

RESTful Web Servisleri ve Node.js Kullanılarak Genel Bir Kullanıcı Doğrulama Sisteminin Raspberry Pi ve RFID Teknolojisi ile Tasarımı ve Gerçekleştirimi

Mehmet KARAKOÇ¹ , Canberk ARDIÇ² , Mehmet Arif Emre ŞEN³ 

¹Alanya Hep Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Antalya, 07400, Türkiye

²Mercedes-Benz Otomotiv, İstanbul, 34538, Türkiye

³Open Business Software Solutions, İstanbul, 34906, Türkiye

Araştırma Makalesi, Geliş Tarihi: 08.12.2020, Kabul Tarihi: 08.06.2021

Özet

Bu çalışmada, genel bir kullanıcı doğrulama sisteminin tasarımı ve gerçekleştirimi ele alınmaktadır. Bu amaçla, donanımsal gerçekleştirme için Raspberry Pi ile RFID teknolojisi ve yazılımsal gerçekleştirme için RESTful web servisleri ile Python ve Node.js kullanılmıştır. Geliştirilen sistemde, (i) Raspberry Pi RFID okuyucu ile kullanıcının kartını okur ve girdi veriyi geliştirilen Python veya Node.js uygulamasına iletir. (ii) bu uygulama bir önceki çalışmada geliştirilmiş olan servis yazılımına ilgili HTTP isteğini gönderir. (iii) bu program kullanıcı doğrulaması için gerekli veri tabanı işlemlerini yapar ve JSON yanıtını döndürür. (iv) çıktı Node.js uygulaması aracılığıyla ekranda görüntülenir. Yapılan deneylerde, Node.js uygulamasının etkin kullanıcı etkileşimiyle Python'daki GUI sorununu aşabilmede oldukça uygun olduğu gözlemlenmiştir. Ayrıca, Raspberry Pi kullanımı kararlı bir sistem geliştirmede iyi bir çözüm olmuştur.

Anahtar Kelimeler: Raspberry Pi, RFID, RESTful web servisleri, Python, Node.js.

Design and Implementation of a Generic User Verification System with Raspberry Pi and RFID Technology Using RESTful Web Services and Node.js

Abstract

In this study, it is addressed to design and implement a generic *user verification system*. To this end, Raspberry Pi with RFID technology for hardware implementation and RESTful web services with Python and Node.js for software implementation were used. In the developed system, (i) Raspberry Pi with RFID reader reads the user's card and transmits the input data to the developed Python or Node.js application. (ii) this application sends the relevant HTTP request to the *service software* that was developed in the previous study. (iii) this program performs the database operations required for the *user verification* and returns the JSON response. (iv) the output is visualized on a screen via the Node.js application. In the experiments conducted, it was observed that the Node.js application is fairly appropriate in overcoming the GUI issue in Python with efficient *user interaction*. Additionally, using Raspberry Pi has become a good solution in developing a stable system.

Keywords: Raspberry Pi, RFID, RESTful web services, Python, Node.js.

¹Sorumlu yazar mehmet.karakoc@alanyahep.edu.tr, ²canberk.ardic@daimler.com, ³maemresen@yazilim.vip

1. GİRİŞ

İncereis ve Akgün'e göre, IoT'de (*Internet of Things* – nesnelerin interneti), *servis yazılımı* geliştirilmeli ve servisler bir oyuncak-bloğu (*lego*) gibi sisteme istenildiği zaman eklenebilmeli, çıkartılabilmeli veya güncellenebilmelidir. Ayrıca, özellikle bir nesne ile aynı anda veri alışverişi yapmak isteyen çok sayıda cihaz varsa, esnek, etkili ve ölçeklenebilir çözümler üretilebilir. Örneğin, bir önceki çalışmada (Ardıç vd, 2019) geliştirilmiş olan *servis yazılımı*, geliştirilmiş olan Android uygulaması ile konuşurken; bu çalışmada ise geliştirilen Python ve Node.js uygulamaları ile konuşmaktadır. Bu kapsamda, makineler arasında ortak bilgi paylaşımı (*çevrim içi bilgi dağıtımı*) sağlanarak, bir önceki çalışmada Arduino ile yapılmış olan gerçeklemler "Raspberry Pi" (R-PI) kullanılarak da yapılmıştır. Bu tür sistemler ve açık-kaynak donanım altyapıları/platformları, bilgi doğrulama, büro güvenliği, ev otomasyonu ve kullanıcı onaylama gibi pek çok amaç için sıklıkla kullanılmaktadır.

1.1. Literatür Taraması

Alan yazını incelendiğinde, bilgi servis, ev otomasyon, katılım güvenlik, katılım yönetim, öğretim yönetim ve sıcaklık ölçümü gibi amaçlara yönelik olarak R-PI kullanıldığı görülmektedir.

Priya ve Harish (2015), duyargalar/sensörler kullanarak sıcaklık, basınç, hız ve yön gibi değişkenleri (*parametre*) ölçme yeteneğine sahip, düşük-maliyetli ve etkin bir sualtı aracı tasarlayıp gerçekleştirmişlerdir. Değişkenleri görüntüleme (doğrudan bir LCD (*liquid crystal display* – sıvı kristal görüntü birimi) ekran ile yapılamayacağından) R-PI ile ara-bağlantısı-olan (*interfaced*) bir ekran yardımıyla, iletişim ise I2C/PC (*inter-integrated circuit* – ara-tümleşik devre) ve SPI (*serial peripheral interface* – seri çevre-birimi arayüzü) protokolleri aracılığıyla gerçekleştirilmiştir. Natarajan vd. (2017)'e göre laboratuvarların sıcaklık ölçüm ve izleme sistemine ihtiyacı vardır. DS18B20 ve "R-PI 2" (Rpi 2) Model B kullanarak düşük-maliyetli bir sıcaklık günlüğü/kayıt sistemi tasarlamışlar, kodlama için Python kullanmışlardır. Python betiği (*script*) çalıştırılarak çıktı veri belirli bir dosyaya kaydedilmiş ve sıcaklıklar çeşitli aralıklar için ölçülmüştür. Bu ölçüm sistemi sıcak bir sistemdeki çevresel etkiyi çalışmak için de kullanılabilir.

Dayıoğlu vd. (2016), sera içindeki sıcaklık değişimlerinin ölçülmesi, web ortamına kablosuz aktarılması ve izlenmesi amacıyla IoT teknolojisi (*uygulayım bilimi*) ile ön/ilk-örnek (*prototype*) bir sistem

geliştirmişlerdir. Sıcaklık ölçümü için İnternete bağlanabilen iki kablosuz sensör ağı/düğümü (*wireless sensor network*) tasarlamışlardır. Her düğüm R-PI 2 tek-kart bilgisayar (*single-board computer*), iki adet sıcaklık sensörü ve "Wi-Fi" (*wireless fidelity* – kablosuz İnternet bağlantısı) adaptöründen oluşmaktadır. Sistem, sera içinde ölçülen hava, su ve toprak sıcaklığı verisi web ortamında izlenerek ve veri tabanına kaydedilerek sınıanmıştır. Ferdoush ve Li (2014) ise Arduino ve R-PI ile ZigBee modülü, XBee S2B ve bir dizi açık-kaynak yazılım paketi kullanarak bir *kablosuz sensör ağı* sistemi geliştirmişlerdir. *Kablosuz sensör ağının* geçit düğümü (*gateway node*), veri tabanı sunucusu ve web sunucusunu tek-kart bilgisayar donanım altyapısında birleştirmişlerdir. R-PI ile monitör, klavye ve fare olmadan çalışmaya kolaylıkla yapılandırma söz konusudur. Onlara göre böyle bir tasarım, pek çok çevresel izleme ve veri toplama uygulamasında kullanışlıdır. Ayrıca, kullanıcıların sensör verisine erişebilecekleri veya uzaktan konuşlandırılmış (*deployed*) sensör düğümlerinin yapılandırma ve yönetimini gerçekleştirebilecekleri bir web uygulaması (kullanışlı bir web arayüzü) geliştirmişlerdir.

Maybel ve Umamakeswari (2018), "R-PI 3" (Rpi 3) kullanarak bir sistemden bir web sunucusuna güvenli görüntü aktarımı sağlayacak sistemli/dizgeli bir yöntem önermişlerdir. Python komutları ile görüntüleri işlemek için OpenCV'nin Python API'si (*application programming interface* – uygulama programlama arayüzü) olan OpenCV-Python kullanmışlardır. Şifrelenmiş görüntü doğrudan aktarılmakta ve başka herhangi bir sistemden uzaktan erişilebilmektedir. Kiran ve Mekala (2018), endüstride çalışanlar/işçiler ve ders boyunca öğrenciler için katılım alacak bir sistem geliştirmişlerdir. Yüz tespit etme ve tanıma yönelik olarak, Open CV kütüphanesi yüklü R-PI ve bağlı R-PI kamera modülü kullanarak yoklamayı yüz tanıma ile almışlardır. İnternet üzerinden erişilebilen veri R-PI'a bağlı hafıza kartında saklanır. Katılım alma için bir zaman-dilimi ayarlanır ve veri tabanı otomatik olarak İnternet bağlantısı üzerinden web sunucusuna yüklenir. Ardıç vd. (2019) ise genel bir kullanıcı doğrulama sisteminin tasarımı ve gerçekleştirmesini ele almışlar, donanımsal gerçekleştirme için Arduino mikrodenetleyici kart ile RFID (*Radio Frequency IDentification* – radyo/telsiz frekansı ile tanımlama) teknolojisi ve yazılımsal gerçekleştirme için RESTful web servisleri ile Android kullanmışlardır.

Tan vd. (2018), Node MCU ve RFID-RC522 gibi açık-kaynak donanım altyapılarını kullanarak, otomatik öğrenci katılım kaydı yapma ve öğrencilerin öğrenme

istekliliğini/motivasyonunu güçlendirmede öğretmene yardımcı olacak IoT-tabanlı bir *öğretim yönetim sistemi* tasarlayıp geliştirmişlerdir. Dersin çevrim içi videolarının bağlantılarına hızlı erişim ve derslikte gerçek-zamanlı etkileşimli yanıt alabilmek (öğrencilerin yer alabilecekleri uygulama alıştırmalarını göstermek) için QR (*quick response* – gecikmesiz/hızlı yanıt) kod kullanmışlardır. Çok-boyutlu öğrenme söz konusudur. Sistem için kullanılacak yeni bir RFID okuyucu geliştirmek amacıyla, “kablosuz İnternet bağlantısı” ağıyla bütünleşik yüksek frekans RFID teknolojisi kullanmışlardır. Kablosuz yöntemle çalışan RFID sistemleri, mevcut derslikte “konuşlandırma / canlıya alma” (*deployment*) için uygundur. Onlara göre, *bilgi teknolojilerinin* gelişmekte olan bir alanı olarak IoT teknolojisi, eğitim uygulamalarında kullanılan *bilgi teknolojilerinin* kaçınılmaz eğilimidir ve yükseköğrenim sınıflarının yönetimine uygulanır.

Zhong ve Liang (2016)’a göre, R-PI IoT’nin hem araştırma hem de uygulamalarında uygun bir ortam olarak giderek daha fazla benimsenmektedir. Ayrıca, öğrencilerin öğrenme başarımlarını/performans ve deneyimini büyük ölçüde iyileştirecek etkili bir araçtır. *Bilgisayar Bilimleri* lisans öğrencilerine yönelik IoT dersinde tasarlanmış yeni bir proje-tabanlı öğretim ve öğrenme yaklaşımı sunmuşlardır. Tasarlanan ders basit donanım öğrenmeden bütünüyle bir ilk örnek sistem inşa etmeye kadar devam etmektedir. Öğrenciler tarafından benzer bir IoT dersinin sonunda tasarlanıp geliştirilen ilk örnek IoT sistemlerini dikkate almışlardır. Proje çalışması, dönem-boyu etkinlik olarak bir dizi proje şeklinde ve “böl ve yönet / parçala ve fethet” (*divide and conquer*) metodolojisi (*yöntem bilimi*) uygulanarak tasarlanmıştır.

Awakhare vd. (2018), etiket (*tag/label*) taşıyan nesneyi radyo dalgaları üzerinden belirleyecek RFID-tabanlı bir *katılım güvenlik sistemi* önermişlerdir. Yoklamalar kontrol edilerek, öğrenci okula girdiğinde ve okuldan ayrıldığında ebeveynleri bilgilendirilir. Etiket taşıyan öğrenci RFID okuyucunun aralığına/menziline girdiğinde veya kartını okuttuğunda, okuyucu bu etiketin verisini okur. Bu veri tabanındaki ile eşleşirse, sistemdeki LCD kimlik doğrulama iletisini görüntüler ve ebeveynler SMS ile bilgilendirilirler. Diğer durumda ise LCD yetkisiz giriş görüntüler. Öğrencilerin yoklama verisini saklamak ve daha sonra kontrol edebilmek için veri tabanı kullanmışlar, tüm sistemi Arduino UNO ile denetlemişlerdir. Ayrıca, ebeveynler sistemin web sitesine girebilir ve çocuklarının durumunu kontrol edebilirler. Ranjana ve Vinoth (2016) ise okula gidiş ve okuldan geliş süresince okul çocuklarının genel güvenliğini artırmak amacıyla, günlük otobüsle okula

gidip gelmelerini takip etmek için RFID-tabanlı bir sistem sunmuşlardır. Bu sistemle, bir çocuk otobüse bindiğinde veya otobüsten ayrıldığında otomatik olarak tespit etmeyi ve bu gerçekleşmediyse ebeveynlere uyarı iletisi göndermeyi amaçlamışlardır. Gelmeyenler tespit edildiğinde ise bu durum ilgili otobüs numarası ve sürücünün telefon numarası ile ebeveynlere SMS olarak iletilir. RFID-tabanlı algılama birimi (*RFID okuyucu*) otobüsün içerisinde olup, çocuklar tarafından giyilen RFID-etiketleri tespit eder. İlgili veriyi bir GSM modem aracılığıyla sistemin veri tabanı sunucusuna gönderir ve yoklamalar kontrol edilerek veri tabanı güncellenir. Ayrıca, ebeveynler sistemin web sitesine girebilir ve çocukları ile ilgili ayrıntıları görebilirler.

Rjeib vd. (2018), akademik alan için *programlanabilir mantık devresi* olan Arduino ve RFID-tabanlı bir *katılım yönetim sistemi* ve *bilgi (servis) sistemi* tasarlayıp geliştirmişlerdir. Bu sistemle, öğrencilerin yoklama verisini yönetmeyi amaçlamışlardır. Sistem, öğrenci katılımını takip etme olanağı sunar. Öğrencilerin ders zamanları, derslik numaraları, günlük zaman çizelgesi ve not çizelgeleri ve fakülte bölüm personeli tarafından sağlanan öğrenciye ilişkin diğer yönergeler ile ilgili bilgileri içeren *bilgi servislerini* de destekler. Kayıtları izleme ve koruma, öğrencileri takip etme, rapor gönderme, veri yönetimi ve *bilgi servislerini* sağlama gibi işlevler içerir. Ayrıca, ek kâğıt işlerine gerek olmadan veriyi saklamada personele kolaylık sağlar.

Tiwari ve Singh (2017), trafik sinyaline göre harekete geçen otonom bir araba (iki tekerleği olan ve kendinden yanıt veren *şasili* (motorlu kara taşıtının iskelet bölümü) robot araba) için R-PI 3 ve R-PI kamera modülü kullanarak dur işaretleri panosu ve kırmızı trafik sinyalini tespit edecek bir teknik geliştirmişlerdir. Tüm sistem için kodlamada Python, görüntü işleme için Open CV kullanmışlardır. Öndeki araçla çarpışmayı önlemede, araç hız kontrolü için uzaklık ölçümüne yönelik ses ötesi (*ultrasonic*) sensörü kullanmışlardır. Kameradan görüntü alındıktan sonra trafikteki kırmızı sinyaller için maskeleyme ve dış hat (*contour*) tekniklerini, trafik panosundaki işaretleri belirlemek için HAAR CASCADE tekniğini kullanmışlardır. Herhangi bir araç aracın menziline girdiğinde, ses ötesi sensörü bir sonraki aracın uzaklığını düzenli aralıkta vermekte, bu mesafe düzenli zamanda alınarak öndeki aracı geçmek için aracın maksimum hızı hesaplanabilmektedir. Küçükkülahlı ve Güler (2015) ise R-PI kullanarak İnternet üzerinden bir web sayfası aracılığıyla gerçek-zamanlı *gezgin/mobil robot kontrolü ve izleme* için bir sistem geliştirmişlerdir. Ek kontrol takımı olmadan gerçekleştirdikleri düşük-maliyetli ve açık-kaynak

gezgin robot tasarımında, robot üzerindeki USB kamera ile çevre-birimi izleme gerçekleştirmişlerdir. R-PI'nin GPIO (*general purpose input output* – genel amaçlı giriş-çıkış) bacaklarını/uçlarını (*pins*) kontrol etmek için “node.jsrpi-gpio” kütüphanesini, daha güvenli ve hızlı bir bağlantı için HTTP yerine WebSocket ve R-PI üzerinden hem bu protokolü hem de DC ve servo motorları yönetmek için Node.js kullanmışlardır.

Kaur vd. (2016), anahtar kelime eşleme mantığında çalışan ve zaman, hava ve bildirim bilgileri içeren ek modüllerle bir *ses komut sistemi* gerçekleştirmişlerdir. Sistemi başlatan kullanıcıdan mikrofon ile ses girdisi alınır ve bilgisayara işlenmek üzere gönderilir. Ses girdisi bir sesten-metne-dönüştürücü kullanılarak bilgisayar tarafından tanınabilen ve işlenebilecek metne dönüştürülür. Üretilen bu metin ayrıştırılır (*parsing*) ve eşlenecek anahtar kelimeler için aranır. Eşleme olursa ilgili metin çıktısı verilir. İlgili bilgi getirildiğinde bir OCR (*optical character recognition* – optik karakter tanıma) sistemi içeren bir metinden sese dönüştürücü kullanılarak ses çıktısına dönüştürülür ve ilgili çıktı kullanıcıya verilir. OCR metni sınıflayıp tanır ve metinden sese dönüştürücü bunu sese dönüştürür. Bu çıktı ise R-PI'nin ses kablu ucuna (*jack*) bağlı hoparlörler ile aktarılır. Sudhakar vd. (2017) ise Android Mobile, Bluetooth ve R-PI kullanarak sesten metne dönüştürme ve görüntülemeyi çalışmışlardır. Onlara göre, bu uygulama, derslik ve konferanslardaki sunumlarda oldukça kullanışlı olup; görme, duyma veya fiziksel engelli kullanıcılara yönelik veri girişi seçenekleri sağlayarak sistemin erişilebilirliğini de artırmaktadır. R-PI'deki uygulama programının kodu Python ile yazılmıştır. Android Mobile üzerinde sesten metne dönüştürme için AMR (*Android Meets Robots*) Voice uygulama yazılımı kullanılmıştır. Ses metin dosyasına dönüştürülür ve Bluetooth HC-05 modülü üzerinden R-PI'ya gönderilip bir monitör üzerinde görüntülenir.

Rani vd. (2018), R-PI 3 ile farklı donanım ve yazılım araçları kullanarak yetkisiz/istenmeyen izinsiz giriş veya hareketi tespit edecek bir *ev güvenlik sistemi* geliştirmişlerdir. Yetkisiz kişiler bölgeye girerlerse, PIR sensörü kişi eylemini tespit eder ve güvenlik sistemi yetkisiz kişinin görüntüsünü alır (PI kamera). R-PI kartındaki dâhilî “kablosuz İnternet bağlantısı” modülüyle, yakalanan görüntü yetkili kişiye e-posta içerisinde gönderilir ve İnternet üzerinden bağlı WAY2SMS kullanılarak ona gerçek-zamanlı uyarı SMS'i iletilir. Uyarı sonrasında ise *Buzzer* etkinleşir ve yüksek hoparlör üzerinden ses mesajları gönderir.

Vimala vd. (2017), uzaktan erişimle bir yerel-alan ağı (LAN/WLAN: (*wireless*) *local-area network*) üzerinden yayınlanan güvenilir bir gerçek-zamanlı video akışı ve izleme (*streaming and surveillance*) sistemini tasarlayıp geliştirmişlerdir. Önerdikleri tasarım Python ile programlanmış gece görüşü kameralı (R-PI NoIR kamera V2 modülü) R-PI 3 Model B ile gerçekleştirilmiştir. R-PI'nin IP adresi alınır ve VNC sunucusu ana cihazda (R-PI'nin da aynı *yerel-alan ağına* bağlı olduğu “kişisel bilgisayar / akıllı telefon”) uzak kullanıcıya sanal masaüstü oluşturmak için kurulur. Bir VNC görüntüleyici çalıştıran ana cihaz üzerinde *canlı video yayını* için Python ile yazılmış kod ile başlatılan VLC ortam yürütücüsü hizmet verir.

Yüzgeç ve Aba (2007), birçok donanım ve sensör kullanarak R-PI 3 tabanlı bir *akıllı ev sistemi* ilk örneğini yazılımsal ve donanımsal olarak gerçekleştirmişlerdir. Engelliler, yaşlılar ve teknolojiyi yaşamlarında kullanmak isteyenlere daha iyi bir yaşam standardı sunulması için birçok senaryoyu dikkate almışlardır. Python kullanmışlar, uzaktan akıllı eve bağlanmak ve akıllı ev üzerindeki R-PI'ya ağ üzerinden erişim için PHP ile web arayüzü geliştirmişlerdir. Çeşitli sensörler ve ikaz uyarı malzemeleri ilk örnek bir ev içerisine yerleştirilmiştir. Bu uygulamayla, kullanıcılar kapıyı açabilecek, ses ötesi sensör ile kapı açıldığı zaman uyarı alabilecek, ışıkları açabilecek/kapatabilecek, sıcaklığı kontrol edebilecek, hareket sensörü ile ev içerisindeki güvenliği takip edebilecek ve evdeki kamera ile ev dışındayken evi izleyip görüntü alabileceklerdir.

1.2. Çalışmanın Katkısı

Bu çalışmada, R-PI, RFID, *web servisleri* ve Node.js tabanlı yeni bir *kullanıcı doğrulama sistemi* geliştirildi. RFID teknolojisi olarak NFC (*near field communication* – yakın alan iletişimi) sistem standardı kullanıldı. NFC özelliğine sahip kartlar ve geliştiricilerin akıllı telefonları RFID okuyucuya yaklaştırılarak kullanıcı doğrulaması yapıldı. Çalışmanın katkısı, (i) R-PI ve RFID okuyucu ile girdi veriyi yakalama-iletme, (ii) geliştirilen Node.js uygulaması aracılığıyla girdi veriyi alma, bir önceki çalışmada geliştirilmiş olan *servis yazılımı* ile konuşma ve çıktıyı görüntüleme ve (iii) *servis yazılımı* üzerinden ilgili veri yönetiminin gerçekleştirilmesi iş-paketlerinden oluşan esnek ve gürbüz bir donanım-yazılım gerçekleştirilmesi oluşturmaktadır. Metnin geri kalan kısmı şu şekilde düzenlenmiştir. 2. Bölüm'de kullanılan teknoloji ve 3. Bölüm'de geliştirilen sistem ile ilgili bilgiler verilmiştir. Yapılan deneysel çalışma 4. Bölüm'de, sonuçlar ve öneriler ise 5. Bölüm'de sunulmuştur.

2. KURAMSAL ALTYAPI

Bu bölümde, çalışma kapsamında geliştirilen sistem için kullanılan teknoloji ile ilgili ayrıntılı bilgi verilmiştir.

2.1. Raspberry Pi

R-PI, İngiltere'deki "Raspberry Pi Foundation" tarafından desteklenen 45 gram kütlesinde ve kredi kartı büyüklüğünde küçük bir bilgisayardır. SPI *iletişim protokolü* ile sensörlerle iletişim kurabilme olanağı sağlayan ve GPIO olarak bilinen bacakları bulunur. İşlemci, işletim sistemi (Debian GNU / Linux, Fedora, Arch Linux ve türevleri), RAM, grafik işlemcisi, USB girişi, ses kablo-ucu, video çıkışı, SD kart okuyucu, CSI (*camera serial interface* – kamera seri arayüzü) bağlantısı, Ethernet ve düşük-seviye çevre-birimlerine sahiptir. Örneğin, "kablolu İnternet bağlantısı" ile İnternete bağlanabilir. Ayrıntılar için "<https://www.raspberrypi.org>" ve "<https://www.robotistan.com/raspberry-pi>" bağlantılarına başvurulabilir. Tablo 1'de Arduino ve R-PI arasında kısa bir karşılaştırma verilmiştir.

Tablo 1. Arduino – R-PI karşılaştırması

| Özellik | Arduino | R-PI |
|---------------------------------------|---|--|
| Analog veri okuma | doğrudan okunabiliyor | ADCs (<i>analog to digital converters</i>) ile mümkün |
| Çıkış | kendi çıkışı yok | HDMI (<i>high-definition multimedia interface</i>) çıkışı (<i>port</i>) ile herhangi bir ekrana bağlanabilir |
| Program çalıştırma (aynı anda) | basit bir anakart ve bir program çalıştırabiliyor | genellikle Linux işletim sistemini içeren genel amaçlı bir bilgisayar ve pek çok program çalıştırabiliyor |
| Programlama | C/C++ destekleniyor | pek çok programlama dili destekleniyor |

Tablo 1'de görüldüğü gibi, özellikle kendi çıkışı ile harici bir ekrana bağlanabilmesi ve farklı programlama dilleri ile geliştirme yapılabilmesi R-PI'nin önemli artıları arasındadır.

2.2. NFC ve RESTful Web Servisleri

Kullanılan NFC sistem standardı ve geliştirilmiş olan *servis yazılımı* ile ilgili ayrıntılar için bir önceki çalışma olan Ardıç vd., 2019 çalışmasına başvurulabilir.

2.3. Node.js

Node.js, JavaScript (JS) kodunu tarayıcı dışında (*sunucuda*) işleten açık-kaynak çapraz-platform bir çalıştırma ortamıdır (*run-time environment*). Linux, Mac OS X, Unix ve Windows gibi işletim sistemlerinde çalışabilir. Kullandığı *asen kron programlama* belleğin etkin kullanımını sağlar. Node.js ile hem *istemci* taraflı hem de *sunucu* taraflı şu uygulamalar gerçekleştirilebilir: (i) form verisi toplama. (ii) dinamik sayfa içeriği üretme. (iii) *sunucuda* dosya oluşturma, açma/kapama, okuma/yazma ve silme gibi işlemler yapma ve içeriği *istemciye* döndürme. (iv) veri tabanında veri ekleme, silme ve değiştirme gibi işlemler yapma. Ayrıntılar için "<https://nodejs.org/>" bağlantısına başvurulabilir.

2.4. Model–View–Controller (MVC)

MVC, kullanıcı arayüzü içeren uygulamalar geliştirmede yaygın olarak kullanılan yazılım mimarisel bir *tasarım örüntüsüdür*. Bir uygulamayı, (1) model, (2) görünüm ve (3) kontrolör/denetçi olarak birbirine-bağlı üç temel ilgi noktasına (*mantıksal bileşenler/nesnelere*) ayırır (Karakoç ve Gunay, 2020). Ayrıca, bu bileşenlerin birbirleri arasındaki iletişim biçimini de tanımlar. Bu yöntemle, kullanıcı arayüzlerini *veri modelleri* ile kolaylıkla ilişkilendirmek, nesne-yönelimli programlama (*object-oriented programming*) ile kısa sürede *genişletilebilir* yazılımlar geliştirmek ve nesne-kodunu yeniden kullanabilmek mümkün olur. Ayrıntılar için "https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm" bağlantısına başvurulabilir.

3. UYGULAMA

Bu bölümde, çalışma kapsamında geliştirilen sistemi oluşturan *donanımsal* ve *yazılımsal* gerçeklemler ile ilgili ayrıntılı bilgi verilmiştir.

3.1. Yapılan Gerçeklemler

Tablo 2’de, R-PI 3 ve *web servisleri* kullanılarak yapılan gerçeklemler verilmiştir.

Tablo 2. Çalışma kapsamındaki gerçeklemler

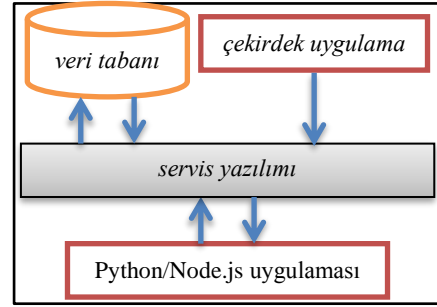
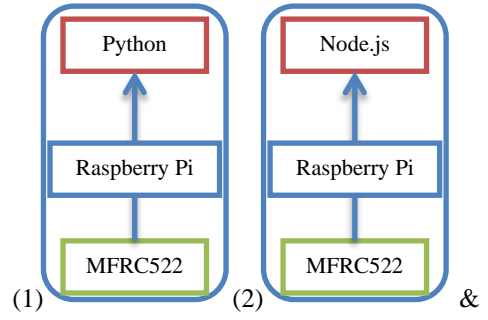
| # | R-PI 3* | GUI [^] | Deneyimleme Sorunları |
|-----|---------|------------------|-----------------------------|
| (1) | Python | - | <i>görüntüleme</i> |
| (2) | Node.js | Node.js | sorunsuz çalışan bir sistem |

* (R-PI gerçeklemleri için bu uygulama kullanılır)

[^] (*graphical user interface* – grafiksel kullanıcı arayüzü: görüntüleme işlemleri için bu uygulama kullanılır)

Tablo 2’de yer alan gerçeklemlere ilişkin süreç şu şekildedir. İlk olarak, geliştirilen Python uygulaması *servis yazılımı* ile konuşur. Daha sonra, Python’da GUI desteği bulunmadığı için Node.js uygulaması geliştirilir. Bu uygulama *servis yazılımı* ile konuşur ve GUI desteğiyle hem istek (*request*) göndermek hem de yanıt (*response*) almak ve çıktıyı görüntülemek mümkün olur. Görüntüleme işlemleri ise harici bir ekran kullanılarak yapılır. Bu şekilde, Python’daki görüntüleme sorunu çözülmüş olur. Şekil 1’de, bu çerçeve genel olarak verilmiştir.

Bu çalışmada, ilk adımda görüntüleme konusundan bağımsız olarak sadece *kullanıcı doğrulama sistemine* odaklanıldığı için bir Python uygulaması geliştirildi, fakat ilerleyen aşamalarda *kullanıcı etkileşiminin* önem kazanması ve Python ekosisteminin mevcut çatı-kütüphane çözümlerinin web arayüzü geliştirmek için bekleneni karşılayamamasına bağlı olarak, cihaz ile iletişim kuran ve sonuçları tarayıcıya döndüren uçtan-uca (*full-stack*) bir Node.js uygulaması geliştirildi.



Şekil 1. Çalışma kapsamındaki gerçeklemler

Şekil 1’de yer alan gerçeklemlerde, “<https://github.com/miguelbalboa/rfid>” bağlantısındaki kütüphane kullanılarak, MFRC522 sensörü ile okunan veri R-PI’ye iletilir. MVC-tabanlı Node.js uygulamasını içeren gerçeklemler (2)’de, arka-uç (*back-end*) işlemleri ve ön-uç (*front-end*) işlemleri (sayfa yordamları ve buton eylemlerini (*events*) yakalama işlemleri) betikler kullanılarak yapıldı. Tüm NFC kart bilgilerini okuma ve iletme işlemlerinde, sensör verisi RC522JS ara-kütüphanesi (*wrapper*) (<https://www.npmjs.com/package/rc522-rfid>) kullanılarak alındı. Ayrıca, R-PI ve MFRC522 arasında iletişim köprüsü kurabilmek ve gerçek-zamanlı veri okuma ve doğrulama için arka-uca gönderme işlemleri yapacak bir uygulama geliştirmek amacıyla soket teknolojisi (<https://www.npmjs.com/package/socket.io>) kullanıldı. PHP için kullanılan bir teknoloji olan Twig bir şablon motoru (*template engine*) olup, şablonları “en uygun şekilde getirilmiş / eniyilemesi yapılmış” (*optimized*) düz (*plain*) PHP koduna derler (*compiling*). Bu çalışmada, Twig kullanmak amacıyla Node.js için bir ara-kütüphaneden (<https://www.npmjs.com/package/twig>) yararlanılmıştır. Ayrıca, ortak-kullanılan şablonlar oluşturup, aynı kodu birçok defa yazmak gerekmeden bu şablonları HTML’e çevirip görüntüleme sayfalarında (*views*) yeniden kullanmak mümkün olmuştur.

Etiket verisini yakalama-iletme R-PI ve RFID okuyucuda olup, geliştirilen Python ve Node.js uygulamaları ise *servis yazılımı* ile konuşmaktadır.

3.2. Donanımsal Gerçekleme

Bu çalışmada, R-PI ile yapılan gerçeklemede, bir önceki çalışmada Arduino ile yapılan gerçeklemeye göre yüksek derece grafiksel destek verebilecek bir altyapı mevcut olduğu için ek donanıma ihtiyaç olmamış ve görüntüleme için sadece bir ekran yeterli olabilmektedir.

3.3. Yazılımsal Gerçekleme

Bir önceki çalışmada oluşturulmuş olan veri tabanı, yeni tablolar eklenerek ve bu tablolar arasında ilişkiler kurularak, katılım verisini etkin bir biçimde saklayabilecek hâle getirildi. Tablo 3'te, bu veri tabanına ilişkin sıradüzen (*hierarchy*) verilmiştir.

Tablo 3. Veri tabanı sıradüzeni

| Tablo* | Bilgi | İlişki | Ayrıntı |
|--------------------------|-----------------|-----------------------|--|
| Participant-Type | Katılımcı türü | tanımlama | ID |
| Card | Kart | katılımcı türü | ID – kod (<i>Hex + Dec</i>) – bilgi – kod bilgisi (kart bilgisi) [kart ekleme + kart kodu ekleme + mevcut kodu yenisi ile güncelleme] |
| Card-Code | Kart kodu | - | - |
| Participant | Katılımcı | katılımcı türü + kart | ID – kod – tanım (katılımcı bilgisi: kart bilgisi + isim bilgisi) tür bilgisi |
| Participant-Model | Katılımcı model | katılımcı | ID – kart kodu – bilgi – isim bilgisi (ekleme) |

| | | | |
|--|-------------------------|-----------------|--|
| Participant-Log | Katılımcı kayıt | katılımcı | katılımcı-ID kart-ID – okuma durumu |
| Custom-Participant-Model-List[^] | Katılımcı model listesi | katılımcı model | - |

* (tablo isimleri her kelime “-” ile birleştirilerek belirtilmiştir)

[^] (kullanıcı tanımlı *katılımcı model* listesi/dizelgesi, *servis yazılımından* dönen bilgiyi kullanabilmek için oluşturulmuştur)

Bu yapıda, “kod verisi” boş-değer (*null*) olan kart veya “kart verisi” boş-değer olan katılımcı *geçersiz* olarak değerlendirilmektedir. Tablo 4'te, geliştirilen uygulamalar ile ilgili ayrıntılar verilmiştir.

Tablo 4. Geliştirilen uygulamalar

| Uygulama | Bilgi | Ayrıntı |
|----------------------|--|--|
| 1 - Core* | Çekirdek/tabana uygulama | Java uygulaması |
| 2 - Rest* | Depolama işlemleri (veri tabanı ve <i>web servisleri</i>) | Java uygulaması (<i>servis yazılımı</i>) |
| 3 - Nodejs | Görüntüleme işlemleri (<i>kullanıcı arayüzleri</i>) | Node.js uygulaması |
| 4 - Raspberry | Donanımsal kısım | R-PI 3 gerçeklemesi |

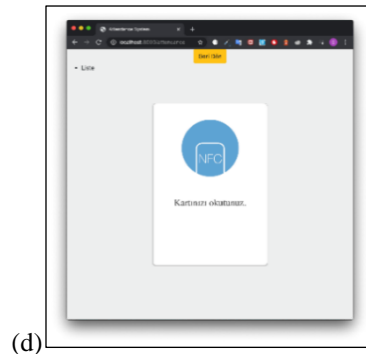
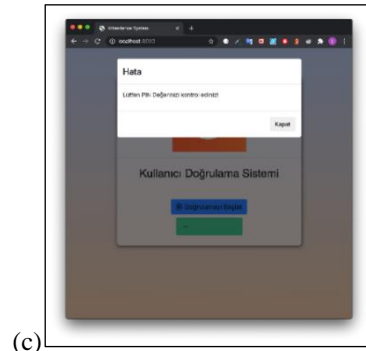
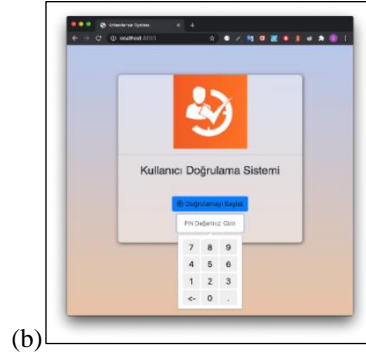
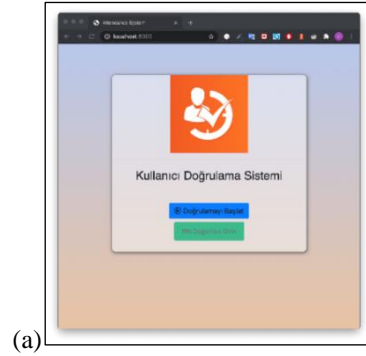
* (bir önceki çalışmada geliştirilmiş ve bu çalışmada R-PI 3 gerçeklemesine uyarlanmışlardır)

Çekirdek uygulamada, *entity* paketinde veri tabanı tablolarına ilişkin sınıflar, *mapper* paketinde sorgu cümlecikleri, *model* paketinde *özelleştirilmiş* sınıf tanımlamaları, *utility* paketinde tarih-zaman, karakter dizisi ve liste işlemleri için geliştirilen özelleştirilebilir yardımcı sınıflar, *configuration* paketinde *servis yazılımı* üzerinden servis edilen API'lere ulaşılacak yapılandırma bilgileri, *constant* paketinde ise sabit tanımlamaları bulunmaktadır. Örneğin, liste yardımcısı ile listedeki tüm boş kayıtlar çıkartılır ve bu liste döndürülür. Kart-ID veya kart-kodu verisi ile kart, “kart-ID verisi ve katılımcı-ID verisi” üzerinden katılımcı veya katılımcı-

model veya doğrudan tüm katılımcılara ilişkin katılımcı(-model) listesi sorgulanabilmektedir. Servis yazılımındaki *controller* paketinde ham sorgulara ilişkin API tanımlamaları, *view* paketinde ise ham sorguların bir arada kullanılmasıyla elde edilen daha büyük veri listelerine ilişkin API tanımlamaları yer almaktadır. Bu program ile ilgili ayrıntılar için bir önceki çalışma olan Ardıç vd. (2019) çalışmasına başvurulabilir (benzer paket tanımlamalarına ilişkin örnek başka bir uygulama için Bkz. Karakoc ve Gunay 2020).

Geliştirilen Node.js uygulaması üç arayüz ekranı içerir: (1) “hoş geldiniz” iletisi ile bekleme, (2) RFID okuyucu ile ilgili işlemler (servise HTTP isteği gönderme) ve (3) veri tabanına kayıtlı olma durumu ile ilgili işlemler (servisten dönen JSON (*JS Object Notation*) yanıtını alma ve çıktığı görüntüleme). Uygulama “localhost:8000” üzerinde başlatıldığında, ilk olarak ana sayfa (Şekil 2(a)) görüntülenir. Daha sonra, ekrana dokunulduğunda/tıklandığında PIN değeri girişi ekranı (Şekil 2(b)) görüntülenir. Yanlış bir PIN değeri girildiğinde hata iletisi görüntülenirken (Şekil 2(c)), doğru PIN değeri girildiğinde ise “(lütfen) kartınızı okutunuz” iletisi ile NFC okuma sayfasına (Şekil 2(d)) yönlendirme yapılır. Bir NFC-etiket MFRC522 sensörüne yakın olduğunda, kart kimliğine ek olarak, okunan etiket verisi veri tabanına kayıtlı ise “giriş izni verildi” ve “merhaba / hoş geldiniz ***(bu veriye ilişkin kullanıcı adı)” iletileri, diğer durumda ise “giriş onaylanmadı/yapılamadı” iletisi ile ekran güncellenir.

Şekil 2’de verilen ekran görüntüleri web tarayıcısı üzerinde çalışan uygulama aracılığıyla elde edilmiştir.



Şekil 2. (a) ana sayfa, (b) yanlış PIN değeri girişi, (c) doğru PIN değeri girişi ve (d) NFC okuma sayfası

4. BULGULAR

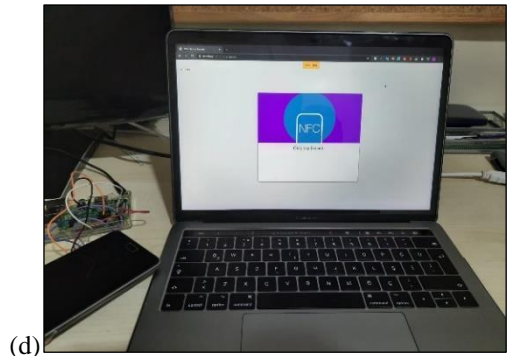
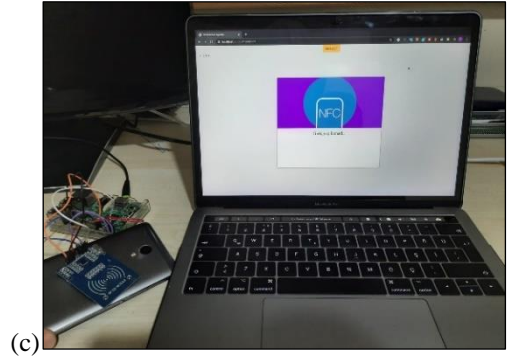
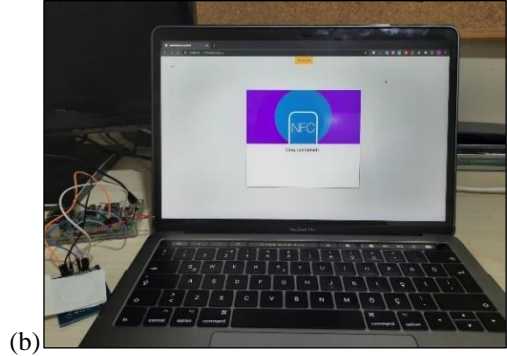
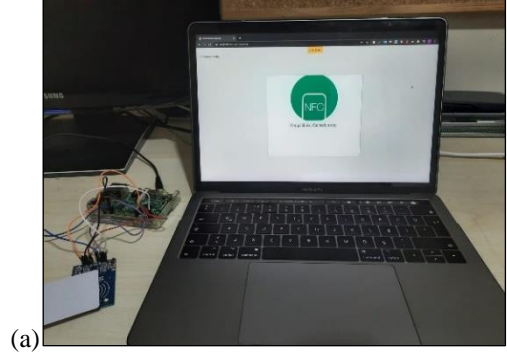
Bu bölümde, deneysel kurulum, yapılan deneyler ve elde edilen sonuçlar ile ilgili ayrıntılı bilgi verilmiştir.

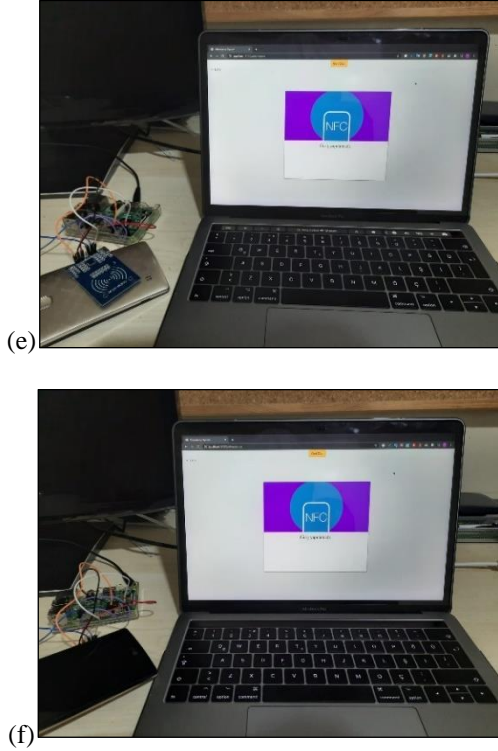
4.1. Deneysel Kurulum

Geliştirilen sistemi sınavabilmek için hem NFC özelliğine sahip kartlar hem de geliştiricilerin akıllı telefonları kullanılmıştır. NFC özelliğine sahip iki tür kart tedarik edilmiştir: (1) geliştiriciler için kredi-kartı şeklinde beyaz renkli 2 (iki) kart ve (2) kullanıcılar için anahtarlık şeklinde mavi renkli 300 kart. Deneylerde bir NFC kart RFID okuyucuya yaklaştırıldığında, bu karta ilişkin benzersiz etiket verisi (UID: *unique identifier* – kimlik: ID – *kart kodu*) okunur ve Node.js uygulaması üzerinden *servis yazılımına* gönderilir. Bu veri tabanına kayıtlıysa (olumlu: *doğrulandı*) veriye ilişkin kullanıcı bilgisi, diğer durumda (olumsuz: *doğrulanmadı*) ise boş-değer döndürülür. Ayrıca, iki durumda da çıktı ekran üzerinde görüntülenir.

4.2. Yapılan Deneyler

İlk deneylerde, her kart RFID okuyucuya tek tek okutuldu. Kart ve kullanıcı tablolarına, karta ilişkin etiket verisini ve kullanıcıya ilişkin isim bilgisini içeren yeni birer kayıt eklendi ve ilgili kayıt bilgisi döndürüldü. Böylece, kullanıcı tablosu, her kullanıcı için otomatik artan benzersiz birincil anahtar (*primary key*) değeri ile dolduruldu. Kullanıcı isimleri ise sınama-amaçlı örnek bir dizi kullanıcıya ilişkin bilgileri içeren mevcut *Excel* listesi temel alınıp tabloya elle girilerek, mevcut kart bilgileri (*bir dizi etiket verisi*) katılımcılarla ilişkilendirildi. Kullanıcı tablosundaki ilgili *birincil anahtar* değeri her kartın üzerine çıkartma (*sticker*) ile yapıştırıldı ve her kart yeniden okutularak kullanıcı doğrulaması yapıldı. Diğer deneylerde ise “NFC uyumlu akıllı telefonlar”, varsayılan olarak kapalı olan bu özellik ilgili menüden etkinleştirilerek, RFID okuyucuya yaklaştırıldı. Kullanıcı doğrulaması için kullanıcıya yapılan dönüşler Şekil 3’te verilmiştir.





Şekil 3. (a) karta *olumlu* dönüş, (b) karta *olumsuz* dönüş ve (c – f) akıllı telefonlara *olumsuz* dönüş (çıkartı: Node.js uygulaması arayüzü)

Şekil 3(a)'da kart tespit edilip kullanıcı doğrulanırken (UID bilgisi alınır), Şekil 3(b)'de ise kart tespit edilmez ve kullanıcı doğrulanmaz. Şekil 3(c-d-e-f)'de ise akıllı telefonlar üzerinden doğrulama yapılamaz.

4.3. Deneysel Sonuçları

Kartlarla yapılan tüm deneylerde, her kart bu karta ilişkin etiket verisinin veri tabanına kayıtlı olma durumu sorgulanarak doğrulanmış ve *olumlu* dönüşler alınmıştır. Akıllı telefonlarla yapılan deneylerde ise telefonlara ilişkin etiket verisi veri tabanına kayıtlı olmadığı için doğrulanamamış ve *olumsuz* dönüşler alınmıştır. Ayrıca, telefona bağlı olarak belirli bir ölçün/standart olmadığı (örneğin base8, base64 vb. biçimlerde okuma) ve standart RFID okuyucunun bu noktada daha uygun olduğu gözlemlenmiştir.

Bir önceki çalışmada, Android'in bileşenlerine bağlı olarak yeni bir tasarım kolaylıkla yapılamamıştır. Bu çalışmada ise Node.js ile yeni arayüz tasarımları yapmak, nesneleri hem *istemci* tarafta hem de *sunucu* tarafta yakalamak ve sunucu-taraflı hafif (*lightweight*) bir uygulamayı kolaylıkla ve hızla oluşturmak mümkün olmuştur. Ayrıca, Node.js'in *istemci-taraflı* bir ortam olmasına bağlı olarak, pek çok şey kolaylıkla özelleştirilebilmiş, HTML ve CSS içeren ekrana-

duyarlı/özel (*responsive*) esnek ve özelleştirilebilir tasarımlar yapılabilmektedir.

Otomatik olarak kullanıcının yoklamasının alınıp onay/ret durumunun iletilmesi kapsamında bakıldığında, Arduino-tabanlı yapı istenilenleri tam olarak karşılayamamıştı. Arduino'nun sayısal/dijital bir görüntü çıkışının bulunmaması sebebi ile ekran bağlantısı yapılamaması ve kullanıcıya net geri dönüşlerin verilememesi söz konusu idi. Bu çalışmayla, gerçekleştirme analog hâlden (Arduino + "16 x 2 LCD ekran / Android tablet") sayısal hâle (R-PI + monitör/ekran) dönüştürülmüş oldu. Görüntüleme için HDMI üzerinden R-PI'ya bağlı bir ekran veya R-PI ile aynı ağdaki bir bilgisayara bağlı bir ekran kullanılabilir.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, R-PI, RFID, *web servisleri* ve Node.js tabanlı yeni bir *kullanıcı doğrulama sistemi* geliştirildi. Bu sistem hem NFC özelliğine sahip kartlar hem de geliştiricilerin akıllı telefonları kullanılarak başarılı bir biçimde sınılandı. Geliştirilen sistem, bir önceki çalışmada da olduğu gibi, "taşınabilir kutu" olarak kullanıma açıktır. Çalışmanın, *kararlı* ve *genişletilebilir* bir donanım-yazılım gerçekleştirilmesi olarak yeni çalışmalara yön verebileceği düşünülmektedir. Ayrıca hem *yönetici* taraflı hem de *kullanıcı* taraflı bir *katılım denetleme bilgi yönetim sistemi* oluşturup, derslerde öğrencilerin ve öğretim elemanlarının okul kimlikleri veya akademik etkinliklerde katılımcıların kartlarını kullanarak ilgili katılımları denetleyebilmek planlanmaktadır.

KAYNAKLAR

Ardıç C., Şen M., Karakoç M. (2019). RESTful Web Servisleri ve Android Kullanılarak Genel Bir Kullanıcı Doğrulama Sisteminin Arduino Mikrodenetleyici Kart ve RFID Teknolojisi ile Tasarımı ve Gerçekleştirimi, Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 12 (1), 19-29.

Awakhare M., Parmal N., Dhawale S., Dongre P., Jamgade S., Tambe A., Deulkar S., Meshram B. (2018). RFID Based E-Attendance System & Child Security System, International Journal of Engineering Science and Computing (IJESC), 8(3), 16162–16164.

Bhagya Maybel J., Umamakeswari A. (2018) Hardware Implementation of Secure Image Transmission in Raspberry PI, International Journal of Mechanical Engineering and Technology (IJMET), 9(2), 670–678.

- Dayıoğlu M. A., Uğur F., Türker U. (2016). Seralarda Nesnelerin İnterneti Teknolojisinin Uygulanması: Tasarım ve Prototip Geliştirme, Gaziosmanpaşa Üniversitesi Ziraat Fakültesi Dergisi (JAFAG), 33 (Ek sayı), 52–60.
- Ferdoush S., Li X. (2014). Wireless Sensor Network System Design using Raspberry Pi and Arduino for Environmental Monitoring Applications, *Procedia Computer Science*, 34, 103–110. The 9th International Conference on Future Networks and Communications (FNC-2014).
- İncereis N., Akgün B.T. (2017). IoT Uygulamaları için Oluşturulan Sistemde Servisler, 19. Akademik Bilişim Konferansı, 8-10 Şubat, Aksaray Üniversitesi, Aksaray.
- Karakoc M., Gunay (2020). Data Bus for Computer Based Assessment via Micro Services, the 3rd International Conference on Advanced Technologies, Computer Engineerind and Science (ICATCES) 03-05 Junes.
- Kaur S., Sharma S., Jain U., Raj A. (2016). Voice Command System Using Raspberry Pi, *Advanced Computational Intelligence: An International Journal (ACII)*, 3(3), 43–49.
- Kiran Kumar R, Mekala S M. E. (2018). Face Recognition Attendance System using Raspberry Pi, *International Journal of Pure and Applied Mathematics*, 118(20), 3061–3065, Special Issue.
- Küçükkülahlı E., Güler R. (2015). Open Source Mobile Robot with Raspberry Pi, *Balkan Journal Of Electrical & Computer Engineering*, Special Issue, 3(4), 242-247.
- Natarajan S., Deepika A., Pradeeba I., Chandramohan R. (2017). Low Cost Temperature Logging System using Raspberry Pi, *International Research Journal of Engineering and Technology (IRJET)*, 4(4), 254–258.
- Priya G. D., Harish I. (2015). Raspberry PI Based Underwater Vehicle for Monitoring Aquatic Ecosystem, *International Journal of Engineering Trends and Applications (IJETA)*, 2(2), 65–71.
- Rani R., Lavanya S., Poojitha B. (2018). IoT Based Home Security System Using Raspberry Pi with Email and Voice Alert, *International Journal of Advanced Research in Computer Science and Software Engineering*, 8(4), 119–123. Special Issue: National Conference on Emerging Trends in Engineering 2018, Conference Held at Sri Venkatesa Perumal College of Engineering & Technology, Puttur, A. P., India.
- Ranjana R., Vinoth K. (2016). Enhanced Security System for School Children and Woman Transportation Using Arduino, *International Journal of Computer Network and Security (IJCNS)*, 8(1), 6–10.
- Rjeib H. D., Ali N. S., Al Farawn A., Al-Sadawi B., Alsharqi H. (2018). Attendance and Information System Using RFID and Web-Based Application for Academic Sector, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(1), 266–274.
- Sudhakar M., Khare V., Kanth V.K. (2017). Speech to text conversion & display using Raspberry Pi, *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(4), 14–18.
- Tan P., Wu H., Li P., Xu H. (2018). Teaching Management System with Applications of RFID and IoT Technology, *Education Sciences*, 8(26), 1–13.
- Tiwari R., Singh D. K. (2017). Dushyant Kumar Singh. Vehicle Control Using Raspberrypi and Image Processing, *Innovative Systems Design and Engineering*, 8(2), 45–49.
- Vimala S. N., Chary M. V., Kiran K. R. (2017). Live Video Streaming from Remote Location Using Raspberry Pi, *Indian Journal of Science and Technology*, 10(38), 1–10.
- Yüzgeç U., Aba Ö. (2007). Raspberry Pi Kullanılarak bir Akıllı Ev Uygulaması Geliştirilmesi, *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi*, 4(1), 21–29.
- Zhong X., Liang Y. (2016). Raspberry Pi: An Effective Vehicle in Teaching the Internet of Things in Computer Science and Engineering, *Electronics*, 5(3), 56.