SMART SECURE LOCK

By
Buket Şentürk
Mehmet Ozan Güven
Safa Günay

Report Delivery Date: 05.05.2019

Report Version No: 1

A Thesis Report Submitted To The

Faculty Of Engineering in Partial Fulfilment Of The

Requirements For The Degree Of

Bachelor Of Science

Department: Computer Engineering

Thesis Advisor:

Yusuf Murat Erten

Izmir Institute Of Technology
Izmir, Turkey
May, 2019

I approve the thesis of Buket ŞENTÜRK, Mehmet Ozan Güven, Safa GÜNAY.

Date of Signature:

...

Yusuf Murat ERTEN

Department of Computer Engineering

CENG415 UNDERGRADUATE THESIS REPORT CONTENTS

1. THE DESCRIPTION OF WORK

Technology continues to change our traditional life very fast. Today, most people use their mobile phones to complete many of the daily tasks and these tasks are increasing day by day. The things that people carry with them are getting overloaded and mobile phones have taken on the functionalities of these things. For instance, today we use NFC technology to pay for our shopping instead of credit cards. For this reason, we decided to implement the this project.

In this system, we aimed to create a structure that allows a dynamic QR code to be unlocked instead of a key. When we show this QR code on our mobile device to the camera located on the lock system, the door will be opened when necessary verification is provided by the cloud. Another important feature of our project is that you can authorize unlocking for another person. This authorization may be for a specified period of time or indefinitely. In addition, thanks to Blockchain technology, you can track who opened the lock. For security, a different QR is defined by the system every 60 seconds thanks to the dynamic QR code structure. This system prevents the QR code from being copied by unwanted persons and unlocking. This system is not only on the door, we can apply on all key structure such as windows, cars etc. So, this system has a flexible structure.

1.1 How does it work?

1.1.1 For User Registration :

- The user downloads the application from the market and registers to the system.
- Uses the registration number on the lock to define the lock on itself.
- Once the registration number is used, it cannot be used again.

1.1.2 For Opening the Door :

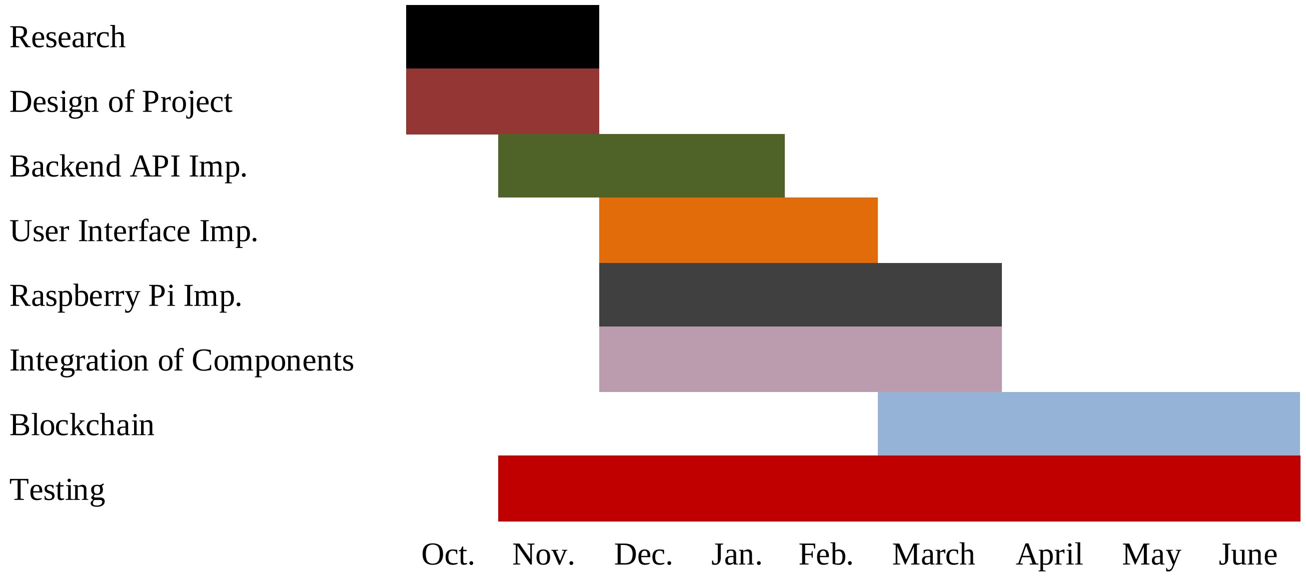
- The user displays the identified QR Code to the camera located on the lock.
- The camera sends the displayed QR code to the system via internet connection.
- The system checks the person via the QR code.
- If the person is authorized, he / she sends the system approval code to the lock.
- If the person is authorized, the door opens, if the person is not authorized, the red lamp is light.

1.1.3 For Giving Permission to Others :

- Firstly, the guest must be registered to the system.
- The guest must activate his/her email in advance.
- Then, the device owner chooses the device from his/her device list.
- The device owner writes the email of the guest and a description of this permission.

- Then the device owner specifies duration of the permission period to be given.
- Finally, confirms this transaction.

2. THE WORK PLAN



Work package title	**Type of activity[1]**	**Lead participant No[2]**	**Lead participant short name**	**Person-months[3]**	**Start month[4]**	**End month[6]**
Research	Research	3	S-O-B	2	October	November
Design of Project	Design	3	S-O-B	2	October	November
Backend API Impl.	Implementation	1	Safa	3	November	January
User Interface Impl.	Implementation	1	Ozan	3	December	February
Raspberry Pi Impl.	Implementation	1	Buket	4	December	March
Integration of Comp.	Integration	3	S-O-B	4	December	March
Blockchain	Implementation	3	S-O-B	4	March	June
Testing	Test	3	S-O-B	8	November	June
	TOTAL			9	October	June

Research

Description of work :

We have done some research about our project to determine technologies that would be used. Finally we have decided to use .NET platform for backend API, Flutter framework for User Interface and Rasperry Pi for lock mechanism.

Leaders : Safa – Buket - Ozan

Deliverables : 2 months

Design of Project

Description of work :

In this part, we have decided the database relation underlying the Backend API. With research, we have tried to design the most efficient system for our needs.

Leaders : Safa – Buket - Ozan

Deliverables : 2 Months

Backend Implementation

Description of work :

We implemented the application to communicate with the server and to manage the database. We used C# language for implementation.

Leaders : Safa

Deliverables : 3 Months

User Interface implementation

Description of work :

We designed an interface to communicate with the user. We used dart programming language to work with all platforms(IOS, Android).

Leaders : Ozan

Deliverables : 3 Months

Raspberry Pi Implementation

Description of work :

We integrated Camera and Lock mechanisms into the system and implemented these mechanisms to work.

Leaders : Buket

Deliverables : 4 Months

Integration of Components

Description of work :

We merged the server, user interface and raspberry Pi. And we made sure that these systems communicate.

Leaders : Safa – Buket - Ozan

Deliverables : 4 Months

****

Blockchain

Description of work :

We will do this system to ensure the security of the database. We will explore how to integrate the blockchain into the system.

Leaders : Safa – Buket - Ozan

Deliverables : 4 Months

Testing

Description of work :

From the first day we started implementation, we are testing our work. We will create test cases and try them.

As a result, we will discover the mistakes we have made and revise them.

Leaders : Safa – Buket - Ozan

Deliverables : 8 Months

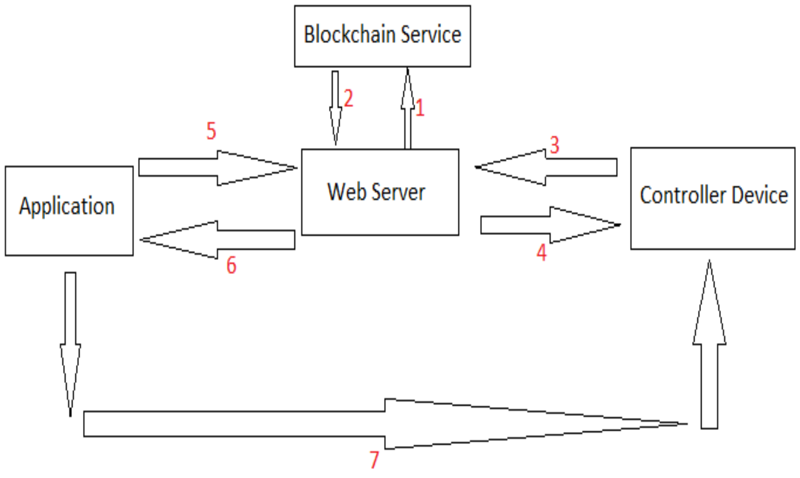
3 ANALYSIS MODEL

3.1 THE FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

- Firstly, users must download the mobile application.
- Then they are prompted to register to the system to login to the system with their created identity.
- When a user is registered. Other users can give him the permission to unlock their devices, or he can obtain his own physical locker device.
- When a user obtains a locker device, he can register it to the system via the special key written on the package of the device. The Mobile application will ask for the special key of the device when the user wants to register it as his own device.
- A device can be owned by only one user. Once the special key of the product is registered, it is not valid anymore.
- For a specific device, only the owner of the device can give other users the permission to unlock.
- If a user owns a device, he has the permission to unlock the device and give other users the permission to unlock the device.
- A user can see all other users who are permitted to unlock his device and remove any given permission at any time.
- When a user wants to unlock a device, he will generate a QR code using the mobile application and show it to any device that he wants to unlock. Then, if he is permitted, the physical lock that is controlled by the device will be unlocked.
- For a specific device, all unlock requests and responses together with the read QR codes will be saved in the blockchain service.
- If a user owns a device, he can see all activity logs of the device which are retrieved from the blockchain service.

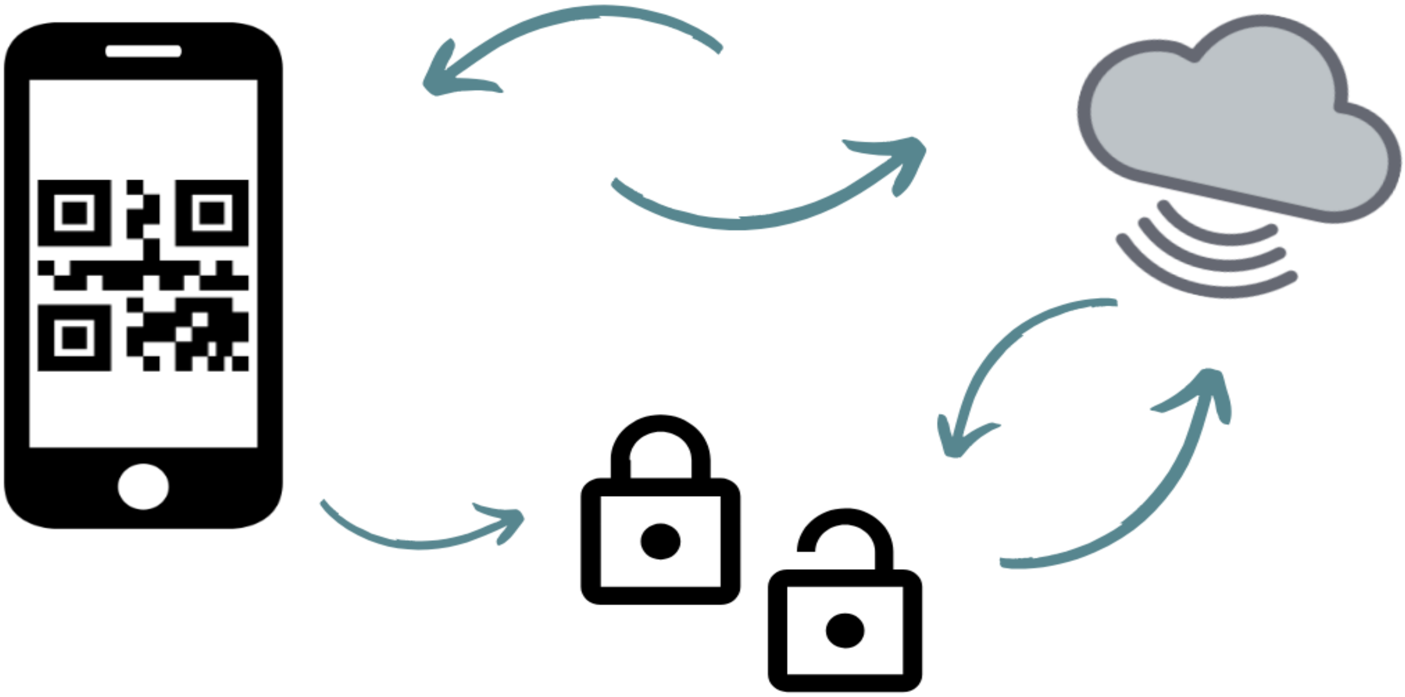
3.2 ANALYSIS MODEL

The model shows the relationships of 3 main component of the end the blockchain service of the entire solution. We can see the data transfers and their meanings in the model as described on the enumerated arrows.

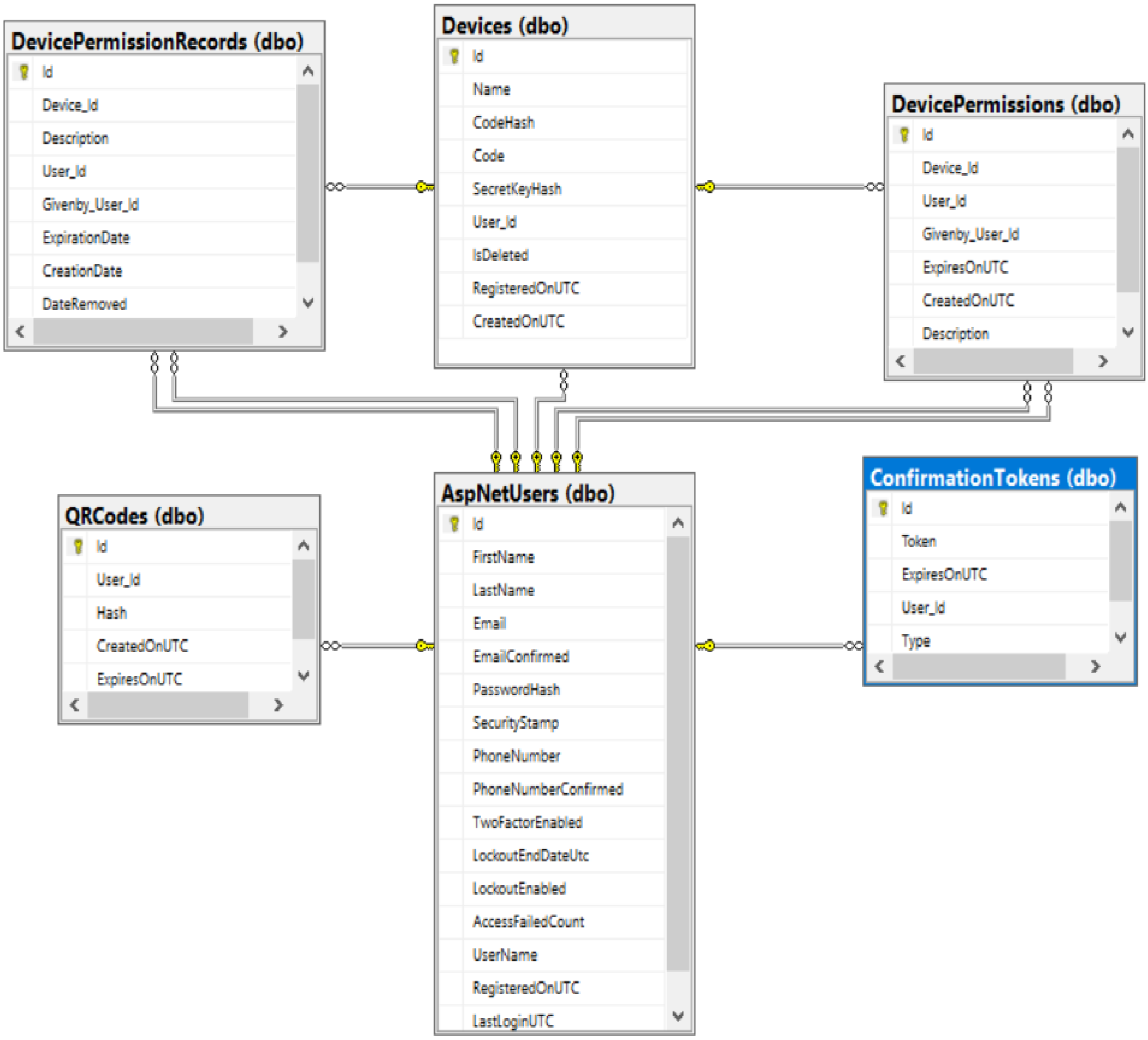


1: Retrieve device activity logs.
2: Get device activity logs of a specific device.
3: Send the read QR code to check the user is permitted to unlock.
4: OK or Denied Http response message of the last request.
5: Special device key, user login credentials, application client requests...
6: Responses of the requests from the application.
7: QR code.

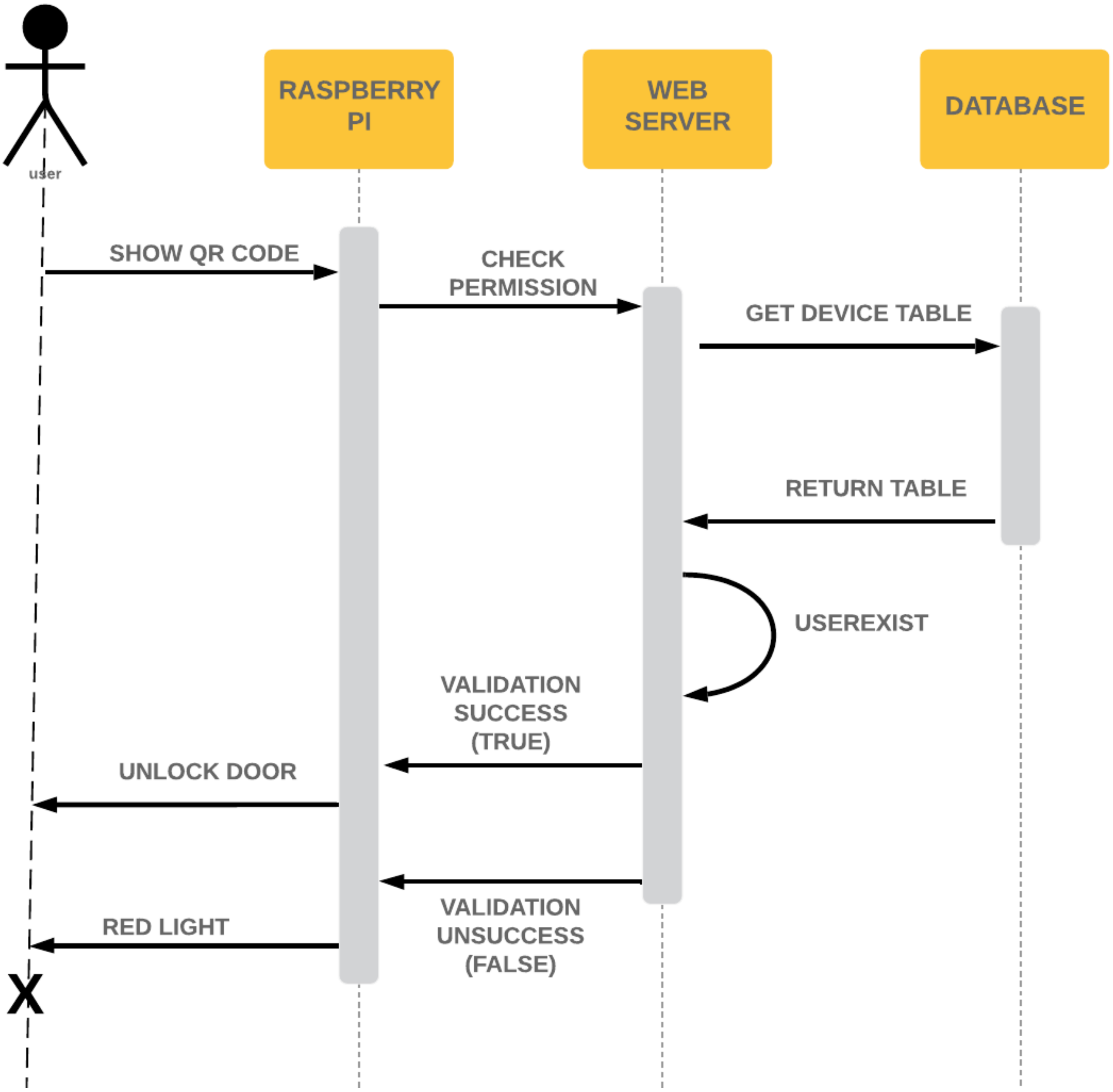
3.3 SCHEME OF THE SYSTEM



3.4 CLASS DIAGRAM

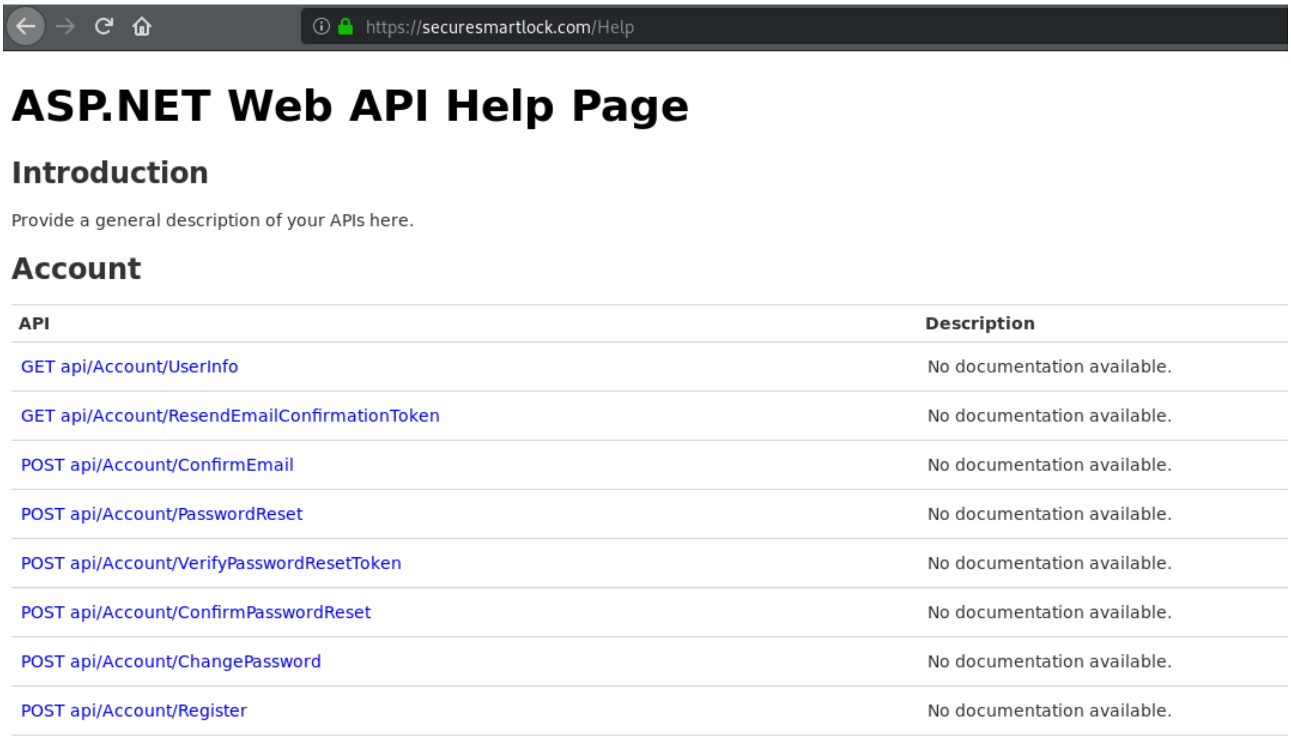


3.5 SEQUENCE DIAGRAM



3.6 BACKEND

3.6.1 ACCOUNTS ENDPOINT



ASP.NET Web API Help Page

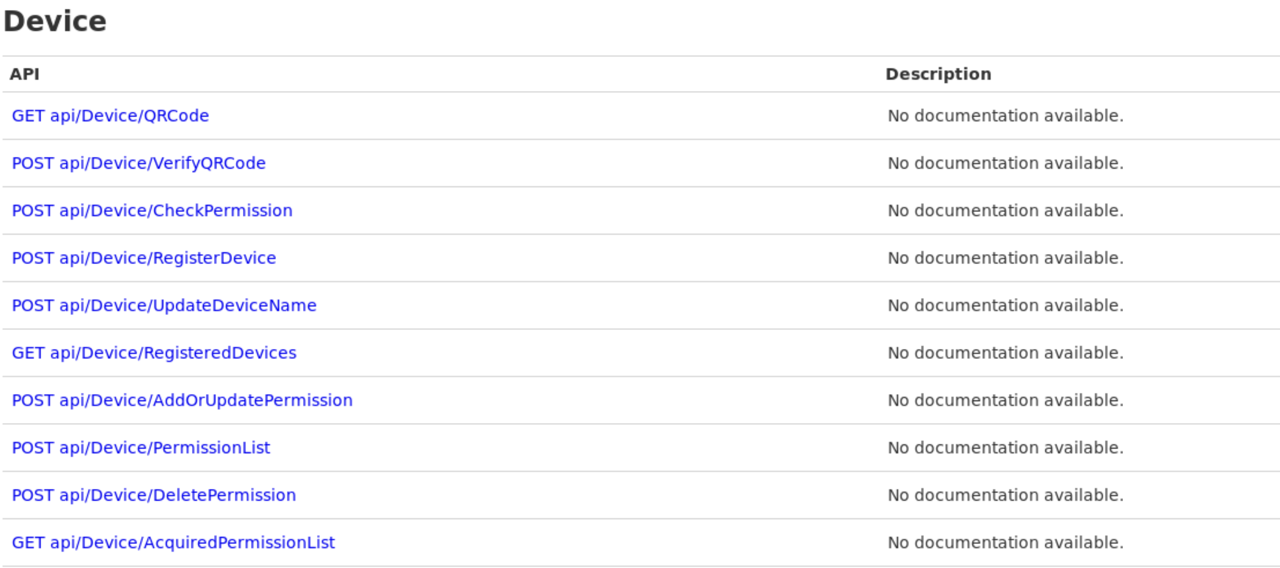
Introduction

Provide a general description of your APIs here.

Account

API	Description
GET api/Account/UserInfo	No documentation available.
GET api/Account/ResendEmailConfirmationToken	No documentation available.
POST api/Account/ConfirmEmail	No documentation available.
POST api/Account/PasswordReset	No documentation available.
POST api/Account/VerifyPasswordResetToken	No documentation available.
POST api/Account/ConfirmPasswordReset	No documentation available.
POST api/Account/ChangePassword	No documentation available.
POST api/Account/Register	No documentation available.

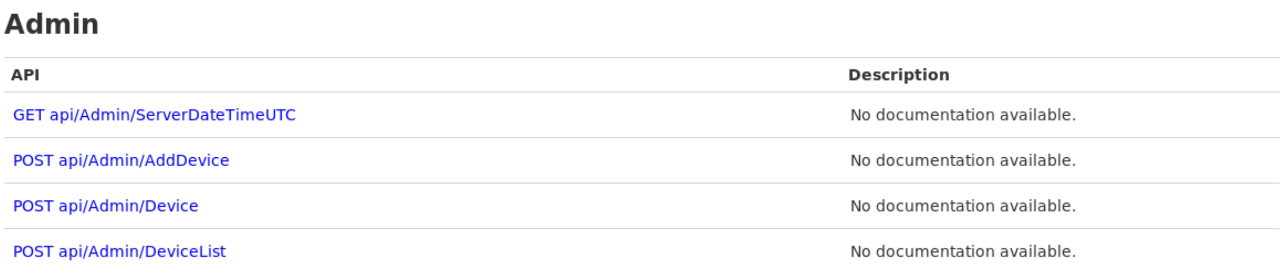
3.6.2 DEVICE ENDPOINT



Device

API	Description
GET api/Device/QRCode	No documentation available.
POST api/Device/VerifyQRCode	No documentation available.
POST api/Device/CheckPermission	No documentation available.
POST api/Device/RegisterDevice	No documentation available.
POST api/Device/UpdateDeviceName	No documentation available.
GET api/Device/RegisteredDevices	No documentation available.
POST api/Device/AddOrUpdatePermission	No documentation available.
POST api/Device/PermissionList	No documentation available.
POST api/Device/DeletePermission	No documentation available.
GET api/Device/AcquiredPermissionList	No documentation available.

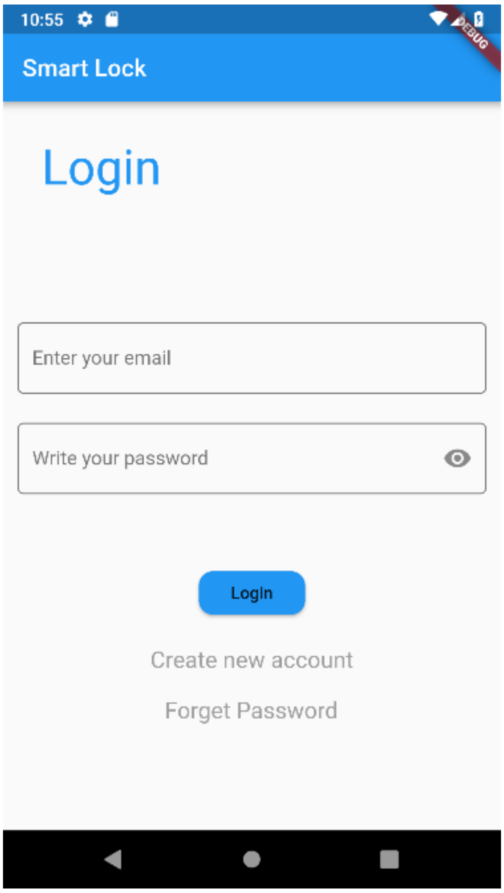
3.6.3 ADMIN ENDPOINT

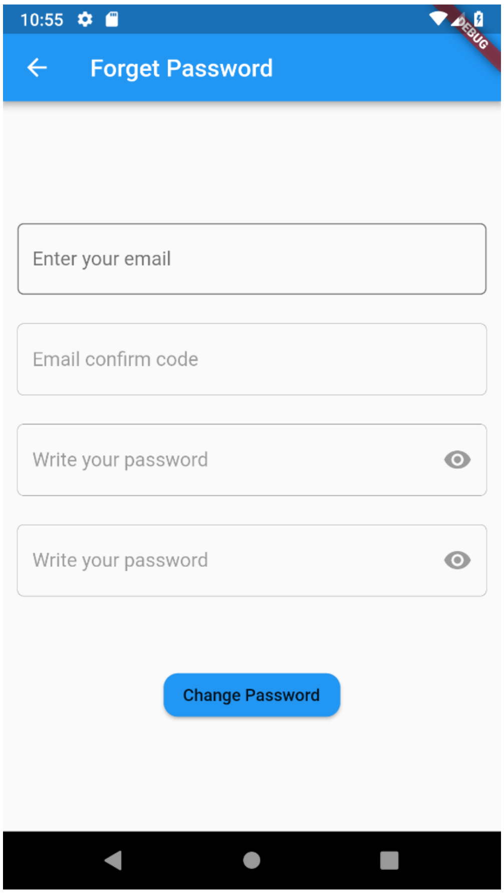


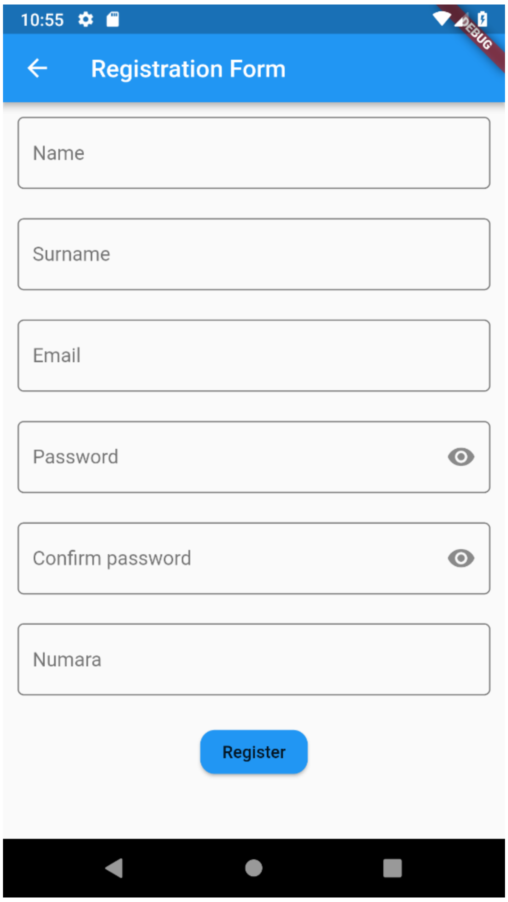
Admin

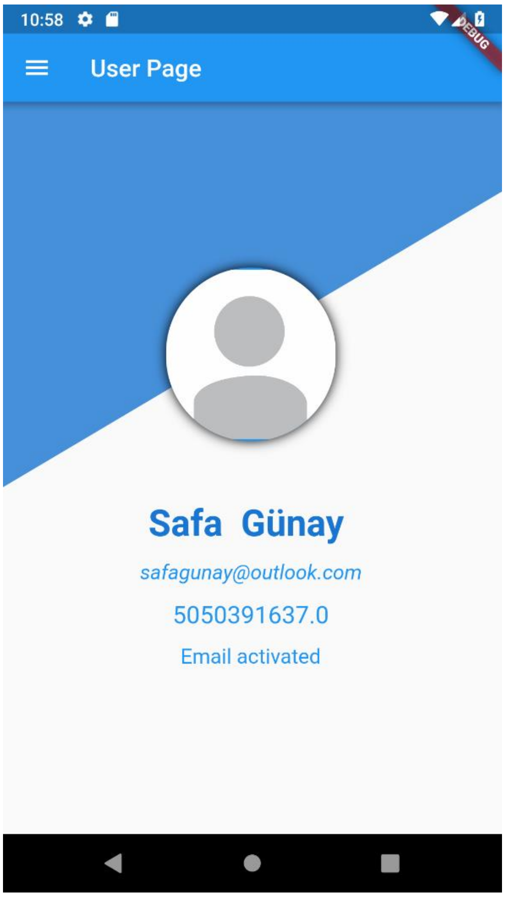
API	Description
GET api/Admin/ServerDateTimeUTC	No documentation available.
POST api/Admin/AddDevice	No documentation available.
POST api/Admin/Device	No documentation available.
POST api/Admin/DeviceList	No documentation available.

3.7 MOBIL APPLICATION







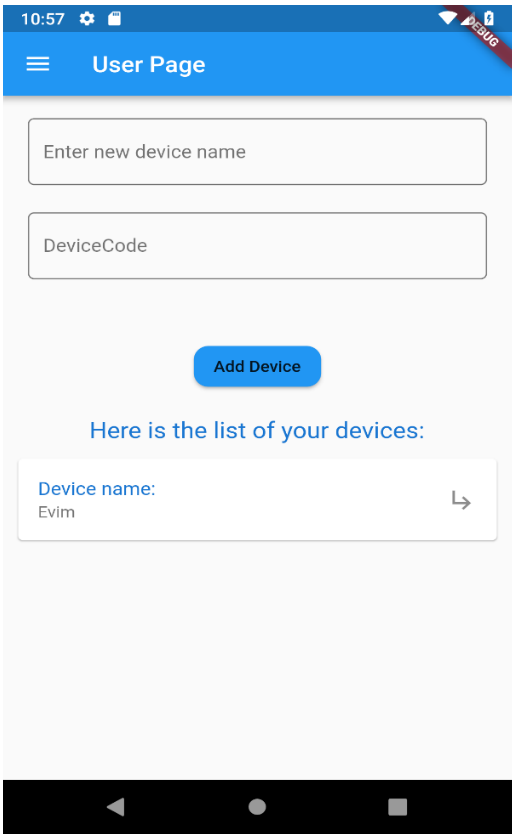


Safa Günay

safagunay@outlook.com

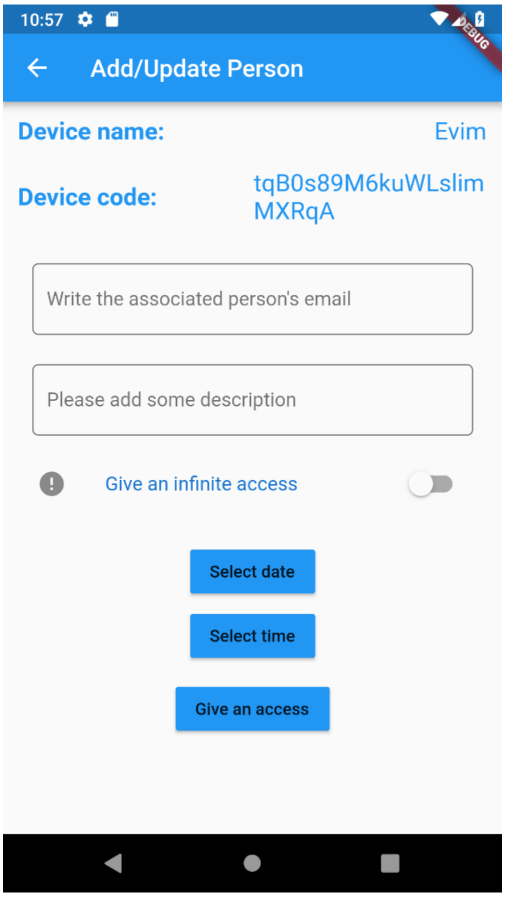
5050391637.0

Email activated



Here is the list of your devices:

Device name:
Evim



10:57

Add/Update Person

Device name: Evim

Device code: tqB0s89M6kuWLslimMXRqA

Write the associated person's email

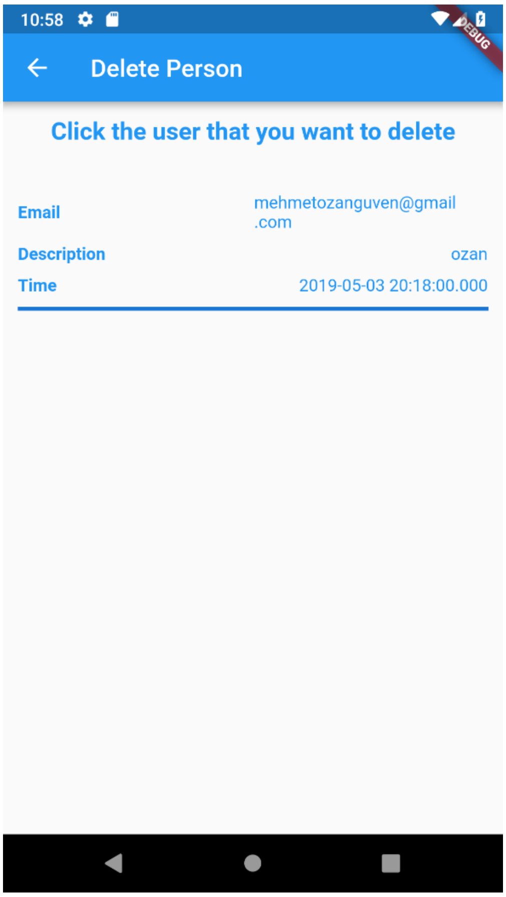
Please add some description

Give an infinite access

Select date

Select time

Give an access



10:58

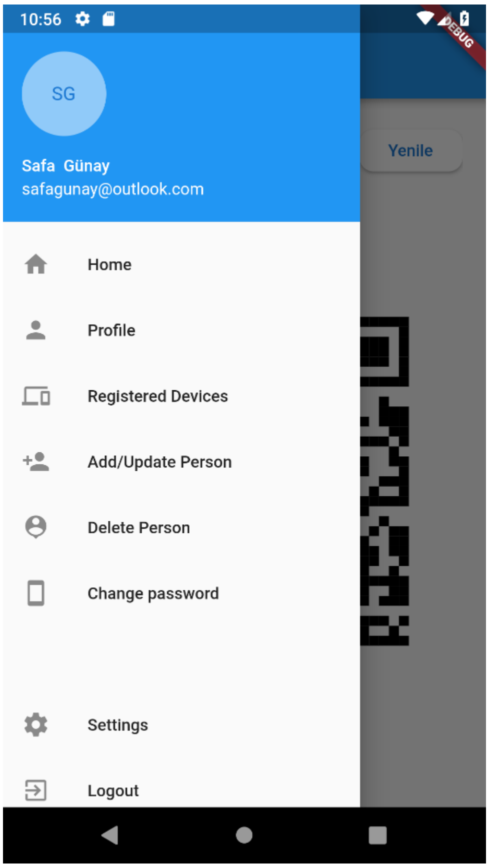
Delete Person

Click the user that you want to delete

Email mehmetozanguven@gmail.com

Description ozan

Time 2019-05-03 20:18:00.000



10:56

SG

Safa Günay
safagunay@outlook.com

Home

Profile

Registered Devices

Add/Update Person

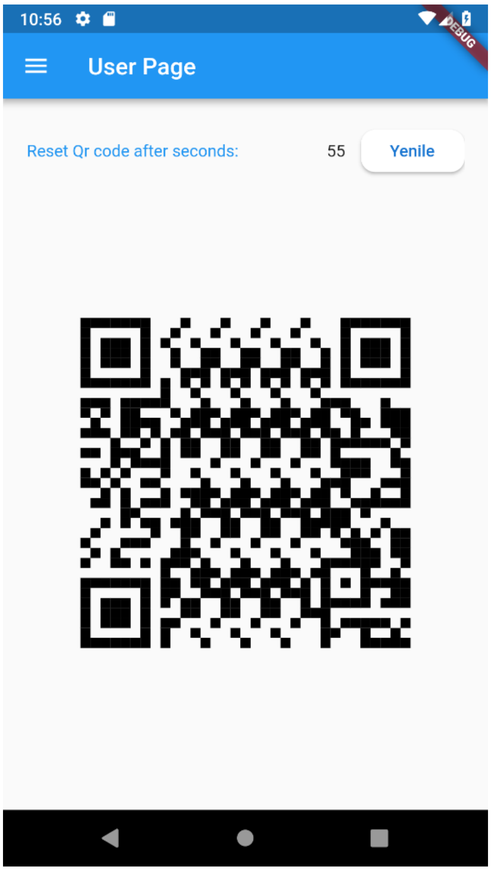
Delete Person

Change password

Settings

Logout

Yenile

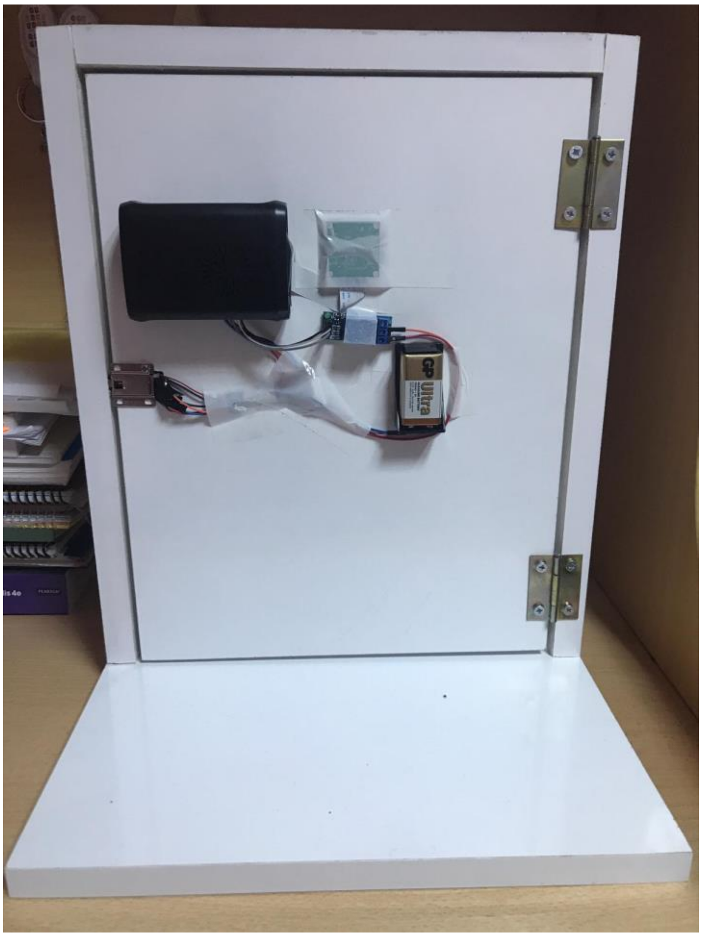


10:56

User Page

Reset Qr code after seconds: 55 Yenile

3.8 RASPBERRY PI



4. SOLUTION

4.1 WEB APPLICATION

The web application serves both the mobile app and the raspberry pi which controls the actual lock mechanism. It specifically serves them with JSON data. We used "Microsoft .Net Web Api 2" project template project to build this application. This project template bundles several .Net Framework Libraries including ASP.NET Identity system which handles user management operations like registration, bearer token generation and token authentication.

4.1.1 How it works?

The web app accepts some data from a client application (app running in raspberry pi or mobile phone) via http protocol, validates the data by checking whether it is in an acceptable format, and if it is, it performs some controls and operations on that data. For example, it queries some information with the given input by the client app in the database. Finally, according to that query results, it returns an http response message to the client app in "JSON" data format.

4.1.2 Security Features

QRCode expiration mechanism: Once a QRCode is generated, it is only for 60 seconds, after that the door unlock attempts with that QRCode won't be accepted. For the generation of QRCode we used "Globally Unique Identifiers (GUID)".

Bearer Token Authentication: Once the client app provides web app with a user's security credentials (username and password) it is given a special string token after which the client app embeds that token to each subsequent http request as an http header for web application to identify user.

Hashing the Critical Data: The web app hashes some critical data before it stores the data into the database. For example; user passwords, secret keys between raspberry pi and web app and QRCodes are hashed for security concerns.

APIKey Message Handler: If anyone could be able to cause the actual code in the web app to run by sending random requests, the web app would be vulnerable to attacks to overwhelm the host machine which runs the web app. To prevent this, message handlers check incoming http request whether they have the correct APIKey. If they don't, they are eliminated before they are routed to endpoint procedures.

SSL (Secure Sockets Layer): The SSL protocol runs in the transport layer which is independent of the application layer protocols, so we were easily able to change from http to https without making any changes to the application code. The reason we enabled this protocol between the client and server is to prevent anyone who could listen the communication between the server and mobile phone or raspberry pi to steal information.

4.2 MOBILE CLIENT APPLICATION

The mobile application serves end-user to communicate with raspberry pi and our cloud solution. The main purpose of the mobile application is to unlock the door with authorization.

4.2.1 Technology

At the beginning of the project mobile application was being written with Java programming language. Then, we have decided to change it because we wanted to work with all platforms (IOS&Android). That's why we moved the new framework called Flutter (created by Google) which uses Dart programming language.

4.2.2 How It Works?

When user opens the application, he/she will see the login page with login buttons, new account and forget password. If user forgets his/her password or wants to create new account, can use the "forget password" or "new account" clickable areas. After the user enters to our system, mobile application will connect to the cloud to get the timer for QR code and QR code itself. Then, user will see the timer at the top and the QR code.

After successful login, there are also sections for profile, home, add new device, update/delete person, change password, save login information for next time.

If this is the first login after creating an account, the user will also see the section "Email Confirmation". This is the section where the user will write email confirmation token to validate his/her mail. If user avoids to do that, he/she can not unlock the door. After entering the validation code, application will logout the user and requests the user to login again.

In Home section, user will see timer and the QR Code.

In Profile section, user will see some information about him/herself.

In Registered Devices section, user will see the devices that are accessible and also the user will see the section where he/she can add new devices.

In Add/Update Person section, user will allow someone to access his/her device or he may update the existing persons. For example, user can postpone the deadline for someone's access.

In Delete Person section, user will be able to delete a person whom she/he does not want to have access.

In Change Password section, user will be able to change password

In Settings section, user will be able to save login information to easy access for next time.

In Logout section, user will be able to log out.

4.3 RASPBERRY PI CONTROLLER

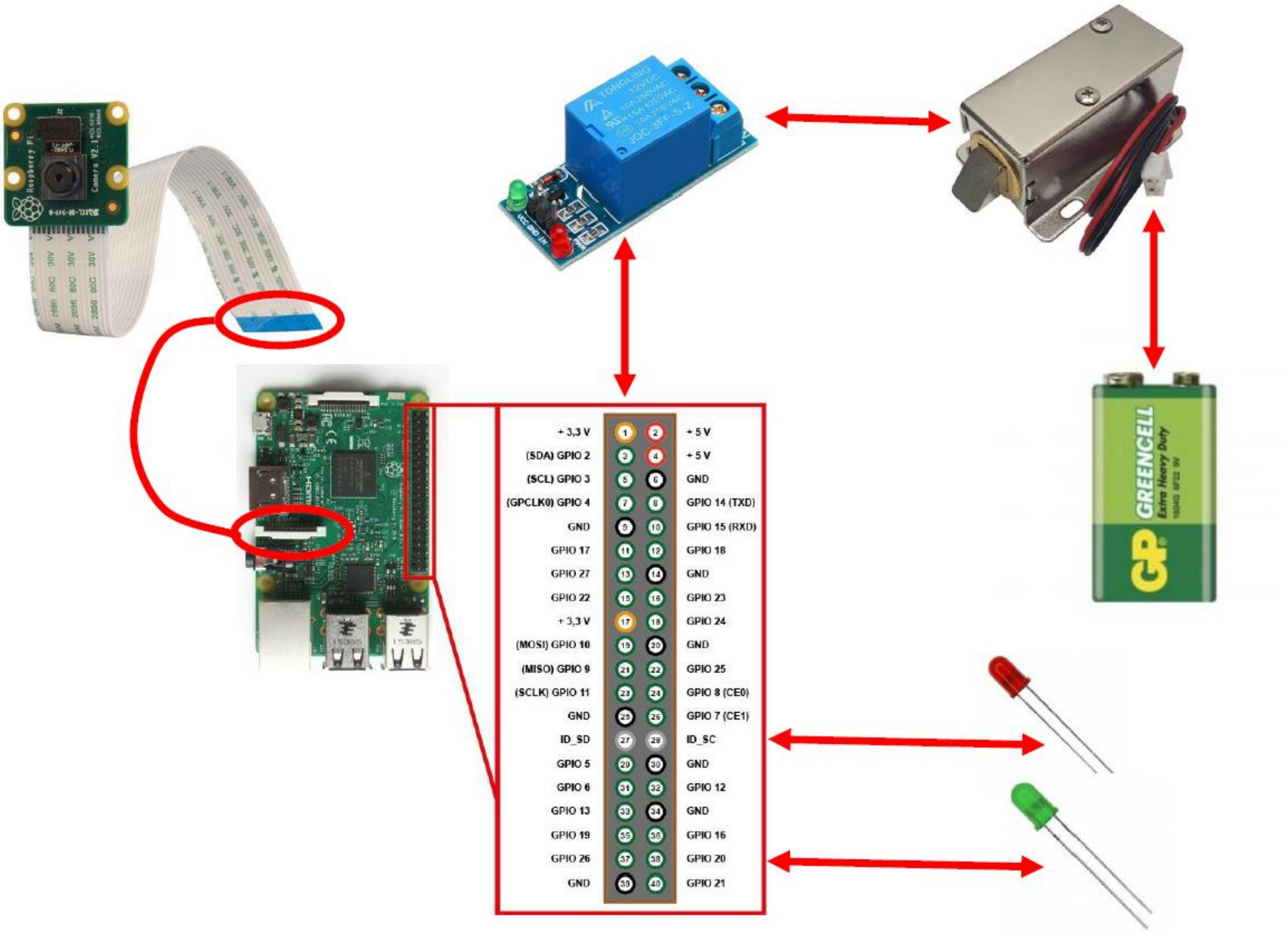
The raspberry pi allows the lock and camera system to be integrated. The camera scans the QR code that the user shows through the mobile application. It sends the scanned QRCode to the cloud.

According to the result from the cloud:

- The door will open and green lamp will light if the user is defined for the lock.
- The red light will light if not defined for the lock.

4.3.1 Technology

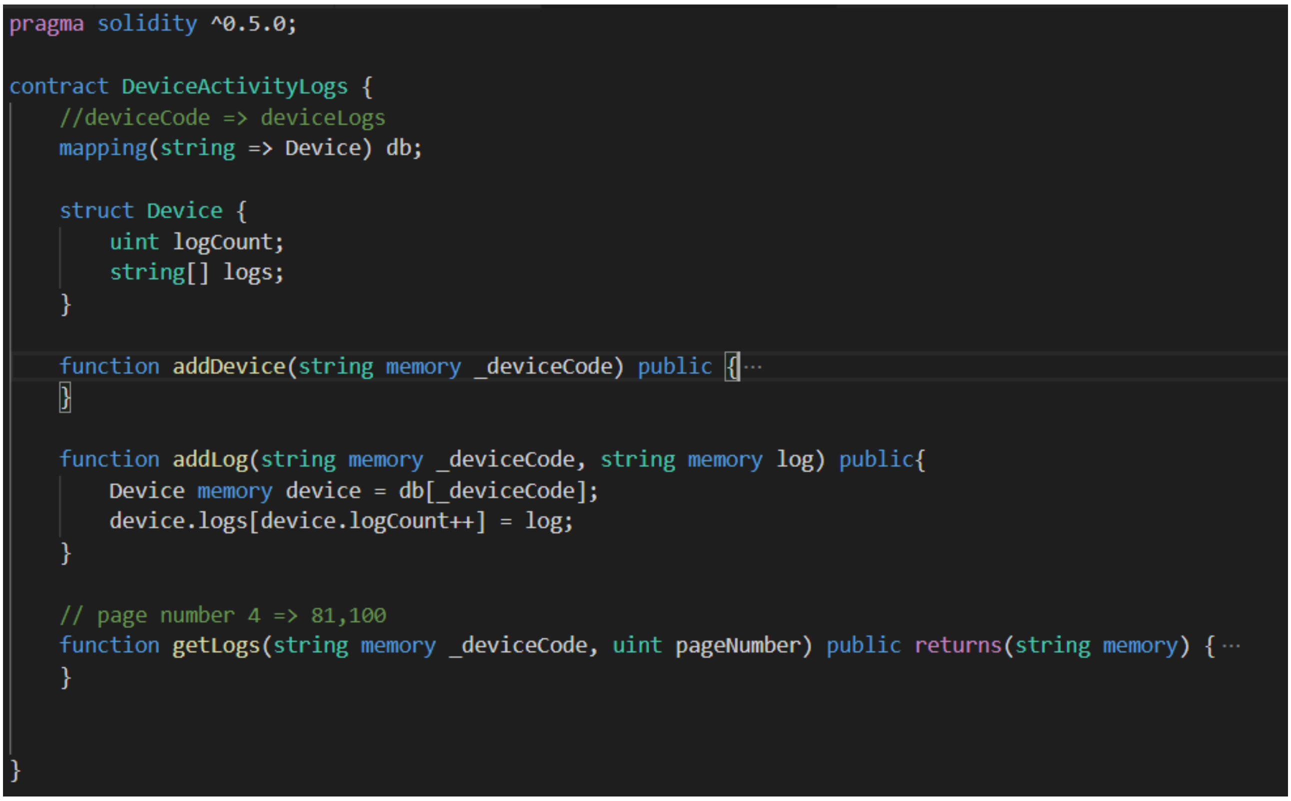
- Python language was used to program the raspberry pi system.
- Solenoid lock was used for the lock.
- The camera module was used to read the QR code.
- The relay was placed between the raspberry pi and the lock to ensure the required current for the lock to work.
- The required voltage for the lock operation is provided via 9V battery.



+ 3,3 V	1	2	+ 5 V
(SDA) GPIO 2	3	4	+ 5 V
(SCL) GPIO 3	5	6	GND
(GPCLK0) GPIO 4	7	8	GPIO 14 (TXD)
GND	9	10	GPIO 15 (RXD)
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	GND
GPIO 22	15	16	GPIO 23
+ 3,3 V	17	18	GPIO 24
(MOSI) GPIO 10	19	20	GND
(MISO) GPIO 9	21	22	GPIO 25
(SCLK) GPIO 11	23	24	GPIO 8 (CE0)
GND	25	26	GPIO 7 (CE1)
ID_SD	27	28	ID_SC
GPIO 5	29	30	GND
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	GND
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
GND	39	40	GPIO 21

4.4 BLOCKCHAIN DATABASE

Blockchain in this system can be thought of a database. The difference of the Blockchain Database from a standard database is it is unchangeable thus contains very trustful data. We implemented it specifically using a smart contract. Smart contracts are programs that resides on a address of the blockchain. They contain state variables which represent the current state of the contract and related functions to modify these variables. In our case, the state variables of the contract which we wrote in “Solidity” language contains the log data of device activities as strings of format: “date, email, success” format. So, we store when a user tries to access a door and the result of this attempt whether it is successful or not.



```
pragma solidity ^0.5.0;

contract DeviceActivityLogs {
    //deviceCode => deviceLogs
    mapping(string => Device) db;

    struct Device {
        uint logCount;
        string[] logs;
    }

    function addDevice(string memory _deviceCode) public {···
    }

    function addLog(string memory _deviceCode, string memory log) public{
        Device memory device = db[_deviceCode];
        device.logs[device.logCount++] = log;
    }

    // page number 4 => 81,100
    function getLogs(string memory _deviceCode, uint pageNumber) public returns(string memory) {···
    }


}
```

5. RELATED WORK/SIMILAR SOLUTIONS

i-Neighbour**[1]**:

This company has many IoT project and one of them is Smart Door Lock. In their project, they have many methods to unlock the door such as voice recognizing, touching different lock screen for each door or reading QR code in front of the door. However, in our project, we have only one QR code to lock or unlock the door. And the other difference is that we generate a QR code for each time, while i-Neighbour is reads a static QR code each time. Also, there is no usage of the Blockchain technology for logging user activities whereas we will use this technology.
In addition, they are providing a cloud-based solution which we will also include in our project.

Kwitset Kevo Bluetooth Smart Lock**[2]**:

This project locks or unlocks the door with Bluetooth connection whereas we are using QR code and the Internet to send it to a cloud server . And in the security part, there is no information about the Blockchain whereas we will use the Blockchain.

Smart Lock **[3]**

This company has many projects and one of them is the smart door lock with QR code access. Similarities between our project and this one is to generate a QR code to unlock the smart lock. And in the security part, there is no information about Blockchain whereas we will use Blockchain.

6. CONCLUSION AND IMPACT

Our project combines the technology and daily activity. The traditional door lock mechanism is not secure as much as desired. With the growth of Internet and its usage, we can give more security to something which is important to people.

Another thing is that our project comes with a compact solution. In the traditional way, you should have many keys if you have many doors. To keep the keys safe and remember which key opens which door is not the big issue in our project. In our project one key will be sufficient to open corresponding door(s).

For the economic part, we plan to design this system modular. So anyone buy this system, he/she shall be able to install it on any door. And, the system itself does not cost too much as its components are highly available in the market and the software is free for anyone.

Finally, our project covers many components in Computer & Software environment such as microcontroller design and implementation, graphical user interface design and implementation, blockchain. Therefore, communication of these components are crucial because many users will use our app at any time.

7.REFERENCES

[1] i-Neighbour, website: https://www.i-neighbour.com

[2] Kwitset Kevo Bluetooth Smart Lock, website: https://www.kwitset.com/kevo/smart-lock/

[3] Smart Lock, website: http://www.homeserva.com

[4] Volley https://github.com/google/volley