is that the web platform is well compatible with Node-Red which will be used. Users can easily access the content without downloading is another reason. Even if it is implemented only on the web, users will have accessibility similar to that of the App.

III. Methodology

This section describes the method that is utilized to develop the system, which consists of two phases: Hardware Design and Software Design. The flow chart, Fig. 1 shown below, will help assist in understanding the process of the system. Initially, the system is accessed through a web page whose management can be limited to the administrator. In terms of initializing the system, the administrator sets the color of each parking spot by using the RGB LED light, to identify which parking spot belongs to whom. After the data of parking spots is changed, the system recognize duplicated values. If there is no duplicated values, the parking data will be updated as certain parking spot is changed to different color. In the hypothetical situation where all parking spots are set, an infrared sensor module is ready to detect the vehicle. Once the parking spot's Infrared Sensor module detects the vehicle's presence, it will send that information to the system. In response to these measurements, the system updates information and displays it on the user's web page and on the LCD display.

A. Hardware Design

The hardware that the system uses and their connections are described in this section. As shown in Fig. 2, the hardware consists of an Arduino Uno, 8 IR Infrared sensor modules, 8 RGB LED light modules, and a 16*2 LCD display module to implement the parking lot system. An example of how the hardware of the parking lot look like is demonstrated in Fig. 2. Within this model, one Arduino supplying power with USB is used to connect eight infrared and LED RGB modules so that can accommodate eight spots in addition to an LCD display situated near the entrance.

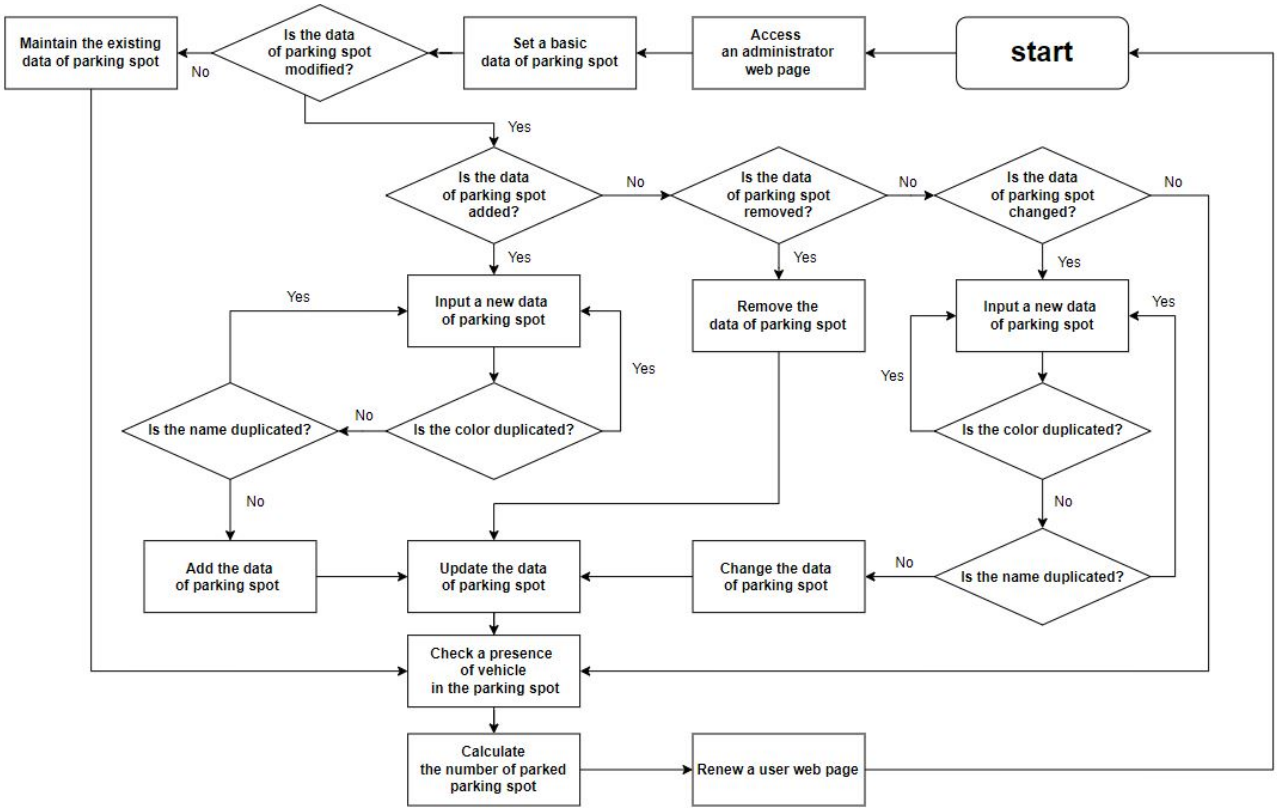

Fig. 1. Flow chart of the system

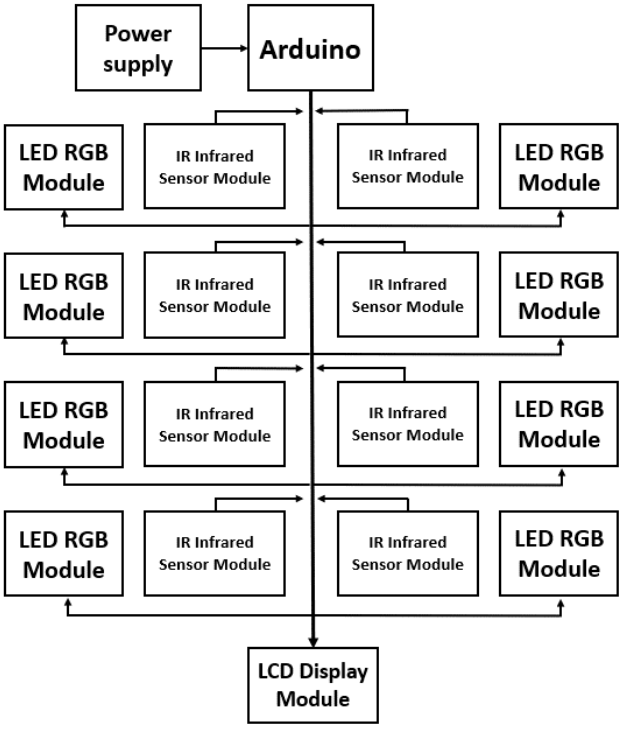

Fig. 2. Components with connection

B. Software Design

Software components include Arduino sketch, Node-Red, and a web application. These components will help to demonstrate how each hardware component interacts with other devices and how each software component processes data

a) Arduino Sketch: An Arduino sketch is an integrated development environment that is used by developers [16]. The sketch also compiles a source code that is uploaded to Arduino [16]. In order to set up the Arduino function, the uploaded code will handle the situation when the color of the light is changed or a vehicle is parked in the parking spot. As each module uploads value, the dynamic parking system is operated according to the code within the functions.

b) Node-Red: A Node-Red module is one of the IoT Platform tools for wiring together hardware devices; it provides an editor that can be used to wire multiple devices together [16]. Node-Red handles all events and contains ”Node” that can be dragged and dropped in the web environment [16]. In addition, the Flow (consisting of Nodes) will control how the system operates. Flows are segmented into three main categories: controlling modules, data communication, publishing web pages.

To begin with, we will use an RGB LED module in order to change the color of parking spots, which will be the main innovation of our system compared to previous other systems, and use an IR infrared module to detect vehicles' presence. With regard to the data communication flow, Node-

Red communicates with Arduino using serial communication, which utilizes one transmission line so that the system can continuously and sequentially send and receive data. As this method is more simple and cheaper than parallel communication, the system adopts serial communication. The final flow of publishing web pages is done by Node-Red, which publishes the web page to interact easily with IoT devices.

c) web Application: A web Application is the space in which users are able to share information, and the programming languages used in this application are JavaScript, HTML, and CSS. In order to prevent unauthorized access, the web page is divided into two web pages: administrator web page and user web page. In the administrator web page, an administrator initializes and updates the information about the parking spot. Updated information is displayed on the user web page and the user can notice whether there exists an available empty parking spot in real time. As users access the user web page from a range of devices, the screen should be optimized flexibly to meet their needs. In order to optimize, this page is responsive web design, as a web page will automatically adapt in terms of both the size of the screen as well as the grid size based on the screen type.

C. System Overview

The overall overview of this projects is processed as Fig. 3. After the aforementioned modules are installed on the Arduino, the Arduino and Node-Red are interconnected. Once they are linked, Node-Red controls all devices including web pages as the platform. On the web page that operates as a Node-Red, the manager can actively adjust who can use parking lot spots. Users who want to park in the parking lot also access to the web in order to check parking lot status.

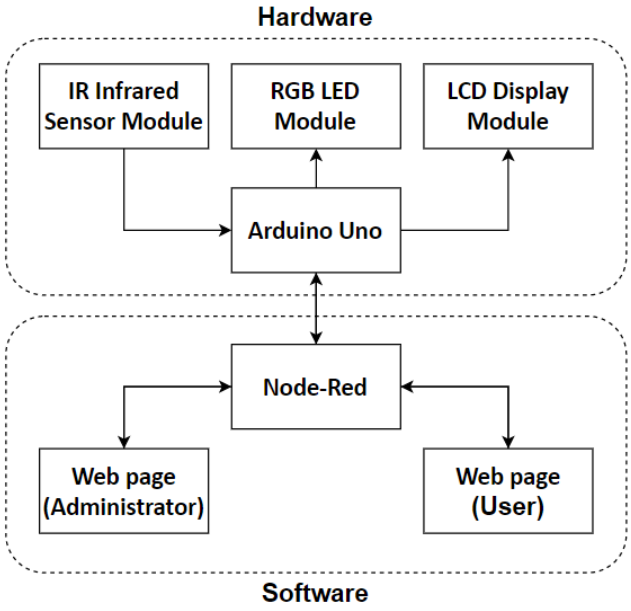

Fig. 3. Overview of the dynamic parking system

IV. Implementation

This section will demonstrate the implementation of a dynamic parking system in eight steps.

1) *Arduino circuit connection with IR sensor, RGB LED:* First step is connecting circuits of the infrared sensor and RGB LED modules to implement eight parking spots. In order to notify that all modules of Arduino are working well, this paper uses cad pages to avoid the breakdown of devices. The longer pins and shift register are additionally equipped to one Arduino. The result can be shown in Fig. 4.

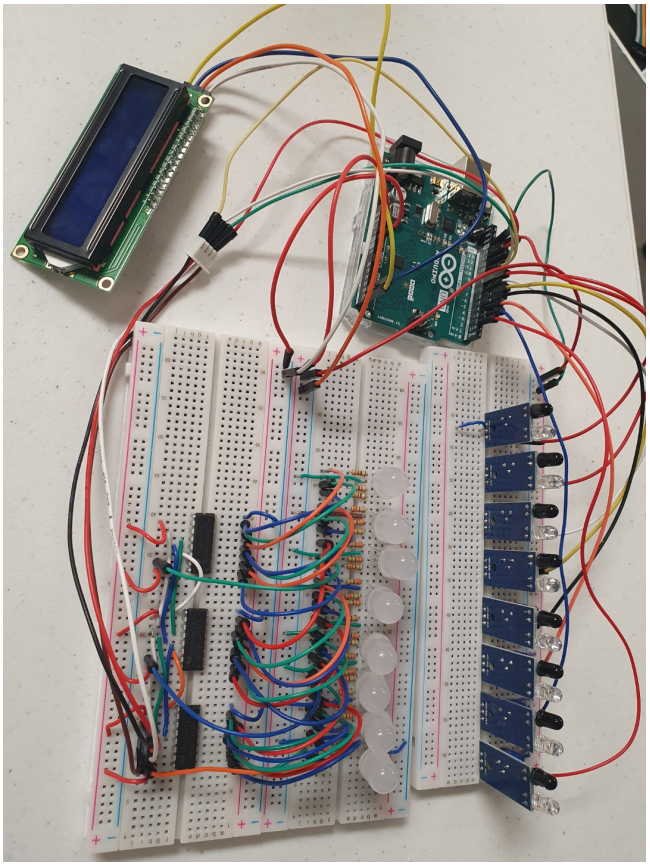

Fig. 4. Blueprint of Arduino circuit connection

2) *Insert function to Arduino using Arduino sketch tool:* Algorithm. 1 is the representative function code of the Arduino sketch. In the sketch, changing RGB LED colors and detecting obstacles functions are prepared. After the value of the serial message is divided into one syllable, in addition, the character transmitted through the iteration is output on the LCD display module. Once all of the code are compiled, those functions are built-in Arduino. Therefore, Arduino will be remotely controlled by connecting Node-Red without compiling separately.
3) *Receiving signal from Node-Red's node connecting Arduino and computer to USB:* Our project's next step is interconnecting devices to Node-Red. First, Arduino is connected to a computer with a USB to replace the

Algorithm 1 Arduino Sketch code for implementing RGB LED, infrared detection modules and LCD module

Set rgb_pin = Arduino pin connected to the rgb module
Set sensor_pin = Arduino pin connected to the ir sensor
Set number_of_RGB = number Of rgb pin
Set number_of_ir_sensor = number of sensor pin
Set color_array = index by number of colors
Set lcd_output = empty string
while $Arduino\ is\ on$ **do**
  if $if\ there\ is\ serial\ message$ **then**
    Set rgb_array to divided serial message
  end if
  for $iteration = 0, 1, \ldots, number\_of\_RGB$ **do**
    if $rgb\_array[i]\ ==\ 1$ **then**
      Red on rgb_pin[i]
      color_array[1] 1 increase
    else if $rgb\_array[i]\ ==\ 2$ **then**
      Blue on rgb_pin[i]
      color_array[2] 1 increase
    else if $rgb\_array[i]\ ==\ 3$ **then**
      Green on rgb_pin[i]
      color_array[3] 1 increase
    else if $rgb\_array[i]\ ==\ 4$ **then**
      Yellow on rgb_pin[i]
      color_array[4] 1 increase
    else if $rgb\_array[i]\ ==\ 5$ **then**
      Purple on rgb_pin[i]
      color_array[5] 1 increase
    else if $rgb\_array[i]\ ==\ 6$ **then**
      Cyan on rgb_pin[i]
      color_array[6] 1 increase
    else
      Turn off rgb_pin[i]
      color_array[0] 1 increase
    end if
  end for
  for $iteration = 0, 1, \ldots, length\ of\ color\_array$ **do**
    lcd_output += value of color_array[i]
    color_array[i] = 0
  end for
  Output lcd_output to lcd display
  lcd_output = empty string
  for $iteration = 0, 1, \ldots, number\_of\_ir\_sensor$ **do**
    output_serial += $sensor_pin$ **end for**

  Output output_serial as serial message
end while =0

battery and receives signals through Node-Red's node. The node with built-in Serial, which can communicate with devices, is created to interconnect with Arduino in the Node-Red's flowchart. The greenlight can be seen under the Serial node in Fig. 5 if devices are successfully linked. Above the Fig. 5, each Web pages for the manager and the customer are connected with Node-Red. The Arduino device is already connected to Node-Red, therefore, all of modules are controlled by using Web pages written by JS (Java Script) code.

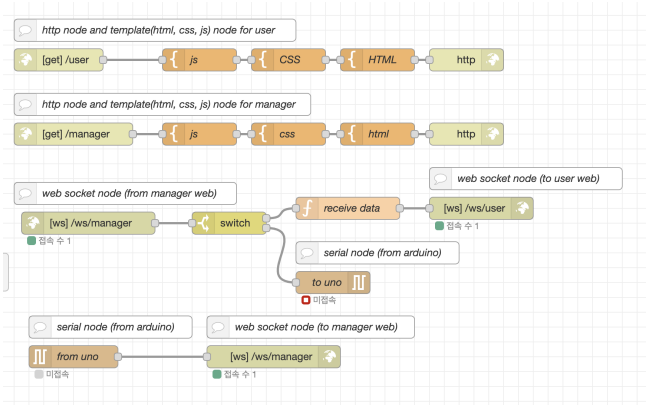

Fig. 5. Interconnecting node of Node-Red flowchart

4) *Making a web page for manager using DashBoard Node:* The parking lot manager adjusts which driver can park by using the web page as Fig. 6. The changed information will be immediately updated in the left side as soon as parking spots are set by color. When visitors' drivers crowd the parking lot due to an event, the manager can allocate more seats for visitors, as shown in Fig. 7. In addition, not only can additional spots be adjusted by adding different color but the color that represents who uses the spot can be changed.

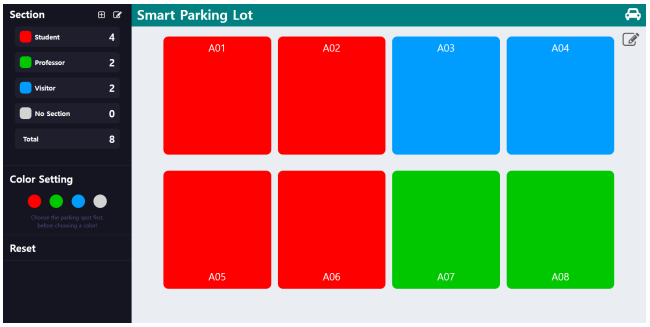

Fig. 6. The initial setting for parking lot spots

5) *Making a web page for users using web Node:* An example of what the user web page will look like in the Desktop or tablet mode is shown on the Fig. 8, and an the user web page on smart phone will be displayed on the Fig. 9. The customer can access our Web page using any devices to check seat availability in real-time.

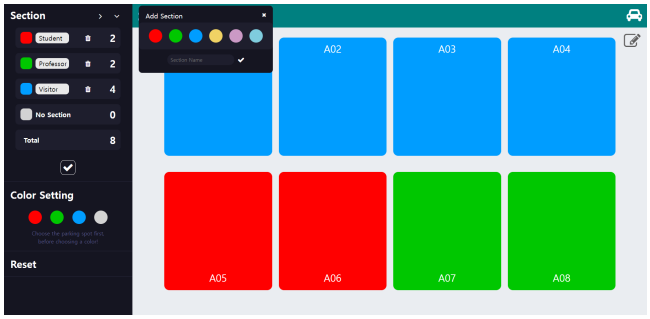

Fig. 7. Adjusting parking lot spots according to demand

Fig. 8. User web page in the desktop

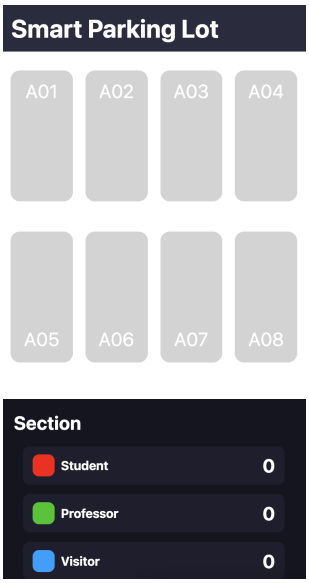

Fig. 9. User web page in the mobile

6) *Implementing parking lot sample:* In the Fig. 10, the sample model is manufactured in order to implement the system. The module use one Arduino supplying power with USB to connect eight infrared and LED RGB modules so that it can accommodate eight parking spots. As for the parking lot entrance, the LCD Display module is installed to check seats availability.
7) *Allocating each spot with RGB colors and Updating the information in LCD, web page:* In the Fig. 11, implemented technologies can be seen at a look in our system. All of the updated information are shared with each other in this system so that the system administrator controls parking lot status in the manageable page as well as the customer checks real-time parking lot status through user Web page.

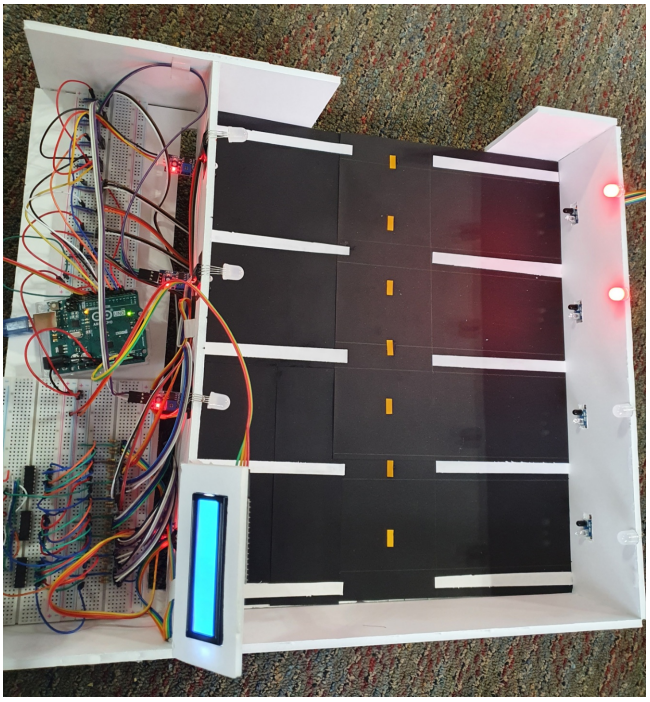

Fig. 10. Conducted parking lot model

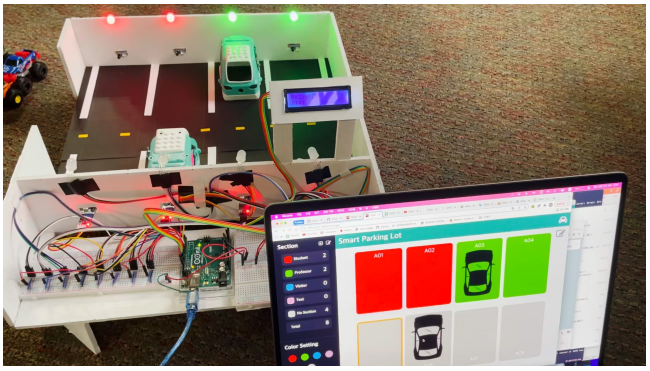

Fig. 11. Connection between the system and the sample model

V. Conclusion

This work aims in order to solve the parking problems by dynamically allocating parking spots. The dynamic parking system efficiently utilizes an existing space by changing unavailable spots not exclusive to the driver's status to available spots. Therefore, if visitors crowd the parking lot due to an event, the dynamic allocating technology will increase visitors' spots to accommodate the high demand of visitors.

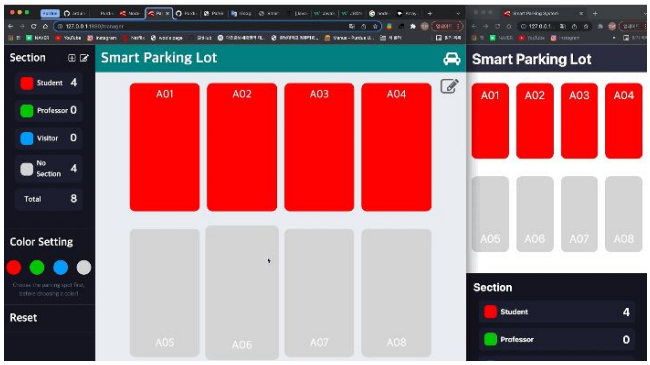

Fig. 12. Real-time update between two pages.

Although there are many intelligent systems, conducting this technology has two advantages compared to other research. The former is that our technology can be installed in any parking lot without changing existing facilities. It has good versatility and can save money and utilize limited space. The latter advantage is that users can check real-time parking lot information accessing the web page or watching LCD Display. As a result, unnecessary time searching vacant parking spots is reduced previously mentioned.

REFERENCES

[1] Arbatskaya, Maria, Mukhopadhaya, Kaushik, Rasmusen, Eric Bennett "The parking lot problem", *SSRN Electronic Journal, 2006*

[2] Soni, Vishal Dineshkumar, "IOT Based Parking Lot", 2018, *Int. Engineering Journal For Research & Development*, Vol.3, No.1, p.9

[3] M. Ramasamy, S. G. Solanki, E. Natarajan, and T. M. Keat, "IOT based smart parking system for large parking lot," *2018 IEEE 4th Int. Symp.posium in Robotics and Manufacturing Automation (ROMA)*, 2018.

[4] H. Chaudhary, P. Bansal, and B. Valarmathi, "Advanced car parking system using Arduino," *2017 4th Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, 2017.

[5] A. Anand, A. Kumar, A. N. Rao, A. Ankesh, and A. Raj, "Smart parking system (S-park) – a novel application to provide real-time parking solution," *2020 Third Int. Conf. on Multimedia Processing, Communication amp; Information Technology (MPCIT)*, 2020.

[6] Thangam, E. Cassin, Mohan, M., Ganesh, J., Sukesh, C. V., "Internet of Things (IoT) based smart parking reservation system using raspberry-pi", *Int. Journal of Applied Engineering Research*, Vol. 13, No. 8, p. 5759-5765

[7] *Digital Transformation - Bigdata, cloud, Security amp; Mobility Solutions*. [Online]. Available: https://www.happiestminds.com/services/. [Accessed: 23-Feb-2022]

[8] L. O. Joel, *Model and solutions to Campus Parking Space Allocation Problem*. Durban, South Africa: University of KwaZulu-Natal, 2013.

[9] Z. Pala and N. Inanc, "Smart parking applications using RFID technology," *2007 1st Annu. RFID Eurasia, 2007.*

[10] K. C. Mouskos, J. Tvantzis, D. Bernstein, and A. Sansil, "Mathematical formulation of a deterministic parking reservation system (PRS) with fixed costs," *2000 10th Mediterranean Electrotechnical Conf. Information Technology and Electrotechnology for the Mediterranean Countries. Proc. MeleCon 2000 (Cat. No.00CH37099).*

[11] Kadhim, Maher. Hassan, "Arduino Smart Parking Manage System based on Ultrasonic", *2018, Int. Journal of Engineering & Technology*, Vol.7, No.3.20, p.494-501

[12] Faheem, S. A. Mahmud, G. M. Khan, M. Rahman, and H. Zafar, "A survey of intelligent car parking system," *Journal of Applied Research and Technology*, vol. 11, no. 5, pp. 714–726, 2013.

[13] A. Khanna and R. Anand, "IOT based Smart Parking System," *2016 Int. Conf. on Internet of Things and Applications (IOTA), 2016.*

[14] V. Hassija, V. Saxena, V. Chamola, and F. R. Yu, "A parking slot allocation framework based on virtual voting and adaptive pricing algorithm," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5945–5957, 2020.

[15] "M. S. Salman, M. N. Karsiti, and N. A. Rozly-Azni, "Dynamic Resource Allocation Strategy for low cost smart parking system," *2018 2nd Int. Conf. on Smart Sensors and Application (ICSSA)*, 2018.

[16] M. S. Bin Mohd Nazri, T. Long Alif Faiqal Bin Tengku Long Gaafar, H. Sofian, and A. A. Bakar Sajak, "IOT parking apps with car plate recognition for smart city using node red," *2020 11th Int. Conf. on Information and Communication Systems (ICICS)*, 2020.

[17] S. Park, S. Seo, B. Lee, J. Byun, and S. Park, "An energy efficient smart LED lighting system for Building Energy Management," *The 18th IEEE Int. Symp.posium on Consumer Electronics (ISCE 2014)*, 2014.

[18] J. Zhou, S. Li, H. Zou, and Q. Yang, "Design of truly unmanned smart parking lots based on Xinghai IOT platform," *2021 7th Int. Conf. on Computer and Communications (ICCC)*, 2021.

[19] Khanna, Abhirup., Anand, Rishi, *"IoT based smart parking system", 2016 Int. Conf. on Internet of Things and Applications (IOTA)*, p. 266-270

[20] H. E.-D. Ibrahim, "Car parking problem in urban areas, causes and solutions," *SSRN Electronic Journal, 2017.*

[21] "United States number of registered vehicles," US Number of Registered Vehicles, 1910 – 2021 — CEIC Data. Online. Available: https://www.ceicdata.com/en/indicator/united-states/number-of-registered-vehicles.

[22] "Red," Node. [Online]. Available: https://nodered.org/.

[23] Shoup, D., and Campbell, H. Gone parkin'. The New York Times Available Online: http://www.nytimes.com/2007/03/29/opinion/29shoup.html, 2007.