# Smart parking monitoring software application

## Aplicație software inteligentă de monitorizare a parcării

Cristina Gabriela Sărăcin[1], Florin Daniel Streche[1]

[1]Universitatea Polithnica din București, România
Splaiul Independenței Nr.313
*E-mail: cristina.saracin@upb.ro, daniel.streche@yahoo.com*

DOI: 10.37789/rjce.2022.13.4.1

**Abstract.**– *The paper presents the software application designed to monitor the car parking occupancy rate. The real situations that can be encountered in a parking lot are simulated with the help of a scale model made based on the Raspberry Pi 3 development board. This board connects the existing hardware components on the scale model, the created software applications, and the Cloud Firebase database. The services offered by Firebase use the real-time database and the authentication service. The first is used to store data received from parking sensors and the second to register and authenticate car parking users. The user interface is made within the Android Studio programming environment. The developed applications are the followings: new user account registration, user authentication, parking space reservation, parking entrance, parking space occupation and parking exit. The smart car parking collects real-time information which contributes to diminish the time spent in traffic, decreases pollution, and why not, significantly reduces of the stress of the drivers.*

**Keywords:** smart car parking, Cloud Firebase, Android Studio, Raspberry Pi 3 Board

## 1. Introduction

The study conducted by the Statista company shows that, from 2006 to 2015, the number of personal cars increased from 679 to 947 million vehicles and, the number of commercial transport vehicles increased from 247 to 335 million units [1]. In 2021 it is globally estimated a total of 1.4 billion cars, which means an increase of over 100 million compared to 2015.

The study conducted by the company Inrix shows that American drivers spend an average of 17 hours per year to find a parking space and for this they annual pay $345. Almost two-thirds of Americas drivers (61%) reported they felt stressed trying to find a parking spot, nearly half (42%) missed an appointment, one-in-three (34%) abandoned a trip due to parking problems and one-quarter (23%) experienced road rage [2].

It is also estimated that the urban population will significantly increase in the coming decades, from the current 3.9 billion (54% of the total global population) to 6.3 billion by 2050. These increases implicitly lead to a real explosion in the number of

vehicles, which dramatically amplifies the time spent in traffic and a much higher level of pollution than at present time.

All the studies presented lead to the widespread implementation of the notion of smart city. The smart city is the city concept that uses the digital, information and communications technologies for support of human lives quality in a city. For realization of concept objectives, it is necessary so the city might the high quality and safe information and communicational system [3]. The concept of smart city entails the fluidization of traffic and the emergence of smart parking. Smart Cities are a current emerging trend that aims to effectively and smartly automate the monitoring, access and usage of the infrastructure underpinning the major services offered to the citizens. Advanced wireless sensing technologies, machine learning techniques, 5G networks and big data analytics tools are among the main enablers of secure and effective management of the often-limited city resources. If its usage is properly optimized such resources can significantly assist in revolutionizing the citizens life through improved education, better and more affordable healthcare and cheaper, greener and more comfortable transport [4]. A "smart city" is not a particularly new concept. Urban planners and city governments have been implementing automated or digital solutions to provide improved services for years. A "smart city" is about more than the sum of the automation and digital endeavours [5].

A novel parking management system consists of hardware and software modules. On the hardware side, Raspberry Pi device is used along with few other parts like sensors and cameras. On the software side, python modules are implemented with efficient data-management. For example, it can be modelled different possible actions of the drivers and ensured that the proposed parking system performs perfectly. In car parking case, three core situations can be outlined. First one is when drivers park correctly, the second one has error-correction, and the third one requires error correction involving multiple cars. This system can be used in any large parking structure in the modern smart cities [6].

This paper intends to develop a smart car parking model and implement software applications to monitor it.

## 2. Smart parking monitoring simulation model

The hardware components used to simulate smart parking monitoring are as follows:

- development board Raspberry Pi 3 model B +;
- two Digital Sharp GP2Y0D810Z0F distance sensors;
- Raspberry Pi v2.1 video camera;
- barrier's servomotor;
- conductive elements and breadboard.

The camera is used to signal the entry of a car into the parking area and at the same time to read the registration number. It also commands the actuator that closes or opens the barrier. The servomotor is SG90 type with supply voltage of 4.8V with rotational speed of 0.12s/60˚ and 0.11s/60 ˚ for a voltage of 6V. Also the rotation angle is 180 degrees. It was connected to terminal 2 of the Raspberry Pi development board. The

integrated network of the development board is connected to an external Wi-Fi network through which the data is transmitted to the parking database. The operating system of the development board is Rasbian. This software runs from a micro-SD card inserted in the dedicated port located on the development board. The programming of sensors and other hardware components is done based on the Python programming language. The two Digital Sharp distance sensors provide information regarding occupancy of parking space. The two distance sensors use infrared technology to detect objects. The supply voltage of this sensor module is between 2.7 and 6.2V. The detection range of an object is between 2 and 10cm. The two sensors were connected to pins 1 and 4 of the Raspberry Pi development board.

The block diagram of the proposed intelligent parking system is shown in Figure 1. It can be seen the moments when, the GSM application and the program running on the Raspberry Pi card exchange information with the Firebase database. A JSON (JavaScript Object Notation) file is required to connect the hardware and software components to the cloud database. JSON is a standard format that helps to transmit data between a server and a web application. It is also an alternative to XML [12]. The JSON file was downloaded after the Firebase cloud database was created and added to the mobile application file package and Raspberry Pi development board memory.

At the same time, it can be observed the steps the driver must go through when he enters into the parking area, as well as the way hardware components operate.

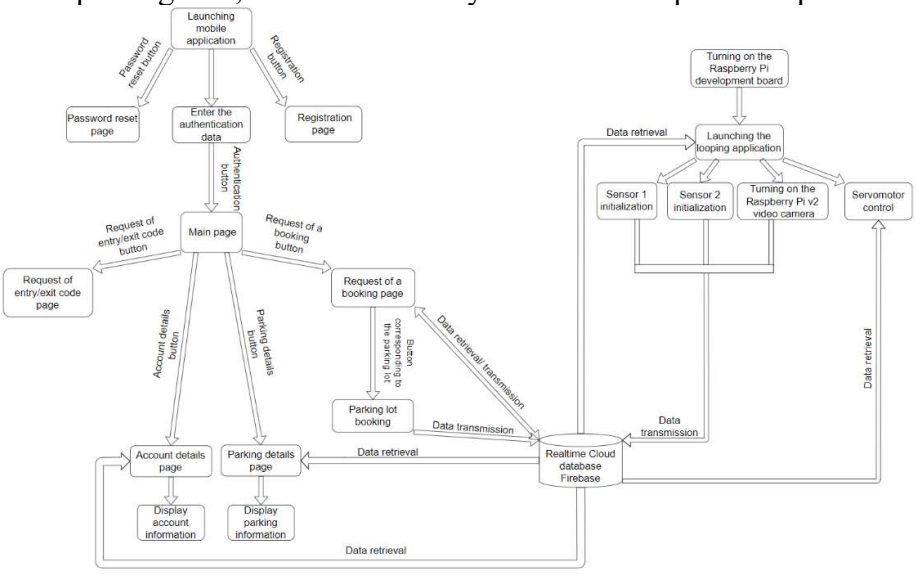



Fig. 1 Block diagram of smart parking system

As it can be seen in figure 1, the program runs on the Raspberry Pi, it takes the information from the distance sensors and the Raspberry Pi v2 camcorder and, transmits it to the database. If the parking space is not occupied, the sensor is in standby, and the program sends the "Free" information to the database. If the parking space is occupied, the sensor is activated, and the program sends the "Busy" information to the database. The camcorder transmits the image to the Raspberry Pi device, and with the help of the program, this processed info enables the scanning of the QR code displayed on the

driver's mobile phone. When the QR code is recognized in the database, the servomotor commands the lifting of the access barrier of the entry or exit smart parking. In figure 1 is also shown the bidirectional transmission between the application installed on the mobile phone and the database of the smart parking.

## 3. Smart car parking monitoring software application

The developed model of monitoring parking spaces consists of the following elements and subsystems:

- entrance/exit system in/from the parking lot;
- system for taking over the data from the parking places;
- GSM interface, consisting of the following pages:
    - New user registration;
    - User authentication;
    - Password reset;
    - Main program;
    - Parking details;
    - Account details;
    - Request a reservation;
    - Requests a code for entering/exiting the car parking.
- cloud database for data storage;
- payment monitoring method.

The software application is implemented in the Android Studio program and is dedicated to cell phones that use the Android operating system [7].

### *3.1. New user registration*

The driver must register if using the car parking for the first time. He can create a new account by pressing the "REGISTER" button in the GSM application running on his phone. The registration page contains the following details:

- name and surname;
- valid e-mail address;
- password;
- phone number;
- registration number of the car.

After completing all the fields, the user is entered in the Firebase authentication service (Fig. 2.) and in the database (Fig. 3).

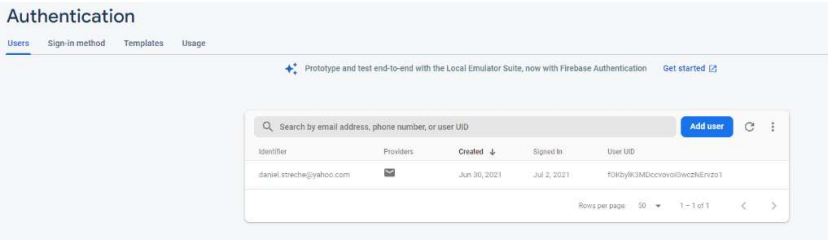



Fig. 2 Entering new user to the Firebase authentication service

At this time a verification email is sent to the driver's email address (Fig. 4). The code sequence required to register a new user is shown in Figure 3. After registration, if all data has been entered correctly, the app will go to the "Login" page. If the user did not fill in the info required for registration correctly, the error message “Registration failed” will appear. An example of an error, it is when the driver wants to register an account with an e-mail address already existing in the database.

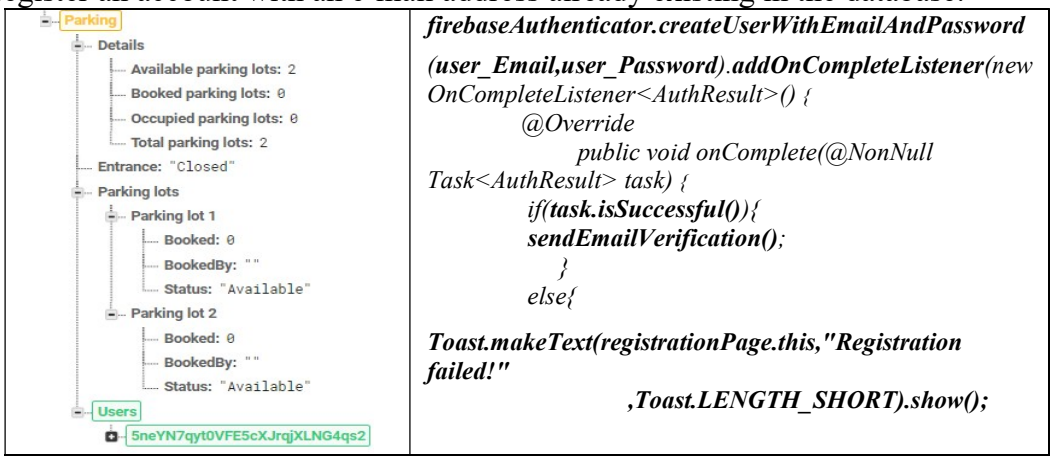



Fig. 3 Entering new user in the Firebase database

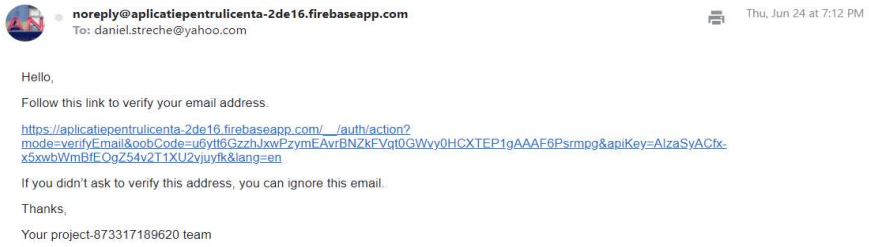
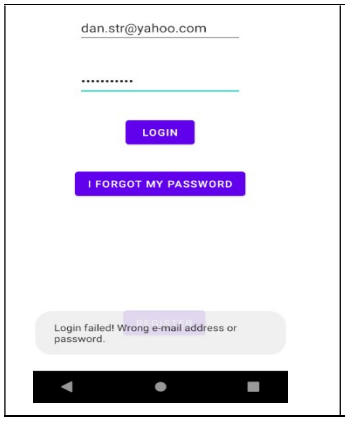



Fig. 4 Email address validation

### *3.2. User authentication*

After entering and verifying the account, the user logs in by entering the email address and password in the corresponding fields on the Login page.

If the e-mail or password is incorrect, the error message "Login failed!" appears (Fig. 5). Figure 5 also shows the code sequence required for authentication.




```
firebaseAuthenticator.signInWithEmailAndPassword(firstName,
userPassword).
addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task)
{
            if(task.isSuccessful()){
               emailVerification();
               finish();
        }
            else{
            Toast.makeText(MainActivity.this,"Login failed!
Wrong e-mail address or
password.",Toast.LENGTH_SHORT).show();
        }
```

Fig. 5 Failed authentication message

If the driver has forgotten the password, the "I FORGOT MY PASSWORD" button can be used. If the email completed by the user is correct (the one from which the account was created) then, a password reset email is sent to that account. Accessing the link (shown in fig. 6) allows to reset the password.

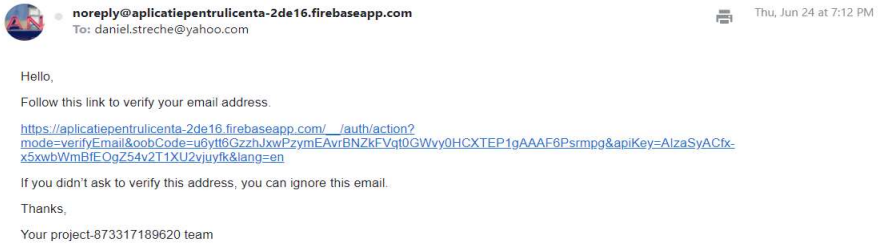



Fig. 6 Password reset email

### *3.3. Parking space reservation*

After the user's authentication, the main page of the graphical interface opens, providing access to the page with parking details, account details, plus a new booking request.

In the parking details page, the user is provided with data on the total number of parking spaces, vacancies, occupied, reserved and the status of the barrier at the parking area entrance.

On the reservation request page, the user can make a reservation of a certain parking lot by pressing the button corresponding to the desired parking space ("PARKING LOT 1" or "PARKING LOT 2"). This page also provides information on parking spaces, namely:

- if they are free, the colour of the buttons is green;
- if they are busy, the colour of the buttons is red;
- if they are reserved, the colour of the buttons is orange;

and, also information on who reserved the parking lot by identifying the registration number (Fig. 7).

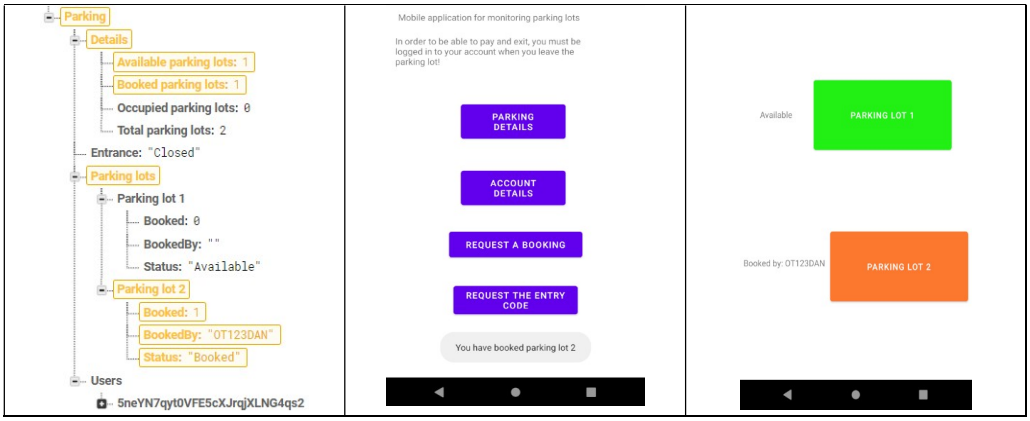



Fig. 7 Updating data in the database after booking Parking lot 2

To reserve a parking space, the button corresponding to the desired parking lot must be pressed. If the place is already reserved, an error message is displayed the following text: parking space "Occupied" or "Reserved".

### *3.4. Car parking entrance procedure*

The car parking entrance code can only be requested if the driver has first reserved a parking space; otherwise, an error message is received. The request of the parking entry code is made by pressing the “REQUEST THE ENTRY CODE” button (Fig. 8). The parking entry request page shows a QR Code image that is required to enter the parking area. The driver positions the phone display in front of the Raspberry Pi v2 digital camera and, after scanning the QR code, the barrier rises. After a certain pre-set time, the barrier is closed, the access to the parking area being blocked.

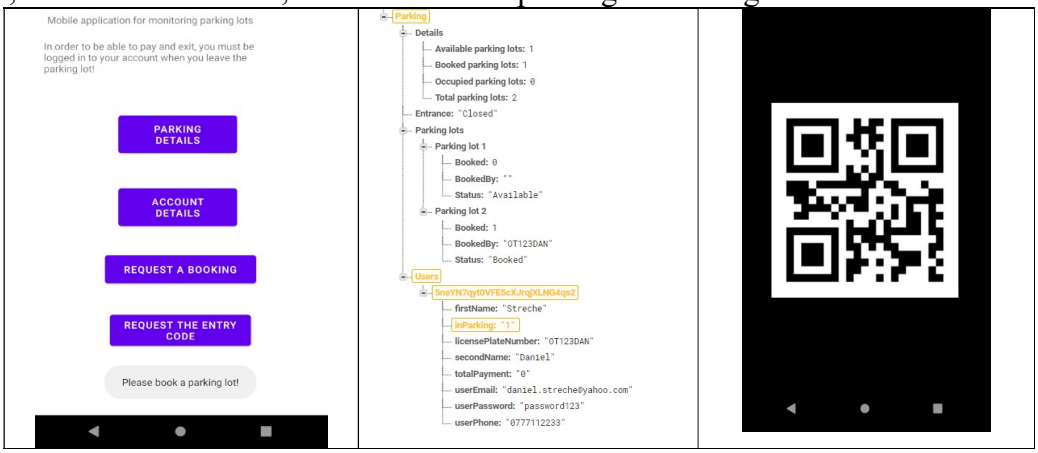



Fig. 8 Booking a parking lot and updating info in database at the request of the entry code

After booking a parking space and entering the parking area, the driver must go to the reserved parking lot to occupy it.

### *3.6. Occupancy of the parking space*

At this moment, the distance sensors related to the respective parking space update the database with the new occupancy of the parking lot (Fig. 9).

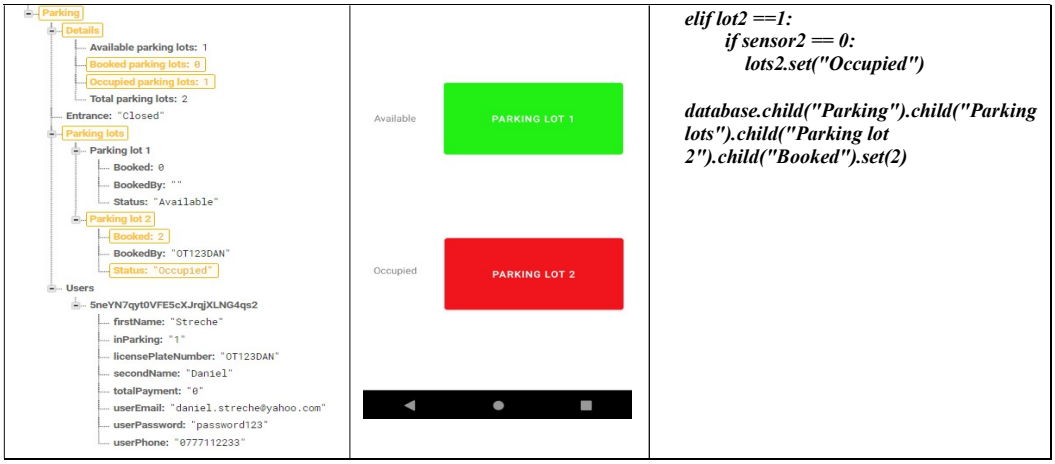



Fig. 9 Updating process in database after the parking lot 2 is occupied

When the mobile phone application retrieves this information from the database, the "REQUEST THE ENTRY CODE" button (Fig. 8), displayed on the cell phone screen, it is replaced by the "PAY AND REQUEST THE EXIT CODE" button (Fig. 10).

### *3.7. Release of the parking lot*

When the driver releases the parking lot, he must open the application on the cell phone and requests the payment and QR code. Otherwise, he has to contact the parking administrator. To remind the driver of the need to request the exit code, a warning message is displayed on the main page of the graphical interface (Fig. 10).

When the parking space is released and the user is authenticated, the payment monitoring method is completed and a message with the total payment amount is displayed. The payment method consists in taking two variables of time i.e. when the user occupied the parking place and the moment when he released it. The mobile application calculates the time between these two variables and multiplies it with the tariff charged by the parking owner.

At the same time, the amount is also recorded in the user's account, and it can be displayed on the account details page. The account details page provides data regarding the account of the authenticated user, such as his full name, the registration number of the car and the total payment for the number of hours he occupied the parking space.

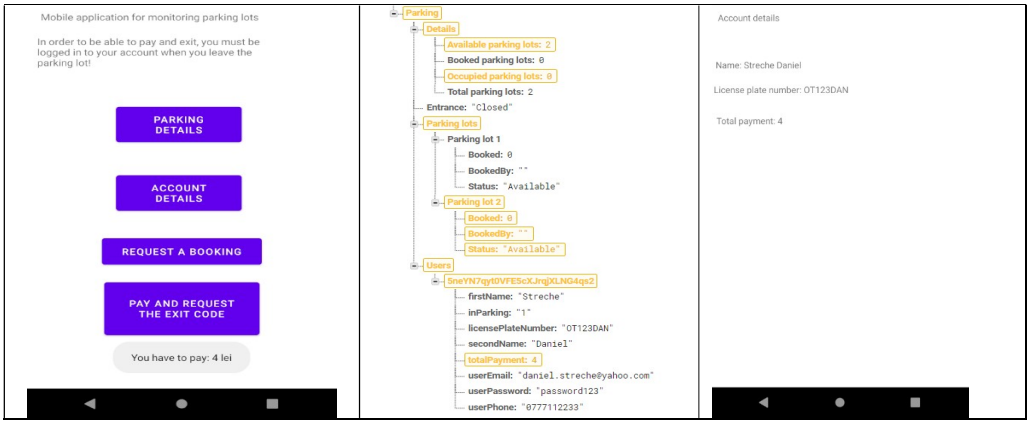



Fig. 10 Updating database when a parking space is released

### *3.7. Exit of the parking area*

To leave the car parking, the user must press the "PAY AND REQUEST THE EXIT CODE" button. This button can only be used if the driver has paid i.e., he has been authenticated when the parking space is cleared. By pressing this button, it will be opened the page that displays the QR code required to exit the parking area (Fig. 11).

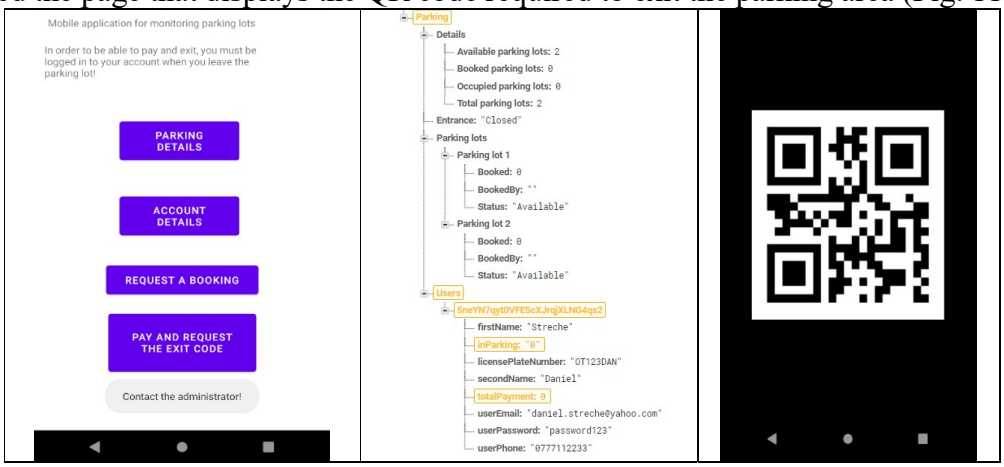



Fig. 11 Updating database when leaving the parking area

The user is allowed to exit from the parking area by positioning the QR code located on the mobile phone screen in front of the video camera located at the parking exit. When scanning the QR code, the exit barrier is lifted by the actuator, and it closes after the car has left. After scanning the QR code, the information encrypted in it is decrypted using the decode function from the pyzbar library and transmitted to the database via the Raspberry Pi development board. The information encrypted on QR code is “Open”.

## 4. Conclusion

In this paper, it is presented a smart parking model based on a Raspberry Pi 3 development board that simulates a real parking using the following components:

- two distance sensors used to check the occupancy and releasing of parking lot;

- a servomotor and a Raspberry Pi video camera used to create a parking entry/exit system.

The designated graphical interface includes and implements several functions that lead to an improved security in car parking, as follows:

- registration and authentication of a new user;
- registration of information regarding the parking status;
- information about user account;
- reservation of parking lot;
- payment method monitoring system;
- method of scanning the QR codes required to enter/exit the car parking.

The total cost of the system presented in this paper is about 100 euros. This is the cost to purchase the electric components.

Using this smart parking system, drivers can save time and money. The information that is provided in real time through the graphical user interface is beneficial for the driver, who can avoid traveling to the fully occupied park.

Also, in addition to information on the status of the car parking received thru the cell phone application, it can be provided information about hourly rate of a car parking.

This system can also help the environment and infrastructure in large cities by improving traffic and it can be implemented for single car parking or large network of car parking. Plus, by implementing a differentiated payment method, users may be charged with different rates depending on the time when the reserved space is occupied and vacated.

And undoubtedly, the smart car parking is the solution to the increasingly congested traffic in major cities and a real solution to reducing the pollution generated by the globally exponential increase of the number of vehicles.

## REFERENCES

[1] - Number of passenger cars and commercial vehicles in use worldwide from 2006 to 2015 Disponibil la: URL: https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/

[2]. - Searching for Parking Costs Americans $73 Billion a Year; Disponibil la URL: https://inrix.com/press-releases/parking-pain-us/

[3] – D. Prochazkova and J. Prochazka, "Smart cities and critical infrastructure," 2018 Smart City Symposium Prague (SCSP), 2018, pp. 1-6, doi: 10.1109/SCSP.2018.8402676.

[4] - R. Hammons and J. Myers, "Smart Cities," in IEEE Internet of Things Magazine, vol. 2, no. 2, pp. 8-9, June 2019, doi: 10.1109/MIOT.2019.8892761.

[5] P. Melnyk, S. Djahel and F. Nait-Abdesselam, "Towards a Smart Parking Management System for Smart Cities", 2019 IEEE International Smart Cities Conference (ISC2), 2019, pp. 542-546, doi: 10.1109/ISC246665.2019.9071740.

[6] – S. Das, A Novel Parking Management System, for Smart Cities, to save Fuel, Time, and Money, IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pag. 950-953, 2019.

[7] - C. Recoșeanu, A. S. Deaconu, A. I. Chirilă, V. Năvrăpescu, I. D. Deaconu, C. Ghiță, Remote monitoring of a permanent magnet synchronous motor using android devices, Scientific Bulletin, University POLITEHNICA Bucharest, Series C: Electrical Engineering, vol. 2, 2013.

[8] - S. N. Pandit, R. M. Krishna, R. Akash and M. Moharir, Cloud Based Smart Parking System for Smart Cities, International Conference on Smart Systems and Inventive Technology (ICSSIT), pag. 358-354, 2019.
[9] - Meet Android Studio – De pe developer.android.com, Disponibil la URL: https://developer.android.com/studio/intro
[10] – G. Yan, S. Olariu, M. Weigle, M. Abuelela, SmartParking: A Secure and Intelligent Parking System Using NOTICE, Conference Intelligent Transportation Systems, 2008. ITSC 2008.
[11] – Introduction to Firebase – Disponibil la URL: https://hackernoon.com/introduction-to-firebase-218a23186cd7
[12] – A. Ahad, Z. R. Khan, S. A. Ahmad – Intelligent Parking System, World Journal of Engineering and Technology, Vol.4 No.2, May 2016.
[13] N. B. A. Warif, M. I. S. S. Azman, N. -S. N. Ismail and M. A. Remli, "IoT-based Smart Parking System using Android Application," 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), 2020, pp. 1-6, doi: 10.1109/ETCCE51779.2020.9350914.
[14] M. Owayjan, B. Sleem, E. Saad and A. Maroun, "Parking management system using mobile application," 2017 Sensors Networks Smart and Emerging Technologies (SENSET), 2017, pp. 1-4, doi: 10.1109/SENSET.2017.8125048.