Erkenly Description

Overview:

Erkenly app is to rent parking spots from anywhere in Egypt, it is a first come first server approach, you can first rent the parking spot but it will be pending until you open the door (you can only rent one spot at a time), if you do not open the garage with in a specific time, the rent is cancelled and the spot is available to be rented , and after the rent is confirmed with opening the garage door, a timer starts and you will pay by the hour when you end the Rent period. We also considered the case of a fire using a FlameSensor, if there is a fire all doors are opened, all rentings are ended, all spots are marked as unavailable, we used 1 servo motor rather than 2, because the second servo motor malfunctioned and we could not get another one , so I hope you understand.

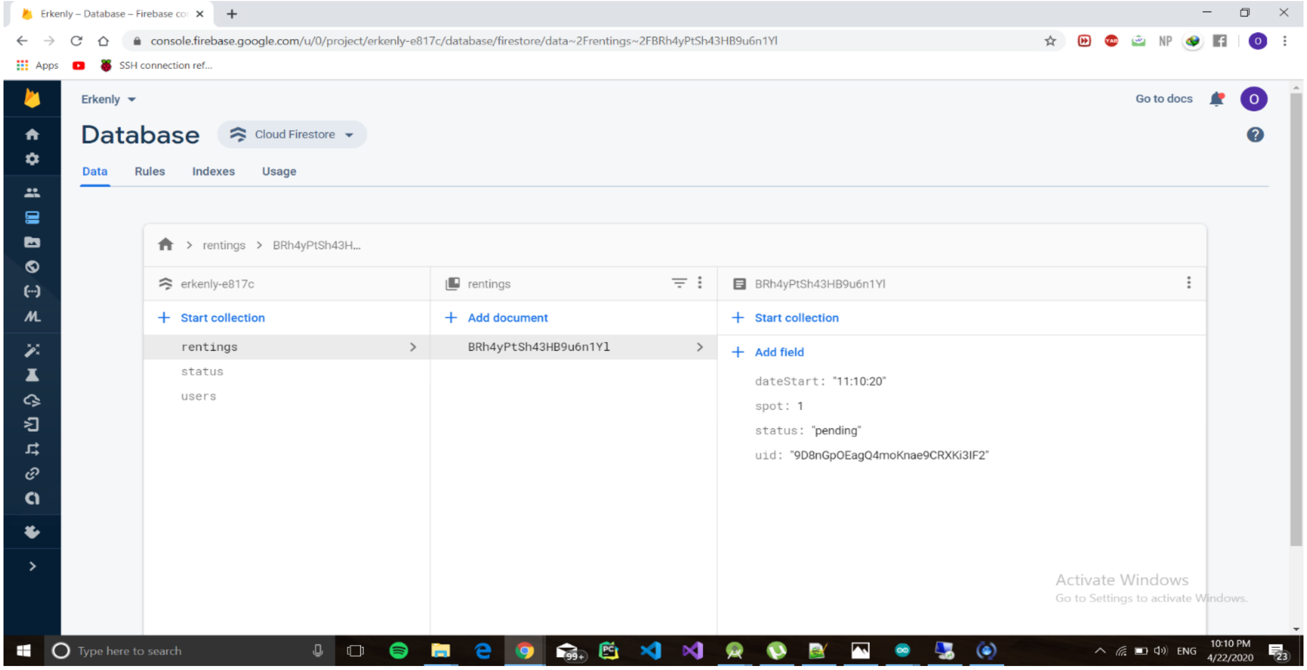
Server and Raspberry pi:

We used the Raspberry pi to make the Server , we used NOIP to get a host name for the Rpi IP address which is dynamic and we used this hostname http://zeplosmarthome.ddns.net , We used Firebase Authentication for Login , Registration using email and Password. Firebase Firestore as the Real Time Database. Firebase Cloud Messaging for push notifications send to the app using the token stored in the Database. Also the python files in the code are all running in parallel in Thread. FCM and HTTPServer and FireAlarm are always running while Parkfinish is only run when someone is opening a garage or ending a rent.

Database:

We have 3 collections (rentings,users,status) here are screenshots and explanation of each one.

Rentings



rentings
status
users

BRh4yPtSh43HB9u6n1Y1

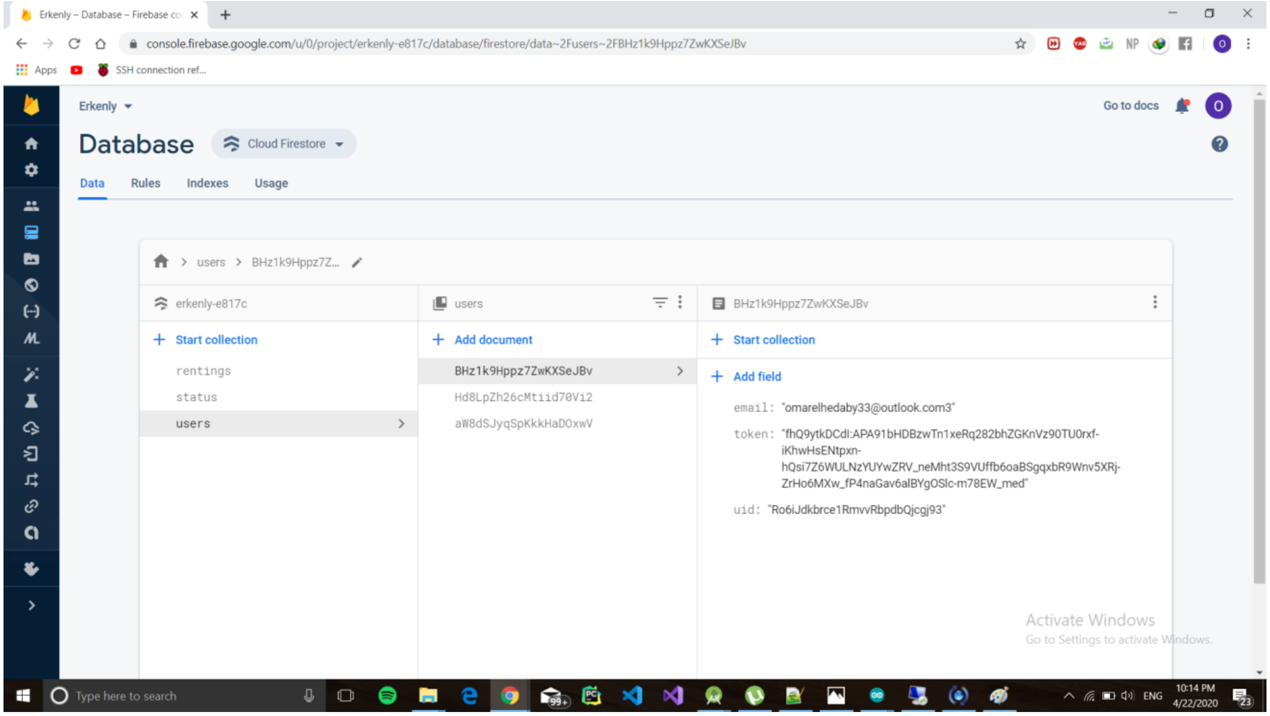
```
dateStart: "11:10:20"
spot: 1
status: "pending"
uid: "9D8nGpOEagQ4moKnae9CRXKi3IF2"
```

dateStart is the time the rent started , uid is the user id

spot is the spot area that is rented

status : pending if the spot is only rented not confirmed (the user didn’t open the garage) and current if the user confirmed the renting

Users



rentings
status
users

BHz1k9Hppz7ZwKXSeJBv
Hd8LpZh26cMtiid70Vi2
aW8dSJyqSpKkkHaDOxwV

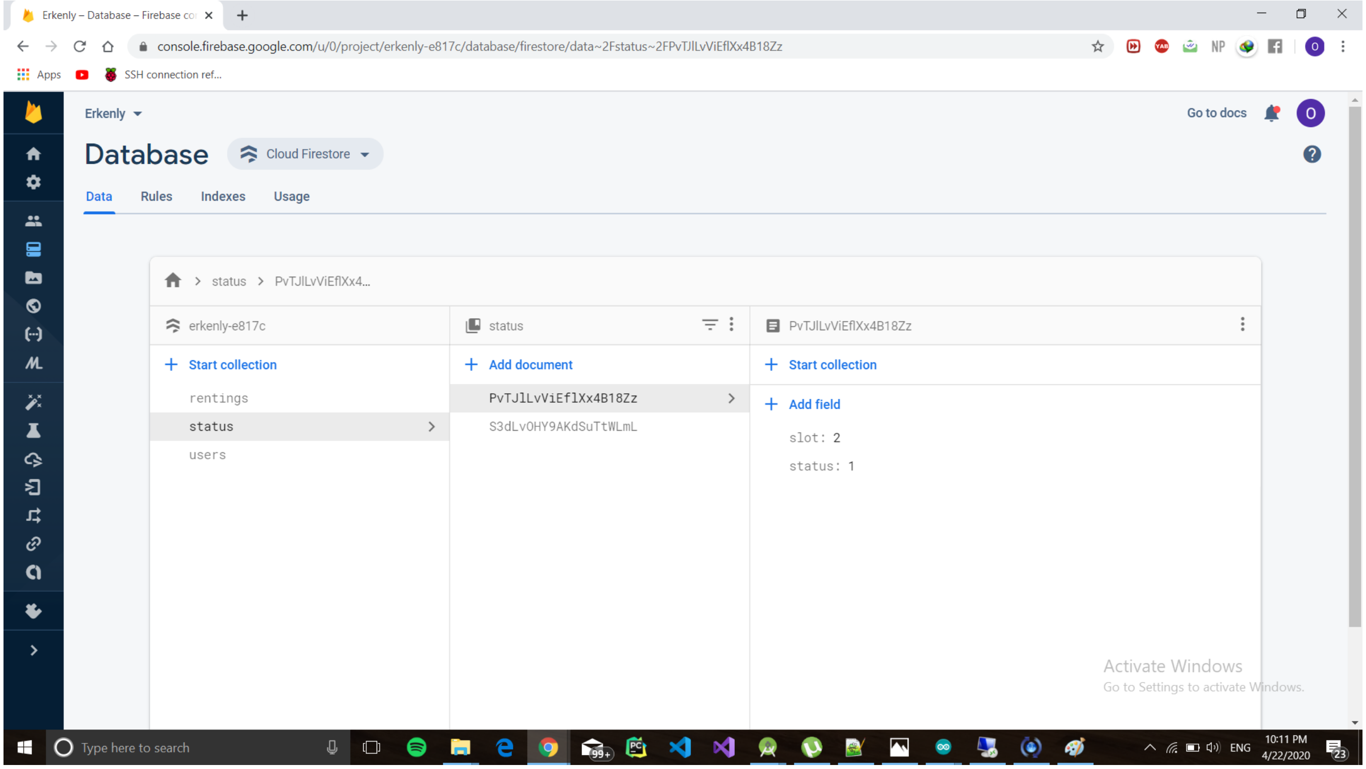
```
email: "omarelhedaby33@outlook.com3"
token: "fhQ9ytkDCdI:APA91bHDBzwTn1xeRq282bhZGKnVz90TU0rxf-
       iKhwHsENtpxn-
       hQsi7Z6WULNzYUYwZRV_neMht3S9VUffb6oaBSgqxbR9Wnv5XRj-
       ZrHo6MXw_fP4naGav6alBYgOSlc-m78EW_med"
uid: "Ro6iJdkbrce1RmvvRbpdbQjcgj93"
```

Email: email address used in registeration

Token : token used for Firebase Cloud Messaging

Uid: user is

Status



rentings
status
users

PvTJlLvViEflXx4B18Zz
S3dLvOHY9AKdSuTtWLmL

```
slot: 2
status: 1
```

Status: 1 for Available , 0 for Non Available , -1 for out of service (in case of fire)

HTTP Code:

HTTPServer: is the server code it responds to requests from Android App and runs queries on the Firebase Firestore, it also is the starting process as it runs all other Threads, FCM and FireAlarm are always running , while ParkFinish is only run when a openGarage or payRent request is requested.

FCM: is the code that listens to updates on the rentings table and records start times that their status is pending and compare these with the time now, when time exceed the interval required it sends a message that the Rent is cancelled and it deletes the Rent data from rentings.

FireAlarm: it reads the Flamesensor readings , If there is a fire sends messages to users that have spots rented telling them that there is a fire, it deletes all rentings. And mark all spots status as -1 (Out of Order) so no one can book a new renting.

ParkFinish: it takes 2 arguments spot number and operation , operation can be close or open. If it is open it opens the door and wait until the car is close to the ultrasonic then closes it again , close does the opposite , it is run on another Thread that is run by HTTPServer

Components Used:

Raspberry pi

Flame Sensor

Ultrasonic Sensor

Servo Motor

Buzzer

2 resistor 330 Ohm