International Conference on Advances in Materials, Computing and Communication Technologies
In Association with International Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

IoT Based Smart Parking System

B. Bharathikalaiselvi[1], R. S. Sangeetha[1], G. Selva Pragathy[1], D. Karthik Prabhu[2]

[1]Department of Electrical and Electronics Engineering, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India

[2]Assistant Professor, Department of Electrical and Electronics Engineering, Ramco Institute of Technology, Rajapalayam, Tamil Nadu, India

ABSTRACT

In recent times, the concept of smart cities has gained popularity. It is because of the growth and usage of Internet of Things effectively which leads to the evolution of smart cities. Despite the technology which has been used in parking system, many vehicles pursue only a small number of parking spaces, which in turn leads to serious traffic congestion. Metropolitan cities noticed that their drivers had real problems to find a parking space easily especially during peak hours, the difficulty roots from not knowing where the parking spaces are available at the given time. The Smart Parking system consists of an IoT module that is used to monitor and signalize the state of availability of each single parking space. A mobile application is also provided for the end user to check the availability of parking space and book a parking slot accordingly.

Keywords: Internet of things, Smart parking system, booking slots.

I. INTRODUCTION

The concept of smart parking was developed in order to address the issue of parking space and management in megacities. Congestion of vehicles is unavoidable due to the growing number of vehicles on the road and the limited number of parking spaces. This congestion will result in driver rage as well as pollution of the environment. Things with identity communication devices were the forerunners of the Internet of Things (IoT) concept. The ideal of creating a Smart City is now becoming possible with the emergence of the Internet of Things. One of the key issues that smart cities relate to is car parking facilities and traffic management systems [10]. Remote devices connected to the Internet could be used to locate, control, and monitor the devices. Cloud computing's scalability and reliability enable developers to build and host applications on it. According to a recent poll conducted by the International Parking Institute [2], there has been an increase in the number of innovative parking system concepts. There are currently several parking systems [3] that claim to provide people with real-time information about available parking spaces. The cloud is an ideal partner for IoT because it serves as a platform for storing and accessing sensor data from remote locations. [1] The proposed smart parking system is implemented via a cloud-connected mobile application. The system

provides real-time information about the availability of parking spaces to the user.

II. IOT-CLOUD INTEGRATION

The advantages of IoT in the cloud by granting third-party access to the infrastructure, public cloud services can easily assist the IoT space. As a result, the integration will assist IoT data or computational components that are used on IoT devices. To share data for worthwhile purposes, IoT devices need a lot of storage [4]. To communicate and connect with one another, the vast quantities of data provided by IoT devices necessitate extreme performance. IoT in the cloud provides the connectivity required to share data between devices and make sense of it in a timely manner.

Storage capacity:

The Internet of Things (IoT) is made up of a vast number of data sources (things) that generate massive quantities of unstructured or semi-structured data. As a result, IoT necessitates large-scale data collection, access, processing, visualization, and sharing [5]. Cloud storage capacity is unrestricted, low-cost, and on-demand, making it the best and most cost-effective solution for dealing with IoT data. Through standard APIs, data stored in the Cloud can be accessed and visualized from anywhere.

Computation power:

The processing capabilities of the devices used in IoT are limited. The data collected from different sensors is typically sent to more powerful nodes for aggregation and processing. [6]. The IoT computing needs can be met by utilizing Cloud's unrestricted processing capabilities and on-demand model. IoT systems could use cloud computing to perform real-time data processing, allowing for more responsive applications.

Communication resources:

IoT's primary function is to allow IP-enabled devices to communicate with one another via a dedicated set of hardware. Cloud computing allows you to connect, track, and manage devices from anywhere on the internet [7] for a low cost. IoT systems could monitor and control things in real time from remote locations thanks to built-in applications.

Scalability:

The cloud offers a scalable approach to IoT. It allows for a dynamic rise or decrease in resources. When cloud integration is available, any number of "things" can be added or removed from the system. [8]. The cloud distributes resources based on the demands of stuff and applications.

The following figure gives an outlined view of the smart parking system.

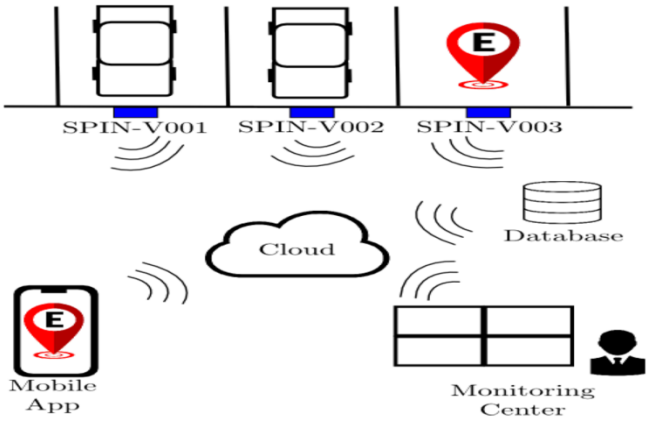


Fig 2.1 Outlined view of Smart parking system

Parking Sensors:

We used sensors such as infrared or ultrasonic for our parking system. These sensors are responsible for sensing the parking area and determining whether or not a parking space is available. We're detecting the presence of a vehicle with infrared sensors in this case. The ESP8266 chip is wired up to the infrared sensors. A self-contained SOC with integrated TCP/IP protocol stack, the ESP8266 Wi-Fi chip, allows any microcontroller to connect to a Wi-Fi network. A 5V supply from an external source is used to power the sensors.

Processing Unit:

The processing unit serves as a link between the sensors and the cloud. The processing unit is connected to all of the sensors. The esp8266 chip receives data from a variety of sensors. The data is then sent over a channel to the IBM MQTT Server using the MQTT protocol [9]. The MQTT (Message Queue Telemetry Transport) Protocol is a "lightweight" publish-subscribe message protocol built on top of the TCP/IP protocol. It's intended for establishing connections between remote locations where only a small amount of data needs to be transferred or where bandwidth is limited.

Mobile application:

End users connect with the system via the mobile application, which serves as an interface. The application is written in JavaScript and constructed with Apache Cordova and the Angular Js framework. The goal of Apache Cordova is to make apps that can run on both Android and iOS using the same source code. The application uses a secure channel and two-factor authentication to communicate with the IBM MQTT server. The intent of this mobile application is to provide information about parking space availability and to enable the end user to book a slot based on that information. Between the IBM MQTT server and the mobile application, data is transferred in JSON format.

The Cloud:

The IBM MQTT server is cloud-based. The cloud serves as a data store for all records pertaining to parking areas and end users with access to the system. It keeps track of every user connected to the system and records data such as the time the car was parked, the length of time it took to park the car, the amount paid by the user, and the method of payment. It's because of the cloud's adaptability, which allows the system to add any number of users at any time of day. Continuous backups of data stored on the cloud are made to ensure convenient and fast data recovery in the event of a system failure.

III. METHODOLOGY USED FOR WEB APPLICATION

Ionic is an HTML5 Software Development Kit that allows us to build our own smartphone apps using web technologies such as HTML, CSS, and JavaScript. It is primarily concerned with the application's appearance and user interface interaction. The biggest benefit of using this Software Development Kit is that we can run the program as a web application or as an Android application with only one line of code.

ANGULAR:

Angular is an HTML and TypeScript-based interface and architecture for developing single-page client applications. TypeScript is used to construct Angular. It's a compilation of TypeScript libraries that you import into your applications to incorporate key and optional features.

An Angular application's architecture is based on a few basic principles. The Angular framework's fundamental building blocks are Angular elements, which are grouped into NgModules. NgModules are functional collections of similar code; an Angular app is represented by a series of NgModules.

CLOUD STORAGE:

The user's account is maintained in the cloud. We use Google Firebase for this, which is a Google-backed mobile development platform that allows developers to create iOS, Android, and Web apps. Firebase offers analytics monitoring, analysis, and software crash fixes, as well as marketing and product experimentation resources.

IV. IMPLEMENTATION & WORKING

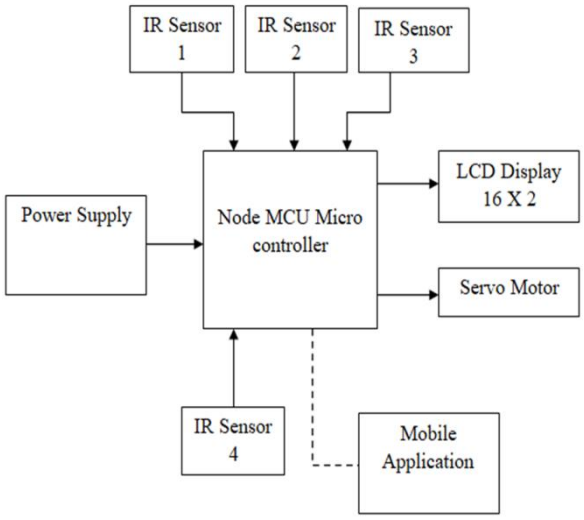


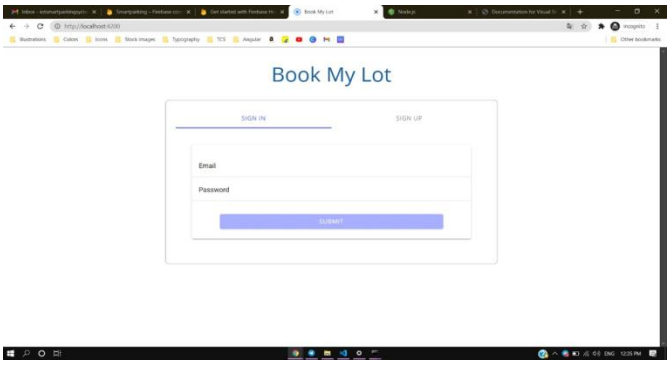
Fig 4.1 Block diagram

This project has two sections. The first is the key device, which will be placed in every parking space in the parking lot. The machine will use sensors to identify the presence of a car in the slot. The device would also employ a monitor and a keypad for user input, such as entering information about the required time. The car is detected with the aid of infrared sensors. Three IR sensors are used for identification, and one IR sensor is used to detect vehicle entry and departure, the output of which is fed into the servo motor, which controls the parking slot gate operation. The whole parking area is separated into several slots, with an infrared sensor located in each slot. All of the sensors in that specific area parking slot are attached to a single microcontroller (Node MCU ESP8266), which is supplied with a 5V, 1A supply via an adaptor. When the car drives into a parking spot, the driver scans the accompanying smartphone app and LCD monitor for open parking spaces. The sensors in and parking space will hold a cloud inventory of open spaces up to date at all times. If the vehicle is present, the sensors can detect it and send a signal to the microcontroller's analogue pin. The software in the microcontroller senses whether or not vehicles are present based on the response.

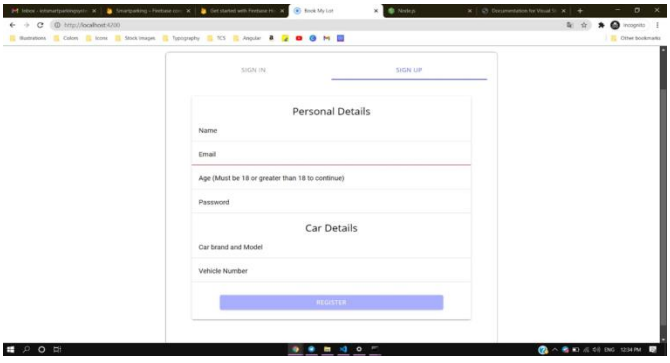
This data is updated on the cloud servers by the microcontroller. The mobile application is updated with this information, and it then recommends a parking slot based on different factors such as proximity and ease of access, or simply an optimal solution.

The driver needs to follow these steps in order to park its car using our parking system.

Step 1: Install the smart parking application on your mobile device.

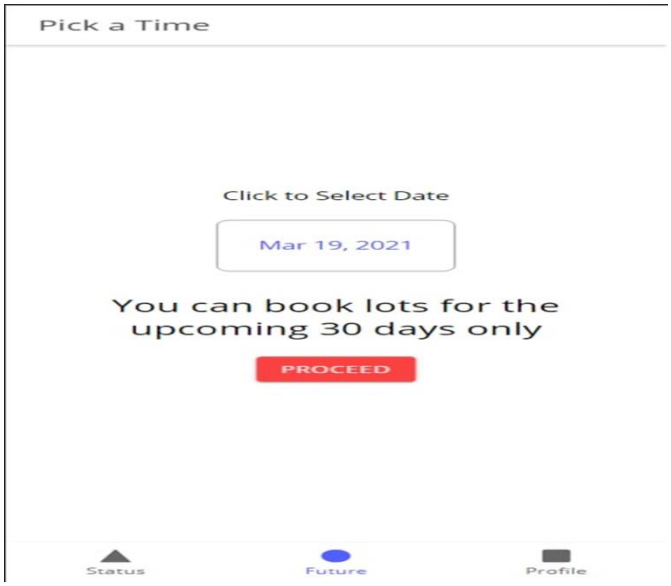


Step 2: Sign up the application for booking the slot.



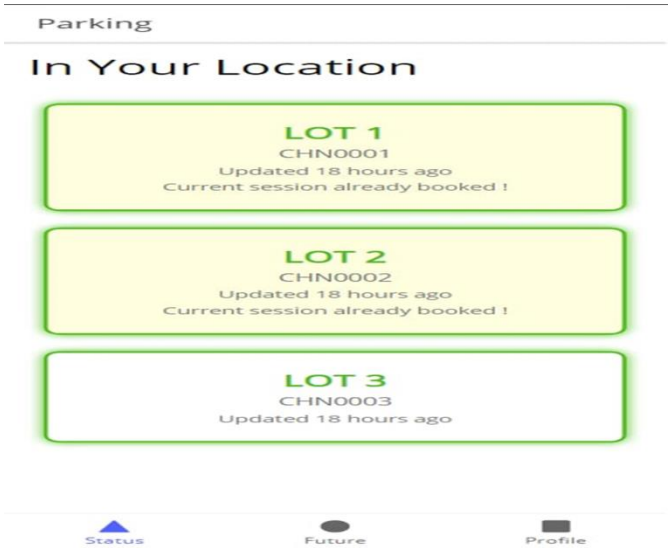
Step 3: The user can see their profile details in the application after signing

Step 4: With the help of the application the user can choose the duration and date for booking the lot.



Step 5: After choosing the date the user can choose the time slot for booking.

Step 6: Select a particular parking lot in your location.



Step 7: Pay the parking charges either with your e-wallet or your credit card

Step 8: Once you have successfully parked your car in the selected parking slot, confirm your occupancy using the mobile application.

V. CONCLUSION

Smart Cities have always been a pipe dream for Mankind. In the last few years, significant progress has been made toward making smart cities a reality. The development of Internet of Things (IoT) and Cloud technologies has opened up new options for smart cities. The construction of smart cities has always included smart parking facilities and traffic management systems. We address the problem of parking in this paper and present an IoT-based Cloud-integrated smart parking system. The system we propose gives real-time information on parking slot availability in a parking lot. Our mobile application allowed users in remote areas to book a parking spot for themselves. The attempts made in this paper are aimed at improving a city's parking facilities and, as a result, improving the quality of life of its residents.

VI. REFERENCES

[1]. Fox, G. C., Kamburugamuve, S., & Hart R.D. (2012, May). Architecture and measured characteristics of a cloud based internet of things. In Collaboration Technologies and Systems (CTS), International Conference on (pp. 6-12). IEEE.

[2]. International Parking Institute, “2012 Emerging Trends in Parking”.

[3]. FastPark System website, http://www.fastprkm.

[4]. Botta, A., de Donato,W., Persico, V., & Pescape, A. (2014, August).On the Integration of Cloud Computing and Internet of Things. In Future Internet of Things and Cloud (FiCl oud), 2014 International Conference on (pp. 23-30). IEEE.

[5]. Rao, B. B. P., Saluia, P., Sharma, N., Mitta A.,& Sharma, S. V. (2012, December). Cloud computing for Internet of Things & sensing based applications. In Sensing Technology (I CST),2012 Sixth International Conference on (pp. 374-380). IEEE.

[6]. Zaslavsky, A., Perera, C., & Georgakopoulo D.(2013). Sensing as a service and big data. arXiv preprint arXiv:1301.0159.

[7]. Suciu, G., Vulpe, A., Halunga, S., Fratu, O. To doran, G., & Suciu, V. (2013, May). Smart cities built on resilient cloud computin g and secure internet of things. In Control Systems and Computer Science (CSCS),2013 19th International Conference on (pp. 513-518). IEEE.

[8]. Sarkar, C., Uttama Nambi SN, A., Prasad, R.,Rahim, A., Neisse, R., & Baldini,G. (2012 DIAT:A Scalable Distributed Architecture for IoT.

[9]. Wikipedia website, https://en.wikipedia.org/wiMQTT.

[10]. Zhou, F., & Li, Q. (2014, November). Parking Guidance System Based on ZigBee and Geomagnetic Sensor Technology. In Di stributed Computing and Applications to Bu siness, Engineering and Science (DCABES), 2014 13th International Symposium on (pp. 268-271) IEEE.