****

**KIRIKKALE ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ RAPORU**

**ANDROİD TABANLI AKILLI GARAJ SİSTEMİ**

**Danışman: Dr. Bülent Gürsel Emiroğlu**

**Ayhan ÜNAL**

**Mert BALASAR**

İÇİNDEKİLER ................................. Sayfa

ÖZET 2

1. GİRİŞ 3

1.1. Zaman Ayarlı Olaylar 3

1.2. Tetikleyici Olaylar 3

1.3.Arduino 4

1.4.Android……………………………………………………………..……………..4

2. Amaç 5

3. Hedefler 5

4. Materyal & Metot 6

4.1. Esp 8266…………………………………………………………………………..8

4.2. Firebase…………………………………………………………………………...9

4.2.1. Firebase Projeye Dahil Etme……………………………………………………9

5. Yapılan Çalışmalar 10

5.1. Garajın Maketi…………………………………………………………………..10

5.2. Arduino Devresi ve Arduino Programı…………………………………………..12

5.3. Windows Konsol Uygulaması…………………………………………………….15

5.4. Android Uygulaması……………………………………………………………...17

5.5. Sistemin Çalışma Şart ve Koşulları………………………………………………..27

5.6. Karşılaşılan Zorluk ve Problemler…………………………………………………28

5.7. Sistemin Teknik Özellikleri………………………………………………………..28

6. Öneriler ve Referanslar……………………………………………………………….29

**ÖZET**

Bu çalışmanın amacı, günümüzde oldukça popüler olan IoT teknolojisini, bir garaj maketine uyarlayıp akıllı sistem geliştirmektir. Bu kapsamda bir garaj maketi yapılıp içerisine arduino devresi yerleştirilecektir. Bu devre sayesinde garajın kapı ve ışık kontrolu ağ’da yer alan gerçek zamanlı bir veritabanı üzerinden sağlanacaktır. Aynı veritabanı android tabanlı geliştirilen uygulama üzerinden de kullanılıp garajın kontrolu, son kullanıcıya hitap eden arayüz ile sağlanacaktır. Bu uygulama da kullanıcı giriş yapıp ağ’da ki veritabanına komutlar gönderebilecektir ve bu komutlar arduino tarafından eşzamanlı okunarak ilgili komut gerçekleşmiş olacaktır. Aynı ağ üzerinde bir bulut veritabanı da oluşturulmuştur bu da sistemde ki kullanıcıları ve yetkilileri depolar. Tüm bu sistemin kontolu Firebase sistemi üzerinden sağlanmıştır. Bu çalışma ile garaja giriş çıkış yapan kullanıcılar çaba sarfetmeden kayıtlı araçları ile doğrudan garaja girebilecektir. Garajın ışıklarını açıp kapatma işlemini cep telefonu üzerinden kontrol edebilecektir. Ayrıca bu çalışma, kullanıcıların giriş çıkış saatleri, giriş ve çıkış yaptığı sırada yüz ifadeleri vb. gibi parametlereleri baz alan bir yapay zeka uygulaması geliştirilmesine zemin hazırlamaktadır.

**ABSTRACT**

The aim of this study is to develop a smart system by adapting IoT technology, which is very popular today, to a garage model. In this context, a garage model will be made and an arduino circuit will be placed inside. Thanks to this circuit, the door and light control of the garage will be provided through a real-time database on the network. The same database will be used on the android based application, and the control of the garage will be provided with an interface that appeals to the end user. In this application, the user will be able to log in and send commands to the database on the network, and these commands will be read simultaneously by arduino and the related command will have been realized. A cloud database has also been created on the same network, which stores users and authorities in the system. The control of all this system was provided through the Fire-base system. With this study, users entering and exiting the garage will be able to enter the garage directly with their registered vehicles without effort. Will be able to control the lights of the garage on and off via mobile phone. In addition, this study covers the entrance and exit times of users, facial expressions when entering and leaving. It prepares the ground for the development of an artificial intelligence application based on such parameters.

1. **GİRİŞ**

Akıllı sistemler**,** aydınlatma ve ısıtma gibi alet ve sistemlerin uzaktan izlenmesini ve yönetilmesini sağlamak için bir ağ ile birbirine bağlı cihazların kullanıldığı sistemler için kullanılan bir terimdir. Akıllı sistem teknolojisi; çoğunlukla ev sahiplerinin sahip olduğu akıllı cihazlarla, akıllı cep telefonlarına ve/veya tabletlerine yüklenmiş olan akıllı uygulamalarla evlerinin güvenliğini, konforunu ve enerji verimliliğini kontrol edebildiği sistemlerdir. Garaj kapısını açıp kapatan sistemler, alarm sistemleri, kahve makinesi zamanlayıcı gibi sistemler akıllı sistemlerinin basit örneklerindendir.

Bununla birlikte, otomasyon ve akıllı sistemlerden bahsederken, yetenek ve örnekler daha ileri seviyelere gidiyor. Ayrı ayrı çalışan cihazlar yerine akıllı bir sistem, bir ana otomasyon denetleyicisi tarafından kontrol edilen birden fazla alt sistemi birleştirir. Bu ana otomasyon denetleyicisi, ev otomasyon sisteminin oyun kurucusu gibidir ve ev çevresindeki tüm aygıtlardan girdileri alır, komutlar verir ve her şeyi kontrol eder. Bu denetleyiciler genellikle çeşitli olaylara dayanan tek veya çoklu eylemleri yürütmelerine olanak tanıyan karmaşık bir yazılım çalıştırır. Bu olaylar birçok biçimde olabilir, ancak esas olarak sadece iki kategoriye ayrılabilir: zaman ayarlı ve tetikleyici olaylar.

* 1. **Zaman Ayarlı Olaylar**

Sistem sahibinin istediği zamanlarda veya sistemin bulunduğu coğrafik konuma göre değişen gün ışığı zamanlarına göre yapılması istenen durumlardır. Bu özellik sayesinde, sistem her gün belirlenen saatlerde belirlenen o saatlere özgü işlemleri gerçekleştirir. Örneğin; her gün sabah garaj pencerelerinin açılarak garajın havalanmasının sağlanması veya gün batımı/doğumuna göre garajın dışarısındaki ışıkların açılıp kapanması gibi.

* 1. **Tetikleyici Olaylar**

Otomasyon sisteminin çalışmasını tetikleyen durumlardır. Bir butona basılması, bir kapının açılması, bir hareketin algılanması, bir sensörden gelen uyarı gibi durumlar tetikleyici olaylardır. Bir butona bastığımda, şu olayı yap diye bir ayar yapabilirsiniz. Veya garaj kapısının açılması durumunda garaj ışığının açılmasını sağlayabilirsiniz. Herhangi bir oda içerisinde 5 dakika boyunca herhangi bir hareket algılanmazsa oda ışığının kapatılmasını sağlayabilirsiniz. Ortamda ki herhangi bir yerde duman algılanması durumunda sistemin kullanıcıya SMS atmasını, e-posta göndermesini veya aramasını sağlayabilirsiniz.

Çeşitli zaman ayarlı ve tetikleyici olayları birleştirerek akıllı bir sistemin yapabileceği şeylerin sınırı oldukça yüksektir. Koşullu mantıkla birleştirildiğinde akıllı sistemler daha da güçlenmektedir. Örneğin; eğer bu butona basarsam ve zaman gece yarısı ile sabah 5 arasında ise o zaman ışığın şiddetini %50 olarak ayarla değilse %90 ayarla gibi bir komut verilebilir.

Evinizde kurulmuş bir ev otomasyon sistemi ile evden ya da dünyanın öbür ucundan internet üzerinden ayrıntılı işlevler gerçekleştirebilirsiniz. Mobil cihazınızdan tek tuşa basarak alarmınızı devreden çıkarabilir, güneşliklerinizi düşürebilir, şömineyi açabilir, ışıkları kısabilir, SPAyı ısıtabilir ve romantik müziği açabilirsiniz.

* 1. **Arduino**

Bu çalışmanın temelinde arduino yatmaktadır. Arduino; basit entegre edilebilen ve kolay kullanım özelliğine sahip yapısıyla, kodlama dilini başlangıç seviyesinde bilen birinin bile gelişmiş yazılımlar yapmasını sağlar. Bu yazılımlar modüllerle, kartlarla ve cihazlarla entegre edilebileceği gibi, interaktif nesneler oluşturulmasını sağlayabilir. Açık kaynaklı bir donanım olan Arduino, elektronik devrelerini kullanarak kendi cihazınızı yapmanıza da yardımcı olur.

Arduino, temelde fiziksel bir programlama modülü olarak tanımlanabilir. Bir giriş/çıkış (input/output) kartına sahip olmakla birlikte, processing diliyle çalışan bir uygulama barındırmaktadır. Temelde bir kart sistemi üzerinden çalışır. Bu kartın donanımında ise Atmel AVR mikrodenetleyici sistemi ana öge olarak kabul edilebilir. Bunun yanında, çeşitli yan elemanlar vasıtasıyla diğer devreler ve modüller ile bağlantı kurulur.

Genellikle sanılanın aksine, Arduino bir mikroişlemci programı değildir. Ancak mikroişlemcilerle bağlantılı bir şekilde çalışır ve onların daha iyi kodlama yapabilmeleri için geliştirici işlevi görür. Mikroişlemciye USB, girdi/çıktı pinleri, LED veya güç girişi yoluyla bağlanarak yardımcı olabilir.

Projenin gerçekleşebilmesi için arduino yanında çeşitli sensör ve malzemeler kullanılmıştır. Bunlar;

* Servo Motor
* ESP 8266
* Rfid
* Led
* Direnç
* Jumper

Gibi temel elektronik malzemeler kullanılmıştır. Bunlar dışında bir çok elektronik devre elemanı ve bağlantı kullanılmıştır bunlar ilerleyen bölümde anlatılacaktır.

* 1. Android

Android, mobil cihazlar yani tablet bilgisayarlar (PDA) ve akıllı cep telefonları için Google ve Open Handset Allience tarafından kodlanmış Linux İşletim Sistemi tabanlı açık kaynak kodlu bir işletim sistemidir. Android işletim sistemi tarafından desteklenen uygulama uzantısı ".apk"dir. Android açık kaynak kodlu olması sebebiyle, aygıtların fonksiyonelliğini arttırmak için uygulamalar yazan ve geliştiren geniş bir geliştirici grubuna sahiptir.

Anroid uygulama geliştirmek için çeşitli geliştirme ortamları ve dilleri mevcut ancak biz projemiz kapsamında geliştirme ortamı olarak Android Studio, programlama dili olarak da Java’yı tercih ettik.

1. AMAÇ

Projenin amacı, kullanıcıların otopark/garaj sistemlerini daha pratik ve kolay kullanabilmesini sağlamaktır. Normal bir garaja giren kullanıcılar garaj kapısını aracından inip insani yollarla veya uzaktan kumanda ile açmakta bu şekilde insanların rahatını bozması ve yanında sürekli kumandayla gezmesi gibi ergonominin yanında bu işin extra maliyeti de oluşmakta. Günümüz teknolojisi bu kadar ileri gitmişken, artık her insan da mevcut olan akıllı cep telefonu ile garajın kapısını açıp kapatabilecek ve telefonundan diğer kontrolleri sağlayabilcek. Üstelik garajın kontrollerini sağlamak için garaja belirli bir mesafede veya garaj ile aynı ağa bağlı olma zorunluluğu da gerekmiyor. İstediğiniz mesafeden kendi mobil ağınız ya da herhangi bir wifi ağıyla garajın kontrolu sağlanabilecektir. Bu gerçeğe dönütte pek anlamlı olmasa da bizim projemizde ki en önemli amacımız böyle bir teknolojiyi kullanıp bir ürüne dönüştürebilmektir.

Proje 2 dönem şeklinde gerçekleştirilecek olup ilk dönemde ;

* Garajın Maketi
* Arduino bağlantıları ve makete entegre edilmesi
* Devrenin kontrolü için bilgisayar ile Arduino haberleştirilecek ve garaja giren-çıkan arabalardan alınan veriler(Rfid ile) bilgisayar ortamında test edilecek.

İkinci dönem de ise ;

* Maketi ve devresi yapılan garajın yönetimi ve arayüzü için android tabanlı uygulama yazılacaktır.

Uygulamada yeni kullanıcı ekleme, kullanıcı girişi, kullanıcı sayfası gibi ekranlar olacaktır. Yukarıda da bahsedildiği üzere kullanıcı, garajın kapısını açma-kapatma, ışıkları açma-kapatma gibi birçok detay göreceği ekranlar tasarlanacak ve arkaplanda arduino ile haberleştirilecektir. Her kullanıcı veri tabanında tutulacaktır.

1. **Hedefler**

Çalışmanın ana amacına ulaşırken, tamamlanması gereken yan amaçlar da olacaktır. Bu bölümde çalışmanın hedef ya da hedefleri mümkün olduğunca spesifik ve ölçülebilir nitelikte belirtilmiştir.

Çalışmada ilk hedef garajın maketini tamamlamaktır. Dikdörtgen bir alan üzerine garaj ve bahçesi yapılacaktır. Garajın içerisine hazırlanan arduino devresi yerleştirilecek olup gerekli bağlanılar yapılacaktır. Garajın kapısı, arduino üzerine bağlanan servo-motor ile sağlanacaktır. Aynı şekilde ışıklandırma gibi garajın tüm özellikleri arduino sayesinde gerçekleşebilecektir.

Makete ek olarak arduino devresinin çalışabilmesi için eşzamanlı olarak arduino programının kodlaması da yapılmıştır. Gerekli pinlerin tanımlanması, input/output elemanlarının belirlenmesi, program ana akış şemasının çıkarılması ve gerekli teknik kodlamanın yapılması gibi programatik adım da tamamlanmıştır.

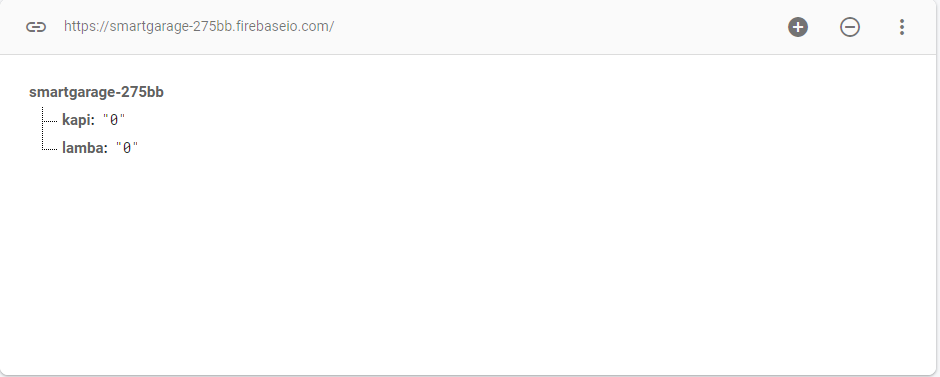
Projenin arduino tarafı tamamlandıktan sonra son kullanıcın garaj ile etkileşime geçebilmesi için akıllı cihazlar üzerinde çalışabilen android işletim sistemli bir uygulama geliştirilecektir. Bu uygulamaya yetkili ve kullanıcı girişi olmak üzere erişim sağlanabilecektir. Program üzerindeki yetkili kişiler, mevcut kullanıcıların resimlerini, mail adreslerini, garaj içinde ki park yerlerini görüntüleyebilecek ve yeni kullanıcı ekleyebilecektir. Mevcut kullanıcılar, kendi kişisel bilgileri ile sisteme giriş yaptıklarında ise garajın kapı ve ışık durumunu görüntüleyip bunları değiştirebilecektir.

1. **Materyal & Metot**

Bu bölümde, proje çalışmasında tanımlanmış olan hedeflere ulaşabilmek için hangi verilerin, hangi kaynaklardan, hangi tekniklerle, hangi araçlarla, kimler tarafından ve nasıl toplanacağı ayrıntılı olarak açıklanmıştır. Araştırma yöntemi, problemin çözümü için gerekli verilerin toplanması, analizi ve yorumlanmasını içerir.

Garajın kapısı yani devrede ki servo-motor ve garajın ışıkları yani ledler, arduino programında ‘kapi’ ve ‘lamba’ değişkenlerinde tutulmaktadır. Bu değikenlerin değeri ‘1’ iken ilgili özellik açık/on durumuna ‘0’ iken kapalı/off durumuna geçmektedir. Bu değişkenlerin değerleri programda yerel olarak değil global biçimde Firebase üzerinde çalışan Real Time Database üzerinde tutulmaktadır. (Resim-1)

**Resim-1 (Firebase Real Time Database)**



Bu veritabanı eşzamanlı olarak çalışmaktadır. Yani hem android hem de arduino programı üzerine dahil edilip değerler üzerinde bir değişiklik yapıldıgında aynı anda iki program tarafından da görülmektedir. Android cihazdan kapı veya lamba değerleri değişdiğinde güncel değer arduino programı tarafından okunup ilgili komut gerçekleşecektir.

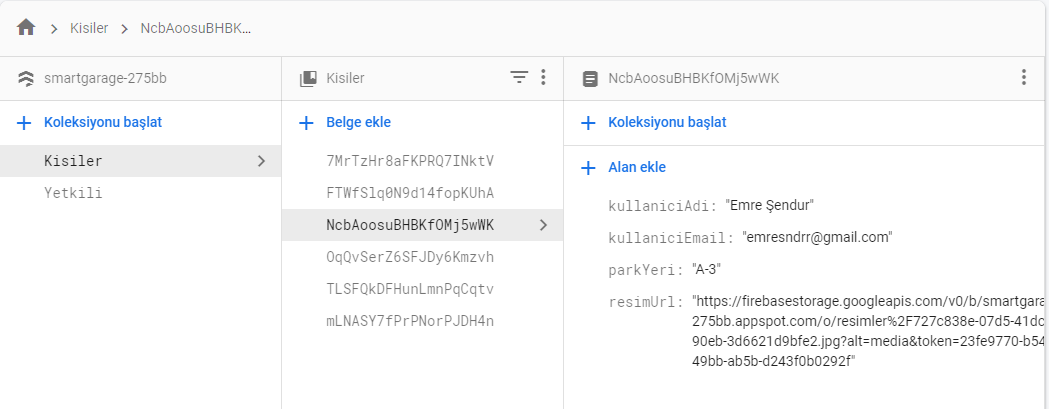
Android programda ki kullanıcılar ise yine Firebase üzerinde bulunan Cloud-Firestore üzerinde tutulmaktadır. Kullanıcılar Firebase üzerinde bulunan Authentication (Resim-2) ile girişleri kontrol ediliyor ve Firestore üzerinde de kullanıcıların bilgileri tutuluyor(Resim-3). Ayrıca kullanıcı resimleri Firebase Storage altında ‘resimler’ klasörü altında yer alıyor.(Resim-4).

**Resim-2 (Firebase Authentication)**



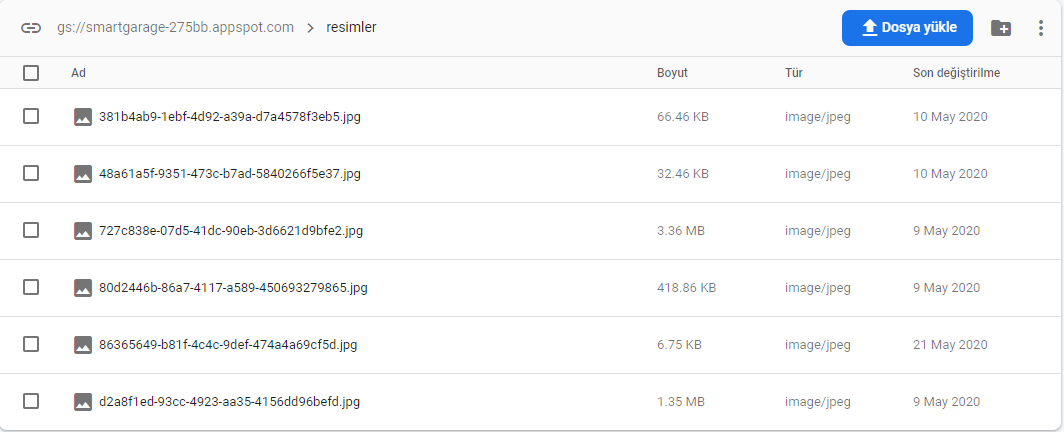
Authentication, mobil uygulamaya kayıtlı kullanıcıları göstermektedir. Kayıtlı kullanıcılar e-posta ve şifreleri ile buradaki kimlik doğrulama sayesinde sisteme giriş yapabilmektedir. Bu kullanıcıların diğer bilgileri(Adı, Soyadı, Park Yeri, Resim Url) Cloud Firestore üzerinde tutulmaktadır.(Resim-3)

**Resim-3 (Firebase Cloud Firestore)**



Bu veritabanı yapısı, NoSQL veritabanlarında görülen Json mimarisine sahiptir. Kişiler ana düğüm ve her kullanıcı için eşsiz bir(ekleme anında rastgele oluşturuluyor) kod ve her kullanıcı için özellikler bulunuyor. Aynı şekilde yetkili kullanıcılar da bu veritabanı altında yer alıyor. Burada ki önemli nokta kullanıcı resimleri değilde resimlerin url adresleri buradaki veritabanında tutuluyor. Resimleri saklamak için Firebase’in Storage alanı kullanılmaktadır. Buraya yüklenen her bir resim internete yükleniyor ve resmin url adresi veritabanına kayıt ediliyor.(Resim-4)

**Resim-4 (Firebase Storage)**



Sistemde kayıtlı olan kullanıcıların resimleri (Resim-4) de görülmektedir. Her resim için programlama anında unique bir değer oluşturulup resim adı olarak atanıyor. Bu resmin Uri değeri(Android’de resmin kimliği gibi düşünülebilir) kullanıcının e-posta adresiyle eşleştirilip veritabanına url adresi olarak kayıt ediliyor. (Bu kısımlar Yapılan Çalışmalar bölümünde detaylı olarak anlatılmıştır.)

Arduino ve Android arasında ki bu haberleşme, Arduino üzerinde bulunan ESP 8266 modülü ile Firebase’e iletilip Firebase’den de android programına gönderilerek sağlanmaktadır.

**4.1. ESP 8266**

ESP8266 Wifi Serial Transceiver Modül oldukça ekonomik ve kullanışlı bir Wifi modüldür. TCP/IP protokolünü desteklemektedir.  
  
ESP8266 üzerinde dahili anten bulunmaktadır. Bu sayede ortamdaki Wifi ağına rahatlıkla bağlanabilmekte, veri paketleri alıp gönderebilmektedir.

**Özellikleri**

* 802.11 b/g/n desteği
* Wi-Fi Direct (P2P) Desteği
* Dahili TCP/IP protokol yığını
* +19,5dBm çıkış gücü (802.11b modunda)
* Dahili düşük güç tüketimine sahip 32-bit'lik işlemci
* SDIO 1.1/2.0, SPI ve UART desteği
* STBC, 1x1 MIMO, 2x1 MIMO
* Uyanma ve veri paketi alma süresi < 2ms
* Stand-by durumunda güç tüketimi < 1mW

**4.2. Firebase**

Firebase servisinin geçmişi alsında James Tamplin ve Andrew Lee tarafından 2011 yılında kurulan Envolve şirketine dayanmakta. Geliştiricilerin web sayfalarına API aracılığıyla online chat özelliği kazandırabilmeleri amacıyla kurulan Envolve geliştiriciler tarafından planlanandan farklı bir şekilde (uygulama datalarını gerçek zamanlı olarak kullanıcılarla senkronize bir şekilde paylaşılması gibi) kullanılır. Kurucular bu sebeble 2012 yılında bildiğimiz FireBase’in de temellerinin atıldığı servis ayrımı kararını verirler. 2012 yılından itibaren pek çok gelişmeyi barındıran, mobil ve web uygulama geliştirme platformu olarak konumlandırılan Firebase 2014 yılında Google tarafından satın alınmıştır.

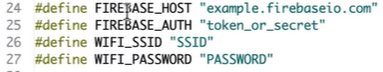
Satın alma sonrasında Google tarafından pek çok ek özellik eklenen servis küçük ve orta ölçekte uygulama geliştirme ihtiyaçlarının tamamını karşılama iddiasında. Bu amaçla ücretsiz kullanım (Spark Plan) imkanı da sunan (ek olarak ) servis başlangıç aşaması itibariyle kontrol paneli üzerinden kullanıcı verisi deposuna, kayıt, oturum ve ekran görüntüleme gibi uygulama içi etkinliklerden bildirim iletimine ve testlere platform farkı olmaksızın erişim ve yönetim imkanı sunmaktadır.

Firebase bu işlemler çerçevesinde ekstra bir sunucuya ve sunucu tarafında herhangi bir kod yazımına gerek olmaksızın yönetim paneli üzerinden çözüm sağlamakta. Bu amaçla kullanılabilecek özellikler; Realtime Database, Notification ve Remote Config. Diğer yandan Firebase Cloud Messaging (FCM) ile anlık bildirim gönderiminde kullanılan Google Cloud Messaging (GCM)’e de alternatif olan Firebase, veritabanı yapısı içerisinde yer alan tablolar ve SQL yerine root-child olarak verileri JSON (JavaScript Object Notation) formatında tutmakta. Bu yapı ve özellikler itibariyle veritabanında gerçekleşen her değişikliğin anında diğer kullanıcıya iletilmesi mümkün hale gelmektedir.

**4.2.1. Firebase Proje’ye Dahil Etmek**

Firebase üzerinden yeni bir proje olusturduktan sonra projeyi programa dahil edebilmek için bir token ve şifre veriliyor. Aynı zamanda android için .xml formatında bir konfigürasyon dosyası da verip android programına bu dosyanın dahil edilmesini istiyor. Arduino için Firebase host, verilen token değeri, Wifi adı ve Wifi şifresi (Wifi kısmı arduinoyu internete bağlamak için) programda tanımlanıyor ve arduino ile Firebase arasında iletişim kuruluyor.(Resim-5)

**Resim-5 (Arduino Firebase Bağlantısı)**



Andoid için ise verilen xml dosyasını proje dosyasına ekledikten sonra, build grade dosyasına Firebase import edildikten sonra android ile firebase arasında da iletişim sağlanmış olmaktadır. (Resim-6)

**Resim-6 (Android Firebase Bağlantısı)**



1. **Yapılan Çalışmalar**

Bu bölümde projenin tamamlanan kısımları aşağıda listelenmiş ve gerçeklenen yazılımlar da ekran görüntüleri ile açıklanmıştır.

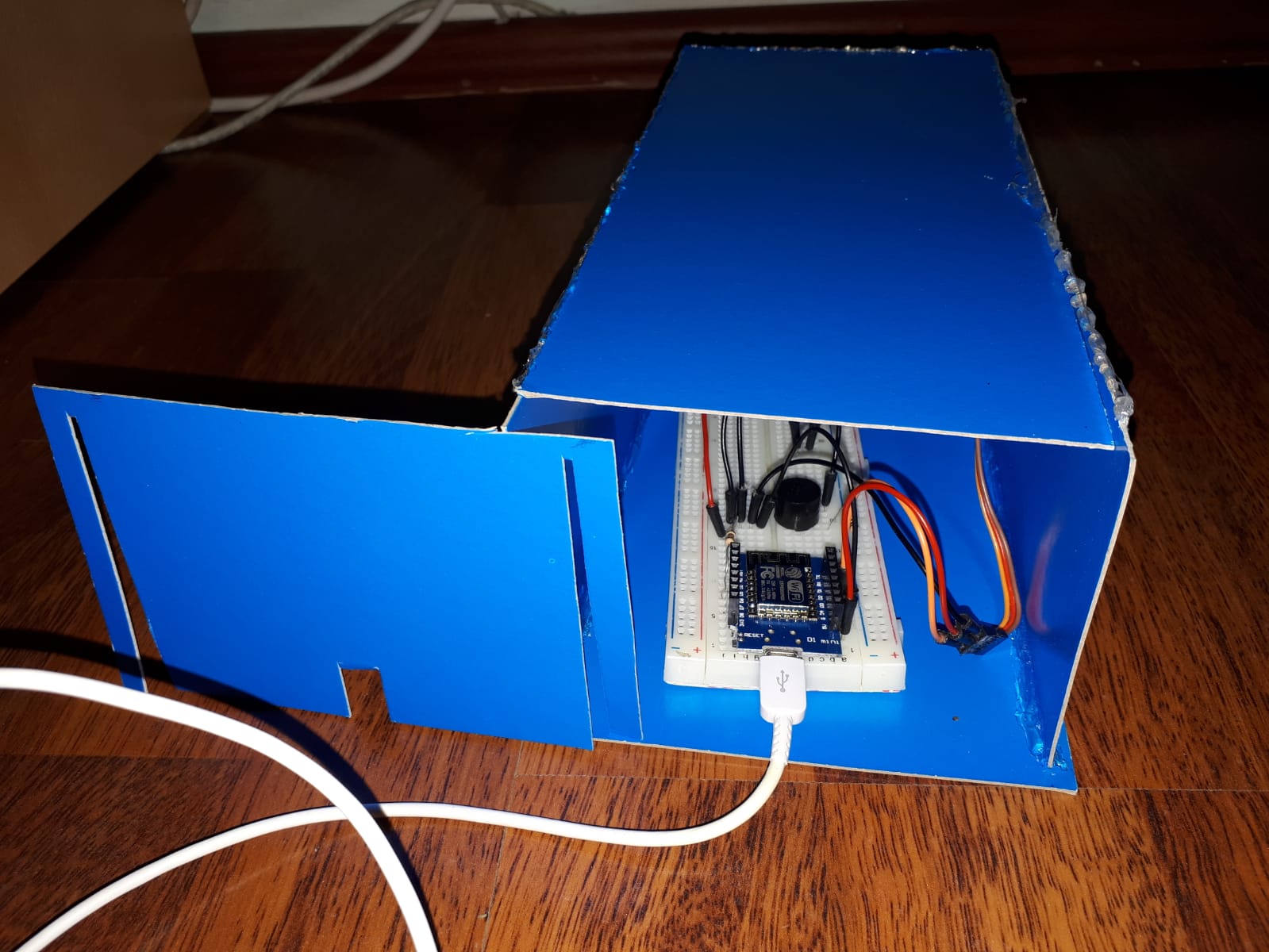
* Garajın Maketi
* Arduino Devresi ve Arduino Programı
* Windows Konsol Uygulaması (Arduino WiFi ayarını arayüz üzerinden yapmak için)
* Android Tabanlı Uygulama
  1. **Garajın Maketi**

Yazdığımız programların çıktısını görebilmek için temsili bir garaj maketi yapılmıştır. Garajın kapısı ve içerisin de led lamba bulunmaktadır. Kontrolu sağlayan Arduino devresi maketin içerisine yerleştirilmiştir. Garajın farklı açılardan resimleri aşağı da listelenmiştir.

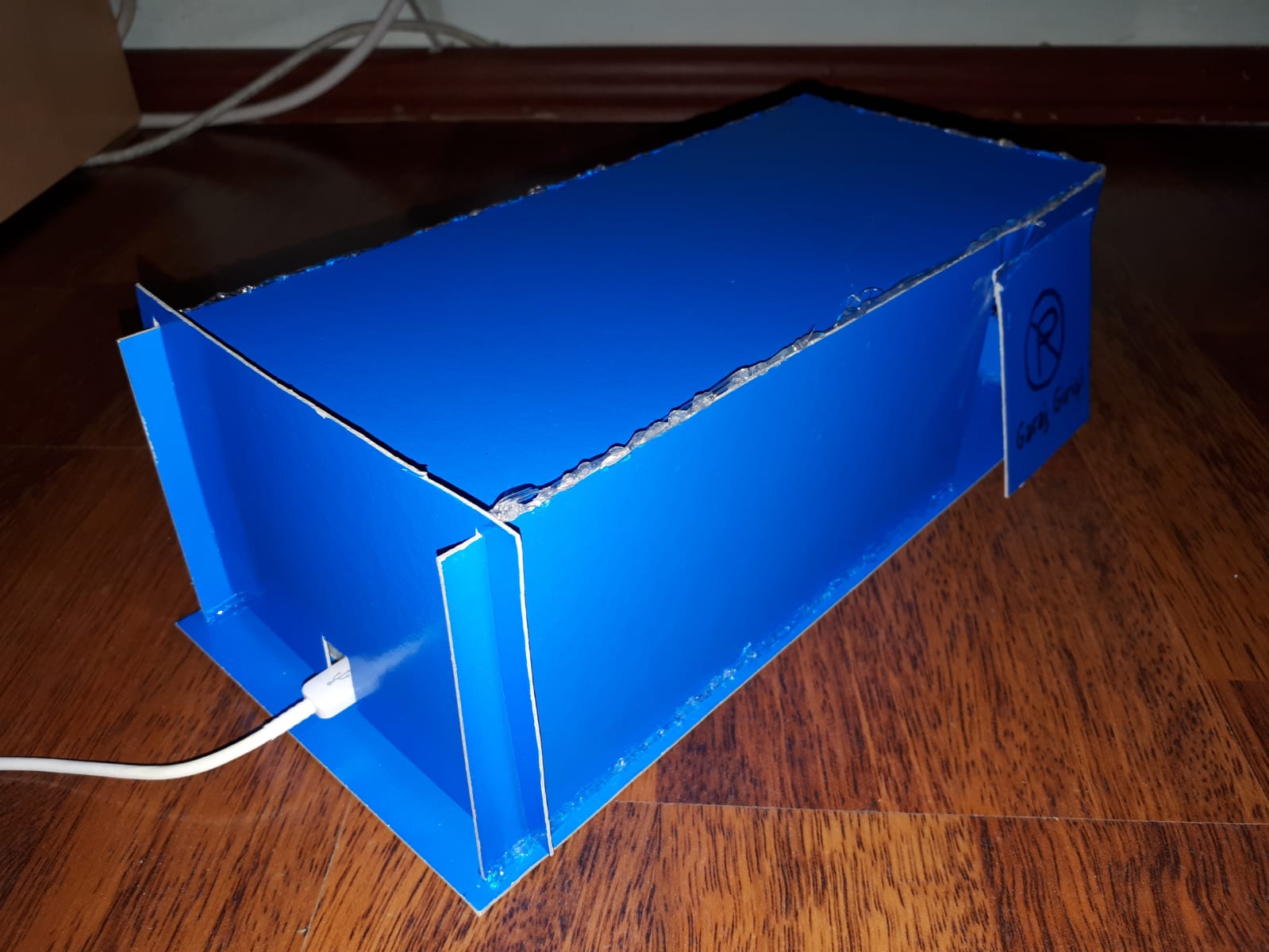
**Resim-7 (Garaj Maketi)**



**Resim-8 (Garaj Maketi)**

****

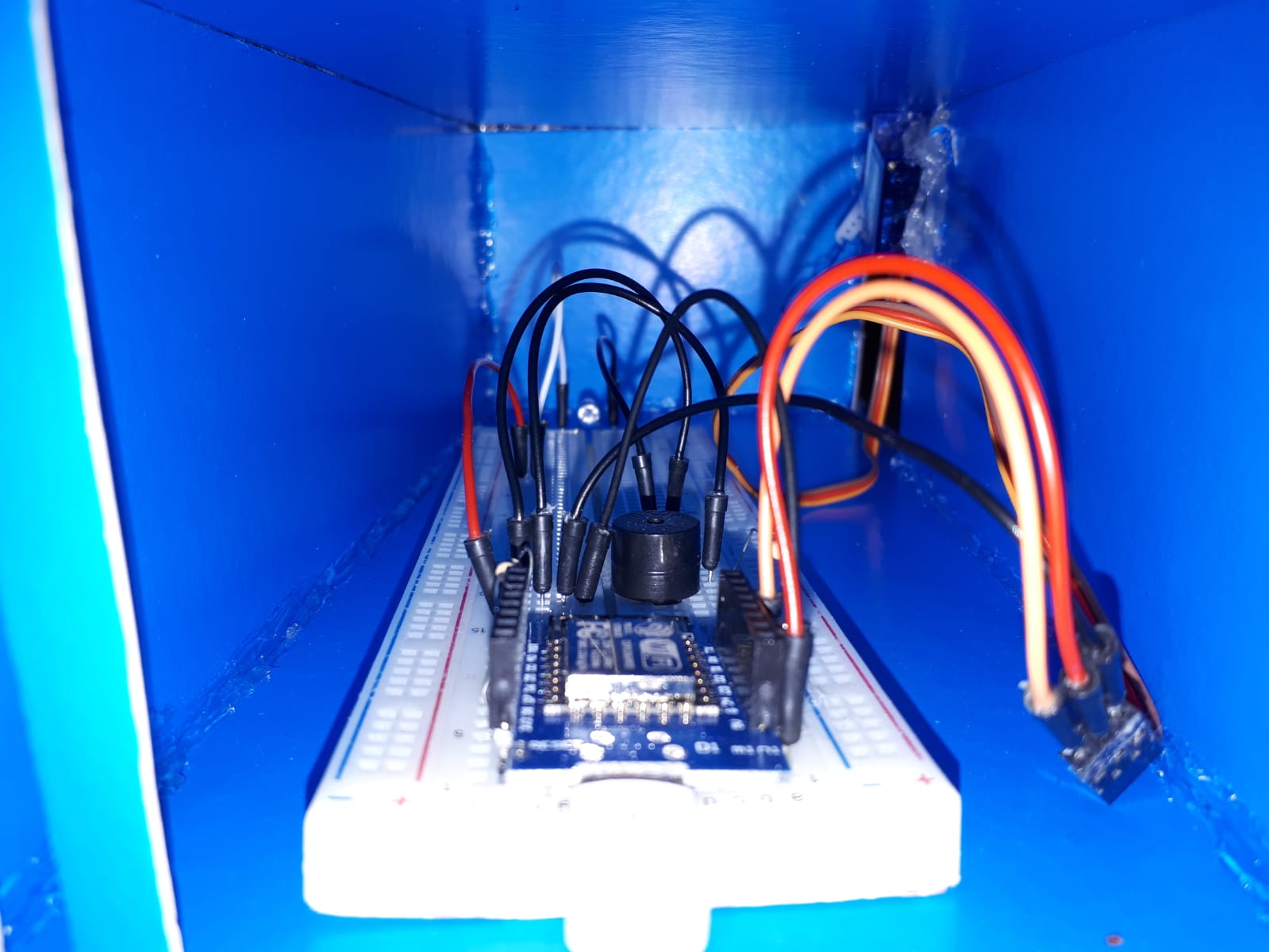
**Resim-9 (Garaj Maketi)**

****

* 1. **Arduino Devresi ve Arduino Programı**

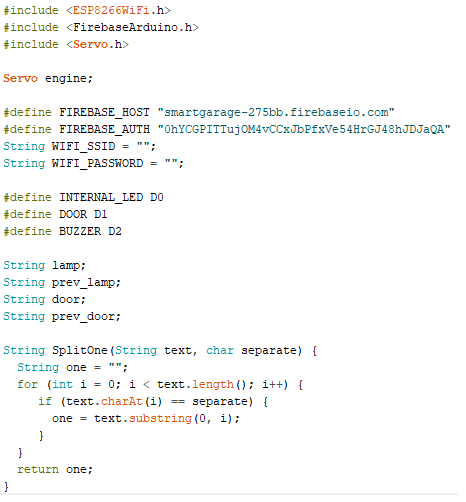
Garaj için kapıyı açıp/kapatma ve lambanın durumunu değiştirme işlemlerini arduinonun yaptığını üst kısımlar da belirtmiştik. Arduino devre bağlantılarını ve bunları kontrol eden kod kısımları aşağıda gösterilmiştir.

**Resim-10 (Arduino Bağlantıları)**

****

Breadboard üzerinde arduino, esp8266, buzzer, servo-motor, led bağlantıları yapılmıştır ve bu devre garaj maketi içine yerleştirilmiştir. (Resim-10)

**Resim-11 (Arduino Kodları)**

****

Bu resimde (Resim-11) gerekli kütüphaneler dahil edilmiş ve Firebase bağlantısı yapılmıştır. Wifi bilgileri burda tanımlanmamıştır çünkü wifi konfigürasyonu için C# uygulaması yapılmıştır. Kapı ve lamba durum değişkenlerini kontrol etmek için değişkenler oluşturulmuştur. SplitOne fonksiyonu C# ile yazılan wifi konfigürasyonundan gelen wifi bağlantı adını döndürmek için kullanılmıştır. C# dan gelen değer bazı karakterlerle geldiği için bunları ayıklama işlemi yapılmıştır.

**Resim-12 (Arduino Kodları)**



SplitTwo fonksiyonu da SplitOne fonksiyonuna benzer olarak wifi şifresini almak için kullanılmıştır. ConnectEthernet fonksiyonun da arduino nun internet bağlantısı gerçekleştirilmiştir. While ile hata olmadığı sürece bağlantı açık kalmıştır. Firebase.begin komutu ile firebase bağlantısı başlatılmıştır.(Resim-12)

**Resim-13 (Arduino Kodları)**

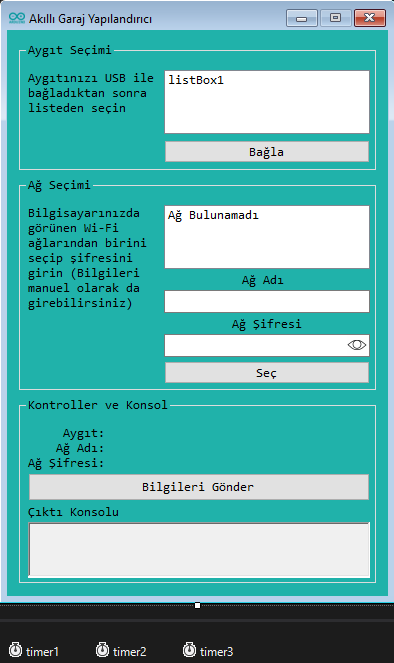
****

SetDoor fonksiyonun da kapı değişkeninin değeri kontrol edilmiş ve eğer “1” e eşit ise Buzzer çalıştırılmış ve kapı servo motor aracılığıyla açılmaya başlanmıştır. Servo motor açı mantığıyla çalıştığı için -50 dereceden 259 dereceye kadar dönmüştür ve kapı açılmıştır. Değer “1” değilse ozaman da kapı kapanmıştır. SetLamp fonksiyonu da aynı mantıkla lamp değeri “1” e eşitse led yakılmıştır “1” e eşit değilse led söndürülmüştür. LampListener fonksiyonu, lamp degeşkenini Firebase den okunan değere eşitler. Gerçek zamanlı veritabanı kullanıldığı için veriler anlık olarak değişmektedir dolayısıyla bir önceki durum prev\_lamp değişkeninde tutulmuştur.(Resim-13)

* 1. **Windows Konsol Uygulaması**

Arduino programında wifi bağlantsını program üzerinden değilde bir arayüz üzerinden yapmak için bu program yapılmıştır. Cihazın bağlı olacağı Wifi adı ve şifresi programdan girildikten sonra arduino bu bilgiler ile internete bağlanabiliyor.(Resim-14)

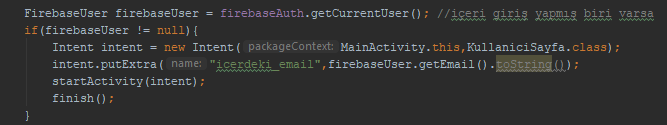
**Resim-14 (Windows Konsol Uygulaması)**

****

* 1. **Android Uygulaması**

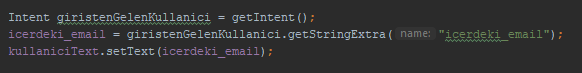
Program ilk çalıştığında MainActivity sınıfı çalışıyor ve sistemde daha önce giriş yapmış bir kullanıcı var ise bu kullanıcının e-mail adresiyle birlikte KullaniciSayfa sınıfına doğru intent başlatılıyor.(Resim-15)

**Resim-15 (Giriş Yapmış Kullanıcı Kontrolu)**



FirebaseUser sınıfından bir nesne oluşturuluyor ve bu nesne getCurrentUser fonksiyonuyla mevcut kullanıcı değerini alıyor. Eğer bu nesne boş değilse yani kullanıcı varsa MainActivity sınıfından KullaniciSayfa sınıfına intent başlatılıyor. KullaniciSayfa sınıfında hangi kullanıcının bulunduğunu kontrol etmek içinse intent.putExtra ile mevcut kullanıcının e-mail adresini de gönderiyoruz.

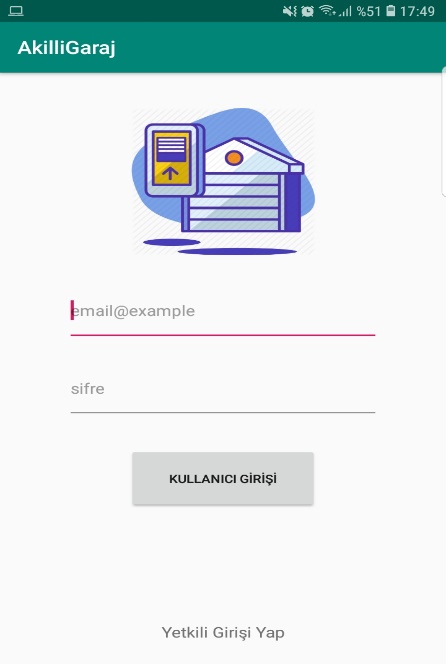
**Resim-16 (Giriş Yapmış Kullanıcı Kontrolu)**



KullaniciSayfa sınıfında MainActivity de verilen anahtar değere sahip kullanıcı alınıyor ve işlemler bu kullanıcı üzerinden oluşturuluyor. (Resim-16)

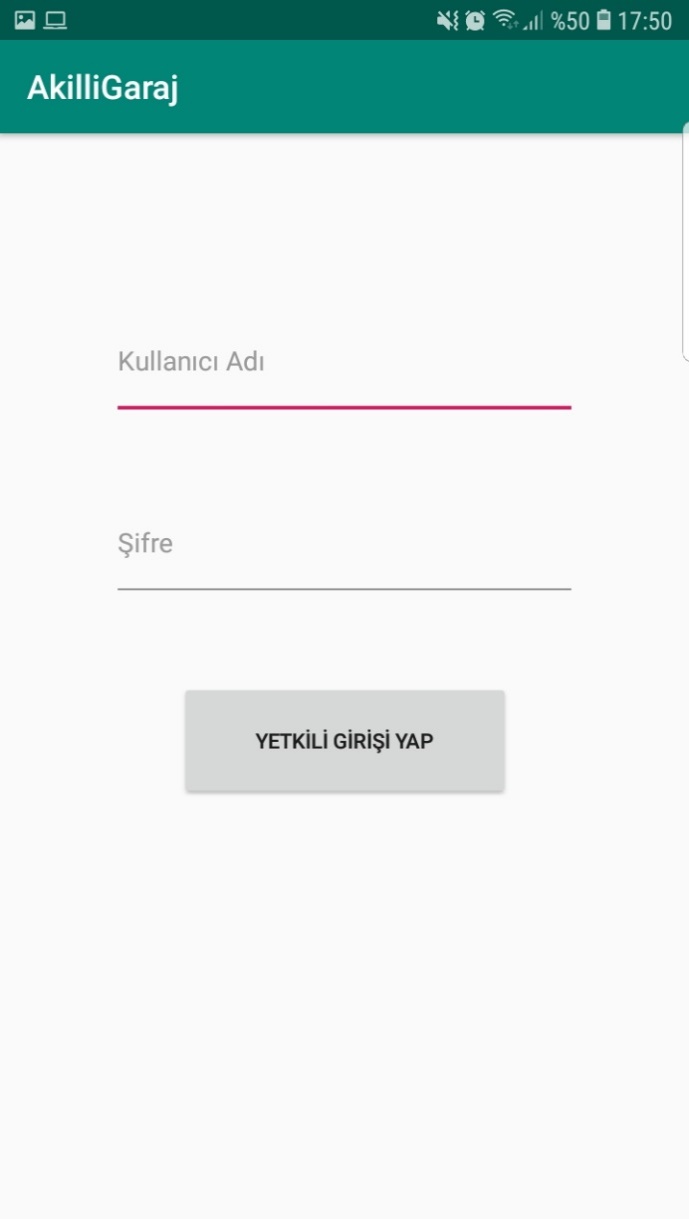
Eğer sistemde mevcut bir kullanıcı yok ise (Resim-17) de ki ekran bizi karşılıyor ve buradan e-mail adresi ve şifre ile sisteme giriş yapılabiliyor. Ayrıca bu sayfada yetkili girişi yapmak için de buton bulunuyor.

**Resim-17 (Ana Sayfa - App)**

****

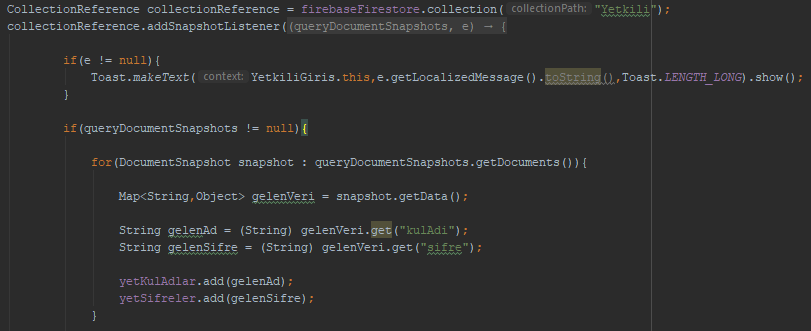
Yetkili girişi butonuna basıldığında (Resim-18) de ki ekran bizi karşılıyor ve eğer girilen bilgiler FireStore da bulunan değerlerle eşleşirse sisteme giriş yapılıyor ve YetkiliSayfa sınıfına yönlendiriyor.

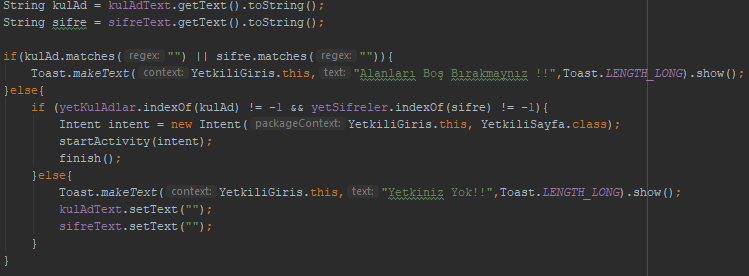
**Resim-18 (Yetkili Giriş - App)**

****

CollectionReferance ile Firestore da ki ‘Yetkili’ koleksiyonundan tüm yetkililer alınıp, adları ve şifreleri ile String dizilerine kayıt ediliyor. Giriş yap butonuna basıldığında ise EditText lere girlen değerler bu dizilerin içinde yer alıyor ise sisteme giriş yapılıyor, yer almıyorsa hata mesajı veriliyor.(Resim-19)

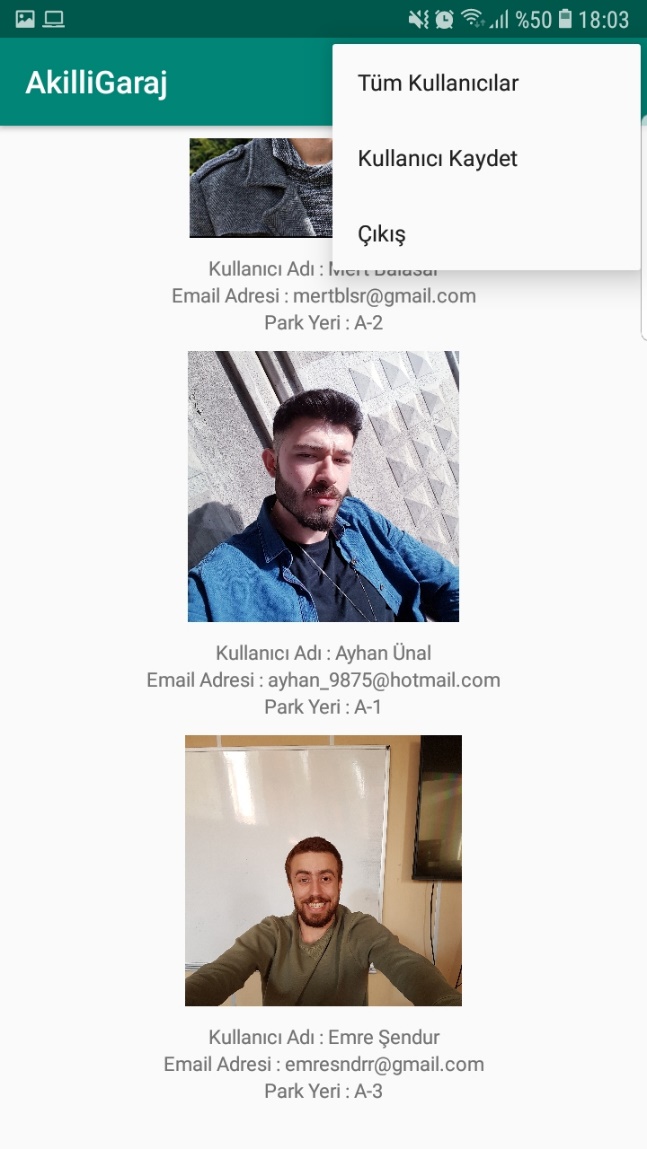
**Resim-19 (Yetkili Giriş - Kod)**





Yetkili sayfasında bir tane RecycleView bulunuyor ve bu listede sistemdeki tüm kullanıcılar ve bilgileri ekranda alt alta gösteriliyor. Sağ üst köşede ki menüden tüm kullanıcıları görüntüleme, yeni kullanıcı kaydetme ve yetkili hesabından çıkış yapma butonları bulunuyor. Tüm kullanıcılar butonuna basıldıgında, yetkilinin sisteme giriş yaptığında karşılaştığı ekran olan yetkili sayfasına yönlendiriyor. Kullanıcı kaydet butonuna basıldığında sisteme yeni kullanıcı kaydetmek üzere ‘YetkiliKullaniciEkle’ sınıfına yönlendiriyor ve yeni kullanıcı kayıt edilebiliyor. Çıkış yap butonu ise yetkilinin hesabından çıkış yapmasını sağlıyor. (Resim-20)

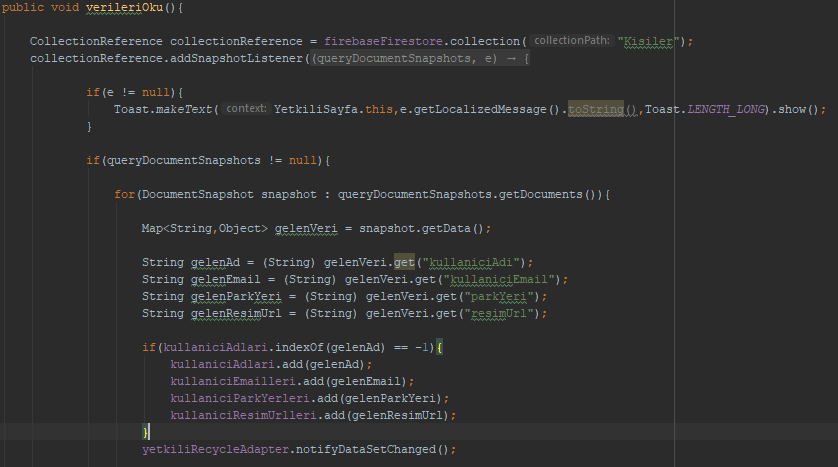
**Resim-20 (Yetkili Sayfa - App)**

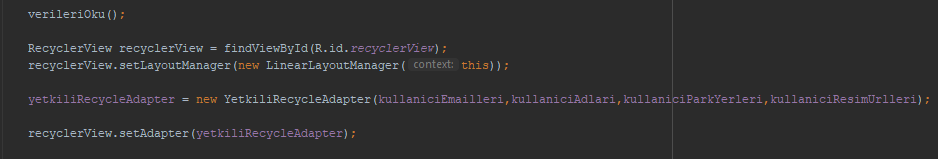
****

Resim-21 de görüldüğü üzere verileriOku adında fonksiyon yazılmıştır. Bu fonksiyon Firestore de bulunan ‘Kisiler’ koleksiyonunda ki kullanıcıları almak için kullanılır. İkinci if bloğu içinde sorgu boştan farklıysa yani sistemde kullanıcı varsa gelen sorgu for döngüsünde tek tek alınıp kullanıcıların adları, emailleri, park yerleri ve resim url leri bunları içeren dizilere kaydedilmiştir. Buna göre ‘kullaniciAdlari’ listesinde sistemde ki tüm kullanıcıların isimleri, ‘kullaniciEmailleri’ listesinde sistemde ki tüm kullanıcıların email adresleri, ‘kullaniciParkYerleri’ listesinde kullanıcıların park yerleri ve ‘kullaniciResimUrlleri’ listesin de ise tüm kullanıcıların resimlerinin url adresleri tutulmaktadır. Bu listelere döngünün aynı iterasyonunda ekleme yapıldığı için her liste için aynı index numarası aynı kullanıcı bilgisini içermektedir.

Oluşturulan bu listeler ile RecyleView in bağlanması için yetkiliRecyleAdapter adında bir sınıf oluşturulmuştur ve bu sınıfta her bir listedeki değerler index numaralarına göre alınıp kullanıcı bilgileriyle eşleştirilmiştir. Sınıfın yapıcı metodunda o kullanıya karşılık gelen bilgiler RecycleView’e set edilmiştir. Bu işlemleri yetkili sayfasında yapabilmemiz için ise OnCreate metodu altında Recycle Adapter kullanılmıştır. (Resim-21)

**Resim-21 (Yetkili Sayfa - Kod)**



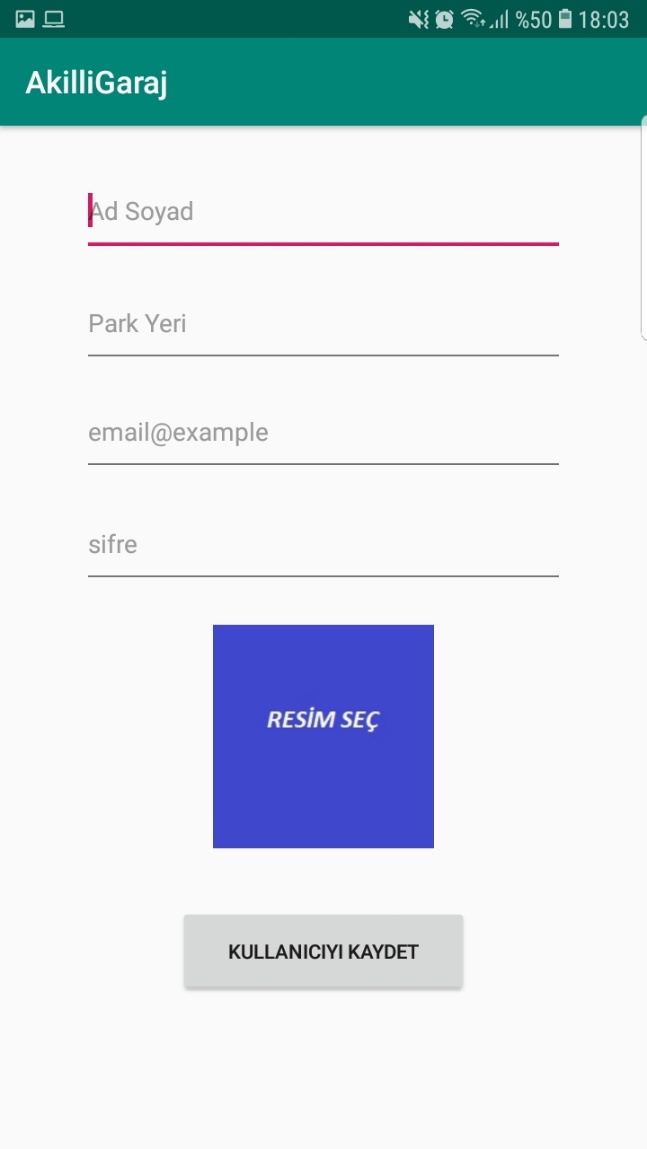


Kullanıcı kaydetme sayfasında yani ‘YetkiliKullaniciEkle’ classında tasarım Resim-22’ de gösterilmiştir.

Yetkili, sisteme ekleyeceği kullanıcı için ad soyad, park yeri, email adresi ve şifre bilgilerini girdikten sonra kullanıcı için galeriden fotograf seçecektir ve tüm alanlar eksiksiz doldurulduğunda

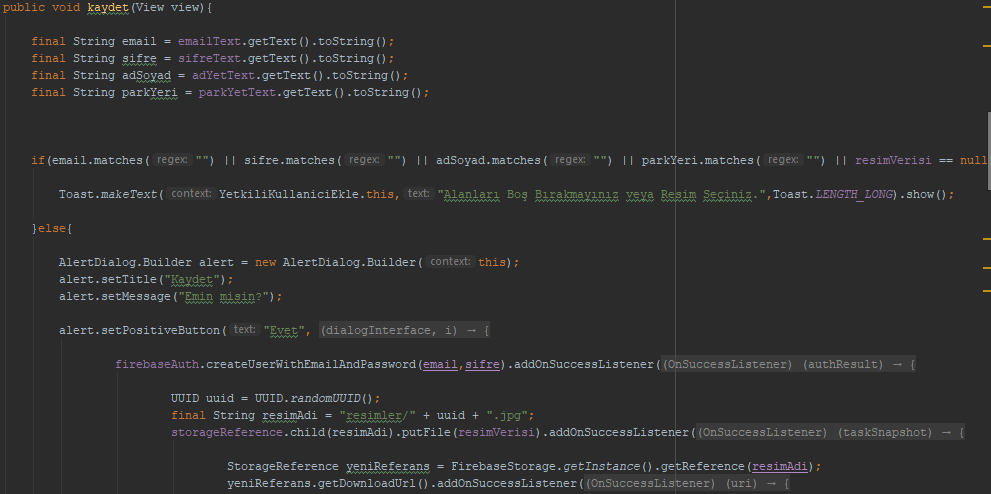
‘Kullanıcıyı Kaydet’ butonuna basacaktır. Eğer bir sorunla karşılaşılmaz ise (aynı kullanıcıyı tekrar ekleme gibi) kullanıcı sisteme eklenecektir ve ekleme işleminden sonra yetkili sayfasına yönlendirilecektir.

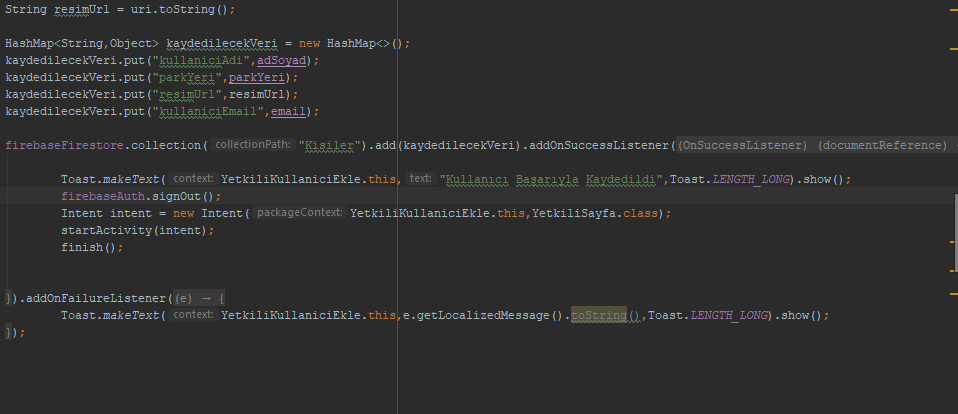
**Resim-22 (Kullanıcı Kaydetme Sayfası - App)**

****

Kaydetme butonuna basıldığında kaydet fonksiyonu çalışacak ve Edit Textlere girilen her bir değer okunup ilgili değişkenlere atanacaktır. İf bloğu içinde girilen degerlerin boş olup olmadığı kontrol edilmiştir ve boş ise Toast mesajı gösterilmiştir. Boş olmadığı durumda kullanıcı kaydedileceltir emin misin gibi uyarı mesajı gösterilecektir ve evet butonuna basıldığında (PositiveButton) öncelikle FirebaseAuthentication sınıfından oluşturulan nesne üzerinden kullanıcı Authentication bölümüne eklenecektir. Ardından kullanıcı için seçilen resim Firebase Strorage de ‘resimler’ dizini altına UUID sınıfından üretilen eşsiz bir isim ile kaydedilecektir. Bu resmin url adresi ‘resimUrl’ değişkeninde tutulmaktadır. Tüm bu işlemlerden sonra kullanıcının bilgileri (adı,parkyeri,email,resim) bir HashMap ‘e eklenip buradan da Firestore de bulunan ‘Kisiler’ koleksiyonuna kaydedilmiştir. 3 aşamalı bu işlemlerin her birine Listener yani dinleyici konulmuştur ve bu sayede işlemlerin başarılı(SuccessListener) ve başarısız (FailureListener) olma durumları için kontrol mekanizması geliştirilmiştir. Daha sonra uygulama, intent ile yetkili sayfasına yönlendirilmiştir. (Resim-23)

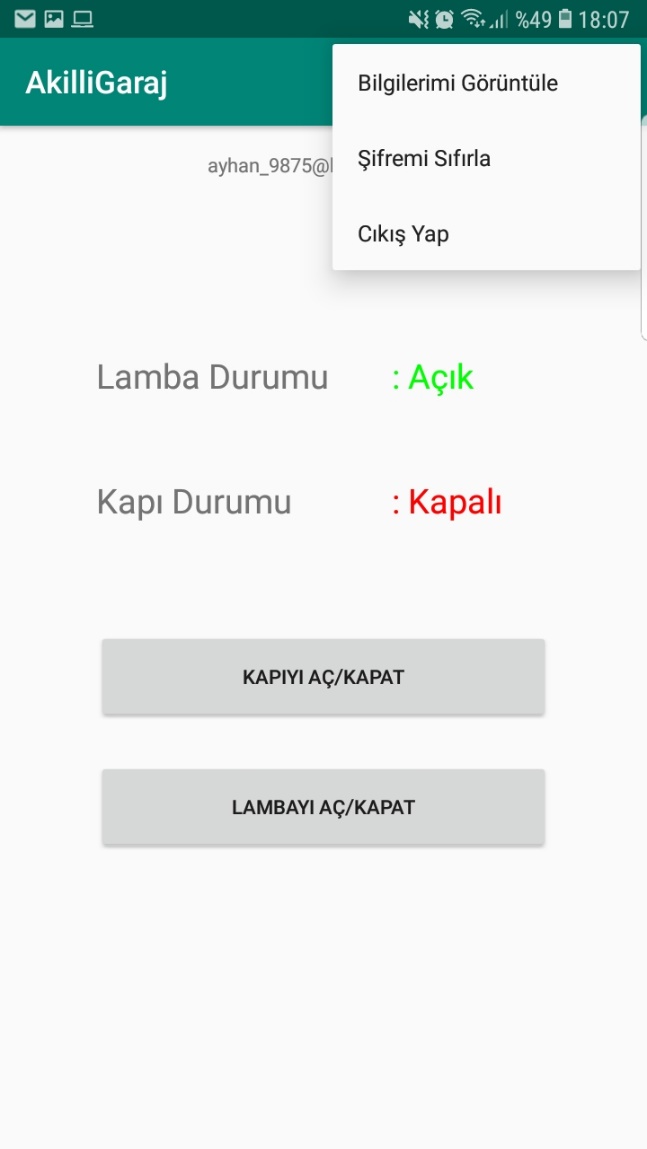
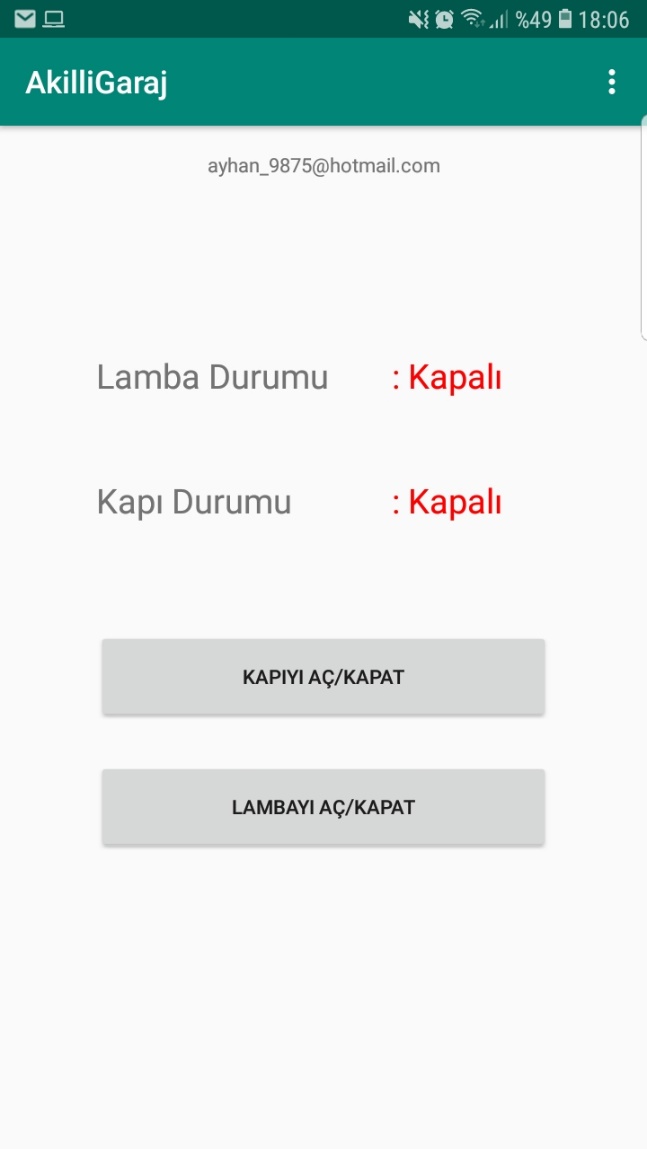
**Resim-23 (Kullanıcı Kaydetme Sayfası - Kod)**





Uygulamanın ana sayfasında kullanıcı girişi yapıldıktan sonra Resim-24 de görülen Kullanıcı Sayfası ekrana çıkmaktadır. Bu sayfada garaj kapısının ve lambasının anlık durumunu gösteren Text View’ler ve bunları değiştirmek için butonlar bulunmaktadır. ‘Kapıyı Aç/Kapat’ yada ‘Lambayı Aç/Kapat’ butonlarına basıldığın da kapı veya lamba kapalı ise açılacak, açık ise kapanacaktır. Ayrıca kullanıcı, sağ üstte bulunan menüden kişisel bilgilerini görüntüleyebilecek, şifresini sıfırlayabilecek ve çıkış yapabilecektir. Eğer kullanıcı çıkış yapmazsa uygulama tekrar kapanıp açıldığında en son giriş yapılan kullanıcı hesabından sisteme giriş yapacaktır dolayısıyla ana sayfa ekranı gelmeden direkt o kullanıcı üzerinden kullanıcı sayfasına yönlenecektir.

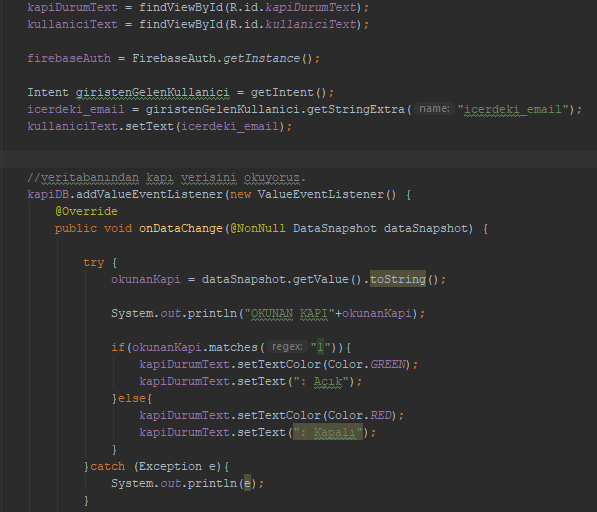
**Resim-24 (Kullanıcı Sayfası - App)**

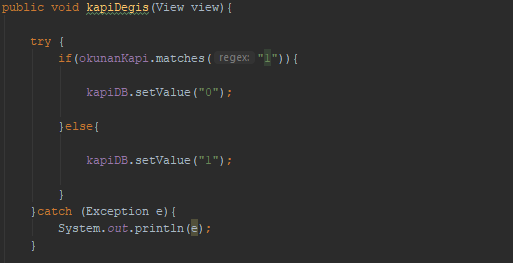
****

onCreate altında, Real Time Database üzerinde bulunan kapı durumu okunuyor ve güncel değer ‘1’ ise Textviewde kapı açık, değilse kapı kapalı yazısı gösteriliyor. Kapıyı aç kapat butonuna basıldığında

kapiDegis fonksiyonu çalışıyor ve okunan değer ‘1’ ise ‘0’ değerini, okunan değer ‘0’ ise ‘1’ değerini veritabanına set ediyor. Veritabanı eşzamanlı çalıştığı için veriler her değiştiğinde okuma işlemi tekrar yapılıyor ve değişim olduğunda TextView da değişiyor. Aynı işlemler lamba için de yapılıyor. (Resim-25)

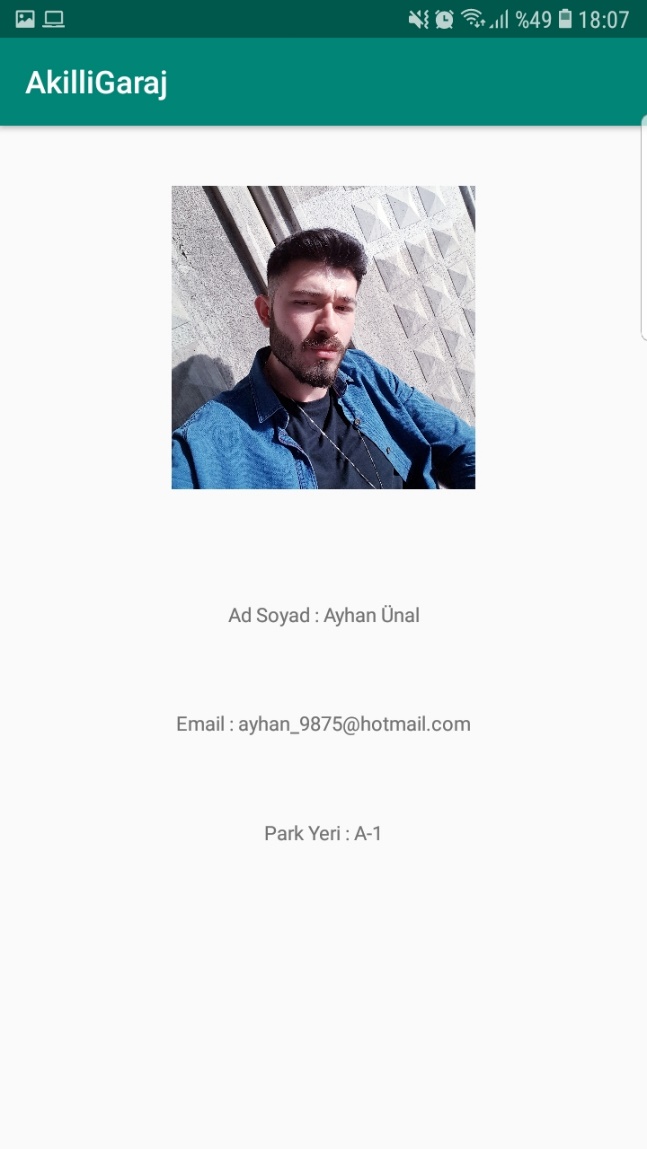
**Resim-25 (Kullanıcı Sayfası - Kod)**





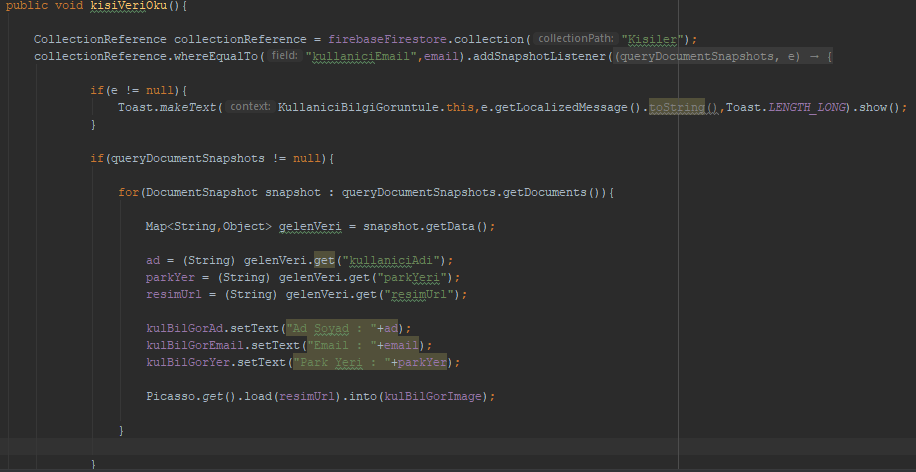
Kullanıcı sayfası üzerinde bulunan menüden Bilgilerimi Görüntüle butonuna basıldığında ‘KullaniciBilgiGoruntule’ sayfası karşılamaktadır. Bu sayfa da Firestore da bulunan kullanıcıların temel bilgileri alınıp ekranda gösteriliyor.(Resim-26)

**Resim-26 (Kullanıcı Bilgi Görüntüleme - App)**

****

‘KullaniciBilgiGoruntule’ sınıfında onCreate metodu aldında çalışan kisiVeriOku fonksiyonu, Firestore da bulunan Kisiler koleksiyonundaki verileri alır ve aynı yetkili sayfa da olduğu gibi listeye ekler. Burada farklı olarak collectionReference objesinin whereEqualTo özelliği kullanılmıştır. Bu özellik sayesinde Firestore da filtreleme yapılmış ve sisteme giriş yapan kullanıcının mail adresine göre veriler çekilmiştir. Dolayısıyla gelen diğer bilgiler; ad soyad, park yeri ve resim url si sisteme giriş yapan kullanıcının bilgileridir. Kullanıcı resimlerinin aslında Storage de bulunduğunu ve Firestore da sadece url adresinin yer aldığını yukarı kısımlarda belirtmiştik. Image View üzerinde bu resmi görüntüleyebilmek için url adresi bulunan resmin bitmap türüne dönüşmesi gerekiyor. Bu yüzden Picasso kütüphanesini projeye dahil ederek url adresi bulunan resim üzerinden indirme işlemi gerçekleştiriyor ve into fonskiyonu sayesinde ilgili Image View üzerinde resmin görüntünlenmesi sağlıyoruz. (Resim-27)

**Resim-27 (Kullanıcı Bilgi Görüntüleme - Kod)**



* 1. **Sistemin Çalışma Şart ve Koşulları**

Sistemin çalışabilmesi için gerekli koşullar aşağıda listelenmiştir.

* Garajın içerisindeki arduino devresinin başlatılabilmesi için 5 voltluk gerilim.(Bizim projemizde arduinonun wifi bağlantısı c# ile yazdığımız konsol uygulaması ile sağlanmıştır. Dolayısıyla konsol uygulamasının çalıştırılacağı bilgisayar gerekmektedir. Ancak wifi bilgileri kodlanma esnasında manuel olarak girilirse bilgisayara ihtiyaç yoktur.)
* Garajın kontrollerini yönetmek için android işletim sistemli(Android 6.0 ve üstü) ve internet erişimi olan bir cihaz.
  1. **Karşılaşılan Zorluk ve Problemler**

Proje süresince birçok problemle ve zorlukla karşılaşılmıştır. Bunlar aşağıda belirtilmiştir.

* İlk dönem yapmış olduğumuz garaj maketi bir kaza sonucu kullanılamaz hale geldi ve yeni garaj maketi yapmak zorunda kaldık.
* Projeyi 2 kişi birlikte yaptık ve malum pandemi sounucunda bu takım çalışmasını uzaktan, kendi evlerimizden senkron bir biçimde gerçekleştirmek zorunda kaldık.
* Pandemi sonucunda bazı elektronik malzemelerin temini konusunda sorunlar yaşadık ve planladığımız proje içeriğinde bir takım değişikliklere gitmek zorunda kaldık.
* Kullanıcıların garaja giriş-çıkış saatlerini görme özelliğini de uygulamaya eklemiştik fakat uygulamaya giriş yapan her kullanıcı Firebase deki aynı veritabanı üzerinde değişiklik yaptığı için bunun bilgisine erişemedik(Firebase engelinden ötürü). Dolayısıyla bu özelliği çıkarmak zorunda kaldık(1. Dönemde bu özelliği bluetooth sensörü sayesinde eklemeyi düşünmüştük fakat pandemi sonucu sensörü temin edemedik).
  1. **Sistemin Teknik Özellikleri**
* Esp Model : ESP8266 D1 Mini NodeMCU
* Mikrodenetleyici : ESP-8266EX
* Saat Hızı : 80/160 MHz
* Program Hafızası : 4 MB
* Analog Giriş Pini Sayısı : 1
* Dijital Giriş/Çıkış Pin Sayısı : 11
* Wi-Fi: 3 (802.11 b/g/n)
* Wi-Fi Frekansı: 2.4 GHz
* Güç/Veri Girişi: USB Type-B (Micro USB)
* USB Baud Rate: 9600 bps (bit per second)
* Android Min. SDK : 23 (Android 6.0)
* Firebase Cloud Firestore – Max Veri : 1 GB
* Firebase Cloud Firestore – Doküman Okuma : 50K/Gün
* Firebase Cloud Firestore – Doküman Yazma ve Silme : 20K/Gün
* Firebase Realtime Database – Eşzamanlı Bağlantı : 100
* Firebase Storage – Depolama : 5 GB
* Firebase Storage – Veri İndirme : 50K/Gün – Totalde : 1 GB/gün
* Firebase Storage – Veri Yükleme : 20K/Gün

1. **Öneriler ve Referanslar**

[1] G. Demiris and B. Hensel, “Technologies for an Aging Society : A Systematic Review of ‘ Smart Home ’ Applications,” 2007.

[2] W. Bolton, Programmable Logic Controllers. 2006.

[3] N. Chatterjee, S. Chakraborty, A. Decosta, and A. Nath, “Real-time Communication Application Based on Android Using Google Firebase,” Int. J. Adv. Res. Comput. Sci. Manag. Stud., 2018.