

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



LİSANS BİTİRME PROJESİ

BUS E-TICKET PLATFORM

Muhammad KALUMIAN

Yazılım Mühendisliği Bölümü

HAZİRAN2022

T.C.
BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



LİSANS BİTİRME PROJESİ

BUS E-TICKET PLATFORM

Muhammad KALUMIAN

DANIŞMAN
Doç. Dr. Serhat KILIÇARSLAN

Yazılım Mühendisliği Bölümü

HAZİRAN 2022

ONAY

..... tarafından hazırlanan “**Proje Adı Koyu Harflerle Yazılacaktır**” adlı proje çalışması .../.../... tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği/oyçokluğu ile Bandırma Onyedli Eylül Üniversitesi, Mühendislik ve Doğal Bilimleri Fakültesi, Yazılım Mühendisliği Bölümünde LİSANS BİTİRME PROJESİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Unvanı, Adı ve Soyadı
(Danışman)

Unvanı, Adı ve Soyadı
(Üye)

Unvanı, Adı ve Soyadı
(Üye)

Unvanı, Adı ve Soyadı
(Üye) (varsa)

Unvanı, Adı ve Soyadı
(Üye) (varsa)

Doç. Dr. Abdullah ELEN

Bölüm Başkanı

ÖZET

Bu yazım kılavuzu, Bandırma Onyedi Eylül Üniversitesi Yazılım Mühendisliği Bölümünde hazırlanan lisans bitirme projesi yazımında birliği sağlamak ve projenin yazımına rehberlik etmek amacıyla Yazılım Mühendisliği Bölüm Başkanlığınca hazırlanmıştır. Proje yazım kuralları; biçimle ilgili kurallar ve proje çalışmasının bölümleri ve bölümlerin içeriği ile ilgili kurallar olmak üzere iki ana başlık altında verilmiştir.

TEŐEKKÖR

Çalışmayı yöneten danışman ve diğler kişilere, varsa destekleyen, olanak ve malzeme sağlayan kişi ve kuruluşlara, her türlü emeğı geçen kişilere, öğrenci arkadaşlarına ve aileye teşekkür edilebilir. Bu kısım öğrenci için özel bir sayfadır, mesela, öğrenci çalışmasını burada bir kişiye adayabilir. Teşekkür bölümünün başlık yazısı iki yana yaslanmış olarak, 12 pt. büyüklüğünde büyük ve kalın harflerle yazılır. Başlığın bulunduğu satırdan önce 6 nk. sonra ise 18 nk. boşluk bırakılır. Teşekkür metninden sonra bir satır boşluk bırakılır ve sonraki satıra 12 pt. büyüklüğünde kalın harflerle öğrencinin adı ve soyadı yazılır.

Muhammad KALUMIAN

İÇİNDEKİLER

1. GİRİŞ	12
1.1 Proje Tanımı.....	12
1.2 Sorunun Tanımı.....	12
1.2.1 Sorunlar.....	13
1.2.2 BUS E-TICKET Platformunun Çözümü:	13
1.3 Amaçlar ve Hedefler	14
1.3.1 Genel Amaç:	14
1.3.2 Detaylı Hedefler:.....	14
1.4 Proje Kapsamı	16
1.4.1 Hedef Kitle:.....	16
1.4.2 Temel Hizmetler:	17
1.4.3 Teknik Avantajlar:	18
1.5 Kısaca Metodoloji	20
1.5.1 Pazar Analizi ve İhtiyaç Belirleme:	20
1.5.2 Teknik Geliştirme:	20
2. PROJE ARKA PLANI VE GEREKÇELER	22
2.1 Sorunun Arka Planı ve Analizi	22
2.1.1 Şehirlerarası Toplu Taşımaya Genel Bakış.....	22
2.1.2 Toplu Taşımada Gelişim Süreci.....	23
2.1.3 Geleneksel Toplu Taşıma Zorlukları	24
2.1.4 Toplu Taşıma Talebindeki Artış (Suudi Arabistan Örneği)	24
2.1.5 Dijital Dönüşümün Önemi.....	25
2.1.6 Dijitalleşmenin Eksikliği Durumunda Etkiler	26
2.1.7 Dijital Çözüm İhtiyacı.....	26
2.2 Paydaşlar	26
2.2.1 Paydaşların Tanımlanması	27
2.2.2 Paydaşların Rol ve Önemi Tablosu.....	28
2.3 Benzer Projeler.....	28
2.3.1 BUS E-TICKET Platform'a Benzer Projelerin Başarısı.....	28
2.3.2 Benzer Projeler ile BUS E-TICKET Platform'un Karşılaştırması.....	30
2.3.3 Sonuç.....	30
2.4 Proje Gerekçeleri.....	31
2.4.1 Toplu Taşıma Sektöründe Dijital Dönüşüm İhtiyacı	31
2.4.2 Hizmet Sağlayıcılar Arasında Entegrasyonu Destekleme	31
2.4.3 Müşteri Deneyimini İyileştirme	31
2.4.4 Suudi Arabistan 2030 Vizyonu'na Destek.....	31
2.4.5 SaaS ve B2C Modellerinin Başarılarından Yararlanma	32
2.4.6 Katma Değer	32
2.4.7 Ekonomik Fırsatlar ve Gelecek Genişleme Potansiyeli.....	32
3. TASARIM VE ANALİZ.....	32
3.1 Gereksinim Analizi	32
3.1.1 Fonksiyonel Gereksinimler	32
3.1.2 Fonksiyonel Olmayan Gereksinimler	34
3.1.3 Use Case Diagram.....	36
3.2 Sistem Mimarisi	37
3.2.1 Yazılım Mimarisi	37
3.2.2 BUS E-TICKET Platform'da Kullanılan Yazılım Mimarisi	40
3.3 Sistem tasarımları.....	41

3.3.1 Sistem Mimarisi (System Architecture).....	41
3.3.2 Kullanım Durumu Diyagramı (Use CaseDiagram)	42
3.3.3 Paket Diyagramı (Package Diagram).....	43
3.3.4 Sınıf Diyagramı (Class Diagram)	44
3.3.5 Schema Diyagramı (Schema Diagram)	45
4. YÖNTEM.....	45
4.1 SQL Veritabanı (SQL Database)	45
4.2 Arka Uç (Back-End)	46
4.2.1 Diller ve Frameworkler	46
4.2.2 Veri Erişim Katmanı (Data Access Layer - DAL).....	46
4.2.3 İş Mantığı Katmanı (Business Logic Layer - BLL).....	48
4.2.4 Sunum Katmanı (Presenter Layer).....	48
4.3 Ön Uç (Front-End)	49
5. UYGULAMA.....	50
6. SONUÇ VE ÖNERİLER.....	51
Ekler	51
KAYNAKLAR	52
ÖZGEÇMİŞ.....	55

TABLO LİSTESİ

ŞEKİL LİSTESİ

SİMGELER VE KISALTMALAR LİSTESİ

1. GİRİŞ

Tarih boyunca insan, yerler arasında seyahat etmek için gereken zamanı azaltmaya çalışmıştır. Bu çaba, hayvanların kullanımından modern araçlara kadar uzanan gelişmiş ulaşım araçlarının ortaya çıkmasını sağlamıştır. Ancak, bireysel otomobillerin kullanımının artmasıyla birlikte, yüksek maliyetler ve yakıt fiyatları gibi çeşitli zorluklar da gündeme gelmiştir. Bu durum, ülkeleri sürdürülebilir ve etkili bir çözüm olarak toplu taşıma sistemlerini geliştirmeye yöneltmiştir ve bu bağlamda toplu taşıma hizmetleri ortaya çıkmıştır. Son on yılda ise dijitalleşme, toplu taşıma hizmetlerinin geliştirilmesinde kritik bir rol oynamaya başlamıştır. Birçok ülke ve şirket, operasyonları yönetmek, bilgi sağlamak ve bilet satışlarını uzaktan gerçekleştirmek için dijital sistemler benimsemiştir. Bu dijitalleşme, ekonomiyi ve kalkınmayı destekleme açısından önemli faydalar sağlamaktadır [1].

Projemiz, bu zorluklara bir yanıt olarak ortaya çıkmıştır ve şehirlerarası toplu taşıma deneyimini iyileştirmek için yenilikçi bir teknolojik çözüm sunmayı amaçlamaktadır. Proje kapsamında, hem şirketlere hem de müşterilere hizmet eden entegre bir dijital sistem sunulacaktır.

1.1 Proje Tanımı

BUS E-TICKET Platform projesi, şehirlerarası toplu taşıma hizmetlerini kullanan müşteriler ile bu hizmetleri sağlayan operatörler için geliştirilmiş entegre bir bulut tabanlı platformdur. Platform, dijital altyapıya sahip olmayan hizmet sağlayıcıları desteklemek amacıyla, Hizmet Olarak Yazılım (Software as a Service - SaaS) modelinde gelişmiş dijital çözümler sunmaktadır [2]. Ayrıca, müşterilerin uygun seyahatleri seçmesine, bilet rezervasyonu yapmasına ve rezervasyonlarını kolayca yönetmesine olanak tanıyan B2C tabanlı bir arayüz sunar [3]. (Hedef kitle hakkında detaylı tanım, proje kapsamı bölümünde sunulacaktır.)

1.2 Sorunun Tanımı

Şehirlerarası toplu taşıma, öğrenciler, çalışanlar ve turistler gibi geniş bir kullanıcı kitlesi için hayati öneme sahiptir. Ancak, bu sektör, kullanıcı deneyimini olumsuz etkileyen çeşitli temel sorunlarla karşı karşıyadır:

1.2.1 Sorunlar

1.2.1.1 Hizmet sağlayıcıların dijital altyapıdan yoksun olması:

Birçok özel şirket ve bireysel operatör, faaliyetlerini yönetmek için gereken teknik bilgiye veya kaynaklara sahip değildir. Bu, verimliliğin düşmesine ve müşterilere erişimin zorlaşmasına neden olur.

1.2.1.2 Yeni girişimlerin zorluklarla karşılaşması:

Bazı yeni girişimler dijital çözümler sunmaya çalışsa da maliyet, verimlilik ve ölçeklenebilirlik konularında zorluklarla karşılaşmaktadır.

1.2.1.3 Başlangıç yapamayan şirketler:

Bazı küçük ölçekli operatörler ve şirketler, dijitalleşmeye nasıl geçileceği konusunda yeterli bilgiye sahip değildir.

1.2.1.4 Yüksek maliyetlerden kaynaklanan engeller:

Bazı hizmet sağlayıcılar dijital dönüşümün önemini fark etse de, gelişmiş dijital sistemlerin yüksek maliyetlerini karşılamakta zorlanmaktadır.

1.2.1.5 Hizmet sağlayıcılar arasında entegrasyon eksikliği:

Mevcut hükümet uygulamaları ve sınırlı çözümler, şirketler arasında etkili bir entegrasyondan yoksundur. Bu durum, müşterilere tüm seçeneklerin tek bir arayüzde sunulmasını engeller.

1.2.1.6 Rezervasyon ve ödeme işlemlerindeki karmaşıklık:

Rezervasyon ve ödeme işlemlerinde kullanılan geleneksel yöntemler, kullanıcılar için rahatsız edici deneyimlere yol açmaktadır.

1.2.1.7 Dijital operasyon altyapısının eksikliği:

Bu durum, sefer yönetimi, ödemeler, müşteri verileri ve raporlama süreçlerinde şeffaflık ve verimliliğin azalmasına neden olmaktadır.

1.2.2 BUS E-TICKET Platformunun Çözümü:

BUS E-TICKET Platformu, yukarıdaki sorunları çözmek için çeşitli hizmet sağlayıcıların ihtiyaçlarını karşılayan entegre dijital çözümler sunmayı

amaçlamaktadır. Platform, **SaaS modeli** üzerine inşa edilmiştir, bu da kolay kullanım ve düşük maliyet avantajı sağlamaktadır.

1.2.2.1 Platform, müşterilere şu imkanları sunar:

- Uygun seyahatleri kolayca seçme.
- Bilet rezervasyonu yapma ve rezervasyonları etkin bir şekilde yönetme.

1.2.2.2 Hizmet sağlayıcılar için ise platform, şu avantajları sağlar:

- Yüksek verimlilikle sefer ve ödeme yönetimi.
- Performansı artırmak için şeffaf ve düzenli raporlardan yararlanma.
- Sağlanan dijital altyapı sayesinde hizmet kapsamını kolayca genişletme.

BUS E-TICKET Platformu, hizmet sağlayıcıların operasyonel maliyetlerini azaltırken, müşterilerin deneyimini iyileştirerek, gelişmiş ve entegre dijital çözümler sunarak ideal bir denge oluşturur.

1.3 Amaçlar ve Hedefler

1.3.1 Genel Amaç:

Şehirlerarası toplu taşıma müşterileri ve hizmet sağlayıcılarına yönelik yenilikçi teknik çözümler sunan, kapsamlı ve entegre bir dijital platform geliştirmek. Bu platform, kullanıcı dostu bir deneyim sağlarken, operasyonel süreçlerin dijitalleşmesini sürdürülebilir ve etkili bir şekilde destekleyecektir.

1.3.2 Detaylı Hedefler:

1.3.2.1 Dijitalleşmenin Sağlanması:

- Dijital altyapıdan yoksun hizmet sağlayıcılar için dijital bir altyapı oluşturmak.
- Veri, ödeme ve rezervasyon yönetimi için araçlar sunmak.
- Hizmet sağlayıcıların sunduğu seferlerin müşteri arayüzünde birleşik bir şekilde sunulmasını sağlamak.

1.3.2.2 Entegrasyonun Artırılması:

- Hizmet sağlayıcılarının tüm hizmetlerini tek bir platformda entegre etmek.

- Müşterilere sunulan tüm seferleri, kullanıcı dostu bir arayüzde toplamak ve çoklu seçenekler sunmak.

1.3.2.3 Müşteri Deneyiminin İyileştirilmesi:

- Hassas ve kolay filtreleme araçları ile seyahat arama sürecini basitleştirmek.
- Müşterilerin güvenli bir şekilde rezervasyon ve online ödeme yapabilmelerini sağlamak.
- Elektronik biletler ve faturalarla önceki seyahatlere erişim sunmak.
- Kullanıcı dostu ve basitleştirilmiş bir oturum açma deneyimi sağlamak.

1.3.2.4 Hizmet Sağlayıcıların Desteklenmesi

- Araçlar, sürücüler ve seferlerin yönetimi için entegre bir kontrol paneli sunmak.
- Operasyonel verimliliği artırmaya yönelik profesyonel raporlar sağlamak.

1.3.2.5 Satışların Artırılması:

- Hizmet sağlayıcıların daha geniş bir müşteri kitlesine ulaşmasını sağlamak.
- Pazarlama kampanyaları ve indirim kuponları oluşturma gibi özellikler eklemek.

1.3.2.6 Güvenliğin Güçlendirilmesi:

- Müşteri verilerini korumak için ileri düzey şifreleme algoritmalarını uygulamak.
- Güçlü kimlik doğrulama prosedürleri ile işlemlerin güvenilirliğini artırmak.
- Kod ve uygulama güvenliğini, en yeni teknolojilerle iyileştirmek.

1.3.2.7 Gelecekteki Genişlemeyi Garanti Altına Alma:

- Platformun esnek bir mimari ile tasarlanarak yeni hizmetlerin kolayca eklenmesini sağlamak.
- Performanstan ödün vermeden daha fazla kullanıcıyı desteklemek.

1.3.2.8 Platform Yönetiminin Sağlanması:

- Platform yöneticileri için hesaplar oluşturmak.

- Hizmet sağlayıcıların kayıt işlemlerini belirli protokollere uygun şekilde yöneticilerin onayına sunmak.
- Müşteri bilgilerine sınırlı ve kontrollü erişim sağlayarak yönetim kalitesini artırmak.

1.3.2.9 Gelecek Vizyonu ve Ek Hedefler:

- Bireysel operatörleri desteklemek için uygun politikalar geliştirmek.
- Dijital altyapıya sahip olup entegrasyon eksikliği yaşayan harici hizmet sağlayıcılarla müşteri sistemi entegrasyonunu sağlamak.
- İş Zekası (BI) sistemleri ile desteklenmiş gelişmiş bir müşteri kontrol paneli eklemek.
- Operasyonel maliyetleri karşılamak için sürdürülebilir gelir politikaları ile platformun devamlılığını sağlamak.
- SaaS hizmetlerini geliştirmek için ileri düzey abonelik paketleri tasarlamak.

1.4 Proje Kapsamı

1.4.1 Hedef Kitle:

1.4.1.1 Hizmet Sağlayıcılar:

Bu sürümde Bus E-Ticket projesi, şehirlerarası toplu taşıma hizmeti sunan ve dijital altyapıya ya da hizmetlerini sergileyecek bir platformla entegrasyona sahip olmayan özel şirketleri desteklemeye odaklanacaktır. Bireysel operatörler veya platform dışında dijital altyapıya sahip şirketler, hedefler bölümünde açıklandığı gibi gelecekteki sürümler için dahil edilmiştir. Bu nedenle, bu gruplar projenin ilk sürümünün hedef kitlesi içinde yer almayacaktır.

1.4.1.2 Müşteriler:

Müşteriler, şehirlerarası toplu taşıma hizmetlerinden faydalanan bireylerdir. Çeşitli kriterlere göre farklı gruplara ayrılabilirler:

- Cinsiyet: Erkek ve kadın.
- Meslek: Çalışanlar ve öğrenciler.
- Kullanım amacı: Turistler, ziyaretçiler veya yerel sakinler.

Bu müşteri çeşitliliği, farklı grupların ihtiyaçlarını karşılayacak esnek ve kapsamlı hizmetlerin sunulması gerekliliğini göstermektedir.

1.4.2 Temel Hizmetler:

1.4.2.1 Hizmet Sağlayıcılar için Platform Hizmetleri:

- Resmi ve sistematik kayıt: Kayıt talebi oluşturulması, gözden geçirilmesi ve onaylanmasından sonra hesap oluşturulması.
- Hesap bilgileri yönetimi: Logo, ad, şifre gibi hesap bilgilerini düzenleme.
- Kullanıcı dostu kontrol paneli: Günlük operasyonları yönetmek için esnek bir arayüz.
- Sefer yönetimi: Seferlerin kolayca oluşturulması, düzenlenmesi ve planlanması.
- Rezervasyon yönetimi: Müşteriler tarafından yapılan rezervasyonların izlenmesi ve işlenmesi.
- Yolcu ve müşteri verilerine erişim: Rezervasyon bilgilerini ve ilgili müşteri adlarını görüntüleme.
- Genel istatistikler: Şirketin satış ve rezervasyon gibi faaliyetlerine dair analiz özetleri.

1.4.2.2 Müşteriler için Platform Hizmetleri:

- Yeni hesap oluşturma: Platform hizmetlerinden yararlanmak için kişisel hesap oluşturma.
- Hesap bilgileri yönetimi: Ad, e-posta ve şifre gibi kişisel bilgilerin güncellenmesi.
- Sefer arama: Belirli kriterlere göre sefer sorgulama.
- Sefer rezervasyonu: Hem kendi hesabı için hem de kayıtsız müşteriler adına rezervasyon yapma.
- Kalkış ve varış noktalarına erişim: Harita veya yönlendirmeler aracılığıyla yerlerin belirlenmesi.
- Rezervasyon ve sefer takibi: Rezervasyon durumu ve sefer detaylarının izlenmesi.
- Geçmiş sefer ve ödemelerin sorgulanması: Önceki sefer ve ödemelere dair kayıtlara erişim.

- Fatura ve bilet erişimi: Elektronik faturaların ve biletlerin indirilmesi.

1.4.2.3 Yöneticiler için Platform Hizmetleri:

- Tüm hesapların yönetimi: Hizmet sağlayıcılar ve müşteriler için tam hesap kontrolü.
- Kayıtsız müşteri verileri sorgulama: Rezervasyonlarla ilişkili kayıtsız müşteri bilgilerine erişim.
- Platform faaliyetlerine dair kapsamlı istatistikler: Platformun genel performansını kapsayan raporların görüntülenmesi.
- Kayıt taleplerinin yönetimi: Yeni hizmet sağlayıcı kayıt taleplerinin gözden geçirilmesi ve onaylanması.
- Sefer ve rezervasyonların genel sorgulaması: Hizmet sağlayıcıların sunduğu tüm seferlerin ve müşterilerin tüm rezervasyonlarının incelenmesi.

1.4.2.4 Coğrafi Kapsam:

Bus E-Ticket platformu, ilk sürümünde büyük bir ihtiyaç ve dijital dönüşüm potansiyeline sahip olduğu için Suudi Arabistan pazarına odaklanacaktır. Ancak, platform gelecekte küresel bir ölçek için tasarlanmış olup, yerel başarıya bağlı olarak kademeli olarak diğer pazarlara genişletilebilecektir.

1.4.3 Teknik Avantajlar:

1.4.3.1 Soyutlama ve Entegrasyon:

Platform, müşteriler ve hizmet sağlayıcılar arasında yüksek soyutlama düzeyine dayanan bir teknik mimariye sahiptir:

- Hizmet Sağlayıcılar Bölümü: SaaS (Hizmet Olarak Yazılım) modeli üzerine kurulmuş bağımsız bir sistem olarak faaliyet gösterir. Bu, hizmet sağlayıcıların sefer, rezervasyon ve operasyonel bilgilerini kolaylıkla yönetmesini sağlar.
- Müşteriler Bölümü: Tamamen bağımsız olarak tasarlanmış ve B2C (işletmeden tüketiciye) modeli üzerine kurulmuştur. Tüm hizmet sağlayıcıların seferleri bu bölümde birleşerek müşterilere sunulmaktadır.
- Bu soyutlama, her bölümün bağımsızlığını ve esnekliğini garanti ederken, aynı zamanda sorunsuz bir entegrasyon sağlar.

1.4.3.2 Güvenlik:

Platform, güvenliği artırmak için en son teknolojileri kullanır:

- Güçlü bir kod yapısını garanti etmek için Nesneye Yönelik Programlama (OOP) prensipleriyle güvenli bir kod tasarımı.
- Parola şifreleme ve kişisel veri koruması için Hash algoritmalarının uygulanması.

1.4.3.3 Ölçeklenebilirlik:

Platform, gelecekteki sürümlerle kolayca genişletilebilecek şekilde tasarlanmıştır:

- SOLID Prensipleri ve Tasarım Desenlerine dayalı modern bir mimari ile arka uç tasarımı.
- Geleneksel 3-Katmanlı Mimari'den N-Katmanlı Mimariye geçiş, daha etkileşimli ve dinamik katmanlar oluşturmayı sağlar.
- Clean Architecture prensipleri ile temel sistem üzerinde değişiklik yapmadan yeni özellikler ekleme kolaylığı.

1.4.3.4 Verimlilik:

Platform, performans açısından en yüksek verimliliği sağlamak üzere tasarlanmıştır:

- Yanıt Süresinin Hızlandırılması: T-SQL sorgularını optimize ederek ve Entity Framework gibi ileri teknolojilerden faydalanarak gecikmeyi azaltma.
- Etkili Kullanıcı Arayüzü: SPA (Tek Sayfa Uygulama) teknolojileriyle hızlı ve sorunsuz bir kullanıcı deneyimi sağlama.
- Performans Testleri: Sistemin kararlılığını ve hızını sağlamak için sürekli performans testleri ve iyileştirme stratejileri.

1.4.3.5 Kapsamın Sınırları:

Projenin ilk aşaması, dijital altyapıdan yoksun hizmet sağlayıcılara odaklanarak onların ihtiyaçlarını karşılayacak kapsamlı dijital çözümler sunmayı hedeflemektedir. Dijital altyapıya sahip olan şirketler, platformun gelecekteki sürümlerine dahil edilecek ve bu şirketlerin sistemlerini platformla entegre ederek verimlilik artırılacaktır.

1.5 Kısaca Metodoloji

1.5.1 Pazar Analizi ve İhtiyaç Belirleme:

Hedef pazarlara ait resmi istatistikler incelenmiş ve özellikle toplu taşıma alanında dijital hizmetlerden yoksun olan pazarlar üzerinde durulmuştur. Yerel ve küresel uygulamalar detaylı bir şekilde analiz edilerek en iyi uygulama örnekleri çıkarılmıştır.

- Gereksinimlerin ve Paydaşların Belirlenmesi:

Süreç, sistemin temel gereksinimlerini belirlemek, müşteri ve hizmet sağlayıcı ihtiyaçlarını analiz etmekle başlamıştır. Paydaşlar analiz edilmiş ve gereksinimleri belgelendirilerek projenin net ve kapsamlı bir vizyonu oluşturulmuştur.

- Planlama ve Tasarım:

Projenin planlamasını ve yapısını belgelemek için Use Case, UML Class, UML Package ve ERD (Schema Diagram) gibi şemalar hazırlanmıştır.

1.5.2 Teknik Geliştirme:

1.5.2.1 Programlama Dilleri ve Çerçeveler:

- C# ve .NET Core API: Arka uç katmanını geliştirmek için kullanılmıştır.
- JavaScript ve React JS: Etkileşimli ve hızlı bir kullanıcı arayüzü oluşturmak için kullanılmıştır.
- CSS, HTML ve Bootstrap: Arayüz tasarımı için yardımcı teknolojiler olarak kullanılmıştır.

1.5.2.2 Veritabanı ve Veri Yönetimi:

- MSSQL: Ana veritabanı olarak kullanılmıştır.
- Entity Framework ve ADO.NET: Verilerin esnek ve hızlı bir şekilde işlenmesi için entegrasyon sağlanmıştır.
- T-SQL: Karmaşık veritabanı işlemlerini optimize etmek ve hız artırmak için kullanılmıştır.

1.5.2.3 SOLID Prensiplerinin Uygulanması:

Kodun temiz, genişletilebilir ve sürdürülebilir olmasını sağlamak için aşağıdaki SOLID prensipleri benimsenmiştir:

- SRP (Single Responsibility): Her sınıf yalnızca bir sorumluluğa sahip olacak şekilde tasarlanmıştır.
- OCP (Open/Closed): Kod, yeni özellikler için açık; mevcut özelliklerin değiştirilmesine kapalıdır.
- LSP (Liskov Substitution): Alt sınıflar, üst sınıfların yerine sorunsuzca kullanılabilir.
- ISP (Interface Segregation): Büyük ve genel arayüzler yerine küçük ve özel arayüzler tercih edilmiştir.
- DIP (Dependency Inversion): Somut sınıflar yerine soyutlamalara bağımlılık sağlanmıştır.

1.5.2.4 Tasarım Desenlerinin Uygulanması:

- Repository Pattern: Veri işlemleri için soyutlama sağlanmıştır.
- Unit of Work Pattern: Birden fazla veri işlemini tek bir birim olarak yönetmek için uygulanmıştır.
- Dependency Injection (DI): Kodun test edilebilirliğini artırmak ve bağımlılıkları ayırmak için kullanılmıştır.

1.5.2.5 Kullanıcı Deneyiminin Geliştirilmesi:

- PWA (Progressive Web Apps): Akıllı telefon uygulamalarına benzer bir deneyim sağlamak için kullanılmıştır.
- SPA (Single Page Application): Web sitesinin performansını artırmak ve yükleme sürelerini azaltmak için uygulanmıştır.

1.5.2.6 Performans ve Ölçeklenebilirlik:

- 3-Katmanlı Mimari (3-Tier Architecture): Geleneksel bir yapı olarak benimsenmiş ve N-Katmanlı Mimari kavramlarıyla geliştirilmiştir. Bu, ölçeklenebilirliği artırmıştır.

1.5.2.6 Güvenlik:

- Parola şifrelemesi için güçlü Hash algoritmaları kullanılmıştır.
- Kod ve veri güvenliği, Nesneye Yönelik Programlama (OOP) prensipleri ile artırılmıştır.

1.5.2.7 Esnek Altyapı:

- Çoklu platform desteği sağlayabilecek güçlü bir API sistemi geliştirilmiştir (şu anda web sitesi, gelecekte mobil uygulamalar).
- Arka uç yapısı, gelecekte dış sistemlerle entegrasyona uygun şekilde tasarlanmıştır.

Bu yaklaşım, ilk sürümde güçlü bir dijital platform sunarken, gelecekteki gereksinimlerin karşılanmasına ve kolayca genişletilebilir bir yapıya imkan tanımaktadır.

2. PROJE ARKA PLANI VE GEREKÇELER

2.1 Sorunun Arka Planı ve Analizi

2.1.1 Şehirlerarası Toplu Taşımaya Genel Bakış

Şehirlerarası toplu taşıma, bölgeler arası uzun mesafeli ulaşım ihtiyaçlarını karşılamayı hedefleyen hayati bir sistemdir. Bu sistem genellikle kamuya veya özel sektöre ait otobüsler tarafından işletilir ve büyük şehirler ile kırsal bölgeleri birbirine bağlayan hatlardan oluşur [4].

Toplu taşımanın önemi şu unsurlara dayanır:

- Kişisel araçlara bağımlılığı azaltma: Trafik sıkışıklığını ve çevreye zararlı gaz emisyonlarını azaltır.

- Uygun maliyet: Araç sahibi olma veya araç kiralama maliyetlerine kıyasla daha hesaplıdır, bu da özellikle orta ve düşük gelirli bireyler için ideal bir seçenektir [4].
- Çeşitli kullanıcı gruplarına hizmet: İşçiler, öğrenciler, iş seyahati yapanlar ve turistler gibi farklı kitlelerin iş, eğitim ve temel hizmetlere erişimini kolaylaştırır [4].

Operasyonel açıdan, şehirlerarası toplu taşıma sistemleri esnek yapıları sayesinde değişen taleplere uyum sağlar. Merkezi duraklar ve stratejik noktalarda bulunan ara duraklar sayesinde kullanıcıların ulaşımı kolaylaştırılır [4].

2.1.2 Toplu Taşımada Gelişim Süreci

2.1.2.1 Şehirlerarası Toplu Taşımanın Başlangıcı:

İlk şehirlerarası toplu taşıma, hayvanlar ve ilkel arabaların kullanımıyla başlamıştır. 19. yüzyıldaki sanayi devrimi sırasında kömür ve buharla çalışan otobüslerin geliştirilmesiyle toplu taşıma sistemleri daha verimli hale gelmiştir [5]. Daha sonra içten yanmalı motorların devreye girmesi, taşımacılığın hızını ve verimliliğini önemli ölçüde artırmıştır [4].

2.1.2.2 Toplu Taşımadaki Gelişim Aşamaları:

- Geleneksel Taşımacılık: Hayvanların çektiği arabalarla yapılan taşımacılık [5].
- Sanayi Devrimi: Kömür ve buharla çalışan otobüslerin kullanımı. Bu yenilikler, seyahat süresini kısaltıp güvenilirliği artırmıştır [4].
- Organizasyon ve Entegrasyon: Artan taleple birlikte daha organize taşıma ağları ve aktarma sistemleri geliştirilmiştir [6].
- Teknolojik Devrim: Elektronik rezervasyon sistemleri, akıllı uygulamalar ve GPS gibi teknolojilerin kullanımıyla hizmet kalitesi ve kullanıcı deneyimi artırılmıştır [5][7].

2.1.3 Geleneksel Toplu Taşıma Zorlukları

2.1.3.1 Rezervasyon Sistemlerinin Manuel Olması:

Geleneksel toplu taşıma sistemleri genellikle rezervasyon işlemlerini manuel yöntemlerle yürütmektedir. Bu durum, yavaşlık, insan hatası ve veri yönetimindeki zorluklar gibi sorunlara yol açar. Elektronik rezervasyon sistemlerinin kullanımı, operasyonel verimliliği artırabilir ve maliyetleri düşürebilir [1][2].

2.1.3.2 İç Operasyonların Yönetimindeki Zorluklar:

Geleneksel sistemlerde dijital araçların eksikliği, operasyonel performansın izlenmesini ve üretkenliğin artırılmasını zorlaştırır. İnsan kaynakları ve zaman çizelgelerinin yönetimi, dijital çözümler olmadan karmaşık hale gelir [2][3].

2.1.3.3 Şirketler Arası Entegrasyon Eksikliği:

Toplu taşıma şirketlerinin birbirinden bağımsız çalışması, seferlerin ve hizmetlerin bağlantısını zorlaştırarak verimliliği düşürür ve kullanıcı deneyimini olumsuz etkiler [6][7].

2.1.3.4 Müşterilere Ulaşmadaki Zorluklar:

Manuel yöntemlere bağlı kalınması, müşteri davranışlarının analiz edilmesini ve yeni kullanıcıların çekilmesini zorlaştırır. Büyük veri ve akıllı uygulamalar, müşteri davranışlarını analiz ederek ihtiyaçlarına uygun hizmetler sunulmasını kolaylaştırabilir [1][4].

2.1.4 Toplu Taşıma Talebindeki Artış (Suudi Arabistan Örneği)

Suudi Arabistan'da şehirlerarası toplu taşıma talebinde belirgin bir artış gözlemlenmiştir. 2022 yılında, otobüslerle taşınan yolcu sayısı 43.5 milyona ulaşmış ve bu sayı 2021 yılına kıyasla %233.9 artış göstermiştir [8]. Bu büyüme, Suudi Arabistan 2030 Vizyonu kapsamında toplu taşımanın geliştirilmesine yönelik çabaları yansıtmaktadır.

2.1.4.1 Artan Talebe Yanıt Olarak Yeni Şirketlerin Kurulması:

Ekim 2023'te, Suudi Arabistan Ulaştırma Bakanı tarafından 200'den fazla şehir ve ilçeyi birbirine bağlayan otobüs hizmetleri projesi başlatılmıştır. Bu projede yer alan başlıca şirketler:

- Darb Watan: Günde 124 seferle 75 şehir ve ilçeyi bağlar.
- North West Bus: Günde 190 seferle 70'ten fazla şehir ve ilçeyi kapsar.
- SAT: Güney bölgesinde 178 günlük seferle 27 rotayı kapsar [9].

2.1.4.2 Şirketler Arası Entegrasyon Eksiklikleri:

Her şirketin ayrı bir dijital platformu olması ve bu platformların birbiriyle entegre olmaması, hizmetlerin tekrarlanmasına ve kaynakların verimli kullanılmamasına yol açmaktadır. Ayrıca, manuel sistemlere bağımlılık operasyonel verimliliği düşürmekte ve insan hatalarına neden olmaktadır [9].

2.1.5 Dijital Dönüşümün Önemi

2.1.5.1 Dijitalleşmenin Getirileri:

Dijital dönüşüm, toplu taşımada şu alanlarda önemli iyileştirmeler sağlar:

- Entegrasyonun Artırılması: Operatörlerin birbirine bağlanmasıyla müşteri deneyiminin geliştirilmesi.
- Anlık Veri Sağlama: Müşterilere gerçek zamanlı sefer bilgileri sunarak seyahat planlamasını kolaylaştırma.
- Operasyonel Maliyetlerin Azaltılması: Süreçlerin otomasyonu ve yapay zeka gibi teknolojilerle verimliliğin artırılması [1][2].

2.1.5.2 Diğer Ülkelerde Başarılı Dijitalleşme Örnekleri:

- Türkiye (O-Bilet ve EnUcuz): Bu uygulamalar, kullanıcıların farklı seçenekleri karşılaştırmasını ve bilet satın almasını sağlayarak entegrasyonu artırmıştır[11].
- Almanya/Avusturya/İsviçre (Verkehrsverbund): Akıllı uygulamalarla desteklenen entegre bir toplu taşıma sistemi oluşturulmuştur [6].

2.1.6 Dijitalleşmenin Eksikliği Durumunda Etkiler

Müşterilere Etkisi:

- Rezervasyon Karmaşıklığı: Ofis ziyareti veya telefon görüşmesi gibi manuel işlemler, müşteri deneyimini zorlaştırır [4].
- Anlık Bilgi Eksikliği: Güzergah ve saat bilgilerine erişimin zorluğu, müşteri memnuniyetini düşürür [11].

Hizmet Sağlayıcılara Etkisi:

- Müşteri Kaybı: Dijital platformlara sahip olmayan sağlayıcılar, müşterileri daha modern alternatiflere kaptırabilir [6].
- Operasyonel Yönetim Zorluğu: Manuel sistemler, operasyonel süreçlerin verimli yönetilmesini engeller [3].

2.1.7 Dijital Çözüm İhtiyacı

Bus E-Ticket Platformu, dijital altyapı eksikliğini gidererek hem sağlayıcıların operasyonlarını iyileştirmeyi hem de müşterilere kolaylık sunmayı amaçlamaktadır.

Hizmet Sağlayıcılara Çözümler:

- Dijital yönetim araçları ile verimliliği artırma.
- Ödeme ve rezervasyon süreçlerini otomatikleştirme.

Müşterilere Çözümler:

- Gelişmiş arama ve rezervasyon özellikleri.
- Elektronik ödeme ve hızlı rezervasyon imkanı.

2.2 Paydaşlar

Paydaşların Tanımlanmasının Önemi

Proje gereksinimlerinin net bir şekilde anlaşılması, paydaşların doğru tanımlanmasını ve gereksinimlerin hassas bir şekilde toplanmasını sağlar. Bu süreç, nihai ürünün paydaşların ihtiyaçlarını karşılamasını garanti eder. Paydaşların tanımlanmasının faydaları:

- Kaynakların Verimli Kullanımı: Gereksiz özelliklere kaynak harcanmasının önüne geçilir.

- Beklentilerin Yönetimi: Paydaşların beklentileri belirlenerek, proje hedefleriyle uyumsuzluk riski en aza indirilir.
- İşbirliğini Artırma: Farklı ekipler (teknik, idari ve hizmet sağlayıcılar) arasında işbirliği ortamı yaratılır.
- Başarı Şansını Artırma: Paydaşlarla sürekli iletişim kurularak, potansiyel zorluklar erken aşamada tespit edilir ve bunlara uyum sağlanır.
- Performans ve Çıktıların Değerlendirilmesi: Her bir paydaş için net başarı kriterleri belirlenir ve proje değerlendirme süreçleri netleştirilir [13].

2.2.1 Paydaşların Tanımlanması

2.2.1.1 Platform Kullanıcıları (End Users):

- Detaylı müşteri grupları tanımlanır.
- Projeden nasıl yararlandıkları açıklanır.

2.2.1.2 Hizmet Sağlayıcılar (Service Providers):

- Yolculuk sunumundaki rolleri tanımlanır.
- Projenin operasyonlarına nasıl destek verdiği açıklanır.

2.2.1.3 Proje ve Platform Yönetimi (Platform Management):

- Sistemi yöneten teknik ve idari ekip.
- Sistem performansı ve kalitesini sürdürülebilir kılmak için çalışır.

2.2.1.4 Düzenleyici ve Hükümet Kurumları:

Standartlar oluşturur ve projenin yasalara uygunluğunu sağlar.

2.2.1.5 Yatırımcılar ve Stratejik Ortaklar:

Mali ve teknik destek sunarlar.

2.2.1.6 Teknoloji Sağlayıcıları (Technical Providers):

Yazılım geliştirme ve barındırma hizmetleri sağlarlar.

2.2.2 Paydaşların Rol ve Önemi Tablosu

Tablo 2.1 Paydaşların Rol ve Önemi

Kategori	Rol	Projeye Önemi	Kategori
Müşteriler	Platformu kullanarak rezervasyon yapar.	Kullanıcı sayısını artırarak projenin başarısını sağlar.	Müşteriler
Hizmet Sağlayıcılar	Seyahat hizmetleri sunar ve operasyonları yönetir.	Platformda sunulan hizmetlerin sürekliliğini sağlar.	Hizmet Sağlayıcılar
Platform Yönetimi	Sistemi geliştirir ve bakımını yapar.	Performans ve kaliteyi korur.	Platform Yönetimi
Hükümet Kurumları	Yasalara uyumu sağlar.	Güvenilirlik ve yasal uyum sağlar.	Hükümet Kurumları
Yatırımcılar	Projeyi finanse eder.	Finansal ve teknik kaynakların sağlanmasını temin eder.	Yatırımcılar
Teknoloji Sağlayıcıları	Sistem geliştirme ve destek sağlar.	Operasyonel verimliliği artırır.	Teknoloji Sağlayıcıları

2.3 Benzer Projeler

2.3.1 BUS E-TICKET Platform'a Benzer Projelerin Başarısı

2.3.1.1 SaaS Modelindeki Projelerin Başarısı:

Yazılım hizmeti (Software as a Service - SaaS), uygulamaların yerel olarak yüklenmesi yerine, internet üzerinden erişilip yönetilmesini sağlayan bir modeldir. Bu model, özellikle maliyetleri düşürmek ve operasyonel verimlilik sağlamak isteyen küçük ve orta ölçekli işletmeler için uygundur [2].

Örnekler:

- Salesforce: Şirketlerin müşteri ilişkilerini ve satış süreçlerini yönetmesini sağlayan bir CRM platformu.
- Google Workspace: Gmail, Google Drive ve Google Docs gibi iş araçlarını bulut üzerinden sunarak iş birliğini kolaylaştırır.

- O-Bilet: Şehirlerarası ulaşım şirketleri için elektronik rezervasyon ve sistem entegrasyonu sağlayan bir SaaS platformu [11].

SaaS Modelindeki Büyüme ve Başarı:

- Küresel Pazar Hacmi: SaaS harcamalarının 2024 yılında 247,2 milyar dolara ulaşması beklenmektedir, bu bir önceki yıla göre %20'lik bir artışı temsil eder [14].
- Şirket Sayısı: Dünya genelinde 42.000'den fazla SaaS şirketi bulunmaktadır; bunların 12.400'ü ABD'dedir [14].
- Uygulama Kullanımı: Ortalama bir şirket, 2017'de yalnızca 16 SaaS uygulaması kullanırken, bugün bu sayı 112'ye çıkmıştır [14].

2.3.1.2 B2C Modelindeki Projelerin Başarısı:

İşletmeden tüketiciye (Business to Consumer - B2C) iş modeli, ürün veya hizmetlerin doğrudan son kullanıcılara sunulmasını ifade eder. Bu model, dijital etkileşim ve kullanım kolaylığı üzerine odaklanır [12].

Örnekler:

- Booking.com: Küresel konaklama ve ulaşım seçeneklerini müşterilere sunan bir rezervasyon platformu [12].
- Amazon: Hızlı teslimat hizmetleri ve kişiselleştirilmiş alışveriş deneyimi sunan bir e-ticaret devi.
- Uber: Uygulama aracılığıyla müşterilerle sürücülerini bir araya getirerek doğrudan ulaşım hizmeti sunar.

B2C Modelindeki Büyüme ve Başarı:

- Küresel Pazar Hacmi: 2023 yılında küresel B2C e-ticaret pazar büyüklüğü 5.47 trilyon dolar olarak tahmin edilmiştir ve 2024-2030 yılları arasında %19,1'lik bir yıllık bileşik büyüme oranı beklenmektedir [15].
- Kategoriler: Giyim ve ayakkabı, 2023 yılında gelirlerin %21,1'ini oluşturmuştur ve büyümeye devam etmektedir [15].
- Bölgesel Dağılım: 2023 yılında Asya-Pasifik bölgesi, toplam B2C gelirlerinin %58'ini oluşturmuş ve yıllık %20,0 büyüme oranıyla lider konumda kalmıştır [15].

2.3.1.3 SaaS & B2C Modellerini Birleştiren Projeler:

SaaS ve B2C modellerini birleştiren projeler, işletmelere yazılım hizmetleri (SaaS) sunarken, bu işletmelerin son tüketicilere (B2C) erişimini de sağlar. Bu modelde, işletmeler platform üzerinde hesap açarak hizmetlerini doğrudan müşterilere sunar. BUS E-TICKET Platform, bu modeli benimseyerek hem SaaS hesapları ile hizmet sağlayıcılara dijital altyapı sunar hem de bu hizmetleri son kullanıcılara ulaştırır.

Örnekler:

- Shopify: İşletmelerin çevrimiçi mağaza oluşturmaya olanak tanıyan bir SaaS platformudur. Şirketler, ürünlerini doğrudan tüketicilere (B2C) sunabilir.

Başarı ve Büyüme:

2023 Yılı Gelirleri: 7.06 milyar dolar, bir önceki yıla göre %26,1'lik bir artış.

Toplam Satış Hacmi (GMV): 2024'ün üçüncü çeyreğinde 62 milyar dolara ulaşarak yıllık %16'lık bir artış göstermiştir [17].

2.3.2 Benzer Projeler ile BUS E-TICKET Platform'un Karşılaştırması

Tablo 2.2, BUS E-TICKET Platform'un aynı sektörde faaliyet gösteren diğer projelerden nasıl farklılaştığını çeşitli yönlerden ele alarak açıklamaktadır. Aşağıdaki tabloda, projeler arasındaki farklar entegrasyon, kullanıcı deneyimi, hedef kitle, kullanıcı arayüzü, iş modeli, dil desteği, özel otobüs kullanım durumu ve platform yapısı açısından detaylandırılmıştır.

Tablet 2.2. BUS E-BİLET Benzer projelerin karşılaştırma tablosu

Entegrasyon	Kullanıcı Deneyimi	Hedef Kitle	Kullanıcı Arayüzü	Model	Diller	Özel Otobüsler	Platformlar	Proje
Yüksek Entegrasyon	Mükemmel, hızlı ve verimli	Hizmet sağlayıcılar ve müşteriler	İyi	SaaS & B2C	İngilizce	Yok	WEB + PWA Desteği	BUS E-TICKET Platform
Entegrasyon Eksikliği	Kötü, yavaş, SPA desteği yok	Müşteriler	Kabul edilebilir	B2C	İngilizce, Yerel	Var	WEB	Darb Watan
Entegrasyon Eksikliği	Kötü, yavaş, SPA desteği yok	Müşteriler	Kabul edilebilir	B2C	İngilizce, Yerel	Var	WEB	North West Bus
Yüksek Entegrasyon	Kabul edilebilir, yavaş, SPA desteği yok	Müşteriler	İyi	B2C	İngilizce, Yerel	Yok	WEB + Mobil Uygulama	O-Bilet

2.3.3 Sonuç

BUS E-TICKET Platform, toplu taşıma pazarının artan ihtiyaçlarını karşılamak için SaaS ve B2C modellerini birleştiren yenilikçi bir teknik çözümdür. Yüksek entegrasyon, mükemmel kullanıcı deneyimi ve esnek, ölçeklenebilir bir altyapı gibi

özellikleri sayesinde, projenin operasyonel verimliliği artırmada ve müşteri memnuniyetini yükseltmede önemli bir başarı elde etmesi beklenmektedir. Platform, yerel ve uluslararası düzeyde güçlü bir genişleme potansiyeline sahip olup, toplu taşıma sektöründe dijital dönüşümü destekleyen sürdürülebilir ve etkili bir yatırım olarak öne çıkmaktadır.

2.4 Proje Gerekçeleri

2.4.1 Toplu Taşıma Sektöründe Dijital Dönüşüm İhtiyacı

- Şehirlerarası toplu taşıma sektörü, dijital altyapının eksikliği nedeniyle önemli operasyonel zorluklarla karşı karşıyadır. Rezervasyon ve operasyon yönetiminde manuel sistemlerin kullanımı, insan hatalarına, düşük verimliliğe ve veri yönetiminde etkinlik eksikliğine yol açmaktadır.
- Artan toplu taşıma talebi, hem müşterilerin hem de hizmet sağlayıcıların ihtiyaçlarını karşılamak için dijital dönüşümü zorunlu hale getirmiştir.

2.4.2 Hizmet Sağlayıcılar Arasında Entegrasyonu Destekleme

- Toplu taşıma hizmeti sağlayıcıları arasında entegrasyonun olmaması, müşterilerin esnek ve tek bir sistemden hizmet almasını zorlaştırmaktadır.
- Proje, hizmet sağlayıcıları merkezi bir dijital ortamda birleştirerek, kullanıcıların erişimini kolaylaştırmayı ve şirketlerin operasyonel verimliliğini artırmayı hedeflemektedir.

2.4.3 Müşteri Deneyimini İyileştirme

- Müşteriler, geleneksel rezervasyon yöntemlerinin karmaşıklığı ve anlık seyahat bilgilerine erişimin olmaması gibi zorluklarla karşılaşmaktadır.
- Proje, müşterilere elektronik bilet rezervasyonu, seyahat takibi ve rezervasyon yönetimi gibi özellikler sunarak kullanım deneyimini iyileştirir ve müşteri memnuniyetini artırır.

2.4.4 Suudi Arabistan 2030 Vizyonu'na Destek

- Suudi Arabistan'da toplu taşıma hizmetlerine yönelik talepte büyük bir artış gözlemlenmektedir.

- Proje, Suudi Arabistan 2030 Vizyonu'nun toplu taşıma altyapısının dijitalleşmesi ve geliştirilmesi hedefleriyle uyumludur.
- Yeni şirketlerin, örneğin "Darb Watan", "North West Bus" ve "SAT", kurulması, bu sektörde sürdürülebilir büyümeyi destekleyecek entegre teknik çözümlere olan ihtiyacı vurgulamaktadır [9].

2.4.5 SaaS ve B2C Modellerinin Başarılarından Yararlanma

- Proje, düşük maliyetle gelişmiş teknik çözümler sunan başarılı bir SaaS (Software as a Service) modeline dayanmaktadır.
- Bireyler için esnek ve hızlı hizmetler sunarak, platformun benimsenmesini ve müşteri tabanının genişlemesini sağlar.

2.4.6 Katma Değer

Proje, performans raporları, dijital entegrasyon ve elektronik ödeme gibi gelişmiş araçlar sunarak operasyonel verimliliği artırır ve şirketlerin operasyonel maliyetlerini azaltır.

2.4.7 Ekonomik Fırsatlar ve Gelecek Genişleme Potansiyeli

- Projenin gelecekteki genişleme potansiyeli, SaaS ve B2C modellerinin küresel büyümesiyle uyumludur.
- Projenin, yenilikçi ve ölçeklenebilir çözümler sunarak yeni yatırımları çekmesi ve dijital ekonomiyi desteklemesi beklenmektedir [1].

3. TASARIM VE ANALİZ

3.1 Gereksinim Analizi

3.1.1 Fonksiyonel Gereksinimler

3.1.1.1 Hesaplar

- Platform üç tür hesap türü içerir: yöneticiler, müşteriler ve hizmet sağlayıcılar.

- Müşteriler, yeni bir hesap oluşturarak veya sosyal medya entegrasyonu ile hızlı kaydı kullanarak platforma katılabilir.
- Hizmet sağlayıcılar, platform üzerinden bir form doldurup talep göndererek katılır; onaylandıktan sonra aktif bir hesap oluşturulur.
- Yalnızca yönetici, yönetici hesapları oluşturabilir.
- Her hesap türü, izin verilen sınırlar içinde kendi hesap bilgilerini yönetebilir.
- Yöneticiler, hesapları geniş kapsamlı şekilde yönetebilir.

3.1.1.2 Seyahatler

- Hizmet sağlayıcılar, yeni seyahatler ekleme, mevcut seyahatleri düzenleme ve silme gibi işlemleri yönetebilir.
- Yöneticiler, tüm seyahat bilgilerine erişebilir.
- Kayıtlı müşteriler ve ziyaretçiler (platforma üye olmayan kullanıcılar), seyahat bilgilerini sorgulayabilir.
- Seyahat bilgileri, hizmet sağlayıcı bilgilerini içerir.

3.1.1.3 Rezervasyonlar

- Hizmet sağlayıcılar, rezervasyonları sorgulayabilir ve rezervasyon yapan müşterilerin bilgilerine sınırlı erişim sağlayabilir.
- Müşteriler, kendileri veya başka bir kişi için rezervasyon yapabilir.
- Yöneticiler, tüm rezervasyonlara geniş erişim sağlayabilir.
- Müşteriler rezervasyonlarına hesapları üzerinden veya rezervasyon sorgulama sayfası aracılığıyla erişebilir.

3.1.1.4 Güvenli Ödeme

- Platform, en az bir güvenli ve yaygın ödeme yöntemini destekler.

3.1.1.5 Ödeme Sorgulama

- Kayıtlı müşteriler, geçmiş ödemelerine erişebilir.
- Hizmet sağlayıcılar, platform içindeki tüm ödemeleri sorgulayabilir.

3.1.1.6 Faturalar ve Biletler

- Rezervasyon tamamlandığında, müşteriye bir fatura ve bilet oluşturulur.
- Müşteriler, geçmiş seyahatlere ait fatura ve biletlerine erişebilir.

- İlgili devlet kurumları, faturalara erişebilir.

3.1.2 Fonksiyonel Olmayan Gereksinimler

3.1.2.1 Performans

- Sistem yanıt süresi, normal koşullarda 3 saniyeyi geçmemelidir.
- Sistem, 500-1000 aktif kullanıcının eş zamanlı taleplerini performans kaybı olmadan desteklemelidir.
- Veritabanı sorguları, T-SQL gibi teknolojilerle optimize edilerek hızlı yanıt sağlanmalıdır.

3.1.2.2 Ölçeklenebilirlik

- Sistem, gelecekteki özelliklerin kolayca eklenmesini sağlamak için esnek bir mimariye (ör. N-Tier Architecture) sahip olmalıdır.
- Tüm hizmetler sunucu tabanlı olmalı ve farklı platformlara (web dışındaki) kolayca genişletilebilmelidir.

3.1.2.3 Güvenlik

- Tüm hassas veriler (şifreler, ödemeler) güçlü şifreleme algoritmaları (Hashing) kullanılarak şifrelenmelidir.
- Güvenli kayıt için iki aşamalı doğrulama (2FA) desteklenmelidir.
- Şifre değişikliği iki adımlı bir süreçle gerçekleştirilmelidir.
- Hizmet sağlayıcıların verilere erişimi sınırlı politikalarla kontrol edilmelidir.
- SQL Injection ve XSS gibi saldırıları önlemek için giriş doğrulaması yapılmalıdır.
- API'ler, Tokens veya OAuth kullanılarak güvence altına alınmalıdır.

3.1.2.4 Kullanım Kolaylığı

- Kullanıcı arayüzü (UI), SPA gibi modern tasarım standartlarıyla sade ve etkileşimli bir şekilde tasarlanmalıdır.
- Kullanıcı deneyimi (UX), hız ve sistem içinde kolay gezinme gibi özelliklerle desteklenmelidir.
- UI, masaüstü, mobil ve tablet cihazlarla uyumlu olmalıdır.

- Etkileşimli bildirimler (rezervasyon onayı, ödeme bildirimleri) ile kullanıcı etkileşimi kolaylaştırılmalıdır.
- Platform, mobil cihazlarda kullanıcı deneyimini artırmak için PWA teknolojisini desteklemelidir.

3.1.2.5 Uyumluluk

- Sistem, Chrome, Edge, Firefox, Safari gibi tüm modern tarayıcılarda çalışmalıdır.
- Daha geniş bir erişim sağlamak için İngilizce dil desteği sunulmalıdır.

3.1.2.6 Bakım

- Kod, SOLID prensiplerine uygun olarak yazılmalı ve bu sayede bakım kolaylığı sağlanmalıdır.
- Hata kayıtları (Logs) açık bir şekilde kaydedilmeli ve gerektiğinde hızlı problem çözümü için erişilebilir olmalıdır.
- Performansı koruyarak sistemin iyileştirilmesi ve yeni özelliklerin eklenmesi için düzenli güncellemeler sağlanmalıdır.

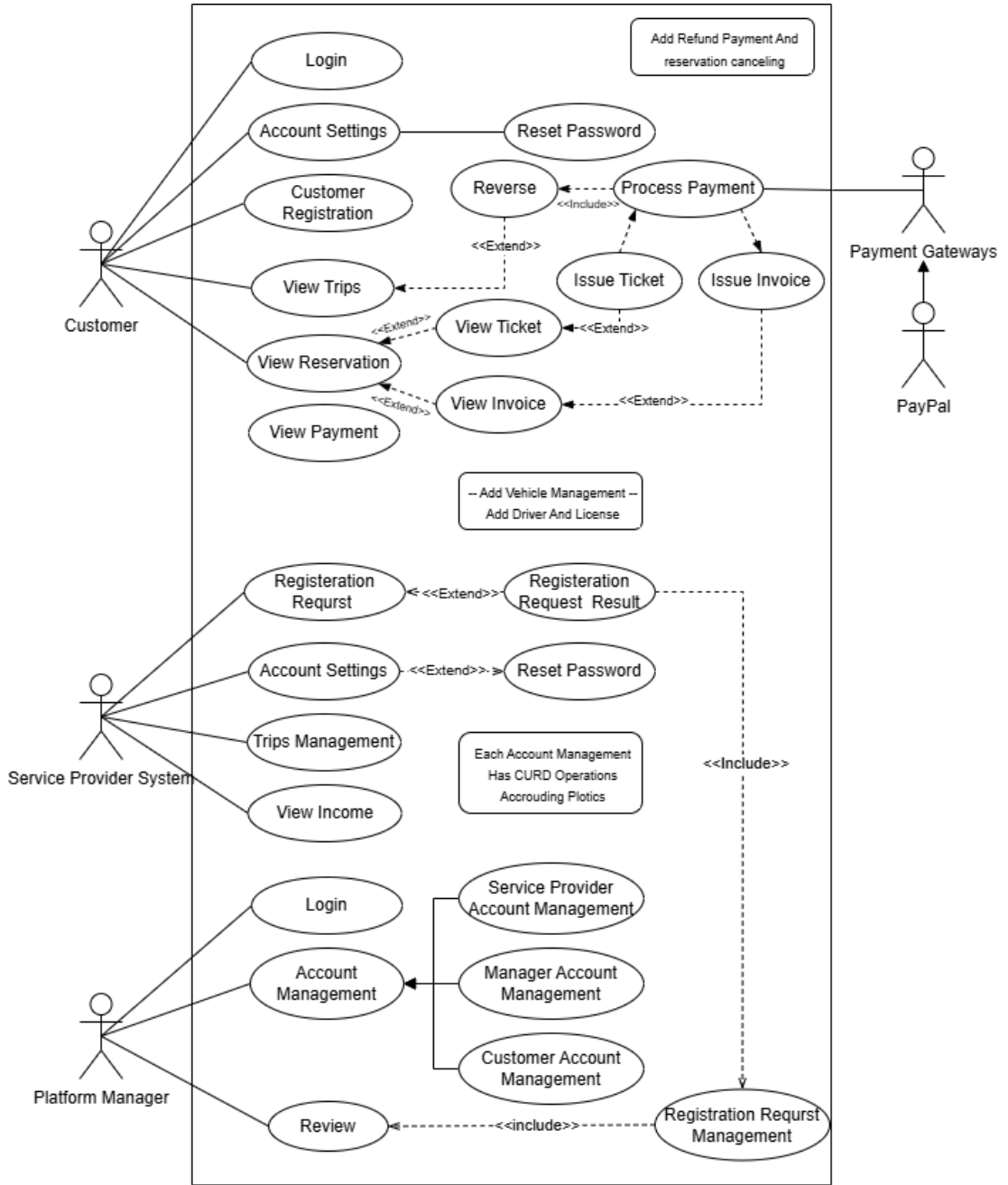
3.1.2.7 1Verimlilik

- Sık okuma işlemlerini hızlandırmak için önbellek (Caching) teknolojisi kullanılmalıdır.
- Veritabanı, artan veri hacmiyle performansı koruyacak şekilde tasarlanmalıdır.

3.1.2.8 Taşınabilirlik

- Sistem, kodda büyük değişikliklere ihtiyaç duymadan farklı sunuculara taşınabilir olmalıdır.
- Azure veya AWS gibi bulut tabanlı sunucular üzerinde çalışmaya uygun olmalıdır.

3.1.3 Use Case Diagram



Durum Adı	Aktör	Senaryo
Giriş Yap (Login)	Müşteri / Hizmet Sağlayıcı / Yönetici	Kullanıcı, kimlik bilgilerini girerek sisteme giriş yapar.
Hesap Ayarları (Account Settings)	Müşteri / Hizmet Sağlayıcı	Kullanıcı, hesap ayarlarını güncelleyerek kişisel bilgilerini düzenler.
Şifre Sıfırlama (Reset Password)	Müşteri / Hizmet Sağlayıcı	Kullanıcı, şifre sıfırlama talebinde bulunur ve bu işlem hesap ayarları üzerinden gerçekleştirilir.
Müşteri Kaydı (Customer Registration)	Müşteri	Müşteri, gerekli bilgileri doldurarak ve onaylayarak sisteme kayıt olur.
Seyahat Görüntüleme (View Trips)	Müşteri	Kullanıcı, mevcut seyahatleri görüntüler. Bu durum, Rezervasyon Görüntüleme (View Reservation) durumunu da kapsar.

Rezervasyon Görüntüleme (View Reservation)	Müşteri	Kullanıcı, hesabına bağlı rezervasyon detaylarını görüntüler. Bu durum, Bilet Görüntüleme (View Ticket) durumunu da içerir.
Ödeme Görüntüleme (View Payment)	Müşteri	Kullanıcı, hesabına bağlı önceki ödemelerin detaylarını görüntüler.
Ödeme İşleme (Process Payment)	Müşteri	Kullanıcı, ödeme işlemini gerçekleştirir. Bu işlem, Bilet Düzenleme (Issue Ticket) durumunu içerir.
Rezervasyon İptali / Düzenleme (Reverse)	Müşteri	Kullanıcı, belirli bir rezervasyonu iptal eder veya değiştirir.
Bilet Düzenleme (Issue Ticket)	Sistem	Ödeme tamamlandıktan sonra müşteriye bilet düzenlenir ve Bilet Görüntüleme (View Ticket) durumuna bağlanır.
Fatura Düzenleme (Issue Invoice)	Sistem	Ödeme tamamlandıktan sonra bir fatura düzenlenir ve Fatura Görüntüleme (View Invoice) durumuna bağlanır.
Bilet Görüntüleme (View Ticket)	Müşteri	Kullanıcı, düzenlenen biletin detaylarını görüntüler.
Fatura Görüntüleme (View Invoice)	Müşteri	Kullanıcı, rezervasyonuna bağlı faturanın detaylarını görüntüler.
Kayıt Talebi (Registration Request)	Hizmet Sağlayıcı	Hizmet sağlayıcı, sisteme katılmak için kayıt talebinde bulunur. Bu durum, Kayıt Talebi Sonucu (Registration Request Result) durumuna bağlanır.
Kayıt Talebi Sonucu (Registration Request Result)	Sistem	Sistem, hizmet sağlayıcı tarafından yapılan kayıt talebinin sonucunu görüntüler.
Seyahat Yönetimi (Trips Management)	Hizmet Sağlayıcı	Hizmet sağlayıcı, seyahatleri ekleme, düzenleme veya silme işlemleriyle yönetir.
Gelir Görüntüleme (View Income)	Hizmet Sağlayıcı	Hizmet sağlayıcı, hesabına bağlı finansal gelirleri görüntüler.
Hesap Yönetimi (Account Management)	Yönetici	Yönetici, kullanıcı hesaplarını yönetir. Bu durum, Hizmet Sağlayıcı Hesap Yönetimi (Service Provider Account Management) ve Müşteri Hesap Yönetimi (Customer Account Management) işlemlerini kapsar.
İnceleme (Review)	Yönetici	Yönetici, hizmet sağlayıcıların kayıt taleplerini inceler. Bu durum, Kayıt Talebi Yönetimi (Registration Request Management) durumunu içerir.

3.2 Sistem Mimarisi

3.2.1 Yazılım Mimarisi

3.2.1.1 Yazılım Mimarisi Tanımı

Yazılım mimarisi, bir yazılım sisteminin bileşenlerini ve bu bileşenler arasındaki ilişkileri düzenleyen temel yapıdır. Ayrıca, sistemin tasarımını ve gelişimini yönlendiren ilkeleri belirler. Yazılım mimarisi, yazılım geliştirme yaşam döngüsünde temel bir adımdır. Bu süreç, sistemi yönetilebilir birimlere ayırarak anlaşılabilirliğini,

bakımını ve gelişimini kolaylaştırır. Yazılım mimarisi, verimliliği artırmak, geliştirme ekipleri arasında iş birliğini kolaylaştırmak ve sistemin gelecekteki genişlemeler için esnekliğini sağlamak gibi birçok avantaj sunar. Ayrıca performansın ve güvenilirliğin artırılmasına, operasyonel maliyetlerin azaltılmasına katkıda bulunur; bu, kodun yeniden kullanılabilirliğini destekleyen ve iyi tasarım standartlarını uygulayan düzenli bir yapı ile sağlanır [1].

3.2.1.2 Örnekler

3.2.1.2.1 Two-Tier Architecture (Client-Server):

Two-Tier Architecture, yazılım mimarilerinin en basit türlerinden biri olup iki katmana ayrılır: istemci ve sunucu.

- İstemci (Client): Masaüstü uygulamaları veya basit web uygulamaları gibi kullanıcıların etkileşimde bulunduğu kullanıcı arayüzünü (User Interface) içerir.
- Sunucu (Server): Veritabanını ve verilerle ilgili hizmetleri barındırır; istemciden gelen istekleri alır ve sonuçları iletir.

Bu mimari, basitliği ve uygulanabilirliği ile bilinir ancak daha karmaşık mimarilere kıyasla esneklik ve ölçeklenebilirlik açısından sınırlıdır.

Örnek: Basit bir kütüphane yönetim uygulaması, burada istemci veritabanına doğrudan sorgu göndererek mevcut kitapları alır [1].

3.2.1.2.2 Three-Tier Architecture:

Three-Tier Architecture, iki katmanlı mimarinin bir geliştirilmiş halidir ve üç ana katmana ayrılır:

- Sunum Katmanı (Presentation Layer): Kullanıcı arayüzünü içerir (örneğin, web veya mobil uygulama).
- İş Mantığı Katmanı (Business Logic Layer): Verileri işleyen ve sistemle ilgili kuralları uygulayan katmandır.
- Veri Katmanı (Data Layer): Verilerin depolandığı ve işlendiği veritabanını barındırır. Bu mimari, kullanıcı arayüzü, iş mantığı ve veri katmanlarını birbirinden ayırarak bakım kolaylığı sağlar, performansı artırır ve geliştirme esnekliğini artırır.

Örnek: Çevrimiçi rezervasyon sistemleri, kullanıcının web arayüzü aracılığıyla talep gönderdiği, iş mantığı katmanının talebi işlediği ve veritabanından gerekli verilerin döndürüldüğü sistemler [18].

3.2.1.2.3 N-Tier Architecture:

N-Tier Architecture, Three-Tier Architecture'ın bir uzantısı olup, sistemin bağımsız katmanlara ayrıldığı bir yapıdır. Bu yapıda, güvenlik, entegrasyon ve web hizmetleri gibi belirli işlevlere yönelik alt katmanlar bulunur.

- Sunum Katmanı (Presentation Layer): Kullanıcıların etkileşimde bulunduğu web veya mobil uygulamalar.
- Hizmet Katmanı (Service Layer): RESTful API gibi arayüzler aracılığıyla sistemin entegrasyonunu kolaylaştırır.
- İş Mantığı Katmanı (Business Logic Layer): Verileri işler ve işlemleri uygular, iş kurallarını ve sistem mantığını uygular.
- Veri Erişim Katmanı (Data Access Layer - DAL): Veritabanına erişimi kolaylaştırır, veri sorguları veya Entity Framework gibi araçlar kullanır.
- Çekirdek Katman (Core Layer): Diğer katmanlar tarafından kullanılan sınıflar (Classes) ve varlıkları (Entities) içeren merkezi bir katmandır. Bu katman, organizasyonun geliştirilmesi ve yeniden kullanımını destekler; OOP ve SOLID ilkelerine uygun şekilde düzenlenmiştir.

Robert C. Martin'in Clean Code kitabına göre, bu mimari, kodun en yüksek seviyede anlaşılabilirliğini ve bakım kolaylığını sağlamak için katmanları düzenleme ilkesine dayanmaktadır. CORE gibi ayrılmış katmanların yeniden kullanımını ve bileşenler arasındaki bağımlılıkları azaltmasını vurgular [19].

3.2.1.2.4 Microservices Architecture:

Bu mimari, sistemi belirli işlevleri yerine getiren bağımsız küçük hizmetlere böler; bu hizmetler API'ler aracılığıyla iletişim kurar.

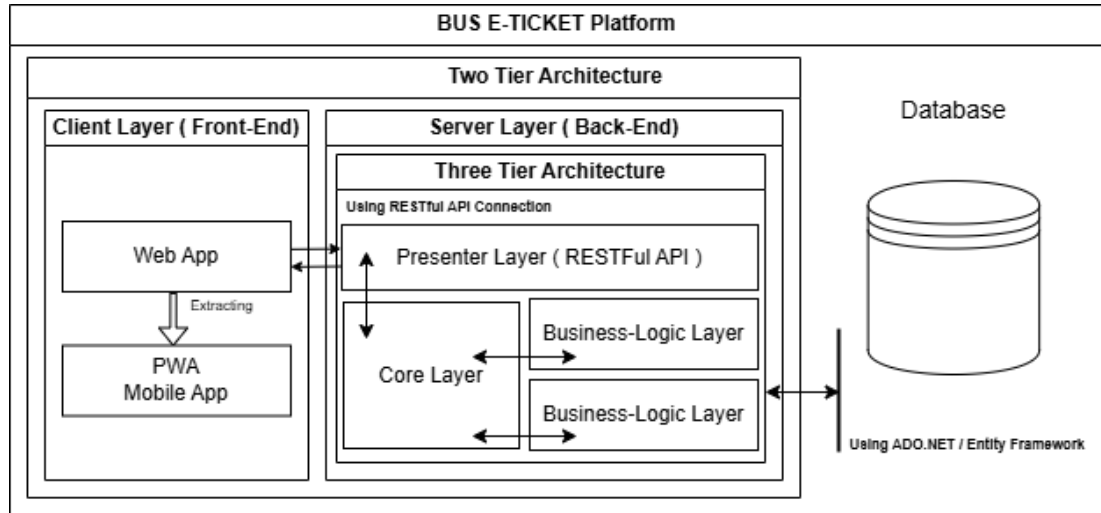
Örnek: Amazon gibi e-ticaret platformları, sipariş yönetimi, envanter ve ödeme için mikro hizmetler kullanır [3].

3.2.1.2.5 Serverless Architecture:

Serverless Architecture, talep üzerine çalışan ve fiziksel sunucu yönetimine gerek duyulmayan bulut hizmetlerine dayanır.

Örnek: Bulut tabanlı sistemlerde görüntü işleme uygulamaları veya hızlı bildirim işlevleri [20].

3.2.2 BUS E-TICKET Platform’da Kullanılan Yazılım Mimarisi



BUS E-TICKET Platform, istemci-sunucu mimarisine dayalı Two-Tier Architecture kullanılarak tasarlanmıştır. Ancak, sunucu tarafında daha karmaşık ve düzenli bir yapı sağlamak için N-Tier Architecture kullanılmıştır.

3.2.2.1 İstemci (Client):

Kullanıcıların etkileşimde bulunduğu kullanıcı arayüzünü (User Interface) temsil eder.

Teknoloji: PWA (Progressive Web App) teknolojisi kullanılarak, çeşitli cihazlarda hızlı ve sorunsuz bir deneyim sunar.

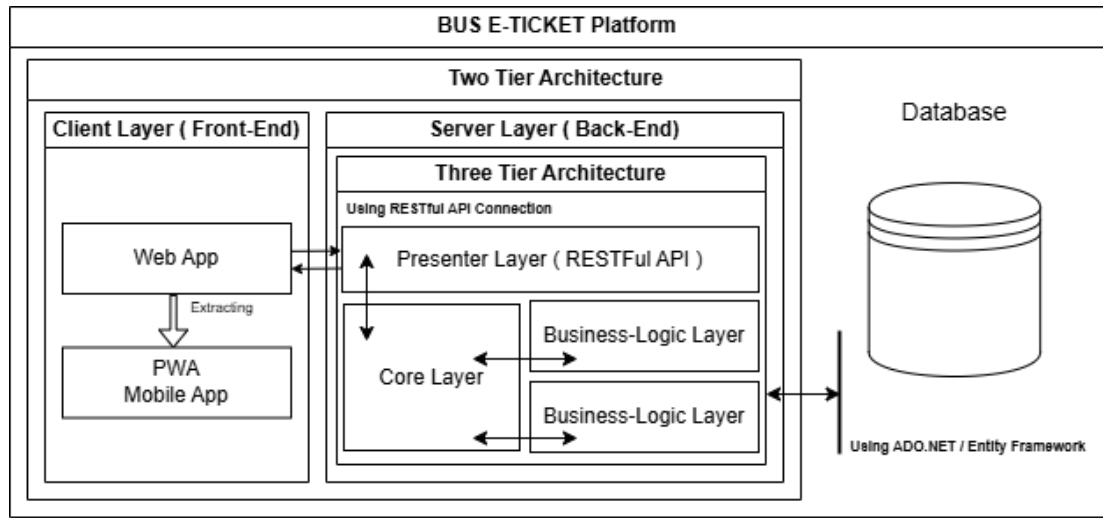
3.2.2.2 Sunucu (Server):

- Sunucu, bağımsız katmanlar ile düzenlenmiş N-Tier Architecture kullanır:
- Veri Erişim Katmanı (Data Access Layer - DAL): Veritabanı ile bağlantıyı sağlar, verileri güvenli ve düzenli bir şekilde işler.
- Çekirdek Katman (Core Layer - CORE): Ortak tanımlar ve temel işlevler içerir; diğer katmanların bağımsız çalışmasını destekler.

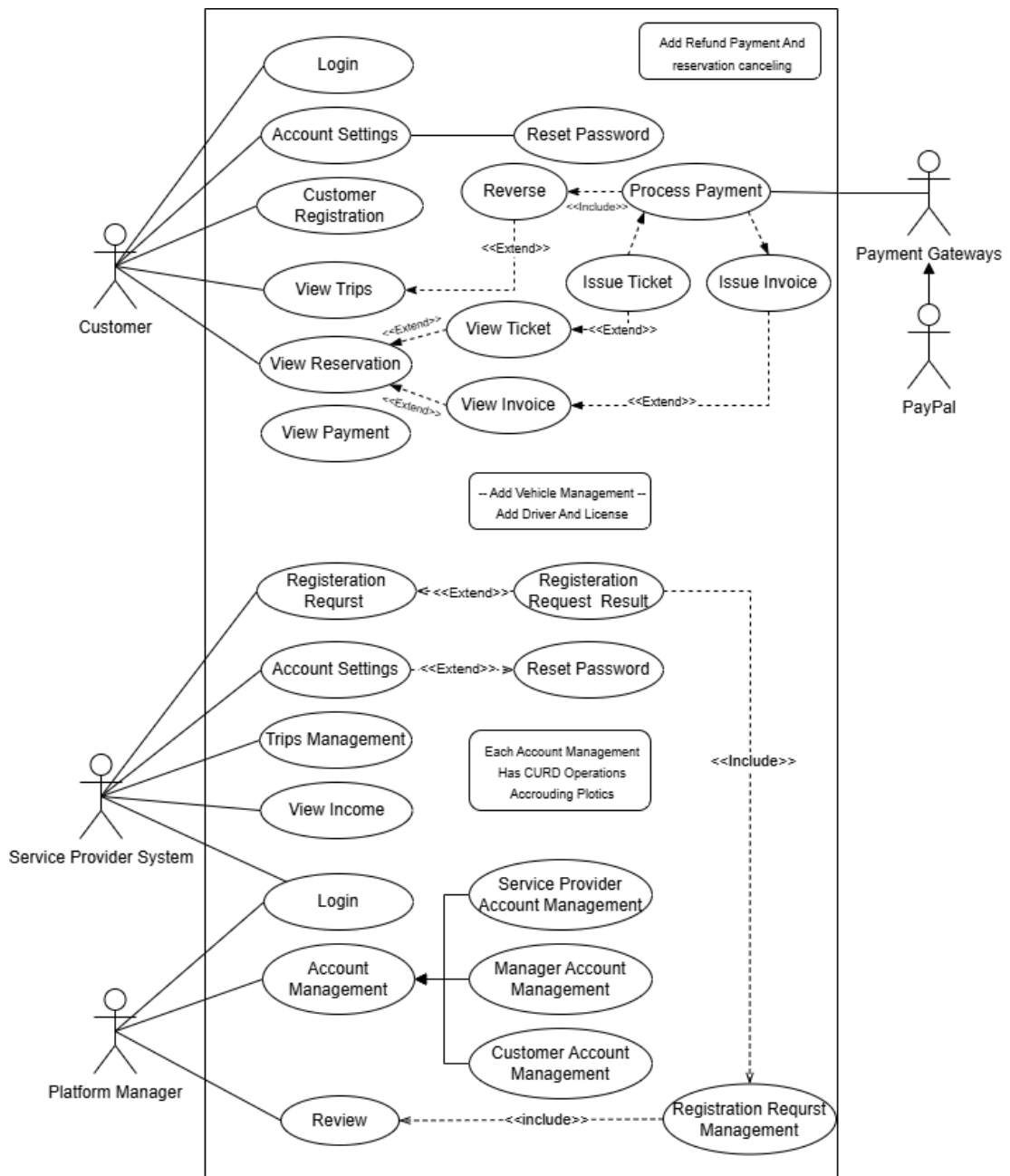
- İş Mantığı Katmanı (Business Logic Layer - BLL): Rezervasyon ve seyahat yönetimi gibi tüm iş kurallarını ve işlemleri yürütür.
- Sunum Katmanı (Presentation Layer - PL): Kullanıcıdan gelen talepleri işler, alt katmanlara iletir ve sonuçları kullanıcıya sunar.

3.3 Sistem tasarımları

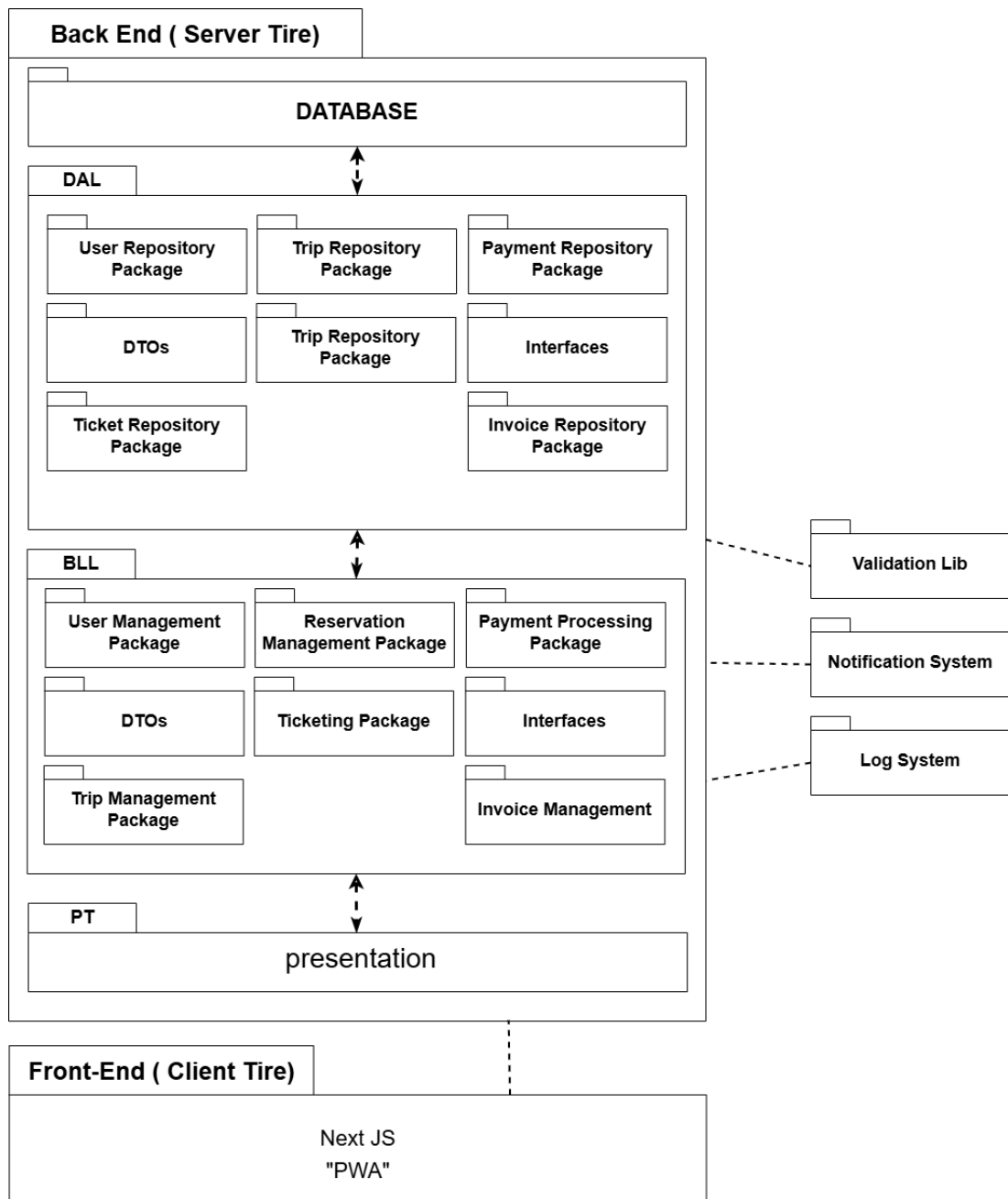
3.3.1 Sistem Mimarisi (System Architecture)



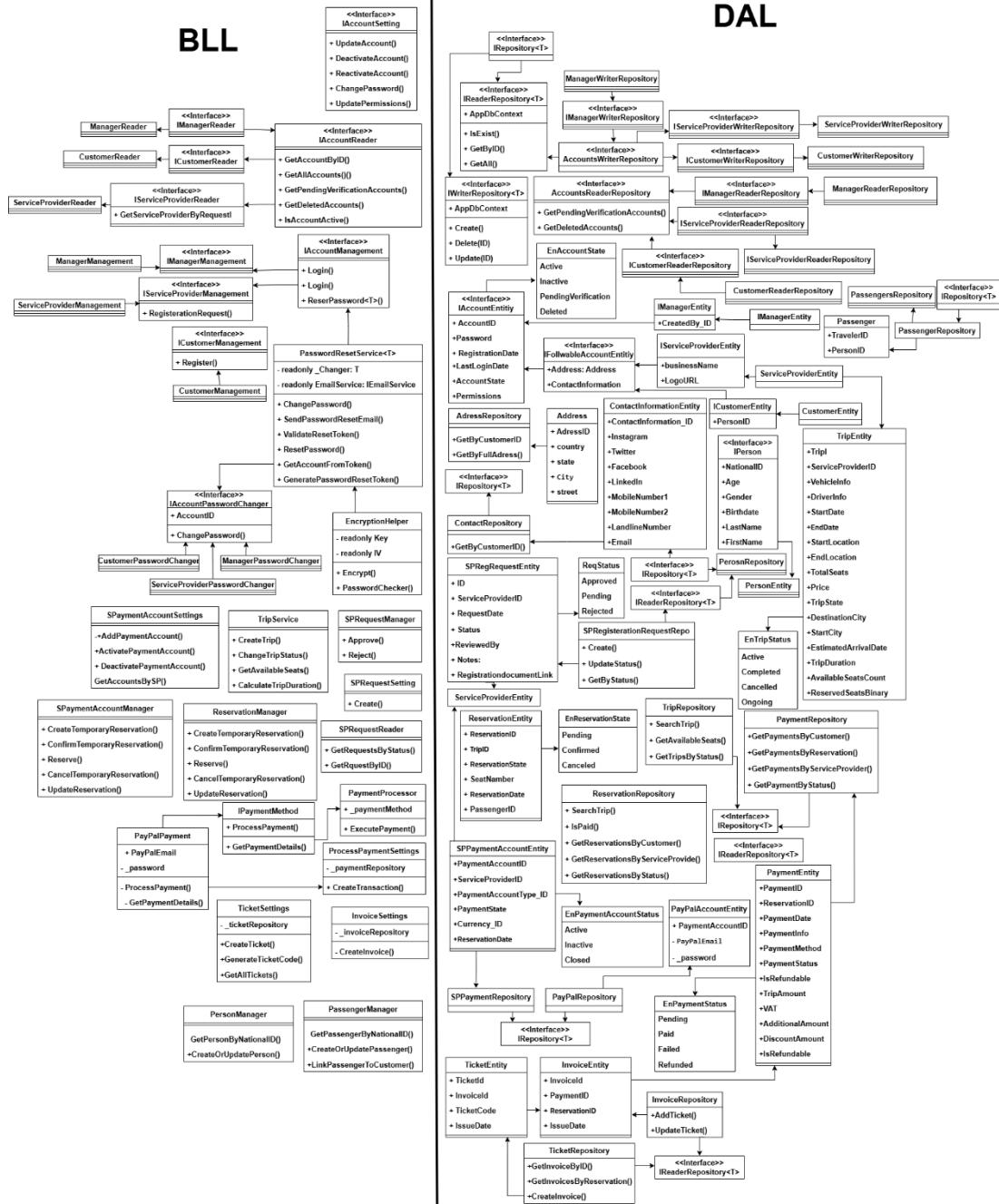
3.3.2 Kullanım Durumu Diyagramı (Use CaseDiagram)



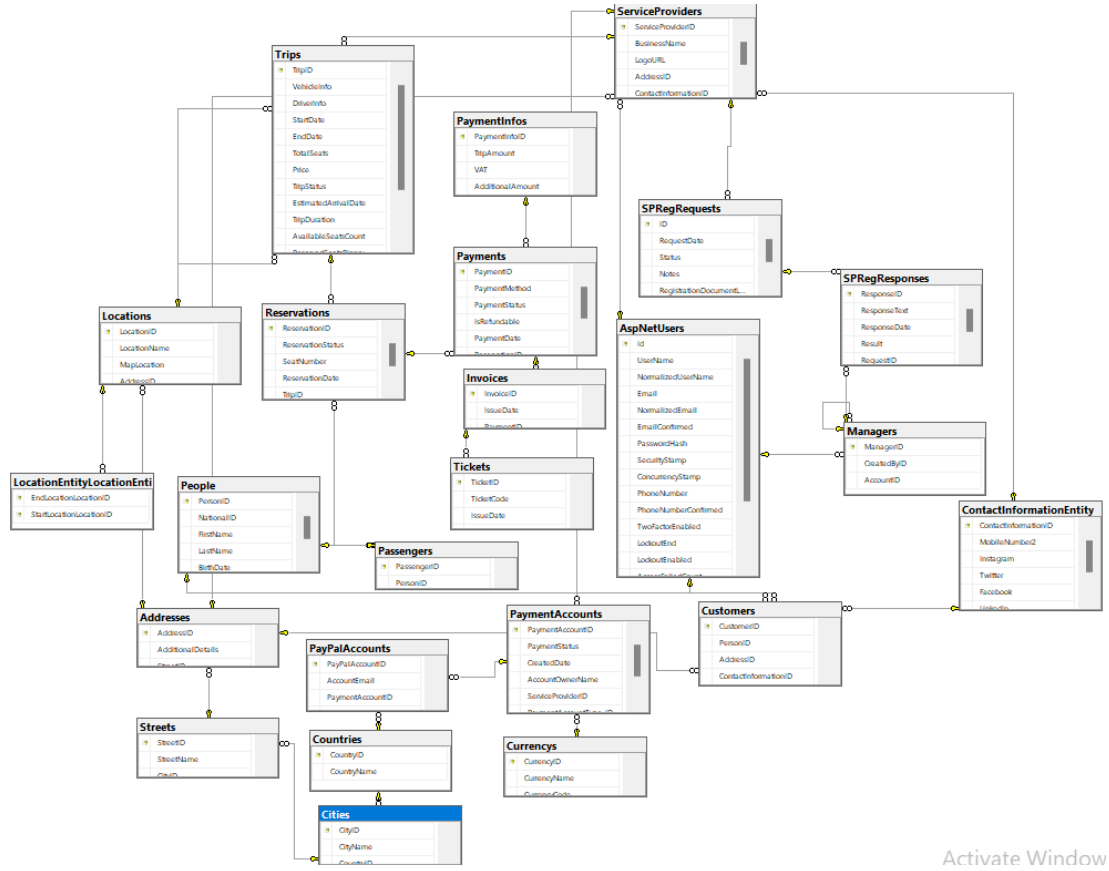
3.3.3 Paket Diyagramı (Package Diagram)



3.3.4 Sınıf Diyagramı (Class Diagram)



3.3.5 Schema Diyagramı (Schema Diagram)



Activate Window

4. YÖNTEM

Proje kapsamında her katman için farklı teknolojiler, diller, kütüphaneler ve frameworkler kullanılmıştır. Bu bölümde, her katman ve kullanılan teknolojiler sıralanmıştır.

4.1 SQL Veritabanı (SQL Database)

Microsoft SQL Server (MSSQL), projenin ana veritabanı olarak seçilmiştir. Bu tercih, güçlü özellikleri ve Microsoft tarafından sunulan kapsamlı destek sayesinde, .NET ortamıyla tam uyumluluk sağlamaktadır. MSSQL, veri yönetimi işlemlerini yüksek verimlilikle destekler ve ASP.NET Core ile Entity Framework Core ile sorunsuz bir şekilde entegre olur [26].

ADO.NET ve EF Core gibi araçlar aracılığıyla esnek ve hızlı işlem yürütmeyi sağlar; bu da sistem performansını artırır ve yanıt süresini kısaltır. MSSQL'in entegre dili olan T-SQL (Transact-SQL), karmaşık sorguların ve veri yönetim işlemlerinin (Stored Procedures, Functions) gerçekleştirilmesine olanak tanır. Bu özellik, sorgu optimizasyonu ile performansı artırır [26].

Gelişmiş güvenlik özellikleri (ör. Veri Şifreleme ve Rol Tabanlı Erişim Kontrolü (RBAC)) sayesinde hassas veriler korunur ve işlemlerin güvenliği sağlanır.

Entity Framework Core, bir ORM (Object Relational Mapper) katmanı olarak çalışır; bu, geliştiricilerin manuel SQL sorguları yazmadan verilerle çalışmasına olanak tanır. LINQ kullanılarak dinamik sorgular oluşturulabilir ve DbContext sınıfı ile esnek bir şekilde veri yönetimi sağlanır. Yüksek performans gerektiren durumlarda T-SQL sorguları kullanılabilir. EF Core ve MSSQL'in bu entegrasyonu, veri yönetiminde büyük bir esneklik ve verimlilik sunar, sistem performansını ve sürdürülebilirliğini artırır [23].

4.2 Arka Uç (Back-End)

Arka uç, modern teknolojilere dayalı olarak ASP.NET Core çerçevesi kullanılarak geliştirilmiştir. Çeşitli kütüphaneler ve araçlar kullanılarak aşağıdaki işlevler gerçekleştirilmiştir:

4.2.1 Diller ve Frameworkler

- C#: Projede kullanılan temel programlama dili.
- ASP.NET Core: Arka uç geliştirme ve RESTful API hizmetleri sağlamak için kullanılan framework.

4.2.2 Veri Erişim Katmanı (Data Access Layer - DAL)

Bu katman, veritabanlarıyla doğrudan etkileşimden ve tüm veri depolama ve geri çağırma işlemlerinin yönetiminden sorumludur. Veri işlemlerinin iş mantığı ve üst katmanlardan ayrılmasını sağlayarak bağımsızlık, bakım kolaylığı ve sistemin genel verimliliğini artırır.

4.2.2.1 Entity Framework Core (EF Core):

- Modern bir ORM (Object Relational Mapper) frameworkü olup, ilişkisel veritabanlarıyla kod aracılığıyla etkileşimi kolaylaştırır.
- LINQ kullanarak, manuel SQL yazmaya gerek kalmadan sorgular yazmayı mümkün kılar, bu da hataları azaltır ve sistemin bakımını kolaylaştırır.
- Veri tabanının detaylarını koddan soyutlayarak bir soyutlama katmanı sağlar.
- Daha düşük seviyeli sorgular için ADO.NET'i kullanır [23].

4.2.2.2 Repository Pattern:

- Veritabanı işlemlerini iş mantığından ayırmak için kullanılır.
- CRUD (Create, Read, Update, Delete) işlemlerini düzenli ve birleştirilmiş bir şekilde gerçekleştiren bir ara katman sağlar.
- Separation of Concerns (SoC) ilkesini uygular, böylece veriyle ilgili sorgular ayrı bir katmana yerleştirilir, bu da kodun test edilmesini ve değiştirilmesini kolaylaştırır.
- Kod tekrarını azaltır ve yeniden kullanımını artırır [24].

4.2.2.3 Unit of Work Pattern:

- Birden fazla veritabanı işlemini tek bir işlem (Transaction) olarak yönetmek için kullanılır.
- Ekleme, güncelleme ve silme gibi birden fazla işlemi nihai kaydetme işleminden önce koordine eder; bu, tutarlılık (Consistency) ve güvenilirliği sağlar.
- Veri kontrolünü artırır ve işlemlerde Atomiklik ilkesini uygular [25].

4.2.2.4 Kimlik Doğrulama ve Yetkilendirme Hizmetleri (Authentication & Authorization)

Kimlik doğrulama ve kullanıcı yönetimi, modern sistemlerde kaynaklara erişimi korumak için önemli bir bölümdür.

- ASP.NET Core Identity: Kimlik doğrulama ve rol yetkilendirme işlemlerini yönetmek için kullanılan framework. Gelişmiş şifreleme algoritmalarını destekler ve yetki yönetiminde verimlilik sağlar [22].

4.2.3 İş Mantığı Katmanı (Business Logic Layer - BLL)

İş Mantığı Katmanı, sistemin işlevsellik kurallarını ve mantığını uygulamakla sorumludur. Bu katman, Veri Erişim Katmanı (DAL) ve Sunum Katmanı (Presentation Layer) arasında bir köprü görevi görür.

4.2.3.1 Servis Sınıfları (Service Classes):

- Service sınıfları, üst katman (ön uç) ile Veri Erişim Katmanı (DAL) arasındaki bağlantıyı sağlar.
- Bu sınıflar, veri yönetimi, giriş işleme ve iş kurallarını uygulama gibi işlemleri gerçekleştiren iş mantığını içerir.
- Verileri geri çağırmak veya güncellemek için DAL'deki Repositories veya Unit of Work ile etkileşime girer [23][24].

4.2.3.2 Arayüzler (Interfaces):

- Arayüzler, uygulamayı tanımdan ayırarak bağımsızlığı artırır ve kodun test edilmesini kolaylaştırır.
- Dependency Injection (DI) prensibini uygulayarak katmanlar arasındaki bağımlılıkları azaltır ve bakım kolaylığı sağlar.

4.2.4 Sunum Katmanı (Presenter Layer)

Sunum Katmanı, İş Mantığı Katmanı (BLL) ile Kullanıcı Arayüzü (Front-End) arasında bir ara katmandır. Bu katman, RESTful API'leri kullanarak verileri düzenler ve kullanıcıya sunar.

- DTO'lar (Data Transfer Objects): Gereksiz verileri azaltarak yalnızca gerekli verilerin taşınmasını sağlar, böylece performansı artırır.
- JSON Formatları: Verilerin düzenli bir şekilde ve ön uçla uyumlu bir şekilde görüntülenmesi için kullanılır.
- Dependency Injection (DI): Sınıfları farklı hizmetlerle ilişkilendirerek kodun bakımını ve test edilmesini kolaylaştırır [23][24].

4.3 Ön Uç (Front-End)

Ön uç, son kullanıcıların sistemle etkileşimde bulunduğu katmandır ve esnek, kullanıcı dostu bir deneyim sağlamak için modern teknolojiler kullanılarak tasarlanmıştır. Ayrıca PWA (Progressive Web App) desteği sunar.

Kullanılan Teknolojiler:

- Next.js: React.js tabanlı bir framework olup Server-Side Rendering (SSR) ve Static Site Generation (SSG) gibi özellikler sunar. Performansı artırır ve sayfa yükleme hızını iyileştirir [27].
- React.js: Dinamik ve hızlı kullanıcı arayüzleri geliştirmek için kullanılan açık kaynaklı bir kütüphane [28].
- Redux: Uygulama durumu yönetimini sağlar ve büyük uygulamalarda performansı artırır [29].
- TypeScript: JavaScript'e tür sistemi ekleyerek hataları azaltır ve kodun kararlılığını artırır [30].
- Bootstrap: Duyarlı ve modern arayüzler tasarlamak için kullanılan açık kaynaklı bir CSS kütüphanesi [31].
- CSS ve HTML: Uygulamanın görselliğini ve yapısını tanımlar [32].
- JSX: React.js ile HTML ve JavaScript'in bir arada yazılmasını sağlar [28].
- PWA (Progressive Web App): Service Workers ve Web App Manifest gibi teknolojilerle, çevrimdışı çalışma ve anlık bildirimler gibi özellikler ekler, mobil cihazlarda uygulama benzeri bir deneyim sunar

5. UYGULAMA

6. SONUÇ VE ÖNERİLER

Ekler

KAYNAKLAR

- [1] World Economic Forum, "The Global Information Technology Report 2013: Growth and Jobs in a Hyperconnected World," Chapter 1.2, 2013. [Online]. Available: https://www3.weforum.org/docs/GITR/2013/GITR_Chapter1.2_2013.pdf. [Accessed: Dec. 15, 2024].
- [2] P. Gupta and A. Gupta, "Exploring the impact of Software-as-a-Service (SaaS) on human resource management," 2022. [Online]. Available: <https://consensus.app/papers/exploring-the-impact-of-software-as-a-service-saas-on-human-gupta-gupta/6bcccd3decf35147ac25ae2dec7f85f5/>. [Accessed: Dec. 15, 2024].
- [3] L. Zhao and F. Guo, "The value creation of B2B2C e-business mode based on SaaS," 2023. [Online]. Available: <https://consensus.app/papers/the-value-creation-of-b2b2c-ebusiness-mode-based-on-saas-zhao-guo/0727695188315c4cbcbcc2304c418f1a/>. [Accessed: Dec. 15, 2024].
- [4] D. Teodorovic and M. Janić, "Public Transportation Systems," *Transportation Engineering*, 2022. [Online]. Available: <https://consensus.app/papers/public-transportation-systems-teodorovic-janic/bcf329c3afbb53b8a9bf67d5325015d6/>. [Accessed: Dec. 15, 2024].
- [5] D. L. Johnson, N. Hayman, and P. Veitch, "The evolution of a reliable transport network," IEEE Commun. Mag., vol. 37, pp. 52-57, 1999. [Online]. Available: <https://consensus.app/papers/the-evolution-of-a-reliable-transport-network-johnson-hayman/d697961c3f6a521dafb5bbe2a0a255f3/>. [Accessed: Dec. 15, 2024].
- [6] R. Buehler, J. Pucher, and O. Dümmler, "Verkehrsverbund: The evolution and spread of fully integrated regional public transport in Germany, Austria, and Switzerland," Int. J. Sustain. Transp., vol. 13, pp. 36-50, 2019. [Online]. Available: <https://consensus.app/papers/verkehrsverbund-the-evolution-and-spread-of-fully-buehler-pucher/c38fbdc7161d57808a8842e043a4765f/>. [Accessed: Dec. 15, 2024].
- [7] W. Kraft and D. Eng., "Opportunities and challenges for integrated transportation," 2001. [Online]. Available: <https://consensus.app/papers/opportunities-and-challenges-for-integrated-kraft-eng/5c4942acf0765273bee9dbc96de8a6df/>. [Accessed: Dec. 15, 2024].
- [8] الهيئة العامة للإحصاء, "ارتفاع عدد ركاب حافلات النقل العام في المملكة خلال عام 2022م," 2023. [Online]. Available: <https://www.stats.gov.sa/ar/news/502>. [Accessed: Dec. 15, 2024].
- [9] وزير النقل يدشن مشروع النقل الجماعي بين المدن في السعودية, "صحيفة الجزيرة," 2023. [Online]. Available: https://www.al-jazirah.com/2023/20231018/ec2.htm?utm_source=chatgpt.com. [Accessed: Dec. 15, 2024].

- [10] "أسماء وأرقام شركات النقل الجماعي في السعودية 2024", موقع خَزَنَة, 2024. [Online]. Available: <https://post.5aznh.com/mass-transportation-companies-ksa>. [Accessed: Dec. 15, 2024].
- [11] "O-Bilet and EnUcuz: Digital integration in Turkish intercity transportation," 2023. [Online]. Available: <https://www.o-bilet.com>. [Accessed: Dec. 15, 2024].
- [12] "Booking.com: Enhancing global transportation integration," 2024. [Online]. Available: <https://www.booking.com>. [Accessed: Dec. 15, 2024].
- [13] IEEE, "Stakeholder Analysis in Software Project Management," IEEE Transactions on Software Engineering, vol. 46, no. 3, pp. 256-269, Mar. 2020. DOI: 10.1109/TSE.2020.2966432.
- [14] Backlinko, "SaaS Statistics: Growth Trends and Adoption," 2023. [Online]. Available: https://backlinko.com/saas-statistics?utm_source=chatgpt.com. [Accessed: Dec. 15, 2024].
- [15] Grand View Research, "B2C E-Commerce Market Size, Share & Trends Analysis Report," 2023. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/b2c-e-commerce-market?utm_source=chatgpt.com. [Accessed: Dec. 15, 2024].
- [16] Backlinko, "Shopify Growth and Adoption Statistics," 2023. [Online]. Available: https://backlinko.com/saas-statistics?utm_source=chatgpt.com. [Accessed: Dec. 15, 2024].
- [17] Capital One Shopping, "Shopify Statistics: Latest Trends & Data," 2023. [Online]. Available: https://capitaloneshopping.com/research/shopify-statistics/?utm_source=chatgpt.com. [Accessed: Dec. 15, 2024].
- [18] D. L. Johnson, N. Hayman, and P. Veitch, "The evolution of a reliable transport network," IEEE Commun. Mag., 1999.
- [19] R. C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008.
- [20] W. Kraft and D. Eng., "Opportunities and challenges for integrated transportation," 2001.
- [21] Micr. Docs, "Entity Framework Core Overview," Microsoft, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/ef/core/>.
- [22] Micr, "Introduction to ASP.NET Core Identity," Microsoft Docs, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity>.
- [23] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.

- [24] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994.
- [25] Microsoft Documentation, "EF Core Overview," 2023. [Online]. Available:
- [26] E. Rahman, SQL Server 2019 Administration Inside Out, Microsoft Press, 2019.
- [27] Micr, "Next.js Documentation," 2024. [Online]. Available: <https://nextjs.org/docs>.
- [28] React Documentation, "React: A JavaScript Library for Building User Interfaces," 2024. [Online]. Available: <https://reactjs.org/docs/getting-started.html>.
- [29] Redux Documentation, "Redux for State Management," 2024. [Online]. Available: <https://redux.js.org/>.
- [30] Micr, "TypeScript Documentation," 2024. [Online]. Available: <https://www.typescriptlang.org/docs/>.
- [31] Micr, "Bootstrap Documentation," 2024. [Online]. Available: <https://getbootstrap.com/docs/>.
- [32] W3C, "CSS & HTML Overview," 2024. [Online]. Available: <https://www.w3.org/>.
- [33] Google Developers, "PWA Overview," 2024. [Online]. Available: <https://developers.google.com/web/progressive-web-apps>.

ÖZGEÇMİŞ

