GU J Sci, Part C, 9(2): 171-181 (2021)

Gazi University

Journal of Science

PART C: DESIGN AND TECHNOLOGY

http://dergipark.gov.tr/gujsc

Application Development on Gas Station and Consumers with the Representational State Transfer Approach

Hüseyin POLAT[1,*] Saadin OYUCU[2]

[1]*Gazi University, Faculty of Technology, Department of Computer Engineering, 06500, Yenimahalle/ANKARA*

[2]*Adıyaman University, Faculty of Engineering, Department of Computer Engineering, 02230, Merkez/ADIYAMAN*

Article Info:

Research article
Received: 03.12.2020
Revision: 22.12.2020
Accepted: 31.12.2020

Highlights

- *JSON.*
- *MongoDB.*
- *Location Services.*

Keywords

Web Services
REST
NoSQL
AngularJS
Mobile Application

Graphical/Tabular Abstract

In this study, a system is proposed to provide interaction and communication between fuel stations and fuel consumers. In the proposed system, there are two basic layers as backend and frontend.

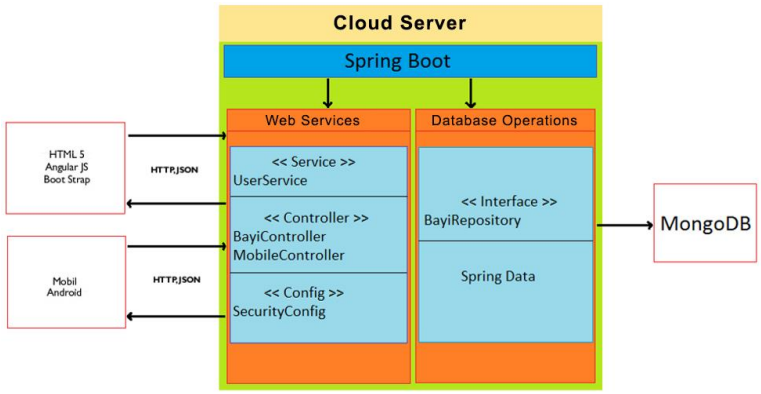


Figure A. Client and web service infrastructure of the proposed system

Purpose: In the backend, web services have been developed with the Representational State Transfer (REST) approach [2]. With the help of the developed web services, more than one fuel station and more than one user can perform transactions simultaneously. Through the developed web interface, fuel stations can save their services and fuel prices to the system. Consumers, on the other hand, can instantly monitor the services offered by fuel stations and fuel prices via the developed web interface or mobile application.

Theory and Methods: The proposed system requires the use of different technologies together. For this reason, the materials and methods used in the development of the proposed system are explained in detail under different headings. Web services, interface development library, non-relational database, location and mapping services are explained in detail.

Results: The basis of the proposed system is the web service structure. The developed web services are based on REST. The web interface is developed using AngularJS and HTML. A mobile application has been developed for users to access this system more easily and to benefit from the possibilities of mobile systems. The developed mobile application provides access to REST web services with its internet infrastructure, and web services to the database when necessary. JSON data format is preferred as a data structure. Thus, more efficient and faster data communication is provided.

Conclusion: In the usage tests of the developed system, it has been seen that the web services provide smooth service on mobile and web platforms. Thanks to the platform-independent operation of the system, it has been ensured that it can be integrated into different applications. The security of the web services is provided by using Spring security components by preferring the Spring Boot framework. With the proposed system, not only a system that will serve gas stations but also an architectural structure that will be used in different areas of use has been presented.

*Corresponding author, e-mail: polath@gazi.edu.tr
DOI: 10.29109/gujsc.835482

GU J Sci, Part C, 9(2): 171-181 (2021)

Gazi Üniversitesi

Fen Bilimleri Dergisi

PART C: TASARIM VE TEKNOLOJİ

http://dergipark.gov.tr/gujsc

Application Development on Gas Station and Consumers with the Representational State Transfer Approach

Hüseyin POLAT[1,*] Saadin OYUCU[2]

[1]*Gazi Üniversitesi, Teknoloji Fakültesi, Bilgisayar Mühendisliği Bölümü, 06500, Yenimahalle/ANKARA*

[2]*Adıyaman Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 02230, Merkez/ADIYAMAN*

Makale Bilgisi

Araştırma makalesi
Başvuru: 03.12.2020
Düzeltme: 22.12.2020
Kabul: 31.12.2020

Keywords

Web Services
REST
NoSQL
AngularJS
Mobile Application

Abstract

Although Turkey is close to countries with rich oil deposits, it is among the countries that use expensive fuel oil. This situation forces consumers to save on fuel expenditures. On the other hand, the high number of fuel stations and the variety of fuel products cause fuel stations to offer different services and prices in a competitive environment. This change in the services and prices offered affects the preferences of the consumers of the fuel stations. However, consumers should have more detailed information about fuel stations. In this study, an application model is proposed to provide communication and interaction between fuel stations and fuel consumers. For this purpose fuel prices and service information provided by the stations were sent to servers in the cloud using a representative state transfer approach. Then this information on cloud servers has been presented to consumers through developed web service, web interface and mobile application. In this way, services and fuel prices provided by fuel stations were monitored via web browsers and smartphones. On account of the proposed application model, the stations that are closest to the consumer, where they can find the service they want, and that offer the lowest fuel price, can be easily identified.

Anahtar Kelimeler

Web Servisler
REST
NoSQL
AngularJS
Mobil Uygulama

Temsili Durum Transferi Yaklaşımı ile Akaryakıt İstasyonu ve Tüketicileri Üzerine Uygulama Geliştirme

Öz

Türkiye zengin petrol yataklarına sahip ülkelere yakın olmasına rağmen pahalı akaryakıt kullanan ülkeler arasında yer almaktadır. Bu durum tüketicileri yakıt harcamalarında tasarruf etmeye zorlamaktadır. Öte yandan akaryakıt istasyonlarının çokluğu ve akaryakıt ürünlerinin çeşitliliği, akaryakıt istasyonlarının rekabet ortamında farklı hizmet ve fiyatlar sunmasına neden olmaktadır. Sunulan hizmet ve fiyatlardaki bu değişiklik tüketicilerin akaryakıt istasyonları tercihlerini etkilemektedir. Ancak tüketicilerin akaryakıt istasyonları hakkında daha detaylı bilgiye sahip olmaları gerekmektedir. Bu çalışmada, akaryakıt istasyonları ile akaryakıt tüketicileri arasındaki iletişim ve etkileşimi sağlamak için bir uygulama modeli önerilmiştir. Bu amaçla istasyonlar tarafından sağlanan akaryakıt fiyatları ve hizmet bilgileri temsili durum transferi yaklaşımı kullanılarak buluttaki sunuculara gönderilmiştir. Daha sonra bulut sunucuları üzerindeki bu bilgiler, geliştirilen web servisi, web ara yüzü ve mobil uygulama aracılığıyla tüketicilere sunulmuştur. Bu sayede akaryakıt istasyonlarının sunduğu hizmetler ve akaryakıt fiyatları web tarayıcıları ve akıllı telefonlar üzerinden takip edildi. Önerilen uygulama modeli sayesinde tüketiciye en yakın, istediği hizmeti bulabileceği ve en düşük akaryakıt fiyatını sunan istasyonlar kolaylıkla tespit edilebilmektedir.

1. GİRİŞ (INTRODUCTION)

Akaryakıt istasyonlarının sağladığı hizmetlerin ve akaryakıt fiyatlarının farklı istasyonlarda değişiklik gösterdiği bilinmektedir. Örneğin bir akaryakıt istasyonu engelli tuvaleti hizmeti, ibadethane ve çocuk bakım odası gibi hizmetler sunarken, başka bir akaryakıt istasyonu bu hizmetleri sağlamayabilir. Akaryakıt istasyonlarının sunduğu hizmetlerin akaryakıt tüketicileri tarafından bilinmesi, akaryakıt

istasyonlarının tüketici tarafından tercih edilmesini sağlayacaktır. Ayrıca akaryakıt istasyonu sayısı çok ve akaryakıt ürünlerinin çeşidi fazladır. Bu nedenle daha uygun fiyatlı akaryakıt satışı sunan istasyonlar hakkında tüketicilerin bilgi sahibi olmaları tüketicinin lehine olacaktır. Bunun yanı sıra hizmet ve uygun fiyat avantajı sağlayan akaryakıt istasyonunun en yakın hangi konumda olduğunun bilinmesi de tasarruf açısından önem arz etmektedir [1].

Günümüzde her sektörde tüketiciler, satın alacakları bir ürün ve hizmet hakkında daha önceki tüketici yorumlarını dikkate almaktadır. Akaryakıt tüketicilerinin de tercih edecekleri istasyonlar hakkında daha önceki tüketici yorumlarını okuyup bu yorumlara göre seçim yapması beklenmektedir. Diğer yandan tüketicinin akaryakıt istasyonları hakkında yaptığı yorumlar akaryakıt istasyonlarının fiyat ve hizmet kalitesini net olarak ortaya koyacaktır. Bu nedenle akaryakıt istasyonları ile tüketiciler arasındaki iletişimi, etkileşimi ve haberleşmeyi sağlayan bir sisteme ihtiyaç duyulmaktadır.

Bu çalışmada akaryakıt istasyonları ve akaryakıt tüketicileri arasındaki iletişimi, etkileşimi ve haberleşmeyi sağlayabilmek için bir sistem önerilmiştir. Önerilen sistem içerisinde arka plan ve ön plan olarak iki temel katman yer almaktadır. Arka planda Temsili Durum Transferi (REST: Representational State Transfer) yaklaşımı ile web servisler geliştirilmiştir [2]. Geliştirilen web servisler yardımı ile birden fazla akaryakıt istasyonu ve birden fazla kullanıcının aynı anda işlem yapabilmesi sağlanmıştır. Geliştirilen web arayüzü aracılığıyla akaryakıt istasyonları, sunmuş olduğu hizmetleri ve akaryakıt fiyatlarını sisteme kaydedebilmektedir. Tüketiciler ise geliştirilen web arayüzü veya mobil uygulama üzerinden akaryakıt istasyonlarının sunmuş olduğu hizmetleri ve akaryakıt fiyatlarını anlık olarak takip edebilmektedir.

Önerilen sistem sayesinde akaryakıt tüketicileri kendilerine en yakın konumda bulunan, en iyi hizmet alacakları ve en düşük akaryakıt fiyatı sunan istasyonları kolay bir şekilde tespit edebilmektedir. Tüketici, her istasyonun iletişim bilgilerini akaryakıt fiyatlarını, istasyonun verdiği hizmetlerin detaylarını (market, yeme-içme, tamir-bakım, ibadethane vb.) ve istasyonun kendi konumuna olan mesafesini görebilmektedir. Aynı zamanda tüketici seçtiği istasyona yol tarifi alabilmekte ve istasyon hakkında daha önce yapılmış kullanıcı yorumları var ise bu yorumları okuyabilmektedir. Mobil uygulama pazarındaki benzer uygulamalardan farklı olarak geliştirilen sistem, kullanıcılara harita üzerinden hizmet verebilmektedir. Kullanıcının konumuna göre yakınında bulunan akaryakıt istasyonları harita üzerinde gösterilmektedir.

Önerilen sistem birden fazla bileşeninin bir arada kullanılması ile oluşmaktadır. Harita, Küresel Konumlama Sistemi (GPS: Global Positioning System), web servisler, veri tabanı ve mobil uygulama önerilen sistemde birlikte kullanılmıştır. Literatür incelendiğinde belirtilen farklı bileşenlerin kullanılarak çok farklı çalışmalar yapıldığı görülmüştür. Günay Gültekin 2014 yılındaki yüksek lisans tezinde [3], akıllı telefonların standart bir özelliği olan GPS özelliğini kullanmıştır. GPS sayesinde konum bilgilerini kullanarak kullanıcıya en yakın elektronik mağazaları bulan bir mobil uygulama geliştirmiştir. Bu çalışmadaki temel amaç kullanıcıların geliştirilen mobil uygulamayı kullanarak zamandan tasarruf etmelerini sağlamaktır. Aynı şekilde Patrick De Haan ve arkadaşlarının geliştirdiği GasBuddy uygulaması [4], kullanıcıların zaman tasarrufu elde edebilmesi için geliştirilmiş bir mobil uygulamadır. Akaryakıt istasyonları için geliştirilen ve yaklaşık 60 milyon kullanıcısı olan bu uygulama sadece Avusturalya, Amerika ve Kanada'da kullanılmaktadır. Akaryakıt fiyatlarını kullanıcılar güncellemektedir. Fiyat bilgisi kullanıcılar tarafından rapor edildikçe kullanıcılar ödüllendirilmektedir. GasBuddy uygulamasına benzer diğer bir uygulama ise Wikicamps şirketinin geliştirdiği Fuel Map uygulamasıdır [5]. Fuel Map Avusturalya genelinde akaryakıt istasyonlarının ve akaryakıt fiyatlarının takip edilebildiği bir uygulamadır. Bilgiler kullanıcılar tarafından uygulamaya eklenmektedir. Harita üzerinde gösterilen akaryakıt istasyonları arasında fiyata göre ve yakınlığa göre listeleme yapılabilmektedir.

Mevcut uygulamalar incelendiğinde her birinin farklı özellikleri olduğu ve farklı teknolojik alt yapı ile geliştirildiği görülmüştür. Bu teknolojilerden biri olan web servisler, önerilen sistemin arka planında iş yönetimini ve organizasyonunu yapacak temel bileşeni oluşturmaktadır. Literatürde web servis yaklaşımları değişik uygulamalarda farklı amaç ve hedefler için kullanılmıştır. Ramanathan ve Korte yaptıkları bir çalışmada web servisleri kullanarak hava durumu verilerine erişmek için yazılım servis mimarisi geliştirmişlerdir [6]. Menghui ve arkadaşlarının yaptığı bir çalışmada ise Javascript Nesne Gösterimi (JSON: JavaScript Object Notation) tabanlı RESTful web servisleriyle tarayıcı tabanlı bir

harita çizme uygulaması geliştirmiştir [7]. Menghui ve arkadaşları, RESTful web servislerini kullanarak dağıtılmış veri entegrasyonu için bir çözüm önermiştir. Çalışmada, geleneksel web hizmetleri ve RESTful web hizmetleri hakkında derinlemesine bir analiz yapılmıştır. RESTful web hizmetlerinin dağıtılmış veri entegrasyonu için daha uygun olduğunu açıklanmıştır.

Literatürdeki çalışmalar incelendiğinde benzer sistemleri geliştirmek için birçok mimarinin önerildiği görülmüştür. Ancak uygulamalar sadece kullanıcı odaklıdır ve kullanıcıların fiyat girmesini istemektedir. Önerilen sistemde ise akaryakıt istasyonları kendi fiyat ve hizmetlerini anlık olarak güncelleyebilmektedir. Tüketiciler ise bu hizmet ve fiyat değişikliği anlık olarak takip edebilmektedir. REST web servis yaklaşımı ile geliştirilen sistem, mobil veya bilgisayar gibi farklı platformlara cevap verebilmektedir. Önerilen sistemin geliştirilmesinde REST web servisleri basit ve kolayca ölçeklenebilir olduğu için tercih edilmiştir. REST sayesinde işlevsellik ve sistemin cevap verme performansı arttırılmıştır. Bulut üzerinde JSON tabanlı ilişkisel olmayan bir veri tabanı kullanılarak isteklere karşı verilecek cevapların hızlı olması sağlanmıştır. Konum servisleri kullanılarak harita üzerinde kullanıcı dostu bir arayüz geliştirilmiştir. Geliştirilen arayüz mobil uygulama, mobil ve web tarayıcıda sorunsuz hizmet vermektedir. Geliştirilen sistem sayesinde akaryakıt istasyonlarının ve akaryakıt tüketicilerinin rahatlıkla kullanabilecekleri bir platform geliştirilmiştir.

2. MATERYAL VE METOTLAR (MATERIALS AND METHODS)

Önerilen sistem faklı teknolojilerinin bir arada kullanılmasını gerektirmektedir. Bu nedenle önerilen sistemin geliştirilmesinde kullanılan materyal ve metotlar farklı başlıklar altında detaylı olarak açıklanmıştır.

2.1. Web Servisler (Web Services)

Web servisler kendi iletişim protokollerini belirlemekte ve veri erişimi için genel bir arabirim oluşturmaktadır. Böylelikle birden fazla platformdaki istemci istenilen hizmete eşit haklara sahip olarak erişebilmektedir. Günümüzde web servis geliştirmek için iki yaklaşım bulunmaktadır. Bunların ilki Basit Nesne Erişim Protokolü (SOAP: Simple Object Access Protocol) diğeri REST web servis yaklaşımıdır.

Basit bir nesne erişim protokolü olarak tanımlanan SOAP, bilgisayar ağlarında yapılandırılmış verinin alışverişinde kullanılan bir protokoldür [8].Web Servis Tanımlama Dili (WSDL: Web Service Definition Language), iletileri web üzerinden erişilebilir hale getirmek için SOAP ile birlikte kullanmaktadır [9]. SOAP, veri alışverişi için kodlama kuralları sağlamakta ve genellikle uç nokta prensibine göre çalışan yaklaşımdır. REST web servis yaklaşımı ise web servisleri geliştirmek için Hiper Metin Transfer Protokolünü (HTTP: Hypertext Transfer Protocol) kullanan bir mimaridir [10]. REST mimarisindeki temel fikir, makineler arasında veri transferi için HTTP katmanının üstünde çalışan bir protokol kullanmak yerine gelişmiş HTTP mimarisi özelliklerini kullanmaktır. REST arayüzü HTTP standart işlemlerini (GET, PUT, POST ve DELETE) kullanarak web servis işlemlerini gerçekleştirmektedir. REST veri odaklı çalışmaktadır. Bu nedenle sunucunun yanıtı JSON tabanlı veri kümesi olarak elde edilebilmektedir. Bu durum kullanım kolaylığı ve esneklik sağlamaktadır.

Bu çalışmada geliştirilen sistem hem mobil uygulama hem de web arayüzü üzerinden hizmet verecektir. Bu nedenle tercih edilecek alt yapının her iki platformda da sorunsuz çalışması gerekmektedir. Mobil uygulamada tercih edilecek mimari veya protokol, büyük ölçüde iş hedeflerine ve elde edilmesi gereken ihtiyaçlara bağlıdır. REST mimarisi basit istek-yanıt prensibine dayandığından SOAP'a göre kolay anlaşılabilir bir yapı içerisindedir. Bu basit istek-yanıt yapısı akıllı cihazlar için geliştirilen uygulamalarda iş hedeflerini oldukça kolaylaştırmaktadır. Ayrıca REST mimarisi herhangi bir teknolojiye bağlı değildir. REST mimarisi herhangi bir hata esnasında REST HTTP hatalarını geliştiriciye bildirdiği için test ve hata takip aşaması SOAP'a göre daha kolaydır. Bu nedenle çalışmada web servislerin REST yaklaşımı ile geliştirilmesinin daha avantajlı olduğu sonucuna varılmıştır.

Önerilen sistem için gerekli olan web servisler Spring kullanılarak geliştirilmiştir. Spring, Java 2 Kurumsal Sürüm (J2EE: Java 2 Enterprise Edition) belirtiminin eksikliklerini telafi etmek için hazırlanmış bir yapıdır [11]. Spring'in ana amacı, J2EE geliştirmenin karmaşıklığını azaltmaya yardımcı

olmaktır. Temel olarak Java nesnelerini özelleştirmek ve yönetmekle görevlidir. Spring modülerdir ve bağımsız olarak çalışabilen bağımsız paketlere ayrılmıştır. Bu durum geliştiriciler için kolaylık sağlamaktadır. Ayrıca, yapılandırma ve mesajlaşma dosyalarını kendi kurulumunda belirleyerek daha akıcı bir platform oluşturmaktadır.

2.2. Arayüz Geliştirme Kütüphanesi (Interface Development Library)

Önerilen sistem için gerekli olan arayüzlerin oluşturulmasında AngularJS açık kaynak kodlu web uygulama çatısı kullanılmıştır. AngularJS, Google tarafından desteklenen ve geliştirilmesi devam eden açık kaynaklı Javascript yapısıdır [12]. AngularJS'in amacı, yalnızca sunucu üzerinde geliştirilmiş mevcut araç ve yetenekleri web istemcisine kazandırmaktır. Böylelikle zengin ve karmaşık web uygulamaları rahatlıkla geliştirilmekte ve sunucu yükü azaltılmaktadır. AngularJS, test edilebilir ve kolayca genişletilebilir uygulamalar oluşturmaya olanak sağlamaktadır. Bu nedenle önerilen sistem için gerekli olan arayüzlerin geliştirilmesinde tercih edilmiştir.

2.3. İlişkisel Olmayan Veri Tabanı (Non-Relational Database)

İstasyonlardan ve kullanıcılardan elde edilen ve önerilen sistemde saklanan veriler oldukça büyük boyutlara ulaşabilmektedir. Büyük miktardaki verileri veri tabanına yazma, okuma ve analiz etme işlemlerinde bazı problemler ortaya çıkmaktadır. Büyük miktardaki veriyi ilişkisel veri tabanlarında saklamak için ilişkisel veri tabanlarında çok fazla alan ve tablo oluşturmak gerekir. Tabloların önceden tanımlanması gerektiği için verinin standart hale getirilmesi işleri zorlaştırmaktadır. Ayrıca çok fazla tablonun oluşturulması çoklu okuma ve yazma işlemelerinde performansı düşürecektir.

Geliştirilen sisteme veriler bir veya birden fazla istasyon ve kullanıcı üzerinden alınmaktadır. Verilerin veri tabanından okunması esnasında birden ise birden fazla kullanıcı veri tabanından farklı isteklerde bulunabilir. Bu durumda veri tabanının cevap verme süresi geç olursa, sistemde kilitlenmeler yaşanır. Kilitlenmeyi engelleyebilmek için performansı yüksek bir belge tabanlı İlişkisel Olmayan Veri Tabanı (NoSQL: Not Only Structured Query Language) veri tabanı tercih nedeni olmalıdır. Bunun için ise MongoDB kendini kanıtlamış bir sistemdir. MongoDB belge tabanlı bir NoSQL veri tabanıdır. MongoDB belge veri modeli, zengin sorgu desteği, yatay ölçeklenebilirlik, yüksek kullanılabilirlik, esneklik ve dinamik şema gibi birçok özelliği sunmaktadır. MongoDB veri saklama işleminde JSON veri yapısını kullanmaya izin vermektedir [13].

NoSQL veri tabanı kavramı, yıllardır bilişim dünyasında kullanılan ilişkisel veri tabanı sistemlerine alternatif olarak ortaya çıkmıştır. NoSQL, günümüzde internet ortamında giderek büyüyen verileri depolayabilmek ve yüksek erişilebilirliğin yanında yatay ölçeklenebilen sistemlere verilen genel bir isim olarak tanımlanmaktadır. MongoDB doküman tabanlı bir NoSQL veritabanı yönetim sistemidir ve karmaşık sorgular için destek sağlamaktadır. İlişkisel veri tabanlarında olduğu gibi verileri tablolarda depolamak yerine, verileri JSON formatındaki dokümanlar da depolamaktadır. NoSQL veri tabanının avantajları veri okuma ve yazma hızı, toplu veri işlemlerini desteklemesi, genişletilmesinin kolay olması ve düşük maliyetli olmasıdır [14]. Bu nedenle önerilen sistemde MongoDB NoSQL veritabanı tercih edilmiştir.

2.4. Konum ve Haritalama Servisi (Location and Mapping Services)

Önerilen sistemin temel özelliklerinden biri olan en yakın akaryakıt istasyonunu bulma özelliğini sağlayabilmek için konum servislerinden yararlanmak gerekir. Android mobil işletim sistemi, geliştirilen mobil uygulamalar için kullanıcının konum bilgisini paylaşan bir kütüphane sunmaktadır. Android işletim sisteminde mobil cihazın konumu temel olarak üç farklı yöntem ile elde edilmektedir [15]. Bu yöntemler GPS, Mobil İletişim için Küresel Sistem (GSM: Global System for Mobile Communications) şebeke sinyali ve pasif konum sistemidir.

Pasif konum sisteminde gerçekleştirilen işlem işletim sisteminin bellekte tuttuğu eski konum bilgilerinden yola çıkarak kullanıcının konumuyla ilgili bilgi edinilmesidir. Diğer yöntemde ise cihazda aktif bir GSM bağlantısı var ise Android cihazın bağlı bulunduğu baz istasyonlarından kullanıcının yeri tahmin edilebilir. Burada yapılan işlem baz istasyonu bilgisinin merkezi bir veri tabanına gönderilerek bu istasyonların konum bilgilerinden kullanıcının yerinin tahmin edilmesidir. Belirtilen bu iki yöntem anlık

konum bilgisinin bulunmasında bazı kısıtlar sunmaktadır. Bu nedenle GPS kullanımı yaygınlaşmıştır. Android işletim sistemi, herhangi bir uygulamadan konum talebi geldiğinde GPS'i devreye alarak kullanıcının konumunu uygulamaya bildirmektedir. GPS, teknolojisi gereği kapalı alanlarda çalışmamaktadır ve tam bağlantı sırasında aşırı derecede güç tüketmektedir. Bu olumsuz durumlarına karşın oldukça yüksek hassasiyette konum bildirdiğinden harita uygulamalarında GPS diğer yöntemlere göre daha çok tercih etmektedir. Bir uygulamanın konum bilgisi servislerinden yararlanması için AndroidManifest.xml dosyasında bazı izinlerin alınması gerekmektedir.

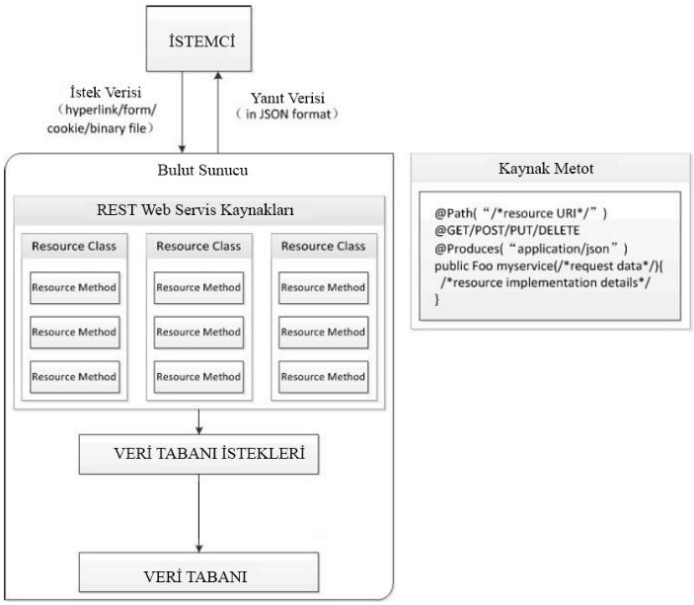
Konum servisleri birçok mobil uygulamanın önemli ve ayırt edici özelliklerinden biri olarak sıklıkla kullanılmaktadır. Konum servislerinden alınan bilgilerin görselleştirilmesi için haritalar servisi kullanılmaktadır. Google, konum servisleri ile birlikte Android için uygulama geliştirene kullanıcıya bulunduğu yeri göstermek ya da aradığı bir konumu harita üzerinde göstermek için Google içinde yer alan Google Maps Android uygulama programlama arayüzünü sunmaktadır [16]. Bu uygulama programlama arayüzünü herhangi bir uygulamaya eklemek ve harita üzerinde istenilen işlemleri gerçekleştirebilmek için Google üzerinden bir uygulama programlama anahtarının elde edinilmesi gerekir. Google, söz konusu anahtarı uygulamaya özel oluşturup kullanıcıya sunmaktadır.

3. UYGULAMA GELİŞTİRME (APPLICATION DEVELOPMENT)

Önerilen sistemin geliştirme adımları iki alt gruba ayrılmıştır. Bu gruplardan ilki arka planda hizmet verecek web servis uygulamasının geliştirilmesi ikincisi ise on planda hizmet verecek mobil uygulama ve web arayüzünün geliştirilmesi olarak belirlenmiştir. Web servis uygulaması içerisinde web istemciler geliştirilerek web arayüzü üzerinden kullanıcıların sisteme erişimi sağlanmıştır. Web servis uygulaması, web ve mobil platformlarının etkileşimli olarak çalışmasını sağlayan REST tabanlı olarak geliştirilmiştir.

3.1. Web Servis Uygulaması (Web Service Application)

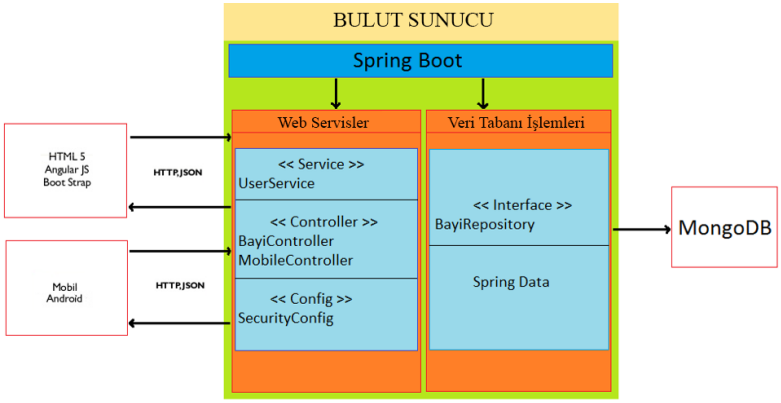
Web servis sunucusu üzerinde yer alan web servisler Spring Boot kullanılarak REST tabanlı geliştirilmiştir. Sunucuya bağlanması gereken web istemci arayüzü için ise Hiper Metin İşaretleme Dili (HTML: Hypertext Markup Language) ve AngularJs tercih edilmiştir. Şekil 1'de önerilen sistemin web servis mimarisi ve katmanları verilmiştir.



Şekil 1. *Önerilen sistemin web servis mimarisi ve katmanları*

Şekil 1'de web servis mimarisi ve kaynak metotlar gösterilmiştir. Web servis sunucusu kendisine gelen iş isteklerini Spring'e, Spring ise gelen isteğin türüne göre işlemleri gerçekleştirmekte ve kullanıcıya yanıt

vermektedir. Kullanıcı isteklerinde veri tabanı işlemi gerekiyorsa veri tabanına ulaşıp gerekli bilgiyi nesne halinde yanıtlama görevi de Spring'e verilmiştir. Kullanıcıya yanıtlar JSON formatında gönderilmiştir. Önerilen sistem için geliştirilen uygulamanın istemci ile etkileşimi gösteren alt yapı şekil 2'de gösterilmiştir.



Şekil 2. *Önerilen sistemin istemci ve web servis alt yapısı*

Şekil 2'deki yapıda belirtilen bileşenler referans alınarak web servis uygulamasının kodlama işlemleri gerçekleştirilmiştir. Sunucu, Model – Görünüm – Denetleyici kalıbıyla kodlama işlemi gerçekleştirilmiştir. Önerilen sistem için geliştirilen uygulamada temel olarak iki yapılandırma bulunmaktadır. Bu yapılandırmalar güvenlik ve veri tabanı erişim yapılandırmalarıdır. Bu yapılandırmalar için web servis sunucusuna yapılacak isteklerin hangi parametreleri taşıyıp taşımayacağı, yapılan istek sonucunda cevapta taşınan nesnenin belirlenmesi ve yapılan isteğin HTTP metodunun belirlenmesi gerekir. Örnek olarak @RequestMapping ile bir web servis bağlantısı oluşturulduğunda web servis bağlantısının içerisinde bir adrsin tanımlanması gerekir. @RequestMapping anotasyounun içinde "/bayiayrintikayi" adresi tanımlanmıştır. Web servis metodu ise POST olarak belirlenmiştir. Cevap olarak gönderilecek "return" değerinin ise JSON tipinde olması gerektiği belirtilmiştir. Gelen istek gövdesinde parametreler bulunmaktadır. Sunucuda bulunan web servisler istasyon girişi, istasyon kaydı, istasyon ayrıntıları oluşturma, istasyon ayrıntıları listeleme, kullanıcı yorum ekleme, kullanıcı yorum listeleme ve istasyon ayarları olarak belirlenmiştir.

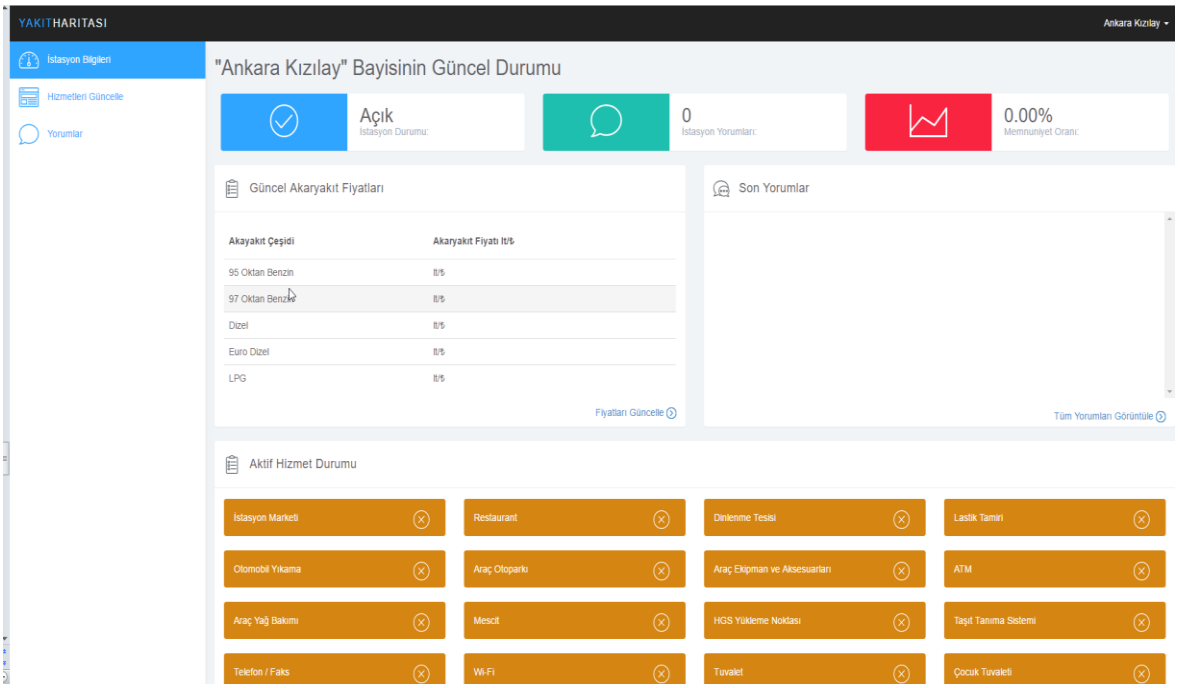
3.2. Mobil Uygulamanın Geliştirilmesi (Development of Mobile Application)

Geliştirilen mobil uygulamanın geliştirilmesinden önce fonksiyonel ihtiyaçlar analiz edilmiştir. Bu ihtiyaçlar yakındaki istasyonları bulma, yakındaki istasyonları listeleme ve sıralama, konuma göre istasyon bulma, seçilen mesafeye göre istasyon bulma, istasyonların akaryakıt fiyatlarını öğrenme ve istasyonların hizmetlerini öğrenme olarak belirlenmiştir.

Belirtilen fonksiyonel ihtiyaçlara göre geliştirilen mobil uygulama Android Studio kullanılarak geliştirilmiştir [17]. Geliştirilen uygulamada kullanıcılar web servis sunucusuna isteklerde bulunmaktadır. Web servisler kendisine gelen istekleri değerlendirmekte ve yapılan isteğe uygun cevabı hazırlayarak mobil uygulamaya göndermektedir. Kullanıcının mobil uygulama üzerinden gerçekleştirdiği isteklerde ve isteklere verilen cevaplarda kullanıcı – mobil uygulama – sunucu arasında çift yönlü bir iletişim yapısı kullanılmıştır.

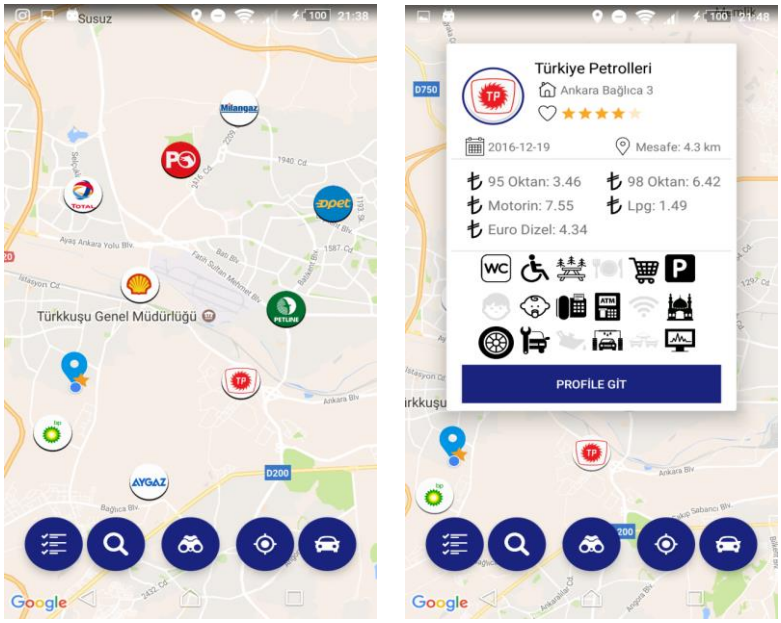
4. TEST İŞLEMLERİ (TEST PROCEDURES)

Önerilen sistemde akaryakıt istasyonlarının akaryakıt fiyatları ve sunduğu hizmetler ile ilgili veriler bir web istemci arayüzü yardımıyla uygulamaya girilmektedir. Akaryakıt istasyonu yetkili personeli uygulama üzerinden kayıt olup giriş yaptıktan sonra şekil 3'te görülen istasyon ekranı tarafından karşılanmaktadır.



Şekil 3. *Akaryakıt istasyonu bilgi girişi*

Şekil 3'te verilen bilgi girişi ekranı üzerinde bulunan seçenekler sayesinde istasyon ayrıntıları sisteme girilmektedir. Güncelle bölümünden akaryakıt istasyonunun sunduğu hizmetler veya akaryakıt fiyatları güncellenebilmektedir. Aynı şekilde akaryakıt tüketicileri de web arayüzü yardımıyla uygulamaya erişebilmekte ve akaryakıt istasyonunun ayrıntılarını görebilmektedir. Ayrıca kullanıcılar harita üzerinden belirlediği konuma en yakın akaryakıt istasyonunu görebilmekte ve belirlediği konumdan akaryakıt istasyonuna yol tarifi alabilmektedir.



Şekil 4. *(a) Harita ekranı, (b) istasyon bilgi diyaloğu ekranı*

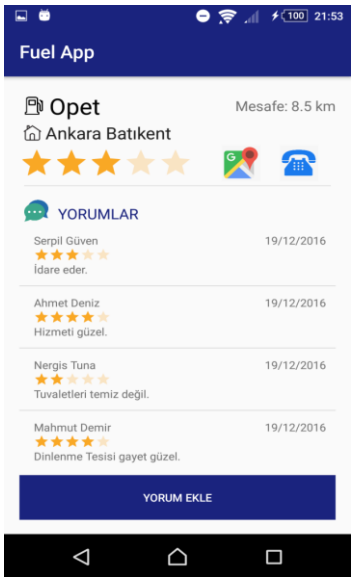
Mobil uygulamanın ana ekranı ise haritanın bulunduğu ekrandır. Ekranın alt bölümünde menü bulunmaktadır. Bu menü sırasıyla istasyon listeleme ve sıralama, konum arama, uzaklık mesafesi

belirleme ve mevcut konuma gitme görevlerini yerine getirmektedir. Harita ekranı üzerindeki herhangi bir akaryakıt istasyonu seçildiğinde bilgi diyalogu açılmaktadır. Bu diyalogda bulunan istasyon "profiline git" seçeneği ile yeni bir ekran açılmaktadır. Bu ekranda istasyon hakkında yorum ve puan ekleme, yol tarifi alma ve telefon etme vb. işlemler gerçekleştirilebilmektedir.

Mobil uygulamanın kullanabilmesi için GPS aktif ise, harita otomatik olarak kullanıcının konumuna odaklanmaktadır. Uzaklık mesafesi varsayılan olarak 10 kilometre olarak ayarlanmıştır. Uygulama ilk başlatıldığında kullanıcıya 10 kilometre yakınlıktaki akaryakıt istasyonları harita üzerinde gösterilmektedir. Harita ekranının ekran görüntüsü şekil 4(a)'da istasyon bilgi diyaloğu ekranı ise şekil 4(b)'de verilmiştir.

Akaryakıt istasyonları bayi logosu ile haritada gösterilmiştir. Akaryakıt istasyonunun bilgilerini görüntülemek için bilgi alınacak istasyonun logosunun seçilmesi yeterlidir. Seçilen istasyonun bilgileri bir diyalog ile kullanıcıya sunulmuştur. Bilgi diyalogunda akaryakıt istasyonunun akaryakıt fiyatları, istasyon hizmetleri hakkında bilgi alınmaktadır. Diyalogda bulunan tarih ise akaryakıt fiyatlarının en son güncellendiği tarihi göstermektedir. Bir istasyonun sunduğu hizmetler bilgi diyalogunda koyu renkle gösterilmiştir. Eğer bir hizmet istasyonda bulunmuyor ise özelliğin logosu açık gri tonundan oluşmaktadır.

Önerilen sistemin diğer bir fonksiyonu da kullanıcılardan geri bildirim almaktır. Bu işlem için yorumlar ekranı hazırlanmıştır. Yorumlar ekranı kullanıcıların akaryakıt istasyonu hakkındaki fikirlerinin, önerilerinin, şikâyetlerinin, beğenilerinin, tavsiyelerinin yorum ve puan olarak paylaşmasına imkan vermektedir. Şekil 5'te yorumlar ekranının ekran görüntüsü verilmiştir.



Şekil 5. *Yorumlar ekranı*

Şekil 5'te verilen ekranın üst bölümünde akaryakıt istasyonunun adı, şube bilgisi, yorum derecelendirme bilgisi ve istasyonun kullanıcıya uzaklığı verilmiştir. Bu bölümün altında ise kullanıcıların yaptığı yorumlar bulunmaktadır. Kullanıcılar yoruma göre akaryakıt istasyonunu seçebilmekte kendi mesafesine göre uzaklığını göre bilmektedir. Ayrıca mesafe alanına göre, yorum puanına göre ve fiyat avantajına göre akaryakıt istasyonlarının listelenmesi gerçekleştirilmiştir.

5. SONUÇ VE ÖNERİLER (CONCLUSION AND RECOMMENDATIONS)

Geliştirilen uygulama sayesinde akaryakıt istasyonlarının sunduğu hizmetlerin akaryakıt tüketicilerine bildirilmesi ve daha uygun fiyatlı akaryakıt satışı gerçekleştiren istasyonlar hakkında tüketicilerin bilgi sahibi olmaları sağlanmıştır. Daha kaliteli hizmet ve uygun fiyat avantajı sağlayan akaryakıt istasyonunun en yakın hangi konumda yer aldığı harita ekranı üzerinden kullanıcıya sunulmuştur. Akaryakıt

tüketicilerinin istasyonlar hakkında yorum yapmaları ve puan vermeleri sağlanmıştır. Bu sayede akaryakıt istasyonlarının fiyat avantajı ve hizmet kalitesi net olarak ortaya konulmuştur. Akaryakıt istasyonlarının kendilerini yenilemesi, artan rekabet şartlarında kendilerine iyi bir yer edinebilmesi için gerçek kullanıcıların yaptığı değerlendirmeler akaryakıt istasyonları ve tüketicilere sunulmuştur. Böylelikle akaryakıt istasyonlarının hizmet eksiklikleri, hizmet kalitesi ve diğer akaryakıt istasyonlarına göre fiyat avantajı belirlenmiştir.

Önerilen sistemin temelini web servis yapısı oluşturmaktadır. Web servisler ile arka plan iş kontrol yapıları hazırlanmıştır. Sistem mobil uygulama ve web arayüzüne sahip bir web istemci ile kullanıcıya sunulmuştur. Geliştirilen web servisler REST tabanlıdır. Web arayüzü AngularJS ve HTML kullanılarak geliştirilmiştir. Oluşturulan bu sisteme kullanıcıların daha rahat erişebilmeleri ve mobil sistemlerin imkânlarından faydalanabilmek için bir mobil uygulama geliştirilmiştir. Geliştirilen mobil uygulama internet alt yapısı ile REST web servislerine, web serviler ise gerektiğinde veri tabanına erişim sağlamaktadır. Veri yapısı olarak JSON veri formatı tercih edilmiştir. Böylelikle daha verimli ve hızlı veri iletişimi sağlanmıştır.

Önerilen sistem web servisler, web arayüzü, mobil uygulama ve NoSQL veri tabanının birlikte kullanılması ile geliştirilmiştir. Kullanım senaryosu olarak birden fazla kullanıcı ve konum içeren akaryakıt istasyonları seçilmiştir. Böylelikle çoklu okuma/yazma işlemlerinde sorunsuz çalışabilecek bir sistem geliştirilmiştir. Geliştirilen sistemin kullanım testlerinde web servislerin mobil ve web platformlarda sorunsuz hizmet verdiği görülmüştür. Sistemin platformdan bağımsız çalışması ile farklı uygulamalara entegre edilebilmesi sağlanmıştır. Web servislerin güvenliği Spring Boot çerçevesi tercih edilerek Spring güvenlik bileşenlerinden yararlanılarak sağlanmıştır. Önerilen sistem ile sadece akaryakıt istasyonlarına hizmet verecek bir sistem değil farklı kullanım alanlarında da kullanılacak bir mimari yapı sunulmuştur.

TEŞEKKÜR (ACKNOWLEDGMENTS)

Bu çalışmada sunulan web servislerin geliştirilmesindeki katkısından dolayı Erdem AYDEMİR'e ve mobil uygulamanın geliştirilmesindeki katkılarından dolayı Merve YOLCU'ya teşekkür ederiz.

KAYNAKLAR (REFERENCES)

[1] Avcılar, M.Y., Açar., M.F.T. (2017). Tüketicilerin akaryakıt istasyonu tercihini etkileyen faktörlerin analitik hiyerarşi prosesi yöntemi ile tespitine yönelik Ankara ilince bir araştırma. Hitit Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 10(2), 969-992.

[2] Oyucu, S., Polat, H. (2017). M2M ve IoT platformları üzerinde prototip uygulama geliştirme. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 9(2), 11-20.

[3] Gültekin, G. (2014). Akıllı Konum Tabanlı Mobil Alışveriş Android Uygulaması, İstanbul Kemerburgaz Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik ve Bilgisayar Mühendisliği Anabilim Dalı.

[4] İnternet: Patrick, D. H, "GasBuddyhttps", www.gasbuddy.com, Son Erişim Tarihi: 16.04.2021.

[5] İnternet: Wikicamps, "Fuel Map", http://www.microsoft.com/en-au/p/fuel-map-australia/, Son Erişim Tarihi: 16.04.2021.

[6] Ramanathan, R., Korte, T. (2014). Software service architecture to access weather data using RESTful web services. Fifth International Conference on Computing, Communications and Networking Technologies Hefei, 1-8.

[7] Yao, M., Ruixue, J. L., Chen, F. G., Zhao, J. (2017). REST based integrated efficient drawing method for reservoir geological profile. 6th International Conference on Measurement, Instrumentation and Automation, 1951-685.

[8] Malik, S., Kim. D. (2017). A comparison of RESTful vs. SOAP web services in actuator networks. International Conference on Ubiquitous and Future Networks, 753-755.

[9] Oliveira, R.R., Vieira, R.V., Sanchez, J. C., Estrella, R., Brusamolin, V. (2013). Comparative evaluation of the maintainability of RESTful and SOAP-WSDL web services, IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, 40-49.

[10] Li, L., Chou, W. (2015). Designing large scale REST APIs based on REST chart. IEEE International Conference on Web Services, 631-638.

[11] Pandey, H., Rastogi, H., Gupta Anuja, C. ()2017. Web-based network management system implemented using Hibernate, JBoss and spring framework. International Conference on Telecommunication and Networks, 1-5.

[12] Ramos, M., Valente, M.T., Terra, R. (2018). AngularJS Performance: A survey study. IEEE Software, 35(2),72-79.

[13] Mearaj, I., Maheshwari, P., Kaur, M. J. (2018). Data conversion from traditional relational database to MongoDB using XAMPP and NoSQL. HCT Information Technology Trends, 94-98.

[14] Polat, H., Oyucu, S. (2019). Heterojen Medikal IoT verilerinin depolanmasında ilişkisel olmayan veritabanına dayalı bir yaklaşım. Politeknik Dergisi, 22(4), 989-998.

[15] İnternet: Geleceği Yazanlar, https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-301/androidde-konum-servisleri, Son Erişim Tarihi: 16.03.2020.

[16] Li, H., Zhijian, L. (2010). The study and implementation of mobile GPS navigation system based on Google Maps. International Conference on Computer and Information Application, 87-90.

[17] Allison, L., Fuad, M.M. (2016). Inter-App communication between android apps developed in app-ınventor and android studio. IEEE/ACM International Conference on Mobile Software Engineering and Systems, 17-18.