



Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü
Gömülü Sistemleri

| Uygulama Konusu |
|--|
| ESP8266 Tabanlı RFID ve Uzaktan Mobil Kontrol Destekli Turnike Geçiş Sistemi |

| Öğrenci Bilgileri | |
|-------------------|------------------|
| Öğr. No | 21100011039 |
| Ad Soyad | Ömer Faruk Akbaş |

Dersin Hocası
Doç. Dr. Muhammed KARAALTUN

Mayıs 2025
Konya

İçindekiler

| | |
|--|----|
| 1. GİRİŞ..... | 4 |
| 2. DONANIM VE YAZILIM BİLEŞENLERİ | 5 |
| 2.1 Donanım Bileşenleri | 5 |
| 2.1.1 ESP8266 (NodeMCU)..... | 5 |
| 2.1.2 RC522 RFID Modül | 5 |
| 2.1.3 RFID Kart (MIFARE Classic) | 5 |
| 2.1.4 SG90 Servo Motor | 5 |
| 2.1.5 Arduino Uno | 5 |
| 2.1.6 16x2 LCD Ekran..... | 5 |
| 2.1.7 EEPROM (ESP8266 dâhili)..... | 6 |
| 2.1.9 Breadboard ve Jumper Kablolar | 6 |
| 2.2 Yazılım Bileşenleri..... | 6 |
| 2.2.1 Arduino IDE | 6 |
| 2.2.2 Firebase Realtime Database | 6 |
| 2.2.3 Firebase Authentication | 6 |
| 2.2.4 Flutter | 6 |
| 2.2.5 LiquidCrystal.h..... | 6 |
| 2.2.6 MFRC522.h..... | 7 |
| 2.2.7 FirebaseESP8266.h..... | 7 |
| 2.2.8 EEPROM.h | 7 |
| 2.2.9 NTPClient.h | 7 |
| 2.2.10 Yazılımda Kullanılan Komut ve Fonksiyonların Açıklamaları | 7 |
| 3. YÖNTEM VE UYGULAMA..... | 8 |
| 3.1 Sistem Mimarisi..... | 8 |
| 3.2 Mobil Uygulama İşleyişi | 8 |
| 3.2.1 Giriş Ekranı | 9 |
| 3.2.2 Kayıt Olma Ekranı | 9 |
| 3.2.3 Admin Paneli..... | 10 |
| 3.2.4 Turnike Kontrol Paneli..... | 10 |
| 3.2.5 Kullanıcı Paneli..... | 11 |
| 3.3 Devre Şeması..... | 12 |
| 3.4 Firebase Veri Yapısı..... | 12 |
| 3.5 Sistem Kesme ve Yeniden Başlatma Mekanizması | 13 |
| 4. Sonuç ve Değerlendirme | 14 |
| 5. KAYNAKLAR..... | 14 |

| | |
|--|-----------|
| 6.Ekler | 15 |
| 6.1 Turnike Sistemi ESP8266 Kodu..... | 15 |
| 6.2 Özel Olarak Yazılan TurnikeSistemi Kütüphanesi..... | 15 |
| 6.2.1 TurnikeSistemi.cpp | 15 |
| 6.2.2 TurnikeSistemi.h | 15 |
| 6.3 Turnike Sistemi Arduino Kodu..... | 15 |
| 6.4 Mobil Uygulama (Flutter) Kodları | 15 |

Öz

Kartlı geiş sistemleri günümüzde yaygın olarak kullanılmakla birlikte, kartın yanınızda olmaması durumunda erişim sorunları yaşanabilmektedir. Bu alışmada, bu probleme özüm olarak “ESP8266 Tabanlı RFID ve Uzaktan Mobil Kontrol Destekli Turnike Geiş Sistemi” tasarlanmış ve uygulanmıştır. Sistem; ESP8266 mikrodnetleyici, RC522 RFID okuyucu, servo motor, EEPROM tabanlı geiş kısıtlaması ve Firebase Realtime Database’ten oluşmaktadır. Kullanıcılar fiziksel kart ile geiş yapabildiği gibi, Flutter tabanlı mobil uygulama üzerinden hesaplarına giriş yaparak turnikeyi uzaktan açabilmektedir. Tüm kullanıcı bilgileri, geiş kayıtları, günlük saya verileri ve kontrol komutları Firebase üzerinde saklanmakta; admin paneli üzerinden yetki yönetimi ve uzaktan komut gönderimi sağlanmaktadır. Gerekleştirilen prototip, mobil uygulama ve fiziksel kart ile güvenli, anlık ve esnek geiş imkânı sunarak erişim kontrolünde güvenliği ve kullanılabilirliği artırmaktadır.

1.GİRİŞ

Günümüzde otomasyon sistemleri, gelişen mikrodnetleyici teknolojileri ve kablosuz iletişim altyapıları sayesinde enerji yönetimi, erişim kontrolü ve güvenlik uygulamalarında önemli roller üstlenmektedir [1]. Özellikle erişim kontrol sistemleri; kamu kurumları, kampüsler ve özel alanlarda güvenli, izlenebilir ve kullanıcı dostu geiş imkanları sunmaktadır. Bu sistemlerde yaygın olarak kullanılan RFID (Radio-Frequency Identification) teknolojisi, fiziksel temas olmaksızın hızlı ve güvenli geiş olanağı sağlamaktadır [2].

Nesnelerin İnterneti (IoT) temelli özüm, Wi-Fi özellikli mikrodnetleyiciler (ör. ESP8266) ile RFID modüllerini birleştirerek, mobil uygulamalar aracılığıyla uzaktan kontrol, kullanıcı yönetimi ve bulut tabanlı veri saklama gibi yeni işlevlerin gerekleştirilmesine olanak tanımaktadır [3], [4]. Firebase gibi gerek zamanlı veri tabanı servisleriyle birlikte çalışan mobil uygulamalar, kullanıcı bazlı yetkilendirme, geiş loglama ve sistem durumu izleme gibi süreçleri merkezi ve güvenli bir şekilde yönetebilmektedir [5]. Firebase Realtime Database’in hiyerarşik veri yapısı sayesinde, her bir kullanıcının erişim geçmişi, günlük sayaları ve son işlem bilgisi ayrı ayrı tutulabilmekte; bu da sistem yöneticilerine anlık analiz, hızlı müdahale ve yetkisiz erişimi izleme imkânı sunmaktadır. Ayrıca Firebase’in kimlik doğrulama (authentication) ve erişim izinleri (security rules) ile desteklenen yapısı, sadece yetkili kullanıcıların verilere erişebilmesini garanti altına almakta ve tüm log kayıtlarını bulut ortamında senkronize şekilde saklamaktadır [6]. Böylece kullanıcı geişlerinin güvenli biçimde izlenmesi, kayıt dışı müdahalelerin önlenmesi ve geçmişe dönük raporlamaların yapılması mümkün hale gelmektedir.

Tasarımı yapılan sistemde, kullanıcılar Flutter tabanlı mobil uygulama aracılığıyla RFID ya da NFC yoluyla kart UID’sini okuyarak kolayca kayıt olabilmekte, yeni kayıtlar admin paneline devre dışı olarak yansıtılmakta ve yalnızca yetkili admin tarafından “aktif” hale getirildikten sonra geiş hakkı kazanmaktadır. Girişler, hem fiziksel RFID kart ile hem de mobil uygulama ile yapılabilmekte, ESP8266 üzerinden UART aracılığıyla Arduino’ya aktarılan “LCD:” etiketli seri mesajlarla kullanıcı anlık olarak bilgilendirilmektedir [7]. Ayrıca EEPROM tabanlı 30 saniyelik geiş kısıtlaması ve Firebase üzerinden gönderilen “turnike aktif/devre dışı” ve “dik kalsın” gibi komutlar sistem güvenliğini arttırmakta ve yetkisiz geişleri önlemektedir [8].

Bu alışmada, “ESP8266 Tabanlı RFID ve Uzaktan Mobil Kontrol Destekli Turnike Geiş Sistemi” tasarlanmış ve uygulanmıştır. İkinci bölümde sistemin donanım ve yazılım bileşenleri,

üçüncü bölümde devre şeması ve sistemin akış diyagramı, dördüncü bölümde deneysel sonuçlar, beşinci bölümde ise genel değerlendirme ve öneriler sunulmaktadır.

2. DONANIM VE YAZILIM BİLEŞENLERİ

2.1 Donanım Bileşenleri

2.1.1 ESP8266 (NodeMCU)

ESP8266, düşük maliyetli ve Wi-Fi destekli bir mikrodenetleyicidir. NodeMCU geliştirme kartı üzerinde kullanılmaktadır. Projede merkezi kontrol birimi olarak görev almıştır. Firebase Realtime Database ile doğrudan bağlantı kurarak kullanıcı verilerini alır, yazar ve turnike kontrol sinyallerini üretir. Kart, Arduino IDE kullanılarak programlanmıştır. GPIO pinleri üzerinden RFID okuyucu, servo motor ve UART haberleşmesi için Arduino ile bağlantı kurulmuştur.

2.1.2 RC522 RFID Modül

RC522, 13.56 MHz frekansında çalışan bir RFID okuyucu modüldür. SPI protokolü üzerinden ESP8266'ya bağlanmıştır. Projede kullanıcı kartlarının UID bilgisini okuyarak turnike geçiş kontrolünde kimlik tanımlama amacıyla kullanılmıştır. Okunan UID Firebase'e gönderilmiş ve geçiş yetkisi kontrol edilmiştir.

2.1.3 RFID Kart (MIFARE Classic)

Her kullanıcının kendine ait fiziksel kartıdır. 4 baytlık benzersiz UID içeren MIFARE kartları kullanılmaktadır. RC522 modül tarafından okutulmakta ve Firebase'deki yetkili kart listesiyle karşılaştırılarak geçişe izin verilmektedir.

2.1.4 SG90 Servo Motor

SG90 küçük boyutlu ve 180° dönüş açısına sahip bir servo motordur. ESP8266 üzerinden PWM sinyali ile kontrol edilmiştir. Geçiş yetkisi onaylanan kullanıcılar için 0° konumuna geçerek turnike açılmış, 5 saniye sonra 90° konumuna dönerek kapatılmıştır. "Turnike dik kalsın" komutu aktifse sürekli açık bırakılmaktadır.

2.1.5 Arduino Uno

Arduino Uno, projede hem LCD ekran üzerinden kullanıcı bilgilendirmesi yapmak hem de 5V çıkışıyla servo motoru beslemek amacıyla kullanılmıştır. ESP8266'dan gelen "LCD:..." formatındaki UART seri mesajları alarak, LiquidCrystal.h kütüphanesi ile 16x2 LCD ekrana iletmıştır.

2.1.6 16x2 LCD Ekran

Kullanıcıya "Kart okundu", "Yetkisiz kart", "Giriş yapıldı" gibi durum bilgilerini vermek için kullanılmıştır. Arduino ile paralel bağlantı üzerinden haberleşmiştir. LiquidCrystal kütüphanesi ile kontrol edilmiştir. Gerçek zamanlı bildirim sağlanmıştır.

2.1.7 EEPROM (ESP8266 dâhili)

ESP8266'nın dâhili EEPROM belleği, her kullanıcının son geçiş zamanını saklamak için kullanılmıştır. Böylece aynı kartla 30 saniye içinde tekrar geçiş yapılması engellenmiştir. EEPROM'a UID hash değeriyle adres belirlenmiş, zaman damgası olarak epoch time yazılmıştır.

2.1.9 Breadboard ve Jumper Kablolar

Tüm devre geçici olarak breadboard üzerinde kurulmuştur. Bağlantılar jumper kablolar ile yapılmıştır. Böylece sistem kolayca test edilmiş, devre değişiklikleri hızlı bir şekilde uygulanabilmiştir.

2.2 Yazılım Bileşenleri

2.2.1 Arduino IDE

Arduino IDE, ESP8266 ve Arduino Uno gibi mikrodenetleyicilerin programlanmasında kullanılan açık kaynaklı bir geliştirme ortamıdır. Projede hem ESP8266'yı hem de Arduino'yu kodlamak için kullanılmıştır. C/C++ tabanlı bu ortamda kodlar yazılarak her iki karta ayrı ayrı yüklenmiştir.

2.2.2 Firebase Realtime Database

Firebase RTDB, verilerin JSON formatında tutulduğu ve gerçek zamanlı olarak senkronize edildiği bir bulut veri tabanı servisi. Projede kullanıcı bilgileri, geçiş logları, günlük sayaçlar ve sistem durumları bu veritabanında tutulmaktadır. Aynı zamanda mobil uygulama ve ESP8266 bu veritabanı üzerinden haberleşmektedir.

2.2.3 Firebase Authentication

Kullanıcıların mobil uygulamaya giriş yapabilmesi için Firebase Authentication servisi kullanılmıştır. Bu servis, e-posta/şifre doğrulama yöntemi ile kimlik denetimi sağlar. Böylece yetkisiz erişim önlenmiş olur.

2.2.4 Flutter

Flutter, Google tarafından geliştirilen bir mobil uygulama çatısıdır. Projede hem admin hem de normal kullanıcıların sisteme giriş yapabileceği, kart UID'si okuyabileceği ve turnike kontrol edebileceği mobil arayüzler Flutter ile hazırlanmıştır.

2.2.5 LiquidCrystal.h

Bu kütüphane, Arduino üzerinden 16x2 LCD ekran kontrolü için kullanılmıştır. ESP8266'dan gelen seri veriler Arduino tarafından alınarak LCD'ye yazdırılmıştır.

2.2.6 MFRC522.h

MFRC522 RFID modülünün ESP8266 ile haberleşmesini sağlayan kütüphanedir. Kullanıcıların kart UID'lerini SPI protokolü ile okumaya olanak tanır.

2.2.7 FirebaseESP8266.h

ESP8266 kartının Firebase veritabanı ile doğrudan haberleşmesini sağlar. Veri okuma, yazma, güncelleme ve dinleme gibi işlemler bu kütüphane ile yapılmıştır.

2.2.8 EEPROM.h

ESP8266'nın dahili EEPROM belleğine veri yazmak ve okumak için kullanılmıştır. Projede UID bazlı geçiş zamanları burada saklanarak "30 saniyelik tekrar geçiş yasağı" uygulanmıştır.

2.2.9 NTPClient.h

İnternet üzerinden epoch zaman bilgisini almak için kullanılmıştır. Geçiş zamanlarını hassas ve senkronize tutmak amacıyla kullanılmıştır.

2.2.10 Yazılımda Kullanılan Komut ve Fonksiyonların Açıklamaları

| Komut / Fonksiyon Adı | Açıklama ve Kullanım Amacı |
|--|---|
| attachInterrupt() | D1 pinine bağlı butondan gelen kesme sinyalini yakalayıp sistemin yeniden başlatılmasını sağlar. |
| os_timer_setfn() / os_timer_arm() | ESP8266 donanımsal zamanlayıcısını her 1 saniyede bir zamanFonksiyonu()'nu tetikleyecek şekilde yapılandırır. |
| EEPROM.get() / EEPROM.put() | UID'nin son geçiş zamanını belleğe yazar ve okur. Aynı kartla sık aralıklarla geçiş yapılmasını engellemek için kullanılır. |
| Servo.attach() / Servo.write() | Servo motorun ilgili pine bağlanmasını sağlar ve belirli bir açıya dönmesini kontrol eder. |
| Firebase.getBool() / getString() / setInt() | Firebase Realtime Database üzerinde veri okuma, yazma ve güncelleme işlemlerinde kullanılır. |
| Firebase.pushJSON() | Firebase veri tabanına geçiş olaylarını (log) kaydeden fonksiyondur. |
| timeClient.update() | NTPClient üzerinden internetten zaman bilgisini günceller, geçişlere zaman damgası eklenmesini sağlar. |
| Lcd.setCursor() / Lcd.print() | LCD ekranın konumunu ayarlamak ve kullanıcıya bilgi mesajları yazdırmak için kullanılır. |

| Komut / Fonksiyon Adı | Açıklama ve Kullanım Amacı |
|-------------------------------------|---|
| PICC_ReadCardSerial() / uid.uidByte | RC522 RFID modülü ile kart UID'sini okumak için kullanılır. UID bilgisi string olarak işlenir. |
| Fonksiyon Adı | Açıklama |
| turnike_kontrol() | Turnikeyi açar, turnike_dik durumuna göre otomatik kapanıp kapanmayacağına karar verir. |
| log_ekle() | Kullanıcının UID'si, adı, geçiş tipi (GİRİŞ/ÇIKIŞ), zaman ve kaynak bilgisi ile birlikte Firebase'e log kaydı oluşturur. |
| kart_yetkilimi() | Okunan kart UID'sinin Firebase'deki kullanıcılar verisiyle eşleşip eşleşmediğini ve yetkili olup olmadığını kontrol eder. |
| gecis_zamani_kontrol() | EEPROM'da saklanan UID'ye ait son geçiş zamanı ile şimdiki zamanı karşılaştırır; 30 saniye sınırını kontrol eder. |
| getUID() | RC522 ile okunan UID'yi string haline dönüştürür ve işler. |
| bugun_str() | Günün tarihini YYYYMMDD formatında string olarak döndürür. Sayaç işlemlerinde tarih tabanlı kullanılır. |

3. YÖNTEM VE UYGULAMA

3.1 Sistem Mimarisi

Geliştirilen sistem; ESP8266 mikrodenetleyicisi, RC522 RFID okuyucu, SG90 servo motor, Arduino üzerinden bağlı 16x2 LCD ekran, Firebase Realtime Database ve Flutter tabanlı mobil uygulama bileşenlerinden oluşmaktadır. Sistemin temel işleyişi; kullanıcının fiziksel kartını RFID okuyucuya okutması veya mobil uygulama üzerinden uzaktan turnike açma komutu göndermesiyle başlar. Her iki senaryoda da ESP8266, Firebase veritabanına bağlanarak ilgili kullanıcının geçiş yetkisini sorgular.

Geçiş onayı alındığında servo motor hareket ederek turnike fiziksel olarak açılır. Aynı anda, ESP8266'dan Arduino'ya UART üzerinden geçiş durumu bilgisi gönderilir. Arduino, gelen mesajları LCD ekran aracılığıyla kullanıcıya iletir. Tüm geçiş işlemleri, geçerli zaman bilgisiyle birlikte Firebase veritabanına loglanır. Sistemde ayrıca EEPROM kullanılarak, aynı kartın kısa sürede tekrar okutulması engellenmektedir.

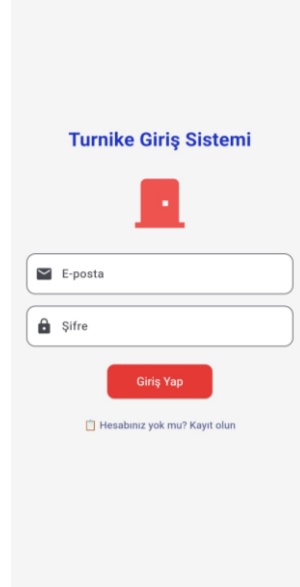
Bu mimari yapı sayesinde sistem, hem yerel donanım üzerinden hem de internet bağlantısı olan herhangi bir cihazla uzaktan izlenebilir ve yönetilebilir hale gelmiştir.

3.2 Mobil Uygulama İşleyişi

Sistemin mobil uygulama arayüzü, Flutter ile geliştirilmiş olup hem kullanıcılar hem de yöneticiler için ayrı işlevler sunmaktadır. Uygulama Firebase Authentication ile kullanıcı kimlik doğrulamasını sağlamakta, Firebase Realtime Database üzerinden veri okuma/yazma işlemleri yapmaktadır. Uygulama üzerinde temel olarak şu ekranlar ve işlevler yer almaktadır:

3.2.1 Giriş Ekranı

Kullanıcılar ve yöneticiler bu ekran üzerinden e-posta ve şifre ile sisteme güvenli bir şekilde giriş yapabilmektedir. Giriş türüne göre kullanıcı, kendi paneline ya da admin kontrol paneline yönlendirilir. Firebase Authentication servisi ile doğrulama sağlanır. Bu ekran Şekil 1’de gösterilmiştir.

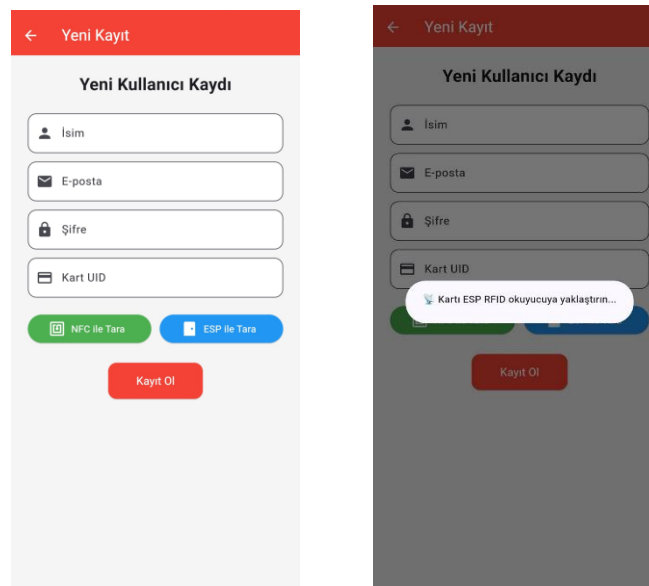


Şekil 1. Turnike Giriş Sistemi Ekranı

3.2.2 Kayıt Olma Ekranı

Yeni kullanıcı, isim, e-posta ve şifresini girdikten sonra kart UID’sini iki farklı yolla okutabilir: Telefonu NFC destekliyorsa doğrudan “NFC ile Tara” seçeneğini kullanabilir.

Alternatif olarak ESP8266 üzerindeki RFID okuyucu ile kartı okutarak “ESP ile Tara” butonuna basabilir. Sistem Firebase üzerindeki /tarama düğümünü izleyerek otomatik UID çekimi yapar. Bu ekran Şekil 2’de gösterilmiştir.



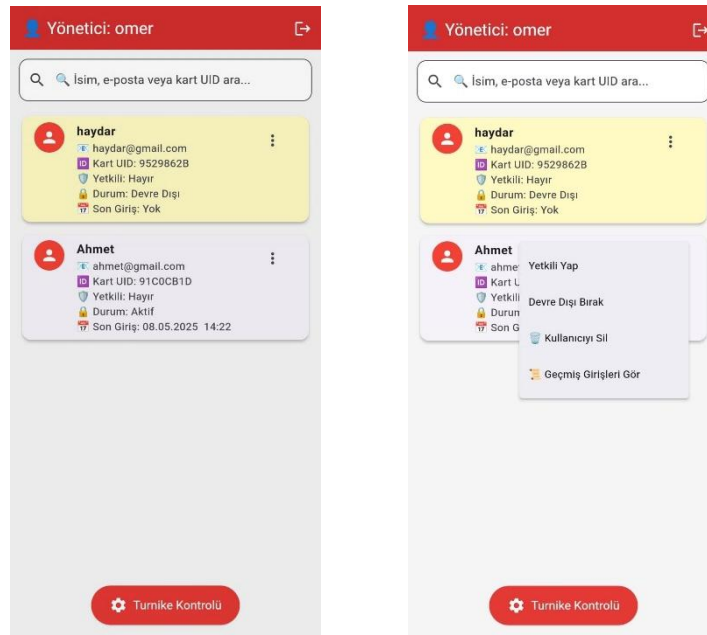
Şekil 2. Kayıt Olma ve Kart Okutma Ekranları

3.2.3 Admin Paneli

Yönetici (admin) hesabıyla sisteme giriş yapan kullanıcılar, tüm kayıtlı kullanıcıların bilgilerini liste halinde görüntüleyebilir. Her kullanıcıya ait isim, e-posta, kart UID'si, yetkili durumu, aktif/devre dışı durumu ve son giriş zamanı gibi bilgiler burada detaylı şekilde sunulmaktadır.

Arama çubuğu sayesinde admin, kullanıcıları e-posta, isim veya UID üzerinden kolayca filtreleyebilir. Ayrıca her kullanıcının kartına özel yönetim işlemleri yapılabilir. Bu işlemler arasında; kullanıcıyı yetkili yapma, devre dışı bırakma, silme veya geçmiş giriş/çıkış kayıtlarını görüntüleme gibi seçenekler yer alır.

Bu panel sayesinde sistemin tüm kullanıcıları üzerinde anlık ve esnek bir kontrol sağlanmakta, güvenlik ve erişim yönetimi kolaylıkla yapılabilir. Bu ekran Şekil 3'te gösterilmiştir.



Şekil 3. Admin Paneli ve Kullanıcı Yönetim Seçenekleri

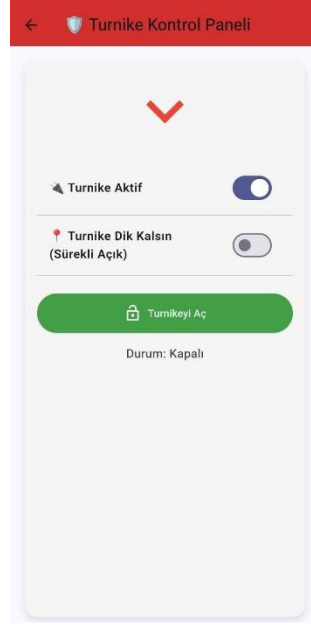
3.2.4 Turnike Kontrol Paneli

Turnike kontrol paneli, yöneticilerin sistemdeki turnikeyi uzaktan kontrol etmesini sağlayan özel bir ekrandır. Bu panelde iki adet anahtar (toggle switch) bulunur:

- **Turnike Aktif:** Sistem genelinde turnikenin çalışır durumda olup olmadığını belirler. Pasif hale getirildiğinde hiçbir kullanıcı geçiş yapamaz.
- **Turnike Dik Kalsın (Sürekli Açık):** Bu özellik aktif edildiğinde, turnike açık pozisyonunda sabit kalır. Özellikle yoğun giriş-çıkış saatlerinde manuel kapatma ihtiyacını ortadan kaldırır.

Ayrıca "Turnikeyi Aç" butonuyla admin, Firebase'e bir açılma komutu göndererek servo motorun turnikeyi anlık olarak açmasını sağlayabilir. Bu komut sadece turnike aktif durumdaysa çalışır.

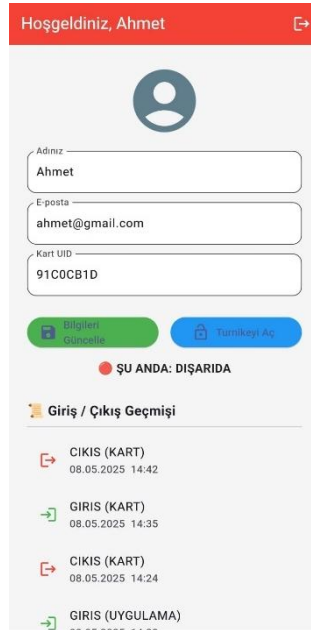
Alt kısımda, turnikenin anlık durumu (örneğin: "Kapalı") metin olarak gösterilmektedir. Bu ekran Şekil 4'te gösterilmiştir.



Şekil 4. Turnike Kontrol Paneli Arayüzü

3.2.5 Kullanıcı Paneli

Normal kullanıcılar sisteme giriş yaptıktan sonra kendi bilgilerini görüntüleyebilir ve “Turnikeyi Aç” butonunu kullanarak mobil uygulama ile geçiş yapabilir. Bu panelde kullanıcıya ait bilgiler yer almakta ve gerektiğinde güncellenebilmektedir. Alt bölümde yer alan “**Giriş / Çıkış Geçmişi**”, kullanıcının hangi tarih ve saatlerde hangi yöntemle (kart veya uygulama) giriş veya çıkış yaptığını ayrıntılı şekilde listeler. Bu sayede kullanıcı kendi erişim geçmişini takip edebilir. Bu ekran Şekil 5’te gösterilmiştir.



Şekil 5. Kullanıcı Paneli ve Geçiş Geçmişi Görüntüleme Ekranı

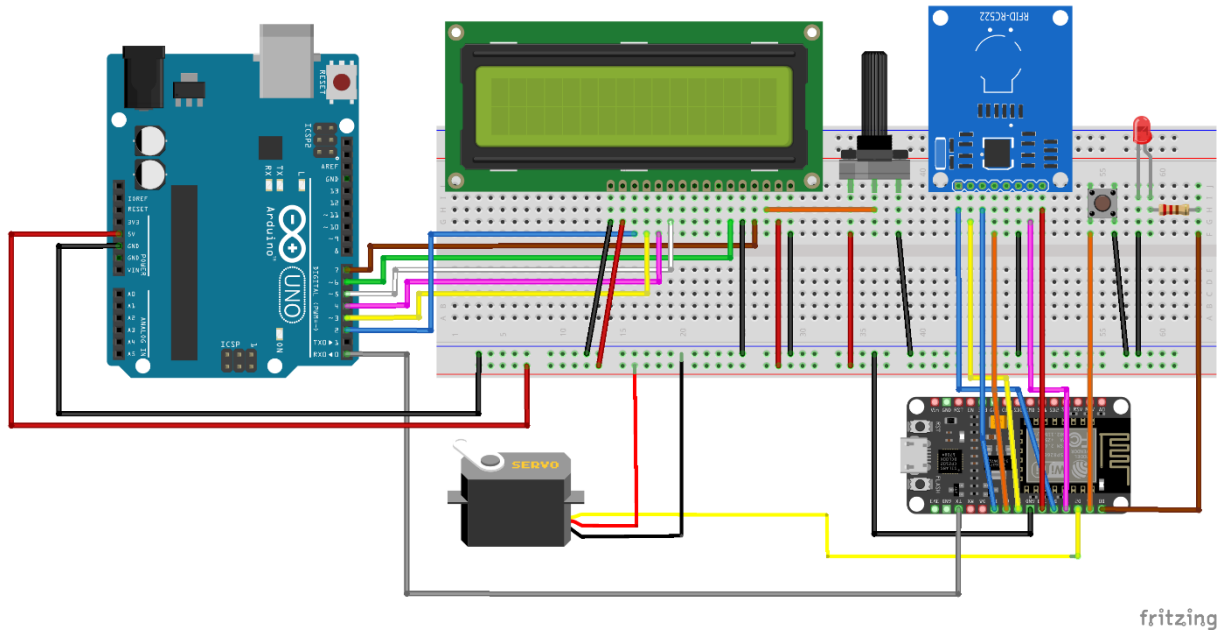
3.3 Devre Şeması

Sistemin fiziksel bağlantıları breadboard üzerinde kurulmuştur ve temel bileşenler arasında ESP8266 NodeMCU, RC522 RFID okuyucu, SG90 servo motor, Arduino Uno ve 16x2 LCD ekran yer almaktadır. Devre, hem geçici prototipleme hem de test amaçlı olarak tasarlanmıştır.

ESP8266, sistemin ana kontrol birimidir ve Wi-Fi üzerinden Firebase ile haberleşir. RFID okuyucu, SPI protokolüyle doğrudan ESP8266'ya bağlanmıştır. Servo motor ise ESP8266'nın **PWM destekli D2 (GPIO4)** pinine bağlanmış ve Servo.h kütüphanesi kullanılarak kontrol edilmiştir. Turnike açma ve kapama hareketi bu pin üzerinden verilen darbe genişliği sinyalleriyle gerçekleştirilmiştir.

LCD ekran, Arduino üzerinden LiquidCrystal.h kütüphanesi ile kontrol edilmekte; ESP8266, Arduino'ya UART (seri haberleşme) üzerinden "LCD:..." formatında mesajlar göndererek kullanıcı bilgilendirmesi sağlamaktadır.

Tüm bileşenler breadboard üzerinde jumper kablolar ile bağlanmış ve güç kaynağı olarak USB çıkışlar kullanılmıştır. Tasarlanan Fritzing devre modeli Şekil 6'da gösterilmiştir.



Şekil 6. Turnike Sistemine Ait Devre Şeması

3.4 Firebase Veri Yapısı

Firestore Realtime Database, sistemdeki tüm kullanıcı, log ve kontrol bilgilerinin tutulduğu merkezi veri tabanıdır. Veriler JSON formatında hiyerarşik bir yapıda organize edilmiştir. Sistemde kullanılan temel veri düğümleri şunlardır:

- /kullanıcılar: Her kullanıcının UID'si altında kişisel bilgileri (isim, e-posta, kart UID, yetki durumu, devre dışı bilgisi) tutulur.
- /kontrol: Turnikenin genel durumu ve uzaktan kontrol parametreleri bu düğümden saklanır (turnike_aktif, dik_kalsin, ac_istegi, son_komut gibi).

- /loglar: Tüm geiş olayları burada saklanır. Her log giriř/ıkıř tr, kart UID'si, kullanıcı ismi, kaynak (uygulama/kart) ve zaman bilgisini ierir. řekil 7'de veritabanı tasarımına ait ekran grnts sunulmuřtur.
- /sayacllar: Gnlk bazda kullanıcı bařına geiř sayısı burada tutulur. rneėin "20250508: 3" gibi.
- /sondurum: Her kullanıcının en son yaptığı geiřin tipi, zamanı ve yntemi bu alanda yer alır.
- /tarama: Kullanıcılar mobil uygulamada "ESP ile Tara" butonuna bastığında bu dėm aktif hle gelir. ESP8266 tarafında srekli izlenen bu alan, kart okutulduğunda UID'nin buraya yazılması ile mobil uygulamaya otomatik olarak aktarılır. UID eřleřtirmesi ve kayıt doėrulama bu sayede hızlı bir řekilde yapılır.

Bu veri yapısı sayesinde kullanıcı iřlemleri, turnike kontrol ve gvenlik loglamaları merkezi olarak senkronize alıřmakta; mobil uygulama, ESP8266 ve Arduino arasında gvenli ve hızlı bir etkileřim saėlanmaktadır.



řekil 7. Firebase Veri Yapısı Ekran Grnts

3.5 Sistem Kesme ve Yeniden Bařlatma Mekanizması

Geliřtirilen sistemde, ESP8266'ya baėlı D1 pinine entegre edilmiř bir buton aracılıėıyla donanımsal kesme (interrupt) oluřturulmuřtur. Bu buton, attachInterrupt() fonksiyonu ile izlenmekte ve kullanıcı tarafından basıldığında sistemin temel iřlevleri yazılım zerinden sıfırlanmaktadır.

Kesme tetiklendiėinde ESP8266 zerindeki kritik sreler Wi-Fi baėlantısı, Firebase eriřimi, servo ve LCD ynetimi gibi iřlemler yeniden bařlatılmakta ve sistemin kararlı řekilde alıřması saėlanmaktadır. Bu yapı, fiziksel reset pininden farklı olarak yazılım iinde tanımlanan ISR (Interrupt Service Routine) iřlevi zerinden gerekleřtirilmekte ve sistemin esnekliėini artırmaktadır.

Bu zellik, zellikle Wi-Fi baėlantısında kopma yařandığında veya Firebase ile haberleřme geici olarak durduğunda manuel mdahale ile sistemin toparlanmasını saėlamak amacıyla tasarlanmıřtır. Bu sayede kullanıcı, tm sistemi bařtan programlamaya gerek kalmadan sadece bir butonla yeniden baėlantı kurulmasını saėlayabilmektedir.

4. Sonuç ve Değerlendirme

Bu çalışmada, geleneksel kartlı geçiş sistemlerine alternatif olarak uzaktan kontrol edilebilir, esnek ve güvenli bir turnike geçiş sistemi tasarlanmış ve uygulanmıştır. ESP8266 mikrodenetleyicisi ile RC522 RFID okuyucu entegre edilerek fiziksel geçişler sağlanmış; aynı zamanda Flutter ile geliştirilen mobil uygulama üzerinden Firebase Realtime Database aracılığıyla uzaktan kontrol ve kullanıcı yönetimi mümkün kılınmıştır.

Geliştirilen sistem, kullanıcıların hem fiziksel kartları hem de mobil cihazları üzerinden geçiş yapabilmelerini sağlamakta; yönetici paneli sayesinde kişi bazlı yetkilendirme, devre dışı bırakma ve geçmiş kayıtların izlenmesi mümkün olmaktadır. Ayrıca, EEPROM kullanılarak 30 saniyelik geçiş kısıtlaması getirilmiş ve bu sayede sistemin kötüye kullanımının önüne geçilmiştir.

Yazılım tabanlı kesme mekanizması ile sistem, herhangi bir bağlantı sorunu durumunda kullanıcı müdahalesiyle hızlıca yeniden başlatılabilmekte, bu da sistemin kararlılığını ve güvenliğini artırmaktadır. Firebase ile tutulan veriler sayesinde hem giriş/çıkış logları hem de günlük sayaçlar gerçek zamanlı olarak izlenmekte ve saklanmaktadır.

Yapılan testlerde, sistemin farklı senaryolarda (kart ile geçiş, mobil ile geçiş, admin müdahalesi vb.) sorunsuz çalıştığı ve geçişlerin güvenli bir şekilde kaydedildiği gözlemlenmiştir. Ayrıca LCD ekran ile kullanıcıya anlık bilgi verilmesi, sistemin kullanıcı dostu olmasını sağlamıştır.

Sonuç olarak, geliştirilen bu sistem düşük maliyetli, modüler ve uzaktan yönetilebilir bir çözüm sunmakta olup benzer geçiş kontrol projeleri için uygulanabilir ve geliştirilebilir bir temel oluşturmaktadır.

5. KAYNAKLAR

- [1] J. Figueiredo and J. S. Da Costa, "A SCADA system for energy management in intelligent buildings," *Energy and Buildings*, vol. 49, pp. 85–98, 2012.
- [2] R. Hunzinger, "SCADA fundamentals and applications in the IoT," in *Internet of Things and Data Analytics Handbook*, pp. 283–293, 2017.
- [3] B. Kara, "IoT tabanlı RFID ile güvenli giriş sistemi tasarımı," *Sakarya Üniversitesi Teknoloji Fakültesi Dergisi*, vol. 10, no. 2, pp. 100–105, 2021.
- [4] A. Demirtaş, "ESP8266 ile mobil kontrollü geçiş sistemleri üzerine bir uygulama," in *Akıllı Sistemler Konferansı*, 2022.
- [5] Google, "Realtime Database Security Rules," *Firebase Documentation*, [Online]. Available: <https://firebase.google.com/docs/database/security>. [Accessed: May 9, 2025].
- [6] Google, "Authentication," *Firebase Documentation*, [Online]. Available: <https://firebase.google.com/docs/auth>. [Accessed: May 9, 2025].
- [7] H. Altın, "LCD seri haberleşme uygulaması," *Kullanıcı kaynaklı proje dosyası*, 2025.
- [8] H. Altın, "Flutter admin panel ve ESP8266-Firebase entegrasyonu," *Kullanıcı kaynaklı proje dosyası*, 2025.

6.Ekler

Proje kapsamında geliştirilen tüm kaynak kodlar, dosya boyutlarının fazlalığı nedeniyle doğrudan rapora eklenmemiştir. Bunun yerine, ilgili dosyalar ve klasörler ayrı başlıklar altında Google Drive bağlantılarıyla paylaşılmıştır. Aşağıda her bir bileşene ait kodların açıklamaları ve bağlantıları yer almaktadır:

6.1 Turnike Sistemi ESP8266 Kodu

ESP8266 üzerinde çalışan ana kod; RFID kart okuma, Firebase bağlantısı, servo kontrolü ve LCD veri gönderimini içermektedir.

<https://drive.google.com/file/d/1Xrr77YuQd8ddFVWWezBGpQCaC7sQJmaO/view?usp=sharing>

6.2 Özel Olarak Yazılan TurnikeSistemi Kütüphanesi

Turnike kontrol mantığını modüler hale getiren, C++ sınıf yapısına sahip özel kütüphane.

6.2.1 TurnikeSistemi.cpp

https://drive.google.com/file/d/1TKHhbTTW5Je6xgIj1jaMC04r76_otyC9/view?usp=sharing

6.2.2 TurnikeSistemi.h

<https://drive.google.com/file/d/1sKDTC3rdq-0UqUG3mcU5HhIk8Y88AMqH/view?usp=sharing>

6.3 Turnike Sistemi Arduino Kodu

ESP8266'dan gelen "LCD:..." mesajlarını UART ile alarak 16x2 LCD ekrana aktaran Arduino kodudur.

<https://drive.google.com/file/d/1ZdZxEn8-arWRQ4MwVcC3izBtYh9AsYha/view?usp=sharing>

6.4 Mobil Uygulama (Flutter) Kodları

Kayıt/giriş, UID kart okuma, admin paneli ve Firebase senkronizasyonunu içeren tüm mobil uygulama kodları aşağıdaki klasörde yer almaktadır:

https://drive.google.com/drive/folders/1jEh19gNPnd7b_3Zabqg4DWP6Gn9nE_fo?usp=drive_link