YAKIN ALAN İLETİŞİMİ TABANLI GÜVENLİ EV OTOMASYON YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ

DEVELOPMENT OF A NEAR FIELD COMMUNICATION BASED SECURE HOME AUTOMATION MANAGEMENT SYSTEM

TOLGA HAKAN ODUNCU

Doç. Dr. ALİ ZİYA ALKAR

Tez Danışmanı

Hacettepe Üniversitesi

Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin

Elektrik ve Elektronik Mühendisliği Anabilim Dalı İçin Öngördüğü

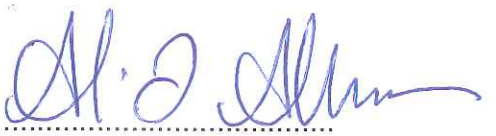
YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

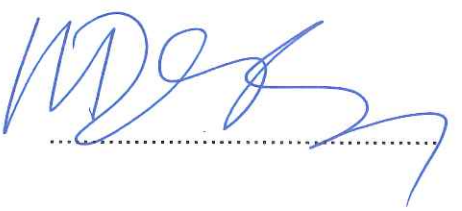
2016

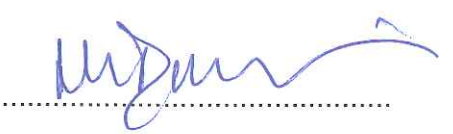
TOLGA HAKAN ODUNCU'nun hazırladığı **"Yakın Alan İletişimi Tabanlı Güvenli Ev Otomasyon Yönetim Sisteminin Geliştirilmesi"** adlı bu çalışma aşağıdaki jüri tarafından **ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Selçuk GEÇİM
Başkan ..... 

Doç. Dr. Ali Ziya ALKAR
Danışman ..... 

Doç. Dr. Umut SEZEN
Üye ..... 

Yrd. Doç. Dr. Derya ALTUNAY
Üye ..... 

Yrd. Doç. Dr. Mehmet DEMİRER
Üye ..... 

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Salih Bülent ALTEN
Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

29/02/2016

Tolga Hakan ODUNCU

ÖZET

YAKIN ALAN İLETİŞİMİ TABANLI GÜVENLİ EV OTOMASYON YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ

TOLGA HAKAN ODUNCU

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Doç Dr. Ali Ziya ALKAR

Şubat 2016, 66 sayfa

Teknolojinin gelişimine paralel olarak, akıllı ev sistemleri ve ev otomasyonları günlük hayatımızda daha yaygın kullanılmaya başlanmıştır. Bu sistemler, insan hayatını kolaylaştıracak bir çok yeniliğin yanısıra bir takım riskler ve yetkisiz erişim gibi çok önemli güvenlik sorunlarını da beraberinde getirmektedir.

Bununla birlikte, hem kullanım kolaylığı hem de sadece çok kısa mesafelerde iletişim kurulabilmesi sayesinde sunduğu fiziksel güvenlik özelliği ile, mobil ödeme, elektronik biletleme ve kimlik tanımlama gibi pek çok alanda sıkça karşılaşılan yeni bir teknoloji olan Yakın Alan İletişimi (NFC), otomasyon sistemlerinde de sistem güvenliği için kullanılabilecek önemli bir teknolojidir. NFC kısaca, akıllı telefon ve benzeri cihazların belirli bir mesafe içinde birbirleri ile güvenli bir radyo iletişimi kurması esasına dayanır.

Bu tez çalışmasında, akıllı telefon veya tablet gibi NFC özelliği bulunan mobil cihazlarla kontrol edilebilen mevcut ev otomasyon sistemlerini, donanım seviyesinde NFC tabanlı bir doğrulama mekanizmasının entegre edilmesi ile daha güvenli hâle getirebilecek bir sistemin tasarlanması, bu sistem için gerekli platformlarda ihtiyaç duyulan yazılımların geliştirilmesi ve eşler arası (peer-to-peer) yakın alan iletişimi ile bu sistemin kullanımını kolaylaştıracak farklı senaryoların işlenmesi ele alınmıştır.

Anahtar Kelimeler: Yakın Alan İletişimi, NFC, Otomasyon, Ev Otomasyon Güvenliği, Mobil Cihaz, Android Uygulaması, Eşler Arası İletişim

ABSTRACT

DEVELOPMENT OF A NEAR FIELD COMMUNICATION BASED SECURE HOME AUTOMATION MANAGEMENT SYSTEM

TOLGA HAKAN ODUNCU

Master of Science, Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Dr. Ali Ziya ALKAR

February 2016, 66 pages

In parallel to the progress in technology, smart home systems and home automations have been widely used in our daily life. These systems come with several risks and very important security issues such as unauthorized access even though they bring new ways of making life easier and more comfortable.

Furthermore, a new technology, Near Field Communication (NFC), often encountered in many fields such as mobile payment, electronic ticketing and identification through the ease of use and capability of providing secure physical communication within only short distances, is an important technology that can also be used for system security of automation systems. Briefly, NFC can be used in smartphones and similar devices in order to establish a secure radio communication located at a certain distance among themselves.

In this study, the aim is to design a system to improve the current home automation systems, which can be managed by NFC-capable mobile devices. This is accomplished by making them more secure by integrating an NFC-based hardware level authentication mechanism, developing the required software for this system in necessary platforms and adding different operation scenarios to facilitate the use of the system with peer-to-peer near field communication.

Keywords: Near Field Communication, NFC, Automation, Home Automation Security, Mobile Device, Android Application, Peer-to-Peer Communication

TEŞEKKÜR

Tez ve proje hazırlama süresi boyunca, verdiği destek, gösterdiği ilgi, anlayış ve yaptığı katkılardan dolayı çok değerli tez danışmanım Sayın Doç. Dr. Ali Ziya ALKAR'a

Yüksek lisans eğitimim boyunca benden hiç bir desteğini esirgemeyen, varlığıyla hayatıma yeni bir anlam katan sevgili eşim Fatma Nur'a

Tezimi hazırlarken kendisiyle geçireceğim çok değerli zamanları, babasının kariyeri için feda eden biricik oğlum Arel Batu'ya

Hayatım boyunca, her zaman yanımda olup verdikleri her türlü destek için kıymetli anneme, babama ve kardeşime

Karşılaştığım zorluklar karşısında bilgi ve tecrübelerini benle paylaşarak bana yol gösteren saygıdeğer iş arkadaşlarıma ve meslektaşlarıma

sonsuz teşekkürlerimi sunuyorum.

İÇİNDEKİLER

Sayfa

ÇİZELGELER

Sayfa

ŞEKİLLER

Sayfa

KISALTMALAR

AID	Uygulama Kimliği (Application Identification)
APDU	Uygulama Protokolü Veri Birimi (Application Protocol Data Unit)
API	Uygulama Programlama Arayüzü (Application Programming Interface)
CPU	Merkezi İşlem Birimi (Central Processing Unit)
EPROM	Silinip Programlanabilir Salt Okunur Bellek (Eraseble Programmable Read-Only Memory)
EHS	Akıllı Binalar İçin Geliştirilmiş Bir Ağ İletişim Protokolü Standardı (European Home Systems)
GSM	Mobil İletişim İçin Küresel Sistem (Global System for Mobile Communications)
HAVi	Eviçi Ses Video Uyumluluğu (Home Audio Video Interoperability)
HCE	Ana Sistem Tabanlı Kart Öykünümü (Host-based Card Emulation)
HTTP	Hiper Metin Aktarım Protokolü (Hypertext Transfer Protocol)
ID	Kimlik (Identification)
IDE	Entegre Geliştirme Ortamı (Integrated Development Environment)
IEC	Uluslararası Elektroteknik Komisyonu (International Electrotechnical Commission)
IEEE	Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
IIS	İnternet Bilgi Servisleri (Internet Information Services)
IMEI	Uluslararası Mobil Cihaz Kimliği (International Mobile Station Equipment Identity)

INS	Talimat (Instruction)
IP	Arayüz ve Protokoller (Interface and Protocols)
ISO	Uluslararası Standartlar Organizasyonu (International Organization for Standardization)
JSON	Javascript Nesne Notasyonu (Javascript Object Notation)
KNX	Akıllı Binalar İçin Geliştirilmiş Bir Ağ İletişim Protokolü Standardı (Network Communications Protocol Standard for Intelligent Buildings)
M2M	Makineden Makineye (Machine-to-Machine)
MAC	Ortam Erişim Denetimi (Media Access Control)
MVC	Model Görünüm Denetleyici (Model View Controller)
N/A	Müsait Değil (Not Available)
NDEF	NFC Veri Değiş Tokuş Biçimi (NFC Data Exchange Format)
NFC	Yakın Alan İletişimi (Near Field Communication)
P2P	Eşler Arası (Peer-to-peer)
RFID	Radyo Frekansı ile Tanımlama (Radio-frequency Identification)
SQL	Yapılandırılmış Sorgu Dili (Structured Query Language)
SSL	Güvenli Soket Katmanı (Secure Socket Layer)
USB	Evrensel Seri Veriyolu (Universal Serial Bus)
VPN	Sanal Özel Ağ (Virtual Private Network)

WAP	Kablosuz Uygulama Protokolü (Wireless Application Protocol)
WPF	Windows Tabanlı Grafiksel Arayüz Sistemi (Windows Presentation Foundation)
XAML	Genişletilebilir Uygulama İşaretleme Dili (Extensible Application Markup Language)
XML	Genişletilebilir İşaretleme Dili (Extensible Markup Language)

SÖZLÜK DİZİNİ

Ağ Geçidi	: Gateway
Arayüz ve Protokoller	: Interface and Protocols
Biçim Vermek	: Casting
Cihaz	: Device
Değer	: Value
Dizi	: Array
Doğrudan Kablosuz Bağlantı	: Wi-Fi Direct
Doğrulama	: Authentication
Doğrulama Sunucusu	: Authentication Server
Dosya Sistemi Yayınlama	: File System Publish
Elektrik ve Elektronik Mühendisleri Enstitüsü	: Institute of Electrical and Electronics Engineers
Entegre Geliştirme Ortamı	: Integrated Development Environment
Eşler Arası	: Peer-to-peer
Eviçi Ses Video Uyumluluğu	: Home Audio Video Interoperability
Evrensel Seri Veriyolu	: Universal Serial Bus
Genişletilebilir İşaretleme Dili	: Extensible Markup Language
Genişletilebilir Uygulama İşaretleme Dili	: Extensible Application Markup Language
Güvenli Soket Katmanı	: Secure Socket Layer
Hiper Metin Transfer Protokolü	: Hypertext Transfer Protocol
Javascript Nesne Notasyonu	: Javascript Object Notation
Kablosuz Uygulama Protokolü	: Wireless Application Protocol
Kimlik	: Identification
Kullanıcı	: User
Makineden Makineye	: Machine-to-Machine
Merkezi İşlem Birimi	: Central Processing Unit
Mevcut Değil	: Not Available
Mobil İletişim İçin Küresel Sistem	: Global System for Mobil Communications
Model Görünüm Denetleyici	: Model View Controller
Nesnelerin Inteneti	: Internet of Things
Ortam Erişim Denetimi	: Media Access Control
Radyo Sinyalleri Huzmesi	: Beam

Salt Okunur	: Read-only
Sanal Özel Ağ	: Virtual Private Network
Sarmal	: Spiral
Sınıf	: Class
Silinip Programlanabilir Salt Okunur Bellek	: Erasable Programmable Read-Only Memory
Sorgu	: Query
Talimat	: Instruction
Uluslararası Mobil Cihaz Kimliği	: International Mobile Station Equipment Identity
Uluslararası Standartlar Organizasyonu	: International Organization for Standardization
Uygulama Kimliği	: Application Identification
Uygulama Programlama Arayüzü	: Application Programming Interface
Uygulama Protokolü Veri Birimi	: Application Protocol Data Unit
Varsayılan	: Default
Veri Değiş Tokuş Biçimi	: Data Exchange Format
Veritabanı	: Database
Veritabanı Motoru	: Database Engine
Yakın Alan İletişimi	: Near Field Communication
Yapılandırılmış Sorgu Dili	: Structured Query Language

1. GİRİŞ

Evde ve ev dışındaki günlük hayatımızda yaptığımız işlerimiz ve görevlerimiz otomatik olarak gerçekleşmeye başladığından beri otomasyon kavramı daha sık karşımıza çıkmaya başlamıştır. Otomasyon terimi, bir işin tamamen ya da kısmen makineler ile yapılması şeklinde ifade edilebilir. Yapılan işin ne kadarının manuel ve ne kadarının otomatik olarak yapıldığı, otomasyon seviyesi için belirleyici genel parametredir. Bazı işler büyük oranda insan kontrolü ile manuel olarak yapılmaktayken, bazı işler tamamen otomatik olarak gerçekleşmektedir. Otomasyon ile gerçekleştirilen işler, bir cihazı basitçe açıp kapatmaktan, robotik cerrahi ameliyatları gibi çok karmaşık operasyonlara kadar değişebilmektedir.

Otomasyon sistemlerinin evlerde kullanılmaya başlanması ile *Akıllı Ev* ve *Ev Otomasyonu* terimleri de hayatımıza girmiştir. Ev otomasyon sistemlerinin odak noktasında bulunan *Akıllı Ev* kavramı ilk defa 1984 yılında Amerikan Ev İnşaatçılar Birliği tarafından kullanılmıştır [1]. İlk başlarda, kullanımdaki zorluklar ve yüksek maliyetler nedeniyle toplumun geneline ulaşamasa da 2010 yılından itibaren, akıllı telefonların yaygınlaşması ve akıllı ev donanımlarının daha uygun fiyatlı hale gelmesiyle daha fazla sayıda evde bu sistemler kullanılmaya başlanmıştır. Akıllı ev sistemleri hakkında yapılan çalışmalar ve yayımlanmış makaleler bu sistemlerin gelişmesine çok katkı sağlamışlardır. Bu çalışmalar sonucunda bir çok standart ve iletişim protokolü oluşturulmuş ve bunları temel alan çeşitli teknolojiler geliştirilmiştir. Örneğin, Nikolova, Mejis ve Voorwinden [2] ev ağına bağlı TV, VCR, kamera ve radyo gibi cihazların WAP aracılığıyla uzaktan kontrol edilmesini sağlayan bir sistem önermişlerdir. Mevcut sistemlerdeki olanakları ve zorlukları araştırabilmek için genel bir ev-mobil ağ geçidi mimarisinin geliştirildiği bu sistemde iletişim protokolü olarak HAVi (Home Audio Video Interoperability) tercih edilmiştir. Alheraish [3] bir GSM modülü dahil ettiği ev otomasyon sisteminde çeşitli uygulamalarda kullanılabilecek bir M2M (Machine-to-Machine) tasarım gerçekleştirmiştir. Mikrodenetleyici ve GSM modülü ile cihazların kontrol edildiği sistemin test edilmesi için bilgisayar tabanlı ortamlar kullanarak iki farklı senaryo sunmuştur. Alkar, Yüksekkaya, Kayalar, Tosun ve Özcan [4] gerçek zamanlı cihaz izleme ve uzaktan kontrolü için internet, GSM ve sesli komutların birlikte kullanıldığı geniş çaplı bir interaktif otomasyon sistemi tasarlamışlardır. Tasarımlarında, sistemin düşük maliyetli, verimli ve kullanıcı dostu olmalarını öne çıkarmışlardır. Gill, Shuang-Hua, Fang ve Xin [5] tasarımlarında

anahtar, valf, sensör gibi çeşitli ekipmanlarla, ZigBee standardının, klasik bir ağ geçidi mimarisine sahip ev otomasyon sistemlerine verilmi bir şekilde nasıl uygulanabileceğini ele almıştır. Çalışmalarında, ev otomasyon sistemlerinin günlük hayata adaptasyonunun yavaş olmasının nedenlerini açıklamışlar ve ZigBee'nin sahip olduğu potansiyel sayesinde bu durumu nasıl değiştirebileceğini değerlendirmişlerdir. Bu çalışmaların yanısıra, NFC özellikli cihazların kullanımı yaygınlaşmadan önce, ev otomasyon sistemlerinde güvenlik bileşenini yazılım seviyesinde gerçekleştiren bazı çalışmalar da bulunmaktadır. Bergstrom, Driscoll ve Kimball [6] özel bir şirket tarafından piyasaya sunulan küresel ev sunucusunun sağlamış olduğu ağ güvenliği altyapısını kullanarak ev otomasyon sistemi ile web üzerinden haberleşmeyi, karşılaşılabilecek tüm atak ve kandırmacalara karşı daha güvenli kılacak yöntemler önermiştir. Alkar ve Buhur [7] çok çeşitli cihazların dahil edilebileceği internet tabanlı, mikrodenetleyiciler aracılığıyla çalışan düşük maliyetli ve güvenilir bir ev otomasyon sistemi önermişlerdir. Sundukları çalışma, basit cihazlar için geliştirilen algoritma ve altyapının daha karmaşık cihazlar için uygulanabilir olduğunu göstermeyi amaçlamaktadır. Mondal, Roy ve Bhattacharya [8] internet tabanlı bir ev otomasyon sistemine biyometrik veri ve SSL sertifika doğrulama yöntemi ile giriş yapmayı öneren bir tasarım gerçekleştirmiştir. Biyometrik veri olarak iris görüntüsü kullanılan çalışmada, biyometrik veriyle birlikte özet fonksiyonunu doğrulama algoritmasına dahil ederek veri güvenliğini sağlamışlardır. Ayrıca, birden fazla konutun birlikte kullanabileceği bir doğrulama sunucusunu tasarımlarına dahil ederek sistemin kullanıcı başına düşen maliyetini düşürmeyi hedeflemişlerdir. De Luca, Lillo, Mainetti, Mighali, Patrono ve Sergi [9] Android işletim sistemli mobil cihazlar için geliştirdikleri bir mobil uygulama ile ev ağına bağlı cihazların KNX protokolü üzerinden kontrol edilmesini sağlayan bir sistem önermişlerdir. Ev içindeki mobil cihazların yerel ağ üzerinden, ev dışındaki mobil cihazların da bir VPN tüneli üzerinden sisteme bağlanabileceği bir mimari tasarım sunarak kullanıcılara esnek ve güvenli bir kullanım sağlamayı amaçlamışlardır.

Otomasyon sistemlerinin yaygınlaşmasına paralel olarak otomasyon yazılımlarına olan bağımlılık arttıkça, bu sistemlere karşı güvenlik tehditleri de artmaktadır. Güvenlik bileşenleri arasına donanım katmanları eklemek bu sistemlerin erişim güvenliğini artırmak için alınabilecek önlemlerden birisidir. Sunduğu güvenlik ve

taşınabilirlik olanakları ile NFC olarak isimlendirilen “Yakın Alan İletişimi” günümüzde bu amaçla kullanılabilecek önemli bir teknolojidir.

Bu tez çalışmasında, mobil cihazlarla kontrol edilebilen mevcut otomasyon sistemlerini, dahili veya harici NFC özelliğine sahip cihazlar ile donanım seviyesinde NFC tabanlı bir doğrulama mekanizmasının entegre edilmesi ile daha güvenli hâle getirebilecek, kullanıcı ve cihaz profilleri yönetiminin bir sistem yöneticisi tarafından yönetildiği güvenli, esnek ve düşük maliyetli bir sistemin tasarlanması, bu sistem için gerekli platformlarda uygun yazılımların geliştirilmesi ve eşler arası yakın alan iletişimi ile bu sistemin kullanımını kolaylaştıracak farklı senaryoların işlenmesi ele alınmıştır. Çalışma kapsamında önerilen yöntemler ve senaryolar, mevcut ev/ofis/otel vb. otomasyonlarındaki sistem güvenliği ve uygulama senaryolarına güvenli ve kullanışlı alternatifler sunmayı amaçlamaktadır.

Bölüm 2‘de, ev/bina otomasyonları hakkında genel bilgiler verilmiş ve bu sistemlerin insan yaşamına sunduğu katkılar değerlendirilmiştir.

Bölüm 3’te NFC hakkında genel bilgiler, NFC’nin operasyon modları, NFC teknolojisinin mevcut iletişim teknolojileriyle karşılaştırılması, günlük hayatta sağladığı avantajlar ve kolaylıklar ve NFC’nin çeşitli otomasyon sistemlerine uyarlanması hakkında yapılan çalışmalar incelenmiştir.

Bölüm 4’te, sistemin genel mimarisi, yazılım ve donanım bileşenleri, sistemin genel işleyişi, NFC teknolojisinin farklı modlarının kullanıldığı genel kullanım senaryosu ve eşler arası kullanım senaryosu ele alınmıştır.

Bölüm 5’te, tasarlanan sistemin işleyişini sağlamak amacıyla çeşitli platformlar için geliştirilmiş uygulamaların teknik detayları anlatılmıştır.

Bölüm 6‘da sistemin farklı uygulama alanlarını değerlendirmek için örnek kullanım senaryoları işlenmiştir.

Bölüm 7’de ise, önerilen çalışmanın benzer çalışmalara göre kıyaslanması ve değerlendirilip yorumlanması gerçekleştirilmiştir.

2. EV OTOMASYONLARI

Ev otomasyonu, kısaca, bir evde/binada bulunan cihazları uzaktan kontrol etmek veya otonom olarak çalışmasını sağlamak şeklinde açıklanabilir. Son yıllarda daha yaygın biçimde kullanılmaya başlanan ev ve bina otomasyonları, modern hayatın önemli bir parçası haline gelmiştir. İletişim ve mobil cihaz teknolojileri alanında yapılan çalışmalar ve bu çalışmalar neticesinde ortaya çıkan gelişmeler, ev ve bina otomasyon teknolojilerinin daha kullanışlı olmasında ve yaygınlaşmasında önemli bir rol oynamıştır.

Ev otomasyonları ile kontrol edilen binalar, *Akıllı Ev* veya *Akıllı Bin*a olarak da adlandırılır. Ev otomasyonları ile Şekil 2.1'deki örnek görselde görüldüğü gibi [10], aydınlatma cihazları, ısıtma-soğutma-havalandırma cihazları, enerji şebekesi, multimedya cihazları, gaz-duman dedektörleri, sensör ve kamera benzeri güvenlik bileşenleri, pencere-kapı-kilit mekanizmaları gibi çok sayıda farklı cihaz/sistem gözlemlenebilir, kumanda edilebilir ve/veya otonom bir şekilde çalışması sağlanabilir.

Şekil 2.1. Ev Otomasyonu Uygulaması Örneği

Bu cihazlar, ISO, IEEE gibi yetkin organizasyonlar tarafından standart olarak tanımlanmış Wi-Fi, Bluetooth, EHS, ZigBee, KNX vb. iletişim protokolleri üzerinden sistem ile haberleşirler.

Cihazların izlenmesi ve kumanda edilmesi, binada bulunan sabit bir giriş aygıtından yapılabildiği gibi mobil bir cihaz ile de yapılabilir. Özellikle, akıllı telefon ve tablet gibi

mobil işletim sistemleri ile çalışan cihazlar, geliştirilen mobil uygulamalar sayesinde, otomasyon sistemlerinde kumanda görevi üstlenmek için ideal bir seçenek haline gelmiştir.

Bir çok ev otomasyon sistemi, [9]'da olduğu gibi, internet ağı üzerinden de sistemin yönetilmesine izin verir. Böylece, kullanıcılar dışarıdayken de evde çalışan cihazlar/sistemler hakkında bilgi sahibi olabilir ve cihazları/sistemleri kumanda edebilirler. Özellikle, evden uzaktayken güvenlik bileşenlerinin izlenebilmesi, hırsızlıkları engelleme ve konut/bina güvenliğini sağlama konusunda önemli bir avantaj sağlamaktadır. Aynı şekilde, ısıtma-soğutma-havalandırma sistemlerinin de uzaktan izlenmesi, kontrol edilmesi ve otonom çalışabilmesi, tehlikeli durumlarda yangın vb. olası bir felakete karşı önlem alınmasını ve ilgili kişi ve kurumların derhal haberdar edilmesini sağlamaktadır. Güvenlik ve emniyet konusunda sağlanan faydaların yanısıra, otomasyon sistemleri, otonom çalışan cihazlarla enerji tüketiminin düşürülmesine katkıda bulunduğu için enerji tasarrufu amacıyla da kullanılmaktadır. Otonom çalışan cihazlar, aynı zamanda kullanıcılara önemli oranda zaman tasarrufu da sağlamaktadır. Bahsedilen bu başlıca avantajlar, ev otomasyon sistemlerinin önemini günden güne artırmakta ve kullanıcıların konforlu bir kullanım deneyimi yaşamalarını mümkün hale getirmektedir.

Günümüzde, tüm insanlar tarafından kullanılmasa da, otomasyon sistemlerinin ve akıllı ev/bina uygulamalarının gelecekte tüm dünya için daha da önem arz edileceği düşünülmektedir. Yakın bir gelecekte karşımıza daha sık çıkacak olan *Nesnelerin Interneti (IoT)* kavramı ise, çevremizde bulunan tüm elektronik cihazların birbirleri ile iletişime geçerek meydana getirecekleri devasa bir nesne iletişim ağını ifade eden terimdir. Nesnelerin interneti sayesinde, akıllı ev uygulamalarının yanısıra, akıllı çevre ve akıllı şehir gibi otomasyon uygulamaları da hayatımızın bir parçası haline gelecektir.

3. NFC VE OTOMASYON

3.1. NFC Hakkında Genel Bilgiler

NFC kısaca, akıllı telefon, tablet, kart okuyucu, akıllı kart vb. cihazların, birbirlerine belirli bir mesafeye kadar yaklaşarak veya fiziksel temas kurarak, aralarında bir radyo iletişimi başlatmaları için tanımlanmış standartlar kümesi olarak tanımlanır. NFC standartları, iletişim protokolleri ve veri değiş tokuş biçimlerini kapsar ve ISO/IEC 14443 ve FeliCa kartlarını da kapsayan mevcut Radyo Frekansı ile Tanımlama (RFID) standartlarına dayanır [11]. NFC, tıpkı mevcut temassız kart teknolojilerinde olduğu gibi, veri değiş tokuşu için, ISO/IEC 18000-3 standartı ile belirlenmiş, lisanssız olarak kullanılabilen evrensel 13.56 MHz frekans bandında çalışan iki spiral antenin elektromanyetik indüksiyonuna ihtiyaç duyar. Bu elektromanyetik indüksiyon neticesinde sağlanan veri iletişimi hızı 106 kbit/s ile 424 kbit/s arasında değişmektedir. Veri transfer hızı yeterince yüksek olmadığı için mobil cihazlar arasında büyük boyutlu verilerin NFC ile iletilmesi verimli bir yöntem değildir. Böyle durumlar için, mobil işletim sistemleri, tüm verinin NFC ile iletilmesi yerine, NFC etkileşimi ile etkinleştirilen Bluetooth veya Wi-Fi Direct bağlantısı ile iletilmesini sağlayan protokoller sunar (ör. Android Beam, S Beam). Böylece, sadece basit bir NFC etkileşimi ile Bluetooth, Wi-Fi Direct gibi bağlantıların konfigürasyonu otomatik olarak tanımlanır ve cihazlar bu protokoller üzerinden birbirlerine yüksek hızlarda veri transferi yapabilir.

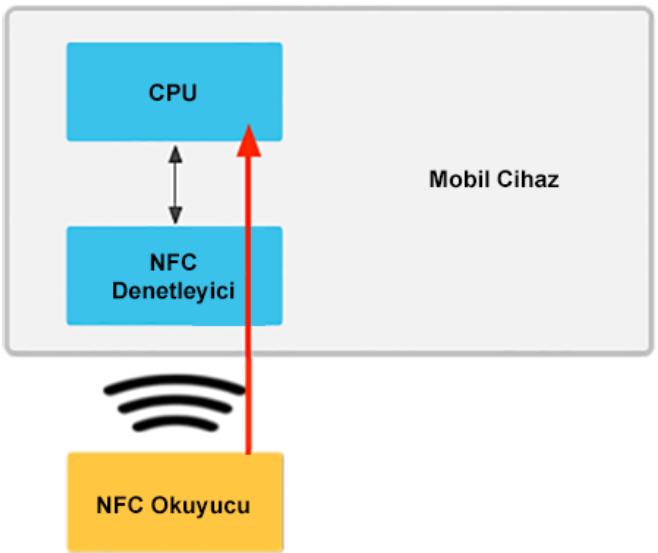
NFC iletişiminde, teoride 10 cm olarak beliritilmiş olmasına rağmen, indüksiyonun kayıpsız ve kararlı bir şekilde gerçekleşebilmesi için öngörülen mesafe pratikte 4 cm civarındadır. Operasyon mesafesinin çok kısa olması NFC'nin güvenli bir iletişim teknolojisi olmasını sağlayan en önemli etkenlerden birisidir. NFC ile haberleşen cihazlar birbirlerine çok yakın olmak zorunda olduğu için, daha uzun mesafeli iletişim yöntemlerinde karşılaşılan, sinyal izleme, sinyal bozma, veri çalma, araya girme, veri bütünlüğü değiştirme gibi ataklar söz konusu değildir. Özellikle mobil ödeme ve kimlik tanımlama gibi uygulamalarda güvenlik öncelikli konudur. NFC etkileşimi için gereken mesafenin çok kısa olmasının yanında, güvenlik seviyesini artırmak için mobil cihazlar ekranları kilitli durumdayken herhangi bir NFC etkileşimine girmez ve veri transferi yapmaz. Böylece cihaz sahibinin bilgisi dışında bir iletişim gerçekleşmez. Ayrıca, iletişim mesafesinin kısa olması, klasik RFID etiketlerini ve cihazlarını okuma yöntemlerine göre çok daha az güç tüketimi gerçekleşmesine

neden olur. Bu nedenle güç tüketiminin önemli bir kriter olduğu sistemlerde de önemli avantaj sağlamaktadır. NFC'nin, Bluetooth ve Wi-Fi donanımlarına karşı avantaj sağladığı diğer önemli bir konu ise NFC donanımlarının daha basit olması sayesinde gerçekleşen daha düşük üretim maliyetleridir.

Günümüzde piyasaya sunulan akıllı telefon ve tablet gibi mobil cihazların birçoğunda dahili NFC özelliği bulunmaktadır. Farklı kullanım senaryolarındaki ihtiyaçları karşılamak amacıyla NFC özellikli cihazlar, ISO/IEC 18092 NFC IP-1 ve ISO/IEC 14443 temassız akıllı kart standartlarını temel alan 3 farklı operasyon modunda çalışabilmektedir [12]. Bu modlar şu şekilde listelenir;

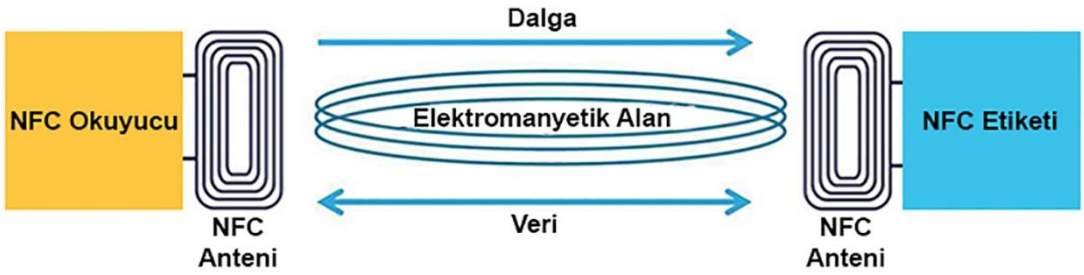
- NFC Kart Emülasyon Modu (HCE)
- NFC Okuma/Yazma Modu (Reader/Writer)
- NFC Eşler Arası Modu (P2P)

NFC kart emülasyon modunda, NFC okuyucu tarafından okunan mobil cihaz, bir temassız kart gibi davranır. Şekil 3.1'de görüldüğü gibi, NFC okuyucudan gelen veri, NFC denetleyici tarafından uygulamanın çalıştığı işlemciye yönlendirilir [12]. Bu veri, içerisinde yer alan AID (Application Identification) ile belirtilmiş mobil uygulama tarafından karşılanır. Uygulama altında çalışan bir servis prosedürü tarafından veri işlenir ve NFC okuyucuya prosedür tarafından belirlenen cevap dönülür. Kimlik tanımlama, ödeme ve biletleme sistemi uygulamalarında NFC özellikli mobil cihazların bu özelliğinden faydalanılır. NFC kart emülasyon (HCE) modu, Android mobil işletim sisteminde 4.4 versiyonundan itibaren desteklenmektedir [13].



Şekil 3.1. NFC HCE Operasyon Modu Şeması

Çalışma şeması Şekil 3.2'de gösterilen NFC okuma/yazma modunda, NFC okuyucu veya NFC özellikli bir mobil cihaz, iletişimi tetiklemek için NFC Forum standartlarına uygun pasif bir temassız kart veya NFC etiketine doğru dalga yayılımı yapar. Dalga NFC etiketinin anteni tarafından yakalanarak etiket içineki elektronik bileşenlere iletilir. Bu etkileşim sonucu, operasyon bir okuma operasyonu ise etiketteki EPROM bellek içerisinde kaydedilmiş veriyi taşıyan sinyal NFC okuyucunun antenine iletilir. Yazma operasyonunda ise, yine NFC okuyucu veya NFC özellikli bir mobil cihaz tarafından gönderilen dalga ile iletişim tetiklendikten sonra, etikete iletilen veriler EPROM belleğin içerisine yazılır ve işlemin başarılı olup olmadığına dair veriyi taşıyan sinyal NFC okuyucuya veya NFC özellikli mobil cihaza iletilir. Yani NFC okuyucular ve NFC etiket/kart okuyabilen mobil cihazlar otomatik olarak yazma yeteneğine de sahiptir. Okuma ve yazma işlemlerini gerçekleştirmek için ilgili platformlar için geliştirilmiş yazılımların çalışır durumda olması gerekmektedir. Bilgisayara bağlı bir NFC okuyucu, masaüstü uygulaması aracılığıyla okuduğu veriyi işler ve ilgili veriyi etikete/karta yazar. Mobil işletim sistemi ile çalışan cihazlarda ise etiketten veya karttan okunan verinin anlamlı bir veriye dönüşmesi için veri içindeki protokole uygun bir uygulama kurulmuş olmalıdır. Örneğin, okunan etikette/kartta bir url adresi varsa (http vb.), kullanıcının seçtiği tarayıcı uygulaması ile bu web adresi açılır. Okunan etikette/kartta bir konum bilgisi varsa, yine kullanıcının seçtiği bir harita uygulaması ile bu konum açılır. Basit veri paylaşımı, bilgi ve eğlence servisleri, multimedya servisleri gibi uygulamalar NFC okuma/yazma modunda çalışan uygulama örnekleri olarak verilebilir.



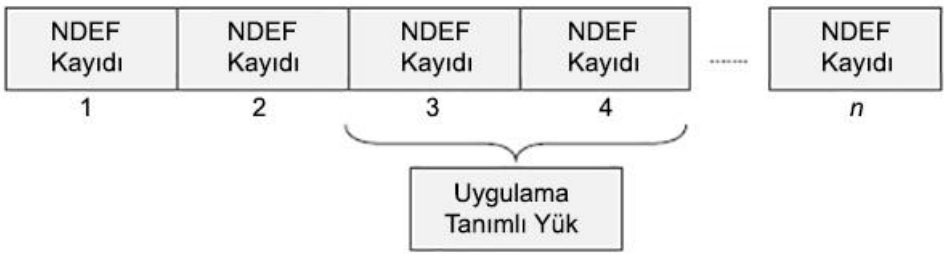
Şekil 3.2. NFC Reader/Writer Operasyon Modu Şeması

Eşler arası iletişim protokolü iki veya daha fazla istemci arasında veri paylaşmak için kullanılan bir ağ protokolüdür. NFC eşler arası modunda ise iki mobil cihaz arasında direk veri alışverişi sağlayan iletişim kurulur. Bu iletişim, ISO/IEC 18092 standardı ile tanımlanmıştır [14]. İletişimin başlaması için mobil cihazlardaki NFC

antenlerinin birbirlerine yeteri kadar yaklaştırılması veya dokundurulması gerekmektedir. Mobil cihazların NFC okuma/yazma veya NFC kart emülasyon modunda olmasına gerek yoktur. Android işletim sistemi'nde NFC donanımları ile eşler arası veri aktarma protokolü Android Beam olarak adlandırılmıştır. İki mobil cihaz arasında etkileşimin meydana gelmesi için bazı ön koşulların sağlanmış olması gereklidir [15]. Bu koşullar kısaca şöyle listelenebilir;

- Her iki mobil cihazda da Android Beam özelliğinin aktif edilmiş olması gerekmektedir.
- İletişimi başlatacak olan cihazda ilgili mobil uygulama ön planda çalışıyor olmalıdır.
- Veri aktarılacak olan mobil cihazın ekranı kilitli olmamalıdır.
- Mobil cihazlar birbirine dokundurulduğu sırada etkileşimi başlatan mobil cihazın ekranında Şekil 5.16'de görünen "Beam'e dokun" mesajına dokunulmalıdır, böylece veri transferi başlar.

NFC okuyucu bir cihaz ile NFC etiketi üzerindeki verinin okunması veya iki NFC uyumlu cihaz arasında gerçekleşen P2P veri transferi, NFC Forum tarafından standardı belirlenen NDEF (NFC Data Exchange Format) ismiyle tanımlanmış bir mesaj biçiminde yapılır. Bu mesaj Şekil 3.3'te görüldüğü gibi bir veya daha fazla sayıda NDEF kayıdından oluşur [16]. Bu kayıtların içinde, aktarılan verinin içeriği, tipi ve hangi uygulama ile kullandırılmak istendiği bilgileri yer alır.



Şekil 3.3. NDEF Mesajının Yapısı

Yukarıda bahsedilen yetenekleri ve avantajları sayesinde NFC, mobil ödeme, elektronik biletleme, kimlik saptama-doğrulama, sosyal ağ gibi kategorilerde çeşitli uygulamaların ortaya çıkmasını sağlayan, son zamanlarda popülaritesi artan bir teknoloji haline gelmiştir. Bu teknolojinin geliştirilmesini, standartlaştırılmasını ve kullanımının yaygınlaştırılmasını, 2004 yılında Nokia, Philips and Sony tarafından kurulan ve daha sonra bir çok teknoloji şirketinin de dahil olduğu NFC Forum isimli

bir organizasyon üstlenmiştir [17]. Bu organizasyon, günümüzde NFC standartlarının geliştirilmesi ve NFC'nin daha fazla uygulama alanlarında etkin bir şekilde kullanılabilmesi için çalışmalarına devam etmektedir.

3.2. Otomasyon Sistemlerinde NFC Uygulamaları

NFC konusunda daha önce yayımlanmış yerli ve yabancı kaynaklı çalışmaların genellikle ödeme, eğlence, kimlik ve profil tanımlama gibi konular üzerine olduğu görülmektedir. Chen, Pan ve Li [18] NFC özellikli mobil cihazlar ve NFC etiketler kullanarak interaktif bir multimedya sistemi önermişlerdir. Her birine farklı fonksiyonlar atanmış NFC etiketlerini, mobil cihazları ile okutarak sistemdeki multimedya cihazlarıyla bağlantı kurabilecekleri, müzik ve video gibi medya dosyalarını oynatabilecekleri senaryolar sunmuşlardır. Basili, Liguori ve Palumbo [19] turistlerin kullanması amacıyla NFC özellikli mobil cihazlarda çalışan ve kullanıcıya bilgi kaynağı, mobil ödeme, mobil biletleme, üyelik, sadakat yönetimi ve konum bazlı servisler sunan bir mobil uygulama önermişlerdir. Çalışmalarının, mobil bilgi sistemleri ile ilgilenen akademik çevrelere ve benzer servisler sunabilecek telekomünikasyon operatörlerine yol gösterici olmasını amaçlamışlardır.

NFC'nin sunduğu bütün bu olanaklar, ev otomasyonlarını da NFC için uygulama alanı olarak düşünebileceğimiz kavramlardan birisi haline getirmiştir. Sayıca fazla olmamakla birlikte NFC teknolojisinin akıllı evlere ve ev otomasyonu sistemlerine entegre edilmesini ele alan akademik çalışmalara da rastlanılmaktadır. Bu konuda yayımlanmış bir konferans makalesinde, De Luca, Lillo, Mainetti, Mighali, Patrono ve Sergi [20] geliştirdikleri bir mobil uygulama ile, KNX protokolü tabanlı bir ev otomasyon sisteminin, klasik doğrulama yöntemlerine ek olarak mobil cihazların NFC okuyucu aracılığıyla sisteme giriş yapabilecekleri bir senaryo ile yönetilmesini önermiştir. Temel aldıkları [9]'daki sistemde mevcut güvenlik katmanlarına ek olarak mobil cihazlara ait özgün kimlik verilerinin bir NFC okuyucu üzerinden doğrulama sunucusuna transfer edildiği bir senaryo ele almışlar ve NFC'nin güvenli bir doğrulama aracı olarak kullanılabileceğini savunmuşlardır. Bir başka konferans makalesinde ise, Chandrakar, Kaul, Mohan**,** Sai Vamsi ve Prabhu [21] akıllı ev sistemine dahil edilen bir NFC okuyucu ile, her kullanıcının NFC etiketlerinde kayıtlı profillerini sisteme tanıtararak, evdeki aydınlatma düzeninin bu profile göre ayarlanmasını sağlayan bir sistem önermiştir. Aynı zamanda stüdyo gibi yerler için de kullanımının uygun olduğu düşünülen tasarım, NFC'nin güvenliğinden ziyade

sunduğu kullanım kolaylığını ve sıradan bir RFID ürününe göre sahip olduğu düşük güç tüketimini öne çıkarmayı amaçlamıştır.

Ev otomasyonu haricinde, farklı otomasyon alanlarında yapılan çalışmalarda da NFC kullanımına rastlamak mümkündür. Örneğin, Lekic ve Mijanovic [22] akaryakıt dağıtım firmaları için bir sistem otomasyonu tasarlamıştır. Önerdikleri NFC tabanlı kimlik tanıma mekanizması, operatörün varlığı olmaksızın yakıt dolumunun otomatik olarak gerçekleşmesini ve tüm işlemlerin kayıt altına alınmasını amaçlamaktadır. Bir başka örnekte, V. Patil , Varma, Vinchurkar ve B. Patil [23], gelişmekte olan ülkeleri hedef alan, düşük maliyetli bir NFC tabanlı sağlık takip otomasyonu önermişlerdir. Teknik bilgisi olmayan sağlık personelinin de kolayca kullanabileceği sistem, NFC etiketleri ve NFC özellikli mobil cihazlar aracılığıyla hastaların kolayca tanımlanmasını ve tehlikeli sonuçlara yol açabilecek olası bir karışıklığın önlenmesini amaçlamaktadır.

4. SİSTEMİN MİMARİSİ VE KULLANIM SENARYOLARI

Bu çalışmada, çeşitli otomasyon sistemlerinin kullanımına çözüm olarak önerilebilecek NFC tabanlı giriş ve kullanım kontrollü bir sistem, akıllı ev otomasyonu örneği kullanılarak önerilmektedir. Bu sayede hali hazırda var olan akıllı sistemlerin kullanımına yeni bir bakış açısı getirilmiş olacaktır. Bu kapsamda öncelikle ortaya çıkarılan sistem mimarisi ve bileşenleri incelenecek, daha sonra da kullanım senaryoları ile birlikte sistemin çalışması ele alınacaktır.

4.1. Sistem Mimarisi ve Bileşenleri

Tez çalışması kapsamında geliştirilen sistemin fiziksel ve/veya donanımsal bileşenleri şunlardır:

- Doğrulama sunucusu
- NFC özellikli mobil cihaz (akıllı telefon veya tablet)
- NFC okuyucu
- Gözlemlenen ve kontrol edilen cihazlar
- Gateway (ağ geçidi)
- NFC etiket

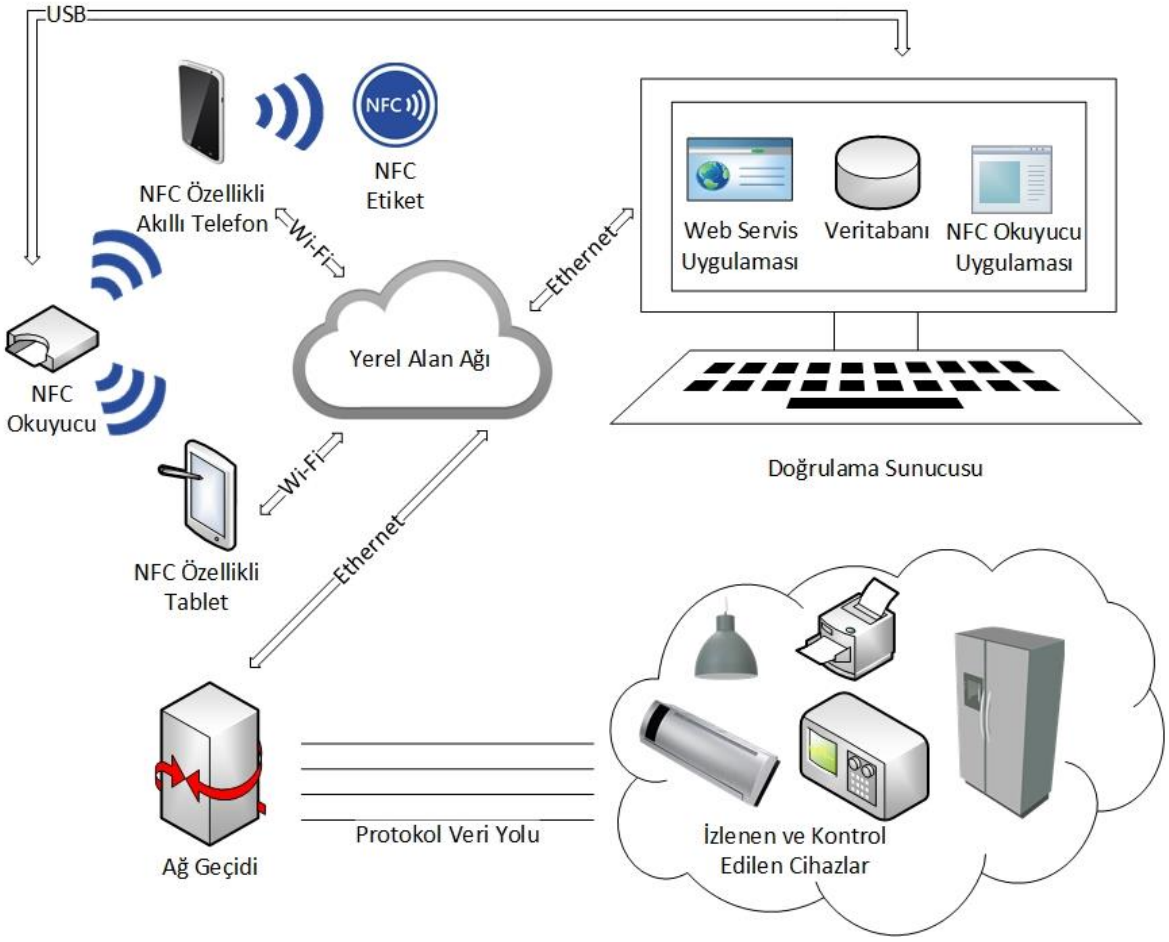
Geliştirilen sistemin yazılımsal bileşenleri ise şunlardır:

- Mobil cihaz uygulaması
- Web servis uygulaması
- NFC okuyucu masaüstü uygulaması
- Veritabanı
- Simülasyon uygulaması

Yukarıdaki bileşenlerden ortaya çıkan sistem mimarisi Şekil 4.1'de gösterilmiştir. Sistemdeki bütün bileşenlerin aynı yerel ağda (LAN) bulunduğu varsayılmaktadır. Bu sayede internet erişiminin mümkün olmadığı senaryolarda da mobil cihazlar ile doğrulama sunucusu veya izlenen ve kontrol edilen cihazlar arasında iletişim mümkün olmaktadır.

Tüm ev otomasyon sistemlerinde olduğu gibi, bu sistemde de anlık olarak izlenen ve kontrol edilen klima, ısıtıcı, lamba gibi cihazlar sistemin en önemli kısımlarını oluşturmaktadır. Bu cihazlar ile yapılan iletişim genellikle KNX, ZigBee ve EHS gibi belirli standartlarla tanımlanmış protokoller üzerinden gerçekleştirilir [9,5,24]. Bu tez

kapsamında odak noktası, NFC teknolojisinin mevcut sistemlere sağlayacağı yenilikler ve kullanım senaryoları olması nedeniyle, tasarlanan sistem herhangi bir protokolden bağımsız olarak ele alınmıştır. Bu nedenle, cihazlarla iletişim kurmak için jenerik bir gateway tasarıma dahil edilmiştir.



Şekil 4.1. Sistem Mimarisi

Kullanıcıların sistemdeki cihazların anlık durumlarını gözetleyebilmesi ve kontrol edebilmesi için, akıllı telefon veya tablet gibi yeni nesil gelişmiş mobil işletim sistemleri ile çalışan NFC özelliğine sahip mobil cihazlara ve bu mobil cihazlarda kullanılacak bir mobil uygulamaya ihtiyaç duyulmaktadır. NFC özelliğine sahip çok sayıda mobil cihazın Android mobil işletim sistemi ile çalışması ve bu işletim sisteminin tezde önerilen bütün senaryolar için gerekli yazılım geliştirme ihtiyacını karşılaması nedeniyle, sistemdeki mobil uygulama Android platformu için geliştirilmiştir.

Öncelikli görevi sistemdeki mobil cihazların özgün kimlik bilgisini okumak olan donanım bileşeni NFC okuyucudur. Kullanıcının mobil cihazını NFC okuyucuya

yeterince yaklaştırması halinde okuyucu, mobil cihaz ile etkileşime girer ve elde ettiği veriyi doğrulama sunucusuna aktarır. Hızlı ve güvenli bir veri iletişimi sağlamak için NFC okuyucu, doğrulama sunucusuna USB bağlantısı ile direk bağlıdır. Bu tez çalışması kapsamında kullanılan ve teknik özellikleri [25]'de belirtilen NFC okuyucu Şekil 4.2'de gösterilmiştir.

Şekil 4.2. NFC Okuyucu

Sistemin diğer bir önemli bileşeni de, ihtiyaç duyulan tüm verileri barındıran veritabanınına ev sahipliği yapan ve kullanıcıların sisteme giriş yapabilmesi için gerekli doğrulama hizmetini sağlayan doğrulama sunucusudur. Doğrulama sunucusu olarak kullanılan bilgisayar, işletim sistemi olarak Windows Server 2012 R2 versiyonu ile çalışan bir sunucu bilgisayardır. Windows Server 2012 R2 işletim sistemi ve MSSQL Server 2014 yazılımının çalışabilmesi için gerekli minimum koşulları sağlayan herhangi bir bilgisayar, doğrulama sunucusu olarak kullanılabilir. Doğrulama sunucusundan istenen operasyonları yerine getirmesi için, sunucu üzerinde üç farklı yazılım bileşeni bulunmaktadır.

Doğrulama sununucusu üzerinde bulunan yazılım bileşenlerinden birincisi, ASP.NET programlama dili ile geliştirilmiş, IIS (Internet Information Services) üzerinde çalışan bir web servis uygulamasıdır. Temel fonksiyonu, mobil uygulama üzerinden gönderilen http isteklerine karşılık olarak gerekli disk ve veritabanı okuma/yazma operasyonları ile elde edilen verileri dönmek ve bu sayede kullanıcıların ve mobil cihazların doğrulanabilmesi için gerekli hizmeti sunmaktır.

İkinci yazılım bileşeni, sistem dahilinde ihtiyaç duyulan tüm verilerin depolandığı, Microsoft SQL Server uygulaması üzerinde çalışan bir veritabanıdır. Bu veritabanında, kullanıcı ve cihazlara ait profil verileri, ev otomasyonu oturum kayıtları ve cihaz kullanımlarını inceleyebilmek amaçlı kaydedilen istatistik verileri bulunmaktadır. Hem web servis uygulaması, hem de NFC okuyucu uygulaması bu veritabanına erişim hakkına sahiptir.

Üçüncü bileşen ise, NFC okuyucu tarafından okunan, kullanıcıların mobil cihazlarına ait özgün kimlik verisini ve giriş denemesi anındaki zaman bilgisini doğrulama sunucusu veritabanına kaydeden, C#.NET programlama dili ile geliştirilmiş masaüstü konsol uygulamasıdır.

Tasarlanan sisteme, isteğe bağlı olarak bir NFC etiketi de dahil edilmiştir. Bu etiket kullanıcının mobil cihazı tarafından okununca, ev otomasyonu mobil uygulamasının otomatik olarak başlatılmasını sağlar. Bu konu hakkında yayımlanmış bir makalede, S. Lee, T. Lee, Kim ve Hong [26] bir NFC etiketine SSID ve parola bilgisi içeren yerel kablosuz ağ verilerini kaydedip ilk defa kablosuz yerel ağa girmek isteyenler için pratik bir yöntem önermişlerdir. Bu sayede, kablosuz yerel ağa hiç bağlanmamış mobil cihazlar, NFC etiketindeki veriler sayesinde kablosuz ağa otomatik olarak bağlanabilirler. Bu kablosuz ağa bağlantının gerçekleşmesini sağlamak için bir mobil uygulamanın, etiketten okunan verileri işleyip mobil cihazın kablosuz bağlantı ayarlarında gerekli güncellemeleri yapması gerekmektedir.

4.2. Sistemin Genel İşleyişi

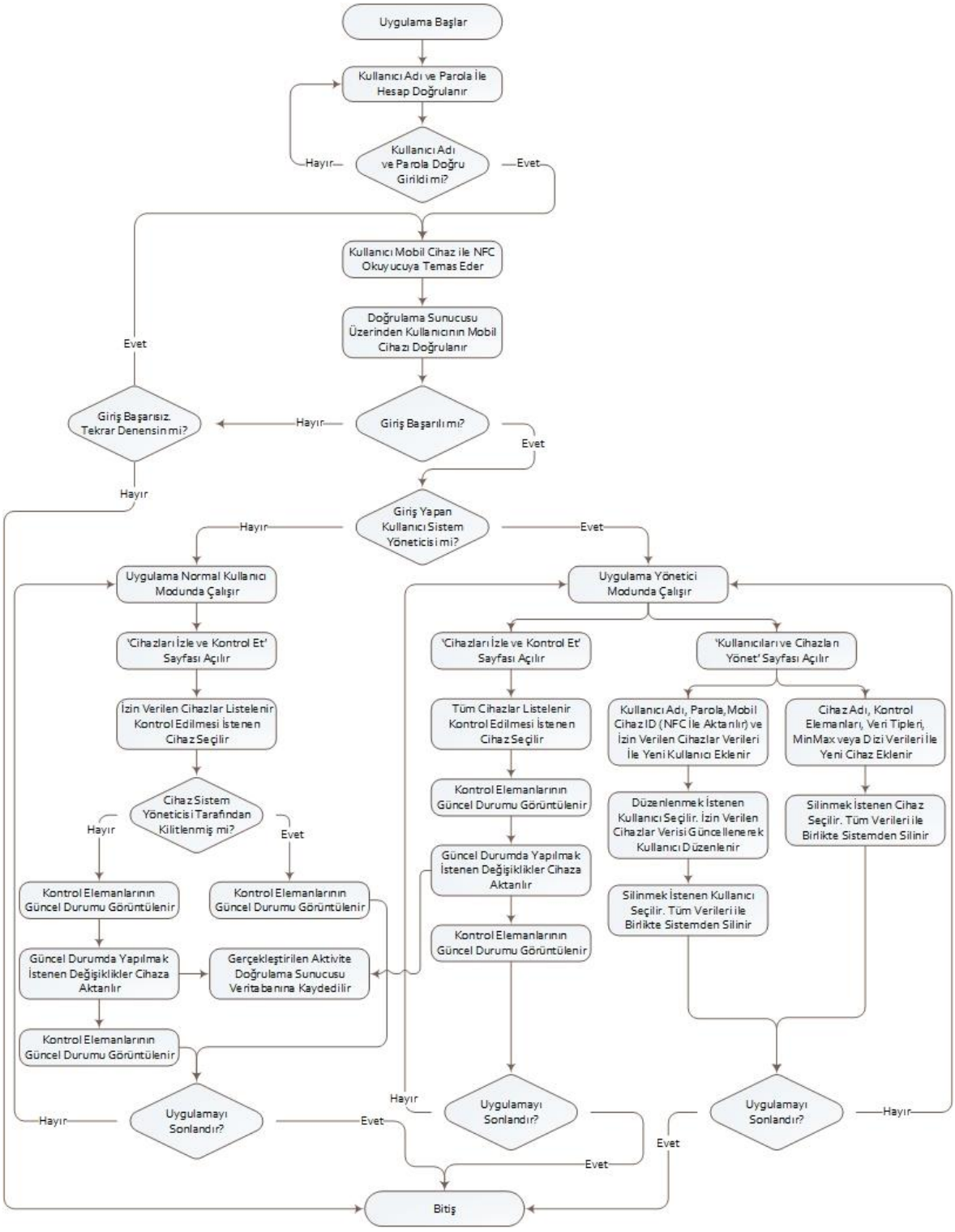
Tez çalışması kapsamında, detayları Bölüm 4.3'te anlatılacak olan iki farklı kullanım senaryosu ele alınacaktır.

İlk önerilen kullanım senaryosunda, sistem yöneticisi ve normal kullanıcılar, yani tüm kullanıcılar öncelikle kullanıcı adı ve kişisel parolaları ile hesaplarını doğrulattıktan sonra, [20]'de önerilen yönteme benzer bir şekilde, mobil cihazlarını sadece NFC okuyucuya okutarak mobil cihaz özgün kimlik numaralarını doğrulama sunucusuna doğrulattıktan sonra sisteme giriş yaparlar ve mobil cihazları ile sistemdeki cihazların anlık durumlarını izleyip cihazları kontrol ederler. Bu senaryoda, normal kullanıcılardan daha fazla yetkiye sahip olan sistem yöneticisi, sisteme yeni cihaz ekleme, mevcut cihazı silme, sisteme yeni kullanıcı ekleme, mevcut kullanıcıyı düzenleme ve silme operasyonlarını gerçekleştirebilir.

Önerilen ikinci kullanım senaryosunda ise normal kullanıcılar sisteme giriş için doğrulama sunucusu tarafından onay almak zorunda değildir. NFC teknolojisinin sağladığı P2P olarak bilinen eşler arası iletişim özelliği ile sistem yöneticisi, sistemde tanımlı olan dilediği bir kullanıcıya giriş onayı verebilir. Bunu gerçekleştirmeden önce sistem yöneticisinin ilk senaryoda olduğu gibi oturum açması gerekmektedir. Daha sonra, sistem yöneticisi seçtiği bir kullanıcıya, belirli cihazları belirli süre boyunca izleyip kontrol edebileceği eşler arası oturum verisini mobil cihazı üzerinden NFC aracılığı ile kullanıcının mobil cihazına aktarır. Kullanıcı bu oturumu başlatmayı kabul ederse yeni bir oturum başlamış olur. Cihazların izlenmesi ve kontrol edilmesi açısından ilk senaryo ile herhangi bir farklılık yoktur. Belirlenen süre bitince oturum otomatik olarak sonlanır.

4.3. Genel Kullanım Senaryosu

Genel kullanım senaryosu için temel alınan akış diyagramı Şekil 4.3'te gösterilmiştir. Diyagramda hem sistem yöneticisi hem de normal kullanıcılar için öngörülen akış birlikte ele alınmıştır.



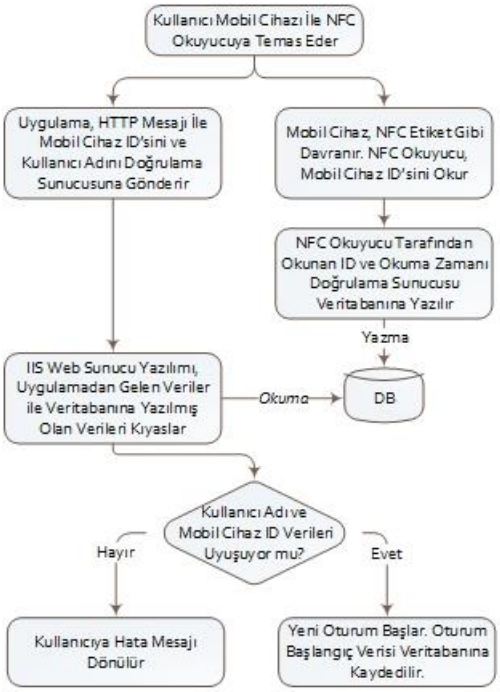
Şekil 4.3. Genel Kullanım Senaryosu Akış Diyagramı

4.3.1. Kullanıcı Hesabının ve Mobil Cihaz Kimliğinin Doğrulanması

Genel kullanım senaryosunda, sistem yöneticisi ve normal kullanıcı aynı şekilde sisteme giriş yapar. Akış diyagramından da görüldüğü üzere, kullanıcı hesabının ve mobil cihaz kimliğinin doğrulanması, toplamda iki aşamalı güvenlik adımından

oluşur. İlk aşamada, kullanıcı manuel olarak veya sisteme dahil edilmişse NFC etiketi mobil cihazına okutarak, mobil cihaz uygulamasını başlatır. Giriş sayfasında, kullanıcı adını ve kişisel parolasını girerek, hesabını doğrulama sunucusu tarafından doğrulatır. Bu hesap doğrulatma işlemi, sistemde tanımlı mobil cihazların başkaları tarafından kullanılmasını kısıtlamak amaçlı olarak giriş algoritmasına dahil edilmiştir. Hesap doğrulatma işlemi başarısız ise kullanıcıya hata mesajı bildirilir. İşlem başarılı ise yine kullanıcıya gerekli bildirim yapılır ve ikinci aşamada kullanıcının mobil cihazını NFC okuyucuya okutması beklenir. Kullanıcı mobil cihazını NFC okuyucuya yeteri kadar yaklaştırınca NFC üzerinden veri alışveriş protokolü aktif hale gelir. Mobil uygulamada çalışan servis sayesinde, mobil cihaz NFC okuyucu tarafından okunduğunda bir NFC etiketi emülasyonu yapar, yani etiket gibi davranır. NFC okuyucu mobil cihazdan, mobil cihaz unique id (özgün kimlik numarası) verisini ister. Mobil uygulama, cihaza ait özgün kimlik verisini okuyucuya cevap olarak döner ve NFC okuyucu uygulaması, bu veriyi doğrulama sunucusu veritabanına kaydeder. Mobil cihaz NFC okuyucudan uzaklaştırıldığı anda, mobil uygulama doğrulama sunucusuna, http üzerinden giriş isteği gönderir. Bu istek parametre olarak kullanıcı adını ve cihazın özgün kimlik bilgisini taşır. Http isteği ile gelen kimlik verisi ile NFC okuyucu uygulaması tarafından veritabanına kaydedilmiş olan veriler, web servisi uygulaması tarafından karşılaştırılır. Veriler aynı zaman aralığında (10 sn. kadar) oluşmuşsa ve tam olarak uyuşuyorsa, doğrulama sunucusu geri dönüş cevabı olarak kullanıcının sisteme girişini onaylayan mesajı gönderir ve bu girişi veritabanına kaydeder. Bu giriş algoritmasının akış diyagramı Şekil 4.4'te gösterilmiştir.

Sisteme giriş için kullanılan mobil cihaz özgün kimlik bilgisi, tüm GSM cihazlarına özgü olan IMEI numarası, tüm Android mobil cihazlarına özgü olan Android ID veya kablosuz adaptör MAC adresi verilerinden birisi olabilir. Tasarlanan sistemde, daha güvenilir ve değiştirilmesi daha zor olması nedeniyle özgün kimlik verisi olarak IMEI tercih edilmiştir. Ancak, IMEI numarasına sahip olmayan cihazların da sistemde kullanılabilmesini sağlamak için, uygulama, mobil cihazın IMEI verisine sahip olmadığını tesbit ettiği anda Android ID veya kablosuz adaptör MAC adresi verilerinden birisini otomatik olarak özgün kimlik numarası olarak kullanmaktadır.



Şekil 4.4. Sistem Giriş Algoritması Akış Diyagramı

Sisteme giriş yaptıktan sonra normal kullanıcılar ve sistem yöneticisi uygulamanın ana sayfasına yönlendirilir. Ana sayfadaki erişim butonları kullanıcı tipine göre aktif veya pasif durumda görünür. Normal kullanıcılar için sadece “Cihazlarımı İzle ve Kontrol Et” ve “Şifremi Değiştir” sayfalarının linkleri aktif iken, sistem yöneticisi için bunlara ek olarak “Cihazları ve Kullanıcıları Yönet” ve “P2P Oturum Başlat” sayfalarının erişim butonları aktif durumdadır.

4.3.2. Cihazların İzlenmesi ve Kontrol Edilmesi

Normal kullanıcılar ve sistem yöneticisi, sistemdeki cihazların anlık durumlarını görmek ve cihazları kontrol etmek için “Cihazlarımı İzle ve Kontrol Et” sayfasına erişir. Bu sayfada kullanıcı tarafından izlenme ve kontrol edilme yetkisi bulunan cihazlar listelenir. Sistem yöneticisi ekranında ise sistemdeki tüm cihazlar listelenir. Kullanıcı, izleyip kontrol etmek istediği cihazı seçtikten sonra cihazdaki kontrol elemanları listesi, kontrol elemanlarının anlık değerleri ve bu değerleri değiştirebilmek için gerekli arayüz bileşeni ekranda görünür. Kullanıcı bu sayfada cihazın kontrol elemanlarının güncel durumunu değiştirebileceği gibi cihazı kapatıp açabilir. Eğer cihaz sistem yöneticisi tarafından kilitlenmiş durumda ise kullanıcı sadece kontrol elemanlarının güncel durumunu gözlemleyebilir. Bu cihaz kilitleme özelliği sayesinde güvenlik, onarım, bakım vb. gibi cihazın kullanılmaması gereken durumlarda kullanımı engellenmiş olur. Tez kapsamında önerilen sistemde herhangi

bir otomasyon standartına bağlı kalınmadığı için, cihazların izlenmesi ve kontrol edilmesi, Bölüm 5.5'te detayları açıklanmış olan bir simülasyon uygulaması ile simüle edilmiştir.

4.3.3. Kullanıcı Profillerinin Yönetilmesi

Daha önce de yazıldığı gibi kullanıcı profillerinin yönetimi sadece sistem yöneticisinin yetkisindedir. Normal kullanıcılar, mobil uygulamada bu sayfalara erişemez. Sistem yöneticisi, uygulamada sisteme giriş yaptıktan sonra ana sayfadaki "Kullanıcıları ve Cihazları Yönet" bağlantısına tıklayarak kullanıcıları yönetmek için butonların bulunduğu sayfaya yönlendirilir. Bu sayfadaki butonlar ile erişilen sayfalar üzerinde, sisteme yeni kullanıcı ekleme, sistemdeki mevcut kullanıcıyı düzenleme ve mevcut kullanıcıyı sistemden silme işlemleri gerçekleştirilir.

4.3.4. Cihaz Profillerinin Yönetilmesi

Cihazların yönetimi de tıpkı kullanıcıların yönetimi gibi sadece sistem yöneticisinin yetkisindedir. Sistem yöneticisi, uygulamada sisteme giriş yaptıktan sonra ana sayfadaki "Kullanıcıları ve Cihazları Yönet" bağlantısına tıklayarak kullanıcı ve cihazları yönetmek için butonların bulunduğu sayfaya yönlendirilir. Bu sayfadaki butonlar ile erişilen sayfalar üzerinde, sisteme yeni cihaz ekleme ve mevcut cihazı sistemden silme işlemleri gerçekleştirilir.

4.4. Eşler Arası Kullanım Senaryosu

Bu kullanım senaryosu, bu konuda yapılan çalışmalarda benzerine rastlanılmamış bir yöntemle, sistem yöneticisine ve kullanıcılara bir takım kolaylıklar ve avantajlar sunmak için önerilmiştir. Bunları şu şekilde sıralayabiliriz.

- Kullanıcının veritabanında kayıtlı mevcut profili değiştirilmeye gerek duyulmadan, belirli cihazları kontrol edebilmesi ve normal zamanlarda kontrol edemediği cihazları kontrol edebilme yetkisine sahip olması,
- Kullanıcının, kendisine kontrol edilme yetkisi verilen cihazları sadece belirli bir süre boyunca kontrol edebilmesi,
- Kullanıcının, uygulama üzerinde hesabına giriş yapmaya ve mobil cihazını NFC okuyucuya okutmaya gerek duymadan, sadece basit bir P2P etkileşimi ile pratik bir şekilde sisteme giriş yapabilmesi.

Bu senaryoyu genel kullanım senaryosundan ayıran en önemli özelliği, sisteme giriş yapmak isteyen kullanıcının doğrulama sunucusu tarafından değil, sistem yöneticisi tarafından doğrulanmasıdır. NFC teknolojisinin sunduğu 3 farklı operasyon modundan birisi olan eşler arası modu bu senaryoda önemli bir rol alır.

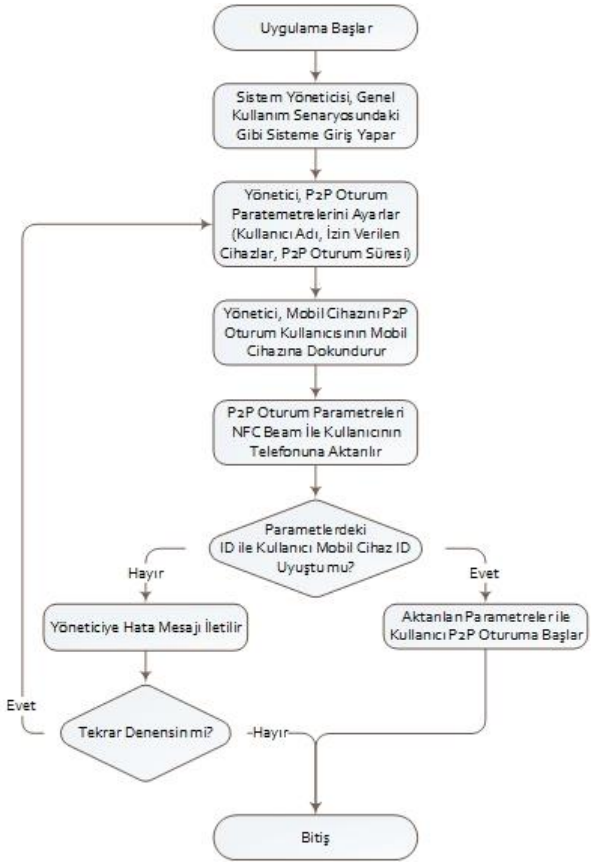
Eşler arası kullanım senaryosu için temel alınan sistem yöneticisi akış diyagramı Şekil 4.5'te gösterilmiştir. Normal kullanıcı için tasarlanan akış diyagramı ise Şekil 4.6'da gösterilmiştir.

Sistem yöneticisi için akış diyagramı Şekil 4.5'te verilen algoritmaya göre, sistem yöneticisini doğrulayacak başka bir mekanizma olmadığı için, sistem yöneticisi genel kullanım senaryosunda olduğu gibi sisteme giriş yapar. Anasayfadan "P2P Oturum Başlat" sayfasına erişir. Bu sayfada, aşağıdaki parameterleri ayarlar:

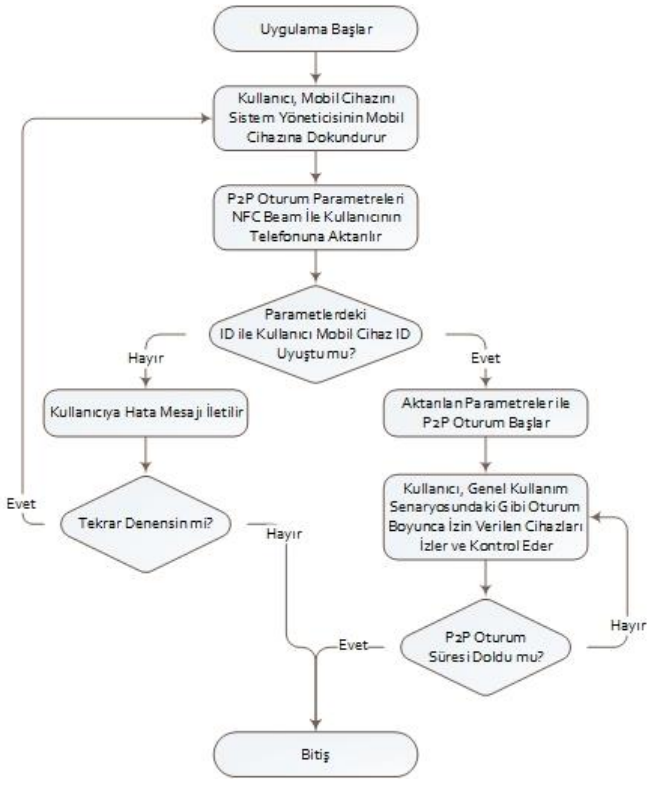
- Kullanıcı adı: P2P oturum başlatacak olan kullanıcı sistemde kayıtlı kullanıcılar arasından seçilir.
- İzin verilen cihazlar: P2P oturum başlatacak olan kullanıcının oturum boyunca izleyip kontrol etmeye yetkisi olduğu cihazların listesidir.
- Oturum süresi: P2P oturumun başlangıcından itibaren otomatik olarak bitişine kadar geçen süredir.

Bu parametreler ayarlandıktan sonra sistem yöneticisi mobil cihazını kullanıcının mobil cihazına yeterince yaklaştırır veya dokundurur. Cihazların NFC antenleri arasında meydana gelen P2P etkileşim ile parametreler kullanıcının mobil cihazına transfer edilir. Bu etkileşimin başarılı bir şekilde gerçekleşmesi için Bölüm 3.2'de bahsedilen ön koşulların sağlanmış olması gerekmektedir.

Gerekli parametreler kullanıcıya aktarıldıktan sonra eğer doğru kullanıcının mobil cihazıyla etkileşim kurulmuşsa sistem yöneticisine işlemin başarılı olduğuna dair, yanlış bir cihazla etkileşim kurulmuşsa veya etkileşim sırasında herhangi bir nedenden dolayı veri transferi gerçekleşemezse işlemin başarısız olduğuna dair bildirim gösterilir.



Şekil 4.5. Sistem Yöneticisi için P2P Kullanım Senaryosu Akış Diyagramı



Şekil 4.6. Normal Kullanıcı için P2P Kullanım Senaryosu Akış Diyagramı

Kullanıcıların eşler arası oturuma dahil olabilmesi için dikkat edilmesi gereken hususlar, mobil uygulamanın daha önceden kurulmuş olması ve işletim sisteminde Android Beam özelliğinin aktif durumda olması gerektiğidir. Kullanıcının mobil cihazı ile sistem yöneticisinin mobil cihazı arasında gerçekleşen P2P etkileşimden sonra kullanıcıda çalışan mobil uygulama transfer edilen verileri kontrol eder. Cihaz doğrulanması gerçekleşmişse kullanıcı ekranında sistem yöneticisinin P2P oturum daveti görüntülenir. Kullanıcı daveti kabul ederse transfer edilen parameterlerle yeni oturum başlar. Eğer kullanıcı daha önceden normal veya eşlerarası doğrulama ile başka bir oturum başlatmışsa o oturum sona erer. Yine sistem yöneticisi ekranında olduğu gibi, yanlış bir cihazla etkileşim kurulmuşsa veya etkileşim sırasında veri transferi başarısız olursa, kullanıcı ekranında işlemin başarısız olduğuna dair bildirim gösterilir.

P2P oturum başladıktan sonra kullanıcılar, kontrol etme yetkisine sahip oldukları cihazları normal kullanım senaryosunda olduğu gibi izleyip kontrol edebilirler. Bu açıdan iki senaryo arasında herhangi bir fark yoktur. Belirlenen süre sona erdikten sonra, oturum otomatik olarak sonlanır ve uygulama kullanıcıyı giriş sayfasına yönlendirir.

5. YAZILIM BİLEŞENLERİ VE GELİŞTİRİLEN UYGULAMALAR

Tez kapsamında önerilen çalışmanın kavramsal ispatını gerçekleştirmek amacıyla tasarlanan sistemin işleyebilmesi ve bilgisayar ortamında simüle edilebilmesi için çeşitli platfomlarda geliştirilen veya kullanılan toplam 5 farklı yazılım bileşeni mevcuttur. Bunlar,

- Doğrulama sunucusu web servis uygulaması,
- Doğrulama sunucusu NFC okuyucu uygulaması,
- Doğrulama sunucusu SQL Server veritabanı,
- Mobil cihaz Android uygulaması,
- Simülasyon uygulaması

olarak sıralanmaktadır. Bu bileşenlerden sadece doğrulama sunucusu SQL Server veritabanı için herhangi bir kodlama yapılmamıştır. Diğer dört uygulama, çeşitli IDE araçlarıyla geliştirilmiş yazılım çözümleridir.

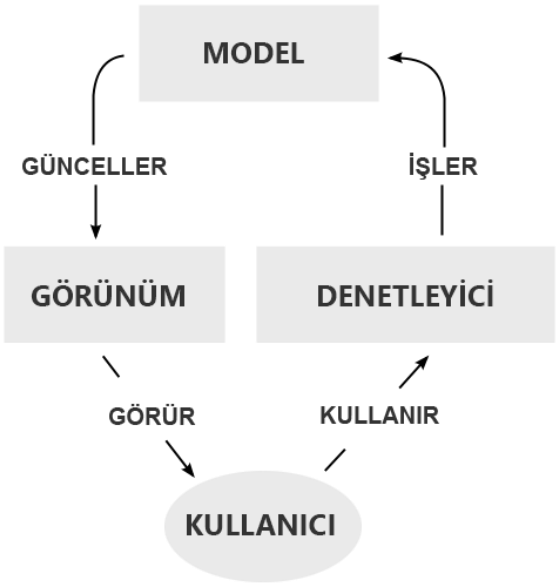
5.1. Doğrulama Sunucusu Web Servis Uygulaması

Bölüm 4.1'de anlatıldığı gibi, temel fonksiyonu, mobil uygulama üzerinden gönderilen http isteklerine karşılık olarak gerekli disk ve veritabanı okuma/yazma operasyonları ile elde edilen verileri dönmek olan bu uygulama, Visual Studio 2013 geliştirme ortamında, MVC mimari örüntüsünde ASP.NET programlama dili kullanılarak geliştirilmiş, IIS üzerinde çalışan bir web uygulamasıdır. ASP.NET MVC, Microsoft tarafından geliştirilmiş Model Denetleyici Görünüm (Model Controller View) örüntüsünü gerçekleştiren çok katmanlı mimari modeli temel alan web tabanlı uygulama çatısıdır [27]. Bu örüntüye göre yazılım birbirinden bağımsız 3 katmana ayrılır:

- Model: Uygulama verisini, mantık ve fonksiyonları içeren bileşendir. Model nesneleri sıklıkla model durumunu bir veritabanına alır ve burada depolar. Örneğin, *User* nesnesi bir veritabanından bilgiler alabilir, bu bilgileri veritabanında çalıştırabilir ve ardından güncelleştirilen bilgileri SQL Server veritabanındaki bir *Users* tablosuna geri yazabilir.
- Denetleyici: Kullanıcıdan gelen veri girişlerini alıp bunları model için bir komuta dönüştüren bileşendir. Örneğin, kontroller kendisine gelen sorgu metinindeki parametreleri işler ve buna karşın bu parametreleri, veritabanını sorgulamak için bu parametreleri kullanabilecek olan model nesnesine iletir.

- Görünüm: Diyagram ve grafik gibi elemanların kullanıcıya sunulduğu uygulama arayüz bileşenidir. Bu arayüz, genellikle model verilerinden oluşturulur. Görünümlere bir örnek, *User* nesnesinin mevcut durumuna dayanan metin kutularını, açılan listeleri ve onay kutularını görüntüleyen, *Users* tablosunun düzenleme görünümü olarak verilebilir.

Bu üç bileşen arasındaki ilişki Şekil 5.1'de olduğu gibi gösterilebilir.



Şekil 5.1. MVC Bileşenleri Arasındaki İlişki

MVC, Web Forms uygulamalarına kıyasla daha hızlı çalışan bir arkaplan hizmeti sağlar. Bu sayede, özellikle geniş kapsamlı projelerde daha yaygın olarak kullanılır. Katmanlı yapısı sayesinde, proje bileşenleri çok daha düzenli olur ve aynı anda farklı katmanların farklı geliştiriciler tarafından geliştirilmesine imkan tanır.

Bu web tabanlı uygulamada mobil cihazdan gelen http isteklerini karşılayan *MainController* isimli sınıf controller görevini üstlenmiştir. Mobil uygulamadan gelen istekler, kullanıcı adı-şifre doğrulama, profil ve cihaz bilgisi alma gibi işlemler için sunucuya iletilir. Bu isteklerin içeriği Bölüm 5.4'te detaylı olarak açıklanacaktır.

MainController sınıfı tarafından karşılanan istekler, *DBOperations* ve *FileOperations* isimli sınıflara yönlendirilir. İsimlerinden de anlaşılacağı gibi, veritabanı ve dosya okuma yazma operasyonlarını içeren bu sınıflar uygulamadaki temel mantık ve veri işleme mekanizmasını çalıştırdığı için örüntüdeki model katmanını oluşturur.

Veritabanı ve dosyalardan okunan verilere göre meydana gelen metot dönüş verileri, istek cevapları olarak mobil uygulamaya dönülür. Gelen cevaplara göre

mobil uygulamadaki ilgili ekrana değişikliklerin yansıyacağı düşünülürse, mobil uygulama arayüzü, örüntüde view katmanı olarak değerlendirilebilir.

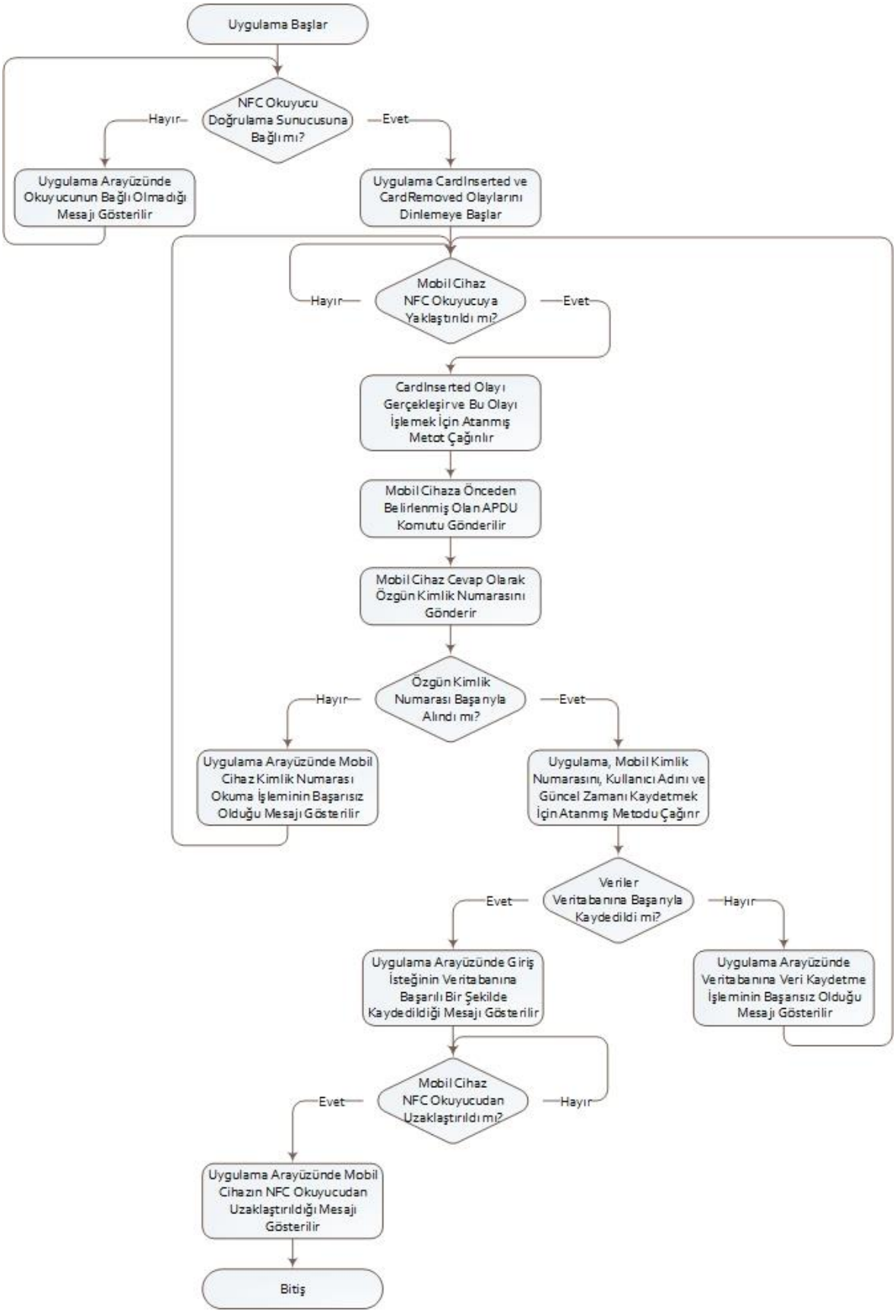
Visual Studio 2013 üzerinde projenin geliştirilmesi tamamlandıktan sonra dosya sistemi yayınlama (file system publish) yöntemi ile kurulum paketi hazırlanır, sunucuya yüklenir ve sunucuda IIS özelliği aktif hale getirildikten sonra uygulama başlatılır.

5.2. Doğrulama Sunucusu NFC Okuyucu Uygulaması

NFC okuyucu uygulaması, doğrulama sunucusunda çalışan NFC Reader Desktop Tool isimli bir masaüstü konsol uygulamasıdır. Uygulama, NFC okuyucunun okuduğu veriyi elde edip, bu verinin kullanıcı doğrulama algoritmasında kullanımını sağlayabilmek için geliştirilmiştir. Visual Studio 2013 geliştirme ortamında C# dili kullanılarak geliştirilmiştir. İhtiyaç duyulmadığı için, Windows Forms veya WPF uygulamaları gibi gelişmiş bir görsel arayüze sahip değildir. Siyah ekran uygulaması olarak da tanımlanan konsol ekranı tipinde arayüzü vardır. Bu arayüzde, NFC okuyucunun sisteme bağlı olup olmadığı, okuma işleminin başarılı olup olmadığı ve okuma zamanları gözlemlenebilir. Uygulama, NFC okuyucu haricinde, klavye vb. herhangi bir giriş biriminden veya bilgisayarda bulunan başka bir uygulamadan veri kabul etmez. Uygulama arayüzü kullanıcılara güvenli bir şekilde bilgi sunmak amacıyla dahil edilmiştir.

NFC okuyucu, USB arayüzünden sisteme bağlandıktan sonra uygulama başlatılır. Uygulama çalışınca ekranda ilk olarak NFC okuyucunun, sisteme olması gerektiği gibi bağlandığını gösteren “ACS ACR122 0 Kart Okuyucusu ile bağlantı kuruldu.” ifadesi yer alır. Eğer okuyucu gerektiği gibi bağlanmamışsa veya henüz aktif hale gelmemişse yani üzerindeki kırmızı led ışık yanmamışsa ekranda “Okuyucu bulunamadı” ifadesi yer alır. Bu durumda bağlantılar kontrol edilmeli ve okuyucunun aktif olduğundan emin olunduktan sonra uygulama yeniden başlatılmalıdır.

Uygulamaya temel alınan algoritmanın akış diyagramı Şekil 5.2’de gösterilmiştir. Bu algoritmaya göre, uygulama başladıktan sonra NFC okuyucu tesbit edilir edilmez uygulama süresiz olarak dinleme moduna geçer. Okuyucuya mobil cihaz yaklaştırılınca, uygulamada, PC/SC (Personal Computer/ Smart Card Resource Manager) API’sinde tanımlı sınıflardan birisi olan ScardMonitor sınıfına ait CardInserted olayı, mobil cihaz uzaklaştırılınca da CardRemoved olayı gerçekleşir.



Şekil 5.2. NFC Okuyucu Uygulaması Akış Diyagramı

Yazılım terminolojisinde olay, gerçekleşmesi muhtemel bir aksiyonun gerçekleşmesi halinde yazılımın bunu karşılayıp işleme alma durumu olarak tanımlanabilir.

CardInserted olayı meydana gelince, yazılım, mobil cihaza Transmit metodu ile hemen bir ileti gönderir. Bu ileti, akıllı kartlar ile kart okuyucular arasındaki haberleşme birimi olan APDU (Application Protocol Data Unit) ile iletilir. APDU'nun yapısı, ISO/IEC 7816-4 standardı ile belirlenmiştir [28]. Bir APDU komutu, Çizelge 5.1'de gösterilen onaltılık sayı alanlarının birleşmesinden meydana gelir.

Alan Adı	Uzunluk (Byte)	Açıklama
Komut APDU		
CLA	1	Talimat sınıfı (Sanayilerarası, Tescilli vb.)
INS	1	Talimat ismi (Veri yaz, Veri oku vb.)
P1-P2	2	Kişisel talimat parametreleri
L_c	0,1 veya 3	Komut verisinin toplam bayt sayısı
Komut Verisi	N_c	Nc byte kadar komut verisi
Le	0,1,2 veya 3	Cevap olarak beklenen maksimum byte sayısı
Cevap APDU		
Cevap Verisi	N_r	Cevap olarak dönülen byte sayısı
SW1-SW2	2	Komut işleme durumu

Çizelge 5.1. Komut APDU ve Cevap APDU Yapısı

Uygulamada daha önceden bahsedilen CardInserted olayı meydana gelince, mobil cihazın özgün kimlik verisini cevap olarak dönmesi için, mobil cihaza Çizelge 5.2'deki APDU komutu gönderilir.

CLA	INS	P1	P2	Lc	Komut Verisi							Le
00h	A4h	04h	00h	07h	F0h	01h	02h	03h	04h	05h	06h	0Fh

Çizelge 5.2. Özgün Kimlik Numarası Almak İçin Gönderilen APDU Komutu

Bu komutu meydana getiren baytları ayrıntılı bir biçimde incelemek gerekirse;

- CLA alanındaki 00h, bu komutun güvenlik belirtisi taşımadığı anlamına gelir.
- INS alanındaki A4h, komuttaki talimatın Select File (Dosya Seç) talimatı olduğunu gösterir.

- P1-P2 alanı uygulamaya özel rastgele seçilmiş parametrelerdir. Mobil cihaz uygulaması sadece bu parametreleri taşıyan komutlara cevap olarak özgün kimlik numarasını göndermesi gerektiğini anlar.
- L_c alanı komut verisindeki 7 bayt veriyi işaret eder.
- Komut verisi alanında, gönderilen komutu mobil cihazda hangi uygulamanın işleme alacağını simgeleyen 7 baytlık uygulama kimlik numarası, AID, yer alır. Bu numara, online ödeme uygulamalarının kullandığı numaralar gibi önceden rezerve edilmiş belirli numaralar dışında rastgele seçilebilir.
- L_e alanında ise gönderilen komuta cevap olarak dönülecek olan 15 baytlık özgün kimlik numarasının boyutunu işaret eden 0Fh yer alır.

Komut mobil cihaza gönderildikten hemen sonra, mobil cihazdaki uygulama cihazın özgün kimlik numarasını cevap olarak döner. Daha önce de belirtildiği gibi bu tez çalışmasında, özgün kimlik numarası olarak, IMEI numarasına sahip cihazlarda cihazların IMEI numarası, diğer cihazlarda kablosuz adaptör MAC adresi veya işletim sistemi tarafından sağlanan Android ID kullanılmaktadır.

NFC okuyucu uygulaması, mobil cihazdan özgün kimlik numarasını başarıyla aldıktan sonra öncelikle bu numaranın sistem veritabanında olup olmadığını kontrol eder. Numara sistemde mevcut değilse giriş isteği başarısız olur. Numara sistemde mevcut ise, veritabanından bu kimlik numarasının hangi kullanıcıya ait olduğu bilgisi elde edilir. Son adımda ise, özgün kimlik numarası, kullanıcı adı ve sunucu saatinden alınan güncel zaman bilgisi veritabanındaki LoginRequest isimli tabloda kaydedilir.

Tüm işlemler başarıyla tamamlandıktan sonra, kullanıcıyı bilgilendirme amacıyla işlem sonucuna dair bilgi arayüzde gösterilir. Herhangi bir nedenden dolayı mobil cihaz kimliği okunamaz ise veya veritabanına yazma işlemi başarısız olursa, yine aynı şekilde uygulama arayüzünde kullanıcı bilgilendirilir.

5.3. Doğrulama Sunucusu SQL Server Veritabanı

SQL Server veritabanı, doğrulama sunucusu üzerinde bulunan bir diğer bileşendir. Öncelikli işlevi, kullanıcı ve cihaz profillerine ait tüm verileri saklamaktır. Sisteme giriş yapılırken daha önce Şekil 4.4. Sistem Giriş Algoritması Akış DiyagramıŞekil 4.4'te gösterildiği gibi doğrulama algoritmasında da önemli bir rol oynar.

Veritabanına tüm veri yazma ve veritabanından veri okuma operasyonları, web servis uygulaması ve NFC okuyucu uygulamasındaki metotlar tarafından gerçekleştirilir. Bu işlemler için başka bir uygulama geliştirilmemiştir.

Veritabanı uygulaması olarak Microsoft SQL Server 2014 versiyonu kullanılmıştır. Microsoft SQL Server, Microsoft şirketi tarafından geliştirilmiş bir ilişkisel veritabanı yönetim sistemidir. Bir veritabanı sunucusu olarak temel görevi diğer uygulamalardan gelen isteklere bağlı olarak veri yazma okuma operasyonlarını yerine getirmektir.

Microsoft SQL Server üzerinde, yeni bir veritabanı, veritabanını oluşturan tablolar ve tablolara ait sütunların hepsi ilgili SQL komutlarının bir yazılım tarafından çalıştırılması ile oluşturabilir. SQL bir programlama dili değildir. Bu tanımın yerine, herhangi bir veri tabanı ortamında kullanılan bir alt dil olarak tanımlanması daha doğru olur. SQL ile yalnızca veri tabanı üzerinde okuma, yazma, değiştirme gibi işlemler yapılabilir.

Ev otomasyonu sistemi için doğrulama sunucusunda oluşturulan veritabanı *HomeAutomationServerDB* isimli veritabanıdır. Bu veritabanı içerisinde 4 adet tablo oluşturulmuştur. Tablolar sırasıyla şunlardır;

Devices: Bu tablo, sistemde tanımlı olan izlenip kontrol edilen cihazlara ait profil bilgisi kayıtlarını bulundurur. Tabloda, cihazın özgün kimlik numarasını tutan *DeviceID*, cihazın ismini tutan *DeviceName*, cihazın yönetici tarafından kilitlendiği bilgisini tutan *LockedByAdmin* isimli kolonlar mevcuttur. Çizelge 5.3'te bu kolonların isimleri ve hangi veri tipinde oluşturulduğu listelenmiştir.

Kolon İsmi	Veri Tipi
DeviceID	int
DeviceName	nvarchar(50)
LockedByAdmin	bit

Çizelge 5.3. Devices Tablosunda Bulunan Kolonlar ve Kolonların Veri Tipi

LoginAttempts: Bu tablo, sisteme en son yapılan giriş denemesine ait kayıtları bulundurur. Tabloda, sisteme en son giriş yapmaya çalışan kullanıcının cihazına ait özgün kimlik bilgisini tutan *MobileDeviceID*, kullanıcı adı bilgisini tutan *Username* ve kullanıcının mobil cihazının NFC okuyucu tarafından okunma zamanını tutan *RequestDateTime* isimli kolonlar mevcuttur. Her bir giriş denemesinde, tablodaki

veriler güncellenir ve tabloda sadece bir satırlık veri bulunur. Çizelge 5.4'te bu kolonların isimleri ve hangi veri tipinde oluşturulduğu listelenmiştir.

Kolon İsmi	Veri Tipi
MobileDeviceID	nvarchar(50)
Username	nvarchar(50)
RequestDateTime	datetime

Çizelge 5.4. LoginAttempts Tablosunda Bulunan Kolonlar ve Kolonların Veri Tipi

LoginRecords: Bu tablo, sistemin ilk çalışmasından itibaren yapılan tüm sistem girişlerine ait kayıtları bulundurur. Tabloda, yapılan her bir girişe ait özgün kimlik numarası tutan *SessionID*, giriş yapan kullanıcının kullanıcı adı bilgisini tutan *Username*, kullanıcının sisteme giriş zamanını tutan *LoginDateTime* ve giriş yapıldıktan sonra mobil cihaz ile sunucu arasındaki iletişimin güvenli olmasını sağlayan şifreyi tutan *SessionToken* isimli kolonlar mevcuttur. Çizelge 5.5'te bu kolonların isimleri ve hangi veri tipinde oluşturulduğu listelenmiştir.

Kolon İsmi	Veri Tipi
SessionID	int
Username	nvarchar(50)
LoginDateTime	datetime
SessionToken	nvarchar(50)

Çizelge 5.5. LoginRecords Tablosunda Bulunan Kolonlar ve Kolonların Veri Tipi

Users: Bu tablo sistemde tanımlı tüm kullanıcılara ait profil bilgisi kayıtlarını bulundurur. Tabloda, kullanıcın kullanıcı adı bilgisini tutan *Username*, kullanıcının cihazına ait özgün kimlik bilgisini tutan *MobileDeviceID*, kullanıcı tipi (sistem yöneticisi/ normal kullanıcı) bilgisini tutan *IsAdministrator*, kullanıcının erişim yetkisi olduğu cihazların bilgisini tutan *ManagedDevicesList* ve kullanıcının parolasını tutan *PersonalPassword* isimli kolonlar mevcuttur. Çizelge 5.6'da bu kolonların isimleri ve hangi veri tipinde oluşturulduğu listelenmiştir.

Kolon İsmi	Veri Tipi
Username	nvarchar(50)
MobileDeviceID	nvarchar(50)
IsAdministrator	bit
ManagedDevicesList	nvarchar(50)
PersonalPassword	nvarchar(50)

Çizelge 5.6. Users Tablosunda Bulunan Kolonlar ve Kolonların Veri Tipi

Sistem yöneticisini Users tablosuna ekleyebilmek için herhangi bir uygulama arayüzü yoktur. Bu nedenle, sisteme SQL Server Management Studio üzerindeki SQL editöründe yazılan bir SQL sorgusuyla veya tablo verisi düzenleyici arayüzler aracılığıyla dahil edilebilir. Mobil cihaz kimliğinin öğrenilmesi için de NFC okuyucu uygulamasından faydalanabilinir.

5.4. Mobil Cihaz Android Uygulaması

5.4.1. Uygulama Altyapısı ve Temel Özellikler

Sistem yöneticisinin ve kullanıcıların, tez kapsamında geliştirilen sistemi etkili ve pratik bir biçimde kullanıp yönetebilmesi için sistem kumandası görevi görecek bir mobil cihaza ihtiyaçları vardır. Mevcut sistemde, bu amaç için yeni bir donanım tasarlamak yerine NFC özelliğine sahip olması nedeniyle piyasada kolayca bulunabilen akıllı telefonlar veya tabletler gibi mobil cihazların kullanılması tercih edilmiştir. Bu nedenle, kullanıcıların sistemi kumanda edebilmesi için faydalanacağı uygulama bu mobil cihazlar için geliştirilmiştir.

Piyasada NFC donanım desteği bulunan, Android, iOS, Windows Phone, Blackberry OS gibi birden fazla mobil işletim sistemi bulunmaktadır. Bunlar arasında iOS ve Android sahip olduğu pazar payı açısından en çok tercih edilen mobil işletim sistemleri olmuştur. iOS işletim sistemi ile çalışan bazı cihazlarda NFC donanımı bulunmasına rağmen, şu ana kadar geliştiriciler için bu donanımı kullanabilecek açık bir API sunulmamıştır. Bu nedenle, hem yaygın olarak kullanılması hem de geliştiricilere sunduğu imkanlar sayesinde, sistemde kullanılacak olan mobil uygulama Android platformu için geliştirilmiştir.

Android İşletim Sistemi, ilk kez 2008 yılında mobil cihazlar için geliştirilmiş Linux tabanlı, açık kaynak kodlu bir işletim sistemidir. Günümüzde resmi versiyonları Google tarafından geliştirilmeye devam edilmektedir. Android uygulamaları

geliştirmek için resmi IDE ise yine Google tarafından geliştirilmiş olan Android Studio uygulamasıdır. Bu ev otomasyon sistemindeki mobil uygulama da Android Studio üzerinde Java ve XML programlama dilleri ile geliştirilmiştir. Bir Android uygulamasında, tüm mantıksal işlemlerin yer aldığı backend olarak adlandırılan kısım Java dili ile geliştirilirken, kullanıcının gördüğü arayüz ekranları ise XML diliyle oluşturulur. Her bir arayüz sayfası Activity olarak isimlendirilen sınıftan türetilir.

Ev otomasyon sistemi mobil uygulamasında, her işlevsellik için ayrı bir sınıf dosyası kodlanmıştır ve arayüz tasarlanmıştır. Ayrıca, yeniden kullanılabilirlik seviyesini artırmak için, farklı bir çok sınıftan çağrılan metotları, global değişkenleri, sabitleri ve alanları içinde barındıran, herhangi bir arayüzle ilişkisi olmayan ortak sınıflar oluşturulmuştur. Uygulama çalışırken, tüm sınıflar tarafından erişilebilmesi gereken statik temel özellikler *SystemManager* isimli sınıfta bulunmaktadır. *SystemManager* sınıfında bulunan tüm özellikler, veri tipleri ve açıklamaları Çizelge 5.7'de gösterilmiştir.

SystemManager dışında, diğer önemli sınıflar ise *User* ve *Device* sınıfıdır. Bu sınıfların hangi senaryolarda kullanıldıkları ilerleyen sayfalarda yer alacaktır. Bu sınıflar rol itibariyle model olarak da değerlendirilebilir. *User* sınıfında kullanıcıya ait özellikler yer alır. *User* sınıfında bulunan tüm özellikler, veri tipleri ve açıklamaları Çizelge 5.8'de gösterilmiştir.

Device sınıfında ise izlenip kontrol edilen cihazlara ait özellikler yer alır. *Device* sınıfında bulunan tüm özellikler, veri tipleri ve açıklamaları Çizelge 5.9'da gösterilmiştir.

Özellik Adı	Veri Tipi	Açıklama
Username	String	Mobil cihaz ile sisteme giriş yapan kullanıcının adıdır.
MobileDeviceID	String	Uygulamanın çalıştığı mobil cihaza ait özgün kimlik numarası bilgisidir.
SessionToken	String	Yeni bir oturum başlatılınca, oturum boyunca doğrulama sunucusu ile güvenli iletişim için kullanılan kod parametresidir.
UserType	Enums.UserType	Kullanıcı tipi bilgisini barındıran Enum türünde bir veridir. UserType.ADMINISTRATOR (sistem yöneticisi) ve UserType.REGULAR (normal kullanıcı) seçeneklerine sahiptir.
User	User	User sınıfı tipinde, kullanıcıya ait özellikleri barındıran veridir.
SessionStartTime	Date	Oturumun tam olarak başlama saatini bildiren Date tipinde bir özelliktir.
SessionType	Enums.SessionType	Oturum tipi bilgisini barındıran Enum türünde bir veridir. SessionType.MANUAL (standart oturum) ve SessionType.P2P (eşler arası oturum) seçeneklerine sahiptir.
SessionDuration	int	Eşler arası tipinde başlayan bir oturumun sona erme süresini dakika cinsinden belirten sayısal değerdir.
P2PSessionStatus	boolean	Kullanıcının herhangi bir anda eşler arası bir oturuma dahil olup olmadığını belirten boolean tipinde bir veridir.

Çizelge 5.7. SystemManager Sınıfı Özellikleri

Özellik Adı	Veri Tipi	Açıklama
Username	String	Mobil cihaz ile sisteme giriş yapan kullanıcının adıdır.
MobileDeviceID	String	Uygulamanın çalıştığı mobil cihaza ait özgün kimlik numarası bilgisidir.
IsAdministrator	boolean	Oturum açmış olan kullanıcının sistem yöneticisi olup olmadığını belirten veridir.
ManagedDeviceList	List<Device>	Kullanıcının erişim ve kontrol etme yetkisine sahip olduğu cihazları barındıran Device sınıfı tipinde bir listedir.

Çizelge 5.8. User Sınıfı Özellikleri

Özellik Adı	Veri Tipi	Açıklama
DeviceID	int	Cihazın sistem veritabanında kayıtlı olan özgün kimlik numarasıdır.
DeviceName	String	Cihazın sistem veritabanında kayıtlı olan adıdır.
LockStatus	boolean	Cihaz kullanımının sistem yöneticisi tarafından engellenip engellenmediğini belirten veridir.
DeviceSettings	String	Cihazın sahip olduğu kontrol elemanlarını ve bu elemanların güncel durum bilgisini bir json nesnesi biçiminde barındıran metin verisidir. Tüm cihazların sahip olduğu kontrol elemanları farklı isimlere sahip olduğu için herhangi ortak bir sınıf tipinde temsil edilememektedir.

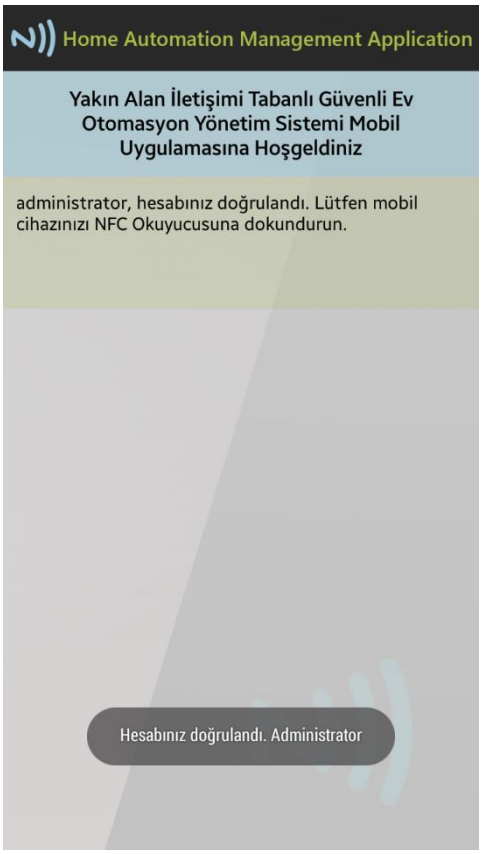
Çizelge 5.9. Device Sınıfı Özellikleri

Tüm kodlar yazılıp başarıyla derlendikten sonra, NFC özelliği bulunan ve Android işletim sistemi ile çalışan bütün mobil cihazlarda çalışabilmesi için kurulum paketi hazırlanır. Android Studio üzerinde bu işlem kolayca yapılabilir ve.apk uzantılı Android uygulama kurulum paketi elde edilir. Uygulama, mobil cihaza kurulduktan sonra, herhangi bir gereksinim olmadan ilk kullanım için hazır hale gelir. Yeni kullanıcı ve yeni mobil cihaz, sistem yöneticisi tarafından sisteme eklenilebilir.

5.4.2. Sisteme Giriş ve Oturum Başlatma

Daha önceki bölümlerde bahsedildiği gibi, uygulama, manuel olarak veya mobil cihazın sisteme dahil olan NFC etikete yaklaştırılması ile başlatılabilir. Uygulama ilk açılınca, kullanıcının hesabına giriş yapabileceği *Welcome* isimli giriş sayfası ekranda yer alır. Kullanıcı bu sayfada, kullanıcı adını ve kişisel parolasını girdikten sonra "Giriş" butonuna tıklar. Bu tıklama ile parametreler, tüm http iletişim operasyonlarının bulunduğu AsyncTask sınıfından türetilmiş *RequestTask* isimli bir sınıfa iletilir. Android API'sinin bir sınıfı olan AsyncTask, uygulamadaki işlem parçacıklarının asenkron bir şekilde, birbirini bekletmeden çalışabilmesini sağlayan bir altyapı sunar. Daha sonra ilgili metot içinde tanımlanmış, yine bir Android API sınıfı olan HttpClient'a ait metotlar ile http isteği asenkron bir şekilde doğrulama sunucusuna iletilir. Eğer http isteği, herhangi bir nedenle doğrulama sunucusuna iletilemezse, bu durum uygulamada kullanıcıya bildirilir.

Doğrulama sunucusu, veritabanından yaptığı sorgulama ile kullanıcı adı ve parolayı karşılaştırır. Veriler uyuşuyorsa http isteğine cevap olarak, oturum boyunca iletişim güvenliği için kullanılacak olan *SessionToken* ile birlikte *SUCCESS* mesajı, uyuşmuyorsa sadece *ERROR* mesajı dönülür. *ERROR* mesajı alınması durumunda arayüzde "Hesabınız doğrulanamadı!" mesajlı bildirim yapılır ve giriş ekranı tekrar aktif olur. *SUCCESS* mesajı alınması durumunda ise yine *WelcomeActivity* sayfasında iken giriş metin alanları ve butonu ekrandan kaybolur ve "Hesabınız doğrulandı. <*Kullanıcı adı*>" bildiriminin ekranda göründüğü Şekil 5.3'teki arayüz görüntülenir. Bu sırada, mobil cihazın NFC okuyucu tarafından okunabilmesi için Android API tarafından sunulan HostApduService sınıfından türetilmiş *MyHostApduService* sınıfı kullanılarak daha önce de bahsedilen kart emülasyon modu aktif hale getirilmiş olur. Bu servis çalışıyorken mobil cihazdaki NFC modülü bir akıllı kart gibi davranır.



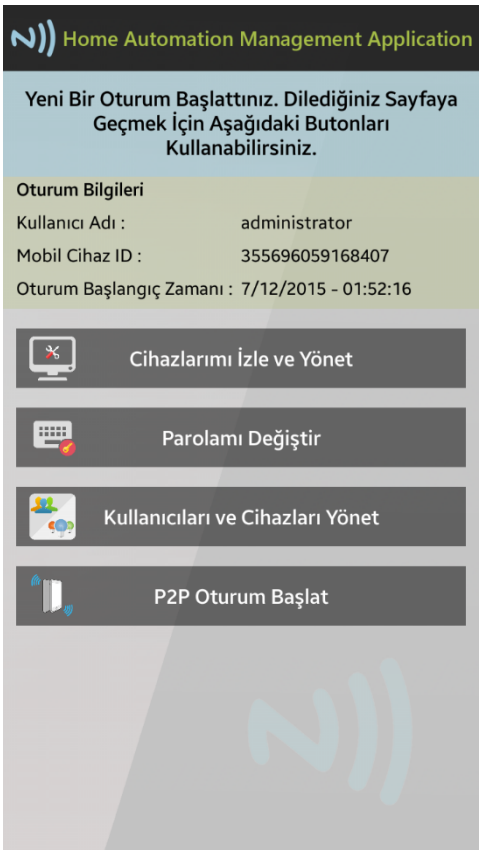
Şekil 5.3. Welcome Hesap Doğrulama Sonuç Ekranı

Bu aşamada uygulama, mobil cihazın NFC okuyucuya yakınlaştırılmasını bekler. Yeni oturum başlatılmadan uygulamadan çıkılırsa, SessionToken geçerliliğini kaybeder. Uygulama yeniden başlatılınca, kullanıcı hesabını tekrar doğrulatmak zorundadır.

Hesap doğrulama işleminden sonra mobil cihaz NFC okuyucuya yeterince yaklaştırılınca, okuyucu mobil cihaza APDU komutu gönderir. Bu komutla ilgili detaylı bilgi Bölüm 5.2'de verilmiştir. APDU mesajı *MyHostApduService* sınıfındaki metotlar tarafından karşılanır. Komutun içeriği işlenir ve komutun NFC okuyucudan geldiği doğrulanır. APDU komutuna cevap olarak uygulama, *SystemManager* sınıfında statik olarak tutulan cihaz özgün kimlik numarasını gönderir. Mobil cihaz, NFC okuyucudan uzaklaştırıldığında, yine *MyHostApduService* sınıfında diğer bir metot bu olayı işler ve doğrulama sunucusuna yeni bir http isteği göndererek yeni oturum başlatma talebini iletmiş olur. Bu istek parametre olarak kullanıcı adı, mobil cihaz özgün kimlik numarası ve *SessionToken* verilerini taşır. Doğrulama sunucusu, mobil cihazın, Bölüm 4.3.1'de yer verilen mobil cihaz doğrulama algoritmasına uygun biçimde NFC okuyucu aracılığıyla oturum başlatma talebinin bulunduğunu

doğrular ve *SUCCESS* mesajı döner. Eğer mobil cihaz 10 sn içerisinde kart okuyucudan uzaklaşmazsa yani, sunucuya oturum açma isteği 10 sn veya daha sonra gönderilirse sunucu *ERROR* cevabı döner. Ayrıca, kullanıcı kendi hesabıyla ilişkilendirilmemiş farklı bir cihaz kullanarak NFC okuyucu aracılığıyla giriş yapmak istediğinde de, kullanıcı adı ile mobil cihaz özgün kimlik numarası uyuşmayacağı için sunucu oturumu onaylamaz ve yine *ERROR* cevabı gönderir. Bunların dışında herhangi bir nedenle oturum açma işlemi başarısız olursa uygulama arayüzünde bir bildirim ile kullanıcı bilgilendirilir ve işlemi veya işlemleri tekrar etmesi istenir.

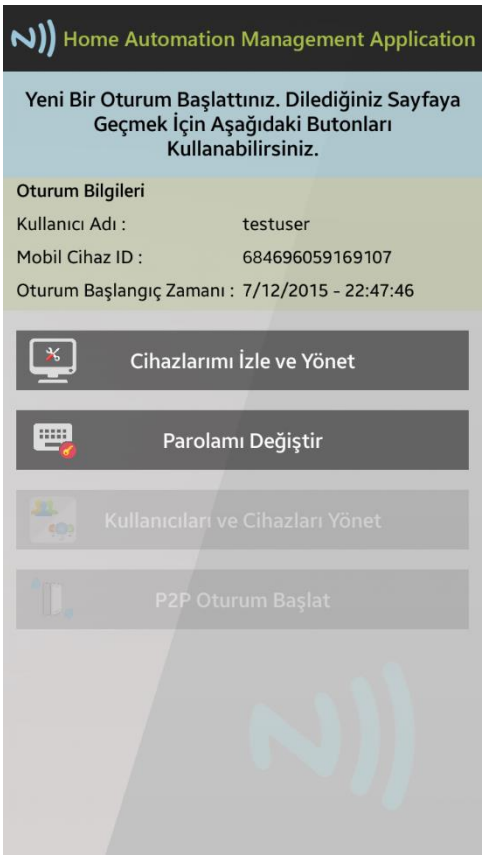
Yeni oturum başlatma talebi onaylandıktan sonra, uygulama tarafından doğrulama sunucusuna bir http isteği daha gönderilerek kullanıcının sistemde kayıtlı olan profil bilgileri string biçiminde alınır. Daha sonra, bu string verisi casting isimli yöntem ile *User* sınıfı tipinde bir nesneye dönüştürülür ve *SystemManager* sınıfındaki *SystemManager*.*User* alanına kaydedilir. Profil de başarıyla alındıktan sonra uygulama oturum moduna geçer ve Şekil 5.4'te görünen *Main* isimli ana sayfaya yönlenir.



Şekil 5.4. Main Anasayfa Sistem Yöneticisi Ekranı

Bu sayfanın ve bu sayfadan erişilen diğer tüm sayfaların üst bölümünde, *Kullanıcı Adı*, *Mobil Cihaz ID* ve *Oturum Başlangıç Zamanı* bilgilerini gösteren *Oturum Bilgileri* grubu yer almaktadır. Bu veriler, oturum boyunca değişmeyen statik bilgiler olduğu için, *SystemManager* sınıfından kolayca erişilebilirler.

Oturum Bilgileri grubunun altında ise "Cihazlarımı İzle ve Yönet", "Parolamı Değiştir", "Kullanıcı ve Cihazları Yönet" ve "P2P Oturum Başlat" sayfalarına bağlantı sağlayan butonlar mevcuttur. Eğer oturum açan kullanıcı sistem yöneticisi değilse Şekil 5.5'te görüldüğü gibi "Kullanıcı ve Cihazları Yönet" ve "P2P Oturum Başlat" butonları pasif ve saydam bir şekilde görünür. Daha önce de belirtildiği gibi bu işlevler sadece sistem yöneticisi tarafından erişilebilir.



Şekil 5.5. Main Anasayfa Normal Kullanıcı Ekranı

5.4.3. Cihaz İzleme ve Kontrolü

Anasayfada listelenen erişim butonlarından ilki "Cihazlarımı İzle ve Yönet" butonudur. Bu butona tıklanıldığında, kullanıcı Şekil 5.6'da görülen *MonitorAndControlDevices* isimli sayfaya yönlendirilir ve *Oturum Bilgileri* grubunun altında, kullanıcının izleme ve kontrol etme yetkisine sahip olduğu tüm cihazların isimleri listelenir.

Şekil 5.6. MonitorAndControlDevices Ekranı

Listelenen cihaz sayısının liste alanına sığmaması durumunda otomatik olarak dikey kaydırma çubukları ekranın kenarında yer alır.

Doğrulama sunucusu veritabanındaki *Devices* tablosundan yalnızca cihaz isimleri, özgün kimlik numaraları ve sistem yöneticisi tarafından kullanımının bloke edilip edilmediği bilgisi temin edilir. Cihaza ait kontrol elemanları listesi, eleman tipleri ve bu elemanların güncel durumları, tüm cihazların direkt olarak bağlı olduğu ağ geçidi üzerinden elde edilir. Bölüm 4.3.2'de bahsedildiği gibi, tasarlanan sistemde, cihazlarla iletişim kurmak için herhangi bir standarta bağlı kalınmadığı için cihaz izleme ve kontrolü simülasyon uygulaması ile yapılmaktadır.

Listelenmiş olan cihazların sahip olduğu kontrol elemanları listesi, eleman tipleri ve bu elemanların güncel durumları o anda kontrol edilmek istenen cihaz ismine ekranda dokunarak seçilmesi durumunda, Bölüm 5.5'te detayları açıklanmış olan simülasyon uygulamasından http isteği ile anlık temin edilir. Böylece, herhangi bir cihaz seçmeden, tüm cihazlara ait veriler alınmayarak, gereksiz veri trafiği ve uygulama yükü engellenmiş olur. Listeden bir cihaz seçildikten sonra uygulama,

cihazın izlenip kontrol edilebileceği Şekil 5.7'de görünen *DeviceSettings* isimli sayfaya yönlenir.

Şekil 5.7. DeviceSettings Ekranı

Ekran arayüzü oluşurken aynı zamanda, mobil uygulama kontrol edilen cihazdan, yani simülasyon uygulamasından cihaz verisini alır. Cihazların doğrulama sunucusu ile direkt olarak bağlantısı bulunmadığı için cihaz verileri doğrulama sunucusu veritabanına kaydedilmez. Tasarlanan sistemde, bu veriler simülasyon uygulamasının çalıştığı bilgisayarda <*CihazKimlikNumarası*>.json isimli bir dosyada JSON verisi biçiminde saklanır. Bu JSON nesnesinde cihazı ve kontrol elemanlarını tanımlayan yapı Şekil 5.8'de gösterilmiştir.

```
{
   "Name": "<CihazAdı>",
   "Controls": [
      "<Kontrol1>",
      "<Kontrol2>",
      "<Kontrol3>"
   ],
   "<Kontrol1>": {
      "UIControl": "SELECTOR",
      "ValueType": "MINMAX",
      "MinValue": “<MinimumDeğer>”,
      "MaxValue": “<MaksimumDeğer>”,
      "Values": "",
      "Default": "<VarsayılanDeğer>"
   },
   "<Kontrol2>": {
      "UIControl": "SELECTOR",
      "ValueType": "ARRAY",
      "MinValue": "",
      "MaxValue": "",
      "Values": [
         "<Değer1>”,
         "<Değer2>",
         "<Değer3>",
         "<Değer4>"
      ],
      "Default": "<Değer1>”
   },
   "< Kontrol3>”: {
      "UIControl": "SELECTOR",
      "ValueType": "ARRAY",
      "MinValue": "",
      "MaxValue": "",
      "Values": [
         "<Değer1>”,
         "<Değer2>",
         "<Değer3>",
         "<Değer4>",
         "<Değer5>”,
         "<Değer6>"
       ],
       "Default": "<Değer6>"
   }
}
```

Şekil 5.8. JSON Biçimli Cihaz Tanımlama Taslağı

JSON, bir nesneye ait olan özelliklerin ve bu özelliklerin değerlerinin Javascript notasyonunda gösterilmesidir [29]. Bir JSON bloğunda iki noktalı ifadelerin sol

tarafında parametre (özellik) ismi, sağ tarafında da bu parametrenin değeri bulunur. Eğer değer, birden fazla elemandan meydana gelen bir diziyse, değer dizisini oluşturan elemanlar köşeli parantez “[ ]” içinde listelenir.

Yukarıda tanımlanan cihaz bloğundaki parametreleri kısaca açıklarsak;

- *Name*, cihazın adıdır.
- *Controls*, cihazın yönetimini sağlayan kontrol elemanları dizisidir.
- *UIControl*, kontrol elemanları dizisindeki her bir kontrol elemanının arayüzde hangi görsel bileşen ile kontrol edileceğini ifade eder.
- *ValueType*, her bir kontrol elemanının alacağı değerlerin sayısal bir aralıktan veya metinsel/sayısal diziden oluştuğunu ifade eder.
- *MinValue*, kontrol elemanın alacağı değerlerin sayısal bir aralıkla ifade edilmesi durumunda sayısal aralığın minimum değerdir.
- *MaxValue*, kontrol elemanın alacağı değerlerin sayısal bir aralıkla ifade edilmesi durumunda sayısal aralığın maksimum değeridir.
- *Values*, kontrol elemanın alacağı değerlerin metinsel/sayısal bir diziyle ifade edilmesi durumunda diziyi oluşturan elemanlar listesidir.
- *Default*, bir cihaz ilk kez açıldığında veya bir kontrol elemanı herhangi bir kullanıcı tarafından değiştirilmeden önce kontrol elemanının sahip olduğu varsayılan değerdir.

Şekil 5.7’deki ekranda, izleyip kontrol etmek için *AirConditioner#2 – Bedroom* isimli cihaz seçilmiştir. Bu cihazın güncel durumunun gözlemlenmesi ve kontrol edilmesi için uygulamaya gelen JSON biçimindeki cihaz verisi Şekil 5.9’da gösterilmiştir.

```
{
    "Name": "AirConditioner#2 - Bedroom",
    "Controls": [
        "Temperature",
        "Fan Speed",
        "Direction",
        "Off Timer"
    ],
    "Temperature": {
        "UIControl": "SELECTOR",
        "ValueType": "MINMAX",
        "MinValue": "15",
        "MaxValue": "29",
        "Values": "",
        "Default": "21"
    },
    "Fan Speed": {
        "UIControl": "SELECTOR",
        "ValueType": "ARRAY",
        "MinValue": "",
        "MaxValue": "",
        "Values": [
            "LOW", "MEDIUM", "HIGH"
        ],
        "Default": "LOW"
    },
    "Direction": {
        "UIControl": "SELECTOR",
        "ValueType": "ARRAY",
        "MinValue": "",
        "MaxValue": "",
        "Values": [
            "DOWN", "LEFT", "RIGHT", "AUTO"
        ],
        "Default": "DOWN"
    },
    "Off Timer": {
        "UIControl": "SELECTOR",
        "ValueType": "ARRAY",
        "MinValue": "",
        "MaxValue": "",
        "Values": [
            "10", "20", "30", "40", "50", "60", "DISABLED"
        ],
        "Default": "DISABLED"
    }
}
```


Şekil 5.9. JSON Biçimli Cihaz Tanımlama Örneği

DeviceSettings sayfasına geçince uygulamaya ulaşan bu verinin anlamlı bir veriye dönüşmesi için Java programlama dilinde Map<string, object> olarak bilinen veri yapısı sınıfı kullanılmıştır. Map nesnesi, bir anahtar ve onun değerini tutan ikili veriden oluşur. Uygulama bu verileri Map nesnesine dönüştürürken öncelikle cihaz taslağına uygun olan *Name* ve *Controls* etiketlerini anahtar olarak kullanır. Daha sonra *Controls* etiketi ile oluşan Map nesnesindeki anahtar değeri, bir sonraki iterasyonda anahtar olarak okunur. Bu şekilde rekürsif bir yöntemle tüm veri Map nesnelerine dönüştürülür ve uygulama içinde kullanılmak istendiğine anahtar kelime parametresiyle anahtarın değeri elde edilir. Veriler, Map nesnelerine dönüştürüldükten sonra arayüzde karşılık gelen yerlere yerleştirilir.

Simülasyon uygulamasının çalıştığı bilgisayarda eğer cihazın anlık durum bilgisini barındıran <*CihazKimlikNumarası*_current>.json isimli dosya varsa, dosyadaki veriler aynı yöntemle alınır ve arayüze yansıtılır. Bu dosyada kontrol elemanlarının güncel durumu ve cihazın güç durumunu belirten JSON biçimli metin bulunur.

Eğer cihaz için böyle bir dosya bulunmuyorsa, kontrol elemanları için arayüzde varsayılan değerler gösterilir ve cihazın açık olduğu (PowerState:ON) varsayılır. Kontrol elemanlarının değerinde yapılan ilk değişiklik sonrası simülasyon uygulaması tarafından <*CihazKimlikNumarası*_current>.json isimli dosya oluşturulur. Bu işlem için, değeri değiştirilmek istenen kontrol elemanı satırında seçmeli listeden yeni değer seçilir ve “Yeni Ayarları Uygula” butonuna basılır. Yeni ayarlar başarıyla uygulandığı takdirde arayüzdeki kontrol elemanlarına ait değerler güncellenir ve kullanıcıya bildirim yapılır.

Cihazın gücünü kapatmak (PowerState:OFF) için de “Cihazı Kapat” butonuna basılır. Kontrol elemanlarının güncel durum bilgisi “Not Available” (N/A) olarak güncellenir. Aynı zamanda “Cihazı Kapat” butonu, tekrar cihazı açabilmek için “Cihazı Aç” butonuna dönüşür ve cihaz artık kontrol edilemediği için “Yeni Ayarları Uygula” butonu pasif hale gelir. “Cihazı Aç” butonuna basılınca cihaz tekrar kontrol edilebilir hale gelir ve kontrol elemanlarının değerleri cihaz kapatılmadan önceki değerlerine eşit hale gelir.

DeviceSettings sayfasındaki diğer bir butonsa “Cihazı Kilitle” butonudur. Oturum açan kullanıcı sistem yöneticisi değilse bu buton pasif ve saydam görünür, kullanıcı tarafından basılamaz. Sistem yöneticisi bu butona tıkladığında mobil cihaz ile

kontrol edilen cihaz arasında herhangi bir iletişim gerçekleşmez. Http isteği direkt olarak doğrulama sunucusuna iletilir ve sunucu veritabanında *Devices* tablosundaki *LockedByAdmin* alanı güncellenir. Böylece o andan itibaren sistem yöneticisi dışında hiçbir kullanıcı cihazın kontrol elemanlarına ait değerleri değiştiremez, sadece mevcut durumlarını gözlemleyebilir. “Cihazı Kapat” butonunda olduğu gibi, “Cihazı Kilitle” butonu da ekranda “Cihaz Kilidini Aç” butonu olarak güncellenir. Sistem yöneticisi “Cihaz Kilidini Aç” butonuna basınca yine http isteği direkt olarak doğrulama sunucusuna iletilir ve sunucu veritabanında *Devices* tablosundaki LockedByAdmin alanı güncellenir. Cihaz tekrardan tüm kullanıcılar tarafından kontrol edilebilir hale gelir.

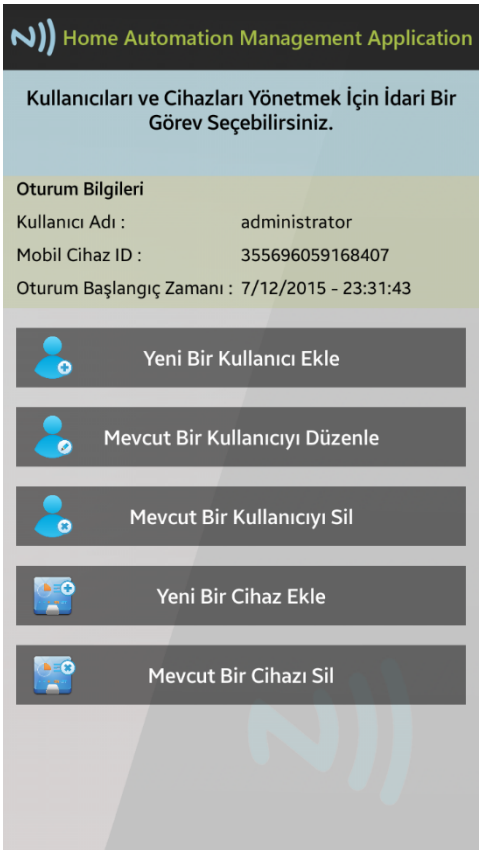
5.4.4. Profil Ayarları

Sistem yöneticisi profilinin sisteme nasıl eklendiği Bölüm 5.3’te son paragrafta anlatılmıştı. Sistem yöneticisi dışındaki tüm kullanıcılar ve mobil cihazları ise, doğrulama sunucusu veritabanına sadece sistem yöneticisi tarafından eklenir. Kullanıcılar sisteme eklenirken hesap parolaları sistem yöneticisi tarafından belirlenir. Kullanıcılar bu parolayı daha sonra Şekil 5.5’te görülen *Main* isimli sayfadaki “Parolamı Değiştir” butonuna basarak açılan sayfada değiştirebilirler. Buton basınca, uygulama *ChangeMyPassword* isimli sayfaya yönlenir. Kullanıcılar, bu ekranda mevcut parolasını bir kez, yeni parolasını iki kez ilgili metin alanlarına yazarak “Parola Değişikliğini Uygula” butonuna basarlar. Uygulama http isteğini doğrulama sunucusuna iletir. İstek başarılı bir şekilde sunucuya ulaşmışsa veritabanında ilgili alan güncellenir. Doğrulama Sunucusundan işlemin başarılı olup olmadığına dair geri cevap gelir ve gelen cevaba göre uygulama ekranında kullanıcıya bildirim yapılır. Bir kullanıcının kendi profiliyle ilgili yapabileceği tek değişiklik parola değiştirmektir. Kullanıcının izleyip kontrol edebildiği cihazlar listesindeki değişiklikler sadece sistem yöneticisi tarafından gerçekleştirilir. Bu işlemlerin detayları Bölüm 5.4.5’te ele alınacaktır.

5.4.5. Kullanıcı ve Cihaz Profillerinin Yönetimi

Kullanıcı profillerinin yönetimini sağlayan ekranların sadece sistem yöneticisi tarafından erişilebilir olduğundan daha önce bahsedilmişti. Bu ekranlara ulaşmak için sistem yöneticisi Şekil 5.4’da görülen *Main* isimli sayfadaki “Kullanıcıları ve Cihazları Yönet” butonuna basarak, kullanıcı ve cihaz profillerini yönetebileceği sayfa olan *ManageUsersAndDevices* isimli sayfaya yönlendirilir. Şekil 5.10’da

görülen bu ekranda, beş farklı işlev için beş adet buton yer almaktadır. Bu butonlar, “Yeni Bir Kullanıcı Ekle”, “Mevcut Bir Kullanıcıyı Düzenle”, “Mevcut Bir Kullanıcıyı Sil”, “Yeni Bir Cihaz Ekle”, “Mevcut Bir Cihazı Sil” şeklinde sıralanmıştır. Mevcut bir cihazın teknik özelliklerinin değişme ihtimalinin, göz ardı edilebilecek kadar küçük olması nedeniyle, cihaz profilini güncellemek için herhangi bir fonksiyon tanımlanmamıştır.

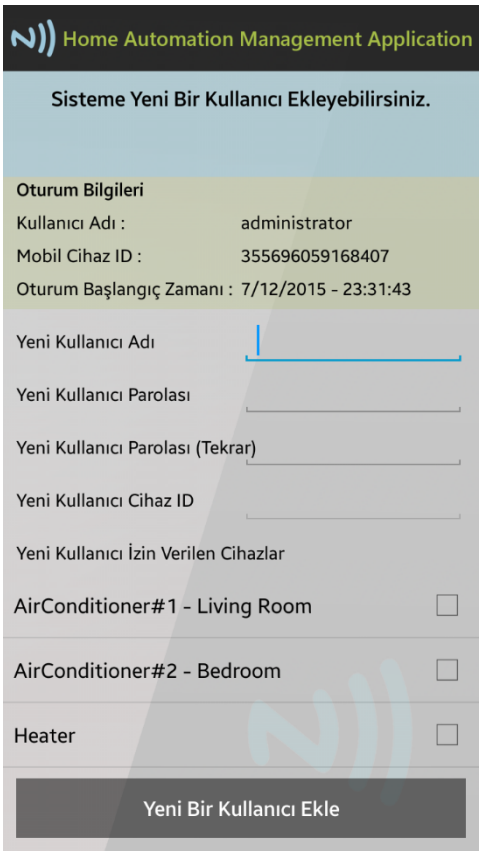


Şekil 5.10. ManageUsersAndDevices Kullanıcı ve Cihaz Profilleri Yönetme Ekranı

Sisteme yeni bir kullanıcı ve kullanıcıyla eşlenmiş bir mobil cihaz dahil etmek için, uygulamanın yeni mobil cihaza kurulmuş olması ve o anda çalışır olması gerekmektedir. Güvenlik nedeniyle, mobil cihaz özgün kimlik numarası elle yazılarak değil, yeni mobil cihazdan NFC aracılığıyla alınır. Sisteme eklenecek mobil cihazda çalışan Android uygulaması ve APDU servisi sayesinde, mobil cihazdaki NFC donanımı, daha önce bahsedildiği gibi akıllı kart emülasyonu (HCE) modunda çalışır.

Sistem yöneticisi “Yeni Bir Kullanıcı Ekle” butonuna basar. Eğer sistem yöneticisinin mobil cihazında Android Beam özelliği açıksa, onu kapatması için bildirim mesajı gösterilir ve işletim sisteminde bu özelliği kapatılmasını sağlayan ilgili ayarlar sayfası

açılır. Herhangi bir Android uygulaması bu özelliği ayarlama yetkisine sahip olamadığı için bu işlem mobil cihaz kullanıcsı tarafından elle yapılmalıdır. Android Beam açıkken işletim sisteminin NFC eşler arası modunda çalışmaya öncelik tanıması nedeniyle, HCE modunda sağlıklı bir iletişim gerçekleşmesi için kapatılması gerekmektedir. Android Beam özelliği kapalıysa veya kapatıldıktan sonra “Yeni Bir Kullanıcı Ekle” butonuna tekrar basılırsa, uygulama, Şekil 5.11’de görülen *AddANewUser* isimli sayfaya yönlenir.



Şekil 5.11. AddANewUser Yeni Kullanıcı Ekleme Ekranı

Yeni kullacının adı bir kez parolası iki kez ilgili metin alanlarına girilir. Kullanıcının izleyip kontrol edilmesine izin verilen cihazlar da, ”Yeni Kullanıcı İzin Verilen Cihazlar” başlığı altındaki listeden, cihaz isimlerinin yanındaki onay kutuları tıklanarak seçilir. Kullanıcı hesabıyla ilişkilendirilecek olan mobil cihazın özgün kimlik numarasının direkt olarak mobil cihazın kendisinden alınması için sistem yöneticisinin mobil cihazı ile yeni kullanıcının mobil cihazı birbirine yakınlaştırılır. Böylece, uygulama yeni kullanıcının mobil cihazından özgün kimlik numarasını isteyen Çizelge 5.10’daki APDU komutunu gönderir. Yeni kullanıcının mobil

cihazında çalışır durumda bekleyen Android uygulaması bu komutu işler ve mobil cihaza ait özgün cihaz kimlik numarasını cevap olarak döner.

CLA	INS	P1	P2	L_c	Komut Verisi							L_e
00h	A4h	04h	02h	07h	F0h	01h	02h	03h	04h	05h	06h	0Fh

Çizelge 5.10. Özgün Kimlik Numarası Almak İçin Gönderilen APDU Komutu

Kullanıcı hesabıyla ilişkilendirilecek olan mobil cihazın özgün kimlik numarası alındıktan sonra numaranın elle değiştirilememesi için numaranın bulunduğu metin alanı sayfada salt okunur (read-only) bir şekilde gösterilir. Yeni kullanıcının sisteme dahil edilebilmesi için tüm metin alanlarının doldurulması, mobil cihazın özgün kimlik numarasının mobil cihazdan NFC etkileşimi ile temin edilmesi ve yeni kullanıcının izleyip kontrol edebileceği en az bir adet cihaz seçilmiş olması gerekmektedir. Gerekli tüm alanlar doldurulmadan önce sayfanın en altındaki “Yeni Bir Kullanıcı Ekle” butonuna basılırsa tüm alanların doldurulması gerektiğine dair bildirim mesajı ekranda gösterilir. Gerekli tüm alanlar doldurulduktan sonra butona basılır ve bu alanlardaki verileri parametre olarak taşıyan http isteği, yeni kullanıcı profilinin sistem veritabanına kaydedilmesini sağlar. Sisteme başarıyla eklenen yeni bir kullanıcı kullanım senaryolarındaki yöntemlerle hemen sisteme hemen giriş yapabilir.

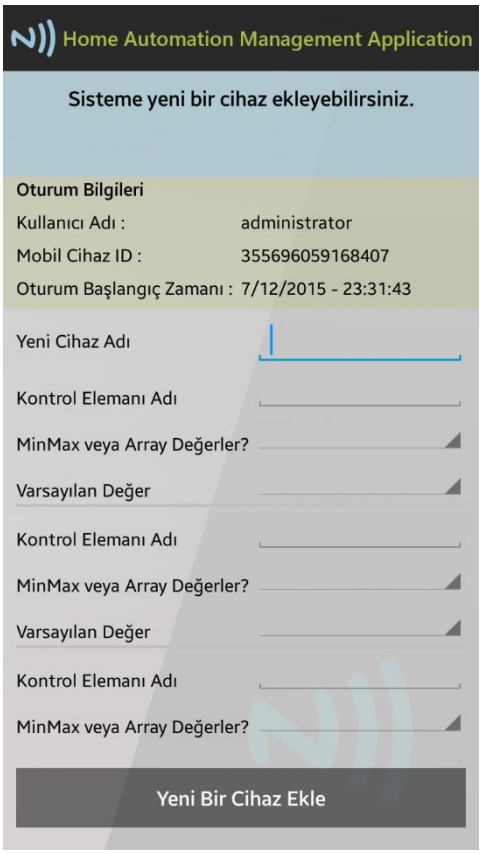
Mevcut bir kullanıcı profilini güncellemek için Şekil 5.10’da görülen ekranda, “Mevcut Bir Kullanıcıyı Güncelle” butonuna basılarak *EditAnExistingUser* isimli sayfaya geçiş yapılır.

Bu ekrana yönlendirilen uygulama, doğrulama sunucusundan sisteme kayıtlı olan tüm güncel kullanıcı profillerini http isteği ile temin eder. Sistem yöneticisi profilini değiştirmek istediği kullanıcıyı, kullanıcı adları listeleyen arayüz elemanı aracılığıyla seçer. Bir kullanıcı seçilince, “Seçilen Kullanıcı İzin Verilen Cihazlar” başlığı altındaki onaylı cihaz listesi o kullanıcının izleyip kontrol ettiği cihazları temsil edecek şekilde güncellenir. Böylece mevcut listenin korunması ve güncellenmesi işlemi daha kolay olur. Sistem yöneticisi, onay kutularında yapmak istediği değişiklikleri yapar ve “Mevcut Bir Kullanıcıyı Düzenle” butonuna basar. Böylece, kullanıcının sistem veritabanındaki profilinin güncellenmesini sağlayacak olan http isteği gönderilir. Sunucudan gelen cevaba göre İşlemin başarılı olup olmadığı ekranda bildirim mesajı olarak gösterilir.

Mevcut bir kullanıcı profilini silmek için Şekil 5.10'da görülen ekranda, "Mevcut Bir Kullanıcıyı Sil" butonuna basılarak *DeleteAnExistingUser* isimli sayfaya geçiş yapılır.

Bu ekrana yönlendirilen uygulama, doğrulama sunucusundan sisteme kayıtlı olan tüm güncel kullanıcı profillerini http isteği ile temin eder. Sistem yöneticisi profilini silmek istediği kullanıcıyı, kullanıcı adları listeleyen arayüz elemanı aracılığıyla seçer. "Mevcut Bir Kullanıcıyı Sil" butonuna basılınca "Kullanıcı Silme Operasyonunu Onaylayın" başlıklı onay penceresi ekranda yer alır. Sistem yöneticisi bu pencerede "Tamam" butonuna basarak silme işlemini onaylar. Böylece, kullanıcının sistem veritabanındaki profilinin silinmesini sağlayacak olan http isteği gönderilir. Sunucudan gelen cevaba göre İşlemin başarılı olup olmadığı ekranda bildirim mesajı olarak gösterilir.

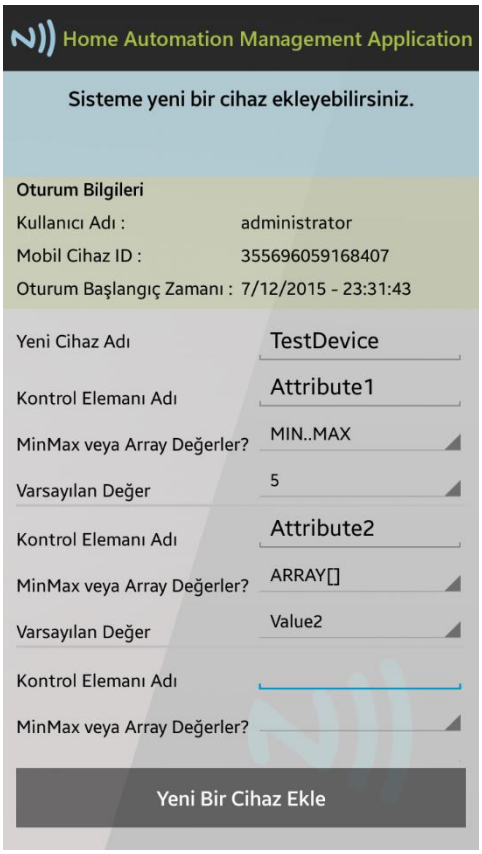
Sisteme yeni bir cihaz eklemek için Şekil 5.10'da görülen ekranda, "Yeni Bir Cihaz Ekle" butonuna basılarak Şekil 5.12'de görülen *AddANewDevice* isimli sayfaya geçiş yapılır.



Şekil 5.12. AddANewDevice Yeni Cihaz Ekleme Ekranı

Bu ekranda, Oturum Bilgileri grubunun altında, cihaz adı, her bir kontrol elemanı için, kontrol elemanının adı, değer tipi (minmax veya array) ve varsayılan değeri tanımlanır. Kontrol elemanının adı yazılıp klavyede "İleri" tuşuna basılınca *MinMax veya Array Değerler?* alanı aktif hale gelir ve bu seçeneklerden birisinin seçilmesi istenir. Değer tipi olarak *MinMax* seçilirse, MinMax değerler düzenleme penceresi açılır. Bu ekranda, kontrol elemanının alabileceği minimum ve maksimum sayısal değerler girilir. "Tamam" butonuna basılır, cihaz ekleme ekranına geri dönülür ve MinMax değerler aralığından bir sayı varsayılan değer olarak seçilir. Eğer MinMax tipinde değerler yerine Array, yani dizi tipinde sayısal veya sözel değerlerin girilmesi istenirse *MinMax veya Array Değerler?* alanında *Array[]* seçeneği seçilir ve Array değerler düzenleme penceresi açılır. Bu pencerede, değerler dizisine eklenmek istenen değer metin alanına yazıldıktan sonra, alanın sağındaki + ikonuna basılır. Değerler eklendikçe, pencerenin alt kısmında hepsi listelenir. Tüm değerler girildikten sonra "Tamam" butonuna basılır, cihaz ekleme ekranına geri dönülür ve Array değerler dizisinden bir eleman varsayılan değer olarak seçilir.

Şekil 5.13'te görüldüğü gibi gerekli alanlar doldurulduktan sonra "Yeni Bir Cihaz Ekle" butonuna basılır. Böylece, cihazın sistem veritabanına kaydedilmesini sağlayacak olan http isteği gönderilir. Sunucudan gelen cevaba göre İşlemin başarılı olup olmadığı ekranda bildirim mesajı olarak gösterilir. Veritabanına yeni cihaz ekleme işlemi başarılıysa, yeni cihaza ait bütün verileri parametre olarak taşıyan başka bir http isteği de simülasyon uygulamasına iletilir. Böylece, cihazın kontrol elemanlarına ve varsayılan değerlerine ait bilgiler içeren JSON biçimli dosya simülasyon uygulaması tarafından oluşturulur. Cihaz sisteme eklendikten sonra, yetkisi olan tüm kullanıcılar cihazı izleyip kontrol edebilirler.



Şekil 5.13. AddANewDevice Veri Girilmiş Yeni Cihaz Ekranı

Mevcut bir cihaz profilini silmek için Şekil 5.10'da görülen ekranda, "Mevcut Bir Cihazı Sil" butonuna basılarak *DeleteAnExistingDevice* isimli sayfaya geçiş yapılır.

Bu ekrana yönlenen uygulama, Doğrulama Sunucusundan sisteme kayıtlı olan tüm güncel cihaz profillerini http isteği ile temin eder. Sistem yöneticisi profilini silmek istediği cihazı, Şekil 5.14'te görülen cihaz adları listeleyen arayüz elemanı aracılığıyla seçer. "Mevcut Bir Kullanıcıyı Sil" butonuna basılınca "Cihaz Silme Operasyonunu Onaylayın" başlıklı onay penceresi ekranda yer alır. Sistem yöneticisi bu pencerede "Tamam" butonuna basarak silme işlemini onaylar. Böylece, cihazın sistem veritabanındaki profilinin silinmesini sağlayacak olan http isteği gönderilir. Sunucudan gelen cevaba göre İşlemin başarılı olup olmadığı ekranda bildirim mesajı olarak gösterilir. Seçilen cihazı veritabanından silme işlemi başarılıysa, simülasyon bilgisayarındaki cihaz ait JSON biçimli dosyalar da simülasyon uygulaması tarafından silinir.

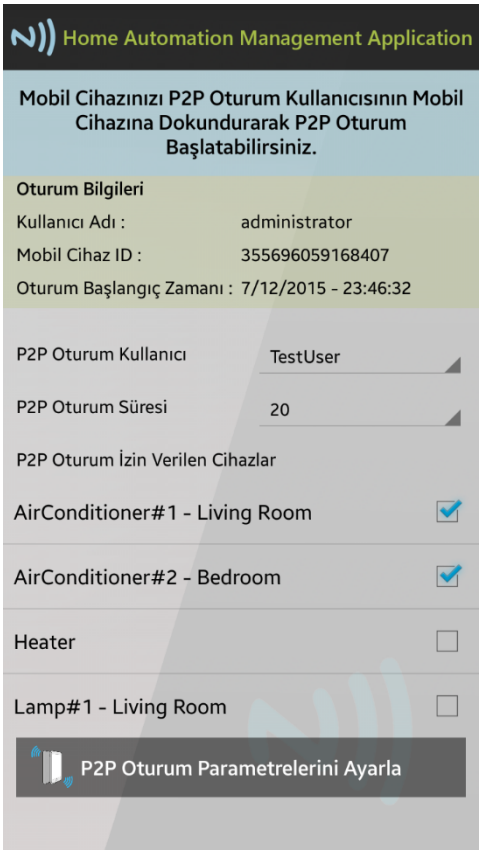
Şekil 5.14. DeleteAnExistingDevice Silinecek Cihaz Seçim Ekranı

5.4.6. Eşler Arası Oturum

Uygulamanın ana sayfasında listelenmiş dört ana işlevden sonuncusu eşler arası iletişim senaryosudur. Bu senaryonun gerçekleşebilmesi için mobil cihazların NFC üzerinden eşler arası modunda haberleşebilmesi amacıyla Android Beam özelliği açık olmalıdır. Android Beam, NFC özelliği bulunan mobil cihazlar arasında bir Bluetooth bağlantısı başlatarak hızlı ve güvenli bir şekilde dosya transfer edebilme özelliği sunar. Bu veri iletişiminde Bluetooth bağlantısı NFC donanımı tarafından konfigüre edilir. Şekil 5.4'te görülen ekranda, "P2P Oturum Başlat" butonuna basılınca eğer Android Beam özelliği kapalıysa kullanıcıya bildirim yapılır ve Android Beam ayar sayfası açılır. Herhangi bir Android uygulaması bu özelliği ayarlama yetkisine sahip olamadığı için bu işlem mobil cihaz kullanıcsı tarafından manuel olarak yapılmalıdır.

"P2P Oturum Başlat" butonuna basılınca Android Beam özelliği açıksa, uygulama Şekil 5.15'te görülen eşlerarası oturum parametrelerinin oluşturulacağı *InitiateAP2Psession* isimli sayfaya geçiş yapar.

Sistem yöneticisi, eşler arası oturum için gerekli parametrelerin oluşturulduğu bu ekranda, oturuma davet edilecek olan kullanıcıyı kullanıcılar listesinden seçer, oturumun otomatik olarak sona ereceği dakika cinsinden süreyi belirler ve bu oturum boyunca kullanıcının izleyip kontrol etme yetkisine sahip olacağı cihaz listesini oluşturur. Gerekli parametreler hazır olduktan sonra sistem yöneticisi ekranın en altındaki “P2P Oturum Parametrelerini Ayarla” butonuna basarak, parametrelerin uygulama tarafından bir NDEF mesajına dönüştürülmesini sağlar ve uygulamayı, bu mesajı iletmek için hazır hale getirir.

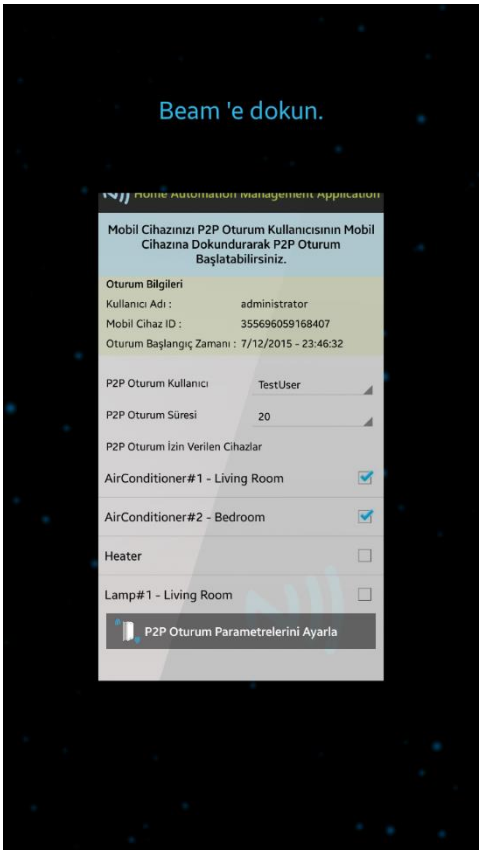


Şekil 5.15. InitiateAP2Psession Eşler Arası Oturum Başlatma Ekranı

Bu senaryoda öncelikle dikkat edilmesi gereken konular, sistem yöneticisinin oturum daveti iletmek için seçtiği kullanıcıyla ilişkilendirilmiş olan cihazın, NFC etkileşim gerçekleştirilecek olan mobil cihazla aynı olması ve kullanıcının mobil cihazında bu Android uygulamasının daha önceden kurulu olması gerektiğidir. Bu senaryo için temel alınan algoritmalar Bölüm 4.4’te detaylıca anlatılmıştır.

Oturum parametrelerinden oluşan NDEF mesajının, eşler arası oturuma dahil olacak kullanıcının mobil cihazına iletilmesi için iki mobil cihazın da NFC antenleri birbirine yönlendirimiş bir şekilde birbirlerine yakınlaştırılması gerekir. Mesafe

yeterince yakın olur olmaz NFC etkileşimi gerçekleşir. Android Beam devreye girer ve hem sistem yöneticisi ekranında, hem de diğer kullanıcının ekranında, Şekil 5.16'da görüldüğü gibi, aktif uygulama penceresi küçülerek, "Beam'e dokun" yazılı bir bildirim ile karşıdaki mobil cihaza NDEF mesajını iletme isteğinin onaylanması beklenir. Bu onay mekanizması, kötü amaçlı ve yanlış kullanımların önüne geçmek için alınmış bir güvenlik önlemidir. Sistem yöneticisi ekrana dokunarak veri iletimini onaylamış olur. Diğer kullanıcının gelen veriyi kabul etmesi için herhangi bir işlem yapmasına gerek yoktur.



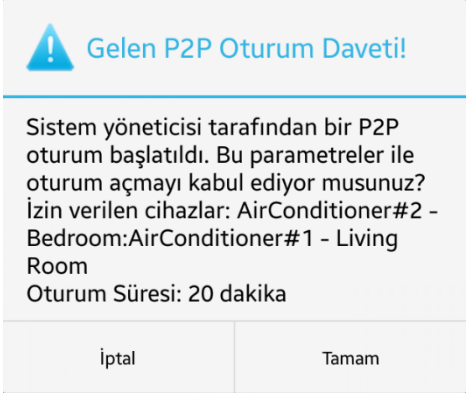
Şekil 5.16. Android Beam P2P Veri Transferi Onay Ekranı

Android Beam üzerinden aktarılacak olan NDEF mesajı içerisinde, oturum parametreleri veri paketi ve bu mesajı alıp işlemesi istenen uygulama paketinin ismi bulunur. Bu sayede, gelen veriyi alan kullanıcının mobil cihazında o anda uygulama çalışmıyor olsa bile veriyi işlemeden hemen önce, uygulama otomatik olarak başlatılır. Eğer NDEF mesajı içerisinde uygulama paketi ismi bulunmazsa, işletim sistemi kullanıcıya verinin hangi uygulama ile açılacağını sorar.

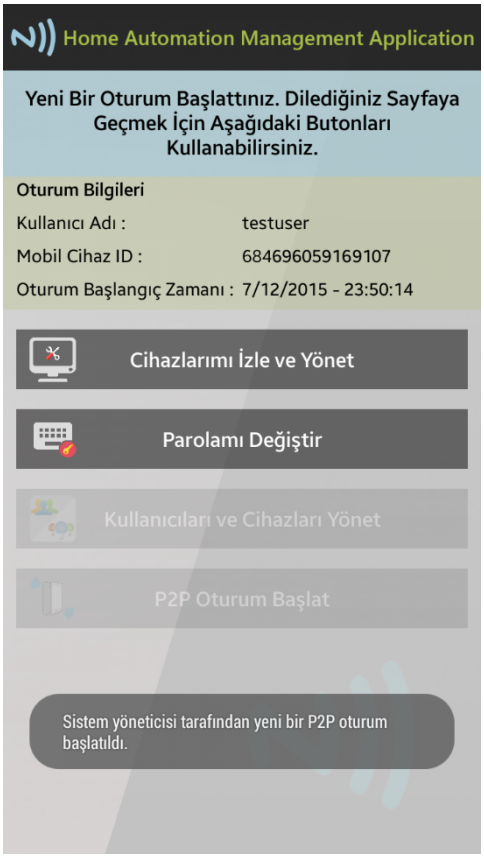
Alınan veri paketi işlenir ve eşler arası oturum parametreleri elde edilir. Bu parametrelerin detaylı bir biçimde belirtildiği Şekil 5.17'de görülen oturum daveti

onay penceresi kullanıcın ekranında yer alır ve oturum süresinin başlaması için kullanıcının onayı beklenir.

Kullanıcı, “İptal” butonuna basarak oturumun başlamasını iptal edebilir. Kullanıcı, “Tamam” butonuna basınca oturum süresini hesaplayan zaman sayacı çalışmaya başlar ve uygulama Şekil 5.18’de gösterildiği gibi bildirim mesajı göstererek normal bir senaryoda oturum başlatılmış gibi *Main* isimli ana sayfa ekranına yönlenir.



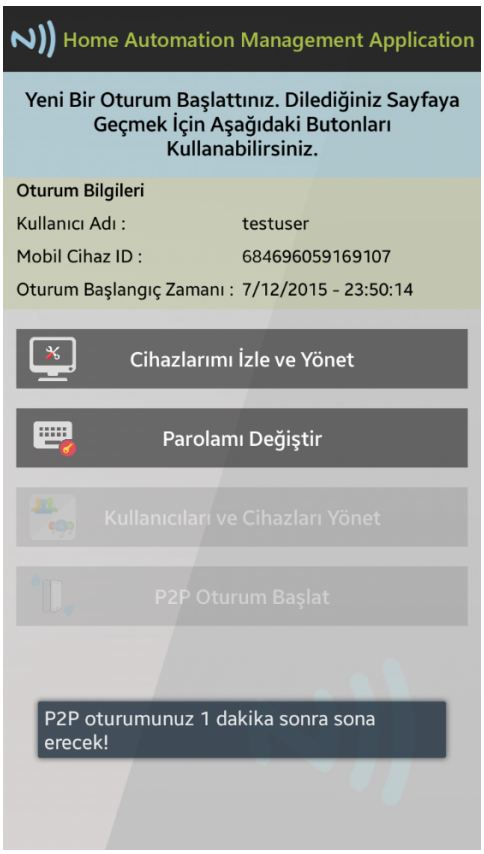
Şekil 5.17. Eşler Arası Oturum Daveti Onay Penceresi



Şekil 5.18. Main Anasayfa Eşler Arası Oturum Ekranı

Daha önceden de belirtildiği gibi, kullanıcı açısından eşler arası bir oturumun NFC okuyucu üzerinden giriş yaparak başlattığı bir oturumla işleyiş olarak herhangi bir farkı bulunmamaktadır. Sadece, izleyip kontrol edebileceği cihazlar değişebilir.

Oturumun süresini takip etmek için oturum başından itibaren çalışan bir zaman sayacı çalışır. Oturumun bitmesine 1 dakika kala Şekil 5.19'da görüldüğü gibi kullanıcıya bildirim yapılır. Süre tamamen dolunca ise oturum sona erer ve uygulama, *Welcome* isimli giriş sayfasına yönlenir.



Şekil 5.19. P2P Oturumu Sırasındaki Kalan Zaman Bildirim Ekranı

5.5. Simülasyon Uygulaması

Tasarlanan sistem, cihaz iletişim protokollerinden bağımsız olarak geliştirildiği için sistemin çalışma biçimini ve performansını gözlemleyebilme ihtiyacını karşılamak için sistemdeki cihazları simüle edecek bir uygulama geliştirilmiştir. Doğrulama sunucusuna ek bir işlemci ve bellek yükü getirmesi istenmediği için, uygulama, yerel ağ içinde bulunan Windows 10 işletim sistemiyle çalışan farklı bir bilgisayarda çalışmaktadır.

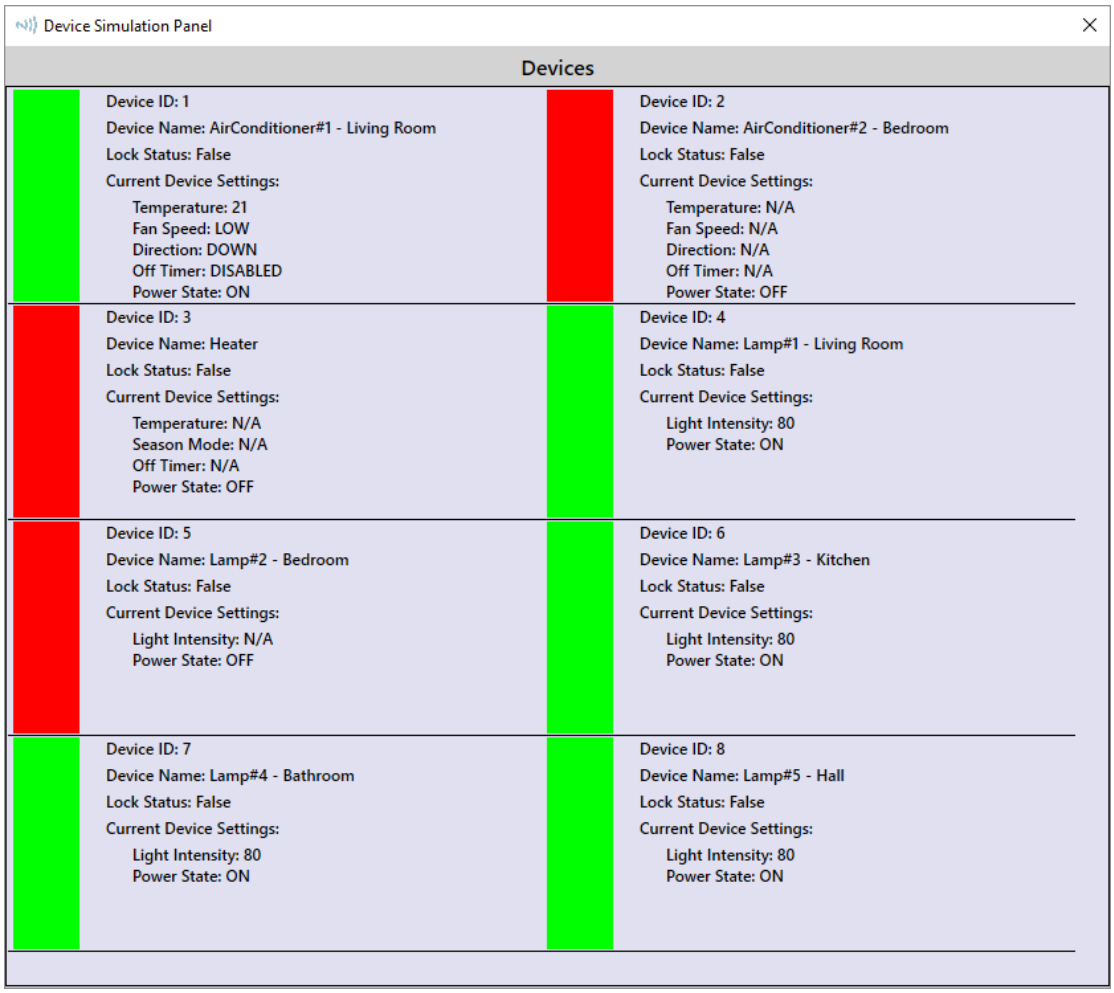
Device Simulation Panel isimli uygulama, Visual Studio 2013 geliştirme ortamında C# ve XAML dilleri ile geliştirilmiş bir WPF uygulamasıdır. WPF, Windows Forms

tipindeki uygulamaların bazı ihtiyaçları tam olarak karşılayamadığı durumlar için geliştirilmiş bir uygulama türüdür. Bütün mantık ve operasyon katmanı C# programlama dili ile geliştirilirken, kullanıcı arayüzü XAML isimli XML tabanlı bir programlama dili ile geliştirilir. Arayüz ve mantık katmanı arasında iletişim kurmayı kolaylaştıran bir yapısı vardır. Arayüz katmanındaki görsel elemanların özelliklerine mantık katmanındaki dinamik değişkenlerin değerleri bağlanır. Böylece mantık katmanı değerlerinde meydana gelen herhangi bir değişim arayüz katmanına aynı anda yansıtılır.

Uygulama, web servis uygulamasında olduğu gibi, mobil cihazlarla http istekleri ile haberleşir. Uygulama çalıştırılmasından kapatılmasına kadar, kendine özel ayrılmış bir porttan gelecek http isteklerini aralıksız dinler. Mobil cihaz uygulamasından gelen http istekleri işlenir ve hangi cihaz kontrol edilmek isteniyorsa arayüzdeki o cihaza ait güncel durum bilgisi ve bu bilgiyi taşıyan <*CihazKimlikNumarası*_current>.json dosyası güncellenir. Bir cihazın güncel durum bilgisi alınmak istenirse, yine mobil cihaz uygulamasından gönderilen başka bir http isteği işlenir ve cevap olarak cihaza ait güncel durum bilgisini taşıyan <*CihazKimlikNumarası*_current>.json dosyası metin biçiminde iletilir. Cihaz ilk kez açılmışsa, kontrol elemanları değer tipleri ve varsayılan değerler bilgisini taşıyan <*CihazKimlikNumarası*>.json dosyası isteğe cevap olarak metin biçiminde iletilir.

Uygulama çalışırken Şekil 5.20'de görüldüğü gibi, sistemde tanımlı tüm cihazların güncel durum bilgisini ana ekranda sunan bir arayüze sahiptir. Mobil cihaz uygulaması dışında bir veri girişi olmadığı için, kullanıcının veri girebileceği metin alanı, buton vb. herhangi bir interaktif arayüz elemanı yoktur. Arayüz, sadece bilgi sunma amacıyla tasarlanmıştır.

Arayüzde her bir cihaz için ayrılan alanda cihaz kimlik numarası (*Device ID*), cihaz adı (*Device Name*), sistem yöneticisi tarafından kilitlenme durumu (*Lock Status*) ve kontrol elemanlarına ait güncel değerler (*Current Device Settings*) bilgisi bulunur. Cihazların açık olup olmadığının kolayca anlaşılması için cihaz hücrelerinin sol tarafında renk çubuğu bulunur. Yeşil renk cihazın açık olduğunu, kırmızı renk ise cihazın kapalı olduğunu gösterir. Sistem yöneticisinin veya herhangi bir kullanıcının bir cihazda yaptığı değişiklik eş zamanlı olarak uygulama arayüzüne de yansıtılır. Böylece, cihazların anlık durumunu simüle edebilme ve mobil uygulama dışında bir ekrandan da gözlemleyebilme imkanı sunulur.



Şekil 5.20. Cihaz Simülasyon Uygulaması Arayüz Ekranı

6. ÖRNEK SENARYOLAR

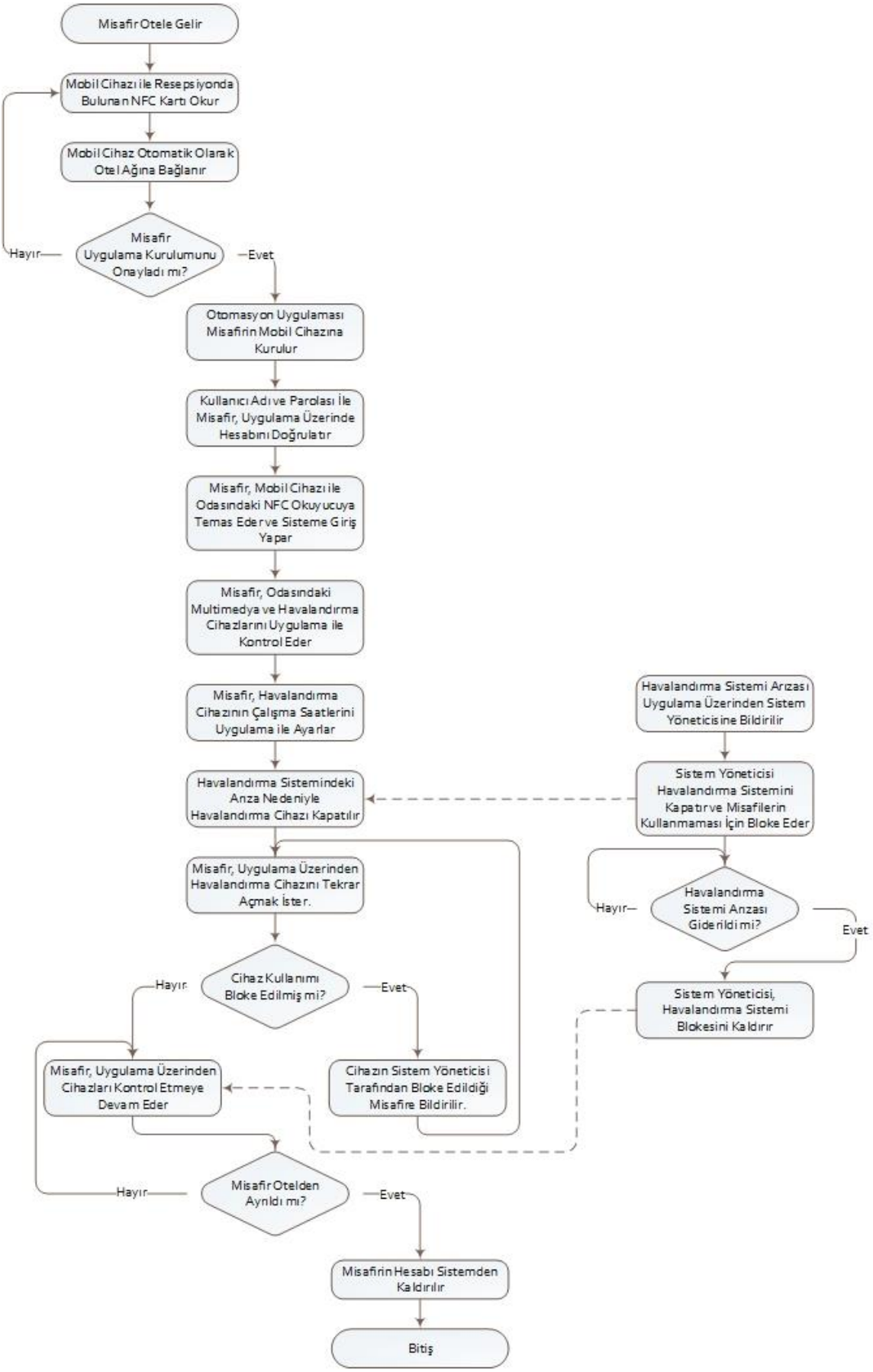
Tez kapsamında tasarlanan sistemde genel anlamda ev otomasyonları ele alınmıştır, fakat önerilen algoritmalar ve kullanım senaryoları otel, işyeri gibi ortak yaşam alanlarında kullanılan otomasyon sistemlerine de uyarlanabilir. Bu bölümde, sistemin, evlerin yanısıra otel ve ofislerde nasıl kullanılabileceği hakkında fikir vermesi amacıyla günlük hayatta karşımıza çıkabilecek örnek kullanım senaryoları işlenmiştir.

6.1. Örnek Senaryo 1 : Otelde Kullanım

Yeni açılan bir otel, konaklayan misafirlerin, odalardaki havalandırma, klima ve multimedya sistemini tek bir merkezi noktadan kontrol edebilmesini istemektedir. Bu kapsamda, tasarlanan sistem ve yazılımlar ihtiyaçları karşılamak için güncelleştirilmiş ve otel otomasyonu sistemine uyarlanmıştır. Senaryonun akış diyagramı Şekil 6.1'de gösterilmiştir.

Resepsiyonda bir NFC etiketi ve her odada bir NFC okuyucu bulunmaktadır. Otele giriş yapan misafirler, NFC özellikli mobil cihazları ile NFC etiketini okuyarak otelin kablosuz ağına otomatik olarak bağlanır ve gerekli otomasyon uygulaması kullanıcının onayı ile otomatik olarak misafirin mobil cihazına kurulur. Daha sonra sistem yöneticisi yetkisine sahip bir otel görevlisi kendisine ait mobil cihazı aracılığıyla Bölüm 5.4.5'te anlatıldığı gibi misafir kullanıcı için yeni bir hesap oluşturur ve bu hesaba yalnızca kalacağı odadaki cihazları kullanma yetkisi tanımlar. Böylece bir misafir kendi odası dışında başka odalardaki cihazlara erişemez. Herhangi bir kötü amaçlı kullanıma karşı önlem alınmış olur.

Misafir, odasına gelince, odadaki NFC okuyucuyu kullanarak mobil cihazı ile sisteme giriş yapar. Odasında mevcut olan cihazlar arasından multimedya cihazını seçer ve cihaza ait kontrol elemanlarının güncel durumunu görüntüler. Daha sonra kendi mobil cihazındaki bir şarkıyı yerel ağ üzerinden multimedya sistemine aktarır ve bu ses dosyasının multimedya sistemi üzerinde çalmasını sağlar.



Şekil 6.1. Otel Otomasyonu Senaryosu Akış Diyagramı

Misafir daha sonra uygulama üzerinde havalandırma cihazını seçer ve cihaza ait kontrol elemanlarının güncel durumunu görüntüler. Cihazın hava üfleme hızını isteği doğrultusunda değiştirir. Misafir yaptığı değişikliğin ne kadar süre geçerli olacağını da belirler. Havalandırma sisteminin akşam 20.00 - 23.00 ve sabah 07.00 - 08.00 saatleri aralığında açık, diğer aralıklarda kapalı olması için cihaz kontrol sayfasında gerekli ayarları yapar.

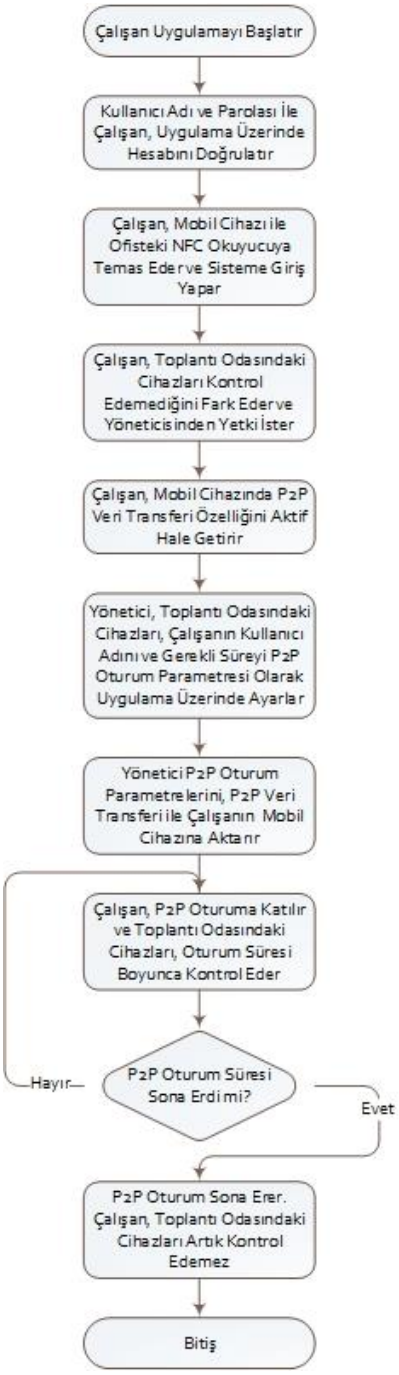
Akşam saat 21.30'da olağan dışı ani bir durum gelişir ve havalandırma sistemindeki ısı sensörü arızası nedeniyle, havalandırma sistemi, odalara üflemesi gerekenden daha soğuk bir hava üflemeye başlar. Oteldeki tüm cihazları kontrol etme yetkisi olan sistem yöneticisi hemen bütün odalardaki havalandırma cihazını uygulama üzerinden kapatır ve misafirlerin kullanımını engellemek için cihazları bloke eder. Bu sayede, misafir kullanıcı cihazı tekrar açıp kontrol etmeye çalışsa bile, cihazın sistem yöneticisi tarafından bloke edildiği uygulama üzerinden kendisine bildirilir. Arıza giderilince sistemler normale döner ve sistem yöneticisi cihazlar üzerindeki blokeyi kaldırır.

Misafir otelden ayrılırken, kendisi için oluşturulan kullanıcı hesabı sistemden silinir.

6.2. Örnek Senaryo 2 : Ofiste Kullanım

Bilgi güvenliği alanında faaliyet gösteren bir şirket, personelin, çalışma alanlarındaki havalandırma, klima ve multimedya sistemini tek bir merkezi noktadan kontrol edebilmesini istemektedir. Bu kapsamda, tasarlanan sistem ve yazılımlar ihtiyaçları karşılamak için güncelleştirilmiş ve ofis otomasyonu sistemine uyarlanmıştır. Senaryonun akış diyagramı Şekil 6.2'de gösterilmiştir.

İşyerinin her katında ve ortak kullanım alanlarında bir adet NFC okuyucu bulunmaktadır. Personel sisteme bu okuyucu ile giriş yaptıktan sonra cihaz kontrol sayfasında sadece kendisinin çalıştığı bölgedeki cihazların listesini görmektedir. Toplantı odaları, kafeterya, dinlenme salonu gibi ortak kullanım alanlarında bulunan cihazlar ise sadece sistem yöneticileri tarafından kontrol edilmektedir.



Şekil 6.2. Ofis Otomasyonu Senaryosu Akış Diyagramı

Bir toplantı odasında öğleden sonra 14.00 - 15.00 saatleri arasında toplantı organize etmek isteyen bir çalışan, odada bulunan klima ve projeksiyon cihazını mobil cihazındaki uygulama ile kontrol etmek istemektedir fakat NFC okuyucu ile giriş yapınca bu cihazlara erişim yetkisi olmadığını farkeder. Bu cihazları kontrol etme yetkisine sahip olmak için toplantıdan 10 dk. önce sistem yöneticisi yetkisine sahip olan birim yöneticisinden talepte bulunur. Sistem yöneticileri dışındaki personelin, şirket politikası gereği sistemde tanımlanmış profilleri, bu cihazları kullanmasına izin

vermez ve profillerinin değişmesi mümkün değildir. Bu nedenle, birim yöneticisi sahip olduğu yetkiyi kullanarak mobil uygulamadaki P2P oturum başlatma özelliğininden faydalanır. Kontrol edilecek cihazlar olarak toplantı odasındaki klima ve projeksiyon cihazını, kullanıcı olarak da toplantıyı organize edecek personeli seçer. Kullanım süresini de 70 dk. olarak ayarlar. Bu parametreler hazırlandıktan sonra P2P veri transferi ile personel yeni bir oturuma dahil olur. Böylece, personel saat 15.00'a kadar toplantı odasındaki klimayı ve projeksiyon cihazını izleyip kontrol etme yetkisine sahip olur. Saat 15.00'da toplantı bitince, 70 dakikalık süre dolduğu için oturum otomatik olarak sona erer.

7. SONUÇLAR

Mobil cihazlar, daha yaygın kullanılmaya başlandığından beri, birçok alanda olduğu gibi ev otomasyon teknolojilerinin de doğal bir parçası haline gelimiştir. Mobil cihazların otomasyon sistemindeki bileşenler ile iletişim şekli, sistemin güvenliği ve kullanışlı olabilmesi konusunda çok önemli bir parametredir. NFC, çok kısa mesafelerde çalışması sayesinde sunduğu güvenlik, düşük güç tüketimi, düşük üretim maliyeti ve kolay kullanımı ile Bluetooth ve Wi-Fi gibi teknolojilerle kıyaslandığında bir çok avantaj sağlamaktadır. İletişim hızı gibi geride kaldığı durumlarda ise Bluetooth ve Wi-Fi bağlantılarıyla birlikte çalışıp (ör. Android Beam) bu dezavantajını ortadan kaldırmaktadır. Bahsedilen bu avantajlar sayesinde, NFC, son zamanlarda, ev otomasyonu da dahil olmak üzere çeşitli amaçlar için NFC okuyucular, NFC özellikli mobil cihazlar ve temassız akıllı kartlar arasında iletişim kurmak için popüler bir yöntem haline gelmiştir. Bu popülarite, güvenlik ve kullanıcı yönetimini kolaylaştırma gibi, NFC kullanımı ile işlenebilen yeni konuların oluşmasına neden olmuştur. Bu tez çalışmasında, altyapısını geliştirmek isteyen ev otomasyonu kullanıcılarının ve geliştiricilerinin bu kavramları daha iyi anlayabilmesi için bu konular detaylıca anlatılmıştır.

Kullanıcılar bir sistem ile etkileşime girince, donanım kimliklerinin her zaman bir yazılım ile kandırılma riski mevcuttur. NFC teknolojisi doğal özelliği sayesinde, bir mobil cihaza ait özgün kimlik numarasının NFC aracılığıyla iletilmesine imkan tanıyarak güvenli bir donanım doğrulama yeteneği sağlar. Bu çalışmada, yazılımda tanımlanmış bir donanım kimliği, reddedilemez bir şekilde doğrulanmış ve böylece ev otomasyon sistemi için son derece güvenli bir doğrulama mekanizması sağlanmıştır.

Ayrıca, NFC'nin farklı modlarda çalışabilmesi ile sağladığı tüm kolaylıklar ve imkanlar değerlendirilmiş, bu imkanlar çerçevesinde çeşitli senaryolar oluşturulmuştur. Bir NFC okuyucu ile doğrulama mekanizmasının ele alındığı bir genel kullanım senaryosunun yanısıra, kullanıcılar açısından sisteme daha kolay bir giriş yöntemi sunan ve daha önce literatürde rastlanılmamış olan eşler arası anlık kullanım senaryosu da çalışmaya dahil edilmiştir. Bu yöntem ile, kullanıcıların doğrulama sunucusundaki profilini değiştirme ihtiyacı duymadan, geçici bir süre boyunca sistemdeki cihazları kontrol edebilme imkanı kullanıcılara sunulmuştur. Tez kapsamında geliştirilen algoritma ve yöntemlerin, otel ve ofis gibi ortak yaşam

alanlarına nasıl uyarlanabileceğine dair örnek senaryolar ile ev dışı kullanım imkanları da değerlendirilmiş ve kullanım alanının ev ile sınırlı olmadığı vurgulanmıştır.

Çalışmada bahsi geçen mobil cihaz uygulaması ve diğer yazılım bileşenleri, önerilen tasarımın nasıl gerçekleştirilebileceğini ve sistemlere nasıl entegre edilebileceğini ispatlamak amacıyla popüler mobil ve bilgisayar platformları için geliştirilmiştir. Sistemin kullanılabilirliğini artırabilmek amacıyla, NFC uyumlu uygulama geliştime arayüzü sunan diğer mobil platformlar için de tezde önerilen algoritmaları temel alan mobil uygulamalar geliştirilebilir. Ayrıca, tezdeki örnek kullanım senaryolarında önerilen ofis ve otel gibi ortak yaşam alanlarının dışında, önerilen algoritma ve yöntemler ihtiyaç doğrultusunda güncellenerek sistemin, okul, restoran, kafe, alışveriş merkezi gibi yerlerde kullanımı da mümkün hale getirilebilir.

KAYNAKLAR

[1] Kozan Demircan, Akıllı Evler, *Popular Science Türkiye*, Aralık **2014**.

[2] M. Nikolova, F. Meijs and P. Voorwinden, “Remote mobile control of home appliances,” *IEEE Transactions on Consumer Electronics*, vol. 49, no. 1, pp. 123-127, Feb. **2003**.

[3] A. Alheraish, “Design and implementation of home automation system,” *IEEE Transactions on Consumer Electronics*, vol. 50, no. 4, pp. 1087-1092, Nov. **2004**.

[4] B. Yuksekkaya, A. A. Kayalar, M. B. Tosun, M. K. Ozcan and A. Z. Alkar, “A GSM, internet and speech controlled wireless interactive home automation system,” *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, pp. 837-843, Aug. **2006**.

[5] K. Gill, Shuang-Hua Yang, Fang Yao and Xin Lu, “A zigbee-based home automation system,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 422-430, May **2009**.

[6] P. Bergstrom, K. Driscoll and J. Kimball, “Making home automation communications secure,” *IEEE Computer Magazine*, Vol. 34, pp. 50-56, Oct. **2001**.

[7] A. Z. Alkar and U. Buhur, “An internet based wireless home automation system for multifunctional devices,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1169-1174, Nov. **2005**.

[8] A. Mondal, K. Roy and P. Bhattacharya, “Secure and simplified access to home appliances using Iris recognition,” *IEEE Workshop on Computational Intelligence in Biometrics: Theory, Algorithms, and Applications*, pp. 22-29, Apr. **2009**.

[9] G. De Luca, P. Lillo, L. Mainetti, V. Mighali, L. Patrono, and I. Sergi, “KNX-based home automation systems for Android mobile devices,” *SMART 2013: The Second International Conference on Smart Systems, Devices and Technologies*, pp. 20-23, June **2013**.

[10] Akıllı Ev Sistemleri, Pusula Bilişim, http://www.pusulabilisim.com/p/akilli-ev-sistemleri.aspx (Erişim Tarihi; Şubat, **2016**).

[11] NFC Forum Technical Specifications, NFC Forum, http://members.nfc-forum.org/specs/spec_list/ (Erişim Tarihi; Şubat, **2016**).

[12] What are the operating modes of NFC devices?, NFC Forum, http://www.nfc-forum.org/resources/what-are-the-operating-modes-of-nfc-devices/ (Erişim Tarihi; Şubat, **2016**).

[13] Host-based Card Emulation, Android API Guides, https://developer.android.com/guide/topics/connectivity/nfc/hce.html (Erişim Tarihi; Şubat, **2016**).

[14] ISO/IEC 18092:2013, *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*, Mar. **2013**.

[15] V. Coskun, K. Ok, B. Ozdenizci, *Professional NFC Application Development for Android*, Wiley Publishing, Inc., pp. 181-183, **2013**.

[16] V. Coskun, K. Ok, B. Ozdenizci, *Professional NFC Application Development for Android*, Wiley Publishing, Inc., pp. 38-39, **2013**.

[17] History of Near Field Communication, http://www.nearfieldcommunication.org/history-nfc.html (Erişim Tarihi; Şubat, **2016**).

[18] L. Chen, G. Pan and S. Li, "Touch-driven Interaction via an NFC-enabled smartphone," *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 504-506, Mar. **2012**.

[19] A. Basili, W. Liguori, F. Palumbo, "NFC Smart Tourist Card: Combining Mobile and Contactless Technologies towards a Smart Tourist Experience," *IEEE 23rd International WETICE Conference*, pp. 249-254, June **2014**.

[20] G. De Luca, P. Lillo, L. Mainetti, V. Mighali, L. Patrono and I. Sergi, "The use of NFC and Android technologies to enable a KNX-based smart home," *21nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1-7, Sept **2013**.

[21] N. Chandrakar, S. Kaul, M. Mohan, C. Sai Vamsi and K. R. Prabhu, "NFC based profiling of smart home lighting system," *International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 338-341, May **2015**.

[22] N. Lekic, Z. Mijanovic, “NFC identification system for fuel dispensing control on petrol station,” *IEEE EUROCON*, pp. 638-644, July **2013**.

[23] V. Patil , N. Varma, S. Vinchurkar, B. Patil, “NFC based health monitoring and controlling system,” *IEEE Global Conference on Wireless Computing and Networking (GCWCN)*, pp. 133-137, Dec **2014**.

[24] S. Cavalieri, O. Mirabella, “Small EHS: proposal for a profile of the European Home System protocol,” *Proceedings of IEEE International Conference on Industrial Technology*, pp. 486-490, Jan **2000**.

[25] ACR122U USB NFC Reader, http://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/ (Erişim Tarihi; Şubat, **2016**).

[26] S. Lee, T. Lee, K. Kim and M. Hong, “A fast and efficient wireless AP connection approach based on NFC Tag,” *International Conference on ICT Convergence (ICTC)*, pp.1076-1077, Oct. **2013**.

[27] Jon Galloway, Phil Haack, Brad Wilson and K. Scott Allen, *Professional ASP.NET MVC 4*, Wiley Publishing, Inc. **2012**.

[28] ISO/IEC 7816-4:2013, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*, Apr. **2013**.

[29] Introducing JSON, http://www.json.org/ (Erişim Tarihi; Şubat, **2016**).

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Tolga Hakan ODUNCU

Doğum Yeri : Ordu

Medeni Hali : Evli

E-posta : thakanoduncu@gmail.com

Adresi : Yenimahalle / ANKARA

Eğitim

Lise : Beyşehir Ali AKKANAT Anadolu Lisesi, KONYA

Lisans : Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü, ANKARA

Yüksek Lisans : Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü, ANKARA

Yabancı Dil ve Düzeyi

İngilizce : İleri

İş Deneyimi

2015 – , TÜRKTRUST Bilgi İletişim ve Bilişim Güvenliği Hizmetleri A.Ş.

2013 – 2015, SEBİT Eğitim ve Bilgi Teknolojileri A.Ş.

2008 – 2011, HAVELSAN Hava Elektronik Sanayi ve Ticaret A.Ş.

2007 – 2008, TÜRK TELEKOMÜNİKASYON A.Ş.

Deneyim Alanları

Masaüstü ve Mobil Yazılım Geliştirme, Gömülü Yazılım Geliştirme, Nesne Yönelimli Programlama, .NET Framework.

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliş ve/veya Poster Sunumu ile Katıldığı Topantılar

-