IoT based car parking management system with an application program interface

prosper philip¹, sathish kumar selvaperumal², ravi lakshmanan³, Chandrasekharan nataraj⁴
¹,²,³,⁴Faculty, of Computing, Engineering and Technology, Asia Pacific University
E-mail:¹pnjuma@yahoo.com, ²dr.sathish@apu.edu.my, ³ravi@apu.edu.my, ⁴chander@apu.edu.my

Abstract—With the population on the rise, urban areas congestion continues to become a major transportation problem. As number of cars increases, the demand for parking spaces rises, thus causing insufficient availability of parking spaces. This paper presents a new cloud based parking management system implemented on the Internet-of-Things technology network architecture that increases user convenience and parking management. Drivers are able to view parking availability online, and proceed to reserve a parking space. The parking system is designed based on zone control architecture using NodeMCU, thus allowing easy management and implementation. An eight hexadecimal character is generated for both entry and exit access codes, and sent to the driver’s email and mobile phone via SMS. Parking information are relayed and stored in the server using RESTful API implemented using Laravel PHP technology. Parking spot information like entry and exit times can be requested through the API.

Index Terms—Internet, smart, system.

1. INTRODUCTION

Population growth is a great concern in urban areas. As the population grows, the number of car also increases and the demand for parking spaces rises. Even though more parking facilities are built, there are still several problems in utilization. A major problem involves the time wasted searching for parking places, which not only causes road congestion and irate drivers searching for empty parking spaces, but also has a big negative impact on the environment.

Clearly, a solution is needed that will be hassle free for the parking users. The solution proposed here allows the driver to make a reservation online. The system will be monitored in a real time and will allow users to make reservations for parking as well as make payments online. [1], [2]

For example, in Israel in 2016, the average time spent looking for parking is twenty (20) minutes each time they park, fifty-five hours (55) is the average time a driver wastes each year looking for parking. Not only is time wasted searching for parking, but thirty (30) percent of urban traffic is caused by drivers searching for parking and four point eight billion (4.8) hours are wasted in traffic. [6]

In addition to the time spent searching for parking, three point nine (3.9) billion gallons of fuel every year is wasted and this in turn causes pollution to our environment. [6] In a current survey, researchers have found that in one year, searching for parking has created the equivalent of thirty-eight (38) trips around the world, burning 1.7 lakh litre of fuel and producing 730 tons of CO₂. [9]

According to [6] about thirty-eight thousand and three hundred (38,300) drivers are killed, four point four (4.4) million are seriously injured searching for parking space.

Parking systems based on plate number recognition have major problems considering that plate numbers are different in terms of size, colour and type depending on countries. These systems are restricted to seven alphanumeric characters, of which four are numbers and 3 letters, and can only analyse the plate number with single row plate number. [8] Hence, there is a need to develop a system that is not dependent on the plate numbers to recognize the vehicle.

Parking systems that utilize image-processing techniques are expensive because cameras are costlier than sensors. They also have larger storage and processing requirements and more specifications. [3] These systems are also susceptible to weather conditions, which confuse the images; however, this problem can be addressed by filtering the images in a high-quality transform. [1], [2] Therefore, there is need to design a system that is robust, affordable and that is not impacted by changing weather conditions.

Most of the existing parking systems have a management structure that uses ticket payment and vehicle guiding, which results into employing many full-time management personnel and high maintenance and management cost. [7]
Many people tend to forget their parking space location, especially when they park their car in unfamiliar lots or when the parking lot is big and this causes frustration and inconvenience to the parking user.\textsuperscript{[7]} It is always a problem for parking managers to decide on a system implementation because of fear of increased cost, time and integration issues with their current system. This result in them taking a slower approach to replacing old systems, which prolongs the problem\textsuperscript{[4]}. This system aims to minimise integration issues and help foster acceptance at a comfortable pace, hence properly managing the changeover process.

There is also wasted space in some current car parking locations. In these locations, there are more parking spaces per vehicle, and most are empty most of the time. The need, location, and cost must be considered and convenience-driven to increase sustainability\textsuperscript{[5]}.

Although smart parking solutions have been successful, there are no citywide solutions across the disjointed public and private parking suppliers, thus making them uncoordinated\textsuperscript{[10]}. If occupancy data exists, they lack standardization or accessibility and do not have an integration API that would allow access of these data. Hence, there is the need of having a universal API to make data more accessible and standardized\textsuperscript{[11]},\textsuperscript{[12]}.

2. PROPOSED METHODOLOGY

2.1. Block Diagram

The designed system is composed of five levels, these are: sensor, network, middleware, application and mechanical. The sensor and mechanical level exists in the parking lot, where they’re used to detect a vehicle, and control entry and exit. Communication and transmission of information to the servers is done wirelessly, and the middleware act as a bridge between the two systems. Fig.1 depicts critical functionalities of each sub-system.

The client parking system primarily handles vehicle detection, uploading of lot status to server, and configuring the LEDs. The parking system server receives these information and processes it to determine the total available parking in the specified location. The information is also saved as historical data, from which can be retrieved using the API.

The reservation system allows the user to make reservation, as it assigns an available parking spot, along with entry and exit access codes to be used at the entrance gate. Confirmation of the reservation is sent via SMS and email to the specified user mobile number and registered email.

The gate control system monitors and allows input of the access codes using a keypad. Verification and validation of the code is done by querying the database server. A result is returned back to the controller; if the status result is “1” the boom gate opens, else if “0” the gate remains closed whilst prompting a message to the user.

2.2 Constructional details

This section describes the production and communication protocol processes for the project. In order to ensure that all components and system algorithm is working, the system is designed and simulated using PROTEUS. After verification and validation of the simulation process, the system is then assembled and tested on a breadboard. When all system components and tests are finalized, the system is then implemented on a PCB and final connectivity and continuity tests are conducted.

The server application developed in an Ubuntu 16.04 Server using Laravel Framework and is hosted via Cloud9. It is designed using the following use-case scenarios: user is unique, email is used for registration, user can make reservations, phone number is required to make reservation, entry and exit access codes are generated upon successful reservation, access codes are sent via SMS & email.
User Registration & Login

The registration page implements a minimalist approach, where only crucial details like name, email, password and acceptance of the T&C are collected to complete the registration. Upon successful registration, the user is automatically logged in and can proceed to use the system. If the user has already registered, can directly proceed in logging in with ease. Authentication is done to ensure the user’s credentials and access to the system as shown in Fig. 2.

![Fig. 2: User login authentication failed](image)

Parking Space Reservation

In order to make a successful reservation, the user needs to enter their mobile phone number, check in and out times, and the parking location zone, as shown in Fig. 3. Failure to do so, results in errors, that the user is prompted to fix.

![Fig. 3: Reservation of parking lot](image)

Access Control

The access control system is comprised of an LCD, 4x4 keypad and servo motor that is attached to a microcontroller that captures input from the user and forwards to the network module that performs verification of the access code from the server.

There are two sets of gates, i.e. main entry and exit, and, reservation entry and exit gates as shown in Fig. 4. The main entry and exit gates allow the user to park in the normal parking lots, and are required to proceed to the reservation parking lots in order to enjoy the reservation.

![Fig. 4: Access gate configuration](image)

In the current setup, the system is always active, thus no need to be activated when a vehicle is present. The system will automatically reset if it is idle for ten seconds after detecting input, and after either a successful or failed attempt to access the gate.
The system is configured to capture a password length of eight characters. When less characters are detected, it automatically issues an error, and dismisses the rest if more than eight. The number of permutations for generating an eight character HEX password at random is as follows:

\[
\text{HEX characters} = C_{16}^8 = 16 \\
\text{Password Length} = L = 8 \\
\text{Permutations} = C_{16}^8 = 16^8 = 4,294,967,296
\]

Safety mechanisms are in place to prevent use of the same access code on gate sets, i.e. when an entry access code is used to enter the main entry gate, it can’t be used again for the main entry gates, but can be used in the reservation entry gates. However, once the entry access code is used at both main and reservation gates, the access code is purged and can’t be used again. Fig. 5 shows the function spec requirements used in the access control system.

Vehicle Detection at Parking Space

This subsystem is designed so as to detect when a vehicle is parked underneath the parking lot. It employs a two-step verification system, which requires two set of conditions to be satisfied. An ultrasonic and Hall Effect sensor are used to accurately detect the presence of a vehicle when their threshold have been met.

The Hall Effect sensor is placed underneath the lot, whilst the ultrasonic sensor is placed on the ceiling, perpendicularly above the parking space. The sensing distance reduces with increase in vehicle height as shown in Fig. 6. The ultrasonic sensor acquires the height of the vehicle as follows:

\[
r_{\text{vehicle}} = R_{\text{max}} - R_{\text{sensed}} \quad \text{(0.1)}
\]

When the ultrasonic sensor detects a wide object at a height above the threshold, and when the Hall effect sensor is triggered, the system records the input as a vehicle present, and proceeds to change the LED colour to RED.

| Colour   | Changeover   | Description                                                                 |
|----------|--------------|-------------------------------------------------------------------------------|
| Red      | Green, Blue  | Red indicates that the parking lot is occupied and not available              |
| Green    | Red          | Green indicates that the parking spot is available for parking by any user    |
| Blue     | Red          | Blue indicates that the parking spot is available for parking by special users, e.g. Valet, or disabled. |
| Yellow   | Yellow       | Yellow indicates that the parking spot is not available for parking and that it is under maintenance. |

Wireless Network Nodes

These nodes are critical to the project, as they provide a TCP/IP connection protocol to the servers. Each device has a unique MAC address which is transmitted along with the data. These devices, called nodes, are preregistered by the admin and are tied to specific parking zones.

There are two types of network nodes, i.e. the parking and gate access node. The parking node handles the communication of parking sensors and
LEDs to the servers. The gate access node captures the access code, transmits it to the servers, and waits for a response, which is then sent back to the individual gate controller to allow or deny access. The current setup is designed for optimum performance and efficiency, thus only two parking spots and gate access controllers are connected to their respective nodes. The designed software program is generic since the node can be reconfigured easily by changing a header identifier.

2.3 Working Principle

The system is composed of four sub-systems, i.e. the user interface application, web server, gate access, and parking space, which work together in order to achieve the objectives, as shown in Fig. 7.

In order to be a complete IoT system, all configurations for the system are made in the web server. The gate access and parking space systems request updated configurations from the server to initialize their functionality. The user is required to valid credentials in order to make a reservation. The parking space sensor system updates the parking availability to the web server, which is responsible to assign a parking space to a user.

If a parking space is available, entry and exit access codes are generated and sent via email and SMS to the registered user’s email and phone number respectively which are used to access the gates. Validation is done on the server end, and a response is sent to the gate access system to allow or deny access.

Fig. 8 shows the reservation process for the system. The parking location and zone, check in and out times, and users’ mobile number is required to complete the reservation. Once submitted, the system backend checks for available parking based on the provided parameters. Access codes are generated if a parking is found, and are sent to the users via SMS and email. If no parking is found, a response is sent to the user.

---

**Fig. 7: Flowchart of the protection system**

**Reservation**
Entry and exit gate access

A keypad is used to capture the access codes in the system. A timeout resets the system if user fails to enter the password at a specified time (7 seconds). Once the password is captured, it is transmitted to the gate node, which transmits it to the server for verification. The server responds back with information regarding accessing the gate. A response 1 means success, and the gates will be open, else they’ll remain closed. Fig. 9 shows the flowchart describing this process.

Fig. 10 shows the flowchart of vehicle detection and server update. An interval timer of three (3) seconds is used to check and update the parking spaces status. Readings from the ultrasonic and hall-effect sensor is collected to detect if a car is parked. If the vehicle height is above threshold, the LED will change to indicate a car parked. The sensor readings are then uploaded to the server for storage.

Vehicle detection at parking spot & Server update

Information request using API

Integrators and users with valid credentials can request parking lot details/information for their own use. A request is composed by the user, which includes the API token/key, information location, zone and time range. RESTful/POST request is used to relay the information to server. Fig. 11 shows the flowchart of the server, as it performs validation and decoding on the request, and processes it. If all details in the request is valid, the server responds back to the user with all requested information.
3. SIMULATION AND HARDWARE RESULTS

Fig. 12 shows the simulation circuit for the gate access system. It is composed of the ATmega328P, LCD, keypad, boom gate and serial interface. There are two I2C modules are interfaced with the LCD and keypad, so as to reduce the number of input pins.

Fig. 13 shows the parking sensor system. It is composed of a microcontroller, ultrasonic sensor, and hall-effect sensor. An ATmega328p was used instead of the NodeMCU in simulation, as the controller is not available in the current release of PROTEUS. A serial terminal is used to view the output of the microcontroller for easy troubleshooting.

Fig. 14 and Fig. 15 show the system hardware implementation. They both show the parking lot layouts with LEDs indications of the state of occupancy for the car park.
4. TESTING RESULTS

4.1 Gate Access system

Simulation results of the gate access system is shown in Fig. 16 yields a success rate total average of 94 percent. The keypad capture input feature scored the lowest success rate of 80 percent. After troubleshooting, it was found the simulation was not running in real time, as the computer was overloaded. The LCD simulation was able to show all outputs but failed on one occasion, thus achieved a 90 percent success rate. The remainder of the system like system reset after entering the access code, opening and closing of gates, and system timeout after incomplete input achieved a 100 percent success rate.

4.2 Vehicle Detection system

The simulation results of the vehicle detection at parking space was tested repeatedly for both the ultrasonic and hall-effect sensor and the data is visualized in Fig. 17. The average accuracy and success rate found for the system operation is 91.67 percent. When the vehicle height was 13 cm, the system was able to accurately detect the correct height 80 percent of the time, whilst the ultrasonic achieved 100 percent accuracy. This test is done to demonstrate the usability of the system to register a new unique user who can proceed to make a reservation.
4.3 User registration and login

This test is done to demonstrate the usability of the system to register a new unique user who can proceed to make a reservation. Table 2 shows that the system is capable of registering the user easily and accurately. Validation is done to ensure that the registered user is unique, through the supplied email. The registration process is easy and simple for anyone to use.

Table 2: User registration and login usability test results

| Setting                   | Expected Result                                                                 | Actual Result                                                |
|---------------------------|---------------------------------------------------------------------------------|---------------------------------------------------------------|
| New user with correct input | Successful registration, and user is automatically logged into the system       | PASSED. User successfully registered, and login to the system. |
| Registration with existing email | Fail to register. Error messages shown                                          | PASSED. User fails to register when email exits.              |

4.4 Parking Reservation

This test aims to determine the usability and performance of the reservation system. It tests issuance of parking spaces when available, generation of access codes, and, sending SMS and email to the user.

Table 3: Parking reservation usability test results

| Setting                                                                 | Expected Result                                                                 | Actual Result                                                                 |
|-------------------------------------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| User does not fill the form properly                                    | An error appears, outputting messages of validation results.                    | PASSED. When the user fails to fill in the forms correctly, an error message is displayed |
| All user details filled correctly, parking is available                 | Invoice displayed detailing parking space, entry and exit access codes, SMS and email sent to user | PASSED. When a parking space is available in the zone, the system automatically assigns the user one, generates access codes, and, sends them |
| All user details filled correctly, no parking is available              | Error message is displayed telling user that no parking is available            | PASSED. When no parking space is available in selected location zone, user is informed, and prompted to select another zone. |

4.5 User at access gate

The aim of this test is to check the functionality of the gate access system. It tests the time it takes from when the user enters the correct code to the time the gate opens. It also tests the systems usability and hardware consistency.

Fig. 18 and Fig. 19 shows the observed angles of the boom gate when they are opening and closing respectively. It can be clearly observed that the error rate is of sinewave, i.e. the boom gate have slippage in the gears and damping. On the first run, the opening angle was 92° and closing angle was 0°, however, on the angles fluctuate whilst opening and closing. This is mostly probably due to the weight of the boom gate and the opening/closing speeds, which causes an offset.
The R-squared mean, also known as the coefficient of determination, is computed to determine how close the data are to the fitted regression line. For opening and closing boom gates the R-squared are 0.0036 and 0.0236 respectively. The overall average angles for opening and closing is 92.65° and 0.95° respectively.

Fig. 20 shows the results of the performance tests. It is noticed that test run seven (7) took the longest time of 8.2 seconds to execute a full cycle of the test. The second slowest test run was eight (8) and nine (9), which both took 8 seconds to complete. T3 is the longest wait time, since it waits for an average of 3.12 seconds till the gate close. On average, the total time it took for the system to complete this test was 7.76 seconds. T1, T2, T3 and T4 took an average of 1.61, 1.55, 3.12, and 1.48 seconds respectively.

In summary, no errors were observed in the usability and performance tests. The system is capable of accepting input from the user via the keypad, process and transmit it to the server, receive a response and open the boom gates as designed.

5. Conclusion

Thus, an IOT based car parking management system with API is designed. The CPMS is composed of WSN that are used to detect and collect information of vehicles at parking spots back to a centralized server, hosted using cloud-based services. All vehicle and parking information can be easily viewed and acquired using a web app and API, respectively.

The system data/information allows users to view available parking lots, and proceed to reserve them. Users are required to register using their emails and can reserve using any internet-enabled device. The system automatically generates two access codes, for entry and exit, when a parking space is available, and then sends...
them to the user’s email and phone (SMS). The results show that the system is capable of operating at an accuracy of 100 percent.

Once the user reaches the car park zone, they’re required to enter the entry access code at the gate access terminal. The gate access terminal captures the input and transmits it to the server for validation and verification. Upon successful verification, the user can enter the parking lot. The same is repeated upon exit, thus increasing security and integrity of the system. It takes roughly less than 1.61 seconds for verification with the server and gates to open.

At the parking space level, the vehicle detection system, which is composed of an ultrasonic sensor and hall-effect sensor, accurately detects a height below 20 cm with 99.79 percent. It can 100 percent detect a vehicle with a ride height below 1 cm. This shows that the combination of these techniques greatly improves vehicle detection at a parking space.

References
[1] H. Al-Kharusi, Intelligent Car Parking Management System, Palmerston North New Zealand, Massey University 2014.
[2] H. Al-Kharusi and Al-Bahadly, Intelligent Parking Management System Based on Image Processing, Scientific Research 2014.
[3] P. Almeida, L. S. Oliveira, E. J. Silva, A. J. Britto, and A. Koerich, Parking Space Detection using Textural Descriptors, International Conference on Systems Man, and Cybernetic 2013.
[4] M. Chen, C. Hu, and T. Chang, The Research on Optimal Parking Space Choice Model in Parking Lots, 3rd International Conference on Computer Research and Development (ICCRD) 2011.
[5] L. L. Elena, Polycarpou, Smart parking solutions for urban areas, 2013.
[6] A. Israel, Intergated Mobility, Transport Redefined, ITS America 2016.
[7] L. Meng, D. Cheng, and Z. Weimin, The Research of Intelligent Parking System based on Internet of Things Technology, International Journal of Computer Applications 124(6) 2015.
[8] Muhammad Mahbubur Rashid, Abiodun Musa Aibinu, MD. Ataur Rahman, Automatic Parking Management System and Parking Fee Collection Based on Number Plate Recognition, International Journal of Machine Learning and Computing 2 2012.
[9] F. I. Shaikh, P. J. Jadhav, and S. P. Bandarkar, Smart Parking System Based on Embedded System and Sensor Network, International Journal of Computer Applications 140(12) p. 51 2016.
[10] R. Yusniata and M. N. Fariza, A Secure Parking Reservation System Using GSM Technology, International Journal of Computer and Communication Engineering 2(4) p. 518-520 2013.
[11] J. Anitha Y Thoyajakshi A Ramya V Sravanu Prashant Kumar, Intelligent Parking System using Android application, International Journal of Pure and Applies mathematics 114(7) p. 165-174 2017.