

Nesnelerin İnterneti ve
Uygulamaları

Proje Ödev 1

Cengiz Altürk G140910048

Uygulamanın Amacı:

Bir iş yeri veya bir grup çalışanın giriş çıkışını takip eden bir sistem yapmaktır.

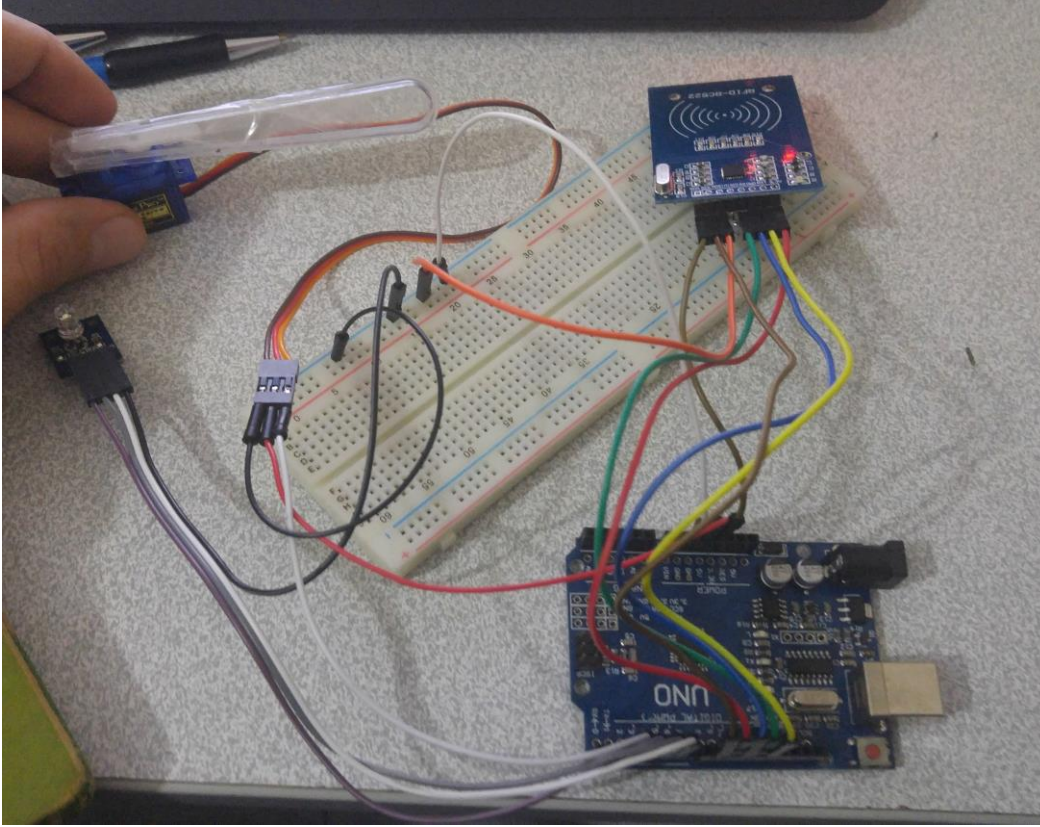
Uygulamanın Tanımı:

Google Firebase bulut platformuna rfid sistemi ile bir etiket okunularak eğer kayıtlı bir etiket ise servo motor rgb led ile yeşil renkte bir uyarı veren eğer kayıtlı değilse rgb led ile kırmızı yanan ve servo motoru çalıştırmayan bir arduino uygulaması üzerinden veri yazma, Mobil uygulaması üzerinden veri okuma gerçekleştirilecektir.

Ekipman Listesi ve Kullanılan Teknolojiler:

- Firebase
- Arduino UNO
- Kart Okuyucu RFID-RC522
- Servo motor SG90
- RGB LED
- Android uygulama geliştirme ortamı (Mit App Inventor 2)
- Android uygulamayı akıllı telefona yükleme yazılımı (Mit App Inventor 2 Companion)
- C# Windows Form (Bu Teknolojiyi Wifi Modülüm olmadığından arduinodan gelen veriyi Serial Port üzerinden haberleştirerek veriyi Bilgisayar üzerinden gönderdim.)

Uygulama Görüntüsü



Firestore Görüntüsü

<https://iotodev-b7f23.firebaseio.com/>

iotodev-b7f23

- + -L0ctHuWqGXDbg3x4G9c
- + -L0ctI3Pc2BV49E-BLhd
- + -L0ctIHcEX28I8QrIL5o
- + -L0ctIaXWKWXws8wkJQ8
- -L0ctJivdVOTIkhjq1sM
 - Name: "Kisi Tarih Saat"
 - Value: "Izinsiz Giris\r 2017:12:18 12:12:"
- -L0dApelVv7ybm-teCQ
 - Name: "Kisi Tarih Saat"
 - Value: "Beyaz kart\r 2017:12:18 13:33:"

Giriş yapan kişilerin eğer kayıtlı ise isimleri ile giriş yaptıkları tarih ve saati kayıt altında tutan eğer kayıtlı değilse de izinsiz giriş isminde kayıt altında tutar.

Arduino Kodları

```
#include <SPI.h>
#include <RFID.h>
#include<Servo.h>
Servo sg90;
int pozisyon=0;
RFID rc522(10,9);
int kirmizi =4 ;
int yesil = 5;
int mavi = 6;
byte kart1[5]={51,44,242,197,40};
byte kart2[5]={112,56,177,128,121};
byte kart3[5]={236,183,86,230,235};
boolean izin1,izin2,izin3,kart1Durum,kart2Durum,kart3Durum;
```

```
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  rc522.init();
  pinMode(kirmizi, OUTPUT);
  pinMode(mavi, OUTPUT);
  pinMode(yesil, OUTPUT);
  sg90.attach(3);
}
```

```
void loop()
{
  izin1=true;
  izin2=true;
  izin3=true;
  kart1Durum=false;
  kart2Durum=false;
  kart3Durum=false;
  if (rc522.isCard()) {
    if (rc522.readCardSerial()) {
      if(kart2Durum==false && kart3Durum==false)
      {
        for(int j=0;j<5;j++)
        {
          if(rc522.serNum[j]!=kart1[j])
          {
```

```

        izin1=false;
    }
}
if(izin1==true)
{
    kart1Durum=true;
    kart2Durum=false;
    izin2=false;
    kart3Durum=false;
    izin3=false;
}
else
{
    izin1=false;
}
}
if(kart1Durum==false && kart3Durum==false)
{
    for(int j=0;j<5;j++)
    {
        if(rc522.serNum[j]!=kart2[j])
        {
            izin2=false;
        }
    }
    if(izin2==true)
    {
        kart2Durum=true;
        kart1Durum=false;
        izin1=false;
        kart3Durum=false;
        izin3=false;
    }
    else
    {
        izin2=false;
    }
}
if(kart1Durum==false && kart2Durum==false)
{
    for(int j=0;j<5;j++)
    {
        if(rc522.serNum[j]!=kart3[j])

```

```
        {
            izin3=false;
        }
    }
    if(izin3==true)
    {
        kart3Durum=true;
        kart1Durum=false;
        izin1=false;
        kart2Durum=false;
        izin2=false;
    }
    else
    {
        izin3=false;
    }
}
```

```
if(kart1Durum){
    if(izin1)
    {
        Serial.println("Beyaz kart");
        ac();
    }
}
```

```
else if(kart2Durum){
    if(izin2){
        Serial.println("Ogrenci kart");
        ac();
    }
}
```

```
else if(kart3Durum)
{
    if(izin3)
    {
        Serial.println("play kart");
        ac();
    }
}
```

```

else
{
    Serial.println("Izinsiz Giris");
    digitalWrite(kirmizi, HIGH);
    digitalWrite(mavi, LOW);
    digitalWrite(yesil, LOW);
    delay(1000);
    digitalWrite(kirmizi, LOW);
}
}
else
{
    //Serial.println("0");
}

rc522.halt();
}
void ac()
{
    sg90.write(40);
    digitalWrite(kirmizi, LOW );
    digitalWrite(mavi, LOW);
    digitalWrite(yesil, HIGH);
    delay(1000);
    digitalWrite(yesil, LOW);
    delay(3000);
    kapa();
}
void kapa()
{
    sg90.write(130);
}

```

Windows Form Kodları

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Newtonsoft.Json;
using System.Net;
using System.IO;

```

```

using System.IO.Ports;

namespace iotForm
{
    public partial class Form1 : Form
    {
        SerialPort sp= new SerialPort("COM3", 9600);
        public Form1()
        {
            sp.Open();
            InitializeComponent();
            while(true)
            {
                string isim = sp.ReadLine();
                if(isim!="")
                {
                    DateTime date = DateTime.Now;
                    string Tarih = date.ToString("yyyy:MM:dd");
                    string Zaman = date.ToString("HH:mm:ss");
                    var json = JsonConvert.SerializeObject(new
                    {
                        Name = "Kisi Tarih Saat",
                        Value = isim + " " + Tarih + " " + Zaman
                    });
                    var request = WebRequest.CreateHttp("https://iotodev-
b7f23.firebaseio.com/.json");
                    request.Method = "POST";
                    request.ContentType = "application/json";
                    var buffer = Encoding.UTF8.GetBytes(json);
                    request.ContentLength = buffer.Length;
                    request.GetRequestStream().Write(buffer, 0, buffer.Length);
                    var response = request.GetResponse();
                    json = (new
StreamReader(response.GetResponseStream())).ReadToEnd();
                }

            }

        }
        private void Form1_FormClosed(object sender, FormClosedEventArgs e)
        {
            // Form kapandığında Seri Port Kapatılmış Olacak.
            if (sp.IsOpen == true)
            {
                sp.Close();
            }

        }
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            string isim = sp.ReadExisting();

            DateTime date = DateTime.Now;
            string Tarih = date.ToString("yyyy:MM:dd");
            string Zaman = date.ToString("HH:mm:ss");
            var json = JsonConvert.SerializeObject(new
            {
                Name = "Kisi Tarih Saat",
                Value = isim + " " + Tarih + " " + Zaman
            }

```



```

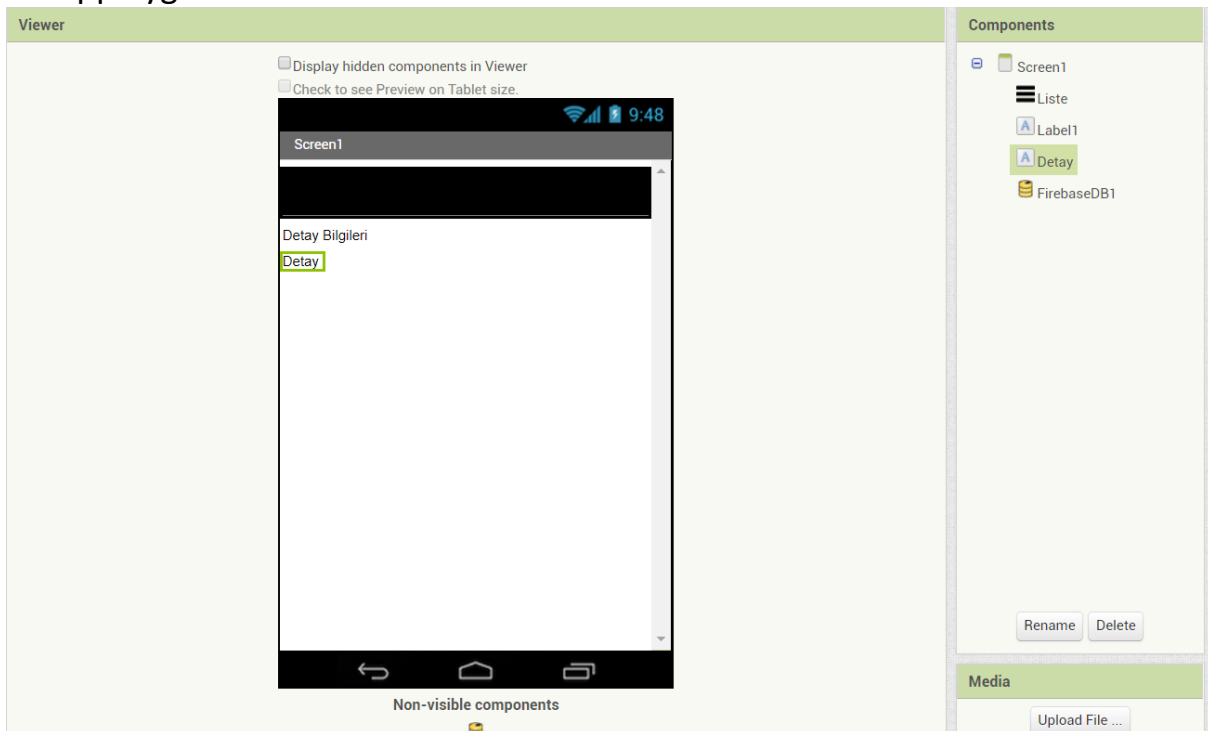
});
var request = WebRequest.CreateHttp("https://iotodev-
b7f23.firebaseio.com/.json");
request.Method = "POST";
request.ContentType = "application/json";
var buffer = Encoding.UTF8.GetBytes(json);
request.ContentLength = buffer.Length;
request.GetRequestStream().Write(buffer, 0, buffer.Length);
var response = request.GetResponse();
json = (new StreamReader(response.GetResponseStream())).ReadToEnd();

}
private void Yaz(string k,string t, string z)
{
    var json = JsonConvert.SerializeObject(new
    {
        Name = "Kisi Tarih Saat",
        Value = k + " " + t + " " + z
    });
    var request = WebRequest.CreateHttp("https://iotodev-
b7f23.firebaseio.com/.json");
    request.Method = "POST";
    request.ContentType = "application/json";
    var buffer = Encoding.UTF8.GetBytes(json);
    request.ContentLength = buffer.Length;
    request.GetRequestStream().Write(buffer, 0, buffer.Length);
    var response = request.GetResponse();
    json = (new StreamReader(response.GetResponseStream())).ReadToEnd();
}

}
}

```

MIT App uygulaması



initialize global Kisi to create empty list

```
when Screen1.Initialize
do
  call FirebaseDatabase1.GetValue
    tag
    valueIfTagNotThere
```

```
when Liste.AfterPicking
do
  call FirebaseDatabase1.GetValue
    tag
    valueIfTagNotThere
```

```
when FirebaseDatabase1.DataChanged
tag value
do
  call FirebaseDatabase1.GetValue
    tag
    valueIfTagNotThere
```

! 0

Show Warnings

```
when FirebaseDatabase1.GotValue
tag value
  Delay.Text to get value
```