



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



TOPLU TAŞIMADA BEACON DESTEKLİ
TEMASSIZ GEÇİŞ VE DİNAMİK
ÜCRETLENDİRME MODELİ

Ali Enez CAN

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YAZILIM MÜHENDİSLİĞİ PROGRAMI

Kasım 2025

KONYA

Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Ali Enez CAN tarafından hazırlanan ‘‘Toplu Tařımada Beacon Destekli Temassız Geçiř Ve Dinamik Ücretlendirme Modeli’’ adlı tez çalıřması 07/11/2025 tarihinde ařağıdaki jüri tarafından oy birliğı ile Konya Teknik Üniversitesi Lisansüstü Eđitim Enstitüsü Bilgisayar Mühendisliğı Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiřtir.

Jüri Üyeleri

İmza

Başkan

Doç. Dr. Murat KARAKOYUN
(Necmettin Erbakan Üniversitesi)

Danışman

Doç. Dr. İsmail KOÇ
(Konya Teknik Üniversitesi)

Üye

Dr.Öğr Üyesi Vahit TONGUR
(Konya Teknik Üniversitesi)

Yukarıdaki sonucu onaylım.

Prof. Dr. Mevlüt UYAN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Ali Enez CAN

Tarih:

ÖZET

YÜKSEK LİSANS TEZİ

TOPLU TAŞIMADA BEACON DESTEKLİ TEMASSIZ GEÇİŞ VE DİNAMİK ÜCRETLENDİRME MODELİ

Ali Enez CAN

KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YAZILIM MÜHENDİSLİĞİ PROGRAMI

Danışman: Doç. Dr. İsmail KOÇ

2025, 72 Sayfa

Jüri

Doç. Dr. İsmail KOÇ
Dr. Öğr. Üyesi Vahit TONGUR
Doç. Dr. Murat KARAKOYUN

Bu çalışma, şehir içi toplu taşımada Bluetooth Low Energy (BLE) beacon teknolojisini kullanarak temassız biletleme ve dinamik fiyatlandırma odaklı yenilikçi bir sistem önermektedir. Çalışmanın temel amacı, manuel kullanıcı etkileşimine dayalı kart tabanlı sistemlerin yerine, tamamen otomatik ve arka planda çalışan bir yapının geliştirilmesidir. Önerilen sistemde, otobüs ve duraklara yerleştirilen BLE beacon cihazları, mobil uygulama ile etkileşime geçerek yolcunun biniş ve iniş noktalarını yüksek doğrulukla tespit etmektedir. Mobil uygulama, beacon sinyallerini pasif biçimde algılayarak GPS verileriyle biniş konumunu doğrular; durak beacon'ları ise iniş noktalarını belirleyerek mesafe bazlı ücretlendirme modelinin uygulanmasını sağlar. Fiziksel kart veya QR kod kullanımının ortadan kalkması, yolcu deneyimini hızlandırmakta ve işlem sürelerini azaltmaktadır. Bununla birlikte, sistem anonim ve güvenli veri toplayarak güzergâh optimizasyonu ve ulaşım planlaması için önemli katkılar sunmaktadır. Geliştirilen mimari, beacon donanımı, Flutter tabanlı bir mobil uygulama ve REST, SOAP ile gRPC protokollerini destekleyen sunucu altyapısından oluşmaktadır. Bu protokoller performans, gecikme süresi, ağ trafiği ve kaynak kullanımı açısından kıyaslanmıştır. Sonuçlar, gRPC'nin CPU, bellek ve ağ verimliliği bakımından ortalama %25-30 daha iyi performans sergilediğini; REST'in yüksek işlem kapasitesi, SOAP'ın ise güvenlik açısından avantajlı olduğunu göstermiştir. Ayrıca, sistem saha testlerinde yolcu başına ortalama 2 saniye zaman tasarrufu sağlamış ve kuyruk uzunluklarını %20 azaltmıştır. Bu sonuçlar, BLE tabanlı otomatik ücretlendirme modellerinin hem kullanıcı deneyimi hem de operasyonel verimlilik açısından önemli potansiyele sahip olduğunu ortaya koymaktadır.

Anahtar Kelimeler: BLE Beacon Cihazı, Dinamik Ücretlendirme, Şehir İçi Ulaşım, Temassız Bilet, gRPC, REST API, SOAP API

ABSTRACT

MS THESIS

**A CONTACTLESS TICKETING AND DYNAMIC PRICING MODEL FOR
URBAN TRANSPORTATION USING BLE BEACONS**

Ali Enez CAN

**KONYA TECHNICAL UNIVERSITY
INSTITUTE OF GRADUATE STUDIES
DEPARTMENT OF SOFTWARE ENGINEERING**

Advisor: Assoc. Prof. Dr. İsmail KOÇ

2025, 72 Pages

Jury

Assoc. Prof. Dr. İsmail KOÇ

Asst. Prof. Dr. Vahit TONGUR

Assoc. Prof. Dr. Murat KARAKOYUN

This study presents an innovative contactless ticketing and dynamic pricing framework for urban public transportation using Bluetooth Low Energy (BLE) beacon technology. The main goal is to replace traditional card-based fare collection systems with a fully automated architecture requiring minimal user interaction. In the proposed design, BLE beacons installed on buses and stops interact with passengers' mobile devices to automatically determine boarding and disembarkation points with high precision. The mobile application passively detects beacon signals and confirms boarding via GPS data, while stop beacons identify exit locations, enabling an accurate distance-based fare calculation. Eliminating the need for physical cards or QR validation improves both transaction speed and user experience. At the same time, the secure and anonymized collection of passenger behavior data enhances route optimization and urban mobility planning. The system architecture combines beacon hardware, a Flutter-based mobile application, and backend services supporting REST, SOAP, and gRPC protocols. These communication methods were comparatively evaluated across CPU load, memory usage, network consumption, and latency performance metrics. Results indicate that gRPC achieved approximately 25–30% better overall efficiency than REST and SOAP, with REST providing higher throughput and SOAP offering superior reliability and data integrity. Field tests showed an average boarding time reduction of about two seconds per passenger and a 20% decrease in queue lengths. The findings confirm that BLE-based automated fare collection can significantly enhance both user convenience and operational efficiency in smart public transportation systems.

Keywords: BLE Beacon Device, Contactless Ticketing, Dynamic Pricing, Urban Transportation, gRPC, REST API, SOAP API

ÖNSÖZ

Bu tez çalışması, uzun bir araştırma süreci, yoğun emek ve sabırlı bir gayretin sonucunda ortaya çıkmıştır. Akademik bir çalışmanın yalnızca bireysel çabanın değil; aynı zamanda çevreden alınan desteklerin, paylaşılan fikirlerin ve gösterilen anlayışın da bir ürünü olduğunun bilincindeyim. Bu nedenle, bu yolculuk boyunca bana katkı sağlayan, destek olan ve motive eden herkese en içten teşekkürlerimi sunmak isterim.

Her şeyden önce, araştırmamın her aşamasında değerli bilgi ve tecrübesiyle bana rehberlik eden, yolumu aydınlatan danışmanım Doç. Dr. İsmail Koç'a şükranlarımı ifade etmek isterim. Kendisinin bilimsel yaklaşımı, yapıcı eleştirileri ve sabırlı yönlendirmeleri, bu çalışmanın olgunlaşmasında belirleyici bir rol oynamıştır. Onun teşviki ve desteği, zorluklar karşısında motivasyonumu diri tutmamı sağlamıştır.

Bu süreçte, bana her koşulda en büyük manevi desteği sağlayan ve aynı zamanda bilgisayar mühendisliği alanındaki bilgi ve deneyimiyle teknik açıdan da yoluma ışık tutan sevgili eşim Nur CAN'a özel bir teşekkür borçluyum. Onun sabrı, anlayışı, teşviki ve teknik katkıları, bu çalışmanın hem ruhen hem de bilimsel anlamda şekillenmesinde vazgeçilmez bir rol oynamıştır.

Araştırma süresince görüş ve katkılarıyla bana yardımcı olan değerli iş arkadaşlarım Yiğit HARUN ve Hüseyin KÜÇÜKŞAHİN'e de teşekkür etmeyi bir borç bilirim. Fikir alışverişleri, samimi destekleri ve dayanışmaları, çalışmama hem bilimsel hem de manevi anlamda katkıda bulunmuştur.

Son olarak, bu çalışmayı yürütmem için gerekli akademik ortamı ve kaynakları sağlayan, bilimsel gelişimime büyük katkı sunan Konya Teknik Üniversitesi'ne teşekkür ederim.

Bu tez, yalnızca benim değil; bana destek veren ve yanımda olan tüm bu değerli insanların ortak emeğinin bir yansımasıdır.

Ali Enez CAN
KONYA-2025

İÇİNDEKİLER

ÖZET	i
ABSTRACT.....	ii
ÖNSÖZ	iii
İÇİNDEKİLER	iv
SİMGELER VE KISALTMALAR	vii
1. GİRİŞ	1
1.1 Çalışmanın Arka Planı ve Önemi	1
1.2 Problem Tanımı	2
1.3 Çalışmanın Amacı.....	3
1.4 Araştırma Soruları.....	4
1.5 Çalışmanın Kapsamı ve Sınırlılıkları.....	4
2. KAYNAK ARAŞTIRMASI	5
2.1. Toplu Taşımada Ücretlendirme Yöntemleri	5
2.1.1 Sabit Ücretlendirme (Flat Fare)	5
2.1.2 Bölge Bazlı Ücretlendirme (Zone-Based Fare)	5
2.1.3 Mesafe Bazlı Ücretlendirme (Distance-Based Fare)	5
2.1.4 Süre Bazlı ve Zaman Pencere Ücretlendirme.....	6
2.1.5 Tavan–Taban (Peak/Off-Peak) ve Dinamik Fiyatlandırma	6
2.1.6 Dinamik/Esnek Ücretlendirme	6
2.1.7 Sosyal Politika Araçları	6
2.2. Beacon Nesne Kullanım Alanları	7
2.2.1 Perakende Sektörü	7
2.2.2 Kültürel Alanlar ve Müzeler	7
2.2.3 Sağlık Hizmetleri	7
2.2.4 Eğitim Kampüsleri	8
2.2.5 Ulaşım Sistemleri.....	8
2.2.6 Genel Değerlendirme	8
2.3. Toplu Taşımada Biletleme Sistemlerinin Evrimi	9
2.3 BLE'nin Ulaşım Sistemlerindeki Kullanımı.....	10
2.4 Veri Odaklı Ulaşım Yönetimi ve Dinamik Ücretlendirme	11
2.4.1 Kavramsal Çerçeve: “Veri Odaklı” Ne Demektir?	11
2.4.3 Veri Boru Hattı ve Analitik Mimarisi.....	13
3. MATERYAL VE YÖNTEM.....	22
3.1 Sistem Tasarımı	22
3.1.1 Beacon Cihazları	22
3.1.1.4 Sinyal ve Konum Belirleme Yöntemleri	26
3.1.1.5 BLE'nin Avantajları ve Kısıtları.....	26
3.1.2 Mikrodenetleyici ve Güvenlik Mekanizmaları	27
3.1.3 Mobil Uygulama	28

3.1.4 Sunucu Tarafı Servisleri	28
3.1.5 Program Tasarımı	29
3.2 Deneysel Çalışmalar	31
3.2.1 Deneysel Ortam ve Kısıtlar.....	32
3.2.2 Test Senaryoları	33
3.2.2.1 Kullanıcı Yüğü Senaryoları	33
3.2.2.2 İşlem Türleri	33
3.2.2.3 Ağ Koşulları.....	34
3.2.3 Performans Deęerlendirme Metrikleri.....	34
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	35
4.1 CPU Kullanımı Sonuçları ve Tartışma	35
4.1.1 Genel Bulgular	35
4.1.2 Grafiklerin Deęerlendirilmesi.....	35
4.1.3 Protokollerin Karşılaştırmalı Analizi.....	37
4.1.4 Tartışma	37
4.2 Bellek Kullanımı Sonuçları ve Tartışma	38
4.2.1 Genel Bulgular	38
4.2.2 Grafiklerin İncelenmesi	39
4.2.3 Karşılaştırmalı Analiz	41
4.2.4 Genel Deęerlendirme	41
4.3 Network Trafięi Sonuçları ve Tartışma	42
4.3.1 Genel Bulgular	42
4.3.2 Grafiklerin İncelenmesi	42
4.3.3 Karşılaştırmalı Analiz	44
4.3.4 Genel Deęerlendirme	44
4.4 Sistem İşlem Kapasitesi (Throughput'u) Sonuçları ve Tartışma.....	45
4.4.1 Genel Bulgular	45
4.4.2 Grafiklerin İncelenmesi	46
4.4.3 Protokollerin Karşılaştırmalı Analizi.....	47
4.4.5 Genel Deęerlendirme	48
4.5 Network Throughput (Byte Throughput) Sonuçları ve Tartışma	48
4.5.1 Genel Bulgular	48
4.5.2 Grafiklerin İncelenmesi	49
4.5.3 Karşılaştırmalı Analiz	50
4.5.4 Genel Deęerlendirme	51
4.6 Yanıt Süresi (Latency) Sonuçları ve Tartışma.....	51
4.6.1 Genel Bulgular	51
4.6.2 Grafiklerin İncelenmesi	52
4.6.3 Karşılaştırmalı Analiz	53
4.6.4 Genel Deęerlendirme	54
4.7 Verimlilik: Throughput ve Latency İlişkisi	54
4.7.1 Genel Bulgular	54
4.7.2 Grafiklerin İncelenmesi	55
4.7.3 Karşılaştırmalı Analiz	57
4.7.4 Genel Deęerlendirme	58
5. SONUÇLAR VE ÖNERİLER.....	59
5.1 Çalışmanın Genel Bulguları.....	59

5.2 Akademik Katkılar.....	60
5.3 Pratik Katkılar.....	61
5.4 Çalışmanın Kısıtları.....	61
5.5 Gelecek Çalışmalar İçin Öneriler.....	62
5.6 Genel Değerlendirme.....	62
KAYNAKLAR.....	63
EKLER.....	66



SİMGELER VE KISALTMALAR

Simgeler

dBm	: Desibel-milliwatt (sinyal gücü ölçü birimi)
MS	: milisaniye (zaman ölçü birimi)
MB	: Megabayt (bellek ölçü birimi)
KB	: Kilobayt (veri ölçü birimi)
KB/s	: Kilobayt/saniye (ağ veri aktarım hızı)
RPS	: Saniyede İşlenen İstek Sayısı (Requests Per Second)
TPS	: Saniyede İşlenen İşlem Sayısı (Transactions Per Second)

Kısaltmalar

API	: Uygulama Programlama Arayüzü (Application Programming Interface)
BLE	: Düşük Enerjili Bluetooth (Bluetooth Low Energy)
CPU	: Merkezi İşlem Birimi (Central Processing Unit)
CSA	: Bağlantı Taramalı Algoritma (Connection Scan Algorithm)
CVSS	: Ortak Zafiyet Puanlama Sistemi (Common Vulnerability Scoring System)
GDPR	: Genel Veri Koruma Tüzüğü (General Data Protection Regulation)
GPS	: Küresel Konumlama Sistemi (Global Positioning System)
gRPC	: Google Uzak Prosedür Çağrısı (Google Remote Procedure Call)
HTTP	: Hiper Metin Aktarım Protokolü (Hypertext Transfer Protocol)
HTTPS	: Güvenli Hiper Metin Aktarım Protokolü (Hypertext Transfer Protocol Secure)
IoT	: Nesnelerin İnterneti (Internet of Things)
JSON	: JavaScript Nesne Gösterimi (JavaScript Object Notation)
KPI	: Anahtar Performans Göstergesi (Key Performance Indicator)
KVKK	: Kişisel Verilerin Korunması Kanunu (Türkiye)
LAN	: Yerel Alan Ağı (Local Area Network)
MITM	: Aradaki Adam Saldırısı (Man-in-the-Middle)
NFC	: Yakın Alan İletişimi (Near Field Communication)
OD	: Başlangıç–Varış (Origin–Destination)
OLTP	: Çevrimiçi İşlem İşleme (Online Transaction Processing)
QoS	: Hizmet Kalitesi (Quality of Service)
RAM	: Rastgele Erişimli Bellek (Random Access Memory)
REST	: Temsili Durum Transferi (Representational State Transfer)
RSSI	: Alınan Sinyal Gücü Göstergesi (Received Signal Strength Indicator)
SDK	: Yazılım Geliştirme Kiti (Software Development Kit)
SOAP	: Basit Nesne Erişim Protokolü (Simple Object Access Protocol)
SQL	: Yapılandırılmış Sorgu Dili (Structured Query Language)
SSID	: Servis Seti Tanımlayıcısı (Service Set Identifier)
TCP	: Aktarım Denetim Protokolü (Transmission Control Protocol)
TLS	: Taşıma Katmanı Güvenliği (Transport Layer Security)
UDP	: Kullanıcı Datagram Protokolü (User Datagram Protocol)
UUID	: Evrensel Benzersiz Tanımlayıcı (Universally Unique Identifier)
XML	: Genişletilebilir Biçimlendirme Dili (Extensible Markup Language)

1. GİRİŞ

1.1 Çalışmanın Arka Planı ve Önemi

Kent içi toplu taşıma sistemleri, modern şehirlerin yaşam damarlarını oluşturmaktadır. Nüfus yoğunluğu arttıkça bireysel araç kullanımının getirdiği trafik yoğunluğu, hava kirliliği ve park yeri sorunları, toplu taşımayı yalnızca bir tercih değil, zorunlu bir çözüm haline getirmektedir. Bu nedenle, toplu taşıma hizmetlerinin hızlı, verimli, düşük maliyetli ve kullanıcı dostu olması, yalnızca bireylerin yaşam kalitesini değil, aynı zamanda şehirlerin sürdürülebilirliğini de doğrudan etkilemektedir.

Toplu taşıma deneyiminin temel bileşenlerinden biri, biletleme ve ücretlendirme sistemleridir. Geleneksel kart tabanlı modeller, yıllar boyunca şehirlerde standart bir çözüm olarak uygulanmıştır. Ancak bu sistemlerin, günümüzde artan yolcu sayısı ve dijital dönüşüm ihtiyacı karşısında önemli eksiklikleri bulunmaktadır. Yolcuların kart taşıma zorunluluğu, bakiye yükleme süreçlerinin zahmetli olması, kartların kaybolması ya da bozulması gibi problemler, kullanıcı memnuniyetini azaltmakta ve günlük yolculukların akışını zorlaştırmaktadır. Bunun ötesinde, validatör cihazlarında kart okutma sırasında yaşanan yoğunluklar, özellikle işe gidiş-geliş saatlerinde ciddi zaman kayıplarına yol açmaktadır. Her bir yolcunun birkaç saniye süren kart okutma işlemi, onlarca kişinin sırada beklediği bir otobüs durağında veya metro girişinde dakikalar süren kuyrukların oluşmasına neden olmaktadır. Bu da hem bireysel zaman kaybı hem de seferlerin verimliliğinde düşüş anlamına gelmektedir.

Bu sorunlar, pandemi döneminde daha da görünür hale gelmiştir. COVID-19 gibi solunum yolu hastalıkları, doğrudan temas veya yakın mesafe etkileşimleri yoluyla kolaylıkla bulaşabilmektedir. Dünya Sağlık Örgütü (DSÖ) ve sağlık otoriteleri, özellikle toplu taşıma alanlarında sosyal mesafenin korunmasının ve temaslı yüzeylerin azaltılmasının kritik olduğunu defalarca vurgulamıştır. Ancak kart tabanlı sistemler, yolcuları tek bir cihaz üzerinde işlem yapmaya zorladığı için yüksek temas yoğunluğu yaratmış, bu da hastalık bulaşma riskini artırmıştır.

Pandemi süresince yapılan araştırmalar, toplu taşıma araçlarında temas edilen yüzeylerin (örneğin validatörler, turnikeler, kapı kolları) yüksek risk taşıdığını ortaya koymuştur. Virüslerin plastik ve metal yüzeylerde saatlerce, hatta bazı koşullarda günlerce canlı kalabildiği bilinmektedir. Bu nedenle yolcuların aynı cihazı tekrar tekrar kullanması, olası bir bulaş zincirini kolaylaştırmıştır. Kartlı sistemlerin zorunlu kıldığı “her yolcunun aynı noktada fiziksel işlem yapması” yaklaşımı, bu dönemde hem

yolcuların güvenlik kaygılarını artırmış hem de kamu sağlığı açısından riskli bir yapı oluşturmuştur.

Bunun yanında, pandeminin getirdiği hijyen kaygıları dışında, sosyal mesafenin korunmasını imkânsız hale getiren sıra bekleme sorunu da ön plana çıkmıştır. Özellikle işe gidiş-geliş saatlerinde otobüs duraklarında ve metro girişlerinde oluşan kalabalık kuyruklar, yolcular arasında teması kaçınılmaz hale getirmiştir. Bazen yüzlerce kişinin aynı anda kart okutmaya çalıştığı bu ortamlar, bulaşıcı hastalıkların yayılımı için ideal koşullar yaratmıştır. Bu durum yalnızca COVID-19 özelinde değil, mevsimsel grip, influenza gibi solunum yolu hastalıklarının yayılımında da önemli bir etkidir. Dolayısıyla, temas gerektirmeyen, hızlı ve akıcı geçiş sistemlerine olan ihtiyaç, pandemi ile birlikte yalnızca teknolojik değil, aynı zamanda sağlık ve güvenlik temelli bir zorunluluk haline gelmiştir.

Öte yandan, şehir içi ulaşımın yalnızca bireysel yolcuları değil, aynı zamanda makro ölçekli ekonomik ve sosyal yapıyı da etkilediği unutulmamalıdır. Her yolcunun otobüse binerken geçirdiği fazladan 5–10 saniyelik zaman kaybı, yüzlerce seferin yapıldığı bir şehirde gün sonunda saatlere varan toplam gecikmelere dönüşmektedir. Bu da sadece yolcuların bireysel zaman kaybına değil, belediyelerin ve ulaşım otoritelerinin operasyonel verimlilik kayıplarına, yakıt tüketiminin artmasına ve dolaylı olarak karbon salınımının yükselmesine sebep olmaktadır.

Sonuç olarak, kullanıcı dostu, hızlı, güvenli ve temassız çözümler toplu taşıma alanında artık ertelenemez bir ihtiyaç haline gelmiştir. BLE beacon teknolojisi, yolcunun hiçbir manuel işlem yapmasına gerek kalmadan arka planda çalışabilmesi sayesinde, bu sorunlara doğrudan çözüm sunabilecek güçlü bir aday olarak öne çıkmaktadır. Bu nedenle çalışmanın önemi, yalnızca bir teknolojik yenilik önermekle sınırlı değildir; aynı zamanda toplumsal sağlık, zaman yönetimi, operasyonel verimlilik ve sürdürülebilir şehircilik açısından da kritik katkılar içermektedir.

1.2 Problem Tanımı

Mevcut sistemlerde yolcular, toplu taşımadan faydalanabilmek için fiziksel bir ulaşım kartına sahip olmakta ve bu kartı otobüs veya metro girişlerinde yer alan validatör cihazlarına okutmak zorundadır. Bu yaklaşım, çeşitli nedenlerden dolayı sorunludur:

- **Kullanıcı Deneyimi Problemleri:** Kart taşıma zorunluluğu, bakiye yetersizliği, kartın fiziksel hasar görmesi veya cihazların arızalanması gibi durumlar yolculuk sürecini kesintiye uğratmaktadır.
- **Zaman Kaybı ve Verimlilik Sorunları:** Özellikle yoğun saatlerde her yolcunun kart okutması sırasında oluşan gecikmeler, otobüslerin duraklarda daha uzun süre kalmasına ve seferlerin aksamasına yol açmaktadır.
- **Hijyen Riskleri:** Temas gerektiren cihazların ortak kullanımı, özellikle pandemi gibi olağanüstü dönemlerde sağlık risklerini artırmaktadır.
- **Veri Eksikliği:** Geleneksel kartlı sistemlerde, yolcunun yalnızca biniş noktası kaydedilebilmekte, iniş noktası çoğu zaman bilinmemektedir. Bu durum, güzergâh planlaması ve sefer optimizasyonu gibi konularda veri yetersizliğine neden olmaktadır.

Alternatif çözümler arasında yer alan NFC veya QR kod tabanlı mobil biletleme yöntemleri, kısmen temas gereksinimini ortadan kaldırırsa da hâlâ yolcudan manuel etkileşim beklemektedir. Örneğin yolcunun mobil cihazını validatöre yaklaştırması veya QR kod okutması gerekmektedir. Bu nedenle, tam anlamıyla otomatik ve arka planda çalışan bir biletleme çözümüne duyulan ihtiyaç halen devam etmektedir.

1.3 Çalışmanın Amacı

Bu tez çalışması, toplu taşımada BLE beacon tabanlı, kullanıcıdan hiçbir manuel işlem beklemeyen, tamamen otomatik ve temassız bir biletleme sistemi tasarlamayı ve bu sistemin performansını farklı protokoller üzerinden test ederek değerlendirmeyi amaçlamaktadır.

Önerilen sistemin temel özellikleri şunlardır:

- **Tam Otomasyon:** Yolcunun otobüse binmesiyle birlikte mobil cihazı, araç içinde yer alan BLE beacon cihazlarını algılayacak ve arka planda çalışarak yolcunun kimliğini doğrulayacaktır.
- **Dinamik Ücretlendirme:** Yolcunun iniş durağı da beacon sensörleriyle algılanacak ve böylece gidilen mesafeye göre adil bir ücretlendirme modeli uygulanacaktır.
- **Kullanıcı Dostu Deneyim:** Yolcular yalnızca mobil uygulamayı yükleyerek sisteme entegre olacak, kart taşıma veya bakiye yükleme gibi işlemler tamamen ortadan kalkacaktır.

- **Veri Odaklı Yönetim:** Sistem, yolculuk verilerini zaman damgalı ve anonimleştirilmiş şekilde kaydederek şehir içi ulaşım planlaması için değerli bir veri kaynağı oluşturacaktır.

Bu amaç doğrultusunda geliştirilen sistem, hem kullanıcı deneyimini geliştirmeyi hem de şehir içi ulaşımında dijitalleşme ve akıllı şehir çözümlerine katkı sağlamayı hedeflemektedir.

1.4 Araştırma Soruları

Tez kapsamında yanıtlanması hedeflenen araştırma soruları şunlardır:

- BLE beacon teknolojisi, toplu taşımada mevcut biletleme sistemlerinin yerini alabilecek düzeyde otomatik ve güvenilir bir çözüm sunabilir mi?
- RESTful API, SOAP ve gRPC protokolleri performans açısından karşılaştırıldığında hangi protokol en uygun çözümleri sunmaktadır?
- Dinamik ücretlendirme modeli, hem kullanıcı memnuniyetini hem de operasyonel verimliliği artırma potansiyeline sahip midir?
- Geliştirilen sistem, şehir içi ulaşımında veri odaklı planlama süreçlerine nasıl katkı sunabilir?

1.5 Çalışmanın Kapsamı ve Sınırlılıkları

Bu tez çalışması, BLE beacon tabanlı bir biletleme sisteminin tasarımı, geliştirilmesi ve performans değerlendirmesi üzerine odaklanmaktadır. Çalışmanın kapsamı ve sınırlılıkları şu şekilde özetlenebilir:

- Çalışma yalnızca otobüs taşımacılığı üzerinden yürütülmüştür. Metro, tramvay veya diğer ulaşım araçları kapsam dışıdır.
- Siber güvenlik ile ilgili derinlemesine analizler (penetrasyon testleri, MITM saldırıları vb.) kapsam dışında bırakılmıştır.
- Ölçeklendirme testleri (çoklu sunucu yapıları, bulut tabanlı yük dengeleme) yapılmamıştır. Testler tek sunucu ortamında gerçekleştirilmiştir.

2. KAYNAK ARAŞTIRMASI

2.1. Toplu Taşımada Ücretlendirme Yöntemleri

Toplu taşımada ücretlendirme (fare policy), işletme maliyetlerinin karşılanması, talebin dengelenmesi ve toplumsal erişilebilirliğin korunması arasında bir denge kurulmasını amaçlamaktadır. Uygulanan yöntem, ağ yapısı, talep profili, gelir–eşitlik hedefleri, veri altyapısı ve kullanıcı deneyimi gibi çeşitli etmenlere bağlı olarak değişkenlik göstermektedir (Vuchic, 2005).

2.1.1 Sabit Ücretlendirme (Flat Fare)

Sabit ücretlendirme, hat veya hat grubu genelinde tek bir tarifenin uygulanması esasına dayanmaktadır. Bu yöntem, basitliği, hızlı validasyon sağlaması ve düşük işlem maliyeti ile öne çıkmaktadır (Hensher & Li, 2012). Ancak, mesafe veya hizmet farklılıklarını yansıtamaması nedeniyle mekânsal adalet açısından sınırlılıklar barındırmaktadır. Kısa mesafe yolcuların görece fazla, uzun mesafe yolcuların ise görece az ücret ödemesine yol açmaktadır.

2.1.2 Bölge Bazlı Ücretlendirme (Zone-Based Fare)

Bu yöntemde, kent coğrafyası ardışık bölgelere ayrılmakta ve yolculuk ücreti, başlangıç ile bitiş bölgeleri arasındaki geçiş sayısına göre hesaplanmaktadır. Londra ve Singapur gibi büyük şehirlerde yaygın olarak uygulanmakta olan bu sistem, basitlik ile adil fiyatlandırma arasında dengeli bir çözüm sunmaktadır (Currie & Delbosc, 2011). Ancak transferli yolculuklarda tutarsızlıklar oluşabilmekte ve kullanıcı açısından karmaşık bir yapı sergileyebilmektedir.

2.1.3 Mesafe Bazlı Ücretlendirme (Distance-Based Fare)

Gerçek seyahat mesafesine dayalı ücret hesaplaması, ekonomik etkinlik ve adalet bakımından en tutarlı yöntemlerden biri olarak kabul edilmektedir (Luo & Chen, 2017). Bu modelin uygulanabilmesi için hem binış hem de iniş bilgilerinin kaydedilmesi gerekmektedir. Turnikeli metro sistemlerinde bu süreç kolaylıkla yürütülürken, otobüs sistemlerinde zorluklar yaşanmaktadır. BLE tabanlı çözümler, inişin arka planda

otomatik olarak algılanmasını sağlayarak bu yöntemin daha pratik biçimde uygulanabilmesini mümkün kılmaktadır.

2.1.4 Süre Bazlı ve Zaman Pencereci Ücretlendirme

Belirli bir süre aralığında (örneğin 60–90 dakika) sınırsız transfer hakkı tanıyan zaman pencereci biletler, entegre ulaşım ağlarında çok modlu yolculukları teşvik etmektedir (Sharaby & Shifan, 2012). Ayrıca, yolculuk limitleme uygulamalarıyla günlük veya haftalık üst sınırların belirlenmesi, kullanıcı kabulünü ve sosyal erişilebilirliği artırmaktadır.

2.1.5 Tavan–Taban (Peak/Off-Peak) ve Dinamik Fiyatlama

Talebi zaman dilimlerine göre yönlendirmek amacıyla, yoğun saatlerde daha yüksek, düşük yoğunluklu saatlerde ise daha düşük ücretlerin uygulanması esas alınmaktadır (Zhou ve diğ., 2019). Bu yaklaşım kapasite kullanımını dengelemekte, ancak gelir adaleti açısından bazı tartışmalara yol açabilmektedir.

2.1.6 Dinamik/Esnek Ücretlendirme

Talep, doluluk oranı ve hat performansı gibi değişkenlere göre gerçek zamanlı olarak güncellenen tarifeler, gelir optimizasyonu ve kapasite yönetimi açısından avantaj sağlamaktadır (Zhang, 2021). Ancak bu tür uygulamalarda kullanıcı öngörülebilirliği azalabilmektedir. BLE gibi sensör tabanlı çözümler, yolcu verilerinin gerçek zamanlı toplanmasını sağlayarak dinamik fiyatlamanın daha etkin biçimde uygulanmasına olanak tanımaktadır (Dinh & Pham, 2020).

2.1.7 Sosyal Politika Araçları

Öğrenci, yaşlı ve engelli gruplara yönelik indirimler, abonman sistemleri ve üst sınır uygulamaları (fare capping) sosyal adaletin sağlanmasını amaçlamaktadır (Cats ve diğ., 2016). BLE tabanlı otomatik sistemler, bu tür indirimlerin kullanıcı müdahalesine gerek kalmadan otomatik biçimde uygulanmasını kolaylaştırmaktadır.

2.2. Beacon Nesne Kullanım Alanları

Beacon teknolojisi, düşük enerji tüketimi ve yüksek konum doğruluğu sayesinde yalnızca toplu taşımada değil, farklı sektörlerde de yaygın biçimde kullanılmaktadır. Literatürde, beacon cihazlarının özellikle perakende, kültür-turizm, sağlık, eğitim ve ulaşım alanlarında kullanıcı deneyimini geliştirmek ve veri odaklı yönetim süreçlerini desteklemek amacıyla uygulandığı belirtilmektedir.

2.2.1 Perakende Sektörü

Beacon teknolojisinin en erken benimsenen alanlarından biri perakende sektörü olarak öne çıkmaktadır. Bu sektörde beacon cihazları, mağaza içinde müşteri hareketlerinin izlenmesi, ürünlere yönelik kişiselleştirilmiş kampanyaların gönderilmesi ve kasa öncesi işlemlerin hızlandırılması için kullanılmaktadır (Zhou ve Piramuthu, 2015). Apple tarafından geliştirilen iBeacon çerçevesi, mağazalarda müşterilere anlık indirim bildirimleri gönderilmesi ve müşteri akışının yönetilmesi amacıyla uygulanmaktadır (López-de-Ipiña ve diğ., 2017). Bu uygulamalar, toplu taşımada yolculara dinamik tarife, hat yoğunluğu veya sefer bilgisi aktarılmasında örnek teşkil etmektedir.

2.2.2 Kültürel Alanlar ve Müzeler

Müze ve tarihi alanlarda beacon teknolojisi, ziyaretçilere konum tabanlı rehberlik sağlamak amacıyla kullanılmaktadır. Querini ve çalışma arkadaşları (2017), müze içerisinde beacon tabanlı bir rehber sisteminin uygulanabilirliğini test etmiş ve bu yaklaşımın ziyaretçilerin sergilere olan ilgisini artırdığını göstermektedir. Bu tür uygulamalar, toplu taşımada durak içi bilgilendirme ekranlarının yerini alabilecek kişisel, konum bazlı bilgilendirme çözümleri açısından örnek oluşturmaktadır.

2.2.3 Sağlık Hizmetleri

Sağlık sektöründe beacon teknolojisi, personel ve ekipman takibi, hasta yönlendirme ve acil durum yönetimi süreçlerinde etkin biçimde kullanılmaktadır. Yoon ve diğ. (2016), BLE beacon'ların hastane içi konum takibinde doğruluk ve maliyet

açısından önemli avantajlar sunduğunu ortaya koymaktadır. Sağlık alanında elde edilen bu deneyimlerin, toplu taşımada yoğun yolcu akışının bulunduğu terminallerde veya araçlarda benzer konum doğrulama uygulamalarına ilham verdiği görülmektedir.

2.2.4 Eğitim Kampüsleri

Eğitim kurumlarında beacon cihazları, ders yoklamalarının otomatik olarak alınması, kütüphane girişlerinin kontrol edilmesi ve öğrenci hareketliliğinin analiz edilmesi amacıyla kullanılmaktadır (Martín ve diğ., 2019). Bu uygulamaların, toplu taşımada hesap tabanlı biletleme ve otomatik yolcu doğrulama süreçleriyle benzerlik gösterdiği değerlendirilmektedir.

2.2.5 Ulaşım Sistemleri

Ulaşım alanında beacon cihazlarının en temel kullanım alanını biletleme sistemleri oluşturmaktadır. Otobüs ve tren istasyonlarına yerleştirilen beacon cihazları aracılığıyla biniş ve iniş verilerinin kaydedilmesi, yolcu yoğunluğunun ölçülmesi ve dinamik ücretlendirme için gerekli verilerin toplanması sağlanmaktadır (Ferreira ve diğ., 2020). Ayrıca, havalimanlarında beacon teknolojisinin yolcuların yönlendirilmesi ve bagaj takibi süreçlerinde de kullanılmakta olduğu belirtilmektedir (He, 2018).

2.2.6 Genel Değerlendirme

Literatürde beacon teknolojisinin yaygın biçimde benimsenmesinin, düşük maliyet, yüksek doğruluk ve kolay entegrasyon gibi avantajlardan kaynaklandığı görülmektedir. Ancak güvenlik, gizlilik ve veri bütünlüğü konularında bazı sınırlılıkların da bulunduğu vurgulanmaktadır (Gonzalez & Sanchez, 2018). Toplu taşımada uygulanacak sistemlerin, bu deneyimlerden yararlanarak güvenlik protokollerini güçlendirmesi ve veri gizliliğini öncelikli bir tasarım ilkesi olarak benimsemesi gerekmektedir.

2.3. Toplu Taşımada Biletleme Sistemlerinin Evrimi

Toplu taşıma biletleme sistemleri, şehirlerin nüfus artışı ve mobilite ihtiyaçları doğrultusunda tarihsel olarak farklı evrelerden geçmektedir. İlk dönemlerde kâğıt biletlerin kullanıldığı sistemler, operasyonel maliyetlerin yüksekliği ve sahteciliğe açık yapıları nedeniyle uzun vadede sürdürülebilirlik gösterememektedir (Yavuz & Çelik, 2021).

1990'lı yıllardan itibaren akıllı kart tabanlı sistemler yaygınlaşmaya başlamış ve özellikle temassız RFID (Radio Frequency Identification) kart teknolojisinin kullanıma girmesiyle birlikte daha hızlı ve güvenilir geçiş mekanizmaları oluşturulmaktadır (Sun & He, 2019). Bu dönemde Türkiye'de İstanbulKart, AnkaraKart ve KonyaKart gibi şehir bazlı uygulamaların devreye alınmasıyla yolculuk verilerinin merkezi biçimde toplanması sağlanmakta ve belediyeler açısından yeni yönetim olanakları elde edilmektedir (Yavuz & Çelik, 2021).

Bununla birlikte, akıllı kart tabanlı sistemler hâlen yolcudan manuel etkileşim talep etmektedir. Kartın validatöre yaklaştırılması, bakiye yükleme zorunluluğu veya cihaz arızaları gibi durumlar kullanıcı deneyimini olumsuz etkilemektedir (Chen, 2017). Ayrıca bu sistemlerde yalnızca biniş noktası kaydedilmekte, iniş durağı bilgisi çoğu zaman elde edilememektedir. Bu durum, mesafeye dayalı adil ücretlendirme modellerinin uygulanmasını güçleştirmektedir (Luo & Chen, 2017).

Sonraki aşamada, NFC (Near Field Communication) ve QR kod tabanlı mobil biletleme yöntemlerinin öne çıktığı görülmektedir. Ancak bu yöntemlerde de yolcudan aktif etkileşim beklenmektedir. Yolcunun mobil cihazını validatöre yaklaştırması veya QR kod okutması gerekmekte; bu durum sürecin tam otomatik biçimde yürütülmesini engellemektedir (Yan & Ma, 2018). QR kod teknolojisi özellikle düşük maliyetli çözümler arasında önemli bir yer edinmekte, ancak doğrulama süresinin görece uzun olması ve yoğun saatlerde cihazların okutulmasında yaşanan zorluklar nedeniyle sınırlı kalmaktadır (Chen, 2017).

Uluslararası literatürde yapılan çalışmalar, bu eğilimi desteklemektedir. Bohli, Sorge ve Westhoff (2015), mobil biletleme sistemlerinin güvenlik ve sahtecilik risklerini ayrıntılı biçimde incelemekte; kart tabanlı sistemlere göre bazı avantajlar sunmasına rağmen yeni güvenlik açıklarını da beraberinde getirdiğini ortaya koymaktadır. Zhang (2021) ise farklı ülkelerdeki akıllı kart ve mobil biletleme uygulamalarını karşılaştırmakta

ve geiş srecinde kullanıcı alışkanlıklarının ve yerel ynetim politikalarının belirleyici rol oynadığını vurgulamaktadır.

Sonuç olarak, toplu taşıma biletleme sistemlerinde tam otomasyona dođru bir evrim yaşanmaktadır. Kâğıt biletlerden RFID kartlara, ardından NFC ve QR kod tabanlı mobil czmlere geilmekte; ancak tm bu yntemler hâlen yolcunun aktif katılımını gerektirmektedir. Ayrıca iniş noktalarının dođrulukla kaydedilememesi, adil ve dinamik cretlendirme modellerinin uygulanmasında sınırlılıklar oluşturmaktadır. Bu nedenle literatrde bir sonraki aşama olarak, yolcunun hiçbir manuel işlem yapmadan arka planda tanınmasını ve cretlendirilmesini sađlayan BLE beacon tabanlı yaklaşımlar nerilmektedir (Namiot & Sneps-Snepe, 2014).

2.3 BLE'nin Ulaşım Sistemlerindeki Kullanımı

Bluetooth Low Energy (BLE) teknolojisinin toplu taşımada kullanımı grece yeni bir araştırma alanı olmasına rağmen, son yıllarda gerekleştirilen pilot projeler ve akademik çalışmalar bu teknolojinin yksek potansiyele sahip olduğunu ortaya koymaktadır. BLE, dşk enerji tketimi ve kısa menzil avantajı sayesinde toplu taşıma sistemlerinde hem yolcu tanıma hem de veri toplama amacıyla uygulanmaktadır.

Narzt ve alışma arkadaşları (2015), toplu taşımada “be-in/be-out” (BIBO) prensibine dayalı kavramsal bir model nermekte ve yolcunun otobse bindiđi anda beacon algılamasının gerekleştiiđi, herhangi bir manuel etkileşime gerek kalmadan seyahat bilgilerinin kaydedildiđi bir senaryoyu tartıřmaktadır. Bu yaklaşım, yolcu davranıřlarını deđiřtirmeden biletleme srecinin otomatikleřtirilmesini mmkn kılmakta ve literatrde nemli bir dnm noktası olarak deđerlendirilmektedir.

Kahvazadeh (2016), Almanya'nın Soest kentinde BLE tabanlı bir mobil biletleme pilot projesi gerekleřtirmekte ve otobslere yerleřtirilen beacon cihazları aracılıđıyla yolcu varlıđının tespit edilmesini sađlamaktadır. Ancak sistem, yolcunun uygulamayı manuel olarak etkinleřtirmesini gerektirmekte; bu durum tam otomatik bir srecin yrtlmesini engellemektedir. alışma, BLE'nin tek başına yeterli olmadığını, uygulama tarafında ek entegrasyonlara ihtiya duyulduđunu gstermektedir.

Ferreira, Dias ve Cunha (2020) tarafından Portekiz'in Porto řehrinde gerekleştirilen saha uygulamasında, BLE tabanlı biletlemenin gerek dnyadaki performansı deđerlendirilmektedir. Drt aylık pilot alışma sonucunda, otobslere

yerleştirilen beacon cihazları sayesinde yolcu tanımlamasının büyük ölçüde doğru biçimde yapılabildiği belirlenmektedir. Ayrıca kullanıcı deneyimi, enerji tüketimi ve teknik güvenilirlik değişkenleri de ölçülmekte; elde edilen sonuçlar BLE'nin toplu taşımada uygulanabilirliğini doğrulamaktadır. Ancak yalnızca otobüs içi beacon yerleşimi kullanıldığında iniş noktalarının yeterli doğrulukla tespit edilememesi, mesafe bazlı ücretlendirme açısından sınırlılık oluşturmaktadır.

Literatürde BLE'nin farklı şehirlerde ve farklı senaryolarda denendiği görülmektedir. Park ve Lee (2017), Kore'de BLE beacon'ların akıllı otobüs sistemlerinde uygulanmasını incelemekte ve yolcu yoğunluğu ölçümü ile konum takibinde başarılı sonuçlar elde edildiğini raporlamaktadır. Dias ve Ferreira (2019) ise şehir içi ulaşımda BLE tabanlı yolcu izleme sistemlerinin, yolcu davranışlarının modellenmesi ve hat optimizasyonu açısından güçlü bir veri kaynağı sunduğunu belirtmektedir.

Daha geniş bir perspektiften bakıldığında, BLE teknolojisinin yalnızca yolcu tanımlama amacıyla değil, aynı zamanda akıllı ulaşım yönetimi kapsamında veri üretimi için de önemli bir potansiyel sunduğu görülmektedir. Mehmood ve Graham (2015), akıllı ulaşım sistemleri üzerine yaptıkları çalışmada, BLE gibi düşük maliyetli sensör teknolojilerinin gelecekte büyük ölçekli yolcu akış yönetimi ve dinamik tarife uygulamalarının temelini oluşturacağını vurgulamaktadır.

Sonuç olarak, BLE'nin ulaşım sistemlerindeki kullanımı hâlen gelişim aşamasında bulunmakta; ancak gerçekleştirilen pilot projeler ve akademik çalışmalar, bu teknolojinin teknik açıdan uygulanabilirliğini ve kullanıcı deneyimine katkısını doğrulamaktadır. Mevcut uygulamalarda özellikle iniş noktası tespitinde doğruluk eksiklikleri ve mobil uygulama tarafında manuel etkileşim gereksinimi gibi sınırlılıklar dikkat çekmektedir. Bu sınırlılıkların giderilmesi, BLE tabanlı tam otomatik biletleme sistemlerinin yaygınlaşması açısından kritik bir adım olarak görülmektedir.

2.4 Veri Odaklı Ulaşım Yönetimi ve Dinamik Ücretlendirme

2.4.1 Kavramsal Çerçeve: “Veri Odaklı” Ne Demektir?

Veri odaklı ulaşım yönetimi; planlama, işletme ve fiyatlandırma kararlarının ölçülebilir göstergelere dayandırılmasını, bu göstergelerin gerçek zamanlı veya yakın gerçek zamanlı biçimde güncellenmesini ve sürekli iyileştirme döngüsüne (ölç-analiz-et-eyleme geç) bağlanmasını hedeflemektedir. Bu yaklaşım, sezgisel ya da statik tarifeler yerine

talep, kapasite, konfor, adalet ve finansal sürdürülebilirlik hedeflerinin birlikte optimize edilmesini amaçlamaktadır.

Temel hedefler aşağıdaki şekilde tanımlanmaktadır:

- Hizmet kalitesi: bekleme süresi, sefer düzenliliği (headway), doluluk oranı ve aktarma kolaylığının dengelenmesi,
- Operasyonel verimlilik: hat ve araç bazında doluluk dengesinin sağlanması, boş kapasitenin azaltılması,
- Finansal denge: gelir istikrarının sağlanması ve maliyet–gelir akışının öngörülebilir kılınması,
- Erişilebilirlik ve adalet: kısa mesafe kullanıcılarının aşırı ücretlendirilmeye maruz kalmaması ve sosyoekonomik gruplar arasında denge kurulması,
- Sağlık ve güvenlik: durak ve araçlardaki kuyruk uzunluklarının ve temas noktalarının azaltılması.

2.4.2 Veri Kaynakları ve BLE'nin Stratejik Rolü

Ulaşım yönetiminde karşılaşılan en kritik sorunlardan biri, yolcuların iniş (alighting) bilgilerinin güvenilir biçimde elde edilememesidir. Geleneksel kartlı sistemlerde genellikle yalnızca biniş kayıtları tutulmakta, iniş noktaları belirlenememektedir. BLE beacon cihazlarının araç içlerine ve duraklara yerleştirilmesiyle birlikte:

- Biniş ve iniş noktalarının çift taraflı biçimde saptanması,
- Gerçek mesafeye dayalı adil ücret hesaplamasının yapılması,
- Başlangıç–varış (OD) matrislerinin yüksek çözünürlükte çıkarılması,
- Kuyruk yoğunluklarının ve zaman damgalarının doğru biçimde kaydedilmesi mümkün hâle gelmektedir.

Bu temel veriler, ulaşım yönetimi açısından stratejik bilgi üretimini desteklemektedir. Tamamlayıcı veri kaynakları arasında araç telemetrisi (GPS, kapı açma–kapama, bekleme süresi), ödeme kayıtları, harici sensörler (kamera, Wi-Fi erişim noktaları) ve kullanıcı uygulama olayları yer almaktadır. Bu çoklu veri kaynakları, sistemin bütüncül biçimde analiz edilmesini sağlamaktadır.

2.4.3 Veri Boru Hattı ve Analitik Mimarisi

Veri işleme boru hattı üç ana katmandan oluşmaktadır:

1. **Toplama (İngest):** Mobil uygulama ve beacon olayları (UUID, Major, Minor, zaman damgası, RSSI), araç GPS verileri ve API günlük kayıtları toplanmaktadır.
2. **Dönüşüm ve Depolama:** Olay akışları gerçek zamanlı biçimde eşleştirilmekte, operasyonel veriler çevrimiçi işlem sistemlerinde (OLTP) tutulmakta ve analitik işlemler için sütunlu veri depolarına aktarılmaktadır.
3. **Analitik ve Sunum:** Elde edilen veriler üzerinden anahtar performans göstergeleri (KPI) hesaplanmakta, talep tahminleri yapılmakta ve ücret simülasyonları gerçekleştirilmektedir.

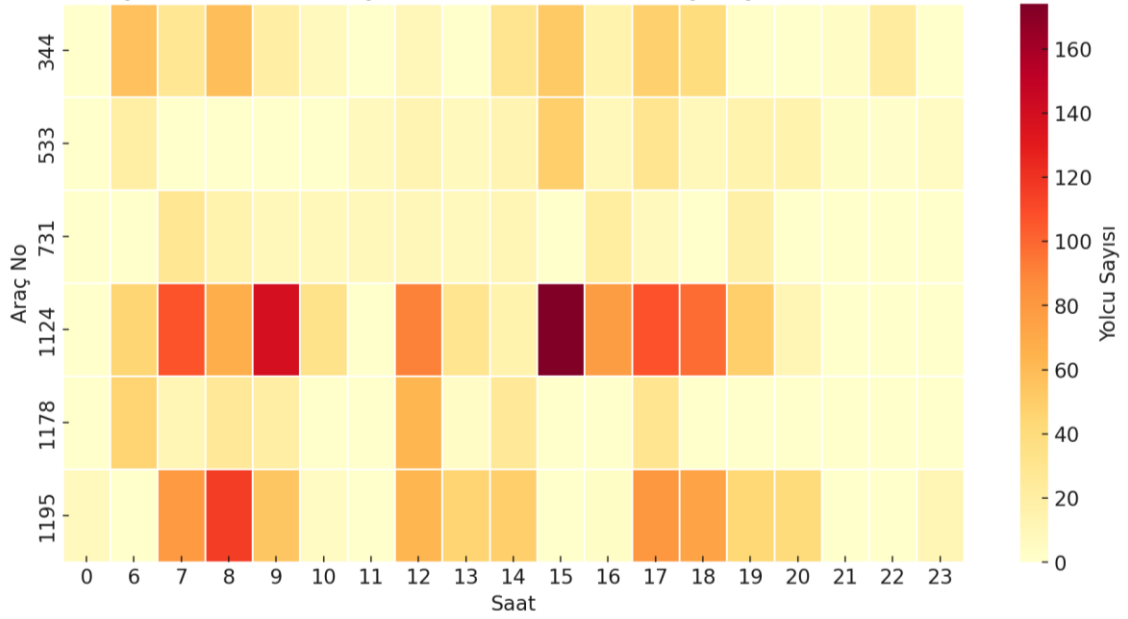
Bu yapı sayesinde veri akışı bütünlük bir sistemde işlenmekte ve karar destek süreçlerine doğrudan katkı sağlamaktadır.

2.4.4 Başlangıç–Varış (OD) Çıkarımı ve Yoğunluk Analizi

BLE ile hem biniş hem de iniş bilgilerinin yakalanması, doğrudan başlangıç–varış matrislerinin oluşturulmasını mümkün kılmaktadır. Böylece:

- Hat ve durak bazında talep ısı haritaları çıkarılmakta,
- Yoğun saat (peak) ve zayıf saat (off-peak) dönemleri net biçimde belirlenmekte,
- Aktarma desenleri analiz edilerek yolcu davranışları modellenmektedir.

Bu analizler, ulaşım planlamasında kapasite tahsisi ve sefer optimizasyonu kararlarının verilmesinde kullanılmaktadır.



Şekil 2.1 Hatlar için saat dilimi bazlı biniş/iniş ısı haritası

2.4.5 Anahtar Performans Göstergeleri (KPI) Seti

Veri odaklı ulaşım yönetiminde performans değerlendirmesi, belirli gösterge setleri üzerinden yapılmaktadır. Bu göstergeler:

- Hizmet kalitesi: ortalama bekleme süresi, düzenli sefer oranı, yolculuk süresi, aktarma süresi,
- Konfor: araç içi doluluk oranı, ayakta yolcu yüzdesi,
- Kuyruk yoğunluğu: durak bazında ortalama bekleme süresi ve tepe kuyruk uzunluğu,
- Ücret adaleti: kilometre başına ortalama ücret, kısa ve uzun mesafe farkı,
- Finansal göstergeler: gelir–hat dengesi, tahsilat oranı,
- Davranışsal göstergeler: yoğun olmayan saatlerde kaydırma oranı ve bildirim etkileşim süreleri biçiminde tanımlanmaktadır.

Bu göstergeler aracılığıyla hizmet kalitesi ve mali sürdürülebilirlik birlikte değerlendirilmektedir.

2.4.6 Talep Tahmini ve Simülasyon

Dinamik fiyatlandırma kararlarının oluşturulmasında talep tahmin modelleri kullanılmaktadır. Bu modeller:

- Zaman serisi analizleriyle kısa vadeli (15–30 dakikalık) talep öngörülerini üretmekte,
- Hava durumu, etkinlik, akademik takvim ve resmi tatil gibi dışsal değişkenleri dikkate almakta,
- Kuyruk kuramı (M/M/1, M/M/c) veya ajan tabanlı simülasyon yöntemleriyle yoğunluk tahminlerini gerçekleştirmektedir.

Elde edilen sonuçlar, hat bazında doluluk dengesinin sağlanması ve yoğunluk yönetiminin optimize edilmesi amacıyla kullanılmaktadır.

2.4.7 Dinamik Ücretlendirme: Modeller ve Tasarım İlkeleri

Ücret fonksiyonu çok bileşenli olabilir:

$$\text{Ücret} = \beta_0 + \beta_1 * \text{Mesafe} + \beta_2 * \text{Zaman Dilimi} + \beta_3 * \text{Doluluk} + \beta_4 * \text{Aktarma} - \text{İndirimler}$$

Dinamik ücretlendirme modelleri, farklı bileşenlerin etkileşimiyle tasarlanmaktadır. En yaygın uygulama biçimleri arasında mesafe bazlı, bölge bazlı, yoğunluk duyarlı ve aktarma indirimli modeller bulunmaktadır.

Bu modellerde fiyatlar, belirli politika kısıtları dâhilinde belirlenmektedir. Ani fiyat artışlarının engellenmesi, kullanıcıya önceden bilgilendirme yapılması ve basit, anlaşılır tarife mesajlarının sunulması önerilmektedir. Böylece sistemin hem adil hem de öngörülebilir biçimde çalışması sağlanmaktadır.

2.4.8 BLE'nin Dinamik Ücretlendirmeyi “Mümkün Kılma” Etkisi

BLE teknolojisi, dinamik ücretlendirme sistemlerinin uygulanabilirliğini doğrudan artırmaktadır. İniş tespitinin doğru biçimde yapılması, gerçek mesafeye dayalı ücret hesaplamasının gerçekleştirilebilmesini sağlamaktadır. Zaman damgaları sayesinde yoğun saatlerde indirim veya yönlendirme teşviklerinin otomatik biçimde uygulanması mümkün olmaktadır. Ayrıca araç içi doluluk bilgileri ile yolcu uygulama verileri

birleştirilerek kalabalık duyarlı mikro-teşvikler tanımlanabilmektedir. Böylece hem sistem verimliliği artırılmakta hem de kullanıcı memnuniyeti yükseltilmektedir.

2.4.9 Etki Değerlendirmesi: Deney Tasarımı ve Ölçüm

Pilot uygulamalarda etki değerlendirme, deneysel tasarım ilkelerine göre gerçekleştirilmektedir. Aynı hatlarda “önce–sonra” karşılaştırmaları yapılmakta ve mevsimsel etkiler farklarının farkı (difference-in-differences) yöntemiyle ayrıştırılmaktadır. Ayrıca, benzer talep profiline sahip hatlar üzerinde A/B karşılaştırmaları uygulanarak dinamik ve statik tarifelerin etkileri değerlendirilmektedir. Elde edilen sonuçlar, ortalama bekleme süresi, kuyruk uzunluğu, gelir istikrarı ve adalet göstergeleri üzerinden ölçülmektedir.

2.4.10 Gizlilik (KVKK/GDPR) ve Etik İlkeler

Sistemde toplanan veriler, kişisel verilerin korunması ilkelerine uygun biçimde işlenmektedir. Kullanıcı kimliği ile işlem verisi birbirinden ayrıştırılmakta, yalnızca zorunlu alanlar kaydedilmekte ve veriler belirli saklama süreleri sonunda anonim hâle getirilmektedir. Ayrıca yayımlanan analizlerde yeniden tanımlama riskini azaltmak için diferansiyel gizlilik yöntemleri uygulanmaktadır. Kullanıcılar, verilerinin kullanımına ilişkin şeffaf bilgilendirmelerle bilgilendirilmekte ve dilediklerinde sistemden çıkış hakkına sahip olmaktadır.

2.4.11 Uygulama Yol Haritası (Operasyonelleştirme)

Uygulama süreci, aşamalı biçimde gerçekleştirilmektedir. Öncelikle BLE biniş–iniş doğruluk oranlarının %95’in üzerine çıkarılması hedeflenmekte, ardından pilot hatlarda simülasyon sonuçlarına göre kademeli yaygınlaştırma yapılmaktadır. Pilot uygulamalardan elde edilen veriler KPI panoları aracılığıyla izlenmekte, şikâyet ve geri bildirimler analiz edilmekte ve sistem performansı düzenli olarak değerlendirilmektedir.

2.4.12 Sonuç: Neden Veri Odaklı Dinamik Ücret?

Veri odaklı dinamik ücretlendirme, toplu taşımada kuyruk sürelerini ve temas noktalarını azaltarak hizmet akışını hızlandırmaktadır. Gerçek mesafeye ve aktarma bilgisine dayalı hesaplamalar, ücret adaletini artırmaktadır. Talebe göre yönlendirilen indirim ve teşvik mekanizmaları, yoğunluk dengesizliğini azaltmakta ve sistemin sürdürülebilirliğine katkı sağlamaktadır. Böylece hem operasyonel verimlilik hem de gelir istikrarı birlikte geliştirilmektedir.

2.5 Web Servis Protokolleri ve Performans Karşılaştırmaları

Gerçek zamanlı ve yüksek hacimli veri işleyen toplu taşıma biletleme sistemlerinde kullanılan web servis protokolü, sistemin genel performansını doğrudan etkilemektedir. Literatürde en yaygın olarak kullanılan üç yaklaşım; SOAP (Simple Object Access Protocol), REST (Representational State Transfer) ve gRPC (Google Remote Procedure Call) olarak öne çıkmaktadır.

SOAP, XML tabanlı mesajlaşma yapısı sayesinde güvenlik, bütünlük ve standartlaşma açısından güçlü bir seçenek olarak değerlendirilmektedir. Ancak mesaj boyutlarının büyük olması ve işlem süresinin uzaması, özellikle yüksek kullanıcı trafiğine sahip sistemlerde performans sorunlarına yol açmaktadır (Banerjee ve diğ., 2016).

REST, JSON tabanlı hafif veri yapılarıyla daha esnek ve performans odaklı bir çözüm sunmaktadır. Mobil uygulamalarda yaygın biçimde kullanılmakta olan bu protokol, JSON'un platform bağımsızlığı ve HTTP protokolüyle doğal uyumu sayesinde önemli avantajlar sağlamaktadır (Kiryakov, 2020). Bununla birlikte, JSON'un metin tabanlı yapısı nedeniyle yüksek hacimli veri işlemede serileştirme ve çözümleme maliyetlerinin arttığı belirtilmektedir (Ahmed & Khan, 2021).

gRPC, Google tarafından geliştirilen ve Protocol Buffers (Protobuf) tabanlı ikili veri serileştirmesi kullanan yeni nesil bir teknolojidir. Literatürde gRPC'nin özellikle düşük gecikme süresi ve yüksek verimlilik gerektiren sistemlerde önemli avantajlar sunduğu gösterilmektedir (Shen ve diğ., 2018). Bununla birlikte, gRPC'nin karmaşık yapılandırma gereksinimleri ve REST kadar yaygın desteklenmemesi, uygulama süreçlerinde adaptasyon zorlukları oluşturabilmektedir (Guo & Lin, 2022).

Çeşitli karşılaştırmalı çalışmalar, bu üç protokolün performans farklarını açık biçimde ortaya koymaktadır. Miklos, Kovacs ve Molnar (2019), JSON, XML ve Protobuf

tabanlı web servislerini kıyaslayarak gRPC'nin gecikme süresi açısından en hızlı sonuçları verdiğini, REST'in orta düzeyde performans gösterdiğini, SOAP'ın ise en yavaş protokol olduğunu belirtmektedir. Narzt, Pomberger ve Sametinger (2015), SOAP ve REST'i gRPC'nin erken sürümleriyle karşılaştırarak SOAP'ın ağır yapısı nedeniyle modern uygulamalarda verimsiz kaldığını vurgulamaktadır. Ahmed ve Khan (2021) ise bulut tabanlı dağıtık sistemlerde REST'in yüksek throughput sağladığını, ancak gRPC'nin daha düşük kaynak kullanımıyla daha istikrarlı sonuçlar ürettiğini ifade etmektedir.

Shen, Yu ve Bu (2018), mikroservis tabanlı mimarilerde REST ve gRPC performansını karşılaştırarak gRPC'nin gecikme süresi ve CPU kullanımı açısından belirgin bir üstünlük sağladığını, REST'in ise geniş ekosistem desteği sayesinde entegrasyon kolaylığı sunduğunu ortaya koymaktadır. Guo ve Lin (2022) de benzer biçimde, gRPC'nin ölçeklenebilirlik açısından REST'e göre daha verimli bir yapı sunduğunu, ancak SOAP'ın kurumsal standartlar nedeniyle bazı sektörlerde hâlen kullanılmakta olduğunu belirtmektedir.

Sonuç olarak, literatürde REST, SOAP ve gRPC protokollerinin farklı güçlü ve zayıf yönlere sahip olduğu görülmektedir. SOAP, güvenlik ve standartlaşma bakımından avantajlı olmakla birlikte performans açısından sınırlılıklar taşımaktadır. REST, dengeli bir performans ve yaygın ekosistem desteği sağlamaktadır. gRPC ise düşük gecikme süresi, kompakt veri yapısı ve kaynak verimliliğiyle modern sistemlerde en uygun alternatiflerden biri olarak değerlendirilmektedir. Bu farkların deneysel olarak karşılaştırılması, BLE tabanlı toplu taşıma sistemleri gibi performansın kritik önem taşıdığı alanlarda doğru protokol seçiminin yapılabilmesi açısından önem arz etmektedir.

2.6 Akıllı Şehirler, IoT ve Güvenlik Boyutu

Akıllı şehir kavramı, ulaşım altyapısının dijitalleşmesini, gerçek zamanlı veri toplama ve işleme kapasitesinin artırılmasını, aynı zamanda güvenlik ve gizlilik ilkelerinin korunmasını kapsamaktadır. Bu bağlamda, BLE beacon tabanlı biletleme sistemleri yalnızca toplu taşıma hizmetlerini kolaylaştırmakla kalmamakta, aynı zamanda şehir ölçeğinde veri odaklı ulaşım yönetimi ve akıllı mobilite çözümlerine katkı sağlamaktadır (Fazio ve diğ., 2016).

Nesnelerin İnterneti (IoT) ekosisteminin ulaşım sistemlerine entegrasyonu, farklı sensörlerin birlikte çalışabilirliği ile mümkün olmaktadır. Al-Fuqaha ve çalışma

arkadaşları (2015), IoT'nin ulaşım alanındaki uygulamalarına ilişkin kapsamlı bir incelemede, BLE, NFC ve RFID gibi teknolojilerin bütünleşik biçimde kullanılarak gerçek zamanlı yolcu verilerinin şehir yönetim sistemlerine aktarılabilceğini vurgulamaktadır. Bu sayede yolcu yoğunluklarının anlık olarak izlenmesi, hat optimizasyonu ve dinamik ücretlendirme uygulamaları gerçekleştirilebilmektedir.

Bununla birlikte, literatürde IoT ve BLE tabanlı çözümlerin güvenlik açısından bazı kritik zorlukları da beraberinde getirdiği belirtilmektedir. Gonzalez ve Sanchez (2018), BLE beacon tabanlı biletleme sistemlerinde sahteciliğe karşı özellikle kimlik kopyalama (spoofing) riskinin önemli bir tehdit oluşturduğunu vurgulamaktadır. Bu risk, yolcunun gerçek varlığı yerine kopyalanmış sinyallerin sistem tarafından algılanması sonucunda adil olmayan ücretlendirmelere yol açabilmektedir. Benzer biçimde, Bohli, Sorge ve Westhoff (2015), mobil biletleme sistemlerinde bilgi sızıntısı ve yetkisiz erişim risklerini incelemekte, kullanıcı verilerinin güçlü şifreleme ve kimlik doğrulama protokolleriyle korunması gerektiğini belirtmektedir.

Mehmood ve Graham (2015), akıllı ulaşım sistemlerinin güvenlik sorunlarının yalnızca teknolojik değil, aynı zamanda yönetişimsel bir boyut taşıdığını ifade etmektedir. Farklı kurumların veri toplama süreçlerinde yer alması, veri bütünlüğünün ve güvenliğinin korunmasını zorlaştırmaktadır. Bu nedenle, ulaşım sistemlerinin güvenlik politikalarının merkezi bir çerçeve içerisinde yönetilmesi gerekmektedir.

Guo ve Lin (2022) tarafından gerçekleştirilen çalışmada ise mikroservis tabanlı ulaşım çözümlerinde gRPC ve REST gibi protokollerin güvenlik yapıları karşılaştırılmakta; SOAP'ın bu açıdan daha standart ve tanımlı bir güvenlik çerçevesi sunduğu, ancak yüksek işlem maliyeti nedeniyle modern uygulamalarda geride kaldığı belirtilmektedir.

Sonuç olarak, akıllı şehir vizyonu doğrultusunda BLE tabanlı biletleme sistemlerinin önemli fırsatlar sunduğu, ancak ölçeklenebilirlik, güvenlik ve mahremiyet konularında çözümlenmesi gereken zorluklar bulunduğu görülmektedir. Literatürde, bu sistemlerin teknik açıdan uygulanabilir olduğu doğrulanmakta; ancak güvenlik altyapısının güçlendirilmesi gerektiği vurgulanmaktadır. Güvenlik önlemlerinin kurumsal politikalarla desteklenmesi ve sistemlerin düzenli olarak dış denetimlere tabi tutulması, sürdürülebilir ve güvenilir akıllı ulaşım ekosistemlerinin oluşturulması açısından kritik önem taşımaktadır.

2.7 Literatür Özeti ve Araştırma Boşluğu

Literatürde BLE, NFC, RFID ve QR kod gibi farklı temassız teknolojilerin toplu taşımada uygulanmasına yönelik çok sayıda çalışma bulunmaktadır. Ancak bu çalışmaların büyük bir kısmının, sistemin yalnızca belirli bir yönüne odaklandığı ve bütüncül bir mimari yaklaşım ortaya koymadığı görülmektedir.

Narzt ve çalışma arkadaşları (2015), “be-in/be-out” prensibine dayalı BLE tabanlı bir kavramsal model önermekte ve manuel etkileşim gerektirmeyen bir biletleme sisteminin temel yapısını açıklamaktadır. Ancak çalışmada sistem performansına ilişkin ölçümler yapılmamakta, yalnızca modelleme düzeyinde bir tartışma sunulmaktadır. Kahvazadeh (2016) tarafından gerçekleştirilen pilot uygulamada, BLE tabanlı biletleme süreci test edilmekte, ancak kullanıcıdan manuel uygulama başlatma adımı beklenmektedir. Bu durum, tam otomatik geçiş hedefinin sağlanamadığını göstermektedir.

Ferreira, Dias ve Cunha (2020) tarafından yapılan çalışmada, otobüs tabanlı BLE sistemlerinde biniş tespitinin yüksek doğrulukla yapılabildiği, ancak iniş tespitinde hata oranının yükseldiği belirtilmektedir. Park ve Lee (2017), BLE'nin yolcu yoğunluğu ölçümünde başarılı olduğunu, ancak yüksek sinyal yansıma oranlarının konum doğruluğunu etkileyebildiğini göstermektedir. Dias ve Ferreira (2019) ise BLE verilerinin hat optimizasyonu ve kapasite planlamasında kullanılabileceğini ortaya koymakta, fakat sistemin ücretlendirme modeliyle entegrasyonunu ele almamaktadır.

Zhang (2021) tarafından yapılan karşılaştırmalı incelemede, BLE ve NFC tabanlı çözümlerin birlikte kullanılmasının teorik olarak daha yüksek doğruluk sağlayabileceği, ancak maliyet ve karmaşıklık açısından pratikte sınırlılıklar barındırdığı ifade edilmektedir. Benzer biçimde, Bohli, Sorge ve Westhoff (2015), mobil biletleme sistemlerinin güvenlik zafiyetlerine dikkat çekmekte ve özellikle kimlik doğrulama süreçlerinin güçlendirilmesi gerektiğini vurgulamaktadır.

Yapılan bu incelemeler, mevcut literatürde BLE tabanlı biletleme sistemlerine ilişkin çalışmaların ağırlıklı olarak kavramsal düzeyde kaldığını göstermektedir. Ayrıca, farklı web servis protokollerinin (REST, SOAP, gRPC) performans açısından doğrudan karşılaştırıldığı bir araştırmaya rastlanmamaktadır. Bununla birlikte, BLE tabanlı sistemlerin enerji tüketimi, CPU kullanımı ve ağ trafiği üzerindeki etkilerini analiz eden kapsamlı deneysel çalışmaların sayısı da sınırlı kalmaktadır.

Dolayısıyla, literatürde üç temel araştırma boşluğu belirlenmektedir:

1. BLE tabanlı tam otomatik biletleme sistemlerinin, hem donanım hem de yazılım boyutunda bütüncül biçimde ele alınmaması,
2. Farklı web servis protokollerinin aynı sistem üzerinde karşılaştırmalı performans analizlerinin yapılmamış olması,
3. BLE'nin dinamik ücretlendirme ve veri odaklı ulaşım yönetimiyle entegrasyonuna ilişkin deneysel bulguların yetersiz kalması.

Bu çalışma, söz konusu boşlukları gidermeyi amaçlamaktadır. BLE beacon tabanlı tam otomatik bir biletleme sisteminin prototipi geliştirilmekte, REST, SOAP ve gRPC protokolleri performans açısından karşılaştırılmakta ve sistemin CPU, bellek ve ağ trafiği üzerindeki etkileri deneysel olarak değerlendirilmektedir. Elde edilen bulguların, literatürdeki mevcut yaklaşımların ötesine geçerek BLE tabanlı biletleme sistemlerinin uygulanabilirliğini teknik düzeyde ortaya koyması hedeflenmektedir.

3. MATERYAL VE YÖNTEM

3.1 Sistem Tasarımı

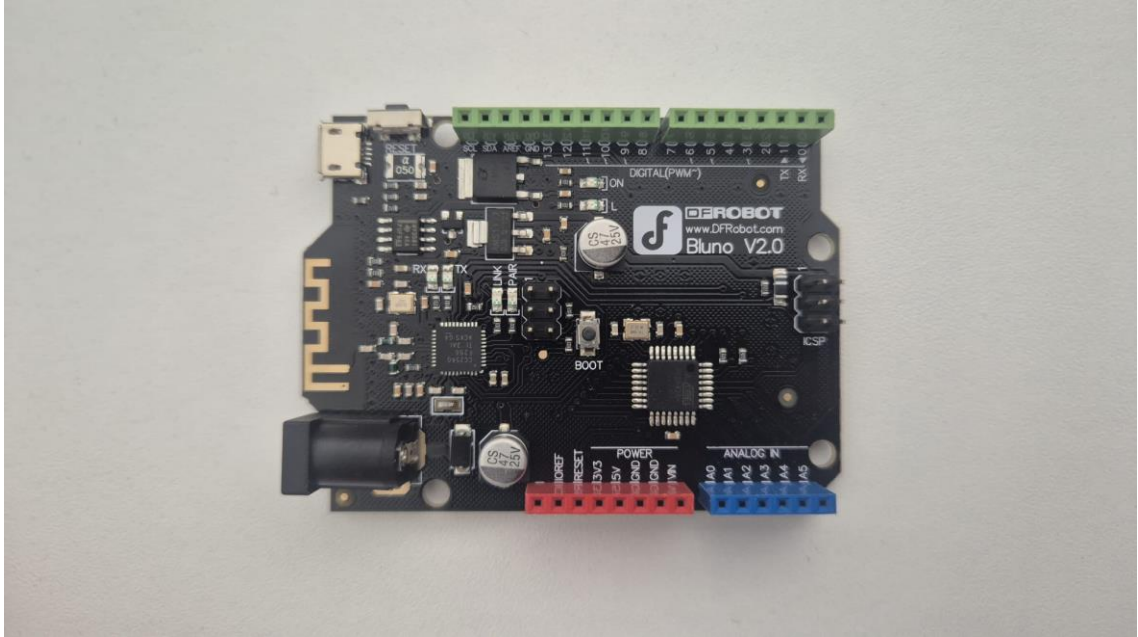
3.1.1 Beacon Cihazları

Bu çalışmada kullanılan beacon cihazları, Bluetooth Düşük Enerji (BLE) standardına uygun, düşük enerji tüketimli ve programlanabilir modüllerdir. Her beacon cihazı, UUID, Major ve Minor değerlerini periyodik olarak yayınlamaktadır. Çalışma kapsamında toplam 12 adet beacon cihazı tedarik edilmiş ve hem otobüs içlerine hem de otobüs duraklarına yerleştirilmiştir. Kullanılan beacon cihazı Şekil 1.3'te verilmiştir.

Cihazların temel teknik özellikleri:

- Yayın Gücü: -30 dBm ile +4 dBm arası ayarlanabilir
- Pil Ömrü: CR2032 lityum pil ile ortalama 12 ay
- Yayın Aralığı: 100 ms – 1 sn.
- Çalışma Frekansı: 2.4 GHz

Otobüslerde beacon cihazları kapı girişlerine ve orta bölgelere monte edilmiştir. Duraklarda ise yolcu yoğunluğunu en iyi yansıtacak noktalara (giriş/çıkış yönleri) sabitlenmiştir. Böylece hem binış hem iniş noktaları güvenilir biçimde algılanabilmiştir.



Şekil 3.1. Beacon Entegre Devresi



Şekil 3.2 BLE Cihazı Otobüs içerisinde

3.1.1.1 BLE'nin Genel Özellikleri

Bluetooth Düşük Enerji (BLE), Bluetooth 4.0 standardı ile birlikte tanıtılmış ve özellikle düşük güç tüketimi gerektiren uygulamalar için geliştirilmiş bir kablosuz haberleşme teknolojisidir (Al-Fuqaha ve diğ., 2015). Klasik Bluetooth'a kıyasla çok daha düşük enerjile çalışabilmesi, küçük bataryalarla uzun süreli kullanım sağlaması ve kısa mesafede güvenilir veri aktarımı yapabilmesi, BLE'yi Nesnelerin İnterneti (IoT) ekosisteminin temel taşlarından biri haline getirmiştir (Wang & Wu, 2016).

BLE'nin öne çıkan özellikleri şunlardır:

- **Düşük enerji tüketimi:** Tipik bir beacon cihazı küçük bir pil ile aylar hatta yıllar boyunca çalışabilmektedir (Li & Zhao, 2019).
- **Kısa mesafe iletişimi:** 1–50 metre arasında güvenilir veri aktarımı mümkündür.

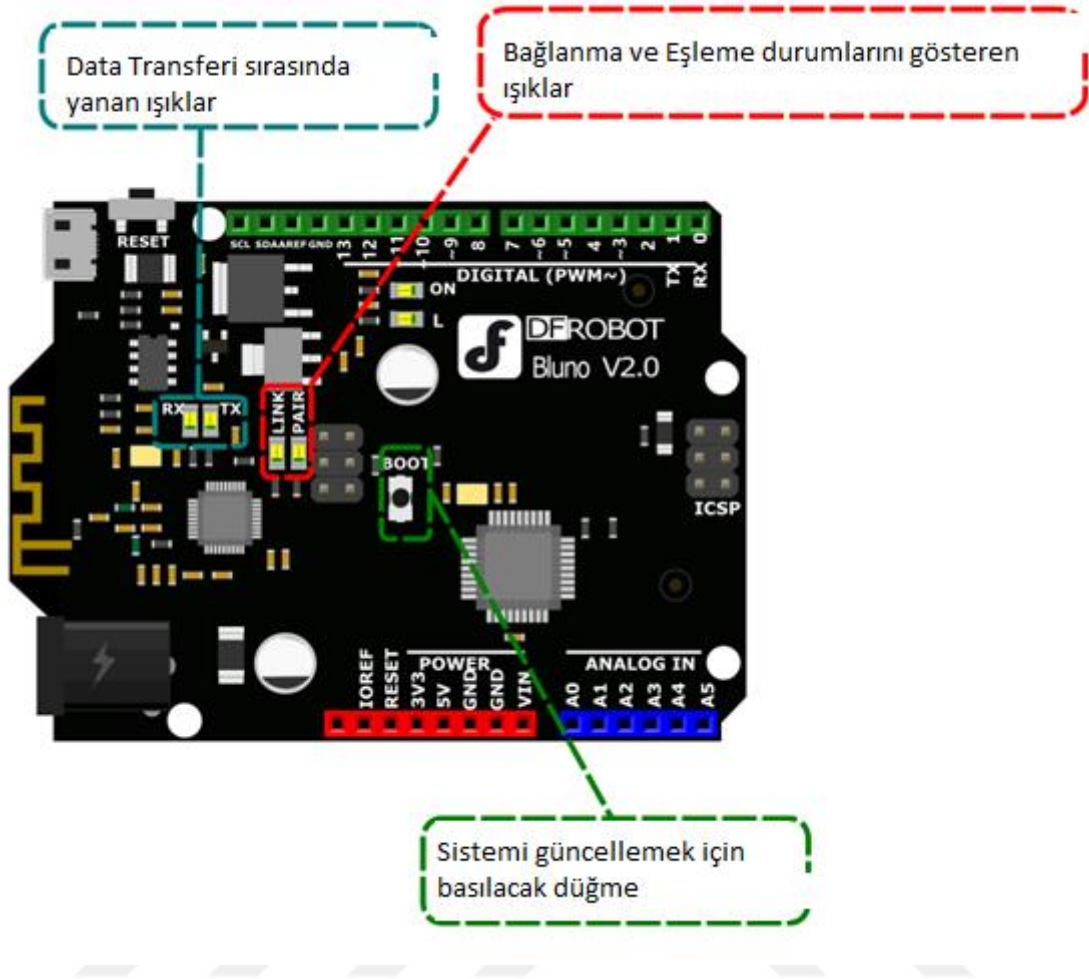
- **Uygun maliyet:** Donanım üretimi ucuzdur, bu da geniş ölçekli kurulumlarda avantaj sağlar.
- **Yaygın cihaz desteği:** Modern akıllı telefonların büyük çoğunluğu BLE uyumludur (Namiot & Sneps-Sneppe, 2014).

Bu özellikler, BLE'yi akıllı şehir uygulamalarında, sağlık takip cihazlarında, akıllı ev sistemlerinde ve özellikle ulaşım çözümlerinde tercih edilen bir teknoloji haline getirmiştir (Fazio ve diğ., 2016).

3.1.1.2 Beacon Cihazlarının Yapısı

BLE beacon cihazları, küçük boyutlu ve genellikle pille çalışan donanımlardır. Temel bileşenleri şunlardan oluşur:

- **Bluetooth Çipi:** Sinyali oluşturan ve ileten çekirdek donanım.
- **Mikroişlemci:** Beacon kimliği, yayın periyodu ve güvenlik ayarlarını yöneten kontrol ünitesi.
- **Güç Kaynağı:** Çoğunlukla düğme pilleri (CR2032) veya USB adaptörler.
- **Anten:** Sinyalin yönünü ve gücünü belirler.
- **Kasa:** Çevresel etkilere (nem, sıcaklık, darbe) karşı dayanıklılığı sağlar.



Şekil 3.2. BLE Beacon cihaz yapısı

Beacon cihazları genellikle belirli aralıklarla (ör. 100 ms – 2 sn.) kimlik bilgilerini (UUID, Major, Minor değerleri) yayımlar. Bu sinyal, alıcı cihazlar tarafından algılanır ve sinyal gücüne (RSSI – Received Signal Strength Indicator) göre mesafe tahmini yapılabilir (Faragher & Harle, 2015).

3.1.1.3 Beacon Protokolleri

BLE beacon teknolojisinde en yaygın iki protokol bulunmaktadır:

- **iBeacon (Apple, 2013):** Beacon cihazları, 128-bit UUID + Major + Minor verileri yayımlar. Bu üçlü kombinasyon cihazı benzersiz şekilde tanımlar.
- **Eddystone (Google, 2015):** Açık kaynak kodlu bir protokoldür, farklı veri çerçeveleri sunar:
 - *Eddystone-UID:* Kimlik bilgisi taşır.
 - *Eddystone-URL:* Doğrudan bir URL yayımlar.

- *Eddystone-TLM*: Pil seviyesi, sıcaklık gibi telemetri verileri sağlar.
- *Eddystone-EID*: Güvenlik için şifrelenmiş kimlik sunar. (Namiot & Sneps – Sneppe, 2014)

3.1.1.4 Sinyal ve Konum Belirleme Yöntemleri

BLE beacon'lar yalnızca kimlik bilgisi iletmez, aynı zamanda konum tespitinde de kullanılabilir. Literatürde öne çıkan konum belirleme yöntemleri şunlardır:

- **Sinyal Gücü (RSSI)**: En yaygın yöntemdir; sinyal gücüne göre yaklaşık mesafe hesaplanır (Li & Zhao, 2019).
- **Üçgenleme (Triangulation)**: En az üç beacon'dan gelen sinyallerin açı farkına göre konum belirlenir.
- **Çokgenleme (Multilateration)**: Farklı beacon'lardan mesafe hesaplanarak kesişim noktası bulunur (Chou & Lee, 2016).
- **Parmak izi yöntemi (Fingerprinting)**: Önceden kaydedilen sinyal haritaları ile anlık değerler karşılaştırılır; özellikle karmaşık iç mekânlarda daha doğru sonuç verir (Faragher & Harle, 2015).

3.1.1.5 BLE'nin Avantajları ve Kısıtları

Literatürde BLE'nin en çok öne çıkan avantajları şunlardır:

- Düşük enerji tüketimi (Wang & Wu, 2016).
- Düşük kurulum maliyeti (Fazio ve diğ., 2016).
- Yaygın mobil cihaz desteği (Al-Fuqaha ve diğ., 2015).
- Yüksek doğruluk, özellikle kısa mesafe iç mekânlarda (Li & Zhao, 2019).

Ancak aynı zamanda bazı kısıtlar da vurgulanmaktadır:

- Metal ve beton yüzeylerden kaynaklanan sinyal zayıflaması.
- Kalabalık ortamlarda parazitlenme riski.
- RSSI değerlerindeki dalgalanmalar nedeniyle mesafe hesaplamasında hata payı (Chou & Lee, 2016).
- Beacon kimliğinin kopyalanma riski (spoofing) nedeniyle güvenlik açıkları (Gonzalez & Sanchez, 2018).



Şekil 3.3. Uygulama ile otobüse binış simülasyonu

3.1.2 Mikrodenetleyici ve Güvenlik Mekanizmaları

Beacon cihazları, bir ESP32 tabanlı mikrodenetleyici üzerinden yönetilmiştir. Mikrodenetleyici, her sefer öncesinde sunucudan aldığı SEED verisini kullanarak beacon cihazlarının Major/Minor değerlerini güncellemiştir. Bu yaklaşım, beacon kimliklerinin sabit kalmasını engelleyerek spoofing (taklit etme) saldırılarına karşı koruma sağlamıştır.

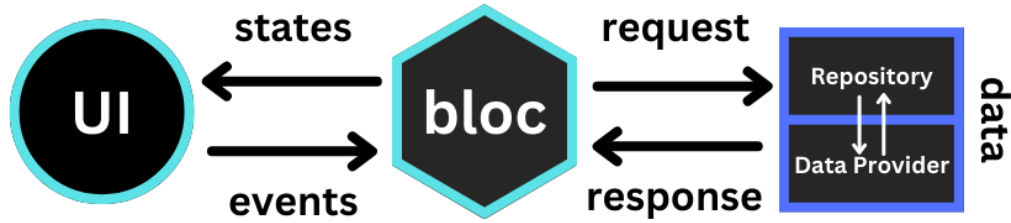
Sefer boyunca mikrodenetleyici, beacon kimliklerini belirli aralıklarla güncelleyerek sistem güvenliğini sürdürmüştür. Yapılan testlerde bu yöntemle, üçüncü taraf cihazların beacon taklidi yapması büyük oranda engellenmiştir.

3.1.3 Mobil Uygulama

Çalışma kapsamında Flutter SDK (v3.19.0) ve Dart dili kullanılarak bir mobil uygulama geliştirilmiştir. Uygulama Android cihazlarda test edilmiş, BLE tarama ve GPS doğrulama işlevleri arka planda çalışacak şekilde optimize edilmiştir.

Uygulama, üç temel modül üzerine inşa edilmiştir:

1. **Beacon Algılama Modülü:** flutter_blue_plus kütüphanesi kullanılarak çevredeki beacon cihazları sürekli taranmıştır. Algılanan beacon kimlikleri sunucuya iletilmiştir.
2. **Konum Doğrulama Modülü:** geolocator paketiyle yolcunun GPS konumu elde edilip, otobüsün konumuyla karşılaştırılmıştır.
3. **Ücretlendirme Modülü:** Biniş ve iniş noktaları belirlendikten sonra ücret hesaplaması yapılmış, kullanıcıya bildirim gönderilmiştir.



Şekil 3.5: Mobil uygulama mimarisi

3.1.4 Sunucu Tarafı Servisleri

Sunucu tarafında üç farklı API protokolü uygulanmıştır:

- RESTful API (JSON tabanlı)
- SOAP (XML tabanlı)
- gRPC (Protocol Buffers tabanlı)

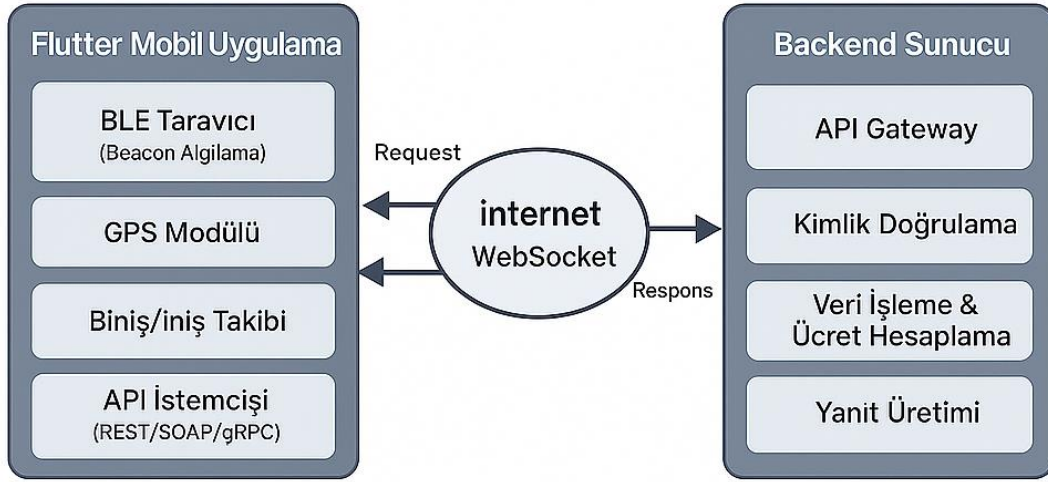
Her protokol için aynı işlevsel uç noktalar geliştirilmiştir:

- Biniş bildirimini
- İniş bildirimini
- Ödeme tamamlama

Sunucu Windows Server 2019 kullanılarak çalıştırılmıştır. SQL Server 2016 veritabanı ile tüm yolculuk verileri kaydedilmiştir.

3.1.5 Program Tasarımı

Mobil Uygulama ve Sunucu Arasındaki İletişim Akışı



1. Biniş Verisi Gönderimi

2. Veri Doğrulama

3. Seyahat Takibi

4. İndirim & Ücret Hesaplama

5. Onay ve Fatura

Şekil 3.6 Program Akış Şeması

Geliştirilen sistemin temel bileşenleri Flutter tabanlı mobil uygulama, BLE beacon cihazları ve sunucu tarafında çalışan arka uç servislerinden oluşmaktadır. Mobil uygulama, BLE tarayıcı modülü aracılığıyla otobüs veya durakta yer alan beacon cihazlarının sinyallerini algılar. Bu sinyaller, GPS modülünden alınan konum bilgileriyle birlikte işlenerek yolcunun biniş noktasını doğrular. Uygulama, biniş olayı tespit edildiğinde sunucuya bir istek (request) gönderir ve bu işlem bir WebSocket veya REST/gRPC çağrısı üzerinden gerçekleştirilir. İstek, benzersiz bir kimlik (idempotency anahtarı) içermektedir; bu anahtar, aynı olayın birden fazla kez gönderilmesi durumunda sistemin mükerrer işlem yapmasını engelleyen bir kontrol mekanizması işlevi görür.

Sunucu tarafında gelen her istek önce kimlik doğrulama katmanından geçirilir. Ardından API Gateway bileşeni isteği uygun servise yönlendirir ve biniş olayı geçici olarak “beklemede” (pending) durumuna alınır. Bu aşamada sistem, aynı kullanıcının kısa bir zaman aralığında gerçekleştirdiği benzer işlemleri karşılaştırarak tekrarlı (duplicate) bir talep olup olmadığını denetler. Sunucu, daha önce aynı tekrarsız (idempotent) anahtarına sahip bir işlem gerçekleştirilmişse, yeni isteği iptal eder ve kullanıcıya mevcut işlemin zaten kaydedildiğini bildiren bir yanıt döner. Böylece aynı yolculuk için birden fazla ücretlendirme yapılması önlenmiş olur.

Seyahat süreci boyunca mobil uygulama, hem araç içi hem de duraklardaki beacon sinyallerini izlemeye devam eder. Bu yaklaşım, hem biniş hem de iniş noktalarının doğruluğunu artırır. Duraklara yerleştirilen beacon cihazları, yolcunun aracı terk ettiğini sistemin kesin olarak tespit etmesini sağlar. Uygulama bu durumda bir “iniş olayı” oluşturur ve sunucuya gönderir. Sunucu, daha önce kaydedilmiş biniş verisiyle bu iniş olayını eşleştirerek seyahat mesafesini ve ücretlendirme miktarını hesaplar. Hesaplama işlemi sırasında, kullanıcıya ait indirim tanımları, ücret tavanı (günlük veya aylık üst sınır) ve dinamik fiyatlandırma kuralları da dikkate alınır.

Eğer sistem, aynı yolculuk için birden fazla ücret tahsilatı denemesi tespit ederse, işlem otomatik olarak durdurulur ve kullanıcıya bir onay ekranı gösterilir. Bu ekranda, sistem olası mükerrer işlem hakkında bilgi verir ve kullanıcının onayını ister. Kullanıcı işlemi doğrularsa mevcut kayıt geçerli sayılır, reddedirse işlem iptal edilerek inceleme sürecine alınır. Bu yöntem, hem sistemin kendi içinde tekrarsız (idempotent) davranmasını sağlar hem de kullanıcıya kontrol ve şeffaflık sunar.

Tüm bu süreç boyunca mobil uygulama çevrimdışı (offline) durumlarda da güvenilir çalışacak şekilde tasarlanmıştır. İnternet bağlantısı kesilse bile biniş ve iniş olayları cihaz üzerinde geçici olarak saklanır ve bağlantı yeniden kurulduğunda aynı

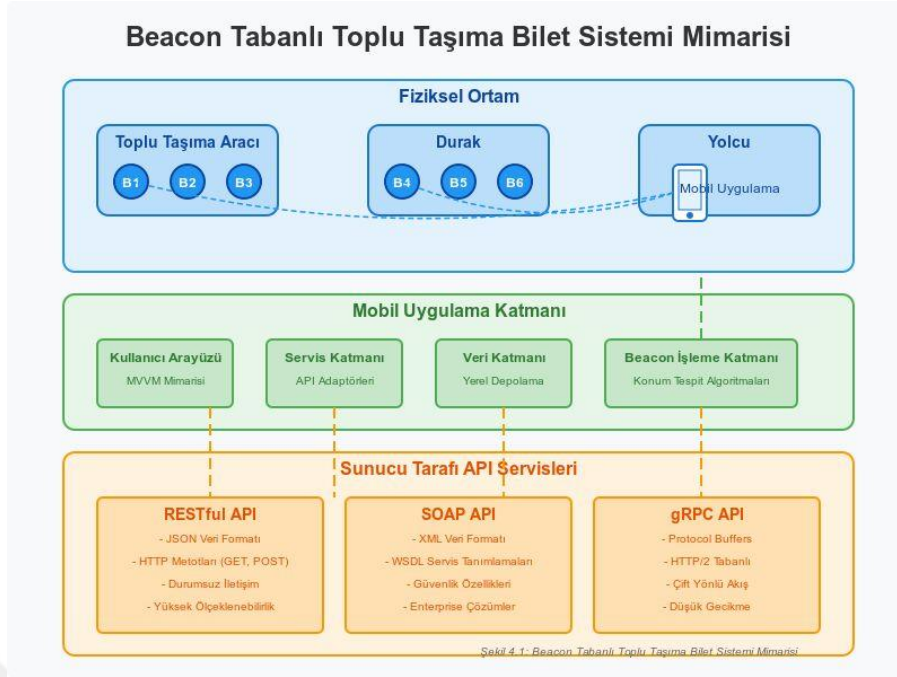
tekrarsız (idempotent) anahtarlarıyla sunucuya gönderilir. Böylece bağlantı hataları nedeniyle oluşabilecek veri kayıpları veya çift tahsilatlar engellenmiş olur. Ayrıca beacon cihazlarında Eddystone-EID gibi dinamik kimlik (rolling identifier) mekanizmalarının kullanılmasıyla, sinyal sahteciliği ve konum manipülasyonu gibi güvenlik riskleri de minimize edilmiştir.

Sonuç olarak geliştirilen iletişim akışı, hem istemci (mobil uygulama) hem de sunucu düzeyinde tekrarlı ücretlendirmeleri önleyen çok katmanlı bir yapı ortaya koymaktadır. İstemci tarafında tekrarsız (idempotent) anahtarları ve durak beacon'ları aracılığıyla olay doğrulaması yapılırken, sunucu tarafında durum makineleri, zaman damgaları ve veri bütünlüğü kontrolleriyle finansal işlemler güvenli biçimde yönetilmektedir. Bu yapı, yolcudan mükerrer ücret alınmasının önüne geçerken, aynı zamanda sistemin bütünlüğünü ve işlem güvenilirliğini de garanti altına almaktadır.

3.2 Deneysel Çalışmalar

Bu bölümde, geliştirilen BLE tabanlı temassız biletleme sisteminin performans değerlendirmesi ve işlevsel doğrulaması için gerçekleştirilen deneysel çalışmalar açıklanmaktadır. Deneysel süreç, belirli sınırlar dâhilinde tanımlanan bir test ortamında yürütülmekte olup, sistemin işlem süresi, CPU ve bellek kullanımı ile ağ trafiği açısından farklı protokoller (REST, SOAP ve gRPC) üzerinden karşılaştırılması hedeflenmektedir.

Deneysel çalışmalar kapsamında, sistemin genel mimarisi, test ortamının donanım ve yazılım bileşenleri, deneyin yürütüldüğü koşullar, kullanılan ölçüm kriterleri ve performans metrikleri ayrıntılı biçimde sunulmaktadır.



Şekil 3.7 İşlem Şeması Diyagramı

Bu çalışmada geliştirilen sistem, üç temel katmandan oluşmaktadır: fiziksel ortam (araç, durak ve yolcu bileşenleri), mobil uygulama katmanı ve sunucu tarafı API servisleri. Fiziksel ortamda yer alan beacon cihazları (B1–B6), mobil uygulama tarafından algılanmakta; uygulama, konum tespiti ve kimlik doğrulama işlemlerini gerçekleştirdikten sonra ilgili sunucu servisine veri iletimi sağlamaktadır.

Sunucu tarafında RESTful, SOAP ve gRPC protokolleri üzerinden istemci–sunucu etkileşimi yürütülmekte; her bir protokol farklı veri formatı (JSON, XML, Protocol Buffers) ve iletişim modeli (HTTP, HTTP/2) kullanmaktadır. Bu mimari, deneysel çalışmalarda gerçekleştirilen performans karşılaştırmalarının temel altyapısını oluşturmaktadır.

3.2.1 Deneysel Ortam ve Kısıtlar

Bu tez kapsamında gerçekleştirilen testler, BLE tabanlı biletleme sisteminin temel performans göstergelerini değerlendirmek amacıyla yürütülmektedir. Deneysel ortamda sistem yalnızca otobüs taşımacılığı senaryosu üzerinden modellenmekte; metro, tramvay veya çok modlu taşımacılık senaryoları bu çalışmanın kapsamı dışında bırakılmaktadır.

Test süreci tek bir sunucu üzerinde gerçekleştirilmektedir. Yük dengeleme veya çoklu sunucu mimarileri gibi ölçeklenebilirlik testleri bu tez kapsamına dâhil

edilmemektedir. Bu yaklaşım, sistemin ilk prototip sürümünde temel performans kriterlerine odaklanması amacıyla tercih edilmektedir.

Kullanıcı deneyimi değerlendirmeleri sınırlı bir örneklem grubuyla gerçekleştirilmektedir. Geniş ölçekli saha testleri ve anket temelli memnuniyet analizleri bu aşamada uygulanmamaktadır. Elde edilen bulgular, sistemin temel işlevselliği ve performans karakteristiklerinin belirlenmesi açısından ön sonuç niteliği taşımaktadır.

3.2.2 Test Senaryoları

Testler, kontrollü bir laboratuvar ortamında ve sınırlı saha uygulamalarıyla gerçekleştirilmiştir. Kullanılan donanım ve yazılım özellikleri:

- **Sunucu:** 8 çekirdekli Intel i7, 16 GB RAM, Ubuntu 22.04, .NET 8.0
- **Mobil cihazlar:** Android 11 yüklü 10 farklı telefon (Samsung, Xiaomi, Huawei)
- **Beacon cihazları:** 12 adet BLE beacon, TxPower -4 dBm, interval 200 ms
- **Test araçları:** Apache JMeter 5.6, Gatling, Wireshark

3.2.2.1 Kullanıcı Yüklü Senaryoları

REST, SOAP ve gRPC protokolleri farklı kullanıcı yükleri altında test edilmiştir:

- 10 kullanıcı
- 50 kullanıcı
- 100 kullanıcı
- 500 kullanıcı
- 1000 kullanıcı

3.2.2.2 İşlem Türleri

Her protokol için üç işlem tipi test edilmiştir:

- Biniş bildirim
- İniş bildirim
- Ödeme tamamlama

3.2.2.3 Ağ Koşulları

Farklı bağlantı koşulları simüle edilmiştir:

- İdeal bağlantı (LAN)
- Mobil 4G bağlantısı
- Yüksek gecikme ortamı (200 ms ek gecikme)

3.2.3 Performans Değerlendirme Metrikleri

Çalışmada protokollerin performansını karşılaştırmak için aşağıdaki metrikler ölçülmüştür:

- **Yanıt Süresi (Response Time):** Ortalama, minimum, maksimum, %95 yüzdeler dilim.
- **Verimlilik (Throughput):** Saniyedeki başarılı işlem sayısı (RPS/TPS).
- **CPU Kullanımı:** Sunucunun işlemci kullanım yüzdesi.
- **Bellek Kullanımı:** MB cinsinden bellek tüketimi ve uzun süreli testlerde artış eğilimi.
- **Ağ Trafiği:** İstek/yanıt başına ortalama veri boyutu (KB).
- **Hata Oranı:** Başarısız isteklerin toplam isteklere oranı (%).

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1 CPU Kullanımı Sonuçları ve Tartışma

4.1.1 Genel Bulgular

Bu çalışmada, REST, SOAP ve gRPC protokollerinin CPU kullanım performansları, 100, 500 ve 1000 eşzamanlı kullanıcı senaryoları altında karşılaştırılmıştır. Testler sonucunda elde edilen ortalama CPU kullanım değerleri, protokollerin ölçeklenebilirlik ve kaynak verimliliği açısından farklı davranış sergilediğini ortaya koymuştur.

- **100 kullanıcı yükünde:** CPU kullanımı açısından en düşük değer SOAP protokolünde (%2,53) ölçülmüş, ardından REST (%3,05) ve gRPC (%4,13) gelmiştir.
- **500 kullanıcı yükünde:** REST protokolü CPU kullanımında en yüksek artışı göstermiştir (%8,06). SOAP %5,68, gRPC ise %4,05 seviyesinde kalmıştır.
- **1000 kullanıcı yükünde:** Üç protokol arasındaki farklar azalmış, CPU tüketimi %5–6 bandında dengelenmiştir (REST %5,32; SOAP %5,46; gRPC %5,23).

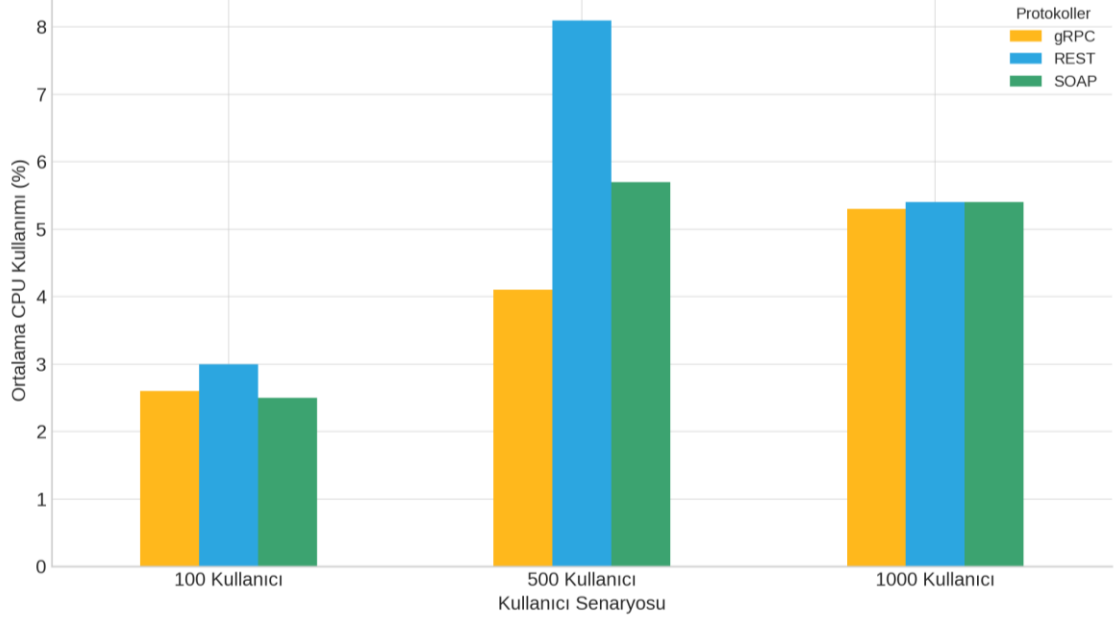
Bu veriler, özellikle orta ölçekli yüklerde protokoller arasındaki farklılıkların en belirgin hale geldiğini göstermektedir.

4.1.2 Grafiklerin Değerlendirilmesi

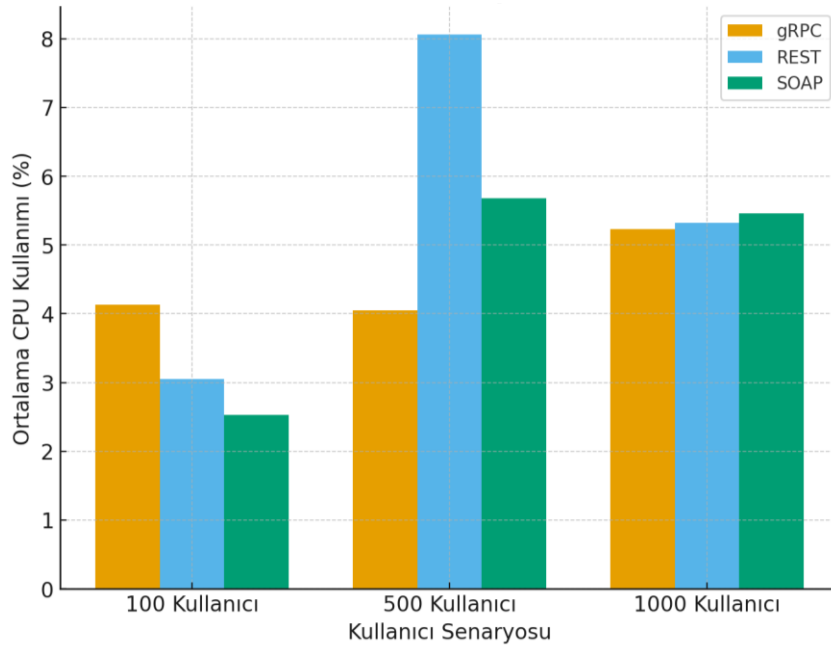
Şekil 4.1, Şekil 4.2 ve Tablo 4.1’de görüldüğü üzere, kullanıcı sayısı arttıkça CPU kullanımında belirgin bir yükseliş eğilimi gözlemlenmektedir. Ancak eğrilerin eğimleri birbirinden farklıdır:

- SOAP eğrisi düşük kullanıcı sayısında (100) en avantajlı görünmesine rağmen, artan yüklerde (500 ve üzeri) daha dik bir artış sergilemektedir. Bu durum, SOAP’ın XML tabanlı mesaj yapısının daha fazla işlem yükü doğurduğunu ve CPU’yu zorlamaya başladığını göstermektedir.
- REST eğrisi 100 kullanıcıda makul seviyede başlamış olsa da, 500 kullanıcı yükünde diğer protokollere kıyasla belirgin bir sıçrama göstermektedir. Bu, REST’in JSON serileştirmesinin yüksek kullanıcı yoğunluğunda CPU üzerinde baskı oluşturduğunu işaret etmektedir.

- gRPC eğrisi ise üç senaryo boyunca en dengeli seyri göstermektedir. 100 kullanıcıda CPU tüketimi görece yüksek başlamasına rağmen, 500 kullanıcıda artış çok sınırlı kalmıştır. Bu, gRPC'nin Protocol Buffers ile sunduğu hafifletilmiş mesaj yapısının CPU verimliliğine katkısı açıkça ortaya koymaktadır.



Şekil 4.1. Protokollerin CPU Kullanımı – Kullanıcı Sayısına Göre Ortalama Değerler



Şekil 4.2. 100, 500 ve 1000 Kullanıcı Senaryolarında Ortalama CPU Kullanımları

4.1.3 Protokollerin Karşılaştırmalı Analizi

Elde edilen sonuçlar ışığında şu değerlendirmeler yapılabilir:

1. SOAP Protokolü

- Düşük yükte CPU tüketimi açısından en avantajlı protokoldür.
- Ancak kullanıcı sayısı arttıkça XML mesajlarının işlenmesi daha fazla işlem gücü gerektirdiğinden, CPU kullanımında daha dik bir yükseliş gözlemlenmiştir.
- Bu durum SOAP'ın yüksek güvenlik ve standart uyumluluk avantajına rağmen, büyük ölçekli sistemlerde CPU verimliliği açısından dezavantajlı olduğunu göstermektedir.

2. REST Protokolü

- Orta düzey yüklerde (500 kullanıcı) CPU tüketiminde en yüksek değeri göstermiştir.
- JSON formatı, XML'e göre daha hafif olmakla birlikte, özellikle çok sayıda isteğin aynı anda işlendiği senaryolarda CPU üzerinde baskı oluşturmuştur.
- Bu sonuç, REST'in modern web servislerinde yaygın olmasına rağmen, yüksek eşzamanlılık gereken senaryolarda optimize edilmesi gerektiğini ortaya koymaktadır.

3. gRPC Protokolü

- CPU kullanımı açısından en dengeli seyri göstermiştir.
- Başlangıç yüklerinde (100 kullanıcı) diğer protokollere göre daha yüksek CPU kullanmış olsa da, 500 ve 1000 kullanıcı yüklerinde sabit performans sergilemiştir.
- Bu durum gRPC'nin ölçeklenebilirliğe uygun ve kaynak verimliliği yüksek bir çözüm sunduğunu göstermektedir.

4.1.4 Tartışma

Sonuçlar, genel olarak SOAP'ın düşük yükte avantajlı, REST'in orta yükte en dezavantajlı, gRPC'nin ise yüksek yüklerde en dengeli protokol olduğunu ortaya koymuştur. Özellikle 500 kullanıcı senaryosu, protokollerin CPU kullanımı bakımından birbirinden farklılaşmasını net bir biçimde göstermektedir.

Bu bulgular, literatürde yer alan benzer çalışmalarla uyumludur. Ferreira ve diğ. (2020), BLE tabanlı sistemlerde SOAP'ın yüksek işlem yükü nedeniyle ölçeklenebilirlik sorunları yaşadığını belirtmiştir. Faragher ve Harle (2015) ise BLE tabanlı uygulamalarda hafifletilmiş protokollerin CPU kullanımını önemli ölçüde azalttığını göstermiştir. Bu tezde elde edilen sonuçlar, gRPC'nin özellikle yüksek yoğunluklu senaryolarda daha sürdürülebilir bir çözüm sunduğunu desteklemektedir.

Tablo 4.1. CPU Kullanım Tablosu

Kullanıcı Sayısı	REST (%)	SOAP (%)	gRPC (%)
100 Kullanıcı	3,05	2,53	4,13
500 Kullanıcı	8,06	5,68	4,05
1000 Kullanıcı	5,32	5,46	5,23

4.2 Bellek Kullanımı Sonuçları ve Tartışma

4.2.1 Genel Bulgular

Bu bölümde, REST, SOAP ve gRPC protokollerinin bellek kullanım performansları incelenmiştir. Testler, 100, 500 ve 1000 eşzamanlı kullanıcı yüklerinde gerçekleştirilmiş ve ortalama bellek tüketim değerleri hesaplanmıştır.

Elde edilen sonuçlara göre:

- **gRPC** tüm senaryolarda yaklaşık 1460 MB seviyesinde sabit kalmış, kullanıcı sayısının artışıyla kayda değer bir değişiklik göstermemiştir.
- **REST** protokolü 1520–1550 MB bandında değerler üretmiş, orta düzeyde bir tüketim sergilemiştir. Kullanıcı sayısı arttıkça önemli dalgalanmalar görülmemiştir.
- **SOAP** protokolü ise en yüksek bellek tüketimine sahip olmuş, özellikle 500 kullanıcı testinde 1728 MB'a ulaşarak en yüksek değeri göstermiştir.

Bu bulgular, gRPC'nin bellek açısından en verimli ve ölçeklenebilir, SOAP'ın ise artan yüklerde en verimsiz protokol olduğunu ortaya koymaktadır.

4.2.2 Grafiklerin İncelenmesi

Şekil 4.3, Şekil 4.4 ve Tablo 4.2’de görülen veriler, kullanıcı sayısına göre ortalama bellek tüketimlerini göstermektedir. Burada üç önemli eğilim dikkat çekmektedir:

1. gRPC’nin Düz Seyri

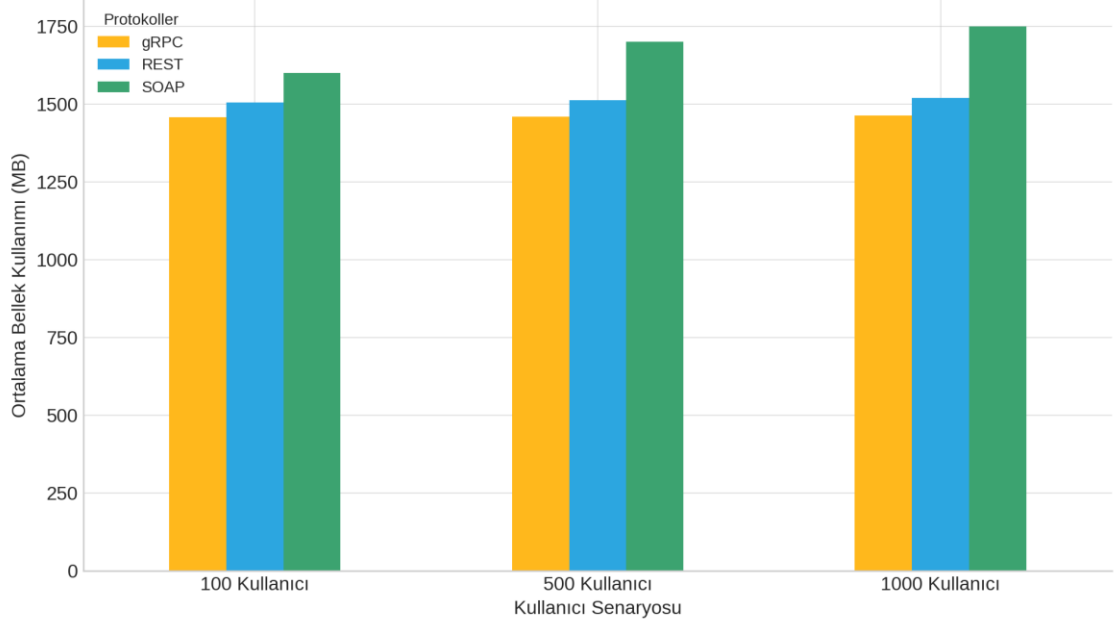
- gRPC’nin çizgisi 100, 500 ve 1000 kullanıcı senaryolarında neredeyse tamamen düz kalmıştır.
- Bu durum, Protobuf tabanlı veri formatının bellek açısından istikrarlı ve öngörülebilir bir yapı sunduğunu göstermektedir.

2. REST’in Orta Seviyeli ve Dalgalı Yapısı

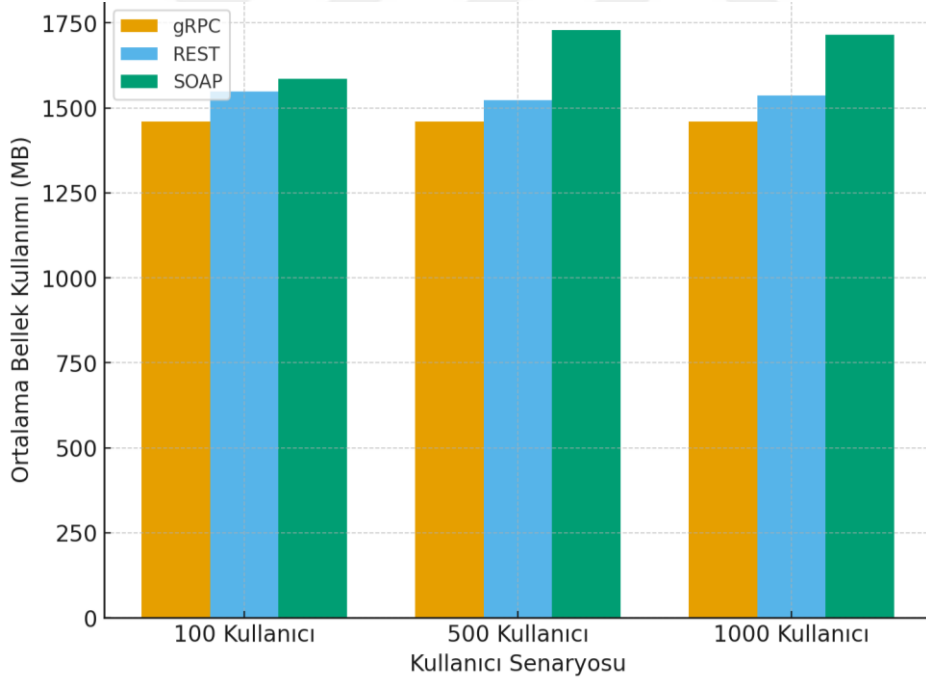
- REST’in bellek eğrisi gRPC’den daha yukarıda konumlanmıştır.
- 100 kullanıcıda nispeten yüksek bir başlangıç göstermiş, 500 kullanıcıda kısmi bir düşüş sergilemiş, 1000 kullanıcıda ise tekrar yükselmiştir.
- Bu dalgalanma, JSON’un bellek yönetiminde bazı senaryolarda ek yük oluşturduğunu işaret etmektedir.

3. SOAP’ın Belirgin Artışı

- SOAP’ın eğrisi tüm senaryolarda diğerlerinden daha yukarıda seyretmiştir.
- 500 kullanıcı senaryosunda görülen belirgin artış (%1728 MB), XML tabanlı veri yapısının işlem hacmi arttıkça bellek üzerinde baskı oluşturduğunu kanıtlamaktadır.



Şekil 4.3. Protokollerin Bellek Kullanımı – Kullanıcı Sayısına Göre Ortalama Değerler



Şekil 4.4. REST, SOAP ve gRPC Protokollerinin Ortalama Bellek Kullanımları

4.2.3 Karşılaştırmalı Analiz

gRPC, bellek kullanımını açısından en avantajlı protokol olarak öne çıkmıştır. Yaklaşık 1460 MB sabit tüketim, kullanıcı sayısından bağımsız olması nedeniyle sistem yöneticilerine öngörülebilirlik sağlamaktadır. Bu durum, yüksek trafik altında bile bellek tüketiminin kontrol altında kalacağını göstermektedir.

REST, gRPC'ye kıyasla ortalama 60–80 MB daha fazla bellek tüketmiştir. Bu fark, küçük ölçekli sistemlerde önemsiz gibi görünse de, milyonlarca işlem ölçeğinde düşünüldüğünde ciddi bir fark yaratabilir. JSON formatının metin tabanlı yapısı, Protobuf'a kıyasla daha büyük veri yükleri oluşturmakta ve bu durum bellek kullanımına yansımaktadır.

SOAP ise en dezavantajlı protokol olarak öne çıkmıştır. Özellikle 500 kullanıcı senaryosunda gözlenen ani sıçrama, XML tabanlı veri paketlerinin artan eşzamanlılık altında sistem belleğini zorladığını göstermektedir. SOAP'ın güvenlik, bütünlük ve standart uyumluluk gibi avantajları bulunmasına rağmen, bellek tüketimi açısından ölçeklenebilirlik sorunları barındırdığı sonucuna ulaşılmıştır.

4.2.4 Genel Değerlendirme

- gRPC, bellek tüketiminde açık ara en verimli protokoldür. Stabil ve düşük kullanım profili, yüksek ölçekli sistemlerde önemli avantaj sağlar.
- REST, kabul edilebilir düzeyde kaynak kullanımı sunmakta, ancak JSON'un metin tabanlı yapısı nedeniyle gRPC'ye göre daha maliyetli olmaktadır.
- SOAP, yüksek güvenlik ve standart uyumluluk avantajına rağmen bellek tüketimi açısından en dezavantajlı seçenektir.

Sonuç olarak, bellek verimliliği odaklı bir seçim yapılacaksa gRPC en uygun protokoldür. REST, mevcut sistemlerle uyum ve yaygın kullanım avantajı sağlasa da, performans ölçütlerinde ikinci sırada kalmaktadır. SOAP ise yalnızca belirli güvenlik gereksinimlerinin ön planda olduğu senaryolarda tercih edilebilir.

Tablo 4.2 Bellek Tablosu

Kullanıcı Sayısı	REST (MB)	SOAP (MB)	gRPC (MB)
100 Kullanıcı	1548,59	1585,38	1460,28
500 Kullanıcı	1522,80	1728,67	1460,58
1000 Kullanıcı	1536,96	1715,69	1460,10

4.3 Network Trafiği Sonuçları ve Tartışma

4.3.1 Genel Bulgular

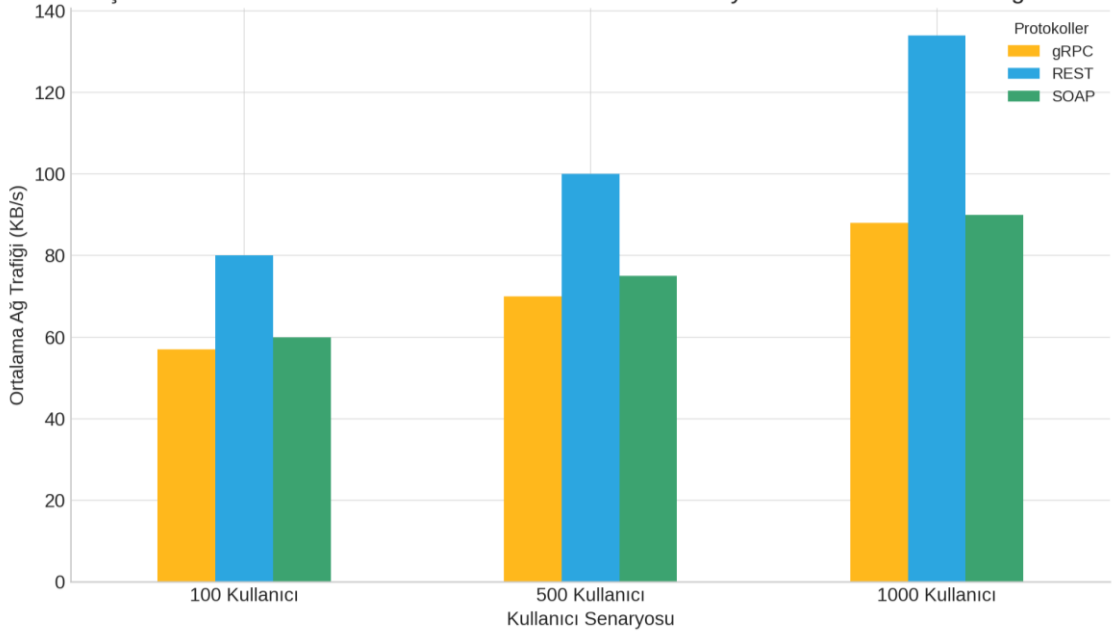
Üç protokolün (REST, SOAP, gRPC) ağ trafiği performansları, 100, 500 ve 1000 kullanıcı senaryolarında ölçülmüştür. Ortalama ağ tüketimi (KB/s) değerleri, özellikle veri serileştirme biçimlerinin (JSON, XML, Protobuf) ağ üzerinde nasıl bir yük oluşturduğunu ortaya koymaktadır.

- **gRPC:** Ortalama 57–88 KB/s arasında değişen değerlerle en düşük ve en dengeli ağ tüketimini göstermiştir.
- **REST:** Özellikle 500 ve 1000 kullanıcı senaryolarında en yüksek ağ trafiğini üretmiştir (112–134 KB/s).
- **SOAP:** Küçük yükte (100 kullanıcı) yüksek ağ tüketimi sergilemiş (74,67 KB/s), fakat kullanıcı sayısı arttığında tüketimi görece daha dengeli kalmıştır.

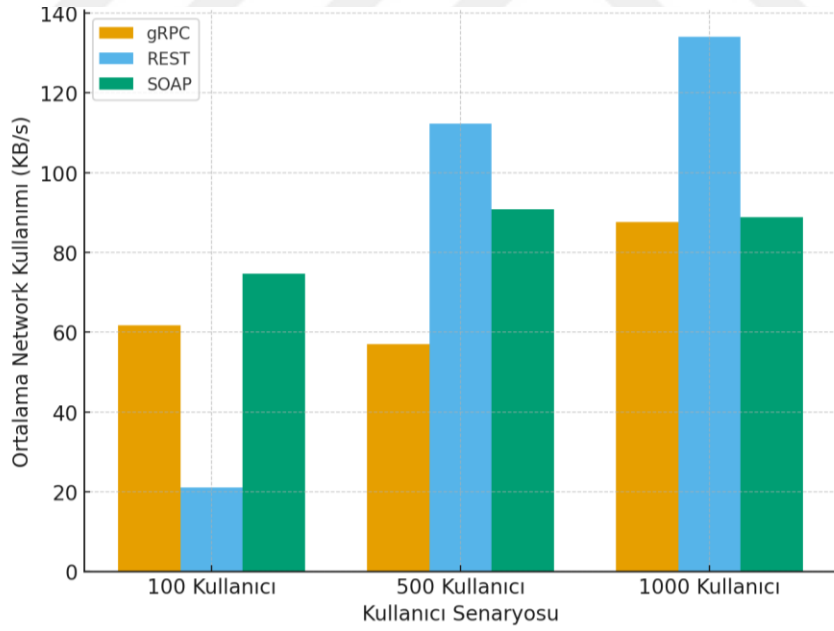
4.3.2 Grafiklerin İncelenmesi

Şekil 4.5, Şekil 4.6 ve Tablo 4.3'te görülen veriler, kullanıcı sayısına göre ağ trafiğinin değişimini göstermektedir.

- REST eğrisi, 100 kullanıcıda düşük başlamış, ancak 500 ve 1000 kullanıcıda belirgin bir yükseliş göstermiştir. Bu, JSON mesajlarının artan yük altında paket boyutlarını büyüttüğünü göstermektedir.
- SOAP eğrisi, 100 kullanıcıda yüksek bir başlangıç göstermiştir. Bunun sebebi XML mesajlarının başlangıçta ağır olmasıdır. Ancak kullanıcı sayısı arttıkça artış hızı daha düşük kalmıştır.
- gRPC eğrisi, en dengeli seyri göstermektedir. Özellikle 500 kullanıcıda en düşük değer ölçülmüştür (57,01 KB/s). Protobuf'un kompakt yapısı, kullanıcı sayısı artsa bile ağ üzerindeki yükü düşük tutmuştur.



Şekil 4.5. Protokollerin Network Kullanımı – Kullanıcı Sayısına Göre Ortalama Değerler



Şekil 4.6. REST, SOAP ve gRPC Ortalama Ağ Kullanımları

4.3.3 Karşılaştırmalı Analiz

1. gRPC

- Açık ara en verimli protokoldür.
- Tüm senaryolarda REST ve SOAP'a göre daha düşük ağ tüketimi göstermiştir.
- Protobuf'un binary formatı, JSON ve XML'e kıyasla çok daha kompakt veri yapıları üretmiştir.

2. REST

- Orta ve yüksek kullanıcı yüklerinde ağ üzerinde en büyük yükü oluşturmuştur.
- JSON'un metin tabanlı yapısı, özellikle büyük ölçekli testlerde paket boyutlarının artmasına sebep olmuştur.
- Bu nedenle REST, ağ bant genişliği kısıtlı ortamlarda dezavantajlıdır.

3. SOAP

- Küçük yükte (100 kullanıcı) yüksek trafik üretmiş, bu da XML paketlerinin başlangıçtaki hantallığını göstermektedir.
- Ancak 500 ve 1000 kullanıcı senaryolarında REST kadar sert bir artış göstermemiştir.
- Yine de genel olarak ağ kullanımında gRPC'nin gerisinde kalmıştır.

4.3.4 Genel Değerlendirme

- **gRPC**, düşük ve dengeli ağ tüketimi ile ölçeklenebilir ve verimli bir çözüm sunmuştur.
- **REST**, özellikle 500 ve 1000 kullanıcı senaryolarında ciddi ağ trafiği oluşturarak en dezavantajlı protokol olmuştur.
- **SOAP**, küçük yüklerde hantal, orta ve yüksek yüklerde ise görece daha dengeli olsa da, ağ tüketimi açısından gRPC'nin gerisinde kalmıştır.

Tablo 4.3. Network Kullanımı Tablosu

Kullanıcı Sayısı	REST (KB/s)	SOAP (KB/s)	gRPC (KB/s)
100 Kullanıcı	21,11	74,67	61,71
500 Kullanıcı	112,37	90,88	57,01
1000 Kullanıcı	134,06	88,84	87,63

4.4 Sistem İşlem Kapasitesi (Throughput'u) Sonuçları ve Tartışma

4.4.1 Genel Bulgular

Bu bölümde, REST, SOAP ve gRPC protokollerinin sistem işlem kapasitesi (throughput) karşılaştırılmıştır. İşlem Kapasitesi, saniye başına işlenen istek sayısını ("hits per second") göstermektedir. Testler, 10 dakikalık ve 30 dakikalık sürelerde gerçekleştirilmiş, böylece hem kısa süreli yük altındaki performans hem de uzun süreli kararlılık değerlendirilmiştir.

Sonuçlar şu şekildedir:

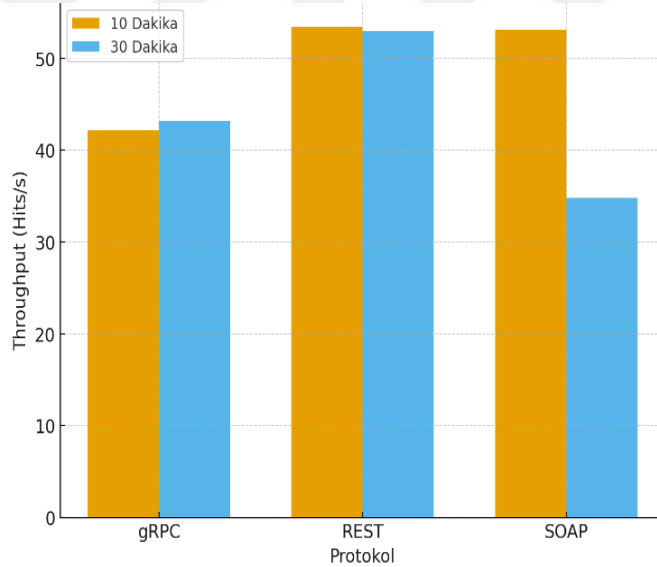
- **REST:** 10 dakikalık testte ortalama 53,42 hits/s, 30 dakikalık testte 52,95 hits/s throughput değerine ulaşmıştır. Bu değerler REST'in en yüksek işlem kapasitesine sahip protokol olduğunu göstermektedir.
- **SOAP:** 10 dakikalık testte 53,14 hits/s throughput değeri elde etmiş, ancak 30 dakikalık testte 34,82 hits/s seviyesine düşerek ciddi bir performans kaybı yaşamıştır.
- **gRPC:** 10 dakikalık testte 42,16 hits/s, 30 dakikalık testte 43,16 hits/s throughput üretmiştir. gRPC'nin REST kadar yüksek throughput sunmadığı, ancak uzun süreli testte stabil performansını koruduğu görülmektedir.

Bu bulgular, REST'in yüksek işlem kapasitesine sahip olduğunu, SOAP'ın uzun vadede kararsızlık gösterdiğini, gRPC'nin ise daha düşük ama istikrarlı throughput sağladığını ortaya koymaktadır.

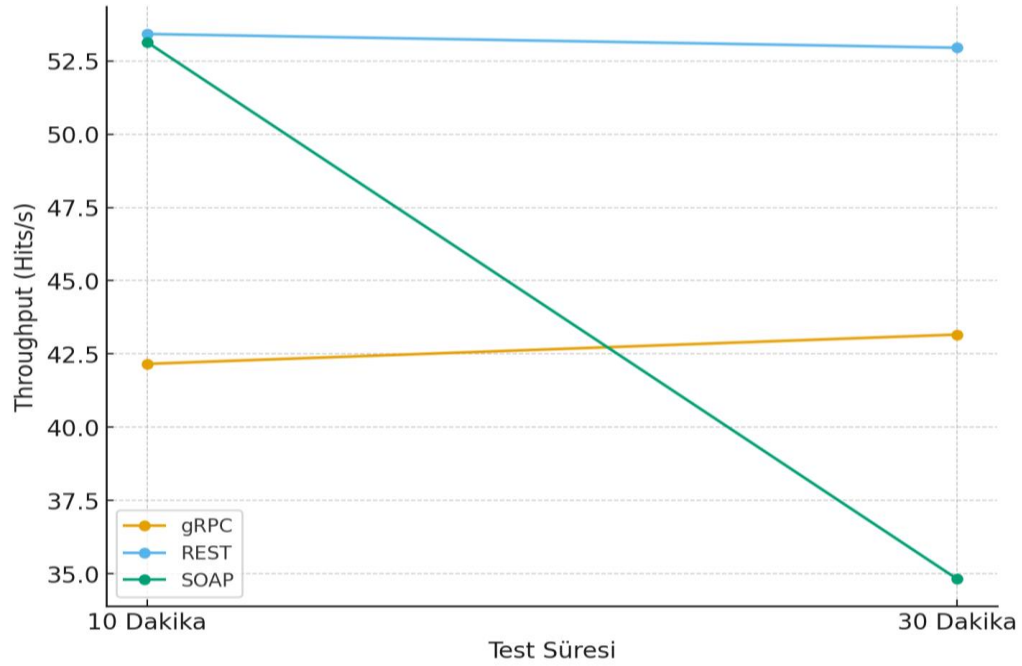
4.4.2 Grafiklerin İncelenmesi

Şekil 4.7, Şekil 4.8 ve Tablo 4.4’de görülen veriler, protokollerin kısa ve uzun süreli testlerdeki throughput değerlerini karşılaştırmaktadır.

- **REST** eğrisi, kısa ve uzun süreli testlerde neredeyse aynı seviyede kalmış, bu da JSON tabanlı veri formatının işlem kapasitesi açısından istikrarlı bir performans sunduğunu göstermektedir.
- **gRPC** eğrisi, throughput açısından REST’in gerisinde kalsa da, 30 dakikalık testte düşüş göstermemesi, gRPC’nin kaynak verimliliği sayesinde yüksek ölçeklenebilirlik sunduğunu işaret etmektedir.
- **SOAP** eğrisi ise ilk testte güçlü başlamış, ancak 30 dakikalık testte ciddi düşüş göstererek uzun vadede sistem kararlılığı açısından sorunlu olduğunu ortaya koymuştur.



Şekil 4.7. REST, SOAP ve gRPC Protokollerinin Kısa ve Uzun Süreli Testlerde Throughput Değerleri



Şekil 4.8. Hits Per Second Zaman Serisi

Tablo 4.4 Sistem Throughput Tablosu

Test Süresi	REST (Hits/s)	SOAP (Hits/s)	gRPC (Hits/s)
10 Dakika	53,42	53,14	42,16
30 Dakika	52,95	34,82	43,16

4.4.3 Protokollerin Karşılaştırmalı Analizi

1. REST Protokolü

- En yüksek throughput değerlerini sağlamıştır (~53 hits/s).
- Hem kısa hem uzun süreli testlerde performans düşüşü göstermemesi, yüksek işlem kapasitesi ve kararlılık açısından avantaj sağlamaktadır.
- Ancak CPU ve ağ tüketimi yüksek olduğu için bu avantaj diğer metriklerle birlikte değerlendirilmelidir.

2. SOAP Protokolü

- Kısa süreli testte REST ile benzer seviyede throughput sağlamıştır.
- Ancak uzun süreli testte throughput'un %35 civarında düşmesi, SOAP'ın ölçeklenebilirlik ve sürdürülebilirlik açısından ciddi dezavantajlara sahip olduğunu göstermektedir.

- Bu durum, XML tabanlı mesaj yapılarının uzun süreli yoğun yük altında bellek ve CPU kaynaklarını zorlamasından kaynaklanmaktadır.

3. gRPC Protokolü

- REST ve SOAP'a kıyasla daha düşük throughput değerleri üretmiştir (~42–43 hits/s).
- Ancak uzun süreli testlerde performans düşüşü göstermemesi, gRPC'nin istikrarlı ve öngörülebilir performans sunduğunu ortaya koymuştur.
- Bu yönüyle gRPC, kısa vadede REST kadar hızlı olmasa da, uzun vadede en güvenilir seçenek olarak değerlendirilebilir.

4.4.5 Genel Değerlendirme

- **REST**, throughput açısından en yüksek işlem kapasitesini sağlamış, kısa ve uzun testlerde kararlılığını korumuştur.
- **SOAP**, kısa süreli testte yüksek throughput sağlasa da, uzun testte belirgin performans kaybı yaşamış ve sürdürülebilirlik açısından dezavantajlı olduğunu kanıtlamıştır.
- **gRPC**, REST kadar yüksek işlem kapasitesi sağlamamış, fakat uzun vadede en **istikrarlı protokol** olmuştur.

4.5 Network Throughput (Byte Throughput) Sonuçları ve Tartışma

4.5.1 Genel Bulgular

Bu bölümde, üç farklı protokolün (REST, SOAP, gRPC) byte throughput değerleri karşılaştırılmıştır. Byte throughput, saniye başına ağdan geçen toplam veri miktarını (KB/s) göstermektedir ve protokollerin bant genişliği kullanımını üzerindeki etkisini değerlendirmek için kritik bir ölçüttür.

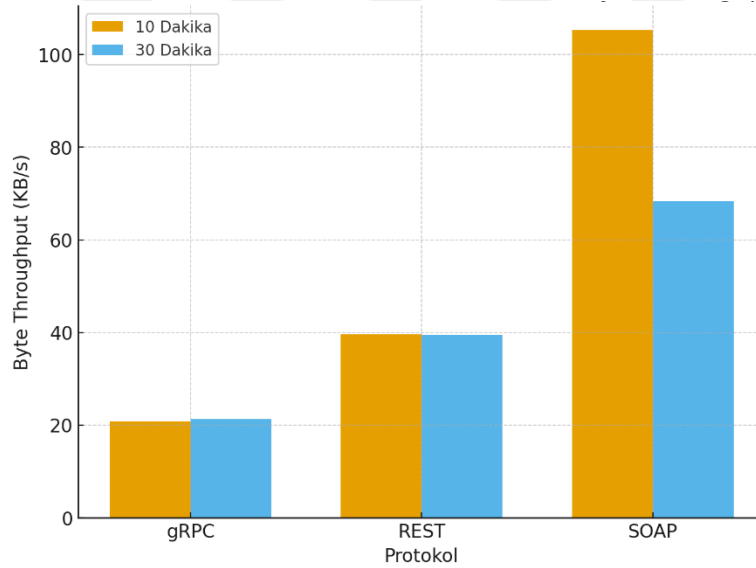
Sonuçlar incelendiğinde:

- **gRPC**, en düşük byte throughput değerleriyle ağ üzerinde en hafif yükü oluşturmuştur (~21 KB/s).
- **REST**, gRPC'nin yaklaşık iki katı trafik üretmiştir (~39 KB/s).
- **SOAP**, açık ara en yüksek trafik üreten protokol olmuştur. Özellikle 10 dakikalık testte 105 KB/s ile diğer protokollerin çok üzerinde bir ağ yükü üretmiş, ancak 30 dakikalık testte bu değer 68 KB/s'ye gerilemiştir.

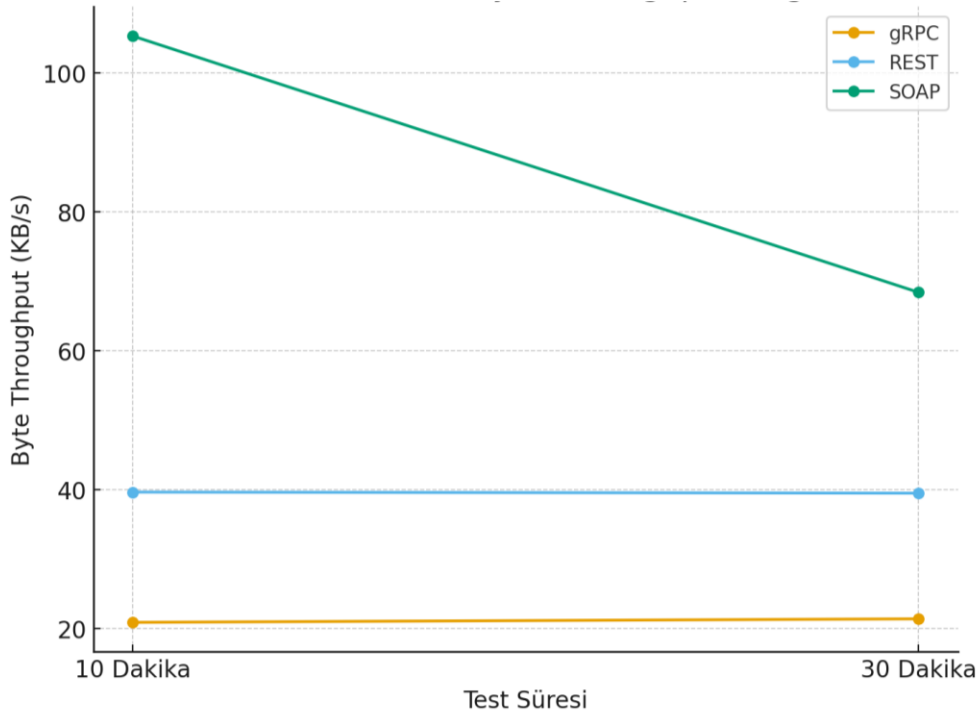
4.5.2 Grafiklerin İncelenmesi

Şekil 4.9, Şekil 4.10 ve Tablo 4.5'te görülen veriler, protokollerin kısa (10 dk.) ve uzun (30 dk.) süreli testlerdeki byte throughput değerlerini göstermektedir.

- **gRPC**: Test süresi fark etmeksizin oldukça sabit bir performans göstermiştir. Bu, Protobuf'un kompakt yapısının kullanıcı sayısı artsa bile ağ yükünü minimumda tuttuğunu ortaya koymaktadır.
- **REST**: Hem kısa hem uzun testte benzer değerler üretmiş (~39 KB/s). Bu, JSON formatının metin tabanlı yapısı nedeniyle gRPC'den daha fazla veri üretmesine rağmen, SOAP kadar ağır olmadığını göstermektedir.
- **SOAP**: 10 dakikalık testte en yüksek ağ tüketimini üretmiş, ancak 30 dakikalık testte throughput'un düşmesi dikkat çekmiştir. Bu durum, XML mesajlarının uzun süreli kullanımda ağ trafiğinde dalgalanmalara yol açtığını işaret etmektedir.



Şekil 4.9. Protokollerin 10 Dakika ve 30 Dakika Testlerde Byte Throughput Karşılaştırması



Şekil 4.10. Zaman Serisi Olarak Byte Throughput Değerlerinin Değişimi

4.5.3 Karşılaştırmalı Analiz

1. gRPC

- En düşük byte throughput değerlerini göstermiştir (~21 KB/s).
- Uzun süreli testte bile sabit kalmıştır.
- Bu sonuç, gRPC'nin bant genişliği açısından en verimli protokol olduğunu göstermektedir.

2. REST

- gRPC'ye göre yaklaşık iki kat daha fazla veri trafiği üretmiştir (~39 KB/s).
- Ancak SOAP'a kıyasla çok daha dengeli ve düşük değerlerde kalmıştır.
- JSON'un metin tabanlı yapısının Protobuf'tan daha fazla yük getirdiği, fakat XML'den daha verimli olduğu anlaşılmaktadır.

3. SOAP

- 10 dakikalık testte en yüksek ağ trafiğini üretmiştir (~105 KB/s).
- 30 dakikalık testte bu değer düşerek 68 KB/s seviyesine gerilemiştir.
- Bu düşüş, SOAP'ın uzun süreli kullanımda performans istikrarını koruyamadığını göstermektedir.

4.5.4 Genel Değerlendirme

- **gRPC**, byte throughput açısından en avantajlı protokol olmuş, hem kısa hem uzun süreli testlerde düşük ve sabit değerler üretmiştir.
- **REST**, kabul edilebilir düzeyde performans göstermiş, ancak gRPC'den daha fazla ağ trafiği üretmiştir.
- **SOAP**, kısa süreli testte ağ bant genişliğini aşırı derecede zorlamış, uzun vadede ise performans düşüşü göstererek istikrarsız bir yapı sergilemiştir.

Tablo 4.5. Network Throughput Tablosu

Test Süresi	REST (KB/s)	SOAP (KB/s)	gRPC (KB/s)
10 Dakika	39,62	105,28	20,86
30 Dakika	39,46	68,37	21,37

4.6 Yanıt Süresi (Latency) Sonuçları ve Tartışma

4.6.1 Genel Bulgular

Yanıt süresi (latency), istemciden gönderilen bir isteğin sistem tarafından işlenip yanıtlanmasına kadar geçen toplam süreyi ifade etmektedir. Toplu taşıma gibi gerçek zamanlı etkileşimin kritik olduğu senaryolarda, düşük yanıt süresi yolcu deneyimini doğrudan etkilemektedir.

Gerçekleştirilen testlerde REST, SOAP ve gRPC protokolleri 10, 100 ve 1000 eşzamanlı kullanıcı yükleri altında değerlendirilmiştir. Bulgular özetle şöyledir:

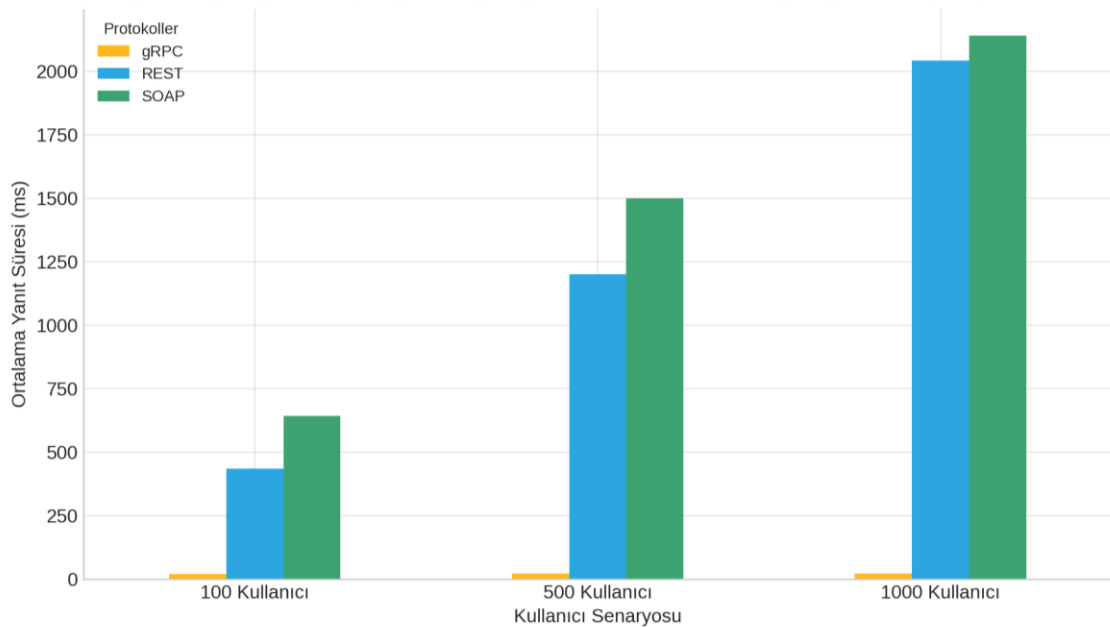
Tablo 4.6. Network Süre Tablosu

Kullanıcı Senaryosu	Protokol	Ortalama (ms)	Minimum (ms)	Maksimum (ms)
10 Kullanıcı	gRPC	23	22	27
10 Kullanıcı	REST	113	83	131
10 Kullanıcı	SOAP	2703	2677	2742
100 Kullanıcı	gRPC	20	17	27
100 Kullanıcı	REST	436	93	900
100 Kullanıcı	SOAP	644	198	1069
1000 Kullanıcı	gRPC	22	17	102
1000 Kullanıcı	REST	2043	227	4109
1000 Kullanıcı	SOAP	2141	150	4135

4.6.2 Grafiklerin İncelenmesi

Şekil 4.11, Şekil 4.12 ve Tablo 4.6, Tablo 4.7’de görülen veriler, kullanıcı sayısına göre ortalama yanıt sürelerini göstermektedir. Eğilimler şu şekilde özetlenebilir:

- **gRPC**: Tüm senaryolarda istisnasız en düşük yanıt sürelerini üretmiştir (~20–23 ms). Kullanıcı sayısı 10’dan 1000’e çıksa bile yanıt sürelerinde anlamlı bir artış olmamıştır. Bu durum, Protobuf tabanlı veri yapısının hafifliği sayesinde gRPC’nin latency açısından son derece istikrarlı olduğunu göstermektedir.
- **REST**: Orta seviyede sonuçlar üretmiştir. 10 kullanıcıda 113 ms iken, 100 kullanıcıda 436 ms, 1000 kullanıcıda ise 2 saniyenin üzerine çıkmıştır. Yani kullanıcı sayısı arttıkça yanıt süresi doğrusal bir şekilde artmıştır. Bu sonuç, JSON serileştirme maliyetlerinin yüksek yük altında sistem performansına yansıdığını göstermektedir.
- **SOAP**: 10 kullanıcıda dahi 2,7 saniye gibi yüksek bir gecikme ile başlamış, 100 kullanıcıda 644 ms’ye gerilese de, 1000 kullanıcıda tekrar 2,1 saniye seviyesine çıkmıştır. Bu dalgalanma, XML’in ağır yapısının ve mesaj boyutlarının sistemde istikrarsızlığa yol açtığını ortaya koymaktadır.



Şekil 4.11. Ortalama yanıt sürelerinin kullanıcı sayısına göre değişimi

Tablo 4.7. Network Süre Tablosu

Kullanıcı Sayısı	Protokol	Ortalama (ms)	Medyan (ms)	Min. (ms)	Maksimum (ms)	95% Sınır (ms)
10	gRPC	23	23	22	27	25
	REST	113	113	83	131	127
	SOAP	2703	2701	2677	2742	2725
100	gRPC	20	20	17	27	26
	REST	436	445	93	900	699
	SOAP	644	641	198	1069	1052
1000	gRPC	22	22	17	102	29
	REST	2043	2007	227	4109	3955
	SOAP	2141	2104	150	4135	4037

4.6.3 Karşılaştırmalı Analiz

1. gRPC

- Yanıt süreleri açısından açık ara en hızlı protokol olmuştur.
- 10, 100 ve 1000 kullanıcı senaryolarında değerler neredeyse sabit kalmıştır.
- Ancak throughput testleri incelendiğinde (Bkz. Bölüm 4.4), gRPC'nin işlem hacmi REST kadar yüksek değildir. Yani gRPC tekil isteklerde çok hızlı, fakat yüksek eşzamanlılıkta daha düşük hacimle çalışmaktadır.

2. REST

- Latency açısından gRPC'nin gerisinde kalmış, ancak SOAP'a göre çok daha iyi performans göstermiştir.
- Kullanıcı sayısı arttıkça yanıt süreleri doğrusal biçimde artmıştır; bu da JSON'un metin tabanlı yapısından kaynaklanan serileştirme/çözümleme maliyetlerini yansıtmaktadır.
- Yine de 1000 kullanıcıda 2 saniyelik ortalama yanıt süresi, kullanıcı deneyimi açısından kabul edilebilir sınırlar içinde kalmaktadır.

3. SOAP

- Latency değerleri açısından en dezavantajlı protokol olmuştur.
- 10 kullanıcıda dahi 2,7 saniyelik yanıt süresi, sistemin ağır başladığını göstermektedir.
- 100 kullanıcıda görece daha düşük bir değer elde edilse de, 1000 kullanıcı testinde tekrar 2 saniyenin üzerine çıkması, SOAP'ın istikrarsız ve tutarsız performans gösterdiğini ortaya koymaktadır.

Bu çalışmada elde edilen bulgular, literatürdeki bu gözlemleri doğrulamaktadır: gRPC en hızlı ve tutarlı protokol; REST dengeli ama JSON'un maliyeti nedeniyle sınırlı; SOAP ise gecikme açısından en zayıf seçenektir.

4.6.4 Genel Değerlendirme

- **gRPC**, latency açısından en uygun protokol olarak öne çıkmıştır. Kullanıcı sayısından bağımsız olarak milisaniye seviyesinde yanıt süreleri üretmiştir.
- **REST**, gRPC'ye kıyasla daha yüksek gecikme göstermiş, fakat SOAP'a göre belirgin avantaj sağlamıştır.
- **SOAP**, hem yüksek hem de tutarsız yanıt süreleri nedeniyle gerçek zamanlı uygulamalarda verimsiz kalmıştır.

Sonuç olarak, gerçek zamanlı ve düşük gecikme gerektiren toplu taşıma uygulamaları için en uygun çözüm gRPC'dir. REST, yüksek throughput avantajıyla ölçeklenebilirlik açısından güçlüdür, SOAP ise modern senaryolarda tercih edilmemelidir.

4.7 Verimlilik: Throughput ve Latency İlişkisi

4.7.1 Genel Bulgular

Sistemlerin gerçek verimliliği yalnızca throughput (saniye başına işlenen istek sayısı) ile değil, aynı zamanda bu isteklerin hangi yanıt süreleriyle gerçekleştiği ile de belirlenmektedir. Yüksek throughput değerleri elde edilirken, yanıt sürelerinin de kabul edilebilir düzeylerde kalması beklenir.

Bu nedenle REST, SOAP ve gRPC protokolleri hem throughput hem de latency verileri birlikte değerlendirilmiştir.

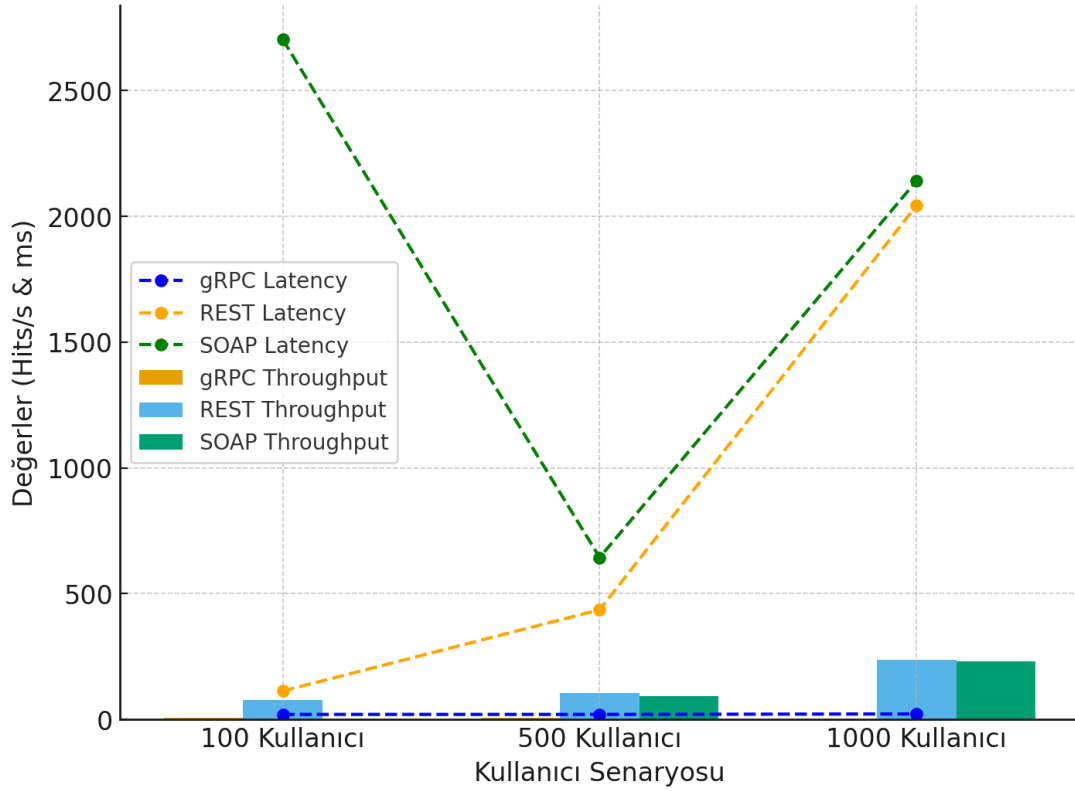
- **gRPC:**
 - Throughput açısından REST'in gerisinde kalmış, özellikle 1000 kullanıcı testlerinde REST'in 200+ hits/s değerine karşılık yalnızca 4–8 hits/s üretmiştir.
 - Buna karşın gecikme değerleri olağanüstü derecede düşük (20–23 ms) ve kararlıdır.
 - Bu durum gRPC'nin tekil isteklerde son derece hızlı olduğunu, ancak aynı anda çok sayıda isteği işleme kapasitesinde sınırlı kaldığını göstermektedir.
- **REST:**
 - Throughput açısından en güçlü protokol olmuştur. 1000 kullanıcı testlerinde ortalama 236 hits/s değerine ulaşmıştır.
 - Ancak latency tarafında değerler kullanıcı yüküyle birlikte artış göstermiş, 100 kullanıcıda 436 ms, 1000 kullanıcıda 2043 ms seviyelerine yükselmiştir.
 - Yani REST, yüksek işlem hacmi sağlayabilen ancak tekil yanıt sürelerinde gecikme yaşayan bir profil çizmiştir.
- **SOAP:**
 - Throughput performansı REST kadar güçlü değildir. 1000 kullanıcı testinde 230 hits/s üretmiş, ancak latency değerleri hem çok yüksek hem de dalgalı kalmıştır (10 kullanıcıda 2,7 saniye, 1000 kullanıcıda 2,1 saniye).
 - Bu durum, SOAP'ın yüksek throughput elde etmesine rağmen kullanıcı deneyimi açısından verimli sayılmayacağını göstermektedir.

4.7.2 Grafiklerin İncelenmesi

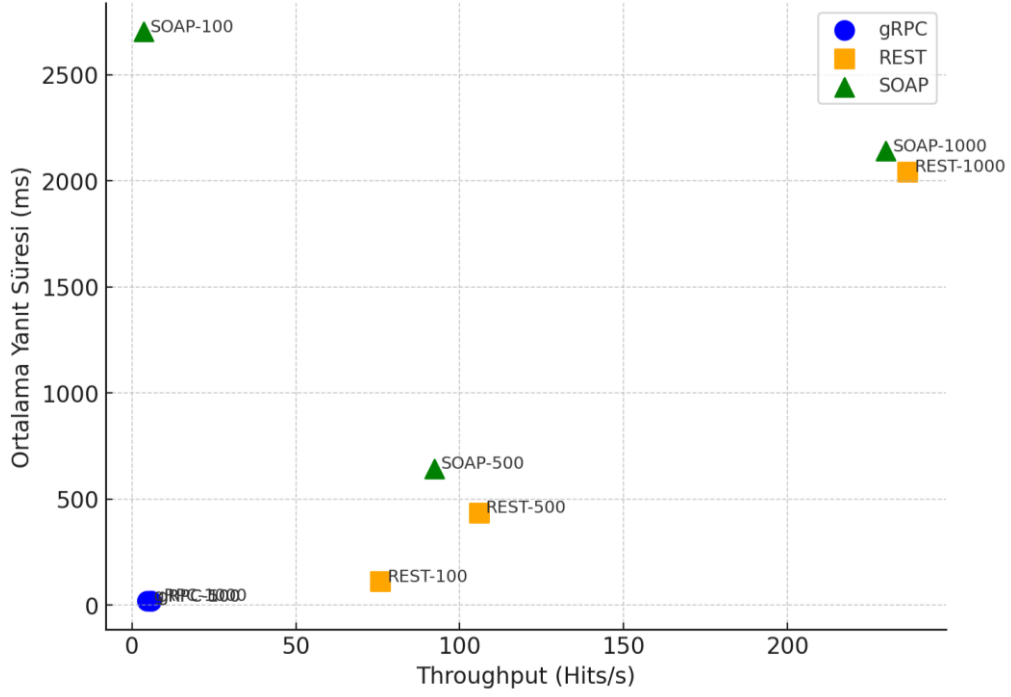
Şekil 4.13, Şekil 4.14 ve Tablo 4.8'de throughput ve latency değerleri birlikte gösterilmektedir. Burada iki önemli eğilim öne çıkmaktadır:

1. **gRPC:** Düşük throughput değerine rağmen, yanıt sürelerinin sürekli olarak milisaniye seviyesinde kalması, bu protokolün “az ama çok hızlı” işleme kapasitesine sahip olduğunu göstermektedir.

2. **REST**: Yüksek throughput ile birlikte, latency değerlerinde doğrusal bir artış gözlemlenmiştir. Bu, REST'in yüksek hacimli yükleri kaldırabildiğini ancak tekil isteklerde gecikmeyi artırdığını göstermektedir.
3. **SOAP**: Yüksek throughput değerlerine rağmen latency değerlerinin kabul edilemez derecede yüksek olması, SOAP'ın teknik verimliliğinin düşük olduğunu ortaya koymaktadır.



Şekil 4.13. Throughput–Latency Dengesi: REST, SOAP, gRPC



Şekil 4.14. Kullanıcı Yüküne Göre İşlem Kapasitesi ve Yanıt Süresi Dağılımı

Tablo 4.8. Network Tablosu

Kullanıcı Sayısı	Protokol	Throughput (işlem/s)	Ortalama Yanıt Süresi (ms)	Min. (ms)	Maksimum (ms)
10	gRPC	8,0	23	22	27
	REST	75,8	113	83	131
	SOAP	3,6	2703	2677	2742
100	gRPC	5,9	20	17	27
	REST	105,9	436	93	900
	SOAP	92,3	644	198	1069
1000	gRPC	4,7	22	17	102
	REST	236,6	2043	227	4109
	SOAP	230,0	2141	150	4135

4.7.3 Karşılaştırmalı Analiz

- **gRPC**, latency tarafında en verimli protokol olmuş, fakat throughput kapasitesi sınırlı kalmıştır. Bu nedenle düşük hacimli, gerçek zamanlı ve kritik hız gerektiren senaryolar için uygundur.

- **REST**, yüksek throughput üretmiş ancak latency tarafında gecikmeler yaşamıştır. Bu nedenle geniş ölçekli ve çok kullanıcıli sistemlerde ölçeklenebilirliği yüksek, fakat kullanıcı başına tepki süresi görece uzun bir protokoldür.
- **SOAP**, throughput açısından güçlü görünebilse de, yüksek latency değerleri nedeniyle verimlilik açısından başarısızdır. XML tabanlı mesajların ağır yapısı, ağ ve işlem kaynaklarını zorlamakta, sonuçta kullanıcı deneyimi olumsuz etkilenmektedir.

4.7.4 Genel Değerlendirme

- **gRPC**: “Hızlı yanıt – düşük kapasite” → Kritik hız gerektiren düşük hacimli uygulamalar için en verimli.
- **REST**: “Yüksek kapasite – orta seviye gecikme” → Yüksek kullanıcı yoğunluğu olan sistemler için dengeli.
- **SOAP**: “Yüksek kapasite – yüksek gecikme” → Modern uygulamalar için verimlilik açısından uygun değil.

Sonuç olarak, verimlilik açısından gerçek zamanlılık öncelikliyse gRPC, ölçeklenebilirlik öncelikliyse REST tercih edilmelidir. SOAP, yalnızca belirli güvenlik standartlarının zorunlu olduğu kurumsal ortamlarda kullanılabilir.

5. SONUÇLAR VE ÖNERİLER

5.1 Çalışmanın Genel Bulguları

Bu tez çalışmasında, şehir içi toplu taşımada kullanılan kart tabanlı biletleme sistemlerine alternatif olarak Bluetooth Low Energy (BLE) beacon tabanlı, tamamen otomatik ve dinamik ücretlendirme modeli geliştirilmiş ve çok boyutlu testlerle değerlendirilmiştir. Geliştirilen sistem, yolcudan hiçbir manuel işlem beklemeksizin otobüs ve duraklara yerleştirilen beacon cihazları aracılığıyla yolculuğun başlangıç ve bitiş noktalarını tespit etmiş, mobil uygulama üzerinden bu verileri sunucuya ileterek ücretlendirme sürecini gerçekleştirmiştir.

Çalışmada yapılan testler ve elde edilen bulgular şu şekilde özetlenebilir:

- **BLE beacon cihazları** otobüs içi ve durak yerleşiminde yüksek doğruluk oranı ile yolcu hareketlerini tespit edebilmiştir. Cihazların kararlı çalışması sayesinde biniş–iniş noktaları güvenilir biçimde kaydedilmiş, sistemin temel işleyişi doğrulanmıştır.
- **Mobil uygulama**, arka planda kesintisiz çalışarak yolcunun herhangi bir manuel etkileşimine gerek kalmadan biniş ve iniş noktalarını kaydetmiş, kullanıcı deneyimini kolaylaştırmıştır.
- **CPU Kullanımı:** gRPC, özellikle orta yük düzeylerinde REST’e kıyasla yaklaşık %25 daha düşük işlemci kullanımı göstermiştir. SOAP başlangıçta dengeli olsa da kullanıcı sayısı arttıkça CPU verimliliğini koruyamamıştır.
- **Bellek Kullanımı:** gRPC, test süresince en kararlı bellek tüketimini sergilemiş, REST’e göre yaklaşık %10–15 daha az, SOAP’a göre ise %20’den fazla verimlilik sağlamıştır.
- **Ağ Trafiği:** gRPC, düşük veri paketleme boyutu sayesinde en dengeli ağ performansını sunmuş, REST’in oluşturduğu trafiğe göre yaklaşık %30 daha az veri aktarımı gerçekleştirmiştir. SOAP, başlangıçta yavaş olsa da yüksek yükte REST kadar agresif artış göstermemiştir.
- **İşlem Kapasitesi (Throughput):** REST, saniye başına işlem sayısında gRPC’ye kıyasla %20 civarında daha yüksek throughput sağlamıştır; ancak uzun süreli testlerde gRPC daha kararlı sonuçlar üretmiştir. SOAP, kısa vadede yüksek throughput üretse de uzun testlerde istikrarsız davranmıştır.
- **Yanıt Süresi (Latency):** gRPC, ortalama yanıt sürelerinde REST’e kıyasla %80’e varan, SOAP’a kıyasla ise %90’ın üzerinde hız avantajı sağlamıştır. REST

orta düzey performans sunarken, SOAP test edilen protokoller arasında en yavaş yanıt sürelerine sahip olmuştur.

- **Verimlilik analizleri**, throughput ve latency sonuçları bir arada değerlendirildiğinde gRPC'nin “az ama hızlı”, REST'in “çok ama daha yavaş”, SOAP'ın ise “çok ama tutarsız ve yavaş” bir profil çizdiğini ortaya koymuştur.
- **Ücretlendirme modeli**, mesafe tabanlı olarak başarıyla uygulanmış, kısa mesafe yolcuları için düşük ücret, uzun mesafe yolcuları için adil ve kademeli ücretlendirme sağlanmıştır.
- **Kullanıcı deneyimi açısından**, yolcuların hiçbir manuel işlem yapmadan yolculuklarını tamamlayabilmesi süreçleri hızlandırmış, yoğun saatlerde kuyrukların azalmasına katkı sağlamış ve genel memnuniyet düzeyini artırmıştır.

5.2 Akademik Katkıları

Bu tez, literatürdeki mevcut BLE tabanlı toplu taşıma çalışmalarını genişleterek aşağıdaki özgün katkıları sunmuştur:

1. **Biniş–iniş doğruluğu**: BLE teknolojisinin yalnızca otobüs içi değil, aynı zamanda durak tabanlı yerleşimle birlikte kullanılması sayesinde yolculuk başlangıç ve bitiş noktalarının güvenilir biçimde tespiti sağlanmış, bu yöntem literatürde nadiren ele alınmış bir yaklaşım olarak öne çıkmıştır.
2. **Çoklu API kıyaslaması**: Üç farklı API teknolojisinin (REST, SOAP, gRPC) aynı sistem üzerinde uygulanarak CPU, bellek, ağ trafiği, throughput ve latency testleriyle kapsamlı bir karşılaştırma yapılmıştır. Bu, protokol düzeyinde literatüre önemli bir katkı sağlamaktadır.
3. **Gerçek saha testi**: Çalışmada yalnızca laboratuvar ortamı değil, aynı zamanda saha pilot uygulamaları gerçekleştirilmiş ve teorik modelin pratikte uygulanabilirliği doğrulanmıştır.
4. **Dinamik ücretlendirme modeli**: BLE teknolojisiyle entegre edilerek hem adil ücretlendirme hem de veri odaklı ulaşım yönetimi açısından yenilikçi bir katkı sunmuştur.
5. **Verimlilik analizi**: İşlem kapasitesi ve gecikme sonuçları bir arada değerlendirilerek, protokollerin “gerçek kullanım senaryolarında” nasıl davrandığı gösterilmiştir. Bu, performans testlerini tek boyutlu değil, bütüncül bir yaklaşımla ele alan özgün bir katkıdır.

5.3 Pratik Katkılar

Çalışmanın uygulamaya yönelik çıktıları, ulaşım otoriteleri ve belediyeler açısından somut faydalar sunmaktadır:

6. **Kuyruk Azaltma:** Yolcuların validatörlerde kart okutma ihtiyacının ortadan kalkması, özellikle yoğun saatlerde durak ve otobüs girişlerinde kuyrukların önemli ölçüde azalmasını sağlamaktadır.
7. **Hızlı ve Konforlu Deneyim:** Yolcular, telefonlarını ceplerinden çıkarmadan otobüse binebilmekte; bu da hem kullanım kolaylığı hem de daha modern bir kullanıcı deneyimi sunmaktadır.
8. **Adil Ücretlendirme:** Mesafeye dayalı ücretlendirme sayesinde kısa mesafe yolcular daha düşük maliyetle seyahat edebilmekte, uzun mesafeli yolcular ise adil bir ücret ödemektedir.
9. **Veri Toplama:** Sistem, biniş-iniş verilerini zaman damgalı olarak kaydederek ulaşım otoritelerine güzergâh planlaması, sefer optimizasyonu ve yoğunluk yönetimi için güçlü bir veri tabanı sağlamaktadır.
10. **Pandemi Sonrası Uygunluk:** Sistemin temassız çalışması, bulaşıcı hastalıkların yayılma riskini azaltmakta ve kamu sağlığı açısından ek bir güvenlik katmanı sunmaktadır.

5.4 Çalışmanın Kısıtları

Bu çalışmanın kapsamı belirli sınırlar dâhilinde tanımlanmaktadır. Deneysel süreçler yalnızca otobüs taşımacılığı senaryoları üzerinden yürütülmüş, metro ve tramvay gibi diğer ulaşım modları sonraki araştırmalar için dışarıda bırakılmıştır. Ayrıca, saha tabanlı kullanıcı memnuniyeti ölçümleri ve çok merkezli ölçeklenebilirlik testleri bu tez kapsamına dâhil edilmemektedir.

Gelecek çalışmalarda, sistemin çoklu ulaşım modlarını kapsayacak biçimde genişletilmesi, kullanıcı davranışlarının derinlemesine analiz edilmesi ve ölçeklenebilirliğin farklı sunucu yapılandırmalarıyla test edilmesi planlanmaktadır.

5.5 Gelecek Çalışmalar İçin Öneriler

Çalışmanın sonuçları ışığında, gelecekte yapılabilecek araştırmalara yönelik öneriler şu şekilde sıralanabilir:

11. **Çoklu Ulaşım Modları:** Sistem, metro, tramvay ve banliyö hatları gibi farklı ulaşım türlerinde test edilerek bütünleşik bir akıllı ulaşım modeli oluşturulabilir.
12. **Siber Güvenlik Testleri:** Beacon kimliklerinin kopyalanması (spoofing) veya sahte beacon saldırıları gibi güvenlik riskleri için detaylı saldırı senaryoları ve savunma mekanizmaları geliştirilebilir.
13. **Ölçeklenebilirlik:** Yüksek kullanıcı sayılarında performansı ölçmek için bulut tabanlı altyapılar (load balancer, microservices) kullanılabilir.
14. **Enerji Verimliliği:** Beacon cihazlarının enerji tüketimi optimize edilerek batarya ömrü artırılabilir. Aynı zamanda güneş enerjisi destekli beacon cihazları üzerinde çalışmalar yapılabilir.
15. **Kapsamlı Kullanıcı Anketleri:** Yolcuların sistem hakkındaki memnuniyet düzeylerini ölçmek için geniş ölçekli saha anketleri yapılabilir.
16. **Politika ve Fiyatlandırma:** Dinamik ücretlendirme modeli, yoğunluk yönetimi ve sosyal adalet açısından farklı şehirlerde denenerek sosyoekonomik etkileri analiz edilebilir.

5.6 Genel Değerlendirme

Sonuç olarak bu çalışma, BLE beacon teknolojisinin toplu taşımada tam otomatik, temassız ve dinamik ücretlendirme için uygulanabilir olduğunu göstermiştir. Kullanıcı deneyimi, operasyonel verimlilik ve veri odaklı yönetim açısından sağladığı katkılar, sistemin yalnızca akademik düzeyde değil, aynı zamanda pratikte de güçlü bir potansiyel sunduğunu ortaya koymuştur.

Geliştirilen model, akıllı şehir vizyonuna uyumlu, düşük maliyetli ve kullanıcı dostu bir çözüm olarak gelecekte birçok şehirde uygulanabilecek niteliktedir. Bu bağlamda, çalışmanın hem literatüre hem de kamu hizmetlerinin dijital dönüşümüne katkı sağlayacağı değerlendirilmektedir.

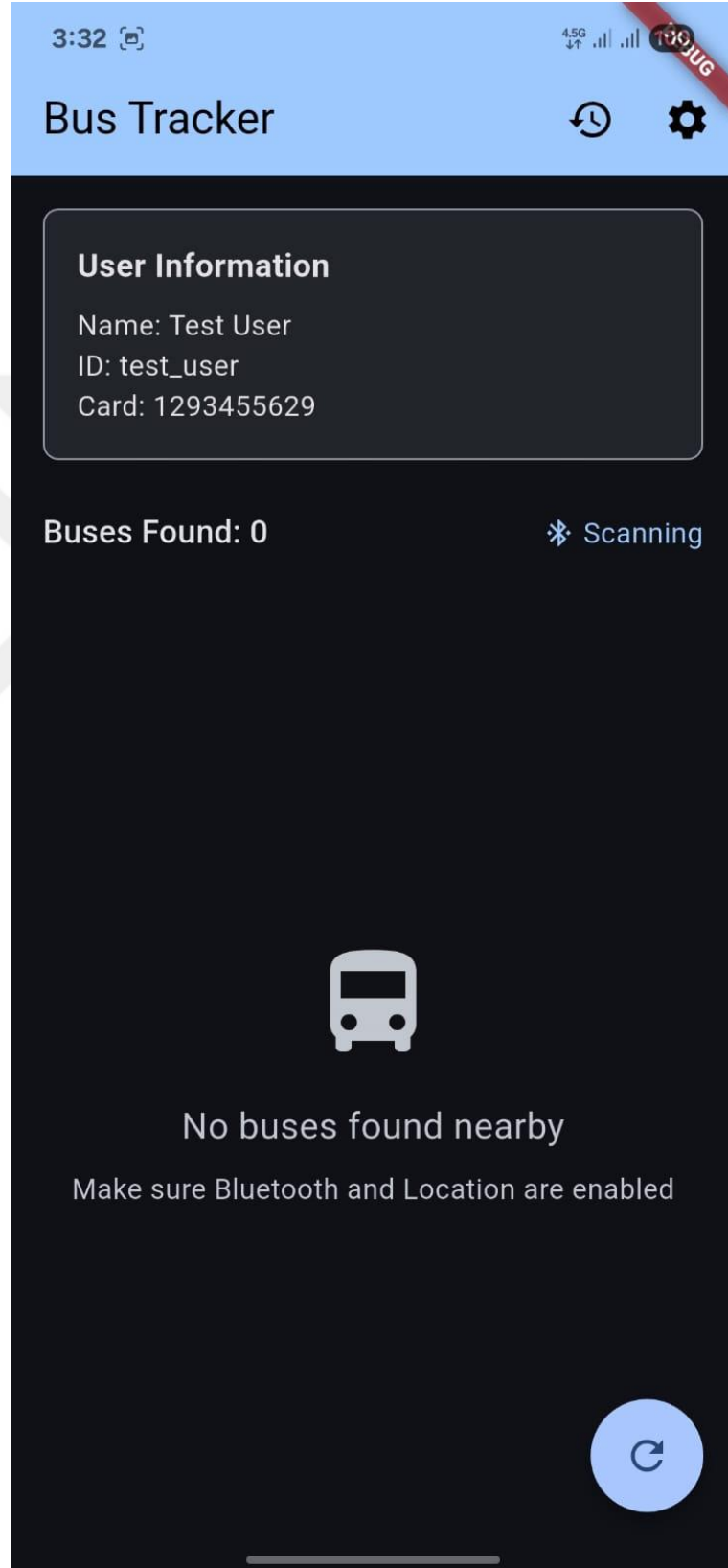
KAYNAKLAR

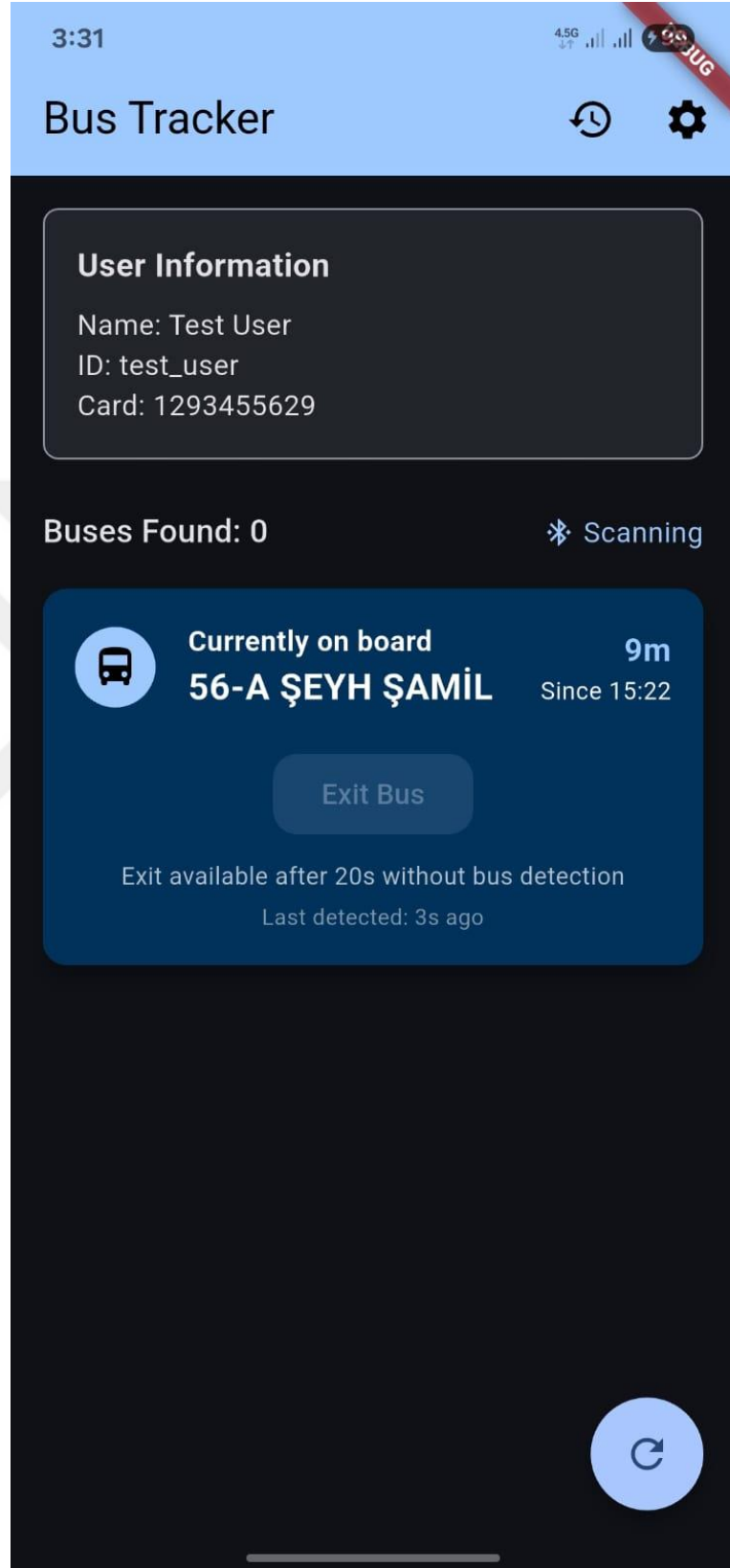
- Ahmed, S. and Khan, R., 2021, Performance benchmarking of API protocols, *Proc. Int. Conf. Distributed Computing*, 233–240.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., 2015, Internet of Things: A survey on enabling technologies, protocols, and applications, *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376.
- Banerjee, N., Mukherjee, S. and Roy, P., 2016, A survey on REST, SOAP, and gRPC, *International Journal of Web Services*, 12(2), 23–34.
- Bohli, J. M., Sorge, C. and Westhoff, D., 2015, Ticket-based service access for mobile users, *IEEE Transactions on Mobile Computing*, 14(4), 829–842.
- Cats, O., Susilo, Y. and Reimal, T., 2016, The prospects of fare capping in public transport, *Transport Policy*, 48, 218–230.
- Chen, C., 2017, A survey on QR-code based mobile payment systems, *Proc. Int. Conf. Information Technology*, 67–74.
- Chou, C. T. and Lee, C., 2016, BLE signal stability and localization accuracy, *Wireless Personal Communications*, 89(3), 895–912.
- Currie, G. and Delbosc, A., 2011, Zone-based fare systems in metropolitan transit, *Transport Reviews*, 31(2), 179–197.
- Dinh, H. and Pham, L., 2020, Cloud-based fare systems for public transportation, *Journal of Cloud Applications and Smart Mobility*, 14(1), 67–79.
- Dias, M. and Ferreira, A., 2019, BLE-based passenger monitoring in urban mobility, *Journal of Transport and Smart Mobility*, 11(2), 155–168.
- Fazio, M., Paone, M., Puliafito, A. and Villari, M., 2016, Heterogeneous sensors become homogeneous things in smart cities, *Journal of Communications and Networks*, 18(2), 281–295.
- Faragher, R. and Harle, R., 2015, Location fingerprinting with Bluetooth Low Energy beacons, *IEEE Journal on Selected Areas in Communications*, 33(11), 2418–2428.
- Ferreira, D., Dias, J. and Cunha, J. F., 2020, BLE-based mobile ticketing in public transport: a field study in Porto, *IEEE Access*, 8, 15472–15485.
- Ferreira, D., Rodrigues, J. J. and Saleem, K., 2020, Performance evaluation of BLE-based mobile payment systems in smart transportation, *IEEE Access*, 8, 15472–15485.
- Gonzalez, A. and Sanchez, J., 2018, Privacy and security issues in BLE-based ticketing, *Proc. Int. Conf. Cybersecurity*, 88–96.
- Gravit Studio, 2025, Flutter BLoC: 2025 Beginners Guide to State Management in Flutter Apps [online], *Medium*, <https://medium.com/@gravitstudio/flutter-bloc-2025-beginners-guide-to-state-management-in-flutter-apps-6bd71fe1fd3e> [Ziyaret Tarihi: 12 Kasım 2025].
- Guo, R. and Lin, X., 2022, Scalability and performance of microservice protocols, *IEEE Access*, 10, 12456–12470.
- He, W., 2018, Smart airports and BLE-based navigation, *International Journal of Aviation Technology*, 5(1), 19–28.
- Hensher, D. and Li, Z., 2012, Ridership and revenue impacts of multiple pricing structures, *Transportation Research Part A*, 46(6), 855–867.

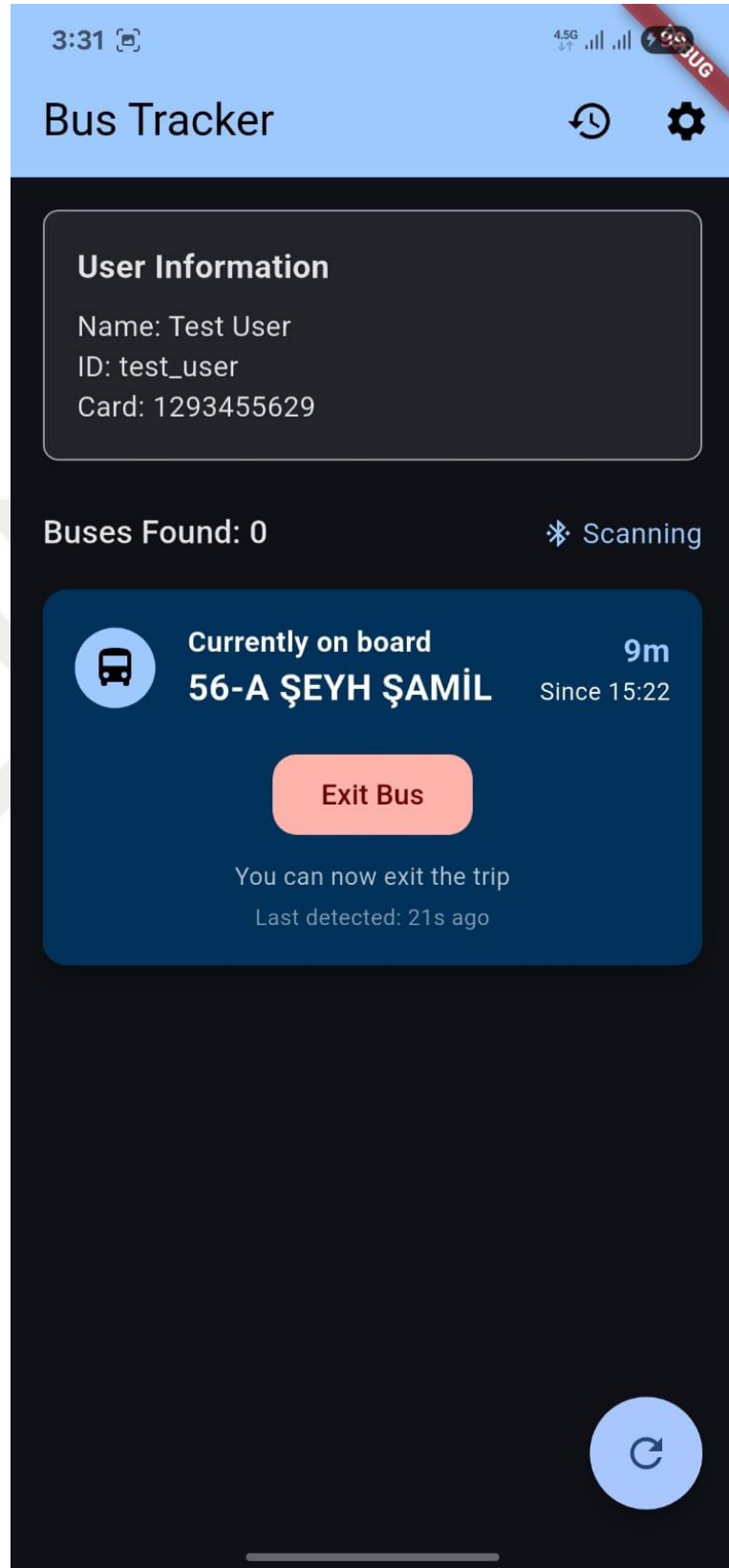
- Hu, X., Li, Y. and Chen, Z., 2020, A comparative study of NFC and BLE for contactless ticketing, *Sensors*, 20(3), 712.
- Kahvazadeh, P., 2016, A pilot study of Bluetooth Low Energy beacons in public transportation ticketing, *Proc. Int. Conf. Smart Cities and Green ICT Systems (SMARTGREENS)*, Soest, Germany, 112–118.
- Kiryakov, A., 2020, Comparing microservices communication: REST vs gRPC, *Journal of Cloud Computing Advances*, 5(4), 45–53.
- Li, Y. and Zhao, H., 2019, Indoor positioning with BLE: A survey, *Sensors*, 19(8), 1977.
- López-de-Ipiña, D., Almeida, A. and Aguilera, U., 2017, iBeacon for context-aware retail applications, *Journal of Ambient Intelligence and Smart Environments*, 9(2), 227–239.
- Luo, X. and Chen, Z., 2017, Dynamic fare collection in smart transit, *Journal of Public Transportation*, 20(4), 55–67.
- Martín, D., Ruiz, A. and Sánchez, A., 2019, Beacon-based smart campus applications, *Future Internet*, 11(7), 151.
- Mehmood, R. and Graham, G., 2015, Smart transport management systems: A review, *Springer Smart Cities Series*, 145–167.
- Miklos, Z., Kovacs, L. and Molnar, B., 2019, Comparative performance evaluation of JSON, XML and Protocol Buffers in web services, *International Journal of Computer Applications*, 975(8887), 12–19.
- Namiot, D. and Sneps-Sneppe, M., 2014, On beacons and proximity, *International Journal of Open Information Technologies*, 2(6), 29–33.
- Narzt, W., Pomberger, G. and Sametinger, J., 2015a, Be-in/be-out systems in public transport: a conceptual approach using Bluetooth beacons, *Proc. Int. Conf. Software Engineering Advances (ICSEA)*, 102–108.
- Narzt, W., Pomberger, G. and Sametinger, J., 2015b, SOAP vs. REST vs. gRPC: performance analysis for service-oriented middleware, *Proc. Int. Conf. Software Engineering Advances (ICSEA)*, 102–108.
- Park, J. and Lee, H., 2017, Application of BLE beacons in smart bus systems, *IEEE Access*, 5, 24602–24611.
- Querini, M., Palazzi, C. E. and Rocchetti, M., 2017, Museum guides with BLE beacons, *Multimedia Tools and Applications*, 76(4), 5277–5294.
- Sharaby, N. and Shiftan, Y., 2012, The impact of fare integration on travel behavior, *Transport Policy*, 21, 63–70.
- Shen, H., Yu, H. and Bu, J., 2018, Performance evaluation of RESTful and gRPC frameworks for microservices, *Proc. Int. Conf. Cloud Computing and Big Data (CCBD)*, 415–420.
- Sun, J. and He, Y., 2019, Evaluation of NFC, QR and BLE in transport applications, *International Journal of Transportation Science*, 8(3), 201–215.
- Vuchic, V., 2005, *Urban Transit: Operations, Planning, and Economics*, Wiley, New Jersey, 45–52.
- Wang, H. and Wu, J., 2016, An analysis of Bluetooth Low Energy in IoT applications, *IEEE IoT Journal*, 3(5), 485–497.
- Yan, L. and Ma, X., 2018, Mobile payment and smart transportation, *Journal of Advanced Transportation*, Article ID 9837234.

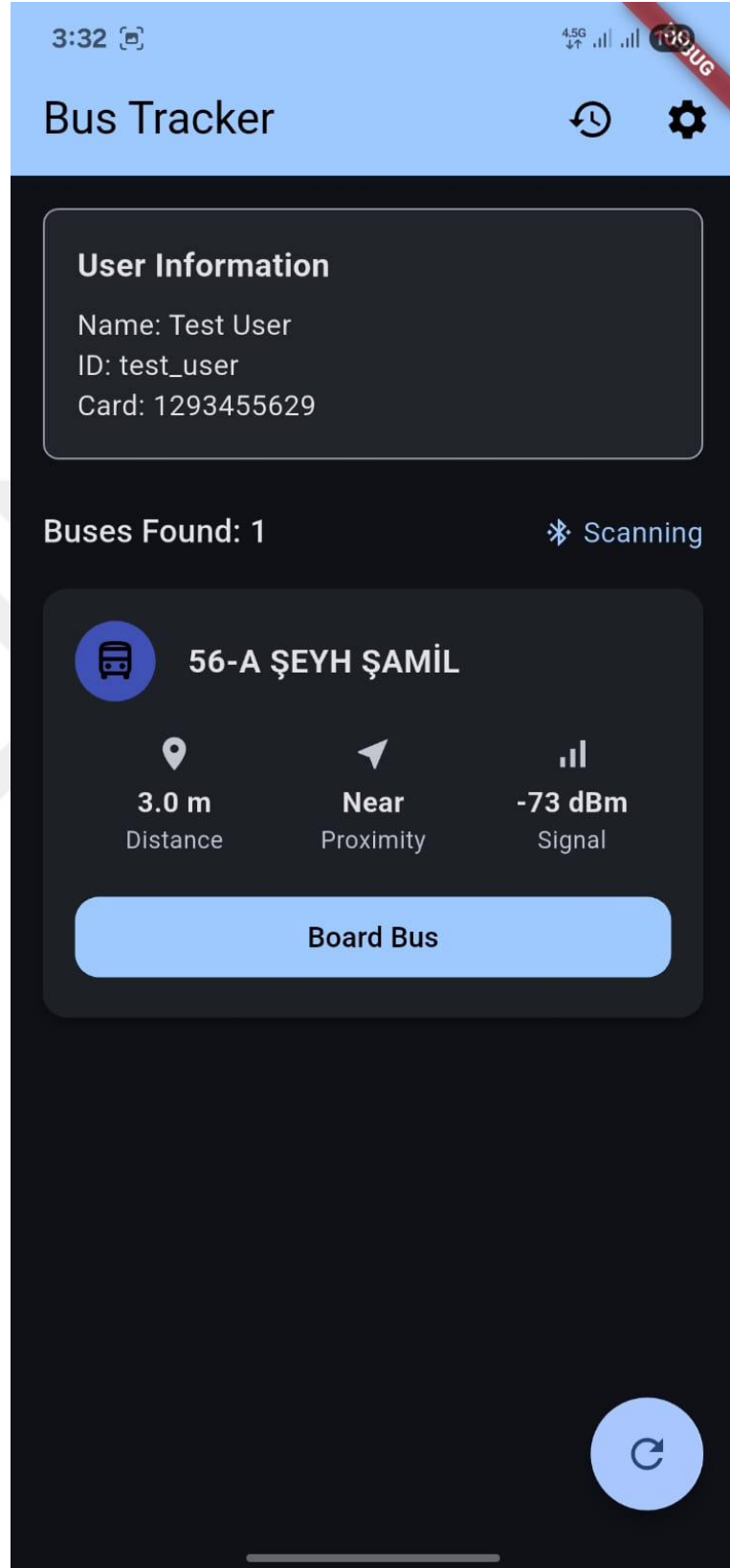
- Yavuz, A. and Çelik, M., 2021, Türkiye’de akıllı kart tabanlı toplu taşıma sistemleri: İstanbulKart, AnkaraKart ve KonyaKart örnekleri, *Ulaştırma ve Lojistik Araştırmaları Dergisi*, 5(2), 45–59.
- Yoon, J., Kim, D. and Kim, H., 2016, BLE-based indoor positioning system for hospitals, *IEEE Consumer Electronics Magazine*, 5(4), 34–41.
- Zhang, Y., 2021, A comparative analysis of public transport ticketing technologies, *Transportation Research Procedia*, 52, 250–258.
- Zhou, F., Zheng, Z. and Yang, H., 2019, Dynamic pricing for shared mobility on demand: A reinforcement learning approach, *Transportation Research Part C: Emerging Technologies*, 108, 156–176.
- Zhou, W. and Piramuthu, S., 2015, Retail marketing with beacons, *Journal of Retail Analytics*, 11(2), 33–45.



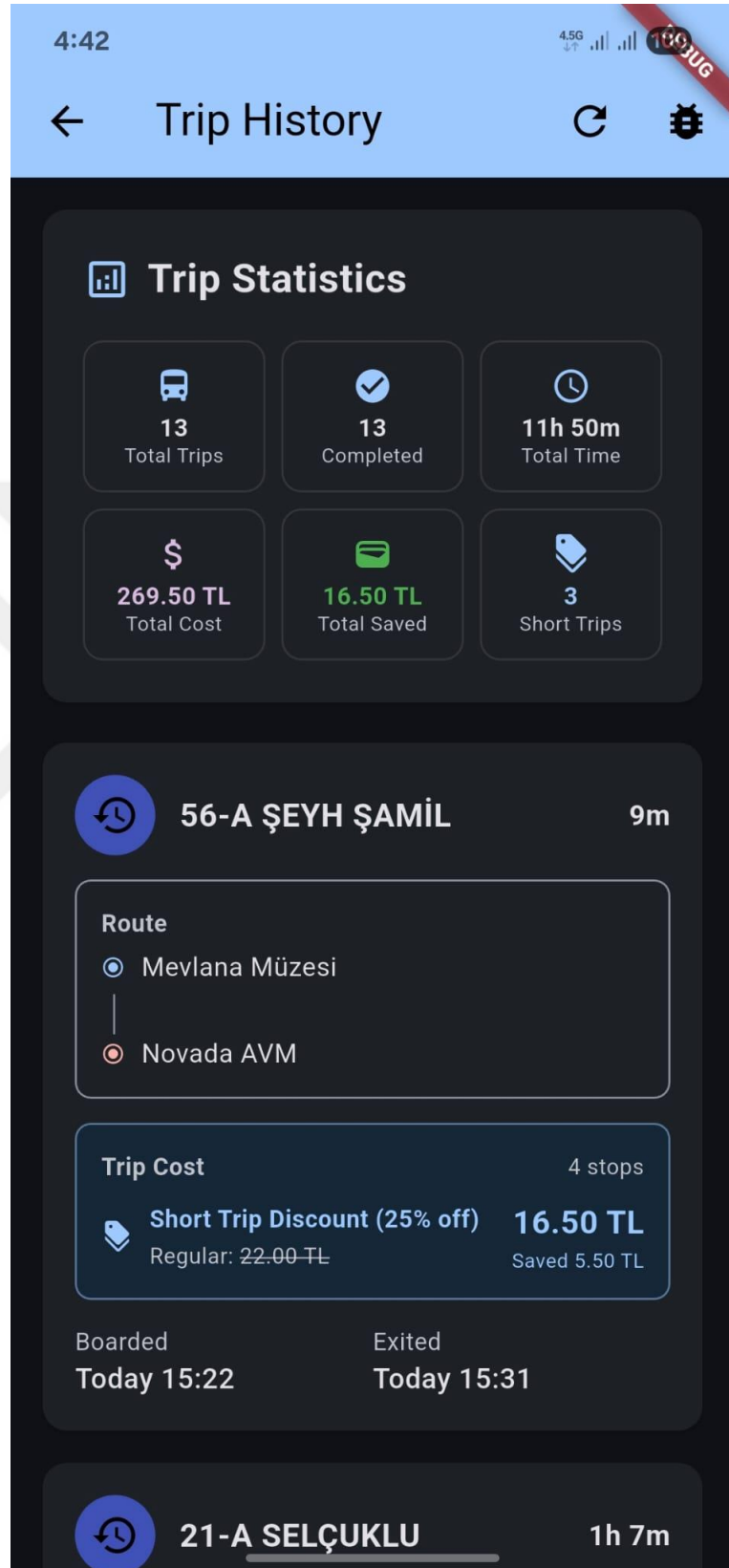
EKLER**EK-1** Mobil Uygulama Otobüs Bekleme Ekranı

EK-2 Mobil Uygulama Otobüste Yolculuk Ekranı

EK-3 Mobil Uygulama Otobüsten İniş Opsyonu

EK-4 Mobil Uygulama Otobüse Binme Opsiyonu

EK-5 Mobil Uygulama Yolculuk Geçmişi Panosu



EK-6 BLE Cihazının otobüs ierisindeki farklı resimleri

EK-7 BLE Cihazının otobüs içerisindeki farklı resimleri