iPark: Intelligent Parking System for Commercial Buildings

Bhumi Prakash Mistry
Carnegie Mellon University
Electrical and Computer Engineering
bmistry@andrew.cmu.edu

Sai Bharath Reddy Mallavarapu
Carnegie Mellon University
Electrical and Computer Engineering
smallava@andrew.cmu.edu

Vaishnavi Rao
Carnegie Mellon University
Electrical and Computer Engineering
vaishnar@andrew.cmu.edu

Project Website: https://ipark6.webnode.com/

Table of Contents

I. MOTIVATION

As cities get more populated and more people use private transportation for regular commute, we face a dearth of places to park vehicles. People spend their precious time in searching for parking spaces, which can be utilized doing something more productive. According to the analysis by INRIX, Americans spend an average of 17 hours per year searching for parking, resulting in a cost of $345 per driver in wasted time, fuel and emissions. Moreover, the most significant reason for vehicular congestion in the urban areas is people wasting time in looking for a parking area[3]. It does not come as a surprise that around 63% of nearly 6000 U.S. drivers reported that they refrained from driving to shopping malls, airports and other destinations to avoid the hassles with parking. With more and more people investing in cars, parking space proves to be an important resource. For instance, as of 2016, there were more than 260 million cars and trucks, and an estimated 722 million to 2 billion parking spaces in United States[1]. These parking conditions are our motivation to come up with a better, more efficient car parking system.

IoT technology grows in various fields of smart applications, it is currently used in major avenues such as Lighting, Energy, Health, Consumer Electronics etc. This technology can be harnessed to help us overcome the drawbacks of current parking systems? To tackle the problem of inefficient use of parking spaces we propose an Intelligent Parking System(iPark) that leverages the scalability and reliability of Wireless Sensor Networks. The primary reason to choose Wireless Sensor Network (WSN) is due to its ability to provide scalable solutions with minimal change in the infrastructure. With advances in wireless technologies, WSNs prove to be a natural choice for sensing applications. Thus, we expected to find an efficient and cost-effective solution to the existing car parking problems[8].

The iPark system not only provides the user with real-time information regarding parking spots via a mobile application, but also to provides reservation functionality as part of user-targeted service. Built on advanced sensing and mobile communication techniques, iPark processes streams of sensing data from sensor network in parking lot and publishes parking availability information. The drivers can retrieve parking information and reserve vacant spaces using the application. The proposed system is completely automated apart from one person at the entry point to check the validation of the reservation[4]. Thus, we aim at providing a feasible, low-cost solution that caters to car owners.

A. <u>Literature Survey and Market Research:</u>

Let's set the background of our proposal by introducing some existing approaches and challenges for smart parking systems. D-Systems Project[5] presented various issues for a reliable WSN system using magnetic sensors and came forward with various recommendations on the use of dynamic & robust routing, delayed retransmissions, etc.[8]. Jatuporn et al.,[7] proposed optical WSN as a vehicle counting system in smart garages. They used two sensor heads to classify the objects of different sizes thereby distinguishing and identifying different vehicles[8].

Iris-Net[6] proposed a wide-area sensor network architecture with video cameras, microphones, and motion detectors to detect the availability of parking spaces in real time. However, the proposed system generated large amount of data resulting in high power scavenging and communication bandwidth which are the major limitations of WSN[8]. Lee, Yoon and Ghosh[4] proposed a hybrid approach for an intelligent parking system using a combination of ultrasonic and magnetic sensors; demonstrating promising results through various real-world experiments. Though the main goal was to count the number of vehicles on each floor and provide a cheap and accurate solution, the scope of their work restricts itself to vehicle detection using WSN than providing a smarter parking management solution[8]. Our Approach is closely related to this system in choice of sensors but focuses on the entire system rather than just the WSN for slot occupancy detection.

II. PROJECT GOALS

The goal of our project is providing an effective solution for the users who waste their time looking for a parking slot. We plan to accomplish it in the following way:

1. Correct detection of the presence/absence of a car in the parking slot.
2. Reliable transmission of this data over the network.
3. User-friendly mobile application.
4. Updating the data in real-time.
5. Providing reservation functionality on the user app.
6. Keeping track of the slot in which the car is parked.

III. USECASES

The iPark system is essentially designed for indoor parking lots with WiFi connectivity. Commercial parking lots are a typical use case for iPark. As the only constraint for the system is a WiFi connection, the system can be deployed in outdoor parking lots as well. Taking a step further, the system can be used to relay street parking information. Since our solution does not require a change in the infrastructure it can prove to be a feasible solution for efficient use of street parking.

Apart from enhancing customer experience, a smart parking solution also has a lot in store for businessmen and parking lot owners. Providing the customers with more information and a better experience, fosters goodwill. Financially, a low maintenance, responsive parking lot will be more marketable and help generate more revenue.

IV. SYSTEM DESIGN

In this section, we describe the design of the iPark System, consisting of a Wireless Sensor Network subsystem, cloud server, management subsystem, client reservation subsystem and an administrative control subsystem. Reiterating, the system shall be able to display real time information related to the availability of parking lots to the users and enable users to reserve parking slot. Figure 1 details the system architecture of iPark. The overall architecture is divided into five major subsystems as follows

A. <u>WSN Subsystem:</u>

WSN subsystems deals with monitoring parking slot status. It consists of sensor-controller pairs aka slave nodes which is powered by Raspberry Pi Zero W, directly interfaced to an ultrasonic sensor. Each node detects the presence/absence of a car in a parking slot and forwards that information to base station via Message Queuing Telemetry Transport protocol (MQTT).

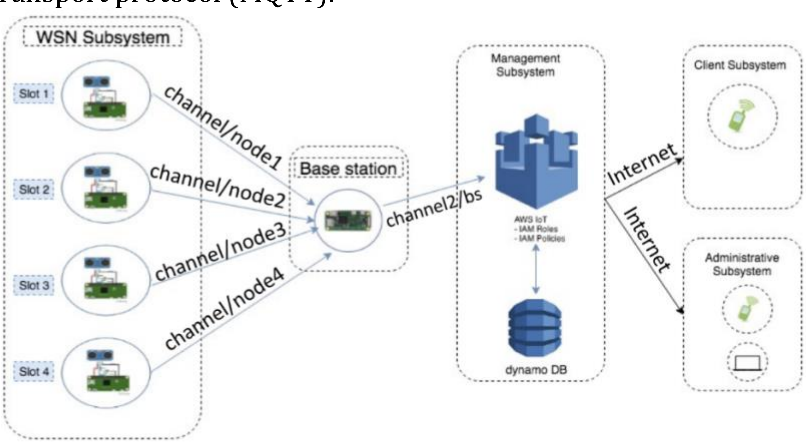


Fig. 1. System Architecture of iPark

B. Base Station Subsystem:

Base Station Subsystem collects parking slot status from the WSN subsystem and delivers it to Parking Management Subsystem via MQTT protocol. It acts as a gateway between wireless sensor network and external network.

C. Management Subsystem:

This subsystem acts as the heart of iPark System. The base station sends data to the AWS IoT core module in the management subsystem, which receives the data, processes it and forwards to the database module. The database module stores event-based sensor data and the parking slot status information of the sensor controller[8].

Whenever client reserves a parking spot from the mobile application, the reservation message will be forwarded to parking reservation information running on the database module. Also, Management system is responsible for retrieving sensor data and based on the availability of the parking slots, update to the android mobile application.

D. Client Reservation Subsystem:

Client Reservation subsystem is the one where clients interact with the iPark system. In this subsystem user has to login to the iPark Android application. This application mainly runs a background process which regularly updates the availability of slots and processes the inputs given by the user through the parking reservation GUI to send the information to parking management subsystem.

E. Administrative Control Subsystem:

Administrative control subsystem is the one where the parking staff interact with the iPark System. In this subsystem, the users with administrative access can control boom barrier. This subsystem functionality is to verify the user's reservation code and actuate the boom barrier based on the number of slots that are free in the lot.

V. SUB-SYSTEM INTERACTION

The WSN subsystem, as depicted in Figure 1, senses the status of parking space and transmits messages through MQTT protocol to the base station subsystem. The base station subsystem processes the messages received from WSN subsystem and sends the parking slot data to the AWS IoT core i.e., Management subsystem via MQTT protocol. This information is in turn stored in the Amazon dynamoDB. Administrative control subsystem gets slot availability from the management subsystem and manually actuates the boom barrier. Also, Client Reservation system(Android mobile application) gets slot availability from the management subsystem and based on the availability it enables the user to reserve the spot and the reservation status is sent to Management subsystem from client reservation subsystem.

VI. PROJECT SCHEDULE AND STATUS

We have implemented a working prototype for 4 parking slots. We conducted preliminary experiments, and in this section, we present our setup, results and project status with respect to the schedule:

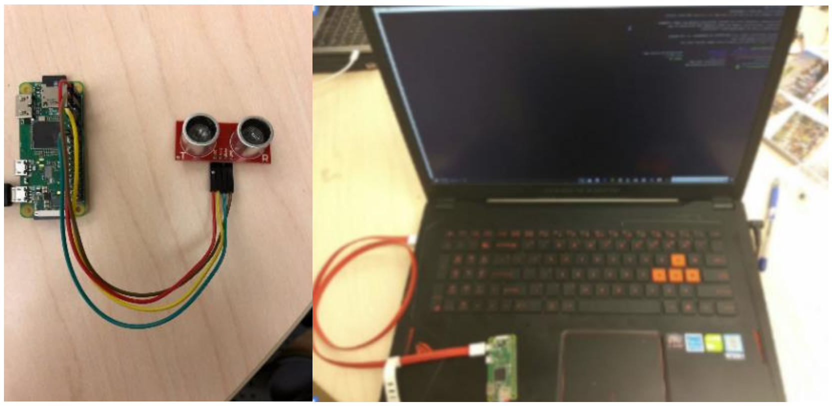
A. Device Specification

1) Sensor Node:

As a part of the model, we have used low cost Raspberry Pi Zero W boards with Raspbian Stretch Lite image installed. We also used HC-SR04 ultrasonic sensors, directly interfaced with RPi to detect the presence/absence of a car. This is shown in figure 2(a). These battery powered sensor nodes are placed at parking slots. To detect an event, we evaluate Mean > Threshold where mean is the average of 10 data samples from the ultrasonic sensor. Considering situations such as a person temporarily standing above the sensor, or an object placed on the sensor needs a less aggressive decision-making scheme as compared to determining the accuracy within one second. We assume that the value of 10 seconds is long enough for an object obstructing the sensors (other than a car), to move away from it.

2) Base Station Node:

Individual sensor nodes communicate with base station via MQTT protocol. The communication between Sensor nodes and base station is shown in figure 2(b). In this figure we see the raspberry pi zero which serves as a gateway node between WSN subsystem and parking management subsystem.



(a) Sensor node (b) Base Station

Fig. 2. Device Specifications

3) Mobile Application:

The graphical user interface is implemented in Android Studio. This application provides the user with real time information regarding the number of vacant spots in a parking lot and helps the user reserve a spot via the mobile application. The android application primarily deals with 3 functionalities:

a. *Making a new reservation*

The users can reserve a parking spot depending on the availability of the slots. Given that there is at least one slot available, the user can easily reserve a spot by clicking on 'Reserve a spot' as shown in figure 3(d). The user can see how many slots are available and then decide, as shown in the figure 3 below. The user is then provided with a verification code for verifying their reservation later. Also each user has a limit of only one reservation.

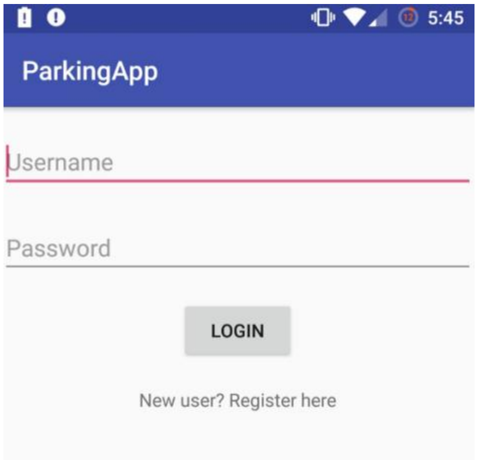
b. *Managing an existing reservation*

After reserving the spot, the user can use the verification code to later provide a proof of reservation as shown in figure 3(f). The code is linked with the user's profile and is used for actuating the boom barrier. Our application can also remember the parking slot for the users who do not want to forget their parking slots. The user can park his car and enter his slot number in the application. When leaving the parking lot, the user can click on the 'Tell me where I parked' button to retrieve the slot information as shown in figure 3(e). The figure below shows how the page appears to a user that has made a reservation. Along with this, we also enable the users to cancel their reservation in case they change their plans.

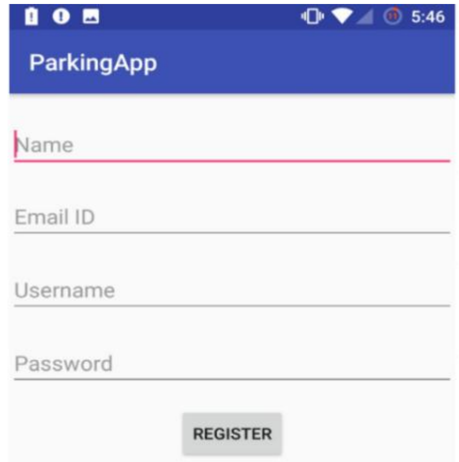
c. *Validating reservation*

The reservation for the user is matched at the boom barrier in the parking lot. Once the user enters the correct verification code in the application, the warden actuates the boom barrier to let the user in.

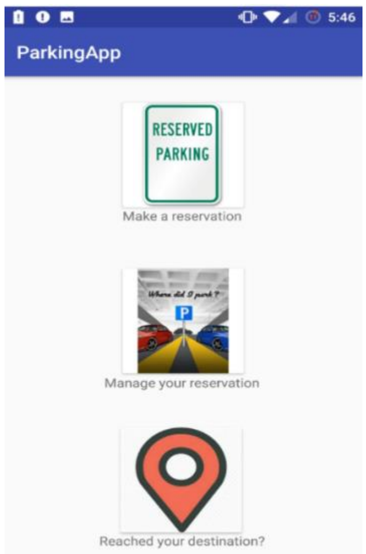
Apart from the above-mentioned functionalities, we have also implemented login (for existing users) and registration functionality (for new users) as shown in figure 3(a) and 3(b).



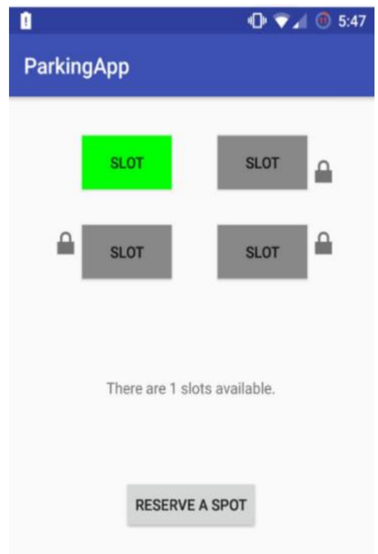
(a) Login page



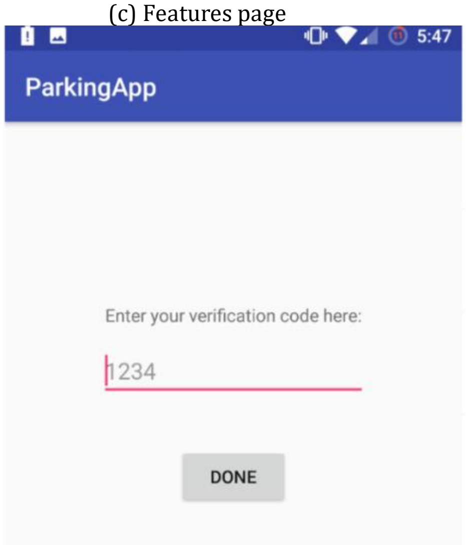
(b) Registration page



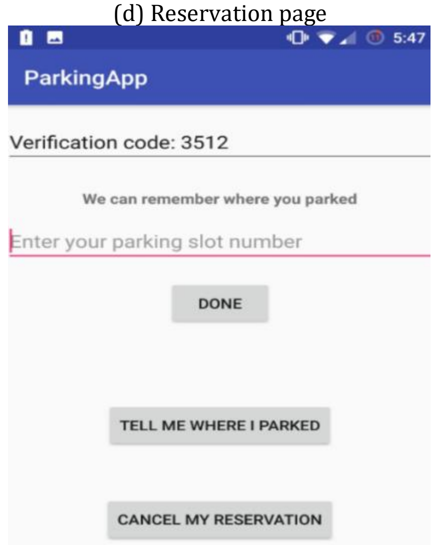
(c) Features page



(d) Reservation page



(e) Remember slot



(f) Authentication page

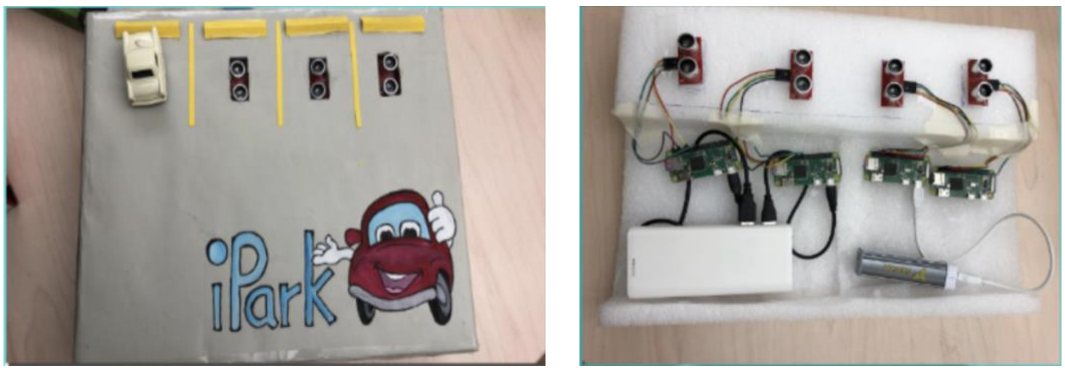
Fig. 3. Mobile Application

B. Experimental Setup

Our iPark System demonstrated using a cardboard model as shown in figure 4. The system includes 4 sensor nodes connected to batteries and 1 basestation system connected to the laptop.

Initially, all sensor nodes check the availability of each parking space and transmit the status report messages to the sink node. The sink node collects the status messages and delivers them to the management server. This information is stored in the database and will later be used by the management server to compute the vacant & occupied spaces. It further transmits the status information to the iPark mobile app[8].

Whenever a car is parked in one of the slots, sensor node (in interval of 10 secs) detects that the parking space is occupied and sends a value of "1" along with the sensor ID to the base station. Otherwise the sensor node sends a value of "0".



(a) Hardware model (b) Prototype model with one car parked

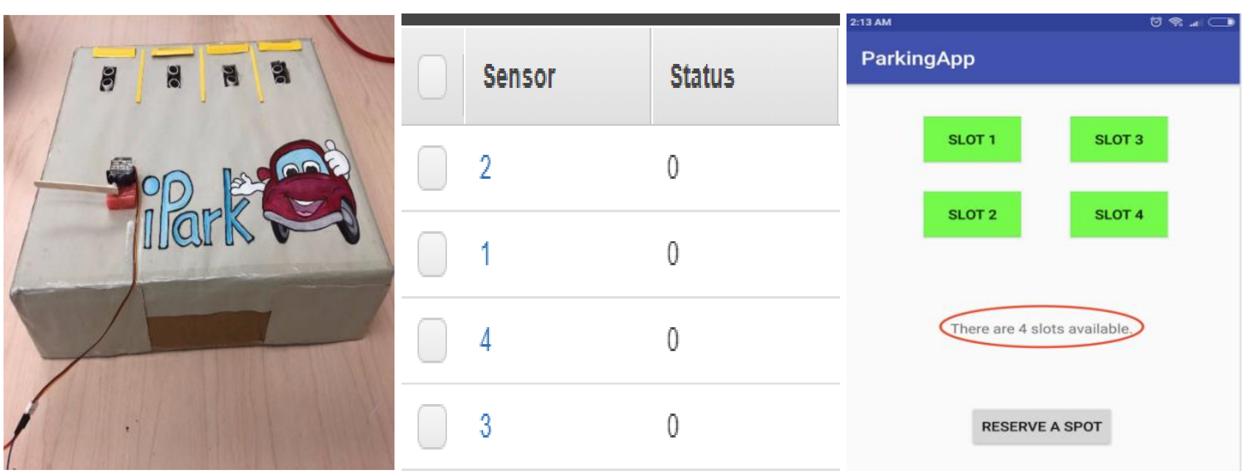
Fig. 4. Experimental setup

C. System Evaluation

iPark Prototype system is developed as a proof of concept to meet the real time requirements of parking systems. We consider 4 cases representing different situations and our system's responses to the same.

a) Scenario 1: Parking lot is vacant

When all the parking slots are vacant, all sensor nodes placed in the parking lot sends a value of "0" along with the sensor id and channel that it is transmitting as shown in figure 5(d). The mobile application shows total vacant spots as '4' and shown in figure 5(c).



(a) Parking lot (b) Database for slot information (c) Mobile Application

Fig. 5. Parking Lot Empty Scenario

```
test_channel/node1
{"Status": "0", "Topic_name": "test_channel/node1", "Sensor": "1"}
test_channel/node2
{"Status": "0", "Topic_name": "test_channel/node2", "Sensor": "2"}
test_channel/node3
{"Status": "0", "Topic_name": "test_channel/node3", "Sensor": "3"}
test_channel/node4
{"Status": "0", "Topic_name": "test_channel/node4", "Sensor": "4"}
```

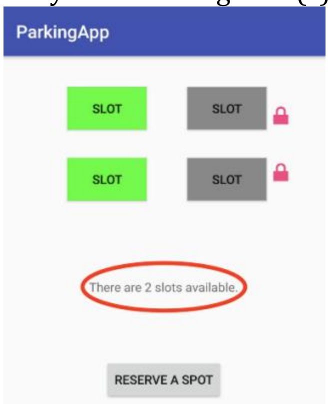
(d) Slot Information at Base Station

Fig. 5. Parking Lot Empty Scenario

b)Scenario 2: Two cars are parked

In this scenario, we park one car at sensor node numbers *one* and *four* each. From figure 6(d) we can deduce that sensor nodes *one* and *four* send a value of "1" whereas sensor nodes *two* and *three* send a value of "0". We see the expected output in mobile application about the availability of slots in figure 6(c).

Sensor	Status
2	0
1	1
4	1
3	0



(a) Parking Lot (b) Database for Slot Information (c) Mobile Application

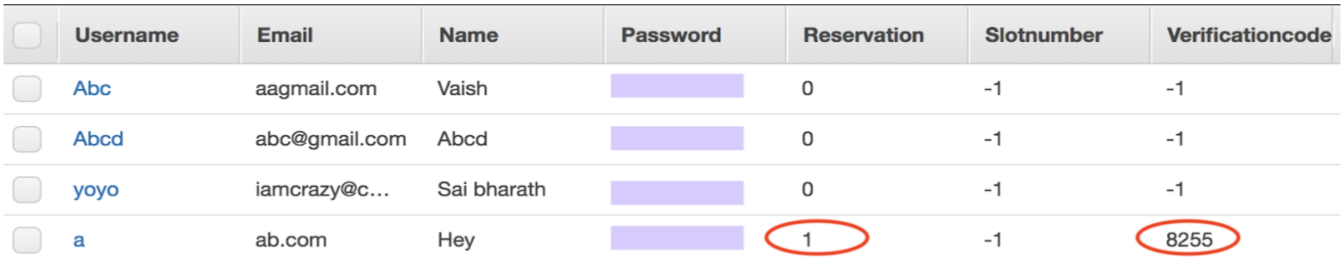
```
test_channel/node1
{"Status": "1", "Topic_name": "test_channel/node1", "Sensor": "1"}
test_channel/node2
{"Status": "0", "Topic_name": "test_channel/node2", "Sensor": "2"}
test_channel/node4
{"Status": "1", "Topic_name": "test_channel/node4", "Sensor": "4"}
test_channel/node3
{"Status": "0", "Topic_name": "test_channel/node3", "Sensor": "3"}
```

(d) Slot Information at base station

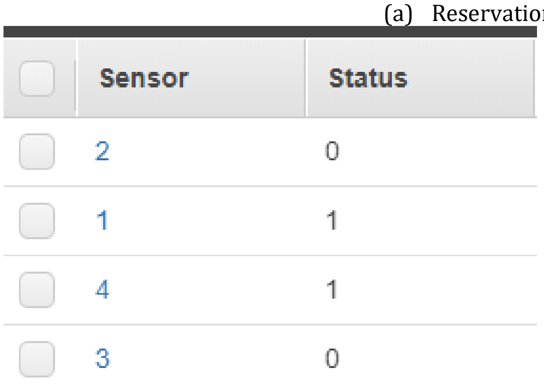
Fig. 6. Parking Lot With two cars parked

c) Scenario 3: Parking Reservation when slots are available

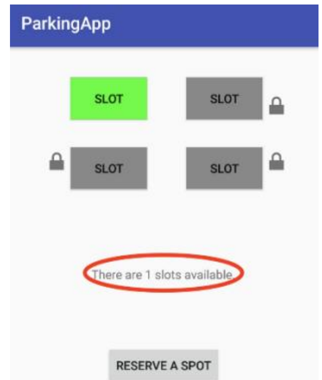
In this scenario, two cars are parked at slot *one* and slot *four*. The user tries to reserve a parking slot. The reservation database is shown in figure 7(a). Once the reservation is made we can clearly see from figure 7(b) and 7(c) that there is only one spot available even though there are two slots that are empty in the parking lot. This indicates that the reservation has been made.



(a) Reservation Database



(b) After Reserving

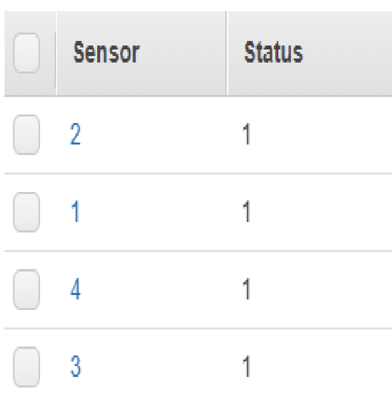


(c) Database after reserving

Fig. 7. Parking Lot Scenario 3

d) Scenario 4: Parking Reservation when slots are not available

In this scenario, Parking slots are not available, this is visible in figure 8(a) and then the user tries to reserve a parking slot. The user is unable to reserve it as there are no slots available. This is shown in figure 8(b).



(a) Database of slot information

(b) Mobile Application

Fig. 8. Parking Lot Scenario 4

D. <u>Milestones and Challenges faced</u>

Completion date	Milestone	Status
03/04/2018	Data sensing and correct detection of occupancy at the sensor nodes.	Completed
03/16/2018	Reliable link from sensor nodes to gateway	Completed
03/26/2018	Establishing a reliable link from gateway to AWS IoT core.	Completed
03/31/2018	Web UI for intermediate project demonstration (Node red)	Completed
04/14/2018	Mobile Application.	Completed
04/25/2018	Support reservation from mobile application.	Completed
04/27/2018	Extra-credit: Boom barrier actuation.	Completed
05/04/2018	Extra-credit: Historical data saving.	Attempted

Challenges:

1) Multihop network using Zigbee - We had planned to implement a mesh network using Zigbee. Serial port for Zigbee communication is inherently blocked in RPi Zero W. To enable this, we had to change the kernel files of the pi. This problem was overcome by implementing an alternative MQTT "push" and "pull" protocol for sending and receiving data. This worked well with our application and we chose to continue with the same.

2) Automating the boom barrier by sending command from user's mobile - We decided to send data to particle cloud for the boom barrier actuation. But "POST" commands to the cloud did not work. Hence, we have currently made a semi-automated opening and closing using a particle's mobile app to trigger .

3) Historical data saving - RPi Zero W has limited computing capabilities and on using the libraries need for historical data sending, the Pi became extremely slow. We thus let go of this functionality in our system.

VII. LESSONS LEARNED

The choice of ultrasonic sensors for detection worked well for us. Based on the RF signals, the sensor sense objects within the range of 2cm – 400cm, we worked towards calibration of the same and successfully detected occupancy. We used MQTT protocol as a backup for Multihop but later decided it to go ahead with the same as it is light-weight and scalable. This protocol also proved to be compatible with AWS IoT Core. AWS IoT subscribes to the WSN channel, retrieves the data and triggers DynamoDB automatically. This was simple to do with AWS IAM roles and Rules. With a little background in Android Studio, we were able to come up with a user-friendly application with required functionalities. The choice of Particle Photon for actuation of the boom barrier was a little problematic as Particle cloud post requests have limited documentation. However, it’s connection to the continuous servo motor was convenient and worked fairly well. For the historical data saving, AWS DynamoDB support for retrieving the entire table didn’t help us much. We then worked with Google Firebase was more straightforward and could have been a great choice if the we were not limited by the computational capability of the RPi Zero W.

VIII. WORK PARTITION

The following is the work partition including extra-credit, also the team collectively contributed towards the presentations, report, final model and setup.

Bhumi Mistry - Basic RPi setup and network configuration, sampling logic for abating false positives at sensor nodes, Android application frontend, historical data saving*.

Sai Bharath Mallavarapu - MQTT protocol, gateway to AWS IoT link, Android application backend(IOT-core roles).

Vaishnavi Rao – Node Red, Android application backend- AWS DynamoDB data retrieval and credentials, boom barrier actuation*.

*Extra-credit

IX. CONCLUSION AND FUTURE WORK

In this report, we described the feasible, low-cost, Intelligent Parking (iPark) management system using Wireless Sensor Networks. We designed a prototype as a proof of concept to realize and understand the real time scenarios in parking systems. Through our design we demonstrated that the proposed architecture can satisfy the requirements of a car parking management system and we believe that Wireless Sensor Networks can be a promising technology to solve future parking hassles.

There is a lot of potential for future work for such a system. We can provide fault tolerance in the system by checking for dead nodes at the gateway. A self-test unit can be implemented to make the single point of failure more robust. Historical data can be collecting for administrative purposes. Payment functionalities can be integrated in the android application to increase revenue for the parking lot owners. Also, an application can be made for other mobile operating systems.

X. REFERENCES

[1]https://www.usatoday.com/story/money/2017/07/12/parking-pain-causes-financial-and-personal-strain/467637001/

[2] http://cityobservatory.org/the-price-of-parking/

[3] Mr Basavaraju S R - "*Automated Smart Parking System using Internet of Things*" International Journal of Scientific and Research Publications, Volume 5, Issue 12,December 2005.

[4] Sangwon Lee, Dukhee Yoon and Amitabha Ghosh, "*Intelligent Parking Lot Application Using Wireless Sensor Networks*", Proceedings of IEEE conference, 978-1-4244-2249-4/08

[5] Barton, J., J. Buckley, B. O'Flynn, S.C. O'Mathuna, J.P. Benson, T. O'Donovan, U. Roedig, and C. Sreenan, The D-Systems Project - "*Wireless Sensor Networks for Car- Park Management*", Proceedings of the 65th IEEE Vehicular Technology Conference, Dublin, Ireland, Apr 2007, pp. 170-173.

[6] IrisNet: Internet-scale Resource-Intensive Sensor Network Service,

[7]Jatuporn Chinrungrueng, Udomporn Sunantachaikul, Satien Triamlumlerd "*Smart Parking: an Application of optical Wireless Sensor Network*", Proceedings of the 2007 International Symposium on Applications and the Internet Workshops (SAINTW'07)

[8] S.V Srikanth, Pramod P.J., Dileep K.P, Tapas S, Mahesh U. Patil, Sarat Chandra Babu N *"Design and implementation of a prototype Smart PARKing system using wireless sensor networks"*, Advanced Information Networking and Application Workshops, 2009. WAINA 09**.**