Park King: An IoT-based Smart Parking System

C. Ajchariyavanich[1], T. Limpisthira[1], N. Chanjarasvichai[1], T. Jareonwatanan[1], W. Phongphanpanya[1], S. Wareechuensuk[1], S. Srichareonkul[1], S. Tachatanitanont[1], C. Ratanamahatana[2], N. Prompoon[2] and M. Pipattanasomporn[3,4]
[1]*International School of Engineering, Chulalongkorn University, Bangkok, THAILAND*
[2] *Department of Computer Engineering, Chulalongkorn University, Bangkok, THAILAND*
[3]*Smart Grid Research Unit, Department of Electrical Engineering, Chulalongkorn University, Bangkok, THAILAND*
[4] *Bradley Department of Electrical and Computer Engineering, Virginia Tech – Advanced Research Institute, Arlington, VA, USA*

***Abstract*—As many universities around the world are currently promoting the smart campus campaign, one possible area to be revolutionized is the parking system. By integrating Internet of Things (IoT) technologies, university's parking systems can be automated and parking spaces can be managed efficiently. In this paper, we present the development and prototyping of Park King – an IoT based cloud-integrated smart parking system for a smart campus. Park King consists of: (i) the IoT module that allows monitoring the availability of each parking spot and controlling a parking flap; and (ii) a web-based application that allows users to reserve a parking space in advance. The system overview, its functional and non-functional requirements, tools and technologies used, prototype development/deployment, together with results from field testing and demonstration, are discussed in this paper. It is expected that this system can serve as a guideline and provide an insight into the development of a smart parking system in a university campus and/or a smart city.**

Keywords— Internet of things (IoT), smart parking, smart campus, smart city.

I. Introduction

In a typical university campus, there are many parking buildings located around the campus. For example, Chulalongkorn University in Thailand has three parking bulidings. These parking buildings have no intelligence and are operated manually by cashiers and security guards. By conducting surveys and interviews the users at Chulalongkorn University, i.e., students, faculty members and staff who drive and park at the university, the team realizes that the main user's pain point is to search for an available parking spot. That is, it takes more than 15 minutes to drive inside a parking building to realize that there is no parking spot available. In addition, for a vehicle to enter the university parking buildings, a parking sticker is required, which is manually checked by a cashier who looks for the sticker at the front windshield of a vehicle.

Hence, the team has identified the following problem statements to be addressed:

- How might we revolutionize the current parking system at a university campus with IoT technologies to create a more convenient environment for students, professors, and staff?
- How might we allow users to make reservation for a parking spot before arriving at a parking building?
- How might we improve the parking sticker system?
- How might we minimize the parking staff required?

Since a smart parking system is quite well-known in the technology field, lessons learned from existing projects can be very well serving as the basis to better design and improve our project. To start with, the team searched for examples of smart parking systems. The self-service car parking control system by Aida Engineering, Ltd. [1] deploys an automated parking flap, thus removing a worker from the parking site. Daimler AG [2] has developed the Parking Reservation App that allows reservation of parking spot(s) for Mercedes subscribers. The parking service system from Advantech Co., Ltd. [3] is deployed as the self-check-in/check-out system to reduce the amount of worker in the parking building. The intelligent parking solutions from Siemens [4] utilize sensor(s) to detect whether a parking lot is occupied on the side of the street. The car parking solution from Smart Parking Ltd. [5] is the automated parking system that lets users check the occupancy of the parking site. TheParkingSpot [6] is the airport parking reservation website to reserve the spot using boarding pass. The automatic car parking system from Shin Woo UBiCos [7] is a vertical-type tower. Authors in [8] also discuss a smart parking system using IoT devices and mobile application to manage the parking spot. Another smart parking system is discussed in [9] that relies on a mobile application for reserving the parking spot. Cloudparc [10] also equips its smart parking system with data analytics to detect vehicles using cameras and predict the parking spot occupancy.

Based on the above examples, our team targets to develop and prototype the IoT-based smart parking system for parking spot reservation at a university campus. The uniqueness of this work is the integration of IoT devices and a cloud-based server to allow monitoring and control of the entire parking system at a university campus. With this approach, the traditional parking sticker/paper can be eliminated and replaced with an electronic QR code. Furthermore, the prototyped smart parking system is in line with the smart campus initiative and the cashless society theme being implemented at many university campuses around the world, like Chulalongkorn University.

II. Technology Description

The system overview, tool/technology building blocks, as well as functional and non-functional requirements of the Park King system are discussed below.

A. System Overview

Park King is a system consisting of hardware and software designed to manage the entire parking system at a university campus. Key Park King functions are: to check the availability of parking spots, to reserve a parking spot in advance and

978-1-7281-0846-9/19/$31.00 ©2019 IEEE

receive a QR code, and to scan the QR code to access the parking building and park at the assigned spot. See Fig. 1.

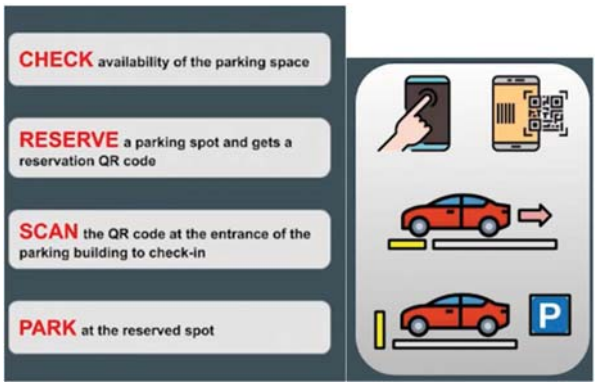

Fig. 1. Park King's system overview.

B. Tool/Technology Building Blocks

Our key tool/technology building blocks include hardware (i.e., sensor/controller module) and a web-based application, as discussed below.

Hardware components are: (i) a QR code scanner that allows a user to check-in using a system-generated QR code after the reservation is made; (ii) a parking flap (controlled by a servo motor) that syncs with the online reservation system so that it can open and close when the reserved vehicle arrives; (iii) an ultrasonic sensor that checks if a vehicle is still parked at a parking spot; and (iv) NodeMCU to monitor and control all three devices mentioned above, and connect them to the Internet. Arduino IDE was used to program NodeMCU to retrieve distance readings from ultrasonic sensors and send control command to a servo motor attached to a parking flap. NodeRed was used for retrieving information and sending control commands to IoT devices via MQTT protocol. Table I summarizes the hardware components of the system.

TABLE I. HARDWARE/SOFTWARE COMPONENTS

Name	Description
QR Code reader	Read the QR Code to gather the arrival and departure times of a vehicle at the reserved parking spot.
Parking Flap + servo motor	Open and close at arrival and departure of the user who reserves a parking spot.
Ultrasonic Sensor	Measure the distance between the ceiling and the floor/vehicle to check the existence of a vehicle.
NodeMCU	CPU board to control hardware components and provide Internet connectivity.

Software tools used to develop the Park King's web application are Atom by GitHub that enables the team to work collaboratively without being at the same place; JavaScript as the back-end development coding language; HTML/JavaScript/ CSS as the front-end development coding languages; as well as NodeJS. With respect to the database, Microsoft Azure—an open platform to manage relational-based databases—was used. The tool allows developers to easily manage the database with simple SQL commands and has many useful features, including real-time data sync, data query, and cloud integration.

C. Functional and Non-functional Requirements

Park King has been designed with the following functional requirements: the system shall allow the user to register his/her information into the system and add vehicle into his/her account. The system shall allow a user to see the number of available spots in real time, allow a user to reserve the parking spot, and cancel his/her reservation if a user does not arrive at the reserved spot within 15 minutes after the reservation is made. In addition, the system has been designed to allow a user to reserve only one parking spot at a time. The user can also make online payment through e-banking application.

Non-functional requirements are as follows: the system shall operate on any web browsers, communicate with the QR code reader and parking flap wirelessly, and automatically back up all user's data. With regard to the payment, the system shall redirect to payment page after scan the QR code, and keep track of only the transaction number after the payment. The system should disable and enable the parking flap within three seconds or less. The system shall allow the user to update his/her data, view his/her reservation history, and allow only admin to alter the parking spot information.

III. USE CASES AND ENTITY-RELATIONSHIP MODEL

This section explains the project's use cases and data model.

A. Park King's Use Cases

The project's use case diagram is shown in Fig. 2. There are five use cases, which can be explained as follows.

Use Case 1-User registration: The initial step for a user to use our IoT-based parking management system is that the user must register for our web-based application.

Use Case 2-Vehicle registration: A user must also register his/her vehicle by providing the license plate number, vehicle's make and model, and picture(s) of the vehicle.

Use Case 3-Check for an available parking spot: When a user would like to park at a parking building, the user must first login into the website and check if the parking spot is available.

Use Case 4-Parking Spot Reservation: When a user would like to reserve a parking spot, the user must choose a registered vehicle, a parking building, and a payment method for reservation. The system then validates the user's cancellation frequency. After which, the system checks if there is an available parking spot at the selected parking building. After everything is verified, a parking spot is reserved and a QR code is generated for the user to enter the selected parking building.

Use Case 5- Payment: When a user would like to leave, the user must pay a parking fee before leaving the parking building. The system will check the QR code and the parking spot status to determine the fee and allow the user to make payment.

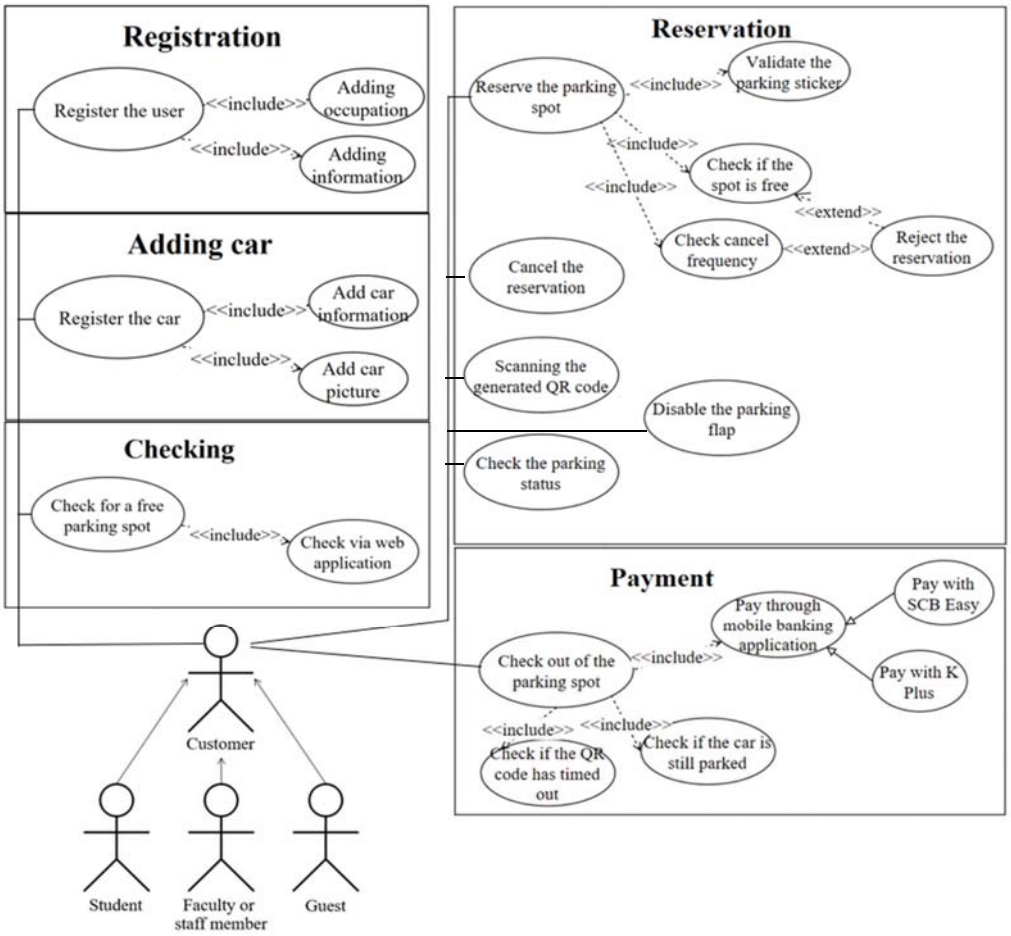

Fig. 2. Park King's use case diagram.

B. Entity-Relationship Model

The Entity-relationship model consists of four entities, i.e., customer, vehicle, parking building and parking spot, and two relationships, i.e., reserve and transaction receipt. These entities and their relationships are discussed below.

Customer entity: This entity is responsible for collecting user information, including username, password, email, first name, last name, customer type (i.e., student, faculty/staff and guest), customer picture and cancellation history.

Vehicle entity: This entity is responsible for collecting the vehicle information, including plate number, car brand, car model, car color and car picture. This entity also has a relationship with the customer entity.

Parking building entity: This entity is responsible for collecting parking building information including building name and parking capacity.

Parking spot entity: This entity is responsible for collecting parking spot information for each building, including floor number of the parking spot, the parking spot number, information from a sensor at a parking spot, and the parking spot status. This entity also has relationship with the parking building entity.

Reserve relationship: This relationship is responsible for collecting the user's reservation information including reservation ID, QR code in, QR code out, time in, time out and paid status. This relationship links the vehicle entity and the parking spot entity together.

TransactionReceipt relationship: This relationship is responsible for collecting the user's transaction information including transaction ID, fee, payment method and total parking time. This relationship links the vehicle entity and the parking spot entity together.

IV. Prototype Development and Software Quality Assurance (SQA) Testing

This section discusses the prototype Park King system, which comprises both the hardware prototype and the web-based application, as well as SQA tests and results.

A. Hardware Prototypes

The sensor module was developed using NodeMCU and an ultrasonic sensor. See Fig. 3. The sensor module prototype was powered by batteries, and was placed on the ceiling of a parking lot where it could sense the presence of a vehicle. The distance between the sensor location (i.e., ceiling) and the roof of a vehicle/the ground level was measured. These distances were used to identify the presence of a vehicle.

Fig. 3. The sensor module prototype and its deployment at a parking spot.

The scaled-down version of the parking flap prototype was made from a cardboard and plastic materials. See Fig. 4. Its control unit (inside the cardboard box) comprises NodeMCU connected with a servo motor. The objective is to demonstrate how responsive the flap can be in responding to the command issued by the system.

Fig. 4. The parking flap prototype.

A laptop camera was used as a camera for scanning the QR code when a user checks in and checks out of the system.

The line graph in Fig. 5 shows an example of the recorded distances from the ultrasonic sensor installed at the test parking lot during six hours of one evening. It can be seen that during minutes 0-21, the measured distance was around 250 cm, which implies that the vehicle was not present. The vehicle returned during minutes 22-45, during which the measured distance reduced to around 100 cm. The vehicle then left again during minutes 46-82, and returned again at minutes 83.

With respect to lessons learned from the hardware prototype deployment at the test parking lot location, since the sensor module operated on Wi-Fi and sent real-time sensor readings to the database at one minute intervals, the batteries depleted within a day and a half. Therefore, future implementation will require the sensor module to be plugged in to the wall outlet. In addition, it can be seen from Fig. 5 that the sensor readings are sometimes noisy. Hence, an appropriate threshold must be identified to classify whether the vehicle is present or absent.

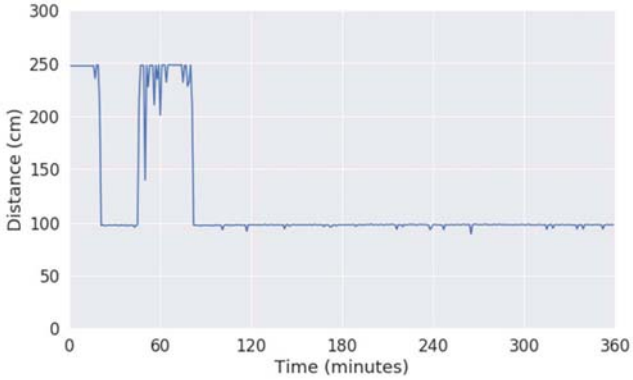

Fig. 5. Readings from the ultrasonic sensor installed at the test parking spot.

B. *Prototype: Web-based Application*

Park King's user interface (UI) was designed as a web-based application to accomplish all the functions specified in the functional requirement section (Section II(b)). These are ranging from logging in to checking and reserving a parking spot. It is our goal to have a minimal and user-friendly UI, so that users can quickly understand and use the system without having to read any user's manual. In addition, the web-based application must work seamlessly along with the hardware prototypes discussed earlier. Figs. 6-8 illustrate the Park King's UI examples.

After a user logs in, the first page is the home page, as shown in Fig. 6 (left). It contains information about available parking spots in each parking building, including number of available spots, the lowest floor with an available spot and the parking spot ID to be assigned to the user if a reservation is made. To reserve a parking spot, the user can click "Reserve" button on the top right. The reservation page appears (See Fig. 6 (right)), where the user can select his/her vehicle and the parking building he/she wishes to park, and submit the request.

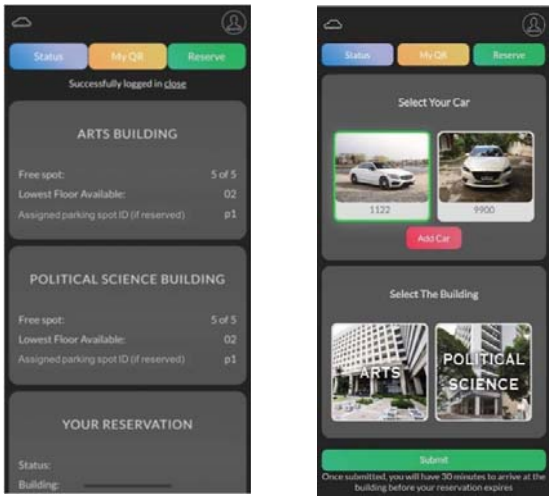

Fig. 6. ParkKing web application: home page (left); and reservation page (right).

Then, the user needs to make payment using the payment page (as shown in Fig. 7 (left). After the payment is made, the QR code page (as shown in Fig. 7 (right)) appears. This code can be scanned at the scanning station to check-in to and check-out from the parking lot.

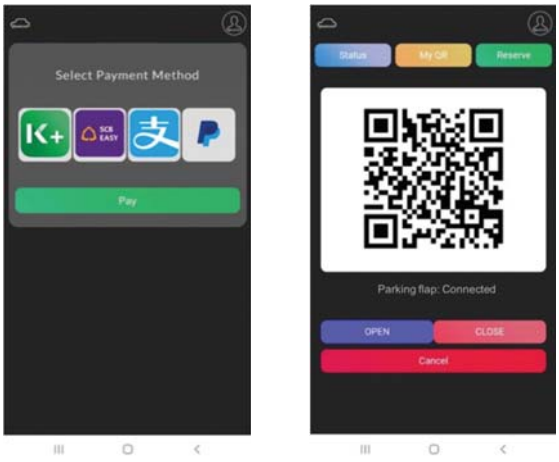

Fig. 7. ParkKing web application: payment page (left); and QR code page (right).

After scanning the code, the reservation status is updated to "reserved". The reservation page appears, which summarizes the information about the reserved vehicle, the selected parking building, the floor number and the parking spot ID assigned to the vehicle. See Fig. 8 (left). Fig. 8 (right) shows the user profile page. The history section of the user information page (not shown) contains user's parking history, i.e., the check-in and check-out times, vehicle(s) selected for parking, the parking fees, and receipts.

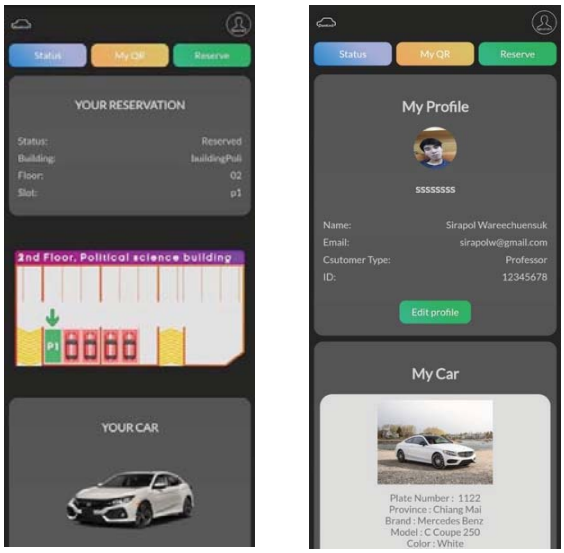

Fig. 8. ParkKing web application: reservation page (left); and user profile page (right).

C. *Software Quality Assurance (SQA) Tests and Results*

In this project, SQA was performed to provide the assurance of the requirement quality, design quality, code quality and quality control effectiveness within the software. The team performed unit testing, integration testing, user interface testing, as well as user and use case testing.

For the unit testing, each component was separately tested. For example, for the user registration page, the team tried to register with invalid inputs, e.g., invalid email address (without @ sign), invalid ID (less/more number of digits than student ID or national ID), etc. All invalid log-in attempts were caught successfully.

For the integration testing, all individual software components were combined and tested. The team checked against different scenarios. One was to see if the software could prevent the access without logging in. This was done by implementing the middleware. The test result indicates that the user was redirected to the login page. The team also checked to see if the system could prevent a user from obtaining a QR code before reserving the parking spot. In this case, the user was redirected to the reservation page if the user did not make a reservation. Overall, results of all integration testing were satisfactory.

In addition, all pages of the web application were thoroughly tested. The system was verified to meet all functionality requirements.

Regarding the non-functionality requirements, Table II summarizes the team's performance objectives, along with success criteria and test results.

Firstly, ParkKing web application were tested to work on any web browser, so the number of web browsers that are compatible with the web-application is limitless. The QR code scanning was very responsive. The parking flap was able to open and close simultaneously with the embedded MQTT commands, and a responding time was found to be less than a second, which was satisfactory. Internet connection plays a big role in system performance.

TABLE II. PERFORMANCE OBJECTIVES AND TEST RESULTS

Performance Objective	*Metric*	*Success Criteria*	*Test Results*
Operating on web browser	Number of web browsers that the system can support	>3	PASS
Response time when scanning QR code to open/close parking flap	Number of seconds that the system reacts to the QR code	<2 seconds	PASS
Response time to automatically backup users' data	Number of second to store the users' data into the database	<2 seconds	PASS
Response time to change the reservation status	Number of seconds required to process the check-in/check-out after scanning the QR code	<2 seconds	PASS
Response time to enable and disable the parking flap	Number of seconds required to open/close the parking flap after pressing the button	<3 seconds	PASS
Allow users to understand the website interface quickly	Time required for the users to completely understand the interface in all aspects	<1-2 minutes	PASS
Allow users to complete each task with a few steps	Number of steps needed to make a parking spot reservation from the beginning to the end	<3 steps	PASS

Since response times of all the required functions rely heavily on the Internet, so if the connection is strong and reliable, all non-functional requirements can meet the proposed criteria.

With respect to the user interface, the team conducted UI tests to gauge users' understanding of the UI. The users took less than five minutes to figure out how the whole system works and was able to use the application without any guide or user manual. Alert messages appear to inform the user of which page or button to use in order to move on to the next step of the activity flow.

V. Conclusion

Park King was developed by a group of engineering students who see an inefficiency in the current parking system at Chulalongkorn University. Park King is a system that potentially provides seamless experience for reserving parking spots in a university campus. It comprises a web-based application that works seamlessly with a set of IoT sensors and controllers to allow users to check for parking spot availability, make a reservation for a parking spot in advance, and make an online payment. The entire Park King prototype, comprising both hardware and software, was developed. The sensor module was deployed at a real-world parking lot to identify any implementation problem. Software quality assurance tests were conducted to test the seamless integration of both the Park King hardware prototype and its web-based application. Test results were satisfactory in terms of system functionality and performance requirements.

Overall, Park King has been developed as a generic parking application so that it can be implemented at any university campus, transforming and networking a traditional parking system into a smart system by integrating IoT technologies. Possible future work includes adding data analytic features to analyze the parking datasets, for example, analyzing prime time for parking, comparing utilization of each parking building on campus, and identifying possible revenue stream based on time-varying parking fees.

References

[1] Aida Engineering LTD., "Self-service Car Parking Control System" [Online]. Available: http://www.aida-eng.co.jp/en/parking/park/. Retrieved: January 2019.

[2] Daimler AG, "Innovative Parking Solutions. Convenient, customer-oriented and efficient" [Online]. Available: https://www.daimler.com/innovation/parking.html. Retrieved: January 2019.

[3] Advantech Co., Ltd., "Parking Service System" [Online]. Available: https://www.advantech.com/iretail-hospitality/solutions/detail/parking-service-system. Retrieved: January 2019.

[4] Siemens, "Intelligent parking solutions" [Online]. Available: https://new.siemens.com/global/en/products/mobility/road-solutions/parking-solutions/intelligent-parking-solutions.html. Retrieved: January 2019.

[5] Smart Parking Ltd., "Smart Parking: Car Parking Solution" [Online]. Available: https://www.smartparking.com. Retrieved: January 2019.

[6] The Parking Spot, "Airport Parking Made Easy" [Online]. Available: https://www.theparkingspot.com. Retrieved: January 2019.

[7] Shin Woo UBiCos., "Automatic Car Parking System" [Online]. Available: http://siamind.velaeasy.com. Retrieved: January 2019.

[8] A. Khanna and R. Anand, "IoT based smart parking system," *2016 International Conference on Internet of Things and Applications (IOTA)*, Pune, 2016, pp. 266-270.

[9] Smart Parking System, "Smart Parking System" [Online]. Available http://smartparkingsystems.com/en/. Retrieved: February 2019.

[10] Cloudparc, "Smart Parking through Technology" [Online]. Available http://cloudparc.com/a/. Retrieved: February 2019.