D24310484C19

Preprint · October 2019

DOI: 10.13140/RG.2.2.26946.09929

1 author:

Hitesh Kumar Sharma
University of Petroleum & Energy Studies
68 PUBLICATIONS **192** CITATIONS

SEE PROFILE

Automated Parking System-Cloud and IoT based Technique

Rajeev Tiwari, Hitesh Kumar Sharma, Shuchi Upadhyay, Shreyansh Sachan, Ayush Sharma

Abstract*: With the emergence of Internet of Things (IoT) for smart communication and intelligent devices, smart cities have got a new direction for establishment. Smart infrastructure is mandatory for smart cities. One of such infrastructures is parking areas and their slots. So, in this paper a novel algorithm is designed and implemented for smart parking of vehicles at smart parking areas. Vehicles are enabled on network using RFID tags and parking areas are acting smart using Arduino and access to cloud server. This smart architecture has increased efficiency od smart parking system (SPS) in terms of cost to reach for parking and efficient usage of parking slots with least time consumed. Mobile applications are designed for users and for parking owner. Central data collection is done on cloud-based data centre, which facilitates users and parking owners to connect from any geographical region and provides scalability and flexibility. We have implemented this work in real life scenarios in Dehradun, Uttarakhand with 50 vehicles and 5 parking areas with 10 parking slots each. This IoT bases SPS has got efficiency in terms of least cost parking, successful parking and efficient usage of parking slots.

Index Terms: IoT RFID Smart Parking System Cloud Android App

I. INTRODUCTION

Parking search is a routine activity for many people in cities around the world. The resulting congestion has become a major concern to search for the parking space. This search waste about one million barrels of the world's oil every day. As the global population continues to urbanize, without a well-planned, convenience-driven retreat from the car these problems will worsen.

The common method of finding a parking space is manual where the driver usually finds a space in the street through luck and experience. This process is time consuming and may also lead to the worst case of failing to find any parking space in a city with high vehicle density.

According to a report, Smart Parking could result in 2, 20,000 gallons of fuels saving till 2030 and approx. 3, 00,000 gallons of fuels saved by 2050, if implemented successfully. Smart cities are no longer the wave of the future. They are here now and growing quickly as the Internet of Things (IoT) expands and impacts municipal services around the globe [1].

With continuous increase in population and urbanisation, Vehicular traffic is continuously increasing around the world, especially in large urban areas. Currently, 31% of India's population lives in cities; these cities also generate 63% of the nation's economic activity. These numbers are rapidly increasing, with almost half of India's population projected to live in its cities by 2030. Impact of this can be seen in vehicular traffic, which is continuously increasing, especially in large urban areas.

In large populated countries like India, the resulting congestion has become a major concern to transportation specialists and decision makers. The existing methods for traffic management, surveillance and control are not adequately efficient in terms of performance, cost, maintenance, and support. IoT has already been introduced to solve the parking problem [2,5]. It typically obtains information about available parking spaces in a particular geographic area and the process is real-time to place vehicles at available positions. When deployed as a system, smart parking thus reduces car emissions in urban centres by reducing the need for people to needlessly circle city blocks searching for parking. However, all these parking systems have not been sufficient enough to ease all the needs of the growing population and traffic.

In this work, smart parking solution is made smarter to next step with the introduction of RFID sensors in IoT. This research provides a mechanism that increases the efficiency and develops a cloud-based network architecture based on the Internet of Things. This work aims to reduce the time and cost of the driver by showing all the available parking space near to him, he can select the suitable parking space and reserve it. The users are provided with the facility to reserve the parking spot. The user will be charged on the monthly basis for the number of hours parking spaces were used and can pay their bill at the end of the month for all the parking spaces used. This project proposes an architecture that introduces the use of active radio-frequency identification (RFID) wireless sensor technologies, object ad-hoc networking and Internet-based information systems in which tagged object can be automatically represented, tracked, and queried over a network.

II. LITERATURE REVIEW

There are various existing systems available for parking management which try to reduce the manual effort, such as Pham et al. [4] who proposed a system that enables users to locate a free parking space at lowest cost by considering the distance to the parking space and number of available parking spots.

Revised Manuscript Received on December 22, 2018.

Rajeev Tiwari, Dept. of Virtualization, SCS/ UPES, Dehradun, India.
Hitesh Kumar Sharma, Dept. of Cybernetics, SCS/ UPES, Dehradun, India.
Shuchi Upadhyay, Dept. of Food Tech., UCALS, UU, Dehradun, India.
Shreyansh Sachan, Software Developer, IBM, Noida, India.
Ayush Sharma, Software Developer, Xebia, Noida, India.

This is used to provide an available parking space to the user and also provide a suggestion about a new space if the allocated space gets occupied before he reaches. The results show that it minimizes the user waiting time, however, there is a worst case for the user to reach a starvation, if the parking space shown gets filled before he reaches.

In "A New Smart Parking", the authors proposed a new algorithm to plan in real-time parking. The algorithm schedules the online problem of a parking system into an offline problem [6, 8]. Offline problem was solved as a linear problem by using the proposed algorithm. Finally, proposed algorithm was evaluated by using trial prototypes of the solution. The trial results showed effective performance. However, these papers fail to provide the reservation mechanism.

In another study "Integration of RFID and WSN technologies in a smart parking system", the authors propose a solution implementing the amalgamation of UHF frequency, RFID and IEEE 802.15.4 Wireless Sensor Network technologies [9]. This system collects the information about occupied spaces and provides a parking spot to the user.

Bonde et al. [10] provided another method to automate the car parking mechanism. This method presented a model based on the number of cars that can be parked in a given area at any given time based on the occupancy of parking area. Entry or exit from the parking is monitored by the sensing devices and controlled by a mobile application. However, the system does not have any type of reservation of resources which may lead to more time and cost in case all the parking spots are occupied [10]. Another useful approach is proposed in literature [11].

Other researchers have proposed intelligent parking assistant (IPA) architecture [12, 13]. Architecture was put forward for overcoming current public parking management hindrances. The architecture provides drivers with information about parking status and a reservation policy along with RFID. The car's entry or exit from the parking spot is monitored by the magnetic loop and is sent to the central controller for updating the car park status. However, it costs more due the magnetic loop detector and this system fails in large parking system.

III. PROBLEM STATEMENT

To ease the parking problem in large dense populated cites by centralizing all the parking areas of the city so to facilitate easy search and automate the parking procedure by removing human interference, also to impose fine on wrong parked vehicles, calculate total bill and payment through payment portal.

IV. OUR CONTRIBUTIONS

For designing a solution for smart parking system for a city, we got inspired by works of researchers in [5, 9]. Few additions we did for making our system better in efficiency, as given below:

1. A new parking algorithm is designed in this proposed system, based on IoT network, mobile application, using Arduino, and RFID tags.
2. IoT network is established over the city to connect and communicate to all parking of the city and users. Drivers are provided with android mobile application to search for the parking area nearby and reserve it. For the detecting the vehicle RFID technology is used.
3. RFID tags with vehicle details are attached to the vehicle, and Arduino with RFID reader is placed at each parking area. Data centre have details of the users and parking areas, and act as the central medium for all overall communication. Data centre serves for the users search request, parking spot allocation, parking availability.
4. Algorithm implemented on data centre servers adopts a mechanism to search car parks at the least cost.
5. Another mobile application is developed for the parking owners to view the current status of their parking areas and payment received by parking owner.
6. The proposed system reduces the number of vehicles failing to find a parking space and thus helping users save time and money.

V. PROPOSED WORK

In this proposed research work, a parking system based on the IOT and android application is designed to save the drivers time and fuel in searching a free parking space. There will be an Arduino and RFID reader on every parking spot and an RFID card on every vehicle which holds the vehicle number. There are two types of clients in this system, first is the driver who want to park its vehicle and the second is the parking space owner who wants to provide his parking space to earn money. The driver first has to register on android application by providing necessary information like vehicle number, after successful registration the user will able to see its current location on the map.

The map shows the driver current location along with the free parking space on different locations. The driver can reserve the parking spot by selecting a particular free parking spot and then has to reach the parking spot in 15 minutes from the time of reservation. When the reservation is done the application automatically starts the navigation and shows the route to parking spot on map. If the driver fails to reach the parking spot in 15 minutes, then the driver has the option to extend time to another 15 minutes; if fails to reach again then the reservation will be cancelled, and user has to again reserve the parking spot.

When the driver reaches parking spot, the RFID reader on parking spot reads the RFID card on vehicle which contains the vehicle number and Arduino sends this information to the server along with arrival time. When the vehicle leaves parking spot Arduino again sends the departure time and the amount is calculated based on total time of parking. The driver has to pay the bill within a month else has to pay a fine. If the driver does not have the android application or the application is not working, then the driver can park on the parking spot with the help of parking owner. The parking owner can also book the parking through the application only on its parking space and can also see the parking details on a date.

VI. DETAILED DESCRIPTION

A.System Architecture

The proposed system architecture contains, Smart phone users, they have installed SPS application in their smart phones. They remain connected to a central public cloud which has got our database(DB) on DB server. DB server has all the data related to users, parking areas and parking slots in them. Such an architecture is shown in Figure 1. Its different components used in it are as given below:

- **Users:** Users are driver of the vehicles who need parking. They have to register for application download. They use mobile application for booking and payments, detailed process is discussed in coming sections.
  Central Cloud: Cloud based infrastructure is used for get-ting access of data of parking areas and availability from any geographical location.
- **Cloud Data Server:** Central databases server or center is used on cloud. Its details are covered in coming sections.
- **IOT Enabled Parking Areas:** Smart parking areas are designed which are IoT enabled. They have a setup of Arduino, with RFID readers and cloud connectivity on server. More details can be found in coming sections.

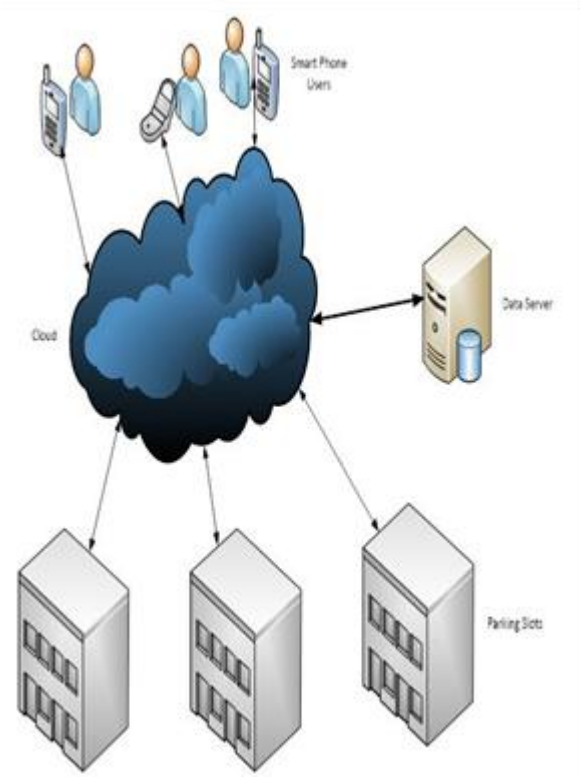

Figure 1: Proposed System Architecture for Smart Parking System

B.IoT Network

In a parking area, an IoT network is enabled to automate the parking process using cloud, wi-fi module and RFID reader. The complete network diagram and its components are shown in Figure 2.

1. **Tags**: Tags Used Vehicle number is to be written in RFID tags and attached in the vehicle. This is done to identify the vehicle details.
2. **Arduino:** Arduino RFID reader along with internet connectivity must be setup for at each parking spot to read the vehicle detail and send it to the cloud datacenter. RFID Module is connected to Arduino with the help of jumper cable as shown in Figure 3.
3. **Internet Connectivity:** Internet connectivity ca be provided to the Arduino by any of the following two ways, either by Ethernet module or Wi-Fi module ESP 8266.

For providing internet connectivity using Ethernet, Ethernet is placed just on the Arduino as shown in the figure 4 and wi-fi module is connected as shown in figure 5. Full setup of Arduino along with RFID reader and wi-fi ESP 8266 setup looks like something as shown in figure 6.

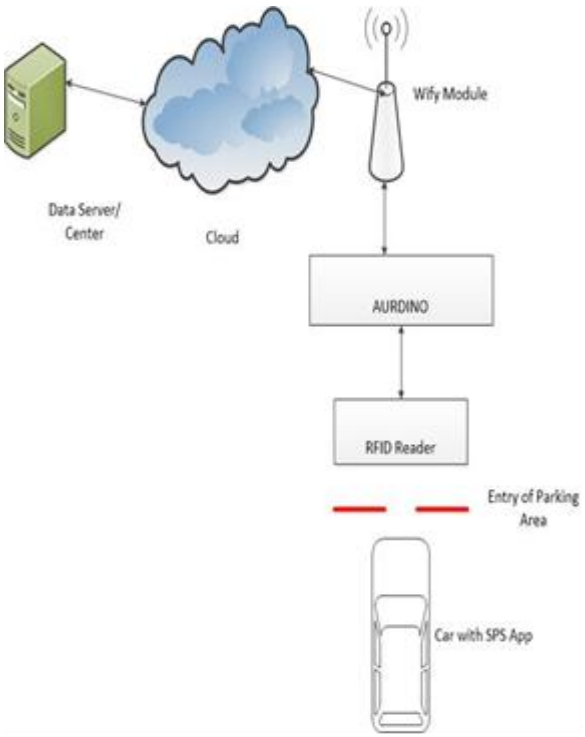

Figure 2: IoT network of a Parking Area

C. Cloud datacenter

Database This work uses **MySQL** database for querying the database.

Database Name: parking

Table Structure:

Table-1: User Details
Attributes: • vehicle number: driver vehicle number (primary key) • name: name of the driver • phone number: phone number of the driver • email: email address of the driver • password: password of the driver • is verified: used for verification of driver.

Table-2: Parking Area
Attributes: • parking id: id of the parking area (primary key)

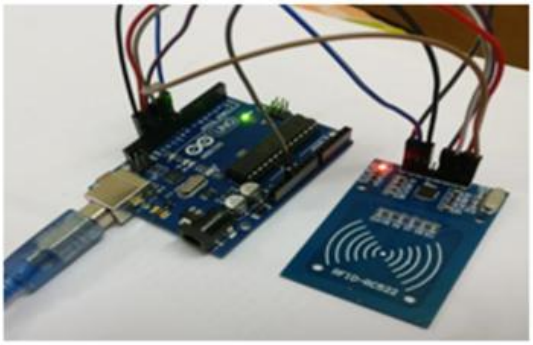

Figure 3: Arduino and RFID connected at Parking Area

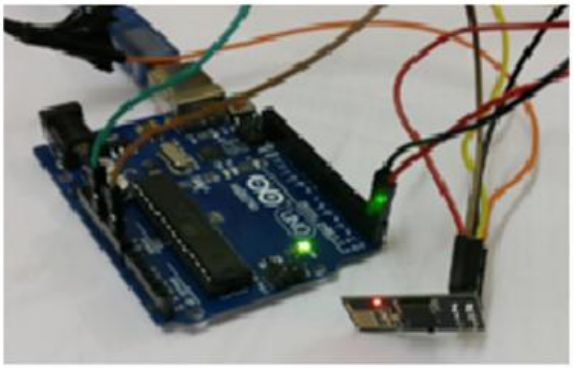

Figure 5: Arduino connected with ESP 8266 wi-fi Module at Parking Area

• parking name: name of the parking area • address: address of the parking area • latitude: latitude of the parking area • longitude: longitude of the parking area • total parking spot: total number of parking spots in parking area • available parking spot: free parking spots

Table-3: Parking Spot
Attributes: • parking id: id of the parking area (foreign key) • parking spot id: id of the parking spot • status: status of parking spot (vacant/occupied/reserved)

Table-4: Owner Details
Attributes: • owner id: id of the parking owner (primary key) • name: name of the owner • parking id: id of the parking area (foreign key) • password: password of the driver

Table-5: Booking Details
Attributes: • vehicle number : driver vehicle number (foreign key) • parking id : id of the parking area (foreign key) • parking spot id : id of the parking spot • booking date : date of booking of parking • entry time : time when vehicle enters • exit time : time when vehicle leaves • amount : total parking amount • amount type : type of amount(normal/fine) • amount status : status of amount (paid/unpaid)

Such Data bases tables are created, which carries entries related to vehicles, their start time, end time along with details of their parking status, which can be fetched from cloud server by different mobile application users

Figure 4: Ethernet Shield attached to Aurdino at Parking Area

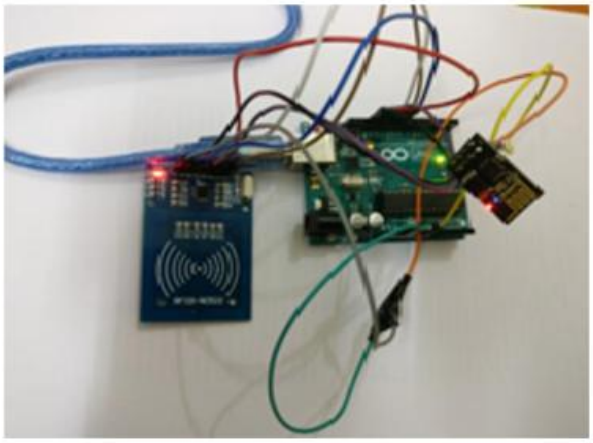

Figure 6: Arduino and all components connected at Parking Area

D. Computational Server

This research work uses XAMPP server to handle all the PHP scripts which make request to the PHP server. The entire request related to free parking space, reserving parking space and calculating the bill are handled by the XAMPP server.

It first generates the OTP and verify at the time of registration of driver. It sends the latitude and longitude of free parking spots to the android application based on the driver current location and reserves the selected parking spot. It maintains the driver, parking area and owner information in the MySQL database and performs the various operations like update, insert, delete on the database.

E. *Working Algorithms*

Symbols Used:
VI = Vehicle,Pj = Parking spot,RFIDPj = RFID tag on parking spot Pj,RN = Vehicle Number, Ta = Arrival Time, Td = Departure Time, Tt = Total Time, lat = latitude of vehicle, long = longitude of vehicle, pid = parking spot ID
Amount = amount to be paid by the user Rate = parking rate

Assumptions:
(i) Distance (d<=5km)
(ii) One user can reserve only one parking at a time

Parking Side Algorithm

Step 1: Vehicle Vi passes over RFIDPj reader in parking spot Pj and fetches data as < VI, RFIDPj, Pj >.
Step2: RFIDPj reads <RN> on < Vi > and sends < RN, Pj > to the Server.
Step 3: Vehicle < VI > leaves parking spot < Pj >; it passes over < RFIDPj > .

Arduino and RFID reader setup established at the parking area reads the vehicle number from the vehicle passing over the reader and send the information such as parking area id, parking spot id and vehicle number to the cloud data center over the internet. Information is send each time the vehicle passes over the information, while entering as well as at the time exit. First reading send is for the arrival of the vehicle and the second time denotes that the vehicle has exit the parking.

Server-Side Algorithm

Input: Database (DB) having fields < RN, pid, lat, long, amount >

Step 1: Server receives < lat, long > of the vehicle < VI >
Step 2: Server searches empty parking spaces based on < lat, long > and return < Pj, pid > using:

```
while (d<=5) do
{
lat1 = lat - (0.0090090 * d);// 0.0090090 = 1 km lat2 = lat
+ (0.0090090 * d);
long1 = long - (0.0090090 * d); long2
= long + (0.0090090 * d);
sql = retrieve free parking spaces within <lat1, lat2,
  long1, long2 >
if (sql)
  return lat, long
else
  d++;
return -1, -1
}
```

Step 3: Vehicle < Vi > reaches < Pj >; DB is updated with entries as < Pid, Pj, RN >
Step 4: Vehicle Vi leaves Pj, DB is updated with entries as < Td, Ta and Tt, amount > and amount is calculated as:
amount = Tt (hours) * rate // amount (in rupees)

When the user searches for the parking space through android application, request with user's lat-long obtained from the mobile GPS is received at the server.

Server looks for the vacant parking area as in step 2 from the server with the square region of the 1 km2, if found, parking areas details are send to the user, if not the region of search is increased with the length of 1km (for 5 times) till the vacant parking areas are obtained. Server also stores the information received by the Arduino setup along with the time of the request in the database. Second time when the request is received from Arduino setup, ie the departure of vehicle from parking spot, it is stored and the amount for the time parking used is calculated and stored is also stored based on the parking rate.

F.Flowchart

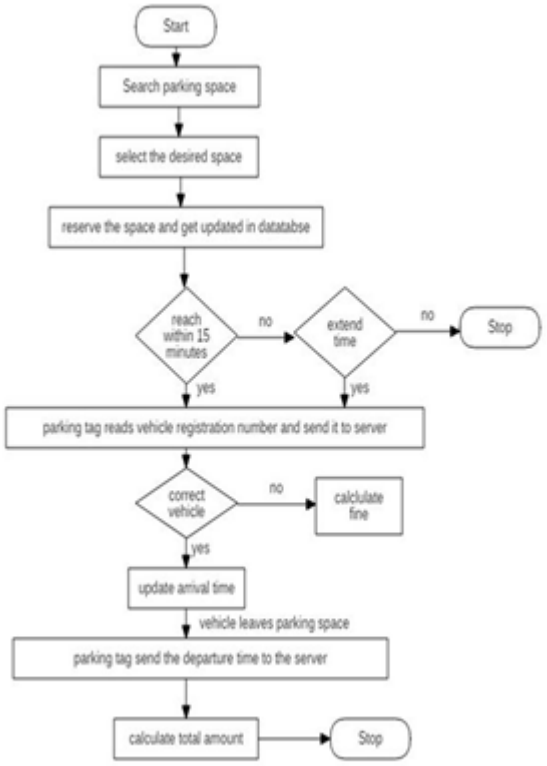

Figure 7: Flowchart of allocation of Parking slot in Smart Parking System

Working flowchart is shown in Figure 7. User simply use the mobile application to find the parking areas nearby, parking areas are shown to the user the user select the best suited and reserve it. User then navigate to the reserved parking, since the reservation is made for only 15 min, so user must reach within time else the reservation would be cancelled. If user feels that he won't be able to reach within time he can extend the reservation time once, through the app. As the user reach the reserved parking spot, Arduino setup established there read the vehicle number from the vehicle tag and send the parking spot and vehicle details to the cloud server. Cloud server decide whether the vehicle being parked is correct or not based on the booking table and details. If vehicle parked is correct its details like vehicle number time of arrival, parking details etc. are stored in the database, if the vehicle being parked is wrong or comes after reservation time is over, it will be treated as wrong parked and will be fined.

When the vehicle moves away from the parking spot its details are again send to the server for the departure time. Here the server computes the amount based on amount of time vehicle parked and rate of the parking.

G. Mobile App and Its Working

Figure 8: User app sign in

Figure 9: Searching of Smart Parking through APP

Figure 10: User gets available parking on APP

Figure 11: User books a parking

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

User Application Mobile application is developed for the user, which has following features for access:

- User can search for available parking areas.
- User can reserve the parking spot.
- View parking history.
- Pay for the parking space used.

As shown below in Figure 8, interface of mobile app for user login for a registered user. Then user can look for parking areas available on SPS, as shown in Figure 9. Depending on present location of user, SPS system will show available parking areas near to user within range of 5 KMs, as shown in Figure 10. User selects an available parking, Bidholi 2 as shown in Figure 11 and books the desirable parking area. Now user has to reach to parking area for physical parking area slot allotment. So SPS application starts navigation for user from current position to the pre-booked parking area, as shown in Figure 12. When user reaches to parking area, his vehicle's RFID tag is being read by RFID receiver mounted on Arduino at entry gate of parking area. Looking at database entry, entry of vehicle is authenticated and allowed for parking in parking area available slot as shown in Figure 13. Using wi-fi module with Arduino, updated count of parking slot is sent to databases server. This way count on parking slots are being maintained on Data Base server.

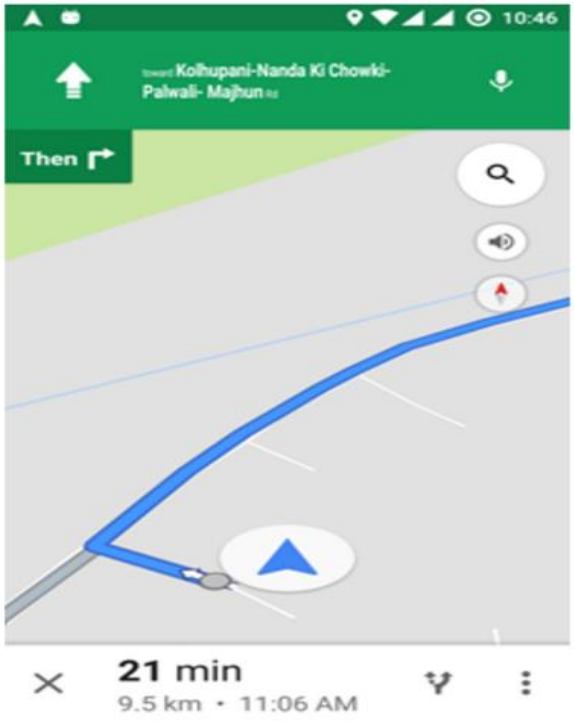

Figure 12: User navigates with App towards booked parking

H. PARKING OWNER APPLICATION

Another application is developed for parking area owner through which they can track the status of the current area off the parking area as shown in figure and they also have the right to make any of the parking spot offline and do offline booking. Parking owner can also view all details of the parking made as shown in figure 14 and figure 15.

Developed application for the parking owners has following features in it.

- To view current status of the parking spot.
- To make booking offline.
- To view history of parking.

Registered parking owner can login on application and can see its parking occupancy details on online interface. Parking area slots occupancy status can be seen, and payments received or pending can also be tracks from the application itself.

I. Working Scenarios

Scenario-1

Client books the parking spot and reaches the Parking Spot in time.

Step 1: User opens the android application and searches for the parking spot.
Step 2: User select the suitable parking area and books it.
Step 3: User reaches the allotted parking spot within the allotted time. When the user will reach the parking spot RFID Reader will scan the Vehicle Number from the RFID tag, and sends the vehicle number, parking spot id to server. Server will update the entries.
Step 4: When user move out of the parking spot, RFID Reader will again scan the tag and send the details to server for updating.
Step 5: Server will compute the amount based on the duration for which the parking spot was used and update the amount in the server and also sends the notification to the client.

Figure 13: Reached for parking at booked area

Scenario-2

Client book the parking spot but fails to reach on time.

Step 1: Client will book the parking spot as in previous step.

Step 2: If the client fails in reaching the allotted parking spot within time, its reservation will be cancelled, client will be informed about cancelation and also the fine will be imposed

Scenario-3

Client doesn't have android application or vehicle does not have required RFID Tag.

Step 1: Client will reach any parking area, tell the owner about the problem.

Step 2: Parking Owner will assign any of the available parking spot to the client.

Step 3: Owner will also notify the server through mobile application as shown in figure that he has booked the spot offline and availability of that spot to be updated as occupied in the server.

Step 4: When the client leaves he will pay to the owner according to bill made.

Step 5: Owner of the parking area will update the parking spot to be available again through mobile application

VII. Conclusion and Future Directions

Table 1: Results determined as per real life scenario implementations

Schemes	Cost of Parking	Parking Fulfillment	Online/ Offline Support
Y.Geng [7]	Less	Medium	Online
X.Zhao [8]	Less	Medium	Online
T.N. Pham [9]	Lesser	High	Online
Proposed	Least	High	Both

This work has proposed a new algorithm for smart parking system. It improves the performance of parking occupancies by offering a 15-minute advanced booking and taking nearby parking slots only within range of 5-kms. Due to which vehicles can have least cost travel to reach to available parking area. Our work is implemented in real life situation with 50 vehicles and 5 parking areas in Dehradun. Our algorithm significantly reduces the parking cost and fulfillment of parking lots of parking areas. Our system achieved the solution where most of the vehicles successfully found a free parking space as shown in Table 1. It also supports online and offline support for parking slot reservations. In our future study, we will consider the security aspects and will try to scale up our implemented system in real life scenarios.

Acknowledgement

This work is seed funded by University of Petroleum and Energy Studies, Dehradun.

REFERENCES

1. Basu, Aditya, Happiest Mind Parking Pvt. Ltd., "Smart Parking", Bangalore India
2. M. Du, J. Fang, and H. Cao, "A new solution for city parking guiding based on Internet of Things and multi-level multi-agent," in Proc. Int. Conf. Electron., Commun. Control (ICECC), 2011, pp. 4093-4096.
3. C. Rhodes, W. Blewitt, C. Sharp, G. Ushaw, and G. Morgan, "Smart routing: A novel application of collaborative path-finding to smart parking systems," in Proc. IEEE 16th Conf. Bus. Infom., Jul. 2014, pp. 119-126.
4. I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," Comput. Netw., vol. 47, no. 4, pp. 445-487, Mar. 2005.
5. N. Mejri, M. Ayari, R. Langar, F. Kamoun, G. Pujolle, and L. Saidane, "Cooperation versus competition towards an efficient parking assign-ment solution," in Proc. IEEE Int. Conf. Commun., Sydney, NSW, Australia, Jun. 2014, pp. 2915-2920.
6. Y. Geng and C. G. Cassandras, "A new 'smart parking' system based on optimal resource allocation and reservations," in Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC), Oct. 2011, pp. 979-984.
7. Y. Geng and C. G. Cassandras, "New smart parking system based on resource allocation and reservations", IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1129-1139, Sep. 2013.
8. X. Zhao, K. Zhao, and F. Hai, "An algorithm of parking planning for smart parking system", in Proc. 11th World Congr. Intell. Control Autom. (WCICA), 2014, pp. 4965-4969.
9. T. N. Pham et al., "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies", in Proc. IEEE Conf. Bus. Infom., vol. 3, Sept. 2015.
10. D. J. Bonde, R. S. Shende, K. S. Gaikwad, A. S. Kedari, and A.
11. Bhokre, "Automated car parking system commanded by Android application," in Proc. Int. Conf. Comput. Commun. Inform. (ICCCI), Coimbatore, India, Jan. 2014, pp. 1-4.
12. Kamble, P., Chandgude, S., Deshpande, K., Kumari, C., & Gaikwad,
13. M., Smart parking system, 2018.
14. Khanna A., Anand R., "IoT based Smart Parking System", in proceed-ing of IEEE, IOTA, Jan. 2016.
15. R. E. Barone, T. Giuffre,` S. M. Siniscalchi, M. A. Morgano, and G. Tesoriere, "Architecture for parking management in smart cities," IET Intell. Transp. Syst., vol. 8, no. 5, pp. 445-452, 2014.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication