ECE 53301: Wireless and Multimedia Computing

Final Project Report - Group 8

Wireless Parking-lot Access System (WPAS)

Arjun N A	– Team Member
Kavyashree Prasad S P	– Team Member
Kunal Mandil	– Team Member

Department of Electrical and Computer Engineering,
Purdue School of Engineering and Technology at IUPUI.

Submitted to:

Course Instructor
Dr. Mohamed El-Sharkawy
Department of Electrical and Computer Engineering, IUPUI.
Submitted as partial fulfillment for the requirement of Fall 2019 - ECE 53301

Date of Submission
December 09, 2019

PROJECT SYNOPSIS

Wireless parking-lot access system (WPAS) is a project inspired by analyzing the difficulties of users using a swipe parking access system. WPAS was aimed at developing a user-friendly system that is capable of circumventing drawbacks of current systems. The project successfully exhibits a system prototype enabling users to gain access to the parking lot through simple push-button press avoiding inconvenient antiquated methods of opening the windows and swiping the magnetic stripe card for access. This improves convenience and faster access helping users to have lesser wait time during rush hours for parking.

WPAS facilitates users who cannot reach out to the QR code scanner by making the model available within the car. The project also puts forward the idea of energy-saving, as cold air will no longer enter the vehicle while enteringparking lots. It also reduces the inconvenience caused to the user whenever he/she forgets their swipe card as this module will be already present in the car.

Wireless parking-lot access system (WPAS) project demonstrates the use of thread protocol in parking lot access system. Thread is an IPv6 based protocol for 6LoWPAN (Low power wireless personal area network with IPv6) devices to communicate on a network. Thread protocol provides IP addressable devices where IPv6 of each element is unique. WPAS uses this feature of thread protocol to create access for registered users. WPAS also includesCoAP(Constrained Application Protocol) messages to communicate from vehicle to infrastructure (i.e., Gate controller).

ACKNOWLEDGEMENT

This project was a result of hardwork and dedication of all the group members;despite it, this will not have been a success without the help and support of many individuals and organizations. We acknowledge everyone who has helped us in the successful completion of the project.

We are highly thankful to our professor: Dr. Mohamed El-Sharkawy, and our teaching assistants: DewantKatare and Niranjan Ravi at the Purdue School of Engineering and Technology at IUPUI for their guidance and support in completing this project. Therefore, we would like to extend our sincere gratitude to all of them.

We are also thankful to Mr. Jeffrey Sears for providing us with access to the labs in the Engineering Science & Technology (SL) Building and with necessary tools to build, develop, solder, and test hardware components for this project. Without his support in providing necessary assistance and components, our team would have been greatly challenged to develop, integrate and test the hardware subsystems and components for this project.

We would like to extend our special gratitude to our seniors, especially Surya kollazhi, Niranjan Ravi, DevantKatare and Debjyoti Sinha, for extending their help whenever needed.

Without the help and aid of the above-mentioned people, who are always generous, the completion of this work would have been very difficult and might not have succeeded to the level it has reached today.

Sincerely,
Project group 2.

Table of Contents

LIST OF FIGURES

1.0 INTRODUCTION

Without dependable gateways, effectively managing access control is not possible. Many systems have come into existence to solve the problem of unmanned gates. These systems, though solve the problem bycreating automated parking-lot access, come with their own downsides as they demand users to open their windows, require time-consuming processing and have large queues during rush hours.

Wireless Parking-lot Access System (WPAS) is designed to efficiently control access and exit gates of parking facilities. One of the notable benefits of wireless parking management solutions is that vehicles can be permitted to enter or exit the parking facility without human intervention on the gate control. Instead of swiping magnetic stripe cards,WPAS enables users to simply approach the gate and just press the push button installed in the vehicle. The gate controller will authenticate the user's vehicle and allow it to pass through.

Other features of WPAS include security, reliability, fast access, which can be easily integrated with existing systems and can be upgraded to different forms. WPAS eliminates the need for parking attendants and avoids causing inconvenience to drivers, especially those who find difficulty in reaching the card swiper or keypad from driver's seat and, therefore, must routinely get out of the vehicle to open the gates. Wireless Parking lot Access Systems are ideal for airport parking, gated communities, college parking, municipal parking, and more.

2.0 MOTIVATION

Wireless Parking-lot Access System is inspired by observing user difficulties in the current parking lot system of the college. It was observed that users must open the windows to swipe their magnetic stripe card to gain access to multi-level parking garages. Considering harsh winters, it will be difficult to do the same as opening windows are not preferred as it leads to unnecessary loss of energy. This can be avoided with the proposed prototype.

The system also addresses the difficulty of differently abled people, as some may not be able to use current systems due to height differences or stretching from the driver's seat. This drawback can be eliminated with the usage of pushbuttons on the board. The user can press a switch to join the network created by the gate controller and push another switch to send a coap message containing its ipv6 address to get authorized access while sitting in the car. This feature has an added advantage of eliminating the delay or inconvenience caused to the driver if he/she forgets their swipe card.

3.0 SYSTEM ARCHITECTURE

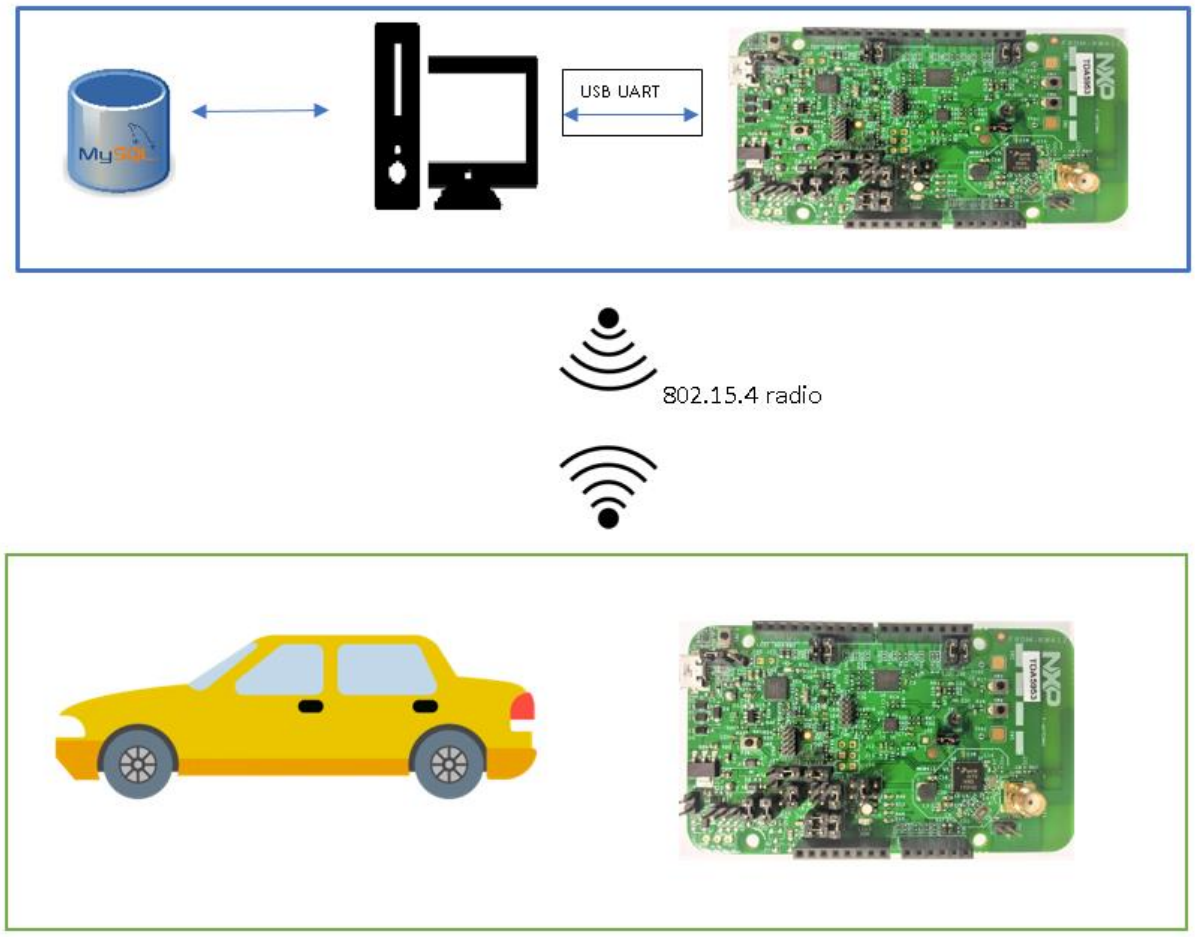

Figure 1.System Architecture

The system consists of two sections, the gate controller and end device. The system mainly works in thread network. The gate controller is configured as a leader router with no restriction on joiners. This enables any user to join the gate controller and the joiners are programmed as end device. The end devices connecting to gate controller creates a thread network. End devices are configured to send a request to open gates via a CoAP message. This enables the gate controller to identify the ipv6 address of the joiner and thus connect to database containing authorized ipv6 addresses and hence accept or decline the request.

Detailed hardware and software design is in the following sections.

3.1 HARDWARE DESCRIPTION

3.1.1 FRDM-KW41Z

FRDM-KW41Z is a development board provided by NXP Semiconductors built on ArmCortexM0+ processor with integrated 2.4 GHz transceiver supporting BluetoothLow Energy (BLE) v4.2, IEEE802.15.4, Generic FSK and Thread. The following is Kinetis FRDM-KW41Z pinout (1).

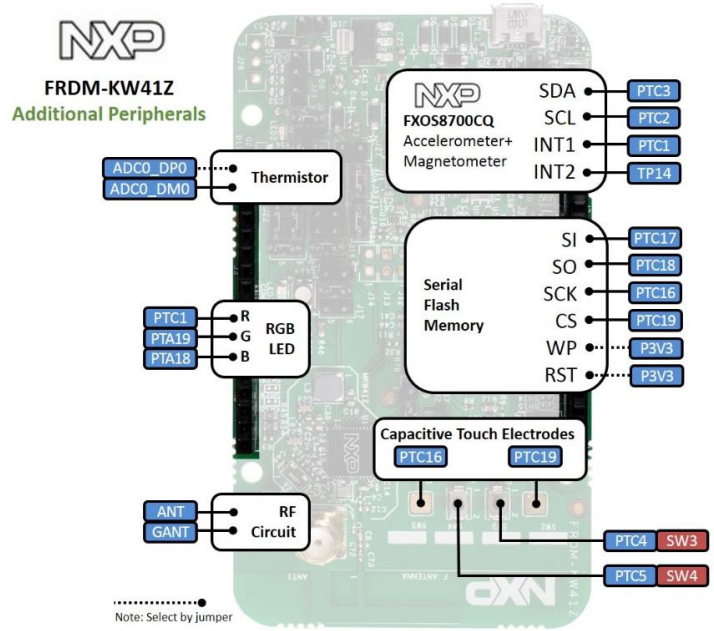

Figure 2.Component Pinout

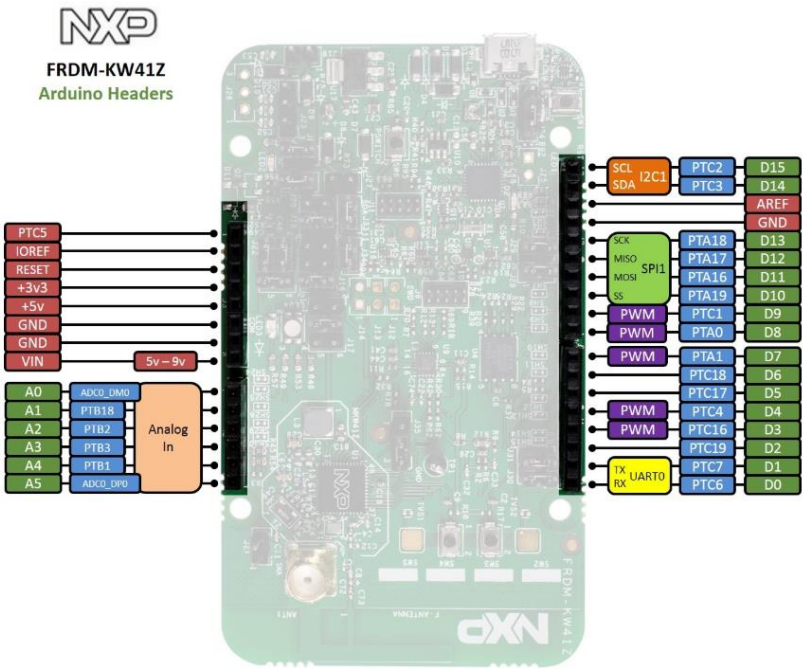

Figure 3. FRDM-KW41Z Header Pinout

3.1.3 COIN BATTERY

A Coin Battery or button cell is a single small battery, shaped like a cylinder typically 5 to 25 mm resembling a button. The negative terminal of the battery is an insulated top cap. Coin cell batteries are used for low power and small portable electronics (3).

Figure 4. Coin battery

3.2 HARDWARE DESIGN

The hardware design of WPAS contains two independent modules, namely, ‘Gate controller’ and ‘End device’. The gate controller, as the name suggests, is installed at the gate and the end device is installed in cars.

3.2.1 CIRCUIT DIAGRAM

A. GATE CONTROLLER

Gate controller is the most important part of WPAS which is installed at the gate to communicate with mechanical actuators through GPIOs and software through serial communication to control access and exit of end devices. The following figure provides a view of the circuit installed in the gate controller

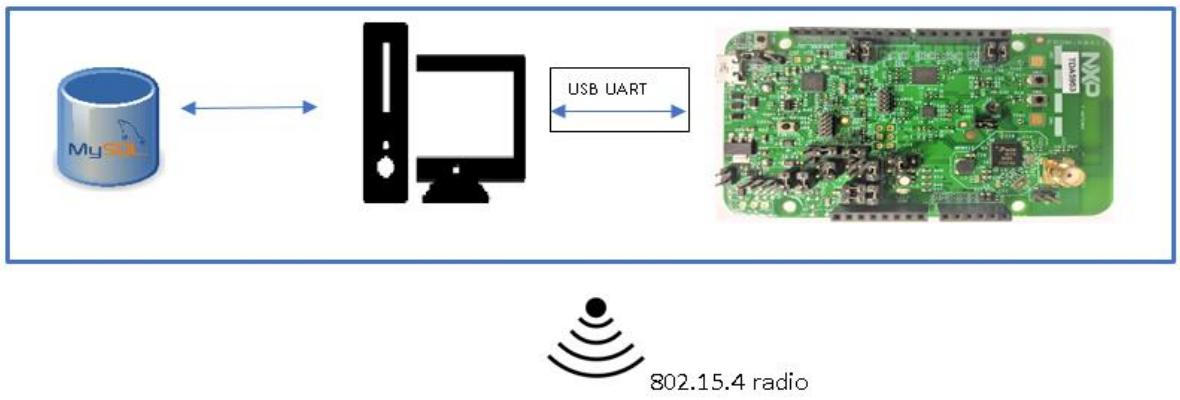

Figure 5.Hardware setup at gate controller

The Servo motor is connected to FRDM KW41Z via the control pin with PWM capabilities. The ground and power pin of the servo motor is connected to ground and power pins of KW41Z respectively. The servo is provided a supply of 5 Volts from the KW41Z power line.

B. END DEVICE

The end device is a standalone KW41Z board powered using coin cell batteries. It will be integrated in the vehicle of the user.

3.3 SOFTWARE DESIGN

The software was developed in MCUXpresso using SDK examples provided by NXP semiconductors in embedded C programming. Software design of the prototype consists of the following steps:

1.Configuration of Gate controller to leader mode- The thread device employed in the gate controller of the parking system was configured in leader mode. This active router enables the joining of end devices into the network with the same passphrase as itself and any EUI 64-bit device address (4).

2.Configuration of end devices- The thread device attached to the vehicles was configured as end devices.

3.Network creation-Every vehicle that enters the parking lot joins the network through the active router. This is achieved by pressing any switch except the reset button of the end device board.

4.Requesting access -Each end device after joining the network send their request of access to the gate controller as a CoAP message and disconnects from the thread network by pressing switch 4 on the board. The leader receives the message along with their Ipv6 address (5). This is then considered by the leader for authentication purposes to ensure only authorized users get access to use their parking space.

Following code depicts the initialization and transfer of coap messages from the end device

```
coapSession = COAP_OpenSession(mAppCoapInstId);
if(NULL != coapSession)
{  coapMsgTypesAndCodes_tcoapMsgType = gCoapMsgTypeNonPost_c;
coapSession->pCallback = NULL;
FLib_MemCpy(&coapSession->remoteAddrStorage.ss_addr,
&gCoapDestAddress,   sizeof(ipAddr_t));
ackPloadSize = strlen((char *)message);
coapSession->pUriPath = (coapUriPath_t*)&gAPP_URI_PATH;
    if(!IP6_IsMulticastAddr(&gCoapDestAddress))
    {
     coapMsgType = gCoapMsgTypeConPost_c;
     COAP_SetCallback(coapSession, APP_CoapGenericCallback);
    }
COAP_Send(coapSession, coapMessageType, message,
ackPloadSize);
}
```

5.Connection to database- This is achieved using python's serial communication package PySerial. This package facilitates communication with the serial port of the leader device to

obtain the received ipv6 address from the end device. Function serial.Serial() was used to access the serial port. It contains parameters namely

- Port number- This field involves specifying the COM port utilized by the gate controller device.
- Baud Rate- This field involves specifying the baud rate used in the communication of the gate controller device.

The function readline() from Pyserial module was used to retrieve data from serial port i.e. fetching ipv6 address sent by the end device.

6.Database search-After reception of ipv6 address from the end device, the gate controller, connects to the localhost database. This was implemented using MySQL. Python connects to the database using the Pyodbc module. The database contains ipv6 addresses of authorized users. If the received ipv6 address is present in the database, access is provided to the user.

7.Gate opening-Once the user is authorized, a message is sent serially through python to the COM port of the leader to activate the motor enabling the opening of gates for the user to pass through. This message sent serially is processed by the board using the `shell_ProcessChr` function.

3.3.1 SOFTWARE FLOW DIAGRAM

Flow charts in the software architecture help disintegrate various requirements of the prototype to address each task in a stepwise manner. It is a pictorial representation of the proposed solution for the envisioned problem. This model for parking access has two parts to be addressed - gate controller software design and user interaction software design. Flow charts for both these requirements are displayed below.

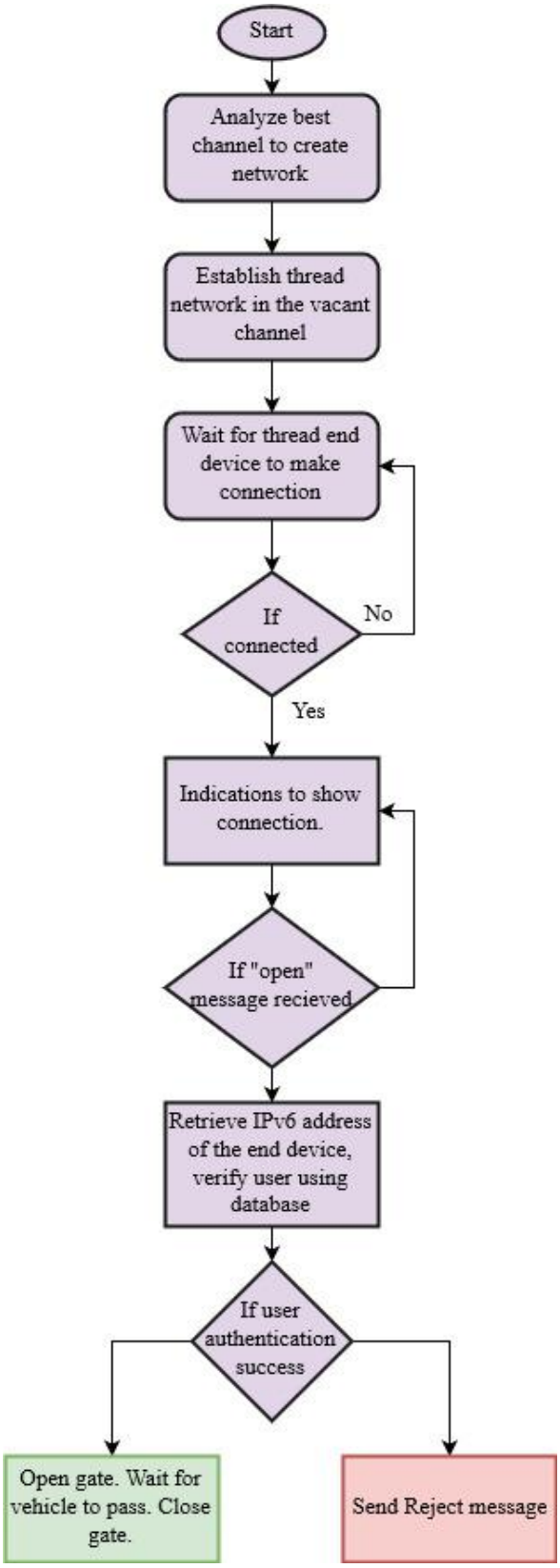

Figure 6. Flow chart for gate controller software design

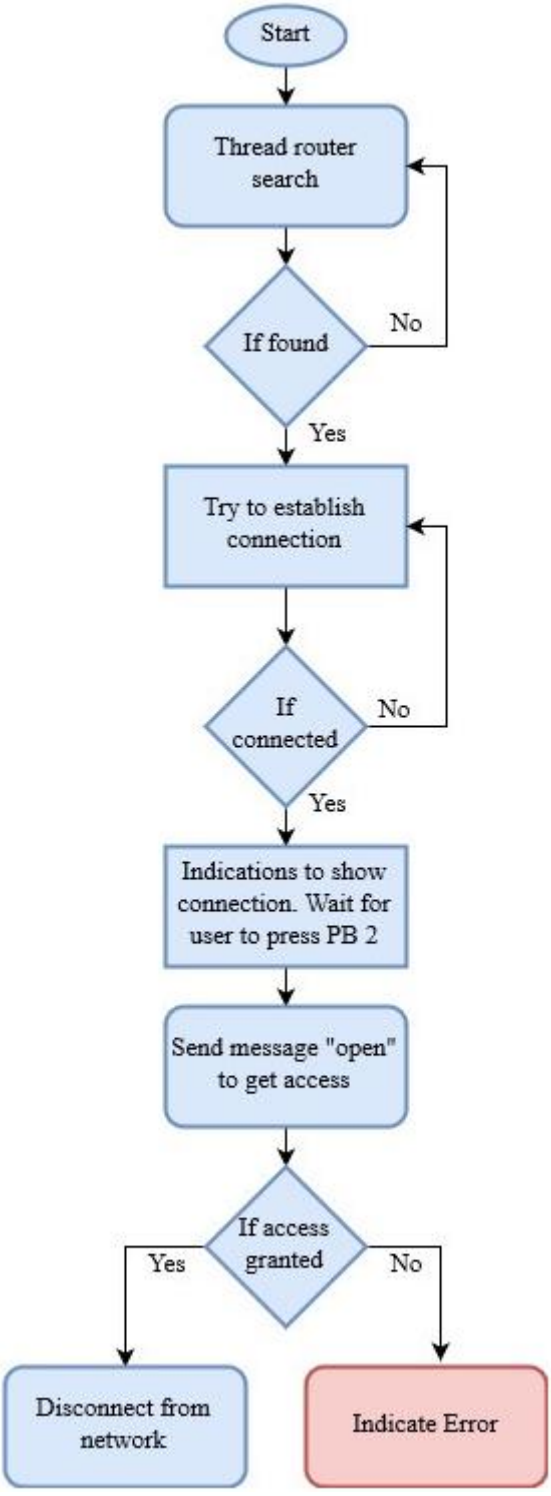

Figure 7. Flowchart for user interaction software design

3.4 ACTIVITY DIAGRAM

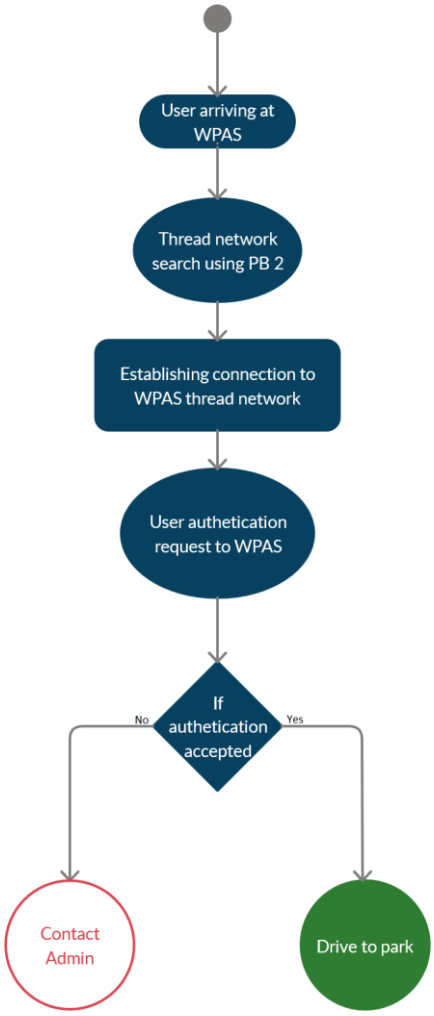

Figure 8. Activity Diagram

Figure [8] is the user activity diagram that demonstrates how this product should be used. This module can be integrated with the car in such a way that only a push-button and an LED must be visible to the user. The user arriving at the toll gate can see that the module gets connected to the gate easily through the LEDs going off. After connection, the user hasto press the push button (PB2) on the board to gain access. If the authentication is accepted, the gate opens, and the user can drive in, to park.

4.0 RESULTS

The project was successfully completed, and the following outputs were received.

Figure 9 indicating acceptance and denial of vehicles depending on the stored ipv6 address in MySQL database. First entry was an unknown device and it was rejected by the program and second entry was authentic user and the user was granted access.

Figure 10 shows the access of MySQL database using phpMyAdmin interface. Left side shows stored ipv6 addresses and right side is logging details of the user.

```
F:\Fall19\wireless_proj>python serial_python.py
fdc4:b10f:17b1:f9d1:44a6:c778:a10b:7edc
 Access Denied
fdc4:b10f:17b1:f9d1:509e:489a:999f:c118
 Access Granted
```

Figure 9. Python Output from Terminal.

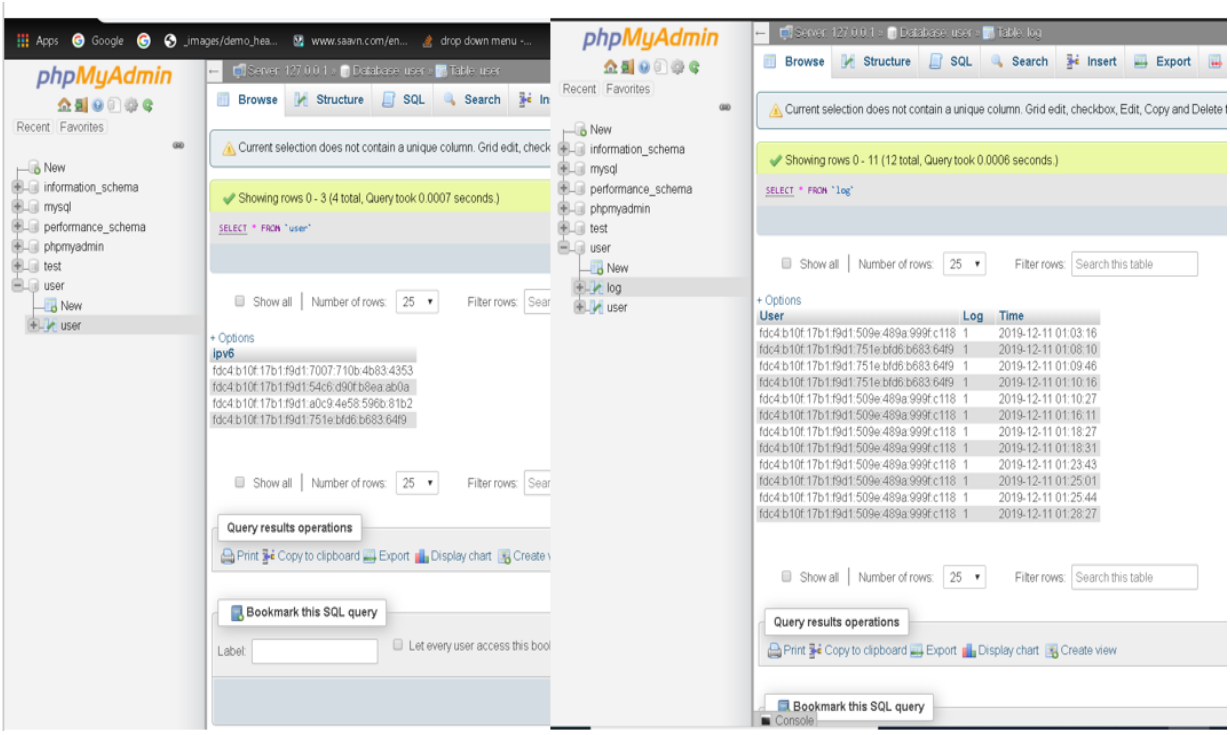

Figure 10. MySQL database.

5.0 CONCLUSIONS

The project Wireless Parking Access System (WPAS) suggests a prototype that can replace existing systems for parking-lot access to be more user-friendly. The system is working on wireless protocols that establish a thread network through IEEE 802.15.4 radio communication. Constraint application (CoAP) is run on both gate and car system to transfer messages as these messages will be small, thus applicable in low power boards like FRDM KW41Z. WPAS transacts with the end device using a unique IPv6 address leading to enhanced security and user identification.

The project successfully demonstrates that we can utilize the potential of thread network for parking lot access system. One major hassle that the users face in a busy parking lot is that they must go through long queues, approximately each car takes about 50-60 secs to pass through the gate. While in our model it just takes about 10 secs for each car to pass through as the connection can be established even before reaching the gate.

This model can be easily integrated into current gate controllers. An additional feature that could be added is logging timing information of users to generate payments based on their usage of parking space. A unified server can be used instead of a local host which would improve the efficiency of the system.

6.0 REFERENCE

1. FRDM-KW41Z |Bluetooth Thread Zigbee enabled Freedom Development Kit | NXP. (2018). Retrieved December 9, 2019, from Nxp.com website: https://www.nxp.com/design/development-boards/freedom-development-boards/wireless-connectivy/freedom-development-kit-for-kinetis-kw41z-31z-21z-mcus:FRDM-KW41Z.

2. Wikipedia Contributors. (2019, September 26). Servomotor. Retrieved October 18, 2019, from Wikipedia website: https://en.wikipedia.org/wiki/Servomotor.

3. Wikipedia Contributors. (2019, March 24). Button cell. Retrieved April 9, 2019, from Wikipedia website: https://en.wikipedia.org/wiki/Button_cell.

4. O. Orrie, B. Silva and G. P. Hancke, "A wireless smart parking system," *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, Yokohama, 2015, pp. 004110-004114.doi: 10.1109/IECON.2015.7392741

5. UserManual.wiki. (n.d.). Kinetis Thread Stack Demo Applications User's Guide. Retrieved from https://usermanual.wiki/Document/Kinetis20Thread20Stack20Demo20Applications20Users20Guide.1600353141/view